Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Visualisation of statistics in resistance training

Dávid Michal Bulko

Supervisor: RNDr. Ondřej Žára
Field of study: Software engineering and technologies
May 2022

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Bulko David**	Personal ID number:	**474582**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Computer Science**		
Study program:	**Open Informatics**		
Specialisation:	**Software Engineering**		

II. Master’s thesis details

Master’s thesis title in English:

Extending data stats in the "FitWolfe" application

Master’s thesis title in Czech:

Rozvoj statistik v aplikaci FitWolfe

Guidelines:

Bibliography / sources:

https://muscleandstrengthpyramids.com
https://developer.mozilla.org/en-US/
https://d3js.org/
https://www.oreilly.com/library/view/visualizing-data/9780596514556/

Name and workplace of master’s thesis supervisor:

RNDr. Ondřej Žára Department of Computer Graphics and Interaction

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **08.02.2022** Deadline for master's thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

RNDr. Ondřej Žára
Supervisor’s signature

Head of department’s signature

prof. Mgr. Petr Páta, Ph.D.
Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.
Date of assignment receipt

Student’s signature

Acknowledgements

I would like to thank primarily my supervisor Ondřej Žára for guidance. Other thanks go to Ondřej Mareš for suggestions on how to improve user interface and Alex Chmel for code reviews in order to keep the code quality of FitWolfe in check. Honorable mentions go to Marek Mach, Angela Zátopek, Matej Drobný and Petr Švenda on the reviews of the thesis paper. I must not leave out coaches that found time in their busy schedules when I researched the needs for the software and for their valuable suggestions on the outputs of this thesis.

Declaration

I hereby declare that this thesis represents my own work which has been done after registration for the Master's degree at Czech Technical University, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications.

Prague, 20 of May 202.

Abstract

The purpose of this master thesis was to thoroughly specify the process of creation and planning of training programs by personal fitness trainers. In order to help clients achieve their goals, the trainers have to consider many things that can influence the progress of their clients. Each client can respond differently to various stimuli and this might change over the course of the client's training career. Additionally, various sources of stress can inhibit their progress. How could this be simplified for the personal trainers? They can easily compare the changes over microcycles, however, the bigger picture may be missed. On the other hand visualising data in other forms such as graphs might resolve this issue. The web application FitWolfe, which already provides ability to create training programs could be extended to allow better tracking of the progress of the clients.

Keywords: web application, resistance training, fitness, software, development, React, JavaScript, Spring, Java

Supervisor: RNDr. Ondřej Žára

Abstrakt

Cieľom tejto diplomovej práce bolo dôkladne špecifickovať proces vytvárania a plánovania tréningových programov fitness trénermi. Na to aby pomohli svojim klientom dosiahnuť ich ciele musia tréneri zvážiť mnoho vecí, ktoré môžu vplývať na pokrok klientov. Každý klient môže reagovať odlišne na rôzne stimuly, čo sa naviac môže meniť v priebehu času. Zároveň existuje mnoho zdrojov stresu, ktoré dokážu inhibovať ich pokrok. Ako by to mohlo byť pre osobných trénerov zjednodušené? Tréneri vedia jednoducho identifikovať zmeny pri porovnaní mikrocyklov, avšak mohli by v tom minúť väčší obraz. Na druhej strane, vizualizácia v iných formách ako sú napríklad grafy, by mohla tento problém vyriešiť. Webová aplikácia Fitwolfe, ktorá už v tejto dobe umožňuje vytvárať tréningové programy, by mohla byť rozšírená funkčnosťami, ktoré by umožňovali lepšie sledovanie pokroku u klientov.

Kľúčové slová: webová aplikácia, rezistentný tréning, fitness, software, vývoj, React, JavaScript, Spring, Java

Contents

Figures

Chapter 1

Introduction

For my bachelor's thesis I have created a web application FitWolfe [1] (previously MyOnlineTrainer) that allows personal trainers to create training programs and schedule them for their clients. The clients can start a training session on FitWolfe mobile and record what they performed. Personal trainers can then view performed training sessions in the web application, assess the progress of the client and if necessary, adjust the program so that steady improvements are made, and recovery is not sacrificed.

During the implementation I have been communicating with other personal trainers in order to gather their feedback. One of the most requested features were training statistics [2]. Even though FitWolfe strives to be as intuitive as possible and also display the information very simply, it still might not be easy to see the full picture in pure numbers.

Adding graphs displaying certain metrics regarding a training program might help the personal trainer understand better how the client is progressing, give more hints about what the issue might be when the client is plateauing or how to design a new training program for a client according to the previous data.

1.1 Programming and periodization in resistance training

Resistance training can be defined as increasing muscle strength, size (hypertrophy), and endurance through contraction of muscles against external resistance such as body weight, machinery, load or similar.

People seeking out help from personal trainers have goals that are most of the time in line with three aforementioned results of resistance training - increasing strength, size or endurance. Therefore, it is very often scheduled for them. Human bodies are very adaptive which means that in order to progress, it is necessary to increase the stimulus on the muscles over time.

Resistance training has been present in the lives of humans at least since Ancient Greece and Persia [3], so its evolution has come quite far. There are many methods on how to program resistance training plans, however, at its core there are three pillars that have to be manipulated: volume, intensity and frequency (shortly VIF). Let us define several terms in order to better understand VIF.

Training session (training unit) - a specific training bout, where a trainee performs several exercises. Between each exercise there is usually a period (minutes) where the trainee rests. Depending on a specific exercise there are several training variables that are tracked:

- **Sets** - the number of individual executions of an exercise, between which there is usually a rest period,
- **Rest period** - the time spent when resting between sets,
- **Load** - the amount of weight added onto the exercise besides the body weight, in order to increase stimulus,
- **Reps (repetitions)** - the number of executions of an exercise during one set,
- **Distance** - the distance travelled during one set,
- **Time** - the time spent during one set,
- **Effort** - the subjective rating of how difficult the set on an exercise was on an RPE or RIR scale.

RPE stands for rating of perceived exertion and the values of the scale for repetition-based exercises are present in the figure below.

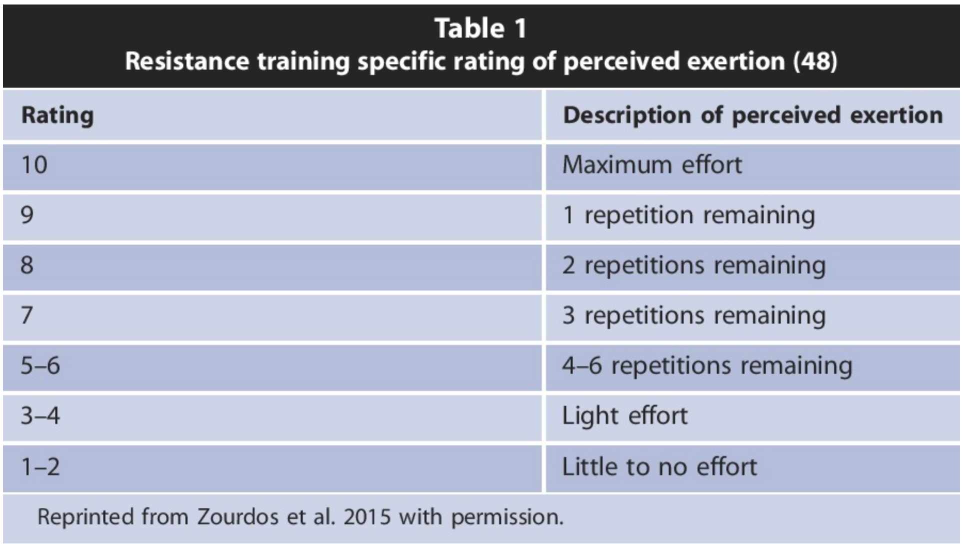


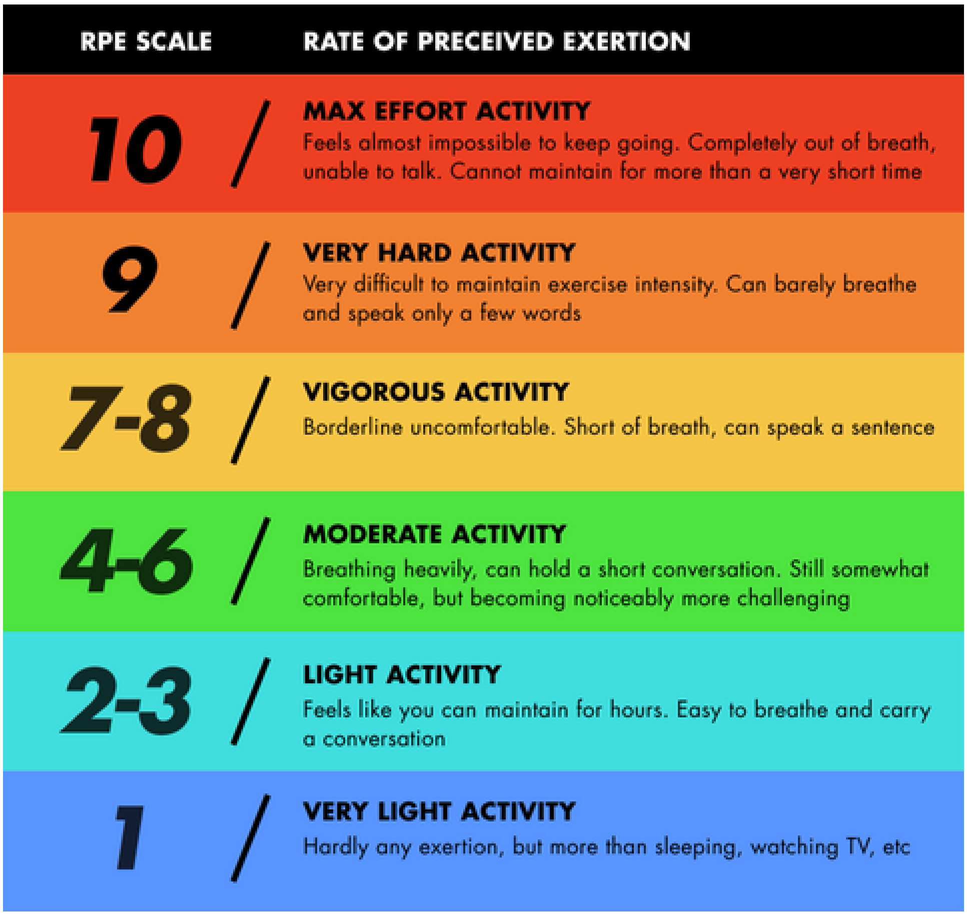
Table 1
Resistance training specific rating of perceived exertion (48)

Rating	Description of perceived exertion
10	Maximum effort
9	1 repetition remaining
8	2 repetitions remaining
7	3 repetitions remaining
5–6	4–6 repetitions remaining
3–4	Light effort
1–2	Little to no effort

Reprinted from Zourdos et al. 2015 with permission.

Figure 1.1: Repetition based RPE scale. [4]

Alternatively, **RIR - repetitions in reserve** is an inverted scale for RPE, where for instance RPE 8 = RIR 2 = two repetitions in reserve. RPE can still be used for exercises that do not track repetitions, usually cardio exercises, for



RPE SCALE	RATE OF PRECEIVED EXERTION
10 /	MAX EFFORT ACTIVITY Feels almost impossible to keep going. Completely out of breath, unable to talk. Cannot maintain for more than a very short time
9 /	VERY HARD ACTIVITY Very difficult to maintain exercise intensity. Can barely breathe and speak only a few words
7-8 /	VIGOROUS ACTIVITY Borderline uncomfortable. Short of breath, can speak a sentence
4-6 /	MODERATE ACTIVITY Breathing heavily, can hold a short conversation. Still somewhat comfortable, but becoming noticeably more challenging
2-3 /	LIGHT ACTIVITY Feels like you can maintain for hours. Easy to breathe and carry a conversation
1 /	VERY LIGHT ACTIVITY Hardly any exertion, but more than sleeping, watching TV, etc

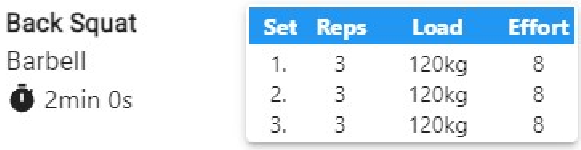
Figure 1.2: Non-repetition based RPE scale. [5]

example running. The meaning of each value on the scale is similar, although the cues are slightly different as it can be seen in the figure 1.2.

RIR is usually understood better, especially for beginners, however RPE seems to be the standard, not only in scientific studies, but also among more advanced lifters. As mentioned before, since RPE is also applicable to other types of exercises, it is typically a better choice for tracking effort, however, it is up to trainees and trainers what they choose. In the practical world, mostly values from 6 to 10 are used for RPE, because the closer the trainee is to failure, the easier it is to estimate how many repetitions to failure are left. Lower values of RPE tend to be very imprecise.

As previously stated, training variables depend on the chosen exercise, in figure 1.3 there is an example of an exercise barbell back squat that tracks sets, load, reps and effort and also treadmill running (in figure 1.4) that tracks sets, time, reps and effort.

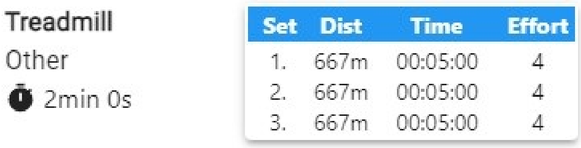
Each exercise trains certain muscle groups, for example Barbell Back squats targets several of them: quadriceps, hamstrings, glutes, hip flexors, calves, and abdominals. Some of them are more active than others, the primary ones are usually called *agonists* and the secondary ones are called *synergists*.



Back Squat
Barbell
⏱ 2min 0s

Set	Reps	Load	Effort
1.	3	120kg	8
2.	3	120kg	8
3.	3	120kg	8

Figure 1.3: Example of Barbell Back Squats in a performed training session in FitWolfe. [1]



Treadmill
Other
⏱ 2min 0s

Set	Dist	Time	Effort
1.	667m	00:05:00	4
2.	667m	00:05:00	4
3.	667m	00:05:00	4

Figure 1.4: Example of Treadmill running in a performed training session in FitWolfe [1]

When creating training sessions, coaches have to take that into account and plan according to the goal of the trainee, for instance a trainee might want to build strength on a specific exercise (use body leverages and muscles to generate maximum force) or build more muscle on specific parts of the body.

Additionally, the total work that is done on a specific muscle should be distributed in some way so that recovery is not compromised. This is usually done in a fixed amount of distinct training sessions over a specific amount of time. These blocks of training sessions are called **microcycles**. They are usually one week long since it is easier for trainees to integrate them into their weekly work/school schedules. For instance, we can have a weeklong microcycle that contains four training sessions. The distributions of sessions can be as follows:

1. Monday - Upper body 1 (Strength focused)
2. Tuesday - Lower body 1 (Strength focused)
3. Wednesday - Rest Day
4. Thursday - Upper body 2 (Hypertrophy focused)
5. Friday - Lower body 2 (Hypertrophy focused)
6. Saturday - Rest Day
7. Sunday - Rest Day

On the upper days, part or an almost entire musculature of upper body might be trained, and similarly on the lower body days part or an almost entire musculature of lower body might be trained. This is only a specific example on how the training volume can be distributed, some might choose to split up the body musculature into seven parts and have seven training sessions per week, or even have several full-body days during the week, where there might be a few exercises that target some lower body muscles and some exercises that target upper body muscles.

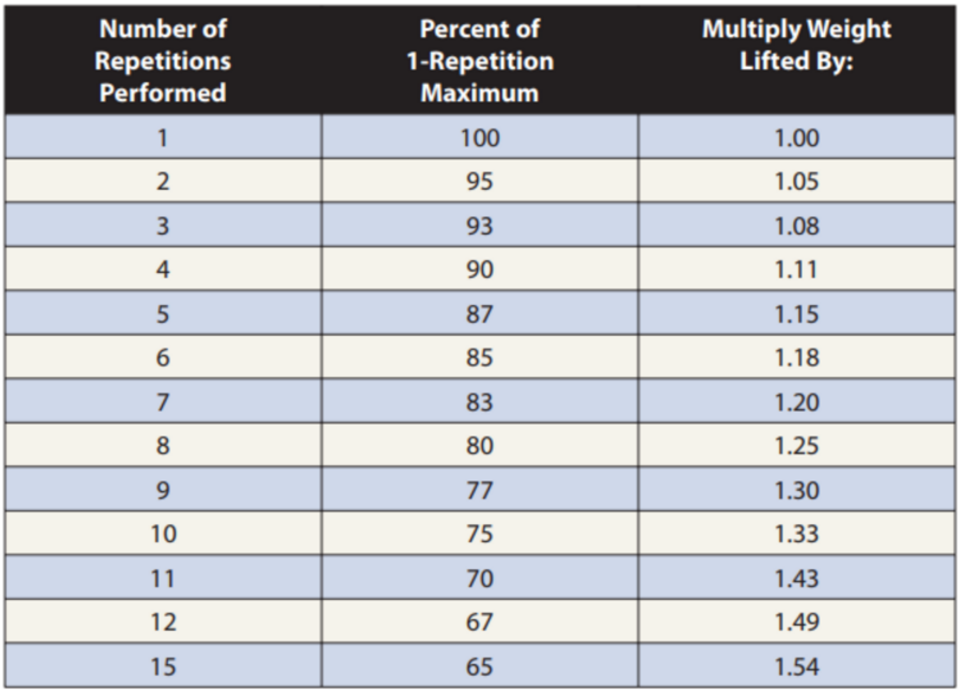
Some might argue that constantly switching exercises provides novel stimulus each time and makes it effective to grow muscles. However, that is not feasible when seeking strength improvements on specific exercises and even better muscle growth can be achieved with more structured approach, because of several reasons. Some of the exercises require complex skill that has to be learnt over a longer period of time and only then it allows for better muscle growth and enough neuromuscular adaptations [6] [7] [8]. Not to mention, more experience with the movement might mean lower risk of injury. It is also easier to gradually increase stimulus, by raising load, repetitions or sets on an exercise from one microcycle to another and therefore doing more work. The progress can also be assessed more easily, since we can compare improvements from microcycle to microcycle.

Now that the key terms are defined, it is time to describe the three pillars of resistance training programming - *VIF*. The first pillar - **volume** can be defined in two ways: training exercise volume which is $sets \times reps \times load$ and muscle group volume, which is number of sets performed on an exercise in a given period of time, mainly a microcycle. Muscle group volume is very important at the start, when a new training program is being created, because in accordance with that, exercises are chosen for the training sessions in a microcycle. The coach needs to find out how much muscle group volume was trainee able to handle before and plan the new program taking that into consideration. Different people can handle different amount of volume on different muscle groups, but as they get more advanced the capacity tends to increase.

The second pillar - **intensity** is the percentage of **one repetition maximum (1RM)** on a given exercise. For example, if a trainee can squat 100 kg for only 1 repetition and no more (reaches failure), it is an intensity of 100%. For 90% intensity, 90 kg in this example, the trainee should be able to perform 4 repetitions. In figure 1.5 we can see a relationship between intensity and number of repetitions. This table is only a generalization, some people might be able to perform more repetitions with certain intensity, others less. [9]

Trainees that prefer increaseing their strength should therefore have higher intensity (higher load) based training sessions [11]. One can also calculate that it is easier to accumulate training exercise volume with medium to high intensity than very high intensity, which is more appropriate for those who seek to increase their muscle size. Sets with very low intensity and a great number of repetitions will also help accumulate a lot of volume, however those sets generate more fatigue. This requires longer rest periods which could also mean less time-optimality of the session, with comparable hypertrophy benefits [12].

The third and the last pillar is **frequency**, which is the number of times a muscle group is trained over a period of time, mainly over a microcycle. If the frequency is only 1, then there is less frequent stimulus onto the muscles to



Number of Repetitions Performed	Percent of 1-Repetition Maximum	Multiply Weight Lifted By:
1	100	1.00
2	95	1.05
3	93	1.08
4	90	1.11
5	87	1.15
6	85	1.18
7	83	1.20
8	80	1.25
9	77	1.30
10	75	1.33
11	70	1.43
12	67	1.49
15	65	1.54

Figure 1.5: One repetition maximum (1rm) calculation table. [10]

grow and additionally, the volume in a given session is quite high. This almost always leads to the session dropping in quality as the given muscle group fatigues. The stronger effect on hypertrophy of higher training frequency was also noted in the scientific literature [13]. In addition, for strength focused individuals, the recovery of strength is usually shorter than one week, therefore it would only slow them down in practicing the movements they are interested in. With higher frequency, it is easier to distribute volume and provide more frequent stimulus for the muscles. Now that the definition of VIF is concluded, more of the structure of training programs can be laid out.

Above a microcycle we have a **mesocycle**, which is a period of several microcycles. This period can be used to assess progress over a longer timeframe, adjust goals or set new small ones, handle fatigue etc. For beginner trainees, cumulative fatigue is usually not an issue. They are able to progress linearly for many microcycles on end. Intermediate and advanced trainees might require a slightly different approach since in order to progress, they need higher muscle group volume than beginners, which comes at a greater recovery cost. The fatigue tends to accumulate faster than the body can adapt.

In order to avoid over-training, we can introduce a lighter microcycle commonly called **deload** at the end of each mesocycle, which helps reducing excessively accumulated fatigue. The combination of dropping volume and intensity can be used, for example, 30% reduction of volume on each training session tends to be sufficient. Light sessions do still have their meaning, as a trainee can practice the technique on exercises and there is still a slight

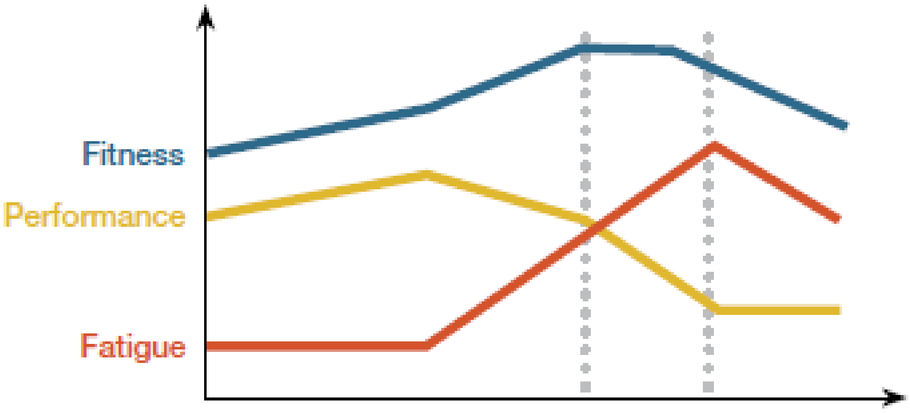


Figure 1.6: Fitness-Fatigue Model (over-training) [14]

stimulus. Another important thing is the client's consistency, a bigger gap between sessions could negatively impact new or existing habits. As an example of a training schedule, we can have a month long mesocycle with four microcycle. The volume and/or intensity rises from microcycle one to microcycle three and the fourth microcycle is a deload.

Finally in the resistance training programming structure we have a **macrocycle** which is usually the training program itself and which consists of several mesocycles. After the macrocycle ends, the overall progress is assessed, new goals are set and a new training program is planned, with possibly different exercise selection, number of sessions in a microcycle and so forth.

Now that it can be seen how training programs are created and adjusted, it is apparent that visualizing some statistics about training programs might help the coaches make better decisions when planning for their clients. Average muscle group volume on a previously performed training program can hint at how much volume should be planned in a new program. Rising one repetition maximum on a given exercise might hint at increasing strength levels and a greater potential to do more volume to stimulate muscle growth. Which exact statistics might be the most useful for the coaches? That will be discussed in the next chapter.

Chapter 2

Research of addition of training statistics in FitWolfe

2.1 Research methodology

The purpose of this research was to first consult the current state of the web application FitWolfe with several personal trainers and then discuss possible changes that would bring functionalities allowing the display of certain training statistics.

The first step was to reach out to the coaches and agree on next meetings. I have performed several meetings with each coach. The intro meeting included presenting FitWolfe user interface at that time and discussing new features. Then the initial user stories were drafted and provided to the personal coaches in a web tool optimalworkshop.com [15] where they were asked to perform *card sorting*. Basically, the user stories were written on individual cards, randomly shuffled and the coaches were asked to categorize them into three groups: necessary, nice to have and not needed. The coaches were reminded that the order in which they select the cards in each group matters and the ones they put more at the top mean they have more priority to them.

After the final user stories were chosen, some primary sketches of the new user interface were designed. These sketches were then used to construct high fidelity prototype in web tool Figma [17]. The prototype included only some core application flows and was meant to provide a simple overview of how the training statistics could be implemented in FitWolfe. The feedback was then collected.

2.2 Research results

The result of the intro meeting were several user stories that can be seen below. Some user stories are similar, the main differences are highlighted in bold. The key terms specific to the FitWolfe application are *Training day* and *Training plan*. Training plan is obviously the training program that is currently assigned to the client, however, Training day is sort of a holder for training sessions of the same type. As we had an example in the previous chapter, there was a microcycle with four training sessions, one of which was

Upper body 1 (UB1) training session. Training day would basically be a holder for all UB1 sessions, therefore, UB1 from the first microcycle, UB2 from the second microcycle and so forth.

[**A**] As a coach, I would like to see the progression of **training volume** on a chosen training exercise **on a Training Day in a specific Training plan**, in order to assess how the client is progressing.

- Training volume = sets×reps×load,
- Other types of exercises don't interest me that much (reps only, timed etc.),
- Specific rep range can be chosen, for example, 3-5 reps or even just triples,
- Line graph where x are dates of training sessions where the exercise was performed and y is training volume,
- On mouse hover over a graph point scheduled and performed sets in the session are displayed.

[**B**] As a coach, I would like to see the progression of **1 repetition maximum** on a chosen training exercise **on a Training day in a specific Training plan**, in order to assess how the client is progressing.

- Specific rep range can be chosen, for example, 3-5 reps or even just triples,
- 1RM will be calculated relative to load and reps, the best set is taken,
- Line graph where x are dates of training sessions where the exercise was performed and y is 1RM,
- On mouse hover over a graph point scheduled and performed sets in the session are displayed,
- The exercise will be compared session to session in the same training day if the rep range is not chosen, otherwise less sessions will be compared.

[**C**] As a coach, I would like to see the progression of **Load** on a chosen training exercise **on a Training day in a specific Training plan**, in order to assess how the client is progressing.

- Specific rep range can be chosen, for example, 3-5 reps or even just triples,
- Line graph where x are dates of training sessions where the exercise was performed and y is Load,
- On mouse hover over a graph point scheduled and performed sets in the session are displayed,

- The exercise will be compared session to session in the same training day if the rep range is not chosen, otherwise less sessions will be compared.

[D] As a coach, I would like to see the progression of **training volume** on a chosen training exercise **in a specified time frame**, in order to assess how the client is progressing.

- Training volume = sets×reps×load,
- Other types of exercises don't interest me that much (reps only, timed etc.),
- Specific rep range can be chosen, for example, 3-5 reps or even just triples,
- Line graph where x are dates of training sessions where the exercise was performed and y is training volume,
- On mouse hover over a graph point scheduled and performed sets in the session are displayed.

[E] As a coach, I would like to see the progression of **1 repetition maximum** on a chosen training exercise **in a specified time frame**, in order to assess how the client is progressing.

- Specific rep range can be chosen, for example, 3-5 reps or even just triples,
- 1RM will be calculated relative to load and reps, the best set is taken,
- Line graph where x are dates of training sessions where the exercise was performed and y is 1RM,
- On mouse hover over a graph point scheduled and performed sets in the session are displayed.

[F] As a coach, I would like to see the progression of **Load** on a chosen training exercise **in a specified time frame**, in order to assess how the client is progressing.

- Specific rep range can be chosen, for example, 3-5 reps or even just triples,
- Line graph where x are dates of training sessions where the exercise was performed and y is Load,
- On mouse hover over a graph point scheduled and performed sets in the session are displayed.

[G] As a coach, I would like to **compare a performed vs scheduled training variable on a chosen training plan**, in order to assess how often I plan more than/the same/less than the client can handle.

- for example, pie chart is used with three sections:
- Percentage of scheduled = performed,
- Percentage of scheduled < performed,
- Percentage of scheduled > performed.

[H] As a coach, I would like to **compare a performed vs scheduled training variable in a specified time frame**, in order to assess how often I plan more than/the same/less than the client can handle.

- for example, a pie chart is used with three sections:
- Percentage of scheduled = performed,
- Percentage of scheduled < performed,
- Percentage of scheduled > performed.

[I] As a coach. I would like to **compare per session RPE on a Training day in a specific Training plan**, in order to assess how the client perceived the difficulty of sessions over time.

- Line graph where x are dates of training sessions where the exercise was performed and y is per session RPE.

[J] As a coach, I want to **compare the goal number of sessions per week we set with the client to the actual number of sessions performed per week**, so that I can assess how well the client follows the plan.

- for example, line graph can be used where x are individual weeks and y is the number of sessions,
- One line a graph can display current goal and a second one could display the number that was actually performed.

[K] As a coach, I want to **view the change in muscle group volume** (number of sets per muscle group) **in a specified time frame**, so that comparing it to other metrics I can plan certain muscle group volume in next training plans.

- for example, line graph can be used where x axis are individual weeks and y is the number of sessions,
- One line in a graph can display current goal and a second one could display the number that was actually performed,
- The muscle groups could be selectable, for example, choose chest and triceps,
- One line in a graph could correspond to each muscle group,

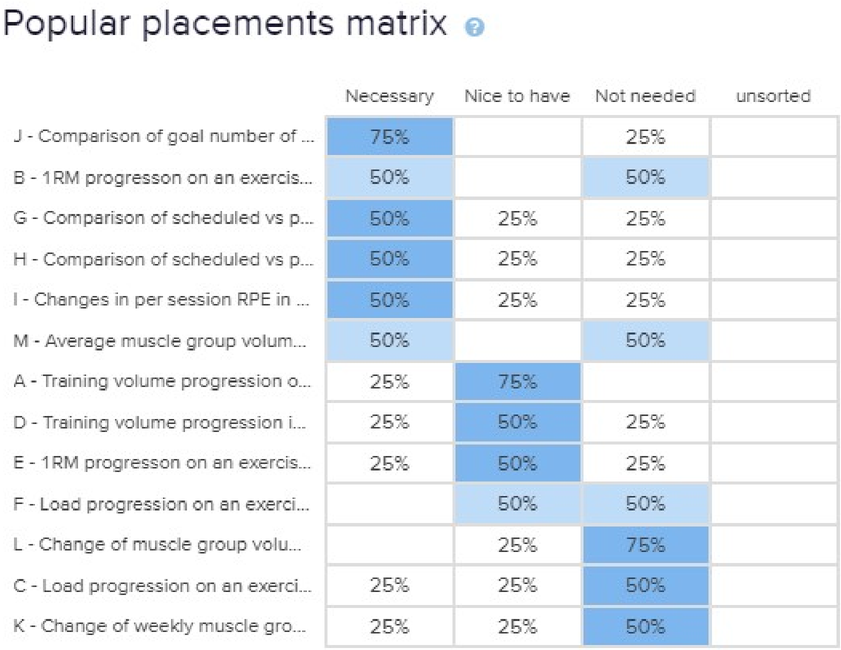
- An alternative to graph could be a table.

[**L**] As a coach, I want to **view muscle group volume per microcycle** (number of sets per muscle group) **in a selected (past or current) training plan**, so that I can assess if it is appropriate to the client, (or for past plans find out how much they performed).

- for example, line graph can be used where x axis are individual microcycles and y is a total number of set per muscle group,
- The muscle groups could be selectable, for example, choose chest and triceps,
- One line in a graph could correspond to each muscle group,
- An alternative to graph could be a table.

[**M**] As a coach, I want to **view the average muscle group volume per microcycle** (number of sets per muscle group) **in a selected (past or current) training plan**, so that I can assess if it is appropriate to the client, (or for past plans find out how much they performed).

Since the amount of user stories is quite high and would require a lot of development time, I needed to choose those that are of the highest priority to the coaches. Card sorting was executed for that purpose and its results are present in the following figure.



Popular placements matrix

	Necessary	Nice to have	Not needed	unsorted
J - Comparison of goal number of ...	75%		25%	
B - 1RM progresson on an exercis...	50%		50%	
G - Comparison of scheduled vs p...	50%	25%	25%	
H - Comparison of scheduled vs p...	50%	25%	25%	
I - Changes in per session RPE in ...	50%	25%	25%	
M - Average muscle group volum...	50%		50%	
A - Training volume progression o...	25%	75%		
D - Training volume progression i...	25%	50%	25%	
E - 1RM progresson on an exercis...	25%	50%	25%	
F - Load progression on an exerci...		50%	50%	
L - Change of muscle group volu...		25%	75%	
C - Load progression on an exerci...	25%	25%	50%	
K - Change of weekly muscle gro...	25%	25%	50%	

Figure 2.1: Card sorting result [15]

User stories J, B, G, H, I, M have clear placements with the highest priority. Furthermore, user story E was chosen, since it has a lot of similarities with the user story B so the implementation would not be drastically more difficult.

Consequently, sketches for user stories J, B, G and M were created in Microsoft Whiteboard desktop application [16]. Low fidelity prototype (the sketches) was not used for user testing, because it was difficult for the coaches to find time for consultations and time was running short. I decided that the sketches will be used as an outline for high fidelity prototype and the final consultations over it would have to suffice.

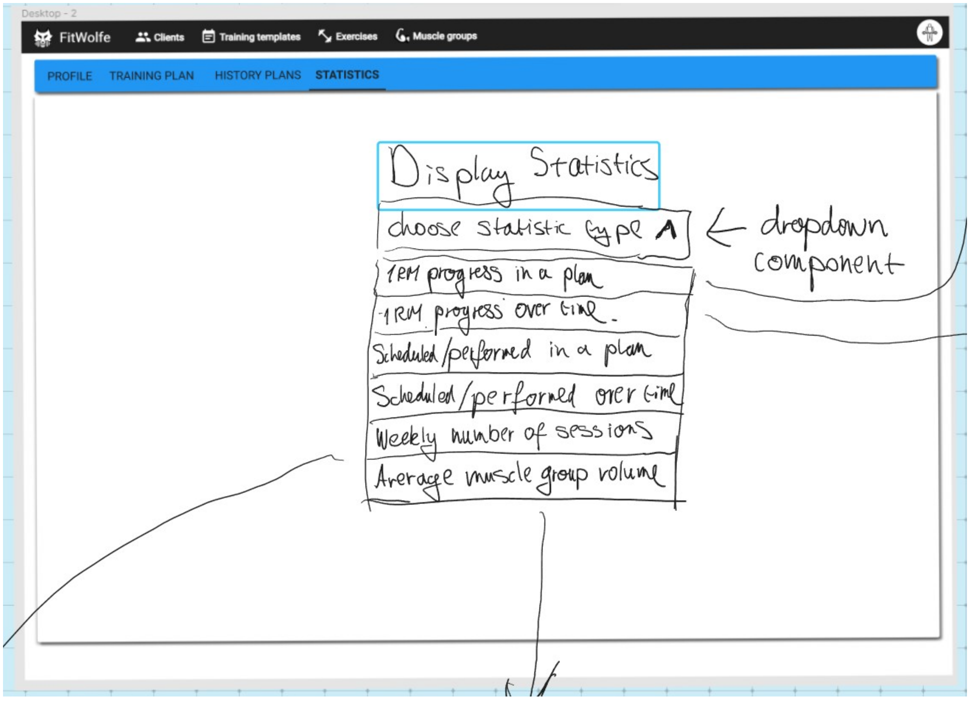


Figure 2.2: Low-fidelity prototype of statistics intro screen in Microsoft Whiteboard [16]

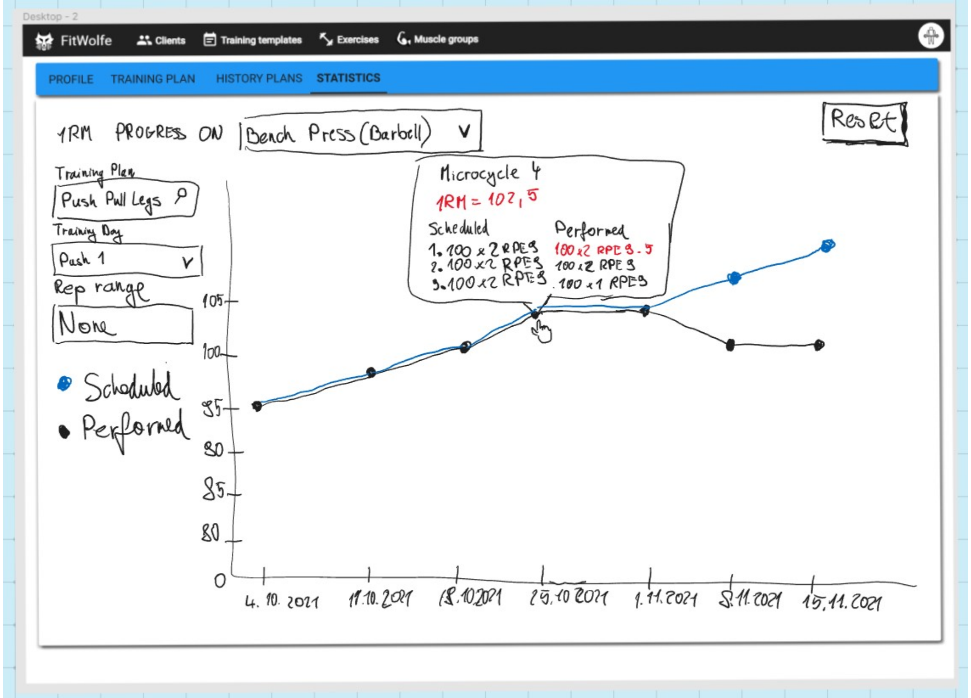


Figure 2.3: Low-fidelity prototype of one rep max progress graph in Microsoft Whiteboard [16]

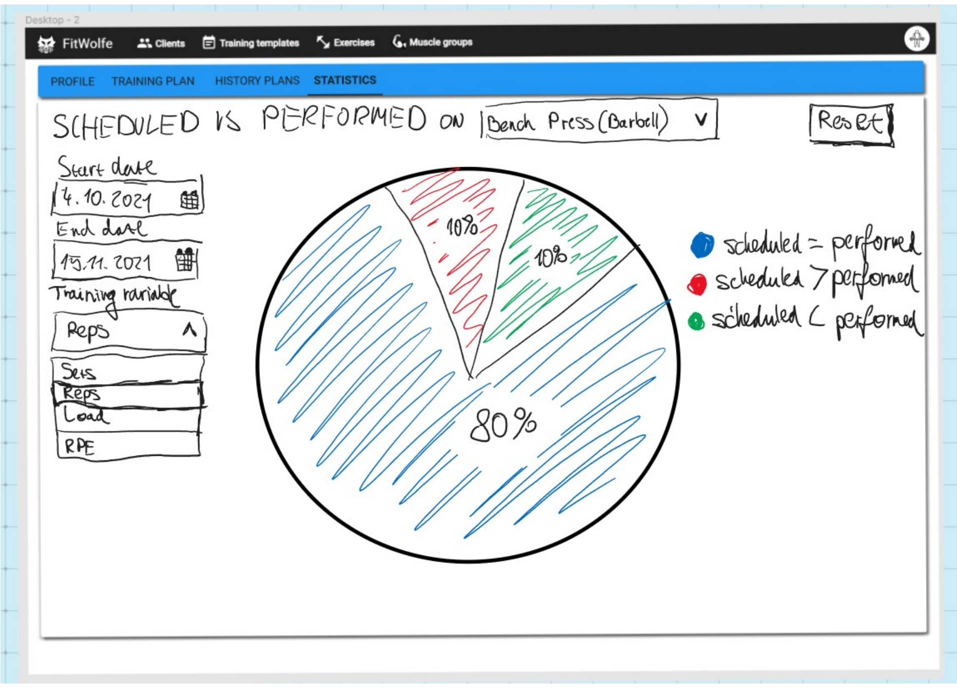
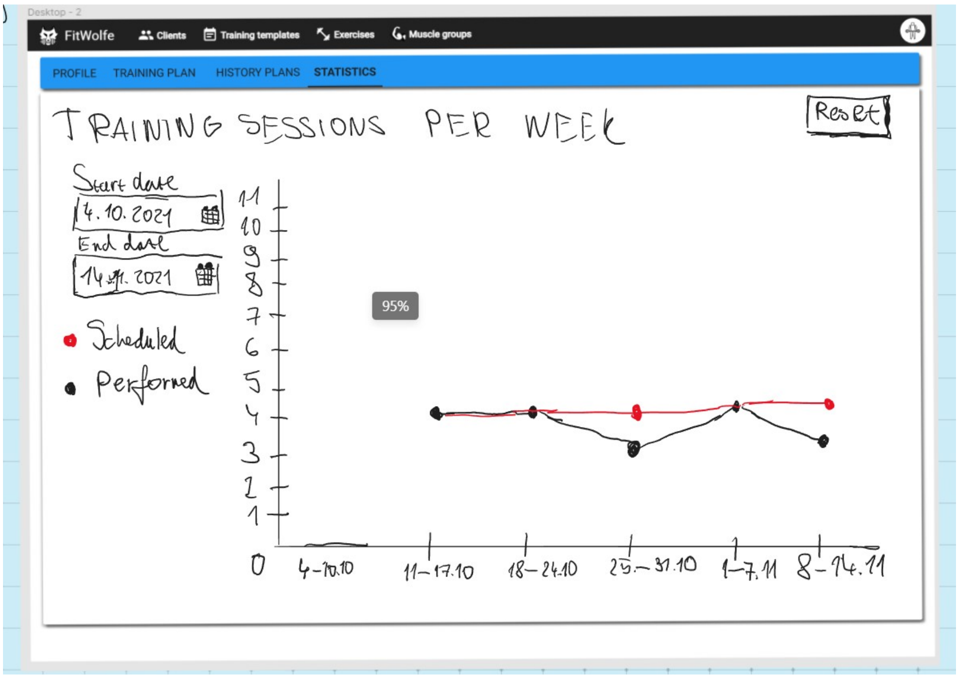


Figure 2.4: Low-fidelity prototype of scheduled vs performed graph in Microsoft Whiteboard [16]



TRAINING SESSIONS PER WEEK

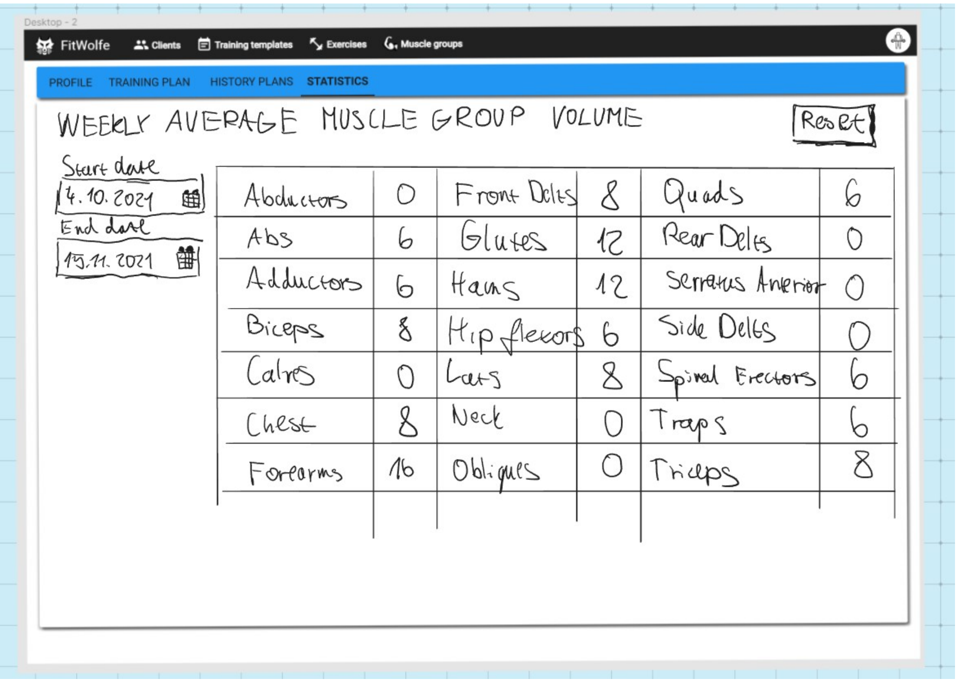
Start date
4.10.2021

End date
14.11.2021

● Scheduled

● Performed

Figure 2.5: Low-fidelity prototype of weekly training sessions graph in Microsoft Whiteboard [16]



WEEKLY AVERAGE MUSCLE GROUP VOLUME

Start date
4.10.2021

End date
15.11.2021

Abductors	0	Front Delts	8	Quads	6
Abs	6	Glutes	12	Rear Delts	0
Adductors	6	Hams	12	Serratus Anterior	0
Biceps	8	Hip flexors	6	Side Delts	0
Calves	0	Lats	8	Spinal Erectors	6
Chest	8	Neck	0	Traps	6
Forearms	16	Obliques	0	Triceps	8

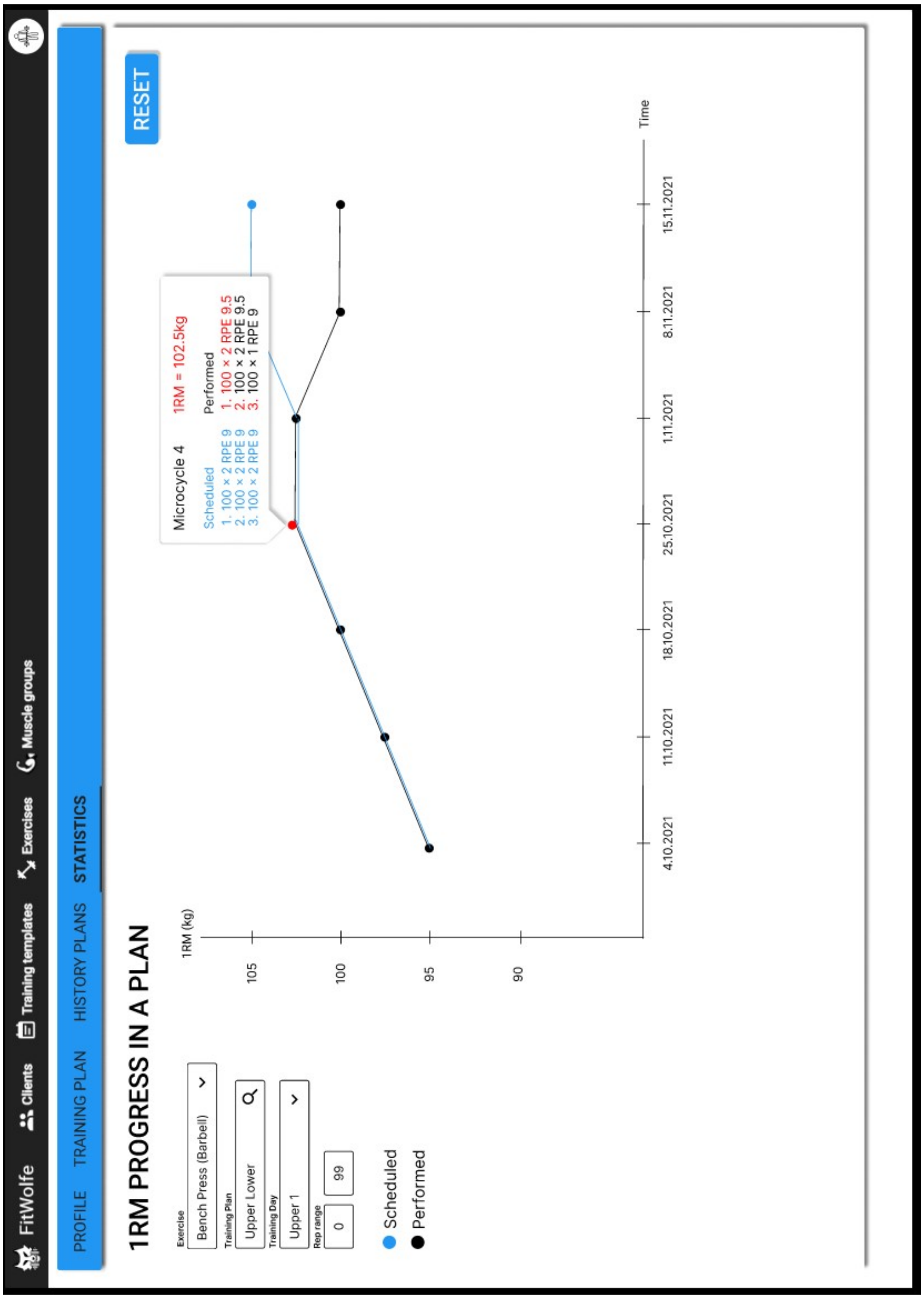
Figure 2.6: Low-fidelity prototype of average volume table in Microsoft Whiteboard [16]

For each user story of the aforementioned four, there was a screen made in tool figma.com [17], where there were basic actions, like hovering over points in a graph to display additional information, all input components and a return back to the menu.

Finally, meetings were conducted with the coaches to discuss the created prototype. Overall, the coaches were satisfied with the design, and there were a few suggestions about what functionalities could be added. Mainly it was extending information provided by the graphs, more possibilities to filter the data coupled with a few improvements in design.

The following figures preset high fidelity prototypes in Figma 2.7. Since most of them were the same in terms of structure as low fidelity prototypes, only two will be presented, one rep max statistic screen as an example and weekly training sessions statistics screen because it was changed. The previous proposition - line graph was not conveying the information as clearly, therefore one of the personal trainers suggested to put it in a bar graph.

At this time, the web application is intended only for desktop devices, therefore these and some of the other figures will be presented in landscape format. The figures would otherwise not be readable, or would have to be split, which would not be ideal to showcase them properly.



1RM PROGRESS IN A PLAN

Exercise: Bench Press (Barbell)

Training Plan: Upper Lower

Training Day: Upper 1

Rep range: 0 99

● Scheduled

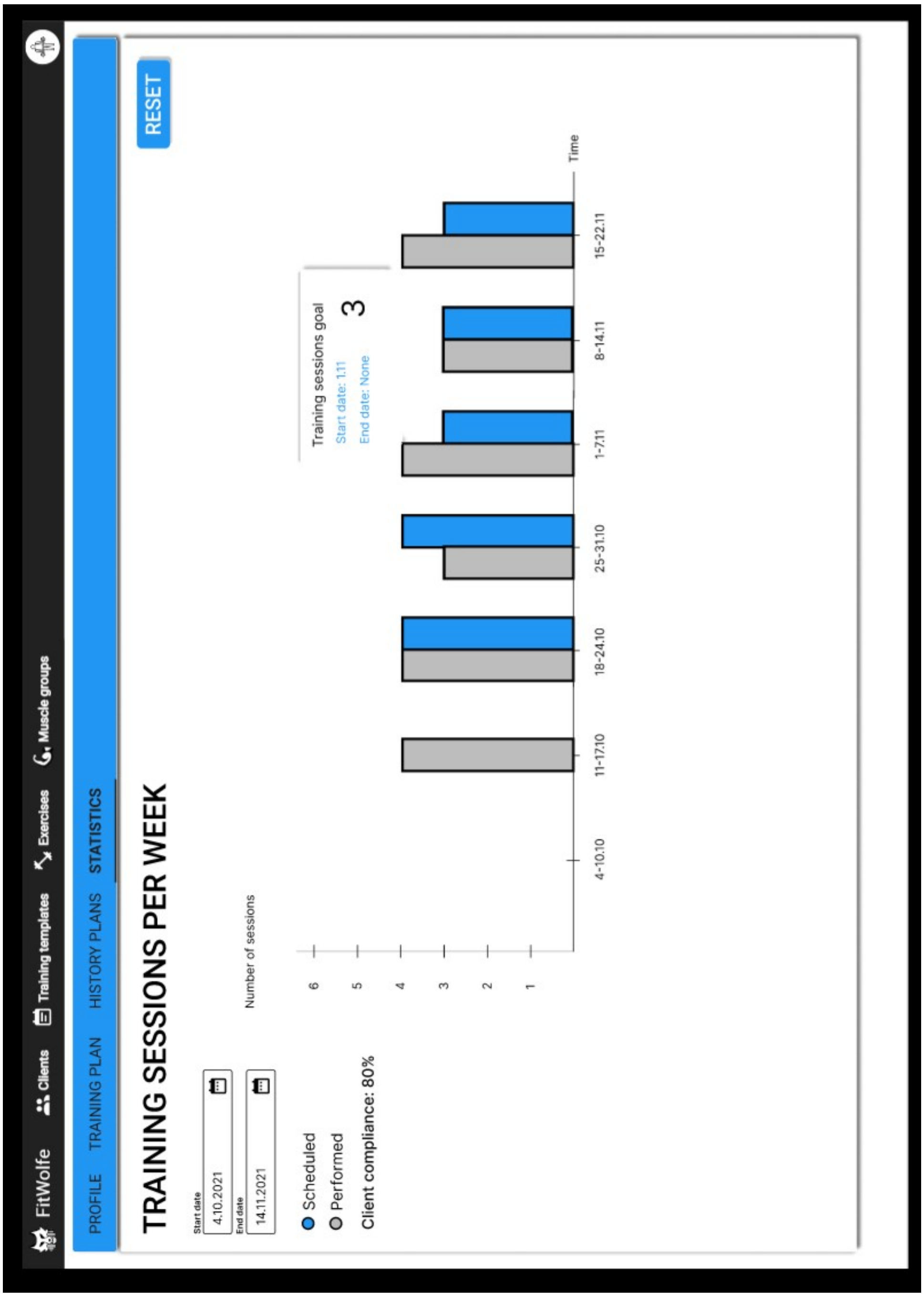
● Performed

Microcycle 4 1RM = 102.5kg

Scheduled
1. 100 × 2 RPE 9
2. 100 × 2 RPE 9
3. 100 × 2 RPE 9

Performed
1. 100 × 2 RPE 9.5
2. 100 × 2 RPE 9.5
3. 100 × 1 RPE 9

Figure 2.7: High-fidelity prototype of one rep max progress graph in Figma [17]



Start date: 4.10.2021

End date: 14.11.2021

Client compliance: 80%

Training sessions goal: 3

Start date: 1.11

End date: None

Figure 2.8: High-fidelity prototype of weekly training sessions graph in Figma [17]

2.3 Research conclusion

The process of programming and periodization of resistance training is a complex topic. Personal coaches have to plan training programs for their clients with several things in mind. The experience of the client, individual variability in responding to different training stimuli, physical capabilities and also variable sources of stress that the clients might have.

A web application FitWolfe already provides possibilities to create training programs in a structured manner, however, an extension should be implemented to allow the observation of training statistics and their progression in time. These statistics should provide information about the capabilities of the client and an overview of their improvements over time.

Several coaches were contacted in order to gather necessary information to begin the analysis and implementation of training statistics. Prototyping approach was used where new functionalities were transformed from the first textual overview - user stories - up to the high-fidelity prototype. All the necessary information was collected during consultations with the personal coaches. Next, we will look at detailed analysis and the following implementation.

Chapter 3
Software analysis

3.1 Overview

In this chapter, a detailed software analysis will be presented, which was performed based on the initial research. A refinement of the previously drafted user stories will be introduced with next proposed changes to class diagram of the application. Additionally, the architecture and technologies of the application will be mentioned together with their adjustments. Finally, the endpoints for new back end API for obtaining new training statistics will be described.

3.2 Refined User Stories

After further analysis following the consultations with the personal trainers it was decided to merge some of the previous user stories, since their implementation turned out to be quite similar.

3.2.1 One Rep Max Progress

One repetition maximum statistic was merged with its *time range* variant (previously user story E) and its *training plan* variant (previously user story B).

User story: As a Coach, I would like to see the progression of **1 repetition maximum** on a chosen training exercise, in order to assess how the client is progressing.

- I want to select a time period from - to.
- I want to be able to specify a training plan and a specific training day.
    - *From* will be restricted to the plan's start date, it can be set to later.
    - *To* will be restricted to the plan's end date, it can be set to sooner.
    - If the end date is not present in the plan, the current date will be used by default.

- Specific rep range can be chosen, for example, 3-5 reps or even just triples.
- Specific RPE-RIR range can be chosen, for example, RPE 8.
- 1RM will be calculated relative to load and reps, the best set is taken.
- Line graph where x are dates of training sessions where the exercise was performed y is 1RM.
- On mouse hover over a graph point scheduled and performed sets in the session are displayed.

3.2.2 Scheduled vs Performed

Similarly to the previous user story the Scheduled vs Performed training statistic was merged from previous user stories G (specific to training plan) and H (specific to time frame).

User story: As a Coach, I would like to **compare a performed vs scheduled training variable on a chosen training exercise**, in order to assess how often I plan more than/the same/less than the client can handle.

- I want to select a time period from - to.
- I want to be able to specify a training plan and a specific training day.
  - *From* will be restricted to the plan's start date, it can be set to later.
  - *To* will be restricted to the plan's end date, it can be set to sooner.
  - If the end date is not present in the plan, the current date will be used by default.
- I want to be able to choose a training variable: Sets, Reps, Load, Effort.
- Pie chart is used with three sections:
  - percentage of scheduled = performed;
  - percentage of scheduled < performed;
  - percentage of scheduled > performed.

3.2.3 Weekly Training Sessions

Weekly training sessions statistic was changed from line graph to bar graph as mentioned previously in the research section. It will be necessary to implement a possibility to set weekly training session goals in order to allow this statistic.

User story: As a Coach, I want to **compare the goal number of sessions per week** we set with the client **to the actual number of**

sessions performed per week, so that I can assess how well the client follows the plan.

- A bar graph will be used where x axis are individual weeks and y is the number of sessions, there are two bars at each x point, one is scheduled the other is performed.
- Include the client's compliance with the goal.

3.2.4 Training Session Effort

The user story Per Session RPE had only a name change to Training Session Effort since some personal trainers might prefer to use RIR.

User story: As a Coach. I would like to compare **effort of training sessions**, in order to assess how the client perceived the difficulty of sessions over time.

- I want to be able to specify a training plan and a specific training day.
- A line graph will be used where x are dates of training sessions where the exercise was performed y is per session effort.
- Graph point can contain more information on hover, for example, the names of sessions, with their microcycle number.

3.2.5 Average Microcycle Volume

Finally, the user story Average Microcycle Volume was extended with a time range filter and a distinction between a number of sets certain muscle group was an agonist (primary mover) on an exercise or a synergist (secondary mover). The latter change will require adjustments in data model of muscle groups and filling the database with the relevant data, so that the statistics are then relevant.

User story: As a Coach, I want to view the average muscle group volume per microcycle (number of sets per muscle group), so that I can assess if it is appropriate for the client, or for past plans find out how much they performed.

- I want to select a time period from - to.
- I want to be able to specify a training plan and a specific training day.
  - *From* will be restricted to the plan's start date, it can be set to later.
  - *To* will be restricted to the plan's end date, it can be set to sooner.
  - If the end date is not present in the plan, the current date will be used by default.

- Include two numbers - the average sets the muscle was a primary mover and the second - where it was a secondary mover
- On hover over a number of sets on a muscle, exercises are listed.
- Synergists should be close to each other in the table.

3.3 System Architecture and Technologies

3.3.1 Deployment

The current system architecture consists of several architectural patterns. The application as a whole is client-server, where the server is Linux Ubuntu hosted by the student organization Silicon Hill under CTU. This machine serves requests from clients. Clients in this instance are machines that the users of the system operate to be able to interact with the application.

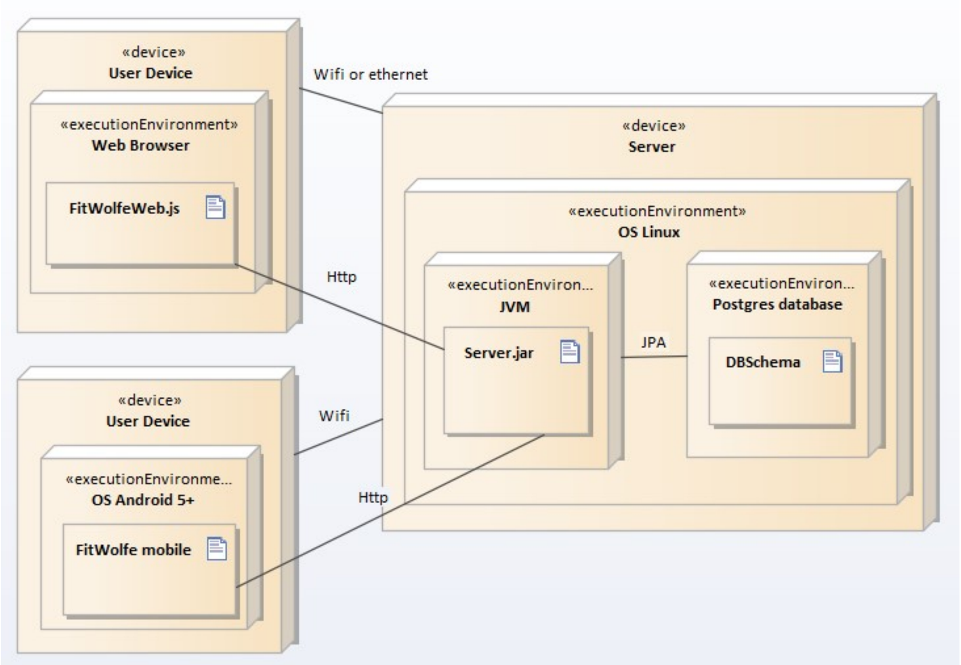


Figure 3.1: Deployment diagram [Author]

There are two types of clients depending on real application users – personal coaches and trainees. The coaches access the system through JavaScript application running in their web browser therefore they can use the system possibly from any device - laptops, desktops, tablets, Android and IOS smartphones. However, the application is currently intended mainly for big screens because there, the user can better compare changes between their client's training sessions. In the future it will be adjusted to be more responsive. The trainees access the system through React Native application on their Android device.

It is possible to use React Native applications on IOS devices, however, in this instance it was not feasible. IOS devices are restricted to use applications that are officially published to Apple Store and the licenses to publish applications are prohibitively expensive considering this is a currently non-commercial project.

Besides client-server architecture, the system also complies with layered architecture, which is quite standard for SpringBoot Java applications. In the following figure we can see a simplified version of layered architecture.

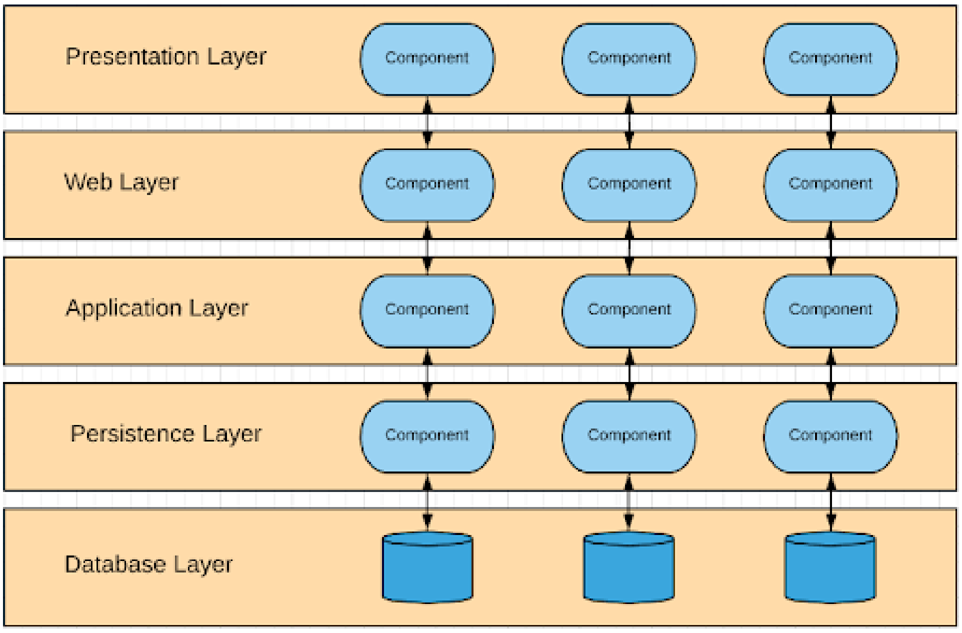


Figure 3.2: Layered architecture [Author]

3.3.2 Back end Technologies

The back end part of Fitwolfe is written in Java 17. Java was chosen due to various reasons:

1. I have had the most experience building applications in Java;
2. It has a lot of resources and examples online, therefore finding solutions to occurring problems is faster;
3. Its support for Object Oriented Programming and strong type safety enforces better coding practices.

It would be quite difficult to implement a web server with pure Java, hence framework SpringBoot [18] was chosen. Again, the reasoning behind it was my experience, but also its ease of use. The developer does not have to write huge amounts of code to be able to serve resources to front end. Moreover, it has an embedded Tomcat, Jetty or Undertow HTTP web server, which reduces the necessity to deploy WAR files.

For dependency management, running unit tests and building the back end and web front end Maven [19] is used. Unit tests are written with the help of framework JUnit [20] which is included in SpringBoot dependencies by default. To simulate necessary data states of the applications mocks are used from the framework Mockito, but also an in-memory database Hsqldb [21] is required in some situations.

In the non-test environment database PostgreSQL [22] is used. To cope with changes to data model of the application, a data migration tool Flyway [23] was employed, which uses developer defined SQL scripts to apply changes to the database. Business logic of applications often requires validation since most of the incoming data is from the user input and might not be correct. In this matter, javax.validation dependency helped. FitWolfe started with the older versions where Java records (code-compact classes) were not yet present and to reduce the amount of code that had to be written Lombok [24] was added as a dependency.

Documentation for applications is important, especially when more developers work on it. Outside of the scope of this project there is one more developer working on FitWolfe and in the future there might be more. Considering that, Springdoc Openapi [25] dependency is also part of the back end. This dependency helps to generate api documentation and also adds an endpoint to the application where this documentation can be read. This aids greatly during the integration of back end and front end. Finally, the last package that should be mentioned is Sentry [26]. It allows for better error monitoring in the back end application.

3.3.3 Web Front end Technologies

Web front end of FitWolfe was built using JavaScript library ReactJS [27]. It focuses on building user interfaces with components and allows re-rendering only those that are necessary when the data changes. Since JavaScript is not a strongly typed language, it is also error prone for not very experienced or inattentive developers. This can be partly eliminated by using TypeScript [28] which is an extension of JavaScript that introduces type inference.

Once the React application grows to a certain point, it starts to be more and more difficult to handle its data state. In this matter a library Redux [29] can help. Its usage also promotes a separation of business logic and data fetching from purely visual part of the front end - components. Bigger front end applications also have several parts, for each a separate route usually exists. A library React Router [30] helps to define routes and an effective traversal between them. The final package that is important in FitWolfe stack is Material UI [31] which is a library that contains pre-defined React components with easier styling and already implemented validation and business logic to handle their state.

3.3.4 Mobile Front end Technologies

The technologies used on mobile front end are similar to web front end, since the codebase of React Native [32] is almost exactly the same as in React, the only difference are mobile native components compared to web components. In place of Material UI, libraries React Native Elements [33] and React Native Paper [34] are used. Unfortunately, no single component library was found that would contain all needed UI components and the creation of custom ones would be more time costly.

In place of React Router, there is React Navigation [35], which is made for mobile applications. At last Expo [36] handles the creation of development and production builds. During the production build, an .apk file is generated from the sources which can be then used to install the FitWolfe mobile on the Android device.

3.3.5 New Technologies

Due to the mobile front end being written in the same language, a lot of code was shared, such as types for representing back end resources, helper functions, wrappers for API calling etc. This led to the creation of a private library fitwolfe-types which groups this code into one place and helps for more efficient changes in both repositories. Regarding the new external dependencies, it was only needed to add one, that would provide an ability to simply create graphs for newly implemented training statistics. The training statistics are currently considered only for personal trainers, thus those libraries that use web components were sufficient.

I performed quick research on Google to look for the current possibilities [37]. They are plentiful, however, not all provide what was necessary for the implementation of the features. Such was the case with library Devextreme reactive [38], it is highly compatible with already used library Material UI, however, it did not provide all the functionalities that were required in the user stories. It was replaced with library Recharts [39], which allows fast creation of graphs that are relatively highly customizable. The graphs provide a simple look, paired with animation and responsiveness thanks to their use of scalable vector graphics (SVG) and Data Driven Documents (D3) [40].

3.4 Data model

3.4.1 Previous data model

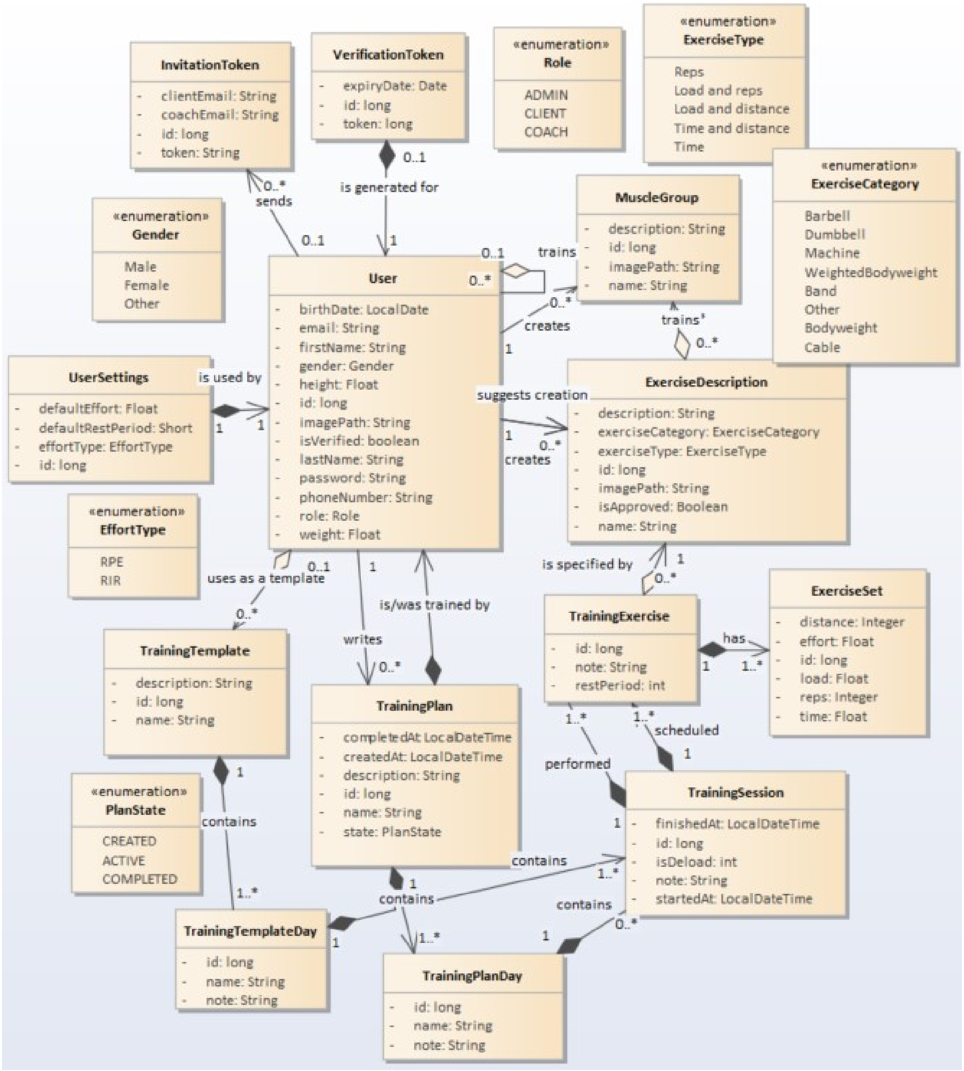


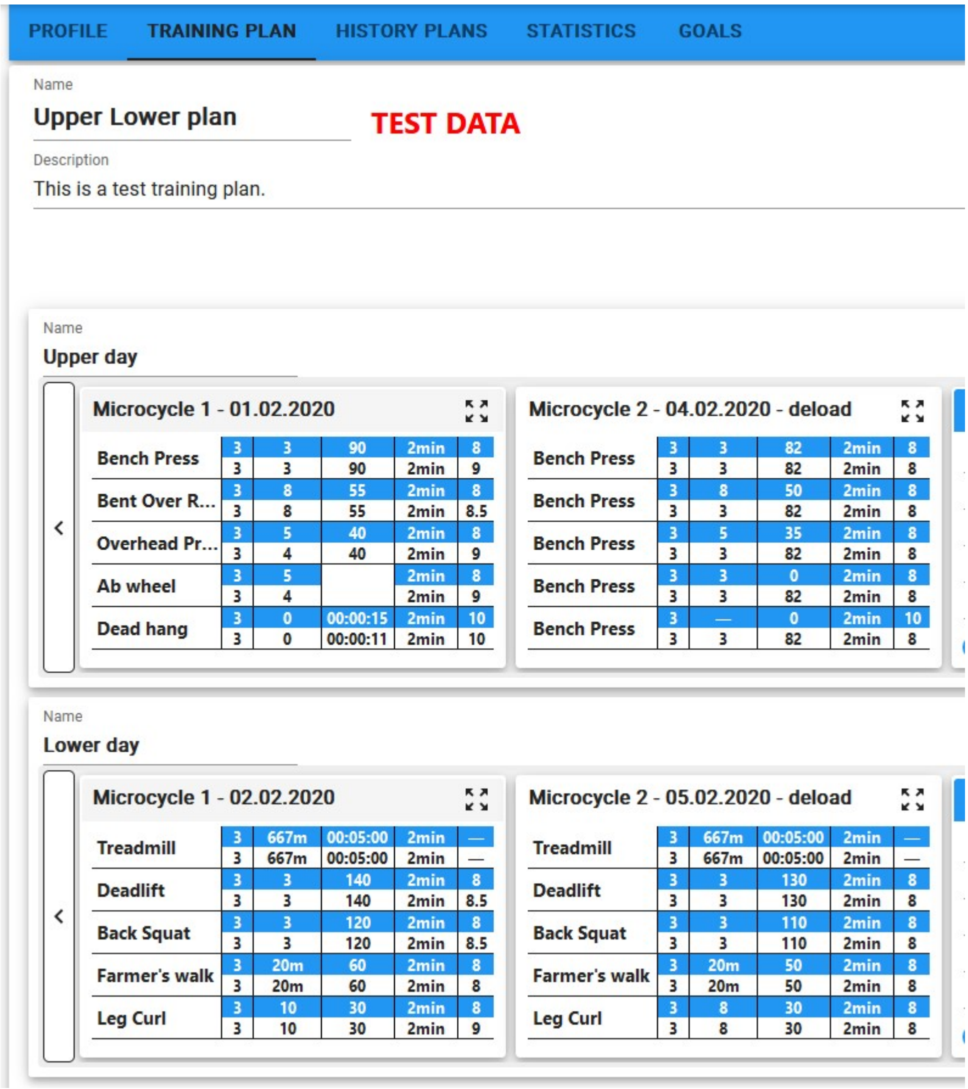
Figure 3.3: Data model before the implementation of statistics [Author]

To quickly describe the relations between the entities in the diagram before implementation of statistics, the most important entity is User. The user obtains a role only after their account is confirmed using the verification token entity.

There are three roles for users. Admin creates and edits muscle groups and exercise descriptions and also approves new exercise descriptions suggested by Coach user. The Coach user creates training templates which can be used to simplify training plan creation for clients. Client user can have a coach, this relationship is created by an invitation whose data is held by the class invitation token.

Clients can additionally have several training plans, only one of them can however have a state CREATED which signifies that the plan is current for the client but does not have any finished training sessions. Once at least one of the training sessions gets finished, the plan transitions into the state ACTIVE. When time comes for the training plan to end, it is then transitioned to the state COMPLETED. Multiple training plans can have this state.

The relationship between training plan and its days and the training sessions inside a day will be better explained in figure 3.4



Name
Upper Lower plan

Description
This is a test training plan.

Name
Upper day

Microcycle 1 - 01.02.2020

Bench Press	3	3	90	2min	8
	3	3	90	2min	9
Bent Over R...	3	8	55	2min	8
	3	8	55	2min	8.5
Overhead Pr...	3	5	40	2min	8
	3	4	40	2min	9
Ab wheel	3	5		2min	8
	3	4		2min	9
Dead hang	3	0	00:00:15	2min	10
	3	0	00:00:11	2min	10

Microcycle 2 - 04.02.2020 - deload

Bench Press	3	3	82	2min	8
	3	3	82	2min	8
Bench Press	3	8	50	2min	8
	3	3	82	2min	8
Bench Press	3	5	35	2min	8
	3	3	82	2min	8
Bench Press	3	3	0	2min	8
	3	3	82	2min	8
Bench Press	3	—	0	2min	10
	3	3	82	2min	8

Name
Lower day

Microcycle 1 - 02.02.2020

Treadmill	3	667m	00:05:00	2min	—
	3	667m	00:05:00	2min	—
Deadlift	3	3	140	2min	8
	3	3	140	2min	8.5
Back Squat	3	3	120	2min	8
	3	3	120	2min	8.5
Farmer's walk	3	20m	60	2min	8
	3	20m	60	2min	8
Leg Curl	3	10	30	2min	8
	3	10	30	2min	9

Microcycle 2 - 05.02.2020 - deload

Treadmill	3	667m	00:05:00	2min	—
	3	667m	00:05:00	2min	—
Deadlift	3	3	130	2min	8
	3	3	130	2min	8
Back Squat	3	3	110	2min	8
	3	3	110	2min	8
Farmer's walk	3	20m	50	2min	8
	3	20m	50	2min	8
Leg Curl	3	8	30	2min	8
	3	8	30	2min	8

Figure 3.4: Example training plan in FitWolfe [Author]

Training plan *Upper Lower* contains two training days which hold their corresponding training sessions. The first microcycle is comprised of a first training session in every training day, therefore entity training day represents sort of a holder for training sessions of the same structure. Training sessions then contain training exercises which are entities that pair the data about training variables, present in the entity exercise set and a specific real-world exercise, for example, Barbell Back Squat which is represented by the entity exercise description.

In the following section we will look at how the data model should change to allow the retrieval of training statistics.

3.4.2 New data model

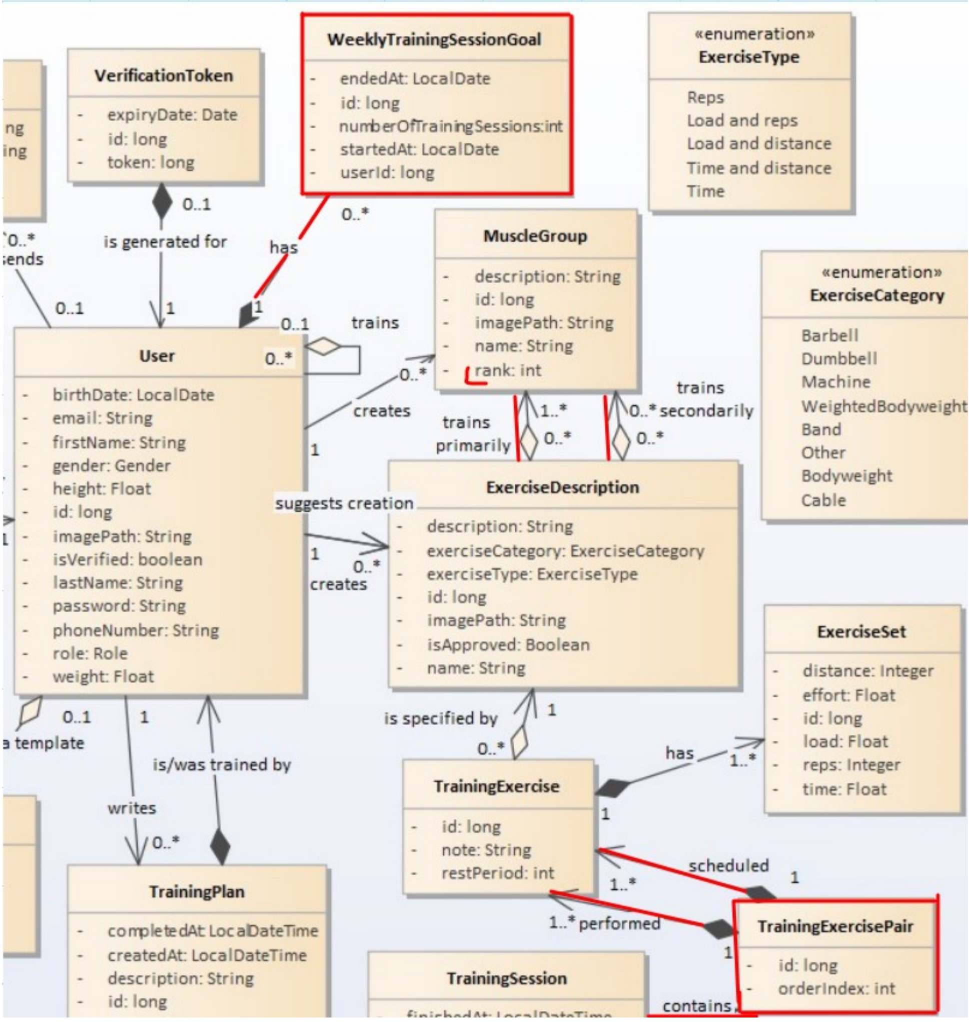


Figure 3.5: Data model after the implementation of statistics [Author]

The figure 3.5 displays only the entities that changed. They are highlighted with red color. User now can have several weekly training session goals. Only one is active in a certain time frame. This data is then displayed in the weekly training session graph where the goal is compared with the actual number of training sessions performed.

The statistics for average muscle group volume required the collection of muscle groups on the exercise description to be split into two, the primary muscle groups which represent the agonist muscles on the exercise and secondary muscle groups which represents synergist muscles. The new attribute rank on a muscle group will help with the ordering of muscle groups in a way so that synergist muscle groups are displayed close to each other in the statistics table.

The last important change was adjusting the relationship between training session and training exercises. Now, the training session contains a collection of training exercise pairs, where each of them contains a scheduled and a performed exercise. In the previous model it would have been a bit difficult to pair scheduled and performed exercises directly, because there can be less or more performed exercises than scheduled. Moreover, if there were two scheduled exercises with the same exercise description it would be difficult to write a database query that would return the correct data while taking into account all possible states the data can be in.

It is now relatively easy to obtain Scheduled vs Performed statistics for training exercise pairs. It also helped simplify some of the logic in the implementation of other statistics and made front end rendering logic of the training session data more straightforward.

3.5 Statistics Api Specification

The specification for the API was first written in the Openapi 3 format based on the user stories. Online Swagger editor [41], where it is possible to write such specification is also able to generate Java classes. Unfortunately, the readability of the generated code is not ideal and also it is not possible to generate some of the annotations that the FitWolfe standardly uses, therefore it was needed to adjust it a little bit. The API is modelled according to REST architectural style and the request and responses both contain JSON values. On all statistics endpoints an HTTP GET request can be performed. Let us look at the individual endpoints for training statistics briefly.

3.5.1 One Rep Max statistics endpoint

The endpoint **/api/users/userId/statistics/oneRepMax** obtains calculated one rep max data for a specific exercise that the user with id *userId* (a number) performed. The request contains:

- Exercise description id
- Date filter object with start and end date
- Object for selecting a specific plan and/or a day in that plan
- Object with a minimum and maximum repetitions that the data regarding one rep max statistics should be limited to

- Object with a minimum and maximum effort that the data regarding one rep max statistics should be limited to

The response then contains a value of absolute one rep max and a collection of objects where each represent a point on a line graph. The attributes of the object are:

- The name of the training session
- The index of the training session, which is microcycle number minus one
- The date when the training session was performed, in order to pair it to the x axis of the line graph
- The value of one rep max, which represents a y axis value.
- The collection of exercise sets, with their data that was used to obtain the value of one rep max
- The best set index - the set that contains the best calculated value of one rep max

3.5.2 Scheduled vs Performed statistics endpoint

The endpoint **/api/users/userId/statistics/scheduledVsPerformed** obtains scheduled vs performed statistics on a specific exercise that the user with id *userId* (a number) had in their training plan. The request contains:

- Exercise description id
- Training variable type, which can be sets, reps, load or effort and specifies what should be compared when looking at the scheduled and performed data
- Date filter object with start and end date
- Object for selecting a specific plan and/or a day in that plan

The response then contains an object with these attributes:

- The percentage when less was performed than scheduled
- The percentage when more was performed than scheduled
- The percentage when the performed equaled to what was scheduled

3.5.3 Weekly Training Sessions Goals endpoints

The endpoint **/api/users/userId/goals/weeklyTrainingSessions** creates a new weekly training session goal when HTTP POST request is called onto it. The data that is needed to create a new goal are:

- The number of training sessions

- The starting date of the goal
- The ending date of the goal

Only one goal can be active in the same period, therefore on a possible duplicate the response returns response code 409 which signifies a conflict. Performing an HTTP PUT request onto the same endpoint updates the currently active goal if it exists. To obtain current and previous weekly training session goals one can perform HTTP GET request onto the same endpoint. The response body contains a collection of previous weekly training session goals and a current weekly training session goal if any. The objects contain the same attributes as are used when creating the goal plus an id of the goal.

3.5.4 Weekly Training Sessions statistics endpoint

The endpoint **/api/users/userId/statistics/weeklyTrainingSessions** obtains statistics about the training sessions per week that the user with id *userId* (a number) performed. The request contains:

- Date filter object with start and end date
- Compliance criterion, which either specifies that the goal number of training sessions per week can be the same as the number of performed training sessions, or that there can be more performed training sessions.

The response then contains compliance percentage, which specifies how often performed number of sessions was in compliance with the goal according to chosen compliance criterion. Additionally, the response also contains the collection of objects that have these attributes:

- The starting date of the week
- The ending date of the week
- The goal number of sessions in that current week
- The actual number of sessions in that current week

3.5.5 Training Session Effort endpoint

The endpoint **/api/users/userId/statistics/trainingSessionEffort** obtains statistics about the change of session effort over time for the user with id *userId* (a number). The request contains:

- Date filter object with start and end date
- Object for selecting a specific plan and/or a day in that plan

The response contains a collection of objects which have these attributes:

- The name of the training session

- The index of the training session, which is microcycle number minus one
- The date when the training session was performed, in order to pair it to the x axis of the line graph
- The average value of effort on exercises in the training session.

3.5.6 Average Microcycle Volume statistics endpoint

The endpoint **/api/users/userId/statistics/averageVolume** obtains statistics about the average muscle group volume per microcycle for the user with id *userId* (a number). The request contains:

- Date filter object with start and end date
- Object for selecting a specific plan and/or a day in that plan
- Object with a minimum and maximum effort that the data regarding one rep max statistics should be limited to
- The length of the microcycle in calendar days

The response then contains a collection of objects with these attributes:

- The muscle group data
- The number of exercise sets the muscle group acted as a primary mover
- The collection of names of exercises in which the muscle group acted as a primary mover.
- The number of exercise sets the muscle group acted as a secondary mover
- The collection of names of exercises in which the muscle group acted as a secondary mover.

Chapter 4

Development

Before the development of training statistics, it was feasible to adjust a few things. A new entity *user settings* was implemented. One of its attributes is a coach's choice to use either RPE or RIR for effort. Previously it was customizable on training plans and templates. This will be important for the Training Session Effort graph.

Next, the effort values were adjusted for training plans, so that users have to input effort between 6.5 and 10 RPE. The values below 6.5 RPE tend not to be very precise, since it is not very easy to estimate failure when it is very far. Furthermore, the calculations for one repetition maximum do not exist for less than 6.5 RPE, certainly because the estimate would not be as precise. More changes followed; however, it is not very important to mention them.

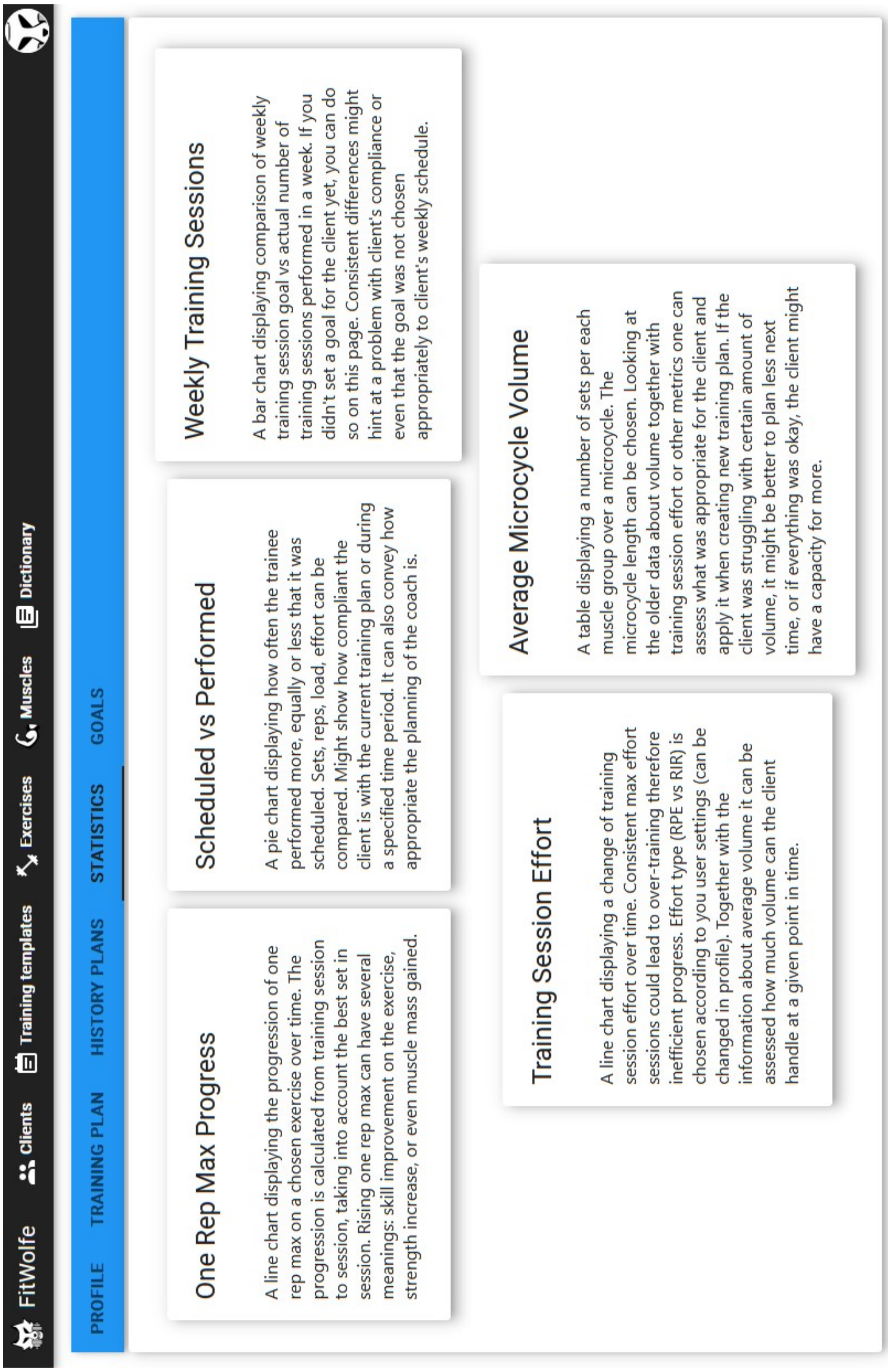
The development of training statistics continued in the same fashion as research, using prototypes. Following high fidelity prototype, the development started on web front end. The user interface was implemented for all of the statistics with the help of library Recharts. The graphs used an example static data, that was prepared to showcase their behavior. Then, it was briefly discussed with the personal trainers.

After some little adjustments the development continued on the back end side. The statistics were implemented one by one, together with the necessary changes to data model, mentioned in the Analysis chapter. After fine tuning the details, the statistic could then be presented to the coaches.

4.1 Statistics starting page

The statistics starting page is situated in the overview of the selected client. A coach can arrive there from their list of clients. As we can see in the following figure there are several tabs such as client profile or training plan. The statistics reside in their own tab.

The initial idea with the dropdown component, which can be seen in the figures in the research chapter, was replaced with cards. Some of the personal trainers mentioned during the consultations that it would be useful for them to have some descriptions of the graphs. Now each card has a short text that aims to summarize what the graph should display and by clicking on it, the page with the specific graph is shown.



One Rep Max Progress

A line chart displaying the progression of one rep max on a chosen exercise over time. The progression is calculated from training session to session, taking into account the best set in session. Rising one rep max can have several meanings: skill improvement on the exercise, strength increase, or even muscle mass gained.

Scheduled vs Performed

A pie chart displaying how often the trainee performed more, equally or less that it was scheduled. Sets, reps, load, effort can be compared. Might show how compliant the client is with the current training plan or during a specified time period. It can also convey how appropriate the planning of the coach is.

Weekly Training Sessions

A bar chart displaying comparison of weekly training session goal vs actual number of training sessions performed in a week. If you didn't set a goal for the client yet, you can do so on this page. Consistent differences might hint at a problem with client's compliance or even that the goal was not chosen appropriately to client's weekly schedule.

Training Session Effort

A line chart displaying a change of training session effort over time. Consistent max effort sessions could lead to over-training therefore inefficient progress. Effort type (RPE vs RIR) is chosen according to you user settings (can be changed in profile). Together with the information about average volume it can be assessed how much volume can the client handle at a given point in time.

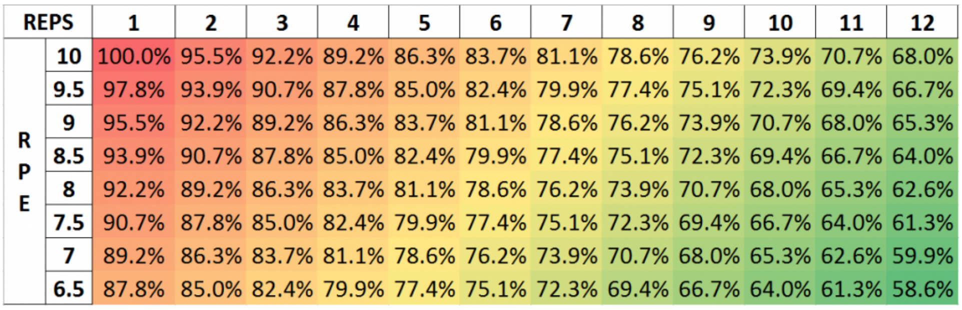
Average Microcycle Volume

A table displaying a number of sets per each muscle group over a microcycle. The microcycle length can be chosen. Looking at the older data about volume together with training session effort or other metrics one can assess what was appropriate for the client and apply it when creating new training plan. If the client was struggling with certain amount of volume, it might be better to plan less next time, or if everything was okay, the client might have a capacity for more.

Figure 4.1: The starting page for statistics [Author]

4.2 One Rep Max statistics

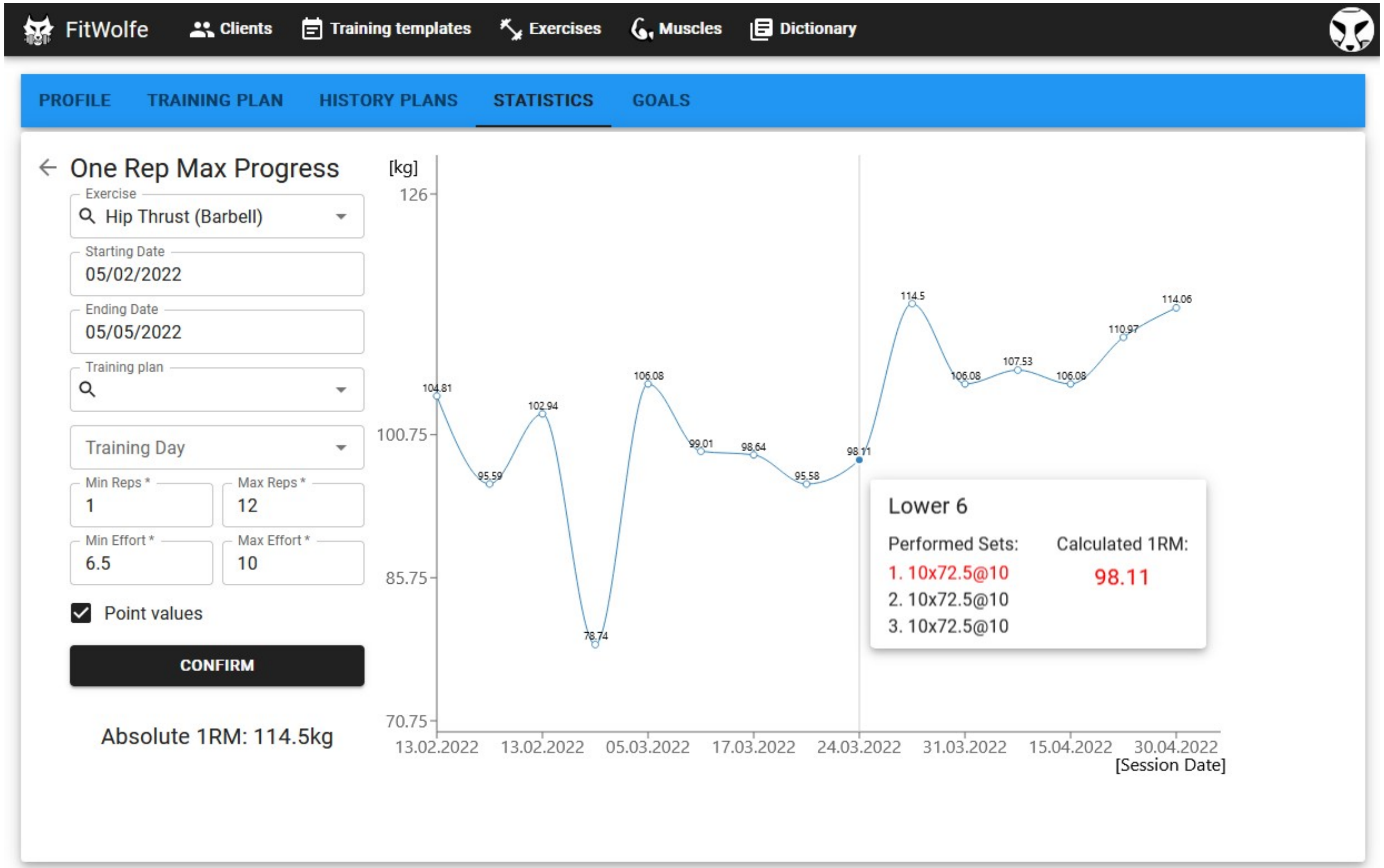
Once the request for one rep max comes for some exercise a client performed, the data is first filtered according to the input body and then the calculation can begin. For all of the relevant performed exercise sets one rep max is calculated using the table present in the figure 4.2. The best set is then taken into account, its repetitions, RPE and load. After that, the data is added to the output collection but also one rep max from the best set is propagated up to calculate absolute one rep max.



REPS		1	2	3	4	5	6	7	8	9	10	11	12
RPE	10	100.0%	95.5%	92.2%	89.2%	86.3%	83.7%	81.1%	78.6%	76.2%	73.9%	70.7%	68.0%
	9.5	97.8%	93.9%	90.7%	87.8%	85.0%	82.4%	79.9%	77.4%	75.1%	72.3%	69.4%	66.7%
	9	95.5%	92.2%	89.2%	86.3%	83.7%	81.1%	78.6%	76.2%	73.9%	70.7%	68.0%	65.3%
	8.5	93.9%	90.7%	87.8%	85.0%	82.4%	79.9%	77.4%	75.1%	72.3%	69.4%	66.7%	64.0%
	8	92.2%	89.2%	86.3%	83.7%	81.1%	78.6%	76.2%	73.9%	70.7%	68.0%	65.3%	62.6%
	7.5	90.7%	87.8%	85.0%	82.4%	79.9%	77.4%	75.1%	72.3%	69.4%	66.7%	64.0%	61.3%
	7	89.2%	86.3%	83.7%	81.1%	78.6%	76.2%	73.9%	70.7%	68.0%	65.3%	62.6%	59.9%
	6.5	87.8%	85.0%	82.4%	79.9%	77.4%	75.1%	72.3%	69.4%	66.7%	64.0%	61.3%	58.6%

Figure 4.2: 1RM RPE table by Mike Tuchscherer [43]

In the figure 4.3 we can see the page for one rep max statistic. On the left there is a form where the personal trainer can input the exercise that interests them and also choose some additional filters. On the right we can see a line graph. The values on the x axis are the dates of training sessions when the chosen exercise was performed and values on y axis are the calculated one rep maxes. Each graph point can be hovered over which displays a tooltip with additional information. In the figure, we can see the tooltip for the point that represents one rep max present in the session *Lower* (named after its training day name) in the sixth microcycle. It contains performed sets data from which the one rep max was calculated, where the best set is highlighted with red color.



One Rep Max Progress

Exercise: Hip Thrust (Barbell)

Starting Date: 05/02/2022

Ending Date: 05/05/2022

Training plan:

Training Day

Min Reps *: 1

Max Reps *: 12

Min Effort *: 6.5

Max Effort *: 10

☑ Point values

Absolute 1RM: 114.5kg

Lower 6

Performed Sets:
1. 10x72.5@10
2. 10x72.5@10
3. 10x72.5@10

Calculated 1RM: 98.11

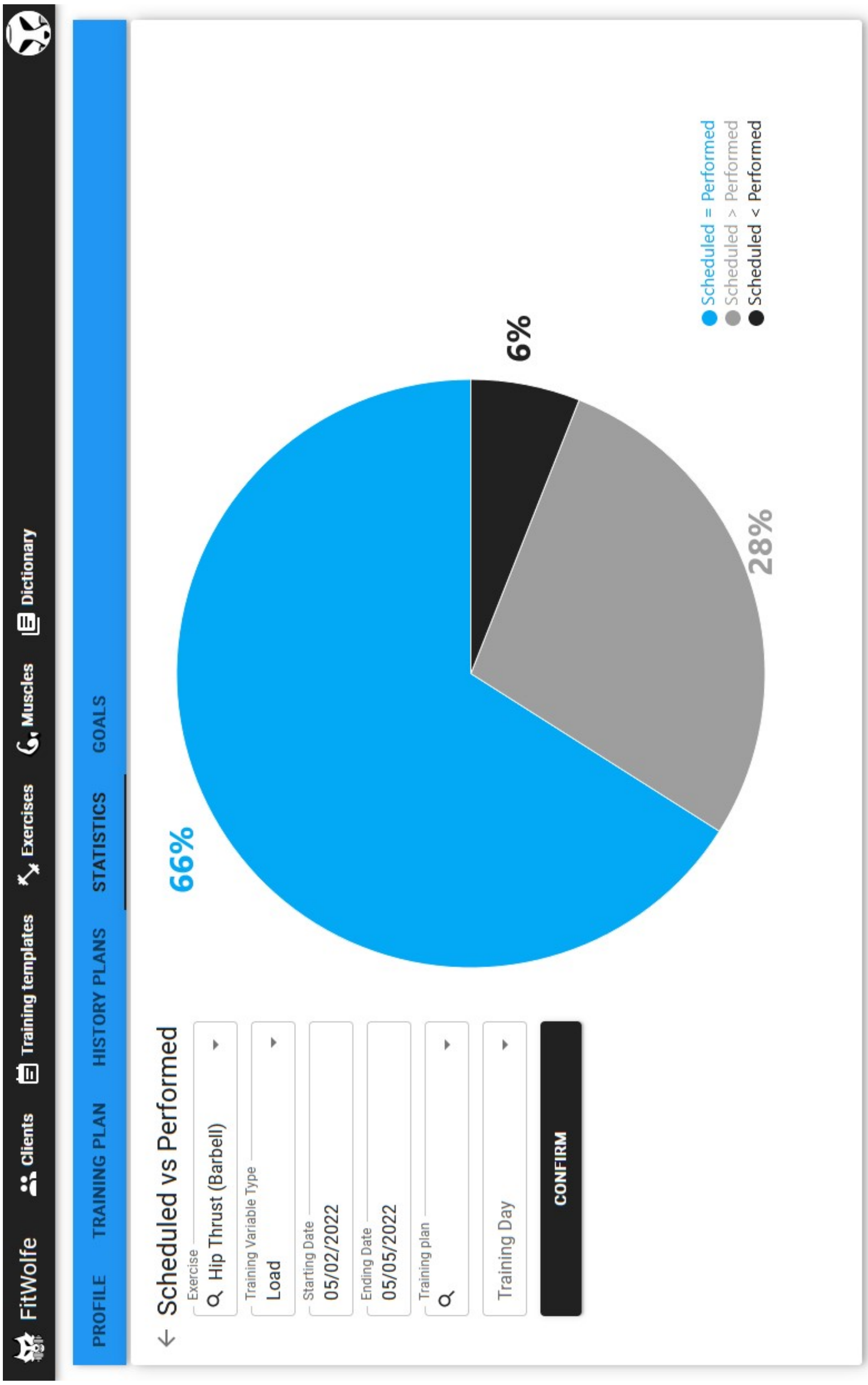
Figure 4.3: 1RM Statistic graph [Author]

4.3 Scheduled vs Performed statistics

The incoming request for scheduled vs performed statistics is handled by first taking all training exercise pairs that satisfy the request filters. Next, if the user chose training variable *sets* the training exercise pairs are compared and if their exercise descriptions match (Barbell Squat is the exercise that was scheduled and also performed) then the number of sets is compared. The result will count this exercise pair into one of three categories: scheduled was more than performed, they were the same or scheduled was less than performed. If the performed exercise was different or missing, then it is counted in the category *scheduled was more than performed*.

Training variables reps, load and effort are treated a bit differently, they are only compared on matching sets, if scheduled exercise has more sets than performed, then those extra sets are not taken into account. Similarly, if there are more performed sets than scheduled. The performed and scheduled exercise has to be the same of course. Each compared set is counted into one of three previously mentioned categories.

In the figure 4.4 user interface for scheduled vs performed statistics is displayed. Similarly, as in the previous statistic, the form with filters is present on the left and the graph is on the right. The graph is a pie chart, with three sections that represent results of comparisons between scheduled and performed training variable data.



Scheduled vs Performed

Exercise: Hip Thrust (Barbell)

Training Variable Type: Load

Starting Date: 05/02/2022

Ending Date: 05/05/2022

Training plan:

Training Day

Figure 4.4: Scheduled vs Performed Statistic graph [Author]

4.4 Weekly Training Sessions statistics

Before we go into details about statistics, it is important to look at the creation of weekly training session goals. The first idea was to create training goals next to the graph, however, some of the personal coaches suggested that it would be better to treat the statistics tab in the detail of the selected client as *read-only*. Therefore, a new tab was added called *Goals* which in future might hold other goals, such as macronutrient numbers, body measurements, body weight etc.

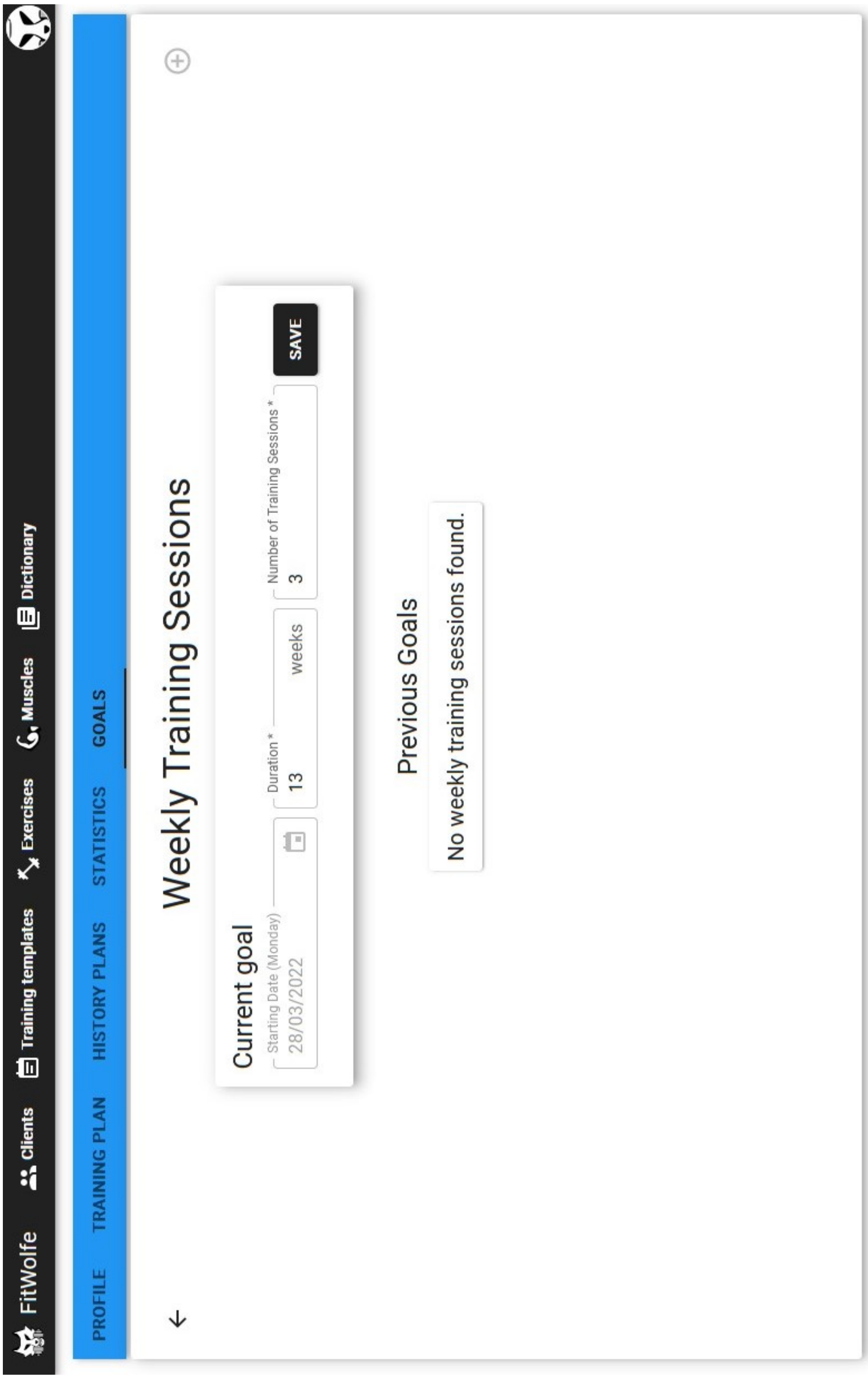
The figure 4.5 presents a user interface for creating and displaying weekly training session goals. The user can add a new goal by the plus sign on the right if no goal is currently active. Then a new card is displayed with a *starting date* (always Monday), *duration in calendar weeks* and a *number of training sessions per week*. Once the goal is created, the starting date cannot be modified, and duration can be set to a minimum number of weeks from the start date to the Sunday of the current week. This business logic was chosen, because it eliminates plenty of complicated states the data could be in, but also it is quite close to the real use case.

The data that actually travels to the back end is slightly different. There is no variable *number of weeks*, but instead the end date is calculated from start date and the number of weeks and then both the start date and the end date are sent. The start date is adjusted to be Monday in the same week as the inputted start date and the end date is adjusted to be Sunday in the same week as the inputted end date. The number of training sessions is better on front end from validation point of view, the end date on back end allows for more efficient calculation of statistics.

Now that we know how the user can create goals, let us look at the statistics. The figure 4.6 shows the input form together with the graph. The main difference between other forms is the field compliance criterion. There, the coach can choose whether he allows the client to perform more training sessions than the goal defines, or it has to be exact. This then reflects the calculated client compliance present under the form. In our case, the client did not perform more training sessions than specified by the goal, so the compliance is the same for both criteria.

While hovering on a section of graph a pair of bars that belong together are highlighted and a tooltip displaying selected week start and end date is shown, paired with the actual number of training sessions and goal number of training sessions at that point in time.

The implementation logic for obtaining the statistics is a bit complicated due to the nature of input and output data. First a collection of goals and a collection of session dates that comply to the filters in the request are obtained. Then the calendar weeks are iterated from the start date in the request to the end date in the request. The collection of goals and collection of sessions are also being iterated depending on how they align to the current calendar week. The goal number of sessions is taken, but also the session dates that belong to the current week are counted, this number is compared each week-iteration and accumulated into the client compliance metric.



Weekly Training Sessions

Current goal

Starting Date (Monday) 28/03/2022

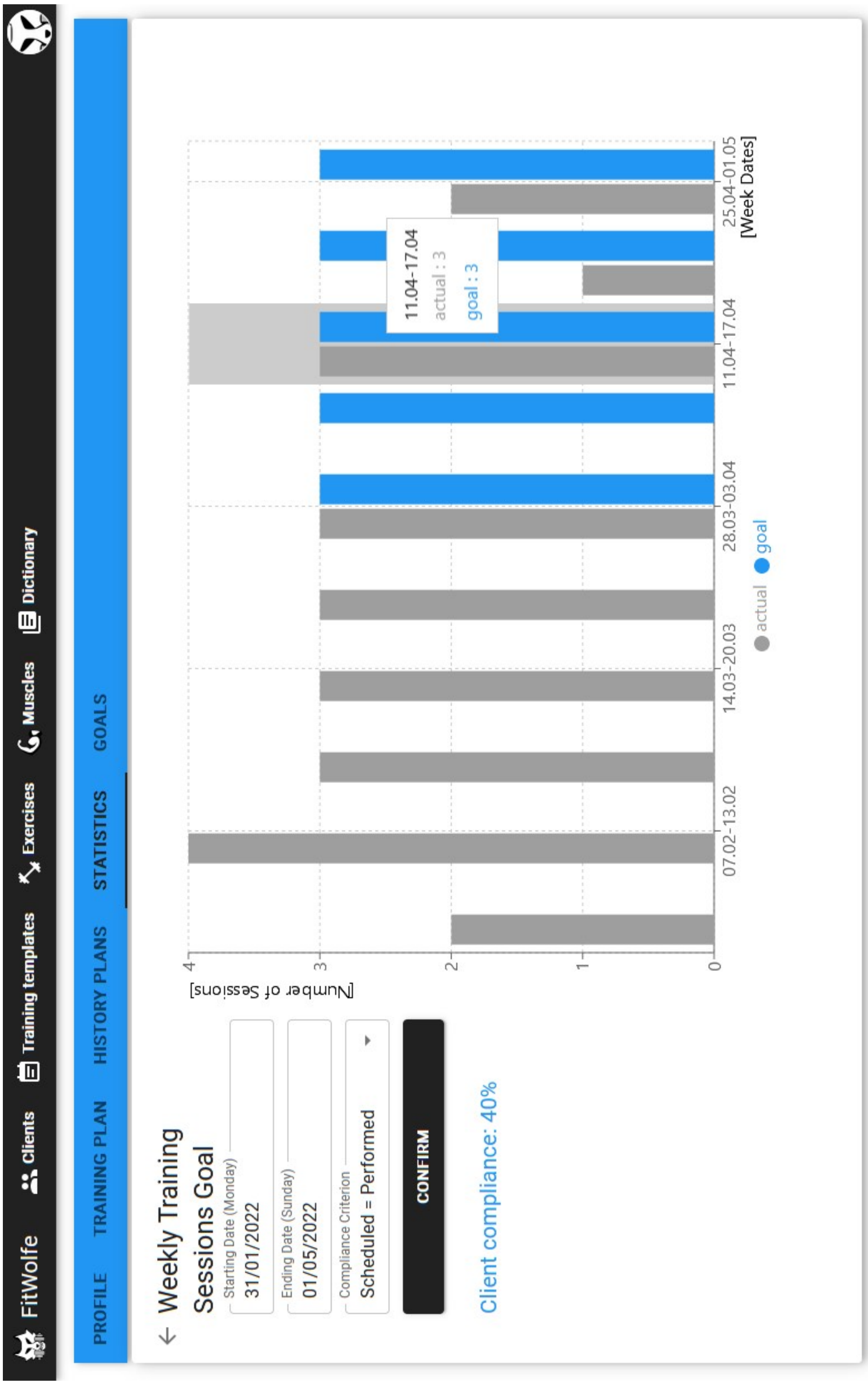
Duration * 13 weeks

Number of Training Sessions * 3

Previous Goals

No weekly training sessions found.

Figure 4.5: Weekly training sessions goals page [Author]



← Weekly Training Sessions Goal

Starting Date (Monday): 31/01/2022

Ending Date (Sunday): 01/05/2022

Compliance Criterion: Scheduled = Performed

CONFIRM

Client compliance: 40%

Figure 4.6: Weekly training sessions statistics page [Author]

4.5 Training Session Effort

The graph for this statistic is a plain line chart as can be seen in the figure 4.7. Each point represents a training session, its x value is the session date and y value is the average effort of the training session. The tooltip displayed when hovering on a specific point shows the name of the training session (or rather the name of the training day it belongs in) and its microcycle number, together with the calculated effort. The unit of effort (RPE/RIR) is chosen depending on the user preferences that they can define in their profile.

The implementation of this statistic is also the most straight forward, the input filters from the request are propagated into the select query from database where the average effort is calculated and then returned as the output.

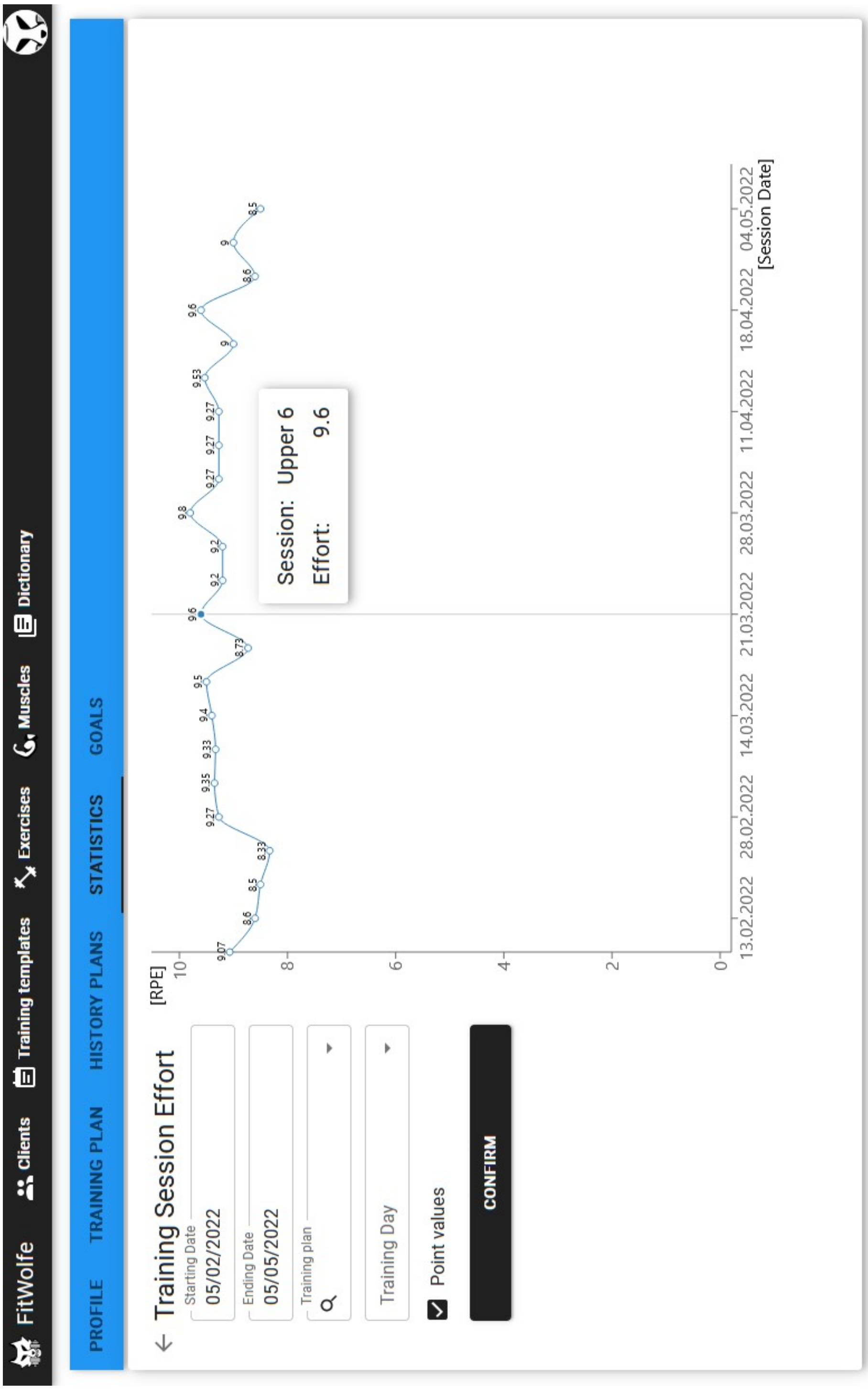
4.6 Average Microcycle Volume statistics

As previously mentioned, in order to obtain average microcycle volume statistics, it was necessary to adjust the model of the exercise descriptions. The single collection of trained muscle groups was split into two, *primary muscle groups* and *secondary muscle groups*. Then it was necessary to correct the data for all the exercises so that the resulting statistics would show relevant data. Besides this change, each muscle group has a new attribute *rank*, which forces a certain display order in the table on the user interface.

Compared to other training statistics, this single one is displayed in a table, which best suits its nature. The figure 4.8 displays the input form and the resulting table. The form contains one field that is different from those in previous statistics. It is *days in microcycle* which is a number of training sessions in the microcycle together with rest days, so all calendar days in a microcycle. If the personal trainer would choose a training plan, theoretically it could hold the information about the microcycle length, however, it would require more complicated adjustments of the model and additionally, it is usually the case that the microcycle "shifts over time as a trainee might miss some training sessions. The current business logic is a rough estimate of data over the selected period of time.

The cells of the table contain a name of the muscle group. Next to the name, there is a number of sets it acted as a primary mover in black color and a secondary mover in blue color. The number of sets is taken as an average over the selected period of time per a microcycle of specified length. When a user hovers over the number of sets, a tooltip is displayed where the names of exercises that correspond to it are written.

When a request is performed onto the back end, the initial step is to obtain training sessions that satisfy the given filters. Once that is done, an iteration over microcycles begins from the starting date to the ending date, depending on the microcycle length. All sessions that belong to that microcycle by their performed date are included in the next processing - all performed exercises



Training Session Effort

Starting Date: 05/02/2022

Ending Date: 05/05/2022

Training plan

Training Day

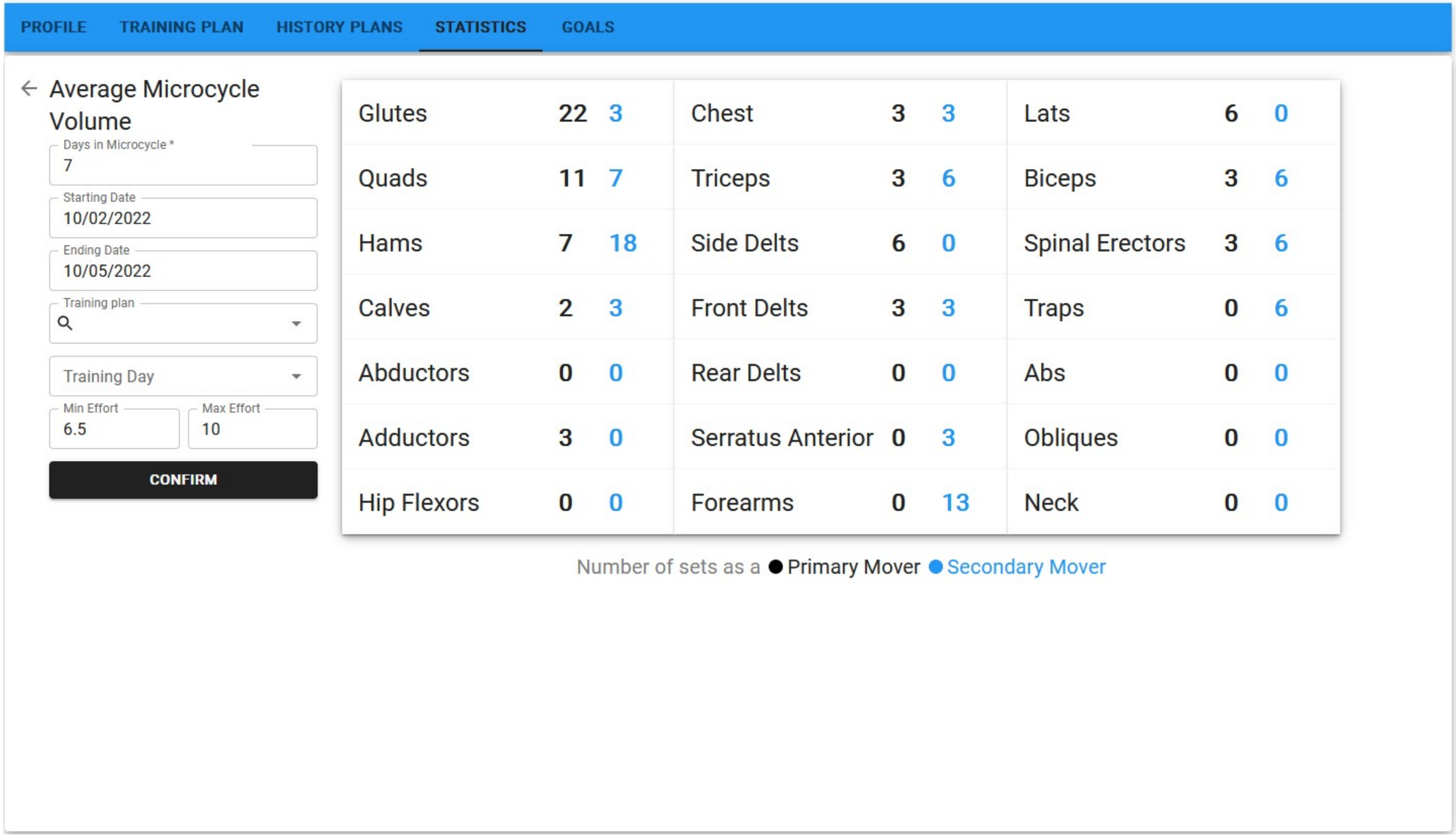
☑ Point values

Session: Upper 6

Effort: 9.6

Figure 4.7: Training session effort statistics page [Author]

are checked for their primary and secondary muscle groups and for each of them the number of sets that was performed is incremented. When all the iterations are finished, finally the averages are calculated.



Average Microcycle Volume

Days in Microcycle * 7

Starting Date 10/02/2022

Ending Date 10/05/2022

Training plan

Training Day

Min Effort 6.5

Max Effort 10

Glutes	22	3	Chest	3	3	Lats	6	0
Quads	11	7	Triceps	3	6	Biceps	3	6
Hams	7	18	Side Delts	6	0	Spinal Erectors	3	6
Calves	2	3	Front Delts	3	3	Traps	0	6
Abductors	0	0	Rear Delts	0	0	Abs	0	0
Adductors	3	0	Serratus Anterior	0	3	Obliques	0	0
Hip Flexors	0	0	Forearms	0	13	Neck	0	0

Number of sets as a ● Primary Mover ● Secondary Mover

Figure 4.8: Average Microcycle Volume statistics page [Author]

Chapter 5

Quality Assurance and evaluation

5.1 Unit testing

Unit testing is the lowest level of quality assurance that can be on an application. It may not discover all the defects that can be present in the system, but it usually catches those that come from the developers' oversight. Since they test only the units, which are pieces of system that can be isolated from each other, they can miss some business errors, because the business use case usually flows through several units.

Well-written unit tests still provide an enormous value, as they are repeatable and very cheap, thanks to their very fast execution, which is usually performed automatically on continuous integration/continuous delivery (CICD) pipeline. CICD is an automated process of building the new changes to the code, running unit tests on it and finally deploying the new code in production. [44]

The unit tests on the back end of FitWolfe ran approximately 5.5 minutes before the implementation of statistics. The class coverage which is the percentage of classes that were included in the unit tests was 79%. The total lines of back end code were covered from 58%. During the implementation of statistics, besides the necessary adjustments of code, there were also a few changes to decrease the development debt. This included also the improvements in unit tests. After the implementation of statistics, the current run time of unit tests is just 3 seconds. The class and code coverage did not change, as all new code was covered by unit tests.

5.2 Factory Acceptance Testing

The factory acceptance tests, are the tests that the quality assurance specialists from the development team execute manually. I chose ad-hoc testing strategy for this project, because of two reasons. More in-depth testing would require a lot of time which was unfortunately very precious after the development. The second reason was that I wanted to pay more attention to user acceptance testing with personal coaches. Some of the test cases for factory acceptance testing are described below. Abbreviation TC stands for test case.

5.2.1 Testing one rep max statistics

TC1 - 1RM - No data available

Situation: Choose an exercise that the trainee did not execute in their training plan.
Expected result: Message *No data available* is displayed.

TC2 - 1RM - Data displayed correctly

Situation: Choose an exercise that the trainee performed.
Expected result: Line graph is displayed, hovering with mouse over a graph point displays more information. Absolute one rep max is displayed next to the graph.

TC3 - 1RM - Training day specific data is displayed

Situation: Choose an exercise that is included in two training days. Choose one of the days.
Expected result: Line graph is displayed, there are less results than when no training day is selected in the form.

5.2.2 Testing Scheduled vs performed statistics

TC4 - SVP - No data available

Situation: Choose an exercise that the trainee did not execute in their training plan.
Expected result: Message *No data available* is displayed.

TC5 - SVP - Data displayed correctly

Situation: Choose an exercise that the trainee performed.
Expected result: Pie graph is displayed with one to three sections depending on the data. The graph has a legend which describes what the sections mean. Next to each section there is a percentage.

5.2.3 Weekly training session goals

TC6 - G - Create a goal

Situation: Click on the create button, fill in the data and click save.
Expected result: The data is saved, on reload the goal is displayed as currently active.

TC7 - G - Update goal

Situation: Update duration and number of training sessions.
Expected result: Starting date is disabled and cannot be changed. Other data is saved, on reload the new data is displayed on the goal.

5.2.4 Weekly training session statistics

TC8 - WTS - Data displayed correctly

Preconditions: The user has some training sessions finished. There is an active weekly training sessions goal.
Situation: Choose a date range that includes currently active goal and performed training sessions dates. Choose compliance criterion *scheduled = performed*.
Expected result: Bar graph is displayed. For each week there is at least one bar displayed. If the trainee performed some sessions in some of displayed weeks, then there is a bar that shows number of training sessions in that week. If a training goal was active during that week, then a bar of different color shows *goal* number of training sessions.

5.2.5 Training session effort statistics

TC9 - TSE - Data displayed correctly

Situation: Choose a date range that includes dates when some training sessions were performed.
Expected result: Line graph is displayed. While hovering over some point a tooltip is displayed with the name of the training session, its microcycle number and calculated average effort for that training session.

5.2.6 Average volume statistics

TC10 - AV - Data displayed correctly

Situation: Choose a date range that includes dates when some training sessions were performed.
Expected result: A table with muscle groups is displayed. Next to each muscle group there are two numbers, average number of sets the muscle acted as a primary mover and average number of sets the muscle acted as a secondary mover. While hovering over the number of sets, the names of exercises are displayed.

5.3 User Acceptance Testing

User acceptance tests were performed during online consultations with several personal trainers. I have one client that has been using the application to track training sessions for over nine weeks. This time range was good enough to test the implemented training statistics in full effect. The screenshots that were presented in the Development chapter contained some of the mentioned data. During each consultation I shared my screen where I presented the final user interface and discussed the intuitiveness, ability to interpret the data, issues that were present and potential new features.

A quick feature that was implemented is *point values on line graphs.* The feature is already present and can be visible on screenshots of one rep max statistic and training session effort statistic. There was also one other issue where muscle groups in the average microcycle volume table were not ordered correctly, this was fixed by correcting the data in the database and also can be seen on a corresponding screenshot. The other requests would require more implementation time therefore, they are postponed to the next development of the application.

Chapter 6
Conclusion

6.1 Summary

Resistance training is a process that requires a lot of thought. Personal trainers have to take into account various factors when they write training programs for their clients. The clients might strive to increase their strength, build muscle, lose fat or just train for health benefits and pleasure. Training programs will vary depending on these goals, since each goal is better attainable with different combination of training volume, intensity and frequency. Great amount of work that the people do during their training session generates a lot of fatigue which has to be taken into account together with other life stressors. This has to be handled with better distribution of work in order to avoid injury and to ensure effective progress.

Consultations with personal coaches were performed in order to identify metrics that would help them see the bigger picture in how their clients are progressing. These metrics were to be added into the existing web application Fitwolfe that allows creation of training programs and recording of performed training sessions. Prototyping approach was used, and the user interface was designed in several steps in order to be intuitive and provide the needed information to the coaches. These changes were then implemented into the web application itself and consulted with the trainers.

6.2 Evaluation

This practical part of this project required a lot of work, and it touched all aspects of the software development life cycle. Previously, I have experimented with the prototyping approach only a few times, mostly during my university courses, but it was not as close to the "real world" as during this project. I have discovered that the consultations with the personal coaches were hard to manage since many of them have tightly packed schedules and some of them could not participate even if they wanted to.

The project influenced all components of the system, back end and both front ends even though the mobile front end only experienced adjustments since training statistics during this project were intended mainly for coaches.

Some of the clients that are more experienced could also use the graphs, however, there are other features that should be implemented first, these will be mentioned in the next section.

The total time spent on the project including writing the thesis was approximately 300 hours, which was much more manageable than the bachelor thesis with more than 400 hours. Personally, I view that the hours that were put into the master thesis project have more quality in them. Bachelor thesis included the implementation of a huge number of features and building the core of the application. Current project was focused more on working with the users, identifying their needs and overall development process had clearer structure and resulted in much better code quality.

Personal coaches that were present on the final consultation viewed the implemented features quite positively. The main praises were for the simplicity and design of the user interface. Not all of them were as experienced as some of the concepts that the statistics portray, but after some explanation it was cleared out. They only suggested to add more descriptions and tooltips so they would catch on more quickly when using it again.

The web application is available now and in the near future at `https://fitwolfe.siliconhill.cz/`. The latest build of android application is available at `https://expo.dev/accounts/bulkodav/projects/fitwolfemobile/builds/9a3ba540-e6f2-43ee-80b8-d79af1aa27ba`. When opening the url in the browser on an android device a download should start. To be able to install the application, one has to allow the installs of apk, which can be done following steps on this page: `https://www.expressvpn.com/support/vpn-setup/enable-apk-installs-android/`. After that, it is necessary to go to the folder where the app was downloaded and simply tap on it to start the installation process. FitWolfe mobile should then be ready to use. It is required, however, to register on the web page and confirm the account before being able to log in to the mobile application.

The part of the code that was created or changed is available in the attachment file *code.zip*.

6.3 Future

The application FitWolfe now likely contains all integral things that the personal coaches and clients need for planning and recording resistance training. The most important features that are to be implemented next are planning for individuals - those that do not have personal trainers and being able to use FitWolfe for recording training on IOS devices. Since licences for mobile apps would be expensive at this state of the system, implementing the same features as are in React Native app on web front end seems like the most feasible option.

Regarding improvements on training statistics, some simple adjustments of the user interface will be implemented most likely during summer, however, there are some bigger features that were mentioned. Adding personal notes to graphs is one of them, at this point it seems difficult, because it would require persistence of the graphs. Changing the concept of weekly training sessions goals and statistics to microcycle is another feature that might be beneficial, as the length of the microcycle is not always a week and some of the personal coaches would rather view it in that way. Because there are many exercises and not all were trained by the client for whom we are generating statistics, it might also help to only include the present ones in forms on statistics pages. All of these features are not integral; therefore they might be implemented later.

It is apparent that the possibilities on how to enhance this application are almost endless and they could help the personal trainers provide more value to their clients through their services. This drives me to work on FitWolfe in the future and help make it happen.

Bibliography

[1] FitWolfe

`https://fitwolfe.siliconhill.cz` [cit. 07.01.2022]

[2] Bulko, David Michal. (2020). *MyOnlineTrainer,* Bachelor's thesis, Czech Technical University, Prague.

[3] Stojiljkovic, Nenad & Ignjatovic, Aleksandar & Savić, Zvezdan & Marković, Živorad & Milanović, Sandra. (2013). *History of resistance training.* Activities in Physical Education and Sport. Vol 3. Pages 135-8.

[4] Helms, Eric & Cronin, John & Storey, Adam & Zourdos, Michael. (2016). *Application of the Repetitions in Reserve-Based Rating of Perceived Exertion Scale for Resistance Training.* Strength and Conditioning Journal. Vol 38. Pages 42-9.

[5] General RPE table

`http://in-move.cz/rpearir/` [cit. 07.01.2022]

[6] Chilibeck, P.D., et al., *A comparison of strength and muscle mass increases during resistance training in young women.* Eur J Appl Physiol Occup Physiol, 1998. Vol. 77. Pages 170–5.

[7] Seynnes, O.R., M. de Boer, and M.V. Narici, *Early skeletal muscle hypertrophy and architectural changes in response to high-intensity resistance training.* J Appl Physiol, 2007. Vol. 102. Pages 368–73.

[8] Fry, A.C., *The role of resistance exercise intensity on muscle fiber adaptations.* Sports Med, 2004. Vol. 34. Pages 663–79.

[9] Richens, B. and D.J. Cleather, *The relationship between the number of repetitions performed at given intensities is different in endurance and strength-trained athletes.* Biol Sport, 2014. Vol. 31. Pages 157–161.

[10] One repetition maximum table

`https://barbend.com/build-your-1-rep-max-calculator/` [cit. 07.01.2022]

[11] Schoenfeld, B.J., et al.,. *Differential effects of heavy versus moderate loads on measures of strength and hypertrophy in resistance-trained men.* J Sports Sci Med, 2016. Vol. 15. Page 715.

[12] Schoenfeld, B.J., et al., *Effects of Low- Versus High-Load Resistance Training on Muscle Strength and Hypertrophy in Well-Trained Men.* J Strength Cond Res, 2015. Vol. 29. Pages 2954-63.

[13] Schoenfeld, B.J., et al., *Ifluence of Resistance Training Frequency on Muscular Adaptations in Well-Trained Men.* J Strength Cond Res, 2015. Vol 29. Pages 1821–9.

[14] The Muscle & Strength Pyramid - Training (p. 57) by E. Helms, A. Morgan and A. Valdez, 2018, Copyright 2018 by Eric Helms.

[15] Screenshot from the results of card sorting study

`https://app.optimalworkshop.com` [cit. 07.01.2022]

[16] Microsoft Whiteboard

`https://www.microsoft.com/en-us/microsoft-365/microsoft-whiteboard/digital-whiteboard-app`

[17] Screenshot from Figma

`https://www.figma.com` [cit. 07.01.2022]

[18] Springboot `https://spring.io/projects/spring-boot` [cit. 03.05.2022]

[19] Maven `https://maven.apache.org/` [cit. 03.05.2022]

[20] Junit framework `https://junit.org/junit5/` [cit. 03.05.2022]

[21] Hyper SQL database `https://hsqldb.org/` [cit. 03.05.2022]

[22] PostgreSQL database `https://www.postgresql.org/` [cit. 03.05.2022]

[23] Flyway migration tool `https://flywaydb.org/` [cit. 03.05.2022]

[24] Lombok `https://projectlombok.org/` [cit. 03.05.2022]

[25] Springdoc Openapi `https://springdoc.org/` [cit. 04.05.2022]

[26] Sentry `https://sentry.io/welcome/` [cit. 03.05.2022]

[27] React `https://reactjs.org/` [cit. 03.05.2022]

[28] TypeScript `https://www.typescriptlang.org/` [cit. 03.05.2022]

[29] Redux `https://redux.js.org/` [cit. 03.05.2022]

[30] React Router `https://reactrouter.com/` [cit. 03.05.2022]

[31] Material UI `https://mui.com/` [cit. 03.05.2022]

[32] React Native `https://reactnative.dev/` [cit. 03.05.2022]

[33] React Native Elements `https://reactnativeelements.com/` [cit. 03.05.2022]

[34] React Native Paper `https://reactnativepaper.com/` [cit. 03.05.2022]

[35] React Navigation `https://reactnavigation.org/` [cit. 03.05.2022]

[36] Expo Dev `https://expo.dev/` [cit. 03.05.2022]

[37] React charts libraries `https://technostacks.com/blog/react-chart-libraries` [cit. 04.05.2022]

[38] Dev extreme reactive library `https://devexpress.github.io/devextreme-reactive/react/chart/` [cit. 11.05.2022]

[39] Recharts library `https://recharts.org/en-US/` [cit. 04.05.2022]

[40] Data Driven Documents `https://d3js.org/` [cit. 04.05.2022]

[41] Swagger Editor `https://editor.swagger.io/` [cit. 04.05.2022]

[42] REST `https://restfulapi.net/` [cit. 04.05.2022]

[43] 1RM RPE table `https://articles.reactivetrainingsystems.com/2015/11/29/beginning-rts/` [cit. 05.05.2022]

[44] Continous Integration, Continous Delivery `https://www.synopsys.com/glossary/what-is-cicd.html` [cit. 10.05.2022]