

Go - Code This!

You are building microservices for taking flying-cars(Dubai) on hire on an hourly basis. This consists of 2 services:

1. Auth: User signs up to the app and can login in future via phone number and passkey. The user can log out too.
2. Garage: The garage stores a list of flying-cars. The user can login to the app and request a flying-car by sending the unique id for that car + phone number. If the car is unavailable, the response should say the same. Moreover, a list of available flying-cars can be given so that the user has options. For every request, the garage service needs to check with the auth service for a valid id and session for that user. If an account does not exist or if the user is not in session, appropriate message should be sent back to the user.
The user can return the vehicle via a service.

We are going to be using Postgres as the database that the go service connects to. The infrastructure will be containerized using docker-compose and Dockerfile. We need a common network that connects postgres on port 5432 with the go service at port 3000. The backend service will use [gin](#) and [graphql](#).

The areas that will be checked would be OOPS and SOLID concepts applied in Go. Database normalisation in Postgres. Understanding of containerized infrastructure and networking.

Please use Github for version control and frequent commits are a plus!

Auth Service

- SignUp
- Login
- Logout
- SessionCheck

Garage Service

- GetAll
- Book
- Return

Postgres Database

It is safe to say there will be a table for users, another for vehicles. Rest is up to you. Here are the meta data points for each:

User

1. First Name
2. Last Name
3. Mobile Number
4. Pass Key (encrypted will be a plus!)

Car

1. Model (FC-001, FC-010, FC-011 are the available models)
2. Date of Manufacture
3. Last Serviced Date
4. Unique ID
5. Last Used Date