Samson Olivero

113479362

Project 3

The Task:

The task is to combine the last few project ideas and to implement them into this project. Except this time, we had to incorporate GUIS and read into a different .txt file(which is an idea we took from Project 1), and then implement a method from Project 1 to find the Hamming Distance values of the StID as the value of the JComponents: JSliders, JTextFields, etc.

Steps:

I decided to take this from the Constructor level of class and build from there. I decided to build the Hamming Distance classe first to make it easier on me in the future to build the GUI JFrame and Components. If I had these methods and calculations already done, it would make the rest of it easier. I took note that there were similarities between this project and the last couple and actually used the same code I used in the last projects and placed them in this one. This made it extremely easier on me, and I just had to tweak the code to fit what this specific project wanted me to do. I had to use my limited knowledge on GUIS and play with the gridlayout and try to make everything fit the way I wanted to. That was probably the trickiest part, getting the slider to work was a close second(it turned out to be three lines of code…) It was difficult because I kept having to try and catch exceptions and I will admit, it was getting frustrating. We did not have to read into a file the last couple labs so this was a new ball game. I figured it out, but it was even harder to keep in bounds, and I had to constantly update my global variables if a button was being pushed.

Samson Olivero

113479362

Project 3

HammingDistance:

This class was pretty straight-forward, you find all the HammingDistances, put them in their own separate arraylists, and make sure to keep track of the number of stations have what distance in relationship to the selected station. I have a lot of getters and one method that calculates the counts and the arrays. I realized that if I wanted this code to work, the .getHammingDistance method needed to pass a String as a parameter for the “selected” station.

HammingDistanceFrame: Project Side

This was extremely difficult to do: I had to organize all the components and sort them to look decently. I ended up using 7-8 JPanels and then separating those JPanels according to the functions they need to achieve. Then I had to incorporate the methods from HammingDistance, and figure out the best way to use the methods I created from said class and use them on JSliders, JButtons, JTextFields, etc. This Class essentially does exactly what it needed to do: create a GUI, calculate the HammingDistances, pick the HammingDistance the user wants to us, Show the Stations that follow those guidlines, and add to the list of Stations temporarily.

HammingDistanceFrame : Fun Side

TicTacToe is a super fun game, and it should be easy to program… so I thought. It took me a while to figure out how to use tictactoe in my code, and luckily I was able to do some research online, combine some code fragments from various others, and then use my knowledge of my own code to make sure it worked with the rest of the program. That was probably the trickiest part : to get the codes to complement each other.

Samson Olivero

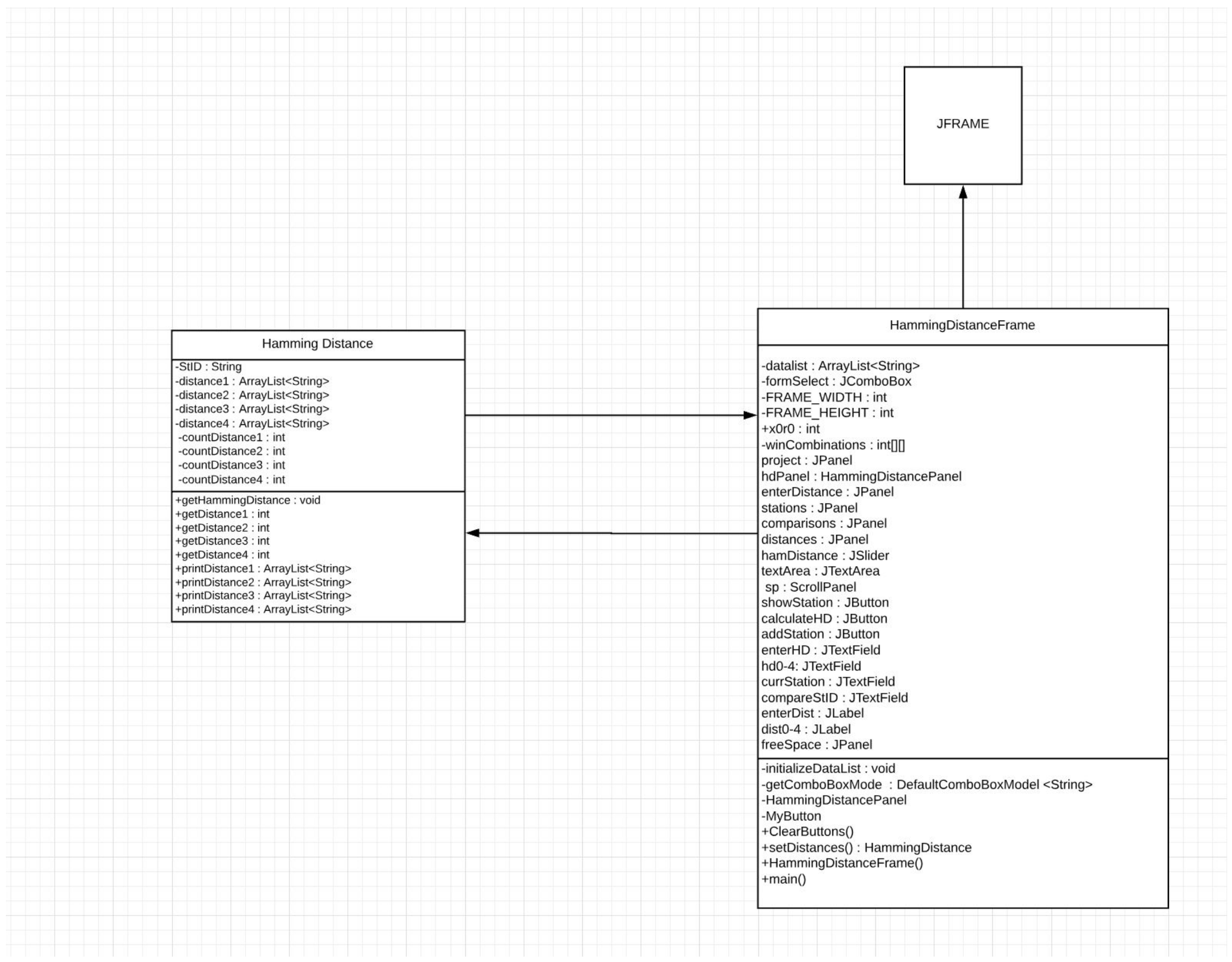
113479362

Project 3

Issues:

The Biggest Issues was making sure that I did not go out of bounds while sorting my ArrayList of strings, and then making sure I keep the Box updated while the GUI was running. It was also super difficult to deal with the GridLayout, I had a lot of trouble getting it to look decent. Other than that, the rest of the project was really fun to do. I just had to look at past code, and build from there.

UML:



Samson Olivero

113479362

Project 3