Topic : Online Fitness Trainer

Group no : MLB_01.01_05

Campus : Malabe

Submission Date : 15 October 2021

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**
IT21020230	Siriwardana S. M. K. S	076 759 3818
IT21022142	Abeysinghe Y. P. P	071 117 1403
IT20607432	Bandara S P M K	071 352 0918
IT21010026	Disanayake A. D. M. M. S	078 674 3993
IT21013614	D.N Bulathsinhala	076 059 4336

Description

Online fitness trainer a system which enables the customers to maintain their physical as well as mental fitness via the virtual medium in this pandemic situation.

Well specialized team of trainers and gym instructors are included in this system while they are managed through a panel of administration and an attentive academic staff. Customers are let to login to the system through a web interface. Registered users are privileged to enroll into the courses. A Single user has the opportunity to enroll into a single course at a time. While workout plans which corresponds to the course selected by the user are definitely included with the course. As an additional option diet plans are also provided which can be a part of the course. But it is not exactly dependent with the course while user can get prescribed it by the nutritionist based on the user preferences.

Users have the access to the proceed the registration and rest of the payment by an online banking system linked with this system.

Exercise 01

Requirements

1. Guest should register to the system
2. After registering user should be login to the system by entering username and password
3. Then registered customers and instructors can view their profiles and they are able to edit or delete their profiles.
4. Then registered customers can select a preferred Course and workout plans depending on the course selected.
5. And registered customers can select a nutrition plan based on their preference
6. After selecting a course and relevant workout plans registered customer should make a payment through online by using PayPal, credit card or debit card.
7. System should check the payment details and confirm the payment details and a transaction report should be sent to the registered customer.
8. Then system should issue a payment receipt to the registered customer and system should display the registered customer and course details in the customer profiles.
9. Instructor can add, create, edit, and delete courses and workout plans.
10. System should display details about payments, courses, progress reports and reminders about diets and exercises to the registered customers.
11. Nutritionist can create, edit, and delete diet plans.
12. Staff maintain a documentation which includes the details about User enrolments.

Noun/Verb Analysis

(Nouns)

- Guest should register to the system
- After registering user should be login to the system by entering username and password
- Then registered customers and instructors can view their profiles and they are able to edit or delete their profiles.
- Then registered customers can select a preferred Course and workout plans depending on the course selected.
- And registered customers can select a nutrition plan based on their preference
- After selecting a course and relevant workout plans registered customer should make a payment through online by using PayPal, credit card or debit card.
- System should check the payment details and confirm the payment details and a transaction report should be sent to the registered customer.
- Then system should issue a payment receipt to the registered customer and system should display the registered customer and course details in the customer profiles.
- Instructor can add, create, edit, and delete courses and workout plans.
- System should display details about payments, courses, progress reports and reminders about diets and exercises to the registered customers.
- Nutritionist can create, edit, and delete diet plans.
- Staff maintain a documentation which includes the details about User enrolments.

(Verbs)

- Guest should register to the system
- After registering user should be login to the system by entering username and password
- Then registered customers and instructors can view their profiles and they are able to edit or delete their profiles.
- Then registered customers can select a preferred Course and workout plans depending on the course selected.
- And registered customers can select a nutrition plan based on their preference
- After selecting a course and relevant workout plans registered customer should make a payment through online by using PayPal, credit card or debit card.
- System should check the payment details and confirm the payment details and a transaction report should be sent to the registered customer.
- Then system should issue a payment receipt to the registered customer and system should display the registered customer and course details in the customer profiles.
- Instructor can add, create, edit, and delete courses and workout plans.
- System should display details about payments, courses, progress reports and reminders about diets and exercises to the registered customers.
- Nutritionist can create, edit, and delete diet plans
- Staff maintain a documentation which includes the details about User enrolments.

Identifying classes using noun/ verb analysis

1. Guest	- Redundant (with User)
2. System	- Out of the scope
3. User	- Class
4. Registered user	- Class
5. Instructors	- Class
6. Course	- Class
7. Workout plans	- Class
8. Payment	- Class
9. Username	- Attribute (of Registered User)
10. Password	- Attribute (of Registered User)
11. Nutritionist	- Class
12. Nutrition plans	- Class
13. Diet plan	- Redundant (with Nutrition plans)
14. Pay Pal	- Attribute (of Payment)
15. Credit card	- Attribute (of Payment)
16. Debit card	- Attribute (of Payment)
17. Details	- Meta Language (system data)
18. Enrolment	- Class
19. Staff	- Class

Selected Classes

1. User
2. Registered User
3. Instructors
4. Course
5. Workout plans
6. Payment
7. Nutritionist
8. Nutrition plans
9. Enrolment
10. Staff

Exercise 02

CRC Cards

Class Name: Instructor	
Responsibility	**Collaborators**
Add workout plans	Workout plans
Update workout plans	Workout plans
Delete workout plans	Workout plans

Class Name: Nutritionist	
Responsibility	**Collaborators**
Create nutrition plans	Nutrition Plans
Delete nutrition plans	Nutrition Plans

Class Name: User	
Responsibility	**Collaborators**
Register to the system	
Preview about Workout plans	Workout plans
Preview about Nutrition plans	Nutrition Plans

Class Name: Register User	
Responsibility	**Collaborators**
Can view their profiles and edit, delete their profiles	
Make payments	Payment
Can select a preferred course and workout plans depending on the course selected	Workout Plan, Course
Get Nutrition plans, nutrition details, and calculate BMI	Nutrition Plan

Class Name: Course	
Responsibility	**Collaborators**
Add Courses	Staff
Update / Delete Courses	Staff
View details about plans of particular courses	Workout Plans, Nutrition Plans

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

Class Name: Enrolment	
Responsibility	**Collaborators**
Get details regarding enrolment date and time	Course, Registered user
Display the details of the enrolment regarding the courses they have registered.	Course, Registered user

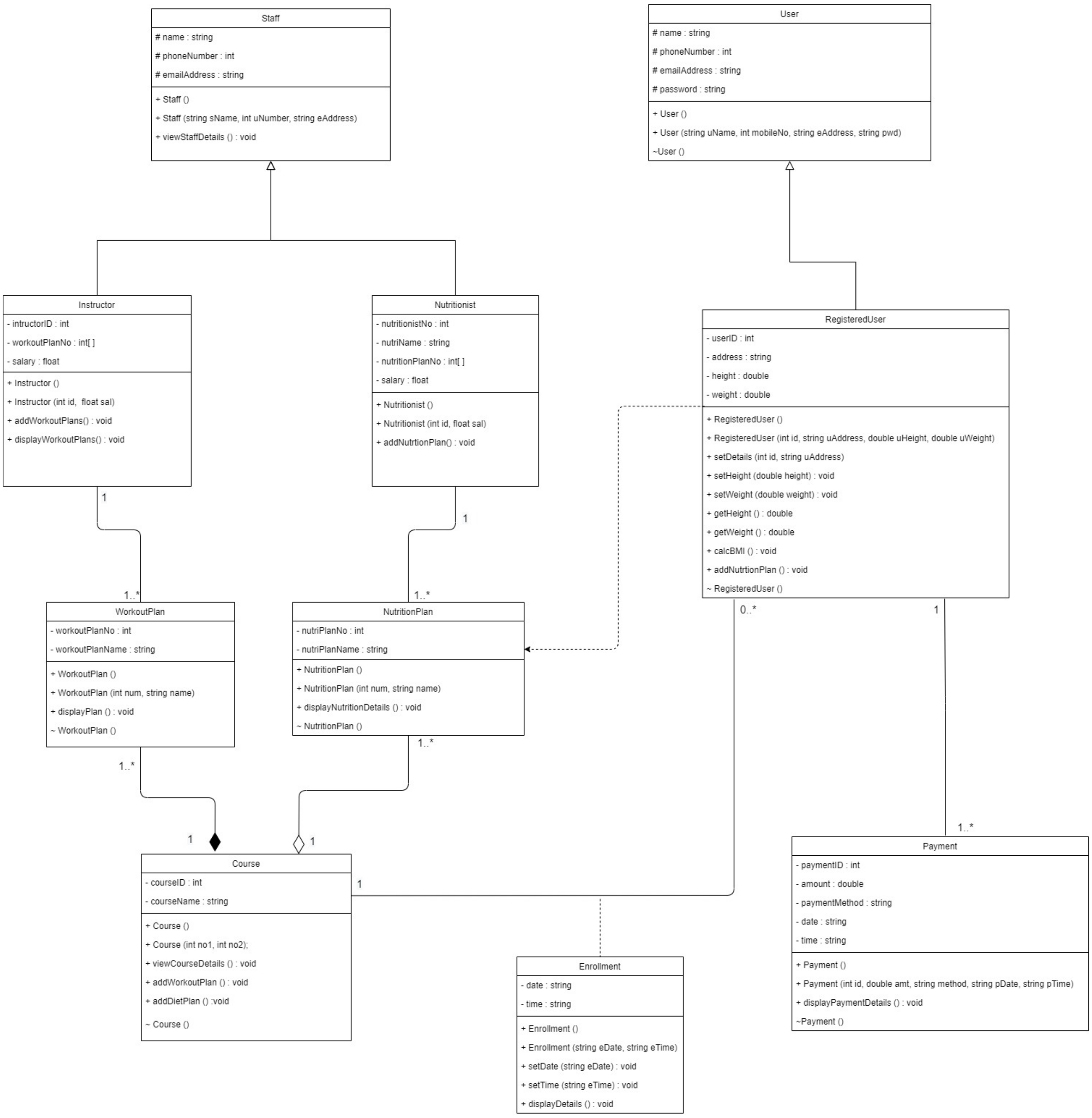
Class Name: Workout Plan	
Responsibility	**Collaborators**
Get the recent updates of instructor update the Workout plans.	Instructor
Display a list of workout plan which includes to the respective courses.	Courses

Class Name: Nutrition Plan	
Responsibility	**Collaborators**
Get the recent updates of instructor update the Nutrition Plans	Nutritionist
Display list of Nutrition Plan selected by user	Registered User

Class Name: Staff	
Responsibility	**Collaborators**
Manage registered users	Registered user
Manage courses	Course

Exercise 03

Class diagram



Exercise 04

Codings

User.h

```
class User {
protected:
    string name;
    int phoneNumber;
    string email;
    string password;

public:
    User();
    User(string uName, int mobileNo, string eAddress, string pwd);
    void displayUserDetails();
    ~User();
};
```

User.cpp

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

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

User::User(string uName, int mobileNo, string eAddress, string pwd)
{
    name = uName;
    phoneNumber = mobileNo;
    email = eAddress;
    password = pwd;
}

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

Staff.h

```
class Staff {
  protected:
    string name;
    int phoneNumber;
    string emailAddress;

  public:
    Staff();
    Staff(string sName, int pNumber, string eAddress);
    void viewStaffDetails();
    ~Staff();

};
```

Staff.cpp

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

Staff::Staff(){
   cout << "Default Constructor of the Staff is called" << endl;
}

Staff::Staff(string sName, int pNumber, string eAddress){
    name = sName;
    phoneNumber = pNumber;
    emailAddress = eAddress;
}

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

Course.h

```
#include <string>
#include "RegisteredUser.h"
#include "NutritionPlan.h"
#include "WorkoutPlan.h"
#define SIZE 3

class Course {
  private:
    int courseID;
    string courseName;
    WorkOutPlan * workPlan[SIZE];
    NutritionPlan * nutriPlan[SIZE];
    RegisteredUser* regUser[SIZE];

  public:
    Course();
    Course(int no1,int no2);
    void addworkoutPlan();
    void addDietPlan(NutritionPlan*p1,NutritionPlan *p2);
    void viewCourseDetails();
    ~Course();
};
```

Course.cpp

```
#include "Course.h"
#include "NutritionPlan.h"
#include "WorkOutPlan.h"
#include <iostream>

using namespace std;

Course::Course() {
     workPlan[0] = new WorkOutPlan(100);
     workPlan[1] = new WorkOutPlan(101);
}
```

```
Course::Course(int no1,int no2) {
     workPlan[0] = new WorkOutPlan(no1);
     workPlan[1] = new WorkOutPlan(no2);
}

void Course::addDietPlan(NutritionPlan *p1,NutritionPlan *p2) {
      nutriPlan[0] = p1;
      nutriPlan[1] = p2;
}

void Course::viewCourseDetails() {
    for(int x=0; x<2 ; x++)
      workPlan[x] -> displayDetails();
    for(int x=0; x<2 ; x++)
      nutriPlan[x] -> displayDetails();
}

Course :: ~Course() {
    cout << "Course is destructed" << endl;
    for(int x=0; x<2 ; x++)
      delete workPlan[x];
    cout << "All workout plans are deleted" << endl;

    for(int x=0; x<2 ; x++)
      delete nutriPlan[x];
    cout << "All nutrition plans are deleted" << endl;
}
```

Enrollment.h

```
#include <string>
#include "RegisteredUser.h"
#include "Course.h"

class  Enrollment {
private:
    string date;
    string time;

public:
    Enrollment();
    Enrollment(string eDate, string eTime);
    void setDate(string eDate);
    void setTime(string eTime);
    void displayDetails();
    RegisteredUser* regUser;
    Course* course;
};
```

Enrollment.cpp

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

Enrollment::Enrollment() {
    cout << "Default constructor of Enrollment called" << endl;
}

Enrollment::Enrollment(string eDate, string eTime) {
    date = eDate;
    time = eTime;
}
```

Instructor.h

```
#include "WorkOutPlan.h"
#include <string>
#define SIZE 2

class Instructor : public Staff{
  private:
    int InstructorID;
    float salary;
     WorkOutPlan *order[SIZE];

  public:
    Instructor();
    Instructor( int id, float sal);
    void addworkOutPlan(WorkOutPlan *WOplan);
    void displayinstructordetails();
};
```

Instructor.cpp

```
#include <iostream>
#include <string>
#include "Instructor.h"
#include "Staff.h"

using namespace std;

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

Instructor::Instructor(int id, float sal){
    InstructorID = id;
    salary = sal;
}

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

Nutritionist.h

```
#include "Staff.h"
#include "NutritionPlan.h"
#include <string>
#define SIZE 3

class Nutritionist : public Staff{
  private:
    int nutritionistNo;
    string nutriName;
    float salary;
```

```
      NutritionPlan *plan[SIZE];

  public:
      Nutritionist();
      Nutritionist( int id, float sal);
      void addNutritionPlan(NutritionPlan *Nplan);
      void displayNutridetails();
};
```

Nutritionist.cpp

```
#include <iostream>
#include <string>
#include "Nutritionist.h"
#include "Staff.h"

using namespace std;

Nutritionist::Nutritionist()
{
    cout << "Default constructor of  Nutritionist class is called"
<< endl;
}

Nutritionist::Nutritionist( int id, float sal)
{
    nutritionistNo = id;
    salary= sal;
}

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

NutritioniPlan.h

```
class NutritionPlan{
  private:
    int nutriPlanNo;
    string nutriplanName;
    Nutritionist *nutri_Person;

  public:
    NutritionPlan();
    NutritionPlan (int no,string name, Nutritionist *nut_Person);
    void displayNutritionDetails();
    void displayDetails();
    ~NutritionPlan();
};
```

NutritioniPlan.cpp

```
#include <iostream>
#include <string>
#include "Nutritionist.h"
#include "NutritionPlan.h"

using namespace std;

NutritionPlan::NutritionPlan(){
  cout << "Default constructor of NutritionPlan called" << endl;
}

NutritionPlan::NutritionPlan(int no, string name, Nutritionist
*nut_Person)
{
     nutriPlanNo = no;
     nutri_Person = nut_Person;
     nutriplanName = name;
     nutri_Person ->addNutritionPlan(this);
}

NutritionPlan::~NutritionPlan(){
   cout << "Nutrition plan destructed" << endl;
}
```

Payment.h

```
#include <string>
#include "RegisteredUser.h"

class Payment {
private:
    int paymentID;
    double amount;
    string paymentMethod;
    string date;
    string time;
    RegisteredUser* regUser;

public:
    Payment();
    Payment(int id, double amt, string method, string pDate, string
pTime);
    void displayPaymentDetails();
    ~Payment();
};
```

Payment.cpp

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

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

Payment::Payment(int id, double amt, string method, string pDate,
string pTime) {
    paymentID = id;
    amount = amt;
    paymentMethod = method;
    date = pDate;
    time = pTime;
}

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

RegisteredUser.h

```
#include <iostream>
#include <string>
#include "User.h"
#include "Payment.h"
#include "Course.h"
#define SIZE 2;

class RegisteredUser: public User {
private:
    int userID;
    string address;
    double height;
    double weight;
    Payment* paymemts[SIZE];
    Course* course;


public:
    RegisteredUser();
    RegisteredUser(int id, string uAddress, double uHeight, double 
uWeight);
    void setDetails(int id, string uAddress);
    void setHeight(double uHeight);
    void setWeight(double uWeight);
    double getHeight();
    double getWeight();
    double calcBMI();
    void addPlans(int noOfPlans, NutritionPlan* NP);
    ~RegisteredUser();
};
```

RegisteredUser.cpp

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

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

RegisteredUser::RegisteredUser(int id, string uAddress, double
uHeight, double uWeight) {
    userID = id;
    address = uAddress;
    height = uHeight;
    weight = uWeight;
}

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

WorkOutPlan.h

```cpp
#include <string>

class WorkOutPlan{
  private:
    int WorkoutPlanNo;
    string WorkoutPlanName;
    Instructor *instruct_Person;

  public:
    WorkOutPlan();
    WorkOutPlan(int no);
    WorkOutPlan (int no,string name, Instructor *inst_Person);
    void displayPlan();
    void displayDetails();
    ~WorkOutPlan();
};
```

WorkOutPlan.cpp

```
#include <iostream>
#include <string>
#include "Instructor.h"
#include "WorkOutPlan.h"

using namespace std;


  WorkOutPlan::WorkOutPlan(){
    cout << "Default constructor of workout plan is called" << endl;
  }
  WorkOutPlan::WorkOutPlan (int no,string name, Instructor
*inst_Person)
{
     WorkoutPlanNo = no;
     instruct_Person = inst_Person;
     WorkoutPlanName = name;
     instruct_Person ->addworkOutPlan(this);
}

  WorkOutPlan::~WorkOutPlan(){
    cout << "WorkOut plan destructed" << endl;
  }
```

Main.cpp

```
#include <iostream>
#include <string>

#include "User.h"
#include "User.cpp"
#include "RegisteredUser.h"
```

```
#include "RegisteredUser.cpp"
#include "Payment.h"
#include "Payment.cpp"
#include "Enrollment.h"
#include "Enrollment.cpp"
#include "Staff.h"
#include "Staff.cpp"
#include "Instructor.h"
#include "Instructor.cpp"
#include "Nutritionist.h"
#include "Nutritionist.cpp"
#include "WorkoutPlan.h"
#include "WorkoutPlan.cpp"
#include "NutritionPlan.h"
#include "NutritionPlan.cpp"
#include "Course.h"
#include "Course.cpp"

using namespace std;

int main() {

    User* user1 = new User("Sam", 723475648, "sam@gmail.com",
"Sam12345");
    User* user2 = new User("Claire", 724563123, "claire@gmail.com",
"Claire789");

    Staff* staff1 = new Staff("Janice", 885471355,
"jan@88gmail.com");
    Staff* staff2 = new Staff("Joe", 456783940, "joetr@gmail.com");

    RegisteredUser* rgUser1 = new RegisteredUser(1001, "No.142,
Pallandeniya, Maspotha", 1.54, 59.7);
    RegisteredUser* rgUser2 = new RegisteredUser(1002, "No.116,
Yanthampalawa, Kurunegala", 1.61, 64.6);

    Payment* payment1 = new Payment(101, 1800.0, "PAYPAL",
"25.09.2021", "11.34");
    Payment* payment2 = new Payment(102, 2500.0, "Online VISA",
"03.10.2021", "14.28");

    Enrollment* enroll1 = new Enrollment("21 / 12 / 2002", "12.21");
    Enrollment* enroll2 = new Enrollment("21 / 10 / 2002", "10.21");

    NutritionPlan* nPlan1 = new NutritionPlan(100, "Low-fat diet",
"J.A Samaraweera");
    NutritionPlan* nPlan2 = new NutritionPlan(200, "Moderate-dietary
diet", "M.M Samaranayake");
```

```
    WorkOutPlan* woPlan1 = new  WorkOutPlan(101, "Treadmilling", 
"S.S Abeysinghe");
    WorkOutPlan* woPlan2 = new  WorkOutPlan(102, "Cycling", 
"P.Weerasinghe");

    Instructor* instructor1 = new Instructor(1020, 60000);
    Instructor* instructor2 = new Instructor(1021, 65000);

    Nutritionist* nutritionist1 = new Nutritionist(1102, 55000);
    Nutritionist* nutritionist2 = new Nutritionist(1103, 63000);

    Course* course1 = new Course(12, 45);
    Course* course2 = new Course(13, 46);

    delete user1;
    delete user2;

    delete staff1;
    delete staff2;

    delete rgUser1;
    delete rgUser2;

    delete payment1;
    delete payment2;

    delete enroll1;
    delete enroll2;

    delete nPlan1;
    delete nPlan2;

    delete woPlan1;
    delete woPlan2;

    delete instructor1;
    delete instructor2;

    delete nutritionist1;
    delete nutritionist2;

    delete course1;
    delete course2;

    return 0;
}
```

Individual Contribution

Registration No	Name	Class Diagram	Codings
IT21020230	Siriwardana S. M. K. S	• RegisteredUser • Payment	• RegisteredUser.h • RegisteredUser.cpp • Payment.h • Payment.cpp
IT21022142	Abeysinghe Y. P. P	• WorkoutPlan • NutritionPlan	• WorkoutPlan.h • WorkoutPlan.cpp • NutritionPlan.h • NutritionPlan.cpp
IT20607432	Bandara S P M K	• Course • Enrollment	• Course.h • Course.cpp • Enrollment.h • Enrollment.cpp
IT21010026	Disanayake A. D. M. M. S	• Instructor Nutritionist	• Instructor.h • Instructor.cpp • Nutritionist.h • Nutritionist.cpp
IT21013614	D.N Bulathsinhala	• User • Staff	• User.h • User.cpp • Staff.h • Staff.cpp

****End of the document****