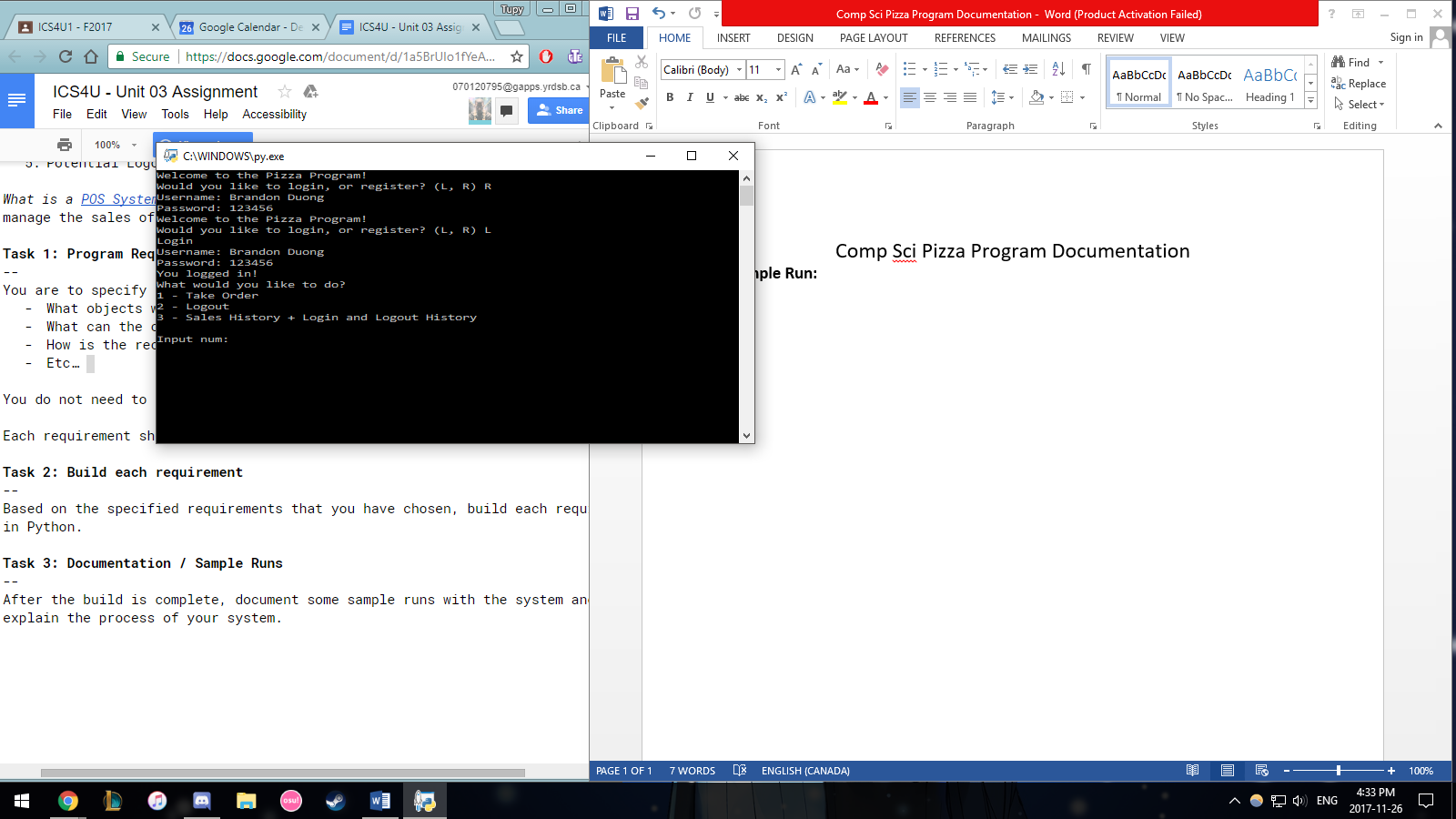
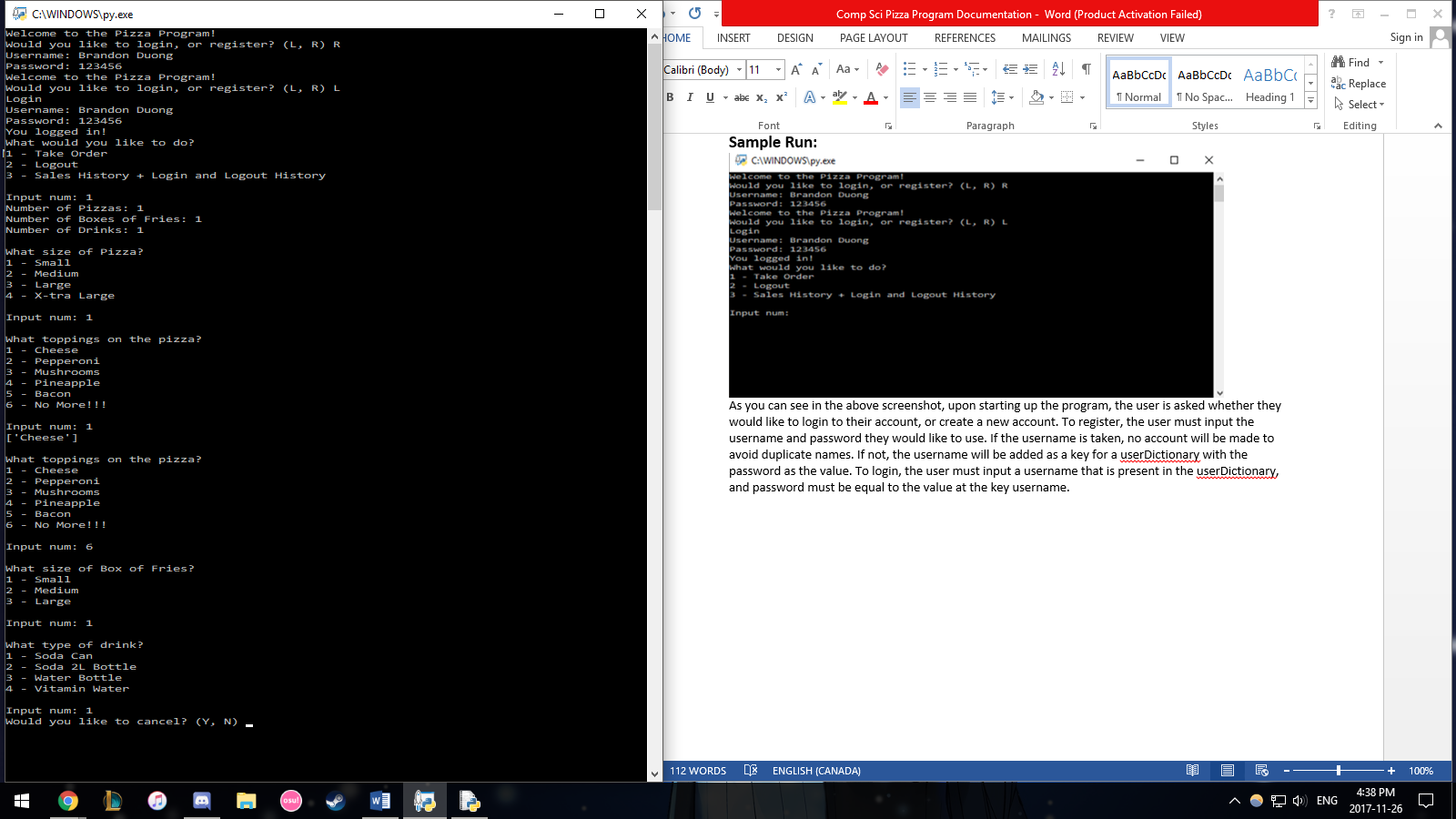
Comp Sci Pizza Program Documentation

**Sample Run:**



As you can see in the above screenshot, upon starting up the program, the user is asked whether they would like to login to their account, or create a new account. To register, the user must input the username and password they would like to use. If the username is taken, no account will be made to avoid duplicate names. If not, the username will be added as a key for a userDict with the password as the value. To login, the user must input a username that is present in the userDict, and password must be equal to the value at the key username. This is when the log in time is recorded, appended to the log.txt file, and object currentUser is made with the username as an attribute and User as the class.



Upon logging in, the user is given 3 options: take order, logout,

Or see the sales history and login and out history. To the side,

You can see the ordering process in which they must input the

Number of pizzas, boxes of fries, and drinks. These act as

Counters in the code to determine how many times to loop the

Ordering process of those 3 items. There are 3 dictionaries

Which are the sizeDict, toppingDict, and drinksDict which all

Have a number as a key, and the corresponding option as the

Value. Entering 1 when asked for the size of pizza tells the

Program to look at sizeDict at key “1” and returns the

Corresponding option, “Small”. The pizza objects attribute,

self.size is then equal to the selected option, in this case “Small”.

Same process for everything other than the toppings of the

Pizza. The program continuously asks for wanted toppings in

Which they are added into a list (with no dupes allowed) until

“6” is input. Then the program assigns the toppings list to the

Attribute self.toppings. All the object attributes are appended

to a list called fullOrder. Cost is calculated using the same

Method. There are many cost dictionaries such as sizeCostDict,

topCostDict, and drinkCostDict in which the keys are options

such as “Small”, and the values are the cost in type int. Once

all options are input, the program scans the objects attributes

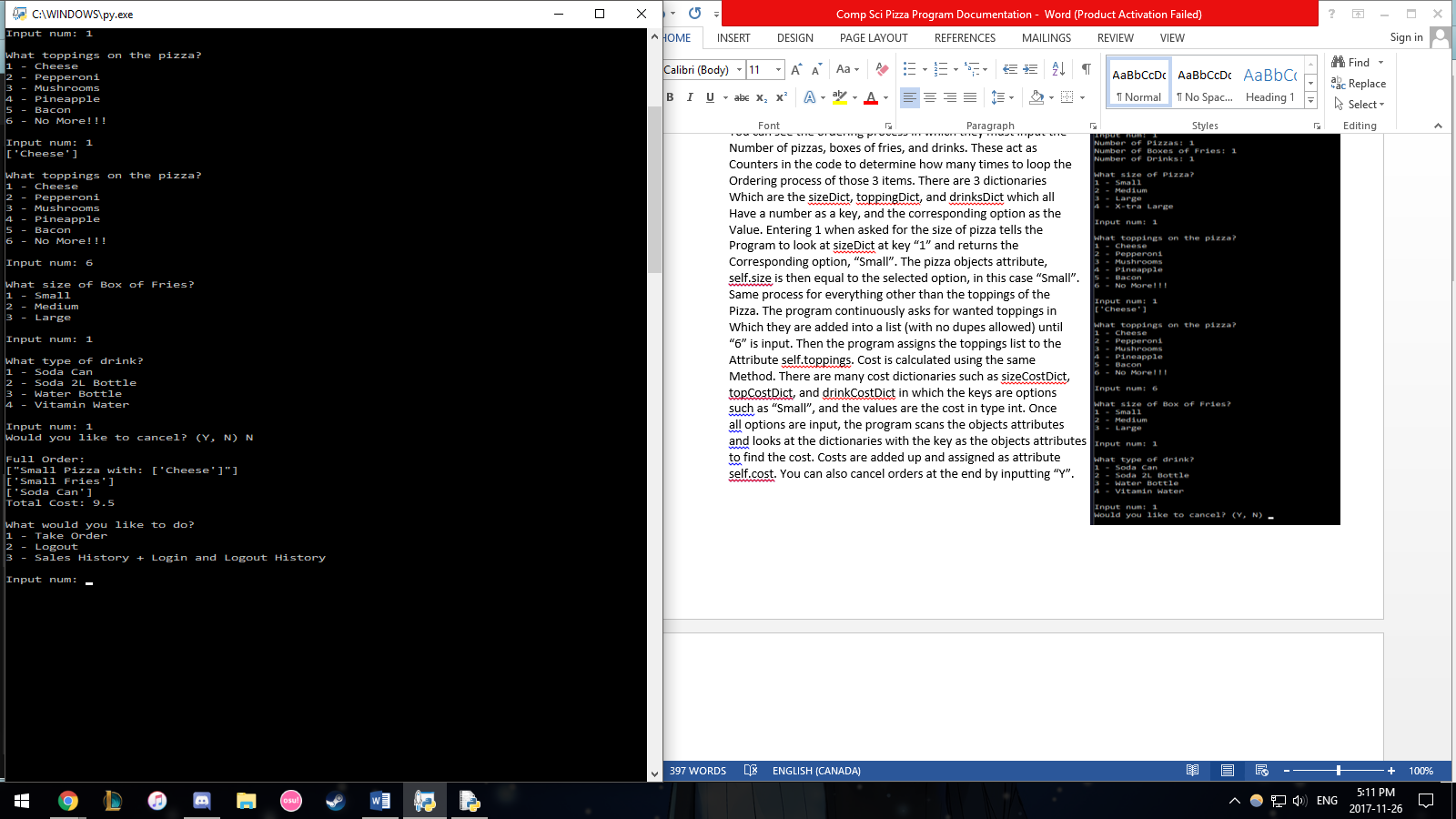
and looks at the dictionaries with the key as the objects attributes

to find the cost. Costs are added up and assigned as attribute

self.cost. After all options chosen, the self.cost attribute of the

current item is added to the total cost of the order, another variable.

You can also cancel orders at the end by inputting “Y”.

Inputting “N” would allow the order to go

Through. A receipt is then made as the

fullOrder is written into a .txt file named

receipt. The receipt is then printed out on

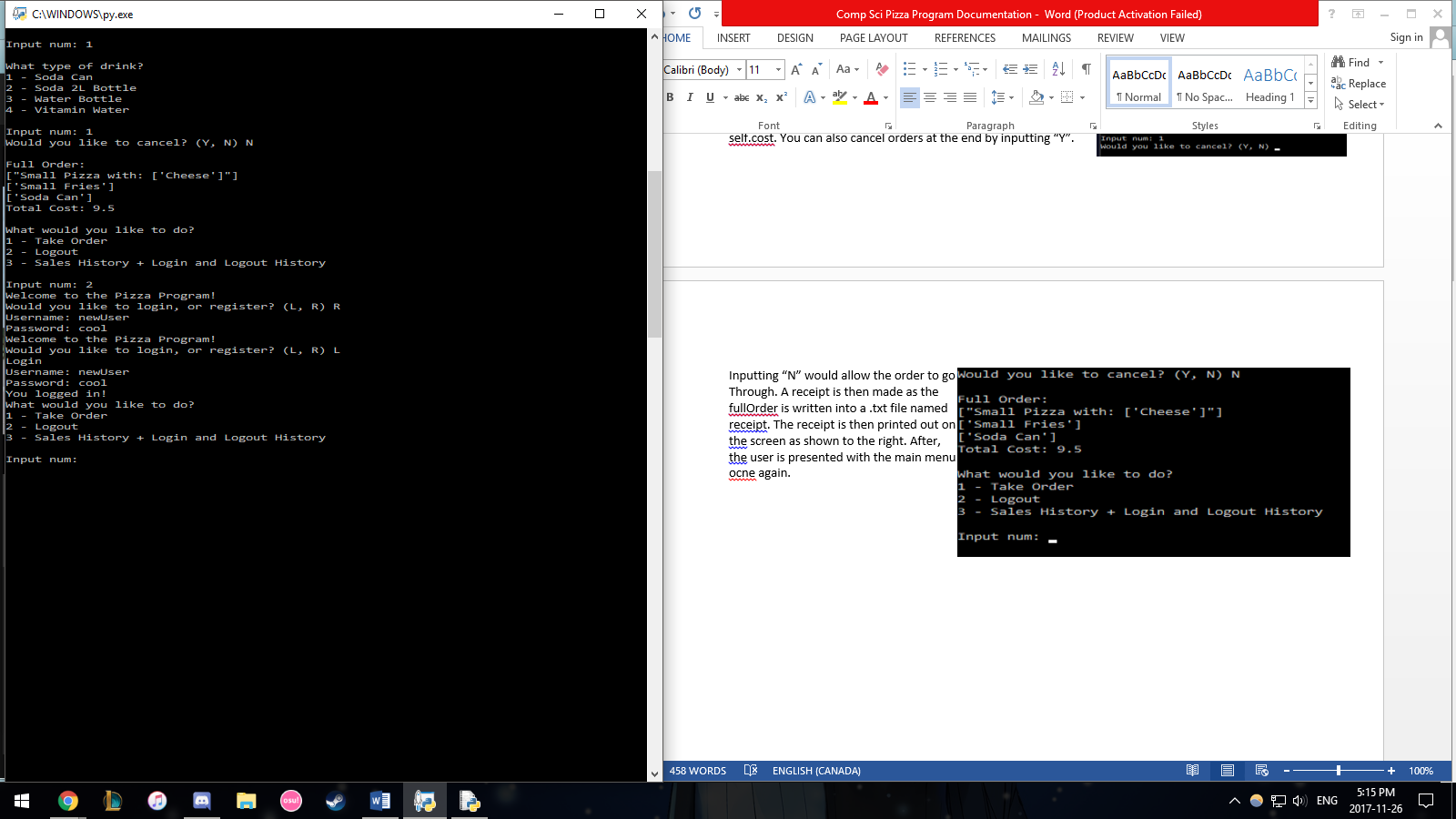
the screen as shown to the right. This

receipt is also appended into the

salesHistory.txt file to track all sales.

After, the user is presented with the main

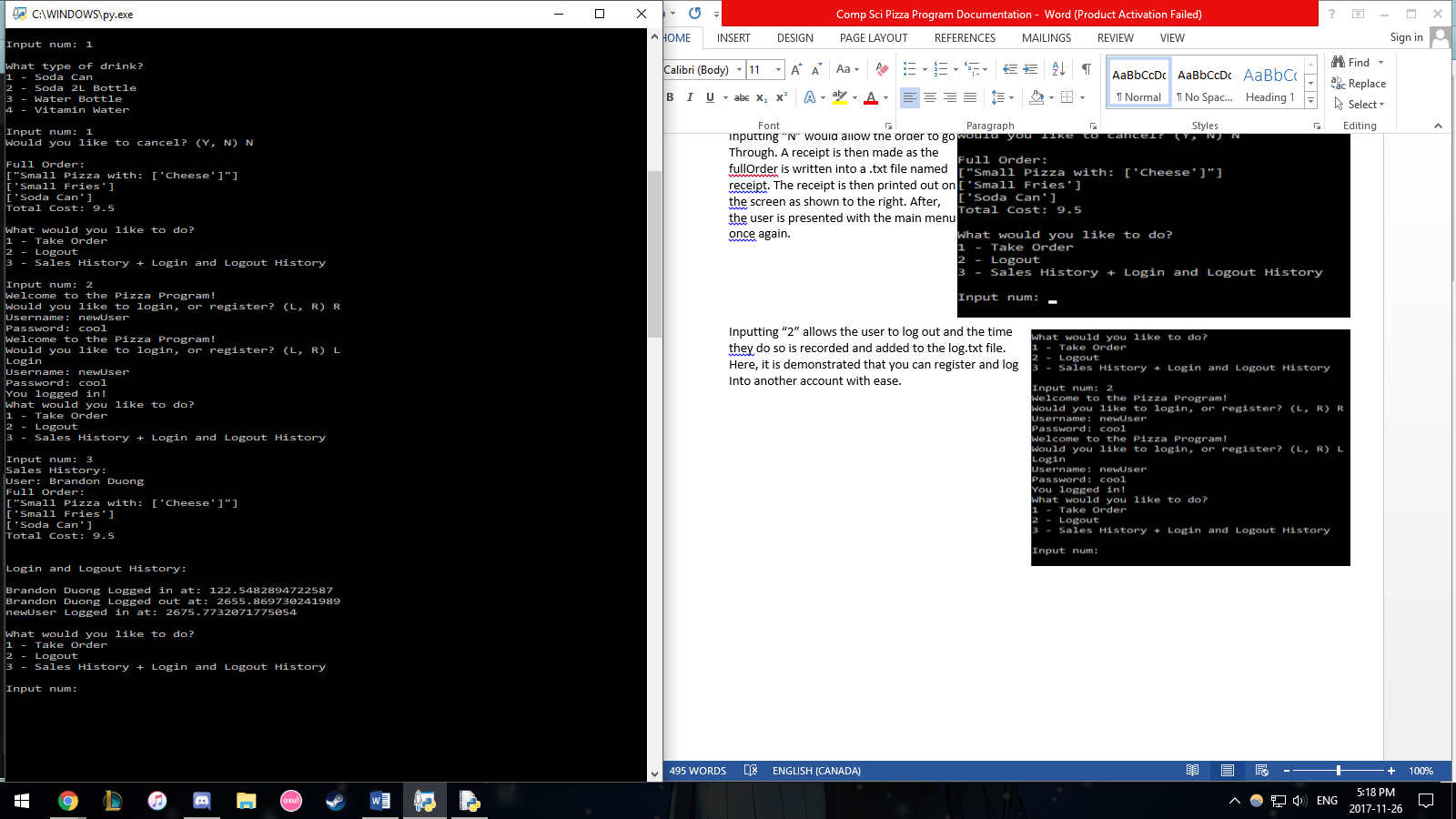
menu once again.

Inputting “2” allows the user to log out and the time

they do so is recorded and added to the log.txt file.

Here, it is demonstrated that you can register and log

Into another account with ease.



Inputting “3” allows the user to view the sales history

and log in and out history. Essentially, the program

reads and prints both the salesHistory.txt and

log.txt.

Here we can see the sale

that was made with the Brandon Duong account,

and the times that we logged in and out of both the

Brandon Duong and newUser account. Then the

User is presented with the main menu yet again.

That about sums up this sick pizza program.