

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*
TECHNOLOGY

CARLOW

At the heart of South Leinster

Institute of Technology Carlow
Software Development

Functional Specification

Gym Performance System

Year 4 Project

By

Daniel Hayden - C00137009

Supervisor - Dr. Greg Doyle

Due Date: 13/11/2017

Table of Contents

Table of Contents	2
1. Introduction	4
2. Project Concept	4
3. Architecture Overview	5
4. Use Case Diagram	6
4.1 Actors	6
5. Brief Use Cases	7
5.1 Mobile Application Use Cases	7
5.1.1 Login	7
5.1.2 Register	7
5.1.3 Settings	8
5.1.4 Logout	8
5.1.4 View Profile	9
5.1.5 News Feed	9
5.1.6 Messages	10
5.1.7 My Exercises	10
5.1.8 My Diet	11
5.1.9 PT Finder	11
5.2 Web Application Use Cases	12
5.2.1 Login	12
5.2.2 Register	12
5.2.3 Settings	13
5.2.4 Logout	13
5.2.5 Client Manager	14
5.2.6 View Profile	14
5.2.7 Messages	15
5.2.8 CRUD Exercise (Manager)	15
5.2.9 CRUD Diet (Manager)	16
5.3 API Use Cases	17
5.3.1 Process Login	17
5.3.2 Process Register	17
5.3.3 Process Settings	17
5.3.4 Process Logout	18

5.3.5 Process Profile	18
5.3.6 Process News Feed	18
5.3.7 Process Message	19
5.3.8 Process Exercise	19
5.3.9 Process Diet	19
5.3.10 Process Finder	20
5.3.11 Process Client	20
6. Supplementary Specification	21
6.1 Functionality	21
6.2 Usability	21
6.3 Reliability	21
6.4 Performance	21
6.5 Supportability	21

1. Introduction

The purpose of this document is to describe the functional and nonfunctional requirements for the gym performance system Total Fitness to be developed in this project. The core functionality will be outlined below for all system components. This functional specification document will also provide us with different use cases for the system and give an overview of the overall architecture of the system.

2. Project Concept

The concept of this project, is to develop a gym personal training performance system. The system will consist of a cross platform mobile application, a web dashboard and a database. There are two possible ways that a gym user can download and use this system.

1. If the client does not have a personal trainer at their gym and would like to have one they can find and hire one on the app.
2. If the personal trainer asks his/her client to download the app so he/she can monitor the client's progress and to assign workout plans.

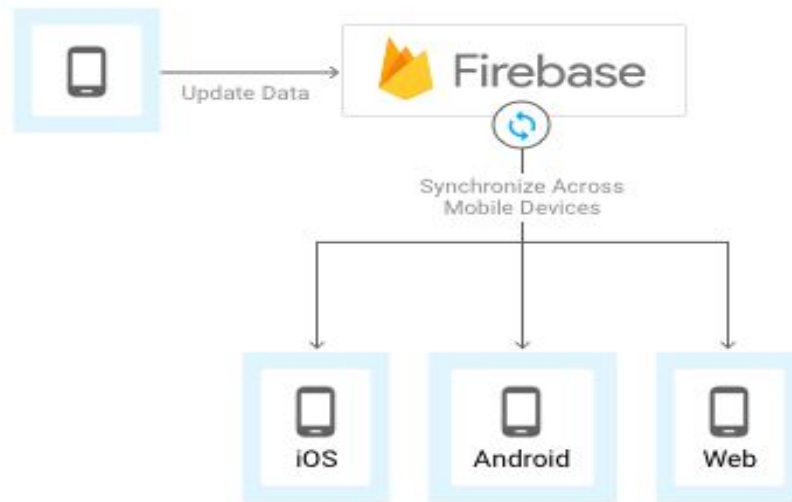
The mobile application is to be used mainly by the personal trainers' clients and people who are looking for a personal trainer. It can be used to find and request a trainer to work with you or if you already have a personal trainer, you can use the mobile application to check which muscle group and more importantly, which exercises you need to work on for any given day of the week. You can also send a private message to your trainer as some personal trainers do not like to give out their private number. Another cool feature worth mentioning is you can check what your personal trainer has planned for you to eat during the day with ease.

The web application will be similar in functionality to the mobile application however, the web application will be marketed towards the personal trainers themselves. The reason for this is it will be easier to create and manage workout and diet plans on a laptop on desktop. The web application will also allow for more functionality than the mobile application.

The Firebase system will provide a live, noSQL database and allow me to use its API to connect both code bases with ease and allow them to communicate seamlessly. The Firebase Auth API will also allow me to forget about having to manage my user login sessions and allow me to open up registration and verification via Facebook, Github, SMS, Email and much more.

3. Architecture Overview

The diagram below was created by Google Firebase and I thought it would be good to include as it gives a very clear overview of the architecture of the system.



(Figure 1. Sample Architecture)

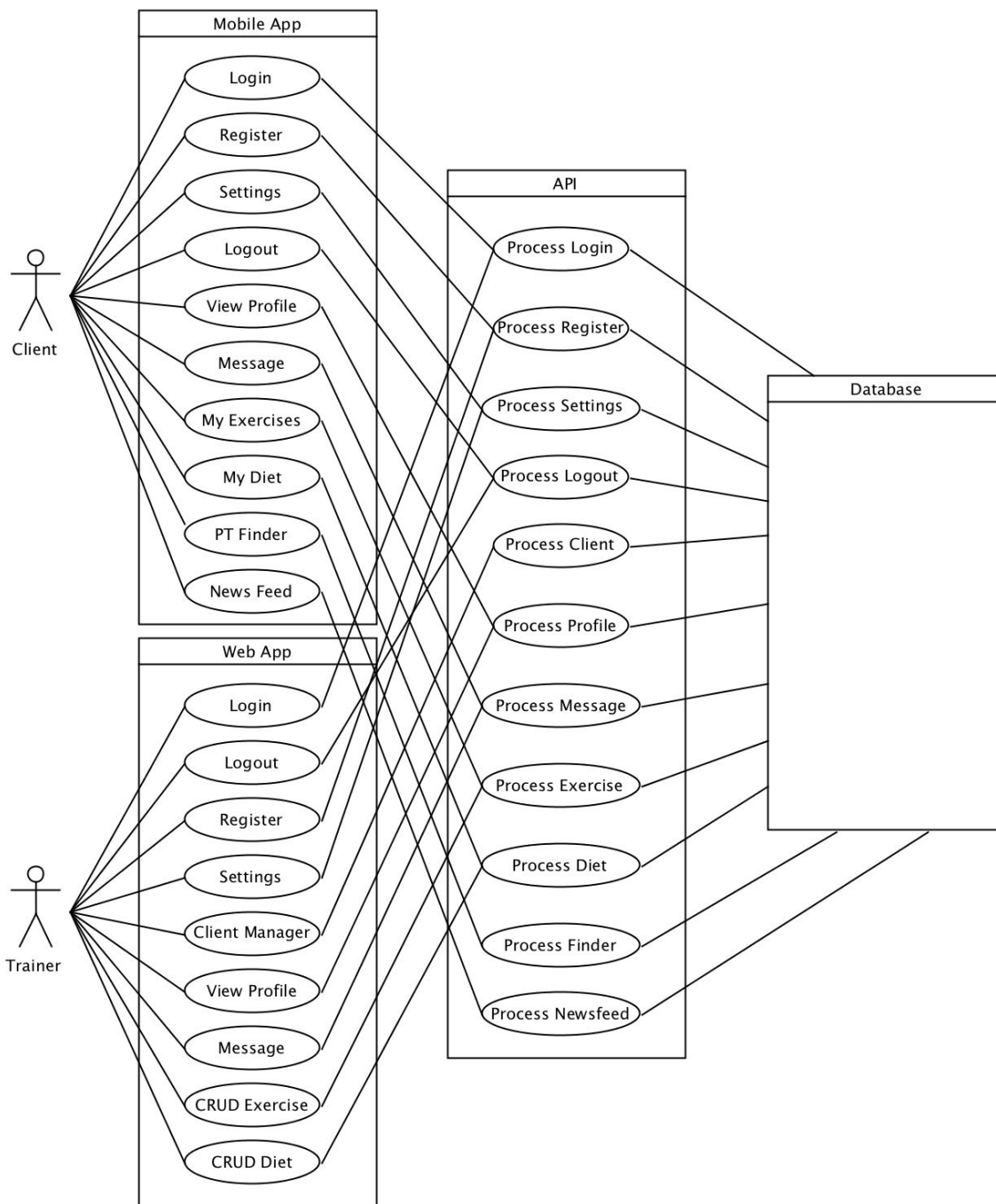
The diagram shows that when a piece of data is updated or changed, the Firebase database system synchronizes the update to each platform. This is known as a live database and it keeps all of the data up to date all of the time. It means no user will be looking at or trying to modify outdated data. Note that for my project the Android and iOS platforms will be maintained by the single ionic code base.

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. Brief Use Cases

5.1 Mobile Application Use Cases

5.1.1 Login

Name	Login
Actors	Client, API, Database
Preconditions	The client wants to login to the mobile application and has the application installed on his/her device.
Use Case	The client opens the application and has not previously saved a cookie to remember his/her login session. The client is presented with a Login and a Sign Up option. The client selects the Login option. The client is presented with a username and password login screen. The client enters his/her credentials into the username and password fields and taps on the Login button. The credentials are sent to the Firebase API and validated against the database and the response is returned.
Consequence	The client has successfully logged into the mobile application.

5.1.2 Register

Name	Register
Actors	Client, API, Database
Preconditions	The client wants to register a new account to use.
Use Case	The client opens the application and has not previously saved a cookie to remember his/her login session. The client is presented with a Login and a Sign Up option. The client selects the Sign Up option. The client is presented with fields to enter such as Username, Email, Password. The client enters his/her information into the corresponding registration fields and taps the Sign Up button. The credentials are sent to the Firebase API and validated against the database and the response is returned.
Consequence	The client has successfully registered an account.

5.1.3 Settings

Name	Settings
Actors	Client, API, Database
Preconditions	The client is logged in on the defaulting landing home screen/news feed and wants to check or update his/her account details or settings such as password.
Use Case	The client opens the side menu and selects Settings. 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. The client enters his/her new password and taps the Update button at the bottom of the form. The new details are sent to the Firebase API and the database is updated and the response is returned.
Consequence	The client has successfully update his/her account settings.

5.1.4 Logout

Name	Logout
Actors	Client, API, Database
Preconditions	The client is logged in and wants to logout.
Use Case	The client opens the side menu and selects Logout. The client details are sent to the Firebase API, the session is destroyed and the response is returned.
Consequence	The client has successfully logged out from the application.

5.1.5 View Profile

Name	View Profile
Actors	Client, API, Database
Preconditions	The client is logged in and wants to view a Client or Trainer (users) profile.
Use Case	The client opens the side menu and selects Profiles. The client is presented with a search bar and a list of the top 10 viewed profiles. The client can search for a user view username or name and tap the Search button. The search details are sent to the Firebase API, and the users with matching search criteria are displayed. The client taps on the user profile her/she wants to view. The user details are sent to the Firebase API, and the user's profile details such as Name, Location, Personal Trainer, Gym are displayed.
Consequence	The client has successfully viewed a Client or Trainer (users) profile.

5.1.6 News Feed

Name	News Feed
Actors	Client, API, Database
Preconditions	The client is logged in and has landed on the default homepage/news feed. The client wants to post a status to his/her news feed.
Use Case	The client is presented with a new feed that contains status updates/posts from any or the Profiles that he/she is following. At the top of the news feed there is a textbox. The client enters his/her status into the textbox and taps on the Post Status button. The client can also like and dislike posts in the news feed.
Consequence	The client has successfully posted a status to his/her news feed.

5.1.7 Messages

Name	Messages
Actors	Client, API, Database
Preconditions	The client is logged in and wants to view a received message.
Use Case	The client opens the side menu and taps on Messages. The client is presented with a table containing rows of inbox messages from other users. The top message is marked unread and the client taps on it to open it. The client is presented with the message and below it there is a text box marked Reply. The client 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.
Consequence	The client has successfully viewed a received message.

5.1.8 My Exercises

Name	My Exercises
Actors	Client, API, Database
Preconditions	The client is logged in and wants to view his/her exercise plan for the day.
Use Case	The client opens the side menu and taps on Exercises. The client is presented with a 7 day view calendar. The overview of the calendar gives a workout category breakdown by muscle group for that day, e.g Biceps and Traps. The client taps on today. The client is then presented with a breakdown of the specific Bicep and Trap exercises that need to be performed. The client taps on Bicep Curls. 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. The client taps on Exercise Complete button at the bottom. The details are sent to the Firebase API and the database is updated and the response is returned.
Consequence	The client has successfully viewed his/her exercise plan for the day.

5.1.9 My Diet

Name	My Diet
Actors	Client, API, Database
Preconditions	The client is logged in and wants to view his/her diet plan for the day.
Use Case	The client opens the side menu and taps on Diet. The client is presented with a 7 day view calendar. The client taps on today. The client is then presented with a table with time of day on the left column and meal on the right column. The client can view what he/she should be eating at any given time of any given day of the week.
Consequence	The client has successfully viewed his/her diet plan for the day.

5.1.10 PT Finder

Name	PT Finder
Actors	Client, API, Database
Preconditions	The client is logged in and wants to find a personal trainer.
Use Case	The client opens the side menu and taps on PT Finder.. 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. The client taps on a personal trainer profile that he/she wants to hire. The client is presented with the personal trainer's profile page. The client taps on a button called Request Session. A request is sent to the personal trainer who can accept or decline.
Consequence	The client has successfully found a personal trainer

5.2 Web Application Use Cases

5.2.1 Login

Name	Login
Actors	Trainer, API, Database
Preconditions	The trainer wants to login to the web app.
Use Case	The trainer visits or domain domain name for the web app. The trainer is presented with a login screen. The trainer enters his login credentials into the login fields and clicks Login. The details are sent to the Firebase API and checked against the database. A valid response is returned.
Consequence	The client has successfully logged into the web application.

5.2.2 Register

Name	Register
Actors	Trainer, API, Database
Preconditions	The trainer wants to register a new account to use.
Use Case	The trainer visits or domain domain name for the web app and has not previously saved a cookie to remember his/her login session. The trainer is presented with a Login and a Sign Up option. The trainer selects the Sign Up option. The trainer is presented with fields to enter such as Username, Email, Password. The trainer enters his/her information into the corresponding registration fields and clicks on the Sign Up button. The credentials are sent to the Firebase API and validated against the database and the response is returned.
Consequence	The trainer has successfully registered an account.

5.2.3 Settings

Name	Settings
Actors	Trainer, API, Database
Preconditions	The trainer is logged in and wants to check or update his/her account details or settings such as password.
Use Case	The trainer selects the Settings option. 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. The trainer enters his/her new password and taps the Update button at the bottom of the form. The new details are sent to the Firebase API and the database is updated and the response is returned.
Consequence	The trainer has successfully updated his/her account settings.

5.2.4 Logout

Name	Logout
Actors	Trainer, API, Database
Preconditions	The trainer is logged in and wants to logout.
Use Case	The trainer navigates the side menu and selects Logout. The trainer details are sent to the Firebase API, the session is destroyed and the response is returned.
Consequence	The trainer has successfully logged out from the application.

5.2.5 Client Manager

Name	Client Manager
Actors	Trainer, API, Database
Preconditions	The trainer is logged in and wants to view a client.
Use Case	The trainer navigates the menu and selects Client Manager. The trainer is then presented with a list of all his current client. The trainer clicks on the client's profile that he/she wants to view. The trainer is presented with the Clients profile. There is also a tab that displays pending client requests.
Consequence	The trainer has successfully viewed a client.

5.2.6 View Profile

Name	View Profile
Actors	Trainer, API, Database
Preconditions	The trainer is logged in and wants to view a Client or Trainer (users) profile.
Use Case	The trainer opens the menu and selects Profiles. The trainer is presented with a search bar and a list of the top 10 viewed profiles. The trainer can search for a user view username or name and tap the Search button. The search details are sent to the Firebase API, and the users with matching search criteria are displayed. The trainer taps on the user profile her/she wants to view. The user details are sent to the Firebase API, and the user's profile details such as Name, Location, Personal Trainer, Gym are displayed.
Consequence	The trainer has successfully viewed a Client or Trainer (users) profile.

5.2.7 Messages

Name	Messages
Actors	Client, API, Database
Preconditions	The trainer is logged in and wants to view a received message.
Use Case	The trainer opens the menu and clicks on Messages. The trainer is presented with a table containing rows of inbox messages from other users. The top message is marked unread and the trainer clicks on it to open it. 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.
Consequence	The trainer has successfully viewed a received message.

5.2.8 CRUD Exercise (Manager)

Name	CRUD Exercise (Manager)
Actors	Trainer, API, Database
Preconditions	The trainer is logged in and wants to create a new exercise for a client.
Use Case	The trainer opens the menu and clicks on Exercise Manager. The trainer is presented with a list of all the current exercise plans that he/she created and has saved. The trainer click on the Create Exercise Plan. 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. The trainer then selects the days of the week to perform the exercise. The trainer clicks on Save Exercise Plan. The client is directed back to the Exercise manager homepage. This exercise manager page also has the option to assign exercise plans to clients.
Consequence	The trainer has successfully viewed his/her exercise plan for the day.

5.2.9 CRUD Diet (Manager)

Name	CRUD Diet (Manager)
Actors	Trainer, API, Database
Preconditions	The trainer is logged in and wants to create a new diet plan.
Use Case	The client opens the menu and clicks on Diet Manager. The trainer is presented with his/her saved Diet plans. The trainer clicks on Create Diet Plan. The trainer is presented with a 7 day view calendar. 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. The trainer fills in the time of day to eat the meal and also fills in the meal details. The trainer click Save Diet Plan.
Consequence	The client has successfully viewed his/her diet plan for the day.

5.3 API Use Cases

5.3.1 Process Login

Name	Process Login
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to login and has submitted the login form.
Use Case	The the login credentials is sent to the Firebase Auth Service and from there is checked against the database. The credentials are verified and a successful response is returned to the application.
Consequence	The client or trainer has successfully logged into the application.

5.3.2 Process Register

Name	Process Register
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to register and has submitted the registration form.
Use Case	The the registration details is sent to the API and from there are stored in the database. A successful response is returned to the application.
Consequence	The client or trainer has successfully registered.

5.3.3 Process Settings

Name	Process Settings
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to update his/her settings and has submitted the form.
Use Case	The the updated details are sent to the API and from there are stored in the database. A successful response is returned to the application.
Consequence	The client or trainer has successfully updates his/her settings.

5.3.4 Process Logout

Name	Process Logout
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to logout and has clicked the logout form.
Use Case	The the userid is sent to the Firebase Auth Service and from there is checked against the database. The users session is destroyed and a successful response is returned to the application.
Consequence	The client or trainer has successfully logout.

5.3.5 Process Profile

Name	Process Profile
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to search for a user profile and the form has been submitted.
Use Case	The search algorithm will use the API to check the submitted search criteria against the database. The matching user profiles will be returned by the API.
Consequence	The client or trainer has searched for a user profile.

5.3.6 Process News Feed

Name	Process News Feed
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to view his/her newsfeed and has loaded it.
Use Case	The news feed algorithm will use the API to display status from all of the profiles/users that the client or trainer is following.
Consequence	The client or trainer has successfully viewed the news feed.

5.3.7 Process Message

Name	Process Message
Actrs	Trainer, Client, API, Database
Preconditions	The client or trainer wants to view his/her message and has loaded it.
Use Case	The message algorithm will use the API to retrieve and display the opened message.
Consequence	The client or trainer has successfully viewed a message

5.3.8 Process Exercise

Name	Process Exercise
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to add an exercise and has submitted it.
Use Case	The exercise details will be sent to the API. The API will then save the details in the database. The API will return a status message to the application.
Consequence	The client or trainer has successfully added an exercise.

5.3.9 Process Diet

Name	Process Diet
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to add a diet and has submitted it.
Use Case	The diet details will be sent to the API. The API will then save the details in the database. The API will return a status message to the application.
Consequence	The client or trainer has successfully added a diet.

5.3.10 Process Finder

Name	Process Finder
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to find a personal trainer and has clicked on one's profile.
Use Case	The API will return request the profile information for the profile selected from the database. The API will return the data to the application.
Consequence	The client or trainer has successfully found a personal trainer.

5.3.11 Process Client

Name	Process Client
Actors	Trainer, Client, API, Database
Preconditions	The client or trainer wants to view his/her client list and is on the client page.
Use Case	The API will request a list of all the clients that have the requested trainer id. The list of clients will be returned.
Consequence	The client or trainer has successfully viewed the client list.

6. Supplementary Specification

6.1 Functionality

1. We need to make sure that all runtime and user generated errors are logged in a log file for both code bases, as well success messages.

6.2 Usability

1. The Android application will be usable on the latest version on Android Nougat and the latest version of Apple's iOS.
2. The Web application will be developed for the latest version of Firefox v56 and above.
3. A new personal trainer who has never used the system, must be able to create a new training plan and assign it to a client in under 8 minutes.

6.3 Reliability

1. we can allow for a total of 5 'connection failed' errors in the error log per day.

6.4 Performance

1. The average response time for the API to process a request and offer a response back to the web application will be no longer than 30 seconds.
2. The maximum accepted response time is 60 seconds or the software has failed.

6.5 Supportability

1. We will need an offsite database backup made daily.