Assignment – 02

IT1050 –Object Oriented Concepts. Year 01 Semester 02

1. Description of the Requirements

- Visitors must register for the website by providing required information such as name, date of birth, password, address, email, and phone number.
- Users can use their chosen username and password to log in after registering.
- All the users can view, browse products
- Upon registration all users can make purchases, add items to cart, add items to Wishlist, check the FAQ section.
- Registered users have several payment options including PayPal, debit cards, and credit cards when making a purchase.
- To place an order, users can browse products, add items to the shopping cart and checkout.
- Customers can unlock an exclusive 5% discount on every transaction over Rs 20,000.
- Customers can choose to cancel their orders if necessary.
- Registered users can inquire about products.
- Registered users can add feedback and reviews on purchased items and share their experience shopping with the website.
- Customers can edit and update their shipping details.
- Administrators of fashion stores can log in, handle consumer inquiries, reply to questions, and offer support as required.
- Administrators have their own login and password to log in to the system.
- Administrators can control product listings, accept, or reject orders, and update inventory information.
- Administrators can activate or deactivate user accounts.
- Managers can log in and manage customer inquiries, respond to questions, and help as necessary.
- Managers can generate reports such as reports, and customer feedback reports to monitor the performance of the online fashion store.

1. Description of the Requirements

Unregistered User Class	
Responsibilities:	**Collaborators:**
Register for the website by providing required information.	
Login using username and password	
Browse product	Product
View product.	Product

Registered User Class	
Responsibilities:	**Collaborators:**
Login using username and password	
View product.	Product
Browse product	Product
Make purchase	Order, Payment
Add item to cart	Order, Product
Add item to Wishlist	Product
View FAQ	
Inquire about products	Inquiry
Cancel orders	Order
Edit/ update user details	
Add feedback and reviews	Feedback
edit and update their shipping	Administrator

Administrator Class	
Responsibilities:	**Collaborators:**
Login using username and password	
Handle consumer inquiries.	Inquiry
Reply to questions.	Inquiry, Feedback
Offer support	
Control product listings	Product
Accept/reject orders	Order
Update inventory information	Product
Activate/deactivate user accounts	Registered user

Product Class	
Responsibilities:	**Collaborators:**
Provide product information	Registered User, Order
Maintain product details	Administrator

Order Class	
Responsibilities:	**Collaborators:**
Manage orders.	Registered User, Administrator
Process payments.	Administrator, Payment

Payment Class	
Responsibilities:	**Collaborators:**
Handle payment processing	Registered User, Order

Inquiry Class	
Responsibilities:	**Collaborators:**
Handle user inquiries	Registered User, Administrator, Manager

Feedback Class	
Responsibilities:	**Collaborators:**
Manage feedback and reviews	Registered user, Administrator

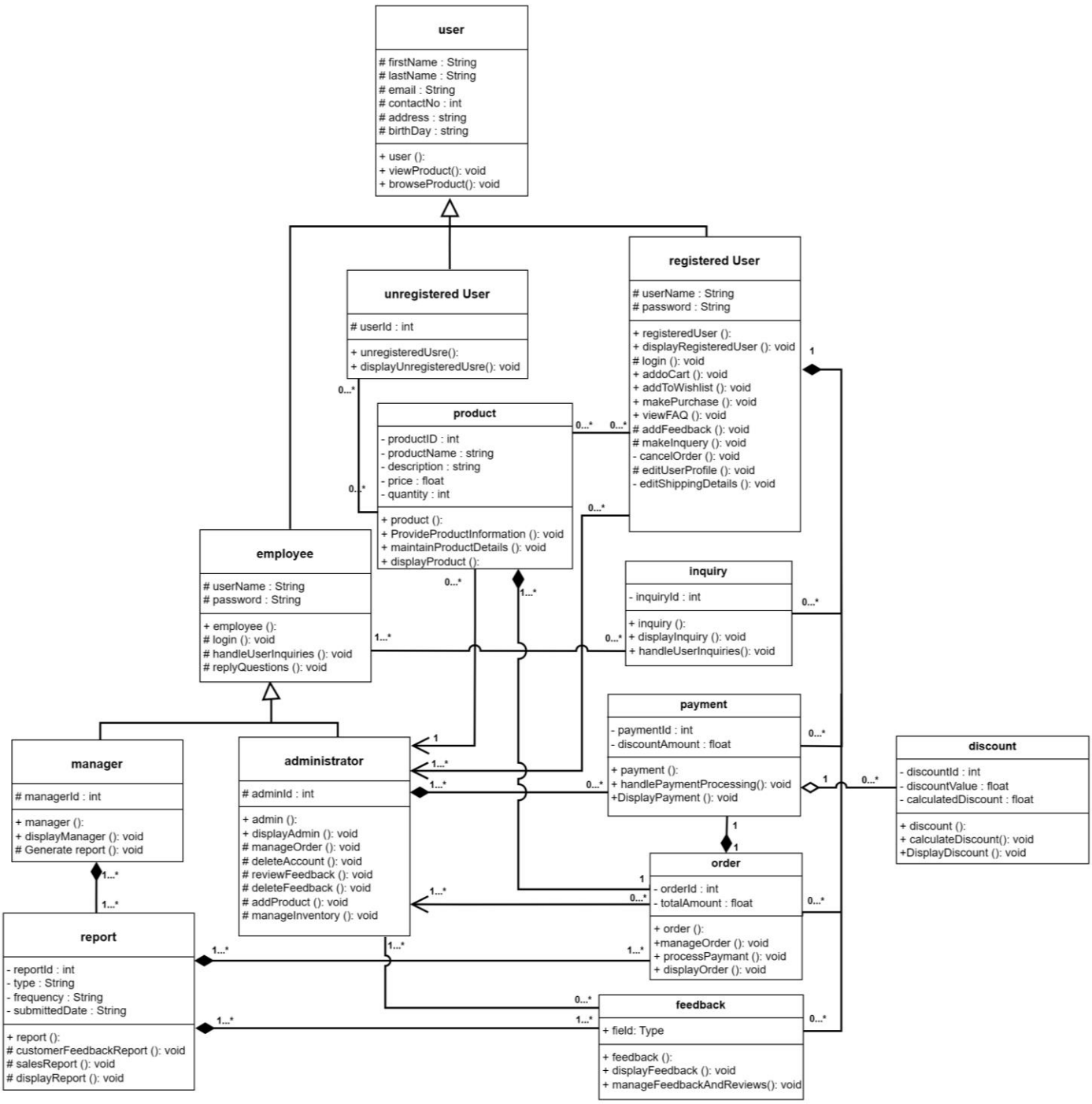
Manager Class	
Responsibilities:	**Collaborators:**
Login using username and password	
manage customer inquiries	Inquiry
respond to questions	Inquiry, Feedback
generate reports	Report

Report Class	
Responsibilities:	**Collaborators:**
customer feedback reports	Feedback, Manager
Sales report	Manager, Order

Discount Class	
Responsibilities:	**Collaborators:**
Handle payment processing	Registered User, Order

Assignment – 02

IT1050 –Object Oriented Concepts. Year 01 Semester 02



Assignment – 02

IT1050 –Object Oriented Concepts. Year 01 Semester 02

Main.cpp

```
#include <iostream>
#include<string>
#include "inquiry.h"
#include "discount.h"
#include "payment.h"
using namespace std;

int main()
{
inquiry* i1 = new inquiry (111, "angeli", "076 2062013", "This is my inquiry");
i1->displayinquiry();
delete i1;

payment* p1 = new payment(222, 40000);
p1->displayPayment();
   delete p1;

   payment* p2 = new payment(333, 3000);
p2->displayPayment();
delete p2;

return 0;
}
```

payment.h

```
#pragma once
#include "discount.h"
class payment
{
private:
int paymentId;
float discountAmount;
float totalAmount;
discount* Discount;

public:
payment();
payment(int id, float total);
void applyDiscount();
void displayPayment();
~payment();

};
```

payment.cpp

```
#include "payment.h"
#include "discount.h"
#include <iostream>
using namespace std;

payment::payment(int id, float total) {
   paymentId = id;
   totalAmount = total;
   discountAmount = 0;
   Discount = new discount(0);
}
void payment::displayPayment() {
   applyDiscount();
   std::cout << "Payment ID     : " << paymentId << endl;
   std::cout << "Total Amount   : " << totalAmount << endl;
   std::cout << "Discount Amount: " << discountAmount << endl << endl;
   std::cout << "Total Payment  : " << totalAmount - discountAmount << endl;
}
void payment::applyDiscount() {
      Discount->calculateDiscount(totalAmount);
      discountAmount = Discount->getCalculatedDiscount();
}

float discount::getCalculatedDiscount() {
   return calculatedDiscount;
}

payment::~payment() {
  delete Discount;
}
```

discount.h

```
#pragma once
class discount
{ private:
int discountId;
float discountValue;
float calculatedDiscount;

public:
discount(int id);
void calculateDiscount(float totalAmount);
float getCalculatedDiscount();
~discount();
};
```

discount.cpp

```cpp
#include "discount.h"
#include <iostream>
using namespace std;

discount::discount(int id) {
    discountId = id;
    discountValue = 0.05;
    calculatedDiscount = 0;
}

void discount::calculateDiscount(float totalAmount) {
    if (totalAmount >= 20000) {
        calculatedDiscount = totalAmount * discountValue;
    }
    else {
        calculatedDiscount = 0;
    }
}
discount::~discount() {
    cout << "....." << endl<< endl;
}
```

inquiry.h

```
#pragma once
#include<string>
#include "inquiry.h"
using namespace std;

class inquiry
{
private:
   int inquiryId;
   string username;
   string contactNo;
   string description;

public:
   inquiry();
   inquiry(int inq, string uname, string tpno, string desc);
   void displayinquiry();
   ~inquiry();
};
```

inquiry.cpp

```
#include<iostream>
#include<string>
#include "inquiry.h"
using namespace std;

inquiry::inquiry() {
   inquiryId = 0;
   username = "";
   contactNo = "";
   description = "";
 }

inquiry::inquiry(int inq, string uname ,string tpno, string desc) {
   inquiryId = inq;
   username = uname;
   contactNo = tpno;
   description = desc;
}
void inquiry::displayinquiry() {
   cout << "inquiry ID    :  " <<inquiryId << endl;
   cout << "Customer name :  " << username << endl;
   cout << "Contact NO    :  " <<contactNo << endl;
   cout << "inquiry       :  " <<description << endl;
}
inquiry::~inquiry() {
   cout << "....................................................." << endl << endl;
}
```

SLIIT
Discover Your Future

BSc (Hons) in Information Technology

Assignment – 02

IT1050 –Object Oriented Concepts. Year 01 Semester 02