Customer Billing System

Description: A customer billing system is a software program or application that is used by businesses to manage and track the billing and payment processes for their customers. The system typically includes a database of customer information, invoice creation and management tools, and payment processing capabilities.

The system usually starts with the creation of a customer account, which includes basic information such as the customer's name, address, and contact details. Once the account is set up, the system allows businesses to create invoices and send them to customers for payment. The invoices typically include details of the products or services purchased, the total amount due, and payment terms and deadlines.

The customer billing system can also track payments made by customers and generate reports on accounts receivable, overdue invoices, and payment histories. Businesses can use this information to manage their cash flow and make informed decisions about their billing and payment processes.

In addition to basic billing functionality, some customer billing systems also offer features such as automated recurring billing, online payment processing, and integration with other business systems such as accounting software and customer relationship management (CRM) tools.

Overall, a customer billing system is an essential tool for businesses that want to streamline their billing and payment processes, reduce errors and delays, and provide a better customer experience.

Purpose:

The purpose of the system is to streamline the billing process, reduce manual errors, and provide accurate billing information to customers.

Modules:

The customer billing system will consist of the following modules:

Item Display Module

This module will display the list of available items on the menu. It will also show the price of each item and the discount if applicable

Item Selection Module

This module will allow the customer to select the items they wish to order. It will also enable the addition of the selected

item to the bill

Item Deletion Module.

This module will enable the customer to delete any selected items from the bill.
Discount Module.

This module will provide discounts to the customer based on certain conditions. For example, a discount can be applied if the customer purchases a specific combination of items.

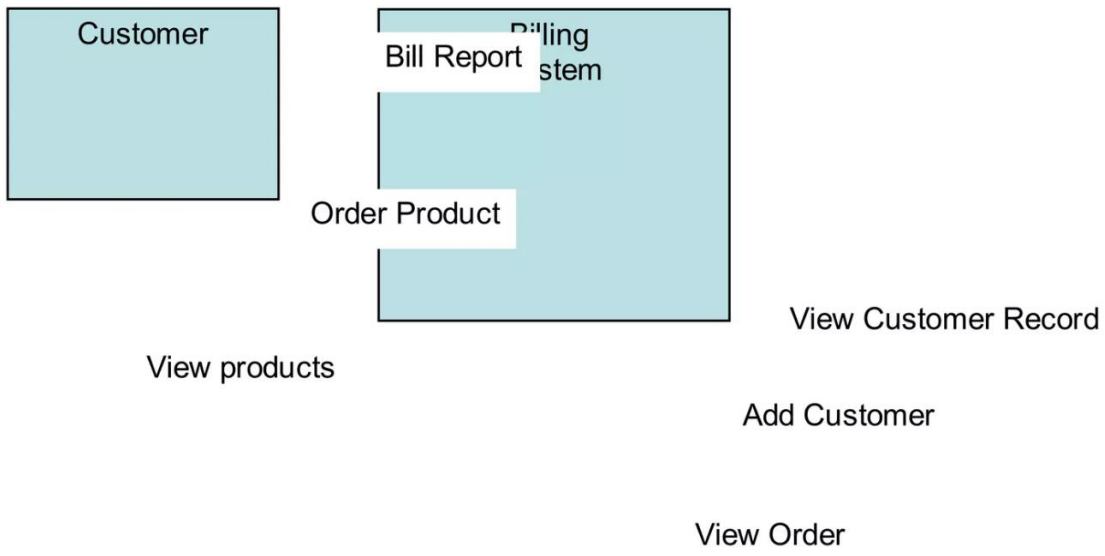
Payment Module

This module will enable the customer to pay for their order through various payment modes such as cash, card, or online payment

Reporting Module:

This module will generate reports that will include the details of the items sold, discounts applied, payment modes used, and other relevant information.

DFD level 0:



Programming Code :

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

struct items{
    char item[20];
    float price;
    int qty;
```

```
};

struct orders{
    char customer[50];
    char date[50];
    int numOfItems;
    struct items itm[50];
};
//functions to generate biils
void generateBillHeader(char name[50],char date[30]){
    printf("\n\n");


        printf("\n\t   -----------------");
        printf("\nDate:%s",date);
        printf("\nInvoice To: %s",name);
        printf("\n");
        printf("---------------------------------------\n");
        printf("Items\t\t");
        printf("Qty\t\t");
        printf("Total\t\t");
        printf("\n---------------------------------------");
        printf("\n\n");
}
void generateBillBody(char item[30],int qty, float price){
    printf("%s\t\t",item);
        printf("%d\t\t",qty);
        printf("%.2f\t\t",qty * price);
        printf("\n");
}



void generateBillFooter(float total){
    printf("\n");
```

```
    float dis = 0.1*total;
    float netTotal=total-dis;
    float cgst=0.09*netTotal,grandTotal=netTotal + 2*cgst;//netTotal
+ cgst + sgst
    printf("---------------------------------------\n");
    printf("Sub Total\t\t\t%.2f",total);
    printf("\nDiscount @10%s\t\t\t%.2f","%",dis);
    printf("\n\t\t\t\t-------");
    printf("\nNet Total\t\t\t%.2f",netTotal);
    printf("\nCGST @9%s\t\t\t%.2f","%",cgst);
    printf("\nSGST @9%s\t\t\t%.2f","%",cgst);
    printf("\n---------------------------------------");
    printf("\nGrand Total\t\t\t%.2f",grandTotal);
    printf("\n---------------------------------------\n");
}
int main(){

    int opt,n;
    struct orders ord;
    struct orders order;
    char saveBill = 'y',contFlag = 'y';
    char name[50];
    FILE *fp;
      //dashboard
    while(contFlag == 'y'){
    system("clear");
    float total = 0;
    int invoiceFound = 0;
    printf ( "\t=======lovely sweets=======");
    printf("\n\nPlease select your prefered operation");
    printf("\n\n1.Generate Invoice");
    printf("\n2.Show all Invoices");
    printf("\n3.Search Invoice");
    printf("\n4.Exit");
```

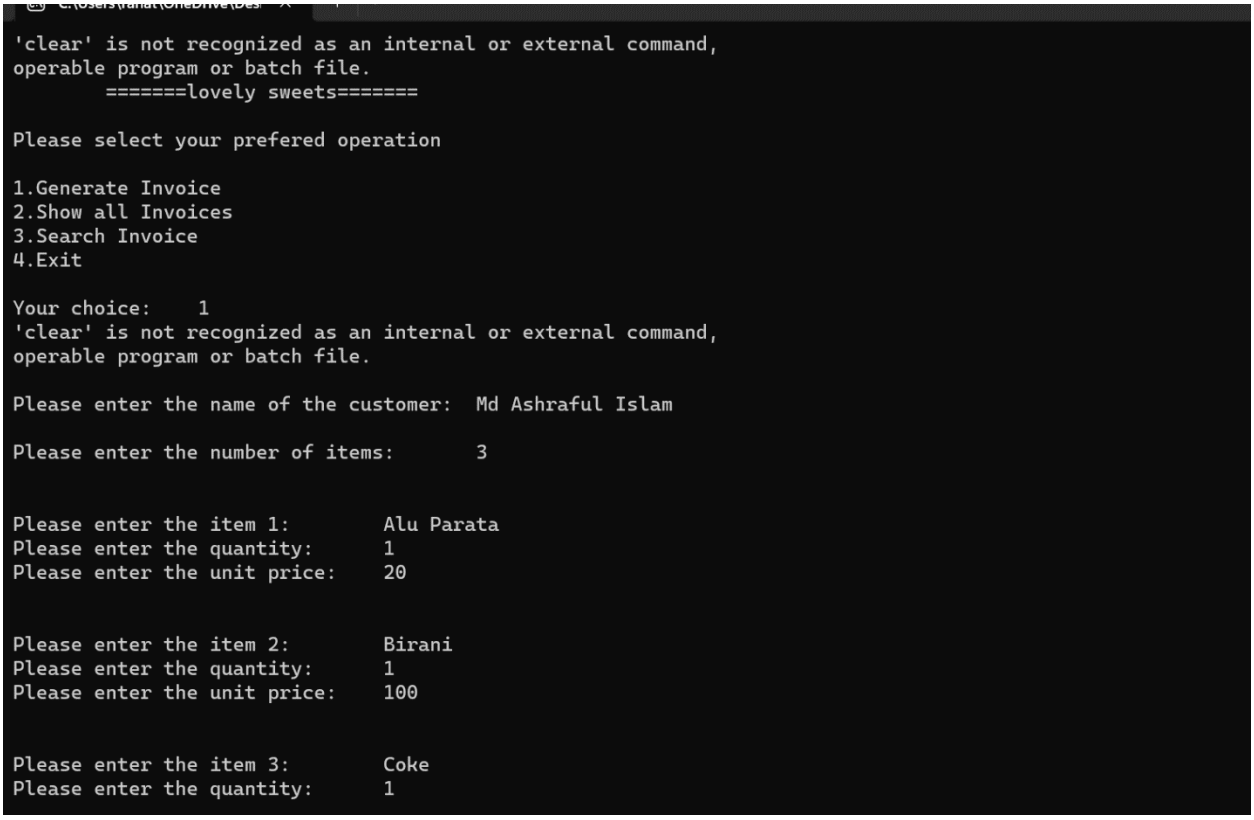
```
    printf("\n\nYour choice:\t");
    scanf("%d",&opt);
    fgetc(stdin);
    switch(opt){
        case 1:
        system("clear");
        printf("\nPlease enter the name of the customer:\t");
        fgets(ord.customer,50,stdin);
        ord.customer[strlen(ord.customer)-1] = 0;
        strcpy(ord.date,_DATE_);
        printf("\nPlease enter the number of items:\t");
        scanf("%d",&n);
        ord.numOfItems = n;
        for(int i=0;i<n;i++){
            fgetc(stdin);
            printf("\n\n");
            printf("Please enter the item %d:\t",i+1);
            fgets(ord.itm[i].item,20,stdin);
            ord.itm[i].item[strlen(ord.itm[i].item)-1]=0;
            printf("Please enter the quantity:\t");
            scanf("%d",&ord.itm[i].qty);
            printf("Please enter the unit price:\t");
            scanf("%f",&ord.itm[i].price);
            total += ord.itm[i].qty * ord.itm[i].price;
        }

        generateBillHeader(ord.customer,ord.date);
        for(int i=0;i<ord.numOfItems;i++){

generateBillBody(ord.itm[i].item,ord.itm[i].qty,ord.itm[i].price);
        }
        generateBillFooter(total);

        printf("\nDo you want to save the invoice [y/n]:\t");
        scanf("%s",&saveBill);
```

```
        if(saveBill == 'y'){
            fp = fopen("RestaurantBill.dat","a+");
            fwrite(&ord,sizeof(struct orders),1,fp);
            if(fwrite != 0)
            printf("\nSuccessfully saved");
            else
            printf("\nError saving");
            fclose(fp);
        }
        break;

        case 2:
        system("clear");
        fp = fopen("RestaurantBill.dat","r");
        printf("\n  **Your Previous Invoices**\n");
        while(fread(&order,sizeof(struct orders),1,fp)){
            float tot = 0;
            generateBillHeader(order.customer,order.date);
            for(int i=0;i<order.numOfItems;i++){

generateBillBody(order.itm[i].item,order.itm[i].qty,order.itm[i].price)
;
                tot+=order.itm[i].qty * order.itm[i].price;
            }
            generateBillFooter(tot);
        }
        fclose(fp);
        break;

        case 3:
        printf("Enter the name of the customer:\t");
        //fgetc(stdin);
        fgets(name,50,stdin);
        name[strlen(name)-1] = 0;
```

```
        system("clear");
        fp = fopen("RestaurantBill.dat","r");
        printf("\t**Invoice of %s**",name);
        while(fread(&order,sizeof(struct orders),1,fp)){
            float tot = 0;
            if(!strcmp(order.customer,name)){
            generateBillHeader(order.customer,order.date);
            for(int i=0;i<order.numOfItems;i++){

generateBillBody(order.itm[i].item,order.itm[i].qty,order.itm[i].price)
;
                tot+=order.itm[i].qty * order.itm[i].price;
            }
            generateBillFooter(tot);
            invoiceFound = 1;
            }

        }
        if(!invoiceFound){
            printf("Sorry the invoice for %s doesnot exists",name);
        }
        fclose(fp);
        break;

    case 4:
    printf("\n\t\t Bye Bye :)\n\n");
    exit(0);
    break;

    default:
    printf("Sorry invalid option");
    break;
    }
    printf("\nDo you want to perform another operation?[y/n]:\t");
    scanf("%s",&contFlag);
```

```
    }
    printf("\n\t\t Bye Bye :)\n\n");
    printf("\n\n");

    return 0;
}
```

Output Snapshots:



```
'clear' is not recognized as an internal or external command,
operable program or batch file.
        =======lovely sweets=======

Please select your prefered operation

1.Generate Invoice
2.Show all Invoices
3.Search Invoice
4.Exit

Your choice:    1
'clear' is not recognized as an internal or external command,
operable program or batch file.

Please enter the name of the customer:  Md Ashraful Islam

Please enter the number of items:       3


Please enter the item 1:        Alu Parata
Please enter the quantity:      1
Please enter the unit price:    20


Please enter the item 2:        Birani
Please enter the quantity:      1
Please enter the unit price:    100


Please enter the item 3:        Coke
Please enter the quantity:      1
```

```
Please enter the item 3:        Coke
Please enter the quantity:      1
Please enter the unit price:    20



          -----------------
Date:Mar 22 2023
Invoice To: Md Ashraful Islam
--------------------------------------
Items           Qty             Total
--------------------------------------

Alu Parata              1               20.00
Birani          1               100.00
Coke            1               20.00

--------------------------------------
Sub Total                       140.00
Discount @10%                   14.00
                                -------
Net Total                       126.00
CGST @9%                        11.34
SGST @9%                        11.34
--------------------------------------
Grand Total                     148.68
--------------------------------------

Do you want to save the invoice [y/n]:
```

Conclusion:

The customer billing system is designed to streamline the billing process and reduce errors. It will provide accurate billing information to the customers and generate reports for management. The system will consist of various modules, including item display, item selection, item deletion, discount, payment, and reporting.