**source code -** [**https://github.com/chen10an/airbnb-rec**](https://github.com/chen10an/airbnb-rec)

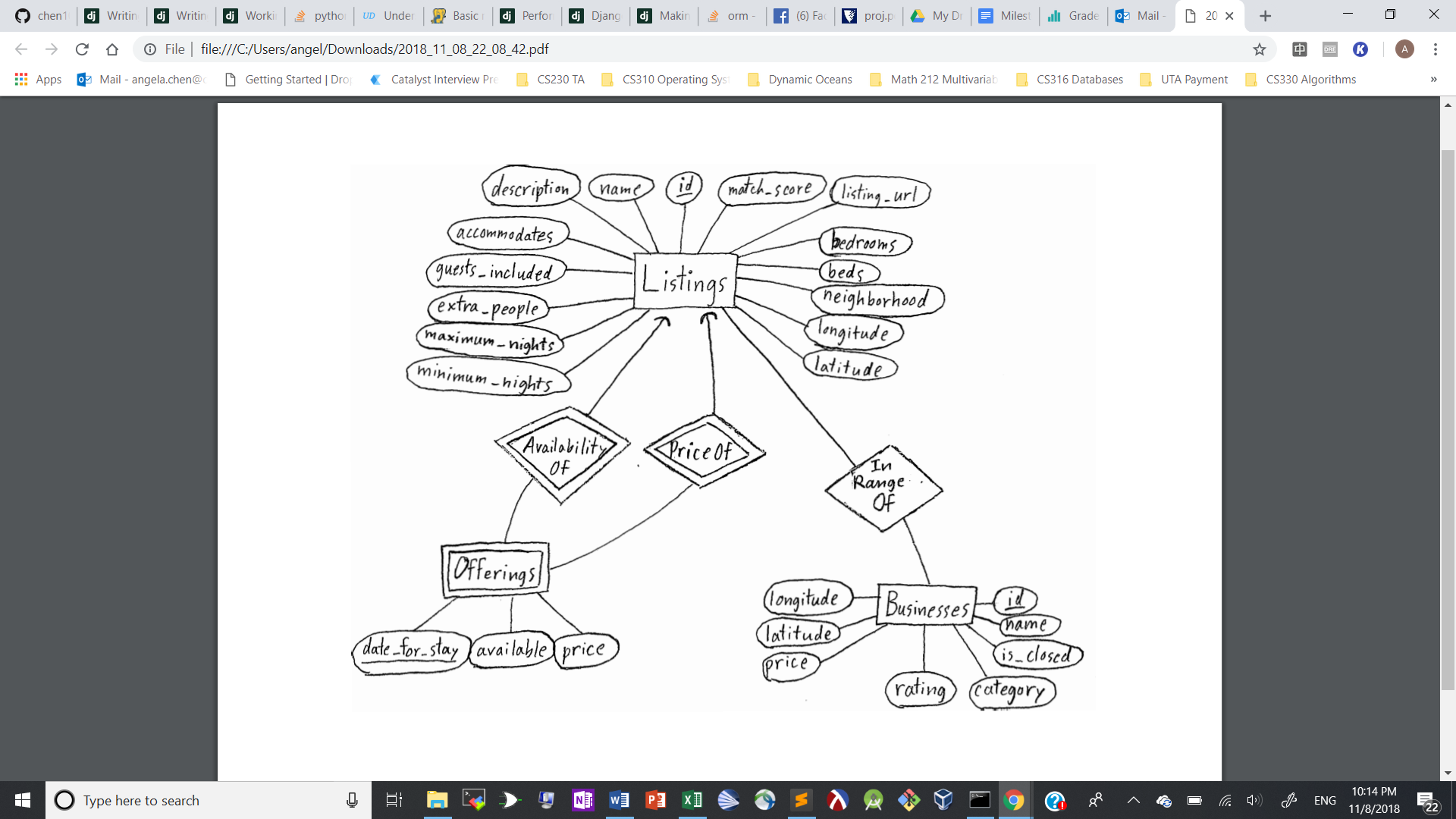
Project Milestone 2.

We have decided to implement an additional aspect to our platform so that it not only mirrors Airbnb, but it also pairs with Yelp to give users the option to select Airbnb listings based off of their food preferences, as well as housing preferences.

Updated Model

We are implementing this food feature into our original feature model. After a user selects their housing preference filters on the first page, the listings will be filtered by those preferences. Next, the user on the second page will have the option to select additional features pertaining to food. We will perform this analysis using the scraped Yelp database. The features they can select from are diversity of food, type of cuisine, type of restaurant, price, and dietary restrictions. They will also rank these features based on importance. Diversity of food will be ranked on a numerical scale of importance. Type of cuisine, type of restaurant, price ($ $$ $$$ $$$$), and dietary restriction will be from a drop down list. Instead of using a black and white filter system on these features, we will use a weighted model to assign the listings (that meet the first page filers) a score and then sort the listings by best score. In order to assign a score, we will be using the Yelp database and a distance radius from the listing’s latitude and longitude. We will count how many restaurants in the area meet each of the features within the set distance of the listing. We will normalize these counts from 0-1 for each feature. Then based off of how the user ranked the importance of features, we will multiply the weight of importance by the normalized count and then sort the listings off that score.

Updated ER Diagram:



Current tables may be found in the creating.sql file. They consist of Listings and Offerings. Another table (Businesses) may be added with the attributes shown above.

CREATE TABLE Listing

(id INTEGER NOT NULL PRIMARY KEY,

match\_score REAL,

listing\_url TEXT NOT NULL,

name TEXT NOT NULL,

description TEXT NOT NULL,

accommodates INTEGER NOT NULL,

guests\_included INTEGER NOT NULL,

extra\_people INTEGER NOT NULL,

bedrooms INTEGER,

beds INTEGER,

neighborhood TEXT NOT NULL,

latitude REAL NOT NULL,

longitude REAL NOT NULL,

minimum\_nights INTEGER NOT NULL,

maximum\_nights INTEGER NOT NULL);

CREATE TABLE Offering

(listing\_id INTEGER NOT NULL REFERENCES Listing(id) ON DELETE CASCADE ON UPDATE CASCADE,

date\_for\_stay DATE NOT NULL,

available CHAR(1) NOT NULL CHECK (available='t' OR available='f'),

price MONEY,

PRIMARY KEY(listing\_id, date\_for\_stay));

Chosen Platform:

We have chosen Django as our platform because Django is quick for out-of-the-box setup: frontend embedding, backend server, and database integration. Additionally, Django allows us to program in mostly python, which has a mature ecosystem for data manipulation tools. Thus, it is easy to plug our Airbnb and Yelp data preprocessing and analysis into an application backend.

Through the Django API, we set up the PostgreSQL database management system, backend serving, and templates for dynamic html generation. For the data preprocessing and analysis, we are using the scipy ecosystem (for analysis) and the requests library (for sending requests to the Yelp API).

Performance:

As of date, there is no noticeable need for adding indexes to improve performance. However, if the need arises after embedding SQL queries into Django, we will likely add indexes on the calendar dates.