

Midterm Project Report

Team Members

Zachary Thomas (thomasza@oregonstate.edu)

Daniel Garlock (garlockd@oregonstate.edu)

Peter Dorich (dorichp@oregonstate.edu)

Alex Edwards (edwarda3@oregonstate.edu)

Ehmar Khan (khaneh@oregonstate.edu)

The problem:

With so many sources to find restaurant reviews, that have different rating systems, formats, and biases, it can be difficult to determine the ideal restaurant to go to. One website may rate an establishment as a five-star restaurant, while another site may rate it on a point scale, or some different system. The main problem is that inconsistency between websites does very little to help people decide where to eat.

In addition, there are multiple options for restaurant rating websites out there. Now a consumer is forced to first decide which website they'd like to use, and then use that to decide which restaurant to eat at. What if restaurant "A" only has reviews on Tripadvisor, and restaurant "B" only has reviews on Eater? If an individual is trying to select between restaurants "A" and "B" this requires them to understand the different formats of the sites and differing grading criteria.

A brief plan on how we plan to address the aforementioned issues:

We will design a site that uses our database to display standardized reviews of restaurants from multiple sources.

Our database will have standardized ratings and combine ratings from various sources to get an average of restaurant reviews. To standardize the reviews per site, we will have to convert the site's score into our own scale, which may differ per site. The method for standardization will require research into the site's own scoring system.

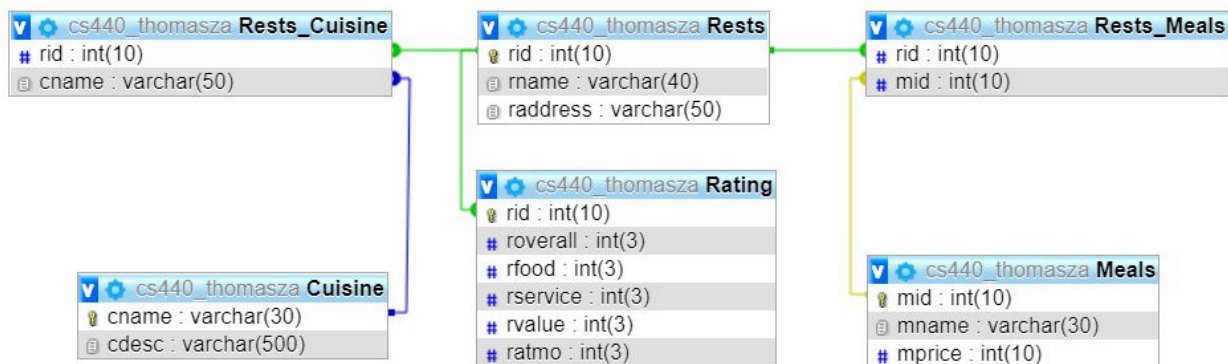
To obtain the data we will use data sets, such as the ones available from Yelp, and Zomato's API. We will design software or use software that can format our data as needed.

What we have done so far, the problems we have faced, and the issues we expect to face in future:

We have designed some tables and the relationships for the database. The main challenge is deciding on data that will be relevant across the world, and across many different restaurant reviewing sites. We have went with a generic address field for now, as some countries do not have states. Additionally some cities share the same name.

We have used generic restaurant IDs and meal IDs. These can simply be incremented as we add new ones to the database. This is to simplify identifying restaurants and meals that may have the same name.

With the wide array of reviewing styles and sources it is likely that we may run into issues with trying to format data to nicely fit into our tables. Additionally, it would be inefficient to manually enter all of our data into our database. This will add an additional challenge of learning and manipulating software to fit our purposes.



The current database is split into the following tables and attributes:

Rests: Stores basic restaurant information.

- **rid:** An incremented ID to easily identify restaurants.
- **rname:** The restaurant name.
- **raddress:** The address of the restaurant.

Rating: Stores information about restaurant reviews. (1 to 1 relationship)

- **rid:** Foreign key that helps match reviews to the Rests table.
- **roverall:** This is the overall rating of the restaurant. The rating is from 1-100.
- **rfood:** This is the rating of food quality. The rating is from 1-100.
- **rservice:** This is the rating for the service. The rating is from 1-100.
- **rvalue:** This is the rating related to price of the food. The rating is from 1-100.

- **ratmo:** This is the rating of the atmosphere. The rating is from 1-100.

**Not all sites may have all aspects of the rating. Allow NULL.*

Meals: Stores information about specific meals.

- **mid:** An incremented ID to easily identify meals
- **mname:** Name of the meal (It is possible for meals to share the same name).
- **mprice:** The price of the meal in USD.

Rests_Meals: M-to-M Relationship of restaurants to meals (May be part of a restaurant chain).

- **rid:** Foreign key that helps match to the Rests table.
- **mid:** Foreign key that helps match to the Meals table.

Cuisine: Stores information about specific cuisine (e.g. Mexican, Chinese)

- **cname:** The name of the type of cuisine.
- **cdesc:** A short description of the cuisine.

Rests_Cuisine: M-to-M Relationship of restaurants to cuisine (May serve more than one type).

- **rid:** Foreign key that helps match to the Rests table.
- **cname:** Foreign key that helps match to the Cuisine table.

We have also found some possible data sources that we can use to populate our database. Some of them do not have open source data, and we can try scraping data from those. A possible data scraping tool we can use is the chrome extension, Web Scraper.

Data Sources:

- TripAdvisor: No open source data or API available (possibly use data scraping).
- Yelp: Large dataset easily obtained.
- Zomato: API available for data access.
- Eater: No open source data or API available (possibly use data scraping).
- NyTimes: No open source data or API available (possibly use data scraping).