Sri Lanka Institute of Information Technology

Topic : Online Laundry Service

Group no : MLB_08.01_04

Campus : Malabe

Submission Date:

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
IT21165252	Kuruppu K.A.G.S.R.	071 081 6191
IT21164330	Arandara.S.D.	077 992 0805
IT21165184	Dewasinghe U.H.	077 518 1446
IT21165870	Dharmawardhana W.G.D.S.	076 553 5004
IT21163418	Bandara K.M.N.M.	070 222 3564

Description of the requirements

Diamond shine is an online laundry service. Anyone can access it through the website or the Diamond shine app. When a visitor visits to the store he/she can only view items, services, feedbacks, contact details, and further information about our store.

A visitor can register to the system as a customer or a delivery person. Selecting the correct register type is required. If a visitor wants to register as a customer, he/she must provide Name, Phone number, E-mail, Address, and a suitable password. If a visitor wants to register as a delivery person, he/she must provide Name, Phone number, E-mail, License number, Date of birth and a suitable password. After fulfilling the asked requirements, a visitor can login to the system using their credentials as a registered user. If a visitor does not wish to register, he/she can continue to view the system as a guest.

A Customer can login to the system using their credentials. After that he/she can view services and items. He/she can order services and bye items from the store. A customer can view feedbacks and also, he/she can leave feedbacks for other users to see. When a customer bye items or place orders, he/she can select a preferred payment method. Selected customers can get discounts under certain conditions.

A delivery person can login to system using their credentials. He/she can view available jobs and take them. Once the delivery person logged in to the system, he/she can view order details. And he/she can the location where the order needs to be delivered. A delivery person must update the order details after order is delivered (whether the order is delivered to the customer or not).

Admins can login to the system by using their credentials. After logging in, an admin can generate reports about the orders. An admin can change the prices of services and items. Admins can respond to feedbacks left by the other users.

System owner also needs to login to the system using his/her credentials. The owner should verify their login by his/her 4-digit pin. System owner can generate and view reports about orders. And change prices of services and items. Only the system owner can remove or add new services and items to the store.

1.Requirements Related to System

- Visitors must register as a customer or delivery persons to access services.
- Visitors can view services list and view item list
- Visitors can view feedback.
- Visitors can view contact details.
- If visitor want to register the system visitor must select the register type.
- Unregistered users must register by providing a name, phone number, email, address, password to register, if he/she want to become a customer.
- During registration Unregistered users must register by providing a name, phone number, email, license number, DOB to register, if he/she want to become a delivery person.
- Registered users can login to system while Visitors can continue as a guest view the system.
- Customers can view service list and item list.
- Customers must be able to buy items.
- Customers must be able to order services.
- Customers must be able to view others feedback and add feedback
- Customers must be able to buy orders using the selected payment methods.
- Customer can get discounts for his/her order
- Delivery persons can login to the system by giving username or email and the password.
- Delivery persons can view jobs and take jobs.
- Delivery persons can view details about the order.
- Delivery persons can see the location of the order.
- Delivery persons must enter the delivery details after the complete the order (Whether the “order is ordered or not”).
- Admins can log in to the system by giving his username and the password.
- Admins can generate reports about the orders.

- Admins can change the prices of existing items and services
- Admins can response to the feedbacks.
- System owners also need to log in to the system by giving username and password.
- System owners must enter his 4-digit pin number for verifications.
- System owners can do everything that admin can do.
- System owners can view reports.
- System owners can Add new service and items.

2. Noun Verb analysis and identify classes

Noun Verb Analysis

Noun 

Verb 

- Visitors must register as a customer or delivery persons to access services.
- Visitors can view services list and view item list
- Visitors can view feedback.
- Visitors can view contact details.
- If visitor want to register the system visitor must select the register type.
- Unregistered users must register by providing a name, phone number, email, address, password to register, if he/she want to become a customer.
- During registration, Unregistered users must register by providing a name, phone number, email, license number, DOB to register, if he/she want to become a delivery person.
- Registered users can login to system while Visitors can continue as a guest view the system.
- Customers can view service list and item list.
- Customers must be able to buy items.
- Customers must be able to order services.

- Customers must be able to view others feedback and add feedback
- Customers must be able to buy orders using the selected payment methods.
- Customer can get discounts for his/her order
- Delivery persons can login to the system by giving username or email and the password.
- Delivery persons can view jobs and take jobs.
- Delivery persons can view details about the order.
- Delivery persons can see the location of the order.
- Delivery persons must enter the delivery details after the complete the order (Whether the “order is ordered or not”).
- Admins can log in to the system by giving his username and the password.
- Admins can generate reports about the orders.
- Admins can change the prices of existing items and services.
- Admins can response to the feedbacks.
- System owners also need to log in to the system by giving username and password.
- System owners must enter his 4-digit pin number for verifications.
- System owners can do everything that admin can do.
- System owners can view reports.
- System owners can Add new service and new items.

Identify classes

Noun	Classes
Visitors	
Service List	
Item List	
Feedbacks	✓
Contact Details	
Register type	
Unregistered users	
Name	
Phone number	
Email	
Address	
Password	
Customer	✓
Registration	
DOB	
License number	
Delivery person	✓
Registered users	✓
Guest	
Items	✓
Services	✓
Orders	✓
Payment methods	
Payment	✓
Discount	✓
His/her	
System	
Username	
Jobs	✓
Admins	✓
Reports	
Prices of existing items and services	
System owners	✓
Pin number	
Verification	
New service	
New item	

Rules for rejecting unwanted nouns

Redundant	An event and operations	Outside of the scope	Metalanguage	An Attributes
Visitors	Verification	System	Guest	Name
He/she	Registration			Phone number
Guest				Email
				Password
				Address
				Register type
				Payment methods
				DOB
				License number
				Username
				Pin number
				Password

Verbs

	Verbs
Visitor	• Register • View • Select
Registered user	• Login
Customer	• login • view • buy • order • add • selected
Delivery person	• login • view • take • view details • see the location • enter • complete
Admins	• login • generate • change • response
System owner	• login • enter • view • add

3. CRC cards

Class Name: Registered users	
Responsibility:	Collaboration:
Register as customer	Customer
Register as a delivery person	Delivery Person
View details	
View feedback	Feedbacks

Class Name: Customer	
Responsibility:	Collaboration:
Store customer details	
View customer details	
Edit customer details	
Add to cart	Orders
Buy items	Items
Buy service	Service
Give feedback	Feedbacks

Class Name: Deliver Person	
Responsibility:	Collaboration:
Store delivery person details	
View Delivery Person details	
Edit delivery Person details	
View Jobs	Jobs
Take Jobs	Jobs
View Details About the order	Orders
See the Location of the order	Orders
Enter the details about the completion of order	Orders

Class Name: Payments	
Responsibility:	Collaboration:
Store Payment details	Orders
Display Payments details	
Calculate Total Payment	
Calculate Discount	Discount

Class Name: Jobs	
Responsibility:	Collaboration:
Store Jobs details	
View Jobs Details	
View Completed Jobs	Delivery Persons

Class Name: Admins	
Responsibility:	Collaboration:
Store Admin details	
Display Admin details	
Update Admin details	
Generate Reports	Payments
Respond to Feedbacks	Feedbacks

Class Name: System Owners	
Responsibility:	Collaboration:
Store system owner details	
View system owner details	
Update system owner details	
Manage items	items
Manage services	Services
Generate reports	payments
Respond to Feedbacks	feedbacks

Class Name: Item	
Responsibility:	Collaboration:
Add items	System owners
Remove items	System owners
Store item details	

Class Name: service	
Responsibility:	Collaboration:
Add service	System owners
Remove service	System owners
Store service details	

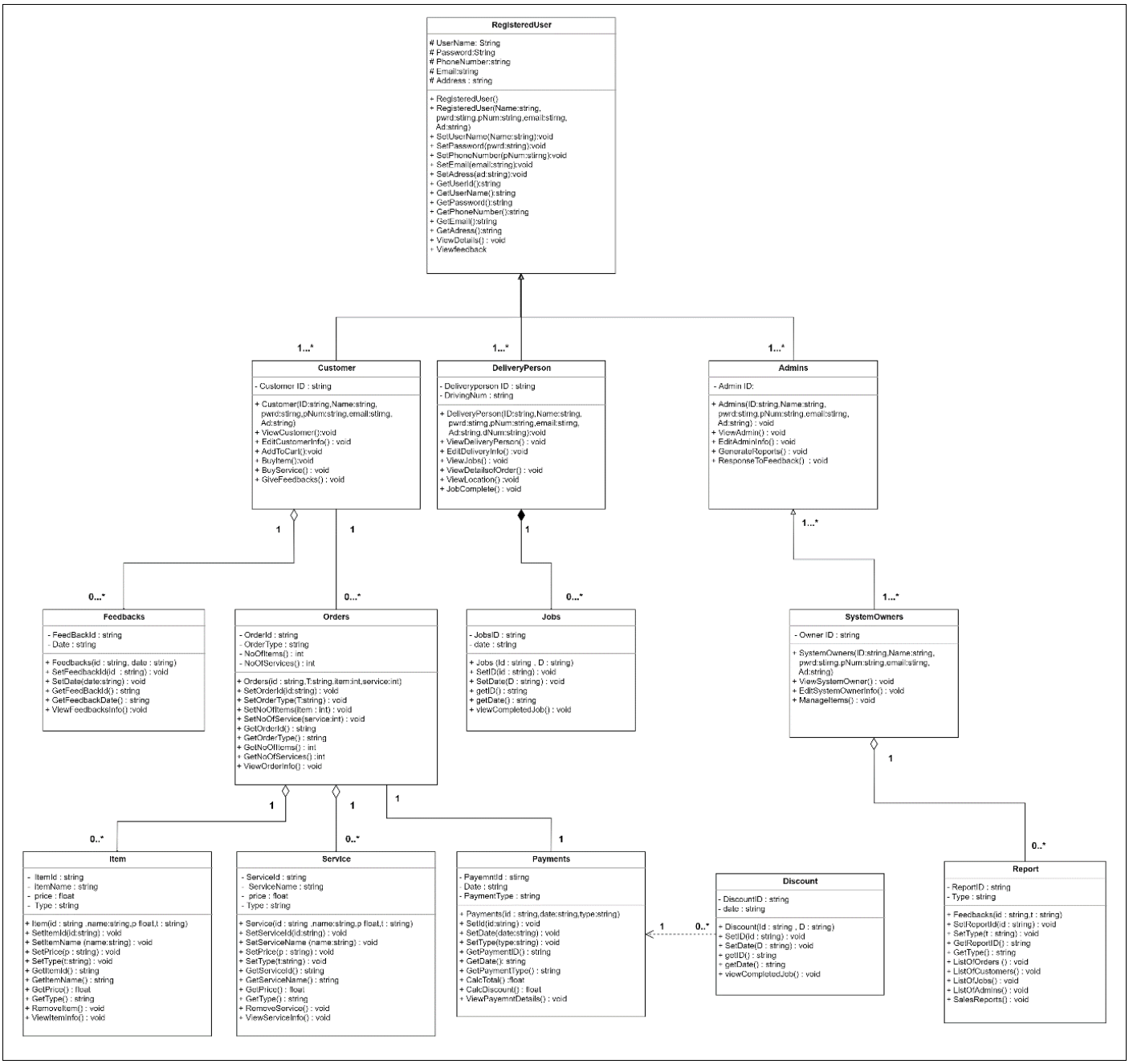
Class Name: Orders	
Responsibility:	Collaboration:
Store order details	Item, services
Display order details	

Class Name: Feedbacks	
Responsibility:	Collaboration:
Store Feedbacks	Customer
View Feedbacks	Customer
Delete Feedback	

Class Name: Report	
Responsibility:	Collaboration:
List of orders	Orders
List of customers	Customers
List of delivery persons	Delivery persons
List of jobs	jobs
List of admins	admins
Sales report	

Class Name: Discount	
Responsibility:	Collaboration:
Store discount details	
Display discount details	
Calculate discount	Payment, Order

Exercise 1


RegisteredUser
UserName: String
Password:String
PhoneNumber:string
Email:string
Address : string
+ RegisteredUser()
+ RegisteredUser(Name:string, pwrd:string,pNum:string,email:string, Ad:string)
+ SetUserName(Name:string):void
+ SetPassword(pwrd:string):void
+ SetPhoneNumber(pNum:string):void
+ SetEmail(email:string):void
+ SetAdress(ad:string):void
+ GetUserId():string
+ GetUserName():string
+ GetPassword():string
+ GetPhoneNumber():string
+ GetEmail():string
+ GetAdress():string
+ ViewDetails() : void
+ Viewfeedback
1...*
Customer
- Customer ID : string
+ Customer(ID:string,Name:string, pwrd:string,pNum:string,email:string, Ad:string)
+ ViewCustomer():void
+ EditCustomerInfo() : void
+ AddToCart():void
+ BuyItem():void
+ BuyService() : void
+ GiveFeedbacks() : void
DeliveryPerson
- Deliveryperson ID : string
- DrivingNum : string
+ DeliveryPerson(ID:string,Name:string, pwrd:string,pNum:string,email:string, Ad:string,dNum:string):void
+ ViewDeliveryPerson() : void
+ EditDeliveryInfo() :void
+ ViewJobs() : void
+ ViewDetailsofOrder() : void
+ ViewLocation():void
+ JobComplete() : void
Admins
- Admin ID:
+ Admins(ID:string,Name:string, pwrd:string,pNum:string,email:string, Ad:string) : void
+ ViewAdmin() : void
+ EditAdminInfo() : void
+ GenerateReports() : void
+ ResponseToFeedback() : void
1
0...*
Feedbacks
- FeedBackId : string
- Date : string
+ Feedbacks(id : string, date : string)
+ SetFeedbackId(id : string) : void
+ SetDate(date:string) : void
+ GetFeedBackId() : string
+ GetFeedbackDate() : string
+ ViewFeedbacksInfo() :void
Orders
- OrderId : string
- OrderType : string
- NoOfItems() : int
- NoOfServices() : int
+ Orders(id : string,T:string,item:int,service:int)
+ SetOrderId(id:string) : void
+ SetOrderType(T:string) : void
+ SetNoOfItems(Item : Int) : void
+ SetNoOfService(service:int) : void
+ GetOrderId() : string
+ GetOrderType() : string
+ GetNoOfItems() : int
+ GetNoOfServices() :int
+ ViewOrderInfo() : void
Jobs
- JobsID : string
- date : string
+ Jobs (Id : string , D : string)
+ SetID(Id : string) : void
+ SetDate(D : string) : void
+ getID() : string
+ getDate() : string
+ viewCompletedJob() : void
SystemOwners
- Owner ID : string
+ SystemOwners(ID:string,Name:string, pwrd:string,pNum:string,email:string, Ad:string)
+ ViewSystemOwner() : void
+ EditSystemOwnerInfo() : void
+ ManageItems() : void
0..*
Item
- ItemId : string
- ItemName : string
- price : float
- Type : string
+ Item(id : string ,name:string,p float,t : string)
+ SetItemId(id:string) : void
+ SetItemName (name:string) : void
+ SetPrice(p : string) : void
+ SetType(t:string) : void
+ GetItemId() : string
+ GetItemName() : string
+ GetPrice() : float
+ GetType() : string
+ RemoveItem() : void
+ ViewItemInfo() : void
Service
- ServiceId : string
- ServiceName : string
- price : float
- Type : string
+ Service(id : string ,name:string,p float,t : string)
+ SetServiceId(id:string) : void
+ SetServiceName (name:string) : void
+ SetPrice(p : string) : void
+ SetType(t:string) : void
+ GetServiceId() : string
+ GetServiceName() : string
+ GetPrice() : float
+ GetType() : string
+ RemoveService() : void
+ ViewServiceInfo() : void
Payments
- PayemntId : string
- Date : string
- PaymentType : string
+ Payments(id : string,date:string,type:string)
+ SetId(id:string) : void
+ SetDate(date:string) : void
+ SetType(type:string) : void
+ GetPaymentID() : string
+ GetDate() : string
+ GetPaymentType() : string
+ CalcTotal() :float
+ CalcDiscount() : float
+ ViewPayemntDetails() : void
Discount
- DiscountID : string
- date : string
+ Discount(Id : string , D : string)
+ SetID(Id : string) : void
+ SetDate(D : string) : void
+ getID() : string
+ getDate() : string
+ viewCompletedJob() : void
Report
- ReportID : string
- Type : string
+ Feedbacks(id : string,t : string)
+ SetReportId(id : string) : void
+ SetType(t : string) : void
+ GetReportID() : string
+ GetType() : string
+ ListOfOrders () : void
+ ListOfCustomers() : void
+ ListOfJobs() : void
+ ListOfAdmins() : void
+ SalesReports() : void

Exercise 2:

Registered_users.h

```
//IT21165252
//Kuruppu K.A.G.S.R.
//Registered_users.h

#pragma once
//Registered User Class
#include <string>

using namespace std;

class Registered_users
{
        protected:
                string username;
                string password;
                string phone_number;
                string email;
                string address;

         public:
                Registered_users();
                Registered_users(string get_username, string get_password, string
      get_pnum, string get_email, string get_address);
                void set_username(string get_username);
                void set_password(string get_password);
                void set_phonenumber(string get_pnum);
                void set_email(string get_email);
                void set_address(string get_address);
                virtual void view_details();
                void view_feedback();
                ~Registered_users();
};
```

Registered_users.cpp

```
//IT21165252
//Kuruppu K.A.G.S.R.
//Registered_users.cpp

#include "Registered_user.h"
#include<iostream>

using namespace std;

Registered_users::Registered_users()
{

}

Registered_users::Registered_users(string get_username, string get_password, string
get_pnum, string get_email, string get_address)
 {
        username = get_username;
        password = get_password;
        phone_number = get_pnum;
        email = get_email;
        address = get_address;
}
void Registered_users::set_username(string get_username)
{
        username = get_username;
}
void Registered_users::set_password(string get_password)
{
        password = get_password;
}
void Registered_users::set_phonenumber(string get_pnum)
{
        phone_number = get_pnum;
}
void Registered_users::set_email(string get_email)
{
        email = get_email;
}
void Registered_users::set_address(string get_address)
{
        address = get_address;
}
```

```
void Registered_users::view_details()
{

}
void Registered_users::view_feedback()
{

}

Registered_users::~Registered_users()
{
        cout << "objects are destroyed" << endl;
}
```

Customer.h

```
//IT21165252
//Kuruppu K.A.G.S.R.
//Customer.h

#pragma once
#include <iostream>
#include "Registered_user.h"
#include "Feedbacks.h"
class Order;


using namespace std;

class Customer:public Registered_users
{
	private:
		string Customer_ID;
		Feedbacks *feed[2];
		Order *order[2];
	public:
		Customer(string get_username, string get_password,
			string get_pnum, string get_email,
			string get_address, string get_Customer_ID);
		void Give_feedback();
		void view_details();
		void Edit_customerinfo();
		void Addtocart();
		void buy_item();
		void buy_service();
		void addOrder(Order *o);
		void AddFeedback(Feedbacks *f1,Feedbacks *f2);
		~Customer();
};
```

Customer.cpp

```
//IT21165252
//Kuruppu K.A.G.S.R.
//Customer.cpp

#include "Customer.h"
#include "Registered_user.h"
#include "Feedbacks.h"
#include "Order.h"//com

Customer::Customer(string get_username, string get_password,
   string get_pnum, string get_email,
   string get_address, string get_Customer_ID) :Registered_users(get_username,
get_password, get_pnum, get_email, get_address)
{

   Customer_ID = get_Customer_ID;

}

void Customer::Give_feedback()
{

}
void Customer::view_details()
{
   cout << "name is : " << username << endl;
   cout << "Password is : " << password << endl;
   cout << "Phone number is : " << phone_number << endl;
   cout << "Email is" << email << endl;
   cout << "address iS " << address << endl;
   cout << "Customer ID is " << Customer_ID << endl;
}
void Customer::Edit_customerinfo()
{

}
void Customer::Addtocart()
{

}
void Customer::buy_item()
{

}
```

```
void Customer::buy_service()
{

}
void Customer::AddFeedback(Feedbacks *f1,Feedbacks *f2)
{
  feed[0]=f1;
  feed[1]=f2;
}
void Customer::addOrder(Order *o){

}
Customer::~Customer()
 {

}
```

Del_person.h

```
//IT21165252
//Kuruppu K.A.G.S.R.
//Del_person.h


#pragma once
#include "Registered_user.h"
#include "Jobs.h"
#include <iostream>
#define size 2

using namespace std;

class Del_person:public Registered_users{
private:
  string Del_Id;
  Jobs *jobs[size];

public:
  Del_person(string get_username, string get_password,
                                  string get_pnum, string get_email,
                                  string get_address, string get_D_ID);
  void viewDeliveryDetails();
  void ViewJobs();
  void ViewDetailsOfOrder();
  void location();
  void CompleteJobs();
  void AddJobs();
};
```

Del_person.cpp

```
//IT21165252
//Kuruppu K.A.G.S.R.
//Del_person.cpp

#include <iostream>
#include "Registered_user.h"
#include "Del_person.h"
#include "Jobs.h"
using namespace std;

Del_person::Del_person(string get_username, string get_password,
```

```
                                string get_pnum, string get_email,
                                string get_address, string
get_D_ID):Registered_users(get_username,get_password,get_pnum,get_email,get_address
)
{
      Del_Id=get_D_ID;
}
void Del_person::viewDeliveryDetails()
{
 cout << "name is : " << username << endl;
   cout << "Password is : " << password << endl;
   cout << "Phone number is : " << phone_number << endl;
   cout << "Email is" << email << endl;
   cout << "address iS " << address << endl;
   cout << "Delivery ID is " << Del_Id << endl;
   cout << "Jobs :" << endl;
   AddJobs();
   jobs[0]->viewJob();
   jobs[1]->viewJob();
}
void Del_person::ViewJobs(){

}
void Del_person::ViewDetailsOfOrder(){

}
void Del_person::location(){

}
void Del_person::CompleteJobs(){

}
void Del_person::AddJobs(){
 jobs[0]= new Jobs("J100","22/07/20200");
 jobs[1]= new Jobs("J101","02/01/2022");
}
```

Feedbacks.h

```
//IT21165184
//Dewasinghe U.H.
//Feedbacks.h

#pragma once
#include<iostream>
#pragma once
#include<string>

using namespace std;

class Feedbacks{
   private:
      string Feedback_ID;
      string Date;

   public:
      Feedbacks(string id,string date);
      void SetFeedbackID(string id);
      void SetDate(string date);
      string GetFeedbackID();
      string GetFeedackDate();
      void ViewFeedbackinfo();
      ~Feedbacks();

};
```

Feedbacks.cpp

```
//IT21165184
//Dewasinghe U.H.
//Feedbacks.cpp

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

Feedbacks::Feedbacks(string id,string date)
{
   Feedback_ID = id;
```

```
    Date = date;
}

void Feedbacks::SetFeedbackID(string id)
{
    Feedback_ID = id;
}

void Feedbacks::SetDate(string date)
{
    Date = date;
}

string Feedbacks::GetFeedbackID()
{
    return Feedback_ID;
}

string Feedbacks::GetFeedackDate()
{
    return Date;
}

void Feedbacks::ViewFeedbackinfo()
{
    cout << "Feedback ID : "<< Feedback_ID << endl;
    cout << "Date : " << Date << endl;
}

Feedbacks::~Feedbacks()
{
    cout << "Destructor is called"<<endl;

}
```

Order.h

```
//IT21164330
//Arandara.S.D.
//Order.h

#pragma once
#include <string>
#include "Customer.h"
#include "Item.h"
#include "Service.h"
#include "payment.h"
using namespace  std;

class Order{
        private:
                string orderId;
                string orderType;
                int noOfItems;
                int noOfServices;
                Customer *c1;
                Item *i1[2];
                Service *ser[5];
                Payments *pay;
        public:
                Order();
                Order(string id,string t,int NoItem,int NoService);
                void setOrderId(string id);
                void setOrderType(string t);
                void setNoOfItems(int NoItem);
                void setNoOfServices(int NoService);
                string getOrderId();
                string getOrderTyoe();
                int getNoOfItems();
                int getNoOfServices();
                void viewOrder();
                void CustomerOfOrder(Customer *c);
                void ItemsInOrder(Item *i01);
                void ServicesInOrder(Service *s);
                void PayPayment(Payments *p);
                ~Order();
};
```

Order.cpp

```
//IT21164330
//Arandara.S.D.
//Order.cpp

#include <iostream>
#include "Order.h"
#include <string>
#include "Customer.h"
#include "Order.h"
#include "Item.h"
#include "Service.h"
using namespace std;

Order::Order(){

}
Order::Order(string id,string t,int NoItem,int NoService)
{
  orderId=id;
  orderType=t;
  noOfItems=NoItem;
  noOfServices=NoService;
}
int Order::getNoOfItems()
{
  return noOfItems;
}
int Order::getNoOfServices()
{
    return noOfServices;
}
 void Order:: setOrderId(string id)
{
    orderId=id;
 }
void Order:: setOrderType(string t){

    orderType=t;
}
void Order:: setNoOfItems(int NoItem)
{
    noOfItems=NoItem;
  }
void Order::setNoOfServices(int NoService)
```

```
{
    noOfServices=NoService;
}
void Order::viewOrder(){

}
void Order::CustomerOfOrder(Customer *c){

}
void Order::ItemsInOrder(Item *i5){

}
void Order::ServicesInOrder(Service *s){

}
void Order::PayPayment(Payments *p){

}
Order::~Order(){
    cout << "Objects are destroyed";
}
```

Item.h

```
//IT21165870
//Dharmawardhana W.G.D.S.
//Item.h

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

class Item {
	private:
		string itemId;
		string itemName;
		float price;
		string type;
	public:
		Item(string item_id,string item_name,float item_p,string item_t);
		void setItemId(string item_id);
		void setItemName(string item_name);
		void setItemPrice(float item_p);
		void setItemType(string item_t);
		void removeItem();
		void viewItemInfo();
		~Item();
};
```

Item.cpp

```
//IT21165870
//Dharmawardhana W.G.D.S.
//Item.cpp

#include "Item.h"

Item::Item(string item_id,string item_name,float item_p,string item_t)
{
  itemId = item_id;
  itemName = item_name;
  price = item_p;
  type = item_t;
}
void Item::setItemId(string item_id)
{
    itemId = item_id;
}
void Item::setItemName(string item_name)
{
   itemName = item_name;
}
void Item::setItemPrice(float item_p)
{
   price = item_p;
}
void Item::setItemType(string item_t)
{
   type = item_t;
}
void Item::removeItem(){

}
void Item::viewItemInfo(){

}
Item::~Item(){

}
```

Service.h

```
//IT21165870
//Dharmawardhana W.G.D.S.
//Service.h

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

class Service{
        private:
                string serviceId;
                string serviceName;
                float price;
                string type;
        public:
                Service(string service_id,string service_name,float service_p,string
service_t);
                void setItemId(string service_id);
                void setItemName(string service_name);
                void setItemPrice(float service_p);
                void setItemType(string service_t);
                void removeService();
                void viewServiceInfo();
                ~Service();
};
```

Service.cpp

```
//IT21165870
//Dharmawardhana W.G.D.S.
//Service.cpp

#include "Service.h"

Service::Service(string service_id,string service_name,float service_p,string service_t)
{
  serviceId = service_id;
  serviceName = service_name;
  price = service_p;
  type = service_t;
}
void Service::setItemId(string service_id)
{
  serviceId = service_id;
}
void Service::setItemName(string service_name)
{
  serviceName = service_name;
}
void Service::setItemPrice(float service_p)
{
  price = service_p;
}
void Service::setItemType(string service_t)
{
  type = service_t;
}
void Service::removeService(){

}
void Service::viewServiceInfo(){

}
Service::~Service(){

}
```

Payments.h

```
//IT21163418
//Bandara K.M.N.M.
//Payments.h

#pragma once
//#include "Order.h"
#include <string>

using namespace std;

class Payments{
        private:
                string PaymentID;
                string Date;
                string PaymentType;
  //Order *or5;
        public:
                Payments();
                Payments(string Id, string date, string type);
                void setId(string id);
                void setDate(string date);
                void setType(string type);
                string getpaymentId();
                string getDate();
                string getPaymentType();
                float CalcDiscount();
                void ViewPaymentDetails();
  //void PaymentOrder(Order *o);
};
```

Payment.cpp

```
//IT21163418
//Bandara K.M.N.M.
//Payments.cpp

#include <iostream>
#include <string>
#include "payment.h"

using namespace std;

Payments::Payments()
{
	PaymentID = "NO ID";
	Date = "No date";
	PaymentType = "No PaymentType";
}

Payments::Payments(string Id, string date, string type)
{
	PaymentID = Id;
	Date = date;
	PaymentType = type;
}
void Payments::PaymentOrder(Order *o){

}
void Payments::setDate(string date)
{

}

void Payments::setId(string id)
{

}

void Payments::setType(string type)
{

}

string Payments::getDate()
{

}
```

```
string Payments::getpaymentId()
{

}

string Payments::getPaymentType()
{

}

float Payments::CalcDiscount()
{

}

void Payments::ViewPaymentDetails()
{

}
```

Discount.h

```
//IT21165184
//Dewasinghe U.H.
//Discount.h

#pragma once
#include<iostream>
#include  "payment.h"
#include <string>
using namespace std;

class Discount{
   private:
      string DiscountID;
      string Date;

   public:
      Discount(string id,string date);
      void SetID(string id);
      void SetDate(string date);
      string GetID();
      string GetDate();
      void viewCompletedJobs();
      ~Discount();
      void appPyament(float price,Payments *p1);
};
```

Discount.cpp

```
//IT21165184
//Dewasinghe U.H.
//Discount.cpp

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

Discount::Discount(string id ,string date)
{
   DiscountID = id;
   Date = date;
}

void Discount::SetID(string id)
```

```
{
   DiscountID = id;
}

void Discount::SetDate(string date)
{
   Date = date;
}

string Discount::GetID()
{
   return DiscountID;
}

string Discount::GetDate()
{
   return Date;
}

void Discount::viewCompletedJobs()
{
   cout<<"Discount ID :"<<DiscountID<<endl;
   cout<<"Date : "<<Date<<endl;
}



Discount::~Discount()
{
   cout<<"Distructor is called"<<endl;
}
void Discount::appPyament(float price,Payments *p1){
  // dependecy
}
```

Jobs.h

```
//IT21163418
//Bandara K.M.N.M.
//Jobs.h

#pragma once
#include <string>

using namespace std;

class Jobs{
        private:
                string JobsID;
                string date;
        public:
                Jobs();
                Jobs(string Id, string D);
                void setID(string id);
                void setDate(string D);
                string getID();
                string getDate();
                void viewJob();
};
```

Jobs.cpp

```
//IT21163418
//Bandara K.M.N.M.
//Jobs.cpp

#include <iostream>
#include <string>
#include "Jobs.h"

using namespace std;

Jobs::Jobs()
{
        JobsID = "No ID";
        date = "No date";
}

Jobs::Jobs(string Id, string D)
{
```

```
        JobsID = Id;
        date = D;
}

void Jobs::setDate(string D)
{

}

void Jobs::setID(string id)
{

}

string Jobs::getDate()
{

}

string Jobs::getID()
{

}

void Jobs::viewJob()
{
        cout <<"Job ID :" <<JobsID << endl;
  cout << "Job date :" << date << endl;
}
```

Admins.h

```
//IT21164330
//Arandara.S.D.
//Admins.h

#pragma once
#include "Registered_user.h"
#include <iostream>

using namespace std;

class Admins: public Registered_users{
        protected:
                string Admin_ID;
        public:
                Admins(string get_username, string get_password,
                        string get_pnum, string get_email,
                        string get_address,string ID);
                void ViewAdmin();
                void editAdmin();
                void GenerateReports();
                void ResponseToFeedbacks();
                ~Admins();
};
```

Admins.cpp

```
//IT21164330
//Arandara.S.D.
//Admins.cpp

#include<iostream>
#include "Registered_user.h"
#include "Admins.h"

using namespace std;

Admins::Admins(string get_username, string get_password,
                                    string get_pnum, string get_email,
                                    string get_address,string ID):Registered_users(get_username,
get_password, get_pnum, get_email, get_address)
{
    Admin_ID=ID;
 }
void Admins::ViewAdmin()
{
   cout << "name is : " << username << endl;
   cout << "Password is : " << password << endl;
   cout << "Phone number is : " << phone_number << endl;
   cout << "Email is" << email << endl;
   cout << "address iS " << address << endl;
   cout << "Admin ID is " << Admin_ID << endl;
}
void Admins::editAdmin(){

}
void Admins::GenerateReports(){

}
void Admins::ResponseToFeedbacks(){

}
Admins::~Admins(){

}
```

SystemOwner.h

```
//IT21164330
//Arandara.S.D.
//SystemOwner.h

#include<iostream>
#include "Registered_user.h"
#include "Admins.h"

using namespace std;

Admins::Admins(string get_username, string get_password,
                                    string get_pnum, string get_email,
                                    string get_address,string ID):Registered_users(get_username,
get_password, get_pnum, get_email, get_address)
{
  Admin_ID=ID;
  }
void Admins::ViewAdmin()
{
   cout << "name is : " << username << endl;
   cout << "Password is : " << password << endl;
   cout << "Phone number is : " << phone_number << endl;
   cout << "Email is" << email << endl;
   cout << "address iS " << address << endl;
   cout << "Admin ID is " << Admin_ID << endl;
}
void Admins::editAdmin(){

}
void Admins::GenerateReports(){

}
void Admins::ResponseToFeedbacks(){

}
Admins::~Admins(){

}


#pragma once
#include "Registered_user.h"
#include "Admins.h"
#include <iostream>
```

```
#include"Report.h"

class SystemOwner : public Admins{
        private:
                string Admin_ID;
                Report *Rep[2];
        public:
                SystemOwner(string get_username, string get_password,
                        string get_pnum, string get_email,
                        string get_address,string ID);
                void ViewSystemOwner();
                void EditSystemOwnerDetails();
                void ManageItems();
                void GenerateReport(Report *r1,Report *r2);
                ~SystemOwner();
};
```

SystemOwner.cpp

```
//IT21164330
//Arandara.S.D.
//SystemOwner.cpp

#include <iostream>
#include "Registered_user.h"
#include "Admins.h"
#include "SystemOwner.h"

using namespace std;

SystemOwner::SystemOwner(string get_username, string get_password,
                        string get_pnum, string get_email,
                        string get_address,string ID):Admins(get_username,
get_password, get_pnum, get_email, get_address,ID){

    }
void SystemOwner::ViewSystemOwner()
{
   cout << "name is : " << username << endl;
   cout << "Password is : " << password << endl;
   cout << "Phone number is : " << phone_number << endl;
   cout << "Email is :" << email << endl;
   cout << "address iS : " << address << endl;
```

```
    cout << "System Owner ID is : " << Admin_ID << endl;
}
void SystemOwner::EditSystemOwnerDetails(){

}
void SystemOwner::ManageItems(){

}
void SystemOwner::GenerateReport(Report *r1,Report *r2){
  Rep[0]=r1;
  Rep[1]=r2;
}

SystemOwner::~SystemOwner(){

}
```

Report.h

```
//IT21164330
//Arandara.S.D.
//Report.h

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

class Report{
        private:
                string ReportID;
                string Type;
        public:
                Report(string id,string t);
                void setReportId(string id);
                void setReportType(string t);
                string getReportId();
                string getReportType();
                void ViewFeddBackReports();
                void ListOfOrders();
                void ListOfServices();
                void SalesRepor();
                ~Report();
};
```

Report.cpp

```
//IT21164330
//Arandara.S.D.
//Report.cpp

#include "Report.h"
#include <string>
#include <iostream>

using namespace std;

Report::Report(string id,string t)
{
 ReportID=id;
 Type=t;
}
void Report::setReportId(string id)
{
```

```
 ReportID=id;
}
void Report::setReportType(string t)
{
 Type=t;
}
string Report::getReportId()
{
 return ReportID;
}
string Report::getReportType()
{
 return Type;
}
void Report::ViewFeddBackReports()
{

}
void Report::ListOfOrders()

{

}
void Report::ListOfServices()
{

}
void Report::SalesRepor()
{

}
Report::~Report(){
 cout << "Object is destroyed" << endl;
}

/*
 Report(string id,string t);
 void setReportId(string id);
 void setReportType(string t;
 string getReportId();
 string getReportType();
 void ViewFeddBackReports();
 void ListOfOrders();
 void ListOfServices();
 void SalesRepor();
 ~Report();
*/
```

main.cpp

```
#include <iostream>
#include "Registered_user.h"
#include "Customer.h"
#include "Admins.h"
#include "SystemOwner.h"
#include "Del_person.h"
#include "Jobs.h"
#include "Feedbacks.h"
#include "Report.h"
#include "Order.h"
#include "Item.h"
#include "Service.h"
using namespace std;

int main(){



//Registered user dynamic objects
Registered_users *R_User1,*R_User2;
R_User1= new
Registered_users("Gihaan","XX5Xs","0771542525","giha@gmial.com","120/1st
road/Colombo");
R_User2= new Registered_users("Shan","sfshs","0112546687","shan@s.com","125/1st
road/Colombo");
cout << "-----------------------------------------------" << endl;

//customer dynamic objects
Customer *cus1,*cus2;
cus1= new Customer("Dammya","12sd","0112545555","dammya@g.com","No.52/2nd
lane/Kaluthara","CUS100");
cus2= new Customer("Dimiya","12sdxs","0112545563","dammya@y.com","No.55/2nd
lane/Kaluthara","CUS100");
            cus1->view_details();
            cus2->view_details();
 cout << "-----------------------------------------------" << endl;

  // Admins dynamic objects
 Admins *Ad1,*Ad2;
 Ad1= new Admins("Nimshan","qwert","1234565522","giha@1.com","No.120/lane
2/colombo","Admin1");
 Ad2= new Admins("Nimshani","qwertw","123478522","gimnan@1.com","No.120/lane
2/colombo 5","Admin2");
```

```
 Ad1->ViewAdmin();
 Ad2->ViewAdmin();
 cout << "-----------------------------------------------" << endl;

 //System Owner
 SystemOwner *SO1,*SO2;
 SO1 = new SystemOwner("Ranil","qazxs","1234785","ranil.com","No.1/lane 2/colombo
5","SystemOwner1");
  SO2 = new SystemOwner("Mahinda","qplm","1234785","Maina@gmail.com","No.85/lane
2/colombo 5","SystemOwner2");
 SO1->ViewSystemOwner();
 SO2->ViewSystemOwner();
cout << "-----------------------------------------------" << endl;

 //delivery person
 Del_person *del1,*del2;
 del1= new Del_person("John","qplm","","Maina@gmail.com","No.85/lane 2/colombo
5","Del100");
 del2= new Del_person("Mahinda","qplm","1234785","Maina@gmail.com","No.85/lane
2/colombo 5","Del101");

 del1-> viewDeliveryDetails();
 del2 -> viewDeliveryDetails();



 cout << "-----------------------------------------------" << endl;

 // Feddbacks
 Feedbacks *f1,*f2;
 f1= new Feedbacks("F100","02/05/2022");
 f2= new Feedbacks("F101","32/05/2021");
 Customer *cus3;
 cus3= new Customer("Dammya","12sd","0112545555","dammya@g.com","No.52/2nd
lane/Kaluthara","CUS100");
 cus3->AddFeedback(f1,f2);

 //Reports
 Report *report1;
 Report *report2;
 report1= new Report("R100","Type A");
 report2= new Report("R101","Type C");

 SystemOwner *SO5;
 SO5=new SystemOwner("Mahinda","qplm","1234785","Maina@gmail.com","No.85/lane
2/colombo 5","SystemOwner2");
```

```
SO5->GenerateReport(report1,report2);

 cout << "-----------------------------------------------" << endl;
//item
Item *it1,*it2;
it1= new Item("IT100","Item1",1200.00,"TypeA");
it2= new Item("IT101","Item2",1200.00,"TypeB");

Order *ord1;
ord1= new Order("O100","TypeA",5,0);

ord1->ItemsInOrder(it1);
ord1->ItemsInOrder(it2);

//Service
Service *ser1;
ser1= new Service("S105","SERVICE1",1000.00,"TYPEG");

 // Delete regsitered user
delete R_User1;
delete R_User2;

// delete customers
delete cus1;
delete cus2;
delete cus3;

// delete admin
delete Ad1;
delete Ad2;

//Delete system admin
delete SO1;
delete SO2;
delete SO5;

//delete delivery person
delete del1;
delete del2;

//Report
delete report1;
delete report2;

// order
delete ord1;
```

```
  // item
  delete it1;
  delete it2;
  //Service
  delete ser1;
}
```

Contributions

Student ID	Student Name	Individual Contribution
IT21165252	Kuruppu K.A.G.S.R.	**CRC:** • Registered users • Customer • Delivery person **Code:** • Registered_users.h • Registered_users.cpp • Customer.h • Customer.cpp • Del_Person.h • Del_Person.cpp • Main.cpp **Report:** • Requirements • Noun verb analysis and identify classes • Class diagram
IT21164330	Arandara.S.D.	**CRC:** • Order • Admins • System Owners **Code:** • Order.h • Order.cpp • Admins.h • Admins.cpp • SystemOwner.h • SystemOwner.cpp • Report.h • Report.cpp • Main.cpp **Report**: • Requirements • Noun verb analysis and identify classes • Class diagram

IT21165184	Dewasinghe U.H.	**CRC:** • Feedbacks • Report Code: • Feedbacks.h • Feedbacks.cpp • Discount.h • Discount.cpp **Report:** • Description of the requirements • Requirements
IT21165870	Dharmawardhana W.G.D.S.	**CRC:** • Item • Service • Jobs **Code**: • Item.h • Item.cpp • Service.h • Service.cpp • Main.cpp **Report:** • Requirements • Noun verb analysis and identify classes • Class diagram • Report Finalize
IT21163418	Bandara K.M.N.M.	**CRC:** • Payment • Discount **Code:** • Payment.h • Payment.cpp • Jobs.h • Jobs.cpp **Report:** • Requirements • Class diagram