|  |
| --- |
| **SIMPLE RESTAURANT BILLING SYSTEM**  **21CSC101T – OBJECT ORIENTED DESIGN AND PROGRAMMING**  **Mini Project Report**  *Submitted by*  **Aman Anand [Reg. No.: RA2211003010130]**  **B.Tech. CSE Section B1**  **SRMIST-01.jpg**  **SCHOOL OF COMPUTING**  **COLLEGE OF ENGINEERING AND TECHNOLOGY**  **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  **(Under Section 3 of UGC Act, 1956)**  S.R.M. NAGAR, KATTANKULATHUR – 603 203  KANCHEEPURAM DISTRICT  **APRIL 2023** |

** COLLEGE OF ENGINEERING & TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE & TECHNOLOGY**

**S.R.M. NAGAR, KATTANKULATHUR - 603203**

**Chengalpattu District**

**BONAFIDE CERTIFICATE**

**Register No\_\_\_\_\_\_\_\_\_\_\_RA2211003010130\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Certified to be the bonafide work done by \_\_\_\_\_\_**Aman Anand**\_\_\_\_\_\_\_\_\_\_\_ of I Year / I Sem B.Tech Degree course in **Programming for Problem Solving 21CSS101J** in **SRM INSTITUTE OF SCIENCE & TECHNOLOGY,** Kattankulathur during the academic year 2022-2023

**DATE : 27-04-2023 LAB INCHARGE**

Mr. Iniyan S,

Associate Professor,

Dept. of CTECH



**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| 1 | Problem Statement | 4 |
| 2 | Methodology / Procedure | 5 |
| 3 | Coding (C++ language) | 6-9 |
| 4 | Sample Input and Output | 10-12 |
| 5 | Conclusion | 13 |
| 6 | UML Diagrams | 14-17 |

**PROBLEM STATEMENT**

* Non existence of an efficient program that quickly generates bills in a restaurant along with the employee details to maintain transparency

**Aim:** To write a C++ Program to prepare a bill for the customer in a restaurant which also displays the employee information.

**Description:**

The advantages of our project are:

1. Efficient billing process: A restaurant billing system can streamline the billing process, making it more efficient and accurate. It can eliminate the need for manual calculations, reducing the chances of errors and providing a more seamless experience for customers.

2. Improved customer service: A billing system can help restaurant staff to process orders quickly and accurately, ensuring that customers receive their bills promptly and without errors. This can improve overall customer satisfaction, leading to repeat business and positive word-of-mouth.

3. Inventory management: A billing system can be integrated with an inventory management system, allowing restaurant managers to keep track of stock levels and monitor food wastage. This can help to reduce costs and improve profitability.

Overall, a restaurant billing system project can help to improve efficiency, customer service. It can be a valuable investment for restaurants of all sizes, helping them to operate more effectively and profitably. Our project can further be expanded to include taxes in the bill.

**Methodology/Procedure**

This program is written in C++ language and implements a simple restaurant billing system. It consists of a class hierarchy with a base class **basicInfo** and a derived class **deptInfo** and employee. The **basicInfo** class contains basic employee information like name, employee ID, and gender. The **deptInfo** class contains department-related information like department name and assigned work. The employee class inherits from both **basicInfo** and **deptInfo** and includes member functions to get and print employee information. This is an example of multiple inheritance.

The main function of the program implements the billing system. It starts by taking employee information using the *getEmployeeInfo* member function of the employee class. It then asks for the date and prompts the user to enter item codes and quantity of each item. The program calculates the unit price and total price for each item and stores them in a two-dimensional array invoice. Finally, it prints the date, employee information, item details, and total bill using the *printEmployeeInfo* and *getUnitprice* functions.

The *getUnitprice* function takes an item code as input and returns its unit price based on a switch statement.

Overall, the program demonstrates the use of class inheritance, member functions, arrays, and switch statements in C++ programming language to implement a simple restaurant billing system.

**CODE**

***C++ Program:***

#include<iostream>

using namespace std;

class basicInfo {

protected:

char name[30];

int empId;

char gender;

public:

void getBasicInfo(void)

{

cout << "Enter Name: ";

cin.getline(name, 30);

cout << "Enter Emp. Id: ";

cin >> empId;

cout << "Enter Gender: ";

cin >> gender;

}

};

// Base Class - deptInfo

class deptInfo {

protected:

char deptName[30];

char assignedWork[30];

public:

void getDeptInfo(void)

{

cout << "Enter Department Name: ";

cin.ignore(1);

cin.getline(deptName, 30);

cout << "Enter assigned work: ";

fflush(stdin);

cin.getline(assignedWork, 30);

}

};

// final class (Derived Class)- employee

class employee : private basicInfo, private deptInfo {

public:

void getEmployeeInfo(void)

{

cout << "Enter employee's basic info: " << endl;

//call getBasicInfo() of class basicInfo

getBasicInfo(); //calling of public member function

cout << "Enter employee's department info: " << endl;

//call getDeptInfo() of class deptInfo

getDeptInfo(); //calling of public member function

}

void printEmployeeInfo(void)

{

cout << "Employee's Information is: " << endl;

cout << "Basic Information...:" << endl;

cout << "Name: " << name << endl; //accessing protected data

cout << "Employee ID: " << empId << endl; //accessing protected data

cout << "Gender: " << gender << endl

<< endl; //accessing protected data

cout << "Department Information...:" << endl;

cout << "Department Name: " << deptName << endl; //accessing protected data

cout << "Assigned Work: " << assignedWork << endl; //accessing protected data

cout << "\n\n-----------------------------\n\n";

}

};

double getUnitprice(int itemCode);

int main(){

employee emp;

emp.getEmployeeInfo();

double invoice[10][4];

int i=0; char more; char date[100];

cout << "\n\n\*\*\*\*\*\*\*\*\* Here are the Item prices for your information\*\*\*\*\*\*\*\*\*\* \n\nItem code\tUnitPrice\n\nPizza\t1\t100\nBurger\t2\t200\nBiryani\t3\t300\nInvalidCode\t0\n\n";

cout << "Enter the date: ";

cin >> date;

do {

cout << "\n\nItem code: ";

cin >> invoice[i][0];

cout << "Quantity : ";

cin >> invoice[i][1];

invoice[i][2] = getUnitprice(invoice[i][0]);

invoice[i][3] = invoice[i][1] \* invoice[i][2];

cout << "Do you have any other items to be entered next (y/n) ? ";

cin >> more;

i++;

} while(more == 'y');

cout << "\n\n-----------------------------\n\n";

cout << "Date : " << date << "\n\n";

cout<<"Pearl Restaurants Pvt. Ltd.\n\n\n";

emp.printEmployeeInfo();

cout << "ItemCode|Quantity|UnitPrice|TotalPrice\n\n";

int tot=0;

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

{

for(int l=0; l<4; l++)

{

cout << invoice[k][l] << "\t\t\t";

}

cout << endl;

tot = tot + invoice[k][3];

}

cout << "\n\nTotal : " << tot<<"\n\n";

cout<<"Thank you!";

cout << "\n\n-----------------------------\n\n\n";

return 0;

}

double getUnitprice(int itemCode){

double price;

switch (itemCode)

{

case 1: price = 100;

break;

case 2: price = 200;

break;

case 3: price = 300;

break;

default: price = 0;

break;

}

return price;

}

**SAMPLE INPUT AND OUTPUT**

Enter employee's basic info:

Enter Name: Raj

Enter Emp. Id: 001

Enter Gender: M

Enter employee's department info:

Enter Department Name: Waiter

Enter assigned work: Order

\*\*\*\*\*\*\*\*\* Here are the Item prices for your information\*\*\*\*\*\*\*\*\*\*

Item code UnitPrice

Pizza 1 100

Burger 2 200

Biryani 3 300

InvalidCode 0

Enter the date: 27/04/2023

Item code: 2

Quantity : 2

Do you have any other items to be entered next (y/n) ? y

Item code: 3

Quantity : 4

Do you have any other items to be entered next (y/n) ? n

-----------------------------

Date : 27/04/2023

Pearl Restaurants Pvt. Ltd.

Employee's Information is:

Basic Information...:

Name: Raj

Employee ID: 1

Gender: M

Department Information...:

Department Name: Waiter

Assigned Work: Order

-----------------------------

ItemCode|Quantity|UnitPrice|TotalPrice

2 2 200 400

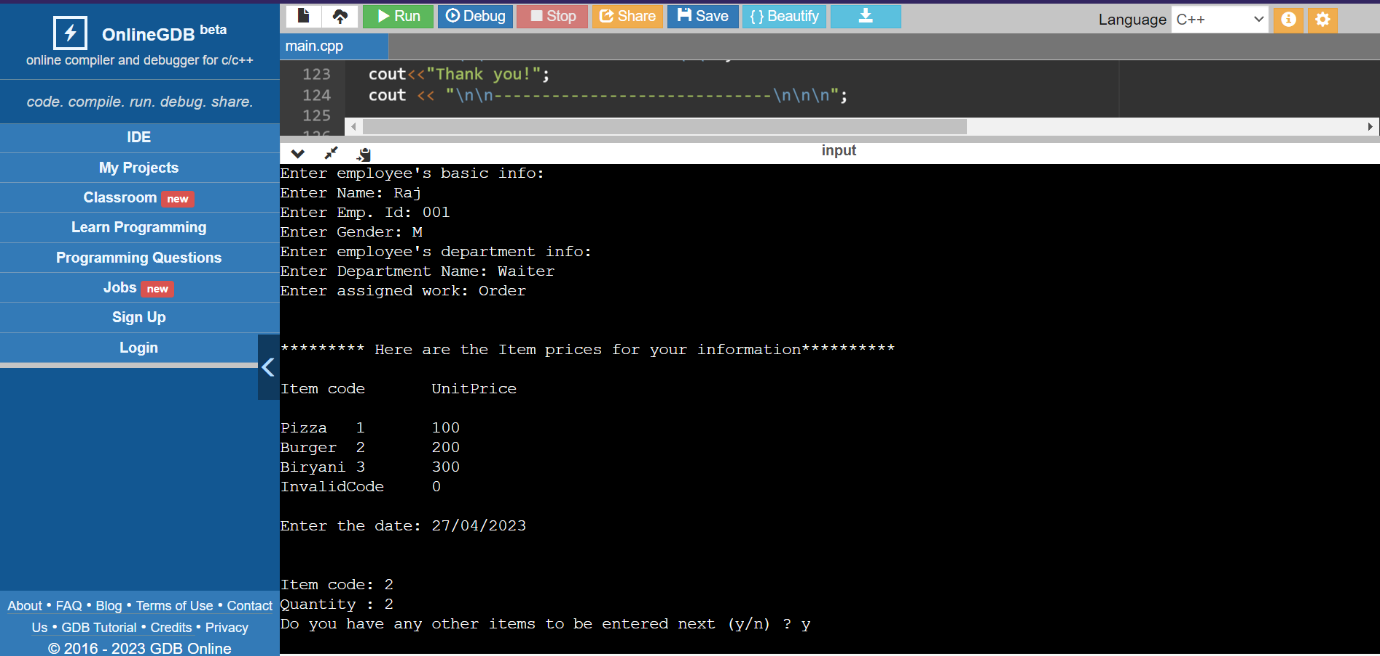
3 4 300 1200

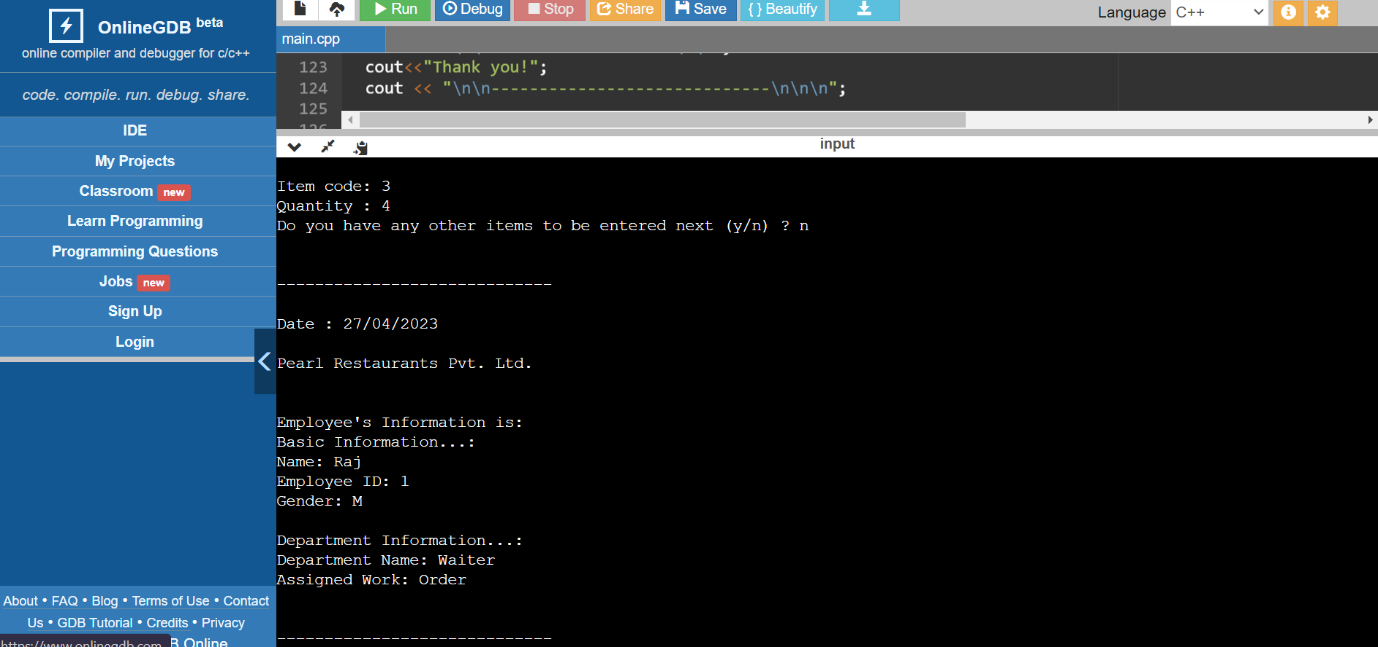
Total : 1600

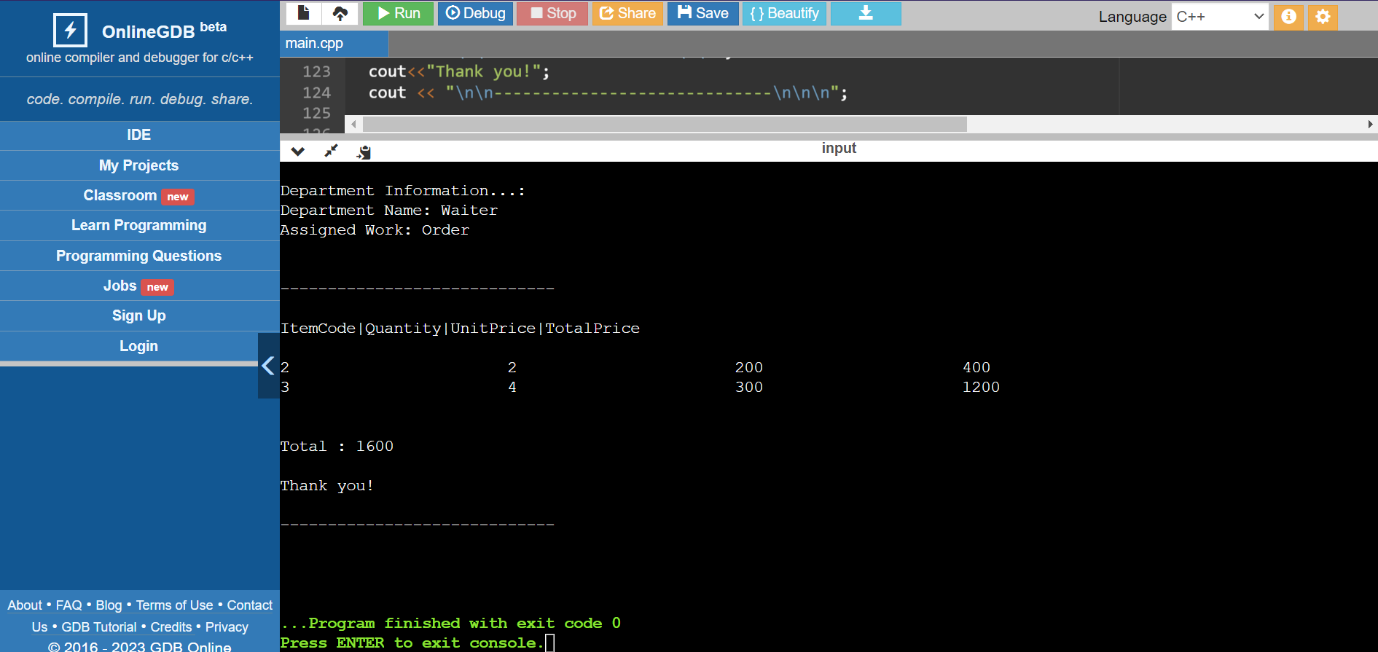
Thank you!

-----------------------------

PICTURES OF SAMPLE INPUT AND OUTPUT



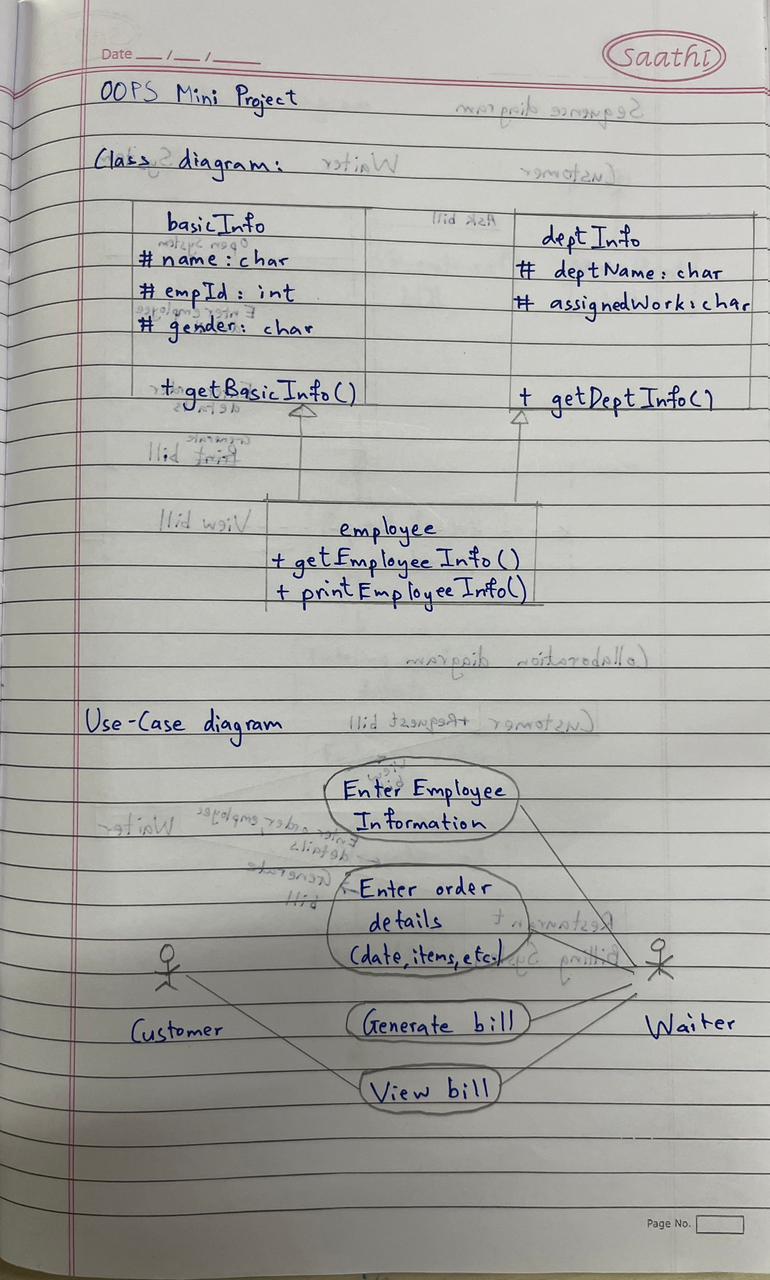


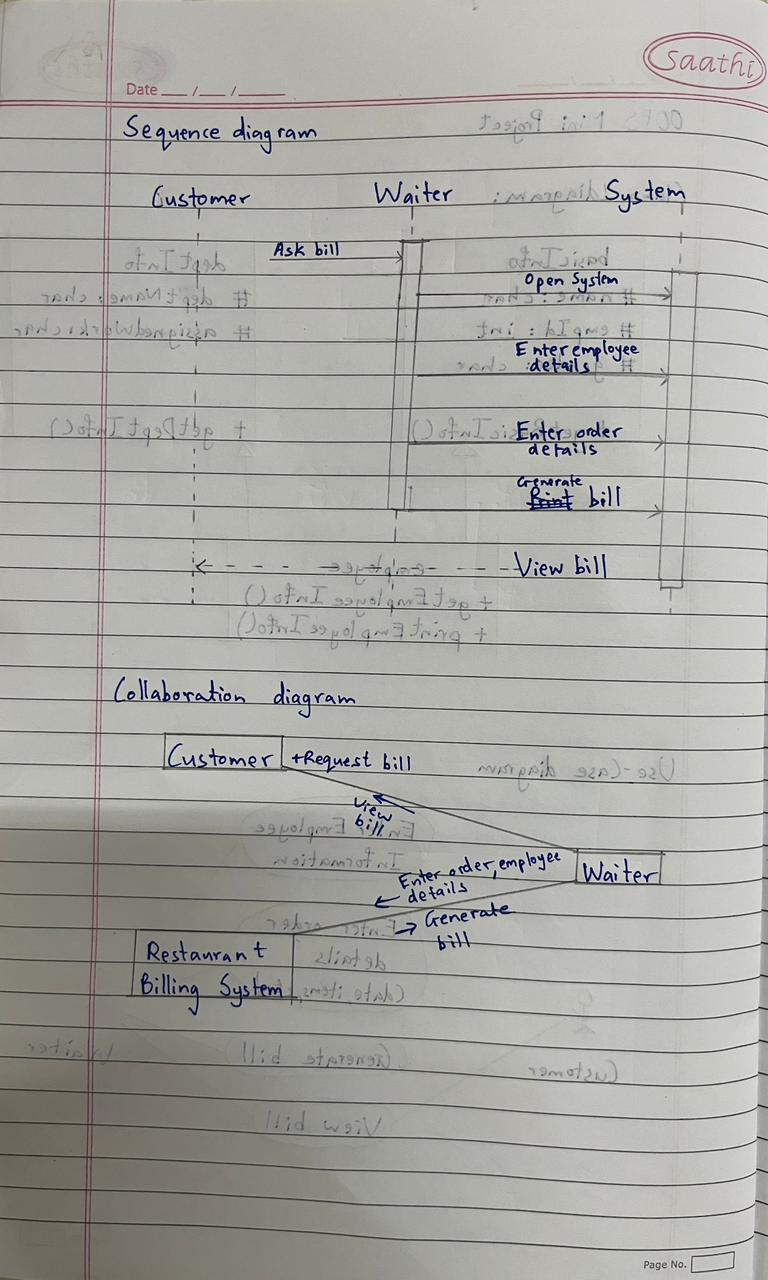


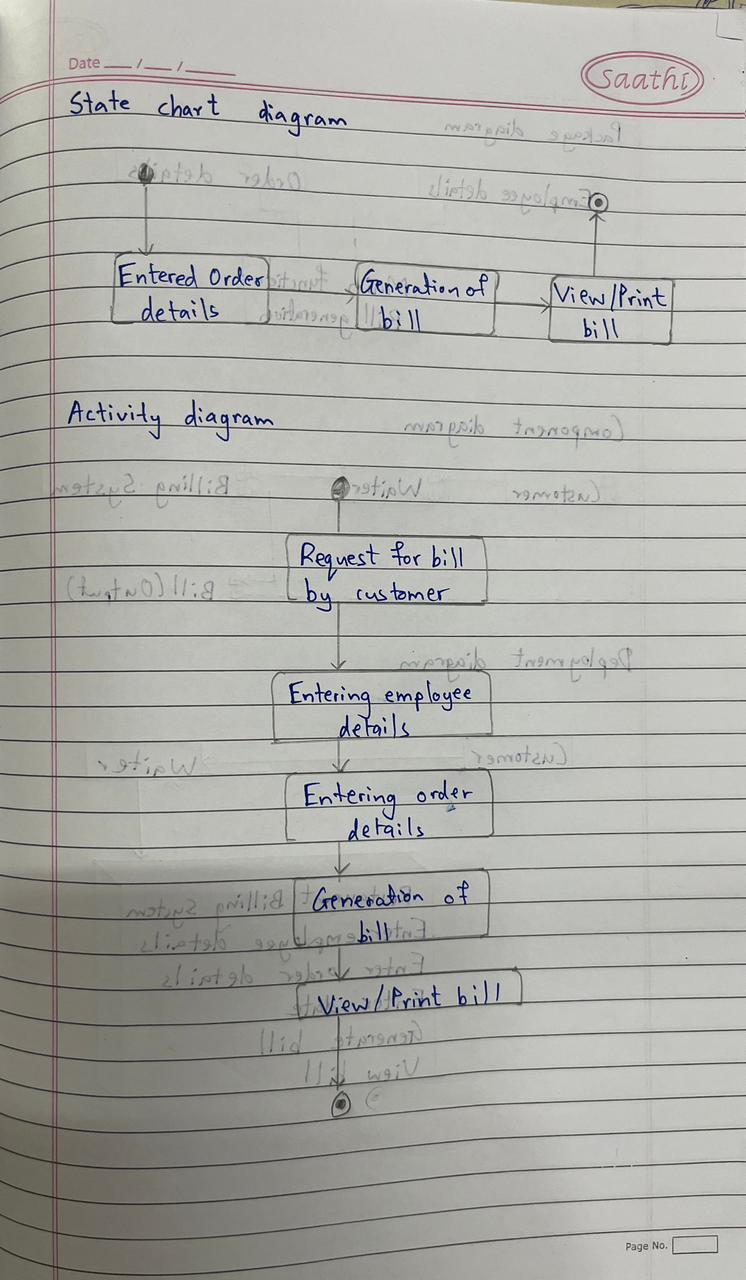
**Conclusion**

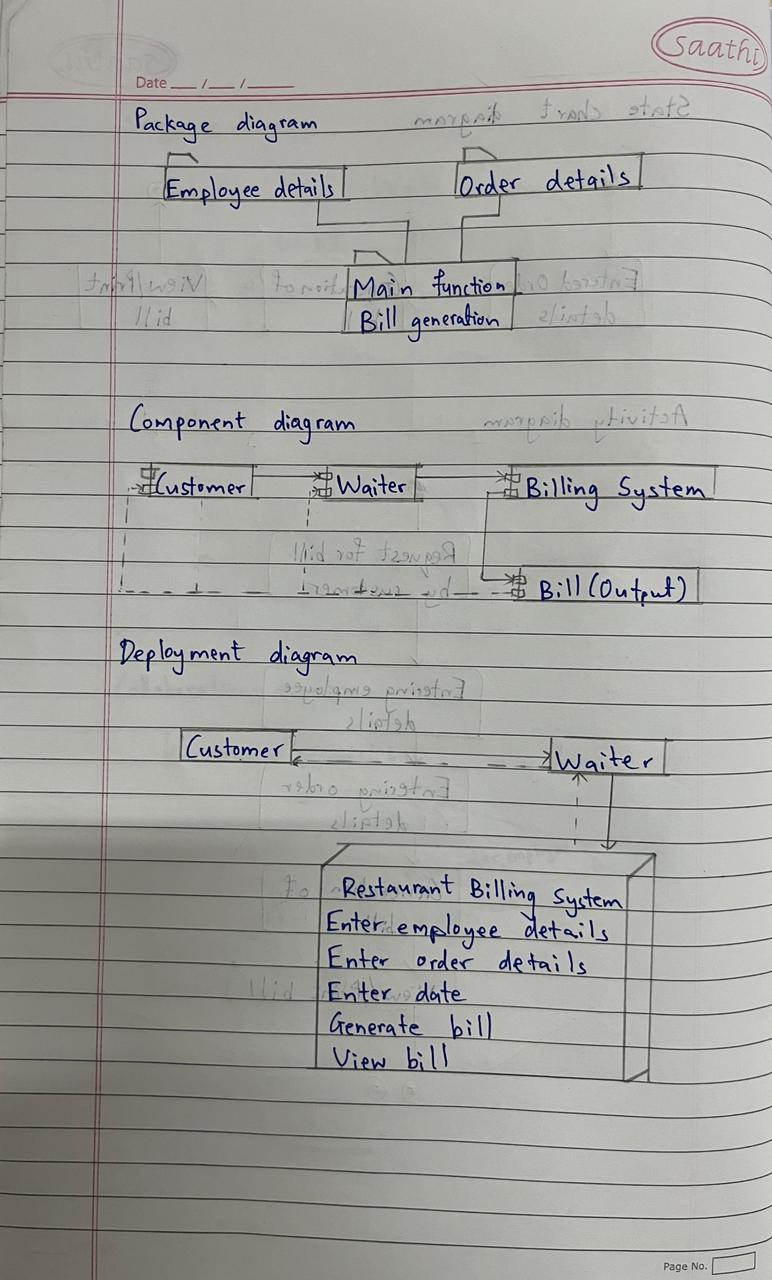
In conclusion, the simple restaurant billing system project presented here is a basic implementation of a program that calculates the total price of an order based on the quantity and price of each item. The program takes inputs such as employee information, date of the order, item code, and quantity, and calculates the total price of the order using a simple formula. This project can be further extended by including additional features such as printing a receipt, storing transaction data, and generating reports or adding taxes. Overall, this project provides a good starting point for beginners to understand the concepts of programming and software development

**UML DIAGRAMS**









**THANK YOU!**