



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

**APPLICATION FOR MODERN SPORTS TRAINING UTIL-
ISING THE SENSORY DATA AND POSITIONAL SYSTEM**

APLIKACE PRO MODERNÍ SPORTOVNÍ TRÉNINK S VYUŽITÍM SENZORICKÝCH DAT A POZIČNÍHO
SYSTÉMU

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR

AUTOR PRÁCE

JAN ONDRUCH

SUPERVISOR

VEDOUCÍ PRÁCE

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2018

Brno University of Technology - Faculty of Information Technology

Department of Computer Graphics and Multimedia

Academic year 2017/2018

Bachelor's Thesis Specification

For: **Ondruch Jan**

Branch of study: Information Technology

Title: **Application for Modern Sports Training Utilising the Sensory Data and Positional System**

Category: User Interfaces

Instructions for project work:

1. Get acquainted with the collecting, managing and pre-processing of sensory data. Focus on the positional, motion and biometric data. Get acquainted with the UI design process and Sewio system API.
2. Design an application, that effectively visualises Sewio sensory data for athletes and coaches. Design the UI and appropriate methods for real-time usage and also for offline statistics.
3. Create a dataset suitable for application development and testing.
4. Design the information structure, the data model and the computational mechanisms of the proposed tool and implement it as a web application.
5. Perform the user tests and evaluate the user experience. Discuss the results and propose improvements.
6. Create a poster and a short video presenting the key results of the project.

Basic references:

- Eric Elliott, Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Modern JS Libraries, O'Reilly Media; July 20, 2014, ISBN-10: 1491950293
- Ian Oppermann, UWB: Theory and Applications, Wiley; October 15, 2004, ISBN-10: 0470869178.

Requirements for the first semester:

Items 1, 2, 3 and partly item 4.

Detailed formal specifications can be found at <http://www.fit.vutbr.cz/info/szz/>

The Bachelor's Thesis must define its purpose, describe a current state of the art, introduce the theoretical and technical background relevant to the problems solved, and specify what parts have been used from earlier projects or have been taken over from other sources.

Each student will hand-in printed as well as electronic versions of the technical report, an electronic version of the complete program documentation, program source files, and a functional hardware prototype sample if desired. The information in electronic form will be stored on a standard non-rewritable medium (CD-R, DVD-R, etc.) in formats common at the FIT. In order to allow regular handling, the medium will be securely attached to the printed report.

Supervisor: **Beran Vítězslav, Ing., Ph.D.**, DCGM FIT BUT

Beginning of work: November 1, 2017

Date of delivery: May 16, 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2
L.S.



Jan Černocký

Associate Professor and Head of Department

Abstract

The thesis deals with the creation of a web application for modern sports training. The application uses positional data and heart rate data from a real-time locating system. The main goal is to design and create a system that processes these data in real time and allows coaches to visualize and manage them. The introductory part summarizes the basic principles of creation of a user interface, it also deals with research methods for collection of user requirements. The following text describes how to design and create web applications. Afterwards, the thesis examines testing of web application design. The practical part consists of exploration of the positional system, collection of end-user requirements, design of the application user interface, its implementation and subsequent testing. The work is concluded by evaluating the system together with ideas for its future expansion.

Abstrakt

Práce se zabývá tvorbou webové aplikace pro moderní sportovní trénink. Aplikace využívá poziční data a data tepové frekvence ze systému určování polohy v reálném čase. Hlavním cílem je návrh a tvorba systému, který tato data v reálném čase zpracovává a umožňuje trenérům jejich vizualizaci a správu. Úvodní část shrnuje základní principy tvorby uživatelského rozhraní, dále se věnuje výzkumným metodám pro sběr uživatelských požadavků. V navazujícím textu jsou popsány postupy návrhu webových aplikací a jejich tvorby. Následně se práce věnuje testování návrhů aplikace. Praktická část sestává z průzkumu pozičního systému, sběru požadavků koncových uživatelů, návrhu uživatelského rozhraní aplikace, jeho realizace a následného testování. Práce je zakončena zhodnocením systému společně s nápady na jeho rozšíření do budoucna.

Keywords

UI, web application, real-time, sports training, web design, design thinking, user testing, Node.js, Angular

Klíčová slova

UI, webová aplikace, real-time, sportovní trénink, webový design, design thinking, uživatelské testování, Node.js, Angular

Reference

ONDRUCH, Jan. *Application for Modern Sports Training Utilising the Sensory Data and Positional System*. Brno, 2018. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Vítězslav Beran, Ph.D.

Rozšířený abstrakt

Tato bakalářská práce se zaměřuje na vytvoření webové aplikace pro moderní sportovní trénink, která využívá poziční a biometrická data ze systému určování polohy v reálném čase. Cílem tohoto projektu bylo zjistit uživatelské požadavky na tuto aplikaci, na jejich základě ji navrhnout a implementovat tak, aby sportovní trenéři, kteří ji budou pro trénování používat, byli maximálně spokojeni. Aplikace by měla trenérům umožnit jak online, tak i offline analýzu dat hráčů a poskytnout jim tak data, která jsou pro trénování velice užitečná a která nikdy dříve neměli k dispozici.

Úvodní část práce se zabývá uživatelským rozhraním a jeho designem. Dále byla popsána metodologie *Design Thinking*, která poskytuje informace o postupu tvorby uživatelských rozhraní. Na základě těchto poznatků je v navazujícím textu vytvořen plán vývoje celé aplikace.

Dalším krokem bylo seznámení se s výzkumnými metodami vhodnými v oblasti tvorby uživatelských rozhraní webových aplikací, kdy je nutno získat uživatelské požadavky na výsledný produkt. Současně bylo důležité nastudovat poziční systém firmy Sewio, který bude aplikace využívat. Dále jsou zmíněny existující sportovní aplikace využívající podobné systémy pracující s daty v reálném čase. Tyto nastudované informace společně se znalostmi získanými v úvodní části práce byly následně aplikovány během setkání s pěti profesionálními trenéry, jakožto budoucími uživateli tohoto systému. Jejich požadavky, potřeby a nápady byly poté agregovány, z čehož bylo možné identifikovat klíčové prvky a funkce, které bude aplikace obsahovat.

Po nastudování principů designu a prototypování byla s využitím nashromážděných požadavků vytvořena informační struktura, na jejímž základě bylo možné navrhnout první prototyp aplikace - digitální *mock-up*, který byl použit během *testu použitelnosti* a *pilotního testování* aplikace. V průběhu jednoho týdne byly provedeny dvě iterace testování, kdy byl prototyp prezentován trenérům. Díky těmto setkáním došlo k upřesnění uživatelských požadavků. Dále byla vytvořena datová struktura systému a upravený návrh aplikace byl tak připraven k implementaci.

Výsledkem práce je systém sestávající z aplikačního serveru, který byl implementován pomocí Node.js a webové aplikace naprogramované v jazyce Angular 5. Pro účely vývoje a testování byl vytvořen simulátor simulující pohyb hráčů a jejich biometrická data v reálném čase, která by jinak byla získána ze serveru umístěného ve sportovní hale. Aplikace je rozdělena do dvou klíčových částí. První je určena pro online nahrávání, během kterého mohou trenéři v reálném čase sledovat více metrik o každém hráči, např. současnou tepovou frekvenci či uběhnutou vzdálenost. Také mají k dispozici funkci pro nahrávání cvičení, mají možnost porovnávat hráče nebo sledovat týmové průměry, což jim umožňuje efektivní analýzu hráčů již během tréninku. Druhá část slouží pro offline - *post analýzu* informací, které byly zaznamenány online. Mezi další funkce aplikace patří správa hráčů, týmů nebo nastavení serveru.

Aplikace byla testována čtyřmi trenéry, jejichž zpětná vazba byla velmi pozitivní. Z výsledků byly zjištěny menší nedostatky v rámci designu, například nedostatečná viditelnost určitých elementů kvůli nízkému kontrastu barev, které byly následně opraveny. Finální produkt byl také představen firmě Sewio a několika klíčovými osobám v aplikačním centru Baluo v Olomouci, odkud byly získány uživatelské požadavky. Obě strany byly s prací spokojeny. Bezprostředně po ukázce aplikace proběhla diskuze ohledně její budoucnosti, možných vylepšení a také postavení na trhu. Systém má tak potenciál stát se velmi přínosným produktem v oblasti moderního sportovního tréninku.

Application for Modern Sports Training Utilising the Sensory Data and Positional System

Declaration

Hereby I declare that this bachelor thesis was prepared as an original author's work under the supervision of Ing. Vítězslav Beran, Ph.D. The supplementary information was provided by Ing. Lubomír Mráz and Mgr. David Prycl. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

.....
Jan Ondruch
May 14, 2018

Acknowledgements

I would like to thank the bachelor thesis supervisor Ing. Vítězslav Beran, Ph.D for professional guidance, consultation, patience and suggestive work. I would also like to thank Sewio, Ing. Lubomír Mráz, Mgr. David Prycl and Baluo Application Centre for their support and the possibility of implementing the project. Last but not least, I would like to thank my family and friends, who were persistently supporting me during my studies.

Contents

1	Introduction	3
2	User Interface Design	4
2.1	What is UI	4
2.2	UI Design	4
2.3	Design Thinking	6
2.4	The Process of Designing the Application Creation Process	7
3	Exploring User Needs	8
3.1	Research Methods	8
3.2	Detailed Project Analysis	9
3.3	Applications for Modern Sports Training	11
3.4	Initial User Meeting Proposal	13
3.5	Initial Meeting Results	14
4	Prototype Design	17
4.1	General Principles	17
4.2	Prototyping	17
4.3	Information Structure	18
4.4	Prototype proposal	20
5	Pilot Testing	22
5.1	Usability Testing	22
5.2	Pilot Testing Proposal	22
5.3	Prototype Presentation	23
5.4	Data Structure	24
6	Implementation	26
6.1	Web Technologies and Web Design	26
6.2	System Architecture	27
6.3	Data Models and Database	33
6.4	Visual Design and Functionality	34
6.5	Testing	37
6.6	Future of the Application	39
7	Conclusion	40
	Bibliography	41

A Interview Documents	44
B CD Contents	46
C Installation Manual	47

Chapter 1

Introduction

In the past decades, team sports haven't only been a way of entertainment and a mere prestige. Over the last century, sports have become very competitive and the domain has grown to be a lucrative business.

Nowadays, with the dawn of real-time systems and advanced technologies, it has become possible to boost sportsmen's performance by introduction of profound in-depth data analysis. Given that this type of technology is becoming more and more accessible and competition is rising, the demand for such systems has increased. This condition leads to the current situation where new companies are being found just to find their place in the new niche of the market. They have been developing various positional and tracking systems to bring competitive advantage for teams. Some focus on tactical analysis, others on physical performance of the players. With such data, teams can better tailor their strategy or training methods.

This project is concerned with the use of one of these real-time locating systems in a web application focused on modern sports training. This will allow coaches to accurately measure certain aspects of the training and provide them with useful data they never had before.

The first part described in chapter 2 focuses on user interface design and its theoretical basis, which is useful for understanding the complete development process. The following section 3 is firstly concerned with the study of research methods to explore user needs, the positional system and existing sports applications. This knowledge is used in practice when meeting with sports coaches to identify key features of the application. Chapter 4 deals with general principles of design and prototyping. Subsequently, information structure of the application is proposed and on its basis a prototype is created. Pilot testing 5 is focused on improving this prototype with the use of usability testing. The end of this chapter presents a data model of the application. Chapter 6 includes the study of web technologies and web design, but it is primarily focused on implementing the application and testing it with professional coaches. The last paragraphs are concerned with user feedback and ideas about the future of the application.

Chapter 2

User Interface Design

The first chapter contains a theoretical basis of *user interface*, hereafter UI, and its design, which is crucial for understanding users. This will enable me to create a well-designed product, that meets the needs of the sports coaches. The following chapters are concerned with the UI - what it is and how to design it. Afterwards, a universal methodological approach called design thinking is presented, which will be very useful during the whole development process of the application.

2.1 What is UI

Nowadays, UI is something that we interact with on a daily basis and mostly without even noticing it. It allows us to effectively operate and control electronic devices. If we want to find an exact definition of UI, we come across various interpretations of this term. This is due to the complexity of the field and its relationship to many other disciplines such as human-computer interaction or human-machine interface [1].

Everett McKay defines UI as *an objective, principled form of human communication, and not as a subjective art* [2]. Other resources provide us with different definitions, e.g. stating UI as *the means by which the user and a computer system interact, in particular the use of input devices and software* [3]. It is possible to encounter countless other definitions, differing in what the stress is put on. However, what all of them share together is an emphasis on the user. Quality of a particular UI can be measured by how self-explanatory it is or how little energy it requires us to think about using it [4]. The less input the operator puts in and the machine returns a desired outcome, the better. A good UI should render itself as if almost invisible, making the interaction seem effortless and straight-forward. Simply put, it has to be intuitive [5].

2.2 UI Design

In order to create any kind of UI, we have to consider many aspects that play a role in the final quality of our design. First and foremost, we have to realize whom and what for is a particular UI going to serve. Since designers are very rarely the users [5], they have to develop a good understanding for the target group. Therefore, cognitive empathy can be a key element determining success of UI [6]. This particular paradigm of the design process can be summarized as follows 2.1:

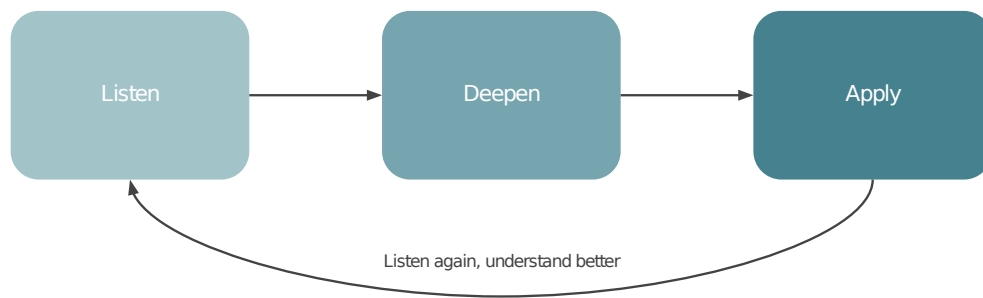


Figure 2.1: Process of understanding users

This requires designers to master their listening skills to be able to truly understand user’s motivation and goals. This intense focus on an individual can go beyond data-driven research methods.

In order to increase product usability, taking human behavior and biases into account, we require a set of *design rules*, that help us to navigate during the design process [1]. Some of the rules are required, others may be considered mere suggestions. They may also vary in their level of abstraction. These rules are then divided into three categories: Principles, Standards and Guidelines. If one embarks on studying these in a greater detail, they will soon find out that it requires a lot of energy to interpret them in any practical form. So few advocates on user-centered design came up with a set of heuristics - a kind of universal design advice, sometimes also called *Golden Rules*. There exist many Golden Rules, however, the most commonly used are Nielsen’s ten heuristics [7], Norman’s seven principles and Shneiderman’s eight golden rules. All of the listed ones are very similar in nature. This is an example of Norman’s seven principles [5]:

- Use both knowledge in the world and knowledge in the head
- Simplify the structure of tasks
- Make things visible: bridge the gulfs of Execution and Evaluation
- Get the mappings right
- Exploit the power of constraints, both natural and artificial
- Design for error
- When all else fails, standardize

All of these design rules apply in general and will help us create a better product for the user. They can be supported by psychological, cognitive, ergonomic, sociological, economic or computational theory. Combined with the cognitive empathy approach, it is possible to design desirable UIs. And as we have seen, designing a good UI is a challenge and often an evolving process. Since we design daily, we solve various problems to make our world better - more intuitive to interact in, eventually shaping our view and perception of it [5], it is important to understand the design process really well. And as it happens, once something new emerges, best practices develop and new methodologies are constructed to find the most effective way of doing things.

2.3 Design Thinking

As regards the UI design, a methodology called *design thinking* was created. Design thinking is a design methodology that provides a solution-based approach to problems solving [8]. By understanding the human needs first, re-framing them, by creating many ideas in a brainstorming session and their subsequent realization via prototypes and testing, it is possible to solve complex problems and design well. The process of design thinking isn't unique to designing UIs, but it's a general methodology for solving problems.

The origin of design thinking dates back to the 1970's and since then a lot of variations of the model were introduced, varying in the number of stages in the process. However, the general idea of all models is the same, thus rendering the nuances negligible. For instance, the first formal design thinking model by Herbert Simon had seven stages. Later developed models differed, such as the Double-Diamond Model [9] produced in 2005 by the British Design Council, that has four stages. The model I chose to work with is a five stage model 2.2 consisting of the following parts [8]:

Empathize is the first stage with a direct usage of the need for empathy, mentioned in the previous chapter. It is necessary to understand and clearly identify the problem (the correct problem). It allows the designers to gain a better insight into the "world" of users. Since all the stages are directly dependent on each other, unless empathising is properly executed, the final product will not yield good results.

Define helps us to gather and then re-frame all information from the previous stage in a human-centric way, addressing exact problems. A well-defined problem statement is a kind of transformation of discontent of a given issue that the user has put into a clearly articulated form. It strips away ambiguity and vagueness.

Ideate is the following stage of the process. After understanding the needs and their analysis, designers are now ready to come up with possible solutions to the problem. At this stage, techniques such as brainstorming, mindmapping, brainwalking or the worst possible idea are used to come up with a variety of the solutions. Some of the techniques allow designers to expand the problem space or to "think outside of the box". Others provide a larger "democratic and safe" environment by having the people come up with solutions anonymously, thereby avoiding the authority bias [10].

Prototype is the moment when designers are ready to create a number of scaled-down, quickly made and inexpensive prototypes and mock-ups. These will serve to further identify possible shortcomings and misconceptions, as well as to select "the best candidates" among the prototypes, that will be later presented to the user. At this point, the prototypes are consulted outside the team or the person who created them to get a more objective feedback from people, who are not biased [11].

Test concludes the whole process. The prototypes are presented to the users. Naturally, designers gain more feedback, insight and nuances regarding the usage of the product. Afterwards, these results define an updated version of the current product. The users can be again re-emphasized, ideas re-defined and re-ideated. This process reflects the iterative nature of the model. If done right, with every step, we get closer to the desired outcome. The iterations continue as long as resources are available - time and money.

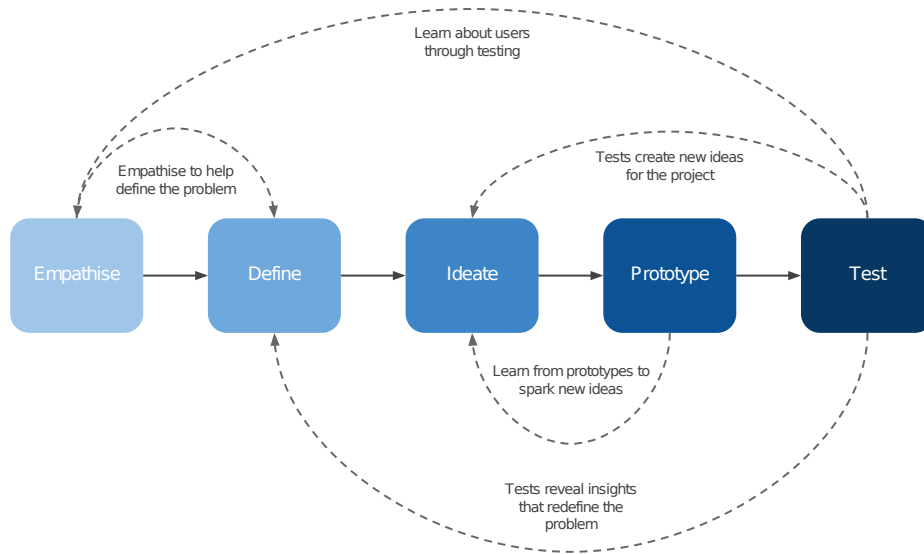


Figure 2.2: Design thinking process

Armed with the knowledge of what UI is, how can it be designed well and how to proceed while designing it with the use of the design thinking methodology, we can embark on “deciphering” the assignment of creating the application for modern sports training.

2.4 The Process of Designing the Application Creation Process

Based on the findings gathered in this chapter, I propose to approach the design process using design thinking, as it’s a versatile and well-proven methodology for designing UIs. Taking advantage of the predefined steps, there is an exact procedure, that I will follow. Thanks to the non-linear iterative nature of the model, it will also be easy to adapt to possible changes at different stages of the development. This will ensure a higher quality of the final product.

I suggest to start with meeting the users (empathize stage), which are in this case coaches of sports teams. After talking to them and understanding their needs, I will aggregate all the acquired information and derive (define stage) clearly articulated needs of the coaches in a human-centered manner. Subsequently, I will come up with various solutions (ideation stage) using some of the ideation techniques and identify the most feasible ones, that will fit the overall idea of a sports application. More tangible representations of the ideas will be put into prototypes, that will be presented to the coaches (prototype stage). As changes in the design based on the user feedback are expected, after a number of iterations of the first four stages of the process, final prototype will be agreed on. This prototype will largely resemble the design of the final application - small changes will be necessary, based on the final testing of the product (test stage).

Chapter 3

Exploring User Needs

The first task in the process is to get to know the user. For this purpose, there are many research methods. The following paragraphs describe those that are important to the design of the UI, in more detail along with their pros and cons. Finally, I choose the best-fitting ones that will be used at the empathize stage of the design thinking process. This chapter also discusses the details of the project, as well as other sports applications similar to the one that I have been developing.

3.1 Research Methods

As mentioned above, a huge amount of various research methods has been developed in order to gather all sorts of data from people. We might encounter different classifications [12], such as experimental and non-experimental methods, laboratory vs. field studies, subjective vs. objective methods or, for example, qualitative and quantitative methods. Then also, selection and application of a particular set of methods strongly depends on the field of study in which we conduct research.

The following is a list of applicable research methods suitable for collecting user requirements for the product:

Interview is one of the basic forms of research, when the interviewer (preferably the designer) is interviewing current or future user of a product. The ideal ratio of interviewers to interviewees is 2:1 or maximum 3:1. Meeting may preferably take place in the context of the product, that is being designed. The reason is to understand the user in the immediate vicinity and thus develop a high level of empathy. One session usually takes anywhere between 1 - 2 hours. There are various forms of an interview such as a *structured interview*, when questions are prepared ahead, or, for instance, a *semi-structured interview*, when sub-questions are prepared, in order to go into further detail of a particular question if necessary [1]. The latter one is oriented on a better quality quantitative research.

Questionnaire is the second query technique. In this case, questions are fixed in advance, which allows less probing. On the other hand, a wider participant group can be reached as this method takes less time to administer. It is crucial, that the questions are well prepared. Compared to the interview, the acquired sample is larger, but at the cost of quality of the answers. There are several styles of questions that can be part of the questionnaire,

including general, open-ended, scalar, multi-choice and ranked questions. Each of them has their own pros and cons, suitable for different situations and purposes.

Focus group usually consists of 5-8 people [4], that react to ideas and designs that are presented to them. It's a form of qualitative research and can be great for determining what the audience wants. The moderator is in charge and directs the session. The client can passively watch the discussion via video cameras or from behind a semi-permeable mirror. This method is great to use in the early stages of the design process for quickly getting a sample of users' opinions and feelings about things. One has to be careful with proper selection of the group to minimize possible authority bias [10], when the more dominant individuals influence the opinions of the rest of the respondents.

Observational research is a type of research in which the researcher observes the on-going behavior [13]. Research can be either direct or indirect. *Direct research*, also known as an observational study, consist in observing the subject in their usual environment with the aim of gathering evaluative information [14]. It is used when the formerly mentioned methods are ineffective and more qualitative data is needed. The observation can be either overt, when the watched subject is aware of the monitoring, or covert, when the subject is unaware of the purpose of the observation. *Indirect research* is based on data collected by other researchers, thus depending on secondary sources. The advantage of this approach is its non-intrusiveness as well as the fact, that the observer and the researcher have different points of view. Disadvantages of this method are to be found in the second-hand data interpretation and their possible incompleteness of the observed situation [15].

Before we conduct a research, it is necessary, that the test takers are informed about the fact, that they will be observed and monitored. The researcher (designer) usually obtains informed consent from the tested subject, where they agree on serving as a research participant after being informed about the nature of the research and their rights and responsibilities [12]. This process is crucial, unless one wants to end up in a lawsuit.

Also, a small study is conducted before the main research is fully started. It is called *Pilot study*, also known as *pilot research*. It is to test specific aspect of the research design and if necessary, to adjust it [16]. In this way, it is possible to, for example, fix ambiguous questions in a questionnaire.

3.2 Detailed Project Analysis

The goal of this thesis is to create a modern sports training application for coaches that will enable them to effectively train players and thus improve overall team performance. The customer is Sewio Networks s.r.o.¹. The company has their own *Real-time locating system* (RTLS²) and they want to develop their own sports application that would use their system. The application should be a web application, universal for team sports, and is not uniquely targeted on one sport. The reason why this sports application is called „modern“ lies in the way it gets and uses the data - and what kind of data it is.

¹<https://www.sewio.net/>

²To avoid ambiguity, RTLS represents a special kind of positional system. However, the words „locating“ and „positional“ are interchangeable in the context of this thesis.

It will utilize data from the RTLS provided by Sewio³. The RTLS platform is a system based on *Ultra Wideband technology* (UWB), which is a sort of radio technology for transmitting information, taking advantage of *Time Difference of Arrival* (TDoA) scheme⁴. *Tags* - transmitters, which are small portable devices⁵ 3.1, transmit information, that they collect (such as position), to *anchors* - receivers, that are usually 6-8 devices located on the sides of the monitored area. Subsequently, the anchors send the information via Wi-Fi or IP Network to the server, where the data is processed. Afterwards, it is stored into a SQL database and also it is possible to directly send the data to the client. This process, depending on technology and settings, can take anywhere from 30 ms to a number of seconds.

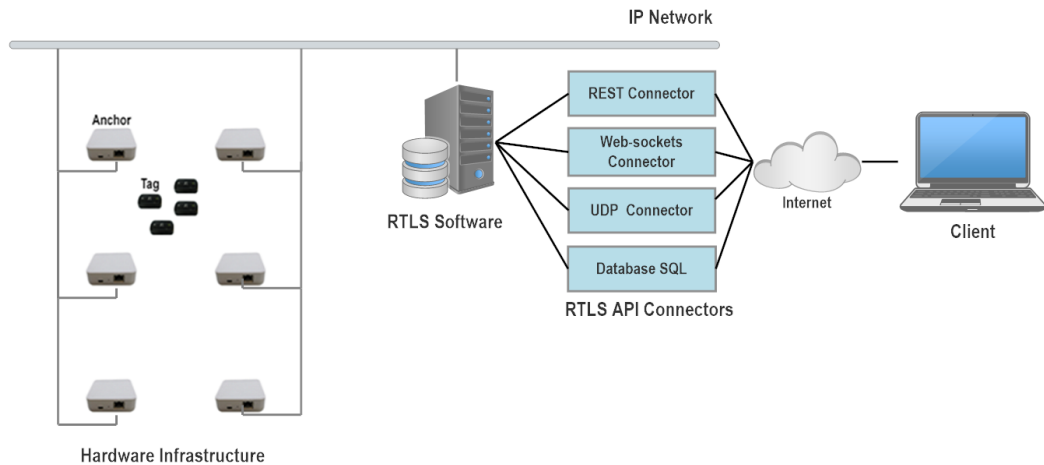


Figure 3.1: RTLS Overview

In sports the tags are attached to players. As the players move, the type of information, that the tags periodically send to the system, which can eventually store it, lists as follows:

- **Position** - x, y, z coordinates
- **Timestamp** - time information
- **Heart rate** - hereafter HR, current heart rate of the player
- **Zone** - area in which the tag finds itself

In the application itself, by using simple physics equations, it will be possible to measure and store more information, such as for example:

- **Distance** - distance, that the player covered.
- **Speed** - current speed of the player along with various aggregates of the speed information, such as average speed, maximum speed, etc. It will also be possible to classify movement - walking, running, sprinting.

³<https://www.sewio.net/technology/>

⁴<https://www.sewio.net/technology/time-difference-of-arrival/>

⁵Image source: <https://goo.gl/aPN9pQ>

- **Average HR, maximum HR, HR zones** - various aggregates of the HR information.
- **Heat map, spaghetti map** - graphical representations of information characterizing player's position.

This kind of information available for the coaches in real-time will be the crucial element of the whole application. This is important to stress, because this real-time system gives the application the competitive edge over other applications for sports training that don't use any similar technology.

3.3 Applications for Modern Sports Training

The following part of the thesis features a number of currently existing applications that focus on modern sports training. All of them utilize some kind of a locating system acquiring real-time data and implement similar functionality to the application, that I have been developing. They will serve as a great inspiration. On top of that, they can be used as „free“ prototypes later during the mock-up presentation stage [4].

3.3.1 EchoSports

EchoSports⁶ is a web application that implements both online and offline training analysis. It uses Sewio's locating system and focuses on a variety of team sports. It provides the user with an in-depth analysis of the training session, allows comparisons across many trainings of both the team and individual players. This is visualized by multiple bar charts. One of the application's prominent features is a timeline 3.2, that enables the user to replay a specific training section. All of the recorded trainings are stored and accessible through an event calendar.

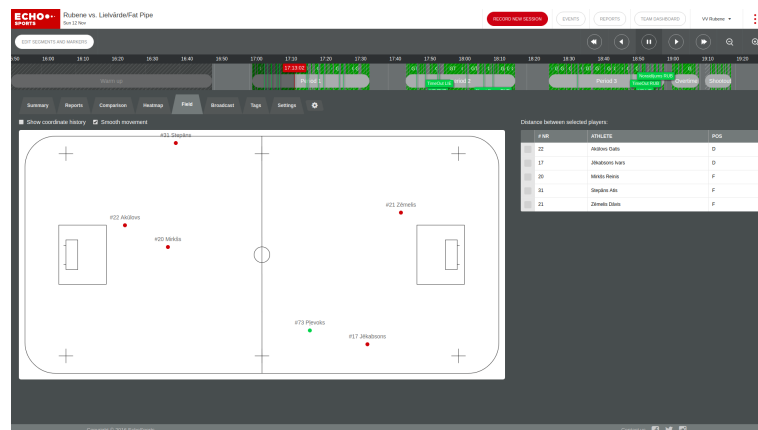


Figure 3.2: EchoSports timeline feature

⁶<http://www.echosports.eu/>

3.3.2 Kinexon

Web application Kinexon⁷ focuses on both on tactics as well as on fitness of the players. Thanks to their locating system connected to a video system, it is possible to perform more-profound tactical analysis of the game. Compared to the other products, their huge advantage is the ability of ball tracking. As regards Kinexon's second field of interest, the fitness module allows the coaches to specify the training load for the players, helps them prevent injuries by using sophisticated data analysis, and provides them with additional data aggregates useful for post-training analysis⁸ 3.3.



Figure 3.3: Kinexon post-match analysis

3.3.3 Catapult

Catapult⁹ is another world-class player in the modern sports training software, whose main goal is to improve team performance and manage the risk of injury. Their mobile application Playertek¹⁰ introduces some social features such as leaderboards or „Friends League“, that boost users' motivation. However, it means that Playertek is primarily designed for players themselves, rather than for coaches. Their other software products are comparable to the above-mentioned applications and offer similar functionality for in-depth analysis¹¹ 3.4.

Studying these sports applications gave me a hint about overall functionality and useful features of such systems. It also helped me to understand possible user needs. This information will be very useful during the meetings with the coaches.

⁷<https://kinexon-sports.com/>

⁸Image source: <https://goo.gl/89P7KF>

⁹<https://www.catapultsports.com/>

¹⁰<https://www.playertek.com/>

¹¹Image source: <https://goo.gl/RDdYkz>



Figure 3.4: Catapult post-match analysis

3.4 Initial User Meeting Proposal

To design a great UI for the application, I will take advantage of some of the above-mentioned methods of research, that I studied. Due to the importance of developing a high level of empathy towards the user, I will focus on using qualitative methods of research. At this stage of the process, it is key to listen to the coaches and find out what they exactly need, how they train now and what their problems or “itches” are.

3.4.1 Overview

I suggest to conduct the initial meeting in the form of a semi-structured face-to-face interview combined with the use of audio recording (if allowed). There is going to be a set of questions prepared primarily focused on *how do the coaches train today* A.1. The basic is that if the application is going to solve some problem or help the coaches to achieve better training results, it has to respect the demand that stems from real situations. Upon completion of the interview and an overview of their problems, I am going to present the coaches with a spreadsheet, that is going to contain information about the possible functionality and data, that the company’s system can provide A.2. At this stage, the identified needs of the coaches will be merged with the system capabilities, generating feasible solutions that I will be able to implement later.

There will be more of these initial meetings, depending on resources, as well as on how well the meetings will take place. As I am going to be the only interviewer there, I will meet maximum up to 2-3 coaches at once. I am also going to take notes during the meetings.

3.4.2 The Interview

The interview is going to be conducted in an informal way to ensure that the coaches feel comfortable and don’t feel pressured. If necessary, the meetings can take a looser form, and the discussion may not necessarily be structured around the questions. I am planning to keep the questions for myself rather as a guide and a reminder of the information that I need to get from the coaches.

3.4.3 The Merging Stage

At this stage I am going to narrow-down all of the ideas gathered during the interview. We are going to talk about specific, tangible solutions and their implementation. In the course of this process, I may draw some of the proposals of the ideas, because if one can't even do that, *the solution probably isn't there* [17]. In the context of the design thinking, we are talking about the define stage of the process.

3.4.4 Post-meeting Process

After all meetings, I am going to gather the acquired information and aggregate it in a clear form. At this moment, I am going to go over the audio recordings, drawings and notes from the meetings. Afterwards, I will be ready to move on the next phase described in 4.

3.5 Initial Meeting Results

Within two weeks, I managed to conduct three meetings with five coaches in total, of which three of them hold the A-license coaching qualification, allowing them to train the first league teams in the Czech Republic. The discussed sports were basketball, ice hockey, floorball, handball and volleyball.

Before the first meeting took place, I firstly carried out pilot testing at home. My brother, who is well-versed in sports, was in the role of a coach. During this testing I was able to identify some shortcomings in my questions as well as in my preparation. In the course of the first meeting, there were several moments when I “froze” and didn't have any follow-up question prepared, thus negatively influencing the quality of the discussion. However, I learned from my mistakes and the rest of the meetings were considerably better. Also the information and ideas gathered on those two were of a higher quality.

3.5.1 Key Features

During the meetings, I found out that coaches rarely use any technology to measure player performance. It's mostly their intuition and experience that tells them how to train. Nonetheless, all of them had a lot of ideas on what could be measured and how it could be beneficial in their training. Finally, in combination with the system capabilities, the following key features were identified across all of the above-mentioned sports 3.1:

Simplicity (general principle): trainings are nowadays quite intensive and coaches don't have much time to keep an eye on the players as well as on the application. Also after training, coaches don't have time for an hour-long analysis.

Features:

Online features	Offline features	Other features
complete training and exercises acquisition	training summary	tags status information
speed and HR zones	individual exercises	teams and players management
current and average HR	events calendar	
sprint distance and number of sprints		
total distance covered		
filtering: HR, distance, sprint distance		

Table 3.1: Key features

During the online (real-time) view, the coaches need to capture the entire training in order to get complete statistics such as total distance covered during the training. Sometimes they need to measure some section of the training separately, therefore exercises acquisition will also be possible. To characterize and conveniently display players' HR and speed, HR and speed zones will be implemented to show relative ratio between the different zones and the time spent in them. Probably the most important information will be current HR of a player, which the coaches really wanted to see in the application, because HR isn't „visible“ under normal circumstances, but it tells a lot about the player's physical condition. Another useful metrics will be sprint distance, number of sprints or average HR. Last but not least, there was a demand for the possibility of filtering players according to different metrics, so the coaches can easily find out, how well is the player doing compared to the others.

In terms of offline features, the coaches will be able to perform a post-analysis of the acquired training along with individual exercises. For this purpose, they required something like an event calendar, that would enable them to browse all the recorded events.

Since the application is going to utilize data about players, it is necessary to divide the players into teams and have the possibility to manage all this information. The coaches also wanted to have some information about the status of the tags, so they can check if the system is functional.

3.5.2 Scenarios

The list of the key features was largely proposed based on actual situations during trainings and their overall form. Here are some examples:

Measuring HR during an exercise: it is often necessary, that the players should reach a certain HR value during exercise, whether it is for the purpose of reaching a given training load or to prepare them for the next part of the training.

Example: the coach wants the players to reach a HR value of 140 bpm¹², so they are ready for a subsequent „3 on 3 game“. As the exercise starts, the coach checks the application to decide whether the exercise is effective or not.

Keeping the players in a particular HR zone for a given amount of time: the coach wants players to be at the aerobic threshold for a certain time to increase their stamina [18].

Example: the coach starts the exercise and wants the players to be at the aerobic threshold that is set at 160 bpm. Then he monitors players' values. Based on this information, he can decide, whether the exercise is effective enough or if it is too demanding, and adjust appropriately.

Simple post-training analysis: the coach can check the overall statistics summarizing the training or, if necessary, to go in greater detail. However, he doesn't want to spend too much time dwelling on the numbers.

Example: during the training, a player gets injured and another seems to be “out of shape”. The coach checks the data and finds out, that the training wasn't unusually demanding or dangerous. But he knows that the player also did two other sports that day

¹²Bpm = Beats per minute, unit for measurement of the speed of the heartbeat.

and that they are probably overtrained. The coach has a clear evidence, that the injury wasn't his fault, but it was rather an aftermath of an ill-considered training load, that the player chose for themselves.

After learning about research methods, Sewio's locating system and a few relevant sports applications, and after conducting the initial meetings and identification of the key features, scenarios and understanding the needs of the coaches, I am now prepared to move onto the next part of the project, which are the ideation and prototype stages of the process.

Chapter 4

Prototype Design

Now that exact specifications and application requirements have been obtained, I am ready to design a prototype. But before doing that, it is important to understand the design process. It is necessary to know how to design, what are some of the approaches to it and what can help us in designing. Afterwards, information structure will be outlined to enable us create the prototype.

4.1 General Principles

Often good design is an evolving process. The design is tested, problems are found, design is modified [5]. The process repeats and continues until resources are depleted. But before we start designing something, it's important to realize the main goal of our design, especially when it's a user-centered design. It's very common for designers to forget that they are doing the work for the user. At first glance this may sound stupid and obvious. However, there are a number of "traps" to step in. There are two deadly temptations for designers [5]:

- **Creeping Featurism** - keep adding useless features until it's too difficult to use
- **The Worshipping of False Images** - make it complex because it looks cool

This is related to the idea, that design is a constant battle between usability and aesthetics. All this is caused by many human biases [19], that the designers have to cope with and ideally overcome. The usefulness of empathy still applies.

4.2 Prototyping

Due to the complexity of human behavior, designers find it very difficult to come up with a „perfect“ solution on the first attempt. Therefore, many iterations of the design process are usually required before the product is of a certain quality [1]. This is why prototypes are used as early-stage communication tools, presenting to the user early ideas and simplified versions of the final application. This saves time and money, that would have been otherwise wasted on building a perfect product, only to find out it's nothing the client had expected it to be. It's basically an experimental model of an idea.

In some areas, technique of *rapid prototyping* can be used, that has been recently strengthened by the broad availability of 3D-printers. However, something like an automatic UI generation can't be applied in the field of UI development yet. Therefore, we

have to use a different approach to rapid prototyping. As regards to web applications development, Google recommends using the following procedure [20]:

1. Sketching and Paper Prototyping is the first step of the process, when low-fidelity prototypes are created. They are great for exploring all the initial ideas that the designer has. It is possible to outline the steps in a user flow, explore a variety of layouts and create a *wireframe*¹ - basic application structure. Then, higher-resolution paper prototypes can be made by adding colors, images and buttons. The next step is to create a paper prototype of the application. Using these, it is possible to simulate key user interactions or for instance explore elevations and shadows. A useful thing when prototyping is to print existing screens to avoid re-drawing them all the time again.

2. Digital Prototyping allows to rapidly explore an idea by building interactive prototypes, that run on device. This process still doesn't require any programming knowledge. However, the product is already visually quite close to the final product and can be shown to the users (sketches can be presented to the users as well, but they may be limited by a relatively poor visual appearance). It also incorporates transitions between the screens along with other interactive elements. In the recent years, many tools such as Principle², InVision³ or Marvel⁴ have been introduced to the market, making this kind of prototyping very effective.

3. Native Prototyping is the final stage of the process that allows to fully explore an idea along with its hardware functionality - finding out, how it will actually work. This means writing some code, HTML, CSS and JavaScript can be used for web applications. Native prototyping can help validate ideas by showing results to users on real devices. This gives clients the most realistic picture of the final product. The disadvantage is the complexity of this process as it requires a lot of resources. Native prototypes are built once we have a very clear idea of the application and want to communicate and sell our vision to the public.

4.3 Information Structure

Depending on the acquired requirements described in 3, we can determine what actions the coaches will perform within the application. It is possible to make use of a *Use Case diagram*⁵, that visualizes actions the user can perform within the given system 4.1.

However, even more useful during this stage will be a *sitemap* 4.2, as it helps to layout a structure of the application. Again, based on the key features described in the previous chapter 3, the application's functionality can be divided into three parts - online, offline and other features.

I suggest to create a dashboard, that will serve as a „signpost“ for the application. From there, the coaches will be able to start recording a new session and thus get into the online part. There will be overview of all of players containing information such as current HR and total distance covered. In addition, the coaches will also have a possibility to compare

¹A two-dimensional illustration of page's interface, representing the skeletal framework of a website.

²<http://principleformac.com>

³<https://www.invisionapp.com>

⁴<https://marvelapp.com>

⁵<https://www.lucidchart.com/pages/uml-use-case-diagram>

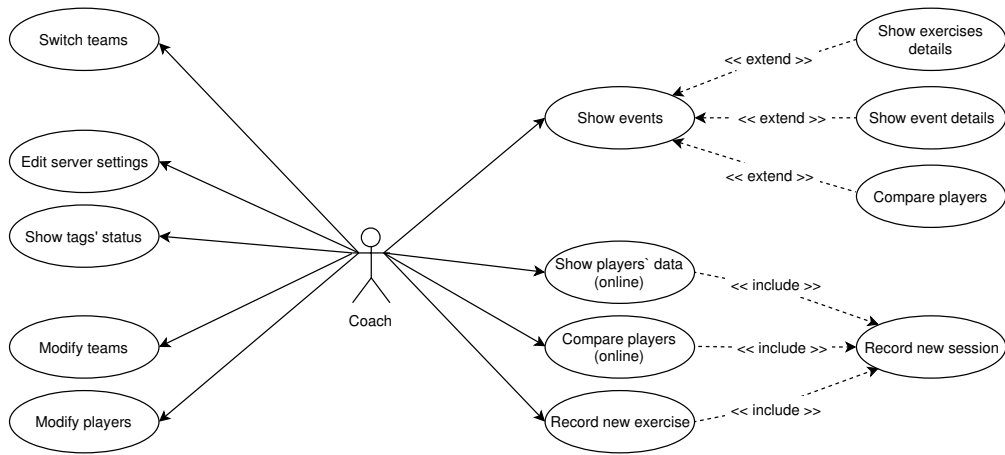


Figure 4.1: Use Case diagram for a coach - key functionality

the players and filter them by different metrics. The offline part will enable the coaches to analyze the recorded training sessions - *events*⁶. It will be possible for them to analyze team summary as well as individual exercises. As in the online part, the coaches will be also able to compare the players. Finally, additional features such as settings, players, and teams will be available to allow the user to modify the application's dynamic data, for example server settings or players and team information.

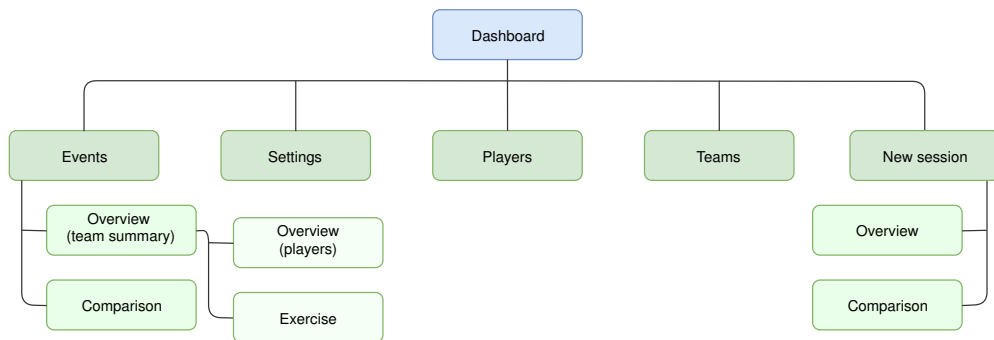


Figure 4.2: Sitemap

Now that I know how the coaches will work with the application and how to structure the required features, I can start designing the first prototypes.

⁶To avoid ambiguity, new or online recorded trainings are called „sessions“, whereas finished recordings are called „events“.

4.4 Prototype proposal

Based on the information I studied, I suggest to start by sketching and creating a number of low-fidelity sketches and paper prototypes. These will allow me to explore a whole range of design layouts of the key elements. In addition, I will be able to study interactions and transitions of screens and windows of the application. Once a satisfactory result is achieved, I will translate the paper version of the application into a digital prototype. These mock-ups will also be printed and presented to the coaches.

After I explored all sorts of compositions using sketches, I came up with this digital prototype 4.3 created in in the aforementioned tool Marvel:

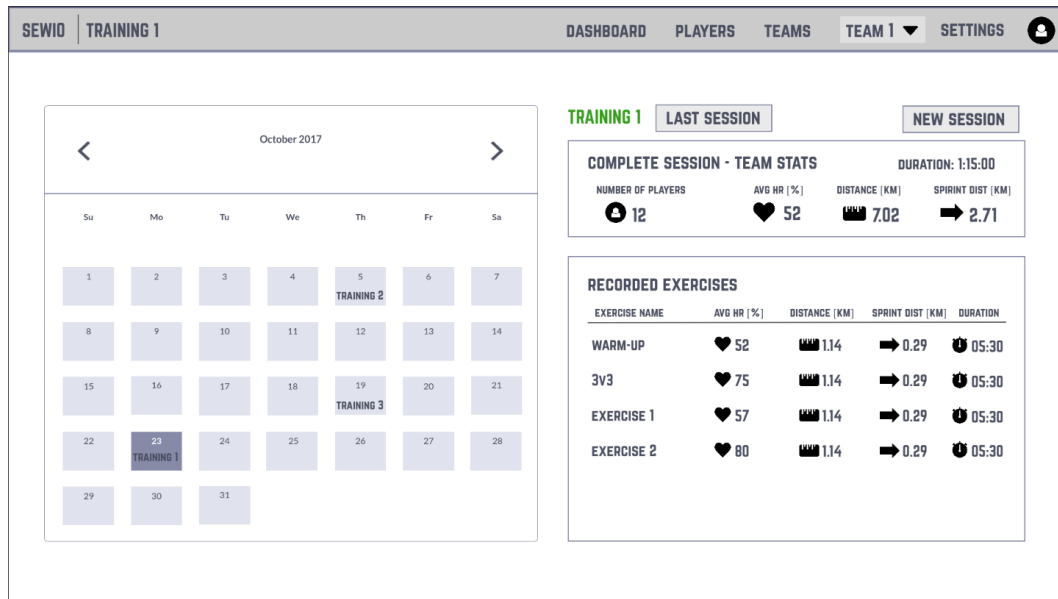


Figure 4.3: Dashboard featuring events calendar and the last training information

The navigation bar menu contains the most important options that should be directly accessible throughout the application. *Dashboard* switches user to the *home* screen (dashboard itself), *teams* allows editing teams (add, edit, remove). *Players* function analogous to the teams, and similarly to these function *settings*. As the coaches always want to know what team is currently active, the team's name will be conveniently displayed right next to the teams option. There also may be a secondary navbar⁷ just below the primary one, depending on the chosen screen.

The dashboard of the application features a calendar containing events. This allows the coaches to effectively browse events. They can also access them for post-training analysis where they can inspect more profound data from the event and draw useful conclusions from it. On the right side of the screen is an overview of the last session. This is due to the fact, that the coaches usually want to analyze the session, that they just recorded. Both event summarization and individual exercises are displayed for brief overview. The new session button enables coaches to start recording a new training. After the user clicks on it, they enter a name for the session and choose which players to record (1- to -n number of players).

⁷ „Navbar“ is an abbreviation for „navigation bar“, commonly used in the web development community.

After starting the session, there is going to be an overview displaying the most important information about each player 4.4. Today, trainings are mostly designed to improve players' agility and speed to simulate real match scenarios. Based on this fact and the given requirements, the following information will be displayed:

- **Relative HR** - information in percent compared to the player's maximum HR, enables to quickly identify, how well is the player doing and draw conclusions from it. The process of faster recognition, such as critical HR (above 90%) is enhanced by color resolution of the numbers.
- **Current HR** - provides more-detailed information about player's condition.
- **Distance** - covered distance is a useful factor for identification of player's abilities.
- **Number of sprints** - measurement unit that directly reflects the need for dynamic and agile training.
- **Sprint distance** - provides similar information to the coaches as the number of sprints unit of measure.

Having this data available in real-time, the coach will be able to immediately see information about his team and draw conclusions from it. In case the coach wants to see more information about each player, or compare the players by the current HR, distance or sprint distance, they can navigate to *Comparison* view. It will be also possible to record exercises to capture specific parts of the training, which will be very valuable for later post-analysis. On the right side of the second navbar, there is going to be displayed total length of the session, so the coaches don't need to measure it separately. The same applies to the exercises - after starting the exercise, its length is going to appear right next to the name of the exercise.

SEWIO TRAINING 1		DASHBOARD PLAYERS TEAMS TEAM 1 ▾ SETTINGS	
OVERVIEW		COMPARISON	
RECORD EXERCISE		DURATION 08:22 SAVE SESSION	
FIRSTNAME SURNAME ♥ 128 HEART RATE 51% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST	FIRSTNAME SURNAME ♥ 162 HEART RATE 74% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST	FIRSTNAME SURNAME ♥ 128 HEART RATE 51% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST	FIRSTNAME SURNAME ♥ 162 HEART RATE 74% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST
FIRSTNAME SURNAME ♥ 128 HEART RATE 51% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST	FIRSTNAME SURNAME ♥ 162 HEART RATE 74% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST	FIRSTNAME SURNAME ♥ 128 HEART RATE 51% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST	FIRSTNAME SURNAME ♥ 162 HEART RATE 74% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST
FIRSTNAME SURNAME ♥ 128 HEART RATE 51% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST	FIRSTNAME SURNAME ♥ 162 HEART RATE 74% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST	FIRSTNAME SURNAME ♥ 128 HEART RATE 51% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST	FIRSTNAME SURNAME ♥ 162 HEART RATE 74% 6.02 28 → 0.89 DISTANCE SPRINTS SPR. DIST

Figure 4.4: Online overview of players' statistics

Chapter 5

Pilot Testing

Once the design is complete and the mock-ups are ready, it's time to submit it to the user. During this process, it is important to ascertain any inconsistencies between design and user expectations. For this purpose, *usability testing* is available. After conducting this testing, the prototypes will be updated and I can design a data structure for the application, that is going to be used in implementation 6.

5.1 Usability Testing

In a usability test, one user at a time is shown something (whether it's a website, a prototype of a website, or some sketches of individual pages) and asked to either (a) *figure out what it is*, or (b) *try to use it to do a typical task* [4].

Usability testing is a very important stage in the development process and should be done as soon as possible. The earlier we show the user our ideas, the more resources we can save. It is recommended that three to four participants take part in each round of tests. Single participant will already find out about 30% of the usability problems and the other 2-3 will uncover the rest of them [1]. There is very little to be gained with more than five users [4]. Also, after the first round of testing is finished and the designs are improved, in the next round, the participants can focus on a new set of problems, because they won't be stuck on the former ones.

5.2 Pilot Testing Proposal

To get good quality feedback from coaches, I suggest to show to them the paper prototypes and directly observe their reactions. Because *nothing beats an honest reaction from the gut* [17]. Instead of using any of the recording technique such as video and lengthy analyze every nuance, I am going to focus on relatively simple methods. Moreover, these monitoring techniques are often very intrusive, which negatively affect user behavior. I am going to rely on a walkthrough through the designs, carefully observing users reactions. I am going to present to them the application and describe how it works, but not in a great detail. That's how I am going to find out, if the UI is self-evident. I am also going to leave a lot of space for questions and observe, which of the elements are taken notice of and which are not registered, suggesting their possible uselessness. At the same time, I am going to ask the user about specific elements and features and how would they use them.

5.3 Prototype Presentation

Within one week, I managed to conduct two iterations of testings with four coaches altogether. During the second iteration, I introduced improved prototypes based on the feedback from the first round of the testing.

The first meeting had positive reactions on the prototypes. The coaches agreed that it's something they would definitely use, and they liked the idea in general. However, during the course of the testing, there were found several shortcomings. Probably the most important one was the absence of aggregates during the online overview - information summarizing team averages give the coach feedback about their team without having to manually estimate them. There was also a request for an always available new session button, that was missing. Lastly, we discussed the importance of some measurement units such as sprint distance or relative HR. We agreed that the number of sprints and the sprint distance show more or less the same information and the screen is already too crowded, so I decided to remove number of sprints.

After the first round of the testing, I updated the prototypes according to the newly obtained requirements 5.1. Aggregate information characterizing the whole team was added on the bottom part of the online overview screen, and the new session button was added to the navbar. I also chose the most-desired visual representation of the information about a player.

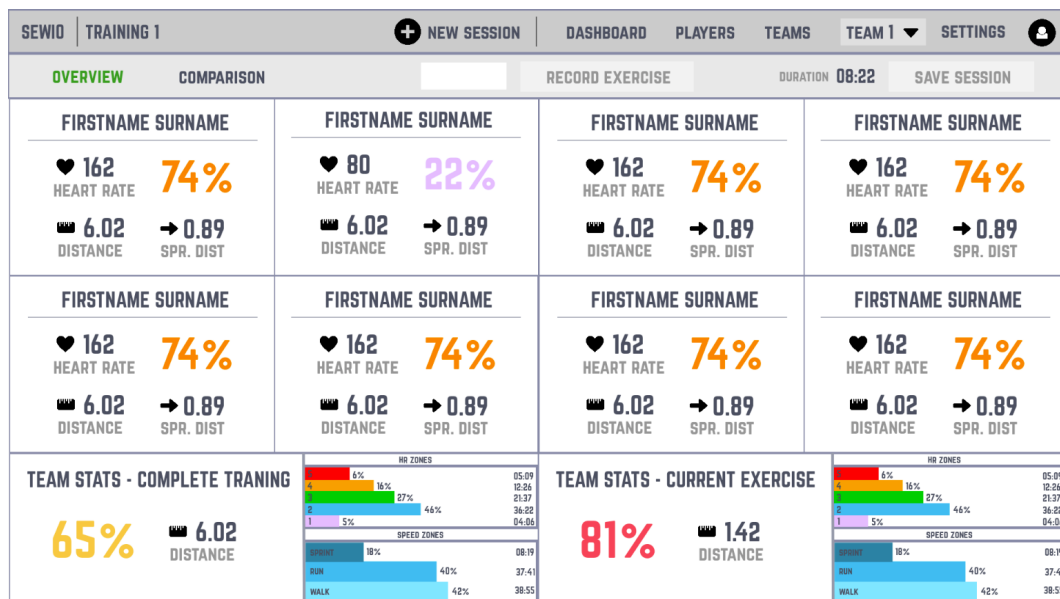


Figure 5.1: New corrected online overview of players' statistics

The second meeting had considerably better responses. The demanded requirements were met and the coaches could now focus on other set of problems or come up with more ideas. This happened after an unexpected visit of a sport nutrition expert, who came up with various propositions, how such data could be used to exactly monitor training load of players [21]. This led to brainstorming, which brought many ideas on how this application could be used in the future, and how many other benefits it could bring. I considered it more as a food for thought, rather than crucial features. On the other hand, sometimes

it's the detail (e.g. a smart aggregate function), which can make all the difference. Some of these features can be tested in the very final stage of the process after the application is fully implemented.

5.4 Data Structure

Dynamically changing data of the application must be stored somewhere and it must be connected in one logical unit. There are many diagrams and graphs that help us visualize the entire data structure. I decided to use *Entity Relationship* (ER) diagram¹. It is composed of entity types and specifies relationships existing between these entities. This diagram 5.2 was created based on the functionality of the application described in the previous paragraphs.

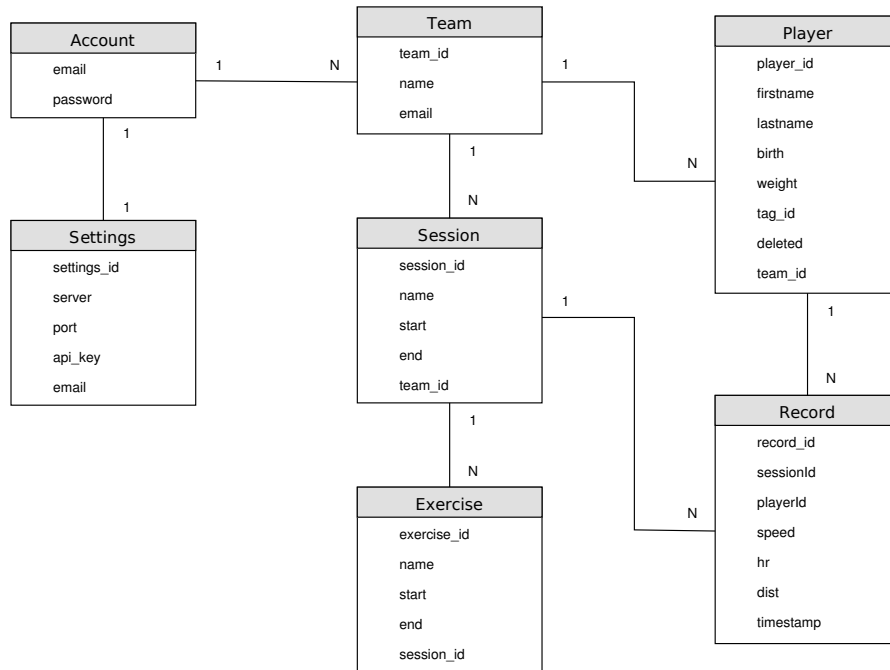


Figure 5.2: Data structure visualized with the use of ER diagram

- **Account** - each account has an email that is unique in the entire application and a password.
- **Settings** - there are specific settings for each account that can be permanently stored.
- **Team** - teams are identified by an identification number. Teams can be created, edited or deleted by a user, to whom the team belongs.
- **Player** - players exist in a given team and the user is able to create, edit or delete them. Each player, in addition to other attributes, corresponds with a *tag_id* - attribute, which assigns a particular tag to them.
- **Session** - recorded sessions, such as complete training.

¹<https://www.lucidchart.com/pages/er-diagrams>

- **Exercise** - recorded exercises during one session. For one session, any number of exercises can be recorded.
- **Record** - table linking individual sessions and the players, who participated in the session. It contains the most important data in the whole application - all the physical data needed to describe the player's activity during training. All other data, such as HR zones, can be aggregated, filtered and derived from this table, as later described in [6](#).

Chapter 6

Implementation

After the pilot-testing is conducted and the mock-up is in line with user's expectations, it's time to start to implement. This chapter describes web technologies and web design, and then the system architecture - web application and application server, as well as a visual appearance of the application and its overall functionality. In the end, testing of the application, results of those tests, possible future improvements of the system along with its evaluation are discussed.

6.1 Web Technologies and Web Design

Nowadays, there are many technologies and combinations of them suitable for web development. However, it is very important to choose the right ones to assure convenient implementation as well as the effective functionality of the application. The architecture of the application can be divided into two separate parts: *front-end* and *back-end*.

6.1.1 Front-end

Front-end, the „client-side“, is responsible for converting data to a graphical user interface, thereby presenting it to the user and enabling them to interact with it. The most popular front-end technologies are:

- HTML5¹ - standard markup language used for creating web pages.
- CSS3² - language describing the style of a HTML document and how HTML elements should be displayed.
- JavaScript³ - interpreted programming language used in web development.

It is possible to work with those in their „native“ form, however, it is way more effective to use some kind of framework, that will ease the process of front-end development. Currently, there are many frameworks, e.g. React⁴, Vue.js⁵ or Angular⁶, that would meet

¹<https://www.w3schools.com/html/default.asp>

²<https://www.w3schools.com/css/default.asp>

³<https://www.w3schools.com/js/default.asp>

⁴<https://reactjs.org/>

⁵<https://vuejs.org/>

⁶<https://angular.io/>

the purposes of this application. I chose Angular 5⁷, because Sewio has used it across multiple projects and it proved well. Moreover, it utilizes TypeScript⁸, which is a superset of JavaScript and provides us with type control, which will help me during the development. On top of that, there also exist other toolkits such as Bootstrap⁹, Foundation¹⁰ or Semantic UI¹¹, that ease up the process of web development by combining HTML, CSS and JavaScript into simple libraries. With the right choice of framework along with other support tools, it is possible to create websites more effectively.

6.1.2 Back-end

Back-end, the „server-side“, consists of a server, database, APIs and application. Similarly to front-end, there are many programming languages and frameworks, that are used for back-end development. Among the most popular languages and frameworks belong Node.js¹², Ruby on Rails¹³, Go¹⁴, or PHP, that has a lot of various frameworks to offer¹⁵.

Since the application is going to work with real-time data, one of the most important criteria for choosing the right language and framework is speed. Sewio used in one of their former projects PHP framework Slim¹⁶, but as I found out, the company ran into performance issues. Therefore, I chose to switch to Node.js, which is a JavaScript framework. According to benchmark tests¹⁷, Node.js performs better, than PHP. Also, it uses *event-driven, non-blocking I/O model*¹⁸, which is very efficient, for example, in parallel database queries. Combined with a package manager npm¹⁹, Node.js proves to be a very powerful, intuitive and effective framework to work with. To bundle all the assets together, I will use Webpack²⁰.

6.2 System Architecture

The application can be divided into two independent parts: web application and application server, corresponding with a *client-server* model solution.

The architecture is based on a *model-view-controller* (MVC) architectural pattern²¹, which is suitable for web application development. It comprises of three basic parts, that are linked together, as demonstrated on 6.1.

- **Model** - manages the data of the application
- **View** - is responsible for visual representation of the view

⁷<https://blog.angular.io/version-5-0-0-of-angular-now-available-37e414935ced>

⁸<https://www.typescriptlang.org/>

⁹<https://getbootstrap.com/>

¹⁰<https://foundation.zurb.com/>

¹¹<https://semantic-ui.com/>

¹²<https://nodejs.org/en/>

¹³<http://rubyonrails.org/>

¹⁴<https://golang.org/>

¹⁵<https://www.dailyrazor.com/blog/best-php-frameworks/>

¹⁶<https://www.slimframework.com/>

¹⁷<http://www.hostingadvice.com/blog/comparing-node-js-vs-php-performance/>

¹⁸<https://nodejs.org/en/docs/guides/blocking-vs-non-blocking/>

¹⁹<https://www.npmjs.com/>

²⁰<https://webpack.js.org/>

²¹<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

- **Controller** - controls interactions between the model and view

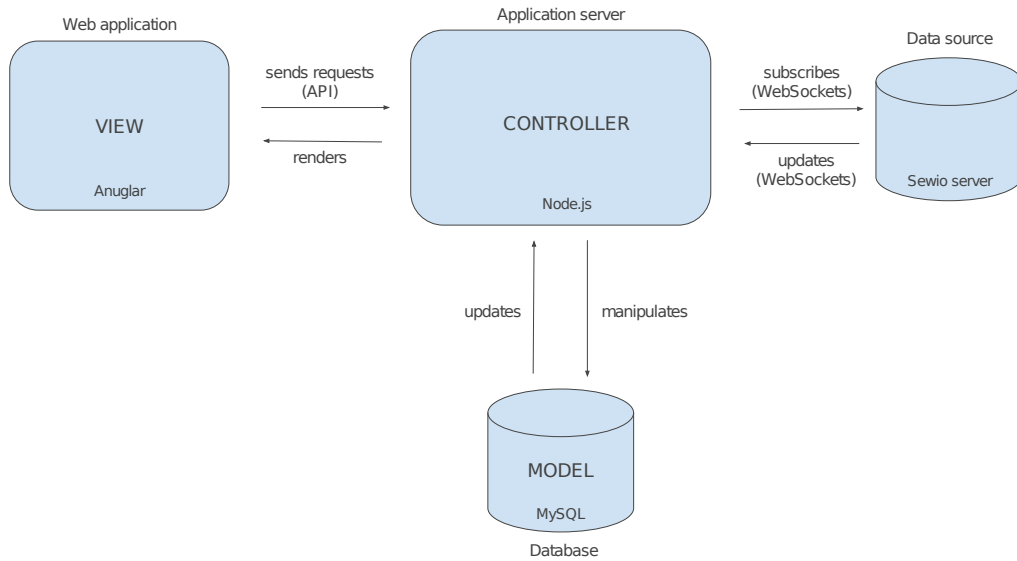


Figure 6.1: System architecture

6.2.1 Data source

The system receives data in real time (if session recording is actively used) from an external data source. In the real usage of the application, the data source is a server from Sewio located in a sports hall that collects data from the tags. For development purposes of the application, its testing and for ensuring its self-sufficiency in the future, a simulator of the server *simulator.js* was developed. This simulator only offers a subset of functions of the real server:

- **Handles multiple client connections** - multiple clients, which are in this case instances of the web application, can connect simultaneously to a given tag via a communication protocol WebSockets²². On connection, in the body of the sent message, the client (the application server) incorporates a *tag_id*, identifying a given tag. If the tag doesn't exist or it's turned off, the simulator sends an error message. Otherwise, it periodically starts sending data, corresponding to the specified *tag_id*, in a JSON format to the application server. The period of sending new data varies, as it depends on many factors, such as tag settings or server load. In sports, the timespan between two sets of data is generally somewhere in between 100 - 400 ms.
- **Provides a mock-up set of data** - mock-up data, that has been partially derived from a real set of data, and that credibly resembles movement of players during a training. The set of mock-up data is then sent in a loop, as it contains only a few minutes of session sample. This turned out to be fully sufficient for simulation purposes.

²²https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

6.2.2 Application server

Application server *server.js* is the core of the system linking all the components together. It comprises of many smaller components, each composing of multiple functions responsible for various tasks of the system. These tasks can be divided into these basic groups:

- **Database manipulation** - this functionality allows the system to create, read, update, or delete (CRUD²³) information from the database. Typically, as data changes the database, it is subsequently retrieved over the model and then passed to the display component so that the status of the application is updated.
- **Data computation** - these functions either calculate metrics such as total distance covered or current speed data before it is stored in the database or after data is retrieved from it and needs to be further processed, before being further passed to the view - for instance average team HR or the time spent in speed zones for an individual player during a given exercise.
- **Other functions** - a set of functions responsible for other minor tasks, such as active monitoring of a connection to the data source, determination of a state of the application, establishing or closing a connection.

These functions are then part of the components, for example:

- **OnlineController** - its responsibility is to manage the entire session recording. It maintains active connections to the data source via WebSockets, handles multiple client connections and preserves state of each of them with the usage of **Session-Connection** class instances, computes data and stores them into a database. Therefore, it's the most complex controller in the application server.
- **EventsController** - its task is to return post-match statistics of a session. For this purpose, it utilizes many data computation functions, which together return one JSON object containing all relevant data of the event.
- **TeamsController** - responsible for anything related to teams, such as new team creating, editing of an existing team or deletion of a team. It also retrieves information about a single team, or about all of existing teams for a given account.
- **PlayersController** - similarly to the **TeamsController**, the controller also comprises of functions for database manipulation, used for a single, or a multiple players management.

There are more controllers in total, such as **LoginController** or **RegistrationController**, that are responsible for authentication and registration of users. What all of them have in common is that they control interaction between the database and the web application.

²³https://en.wikipedia.org/wiki/Create,_read,_update_and_delete

```

{
  "body":{
    "id":"12",
    "datastreams":[
      {
        "id":"posX",
        "current_value":"24.556420421554",
        "at":"2018-03-16 11:38:30.063210"
      },
      {
        "id":"posY",
        "current_value":"8.0526151107857",
        "at":"2018-03-16 11:38:30.063210"
      },
      {
        "id":"posZ",
        "current_value":"0.64529517502524",
        "at":"2018-03-16 11:38:30.063210"
      },
      {
        "id":"hr",
        "current_value":"70",
        "at":"2018-03-16 11:38:30.063210"
      },
      {
        "id":"batLevel",
        "current_value":"90%",
        "at":"2018-03-16 11:38:30.063210"
      }
    ]
  },
  "resource":"/feeds/12"
}

```

Figure 6.2: JSON object payload for one tag

After the application server connects to the data source, it extracts and computes the received data. The computational process is implemented in the `calcData()` function, transforming the raw data from the received JSON object 6.2 into the following metrics (for each connected tag): current speed, current HR and total distance covered. Total distance and current speed are computed with the use of basic mathematical and physics equations. Current HR is a simple `current_value` extraction from the current JSON object payload corresponding to the `„id“ : „hr“` key. The following is an example of a part of the function:

```

// onlineController.js; calcData() function extract
this.t2 = moment(data.body.datastreams[0].at);
...
// calculate distance between two positions A1 <---> A2
let x1_x2 = Math.sqrt(Math.pow((this.x1 - this.x2), 2) +
    Math.pow((this.y1 - this.y2), 2));

// calculate time difference between two positions A1 <---> A2
let t1_t2 = this.t2 - this.t1; // this.t1 initial value set on start

this.distance += x1_x2;
this.speed = x1_x2 / t1_t2;

// save the current point as the previous one for the next iteration
this.x1 = this.x2;
this.y1 = this.y2;
this.t1 = this.t2;

// extract the current HR from the JSON object
this.hr = parseInt(data.body.datastreams[3].current_value);

```

The calculated data is continually being saved in the application server in attributes of `SessionConnection` class instances, and on a continual basis stored in a database. In this case, MySQL database was used because of its simplicity and sufficiency for the system. The data is stored every second with the use of `saveSessionData()` method to ensure that no information is lost if the application server fails. One second storage intervals were identified by the coaches as a sufficient rate to guarantee desired precision of the this system. If the intervals were bigger, the data would lost their accuracy. On the other hand, too frequent saving turns out to be merely a waste of resources, because fluctuations in players' physical data aren't so large²⁴.

When the application is running, it requires data from the application server so it can be displayed to the user. Communication between the application server and the web application (client) is facilitated by the *REST application programming interface* (API) architecture²⁵. The client makes an API call and the router on the server side (file `index.js` in the `router` folder) decides, what action should be taken - such as saving a new exercise, turning off a current recording, or requesting the player's physical data in that session. In the latter case, the data is retrieved from the database, possibly aggregated and calculated (for example, time in HR zones is calculated) and then sent to the web application as a JSON object. It can then finally update the view, hereby presenting the data to the user.

6.2.3 Web application

Not only does the web application provide a view, but it also partially performs on the controller and model parts of the entire system.

²⁴The players would need to accelerate and decelerate so fast, that it would be humanly impossible to do so to radically influence the computed data. The same applies to HR frequency.

²⁵<http://www.restapitutorial.com/>

As demonstrated on 6.3, the client communicates with the application server through *services*. Services are modules that contain sets of functions that define API calls. Here is an example of `getTeams()` function:

```
// data.service.ts; get all teams
getTeams(): Observable<Team[]> {
  return this._http.get("/api/teams", this.options)
    .map((response: Response) => <any>response.json().data);
}
```

The application uses multiple services such as *data.service.ts* or *authentication.service.ts*, which divide the functions according to their usage. They can be then *injected* into controllers²⁶. Controllers control the data, which is defined by the model. Controllers define components that are the basic building blocks of an UI in an Angular application. Together with templates, they manage what the user sees or can do. Components play the part of the controller/viewmodel, and the templates, with the use of *.html* and *.css* files, represent the view.

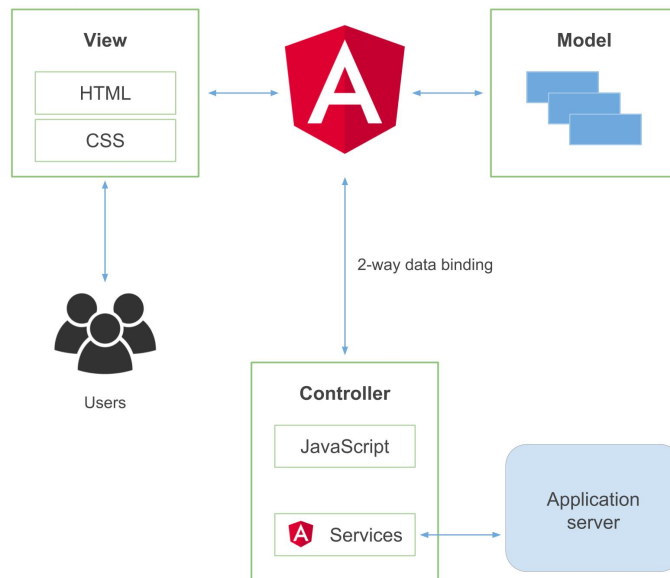


Figure 6.3: Angular MVC architecture

2-way data binding ensures continuous synchronization between the model and the view. The client also uses its own routing table, defined in the *app.routing.ts* file, to render different components depending on the current url. If the entered url doesn't match any of the routes specified, the default fallback route is set to `HomeComponent`, which represents the application's home page.

²⁶Through so called dependency injection.

6.3 Data Models and Database

Each data model corresponds to a table from the data model (ER diagram). Due to the architecture of the system, there are models defined in both the application server and the web application.

The application server defines the models in a */models* directory, for example */models/team.js*. These files are then imported to the controllers. For illustration, this is an example of the *team* model export and import:

```
// team.js; exports a constructor function representing a team
team = function(team_id, name, email) {
  this.team_id = team_id;
  this.name = name;
  this.email = email;
};
module.exports = team;

...
// teamsController.js; imports the exported constructor function
const _team = require('../models/team');
```

MySQL database manipulation is carried out with the help of *mysql* library²⁷. The library allows to connect to a MySQL database and perform queries. Such a query may look like this²⁸:

```
// eventsController.js
let sql = 'SELECT DISTINCT playerId from session_player
          WHERE sessionId = ?';
con.query(sql, sessionId, (err, rows) => {
  if (err) {
    reject();
  } else {
    ...
    resolve();
  }
});
```

In the web application, the models are defined in the *data-model.ts* file. Analogously to the application server, the model must be firstly exported from the *data-model.ts* file and then imported to the controllers as needed:

²⁷<https://www.npmjs.com/package/mysql>

²⁸This query example partly uses Promises https://developer.mozilla.org/cs/docs/Web/JavaScript/Reference/Global_Objects/Promise

```

// data-model.ts
export class Team {
  team_id: number;
  name: string;
  email: string
}

// ../teams/teams.component.ts
import { Team } from '../data-model';

...
private teams: Team[]; // all teams

```

6.4 Visual Design and Functionality

In line with the principles of the design (described in the chapters 2 and 4), together with the mock-ups (presented in 5), I created a simple UI that allows the coaches to use the sports application effectively. I named the application „Sporterizer“.

6.4.1 General Appearance

To make Sporterizer visually appealing, I took advantage of CSS3 in combination with the Bootstrap and ng-bootstrap frameworks. All views (apart from the home, login and registration pages) incorporate a navbar. The green color of the navbar matches with other products and software developed by Sewio. It was very important to enhance the feeling of using a sort of sports application, but also maintain the „clean“ look of it. This can be done, for example, by color adjustment [22] or font change [23]. That’s why I decided to use the white background color and sans-serif font Oswald, which together have the intended psychological effect. Icons in the application are used from Font Awesome toolkit²⁹.

6.4.2 Functionality

After the coaches register and log into the application, they are presented with the dashboard 6.4 implemented in the `DashboardComponent`. Before the component gets the data to display (events, last session data) through services, as described in 6.2.3, it displays a loading icon, so the user is never presented with a blank screen and does not know what is going on. This functionality applies across all components, thereby increasing the user-friendliness of Sporterizer.

The calendar by default shows events that were recorded during the current day. Time (start of training) and name clearly identify a given event. It is also possible to view any given month, week or day. An event can be selected by either clicking on the red dot or the „time event-name“ text. This solution makes the calendar both fast and easy to interact with - which is exactly what the coaches need.

The right part of the dashboard screen features last event overview. HR zones and speed zones are one of the key UI elements of the whole application. In this case, they display team averages of the complete training. HR zones display five different zones that represent HR ranges in which a sportsman can be [24]. The higher span of values the zone represents,

²⁹<https://fontawesome.com/>

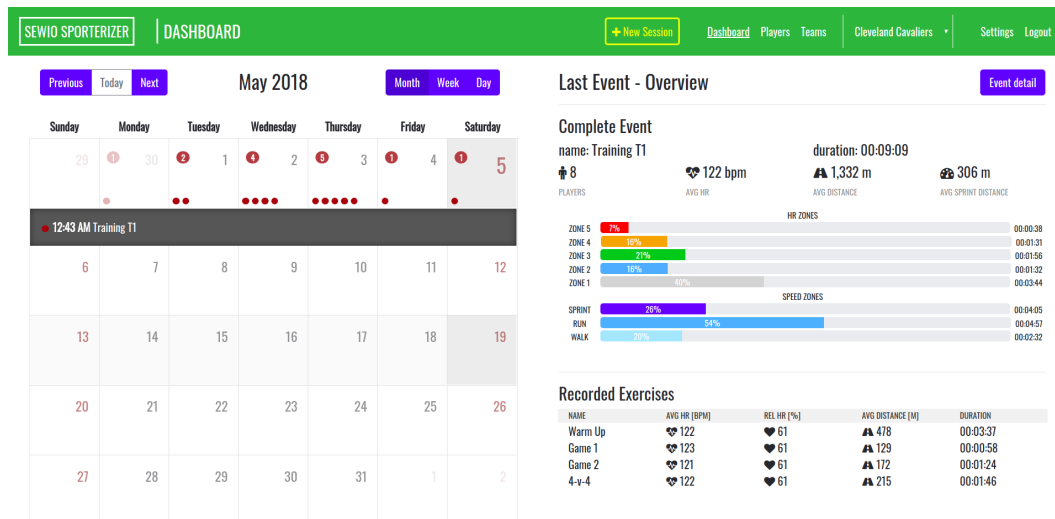


Figure 6.4: Dashboard

the more „vigorous“ color it has - from light gray showing very little of importance, across green meaning the optimal activity, all the way to red marking very intensive performance, that should be taken notice of. Speed zones function analogously. The UI elements were implemented as a variation of ng-bootstrap’s HTML element `<ngb-progressbar>`. Tags that are offline during the session recording are taken into account and don’t skew any of the aggregated data - neither the whole session nor exercise summaries are affected.

The new session button was designed to be the most prominent element inside the navbar. Its yellow border and text color contrast well on the green background and when the user hovers over it, the whole buttons turns yellow and becomes even more forward. When a session is currently being recorded, the button turns red and changes its text to „Recording“ - this allows the coach to clearly see, that the recording is active, thus also preventing them to accidentally want to start simultaneously recording another session (which the system wouldn’t handle). After clicking on the button, the `NewSessionComponent` is rendered. While choosing the players to record, tags that are online were designed in green color, whereas those offline in red, so the coach can quickly see their status. Unless name for the new session is entered and at least one player is chosen, there will be a red warning message displayed, clearly notifying the user, that they forgot something. Afterwards, new session is created and the view of `OnlineComponent` 6.5 is displayed.

The view is regularly updated with current data every three seconds. From a practical point of view, the frequency of data update is sufficient and does not need to be higher, because the coaches don’t need to see the current data immediately³⁰. API calls last between 50 - 400 ms before fetching new data depending on the number of selected players³¹. This

³⁰Even a few seconds delay is perfectly fine because the coaches are more interested in aggregate numbers and the information whether a player has sprinted 524m or 528 m makes practically no difference during online recording.

³¹These numbers were obtained during testing of the application on one computer running both server, simulator and the application.

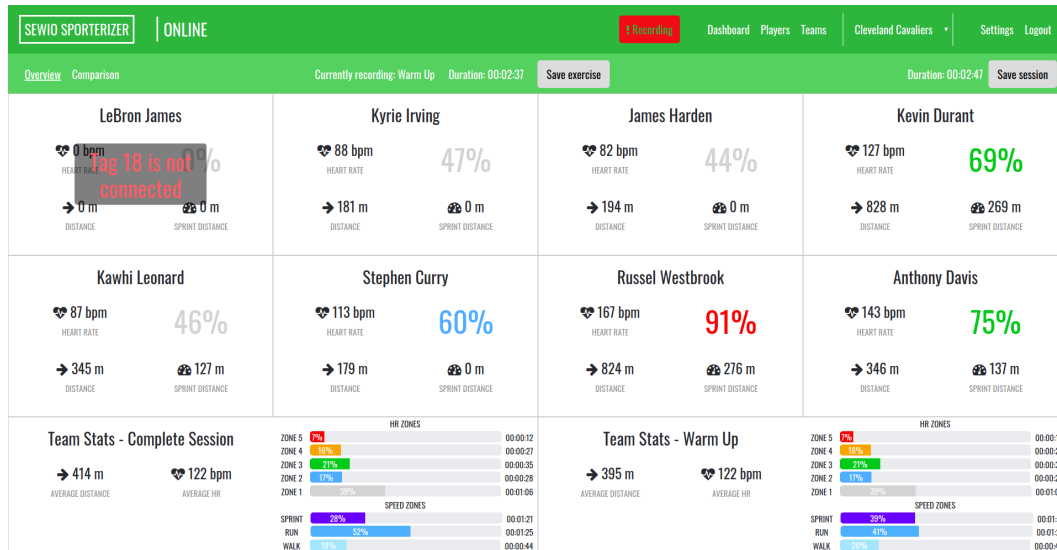


Figure 6.5: Online overview

is done using the RxJs library³². An *observer* in the controller subscribes to an Observable `getSessionData` (defined in `data.service.ts`) and the observer reacts to the data that the Observable emits. On this event, the model as well as the view are updated and the user is presented with new data. Subscription frequency is moderated with the use of `TimerObservable` and `takeWhile` operators.

The coach can during the recording also notice which tags may be offline - marked with a red text warning of the top of the player „card“. They can clearly see the current HR value relative to the maximum HR value of the player, calculated as $220 - age$ [25]. This is the most important metric that a coach has at his disposal during recording. Therefore, the numbers are larger and distinguished by different colors that correspond to the colors in HR zones, which makes the application more comfortable to work with. The lengths, as well as the buttons of the currently recorded exercise and the whole session, were designed in neutral white and gray, so they are sufficiently visible on the light-green color of the navbar, but don't draw too much attention from the more important elements.

While recording is active, it is possible to leave the current view or even exit the application. The recording will stay active as long as the user doesn't save it, or it will continue to capture the data for another three hours. This is implemented on the application server as a timeout (`closeSessionTimeout()`). It is also possible to browse through the application - view the status of tags in the settings, analyze already recorded events, etc. However, some functionality such as teams, players and server modification is for the sake of application's consistency disabled. When the user visits the views containing this functionality, they are presented with a warning message.

Once the recording is complete, the observer unsubscribes from the observer, the session is saved and the user is presented with Dashboard. `PlayersComponent` and `TeamsComponent` allow the coaches to add, edit and delete players and teams. Information about the entities

³²<http://reactivex.io/rxjs/>

is presented using *reactive forms*³³. This solution offers a better form validation of the entered data and comes across as more user-friendly. Also, as described in 2, there should be some feedback from the application on any action the user is doing. Therefore, when any of the forms in Sporterizer is submitted, an informative message pops up, describing result of the executed action. Finally when the coaches log out of the application, they are redirected to the `LoginComponent` and presented with a message informing them about a successful logout, which is prominently displayed in green. All these types of notifiers throughout the application are designed in the same way (except for colors - green for success and red for warning or failure) to preserve a certain visual unity.

6.4.3 Application Security

Security on the authentication level is ensured with the use of JSON Web Token³⁴. The standard defines an independent way for secure transmission of information between parties as a JSON object. This information is digitally signed on the application server with the use of a RSA key, therefore it can be trusted. Implemented in the *authentication.service* and *auth.guard.ts*, it prevents unauthorized users to access pages of the application visible only for those that are logged in. On login, the user's credentials are posted to the API and the response is checked for a valid JWT token. On success, the authentication is successful and an updated expiration timestamp is stored in the `localStorage`³⁵. In this way, until the user logs out, they stay logged in, even if the browser is refreshed or browser sessions are changed.

6.5 Testing

The last part of the design thinking process is testing. Now the user decides if they like the implemented application and provides the designer with valuable feedback, that will help them to improve it.

6.5.1 Testing proposal

The optimal case would be to firstly test overall functionality of the application (such as tag connection, modules functionality) in the Baluo application center in Olomouc³⁶, where Sewio has their locating system set up, and subsequently test the application with coaches during the training. However, due to currently not yet fully functional³⁷ system, that was not possible. Therefore, I had to abandon this process and rely on the data server simulator, that I created. This allowed me at least to explore usability of the application itself and intuitiveness of the UI.

As Krug recommends in his book, *the best way to learn how to make anything more usable is to watch people actually try to use it* [4]. Based on this idea, I suggest to directly observe the coaches use the application. I am going to present them with a set of tasks to be done, and with the use of *think aloud protocol* [26] find out what their thought process is. I will also measure total execution time of the set of tasks and see how fast the coaches

³³<https://angular.io/guide/reactive-forms>

³⁴<https://jwt.io>

³⁵<https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>

³⁶<https://www.acbaluo.cz/>

³⁷The system should have been ready for the testing, but Sewio had some technical issues with implementation of HR measurement.

are going to complete it. Because this type of application will be used regularly, I assume that the first interaction with the system will be slower, as the user is going to be faced with a lot of novelty. This is why they will perform a complete set of tasks twice - the second iteration should be considerably faster because the user will be more familiar with the system. Finally, once the set of the tasks is done, I will ask the users about their impressions. The following is the set of prepared tasks:

- Log into the application (email: lorem@ipsum.dolor, password: secret)
- Add a new player (firstname: Dwayne, lastname: Wade, year: 1982, month: April, day: 13, weight: 106 kg, tag: 12)
- Record a new session (name: Test 1). Add at least five players
- Record an exercise (name: Warm up)
- Compare players and filter them by current HR
- Save the exercise and the session
- Find out who covered the most distance during the „Warm up“ exercise in the „Test 1“ event
- Log out of the application
- Execute the tasks in the list one more time (you are going to do the list twice in total)

6.5.2 Testing results

In the course of one week, I managed to meet with four professional coaches and a few other people who have a good knowledge of sports training. The testing took place informally so the testers felt comfortable and without pressure. As I gained some experience during the research part of the thesis [3](#), the interaction with users was considerably better.

Following the procedure above, I received many positive comments on the application, but also many ideas and recommendations on what could be improved. All of the tested users liked the UI of the application and said that it was visually very appealing. They appreciated the functionality and possibilities of the system.

Just as I expected, the first time, when the coaches completed the set of tasks, was way slower than the second time. However, it was mostly during the first iteration, when I discovered most of the shortcomings of the application - the users were struggling with recording a new exercise and finding out how to compare players. Thanks to the think aloud protocol as well as to user impressions (conducted after the testing of the set of tasks), I found out that it was because of the green color of the navbar combined with „gray“ text. Some of the coaches reported that for the first time, when they saw the application, there was too much new information for them. This implies that the application's structure or at least its design could be improved. However, the big time difference in completion of the sets of tasks was also caused by the fact, that the users weren't used to the keyboard of the laptop, on which they were tested. Especially typing of numbers was critical and it got significantly better during the second round of the tasks execution. There is one more factor why it could be more difficult to complete the tasks - English is none of the tester's native language and some of them don't speak it fluently, therefore it took them more time to figure out, where to look for certain things.

There were many comments on what could be improved, such as having a possibility to add or delete all players at once at the start of a new session. There was a demand for having the *tag_id* displayed next to the player's name during the online recording. As mentioned above, there was a need for better navbar design, so the buttons and texts on it are more readable. One coach said, that some of the lines, which are dividing different information, such as in the `EventsComponent` or `OnlineComponent` could be bolder to enhance quicker visual navigation throughout the information. Another pointed out that having to wait for the application to render a new screen for longer than two seconds could be really frustrating, so it's really important to keep the application as fast as possible. There was also an idea to add different colors to the metrics in comparison view (both online and offline) to improve their readability. Finally, one coach thought that it would be great to have a possibility to choose what metrics to display during the online view.

Based on the feedback, I fixed some of the issues and left the rest for possible improvements in the future, as described in the next part.

6.6 Future of the Application

After testing I visited Sewio and presented to them a final version of the application. They were thrilled by it and said, that the result had exceeded their expectations. This was followed by the discussion about the possibility of testing and deploying the application in the Baluo application center, once the system is ready. Similarly to Sewio, I also presented my work in Baluo to some people in the center (including the director), and their reactions were really positive - we immediately started thinking about deploying the application, its possible extensions and market place.

In the future, it is possible to add many new functions, modules and metrics to the current application. As I described in [3](#), the coaches had countless ideas about what such an application could do - everything from recording sessions, profound in-depth post-analysis to eventually using the application in a real match and linking it with another system to broadcast the data in real-time on TV.

Therefore, it is important to first find the target audience, identify a unique selling point of the application, and so focus it on a niche segment of the market. Only then will it be clearer, what functions exactly should the final product have.

One option is to take the current application and use it as it is. The second is to use the know-how, that I have acquired during my work on this thesis and build a new application from scratch, that will be better targeted. Both of the possibilities are currently in discussion with Sewio and Baluo. The latest idea was to create an application for coaches, parents and children and use it to monitor training load of the children and be able to report to their parents, if their child is training healthy.

Chapter 7

Conclusion

The main goal of the thesis was to develop a web application for modern sports training, that utilizes data from a real-time locating system. While working on the project I tried to find out what data and functionality the application should offer, so the sports coaches who use it are maximally satisfied and will always want to have this tool in trainings.

The introductory part deals with the user interface and its design. After learning what design thinking is and what stages it comprises of, I had a clear idea of the complete process of the application development.

Acquainted with the knowledge of building UIs, I started studying research methods, that would enable me to identify the needs of coaches. Simultaneously, I studied the real-time locating system used by Sewio. After learning how the system collects, manages and pre-processes data, I could finally meet up with the coaches, understand their needs and merge them with the capabilities of the system.

After meeting with five professional coaches, I aggregated their needs and identified the most useful application features. I learned about web technologies and designing web applications. I created an information structure based on the user requirements. I studied prototyping to later create mock-ups of the application, that helped me translate the aggregated information into a more tangible form, that I could present to the coaches. After several iterations of this process, the final version of the application was agreed on.

The result of the thesis is a system consisting of an application server that was implemented with Node.js and a web application programmed in Angular 5. For development and testing purposes, I also created a simulator simulating real-time motion and biometric data of players that would otherwise be obtained from a server located in a sports center. The application is divided into two key parts. The first is dedicated to online recording, where the coaches can see in real-time multiple metrics about each player, such as current heart rate or total distance covered. It is also possible to record exercises, compare players or check team averages. The second part is for offline post-analysis of the information that was recorded online. Other features include managing players, teams, or server settings.

The application was tested with four professional coaches, whose feedback was very positive. I also presented the final product to Sewio and to a few key people in Baluo application center. Both parties were thrilled about my work, which I was very happy about. Immediately after my presentation, there was a discussion about the future of the application, its possible improvements and market position. I really want to continue working on this product and it seems to me that I have full support in this.

Bibliography

- [1] DIX, A.; FINLAY, J.; ABOWD, G. D.; et al.: *Human-Computer Interaction*. Pearson Education Limited. 2004. ISBN 978-0-13-046109-4.
- [2] MCKAY, E.: *Ui is Communication: How to design intuitive, user-centered interfaces by focusing on effective communication*. Online. February 2014. [Online; accessed 26.02.2018].
Retrieved from: <https://www.slideshare.net/EverettMcKay/ui-is-communication-how-to-design-intuitive-usercentered-interfaces-by-focusing-on-effective-communication>
- [3] University Oxford Press: *User interface*. [Online; accessed 26.02.2018].
Retrieved from: https://en.oxforddictionaries.com/definition/user_interface
- [4] KRUG, S.: *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders. 2006. ISBN 978-0321344755.
- [5] NORMAN, D.: *The Design of Everyday Things*. Basic Books. 2013. ISBN 978-0-465-05065-9.
- [6] YOUNG, I.: *Practical Empathy: For Collaboration and Creativity in Your Work*. Rosenfeld Media. 2015. ISBN 978-1933820484.
- [7] NIELSEN, J.: *10 Usability Heuristics for User Interface Design*. January 1995. [Online; accessed 27.02.2018].
Retrieved from: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [8] DAM, R.; SIANG, T.: *5 Stages in the Design Thinking Process*. September 2017. [Online; accessed 02.03.2018].
Retrieved from: <https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process>
- [9] Design Council: *The Design Process: What is the Double Diamond?* [Online; accessed 02.03.2018].
Retrieved from: <https://www.designcouncil.org.uk/news-opinion/design-process-what-double-diamond>
- [10] Board of Innovation: *16 cognitive biases that can kill your decision making*. August 2017. [Online; accessed 02.03.2018].
Retrieved from: <https://www.boardofinnovation.com/blog/2017/08/02/16-cognitive-biases-that-kill-innovative-thinking>

- [11] KAHNEMAN, D.: *Thinking, Fast and Slow*. Farrar, Straus and Giroux. 2011. ISBN 978-0374275631.
- [12] BORDENS, K. S.; ABBOTT, B. B.: *Research Design and Methods: A Process Approach*. McGraw-Hill Education. 2013. ISBN 978-0078035456.
- [13] Atlas.ti: *Observational Research*. [Online; accessed 03.03.2018]. Retrieved from: <http://atlasti.com/observational-research/>
- [14] HOLMES, A.: *Direct Observation*. New York, NY: Springer New York. 2013. ISBN 978-1-4419-1698-3. pp. 980–981. doi:10.1007/978-1-4419-1698-3_1758. Retrieved from: https://doi.org/10.1007/978-1-4419-1698-3_1758
- [15] CARLSON, J.: *Indirect Observation: Characteristics, Advantages and Disadvantages*. September 2017. [Online; accessed 03.03.2018]. Retrieved from: <https://www.lifepersona.com/indirect-observation-characteristics-advantages-and-disadvantages>
- [16] University Oxford Press: *Pilot study*. [Online; accessed 05.03.2018]. Retrieved from: <https://www.aqr.org.uk/glossary/pilot-study>
- [17] MCCUE, M.: *How Google Ventures Uses Design Sprints to Ship Great Products*. February 2016. [Online; accessed 06.03.2018]. Retrieved from: <https://99u.adobe.com/articles/52707/how-google-ventures-uses-design-sprints-to-ship-great-products>
- [18] FRIEL, J.: *Aerobic Threshold Training*. January 2006. [Online; accessed 09.03.2018]. Retrieved from: <https://www.active.com/triathlon/articles/aerobic-threshold-training-882916>
- [19] SUN, A.: *6 common cognitive biases UXers should know*. November 2017. [Online; accessed 16.03.2018]. Retrieved from: <https://medium.muz.li/6-common-cognitive-biases-uxers-should-know-750b8c7af1a8>
- [20] Google: *Rapid Prototyping 1 of 3: Sketching & Paper Prototyping*. August 2016. [Online; accessed 17.03.2018]. Retrieved from: <https://www.youtube.com/watch?v=JMjozqJS44M>
- [21] Firstbeat: *Training Load*. [Online; accessed 20.03.2018]. Retrieved from: <https://www.firstbeat.com/en/consumer-feature/training-load/>
- [22] CHAPMAN, C.: *Color Theory for Designers, Part 1: The Meaning of Color*. January 2010. [Online; accessed 08.04.2018]. Retrieved from: <https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/>
- [23] CHO, M.: *The Psychology of Fonts (and how they make you feel)*. March 2014. [Online; accessed 08.04.2018]. Retrieved from: <https://crew.co/blog/the-psychology-of-fonts/>

- [24] Polar: *Running heart rate zones / The basics*. April 2016. [Online; accessed 08.04.2018].
Retrieved from:
<https://www.polar.com/blog/running-heart-rate-zones-basics/>
- [25] Polar: *How to calculate your maximum heart rate for running*. July 2016. [Online; accessed 08.04.2018].
Retrieved from:
<https://www.polar.com/blog/calculate-maximum-heart-rate-running/>
- [26] NIELSEN, J.: *Thinking Aloud: The #1 Usability Tool*. January 2012. [Online; accessed 23.04.2018].
Retrieved from:
<https://www.mngroup.com/articles/thinking-aloud-the-1-usability-tool/>

Appendix A

Interview Documents

Aplikace pro moderní sportovní trénink s využitím senzorických dat a pozičního systému

Otázky (vodítka)

20.11.2017, Olomouc Baluo

1. Jak výkonnost hráčů měříte dnes?
2. Uchovávejte si nějaké výkonnostní statistiky hráčů? Jaké a jakým způsobem?
3. Porovnávejte jednotlivé hráče mezi sebou?
4. Jaké fyzické metriky u hráčů jsou pro vás podstatné (rychlost, vytrvalost,..)? Na kterých pracujete (trénujete je), aby se v nich hráči zlepšili?
5. Jakým způsobem vyhodnocujete, že je trénink (cvičení) efektivní?
6. Sledujete statistiky a výkonnost hráčů během krátkodobých intervalů (týdny), nebo si přehledy a čísla uchovávejte pro sledování dlouhodobých trendů?
7. Co je při těchto úkonech neefektivní? Jaké funkce či nástroje by vám k tomu pomohly?
8. Co je asi nejpálčivější věc, kterou by nový systém nejvíce zlepšil?

Figure A.1: Interview questions

Aplikace pro moderní sportovní trénink s využitím senzorických dat a pozičního systému

Možnosti systému Sewio

20.11.2017, Olomouc Baluo

Co aplikace může umět?

ONLINE

- **Měření výkonnosti hráčů**
 - rychlost, uběhnutá vzdálenost, tepová frekvence, průměrná rychlost, klasifikace pohybu (chůze, běh, sprint), load balance, ...
- **Akvizice a anotace tréninku**
 - start / pozastavení / stop nahrávání
- **Pozice hráčů na hřišti**
 - Vykreslení pozice, vizualizace pohybu

OFFLINE

- **Porovnání částí tréninku**
 - jednotlivých hráčů a jejich výkonnosti (jednotlivé tréninky, časová období, dny, týdny, měsíce, ...)
- **Výkonnostní statistiky (historie tréninků)**
 - Výkony hráče za časová období, změny ve výkonnosti
- **Zobrazovací funkce**
 - heatmaps, zóny
- **Správa týmu**
 - Tréninkové skupiny, správa hráčů, tagů

Co aplikace neumí?

- Neřeší taktický trénink a pohyb balónu
- Nezná polohu míče
- Neumí počítat body
- Neumí počítat relativní pozici hráčů
- Nemí určena k reálným zápasům

Figure A.2: Sewio system options

Appendix B

CD Contents

- src
 - Source files for the application
- documents
 - Technical report
 - Video
 - Poster
- TRsrc
 - Source files for the technical report

Appendix C

Installation Manual

Since the system is implemented in Node.js and Angular 5, to run the application on a local computer, it is necessary to install some tools. Then specific commands must be run in order to install and start the application. The following are the steps for the installation:

1. Download and install Node.js¹, npm², Angular CLI³ and MySQL⁴ (in this order⁵).
2. Create a local MySQL database and populate it with the content of the enclosed `db.sql` file.
3. Change to the `src` folder (root folder of the application, where `package.json` file is located) and run the command `npm install` and then `ng build`.
4. Run `npm start` and visit `localhost:3000` to start the application.
5. Login with the test account - email: *lorem@ipsum.dolor*; password: *secret*. Or register and create a new account.

¹<https://nodejs.org/>

²<https://www.npmjs.com/>

³<https://cli.angular.io/>

⁴<https://www.mysql.com/>

⁵Detailed installation instructions of these tools can be found on their websites.