Taxi Booking System

API Design Specification

Created by: Tan Si Ling
Latest as of 28th December 2017

Table of Contents

Introduction 3
API Services 3
System Overview 4
Design Consideration 4
Mathematical Computation 5
Future Work 5

Introduction

This document details the system architecture. It provides the rationale and design decisions made.

API Services

S/No	Name of API Services	Description	HTTP Method	URL (Endpoint of API)
1	Book a Taxi	1. Picks a nearest available car to your location. 2. Returns total time for car to bring passenger to their destination.	POST	http://TaxiCustomerA:CustA123@localhost:5000/api/book
2	Increment Time Stamp	Increments the system time stamp by 1 time unit.	POST	http://localhost:5000/api/tick
3	Reset Taxi System	Reset all cars data back to initial state regardless of cars that are currently booked.	POST	http://localhost:5000/api/reset

System Overview

Taxi Booking System provides a list of API. These APIs are REST (REpresentational State Transfer) web service that is using Python and Flask microframework. For authorization of the client calling the API endpoint, it uses a small Flask extension *Flask-HTTPAuth* written by Miguel.

Design Consideration

1. Data in this system is stored in-memory as opposed to persistent storage (e.g. using a database). This constrained the system, such that if the program terminates (i.e. virtualenv is restarted), entire program data is reset and is reinitialized.

The choice of not adopting persistent storage is a fixed requirement.

2. **Algorithm Consideration**

Distance from point A (x1,y1) to point B (x2,y2) is stored as time unit. This is because 1 distance unit corresponds to 1 time unit. The computation of distance unit is based on Manhattan distance, as per requirement.

Time Complexity

O(n) for looping through all the car's current position relative to the customer's position. After that O(1) to compute the time to for each car to reach the customer.

O(nlgn) for sorting {timeToTravelToCustomer, carID} objects, based on first criteria nearest car to the customer and second criteria smallest carID.

Overall time complexity of this algorithm is O(nlgn).

Mathematical Computation

$$\text{Time to travel from point A to B} = |\,(x_2 - x_1)\,| + |(y_2 - y_1)|$$

$$\text{where Point A } (x_1, y_1) \text{ and Point B } (x_2, y_2)$$

Future Work

There are many enhancements that can be made to Taxi Booking System. Some of these include the following:

- Securing a RESTful web services by using https instead of http.
- To move from basic authentication to one that is using OAuth 2.0
- To include authentication for the other 2 APIs ("Increment Time Stamp" and "Reset Taxi System")
- To host this API on a privately own server. Currently, this application is hosted on Flask's built-in-server, only one request at a time.
- Adopting of a persistent memory using database storage.