Topic : Bus Booking & Scheduling System

Group no : MLB_07.01_11

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**
IT21155666	Perera B.S.M.	0767881169
IT21156724	W.G.P. Shehani	0710151369
IT21156588	S.S. Withanachchi	0767875792
IT20657314	R. Sujeewan	0771693414
IT21157578	A.H.S. Dilsara	0704244072

Object Oriented Concepts

B.sc (Hons) in Information Technology

Exercise - 01

01) User Requirements

1. An Unregistered Guest User can overview the System and access to the Bus Time Table.
2. An online bus booking system requires a Guest user to register by supplying information such as their name, address, NIC and contact number.
3. Then the Registered User allow to log into System using their Credentials (User Name & Password).
4. The Registered user Can Choose the Bus Type, Category, Seat No., Date & Time,
5. After select these details, Registered user has the available busses for the Reservation.
6. Then Registered user can Customize his reservation by his / her opinion.
7. Registered user can select payment method (Credit or debit card) and enters passenger details and payment details.
8. Registered user confirms the Bus Reservation and payment.
9. After the payment is confirmed by Bank and System, Registered User receives ticket number through a message.
10. Registered user gives the Feedbacks
11. Customer care officer log in to the System.
12. Customer care officer reply to the feedbacks.
13. Customer care officer update the User details.
14. Registered User can cancel the Booking if he/she want.
15. Management Assistant Log in to the System.
16. Management Assistant generates the list of Bus Bookings and check the details.
17. System Admin can Add, remove or update route

01) Nouns & Verb Analysis

Nouns - Blue

Verbs - Red

1. An Unregistered Guest User can overview the System and access to the Bus Time Table.
2. An online bus booking system requires a Guest user to register by supplying information such as their name, address, NIC and contact number.
3. Then the Registered User allow to log into System using their Credentials (User Name & Password).
4. The Registered user Can Choose the Bus Type, Category, Seat No., Date & Time,
5. After select these details, Registered user has the available busses for the Reservation.
6. Then Registered user can Customize his reservation by his / her opinion.
7. Registered user can select payment method (Credit or debit card) and enters passenger details and payment details.
8. Registered user confirms the Bus Reservation and payment.
9. After the payment is confirmed by Bank and System, Registered User receives ticket number through a message.
10. Registered user gives the Feedbacks
11. Customer care officer log in to the System.
12. Customer care officer reply to the feedbacks.
13. Customer care officer update the User details.
14. Registered User can cancel the Booking if he/she want.
15. Management Assistant Log in to the System.
16. Management Assistant generates the list of Bus Bookings and check the details.
17. System Admin can Add, remove or update route

Classes

1. Unregistered Guest User
2. Guest user
3. Registered User
4. Feedbacks
5. Management Assistant
6. System Admin
7. Customer care officer
8. Payment
9. Ticket

Redundant

- Unregistered guest user
- Registered customers
- Admins
- Unregistered user
- Suggestions
- Reservations
- Registered customer

Attribute

1. Registered User ID
2. Email
3. Phone number
4. Customer details
5. debit/credit card
6. Prepaid monthly reservations
7. Bus reports
8. Financial reports

CRC Cards

Unregistered Guest User	
Responsibility	Collaborators
Register to the system	
Overview the bus time table	Bus

Registered User	
Responsibility	Collaborators
Login system using credentials	
can choose the bus	Bus
confirm the reservation	Reservation, payments

Management Assistance	
Responsibility	Collaborators
Generates the bus booking report	Generate Report Reservation
check the bus details	Registered user, Bus reservation

Customer care officer	
Responsibility	Collaborators
Log in	
Update the user details	Registered user

Bus Reservation	
Responsibility	Collaborators
Check availability of bus	Bus
Enter passenger details	
Confirm Reservation	payment
Cancel the booking	

Generate report	
Responsibility	Collaborators
List of bus reservation	Bus reservation
Display reservation table	

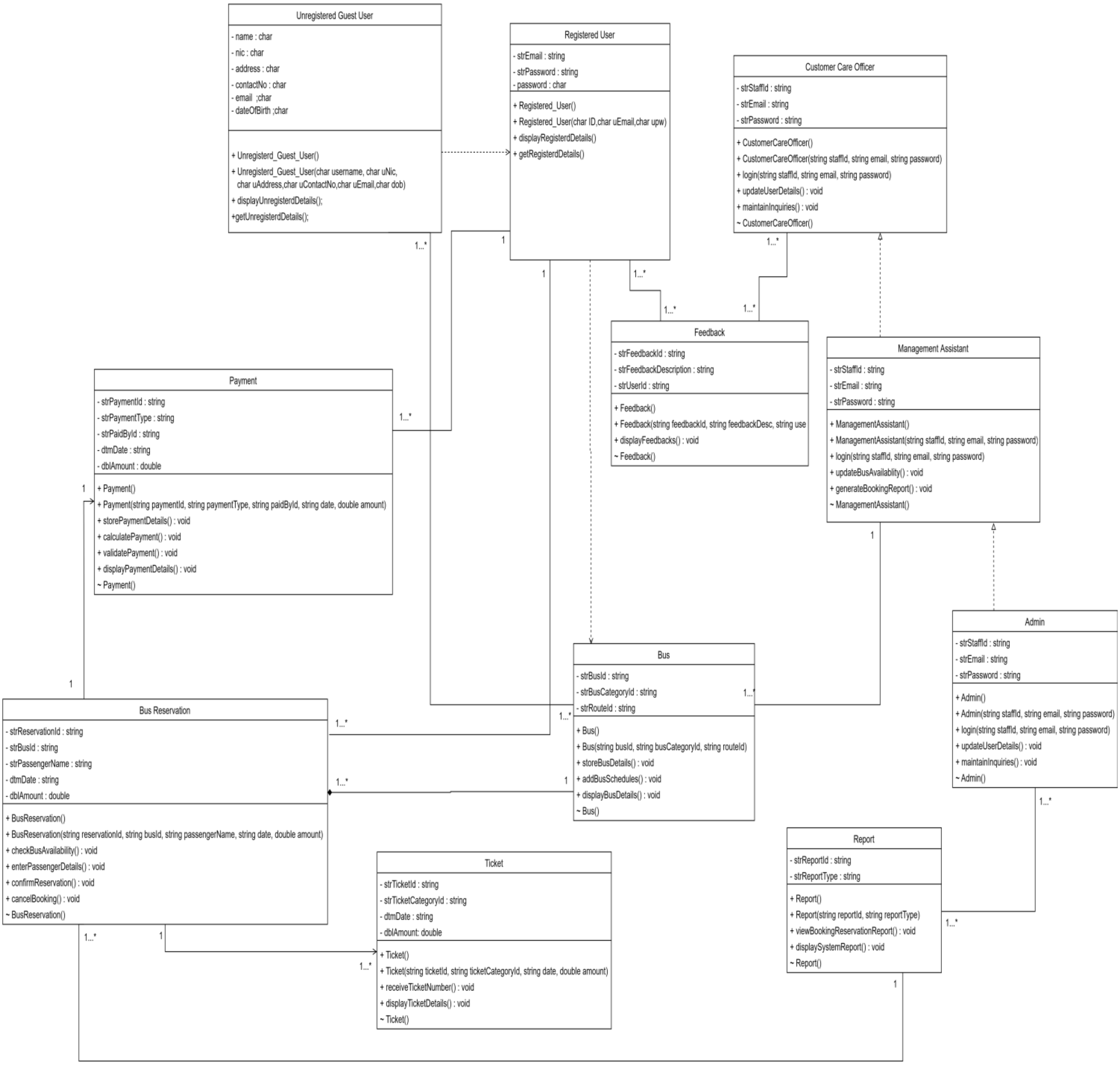
Feedback	
Responsibility	Collaborators
Add Feedback	Customer care officer
Display feedback	

Ticket	
Responsibility	Collaborators
Receives ticket number	
Display tickets	

Payment	
Responsibility	Collaborators
Store Payment details	Registered user
calculate the payment	Reservation
Validate	
Display payment details	

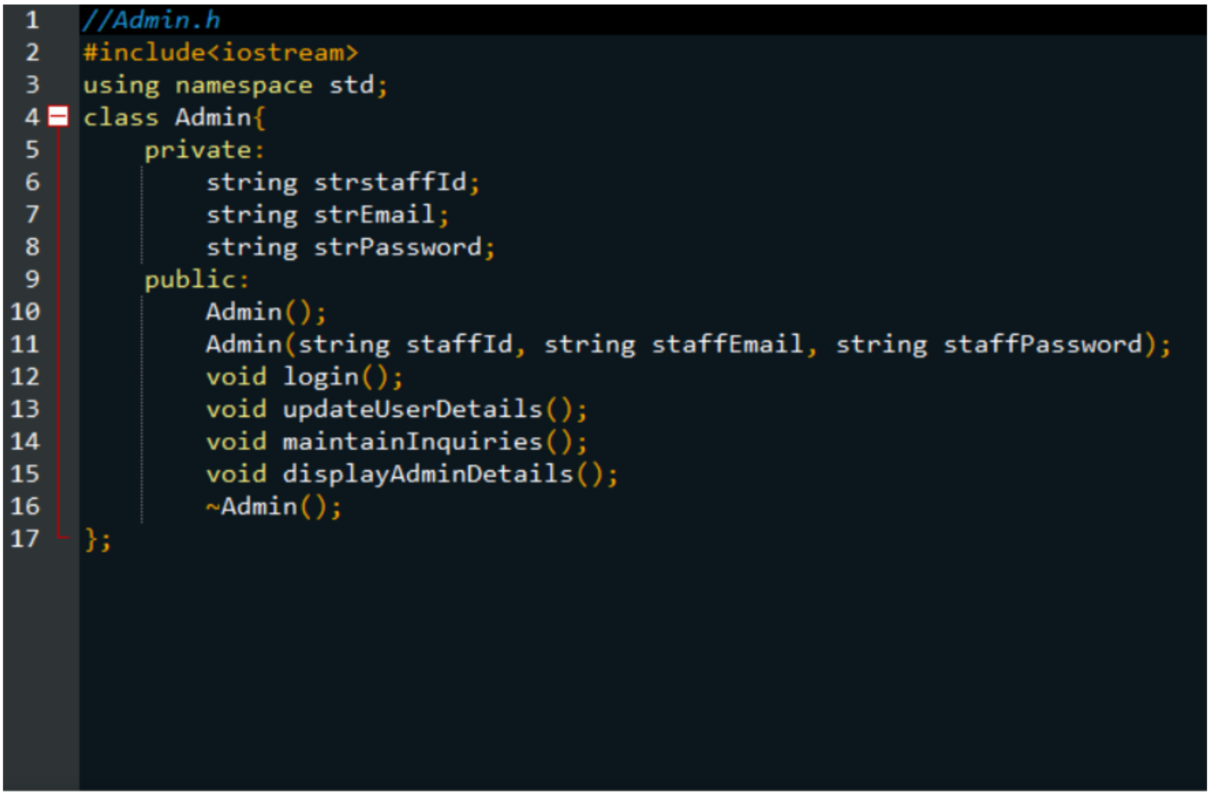
Bus	
Responsibility	Collaborators
Store the bus details	Management Assistant
Add bus schedules	
Display bus details	

Class Diagram


Unregistered Guest User
- name : char
- nic : char
- address : char
- contactNo : char
- email ;char
- dateOfBirth ;char
+ Unregisterd_Guest_User()
+ Unregisterd_Guest_User(char username, char uNic, char uAddress,char uContactNo,char uEmail,char dob)
+ displayUnregisterdDetails();
+getUnregisterdDetails();
Registered User
- strEmail : string
- strPassword : string
- password : char
+ Registered_User()
+ Registered_User(char ID,char uEmail,char upw)
+ displayRegisterdDetails()
+ getRegisterdDetails()
Customer Care Officer
- strStaffId : string
- strEmail : string
- strPassword : string
+ CustomerCareOfficer()
+ CustomerCareOfficer(string staffId, string email, string password)
+ login(string staffId, string email, string password)
+ updateUserDetails() : void
+ maintainInquiries() : void
~ CustomerCareOfficer()
1...*
1
1
1...*
1...*
1...*
1...*
Feedback
- strFeedbackId : string
- strFeedbackDescription : string
- strUserId : string
+ Feedback()
+ Feedback(string feedbackId, string feedbackDesc, string use
+ displayFeedbacks() : void
~ Feedback()
Management Assistant
- strStaffId : string
- strEmail : string
- strPassword : string
+ ManagementAssistant()
+ ManagementAssistant(string staffId, string email, string password)
+ login(string staffId, string email, string password)
+ updateBusAvailablity() : void
+ generateBookingReport() : void
~ ManagementAssistant()
Payment
- strPaymentId : string
- strPaymentType : string
- strPaidById : string
- dtmDate : string
- dblAmount : double
+ Payment()
+ Payment(string paymentId, string paymentType, string paidById, string date, double amount)
+ storePaymentDetails() : void
+ calculatePayment() : void
+ validatePayment() : void
+ displayPaymentDetails() : void
~ Payment()
1...*
1
1
1
Admin
- strStaffId : string
- strEmail : string
- strPassword : string
+ Admin()
+ Admin(string staffId, string email, string password)
+ login(string staffId, string email, string password)
+ updateUserDetails() : void
+ maintainInquiries() : void
~ Admin()
Bus
- strBusId : string
- strBusCategoryId : string
- strRouteId : string
+ Bus()
+ Bus(string busId, string busCategoryId, string routeId)
+ storeBusDetails() : void
+ addBusSchedules() : void
+ displayBusDetails() : void
~ Bus()
1...*
1...*
1
Bus Reservation
- strReservationId : string
- strBusId : string
- strPassengerName : string
- dtmDate : string
- dblAmount : double
+ BusReservation()
+ BusReservation(string reservationId, string busId, string passengerName, string date, double amount)
+ checkBusAvailability() : void
+ enterPassengerDetails() : void
+ confirmReservation() : void
+ cancelBooking() : void
~ BusReservation()
1...*
1...*
1...*
Ticket
- strTicketId : string
- strTicketCategoryId : string
- dtmDate : string
- dblAmount: double
+ Ticket()
+ Ticket(string ticketId, string ticketCategoryId, string date, double amount)
+ receiveTicketNumber() : void
+ displayTicketDetails() : void
~ Ticket()
1...*
1
1...*
Report
- strReportId : string
- strReportType : string
+ Report()
+ Report(string reportId, string reportType)
+ viewBookingReservationReport() : void
+ displaySystemReport() : void
~ Report()
1...*
1...*
1

Codes

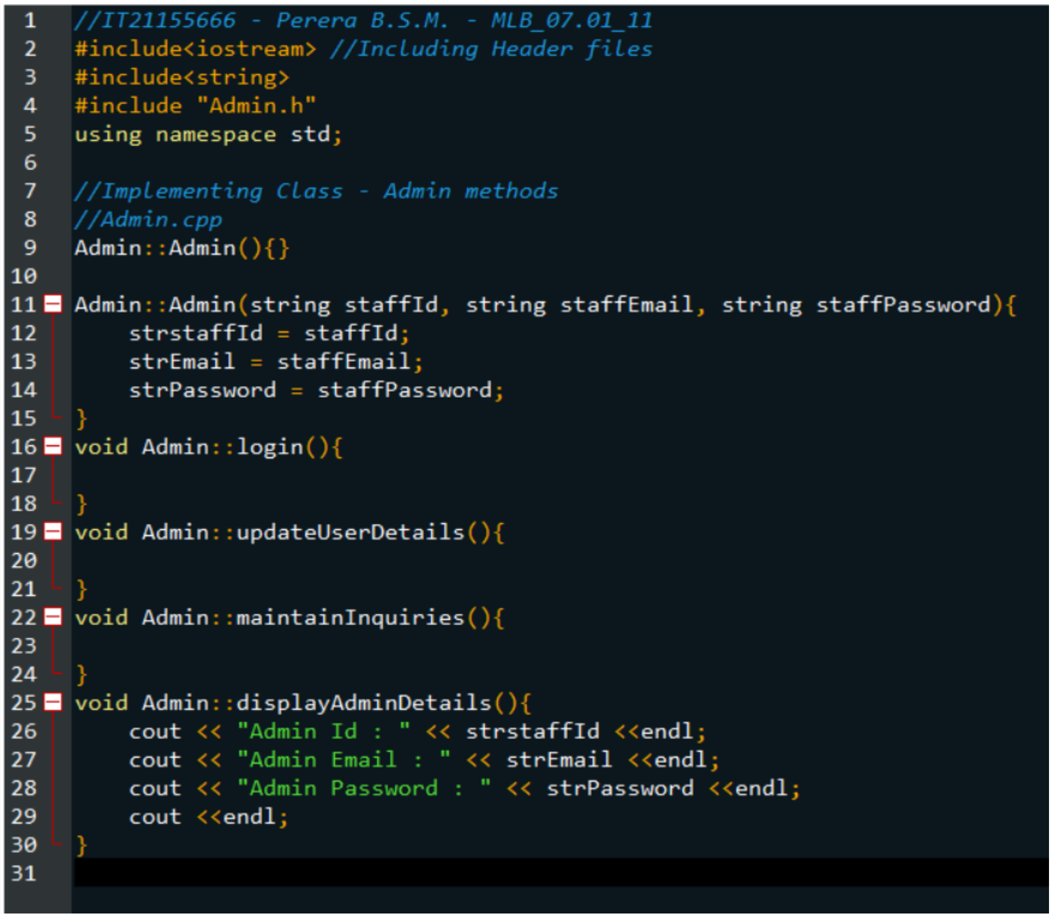
```
//Admin.h
#include<iostream>
using namespace std;
class Admin {
        private:
                string strstaffId;
                string strEmail;
                string strPassword;
        public:
                Admin();
                Admin(string staffId, string staffEmail, string staffPassword);
                void login();
                void updateUserDetails();
                void maintainInquiries();
                void displayAdminDetails();
                ~Admin();
};
```



```
//Admin.h
#include<iostream>
using namespace std;
class Admin{
    private:
        string strstaffId;
        string strEmail;
        string strPassword;
    public:
        Admin();
        Admin(string staffId, string staffEmail, string staffPassword);
        void login();
        void updateUserDetails();
        void maintainInquiries();
        void displayAdminDetails();
        ~Admin();
};
```

//Admin.cpp

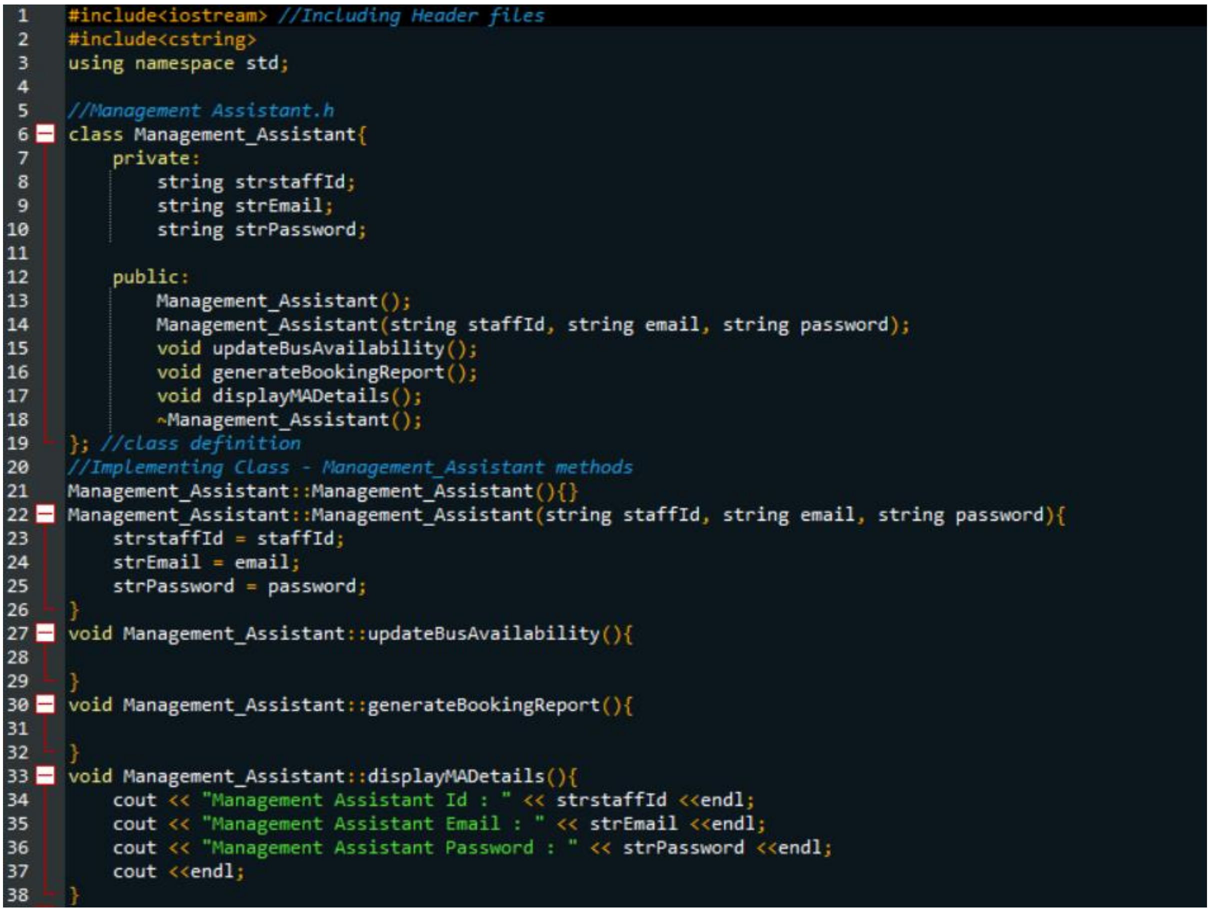
```
#include<iostream> //Including Header files
#include<string>
#include "Admin.h"
using namespace std;

//Implementing Class - Admin methods
//Admin.cpp
Admin::Admin(){}
Admin::Admin(string staffId, string staffEmail, string staffPassword){
                strstaffId = staffId;
                strEmail = staffEmail;
                strPassword = staffPassword;
}
void Admin::login(){}
void Admin::updateUserDetails(){}
void Admin::maintainInquiries(){}
void Admin::displayAdminDetails(){
                cout << "Admin Id : " << strstaffId <<endl;
                cout << "Admin Email : " << strEmail <<endl;
                cout << "Admin Password : " << strPassword <<endl;
                cout <<endl;
}
```



```
//IT21155666 - Perera B.S.M. - MLB_07.01_11
#include<iostream> //Including Header files
#include<string>
#include "Admin.h"
using namespace std;

//Implementing Class - Admin methods
//Admin.cpp
Admin::Admin(){}

Admin::Admin(string staffId, string staffEmail, string staffPassword){
    strstaffId = staffId;
    strEmail = staffEmail;
    strPassword = staffPassword;
}
void Admin::login(){

}
void Admin::updateUserDetails(){

}
void Admin::maintainInquiries(){

}
void Admin::displayAdminDetails(){
    cout << "Admin Id : " << strstaffId <<endl;
    cout << "Admin Email : " << strEmail <<endl;
    cout << "Admin Password : " << strPassword <<endl;
    cout <<endl;
}

```

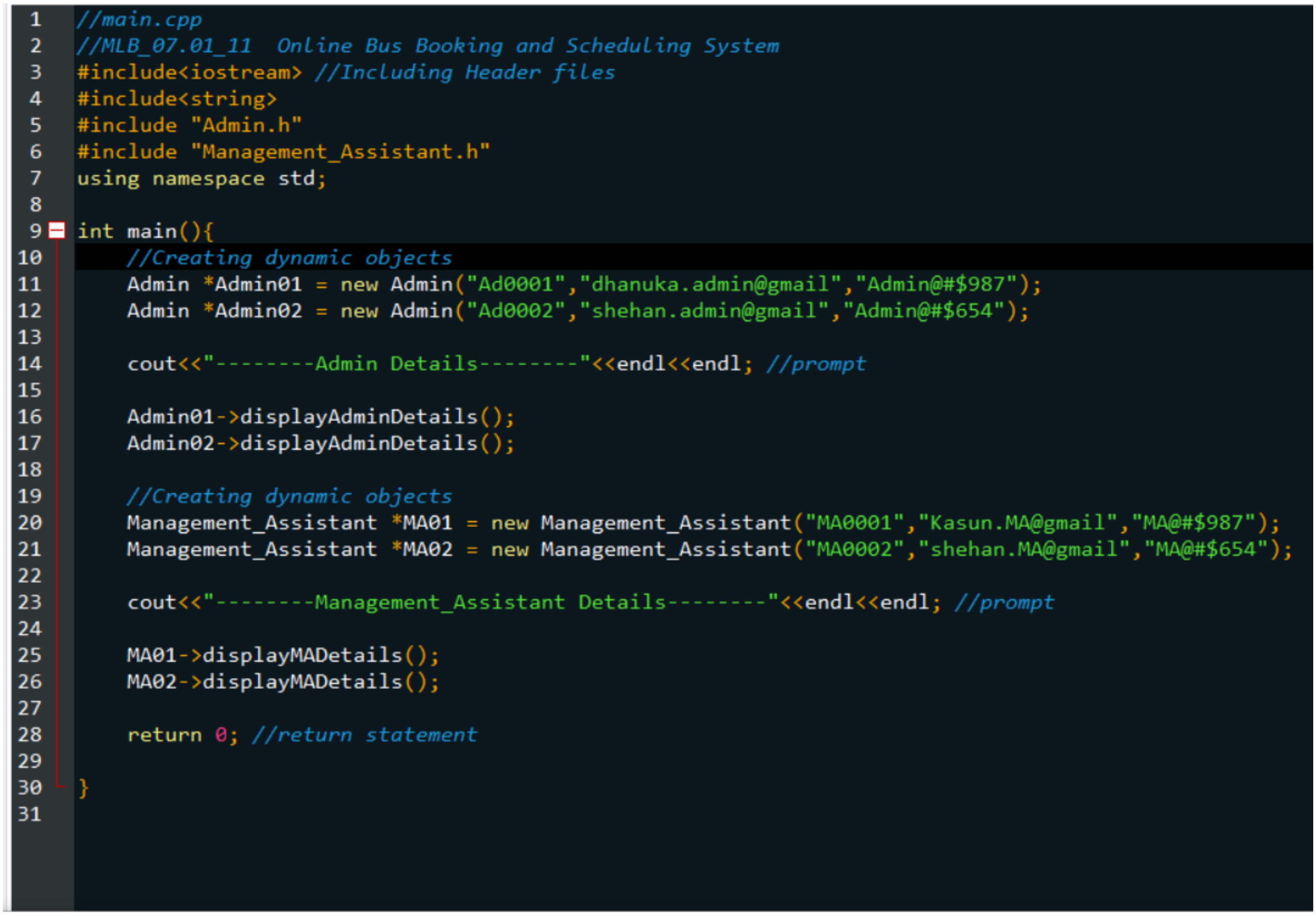
//main.cpp

```
#include<iostream> //Including Header files
#include<string>
#include "Admin.h"
#include "Management_Assistant.h"
using namespace std;

int main(){
	//Creating dynamic objects
	Admin *Admin01 = new Admin("Ad0001","dhanuka.admin@gmail","Admin@#$987");
	Admin *Admin02 = new Admin("Ad0002","shehan.admin@gmail","Admin@#$654");

	cout<<"--------Admin Details--------"<<endl<<endl; //prompt

	Admin01->displayAdminDetails();
	Admin02->displayAdminDetails();

	//Creating dynamic objects
	Management_Assistant *MA01 = new Management_Assistant("MA0001","Kasun.MA@gmail","MA@#$987");
	Management_Assistant *MA02 = new Management_Assistant("MA0002","shehan.MA@gmail","MA@#$654");

	cout<<"--------Management_Assistant Details--------"<<endl<<endl; //prompt

	MA01->displayMADetails();
	MA02->displayMADetails();

	return 0; //return statement

}
```



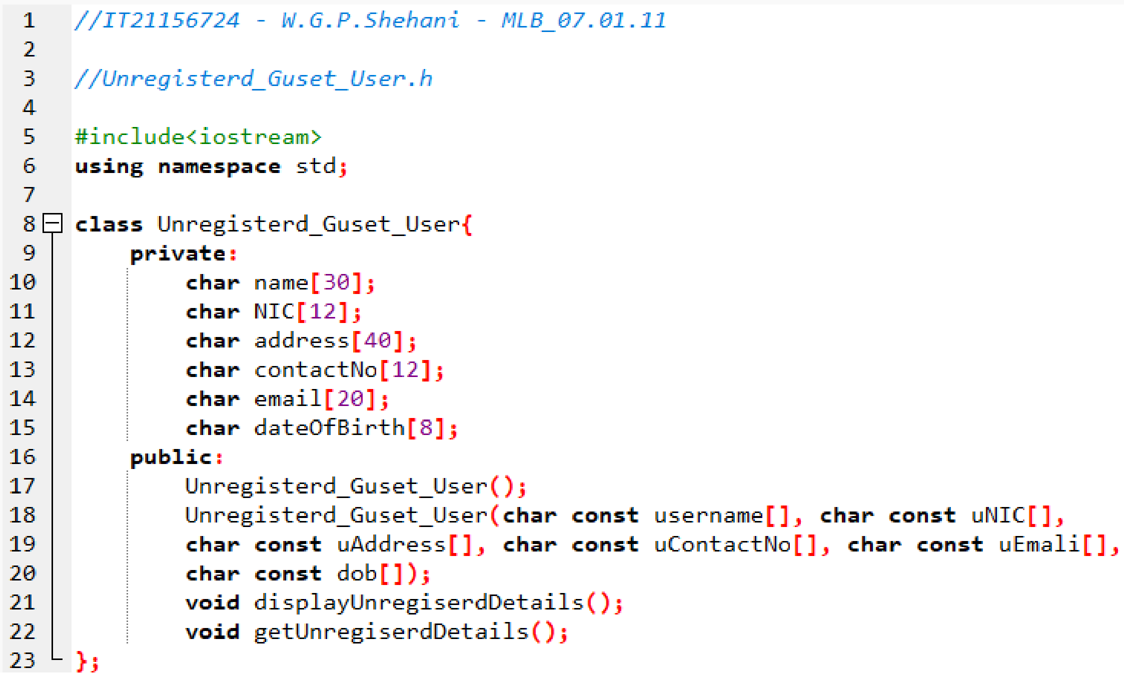
```cpp
#include<iostream> //Including Header files
#include<cstring>
using namespace std;

//Management Assistant.h
class Management_Assistant{
    private:
        string strstaffId;
        string strEmail;
        string strPassword;

    public:
        Management_Assistant();
        Management_Assistant(string staffId, string email, string password);
        void updateBusAvailability();
        void generateBookingReport();
        void displayMADetails();
        ~Management_Assistant();
}; //class definition
//Implementing Class - Management_Assistant methods
Management_Assistant::Management_Assistant(){}
Management_Assistant::Management_Assistant(string staffId, string email, string password){
    strstaffId = staffId;
    strEmail = email;
    strPassword = password;
}
void Management_Assistant::updateBusAvailability(){

}
void Management_Assistant::generateBookingReport(){

}
void Management_Assistant::displayMADetails(){
    cout << "Management Assistant Id : " << strstaffId <<endl;
    cout << "Management Assistant Email : " << strEmail <<endl;
    cout << "Management Assistant Password : " << strPassword <<endl;
    cout <<endl;
}
```

//main.cpp



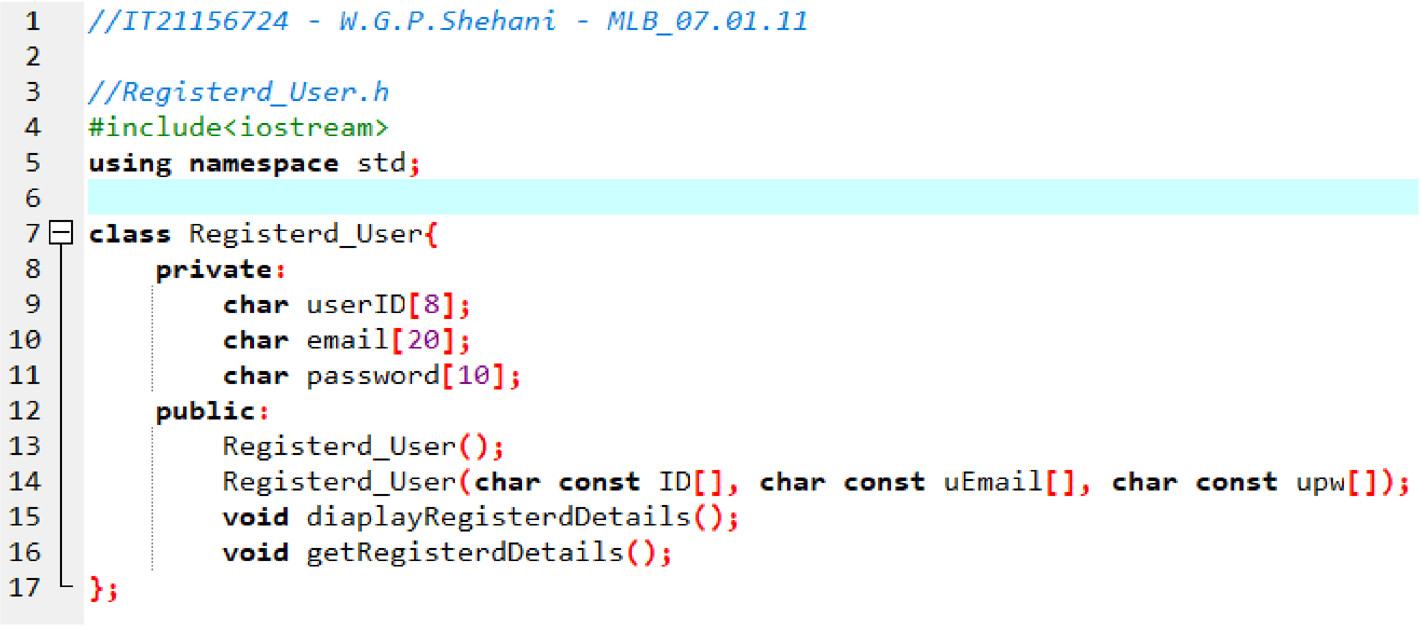
```cpp
//main.cpp
//MLB_07.01_11  Online Bus Booking and Scheduling System
#include<iostream> //Including Header files
#include<string>
#include "Admin.h"
#include "Management_Assistant.h"
using namespace std;

int main(){
    //Creating dynamic objects
    Admin *Admin01 = new Admin("Ad0001","dhanuka.admin@gmail","Admin@#$987");
    Admin *Admin02 = new Admin("Ad0002","shehan.admin@gmail","Admin@#$654");

    cout<<"--------Admin Details--------"<<endl<<endl; //prompt

    Admin01->displayAdminDetails();
    Admin02->displayAdminDetails();

    //Creating dynamic objects
    Management_Assistant *MA01 = new Management_Assistant("MA0001","Kasun.MA@gmail","MA@#$987");
    Management_Assistant *MA02 = new Management_Assistant("MA0002","shehan.MA@gmail","MA@#$654");

    cout<<"--------Management_Assistant Details--------"<<endl<<endl; //prompt

    MA01->displayMADetails();
    MA02->displayMADetails();

    return 0; //return statement

}

```

```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Unregisterd_Guset_User.h

#include<iostream>

using namespace std;

class Unregisterd_Guset_User{

        private:

            char name[30];

            char NIC[12];

            char address[40];

            char contactNo[12];

            char email[20];

            char dateOfBirth[8];

        public:

            Unregisterd_Guset_User();

            Unregisterd_Guset_User(char const username[], char const uNIC[], char const uAddress[], char const uContactNo[], char const uEmali[], char const dob[]);

            void displayUnregiserdDetails();void getUnregiserdDetails();

};
```



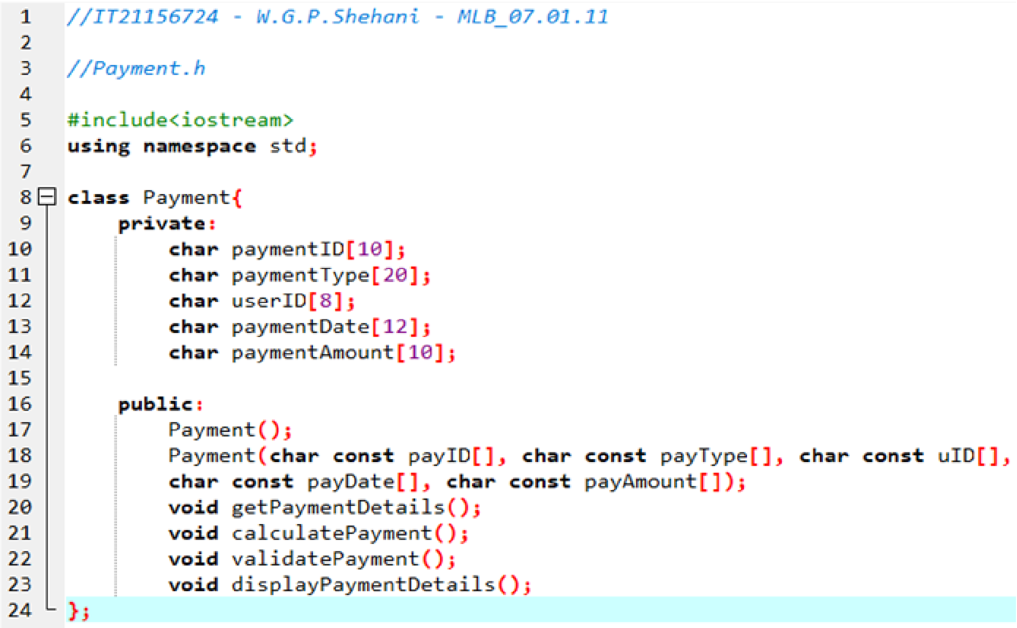
```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Unregisterd_Guset_User.h

#include<iostream>
using namespace std;

class Unregisterd_Guset_User{
    private:
        char name[30];
        char NIC[12];
        char address[40];
        char contactNo[12];
        char email[20];
        char dateOfBirth[8];
    public:
        Unregisterd_Guset_User();
        Unregisterd_Guset_User(char const username[], char const uNIC[],
        char const uAddress[], char const uContactNo[], char const uEmali[],
        char const dob[]);
        void displayUnregiserdDetails();
        void getUnregiserdDetails();
};
```

```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Registerd_User.h

#include<iostream>

using namespace std;


class Registerd_User{

            private:

                char userID[8];

                char email[20];

                char password[10];

            public:

                Registerd_User();

              Registerd_User(char const ID[], char const uEmail[], char const upw[]);

                void diaplayRegisterdDetails();

                void getRegisterdDetails();

};
```



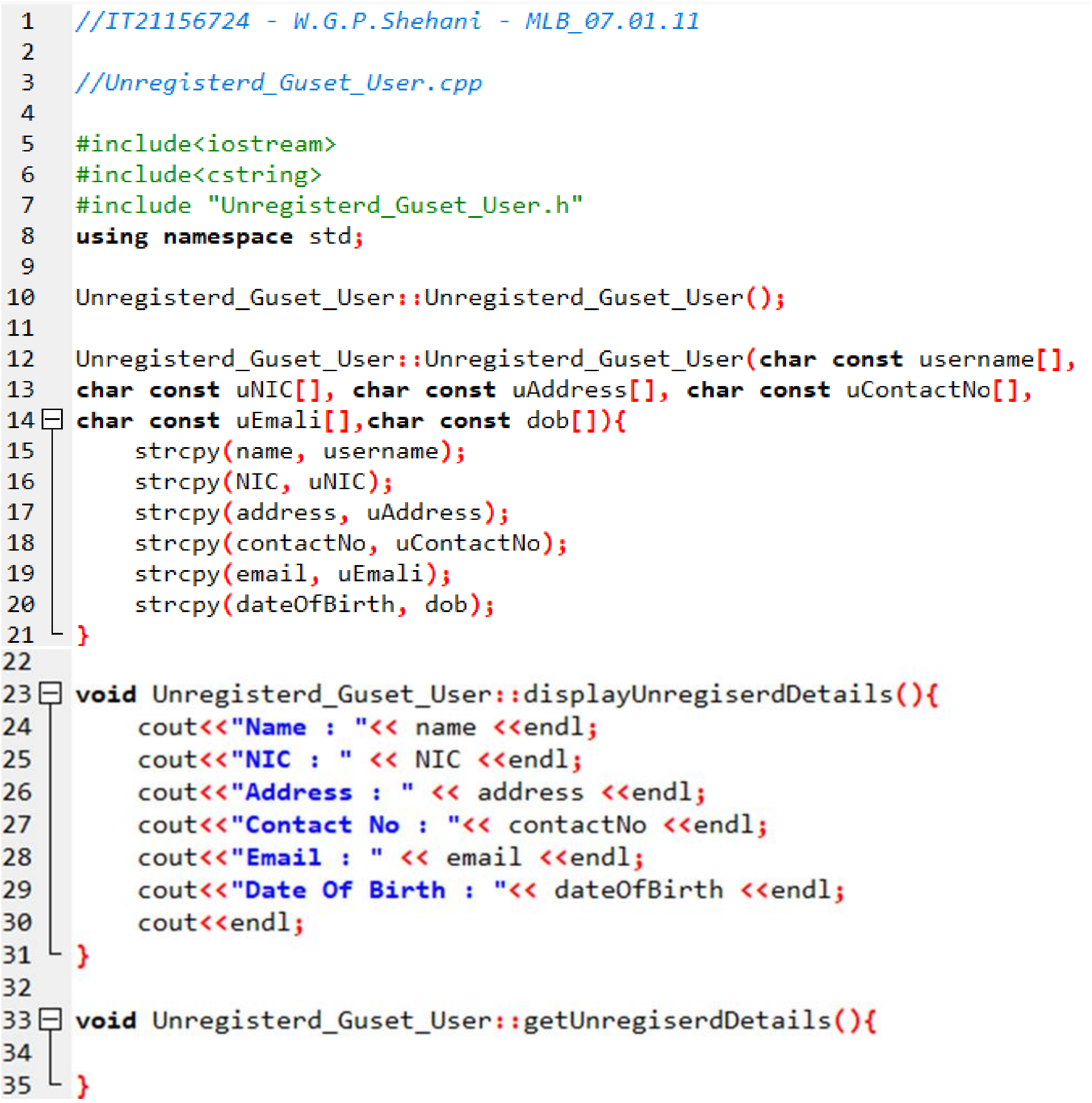
```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Registerd_User.h
#include<iostream>
using namespace std;

class Registerd_User{
    private:
        char userID[8];
        char email[20];
        char password[10];
    public:
        Registerd_User();
        Registerd_User(char const ID[], char const uEmail[], char const upw[]);
        void diaplayRegisterdDetails();
        void getRegisterdDetails();
};
```

```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Payment.h

#include<iostream>

using namespace std;

class Payment{

            private:

                char paymentID[10];

                char paymentType[20];

                char userID[8];

                char paymentDate[12];

                char paymentAmount[10];

            public:

                Payment();

              Payment(char const payID[], char const payType[], char const uID[], char const
payDate[], char const payAmount[]);

                void getPaymentDetails();

                void calculatePayment();

                void validatePayment();

                void displayPaymentDetails();

};
```



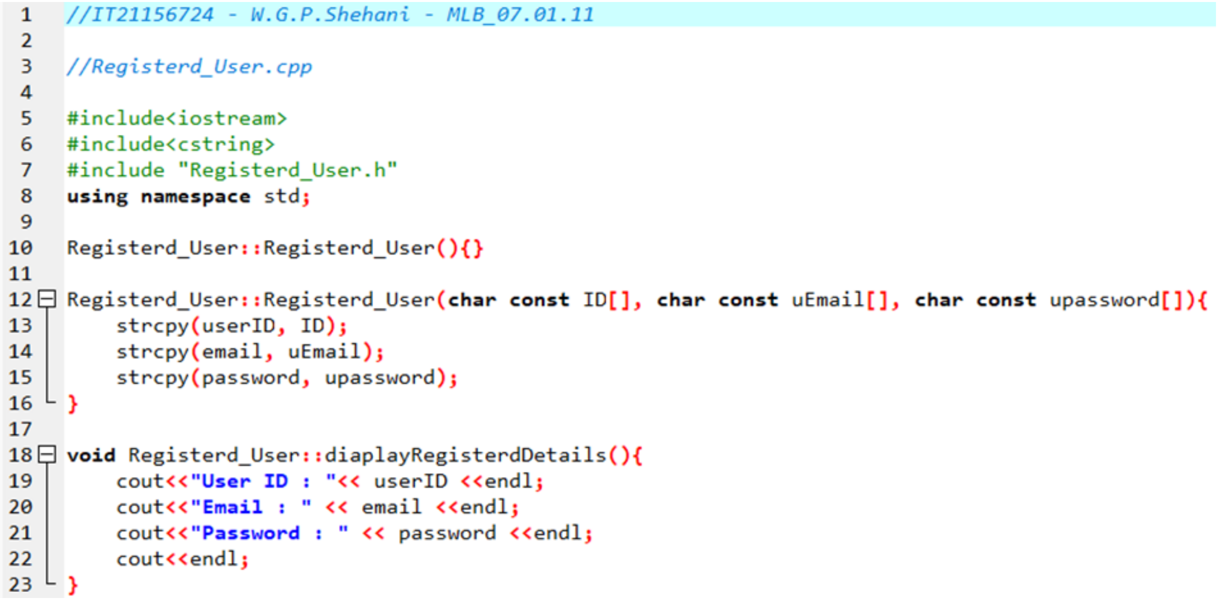
```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Payment.h

#include<iostream>
using namespace std;

class Payment{
    private:
        char paymentID[10];
        char paymentType[20];
        char userID[8];
        char paymentDate[12];
        char paymentAmount[10];

    public:
        Payment();
        Payment(char const payID[], char const payType[], char const uID[],
        char const payDate[], char const payAmount[]);
        void getPaymentDetails();
        void calculatePayment();
        void validatePayment();
        void displayPaymentDetails();
};
```

```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11
//Unregisterd_Guset_User.cpp
#include<iostream>
#include<cstring>
#include "Unregisterd_Guset_User.h"
using namespace std;


Unregisterd_Guset_User::Unregisterd_Guset_User();
Unregisterd_Guset_User::Unregisterd_Guset_User(char const username[], char const uNIC[], char const uAddress[], char const uContactNo[], char const uEmali[],char const dob[]){
            strcpy(name, username);
            strcpy(NIC, uNIC);
            strcpy(address, uAddress);
            strcpy(contactNo, uContactNo);
            strcpy(email, uEmali);
            strcpy(dateOfBirth, dob);
}


void Unregisterd_Guset_User::displayUnregiserdDetails(){
            cout<<"Name : "<< name <<endl;
            cout<<"NIC : " << NIC <<endl;
            cout<<"Address : " << address <<endl;
            cout<<"Contact No : "<< contactNo <<endl;
            cout<<"Email : " << email <<endl;
            cout<<"Date Of Birth : "<< dateOfBirth <<endl;
            cout<<endl;
}
void Unregisterd_Guset_User::getUnregiserdDetails(){


}
```



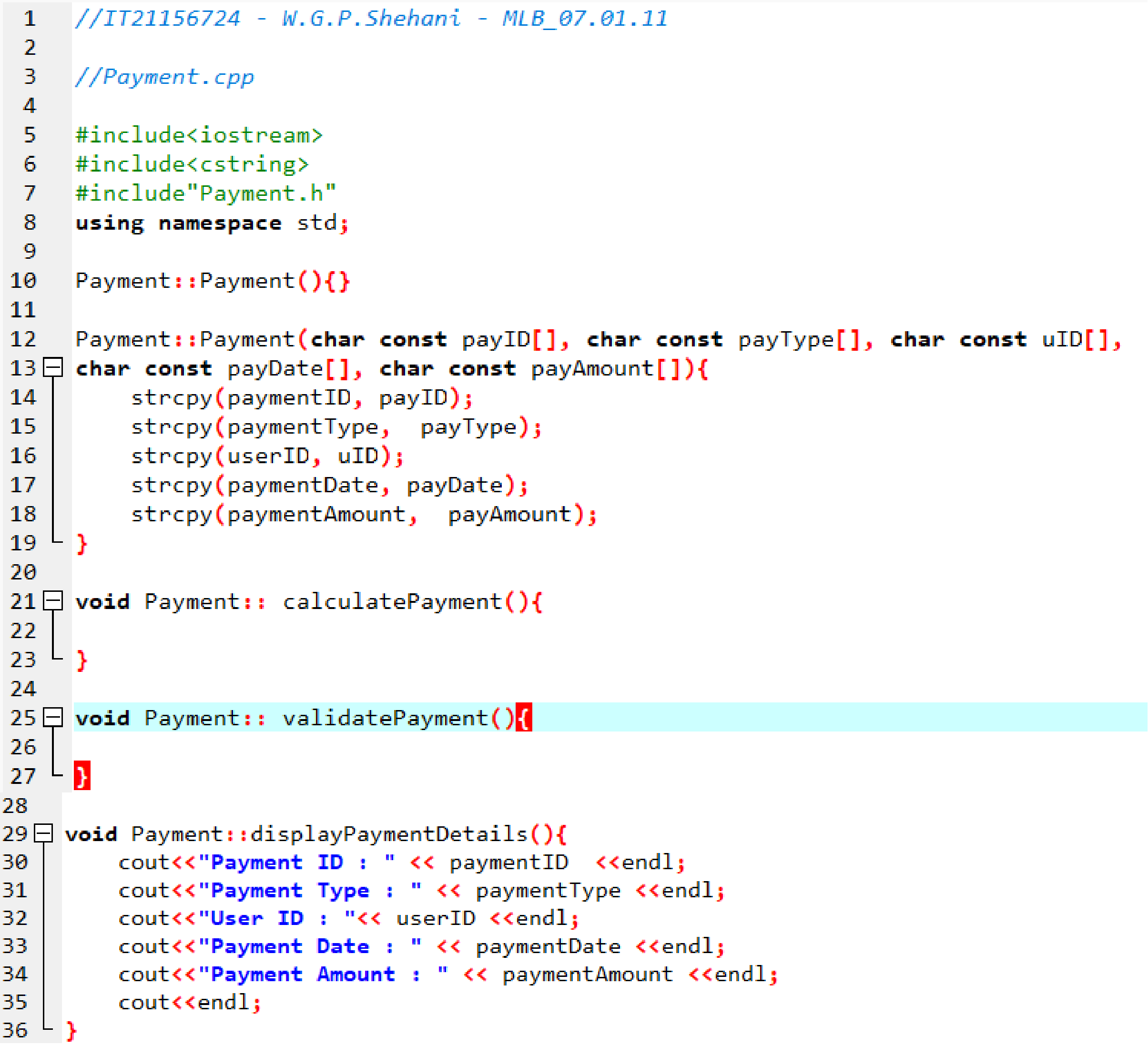
```cpp
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Unregisterd_Guset_User.cpp

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

Unregisterd_Guset_User::Unregisterd_Guset_User();

Unregisterd_Guset_User::Unregisterd_Guset_User(char const username[],
char const uNIC[], char const uAddress[], char const uContactNo[],
char const uEmali[],char const dob[]){
    strcpy(name, username);
    strcpy(NIC, uNIC);
    strcpy(address, uAddress);
    strcpy(contactNo, uContactNo);
    strcpy(email, uEmali);
    strcpy(dateOfBirth, dob);
}

void Unregisterd_Guset_User::displayUnregiserdDetails(){
    cout<<"Name : "<< name <<endl;
    cout<<"NIC : " << NIC <<endl;
    cout<<"Address : " << address <<endl;
    cout<<"Contact No : "<< contactNo <<endl;
    cout<<"Email : " << email <<endl;
    cout<<"Date Of Birth : "<< dateOfBirth <<endl;
    cout<<endl;
}

void Unregisterd_Guset_User::getUnregiserdDetails(){

}
```

```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Registerd_User.cpp

#include<iostream>

#include<cstring>

#include "Registerd_User.h"

using namespace std;


Registerd_User::Registerd_User(){}

Registerd_User::Registerd_User(char const ID[], char const uEmail[], char const upassword[]){

                strcpy(userID, ID);

                 strcpy(email, uEmail);

                 strcpy(password, upassword);

}

void Registerd_User::diaplayRegisterdDetails(){

                 cout<<"User ID : "<< userID <<endl;

                 cout<<"Email : " << email <<endl;

                 cout<<"Password : " << password <<endl;

                 cout<<endl;

}
```



```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Registerd_User.cpp

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

Registerd_User::Registerd_User(){}

Registerd_User::Registerd_User(char const ID[], char const uEmail[], char const upassword[]){
    strcpy(userID, ID);
    strcpy(email, uEmail);
    strcpy(password, upassword);
}

void Registerd_User::diaplayRegisterdDetails(){
    cout<<"User ID : "<< userID <<endl;
    cout<<"Email : " << email <<endl;
    cout<<"Password : " << password <<endl;
    cout<<endl;
}
```

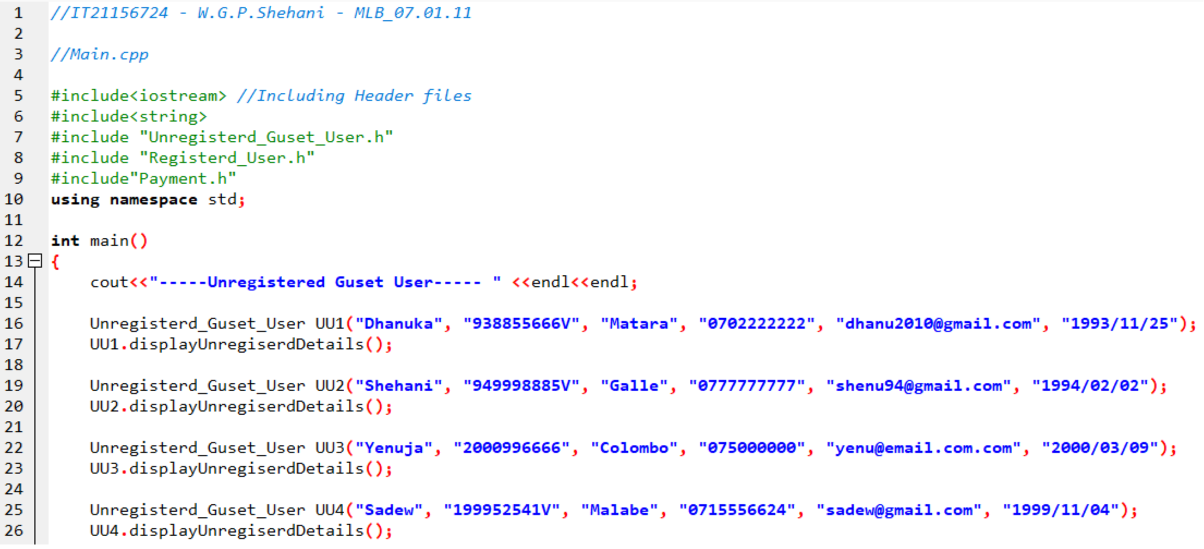
```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11
//Payment.cpp
#include<iostream>
#include<cstring>
#include"Payment.h"
using namespace std;

Payment::Payment(){}
Payment::Payment(char const payID[], char const payType[], char const uID[],
char const payDate[], char const payAmount[]){
   strcpy(paymentID, payID);
                strcpy(paymentType,  payType);
                strcpy(userID, uID);
                strcpy(paymentDate, payDate);
                strcpy(paymentAmount,  payAmount);
}
void Payment:: calculatePayment(){

}
void Payment:: validatePayment(){

}
void Payment::displayPaymentDetails(){
                cout<<"Payment ID : " << paymentID  <<endl;
   cout<<"Payment Type : " << paymentType <<endl;
                cout<<"User ID : "<< userID <<endl;
                cout<<"Payment Date : " << paymentDate <<endl;
   cout<<"Payment Amount : " << paymentAmount <<endl;
   cout<<endl;
}
```



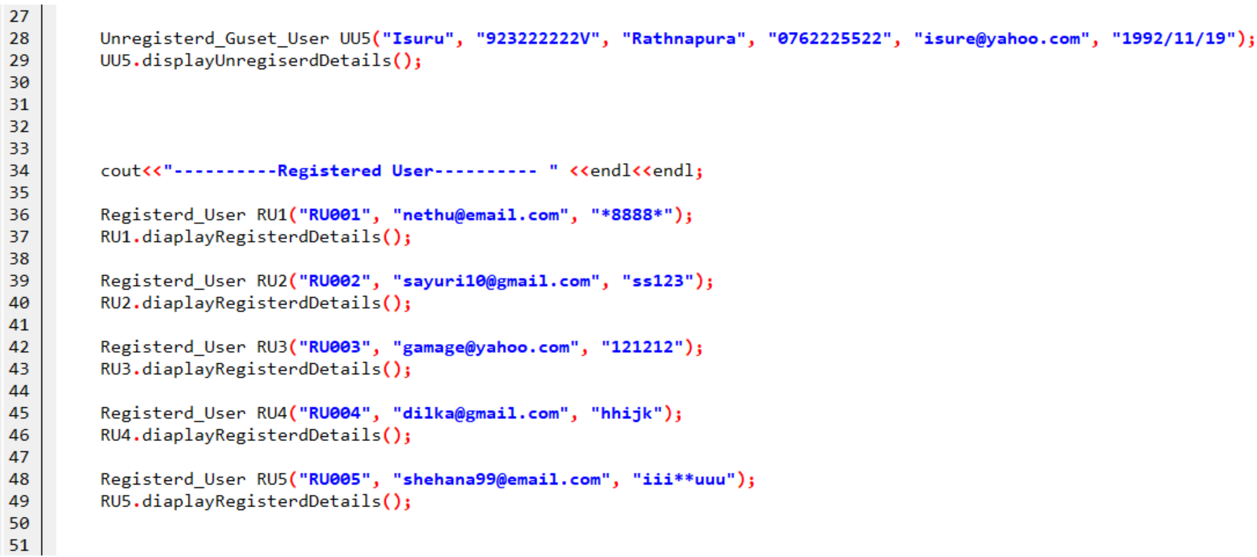
```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Payment.cpp

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

Payment::Payment(){}

Payment::Payment(char const payID[], char const payType[], char const uID[],
char const payDate[], char const payAmount[]){
    strcpy(paymentID, payID);
    strcpy(paymentType,  payType);
    strcpy(userID, uID);
    strcpy(paymentDate, payDate);
    strcpy(paymentAmount,  payAmount);
}

void Payment:: calculatePayment(){

}

void Payment:: validatePayment(){

}

void Payment::displayPaymentDetails(){
    cout<<"Payment ID : " << paymentID  <<endl;
    cout<<"Payment Type : " << paymentType <<endl;
    cout<<"User ID : "<< userID <<endl;
    cout<<"Payment Date : " << paymentDate <<endl;
    cout<<"Payment Amount : " << paymentAmount <<endl;
    cout<<endl;
}
```

```
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Main.cpp

#include<iostream> //Including Header files

#include<string>

#include "Unregisterd_Guset_User.h"

#include "Registerd_User.h"

#include"Payment.h"

using namespace std;


int main()

{

            cout<<"-----Unregistered Guset User----- " <<endl<<endl;


            Unregisterd_Guset_User UU1("Dhanuka", "938855666V", "Matara", "0702222222",
"dhanu2010@gmail.com", "1993/11/25");

  UU1.displayUnregiserdDetails();

             Unregisterd_Guset_User UU2("Shehani", "949998885V", "Galle", "0777777777",
"shenu94@gmail.com", "1994/02/02");

  UU2.displayUnregiserdDetails();

             Unregisterd_Guset_User UU3("Yenuja", "2000996666", "Colombo", "075000000",
"yenu@email.com.com", "2000/03/09");

  UU3.displayUnregiserdDetails();

             Unregisterd_Guset_User UU4("Sadew", "199952541V", "Malabe", "0715556624",
"sadew@gmail.com", "1999/11/04");

  UU4.displayUnregiserdDetails();

             Unregisterd_Guset_User UU5("Isuru", "923222222V", "Rathnapura", "0762225522",
"isure@yahoo.com", "1992/11/19");

  UU5.displayUnregiserdDetails();
```

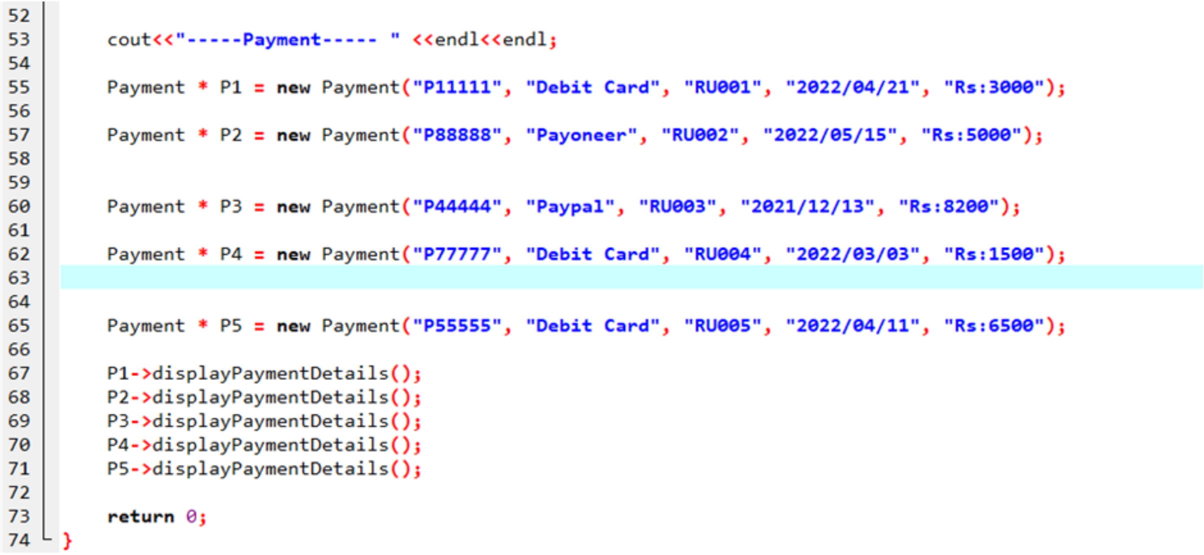
```
   cout<<"----------Registered User---------- " <<endl<<endl;
                        Registerd_User RU1("RU001", "nethu@email.com", "*8888*");
   RU1.diaplayRegisterdDetails();
                        Registerd_User RU2("RU002", "sayuri10@gmail.com", "ss123");
   RU2.diaplayRegisterdDetails();
                        Registerd_User RU3("RU003", "gamage@yahoo.com", "121212");
   RU3.diaplayRegisterdDetails();
                        Registerd_User RU4("RU004", "dilka@gmail.com", "hhijk");
   RU4.diaplayRegisterdDetails();
                        Registerd_User RU5("RU005", "shehana99@email.com", "iii**uuu");
   RU5.diaplayRegisterdDetails();



   cout<<"-----Payment----- " <<endl<<endl;
Payment * P1 = new Payment("P11111", "Debit Card", "RU001", "2022/04/21", "Rs:3000");
Payment * P2 = new Payment("P88888", "Payoneer", "RU002", "2022/05/15", "Rs:5000");
Payment * P3 = new Payment("P44444", "Paypal", "RU003", "2021/12/13", "Rs:8200");
Payment * P4 = new Payment("P77777", "Debit Card", "RU004", "2022/03/03", "Rs:1500");
Payment * P5 = new Payment("P55555", "Debit Card", "RU005", "2022/04/11", "Rs:6500");

                  P1->displayPaymentDetails();
                  P2->displayPaymentDetails();
                  P3->displayPaymentDetails();
                  P4->displayPaymentDetails();
                  P5->displayPaymentDetails();

   return 0;
}
```



```cpp
//IT21156724 - W.G.P.Shehani - MLB_07.01.11

//Main.cpp

#include<iostream> //Including Header files
#include<string>
#include "Unregisterd_Guset_User.h"
#include "Registerd_User.h"
#include"Payment.h"
using namespace std;

int main()
{
    cout<<"-----Unregistered Guset User----- " <<endl<<endl;

    Unregisterd_Guset_User UU1("Dhanuka", "938855666V", "Matara", "0702222222", "dhanu2010@gmail.com", "1993/11/25");
    UU1.displayUnregiserdDetails();

    Unregisterd_Guset_User UU2("Shehani", "949998885V", "Galle", "0777777777", "shenu94@gmail.com", "1994/02/02");
    UU2.displayUnregiserdDetails();

    Unregisterd_Guset_User UU3("Yenuja", "2000996666", "Colombo", "075000000", "yenu@email.com.com", "2000/03/09");
    UU3.displayUnregiserdDetails();

    Unregisterd_Guset_User UU4("Sadew", "199952541V", "Malabe", "0715556624", "sadew@gmail.com", "1999/11/04");
    UU4.displayUnregiserdDetails();
```

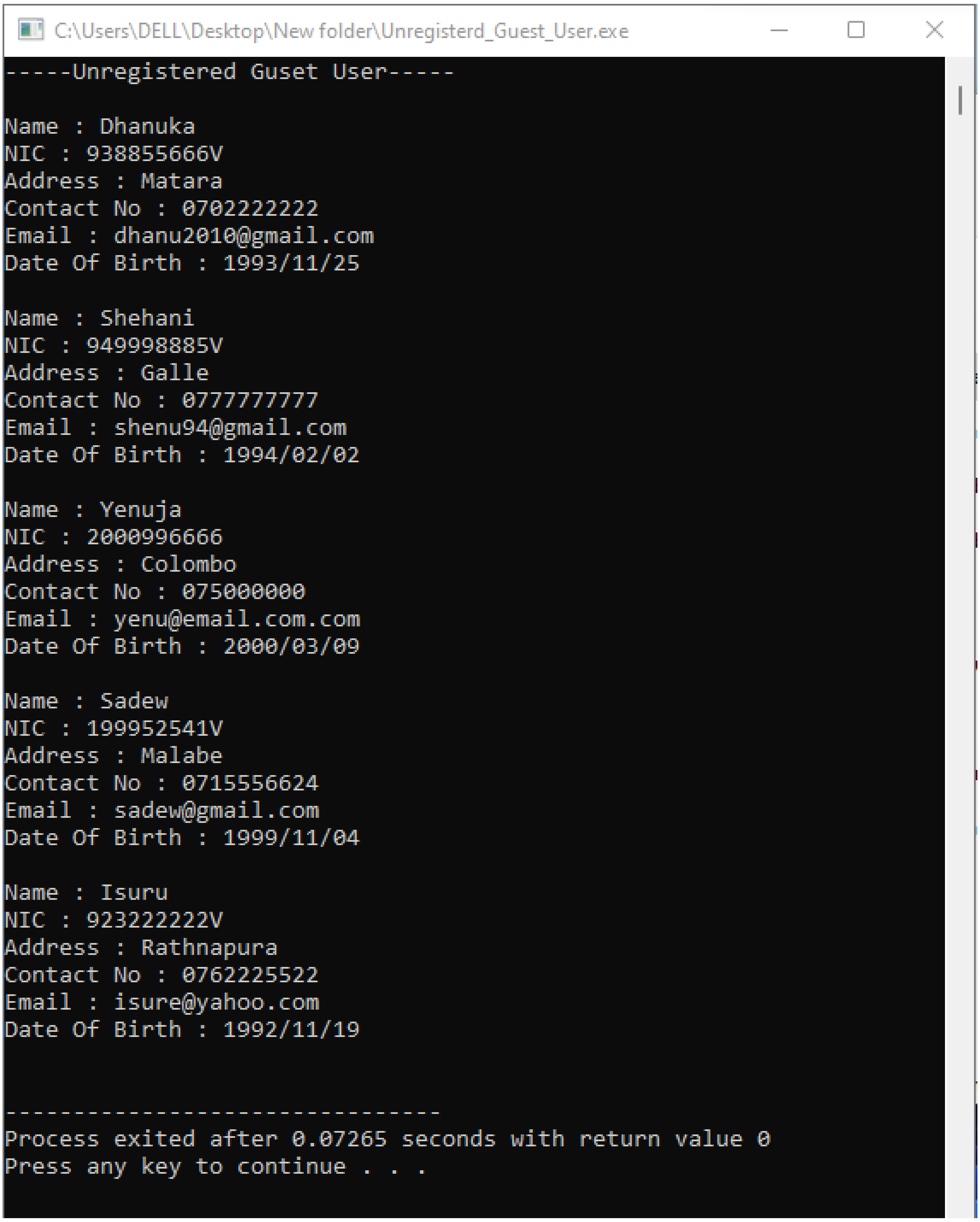


```cpp

    Unregisterd_Guset_User UU5("Isuru", "923222222V", "Rathnapura", "0762225522", "isure@yahoo.com", "1992/11/19");
    UU5.displayUnregiserdDetails();




    cout<<"----------Registered User---------- " <<endl<<endl;

    Registerd_User RU1("RU001", "nethu@email.com", "*8888*");
    RU1.diaplayRegisterdDetails();

    Registerd_User RU2("RU002", "sayuri10@gmail.com", "ss123");
    RU2.diaplayRegisterdDetails();

    Registerd_User RU3("RU003", "gamage@yahoo.com", "121212");
    RU3.diaplayRegisterdDetails();

    Registerd_User RU4("RU004", "dilka@gmail.com", "hhijk");
    RU4.diaplayRegisterdDetails();

    Registerd_User RU5("RU005", "shehana99@email.com", "iii**uuu");
    RU5.diaplayRegisterdDetails();

```



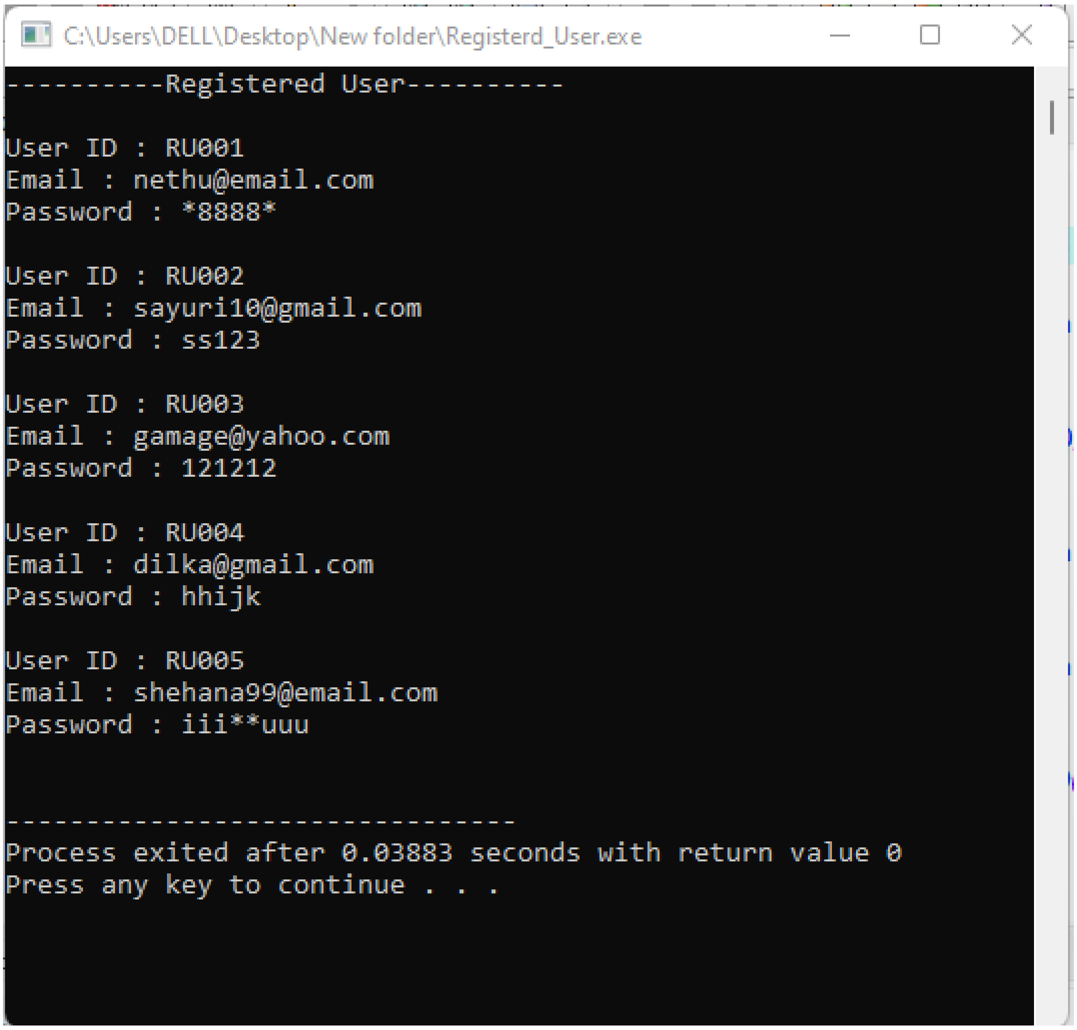
```cpp

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

    Payment * P1 = new Payment("P11111", "Debit Card", "RU001", "2022/04/21", "Rs:3000");

    Payment * P2 = new Payment("P88888", "Payoneer", "RU002", "2022/05/15", "Rs:5000");


    Payment * P3 = new Payment("P44444", "Paypal", "RU003", "2021/12/13", "Rs:8200");

    Payment * P4 = new Payment("P77777", "Debit Card", "RU004", "2022/03/03", "Rs:1500");


    Payment * P5 = new Payment("P55555", "Debit Card", "RU005", "2022/04/11", "Rs:6500");

    P1->displayPaymentDetails();
    P2->displayPaymentDetails();
    P3->displayPaymentDetails();
    P4->displayPaymentDetails();
    P5->displayPaymentDetails();

    return 0;
}
```

Sample Output

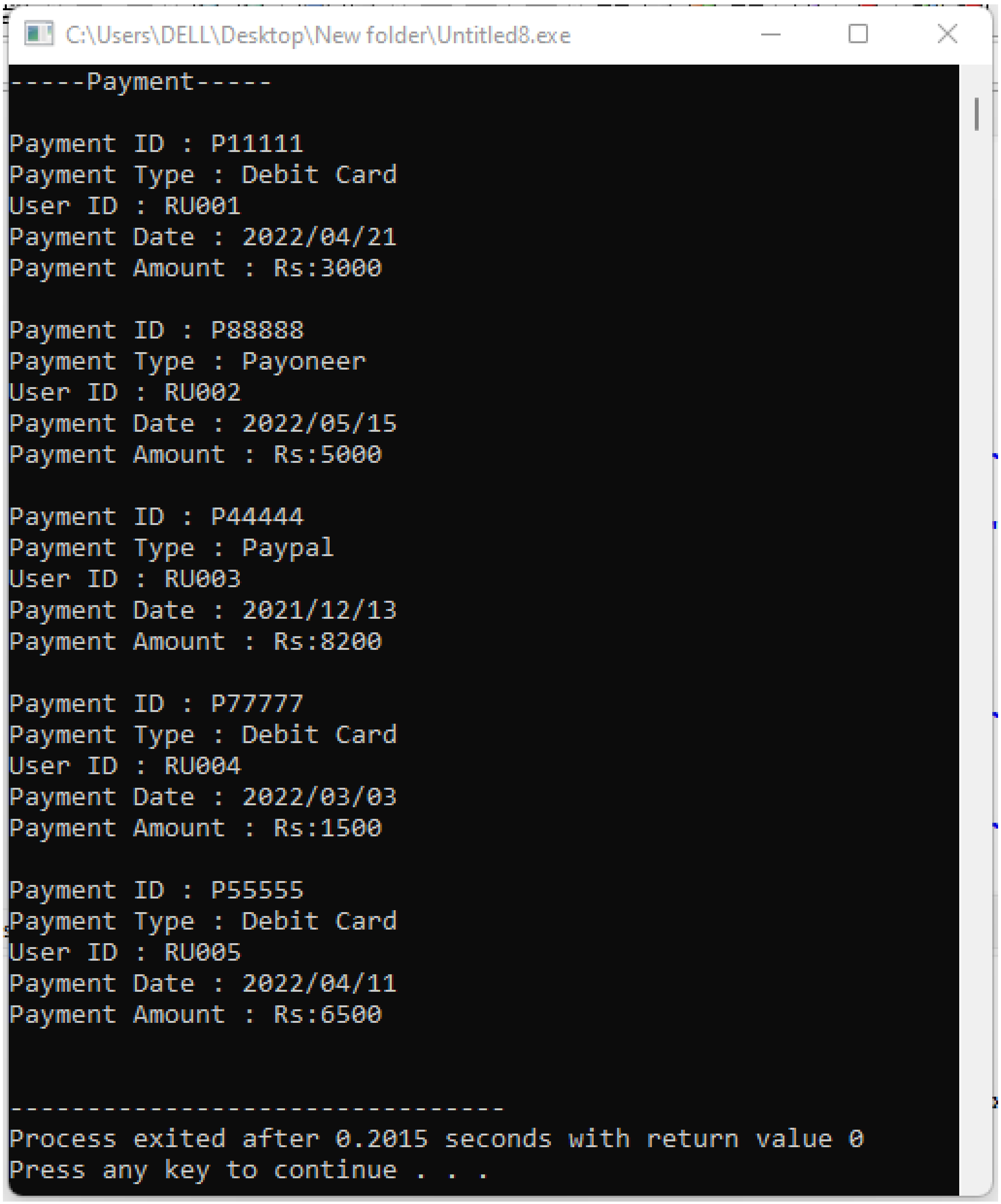


C:\Users\DELL\Desktop\New folder\Unregisterd_Guest_User.exe

```
-----Unregistered Guset User-----

Name : Dhanuka
NIC : 938855666V
Address : Matara
Contact No : 0702222222
Email : dhanu2010@gmail.com
Date Of Birth : 1993/11/25

Name : Shehani
NIC : 949998885V
Address : Galle
Contact No : 0777777777
Email : shenu94@gmail.com
Date Of Birth : 1994/02/02

Name : Yenuja
NIC : 2000996666
Address : Colombo
Contact No : 075000000
Email : yenu@email.com.com
Date Of Birth : 2000/03/09

Name : Sadew
NIC : 199952541V
Address : Malabe
Contact No : 0715556624
Email : sadew@gmail.com
Date Of Birth : 1999/11/04

Name : Isuru
NIC : 923222222V
Address : Rathnapura
Contact No : 0762225522
Email : isure@yahoo.com
Date Of Birth : 1992/11/19


--------------------------------
Process exited after 0.07265 seconds with return value 0
Press any key to continue . . .
```



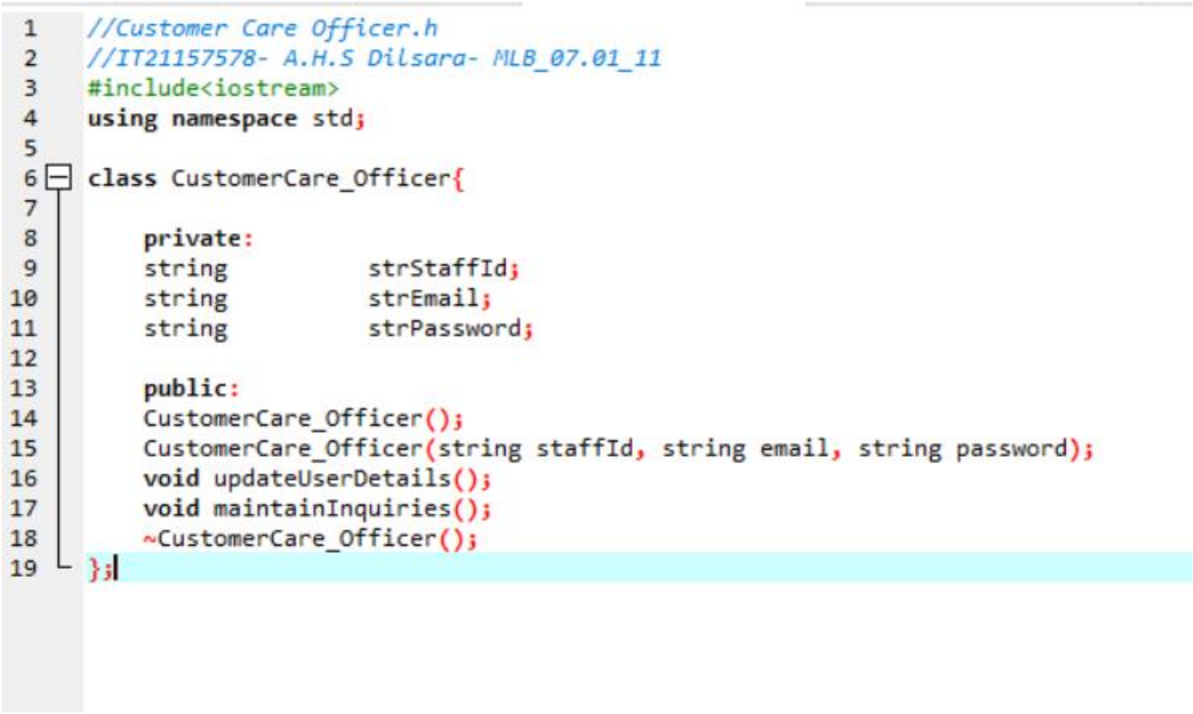
```
----------Registered User----------

User ID : RU001
Email : nethu@email.com
Password : *8888*

User ID : RU002
Email : sayuri10@gmail.com
Password : ss123

User ID : RU003
Email : gamage@yahoo.com
Password : 121212

User ID : RU004
Email : dilka@gmail.com
Password : hhijk

User ID : RU005
Email : shehana99@email.com
Password : iii**uuu


--------------------------------
Process exited after 0.03883 seconds with return value 0
Press any key to continue . . .
```



```
-----Payment-----

Payment ID : P11111
Payment Type : Debit Card
User ID : RU001
Payment Date : 2022/04/21
Payment Amount : Rs:3000

Payment ID : P88888
Payment Type : Payoneer
User ID : RU002
Payment Date : 2022/05/15
Payment Amount : Rs:5000

Payment ID : P44444
Payment Type : Paypal
User ID : RU003
Payment Date : 2021/12/13
Payment Amount : Rs:8200

Payment ID : P77777
Payment Type : Debit Card
User ID : RU004
Payment Date : 2022/03/03
Payment Amount : Rs:1500

Payment ID : P55555
Payment Type : Debit Card
User ID : RU005
Payment Date : 2022/04/11
Payment Amount : Rs:6500



--------------------------------
Process exited after 0.2015 seconds with return value 0
Press any key to continue . . .
```

Codes

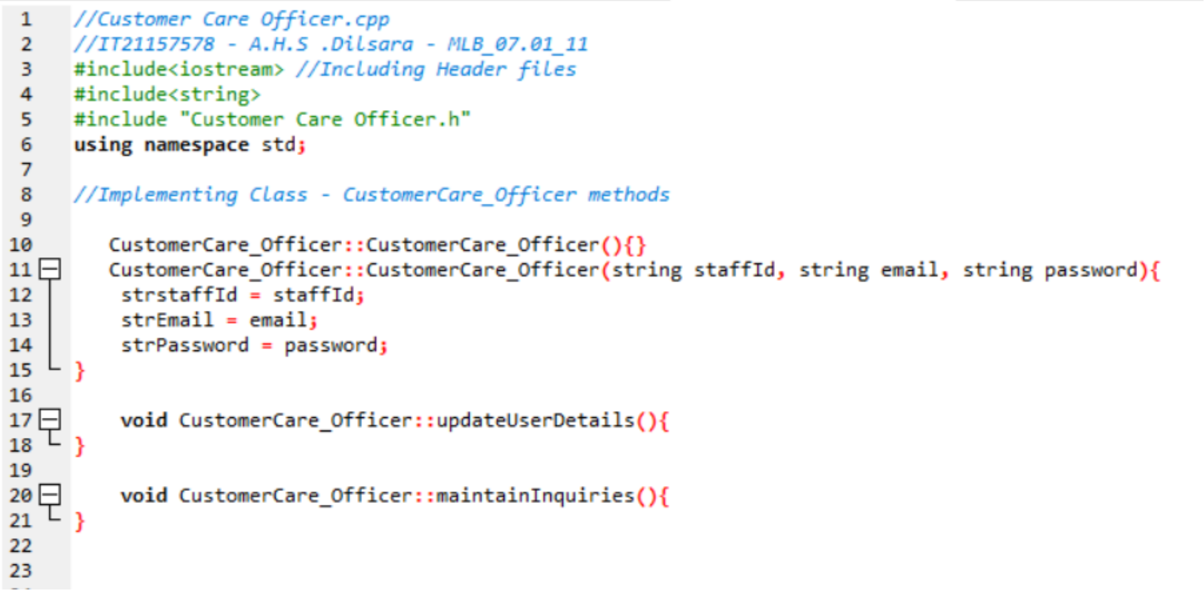
```
//Customer Care Officer.h

//IT21157578- A.H.S Dilsara- MLB_07.01_11

#include<iostream>

using namespace std;


class CustomerCare_Officer{


private:

   string     strStaffId;

   string        strEmail;

   string        strPassword;


   public:

   CustomerCare_Officer();

   CustomerCare_Officer(string staffId, string email, string password);

   void updateUserDetails();

   void maintainInquiries();

  ~CustomerCare_Officer();

};
```



```
//Customer Care Officer.h
//IT21157578- A.H.S Dilsara- MLB_07.01_11
#include<iostream>
using namespace std;

class CustomerCare_Officer{

    private:
    string          strStaffId;
    string          strEmail;
    string          strPassword;

    public:
    CustomerCare_Officer();
    CustomerCare_Officer(string staffId, string email, string password);
    void updateUserDetails();
    void maintainInquiries();
    ~CustomerCare_Officer();
};
```

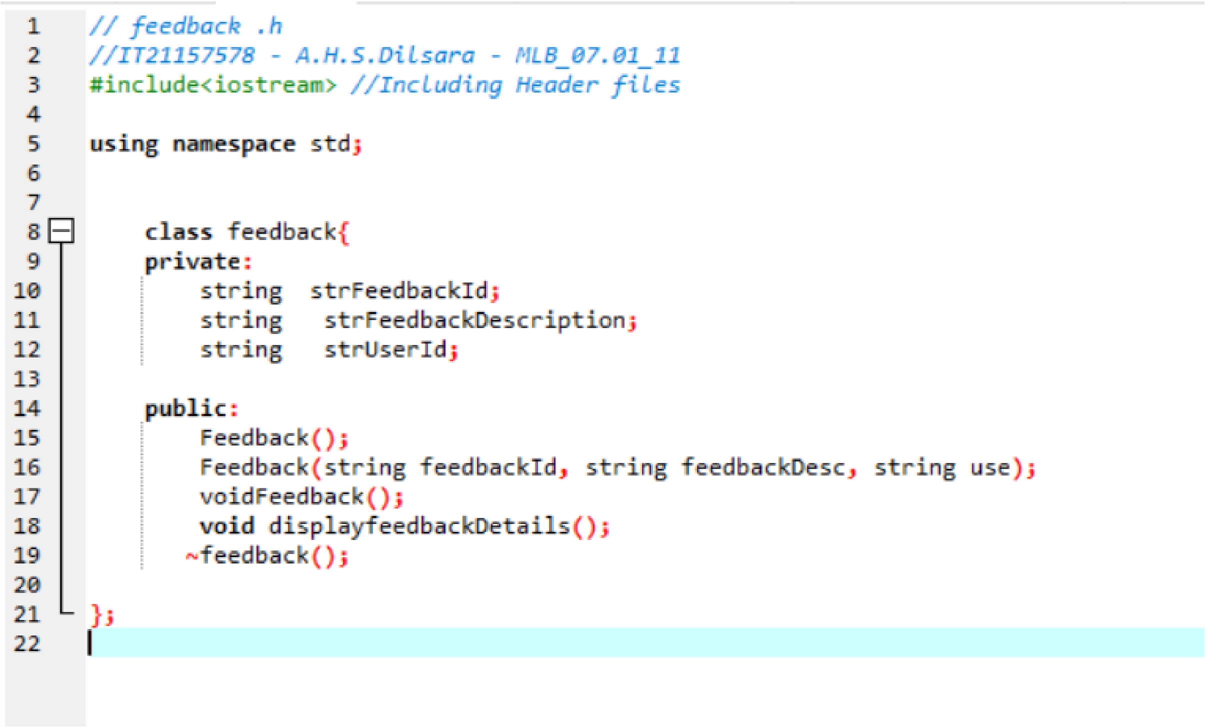
```
//Customer Care Officer.cpp

//IT21157578 - A.H.S .Dilsara - MLB_07.01_11

#include<iostream> //Including Header files

#include<string>

#include "Customer Care Officer.h"

using namespace std;


//Implementing Class - CustomerCare_Officer methods
```

```
  CustomerCare_Officer::CustomerCare_Officer(){}

  CustomerCare_Officer::CustomerCare_Officer(string staffId, string email, string password){

   strstaffId = staffId;

   strEmail = email;

   strPassword = password;

}


   void CustomerCare_Officer::updateUserDetails(){

}


   void CustomerCare_Officer::maintainInquiries(){

}
```



```
//Customer Care Officer.cpp
//IT21157578 - A.H.S .Dilsara - MLB_07.01_11
#include<iostream> //Including Header files
#include<string>
#include "Customer Care Officer.h"
using namespace std;

//Implementing Class - CustomerCare_Officer methods

   CustomerCare_Officer::CustomerCare_Officer(){}
   CustomerCare_Officer::CustomerCare_Officer(string staffId, string email, string password){
    strstaffId = staffId;
    strEmail = email;
    strPassword = password;
}

    void CustomerCare_Officer::updateUserDetails(){
}

    void CustomerCare_Officer::maintainInquiries(){
}


```

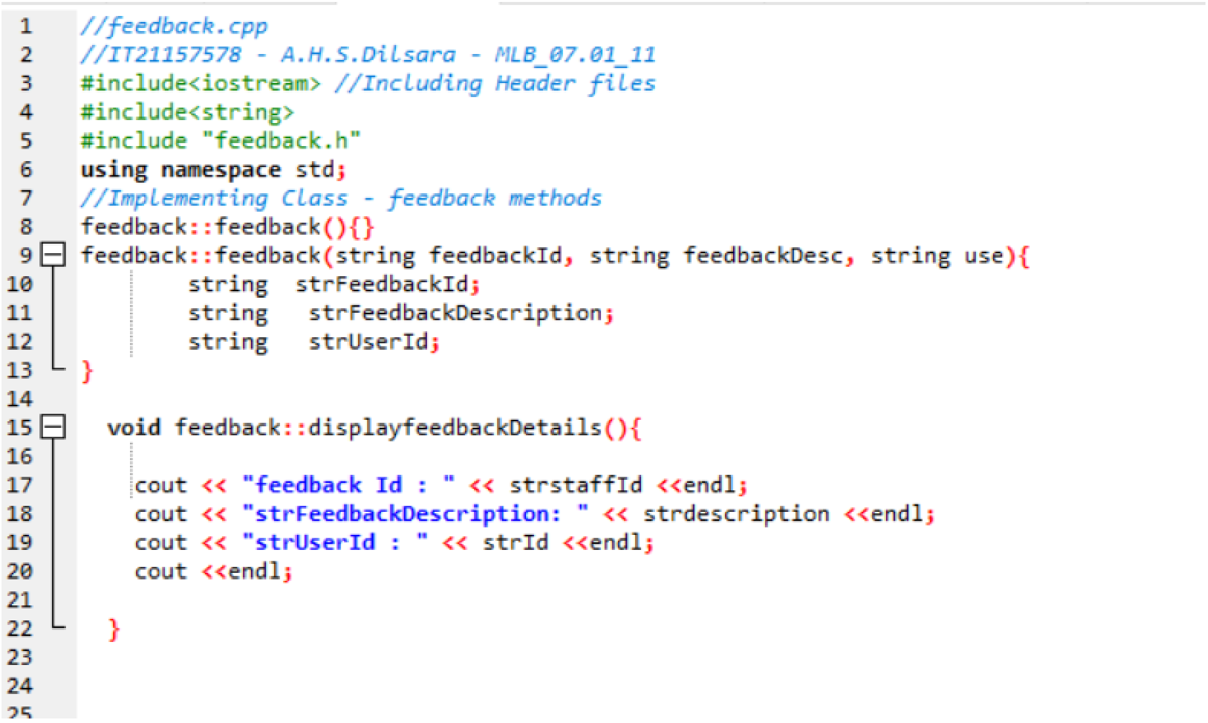
```
// feedback .h

//IT21157578 - A.H.S.Dilsara - MLB_07.01_11

#include<iostream> //Including Header files


using namespace std;



class feedback{

private:

   stringstrFeedbackId;

      string   strFeedbackDescription;

      string   strUserId;

public:

  Feedback();

      Feedback(string feedbackId, string feedbackDesc, string use);

      voidFeedback();

      void displayfeedbackDetails();

  ~feedback();

};
```



```
// feedback .h
//IT21157578 - A.H.S.Dilsara - MLB_07.01_11
#include<iostream> //Including Header files

using namespace std;


    class feedback{
    private:
        string  strFeedbackId;
        string   strFeedbackDescription;
        string   strUserId;

    public:
        Feedback();
        Feedback(string feedbackId, string feedbackDesc, string use);
        voidFeedback();
        void displayfeedbackDetails();
       ~feedback();

};

```

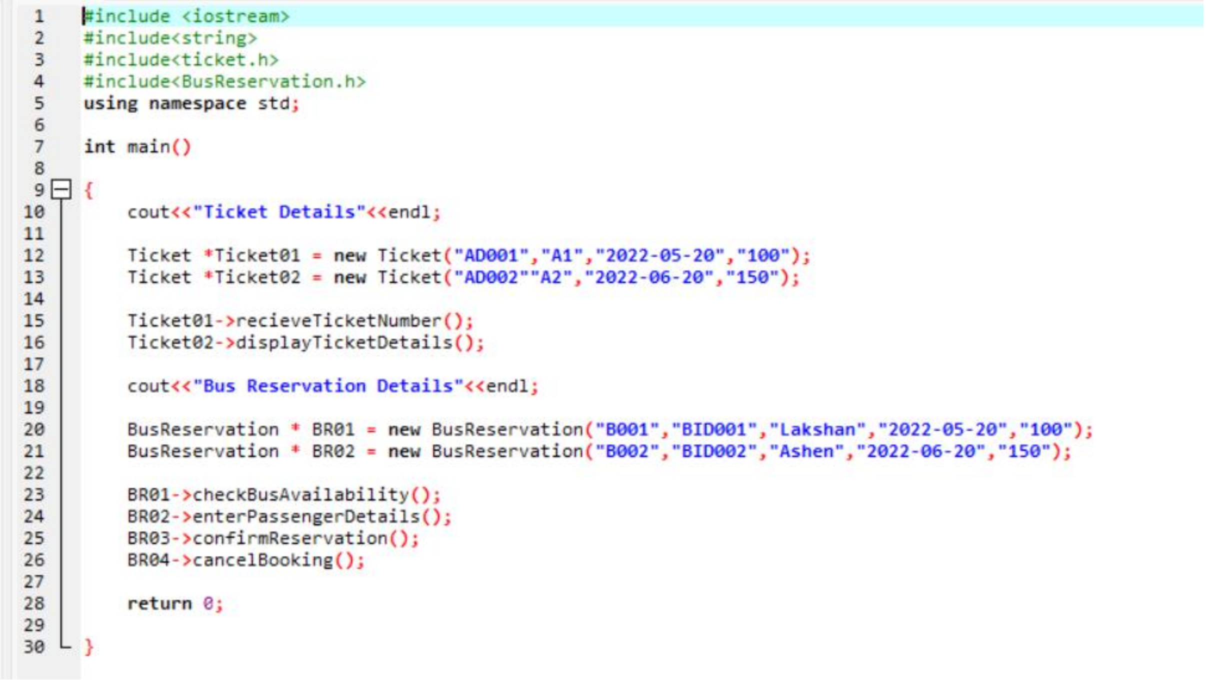
```
//feedback.cpp

//IT21157578 - A.H.S.Dilsara - MLB_07.01_11

#include<iostream> //Including Header files

#include<string>

#include "feedback.h"

using namespace std;

//Implementing Class - feedback methods

feedback::feedback(){}

feedback::feedback(string feedbackId, string feedbackDesc, string use){

                                                          string   strFeedbackId;

    string   strFeedbackDescription;

    string   strUserId;

}
```

```
 void feedback::displayfeedbackDetails(){


cout << "feedback Id : " << strstaffId <<endl;

cout << "strFeedbackDescription: " << strdescription <<endl;

cout << "strUserId : " << strId <<endl;

cout <<endl;


 }
```



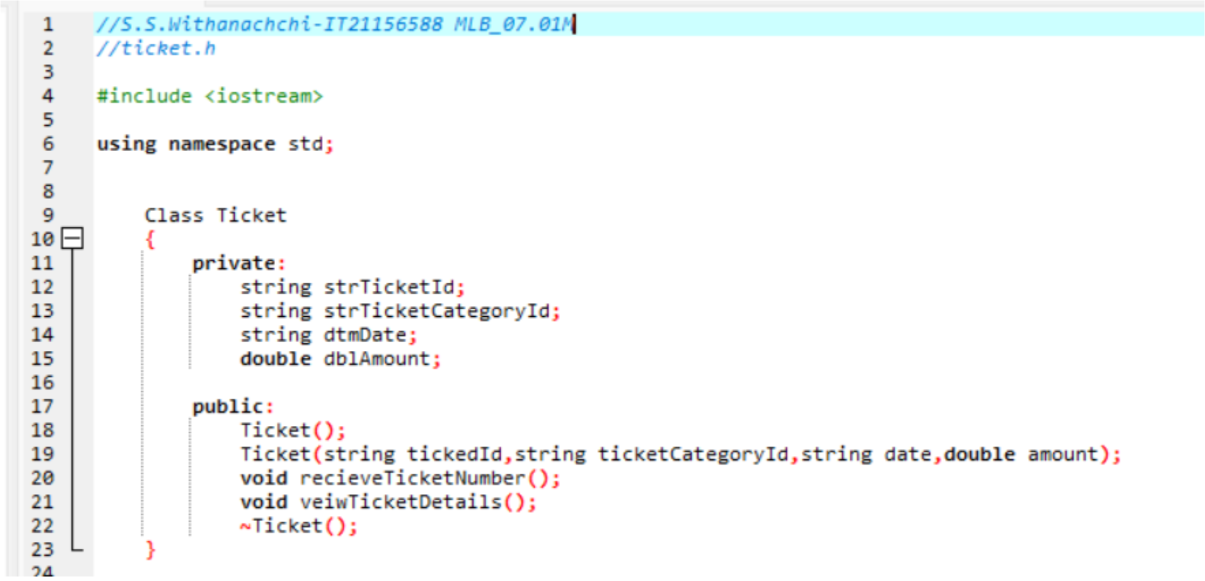
```
//feedback.cpp
//IT21157578 - A.H.S.Dilsara - MLB_07.01_11
#include<iostream> //Including Header files
#include<string>
#include "feedback.h"
using namespace std;
//Implementing Class - feedback methods
feedback::feedback(){}
feedback::feedback(string feedbackId, string feedbackDesc, string use){
        string  strFeedbackId;
        string   strFeedbackDescription;
        string   strUserId;
}

 void feedback::displayfeedbackDetails(){

    cout << "feedback Id : " << strstaffId <<endl;
    cout << "strFeedbackDescription: " << strdescription <<endl;
    cout << "strUserId : " << strId <<endl;
    cout <<endl;

 }
```

```
//S.S.Withanachchi-IT21156588 MLB_07.01M

//main.cpp

#include <iostream>

#include<string>

#include<ticket.h>

#include<BusReservation.h>

using namespace std;


int main()

{

cout<<"Ticket Details"<<endl;

Ticket *Ticket01 = new Ticket("AD001","A1","2022-05-20","100");

Ticket *Ticket02 = new Ticket("AD002""A2","2022-06-20","150");


Ticket01->recieveTicketNumber();

Ticket02->displayTicketDetails();


cout<<"Bus Reservation Details"<<endl;


BusReservation * BR01 = new BusReservation("B001","BID001","Lakshan","2022-05-20","100");

BusReservation * BR02 = new BusReservation("B002","BID002","Ashen","2022-06-20","150");


BR01->checkBusAvailability();

BR02->enterPassengerDetails();

BR03->confirmReservation();

BR04->cancelBooking();

return 0;                                        }
```



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

int main()

{
    cout<<"Ticket Details"<<endl;

    Ticket *Ticket01 = new Ticket("AD001","A1","2022-05-20","100");
    Ticket *Ticket02 = new Ticket("AD002""A2","2022-06-20","150");

    Ticket01->recieveTicketNumber();
    Ticket02->displayTicketDetails();

    cout<<"Bus Reservation Details"<<endl;

    BusReservation * BR01 = new BusReservation("B001","BID001","Lakshan","2022-05-20","100");
    BusReservation * BR02 = new BusReservation("B002","BID002","Ashen","2022-06-20","150");

    BR01->checkBusAvailability();
    BR02->enterPassengerDetails();
    BR03->confirmReservation();
    BR04->cancelBooking();

    return 0;

}
```

```
//S.S.Withanachchi-IT21156588 MLB_07.01M

//ticket.h


#include <iostream>

using namespace std;

Class Ticket

{

private:

string strTicketId;

string strTicketCategoryId;

string dtmDate;

double dblAmount;


public:

Ticket();

Ticket(string tickedId,string ticketCategoryId,string date,double amount);

void recieveTicketNumber();
void veiwTicketDetails();

 ~Ticket();

}
```



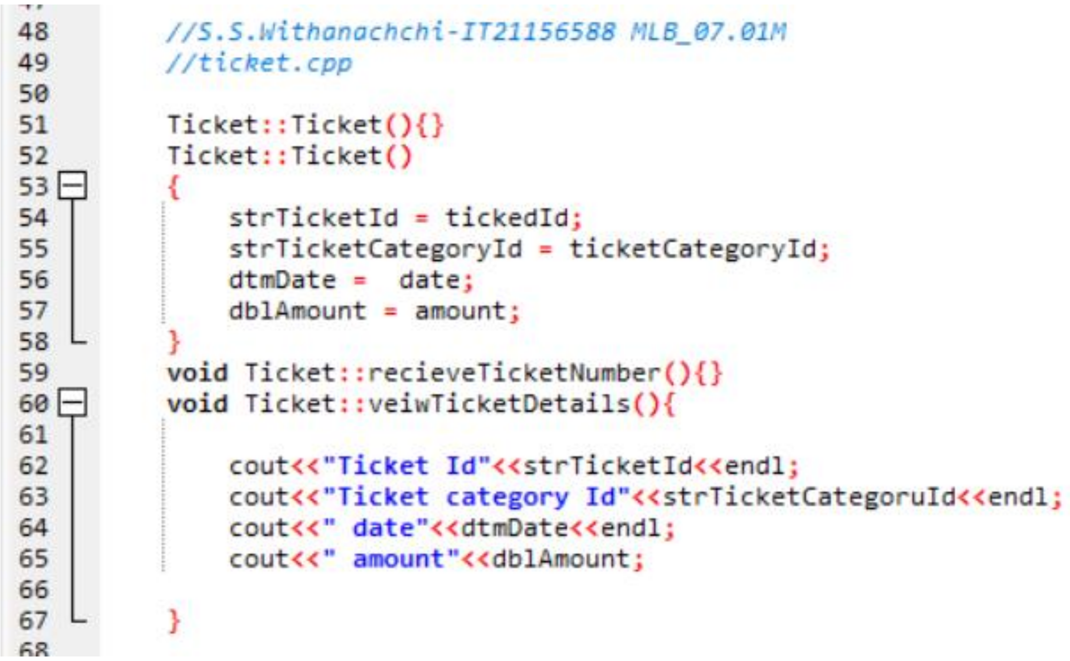
```
//S.S.Withanachchi-IT21156588 MLB_07.01M
//ticket.h

#include <iostream>

using namespace std;


    Class Ticket
    {
        private:
            string strTicketId;
            string strTicketCategoryId;
            string dtmDate;
            double dblAmount;

        public:
            Ticket();
            Ticket(string tickedId,string ticketCategoryId,string date,double amount);
            void recieveTicketNumber();
            void veiwTicketDetails();
            ~Ticket();
    }
```

```
//S.S.Withanachchi-IT21156588 MLB_07.01M

//Ticket.cpp

Ticket::Ticket(){}

Ticket::Ticket()

{

strTicketId = tickedId;

strTicketCategoryId = ticketCategoryId;

dtmDate =  date;

dblAmount = amount;

}

void Ticket::recieveTicketNumber(){}

void Ticket::veiwTicketDetails(){



cout<<"Ticket Id"<<strTicketId<<endl;

cout<<"Ticket category Id"<<strTicketCategoruId<<endl;
```

cout<<" date"<<dtmDate<<endl;

cout<<" amount"<<dblAmount;

}



```
//S.S.Withanachchi-IT21156588 MLB_07.01M
//ticket.cpp

Ticket::Ticket(){}
Ticket::Ticket()
{
    strTicketId = tickedId;
    strTicketCategoryId = ticketCategoryId;
    dtmDate =  date;
    dblAmount = amount;
}
void Ticket::recieveTicketNumber(){}
void Ticket::veiwTicketDetails(){

    cout<<"Ticket Id"<<strTicketId<<endl;
    cout<<"Ticket category Id"<<strTicketCategoruId<<endl;
    cout<<" date"<<dtmDate<<endl;
    cout<<" amount"<<dblAmount;

}
```

//S.S.Withanachchi-IT21156588 MLB_07.01M

//Busreservation.h

class BusReservation

{

private:

string strReservationId;

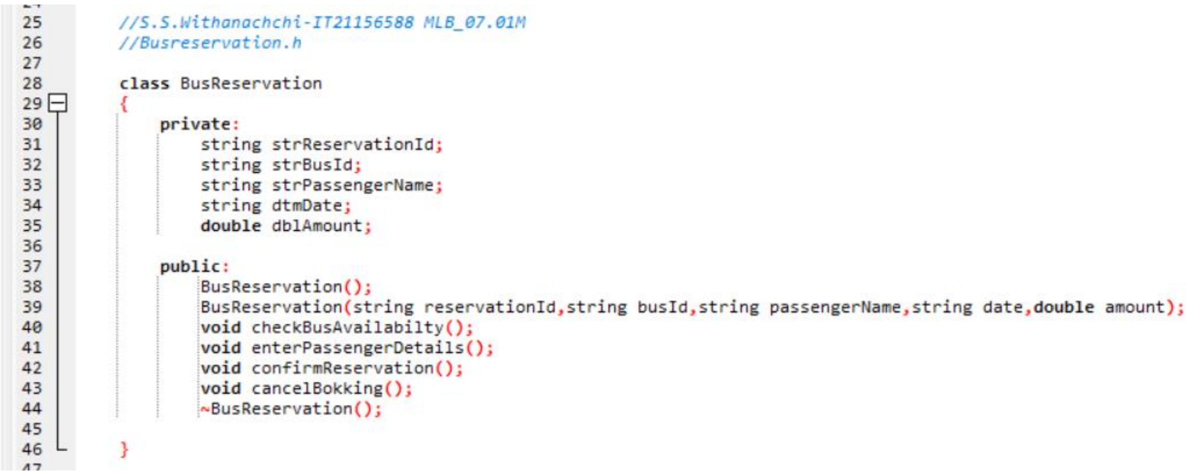
string strBusId;

string strPassengerName;

string dtmDate;

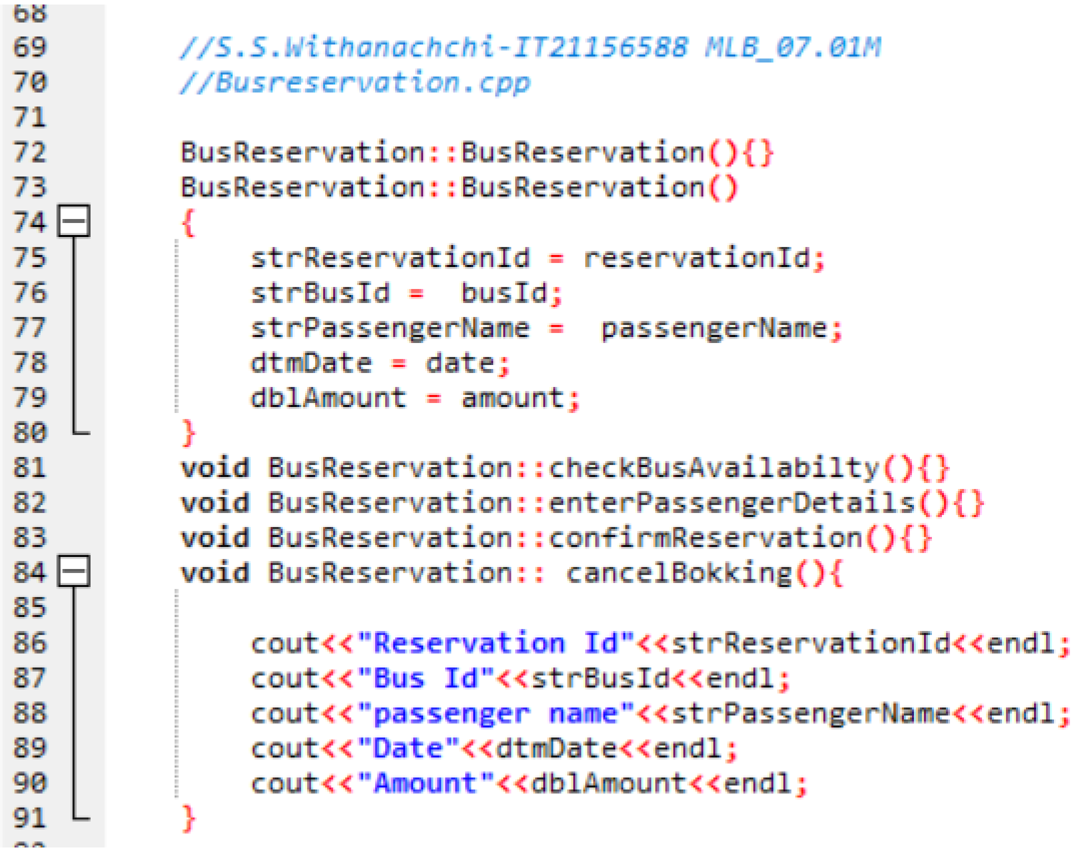
double dblAmount;

```
public:

BusReservation();

BusReservation(string reservationId,string busId,string passengerName,string date,double amount);

void checkBusAvailabilty();

void enterPassengerDetails();

void confirmReservation();

void cancelBokking();

~BusReservation();

}
```



```
//S.S.Withanachchi-IT21156588 MLB_07.01M
//Busreservation.h

class BusReservation
{
    private:
        string strReservationId;
        string strBusId;
        string strPassengerName;
        string dtmDate;
        double dblAmount;

    public:
        BusReservation();
        BusReservation(string reservationId,string busId,string passengerName,string date,double amount);
        void checkBusAvailabilty();
        void enterPassengerDetails();
        void confirmReservation();
        void cancelBokking();
        ~BusReservation();

}
```

```
//S.S.Withanachchi-IT21156588 MLB_07.01M

//Busreservation.cpp


BusReservation::BusReservation(){}

BusReservation::BusReservation()

{

strReservationId = reservationId;

strBusId =  busId;

strPassengerName =  passengerName;
dtmDate = date;
dblAmount = amount;
}
void BusReservation::checkBusAvailabilty(){}
void BusReservation::enterPassengerDetails(){}
void BusReservation::confirmReservation(){}
void BusReservation:: cancelBokking(){
cout<<"Reservation Id"<<strReservationId<<endl;
cout<<"Bus Id"<<strBusId<<endl;
cout<<"passenger name"<<strPassengerName<<endl;
cout<<"Date"<<dtmDate<<endl;
cout<<"Amount"<<dblAmount<<endl;
}
```



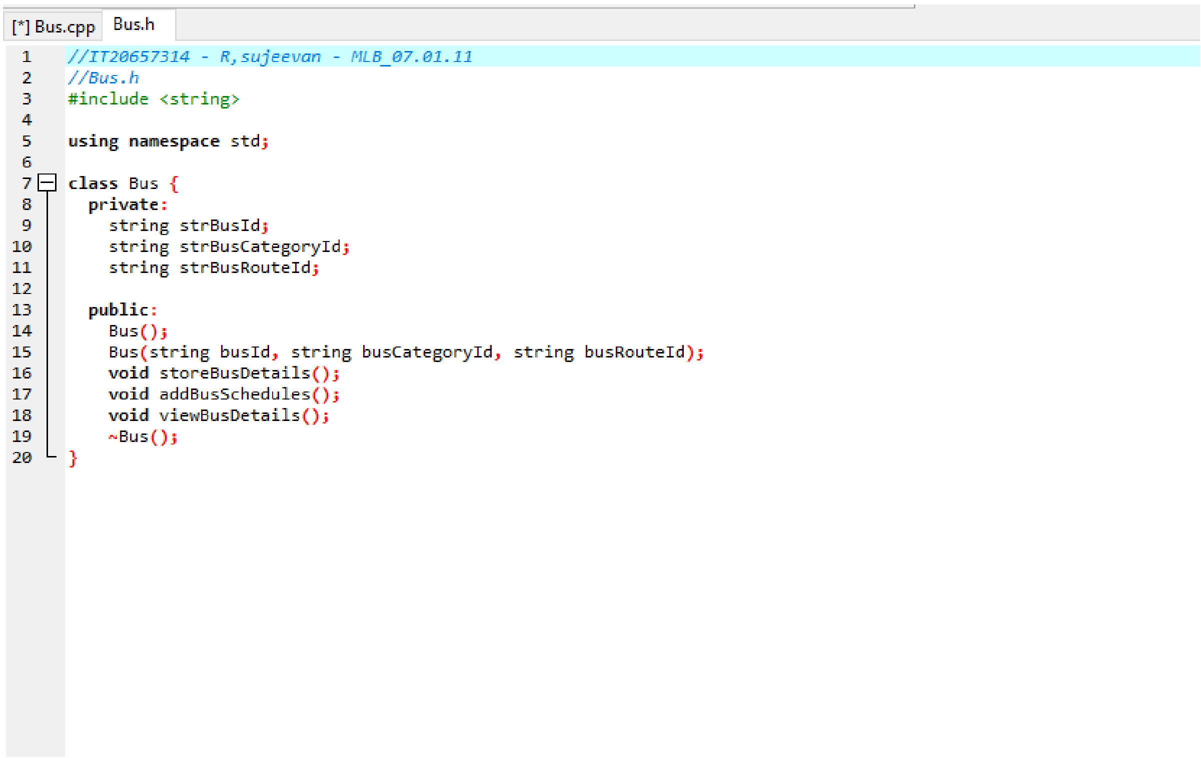
```

//S.S.Withanachchi-IT21156588 MLB_07.01M
//Busreservation.cpp

BusReservation::BusReservation(){}
BusReservation::BusReservation()
{
    strReservationId = reservationId;
    strBusId =  busId;
    strPassengerName =  passengerName;
    dtmDate = date;
    dblAmount = amount;
}
void BusReservation::checkBusAvailabilty(){}
void BusReservation::enterPassengerDetails(){}
void BusReservation::confirmReservation(){}
void BusReservation:: cancelBokking(){

    cout<<"Reservation Id"<<strReservationId<<endl;
    cout<<"Bus Id"<<strBusId<<endl;
    cout<<"passenger name"<<strPassengerName<<endl;
    cout<<"Date"<<dtmDate<<endl;
    cout<<"Amount"<<dblAmount<<endl;
}
```

Codes

//IT20657314 - R,sujeevan - MLB_07.01.11

//Bus.h

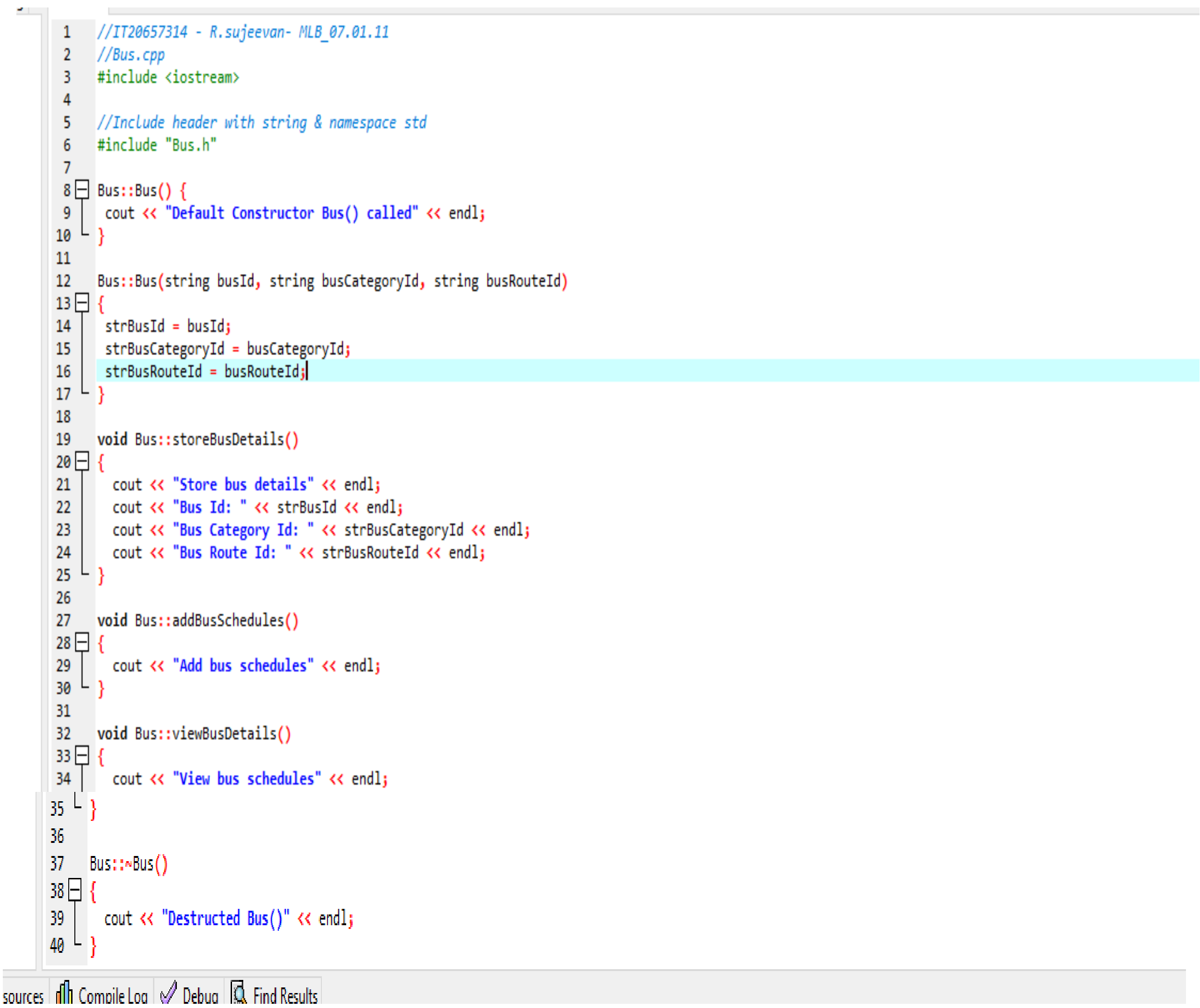
```
#include <string>


using namespace std;


class Bus {
  private:
    string strBusId;
    string strBusCategoryId;
    string strBusRouteId;


  public:
    Bus();
    Bus(string busId, string busCategoryId, string busRouteId);
    void storeBusDetails();
    void addBusSchedules();
    void viewBusDetails();
    ~Bus();
}
```



```
//IT20657314 - R,sujeevan - MLB_07.01.11
//Bus.h
#include <string>

using namespace std;

class Bus {
  private:
    string strBusId;
    string strBusCategoryId;
    string strBusRouteId;

  public:
    Bus();
    Bus(string busId, string busCategoryId, string busRouteId);
    void storeBusDetails();
    void addBusSchedules();
    void viewBusDetails();
    ~Bus();
}
```

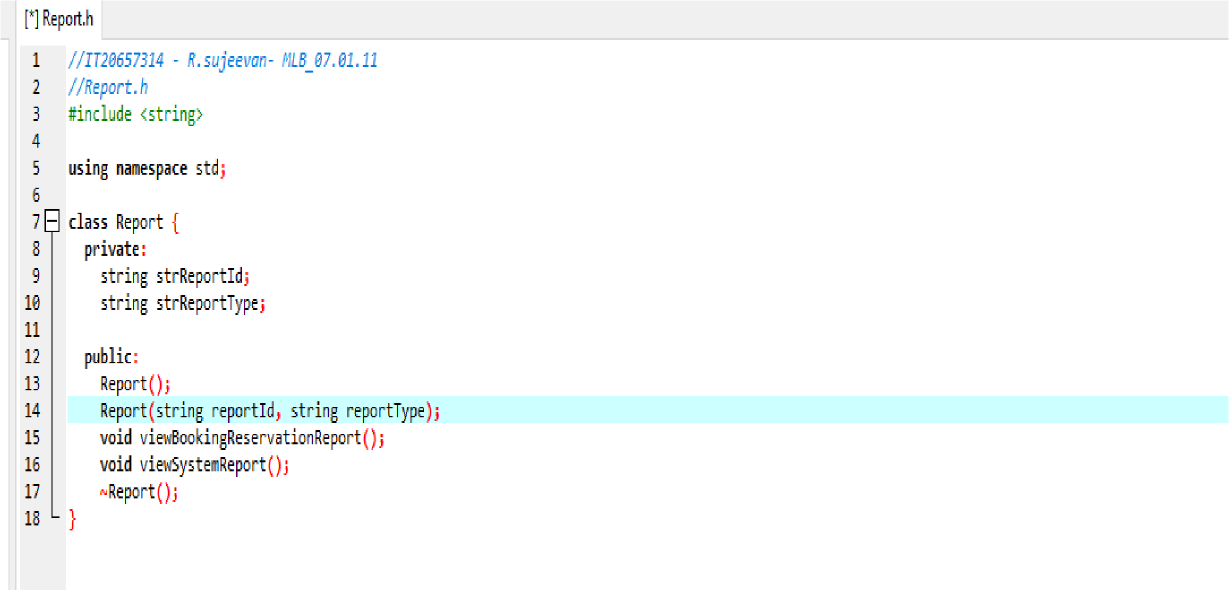
```
//IT20657314 - R.sujeevan- MLB_07.01.11
//Bus.cpp
#include <iostream>

//Include header with string & namespace std
#include "Bus.h"

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

Bus::Bus(string busId, string busCategoryId, string busRouteId)
{
 strBusId = busId;
 strBusCategoryId = busCategoryId;
 strBusRouteId = busRouteId;
}

void Bus::storeBusDetails()
{
  cout << "Store bus details" << endl;
  cout << "Bus Id: " << strBusId << endl;
  cout << "Bus Category Id: " << strBusCategoryId << endl;
  cout << "Bus Route Id: " << strBusRouteId << endl;
}

void Bus::addBusSchedules()
```

```
{
  cout << "Add bus schedules" << endl;
}

void Bus::viewBusDetails()
{
  cout << "View bus schedules" << endl;
}

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



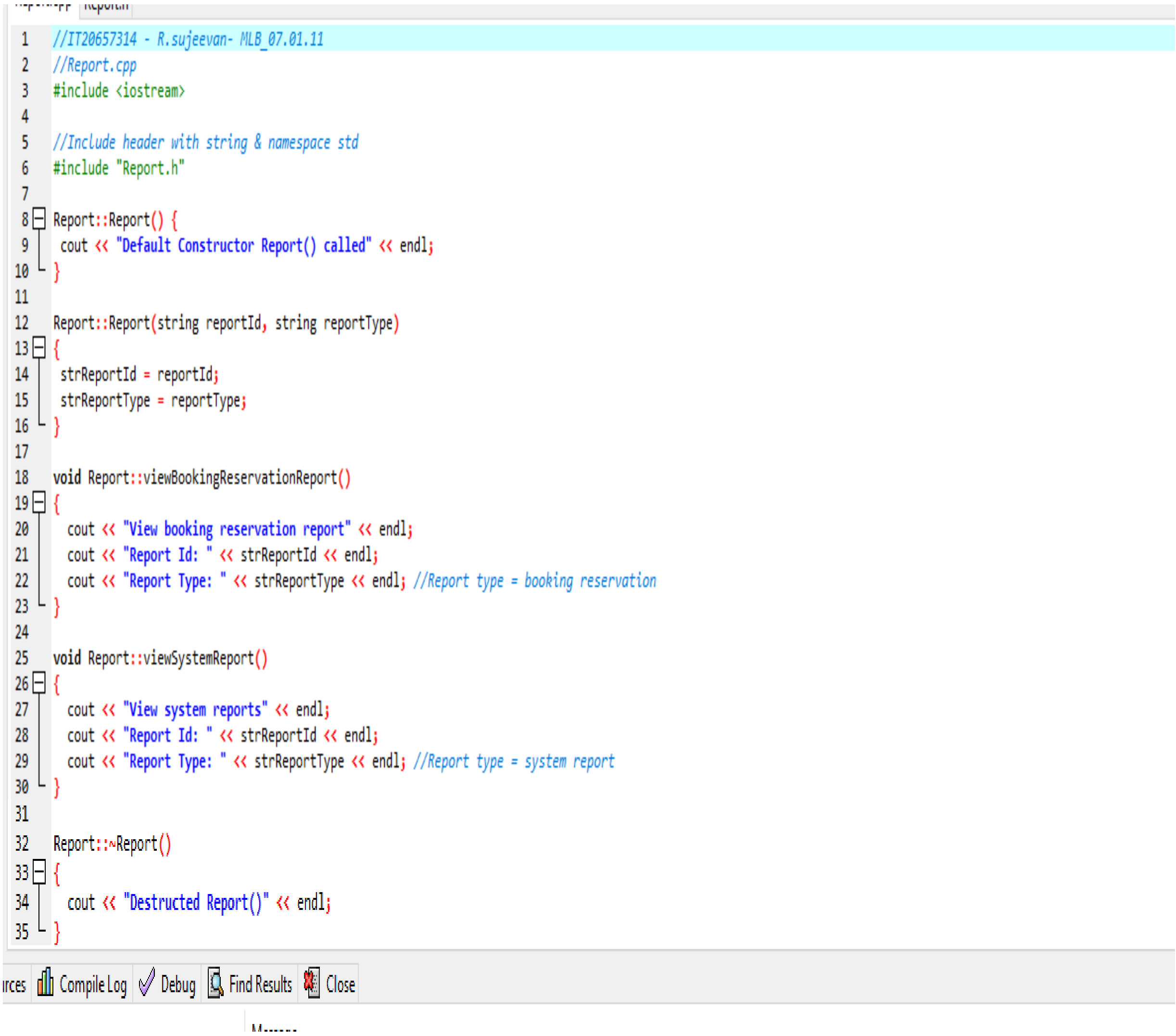
```
//IT20657314 - R.sujeevan- MLB_07.01.11
//Bus.cpp
#include <iostream>

//Include header with string & namespace std
#include "Bus.h"

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

Bus::Bus(string busId, string busCategoryId, string busRouteId)
{
 strBusId = busId;
 strBusCategoryId = busCategoryId;
 strBusRouteId = busRouteId;
}

void Bus::storeBusDetails()
{
  cout << "Store bus details" << endl;
  cout << "Bus Id: " << strBusId << endl;
  cout << "Bus Category Id: " << strBusCategoryId << endl;
  cout << "Bus Route Id: " << strBusRouteId << endl;
}

void Bus::addBusSchedules()
{
  cout << "Add bus schedules" << endl;
}

void Bus::viewBusDetails()
{
  cout << "View bus schedules" << endl;
}

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

```
//IT20657314 - R.sujeevan- MLB_07.01.11

//Report.h

#include <string>


using namespace std;


class Report {

  private:

    string strReportId;

    string strReportType;


  public:

    Report();

    Report(string reportId, string reportType);

    void viewBookingReservationReport();

    void viewSystemReport();

    ~Report();

}
```



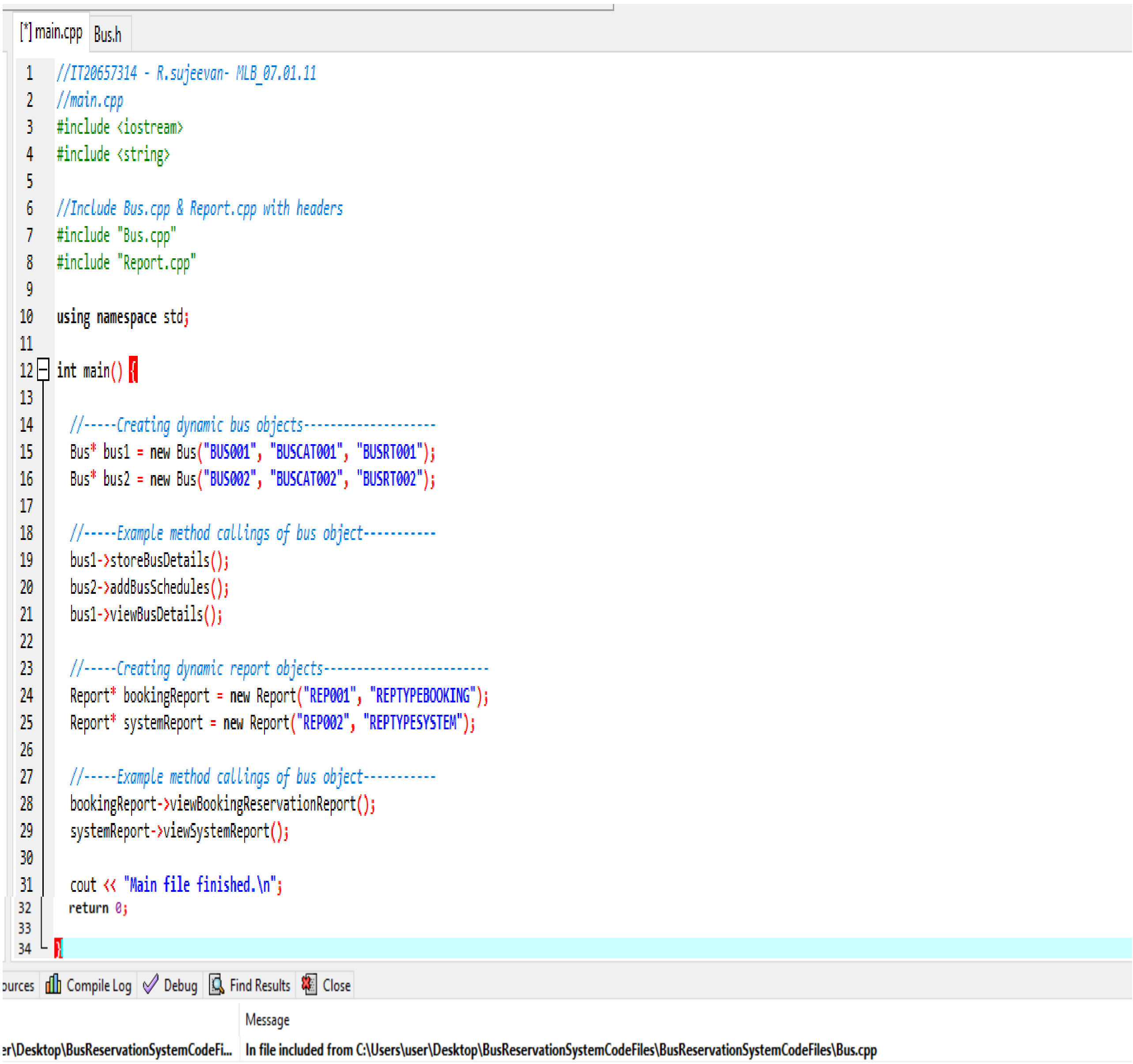
```
//IT20657314 - R.sujeevan- MLB_07.01.11
//Report.h
#include <string>

using namespace std;

class Report {
  private:
    string strReportId;
    string strReportType;

  public:
    Report();
    Report(string reportId, string reportType);
    void viewBookingReservationReport();
    void viewSystemReport();
    ~Report();
}
```

```
//IT20657314 - R.sujeevan- MLB_07.01.11

//Report.cpp

#include <iostream>


//Include header with string & namespace std

#include "Report.h"


Report::Report() {

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

}


Report::Report(string reportId, string reportType)

{

 strReportId = reportId;

 strReportType = reportType;

}


void Report::viewBookingReservationReport()

{

  cout << "View booking reservation report" << endl;

  cout << "Report Id: " << strReportId << endl;

  cout << "Report Type: " << strReportType << endl; //Report type = booking reservation

}


void Report::viewSystemReport()

{

  cout << "View system reports" << endl;
 cout << "Report Id: " << strReportId << endl;

 cout << "Report Type: " << strReportType << endl; //Report type = system report

}


Report::~Report()

{

 cout << "Destructed Report()" << endl;

}
```



```
//IT20657314 - R.sujeevan- MLB_07.01.11
//Report.cpp
#include <iostream>

//Include header with string & namespace std
#include "Report.h"

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

Report::Report(string reportId, string reportType)
{
 strReportId = reportId;
 strReportType = reportType;
}

void Report::viewBookingReservationReport()
{
  cout << "View booking reservation report" << endl;
  cout << "Report Id: " << strReportId << endl;
  cout << "Report Type: " << strReportType << endl; //Report type = booking reservation
}

void Report::viewSystemReport()
{
  cout << "View system reports" << endl;
  cout << "Report Id: " << strReportId << endl;
  cout << "Report Type: " << strReportType << endl; //Report type = system report
}

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

```
//IT20657314 - R.sujeevan- MLB_07.01.11

//main.cpp

#include <iostream>

#include <string>


//Include Bus.cpp & Report.cpp with headers

#include "Bus.cpp"

#include "Report.cpp"


using namespace std;


int main() {


  //-----Creating dynamic bus objects--------------------

  Bus* bus1 = new Bus("BUS001", "BUSCAT001", "BUSRT001");

  Bus* bus2 = new Bus("BUS002", "BUSCAT002", "BUSRT002");


  //-----Example method callings of bus object-----------

  bus1->storeBusDetails();

  bus2->addBusSchedules();

  bus1->viewBusDetails();


  //-----Creating dynamic report objects-------------------------

  Report* bookingReport = new Report("REP001", "REPTYPEBOOKING");

  Report* systemReport = new Report("REP002", "REPTYPESYSTEM");


  //-----Example method callings of bus object-----------
  bookingReport->viewBookingReservationReport();

  systemReport->viewSystemReport();


  cout << "Main file finished.\n";

  return 0;


}
```



```
//IT20657314 - R.sujeevan- MLB_07.01.11
//main.cpp
#include <iostream>
#include <string>

//Include Bus.cpp & Report.cpp with headers
#include "Bus.cpp"
#include "Report.cpp"

using namespace std;

int main() {

  //-----Creating dynamic bus objects-------------------
  Bus* bus1 = new Bus("BUS001", "BUSCAT001", "BUSRT001");
  Bus* bus2 = new Bus("BUS002", "BUSCAT002", "BUSRT002");

  //-----Example method callings of bus object----------
  bus1->storeBusDetails();
  bus2->addBusSchedules();
  bus1->viewBusDetails();

  //-----Creating dynamic report objects------------------------
  Report* bookingReport = new Report("REP001", "REPTYPEBOOKING");
  Report* systemReport = new Report("REP002", "REPTYPESYSTEM");

  //-----Example method callings of bus object----------
  bookingReport->viewBookingReservationReport();
  systemReport->viewSystemReport();

  cout << "Main file finished.\n";
  return 0;

}
```

```
er\Desktop\BusReservationSystemCodeFi...   In file included from C:\Users\user\Desktop\BusReservationSystemCodeFiles\BusReservationSystemCodeFiles\Bus.cpp
```

Individual Contributions

IT21155666 - Perera B.S.M.

- Noun Verb Analysis
- Identify All Classes and Objects
- Identify the All User requirements
- Coding of Admin, Management Assistant classes.
- Coordinate All Members
- Created Final Report

IT21156724 - W.G.P. Shehani

- Identify Responsibility and Collaborators.
- Created all CRC Card.
- Coding of Unregistered Guest User, Registered User and Payment classes.
- Supports to coordinate the members
- Motivate other members to achieve the Final output.

IT20657314 - R. Sujeewan

- identify attributes and methods
- drew complete class diagram
- coding of bus and report class
- help other group members to complete the project .

IT21156588 - S.S. Withanachchi

- Created codes Bus Reservation and ticket
- Typed CRC cards
- Help to group members complete the project

IT21157578 - A.H.S. Dilsara

- Created codes Feedback and customer care officer
- Help to group members complete the project