



Topic : Online Shopping Mall

Group no : 7

Campus : Malabe

Submission Date: 05 / 19 /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
IT21462320	Perera K.R.M	0771596475
IT21365232	Bandara U.G.B.D	0771168665
IT21369810	Wickramaarachchi J.C	0716787917
IT21372308	Dissanayake W.P.D.B.	0767369434 /0701899631
IT21365164	N.A.S.L.Dissanayake	0762177832

Part I

Requirements for Online Shopping Mall

- User can login using a username and a password.
- User can view, browse a product on the home page.
- Customer can add, view, update and remove items in the shopping cart and change quantity of the added items.
- Registered customer can place an order and check the status of the order.
- Customer can select a shipping method and update shipping info when placing the order.
- Users can update their profile details, reset password.
- Buyer can specify a payment method and update payment details when checking out.
- Administrator can add users, delete users, verify users.
- Administrator can add categories, validate listed products, and delete products from the system.
- System calculates total, total discount, total sales tax, and subtotal for products when customer viewing order details and also calculates shipping cost when he/she selects a shipping method.
- Administrator can be able to view all the users' profiles.
- Seller can add new items, edit product details to his/her shop.

Identified Classes

1. User
2. Customer
3. Seller
4. Administrator
5. Product
6. Shopping Cart
7. Order
8. Order Details
9. Shipping Info
10. Payment

CRC Cards for Online Shopping Mall

User Class	
Responsibilities	Collaborations
Store details of user Validate user	

Administrator Class	
Responsibilities	Collaborations
Store profile details Manage users	User

Seller Class	
Responsibilities	Collaborations
Store profile details Manage products	Product

Customer Class	
Responsibilities	Collaborations
Store profile details Browse products Add products to shopping cart	Product Shopping Cart

Shopping Cart Class	
Responsibilities	Collaborations
Store details of shopping cart View shopping cart	

Product Class	
Responsibilities	Collaborations
Store details of item View products	

Order Class	
Responsibilities	Collaborations
Place order Status of order	Payment , Shipping Info

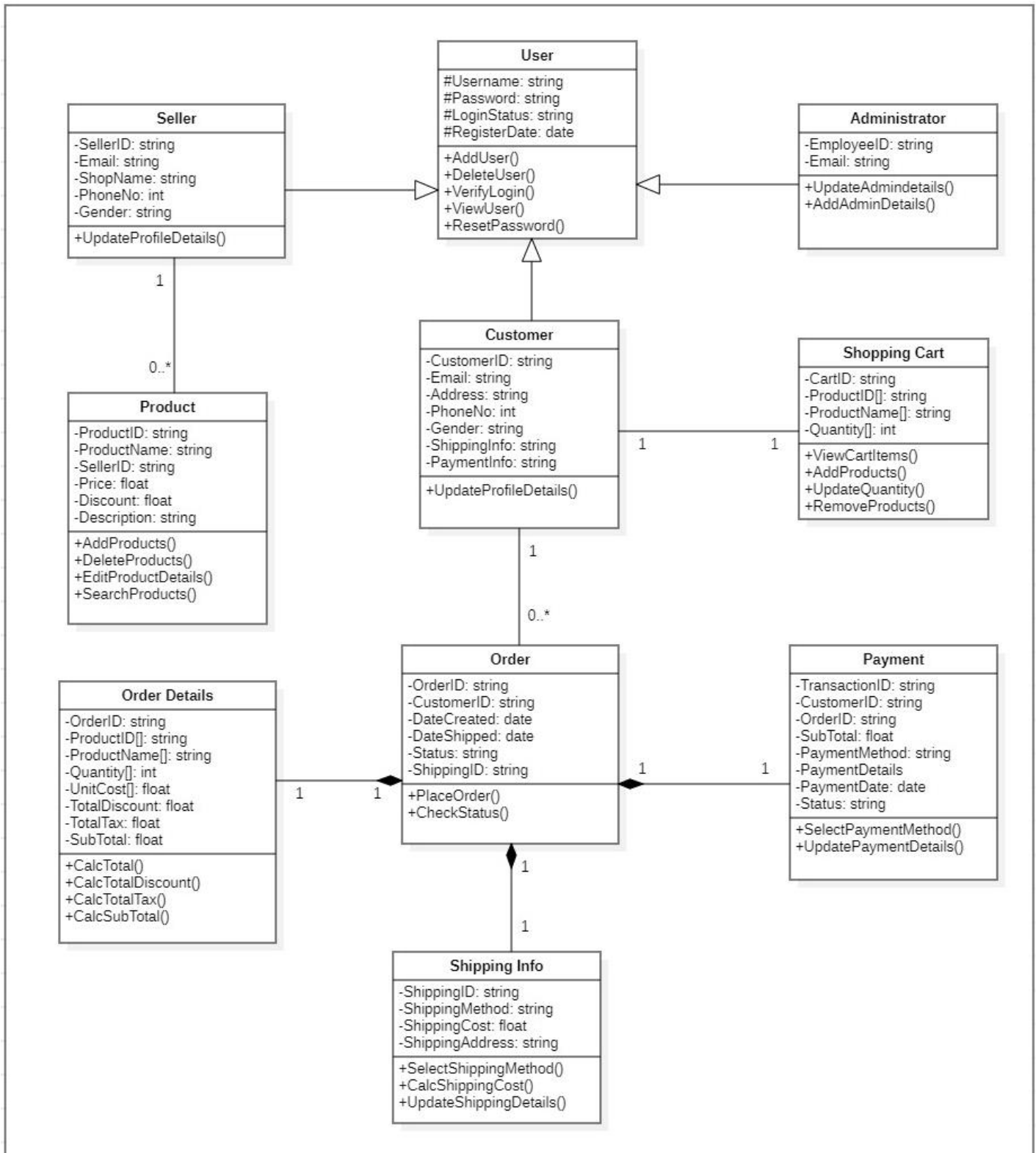
Order Details Class	
Responsibilities	Collaborations
Calculate sub total	Shipping Info

Shipping Info Class	
Responsibilities	Collaborations
Store shipping details Calculate shipping cost	

Payment Class	
Responsibilities	Collaborations
Store payment details	

Exercise 1:

Class diagram



Exercise 2:

Coding

```
#include <iostream>
#include <string>
#define SIZE1 10
#define SIZE2 5
#define SIZE3 100
#define SIZE4 7

using namespace std;

class User {
protected:
    string Username;
    string Password;
    string LoginStatus;
    string RegisterDate;

public:
    User();
    User( string pUsername, string pPassword, string pLoginStatus,
          string pRegisterDate);
    void AddUser();
    void DeleteUser();
    void VerifyLogin();
    void ViewUser();
    void ResetPassword();
    ~User();
};

class Customer : public User {
private:
    string CustomerID;
    string Email;
    string Address;
    int PhoneN;
    string Gender;
    string ShoppingInfo;
    string PaymentInfo;
    ShoppingCart* cart;
    Order* order[SIZE4];

public:
    Customer();
    Customer( string pUsername, string pPassword, string pLoginStatus,
              string pRegisterDate, string pCustomerID, string pEmail,
              string pAddress, int pPhoneN, string pGender,
              string pShoppingInfo, string pPaymentInfo);
    void UpdateProfileDetails();
    ~Customer();
};
```



```

class Administrator : public User {
private:
    string EmployeeID;
    string Email;
public:
    Administrator();
    Administrator( string pUsername, string pPassword, string pLoginStatus,
                  string pRegisterDate, string EmployeeID, string Email);
    void UpdateAdminDetails();
    void AddAdminDetails();
    ~Administrator();
};

class Seller : public User {
private:
    string SellerID;
    string Email;
    string ShopName;
    int PhoneNo;
    string Gender;
    Product* item[SIZE4];

public:
    Seller();
    Seller( string pUsername, string pPassword, string pLoginStatus,
            string pRegisterDate, string SellerID, string Email, string
ShopName,
            int PhoneNo, string Gender);
    void UpdateProfileDetails();
    ~Seller();
};

class ShoppingCart {
private:
    string CartID;
    string ProducID[SIZE1];
    string ProductName[SIZE1];
    int Quantity[SIZE1];
    Customer* cus2;
public:
    ShoppingCart();
    ShoppingCart( string CartID, string ProducID[], string ProductName[],
                  int Quantity[]);
    void ViewCartItems();
    void AddProducts();
    void UpdateQuantity();
    void RemoveProducts();
    ~ShoppingCart();
};

```

```

class Payment {
private:
    string TransactionID;
    string CustomerID;
    string OrderID;
    float Subtotal;
    string PaymentMethod;
    string PaymentDetails;
    string PaymentDate;
    string Status;
public:
    Payment();
    Payment( string TransactionID, string CustomerID, string OrderID,
            float Subtotal, string PaymentMethod, string PaymentDetails,
            string PaymentDate, string Status);
    void SelectPaymentMethod();
    void UpdatePaymentDetails();
    ~Payment();
};

class ShippingInfo {
private:
    string ShippingID;
    string ShippingMethod;
    float ShippingCost;
    string ShippingAddress;
public:
    ShippingInfo();
    ShippingInfo( string ShippingID, string ShippingMethod,
                float ShippingCost, string ShippingAddress);
    void SelectShippingMethod();
    float CalcShippingCost();
    void UpdateShippingDetails();
    ~ShippingInfo();
};

class OrderDetails {
private:
    string OrderID;
    string ProductID[SIZE2];
    string ProductName[SIZE2];
    int Quantity[SIZE2];
    float UnitCost[SIZE2];
    float TotalDiscount;
    float TotalTax;
    float SubTotal;
public:
    OrderDetails();
    OrderDetails( string OrderID, string ProductID[], string ProductName[],
                int Quantity[], float UnitCost[], float TotalDiscount,
                float TotalTax, float SubTotal);
    float CalcTotal();
    float CalcTotalDiscount();
    float CalcTotalTax();
    float CalcSubTotal();
    ~OrderDetails();
};

```

```

class Order {
private:
    string OrderID;
    string CustomerID;
    string DateCreated;
    string DateShipped;
    string Status;
    string ShippingID;
    Customer* cus1;
    Payment* pay1;
    ShippingInfo* shipinfo1;
    OrderDetails* orderinfo1;
public:
    Order();
    Order( string OrderID, string CustomerID, string DateCreated,
          string DateShipped, string Status,
          string ShippingID, Customer* pCus);
    void PlaceOrder();
    void CheckStatus();
    ~Order();
};

class Product {
private:
    string ProductID;
    string ProductName;
    string SellerID;
    float Price;
    float Discount;
    string Description;
    Seller* s01;

public:
    Product();
    Product( string ProductID, string ProductName, string SellerID,
            float Price, float Discount, string Description);
    void AddProducts();
    void DeleteProducts();
    void EditProductDetails();
    void SearchProducts();
    ~Product();
};

int main() {

    Customer *c1 = new Customer();
    Administrator* a1 = new Administrator();
    Seller* s1 = new Seller();
    Product* item1 = new Product();
    Order* order1 = new Order();
    ShoppingCart* SC1 = new ShoppingCart();

    delete c1;
    delete a1;
    delete s1;
    delete item1;
    delete order1;
    delete SC1;

    return 0;
}

```

Individual Contribution

- IT21462320 Perera K.R.M
 - Drew User and Administrator classes in class diagram.
 - Wrote code for User class, Administrator class and the main program.
- IT21365232 Bandara U.G.B.D
 - Drew Order and Order Details classes in class diagram.
 - Wrote code for Order class and Order Details class.
- IT21369810 Wickramarachchi J.C
 - Drew Customer and Shopping Cart classes in class diagram.
 - Wrote code for Customer class and Shopping Cart class.
- IT21372308 Dissanayake W.P.D.B.
 - Drew Seller and Product classes in class diagram.
 - Wrote code for Seller class and Product class.
- IT21365164 N.A.S.L.Dissanayake
 - Drew Shipping Info and classes in class diagram.
 - Wrote code for Shipping Info class and Payment class.