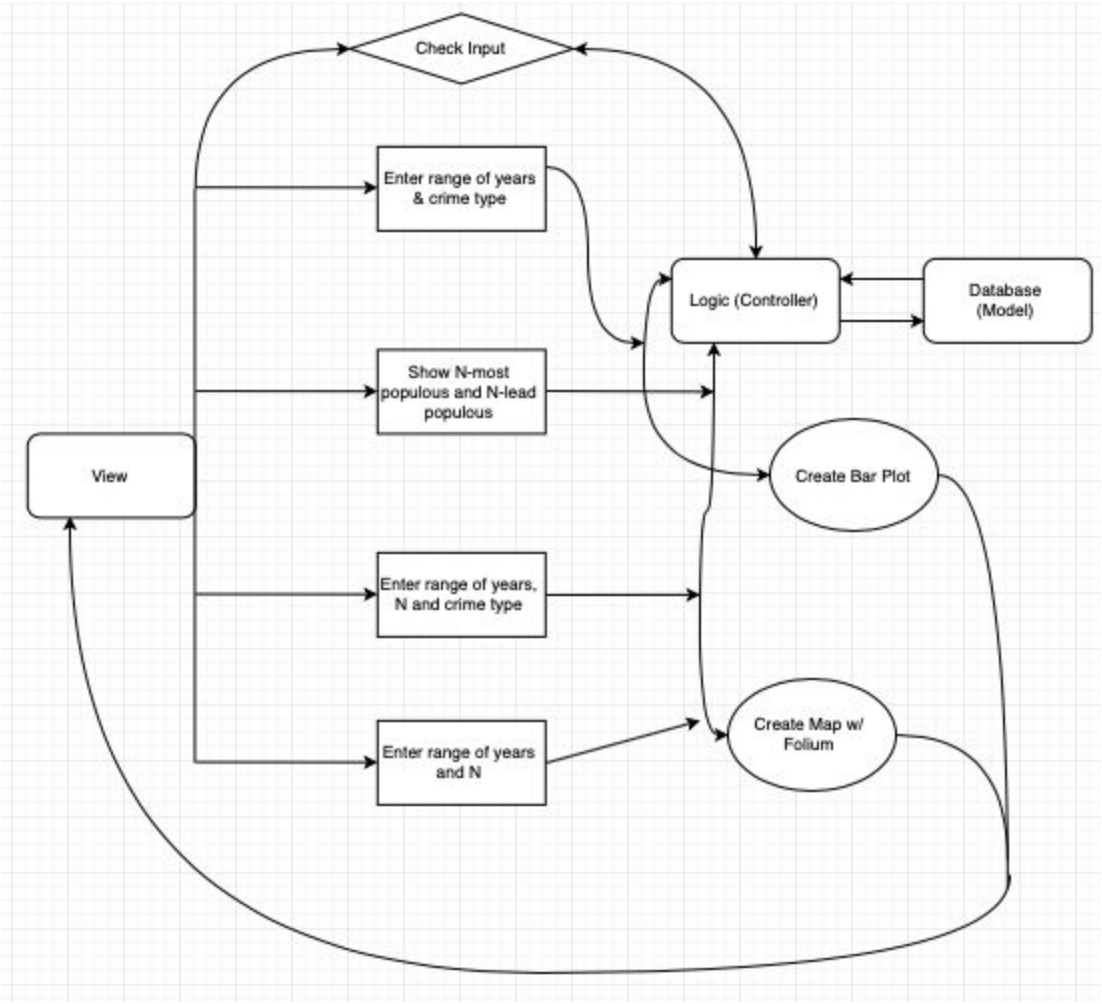
(a) Detailed Design of Software → Coding and Logical Flow

*****NOTE: This diagram was only used for planning purposes and holds no bearings with REAL data mapping*****



(b) User Guide

Startup

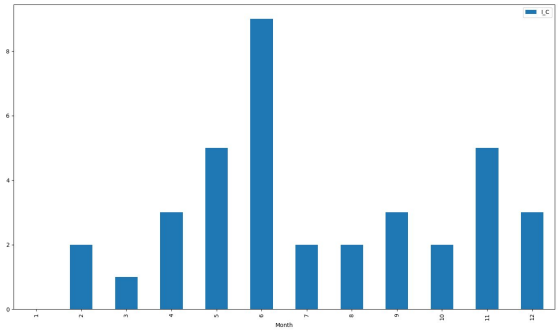
To run please use commands:
Pip3 install flask
Python3 handle_request.py
Please run web interface on incognito browser or else results will cache.

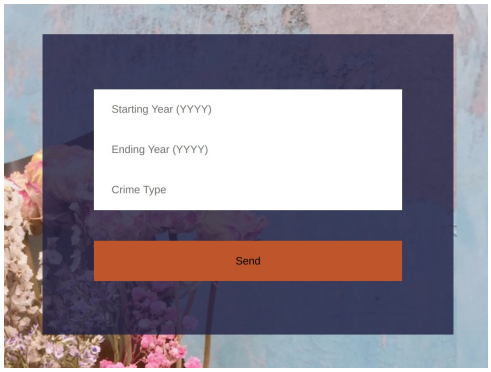
When the program is loaded, the user is prompted to enter a database. Then the user is prompted to enter an integer corresponding to the 4 tasks, or enter integer 5 to quit the program. After the execution of the function, the program will loop back to the main page, where the user can select another task to be performed.

Specifics

Running Q1:

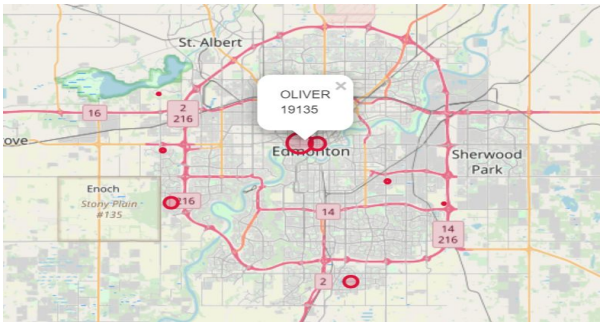
The user is prompted to enter a range of years and a crime type. The month wise total count of the number of the crime incidents that occurred within the given range is displayed on a bar plot. After the execution of the function, the program will loop back to the main page, where the user can select another task to be performed again.

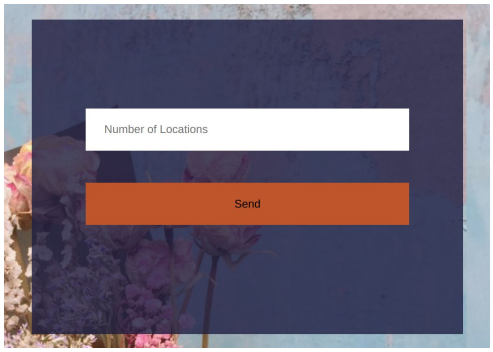




Running Q2:

The user is prompted to enter an integer N. The N most/least populous neighbourhoods along with their population counts is shown on a map.

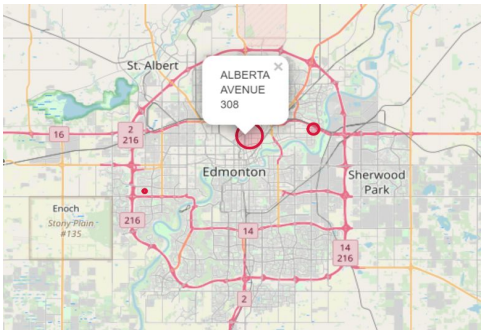


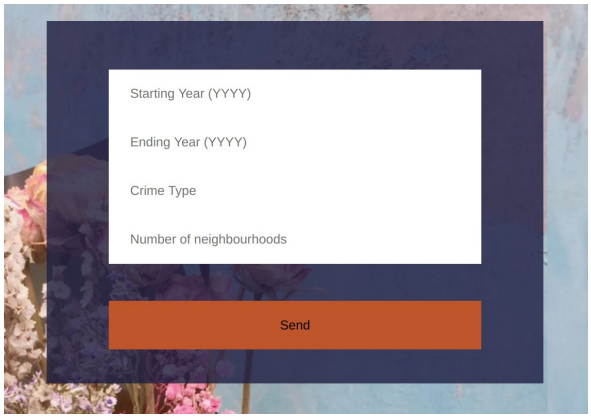


After the execution of the function, the program will loop back to the main page, where the user can select another task to be performed again.

Running Q3:

The user is prompted to enter a range and an integer N. The top N neighbourhood and their crime counts where the given crime occurred most within the given range is shown on a map.

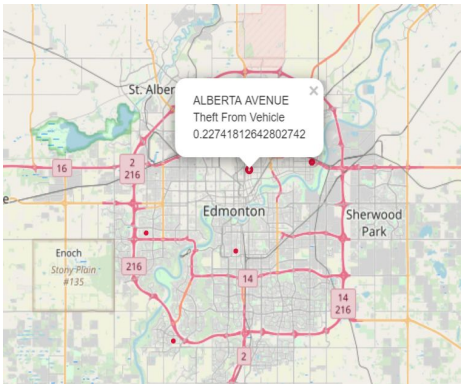


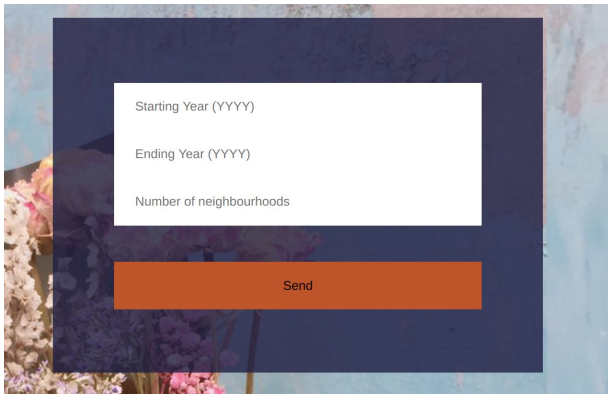


After the execution of the function, the program will loop back to the main page, where the user can select another task to be performed again.

Running Q4:

The user is prompted to enter a range and an integer N. The top-N neighborhoods with the highest crimes to population ratio within the provided range is shown on a map. The most frequent crime type in each of these neighborhoods is also shown.





After the execution of the function, the program will loop back to the main page, where the user can select another task to be performed again.

Beyond Scope of Assignment

- We decided to implement a graphical user interface, using python with the flask framework. It took a bit extra of time to learn html, CSS and flask.

(c) Testing Strategy

- Wrote unit tests for each question → Test minor components of each function and print information along the way to verify correctness (We printed out all the information in the beginning, manually calculated a few ratios to double-check).
- Each person was assigned to test/review another person's function.
- Made an expanded database with tie cases to see if things were properly executed.

(d) Group Work Strategy

- To assign questions, we used a random name picker to assign the first 3 tasks. The longest task was Q4, so we decided to all schedule a day to work on it together.
- Each group member was also in charge of reviewing another person's task and perform testing on it, to ensure everything worked.
- We also required all group members to write good comments, so that when we are debugging each others' functions it would be easier to understand what is going on and to validate the correctness of output.
- We required Q1-3 to be finished in the first week, and Q4 and testing to be completed the due date week. Meet-ups scheduled as needed.

Name	Work Items	Time
Asma	Q1, Tester: Joe	~3 Hours per Question including writing tests ~8 Hours to meet, do Q4, combine code, testing(including testing each other), debugging, write design document ~10 Work on web interface, learn flask framework
Joe	Q2, Tester: Deb	
Debangana	Q3, Tester: Asma	