CAFÉ BILLING SYSTEM

A PROJECT REPORT

Submitted by

Akshat Neolia [RA2211031010080]

Under the Guidance of

Mr. K. MANIKANDAN

Assistant Professor, NWC

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

with specialization in IT

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

MAY 2023

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Project Report titled **“CAFÉ BILLING SYSTEM”** is the bonafide work done by **AKSHAT NEOLIA [RA2211031010080]** who completed the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE	**SIGNATURE**
Mr. K. Manikandan	Dr. Annapurani.K
OODP – Course Faculty	**Head of the Department**
Assistant Professor	Department of NWC
Department of NWC	SRMIST
SRMIST	

TABLE OF CONTENTS

1. PROBLEM STATEMENT

Our code helps to generate an invoice for the bill of a Cafe's menu. A billing system can be very useful within a business environment. Instead of making bills manually or to sum up the total manually, it is very much time consuming and may have some human errors like adding up the wrong total or adding wrong items into the bill. When making a hand written bill the owner and customer both have to repeatedly check the total, items added, etc.

It also sometimes results in a Bad Impression towards the Café from a Customer. Ideally, users should be able to generate bills without any mistakes and quickly, enabling them to fasten or improve their process. To overcome this problem, we have come up with this project, that is, Café Billing System Using C++.

PROJECT NAME	CAFÉ BILLING SYSTEM
ABSTRACT	Cafe Billing System UML Diagrams are used to represent the admin system as well as its pri users, roles, activities, or classes.
UML DIAGRAM	Sequence Diagram, Activity Diagram, Class Diag Use Case Diagram, Collaboration Diag Deployment Diagram, Component Diagram, Pac Diagram, State Chart Diagram
USERS	Cafe Owner, Receptionist, Staff members
TOOLS USED	Hand Draw

2. MODULES DESCRIPTION

Menu Management Module: This module will allow the cafe owner to manage the menu items and their prices. It will also allow the owner to add new items, remove items, and update prices.

Order Management Module: This module will allow the staff members to take orders from the customers and process them. .

Payment Management Module: This module will manage the payment process for the cafe by. It will allow the staff members to generate final receipts including GST.

2. UML DIAGRAM

- Unified Modelling Language (UML) diagrams are visual representations used for modelling object-oriented software systems.
- UML diagrams provide a standardized way to visualize and communicate software designs, making them easier to understand and implement.
- There are two main types of UML diagrams:
- behavioral
- structural
- Structural UML diagrams describe the static structure of a system, including its components, classes, interfaces, and relationships between them.
- Behavioral UML diagrams describe the dynamic behavior of a system, including the interactions between objects and the changes in their states.

Structural UML Diagrams include:

1. Class Diagrams
2. Object Diagrams
3. Component Diagrams
4. Deployment Diagrams
5. Package Diagrams

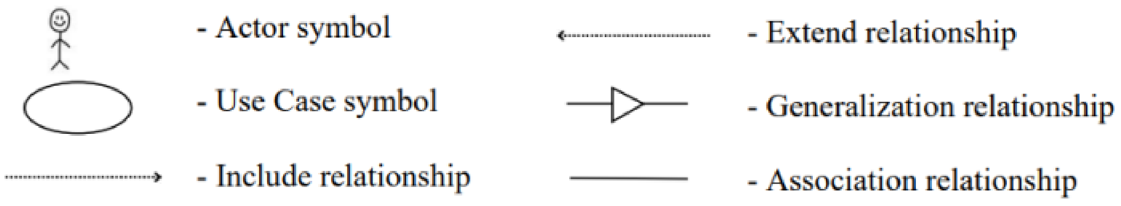
Behavioral UML Diagrams include:

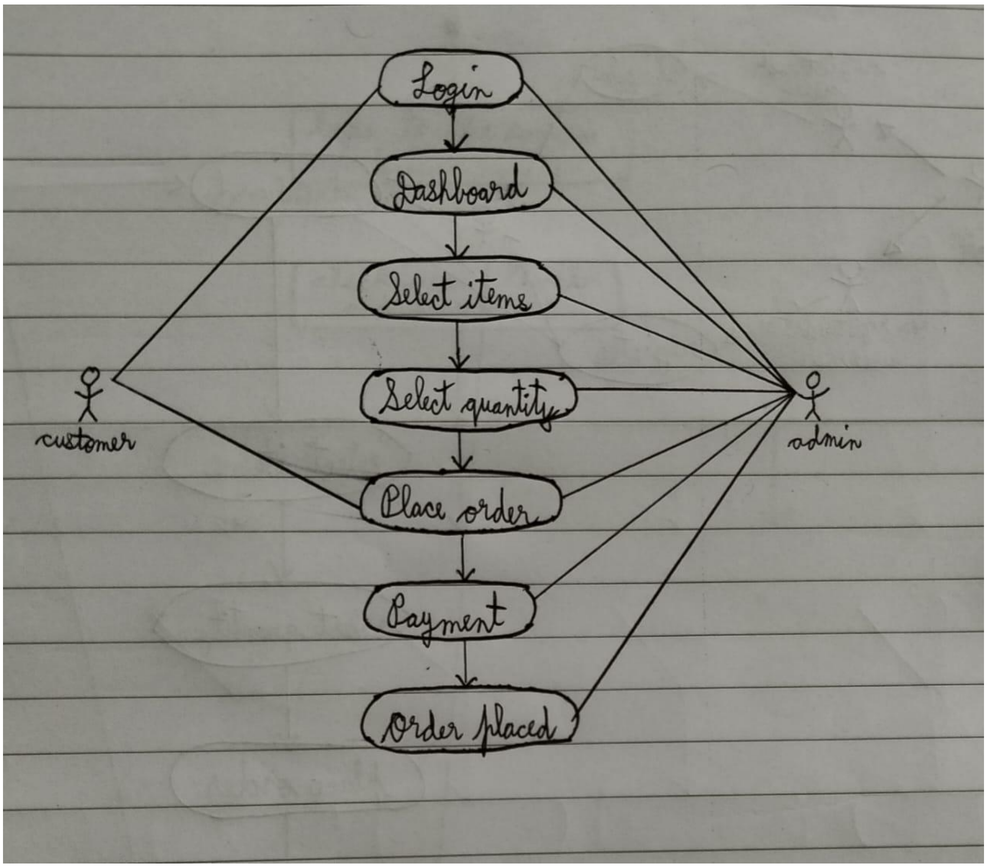
1. Use Case Diagrams
2. Sequence Diagrams
3. Collaboration Diagrams
4. State Diagrams
5. Activity Diagrams

3.1 Use Case Diagram

A use case diagram is a type of UML behavior diagram that represents the interactions between actors and a system to achieve specific goals or tasks. It visually depicts the different use cases and actors involved in a system, providing an overview of the system's functionalities and requirements.

NOTATIONS USED:



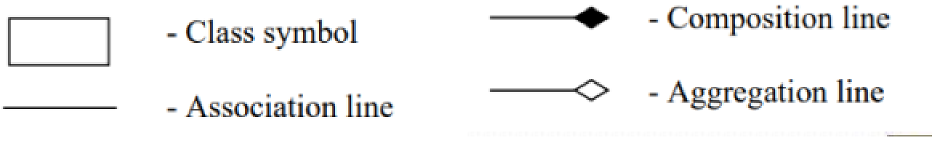


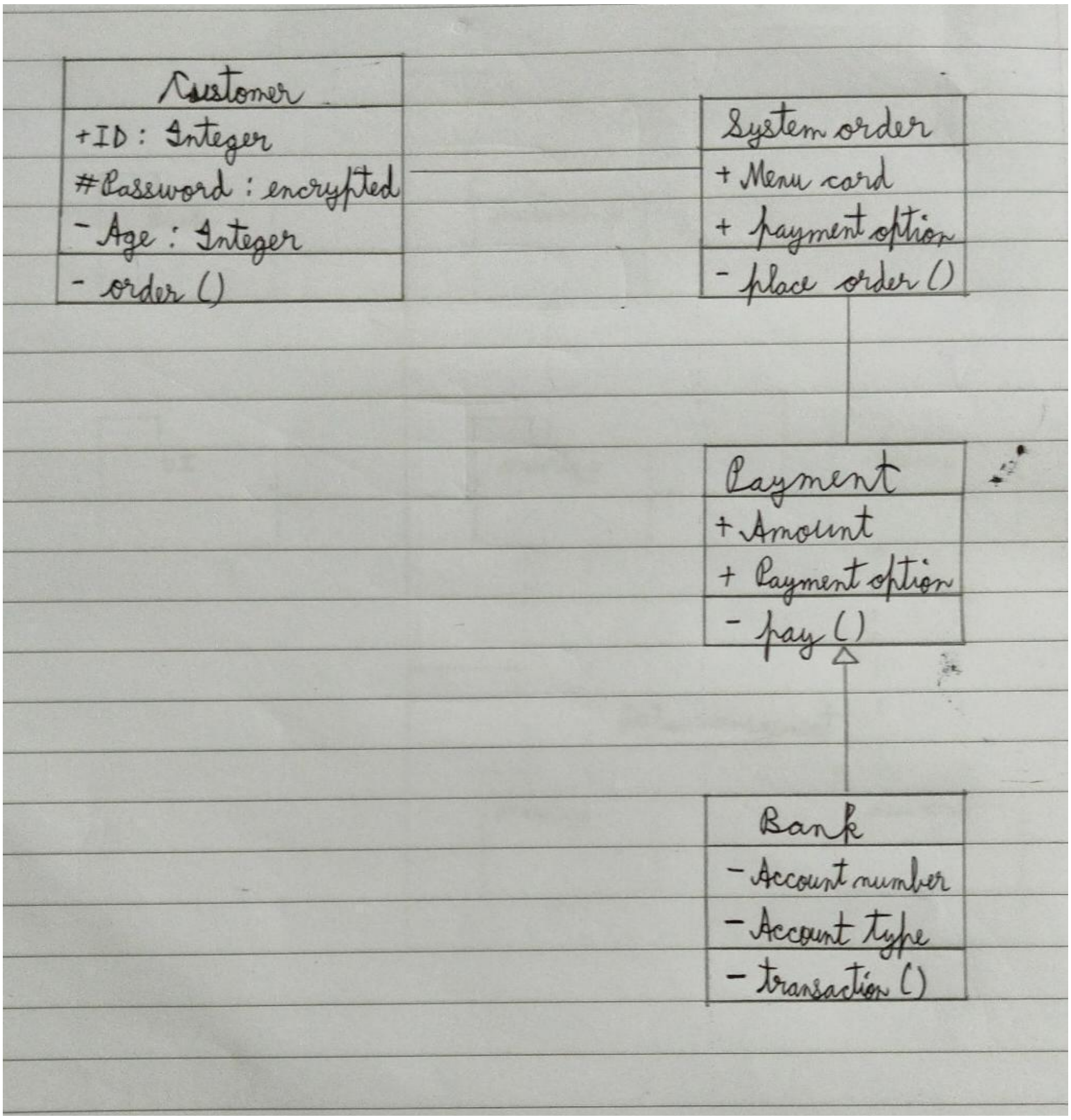
In this use case diagram, there are three actors: the customer, the cafe bill management system, and the cashier. The customer interacts with the system by adding items to their order, modifying the quantity of items, and removing items from their order. The system calculates the total bill and accepts payment from the customer. Finally, the cashier interacts with the system to manage the payment and complete the transaction.

3.2 Class Diagram

A class diagram is a UML diagram that depicts the structure of a system by showing the classes, objects, interfaces, and their relationships. It provides a static view of the system and is used to visualize the objects and their interactions within the system. Classes are represented as boxes with their attributes and methods, and relationships between classes are shown using lines with various symbols.

NOTATIONS USED:



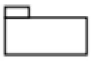


In this diagram, the Cafe Bill Management class has a list of menu items and a list of order items. The add_item, remove_item, and modify_quantity methods allow the customer to interact with their order. The calculate_total_bill method calculates the total cost of the order, and the accept_payment method accepts payment from the customer.

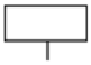
3.3 Sequence Diagram

A sequence diagram is a type of UML diagram that illustrates the interactions between objects or components in a system over time. It depicts the order in which objects interact with each other to achieve a specific task or goal. It is a dynamic view of the system that shows the flow of messages exchanged between objects and the timing of their interactions.

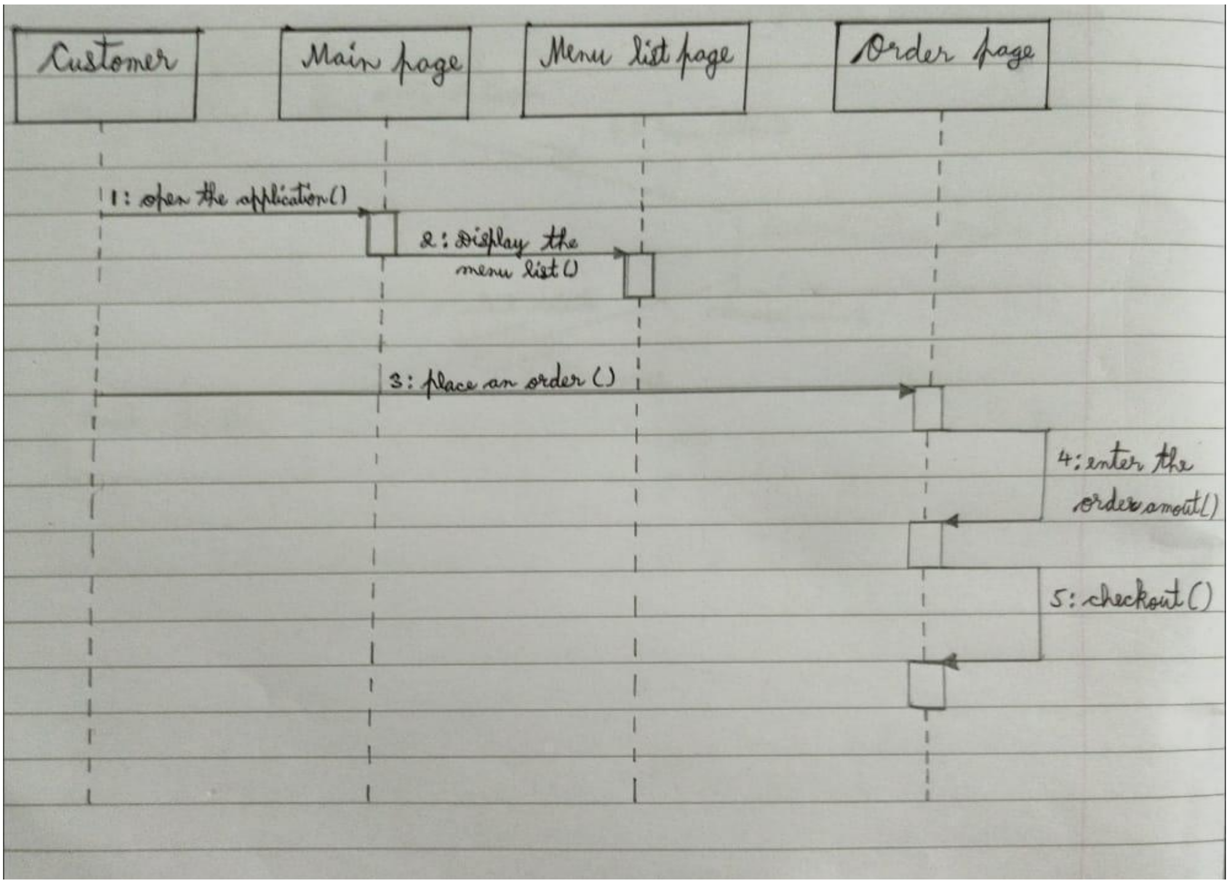
NOTATIONS USED:



- Package symbol



- Lifeline symbol


Customer
Main page
Menu list page
Order page
1: open the application()
2: Display the menu list()
3: place an order ()
4: enter the order amout()
5: checkout()

In this diagram, the customer first selects menu items and specifies the quantity of each item. The select_menu_item message is sent to the Cafe Bill Management system, which responds by adding the item and quantity to the order using the add_item method.

The customer can then modify the quantity of an item or remove an item from the order, using the modify_quantity and remove_item methods respectively.
Once the order is finalized, the customer requests the total bill using the calculate_total_bill method. The Cafe Bill Management system responds with the total bill amount, and the customer can then request a printed bill using the generate_bill method.

3.4 Collaboration Diagram

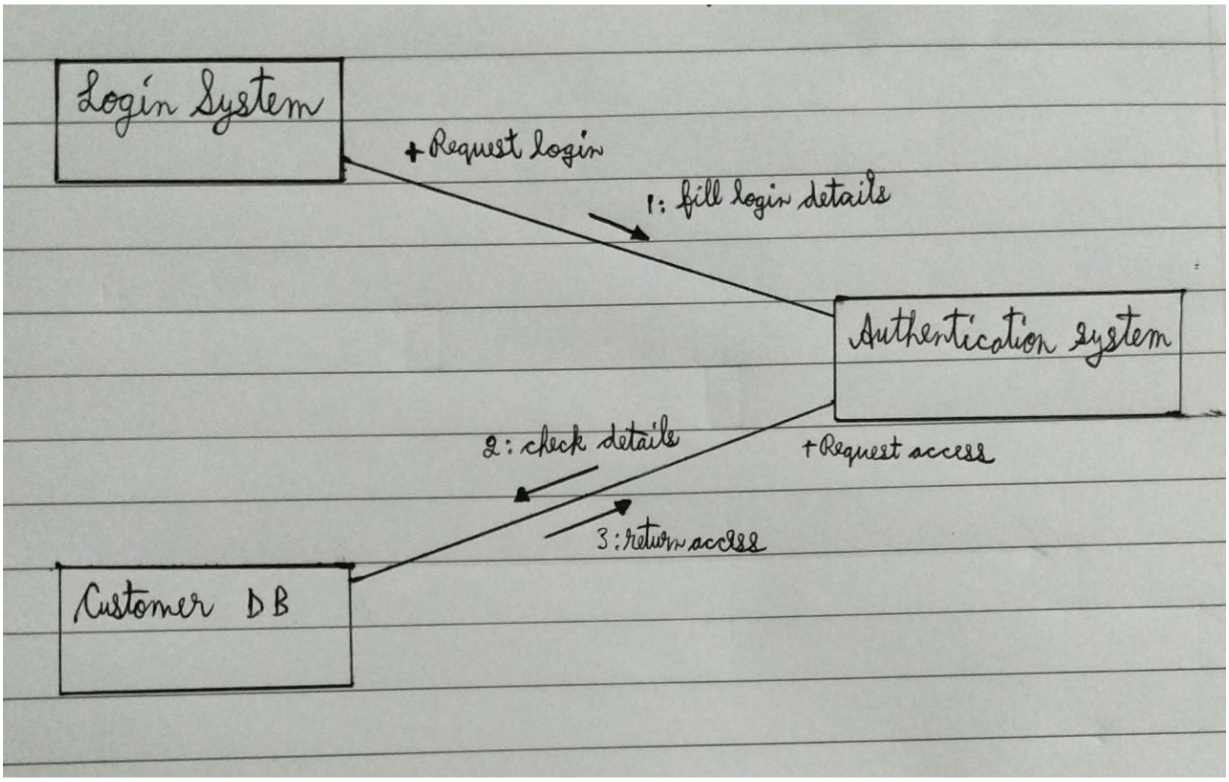
A collaboration diagram is a type of UML diagram that shows the interactions between objects or components in a system to achieve a specific task or goal. It is a visual representation of the collaboration and communication among the objects in a system.

NOTATIONS USED:

□ - Object

——— - Relation/Association

↓ Message - Messages



In this collaboration diagram, the messages are shown passing between the customer and the Cafe Bill Management system. The customer first selects menu items and specifies the quantity of each item. The select_menu_item message is sent to the Cafe Bill Management system, which responds by adding the item and quantity to the order using the add_item method.

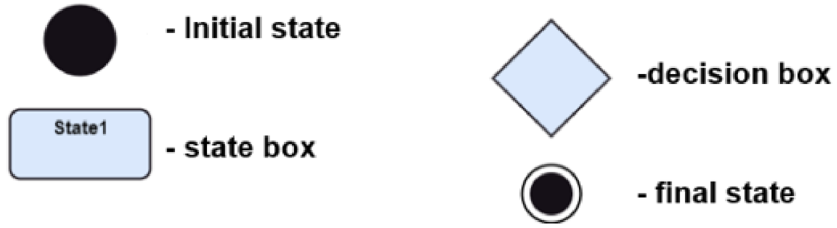
The customer can then modify the quantity of an item or remove an item from the order, using the modify_quantity and remove_item methods respectively.

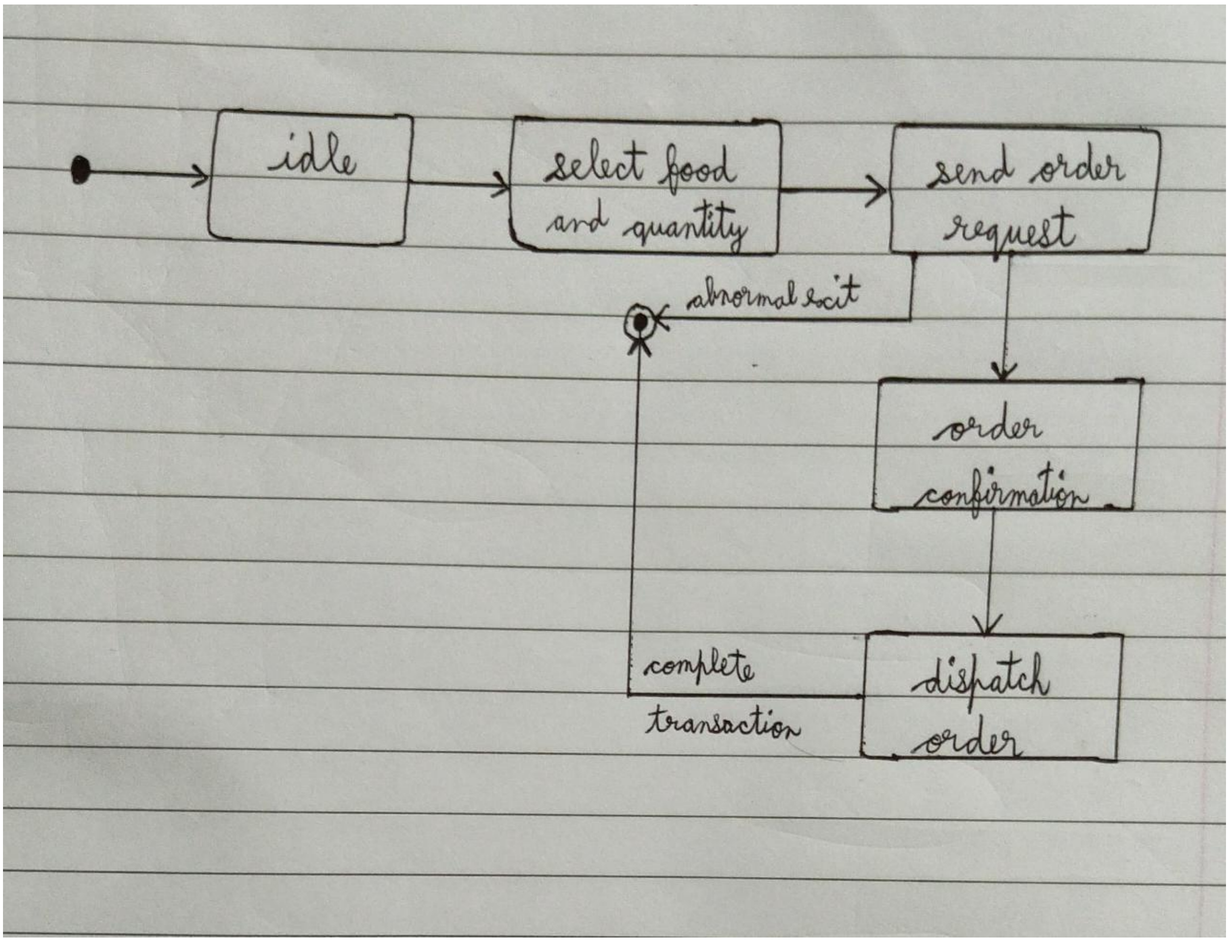
Once the order is finalized, the customer requests the total bill using the calculate_total_bill method. The Cafe Bill Management system responds with the total bill amount, and the customer can then request a printed bill using the generate_bill method.

3.5 State Chart Diagram

A state chart diagram is a type of UML diagram that shows the different states and transitions of an object or system over time. It is a visual representation of the behavior of an object or system, and how it responds to external stimuli or events.

NOTATIONS USED:





In this state chart diagram, the Cafe Bill Management system begins in the IDLE state, where it is waiting for input from the customer.

Once the customer selects a menu item and specifies the quantity, the system transitions to the ADD ITEM state. Here, the system adds the item and quantity to the order.

If the customer modifies the quantity of an item, the system updates the order with the new quantity. If the customer removes an item, the system removes that item from the order.

3.6 Activity Diagram

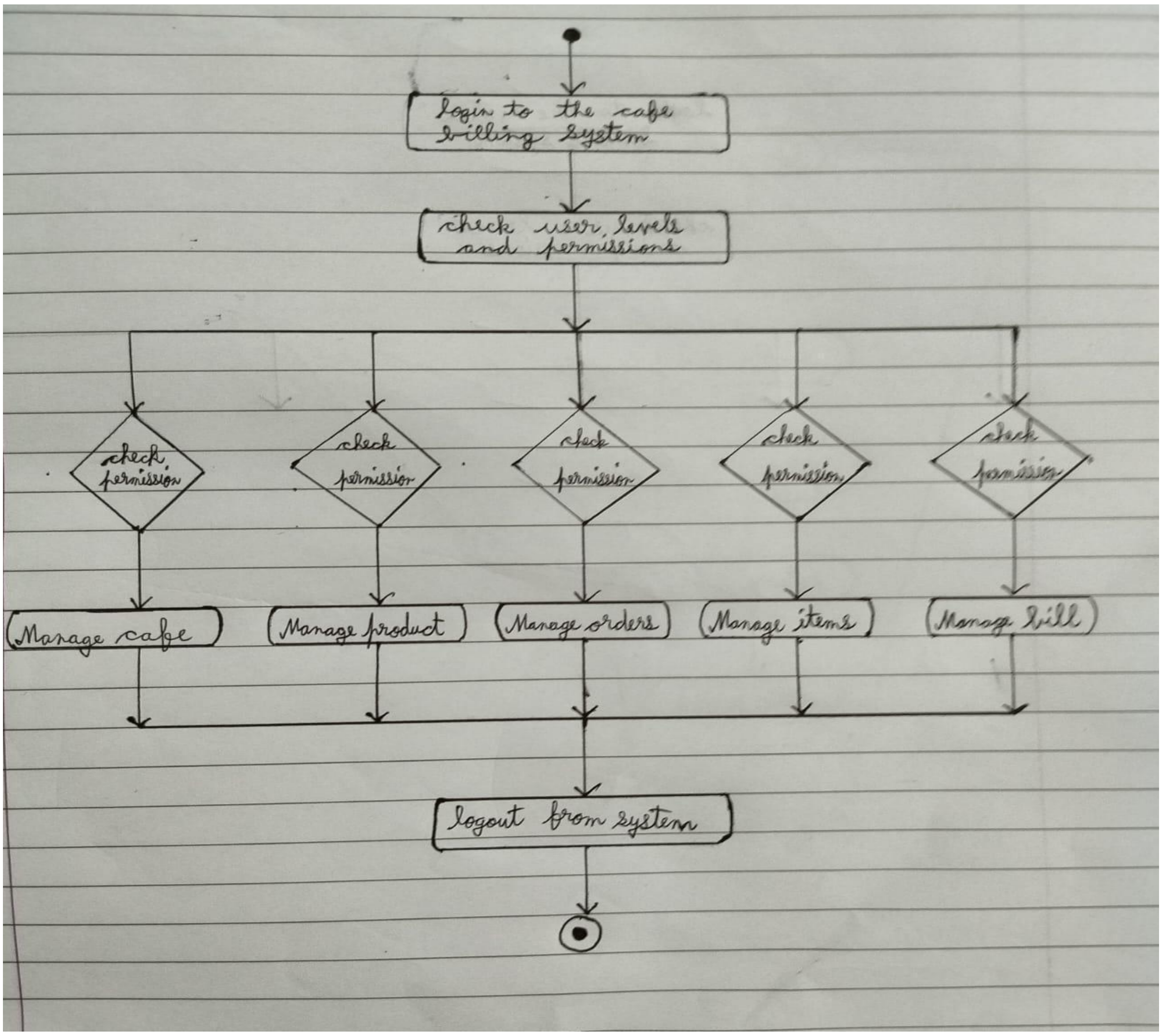
An activity diagram is a type of UML diagram that shows the workflow or activities of a system or process. It depicts the sequence of activities, decision points, and the flow of control or data between them.

NOTATIONS USED:

● - Start symbol

◉ - End symbol ◇ - Decision symbol

▭ - Activity symbol → - Object flow arrow



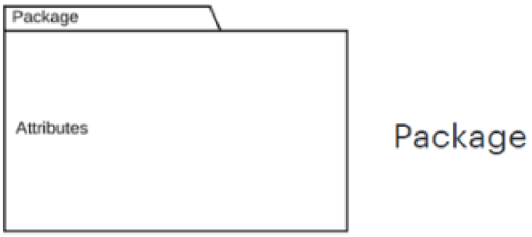
If the customer wants to modify or remove an item, the system moves to the Modify/Remove Item activity to make changes in the order.

After all items have been added or modified/removed, the system proceeds to the Calculate Total Bill activity, where it calculates the total cost of the order. Once the total bill is calculated, the system moves to the Generate Bill activity to produce the printed bill.

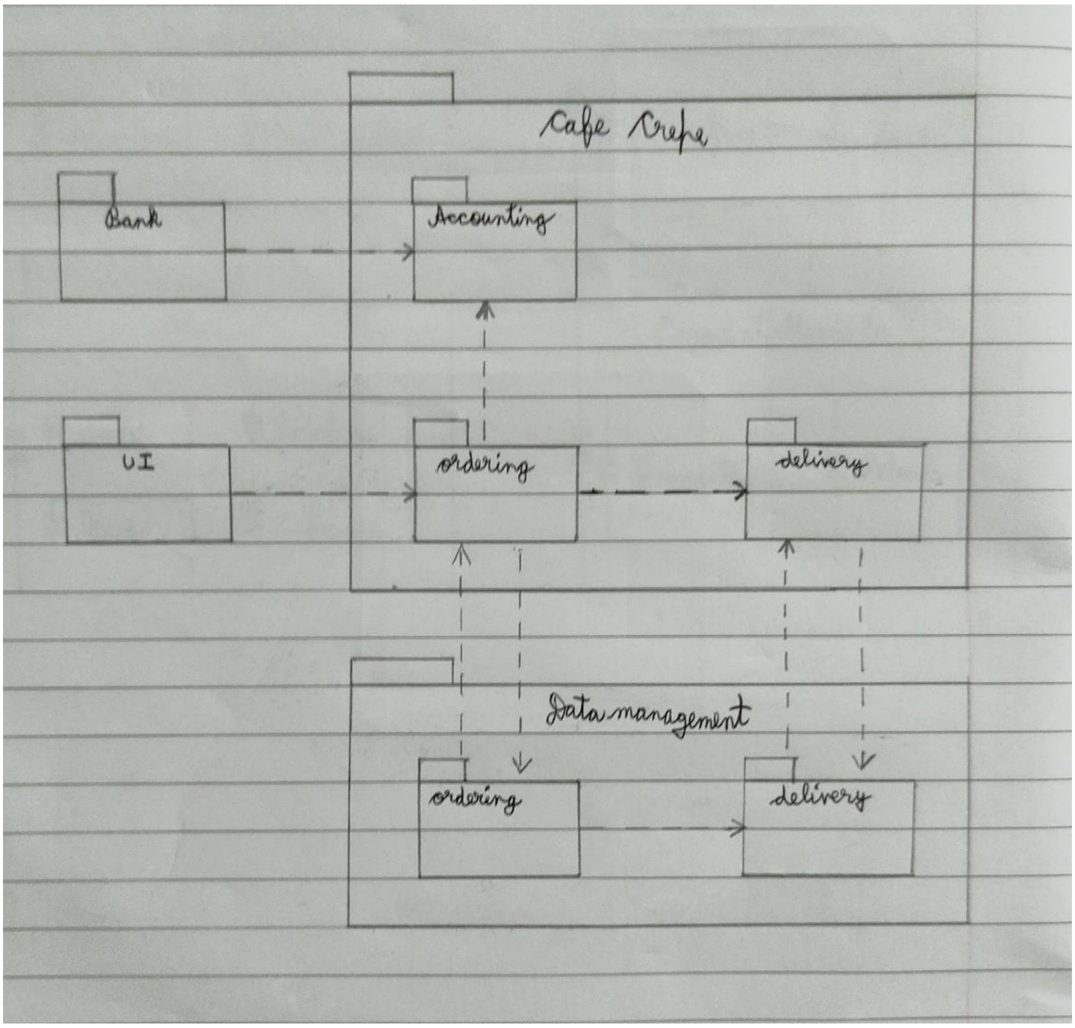
3.7 Package Diagram

A package diagram is a type of UML diagram that shows the dependencies and organization of the packages or modules in a system. It depicts the relationships between packages and the elements they contain, such as classes, interfaces, and other packages.

NOTATIONS USED:



Dependency



In this package diagram, the Cafe Bill Management system is represented as a top-level package. The system is dependent on the Order Management package, which manages the overall order process.

The Order Detail package represents the details of each item ordered, including the name, quantity, and price.

3.8 Component Diagram

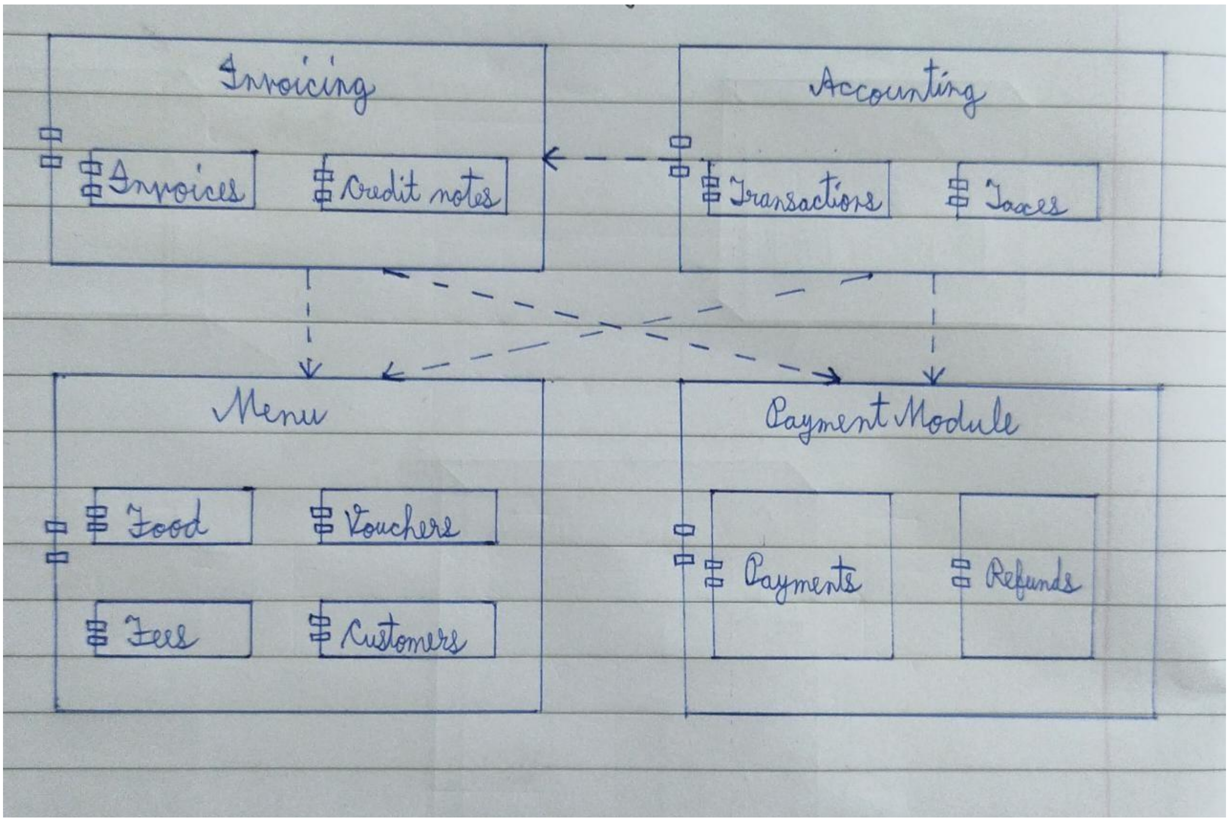
A component diagram is a type of UML diagram that shows the components, interfaces, and dependencies of a system or software application. It depicts the physical or logical components and their relationships, as well as the interfaces between them.

NOTATIONS USED:

▭ - Component symbol

⇢ - Dependency arrow

⊟ - Port symbol



In this package diagram, the Cafe Bill Management system is represented as a top-level package. The system is dependent on the Order Management package, which manages the overall order process.

The Order Management package includes two sub-packages, Item and Quantity, which handle the items and quantities of the order, respectively.

The Order Detail package represents the details of each item ordered, including the name, quantity, and price.

3.9 Deployment Diagram

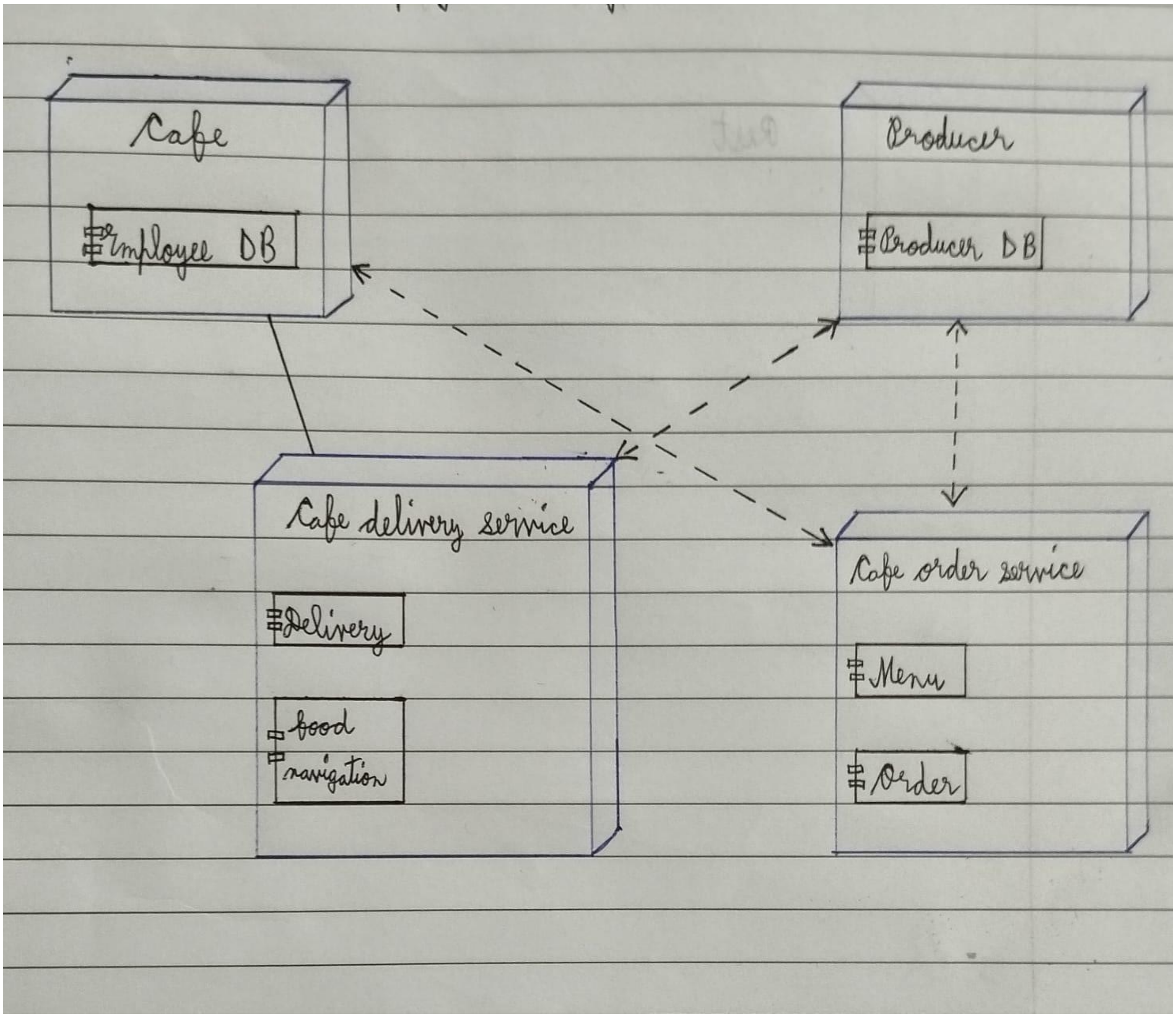
A deployment diagram is a type of UML diagram that shows the physical or logical deployment of a system or software application. It depicts the hardware or software nodes, the artifacts or components deployed on them, and the communication and dependencies between them.

NOTATIONS USED:

□ - Artifact symbol

→ - Association relationship

⇢ - Dependency relationship



In this deployment diagram, the Cafe Bill Management system is represented as a web application deployed on a web server running Tomcat.

The web server communicates with a separate database server running MySQL, which stores the menu items, order details, and other relevant data.

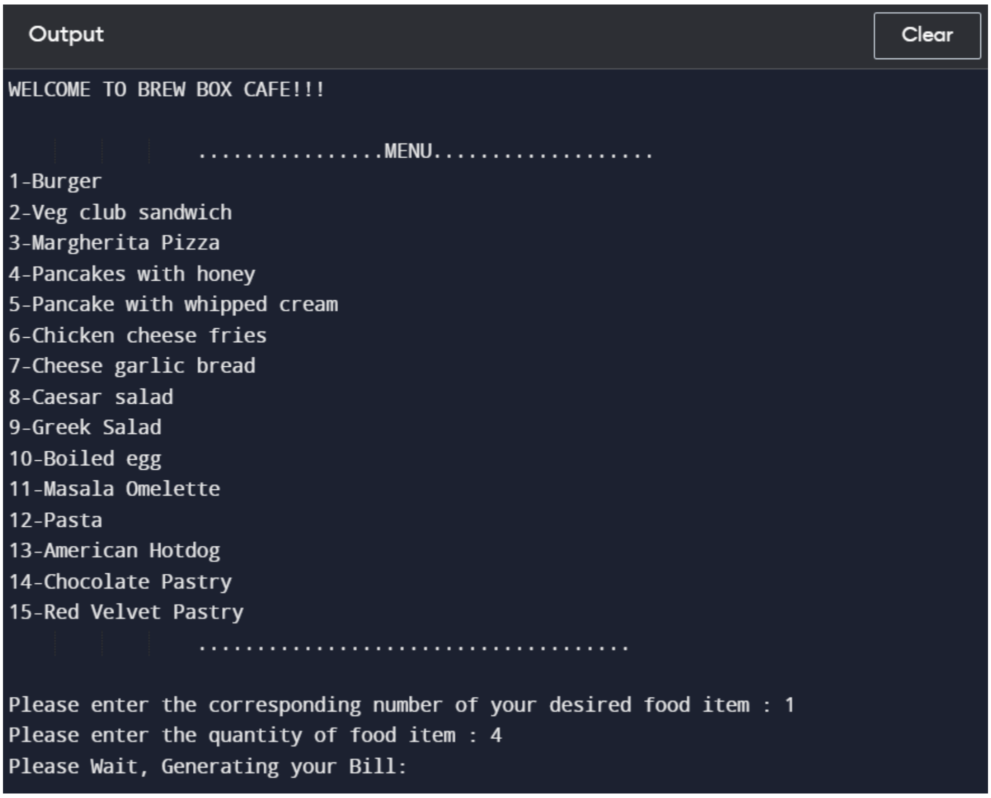
4. SOURCE CODE

```
#include <iostream>
#include <map>
using namespace std;

int main()
{
   cout << "WELCOME TO BREW BOX CAFE!!!\n\n";
   cout << "\t\t\t\t................MENU...................\n1-Burger\n2-Veg club sandwich\n3-Margherita
Pizza\n4-Pancakes with honey\n5-Pancake with whipped cream\n6-Chicken cheese
fries\n7-Cheese garlic bread\n8-Caesar salad\n9-Greek Salad\n10-Boiled egg\n11-Masala
Omelette\n12-Pasta\n13-American Hotdog\n14-Chocolate Pastry\n15-Red Velvet
Pastry\n\t\t\t\t....................................\n\n";

   map<int, string> database2 = {{1,"Burger"},{2,"Veg club sandwich"},{3,"Margherita
Pizza"},{4,"Pancakes with honey"},{5,"Pancake with whipped cream"},{6,"Chicken cheese
fries"},{7,"Cheese garlic bread"},{8,"Caesar salad"},{9,"Greek Salad"},{10,"Boiled
egg"},{11,"Masala Omelette"},{12,"Pasta"},{13,"American Hotdog"},{14,"Chocolate
Pastry"},{15,"Red Velvet Pastry"}};
   map<int, int> database1 =
{{1,70},{2,40},{3,150},{4,90},{5,100},{6,130},{7,70},{8,80},{9,90},{10,30},{11,40},{12,110},{
13,80},{14,60},{15,20}};

   int fitem;
   cout << "Please enter the corresponding number of your desired food item : ";
   cin >> fitem;
if (fitem >= 1 && fitem <= 15) {
      int fquan;
      cout << "Please enter the quantity of food item : ";
      cin >> fquan;
      int b = database1[fitem];
      string c = database2[fitem];
      int d = b * fquan;
      double GST = (d * 18) / 100.0;
      double e = d + GST;
      cout << "\nPlease Wait, Generating your Bill:\n\n";
      cout <<
"***************************************************************************
*****" << endl;
      cout << "\t\t\t\t\t\tTHE BREW BOX CAFE ,Inc.\n\t\t\t\t\t\tSRM
KTR;\n\t\t\t\t\t\tCHENNAI.\n\n";
      cout <<
"======================================================================
==============" << endl;
```

```
        cout << "FOOD NAME\t\t\t\t\t\tPRICE\t\t\t\t\t\t\tQUANTITY\n" << c << "\t\t\t\t\t\tRs." << b
<< "\t\t\t\t\t\t\t\t" << fquan << "\n\n";
         cout <<
"==============================================================================
==============" << endl;
         cout << "TOTAL\t\t\t\t\t\tGST\t\t\t\t\t\tTOTAL(GST)\nRs." << d << "\t\t\t\t\tRs." << GST <<
"\t\t\t\t\t\t\tRs." << e << "\n\n";
         cout <<
"==============================================================================
==============" << endl << endl;
         cout << "\t\t\t\tTHANK YOU FOR ORDERING FOOD WITH US.\n\n";
         cout <<
"********************************************************************************
*****" << endl;
     } else {
         cout << "Enter a Valid Corresponding number\n";
 }
     cout << "\nWE HOPE TO SEE YOU AGAIN AND HAVE A NICE DAY!!!\n";
                                                      return 0;}
```

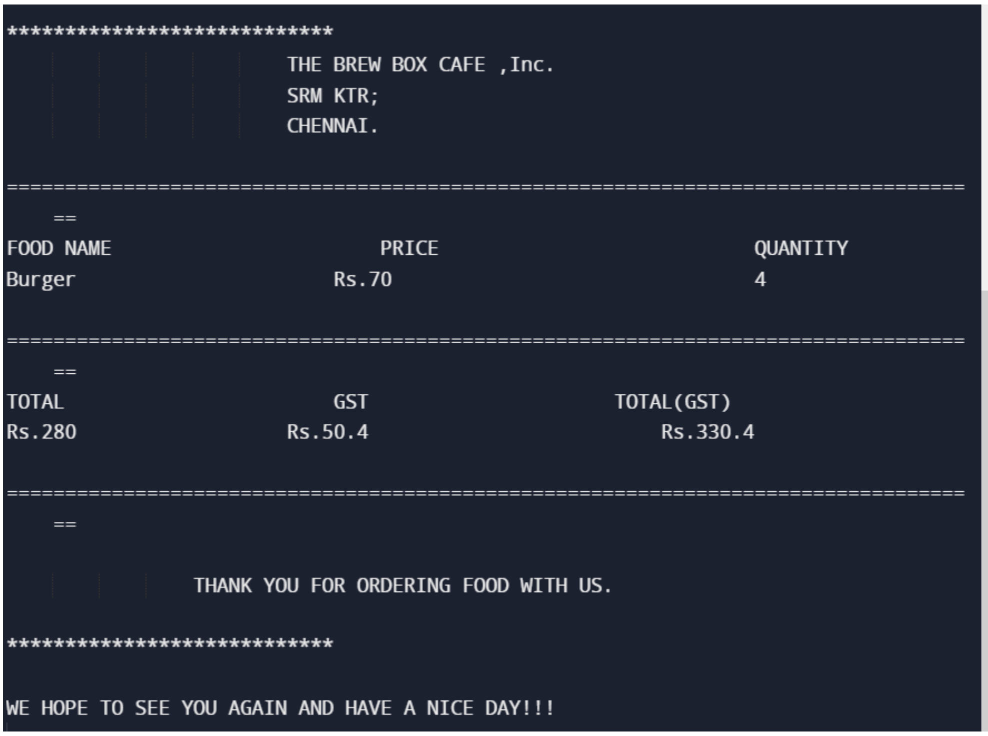
5. INPUT/OUTPUT

INPUT 1:



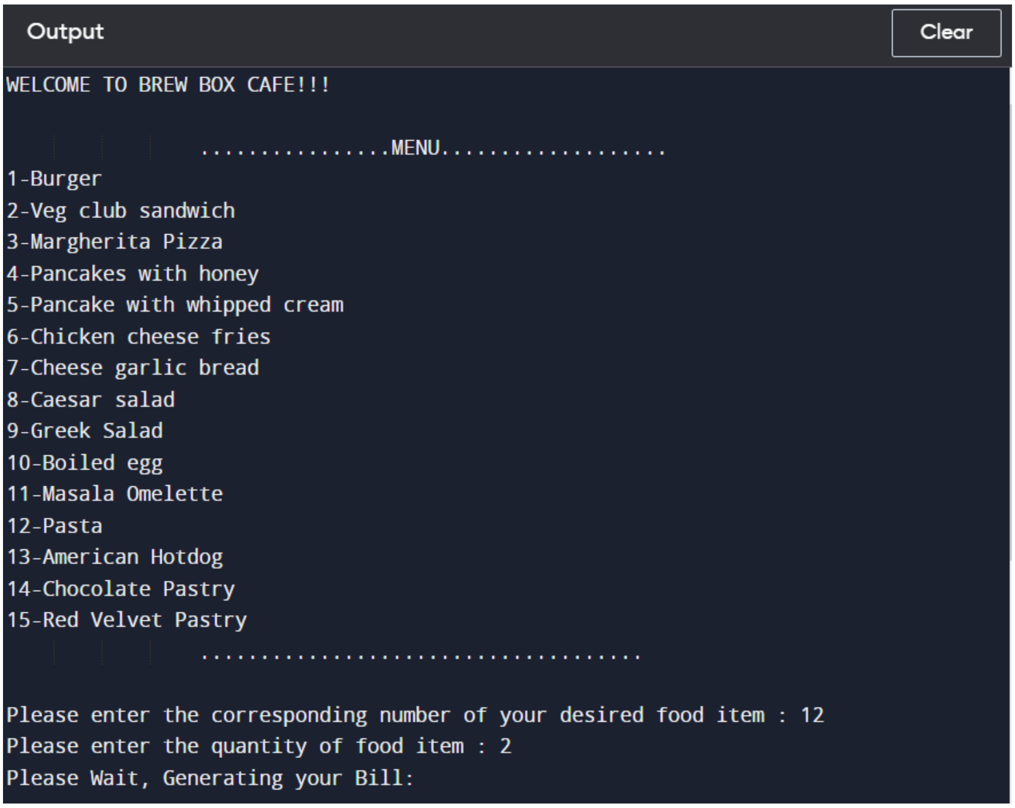
```
WELCOME TO BREW BOX CAFE!!!

                ................MENU..................
1-Burger
2-Veg club sandwich
3-Margherita Pizza
4-Pancakes with honey
5-Pancake with whipped cream
6-Chicken cheese fries
7-Cheese garlic bread
8-Caesar salad
9-Greek Salad
10-Boiled egg
11-Masala Omelette
12-Pasta
13-American Hotdog
14-Chocolate Pastry
15-Red Velvet Pastry
                ....................................

Please enter the corresponding number of your desired food item : 1
Please enter the quantity of food item : 4
Please Wait, Generating your Bill:
```

OUTPUT 1:



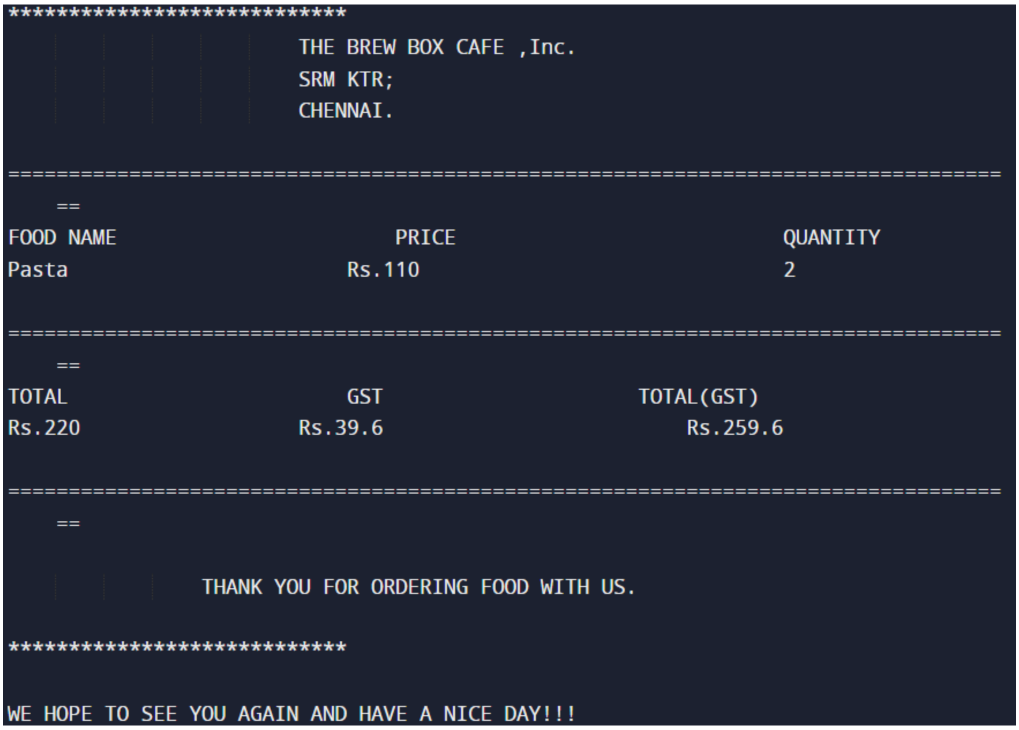
```
****************************
                        THE BREW BOX CAFE ,Inc.
                        SRM KTR;
                        CHENNAI.

=================================================================================
    ==
FOOD NAME                       PRICE                               QUANTITY
Burger                      Rs.70                                   4

=================================================================================
    ==
TOTAL                       GST                     TOTAL(GST)
Rs.280                  Rs.50.4                         Rs.330.4

=================================================================================
    ==

                THANK YOU FOR ORDERING FOOD WITH US.

****************************

WE HOPE TO SEE YOU AGAIN AND HAVE A NICE DAY!!!
```

INPUT 2:



```
WELCOME TO BREW BOX CAFE!!!

                ................MENU..................
1-Burger
2-Veg club sandwich
3-Margherita Pizza
4-Pancakes with honey
5-Pancake with whipped cream
6-Chicken cheese fries
7-Cheese garlic bread
8-Caesar salad
9-Greek Salad
10-Boiled egg
11-Masala Omelette
12-Pasta
13-American Hotdog
14-Chocolate Pastry
15-Red Velvet Pastry
                ....................................

Please enter the corresponding number of your desired food item : 12
Please enter the quantity of food item : 2
Please Wait, Generating your Bill:
```

OUTPUT 2:



```
****************************
                        THE BREW BOX CAFE ,Inc.
                        SRM KTR;
                        CHENNAI.

==================================================================================
    ==
FOOD NAME                       PRICE                           QUANTITY
Pasta                       Rs.110                              2

==================================================================================
    ==
TOTAL                       GST                     TOTAL(GST)
Rs.220                  Rs.39.6                         Rs.259.6

==================================================================================
    ==

                THANK YOU FOR ORDERING FOOD WITH US.

****************************

WE HOPE TO SEE YOU AGAIN AND HAVE A NICE DAY!!!
```

6. CONCLUSION

We would like to thank you for providing us this opportunity to collaborate, to make the project "Café Billing System" in C++.
It was a great learning experience for the both of us as it helped us in bonding and working as a team.
It also helped us practice and learn more about C++.

We learned to create a Café billing system which is very helpful in Café and Restaurants as it has many benefits:

- Efficient
- Time saving
- No human errors

7. REFERENCES

- https://code-projects.org/restaurant-order-system-c-source-code/
- https://www.javatpoint.com/billing-management-system-in-cpp
- https://www.geeksforgeeks.org/c-plus-plus/