Project 1 Design Document
C.S. John Lee
10/27/2019 (Finalized after major completion of program)

Introduction:

The cost of living in certain cities has increased dramatically with gentrification pushing citizens out who have lived in their homes for most or all of their lives. I have classmates who can't afford to rent or buy new homes in the Bay Area. However, there is the option to move to other cities in different states. In the past four years, I've moved to three different states. Each time, I have the fear that the next move will be worse than where I currently live. While certain aspects are worse, other aspects of the new location definitely increase my level of happiness. For my project 1, I am creating a program to help a user determine either (1) the happiness of the user at a specific location or (2) the location that will give the user the most happiness.

Disclaimer Since this project is used as a learning tool for my understanding of object-oriented programming, this program will only apply simple google research information on locations and include theoretical weights to each attribute of a location to determine happiness.

Running the program (user interactions):

This program will ask the user for their preference of an attribute of a city, for instance, do you like a cooler or a warmer climate? Then, it will ask the user what action they would like to do: (1) Move or (2) Find forever home.

The Move action will prompt for the location to move to. Then it will calculate the happiness score based on this location, and the details regarding this location (showing all of its attributes). Then it will return to the main menu. The Move action also has another option to see a list of places the user has "moved to".

The Find Forever Home action will calculate the user's happiness score for each location. Then it will return the list of locations ranked by happiness.

Error will be raised when inputs are incorrect or something invalid happens.

Specifics/Purpose for each class:

Class Person will store the characteristics of the user. It will include weights for each attribute based on user input. The inputs/attributes with weights are: daylight fluctuations throughout a year, climate, median salary, home value, population density, and property crime rate. For example, the prompt would ask the user to choose hot or cold climate. If cold, the locations with more heating required than cooling would be ranked higher by having larger weights. These weights are used in the Happiness class.

Methods will generally be the __init__ and the __str__ or __repr__ methods. __str__ will be the output for User class.

Class Location will define the characteristics of many pre-determined locations. It will include, as inputs, each of those attributes listed before (daylight, climate, salary, home value, population density, crime) per city. Methods will be __init__, __str__, __repr__, and “get_details”, all of which allows usage of the Location class with the Happiness class or the Move class. The get_details method will return the characteristic of the city in the command prompt.

Class Happiness will calculate the happiness of each location based on inputs from the person (weight of attributes) and inputs from the locations (attributes of each location). Firstly, happiness will store a list of all the locations as well as the statistics (sum, average, max, min, number of cities) using its methods “addtolist” and “statslocation”. Additionally, the method “hindex” will actually calculate the happiness score for a specific location, the equation essentially adds all six attributes with a 1/6 weight (equal weighted per attribute). The user’s input will switch the preferences of (x3) attributes so the happiness index of different users will likely be different. This class, overall, calculates the happiness index, and stores it afterwards so that it can be returned in the command prompt.

Class Action will be a child class to Happiness. This class actually is the actual “action” from the user which will actually call the methods in the Happiness class. The method “move” will accept a location specified by the user and call the hindex method from the Happiness class to get the happiness index. The method “traveled” documents all the location the user has selected to move to and returns this information. The method “Fhome” will print a list of all the locations available ranked by highest happiness index.

Class commToTheEndUser is a class just to interact with the user. By including all of the print statements and verification methods (to verify that the user inputted correct information, if not, an error is raised) in this one class, it keeps the prompt more clean and allows multiple forms of communication. For example, the command prompt can choose to just accept the inputs and the commToTheEndUser can be ignored.

Complexity:

There are 5 large classes: one child action class, one parent calculation class, 2 attribute classes, and one command prompt class. Each of the classes helps separate the different parts of this project clearly and allows interaction with each other. The interaction of each class will be a challenge as the happiness class and action class pulls information from other classes. This project takes advantage of class structures and demonstrates understanding of object-oriented programming.