Topic : Online Grocery Store and Delivery

Group no :Group_08

Campus : Kandy

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**
IT21355264	Wijesinghe W.M.H.D. K	0702616609
IT21238758	Abeyrathne R.M.S.N	0764512630
IT21278730	Wickramanayaka D.N. N	0772226936
IT21297090	Jayamini Y.C.M	0773889136
IT21367458	Siyambalapitiya H. N	0778871511

Contents

System Requirements

- Our system is open 24 hours a day, 7 days a week, 365 days a year
- A guest user logs into the system for data analysis
- Registration is mandatory for the guest user to access the data in this system
- Customers can delete or update the status of the login details (username, password)
- System generates the unique id for each customer
- The following information is required for such registration (name, NIC, Email, contact details, Age)
- When the supplier delivers products to the system, the customer purchases them
- The supplier must deposit some amount in the system before selling the products
- The inventory manager stores all information about the products that the supplier has entered the system
- Products have product id, product name, product type, product brand
- Cart search and adds products selected by the customer
- To place an order customer must give order amount, order type
- After adding to the cart, the customer enters the delivery details
- Customers must enter their payment details like payment methods, card details
- There are different payment methods available, and the customer can choose one of them and make the payment.
- After the payment, payment id is generated and then customer can place delivery.
- When the customer submits the relevant data to the computer when making an card payment, the verification of the data entered and the delivery information is sent to the customer via SMS.
- Freight forwarders receive feedback from customers.
- Feedbacks are displayed in a system.
- Analyzes update and enter the delivery details to the system.

(Noun & Verb Analysis)

(NOUNS)

- Our system is open 24 hours a day, 7 days a week, 365 days a year
- An guest user logs into the system for data analysis
- Registration is mandatory for the guest user to access the data in this system
- Customers can delete or update the status of the login details (username, password)
- System generates the unique id for each customer
- The following information is required for such registration (name, NIC, Email, contact details, Age)
- When the supplier delivers products to the system, the customer purchases them
- The supplier must deposit some amount in the system before selling the products
- The inventory manager stores all information about the products that the supplier has entered the system
- Products have product id, product name , product type, product brand
- Cart search and adds products selected by the customer
- To place an order customer must give order amount, order type
- After adding to the cart, the customer enters the delivery details
- Customers must enter their payment details like payment methods, card details
- There are different payment methods available, and the customer can choose one of them and make the payment.
- After the payment, payment id is generated and then customer can place delivery.
- When the customer submits the relevant data to the computer when making an card payment, the verification of the data entered and the delivery information is sent to the customer via SMS.
- Freight forwarders receive feedback from customers.
- Feedbacks are displayed in a system.
- Analyzes update and enter the delivery details to the system.

Identified Classes

- Guest User
- Customer
- Supplier
- inventory manager
- products
- payment
- Delivery
- Feedback
- Freight forwarders
- Order

Reasons for rejecting other nouns

- Redundant: Customers, payment, guest user, delivery manager, Freight forwarders, products
- An Event or an operation: search and adds items
- Outside scope of system: System, Bank, trusted resources
- Meta-language: them
- An attribute: login details (username, password), registration (name, NIC, Email, contact details, Age), Products (product id, product name, product type, product brand), payment(payment id , payment type, card details, payment methods) , selling details, delivery details.

Noun & Verb Analysis

(VERBS)

- Our system is open 24 hours a day, 7 days a week, 365 days a year
- A guest user logs into the system for data analysis
- Registration is mandatory for the guest user to access the data in this system
- Customers can delete or update the status of the login details (username, password)
- System generates the unique id for each customer
- The following information is required for such registration (name, NIC, Email, contact details, Age)
- When the supplier delivers products to the system, the customer purchases them
- The supplier must deposit some amount in the system before selling the products
- The inventory manager stores all information about the products that the supplier has entered the system
- Products have product id, product name, product type, product brand
- Cart search and adds products selected by the customer
- To place an order customer must give order amount, order type
- After adding to the cart, the customer enters the delivery details
- Customers must enter their payment details like payment methods, card details
- There are different payment methods available, and the customer can choose one of them and make the payment.
- After the payment, payment id is generated and then customer can place delivery.
- When the customer submits the relevant data to the computer when making a card payment, the verification of the data entered, and the delivery information is sent to the customer via SMS.
- Freight forwarders receive feedback from customers.
- Feedbacks are displayed in a system.
- Analyzes update and enter the delivery details to the system.

Methods

- Guest User - Access the system to view the system

- Customers - Log in to the system by using login credentials

  Delete details

  Update Customers details
- Supplier - Deposit Advance details

  Enter the products into system

- inventory manager - Stores products information
- Products - Generate product id,

  Add Products details,

  Delete Products details

  Update Products details
- Payment - Add Payment details

  Generate Payment id
- Delivery - Generate delivery id

  Enter the delivery details

  Delete Delivery details

  Update Delivery details
- Freight forwarder - Get feedback from customer
- Feedback - Display the feedback details
- Order - Generate Order id

  Add order details to the system

  Adds products to the system

CRC Cards

<table>
<tr><th colspan="2">Guest User</th></tr>
<tr><th><b>Responsibility</b></th><th><b>Collaborators</b></th></tr>
<tr><td>Register to the system</td><td></td></tr>
<tr><td>Allow to view the product</td><td>Product</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Customer</th></tr>
<tr><th><b>Responsibility</b></th><th><b>Collaborators</b></th></tr>
<tr><td>Log in to the system by using login credentials</td><td>Order</td></tr>
<tr><td>Delete Customers details</td><td>Freight forwarders</td></tr>
<tr><td>update Customers details</td><td>payment</td></tr>
<tr><td></td><td></td></tr>
</table>

Supplier	
Responsibility	**Collaborators**
Deposit Advance	inventory manager
Enter the products	Products

Feedback	
Responsibility	**Collaborators**
Display feedbacks	Customer, Freight forwarder

Inventory manager	
Responsibility	**Collaborators**
stores system information	Products, payment

Products	
Responsibility	**Collaborators**
Generate product id	Supplier
Add products details	Products
Delete products details	
Update products details	

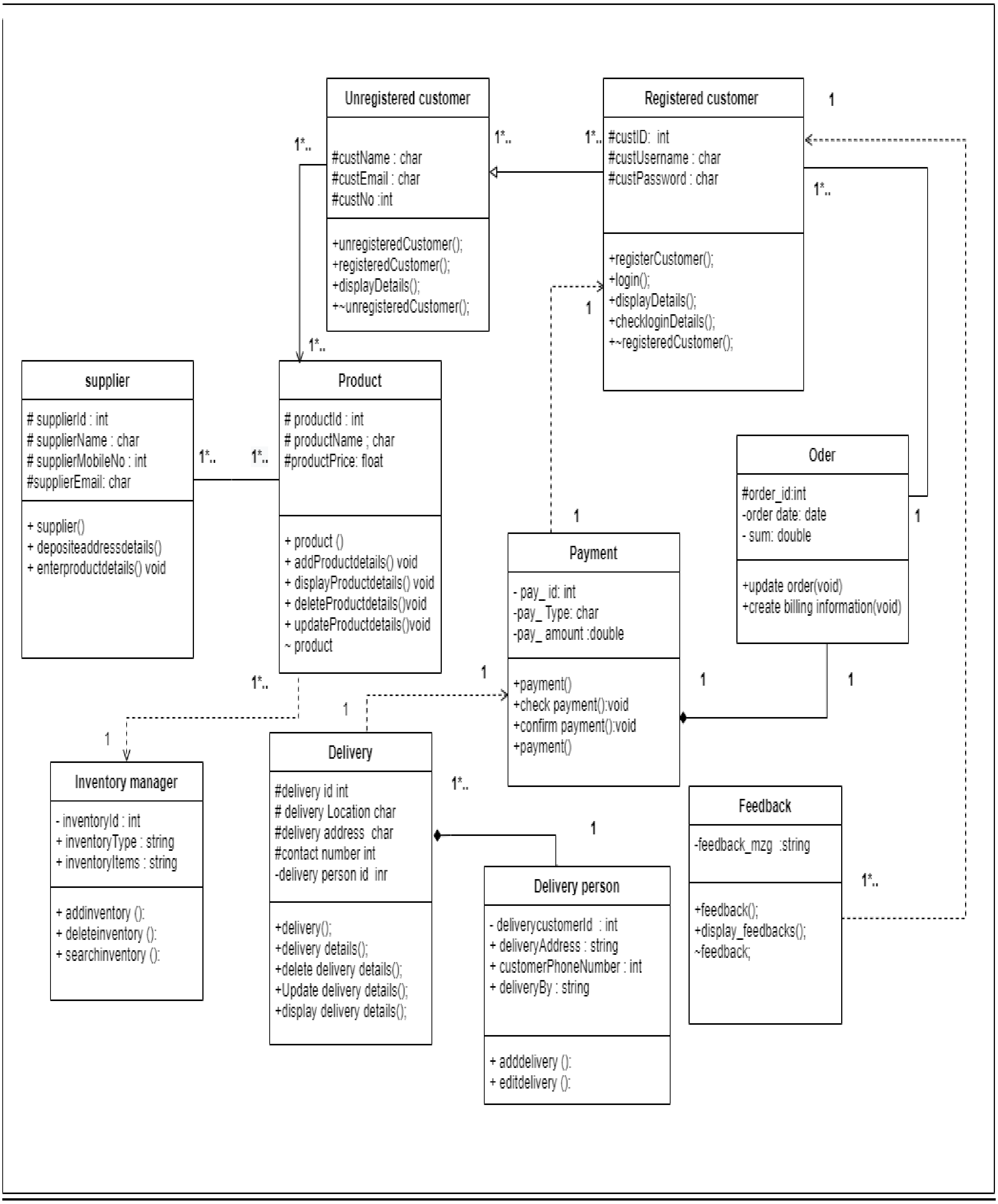
Payment	
Responsibility	**Collaborators**
Add Payment details	Products
Generate Payment id	Customers
	Delivery manager

Delivery	
Responsibility	**Collaborators**
Enter the delivery details	Payment
Delete delivery details	
Update delivery details	Freight forwarders

Freight forwarders	
Responsibility	**Collaborators**
Get feedback from customer	Customers

order	
Responsibility	**Collaborators**
Generate Order id	Customers
Add order details	Payment
Search products	Products
Adds products	Customers

Class Diagram (UML Notation)



Class Header Files

Guest User

UnregisteredCustomer.h

```
#pragma once
#include "Product.h"
#define size 2

class UnregisterdCustomer

{

protected:

    int custNIC;

    char custName[50];

    char custAddress[30];

    char custEmail[50];

    int custMobile;


public:

    UnregisterdCustomer();

    UnregisterdCustomer(int unr_NIC,const char unr_name[],const char
unr_address[],const char unr_email[],int unr_mobi);

    void displayDetails();

    ~UnregisterdCustomer();

};
```

Customer

RegisteredCustomer.h

```
#include "UnregisteredCustomer.h"


class RegisteredCustomer:public UnregisterdCustomer

{

protected:

    int custID;

    char userName[50];

    char password[30];


public:

    RegisteredCustomer();

    RegisteredCustomer(int rcu_ID,const char ru_Name[],const char rc_password[],int unr_NIC,const char unr_name[],const char unr_address[],const char unr_email[],int unr_mobi);

    void displayDetails();

    void login();

    void checkLoginDetails();

    ~RegisteredCustomer();

};
```

Supplier

Supplier.h

```
#include "Inventorymanager.h"

class supplier

{

protected:

            int supplierID;

            char supplierName[50];

            int supplierMobilenumber;

            char supplierEmail[50];

   Inventorymanager *inv;



public:

            supplier();

            supplier(int tsupplierID, char tsupplierName[], int tsupplierMobilenumber, char tsupplierEmail[],Inventorymanager *inv);

            void  depositaddressdetails();

            void enterproductdetails();

            ~supplier();

};
```

inventory manager

Inventorymanager.h

```
#include <string>
class Inventorymanager
{
private:
        int inventoryId;
        char inventorytype[50];
        char inventoryItems[50];

public:
   Inventorymanager();
   Inventorymanager(int i_id,const char in_type[],const char in_item[]);
                void addInvetory();
         void deleteInvetory();
                void searchInventory();
   ~Inventorymanager();

};
```

products

products.h

```
#include "supplier.h"

class Product

{

protected:

        int productID;

        char productName[50];

        double price;

public:

        Product();

        Product(int tproductID,const char tproductName[],double tprice);

        void displayProductDetails(int rproductID,const char rproductName[],double rprice);

        void addproductdetails();

  void updateproductdetails();

        ~Product();

};
```

Payment

Payment.h

```
#include"order.h"

#include<cstring>

#define SIZE 3

class payment

{

private:

  int payID;

  char payType[50];

  double payamount;

  order*odr[SIZE];

public:

   payment();

   payment(int PID, const char p_Type[],double pay_amount,order *coder);

   void checkpayment();

   void confirmpayment();

   void displaypaymentDetails();

   ~payment();

}
```

```
};
```

Delivery

Delivery.h

```
#pragma once

#include "payment.h"

#include "Deliveryperson.h"

#define SIZE 2

class delivery

{

protected:

		int delivery_id;

		char delivery_address[50];

		int customer_contact_number;

private:

		   int delivary_person_id;

		payment *pay[SIZE];

		Deliveryperson *dp[SIZE];

public:

    delivery();

    delivery(int dp1,int dp2);

		void deliveryDetails( const char d_add[], int c_con, int dp_id,payment *cpay,Deliveryperson
*delip);

		void deleteDeliveryDetails();

		void  UpdateDeliveryDetails();

		void  displayDeliveryDetails();

		~delivery();



};
```

Feedback

Feedback.h

```
#include "Deliveryperson.h"

#include "RegisteredCustomer.h"

#include <cstring>

class feedback

{

private:

        char feedback_mzg[100];

        RegisteredCustomer *customer;

        Deliveryperson *dp;

public:

        feedback();

        feedback(const char  feed_mzg[], RegisteredCustomer *customer, Deliveryperson *dp);

        void display_feedback();

        ~feedback();

};
```

Freight forwarders

Deliveryperson.h

```
class Deliveryperson

{

private:

     int Deliverypersonid;

    int deliveryCustomerId;

   char deliveryAddress[50];

   int customerPhoneNumber;

   char deliveryBy[50];

public:

    Deliveryperson();

    Deliveryperson (int p_id)

{

      Deliverypersonid=p_id;

}

    Deliveryperson(int c_id,const char d_address[],int c_phonenumber, const char
d_by[]);

      void addDelivery();

                  void editDelivery();

};
```

Order

Oder.h

```
#include"RegisteredCustomer.h"

#define SIZE2


class order

{

protected:

   int orderID;

private:

   char orderDate[20];

   int orderQty;

   int  orderNo;


  RegisteredCustomer *customer;

  public:

   order();

   order(int o_ID,int oNo,int qty,const char o_date[]);

   void update_order();

   void remove_order();

   double getTotal();

};
```

Class Cpp Files

Guest User

UnregisteredCustomer.cpp

```
#include "UnregisteredCustomer.h"
#include <cstring>
#define size 2
UnregisterdCustomer::UnregisterdCustomer()
{
custNIC=0;
    strcpy(custName,"");
    strcpy(custAddress,"");
    strcpy(custEmail,"");
    custMobile=0;
}
UnregisterdCustomer::UnregisterdCustomer(int unr_NIC,const char unr_name[],const char unr_address[],const char unr_email[],int unr_mobi)
{
    custNIC=unr_NIC;
    strcpy(custName,unr_name);
    strcpy(custAddress,unr_address);
    strcpy(custEmail,unr_email);
    custMobile=unr_mobi;
}
void UnregisterdCustomer::displayDetails()
{

}
UnregisterdCustomer::~UnregisterdCustomer()
{
    //Destructor
}
```

Customer

RegisteredCustomer.cpp

```
#include "RegisteredCustomer.h"
#include "UnregisteredCustomer.h"
#include <cstring>

RegisteredCustomer::RegisteredCustomer()
{
    custID=0;
    strcpy(userName,"");
    strcpy(password,"");
}
RegisteredCustomer::RegisteredCustomer(int rcu_ID,const char ru_Name[],const char
rc_password[],int unr_NIC,const char unr_name[],const char unr_address[],const char
unr_email[],int unr_mobi)
{
    custID=rcu_ID;
    strcpy(userName,ru_Name);
    strcpy(password,rc_password);
}
void RegisteredCustomer::displayDetails()
{

}
void RegisteredCustomer::login()
{

}
void RegisteredCustomer::checkLoginDetails()
{

}
RegisteredCustomer::~RegisteredCustomer()
{
    //Destructor
}
```

Supplier

supplier.cpp

```
#include "supplier.h"
#include <cstring>

#define SIZE 2

supplier::supplier()
{
    supplierID = 0;
    strcpy(supplierName, "");
    supplierMobilenumber = 0;
    strcpy (supplierEmail, "");
}

supplier::supplier(int tsupplierID, char tsupplierName[], int tsupplierMobilenumber, char
tsupplierEmail[],Inventorymanager *inv)
{
    supplierID = tsupplierID;
    strcpy(supplierName, tsupplierName);
    supplierMobilenumber = tsupplierMobilenumber;
    strcpy(supplierEmail, tsupplierEmail);
}
void supplier::depositaddressdetails()
{
}
void supplier::enterproductdetails()
{

}
supplier::~supplier()
{

}
```

inventory manager

Inventorymanager.cpp

```
#include "Inventorymanager.h"
#include <cstring>

Inventorymanager::Inventorymanager()
{
     inventoryId=0;
       strcpy(inventorytype,"");
     strcpy(inventoryItems,"");
}
Inventorymanager::Inventorymanager(int i_id,const char in_type[],const char in_item[])
{
    inventoryId=0;
       strcpy(inventorytype,in_type);
    strcpy(inventoryItems,in_item);
}
void Inventorymanager::addInvetory()
{
}
void Inventorymanager::deleteInvetory()
{
}
void Inventorymanager::searchInventory()
{
}
 Inventorymanager::~Inventorymanager()
{


}
```

products

Product.cpp

```
#include "Product.h"
#include <cstring>


Product::Product()
{
    productID=0;
    strcpy(productName,"");
    price=0.0;
}
Product::Product(int rproductID,const char rproductName[],double rprice)
{
    productID=rproductID;
    strcpy(productName,rproductName);
    price=rprice;
}
void Product::displayProductDetails(int rproductID,const char rproductName[],double
rprice)
{
}
void Product::addproductdetails()
{
}
void Product::updateproductdetails()
{
}
Product::~Product()
{
    //Destructor
}
```

Payment

payment.cpp

```
#include<iostream>
#include"payment.h"
#include<cstring>

payment::payment()
{
  payID =0;
  strcpy(payType,"");
  payamount = 0;
}
payment::payment(int PID, const char p_Type[],double pay_amount,order *coder)
{
  payID =PID;
  strcpy(payType,p_Type);
  payamount = pay_amount;

}
void payment::checkpayment()
{

}
void payment::confirmpayment()
{

}
void payment::displaypaymentDetails()
{

}
payment :: ~payment()
{

}
```

Delivery

delivery.cpp

```
#include "delivery.h"

#include<cstring>


#define SIZE 2


delivery::delivery()
{
      delivery_id = 0;
      strcpy(delivery_address, "");
      customer_contact_number = 0;
      delivary_person_id = 0;
}
 delivery::delivery(int dp1,int dp2)
         {
           dp[0] = new Deliveryperson(dp1);
           dp[1] = new Deliveryperson(dp2);
         }
void delivery::deliveryDetails( const char d_add[], int c_con, int dp_id,payment
*cpay,Deliveryperson *delip)
{
      strcpy(delivery_address, d_add);
      customer_contact_number = c_con;
      delivary_person_id = dp_id;
}
void delivery::deleteDeliveryDetails()
{
}
void  delivery::UpdateDeliveryDetails()
{
}
void  delivery::displayDeliveryDetails()
{


}
delivery::~delivery()
{  //distuctor
}
```

Feedback

Feedback.cpp

```
#include "feedback.h"
#include <cstring>

feedback::feedback()
{
  strcpy(feedback_mzg,"");
}
feedback::feedback(const char feed_mzg[], RegisteredCustomer
*customer,Deliveryperson *dp)
{
  strcpy(feedback_mzg, "feed_mzg");
}
void feedback::display_feedback()
{

}
feedback::~feedback()
{

}
```

Freight forwarders

Deliveryperson.cpp

```
#include "Deliveryperson.h"
#include<cstring>

Deliveryperson::Deliveryperson()
{
    deliveryCustomerId=0;
                strcpy(deliveryAddress,"");
                customerPhoneNumber=0;
       strcpy(deliveryBy,"");
}
Deliveryperson:: Deliveryperson(int p_id,const char
     d_address[],int c_phonenumber, const char d_by[])
{

}
void Deliveryperson:: addDelivery()
{

}
void Deliveryperson:: editDelivery()
{

}
```

Order

Oder.cpp

```
#include"order.h"
#include <cstring>

order::order()
{
  orderID=0;
  orderNo=0;
  orderQty=0;
  strcpy(orderDate,"");
}
order:: order(int o_ID,int oNo,int qty,const char o_date[])
{
  orderID=
  orderNo =oNo;
  orderQty = qty;
}
void order::update_order()
{

}
void order::remove_order()
{

}
double order:: getTotal()
{

}
```

Main program

Main.cpp

```
#include "Inventorymanager.h"
#include "RegisteredCustomer.h"
#include "UnregisteredCustomer.h"
#include "Deliveryperson.h"
#include "delivery.h"
#include "feedback.h"
#include "supplier.h"
#include "Product.h"
#include"payment.h"
#include"order.h"


int main()
{
   Inventorymanager *inv = new Inventorymanager();//object-Inventorymanager
   delivery *deliv = new delivery();//object-delivery
   payment *pay = new payment();//object-payment
   Deliveryperson *dp = new Deliveryperson();//object-delivery person
   feedback *feed = new feedback();//object-feedback
   RegisteredCustomer *customer = new RegisteredCustomer();//object-RegisteredCustomer
   order*odr =new order();//object-order
   Product*pro =new Product();//object-Product


   deliv->UpdateDeliveryDetails();
   deliv->displayDeliveryDetails();

   dp->addDelivery();
   dp->editDelivery();

   feed->display_feedback();

   odr->update_order();

   inv->addInvetory();

   pay-> displaypaymentDetails();

   pro->addproductdetails();
   pro->updateproductdetails();
```

```
    customer->displayDetails();
    customer-> login();


    delete deliv;
    delete customer;
    delete pro;
    delete pro;

    return 0;

}
```