



Topic : Online Customer Support System

Group no : MLB_6.01_05

Campus : Malabe

Submission Date : 19.05.2022

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT21310546	Hewamadduma. N. A	0764060449
IT21309106	Imasha. P. U	0716833969
IT21308598	Mahawaththa.N.T.M.A.S.M	0753558647
IT21307980	Rodrigo.W.S. A	0772401139
IT21312694	Sandeeptha.P.K. T	0719871480

Exercise 01

Requirements

1. Customers can register to the system by giving their details such as name, address, email, contact number.
2. Customer can update user details and view current details.
3. Customer can search for products.
4. Customer can add products to the cart after searching for the required products.
5. Customer can select services for the ordering issues.
6. Customer can place the order.
7. Customer can select deliver method
8. Customer can pay bills.
9. Customer specifies a payment method for each order such as Visa cards and Credit cards.
10. Customer can cancel the order.
11. User can give feedback about the service.

Noun Verb Analysis

❖ Yellow color words indicate Nouns

❖ Blue color words indicate Verbs

1. Customer can register to the system by giving their details such as name, address, email, contact number.
2. Customer can update user details and view current details.
3. Customer can search for items.
4. Customer can add items to the cart after searching for the required items.
5. Customer can select services for the ordering issues.
6. Customer can place the order.
7. Customer can select deliver method.
8. Customer can pay bills.
9. Customer specifies a payment method for each order such as Visa cards and Credit cards.
10. Customer can cancel the order.
11. User can give feedback about the service.

Nouns

- Customer
- User
- System
- User Details
- Current Details
- Name
- Address
- Email
- Contact Number
- Current Details
- Items
- Cart
- Required products
- Services
- Ordering Issues
- Order
- Deliver
- Bills
- Payment
- Visa Cards
- Credit Card
- Feedback

Class

- Customer
- Cart
- Item
- Services
- Order
- Bills
- Deliver
- Payment
- Feedback

Redundant

- User

Out Of Scope

- System
- Required Products

Attributes

- Name
- Address
- Email
- Contact Number
- Visa Cards
- Credit Cards

Verbs

- Register
- Giving
- Update user details
- View current details
- Search products
- Add products
- Searching for the required products
- Select services
- Place order
- Pay bills
- Cancel order
- Give feedback

Customers

- Register
- Giving
- Update user details
- View current details
- Select services
- Place order
- Pay bills
- Cancel order
- Give feedback

Items

- Search items
- Searching for the required items

Cart

- Add products

Exercise 02

CRC Cards

Class Name: Customer	
Responsibilities	Collaborators
Register	
Update user details	
Display user details	
Place order	Order
Make payment	Payment

Class Name: Cart	
Responsibilities	Collaborators
Add cart details	Product
Display cart details	

Class Name: Items	
Responsibilities	Collaborators
Add item Details	
Display item details	

Class Name: Services	
Responsibilities	Collaborators
Add service details	
Update service details	

Class Name: Order	
Responsibilities	Collaborators
Add order details	Product, Deliver
Delete order details	Product, Payment
Display order details	Customer, Product, Payment

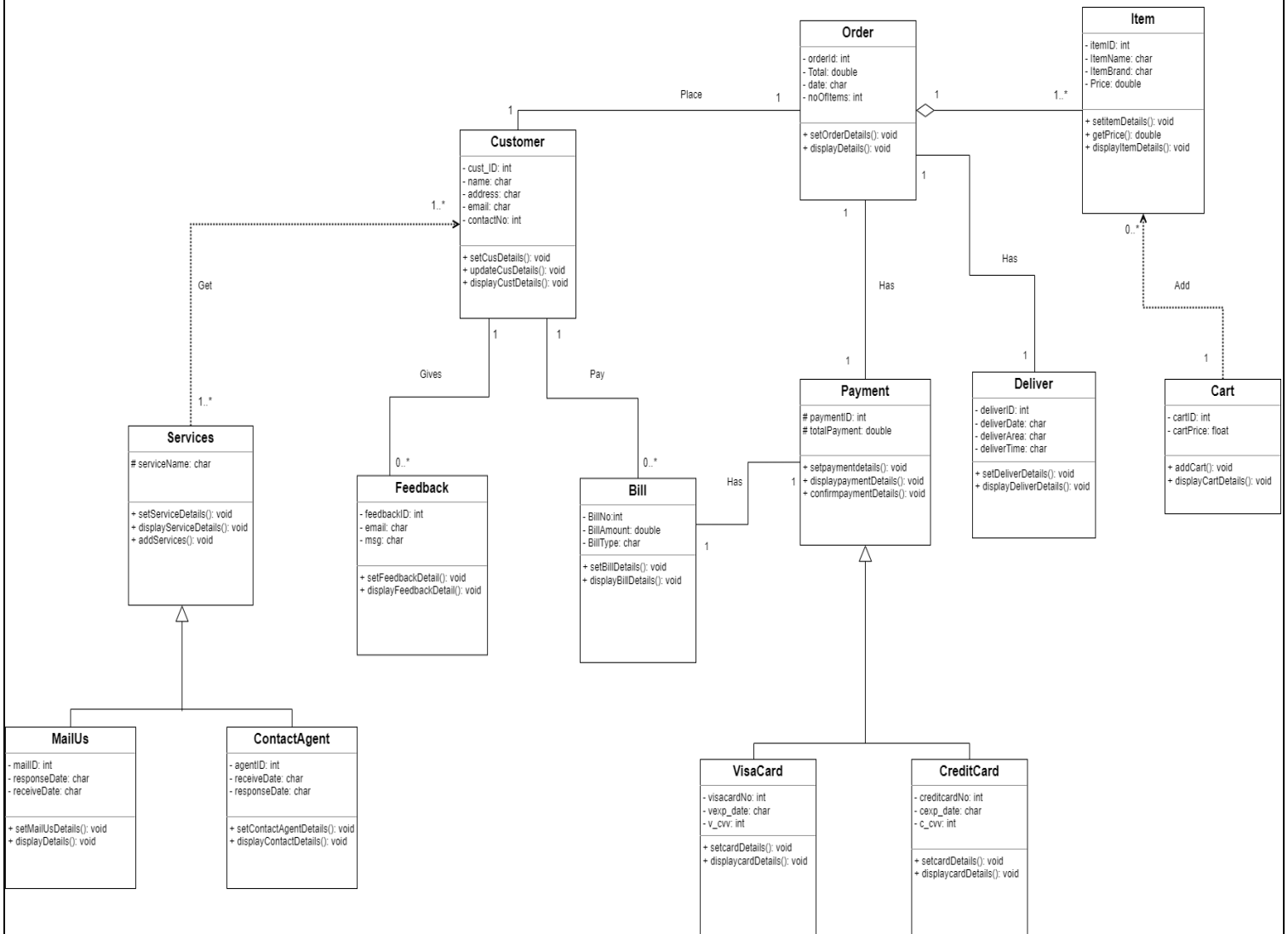
Class Name: Bills	
Responsibilities	Collaborators
Add bill details	Customer
Display Bill details	

Class Name: Deliver	
Responsibilities	Collaborators
Add deliver details	Customer, Order, Payment
Display delivers details	

Class Name: Payment	
Responsibilities	Collaborators
Add payment details	
Confirm payment	Customer
Display payment details	

Class Name: Feedback	
Responsibilities	Collaborators
Get feedback	
Display feedback	

Class Diagram



Codes

item.h

```
class Item
{
    private:
        int itemID;
        char IName[20];
        char IBrand[20];
        double Price;

    public:
        Item();
        Item(int iID, char iName[], char iBrand[], double price);
        void setItemDetails(int ID, char Name[], char Brand[], double lprice);
        double getPrice();
        void displayItemDetails();
        ~Item();
};
```

item.cpp

```
#include<iostream>
#include<cstring>
#include "item.h"
using namespace std;

Item :: Item()
{
    itemID=0;
    strcpy(IName,"");
    strcpy(IBrand,"");
    Price=0;
}

Item :: Item(int iID, char iName[], char iBrand[], double price)
{
    itemID=iID;
    strcpy(IName,iName);
    strcpy(IBrand,iBrand);
}
```

```

    Price=price;
}

void Item :: setItemDetails(int ID, char Name[], char Brand[], double lprice)
{
    itemID=ID;
    strcpy(IName,Name);
    strcpy(IBrand,Brand);
    Price=lprice;
}

double Item :: getPrice()
{
    return Price;
}

void Item:: displayItemDetails()
{
    cout << endl;
    cout << "Item ID = " << itemID << endl;
    cout << "Item Name = " << IName << endl;
    cout << "Item Brand = " << IBrand << endl;
    cout << "Item price = " << Price << endl;

}

Item :: ~Item()
{
    cout << endl;
    cout << "Deleted" << endl;
}

```

order.h

```
#include<cstring>
#include"deliver.h"
#include"item.h"
#define SIZE 2

using namespace std;

class Order{
private:
    int orderId;
    double Total;
    char date[10];
    int noOfItems;
    Deliver*deliver;
    Item*item[SIZE];

public:
    Order();
    Order(int oid, float total, char Date[10], int Nitems,Deliver*dev);
    void addItem(Item*item1, Item*item2);
    void displayItem();
    void setOrderDetails(int OID,float Ototal, char Odate[10], int nItems,Deliver*cDev);
    void displayDetails();
    ~Order();
};
```

order.cpp

```
#include<iostream>
#include<iomanip>
#include<cstring>
#include"order.h"
#include"item.h"
#define SIZE 2
using namespace std;

Order::Order()
{
    int orderId=0;
    float Total=0.00;
```



```
char date=0;
int noOfItem=0;
}
```

```
Order::Order(int oid, float total, char Date[10], int Nitems, Deliver*dev)
{
    orderId=oid;
    Total=total;
    strcpy(date,Date);
    noOfItems=Nitems;
    deliver=dev;
}
```

```
void Order::setOrderDetails(int OID,float Ototal, char Odate[10],int nItems,Deliver*cDev)
{
    orderId=OID;
    Total=Ototal;
    strcpy(date,ODate);
    noOfItems=nItems;
    deliver=cDev;
}
```

```
void Order::addItem(Item*item1,Item*item2)
{
    item[0]=item1;
    item[1]=item2;
}
```

```
void Order::displayItem()
{
    for(int r=0; r<SIZE;r++)
    {
        item[r]->getItemDetails();
    }
}
```

```
void Order::displayDetails()
{
    cout<<"Order Id="<<orderId<<endl;
    cout<<"Total="<<Total<<endl;
    cout<<"Date="<<date<<endl;
    cout<<"No Of Items="<<noOfItems<<endl;
}
```

```
Order::~~Order()
{
    cout<<endl;
    cout<<"Order Cancelled"<<endl;
}
```

cart.h

```
#include<iomanip>
#include"item.h"

class Cart{
    private:
        int cartID;
        float cartPrice;

    public:
        Cart();
        Cart(int CartID);
        void addCart(int qty, Item*item);
        void displayCartDetails();
        ~Cart();
};
```

cart.cpp

```
#include<iostream>
#include<iomanip>
#include"cart.h"
#include"item.h"
using namespace std;

Cart::Cart(){}

Cart::Cart(int CartID)
{
    cartID=CartID;
    cartPrice=0;
}

void addCart(int qty, Item*item)
```

```
{
    cartPrice=qty*item->getPrice();
}

void Cart::displayCartDetails()
{
    cout<<"Cart ID:"<<cartID<<endl
    cout<<"Cart Price:"<<setiosflags(ios::fixed)<<setprecision(2)<<cartPrice<<endl;
}

Cart::~~Cart()
{
    cout<<endl;
    cout<<"Cancel"<<endl;
}
```

deliver.h

```
#include<cstring>
#include"order.h"

class Deliver{
private:
    int deliverID;
    char deliverDate[10];
    char deliverArea[20];
    char deliverTime[5];
    Order*order;

public:
    Deliver();
    Deliver (int DID, char DDate[10],char DArea[20], char DTime[5],Order*pOrder);
    void setDeliverDetails(int dID, char dDate[10], char dArea[20],char dTime[5],Order*cOrder);
    void displayDeliverDetails();
    ~Deliver();
};
```

deliver.cpp

```
#include<iostream>
#include<cstring>
#include"deliver.h"
using namespace std;

Deliver::Deliver()
{
    int deliverID=00;
    char deliverDate;
    char deliverArea;
    char deliverTime;
}

Deliver:: Deliver(int DID, char DDate[10],char DArea[20], char DTime[5],Order*pOrder)
{
    deliverID=DID;
    strcpy(deliverDate,DDate);
    strcpy(deliverArea,DArea);
```

```
    strcpy(deliverTime,DTime);  
    order=pOrder;  
}
```

```
void Deliver::setDeliverDetails(int dID, char dDate[10], char dArea[20],char dTime[5],Order*cOrder)  
{  
    deliverID=dID;  
    strcpy(deliverDate,dDate);  
    strcpy(deliverArea,dArea);  
    strcpy(deliverTime,dTime);  
    order=cOrder;  
}
```

```
void Deliver::displayDeliverDetails()  
{  
    cout<<"Deliver ID:"<<deliverID<<endl;  
    cout<<"Deliver Date:"<<deliverDate<<endl;  
    cout<<"Deliver Area:"<<deliverArea<<endl;  
    cout<<"Deliver Time:"<<deliverTime<<endl;  
}
```

```
Deliver::~~Deliver()  
{  
    cout<<endl;  
    cout<<"Deliver Cancelled"<<endl;  
}
```

customer.h

```
#include "order.h"
#include "feedback.h"
#include "bill.h"
class Customer
{
private:
    int cust_ID;
    char name[20];
    char address[50];
    char email[30];
    char contactNo[10];
    Feedback *feed[size];
    Bill *bill[size];
    Order *order;

public:
    Customer();
    Customer(int Cid, char cName[], char cAddress[], char cEmail[], char cContactNo[], Feedback *feedback, Bill *pbill,
    Order *porder);
    void setCusDetails(int id, char Name[], char Address[],char Email[], char ContactNo[]);
    void updateCusDetails();
    void diaplayCusDetails();
    ~Customer();
};
```

customer.cpp

```
#include<iostream>
#include<cstring>
#include "customer.h"
#include "feedback.h"
#include "bill.h"
using namespace std;

Customer::Customer()
{
    cust_ID=0;
    strcpy(name,"");
    strcpy(address,"");
    strcpy(email,"");
    strcpy(contactNo,"");
```

```
}
```

```
Customer::Customer(int Cid, char cName[], char cAddress[], char cEmail[], char cContactNo[], Feedback *feedback,  
Bill *pbill, Order *porder)
```

```
{
```

```
    cust_ID=Cid;  
    strcpy(name,cName);  
    strcpy(address,cAddress);  
    strcpy(email,cEmail);  
    strcpy(contactNo,cContactNo);  
    feed=feedback;  
    bill=pbill;  
    order=porder;
```

```
}
```

```
void Customer:: setCusDetails(int id, char Name[], char Address[],char Email[], char ContactNo[])
```

```
{
```

```
    cust_ID=id;  
    strcpy(name,Name);  
    strcpy(address,Address);  
    strcpy(email,Email);  
    strcpy(contactNo,ContactNo);
```

```
}
```

```
void Customer:: updateCusDetails()
```

```
{
```

```
    cout << "Input customer ID: ";  
    cin >> cust_ID;  
    cout << "Input customer name: ";  
    cin >> name;  
    cout << "Input customer address: ";  
    cin >> address;  
    cout << "Input E-mail address: ";  
    cin >> email;  
    cout << "Input customer contact number: ";  
    cin >> contactNo;
```

```
}
```

```
void Customer:: diaplayCusDetails()
```

```
{
```

```
    cout << "Customer ID = " << cust_ID << endl;  
    cout << "Customer name = " << name << endl;  
    cout << "Customer address = " << address << endl;
```

```
cout << "Customer e-mail = "<< email << endl;
cout << "Customer contact number = "<< contactNo << endl;
}
```

```
Customer::~Customer()
{
    cout << "Deleted" << endl;
}
```

feedback.h

```
#include "customer.h"
class Feedback
{
    private:
        int feedbackID;
        char email[20];
        char msg[50];
        Customer *cus;

    public:
        Feedback();
        Feedback(int feedID, char feedEmail[], char feedMsg[] Customer *pcus);
        void setFeedbackDetails(int FID, char FEmail[], char Fmsg[]);
        void displayFeedbackDetail();
        ~Feedback();
};
```

feedback.cpp

```
#include "feedback.h"
#include <iostream>
#include <cstring>
using namespace std;

Feedback::Feedback()
{
    int feedbackID=0;
    strcpy(email,"");
    strcpy(msg,"");
}
```



```

}
Feedback::Feedback(int feedID, char feedEmail[], char feedMsg[],Customer *pcus)
{
    int feedbackID=feedID;
    strcpy(email,feedEmail);
    strcpy(msg,feedMsg);
    cus =pcus;
}
void Feedback:: setFeedbackDetails(int FID, char FEmail[], char Fmsg[])
{
    int feedbackID=FID;
    strcpy(email,FEmail);
    strcpy(msg,Fmsg);
}
void Feedback:: displayFeedbackDetail()
{
    cout << "Feedback Id = " << feedbackID << endl;
    cout << "Email address = " << email << endl;
    cout << "Feedback = " << msg << endl;
}
Feedback::~Feedback()
{
    cout<< "Deleted"<< endl;
}

```

bill.h

```
#include<cstring>
#include"payment.h"
#include"customer.h"
class Bill{

private:
Customer* customer;
Payment* payment;
int BillNo;
double BillAmount;
char BillType[20];

public:
Bill();
Bill(int b_No,double b_Amount,char b_type[],Payment* pay, Customer* cus);
void setBillDetails(int B_No, double B_Amount, char B_Type[],Payment* paymnt, Customer* cusmer);
void displayBillDetails();
~Bill();
};
```

bill.cpp

```
#include <iostream>
#include <cstring>
#include"bill.h"
using namespace std;

Bill::Bill()
{
BillNo = 0;
BillAmount = 0;
strcpy(BillType, "");
}
Bill::Bill(int b_No,double b_Amount,char b_type[] ,Payment* pay, Customer* cus)
{
BillNo=b_No;
BillAmount=b_Amount;
strcpy(BillType,b_type);
payment=pay;
customer=cus;
}
void Bill::setBillDetails(int B_No, double B_Amount, char B_Type[], Payment* paymnt, Customer* cusmer)
```

```

{
    BillNo = B_No;
    BillAmount = B_Amount;
    strcpy(BillType,B_Type);
    payment=paymnt;
    customer=cusmer;
}
void Bill::displayBillDetails()
{
    cout<<"billnumber:"<<BillNo<<endl;
    cout<<"billamont:"<<BillAmount<<endl;
    cout<<"billtype:"<<BillType<<endl;
}
Bill::~~Bill(){
    cout<<"delete billdetails:"<<endl;
}

```

payment.h

```

#include <iostream>
#include "bill.h"
#include "order.h"
class payment{
protected:
    int paymentID;
    double totalPayment;
    Bill*bill;
    Order*ord;

public:
    payment();
    payment(int pPayemtID,double tTotalpayment,Bill*cbill, Order*cord );
    void setpaymentdetails(int pPayemtID,double tTotalpayment);
    void displaypaymentDetails();
    void confirmpaymentDetails();
    ~payment();
};

```

payment.cpp

```
#include <iostream>
#include <iomanip>
#include "payment.h"
#include "bill.h"
#include "order.h"
using namespace std;

payment::payment()
{
    int paymentID=0;
    double totalPayment=0.00;
}

payment::payment(int pPayemtID,double tTotalpayment,Bill*cbill, Order*cord );
{
    paymentID= pPayemtID;
    totalPayment=tTotalpayment;
    bill=cbill;
    ord=cord;
}

void payment::setpaymentdetails(int pPayemtID,double tTotalpayment)
{
    paymentID= pPayemtID;
    totalPayment=tTotalpayment;
}

void payment::displaypaymentDetails()
{
    cout<<"Payment ID:"<<paymentID<<endl;
    cout<<"Total Payment:"<<setiosflags(ios::fixed)<<setprecision(2)<<totalPayment<<endl<<endl;
}

void payment::confirmpaymentDetails()
{
}

payment::~~payment()
{
    cout<<"Payment is deleted"<<endl;
}
```

visacard.h

```
#include "payment.h"
class visacard :public payment {

private:
    int visacardNo;
    char vexp_date[10];
    int v_cvv;

public:
    visacard();
    visacard(int pPayemtID,double tTotalpayment,int v_vCardNo,char vExpDate[],int vCw,Order*v_order,Bill*v_bill);
    void setcardDetails(int cPayemtID,double cTotalpayment,int vc_vCardNo,char vcExpDate[],int
vc_Cw,Order*vc_order,Bill*vc_bill);
    void displaycardDetails();
    ~visacard();

};
```

visacard.cpp

```
#include <iostream>
#include<cstring>
#include"visacard.h"
using namespace std;

visacard::visacard()
{
    int visacardNo = 0;
    char vexp_date[10];
    int v_cvv;
}

visacard::visacard(int pPayemtID,double tTotalpayment,int v_vCardNo,char vExpDate[],int
vCw,Order*v_order,Bill*v_bill)
{
    visacardNo=v_vCardNo;
    strcpy(vexp_date,vExpDate);
    v_cvv=vCw;
    paymentID=pPayemtID;
    totalPayment=tTotalpayment;
    bill=v_bill;
    ord=v_order;
```

```
}
```

```
void visacard::setcardDetails(int cPayemtID,double cTotalpayment,int vc_vCardNo,char vcExpDate[],int  
vc_Cvw,Order*vc_order,Bill*vc_bill)
```

```
{
```

```
    visacardNo=vc_vCardNo;
```

```
    strcpy(vexp_date,vcExpDate);
```

```
    v_cvw=vc_Cvw;
```

```
    paymentID=cPayemtID;
```

```
    totalPayment=cTotalpayment;
```

```
    bill=vc_bill;
```

```
    ord=vc_order;
```

```
}
```

```
void visacard::displaycardDetails()
```

```
{
```

```
    cout<<"Visa Card No:"<<visacardNo<<endl;
```

```
    cout<<"Expire Date:"<<vexp_date<<endl;
```

```
    cout<<"CVV :"<<v_cvw<<endl<<endl;
```

```
}
```

```
visacard::~~visacard()
```

```
{
```

```
    cout<<"Visacard is deleted"<<endl;
```

```
}
```

creditcard.h

```
#include "payment.h"

class creditcard:public payment{

private:
    int creditcardNo;
    char cexp_date[10];
    int c_cvv;

public:
    creditcard();
    creditcard(int pPayemtID,double tTotalpayment,int cCardNo,char cExpDate[],int
cCvv,Order*v_order,Bill*v_bill);
    void setcardDetails(int cPayemtID,double cTotalpayment,int cCardNo,char cExpDate[],int
cCvv,Order*vc_order,Bill*vc_bill);
    void displaycardDetails();
    ~creditcard();
};
```

creditcard.cpp

```
#include <iostream>
#include<cstring>
#include"creditcard.h"
using namespace std;

creditcard::creditcard()
{
    int creditcardNo = 0;
    char cexp_date[10];
    int c_cvv;;
}

creditcard::creditcard(int pPayemtID,double tTotalpayment,int cCardNo,char cExpDate[],int
cCvv,Order*v_order,Bill*v_bill)
{
    creditcardNo=cCardNo;
    strcpy(cexp_date,cExpDate);
    c_cvv=cCvv;
    paymentID=pPayemtID;
    totalPayment=tTotalpayment;
```

```

    bill=v_bill;
    ord=v_order;

}

void creditcard::setcardDetails(int cPayemtID,double cTotalpayment,int cCardNo,char cExpDate[],int
cCvv,Order*vc_order,Bill*vc_bill);
{
    creditcardNo=cCardNo;
    strcpy(cexp_date,cExpDate);
    c_cvv=cCvv;
    paymentID=cPayemtID;
    totalPayment=cTotalpayment;
    bill=vc_bill;
    ord=vc_order;
}

void creditcard::displaycardDetails()
{
    cout<<"Credit card No:"<<creditcardNo<<endl;
    cout<<"Expire Date:"<<cexp_date<<endl;
    cout<<"CVV :"<< c_cvv<<endl<<endl;
}

creditcard::~~creditcard()
{
    cout<<"Creditcardd is deleted"<<endl;
}

```


Services.h

```
#include "customer.h"

class Services {
protected:
    char serviceName[20];

public:
    Services ();
    Services(char sname[]);
    void setServiceDetails(char S_Name[]);
    void addServices(Customer* c);
    void displayServiceDetails();
    ~Services();
};
```

Services.cpp

```
#include <iostream>
#include "Services.h"
#include <cstring>
using namespace std;

Services::Services()
{
    strcpy(serviceName, "");
}

Services:: Services(char sname[])
{
    strcpy (serviceName, sname);
}

void Services:: setServiceDetails(char S_Name[])
{
    strcpy(serviceName, S_Name);
}

void Services:: addServices(Customer* c)
{
}

void Services:: displayServiceDetails()
{
}
```

```
    cout << "Service Name : " << serviceName;
}
```

```
Services::~Services()
{
    cout << "Service deleted"<<endl;
}
```

MailUs.h

```
#include "Services.h"
class MailUs : public Services {

private:
    int mailID;
    char responseDate[12];
    char receiveDate[12];

public:
    MailUs();
    MailUs(char MS_Name[], int m_ID, char res_Date[], char rec_Date[]);
    void setMailUsDetails(char service[], int mail_ID, char response_Date, char receive_Date);
    void displayDetails();
    ~MailUs();
};
```

MailUs.cpp

```
#include<iostream>
#include "MailUs.h"
#include <cstring>
using namespace std;
```

```
MailUs::MailUs()
{
    int mailID = 0;
    char responseDate[12];
    char receiveDate[12];
}
```

```
MailUs(char MS_Name[], int m_ID, char res_Date[], char rec_Date[])
```

```

{
    strcpy (serviceName,MS_Name)
    mailID = m_ID;
    strcpy (responseDate, res_Date);
    strcpy (receiveDate, rec_Date);
}

void MailUs:: setMailUsDetails(char service[], int mail_ID, char response_Date, char receive_Date)
{
    strcpy (serviceName,service);
    mailID = mail_ID;
    strcpy(responseDate, response_Date);
    strcpy(receiveDate, receive_Date);

}

void ContactAgent::displayContactDetails()
{
    cout <<"Mail ID : "<< mailID <<" "
        <<"Received Date : "<< receiveDate <<" "
        <<"Responded Date : "<< responseDate ;
}

MailUs::~MailUs()
{
    cout << "Mail deleted"<<endl;
}

```

ContactAgent.h

```
#include "Services.h"

class ContactAgent: public Services {

private:
    int agentID;
    char receiveDate;
    char responseDate;

public:
    ContactAgent();
    ContactAgent(char MS_Name[], int agent_ID, char receive_Date, char response_Date);
    void setContactAgentDetails(char MSame[],int ag_ID, char rec_Date, char resp_Date);
    void displayContactDetails();
    ~ContactAgent();
};
```

ContactAgent.cpp

```
#include "ContactAgent.h"
#include <iostream>
#include <cstring>
using namespace std;

ContactAgent::ContactAgent()
{

}

ContactAgent::ContactAgent(char MS_Name[], int agent_ID, char receive_Date, char response_Date)
{
    strcpy (sname, MS_Name)
    agentID = agent_ID;
    strcpy(responseDate, response_Date);
    strcpy(receiveDate, receive_Date);
}

void ContactAgent:: setContactAgentDetails(char MSame[],int ag_ID, char rec_Date, char resp_Date)
{
    strcpy (sname, MSName);
    ag_ID = agent_ID;
    strcpy(responseDate, resp_Date);
    strcpy(receiveDate, rec_Date);
}

void ContactAgent::displayContactDetails()
```

```
{  
  cout <<"Agent ID : "<<agentID <<" "  
    <<"Received Date : "<<receiveDate <<" "  
    <<"Responded Date : "<<responseDate ;  
}
```

```
ContactAgent::~~ContactAgent()  
{  
  
}
```

Main.cpp

```
#include <iostream>
#include "item.h"
#include "order.h"
#include "cart.h"
#include "deliver.h"
#include "customer.h"
#include "feedback.h"
#include "bill.h"
#include "payment.h"
#include "visacard.h"
#include "creditcard.h"
#include "Services.h"
#include "MailUs.h"
#include "ContactAgent.h"
using namespace std;

int main()
{
    //objects for item class
    Item *item1 = new Item(11111,"Biscuit","Munchee",60.00);
    Item *item2 = new Item(22222,"Chocolate","Kandos",350.00);
    Item *item3 = new Item(33333,"Ice-Cream","Magic",1500.0);
    Item *item4 = new Item(44444,"Yougurt","Highland",50.00);

    //objects for order class
    Order *ord1 = new Order(01212,2500.00,"12.05.2022",2);
    Order *ord2 = new Order(01313,750.00,"08.05.2022",7);

    ord1->addItem(item1,item2);
    ord2->addItem(item3,item4);

    //objects for cart class
    Cart *crt1 = new Cart(0001);
    Cart *crt2 = new Cart(0002);
    Cart *crt3 = new Cart(0003);
    Cart *crt4 = new Cart(0004);

    crt1->addCart(5,item1);
    crt2->addCart(1,item2);
    crt3->addCart(9,item3);
    crt4->addCart(8,item4);

    //objects for Deliver class

    Deliver *deliver1 = new Deliver(11111,"24.01.2022","Gampaha","17.30");
```

```
Deliver *deliver2 = new Deliver(11112,"05.12.2021","Matara","12.30");
Deliver *deliver3 = new Deliver(11113,"09.03.2022","Kottawa","14.30");
Deliver *deliver4 = new Deliver(11114,"04.05.2022","Ragama","13.30");
```

//objects for customer class

```
Customer *customer1 = new Customer(00001,"Uwani","22/A,Homagama","uwani@gmail.com","0711111189");
Customer *customer2 = new
Customer(00002,"Tharusha","1/5A,Gampaha","tharusha@gmail.com","07111111320");
Customer *customer3 = new
Customer(00003,"Sewmini","79/6A,Kegalle","sewmini@gmail.com","07111111450");
Customer *customer4 = new Customer(00001,"Shamal","74A,Nittabuwa","shamal@gmail.com","07111111150");
```

//objects for Feedback class

```
Feedback *feedback1 = new Feedback(00111,"kamal123@gmail.com","Good service");
Feedback *feedback2 = new Deliver(00222,"sunil123@gmail.com","excellent service");
Feedback *feedback3 = new Deliver(00333,"Amara123@gmail.com","Good service");
Feedback *feedback4 = new Deliver(00444,"meena123@gmail.com","excellent service");
```

//objects for Bill class

```
Bill *bill1 = new Bill(00011,4500.00,"WaterBill");
Bill *bill2 = new Bill(00012,1500.00,"Electricity Bill");
Bill *bill3 = new Bill(00013,750.00,"Insurance Bill");
Bill *bill4 = new Bill(00014,5000.00,"Mobile Bill");
```

//objects for visacard class

```
Visacard *visacard1 = new Visacard(01111,7500.00,1234567891011213,"04.05.2025",485);
Visacard *visacard2 = new Visacard(01112,3600.00,1234567891011895,"01.08.2024",120);
Visacard *visacard3 = new Visacard(01113,950.00,1234567891011900,"31.12.2022",407);
Visacard *visacard4 = new Visacard(01114,6100.00,1234567891011200,"14.05.2026",501);
```

//objects for creditcard class

```
Ceditcard *creditcard1 = new creditcard(00111,6900.00,1234567891896213,"28.05.2024",789);
Ceditcard *creditcard2 = new creditcard(00112,7200.00,1234567891256895,"16.01.2024",465);
Ceditcard *creditcard3 = new creditcard(00113,8000.00,1234567891000900,"31.01.2023",123);
Ceditcard *creditcard4 = new creditcard(00114,12000.00,123456789178200,"30.05.2024",741);
```

//objects for Mailus class

```
MailUs *mailus1 = new MailUs(00005,"28.05.2022","07.07.2023");
MailUs *mailus2 = new MailUs(00006,"28.07.2023","07.08.2024");
MailUs *mailus3 = new MailUs(00007,"28.10.2024","07.12.2025");
```

```
MailUs *mailus4 = new MailUs(00008,"28.12.2025","07.01.2026");
```

```
//objects for ContactAgent class
```

```
ContactAgent *contctagent1 =new contactagent(00012,"01.01.2022","10.07.2023");
```

```
ContactAgent *contctagent2 =new contactagent(00013,"02.02.2022","11.07.2023");
```

```
ContactAgent *contctagent3 =new contactagent(00014,"03.03.2022","12.07.2023");
```

```
ContactAgent *contctagent4 =new contactagent(00015,"04.04.2022","13.07.2023");
```

```
item1-> displayItemDetails();
```

```
item2-> displayItemDetails();
```

```
item3-> displayItemDetails();
```

```
item4-> displayItemDetails();
```

```
cout << "" << endl;
```

```
ord1-> displayDetails();
```

```
ord2-> displayDetails();
```

```
cout << "" << endl;
```

```
crt1-> displayCartDetails();
```

```
crt2-> displayCartDetails();
```

```
crt3-> displayCartDetails();
```

```
crt4-> displayCartDetails();
```

```
cout << "" << endl;
```

```
deliver1-> displayDeliverDetails();
```

```
deliver2-> displayDeliverDetails();
```

```
deliver3-> displayDeliverDetails();
```

```
deliver4-> displayDeliverDetails();
```

```
cout << "" << endl;
```

```
customer1->displayCusDetails();
```

```
customer2->displayCusDetails();
```

```
customer3->displayCusDetails();
```

```
customer4->displayCusDetails();
```

```
cout << "" << endl;
```

```
feedback1->displayFeedbackDetail();
```

```
feedback2->displayFeedbackDetail();
```

```
feedback3->displayFeedbackDetail();
```

```
feedback4->displayFeedbackDetail();
```

```
cout << "" << endl;
```

```
bill1->displayBillDetails();
```

```
bill2->displayBillDetails();
```

```
bill3->displayBillDetails();
```

```
bill4->displayBillDetails();
```



```
cout << "" << endl;
```

```
visacard1->displaycardDetails();  
visacard2->displaycardDetails();  
visacard3->displaycardDetails();  
visacard4->displaycardDetails();  
cout << "" << endl;
```

```
creditcard1->displaycardDetails();  
creditcard2->displaycardDetails();  
creditcard3->displaycardDetails();  
creditcard4->displaycardDetails();  
cout << "" << endl;
```

```
mailus1->displayDetails();  
mailus2->displayDetails();  
mailus3->displayDetails();  
mailus4->displayDetails();  
cout << "" << endl;
```

```
contctagent1->displayContactDetails();  
contctagent2->displayContactDetails();  
contctagent3->displayContactDetails();  
contctagent4->displayContactDetails();  
cout << "" << endl;
```

```
delete item1;  
delete item2;  
delete item3;  
delete item4;  
delete ord1;  
delete ord2;  
delete crt1;  
delete crt2;  
delete crt3;  
delete crt4;  
delete deliver1;  
delete deliver2;  
delete deliver3;  
delete deliver4;  
delete customer1;  
delete customer2;  
delete customer3;  
delete customer4;  
delete feedback1;  
delete feedback2;  
delete feedback3;
```

```
delete feedback4;
delete bill1;
delete bill2;
delete bill3;
delete bill4;
delete visacard1;
delete visacard2;
delete visacard3;
delete visacard4;
delete creditcard1;
delete creditcard2;
delete creditcard3;
delete creditcard4;
delete mailus1;
delete mailus2;
delete mailus3;
delete mailus4;
delete contactagent1;
delete contactagent2;
delete contactagent3;
delete contactagent4;

return 0;
}
```

Individual Contribution

Registration No	Name	Individual Contribution
IT21310546	Hewamadduma. N. A	Payment VisaCard CreditCard
IT21309106	Imasha. P. U	Customer Feedback Item
IT21308598	Mahawaththa.N.T.M.A.S.M	Order Deliver Cart
IT21307980	Rodrigo.W.S. A	MailUs Contact Agent
IT21312694	Sandeeptha.P.K. T	Bill Services