



Topic : Online Food delivery and Takeout.

Group no : KGL_05

Campus : Kurunegala

Submission Date: 20 May 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
IT21247668	Ranathunga S.W.H.D.D	0774992822
IT21340628	Abeywickrama A.A.D	0717415410
IT21356636	Perera P.K.L	0763716372
IT21358616	Munasingha M. H. C	0764321206
IT21360046	Jayasekara B.B.R.Y	0719620114

Description

Online food delivery system where customers can order and get food online. Food that customers order online are delivered by the restaurant to a delivery agent. The delivery agent delivers the food to the customer. All payments here can be made using online or cash on delivery.

However, all money transfer systems are analysed and stored by the accountant.

Requirement

1. Guest should register to the system.
2. After registering, the customer must enter the username, password and login to the system.
3. The registered customers who have access to the system can view their profile and edit or delete the profile.
4. The registered customer chooses the restaurant of their choice.
5. Then select the food of your choice from the menu card of the selected restaurant.
6. Customers can order the food items of their choice.
7. Once the food is ordered, choose whether to pay online or cash on delivery.
8. Checking the payment details system of customers who make payments online.
9. Then send the confirmed correct transaction report to the registered customer.
10. Giving a receipt for payment to the customer who pays online.
11. The system then provides customer information to the deliverer.
12. The deliverer takes the item to the appropriate location according to the customer's relevant information.
13. The customer who opts for the cash on delivery method will make the payment to the delivery driver upon receipt of the item.
14. The delivery driver then inserts the money into the system.
15. The customer who receives the product gives the system their feedback on our service.
16. The admin updates the system and manages the system.

Nouns

1. **User** should register to the **system**.
2. After registering, the **customer** must enter the **username**, **password** and login to the **system**.
3. The **registered customers** who have access to the **system** can view their profile and edit or delete the profile.
4. The **registered customer** chooses the **restaurant** of their choice.
5. Then select the food of your choice from the **menu card** of the selected **restaurant**.
6. **Customers** can order the **food items** of their choice.
7. Once the **food** is ordered, choose whether to pay **online** or **cash on delivery**.
8. Checking the **payment details** system of **customers** who make **payments** online.
9. Then send the confirmed correct **transaction report** to the **registered customer**.
10. Giving a **receipt** for **payment** to the **customer** who pays online.
11. The **system** then provides **customer information** to the **deliverer**.
12. The **deliverer** takes the item to the appropriate **location** according to the customer's relevant **information**.
13. The **customer** who opts for the cash on delivery method will make the **payment** to the **delivery driver** upon **receipt** of the **item**.
14. The **delivery driver** then inserts the **money** into the **system**.
15. The **customer** who receives the product gives the system their **feedback** on our service.
16. The **admin** updates the system and manages the system.

Verbs

1. User **should register** to the system.
2. After registering, the customer **must enter** the username, password and **login** to the system.
3. The registered customers who have access to the system can **view their profile** and **edit** or **delete** the profile.
4. The registered customer **chooses** the restaurant of their choice.
5. Then **select the food** of your choice from the menu card of the selected restaurant.
6. Customers **can order** the food items of their choice.
7. Once the food is ordered, choose whether to **pay** online or cash on delivery.
8. **Checking the payment details** system of customers who **make payments** online.
9. Then **send the confirmed** correct transaction report to the registered customer.
10. **Giving a receipt** for payment to the customer who pays online.
11. The system then **provides customer information** to the deliverer.
12. The deliverer **takes the item** to the appropriate location according to the customer's relevant information.
13. The customer who opts for the cash on delivery method **will make the payment** to the delivery driver upon receipt of the item.
14. The delivery driver then **inserts the money** into the system.
15. The customer who receives the product **gives the system their feedback** on our service.
16. The admin **updates the system** and **manages** the system.

Identifying classes - Nouns Verbs Analysis

- | | |
|------------------------|----------------------------------|
| 1. User | - Class |
| 2. Registered Customer | - Class |
| 3. Username | - Attribute (of Registered user) |
| 4. Password | - Attribute (of Registered user) |
| 5. Restaurant | - Class |
| 6. Menu card | - Class |
| 7. Food item | - Class |
| 8. Online payment | - Attribute (of payment) |
| 9. Cash on delivery | - Attribute (of payment) |
| 10. Transaction report | - Attribute (of payment) |
| 11. Payment | - Class |
| 12. Location | - Attribute (of deliverer) |
| 13. Feedback | - Class |
| 14. Admin | - Class |
| 15. Order | - Class |
| 16. Deliverer | - Class |

Selected Classes

1. User
2. Registered customer
3. Restaurant
4. Menu card
5. Food item
6. Payment
7. Feedback
8. Admin
9. Order
10. Deliverer

CRC Cards

Class Name: User	
Responsibility	Collaborators
Register to the system	
Preview food items	Food item

Class Name: Registered_customer	
Responsibility	Collaborators
Can view their profile and edit or delete it.	
Choose the restaurant	Restaurant
Make payment	Payment
Order the food items	Food item, Order
Choose payment method	Payment
Gives the feedback	Feedback

Class Name: Restaurant	
Responsibility	Collaborators
Add restaurant	Restaurant
Select restaurant	Restaurant
Remove restaurant	Restaurant

Class Name: Menu_card	
Responsibility	Collaborators
View the food item	Registered customer, food item, restaurant
Select the food item	Registered customer, menu card

Class Name: Food_Item	
Responsibility	Collaborators
Add food item	Food item
Remove food item	Food item

Class Name: Payment	
Responsibility	Collaborators
Confirm payment information	Registered customer
View payments details	

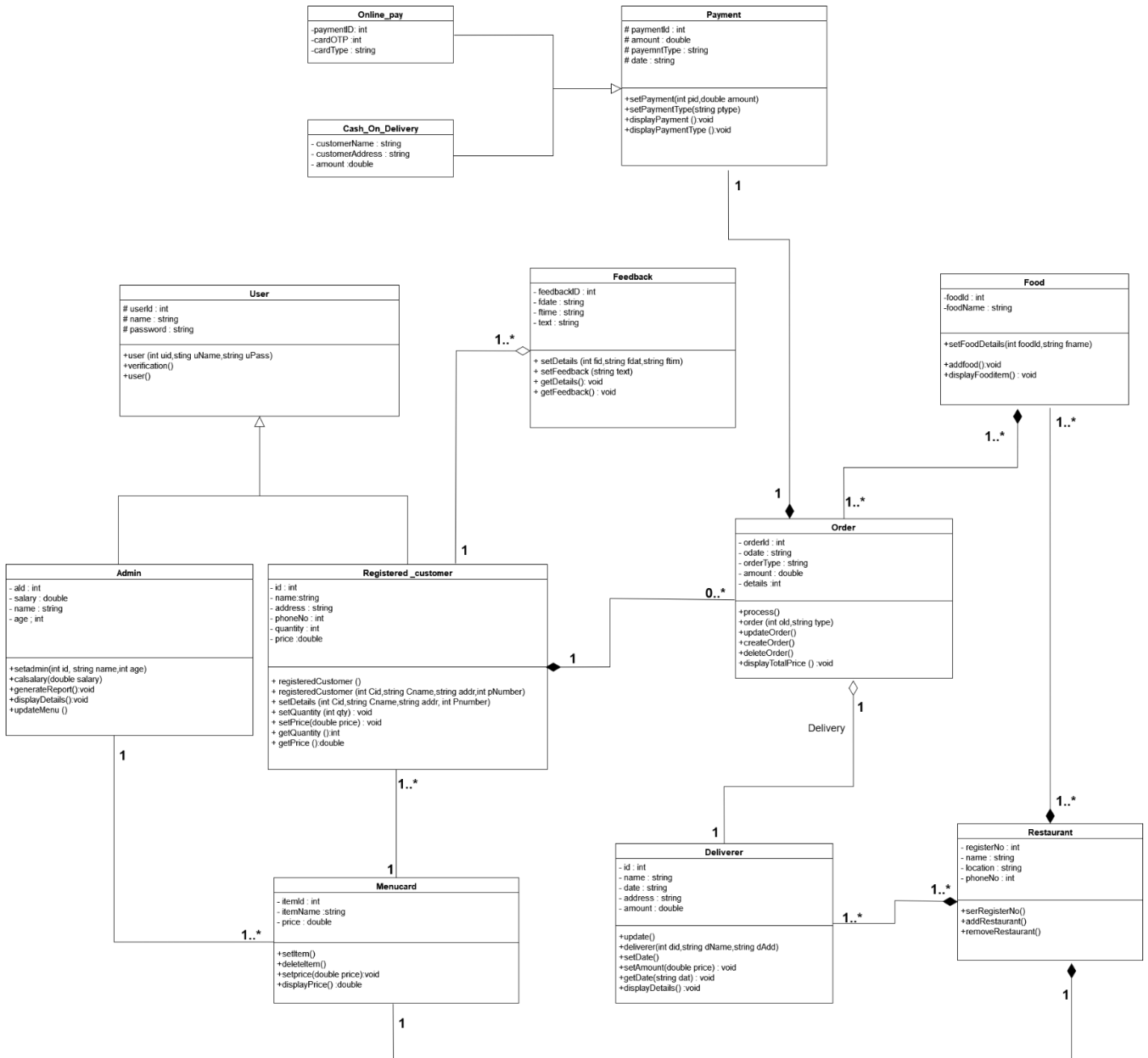
Class Name: Feedback	
Responsibility	Collaborators
Given the feedback	Registered customer
View feedback	Admin, restaurant

Class Name: Admin	
Responsibility	Collaborators
System Updates	Restaurant, food item
System manage	

Class Name: Order	
Responsibility	Collaborators
Informing customers about prices	Registered customer
Cancel order	Order

Class Name: Deliver	
Responsibility	Collaborators
Check receiver	Order
Check order number	Order
Follow routes and time schedule	Deliverer

Class diagram



Exercise 2: Coding

User.h

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

class user
{
protected:
    int id;
    string name;
    string email;
    string password;

public:
    user();
    void setDetails(int uId, string uName, string umail, string uPass);
    void displayUser();
};
```

User.cpp

```
#include "user.h"

user::user()
{
    cout << "contractor call" << endl;
}

void user::setDetails(int uId, string uName, string umail, string uPass)
{
    id = uId;
    name = uName;
    email = umail;
    password = uPass;
}

void user::displayUser()
{
    cout << "ID:"<< id << endl;
    cout << "Name:" << name << endl;
    cout << "Email:" << email << endl;
    cout << "Password:" << password << endl;
}
```

Registered_customer.h

```
#include <iostream>
#include <string.h>
#include <user.h>
#include <payment>
#include <order.h>

using namespace std;

class registeredCustomer:public user
{
private:
    int customerId;
    string address;
    string name;
    double price;
    int quantity;
    Payment* paymentts[SIZE];

public:
    registeredCustomer ();
    registeredCustomer (int cId, string cAddress, double cprice, int qty);
    void setprice (double cprice);
    void setquantity (int qty);
    double calPrice();
    ~registeredCustomer();
};
```

Registered_customer.cpp

```
#include "registeredCustomer.h"

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

registeredCustomer::registeredCustomer(int cId, string cAddress, double cprice,
int qty)
{
    customerId = cId;
    address = cAddress;
}

void registeredCustomer::setprice(double cprice)
{
    price = cprice;
}

void registeredCustomer::setquantity(int qty)
{
    quantity = qty;
```

```

}

double registeredCustomer::calPrice()
{
    cout << "payment:" << quantity * price << endl;
}

registeredCustomer::~~registeredCustomer()
{
    cout << "Destructor of registeredCustomer called" << endl;
}

```

Order.h

```

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

class order
{
public:
    int orderID;
    string date;
    string orderType;
    double amount;
    int details;

public:
    order();
    order(int orderID, string date, string orderType, double amount, int
details);
    void updateOrder();
    void totalPrice();
    void deleteOrder();
    ~order()

};

```

Order.cpp

```

// order.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "order.h"
using namespace std;

order::order(); {
    cout << "Default constructor of the order is called" << endl;
}

order::order(int orderId, string data, string orderType) {

```

```

        Id    = orderId;
        date = oDate;
        type = oType;
    }

void order::updateOrder()
{
    cout << "update order" << endl;
}

```

Deliverer.h

```

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

class driver
{
private:
    int id;
    string name;
    string address;
    string date;
    double amount;

public:
    driver();
    deliver(int id, string dName, string dAddress, string dDate, double
amount)
    void setDate();
    void setAmount();
    void displayDetails();
    ~driver()
};

```

Deliverer.cpp

```

// driver.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "driver.h"
using namespace std;

driver::driver() {
    cout << "default constructor of the driver is call" << endl;
}

driver::driver(int id, string dName, string address, string date, double amount);
{
    id = dId;
    name = dName;
    address = dAddress;
    date = dDate;
    amount = dAmount;
}

```

```

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

```

Admin.h

```

#include <iostream>
#include <string.h>
using namespace std;
class admin
{
private:
    int aId;
    double salary;
    string name;
    int age;

public:
    admin();
    admin(int Id, string aname, int aage);
    double calsalary;
    void generateReprot();
    void displayDetails();
    ~admin();
};

```

Admin.cpp

```

#include "stdafx.h"
#include "admin.h"

admin::admin()
{
    cout << "default constructor of admin called" << endl;
}
admin::admin(int Id, string aname, int aage)
{
    age = aage;
    name = aname;
    aId = Id;
}

void admin::generateReprot()
{
    cout << "Input the name and Id" << aId << ":"; cin >> aId;
}

void admin::displayDetails()
{
    cout<<"Age="<<age<<"Name="<<name<<"AId="<<aId<<endl<<endl
}

admin::~~admin()
{
    cout << "destructure of admin called" << endl;
}

```

Feedback.h

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

class feedback
{
private:
    int feedbackID;
    string fdate;
    string ftime;
    string text;
public:
    feedback();
    setDetails(int fid, string fdate, string ftime);
    setFeedback(string text);
    void getDetails();
    void getFeedback();
    ~feedback();
};
```

Feedback.cpp

```
// feedback.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "feedback.h"

feedback::feedback()
{
    cout << "default constructor of feedback called" << endl;
}

feedback::feedback(int fid, string fdate, string ftime)
{
    fid = id;
    fdate = date;
    ftime = time;
}

void feedback::getDetails()
{
    cout << "Fid=" << fid << "Fdate=" << fdate << "Ftime=" << ftime << < endl
<< endl;
}

void feedback::getFeedback()
{
    cout << "input the feedback" << fid << fdate << ftime << ":"; cin >> fid;
}

feedback::~~feedback()
{
    cout << "destructor of feedback called" << endl;
}
```

Restaurnt.h

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

class Resturent
{
private:
    char regNo;
    string name;
    string location;
    string email;
    int phoneNo;

public:
    Resturent();
    Resturent(string rregNo, string rname, string rlocation, string remail,
int rphoneNo);
    void setResturentNo();
    void addResturent(Resturent * r1, Resturent * r2);
    void removeResturent();
    ~Resturent();

};
```

Restaurant.cpp

```
#include "stdafx.h"
#include "Resturent.h"//Resturent class
#include<iostream>
#include <string>
#include<iomanip>
#define SIZE; //define
using namespace std;

Resturent::Resturent()//Default Constructor
{
    regNo = 0;
    name[20] = 0;
    location = 0;
    email = 0;
    phoneNo = 0;
}

Resturent::Resturent(string regNo, string rname, string rlocation, string remail,
int rphoneNo)
{
    regNo = regNo;
    name = rname;
    location = rlocation;
    email = remail;
    phoneNo = rphoneNo;
}
```



```

void Resturent::setResturentNo()//Methods
{

}

void Resturent::addResturent(Resturent* r1,Resturent* r2)//Methods
{

}

void Resturent::removeResturent()//Methods
{

}

Resturent::~~Resturent()
{

}

```

Menu_card.h

```

#pragma once
#include <string>
using namespace std;

class Menu_card // class name
{
private://Atributes
    int itemID;
    string name;
    float price;

public://Method
    Menu_card();
    Menu_card(int itemID, string name, float price);
    void setItem(Menu_card* m1, Menu_card* m2);
    void deleteItem();
    ~Menu_card();
};

```

Menu_card.cpp

```

#include "stdafx.h"
#include "Menu_card.h"
#include <string>
#include <iostream>
using namespace std;

Menu_card::Menu_card()//Default Constructor
{
    itemID = 0;
    name = "";
    price = 0;
}

```

```

}

Menu_card::Menu_card(int fitemID, string fname, float fprice)//Overload
Constructor
{
    itemID = fitemID;
    name = fname;
    price = fprice;
}

void Menu_card::setItem(Menu_card * m1, Menu_card * m2)
{

}

void Menu_card::deleteItem()//Method
{
}

Menu_card::~Menu_card()
{
}

```

Payment.h

```

class Payment
{
};

#include <string>
#include "order.h"
class Payment {
private:
    int paymentID;
    double amount;
    string paymentMethod;
    string date;
    Order* orderID;
public:
    Payment();
    Payment(int id, double amt, string method, string pDate);
    void displayPaymentDetails();
    ~Payment();
};

```

Payment.cpp

```

#include "Payment.h"
#include <iostream>;

```

```

#include <string>

using namespace std;
Payment::Payment() {
    cout << "Default Constructor of Payment called" << endl;
}
Payment::Payment(int id, double amt, string method, string pDate) {
    paymentID = id;
    amount = amt;
    paymentMethod = method;
    date = pDate;
}
Payment::~~Payment() {
    cout << "Destructor of Payment called" << endl;
}

```

Food.h

```

class Food
{
};
#include<iostream>
#include<string.h>
using namespace std;
class food
{
protected:
    int foodid;
    string foodname;

public:
    user();
    void setDetails(int fId, string fName);
    void displayfood();
};

```

Food.cpp

```

#include "food.h"
food::food()
{
    cout << "food details" << endl;
}
void food::setDetails(int fId, string fName)
{
    fid = fId;
    fname = fName;
}
void food::displayFooditem()
{
    cout << "ID:" << fid << endl;
    cout << "Name:" << fname << endl;
}

```

Main.cpp

```
// main.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "Order.h"
#include "Payment.h"
#include "Admin.h"
#include "User.h"
#include "Resturent.h"
#include "Menu_card.h"
#include "Food.h"
#include "Deliverer.h"
#include "Feedback.h"
#include "Registered_customer"
#include<iostream>
#include <string>
#include<iomanip>
#define SIZE;

using namespace std;

int main()
{
    User* u1 = new User(001, "Harindu", "harindu12@gmail.com",
    "hriu52"); //Overloaded constructor
    //User* u1 = new User(); // Default constructor

    Register_customer* r1 = new Register_customer(1,
    "No.56,Malkaduwawa,Kurunegala"); //Overloaded constructor
    //RegisterCustomer* r1 = new RegisterCustomer(); // Default constructor

    Order* o1 = new Order(200, 05/04/2020, "Ready Deliver"); //Overloaded
    constructor
    //Order* o1 = new Order(); // Default
    constructor

    Deliver* d1 = new Deliver(1001, "Dilshan", "No.65,Ynathampalawa,
    Kurunegala", "04/07/2022", 1520.00); //Overloaded constructor
    //Deliver* d1 = new Deliver(); // Default constructor

    Admin* a1 = new Admin(52, "Ravin", 12001); //Overloaded constructor
    //Admin* a1 = new Admin(); // Default constructor

    Feedback* f1 = new Feedback(5001, "05/01/2022", "10.20 AM"); //Overloaded
    constructor
    Feedback* f2 = new Feedback(5002, "05/02/2022", "11.40 AM");
    //Feedback* f1 = new Feedback(); // Default
    constructor

    Resturent* r1 = new Resturent("REG001", "Mr.kottu", "Mallawapitiya",
    "kavi23@gmail.com", "0785424165"); //Overloaded constructor
    //Resturent* r1 = new Resturent(); // Default constructor
```

```

        Menu_card* m1 = new Menu_card(150, "Chicken Kottu", 600.00); //Overloaded
constructor
        //MenuCard* m1 = new MenuCard(); //
Default constructor

        Payment* p1 = new Payment(10010, "Cash on Delivery"); //Overloaded
constructor
        //Payment* p1 = new Payment(); //
Default constructor

        Food* fd1 = new Food(1001, "Pasta"); //Overloaded constructor
        Food* fd2 = new Food(1001, "Pizza");
        Food* fd3 = new Food(1001, "Burger");
        //Food* f1 = new Food(); // Default constructor

        //-----Method Call-----//
        u1->registeredtoSystem();

        r1->Login();
        r1->MakePayment();
        r1->SetFeedback();

        p1->setPayment();
        p1->setPaymentType();
        p1->displayPaymentMethod();
        p1->displayPaymentType();

        o1->createOrder();
        o1->updateOrder();
        o1->displayTotalPrice();
//inheritance relationship

        d1->setAmount();
        d1->update();

        fd1->addFood();
        fd1->displayFoodItem();

        a1->generateReport();
        a1->displayDetails();

        f1->getDetails();
        f1->getFeedback();

        delete u1;
        delete r1;
        delete p1;
        delete o1;
        delete fd1;
        delete d1;
        delete a1;
        delete f1;

        return 0;
}

```

Individual Contribution.

Student ID	Student Name	Class diagram	Cording
IT21247668	Ranathunga S.W.H.D.D	<ul style="list-style-type: none">• User• Registered customer	<ul style="list-style-type: none">• user.h• user.cpp• registeredCustomer.h• registeredCustomer.cpp
IT21340628	Abeywickrama A.A.D	<ul style="list-style-type: none">• Food item• Payment	<ul style="list-style-type: none">• payment.h• payment.cpp• food.h• food.cpp
IT21356636	Perera P. K. L	<ul style="list-style-type: none">• Restaurant• Menu card	<ul style="list-style-type: none">• restaurant.h• restaurant.cpp• menucard.h• menucard.cpp
IT21358616	Munasingha M. H. C	<ul style="list-style-type: none">• Feedback• Admin	<ul style="list-style-type: none">• admin.h• admin.cpp• feedback.h• feedback.cpp
IT21360046	Jayasekara B.B.R.Y	<ul style="list-style-type: none">• Order• Deliverer	<ul style="list-style-type: none">• order.h• order.cpp• deliverer.h• deliverer.cpp