



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**SCSJ2013-06 Data Structure And Algorithm**

---

**2020/2021-1**

## **Assignment 2 Report**

### **Group members**

<b>Omar hamed abdellatif Ibrahim</b>	<b>A18CS4061</b>
<b>Alaa Alrhman Mohammed Raweh AL-Shaibani</b>	<b>A18CS4037</b>
<b>Habiba Elgammal</b>	<b>A18CS0303</b>

## **Contents:**

1. Introduction	1
2. Functionality	1
3. Objective	4
4. Use Case	5
5. Class Diagram	6
6. Appendices	7
6.1. Data Files	7
6.2. Source Code	8

# 1. Introduction:

Nowadays during the covid-19, pandemic there is no argument that the online ordering is one of the most important and useful services that have been increasingly used especially in the field of ordering food as each restaurant has only a limited number of customers to be dine in. If you need to get a meal, you have to go through a time-consuming process especially when you go to a place without having reservation or the restaurant cannot serve all your family members in one table. Facing such difficulties made online food ordering a good choice for customers to get their favorite meals. As a result of that we choose to develop a food ordering system as our assignment objective.

The developed food ordering system, will provide the customer with the option of selecting the meal by giving him the chance to order meal by showing him the ordered list that contains all options. It also will give the staff the ability to update the menu by adding new meals and put them in the right alphabetical order as well as the ability of searching for food item and other functionalities that will be furthered discussed.

# 2. Functionality:

The user will be provided 2 options as the picture shows:

```
Welcome to Food Ordering System
1. Im Am a Customer
2. Im a Staff.
```

If the user didn't pick either customer or staff then the loop will end.

```
    }
} while (SoIType != (1) && SoIType != (2));
int choice_I = 0;
int choice_S = 0;
```

In case the user was a customer, the customer menu will be provided to him to choose an action to proceed.

```

int menu_Customer()
{
    //dashboard for the staff it can display foods available ,add foods ,delete f
oods,search foods
    int men_choice;
    cout << "Welcome to Customer Dashboard " << endl;
    cout << "1.Show list of Foods" << endl;
    cout << "2.Order Food" << endl;
    cout << "3.Exit" << endl;
    cout << "Please enter your choice from 1..3" << endl;
    cin >> men_choice;
}

```

If the customer chose to show list of food, a list of available food will be shown. The list is ordered alphabetically, by implementing Insertion Sorting Algorithm that is done using InsertionSortFood(). The food will be displayed in order.

```

Please enter your choice from 1..3
1
FoodID Name details
3 Double big mac
1 hotdog roll
2 Nutella crepe
4 SuperSuprem Pizza cheese,large,combo

```

After he has viewed the food list, the customer will be required to choose an option from the customer's main menu. Let's say he choose the second option which is order food. Then will be asked if he is new customer or not. If yes he will be required to enter his information which are id , name, phone no., and address. If he is an existing customer he will be required to enter his id .the system will try to search his id if it was found then he will be able to directly choose the food.but if his id doesn't not exist in the database then a message will be sent saying that the system has failed to find his id.

```
Welcome to Customer Dashboard
1.Show list of Foods
2.Order Food
3.go back to the pervious menu
4.Exit
Please enter your choice from 1..3
```

```
}
cout << "Please Enter The Customer ID: ";
cin >> key_j;
cout << endl;
cout << "Search Results" << endl;
if (result_j == -1)
{
    cout << "Customer Not Found." << endl;
}
```

Going back to the first main menu the provide the user with the using mood so he can be a customer or a staff. when choosing the second option which is to be a staff, the new menu will be executed. this menu contains the actions can be done by staff

```
int menu_Staff()
{
    //dashboard for the staff it can display foods available ,add foods ,delete f
oods,search foods
    int men_choice;
    cout << "Welcome to Staff Managment System " << endl;
    cout << "1.Show list of customers" << endl;
    cout << "2.Add Food Item" << endl;
    cout << "3.Delete a Food Item" << endl;
    cout << "4.Search for a food Item" << endl;
    cout << "5.Exit" << endl;
    cin >> men_choice;
    cout << endl;

    switch (men_choice)
```

AS the picture shows , the user as a staff will have different features available for them.

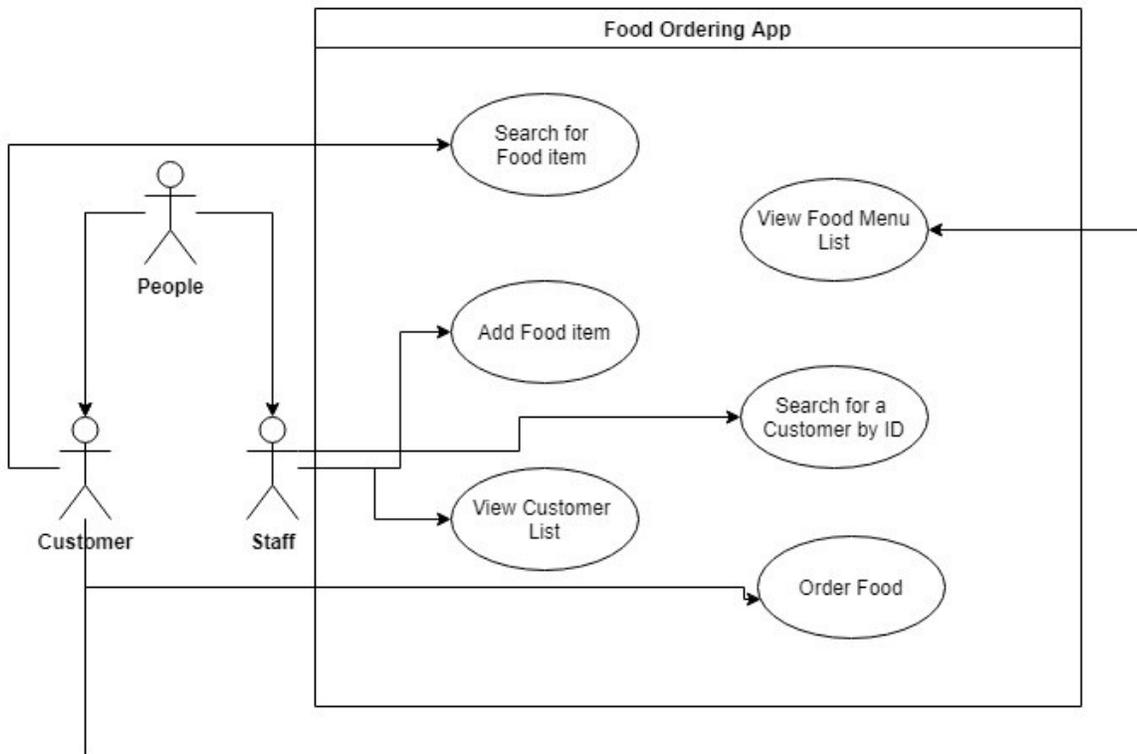
They will have the ability to add or delete from the restaurant's list according to the availability of the menu .

If the staff choses an invalid number from the provided action list, an error message will be sent on screen directing the staff to choose a number from the provided list.

### **3. Objective of The Assignment:**

- To develop a system that will surely satisfy the customer service.
- To design a system able to accommodate huge amount of orders at a time.
- To evaluate its performance and acceptability in terms of security, user-friendliness, accuracy and reliability.
- To improve the communication between the client and the server and minimize the time of ordering.
- One of the main objectives of a restaurant to ensure customer satisfaction. Manual listing of orders by the waiters/waitresses may result to slow response in customer service. Hence, if the restaurant uses the proposed system, manipulation of orders to the customers be so easy and quick

## 4. System's Use Case:



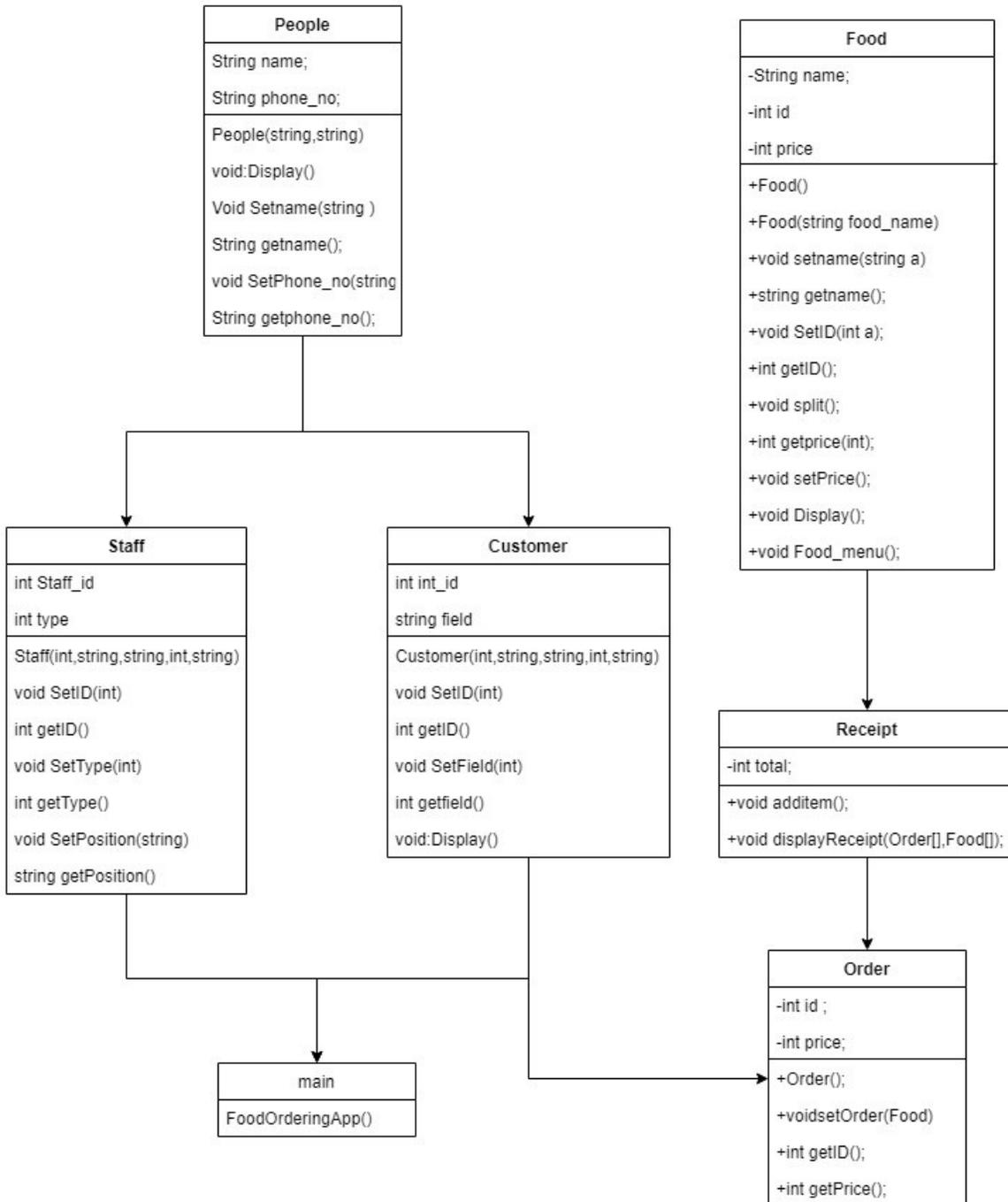
### Description:

In our food ordering system, there are multiple of functions available for both the customers and the staff.

Both of them will be able to search for food items. But only the staff has the eligibility to add or delete food items as well as seeing the whole list of customers registered with the restaurant.

While the customers will be able to search for food and make their preferred order which is limited to only the registered customers.

## 5. Class Diagram:



## 4. Appendices:

### 4.1 Data files:

#### Customer-

4	Omar	22	01123872357	Carls jr	Mutiara Mas 81300
2	Ali	21	01126733245	Macdonalds	Danga bay 81200
3	Alaa	20	01102654865	Pizza hut	Melana apartment 81300
1	Habiba	26	01111892463	Subway	Setia Tropika 81500

#### Food-

3	Double big mac	beef,large,combo
4	SuperSuprem Pizza	cheese,large,combo
1	hotdog roll	spicy,large,combo
2	Nutella crepe	nuts,larger,combo

### 4.2. Source Code:

## Customer.cpp class

```
#include <iostream>
#include "Customer.hpp"

using namespace std;
Customer::Customer() : name_f("") ,name(""),id(0),phoneNum(0) {}
Customer::Customer( string f) : name_f(f) {}
void Customer::SetID(int a) { id = a; } //class member function to set value
of the id variable
int Customer::getID() { return id; } //class member function to get value
of the id variable
long Customer::getPhoneNum() { return phoneNum; } //class member function to set
value of the field variable
void Customer::setPhoneNum(long a) { phoneNum=a; } //class member function to
get value of the field variable
// string getFood_ordered() { return Food_ordered; } //class member function to g
et value of the food_ordered variable
void Customer::Setname(string a){name=a;}
void Customer::Display()
{
    cout << name << endl;
}
void Customer::split()
{
    id =stoi(name.substr(0,1));
}
}
```

## Customer.hpp

```
using namespace std;

class Customer
{ //class Customer is a child class of class People

private:
    string name_f;
    int id;
    string name;
    long phoneNum;

public:
    int cus_num=0;
    Customer();
    Customer(string);
    void SetID(int a);           //class member function to set value of the id variable
    int getID();                //class member function to get value of the id variable
    void Setname(string a); //class member function to set value of the field variable
    string getname();
    long getPhoneNum();
    void setPhoneNum(long );

    //class member function to get value of the field variable
    // string getFood_ordered() { return Food_ordered; } //class member function
to get value of the food_ordered variable
    void split();
    void Display();
    // int menu_Customer();
};
```

## Food.cpp class

```
#include <iostream>
#include "Food.hpp"
using namespace std;

Food::Food() {
    id=0;
    food_name="";
    price=0;
    name="";

} //default Constructor for class food

Food::Food(string f) : food_name(f) {}

void Food::setname(string a) { food_name = a; } //class member function to
to set value of the name variable
string Food::getname() { return food_name; } //class member function to
to get value of the name variable
// void Food::Setrequired(string a) { required = a; } //class member function to
set value of the required variable
// string Food::getrequired() { return required; } //class member function to
get value of the required variable
void Food::SetID(int a) { id = a; } //class member function to set
value of the id variable
int Food::getID() { return id; }
void Food::Display() { cout << food_name << endl; } //class member function to
get value of the id variable
void Food::split(){
id =stoi(food_name.substr(0,1));
price = stoi(food_name.substr(food_name.length()-2,food_name.length()-1));
// name=food_name.substr(3,food_name.length()-4);

}
int Food::getprice(){return price;}
void Food::setPrice(int p){price=p;}
// void Food:Food_men(){
//     int food_arr[food_num];

// }
```

## Food.hpp

```
#ifndef Food_h
#define Food_h
#include <iostream>

using namespace std;

class Food
{
private:
    string name;
    int id;
    int price;
    string food_name;

public:
    int food_num=0;
    Food(); //default Constructor for class food

    Food(string food_name);

    void setname(string a); //class member function to set value of the name
variable
    string getname(); //class member function to get value of the name
variable
    void Setrequired(string a); //class member function to set value of the requi
red variable
    string getrequired(); //class member function to get value of the requi
red variable
    void SetID(int a); //class member function to set value of the id va
riable
    int getID();
    void split();
    int getprice();
    void setPrice(int);
    void Display(); //class member function to get value of the id variable
    void Food_men();
};
#endif
```

## Functions.cpp

```
#include "funtions.hpp"
#include <fstream>
#include "Food.hpp"
#include<iostream>
using namespace std;

int menus(){
    int SoIType;
    cout << "Welcome to Food Ordering System " << endl;
    cout << "1. Im Am a Customer" << endl;
    cout << "2. Im a Staff." << endl;
    cout<<"Enter your choice :";
    cin >> SoIType; // enter the type of roles

    switch (SoIType)
    {
    case 1:
    cout<<endl<<endl<<endl;
        cout << "Welcome to Customer Dashboard " << endl;
        cout << "1.Show list of Foods" << endl;
        cout << "2.Order Food" << endl;
        cout << "3.Exit" << endl;
        cout << "Please enter your choice from 1..3" << endl;
        return 1;
        break;
    case 2 :
    cout << "Welcome to Staff Managment System " << endl;
        cout << "1.Show list of Customers" << endl;
        cout << "2.Add Food Item" << endl;

        cout << "3.Search for a food Item" << endl;
        cout << "4.Exit" << endl;
        return 2;
    break;
    default:
    cout<<"Sorry your choice is invalid ^-^.."<<endl;
        system("pause");
    }
    return menus();
}
```

```

}

void InsertionSortFood(int a[], int n)
{
    int i, j, min, temp;
    for (i = 0; i < n - 1; i++)
    {
        min = i;
        for (j = i + 1; j < n; j++)
            if (a[j] < a[min])
                min = j;
        temp = a[i];
        a[i] = a[min];
        a[min] = temp;
    }
}

```

## Functions.hpp

```

int menus();
void InsertionSortFood(int a[], int n);

```

## Order.cpp

```

#include "order.hpp"
#include "Food.hpp"
order::order() : id(0) ,price(0) {}
void order ::setOrder(Food f){
id=f.getID();
price=f.getprice();
}
int order::getId(){return id;}
int order::getPrice(){return price;}

```

## Order.hpp

```
#ifndef PEOPLE_H
#define PEOPLE_H
#include "Food.hpp"
class order{
    private:

    int id , price;
    public :
    int num_order=0;
    order();
    void setOrder(Food);
    int getId();
    int getPrice();
};
#endif
```

## People.hpp

```
#ifndef PEOPLE_H
#define PEOPLE_H
#include <iostream>

using namespace std;

class People
{
protected:
    string name;

public:
    People(string n);

    void Setname(string a);
    string getname();
};
#endif
```

## Receipt.cpp

```
#include <iostream>
#include "order.hpp"
#include "recept.hpp"
#include "Food.hpp"
using namespace std;
void receipt::displayReceit(order o[], Food f[]){
    cout<<endl<<endl;
    cout<<" here are the orders : "<<endl;
    cout << "FoodID"
        << " "
        << "\t"
        << "Name"
        << " "
        << "\t\t\t"
        << "Price" << endl;
    for(int i=0;i<o->num_order;i++){
        for (int j = 0; j < f->food_num; j++)
        {
            if (f[j].getID() == o[i].getId())
            {
                f[j].Display();
                total+=o[i].getPrice();
            }
        }
    }

    cout<<endl <<endl;
    cout<<"the total is : "<<total<<endl;
}
```

## Reciept.hpp

```
#include <iostream>
#include "order.hpp"
#include "Food.hpp"
using namespace std;
class receipt
{
private:
    int total = 0;

public:
    void addItem();
    void displayReceipt(order[], Food[]);
};
```

## Staff.cpp

```
#include "Staff.hpp"
#include<iostream>
using namespace std;
Staff::Staff(){ }
Staff::Staff(int i, string ns, string pno, int t, string e) : Staff_id(i), type(
t), position(e) { }
void Staff::SetID(int a) { Staff_id = a; } //class member function to se
t value of the staff id variable
int Staff::getID() { return Staff_id; } //class member function to ge
t value of the staff id variable
void Staff::SetType(int a) { type = a; } //class member function to se
t value of the type variable
int Staff::getType() { return type; } //class member function to ge
t value of the type variable
void Staff::SetPosition(string a) { position = a; } //class member function to se
t value of the position variable
string Staff::getPosition() { return position; } //class member function to ge
t value of the position variable
void Staff:: showCustomers(){ }
void Staff:: showMenue(){ }
void Staff:: addfood(){ }
```

## Staff.hpp

```
using namespace std;
#include<iostream>
class Staff
{ //class Staff is a child class of class People
private:
    int Staff_id, type;
    string position;

public:
Staff();
    Staff(int i, string ns, string pno, int t, string e);
    void SetID(int a);           //class member function to set value of the staff
id variable
    int getID();                //class member function to get value of the staff
id variable
    void SetType(int a);        //class member function to set value of the type
variable
    int getType();              //class member function to get value of the type
variable
    void SetPosition(string a); //class member function to set value of the posit
ion variable
    string getPosition();       //class member function to get value of the posit
ion variable
    void showCustomers();
    void showMenue();
    void addfood();
};
```

## Main.cpp()

```
#include <iostream>
#include "funtions.hpp"
#include "Food.hpp"
#include "receit.hpp"
#include "Customer.hpp"
```

```
#include "order.hpp"
#include <fstream>
#include <cstring>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
bool check = true;

struct Node
{
    char foodname[40];
    int foodId;
    int price;
    Node *next;
};

typedef struct Node node;

node *head = NULL;
node *list,*list2;

void insert(int foodId, char foodname[], int price)
{
    node *temp;

    temp = (node *)malloc(sizeof(node));

    temp->foodId = foodId;

    temp->price = price;

    strcpy(temp->foodname, foodname);

    temp->next = head;

    head = temp;

    list = head;
}

void deletefood(int id)
{
```

```

Node *ptr = NULL;
Node *prev = NULL;
Node *current = NULL;
prev = head;
current = head;
while (current != NULL && current->foodId != id)
{
    prev = current;
    current = current->next;
    if (current == NULL)
    {
        cout << "\n id is not valid...\n"
            << endl;
        return;
    }
    prev->next = current->next;
    current->next = NULL;
    delete current;
    cout << endl
        << "\nOrder Deleted\n";
}
}

```

```

void updatefood()
{
    node *ptr;
    node *prev = NULL;
    node *current = NULL;
    int id3;
    cout << "Enter food to modify:";
    cin >> id3;
    prev = head;
    current = head;
    while (current != NULL && current->foodId != id3)
    {
        prev = current;
        current = current->next;
    }
    if (current == NULL)
    {
        cout << "\nfood id is not valid...\n"
            << endl;
        return;
    }
    cout << "\nWhat you want to update:"

```

```

        << "\n";
    cout << "1. Food id"
        << "\n";
    cout << "2. Name"
        << "\n";
    cout << "3. Price"
        << "\n";

    cout << "Enter your choice"
        << "\n";
    int temp_choice;
    cin >> temp_choice;

    switch (temp_choice)
    {
    case 1:
        cout << "\nEnter new food code: ";
        cin >> current->foodId;
        break;
    case 2:
        cout << "\nEnter new name: ";
        cin >> current->foodname;
        break;
    case 3:
        cout << "\nEnter new Price: ";
        cin >> current->price;
        break;

    default:
        cout << "\n"
            << "Choice not available";
        break;
    }
    cout << endl
        << "\nRecord has been updated successfully\n";
    cout << endl;
    cout << endl;
}
int Iempty()
{

    node *temp;

    temp = list;

```

```

int countitem = 0;

if (temp == NULL)
{
    countitem = 0;
}
else
{
    countitem = 1;
    while (temp->next != NULL)
    {
        countitem++;
        temp = temp->next;
    }
}

return countitem;
}
void foodlist()
{
    struct Node *ptr;
    ptr = head;
    int items = Iseempty();
    if (items == 0)
    {
        cout << "the list is empty" << endl;
    }
    else
    {
        cout << "FoodID"
            << " "
            << "\t"
            << "Name"
            << " "
            << "\t\t\t"
            << "Price" << endl;
        while (ptr != NULL)
        {
            cout << ptr->foodId << " ";
            cout << "\t" << ptr->foodname << " ";
            cout << "\t\t" << ptr->price << endl;
            ptr = ptr->next;
        }
        cout << endl;
        cout << endl;
    }
}

```

```

    }
}

int main()
{
    Customer customer[100];
    receipt r;
    insert(1, "doubleCheeseBurger",15);
    insert(2, "CheeseBurger",12);
    insert(3, "DoubleBigMac",16);
    insert(4, "NutellaCrepe",14);
    insert(5, "Coke",3);

    ifstream Cus1("food data.txt");

    int menu = 1;

    do
    {
        mainmenu:
        int SoIType;
        cout << "Welcome to Food Ordering System " << endl;
        cout << "1. Im Am a Customer" << endl;
        cout << "2. Im a Staff." << endl;
        cout << "Enter your choice :";
        cin >> SoIType;
        cout << endl
            << endl;

        switch (SoIType)
        {
            int choice;
            case (1):
            {
                int op = 1;
                while (op == 1)
                {
                    cout << endl
                        << endl
                        << endl;
                    cout << "Welcome to Customer Dashboard " << endl;
                    cout << "1.Show list of Foods" << endl;
                    cout << "2.Order Food" << endl;
                }
            }
        }
    }
}

```

```

cout << "3.go back to the pervious menu " << endl;
cout << "4.Exit" << endl;
cout << "Please enter your choice from 1..3" << endl;
cin >> choice;
switch (choice)
{
case (1):
    foodlist();
    cout << endl
        << endl;
    break;
case (2):{
    cout << "Are you a new Customer or an excisting Customer ?" <
< endl;

    cout << "For new Customer type (1)"
        << endl
        << "For excisting Customer type (2)" << endl;
    int gg;
    cin >> gg;
    int total=0;
    if (gg == 1)
    {
        int id;
        string n2;
        long phone_n;
        cout << "Enter your ID: ";
        cin >> id;
        cout << "Enter your Name: ";
        cin >> n2;
        cout << "Enter your Phone Number: ";
        cin >> phone_n;
        ofstream Cus1("Customers@data.txt", ios::app);
        Cus1 << id << " " << n2 << " " << phone_n << endl;
1;

        cout << endl
            << endl;
    }
    else if (gg == 2)
    {
        cout << "Enter your id: ";
        int id;
        cin >> id;
    }
    addOrder:

```

```

Node *temp;
temp = new Node;

cout << "Enter foodid: ";
cin >> temp->foodId;
cout << "Enter name: ";
cin >> temp->foodname;
cout << "Enter price: ";
cin >> temp->price;
total+=temp->price;
temp->next = NULL;

if (check)
{
    head = temp;
    list2 = temp;
    check = false;
}
else
{
    list2->next = temp;
    list2 = temp;
}
cout << "\n item with id " << temp-
>foodId << " added to the order " << "\n";
cout << "do you still more food .. if yes enter 1 else please
enter 0 : ";

int add;
cin >> add ;
if (add==1){
    goto addOrder;
}

else{ cout<<"the total price for this order is : "<<total<<en
d1;}

}
break;
case (3):
    goto mainmenu;
    break;
default:
    op = 0;
    break;
}

```

```

    }
}
break;
case (2):
{
    int opt = 1;
    while (opt == 1)
    {
        cout
            << "Welcome to Staff Managment System "
            << endl;

        cout << "1.Show list of food" << endl;
        cout << "2.Add Food Item" << endl;
        cout << "3.delete food Item" << endl;
        cout << "4.update food Item" << endl;
        cout << "5.go back to the pervious menu " << endl;
        cout << "6.Exit" << endl;
        cout << "Please enter your choice from 1..5" << endl;
        cin >> choice;

        switch (choice)
        {
            case (1):
                foodlist();

                break;
            case (2):
                char ffoodname[40];
                int fid;
                int fprice;
                int fposi;
                cout << "Enter food id :";
                cin >> fid;
                cout << endl
                    << "Enter food name : " << endl;
                cin >> ffoodname;
                cout << "Enter the price : " << endl;
                cin >> fprice;

                insert(fid, ffoodname, fprice);

                break;
            case (3):
                int id;

```

```
        cout << "Enter food id of the food to be deleted : ";
        cin >> id;
        deletefood(id);

        break;
    case (4):
        updatefood();
        break;
    case (5):
        goto mainmenu;
        break;
    default:
        opt = 0;
        break;
    }
}
break;

default:
    menu = 0;
    cout << "Sorry your choice is invalid ^-^.." << endl;
    cout << endl
        << endl;
    break;
}
} while (menu == 0);

system("pause");
}
```