Object Oriented Concepts – IT1050

Assignment 2

Topic: Online Fashion Store

Group Number: MLB_WD_CSNE_01.01_06

Campus: Malabe

Submission Date: 13/06/2023

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 Numbers
IT22360496	D R WICKRAMA ARACHCHI	0740654079
IT22323798	I D N K SIRIWARDHANA	0718559147
IT22324306	K C S P PERERA	0701184851
IT22365378	W M S L WICKRAMASINGHE	0741848886
IT22311122	W V A D N BINATH	0766723176

Description of the Requirements

1. A customer in an online fashion store can browse and select items.

2. A customer can create an account on the website.

3. Once the account is created, a customer can save their shopping cart, place orders, and receive customised recommendations.

4. A customer can add and remove items to their shopping cart before finalising their order.

5. A customer can view the details and price of an item.

6. A customer can view their order history.

7. An order calculates the total price before payment.

8. A customer can receive discounts and promotions and take part in loyalty programmes.

9. A customer can pay for their order using credit cards, debit cards, PayPal, or other payment options.

10. Each payment method is authorised by the website.

11. A content administrator of the online store can manage the products, orders, and customer data on the website.

12. A customer can write one or more reviews for an item.

Identified Classes

1. Client

2. Item

3. Order

4. Payment

5. Loyalty Programme

6. Recommendation

7. Promotion

8. Report

9. Review

CRC Cards

Client Class	
Responsibilities	**Collaborations**
Register	

Item Class	
Responsibilities	**Collaborations**
Browse for an item	
Select an item	
Add an item to cart	
Remove an item from cart	
Display item details	
Display item price	

Order Class	
Responsibilities	**Collaborations**
Place an order	
Finalise an order	Payment
Calculate total price	

Payment Class	
Responsibilities	**Collaborations**
Store payment details	
Authorise different payment methods	

Loyalty Programme Class	
Responsibilities	**Collaborations**
Take part in a loyalty programme	

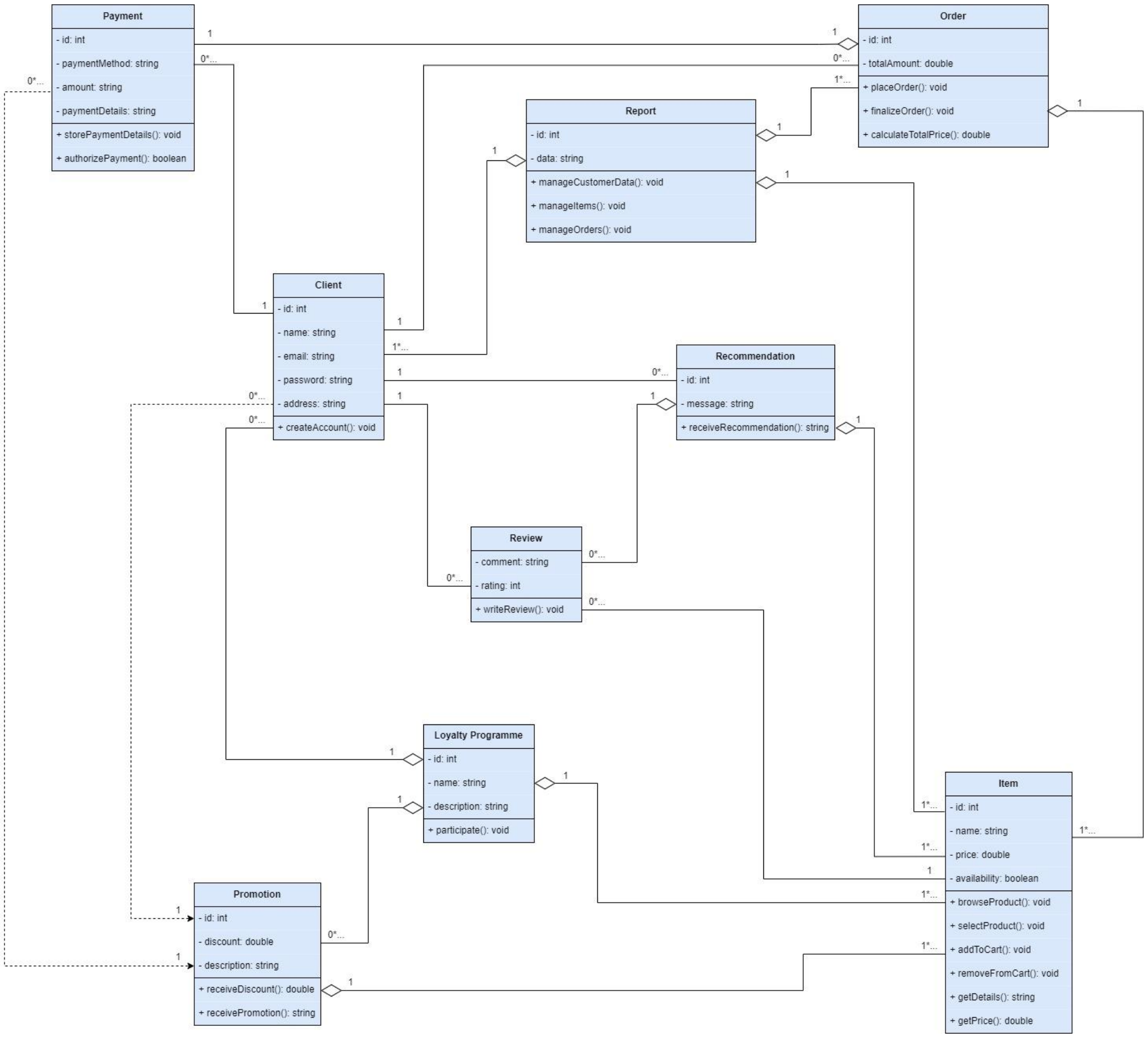
Recommendation Class	
Responsibilities	**Collaborations**
Receive recommendation	

Promotion Class	
Responsibilities	**Collaborations**
Receive discount/promotion	

Report Class	
Responsibilities	**Collaborations**
Manage data of customers	Client
Manage items	Item
Manage orders	Order, Client

Review Class	
Responsibilities	**Collaborations**
Write review	

Class Diagram with UML Notations



Class Coding and Object Creation

Client Class

```
#pragma once
#include "Order.h"
#include "Payment.h"
#include "Recommendation.h"
#include "Review.h"
#include "Promotion.h"
#include <iostream>
using namespace std;

#define SIZE 10

class Client
{
private:
	int id;
	string name;
	string email;
	string password;
	string address;
	//Association with Order
	Order* order[SIZE];
	//Association with Payment
	Payment* payment[SIZE];
	//Association with Recommendation
	Recommendation* recommendation[SIZE];
	//Association with Review
	Review* review[SIZE];

public:
	Client()
	{
		id = 0;
		name = "";
		email = "";
		password = "";
		address = "";
	};
	Client(int pid, string pname, string pemail, string ppassword, string
paddress)
	{
		id = pid;
		name = pname;
		email = pemail;
		password = ppassword;
		address = paddress;
	};

	void createAccount() {}
	void placeOrder(Order* Ord1);
	void addPayment(Payment* Pay1);
	void writeReview(Review* Rev1);
	//Dependency with Promotion
	void usePromotion(Promotion* Pro1) {}
	void receiveRecommendation(Recommendation* Rec1) {}
	void displayClient() {}

	~Client()
	{
		cout << "Deleting Client" << id << endl;
	}
};
```

Item Class

```
#pragma once
#include "Review.h"
#include <iostream>
using namespace std;

#define SIZE 10

class Item
{
private:
        int id;
        string name;
        double price;
        bool availability;
        //Association with Review
        Review* review[SIZE];

public:
        Item()
        {
                id = 0;
                name = "";
                price = 0;
                availability = 0;
        }

        Item(int pid, string pname, double pprice, bool pavailability)
        {
                id = pid;
                name = pname;
                price = pprice;
                availability = pavailability;
        }

        void browseProduct() {}
        void selectProduct() {}
        void addToCart() {}
        void removeFromCart() {}
        void getDetails() {}
        void getPrice() {}
```

```
    void addReview(Review* Rev1) {}
    void displayItem() {}

    ~Item()
    {
        cout << "Deleting Item" << id << endl;
    }
};
```

Order Class

```
#pragma once
#include "Item.h"
#include "Payment.h"
#include <iostream>
using namespace std;

#define SIZE 10

class Order
{
private:
        int id;
        double totalAmount;
        //Association with Client
        Client* Cli;
        //Aggregation with Item
        Item* Ite[SIZE];
        //Aggregation with Payment
        Payment* Pay[SIZE];

public:
        Order()
        {
                id = 0;
                totalAmount = 0;
        };

        Order(int pid, double ptotalAmount, Client* pCli, Item* pIte, Payment* pPay)
        {
                id = pid;
                totalAmount = ptotalAmount;
                Cli = pCli;
                *Ite = pIte;
                *Pay = pPay;
        }

        void addItem(Item* Ite1, Item* Ite2) {}
        void addPayment(Payment* Pay1, Payment* Pay2) {}
        void displayOrder() {}
```

```
    ~Order()
    {
        cout << "Deleting Order" << id << endl;
    }
};
```

Payment Class

```
#pragma once
#include "Client.h"
#include <iostream>
using namespace std;

#define SIZE 10

class Payment
{
private:
	int id;
	string paymentMethod;
	double amount;
	string paymentDetails;
	//Association with Client
	Client* Cli;

public:
	Payment()
	{
		id = 0;
		paymentMethod = "";
		amount = 0;
		paymentDetails = "";
	}

	Payment(int pid, string ppaymentMethod, double pamount, string ppaymentDetails, Client* pCli)
	{
		id = pid;
		paymentMethod = ppaymentMethod;
		amount = pamount;
		paymentDetails = ppaymentDetails;
		Cli = pCli;
	}

	void storePaymentDetails() {}
	bool authorizePayment() {}
	void displayPayment() {}
```

```
        ~Payment()
        {
            cout << "Deleting Payment" << id << endl;
        }
    };
```

Loyalty Programme Class

```
#pragma once
#include "Client.h"
#include "Item.h"
#include "Promotion.h"
#include <iostream>
using namespace std;

#define SIZE 10

class LoyaltyProgramme
{
private:
	int id;
	string name;
	string description;
	//Aggregation with Client
	Client* Cli[SIZE];
	//Aggregation with Item
	Item* Ite[SIZE];
	//Aggregation with Promotion
	Promotion* Pro[SIZE];

public:
	LoyaltyProgramme()
	{
		id = 0;
		name = "";
		description = "";
	}

	void addClient(Client* Cli1, Client* Cli2) {}
	void addItem(Item* Ite1, Item* Ite2) {}
	void addPromotion(Promotion* Pro1, Promotion* Pro2) {}

	void participate() {}
	void displayLoyaltyProgramme() {}

	~LoyaltyProgramme()
	{
```

```
                cout << "Deleting Loyalty Programme" << id << endl;
        }
};
```

Promotion Class

```
#pragma once
#include "Item.h"
#include "Payment.h"
#include <iostream>
using namespace std;

#define SIZE 10

class Promotion
{
private:
	int id;
	double discount;
	string description;
	//Aggregation with Item
	Item* Ite[SIZE];

public:
	Promotion()
	{
		id = 0;
		discount = 0;
		description = "";
	};

	Promotion(int pid, double pdiscount, string pdescription)
	{
		id = pid;
		discount = pdiscount;
		description = pdescription;
	}

	void addItem(Item* Ite1, Item* Ite2) {}
	double receiveDiscount() {}
	double receivePromotion() {}
	//Dependency with Payment
	void addPayment(Payment* Pay1) {}
	void displayPromotion() {}
```

```cpp
	~Promotion()
	{
		cout << "Deleting Promotion" << id << endl;
	}
};
```

Recommendation Class

```
#pragma once
#include "Client.h"
#include "Item.h"
#include "Review.h"
#include <iostream>
using namespace std;

#define SIZE 10

class Recommendation
{
private:
	int id;
	string message;
	//Association with Client
	Client* Cli;
	//Aggregation with Item
	Item* Ite[SIZE];
	//Aggregation with Review
	Review* Rev[SIZE];

public:
	Recommendation()
	{
		id = 0;
		message = "";
	};

	Recommendation(int pid, string pmessage, Client* pCli)
	{
		id = pid;
		message = pmessage;
		Cli = pCli;
	}

	void receiveRecommendation() {}
	void addItem(Item* Ite1, Item* Ite2) {}
	void addReview(Review* Rev1, Review* Rev2) {}
	void displayRecommendation() {}
```

```
    ~Recommendation()
    {
        cout << "Deleting Recommendation" << id << endl;
    }
};
```

Report Class

```cpp
#pragma once
#include "Client.h"
#include "Item.h"
#include "Order.h"
#include <iostream>
using namespace std;

#define SIZE 10

class Report
{
private:
	int id;
	string data;
	//Aggregation with Client
	Client* Cli[SIZE];
	//Aggregation with Item
	Item* Ite[SIZE];
	//Aggregation with Order
	Order* Ord[SIZE];

public:
	Report()
	{
		id = 0;
		data = "";
	}

	void addClient(Client* Cli1, Client* Cli2) {}
	void addItem(Item* Ite1, Item* Ite2) {}
	void addOrder(Order* Ord1, Order* Ord2) {}
	void displayReport() {}

	~Report()
	{
		cout << "Deleting Report" << id << endl;
	}
};
```

Review Class

```
#pragma once
#include "Client.h"
#include "Item.h"
#include <iostream>
using namespace std;

#define SIZE 10

class Review
{
private:
        string comment;
        int rating;
        //Association with Client
        Client* Cli;
        //Association with Item
        Item* Ite;

public:
        Review()
        {
                comment = "";
                rating = 0;
        }

        Review(string pcomment, int prating, Client* pCli, Item* pIte)
        {
                comment = pcomment;
                rating = prating;
                Cli = pCli;
                Ite = pIte;
        }

        void writeReview();
        void displayReview();

        ~Review()
        {
                cout << "Deleting Review" << endl;
        }
    };
```

Main Program

```
#include"Client.h"
#include"Item.h"
#include"Order.h"
#include"Payment.h"
#include"LoyaltyProgramme.h"
#include"Promotion.h"
#include"Recommendation.h"
#include"Report.h"
#include "Review.h"
#include <iostream>

int main()
{
	Client* Cli1;
	Item* Ite1;
	Order* Ord1;
	Payment* Pay1;
	LoyaltyProgramme* Loy1;
	Promotion* Pro1;
	Recommendation* Rec1;
	Report* Rep1;
	Review* Rev1;

	Client* Cli2;
	Item* Ite2;
	Order* Ord2;
	Payment* Pay2;
	LoyaltyProgramme* Loy2;
	Promotion* Pro2;
	Recommendation* Rec2;
	Report* Rep2;
	Review* Rev2;

	return 0;
}
```