COMP 1510 Final-Exam Hackathon Project Proposal

Ronald Liang – A01199458
April 13, 2020

The Proposal

The Python application that I propose will be designed to inform the user of the lowest-traffic and open grocery stores and supermarkets close to their location at the current time. This will help mitigate the risk of the COVID-19 disease spreading to or from the user while they shop for essentials by avoiding groups of people.

Project Dependencies

This project will require the querying of services with detailed geographic information systems data, including:

- Places,
- Place type (to differentiate supermarkets and grocery stores),
- Operational status (inferable from hours of operation),
- Real-time foot traffic, and
- Place name and address.

The services must also support querying for places relative to a provided location. In order to fulfill these requirements, the project will require the following dependences:

- Google Places API, and
  https://developers.google.com/places/web-service/intro
- Populartimes library.
  https://github.com/m-wrzr/populartimes - under MIT license

The Populartimes library is required in order to query real-time foot traffic data and is dependent on the Google Places API. The Google Places API supports all other requirements.

User Experience Design

The application will be operated from the terminal. The user will be prompted for a latitude and longitude in decimal degrees as well as a search radius in meters. Given inputs that would result in at least one currently open supermarket or grocery store located in the search area, the application will print a list of up to twenty places, sorted from least to most by current relative popularity. Useful place details will be included where available, such as the address and phone number for contact. Other cases, such as finding only closed locations or no locations at all, will have useful informative messages, informing the user of the specific case and how they might improve the search.

Technical Design

The application will be fully and properly documented and unit tested. All possible sources of runtime exceptions such as failed get requests, improper query responses, and improper user input will be handled by a suite of data validation techniques, possibly including but not limited to if-statements and exception handling.

HTTP get-requests will be used to query from the APIs. The responses are expected to be JSON data which will be parsed into python dictionaries. Given valid inputs, the Google Places API will be queried for a collection of desired places. These places' current popularity will then be queried for by methods in the Populartimes library with the place ID provided by the initial Google Places API query.