Server App Coding Challenge

A restaurant review software startup provides rating services to consumers. The company needs a RESTful API service to create and query rating info of restaurants. Your task is to design and implement the API service. We have provided the following models and fields as a starting point for you to design the API:

User	Restaurant	Address	Rating
First Name	Name	Address	Cost
Last Name	Category	State	Food
Phone	Address	City	Cleanliness
		Zip code	Service
			Total Score
			Restaurant
			User
			Date

A restaurant category can be Mexican, Burger, French, Thai, Chinese, Japanese, etc. A restaurant may have multiple locations (addresses). A user can only give a rating to a restaurant per location.

Cost, Food, Cleanliness, and Service are required rating criteria with min/max values from 1 to 5 (star) for user to rate the restaurant; the rating must comply with the range constraint. The API will calculate the total score by averaging these criteria. A user can give a rating to the same restaurant no more than once a month. If the total score is 1, the user must provide a comment. The API should return an appropriate error message and error code if the validation failed.

Requirements

The API must provide services for the following requests:

Request	Description
Create a user	A user is created
Update a user	A user is updated
Get user(s) info	Get a single or a list of user
Create a restaurant	A restaurant is created

Update a restaurant	A restaurant is updated
Get restaurant(s) by name / city/ category/total score	Query restaurants with possible filters by restaurant name, city, category or total score. Example: Find Mexican restaurant(s) in San Jose (or zip code) with total score above 3 stars
Create a rating for a restaurant by a user	A rating is created for a restaurant by a user
Update a rating for a restaurant by a user	A rating is updated
Get rating(s) by user	Get a rating or a list of rating given by a user (include all restaurants)
Get rating(s) by restaurant(s)	Get a rating or a list of rating by restaurant. All users who gave the scores to the restaurant(s) will be aggregated to total score.

The data must be persisted in a database. Select a database that best meets your needs and experience. The database is not limited to the four models and fields described earlier. You should design models that make the most sense from the API provider's perspective. You're also free to use any programming language to implement the solution.

Please provide a readme describing how to run the application.

It is desirable that you provide architecture and database design diagrams of the solution to justify the design decisions.

We also suggest not spending more than 5 hours to complete the coding challenge.

Additional Information

At Tesla, we're looking for an exceptional engineer who is willing to go above and beyond! When reviewing your code, there are multiple areas (and bonus points) we're evaluating. Be prepared to explain the implementation and technology chosen. We encourage creativity in your solution, but please try to adhere to good software engineering fundamentals.

Submission

You can submit the solution as a zip file, on GitHub (**preferred**, but make sure it's private), or any appropriate means by which we can review your solution.

Have fun and good luck!