Cafeteria Management System

A Project Report

Submitted By

Sabrina Hossain (C211211)

Submitted To

MD. Faisal Bin Al Abid

Assistant Professor, Department of CSE,

IIUC.

***A project report submitted in partial fulfillment of the requirements for the course of Software Development** 1.*

Department of Computer Science and Engineering International

Islamic University Chittagong , Bangladesh.

ABSTRACT

The project titled “CAFETERIA MANAGEMENT SYSTEM” is design with Code Blocks. And my project name is “IIUC CAFETERIA” which is proposed for helping the owner and students both by saving their valuable time .In the university students have their meal and attend their classes timely, also café owner has a huge rush while serving the customers in time. This unique application will help to take the order and billing process easy for the owner. Café owner has to maintain records of daily billing, which will be used to manage sale report. There will be many items available in the café, many customers will be coming at different time for having food, they will be selecting items from the displayed menu and their bills will be generated. This system will save time and will be easy to use when compared to manual work that was done in paper.

TABLE OF CONTENTS

CHAPTER 4: SOFTWARE DESIGN & IMPLEMENTATION

CHAPTER 5: CONCLUSION

CHAPTER-1

Introduction

Cafeteria Management System is a simple application which can automate all kinds of order taking and billing process in a Cafeteria. The purpose of the software is to manage the menu, taking the order and generate the bill for customers in Cafeteria to make the whole process easy and understandable for the customers. Cafeteria management especially manages the menu items, gives interface to the orderings, expenditure report. User to manage the item and manage the category of food too. This project also gives the interface for the bill printing purpose in instant time .It is designed to increase operational efficiency, saving our money and time, maximizing profit and provide more security.it is designed to avoid paperwork. This all be so user friendly so that customer will not find any trouble while purchasing from the cafeteria.

1.1 Objective

- Develop a desktop application, which can easily & effectively manage the ordering and billing of food items.
- Reduce paperwork.

1.2 Limitation

- It provides the service only customers inside the university
- Home delivery service is not available.

1.3 Scope

The main goal of the system is to business automate the process carried out in the organization with improved performance and realize the vision of paperless business. There is different module such as admin and customer. They have separate area in the system. Manager can work in all side of the system. Customer can only view menu of the items, cost, bill etc. The admin can measure how many food items are occupied and how many food items are available in the Cafeteria. Then customer can view and order their food according to their choices. Multi departmental tasks are handling with a centralized database management.

1.4 Problem of existing system

The existing management system of Cafeteria management are generally pen paper based. It is not very effective as the system takes a lot of time and manpower in performing various tasks. Overall the detailing is not maintaining and sometimes there can be some problem to maintain the whole management system. A manual Cafeteria management system requires manager to write down each time. If the manager forgets to mention that the whole system will suffer for the problem. So, this is another problem of the system. The current system of Cafeteria is not computerized. There is no billing system all work is done by man power and memorization of the manager so there is no record preserved of the selling .The manager of cafeteria cannot take a timely order and provide a timely service, due to a lack of staff and waiter.

1.5 Summary

We have descripted the whole process such as existing system, our system, objective in this chapter. Also, the different module in our system and their role or work are described.

CHAPTER -2

Background

2.1 Proposed System

Cafeteria Management System is a computerized application which refers to food ordering system. It is used by Cafeteria manager to manage the food item information, availability, billing information etc. This system is a bunch of benefits from various points of view. This application enables the end-users to select the food items of their choice from the menu list, and order food. All these modules are able to help Cafeteria manager to manage the Cafeteria with more convenience and in a more efficient way as compared to general cafeteria management system which are not computerized.

2.2 Why my system is better?

The main benefits of my Cafeteria Management system are as follows:

- Manage Menu and Costs
- Shows Available food items
- Easy ordering
- Manage billing process
- Record the total profit

CHAPTER 3

METHODOLOGY

3.1 Introduction of Methodology

A methodology or system development methodology in software engineering is a framework that is used to structure, plan and control the process of developing an information system. Without a methodology, we would be neither competitive nor successful. Software engineering is the practice of using selected process techniques to improve the quality of a software development effort. The documented collection of policies, processes and procedures used by a development team or organization to practice software engineering is called its software development methodology.

3.2 Software/Application Development Life Cycle

Software Development Life Cycle is a framework that defines the steps that involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software. Software Development Life Cycle gives us an overview and guidelines to develop quality software. For Cafeteria Management system, we follow SDLC to make it reliable for the user. SDLC stages cover the complete life cycle of a software.

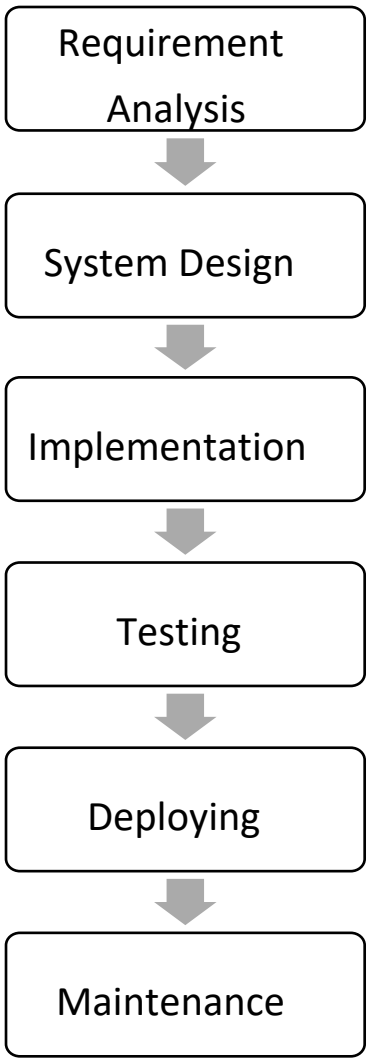


Figure 3.2: Software Development Life Cycle

3.3 Requirement collection and analysis

Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for a new software being built or modified. In software engineering, it is sometimes referred to loosely by names such as requirements gathering or requirements capturing.

Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

RE involves three major frames are:

- Requirement Analysis
- Requirement Specification

- Requirement Validation

In system specification, our first concern is ERD (Entity Relationship Diagram), which shows entity with their attributes, relationship with entities involving one to one, one too many, many to one or many to many, assigning primary keys & perform normalizations to reduce data redundancy.

3.4 Feasibility study

Feasibility Study in Software Engineering is a study to evaluate feasibility of proposed project or system. Feasibility study is one of stage among important four stages of Software Project Management Process. The feasibility study is the second step of the SDLC. It is the stage at which all of the software needs and requirements are written down and thoroughly documented. This document is created in this stage with the assistance of the Software Requirement Specification document. It is referred to as the SRS paper.

3.5 Use case diagram

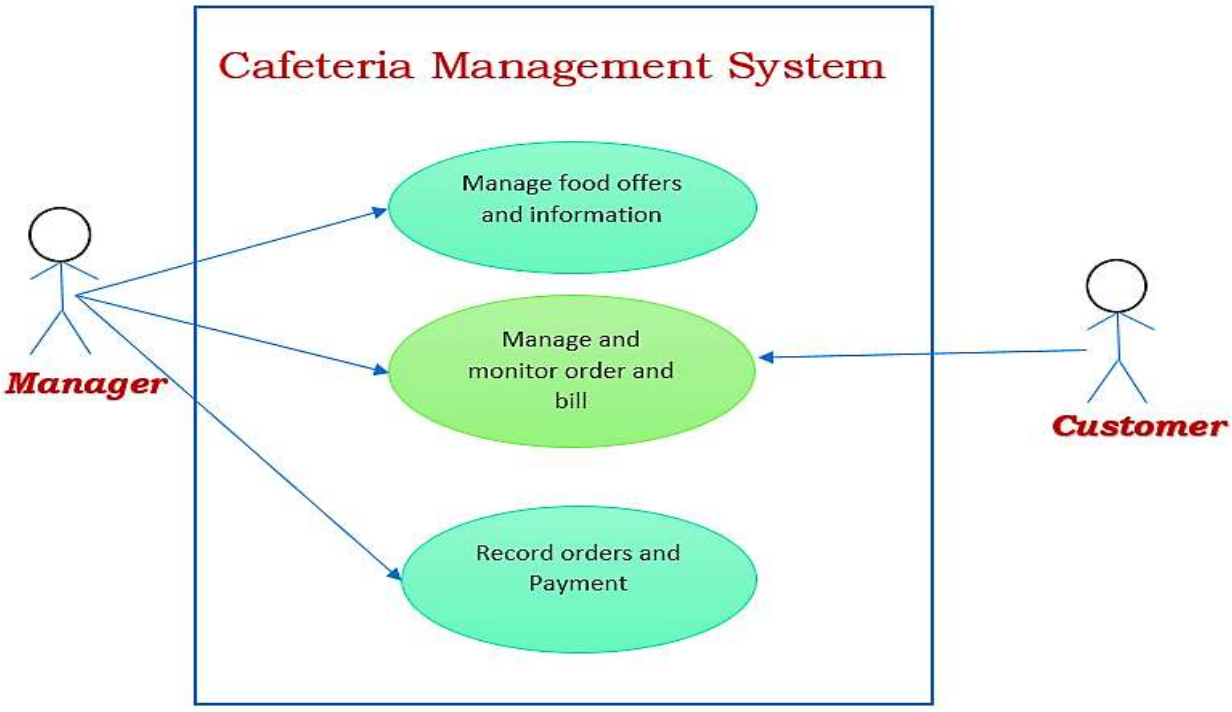


Figure 3.5: Use Case Diagram of Cafeteria Management System

Actor:

An actor is a user of a system in a particular role. An actor can be human or an external object. In the above diagram admin and customer are the actors.

System Boundary:

A System Boundary rectangular separate the Cafeteria management system from the external actors.

Use Case shapes:

The UML Use Case stencil and place them inside the subsystem boundary, and then drag Actor shapes to the outside of the subsystem boundary.

Connector Description:

Connector Description shows the relationship of an actor to a use case. Here, it shows that manage food offers and information is connected to manager.

Generalization:

A use case generalization shows that one use case is simply a special kind of another.

CHAPTER 4

SOFTWARE DESIGN & IMPLEMENTATION

4.1 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a traditional visual graphical representation of the information flows within a system. The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system.

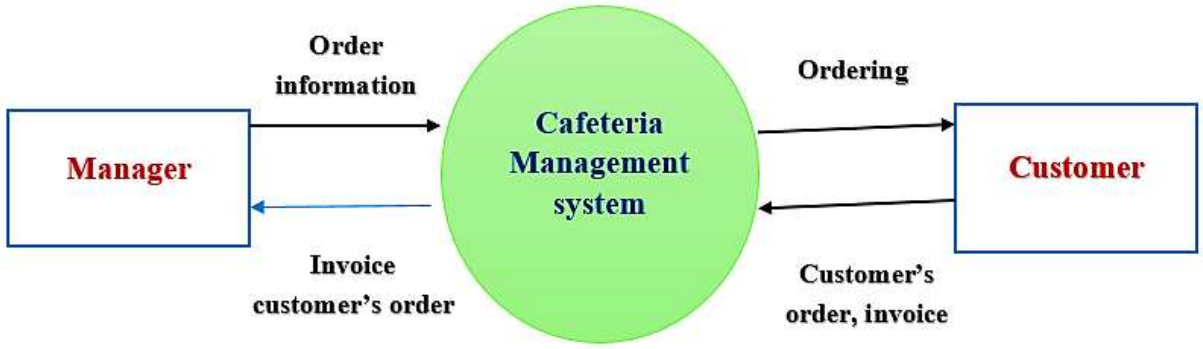


Figure 3.5: Data Flow Diagram of Cafeteria Management System

Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a traditional visual graphical representation of the information flows within a system. It shows how data entered & leaves the system & where it is stored. DFD depicts the graphical representation of the proper amount of system requirement. The objective of DFD is to represent the scopes & boundaries of the whole system.

DFD has set of components to represent destination, source, storage and flow of data, these are:

• **Entities** - Entities are source and destination of information data. Entities are represented by rectangles with their respective names.

• **Process** - Activities and action taken on the data are represented by Circle or Round-edged rectangles.

• **Data Storage** - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.

• **Data Flow** - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

4.2 Entity Relationship Diagram

Entity Relationship Diagram (ERD) is a modeling method of data to produce a conceptual data model of an information system in software Engineering.

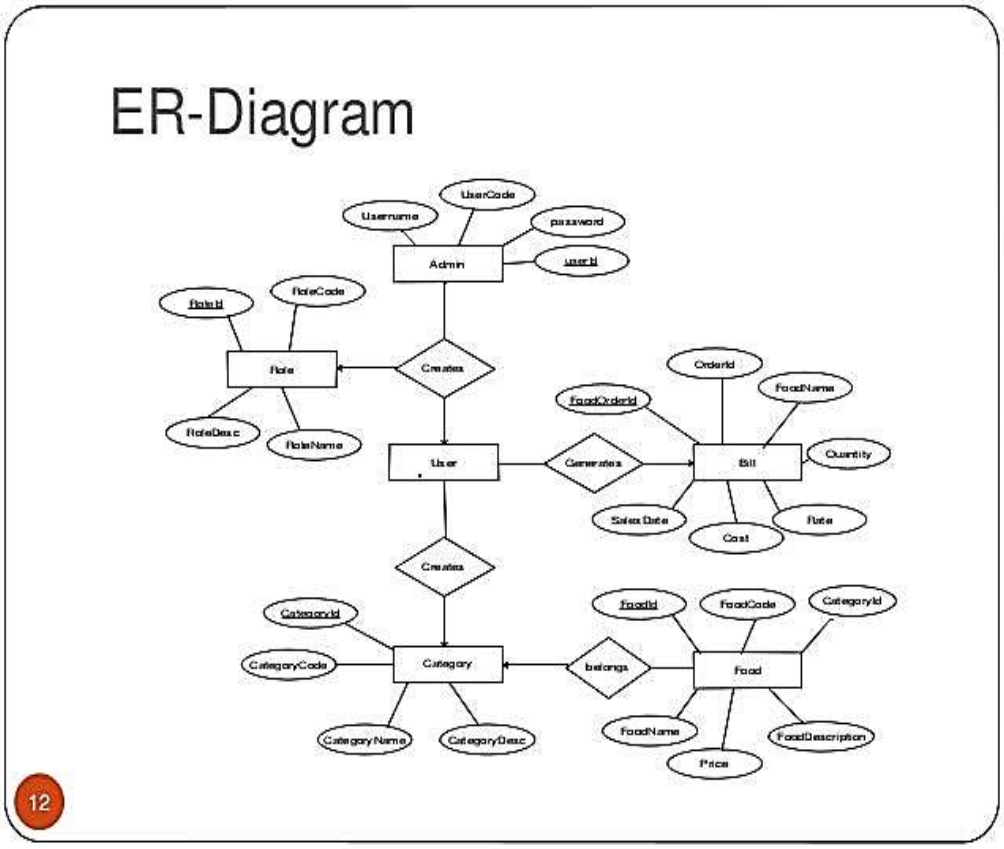


Fig 4.2: ER Diagram of Cafeteria Management System

4.3 Technologies used

My project is only done with C++.

The programming language, C++ has popular uses in: game programming, software engineering, data structures, developing browsers, operating systems, web development, compiler writing, graphics designing, and desktop applications and so on.

4.4 Tools Used

Hardware Tools

- CPU
- Monitor
- Keyboard
- Mouse

Software Tools

Any Windows Operating System, such as

- Visual Studio
- Code Blocks
- Notepad++

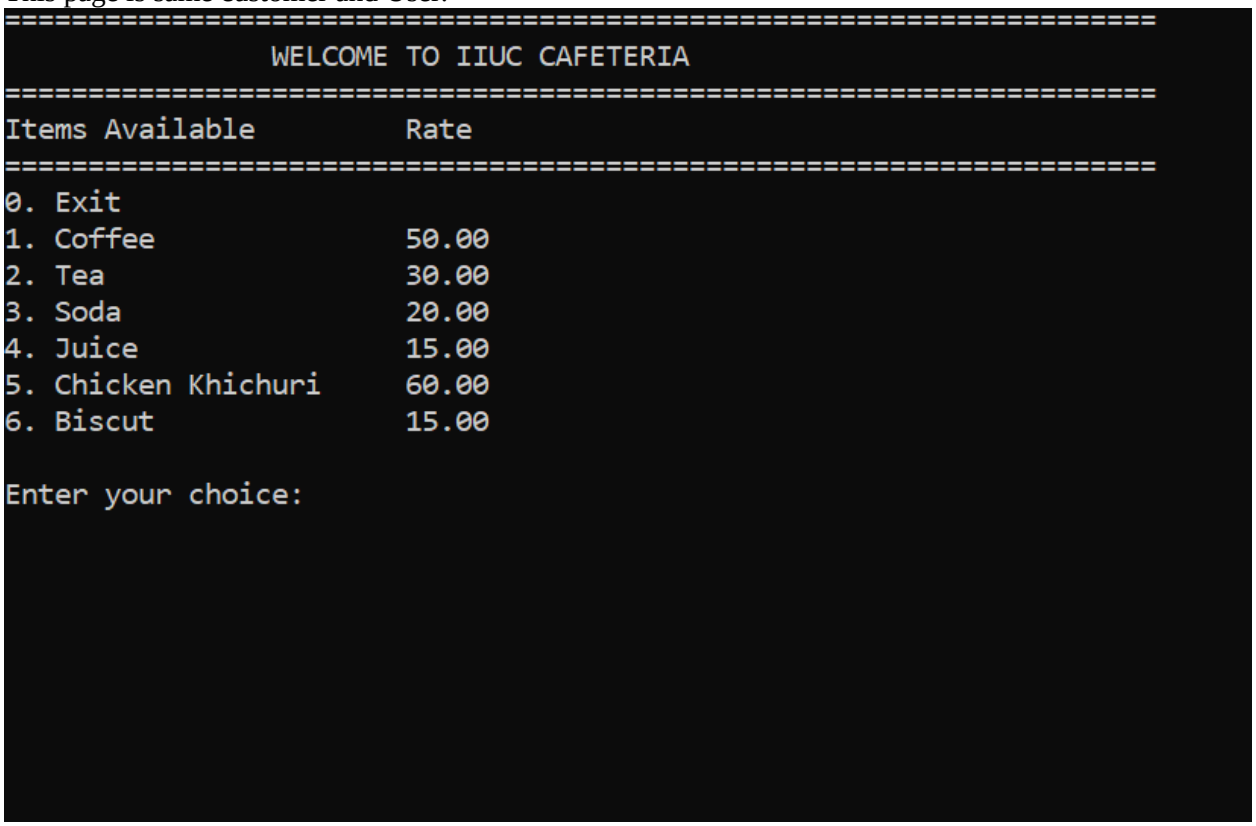
Files & Tools

- Microsoft Word
- Microsoft Office
- Notepad

4.5 Interface Design

Interface design defines the set of interface objects and actions. In our event management system the following steps are done:

1. At first, when the code is run, the page named 'MENU' is shown with all the input ordering options. This page is same customer and User.



2. For ordering customer has to choose their items by giving inputs and their quantity too.

```
=========================================================================
                WELCOME TO IIUC CAFETERIA
=========================================================================
Items Available         Rate
=========================================================================
0. Exit
1. Coffee               50.00
2. Tea                  30.00
3. Soda                 20.00
4. Juice                15.00
5. Chicken Khichuri     60.00
6. Biscut               15.00

Enter your choice: 1

No. of Coffees: 5
```

3. After that it will show an option that if a customer wants to add more

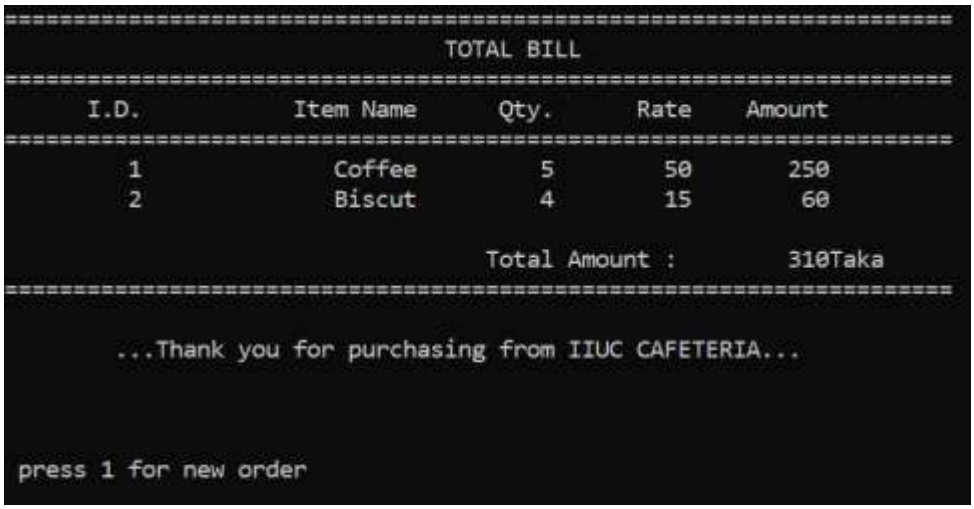
```
=========================================================================
                WELCOME TO IIUC CAFETERIA
=========================================================================
Items Available         Rate
=========================================================================
0. Exit
1. Coffee               50.00
2. Tea                  30.00
3. Soda                 20.00
4. Juice                15.00
5. Chicken Khichuri     60.00
6. Biscut               15.00

Enter your choice: 1

No. of Coffees: 5

Do You Want to Add More (Y/N)? y
```

4. By this option a customer can purchase more or can finish his/her purchasing by pressing N.
5. After pressing N/n the bill will be shown to the customer.



6. After that a new customer can purchase their items by pressing 1, and a new bill will be produced for new purchasing.

7. Admin can Log out from the system by pressing exit (0) option.

4.6 Source code:

```
//IIUC CAFETERIA BY C211211

#include<instream>

#include<stdlib.h>

#include<string.h>

#include<iomanip>

using namespace std;



//Global Variable

float amount=0;                                    //To calculate the total amount.

int size=0;

int i=0;                                           //Total Items Added to Order
```

```
//Structure to store the ordered items & generate bill

struct Items

{

   int id;

   char item[20];

   int qty;

   float rate;

   float amt;

} I[20];


//Function Prototyping

void drawLine();                                        //Function to drawLines for
presentaiton

void Header();                                          //Display Heading in Invoice

void menu();                                            //Displaying Menu of Items

void showBill();                                        //Displaying Invoice


void drawLine()

{

   for(int i=1; i<70; i++)

      cout<<"=";
```

```
    cout<<endl;
}

void Header()
{
    cout<<setw(10)<<"I.D."
        <<setw(20)<<"Item Name"
        <<setw(10)<<"Qty."
        <<setw(10)<<"Rate"
        <<setw(10)<<"Amount"
        <<endl;
}

void showBill()
{
    system("cls");
    drawLine();
    cout<<"\t\t\t\tTOTAL BILL\n";
    drawLine();
    Header();
    drawLine();
```

```
    for(int i=0; i<size; i++)
    {
        cout<<setw(10)<<I[i].id
            <<setw(20)<<I[i].item
            <<setw(10)<<I[i].qty
            <<setw(10)<<I[i].rate
            <<setw(10)<<I[i].amt
            <<endl;
    }
    cout<<endl;
    cout<<setw(50)<<"Total Amount : "
        <<setw(10)<<amount<<"Taka"<<endl;
    drawLine();
    cout<<"\n\t...Thank you for purchasing from IIUC CAFETERIA...\n\n";

}

void menu()
{
    system("cls");
    drawLine();
```

```
    cout<<"\t\tWELCOME TO IIUC CAFETERIA\n";
    drawLine();
    cout<<"Items Available\t\tRate\n";
    drawLine();
    cout<<"0. Exit\n";
    cout<<"1. Coffee\t\t50.00\n";
    cout<<"2. Tea\t\t\t30.00\n";
    cout<<"3. Soda\t\t\t20.00\n";
    cout<<"4. Juice\t\t15.00\n";
    cout<<"5. Chicken Khichuri\t60.00\n";
    cout<<"6. Biscut\t\t15.00\n";
    cout<<"\nEnter your choice: ";
}

int main()
{
    int ch, i=0, qty;
    char choice='n';
start:
    do
    {
```

```
system("cls");
menu();
cin>>ch;
switch(ch)
{
case 0:
   break;
case 1:
   cout<<"\nNo. of Coffees: ";
   cin>>I[i].qty;
   I[i].id=i+1;
   strcpy(I[i].item,"Coffee");
   I[i].rate=50;
   I[i].amt=I[i].qty*I[i].rate;
   amount+=I[i].amt;
   i++;
   size++;
   break;

case 2:
   cout<<"\nNo. of Teas: ";
```

```
    cin>>I[i].qty;
    I[i].id=i+1;
    strcpy(I[i].item,"Tea");
    I[i].rate=30;
    I[i].amt=I[i].qty*I[i].rate;
    amount+=I[i].amt;
    i++;
    size++;
    break;

case 3:
    cout<<"\nNo. of Sodas: ";
    cin>>I[i].qty;
    I[i].id=i+1;
    strcpy(I[i].item,"Soda");
    I[i].rate=20;
    I[i].amt=I[i].qty*I[i].rate;
    amount+=I[i].amt;
    i++;
    size++;
    break;
```

```
case 4:
    cout<<"\nNo. of Juices: ";
    cin>>I[i].qty;
    I[i].id=i+1;
    strcpy(I[i].item,"Juice");
    I[i].rate=15;
    I[i].amt=I[i].qty*I[i].rate;
    amount+=I[i].amt;
    i++;
    size++;
    break;

case 5:
    cout<<"\nNo. of Chicken Khichuri's: ";
    cin>>I[i].qty;
    I[i].id=i+1;
    strcpy(I[i].item,"Chicken Khichuri");
    I[i].rate=60;
    I[i].amt=I[i].qty*I[i].rate;
    amount+=I[i].amt;
```

```
    i++;
   size++;
    break;
case 6:
    cout<<"\nNo. of Biscuts : ";
    cin>>I[i].qty;
    I[i].id=i+1;
    strcpy(I[i].item,"Biscut");
    I[i].rate=15;
    I[i].amt=I[i].qty*I[i].rate;
    amount+=I[i].amt;
    i++;
    size++;
    break;
default:
    break;
}
cout<<"\nDo You Want to Add More (Y/N)? ";
cin>>choice;

if(choice=='Y'||choice=='y')
```

```
        goto start;
    else if(choice!='Y'||choice!='y')
        ch=0;

}
while(ch!=0);

if((choice!='y'||choice!='Y') && amount>0)
    showBill();
amount=0;
size=0;
int ax;
cout<<"\n\n press 1 for new order ";
cin>>ax;
if(ax==1)
{
    main();
    system("cls");
}
else
    cout<<"\nNo Order Placed Yet...\n";
```

```
    system("pause");

}
```

CHAPTER 5

CONCLUSION

5.1 Scope of Future Application

Cafeteria management system still has a lot of future scope to make, some of them are:

- More advance features for the system according to user demand can be done. Food ordering is more advanced, has accuracy & is also cheap.

- At present, this system is fast & convenient. It does not have Credit card facility till now. But build-in banking system is provided in this system for online payment. Thus, it saves time & cost in a way. If the demand increases, these modules can be added in future.

- In case of system failure, there is no option for backup system. We have to implement the backup mechanism. Utmost care and back-up procedures must be established to ensure 100% successful implementation of the computerized ordering system.

5.2 Conclusion

Finding a perfect cafeteria after having particular customer orders is quite time consuming . Here comes the importance of computerized food ordering facility. By this system people will easily book their order in a computerized way. Food purchasing is one of the latest techniques in the arena of internet that allows person to book his/her food items in the shortest time without standing in a rush.In other words, Cafeteria management system is one of the latest facilities of the internet. Purchacing by an app is user friendly, fast & convenient but also very cheap.

References

1. https://phoenixnap.com/blog/software-development-life-cycle#:~:text=Software%20Development%20Life%20Cycle%20is,%2C%20Test%2C%20Deploy%2C%20Maintain.

2. https://www.lucidchart.com/pages/templates/data-flow-diagram-level-0

3. https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/

4. https://www.javatpoint.com/software-engineering-data-flow-diagrams

5. https://www.tutorialspoint.com/software_engineering/software_user_interface_design.htm