

Functional Manual

Powerlifting app using Movesense wearable technology

**Student Name: Samuel Orekoya**

**Course Name: BSc (Hons) in Software Development**

**Student ID: C00215885**

**Supervisor: Joseph Kehoe**

**Submission Date: 15/11/2019**

# Introduction

The aim of this project is to create an application to help athletes or users that wish to improve their strength training by analysing their main lifts and displaying valuable information like speed and power. This document is to show, elaborate and specify the functionality of the application that is being developed “Power Athlon”. Also, within this document defining its scope and vision for this project will be discussed. The document will give a written account of all the functionality that will be implemented in the creation of this power lifting application using movesense technology. As this product is made to improve strength for a number of sports athletes lots of the functionality that will be implement will be common functionality used in most fitness application like displaying important information to a user.

This functional document is broken down into multiple sections Target users, Use cases, Furps+, metrics, testing. While this document will provide a good overview of the development of this project. This does not mean that all the functionality are fixed and cannot be changed as the dynamic nature of software changes will be done throughout the development part of the project. While developing this application the goal for the project will be to evolve and continue to be evolving which will in turn create an application that customers will want.

Contents

[1. Introduction 1](#_Toc24743769)

[2. Target users 3](#_Toc24743770)

[3. Main use cases 4](#_Toc24743771)

[4. Brief Use Cases 5](#_Toc24743772)

[4.1 Register user 5](#_Toc24743773)

[4.2 Input / edit height and weight 6](#_Toc24743774)

[4.3 Display calories burnt, hr, reps, power and speed output 7](#_Toc24743775)

[4.4 Login 8](#_Toc24743776)

[4.5 Logout 9](#_Toc24743777)

[4.6 CRUD Account 10](#_Toc24743778)

[4.6.1 Update Account 10](#_Toc24743779)

[4.6.2 Delete Account 11](#_Toc24743780)

[4.7 Start workout 12](#_Toc24743781)

[5. FURPS+ 13](#_Toc24743782)

[5.1 Functionality 13](#_Toc24743783)

[5.2 Usability 13](#_Toc24743784)

[5.3 Reliability 14](#_Toc24743785)

[5.4 Performance 14](#_Toc24743786)

[5.5 Supportability 14](#_Toc24743787)

[5.6 + 14](#_Toc24743788)

[6. Metrics 15](#_Toc24743789)

[7. Testing 16](#_Toc24743790)

[8. References 17](#_Toc24743791)

# Target users

Power Athlon targeted users are anyone with any level of strength training from novice to experienced lifters who wish to increase strength in the 3 main compound lifts (squat, bench and deadlift) primality used in the powerlifting meets. The use of movesense sensors will enable user to be able to retrieve valuable information while performing lifts with great accuracy. Then in real time calculating and displaying information of the lift just performed by the user to help in the coaching of the users.

This application will also be very beneficial for user that may not be power lifters and just want to improve their strength on certain lifts like the incline bench press. As novice user and experienced user will be able to use this application it will have to have a very easy to use interface giving the user all the information he needs at a click of a button.

# Main use cases

Here is the main use case of the application Power Athlon which displays the user’s interactions.

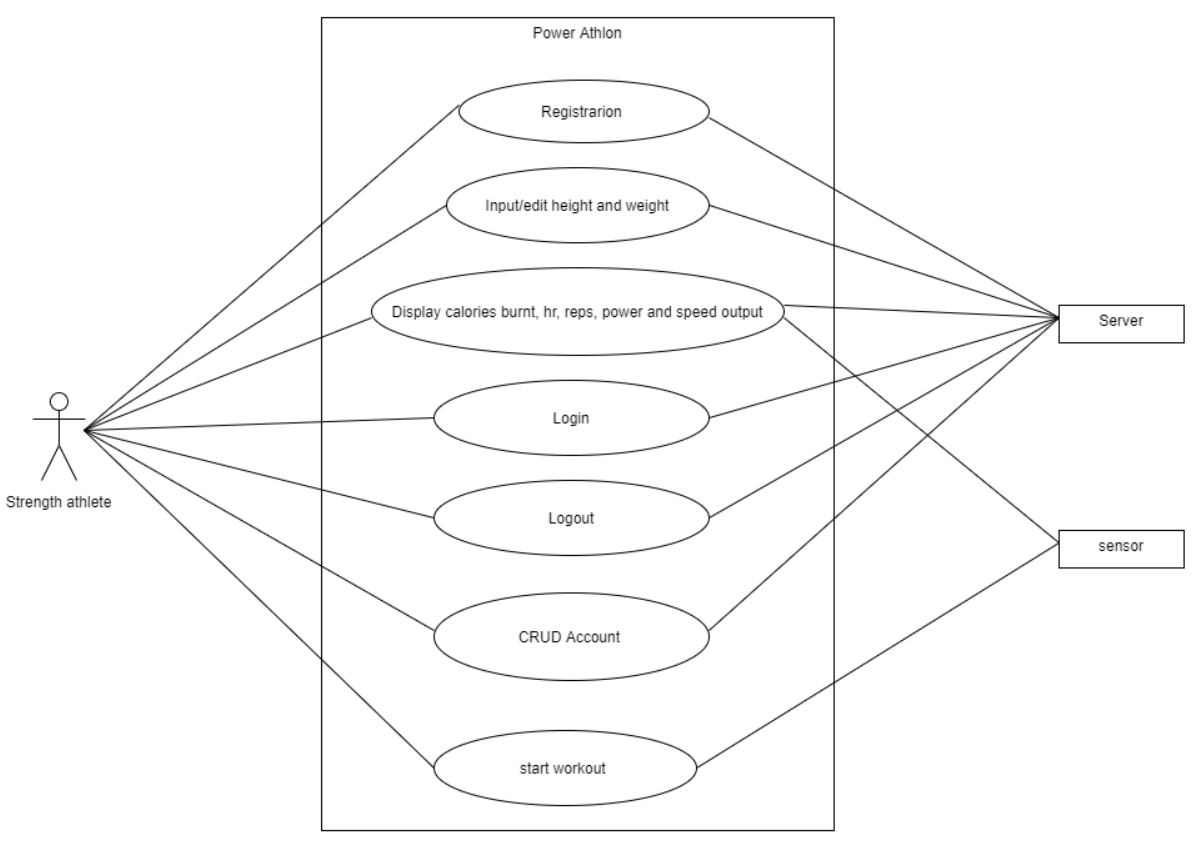


Fig 1. Main use case

# Brief Use Cases

## Register user

**Actor:** Strength athlete

**Brief Description:** This use case start when the given actor wishes to register for an account. The actor will completely fill out all the given requirement and submit the information. This use case will end as soon as the actor successfully registers with the system.

**Main Success Scenario:**

1. The Register option is selected by the actor.
2. The system displays the registration form for the actor.
3. The actor enters in his / her details into the fields provide on the form.
4. When the form is completed the form is then submitted.
5. The system does a check to see if all required fields have been filled in.
6. The system then display a “User successfully register” message to the actor and redirects him to the login page.

**Alternatives:**

5a. Not all of the required fields were entered by the actor.

1. Fields that were not entered are highlighted by the system and then prompts the actor to enter them and submit again.
2. Step 3 is repeated again.

5b. Username entered already exists in the system.

1. A message with “Username already exists” will be display by the system and will prompt the actor to enter a new username.
2. Step 3 is repeated again.

5c. The password entered does not contain an uppercase and lowercase letter, a number or the two passwords entered do not match.

1. The appropriate error message will be displayed and then prompts the actor to enter a new password again
2. Step 3 is repeated again.

## Input / edit height and weight

**Actor:** Strength athlete

**Brief Description:** This use case start when the given actor wishes to input or edit their height and weight. When the actor selects “Input / edit height and weight” from the application menu the actor will then be redirected to that page there will be a form with fields where the actor can input or edit their height and weight. This use case will end when the actor has updated or entered their physical details.

**Main Success Scenario:**

1. The “Input / edit height and weight” option is selected by the actor from the menu.
2. The system displays the Input or edit height and weight page where the actor can input or edit their details.
3. The system will then display a form where users can input or edit their physical details.
4. Inside the fields the actor will either have their previous detailed enter and will be editable or field that have not been entered by the actor with the option to input height and weight.
5. The actor will confirm their input or edit by submitting the form to the system.
6. The system will then redirect the actor to the menu page.

**Alternatives:**

4a. The actor leaves a required field blank.

1. The system will display “error field cannot be left blank”.
2. Repeat step 3 again.

## Display calories burnt, hr, reps, power and speed output

**Actor:** Strength athlete

**Brief Description:** This use case start when the given actor wishes to view their most recent workout information. The actor will just select the “display last workout” section on the menu page. This use case will end as soon as the actor successfully reviews their workout information then returns to the menu page of system.

**Main Success Scenario:**

1. The “display last workout” option is selected by the actor from the menu.
2. The system will then display the information from the last workout (calories burnt, hr, reps, power and speed output)
3. When the actor has completed reviewing their last workout there will be a “return to menu” button at the end of the page.
4. The actor will confirm by clicking the button.
5. A message from the system will display “Are you sure they want to leave this page” if actor clicks Yes.
6. The system will then redirect the actor to the menu page.

**Alternatives:**

5a. The actor doesn’t want to leave display last workout page.

1. The actor selects No instead of yes when asked if they want to leave.
2. The system will remove the alert message and actor can continue viewing page.

## Login

**Actors:** Strength athlete

**Brief Description:** This use case start when the given actor wishes to Login to their account. The actor will enter their username and password into the fields provided on the login page. completely fill out all the given requirement and submit the information. This use case will end as soon as the actor successfully logs into their account.

**Main Success Scenario:**

1. The actor will select the Login option.
2. The system displays a login page with a login form.
3. The actor enters in their login details username and password into the fields provide on the form.
4. The system then checks the details (username and password) against the database.
5. The system will redirect the actor to the menu pages.

**Alternatives:**

4a. The username entered by the actor was not found in the database.

1. A message with “Username or Password does not exist” will be display by the system and will prompt the actor to enter login details again.
2. Step 3 is repeated again.

4b. The actor has entered their details incorrect 5 times.

1. A message with “user has been locked for 10 mins”.
2. Actor will wait for 10 mins before retrying again.

## Logout

**Actor:** Strength athlete

**Brief Description:** This use case start when the given actor wishes to Logout of their account. The actor selects logout from the menu. A message will display on the screen asking if they are sure they want to log out. The actor selects yes then the use case ends when the actor is successfully logged out.

**Main Success Scenario:**

1. The logout option is selected by the actor from the menu
2. A message from the system asking the actor if they are sure they want to log out.
3. The actor then agree with the message by pressing yes.
4. The session is killed by the system and then logs them out which then redirects them to the login page.

**Alternatives:**

3a. The actor doesn’t want to actually logout.

1. The actor selects No instead of yes when asked if they want to log out.

## CRUD Account

## Update Account

**Actors:** Strength athlete

**Brief Description:** This use case begins when the actor wants to update their account details. When the actor selects “Update Account” from the application menu the actor will then be redirected to that page where their will be a form similar to the registration form where their information can be updated. This use case will end when the actor has updated their account details.

**Main Success Scenario:**

1. The “Update Account” option is selected by the actor from the menu.
2. The system displays the update account page where the actor can update their details.
3. The system will then display a form similar to the registration form enter on registration.
4. Inside the fields the actor will have their previous detailed enter and will be able to edit then.
5. The actor will confirm their changes by resubmitting the form.
6. The system will then redirect the actor to the menu page.

**Alternatives:**

5a. The actor leaves a required field blank.

1. The system will display “error field cannot be left blank”.
2. Step 4 is repeated again.

## Delete Account

**Actors:** Strength athlete

**Brief Description:** This use case begins when the actor wants to delete their account. When the actor selects “Delete Account” from the application menu the actor will then be redirected to that page. Then the actor will be requested to input their password before receiving a message asking if they are sure they want to delete their account. This use case will end when the actor has deleted their account.

**Main Success Scenario:**

1. The “Delete Account” option is selected by the actor from the menu.
2. The system displays a field for where an actor has to input their password to authenticate.
3. After the password has been authenticated a message from the system will display “Are you sure they want to delete this account”.
4. The actor then agree with the message by pressing yes.
5. The session is killed by the system and then logs them out which then redirects them to the login page.

**Alternatives:**

2a. The actor has entered their password incorrect 5 times.

1. A message with “user has been logged out and locked for 10 mins”.
2. The session is killed by the system and then logs them out which then redirects them to the login page.
3. Actor will wait for 10 mins before retrying again.

4a. The actor doesn’t want to delete their account.

1. The actor selects No instead of yes when asked if they want to delete this account.
2. The system will redirect the actor to menu page.

.

## Start workout

**Actors:** Strength athlete

**Brief Description:** This use case begins when the actor wants to start a workout on their account. When the actor selects “Start workout” from the application menu the actor will then be redirected to that page. Then the actor will then be given 3 option on what lift they wish to complete then under the 3 option will be a large button where when they are ready can click on it and start their workout. This use case will end when the actor has clicked on the end workout button.

**Main Success Scenario:**

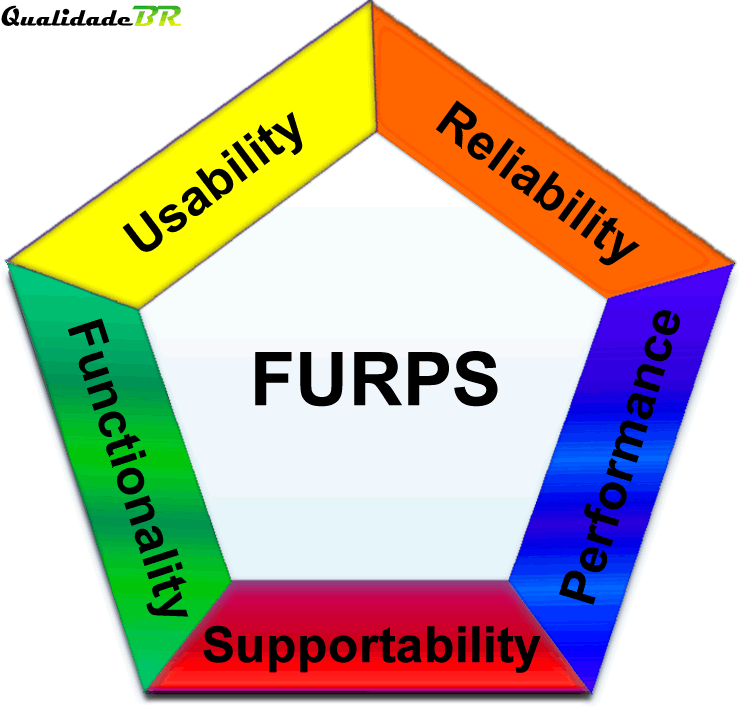
1. The “Start workout” option is selected by the actor from the menu.
2. The system will then display the 3-workout option and the large button in the middle of the page that saying, “Start workout”.
3. When the lift has been selected and the “start button” has been pressed the actor will then start their workout
4. Once the workout has been started the info from the sensor attached to the actor should then start to display on the screen.
5. When the actor workout has been completed the actor will been press on the button that say “workout completed” to end the workout.
6. Once the workout has been ended the system will then redirect the actor to the menu page.

**Alternatives:**

3a. The actor fails to select one of the 3-workout option before pressing start button.

1. A message from the system saying “please select a lift before workout can begin” will display on the screen.
2. Step 2 is repeated again.

# FURPS+

In development of an application functional and non-functional requirement need to be established. Functional requirement are essential requirements like adding an item to a cart on a shopping app which is describing how that part of the system will work. While non-functional requirement are how the application developed should behave when created. Furps+ is used as an acronym for non-functional requirements. Furps+ is also used to validate the prioritised requirements given from a customer or client.

* Functionality
* Usability
* Reliability
* Performance
* Supportability
* +

Fig 1. [1] Furps

## Functionality

Functionality in Furps+ represent the main product features of the application that is being developed. In the case of the wearable strength app functionalities like displaying data on how fast and how powerful the lift was is important. This application most allow information to be retrieve from the movesense wearable important information like heart rate, amount of reps completed and speed of the lift. Then with this information in real time should display valuable information to a given users which they can in turn use to improve their strength training [1].

## Usability

Usability will usually involve requirements based on the user interface like accessibility and interface aesthetics. Within this application it will be cross platform so it will be able to be used on Android and IOS. The version that will be supported will be the latest version on both devices [1].

When giving usability the ease of use is extremely important and needs to be measurable to give developers a goal to aim for when developing. Like for power Athlon after a set of reps the speed and power of lifts should be calculated and displayed on the given device in under 40 secs 85% of the time. When a user is logged in the application can never log out the user unless the user clicks on the log out button. When developing this app all types of users will be catered for from novice users to experienced users. This requirement is important as it is quantifiable and can be tested in a later state in development [1].

## Reliability

In reliability accuracy, recoverability and availability are very important in app development. As software cannot be 100% reliable the goal is to get to the highest level of reliability possible. Reliability on both device’s Android and IOS should be suitable for users by the app power Athlon should only crash every 6 months and not every 6 days. Accuracy is extremely in applications where real time data is collected calculated and displayed in a user-friendly format. For example, if a user brought a Fitbit to count their calories while running and training in the gym and it was providing inaccurate/ wrong information it would be a very useless app and would be a waste of money for the user. While being 100% accurate 100% of the times would be extremely hard but giving as accurate as possible information it will give the user great user experience. As the technology being used is movesenses sensor technology accuracy in information like heart rate and speed of lift should be very accurate as the hardware is created for retrieving that type of info [1].

## Performance

Performance within this will involve things like system response time which also relate to usability of the app, throughput recovery time and start up time. The same way usability requirements need to be quantified also does performance. With this app response time will be important and will be one of the goals that need to be reached to provide the best user experience having a response time of 35 sec on opening of the application is essential. Throughput of the app will not be to large but will be sufficient enough to provide a suitable number of users a more than satisfactory performance of the application. Most of the performance factor will be looked after by the cloud provider of choice [1].

## Supportability

Supportability is when software is designed in a certain way to make it easy to be customized, enhanced and repaired. This app will be cross platform getting it access to both IOS and Android users. A customizing example would be given a user the ability to input more sets to their workout. Enhancement to the software will all come down to the scalability of the software developed. The more scalable the software is the more ability of enhancement will be done in the coming years [1].

## +

The + in Furps+ is to cover all the other sections of non-functional requirements. From the other requirement there is another aspect that has not been discussed and is very important when it comes to application development which is security requirements. A common goal for developers are to create application that are protected and secured. In this application security will not be that important as there will be not sensitive info being saved by a user. But having security on the software is still very necessary when creating app where user’s data will be saved on [1].

# Metrics

* The application being created should run smoothly on all platforms.
* The user information should be accurate (Heart rate, reps completed, speed etc.)
* The application should be able to retrieve data from the Movesense sensor.
* The information on the users training should be displayed in real time.
* The user should be able to CRUD account details at any time they wish.
* All the non-functional requirement goals should be achived.

# Testing

As testing is a key part for development, there will be a need to do functional testing to see if all the use cases are working the way they should be. When developing tests that need to be done the use cases above will extremely help in the decision making of the test to see if a test is successful or not. If a test does not follow the “Main Success Scenario” it should then be classified as an unsuccessful test and then from that information the bugs should be found in system and fix it as soon as possible. While theses are the functional requirement that need to be meet so do the non-functional requirement. The Furps+ acronym is used for non-functional requirements. If the Furps+ requirements of Functionality, Usability, Reliability, performance, supportability and + (security) are achieved this will in turn display that the system from a non-functional standpoint is working well. As this project will need some sort of hardware to be a success the hardware will also need to be tested. The way the hardware will be tested is by using test subjects and compare readings like heart rate with other sensors or wearable technology with sensors like Fitbit or the Apple watch. If all the readings are the same or very similar this will mean the hardware is working as it is meant to be. While having test subject for the sensor hardware testing is useful the same tester can also test the application. As a developer testing the application myself would be useless as I know the system inside out and would give bias results as this would be my own work. A large range of users to test the system would beneficial. Users from experienced to novices in weight training and from being a technological app user to non-technological user this will share a wide range of users to give results. All the test subject will have to complete a survey with questions like how the experience was to use the app and how useful was it for their strength training. Each user will circle the answer they agree with the most this will help and reduce the change of human error when calculating the results.

# References & Fig

1. in (2019). 3.2 Tutorial on Requirements in the UP. [online] Google Docs. Available at: https://docs.google.com/document/d/1IPDyDzqnCR2VQf1liPh-j\_5eUEjQmz9ABnFNAYPZ4sI/edit [Accessed 15 Nov. 2019].

**Fig**

1. <https://www.pinterest.de/pin/529665606157139439/>