Sri Lanka Institute of Information Technology

Topic – Online Shopping Mall

Group Number – MLB_03.01_08

Campus – Malabe

Submission Date – 17th 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 Number	**Student Name**	**Contact Number**
IT21228612	De Silva S.J.W.	0719516868
IT21225710	Kannangara S.D.R.Y.L.	0771587216
IT21226250	Wickramarathne K.H.I.P.	0703079653
IT21225260	Weerage S.W.Y.W.	0712993662
IT21226632	Weerasinghe W.W.A.B.M.	0717486253

Table of Contents

1 Introduction

Online shopping is the most famous shopping method nowadays. The main reason behind this is the rapid development of technology over past few decades. Also, the current pandemic situation in the world motivates people for online shopping rather than physically going to stores. Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser or a mobile app.

This project is about identifying requirements and classes, designing CRC cards, developing a Class Diagram and writing the codes for an online shopping store where consumers can buy various items.

This report includes Requirements of the system, identified classes, CRC cards for the classes , Class Diagram, and the Coding for the online shopping mall which is named as FlyBuyLk. Coding is written in C++ language.

2 System Requirements

- A User can register to the system and become a registered customer.
- To become a registered customer a user must fill the registration form and create an account.
- Registered customers can enter their credential such as username and password at the login page and log on to the system.
- Guests and registered customers both can search for items.
- Items are displayed after a search with item details such as price, rating, and supplier.
- Only Registered customers can place orders.
- Registered customers can add items to cart.
- An order may contain multiple items.
- Customer can choose a payment method for each order.
- Once the customer confirms the order; the payment is validated, the order is placed and items on the system are updated.
- Registered customers can see the status of ongoing orders and see the list of previous orders.
- Registered customers can rate and give feedback to the items on their order once the order is delivered.
- The staff of the company has their own accounts and credentials to log on to the system.
- Staff members can add, delete, and update the items on system.
- Staff members can delete and view user accounts.
- Staff members can view order details.
- Staff members can generate a report on list of items.
- Delivery manager can assign delivery person for deliveries.

3 Identified Classes

- User
- Registered customer
- Item
- Payment
- Order
- Feedback
- Staff
- Report
- Delivery

4 CRC Cards

Class Name : User	
Responsibilities	**Collaborations**
Register to the system	
Store user details	

Class Name : Registered customer	
Responsibilities	**Collaborations**
Store customer details	user
Place orders	Order
Make payments	Payment
Give feedback for items	Feedback
Search for items	Item

Class Name : Staff	
Responsibilities	**Collaborations**
Store employee credentials	User
Add, Delete, Update items	Item
Generate reports	Item
Assign delivery persons	Delivery

Class Name : Order	
Responsibilities	**Collaborations**
Store order details	
Delete orders	
Calculate order total	
Confirm order	Payment

Class Name : Payment	
Responsibilities	**Collaborations**
Store payment details	Order
Authorize payment	

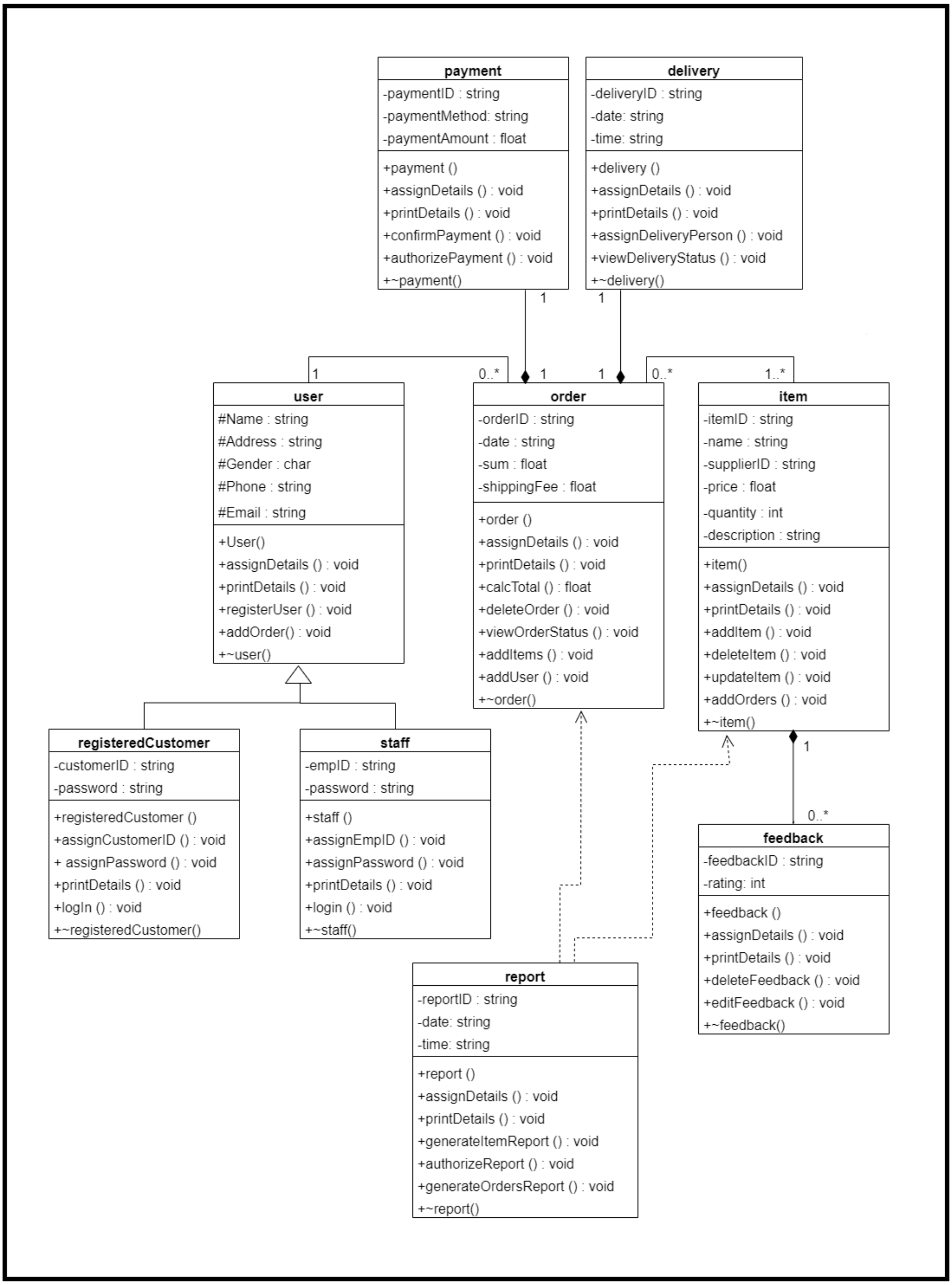
Class Name : Feedback	
Responsibilities	**Collaborations**
Store feedback details	Item
Delete feedback	

Class Name : Item	
Responsibilities	**Collaborations**
Store item details	
Add items	
Delete items	
Update items	

Class Name : Report	
Responsibilities	**Collaborations**
Generate item reports	Item
Generate order reports	Order
Store report details	

Class Name : Delivery	
Responsibilities	**Collaborations**
Store delivery details	
Update delivery status	

5 Class Diagram



6 Coding

6.1 User.h

```
#include <iostream>
#include <cstring>
#include "order.h"
#define SIZE 10

using namespace std ;

//class user implementation
class user {
        protected:
                order * odr[SIZE] ;
                char Name[30] ;
                char Address[80] ;
                char Gender[6] ;
                char Phone[10] ;
                char Email[25] ;

        public:
                user() ; // default constructor
                user (char name[] , char address[] , char gender[] , char phone[] , char email[]) ; // overloaded constructor
                void assignDetails (char Uname[] , char Uaddress[] , char Ugender[] , char Uphone[] , char Uemail[]) ;
                void addOrder (order * ODR) ;
                virtual void printDetails () ;
                void registerUser () ;
                ~user() ; //destructor
};
```

6.2 User.cpp

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

using namespace std ;

user::user() // default constructor  implementation
{

        strcpy(Name , "") ;
        strcpy(Address , "") ;
        strcpy(Gender , "") ;
        strcpy(Phone , "") ;
        strcpy(Email , "") ;

        cout<<"Default constructor ran successfully" << endl << endl ;

}

user::user(char name[] , char address[] , char gender[] , char phone[] , char email[]) //
overloaded constructor implementation
{

        strcpy(Name , name) ;
        strcpy(Address , address) ;
        strcpy(Gender , gender) ;
        strcpy(Phone , phone) ;
        strcpy(Email , email) ;
        cout<<"Overloaded constructor ran successfully" << endl << endl ;
}

user::~user() //destructor
{

}

void addOrder (order * ODR)
{
        //method implementation
}

void user::assignDetails (char Uname[] , char Uaddress[] , char Ugender[] , char
Uphone[] , char Uemail[])
{
        //method implementation
}

void user::printDetails ()
{
        //method implementation
}

void user::registerUser ()
{
        //method implementation
}
```

6.3 Staff.h

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

using namespace std ;

//class staff implementation
class staff : public user
 {
	private:
		char empID[15] ;
		char password[15] ;

	public:
		staff() ; // default constructor

		// overloaded constructor
		staff (char name[] , char address[] , char gender[] , char phone[] , char
email[], char EmpID[] , char Password[] )
		: user (name , address , gender , phone , email) ;

		void assignEmpID (char EmployeeID[]) ;
		void assignPassword (char Pass[]) ;
		void printDetails () ;
		void login (char EmployeeID[], char Pass[]) ;
		~staff() ; //destructor
};
```

6.4 Staff.cpp

```
#include <iostream>
#include <cstring>

#include "staff.h"
#include "user.h"

using namespace std ;

        staff::staff() // default constructor  implementation
{
        strcpy(empID , "") ;
        strcpy(password , "") ;
}

// overloaded constructor implementation
staff::staff (char name[] , char address[] , char gender[] , char phone[] , char email[],
char EmpID[] , char Password[] )
        : user (name , address , gender , phone , email)
{

        strcpy(empID , EmpID) ;
        strcpy(password , Password) ;
}

staff::~staff() //destructor
{

}



void staff::assignEmpID (char EmployeeID[])
{
	//method implementation
}

void staff::assignPassword (char Pass[])
{
	//method implementation
}

void staff::printDetails ()
{
	//method implementation
}

void staff::login (char EmployeeID[], char Pass[])
{
	//method implementation
}
```

6.5 registeredCustomer.h

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

using namespace std ;

//class registeredCustomer implementation
class registeredCustomer : public user
 {
        private:
                char customerID[15] ;
                char password[15] ;

        public:
                registeredCustomer() ; // default constructor

                // overloaded constructor
                registeredCustomer (char name[] , char address[] , char gender[] , char
phone[] , char email[], char RCcustomerID[] , char RCpassword[] )
                : user (name , address , gender , phone , email) ;

                void assignCustomerID (char cusID[]) ;
                void assignPassword (char pw[]) ;
                void printDetails () ;
                void login (char cusID[], char pw[]) ;
                ~registeredCustomer() ; //destructor
};
```

6.6 registeredCustomer.cpp

```
#include <iostream>
#include <cstring>
#include "registeredCustomer.h"
#include "user.h"

using namespace std ;

registeredCustomer::registeredCustomer() // default constructor  implementation
{

        strcpy(customerID , "") ;
        strcpy(password , "") ;

}

// overloaded constructor implementation
registeredCustomer::registeredCustomer (char name[] , char address[] , char gender[] ,
char phone[] , char email[], char RCcustomerID[] , char RCpassword[] )
        : user (name , address , gender , phone , email)
{

        strcpy(customerID , RCcustomerID) ;
        strcpy(password , RCpassword) ;
}

registeredCustomer::~registeredCustomer() //destructor
{

}



void registeredCustomer::assignCustomerID (char cusID[])
```

```
{
        //method implementation
}

void registeredCustomer::assignPassword (char pw[])
{
        //method implementation
}

void registeredCustomer::printDetails ()
{
        //method implementation
}

void registeredCustomer::login (char cusID[], char pw[])
{
        //method implementation
}
```

6.7 delivery.h

```
#include <iostream>
#include <cstring>

using namespace std ;

//class delivery implementation
class delivery {
	private:
		string deliveryID ;
		string date ;
		string time ;

	public:
		delivery() ; // default constructor
		delivery (string delID , string Ddate , string Dtime ) ; // overloaded
constructor
		void assignDetails (string delID , string Ddate , string Dtime ) ;
		void printDetails () ;
		void assignDeliveryPerson () ;
		void viewDeliveryStatus () ;
		~delivery() ; //destructor
};
```

6.8 delivery.cpp

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

using namespace std ;



delivery::delivery() // default constructor  implementation
{
	deliveryID = "" ;
	date = "" ;
	time = "" ;
}

delivery::delivery (string delID , string Ddate , string Dtime )  // overloaded
constructor implementation
{
	deliveryID = delID ;
	date = Ddate ;
	time = Dtime ;
}

delivery::~delivery() //destructor
{


}

void delivery::assignDetails (string delID , string Ddate , string Dtime )
{
	//method implementation
}
```

```
void delivery::printDetails ()
{
	//method implementation
}

void delivery::assignDeliveryPerson ()
{
	//method implementation
}

void delivery::viewDeliveryStatus ()
{
	//method implementation
}
```

6.9 feedback.h

```
#include <iostream>
#include <cstring>

using namespace std ;

//class feedback implementation
class feedback
{
        private:
                string feedbackID ;
                int rating ;



        public:
                feedback() ; // default constructor
                feedback (string FeedbackID , int Rating) ; // overloaded constructor
                void assignDetails (string FeedbackID , int Rating) ;
                void printDetails() ;
                void deleteFeedback () ;
                void editFeedback () ;
                ~feedback() ; //destructor
};
```

6.10 feedback.cpp

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

using namespace std ;

feedback::feedback() // default constructor  implementation
{
        feedbackID = "" ;
        rating = 0 ;

}

feedback::feedback (string FeedbackID , int Rating)  // overloaded constructor
implementation
{

        feedbackID = FeedbackID ;
        rating = Rating ;

}

feedback::~feedback() //destructor
{

}

void feedback::assignDetails (string FeedbackID , int Rating)
{
        //method implementation
}
```

```
void feedback::printDetails()
{
        //method implementation
}

void feedback::deleteFeedback ()
{
        //method implementation
}

void feedback::editFeedback ()
{
        //method implementation
}
```

6.11 item.h

```
#include <iostream>
#include <cstring>
#define SIZE 1
#include "feedback.h"
#include "order.h"

using namespace std ;

//class item implementation
class item {
	private:

		order *O[SIZE] ;
		feedback *F[SIZE] ;
		string itemID ;
		string name ;
		string supplierID ;
		string description ;
		float price ;
		int quantity ;

	public:
		item() ; // default constructor
		item(string IID , string Iname ,string SID , string des , float Iprice , int
IQty , string FeedbackID , int Rating ) ; // overloaded constructor
		void assignDetails(string IID , string Iname ,string SID , string des ,
float Iprice , int IQty )  ;
		void addOrders(order *od) ;
		void printDetails () ;
		void addItem () ;
		void deleteItem () ;
		void updateItem () ;
```

```
            ~item() ; //destructor
};
```

6.12 item.cpp

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



using namespace std ;

item::item() // default constructor  implementation
{
        F[0] = new feedback () ;
        itemID = "" ;
        name = "" ;
        supplierID = "" ;
        description = "" ;
        price = 0 ;
        quantity = 0 ;

}

item::item(string IID , string Iname ,string SID , string des , float Iprice , int IQty ,
string FeedbackID , int Rating )  // overloaded constructor implementation
{
        F[0] = new feedback (FeedbackID , Rating ) ;
        itemID = IID ;
        name = Iname ;
        supplierID = SID ;
        description = des ;
        price = Iprice ;
        quantity = IQty ;

}

item::~item() //destructor
{

}

void addOrders(order *od)
{
        //method implementation
}

void item::assignDetails(string IID , string Iname ,string SID , string des , float Iprice
, int IQty )
{
        //method implementation
}

void item::printDetails ()
{
        //method implementation
}

void item::addItem ()
{
        //method implementation
}

void item::deleteItem ()
{
        //method implementation
}
```

```
void item::updateItem ()
{
        //method implementation
}
```

6.13 order.h

```
#include <iostream>
#include <cstring>
#include "payment.h"
#include "delivery.h"
#include "item.h"
#include "user.h"
#define SIZE 10

using namespace std ;

//class order implementation
class order {
        private:
                user *U ;
                delivery *D ;
                payment *P ;
                item *I[SIZE] ;
                string orderID ;
                string date ;
                float sum ;
                float shippingFee ;

        public:
                order() ; // default constructor
```

```
                    // overloaded constructor
                    order(string OID, string Odate , float Osum , float ShFee, string payID
, string PM , float payAmt, string delID , string Ddate , string Dtime  ) ;

                    void assignDetails(string OID, string Odate , float Osum , float ShFee )
;
                    void addItems (item *it) ;
                    void addUser(user *usr) ;
                    void printDetails () ;
                    float calcTotal () ;
                    void deleteOrder () ;
                    void viewOrderStatus () ;
                    ~order() ; //destructor
};
```

6.14 order.cpp

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

using namespace std ;

order::order() // default constructor  implementation
{
          D = new delivery() ;
          P = new payment() ;
          orderID = "" ;
          date = "" ;
          sum = 0 ;
          shippingFee = 0;

}
```

```
// overloaded constructor implementation
order::order(string OID, string Odate , float Osum , float ShFee, string payID , string
PM , float payAmt, string delID , string Ddate , string Dtime  )
{
        D = new delivery (delID , Ddate , Dtime ) ;
        P = new payment(payID , PM , payAmt) ;
        orderID = OID ;
        date = Odate ;
        sum = 0 ;
        shippingFee = 0;

}

order::~order() //destructor
{

}



void order::assignDetails(string OID, string Odate , float Osum , float ShFee )
{
        //method implementation
}

void addItems (item *it)
{
        //method implementation
}

void addUser(user *usr)
{
        //method implementation
}
```

```
void order::printDetails ()
{
        //method implementation
}

float order::calcTotal ()
{
        //method implementation
}

void order::deleteOrder ()
{
        //method implementation
}

void order::viewOrderStatus ()
{
        //method implementation
}
```

6.15 payment.h

```
#include <iostream>
#include <cstring>


using namespace std ;

//class payment implementation
class payment {
	private:
		string paymentID ;
		string paymentMethod ;
		float paymentAmount ;

	public:
		payment() ; // default constructor
		payment(string payID , string PM , float payAmt ) ; // overloaded
constructor
		void assignDetails(string payID , string PM , float payAmt ) ;
		void printDetails () ;
		void confirmPayment () ;
		void authorizePayment () ;
		~payment() ; //destructor
};
```

6.16 payment.cpp

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

using namespace std ;

payment::payment() // default constructor  implementation
{
	paymentID = "" ;
	paymentMethod = "" ;
	paymentAmount = 0 ;

}

payment::payment(string payID , string PM , float payAmt ) // overloaded constructor
implementation
{


	paymentID = payID ;
	paymentMethod = PM ;
	paymentAmount = payAmt ;

}

payment::~payment() //destructor
{

}

void payment::assignDetails(string payID , string PM , float payAmt )
{
	//method implementation
}

void payment::printDetails ()
{
        //method implementation
}

void payment::confirmPayment ()
{
        //method implementation
}

void payment::authorizePayment ()
{
        //method implementation
}
```

6.17 report.h

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

using namespace std ;

//class report implementation
class report
{
	private:
		string reportID ;
		string date ;
		string time ;

	public:
		report() ; // default constructor
		report (string repID, string Rdate, string Rtime ) ; // overloaded
constructor
		void assignDetails(string repID, string Rdate, string Rtime ) ;
		void printDetails () ;
		void generateItemReport (item *ITM) ;
		void generateOrdersReport (order * ORD) ;
		void authorizeReport () ;
		~report() ; //destructor
};
```

6.18 report.cpp

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

using namespace std ;

report::report() // default constructor  implementation
{
        reportID = "" ;
        date = "" ;
        time = "" ;

}

report::report(string repID, string Rdate, string Rtime ) // overloaded constructor
implementation
{

        reportID = repID ;
        date = Rdate ;
        time = Rtime ;

}

report::~report() //destructor
{

}

void assignDetails(string repID, string Rdate, string Rtime )
{
        //method implementation
}

void printDetails ()
{
        //method implementation
}

void generateItemReport (item *ITM)
{
        //method implementation
}

void generateOrdersReport (order * ORD)
{
        //method implementation
}

void authorizeReport ()
{
        //method implementation
}
```

6.19 main.cpp

```
#include <iostream>
#include <cstring>

#include "delivery.h"
#include "feedback.h"
#include "item.h"
#include "order.h"
#include "payment.h"
#include "registeredCustomer.h"
#include "staff.h"
#include "user.h"
#include "report.h"


int main () // main program
{
        delivery del1 ; //object creation using default constructor
        delivery del2 ("DEL0000001" , "2022.05.18" , "08:45") ; //object creation
using overloaded constructor


        feedback feed1 ; //object creation using default constructor
        feedback feed2 ("FB00000001" , 4 ) ; //object creation using overloaded
constructor


        item item1 ; //object creation using default constructor
        //object creation using overloaded constructor
        item *item2 ;
        item2 = new item ("IT00000001", "Cap" , "SUP0000001" , "trendy cap.
suitable for teenagers." , 5.75 , 100,"FB00000001" , 4 ) ;
```

```
        order odr1 ; //object creation using default constructor
        order *odr2 ;
        //object creation using overloaded constructor
        odr2 = new order ("ODR0000001" , "2022.04.30" , 20.50 , 1.99 ,
"PID0000001" , "Card" , 15.35, "DEL0000001" , "2022.05.18" , "08:45") ;


        payment pay1 ; //object creation using default constructor
        payment pay2 ("PID0000001" , "Card" , 15.35) ; //object creation using
overloaded constructor


        registeredCustomer customer1 ; //object creation using default constructor
        //object creation using overloaded constructor
        registeredCustomer customer2 ("Ravindu Dissanayake" , "05 , Galwala road ,
Elpitiya" , "Male" , "0767245612", "Ravi@yahoo.com" , "CUS0000001" ,
"Ravidis#2000") ;


        staff emp1;  //object creation using default constructor
        //object creation using overloaded constructor
        staff emp2 ("Yasas Ranaweera" , "07 , Udupila road , Gampaha" , "Male" ,
"0764337612", "yasas@yahoo.com" , "EMP0000001" , "Yasa@2001#") ;


        user user1 ; //object creation using default constructor
        //object creation using overloaded constructor
        user user2 ("Jeewantha" , "241/5 , kekirideniya road , kaduwela" , "Male" ,
"0719518688" , "jeeewantha123@gmail.com") ;


        report r1; //object creation using default constructor
```

```
	report *r2 = new report ("RPT0000001" , "2022.02.14" , "10:45") ;  //object
creation using overloaded constructor

	return 0 ;
}
```

7 Individual Contribution

Name	Student ID	Identified Classes	Coded Header files	Coded C++ files
De Silva S.J.W. (Leader)	IT21228612	User registeredCustomer staff	User.h registeredCustomer.h staff.h	User.cpp registeredCustomer.cpp staff.cpp main.cpp
Kannangara S.D.R.Y.L.	IT21225710	Order payment	Order.h Payment.h	Order.cpp Payment.cpp main.cpp
Wickramarathne K.H.I.P.	IT21226250	Feedback	Feedback.h	Feedback.cpp main.cpp
Weerage S.W.Y.W.	IT21225260	Report	Report.h	Report.cpp main.cpp
Weerasinghe W.W.A.B.M.	IT21226632	Delivery Item	Delivery.h Item.h	Delivery.cpp Item.cpp main.cpp