



School of Computing

Faculty of Engineering

UNIVERSITI TEKNOLOGI MALAYSIA

DATA STRUCTURE & ALGORITHM

(SECJ 2013)

SEMESTER 1 2020/2021

MINI PROJECT DOCUMENTATION

FOOD ORDERING SYSTEM

By

**MD MONIRUL ISLAM MOLLA (A19EC4013) - LEADER
MUHAMMAD SHAFAYET HOSSAIN (A19EC4016)
MD ABU SALEH NAYEEM (A19EC4012)
ANIKA SHAHZAD (A19EC4007)**

SECTION 05

Lecturer:

DR. JOHANNA BINTI AHMED

Date: 27-01-2021

TABLE OF CONTENTS

1.0 PART 1: INTRODUCTION

1.1 SYNOPSIS PROJECT

1.2 OBJECTIVE OF PROJECT

2.0 PART 2: SYSTEM ANALYSIS AND DESIGN

2.1 SYSTEM REQUIREMENTS

2.1.1 USE CASE DIAGRAM

2.2 SYSTEM DESIGN

2.2.1 FLOWCHART

2.3 CLASS DIAGRAM

3.0 PART 3: SYSTEM PROTOTYPE

4.0 PART 4: DEVELOPMENT ACTIVITIES

5.0 PART 5: APPENDIX

1.0 INTRODUCTION

1.1 SYNOPSIS PROJECT

The main target of the project is to construct a food ordering system to be used in the food service sector which really will facilitate restaurants to conveniently and easily accommodate an online menu that customers should access and use it with merely a few taps for place an order.

We have used a queue & linked list concepts throughout this procedure to insert, delete, sorting, searching etc. for users. The online ordering system which we implement here massively optimizes both the user and the restaurant staff/ordering administrator's process. The framework offers an accessible and up-to-date menu in an easy-to-use way for all current alternatives.

Users and executives both could indeed browse for commodities and organize them through value. Users might select any or even more goods which will appear in the Food Cart to make a purchase. Until checking out, users could see all the customer orders throughout the cart. In the end, the user needs order authorization information. The order is stored in the database and is accessed and extracted almost in perfect sync. It stimulates restaurant workers to swiftly go through requests as they are generated and manage all orders with minimal delays and uncertainty.

In this system we have applied a queue linked list for inserting, deleting, sorting etc. for both users. Online ordering system that we are proposing here greatly simplifies the ordering process for both the customer and the restaurant staff/administrator. System presents an interactive and up-to-date menu with all available options in an easy-to-use manner. Customers and administrators can search any products and do sorting by price. Customers can choose one or more items to place an order which will land in the Food Cart. Customers can view all the order details in the cart before checking out. At the end, the customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Restaurant Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.

1.2 OBJECTIVE OF THE PROJECT

The study's ultimate goals are to maintain a continuous, flexible and comprehensive ordering system. The primary objective of the Food Ordering System online is to administer the users' details of the item type, price, food, delivery address, order and food cart.

By limiting the operating costs associated, our system would allow hotels and restaurants to broaden the definition of operation. The system also provides flexibility and conveniently navigates an online menu which users can navigate and use with just a few clicks to make purchases. Employees of restaurants use these orders for efficient delivery through an easy to navigate graphical GUI.

This research sets out a structure for designing and marketing a new system for full usage and for providing a channel across the web whereby users can log in to our server and choose any items or food they want and then pay through the website.

The targets this would provide include the following:

- Increasing productivity and enhancing customer service through optimizing the technology and innovation in everyday activities.
- To establish a structure that will inevitably accommodate service quality.
- To be reluctant to step out in the food service industry from powerhouses. The browse resources for numerous categories of food can be acquired to enable users to request meals from the menu and users to sort by pricing, delete/insert food into the food cart.
- To allow users to deliver visual confirmation of the correct location of the order.
- To reduce restaurant's food wastage
- To ensure correct placement of orders through visual confirmation
- Improve efficiency of restaurant staff
- Eliminate paperwork and increase level of accuracy.
- Increase speed of service, sales volume and customer satisfaction

2.0 SYSTEM ANALYSIS AND DESIGN

2.1 SYSTEM REQUIREMENTS

2.1.1 USE CASE DIAGRAM

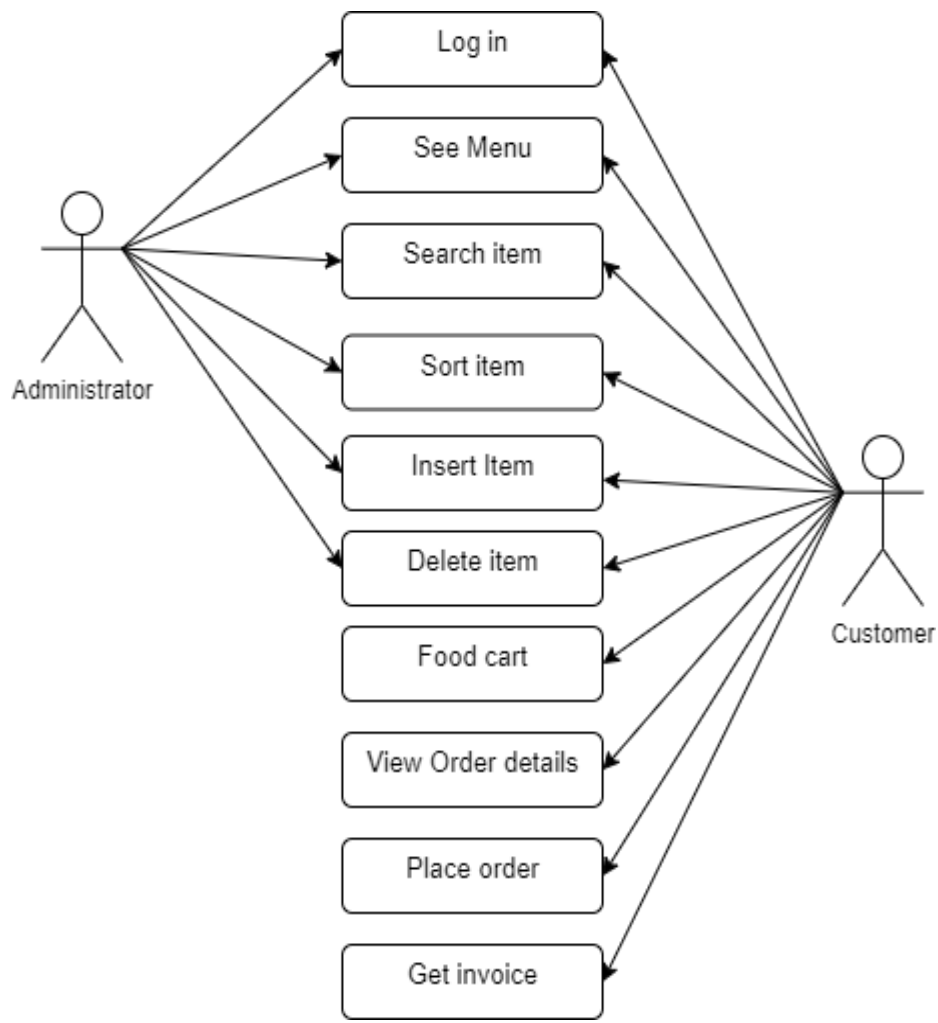


Figure 1: Use Case Diagram for Food Ordering System

2.1.2 USE CASE DESCRIPTION

Actor	Task
Administrator	Administrator can Log in, See the menu, Search item, Sort item, insert item into Menu, Delete item from menu.
Customer	Customers can Log in, see menu, Search item from the menu, Sort item by price, insert item into food cart, delete item from food cart, can view order details, Place order, and finally Get invoice.

2.1.3 DETAIL DESCRIPTION

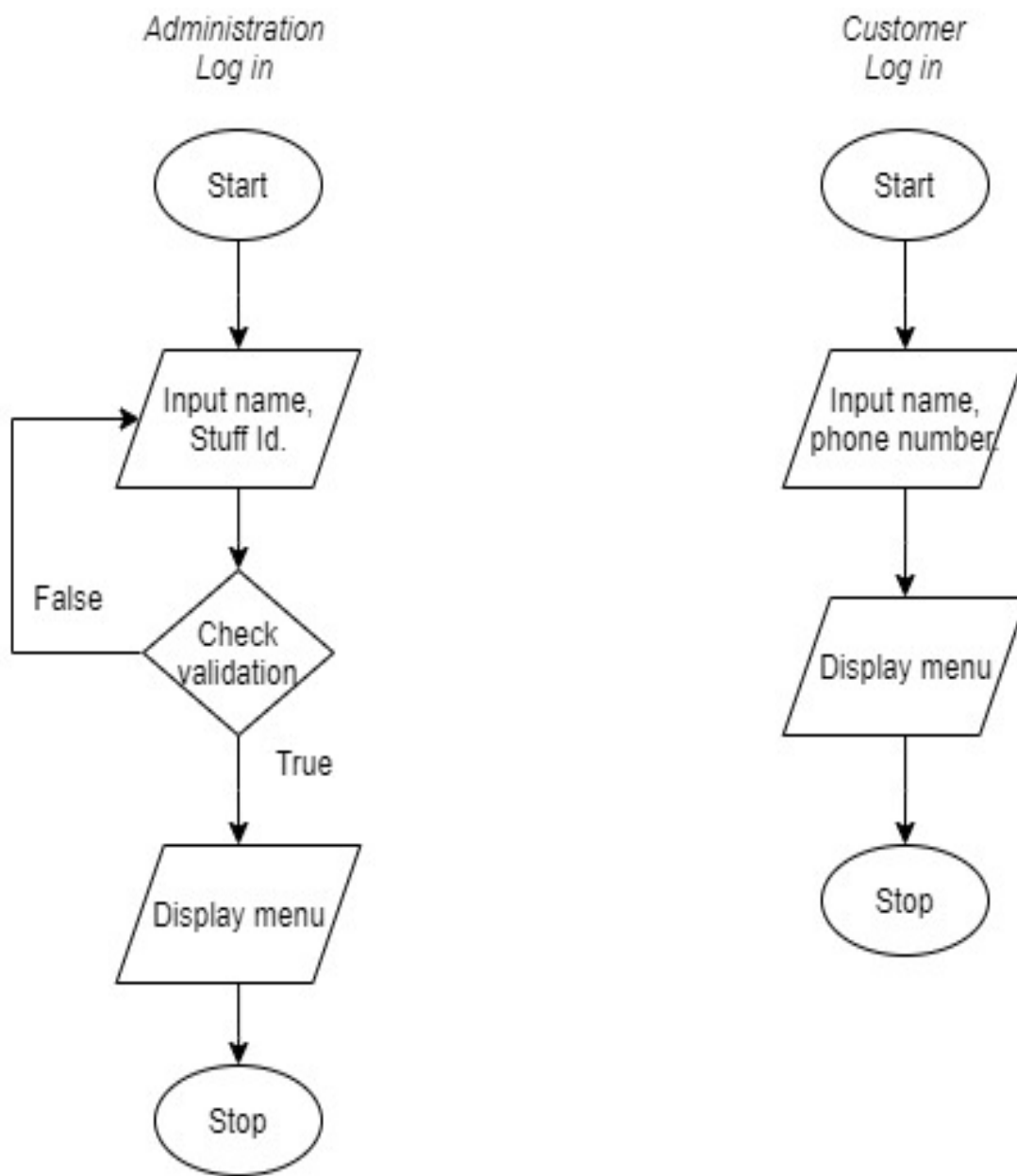
Use Case	Purpose
Log in	The user has to login to get access to the system.
See menu	It provides the food list so that users can see the available food.
Search item	The user can easily search the item from the main menu.
Sort item	The items in the system can efficiently be sorted throughout the system.
Insert item	The items can be inserted into the lists in the system.
Delete item	The items in the list can be deleted with the help of this function.
Food cart	The Food cart contains the item the customer wants to order.
View order details	The customer can conveniently view the order and make changes in it.
Place order	The order can be placed by the customer once the order is confirmed.
Get invoice	The customer will get the details of the order with the customer's name, customer's mobile number, FoodID, FoodName, Price and quantity of items.

2.2 SYSTEM DESIGN

2.2.1 FLOWCHART

Flowchart 1: Log in for administration and customer

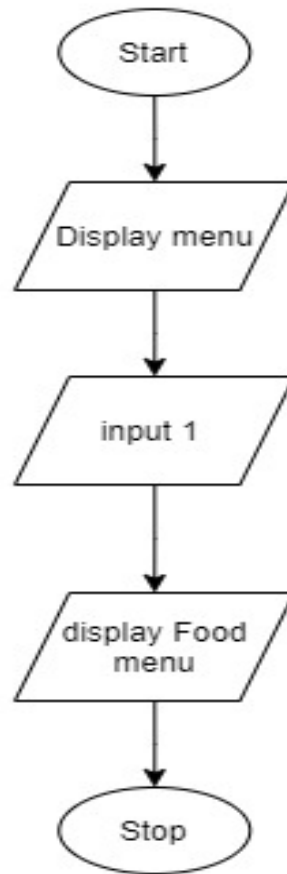
Prepared by: Md Abu Saleh Nayeem



Flowchart 2: See menu

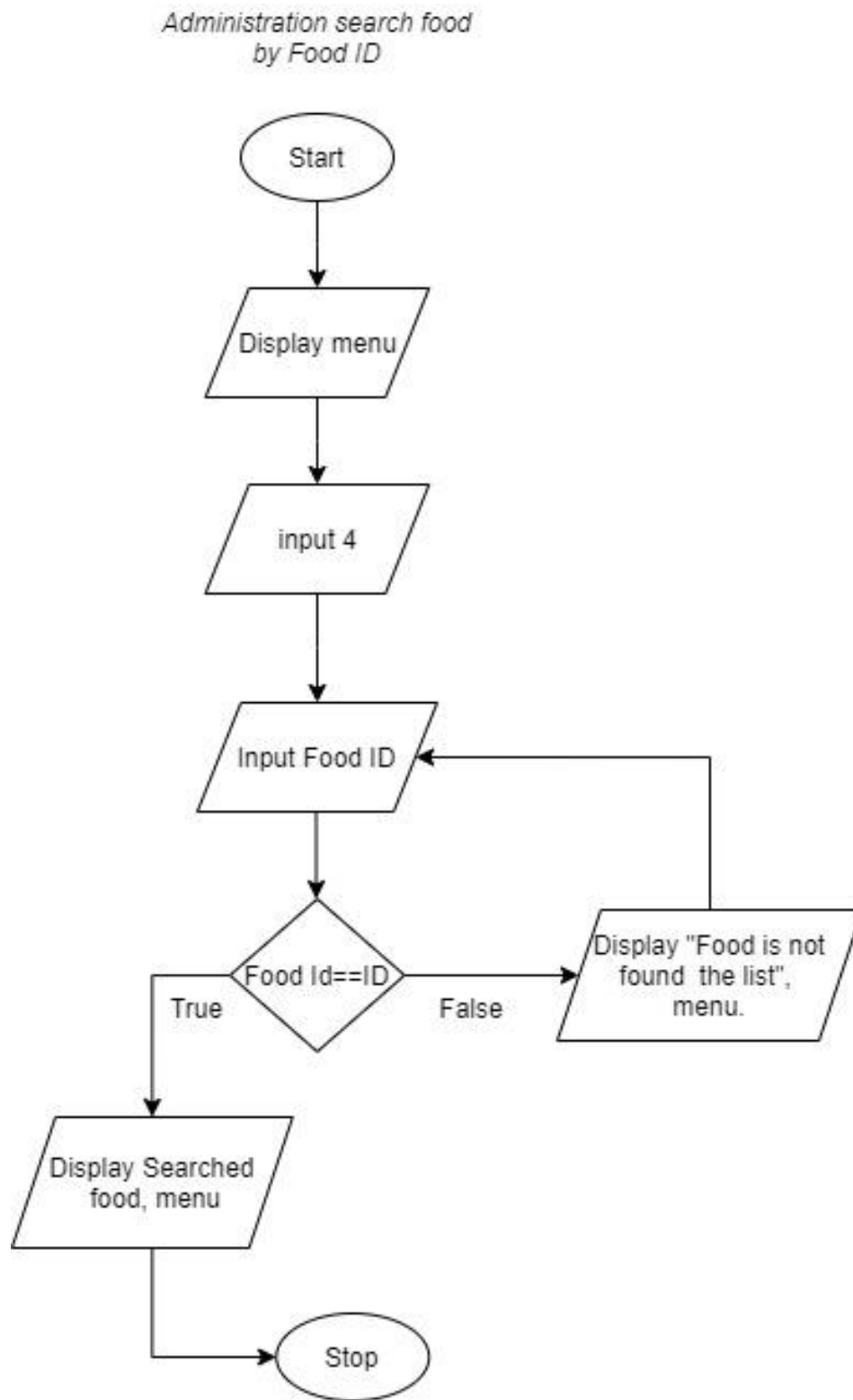
Prepared by: Md Abu Saleh Nayeem

*Administration / customer
see food menu*



Flowchart 3: Search Item

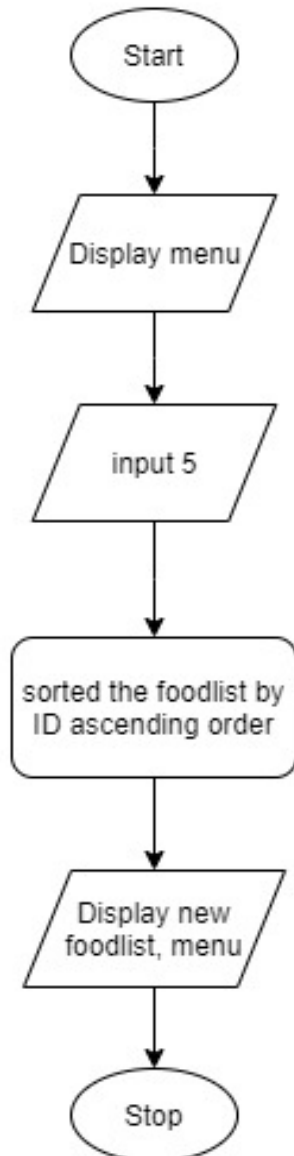
Prepared by: Md Abu Saleh Nayeem



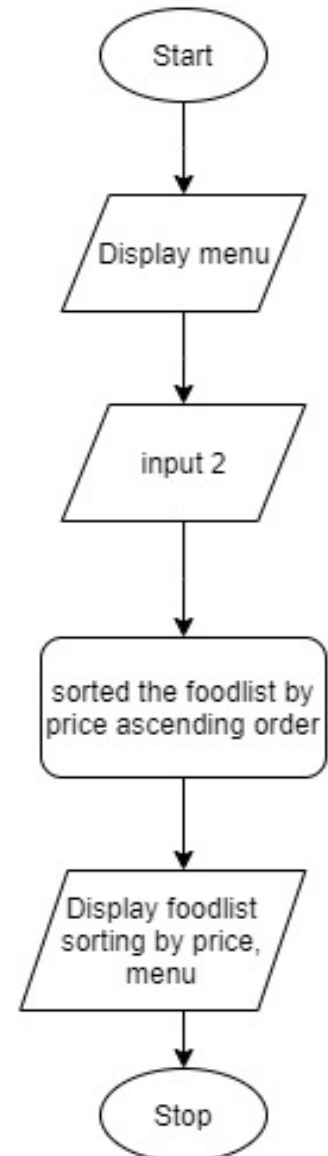
Flowchart 4: Sorting Food List for Administration and customer

Prepared by: Md Abu Saleh Nayeem

*Administration sort foodlist
by Food ID*

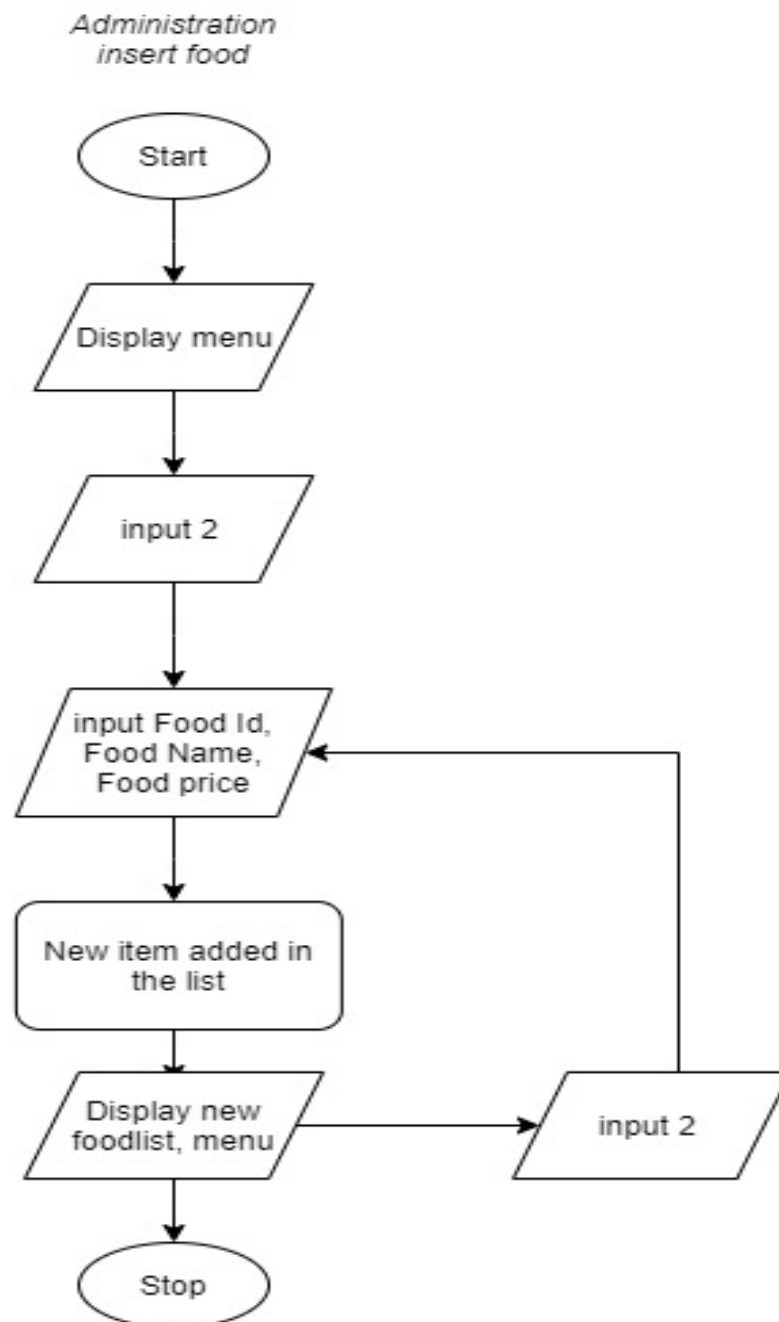


*Customer sort foodlist by
price*



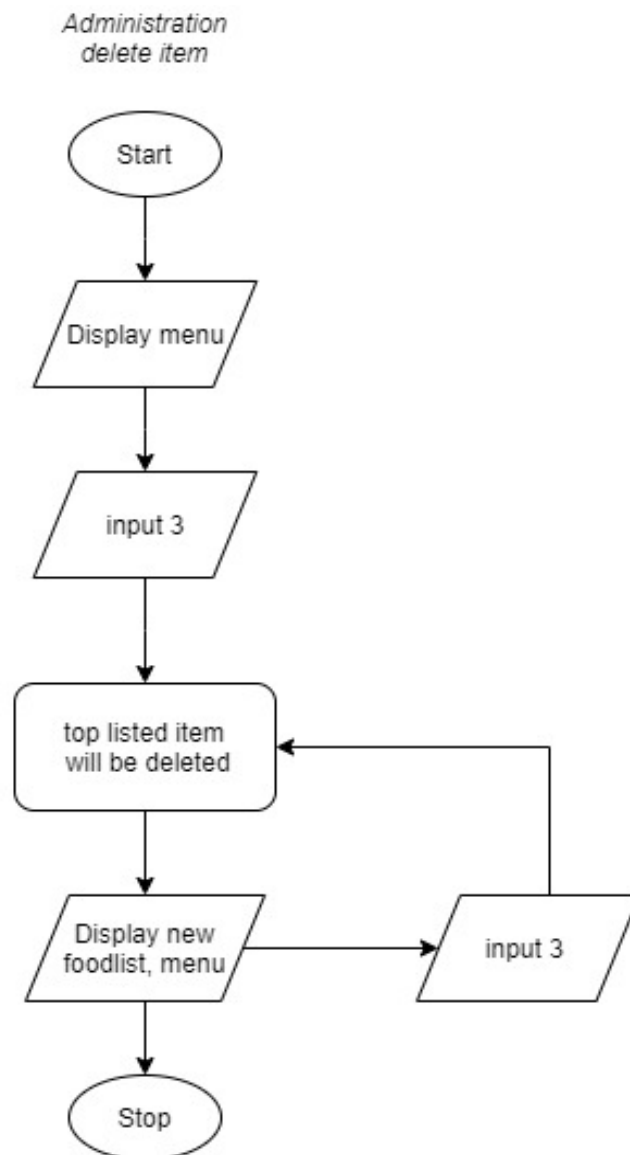
Flowchart 5: Insert food in the food list

Prepared by: Md Abu Saleh Nayeem



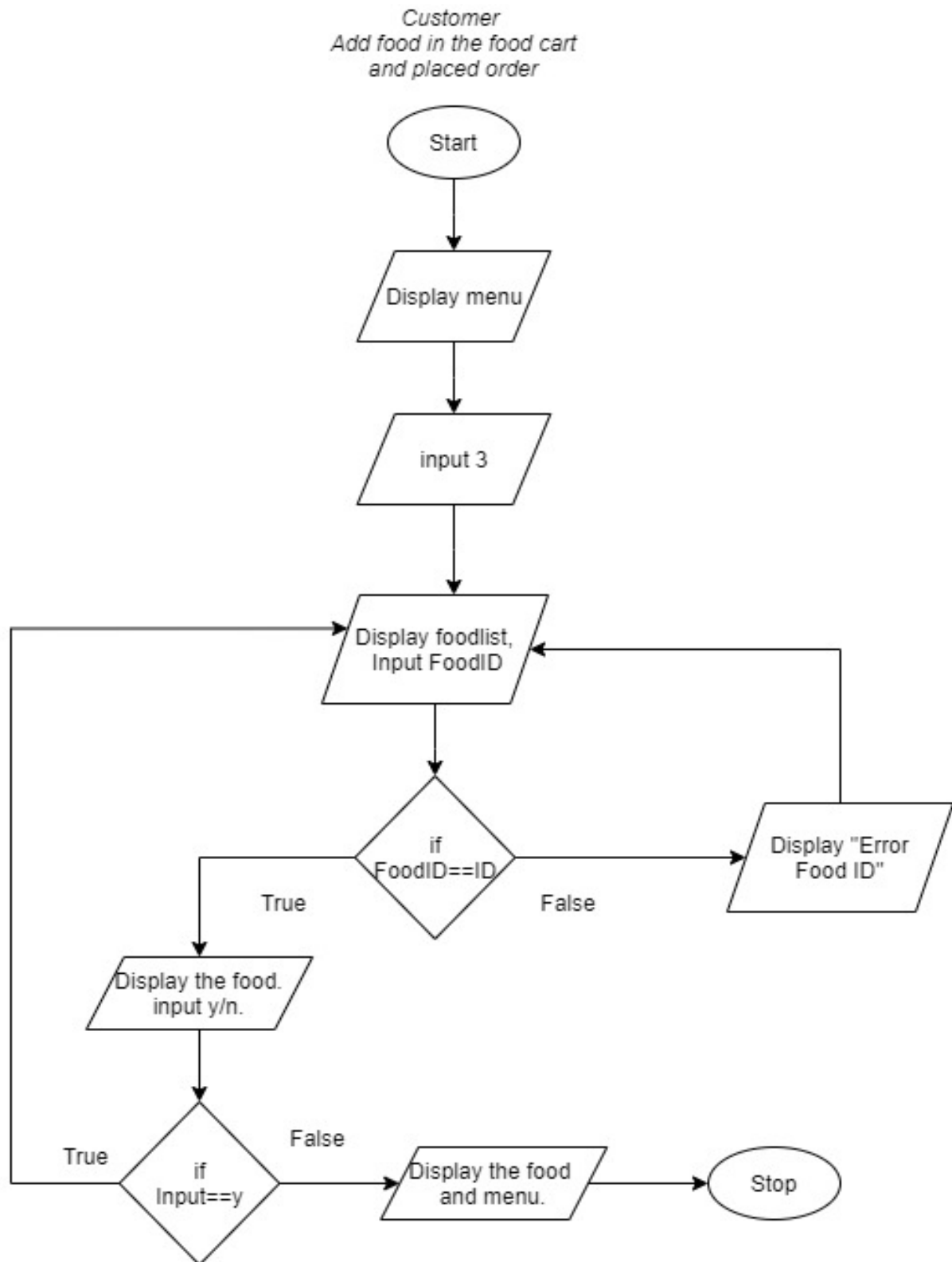
Flowchart 6: Delete Item from the Food List

Prepared by: Md Abu Saleh Nayeem



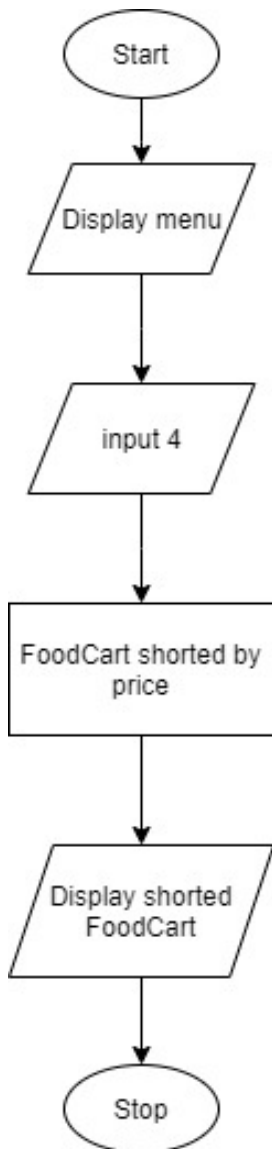
Flowchart 7: Add food in the cart

Prepared by: Md Abu Saleh Nayeem

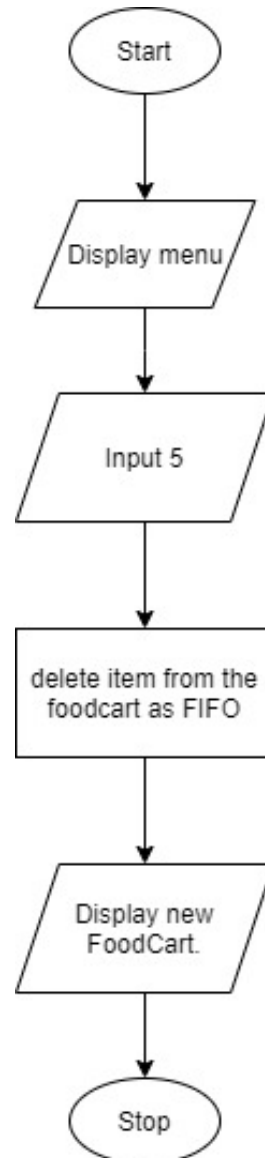


Flowchart 8: Sorting the foodcart

Prepared by: Md Abu Saleh Nayeem

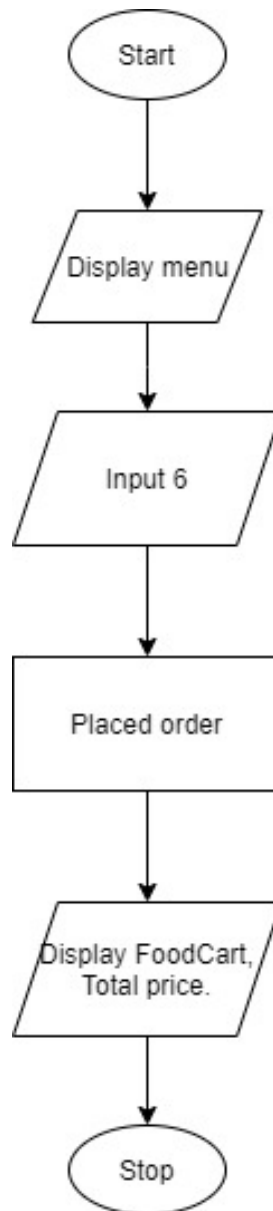


Flowchart 9: Delete food from foodcart



Flowchart 10: place order

Prepared by: Md Abu Saleh Nayeem



2.3 CLASS DIAGRAM

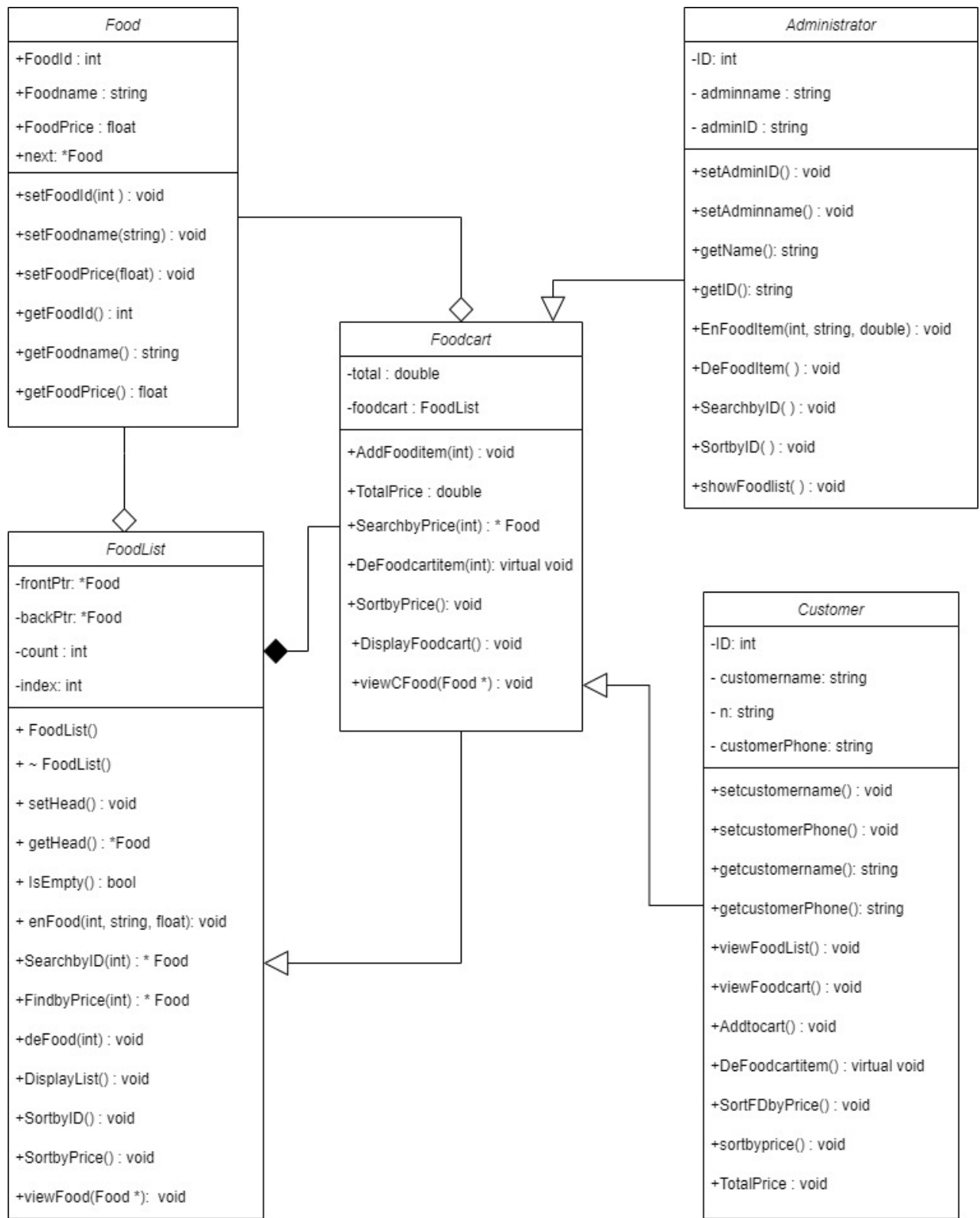


Figure 2: Class Diagram for Food Ordering System

3.0 SYSTEM PROTOTYPE

(Prepared by ANIKA SHAHZAD)

Our system has given the user 2 options, whether the user wants to have access as Administrator or Customer.

```
*****
|           WELCOME TO FOOD ORDERING SYSTEM           |
*****

DEAR USER, YOU WANT TO ACCESS AS :

      (1). Administrator
      (2). Customer
      (3). Exit

PLEASE CHOOSE AN OPTION-
```

Welcome Screen

Screen 1: The **Screen 1** shows the main menu of our system. The user will get the above screen when the user accesses our system. The user has to insert integer values from 1-3. The system will ask the user “**PLEASE CHOOSE AN OPTION**”, the user will enter 1 or 2 if the user wants to have access as **Administrator** or **Customer** respectively, and 3 if the user wants to **Exit** the system.

AS ADMINISTRATOR:

```
*****
|           WELCOME TO FOOD ORDERING SYSTEM           |
*****

DEAR USER, YOU WANT TO ACCESS AS :

      (1). Administrator
      (2). Customer
      (3). Exit

PLEASE CHOOSE AN OPTION- 1
```

Screen 1: Main Menu

Screen 1: To access as Administrator, the user has to choose option **1**.

```
ENTER YOUR NAME : ANIKA
ENTER YOUR STAFF ID-NO (DEFAULT:123456): 123456
```

Screen 2: Login As Admin

Screen 2: If the user will choose Option 1, the user will access as administrator. To login as administrator, the system asks the user to enter the user's name with the message “**ENTER YOUR NAME**” and staff ID-NO with the message “**ENTER YOUR STAFF ID-NO :**”.. The system is set to default(123456) for staff ID-NO.

```
ENTER YOUR NAME : ANIKA
ENTER YOUR STAFF ID-NO (DEFAULT:123456): 123456
-----
|  DEAR ADMIN, WE GIVE YOU A WARM WELCOME ^_^  |
-----

Dear Admin,
(1). Food List
(2). Insert Food to Food List as Queue
(3). Delete Food as Queue
(4). Search Food by ID
(5). Sort Food by ID
(6). Logout
(7). Exit
Choose an option-
```

Screen 3: Main Menu As Admin

Screen 3: After the successful login, the user will be welcomed with the message “**DEAR ADMIN, WE GIVE YOU A WARM WELCOME**” and will be also provided with the menu. The user has to insert integer values from 1-7. The system will ask the user to “**Choose an option**”.

CURRENT STOCK		
FOOD-ID	FOOD-NAME	FOOD-PRICE
10	ESPRESSO	5
82	NasiLemak	4.5
778	Nasi_lemak_Ayam	8.5
13	Nasi_Goreng_Ayam	6.5
14	Nasi_Goreng_Kampung	6
18	Nasi_Goreng_Dagging	8
16	Nasi_Goreng_Kambing	8.5
20	Nasi_Goreng_Biasa	4.5
17	Maggi_Goreng_Biasa	4
30	Maggi_Goreng_Ayam	7
29	Mee_Goreng_Biasa	4
21	Mee_Goreng_Ayam	7
22	Bihoon_Goreng_Biasa	3.5
19	Bihoon_Goreng_Ayam	6.5
28	Kuey_Teow_Goreng_Biasa	3.75
23	Kuey_Teow_Goreng_Ayam	7.75
31	Roti_Canai_Kosong	1
80	Roti_Teluer	2
32	Roti_Teluer_Bawang	2.5
41	Roti_Teluer_Bawang_Chili	3
42	Roti_Sambal	5
111	Teh_o	1
145	Teh_Tarik	2
121	Coffee_Panas	3
119	Coffee_Tarik	3.5
167	Teh_o_limau	1.5
165	Teh_O_ice	1.8
170	Teh_Ice	2.5
198	Milo_Ice	3.5
201	Siraf_Ice	1.7
197	coffee_ice	4
195	Nescafe_Ice	3.8
208	Sifar_Bundong_ice	2
511	Juice_Tambikai	4
523	Juice_Pisang	4
512	juice_carrot	4

Number of Foods in the FoodList:36

Screen 4: Food List

Screen 4: If the user chooses option **1**, the user will get the list of the food which is in the menu. The food list contains the **FOOD-ID**, **FOOD-NAME** and **FOOD-PRICE**.

```

-----
|               Insert Food to Food List               |
-----

Please Enter the Food ID : 23
Please Enter the Food Name : Kuey_Teow_Goreng_Ayam
Please Enter the Food Price : 7.75

```

Screen 5: Insert Food

Screen 5: The user as an admin is given the option to insert the food to the food list if he chooses option 2 from the admin's main menu. For inserting the food, the admin has to provide a food ID and food name with the messages "**Please Enter the Food ID :**" and "**Please Enter the Food Name :**" respectively and also the system will ask the user "**Please Enter the Food Price :**" to get the price of the food item.

523	Juice_Pisang	4
512	juice_carrot	4
23	Kuey_Teow_Goreng_Ayam	7.75
Number of Foods in the FoodList:37		

Screen 6: Food List

Screen 6: After providing the details system asked for, the food item will be added to the actual food menu. Therefore, the total food items will be 37, before insertion, there will be 36 items.

23	Kuey_Teow_Goreng_Ayam	7.75
Number of Foods in the FoodList:36		

Screen 7: Deleted Item

Screen 7: If the admin wants to delete the food items from the food list, the admin will choose the option 3. The food item then be deleted from the Food List.

Search Food by ID

Please enter the food ID :

Screen 8: Search Food by ID

Screen 8: The user has given the option to search the food by only its ID. For doing this, the user has to choose option 4 from the admin's main menu. The user will be asked "**Please enter the food ID :**".

Search Food by ID		
Please enter the food ID : 019		
FOOD-ID	FOOD-NAME	FOOD-PRICE
19	Bihoon_Goreng_Ayam	6.5

Screen 9: Food by ID

Screen 9: After entering the food item, the FOOD-ID following the FOOD-NAME and FOOD-PRICE will be displayed.

Sort Food by ID		
FOOD-ID	FOOD-NAME	FOOD-PRICE
10	ESPRESSO	5
13	Nasi_Goreng_Ayam	6.5
14	Nasi_Goreng_Kampung	6
16	Nasi_Goreng_Kambing	8.5
17	Maggi_Goreng_Biasa	4
18	Nasi_Goreng_Dagging	8
19	Bihoon_Goreng_Ayam	6.5
20	Nasi_Goreng_Biasa	4.5
21	Mee_Goreng_Ayam	7
22	Bihoon_Goreng_Biasa	3.5
23	Kuey_Teow_Goreng_Ayam	7.75
28	Kuey_Teow_Goreng_Biasa	3.75
29	Mee_Goreng_Biasa	4
30	Maggi_Goreng_Ayam	7
31	Roti_Canai_Kosong	1
32	Roti_Teluer_Bawang	2.5
41	Roti_Teluer_Bawang_Chili	3
42	Roti_Sambal	5
80	Roti_Teluer	2
82	Nasilemak	4.5
111	Teh_o	1
119	Coffee_Tarik	3.5
121	Coffee_Panas	3
145	Teh_Tarik	2
165	Teh_O_ice	1.8
167	Teh_o_limau	1.5
170	Teh_Ice	2.5
195	Nescafe_Ice	3.8
197	coffee_ice	4
198	Milo_Ice	3.5
201	Siraf_Ice	1.7
208	Sifar_Bundong_ice	2
511	Juice_Tambikai	4
512	juice_carrot	4
523	Juice_Pisang	4
778	Nasi_lemak_Ayam	8.5

Number of Foods in the FoodList:36

Screen 10: Sort Food by ID

Screen 10: The food list can be sorted by ID from choosing the option 5 from the admin's main menu. The list will be sorted numerically into ascending order.

```
Dear Admin,  
  (1). Food List  
  (2). Insert Food to Food List as Queue  
  (3). Delete Food as Queue  
  (4). Search Food by ID  
  (5). Sort Food by ID  
  (6). Logout  
  (7). Exit  
Choose an option- 6
```

Screen 11: Logout

```
Dear Admin,  
  (1). Food List  
  (2). Insert Food to Food List as Queue  
  (3). Delete Food as Queue  
  (4). Search Food by ID  
  (5). Sort Food by ID  
  (6). Logout  
  (7). Exit  
Choose an option- 7
```

Screen 12: Exit

Screen 11 and Screen 12: The user can easily logout as administrator by choosing option **6** from the admin's main menu. The user will get back to the system main menu where the user has been asked whether the user wants to login as administrator or customer. And the user can also exit by choosing the option 7 to for exiting the system as shown in Screen 12.

As Customer

```
*****
| WELCOME TO FOOD ORDERING SYSTEM |
*****

DEAR USER, YOU WANT TO ACCESS AS :

    (1). Administrator
    (2). Customer
    (3). Exit

PLEASE CHOOSE AN OPTION- 2
```

Screen 1: Main Menu

Screen 1: To access as Administrator, the user has to choose option 2.

```
-----
| DEAR CUSTOMER, WE GIVE YOU A WARM WELCOME ^_^ |
-----

ENTER YOUR NAME : ANIKA
ENTER YOUR PHONE NUMBER: 012345678
```

Screen 2: Login As User

Screen 2: If the user will choose Option 2, the user will access as a customer. To login as a customer, the system asks the user to enter the user's name with the message “**ENTER YOUR NAME**” and the user phone number with the message “**ENTER YOUR PHONE NUMBER :**”.

```
|-----|
|  DEAR CUSTOMER, WE GIVE YOU A WARM WELCOME  ^_^  |
|-----|

ENTER YOUR NAME : ANIKA
ENTER YOUR PHONE NUMBER: 012345678

Dear Customer,
(1). Food List
(2). Sort Food List By Price
(3). Add Item to FoodCart
(4). Sort FoodCart by Price
(5). Delete Food as Queue from FoodCart
(6). Place Order
(7). Logout
(8). Exit
Choose an option-
```

Screen 3: Main Menu as Customer

Screen 3: After the successful login, the user will be welcomed with the message “**DEAR CUSTOMER, WE GIVE YOU A WARM WELCOME**” and will be also provided with the menu. The user has to insert integer values from 1-8. The system will ask the user to “**Choose an option**” to proceed further.

CURRENT STOCK		
FOOD-ID	FOOD-NAME	FOOD-PRICE
10	ESPRESSO	5
82	Nasilemak	4.5
778	Nasi_lemak_Ayam	8.5
13	Nasi_Goreng_Ayam	6.5
14	Nasi_Goreng_Kampung	6
18	Nasi_Goreng_Dagging	8
16	Nasi_Goreng_Kambing	8.5
20	Nasi_Goreng_Biasa	4.5
17	Maggi_Goreng_Biasa	4
30	Maggi_Goreng_Ayam	7
29	Mee_Goreng_Biasa	4
21	Mee_Goreng_Ayam	7
22	Bihoon_Goreng_Biasa	3.5
19	Bihoon_Goreng_Ayam	6.5
28	Kuey_Teow_Goreng_Biasa	3.75
23	Kuey_Teow_Goreng_Ayam	7.75
31	Roti_Canai_Kosong	1
80	Roti_Teluer	2
32	Roti_Teluer_Bawang	2.5
41	Roti_Teluer_Bawang_Chili	3
42	Roti_Sambal	5
111	Teh_o	1
145	Teh_Tarik	2
121	Coffee_Panas	3
119	Coffee_Tarik	3.5
167	Teh_o_limau	1.5
165	Teh_O_ice	1.8
170	Teh_Ice	2.5
198	Milo_Ice	3.5
201	Siraf_Ice	1.7
197	coffee_ice	4
195	Nescafe_Ice	3.8
208	Sifar_Bundong_ice	2
511	Juice_Tambikai	4
523	Juice_Pisang	4
512	juice_carrot	4

Number of Foods in the FoodList:36

Screen 4: Food List

Screen 4: If the user chooses option **1** from the customer's main menu, the user will get the list of the food which is in the menu. The food list contains the **FOOD-ID**, **FOOD-NAME** and **FOOD-PRICE**.

Sort Food by Price		
FOOD-ID	FOOD-NAME	FOOD-PRICE
31	Roti_Canai_Kosong	1
111	Teh_o	1
167	Teh_o_limau	1.5
201	Siraf_Ice	1.7
165	Teh_O_ice	1.8
80	Roti_Teluer	2
145	Teh_Tarik	2
208	Sifar_Bundong_ice	2
32	Roti_Teluer_Bawang	2.5
170	Teh_Ice	2.5
41	Roti_Teluer_Bawang_Chili	3
121	Coffee_Panas	3
22	Bihoon_Goreng_Biasa	3.5
119	Coffee_Tarik	3.5
198	Milo_Ice	3.5
28	Kuey_Teow_Goreng_Biasa	3.75
195	Nescafe_Ice	3.8
17	Maggi_Goreng_Biasa	4
29	Mee_Goreng_Biasa	4
197	coffee_ice	4
511	Juice_Tambikai	4
523	Juice_Pisang	4
512	juice_carrot	4
82	Nasilemak	4.5
20	Nasi_Goreng_Biasa	4.5
10	ESPRESSO	5
42	Roti_Sambal	5
14	Nasi_Goreng_Kampung	6
13	Nasi_Goreng_Ayam	6.5
19	Bihoon_Goreng_Ayam	6.5
30	Maggi_Goreng_Ayam	7
21	Mee_Goreng_Ayam	7
23	Kuey_Teow_Goreng_Ayam	7.75
18	Nasi_Goreng_Dagging	8
778	Nasi_lemak_Ayam	8.5
16	Nasi_Goreng_Kambing	8.5

Number of Foods in the FoodList:36

Screen 5: Sort Food by Price

Screen 5: The user as a customer has given the option to sort the food items according to their prices from low to high. For doing this, the user has to choose the option 2 from the customer's main menu.

```
Number of Foods in the FoodList:36

Enter the FoodID to add the food in the FoodCart :
```

Screen 6: Add Item to Food Cart

Screen 6: To make a purchase, the customer has to put the items in the cart. Option 3 in the customer's main menu provides this facility to users. The user will key in the food ID after getting the message "Enter the FoodID to add the food in the FoodCart :".

```
Enter the FoodID to add the food in the FoodCart : 512
      FOOD-ID          FOOD-NAME          FOOD-PRICE
-----
      512              juice_carrot          4
Number of Foods in the FoodList:1

Do you want Add more Food? (y/n) :
```

Screen 7: FoodCart

Screen 7: After entering the FoodID, the food will be added to the FoodCart. The system will show the quantity of food which is in the cart with the **FOOD-ID**, **FOOD-NAME** and **FOOD-PRICE**. The system will again ask the user whether they wanted to add more in the cart or not, the customer has to reply with the yes or no represented with **y** or **n** respectively, after showing the message “**Do you want Add more Food? (y/n) :**”.

```
Do you want Add more Food? (y/n) : y
Enter the FoodID to add the food in the FoodCart : 10
      FOOD-ID          FOOD-NAME          FOOD-PRICE
-----
      512              juice_carrot          4
      10                ESPRESSO            5
Number of Foods in the FoodList:2

Do you want Add more Food? (y/n) :
```

Screen 8: FoodCart

Screen 8: If the customer wants to add more items, the user responds with y and another item is added to the cart. Now, the total items are 2 in FoodCart.

```
Do you want Add more Food? (y/n) : n
FOOD-ID          FOOD-NAME          FOOD-PRICE
-----
512              juice_carrot         4
10              ESPRESSO             5
Number of Foods in the FoodList:2

Dear Customer,
(1). Food List
(2). Sort Food List By Price
(3). Add Item to FoodCart
(4). Sort FoodCart by Price
(5). Delete Food as Queue from FoodCart
(6). Place Order
(7). Logout
(8). Exit
Choose an option-
```

Screen 9: FoodCart

Screen 9: When the user with the selection wants to finalize the order, the customer will reply with **n** after getting the message “**Do you want Add more Food? (y/n) :**”. After doing so, the user will again go to the main menu of the customer.

```
| Sort FoodCart by Price |
-----
FOOD-ID          FOOD-NAME          FOOD-PRICE
-----
512              juice_carrot         4
10              ESPRESSO             5
Number of Foods in the FoodList:2

Dear Customer,
(1). Food List
(2). Sort Food List By Price
(3). Add Item to FoodCart
(4). Sort FoodCart by Price
(5). Delete Food as Queue from FoodCart
(6). Place Order
(7). Logout
(8). Exit
Choose an option-
```

Screen 10: Sort FoodCart by Price

Screen 10: The user can easily sort the items in the FoodCart by Price through entering the option 4 from the customers main menu. Hence, the items in the FoodCart will be sorted according to price from low to high.

FOOD-ID	FOOD-NAME	FOOD-PRICE
10	ESPRESSO	5

Number of Foods in the FoodList:1

Dear Customer,

- (1). Food List
- (2). Sort Food List By Price
- (3). Add Item to FoodCart
- (4). Sort FoodCart by Price
- (5). Delete Food as Queue from FoodCart
- (6). Place Order
- (7). Logout
- (8). Exit

Choose an option-

Screen 11: Delete Food from FoodCart

Screen 11: The customer can easily make changes to the order. If the customer wants to delete any item from the FoodCart, the customer has to select the option **5**. The item will be deleted from the cart.

Place Order		
Customer Name : ANIKA		
Customer Phone Number : 0123456789		
FOOD-ID	FOOD-NAME	FOOD-PRICE
10	ESPRESSO	5

Number of Foods in the FoodList:1

Total Price : 5

Dear Customer,

- (1). Food List
- (2). Sort Food List By Price
- (3). Add Item to FoodCart
- (4). Sort FoodCart by Price
- (5). Delete Food as Queue from FoodCart
- (6). Place Order
- (7). Logout
- (8). Exit

Choose an option-

Screen 12: Place Order

Screen 12: The customer has to choose option 6 to place an order. In the interface of Place Order, the system will provide Customer Name Customer Phone Number, FOOD-ID, FOOD-NAME, FOOD-PRICE. The quantity of food with price will also be displayed. In the above case, Customer Name is ANIKA following the Customer Phone Number which is 0123456789, FOOD-ID is 10, FOOD-NAME and FOOD-PRICE is ESPRESSO and FOOD-PRICE respectively.

```
Dear Customer,  
  (1). Food List  
  (2). Sort Food List By Price  
  (3). Add Item to FoodCart  
  (4). Sort FoodCart by Price  
  (5). Delete Food as Queue from FoodCart  
  (6). Place Order  
  (7). Logout  
  (8). Exit  
Choose an option- 8
```

Screen 13: Logout

```
Dear Customer,  
  (1). Food List  
  (2). Sort Food List By Price  
  (3). Add Item to FoodCart  
  (4). Sort FoodCart by Price  
  (5). Delete Food as Queue from FoodCart  
  (6). Place Order  
  (7). Logout  
  (8). Exit  
Choose an option- 8
```

Screen 14: Exit

Screen 13 and **Screen 14:** After the successful placement of order, the user can easily logout with pressing option **7** and if he wants to directly exit the system, the user will press option **8**.

4.0 DEVELOPMENT ACTIVITIES

Meeting Date	Members Participate in the meeting	Activity	Task for each member	Task Achieved (Yes/No)
18th January, 2021	MD MONIRUL ISLAM MOLLA	Task Distribution	Coding	Yes
	MUHAMMAD SHAFAYET HOSSAIN		Slide	Yes
	MD.ABU SALEH NAYEEM		Report	Yes
	ANIKA SHAHZAD		Report	Yes

Meeting Date	Members Participate in the meeting	Activity	Task for each member	Task Achieved (Yes/No)
24th January, 2021	MD MONIRUL ISLAM MOLLA	Task Distribution	Slide/Record/Use Case Diagram and UML	Yes
	MUHAMMAD SHAFAYET HOSSAIN		Coding/Record	Yes
	MD.ABU SALEH NAYEEM		Record/Report/ Algorithms/Flowchart	Yes
	ANIKA SHAHZAD		Record/Report/ Introduction/Prototype	Yes

5.0 APPENDIX

5.1 SOURCE CODE

```
1. ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2. // School of Computing, Faculty of Engineering, UTM
3. // SCSJ2013- Data Structures and Algorithms
4. // Semester 1, 2019/2020
5. // Mini Project - Food Ordering System (QUEUE)
6. //MD MONIRUL ISLAM MOLLA (A19EC4013)
7. //MUHAMMAD SHAFAYET HOSSAIN (A19EC4016)
8. //MD ABU SALEH NAYEEM (A19EC4012)
9. //ANIKA SHAHZAD (A19EC4007)
10. ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
11.
12.
13. #include <iostream>
14. #include <fstream>
15. #include <string>
16. #include <iomanip>
17. #include <vector>
18. #include <ctime>
19. #include <cstdlib>
20.
21. using namespace std;
22.
23. const int SIZE=50;
24.
25.
26. class Food
27. {
28. public:
29.     int FoodId;
30.     string Foodname;
31.     float FoodPrice;
32.     Food* next;
33.
34.     void setFoodId(int FID)
35.     { FoodId = FID;}
36.
37.     void setFoodname(string Fname)
38.     { Foodname = Fname;}
39.     void setFoodPrice(float Fprice)
40.     { FoodPrice = Fprice; }
41.
42.     int getFoodId()
43.     {return FoodId;}
44.
45.     string getFoodname()
46.     { return Foodname;}
47.
48.     float getFoodPrice()
49.     { return FoodPrice;}
50. };
51.
52.
53.
54. class FoodList
55. {
56. private:
57.     int count=0;
```



```

58.     int index=-1;
59.     Food* frontPtr;
60.     Food* backPtr;
61.
62. public:
63.     FoodList() { frontPtr= backPtr= NULL;}
64.     ~FoodList()
65.     {
66.         Food* currFood = frontPtr, *nextNode = NULL;
67.
68.         while (currFood != NULL)
69.         {
70.             nextNode = currFood->next;
71.             delete currFood;
72.             currFood = nextNode;
73.         }
74.     }
75.
76.     void setHead (Food*f)
77.     {
78.         frontPtr=f;
79.     }
80.     Food *getHead ()
81.     {
82.         return frontPtr;
83.     }
84.
85.     bool IsEmpty()
86.     {
87.         return frontPtr== NULL;
88.     }
89.
90.     /*void enFood(int FId, string Fname, float FPrice)
91.     {
92.         Food* currFood = frontPtr;
93.
94.         Food* newFood = new Food;
95.         newFood->FoodId= FId;
96.         newFood->Foodname= Fname;
97.         newFood->FoodPrice= FPrice;
98.
99.         newFood->next = frontPtr;
100.        frontPtr = newFood;
101.
102.        }*/
103.
104.        void enFood(int FId, string Fname, float FPrice)
105.        {
106.            Food* newFood = new Food;
107.            newFood->FoodId= FId;
108.            newFood->Foodname= Fname;
109.            newFood->FoodPrice= FPrice;
110.            newFood->next=NULL;
111.
112.            if(frontPtr==NULL)
113.            {
114.                frontPtr=backPtr=newFood;
115.            }
116.            else
117.            {
118.                Food* currFood = backPtr;
119.                currFood->next=newFood;
120.                backPtr=newFood;
121.            }
122.
123.        }
124.

```

```

125.
126.         Food * SearchbyID(int x)
127.         {
128.             Food* currFood=frontPtr;
129.             int currIndex= 1;
130.
131.             while (currFood && currFood->FoodId!=x)
132.             {
133.                 currFood=currFood->next;
134.                 currIndex++;
135.             }
136.             if (currFood)
137.                 return currFood;
138.             else
139.                 return 0;
140.         }
141.
142.         Food * FindbyPrice(int x)
143.         {
144.             Food* currFood=frontPtr;
145.             int currIndex= 1;
146.
147.             while (currFood && currFood->FoodPrice!=x)
148.             {
149.                 currFood=currFood->next;
150.                 currIndex++;
151.             }
152.             if (currFood)
153.                 return currFood;
154.             else
155.                 return 0;
156.         }
157.     /*
158.     void deFood(int x)
159.     {
160.         Food* prevNode= NULL;
161.         Food* currFood = frontPtr;
162.         int currIndex=1;
163.
164.         while (currFood && currFood->FoodId !=x)
165.         {
166.             prevNode = currFood;
167.             currFood = currFood->next;
168.             currIndex++;
169.         }
170.
171.         if (prevNode)
172.         {
173.             prevNode->next = currFood->next;
174.             delete currFood;
175.         }
176.         else
177.         {
178.             frontPtr = currFood->next;
179.             delete currFood;
180.         }
181.
182.     }
183. */
184.     void deFood()
185.     {
186.
187.         if(frontPtr==NULL)
188.         {
189.             cout<<"\nPlease enqueue item first, there is no item to delete.
190.             \n\n";
191.         }

```

```

191.         else
192.         {
193.
194.             Food* currFood = frontPtr;
195.
196.             frontPtr=currFood->next;
197.             currFood->next=NULL;
198.             cout<<"Deleted Food : \n"
199.             << "FOOD-ID      : " <<currFood->FoodId
200.             << "\nFOOD-NAME  : " << currFood->Foodname
201.             << "\nFOOD-PRICE : " << currFood->FoodPrice<< endl
202.             <<"-----\n\n";
203.
204.
205.
206.             delete currFood;
207.         }
208.
209.     }
210.
211.
212.     void DisplayList()
213.     {
214.         int num = 0;
215.         Food* currFood = frontPtr;
216.
217.         cout<< setw(12)<< "FOOD-ID" << setw(35) << "FOOD-NAME" << setw(27)
218.         << "FOOD-PRICE\n" <<"-----\n\n";
219.
220.         while (currFood != NULL)
221.         {
222.             cout << setw(10)<< currFood->FoodId << setw(40)
223.             << currFood->Foodname<< setw(20)
224.             << currFood->FoodPrice<< endl;
225.
226.             currFood = currFood->next;
227.             num++;
228.         }
229.
230.         cout<<"Number of Foods in the FoodList:"<<num<<endl;
231.     }
232.
233.     void SortbyID()
234.     {
235.         Food* curr = frontPtr;
236.         int count = 0;
237.         while(curr!=NULL)
238.         {
239.             count++;
240.             curr = curr->next;
241.         }
242.         for(int i = count ; i > 1 ; i-- )
243.         {
244.             Food *temp, *swap1;
245.             swap1 = frontPtr;
246.             for(int j = 0 ; j < count-1 ; j++ )
247.             {
248.                 if(swap1->getFoodId() > swap1->next->getFoodId())
249.                 {
250.                     Food *swap2 = swap1->next;
251.                     swap1->next = swap2->next;
252.                     swap2->next = swap1;
253.                     if(swap1 == frontPtr)
254.                     {

```

```

255.                 frontPtr = swap2;
256.                 swap1 = swap2;
257.             }
258.             else
259.             {
260.                 swap1 = swap2;
261.                 temp->next = swap2;
262.             }
263.         }
264.         temp = swap1;
265.         swap1 = swap1->next;
266.     }
267. }
268. }
269.
270. void SortbyPrice()
271. {
272.     Food* curr = frontPtr;
273.     while(curr!=NULL)
274.     {
275.         count++;
276.         curr = curr->next;
277.     }
278.     for(int i = count ; i > 1 ; i-- )
279.     {
280.         Food *temp, *swap1;
281.         swap1 = frontPtr;
282.         for(int j = 0 ; j < count-1 ; j++ )
283.         {
284.             if(swap1->getFoodPrice() > swap1->next->getFoodPrice())
285.             {
286.                 Food *swap2 = swap1->next;
287.                 swap1->next = swap2->next;
288.                 swap2->next = swap1;
289.                 if(swap1 == frontPtr)
290.                 {
291.                     frontPtr = swap2;
292.                     swap1 = swap2;
293.                 }
294.                 else
295.                 {
296.                     swap1 = swap2;
297.                     temp->next = swap2;
298.                 }
299.             }
300.             temp = swap1;
301.             swap1 = swap1->next;
302.         }
303.     }
304. }
305.
306. void viewFood(Food *food)
307. {
308.
309.     cout<< setw(15) << "FOOD-ID" << setw(35) << "FOOD-NAME" << setw(25)
    << "FOOD-PRICE\n";
310.     cout << setw(10) << food->getFoodId() << setw(40)
    << food->getFoodname()<< setw(20)
311.     << food->getFoodPrice() << endl;
312.
313. }
314.
315. };
316.
317.
318.
319. class Foodcart : public FoodList
320. {

```

```

321.     private:
322.
323.         double total=0.0;
324.         FoodList foodcart;
325.
326.     public:
327.
328.         void AddFooditem(int fid)
329.         {
330.             Food *food = SearchbyID(fid);
331.             total+=food->getFoodPrice();
332.
333.             foodcart.enFood(food->getFoodId(), food->getFoodname(), food-
>getFoodPrice());
334.         }
335.
336.         double TotalPrice()
337.         {
338.             return total;
339.         }
340.         Food * SearchbyPrice(int x)
341.         {
342.             return FoodList::FindbyPrice(x);
343.         }
344.
345.         virtual void DeFoodcartitem()
346.         {
347.
348.             total-=foodcart.getHead()->getFoodPrice();
349.             foodcart.deFood();
350.         }
351.
352.         void SortbyPrice()
353.         {
354.             foodcart.SortbyPrice();
355.         }
356.
357.         void DisplayFoodcart()
358.         {
359.             foodcart.DisplayList();
360.         }
361.
362.         void viewCFood(Food *i)
363.         {
364.             viewFood(i);
365.         }
366.
367.     };
368.
369.
370.     class Administrator : public Foodcart
371.     {
372.     private:
373.
374.
375.         int ID;
376.         string adminID;
377.         string adminname;
378.
379.     public:
380.
381.         void setAdminID()
382.         {
383.             A: cout << "ENTER YOUR STAFF ID-NO (DEFAULT:123456): ";
384.             cin >> adminID;
385.
386.             if ((adminID == "123456") || (adminID == "223456") || (adminID ==

```

```

    "323456") || (adminID == "423456") || (adminID == "23456"))
387.         {
388.         }
389.
390.         else
391.         {
392.             cout << "\nInput Error! Please try again.\nIf you don't have
staff ID, contact our office. \n";
393.             goto A;
394.         }
395.
396.     }
397.
398.     void setadminname()
399.     {
400.         cout << "ENTER YOUR NAME : ";
401.         cin.ignore();
402.         getline(cin,adminname);
403.     }
404.
405.     string getName()
406.     {
407.         return adminname;
408.     }
409.
410.     string getID()
411.     {
412.         return adminID;
413.     }
414.
415.     void EnFoodItem(int FoodId,string Foodname,double FoodPrice)
416.     {
417.         FoodList::enFood(FoodId,Foodname,FoodPrice);
418.
419.     }
420.
421.     void DeFoodItem()
422.     {
423.
424.
425.         FoodList::deFood();
426.     }
427.
428.     void SearchbyID()
429.     {
430.         cout<<"\nPlease enter the food ID : ";
431.         cin>>ID;
432.         Food *food = FoodList::SearchbyID(ID);
433.
434.         if (food==0)
435.         {
436.             cout<<" The Food Item is not in the FoodList. \n";
437.         }
438.         else
439.         {
440.             FoodList::viewFood(food);
441.         }
442.     }
443.
444.     void SortbyID()
445.     {
446.         FoodList::SortbyID();
447.     }
448.
449.     void showFoodlist()
450.     {
451.         FoodList::DisplayList();

```

```

452.         }
453.
454.     };
455.
456.     class Customer : public Foodcart
457.     {
458.     private:
459.         int ID;
460.         string customername;
461.         string n;
462.         string customerPhone;
463.
464.     public:
465.
466.         void setcustomername()
467.         {
468.             cout << "ENTER YOUR NAME : ";
469.             cin.ignore();
470.             getline(cin, customername);
471.         }
472.
473.         void setcustomerPhone()
474.         {
475.             cout << "ENTER YOUR PHONE NUMBER: ";
476.             cin >> customerPhone;
477.         }
478.
479.         string getcustomername()
480.         {
481.             return customername;
482.         }
483.
484.         string getcustomerPhone()
485.         {
486.             return customerPhone;
487.         }
488.
489.         void viewFoodList()
490.         {
491.             FoodList::DisplayList();
492.         }
493.
494.         void viewFoodcart()
495.         {
496.             Foodcart::DisplayFoodcart();
497.         }
498.
499.         void Addtocart()
500.         {
501.
502.
503.             A: cout<<" Enter the FoodID to add the food in the FoodCart : " ;
504.             cin>> ID;
505.             Food *food =FoodList::SearchbyID(ID);
506.             if(food==0)
507.             {
508.                 cout<<"Error FoodId. \n";
509.                 goto A;
510.             }
511.             Foodcart::AddFooditem(ID);
512.
513.             Foodcart::DisplayFoodcart();
514.             cout<<" \n Do you want Add more Food? (y/n) : " ;
515.             cin>> n;
516.
517.
518.

```

```

519.         if (n=="y")
520.             goto A;
521.     }
522.
523.     virtual void DeFoodcartitem()
524.     {
525.         Foodcart::DeFoodcartitem();
526.     }
527.
528.     void SortFDbyPrice()
529.     {
530.         FoodList::SortbyPrice();
531.     }
532.
533.     void sortbyprice()
534.     {
535.         Foodcart::SortbyPrice();
536.     }
537.
538.     void TotalPrice()
539.     {
540.         cout<<"Total Price : " <<Foodcart::TotalPrice()<<endl;
541.     }
542.
543. };
544.
545.
546.
547. int main()
548. {
549.     cout << "*****" << endl
550.     << "|          WELCOME TO FOOD ORDERING SYSTEM          |" << endl
551.     << "*****" << endl
552.     << endl;
553.
554.     ifstream input("FoodList.txt", ios::in);
555.     if (!input)
556.     {
557.         cout<<" Error : The file Can not read. \n" ;
558.         exit(0);
559.     }
560.
561.     Administrator admin;
562.     Customer cust;
563.
564.     int FoodId , count=0;
565.     string Foodname;
566.     float FoodPrice;
567.     int ch;
568.     int choice;
569.
570.     while(!input.eof())
571.     {
572.         input>>FoodId ;
573.         input>> Foodname;
574.         input >> FoodPrice;
575.
576.         admin.enFood(FoodId,Foodname,FoodPrice);
577.
578.     }
579.
580.
581.     A:
582.     cout << "\nDEAR USER, YOU WANT TO ACCESS AS :\n\n"
583.     << "    \t(1). Administrator \n"
584.     << "    \t(2). Customer\n"
585.     << "    \t(3). Exit  \n\n"

```



```

586.         << "PLEASE CHOOSE AN OPTION- ";
587.     cin >> choice;
588.     std::system("cls");
589.
590.     if (choice == 1)
591.     {
592.         admin.setadminname();
593.         admin.setAdminID();
594.         cout << "-----" << endl
595.             << "|    DEAR ADMIN, WE GIVE YOU A WARM WELCOME ^_^    |" << endl
596.             << "-----" << endl
597.             << endl;
598.
599.     C:
600.         cout << "\n  Dear Admin,\n"
601.             << "      (1). Food List \n"
602.             << "      (2). Insert Food to Food List as Queue\n"
603.             << "      (3). Delete Food as Queue\n"
604.             << "      (4). Search Food by ID  \n"
605.             << "      (5). Sort Food by ID \n"
606.             << "      (6). Logout \n"
607.             << "      (7). Exit  \n"
608.             << "      Choose an option- ";
609.         cin >> ch;
610.
611.         if (ch == 1)
612.         {
613.             std::system("cls");
614.             cout << "\t-----" <<
endl
615.                 << "\t|                                |" <<
endl
616.                 << "\t-----" <<
endl
617.                 << endl;
618.
619.             admin.showFoodlist();
620.             goto C;
621.         }
622.
623.     else if (ch == 2)
624.     {
625.         std::system("cls");
626.         cout<< "-----" <<
endl
627.             << "|                                Insert Food to Food List                                |" <<
endl
628.             << "-----" <<
endl
629.             << endl;
630.
631.         cout<<" Please Enter the Food ID : ";
632.         cin>>FoodId;
633.         cout<<" Please Enter the Food Name : ";
634.         cin.ignore();
635.         getline(cin,Foodname);
636.         cout<<" Please Enter the Food Price : ";
637.         cin>>FoodPrice;
638.
639.         admin.EnFoodItem(FoodId,Foodname,FoodPrice);
640.         admin.showFoodlist();
641.         cout<<endl<<endl;
642.         goto C;
643.     }
644.
645.     else if (ch == 3)
646.     {

```

```

647.         std::system("cls");
648.         cout<< "-----" <<
        endl
649.         << "|                Delete Food                |" <<
        endl
650.         << "-----" <<
        endl
651.         << endl;
652.
653.         admin.DeFoodItem();
654.         admin.showFoodlist();
655.         cout<<endl<<endl;
656.
657.         goto C;
658.     }
659.
660.     else if (ch == 4)
661.     {
662.         std::system("cls");
663.         cout<< "-----" <<
        endl
664.         << "|                Search Food by ID                |" <<
        endl
665.         << "-----" <<
        endl
666.         << endl;
667.
668.         admin.SearchbyID();
669.
670.         goto C;
671.     }
672.
673.     else if (ch == 5)
674.     {
675.         std::system("cls");
676.         cout<< "\t-----" <<
        endl
677.         << "\t|                Sort Food by ID                |" <<
        endl
678.         << "\t-----" <<
        endl
679.         << endl;
680.         admin.SortbyID();
681.         admin.showFoodlist();
682.         cout<<endl<<endl;
683.         goto C;
684.     }
685.
686.     else if (ch == 6)
687.     {
688.         std::system("cls");
689.         goto A;
690.     }
691.
692.     else if (ch == 7)
693.     {
694.         exit(0);
695.     }
696.
697.     else
698.     {
699.         cout << "\n Input Error 0_0  !\n";
700.         goto C;
701.     }
702. }
703.
704. //customer part start

```

```

705.
706.         else if (choice == 2)
707.         {
708.             cust.setHead(admin.getHead());
709.             cout << "-----"
710.             << endl
711.             << "|          DEAR CUSTOMER, WE GIVE YOU A WARM WELCOME ^_^"
712.             << endl
713.             << "-----"
714.             << endl;
715.             cust.setcustomername();
716.             cust.setcustomerPhone();
717.         D:
718.             cout << "\n  Dear Customer,\n"
719.             << "      (1). Food List \n"
720.             << "      (2). Sort Food List By Price \n"
721.             << "      (3). Add Item to FoodCart \n"
722.             << "      (4). Sort FoodCart by Price \n"
723.             << "      (5). Delete Food as Queue from FoodCart  \n"
724.             << "      (6). Place Order \n"
725.             << "      (7). Logout \n"
726.             << "      (8). Exit  \n"
727.             << "      Choose an option- ";
728.             cin >> ch;
729.
730.             if (ch == 1)
731.             {
732.
733.                 std::system("cls");
734.                 cout << "\t-----" <<
735.                 << endl
736.                 << "\t|          CURRENT STOCK          |" <<
737.                 << endl
738.                 << "\t-----" <<
739.                 << endl;
740.                 cust.viewFoodList();
741.                 goto D;
742.             }
743.
744.             else if (ch == 2)
745.             {
746.
747.                 std::system("cls");
748.                 cout<< "-----" <<
749.                 << endl
750.                 << "<<\"|          Sort Food by Price          |\" <<
751.                 << endl
752.                 << "-----" <<
753.                 << endl;
754.                 cust.SortFDbyPrice();
755.                 cust.viewFoodList();
756.                 cout<<endl<<endl;
757.                 goto D;
758.             }
759.
760.             else if (ch == 3)
761.             {
762.                 std::system("cls");

```

```

763.                cout<< "-----" <<
endl
764.                << "|          Add Item to FoodCart          |" <<
endl
765.                << "-----" <<
endl
766.                << endl;
767.
768.                cust.viewFoodList();
769.                cout<<endl<<endl;
770.                cust.Addtocart();
771.                cust.DisplayFoodcart();
772.                goto D;
773.            }
774.
775.            else if (ch == 4)
776.            {
777.                std::system("cls");
778.                cout<< "-----" <<
endl
779.                <<"|          Sort FoodCart by Price          |" <<
endl
780.                <<"-----" <<
endl
781.                << endl;
782.                cust.sortbyprice();
783.                cust.viewFoodcart();
784.                goto D;
785.            }
786.
787.            else if (ch == 5)
788.            {
789.                cout<< "-----" <<
endl
790.                << "|          Delete Food from FoodCart          |" <<
endl
791.                << "-----" <<
endl
792.                << endl;
793.
794.                cust.DeFoodcartitem();
795.                cust.viewFoodcart();
796.                cout<<endl<<endl;
797.
798.                goto D;
799.            }
800.
801.            else if (ch == 6)
802.            {
803.                std::system("cls");
804.                cout<< "\t-----" <<
endl
805.                <<"\t|          Place Order          |" <<
endl
806.                <<"\t-----" <<
endl
807.                << endl;
808.                cout<<"Customer Name : " << cust.getcustomername()<<endl;
809.                cout<<"Customer Phone Number : "
<<cust.getcustomerPhone()<<endl<<endl;
810.                cust.viewFoodcart();
811.                cout<<endl;
812.                cust.TotalPrice();
813.                cout<<endl;
814.                goto D;
815.            }
816.

```

```

817.         else if (ch == 7)
818.         {
819.             std::system("cls");
820.             goto A;
821.         }
822.
823.         else if (ch == 8)
824.         {
825.             exit(0);
826.         }
827.         else
828.         {
829.             cout << "\n Input Error 0_0  !\n";
830.             goto D;
831.         }
832.     }
833.     else if (choice==3)
834.     {
835.         exit(0);
836.     }
837.     else
838.     {
839.         cout << "\n Input Error 0_0  !\n";
840.         goto A;
841.     }
842.
843.     return 0;
844. }

```

5.2 DATA FILE

```

010 ESPRESSO 5.00
082 NasiLemak 4.50
778 Nasi_lemak_Ayam 8.50
013 Nasi_Goreng_Ayam 6.50
014 Nasi_Goreng_Kampung 6.00
018 Nasi_Goreng_Dagging 8.00
016 Nasi_Goreng_Kambing 8.50
020 Nasi_Goreng_Biasa 4.50
017 Maggi_Goreng_Biasa 4.00
030 Maggi_Goreng_Ayam 7.00
029 Mee_Goreng_Biasa 4.00
021 Mee_Goreng_Ayam 7.00
022 Bihoon_Goreng_Biasa 3.50
019 Bihoon_Goreng_Ayam 6.50
028 Kuey_Teow_Goreng_Biasa 3.75
023 Kuey_Teow_Goreng_Ayam 7.75
031 Roti_Canai_Kosong 1.00
080 Roti_Teluer 2.00
032 Roti_Teluer_Bawang 2.50
041 Roti_Teluer_Bawang_Chili 3.00
042 Roti_Sambal 5.00
111 Teh_o 1.00
145 Teh_Tarik 2.00
121 Coffee_Panas 3.00
119 Coffee_Tarik 3.50
167 Teh_o_limau 1.50
165 Teh_O_ice 1.80
170 Teh_Ice 2.50
198 Milo_Ice 3.50

```

201 Siraf_Ice 1.70
197 coffee_ice 4.00
195 Nescafe_Ice 3.80
208 Sifar_Bundong_ice 2.00
511 Juice_Tambikai 4.00
523 Juice_Pisang 4.00
512 juice_carrot 4.00

5.3 OUTPUT

FOR ADMINISTRATOR

```
*****  
|           WELCOME TO FOOD ORDERING SYSTEM           |  
*****  
  
DEAR USER, YOU WANT TO ACCESS AS :  
  
    (1). Administrator  
    (2). Customer  
    (3). Exit  
  
PLEASE CHOOSE AN OPTION-
```

```
*****  
|           WELCOME TO FOOD ORDERING SYSTEM           |  
*****  
  
DEAR USER, YOU WANT TO ACCESS AS :  
  
    (1). Administrator  
    (2). Customer  
    (3). Exit  
  
PLEASE CHOOSE AN OPTION- 1
```

```
ENTER YOUR NAME : ANIKA
ENTER YOUR STAFF ID-NO (DEFAULT:123456): 123456
```

```
ENTER YOUR NAME : ANIKA
ENTER YOUR STAFF ID-NO (DEFAULT:123456): 123456
-----
|   DEAR ADMIN, WE GIVE YOU A WARM WELCOME ^_^   |
-----

Dear Admin,
(1). Food List
(2). Insert Food to Food List as Queue
(3). Delete Food as Queue
(4). Search Food by ID
(5). Sort Food by ID
(6). Logout
(7). Exit
Choose an option-
```

CURRENT STOCK		
FOOD-ID	FOOD-NAME	FOOD-PRICE
10	ESPRESSO	5
82	NasiLemak	4.5
778	Nasi_lemak_Ayam	8.5
13	Nasi_Goreng_Ayam	6.5
14	Nasi_Goreng_Kampung	6
18	Nasi_Goreng_Dagging	8
16	Nasi_Goreng_Kambing	8.5
20	Nasi_Goreng_Biasa	4.5
17	Maggi_Goreng_Biasa	4
30	Maggi_Goreng_Ayam	7
29	Mee_Goreng_Biasa	4
21	Mee_Goreng_Ayam	7
22	Bihoon_Goreng_Biasa	3.5
19	Bihoon_Goreng_Ayam	6.5
28	Kuey_Teow_Goreng_Biasa	3.75
23	Kuey_Teow_Goreng_Ayam	7.75
31	Roti_Canai_Kosong	1
80	Roti_Teluer	2
32	Roti_Teluer_Bawang	2.5
41	Roti_Teluer_Bawang_Chili	3
42	Roti_Sambal	5
111	Teh_o	1
145	Teh_Tarik	2
121	Coffee_Panas	3
119	Coffee_Tarik	3.5
167	Teh_o_limau	1.5
165	Teh_O_ice	1.8
170	Teh_Ice	2.5
198	Milo_Ice	3.5
201	Siraf_Ice	1.7
197	coffee_ice	4
195	Nescafe_Ice	3.8
208	Sifar_Bundong_ice	2
511	Juice_Tambikai	4
523	Juice_Pisang	4
512	juice_carrot	4

Number of Foods in the FoodList:36

Insert Food to Food List

Please Enter the Food ID : 23
Please Enter the Food Name : Kuey_Teow_Goreng_Ayam
Please Enter the Food Price : 7.75

523	Juice_Pisang	4
512	juice_carrot	4
23	Kuey_Teow_Goreng_Ayam	7.75

Number of Foods in the FoodList:37

23	Kuey_Teow_Goreng_Ayam	7.75
----	-----------------------	------

Number of Foods in the FoodList:36

Search Food by ID

Please enter the food ID :

Search Food by ID

Please enter the food ID : 019

FOOD-ID
19

FOOD-NAME
oreng_Ayam

FOOD-PRICE
6.5

| Sort Food by ID |

FOOD-ID

FOOD-NAME

FOOD-PRICE

10	ESPRESSO	5
13	Nasi_Goreng_Ayam	6.5
14	Nasi_Goreng_Kampung	6
16	Nasi_Goreng_Kambing	8.5
17	Maggi_Goreng_Biasa	4
18	Nasi_Goreng_Dagging	8
19	Bihoon_Goreng_Ayam	6.5
20	Nasi_Goreng_Biasa	4.5
21	Mee_Goreng_Ayam	7
22	Bihoon_Goreng_Biasa	3.5
23	Kuey_Teow_Goreng_Ayam	7.75
28	Kuey_Teow_Goreng_Biasa	3.75
29	Mee_Goreng_Biasa	4
30	Maggi_Goreng_Ayam	7
31	Roti_Canai_Kosong	1
32	Roti_Teluer_Bawang	2.5
41	Roti_Teluer_Bawang_Chili	3
42	Roti_Sambal	5
80	Roti_Teluer	2
82	NasiLemak	4.5
111	Teh_o	1
119	Coffee_Tarik	3.5
121	Coffee_Panas	3
145	Teh_Tarik	2
165	Teh_O_ice	1.8
167	Teh_o_limau	1.5
170	Teh_Ice	2.5
195	Nescafe_Ice	3.8
197	coffee_ice	4
198	Milo_Ice	3.5
201	Siraf_Ice	1.7
208	Sifar_Bundong_ice	2
511	Juice_Tambikai	4
512	juice_carrot	4
523	Juice_Pisang	4
778	Nasi lemak Ayam	8.5

```
Number of Foods in the FoodList:36
```

```
Dear Admin,
(1). Food List
(2). Insert Food to Food List as Queue
(3). Delete Food as Queue
(4). Search Food by ID
(5). Sort Food by ID
(6). Logout
(7). Exit
Choose an option- 6
```

```
Dear Admin,
(1). Food List
(2). Insert Food to Food List as Queue
(3). Delete Food as Queue
(4). Search Food by ID
(5). Sort Food by ID
(6). Logout
(7). Exit
Choose an option- 7
```

FOR CUSTOMER

```
*****
|          WELCOME TO FOOD ORDERING SYSTEM          |
*****

DEAR USER, YOU WANT TO ACCESS AS :

      (1). Administrator
      (2). Customer
      (3). Exit

PLEASE CHOOSE AN OPTION- 2
```

```
-----
|          DEAR CUSTOMER, WE GIVE YOU A WARM WELCOME ^_^          |
-----

ENTER YOUR NAME : ANIKA
ENTER YOUR PHONE NUMBER: 012345678
```

```
|-----|
| DEAR CUSTOMER, WE GIVE YOU A WARM WELCOME ^_^ |
|-----|

ENTER YOUR NAME : ANIKA
ENTER YOUR PHONE NUMBER: 012345678

Dear Customer,
(1). Food List
(2). Sort Food List By Price
(3). Add Item to FoodCart
(4). Sort FoodCart by Price
(5). Delete Food as Queue from FoodCart
(6). Place Order
(7). Logout
(8). Exit
Choose an option-
```

CURRENT STOCK		
FOOD-ID	FOOD-NAME	FOOD-PRICE
10	ESPRESSO	5
82	Nasilemak	4.5
778	Nasi_lemak_Ayam	8.5
13	Nasi_Goreng_Ayam	6.5
14	Nasi_Goreng_Kampung	6
18	Nasi_Goreng_Dagging	8
16	Nasi_Goreng_Kambing	8.5
20	Nasi_Goreng_Biasa	4.5
17	Maggi_Goreng_Biasa	4
30	Maggi_Goreng_Ayam	7
29	Mee_Goreng_Biasa	4
21	Mee_Goreng_Ayam	7
22	Bihoon_Goreng_Biasa	3.5
19	Bihoon_Goreng_Ayam	6.5
28	Kuey_Teow_Goreng_Biasa	3.75
23	Kuey_Teow_Goreng_Ayam	7.75
31	Roti_Canai_Kosong	1
80	Roti_Teluer	2
32	Roti_Teluer_Bawang	2.5
41	Roti_Teluer_Bawang_Chili	3
42	Roti_Sambal	5
111	Teh_o	1
145	Teh_Tarik	2
121	Coffee_Panas	3
119	Coffee_Tarik	3.5
167	Teh_o_limau	1.5
165	Teh_O_ice	1.8
170	Teh_Ice	2.5
198	Milo_Ice	3.5
201	Siraf_Ice	1.7
197	coffee_ice	4
195	Nescafe_Ice	3.8
208	Sifar_Bundong_ice	2
511	Juice_Tambikai	4
523	Juice_Pisang	4
512	juice_carrot	4
Number of Foods in the FoodList:36		

Sort Food by Price		
FOOD-ID	FOOD-NAME	FOOD-PRICE
31	Roti_Canai_Kosong	1
111	Teh_o	1
167	Teh_o_limau	1.5
201	Siraf_Ice	1.7
165	Teh_O_ice	1.8
80	Roti_Teluer	2
145	Teh_Tarik	2
208	Sifar_Bundong_ice	2
32	Roti_Teluer_Bawang	2.5
170	Teh_Ice	2.5
41	Roti_Teluer_Bawang_Chili	3
121	Coffee_Panas	3
22	Bihoon_Goreng_Biasa	3.5
119	Coffee_Tarik	3.5
198	Milo_Ice	3.5
28	Kuey_Teow_Goreng_Biasa	3.75
195	Nescafe_Ice	3.8
17	Maggi_Goreng_Biasa	4
29	Mee_Goreng_Biasa	4
197	coffee_ice	4
511	Juice_Tambikai	4
523	Juice_Pisang	4
512	juice_carrot	4
82	NasiLemak	4.5
20	Nasi_Goreng_Biasa	4.5
10	ESPRESSO	5
42	Roti_Sambal	5
14	Nasi_Goreng_Kampung	6
13	Nasi_Goreng_Ayam	6.5
19	Bihoon_Goreng_Ayam	6.5
30	Maggi_Goreng_Ayam	7
21	Mee_Goreng_Ayam	7
23	Kuey_Teow_Goreng_Ayam	7.75
18	Nasi_Goreng_Dagging	8
778	Nasi_lemak_Ayam	8.5
16	Nasi_Goreng_Kambing	8.5

Number of Foods in the FoodList:36

Number of Foods in the FoodList:36

Enter the FoodID to add the food in the FoodCart :

Enter the FoodID to add the food in the FoodCart : 512

FOOD-ID	FOOD-NAME	FOOD-PRICE
512	juice_carrot	4

Number of Foods in the FoodList:1

Do you want Add more Food? (y/n) :

Do you want Add more Food? (y/n) : y

Enter the FoodID to add the food in the FoodCart : 10

FOOD-ID	FOOD-NAME	FOOD-PRICE
512	juice_carrot	4
10	ESPRESSO	5

Number of Foods in the FoodList:2

Do you want Add more Food? (y/n) :

Enter the FoodID to add the food in the FoodCart : 512

FOOD-ID	FOOD-NAME	FOOD-PRICE
512	juice_carrot	4

Number of Foods in the FoodList:1

Do you want Add more Food? (y/n) : y

Enter the FoodID to add the food in the FoodCart : 10

FOOD-ID	FOOD-NAME	FOOD-PRICE
512	juice_carrot	4
10	ESPRESSO	5

Number of Foods in the FoodList:2

Do you want Add more Food? (y/n) : n

```
Do you want Add more Food? (y/n) : n
      FOOD-ID          FOOD-NAME          FOOD-PRICE
-----
      512             juice_carrot          4
      10             ESPRESSO             5
Number of Foods in the FoodList:2

Dear Customer,
(1). Food List
(2). Sort Food List By Price
(3). Add Item to FoodCart
(4). Sort FoodCart by Price
(5). Delete Food as Queue from FoodCart
(6). Place Order
(7). Logout
(8). Exit
Choose an option-
```

```
Dear Customer,
(1). Food List
(2). Sort Food List By Price
(3). Add Item to FoodCart
(4). Sort FoodCart by Price
(5). Delete Food as Queue from FoodCart
(6). Place Order
(7). Logout
(8). Exit
Choose an option- 4
```

```
-----
|          Sort FoodCart by Price          |
-----
      FOOD-ID          FOOD-NAME          FOOD-PRICE
-----
      512             juice_carrot          4
      10             ESPRESSO             5
Number of Foods in the FoodList:2

Dear Customer,
(1). Food List
(2). Sort Food List By Price
(3). Add Item to FoodCart
(4). Sort FoodCart by Price
(5). Delete Food as Queue from FoodCart
(6). Place Order
(7). Logout
(8). Exit
Choose an option-
```

```
-----  
FOOD-ID          FOOD-NAME          FOOD-PRICE  
-----  
10              ESPRESSO              5  
Number of Foods in the FoodList:1  
  
Dear Customer,  
  (1). Food List  
  (2). Sort Food List By Price  
  (3). Add Item to FoodCart  
  (4). Sort FoodCart by Price  
  (5). Delete Food as Queue from FoodCart  
  (6). Place Order  
  (7). Logout  
  (8). Exit  
  Choose an option-
```

```
-----  
| Place Order |  
-----  
  
Customer Name : ANIKA  
Customer Phone Number : 0123456789  
  
FOOD-ID          FOOD-NAME          FOOD-PRICE  
-----  
10              ESPRESSO              5  
Number of Foods in the FoodList:1  
  
Total Price : 5  
  
Dear Customer,  
  (1). Food List  
  (2). Sort Food List By Price  
  (3). Add Item to FoodCart  
  (4). Sort FoodCart by Price  
  (5). Delete Food as Queue from FoodCart  
  (6). Place Order  
  (7). Logout  
  (8). Exit  
  Choose an option-
```



```
Dear Customer,  
  (1). Food List  
  (2). Sort Food List By Price  
  (3). Add Item to FoodCart  
  (4). Sort FoodCart by Price  
  (5). Delete Food as Queue from FoodCart  
  (6). Place Order  
  (7). Logout  
  (8). Exit  
Choose an option- 7
```

```
Dear Customer,  
  (1). Food List  
  (2). Sort Food List By Price  
  (3). Add Item to FoodCart  
  (4). Sort FoodCart by Price  
  (5). Delete Food as Queue from FoodCart  
  (6). Place Order  
  (7). Logout  
  (8). Exit  
Choose an option- 8
```