

**School of Computing**

**Faculty of Engineering**

**UNIVERSITI TEKNOLOGI MALAYSIA**

DATA STRUCTURE & ALGORITHM

**(SECJ2013)**

SEMESTER 1 2020/2021

Mini PROJECT Documentation

**RESTAURANT MANAGEMENT SYSTEM**

**By**

**Narendran A/L Rammudo (A19EC0113)**

**Kishen A/L Prekash (A19EC0063)**

**Thanneermalai A/L Udayappan (A19EC0171)**

**SECTION 03**

**Lecturer**

Ms. Lizawati binti Mi Yusuf

**Date**

30 January 2021

**Part 1: INTRODUCTION**

**1.1 Synopsis Project:**

**This project describes the restaurant management system to help the restaurant manager to manage the restaurant more effectively and efficiently by computerizing meal ordering, billing and inventory control. This system is designed for Fast Food Restaurant. Taking orders, searching orders, sort orders and also displaying orders will be done using linked list (Queue).**

**The restaurant management system is able to create new orders, search for orders, display orders, and also sort the orders in ascending order.**

* 1. **Objective of The Project**

1. **To develop a system that will surely satisfy the customer service.**

1. **To design a system able to accommodate huge amounts of orders at a time.**

1. **To automatically compute the bill.**

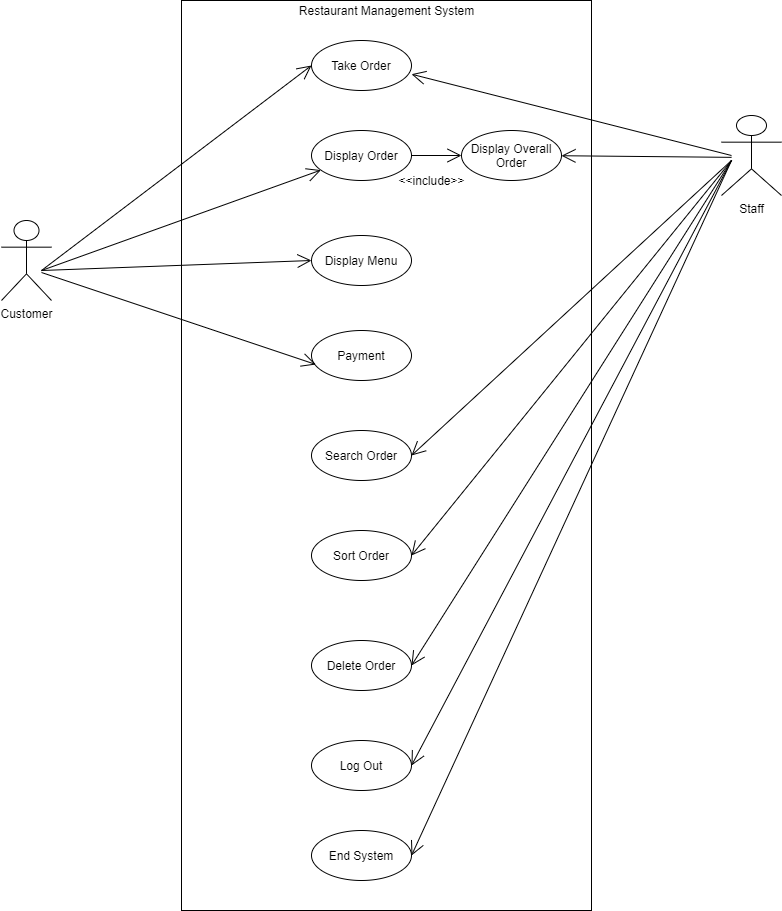
1. **To evaluate its performance and acceptability in terms of security, user-friendliness, accuracy and reliability.**

1. **To improve the communication between the client and the server and minimize the time of ordering.**

**PART 2: SYSTEM ANALYSIS AND DESIGN (USE CASE, FLOWCHART AND CLASS DIAGRAM)**

**2.1 System Requirements**

Use case Diagram



**Figure 1: Use Case Diagram for Restaurant Management System**

**Use Case Description for Restaurant Management System**

The system users are Customer and Staff.

|  |  |
| --- | --- |
| Actor | Task |
| Customer | * Take Order * Display Menu * Payment |
| Staff | * Search Order * Sort Order * Display Order * Logout * Delete Order * End the System |

**Detail Description for Each Use Cases**

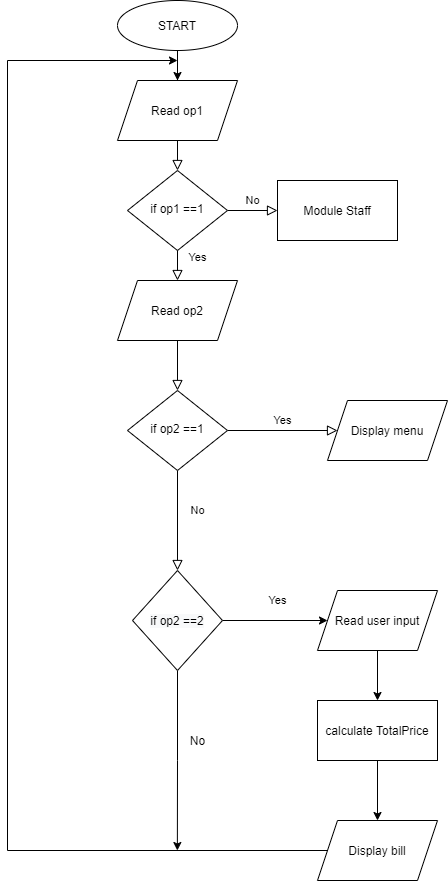
The system has 9 main use cases.

|  |  |
| --- | --- |
| Use Case Purpose | Use Case Purpose |
| Take Order | Customers can order their preferred food. |
| Payment | Customers will be given summary of their orders as bill |
| Display Menu | Customers and Staff can see what food is available. |
| Search Order | Staff can search for the order details of the customer using a unique customer ID. |
| Sort Order | Staff can sort all the orders that are recorded. |
| Display Order | Staff can display all the recorded customer orders. |
| Logout | Staff can log out from staff module |
| Delete Order | Staff can delete the first customer’s order |

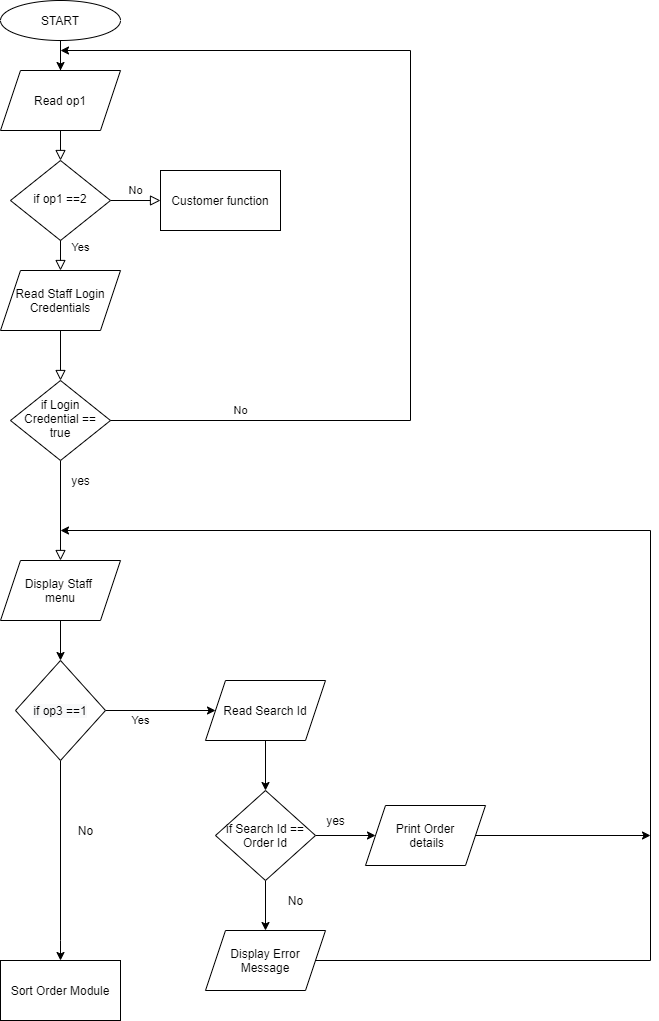
**2.2 System Design**

**Algorithm: Flowchart for each module.**

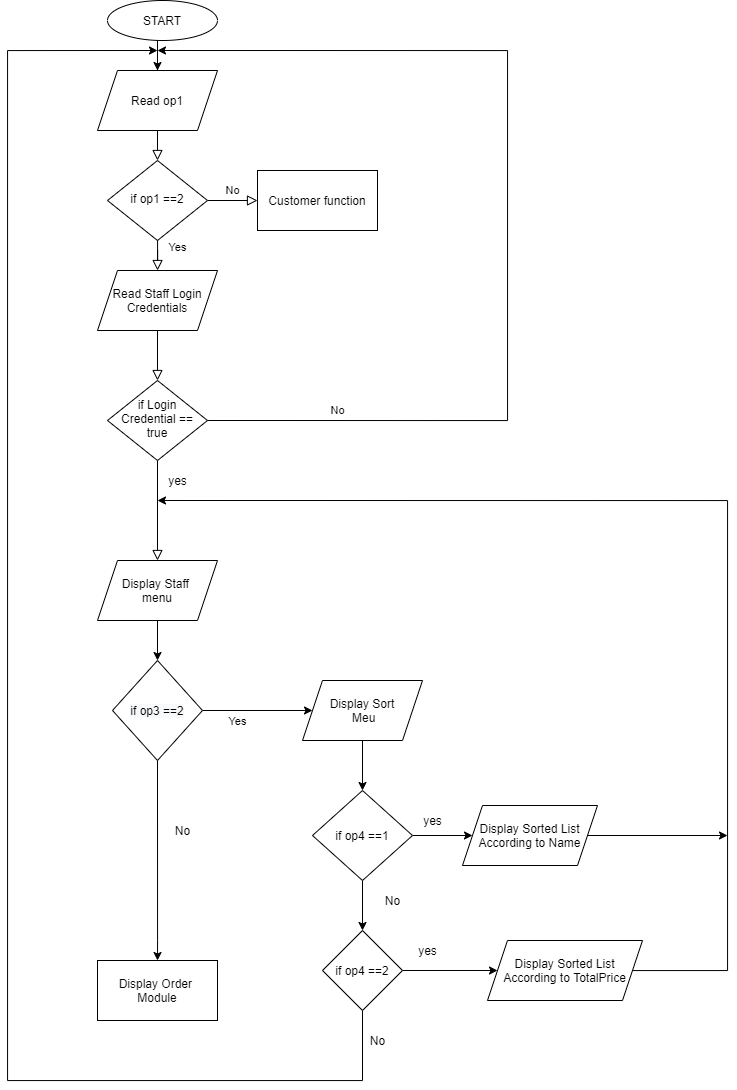
**Flowchart 1: Take Order by Customer**



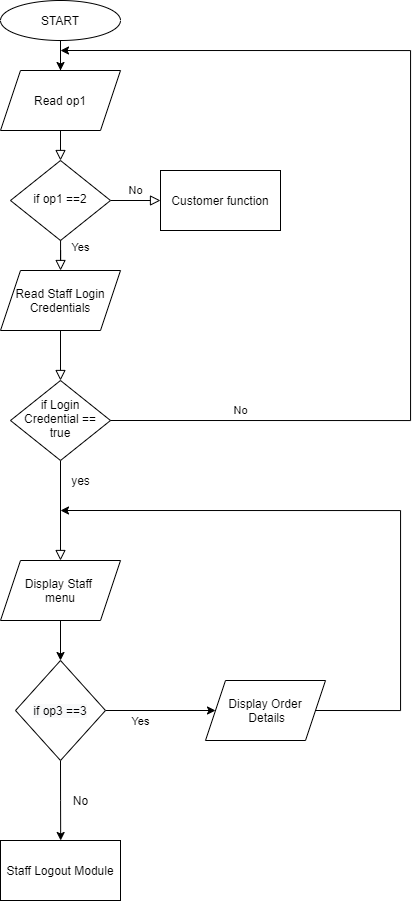
**FlowChart 2: Search Order by Staff**



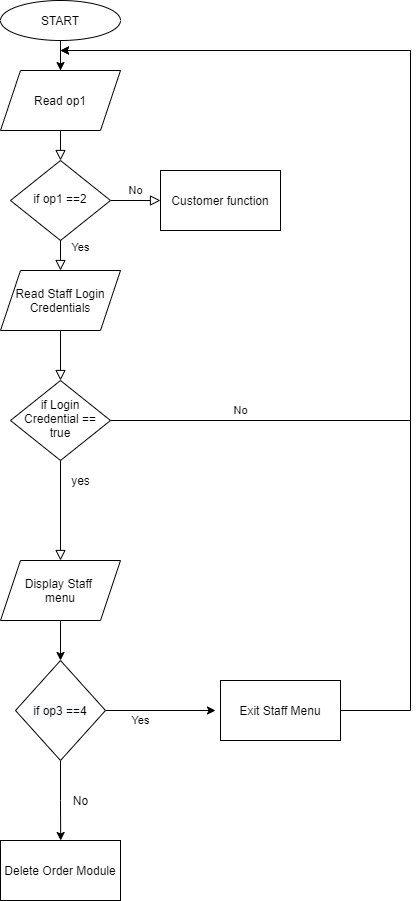
**Flowchart 3: Sort Order by Staff**



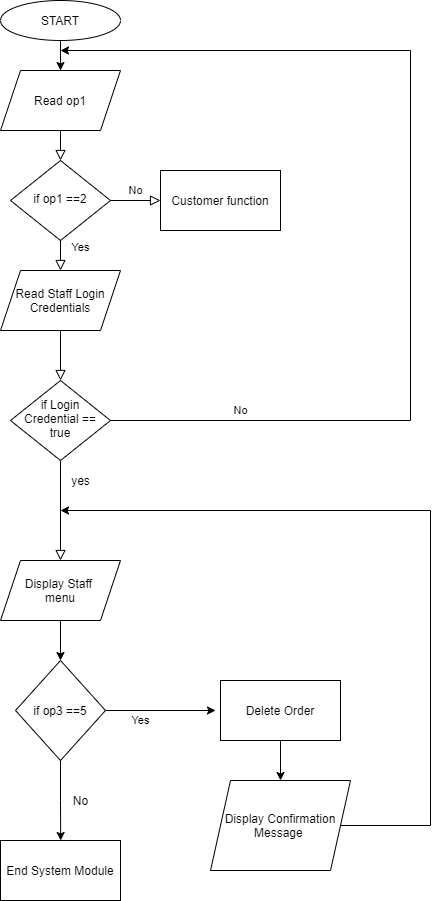
**Flowchart 4: Display Order by Staff**



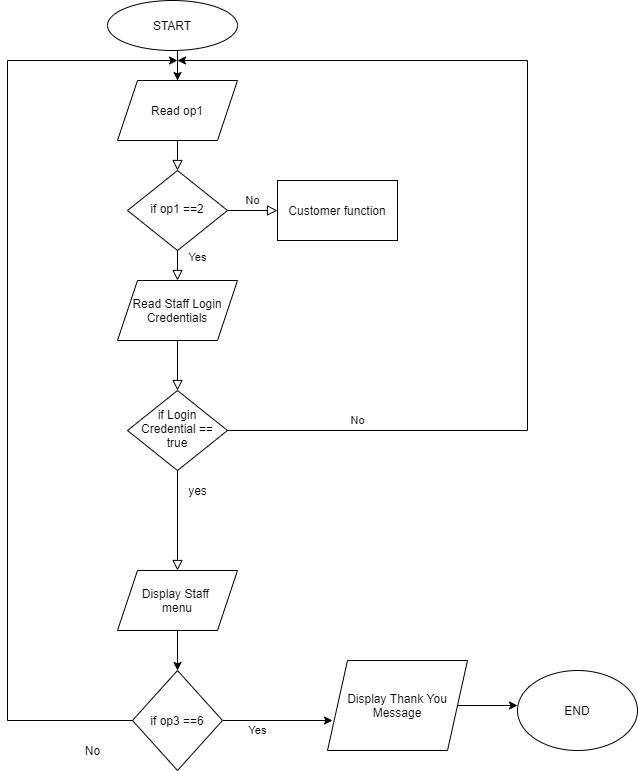
**Flowchart 5: Logout by Staff**



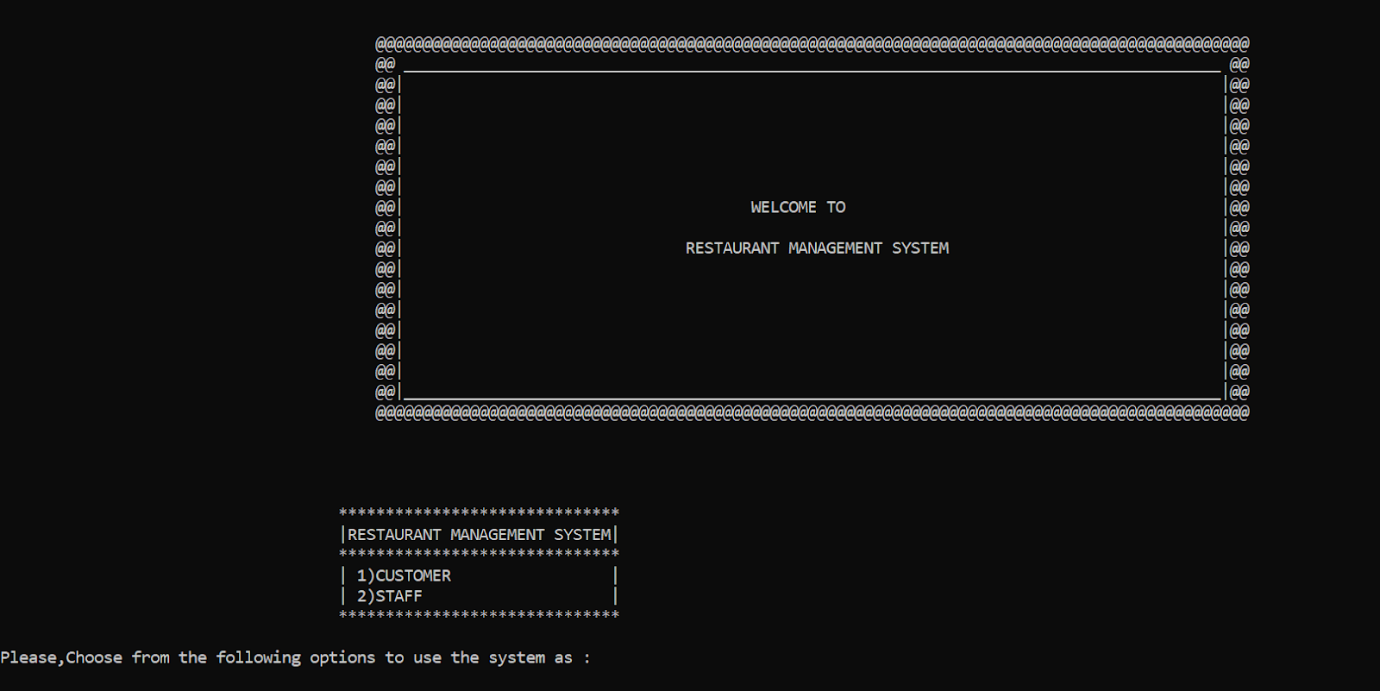
**Flowchart 6: Delete Order by Staff**



**Flowchart 7 :Exit Program by Staff**



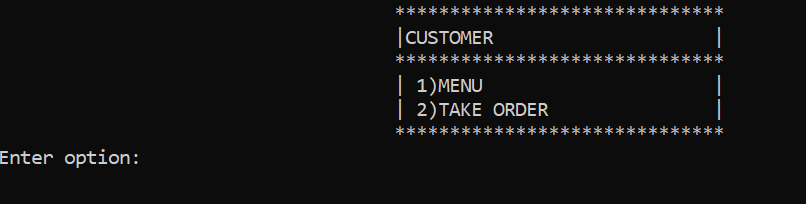
**Part 3: SYSTEM PROTOTYPE**

****

**Screen 1: Main Menu**

**Screen 1: The user must insert an integer value in the range 1-2 to use the system as a customer or staff. If the user enters another number, the system will prompt an error message and the screen is displayed again.**

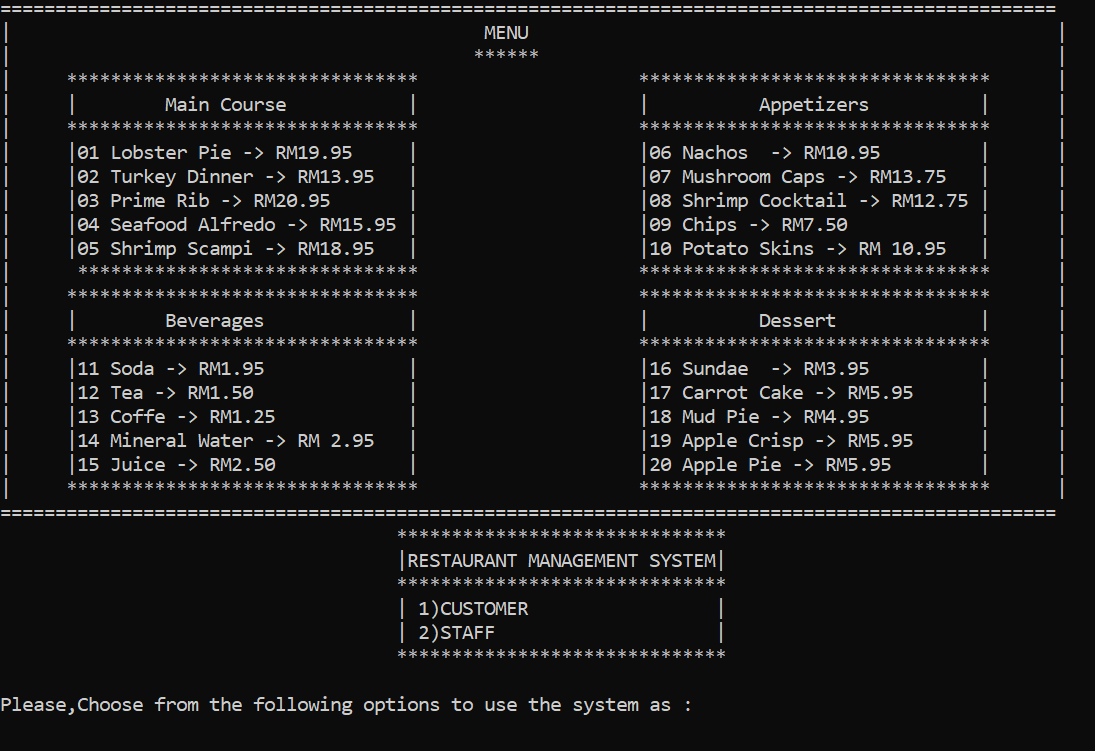
**Prepared By:Narendran.**

****

**Screen 2: Customer’s Main Menu**

**Screen 2: If the user enters the integer value of 1, the main menu of the customer will be displayed. Then the user must insert an integer value in the range 1-2 to view the menu or take order. If the user enters another number, the system will prompt an error message and the screen is displayed again.**

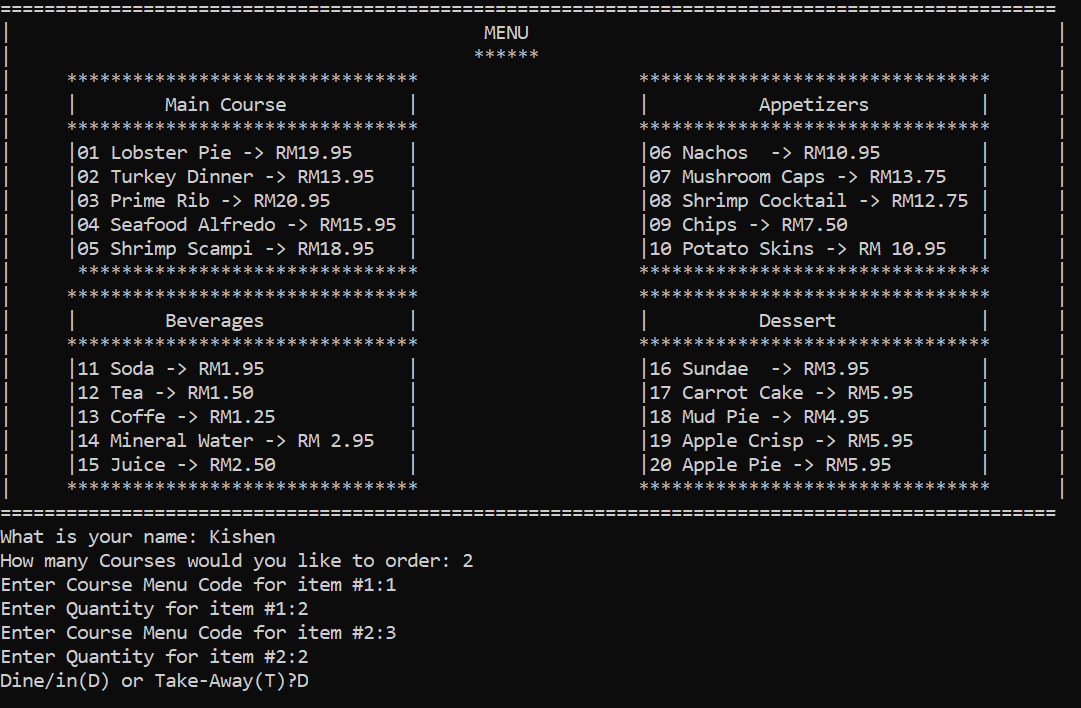
**Prepared By:Narendran.**



**Screen 2.1: Food Menu**

**Screen 2.1: If the user enters the integer value of 1, the food menu of the customer will be displayed. Then the user will be redirected to the main menu of the system.**

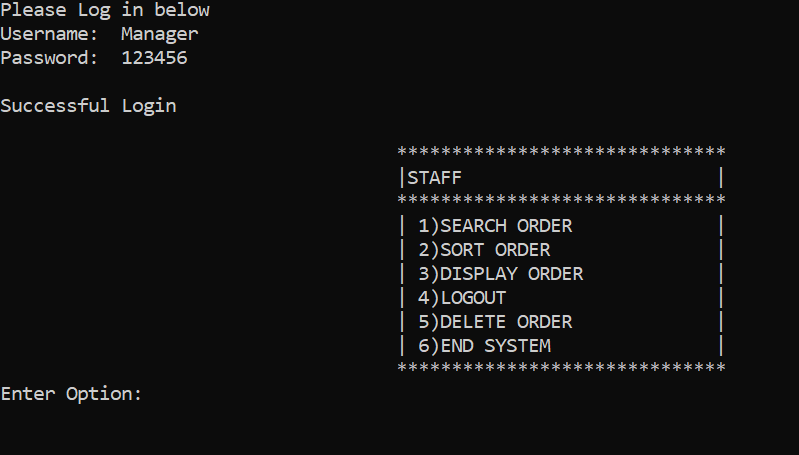
**Prepared By:Kishen.**



**Screen 2.2: Take Order**

**Screen 2.2: If the user enters the integer value of 2 in the customer’s main menu, the user will be displayed with the food menu and asked to input the name of customer, number of courses would like to order, quantity of each item and also dine option. Then user will be displayed with the bill of the customer’s order with total price.**

**Prepared By:Kishen.**



**Screen 3: Staff Login**

**Screen 3: The user must insert an integer value in the range 2 to use the system as the staff. After entering the integer value, the user needs to input username and password to access the staff main menu. If the user enters the wrong username or password, the system will prompt an error message and the screen is displayed again.**

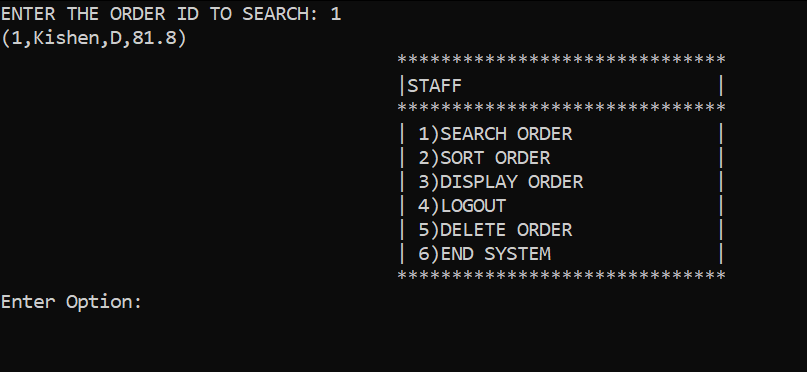
**Prepared By: Thanneermalai**

****

**Screen 4: Staff Main Menu**

**Screen 4: The user must insert an integer value in the range 1-6 to use the system for uses like search order, sort order, display order, delete order, logout and also end system. If the user enters another number, the system will prompt an error message and the screen is displayed again.**

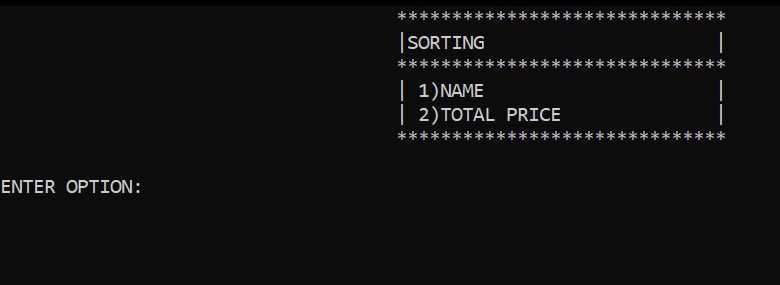
**Prepared By: Thanneermalai**

****

**Screen 4.1: Search Order**

**Screen 4.1: If the user enters the integer value of 1 in the staff’s main menu, the user will be redirected to search order screen and asked to input the order ID of the customer needed to search. Then user will be displayed with the details of the customer such as orderID, customer name, Dine option and also total price.**

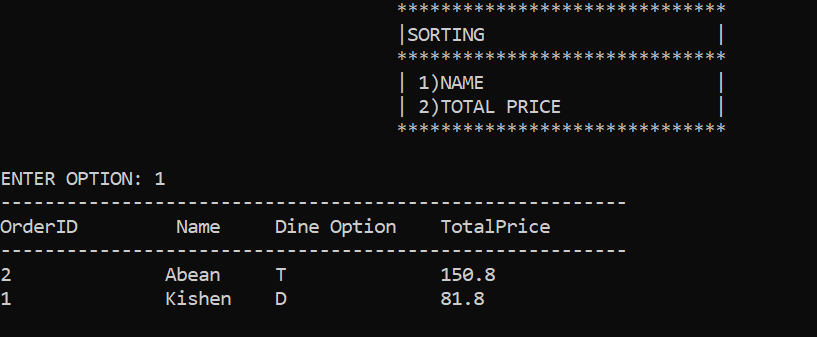
**Prepared By: Narendran**

****

**Screen 4.2: Sort Order**

**Screen 4.2: If the user enters the integer value of 1 in the staff’s main menu, the user will be redirected to the sort order menu. The user will be asked to input an integer value 1 or 2 to sort the order by customer name or total price.**

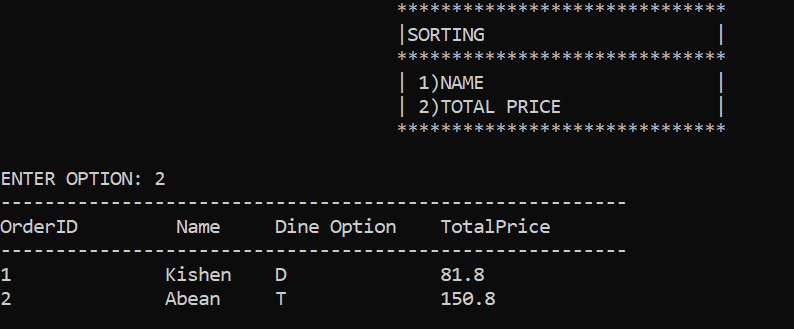
**Prepared By: Kishen**

****

**Screen 4.2.1: Sort by name**

**Screen 4.2.1: If the user input the integer value of 1 in the sort order menu, the user will be directly displayed with orders which are sorted by name in ascending order.**

**Prepared By: Thanneermalai**

****

**Screen 4.2.2: Sort by price**

**Screen 4.2.2: If the user inputs the integer value of 2 in the sort order menu, the user will be directly displayed with orders which are sorted by total price in ascending order.**

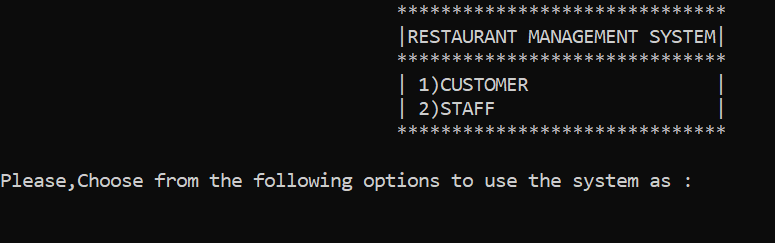
**Prepared By: Kishen**



**Screen 4.3.: Display order**

**Screen 4.3: If the user enters the integer value of 3 in the staff’s main menu, the user will be directly displayed with the order details of all customers stored.**

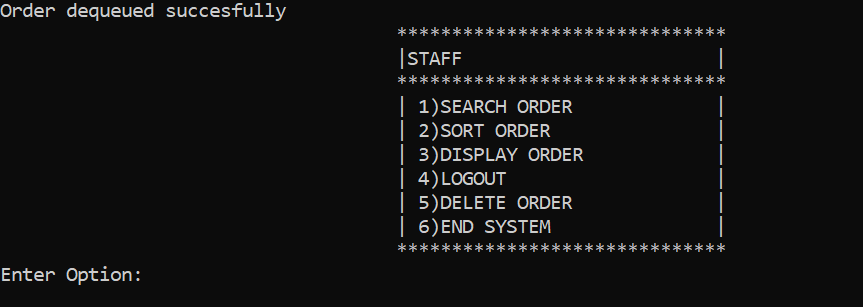
**Prepared By: Narendran**

****

**Screen 4.4: Logout**

**Screen 4.4: If the user enters the integer value of 4 in the staff’s main menu, the user will be directly redirected to the main menu of the system where the user will be asked again whether the user wants to use the system as customer or staff.**

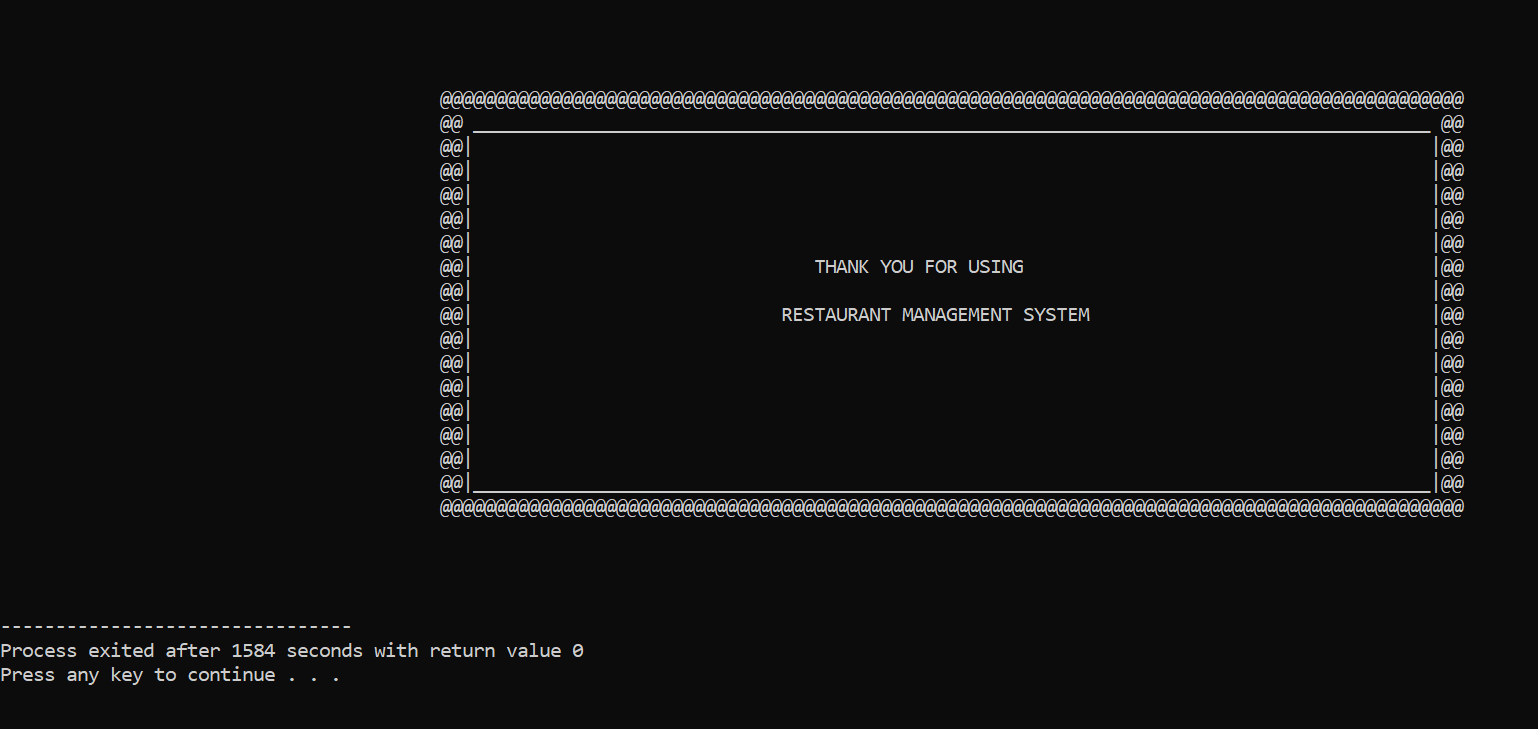
**Prepared By: Thanneermalai**

****

**Screen 4.5: Delete Order**

**Screen 4.5: If the user enters the integer value of 5 in the staff’s main menu, the first order stored in the system will be deleted and “Order Dequeued successfully” message will be displayed to the user to have a confirmation.**

**Prepared By: Kishen**

****

**Screen 4.6: End System**

**Screen 4.6: If the user enters the integer value of 6 in the staff’s main menu, the user will be displayed with a “Thank you” message and the system will end.**

**Prepared By: Narendran**

**Part 4: DEVELOPMENT ACTIVITIES**

\*We didn’t separate task as we did together \*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Online Meeting Date** | **Members Participate in the meeting** | **Activity** | **Task for each member** | **Task Achieved (yes/No)** |
| **13/01/2021** | **Thanneermalai**  **Kishen**  **Naren** | **CUSTOMER:**  **Take Order**  **Payment**  **Debugging** | **Thanneermalai**  **- Take Order**  **Kishen**  **-Payment**  **Naren**  **-Debugging** | **YES** |
| **14/01/2021** | **Thanneermalai**  **Kishen**  **Naren** | **STAFF:**  **Staff Login**  **Search Order**  **Debugging** | **Thanneermalai**  **-Debugging**  **Kishen**  **-Search Order**  **Naren**  **-Staff Login** | **YES** |
| **15/01/2021** | **Thanneermalai**  **Kishen**  **Naren** | **STAFF:**  **Sort Order**  **Display Order**  **Debugging** | **Thanneermalai**  **-Display Order**  **Kishen**  **-Debugging**  **Naren**  **-Sort Order** | **YES** |
| **16/01/2021** | **Thanneermalai**  **Kishen**  **Naren** | **STAFF:**  **Logout**  **Delete Order**  **Debugging** | **Thanneermalai**  **- Logout**  **Kishen**  **-Delete Order**  **Naren**  **-Debugging** | **YES** |
| **17/01/2021** | **Thanneermalai**  **Kishen**  **Naren** | **Project finalisation, final debugging, final test run,**  **Preliminary report discussion.** | **Thanneermalai**  **-final debugging**  **Kishen**  **-final debugging**  **Naren**  **-final debugging** | **YES** |
| **19/01/2021** | **Thanneermalai**  **Kishen**  **Naren** | **Project Report Part 1 &2** | **Thanneermalai**  **-Flow chart**  **Kishen**  **-Use case diagram & description**  **Naren**  **-Flow chart** | **YES** |
| **21/01/2021** | **Thanneermalai**  **Kishen**  **Naren** | **Project Report Part 3 &4** | **Thanneermalai**  **-All together**  **Kishen**  **-All together**  **Naren**  **-All together** | **YES** |
| **22/01/2021** | **Thanneermalai**  **Kishen**  **Naren** | **Preparation of Presentation Slide** | **Thanneermalai**  **-All together**  **Kishen**  **-All together**  **Naren**  **-All together** | **YES** |
| **23/01/2021** | **Thanneermalai**  **Kishen**  **Naren** | **Video Recording** | **Thanneermalai**  **-All together**  **Kishen**  **-All together**  **Naren**  **-All together** | **YES** |

Part 5: APPENDIX

#include<iostream>

#include<iomanip>

#include<fstream>

#define SIZE 20

using namespace std;

class item{

public:

int Order\_Code[20];

string item\_Name[20];

float price[20];

int quantity[20];

float totalPrice;

};

class Node

{

public:

int Order\_ID;

string cust\_name;

float Total\_Price;

float totalP;

char Dine\_option;

item obj[SIZE];

int quant;

Node \*nextnode;

Node()

{

Order\_ID=0;

cust\_name=" ";

Total\_Price=0.0;

Dine\_option='X';

quant=0;

nextnode=NULL;

for(int i=0;i<SIZE;i++){

obj[i].Order\_Code[i]=0;

obj[i].item\_Name[i]=" ";

obj[i].price[i]=0.0;

obj[i].quantity[i]=0;

}

}

};

class Order{

public:

Node \*backPtr;

Node \*frontPtr;

void createQueue(){

backPtr = NULL;

frontPtr = NULL;

}

bool isEmpty()

{

if(backPtr == NULL && frontPtr == NULL)

{

return true;

}

else

return false;

}

void enqueue(Node \*n1)

{

if(isEmpty())

{

n1->nextnode=NULL;

frontPtr=backPtr=n1;

cout<<"Order enqueued succesfully "<<endl;

}

else

{

backPtr->nextnode=n1;

backPtr=n1;

cout<<"Order enqueued succesfully "<<endl;

}

}

void find(int x)

{

Node \*temp=frontPtr;

for(int i=0;temp->nextnode != NULL && temp->Order\_ID != x;i++)

{

temp = temp->nextnode;

}

if(temp->Order\_ID == x)

{

cout<<"("<<temp->Order\_ID<<","<<temp->cust\_name<<","<<temp->Dine\_option<<","<<temp->Total\_Price<<")";

cout<<endl;

}

else if(temp->nextnode == NULL)

{

cout << "Error: Number Not found..." << endl;

}

}

void display()

{

if(isEmpty())

{

cout << "Queue is Empty" << endl;

}

else

{

Node \*temp=frontPtr;

cout<<left;

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

cout<<setw(15)<<"OrderID "<<setw(10)<<" Name "<<setw(15)<<"Dine Option"<<setw(20)<<"TotalPrice"<<endl;

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

while(temp!=NULL)

{

cout<<setw(15)<<temp->Order\_ID<<setw(10)<<temp->cust\_name<<setw(15)<<temp->Dine\_option<<setw(20)<<temp->Total\_Price<<endl;

temp=temp->nextnode;

}

cout<<endl;

}

}

void deleteQueue()

{

Node \*temp=frontPtr;

if(isEmpty())

{

cout << "Queue is Empty. Nothing to be deleted" << endl;

}

else

{

frontPtr=temp->nextnode;

delete temp;

temp=frontPtr;

cout<<"Order dequeued succesfully"<<endl;

}

}

void sortList1() {

Node \*current = frontPtr, \*index = NULL;

string temp;

int temp1;

char temp2;

float temp3;

if(frontPtr == NULL) {

return;

}

else {

while(current != NULL) {

index = current->nextnode;

while(index != NULL) {

if(current->cust\_name > index->cust\_name) {

temp = current->cust\_name;

current->cust\_name = index->cust\_name;

index->cust\_name = temp;

temp1 = current->Order\_ID;

current->Order\_ID = index->Order\_ID;

index->Order\_ID = temp1;

temp2 = current->Dine\_option;

current->Dine\_option = index->Dine\_option;

index->Dine\_option = temp2;

temp3 = current->Total\_Price;

current->Total\_Price = index->Total\_Price;

index->Total\_Price = temp3;

}

index = index->nextnode;

}

current = current->nextnode;

}

}

}

void sortList2() {

Node \*current = frontPtr, \*index = NULL;

string temp;

int temp1;

char temp2;

float temp3;

if(frontPtr == NULL) {

return;

}

else {

while(current != NULL) {

index = current->nextnode;

while(index != NULL) {

if(current->Total\_Price > index->Total\_Price) {

temp = current->cust\_name;

current->cust\_name = index->cust\_name;

index->cust\_name = temp;

temp1 = current->Order\_ID;

current->Order\_ID = index->Order\_ID;

index->Order\_ID = temp1;

temp2 = current->Dine\_option;

current->Dine\_option = index->Dine\_option;

index->Dine\_option = temp2;

temp3 = current->Total\_Price;

current->Total\_Price = index->Total\_Price;

index->Total\_Price = temp3;

}

index = index->nextnode;

}

current = current->nextnode;

}

}

}

};

void menu(){

cout<<"================================================================================================"<<endl;

cout<<"| MENU |"<<endl;

cout<<"| \*\*\*\*\*\* |"<<endl;

cout<<"| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |"<<endl;

cout<<"| | Main Course | | Appetizers | |"<<endl;

cout<<"| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |"<<endl;

cout<<"| |01 Lobster Pie -> RM19.95 | |06 Nachos -> RM10.95 | |"<<endl;

cout<<"| |02 Turkey Dinner -> RM13.95 | |07 Mushroom Caps -> RM13.75 | |"<<endl;

cout<<"| |03 Prime Rib -> RM20.95 | |08 Shrimp Cocktail -> RM12.75 | |"<<endl;

cout<<"| |04 Seafood Alfredo -> RM15.95 | |09 Chips -> RM7.50 | |"<<endl;

cout<<"| |05 Shrimp Scampi -> RM18.95 | |10 Potato Skins -> RM 10.95 | |"<<endl;

cout<<"| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |"<<endl;

cout<<"| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |"<<endl;

cout<<"| | Beverages | | Dessert | |"<<endl;

cout<<"| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |"<<endl;

cout<<"| |11 Soda -> RM1.95 | |16 Sundae -> RM3.95 | |"<<endl;

cout<<"| |12 Tea -> RM1.50 | |17 Carrot Cake -> RM5.95 | |"<<endl;

cout<<"| |13 Coffe -> RM1.25 | |18 Mud Pie -> RM4.95 | |"<<endl;

cout<<"| |14 Mineral Water -> RM 2.95 | |19 Apple Crisp -> RM5.95 | |"<<endl;

cout<<"| |15 Juice -> RM2.50 | |20 Apple Pie -> RM5.95 | |"<<endl;

cout<<"| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |"<<endl;

cout<<"================================================================================================"<<endl;

}

void displayBill(Node \*n1,int x)

{ ofstream out;

out.open("output.txt", ios::out | ios::app | ios::binary);

float sum1=0;

static int count;

count++;

out<<"Customer"<<"#"<<count<<endl;

cout<<left;

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

cout<<"| RESTRAURANT ABC |"<<endl;

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

cout<<"| OrderID= "<<n1->Order\_ID<<" "<<"Customer Name= "<<n1->cust\_name<<" "<<"DineOption= "<<n1->Dine\_option<<" |"<<endl;

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

cout<<setw(10)<<"ItemCode"<<setw(20)<<"ItemName"<<setw(15)<<"Quantity"<<setw(5)<<"Price"<<endl;

cout<<setw(10)<<"\*\*\*\*\*\*\*\*"<<setw(20)<<"\*\*\*\*\*\*\*\*"<<setw(15)<<"\*\*\*\*\*\*\*\*"<<setw(5)<<"\*\*\*\*\*"<<endl<<endl;

for(int i=0;i<x;i++)

{

sum1=sum1+(n1->obj[i].price[i]\*n1->obj[i].quantity[i]);

cout<<setw(10)<<n1->obj[i].Order\_Code[i]<<setw(20)<<n1->obj[i].item\_Name[i]<<setw(15)<<n1->obj[i].quantity[i]<<"RM "<<(n1->obj[i].price[i]\*n1->obj[i].quantity[i]);

cout<<endl<<endl;

out<<"("<<n1->obj[i].Order\_Code[i]<<","<<n1->obj[i].item\_Name[i]<<","<<n1->obj[i].quantity[i]<<","<<"RM "<<(n1->obj[i].price[i]\*n1->obj[i].quantity[i])<<")"<<endl<<endl;

}

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

cout<<"TotalPrice= RM "<<sum1<<endl;

cout<<endl<<endl<<endl;

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

cout<<"| THANK YOU FOR DINING WITH US! |"<<endl;

cout<<"| PLEASE COME AGAIN |"<<endl;

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

}

int checkpword(string user[],string pword[], int x, string username,string pass)

{

int temp=-1;

for(int i=0;i<x;i++)

{

if(user[i]==username)

{

cout<<"Welcome"<<endl;

return temp = 0;

break;

}

}

if(temp==-1)

{

cout<<"Try again "<<endl;

return temp;

}

}

int main()

{

int menu\_option=1,odid;

int item\_code[20];

int item\_quantity[20];

float item\_price[20];

int Course\_Quantity;

string item\_name[20];

string c\_name;

char c,bill;

int order=1;

Order s;

string name[10];

string pword[10];

int op1,op2,op3,op4;

string username=" ";

string password=" ";

bool loginsuccess = false;

ifstream in;

in.open("username&password.txt");

cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n";

cout<<"\t\t\t\t\t@@ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ @@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| WELCOME TO |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| RESTAURANT MANAGEMENT SYSTEM |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|@@\n";

cout<<"\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n\n\n\n"<<endl;

do{

float TotalPrice=0;

B:

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<" |RESTAURANT MANAGEMENT SYSTEM| "<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<" | 1)CUSTOMER | "<<endl;

cout<<" | 2)STAFF | "<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl<<endl;

cout<<"Please,Choose from the following options to use the system as : "<<endl<<endl;

cin>>op1;

system("CLS");

if(op1==1)

{

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<" |CUSTOMER | "<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<" | 1)MENU | "<<endl;

cout<<" | 2)TAKE ORDER | "<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<"Enter option: "<<endl;

cin>>op2;

system("CLS");

if(op2==1)

{

menu();

}

else if(op2==2)

{

Node \*n1=new Node();

menu();

cin.ignore();

cout<<"What is your name: ";

getline(cin,c\_name);

cout<<"How many Courses would you like to order: ";

cin>>Course\_Quantity;

for(int i=0;(i<Course\_Quantity && i<=20);i++){

cout<<"Enter Course Menu Code for item #" <<i+1<<":";

cin>>item\_code[i];

cout<<"Enter Quantity for item #"<<i+1<<":";

cin>>item\_quantity[i];

if(item\_code[i]==1){

item\_price[i]= 19.95;

item\_name[i]="Lobster Pie";

}

if(item\_code[i]==2){

item\_price[i]= 13.95;

item\_name[i]="Turkey Dinner";

}

if(item\_code[i]==3){

item\_price[i]= 20.95;

item\_name[i]="Prime Rib ";

}

if(item\_code[i]==4){

item\_price[i]= 15.95;

item\_name[i]="Seafood Alfredo";

}

if(item\_code[i]==5){

item\_price[i]= 18.95;

item\_name[i]="Shrimp Scampi";

}

if(item\_code[i]==6){

item\_price[i]= 10.95;

item\_name[i]="Nachos";

}

if(item\_code[i]==7){

item\_price[i]= 13.75;

item\_name[i]="Mushroom Caps";

}

if(item\_code[i]==8){

item\_price[i]= 12.75;

item\_name[i]="Shrimp Cocktail";

}

if(item\_code[i]==9){

item\_price[i]= 7.50;

item\_name[i]="Chips";

}

if(item\_code[i]==10){

item\_price[i]= 10.95;

item\_name[i]="Potato Skins";

}

if(item\_code[i]==11){

item\_price[i]= 1.95;

item\_name[i]="Soda";

}

if(item\_code[i]==12){

item\_price[i]= 1.50;

item\_name[i]="Tea";

}

if(item\_code[i]==13){

item\_price[i]= 1.25;

item\_name[i]="Coffee";

}

if(item\_code[i]==14){

item\_price[i]= 2.95;

item\_name[i]="Mineral Water ";

}

if(item\_code[i]==15){

item\_price[i]= 2.50;

item\_name[i]="Juice ";

}

if(item\_code[i]==16){

item\_price[i]= 3.95;

item\_name[i]="Sundae";

}

if(item\_code[i]==17){

item\_price[i]= 5.95;

item\_name[i]="Carrot Cake";

}

if(item\_code[i]==18){

item\_price[i]= 4.95;

item\_name[i]="Mud Pie";

}

if(item\_code[i]==19){

item\_price[i]= 5.95;

item\_name[i]="Apple Crisp";

}

if(item\_code[i]==20){

item\_price[i]= 5.95;

item\_name[i]="Apple Pie";

}

}

cout<<"Dine/in(D) or Take-Away(T)?";

cin>>c;

cout<<endl<<endl;

for(int i=0;i< Course\_Quantity;i++)

{

TotalPrice = TotalPrice +(item\_price[i] \* item\_quantity[i]);

}

n1->Order\_ID=order;

n1->cust\_name=c\_name;

n1->Dine\_option=c;

n1->Total\_Price=TotalPrice;

for(int i=0;i<Course\_Quantity;i++)

{

n1->obj[i].item\_Name[i]=item\_name[i];

n1->obj[i].Order\_Code[i]=item\_code[i];

n1->obj[i].price[i]=item\_price[i];

n1->obj[i].quantity[i]=item\_quantity[i];

}

system("CLS");

displayBill(n1,Course\_Quantity);

order++;

s.enqueue(n1);

}

}

else if(op1==2)

{

system("CLS");

cout<<"Please Log in below"<<endl;

do{

cout<<"Username: ";

cin>>username;

cout<<"Password: ";

cin>>password;

if(username=="Manager"&&password=="123456")

{

cout<<"\nSuccessful Login\n\n";

loginsuccess=true;

}

else

{

cout<<"Incorrect username or password \n";

cout<<"Please try to login again\n";

goto B;

}

}while(!loginsuccess);

A:

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<" |STAFF | "<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<" | 1)SEARCH ORDER | "<<endl;

cout<<" | 2)SORT ORDER | "<<endl;

cout<<" | 3)DISPLAY ORDER | "<<endl;

cout<<" | 4)LOGOUT | "<<endl;

cout<<" | 5)DELETE ORDER | "<<endl;

cout<<" | 6)END SYSTEM | "<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<"Enter Option: ";

cin>>op3;

system("CLS");

if(op3==1)

{

cout<<"ENTER THE ORDER ID TO SEARCH: ";

cin>>odid;

s.find(odid);

goto A;

}

else if(op3==2)

{

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<" |SORTING | "<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl;

cout<<" | 1)NAME | "<<endl;

cout<<" | 2)TOTAL PRICE | "<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<endl<<endl;

cout<<"ENTER OPTION: ";

cin>>op4;

if (op4==1)

{

s.sortList1();

s.display();

goto A;

}

else if (op4==2)

{

s.sortList2();

s.display();

goto A;

}

}

else if(op3==3)

{

s.display();

goto A;

}

else if(op3==4)

{

goto B;

}

else if(op3==5)

{

s.deleteQueue();

order--;

goto A;

}

}

}while(op3!=6);

cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n";

cout<<"\t\t\t\t\t@@ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ @@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| THANK YOU FOR USING |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| RESTAURANT MANAGEMENT SYSTEM |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@| |@@\n";

cout<<"\t\t\t\t\t@@|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|@@\n";

cout<<"\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n\n\n\n\t\t\t\t\t";

}