Project Report

Sara Padula

Requirement – What did the instructor ask you to do?

For this project, my instructor asked me to create a chatbot, wherein the user can ask questions in the console and the program should be able to respond to any query and provide the appropriate answer. Specifically, he wanted this chatbot to provide information about a South Carolina representative. To do this, he broke up the assignment into 5 parts. Firstly, the program needs to be able to extract data from the district website. Then, the program should process the data based on the possible questions that could be asked about the representative. Then, based on specific user queries the representative information should be available to the user. Then, given any query by the user the program should be able to produce an answer – even if there are misspellings or different wordings of the question. Finally, he also expected the program to report statistics about the interaction of a chat session, across all chat sessions that have been done in the program.

Specification – What did you do, what scope you selected and what decisions you made?

For a project of this size, each step could go many different directions and there are many decisions that need to be made. My first decision was choosing C++ as my programming language for this project. I have previously learned Java in other classes and considered this language at first, but I wanted to expand my knowledge and abilities. Thus, I decided to challenge myself by using C++ as my language to obtain a greater understanding of a new programming language. In addition, I came to appreciate this decision more as the project went on, because C++ is a more succinct language to use than Java.

For the first part of this project – extraction – I decided to save the HTML code from the representative’s website into a txt file. I did this, because I had a lot of trouble trying to create a program that would parse through the actual website. Then, I created a void parser function that passes in char\* S. To make this function work, I created two for loops and a while loop that would populate array S with the representative’s information from the website. Using my knowledge of the HTML language, I made the program so the array would begin being populated when it came to a ‘>’ and the array would halt being populated when it came to a ‘<’. This allowed me to only extract the data that appears on the website rather than the extraneous code that formats the website. Then, within this function I output the information from the char\* S array to a txt file named “output.txt.”

For the second part of the project – processor – I first read in my “output.txt” file from the first part of the project. Then, I organized the data into various categories by hand because I found it difficult to use regex to find specific parts of the document. Thus, I provided the user 3 categorical options on the console: contact, personal, and committee. Within this decision tree I used a string array to extract the exact location of the information that fell within each category, then output this information to the user.

For the third part of the project – ui – I read in my “output.txt” file from the first part of the project and reused the code I used to extract the information from that txt file. Then, I created an infinite loop for the chatbot itself, which would allow the user to continually ask the console questions unless they chose to exit the program. To handle user queries, I first created a toLower() function that would convert all user input to lowercase, so I did not have to worry about handling words of varying case. Then, I created a series of else if statements where I wrote out as many possible ways a person can ask questions about the information I have available about the representative. However, if the user asks a question that is not supported they will receive a message that they either need to reformat the question or that we don’t have that information available to them.

For the fourth part of the project – userintent2querymapper – I used all previous code from the earlier projects and added a function “double compareString(string str1, string str2).” I did this because the aim of this part of the project was to handle any user query rather than expecting the user to ask the exact questions that I had written in my else if statements. This function essentially compares the string the user inputted to the string within my else if statements. Then, it goes through each string letter by letter and calculates the percentage of character matches between the two strings. Then, I called this function in each else if statement for each possible query I had written, and if the percentage of matches is greater than 70 percent, then the match would be successful, and it would output the information from the else if statement.

For the final part of the project – session logger – I was tasked with calculating and outputting statistics about all chat sessions that occurred in the program. To do this, I used all the same code from my previous projects an added many things necessary to implement this program. Firstly, I added a clock that times the chat session, an int variable that is iterated every time the user chats, and an int variable that is iterated every time the computer chats. I also wrote code that would ask the user to enter the date and time before each chat session and use that information to create a new txt file where the chat log of the chat session would be saved. Additionally, I created a count txt file that would add 1 to the file each time a chat session is begun, so I can then add all the ones within the count file to determine the chat ID of that specific chat session. Then, I created a function create() that takes in user iterations, computer iterations, time (from the clock), the txt filename, and the count file. Then, this function adds a line to an already existing csv file to save the data to later be extracted. Then I created a data structure that would save every chat sessions statistics into an easily accessible format. Then I used these data structures to extract, then calculate the necessary information for the statistics that the user asks about.

Development highlights – How was your code implemented? How did you test? What problems did you face and how did you solve them?

I implemented this code using C++ language. I wrote the program in Visual Studio and tested and ran the program in Visual Studio as well. When testing my program, I ensured to try user queries I hard-coded into the program, user queries that were spelt wrong, user queries that were worded differently, and user queries that were not possible to ensure my program could handle all of them as necessary.

I faced many problems when trying to implement this code. My first problem was trying to extract the information directly from the district website. To solve this, I copied and pasted the HTML code from the website into a txt file and extracted the information from there. Another problem I had was trying to find specific words from my txt file that contained my extracted information. I solved this by going through the txt file by hand and saving each individual piece of information in its own string. Another problem I faced was getting the system date and time using commands such as ctime, because visual studio gives a warning when these commands were used. Thus, I had to ask the user directly for the date and time, to create unique text files to save the chatlogs in for the session logger part of the project. I had many other issues trying to make the program handle all user queries without hard coding all possible inputs within else if statements, and if I had more time to perform trial and error with the program I could likely make the code more efficient and usable.

Reuse – What did you do to make your code reusable? What code did you use and why? Who is using your code and why? What challenges did you face?

Most of my code is likely not entirely reusable since I read in the information from the output file line by line, but my useful functions for certain aspects of the project are very reusable. Specifically, my functions for comparingstrings, creating a data structure of the information for the session logger, and my function to add lines to an existing csv file can be implemented in anyone’s code to easily address those aspects of the project. I used all of my own code, but I got ideas from forums online for specific aspects of the project. For example, I referenced a forum to create my toLower() function that makes any string all lowercase so it is easier to work with. I made my repository public but as far as I know no one has used my code and I have not used anyone else’s in the class. I faced challenges making my entire programs useable for other people in the class.

Future Work – What more can be done to make your chatbot useful? How will the code need to be changed over time?

My chatbot would be more useful if it more efficiently extracted information directly from the website, if the data were better organized into categories, if my program could handle more user queries, if my program could get the date and time at the time of a chat session without having to directly ask the user, and if my program was entirely just a command line interface rather than having to go through a menu of options. To address these issues with the code I will need to organize my code more efficiently into different categories. I will need my code to adapt to the website as it is updated and still be able to extract the necessary information. I will have to write my code in a way that can handle any user query without hardcoding all possible user query in else if statements. And I will need to utilize regex to make my program entirely a command line interface, rather than using a menu of options for the user to go through.