Project Design

1. Description of the REST APIs I will use:

REST API: Opendata Vancouver

URL: https://opendata.vancouver.ca/api/explore/v2.1

Documentation: https://opendata.vancouver.ca/api/explore/v2.1

Description: I will fetch data about Vancouver libraries and rapid transit stations from the Opendata Vancouver REST API. I will use the data to display information about library names, addresses, geographical coordinates (latitude and longitude), and additional details.

Endpoints:

*/catalog/datasets/libraries/records?limit=100&offset=0&timezone=UTC&include_links=false&include_app_metas=false

- get a dictionary of library records. The value corresponding to the key "result" is a list of dictionaries of library information.

*/catalog/datasets/rapid-transit-stations/records?limit=100&offset=0&timezone=UTC&include_links=false&include_app_metas=false

- get a dictionary of station records. The value corresponding to the key "result" is a list of dictionaries of station information.

2. All features (and short description of each):

The following features use the same models and endpoints:

Model: Library, Station

REST API endpoint:

/catalog/datasets/libraries/records?limit=100&offset=0&timezone=UTC&include_links=false&include_app_metas=false

/catalog/datasets/rapid-transit-stations/records?limit=100&offset=0&timezone=UTC&include_links=false&include_app_metas=false

i. ## Feature: display library-station map with different center points

Description: a Vancouver map with markers of all libraries and rapid transit stations is shown by default. Users are able to choose a certain library from a select

box to get the names and distances of nearby transit stations accordingly. The map is then refreshed to have the chosen library and its nearby stations only. Clicking on the markers of library and station will have their names displayed. Clicking on the names of libraries then enables users to jump to their public website.
Page of view: Maps

ii. ##Feature: create a want-to-go list
Description: users can add libraries into a want-to-go list and review their list in a second page. The want-to-go list contains the names of libraries, nearby station names, library addresses.
Page of view: Favourite list

iii. ##Feature: display library and station information
Description: all the information of libraries and stations can be displayed when chosen. Users will be able to select from a radio and see a table of stations or libraries.
Pages: Data Display

3. Interaction to have users make that impacts data display and / or visualization:

i. Interaction with the map. As mentioned above, users can enlarge and detail the map, click on markers to see more information and make selections in a select box to change markers displayed on the map.
ii. Interaction with the library information. By selecting libraries from a select box, users can have a table displayed consisting useful information regarding with nearby stations.
iii. Interaction with tables of data of library and stations. A radio enables users to switch between displaying tables of library data or station data.
iv. Interaction with the want-to-go list. By clicking on the "Add" button, users are able to add libraries into the list.

4. **All specific data classes (description can be your docstrings of the classes) + list of objects:**

i. Library (class): represents a library with attributes of name, longitude, latitude, url link and address; has the method find_nearby_station(), which is to return a list of station objects near this library.

ii. Station (class): represents a station with attributes of name, longitude and latitude; has the method calculate_distance_from_library, which is to calculate the distance from the station to a given library.
iii. library_list: a list of library objects, created by the function create_list_of_library_objects()
iv. station_list: a list of station objects, created by the function create_list_of_station_objects()

5. **List of all data structures:**

List:

i. library_list and station_list as mentioned above.
ii. distance_list: consists of distances from a library to its nearby stations. The reason I chose to use two lists (nearby_station_name_list and distance_list) instead of a dictionary is the requirement of the DataFrame method in pandas.
iii. nearby_station_name_list: a list of name attributes of nearby stations.
iv. library_name_list, library_url_list, library_address_list: list of corresponding library attributes, created by iterating on each library objects.
v. station_name_list, station_longitude_list, station_latitude_list: list of corresponding station attributes, created by iterating on each station objects.
vi. favourite_list: a list of users' chosen favourite library objects, stored in the session state.

Dictionary:

library_dictionary and station_dictionary: in the form of {"Name": name_list, "Address": address_list, "URL": url_list}, in order to create a streamlit dataframe

6. **Functions for fetching and cleaning data:**

read_data() in read_data.py

7. **Pages and /or views:**

i. Graphs: I will visualize the coordinates of libraries and stations with folium map.

ii. Data analysis: I will calculate the distances between libraries and stations and save the distances into a list. Then, I will compare the distances with a decided boundary distance and display those that are less than the boundary value.

8. **Reference to external resources:**

i. Streamlit tutorials: https://docs.streamlit.io/knowledge-base/tutorials

ii. Folium guidance: https://python-visualization.github.io/folium/latest/getting_started.html