

Bachelor Project



**Czech
Technical
University
in Prague**

F3

**Faculty of Electrical Engineering
Department of Mathematics**

MyOnlineTrainer

Dávid Michal Bulko

Supervisor: Ing. Pavel Náplava, Ph.D.

Field of study: Software engineering and technologies

May 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Bulko** Jméno: **Dávid Michal** Osobní číslo: **474582**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

MyOnlineTrainer

Název bakalářské práce anglicky:

MyOnlineTrainer

Pokyny pro vypracování:

Vytvořte aplikaci pro podporu komunikace mezi fitness trenéry a jejich klienty. Základem aplikace bude možnost tvorby tréninkových plánů různými trenéry a jejich následné sdílení klientům. Klienti budou moci během jednotlivých tréninků sledovat plán a na konci zaznamenat svůj výkon. Trenér následně na základě zaznamenaných výkonů plán přizpůsobí. Na začátku tvorby aplikace proveďte analýzu potřeb a vyhodnoťte již existující aplikace, včetně zdůvodnění, zda a proč má smysl tvořit novou aplikaci. Pro ověření funkčnosti aplikace použijte uživatelské testování na vybrané skupině fitness trenérů a jejich klientů.

Seznam doporučené literatury:

- [1] Helms E., Morgan A., Valdez A. The muscle & strength pyramid – training. Second edition. Retrieved from <https://muscleandstrengthpyramids.com/>
- [2] <https://www.baeldung.com/>
- [3] <https://spring.io/>
- [4] <https://reactjs.org/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Náplava, Ph.D., katedra ekonomiky, manažerství a humanitních věd FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **11.02.2020**

Termín odevzdání bakalářské práce: **22.05.2020**

Platnost zadání bakalářské práce: **30.09.2021**

Ing. Pavel Náplava, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Acknowledgements

I would like to thank primarily my supervisor Pavel Náplava for guiding me through the process of creation of the thesis, as well as for providing me with valuable feedback especially on the written part. Another huge thanks goes to my colleague Daniel Groschup, who helped me on the mobile frontend side. I could not do it without Tomáš Biganič, Jan Zeman, Angela Bulková. They directed me to a better and more sensible written part of the thesis. Other honorable mentions go to Ondřej Mareš, David Stražovan who gave me great insights when I was stuck during implementation or when I wanted to improve something. 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 Bachelor'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 2020.

Abstract

The purpose of this thesis was to look at the process of creation and planning of training programs by fitness coaches. Is there any room for improvement? A present-day trend of digitalization proposed an idea: a dedicated software. The current state of the market for such tools was briefly researched and consulted with several personal trainers. Concluding the findings, a design for a new system was created, which was then developed and put to the test by coaches to see whether it would help them speed up and improve the management of training plans of their clients.

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

Supervisor: Ing. Pavel Náplava, Ph.D.
ČVUT FEL,
Technická 2,
166 27 Praha 6

Abstrakt

Cielom tejto práce bolo identifikovať proces vytvárania a plánovania tréningového programu fitness trénermi. Existuje spôsob ako to zlepšiť? Dnešný trend digitalizácie ponúka jednu myšlienku: dedikovaný softvér. Súčasný stav trhu softvérových nástrojov bol podrobený krátkej rešerši a konzultovaný s niekoľkými osobnými trénermi. Na základe zistení sa vytvoril návrh systému, ktorý bol následne vyvinutý a otestovaný trénermi s cieľom zistiť, či im pomôže urýchliť a zlepšiť správu tréningových plánov ich klientov.

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

Contents

1 Introduction	1	5.2 Client functionalities	35
1.1 Motivation	1	6 Conclusion	36
1.2 Issue description	1	Bibliography	37
1.3 History of Resistance Training	3		
2 Research of the present state of training planning	5		
2.1 Process of research	5		
2.2 Process of training planning	5		
2.3 Currently used planning tools	7		
2.4 Search results	7		
2.5 Tool classification	8		
2.5.1 Sample plan tools	8		
2.5.2 Individual planning tools	8		
2.5.3 Text-based planning tools	10		
2.5.4 Specialized tools for personal coaches	12		
2.6 Research Conclusion	14		
3 Software analysis	15		
3.1 Overview	15		
3.2 List of user roles	15		
3.2.1 Coach	15		
3.2.2 Client	16		
3.2.3 Administrator	16		
3.3 User stories	16		
3.3.1 User	16		
3.3.2 Administrator	16		
3.3.3 Coach	17		
3.3.4 Client	18		
3.4 Use case diagrams	19		
3.5 Technologies	20		
3.5.1 Requirements	20		
3.5.2 Assessment	21		
3.5.3 Backend	21		
3.5.4 Frontend	22		
3.5.5 Database	23		
3.5.6 Deployment	23		
3.5.7 System architecture	25		
3.5.8 Data design	27		
4 Development	30		
4.1 Backend	30		
4.2 Web Frontend	31		
4.3 Deployment	32		
4.4 Mobile Frontend	33		
5 Testing in Production	34		
5.1 Coach functionalities	34		

Figures

Tables

1.1 Diskolobos of Myron [8]	3
2.1 Sample training day [<i>author</i>]	6
2.2 Rate of perceived exertion scale. [10]	6
2.3 Freetrainers logo [12]	8
2.4 Fitify workouts logo [13]	8
2.5 Gym workout plan for weight training logo [14]	8
2.6 Strong logo [15]	9
2.7 Fitnotes logo [16]	9
2.8 JeFit logo [17]	10
2.9 Excel logo [18]	11
2.10 Google sheets logo [19]	11
2.11 True coach logo [20]	12
2.12 Fitr logo [21]	13
2.13 Trainerize logo [22]	14
3.1 Coach use case diagram [<i>author</i>]	19
3.2 Client use case diagram [<i>author</i>]	20
3.3 Admin use case diagram [<i>author</i>]	20
3.4 Current frontend technology usage [25]	22
3.5 Deployment diagram [<i>author</i>] ..	24
3.6 Architecture diagram [<i>author</i>] ..	25
3.7 Domain model [<i>author</i>]	27
3.8 Class model [<i>author</i>]	28
4.1 Training exercises page [<i>author</i>]	32
4.2 Training exercises page [<i>author</i>]	33

Chapter 1

Introduction

1.1 Motivation

My motivation for choosing this topic was that I am a big fitness enthusiast and a personal coach on the side. Some of my friends and acquaintances are also personal coaches. Our collective experience of training planning and the fact that I study software engineering gave me an idea to implement an application that would help coaches automate and maybe even enhance the performance of some tasks they have to do every day. Especially training planning. Before the system is implemented, the requirements of the coaches should be gathered, but also a research of currently used tools should be performed. The application should contain the fundamental features that are needed for training planning. It should also be tested by the coaches after it is deployed and assessed if it is feasible to develop further.

1.2 Issue description

Us humans have evolved over millions of years because of our superior intellect and our ability to adapt. Our body developed to endure traveling over long distances on foot. It was likely to allow us either to forage for food when it was scarce or to chase after prey which could easily escape over short distances but lacked the stamina over the long ones. However, our rapid progress might have brought us new problems to deal with.

In this era, as our technology surpassed our physical abilities, we have no need to rely on them anymore. We have means of transportation that are faster than ever, food is very easily available to us in restaurants and robots and other machinery do the manual work for us. Our body, which evolved to be under almost constant activity, now does not get enough of it. The muscles which do not get enough stimulus become atrophied and their range of motion shortens. This might lead to acute or even chronic injuries. The lack of activity might not be the sole problem.

The current abundance of food in the majority of world countries might be the other one. Because the improvement of technology and with it easier access to food happened over a relatively short time span, we could not adapt

to it neither biologically nor habitually. Since we have been struggling with a deficit of available food for millions of years, our bodies are not used to the current surplus and in many cases, this leads to overeating. If this spans over longer periods obesity develops and with it, more health risks might spring up. Many people choose to seek out the help of personal trainers who help them to improve their diet and fitness. The latter is usually done by resistance training.

Resistance training can be defined as increasing muscle strength, size, and endurance through contraction of muscles against external resistance such as body weight, machinery, load or similar. Other than improving body composition [3, 4, 5], there have been observed many other benefits for people who perform resistance training. These include improving cardiovascular health [3, 5, 6] by reducing blood pressure, LDL cholesterol, triglycerides and increasing HDL cholesterol. Furthermore, there have been cases of maintenance or even increase in bone mineral density [3, 4, 5] and enhanced insulin sensitivity [3, 5]. In one paper [6], there was an observation of decreased worker absenteeism.

As the body gradually adapts to a certain stimulus, muscle and strength growth slows down or even stops altogether. In order to continue progressing, it is necessary to increase the stimulus incrementally and plan it over time. Optimally it should take into account the process of physical regeneration and also overall stress put onto an individual. This is one of the things that personal coaches specialize in. Before the process of planning is assessed, it might be important to look at the history of resistance training.

1.3 History of Resistance Training

There is no clear indication when the resistance training exactly started in its true form, but humanity has been aware of its need for a long time. One of the first civilizations to embrace exercise were Greeks and Persians. We have more historical evidence of the former. Many Greek works of art such as statues (figure 1.1), pottery, and literature depicting athletic bodies with broad shoulders and slim waists can be found in the archeological record.

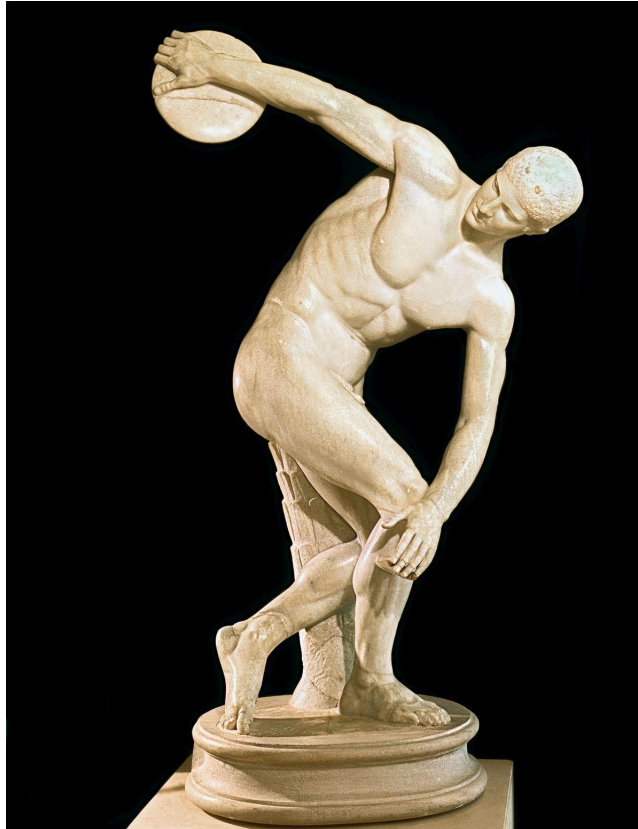


Figure 1.1: Diskobolos of Myron [8]

In Ancient Greece, physical fitness was not ingrained in culture solely for aesthetic purposes, but it played a huge role in their military strength as well. The exercise was an integral part of the conditioning of soldiers. Greeks even had buildings dedicated to training their body called "gymnasiums" (from which the current nomenclature gym originates) and "palestra" where they prepared not only for military purposes but also for the Olympic games.

Another example of a civilization, where the training was very important, is the Roman Empire. A renowned physician of that era called Galen wrote "Preservation of Health" where he stressed the importance of weight training.

As time went on, the message which ancient civilizations established got lost. This was probably due to a shift of thinking in Europe. With the rising popularity of Christianity, people started putting much more emphasis on

Chapter 2

Research of the present state of training planning

2.1 Process of research

The research of available training-planning tools was done as follows: first, a number of coaches were interviewed about their approach to the training-planning process and what data they strive to gather. Then, their preferences for current training-planning tools were collected. Finally, the market for tools was analyzed. It consisted of searching for key phrases on the page google.com and in Google Play store android application. The phrases were: “planning tools for gym coaches”, “training planner for gym coaches” in the former and “training planning”, “gym training planning”, “workout planner”, “workout planning” in the latter.

Some of the apps were searched for specifically by their names through the suggestions from the coaches. Next, the apps were classified into a few categories based on their main focus and the type of targeted user. According to that, the advantages and disadvantages of each tool were determined. The results were only limited to a maximum of 3 most relevant apps in each category.

2.2 Process of training planning

In order to assess the validity of available training-planning tools for personal coaches, it is necessary to understand the process of training planning and the needs of coaches. Based on the initial data which the coach collected from the client, the focus of the plan, its duration and division of the plan to cycles is determined. The smallest cycle is usually a week which consists of several training days. Each training day consists of a number of exercises that are done in a timely or repetitive fashion, possibly with some load (below we can see a sample training day created by me).

Lower body	Sets	Repetitions	Load (kg)	RPE
Squat	3	6	100	8
Hip thrusts	3	8	80	8
Lunges	3	8	30	8
Hyperextensions	3	8	body weight	8
Standing calves	4	8	50	8

Figure 2.1: Sample training day [author]

What is often done, notably by coaches with more experience or those who base their work on scientific research, is that maximum effort for a given exercise is also tracked. The standard method for identifying that is the rating of perceived exertion or RPE. A table explaining RPE for exercises using repetitions is listed below. It is very similar to the original Borg CR10 scale for measuring exertion [11] which is used for aerobic exercises where we usually quantify the time performed rather than repetitions. This gives a more subjective opinion on perceived exertion compared to when it is quantifiable by repetitions.

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 2.2: Rate of perceived exertion scale. [10]

It is expected that the plan incrementally changes, typically by increasing load. This is usually done on a weekly basis as the client progresses. The progress is assessed based on the inputs provided by the client, or the coach if they train together. In any case, sharing the plan between the coach and the client allows for quicker changes and more transparency about achieved performance.

2.3 Currently used planning tools

A total of 10 coaches were asked about their experience with training planning and the tools used for this purpose. Most coaches use the Excel sheets to plan their workouts and email, WhatsApp or Messenger for more direct communication with their clients. The majority of them have not tried any other tool. Some did not like that certain apps are too overwhelming with their functionalities from the start compared to simple spreadsheets. One particular coach reported that the apps he tried tracked too much information and it made his clients neurotic.

The most used app – Excel – provides coaches with a good overview of the whole training and usually one training week can be compared to the other by mere glance. They are used to this tool and like it's simplicity. However, as said above the disadvantage is that more writing must be done, statistics have to be set up manually and it is not easily shared with the clients. The problem with sharing can be eliminated by using the Google Sheets tool, which is the second most popular app used. The rest of them uses mostly text-based tools, such as basic pen and paper or Microsoft Word. Five of the coaches are totally satisfied with this approach and would not change it. Five of them would be interested in trying out a new tool if it helped them create training plans or do other tasks more effectively.

The most looked for functionalities are:

1. ability to track sets, reps, load,
2. quick and easy modification of the training plan,
3. generating progression or adherence statistics,
4. ability to share the plan directly with a client,
5. notifications about certain actions.

2.4 Search results

When searching for viable options existing on the market several tools were found, out of which a total of 8 were chosen based on their popularity among coaches and relevancy in the search engines (Freetraines, Fitify workouts, Gym workout plan for weight training, Strong, JeFit, FitNotes, Microsoft excel, Google sheets, Truecoach, Fitr, Trainerize). Based on their main focus I was able to divide them into a few categories and rate the relevancy and satisfaction of wanted features.

2.5 Tool classification

The tools were divided into four categories. In each category there are three most relevant apps. For each app, their name and logo is included. A link to the app is listed in the bibliography with a reference number under the logo.

2.5.1 Sample plan tools

These tools provide already set plans which a person can follow and most of the time without an ability to modify them. Some of them allow slight customization such as generating a plan depending on some information the person provides about themselves or choosing a particular exercise variation from a group that trains similar movement patterns. Since these tools are made for individuals without personal trainers, we will not list their potential advantages and disadvantages.



Figure 2.3: Freetrainers logo [12]

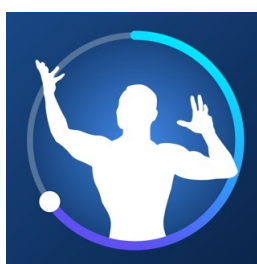


Figure 2.4: Fitify workouts logo [13]

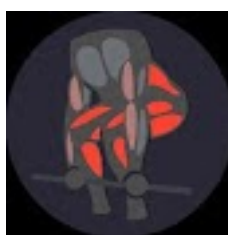


Figure 2.5: Gym workout plan for weight training logo [14]

2.5.2 Individual planning tools

These tools allow an individual to create and adjust training by himself, which could be more useful for the clients of personal coaches to track their workouts. However, it would be necessary for them to copy the data that the coach provided to assemble the workouts first. After each training, the clients

would need to report the data back to the coach. That could be complicated if the app does not allow export of data in a usable format for the coach. The advantages and disadvantages in this category are mainly from the client's point of view and were put together only by my personal observations.

■ Strong



Figure 2.6: Strong logo [15]

Advantages:

- It contains an extendable database of exercises.
- Native tracking for sets, repetitions, rest pauses, RPE.
- Ability to add notes to individual exercises and training days.
- Available offline.
- Provides statistics about training sessions and individual exercises.

Disadvantages:

- Unshareable with the coach directly, unless a screenshot is sent, which might not always contain all the exercises
- Sometimes, the application has lag spikes.
- Additional modification of newly added exercise is limited.

■ Fitnotes



Figure 2.7: Fitnotes logo [16]

Advantages:

- It contains an extendable database of exercises.

- Native tracking for sets, repetitions, rest pauses.
- Ability to add notes to individual exercises and training days.
- Available offline.
- Provides statistics about training sessions and individual exercises.
- Ability to export workout in a textual form.

Disadvantages:

- Unshareable with the coach directly for modification.

■ **JeFit**



Figure 2.8: JeFit logo [17]

Advantages:

- It contains an extendable database of exercises.
- Native tracking for sets, repetitions, rest pauses.
- Available offline.
- Provides statistics about training sessions and individual exercises.

Disadvantages:

- Unshareable with the coach directly, unless a screenshot is sent, which might not always contain all the exercises.

■ **2.5.3 Text-based planning tools**

This category contains tools that display mainly text data, usually in a format of a table. The advantages and disadvantages are based on feedback from coaches, but also my personal observations.

■ **Excel**

This application is a standard tool for the coaches and the majority of them use it.



Figure 2.9: Excel logo [18]

Advantages:

- A good overview of the whole training plan.
- Possibility to create charts, statistical calculations. (The majority of coaches do not use this)
- Many people know this app and are able to use it quite promptly.
- Available offline.

Disadvantages:

- A lot of manual writing.
- Unshareable with the client directly, unless the file or a screenshot is sent.

Excel provides overall 3 out of 5 most looked for functionalities by the coaches.

■ **Google sheets**



Figure 2.10: Google sheets logo [19]

Advantages:

- A good overview of the whole training plan.
- Possibility to create charts, statistical calculations. (The majority of coaches do not use this)
- Many people know this app and are able to use it quite promptly.
- Shareable with the client through a web link.

Disadvantages:

- A lot of manual writing.
- Limited functionality compared to excel.

Google sheets technically provides all of the most looked for functionalities by the coaches, but it is up for a debate. Firstly, the plan can be quickly modified, except when a certain exercise needs to be changed. It might "break" some statistics that the coach prepared and it might not be easy for him to display the change very well visually. Secondly, the progression statistics are quite difficult to set up, so many coaches do not even bother to create them. Lastly, some of the coaches reported that email notifications do not always work.

■ Paper

Advantages:

- Simple.
- offline.

Disadvantages:

- A lot of manual writing.
- No statistics, unless manual calculations are done by coach.
- Unshareable with the client directly, unless a copy is sent.

Paper provides overall 1 out of 5 most looked for functionalities by the coaches.

■ 2.5.4 Specialized tools for personal coaches

The applications in this category are build specifically for personal trainers to help them with their tasks. Some can contain functionalities that are not tied only to training planning. The advantages and disadvantages are based on feedback from coaches, but also my personal observations.

■ TrueCoach



Figure 2.11: True coach logo [20]

Advantages:

- Ability to create templates for training plans which can be assigned to clients and customized later.

- Videos for the exercises in the database.
- Communication through built-in chat, comments on training sessions.
- Provides email notifications on certain actions in the app.
- Logged training sessions are shared both with the client and the coach.
- Support for tracking body measurements.
- Ability to schedule training on specific days.
- Customizable app with a coach's own brand.
- Compliance statistics for every client.

Disadvantages:

- The tracking of sets, reps, rest pauses, RPE is just textual. Statistics about workouts nor specific exercises cannot be generated.

TrueCoach provides overall 3 out of 5 most looked for functionalities by the coaches.

■ **Fitr**



Figure 2.12: Fitr logo [21]

Advantages:

- Ability to schedule training on specific days.
- Built-in chat.
- Notifications in the app and through email.
- Ability to create templates for the training sessions.
- Custom videos can be added to the training plan to show the execution of exercises.
- Logged training sessions are shared both with the client and the coach.

Disadvantages:

- Very unintuitive from the start.
- Seemingly no native support for tracking sets, reps, rest pauses, RPE (just textual).
- It cannot generate any statistical data.

Fitr provides overall 3 out of 5 most looked for functionalities by the coaches.

■ Trainerize



Figure 2.13: Trainerize logo [22]

Advantages:

- Ability to create templates for training plans which can be assigned to clients and customized later.
- Videos for the exercises in the database.
- Communication through built-in chat.
- Logged training sessions are shared both with the client and the coach.
- Support for tracking body measurements.
- Ability to schedule training on specific days.
- Customizable app with a coach's own brand.
- Quick editing as in spreadsheets.

Disadvantages:

- Too many functionalities/overwhelming from the start.

Trainerize provides overall 4 out of 5 most looked for functionalities by the coaches.

■ 2.6 Research Conclusion

The best possible candidates among those analyzed in this research are TrueCoach and Trainerize. They contain the most features the coaches look for when planning training programs and compared to Google Sheets they provide additional features. Nevertheless they still have some disadvantages. TrueCoach is missing a possibility to generate statistics about a client's performance in training, for example how many kilograms of weight has a client added on some exercise. Trainerize is on the other hand quite overwhelming with functionalities compared to spreadsheets that the coaches are used to. If the user interface was simplified, users could navigate through the app better.

If TrueCoach and Trainerize improved their shortcomings, they could be a main choice for the coaches on how to improve their training planning. Since these apps are not that well-known by coaches, there still might be a chance for a new app to be developed that would compete with them by building more on top of the coach's needs

Chapter 3

Software analysis

3.1 Overview

The goal of the practical part of this thesis is to create a solid base for a training-planning application. The tool should contain fundamental features necessary for the coaches, which are: creating a plan that can track sets, reps, load and effort on a training exercise, quick modification of the training plan, ability to share the plan directly with a client and finally there should be a possibility for the clients to easily share their training sessions with the coach. Furthermore, it should be easily extendable, so that more functionalities could be added in the future to cover all the needs of the coaches and maybe even compete with existing software on the market. In the following section we will look at the user roles in the system and what they should be available to do in the system.

3.2 List of user roles

3.2.1 Coach

Coaches in the new application should be able to manage their list of clients: add and remove a client. They should be able to view basic information about their clients and create a training plan for them. There should be an option to add and remove training days in the training plan. It is necessary that the exercise can be added into each training day with the possibility to specify the number of sets, repetitions and load for the exercises.

After the client completes the training session, their coach should have an ability to view how they did and then be able to adjust their training accordingly. If the coach decided that it is time for a new plan, there should be a possibility to discard the old one so that a new one can be assigned. To make this process quicker a coach should have a list of training templates that he can “copy” for their clients and then adjust if necessary. This coach should be able to fill this list of templates either directly, or by copying an already existing plan of some of his clients.

■ 3.3.3 Coach

[CO1] As a coach, I want to manage the list of my clients, so that it is up to date with my current business and I can find out information about my clients. This should consist of:

- viewing the list of my clients.
- searching in the list by first and last name.
- viewing the profile of an individual client.
- inviting a new client to my list by email.
- removing a client from my list.

[CO2] As a coach, I want to manage training plans for a client on my list, so that they know what to train and make progress. This should consist of:

- creating a training plan and setting a name for it.
- discarding a client's training plan.
- adding a training day to a plan.
- removing a training day from a plan.

[CO3] As a coach, I want to create and manage training day for my client, so that he knows what to train and progress. This should consist of:

- creating a training day and set a name for it.
- choosing an exercise from the database and adding it into the training day.
- specifying sets, repetitions and load for the exercise and duration. There are exercises that have either sets, reps, load or sets, load, duration, or sets, duration.
- deleting the exercise from the training day

[CO4] As a coach, I want to create and manage templates for training plans, so that I do not have to create them for scratch for every client. This should consist of:

- viewing the list of my training templates.
- creating a new template.
- deleting a template.
- editing a template.
- adding/replacing a client's training plan with a template and being able to adjust it.

- being able to see the time I spent. I want to finish the training session.
- discarding a running training session.

[CL5] As a client, I want to be able to see the history of my training sessions, so that I can see how I am progressing.

[CL6] As a client, I want the app to look well on the phone so that I can easily use it when I am in the gym.

3.4 Use case diagrams

Below we can see use case diagrams, one for each role. In the model they are interconnected, although they were cut so they could be displayed correctly in the document.

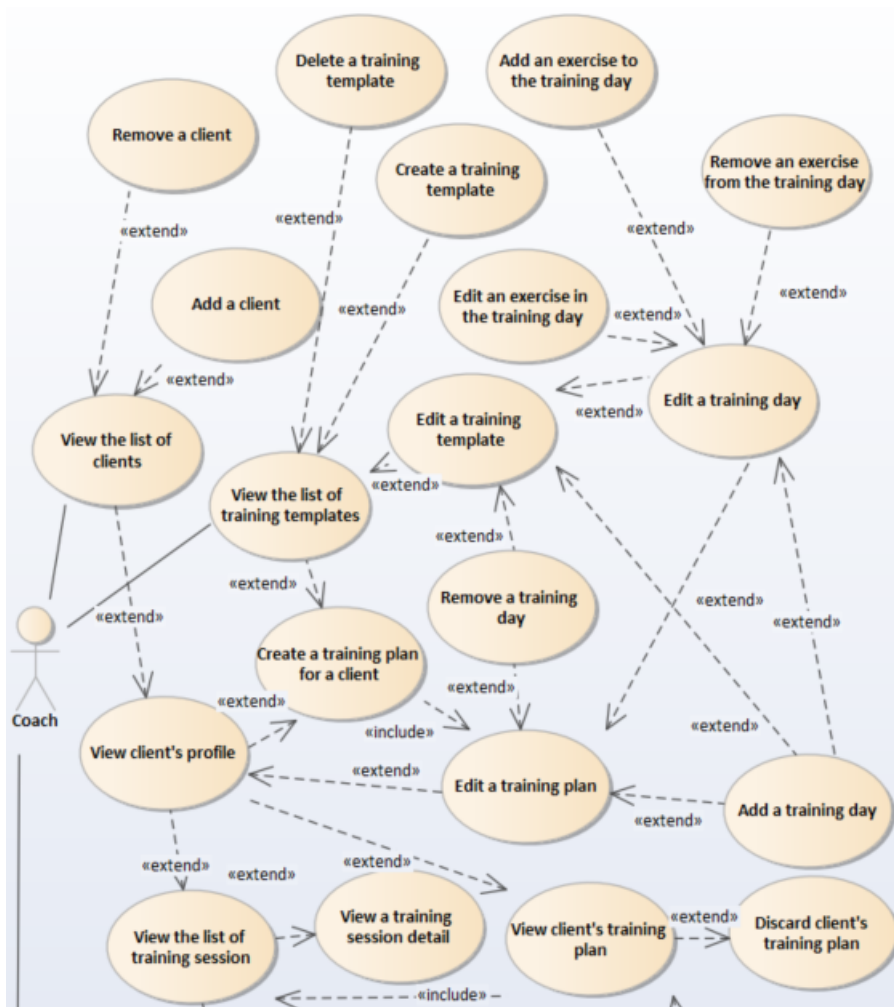


Figure 3.1: Coach use case diagram [author]

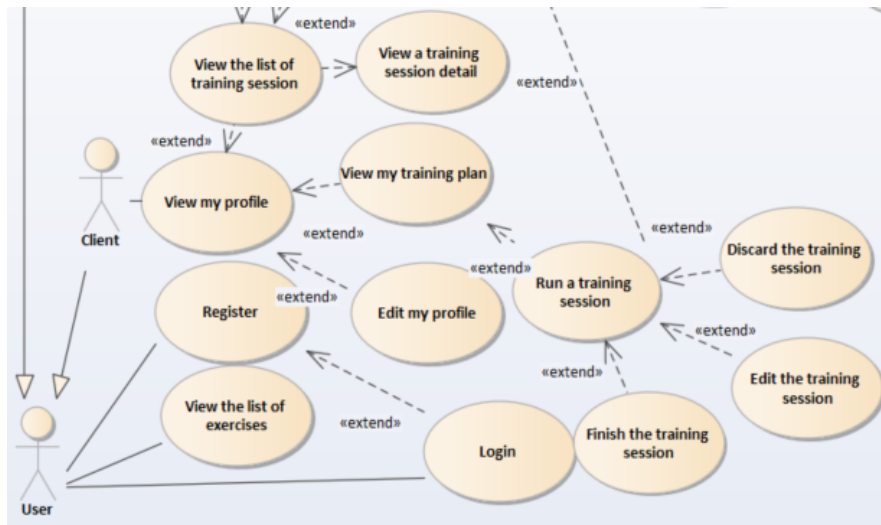


Figure 3.2: Client use case diagram [author]

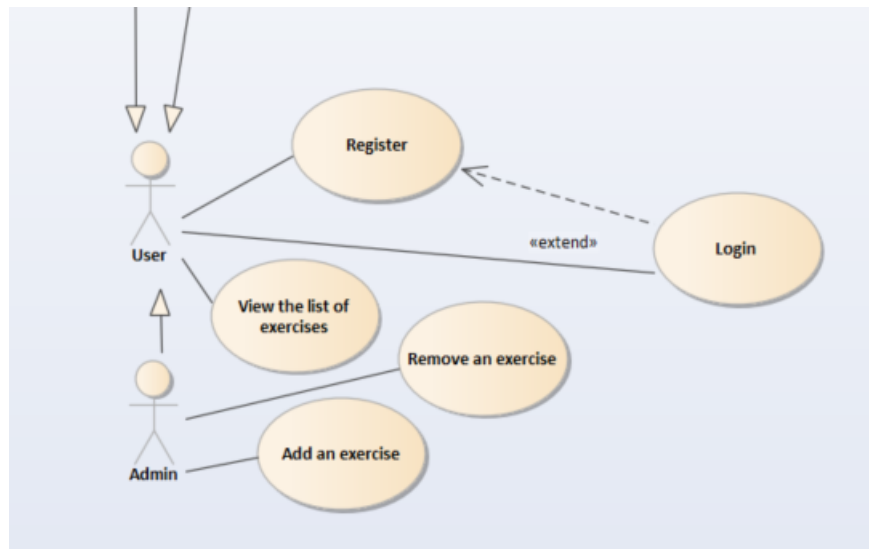


Figure 3.3: Admin use case diagram [author]

3.5 Technologies

3.5.1 Requirements

The platform on which the system should be available depends on the type of user. The clients will probably use the system mainly on their smartphones, since they are the most portable devices and can be easily used inside the gym while training. Also they are very popular. On the other hand, when they are at home, they might want to view their training history more comfortably on a bigger screen for example a desktop computer or a tablet. The coaches use mainly desktop computers for managing training plans, since it is the

easiest to edit them with a physical keyboard. When they train directly with their clients, they might want to rather use portable devices such as tablets or smartphones.

■ 3.5.2 Assessment

From the above requirements, it is clear that the system should be available on several different platforms whether desktop computers or portable devices. The actions performed in the application will probably not require a lot of computing power. What might thought, are the requests on the data. It might not be the case during this project, since it is only a proof of concept. Anyway, the data must be shared between the users, so at least the database needs to be deployed on a server. Considering all this information, it makes most sense to develop a web application. In web applications, the most important components are backend, frontend and a database.

■ 3.5.3 Backend

The most popular programming languages currently and the ones which I have encountered previously are Python, Java, Javascript and PHP. Let's look at the latter 3 languages rated by William Hurley and Bernhard Støcker on quora. [23]

1. Java

Pros Relatively fast, many libraries and frameworks, ubiquitous, type-safe, strong support for OOP.

Cons A lot of boilerplate code, run-time environments tend to be resource heavy.

2. Javascript

Pros Frontend and Backend can be written in the same language, ubiquitous.

Cons Poor library management, dynamically typed.

3. PHP

Pros Easy to deploy, many libraries, ubiquitous.

Cons Not scalable, poor code design, relatively slow.

Considering the advantages of Java and my experience with it, I am choosing this as the backend language of the system. The next question is a framework. Some of the frequently used are JSP, Grails and Spring. Since I have some experience with Spring, from previous years at university, I decided to go with it. Spring is a very good framework for implementing Representational state transfer (REST). It is an architectural style which allows separation of frontend and backend. The latter provides endpoints which a former can

query using HTTP protocol to obtain or manipulate the data. REST is still probably the standard for web applications. GraphQL has emerged as a new alternative, although I do not have any experience with it. Keeping in mind the scope of the project, it is probably easier to lean towards technologies known to me, to be able to deliver the outputs on time.

3.5.4 Frontend

The languages for frontend are quite straightforward. HTML, CSS, Javascript. There is a possibility to write it just plainly or use some framework to ease up the work. Currently the biggest frameworks are React, Vue, Angular and AngularJS.

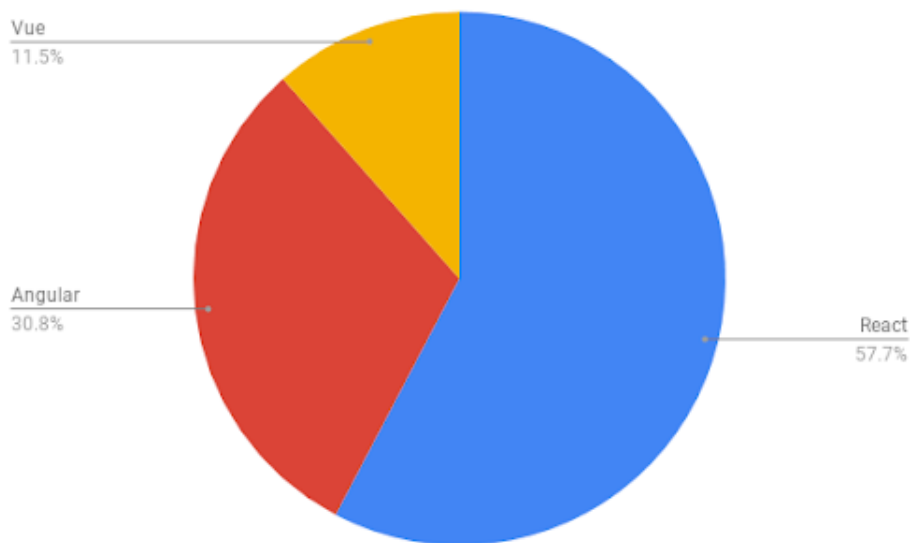


Figure 3.4: Current frontend technology usage [25]

Let's look at the pros and cons of the former three frameworks languages (also seen on a graph above by their popularity) according to the site dev.to [24].

1. React

Technically not a framework but a library.

Pros Maintained by facebook, evolving community, high performance.

Cons Not SEO friendly, more libraries needed to build complex applications.

2. Vue

Pros Growing popularity, easy to learn, high performance, typescript support.

Cons Too much flexibility might be problematic, not maintained by a big company.

3. Angular

Pros Strong community, 2-way data binding, typescript.

Cons Complex, hard to learn.

4. AngularJS

Pros 2-way data binding, fast development, highly testable.

Cons Old version, hard to learn, unpopular.

My personal experience includes React and Vue. Even though Vue is easier to learn, there is more material about React (tutorials, documentation etc.) on the internet, therefore it might be faster to develop in React. SEO (Search engine optimization), or faster accessibility to new users in search engines such as Google, might be a problem in React. However, it is not yet necessary for this application, but later if the development continues, it can be overcome by using different practices. Therefore I consider React as probably the best option for this project.

3.5.5 Database

There are mainly two types of databases nowadays. Relational SQL and non-relational No-SQL. Although non-relational databases might be good for new applications in which the data model might frequently change, they do not provide as much integrity for the data as relational ones. The data in this app will have a certain structure, which a No-SQL database does not consider. Paired with the fact that I do not have any experience with non-relational databases from previous semesters at the university, the relational database might be more suitable. We have many options for a database, some of them are Oracle XE, MySQL, PostgreSQL and DB2. PostgreSQL and MySQL are the one I am familiar with, but I cannot objectively evaluate which one would be a better fit. I had chosen PostgreSQL.

3.5.6 Deployment

Requirements

The last thing we need to consider is where the system should be deployed. During development it can run on my local machine, although after it is finished, it should be accessible to a few personal coaches, so that they can test it as potential future users. The server on which the app will be deployed should support all the technologies which are going to be used in the application. Because this project is a proof of concept, the best options should be free or reasonably cheap.

Options

1. **Heroku** This platform offers two viable options for our project. A free plan for hosting an application up to 550 hours of running time per month extendable to 1000 running hours when linking a credit card (the runtime can be divided into more apps). However, there is a 30 second sleep time after there was no request made to our application for 30 minutes, which might worsen user experience. A second option is 7\$ per month with a constantly running server. I have tried a free option in a previous project.
2. **Microsoft Azure** Azure offers a free 12 month trial for a 750-hour virtual machine running time 250 GB of SQL database and many other features. This should be enough “air-time” after the app is developed.
3. **Amazon Web Server** The free offer is similar to that of Microsoft Azure.
4. **Virtual Machine at Silicon Hill** Student’s organisation Silicon Hill at my dormitory offers virtual machines for the organisation’s active members. Technically I could deploy my application there as long as I am an active member.

The most feasible option is the fourth one, therefore I decided to choose it.

Deployment Diagram

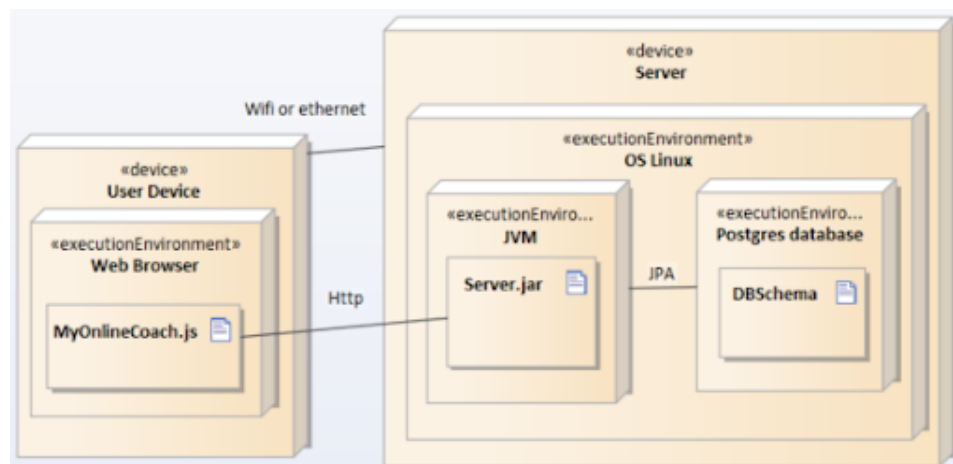


Figure 3.5: Deployment diagram [author]

Above we can see a deployment diagram of the system. The backend will run on a server with Linux operation system. It will access Postgres database using JPA (Java persistence API), which will also run on the server. The frontend application that the user accesses will run in the device’s browser and communicate with backend via HTTP protocol.

■ 3.5.7 System architecture

■ Description

The application will be built with Layered architecture. Each layer will have a specific role containing a number of components. A diagram of Layered architecture can be seen down below

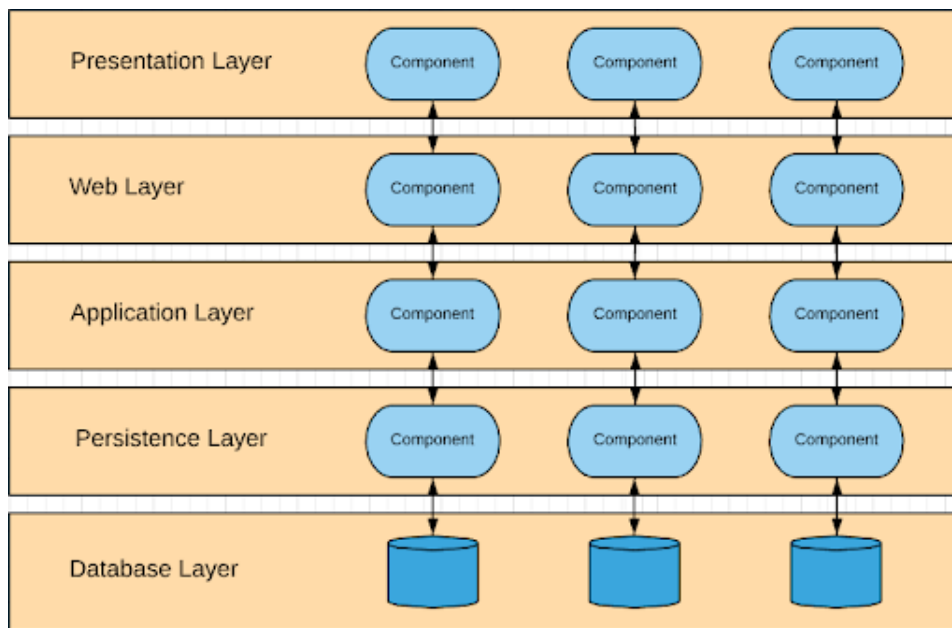


Figure 3.6: Architecture diagram [author]

■ Presentation layer

The presentation layer will be written using the Javascript library React. Its main function will be to display data from the backend and help the user to interact with the application. It will communicate with the web layer.

■ Web layer

This layer will contain servlets and components for handling client's requests and returning responses. In it, all the exceptions thrown in the will be handled. The API will be RESTful When handling requests. Web layer communicates with the application layer.

■ Application layer

Application layer is basically the core of the whole application. It will contain services in which there will be entire logic and all functionalities. Authorization will also be handled here.

■ **Persistence layer**

This layer will consist of Data Access Objects (DAOs) and the model. We will use object relational mapping by JPA. DAOs will represent an abstract interface to the database.

■ **Database layer**

PostgreSQL Database will handle physical reading and writing of data onto the disk.

3.5.8 Data design

Domain model

On the domain model below we can see entity Credentials which include email, id and password. Every type of user should use these. Unlike Admin, Client and Coach need additional data, therefore they inherit from the entity User. Client differs from Coach by additional data, but also with a possibility to create a training session, which can be viewed by the coach after completion. Training session is based on a Training Day from the client's Training plan, which is written by the client's coach. Coach can also create a training template on which a training plan can be based. Training day consists of multiple exercises.

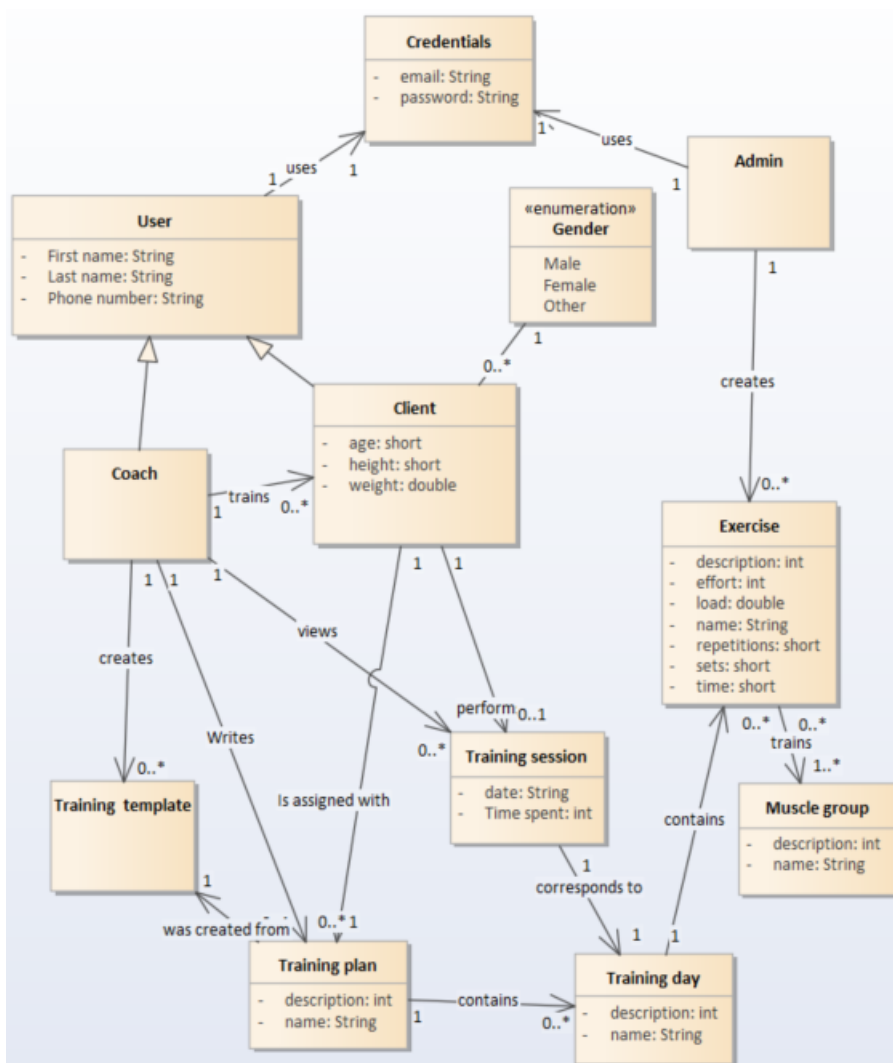


Figure 3.7: Domain model [author]

be represented through the enum Role.

Additionally, usingMetricSystem and isVerified property were introduced. The former can be used in the future if a coach uses a metric system and the client uses imperial. Calculations that convert between weight systems might be performed in the application. The isVerified signifies that the user has confirmed that the email on which he registered is his. This is done through a uniquely generated token which has its own entity VerificationToken.

CoachingInviteToken is used when a coach invites a client to his client's list.

Class Training template disappeared. In the system it will be represented as a class TrainingPlan, but with value of property isTemplate equal to true. This will ease up manipulation with it when creating a training plan from a template.

ExerciseDescription is a representation for an in-real life exercise. This will be the one which will be displayed in a list of exercises for coach and client separately from training plans.

TrainingExercise represents a specific ExerciseDescription in a training day. It contains a collection of ExerciseSet. In reality, not all sets on an exercise can have the same data, a person might have less energy and for example lift less load on set two compared to set one.

Some exercises, such as running, track time and distance compared to load and reps tracked on deadlift. This is reflected in the ExerciseSet entity, although implemented with a strategy "Mapped super class", which is better for this particular case than creating more complexity using hierarchy.

TrainingSession entity now holds data by having two collections of TrainingExercises one for scheduled exercises from the plan and performed exercises, done during the session.

Chapter 4

Development

This chapter is dedicated to a short description of the implementation process. I will try to describe what was changed compared to the analysis as well as mentioning some problems I have encountered. I would like to list some pages that I visited when trying to find some solutions when I had a problem or to find out how I could implement certain new things. I have included their links in the literature list. [26] [27] [28] [29] [30] [31] [32] [33] [34] [35]

4.1 Backend

First step in the implementation was to create a data model. I made one according to the first Class model I have drawn up and established in the analysis (the one listed in this document corresponds to the current version of the app). However, I realised some things that it did not account for, therefore I had to make some adjustments. Next, I implemented the application layers in this successive order: DAO (Persistence layer), Service (Application layer), Controller (Web layer). Then I covered the DAO layer with unit tests and fixed issues that were discovered by them.

After that, I went on to implement security for the application - authorization on specific endpoints and also authentication using JSON Web Token. JWTs are a standardized format for authentication in web applications. I decided to add a confirmation link that is sent to the user through email, therefore the application knows they are the owner of the email on which they registered. This forbids an individual to use the application with an email that does not belong to them. When the user clicks on the link, they are taken to the page where a request is sent to the backend and the validity of the link is checked with a token stored in the database. If the token has expired a user is notified on the page and has a possibility to resend a verification link in which a new token, generated from the old one, is stored. If the token was valid, the information that the user can log in is displayed.

Similarly, but without link expiration, I implemented the client invitation process. The application is prepared for the client or coach initiating the coaching relationship, although currently only the coach can invite a client by an email on the frontend.

When creating Swagger documentation for the api using springdoc openapi

dependency and consulting it with Daniel Groschup, to ensure everything will fit also for mobile frontend, he proposed that some requests might be "resource heavy" by returning too much data up front. I resolved this by implementing Spring HATEOAS, which basically allows obtaining just a part of the data in the request plus a link where the rest of the data could be found. Furthermore, I implemented lazy loading, which will help with resources being "lighter" on the backend. The next step was to start the implementation of the web frontend.

4.2 Web Frontend

First, I created a graphical interface for web pages. A React library Material UI [30] was useful during this process, as it provides a developer with pre-made graphical components that they could base their pages upon and develop faster. To put things together I used "dummy" data which I generated manually mirroring the backend data model.

For routing between pages I used the library React Router [31], which is able to modify the URL and visually distinguish for the user in the browser url field, on which page they are currently present. Another important thing it allows is to recreate a state of the application when they decide to manually reload the page, or if they want to start at a specific page. This is important for Single Page Applications as are created in React, because in them one initial page is modified based on user interaction, rather than taking a user to a separate new file which represents a web page.

Because the coach in the current state of application will not have a large amount of clients and it will be only used as a proof of concept and light testing, I decided to not implement the removal of a client on the frontend side. In the future when it is implemented, some clients might end up in a state without a coach. This is where the possibility for the client to send an email to a new coach as an "invitation" (implemented on the backend) will come in useful.

In order to finish the development on time I decided to keep the responsiveness of the web pages only for tablet and desktop, considering other devices are used for planning very rarely. As a result, the ability for a coach to create a training session for their client was ditched and the web frontend is only dedicated for coach users. On the other hand, mobile frontend is dedicated only to client users. Continuing the implementation, the next step was to connect the frontend with the backend. This required several adjustments on both sides.

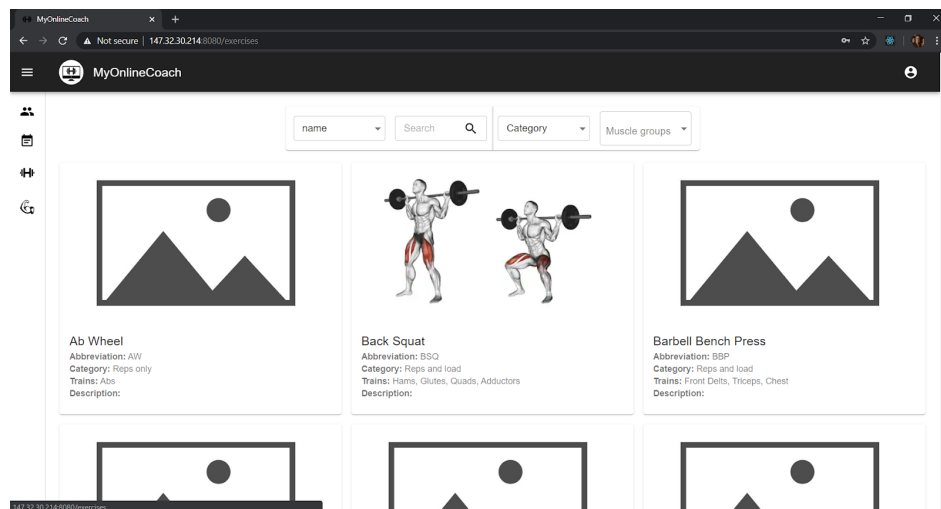


Figure 4.1: Training exercises page [author]

For a demonstration, above can be seen a screenshot from the web application, displaying the list of available exercises.

4.3 Deployment

Once everything above was resolved, the deployment of the application followed. I used the virtual server of Silicon Hill student's organisation, which kindly provided it for me. There were huge difficulties creating a JAR which would contain both backend and frontend that could be run together. After starting the application, backend endpoints were accessible, as well as those on which the Swagger documentation was present. However the application could not find endpoints on which the frontend was present. After the decision to incrementally move the project part by part into a fresh new project I found that the Springdoc Openapi somehow clashes with the frontend endpoints. This resulted in only Swagger documentation being accessible. I temporarily disabled the dependency so that the frontend could be reached on the endpoints. This was a drawback, since it was inconvenient for the development of the mobile app and more adjustments on the web side, but it was necessary. I searched for an alternative dependency, but others posed their own particular problems, the resolution of which would only further delay the implementation. The deployed application was first filled with needed data (admin user entering muscle groups and exercises) and then tested for every present functionality. I made sure to fix all the bugs that appeared from it and also to adjust the styles of the application so it would provide better user experience.

4.4 Mobile Frontend

Once the main application was deployed I joined Daniel Groschup on development of a mobile frontend for client users. It was a part of a semestral project for another university course: Principles of Mobile Applications. Most of the user interface (UI) was already prepared, including navigation between all of the pages. I adjusted the UI so it would be consistent with the theme of the web frontend.

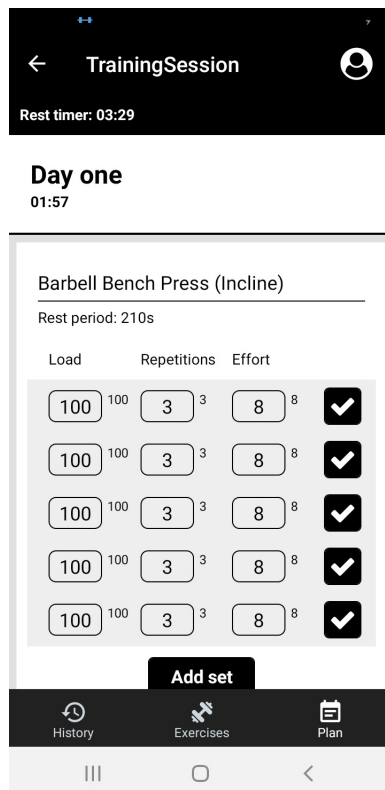


Figure 4.2: Training exercises page [author]

Above we can see one of the screens in mobile application as a demonstration. It shows a running training session.

We decided it would be better to develop the registration of the client on the web side, since it would cost us less time. Hence I put myself into it while Daniel implemented the Login process for the client. As soon as it was done, I followed with connecting the rest of the pages with the backend while Daniel focused on fixing issues and improving the performance of the app.

One of our big challenges in this app was the logic behind training session, to which I dedicated my time, and refreshing the history of training sessions which Daniel successfully resolved.

Chapter 5

Testing in Production

Besides unit tests and ad-hoc testing of UI during the development, the app was also tested by potential future users after the deployment. In this chapter we will look into how it proceeded and what findings were discovered.

5.1 Coach functionalities

The web application was distributed to the coaches, quite sooner than the implementation of the mobile app was finished. Generation of testing data was implemented, so after the coach confirms their account, they could test all of the functionalities as if they would have a real client in the app.

All of the coaches who were contacted for the initial research were invited to try the functionalities together with a couple of other coaches that did not participate before. From the total of 12 contacted coaches, 8 tested the application with the help of a user's guide that was sent to each of them. Each coach was free to explore application on their own. This approach was chosen, because it yielded two advantages: more authentic user experience and simple execution. The feedback was collected through email or direct messages on social platforms. Out of these 8 coaches, 2 reported that they would use it straight away and another 3 reported that they would use the app in the future improved version at least for a part of their clients. The rest of the coaches were unsure if they would use the app in the future. They would need to try it out again to make their decision.

The most requested new features were:

1. Training statistics
2. Adding custom new exercises
3. Videos or pictures for exercises
4. Possibility to plan for at least one mesocycle (4-6 weeks)
5. Adding notes in the plan and sessions

Besides that, there were many more requested features, but almost none of those requests were made by at least two coaches. This might hint at each coach having their own planning style, but also could be influenced by a small number of testing subjects, poor feedback collection etc.

The next step should be to perform a second analysis of the system and outline new functionalities with the highest priority. These might not inherently be the most requested ones by the coaches, such as exercise videos because of their "cost to benefit" ratio. On the other hand, adding new custom exercises, showing total training volume performed in a training session can be implemented relatively quickly and add more value. Additionally, the stability of the system needs more improvement, such as faster email sending, more efficient request to the server, more code covered by tests etc.


5.2 Client functionalities

Three individuals were chosen to take on the role of an actual fitness client and test the application. All of them have several years of experience with working out in the gym. I met with all of them in person for a testing session. Each individual used their own smartphone for testing, while I moderated the execution following a test scenario:

1. Login to your account
2. Go to the exercises screen. Check the exercise detail and try to filter the list by exercise name, category and muscle group.
3. Go to your training plan. Check the info, your coach and choose some training day.
4. Check the info and exercises in the training day. Start the session when you are ready.
5. Write down training variables you performed during the training. Pay attention to the rest timer. If you are done, finish the training session.
6. Go to history. Check that it contains the training session you just performed and that it contains correct data.
7. Go to your profile. Check your info, change if you wish. Go to your coach's profile. If you are done, log out.

At each step they could freely explore the current part of the application, so that their natural behaviour could be simulated. When they were complete with the step I have written down their experience before we moved on.

Overall, besides adding a button to reset a chosen category and muscle groups in the filter of exercises, there were no new features requested. In contrast, several bugs were found during testing most of them related to the user interface. There also sometimes seems to be a problem with the stability of the app. All of the issues were documented and their fix should be an aim of further steps of the mobile app development.



Chapter 6

Conclusion

Resistance training is currently a very popular way to pass free time and increase daily activity. Humans have practiced it at least since antiquity and we have refined it over centuries to the shape it has today. An experienced person can create a training plan which will help the trainee achieve visible results in just a few months and possibly even improve their health. The process of planning and tracking performance has its difficulties and an effective approach to mitigate them is to use technology which is nowadays available to almost anyone.

Many coaches have their approach to training planning adjusted to the tools they currently use, although some would like to improve their process overall while others are missing a few features that they feel would help them better their services. A viable option for them would be to start using a specialized tool. However, the market contains a very small amount of the ones that properly satisfy the needs of the coaches. Due to this fact, I decided to implement an application that would be based on coaches' requirements. The goal was not to develop the system that would contain all of the requested features straight away, but to implement the most fundamental functionalities: creating an easily modifiable plan that would track sets, reps load and effort on exercises, which could also be shared with the client. All of the fundamental functionalities were implemented and the result was presented to fitness coaches and individual trainees who assessed its feasibility. Overall the collected feedback was quite positive and brought in also many suggestions for new features that could be added in the future. The system in the present state could already help coaches improve their training planning. It is currently (summer 2020) accessible here: <http://147.32.30.214:8080/>

Personally, I gained a lot of experience as it brought me the closest to the development of an application that could be used as a real product. I have tried out every aspect of the development cycle, especially the deployment process which I had no previous experience in. This thesis was quite intensive, as it took approximately 430 hours to complete, including writing this text, but I must say the time investment was worth it. The next step should be to find more fitness enthusiasts and developers to continue with this project and make this solid application base a new powerful system that could be used by fitness coaches every day for creating and managing training plans.



Bibliography

- [1] Dennis M. Bramble and Daniel E. Lieberman. (2004). *Endurance running and the evolution of Homo*. Nature. Vol. 432, Pages 345-352.
- [2] Dr. Mfrekemfon P. Inyang and Okey-Orji, Stella. (2015). *Sedentary Lifestyle: Health Implications*. IOSR Journal of Nursing and Health Science. IOSR Journal of Nursing and Health Science. Vol. 4. Pages 20-25.
- [3] Jan Sundell. (2011). *Resistance Training Is an Effective Tool against Metabolic and Frailty Syndromes*. Advances in Preventive Medicine. Vol. 2011. 7 pages.
- [4] Hart, P. D., & Buck, D. J. (2019). *The effect of resistance training on health-related quality of life in older adults: Systematic review and meta-analysis*. Health promotion perspectives, Vol. 9. Pages 1–12.
- [5] Westcott, Wayne L. PhD. (2012). *Resistance Training is Medicine: Effects of Strength Training on Health*. Current Sports Medicine Reports. Vol. 11. Pages 209–216.
- [6] Zavanela, Plinio M; Crewther, Blair T; Lodo, Leandro; Florindo, Alex A; Miyabara, Elen H; Aoki, Marcelo S. (2012). *Health and Fitness Benefits of a Resistance Training Intervention Performed in the Workplace*. Journal of Strength and Conditioning Research. Vol. 26. Pages 811-817
- [7] Stojiljkovic, Nenad & Ignjatovic, Aleksandar & Savić, Zvezdan & Marković, Živorad & Milanović, Sandra. (2013). *History of resistance training*. Activities in Physical Education and Sport. Vol 3. Pages 135-138.
- [8] Diskolobos picture
<https://17green.wordpress.com/2014/01/11/diskobolos> [cit. 26.10.2019]
- [9] Personal trainer history
<https://www.sportsrec.com/412711-personal-trainer-history.html> [cit. 26.10.2019]

- [10] Nerys Williams. (2017) *The Borg Rating of Perceived Exertion (RPE) scale*. Occupational Medicine. Vol 67. Pages 404-405.
- [11] 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 42-49.
- [12] FreeTrainers app
<https://www.freetrainers.com> [cit. 27.10.2019]
- [13] Fitify workouts app
<https://play.google.com/store/apps/details?id=com.fitifyworkouts.bodyweight.workoutapp&hl=en> [cit. 27.10.2019]
- [14] Gym workout app
<https://play.google.com/store/apps/details?id=com.smartfit.gymworkout&hl=en> [cit. 27.10.2019]
- [15] Strong app
<https://play.google.com/store/apps/details?id=io.strongapp.strong&hl=en> [cit. 27.10.2019]
- [16] Fitnotes app
<https://play.google.com/store/apps/details?id=com.github.jamesgay.fitnotes&hl=en> [cit. 27.10.2019]
- [17] JeFit app
<https://play.google.com/store/apps/details?id=je.fit&hl=en> [cit. 27.10.2019]
- [18] Excel app
<https://products.office.com/en/excel> [cit. 27.10.2019]
- [19] Google sheets app
<https://www.google.com/sheets/about> [cit. 27.10.2019]
- [20] Truecoach app
<https://truecoach.co/> [cit. 27.10.2019]
- [21] Fitr app
<https://get.fitr.training/coaches> [cit. 27.10.2019]
- [22] Trainerize app
<https://www.trainerize.com> [cit. 27.10.2019]

- [23] Comparison of backend languages
<https://www.quora.com/What-are-the-advantages-and-disadvantages-of-various-backend-programming-languages-and-frameworks> [cit. 27.11.2019]
- [24] Comparison of frontend frameworks.
<https://dev.to/duomly/the-best-front-end-framework-to-learn-in-2019-dn7> [cit. 28.11.2019]
- [25] Comparison of frontend frameworks
<https://medium.com/javascript-scene/top-javascript-frameworks-and-topics-to-learn-in-2019-b4142f38df20> [cit. 30.11.2019]
- [26] Spring framework website
<https://spring.io> [[cit. 9.5.2020]
- [27] Java blog for developers
<https://www.baeldung.com> [cit. 9.5.2020]
- [28] Java blog for developers
<https://howtodoinjava.com> [cit. 9.5.2020]
- [29] ReactJS framework documentation website
<https://reactjs.org/docs> [cit. 9.5.2020]
- [30] MaterialUI framework website
<https://material-ui.com> [cit. 9.5.2020]
- [31] ReactRouter framework documentation website
<https://reacttraining.com/react-router/web/guides> [cit. 9.5.2020]
- [32] React Native framework documentation website
<https://reactnative.dev/docs/getting-started> [cit. 9.5.2020]
- [33] ReactNavigation framework documentation website
<https://reactnavigation.org/docs/getting-started> [cit. 9.5.2020]
- [34] React Native learning course
<https://www.udemy.com/course/react-native-the-practical-guide> [cit. 9.5.2020]
- [35] Blog for developers with common encountered problems and solutions.
<https://stackoverflow.com/> [cit. 9.5.2020]