Azariah Laulusa
Professor Biplav Srivastava
CSCE 240
02/21/2023

Project Assignment #2: Data Processor

The goal of this project assignment was to take in the extracted data from the Project Assignment #1 and use that data to answer a user's questions. In this case, my program will prompt a user for what disease they are interested in getting information on, and then, my program will let the user ask questions and answer them, if possible. To answer the user's questions, the program takes in the outputs, *rabies-cdc-output.txt* and *rabies-webmd-output.txt*, from the first project and process the data by deciding which parts of the data answers the user's question the best. The program will interact with the user and print out the answers in the terminal. For the test output, the program will output a file called *test_output.txt* , showcasing the question the user asked and the answer the program processed.

In the main method, the program will introduce itself and prompt the user for a disease. I used a while loop so that the user can continuously try inputting a disease. If the user inputs a disease that the program has no data on, the user is informed and is asked for a disease again. Since the program only has data on Rabies, the program will only accept rabies as an answer. I incorporated *regex* so that if the user typed something close to rabies, the program will ask the user if they meant to say rabies. If the user inputs rabies, the program calls the *readFilesAndAnswerQuestions()* function. This function doesn't return anything, but it takes in the user's answer as an input.

In the *readFilesAndAnswerQuestions()* function, the program prompts the user to ask a question. The program takes in the question as input and searched for the question using if, else if, and else statements. If the question is found, then the program will either open the *readCDCFile()* function or the *readWebMDFile()* function based on which gives an answer to the user's question. These functions are used to find the data need to answer the user's question in the files *rabies-cdc-extracted*.txt and *rabies-webmd*-extracted found in the data file. On the other hand, if the question is not found, the program with let the user know it can't answer their question, and it will prompt the user to ask another one. Using a while loop, the user can

continue to ask questions. If two files can answer the same question, the user will be asked to decide which data file they want the answer from.

Overall, the project was fun to code. I do wish I put more time into making the code even more user-friendly and unique, but I am happy what I came up with in the time I had. From now on, I do want to focus on making my code reusable and including exception handling. I also want to add more detail to my code. For example, if the user keeps asking the same questions or if the user continues to input random things, I want my program to be able to address this to the user and find a solution to solve this problem. These ideas are definitely what I'll carry with me while moving on to the next projects. I am excited to learn more and practice more so that I can use these skills in the future.

Works Cited

Editorial Contributors, WebMD. “Rabies: 9 Symptoms & What Do If You Are Bitten by a Rabid Animal.” *WebMD*, WebMD, 10 June 2022, https://www.webmd.com/a-to-z-guides/what-is-rabies.

“Rabies.” *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 5 May 2022, https://wwwnc.cdc.gov/travel/diseases/rabies.