Approach

I took a classic n-layer approach for the software design as I find it's a nice way to separate concerns into repository, service, and presentation layers which allows for easier unit-testing and future feature development. I opted to use Golang for it's portability (can target different operating systems), speed (no JVM, just a simple binary), and most important it's readability. And for me that last reason is the most significant as software is maintained and written by humans so ensuring that we can make sense of it is critical.

Proof of Concept to Production Web App

First and foremost we need to build up some test coverage in this project. After it's working as expected from a unit-test (business logic) perspective one of the next things to consider is, will it scale? Initially I have suspicions that the San Francisco City API is able to handle a load of our program calling their API for every request from one of our users, so since this data doesn't seem to go "stale" quickly I would assume we could set up a memory cache (local, Redis, or an RDB if we have long term storage needs) with a 15 min expiration before reaching out to the API to renew the cache.

With the basic stability of the app established I would then move onto iterative ways in which we can add user value. A simple one is to add an interface that allows them to perform past, or more importantly future, queries as to the location of food trucks in the city based upon a date and time entry.

After that long term plans would include the ability for a user to record a log of a food trucks they've eaten at and provide a rating (let's say out of 5 stars), then with that information we can provide a notification to the effect of *"Hey! You ate at Gary's Gyro Hut last week and gave it 5 stars, did you know they are parked at the same location today?"* Once we've spent a couple of months gathering their food truck location, type, and time preferences we can start making more accurate suggestions such as if we've determined that they eat at the same location regardless of the truck type our alerts can be based around that, *"Hey! Today at 5th and Pine is Pauly's Poke!"*.