



Topic : Online Shopping Mall

Group no : MLB_02.02_10

Campus : Malabe

Submission Date : 14/10/2021

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
IT21008696	Leo M.P.C	0769839398
IT21105302	Fernando U.S.L	0779389014
IT21101274	Dolawatta T.Y.R	0770438773
IT21126956	Wijethunge W.D.S.H	0787706921
IT21111488	Dilara V.G.N	0710192886

Exercise 1

eZBuy is a Online Shopping Mall which can be accessed through a web browser. When user visits a store they can browse items as a visitor and they can become a registered customer by filling a simple online form to expand their privileges such as purchasing items, adding items to the personal cart, rating the items.

A user can also register as a seller in order to sell items through eZBuy by requesting permission from the system admin through a seller form. System admin can approve the seller requests. A seller can publish their products and sell them under the supervision of system manager.

Technical support person can view the customer and seller complaints, manage them and also manage the FAQs.

Web developer maintain the website, fix any bugs and manage protocols. System manager controls the database and promotion requests.

System Administrator can examine the generated reports, recruit employees and define access levels in the website. System Database, stores all data such as customer details, store details, item details and seller details. Online payment system validate the payment details which are entered to the system by customers and sellers .

Requirements

1)

1. A visitor can create an account as a customer or seller. To register both types of users should provide necessary information for the system such as name, date of birth, NIC, address, email address and the most importantly type of role.
2. Registered users can login to the system using their credentials.
3. Visitors, sellers and customers can browse and discover items in the system.
4. Registered customers and sellers can update user details and view the current details.
5. A registered customer can add items to their personal Cart and also manage their personal cart by removing or updating information.
6. Most importantly a registered customer can place order using their credit/debit card through this system as a payment method.
7. A registered customer can rate and review the items that they bought through the system.
8. A seller can publish items via the system.
9. Seller can read and reply the reviews given by the registered customers.
10. A seller can create promotion codes which can be applicable to buy their items. And the promotion codes are sent to the regular customers via emails.
11. A seller can request from the system manager to advertise their items in the system.

12. Sellers can view the seller dashboard to examine their progress in the system.
13. Seller and registered customer, both can provide feedback about the system.
14. Technical support employee can view the feedbacks given by the registered users and reply to them accordingly.
15. System manager can create desired reports based through the system.
16. System admin can view the reports.

2)

<u>Key</u>	
Verb –	
Noun –	

1. A visitor can create an account as a customer or seller. To register both types of users should provide necessary information for the system such as name, date of birth, NIC, address, email address and the most importantly type of role.
2. Registered users can login to the system using their credentials.
3. Visitors, sellers and customers can browse and discover items in the system.
4. Registered customers and sellers can update user details and view the current details.
5. A registered customer can add items to their personal Cart and also manage their personal cart by removing or updating information.
6. Most importantly a registered customer can place orders using their credit/debit card through this system as a payment method.
7. A registered customer can rate and review the items that they bought through the system.
8. A seller can publish items via the system.
9. Seller can read and reply the reviews given by the registered customers.
10. A seller can create promotion codes which can be applicable to buy their items. And the promotion codes are sent to the regular customers via emails.
11. A seller can request from the system manager to advertise their items in the system.
12. Sellers can view the seller dashboard to examine their progress in the system.
13. Seller and registered customer, both can provide feedback about the system.

14. Technical support employee can view the feedbacks given by the registered users and reply to them accordingly.
15. System manager can create desired reports based through the system.
16. System admin can view the reports.

3)

Nouns

Visitor
 Account
Customer
Seller
User
 Information
 System
Payment
 Name
 Date Of Birth
 NIC
 Address
 Email Address
 Role
 Registered Users
 Credentials

Items
 Registered Customer
Orders
 User details
 Current Details
Personal Cart
 Credit/Debit Card
Reviews
Promotion Codes
 Regular Customers
 Emails
System Manager
 Seller Dashboard
Feedback
Technical Support
Employee
Report

Rejecting Unwanted Nouns.

1.Redundant

Registered Users
 Registered Customer
 Visitor
 Information
 Credentials
 User details
 Current Details
 Regular Customers

2. Event or operation

Emails
 Seller Dashboard

3.Outside scope of system

System
 System manager
 Account

4.Attributes

Name
 Date Of Birth
 NIC
 Address
 Email Address
 Role
 Credit/Debit Card

Classes

Classes

- 1.Customer
- 2.Seller
- 3.User
- 4.Payment
- 5.Items
- 6.Orders
7. Personal Cart
- 8.Reviews
- 9.Promotion Codes
- 10.Feedback
- 11.Technical Support
- 12.Report
- 13.Employee

Verbs

1. User

Create
Register
Provide
Login

2.Customer

Update
Buy
Remove
Add
Place Order
Rate

3.Seller

Update
Buy
Remove
Add
Publish
Request
View
Reply

4.Technical Support

Reply
View

Exercise 2

CRC Cards

Class name: User	
Responsibilities:	Collaborations:
Store User Details	
Display User Details	
Update User Details	

Class name: Customer	
Responsibilities:	Collaborations:
Store Customer ID	
Store Login Credentials	
Validate Customer Credentials	

Class name: Seller	
Responsibilities:	Collaborations:
Store Seller ID	
Store Login Credentials	
Validate Seller Credentials	

Class name: Payment	
Responsibilities:	Collaborations:
Store Card Details	Order
Store Payment Type	Order
Display Payment Details	

Class name: Item	
Responsibilities:	Collaborations:
Store Item Details	
Display Item Details	
Update Item Details	

Class name: Order	
Responsibilities:	Collaborations:
Place Order	Items, Promotion Codes
Calculate Subtotal	
Get subtotal	
Display Order Details	

Class name: Cart	
Responsibilities:	Collaborations:
Store Wish-list Details	Item
Display Wish-list Details	Item
Update Wish-list Details	Item

Class name: Review	
Responsibilities:	Collaborations:
Add Reviews about Items	Customer, Item
View Reviews	Seller, Item

Class name: PromotionCode	
Responsibilities:	Collaborations:
Create Promotion Codes	Item

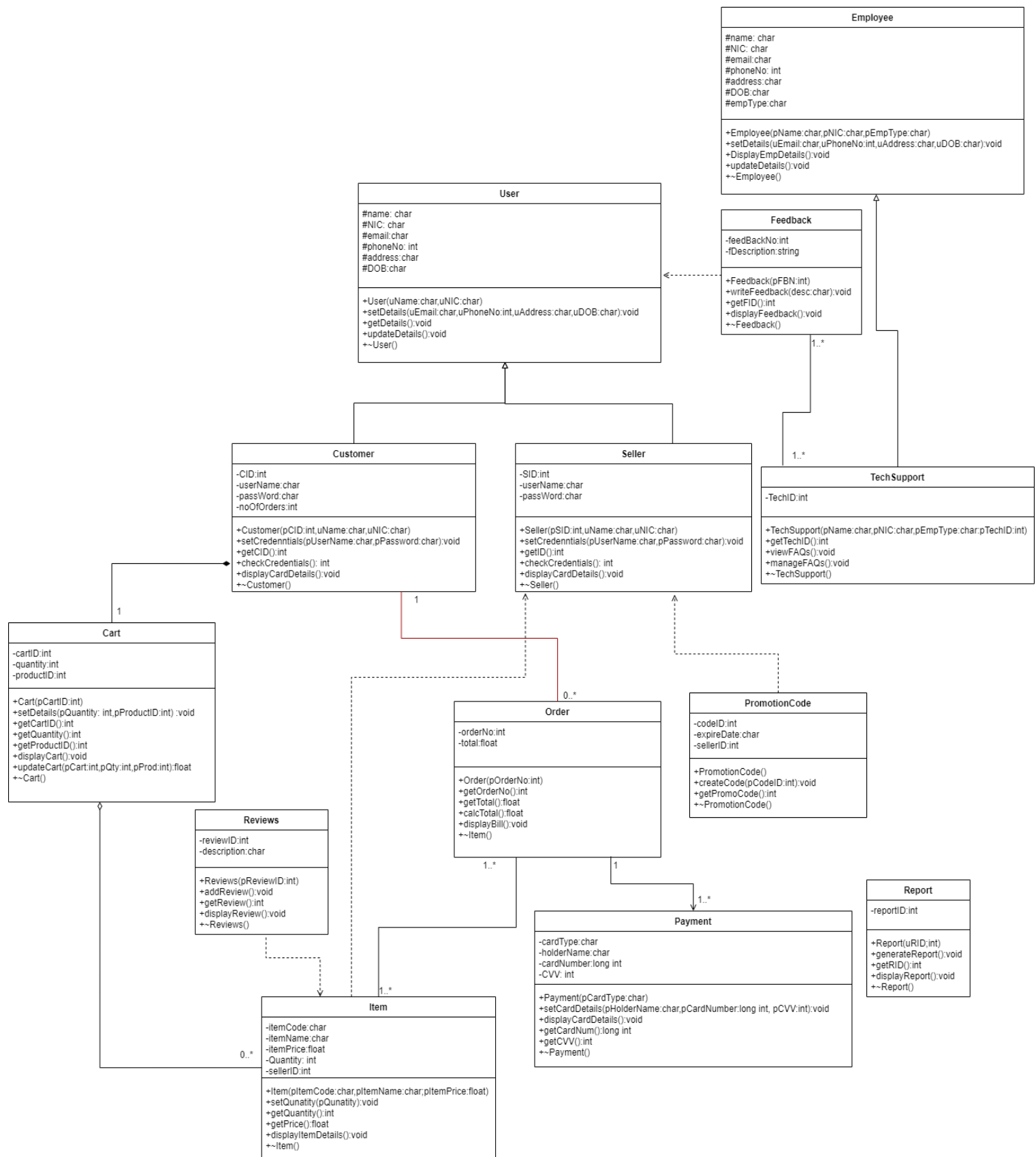
Class name: Feedback	
Responsibilities:	Collaborations:
Add Feedback	User
View Feedback	Technical Support

Class name: Employee	
Responsibilities:	Collaborations:
Store Employee Details	
Display Employee Details	
Update Employee Details	

Class name: TechSupport	
Responsibilities:	Collaborations:
Store Tech ID	
Manage FAQs	Feedback

Class name: Report	
Responsibilities:	Collaborations:
Generate Report	
Display Report	

Exercise 03



Contributions

	Student ID	Student Name	Individual Contribution
1	IT21008696	Leo M.P.C	CRC: 1. User 2. Customer 3. Seller Code: 1. Main.cpp 2. User.h 3. User.cpp 4. Seller.h 5. Seller.cpp 6. Customer.h 7. Customer.cpp
2	IT21105302	Fernando U.S.L	CRC: 1. Item 2. Cart 3. Order Code: 1. Main.cpp 2. Item.h 3. Item.cpp 4. Cart.h 5. Cart.cpp 6. Order.h 7. Order.cpp
3	IT21101274	Dolawatta T.Y.R	CRC: 1. Employee 2. TechSupport 3. Feedback Code: 1. Main.cpp 2. Employee.h 3. Employee.cpp 4. TechSupport.h 5. TechSupport.cpp 6. Feedback.h 7. Feedback.cpp

4	IT21126956	Wijethunge W.D.S.H	CRC: <ol style="list-style-type: none"> 1. Review 2. Report Code: <ol style="list-style-type: none"> 1. Main.cpp 2. Review.h 3. Review.cpp 4. Report.h 5. Report.cpp
5	IT21111488	Dilara V.G.N	CRC: <ol style="list-style-type: none"> 1. Payment 2. PromotionCode Code: <ol style="list-style-type: none"> 1. Main.cpp 2. Payment.h 3. Payment.cpp 4. PromotionCode.h 5. PromotionCode.cpp

Exercise 4

Sample Code

Main.cpp

```
#include <iostream>
//#include "User.h"
//#include "Customer.h"
//#include "Seller.h"
//#include "Payment.h"      These are not needed as they are included in below header files
//#include "Employee.h"
//#include "Item.h"
//#include "Order.h"
//#include "TechSupport.h"
#include "Feedback.h"
#include "PromotionCode.h"
#include "Cart.h"
#include "Review.h"
#include "Report.h"

using namespace std;
int main() {

    //IT21008696
    cout<<"-----Customer-----\n"<<endl;
    Customer*C1=new Customer("Bruce Wayne","9223456789V",888);
    C1->setDetails("brucewayne@wayne.lk",778899551,"Rosmead Place,Colombo 7","Feb 31");

    C1->DisplayDetails();
    cout<<"Customer ID: "<<C1->getCID();

    cout<<"\n-----Seller-----\n"<<endl;
    Seller*S1=new Seller("Ryan Reynolds","8523456888V",511);
    S1->setDetails("rreynolds@gmail.com",765511888,"Senevirathne Road,Negombo","March 31");
    S1->DisplayDetails();
    cout<<"Seller ID: "<<S1->getID();
```

```

//IT21105302
cout<<"\n-----Item-----\n"<<endl;
Item *i1 = new Item("IT01", "Pen Holder", 225.25);
i1->setQuantity(15);
i1->addItem(S1);
Item *i2 = new Item("IT02", "Eraser", 5.25);
i2->setQuantity(25);
i2->addItem(S1);
Item *i3 = new Item("IT03", "Pen", 20);
i3->setQuantity(35);
i3->addItem(S1);

i1->displayItemDetails();
i2->displayItemDetails();
i3->displayItemDetails();

cout<<"\n-----Cart-----\n"<<endl;
Cart *cart[1];
cart[0] = new Cart(1);
cart[0]->addItem(i1, i3);

cart[0]->displayCart();

cout<<"\n-----Order-----\n"<<endl;
Order *O1 = new Order(001, C1);
Order *O2 = new Order(002, C1);

O1->displayBill();
O2->displayBill();

//IT21111488
cout <<"\n-----Payment-----\n";
Payment *p1 = new Payment("Visa");
p1-> setCardDetails("776388421", 371564844421685, 164);
p1->displayCardDetails();
cout <<"Card number: "<< p1->getCardNum() <<endl;
cout <<"CVV: "<< p1->getCVV() <<endl;

cout <<"\n-----PromotionCode-----\n"<<endl;
PromotionCode *pc1 = new PromotionCode();
pc1->createCode(46575, S1);
cout <<"Promotion Code: " <<pc1->getPromoCode();

```

```

//IT21101274
cout<<"\n-----Technical Support-----\n"<<endl;
TechSupport*TS1 = new TechSupport("Floki LeBateau","9527451789V","Technical Support",151);
TS1->setDetails("flokilb@ezbuy.lk",715864255,"Temple Road,Wattala","June 10");
TS1->DisplayEmpDetails();
cout<<"Employee ID: "<<TS1->getTechID()<<endl;

//IT21126956
cout << "\n-----Review-----\n" <<endl;
Review *r1 = new Review(46571);
r1->addReview("This is a good item.", i1);
cout << "Review ID: " <<r1->getReviewID() <<endl;
r1-> displayReview();

cout << "\n-----Report-----\n" <<endl;
Report *RE1 = new Report(9463);
cout<< "Report ID: " <<RE1->getRID() <<endl;
RE1->generateReport();
RE1->displayReport();

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

delete C1;
delete S1;
delete i1;
delete i2;
delete i3;
delete cart[0];
delete O1;
delete O2;
delete p1;
delete TS1;
delete pc1;
delete RE1;

return 0;
}

```

User.h

```
//IT21008696
#pragma once
class User{
    protected :
        char name[20];
        char NIC[15];
        char email[30];
        int phoneNo;
        char address[50];
        char DOB[10];

    public :
        User(const char uName[],const char uNIC[]);
        void setDetails(const char uEmail[],int uPhoneNo,const char uAddress[],const char
uDOB[]);
        void DisplayDetails();
        void updateDetails();
        ~User();
};
```

User.cpp

```
//IT21008696
#include "User.h"
#include <iostream>
#include <cstring>

using namespace std;

User::User(const char uName[],const char uNIC[]){
    strcpy(name, uName);
    strcpy(NIC, uNIC);
}

void User::setDetails(const char uEmail[],int uPhoneNo,const char uAddress[],const char
uDOB[]){
    strcpy(email,uEmail);
    phoneNo = uPhoneNo;
    strcpy(address,uAddress);
    strcpy(DOB,uDOB);
}

void User::DisplayDetails(){
    cout << "Name : " << name<<endl;
    cout << "NIC : " << NIC <<endl;
    cout << "Email : " <<email << endl;
    cout << "Phone Number : " <<phoneNo <<endl;
    cout << "Address : " <<address <<endl;
    cout << "DOB : " <<DOB <<endl;
```



```

}
void User::updateDetails(){

}
User::~User(){
    cout << "Deleting user name " <<name <<endl;
}

```

Customer.h

```

//IT21008696
#pragma once

#include "User.h"
#include "Cart.h"
#include "Order.h"
#define SIZE5 5

class Cart;
class Order;

class Customer:public User{
private:
    Cart *cart[1];
    int CID;
    char userName[30];
    char passWord[15];
    Order*ORD[SIZE5];
    int noOfOrders;

public:
    Customer();
    Customer(const char uName[],const char uNIC[],int pCID);
    void setCredentials(char pUserName[], char pPassword[]);
    int getCID();
    int checkCredentials();
    void addOrder(Order *O);
    void displayCartDetails();
    ~Customer();
};

```

Customer.cpp

```
//IT21008696
#include<iostream>
#include<cstring>
#include"Customer.h"

using namespace std;

Customer::Customer(const char uName[],const char uNIC[],int pCID):User(uName,uNIC){
    CID=pCID;
    Cart *cart[1];
    cart[0] = new Cart(1);
}
void Customer::setCredentials(char pUserName[], char pPassword[]){

}
int Customer::getCID(){
    return CID;
}
int Customer::checkCredentials(){
    return 1;
}
void Customer::addOrder(Order *O)
{
    if (noOfOrders < SIZE5)
        ORD[noOfOrders] = O;
    noOfOrders++;
}
void Customer::displayCartDetails(){

}

Customer::~~Customer(){
    cout << "Deleting Customer " << CID << endl;
    delete cart[0];
    cout << "The end" << endl ;
}
```

Seller.h

```
//IT21008696
#pragma once
#include "User.h"

class Seller:public User{
private:
    int SID;
    char userName[30];
    char passWord[15];

public:
    Seller(const char uName[],const char uNIC[],int pSID);
    void setCredentials(const char pUserName[],const char pPassword[]);
    int getID();
    int checkCredentials();
    void displayCarDetails();
    ~Seller();
};
```

Seller.cpp

```
//IT21008696
#include<iostream>
#include<cstring>
#include"Seller.h"

using namespace std;

Seller::Seller(const char uName[],const char uNIC[],int pSID):User(uName,uNIC){
    SID=pSID;
}

void Seller::setCredentials(const char pUserName[],const char pPassword[]){

}

int Seller::getID(){
    return SID;
}

int Seller::checkCredentials(){
    return 1;
}

void Seller::displayCarDetails(){

}

Seller::~Seller(){
    cout << "Deleting SID" <<SID <<endl;
}
```

Item.h

```
//IT21105302
#pragma once
#include "Order.h"
#include "Seller.h"
#define SIZE3 5

class Order;
class Seller;

class Item{
private:
    int sellerID;
    char itemCode[6];
    char itemName[20];
    float itemPrice;
    int Quantity;
    Order*ORD[SIZE3];

public:
    Item(const char pItemCode[],const char pItemName[],float pItemPrice);
    void addItem(Seller*Sell);
    void setQuantity(int pQuantity);
    int getQuantity();
    float getPrice();
    void displayItemDetails();
    ~Item();
};
```

Item.cpp

```
//IT21105302
#include<iostream>
#include<cstring>
#include"Item.h"

using namespace std;

Item::Item(const char pItemCode[],const char pItemName[],float pItemPrice){
    strcpy(itemCode, pItemCode);
    strcpy(itemName, pItemName);
    itemPrice = pItemPrice;
}

void Item::addItem(Seller*Sell){
    sellerID= Sell->getID();
}

void Item::setQuantity(int pQuantity){
    Quantity=pQuantity;
}

int Item::getQuantity(){
    return Quantity;
}

float Item::getPrice(){
    return itemPrice;
}

void Item::displayItemDetails(){
    cout<< "Description of Item " << itemCode << endl<<endl;
    cout << "Item Name : "<< itemName << endl;
    cout << "Price : Rs."<< itemPrice << endl;
    cout << "Available quantity : "<< Quantity << endl;
    cout << "SellerID : "<< sellerID << endl<< endl<<endl;

}

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

Cart.h

```
//IT21105302
#pragma once
#include "Item.h"

class Item;

class Cart{
private:
    int cartID;
    int quantity;
    int productID;
    Item *item[2];

public:
    Cart(int pCartID);
    void setDetails(int pQuantity, int pProductID);
    void addItem(Item *item1, Item *item2);
    int getCartID();
    int getQuantity();
    int getProductID();
    void displayCart();
    float updateCart(int pCart, int pQty, int pProd);
    ~Cart();
};
```

Cart.cpp

```
//IT21105302
#include <iostream>
#include "Cart.h"
using namespace std;

Cart::Cart(int pCartID){
    cartID = pCartID;
    cout << "Cart constructor called" << endl;
}
void Cart::setDetails(int pQuantity, int pProductID){
    quantity = pQuantity;
    productID = pProductID;
}
void Cart::addItem(Item *item1, Item *item2){
    item[0]= item1;
    item[1]= item2;
}
}
```

```

int Cart::getCartID(){
    return cartID;
}

int Cart::getQuantity(){
    return quantity;
}

int Cart::getProductID(){
    return productID;
}

void Cart::displayCart(){
    cout << "Cart id :" << cartID<<endl;
    cout <<"    Items    "<<endl;
    for(int i=0;i<2;i++){
        item[i]->displayItemDetails();
    }
}

float Cart::updateCart(int pCart, int pQty, int pProd){
    //code
    return 1;
}

Cart::~~Cart(){
    cout << "Cart destructor called" << endl;
}

```

Order.h

```

//IT21105302
#pragma once
#include "Customer.h"
#include "Item.h"
#include "Payment.h"
#define SIZE2 10

class Customer;
class Item;
class Payment;

class Order{
public:
    int orderNo;
    float total;
    Customer*Cus;
    Item*ITM[SIZE2];
    Payment*PAID[SIZE2];
}

```

```

public:
    Order(int pOrderNo, Customer *pCus);
    int getOrderNo();
    float getTotal();
    float calcTotal();
    void displayBill();
    ~Order();
};

```

Order.cpp

```

//IT21105302
#include<iostream>
#include<cstring>
#include"Order.h"

using namespace std;

Order::Order(int pOrderNo, Customer *pCus){
    orderNo = pOrderNo;
    Cus = pCus;
    Cus -> addOrder(this);
}
int Order::getOrderNo(){
    return orderNo;
}
float Order::getTotal(){
    return 1;
}
float Order::calcTotal(){
    return 1;
}
void Order::displayBill(){
    cout << "Order ID : "<< orderNo << endl;
    cout << "Customer ID " << Cus->getCID()<< endl<< endl;
}
Order::~~Order(){
}

```


Employee.h

```
//IT21101274
#pragma once
class Employee{
protected:
    char name[30];
    char NIC[15];
    char email[30];
    int phoneNo;
    char address[50];
    char DOB[10];
    char empType[20];

public:
    Employee(const char pName[], const char pNIC[],const char pEmpType[]);
    void setDetails(const char uEmail[], int uPhoneNo,const char uAddress[],const char
uDOB[]);
    void DisplayEmpDetails();
    void updateDetails();
    ~Employee();
};
```

Employee.cpp

```
//IT21101274
#include<iostream>
#include<cstring>
#include"Employee.h"

using namespace std;

Employee::Employee(const char pName[], const char pNIC[],const char pEmpType[]){
    strcpy(name,pName);
    strcpy(NIC,pNIC);
    strcpy(empType,pEmpType);
}
void Employee::setDetails(const char uEmail[], int uPhoneNo,const char uAddress[], const
char uDOB[]){
    strcpy(email,uEmail);
    phoneNo=uPhoneNo;
    strcpy(address,uAddress);
    strcpy(DOB,uDOB);
}

}
```

```

void Employee::DisplayEmpDetails(){
    cout << "Employee Name : " << name<<endl;
    cout << "Employee NIC : " << NIC <<endl;
    cout << "Employee Type: "<<empType<<endl;
    cout << "Employee Email : " <<email << endl;
    cout << "Employee Phone Number : " <<phoneNo <<endl;
    cout << "Employee Address : " <<address <<endl;
    cout << "Employee DOB : " <<DOB <<endl;
}
void Employee::updateDetails(){

}
Employee::~Employee(){
    cout << "Employee name deleted" <<name <<endl;
}

```

TechSupport.h

```

//IT21101274
#pragma once
#include "Employee.h"
#include "Feedback.h"
#define SIZE1 2

class Employee;
class Feedback;

class TechSupport:public Employee{
private:
    int techID;
    Feedback*FB[SIZEl];
public:
    TechSupport(const char pName[], const char pNIC[],const char pEmpType[],int pTechID);
    int getTechID();
    void viewFAQs();
    void manageFAQs();
    ~TechSupport();
};

```

TechSupport.cpp

```
//IT21101274
#include<iostream>
#include<cstring>
#include"TechSupport.h"

using namespace std;

TechSupport::TechSupport(const char pName[], const char pNIC[],const char pEmpType[],int
pTechID):Employee(pName,pNIC,pEmpType){
    techID = pTechID;
}
int TechSupport::getTechID(){
    return techID;
}
void TechSupport::viewFAQs(){

}
void TechSupport::manageFAQs(){

}
TechSupport::~TechSupport(){
    cout << "Deleting techID" <<techID <<endl;
}
```

Feedback.h

```
//IT21101274
#pragma once
#include "TechSupport.h"
#include "User.h"
#define SIZE4 5

class TechSupport;

class Feedback{
private:
    int feedBackNo;
    char fDescription[200];
    TechSupport *tech[SIZE4];

public:
    Feedback(int pFBN);
    void writeFeedback(const char description[],User *user);
    int getFID();
    void displayFeedback();
    ~Feedback();
};
```

Feedback.cpp

```
//IT21101274
#include "Feedback.h"
#include <cstring>
#include <iostream>

using namespace std;

Feedback::Feedback(int pFBN){
    feedBackNo=pFBN;
}
void Feedback::writeFeedback(const char description[],User *user){
    user->DisplayDetails();
    strcpy(fDescription,description);
}
int Feedback::getFID(){
    return feedBackNo;
}
void Feedback::displayFeedback(){
}
Feedback::~Feedback(){
    cout << "Deleted feedback number " <<feedBackNo <<endl;
}
```

Payment.h

```
//IT21111488
#pragma once

class Payment{
private:
    char cardType[30];
    char holderName[30];
    long int cardNumber;
    int CVV;

public:
    Payment(const char pCardType[]);
    void setCardDetails(const char pHolderName[],long int pCardNumber, int pCVV);
    void displayCardDetails();
    long int getCardNum();
    int getCVV();
    ~Payment();
};
```

Payment.cpp

```
//IT21111488
#include "Payment.h"
#include <cstring>
#include <iostream>

using namespace std;

Payment::Payment(const char pCardType[]){
    strcpy(cardType, pCardType);
}

void Payment::setCardDetails(const char pHolderName[],long int pCardNumber, int pCVV){
    strcpy(holderName,pHolderName);
    cardNumber = pCardNumber;
    CVV = pCVV;
}

void Payment::displayCardDetails(){
    cout << "Card Type : " <<cardType <<endl;
    cout << "Holder Name : " << holderName << endl;
}

long int Payment::getCardNum(){
    return cardNumber;
}

int Payment::getCVV(){
    return CVV;
}

Payment::~Payment(){
    cout << "Deleting payment " <<endl;
}

}
```

PromotionCode.h

```
//IT21111488
#pragma once
#include "Seller.h"

class PromotionCode{
private:
    int sellerID;
    int codeID;
    char expireDate[10];
}
```

```

    public:
        PromotionCode();
        void createCode(int pCodeID, Seller* sell);
        int getPromoCode();
        ~PromotionCode();
};

```

PromotionCode.cpp

```

//IT21111488
#include <iostream>
#include <cstring>
#include "PromotionCode.h"
using namespace std;

PromotionCode::PromotionCode(){
    cout<< " ";
}

void PromotionCode::createCode(int pCodeID, Seller* sell){
    codeID = pCodeID;
    sellerID = sell->getID();
}

int PromotionCode::getPromoCode(){
    return codeID;
}

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

```

Review.h

```

//IT21126956
#pragma once
#include "Item.h"

class Review{
    private:
        int reviewID;
        char description[200];
}

```

```

public:
    Review(int pReviewID);
    void addReview(const char desc[],Item *item);
    int getReviewID();
    void displayReview();
    ~Review();
};

```

Review.cpp

```

//IT21126956
#include <iostream>
#include "Review.h"
#include <cstring>

using namespace std;

Review::Review(int pReviewID){
    reviewID = pReviewID;
}

void Review::addReview(const char desc[],Item *item){
    item->displayItemDetails();
    strcpy(description,desc);
}

int Review::getReviewID(){
    return reviewID;
}

void Review::displayReview(){
    cout << description <<endl;
}

Review::~~Review(){
    cout <<"Deleting review ID" << reviewID <<endl;
}

```

Report.h

```

//IT21126956
#pragma once
class Report{
private:
    int reportID;

```

```
public:
    Report(int uRID);
    void generateReport();
    int getRID();
    void displayReport();
    ~Report();
};
```

Report.cpp

```
//IT21126956
#include <iostream>
#include "Report.h"
using namespace std;

Report::Report(int uRID){
    reportID = uRID;
}

int Report::getRID(){
    return reportID;
}

void Report::generateReport(){

}

void Report::displayReport(){
    //code
}

Report::~~Report(){
    cout <<"Deleting report ID"<< reportID <<endl;
}
```