Candidate Test

Please time-box your work on this task to no more than 2-4 hours. We're more interested in how you approach the problem, than how far through you get.

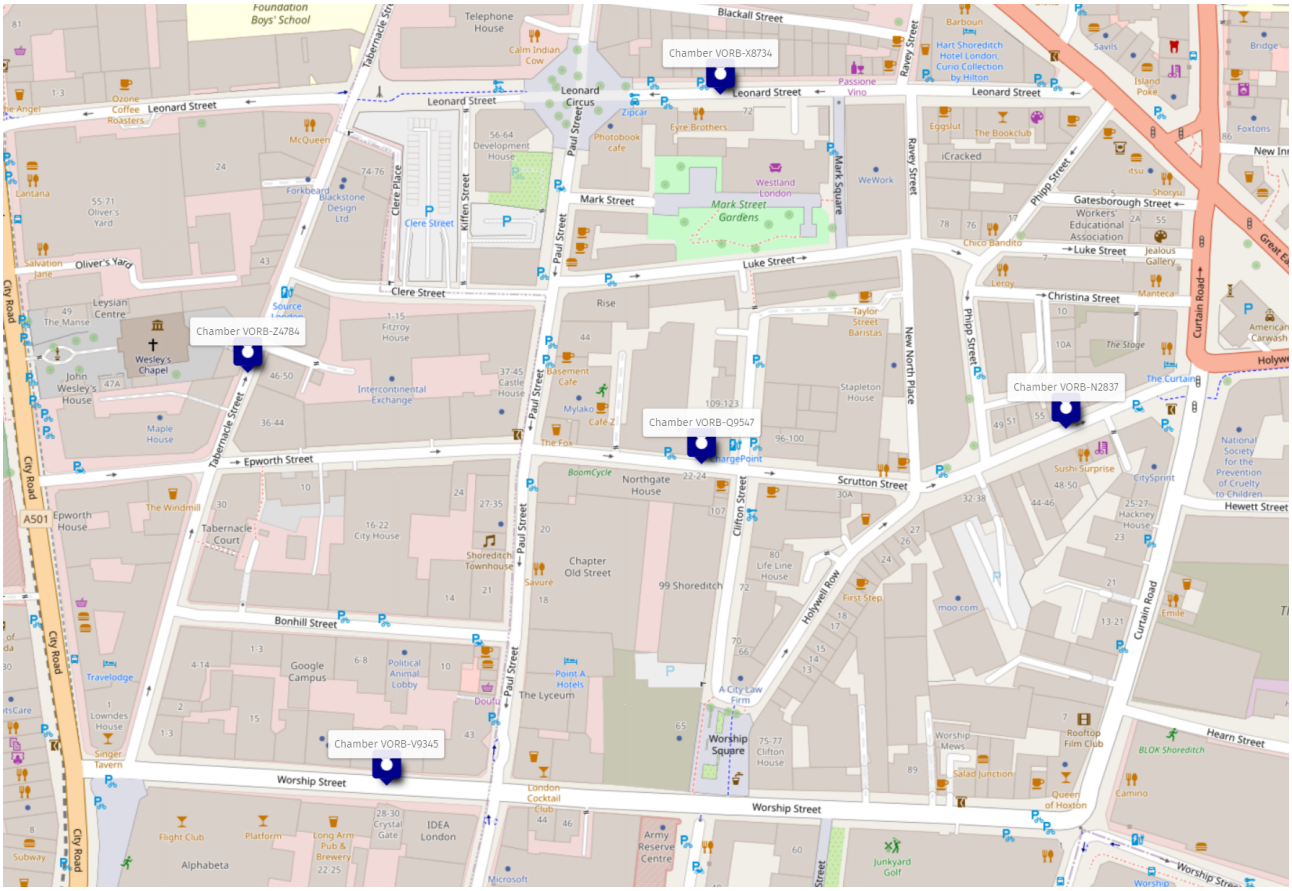
Here at Vorboss, we are building our own Fibre network throughout London. As you can imagine, this comes with a whole bunch of complexities - but the one we'd like you to focus on is the problem of capacity at our connection points.

Chambers

Across London, we have a lot of connection points in the ground - these are called "Chambers". *(Keep your eyes on the ground around Central London and you might see our logo on the chamber lids!)*

Here's a few of the chambers located on a map:

Obviously this isn't real data - don't expect to find a real life chamber at these locations!



Vorboss Fibre Chambers

And a bit more data in a table format:

Chamber ID	Capacity	Used Capacity	Latitude	Longitude
VORB-X8734	100Gbps	70Gbps	51.52466903333144	-0.08320212364196779
VORB-Z4784	100Gbps	10Gbps	51.523641015718525	-0.08601307868957521
VORB-N2837	100Gbps	40Gbps	51.523434943212514	-0.08114755153656007
VORB-V9345	100Gbps	30Gbps	51.52211691871454	-0.0851869583129883
VORB-Q9547	100Gbps	70Gbps	51.523304662537235	-0.08331477642059326

The Problem

We've had an influx of new customers wanting our 10Gbps internet product! We need a way of identifying the closest chamber to the potential customer, ensuring we have capacity to connect them, and give us a heads up when we might need to increase capacity near a chamber.

What we need from you

Write a web application that allows a Vorboss sales team member to enter details about a new customer ordering our 10Gbps internet product. The application should call a backend API that does the following:

1. Persist the details of the new customer in a data store
2. Provide the user with the ID of the closest chamber that has available capacity.
    a. Alert the user if the nearest chamber doesn't have capacity, but still provide the next closest that does have capacity.
3. Alert the user if the provided chamber is now at capacity with the addition of the new customer.

Some important points:

- You can assume there are already customers using the "used capacity" in each chamber. Add some fake customer details to your database for these customers if you need to.
- The sales team member will have the following details about a customer:
    - Customer Name
    - Street Address
    - Postcode

        - Building Latitude
        - Building Longitude
- Eventually all the chambers will become full. Don't accept new customers if we don't have capacity for them!
- Use "direct" paths to the chambers - don't worry about routing along streets

The web application should also include an overview of the chambers and their available capacity.

Include a README file that lays out the following:

- What technologies, libraries, or frameworks you chose, and why you chose them.
    - Note that we require the use of Node.js (or similar runtimes) for your backend
    - While we provide a docker-compose file and postgres database, the data store can be in-memory (either as sqlite or simply an object cache). If your solution isn't what you would choose in production, please discuss your decisions and what you would use in a real-world app in the README
- What design decisions you made when choosing the UI layout and the user flow
    - Why did you make the application flow in this way? What other options did you consider?
- How to run your application locally.

We'll want to talk through your solution in a future interview. Start thinking about how it could be expanded for new products, new chambers, and new functionality.