School of Computer Sciences
Universiti Sains Malaysia

CPT111 – Principles of Programming
Academic Session 2020/2021, Semester 1

Assignment 2

Date: 17 JANUARY 2021

Name	**Muhammad Armand Bin Muhammad Fazli**
Matric No.	**153857**
Group	**B**

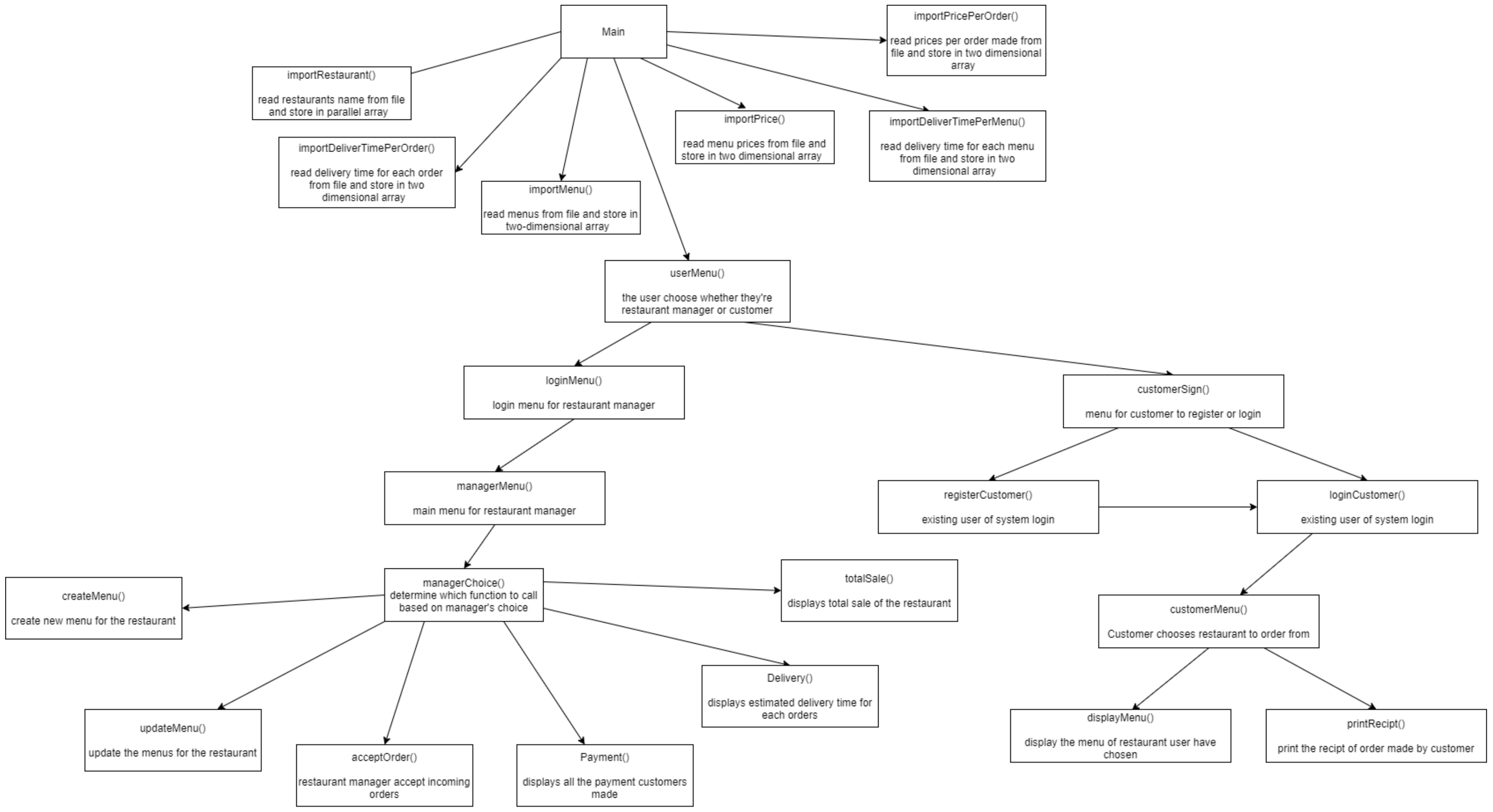
INTRODUCTION

This program is developed to allow customers to order food from restaurants to be delivered straight to their door, and allows restaurant managers to manage their restaurant as in , creating a new menu, updating menu, accepting orders, and more to maintain their business.It is a simple Food Ordering System made for convenience for both the customers and restaurant managers to use. A lot of two-dimensional array and parallel array are used to process information imported from multiple files . This is so that the program have continuity so that the program doesn’t lose data on every execution.

PROBLEM ANALYSIS

With COVID-19 making social distancing becoming the norm and eating in a crowded restaurant no longer being possible , it is a huge problem for people’s who want to enjoy delicious food from restaurants without having to go out and be in the risk of getting COVID-19. It is also a big problem for restaurants to maintain their business in this tough times we live in. Therefore, a Food Ordering System is needed to be developed to solve this problem where users can order food to be delivered straight to their door. The system also need to be full of features for the Restaurant Managers to use such as , creating a new menu ,updating menu price, calculate total sales, and more.

STRUCTURE CHART

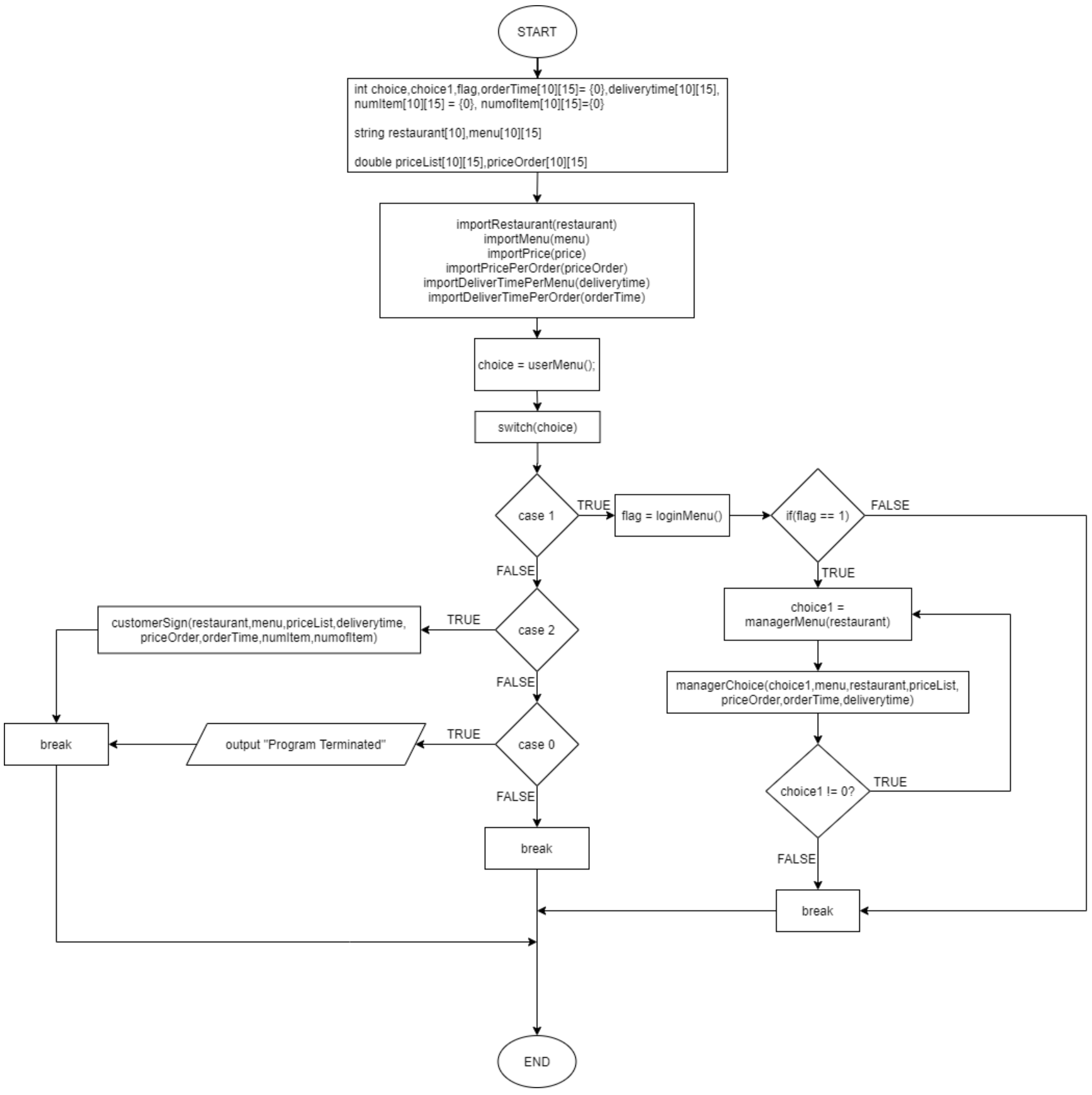


Main
importPricePerOrder()
read prices per order made from file and store in two dimensional array
importRestaurant()
read restaurants name from file and store in parallel array
importPrice()
read menu prices from file and store in two dimensional array
importDeliverTimePerMenu()
read delivery time for each menu from file and store in two dimensional array
importDeliverTimePerOrder()
read delivery time for each order from file and store in two dimensional array
importMenu()
read menus from file and store in two-dimensional array
userMenu()
the user choose whether they're restaurant manager or customer
loginMenu()
login menu for restaurant manager
customerSign()
menu for customer to register or login
managerMenu()
main menu for restaurant manager
registerCustomer()
existing user of system login
loginCustomer()
existing user of system login
managerChoice()
determine which function to call based on manager's choice
createMenu()
create new menu for the restaurant
totalSale()
displays total sale of the restaurant
customerMenu()
Customer chooses restaurant to order from
Delivery()
displays estimated delivery time for each orders
updateMenu()
update the menus for the restaurant
acceptOrder()
restaurant manager accept incoming orders
Payment()
displays all the payment customers made
displayMenu()
display the menu of restaurant user have chosen
printRecipt()
print the recipt of order made by customer

(For sharper image, refer to Structure.png included in the folder)

FLOWCHART

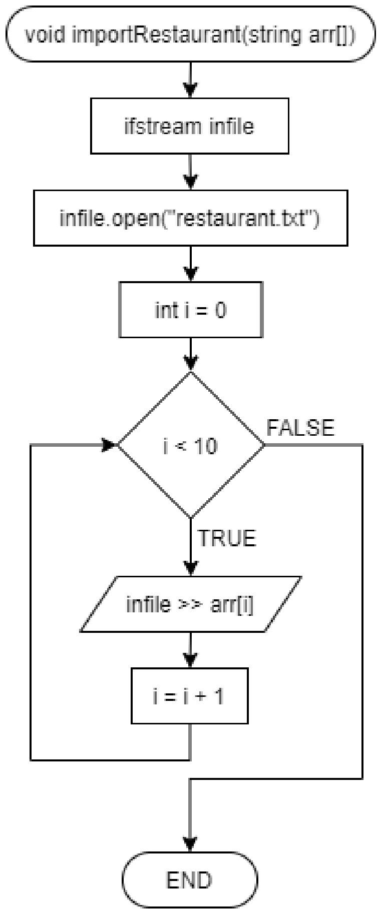
Main Flow Chart


START
int choice,choice1,flag,orderTime[10][15]= {0},deliverytime[10][15], numItem[10][15] = {0}, numofItem[10][15]={0}
string restaurant[10],menu[10][15]
double priceList[10][15],priceOrder[10][15]
importRestaurant(restaurant)
importMenu(menu)
importPrice(price)
importPricePerOrder(priceOrder)
importDeliverTimePerMenu(deliverytime)
importDeliverTimePerOrder(orderTime)
choice = userMenu();
switch(choice)
case 1
TRUE
flag = loginMenu()
if(flag == 1)
FALSE
TRUE
choice1 = managerMenu(restaurant)
managerChoice(choice1,menu,restaurant,priceList, priceOrder,orderTime,deliverytime)
choice1 != 0?
TRUE
FALSE
break
FALSE
case 2
TRUE
customerSign(restaurant,menu,priceList,deliverytime, priceOrder,orderTime,numItem,numofItem)
FALSE
case 0
TRUE
output "Program Terminated"
break
FALSE
break
END

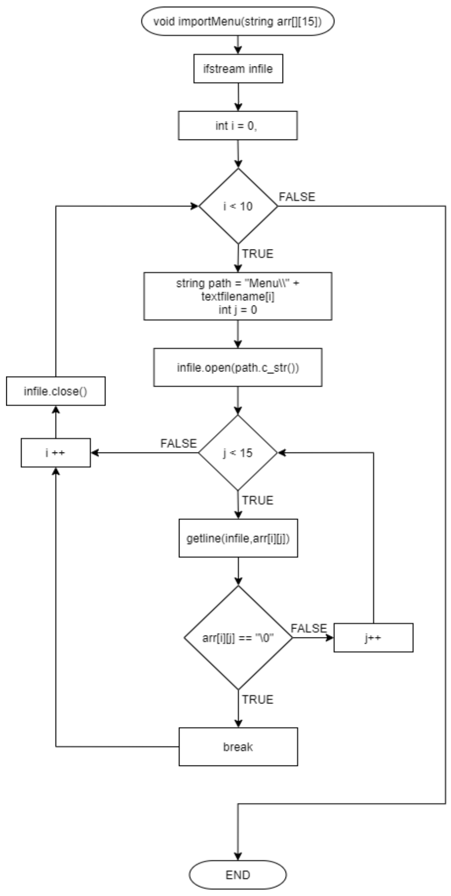
(For clearer image, refer to main.png in the Flowchart folder)

<u>Importing Datas Into Arrays Flowchart</u>

<u>Import Restaurant</u>



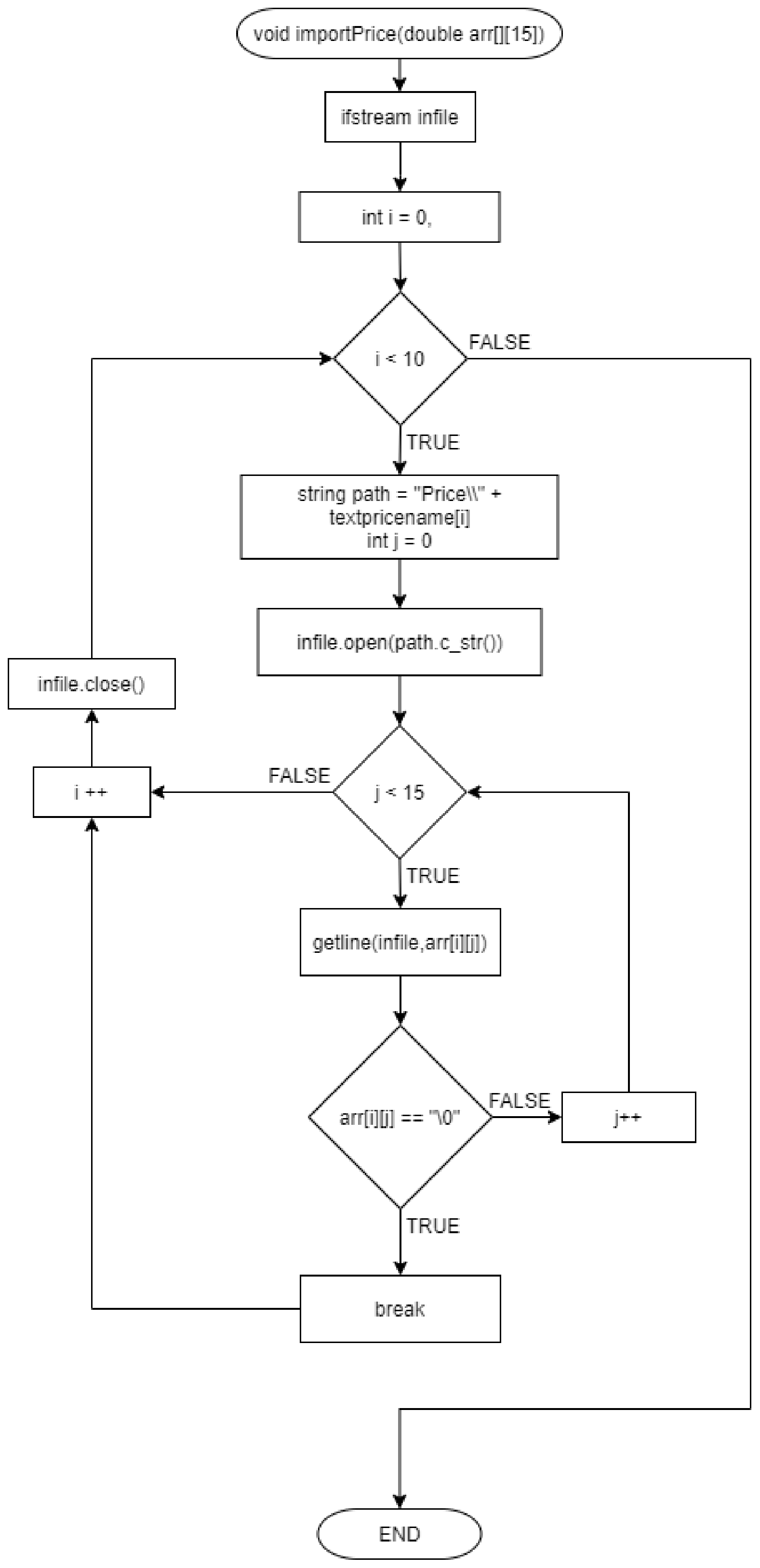
<u>Import Menus</u>


void importMenu(string arr[][15])
ifstream infile
int i = 0,
i < 10
FALSE
TRUE
string path = "Menu\\" + textfilename[i] int j = 0
infile.open(path.c_str())
infile.close()
i ++
FALSE
j < 15
TRUE
getline(infile,arr[i][j])
arr[i][j] == "\0"
FALSE
j++
TRUE
break
END

Import Restaurant is for importing the restaurant names into a parallel array meanwhile, Import Menu is for importing the menus of each restaurant into a two-dimensional array.

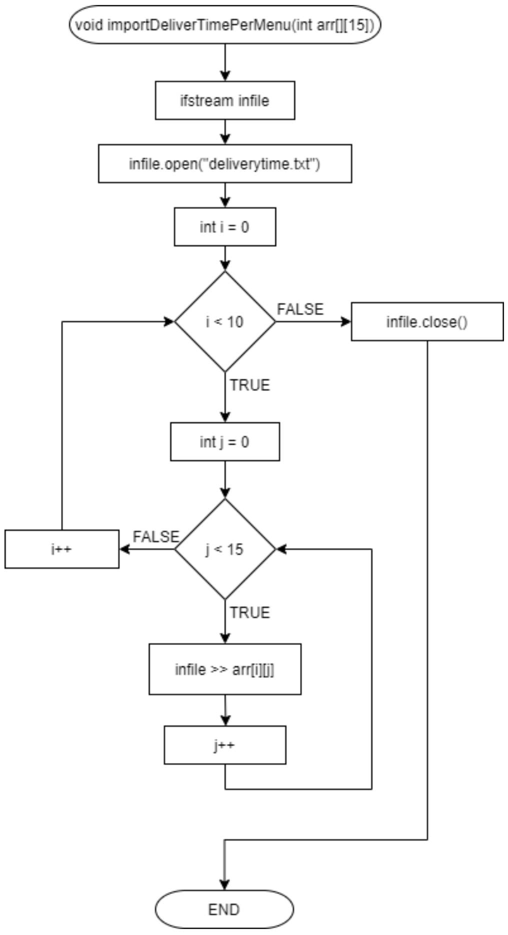
The names of the restaurant is imported from restaurant.txt meanwhile the menus are all imported from its respective file. Since there are 10 restaurants, there are 10 txt files containing the menus of each restaurant respectively.

Import Price

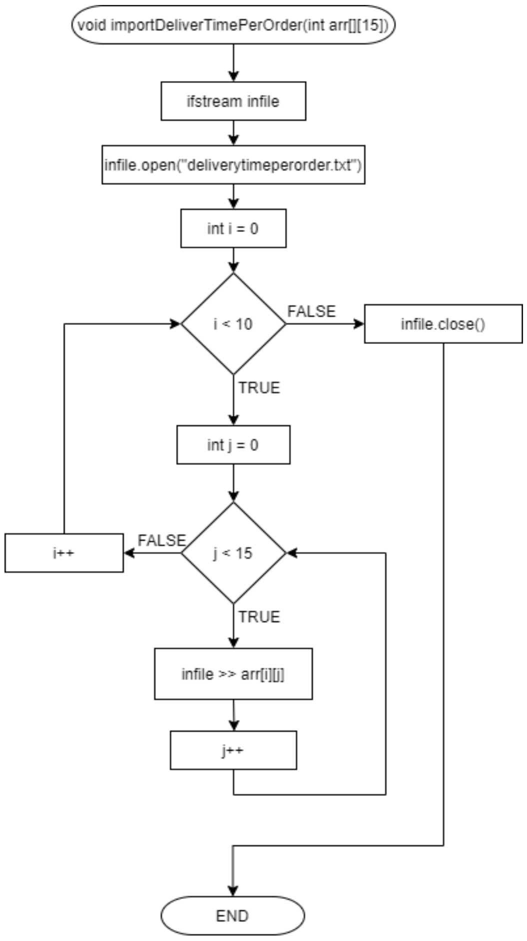

void importPrice(double arr[][15])
ifstream infile
int i = 0,
i < 10
FALSE
TRUE
string path = "Price\\" + textpricename[i]
int j = 0
infile.open(path.c_str())
infile.close()
i ++
FALSE
j < 15
TRUE
getline(infile,arr[i][j])
arr[i][j] == "\0"
FALSE
j++
TRUE
break
END

Import all of the prices of each menu of each restaurant into a two dimensional array from files

Import Delivery Time per Menu

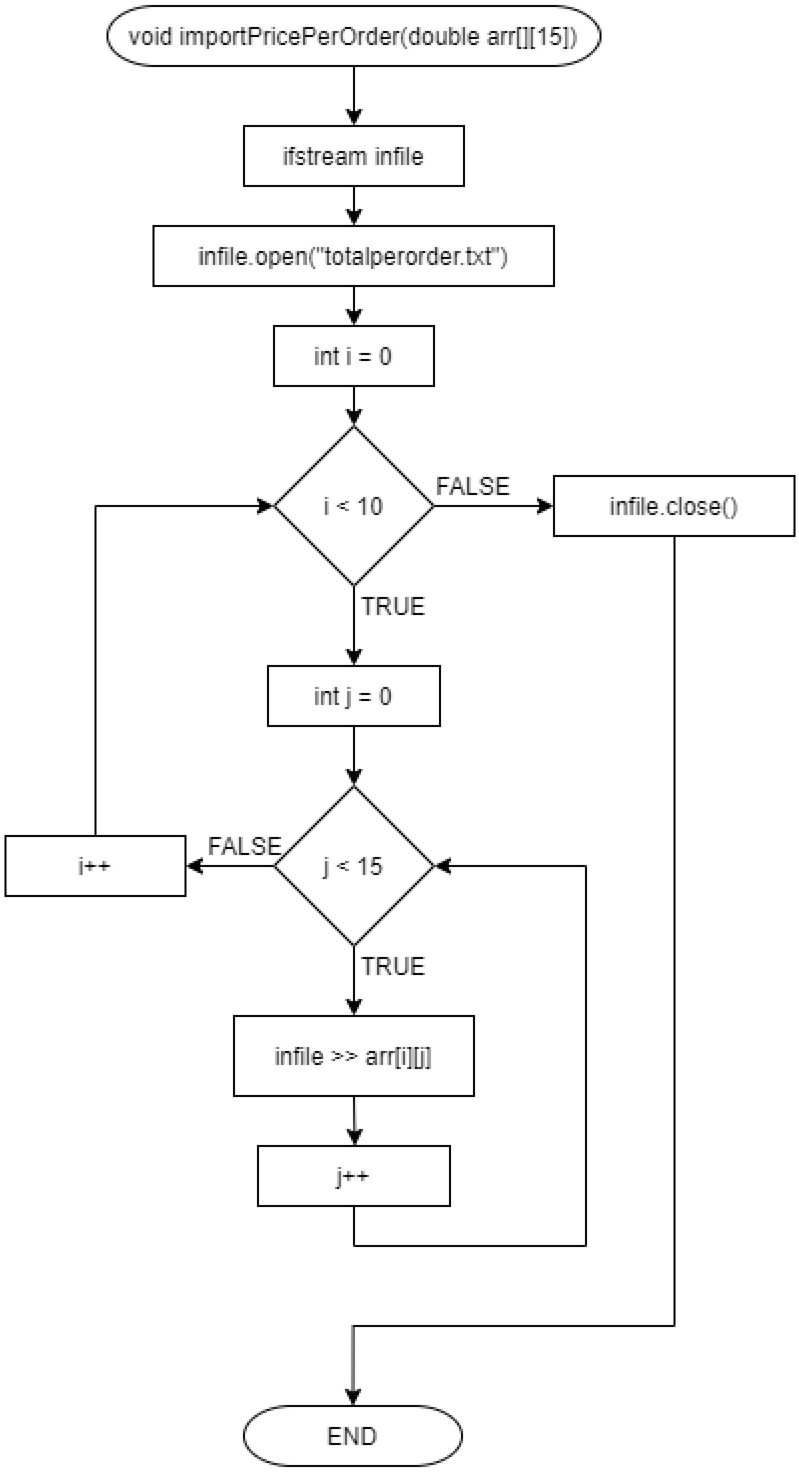


Import Delivery Time per Order



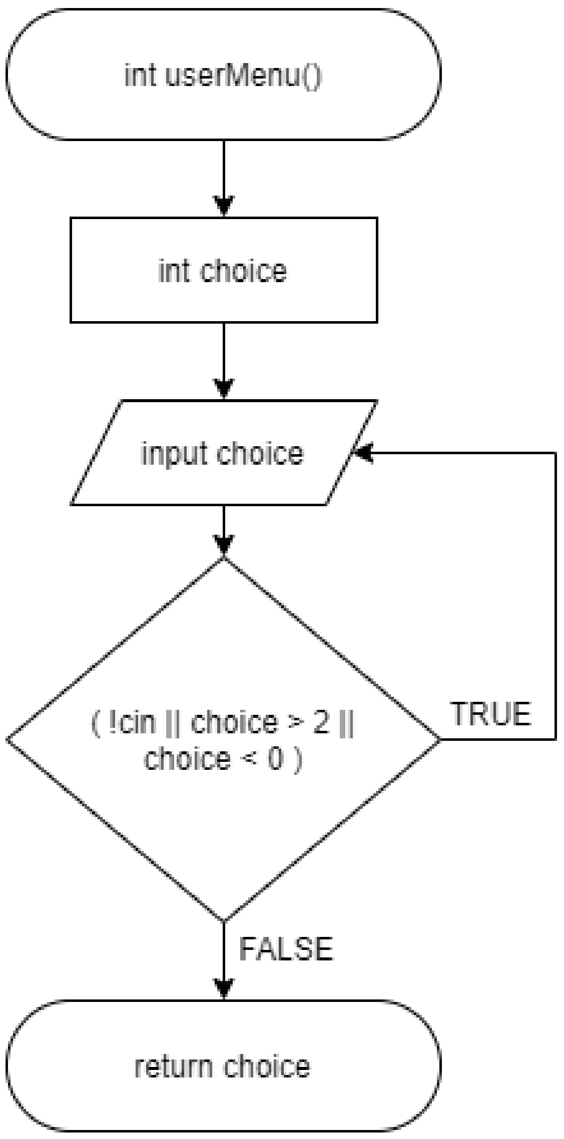
Import Delivery Time per Menu is to import all of the delivery time of each menu for each restaurant as all of the menu has different preparation time ,therefore different estimated delivery time. Import Deliver Time per Order is to import all of the delivery time for each order made to be accessed by restaurant managers later for the function Delivery()

Import Price Per Order



Import all the total price per order which was stored in a file, then stored into an array for it to be accessed later by restaurant managers in the function Payment()

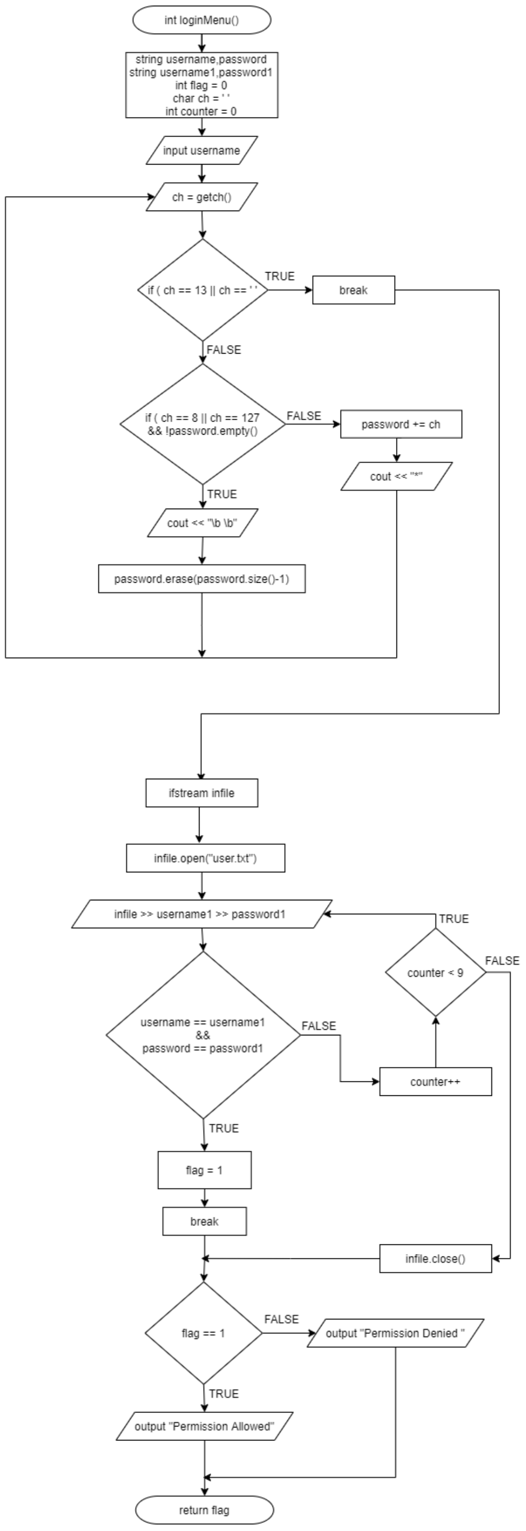
User Menu



User Menu is the first menu shown, to decide whether the user is a Restaurant Manager or a Customer

Manager’s Function Flowcharts

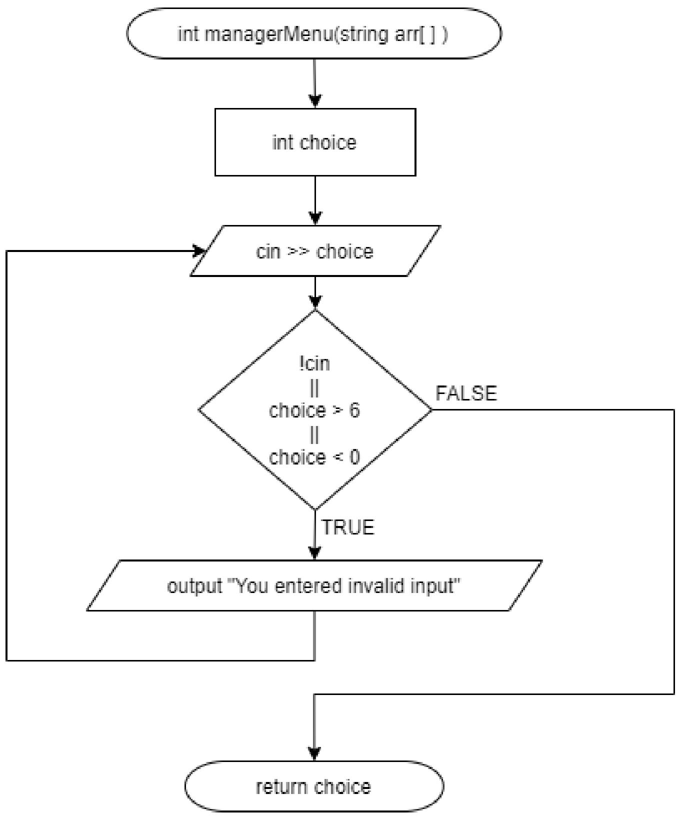
Login Menu



(For sharper image, refer to loginMenu.png included in the Flowchart folder)

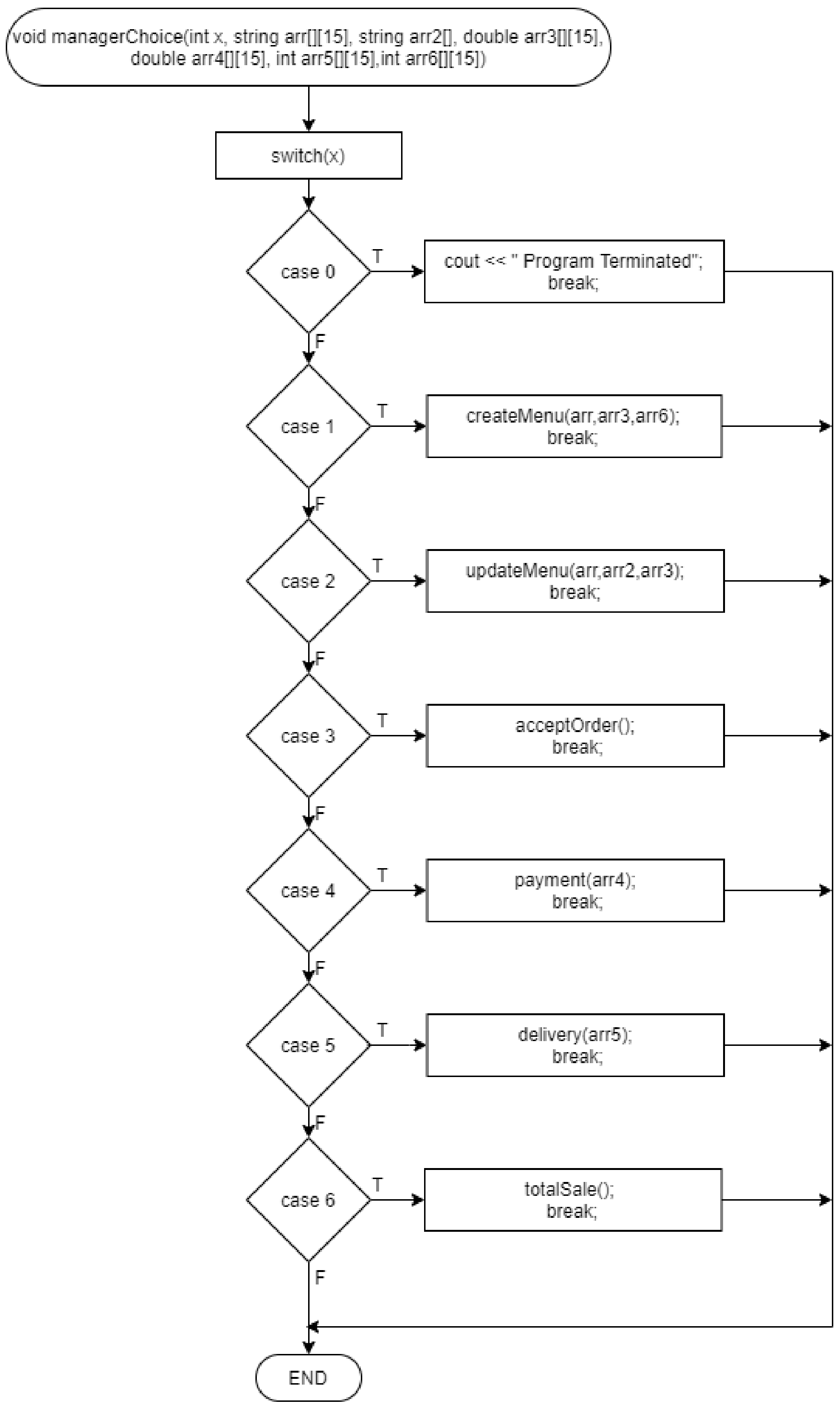
Login Menu is for the Restaurant Manager to input username and password of their account.

Manager Menu



Manager Menu is for the manager to choose on what to do, whether it is to create menu, update menu, accept order, display estimated delivery time, display total sale

Manager Choice

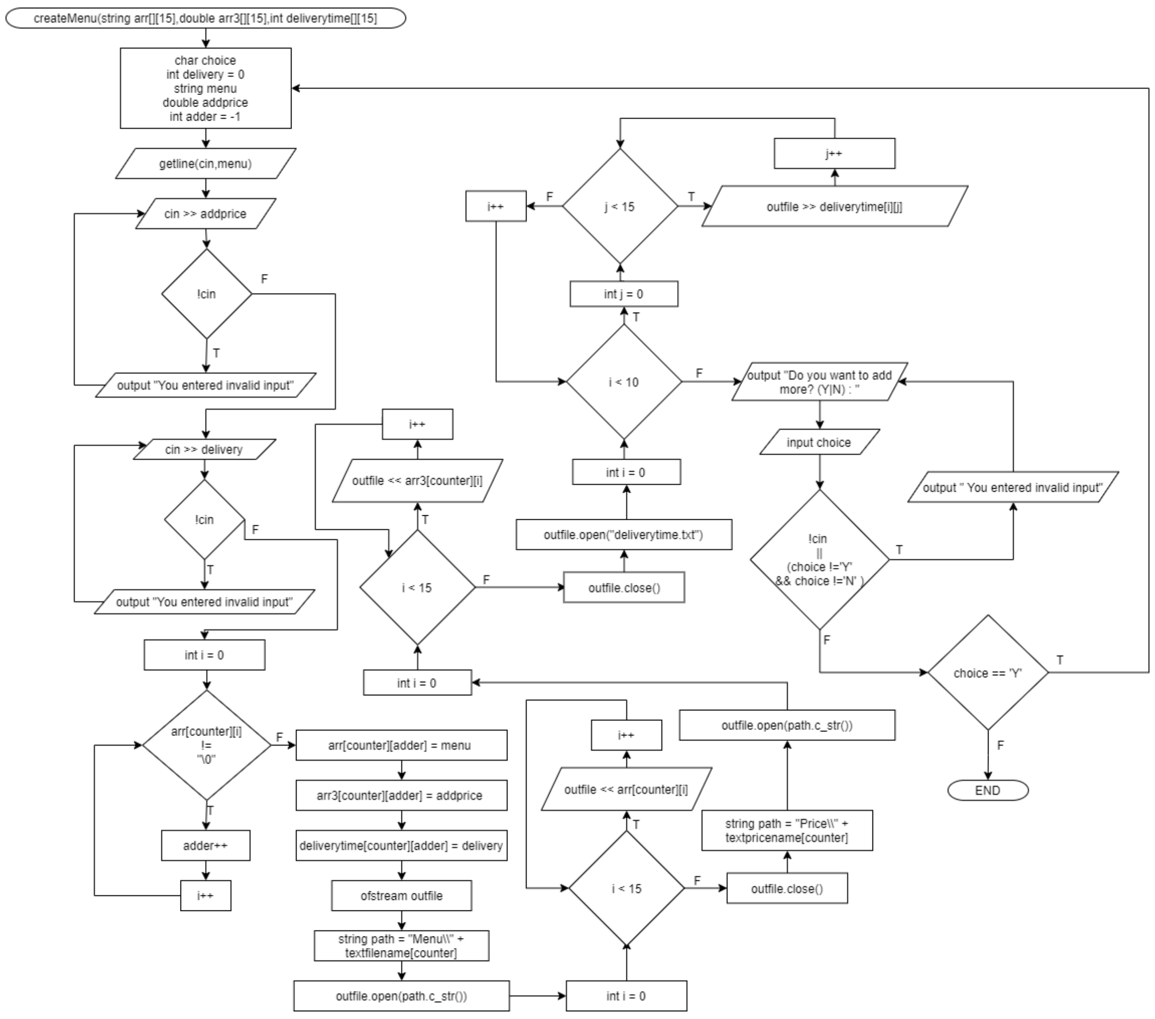


void managerChoice(int x, string arr[][15], string arr2[], double arr3[][15], double arr4[][15], int arr5[][15],int arr6[][15])
switch(x)
case 0
T
cout << " Program Terminated"; break;
F
case 1
createMenu(arr,arr3,arr6); break;
case 2
updateMenu(arr,arr2,arr3); break;
case 3
acceptOrder(); break;
case 4
payment(arr4); break;
case 5
delivery(arr5); break;
case 6
totalSale(); break;
END

(For sharper image, refer to managerChoice.png included in the Flowchart folder)

Manager Choice is the function that acts as the main of Manager's functions. It calls functions accordingly to the choice input by user in the Manager Menu function

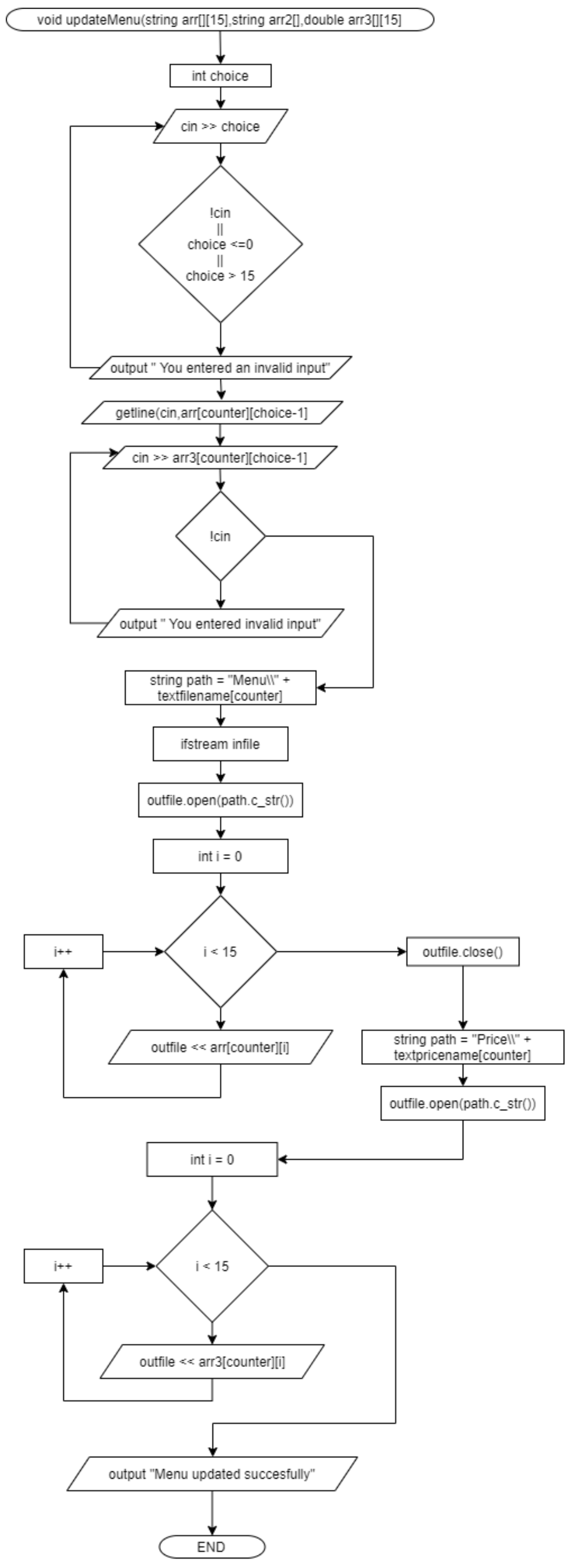
Create Menu


createMenu(string arr[][15],double arr3[][15],int deliverytime[][15]
char choice
int delivery = 0
string menu
double addprice
int adder = -1
getline(cin,menu)
cin >> addprice
!cin
F
T
output "You entered invalid input"
cin >> delivery
!cin
F
T
output "You entered invalid input"
int i = 0
arr[counter][i]
!=
"\0"
F
T
adder++
i++
arr[counter][adder] = menu
arr3[counter][adder] = addprice
deliverytime[counter][adder] = delivery
ofstream outfile
string path = "Menu\\" +
textfilename[counter]
outfile.open(path.c_str())
int i = 0
i < 15
T
F
outfile << arr[counter][i]
i++
outfile.close()
string path = "Price\\" +
textpricename[counter]
outfile.open(path.c_str())
int i = 0
i < 15
T
F
outfile << arr3[counter][i]
i++
outfile.close()
outfile.open("deliverytime.txt")
int i = 0
i < 10
T
F
int j = 0
j < 15
T
F
outfile >> deliverytime[i][j]
j++
i++
output "Do you want to add
more? (Y|N) : "
input choice
!cin
||
(choice !='Y'
&& choice !='N')
T
F
output " You entered invalid input"
choice == 'Y'
T
F
END

(For sharper image, refer to createMenu.png included in the Flowchart folder)

Create menu is a function that enables restaurant manager to add in new menus, their price, and estimated delivery time (estimated time to prepare and deliver) .

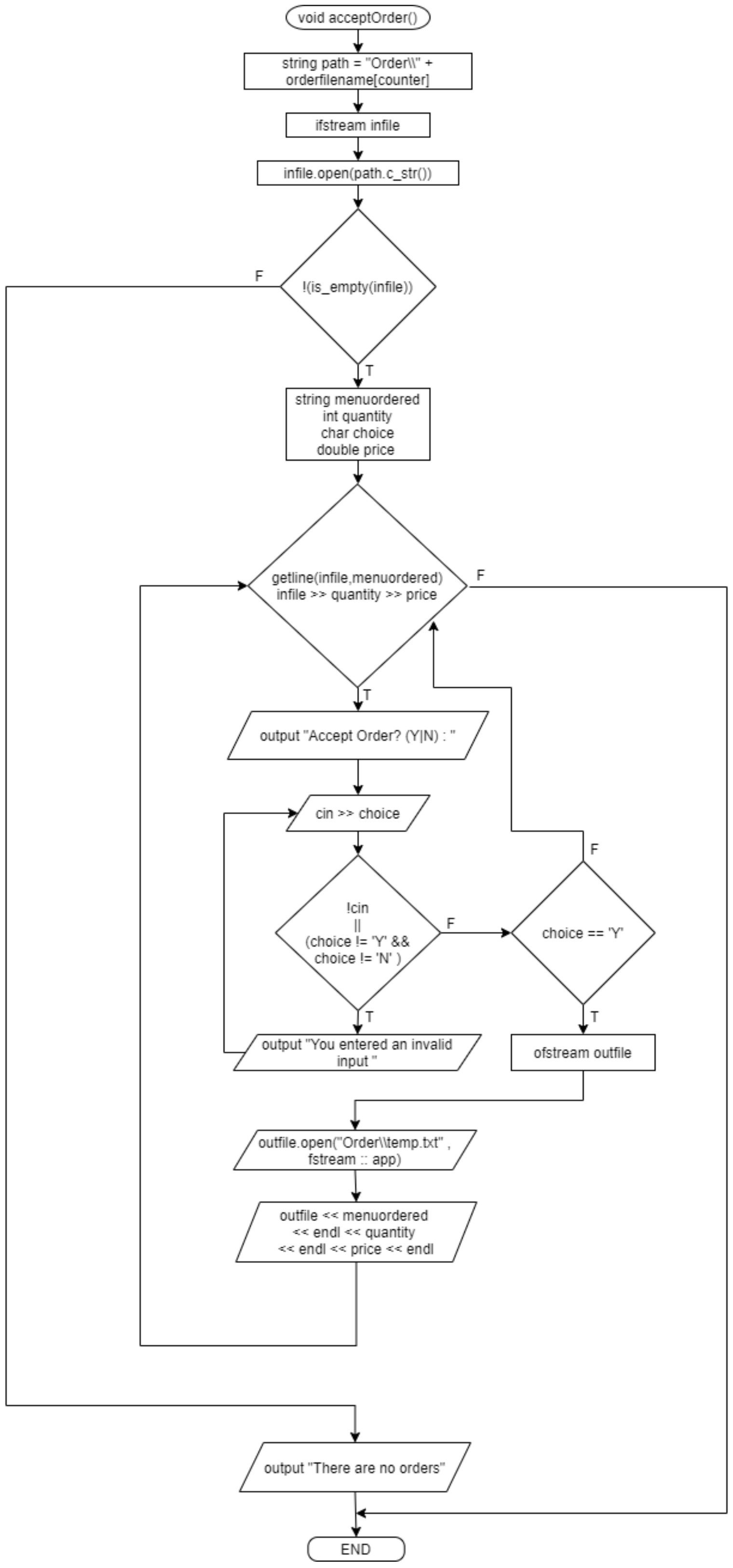
Update Menu


void updateMenu(string arr[][15],string arr2[],double arr3[][15]
int choice
cin >> choice
!cin || choice <=0 || choice > 15
output " You entered an invalid input"
getline(cin,arr[counter][choice-1]
cin >> arr3[counter][choice-1]
!cin
output " You entered invalid input"
string path = "Menu\\" + textfilename[counter]
ifstream infile
outfile.open(path.c_str())
int i = 0
i++
i < 15
outfile.close()
outfile << arr[counter][i]
string path = "Price\\" + textpricename[counter]
outfile.open(path.c_str())
int i = 0
i++
i < 15
outfile << arr3[counter][i]
output "Menu updated succesfully"
END

(For Sharper Image, refer to updateMenu.png included in the Flowchart folder)

Update Menu is a function that allows the manager to update old menus with new menus and their new price

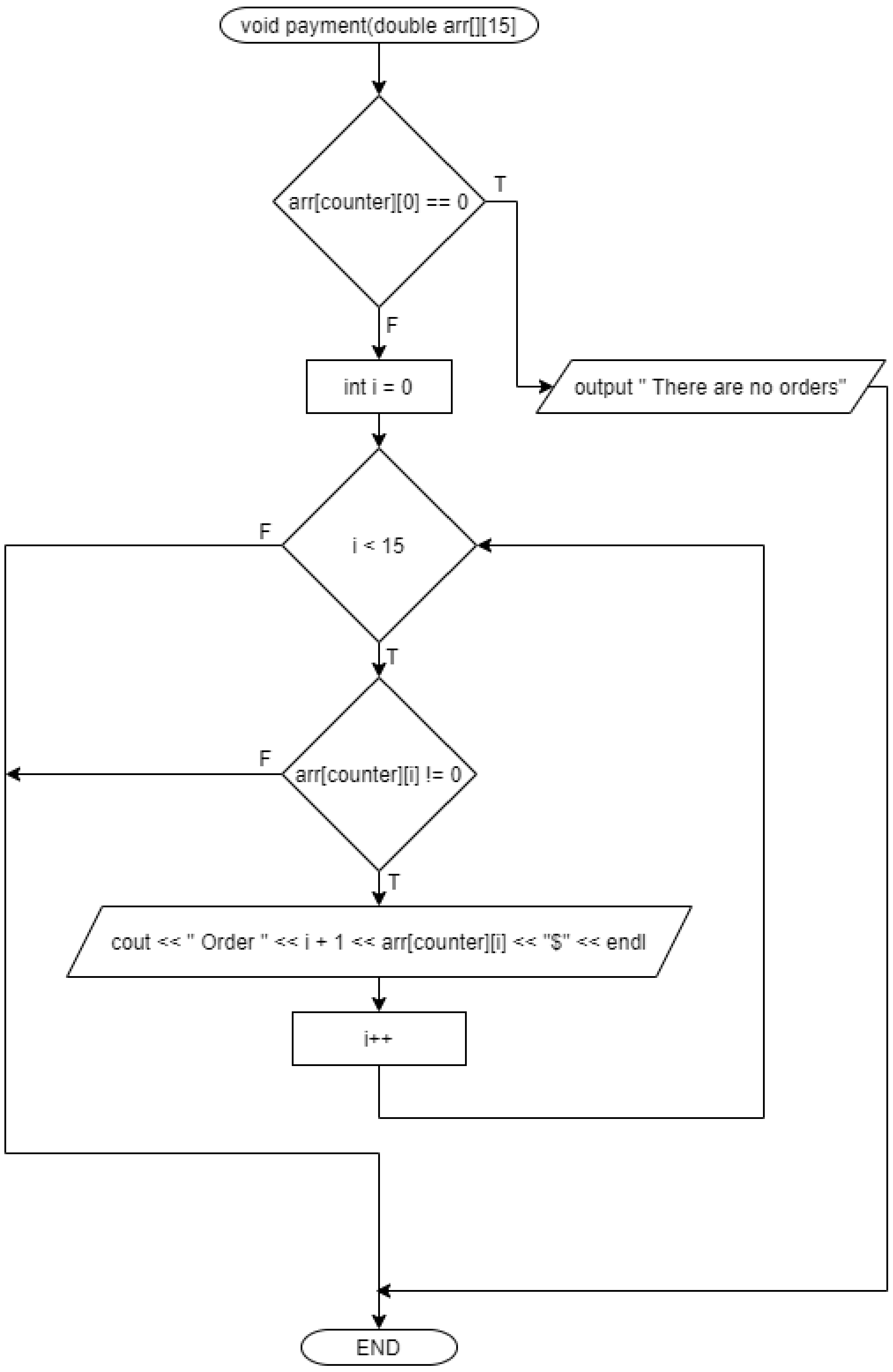
Accept/Reject Order


void acceptOrder()
string path = "Order\\" + orderfilename[counter]
ifstream infile
infile.open(path.c_str())
!(is_empty(infile))
F
T
string menuordered
int quantity
char choice
double price
getline(infile,menuordered)
infile >> quantity >> price
F
T
output "Accept Order? (Y|N) : "
cin >> choice
!cin
||
(choice != 'Y' &&
choice != 'N')
F
T
choice == 'Y'
F
T
output "You entered an invalid input "
ofstream outfile
outfile.open("Order\\temp.txt" , fstream :: app)
outfile << menuordered
<< endl << quantity
<< endl << price << endl
output "There are no orders"
END

(For sharper image, refer to acceptOrder.png included in the Flowchart folder)

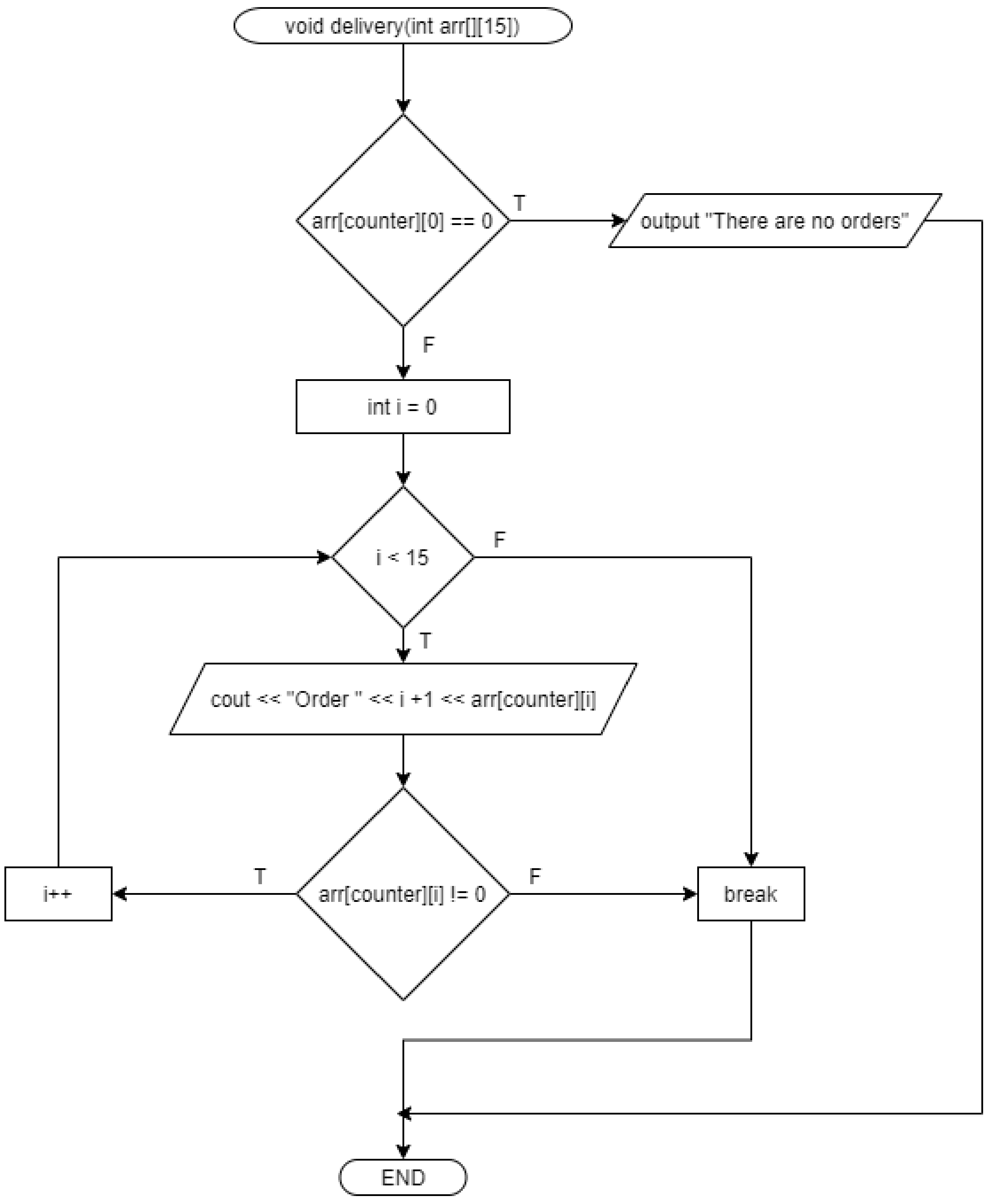
Accept Order function is a function that allows user to accept or reject orders made by users.

Payment


void payment(double arr[][15]
arr[counter][0] == 0
T
F
int i = 0
output " There are no orders"
i < 15
F
T
arr[counter][i] != 0
F
T
cout << " Order " << i + 1 << arr[counter][i] << "$" << endl
i++
END

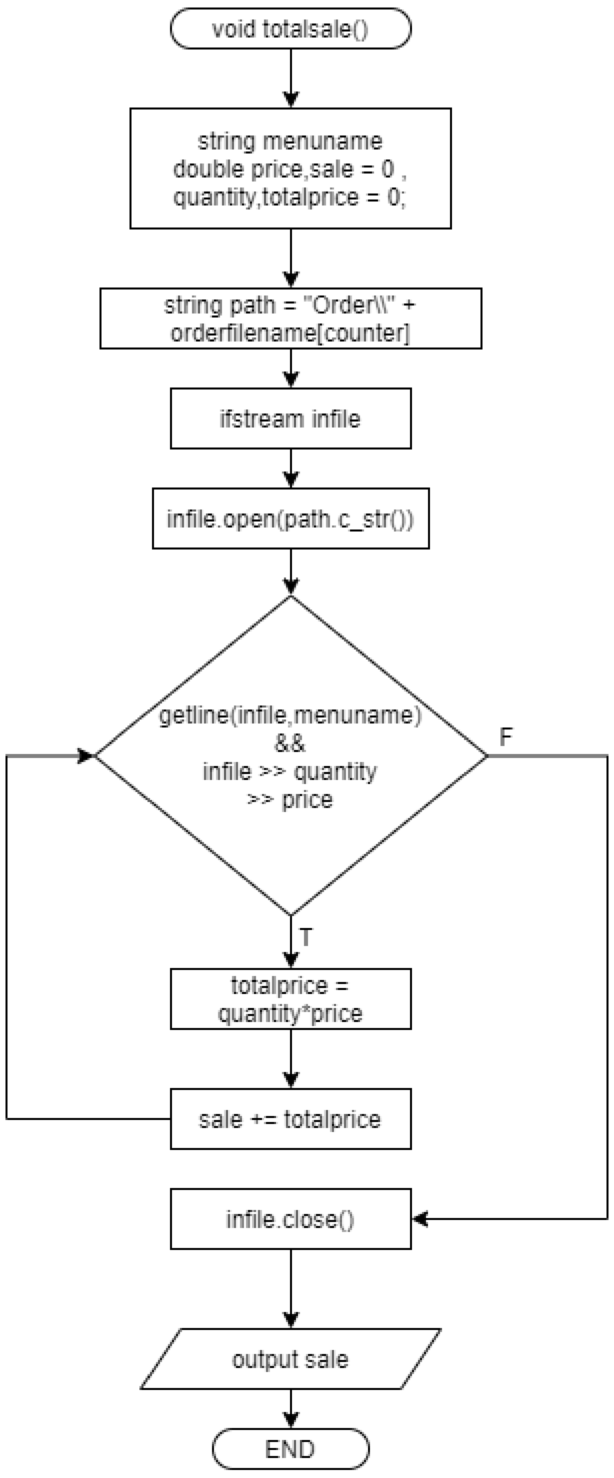
Payment is a function that display to the manager payments made by customers who ordered the foods from their restaurant

Delivery


void delivery(int arr[][15])
arr[counter][0] == 0
T
output "There are no orders"
F
int i = 0
i < 15
F
T
cout << "Order " << i +1 << arr[counter][i]
T
arr[counter][i] != 0
F
i++
break
END

Delivery is a function to shows the manager what is the estimated delivey time for each Orders made by customers.

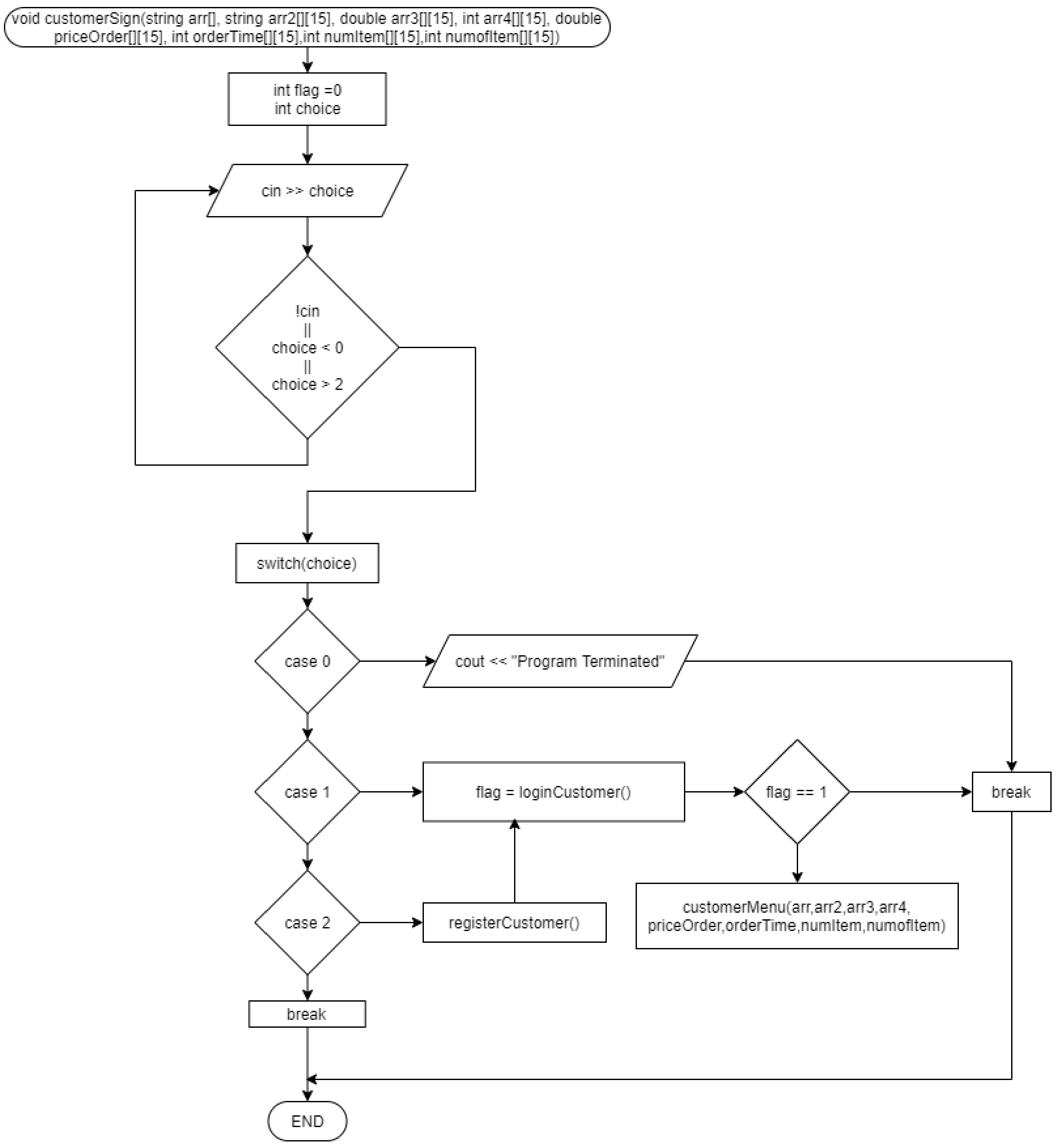
Total Sale



Total Sale is a function to calculate the total sales of the restaurant based on orders made by customers.

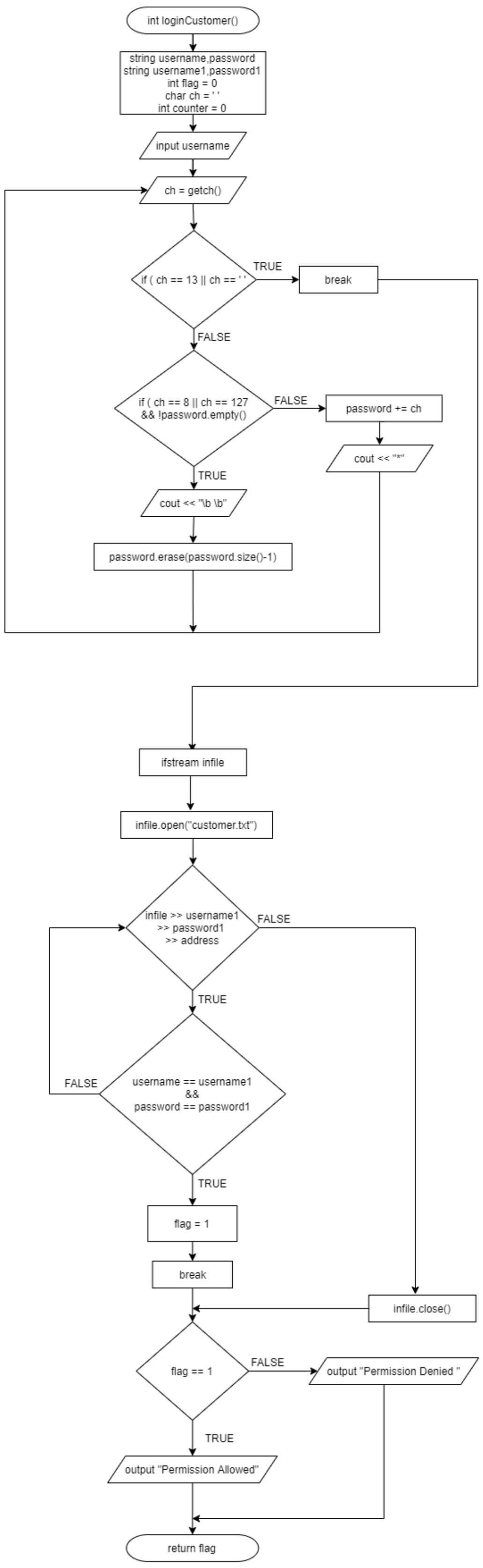
Customer’s Function Flowchart

Customer Sign


void customerSign(string arr[], string arr2[][15], double arr3[][15], int arr4[][15], double priceOrder[][15], int orderTime[][15],int numItem[][15],int numofItem[][15])
int flag =0
int choice
cin >> choice
!cin || choice < 0 || choice > 2
switch(choice)
case 0
cout << "Program Terminated"
case 1
flag = loginCustomer()
flag == 1
break
case 2
registerCustomer()
customerMenu(arr,arr2,arr3,arr4, priceOrder,orderTime,numItem,numofItem)
break
END

Customer Sign is a function that allows the user to register or login, which depends on whether they are a new user, or an existing user

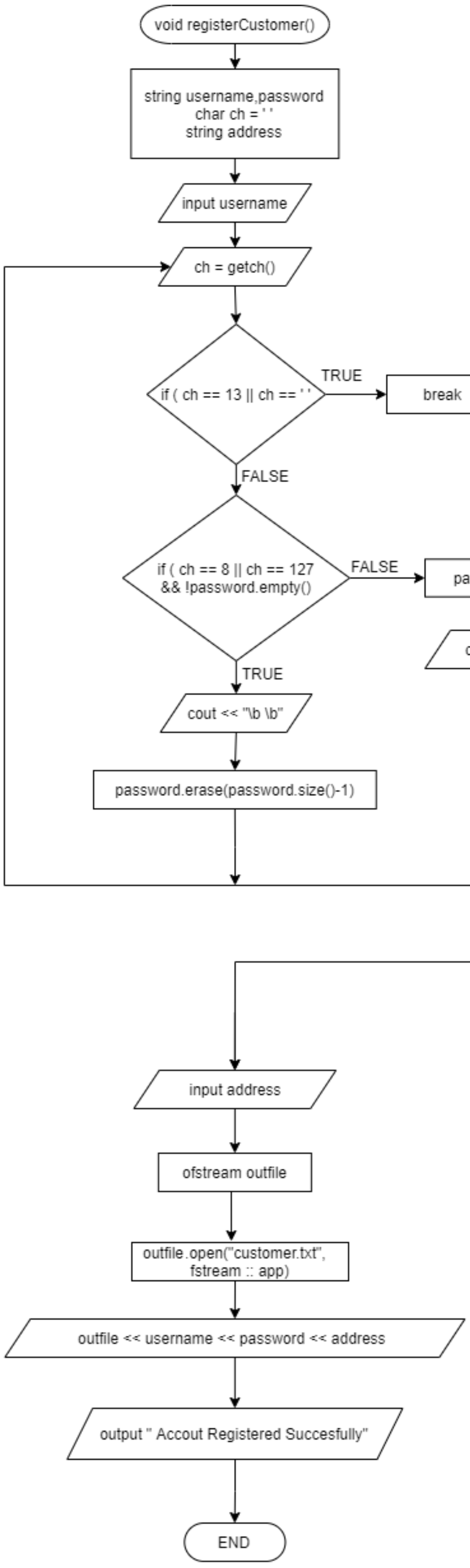
Login Customer


int loginCustomer()
string username,password
string username1,password1
int flag = 0
char ch = ' '
int counter = 0
input username
ch = getch()
if (ch == 13 || ch == ' '
TRUE
break
FALSE
if (ch == 8 || ch == 127 && !password.empty()
FALSE
password += ch
cout << "*"
TRUE
cout << "\b \b"
password.erase(password.size()-1)
ifstream infile
infile.open("customer.txt")
infile >> username1 >> password1 >> address
FALSE
TRUE
username == username1 && password == password1
FALSE
TRUE
flag = 1
break
infile.close()
flag == 1
FALSE
output "Permission Denied "
TRUE
output "Permission Allowed"
return flag

(For sharper image, refer to loginCustomer.png included in the Flowchart Folder)

Login Customer is a function that allows existing users (customers) in the system to login.

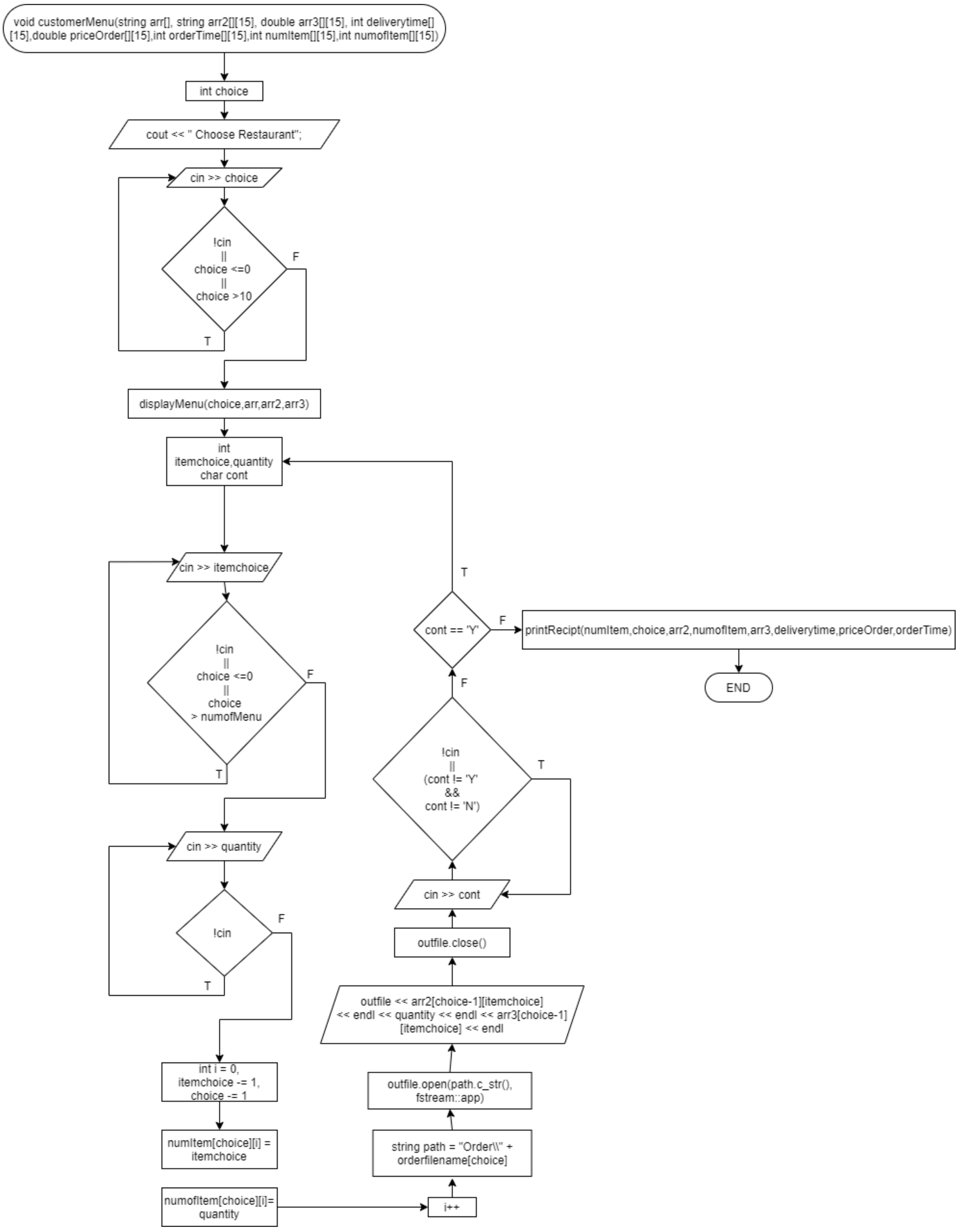
Register Customer



(For sharper image, refer registerCustomer.png which is included in the Flowchart Folder)

Register Customer is a function that allows new user to register an account to use the system.

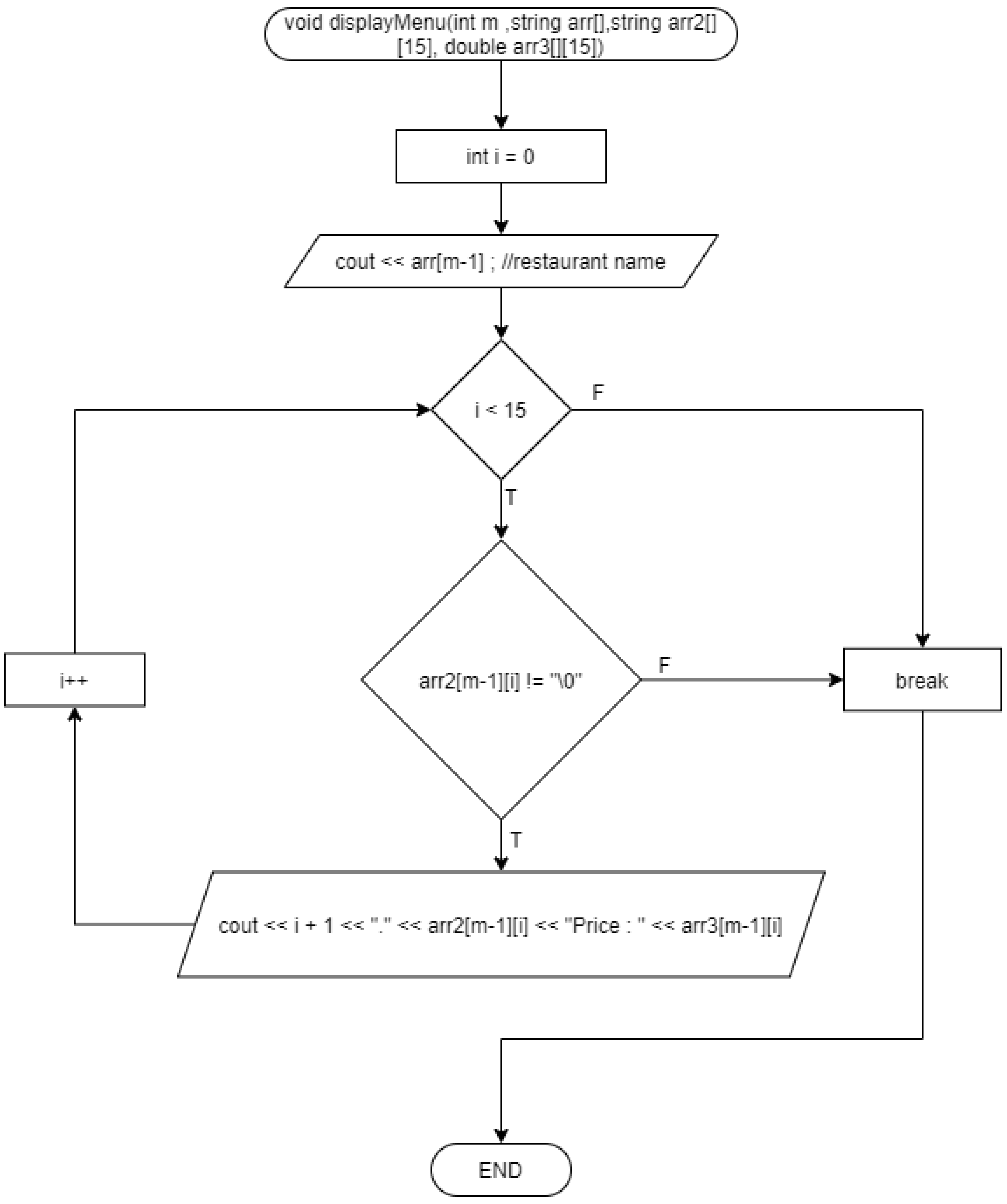
Customer Menu


void customerMenu(string arr[], string arr2[][15], double arr3[][15], int deliverytime[][15],double priceOrder[][15],int orderTime[][15],int numItem[][15],int numofItem[][15])
int choice
cout << " Choose Restaurant";
cin >> choice
!cin || choice <=0 || choice >10
F
T
displayMenu(choice,arr,arr2,arr3)
int itemchoice,quantity char cont
cin >> itemchoice
!cin || choice <=0 || choice > numofMenu
F
T
cin >> quantity
!cin
F
T
int i = 0, itemchoice -= 1, choice -= 1
numItem[choice][i] = itemchoice
numofItem[choice][i]= quantity
i++
string path = "Order\\" + orderfilename[choice]
outfile.open(path.c_str(), fstream::app)
outfile << arr2[choice-1][itemchoice] << endl << quantity << endl << arr3[choice-1][itemchoice] << endl
outfile.close()
cin >> cont
!cin || (cont != 'Y' && cont != 'N')
T
F
cont == 'Y'
T
F
printRecipt(numItem,choice,arr2,numofItem,arr3,deliverytime,priceOrder,orderTime)
END

(For sharper image, refer to customerMenu.png included in the Flowchart folder)

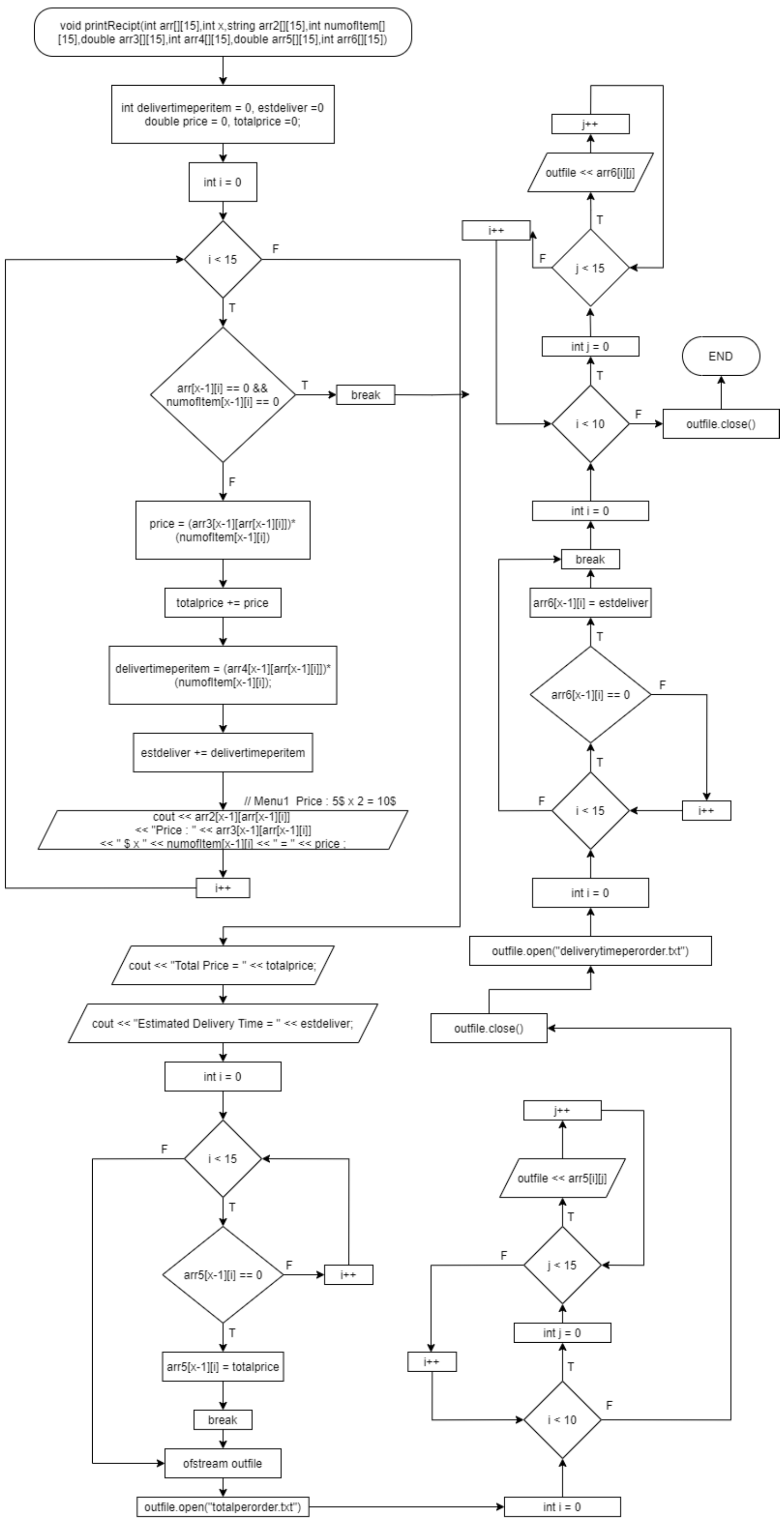
Customer Menu is essentially the main menu for customers. The user decides on which restaurant they want to order from which will then executes displayMenu based on user's choice. The user then orders food and state the quantity of it. Finally, the printRecipt will execute which will print the Recipt for the user.

Display Menu


void displayMenu(int m ,string arr[],string arr2[][15], double arr3[][15])
int i = 0
cout << arr[m-1] ; //restaurant name
i < 15
F
T
arr2[m-1][i] != "\0"
F
break
T
i++
cout << i + 1 << "." << arr2[m-1][i] << "Price : " << arr3[m-1][i]
END

Displays the menus of the restaurant user has chosen.

Print Recipt


void printRecipt(int arr[][15],int x,string arr2[][15],int numofItem[][15],double arr3[][15],int arr4[][15],double arr5[][15],int arr6[][15])
int delivertimeperitem = 0, estdeliver =0
double price = 0, totalprice =0;
int i = 0
i < 15
T
F
arr[x-1][i] == 0 && numofItem[x-1][i] == 0
break
price = (arr3[x-1][arr[x-1][i]])*(numofItem[x-1][i])
totalprice += price
delivertimeperitem = (arr4[x-1][arr[x-1][i]])*(numofItem[x-1][i]);
estdeliver += delivertimeperitem
// Menu1 Price : 5$ x 2 = 10$
cout << arr2[x-1][arr[x-1][i]] << "Price : " << arr3[x-1][arr[x-1][i]] << " $ x " << numofItem[x-1][i] << " = " << price ;
i++
cout << "Total Price = " << totalprice;
cout << "Estimated Delivery Time = " << estdeliver;
int i = 0
i < 15
arr5[x-1][i] == 0
i++
arr5[x-1][i] = totalprice
break
ofstream outfile
outfile.open("totalperorder.txt")
int i = 0
i < 10
int j = 0
j < 15
outfile << arr5[i][j]
j++
i++
outfile.close()
outfile.open("deliverytimeperorder.txt")
int i = 0
i < 15
arr6[x-1][i] == 0
i++
arr6[x-1][i] = estdeliver
break
int i = 0
i < 10
int j = 0
j < 15
outfile << arr6[i][j]
j++
i++
outfile.close()
END

(For sharper image, refer to printRecipt.png included in the Flowchart folder)

Prints the recipt for user and store the user’s order in a file , to be later accessed by restaurant managers

PROGRAM LISTING

```cpp
#include<iostream>
#include<fstream>
#include<iomanip>
#include<string>
#include<algorithm>
#include<iterator>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<cmath>
#include<vector>

// disclaimer : string(int,char) is a function that repeats a char, int times. This is used a lot in the program for the UI , for example, string(100,'-') will create a string that consists of 100 '-'


using namespace std;
int counter = 0;
int counter2 = 0;
ofstream outfile;
ifstream infile;

//stores the filename of the restaurant menus so that it can be accessed later by ifstream or ofstream to input and output ( file handling )
string textfilename[10] = {"tiffzmenu.txt", "nanathaimenu.txt", "claypotmenu.txt" , "hwapohmenu.txt", "mamamiamenu.txt",
                           "kingmenu.txt", "gloriamenu.txt","chocolatmenu.txt", "mykorimenu.txt","breadhousemenu.txt"};

//stores the filename of all the prices for all different menus on each restaurant

string textpricename[10] = {"tiffzmenu.txt", "nanaprice.txt", "claypotprice.txt","hwapohprice.txt","mamamiaprice.txt",
                            "kingprice.txt","gloriaprice.txt","chocolatprice.txt","mykoriprice.txt","breadhouseprice.txt"};

//stores the filename of all orders made by customers for each different restaurant

string orderfilename[10] = {"tiffzorder.txt","nanaorder.txt","claypotorder.txt","hwapohorder.txt","mamamiaorder.txt",
                             "kingorder.txt","gloriaorder.txt","chocolatorder.txt","mykoriorder.txt","breadhouse.txt"};

//========================================================[Function Prototype]==========================================================//
int userMenu();
void clear();
void importRestaurant(string arr[]);
void importMenu(string arr[][15]);
void importPrice(double arr[][15]);
int loginMenu();
int managerMenu(string arr[]);
void managerChoice(int x, string arr[][15], string arr2[], double arr3[][15], double arr4[][15], int arr5[][15],int arr6[][15]);
void importPricePerOrder(double arr[][15]);
void importDeliverTimePerOrder(int arr[][15]);
void importDeliverTimePerMenu(int arr[][15]);
void customerSign(string arr[], string arr2[][15], double arr3[][15], int arr4[][15], double priceOrder[][15], int orderTime[][15],int numItem[][15], int numofItem[][15]);

bool is_empty(std::ifstream& pFile)
```

```cpp
{
    return pFile.peek() == std::ifstream::traits_type::eof();
}
//=============================================================================================================================================//

//==========================================================[Main]=================================================================================//
int main(){
    int choice ,flag;
    string restaurant[10]; // stores the name of restaurant
    string menu[10][15]; // stores menus of each restaurant
    double pricelist[10][15]; // stores the prices of each menu
    double priceOrder[10][15] = {0}; //an array to store the total payment per order to display to the restaurant managers
    int orderTime[10][15] = {0}; // stores the delivery time of each orders made
    int deliverytime[10][15]; // stores the deliverytime of each menus
    int numItem[10][15] = {0}; //intialize all of the array to 0 so that it can be managed easily [This array contains the item code of the menu chosen by user,this item code can be used to access the array of menu[10][15]
    int numofItem[10][15] = {0}; // this array will store the quantity of order made of a certain item which corresponds to the array above , numItem[x][0] = 2 [Himitsu Pizza] and numofItem[x][0] = 1 [ the quantity of himitsu burger ordered ]

    importRestaurant(restaurant); //importing and storing the restaurant names from the txt file into the array
    importMenu(menu); // importing and storing the menus of the restaurant from the txt file into a 2d array
    importPrice(pricelist); //importing and storing the prices of the menus of each restaurant
    importPricePerOrder(priceOrder); //importing and storing all of the payment per order for each restaurant made so that it can be accessed by restaurant managers
    importDeliverTimePerMenu(deliverytime);//importing delivery time for each menu of each restaurant
    importDeliverTimePerOrder(orderTime); // importing delivery time for each order made



    choice = userMenu(); // the user choose wheter they're a restaurant manager or a customer
    clear(); // clear screen function is called

    //a switch case to decide whether the user is a restaurant manager or customer based on choice picked on the userMenu() function
    switch(choice)
    {
        //case 1 is for restaurant managers
        case 1 :    flag = loginMenu(); // flag is used to indicate wheter the restaurant manager managed to Log In with their respective username and password
                    if(flag == 1){ // if the restaurant manager entered the correct username and password, the following code inside the if statement will be executed

                        int choice1; // declaration of choice to store the return value of managerMenu();

                        // a do-while loop so that the user won't exit the program after doing one operation, this ables the user to only exit the program when choice1 is 0
                        do{
                            choice1 = managerMenu(restaurant); //stores the value of user choice in the function managerMenu() so that the do-while loop can run correctly
                            managerChoice(choice1, menu, restaurant, pricelist, priceOrder,orderTime,deliverytime); //managerChoice function is called.It is a function that consists of switch-case to determine the operation the user want to do
                        }while(choice1 != 0);
                    }else // if the user didn't input the username and password correctly, access will be denied to the user and the program is terminated, therefore, break;
                        break;
                    break; // break; after all the operations in the case are done

        //case 2 is for customers
        case 2 :    customerSign(restaurant,menu,pricelist,deliverytime,priceOrder,orderTime,numItem,numofItem); break; //customerSign is a function that is called to prompt up a menu to ask wheter the user is an existing user, or a new user, basically a log in and sign up page

        case 0 :    cout << " " << endl;cout << "  Program Terminated "; break;
    }


       }

       //============================================================================================================================================//

       void clear(){ //function to clear the console screen

           system("cls");
       }

       //=============================================================[MAIN MENU]==============================================================//
       int userMenu(){

           //userMenu is the first prompt in the console, which is why it is the Main Menu, the console output a menu and the user will choose whether they are a customer or a restaurant manager
           int choice;
           cout << "                            " << endl;
           cout << "  --------------------------" << endl;
           cout << "  |        MAIN MENU       | " << endl;
           cout << "  --------------------------" << endl;
           cout << "  | 1.Restaurant Manager   |" << endl;
           cout << "  | 2.Customer             |" << endl;
           cout << "  | 0.Quit                 |" << endl;
           cout << "  --------------------------" << endl;

           //input validation so that the user can only input 1,2,or 3 as an input, or else, it will go through a do-while loop asking for the input again until the input is entered correctly by user

           cout << "  Choose : ";
           while(!(cin >> choice) || choice > 2 || choice < 0){
               cout << "\n  You entered an invalid input. Try again.";
               cout << "\n  Choose : ";
               cin.clear();
               cin.ignore(123,'\n');
           }


            return choice; // return the choice to main ( for the switch-case )
       }
       //==========================================================================================================================================//


       //==========================================[IMPORTING ALL THE INPUTS FROM TXT FILE]=====================================================//
```

```cpp
        void importRestaurant(string arr[]){
            infile.open("restaurant.txt");

            for(int i = 0 ; i<10 ; i++){
                getline(infile, arr[i]);
            }
            infile.close();
        }

        void importMenu(string arr[][15]){

            for(int i = 0 ; i < 10 ; i++){
                string path = "Menu\\" + textfilename[i];
                infile.open(path.c_str());
                for(int j = 0; j < 15 ; j++){

                    if(getline(infile, arr[i][j]) == "\0"){
                        break;
                    }
                }
                infile.close();
            }


        }

        void importPrice(double arr[][15]){

            for(int i = 0 ; i < 10 ; i++){
                string path = "Price\\" + textpricename[i];
                infile.open(path.c_str());
                for(int j = 0; j < 15 ; j++){

                    if((infile >> arr[i][j]) == "\0"){
                        break;
                    }
                }
                infile.close();
            }


        }

        void importPricePerOrder(double arr[][15]){
            infile.open("totalperorder.txt");
            for(int i = 0; i < 10 ; i++){
                for(int j = 0; j < 15 ; j++){
                    infile >> arr[i][j];
                }
            }
            infile.close();
        }

        void importDeliverTimePerMenu(int arr[][15]){
            infile.open("deliverytime.txt");
            for(int i = 0 ; i < 10 ; i++){
                for(int j = 0; j < 15 ; j++){
                    infile >> arr[i][j];
                }
            }
            infile.close();
        }

        void importDeliverTimePerOrder(int arr[][15]){
            infile.open("delivertimeperorder.txt");
            for(int i = 0 ; i < 10 ; i++){
```

```cpp
            for(int j = 0; j <15 ; j++){
                infile >> arr[i][j];
            }
        }
        infile.close();
    }
    //==================================================================================================================================//



    //========================================[FUNCTIONS FOR RESTAURANT MANAGER]===========================================================//
    int loginMenu(){ // function for the Restaurant Manager to log in into their respective account, each restaurant have different accounts so that they can manage their own restaurant effectively
        string username,password; //declaraction of username and password to be inputted by the user
        string username1,password1; // declaraction of username1 and password1 which will be used to validate wheter the username and password are valid or not
        int flag = 0; // stores a flag as 0
        char ch = ' '; // used for the password inputting

        //output the login menu
        cout << "                             " << endl;
        cout << "  --------------------------" << endl;
        cout << "  |          LOGIN          |  " << endl;
        cout << "  --------------------------" << endl;
        cout << "    Username : " ; cin >> username; // user input their username
        cout << "    Password : " ;
        //a function to display each character as * when inputting the password for privacy

        do{
            ch = getch();

            if(ch == 13 || ch == ' ')
            {
                break;
            }
            if((ch==8 || ch==127) && !password.empty()){
                cout << "\b \b";
                password.erase(password.size()-1);
            }else{
                password+=ch;
                cout << "*";

            }
        }while(ch != 13 || ch != ' ');


        cout << endl << "  ---------------------------" ;

        //ifstream is used to open the user.txt file which stores all the username and passwords for all of the restaurant
        infile.open("user.txt");
        //a loop to input username1 and password1 again and again until the end of file
        while(infile >> username1 >> password1){
            if(username == username1 && password == password1){ // checks if the username and password input by user is valid by comparing to the username and password stored in the user.txt
                flag = 1; // allocate value 1 to flag if the above statement is correct
                break; //break is used so that when the username and password is found, the loop end as there is no longer the need to check until the end of file
            }else
                counter++; //counter is a global variable that is used to count how many time the loop repeats so that we can use it to access the restaurants and menus later because the username and password are stored chronologically to the restaurant in the user.txt file
        }

        infile.close(); // close the user.txt file

        if(flag == 1){ // if the username and password are valid, output the statement below
            cout << "\n                    ";
            cout << "\n  Permission Allowed";
            cout << "\n  Press Enter To Continue...";
            cin.ignore();
            cin.get(); // makes the user press Enter for the program to continue
            clear(); //clear the console
        }
        else{ // if the username and password are invalid , output the statement below
            cout << "\n                    ";
            cout << "\n  Permission Denied";
        }

        return flag; // return the flag to the main
    }

    int managerMenu(string arr[]){ // managerMenu is basically the menu displayed after the restaurant manager logged in. The value passed ( string arr[] ) is the array that contains the restaurant name
        int choice;
        int l = arr[counter].length(); // determine the length of the name of the restaurants because the table/menu displayed varies in size for each restaurant, this is to make the allignment of the menu more clean

        int tablel = l*3; // the size of the table/menu is 3 times the length of the restaurant name
        cout << "                                       " << endl;
        cout << "  " << string(tablel+1,'-') <<endl;
        cout << "  |"<< setw(l) <<""<< arr[counter] << setw((tablel-l)-l) << "|"<<  endl; //setw() is also used for allignment purposes
        cout << "  " << string(tablel+1, '-') << endl;
        //display all the choices of operation the restaurant manager can choose
        cout << "  |" <<  " "<<"1.Create Menu" << setw(tablel-14)<<"|" <<endl;
        cout << "  |" <<  " "<<"2.Update Menu" << setw(tablel-14)<<"|" <<endl;
        cout << "  |" <<  " "<<"3.Accept Orders" << setw(tablel-16)<<"|" <<endl;
        cout << "  |" <<  " "<<"4.Payment" << setw(tablel-10)<<"|" <<endl;
        cout << "  |" <<  " "<<"5.Delivery" << setw(tablel-11)<<"|" <<endl;
        cout << "  |" <<  " "<<"6.Total Sale" << setw(tablel-13)<<"|" <<endl;
        cout << "  |" <<  " "<<"0.Quit" << setw(tablel-7)<<"|" <<endl;
        cout << "  " << string(tablel+1,'-') << endl;
        cout << endl;
        cout << "\n  Choose : ";
        while(!(cin >> choice) || choice > 6 || choice < 0){ // input validation
            cout << "\n  You entered an invalid input. Try again.";
            cout << "\n  Choose : ";
            cin.clear();
            cin.ignore(123,'\n');
        }

        return choice; //the choice is returned to main to be sent into another function which is managerChoice() for a switch-case
    }

    void createMenu(string arr[][15], double arr3[][15], int deliverytime[][15]) { //create menu is the function to add in a menu into the menu of the restaurant, this is case 1
            char choice;
            int delivery = 0;
            do{ // a do while loop so that the user can enter as many new menu as they want
                string menu; // declared to store the name of the menu want to be created/added
                double addprice; //declared to store the price of the new added menu
                int adder=0; // adder is declared to determine the index of the new menu and new price in the array of menu[10][15] and pricelist[10][15]
                adder -= 1; // to make adder value to be -1 because the index will count from 0,
                outfile.open("Menu\\temp.txt"); // a temp.txt which is a temporary file is created
                cin.ignore();
                cout << "                                " << endl;
                cout << "  Enter menu name to be added : " ;
                getline(cin,menu); // user input the new menu to be added
                cout << "  Enter price of menu : ";

                while(!(cin >> addprice)){ // input validation
                    cout << "\n  You entered an invalid input. Try again.";
                    cout << "\n  Enter price of menu : ";
                    cin.clear();
                    cin.ignore(123,'\n');
                }
                cout << "  Enter estimated delivery time for this menu : ";
                while(!(cin >> delivery)){ // input validation
                    cout << "\n  You entered an invalid input. Try again.";
                    cin.clear();
                    cin.ignore(123,'\n');
                }

                //cin >> addprice; // user input the price for the new menu to be added

                for(int i = 0 ; i < 15 ; i++){ // a for loop to store the new values input
                    if(arr[counter][i] != "\0"){ // since the array size is [10][15] , but the only data stored are [10][10] so far, the are 5 values that are not initialised for every restaurant. \0 is the value of an empty line, the statement is that if the value is not empty, output the value to the temp.txt
                        outfile << arr[counter][i] << endl;
                        adder++; // used to keep track of the index
                    }
                    else if(arr[counter][i] == "\0"){ // when an empty line in the array is found, it will be allocated by the new menu, therefore,the new menu will be inserted into the array without deleting or replacing any old values
                        outfile << menu;
                        arr[counter][i] = menu;
                        adder++; // used to keep track of the index
                        break; // break when the new menu is output into the temp.txt file
                    }
                }
                // arr3 is the array of priceList which stores the prices of the menus of each respective restaurant
                arr3[counter][adder] = addprice; //since adder was used above the keep track of the index that is supposed to store the new menu,the price can be easily stored in the same index as the new menu as they all correspond as i organised the file accordingly to the menu and price
                outfile.close();

                //now that the price is stored in the array, it will be useless if it isn't outputted to a file to be stored
                outfile.open("Price\\temp.txt"); // open a new file named temp.txt in the folder Price
                for(int i = 0 ; i < 15 ; i++){
                    outfile << arr3[counter][i] << endl; //output every pricelist into the file
                }
                outfile.close();
                //for the PRICE
                string def4 = "Price\\";
                string pathnameprice = def4 + textpricename[counter]; // this method is used because each prices are stored in different files for each restaurant, therefore this method is used for flexibility of the program to output to the correct file respective to the restaurant
                remove(pathnameprice.c_str()); // remove the original pricelist file
                rename("Price\\temp.txt", pathnameprice.c_str()); // rename the temp.txt file to the name of the original pricelist file

                //for delivery time of the new menu
                for(int i = 0 ; i < 15 ; i++){
                    if(deliverytime[counter][i]==0){
                        deliverytime[counter][i] = delivery;
                        break;
                    }
                }
                outfile.open("deliverytime.txt");
                for(int i = 0 ; i < 10 ; i++){
                    for(int j = 0 ; j < 15 ; j++){
                        outfile << deliverytime[i][j] << endl;
                    }
                }
                outfile.close();

                cout << "  Menu added succesfully." << endl;
                cout << "  Do you want to add more menu? (Y|N): ";
                // user input Y or N to determine whether they want to add more menus or end their order
                while(!(cin >> choice) || (choice != 'Y'  && choice != 'N')){ // input validation
                    cout << "\n  You entered an invalid input. Try again.";
                    cout << "\n  Do you want to add more menu? (Y|N) : ";
                    cin.clear();
                    cin.ignore(123,'\n');
                }
                //for the MENU
                string def = "Menu\\";
                string pathname = def + textfilename[counter];
                remove(pathname.c_str());
                rename("Menu\\temp.txt", pathname.c_str());

            }while(choice == 'Y');

            cout << "\n  Press any key to go back to main menu ...";
            cin.ignore();
            cin.get();
            clear();
    }

```

```cpp
        void updateMenu(string arr[][15], string arr2[], double arr3[][15]){ //a function so that the restaurant manager can update the menu, and their price

                int choice;
                //============for menu size allignment optimization/flexibility=============
                int l = arr2[counter].length();
                int tablel = l*3;
                //=====================================================================
                cout << "                                       " << endl;
                cout << "  " << string(tablel+1,'-') <<endl;
                cout << "  |"<< setw(l) <<""<< arr2[counter] << setw((tablel-l)-l) << "|"<<  endl;
                cout << "  " << string(tablel+1, '-') << endl;
                int j = 0;
                for(int i = 0 ; i<15 ; i++){ // a for loop to print the menus of the restaurant and the prices of the menus respectively
                    int character = arr[counter][i].length();

                    if(arr[counter][i] != "\0"){
                        j = j + 1; // used to count how many menus are inside the array so that it can be used for input validation later
                        if(i+1 < 10){ // check if the the number of the menu is less than 10 or not because the allignment are different because (1-9) are 1 digit meanwhile (10 and above ) are two digits
                            if(arr3[counter][i] < 10){ // check if the price of the menu is less that 10 or not because the allignment are different
                                cout << "  |" << " " << i+1 << "  - " << arr[counter][i] <<right << setw(tablel-character-15) << " " << fixed << setprecision(2) << arr3[counter][i] << "$"<< setw(tablel-(6+character+(tablel-character-15)+5)) <<"|" << endl;
                            }else{
                                cout << "  |" << " " << i+1 << "  - " << arr[counter][i] <<right << setw(tablel-character-15) << " " << fixed << setprecision(2) << arr3[counter][i] <<"$"<< setw(tablel-(6+character+(tablel-character-15)+6)) <<"|" << endl;
                            }
                        }else{
                            if(arr3[counter][i] < 10){
                                cout << "  |" << " " << i+1 << " - " << arr[counter][i] << right << setw(tablel-character-15) << " " << fixed << setprecision(2) << arr3[counter][i] <<"$" << setw(tablel-(5+character+(tablel-character-15)+6)) << "|" << endl;
                            }else{
                                cout << "  |" << " " << i+1 << " - " << arr[counter][i] <<right << setw(tablel-character-15) << " " << fixed << setprecision(2) << arr3[counter][i] <<"$"<< setw(tablel-(6+character+(tablel-character-15)+6)) <<"|" << endl;
                            }
                        }

                    }
                }
                cout << "  " << string(tablel+1,'-') << endl;
                cout << endl;
                cout << "  Choose the menu number you want to update : ";
                while(!(cin >> choice) || (choice <= 0 || choice > j)){ // input validation
                    cout << "\n  You have entered an invalid input. Try again.";
                    cout << "\n  Choose the menu number you want to update : ";
                    cin.clear();
                    cin.ignore(123,'\n');

                }



                cin.ignore();
                cout << "  New name of the menu : ";
                getline(cin, arr[counter][choice-1]); // user input the new name of the menu to be updated, it replace the old menu in the menu array, choice-1 is used as index because the user choose the menu from [1- ?] , An array is counted from [0-?]
                cout << "  Price of the updated menu : ";
                 // user input the new price for the menu to be updated , it replace the old price in the priceList array
                while(!(cin >> arr3[counter][choice-1])){ // input validation
                    cout << "\n  You entered an invalid input. Try again.";
                    cout << "\n  Price of the updated menu : ";
                    cin.clear();
                    cin.ignore(123,'\n');
                }
                cout << "  Menu Updated Succesfuly";
                cout << "\n  Press any key to return to main menu ...";
                cin.ignore();
                cin.get();


                outfile.open("Menu\\temp.txt"); // open/create a temp.txt which is a temporary file in the menu folder

                for(int i = 0 ; i < 15 ; i++){
                    if(arr[counter][i] == "\0"){ // if the array at index i is empty, break the loop
                        break;
                    }else{ // else if the array at index i is not empty, output it into the temp.txt file, this temp.txt file stores the latest value ( updated value )

                        outfile << arr[counter][i] << endl;
                    }
                }
                outfile.close(); // file is closed

                string def1 = "Menu\\";
                string pathname1 = def1 + textfilename[counter];
                remove(pathname1.c_str()); // remove the original menu file
                rename("Menu\\temp.txt", pathname1.c_str()); // rename the temp.txt file to the name of the original menu file

                outfile.open("Price\\temp.txt"); // create/open a temp.txt file which is a temporary file in the Price folder

                for(int i = 0 ; i < 15 ; i++){
                        outfile << arr3[counter][i] << endl; // output all of the latest price to the temp.txt file
                }
                outfile.close();
                string def2 = "Price\\";
                string pathname2 = def2 + textpricename[counter];
                remove(pathname2.c_str()); // the original price file is removed
                rename("Price\\temp.txt" , pathname2.c_str()); // the temp.txt file is renamed to the original price file name
                clear(); // clear console

        }

        void acceptOrder(){
            string path = "Order\\" + orderfilename[counter]; // store the string of the path of the order file, each restaurant have its own respective order file, which is why this method is used
```

```cpp
            infile.open(path.c_str()); // open the order file of the restaurant
            outfile.open("Order\\temp.txt", fstream::app); // create/open a temp file in the folder Order

            if(!(is_empty(infile))){
                string menuordered; // used to store the name of the menu ordered by customers
                int quantity; // store the quantity of the menu ordered
                char choice;
                double price; // store the price of the menu
                cout << "  Orders" << setw(20) << " "<<"Quantity" << endl;
                //a loop to read the file opened until the end of file
                while(getline(infile,menuordered) && infile >> quantity >> price){
                    cout << " " << endl;
                    cout << "  " << menuordered << setw(30-(menuordered.length())) << " " << quantity << endl; // output the name of the menu ordered ( which is stored in the file order ) and the quantity ordered
                    cout << "  Accept Order? (Y|N) : " ;  // asks the user wheter to accept the order or not


                    while(!(cin >> choice) || (choice != 'Y'  && choice != 'N')){ // input validation
                        cout << "\n  You entered an invalid input. Try again.";
                        cout << "\n  Accept Order? (Y|N) : ";
                        cin.clear();
                        cin.ignore(123,'\n');


                    }
                    if(choice=='Y'){ // if the user input Y , ( accept the order ) , the order will be outputted into the temp.txt file, else, if the user pick N (Decline the order ), the order will not be outputted
                        outfile << menuordered << endl << quantity << endl << price << endl;
                    }

                    infile.ignore();
            }

                infile.close();
                outfile.close();

                remove(path.c_str()); // remove the original order file menu
                rename("Order\\temp.txt", path.c_str()); // rename the temp.txt to the original order file name so that it stores the latest values because accepted orders are stored , meanwhile declined orders are esentially removed
                clear();
            }else{
                cout << " " << endl;
                cout << "  There are no orders.";
                cout << "\n  Press Enter to go back to main menu ...";
                cin.clear();
                cin.ignore();
                if(cin.get()){
                    clear();
                }
            }
        }

        void payment(double arr[][15]){ // a function to show the restaurant managers the payment/price per order made ( each customer has different price per order made )
            if(arr[counter][0] == 0){
                cout << " " << endl;
                cout << "  There are no orders." << endl;
```

```cpp
    }else{
        cout << " " << endl;
        for(int i = 0 ; i < 15 ; i++){
            if(arr[counter][i] != 0){ // priceOrder is a global array that is used to store the total price per order which is calculated in a Customer's function , the statement below will run aslong as the value inside the array is not empty
                cout << "  Order " << i+1 << " : " << fixed << setprecision(2) << arr[counter][i] << "$" << endl;
            }
        }
    }

    cout <<"  Press Enter to go back to the Main Menu ...";
    cin.ignore();
    cin.get();
    clear();

}

void delivery(int arr[][15]){
    if(arr[counter][0] == 0){
        cout << " " << endl;
        cout << "  There are no orders." << endl;
    }else{
        cout << " " << endl;
        cout << "  Estimated Delivery Time : " << endl;
        for(int i = 0; i < 15 ; i++){
            if(arr[counter][i] != 0){
                cout << "  Order " << i+1 << " : " << fixed << setprecision(2) << arr[counter][i] << " minutes" << endl;
            }
        }
    }
    cout << "  Press Enter to go Back to Main Menu ..." ;
    cin.ignore();
    cin.get();
    clear();
}

void totalSale(){ // a function to calculate the total sale of the restaurant
    string menuname;
    double price;
    double sale = 0; // initialise the sale as 0
    double quan; // shortform of quantity to store the quantity of the food ordered
    double totalprice = 0; // initialise the total price as 0, total price is used to store the value of price*quan which is The food price x The Quantity Ordered

    string path = "Order\\" + orderfilename[counter];
    infile.open(path.c_str());


    //loop to read until the end of file
    while(getline(infile,menuname) && infile >> quan >> price){
        infile.ignore();
        totalprice = quan*price; // calculate the total price of an item ordered ( because quantities and price varies for each item ordered )
        sale += totalprice; // sale is the total of the totalprice of each item

    }

    infile.close();
    cout << "                    " << endl;
    cout << fixed << setprecision(2) << "  Total Sale : " << sale << "$";
    cout << "\n  Press Enter to return to Menu ...";
    cin.ignore();
    cin.get();
    clear();


}

void managerChoice(int x, string arr[][15], string arr2[], double arr3[][15], double arr4[][15], int arr5[][15],int arr6[][15]){ //managerChoice is esentially the main() function for the Restaurant Manager because it calls out all the functions above ( functions for restaurant manager )
    clear();

    switch(x)
    {
        case 0 : cout << endl << "  Program Terminated."; break; // if the passed value x which is choice made at managerMenu() is 0, Program Terminated
        case 1 : createMenu(arr,arr3,arr6);break; // if the user chooses to Create Menu, createMenu function is called
        case 2 : updateMenu(arr,arr2,arr3); break; // if the user chooses to update menu, updateMenu function is called
        case 3 : acceptOrder(); break; // if the user chooses to accept Order, acceptOrder function is called
        case 4 : payment(arr4); break; // if the user chooses the payment option, payment function is called
        case 5 : delivery(arr5); break; // if the user chooses delivery option, delivery function is called
        case 6 : totalSale(); break; // if the user chooses Total Sale, totalSale function is called

    }


}
//=========================================================================================================================================//



//=================================================[CUSTOMER FUNCTIONS]================================================================//

void displayMenu(int m ,string arr[],string arr2[][15], double arr3[][15]){ // a function that displays the menu of restaurant
        int l = arr[m-1].length(); // menu flexibility purposes
        int tablel = l*3;
        cout << "                                       " << endl;
        cout << "  " << string(tablel+1,'-') <<endl;
        cout << "  |"<< setw(l) <<""<< arr[m-1] << setw((tablel-1)-1) << "|"<<  endl; // cout the restaurant name at the top of the menu
        cout << "  " << string(tablel+1, '-') << endl;
        for(int i = 0 ; i<15 ; i++){
            int character = arr2[m-1][i].length(); // the length of each menu names are initialised for aliggnment menu flexibility

            if(arr2[m-1][i] != "\0"){
                counter2 = i+1;
                if(i+1 < 10){
                    if(arr3[m-1][i] < 10){
                        cout << "  |" << " " << i+1 << "  - " << arr2[m-1][i] <<right << setw(tablel-character-
15) << " " << fixed << setprecision(2) << arr3[m-1][i] << "$"<< setw(tablel-(6+character+(tablel-character-15)+5)) <<"|" << endl;
                    }else{
                        cout << "  |" << " " << i+1 << "  - " << arr2[m-1][i] <<right << setw(tablel-character-15) << " " << fixed << setprecision(2) << arr3[m-1][i] <<"$"<< setw(tablel-(6+character+(tablel-character-15)+6)) <<"|" << endl;
                    }
                }else{
                    if(arr3[m-1][i] < 10){
                        cout << "  |" << " " << i+1 << " - " << arr2[m-1][i] << right << setw(tablel-character-15) << " " << fixed << setprecision(2) << arr3[m-1][i] <<"$" << setw(tablel-(5+character+(tablel-character-15)+6)) << "|" << endl;
                    }else{
                        cout << "  |" << " " << i+1 << " - " << arr2[m-1][i] <<right << setw(tablel-character-15) << " " << fixed << setprecision(2) << arr3[m-1][i] <<"$"<< setw(tablel-(6+character+(tablel-character-15)+6)) <<"|" << endl;
                    }
                }

            }
        }


        cout << "  " << string(tablel+1,'-') << endl;
        cout << endl;
}

int loginCustomer(){ //function for the customer to login to the system ( the same login menu as restaurant manager loginmenu
    clear();
    string username,password;
    string address;
    string username1,password1;
    int flag = 0;
    char ch;
    cout << "                            " << endl;
    cout << "  -----------------------------------------------" << endl;
    cout << "  |                   LOGIN                    |" << endl;
    cout << "  -----------------------------------------------" << endl;
    cout << "    Username : " ; cin >> username;
    cout << "    Password : " ;

    do{
        ch = getch();

        if(ch == 13 || ch == ' ')
        {
            break;
        }
        if((ch==8 || ch==127) && !password.empty()){
            cout << "\b \b";
            password.erase(password.size()-1);
        }else{
            password+=ch;
            cout << "*";

        }

    }while(ch != 13 || ch != ' ');

    cout << endl << "  -----------------------------------------------" ;

    infile.open("customer.txt");
    while(infile >> username1 >> password1 && getline(infile,address)){
        if(username == username1 && password == password1){
            flag = 1;
            break;
        }
    }
    if(flag == 1){
        cout << "\n                    ";
        cout << "\n  Permission Allowed";
        cout << "\n  Press Enter To Continue...";
        cin.ignore();
        cin.get();

        clear();
    }
    else{
        cout << "\n                    ";
        cout << "\n  Permission Denied";
    }

    return flag;
}

void registerCustomer(){ // function to register a new user to the system
    string username,password;
    string address;
    clear();
    int flag = 0;
    char ch;
    cout << "                            " << endl;
    cout << "  -----------------------------------------------" << endl;
    cout << "  |                  REGISTER                  |" << endl;
    cout << "  -----------------------------------------------" << endl;
    cout << "    Username  : " ; cin >> username; //user input their username
    cin.clear();
    cin.ignore(123,'\n');
    cout << "    Password  : " ;
    //user input password ( the program will display it every character input as *  for privacy )============
    do{
        ch = getch();

        if(ch == 13 || ch == ' ')
        {
            break;
        }
        if((ch==8 || ch==127) && !password.empty()){
            cout << "\b \b";
            password.erase(password.size()-1);
        }else{
            password+=ch;
            cout << "*";

        }

    }while(ch != 13 || ch != ' ');

    //=============================================================================================

    cout << "\n    Address   : ";
    getline(cin,address); //user input their delivery address
    cout << endl << "  -----------------------------------------------" ;

    outfile.open("customer.txt" , fstream::app); // customer.txt is opened to store the login details of all the user, which is why fstream::app is used so that it will not overwrite everything inside
    outfile << username << endl << password; // output the username and password to the file
    outfile <<  endl << address << endl; // output the address to the file
    outfile.close();


    outfile.close();
    cout << "\n  Account Registered Succesfully!" << endl;
    cout << "\n  Press Enter to go to Login Menu ..." ;
    cin.get();
    clear();

}
                    //DELIVERYTIME         //PRICEORDER         //ORDERTIME
void printRecipt(int arr[][15],int x,string arr2[][15],int numofItem[][15],double arr3[][15],int arr4[][15],double arr5[][15],int arr6[][15]){
        clear();
        int delivertimeperitem = 0,estdeliver = 0;
        double price=0,totalprice=0;

        int l = 20;
        int tablel = l*3;
        cout << "                                       " << endl;
        cout << "  " << string(tablel+2,'-') <<endl;
        cout << "  |"<< setw(l+6) <<""<< "RECIPTS" << setw((tablel-l)-12) << "|"<<  endl;
        cout << "  " << string(tablel+2, '-') << endl;

        for(int i = 0 ; i<15 ; i++){
            int character = arr2[x-1][arr[x-1][i]].length(); // for alignment flexibility use
            if(arr[x-1][i] == 0 && numofItem[x-1][i] == 0){
                break;
            }else{


                price = (arr3[x-1][arr[x-1][i]])*(numofItem[x-1][i]); //THE PRICE IS PRICE OF ITEM X THE QUANTITY OF ITEM,
                totalprice += price; // the total price of all the items are the total of the price of the items
                delivertimeperitem = (arr4[x-1][arr[x-1][i]])*(numofItem[x-1][i]);
                estdeliver += delivertimeperitem;
                //alignmment flexibility/problem which is why if and else are used a lot to make sure the menu allign properly with different values of price, etc
                if(arr3[x-1][arr[x-1][i]] < 10){
                        if(price < 10)
                            cout << "  |" << " " << i+1 << "  - " << arr2[x-1][arr[x-1][i]] <<right << setw(tablel-character-27) << " " << fixed << setprecision(2) << arr3[x-1][arr[x-1][i]] << "$ x " << numofItem[x-1][i] << " = "<< price << "$" << setw(tablel-(6+character+(tablel-character-15)+4)) <<"|" << endl;
                        else
                            cout << "  |" << " " << i+1 << "  - " << arr2[x-1][arr[x-1][i]] <<right << setw(tablel-character-27) << " " << fixed << setprecision(2) << arr3[x-1][arr[x-1][i]] << "$ x " << numofItem[x-1][i] << " = "<< price << "$" << setw(tablel-(6+character+(tablel-character-15)+5)) <<"|" << endl;
                }else{
                    cout << "  |" << " " << i+1 << "  - " << arr2[x-1][arr[x-1][i]] <<right << setw(tablel-character-
```

```cpp
28) << " " << fixed << setprecision(2) << arr3[x-1][arr[x-1][i]] << "$ x " << numofItem[x-1][i] << " = "<< price << "$" << setw(tablel-(6+character+(tablel-character-15)+5)) <<"|" << endl;
                        }
                    }

                }
                cout << "  " << string(tablel+2,'-') << endl;
                cout << "  Total Price = " << totalprice << endl; //print out the total price of the orders
                cout << "  Estimated Delivery Time = " << estdeliver << " minutes";

                for(int i =0 ; i < 15 ; i++){
                    if(arr5[x-1][i] == 0){ //priceOrder( arr5 ) is an that stores the payment per order price,since the price per order is only inputted one at a time in a single run, it only allocate the array if the value is 0 ( empty )
                        arr5[x-1][i] = totalprice; //totalprice is stored in the array priceOrder
                        break;
                    }
                }
                outfile.open("totalperorder.txt");
                for(int i = 0 ; i < 10 ; i++){
                    for(int j = 0 ; j < 15 ; j++){
                        outfile << arr5[i][j] << endl; // output the pricerperOrder to a file to be accesed later
                    }
                }
                outfile.close();

                for(int i = 0 ; i< 15 ; i++){
                    if(arr6[x-1][i] == 0){
                        arr6[x-1][i] = estdeliver; // estimated delivery time is stored in the arr6
                        break;
                    }
                }

                outfile.open("delivertimeperorder.txt");
                for(int i = 0; i<10 ; i++){
                    for(int j = 0 ; j<15 ; j++){
                        outfile << arr6[i][j] << endl; // estimated delivery time is output into a file for restaurant manager to access later
                    }
                }
                outfile.close();
                cout << endl;


        }

        void customerMenu(string arr[], string arr2[][15], double arr3[][15], int deliverytime[][15],double priceOrder[][15],int orderTime[][15],int numItem[][15],int numofItem[][15]){

            int choice = 0;
            int j = 1;
            //======table size optimization/for flexibility of menu==========//
                int l = 20;
                int tablel = l*3;
            //================================================================//
                cout << "                                       " << endl;
                cout << "  " << string(tablel+2,'-') <<endl;
                cout << "   |"<< setw(l+2) <<""<< "RESTAURANTS" << setw((tablel-l)-12) << "|"<<  endl;
                cout << "   " << string(tablel+2, '-') << endl;
                for(int i = 0 ; i<10 ; i++){
                    int character = arr[i].length();
                    if(arr[i] != "\0"){ //if the value of array at the index is not empty, execute below statement , the array passed is restaurant[] which contains the restaurant names
                    //if and else is used for allignment purpose because the number 1-9 are 1 digit meanwhile 10 and above are two digit which will have different allignment settings
                        if(i+1 < 10){
                            cout << "   |" << " " << i+1 << "  - " << arr[i]  << setw(tablel-(character+6)) << " " <<"|" << endl;
                        }else{
                            cout << "   |" << " " << i+1 << " - " << arr[i]  << setw(tablel-(character+6)) << " " << "|" << endl;
                        }

                    }else if(arr[i] == "\0"){ //this ensures that the loop stop when the value inside the array is empty, we only want to print out all of the restaurant names, not print out empty lines
                        break;
                    }
                }
                cout << "   " << string(tablel+2,'-') << endl;
                cout << endl;
                cout << "   Choose restaurant : ";
                while(!(cin >> choice) || choice <= 0 || choice > 10){
                    cout << "\n  You have entered an invalid input. Try again.";
                    cout << "\n  Choose restaurant : ";
                    cin.clear();
                    cin.ignore(123,'\n');

                }
                 // user choose the restaurant they want to order from

                clear();
                displayMenu(choice,arr,arr2,arr3); // displayMenu function is called to display the Menu of the restaurant user picked

                int itemchoice;
                int quantity;
                char cont; //shortform of continue
                string path = "Order\\" + orderfilename[choice-1]; // order file of the respective restaurant is opened to store the orders of the customer
                outfile.open(path.c_str(), fstream::app); //fstream:: app is so that the file is not overwritten everytime a user orders, instead, it will append in the file
                int i = 0; //initialize i as 0 to input in the array of numItem and numofItem in the do-while loop
                do{
                    cout << "  Choose an item : ";
                    // the user chooses the item displayed by displayMenu function
                    while(!(cin >> itemchoice) || (itemchoice <= 0 || itemchoice > counter2 ) ){
                    cout << "\n  You have entered an invalid input. Try again.";
                    cout << "\n  Choose an item : ";
                    cin.clear();
                    cin.ignore(123,'\n');
                    }
                    itemchoice = itemchoice - 1;
                    numItem[choice-1][i] = itemchoice; //choice-1 is used because the number displayed in the menu are 1-
```

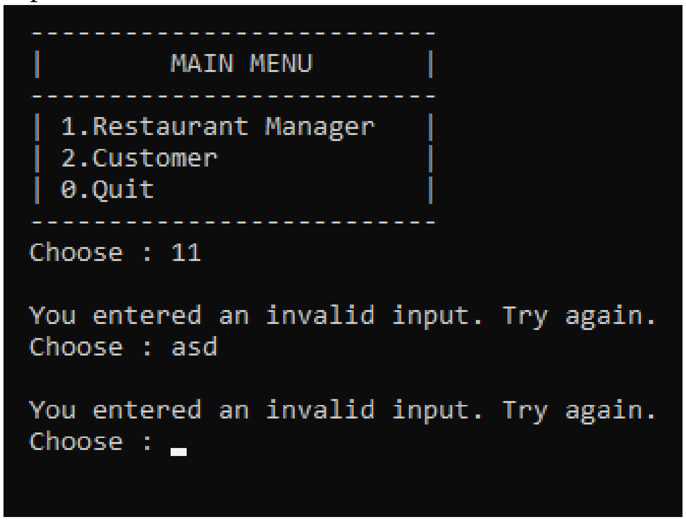
```cpp
10 while an array are counted from 0-9, the choice of item the user picked are stored in the array to be accesed later in another function
                   cout << "  Quantity? : ";
                   //user input the quantity of the item want to be ordered
                   while(!(cin >> quantity)){
                   cout << "\n  You have entered an invalid input. Try again.";
                   cout << "\n  Quantity? : ";
                   cin.clear();
                   cin.ignore(123,'\n');
                   }
                   numofItem[choice-1][i] = quantity; // stored in the array numofItem to be accessed later
                   i++; // i is incremented
                   outfile << arr2[choice-1][itemchoice] << endl << quantity << endl << arr3[choice-1][itemchoice] << endl; // output all of the choices made into the order file
                   cout << "  Continue(Y|N) : "; // asks the user if they want to add more orders or stop
                   // user input Y or N
                   while(!(cin >> cont) || (cont != 'Y' && cont != 'N')){
                   cout << "\n  You have entered an invalid input. Try again.";
                   cout << "\n  Continue(Y|N) : ";
                   cin.clear();
                   cin.ignore(123,'\n');
                   }

               }while(cont == 'Y'); // continue the loop aslong as user picks Y


               outfile.close(); //close file
               printRecipt(numItem,choice,arr2,numofItem,arr3,deliverytime,priceOrder,orderTime); // print the recipt of the order just made

       }

       void customerSign(string arr[], string arr2[][15], double arr3[][15], int arr4[][15], double priceOrder[][15], int orderTime[][15],int numItem[][15],int numofItem[][15]){ // esentially the first menu of customer and also kind of act like the main for the customer functions
           int flag=0;
           int choice;
           cout << "                            " << endl;
           cout << "  --------------------------" << endl;
           cout << "  |      CUSTOMER MENU     |  " << endl;
           cout << "  --------------------------" << endl;
           cout << "  | 1.Existing User        |" << endl;
           cout << "  | 2.New User             |" << endl;
           cout << "  | 0.Quit                 |" << endl;
           cout << "  --------------------------" << endl;

            cout << "\n  Choose : ";
            while(!(cin >> choice) || (choice < 0 || choice > 2)){
               cout << "\n  You entered an invalid input, Try again.";
               cout << "\n  Choose : ";
               cin.clear();
               cin.ignore(123,'\n');
            }

            //a swtich case, if the user an existing user (case 1) the console will prompt up login menu in the LoginCustomer function
            switch(choice)
            {
               case 1 : flag = loginCustomer(); // login menu for customer
                        if(flag == 1){ //if the username and password input by user is correct, the Menus / Restaurant options will be prompted
```

```
                           customerMenu(arr,arr2,arr3,arr4,priceOrder,orderTime,numItem,numofItem); // prompt Restaurant options the user can choose
                         }
                         else
                           break;
                        break;
               case 2 : registerCustomer(); // if the user is a new user, the console will prompt up a register menu first, then store the login detail in customer.txt file so that the user can login later
                        flag = loginCustomer(); // login after account is registered
                        if(flag == 1){ // if the username and password input by user is valid, the menus/restaurant option will be prompted
                           customerMenu(arr,arr2,arr3,arr4,priceOrder,orderTime,numItem,numofItem);
                         }
                         else{
                           break;
                         }
                        break;
               case 0 : clear();cout << " " << endl ; cout << "  Program Terminated" ; break;
            }
       }

       //================================================================================================================================//
```

Input Output of The Program

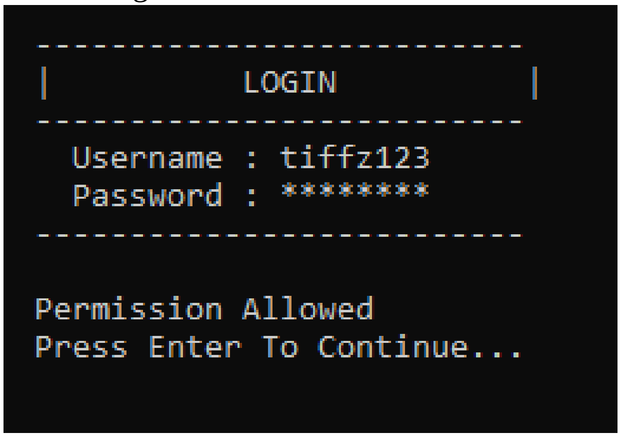
Main Page

1. **User choose Restaurant Manager or Customer**
   a. Input Validation



```
--------------------------
|        MAIN MENU       |
--------------------------
| 1.Restaurant Manager   |
| 2.Customer             |
| 0.Quit                 |
--------------------------
Choose : 11

You entered an invalid input. Try again.
Choose : asd

You entered an invalid input. Try again.
Choose : _
```

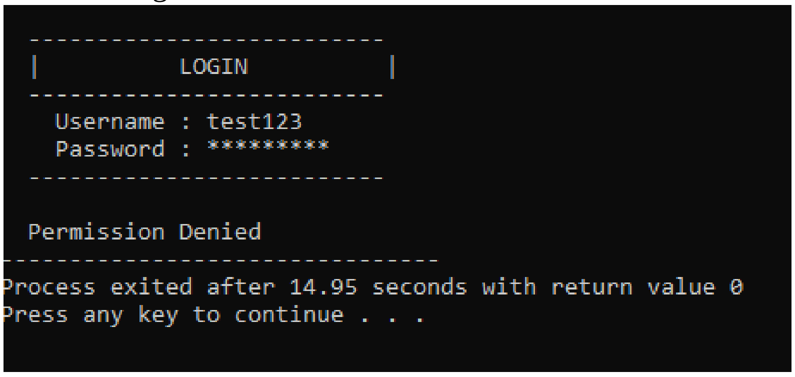
2. **User choose Restaurant Manager**
   a. Correct login details



```
--------------------------
|          LOGIN         |
--------------------------
  Username : tiffz123
  Password : ********
--------------------------

Permission Allowed
Press Enter To Continue...
```

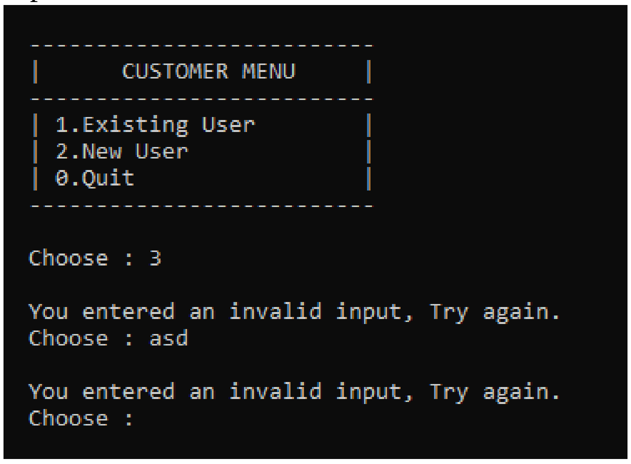
(For Login Details of Restaurant, refer to Logindetails.pdf file included)

   b. Incorrect login details



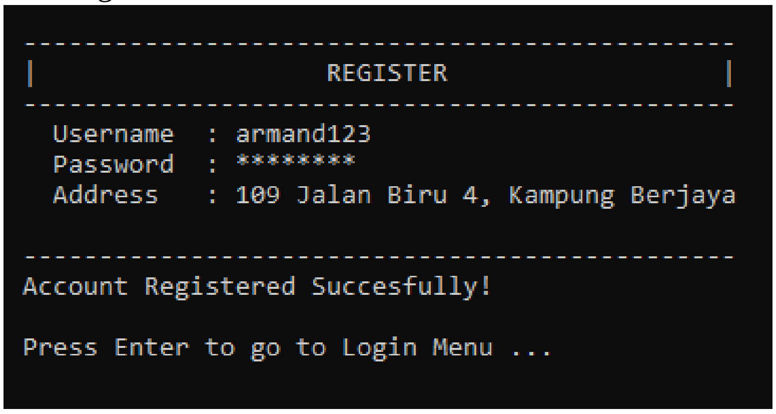
```
--------------------------
|          LOGIN         |
--------------------------
  Username : test123
  Password : *********
--------------------------

 Permission Denied
--------------------------------
Process exited after 14.95 seconds with return value 0
Press any key to continue . . .
```

3. **User Choose Customer**
    a. Input validation



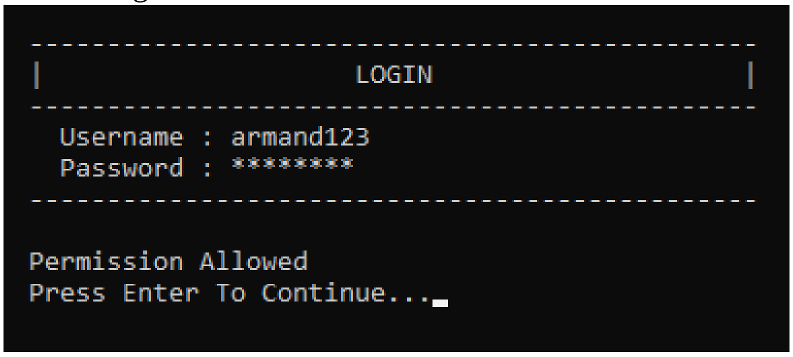
```
-------------------------
|      CUSTOMER MENU    |
-------------------------
| 1.Existing User       |
| 2.New User            |
| 0.Quit                |
-------------------------

Choose : 3

You entered an invalid input, Try again.
Choose : asd

You entered an invalid input, Try again.
Choose :
```

4. **User Choose New User**
    a. User register



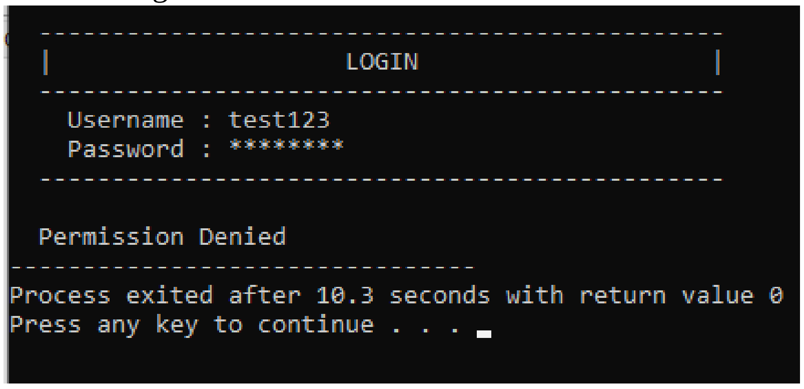
```
-----------------------------------------------
|                  REGISTER                   |
-----------------------------------------------
  Username  : armand123
  Password  : ********
  Address   : 109 Jalan Biru 4, Kampung Berjaya

-----------------------------------------------
Account Registered Succesfully!

Press Enter to go to Login Menu ...
```

5. **User Choose Existing User**
    a. Correct login details



```
-----------------------------------------------
|                   LOGIN                     |
-----------------------------------------------
  Username : armand123
  Password : ********
-----------------------------------------------

Permission Allowed
Press Enter To Continue...
```

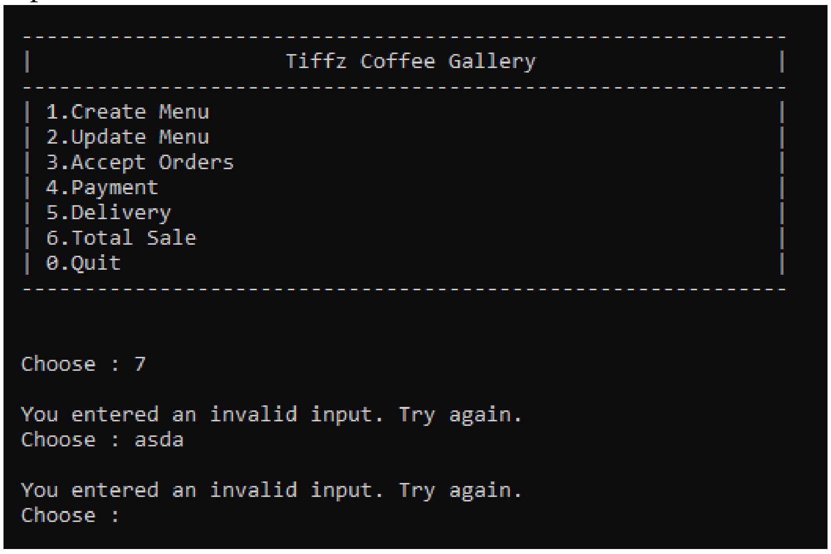
b. Incorrect login details



```
 -----------------------------------------------
 |                    LOGIN                     |
 -----------------------------------------------
   Username : test123
   Password : ********
 -----------------------------------------------

  Permission Denied
--------------------------------
Process exited after 10.3 seconds with return value 0
Press any key to continue . . .
```

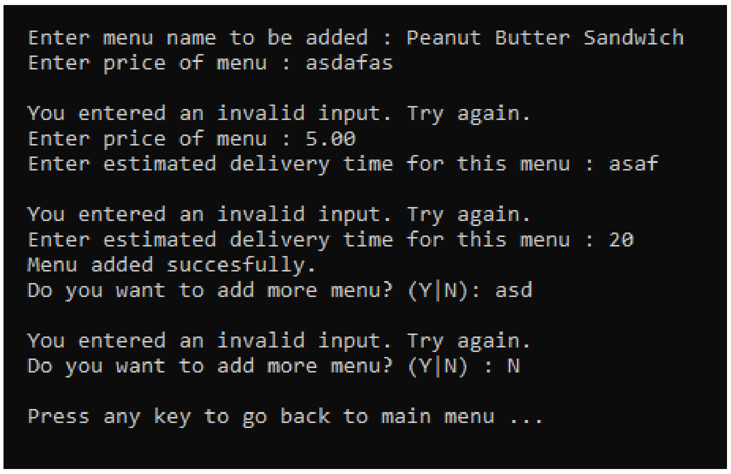
Manager's Input Output

1. **Manager's Menu**
   a. Input Validation



```
-------------------------------------------------------------
|                    Tiffz Coffee Gallery                    |
-------------------------------------------------------------
| 1.Create Menu                                              |
| 2.Update Menu                                              |
| 3.Accept Orders                                            |
| 4.Payment                                                  |
| 5.Delivery                                                 |
| 6.Total Sale                                               |
| 0.Quit                                                     |
-------------------------------------------------------------


Choose : 7

You entered an invalid input. Try again.
Choose : asda

You entered an invalid input. Try again.
Choose :
```

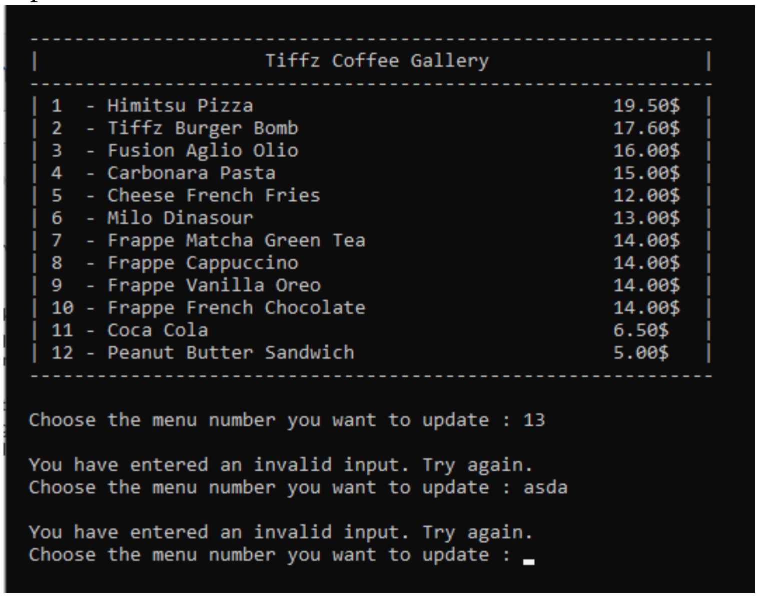
2. **User Choose Create Menu**
   a. Input Validation



```
Enter menu name to be added : Peanut Butter Sandwich
Enter price of menu : asdafas

You entered an invalid input. Try again.
Enter price of menu : 5.00
Enter estimated delivery time for this menu : asaf

You entered an invalid input. Try again.
Enter estimated delivery time for this menu : 20
Menu added succesfully.
Do you want to add more menu? (Y|N): asd

You entered an invalid input. Try again.
Do you want to add more menu? (Y|N) : N

Press any key to go back to main menu ...
```

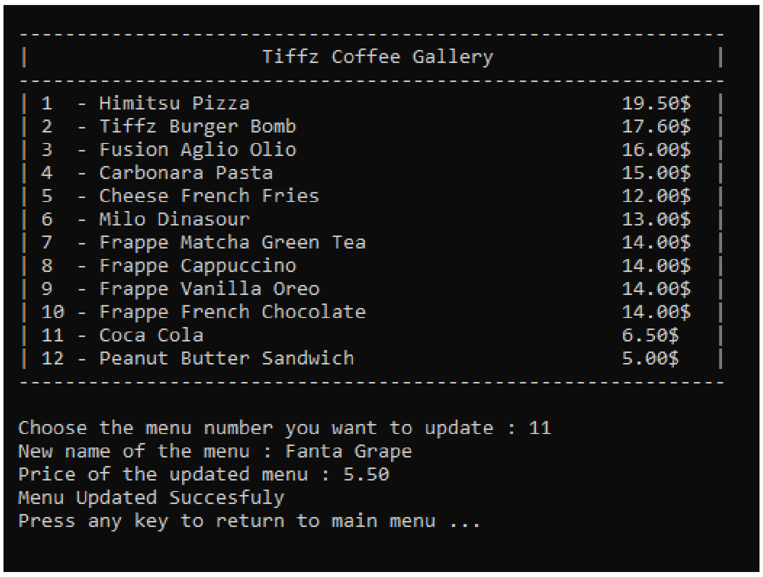
3. User Choose Update Menu

a. Input Validation

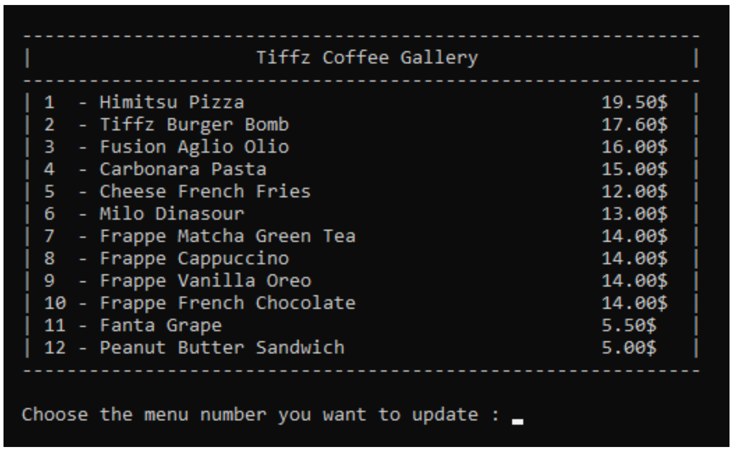


```
-----------------------------------------------------------
|                   Tiffz Coffee Gallery                  |
-----------------------------------------------------------
| 1  - Himitsu Pizza                          19.50$   |
| 2  - Tiffz Burger Bomb                      17.60$   |
| 3  - Fusion Aglio Olio                      16.00$   |
| 4  - Carbonara Pasta                        15.00$   |
| 5  - Cheese French Fries                    12.00$   |
| 6  - Milo Dinasour                          13.00$   |
| 7  - Frappe Matcha Green Tea                14.00$   |
| 8  - Frappe Cappuccino                      14.00$   |
| 9  - Frappe Vanilla Oreo                    14.00$   |
| 10 - Frappe French Chocolate                14.00$   |
| 11 - Coca Cola                              6.50$    |
| 12 - Peanut Butter Sandwich                 5.00$    |
-----------------------------------------------------------

Choose the menu number you want to update : 13

You have entered an invalid input. Try again.
Choose the menu number you want to update : asda

You have entered an invalid input. Try again.
Choose the menu number you want to update : _
```

b. User Update Menu



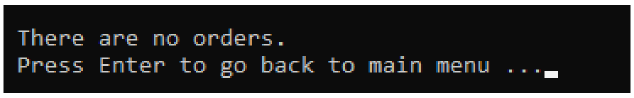
```
------------------------------------------------------------
|                    Tiffz Coffee Gallery                  |
------------------------------------------------------------
| 1  - Himitsu Pizza                              19.50$   |
| 2  - Tiffz Burger Bomb                          17.60$   |
| 3  - Fusion Aglio Olio                          16.00$   |
| 4  - Carbonara Pasta                            15.00$   |
| 5  - Cheese French Fries                        12.00$   |
| 6  - Milo Dinasour                              13.00$   |
| 7  - Frappe Matcha Green Tea                    14.00$   |
| 8  - Frappe Cappuccino                          14.00$   |
| 9  - Frappe Vanilla Oreo                        14.00$   |
| 10 - Frappe French Chocolate                    14.00$   |
| 11 - Coca Cola                                  6.50$    |
| 12 - Peanut Butter Sandwich                     5.00$    |
------------------------------------------------------------

Choose the menu number you want to update : 11
New name of the menu : Fanta Grape
Price of the updated menu : 5.50
Menu Updated Succesfuly
Press any key to return to main menu ...
```



```
------------------------------------------------------------
|                    Tiffz Coffee Gallery                  |
------------------------------------------------------------
| 1  - Himitsu Pizza                              19.50$   |
| 2  - Tiffz Burger Bomb                          17.60$   |
| 3  - Fusion Aglio Olio                          16.00$   |
| 4  - Carbonara Pasta                            15.00$   |
| 5  - Cheese French Fries                        12.00$   |
| 6  - Milo Dinasour                              13.00$   |
| 7  - Frappe Matcha Green Tea                    14.00$   |
| 8  - Frappe Cappuccino                          14.00$   |
| 9  - Frappe Vanilla Oreo                        14.00$   |
| 10 - Frappe French Chocolate                    14.00$   |
| 11 - Fanta Grape                                5.50$    |
| 12 - Peanut Butter Sandwich                     5.00$    |
------------------------------------------------------------

Choose the menu number you want to update : _
```

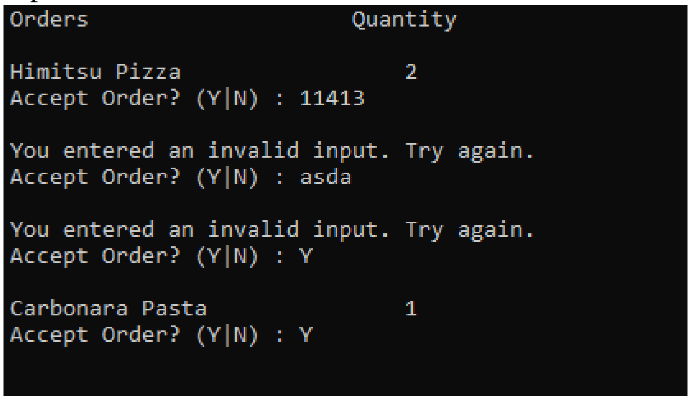
4. User Choose Accept Order

a. No orders from customers



```
There are no orders.
Press Enter to go back to main menu ..._
```

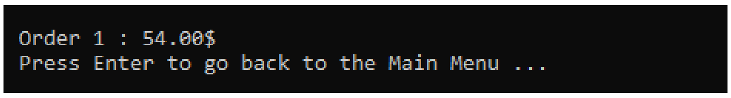
b. Input Validation



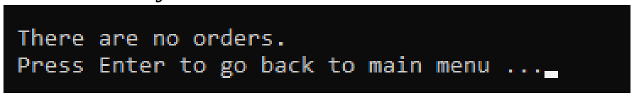
```
Orders                     Quantity

Himitsu Pizza                  2
Accept Order? (Y|N) : 11413

You entered an invalid input. Try again.
Accept Order? (Y|N) : asda

You entered an invalid input. Try again.
Accept Order? (Y|N) : Y

Carbonara Pasta                1
Accept Order? (Y|N) : Y
```

5. User Choose Payment

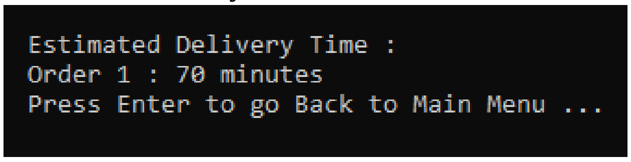
a. Payment made by customers


```
Order 1 : 54.00$
Press Enter to go back to the Main Menu ...
```

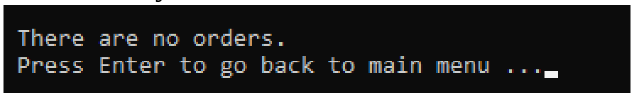
b. No orders by customers


```
There are no orders.
Press Enter to go back to main menu ...
```

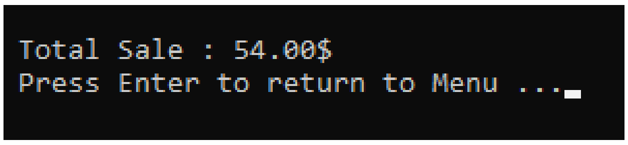
6. **User Choose Delivery**
   a. Estimated delivery time for orders


```
Estimated Delivery Time :
Order 1 : 70 minutes
Press Enter to go Back to Main Menu ...
```

   b. No orders by customers

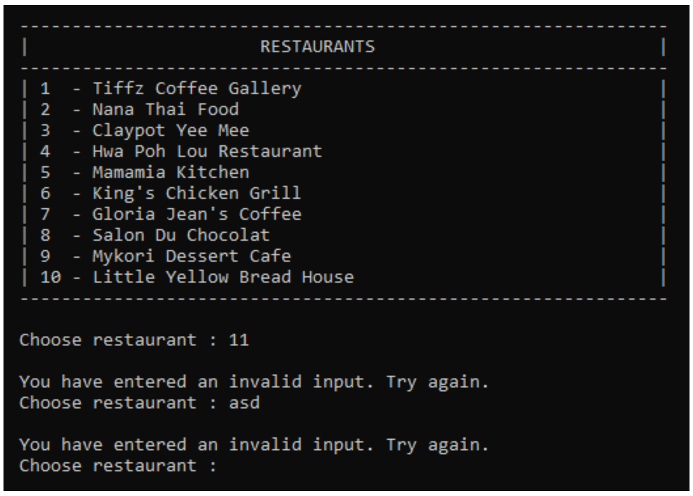

```
There are no orders.
Press Enter to go back to main menu ...
```

7. **User Choose Total Sale**
   a. Total Sale of the restaurant


```
Total Sale : 54.00$
Press Enter to return to Menu ...
```

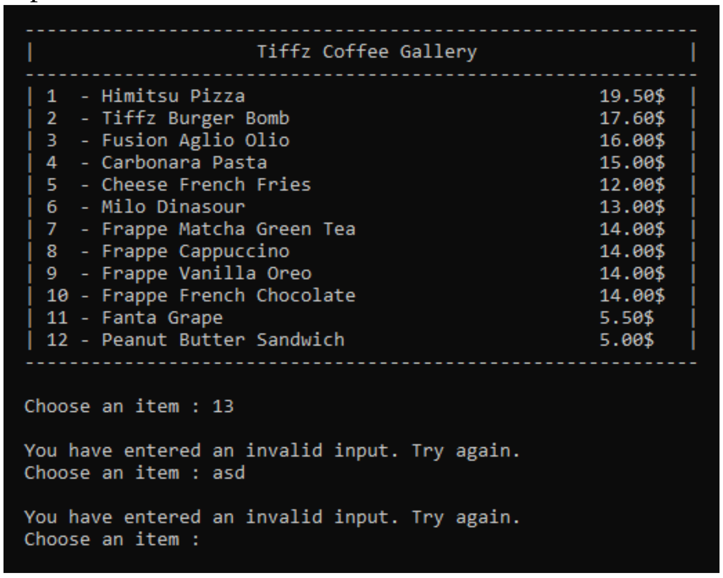
<u>Customer's Input Output</u>

1. **User Choose Restaurant**
   a. Total Sale of the restaurant


```
-------------------------------------------------------------
|                       RESTAURANTS                         |
-------------------------------------------------------------
| 1  - Tiffz Coffee Gallery                                 |
| 2  - Nana Thai Food                                       |
| 3  - Claypot Yee Mee                                      |
| 4  - Hwa Poh Lou Restaurant                               |
| 5  - Mamamia Kitchen                                      |
| 6  - King's Chicken Grill                                 |
| 7  - Gloria Jean's Coffee                                 |
| 8  - Salon Du Chocolat                                    |
| 9  - Mykori Dessert Cafe                                  |
| 10 - Little Yellow Bread House                            |
-------------------------------------------------------------

Choose restaurant : 11

You have entered an invalid input. Try again.
Choose restaurant : asd

You have entered an invalid input. Try again.
Choose restaurant :
```

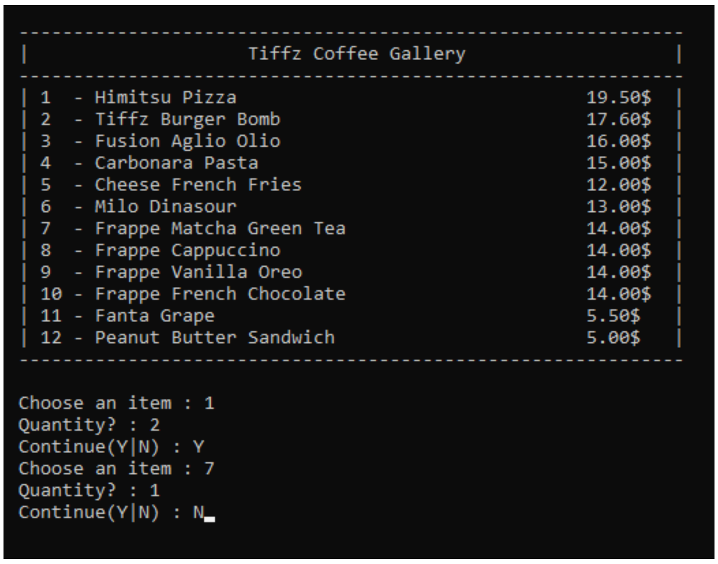
2. User Choose Menu

a. Input Validation



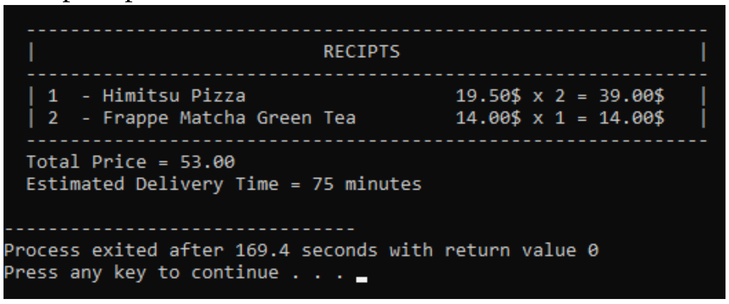
```
---------------------------------------------------------------
|                    Tiffz Coffee Gallery                      |
---------------------------------------------------------------
| 1  - Himitsu Pizza                                19.50$   |
| 2  - Tiffz Burger Bomb                            17.60$   |
| 3  - Fusion Aglio Olio                            16.00$   |
| 4  - Carbonara Pasta                              15.00$   |
| 5  - Cheese French Fries                          12.00$   |
| 6  - Milo Dinasour                                13.00$   |
| 7  - Frappe Matcha Green Tea                      14.00$   |
| 8  - Frappe Cappuccino                            14.00$   |
| 9  - Frappe Vanilla Oreo                          14.00$   |
| 10 - Frappe French Chocolate                      14.00$   |
| 11 - Fanta Grape                                  5.50$    |
| 12 - Peanut Butter Sandwich                       5.00$    |
---------------------------------------------------------------

Choose an item : 13

You have entered an invalid input. Try again.
Choose an item : asd

You have entered an invalid input. Try again.
Choose an item :
```

b. User orders food



```
---------------------------------------------------------------
|                    Tiffz Coffee Gallery                      |
---------------------------------------------------------------
| 1  - Himitsu Pizza                                19.50$   |
| 2  - Tiffz Burger Bomb                            17.60$   |
| 3  - Fusion Aglio Olio                            16.00$   |
| 4  - Carbonara Pasta                              15.00$   |
| 5  - Cheese French Fries                          12.00$   |
| 6  - Milo Dinasour                                13.00$   |
| 7  - Frappe Matcha Green Tea                      14.00$   |
| 8  - Frappe Cappuccino                            14.00$   |
| 9  - Frappe Vanilla Oreo                          14.00$   |
| 10 - Frappe French Chocolate                      14.00$   |
| 11 - Fanta Grape                                  5.50$    |
| 12 - Peanut Butter Sandwich                       5.00$    |
---------------------------------------------------------------

Choose an item : 1
Quantity? : 2
Continue(Y|N) : Y
Choose an item : 7
Quantity? : 1
Continue(Y|N) : N
```

c. Recipt is printed



```
---------------------------------------------------------------
|                          RECIPTS                             |
---------------------------------------------------------------
| 1  - Himitsu Pizza                  19.50$ x 2 = 39.00$    |
| 2  - Frappe Matcha Green Tea        14.00$ x 1 = 14.00$    |
---------------------------------------------------------------
Total Price = 53.00
Estimated Delivery Time = 75 minutes

--------------------------------
Process exited after 169.4 seconds with return value 0
Press any key to continue . . .
```