Topic : Online Fitness Trainer

Group no :MLB_04.01_05

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

Submission Date : 15/10/21

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
IT21004636	Perera G.N.P	0772490483
IT21140648	Wickramathilake K.W.S.D	0767620605
IT21129612	Pietersz N.J	0741158894
IT21127014	Dissanayake D.M.G.D	0763583357
IT21015458	Thiwanka W.A.D.A	0787805991

Exercise 1:

1) Requirements

- An online fitness trainer has a website that offers fitness programs
- There are 2 types of users for the system unregistered and registered members
- An unregistered member can visit the website and view the fitness programs offered by the personal trainer.
- They can become members by registering for the program. A user can only register for one program at a time
- To register they must make a payment by credit or debit card or PayPal. They also need to enter their personal details like name, contact details etc.
- Once the payment is validated and approved, they will become official members.
- users can also submit enquiries regarding any questions they may have, and they will receive a response via email
- Once the user logs in to their portal, they can view their workout plan and meal plans
- Many trainers can be assigned to a program and each member will have one trainer to coach them.
- Trainer can conduct live video workout sessions with member
- Trainer should customize meal plans and add workouts according to program as well as member's requirements
- System administrator can add or update programs and generate monthly reports on attendance and revenue
- Once the member completes a program, they can provide their review on the website as a testimonial. A member is accustomed to only review for a program.

2) Noun – verb analysis

Redundant	Attributes	Out of scope	Classes
• User • testimonial	• Credit, debit card, PayPal • name, contact details • attendance • revenue	• Website • Email • System admin	• Unregistered member • Registered member • Program • Payment • Trainer • Workout plan • Meal plan • Enquiries • reviews

3) Classes identified

1. Unregistered member
2. Registered member
3. Trainer
4. Program
5. Payment
6. Workout Plan
7. Enquiry
8. Report
9. Review
10. Meal plan

Exercise 2: CRC Cards

Class Name: Unregistered member	
Responsibilities:	Collaborations:
Register	

Class Name: Registered member	
Responsibilities:	Collaborations:
Log in	
Workout	Trainer

Class Name: Trainer	
Responsibilities:	Collaborations:
Conduct session	
Assign trainer	Program
Log in	
View member details	Registered member

Class Name: Program	
Responsibilities:	Collaborations:
Add program	
Update program	
Store program details	
View programs	Unregistered member

Class Name: Payment	
Responsibilities:	Collaborations:
Store payment details	
Store personal details	
Validate	

Class Name: Enquiry	
Responsibilities:	Collaborations:
Submit enquiry	
Receive email	Unregistered member

Class Name: Workout Plan	
Responsibilities:	Collaborations:
Add workout	
Update workout	
Assign workout	Trainer
View workout plan	Registered member

Class Name: Meal Plan	
Responsibilities:	Collaborations:
Add meal plan	
Update meal plan	
Assign meal plan	Trainer
View meal plan	Registered member

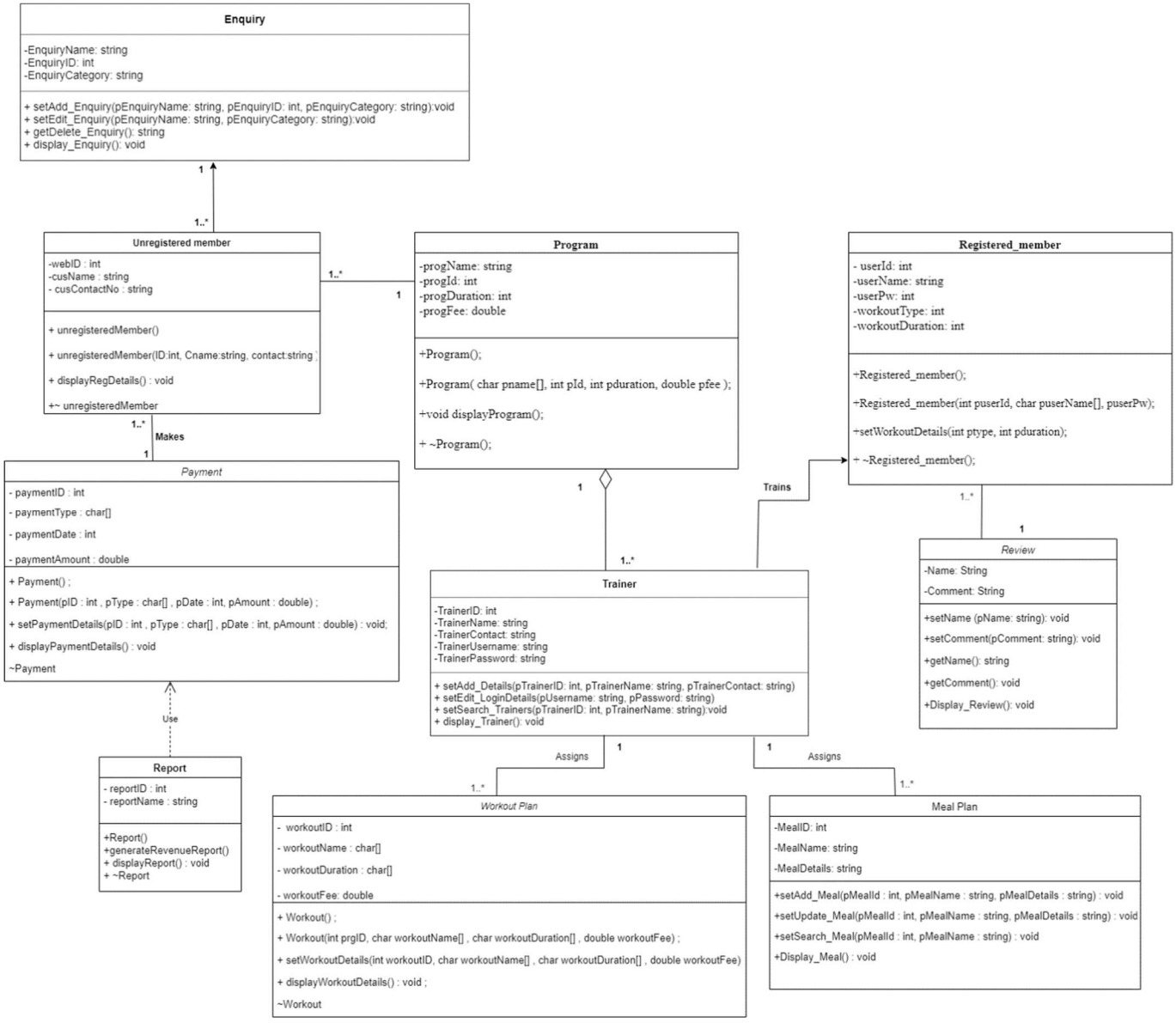
Class Name: Report	
Responsibilities:	Collaborations:
List of regular members	Registered member
Generate revenue report	Payment
Display report	

Class Name: Review	
Responsibilities:	Collaborations:
Submit review	Registered member
Delete review	

Individual Contribution

Registration No	Name	Classes designed
IT21004636	Perera G.N.P	Unregistered Program / Report
IT21140648	Wickramathilake K.W.S.D	Registered Member / Program
IT21129612	Pietersz N.J	Trainer / Enquiry
IT21127014	Dissanayake D.M.G.D	Payment / Workout Plan
IT21015458	Thiwanka W.A.D.A	Meal Plan / Reviews

Link to draw.io file(UML Diagram):
https://drive.google.com/file/d/1DjAZOGYTQmEVfXvacOCdZNyzt8rbIA2f/view?usp=sharing



Wickramathilake K.W.S.D

Unergistered_member.h

```
class Unregistered_member;

class program;


class Unregistered_member

{

 private:

                    int webId;

                    char cusName[20];

                    char cusDob[12];

                    int cusContactNo;

                    program* pro;


public:

                   Unregistered_member(int pWebId, char pname[], char pdob[], int pcontact);

                   void addProgram(program* pro);

                   void inputRegistration();

                   void displayRegDetails();

                   ~Unregistered_member();


};
```

Unregistered_member.cpp

```
#include<iostream>

#include "Unregistered_member.h"

#include<string.h>

using namespace std;
```

```
Unregistered_member::Unregistered_member(int pWebId, char pname[], char pdob[], int pcontact)
{
		webId = pWebId;
		strcpy(cusName, pname);
		strcpy(cusDob, pdob);
		cusContactNo = pcontact;
}

void Unregistered_member::addProgram(program* pro)
{

}

void Unregistered_member::inputRegistration()
{
}

void Unregistered_member::displayRegDetails()
{
		cout << "Web ID =" << webId << endl;
		cout << "Member name =" << cusName << endl;
		cout << "Customers date of birth = " << cusDob << endl;
		cout << "Contact No. = " << cusContactNo << endl;
}

Unregistered_member::~Unregistered_member()
{
```

}

Trainer.h

```
#pragma once

class trainer

{

private:

		int tainerId;

		char trainerName[20];

		int trainerContact;

		char trainerUsername[20];

		int trainerPw;


public:

		void setAddDetails(int ptrainerId, char ptrainerName[], int ptrainerContact);

		void setEditLoginDetails(char pUsername[], int pUserPw);

		void setSearchTrainers(int ptrainerId, char ptrainerName[]);

		void displayTrainer();
```

trainer.cpp

```
#include "trainer.h"

void trainer::setAddDetails(int ptrainerId, char ptrainerName[], int
ptrainerContact)
{
}

void trainer::setEditLoginDetails(char pUsername[], int pUserPw)
{
}

void trainer::setSearchTrainers(int ptrainerId, char ptrainerName[])
{
}
```

```
void trainer::displayTrainer()
{
}
```

Main.cpp

```
#include<iostream>

#include "Unregistered_member.h"

#include "program.h"

#include<string.h>

using namespace std;


int main()

{



}
```

Trainer.h

```
#pragma once
class trainer
{
private:
  int tainerId;
  char trainerName[20];
  int trainerContact;
  char trainerUsername[20];
  int trainerPw;

public:
  void setAddDetails(int ptrainerId, char ptrainerName[], int
ptrainerContact);
  void setEditLoginDetails(char pUsername[], int pUserPw);
  void setSearchTrainers(int ptrainerId, char ptrainerName[]);
  void displayTrainer();
```

trainer.cpp

```
#include "trainer.h"

void trainer::setAddDetails(int ptrainerId, char ptrainerName[], int
ptrainerContact)
{
}

void trainer::setEditLoginDetails(char pUsername[], int pUserPw)
{
}

void trainer::setSearchTrainers(int ptrainerId, char ptrainerName[])
{
}

void trainer::displayTrainer()
{
}
```

Dissanayake DMGD

Workout.h

```
class Workout{
private:
  int workoutID;
  char workoutName[20];
  char workoutDuration[20];
  double workoutFee;
public:
  Workout();
  Workout(int prgID, char prgName[], char prgDuration[], double prgFee);
  void setWorkoutDetails(int prgID, char prgName[], char prgDuration[], double
prgFee);
  void displayWorkoutDetails();
  ~Workout();
};
```

Workout.cpp

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

Workout::Workout(){
  workoutID = 0;
```

```
  strcpy(workoutName,"");

  strcpy(workoutDuration,"");
  workoutFee = 0.00;
}
Workout::Workout(int prgID, char prgName[], char prgDuration[], double
prgFee){
  workoutID = prgID;
  strcpy(workoutName,prgName);
  strcpy(workoutDuration ,prgDuration);
  workoutFee = prgFee;
}
void setWorkoutDetails(int prgID, char prgName[], char prgDuration[], double
prgFee){

}

void Workout::displayWorkoutDetails(){
  cout<<"Workout ID :"<<workoutID<<endl;
  cout<<"Workout Name :"<< workoutName<<endl;
  cout<<"Duration :"<<workoutDuration<<endl;
  cout<<"Fee :"<<workoutFee<<endl;
}
Workout::~Workout(){

}
```

Payment.h

```
class Payment{
private:
  int paymentID;
  char paymentType[20];
  int date;
  double paymentAmount;
public:
  Payment();
  Payment(int pID, char pType[], int pDate, double pAmount);
  void setPaymentDetails(int pID, char pType[], int pDate, double pAmount);
  void displayPaymentDetails();
  ~Payment();
};
```

Payment.cpp

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

```
using namespace std;

Payment::Payment(){
  paymentID = 0;
  strcpy(paymentType,"");
  date = 0;
  paymentAmount = 0.0;
}
Payment::Payment(int pID, char pType[], int pDate, double pAmount){
  paymentID = pID;
  strcpy(paymentType,pType);
  date = pDate;
  paymentAmount = pAmount;
}
void Payment::setPaymentDetails(int pID, char pType[], int pDate, double
pAmount){


}
void Payment::displayPaymentDetails(){
  cout<<"Payment ID :"<<paymentID<<endl;
  cout<<"Payment Type :"<<paymentType<<endl;
  cout<<"Date :"<< date<<endl;
  cout<<"Amount :"<<paymentAmount<<endl;
}
Payment::~Payment(){

}
```

Perera GNP

Report.h

```
class Report{
  private:
      int reportID;
      char reportName [20];

  public:
      Report(int rtID, char rName[])
      void generateRevenueReport(Payment* p);
      void displayReport();
      ~Report();
}
```

Report.cpp

```
Report::Report(int rID, char rName[]){
    reportNo = rID;
    strcpy(reportName,rName);
}

void Report::generateRevenueReport(Payment* p){

}

void Report::displayReport(){

}

Report::~Report(){

}
```

Main.cpp

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

int main(){
    Report* r1;

    r1 = new Report(121,"May 2021 Report");
    r1->generateRevenueReport(pay);
    r1->displayReport();
    delete r1;
}
```

Thiwanka W.A.D.A

Meal_Plan.h

```
class Meal_Plan{

  private:

    int MealID;
```

```
    std::string MealName;
    std::string MealDetails;

  public:
    void setAddMeal(int pMealID,std::string pMealName,std::string pMealDetails);
    void setUpdateMeal(int pMealID,std::string pMealName,std::string pMealDetails);
    void setSearchMeal(int pMealID,std::string pMealName);
    void Display();
};
```

Meal_plan.cpp

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

void Meal_Plan::setAddMeal(int pMealID,string pMealName,string pMealDetails){
  MealID=pMealID;
  MealName=pMealName;
  MealDetails=pMealDetails;
}
void Meal_Plan::setUpdateMeal(int pMealID,string pMealName,string pMealDetails){

}
void Meal_Plan::setSearchMeal(int pMealID,string pMealName){
```

```
}
void Display(){


}
```

Main.cpp

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

int main() {
  Meal_Plan Meal1;

  Meal1.Display();

  return 0;
}
```

Review.h

```
class Review{
  private:
    std::string Name;
    std::string Comment;

  public:
    void setName(std::string pName);
    void setComment(std::string pComment);
```

```
    std::string getName();
    std::string getComment();
    void Display();
};
```

Review.cpp

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



void Review::setName(string pName){
  Name=pName;
}
void Review::setComment(string pComment){
  Comment=pComment;
}
string Review::getName(){
  return Name;
}
string Review::getComment(){
  return Comment;
}
void Review::Display(){
  cout << "Your Review"<<endl<<endl;
  cout << "Name : " << getName() << endl;
  cout << "Comment : " <<getComment();
```

```
}
```

Main.cpp

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

int main() {
  Review myr;

  std::string Name,Comment;
  cout << "Enter Name : ";
  cin >>Name;
  cout << "Enter Comment : ";
  cin >>Comment;

  //seters assign Name, Comment
  myr.setName(Name);
  myr.setComment(Comment);

  myr.Display();

  return 0;
}
```

Pietersz NJ

Trainer.h

```
class Trainer{
  private:
  int trainerID;
  char trainerName[10];
  char trainerContactNo[10];
  char trainerUsername[15];
  char trainerPassword[10];

 public:
 void setAdd_Details(int pTrainerID, char pTrainerName[], char pTrainerContact[]);
 void setEdit_LoginDetails(char pUsername[], char pPassword[]);
 void setSearch_Trainers(int pTrainerID[], char pTrainerName[]);
 void displayTrainer_Details();
 ~Trainer();
};
=======
#pragma once
class trainer
{
private:
        int tainerId;
        char trainerName[20];
        int trainerContact;
        char trainerUsername[20];
        int trainerPw;
```

```
public:
        void setAddDetails(int ptrainerId, char ptrainerName[], int ptrainerContact);
        void setEditLoginDetails(char pUsername[], int pUserPw);
        void setSearchTrainers(int ptrainerId, char ptrainerName[]);
        void displayTrainer();
```

trainer.cpp

```
void Trainer::setEdit_LoginDetails(char pUsername[], char pPassword[]){
    strcpy(trainerUsername,pUsername);
    strcpy(trainerPassword,pPassword);
}

void Trainer::setSearch_Trainers(int pTrainerID[], char pTrainerName[]){

}

void Trainer::displayTrainer_Details(){
  cout << "Trainer ID: " << trainerID << endl;
  cout << "Trainer Name: " << trainerName << endl;
  cout << "Trainer Contact No: " << trainerContactNo << endl;
  cout << "Trainer Username: " << trainerUsername << endl;
  cout << "Trainer Password: " << trainerPassword << endl;

}
=======
```

```
#include "trainer.h"


void trainer::setAddDetails(int ptrainerId, char ptrainerName[], int ptrainerContact)

{

}


void trainer::setEditLoginDetails(char pUsername[], int pUserPw)

{

}


void trainer::setSearchTrainers(int ptrainerId, char ptrainerName[])

{

}


void trainer::displayTrainer()

{

}
```

Program.h

```
class Program{
  private:
  int programID;
  char programName[20];
  int programDuration;
  double programFee;


  public:
```

```
 void set_Program(char pName[], int pID, int pDuration, double pFee);
 void displayProgram();
 ~Program();

}
=======
#define SIZE
class Unregistered_member;
class program;

class program

{
private:
        char progName[20];
        int progId;
        int progDuration;
        double progFee;
        Unregistered_member * UM[SIZE];

public:
        program();
        program(char pname[], int pId, int pduration, double pfee);
        void displayProgram();
        ~ program();

};
```

Program.cpp

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

void Program::set_Program(char pName[], int pID, int pDuration, double pFee){
  programID = pID;
  strcpy(programName,pName);
  programDuration = pDuration;
  programFee = pFee;

}

void Program::displayProgram(){
   cout << "Program ID: " << programID << endl;
   cout << "Program Name: " << programName << endl;
   cout << "Program Duration: " << programDuration << endl;
   cout << "Program Fee: " << programFee << endl;

}

Program::~Program(){
   cout << "Successful Joined" << endl;
}
=======
#include<iostream>
```

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

program::program()
{
}

program::program(char pname[], int pId, int pduration, double pfee)
{
}

void program::displayProgram()
{
}

program::~program()
{
}
```

Enquiry.h

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

class Enquiry {
  private:
  int enquiryID;
```

```
 char enquiryName[20];
 char enquiryCategory[30];

 public:
 void setadd_enquiry(int penID, char penName[], char penCategory[]);
 void setedit_enquiry(char penName[], penCategory[]);
 void getdelete_enquiry();
 void dispaly_enquiry();
 ~Enquiry();

};
```

Enquiry.cpp

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



void Enquiry::setadd_enquiry(int penID, char penName[], char penCategory[]){
  enquiryID = penID;
  strcpy(enquiryName,penName);
  strcpy(enquiryCategory,penCategory);
}

void Enquiry::setedit_enquiry(char penName[], penCategory[]){
```

```
}

void Enquiry::getdelete_enquiry(){
   return delete_enquiry;
}

void Enquiry::dispaly_enquiry(){
   cout << "Enquiry ID: " << enquiryID << endl;
   cout << "Enquiry Name: " << enquiryName << endl;
   cout << "Enquiry Category: " << enquiryCategory << endl;
}

Enquiry::~Enquiry(){
   cout << "Waiting for your Response" << endl;
}
```

Unregistered_member.h

```
class Unregistered_member;
class program;

class Unregistered_member
{
 private:
        int webId;
        char cusName[20];
        char cusDob[12];
        int cusContactNo;
```

```
        program* pro;

public:
        Unregistered_member(int pWebId, char pname[], char pdob[], int pcontact);
        void addProgram(program* pro);
        void inputRegistration();
        void displayRegDetails();
        ~Unregistered_member();

};
```

Unregistered_member.cpp

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



Unregistered_member::Unregistered_member(int pWebId, char pname[], char pdob[], int pcontact)
{
        webId = pWebId;
        strcpy(cusName, pname);
        strcpy(cusDob, pdob);
        cusContactNo = pcontact;
}

void Unregistered_member::addProgram(program* pro)
```

```
{


}

void Unregistered_member::inputRegistration()
{
}

void Unregistered_member::displayRegDetails()
{
        cout << "Web ID =" << webId << endl;
        cout << "Member name =" << cusName << endl;
        cout << "Customers date of birth = " << cusDob << endl;
        cout << "Contact No. = " << cusContactNo << endl;
}

Unregistered_member::~Unregistered_member()
{


}
```