Topic : Online Marketing Store

Group no : MLB_7.1_03

Campus : Malabe

Submission Date: 2022/05/19

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**
IT21154812	ATHIF M.T.M	0771281946
IT21155116	M.A.D.A. DILSHAN	0772773096
IT21156342	G.C. PIYUMAL	0770622236
IT21154980	ASIFF. Y.N. A	0712742480
IT21155352	H.R. KAMESH DIVIYANJANA	0757233349

System Requirements

1. People can visit the system and search for products in the system.
2. Users can create a buyer or seller account by entering information such as their name, email, contact number, and address.
3. The registered buyer and registered seller must create a new password for the system. And system validates the new password.
4. Registered users are two types called registered buyers and registered sellers they should log into the system by entering their email and password.
5. Registered buyers can search for products, select the products, add products to the cart, purchase products, and edit buyer account details inside the system.
6. Registered sellers can add products, and product details into the system, edit seller account details, and check seller financial details within the system.
7. Registered sellers can add the discount for particular products and registered buyers can use the valid discount and the system admin validates the discount.
8. Registered buyers can purchase the selected product, Registered sellers can inspect what customer orders products, and system admin update the order details.
9. Registered buyers select the payment method PayPal, cash on delivery, or bank card payment method such as visa or master card.
10. Registered buyers enter the payment details, and the system admin validates the payment details and sent the payment confirmation details to the registered buyer.
11. Registered seller received the payment through the bank account, and the admin update the payment details and generate the payment report.
12. Support service generates the list of order reports and sent the order payment invoice, and buying and selling details through email to the registered user.
13. Support service updates the list of order details and other related details.
14. Registered users send the feedback, support service check, and respond to the feedback.
15. System admin assigns the delivery team and delivery partner to deliver the order to the customer.
16. Delivery team updates the delivered order details.
17. Admin generates the weekly and monthly orders, financial, and user details reports.

System Requirements

1. People can visit the system and search for products in the system.
2. Gust users can create a buyer or seller account by entering information such as their name, email, contact number, and address.
3. The registered buyer and registered seller must create a new password for the system. And system validates the new password.
4. Registered users are two types called registered buyers and registered sellers they should log into the system by entering their email and password.
5. Registered buyers can search for products, select the products, add products to the cart, purchase products, and edit buyer account details inside the system.
6. Registered sellers can add products, and product details into the system, edit seller account details, and check seller financial details within the system.
7. Registered sellers can add the discount for particular products and registered buyers can use the valid discount and the system admin validates the discount.
8. Registered buyers can purchase the selected product, Registered sellers can inspect what customer orders products and system admin update the order.
9. Registered buyers select the payment method PayPal, cash on delivery, or bank card payment method such as visa or master card.
10. Registered buyers enter the payment details, and the system admin validates the payment details and sent the payment confirmation details to the registered buyer.
11. Registered seller received the payment through the bank account, and the admin update the payment details and generate the payment report.
12. Support service generates the list of order reports and sent the order payment invoice, and buying and selling details through email to the registered user.
13. Support service updates the list of order details and other related details.
14. Registered users send the feedback, support service check, and respond to the feedback.
15. System admin assigns the delivery team and delivery partner to deliver the order to the customer.
16. Delivery team updates the delivered order details.
17. Admin generates the weekly and monthly orders, financial reports, and user details reports.

System Requirements

1. People can visit the system and search for products in the system.
2. Gust users can create a buyer or seller account by entering information such as their name, email, contact number, and address.
3. The registered buyer and registered seller must create a new password for the system. And system validates the new password.
4. Registered users are two types called registered buyers and registered sellers they should log into the system by entering their email and password.
5. Registered buyers can search for products, select the products, add products to the cart, purchase products, and edit buyer account details inside the system.
6. Registered sellers can add products, and product details into the system, edit seller account details, and check seller financial details within the system.
7. Registered sellers can add the discount for particular products and registered buyers can use the valid discount and the system admin validates the discount.
8. Registered buyers can purchase the selected product, Registered sellers can inspect what customer orders products, and system admin update the order details.
9. Registered buyers select the payment method PayPal, cash on delivery, or bank card payment method such as visa or master card.
10. Registered buyers enter the payment details, and the system admin validates the payment details and sent the payment confirmation details to the registered buyer.
11. Registered seller received the payment through the bank account, and the admin update the payment details and generate the payment report.
12. Support service generates the list of order reports and sends the order payment invoice, and buying and selling details through email to the registered user.
13. Support service updates the list of order details and other related details.
14. Registered users send the feedback, support service check, and respond to the feedback.
15. System admin assigns the delivery team and delivery partner to deliver the order to the customer.
16. Delivery team updates the delivered order details.
17. Admin generates the weekly and monthly orders, financial, and user details reports.

Nouns

1. Guests
2. Users
3. System
4. Products
5. Buyer or seller account
6. People
7. Name, email, contact number, and address.
8. Registered buyer
9. Registered seller
10. New password
11. Email and password
12. Cart, purchase products
13. Buyer account details
14. Seller account details
15. Financial details
16. Discount
17. System admin
18. Orders products
19. Orders
20. Order
21. Payment method
22. PayPal
23. Cash on delivery
24. Bank card payment
25. Visa or master card
26. Payment details
27. Payment
28. Bank account
29. Payment report
30. List of order reports
31. Order payment invoice
32. Email
33. Registered user
34. Support service
35. Feedback
36. Delivery team
37. Delivery partner
38. Customer
39. Weekly and monthly orders, financial reports, and user details reports

Class:

1. Admin
2. User
3. Register buyer
4. Register seller
5. Product
6. Discount
7. Order
8. Payment
9. Support service
10. Feedback
11. Reports

Reasons for rejecting other nouns

Redundant:

1. Guest/customer – refers to the “user”
2. System Admin – refers to the “Admin”
3. financial reports, user reports, list of order reports – refers to the “Reports”
4. PayPal, bank card – refers to the “Payment”

Outside the Scope:

1. System
2. Bank
3. Delivery partner
4. Delivery team

Attributes:

1. Name, email, contact number, address
2. Password
3. Card
4. Cart, purchase products
5. Buyer account details
6. Seller account details

Meta language:

1. People

An event or an operation:

1. Orders products
2. Purchase products

Verb

1. Visit
2. Search
3. Create
4. Entering information
5. Must create new password
6. Validates the new password
7. Login
8. Entering their email and password
9. Select
10. Add
11. Purchase
12. Inspect
13. Update
14. Enter
15. Received
16. Generate
17. Sends
18. Check
19. Responds
20. Assign
21. Deliver

Methods

1. Admin - Generates reports
   Updates reports
   Check and validate details
   Validate discount

2. User - Visit to page
   Search products
   Register to the system entering information

3. Register buyer - Create an account
   Login into the system
   Search product
   Select the product
   Add the product to the cart
   Purchase the product
   Edit the account details
   Select the payment method
   Display register buyer details

4. Register seller - Create account
   Add the product
   Edit the seller account details
   Check the seller financial details
   Inspect the orders
   Add the discount
   Display registered seller details

5. Product	-	Display the Product list of details Add the product
6. Discount	-	Display discount details Calculate discount details
7. Order	-	list of order details Display list of orders details
8. Payment	-	Keep registered buyer and seller payment details Calculate the total amount of purchasing Confirm payment Validate payment Display payment details
9. Support service	-	Generate the list of order reports Validate the user feedback Send the invoice to user
10. Feedback	-	Store feedback details Display the registered buyer feedback Display the registered seller feedback
11. Reports	-	Generate the reports List the reports

Exercise 2:

CRC Cards for the online Airline Ticket Reservation system

Admin	
Responsibility	Collaborators
Updates reports	Registered buyer, Registered seller, Payment, Orders, Reports
Check and validate details	
Validate discount	Discount

User	
Responsibility	Collaborators
Register to the system entering information	
Create an account	Registered buyer, Registered saller

Register buyer	
Responsibility	Collaborators
Add the product to the cart	Products
Purchase the product	Product, payment
Edit the account details	Reports
Display register buyer details	

Register seller	
Responsibility	Collaborators
Edit account details	
Add the product details	Product
Add the discount amount	
Display the seller financial details	Payment
Display registered seller details	
Display discount	Discount

Product	
Responsibility	Collaborators
Add the product details	
Display the Product list of details	

Discount	
Responsibility	Collaborators
Calculate discount details	
Display discount details	

Order	
Responsibility	Collaborators
list of order details	
Display list of orders details	

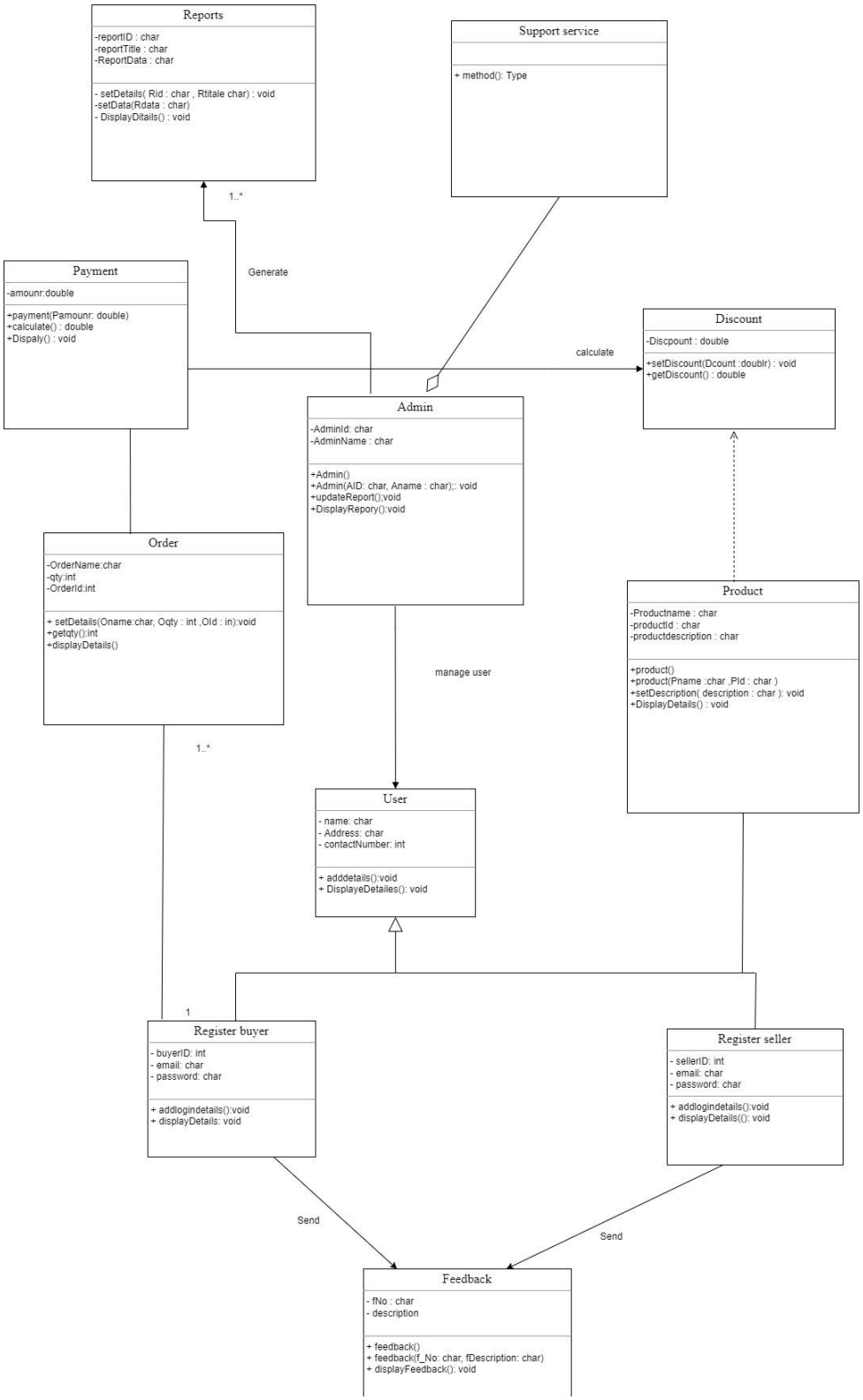
Payment	
Responsibility	Collaborators
Store registered buyer and seller payment details	Registered buyer, Registered seller
Calculate the total amount of purchasing	
Display payment details	

Support service	
Responsibility	Collaborators
Validate the user feedback	Registered buyer, Registered seller
Send the invoice to user	

Feedback	
Responsibility	Collaborators
Store feedback details	Support service
Display the registered buyer feedback	Registered buyer
Display the registered seller feedback	Registered seller

Reports	
Responsibility	Collaborators
Enter the report details	Admin
List the reports details	

Class Diagram



Class: User, RegisterBuyer, RegisterSeller (Inheritance Relationship and association relationship)

```
//IT21154812 Aathif M.T.M
#include<iostream>
#include<cstring>
using namespace std;

class User {
protected:
    char name[20];
    int age;
    char address[40];
    string contactNumber;
public:
    User();
    User(const char uName[], int uAge, const char uAddress[], string
uContactNumber);
    void displayDetails();
};

class feedback {
private:
    char fNo[20];
    char description[50];
public:
    feedback();
    feedback(const char f_NO[], const char fDescription[]);
    void displayFeedback();
};


class RegisterBuyer : public User {
private:
    int buyerID;
    char email[20];
    char password[20];
    feedback* fb_buyer;
public:
    RegisterBuyer();
    RegisterBuyer(const char uName[], int uAge, const char uAddress[], string
uContactNumber, int rb_BuyerID, const char rb_Email[], const char rb_Password[],
feedback* fb_b);
    void displayDetails();
};



class RegisterSeller : public User {
private:
    int sellerID;
    char email[20];
    char password[20];
    feedback* fb_seller;
public:
    RegisterSeller();
```

```
    RegisterSeller(const char uName[], int uAge, const char uAddress[], string
uContactNumber, int rs_SellerID, const char rs_Email[], const char rs_Password[],
feedback* fb_seller);
    void displayDetails();
};

// --- user implementation ---------------

User::User() {

}
User::User(const char uName[], int uAge, const char uAddress[], string
uContactNumber) {
    //cout << "User called" << endl << endl;
    strcpy(name, uName);
    age = uAge;
    strcpy(address, uAddress);
    contactNumber = uContactNumber;
}
void User::displayDetails() {
    cout << "User Name : " << name << endl;
    cout << "Address : " << address << endl;
    cout << "Contact Number : " << contactNumber << endl;
}

// --- feedback imaplementation -----------

feedback::feedback() {

}
feedback::feedback(const char f_No[], const char fDescription[]) {
    strcpy(fNo, f_No);
    strcpy(description, fDescription);
}
void feedback::displayFeedback() {
    cout << "Feedback number : " << fNo << endl;
    cout << "Description : " << description << endl;
    cout << endl;
}

// --- RegisterBuyer implementation ----------

RegisterBuyer::RegisterBuyer() {

}
RegisterBuyer::RegisterBuyer(const char uName[], int uAge, const char uAddress[],
string uContactNumber, int rb_BuyerID, const char rb_Email[], const char
rb_Password[], feedback* fb_b)
    :User(uName, uAge, uAddress, uContactNumber) {
    cout << "RegisterBuyer called" << endl << endl;
    buyerID = rb_BuyerID;
    strcpy(email, rb_Email);
    strcpy(password, rb_Password);
    fb_buyer = fb_b;
}
void RegisterBuyer::displayDetails() {
    cout << endl;
    User::displayDetails();
```

```
    cout << "Register buyer ID : " << buyerID << endl;
    cout << "Register buyer email : " << email << endl;
    cout << "Register buyer password : " << password << endl;
    fb_buyer->displayFeedback();
    cout << endl;
}

// --- RegisterSeller implementation ------------

RegisterSeller::RegisterSeller() {

}
RegisterSeller::RegisterSeller(const char uName[], int uAge, const char uAddress[],
string uContactNumber, int rs_SellerID, const char rs_Email[], const char
rs_Password[], feedback* fb_s)
    :User(uName, uAge, uAddress, uContactNumber) {
    cout << "RegisterSeller called" << endl << endl;
    sellerID = rs_SellerID;
    strcpy(email, rs_Email);
    strcpy(password, rs_Password);
    fb_seller = fb_s;

}
void RegisterSeller::displayDetails() {
    User::displayDetails();
    cout << "Register seller ID : " << sellerID << endl;
    cout << "Register seller email : " << email << endl;
    cout << "Register seller password : " << password << endl;
    fb_seller->displayFeedback();
    cout << endl;
}
```

Class: Admin, Report, SupportService (Aggregation and association)

```
//IT21155352 Kamesh

#include <iostream>
#include<cstring>
#define size 2
using namespace std;

class supportservice
{

private:
	char SuportID[20];
	char SuportType[20];
public:
	void setDetails(const char Sid[], const char sType[])
	{
		strcpy_s(SuportID, Sid);
		strcpy_s(SuportType, sType);
	}
	void Display()
	{
		cout << "Support service ID  :" << SuportID << endl;
		cout << "Support service  type :" << SuportType << endl;

	}
	void displayeedback();
	void DispalyInvoise();
};

class report {

private:
	char ReportID[15];
	char ReportTitale[20];
	char Reportdata[100];

public:
	void setDetails(const char id[], const char titale[])
	{
		strcpy_s(ReportID, id);
		strcpy_s(ReportTitale, titale);
	}
	void setdata(const char data[])
	{
		strcpy_s(Reportdata, data);
	}

	void displayDitalse()
	{
		cout << "ID : " << ReportID << endl;
		cout << "------" << ReportTitale << "----------" << endl << endl;
		cout << Reportdata << endl;

	}
};
```

```
class admin {

private:
	char AdminId[20];
	char AdminName[20];
	supportservice* Sservice;//aggrigation
	report* rep[size];//uni association

public:

	admin()
	{
		strcpy_s(AdminId, "");
		strcpy_s(AdminName, "");
	}
	admin(const char ID[], const char name[], supportservice* ss1)
	{
		strcpy_s(AdminId, ID);
		strcpy_s(AdminName, name);
		Sservice = ss1;
	}

	void report(report* r1, report* r2)
	{
		rep[1] = r1;
		rep[2] = r2;
	}

	void UpdateReport()
	{

	}
	void dispaly()
	{
		cout << " Admin ID : " << AdminId << endl;
		cout << " Adimn name : " << AdminName << endl;

		Sservice->Display();
	}
	void Displayreport()
	{

		rep[1]->displayDitalse();
		rep[2]->displayDitalse();
	}
};
```

Class: Payment, order (Association)

```cpp
//IT21155116 - M.A.D.A.DILSHAN

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

class payment;
class Order;

class payment {
private:
	int paymentId;
	string paymentType;
	double paymentAmount;
	Order* order;
public:
	payment();
	payment(int pId, string pType, double pAmount);
	void addOrder(Order* o);
	void displayPayment();
};

class Order {
private:
	int orderId;
	payment* Payment;
public:
	Order(int OId, payment* P);
	void displayOrders();
};

Order::Order(int OId, payment* P)
{
	orderId = OId;
	Payment = P;
}
void Order::displayOrders()
{
	cout << "orderID : " << orderId << endl;
}

payment::payment(int pId, string pType, double pAmount)
{
	paymentId = pId;
	paymentType = pType;
	paymentAmount = pAmount;
}

void payment::displayPayment()
{
	cout << "payment details  for the order " << endl;
	cout << "id number : " << paymentId << endl;
	cout << "type : " << paymentType << endl;
	cout << "amount : " << paymentAmount << endl << endl;
}
```

Class: Product,discount (Dependency)

```
//IT21156342 G.CPiyumal

/ Dependency relationships

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

class product;
class discount;

class Product
{
private:
	string productName;
	string productID;
	string productDescription;
	float price;
public:
	Product() {}
	Product(string pID, string pname, float pprice)
	{
		productID = pID;
		productName = pname;
		price = pprice;
	}
	void setDescription(string descrip)
	{
		productDescription = descrip;
	}
	float getPrice() {
		return price;
	}
	void displayDetails()
	{
		cout << "Product ID =" << productID << endl;
		cout << "Product name =" << productName << endl;
		cout << "Product Description = " << productDescription << endl;
	}
};
class Discount
{
private:
	double discount;
public:
	void setDiscount(double dcount) {
		discount = dcount;
	}

	void addProduct(int qty, Product* p) {
	}

	void getDiscount() {
		cout << "Discount = " << discount << endl;
	}

};
```

Class: Payment, discount (Association)

```
//IT21154980 Asiff Y N A

#include <iostream>
using namespace std;

class Discount {
private:
	double discount;

public:
	Discount();
	Discount(double pdiscount) {
		discount = pdiscount;
	}
	double getDiscount() {
		return discount;
	}
	void displayDisocunt() {
		cout << "Discount = " << discount << endl;
	}
};

class Payement {
private:
	double amount;
	char
		Discount* dis;

public:
	Payement(double pamount, Discount* d) {
		amount = pamount;
		dis = d;
	}

	void displayTotal() {
		cout << "Total : " << amount << endl;;
	}

	void displaySubTotal() {
		cout << "Sub Total : " << amount - dis->getDiscount();
	}
};
```

Main program

```
int main()
{

            feedback* fb = new feedback("FB123", "Delivery Issues");
    fb->displayFeedback();
    feedback* fs = new feedback("FS123", "Payment issues");
    fs->displayFeedback();

    User* u1 = new User("jone", 40, "New york city", "0112132312");

    RegisterBuyer* rB1 = new RegisterBuyer("David", 28, "London", "0101212123",
1234, "david11@gmail.com", "david123#", fb);

    RegisterSeller* rS1 = new RegisterSeller("Parker", 33, "Mexico", "0156221123",
5678, "parker23@gmail.com", "parker123#", fs);


    u1->displayDetails();
    rB1->displayDetails();
    rS1->displayDetails();

    delete fb;
    delete fs;
    delete u1;
    delete rB1;
    delete rS1;



            supportservice* service1;
            admin* ad;
            report* re1, * re2;

            service1->setDetails("SA1253", "complen");
            ad = new admin("fg123", "Sunile", service1);
            ad->Displayreport();
            re1->setDetails("h475", "active customer");
            //delete ad;


            char ch;
            payment* p1 = new payment(001, "cash", 1000.00);
            payment* p2 = new payment(002, "card", 2000.00);
            payment* p3 = new payment(003, "check", 5000.00);

            p1->displayPayment();
            p2->displayPayment();
            p3->displayPayment();

            cin >> ch;

            char ch;
            Product* P1 = new Product("P001", "Mugs", 150.00);
            P1->setDescription("For camping");
            Discount* D1 = new Discount();
```

```
            D1->setDiscount(12.5);
            D1->addProduct(5, P1);
            D1->getDiscount();
            cin >> ch;

            Discount* d = new Discount(250);
            d->displayDisocunt();

            Payement* p = new Payement(1000, d);
            p->displayTotal();
            p->displaySubTotal();

            delete d;
            delete p;

            return 0;
}
```

Individual Contributions

IT21154812 – Aathif M.T.M

- Created the CRC Card for User, register buyer, register seller and feedback classes.
- Created the class diagram for User, register buyer, register seller and feedback classes.
- Implemented the coding for the User, register buyer, register seller and feedback classes. (Inheritance relation and association relation)

IT21156342 – G.C. Piyumal

- Created the CRC Card for Product and discount classes.
- Created the class diagram for Product and discount classes
- Implemented the coding for the Product and discount classes (Dependency relation)

IT21155116 – M.A.D.A. DILSHAN

- Created the CRC Card for Payment and order classes.
- Created the class diagram for Payment and order classes
- Implemented the coding for the Payment and order classes (Association)

IT21154980 – ASIFF. Y.N.A

- Created the CRC Card for Payment and discount classes.
- Created the class diagram for Payment and discount classes.
- Implemented the coding for the Payment and discount classes. (Association)

IT21155352 – H.R. KAMESH DIVIYANJANA

- Created the CRC Card for admin, support service and reports classes.
- Created the class diagram for admin, support service and reports classes.
- Implemented the coding for the admin, support service and reports classes.
- (Association, composition)