**Project Title: Fitness Record System**

**Functional Requirements Document**



**Submitted To: Ma’am Romana**

**Submitted By**

**Adeel ur Rehman**

**Enroll : 01-235192-106**

**Inshirah Nasir**

**Enroll : 01-235192-096**

**BS IT 3B**

TABLE OF CONTENTS

1 INTRODUCTION 3

2 PURPOSE OF THE SYSTEM 3

3 FUNCTIONAL REQUIREMENTS 3

3.1 USE CASE DIAGRAMS………………………………………………………...4

4 GRAPHICAL USER INTERFACE REQUIREMENTS……………………………………..4

* **INTRODUCTION**

A Fitness Record System which will provide various functions for a front desk guy, Instructor and client.

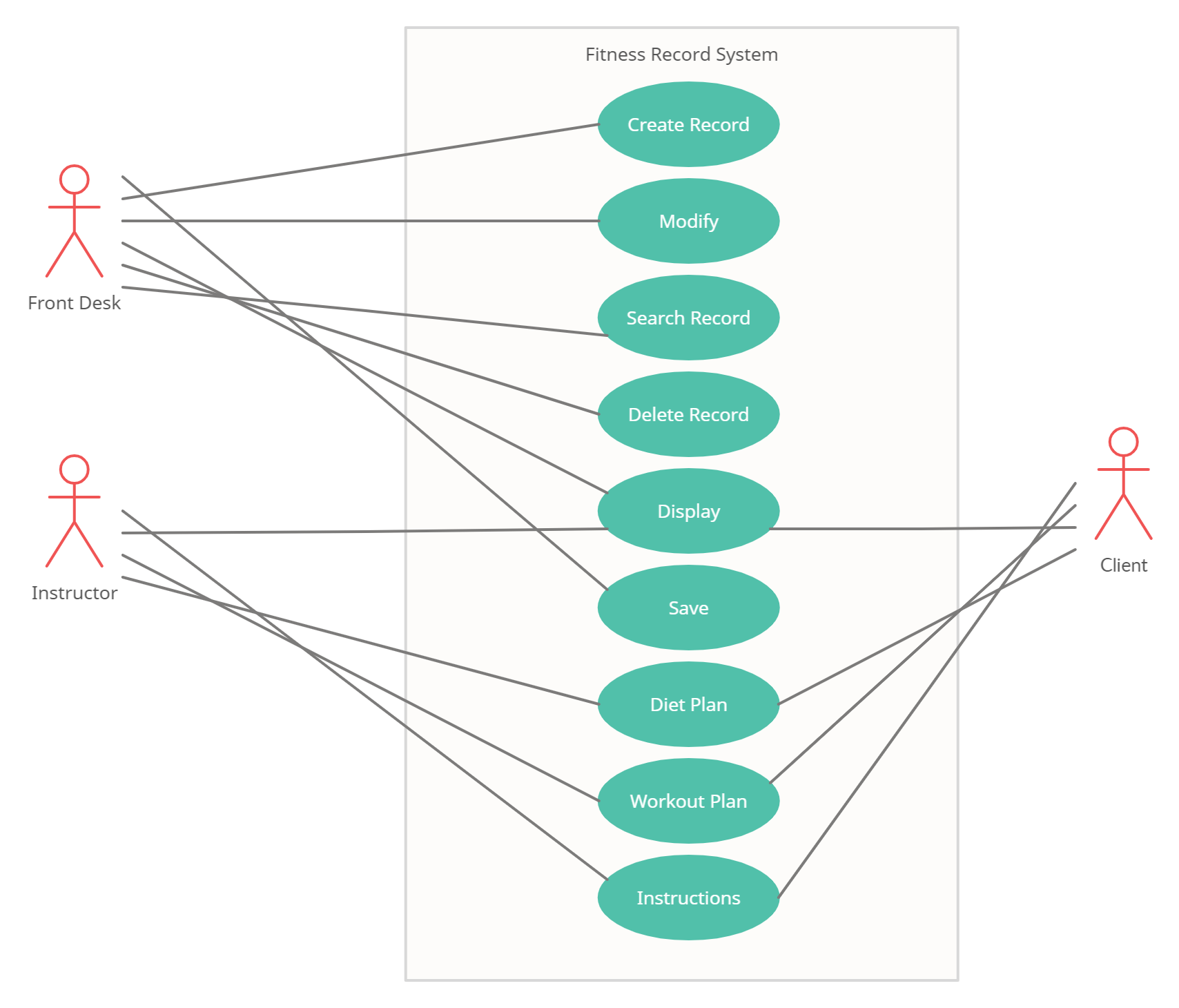
# PURPOSE OF THE SYSTEM

The system can be used in any Gym to keep the record of each client. The data can be saved in external files and displayed from it. It will provide multiple functionalities to the three users according to their authority.

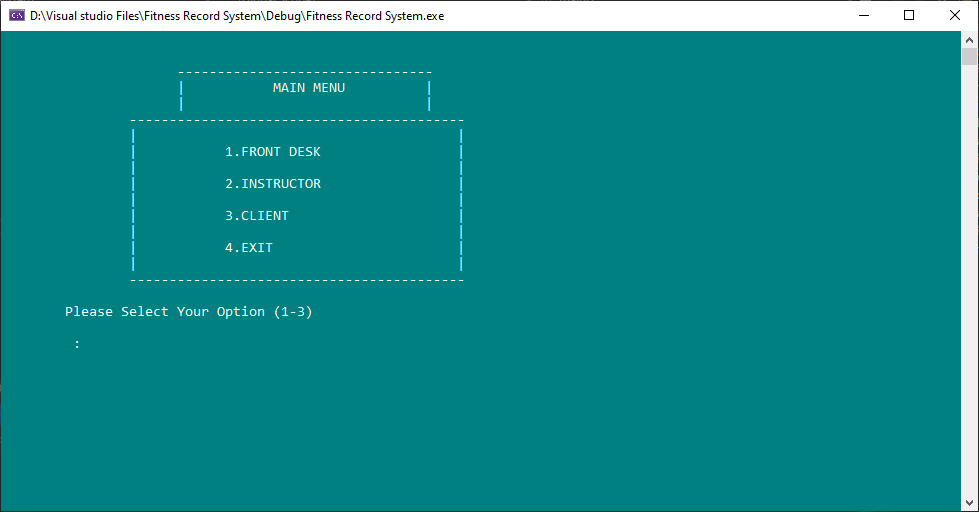
1. **FUNCTIONAL REQUIREMENTS**

The system will ask the user to select among three choices that is Front Desk User , Instructor or Client. Each user has their own authority. A front desk user has more authority and access which will required a username and password. He can register, search, save, display, delete or modify clients. An instructor can assign diet plans, workout plans, send instructions to client or see the progress of client. A client can see his diet plan, workout plans, instructor’s message or his profile.

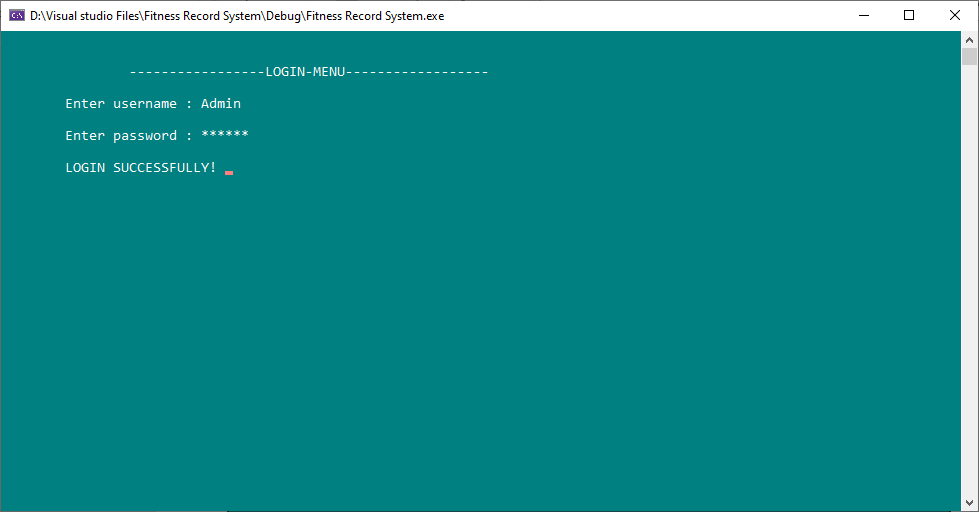
**3.1 USE CASE DIAGRAMS**

****

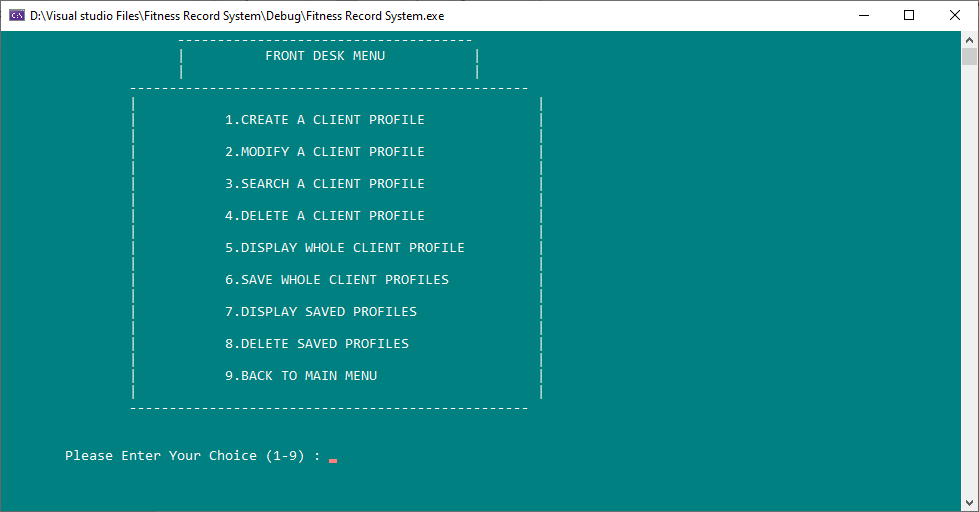
1. **GRAPHICAL USER INTERFACE REQUIREMENTS**



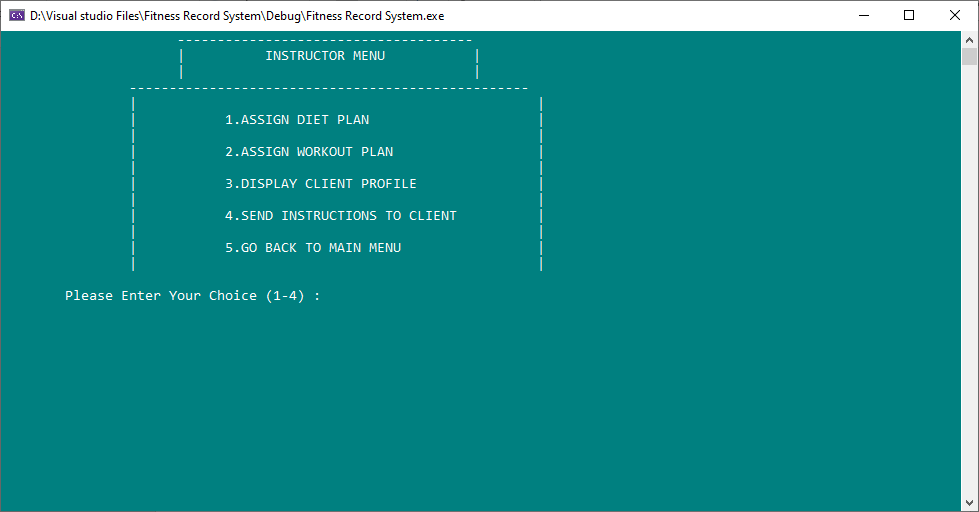
MAIN MENU

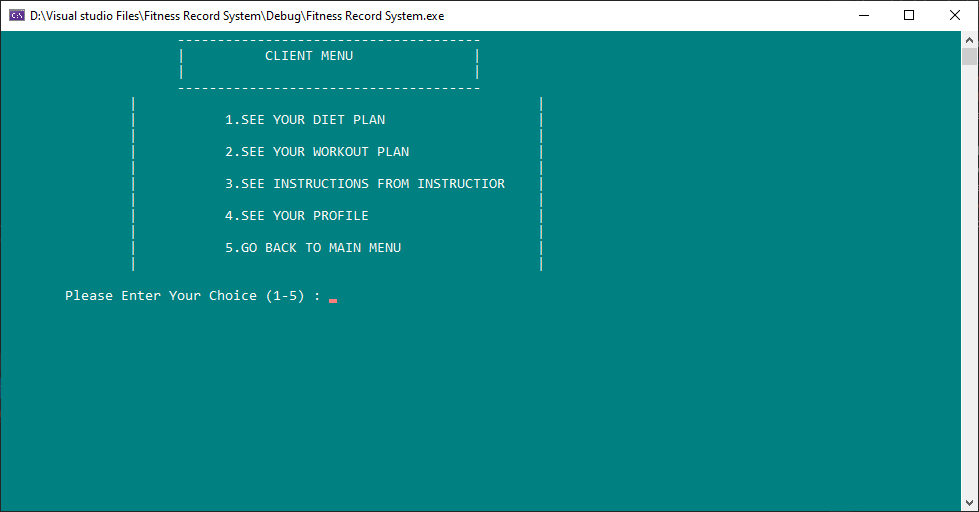


Login Screen for Front Desk Guy



Front Desk Menu

Instructor Menu



Client Menu

**Project Code**

**Datalist.h**

#ifndef Datalist\_HEADER

#define Datalist\_HEADER

#include <string>

#include <fstream>

#include <conio.h>

#include <iostream>

using namespace std;

class Datalist

{

public:

Datalist();

string name, email, address;

int age, id;

float weight;

float height;

float BMI;

string diet,workout,message;

Datalist\* next;

};

#endif

**Datalist.cpp**

#include "dataList.h"

Datalist::Datalist()

{

diet = "Not assigned";

workout = "Not assigned";

message = "No instructions yet";

}

**FRS.h**

#ifndef FRS\_HEADER

#define FRS\_HEADER

#include "dataList.h"

class FRS:public Datalist

{

private:

Datalist\* search();

public:

Datalist\* head;

FRS();

bool isEmpty();

void createProfile();

void searchProfile();

void modify();

void Delete();

void display();

void frontMenu();

void save();

void fileRecord();

void deleteoldRecord();

void dietPlans();

void DisplayClient();

void instructorMenu();

void workoutPlan();

void instructions();

void clientMenu();

};

#endif

**FRS.cpp**

#include "FRS.h"

FRS::FRS()

{

head = 0;

}

bool FRS::isEmpty()

{

if (head == 0)

return true;

else return false;

}

void FRS::createProfile()

{

Datalist\* temp = new Datalist, \* ptr = new Datalist;

string name1, email1, address1;

int age1, id1;

float weight1, height1;

float BM;

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | PROFILE REGISTERATION |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

cout << "\n\t Enter your id : ";

cin >> id1;

cout << "\n\t Enter your name : ";

cin >> name1;

cin.ignore();

cout << "\n\t Enter your age : ";

cin >> age1;

cout << "\n\t Enter your email : ";

cin >> email1;

cout << "\n\t Enter your address : ";

cin >> address1;

cin.ignore();

cout << "\n\t Enter your weight in kg : ";

cin >> weight1;

cout << "\n\t Enter your height in meters : ";

cin >> height1;

cout << "\n\t----------Profile Successfully Created!----------\n\n";

BM = (weight1 / (height1 \* height1));

if (isEmpty())

{

head = temp;

temp->id = id1;

temp->name = name1;

temp->age = age1;

temp->email = email1;

temp->address = address1;

temp->weight = weight1;

temp->height = height1;

temp->BMI = BM;

temp->id = id1;

temp->next = NULL;

}

else

{

temp = head;

while (temp->next != NULL)

{

temp = temp->next;

}

ptr->id = id1;

ptr->name = name1;

ptr->age = age1;

ptr->email = email1;

ptr->address = address1;

ptr->weight = weight1;

ptr->height = height1;

ptr->BMI = BM;

ptr->id = id1;

ptr->next = NULL;

temp->next = ptr;

}

}

Datalist\* FRS::search()

{

Datalist\* temp = head, \* ptr = new Datalist;

int ch, idd;

bool flag = false;

string Name;

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | CLIENT Sign-in |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

if (temp == NULL)

{

return NULL;

}

cout << "\n\t Select a choice : \n";

cout << "\n\t 1) Sign in using id : ";

cout << "\n\t 2) Sign in using name : \n\n\t : ";

cin >> ch;

if (ch == 1)

{

cout << "\n\t Enter your id to SIGN-IN : \n\t ";

cin >> idd;

while (temp != NULL)

{

if (temp->id == idd)

{

flag = true;

return temp;

}

temp = temp->next;

}

if (flag == false)

{

return NULL;

}

}

else if (ch == 2)

{

cout << "\n\t Enter your name to SIGN-IN : \n\t ";

cin >> Name;

while (temp != NULL)

{

if (temp->name == Name)

{

flag = true;

return temp;

}

temp = temp->next;

}

if (flag == false)

{

return NULL;

}

}

}

void FRS::searchProfile()

{

Datalist\* temp = head, \* ptr = new Datalist;

int ch, idd;

bool flag = false;

string Name;

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | PROFILE SEARCH |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

if (temp == NULL)

{

cout << "\n\n\n\t LIST IS EMPTY ";

cout << "\n\n\n\t THERE IS NO DATA TO SEARCH! ";

return;

}

cout << "\n\t Select a choice : \n";

cout << "\n\t 1) Search by id : ";

cout << "\n\t 2) Search by name : \n\n\t : ";

cin >> ch;

if (ch == 1)

{

cout << "\n\t Enter id to search : \n\t ";

cin >> idd;

while (temp != NULL)

{

if (temp->id == idd)

{

flag = true;

cout << "\n\t ID : " << temp->id;

cout << "\n\t NAME : " << temp->name;

cout << "\n\t AGE : " << temp->age;

cout << "\n\t EMAIL : " << temp->email;

cout << "\n\t ADDRESS : " << temp->address;

cout << "\n\t Weight (KG) : " << temp->weight;

cout << "\n\t HEIGHT (m) : " << temp->height;

cout << "\n\t BODY MASS INDEX : " << temp->BMI;

cout << "\n\t DIET PLAN : " << temp->diet;

cout << "\n\t WORKOUT PLAN : " << temp->workout;

cout << "\n\t INSTRUCTIONS FOR CLIENT : " << temp->message;

}

temp = temp->next;

}

if (flag == false)

{

cout << "\n\t Profile not found! ";

}

}

else if (ch == 2)

{

cout << "\n\t Enter name to search : \n\t ";

cin >> Name;

while (temp != NULL)

{

if (temp->name == Name)

{

flag = true;

cout << "\n\t ID : " << temp->id;

cout << "\n\t NAME : " << temp->name;

cout << "\n\t AGE : " << temp->age;

cout << "\n\t EMAIL : " << temp->email;

cout << "\n\t ADDRESS : " << temp->address;

cout << "\n\t Weight (KG) : " << temp->weight;

cout << "\n\t HEIGHT (m) : " << temp->height;

cout << "\n\t BODY MASS INDEX : " << temp->BMI;

}

temp = temp->next;

}

if (flag == false)

{

cout << "\n\t Profile not found! ";

}

}

}

void FRS::modify()

{

Datalist\* temp = head;

string name;

char cch;

bool flag = false;

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | PROFILE MODIFICATION |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

if (temp == NULL)

{

cout << "\n\n\n\t LIST IS EMPTY ";

cout << "\n\n\n\t THERE IS NO DATA TO MODIFY! ";

return;

}

cout << "\n\t Enter your name to get the details : \n\n\t : ";

cin >> name;

while (temp != NULL)

{

if (temp->name == name)

{

flag = true;

cout << "\n\t\t------------Current Details-------------\n";

cout << "\n\t ID : " << temp->id;

cout << "\n\t NAME : " << temp->name;

cout << "\n\t AGE : " << temp->age;

cout << "\n\t EMAIL : " << temp->email;

cout << "\n\t ADDRESS : " << temp->address;

cout << "\n\t Weight (KG) : " << temp->weight;

cout << "\n\t HEIGHT (m) : " << temp->height;

cout << "\n\t BODY MASS INDEX : " << temp->BMI;

cout << "\n\n\t Are you sure to modify the details? (y/n) : \n\t : ";

cin >> cch;

if (cch == 'y' || cch == 'Y')

{

system("cls");

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | ENTER NEW DETAILS |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

cout << "\n\t Enter New Name : ";

cin >> temp->name;

cin.ignore();

cout << "\n\t Enter New Age : ";

cin >> temp->age;

cout << "\n\t Enter New Email : ";

cin >> temp->email;

cout << "\n\t Enter New Address : ";

cin >> temp->address;

cout << "\n\t Enter New Weight : ";

cin >> temp->weight;

cout << "\n\t Enter New Height : ";

cin >> temp->height;

float BM = (temp->weight / (temp->height \* temp->height));

temp->BMI = BM;

cout << "\n\t Details Successfully Updated! ";

}

else break;

}

temp = temp->next;

}

if (flag == false)

{

cout << "\n\t Profile not found! ";

}

}

void FRS::Delete()

{

Datalist\* temp = head, \* ptr = new Datalist, \* ptr2 = head;

int id;

char ch = 'n';

bool flag = false;

if (temp == NULL)

{

cout << "\n\n\n\t LIST IS EMPTY ";

cout << "\n\n\n\t THERE IS NO DATA TO DELETE! ";

return;

}

cout << "\n\t Enter Profile id to delete : \n\t ";

cin >> id;

while (temp != NULL)

{

if (temp->id == id)

{

flag = true;

cout << "\n\t Details found! ";

cout << "\n\t ID : " << temp->id;

cout << "\n\t NAME : " << temp->name;

cout << "\n\t AGE : " << temp->age;

cout << "\n\t EMAIL : " << temp->email;

cout << "\n\t ADDRESS : " << temp->address;

cout << "\n\t Weight (KG) : " << temp->weight;

cout << "\n\t HEIGHT (m) : " << temp->height;

cout << "\n\t BODY MASS INDEX : " << temp->BMI;

cout << "\n\t DIET PLAN : " << temp->diet;

cout << "\n\t WORKOUT PLAN : " << temp->workout;

cout << "\n\t INSTRUCTIONS FOR CLIENT : " << temp->message;

cout << "\n\t Are you sure to delete this profile? (y/n) ";

cin >> ch;

if (ch == 'y' || ch == 'Y')

{

cout << "\n\t Profile Deleted Successfully! ";

break;

}

else

{

return;

}

}

temp = temp->next;

}

if (flag == false)

{

cout << "\n\t Profile not found! ";

return;

}

if (head->id == id)

{

head = head->next;

}

while (ptr2->id != id)

{

ptr = ptr2;

ptr2 = ptr2->next;

}

if (ptr2->next == NULL)

{

ptr->next = NULL;

delete ptr2;

}

else

{

ptr->next = ptr2->next;

delete ptr2;

}

}

void FRS::display()

{

Datalist\* temp = head;

if (head == NULL)

{

cout << "\n\n\n\t LIST IS EMPTY! ";

cout << "\n\n\t THERE IS NO DATA TO DISPLAY! ";

}

else

{

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | DISPLAYING PROFILES |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

while (temp != NULL)

{

cout << "\n\t-------------------------------";

cout << "\n\t ID : " << temp->id;

cout << "\n\t NAME : " << temp->name;

cout << "\n\t AGE : " << temp->age;

cout << "\n\t EMAIL : " << temp->email;

cout << "\n\t ADDRESS : " << temp->address;

cout << "\n\t Weight (KG) : " << temp->weight;

cout << "\n\t HEIGHT (m) : " << temp->height;

cout << "\n\t BODY MASS INDEX : " << temp->BMI;

cout << "\n\t DIET PLAN : " << temp->diet;

cout << "\n\t WORKOUT PLAN : " << temp->workout;

cout << "\n\t INSTRUCTIONS FOR CLIENT : " << temp->message;

cout << endl;

temp = temp->next;

}

}

}

void FRS::frontMenu()

{

char ch;

char chh = 'y';

do

{

cout << "\t\t -------------------------------------" << endl;

cout << "\t\t | FRONT DESK MENU |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t--------------------------------------------------" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 1.CREATE A CLIENT PROFILE |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 2.MODIFY A CLIENT PROFILE |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 3.SEARCH A CLIENT PROFILE |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 4.DELETE A CLIENT PROFILE |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 5.DISPLAY WHOLE CLIENT PROFILE |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 6.SAVE WHOLE CLIENT PROFILES |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 7.DISPLAY SAVED PROFILES |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 8.DELETE SAVED PROFILES |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 9.BACK TO MAIN MENU |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t--------------------------------------------------" << endl;

cout << endl;

cout << "\n\tPlease Enter Your Choice (1-9) : ";

cin >> ch;

switch (ch)

{

case '1':

{

char op = 'y';

do

{

system("cls");

createProfile();

cout << "\n\t Register another client? (y/n) : ";

cin >> op;

if (op != 'y' || op != 'Y')

system("cls");

} while (op == 'y' || op == 'Y');

break;

}

case '2':

system("cls");

cin.ignore();

modify();

\_getch();

system("cls");

break;

case '3':

system("cls");

searchProfile();

\_getch();

system("cls");

break;

case'4':

system("cls");

Delete();

\_getch();

system("cls");

break;

case '5':

system("cls");

display();

\_getch();

system("cls");

break;

case'6':

system("cls");

save();

\_getch();

system("cls");

break;

case '7':

system("cls");

fileRecord();

\_getch();

system("cls");

break;

case '8':

system("cls");

deleteoldRecord();

\_getch();

system("cls");

break;

case '9':

system("cls");

break;

default: cout << "\n\t\t INVALID CHOICE! ";

break;

}

} while (ch != '9');

}

void FRS::save()

{

Datalist\* ptr = head;

ofstream out;

out.open("record.txt", ios::app);

if (!out.is\_open())

{

cout << "\n\t Unable to open the file! ";

}

else

{

if (ptr == NULL)

{

cout << "\n\n\n\t THERE IS NO DATA TO SAVE! ";

cout << "\n\n\t FOR SAVING DATA, FIRST YOU NEED TO ENTER THE DATA! \n";

}

else

{

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | SAVING PROFILES IN FILES |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

cout << endl << endl;

while (ptr != NULL)

{

out << " Id : " << ptr->id << endl;

out << " Name : " << ptr->name << endl;

out << " Age : " << ptr->age << endl;

out << " Address : " << ptr->address << endl;

out << " Weight : " << ptr->weight << endl;

out << " Height : " << ptr->height << endl;

out << " Body Mass Index : " << ptr->BMI << endl;

out << " Diet Plan : " << ptr->diet << endl;

out << " Workout Plan : " << ptr->workout << endl;

out << "--------------------" << endl << endl;

ptr = ptr->next;

}

cout << "\n\n\n\t THE DATA IS SAVED SUCCESSFULLY! \t \n\n";

}

}

out.close();

}

void FRS::fileRecord()

{

ifstream in("record.txt", ios::ate); // ios::ate means open at end

if (in.tellg() == 0)

{

cout << "\n\n\t\t FILE IS EMPTY! ";

cout << "\n\n\t\t THERE IS NO DATA IN THE FILE TO DISPLAY! ";

return;

}

in.close();

string line;

ifstream read;

read.open("record.txt");

if (!read)

cout << "\n\t Unable to open the file! ";

else

{

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | DISPLAYING SAVED PROFILES |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

cout << endl << endl;

while (!read.eof())

{

getline(read, line);

cout << line << endl;

}

}

read.close();

}

void FRS::deleteoldRecord()

{

ofstream del;

ifstream read("record.txt", ios::ate); // ios::ate means open at end

if (read.tellg() == 0)

{

cout << "\n\n\t\t FILE IS EMPTY! ";

cout << "\n\n\t\t THERE IS NO DATA IN THE FILE TO DELETE! ";

return;

}

del.open("record.txt");

if (!del)

cout << "\n\t Unable to open the file! ";

else

{

cout << "\t\t ----------------------------------------" << endl;

cout << "\t\t | DELETING OLD RECORD |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t ----------------------------------------" << endl;

del << "";

cout << "\n\n\t\t Old Record Successfully Deleted!";

}

del.close();

return;

}

// For GYM Instructor

void FRS::dietPlans()

{

Datalist\* temp = head;

string name;

char cch;

bool flag = false;

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | ASSIGNING DIET PLAN |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

if (temp == NULL)

{

cout << "\n\n\n\t LIST IS EMPTY ";

cout << "\n\n\n\t THERE IS NO DATA TO SHOW! \n";

return;

}

cout << "\n\t Enter client name to get the details : \n\n\t : ";

cin >> name;

while (temp != NULL)

{

if (temp->name == name)

{

flag = true;

cout << "\n\t\t------------Current Details-------------\n";

cout << "\n\t ID : " << temp->id;

cout << "\n\t NAME : " << temp->name;

cout << "\n\t AGE : " << temp->age;

cout << "\n\t EMAIL : " << temp->email;

cout << "\n\t ADDRESS : " << temp->address;

cout << "\n\t Weight (KG) : " << temp->weight;

cout << "\n\t HEIGHT (m) : " << temp->height;

cout << "\n\t BODY MASS INDEX : " << temp->BMI;

cout << "\n\t DIET PLAN : " << temp->diet;

cout << "\n\t WORKOUT PLAN : " << temp->workout;

cout << "\n\t INSTRUCTIONS FOR CLIENT : " << temp->message;

cout << "\n\n\t Do You Want To Assign Diet Plan (y/n) : \n\t : ";

cin >> cch;

cin.ignore();

if (cch == 'y' || cch == 'Y')

{

system("cls");

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | ENTER DIET PLAN |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

cout << "\n\t Enter Diet Plan : ";

getline(cin, temp->diet);

}

else break;

}

temp = temp->next;

}

if (flag == false)

{

cout << "\n\t Profile not found! ";

}

}

void FRS::DisplayClient()

{

Datalist\* temp = head;

if (head == NULL)

{

cout << "\n\n\n\t LIST IS EMPTY! ";

cout << "\n\n\t THERE IS NO DATA TO DISPLAY! ";

}

else

{

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | DISPLAYING PROFILES |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

while (temp != NULL)

{

cout << "\n\t-------------------------------";

cout << "\n\t ID : " << temp->id;

cout << "\n\t NAME : " << temp->name;

cout << "\n\t AGE : " << temp->age;

cout << "\n\t CURRENT WEIGHT (KG) : " << temp->weight;

cout << "\n\t HEIGHT (m) : " << temp->height;

cout << "\n\t BODY MASS INDEX : " << temp->BMI;

cout << "\n\t DIET PLAN : " << temp->diet;

cout << "\n\t WORKOUT PLAN : " << temp->workout;

cout << "\n\t INSTRUCTIONS FOR CLIENT : " << temp->message;

cout << endl;

temp = temp->next;

}

}

}

void FRS::instructorMenu()

{

char ch = 'y';

do

{

cout << "\t\t -------------------------------------" << endl;

cout << "\t\t | INSTRUCTOR MENU |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t--------------------------------------------------" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 1.ASSIGN DIET PLAN |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 2.ASSIGN WORKOUT PLAN |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 3.DISPLAY CLIENT PROFILE |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 4.SEND INSTRUCTIONS TO CLIENT |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 5.GO BACK TO MAIN MENU |" << endl;

cout << "\t\t| |" << endl;

cout << "\n\tPlease Enter Your Choice (1-4) : ";

cin >> ch;

switch (ch)

{

case '1':

{

system("cls");

dietPlans();

\_getch();

system("cls");

break;

}

case '2':

system("cls");

workoutPlan();

\_getch();

system("cls");

break;

case '3':

system("cls");

DisplayClient();

\_getch();

system("cls");

break;

case '4':

system("cls");

instructions();

\_getch();

system("cls");

break;

case '5':

system("cls");

break;

}

} while (ch != '5');

}

void FRS::workoutPlan()

{

Datalist\* temp = head;

string name;

char cch;

bool flag = false;

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | ASSIGNING WORKOUT PLAN |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

if (temp == NULL)

{

cout << "\n\n\n\t LIST IS EMPTY ";

cout << "\n\n\n\t THERE IS NO DATA TO SHOW! \n";

return;

}

cout << "\n\t Enter client name to get the details : \n\n\t : ";

cin >> name;

while (temp != NULL)

{

if (temp->name == name)

{

flag = true;

cout << "\n\t\t------------Current Details-------------\n";

cout << "\n\t ID : " << temp->id;

cout << "\n\t NAME : " << temp->name;

cout << "\n\t AGE : " << temp->age;

cout << "\n\t Weight (KG) : " << temp->weight;

cout << "\n\t HEIGHT (m) : " << temp->height;

cout << "\n\t BODY MASS INDEX : " << temp->BMI;

cout << "\n\t DIET PLAN : " << temp->diet;

cout << "\n\t WORKOUT PLAN : " << temp->workout;

cout << "\n\t INSTRUCTIONS FOR CLIENT : " << temp->message;

cout << "\n\n\t Do You Want To Assign Workout Plan (y/n) : \n\t : ";

cin >> cch;

cin.ignore();

if (cch == 'y' || cch == 'Y')

{

system("cls");

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | ENTER WORKOUT PLAN |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

cout << "\n\t Enter Workout Plan : ";

getline(cin, temp->workout);

}

else break;

}

temp = temp->next;

}

if (flag == false)

{

cout << "\n\t Profile not found! ";

}

}

void FRS::instructions()

{

Datalist\* temp = head;

string name;

char cch;

bool flag = false;

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | SENDING INSTRUCTIONS TO CLIENT |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

if (temp == NULL)

{

cout << "\n\n\n\t LIST IS EMPTY ";

cout << "\n\n\n\t THERE IS NO DATA TO SHOW! \n";

return;

}

cout << "\n\t Enter client name to get the details : \n\n\t : ";

cin >> name;

while (temp != NULL)

{

if (temp->name == name)

{

flag = true;

cout << "\n\t\t------------Current Details-------------\n";

cout << "\n\t ID : " << temp->id;

cout << "\n\t NAME : " << temp->name;

cout << "\n\t AGE : " << temp->age;

cout << "\n\t Weight (KG) : " << temp->weight;

cout << "\n\t HEIGHT (m) : " << temp->height;

cout << "\n\t BODY MASS INDEX : " << temp->BMI;

cout << "\n\t DIET PLAN : " << temp->diet;

cout << "\n\t WORKOUT PLAN : " << temp->workout;

cout << "\n\t INSTRUCTIONS FOR CLIENT : " << temp->message;

cout << "\n\n\t Do You Want To Send Instructions (y/n) : \n\t : ";

cin >> cch;

cin.ignore();

if (cch == 'y' || cch == 'Y')

{

system("cls");

cout << "\t\t -------------------------------------------" << endl;

cout << "\t\t | INSTRUCTIONS FOR CLIENT |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t -------------------------------------------" << endl;

cout << "\n\t Enter Your Instructions : ";

getline(cin, temp->message);

}

else break;

}

temp = temp->next;

}

if (flag == false)

{

cout << "\n\t Profile not found! ";

}

}

// For Client

void FRS::clientMenu()

{

char ch = 'y';

Datalist \*n=search();

if (head == NULL)

{

cout << "\n\t\t LIST IS EMPTY! ";

cout << "\n\t\t NO CLIENT DATA EXIST! ";

\_getch();

system("cls");

}

else if (n == NULL)

{

cout << "\n\t\t COULDN'T FIND DETAILS! ";

cout << "\n\t\t ID OR NAME DOESN'T EXIST! ";

\_getch();

system("cls");

}

else

{

system("cls");

do

{

cout << "\t\t --------------------------------------" << endl;

cout << "\t\t | CLIENT MENU |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t --------------------------------------" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 1.SEE YOUR DIET PLAN |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 2.SEE YOUR WORKOUT PLAN |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 3.SEE INSTRUCTIONS FROM INSTRUCTIOR |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 4.SEE YOUR PROFILE |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 5.GO BACK TO MAIN MENU |" << endl;

cout << "\t\t| |" << endl;

cout << "\n\tPlease Enter Your Choice (1-5) : ";

cin >> ch;

switch (ch)

{

case '1':

{

system("cls");

cout <<"\n\t YOUR DIET PLAN : "<<n->diet;

\_getch();

system("cls");

break;

}

case '2':

system("cls");

cout << "\n\t YOUR WORKOUT PLAN : " << n->workout;

\_getch();

system("cls");

break;

case '3':

system("cls");

cout << "\n\t INSRUCTIONS FOR YOU : " << n->message;

\_getch();

system("cls");

break;

case '4':

system("cls");

cout << "\n\t\t------------YOUR PROFILE-------------\n";

cout << "\n\t ID : " << n->id;

cout << "\n\t NAME : " << n->name;

cout << "\n\t AGE : " << n->age;

cout << "\n\t Weight (KG) : " << n->weight;

cout << "\n\t HEIGHT (m) : " << n->height;

cout << "\n\t BODY MASS INDEX : " << n->BMI;

\_getch();

system("cls");

break;

case '5':

system("cls");

break;

}

} while (ch != '5');

}

}

**Main.cpp**

#include"FRS.h"

#include "dataList.h"

using namespace std;

int main()

{

FRS h;

char ch;

do

{

system("color 3F");

cout << endl << endl;

cout << "\t\t --------------------------------" << endl;

cout << "\t\t | MAIN MENU |" << endl;

cout << "\t\t | |" << endl;

cout << "\t\t------------------------------------------" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 1.FRONT DESK |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 2.INSTRUCTOR |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 3.CLIENT |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t| 4.EXIT |" << endl;

cout << "\t\t| |" << endl;

cout << "\t\t------------------------------------------" << endl;

cout << "\n\tPlease Select Your Option (1-3) \n\n \t : ";

cin >> ch;

cout << endl;

switch (ch)

{

case '1':

{

string username = "Admin", uname;

string pass1 = "Bahria", pass;

int i = 0;

do

{

char ps;

system("cls");

pass = "";

cout << "\n\n\t\t-----------------LOGIN-MENU------------------\n";

cout << "\n\tEnter username : ";

cin >> uname;

cout << "\n\tEnter password : ";

ps = \_getch();

while (ps != 13) {// 13 is ascii code for enter button

pass.push\_back(ps);

cout << "\*";

ps = \_getch();

}

if (uname == username && pass == pass1)

{

cout << "\n\n\tLOGIN SUCCESSFULLY! ";

\_getch();

system("cls");

h.frontMenu();

}

else

{

system("cls");

cout << "\n\t INVALID USERNAME OR PASSWORD! TRY AGAIN ";

++i;

}

} while (uname != username && pass != pass1 && i < 3);

break;

}

case '2':

system("cls");

h.instructorMenu();

break;

case '3':

system("cls");

h.clientMenu();

break;

case '4':

exit(0);

default:

system("cls");

cout << "\a";

}

} while (ch != 4);

\_getch();

}