

Xilin Wang, Xinyue Wu

Project Description:

Restaurant is the theme of our project and we would like our database to contain useful information to help the user find the restaurant they like. Since restaurants everywhere are slowly reopening, customers might feel confused about the dining options available (Dine-in, Take-out, Delivery, Closed, etc). Thus, the interesting part about the project is how we collect the order options of each restaurant and update it on time. The challenging part is that we want to offer as much information as possible to users (such as the cuisine, dietary restrictions, whether dine-in is available, etc) while avoiding redundancy.

Entities that we'll have include Restaurant, Users and then details like menu, location, customer reviews, employee reviews about specific restaurants. First, we will have users sign up and we'll take their email addresses and user id, which are both required to be unique. The users can choose a screen name or just use their real name, which can be the same for different users. In the Restaurant entity set, the primary key is the id number we assign to each restaurant. The restaurant can operate in different locations since there are a lot of fast-food chains and we want to account for that. Users can view different order options. We have dietary needs as a normal entity set. The primary key is the type of the special dietary need (ex.halal, vegan). The user review contains a specific description of their experiences and a rating. We will also have employee reviews. The primary key is the id of the review. The users can view the employees' occupation, rating of their working environment, and the specific description of the restaurant.

We will follow the web front-end option for part 3.

Data Plan

We would like to populate our database with real life data that we will scrap from Google Map and various food-related websites such as yelp.

User Interaction description

The entities involved in our application are: Restaurants, information about the restaurant (Locations, Order Options, Menu, etc) and Users. Similar to Yelp and GoogleMap, we would like our application to either take a user's input of information about the restaurant and return all restaurants that satisfies the user's input or take a user's input of a restaurant name and return all details we have on that restaurant. For example, a user can specify a restaurant category (for instance, "Mexican food") or location (for instance, "70 Morningside Dr") and our application will return restaurants that match the user's inputs if we have them in the database. If we don't we'll let the user know that there's no restaurant that matches the specified requirements. The user can also search for dietary needs and we'll return all restaurants that satisfy the needs. Also, the user will be able to see the link of the menu and the price level of each restaurant (ex. \$/\$/\$/\$/\$/\$). If the user's input is a restaurant name, our application will return the details of that restaurant, such as order options (Dine-in, take-out, delivery, etc), menus, location, recent user reviews and employee reviews. Each order option will show different commission fees (this is for the option of delivery). For dine-in and take-out, the commission fee defaults to 0. Also, we

decided to include employee reviews here because we want the user to be able to support the employees there by taking actions based on how employees reviewed their work experience. For instance, if the employees review their working experience as “Such a terrible place to work in. I’m constantly overworked.”, then the customers might choose not to visit this restaurant since it’s not respecting the employees. We found that the customers are not so aware of the working conditions of the employees and we intend to fill that information gap. Users of our database will also be able to write reviews for restaurants, which will be stored in the database. Users will also be able to specify their dietary needs, which will also be stored in the database. Then, given a user id, the user’s previous reviews, and their dietary needs, our application can recommend the user to other restaurants in the category that matches their preference based on a recommendation algorithm.

Contingency Plan

If the project were to be downgraded, we will still follow the web front-end option. We will delete order options, menu, and dietary_needs. We will only have users, user reviews, employee reviews, and locations for restaurants to keep the diagram simple and straightforward.