



Topic : Online Fashion Store

Group no : MLB_05.01_3

Campus : Malabe

Submission Date : 2021/05/30

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
IT20062088	EDIRISINGHE.D.B.	0766059061
IT20150648	JAYAKODY.J.A.M.G.	0766927655
IT20602550	RATHNAYAKA.D.A.L.	0774933881
IT20298258	JAYAMANNA.H.E.	0710441728
IT20234034	MADIWILAGE.I.E.	0778661910

Exercise 1

(1) Requirements

1. Customer can register to the system by providing name, NIC, address, phone number, username and password.
2. Registered customer can login to the system using username and password.
3. Customer can search items.
4. Registered customer can place an order.
5. Registered customer can make payment via card
6. Registered customer can enter the payment details such as name, card number, CVC and bank.
7. Register customer can cancel payment.
8. Registered customer can remove order or reorder.
9. To complete delivery, register customer needs to enter details such as name, address, email and phone number.
10. Register customer can make a feedback.
11. Administrator can generate reports.
12. Report has a list of order.
13. Administrator can update items and prices.

(2) **Noun Verb analysis**

Customer - Class

System – Outside of the scope

Name - Attribute

NIC - Attribute

Address - Attribute

phone number - Attribute

username - Attribute

password – Attribute

Registered customer - Class

items - Class

order - Class

payment - Class

card - Attribute

payment details – Redundant

Card number – Attribute

CVC – Attribute

Bank – Attribute

Delivery - Class

Details - Redundant

email - Attribute

feedback - Class

Administrator – Outside of the scope

reports – Class

list of order - Attribute

prices – Attribute

Register - Method

Login - Method

Search - Method

Add - Method

Place - Method

Make - Method

complete - Method

Cancel - Method

Request - Method

Remove - Method

Update - Method

Generate - Method

Create - Method

Assign - Method

Analyse - Method

(3) Classes

Customer

Registered customer

Payment

Item

Report

Feedback

Order

Delivery

Exercise 2

(1) CRC Cards

Class -Registered Customer	
Responsibilities	Collaborations
Login to the system	
Search items	Item

Class -Customer	
Responsibilities	Collaborations
Register to the system	
View details	

Class -Payment	
Responsibilities	Collaborations
Make Payment	Registered Customer, Order
Validate Payment	
Verify the payment	
Cancel Payment	Registered Customer, Order

Class -Item	
Responsibilities	Collaborations
Display item list	
Update item details	Administrator
Store item details	
Remove item	Administrator

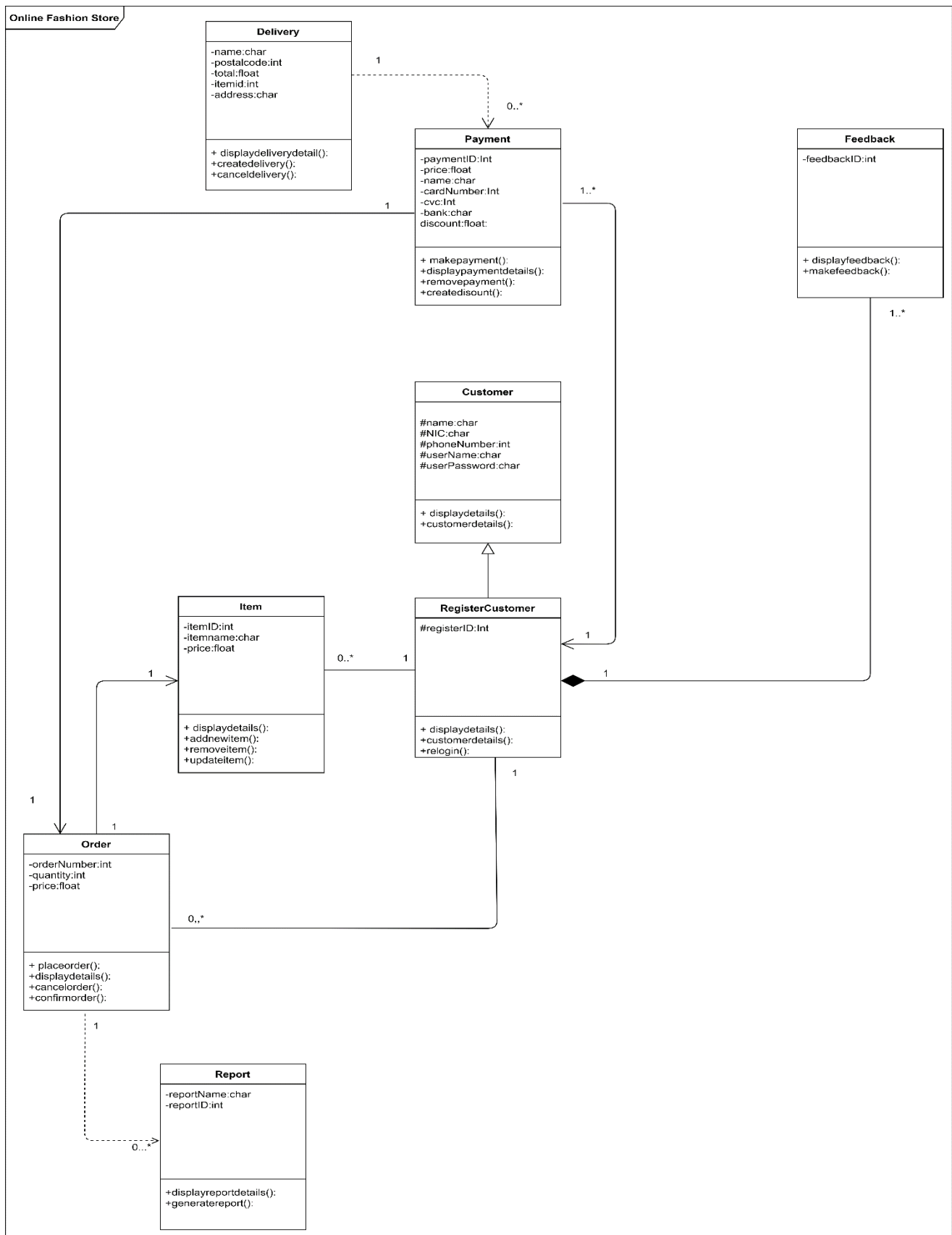
Class -Report	
Responsibilities	Collaborations
Generate Sales report	Order

Class -Feedback	
Responsibilities	Collaborations
Add Feedback	Registered Customer
Display feedback	

Class -Deliveries	
Responsibilities	Collaborations
Display delivery details	
Delivery details	Payment
Cancel delivery	

Class -Order	
Responsibilities	Collaborations
Place an order	Registered Customer, Item
Cancel order	Registered Customer
Confirm order	Payment

Exercise 3



Exercise 4

Customer.h

```
//Customer Class
#pragma once

class Customer{

protected:

    char name[20];
    char NIC[12];
    int phoneNumber;
    char userName[10];
    char userPassword[10];

public:

    Customer();
    Customer(char pName[],char pNIC[],int pPnum,char pUname[],char pUpass[]);
    char *getUname();
    char *getPassword();
    void displaydetails();
    void registercustomer();

~Customer();

};
```

Customer.cpp

```
//implementing Customer Class
#include "Customer.h"
#include <iostream>
#include <cstring>
using namespace std;

Customer::Customer(){

}

Customer::Customer(char pName[],char pNIC[],int pPnum,char pUname[],char
pUpass[]){

strcpy(name,pName);
```



```

strcpy(NIC,pNIC);
phoneNumber = pPnum;
strcpy(userName,pUname);
strcpy(userPassword,pUpass);

}
void Customer:: displaydetails()
{
    cout<<"Name="<<name<<endl;
    cout<<"id="<<NIC<<endl;
    cout<<"Phone Number="<<phoneNumber<<endl;
    cout<<"User Name="<<userName<<endl;
    cout<<"User Password="<<userPassword<<endl;

}

    char * Customer::getUname(){

return userName;

    }
    char * Customer ::getPassword(){

return userPassword;
    }

Customer::~~Customer(){

}

```

Delivery.h

```

//Delivery Class
#include "Payment.h"

class Delivery{

    private:

        char name[20];
        int postalcode;
        float total;
        int itemid;
        char address[50];
        Payment * Pay;

```

```

    public:
    Delivery();
    Delivery(char aname[],int apostalcode,float atotal,int aitemid,char
aaddress[],Payment * Paye);

    void displaydeliverydetails();
    void canceldelivery();

    ~Delivery();

};

```

Delivery.cpp

```

//implementing Delivery class
#include "Delivery.h"
#include <iostream>
#include<cstring>
using namespace std;

Delivery::Delivery() {

}

Delivery::Delivery(char aname[],int apostalcode,float atotal,int aitemid,char
aaddress[],Payment * Paye){

    strcpy(name,aname);
    postalcode = apostalcode;
    total = atotal;
    itemid = aitemid;
    strcpy(address,aaddress);
    Pay=Paye;

}

void Delivery::displaydeliverydetails() {
    cout<<"postalcode="<<postalcode<<endl;
    cout<<"itemid="<<itemid<<endl;
    cout<<"Name="<<name<<endl;
    cout<<"Total="<<total<<endl;
    cout<<"Address="<<address<<endl;

}

void Delivery::canceldelivery(){
}

```

Feedback.h

```
//Feedback Class
#include "Customer.h"

class Feedback{

private:

    int feedbackID;
    char mass[60];
    Customer * Regi;
public:

    Feedback();
    Feedback(int fID,char ms[], Customer *Regis);

    void displayfeedback();

    ~Feedback();

};
```

Feedback.cpp

```
//Implementing Feedback Class
#include "Feedback.h"
#include <iostream>
#include<cstring>
using namespace std;

Feedback::Feedback(){

}

Feedback::Feedback(int fID,char ms[],Customer * Regis )
{

    strcpy(mass,ms);
    feedbackID=fID;
    Regi=Regis;

}

void Feedback::displayfeedback()
{
    cout<<"Feedback ID="<<feedbackID<<endl;
    cout<<"FeedBack"<<mass<<endl;
    cout<<"Name ="<<Regi->getUname()<<endl;
}
```

Item.h

```
//Item Class
#include "RegisterCustomer.h"
class Item
{
    private:
        int itemID;
        float price;
        char iname[20];
        RegisterCustomer * rcs;

    public:
        Item();
        Item(float iitemprice, char iitemname[],int iitemID);
        void displayitemdetails();
        void addnewitems();
        void removeitem();
        ~Item();
};
```

Item.cpp

```
//implementing Item Class
#include "Item.h"

#include <iostream>
#include<cstring>
using namespace std;

Item::Item() {

}
Item::Item(float iitemprice, char iitemname[],int iitemID )
{
    price = iitemprice;
    strcpy(iname,iitemname);
    itemID = iitemID;

}

void Item::displayitemdetails()
{

    cout<<"Item name="<<iname<<endl;
```

```

        cout<<"Itemid="<<itemID<<endl;
        cout<<"Price="<<price<<endl;

    }
    void Item ::addnewitems(){

    }

    void Item ::removeitem(){

    }

    Item::~Item(){

    }
}

```

Order.h

```

//Order Class
#include "Item.h"
#include "RegisterCustomer"
#define SIZE 2
class Order{
    private:

        int orderNumber;
        int quantity;
        Item *it ;
        RegisterCustomer * reg[SIZE];

    public:

        Order();
        Order(int pOnum,int pQua,Item *O);

        void displayorderdetails();
        void cancelorder();
        void confirmorder();

        ~Order();

};

```

Order.cpp

```
//implementing Order Class
#include "Order.h"
#include <iostream>
#include <cstring>

using namespace std;

    Order::Order(){

    }

    Order:: Order(int pOnum,int pQua,Item *O){

    orderNumber = pOnum;
    quantity = pQua;
    it = O;

    }
    void Order::displayorderdetails()

    {
    cout<<"Order Number="<<orderNumber<<endl;
    cout<<"Order Quantity="<<quantity<<endl;
    it->displayitemdetails();
    }

    void Order:: cancelorder()
    {

    }

    void Order:: confirmorder(){

    }

    Order :: ~Order(){

    }
```

Payment.h

```
//Payment Class
#include "Order.h"
#include "Customer.h"
class Payment
{
    private:
        int paymentID;
        float price;
        char name[20];
        int cardNumber;
        int cvc;
        char bank[20];
        float discount;
        Order *orv;
        Customer *cuis;
    public:
        Payment();
        Payment(int bPayID, float bPrice, char bname[], int bCnum, int bCvc, char
bBank[], float bDis,Order *order);

        void displaypaymentdetails();

        void cancelpayment();
        void creatediscount();

        ~Payment();

};
```

Payment.cpp

```
//implementing Payment Class
#include "Payment.h"
#include <iostream>
#include <cstring>
using namespace std;

Payment::Payment(){

}

Payment::Payment(int bPayID,float bPrice,char bname[],int bCnum,int bCvc, char
bBank[], float bDis,Order *order){

paymentID=bPayID;
```

```

price=bPrice;
strcpy(name,bname);
cardNumber=bCnum;
cvc=bCvc;
strcpy(bank,bBank);
discount=bDis;
orv = order;

}

void Payment::displaypaymentdetails()
{
    cout<<"payment ID="<<paymentID<<endl;
    cout<<"price="<<price<<endl;
    cout<<"name="<<name<<endl;
    cout<<"card number="<<cardNumber<<endl;
    cout<<"cvc="<<cvc<<endl;
    cout<<"bank="<<bank<<endl;
    cout<<"discount="<<discount<<endl;
    orv->displayorderdetails();
}
Payment::~~Payment(){

}

```

RegisterCustomer.h

```

//RegisterCustomer Class
#include "Customer.h"
#include "Feedack.h"
#define SIZE 2

class RegisterCustomer : public Customer {

protected:

    int registerID;
    Customer * cus;
    Feedback * fed[SIZE];

public:

    RegisterCustomer();
    RegisterCustomer(char pName[],char pNIC[],int pPnum,char pUname[],char
    pUpass[],int pid, Customer *uCus);
    void displayRdetails();

```



```

void addingfeedbacks(FeedBack *fr[]);
int getRID();
void reglogin(char userAtte[],char passwordAtte[],char *username,char
*password);
~RegisterCustomer();

};

```

RegisterCustomer.cpp

```

//implementing RegisterCustomer Class
#include "RegisterCustomer.h"
#include <iostream>
#include <cstring>
using namespace std;

RegisterCustomer::RegisterCustomer(){

}

RegisterCustomer::RegisterCustomer(char pName[],char pNIC[],int pPnum,char
pUname[],char pUpass[],int pid,Customer *uCus){

strcpy(name,pName);
strcpy(NIC,pNIC);
phoneNumber = pPnum;
strcpy(userName,pUname);
strcpy(userPassword,pUpass);
registerID=pid;
cus =uCus;
}

void RegisterCustomer ::displayRdetails(){

cout<<"registerID ="<<registerID<<endl;
cus->displaydetails();
}

int RegisterCustomer::getRID(){

return registerID;
}

void RegisterCustomer ::reglogin(char userAtte[],char passwordAtte[],char
*username,char *password){

if(strcmp(userAtte,username) ==0 && strcmp(passwordAtte,password)==0){

cout << "Login success" <<endl;
}
}

```

```

else {
cout << "Login fail" <<endl;
}
}
void addingfeedbacks(FeedBack *fr[]){

}
RegisterCustomer::~RegisterCustomer(){

}
}

```

Report.h

```

//Report Class
#include "Order.h"
#include "Delivery.h"
class Report{

    private:

        char reportName[20];
        int reportID;
        Delivery * deli;
    public:
        Report();
        Report(char rreportName[],int rreportID,Delivery *del);

        void displayreportdetails();
        void generatereport();

        ~Report();

};

```

Report.cpp

```
//implementing Report class
#include "Report.h"
#include <iostream>
#include<cstring>
using namespace std;

Report::Report() {

}

Report::Report(char rreportName[],int rreportID ){

strcpy(reportName,rreportName);
reportID = rreportID;

}

void Report::displayreportdetails(){
    cout<<"name="<<reportName<<endl;
    cout<<"report ID="<<reportID<<endl;

}

void Report::generatereport(){

}

Report::~~Report(){

}
```

Main.cpp

```
#include "Delivery.h"
#include "Customer.h"
#include "Feedback.h"
#include "RegisterCustomer.h"
#include "Item.h"
#include "Order.h"
#include "Report.h"
#include "Payment.h"
#include<iostream>
#include<cstring>
using namespace std;

int main() {

Customer *C1;
C1 = new Customer((char *)"malsha",(char *)"98765v",0772340,(char
*)"malshi",(char *)"malsha89");

RegisterCustomer *Rc1,*Rc2;
Rc1 = new RegisterCustomer((char *)"malsha",(char *)"98765v",0772340,(char
*)"malshi",(char *)"malsha89",123,C1);

Rc1->displayRdetails();

Rc1->reglogin((char *)"malsha",(char *)"malsha89",C1->getUname(),C1-
>getPassword());

Delivery *D1;

D1= new Delivery((char *)"driver",54664,321,0-98,(char *)"colombo
8hhadad",P1);
D1->displaydeliverydetails();

Feedback *F1= new Feedback(1234,(char *)"sdsdsddaddada", C[0]);
F1->displayfeedback();

Item *I1=new Item(1000,(char *)"item",345);
I1->displayitemdetails();
```

```
Order *O2=new Order(1243,5,I1);  
O2->displayorderdetails();
```

```
Payment *P1;  
P1=new Payment(1213,44342,(char *)"mine",2311331,856,(char *)"BOC",4000,O2);  
P1->displaypaymentdetails();
```

```
Report *R1 =new Report((char *)"Hino",12232,D1);  
R1->displayreportdetails();
```

```
delete C1;  
delete Rc1;  
delete D1;  
delete F1;  
delete I1;  
delete O2;  
delete P1;  
delete R1;
```

```
}
```