Topic : Online Dry Cleaning & Laundry Service

Group no : KGL_06

Campus : Kurunegala

Submission Date : 20/05/0222

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
IT21212772	Koggala D.B.K.D	0763234926
IT21240324	Kulasuriyalage H.K.K	0772711204
IT21295942	Konara K. M. N. S	0703191719
IT21005008	Nanayakkara S.N.C	0717027646

Introduction

- With the current lifestyle, people have become very busy. Today, with the advancement of technology, it provides solutions to the problems faced by those people. Nowadays it has become difficult for people to even wash their clothes. This project is designed to aid the online laundry and dry cleaning system for busy people that you can clean your every kind of clothes through the online platform. The system helps to the users to find cleaning items, their price and payment method. User can simply login to the system and they can order their cleaning things and they can delivered their items to the door step. This system provides you to make payments online. Customers can purchase the laundry items at any time and made it easy for them via credit and debit. The system provides purchased history for customers if they want to look back on what they have done. This function showed in the User profile if you are a registered customer. This system also provides higher customer care service. If you have any inquiries, you can contact the administration department using feedback forms or the Hotline.

System Requirements

• Unregistered users should be able to register to the system and overview the system. They should fill Name, Address, Email, Phone number, Username, password.

•After registering to the system, the User ID should be generated and displayed.

• Registered users should be able to log in to the system by entering username and password.

• Both users should be able to search items.

• Customer should be able to add the items to the cart.

• And also, customer can delete some items from the cart.

• Customer should be able to place the order.

• Customer should be able to make payments online through credit or debit card and also customer can make the payment on delivery.

• After payment, a payment ID should be generated.

• Customer should be able to give their feedback about the website.

• After giving feedback, the feedback ID should be generated.

• Customer should be able to edit profile.

• System should be able to validate customer details.

• System should be able to store user details in the database.

• Admin should be able to validate payment information.

• System should be able to record the transaction.

• In our website we have some categories, and their data should be in the database.

• Admin should be able to log in to the system.

• Admin should be able to update the website.

• Admin should be able to access the database.

1.Unregisterd User

Log in to the system

Search Items

2.Customer

Log in to the system

Search items

Add items

Delete items

Place the order

Make payments

Send feedbacks

Edit the profile

3. Admin

Log in to the system

Update the website

Access database

Validate payment information

4.Payment

Display payment details

Generate payment ID

Validate

5.Items

Add to cart

Delete from cart

6.Feedback

Display feedback details

Generate feedback ID

7. Reject system because it is outside of the scope

CRC CARDS

1)

Class Name: Registered User	
Responsibility	**Collaborators**
Can view their profile and edit the profile	System
Make payment	Payment
Can select items from store	System
Get item description	Product

2)

Class Name: User	
Responsibility	**Collaborators**
Can view store items	
Can get item description	item

3)

Class Name: Unregistered User	
Responsibility	**Collaborators**
Register to the system	System
Create the profile	System

4)

Class Name: Admin	
Responsibility	**Collaborators**
Update the site	
Manage the site	

5)

Class Name: Cashier	
Responsibility	**Collaborators**
Manage payment	payment
Manage the orders	order
Generate report	
Mange delivery	

6)

Feedback	
Responsibility	**Collaborations**
Generate feedback ID	
Display feedback details	
Store feedback details	

7)

Class Name: item	
Responsibility	**Collaborators**
Generate item ID	
Add item	Admin
Update item	Admin
Display details	

8)

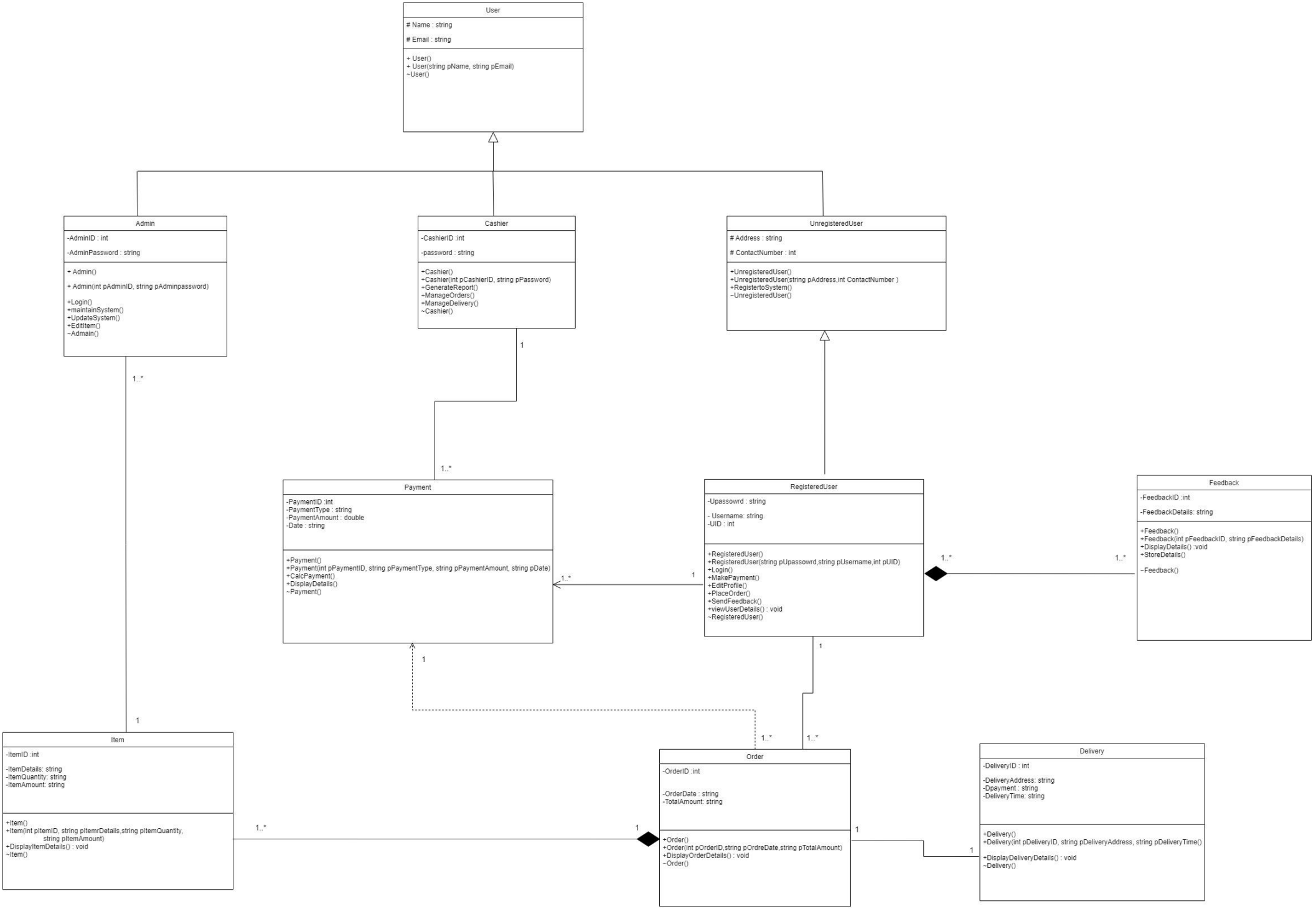
Class Name: Payment	
Responsibility	**Collaborators**
Confirm payments	Registered customer
Validate the payment	Cashier
Display details about payment	

9)

Class Name: Delivery	
Responsibility	**Collaborators**
Accept the delivery	Cashier
Handover the order	Registered user

10)

Class Name: Order	
Responsibility	**Collaborators**
Accept the order	Casher
Deliver the orders	Delivery


User
Name : string
Email : string
+ User()
+ User(string pName, string pEmail)
~User()
Admin
-AdminID : int
-AdminPassword : string
+ Admin()
+ Admin(int pAdminID, string pAdminpassword)
+Login()
+maintainSystem()
+UpdateSystem()
+EditItem()
~Admain()
Cashier
-CashierID :int
-password : string
+Cashier()
+Cashier(int pCashierID, string pPassword)
+GenerateReport()
+ManageOrders()
+ManageDelivery()
~Cashier()
UnregisteredUser
Address : string
ContactNumber : int
+UnregisteredUser()
+UnregisteredUser(string pAddress,int ContactNumber)
+RegistertoSystem()
~UnregisteredUser()
Payment
-PaymentID :int
-PaymentType : string
-PaymentAmount : double
-Date : string
+Payment()
+Payment(int pPaymentID, string pPaymentType, string pPaymentAmount, string pDate)
+CalcPayment()
+DisplayDetails()
~Payment()
RegisteredUser
-Upassowrd : string
- Username: string.
-UID : int
+RegisteredUser()
+RegisteredUser(string pUpassowrd,string pUsername,int pUID)
+Login()
+MakePayment()
+EditProfile()
+PlaceOrder()
+SendFeedback()
+viewUserDetails() : void
~RegisteredUser()
Feedback
-FeedbackID :int
-FeedbackDetails: string
+Feedback()
+Feedback(int pFeedbackID, string pFeedbackDetails)
+DisplayDetails() :void
+StoreDetails()
~Feedback()
Item
-ItemID :int
-ItemDetails: string
-ItemQuantity: string
-ItemAmount: string
+Item()
+Item(int pItemID, string pItemrDetails,string pItemQuantity,
string pItemAmount)
+DisplayItemDetails() : void
~Item()
Order
-OrderID :int
-OrderDate : string
-TotalAmount: string
+Order()
+Order(int pOrderID,string pOrdreDate,string pTotalAmount)
+DisplayOrderDetails() : void
~Order()
Delivery
-DeliveryID : int
-DeliveryAddress: string
-Dpayment : string
-DeliveryTime: string
+Delivery()
+Delivery(int pDeliveryID, string pDeliveryAddress, string pDeliveryTime()
+DisplayDeliveryDetails() : void
~Delivery()
1
1..*

Feedback.h

```
#include <string>

class Feedback{


private:


int FeedbackID;

string FeedbackDetails;


public:


Feedback();

Feedback(int pFeedbackID,string pFeedbackDetails);

void GenerateReport();

void DisplayDetails();

void StoreDetails();


~Feedback();


};
```

Feedback.cpp

```
#include<iostream>

using namespace std;

#include <string>

#include"Feeback.h"


Feedback::Feedback()

{

cout << "Default Constructor Admin() called" << endl;

}


Feedback::Feedback(int pFeedbackID,string pFeedbackDetails){

  FeedbackID=pFeedbackID;

  FeedbackDetails=pFeedbackDetails;

}

Feedback::~Feedback()

{

 cout << "Destructed" << endl;

}
```

Payment.h

```
#define SIZE 10

#include <string>

#include"Cashier.h"

class Payment{


private:

Cashier* Cashiers;

int PaymentID;

string PaymentType;

double PaymentAmount;C

string Date;

public:

Payment();

Payment(int pPaymentID, string pPaymentType,double pPaymentAmount,string
pDate, Cashier* pCashiers);

void CalcPayment();

void DisplayDetails();

~Payment();

};
```

Payment.cpp

```
#include<iostream>

using namespace std;

#include <string>

#include"Payment.h"

Payment::Payment()

 {

cout << "Default Constructor Payment() called" << endl;

}

Payment::Payment(int pPaymentID, string pPaymentType,double
pPaymentAmount,string pDate){

PaymentID=pPaymentID;

PaymentType=pPaymentType;

PaymentAmount=pPaymentAmount;

Date=pDate;

}

Payment::~Payment()

{

 cout << "Destructed" << endl;

}
```

Cashier.h

```
#define SIZE[10]
#include"Payment.h"
#include"User.h"
class Cashier : public User{
private:
int CashierID;
string Password;
Payment* Payments[SIZE];
public:
Cashier();
Cashier(int pCashierID,string pPassword);
Void addPayment(Payment *1);
void GenerateReport();
void ManageOrders();
void ManageDelivery();
~Cashier();
};
```

Cashier.cpp

```
#include<iostream>
using namespace std;
#include <string>
#include"Cashier.h"
Cashier::Cashier()
{
cout << "Default Constructor Cashier() called" << endl;
}
Cashier::Cashier(int pCashierID,string pPassword)
{
  CashierID=pCashierID;
  Password=pPassword;
}
Cashier::~Cashier()
{
 cout << "Destructed" << endl;
}
```

RegisteredUser.h

```
#include “Payment.h”
#include “Orders.h”
#include “Feedback.h”
#include <string>
#define SIZE 5;
class RegisteredUser : public RegisteredUse  {
   int UID;
  string Upassowrd;
  string Username;
Payment * payment[SIZE];
Order * orders[SIZE];
Feedback * feedback[SIZE];
 public:
RegisteredUser();
RegisteredUser(int pUID,string pUPassowrd, string pUsername);
 void login(string username, string password);
void MakePayment();
void EditProfile();
void PlaceOrder();
void SendFeedback();
void viewUserDetails();
~RegisteredUser();
};
```

RegisteredUser.cpp

```
#include <iostream>
#include<string>
using namespace std;
#include "RegisteredUser.h"

RegisteredUser::RegisteredUser() {
 cout << "Default Constructor RegisteredUser() called" << endl;
}

RegisteredUser:: RegisteredUser(int pUID,string pUPassowrd, string pUsername);
{
UID=pUID;
Username= pUPassowrd;
Upassowrd= pUsername;
}

RegisteredUser::~RegisteredUser()
{
 cout << "Destructed" << endl;
}
```

_Delivery.h

```
#include <iostream>

#include<string>

using namespace std;


class Delivery {

  private:

   int DeliveryID;

  string DeliveryAddress;

  string Dpayment;

 srting DeliveryTime;


  public:

    Delivery();

    Delivery(int pDeliveryID,string pDeliveryAddress,string
pDeliveryTime);

    void DisplayDeliveryDetails();

    ~Delivery();

};
```

Delivery.h

```
#include <iostream>

#include<string>

using namespace std;

#include "Delivery.h"




Delivery::Delivery() {

 cout << "Default Constructor Delivery() called" << endl;

}

Delivery::) Delivery(int pDeliveryID,string pDeliveryAddress,string pDeliveryTime);

{

int DeliveryID;

  string DeliveryAddress;

  string Dpayment;

 srting DeliveryTime;

}

Delivery::~Delivery()

{

  cout << "Destructed" << endl;

}
```

UnregisteredUser.h

```
#include <iostream>

#include<string>

using namespace std;


class UnregisteredUser {

protected:

  string Address;

  int ContactNumber;

  Admin *ad;


public:

  UnregisteredUser();

  UnregisteredUser(string pAddress,int pContactNumber);

  void RegistertoSystem();

  ~UnregisteredUser();


};
```

UnregisteredUser.cpp

```
#include "UnregisteredUser.h"

#include <iostream>

#include<string>

using namespace std;


UnregisteredUser::UnregisteredUser(){

  cout<<"Default constructor UnregisterUser() called"<<endl;

}


UnregisteredUser::UnregisteredUser(string pAddress,int pContactNumber,Admin *pad)

{

  Address = pAddress;

  ContactNumber = pContactNumber;

}


void UnregisteredUser::RegistertoSystem()

{

  cout<<"New user add to the system"<<endl;

}
```

User.h

```
#include<string>

class User{
 protected:
  string Name;
  string Email;

 public:
  User();
  User(string pName,string pEmail);
  ~User();

};
```

User.cpp

```
#include <iostream>
#include<string>
using namespace std;
#include "user.h"

User::User(){
  cout << "Default Constructor User() called" << endl;
}

User::User(string Name, string Email)
{
  Name= Name;
  Email=Email;
}

User::~User()
{
  cout << "Destructed" << endl;
}
```

Admin.h

```
#include<string>


class Admin:public user{
 private:
  int AdminID;
  string AdminPassword;


 public:
  Admin();
  Admin(int pAdminID,string pAdminPassword);
  void login();
  void maintainSystem();
  void UpdateSystem();
  void EditItem();
  ~Admin();


};
```

Admin.cpp

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


Admin::Admin(){
  cout << "Default Constructor Admin() called" << endl;
}


Admin::Admin(int pAdminID, string pAdminPassword)
{
  AdminID = pAdminID;
  AdminPassword = pAdminPassword;
}


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

Order.h

```
#include <string>
#define SIZE[10];
#include"Registeruser.h"
#include"Payment.h"
#include"Delivery.h"
#include"item.h"
class Order{
private:
int OrderID;
string OrderDate;
string TotalAmount;

RegisterUser *reg;
Dilivery *del;
Payment *p;
Item *items[10];

public:
Order();
Order(int OrderID,string pOrderDate,string pTotalAmount);
void OrderDetails();
};
```

Order.cpp

```
#include<iostream>
using namespace std;
#include <string>
#include"Order.h"

Order::Order()
{
  cout << "Default Constructor Order() called" << endl;
}
Order::Order(int pOrderID,string
 pOrderDate,string pTotalAmount)
{
  OrderID=pOrderID;
   OrderDate=pOrderDate;
  TotalAmount=pTotalAmount;
}
Order::~Order()
{
 cout << "Destructed" << endl;
}
```

Item.h

```
#include<string>

#include <iostream>

using namespace std;

class Item {

private:

int ItemId;

string ItemDetails;

 int ItemQuantity;

double ItemAmount;

Admin *adm;

public:

Item();

Item(int pItemId,string pItemDetails,int pItemQuantity,double
pItemAmount);

 void DisplyItemDetails();

~Item();

};
```

Item.cpp

```
#include “Item.h”

#include <iostream>

#include<string>

using namespace std;

Item::Item()

{

cout<<"Default constructor Item() called"<<endl;

}

Item::Item(int pItemId,string pItemDetails,int pItemQuantity,double pItemAmount,Admin *adm)

{

ItemId = pItemId;

pItemDetails = ItemDetails; ItemQuantity = pItemQuantity; ItemAmount=pItemAmount;

} Item::~Item()

{ cout<<"Destructed"<<endl;

UnregisteredUser::~UnregisteredUser()

{

  cout<<"Desructed"<<endl;

}
```

Main program

```
#include <iostream>

using namespace std;



#include "User.cpp"

#include "Feedback.cpp"

#include "Payment.cpp"

#include "Cashier.cpp"

#include "RegisteredUser.cpp"

#include "Delivery.cpp"

#include "UnregisteredUser.cpp"

#include  "Admin.cpp"

#include "order.cpp"

#include  "Item.cpp"



int main() {




Casheir *C1=new Casheir("123","Bigbang");

Casheir *C2=new Casheir("124","Bigbang2");


Payment *P1=new Payment("001","COD",2000.00,"2001-02-03",C1);

Payment *P2=new Payment("002","Online",1000.00,"2001-04-03",C2);


Feedback *F1= new Feedback(1,"good");

Feedback *F2= new Feedback(2,"bad");
```

```
RegisteredUser *R1=(1,"panzer","jude");

RegisteredUser *R2=(2,"panze445r","june");


Delivery *D1=(1,"Kurunegala","8.00 PM");

Delivery *D2=(2,"Malabe","7.30 AM");



User *U1=("Rogar","Rogar@gmail.com");

User *U2=("umarr","Umarr@gmail.com");


UnregisteredUser *UU1=("Kurunegala","01123552");

UnregisteredUser *UU2=("Malabe","09952112");


Admin *A1=(1125,"Toolsh56");

Admin *A2=(12336,"Tuupllsh56");


Item *I1=(1,"Bag",4,200.00);

Item *I2=(2,"Shoes",4,900.00);


Order *O1=(2,"2022-2-9",2000.00);

Order *O2=(5,"2022-2-9",8000.00);



  return 0;

}
```