**Mohammed Mahin Ibnay Mamun (346584)**

**Unit 4 2019 pizza Evaluation**

My code overall is amazingly effective as it meets all the requirements asked for by the local pizza restaurant. As a junior software developer, my job at the restaurant was to create a program which would result in calculating the pizza orders. The first thing I had to know about this project was what I needed to do and ask for. I needed to create a program which allowed staff at the restaurant to enter this data:

* customer details:
* name
* address
* phone number
* the quantity of pizzas being ordered
* the size of each pizza. An order can include different pizza sizes
* an option to add more toppings to the standard pizzas if required
* an option to request delivery.

By completing this project, I used a range of different coding techniques such as lists loops operators and much more. The code was quite repetitive which made it easier, however there were areas which needed altering. For example, the customer details. Asking for a name, address and number only requires a few lines of code which can be copied, pasted, and altered. However, when it came to testing (normal / extreme and boundary) extra code needed to be added. A code which was first 3 lines became 6 lines. This took some extra time but was not too hard to work out due to the experience and resources I had to guide me from my classroom. In this area, the tricky parts were adding the data validations. Once I got the data validations working (e.g. – name should not have any number. Number should not have letters) the rest was just simply putting these into while loops so it will infinitely run until the condition is met.

Another area in this code where I struggled was with the if statement for the quantity of pizza. I did not really struggle it was more of the code not running as expected. I had 2 code ideas to try. However, when one was working the other was not and it was messing up. This could have been a software issue. This code was == if pizza\_quantity < 1 or pizza\_quantity >6: and the other was == if pizza\_quantity < 1 and pizza\_quantity >6: both these codes worked but also did not at the same time. However, in the end I managed to get it working.

Another issue that came up during the code was with the toppings. I used a code which had an if statement and 3 Elif statements. However, the last Elif was not running. I thought it was because I could only use 2 however after getting lots of debugs, I figured out it was an issue with my indents. This simply made my code look quite messy, but I managed to fix the indents.

When it came to adding an option for the delivery, this part was quite easy. All I needed was user input. If they said yes, then it would charge 2.50 to the total. If no is entered, then it will print out please come and collect your food it will be ready soon. Now I had the base set and running, I tested to see if it worked, and it did. Then I considered adding if delivery == “yes” or “y”: I also added if delivery == “no” or “n”: the next part was simple due to repetition. I asked myself what if the user types something other than “yes” / “y” and “no”/” n.” So, I added a while loop so my code will keep repeating until the user types yes/no or y/s.

By this point my code was still working as expected. There were some areas I had to go back to make amendments. Creating the discount part was by far probably the easiest area on this project. I managed to add a discount to my code if the user spends more than £20 using 6 lines of code. This part of the code does not require any input from the user which made it slightly easier. This meant I would have to worry about any input validations. My code begins with if total > 20: print (“you get 10% discount”) inside this if statement I added a line of code which calculated this discount total = total – discount for my else statement it would very simply just print “you didn’t gain any discount because u didn’t spend more than £20” it displays the total.

The decisive point I needed to add to this code which I remembered by referring to the assignment brief was to add a customer receipt. This part was quite challenging because I have not created any before. However, after a while of trial and error I managed to think of an idea and tested it out. My idea was to run an array then print it off. I created a 2d model array which let I then used in me for loop so I could print out both the price of cost and reason of cost. For example ["name", customer\_name], ["address", customer\_address] . To run this code, I used quite a complex code. for item in receipt: print (":", item [0], " " \* (3 - Len (item [0])), ":", item [1], " " \* (6 - Len (item [0]).

By this stage, my code was running successfully, and I was ready to move onto my test plan. Overall, I think I did well with this code because I managed to complete it before due Aswell as ticking off all requirements from the assignment brief. I found it extremely helpful to make a flowchart and pseudocode for this scenario before creating the code, therefore I can use it as a guide. After this I created a test plan. My test plan states 3 diverse types of testing ()