Topic : Online Teacher Trainer

Group no : MLB_08.01_12

Campus : Malabe / Metro / Matara / Kandy / Kurunegala / Kandy / Jaffna

Submission Date: 17/05/2022

We declare that this is our own work, and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT21162428	C.D. Rathnaweera	076-3855055
IT21162046	W.A.R.R. Ubeysinghe	076-3964293
IT21160998	N.R. Wijethunge	072-4889822
IT21162800	R.M.P.N. Rathnayake	077-5427282
IT21160752	I.R. Sirigampala	078-5850863

Requirement Analysis

- This is an Online Teacher Trainer Website which provides teaching courses for students.
- Unregistered users should register to the website by making an user account.
- Once they register to the website, the administrator will send the username and password.
- Then the system will send a verification code to the user to verify the user.
- Unregistered users can access the details of courses and more information.
- Registered users must log in to the website to access their accounts.
- System must allow the registered users to reset the password.
- Students must have to enroll to the respective courses they follow and do the payments using payment methods such as credit card, debit card or paypal.
- If they pay by their PayPal account, they should provide their user ID and password. Otherwise they should provide their card name, card number, expiry date, security code if they are using a credit card or a debit card
- Students can download lecture materials and upload their exercises.
- Students must be able to submit feedbacks and questionnaires to the system.
- The system should notify the examination dates, announcements, timetable and etc to the students.
- The lecturer should upload the lecture materials and download students' exercises.
- Course Coordinator must be able to update, modify and delete subject details.
- Course Coordinator must be able to reschedule and cancel lectures, timetables.
- Student, lecturer and Course Coordinator can add, remove details with permission of the administrator. Administrator also can reject the permission.
- Employee management administrator must be able to add or remove new employee details.
- Student management administrator must be able to add or remove new student details.

Classes

- Courses
- Registered User
- Students
- Unregistered User
- Administrator
- Lecturer
- Course Coordinator
- Subject
- Payment
- Credit Card
- Debit Card
- Paypal
- Enrolment

CRC Cards

Unregistered User	
Responsibility	**Collaborators**
Register to the website by making an account	
Receive username and password	Administrator
Access the course details	Course

Registered User	
Responsibility	**Collaborators**
Log into the system with credentials	

Administrator	
Responsibility	**Collaborators**
Send Username and Password	
Give/Reject permission to Add, Remove student details	Registered User
Give/Reject permission to Add, Remove lecture details	Registered User
Give/Reject permission to Add, Remove course coordinator details	Registered User

Course Coordinator	
Responsibility	**Collaborators**
Log into the system	Registered User
Add, Remove course coordinator details	Administrator, Registered User
Update, Delete Subject details	Subject
Reschedule and cancel lectures and timetables	

Lecturer	
Responsibility	**Collaborators**
Log into the system	Registered User
Add, Remove lecturer details	Administrator, Registered User
Upload lecture materials	
Download student Exercises	

Student	
Responsibility	**Collaborators**
Log into the system	Registered User
Enrol for the course	Course, Payment
Add, Remove student details	Administrator, Registered User
Download lecture materials	
Upload Exercises	
Submit feedback and questionnaires	

Subject	
Responsibility	**Collaborators**
Store Subject details	
Display subject details	
Add, Remove subject details	Course coordinator

Course	
Responsibility	**Collaborators**
Store course details	
Display course details	

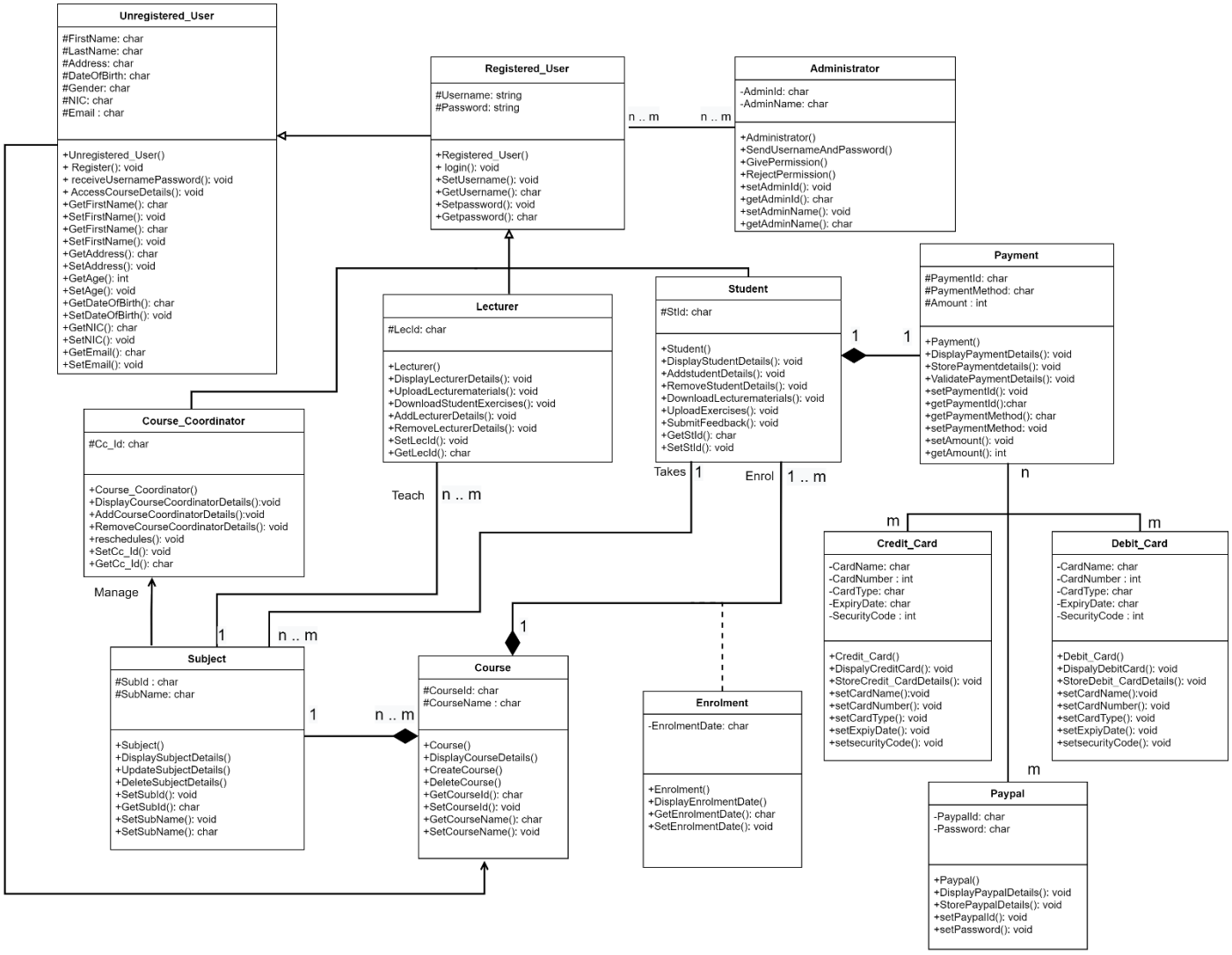
Payment	
Responsibility	**Collaborators**
Store Payment details	Credit card, Debit Card, PayPal
Validate payment details	

Credit card	
Responsibility	**Collaborators**
Store Credit Card details	Payment

Debit card	
Responsibility	**Collaborators**
Store Debit Card details	Payment

Paypal	
Responsibility	**Collaborators**
Store PayPal details	Payment

Exercise 1: Class diagram for online teacher trainer website



Exercise 2: C++ Code

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

//Unregistered User class
class Unregistered_User{
                private:
                    Course *cours;

                protected:
                    char FirstName[50];
                    char LastName[50];
                    char Address[100];
                   char DateOfBirth[10];
                    char NIC[10];
                    char Email[40];

                public:
                    Unregistered_User();
                    Unregistered_User(char f_name[],char l_name[],char address[],char
DOB,char nic[],char email[]);
                    void Register();
                    void receiveUsernamePassword();
                    void AccessCourseDetails();
                    void setFirstName(char f_name[]);
```

```
                    char getFirstName();
                    void setLastName(char l_name[]);
                    char getLastName();
                    void setAddress(char address[]);
                    char getAdress();
                    void setDateOfBirth(char DOB[]);
                    char getDateOfBirth();
                    void setNIC(char nic[]);
                    char getNIC();
                    void setEmail(char email[]);
                    char geteEmail();
};

                Unregistered_User::Unregistered_User(){
                    strcpy((FirstName),"");
                    strcpy((LastName),"");
                    strcpy((Address),"");
                    strcpy((DateOfBirth),"");
                    strcpy((NIC),"");
                    strcpy((Email),"");
                }

                Unregistered_User::Unregistered_User(char f_name[], char l_name[], char
adddress[], char DOB[], char nic[], char* email[]){
                    FirstName = f_name;
                    LastName = l_name;
                    Address = address;
                    DateOfBirth = DOB;
```

```
    NIC = nic;
    Email = email;
}

void Unregistered_User::Register(){

}

void Unregistered_User::receiveUsernamePassword(){

}

void Unregistered_User::setFirstName(char f_name[]){

}

char Unregistered_User::getFirstName(){
    return *FirstName;
}

void Unregistered_User::setLastName(char l_name[]){

}

char Unregistered_User::getLastName(){
    return *LastName;
}
```

```
void Unregistered_User::setAddress(char address[]){
}

char Unregistered_User::getAdress(){
    return *Address;
}

void Unregistered_User::setDateOfBirth(char DOB){

}

char Unregistered_User::getDateOfBirth(){
    return *DateOfBirth;
}

void Unregistered_User::setNIC(char nic[]){

}
char Unregistered_User::getNIC(){
    return *NIC;
}

void Unregistered_User::setEmail(char email[]){

}
```

```
char Unregistered_User::geteEmail(){
    return *Email;
}

//Registered User class
class Registered_user : public Unregistered_User{
    private:
        Administrator *admin;

    protected:
        char Username[10];
        char password[20];

    public:
        Registered_user();
        Registered_user(char username[],char pwd[]);
        void login();
        void setUsername(char username[]);
        char getUsername();
        void setPassword(char pwd[]);
        char getPassword();
};

Registered_user::Registered_user(){
    strcpy((Username),"");
    strcpy((password),"");
```

```
}

Registered_user::Registered_user(char username[],char pwd[]){
    Username = username;
    password = pwd;
}

void Registered_user::login(){

}

void Registered_user::setUsername(char username[]){

}

char Registered_user::getUsername(){
    return *Username;
}

void Registered_user::setPassword(char pwd[]){

}

char Registered_user::getPassword(){
    return *password;
}
```

```
//Student class
class Student : public  Registered_user{
                        private:
                            Subject *sub[SIZE];
                            Payment *pay[SIZE];

                        protected:
                            char StId[10];

                        public:
                            Student();
                            Student(char stid[]);
                            void DisplayStudentDetails(char stid[]);
                            void AddStudentDetails();
                            void RemoveStudentDetails();
                            void DownloadLecturematerials();
                            void UploadExercises();
                            void SubmitFeedback();
                            void SubmitQuestionnaires();
                            void setStId(char stid[]);
                            char getStId();
};
                        Student::Student(){
                            strcpy((StId),"");
                        }

                        Student::Student(char stid[]){
```

```
    StId = stid;
}

void Student::DisplayStudentDetails(char stid[]){

}

void Student::AddStudentDetails(){

}

void Student::RemoveStudentDetails(){

}

void Student::DownloadLecturematerials(){

}

void Student::UploadExercises(){

 }

void Student::SubmitFeedback(){

 }
```

```
void Student::SubmitQuestionnaires(){

}

void Student::setStId(char stid[]){

}

char Student::getStId(){
    return *StId;
}

//lecturer class
class Lecturer : public Registered_user{
    private:
        Subject *sub;

    protected:
        char LecId[10];

    public:
        Lecturer();
        Lecturer(char lecid[]);
        void DispalyLecturerDetails(char lecid[]);
        void UploadLecturematerials();
        void DownloadStudentExercises();
        void AddLecturerDetails();
```

```
        void RemoveLecturerDetails();
        void setLecId(char lecid[]);
        char getLecId();
};
Lecturer::Lecturer(){
    strcpy((LecId),"");
}

Lecturer::Lecturer(char lecid[]){
    LecId = lecid;
}

void Lecturer::DispalyLecturerDetails(char lecid[]){

}

void Lecturer::UploadLecturematerials(){

}

void Lecturer::AddLecturerDetails(){

}

void Lecturer::RemoveLecturerDetails(){

}
```

```
                    void Lecturer::setLecId(char lecid[]){


                    }


                    char Lecturer::getLecId(){
                        return *LecId;
                    }


//Course Coordinator class
class Course_coordinator : public Registered_user{
                    protected :
                        char Cc_Id[10];


                    public:
                        Course_coordinator();
                        Course_coordinator(char ccid[]);
                        void DispalyCourseCoordinatorDetails(char ccid[]);
                        void AddCourseCoordinatorDetails();
                        void RemoveCourseCoordinatorDetails();
                        void reschedules();
                        void setCc_Id(char ccid[]);
                        char getCc_Id();
};


                    Course_coordinator::Course_coordinator(){
                        strcpy((Cc_Id),"");
```

```
}

Course_coordinator::Course_coordinator(char ccid[]){
    Cc_Id = ccid;
}

void Course_coordinator::DispalyCourseCoordinatorDetails(char ccid[]){

}

void Course_coordinator::AddCourseCoordinatorDetails(){

}

void Course_coordinator::RemoveCourseCoordinatorDetails(){

}

void Course_coordinator::reschedules(){

}

void Course_coordinator::setCc_Id(char ccid[]){

}

char Course_coordinator::getCc_Id(){
```

```
                return *Cc_Id;
        }

//Adminstrator class
class Administrator{
        private:
                char AdminId[10];
                char AdminName[50];
                registered_User *regUser[SIZE];

        public:
                Administrator();
                Administrator(char AId[],char AName[]);
                void SendUsernameAndPassword();
                void GivePermission();
                void RejectPermission();
                void setAdminId(char AId);
                char getAdminID();
                void setAdminName(char AName);
                char getAdminName();
};

        Administrator::Administrator(){
                strcpy((AdminId),"");
                strcpy((AdminName),"");
        }
```

```
Administrator::Administrator(char AId[], char AName[]){
    AdminId = AId;
    AdminName = AName;
}

void Administrator::SendUsernameAndPassword(){

}

void Administrator::GivePermission(){

}

void Administrator::RejectPermission(){

}

void Administrator::setAdminId(char AID){

}

char Administrator::getAdminID(){
    return *AdminId;
}

void Administrator::setAdminName(char AName){
```

```
}

char Administrator::getAdminName(){
    return *AdminName;
}

//Course class
class Course{
    protected:
        char CourseId[5];
        char CourseName[40];

    public:
        Course();
        Course(char cid[],char cname[]);
        void DisplayCourseDetails(char cid[],char cname[]);
        void CreateCourse();
        void DeleteCourse();
        void setCourseId(char cid[]);
        char getCourseId();
        void setCourseName(char cname[]);
        char getCourseName();
};

Course::Course(){
    strcpy((CourseId),"");
    strcpy((CourseName),"");
```

```
}

Course::Course(char cid[], char cname[]){
    CourseId = cid;
    CourseName = cname;
}

void Course::DisplayCourseDetails(char cid[], char cname[] ){

}

void Course::CreateCourse(){

}

void Course::DeleteCourse(){

}

void Course::setCourseId(char cid[]){

}

char Course::getCourseId(){

}
```

```
void Course::setCourseName(char cname[]){

}

char Course::getCourseName(){

}

//Subject class
class Subject {
private:
    Student *std;
    Lecturer *lec[SIZE];
    Course_coordinator *ccodinator;

protected:
    char SubId[5];
    char SubName[40];

public:
    Subject();
    Subject(char subid[],char subname[]);
    void DisplaySubjectDetails(char subid[],char subname[]);
    void UpdateSubjectDetails();
    void DeleteSubjectDetails();
    void setSubId(char subid[]);
    char getSubId();
```

```
        void setSubName(char subname[]);
        char getSubName();
};

Subject::Subject(){
    strcpy((SubId),"");
    strcpy((SubName),"");
}

Subject::Subject(char sunid[], char subname[]){
    SubId = subid;
    SubName = subname;
}

void Subject::DisplaySubjectDetails(){

}

void Subject::UpdateSubjectDetails(){

}

void Subject::DeleteSubjectDetails(){

}

void Subject::setSubId(char subid[]){
```

```
}

char Subject::getSubId(){
    return *SubId;
}

void Subject::setSubName(char subname[]){

}

char Subject::getSubName(){
    return *SubName;
}


//Payment class
class Payment {
    private:
        Credit_Card *Cc_card[SIZE];
        Debit_Car *Dcard[SIZE];
        Paypal *paypal[SIZE];
        Student *std;

    protected:
        char paymentID[10];
        char PaymentMethod[20];
```

```
        int Amount;

    public:
        Payment();
        Payment(char PayId[],char Paymethod[], int PAmount);
        void DisplayPaymentDetails();
        void StorePaymentdetails();
        void ValidatePaymentDetails();
        void setPaymentId(char PayId[]);
        char getPaymentId();
        void setPaymentMethod(char Paymethod[]);
        char getPaymentMethod();
        void setAmount(int PAmount);
        int getAmount();
};

    Payment::Payment(){
        strcpy((paymentID),"");
        strcpy((PaymentMethod),"");
        Amount=0;

    }

    Payment::Payment(char payId[], char Paymethod[], int PAmount){
        paymentID = payId;
        PaymentMethod = Paymethod;
        Amount = PAmount;
```

```
}

void Payment::DisplayPaymentDetails(){

}

void Payment::StorePaymentdetails(){

}

void Payment::ValidatePaymentDetails(){

}

void Payment::setPaymentId(char PayId[]){

}

char Payment::getPaymentId(){
    return *paymentID;
}

void Payment::setPaymentMethod(char Paymethod[]){

}

char Payment::getPaymentMethod(){
```

```
                return *PaymentMethod;
        }

        void Payment::setAmount(int PAmount){

        }

        int Payment::getAmount(){
                return Amount ;
        }

//Credit card class
class Credit_Card{
        private:
                char CardName[50];
                int CardNumber;
                char CardType[30];
                char ExpiryDate[5];
                int SecurityCode;
                Payment *paymt[SIZE];
                Subject *sub[SIZE];
                Student *std[SIZE];

        public:
                Credit_Card();
                Credit_Card(char Cardname[],int CardNum,char Cardtype[],char ExpDate[],int
cvv);
                void DisplayCredit_CardDetails();
```

```
                    void StoreCredit_CardDetails();
                    void setCardName(char Cardname[]);
                    void setCardNumber(int CardNum);
                    void setCardType(char Cardtype[]);
                    void setExpiyDate(char ExpDate[]);
                    void setsecurityCode(int cvv);
};
                Credit_Card::Credit_Card(){
                    strcpy((CardName),"");
                    CardNumber = 0;
                    strcpy((CardType),"");
                    strcpy((ExpiryDate),"");
                    SecurityCode = 0;

                }

                Credit_Card::Credit_Card(char Cardname[],int CardNum,char Cardtype[],char
ExpDate[],int cvv){
                    CardName = Cardname;
                    CardNumber = CardNum;
                    CardType = Cardtype;
                    ExpiryDate = ExpDate ;
                    SecurityCode = cvv;
                }

                void Credit_Card::DisplayCredit_CardDetails(){

                }
```

```
void Credit_Card::StoreCredit_CardDetails(){

}

void Credit_Card::setCardName(char CardName[]){

}

void Credit_Card::setCardNumber(int CardNum){

}

void Credit_Card::setCardType(char Cardtype[]){

}

void Credit_Card::setExpiyDate(char ExpDate[]){

}

void Credit_Card::setsecurityCode(int cvv){

}

//Debit card class

class Debit_Card{
```

```
                private:
                    char CardName[50];
                    int CardNumber;
                    char CardType[30];
                    char ExpiryDate[5];
                    int SecurityCode;
                    Payment *paymt[SIZE];

                public:
                    Debit_Card();
                    Debit_Card(char Cardname[],int CardNum,char Cardtype[],char ExpDate[],int
cvv);
                    void DisplayDebit_CardDetails();
                    void StoreDebit_CardDetails();
                    void setCardName(char Cardname[]);
                    void setCardNumber(int CardNum);
                    void setCardType(char Cardtype[]);
                    void setExpiyDate(char ExpDate[]);
                    void setsecurityCode(int cvv);
};

                Debit_Card::Debit_Card(){
                    strcpy((CardName),"");
                    CardNumber = 0;
                    strcpy((CardType),"");
                    strcpy((ExpiryDate),"");
                    SecurityCode = 0;
                }
```

```
Debit_Card::Debit_Card(char Cardname[],int CardNum,char Cardtype[],char
ExpDate[],int cvv){
    CardName = Cardname;
    CardNumber = CardNum;
    CardType = Cardtype;
    ExpiryDate = ExpDate ;
    SecurityCode = cvv;
}

void Debit_Card::DisplayDebit_CardDetails(){

}

void Debit_Card::StoreDebit_CardDetails(){
}

void Debit_Card::setCardName(char CardName[]){

}

void Debit_Card::setCardNumber(int CardNum){

}

void Debit_Card::setCardType(char Cardtype[]){
```

```
}

void Debit_Card::setExpiyDate(char ExpDate[]){

}

void Debit_Card::setsecurityCode(int cvv){

}

//paypal class
class Paypal{
    private:
        char PaypalId[50];
        char Password[25];
        Payment *paymt[SIZE];

    public:
        Paypal();
        Paypal(char paypalId[]);
        void StorePaypalDetails();
        void setPaypallD(char paypalid[]);
        void setPassword(char pwd[]);
};
    Paypal::Paypal(){
    strcpy((PaypalId),"");
    strcpy((Password),"");
```

```
}

Paypal::Paypal(char paypalid,char pswd){
    PaypalId= paypalid;
    Password= pswd;
}
void Paypal::StorePaypalDetails(){

}
void Paypal::setPaypalID(char paypalid[]){

 }

void Paypal::setPassword(char pwd[]){

}

//Enrolment class
class Enrolment{
    private:
        Student *std;
        Course *cours;
        char EnrolmentDate[10];

    public:
        Enrolment();
        Enrolment(student *estd, course *ecourse, char edate);
```

```
        void DisplayEnrolmentDate(){
                std->DisplayStudentDetails(char stid[]);
                cours->DisplayCourseDetails(char cid[], char caname[]);
        }
        void setEnrolmentDate(char enroldate[]);
        char getEnrolmentDate();
};

Enrolment::Enrolment(){
        strcpy((EnrolmentDate),"");
}

Enrolment::Enrolment(student *estd, course *ecourse, char edate){
        std = estd;
        cours = ecourse;
        EnrolmentDate = edate;
}

void Enrolment::setEnrolmentDate(char enroldate[]){

}

char Enrolment::getEnrolmentDate(){
        return *EnrolmentDate;
}
```

```
int main(){

                    Unregistered_User *UnUser1;

                    UnUser1 = new Unregistered_User();

                    UnUser1->setFirstName("Ann");

                    UnUser1->setLastName("Fernando");

                    UnUser1->setAddress("20C, Temple Rd, Malabe.");

                    Unuser1->setDateOfBirth("1996/01/12");




                    Registered_user *RUser1;

                    RUser1 = new Registered_User();

                    RUser1->Unregistered_User("Evan","Rathnsingha","05/23, Nadun Uyana,
Galle","1995/05/16","199568471258","Evanrath.95@gmail.com");

                    RUser1->getFirstName();




                    Administrator *admin1;

                    admin1 = new Administrator("AD58979852","R M Perera");

                    admin1->GenerateUsernameAndPassword();

                    admin1->getAdminID();

                    admin1->getAdminName();
```

```
Student *std1;
std1 = new Student("STD1111111);
std1->getStId();
std1->DisplayStudentDetails();



Lecturer *lec1;
lec1 = new Lecturer("LC12345678);
lec1-> AddStudentDetails();
lec1->setFirstName("Sam");
lec1->setLastName("Perera");



Course_coordinator *CCoordinator1;
CCoordinator1 = new Course_coordinator("CC75398460");
CCoordinator1-> RemoveCourseCoordinatorDetails();
CCoordinator1->setNIC("197584546598");



Course *cours1;
cours1 = new Course("IT789","Information Technology");
cours1->DisplayCourseDetails();
cours1->DeleteCourse();
```

```
Subject *sub1;
sub1 = new Subject("IT105","Object Oriented Concept");
sub1->UpdateSubjectDetails();
sub1->getSubId();
sub1->DeleteSubjectDetails();



Payment *paymnt1;
paymnt1 = new Payment("PAY4545454","Credit card",100000);
paymnt1->getPaymentId();
paymnt1->setAmount(100000);



Credit_Card *Ccard1;
Ccard1 = new Credit_card();
Ccard1->setCardNumber(42364569985278);
Ccard1->setCardType("Visa");



Debit_Card *Dcard1;
Dcard1 = new Debit_Card();
Dcard1->setCardName("A R Wickramasingha");
Dcard1->setsecurityCode(879);
```

```
    Paypal *paypal1;
    paypal1 = new Paypal();
    paypal1->setPaypalID("UserABC@gmail.com");



    Enrolment *enrol1;
    enrol1 = new Enrolment("STD1112223334","IT002","2021/01/01");
    enrol1->Enrolment();
    enrol1->setEnrolmentDate("2022/03/01");



    delete UnUser1;
    delete RUser1;
    delete admin1;
    delete std1;
    delete lec1;
    delete CCoordinator1;
    delete cours1;
    delete sub1;
    delete paymnt1;
    delete Ccard1;
    delete Dcard1;
    delete paypal1;
    delete enrol1;
    return 0;
}
```

Individual Contributions

IT21162428 - C.D. Rathnaweera

- Created CRC Card for the Unregistered User and PayPal classes.
- Created the class diagram for Unregistered User and PayPal classes.
- Implemented the coding for the Unregistered User class and PayPal class .

IT21162046 - W.A.R.R. Ubeysinghe

- Created CRC Card for the Credit card, Debit card and Lecturer classes.
- Created the class diagram for the Credit card, Debit card and Lecturer classes.
- Implemented the coding for the Credit card, Debit card classes and Lecturer class.

IT21160998 - N.R. Wijethunge

- Created CRC Card for the Registered User and Payment classes.
- Created the class diagram for Registered User and Payment classes.
- Implemented the coding for the Registered User class and Payment class .

IT21162800 - R.M.P.N. Rathnayake

- Created CRC Card for the Student, Course and Enrolment classes.
- Created the class diagram for Student, Course and Enrolment classes.
- Implemented the coding for the Student, Course class es and Enrolment class .

IT21160752 - I.R. Sirigampala

- Created CRC Card for the Administrator, Course Coordinator and subject classes.
- Created the class diagram for the Administrator, Course Coordinator and Subject classes.
- Implemented the coding for the Administrator, Course Coordinator classes and Subject class.