Sri Lanka Institute of Information Technology

Topic : Online Grocery Store and Delivery

Group No : Y1_S2_22_MTR_07

Campus : Matara

Submission Date : 2022/05/16

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.

Group members -

IT number	Name
IT21318498	Didulanga E.A.D
IT21213830	M.G.H Hansana
IT21269448	T.W.O.T Sandeep
IT21318634	Gunawardhana T.G

Exercise 01

Requirements Analysis

❖ This is a website for online Grocery store and delivery. People who need to buy their groceries, who need to search for groceries or else who need to buy or search other goods items can use this website

❖ If user is a guest, he/she must sign in by entering required details. They have to provide first name, last name, username, email address and password. After the sign into the system, it stores profile details.

❖ As a registered customer, he/she can log in to the system by giving email address and password

❖ Customer can view customer profile.

❖ Customer can update or edit customer profile.

❖ Then user can browse for groceries or goods items in certain categories such as fruits, vegetables, meat, fish and shelifish,snacks,bread & bakery and also other health care. In addition to that customer can upload their prescriptions to the system and can make orders for medicines.

❖ Customer can make payments easily through online by their credit and debit cards.

❖ Customer can add feedbacks about the service and also can report complains through the contact page.

❖ User can log in to the system as a shopkeeper. In there he/she have to provide email address, password, and user ID.

❖ Shopkeeper can view the orders and analysis them according to the type ‘With Prescription’ or ‘Without Prescription’ .

❖ Shopkeeper prepares the orders and redirect to the delivery person.

❖ Shopkeeper can update the system with new stores updates.

❖ As an owner, he/she have to log in to the system with the user ID and password.

❖ Owner can view ratings of the customers and can direct them to the admin for make relevant changes.2

❖ Owner can view the stock availability and make orders for the distributors.

❖ As admin, he/she have to log in to the system by providing user ID and password.

❖ Admin can make changes in the category page and the system which the owner directed to him/her.

❖ Admin can generate summary reports such as stock availability, payment details, order details.

Exercise 02

(2) Noun Verb Analysis

❖ This is a website for online Grocery store and delivery. People who need to buy their groceries, who need to search for groceries or else who need to buy or search other goods items can use this website

❖ If user is a guest, he/she must sign in by entering required details. They have to provide first name, last name, username, email address and password. After the sign into the system, it stores profile details.

❖ As a registered customer, he/she can log in to the system by giving email address and password

❖ Customer can view customer profile.

❖ Customer can update or edit customer profile.

❖ Then user can browse for groceries or goods items in certain categories such as fruits,vegetables , meat, fish and shelifish,snacks,bread & bakery and also other health care. In addition to that customer can upload their prescriptions to the system and can make orders for medicines.

❖ Customer can make payments easily through online by their credit and debit cards.

❖ Customer can add feedbacks about the service and also can report complains through the contact page.

❖ User can log in to the system as a shopkeeper. In there he/she have to provide email address, password, and user ID.

❖ Shopkeeper can view the orders and analysis them according to the type 'With Prescription' or 'Without Prescription' .

❖ Shopkeeper prepares the orders and redirect to the delivery person.

❖ Shopkeeper can update the system with new stores updates.

❖ As an owner, he/she have to log in to the system with the user ID and password.

❖ Owner can view ratings of the customers and can direct them to the admin for make relevant changes.4

❖ Owner can view the stock availability and make orders for the distributors.

❖ As admin, he/she have to log in to the system by providing user ID and password.

❖ Admin can make changes in the category page and the system which the owner directed to him/her.

❖ Admin can generate summary reports such as stock availability, payment details, order details.

Nouns

Website	User ID
Prescriptions	Last name
Online grocery store and delivery	With prescription
Orders	Email address
People	Without prescription
Payments	Password
Their	Delivery
Credit cards	person
Medicines	System
Debit cards	Stores
fruits	updates
Feedbacks	Profile
User	details
Service	Ratings
Guest	Registered customer
Complains	Admin
He/she	Customer
Contact page	Stock availability
Details	Customer profile
Shopkeeper	Distributers
First name	Categories
His/her6	Baby
Products	Summery reports
Payment details	vegetables

meat

fish and shelifish

snacks

bread & bakery

health care

Order details

Exercise 03

CRC Cards

Customer	
Responsibilities	**Collaboration**
Search groceries	Category
Store customer details	
View customer profiles	
Make orders	Order, payment
Update customer details	
Give feedbacks	Feedback

Shopkeeper	
Responsibilities	**Collaboration**
Update goods lists	Category
Store profile details	
View orders	Order, payment
Edit profile	
Update the profile	

owner	
Responsibilities	**Collaboration**
Store owner details	
View ratings	Feedback
View stock availability	Category
Decide the daily item give discounts	
Decide the item prices	

Admin	
Responsibilities	**Collaboration**
Store admin details	
Add goods/items	Category
Generate reports	Report
Update the daily item give discounts	Category
Update profile details	

Payment	
Responsibilities	**Collaboration**
Display payments	Reports
Store payments	Customer
View payments	Shopkeeper
Validate payment	

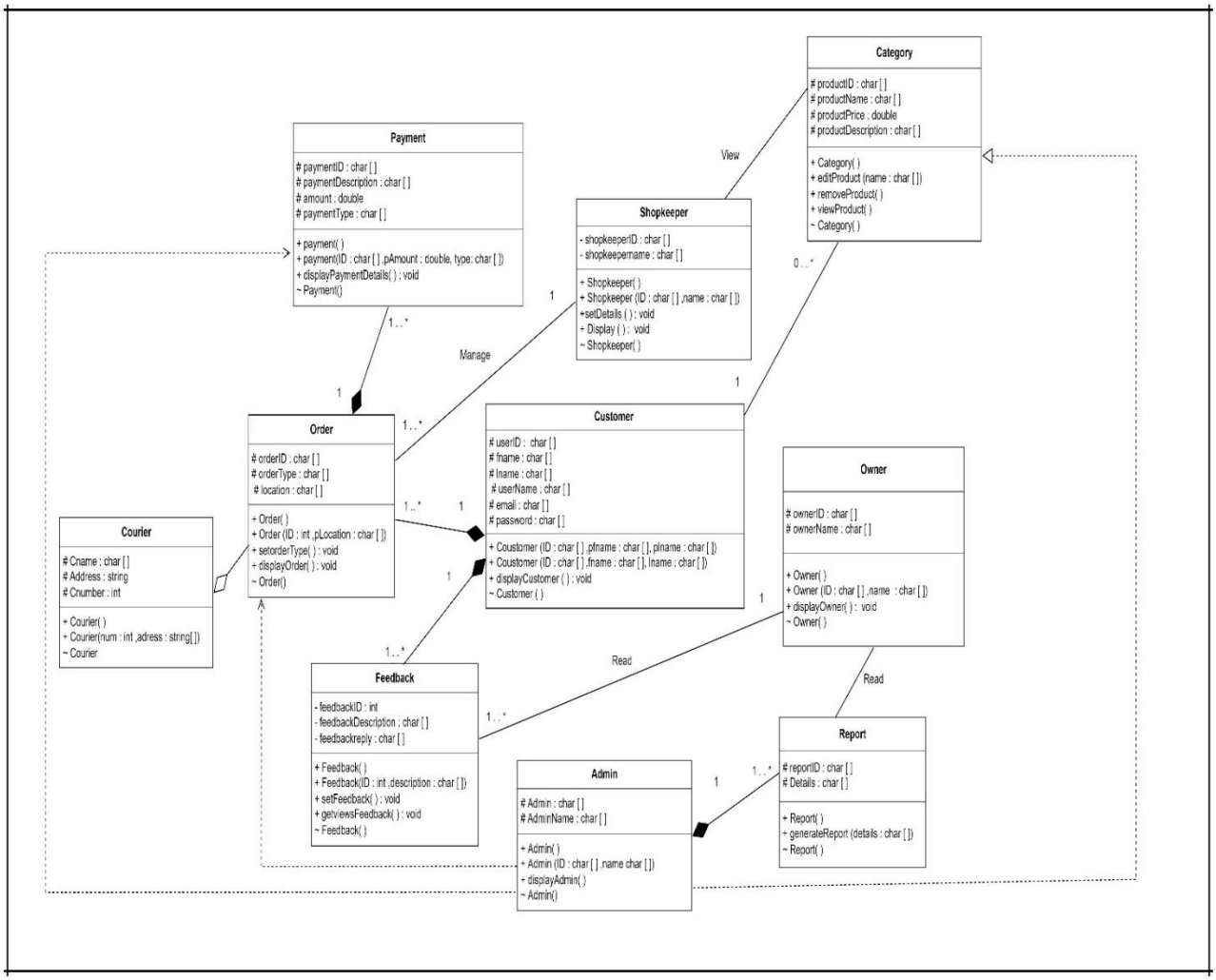
Category	
Responsibilities	**Collaboration**
Store goods/ items	Admin
Update goods/ items	Shopkeeper
View goods/ items	Customer/ owner

Feedback	
Responsibilities	**Collaboration**
View feedbacks	Owner
Store feedbacks	Customer

Report	
Responsibilities	**Collaboration**
List of stock availability	Category
List of payments	Payment
List of orders	Order
List of delivery	

Order	
Responsibilities	**Collaboration**
Store orders	Customer
View orders	Shopkeeper, Report
Deliver order	courier
Conform order	Payment

Courier	
Responsibilities	**Collaboration**
Deliver the goods relevant place	


Category
productID : char [ ]
productName : char [ ]
productPrice : double
productDescription : char [ ]
+ Category( )
+ editProduct (name : char [ ])
+ removeProduct( )
+ viewProduct( )
~ Category( )
Payment
paymentID : char [ ]
paymentDescription : char [ ]
amount : double
paymentType : char [ ]
+ payment( )
+ payment(ID : char [ ] ,pAmount : double, type: char [ ])
+ displayPaymentDetails( ) : void
~ Payment()
View
Shopkeeper
- shopkeeperID : char [ ]
- shopkeepername : char [ ]
+ Shopkeeper( )
+ Shopkeeper (ID : char [ ] ,name : char [ ])
+setDetails ( ) : void
+ Display ( ) : void
~ Shopkeeper( )
0 . . *
1
1 . . *
Manage
1
1 . . *
Order
orderID : char [ ]
orderType : char [ ]
location : char [ ]
+ Order( )
+ Order (ID : int ,pLocation : char [ ])
+ setorderType( ) : void
+ displayOrder( ) : void
~ Order()
Customer
userID : char [ ]
fname : char [ ]
lname : char [ ]
userName : char [ ]
email : char [ ]
password : char [ ]
+ Coustomer (ID : char [ ] ,pfname : char [ ], plname : char [ ])
+ Coustomer (ID : char [ ] ,fname : char [ ], lname : char [ ])
+ displayCustomer ( ) : void
~ Customer ( )
1
Owner
ownerID : char [ ]
ownerName : char [ ]
+ Owner( )
+ Owner (ID : char [ ] ,name : char [ ])
+ displayOwner( ) : void
~ Owner( )
Courier
Cname : char [ ]
Address : string
Cnumber : int
+ Courier( )
+ Courier(num : int ,adress : string[ ])
~ Courier
1 . . *
1
Feedback
- feedbackID : int
- feedbackDescription : char [ ]
- feedbackreply : char [ ]
+ Feedback( )
+ Feedback(ID : int ,description : char [ ])
+ setFeedback( ) : void
+ getviewsFeedback( ) : void
~ Feedback( )
Read
1 . . *
Read
Report
reportID : char [ ]
Details : char [ ]
+ Report( )
+ generateReport (details : char [ ])
~ Report( )
1 . . *
1
Admin
Admin : char [ ]
AdminName : char [ ]
+ Admin( )
+ Admin (ID : char [ ] ,name char [ ])
+ displayAdmin( )
~ Admin()

Exercise 4

```
//header file
#pragma once
class Customer
{
	private:
	char UserID[5];
	char Fname[15];
	char Lname[15];
	char Username[10];
	char Email[30];
	char password[10];
	Feedback* feedbacks[5];
	Order* orders[4];
	Category* categories[2];
	public:
	Customer();
	Customer (char ID[5], const char fname[15], const char lname[15]);
	void displaycustomer() {
	feedbacks[1] = new Feedback();
	o1[1] = new Order();
	c1[1] = new Category();
}
~Customer();
};

class Shopkeeper
{
	private:
```

```
        char shopkeeperID[5];
        char shopkeepername[20];
        Order* o1[10];
        public:
        Shopkeeper();
        Shopkeeper(char ID[], const char name[]);
        void Display();
        ~Shopkeeper();
};

class Owner
{

        private:
        char ownerID[5];
        char ownername[20];
        Feedback* f1[5];
        public:
        Owner();
        Owner(char ID[5], const char name[20]);
        void Displayowner();
        ~Owner();
};

class Admin
{
        private:
        char adminID[5];
        char AdminName[25];
```

```
        Report* r1[4];
        public:
        Admin();
        Admin(char ID[], const char name[], Category*c1, Order*o1, Payment*p1);
        void displayAdmin() {
        r1[1] = new Report();
}
~Admin();
};

class Payment
{
        private:
        char paymentID[5];
        char paymentdescription[25];
        double Amount;
        char paymentType[5];
        public:
        Payment();
        Payment(const char ID[], const char description[], double amount, char type[] );
        void displaypaymentDetails();
        ~Payment();
};

class Category

{
        char ProductID[5];
        char Productname[20];
```

```
        double Productprice;
        char Productdescription[20];
        Customer* c1;
        public:
        Category();
        void editproduct(const char name[]);
        void removeproduct();
        void viewproduct();
        ~Category();
};

class Report
{

        private:
        char reportID[5];
        char Details[100];
        Owner* o1[2];
        public:
        Report();
        int getgenerateReport();
        ~Report();
};

class Order
{
        private:
        char orderID[5];
        char orderType[5];
```

```
        char Location[20];
        Payment* p1[5];
        public:
        Order();
        Order(char ID[], char location[]);
        void setordertype(char type);
        void Displayorder() {
        p1[1] = new Payment();
}
~Order();
};

class Feedback
{
        private:
        int feedbackID;
        char feedbackdescription[45];
        int feedbackview;
        Owner* o1[];
        public:
        Feedback();
        Feedback(int id, const char description[]);
        int getviewfeedback();
        ~Feedback();
}
```

```
//.cpp file
#include "ooc.h"
#include <cstring>
#include <iostream>
using namespace std;
//customers
Customer::Customer()
{
            strcpy(UserID,"");
    strcpy(Fname,"");
    strcpy(Lname,"");
   strcpy(Username,"");
   strcpy(Email,"");
   strcpy(password,"");
}

Customer::Customer(char ID[5], const char fname[15], const char lname[15])
{

            strcpy(UserID, ID);
            strcpy(Fname, fname);
            strcpy(Lname, lname);
}

void Customer::displaycustomer()
{
            feedbacks[5] = new Feedback();
            orders[5] = new Order();
            categories[5] = new Category();
```

```
}
Customer::~Customer()
{
        cout << "User id :" << UserID << "Deleted " << endl;
}

//Composition feedbacks
char feedbackdescription[100];
 cout<<"Description :";
 cin>>feedbackdescription[100];
char feedbackreply[100];
 cout<<"Reply :";
 cin>>feedbackreply[100];
feedbacks[5]= new Feedback(feedbackdescription[100],feedbackreply[100] );

//Composition orders
char orderID[5];
 cout<<"ID:";
 cin>>orderID;
char orderType[5];
 cout<<"type :";
 cin>>orderType;
char Location[20];
 cout<<"location :";
 cin>>Location;
orders[5]= new Order(orderID,orderType,Location[20]);

//Shopkeeper
Shopkeeper::Shopkeeper()
```

```
{
	strcpy_s(shopkeeperID, "");
	strcpy_s(shopkeepername, "");
}
Shopkeeper::Shopkeeper(char ID[], const char name[])
{

	strcpy_s(shopkeeperID, ID);
	strcpy_s(shopkeepername, name);
}
void Shopkeeper::Display()
{
	cout << "Shopkeeper id : " << shopkeeperID << endl;
	cout << "Shopkeeper name : " << shopkeepername << endl;
}
Shopkeeper::~Shopkeeper()
{
	cout << "shopkeeper id :" << shopkeeperID << "Deleted " << endl;
}



//Owner
Owner::Owner()
{
	strcpy(ownerID, "");
	strcpy(ownername, "");
}
Owner::Owner(char ID[5], const char name[20])
{
```

```
	strcpy(ownerID, ID);
	strcpy(ownername, name);
}
void Owner::Displayowner()
{
	cout << "Owner id : " << ownerID << endl;
	cout << "Ownername : " << ownername << endl;
}
Owner::~Owner()
{
	cout << "Owner id :" << ownerID << endl;
}



//Admin
Admin::Admin()
{
	strcpy_s(adminID, "");
	strcpy_s(AdminName, "");
}
Admin::Admin(char ID[], const char name[], Category * c1, Order * o1, Payment * p1)
{
	strcpy_s(adminID, ID);
	strcpy_s(AdminName, name);
}
void Admin:: displayAdmin()
{
	r1[1] = new Report();
}
```

```
Admin::~Admin()
{
	cout << "Admin id :" << adminID << endl;
};



//Payment
Payment::Payment()
{
	strcpy_s(paymentID, "");
	strcpy_s(paymentdescription, "");
	Amount = 0;
	strcpy_s(paymentType, "");
}
Payment::Payment(const char ID[], const char description[], double amount, char type[])
{
	strcpy_s(paymentID, ID);
	strcpy_s(paymentdescription, description);
	Amount = amount;
	strcpy_s(paymentType, type);
}
void Payment::displaypaymentDetails()
{
	cout << "Payment id:" << paymentID << endl;
	cout << "Payment description : " << paymentdescription << endl;
	cout << "Payment amount " << Amount << endl;
	cout << "Paymnet type :" << paymentType << endl;
}
Payment::~Payment() {
```

```
cout << "Payment id :" << paymentID << "Deleted " << endl;
};



//category
Category::Category()
{
	strcpy_s(ProductID, "");
	strcpy_s(Productname, "");
	Productprice = 0;
	strcpy_s(Productdescription, "");
}
void Category::editproduct(const char name[])
{
	strcpy_s(Productname, name);
}
void Category::removeproduct()
{
	cout << "Product Id :" << ProductID << endl;
}
void Category::viewproduct()
{
	cout << "Product id :" << ProductID << endl;
}
Category::~Category()
{
	cout << "product Id :" << ProductID << "Deleted" << endl;
};
```

```
//report
Report::Report()
{
	strcpy_s(reportID, "");
	strcpy_s(Details, "");
}
int Report:: getgenerateReport()
{
	return Details[75];
}
Report::~Report()
{
	cout << "Report id :" << reportID << "Deleted " << endl;
};

//order
Order::Order()
{
	strcpy_s(orderID, "");
	strcpy_s(orderType, "");
	strcpy_s(Location, "");
}
Order::Order(char ID[], char location[])
{
	strcpy_s(orderID, ID);
	strcpy_s(Location, location);
}
void Order::setordertype(char type)
```

```
{
        cout << "Order Type:" << orderType << endl;
        cin >> orderType;
};
void Order:: Displayorder()
{
        p1[1] = new Payment();
}
Order::~Order()
{
        cout << "Order Id :" << orderID << "deleted" << endl;
};
```

Main Program

```
#include "ooc.h"
#include <iostream>
#include "ooc.cpp"

using namespace std;
int main()
{
	Customer *c1;
	c1->Displayowner();
	Shopkeeper *s1;
	s1->Display();
	Owner *o1;
	o1->Displayowner();
	Admin *a1;
	a1->displayAdmin();
	Payment* p1;
	p1->displaypaymentDetails();
	Order* order1;
	order1->Displayorder();
	Report* r1;
	r1->getgenerateReport();
	Category* items;
	items->viewproduct();
	Feedback* feed;
	feed->getviewfeedback();
	return 0 ;
}
```