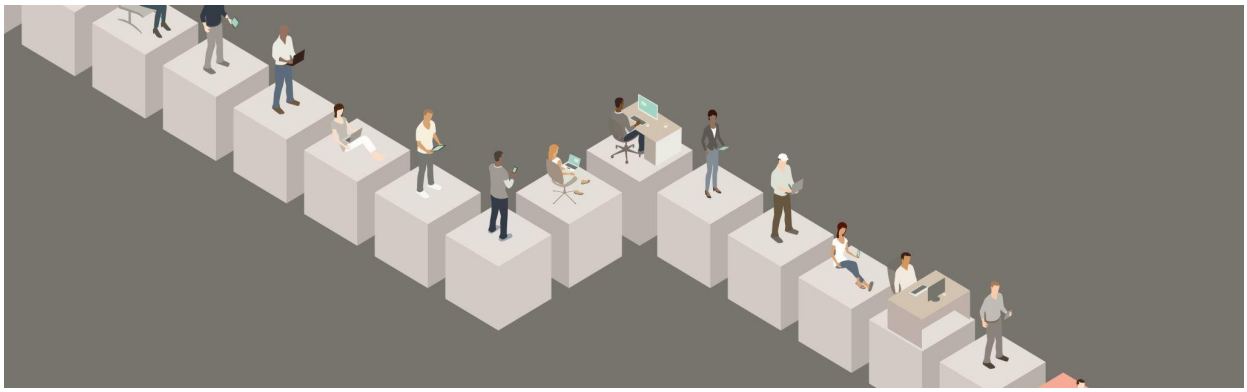
PROJECT PROPOSAL - team065



a. Project Title: Matching restaurants and home-cooks with customers.

b. Project Summary:

This project considers an application which matches customers to food-services. More specifically, restaurants and home-cooks will be able to register to the application, post their menus and receive orders from customers. Some of the features of the application are specifying how expensive a vendor is, displaying ratings for a given vendor and rating a vendor.

Home-cooks are defined as individuals, who do not necessarily own a restaurant, but are willing to sell food to customers, either because they wish to demonstrate their cooking skills, their culture's cuisine, or make extra profit out of this service. Moreover, this enables customers to find food even when most restaurants are past their closing time.

c. Description:

This application aims to expand food-service choices for customers wishing to receive food services by extending food-service hours, offering a greater cuisine variety, and possibly a larger area of service. Moreover, home-cooks wishing to test their cooking skills and showcase their

are enabled to do that. Restaurants will also be included in the application as they are the main food-service providers in most areas.

This ensures that local home-cooks are able to profit from part-time (or even full-time) cooking while the customers will have more food options available. Restaurants can profit from this application as well given that a part of the customer population will choose a restaurant as its dining option.

d. Usefulness:

This application is similar to applications such as Grubhub, or Doordash. What makes this application novel is the inclusion of home-cooks. Home-cooks that register with the application could provide food services to customers even when most restaurants are closed. Moreover, including home-cooks offers a greater variety of type of foods, since home-cooks are able to offer dishes produced according to their home/local culture that is not represented by a restaurant in the area.

Another benefit is that customers located in an area with no restaurants nearby (e.g. suburban, rural) could be potentially served by a local that is registered with the application and provides such services. Restaurants will be included in this application since their operating hours could be more stable than those of the home-cooks offering a more robust selection of food-services.

e. Realness:

The data will concern details about various food-service providers, that is, restaurants and home-cooks (e.g. providerId, location etc.) for a choice of areas/cities, customer data (e.g. customerId, name etc.), order data (e.g. orderId etc). It will possibly include data about customer experiences (e.g. review ratings). The data will either be synthetic (e.g. for home cooks, or orders) or real from sites such as https://datasetsearch.research.google.com/.

f. Functionality:

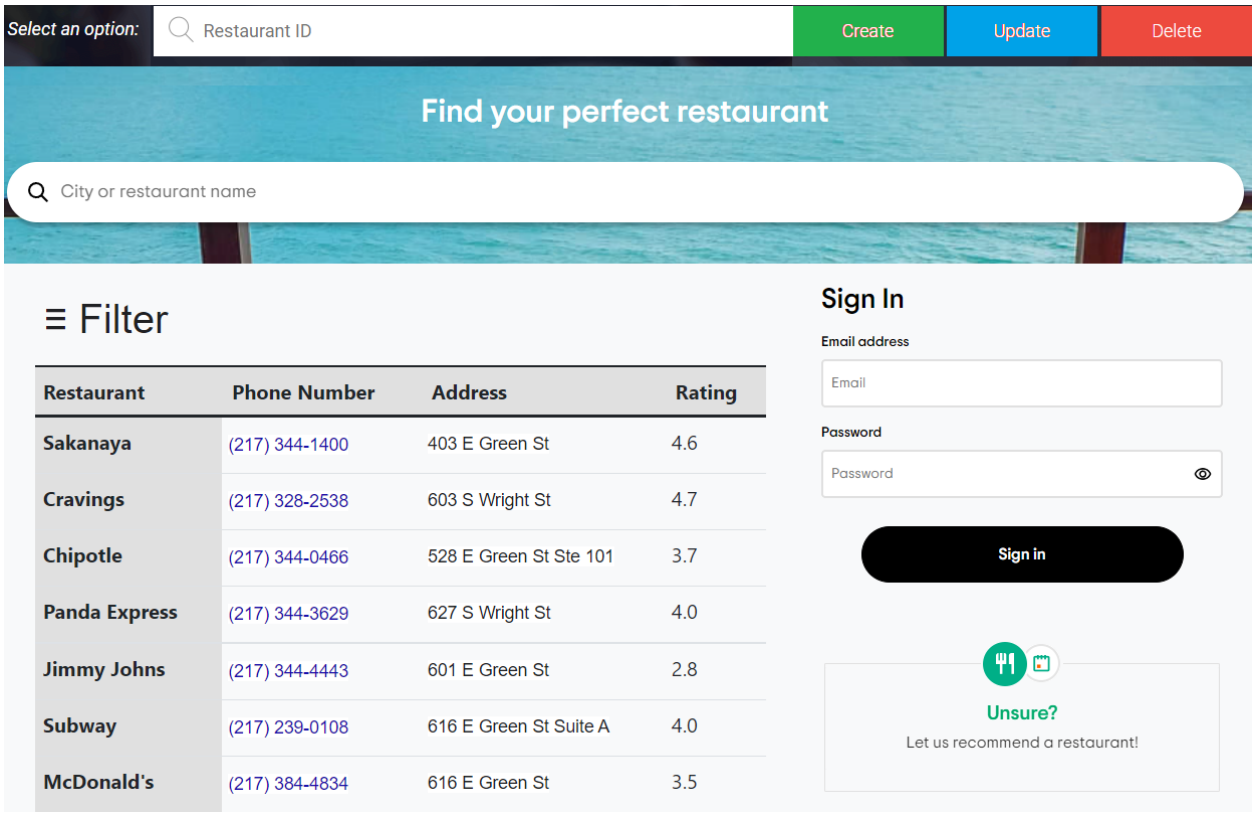
The tables and their attributes are presented below:

- User(userId, phoneNumber, Address, firstName, lastName, orders, orderId, complaints)
- Orders(orderId, vendorId)
- Vendor(vendorId, phoneNumber, Address, firstName, lastName, vendorName, type, menu)
- Menu(restaurantId, itemName, price)

- Coupons(couponId, discountPercentage, expiryDate)

The functionalities of this application that concern CRUD and Search operations include inserting new rows (e.g. new dishes) to the database, or allowing the user to search for an entity (e.g. restaurant) and return the result to the interface. Moreover, they include updating records on the database (e.g. update existing prices for dishes) and deleting rows from the database (e.g. a dish is no longer being made, or a restaurant delete its account with the application). Also, they will include advanced SQL queries such as Subqueries, and Aggregation and Group By (e.g. when a customer asks for the average price of all the dishes of a restaurant). Moreover, functionalities will contain triggers (e.g. when a customer enters an invalid restaurantId a trigger will be activated) and procedures (e.g. when a user wants to calculate the average price of all the items of a restaurant).

g. A low fidelity UI mockup:



h. Project work distribution:

Aviral- backend and creating the API.

Neil - Creating web interface.

Wenjun/Theodoros - How to host API on GCP, how to host website on GCP.

Each will be responsible for one or two tables, i.e., will be responsible for writing related-queries, maintaining it and finding or creating the data to populate the table.