Sri Lanka Institute of Information Technology

Assignment 2

KDY_12

Online food delivery and takeout

Object oriented concepts – IT1100 B.Sc. (Hons) in Information Technology

Group Details

Group Number : KDY_12

Student ID	Student Name	Emai	Contact Number
IT21281396	Harischandara C.R	IT21281396@my.sliit.lk	+94 76 094 9741
IT21281464	Wijethunga I.A	IT21281464@my.sliit.lk	+94 71 892 2774
IT21352744	Amarathunga S.Y	IT21352744@my.sliit.lk	+94 72 446 3655
IT21344756	Kasthuriarachchi K.A.B.L	IT21344756@my.sliit.lk	+94 75 452 0826
IT21295874	Chamika W.M.S	IT21295874@my.sliit.lk	+94 76 757 4410

Table of Contents

System Requirements

1. Unregistered users are unable to see the system; instead, they must register with the system by entering information such as their name, address, email, and phone number.
2. Customers and administrators are two categories of registered users who may log into the system by inputting the right username and password.
3. They can use the system to 'buy' and’ sell' food and beverages.
4. Food data such as pricing, promotion, picture, and food item should be added by the administrator
5. The meal information can be deleted or updated by the administrator. Following the confirmation of details, the system should produce a unique id for the meal and the customer.
6. The admin receives the order date and order id when the order is placed.
7. Customers should be allowed to add numerous items to their shopping carts.
8. Customers should be able to add their favorite foods to a wish list.
9. Food should be filtered by type, price, and rating for the customer.
10. Customers can order ahead of time by selecting foods.
11. The payment must be fulfilled by all resisted customers.
12. The registered customer must be able to submit payment information such as payment type and card information.
13. Following the payment, a "payment id" is produced for the admin's "admin id" and the costumers' "food id."
14. A report with the Oder information is delivered to the customer and admin when the payment is validated by the bank and other trusted answers.

Noun & Verb Analysis

(NOUNS)

1. Unregistered users are unable to see the system; instead, they must register with the system by entering information such as their name, address, email, and phone number.
2. Customers and admin are two categories of registered users who may log into the system by inputting the right username and password.
3. They can use the system to 'buy' and' sell' food and beverages.
4. Food data such as pricing, promotion, picture, and food item should be added by the admin
5. The meal information can be deleted or updated by the admin. Following the confirmation of details, the system should produce a unique id for the meal and the customer.
6. The admin receives the order date and order id when the order is placed.
7. Customers should be allowed to add numerous items to their shopping carts.
8. Customers should be able to add their favorite foods to a wish list.
9. Food should be filtered by type, price, and rating for the customer.
10. Customers can order ahead of time by selecting foods.
11. The payment must be fulfilled by all registered customers.
12. The registered customer must be able to submit payment information such as payment type and card information.
13. Following the payment, a "payment id" is produced for the admin's "admin id" and the costumers' "food id."
14. A report with the Oder information is delivered to the customer and admin when the payment is validated by the bank and other trusted resourese.

Identified Classes

1. Unregister user
2. Registered user
3. Admin
4. Customer
5. Food
6. Preorder
7. Payment
8. Cart
9. favorite list

Redundant

1. Admin
2. Buyer

Out of scope

1. System
2. Bank
3. Trusted resources

Attributes

1. Details
2. Name
3. Address
4. NIC
5. Email
6. Contact
7. Password
8. Price
9. Status
10. Food ID
11. Date of sell

Noun & Verb Analysis

(VERBS)

1. Unregistered users are unable to see the system; instead, they must register with the system by entering information such as their name, address, email, and phone number.
2. Customers and admin are two categories of registered users who may log into the system by inputting the right username and password.
3. They can use the system to 'buy' and’ sell' food and beverages.
4. Food data such as pricing, promotion, picture, and food item should be added by the admin
5. The meal information can be deleted or updated by the admin. Following the confirmation of details, the system should produce a unique id for the meal and the customer.
6. The admin receives the order date and order id when the order is placed.
7. Customers should be allowed to add numerous items to their shopping carts.
8. Customers should be able to add their favorite foods to a wish list.
9. Food should be filtered by type, price, and rating for the customer.
10. Customers can order ahead of time by selecting foods.
11. The payment must be fulfilled by all registered customers.
12. The registered customer must be able to submit payment information such as payment type and card information.
13. Following the payment, a "payment id" is produced for the admin's "admin id" and the costumers' "food id."
14. A report with the Oder information is delivered to the customer and admin when the payment is validated by the bank and other trusted recourses.

Method

1. Admin
    - Add food to the system
    - Delete food from the system
    - update food item

2. Unregistereduser
    - Register to the system by providing details.
    - Allow to view the Food Items
3. Registereduser
    - Login to system
    - Can view the Food Items
    - Add food items to cart
    - Buy Food items
    - Search Food Items

4. Payment
    - Make a new payment
    - Generate payment id
    - Check payment details
    - Confirm payment details

5. Cart
    - Add food items
    - Delete food items
6. Order
    - Add food items
    - Remove Food items

7. Food

    - Add food Details
    - Delete food Details
    - Update food Details

CRC Cards

Admin	
Responsibility	**Collaborators**
Add food to the system	Food
Delete food from the system	Food
update food item	Food

Unregistereduser	
Responsibility	**Collaborators**
Register to the system	
Allow to view the Food Items	Food

Registereduser	
Responsibility	**Collaborators**
Login to system	
Can view the Food Items	Food
Add food items to cart	Cart
Buy Food items	Order, Payment
Search Food Items	Food

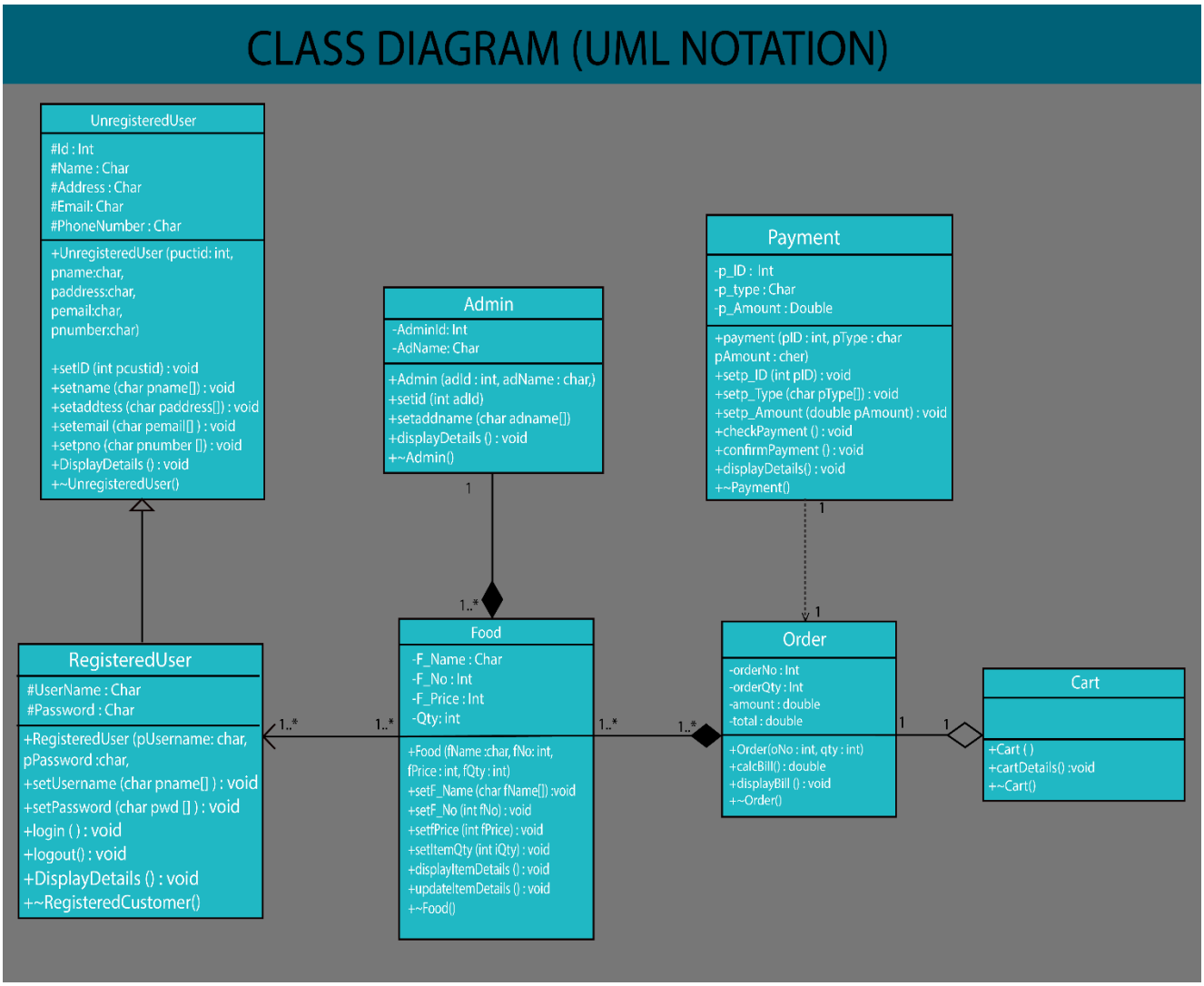
Payment	
Responsibility	**Collaborators**
Make a new payment	
Generate payment id	Payment
Check payment details	Ordre, Payment
Confirm payment details	

Cart	
Responsibility	**Collaborators**
Add food items	Cart, Food
Delete food items	Cart

Order	
Responsibility	**Collaborators**
Add food items	Order, Food
Remove Food items	Order

Food	
Responsibility	**Collaborators**
Add food Details	Admin
Delete food Details	Admin
Update food Details	Admin

Class Diagram(UML notation)



C++ coding part

Header Files

RegisteredUser.h

```
#include"Food.h"

#include"UnregisteredUser.h"



class RegisteredUser :public UnregisteredUser

{

protected:

char Username[10];

char Password[10];


public:


RegisteredUser();

RegisteredUser(char pUsername[],char pPassword[]);

void setUsername(char pname[] );

void setPassword(char pwd []);

void displayDetails();

void login();
```

```
void logout();
void searchFood(Food *pfood);
~RegisteredUser();
};
```

order.h

```
#include"Food.h"
class Order
{
private:
  int orderNo;
  int orderQty;
  double amount;
  double total;
public:
  Order();
  Order(int oNo, int qty);
  double calcBill();
  void displayBill(Food* p);
  ~Order();
};
```

Payment.h

```
#include "Order.h"

class Payment

{

private:


int p_ID;

char p_Type[20];

double p_Amount;

public:


Payment();

Payment(int pID, char pType[],double pAmount);

void setp_ID( int pID);

void setp_Type(char pType[]);

void setp_Amount(double pAmount);

void checkPayment(Order*p);

void confirmPayment();

void displayDetails();

~Payment();

};
```

Food.h

```
#include "Admin.h"
class Food
{
private:
  char F_Name[30];
  int F_No;
  int F_Price;
  int Qty;
  Admin *fAdmin[2];
public:
  Food();
  Food( char fName[], int fNo, int fPrice, int Qty);
  void setQty(int iQty);
  void setF_Name(char fName[]);
  void setF_No( int fNo);
  void setfPrice(int fPrice);
  void displayItemDetails();
  void updateItemDetails();
  ~Food();
};
```

Cart.h

```
#include "Order.h"
class Cart
{
private:
  Order* order[5];
public:
  Cart();
  Cart(Order* order[]);
  void cartDetails();
  ~Cart();

};
```

Admin.h

```
class Admin
{
private:
    int adminId;
    char addName[2];

public:
  Admin();
```

```
 Admin(int adId,char adName[]);
 int setid(int adId);
 char setaddName(char adname[]);
 void displayDetails();
 ~Admin();
};
```

UnregisteredUser.h

```
#include "Food.h"
class UnregisteredUser
{
protected:
int ID;
char Name[20];
char Address[50];
char Email[30];
char phoneNumber[10];
public:

UnregisteredUser();
UnregisteredUser(int pcustid,  char pname[],  char paddress[],  char pemail[] ,char pnumber []);

void setID(int pcustid);
void setname(char pname[]);
```

```
void setaddress(char paddress[]);

void setemail(char pemail[] );

void setpno(char pnumber []);



 void displayDetails();

~UnregisteredUser();

};
```

CPP File

RegisteredUser.cpp

```
#include "RegisteredUser.h"

#include <cstring>

RegisteredUser::RegisteredUser()

{

strcpy(Username, "");

strcpy(Password, "");

}

RegisteredUser::RegisteredUser(char pUsername[],char pPassword[]):UnregisteredUser(
pcustid, pname,   paddress,  pemail , pnumber )

{

  strcpy(Username,pUsername);

  strcpy(Password,pPassword);
```

```
}
void RegisteredUser::setUsername(char pname[] )
{


}
void RegisteredUser::setPassword(char pwd [])
{


}
void RegisteredUser::displayDetails()
{


}
void RegisteredUser::login()
{


}
void RegisteredUser::logout()
{


}
void RegisteredUser::searchFood(Food *pfood)
{
```

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

oder.cpp

```
#inculde"Order.h"
#include<iostream>
using namespace std;
#include<cstring>Order::Order()
{
orderNo=0;
orderQty=0;
amount=0;
total=0;
}Order::Order(int oNo, int qty);
{
orderNo = oNo;
orderQty = qty;
}void Order::calcBill()
{}
```

```
void Order::displayBill()
{}
//DestructorOrder::~Order()
{
}
```

Payment.cpp

```
#inculde"Order.h"
#include<iostream>
usingnamespace std;
#include<cstring>


Payment::Payment()
{
P_ID = 0;
strcpy(P_type,"");
P_Amount = 0;
}
Payment::Payment(int p_Id,char p_type[],char p_Amount[]);
{
P_ID = p_ID ;
```

```
strcpy(P_type, "p_type");
 P_Amount = p_Amount;
}
void Payment::checkPayment(Order*p)
{
}
void Payment::confirmPayment()
{
}
void Payment::displayPaymentDetails()
{
}
Payment::~Payment()
{
//Destructor
}
```

Food.cpp

```
#include "Admin.h"
#include<iostream>
using name std;
#include<cstring>
```

```
Food::Food()
{
  strcpy(F_Name,"");
  F_No=0;
  F_Price=0;
  Qty=0;

}
Food::Food(Food( char fName[], int fNo, int fPrice, int Qty);)
{
 F_name=fName
 F_No=fNo;
 F_Price=fPrice;
 qty=Qty;



}
void Food::Qty(int iQty)
{

}
void Food::setF_Name(char fName[])
{
```

```
}
void Food::setF_No( int fNo)
{

}
void Food::setfPrice(int fPrice)
{

}
void Food::displayItemDetails()
{

}
void Food::updateItemDetails()
{

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

Cart.cpp

```
#include "Cart.h"
#include<iostream>
using namespace std;
#include<cstring>

//constructors
Cart::Cart ()
{

}
//setters
void Cart::Cart(Order* order[])
{

}
void Cart::cartDetails()
{

}
//Destructor
Cart::~cart()
{
```

```
}
```

Admin.cpp

```
#inculde"Admin.h"
#include<iostream>
using namespace std;
#include<cstring>
Admin::Admin()
{
strcpy(addName,"");
adminId=0;
}
Admin::Admin(int adId,char adName[]);
{
adminId=aID;
strcpy(addName,"adName");
}
void Admin::setaddName(char adname[])
{
}
void Admin::displayDetails()
{
```

```
}
Admin::~Admin()
{
};
```

UnregisteredUser.cpp

```
{
 int ID=0;
 strcpy( Name, "");
 strcpy(Address, "");
 strcpy( Email, "");
 strcpy( phoneNumber, "");

}
UnregisteredUser(int pcustid,  char pname[],  char paddress[],  char pemail[] ,char pnumber [])
{
 ID=pcustid
 strcpy( Name, pname[]);
 strcpy(Address, paddress[] );
 strcpy( Email,  pemail[] );
 strcpy( phoneNumber, pnumber []);
}

void UnregisteredUser:: setID(int pcustid)
{
```

```
}
void UnregisteredUser::setname(char pname[])
{

}
void UnregisteredUser::setaddress(char paddress[])
{

}
void UnregisteredUser::setemail(char pemail[] )
{

}
void UnregisteredUser::setpno(char pnumber [])
{

}
void UnregisteredUser::displayDetails()
{

}
UnregisteredUser::~UnregisteredUser()
{
}
```

main.cpp

```
#include "Admin.h"
#include "Cart.h"
#include "Food.h"
#include "Payment.h"
#include "Order.h"
#include "RegisteredUser.h"
#include "UnregisteredUser.h"

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

int main() {

UnregisteredUser unreg1;
RegisteredUser  reg1;
Admin Ad1;
Order* O1=new Order;
Payment* P1=new Payment;
Food* F1=new Food;
Cart* C1=new Cart;


 //----Method Calling------
  unreg1.displayDetails();

  reg1.login();
  reg1.logout();
  reg1.displayDetails();
  reg1.searchFood();


  Ad1.displayDetails();

  O1->calcBill();
  O1->displayBill();

  P1->checkPayment();
```

```
  P1->confirmPayment();
  P1->displayDetails();

  F1->displayItemDetails();
  F1->updateItemDetails();

  C1->cartDetails();

  delete unreg1;
  delete reg1;
  delete Ad1;
  delete O1;
  delete P1;
  delete F1;
  delete C1;
}
```