MCIT-5940 Project Report

Tuan Yi Tan, Zhenyang Xu

(1) Additional Feature:

In this additional feature we implemented a method which calculates a numerical factor can be used to measure the covid-prevention potential of a certain area in Philadelphia. Its mathematical expression is as follows:

$$prodLaAndVacc(zip, date) = LivableArea(zip) * \frac{(partial\_vacc(zip, date) + full\_vacc(zip, date))}{population(zip)}$$

Here, LivableArea(zip) comes from the property dataset, partial_vacc(zip, date) and full_vacc(zip, date) comes from COVID dataset, and population(zip) comes from population dataset, so this calculation requires the presence of all three data sets.

The intent of calculating this factor takes two situations into consideration:

(1) The economic situation. Since covid is primarily air-contagious, a larger Livable Area per capita means the less possibility a person gets infected in that area, which will increase the potential of this area to prevent against covid's spreading.

(2) The medical situation. The rate of population who get vaccinated to total population in that area is another important factor affecting the possibility of infection people get in that specific area.

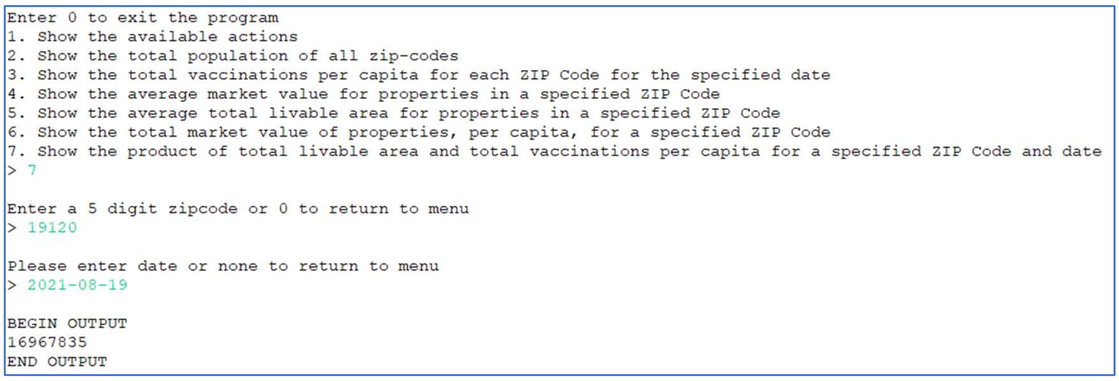
A best way of validating this feature is to passing several numerical tests:
For example, if we chose option 7 which executes additional feature, we continue to pick zip code as "19120" and select date as "2021-08-19", this will give us calculated result as 16967835. If we look at the component value of above mathematical expression:
We have:

$$LivableArea("19120") = 3.5978 * 10^7;$$
$$\text{Partial\_vacc}("19120", "2021\text{-}08\text{-}19") = 6370;$$
$$\text{Full\_vacc}("19120", "2021\text{-}08\text{-}19") = 24840;$$
$$\text{Population}("19120") = 66177;$$

Which yield:

$$\text{prodLaAndVacc} = 3.5978 * 10^7 * \frac{(6370 + 24840)}{66177} = 16967835;$$

which matches the result output by the program (as shown on next page).



```
Enter 0 to exit the program
1. Show the available actions
2. Show the total population of all zip-codes
3. Show the total vaccinations per capita for each ZIP Code for the specified date
4. Show the average market value for properties in a specified ZIP Code
5. Show the average total livable area for properties in a specified ZIP Code
6. Show the total market value of properties, per capita, for a specified ZIP Code
7. Show the product of total livable area and total vaccinations per capita for a specified ZIP Code and date
> 7

Enter a 5 digit zipcode or 0 to return to menu
> 19120

Please enter date or none to return to menu
> 2021-08-19

BEGIN OUTPUT
16967835
END OUTPUT
```

(2) Use of Data Structures:

In this project, we have used three different data structures to store the information extracted from different file. They are as follows:

(2.1) Map<String, Map<Long, Vaccination>> for covid data

Covid data extracted from covid_data.csv or covid_data.json contains two important data fields: partially_vaccinated and fully_vaccinated, which represent the number of people who have completed their first or second dose of vaccination. These two fields are classified by joint index combining zip code and time stamp. For this reason, we have picked a nested map<key, value> as our first data structure. In this project, the method of searching and getting the value associated with a given key will be frequently performed, so a map implemented by HashMap will be a good choice. As the get(), put(), and containsKey() methods require O(1) time complexity in HashMaps.

The String in this data structure represent the specified date formatted as “yyyy-mm-dd”. The Long field represents the specified zip code since the zip code can be at least five digits long with extensions. The Vaccination is a custom object belongs to Vaccination class in utility tier. This class has a field named vaccStatus which is an array, its first element vaccStatus[0] is used to represent the partially vaccinated number and second element vaccStatus[1] used for the fully vaccinated number.

(2.2) Map<Long, Long> for population data

The data extracted from polulation.csv is a simple mapping from zipcode to population number. A HashMap<key, value> should satisfy the storage of this kind of data as we require no remove duplication action (which is common in Set) or sorting action (which can be implemented with a TreeMap). Here the first Long field is zip code, second Long field is population number associated with this specific zip code (or area). As we discussed before, these numbers can be very long as to exceeds the bit size of an integer, a safe way of avoiding overflow will be initializing them as Long.

(2.3) Map<Long, Property> for property data

We used a HashMap to store the data extracted from property.csv. Each property record is associated with a zip code as key, so the first key field is a Long format. The value field is a

Property object which should be instance of Property class defined in utility tier. The Property class has fields storing the total number of market value and livable area as well as their counts, which can be used to calculate their averages.

(3) Lessons Learned

In this project, we have tried to implement the work process taught in class to implement this project. The needs are analyzed first. A N-tier architecture is used to represent different components of program. We have tried to implement the logger tier with singleton design pattern and implement average total livable area with strategy design pattern. Files are read in with different data structures to store information extracted from them. A processor tier is implemented to perform backend calculations along with a user-interface tier to perform frontend user interaction such as take in user's option and output formatted results. We have learned a lot through this project as it combines nearly all materials taught in class and deepens our understating of class materials through practical programming. Going forward, we will be more careful about design pattern implementation aspect of a future project.

In this project we used slack as our primary method of communication and we used GitHub as our code version control method.