UNIVERSITY OF CALGARY
DEPARTMENT OF COMPUTER SCIENCE

CPSC 471: Database Management Systems

Final Report

Group #37
Noor (Rose) Muhammed (30071227)
Haider Tawfik (30097912)
YuJeong Chae (30091378)

Abstract

In recent years there has been an increased public interest in health and wellness. Rates of obesity are only increasing and therefore, more and more people are looking for help losing weight and staying fit. We devised a food and fitness tracker that helps users with this goal. Once logged in, our app allows users to input the foods they eat and the exercise they perform, as well as input their body measurements. The system shows a visual of how many calories they have burned and consumed today in proportion to their goal amounts. The system also allows users to connect with the relevant professionals (e.g. personal trainers and nutritionists), set up appointments with them, and receive recommended diet or exercise regimens from them, and view feedback from them. In addition to connecting with professionals, users can also add other users as friends. Taken together, this app will give users the tools they need to set them up for success in their weight loss and fitness goals.

Introduction

Over the pandemic, 48% of the population gained a significant amount of weight [1]. Obesity affects 26.4% of Canadians [2] -- and this number is only increasing. With this comes an increased interest in weight loss and physical fitness. In order for people to effectively lose weight, they must monitor their food consumed (and the associated calories) and the calories burned. People need to be able to record their physical measurements such as waist circumference and weight in order to gauge their progress and adjust their meal plan if necessary. Personal trainers and nutritionists must also be able to view their clients' progress. Many people use a pen-and-paper system for tracking. There exist some apps, such as MyFitnessPal [3] or Noom [4], for helping users track weight loss; however, the vast majority of these are paid. They also lack some functionality such as connecting with personal trainers and nutritionists.

Description

Our program is basically designed to allow each user to check their physical status. Each user can check the amount of calories s/he consumed on each day in a graphical format along with the workout routine and recommended meal plans by the professionals. By doing so, users can pay more attention to their health with our program.

Project Design

There are two types of users in our system: (1) clients who can communicate with professionals and record their exercise and eating habits, and (2) professionals who communicate with these clients by consulting and giving feedback. Clients and professionals must log in with a username and password. If they do not have an account, they can create one.

Once logged in, clients can view the recommended meal plan and workout routine assigned by their nutritionist or personal trainer, as well as any feedback comments those professionals may have, by clicking the name in the "professionals" list. They can schedule an appointment with each professional by clicking the "request" button. The earliest available date will be assigned to both the client and the professional.

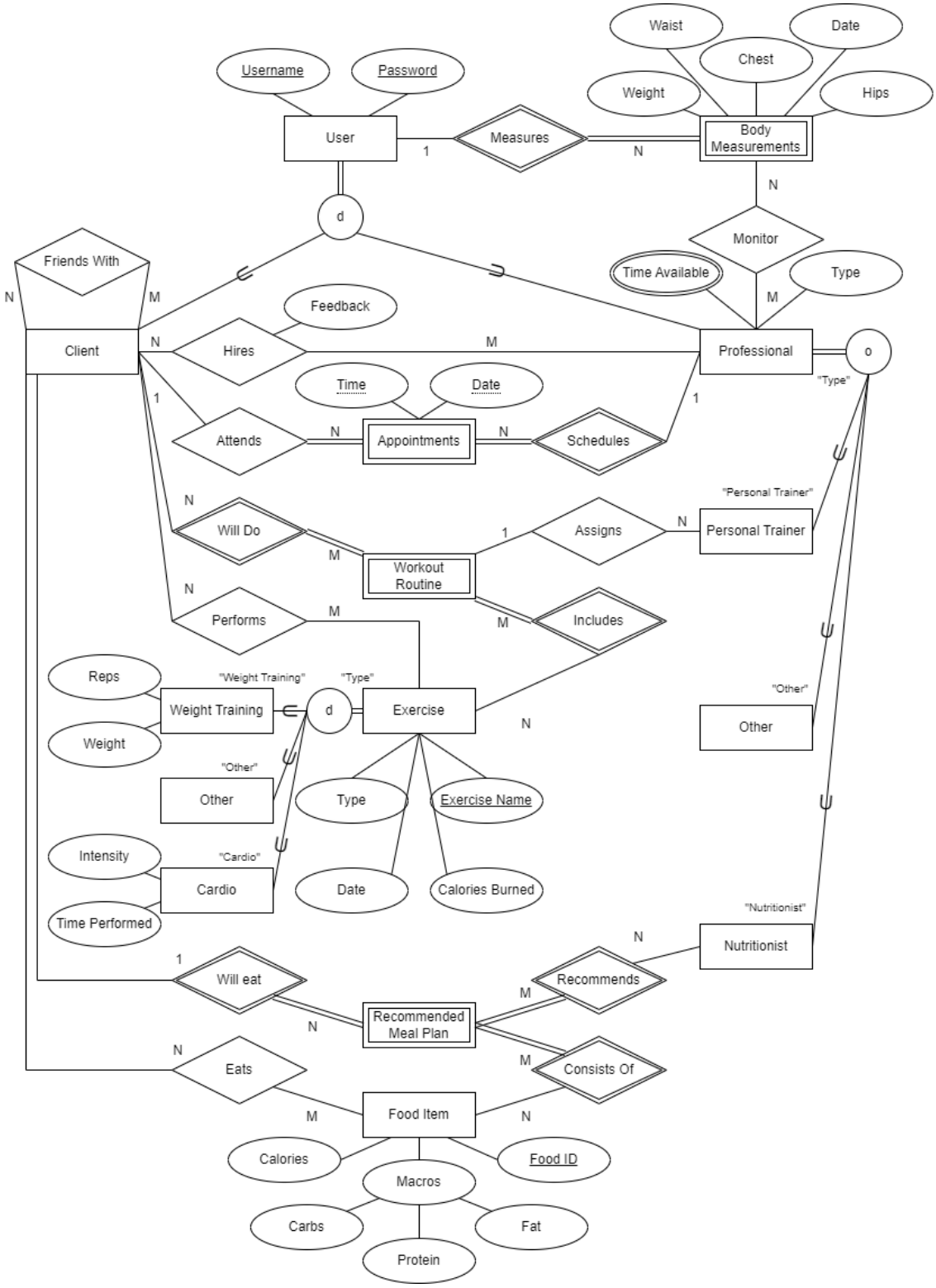
Professionals have a yearly interactive calendar, whereby clicking on the date allows them to view the appointments that they have scheduled for that day, and the time for the appointments. Professionals can also quickly view the next upcoming appointment at the top of the calendar so they do not have to check each day for the next appointment. Once clicked on a date, a professional can add a time that they are available for an appointment to that date.

Professionals can view their client's progress and give feedback and recommendations. On the side, the professional can view all their clients, and by clicking on their name they can see their most recent body measurements. If the professional is a personal trainer, they can add workouts to the client's workout plan, or remove them as they see fit. Or, if the professional is a nutritionist, they can add foods to their meal plan or remove them. Professionals can also send messages to their clients to provide feedback on their progress, which the client can view on their home page.

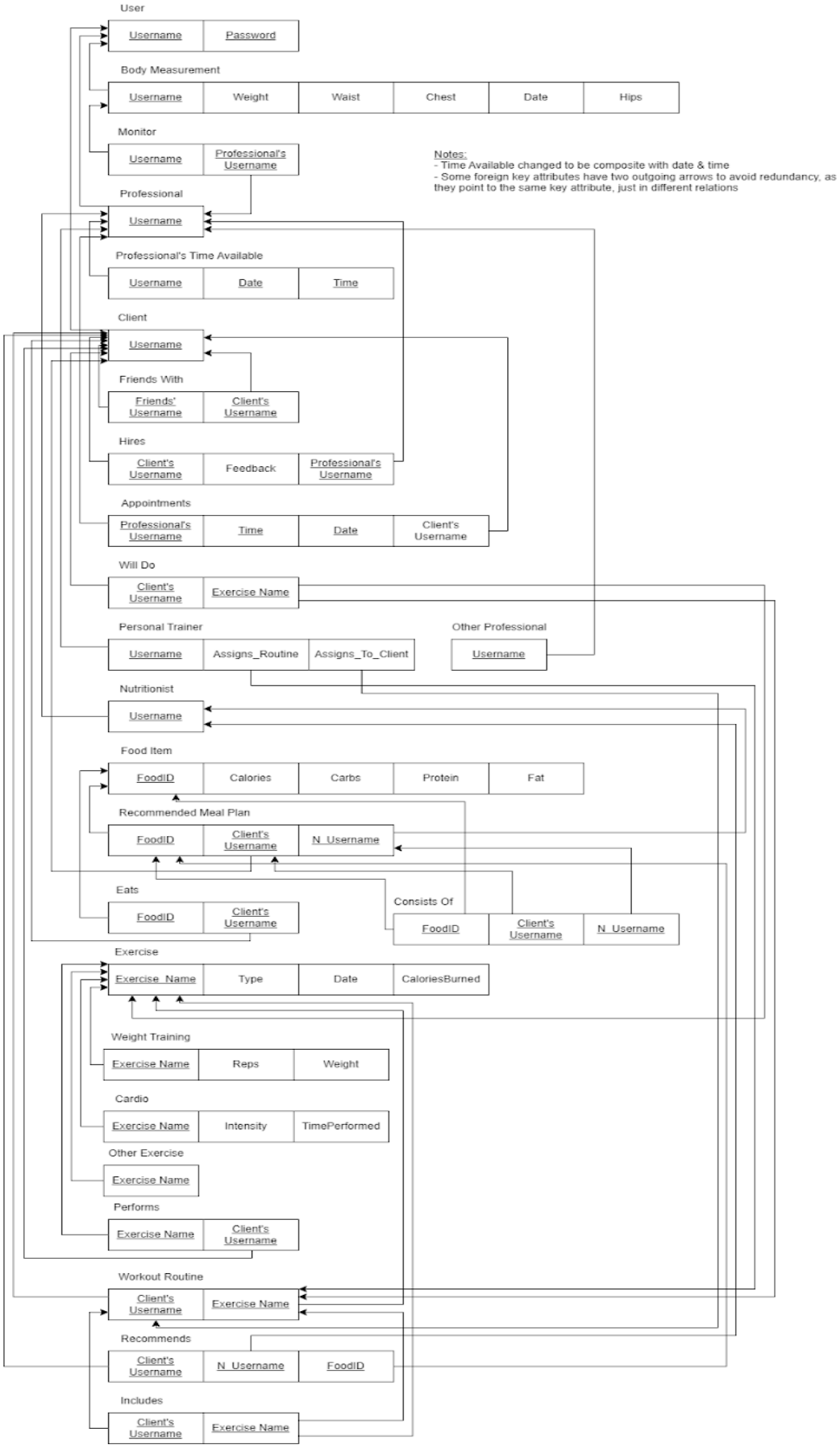
Clients can also input their diet and exercise from options in the database. The calories associated with the food item or exercise will be added to the client's total for the day. A client can also view a donut chart that compares the calories consumed in proportion to their goal, allowing them to keep in mind how much more they can eat before they over-do it. This chart is updated each time a client inputs a food that they have consumed on that specific day to provide real time progress. The calorie goal depends on the recommended meal plan and their completed exercise. For example, if the nutritionist gives a meal plan with a total of 2000 calories, and the user inputs 300 calories worth of exercise, then the goal calorie amount is 2300. There is also a circular progress bar, where each client will also have the ability to keep track of how many calories they have burned in relation to the assigned workout, and is updated each time a new exercise is input for that day. For example, if they are assigned 300 calories worth of exercise and they actually burn 150 calories, then the progress bar will be half-filled. Clients are also able to view their most recent body measurements, as this provides an idea of the progress they have made since the most recent measurement. Clients can quickly update their body measurements through the "edit profile" tab. In addition to connecting with professionals, users can also add other users as friends.

Implementation

EERD:

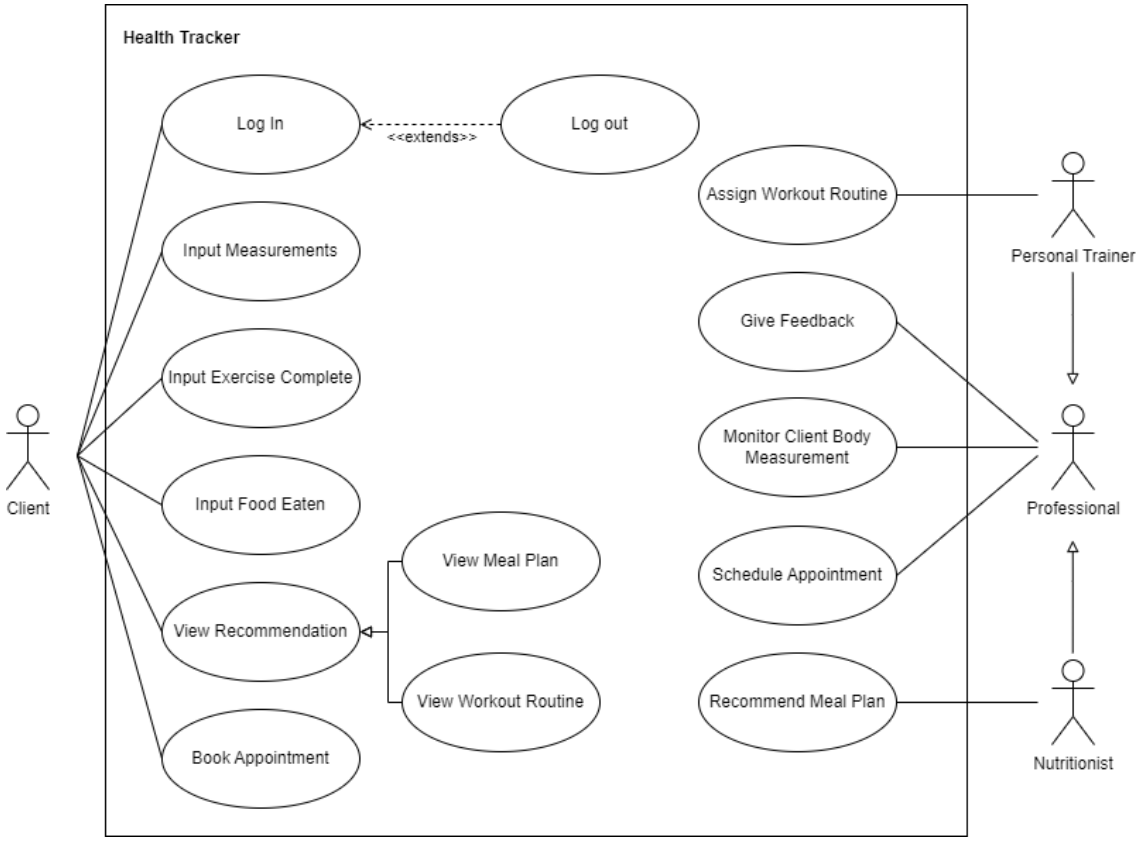


Relational Model:

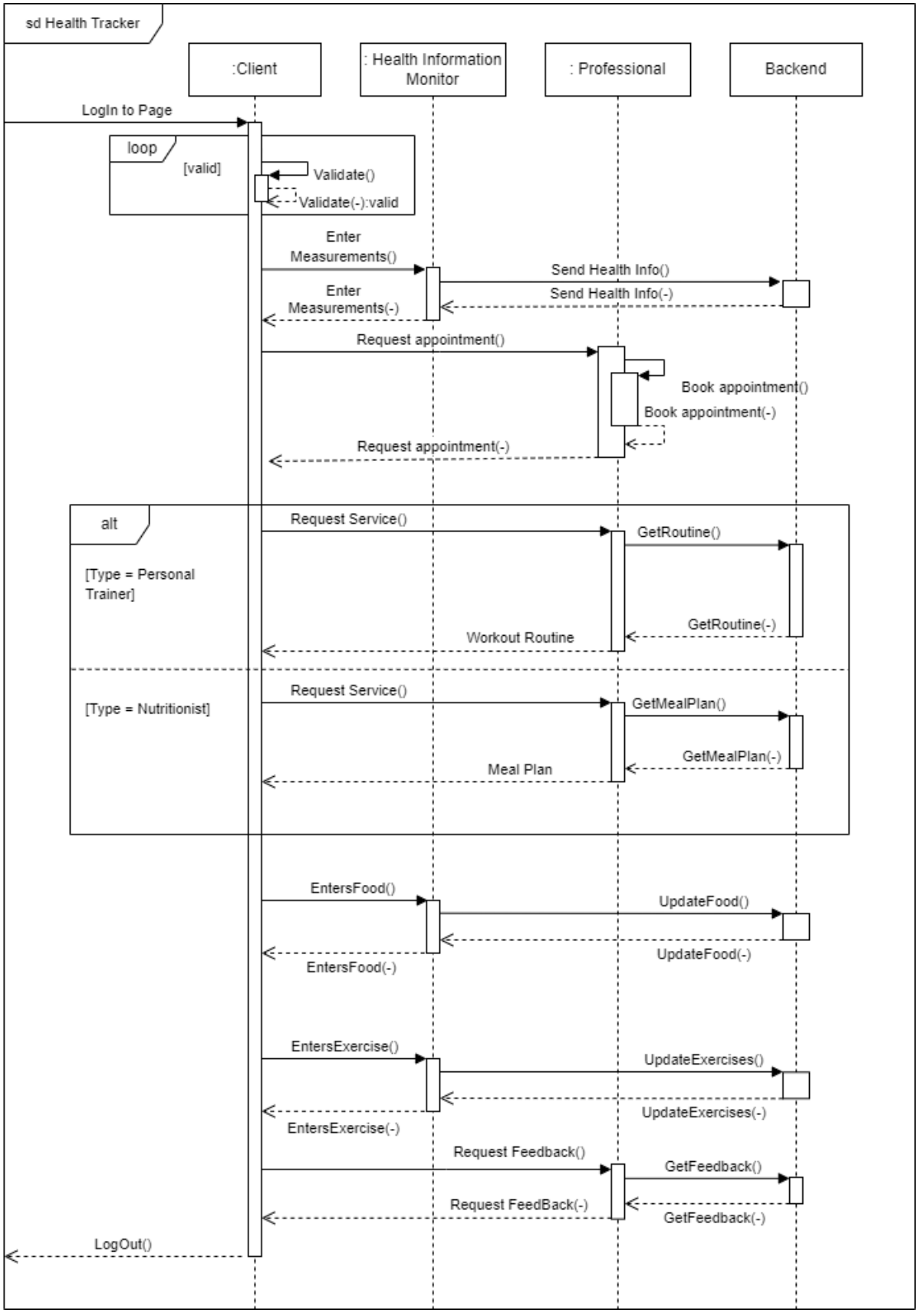


For converting our EERD to a RM we followed the algorithm as described in tutorials and lectures. The only unusual decision we made when following the algorithm to convert the EERD to a Relational model is when we decided to have multiple outgoing arrows from an attribute to avoid redundancy.

UML Diagram:



Sequence Diagram:



Looking at the sequence diagram, if the client succeeds in logging in, his or her body measurement may be updated and the updated information is stored in the database. Each client can make appointments with professionals and receive workout routines or diets from different types of professionals. This is recorded in the database along with the exercise performed by the client and the calories consumed.

The DBMS selected for our project implementation is MySQL and the following SQL statements are used for each of the transactions:

1. Users (both client & professionals)
    a. Check if the user is a client. Otherwise, they are professionals. (Avoid SQL injection)
        i. `$stmtprof = $connection -> prepare('SELECT * FROM Client WHERE Username = ?');`
        ii. `$stmtprof -> bind_param('s', $_POST['username']);`
        iii. `$stmtprof -> execute();`
        iv. `$stmtprof -> store_result();`
    b. Check if the user is a trainer or a nutritionist. (Avoid SQL injection)
        i. `$stmtprof = $connection -> prepare('SELECT * FROM Personal_Trainer WHERE Username = ?');`
        ii. `$stmtprof -> bind_param('s', $_POST['username']);`
        iii. `$stmtprof -> execute();`
        iv. `$stmtprof -> store_result();`
    c. Check if the user already exists. (Avoid SQL injection)
        i. `$stmt = $connection -> prepare('SELECT * FROM User WHERE Username = ?');`
        ii. `$stmt -> bind_param('s', $_POST['newusername']);`
        iii. `$stmt -> execute();`
        iv. `$stmt -> store_result();`
    d. Insert a new user (Avoid SQL injection) with his/her body measurements
        i. `$stmt = $connection -> prepare('INSERT INTO User (username, password) VALUES (?,?)');`
        ii. `$stmt -> bind_param('ss', $_POST['newusername'], $_POST['password1']);`
        iii. `$stmt -> execute();`
        iv. `$stmt = $connection -> prepare('INSERT INTO Body_Measurement (Username, Date, Weight, Waist, Chest, Hips) VALUES (?,?,?,?,?,?)');`
        v. `$stmt -> bind_param('ssdddd', $_POST['newusername'], $today, $temp, $temp, $temp, $temp);`
        vi. `$stmt -> execute();`
    e. Insert the corresponding user into the client. (Avoid SQL injection)

i. `$stmt = $connection -> prepare('INSERT INTO Client (username) VALUES (?)');`
ii. `$stmt -> bind_param('s', $_POST['newusername']);`
iii. `$stmt -> execute();`

f. Insert the corresponding user into the professionals. (Avoid SQL injection)
i. `$stmt = $connection -> prepare('INSERT INTO Professional (username) VALUES (?)');`
ii. `$stmt -> bind_param('s', $_POST['newusername']);`
iii. `$stmt -> execute();`

g. Insert the corresponding user into the personal trainer. (Avoid SQL injection)
i. `$stmt = $connection -> prepare('INSERT INTO Professional (username) VALUES (?)');`
ii. `$stmt -> bind_param('s', $_POST['newusername']);`
iii. `$stmt -> execute();`

h. Insert the corresponding user into the nutritionist. (Avoid SQL injection)
i. `$stmt = $connection -> prepare('INSERT INTO Nutritionist (username) VALUES (?)');`
ii. `$stmt -> bind_param('s', $_POST['newusername']);`
iii. `$stmt -> execute();`

i. To update the user's body measurement
i. `"UPDATE Body_Measurement SET Date = '{$date}', Weight = {$weight}, Waist = {$waist}, Chest = {$chest}, Hips = {$hips} WHERE Username = '{$username}'"`

2. Clients
a. To display body measurements' status
i. `"SELECT * FROM Body_Measurement where Username = '".$username."'"`

b. To display their upcoming appointments with professionals
i. `"SELECT * FROM Appointments where Client_Username = '".$username."'"`

c. To add exercise names they performed to the list/database
i. `"SELECT * FROM Exercise where Exercise_Name = '".$newExercise."'"`
ii. `"INSERT INTO Performs (Exercise_Name, Client_Username, Date) VALUES ('".$newExercise."','".$username."','".$today."')"`

d. To add food IDs they consumed to the list/database

i. `"SELECT * FROM Food_Item where FoodID = '".$newFood."'"`

ii. `"INSERT INTO Eats (FoodID, Client_Username, Date) VALUES ('".$newFood."','".$username."','".$today."')"`

e. To add friends' usernames that they added to the list/database

i. `"SELECT * FROM Client where Username = '".$newFriend."'"`

ii. `"INSERT INTO Friends_With (Friends_Username, Client_Username) VALUES ('".$newFriend."','".$username."')"`

f. To remove exercise names they performed from the list/database

i. `"SELECT * FROM Performs where Client_Username = '".$username."'"`

ii. `"DELETE FROM Performs WHERE Exercise_Name = '".$deleted."'"`

g. To remove food IDs they consumed from the list/database

i. `"SELECT * FROM Eats where Client_Username = '".$username."'"`

ii. `"DELETE FROM Eats WHERE FoodID = '".$deleted."'"`

h. To remove friends' usernames that they added from the list/database

i. `"SELECT * FROM Friends_With where Client_Username = '".$username."'"`

ii. `"DELETE FROM Friends_With WHERE Friends_Username = '".$deleted."'"`

i. To get the performed exercises to display on the web page

i. `"SELECT * FROM Performs where Client_Username = '".$username."'"`

ii. To get the total amount of calories consumed by performed exercises

1. `"SELECT * FROM Exercise where Exercise_Name = '".$performedExercise[$i]."'"`

j. To get the workout routine to display on the web page

i. `"SELECT * FROM Workout_Routine where Client_Username = '".$username."'"`

ii. To get the total amount of calories to be consumed according to a given exercise routine

1. `"SELECT * FROM Exercise where Exercise_Name = '".$allExercise[$i]."'"`

k. To get the consumed food to display on the web page

i. `"SELECT * FROM Eats where Client_Username = '".$username."'"`

ii. To get the total amount of calories consumed by the client

1. "SELECT * FROM Food_Item where FoodID = '".$consumedFood[$i]."'"

l. To get the recommended meal plan to display on the web page
   i. "SELECT * FROM Recommended_Meal_Plan where Client_Username = '".$username."'"
   ii. To get the total amount of calories to be consumed according to a given diet
      1. "SELECT * FROM Food_Item where FoodID = '".$allFood[$i]."'"

m. To get the friends
   i. "SELECT * FROM Friends_With where Client_Username = '".$username."'"

n. To get the professionals
   i. "SELECT * FROM Hires where Client_Username = '".$username."'"

o. To schedule an appointment
   i. "INSERT INTO Appointments (Professional_Username, Time, Date, Client_Username) VALUES ('".$prof."','".$availables['Time']."','".$availables['Date']."','".$client."')"

3. Professionals
   a. Get next Appointment
      i. "SELECT * FROM Appointments WHERE Professional_Username = '".$username."' AND Date >= '".$cdate."' ORDER BY Date ASC, Time";
   b. Get clients of the professional, and their body measurements
      i. "SELECT Client_Username FROM Hires where Professional_Username = '".$username."'"
      ii. SELECT * FROM Body_Measurement where Username = '".$user."'
   c. Get appointment on selected date
      i. "SELECT * FROM Appointments WHERE Professional_username = '".$username."' AND Date = '".$cdate."'"
   d. Add Time Available
      i. "INSERT INTO `Professional's_times_available` (Username, Date, Time) VALUES ('".$username."', '".$_GET['Cdate']."','".$avail."')"
   e. Add a Client
      i. "INSERT INTO `Hires` (Client_Username, Professional_Username) VALUES ('".$addClient."', '".$username."')"
   f. Get Appointment on selected day

i. `"SELECT * FROM Body_Measurement where Username = '".$user."'"`

g. Add an exercise for a workout routine for a client
   i. `"INSERT INTO workout_routine (Exercise_Name, Client_Username) values('".$to_add."','".$client."')"`

h. Get client current workout routine
   i. `"SELECT * FROM workout_routine where Client_Username = '".$client."'"`

i. Add feedback for a client
   i. `"UPDATE Hires SET Feedback = '".$to_add."' where Professional_Username = '".$username."' AND Client_Username = '".$client."'"`

j. Add food for a meal plan for a client
   i. `"INSERT INTO recommended_meal_plan (FoodID, Client_Username, Nutritionists_Username) values('".$to_add."','".$client."', '".$username."')"`

k. remove an exercise for a workout routine for a client
   i. `"DELETE FROM workout_routine where Client_Username = '".$client."' and Exercise_Name = '".$toRemove."'"`

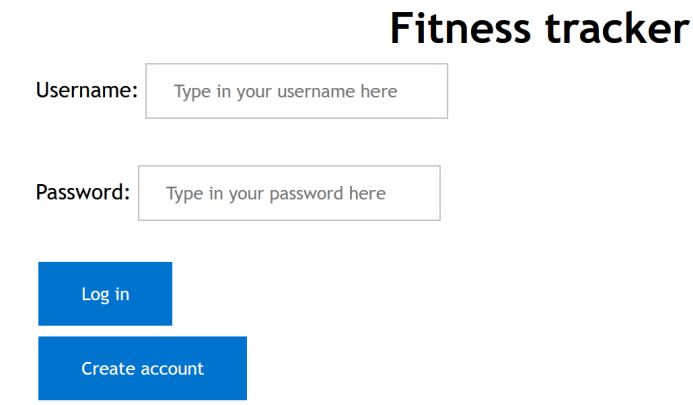
l. remove a food for a meal plan for a client
   i. `"DELETE FROM recommended_meal_plan where FoodId = '".$toRemove."' and Client_Username = '".$client."' and  Nutritionists_Username = '".$username."'"`

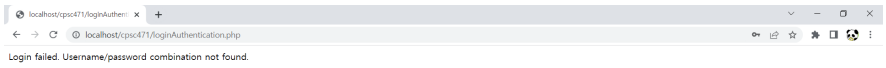
m. Get the current workout plan
   i. `"SELECT * FROM recommended_meal_plan where Client_Username = '".$client."' and Nutritionists_Username = '".$username."'"`

User Manual

Log in page:

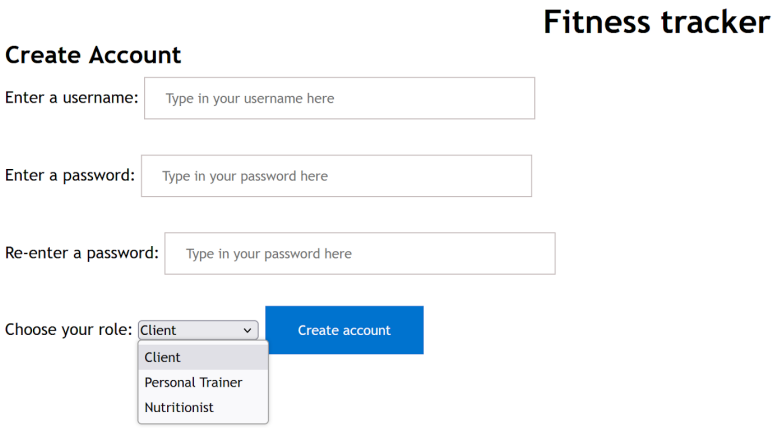
1. Users, both clients and professionals, must first log in with their username and password. If either the username or password are incorrect, they are notified of the error. If the username or password fields are empty, they are also notified of the error. If the user does not have an account, they simply click “Create account” and are brought to the registration page.





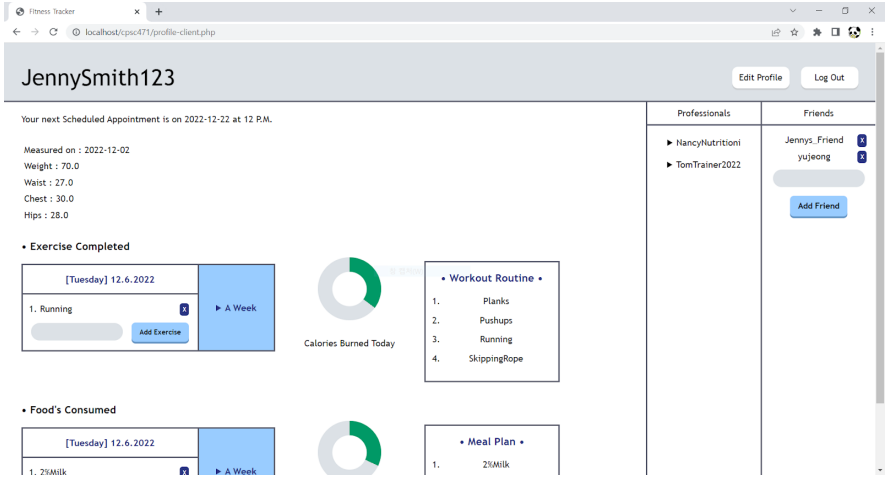
Create an account page:

1. If a user does not have an account, they can create one easily. They must enter a username, a password, and re-enter the password for verification. They must also indicate which user type they are: either “client”, “personal trainer”, or “nutritionist.” If the username they enter already exists, the user will be prompted to choose a different one. If their passwords do not match, they will be notified of the error.

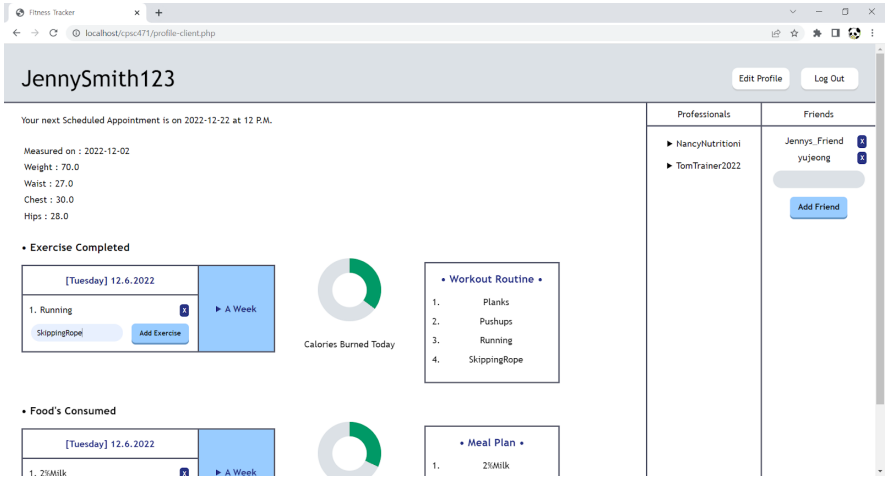


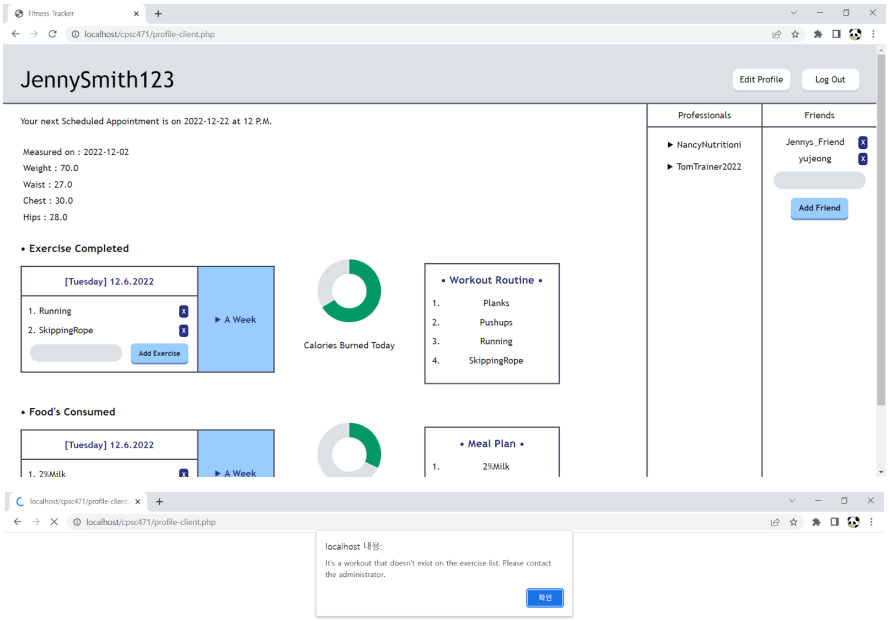
Profile page for clients:

1. Clients can check his/her body measurements on the left side of our webpage along with the list of exercises they completed and food they consumed. The list of professionals and friends are on the right side of the page.

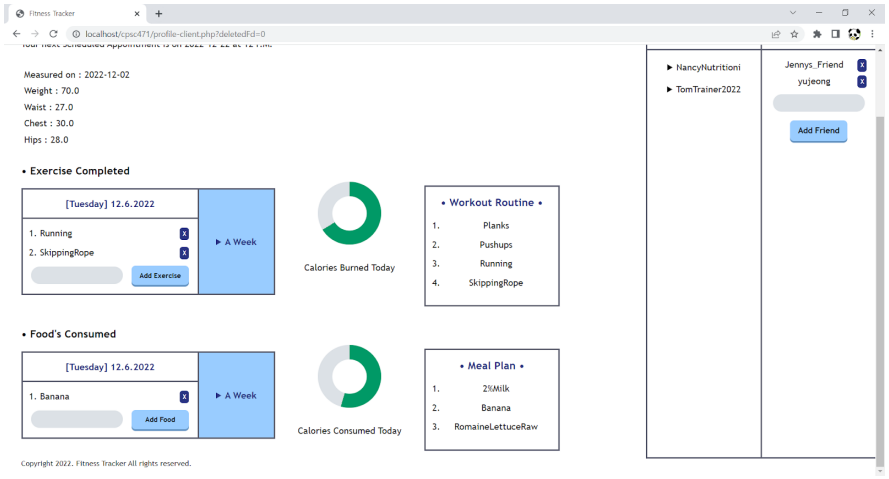


2. Each client can enter a value to the space and click the button to add that value into the list (and also the database). However, any data that does not exist on our exercise, food, and client database will give an alert.

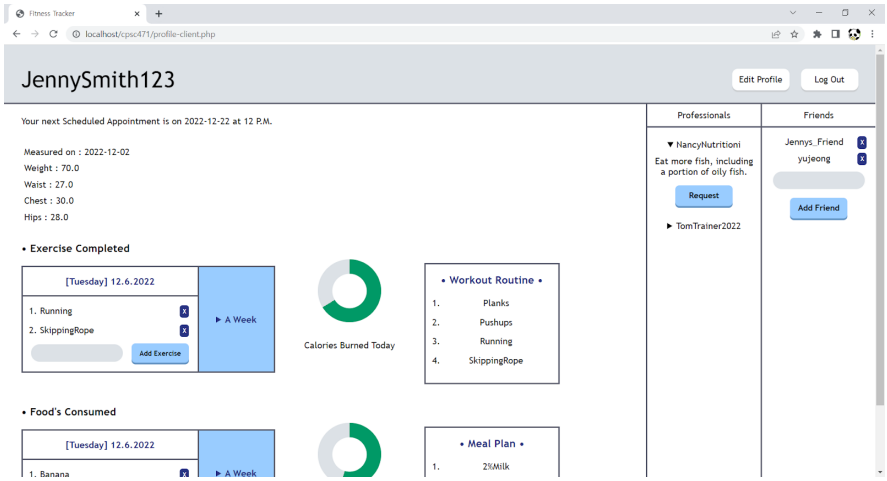




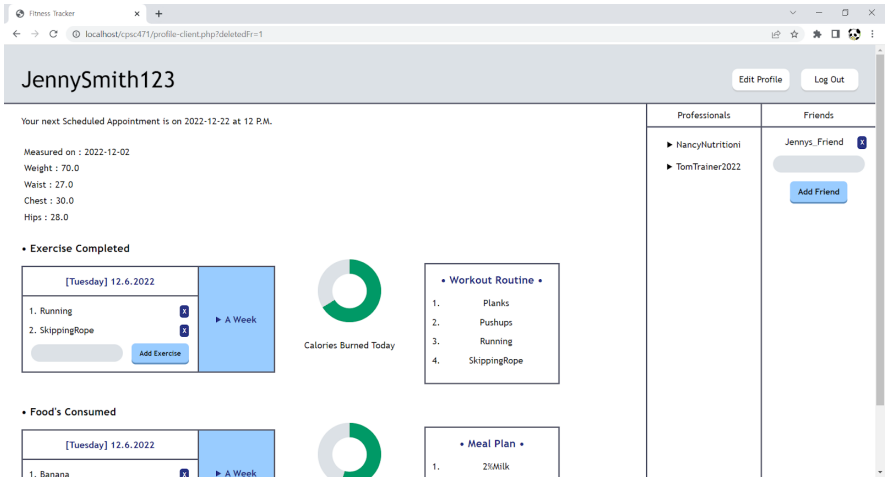
3. Same thing applies for the food that the user consumed. This time, the client removed 2%milk from the list (also from the database) and s/he added bananas.



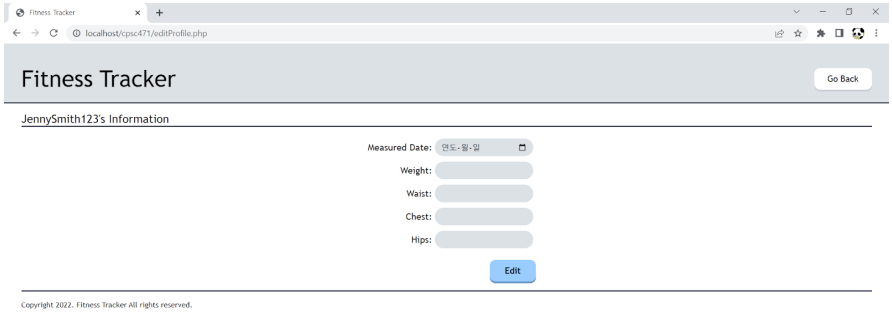
4. When the client clicks each professional's username, the feedback and request button for scheduling an appointment appear.



5. Every client can remove and add other clients for their friends.

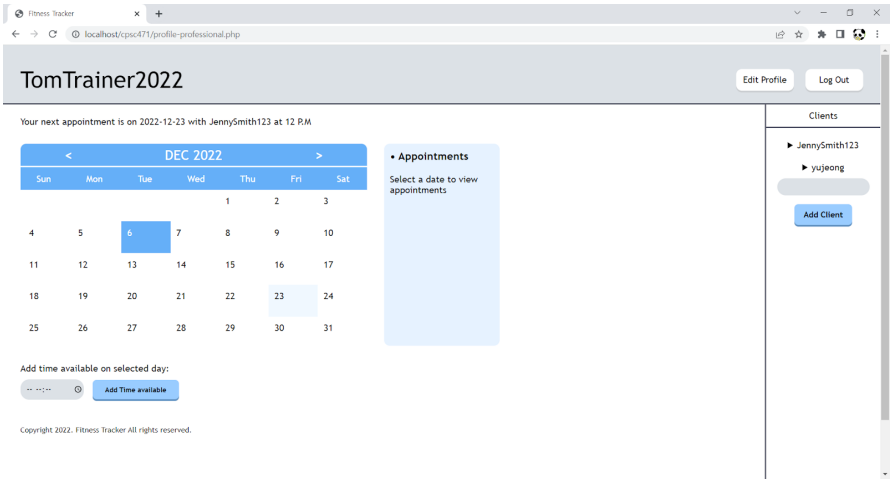


6. When they click the “Edit Profile” button, they move to a different page so that they can update their body measurements.

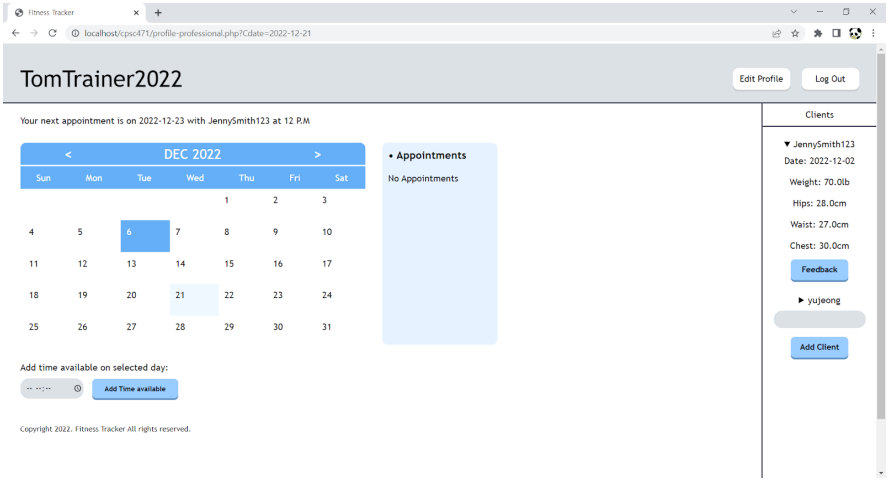


Profile page for professionals (both trainer and nutritionist):

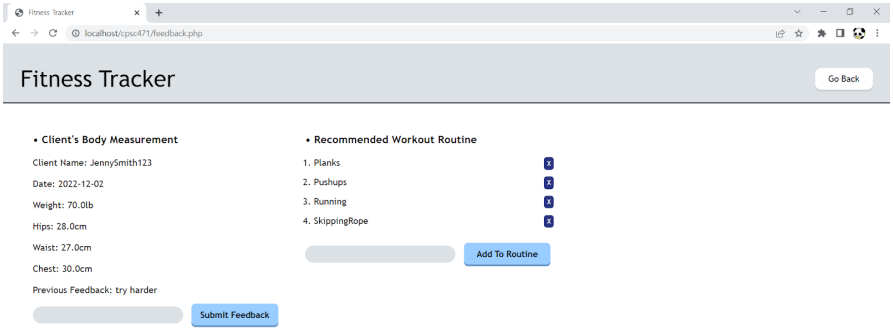
1. The upcoming appointment with a client is at the top of our webpage for professionals. The list of clients are on the right side of the page. Current date is colored blue on the calendar.



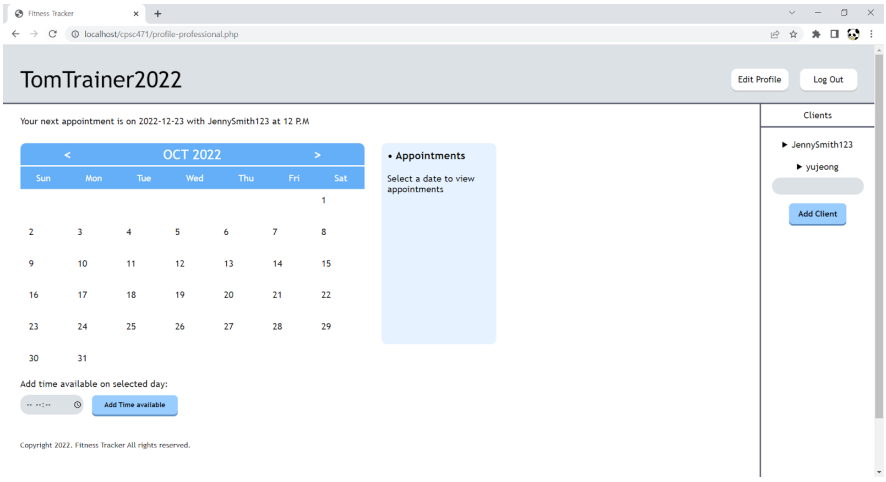
2. When the professionals click the client's name from the clients' list, his/her body measurements appear and they can move to a different webpage to send a feedback message or assign the workout routine/meal plans to each client. They are able to add clients themselves.



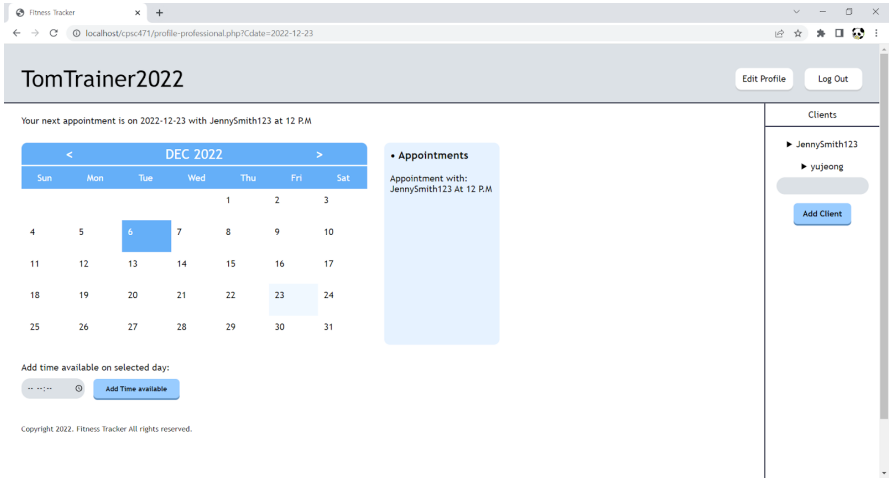
3. This is where the professionals can re-check their clients' body measurements and send their feedback and workout routine/meal plans to their clients. They can also remove elements from each list.



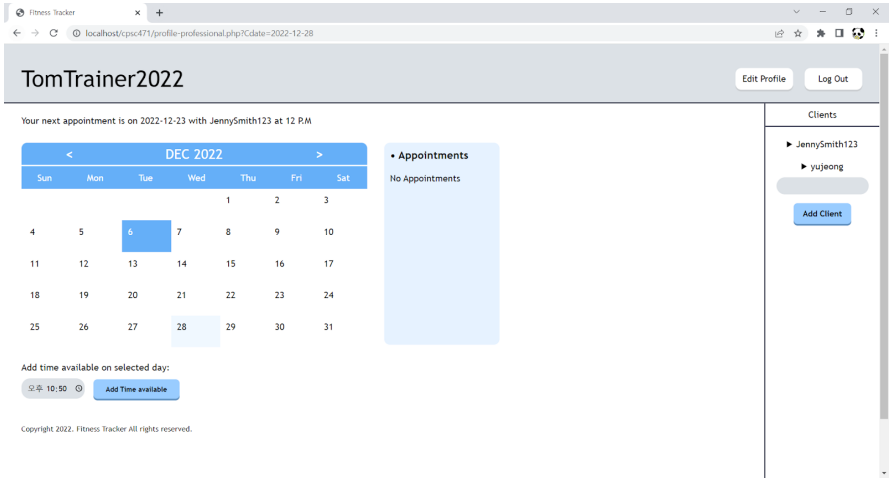
4. The professional can move to the previous or next month by clicking either “<” or “>”.



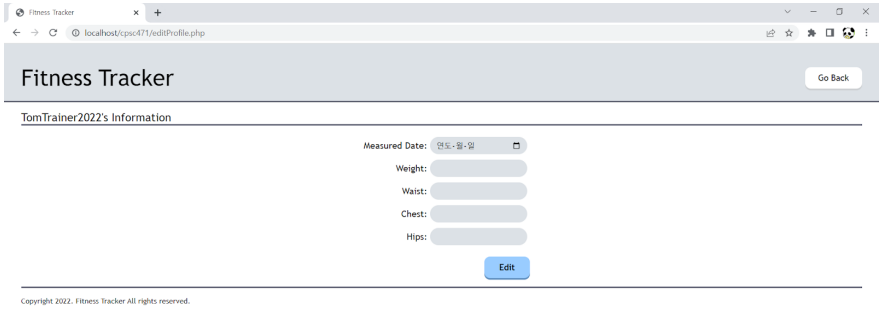
5. After selecting a day, if there are any appointments, they will appear on the blue panel labeled Appointments, and the date selected will be shown in light blue on the calendar.



6. The professionals can add an available time slot after selecting a date. The screenshot below shows “TomTrainer2022” is adding 10:50 P.M. on December 28th to his available time slot.



7. The professionals can also edit their body measurements by clicking “Edit Profile”.



Appendix (Examples)

- Users

Username	Password
JennySmith123	Password321
TomTrainer2022	Tom’sPW

- Professional

Username
TomTrainer2022

- Personal_Trainer

Username
TomTrainer2022

- Client

Username
JennySmith123

- HIRES

Client_Username	Feedback	Professional_Username

JennySmith123	Good Work	TomTrainer2022

- Food Items

	A	B	C	D	E	F
1	FoodID	Calories	Carbs	Protein	Fat	
2	2%Milk	50	4.9	3.7	1.9	All retrieved from https://fdc.nal.usda.gov/
3	Chicken	149	0	23.9	5.95	All are for 100g.
4	FujiAppleRaw	65	15.6	0.15	0.16	
5	WhiteBread	267	49.2	9.43	3.59	
6	Banana	85	20.1	0.73	0.22	
7	White rice	359	79.8	3.36	1.3	
8	RomaineLettuceRaw	21	4.06	0.98	0.07	
9	Green grapes	80	18.6	0.9	0.23	
10	Ketchup	117	26.8	1.11	0.55	
11	BreakfastSausage	341	3.37	13.3	28.7	

- List of Exercises

	A	B	C	D	E	F	G
1	Exercise_Name	Type	Calories_Burned				
2	Pushups	Weight_Training	70	All calories correspond to calories burned in 10 minutes of activity			
3	Running	Cardio	114	Retrieved from Healthline			
4	Planks	Weight_Training	39				
5	SkippingRope	Cardio	100				
6	Stretching	Other	27				
7	Squat	Weight_Training	80				
8	ChestPress	Weight_Training	67				
9	Deadlift	Weight_Training	84				
10	LegPress	Weight_Training	44				
11	Martial Arts	Other	79				

References

1. Khubchandani, J., Price, J. H., Sharma, S., Wiblishauser, M. J., & Webb, F. J. (2022). COVID-19 pandemic and weight gain in American adults: A nationwide population-based study. Diabetes & metabolic syndrome, 16(1), 102392. https://doi.org/10.1016/j.dsx.2022.102392

2. Twells LK, Janssen I, Kuk JL. Canadian Adult Obesity. Clinical Practice Guidelines: Epidemiology of Adult Obesity. Available from: https://obesitycanada.ca/guidelines/epidemiology. Accessed 09/26/2022.

3. MyFitnessPal. https://www.myfitnesspal.com/ .

4. Noom. https://www.noom.com/.