Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY CARLOW

At the heart of South Leinster

Institute of Technology Carlow

Software Development

Design Document

Gym Performance System

Data Science Year 4 Project

By

Daniel Hayden - C00137009
Supervisor - Dr. Greg Doyle

Due Date: 23/11/2017

Table of Contents

1. Introduction

The purpose of this document is to detailed breakdown and explanation of how the different core components will work together in this system. By utilizing a use case diagram and detailed use cases, you will gain a better understanding of the system and its architecture in a more detailed view than the previous documents related to this project. Also included in this document will be the latest and most accurate user interface designs and important system sequence diagrams that will all allow us to develop our classes, objects, controllers and algorithms faster and most efficiently. This document could also be used as a reference guide for a new developer to get familiar with the system and its functions.

The purpose of this project is to develop a Gym Client Performance System. The system will comprise of two user applications and a database. A hybrid mobile application for gym clients to input their progress and track their target goals and even join the local gym. This mobile app will also include a training schedule created by a personal trainer for each client. Also included will be a PHP Admin dashboard to CRUD gyms, trainers and clients on the system and also a space to allow trainers to interact with clients. Finally, A Google Firebase noSQL database with its RESTful API will be used.

For an overview of the functional specifications for the system see my previous Functional Specification Document HERE.

See the research I carried out before designing this system in my Research Document HERE.

1.1 Business Case Summary

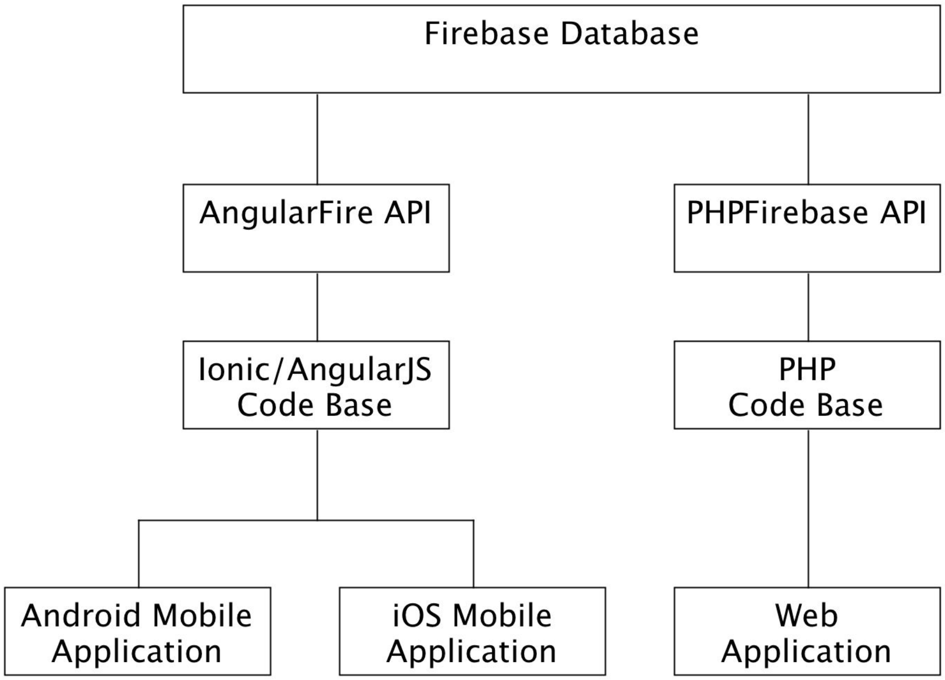
Total Fitness will make revenue from google and paid user ads in the future. As a Personal Trainer or Gym, with the Total Fitness System you will see revenue increase as you use the application to both find new clients, and provide a better service to outstanding clients. Total Fitness will provide free advertising to their direct market demographic by promoting each personal trainer on the application and giving personal trainers an active online presence. Personal trainers can promote themselves to all of their connections by posting content on their newsfeed. Essentially, Total Fitness will be a free marketing tool for Gyms and Personal trainers

1.2 Customer Segment

The system would provide your gym users with a better experience by letting them know you care for them and facilitating them with the best technologies on the market in order to help them achieve their fitness goals. The Total Fitness System aims to encourage gym clients to continue to use their gym and hit their personal fitness targets by providing your clients with a portable access hub to the gym, it's personal trainers and the client's fitness progress.

2. Architecture

In the previous Functional Specification document we gave a general overview of the systems architecture. Since then we have gained a more detailed understanding of the system and can now give a more accurate diagram of the architecture.



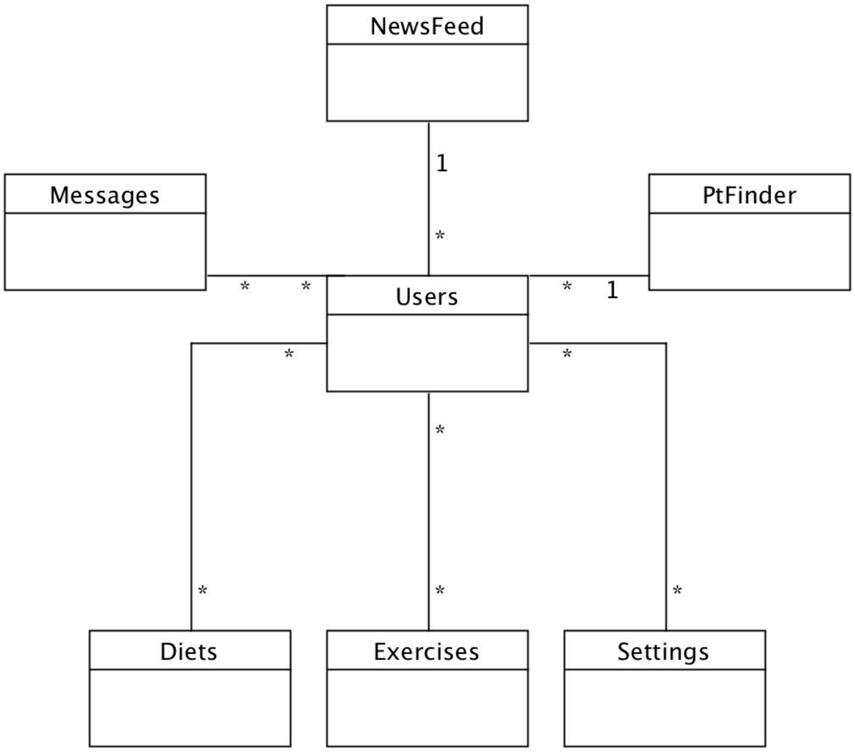
[Figure 1. Total Fitness Architecture]

As you can see in Figure 1 above, we will have three end applications, An Android Mobile Application, iOS Mobile application and Web Application. The Android and iOS Mobile applications will both be packaged from the single Ionic Framework codebase. The web application will have its own PHP based code base. Both codebases with use html, css javascript and the bootstrap framework mainly for user interface designs. Both code bases will then communicate with the Firebase Database via their API'S. Ionic has its own API or Service called AngularFire that allows an Ionic application to work with a Firebase database. There is also a firebase RESTful API we will use with the PHP Web Application.

"AngularFire is the officially supported AngularJS binding for Firebase. This binding lets you associate Firebase references with Angular models so that they will be transparently and immediately kept in sync with the database and with all other clients currently using your application."[1]

3. Database Structure

Figure 2 below gives a graphical view of the database structure. The database was designed to be as lightweight as possible. We plan to have the clients and personal trainers stored in a single table called ‘users’. There will be a boolean value to differentiate between standard users/clients and personal trainers. This is an early design of the database, final database will likely have more tables as development progress is made.



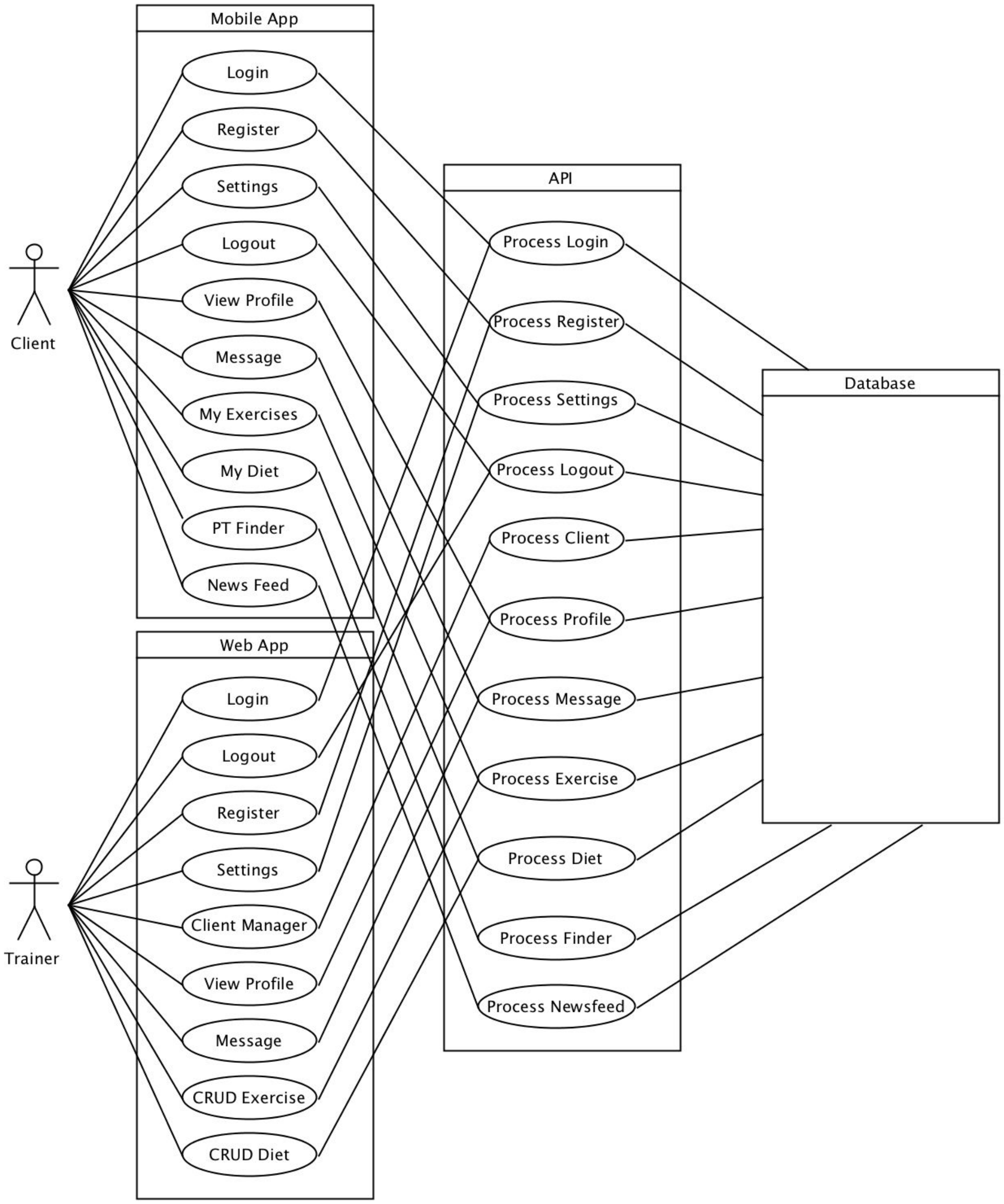
[Figure 2. Database Structure]

4. Use Case Diagram

4.1 Actors

Trainer - Gym Personal Trainer

Client - The Personal Trainers client or a user who is seeking a Personal Trainer



5. Detailed Use Cases

5.1 Mobile Application Use Cases

5.1.1 Login

Actors: Client, API, Database

Description: This use case starts when the client wants to login to the mobile application.

Preconditions: The client has registered and has the application installed on his/her mobile device.

Main Success Scenario:

1. The client is presented with a Login and a Sign Up option.
2. The client selects the Login option.
3. The client is presented with a username and password login screen.
4. The client enters his/her credentials into the username and password fields and taps on the Login button.
5. The credentials are sent to the Firebase API and validated against the database and the response is returned.
6. The user is logged in.

Alternatives:

3a. The login screen fails to load and is not presented.

4a. The email format is incorrect.

5a. The credentials are incorrect and returned false.

5b. The user is not logged in

5.1.2 Register

Actors: Client, API, Database

Description: This use case starts when the client wants to register an account on the mobile application.

Preconditions: The client has has the application installed on his/her mobile device.

Main Success Scenario:

1. The client opens the application and has not previously saved a cookie to remember his/her login session.
2. The client is presented with a Login and a Sign Up option.
3. The client selects the Sign Up option.
4. The client is presented with fields to enter such as Username, Email, Password.
5. The client enters his/her information into the corresponding registration fields and taps the Sign Up button.
6. The credentials are sent to the Firebase API and validated against the database and the response is returned.

Alternatives:

5a. The user input was incorrect and failed validation.

5.1.3 Settings

Actors: Client, API, Database

Description: This use case starts when the client wants to check or update his/her account details or settings such as password.

Preconditions: The client is logged in on the defaulting landing home screen/news feed .

Main Success Scenario:

1. The client opens the side menu and selects Settings.
2. The client is presented with a screen that contains all of the modifiable settings that the applications has including the client's own details such as email address and password.
3. The client enters his/her new password and taps the Update button at the bottom of the form.
4. The new details are sent to the Firebase API and the database is updated and the response is returned.

Alternatives:

3a. The user input was incorrect and failed validation.

5.1.4 Logout

Actors: Client, API, Database

Description: This use case starts when the client wants to logout.

Preconditions: The client is logged in.

Main Success Scenario:

1. The client opens the side menu and selects Logout.
2. The client details are sent to the Firebase API, the session is destroyed and the response is returned.

Alternatives:

2a. The client details are not recognised by Firebase and and error is returned.

5.1.5 News Feed

Actors: Client, API, Database

Description: This use case starts when the client wants to post a status to his/her news feed.

Preconditions: The client is logged in.

Main Success Scenario:

3. The client opens the side menu and selects Logout.
4. The client details are sent to the Firebase API, the session is destroyed and the response is returned.

Alternatives:

2a. The client details are not recognised by Firebase and and error is returned.

5.1.6 View Profile

Actors: Client, API, Database

Description: This use case starts when the client wants to view a profile.

Preconditions: The client is logged in.

Main Success Scenario:

1. The client opens the side menu and selects Profiles.
2. The client is presented with a search bar and a list of the top 10 viewed profiles.
3. The client can search for a user view username or name and tap the Search button.
4. The search details are sent to the Firebase API, and the users with matching search criteria are displayed.
5. The client taps on the user profile her/she wants to view.
6. The user details are sent to the Firebase API, and the user's profile details such as Name, Location, Personal Trainer, Gym are displayed.

Alternatives:

6a. The client details are not recognised by Firebase and and error is returned.

5.1.7 Messages

Actors: Client, API, Database

Description: The client wants to view a received message.

Preconditions: The client is logged in.

Main Success Scenario:

1. The client opens the side menu and taps on Messages.
2. The client is presented with a table containing rows of inbox messages from other users
3. The top message is marked unread and the client taps on it to open it.
4. The client is presented with the message and below it there is a textbox marked Reply.
5. The client can enter text into the textbox and tap on the Reply button to send a reply.
6. The message is sent to the Firebase API and the database is updated and the response is returned.

Alternatives:

6a. The client details are not recognised by Firebase and and error is returned.

5.1.8 My Exercises

Actors: Client, API, Database

Description: The client wants to view a his/her exercise plan for the day

Preconditions: The client is logged in and has an exercise plan set up.

Main Success Scenario:

1. The client opens the side menu and taps on Exercises.
2. The client is presented with a 7 day view calendar.
3. The overview of the calendar gives a workout category breakdown by muscle group for that day, e.g Biceps and Traps.
4. The client taps on today.
5. The client is then presented with a breakdown of the specific Bicep and Trap exercises that need to be performed.
6. The client taps on Bicep Curls.
7. The client is then presented with pictures and steps on how to perform correct bicep curls and the number of reps, sets and weight required by the personal trainer.
8. The client taps on Exercise Complete button at the bottom.
9. The details are sent to the Firebase API and the database is updated and the response is returned.

Alternatives:

6a. The client details are not recognised by Firebase and and error is returned.

5.1.9 My Diet

Actors: Client, API, Database

Description: The client wants to view his/her diet for the day.

Preconditions: The client is logged in and has a diet plan set up.

Main Success Scenario:

1. The client opens the side menu and taps on Diet.
2. The client is presented with a 7 day view calendar.
3. The client taps on today.
4. The client is then presented with a table with time of day on the left column and meal on the right column.
5. The client can view what he/she should be eating at any given time of any given day of the week.

Alternatives:

6a. The client details are not recognised by Firebase and and error is returned.

5.1.10 PT Finder

Actors: Trainer, API, Database

Description: The client wants to view a received message.

Preconditions: The client is logged in.

Main Success Scenario:

1. The client opens the side menu and taps on PT Finder.
2. The client is presented with a similar screen to the Profile Finder with the text search box, except this time only personal trainer in the location are displayed.
3. The client taps on a personal trainer profile that he/she wants to hire.
4. The client is presented with the personal trainer's profile page.
5. The client taps on a button called Request Session.
6. A request is sent to the personal trainer who can accept or decline.

Alternatives:

6a. The client details are not recognised by Firebase and and error is returned.

5.2 Mobile Application Use Cases

5.2.1 Login

Actors: Client, API, Database

Description: This use case starts when the trainer wants to login to the web application.

Preconditions: The trainer has registered and has loaded the web application

Main Success Scenario:

1. The trainer is presented with a login screen.
2. The trainer enters his login credentials into the login fields and clicks Login.
3. The details are sent to the Firebase API and checked against the database. A valid response is returned.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

3b. The user is not logged in.

5.2.2 Register

Actors: Client, API, Database

Description: This use case starts when the trainer wants to register for the web application.

Preconditions: The trainer has registered and has loaded the web application

Main Success Scenario:

1. The trainer visits or domain domain name for the web app and has not previously saved a cookie to remember his/her login session.
2. The trainer is presented with a Login and a Sign Up option.
3. The trainer selects the Sign Up option.
4. The trainer is presented with fields to enter such as Username, Email, Password.
5. The trainer enters his/her information into the corresponding registration fields and clicks on the Sign Up button.
6. The credentials are sent to the Firebase API and validated against the database and the response is returned.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

3b. The user is not logged in

5.2.3 Settings

Actors: Trainer, API, Database

Description: This use case starts when the trainer wants to check or update his/her account details or settings such as password.

Preconditions: The trainer has logged in and has loaded the web application

Main Success Scenario:

1. The trainer selects the Settings option.
2. The trainer is presented with a screen that contains all of the modifiable settings that the application has including the trainer's own details such as email address and password.
3. The trainer enters his/her new password and taps the Update button at the bottom of the form.
4. The new details are sent to the Firebase API and the database is updated and the response is returned.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

3b. The user is not logged in

5.2.4 Logout

Actors: Trainer, API, Database

Description: This use case starts when the trainer wants to login to the web application.

Preconditions: The trainer is logged in and has loaded the web application.

Main Success Scenario:

1. The trainer navigates the side menu and selects Logout.
2. The trainer details are sent to the Firebase API, the session is destroyed and the response is returned.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

3b. The user is not logged in

5.2.5 Client Manager

Actors: Trainer, API, Database

Description: This use case starts when the trainer wants to view a client's details.

Preconditions: The trainer is logged in and has loaded the web application.

Main Success Scenario:

1. The trainer navigates the menu and selects Client Manager.
2. The trainer is then presented with a list of all his current client.
3. The trainer clicks on the client's profile that he/she wants to view.
4. The trainer is presented with the Clients profile.
5. There is also a tab that displays pending client requests.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

3b. The user is not logged in

5.2.6 View Profile

Actors: Trainer, API, Database

Description: This use case starts when the trainer wants to view a Client or Trainer (users) profile.

Preconditions: The trainer is logged in and has loaded the web application.

Main Success Scenario:

1. The trainer opens the menu and selects Profiles.
2. The trainer is presented with a search bar and a list of the top 10 viewed profiles.
3. The trainer can search for a user view username or name and tap the Search button.
4. The search details are sent to the Firebase API, and the users with matching search criteria are displayed.
5. The trainer taps on the user profile her/she wants to view.
6. The user details are sent to the Firebase API, and the user's profile details such as Name, Location, Personal Trainer, Gym are displayed.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

3b. The user is not logged in

5.2.7 View Profile

Actors: Trainer, API, Database

Description: This use case starts when the trainer wants to view a received message.

Preconditions: The trainer is logged in and has loaded the web application.

Main Success Scenario:

1. The trainer opens the menu and clicks on Messages.
2. The trainer is presented with a table containing rows of inbox messages from other users.
3. The top message is marked unread and the trainer clicks on it to open it.
4. The trainer is presented with the message and below it there is a text box marked Reply. The trainer can enter text into the textbox and tap on the Reply button to send a reply. The message is sent to the Firebase API and the database is updated and the response is returned.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

3b. The user is not logged in

5.2.8 CRUD Exercise (Manager)

Actors: Trainer, API, Database

Description: This use case starts when the trainer wants to create a new exercise for a client.

Preconditions: The trainer is logged in and has loaded the web application.

Main Success Scenario:

1. The trainer opens the menu and clicks on Exercise Manager.
2. The trainer is presented with a list of all the current exercise plans that he/she created and has saved.
3. The trainer click on the Create Exercise Plan.
4. The trainer is presented with a exercise creation screen where he she can select from a list of exercises and enter any required details needs to perform the exercise such as weight, reps and sets.
5. The trainer then selects the days of the week to perform the exercise.
6. The trainer clicks on Save Exercise Plan.
7. The client is directed back to the Exercise manager homepage.
8. This exercise manager page also has the option to assign exercise plans to clients.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

3b. The user is not logged in

5.2.9 CRUD Diet (Manager)

Actors: Trainer, API, Database

Description: This use case starts when the trainer wants to create a new diet plan for a client.

Preconditions: The trainer is logged in and has loaded the web application.

Main Success Scenario:

1. The client opens the menu and clicks on Diet Manager.
2. The trainer is presented with his/her saved Diet plans.
3. The trainer clicks on Create Diet Plan.
4. The trainer is presented with a 7 day view calendar.
5. The client taps on Monday. The trainer is then presented with a table with time of day on the left column and meal on the right column.
6. The trainer fills in the time of day to eat the meal and also fills in the meal details.
7. The trainer click Save Diet Plan.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

3b. The user is not logged in

5.3 API Use Cases

5.3.1 Process Login

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to login and has submitted the login form.

Preconditions: The trainer or client has registered an account on the system.

Main Success Scenario:

1. The the login credentials is sent to the Firebase Auth Service and from there is checked against the database.
2. The credentials are verified and a successful response is returned to the application.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

5.3.2 Process Register

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to register and has submitted the registration form.

Preconditions: The register from has been submitted.

Main Success Scenario:

1. The the registration details is sent to the API and from there are stored in the database.
2. A successful response is returned to the application.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

5.3.3 Process Settings

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to update his/her settings and has submitted the form.

Preconditions: The trainer or client has updated their settings.

Main Success Scenario:

1. The the updated details are sent to the API and from there are stored in the database.
2. A successful response is returned to the application.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

5.3.4 Process Logout

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to logout and has clicked the logout form.

Preconditions: The trainer or client has registered an account on the system.

Main Success Scenario:

1. The the userid is sent to the Firebase Auth Service and from there is checked against the database.
2. The users session is destroyed and a successful response is returned to the application.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

5.3.5 Process Profile

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to search for a user profile and the form has been submitted.

Preconditions: The trainer or client has registered an account on the system.

Main Success Scenario:

1. The search algorithm will use the API to check the submitted search criteria against the database.
2. The matching user profiles will be returned by the API.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

5.3.6 Process News Feed

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to view his/her newsfeed and has loaded it.

Preconditions: The trainer or client has registered an account on the system.

Main Success Scenario:

1. The news feed algorithm will use the API to display status from all of the profiles/users that the client or trainer is following.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

3a. The credentials are incorrect and returned false.

5.3.7 Process Message

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to view his/her message and has loaded it.

Preconditions: The trainer or client has registered an account on the system.

Main Success Scenario:

1. The the userid is sent to the Firebase Auth Service and from there is checked against the database.
2. The users session is destroyed and a successful response is returned to the application.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

5.3.8 Process Exercise

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to add an exercise and has submitted it.

Preconditions: The trainer or client has registered an account on the system.

Main Success Scenario:

1. The exercise details will be sent to the API.
2. The API will then save the details in the database.
3. The API will return a status message to the application.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

5.3.9 Process Diet

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to add a diet and has submitted it.

Preconditions: The trainer or client has registered an account on the system.

Main Success Scenario:

1. The diet details will be sent to the API.
2. The API will then save the details in the database.
3. The API will return a status message to the application.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

5.3.10 Process Finder

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to find a personal trainer and has clicked on one's profile.

Preconditions: The trainer or client has registered an account on the system.

Main Success Scenario:

1. The API will return request the profile information for the profile selected from the database.
2. The API will return the data to the application.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

5.3.11 Process Client

Actors: Trainer, Client, API, Database

Description: This use case starts when the client or trainer wants to view his/her client list and is on the client page.

Preconditions: The trainer or client has registered an account on the system.

Main Success Scenario:

1. The API will request a list of all the clients that have the requested trainer id.
2. The list of clients will be returned.

Alternatives:

1a. The login screen fails to load and is not presented.

2a. The email format is incorrect.

6. References

[1] https://www.firebase.com/docs/web/libraries/angular/api.html