**Quartet Health –Take Home Assessment**

Arindrima Datta

5/4/2017

This take home exercise is done in a way to set up the foundations for four different areas as described below:

1. Initial load of the data
2. Establishing a nightly job to pull in the new facilities added to this master list and update our data store
3. Identifying any clinic in this NYC master list that match active clinics within Quartet
4. Monitoring and alerting to let you know if the nightly update and match jobs fail.

As a result the technology chosen is simplistic and serves the general purpose of the tasks. This can be thought as a narrative complete prototype as opposed to a production level implementation. Hence, there are plenty of scopes for improvements, some of which are described in this write-up.

I have set up a python flask app with a postgresSQL database, and a database model (schema) created with SQLAlchemy. This prototype has a functioning model layer (out of the MVC layers) and can be used as a backbone for any UI (web, console or otherwise).

The database has two tables

1. **Clinics**: that holds the master list of NYC clinics. While loading this database with data from the API, I checked for duplicates and stored only the unique clinics. The logic to check for duplicates and eliminate false positives is described later.

The clinics table has the following fields

* id
* name\_1
* name\_2
* street\_1
* street\_2
* city
* zip
* latitude
* longitude
* active (whether this clinic is active in Quartlet)

1. **Quartet**: this is a table that lists all the clinics active inside Quartet. This table has the following fields

* id
* name\_1
* name\_2
* street\_1
* street\_2
* city
* zip

In the data records collected from the API, there was no unique identifier for the individual clinics. Therefore, I created a UUID using the following logic:

* Concatenated the name\_1, street\_1, city, zip of each clinic,
* Made the string lower case,
* Computed the hash and
* Took the last 8 digits of the hash.

In using this logic, I am assuming that clinics, which are in the same street address and have the same zip code, and have the same first name, are the same. Therefore clinics at the same street location and zip code but having a different second name (name\_2) are considered to be the same clinic. I interpreted the name\_1 and name\_2 as Management Company and branch names, but sometimes the clinics don’t have branch names, so it is not possible to compare clinics based on name\_2. Similarly, clinics with the same street location and zip code but different apartment units (street\_2) were grouped together (clinics in the same building were considered the same). This logic is used to check for duplicates while initially loading the data, as well as checking for whether these were active in the Quartet database.