

Project 1 - Summary and Report

Group 3: Kien Hoang, Rameel Sethi, Yiming Chen, Bikram Shrestha

[Project Overview](#)

[Database Schema](#)

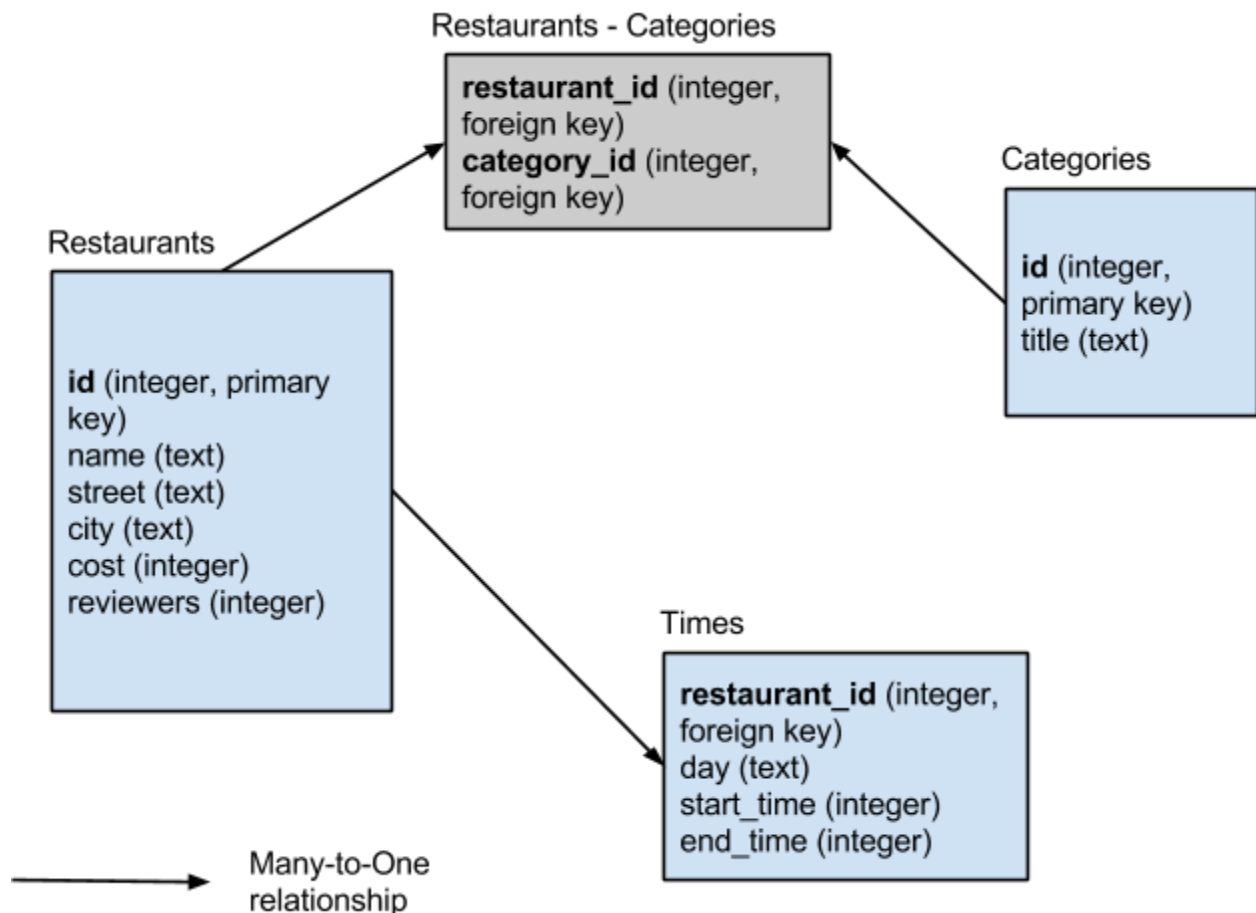
[Project Architecture](#)

Project Overview

The goal of the project is to implement MiniYelp, a Lite version of Yelp with the search functionality on Android. The users can open the application on any Android device and search for restaurants based on the following criteria: city, category, cost, and time. The results are then displayed on a separate screen and sorted by: whether they are OPEN at the time of the search, city, rank, category and cost.

Although the main objective of the project is to build a working version of MiniYelp, another important objective is to design the database that the app uses using standard database design principles that we learned in the class. The schema of the database is covered in the next section.

Database Schema



There's a Many-to-Many relationship between **Restaurants** and **Categories**, so a joint table **RestaurantsCategories** is needed, which stores two foreign keys: `restaurant_id` and `category_id`. There's a **Many-to-One** relationship between **Restaurants** and **RestaurantsCategories**, and between **Categories** and **RestaurantsCategories**, respectively.

And since for each restaurant, there can be multiple opening/closing times, we need a separate table to store all the times, called **Times**, which has a foreign key: `restaurant_id` pointing to the restaurant that has that time. There's a Many-to-One relationship between **Restaurants** and **Times**.

Project Architecture

In terms of the project architecture, we have Object classes such as **Restaurant**, **Category**, **RestaurantCategory**, **RestaurantTime** that represent the rows of the corresponding tables in the database, respectively. There are also classes that represent table operations: **RestaurantTable**, **CategoryTable**, **RestaurantCategoryTable**, **RestaurantTimeTable** that provide operations such as Create on the entire tables. All of these classes are called in **MiniYelpSQLiteHelper**, which is responsible for creating the database and the tables.

Another component of the project is the **InputFileParser**, which is responsible for parsing the input file and persist data to the database, using the table classes mentioned above.

The last component is the **MiniYelpSQLiteHandler**, which constructs queries from the search the user conducts and evaluates, returning the results that will be rendered on the

Results Activity.

