CMPUT 291 Assignment 4 Design Report

By: Justin Tan, Thi Ny Tran

Overview

The flow our our application is shown below on *figure 1.* From the main menu the user can select any of those options listed. The program uses python with sqlite3 embedded within it to alter and view the database. All options are presented on the command line.

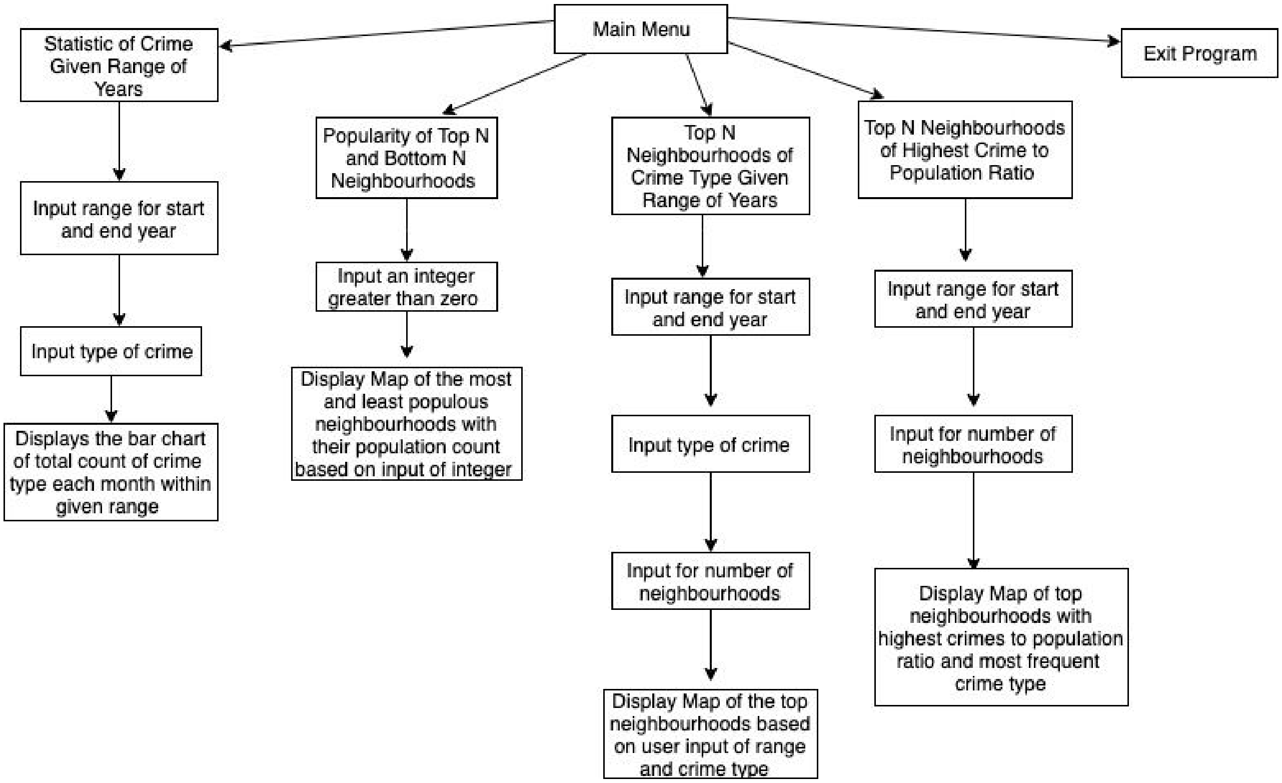
Main Menu
Statistic of Crime Given Range of Years
Input range for start and end year
Input type of crime
Displays the bar chart of total count of crime type each month within given range
Popularity of Top N and Bottom N Neighbourhoods
Input an integer greater than zero
Display Map of the most and least populous neighbourhoods with their population count based on input of integer
Top N Neighbourhoods of Crime Type Given Range of Years
Input range for start and end year
Input type of crime
Input for number of neighbourhoods
Display Map of the top neighbourhoods based on user input of range and crime type
Top N Neighbourhoods of Highest Crime to Population Ratio
Input range for start and end year
Input for number of neighbourhoods
Display Map of top neighbourhoods with highest crimes to population ratio and most frequent crime type
Exit Program

Figure 1

User Guide

Our assignment is a command line application where we've implemented it in Python along with sqlite3, folium and matplotlib.pyplot embedded within it. In order to run our application on the command line, the user should type "**python3 a4.py**"

When the application runs, the user will be prompted to enter a database file with the format "aDatabase.db" where "aDatabase" is the database file that the user wants the application to read from **(NOTE: the database file must be within the same current directory)**. Once the application connects to the database file, the user will be met with the main menu where they can select any of the displayed option by inputting any number from 1 - 4 or they can exit the program.

The foundation of this program is the main menu and then the user is prompted to choose an option from one of the 4 main functions of the program;

1. Statistic of Crime Given Range of Years
2. Popularity of Top N and Bottom N Neighbourhoods
3. Top N Neighbourhoods of Crime Type Given Range of Years
4. Top N Neighbourhoods of Highest Crime to Population Ratio

*<u>**NOTE**</u>**: at the end of each execution of a selected option, an html file will created and store into the current directory**

If the user selects the **first** option then the user will have to input the start year, end year and type of crime. Afterwards a bar chart is displayed from the created query with the months (from 1 to 12) labelled on the x-axis and the count representing the total number of incidents being on the y-axis. If there is an instance where no incidents occurred in a particular month for the crime type, the bar chart will still display the month having a count of zero. The screen will clear once an option is executed and the main menu will be presented again where the user is prompted to choose a new option.

Selecting the **second** option, the user will have to input an integer greater than zero. Upon entering the integer, a html file will be created through Folium. This file is a map representation of the top N most populated neighbourhoods and the Top N least populated neighborhoods. The circles on the map from the html file will display neighborhoods name and population count when clicked on. If there was an instance of a tie, then it will display the Top N +1 and/or bottom N + 1 neighbourhoods depending on where ties are present. The **blue** outlined circles represent the top N (+1) **most** populated neighbourhoods and the **green** outlined circles represent the top N (+1) **least** populated neighbourhoods. The screen will clear once an option is executed and the main menu will be presented again where the user is prompted to choose a new option.

The **third** option will need the user to input a start year, end year, type of crime and number of neighbourhoods. After the last input, a html file will be generated through Folium which represents a map displaying the top neighbourhoods based on user input of range of years and crime type. The **blue** outlined circles on the map will display the neighbourhood name as well as the crime count of the specified crime type within the range of years when clicked on. If there was an instance of a tie, then it will display the Top N +1 and/or bottom N + 1 neighbourhoods depending on where ties are present. The screen will clear once an option is executed and the main menu will be presented again where the user is prompted to choose a new option.

Finally the **fourth** option, the user will be prompted to input the start year, end year and the number of neighborhoods. A html file will be created through the use of Folium to display a map of the top N neighbourhoods with the highest crimes to population ratio as well as the most frequent crime type. The blue outlined circles on the map will display the neighbourhood name, type of crime as well as the ratio of crime to population count when clicked on. The screen will clear once an option is executed and the main menu will be presented again where the user is prompted to choose a new option.

With each execution of each option, it will produce a new html file with a count that is in correspondence with the number of times that specific option is executed. Another note, the border of the circles that are produced are going to be colored blue as its contrast to the map will help locate the data produced easier.

If the user enters “exit” while on the main menu, then the program will finish and exit out. After selecting any option on the main menu and finishing any of the selected functions, the screen will always clear after being prompted to input “Enter” for a clean UI experience.

Testing Strategy

Our general strategy for testing would be taking into consideration what happens if the user inputs something that is not desired when running the program and how would we deal with the input. For example, in Question 1 the user has to input a range of years and the boundaries are inclusive but what if the user inputs a year that is not in the data? Likewise, what happens if the user inputs a crime type that is not listed in the table? These are the type of scenarios we run by for each question to think of ways to manage them for our program to run efficiently. If we ran something that did not give us the correct output, we would traceback our steps and look at our code to see where we could have possibly went wrong and try to solve our problems one by one.

Group Work Breakdown

We split the 4 questions so that each member was assigned 2 questions worth 30 points.

- **Questions 1 & 4**: Thi Ny Tran (time spent: 9 hours)
- **Questions 2 & 3**: Justin Tan (time spent: 10 hours)
- User interface: Each member contributed to the creation of the main menu UI.

The progress made by each member varied when each member finished their assigned questions as some questions required more thought taking into consideration of ties. However, once we were done our questions, we came together and went through each question to fix minor bugs and ultimately, achieve consensus on the entire program. Communication between us happened frequently as we asked each other questions (online and in-person) about the assignment whenever we were unsure about certain implementations and criterias for the questions. This way, we were able to keep the project on track by constantly keeping ourselves updated with questions each of us had for one another during the process of completing the assignment.