



Topic :Online Bookstore

Group no :MLB_07.01_05

Campus : Malabe

Submission Date : 20/05/2022

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
IT21156960	C.K.KODITHUWAKKU	0786714662
IT21155802	D.S.C.WIJESURIYA	0773308505
IT21157646	W.U.P.SANDARUWAN	0716211256
IT21155048	GANEPOLA G.A.N.B	0770411600
IT21184826	W.D.K.D.H. PATHIRANA	0717624117

Online Bookstore

Requirements

1. First an unregistered user of the Online Book store system needs to register by providing details such as name, passport number, and email address.
2. Then registered user can log in to the system by using user credentials.
3. Customer can edit the profile details in user account according to their preferences
4. Customer can search books by book title and author's name.
5. Registered user can add selected books to the cart.
6. Registered user can edit the number of books in the cart.
7. System displays the total amount including offers if they are available.
8. Customer selects the payment methods such as debit card, credit card, and PayPal according to their preferences.
9. Then the customer can confirm or cancel the order.
10. According to the book type customer can either download or receive it as a delivery.
11. Customer can give feedbacks to the system.
12. Customer can request books.
13. Financial staff validates payments.
14. Financial staff sends an order confirmation message to the customer who placed the order.
15. Financial staff checks and send the order details to the delivery staff.
16. Delivery staff checks order details and arranges the delivery.
17. Financial staff creates monthly financial reports.
18. Product manager can modify and restock the books in the system.
19. Product manager generate the list of restock books.
20. Admin can restrict the customer's accounts and respond to requests and feedbacks.

NOUN VERB ANALYSIS

(NOUN in red color VERB in blue color)

1. First an **unregistered user** of the **Online Book store** system needs to **register** by **providing** details such as **name**, **passport number**, and **email address**.
2. Then **registered user** can **log in** to the system by **using** user credentials.
3. **Customer** can **edit** the profile details in **user account** according to their preferences
4. **Customer** can **search books** by **book title** and **author's name**.
5. **Registered user** can **add** selected **books** to the **cart**.
6. **Registered user** can **edit** the number of **books** in the **cart**.
7. **System** **displays** the **total amount** including **offers** if they are available.
8. **Customer** **selects** the **payment** methods such as **debit card**, **credit card**, and **PayPal** according to their preferences.
9. Then the **customer** can **confirm or cancel** the **order**.
10. According to the book type **customer** can either **download** or **receive** it as a **delivery**.
11. **Customer** can **give feedbacks** to the **system**.
12. **Customer** can **request books**.
13. **Financial staff** **validates** **payments**.
14. **Financial staff** **sends** an **order confirmation message** to the **customer** who placed the **order**.
15. **Financial staff** **checks** and **send** the **order details** to the **delivery staff**.
16. **Delivery staff** **checks** **order details** and **arranges** the **delivery**.
17. **Financial staff** **creates** monthly **financial reports**.
18. **Product manager** can **modify** and **restock** the **books** in the **system**.
19. **Product manager** **generates** the **list of restock books**.
20. **Admin** can **restrict** the customer's accounts and **respond** to **requests** and **feedbacks**.

Identified Classes

Customer

Book

Cart

PayPal

Order

Feedbacks

Financial staff

Report

Bank card

Delivery staff

Payment

Reasons for rejected nouns

Unregistered user - Redundant

Online Book store - Outside scope of system

Name, passport number - An attribute

Email address - An attribute

Registered user - Redundant

User account - Meta language

Title - An attribute

Author's name - An attribute

System - Outside scope of system

Total amount - An event or an operation

Offers - An event or an operation

Debit card - Redundant

Credit card - Redundant

Delivery - An event or an operation

Order confirmation message - An event or an operation

Order details - An attribute

Product manager - Outside scope of system

List of restock books - Redundant

Admin - Outside scope of system

Requests - Meta language

CRC CARDS FOR IDENTIFIED CLASSES

Customer	
Responsibilities	Collaborators
Register to the Online Bookstore	
Log in	
Edit Account details	
Edit the cart	cart
Add feedback	Feedback

Books	
Responsibilities	Collaborators
Store book details	
Add books	
Update books	
Delete books	

Payment	
Responsibilities	Collaborators
Calculate total amount	Books
Store payment details	
Validate payment	

Credit/Debit card	
Responsibilities	Collaborators
Store card details	Payment

PayPal	
Responsibilities	Collaborators
Store PayPal details	Payment

Financial staff	
Responsibilities	Collaborators
Send an order confirmation message	Customer
Creates a monthly financial report	

Delivery staff	
Responsibilities	Collaborators
Arrange delivery	Order

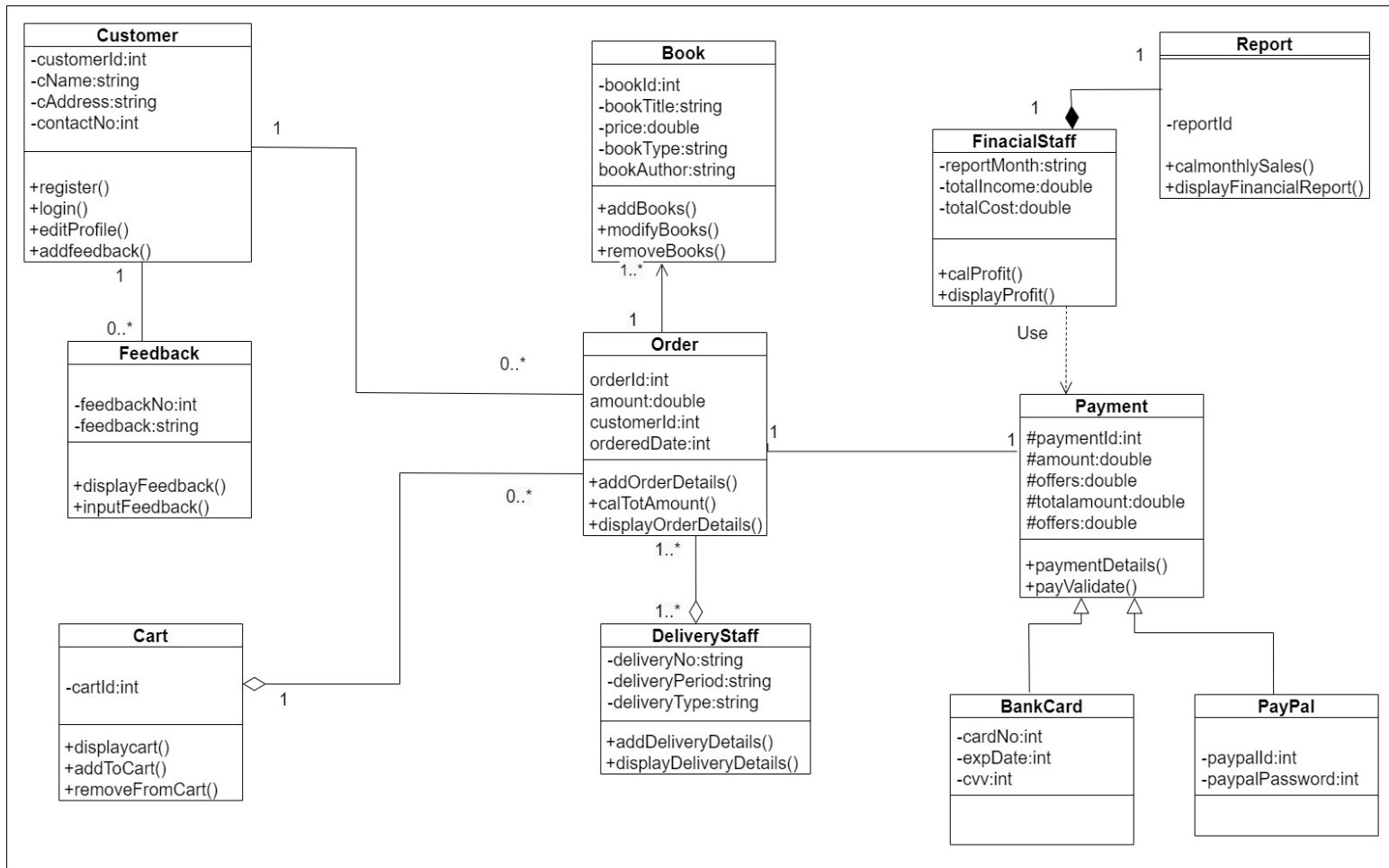
Cart	
Responsibilities	Collaborators
Store order details	Order

Order	
Responsibilities	Collaborators
Make an order	Book, customer
Confirm the order	Payment

Feedback	
Responsibilities	Collaborators
Display feedbacks	Customers
Store description of feedback	

Report	
Responsibilities	Collaborators
Store list of previous orders	Order, customer
List of restock books	Book

Class Diagram for Online Bookstore



Code

```
#include <iostream>

#include<cstring>

using namespace std;


class Book;

class Customer;

class Order;

class Cart;

class DeliverStaff;

class Payment;

class BankCard;

class PayPal;

class Report;

class Feedback;

class FinancialStaff;


class Book {
    private:
        string bookId;
        string booktitle;
        double bookPrice;
        string bookType;
        string bookAuthor;
    public:
```

```
    Book();

    Book(string bookid, double price, string book_type, string book_author);

    void addBooks();

    void modifyBooks();

    void removeBooks();

};
```

```
class Customer{

    private:

        string customerId;

        string cName;

        string cAddress;

        string contactNo;

        Order *order[10]; //Bi-directional Association

    public:

        Customer();

        Customer(string cId, string name, string address,string contactno);

        void CustomerRegister();

        void login();

        void editProfile();

        void addFeedback();

};
```

```
class Order{

    private:

        string orderId;
```

```

        double amount;

        string orderedDate;

        Customer *customer; //Bi-directional Association

        Book *Obook; //Uni-directional Association

    public:

        Order();

        Order(string orderId,double amount, string orderddate);

        void addOrderDetails(string orderId,double amount, string orderddate,Book
*book);

        void calTotAmount();

        void displayOrderDetails();

};

```

```

class Cart {

private:

    string cartId;

    Order *order[10];

public:

    Cart();

    Cart(string cardId);

    void displayCart();

    void addToCart(Order *getOrder);

    void removeFromCart();

};

```

```

class DeliveryStaff {

```

private:

string deliveryNo;

string deliveryPeriod;

string deliveryType;

Order *confirmOrder[10]; //orders which paid and confirmed

public:

DeliveryStaff();

DeliveryStaff(string delivery_no,string delivery_period,string delivery_type);

void addDeliveryDetails(Order *cOrder);

void displayDeliveryDetails();

};

class Payment {

protected:

string paymentId;

double amount;

double offers;

double totalamount;

public:

Payment();

Payment(string payment_id,double Amount,double Offers,double total_amount);

void paymentDetails();

void payValidate();

};

```
class BankCard: public Payment {  
private:  
    int cardNo;  
    string expDate;  
    int cvv;  
public:  
    BankCard();  
    BankCard(int card_no,string exp_date,int Cvv);  
};
```

```
class PayPal :public Payment {  
private:  
    string paypalId;  
    string paypalPassword;  
public:  
    PayPal();  
    PayPal(string paypal_id,string paypalPassword);  
};
```

```
class FinancialStaff {  
private:  
    string reportMonth;  
    double totalIncome;  
    double totalCost;  
    Report* report[8];
```

```

        Payment* payment[7];

public:

    FinancialStaff();

    FinancialStaff(string month, double income, double cost);

        double calProfit(Payment *pay1,Payment *pay2);

        void displayProfit();

};

```

```

class Report {

private:

        string reportId;

public:

    Report();

    Report(string report_id);

        double calmonthlySale();

        void dsiplayfinancialReport();

};

```

```

class Feedback {

private:

        int feedbackNo;

        string feedback;

        Customer* customer[20];

public:

    Feedback();

    Feedback(int feedback_no,string feedBack);

```

```
void displayFeedback(Customer *customer1, Customer *customer2);

void inputFeedback(int feedbackno, string description);

};
```

Cpp

```
#include "bookstore.h"

#include <string>
```

```
Book::Book(){

    bookId="";

    bookPrice=0;

    bookType="";

    bookAuthor="";

}
```

```
Book::Book(string bookid, double price, string book_type, string book_author){

    bookId=bookid;

    bookPrice=price;

    bookType=book_type;

    bookAuthor=book_author;

}
```

```
Customer::Customer(){

    customerId = "";

    cName = "";
```



```
cAddress = "";  
contactNo = "";
```

```
}
```

```
Customer::Customer(string cId, string name, string address,string contactno) {
```

```
    customerId = cId;
```

```
    cName = name;
```

```
    cAddress = address;
```

```
    contactNo = contactno;
```

```
}
```

```
void Customer::login(){
```

```
}
```

```
Order::Order(){
```

```
    orderId = "";
```

```
amount = 0;
```

```
orderedDate = "";
```

```
}
```

```
Order::Order(string orderID, double Amount, string ordered_date) {
```

```
    orderId = orderID;
```

```
    amount = Amount;
```

```
    orderedDate = ordered_date;
```

```
}
```

```
Cart::Cart(){
```

```
    cartId="";  
}  
Cart::Cart(string cart_id){  
    cartId=cart_id;  
}
```

```
DeliveryStaff::DeliveryStaff(){  
    deliveryNo="";  
    deliveryPeriod="";  
    deliveryType="";  
}
```

```
DeliveryStaff::DeliveryStaff(string delivery_no,string delivery_period,string delivery_type){  
    deliveryNo= delivery_no;  
    deliveryPeriod=delivery_period;  
    deliveryType=delivery_type;  
}
```

```
Payment::Payment(){  
    paymentId="";  
    amount=0;  
    totalamount=0;  
}
```

```
Payment::Payment(string payment_id,double Amount,double Offers,double total_amount){  
    paymentId=payment_id;  
    amount=Amount;
```

```
offers=Offers;
```

```
totalamount=total_amount;
```

```
}
```

```
BankCard::BankCard(){
```

```
cardNo=0;
```

```
expDate="";
```

```
cvv=0;
```

```
}
```

```
BankCard::BankCard(int card_no,string exp_date,int Cvv){
```

```
cardNo= card_no;
```

```
expDate=exp_date;
```

```
cvv=Cvv;
```

```
}
```

```
PayPal::PayPal(){
```

```
paypalId="";
```

```
paypalPassword="";
```

```
}
```

```
PayPal::PayPal(string paypal_id,string paypalPassword){
```

```
paypalId=paypal_id;
```

```
paypalPassword=paypalPassword;
```

```
}
```

```
FinancialStaff::FinancialStaff(){
```

```
reportMonth="";
```

```

    totalIncome=0.00;

    totalCost=0.00;
}

FinancialStaff::FinancialStaff(string month, double income, double cost){

    reportMonth=month;

    totalIncome=income;

    totalCost=cost;
}


Report::Report(){

    reportId="";
}

Report::Report(string report_id){

    reportId=report_id;
}


Feedback::Feedback(){

    feedbackNo=0;

    feedback="";
}

Feedback::Feedback(int feedback_no,string Feedback){

    feedbackNo=feedback_no;

    feedback=Feedback;
}

```

Main CPP File

```
#include<iostream>

#include"bookstore.h"

using namespace std;

int main()

{

    Customer *c1=new Customer();

    Book *b1=new Book();

    Order *o1=new Order();

    Cart *cart1= new Cart();

    FinancialStaff *f1= new FinancialStaff();

    DeliveryStaff *d1=new DeliveryStaff() ;

    Payment *payment1=new Payment();

    BankCard *card1=new BankCard();

    PayPal *pal1=new PayPal();

    Report *report1=new Report();

    Feedback *feedback1=new Feedback();


}
```