Assessment - Director / Manager of Engineering

Lalamove Technical Challenge

Lalamove is a last-minute intra-city delivery technology in Asia. One of the main goal of the company to build a eco-friendly business that make use of sharing economy to reduce the number of idle vehicle, maximizing the efficiency of each vehicle in the city.

In this challenge, you are required to prototype an interface and backend that serve the following

1. Create an api that randomly create order randomly on a city map (you may assume only use Hong Kong). Order will have the following interface, the pick up and delivery time is always end with 0, 15, 30, 45 in the next hour on the map. (no more than 5 orders can exist on the map)

order info

```
{
  serviceType: 'A'         // assume you have type A, B and C
, pick up time: '13:00'    // pick up time is by every 15 mins
, delivery time: '14:50'
, pickUpLatLng: ['114.53254','21.2314214']
, dropOffLatLng: ['114.2341','23.4543253']
}
```

2. An optimize api that take in 2 driver and split out the best path for the vehicle to take. Notice that none of these vehicle can take no more than 3 parcel on board at the at time.
3. create an api that remove a random order on the map

You may assume the interface is a google map and a table that should us the different vehicle paths and which stop would be late

Technical Spec

The code and architecture will be split between a back-end and a web front-end. Please feel free to use any of the following

- node.js
- PHP
- ruby
- Java

We understand that engineer are usually better in some area and weaker in other, let us know in the readme of which area you are more interested in

1. Full-stack Engineer: include both front-end and back-end.
2. Back-end Engineer: Plain frontend / only API documentation. Focus on making the backend as robust as possible.
3. Front-end track: include a minimal back-end or include static set of data. Make the frontend as polished as possible.

Frontend should be an one page index.html with css linked to it.

Hosting

When you're done, host it somewhere (e.g. on Amazon EC2, Heroku, Google AppEngine, etc.)

How we review

Accuracy - does the program do what it is suppose to do? Does it handle any boundary cases?

Architecture - separation of concern between frontend code and backend code

Clarity - README documentation explaining concerns and step to run the program?

Clean - How good the code quality is, is it easy to understand. Is the coding style consistent with the language / framework's guideline.

Testing - automated testing is heavily used in lalamove. We don't expect full test coverage due to the time constraint. But we look for the sense of the test cases you are writing.

Deliverable

- A link to see the documentation index.html / frontend working html
- A zip file to send Lalamove for code review