Master Thesis

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Analysis and implementation of the software for personal sports records

Bc. Barbora Rolandová

Supervisors: Ing. Pavel Šedek, Dr. ir. Femke De Backere
Field of study: Open Informatics
Subfield: Software Engineering
May 2019

ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	**Rolandová**	Jméno:	**Barbora**	Osobní číslo:	**419013**
Fakulta/ústav:	**Fakulta elektrotechnická**				
Katedra/ústav:	**Katedra počítačů**				
Studijní program:	**Otevřená informatika**				
Studijní obor:	**Softwarové inženýrství**				

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Analysis and implementation of the software for personal sports records

Název diplomové práce anglicky:

Analysis and implementation of the software for personal sports records

Pokyny pro vypracování:

Perform a collection of requirement specification for creating software for personal sports records and analysis based on the history of physical aktivity. Make a research of existing solutions. Find and compare existing software for similar purposes. Analyse available technologies that meet the specified requirements and selection of appropriate technology for implementation. Design an application architecture. Prepare determination of test scenarios. Program a solution in selected technology. Evaluate individual requirements according to their specification. Verify test scenarios

Seznam doporučené literatury:

[1] UML 2 and the Unified Process - Ila Neustadt, Jim Arlow
[2] Software testing: concepts and operations - Mili, Ali; Tchier, Fairouz

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Pavel Šedek, katedra softwarového inženýrství FIT

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

Datum zadání diplomové práce: **06.08.2018** Termín odevzdání diplomové práce: ____________

Platnost zadání diplomové práce: **30.09.2019**

Ing. Pavel Šedek podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomantka bere na vědomí, že je povinna vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

21/2/18 Datum převzetí zadání	Podpis studentky

Acknowledgement

First and foremost, I would like to thank my supervisors Ing. Pavel Šedek of the Faculty of Electrical Engineering at Czech Technical University in Prague and Dr. ir. Femke De Backere of the Internet Technology and Data Science Lab at Ghent University for their valuable guidance, time and advices in a different stage of the thesis research and development.

Last but not least, my thank you goes to my family and my close friends for their understanding and continued encouragement. I could not have done it without you.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

In Prague on 24th May 2019

....

Abstract

Physical activity has important benefits in maintaining good health, preventing chronic diseases and it has a positive impact on a mental health and an overall quality of the life. This diploma thesis deals with this issue. The objectives of the thesis are to understand a construct of the physical activity, to make a research of existing solutions on the market and to collect a requirement specification. Consequently, a design and implementation of the web application in a respect of the usability is provided. The proposed application encourages to perform a physical activity and it can potentially be further developed.

Keywords: Physical activity, Physical activity analysis, Physical activity diary apps, Web application, Usability, React, Redux, Spring Boot

Abstrakt

Fyzická aktivita plní niekoľko dôležitých benefitov pri udržiavaní dobrého zdravia, prevencii voči chronických ochoreniam a má pozitívny vplyv na duševné zdravie a celkovú kvalitu života. Táto diplomová práca sa zaoberá touto problematikou. Cieľom diplomovej práce je pochopiť koncept fyzickej aktivity ako takej, preskúmať existujúce riešenia na trhu a vytvoriť špecifikáciu požiadaviek. Súčasťou je návrh architektúry a implementácia webovej aplikácie s ohľadom na použiteľnosť. Navrhnutá aplikácia podporuje fyzickú aktivitu a je pripravená pre jej ďalšie rozšírenie.

Klíčová slova: Fyzická aktivita, Analýza fyzickej aktivity, Aplikácie pre zaznamenávanie fyzickej aktivity, Webová aplikácia, Použiteľnosť, React, Redux, Spring Boot

Contents

Figures

Tables

Source codes

List of Abbreviations

ACID *Atomicity, Consistency, Isolation, Durability*

API *Application Programming Interface*

App *Application*

BCT *Behaviour Change Technique*

BMI *Body Mass Index*

DI *Dependency Injection*

DOM *Document Object Model*

EE *Energy Expenditure*

HR *Heart Rate*

HTML *Hypertext Markup Language*

HTTP *Hypertext Transfer Protocol*

HQL *Hibernate Query Language*

IoC *Inversion of Control*

JPA *Java Persistence API*

JSON *JavaScript Object Notation*

JWT *JSON Web Token*

kcal *Kilocalories*

MEMS *Microelectromechanical Systems*

MET *Metabolic Equivalent*

MVC *Model-View-Controller*

NCDs *Noncommunicable Diseases*

PA *Physical Activity*

PAQs *Physical Activity Questionnaires*

REST *Representational State Transfer*

SoC *Separation of Concerns*

SQL *Structured Query Language*

UI *User Interface*

WHO *World Health Organization*

Chapter 1

Introduction

This chapter gives a general introduction to this master's thesis. Firstly, in Section 1.1 the problem statement is introduced. Next, Section 1.2 explains the goals of the thesis. Finally, in Section 1.3 the outline of this thesis is presented.

1.1 Problem statement

Physical inactivity is a leading risk factor for Non-Communicable Diseases **(NCDs)**, such as cardiovascular diseases, stroke, breast, colon cancer and diabetes and has a negative effect on mental health, quality of life and well-being [16] [30]. Globally, one in four adults and three in four adolescents, aged 11-17 years, are insufficient physically active. As a consequence of economic development, increased use of technology, changing patterns of transportation, urbanization and cultural values, levels of physical inactivity may be as high as 70% in some countries [30].

To prevent NCDs concerning physical activity **(PA)**, the World Health Organization **(WHO)** launched recommendations at population level. The recommendations are intended to three age groups.

- Children and adolescents, aged 5-17 years, should do at least 60 minutes of moderate to vigorous-intensity PA daily [28].

- Adults, aged 18-64 years, should do at least 150 minutes of moderate intensity PA throughout the week, or do at least 75 minutes of vigorous intensity PA throughout the week, or an equivalent combination of moderate and vigorous intensity activity [28].
- Elderly adults, aged 65 years and above, should do at least 150 minutes of moderate-intensity PA throughout the week, or do at least 75 minutes of vigorous-intensity aerobic PA throughout the week or an equivalent combination of moderate and vigorous intensity activity [28].

1.1.1 Levels of Intensity PA

Intensity is defined as a rate of the effort required to perform an activity. Prior exercise experience and a level of fitness of an individual is depended on the intensity of PA [32].

A Metabolic Equivalent **(MET)** is a unit to express the intensity of PA. 1 MET is defined as the energy cost of sitting quietly, and is equivalent to a caloric consumption of one kcal/kg/hour [32].

- **Light-Intensity PA:** (less than 3 METs)
  Requires standing up and moving around.
  Examples include: walking slowly, sitting using computer, fishing-sitting, darts, billiards and playing most musical instruments [32] [34].
- **Moderate-Intensity PA:** (approximately 3-6 METs)
  Requires a moderate amount of effort and marked acceleration of the HR.
  Examples include: brisk walking, household chores, gardening, ballroom dancing, swimming leisurely, bicycling on flat surface - light effort and tennis doubles [32] [34]
- **Vigorous-Intensity PA:** (greater than 6 METs)
  Requires a large amount of effort and causes rapid breathing and a substantial increase in HR.
  Examples include: running, hiking, moderate/hard swimming, bicycling on flat surface - moderate/fast effort, basketball game and tennis singles [32] [34].

Outcomes of longitudinal studies indicate, that PA is associated with improving health. Moreover PA has a positive long-term influence as a prevention of developing NCDs. Performing regular PA provides many other healthy benefits, including: enhanced posture and body conditioning, reduced body fat, enhanced mental wellbeing through reduction of stress and anxiety and promotes better sleep [43].

Based on these conclusions and findings described in the previous paragraphs, the idea, a software for personal sports records came up to the author of this master thesis from its sports background. The author used to represent an athletic sports club at primary school at all competitive levels. After a termination of an active and competitive athletic activity in that sports club, author has still continued as an active and passionate sportsman in different kind of sports. From the beginning of the joining and competition in that sports club the author had started to write a sports training diary in a paper form and has been continuing to write all kind of PAs up to the present time to have an overview over that.

1.2 Goals of the thesis

A main goal of the master thesis is creating a software tool, that is able to record personal physical activity and its analysis based on the history of that physical activity. Before implementing this software solution, it is important to understand principles of performed PA. Therefore, the software is developed based on a literature and market research. Following issues related to promote PA are studied: a construct of PA, levels of intensity PA and assessments for measuring PA. The set of the features, that encourages to perform PA is assembled and on the basis of that, an analysis and a comparison of available solutions on the market are described in detail. Based on that, functional and non-functional requirements are created. An architecture of the software solution is designed to be able to cover existing and potentially increasing requirements and to maintain scalability of the app. After that, the choice of appropriate technologies for an implementation is given. In the last phase, evaluating of the implemented application is provided and it is verified by creating test scenarios. The developed solution brings a summary of PAs, its analysis and a support for healthy lifestyle. The app is aimed mainly to an individualist, not for a team.

1.3 Outline of the thesis

The outline of the thesis is structured as follows. Firstly, Chapter 2 discusses the literature and market research related PA and existing applications promoting PA. Next, Chapter 3 consists of functional and non-functional requirements for the topic of this thesis. Chapter 4 gives an architectural overview of the developed software. Chapter 5 specifies the implementation of the given architecture. The testing and evaluation phase is discussed in Chapter 6. Finally, a general conclusion of this thesis is given in Chapter 7.

Chapter 2

Literature and market research

Before elaborating on the actual thesis work itself in the next chapters, the results of the literature and market research are presented in this chapter. Four sections can be distinguished. Firstly, in Section 2.1, a construct, definition and terminology related to physical activity is given. Next, Section 2.2 shows the assessments of physical activity. In Section 2.3, existing solutions are given. Finally, in Section 2.4 a summary of literature and market research is given.

2.1 Construct of physical activity

According to WHO, PA is defined as "any bodily movement produced by skeletal muscle that requires energy expenditure **(EE)**". EE is "the amount of energy, measured in kilocalories or kilojoules, that an individual uses in a given time. Briefly 24-hour expenditure can be divided into three components: basal metabolic rate, energy expenditure of PA or thermic effect of activity and the thermic effect of food" [29].
PA includes activities undertaken while walking, working, carrying out household chores, travelling or doing sports [31] [19]. The term PA should not be mistaken with "exercise", as this is a subcategory of PA that is planned, structured, repetitive, and purposive, in the sense that the improvement or maintenance of one or more components of physical fitness is the objective [30].
PA can be identified by duration (minutes or hours of PA), by frequency (number of days of PA) and by intensity of a specific PA, performed during a

time frame of PA such as past 24 hours, past week, past three months, past year or entire lifetime [2].

2.2 Assessment of physical activity

There are two main assessment tools to measure physical activity. The first assessment tool is based on subjective measures and is described in Section 2.2.1. The second assessment tool concerns objective measures provided by wearable devices, which is described in Section 2.2.2.

2.2.1 Self-report

Self-report measures of PA commonly include administrations of questionnaires, short PA logs and detailed PA diaries. It is an indirect approach that involves recording the activity by an user.

Physical activity questionnaires

Physical activity questionnaires **(PAQs)** are the most commonly used as a subjective assessment for PA surveillance activities and population-based cohort studies. They are completely retrospective. The respondents answer questions that require to recall the duration, frequency and intensities of PA in different domains, such as housework, leisure time, occupation, traveling, exercise, or a combination of the domains, performed in a time frame. The time frame of PA varies from the past 24 hours, past week, past three months, past year to entire lifetime. Three type of questionnaires are specified below [2]:

- **Global questionnaires** are basically short. They include brief one-to-four questions of PA that provide a classification of one's PA status as active or inactive. Global questionnaires require the least possible information to classify respondents.
  For example, the Behavioral Risk Factor Surveillance System questionnaire [6] uses only one question to determine that respondents are involved

or not involved in his or her leisure-time PA. The main advantage of global questionnaires lies in their simplicity and they are easy by administrated. The main disadvantage is inability to measure compliance with PA guidelines [2].

- **Short-term recall questionnaires** consist of seven-to-thirty questions that assess the duration, frequency and intensity of particular PA for the past week or month. The particular PA measured can be domain-specific activities, intensity specific, or age specific [2].
  For example, a long version of the International Physical Activity Questionnaire, collects information in five domains such as 1) job-related; 2) transportation; 3) housework, house maintenance, caring for family; 4) recreation, sport, leisure-time; 5) time spent sitting. This questionnaire consists of twenty-seven questions, and identifies to various PAs and sitting behaviours performed by adults during the past seven days. The main advantage of short-term recall questionnaires is to measure compliance with PA guidelines. The main disadvantage is a detailed recall of PA and an inability to average frequencies and durations by some respondents during the past week or month [9].

- **Quantitative history recall questionnaires** are the longest and most detailed PAQs. They commonly have sixty or more questions and assess the duration, frequency and intensity of numerous types of PA from one year to an entire lifetime. The main advantage of quantitative history recall questionnaires lies in important value when relating activity exposure to diseases with lengthy incubation periods, such as cancer or osteoporosis. The main disadvantage is to detailed recall PA from the past year or during a lifetime [25] [2].

PA logs

PA logs are checklists used to record specific types and duration of activities undertaken at the end of a single day or during the day in discrete time intervals, for example thirty minutes. Plenty of PA have been already coded and published in a Compendium of Physical Activities [4], where each activity is classified into a domain and is expressed in MET units. The main advantage of PA logs is the simplicity in checking a performed activity and to drop recalling the duration of PA. The main disadvantage is a maintaining PA log during the day or detailed recalling at the end of the day [2].

PA diaries

PA diaries provide detailed information regarding different aspects of PA, tracked from one day to multiple weeks. Records can contain information about domains (job-related, transportation, housework, leisure-time), duration, self-perceived and referenced intensities. PA diaries can be used to recognize the types of PA that a particular group of people performed. The particular group can incorporate unemployed adults, retirees, mothers with young children and in behaviour change studies to analyze adoption of PAs or help to conclude with sedentary behaviour. The main advantage of PA diaries is a in-depth information about each performed PA during a time interval. The main disadvantage is a detailed maintaing of PA during the day [2] [25].

Conclusion

The basic weakness of all subjective measures like PAQs, PA logs and PA diaries is connected to the accuracy of recall and bias in reporting [19]. All self-report instruments can contain some level of measurement error [3].

2.2.2 Objective measures

Objective measures of PA contain heart rate monitors, pedometers, accelerometers and armbands. The field of PA assessment is increasingly using wearable devices to directly measure various components of PA. Objectives measures do not rely on information provided by the person.

Heart rate monitors

Heart rate **(HR)** monitors are physiological indicator of PA and EE that record real-time data on the duration, frequency and intensity of PA. Devices can be worn as watches or on the chest as a strap. Heart rate monitoring is the most appropriate way to classify levels of PA [44].
The main advantages of HR monitors is that they are easy to use and can be used in some cases for water-based activities [42]. The main disadvantage is

that HR and EE do not share a linear relationship at rest and low-intensity or high-intenisty PA and that HR can be influenced by different factors, such as body position, caffeine, stress, age, body size, muscle mass, sex, cardiorespiratory system and fitness level [44]. Therefore, accuracy can be improved, if hear rate monitors is used in combination with an accelerometer [42].

Pedometers

Pedometers, considered MicroElectroMechanical systems **(MEMS)**, are small, cost-effective motion sensors that register steps taken during walking and running activities. They do not measure duration, frequency and intensity of PA. Devices can be worn on the wrist or hip. The recommendation for adults is 10,000 steps per day [19].
Strengths of pedometers are that they are low cost, non-invasive, easy to use and outcome data may be used to raise knowledge concerning PA-levels and consequently encourage sedentary or inactive persons to become more physically active. Weaknesses of pedometers reside in accuracy at slow speeds (less than 60m/min) that could lead to incorrect measures with elder adults. Pedometers have also been criticized for a manipulation in increasing the total number of recorded steps by shaking a device. As a result of accessibility of sub-standard pedometers, selection of the device for trustworthy measurement of steps should be thoroughly considered [19].

Accelerometers

During the past decades, accelerometers, also considered MEMS, have become more popular, because of their their accuracy, processing a large amounts of data and simplifying administration, especially in large studies. Accelerometers are motion sensors that record acceleration (counts) in real-time and detect movement in up to three orthogonal planes [44]. Acceleration is represented as the rate of change in velocity of an object with a time. Consequently, the duration, intensity and frequency of PA can be determined as a function of human movement and can be monitored over days, weeks and even longer. Accelerometers provide an outcome for movement in counts per unit time that is indicated as an epoch. Devices can be worn on the waist, hip, lower back, thigh and ankle.
Strengths of accelerometers are accuracy with PA and sedentary behaviour, large memory capacity, and they are considered to be non-invasive, adjustable and unobtrusive [42]. Weaknesses of accelerometers are high cost, require-

ments of technical expertise, need for specialized hardware, software and individual programming [44] and a lack of standard protocols for managing or reducing data and propietary algorithms. Some of accelerometers are inadequate to distinguish different types of activities like walking intensity, cycling, weight lifting or body position, such as sitting, lying or standing [7] [44].

Wristbands

Over the last few years, wristbands have become more popular. Various versions of wristbands exist on the market. They accumulate data from different sensors, such as accelerometers, galvanic skin response sensor and heat-based sensors, like heat flux sensor, skin temperature sensor and near-body temperature sensor. The combination of signals from these sensors allow to measure EE and monitor metabolic PA. Using two-fold measurement attitude like body temperature and motion,it is more sensitive to evaluate the EE for complex and non-ambulatory activities, like walking with carrying a heavy load. Accordingly, armbands are being now used for tasks of daily life or low- to moderate-intensity of PA, but are not suitable for higher intensity of PA [44].

Conclusion

PA is a multi-dimensional construct and there is no measure that can assess all aspects of PA. Based on many papers [19] [44] [2], a significant approach is to combine self-report assessment tools that have the potential to provide rich descriptive data with objective assessment tools, such as accelerometers, hear rate monitors, pedometers and armbands.

2.3 Existing solutions

Primarily, in Section 2.3.1 significant features of PA diary applications are explored. Then, in Section 2.3.2, a methodology for selecting PA applications is given. In Section 2.3.3 a study of applications and their features is discussed in detail. In Section 2.3.4, particular existing solutions are compared. Lastly, in Section 2.3.5, a conclusion is presented.

2.3.1 Features of diary applications

There are plenty software solutions existing on the market, that promote PA. In this thesis, a focus is largely aimed at PA diaries, that describe performed PAs more detailed. According to studies, a recommendation is to develop a software app, that respects key physical activity concepts and encourage PA applying by evidence-based features and health behaviour change theory **(BCT)** [11] [33] [38] [17]. Therefore, following particular features should be concluded in PA diaries to effectively encourage PA:

1. Adding a wide-range of type of PA that respects key physical activity concepts, including the domain of the PA (occupational, transportation, house-hold, leisure-time). Examples are running, bicycling, jumping, dancing, power lifting, yoga, tai chi or gardering [46] [38].
2. Collecting data from wearable devices, such as wristbands, HR sensors, belt sensors, shoe sensors or smart watches and synchronize with PA app. Once is an app paired with a wearable device, it can collect data, such as HR, sleep stages and its overall quality, number of steps, distanced traveled, calories burned, pace, elevation and these information synchronize with other PA apps [17].
3. Tabulating data, that contains total energy from workouts, total work-out data (in miles, speed, repetitions or laps), body mass index **(BMI)** output from personal height and weight or only body weight data and data of the sleep quality [17].
4. Doing periodic summaries and analyzes, based on PA that was performed, such as graphs, tables, statistics to track personal progress [38]
5. Goal-setting functionality, that helps with detailed planning and to set personal goals [12]
6. Create a personal training plan to help to outline a content of the planned PA [8].
7. Using social support by social networks, that can increase a personal motivation and result in friendly competition among strangers. There are many social networks that are using for sharing information, such as Facebook, Twitter, Instagram, Pinterest, YouTube or Google+ [23].
8. Providing tailored feedback when a person achieve some health goals. It can be 10 000 steps a day or 150 minutes of moderate-intensity PA [17].
9. User friendly interface [17].

2.3.2 Methodology

In this study, applications were evaluated based on the discussed features of PA diaries in Section 2.3.1. The focus was also aimed on apps, which support a wide range of PA, not just one type or small group of workouts, such as a gym workout. If an app contains nutrition, blood pressure or another kind of tracking, the focus was principally directed at the activity recording. In the first approach apps were identified through the Google Play Store, where apps were drawn from the free top-ranked list in "Health and Fitness" category. The second approach was exploring articles, regarding to studied apps, respecting PA guidelines and using BCT in Google Scholar, Google Search and PubMed. Keywords were based on Boolean logic operators and included AND or OR combinations of "physical activity", "physical activity diary", "apps", "best apps", "top ranked" and "2018". A list of apps was gathered on 3 November, 2018.

2.3.3 Clustering of apps

10 apps were downloaded to an Android smartphone or through the website, reviewed and they are discussed. There can be differences between features in the mobile apps and a websites, such as an activity log form. User interface, available features and functionality were explored.

Fitbit app <https://www.fitbit.com>

Fitbit is an app intended for Android, iOS and Windows 10, that primarily track exercises using GPS, food intake, weight, sleep and woman period. It is designed to work with Fitbit wearable devices. An activity log depends on the type of PA contains basic data, such as title, date, start time, duration, distance in imperial and metric units, estimation of burned calories and steps. As summary of PA data, including the number of steps, distance, floors and calories burned are provided for a time frame, such as the current day, the past week, the past month and the past year and it is possible to customize it. Weight log contains information about actual weight and body fat and is depicted in a graph for a time span of a week, month or year. For logging sleep, it is required to connect with a tracker, that supports sleep tracking. The food log affords data about food and water consumed. It counts daily macro-nutrients breakdown in addition to your caloric intake. The app contains food databases for several countries and it supports a barcode scanning of food. It is available to set weekly goals and there is an overall 7 day summary, that tracks progress towards set the goal, based on activity in the past week, such as number of steps, number of exercise days, weight change or water consumption per day. Goals can be personalized. There is a social support from other Fitbit users and the possibility to join several groups that strive to achieve some goal. Synchronization with many top-ranked PA apps like Strava, MyFitnessPal, Lose It! or Runkeeper is feasible. The advantages are possibility to create a custom activity, if some activity is missing in the list. It is the app, that tracks not only PA, but also food, sleep, weight and woman period and therefore a user can gain an overview about own healthy status. The disadvantages are an inability to edit an activity log from activity history, a deletion of activity log is possible only. Synchronization is possible only with FitBit wearable devices. Creating a training plan is missing.

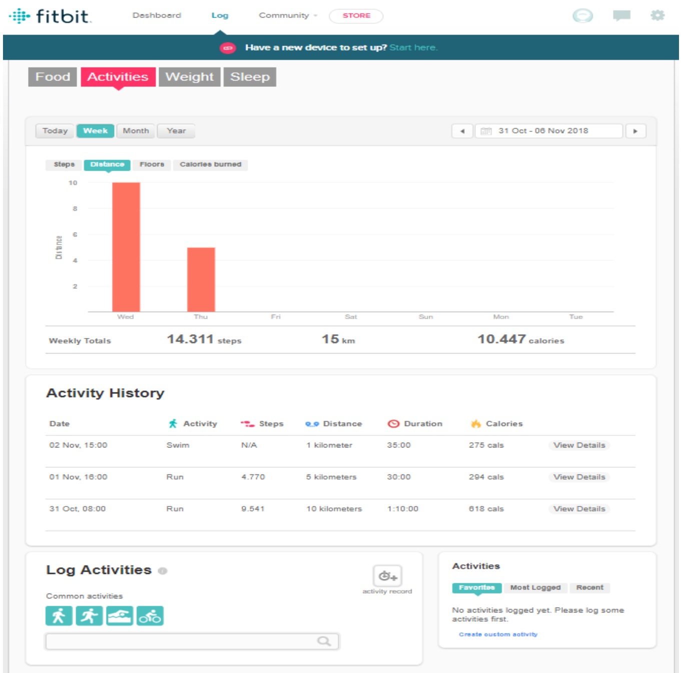

Figure 2.1: Fitbit website for logging activities

Fitocracy <https://www.fitocracy.com>

Fitocracy runs on Android and iPhone smartphones and contains multiple activities to be tracked. The activity log contains basic and advanced data, that is generated depending on an activity type. For example, running has basic parameters, such as name, date, duration and distance and advanced parameters containing pace, average HR, terrain and pack weight. Units of parameters can be customized. Each workout can be shared via Twitter, Facebook, Tumblr and the Fitocracy community. A picture of the activity can be uploaded. There is an option to logs multiple activities at once. The app use elements of gamification and for an activity is possible to be rewarded with points or badges. There are challenges, quests or duels, that user can participate in. Synchronization with other apps, like UP by Jawbone and Runkeeper is possible only. The primary advantages are a group workout routines and tracking a wide spectrum of PA activities. An expert virtual coach is available in a pro version to help with personalized training plans, tailored nutrition plans and a professional support to help achieve personal goals. The primary disadvantages are an inability to synchronize with wearable devices. A performance analysis is simple and it is expressed by a number of distance and duration for all records, maximum for the work out or a personal record over time for a selected activity.

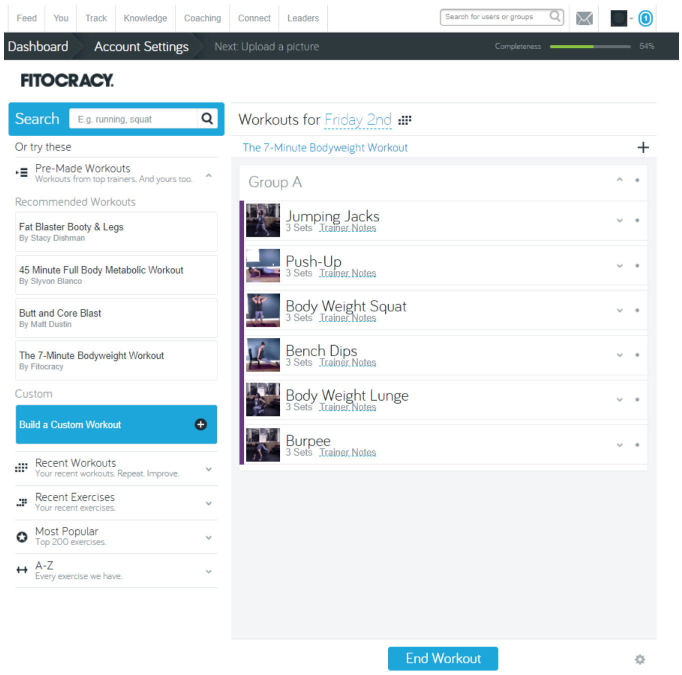

Figure 2.2: Fitocracy website for logging activities

Google Fit <https://fit.google.com>

Google Fit is an app for Android and iPhone smartphones and it is able to record an activity, weight and blood pressure. An activity log contains elementary fields, such as title of PA, date, start time and end time and optional fields, such as notes, calories, steps and distance. There is an automatic calculation of duration from start time and end time, steps, calories burned and pace. The activity log also contains information about move minutes and heart points that can be set as a personal goal. These two activity goals are based on the American Heart Association and WHO's activity recommendations. Move minutes is counted as each minute with thirty or more steps. Heart points define the intensity of PA. The app rewards one point for each minute of the moderate activity and two points for the vigorous activity. It is possible to track a workout by GPS. A weight log or blood pressure log contains the same information, like a weight or blood value, date and time of the measurement. The summary of an activity shows the number of move minutes, heart points, steps, distance, calories for the past day, past week or past month. The summary of weight or blood pressure depicts an increase or decrease during the last week, month, three months or year. Google Fit enable to add data directly or synchronize with other apps and devices. Therefore, it can store data about nutrition and hydration and HR as well. These include for example applications, such as RunKeeper, Strava, MapMyFitness, Lose It! or wearable devices, like Wear OS, Xiaomi Mi bands, Nike+ or Withings. The main advantages are goal setting, such as move minutes and heart points that follow rules of PA construct and synchronization with many other apps and devices. The main disadvantages lie, that a social network support and an option to create a training plan are absent.

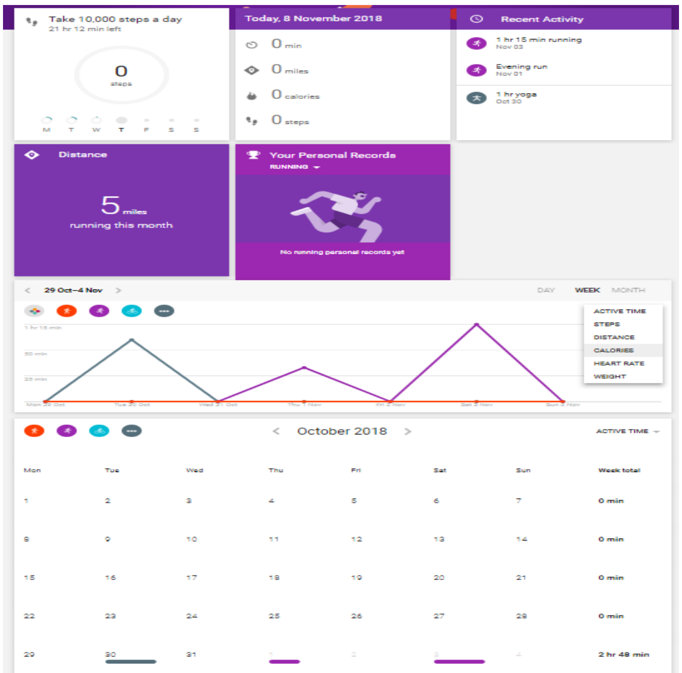

Figure 2.3: Google Fit website dashboard

Lose It! <https://www.loseit.com>

Lose It! is a weight loss program based on personal goals, such as weight, body fat, sleep, nutrients, hydration and exercise, that tracks calorie-income of food and calories burned with exercise. The app runs on Android and iPhone devices. An exercise log has numerous exercises and contains varied info, for example basketball has a detailed description about which way it is performed, like officiating, shooting baskets, wheelchair, non-game or game, information about duration and calculated calories burned. It is possible to create your own exercises. Exercises logs can be shared via Lose It community or Twitter. It is possible to participate in different food, exercise or another type of challenges. Setting other goals than regarding weight is possible in a paid version. An analysis contains information only about food calories consumed and exercise calories burned for a current day or the past week and a daily nutrient composition. There is an option to connect with wearable devices, like Apple Watch, Withings, Fitbit Aria or in premium version with apps, such as RunKeeper, Strava, Fitbit, MapMyFitness and more. The main advantage is extensive features for tracking food intake. The disadvantages lies, that an analysis of exercise and an option to create a training plan are absent.

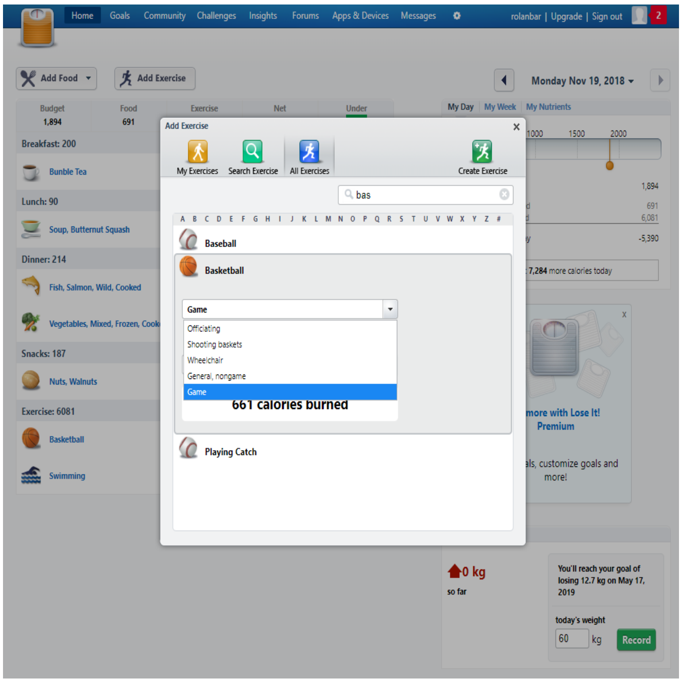

Figure 2.4: Lose It! website for logging activities

MapMyFitness <https://www.mapmyfitness.com>

MapMyFitness runs on Android and iPhone smartphones that log over 600 activities with a large selection of activity types. The activity log contains title, date, gear used, start time and notes. Depending the activity, other fields are generated. For example, running contains route, that can be created, duration, distance, steps, calories burned, average HR and average speed. Pace and steps are automatically calculated. For a gym workout is a different log form is used with duration, calories burned and average HR. The activity log is enable to share via Facebook, Twitter and mail. There are several analysis in the free version of the app. One analysis provides monthly stats of duration, distance, calories and steps and other analysis offer monitoring of the progress of duration, speed, pace, calories, distance or steps in a time. The summary of activities shows performed activity, filtered by an activity type and are viewed as a list or a calendar for a particular month. There is also data about overall distance, duration, calories and number of work out for this month. The goal log contains the activity name, goal type, such as number of workouts, distance and duration, goal target as a concrete number of the goal type and start time. Is it possible to create your own training plans or choose from five-kilometers to marathon plans, that require a premium account. Support for personal improvement by Map My Fitness community is available and create or participate in challenges is possible. There is the option to synchronize large amounts of wearable devices, such as Garmin, Polar, Fitbit, Jawbone, Nike+ and with many popular fitness apps. The main advantages are wide-range of PA, that are divided in categories, like aerobic, gym work out, dancing and many others. Creating own training plans. Support for numerous wearable devices and Under Armour shoes to track PA. The app do not contain any noticeably disadvantages, according to assembled features.

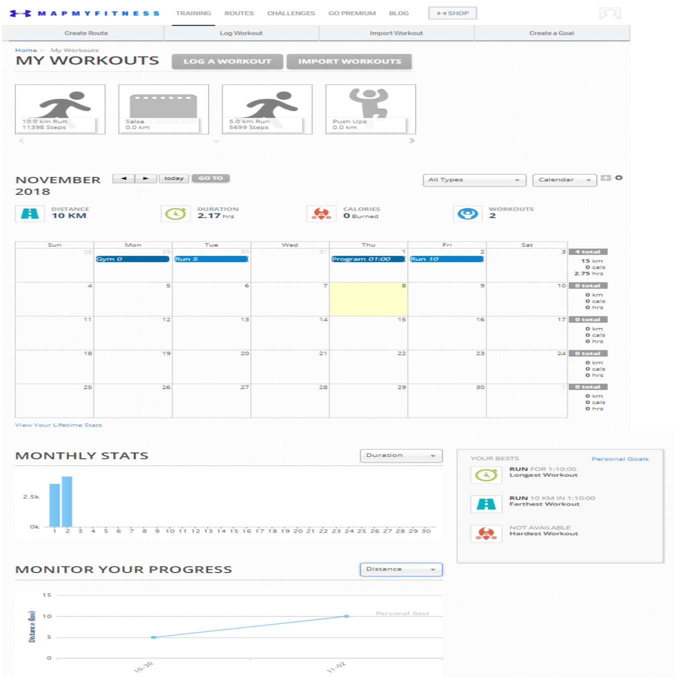

Figure 2.5: MapMyFitness website of workouts

Noom Coach <https://www.noom.com>

Noom Coach provides evidence-based weight loss strategies and tracks income food calories, calories burned by an exercise, blood pressure and blood sugar. Before signing in and making a customized plan, the app asks a series of questions regarding the goals concerning the ideal weight, the actual weight and height, gender, age, eating habits, health issues like a treatment for diabetes, antibiotic taken in the past, environment where the person lives. Next, current and target BMI is counted. After that, a focus on nutrition, PA, building good habits or other is defined by a user. As second series of questions are asked about physical limitations, dietary restriction, and food allergies. Afterwards, in a paid version, the plan is developed in collaboration with doctors and psychologists and thus a coaching team helps to set personal goals, weight loss and build healthy habits. In regard to logging an activity, an activity log includes name, time of the performed activity and its duration. There is an option to adjust daily steps, as well. There is possible to synchronize steps, weight, blood pressure and blood glucose data with apps, like Fitbit, Runkeeper, Garmin Connect and wearable devices, like Misfit, iHealth, Withings and others. The main advantage is the course is proven by medical journals and institutions. The main disadvantage is that course is feasible only in the paid version.

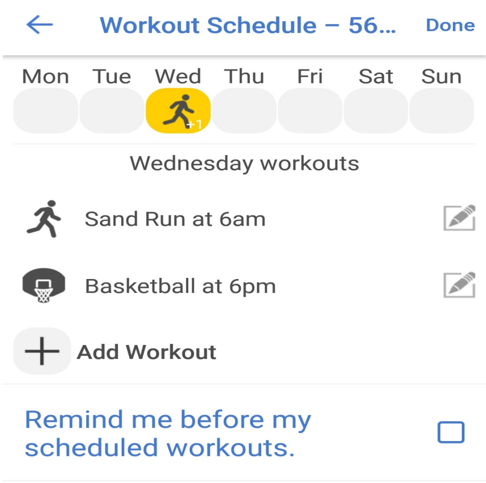

Figure 2.6: Noom Coach workout tracking in a mobile version

RunKeeper <https://runkeeper.com/>

RunKeeper tracks more than thirty exercises on Android and iPhone devices. An activity log collects data about the activity type, equipment type, route, distance, duration, pace, average HR and its time. A feature to tag friend to this activity foreseen. The average pace and calories burned are automatically calculated. User can create and save routes. In the free version, there is only a report of distance over a time. More analysis are available in the paid version. There is a possibility to set several goals, like achieving a distance, losing weight or finishing a race. The app enables to synchronize with partner apps, including Fitbit, Fitocracy, Garmin Connect and synchronizes with devices, such as Android Wear, Pebble or Garmin watches. Users can join challenges to stay motivated, collect rewards and share work outs with friends on Facebook, Twitter or in RunKeeper community. Custom training plans are built by specialists in a paid version. The main advantages are a possibility to upload up to one-hundred files at once by a bulk importer. Synchronization with multiple apps and devices. Building training plans by specialists. The main disadvantage is a simple analysis, that is available only in a free version.

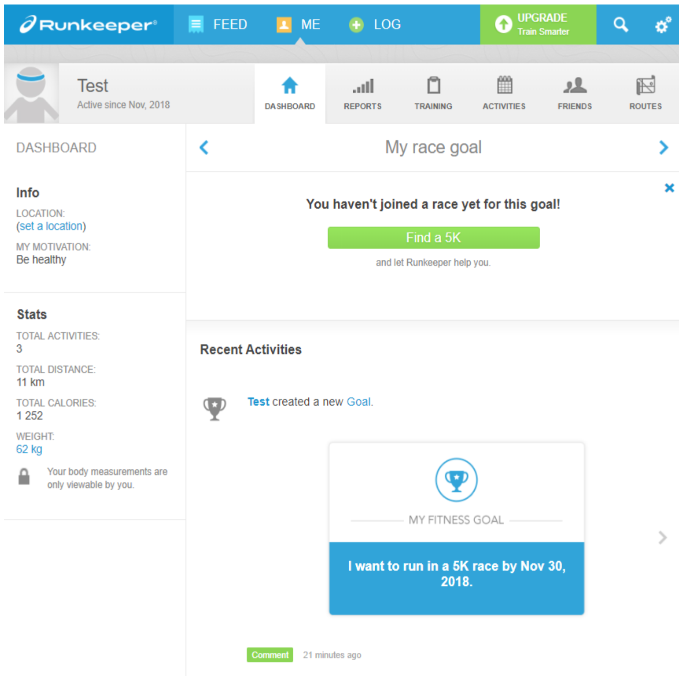

Figure 2.7: RunKeeper website of workouts

Runtastic <https://www.runtastic.com>

Runtastic is intended for Android and iPhone smartphones recording more than sixty work outs. A work out log contains information about date, start time, notes, distance, elevation, duration, pause, calories, HR, feelings, surface, weather and its temperature. The work out log can be shared on Facebook, Google+, Twitter and among Runtastic friends. Extensive statistics are provided. Performance progress is expressed as average duration, average distance, average speed, average pace or average HR in a different time frame and training history, that contains overall distance, overall duration, overall calories burned and others. A paid version has a statistics about training habits or personal records. All statistics can be exported or printed. There is a possibility to create own your routes or search them. Setting yearly running goal for a particular year is possible. Limited training plans are offered in a free version. These are designed by experts and help to increase endurance in a particular run, burn extra calories and preparing for long runs. The app support connection for Fitbit, Withings or apps, like Garmin Connect and MyFitnessPal. The main advantages is integrated life coach and advanced statistics are offered in a paid version. The main disadvantage is setting and reaching more personal goals are accessible in the paid version.

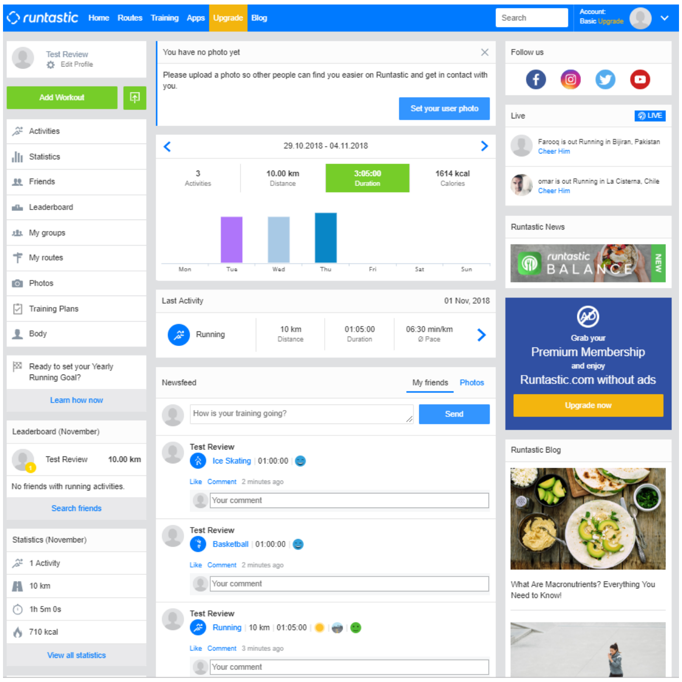

Figure 2.8: Runtastic website of workouts

Sports Tracker <https://www.sports-tracker.com>

Sports Tracker records several work outs, using GPS on Android and iPhone smartphones. A work out log contains a list with more than fifty activities. Each activity has fields about the start time, distance, duration, description, average heart rate, maximal heart rate, energy and steps. Average speed and average pace are automatic calculated. Photos can be added for the work out. The work out can be shared and supported by other users of Sports Tracker or on Facebook and Twitter. A summary is comprised of the number of work outs, total time of performed activities, total number of kilometers and total number of calories burned for past month or past week. It is possible to create and plan routes. There is synchronization with Apple Watch and Suunto and connecting with a speed and cadence senors is possible. The main advantage is connecting a Sports Tracker HR sensor and record an intensity during and after activity. The main disadvantages are no option to synchronize with other PA promoting apps and an inability to set your own goals, training plans and to get a tailored feedback.

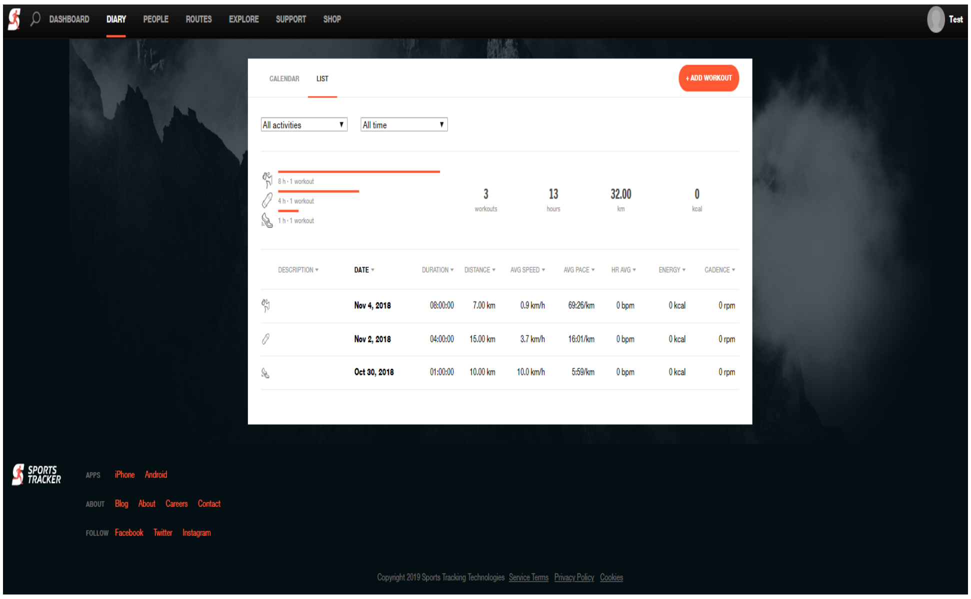

Figure 2.9: Sports Tracker website of workouts

Strava <https://www.strava.com>

Strava is an app for Android and iPhone devices, that is mainly determined to track running, cycling, swimming and build to and share your own routes. The app can be used for over more than thirty activities, such as Alpine Ski, Hike, Crossfit, Kayak or Surf. In the activity log, it is possible to upload data manually or synchronize with devices, like Garmin, Suunto, Fitbit or Polar or apps including MapMyFitness, Nike+, Runkeeper or Runtastic. A manual activity log contains data about the distance, duration, calculated path, elevation, sport title, sport type, like race or work out, description, date and time, shoes or bike information and visibility of a particular activity log for Strava community. Logs are possible to be shared on Facebook, Twitter or Blog. There is a training calendar that gives a monthly view or the past year about the amount of performed activity expressed in hours, kilometers, number of personal records and number of activities. The paid version of the app contains training plans, customization of goals and advanced stats. Running training plans comprise five-kilometers run, ten-kilometers run, half-marathon and marathon plan, that are tailored based on personal race date and experience level. Cycling training plans in- clude sixty minute climb, ninety second sprint, beginner indoor training plan and others, that are customized based on training volume, from five hours per week to twelve. Advanced stats get more insight from GPS, HR or power meter. Users can participate in many challenges and compete with others. The main advantages are tailored training plans for runners and cyclist and synchronization with many devices.

The main disadvantage is these tailored training plans and advanced analyzes are fea- sible in the paid version.

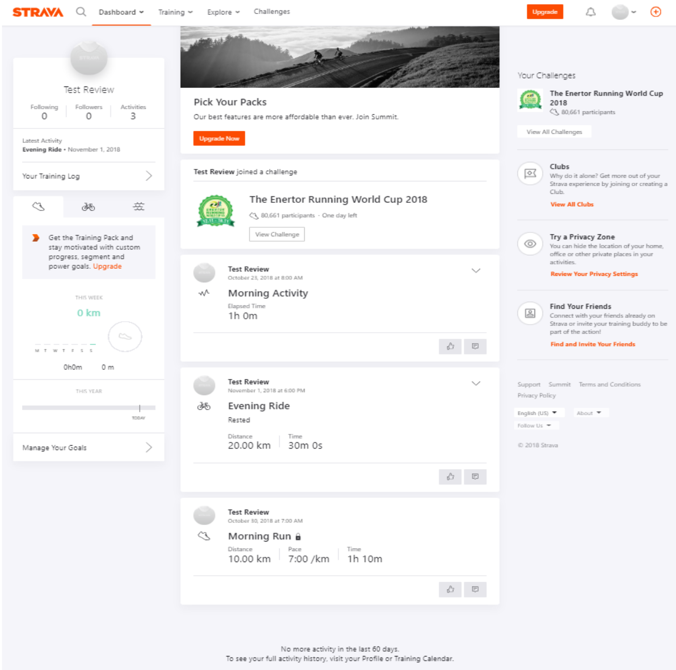

Figure 2.10: Strava website of an activity feed

2.3.4 Comparison of existing solutions

Following table presents how existing solutions meet the criteria:

Table 2.1: A feature comparison for existing applications

App name/ Feature	**Price**	**Range of PA**	**Tabulate a data**	**Periodic Summary**	**Goal Setting**	**Training plan**	**Social Support**	**Sync with Other Apps**	**Sync with Other Devices**	**Tailo -red Feed-back**	**User Friendly interface**
Fitbit	free	0	+	+	+	- -	+	+	0	0	+
Fitocracy	free lite	++	+	-	0	+	++	0	- -	+	+
Google Fit	free	+	+	+	+	- -	- -	++	+	0	+
Lose It!	free lite	+	0	0	+	- -	+	+	0	0	+
MapMyFitness	free lite	++	+	+	0	0	+	++	++	0	+
Noom Coach	free lite	0	0	0	++	++	+	+	+	++	+
RunKeeper	free lite	0	+	0	+	++	+	++	++	+	+
RunTastic	free lite	+	+	++	0	++	+	0	0	+	+
Sports Tracker	free	+	+	0	- -	- -	+	- -	0	- -	+
Strava	free lite	0	+	+	+	+	+	0	++	0	+

(++) Excellent, (+) Exceeds Expectations, (0) Meets Expectations, (-) Improvement needed, (- -) None

*Free lite version contains basic features

2.3.5 Conclusion

According to the objective features, the study of ten existing solutions differs in each other's way of promoting PA. One group of apps, like RunKeeper, Runtastic, Sports Tracker, Strava is intended for intense cardio. A second group, apps such as Fitbit, Fitocracy, Google Fit, MapMyFitness focuses more on tracking work outs and analyse them. The last group includes LoseIt! and NoomCoach that have a focal point in weight loss.

2.4 General conclusion

A combination of self-report and objective measures methods is a significant way of how to capably encourage performing PA. To incorporate persuasive features, such as social support, rewards, reminders, suggestions and tailored feedback can help users to achieve personal goals. There are many ways to promote PA effectively using information technology, but it depends on personal desires and what goals want to be accomplished.

Chapter 3

Requirements

This chapter deals with requirements, that are proposed from outcomes of literature and market research. In Section 3.1, functional requirements are analyzed. In Section 3.2, nonfunctional requirements of the application are presented.

3.1 Functional requirements

This section discusses the functional requirements of the application. In Section 3.1.1, a set of functional requirements is created. In Section 3.1.2, a synopsis of use cases is assembled. In Section 3.1.3, the three most used use case scenarios are given. Finally, the link between functional requirements and use cases is presented in Section 3.1.4.

3.1.1 Overview of functional requirements

In the following paragraphs, 9 functional requirements are thoroughly discussed. Functional requirements describe the functionality of the software. The application has only single role as a user and so no admin or another role can do different functionality.

F1: Registration

The app enables an unauthenticated user to create a new account. For the registration process, it is required to fulfill a first name, last name, email address, password and its confirmation, gender, age, height and weight. The password must be a minimum of 8 characters in length and the email address must be unique. The weight must be filled in to be able to automatically calculate calories of a specified PA.

F2: Authentication

The app enables an unauthenticated user to log in to an account. A successful authentication is provided through a unique email address and password and after that, the user is logged in to the app. Login data is entered by the user in the registration form in functional requirement F1. When the user is logged in to the app, it is possible to log out that user safely.

F3: Reset a forgotten password

The app enables a registered user to reset a forgotten password. The registered user enters a valid email address, that is used in the registration form in functional requirement F1. Afterwards, a link for resetting the password is emailed to the user on a provided email address and the user can set a new password within this link.

F4: User profile management

An authenticated user can edit the following 2 parts of user profile and their fields:

- Personal profile - first name, last name, age, gender
- Body statistics - body fat, height, weight, resting heart rate.

To be able to automatically calculate calories, the age, gender and weight are required.

F5: Creating and editing a physical activity

The app enables an authenticated user to create a new PA and subsequently edit that PA. A PA form contains basic fields, such as activity type, date, description, start time, title and advanced fields that are generated depends on a category of the chosen PA type from a final list. The table below describes a categorization of PA types and their generated advanced fields, based on that category. For example category "A" with PA Type - Hike, Run and Walk has a PA form, that is generated with the same advanced fields. In addition, each PA type contains a PA subtype, that represents one PA, like basketball or a more detailed described PA, like dog walk. The complete categorization of PAs is assembled based on the Compendium of PA [5]. The Compendium of PA classifies activities by function, specific type of activity and intensity.

Table 3.1: A categorization of physical activities

Category	Activity Type	Activity Subtype	Advanced Fields
A	Hike	Cross country hiking, General hike, Hills, Rock climbing, Trekking	Activity subtype, Average HR, Average pace, Average speed, Calories, Distance, Duration, Elevation gain/loss, Laps, Laps length, Terrain, Steps
	Run	Cross country run, Dog run, General run, Group Run, Intervals, Jogging, Long run, Run race, Sprints, Stairs, Track run, Trail run	
	Walk	Brisk walk, Dog walk, General walk, Intervals, Long walk, Nordic walking, Power walk, Stairs, Sightseeing, Treadmill, Walk Race	
B	Bike ride	BMX, Bicycle touring, Mountain biking, Road cycling, Track cycling	Activity subtype, Average HR, Average pace, Average speed, Cadence, Calories, Distance, Duration, Elevation gain/loss, Power, Terrain
C	Swim	Backstroke, Breaststroke, Butterfly, Drills, Freestyle, Front crawl, Individual medley, Medley relay, Sidestroke, Synchronized	Activity subtype, Average HR, Average pace, Calories, Distance, Duration
D	Gym/Weight workout	Crunch, Deadlift, Dips, High stepping, Kettlebell, Leg curls, Leg Press, Plank, Pull-ups, Push-ups, Squats, Superman, Wall sit	Activity subtype, Average HR, Calories, Duration, Repetitions, Weight
E	Ball game	Badminton, Baseball, Basketball, Bowling, Cricket, Floorball, Football, Golf, Handball, Hockey, Lacrosse, Paintball, Pétanque	Activity subtype, Average HR, Calories, Duration
	Dance	Ballet, Ballroom dancing, Disco, Hip-hop, Latin dance, Salsa, Street dance, Swing dance	
	Fitness activity	Abs, Aerobic, Bodypump, Bodystep, Bootcamp, Circuit training, CrossFit, Dance fitness, HIIT, Pilates, Rope skipping, Rowing training, Spinning, Tabata, Trampoline, TRX, Yoga, Zumba	
	Martial art	Aikido, Archery, Brazilian jiu-jitsu, Boxing, Capoeira, Fencing, Judo, Jujitsu, Karate, Kickboxing, Kung Fu, Mixed martial arts, Muay Thai, Tae kwon do, Taichi	
	Occupational	Gardening, Farming, Fishing, Horse riding	
	Water activity	Aqua jogging, Canoeing, Diving, Kayaking, Kitesurfing, Paddleboarding, Parasailing, Rafting, Rowing, Sailing, Scuba diving, Snorkeling, Surfing, Wakeboarding, Water aerobics, Water polo, Water running, Water skiing, Windsurfing, Yachting	
	Winter activity	Bobsledding, Ice Hockey, Ice Skating, Mountaineering, Skiing, Snowboarding	

F6: Data calculations

When the user is creating a new PA, as it is described in the functional requirement F5, data calculations are being performed. Each of the following automatic calculation is provided, when a field of the PA form for this calculation is occurred in a category of the particular PA.

- **Average pace**:
$$average\ pace = \frac{total\ time}{total\ distance}$$

- **Average speed**:
$$average\ speed = \frac{total\ distance}{total\ time}$$

- **Calories**
Evaluation methods of EE can be determined using indirect calorimetry, bioelectrical impedance, doubly labeled water, predictive equations, PA records and many others. Indirect calorimetry and doubly labeled water are more accurate methods, but they are expensive. Other methods have limitations, but they are considered as convenient and less expensive and can be used with some caution [35].
In case of manually creating a new PA, the prediction equations of calculating calories from HR for a particular PA can be suitable as a complementary method for assessing the EE. The advantages are its low cost, simplicity and the EE can be predicted with better accuracy from HR than from MET in a moderate-intensity and high-intensity of the physical exercise. Individual EE in kilocalories **(kcal)** is possible to calculated by this formula for both gender [24] [37] [15]:

  For **male**:

$$EE[kcal.min^{-1}] = \frac{((-55.0969 + (0.6309 \times HR) + (0.1988 \times W) + (0.2017 \times A))}{4.184) \times 60 \times T}$$

  For **female**:

$$EE[kcal.min^{-1}] = \frac{((-20.4022 + (0.4472 \times HR) - (0.1263 \times W) + (0.074 \times A))}{4.184) \times 60 \times T}$$

where

HR = Heart rate (in beats per minute)
W = Weight (in kilograms)
A = Age (in years)
T = Exercise duration time (in hours)

- **Steps**
  Steps are calculated based on an approximately equation 2000 steps equals 1 mile or 1243 steps equals 1 kilometer [36].

F7: Deleting a physical activity

An authenticated user can delete a particular PA.

F8: A summary of performed physical activities

The app enables to view the personal PA history. The summary contains a list of particular, performed PAs and their attributes, such as calories, date, distance, heart rate and title.

F9: An analysis of performed physical activities

The analysis of performed PA is represented by graphs. The graphs contains information about a total number of the user's calories, distance or steps during the certain day for the past month.

3.1.2 Use cases

The summary of use cases are listed in Table 3.2. For each use case, a title of use case and its priority, frequency and complexity is given. The frequency is classified on a relative scale from 1 to 10; the priority and complexity as low, medium and high and they are determined by the method of estimation.

The first five use cases are related to general use cases and the last six use cases are related to recording, viewing and progression of the PA.

ID	Use Case Name	Priority	Frequency	Complexity
UC1	Sign in to the app	High	1	High
UC2	Login to the app	High	4	High
UC3	Logout from the app	High	3	High
UC4	Reset a forgotten password	High	2	Medium
UC5	View and update the user profile	Medium	3	Low
UC6	Create an physical activity	High	8	High
UC7	Search in an activity dictionary	High	7	Medium
UC8	Update an existing physical activity	Medium	3	Medium
UC9	Delete an existing physical activity	Low	1	Low
UC10	View a history of performed physical activities	High	8	Medium
UC11	Watch the statistic graph of performed physical activities	High	7	High

Table 3.2: Priority, frequency and complexity of the use cases for the app

3.1.3 Use case scenarios

In this section, based on Table 3.2, the three most used use cases are presented and their use case scenarios are discussed in detail. The other use cases are described further in Appendix A.

UC6: Create an physical activity

- **Description:**
  A user creates a new PA. Intended PA is possible to search in an activity dictionary. Each physical activity from the activity dictionary belongs to a particular category. Based on that category, advanced fields of the activity form are generated. If the user fills up a duration, distance and heart rate fields, there are automatically calculated average pace, average speed, calories and steps fields.
- **Trigger:**
  The user presses the "Create an activity" button.
- **Preconditions:**
  The user is already logged in to the app.
- **Normal Flow:**

1. The user searches a physical activity type in an "Activity Type" select menu.
2. Based on the selected activity type, the app generates other advanced fields.
3. The user searches a desired physical activity in "Activity Subtype" select menu and fill out generated fields.

- **Alternative Flow:**

1. The user did not find a desired physical activity in an "Activity Subtype" select menu.
2. The user type the name of the physical activity to the "Title" field.
3. The user fill out default fields.

- **Postconditions:**
The physical activity is created and subsequently displayed in history of performed physical activities.

UC10: View a history of performed physical activities

- **Description:**
A user views a list of past physical activities. Each physical activity shows a detail information about title of PA and its attributes, such as calories, date, distance and heart rate.

- **Trigger:**
The user selects the "Dashboard" link.

- **Preconditions:**
The user is already logged in to the app.

- **Normal Flow:**

1. The user views a list of all performed PAs with their specified atrributes.

- **Postconditions:**
The list of performed PAs is shown.

UC11: Watch the statistic graph of performed physical activities

- **Description:**
  A user watches the performed PA statistic graph.
- **Trigger:**
  The user selects the "Analysis" link.
- **Preconditions:**
  The user is already logged in to the app.
- **Normal Flow:**
  The app displays following parameters for the certain day for the past month in the 2D bar graph.
  1. The total number of the calories.
  2. The total number of the distances.
  3. The total number of the steps.
- **Postconditions:**
  The statistic graphs are presented.

3.1.4 Mapping between functional requirements and use cases

The following table shows the realization of the particular functional requirements to the identified use cases. This table is possible use for a control of completed all functional requirements related to identified use cases. Each functional requirement is linked to at least one or more use cases and the use case UC6 and UC8 are covered with more than one functional requirement.

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11
F1	X										
F2		X	X								
F3				X							
F4					X						
F5						X	X	X			
F6						X		X			
F7									X		
F8										X	
F9											X

Table 3.3: Linking functional requirements to use cases

3.2 Nonfunctional requirements

Non-functional requirements are conditions to be met in order to be effective, or constraints to be taken into consideration. Main focus is aimed at the usability of the application.

3.2.1 Usability

Usability is the qualitty atrribute that asseses how easy a user interface **(UI)** is to use [22]. To ensure the usability of a UI, heuristic evaluations can be applied. Heuristic evaluation is a usability engineering method for a systematic inspection of a UI design for usability [26]. These principles are named heuristics because "they are broad rules of thumb and not specific usability guidelines" [21].

The frequently used heuristics for UI design are the ten usability heuristics for UI design by Jakob Nielsen [21]. These heuristics, exactly how they are represented by Jakob Nielsen, are [26] [21]:

1. **Visibility of system status:** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. **Match between system and the real world:** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. **User control and freedom:** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4. **Consistency and standards:** Users should not have to wonder whether different words, situations, or actions mean the same thing.
5. **Error prevention:** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. **Recognition rather than recall:** Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. **Flexibility and efficiency of use:** Accelerators - unseen by the novice user - may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. **Aesthetic and minimalist design:** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. **Help users recognize, diagnose, and recover from errors:** Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. **Help and documentation:** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Chapter 4

Architectural overview

This chapter describes an architectural overview of the application and it is depicted in Figure 4.1. Based on the created software requirements, a development of the web application has been considered. The architecture comprises of the three main components: the back-end system, the front-end system and the database. These components are discussed in Section 4.1, Section 4.2 and Section 4.3 respectively. The web application is designed as a *3-tier architecture*, which comprises from a data tier, business tier and a presentation tier.

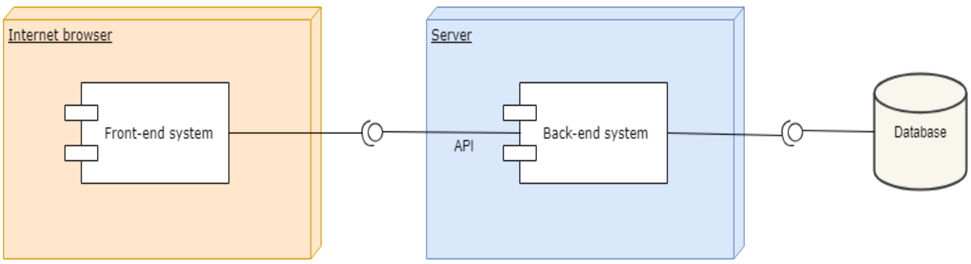

Figure 4.1: Overview of the proposed architecture

4.1 Back-end

The back-end uses the *Model-View-Controller* **(MVC)** architectural pattern depicted in Figure 4.2 and these three components are available as follows:

- **Model** handles the application data and the state. Processing data to and from a database, is among its responsibilities. Model receives and update data from a controller.
- **View** generates an appropriate UI of the app based on a user action. The view displays user data in a proper interface based on controller information received.
- **Controller** acts as an interface between the model and the view. It reacts on the basis of the user input and subsequently update the model.

The advantage of the MVC pattern is to enable a separation of concerns and code. The model does not depend on the view, therefore, any modifications in the model does not affect the entire architecture. It is also easy to test each of these components and in addition to that, components are reusable. It is possible to create multiple views for a particular model. In the case of working in a development team, it allows to work concurrently on a different components of the web app without influencing one another. The main disadvantage of the MVC lies in a complexity of this architectural pattern. It requires to understand an information flow among components [20].

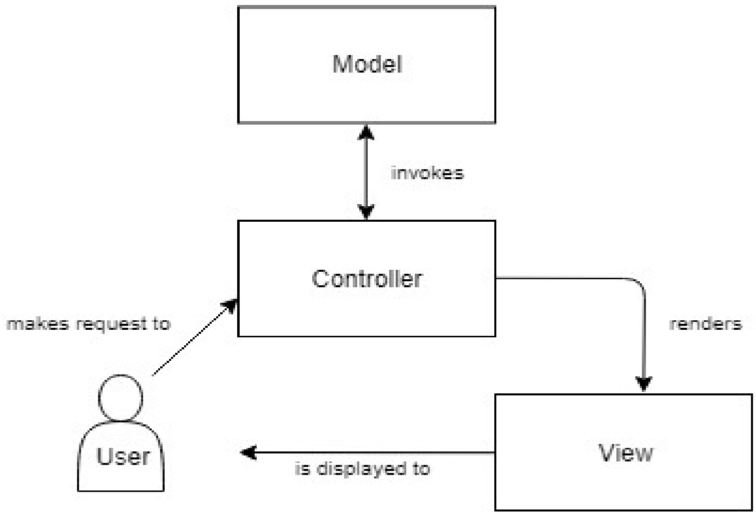

Figure 4.2: Model-View-Controller architectural pattern

4.1.1 API controller

To handle the communication between back-end and front-end, an API is provided. The API controller is used to execute back-end requests. One method is provided for each available endpoint. This controller builds an HTTP request based on the arguments of the method.

4.2 Front-end

The front-end is running in the internet browser that allows the user interacts with the app in a user friendly way. It uses MVC-like pattern as well, even if not explicitly [27]. The front-end uses a *Separation of concerns* **(SoC)** design principle. It is a principle that enables to separate different logic such as UI functionality, business logic and infrastructure logic. SoC also includes the MVC architectural pattern [20]. The implementation of this principle and this pattern are shown in Section 5.2.

4.2.1 Usability

An essential aspect during the development of the app is usability, that is clarified in Section 3.2.1. Interpretation of how a particular usability heuristic is implemented in the app is given in Section 5.2.6.

4.3 Database

The back-end handles and stores the personal data of users. For this reason, security is an important aspect. In the case of multiple users interact with the app, data availability should be ensured at all times and the data should be handled as expected. It is important to guarantee a transaction atomicity during the database operations as well. Moreover, the database should assure a high performance, while retrieving data.

Chapter 5

Implementation

This chapter describes implementation details concerning the architectural components are given, as well as a technology choice regarding to them. An overall technology choice is realized with respect to the application scalability and a potential increase of the requirements. The implementation of the three architectural components of the app is elaborated in Section 5.1, Section 5.2 and Section 5.3 respectively. Furthermore, a separate module, for representing each of the logic, which are a data logic, business logic and a presentation logic, has been created in the app's source code as `persistence`, `backend` and `frontend` respectively.

5.1 Back-end

First, an introduction to the technologies, that have been used throughout development, are discussed in Section 5.1.1. Next, an authentication of the app is addressed in Section 5.1.2. Following sections describe classes that define a domain model - Section 5.1.3, repository - Section 5.1.4, service - Section 5.1.5, controller - Section 5.1.6 and view - Section 5.1.7.

The project structure of the back-end code is separated to the folders following code structure recommendations in Spring boot apps [45] [14]: `model` and `repository` are located in the `persistence` module and `service` and `controller` are located in the `backend` module.

5.1.1 Technology choice

There are many frameworks and programming languages that implement a web application applying *REST*[1] interface. The choice of the appropriate technology is focused on java-based frameworks. Using frameworks, it is enables to concentrate on the business logic of the app contrary to writing a basic functionality, like a database connection. Furthermore, it is suitable for developing a robust architecture and it has a higher level of the security. For example, java-based web frameworks for developing REST API are Grails, JavaServer Faces, Play, Spring or Vaadin. For this application, *Spring Boot*[2] of the version 2 as an extension of the *Spring framework*[3] is chosen and the reasons for this choice are further described.

Spring framework

Spring framework is an open source Java platform. It enables comprehensive programming and configuration model for developing robust and maintainable java-based enterprise applications, that are possible to deploy on any platform. Spring framework is mainly characterized by utilizing two concepts, that are *Inversion of control* **(IoC)** and *Dependency injection* **(DI)**. IoC is a software design principle in which a software system, including a framework, obtains control flow from reusable code. Compared to traditional procedural programming, in which a software system makes calls to a reusable library. By allowing the reusable code to call into the software system, the IoC inverts this control flow [20]. IoC is achieved through DI in Spring. DI is a technique that provides dependencies to a class and it achieves dependency inversion in this way. Dependencies are injected to a client that requires it. The main advantage of the DI is writing loosely coupled code and it makes Spring apps easier to test, maintain and extend [20]. Spring offers a *container*, that implement DI and it is referred to as the Spring *application context*. The Spring container is responsible for creating and managing objects known as beans and handling their life-cycle.

[1] <https://www.restapitutorial.com/>
[2] <https://spring.io/projects/spring-boot>
[3] <https://spring.io/projects/spring-framework#overview>

Spring Boot

Spring Boot is a part of the Spring ecosystem and it creates stand-alone Spring apps. The main advantage of using Spring Boot contrary to Spring is elimination the boilerplate configurations needed for setting up Spring app and 3rd party libraries and on that account, it enables faster and efficient development. This is realized by a process called *auto-configuration*. Next advantage is a *Starter dependencies*, that facilitate the project build dependencies, for example by adding only one jar file spring-boot-starter-web instead of adding jar files Spring core, Spring Web, Spring Web MVC and Servlet. The last advantage but not least, Spring boot includes an embedded Tomcat, Jetty or Undertow. This application uses Tomcat servlet container, which is default. It processes services for client requests and it helps to avoid complexity in deployment of the app. Key components of Spring Boot are depicted in Figure 5.1.

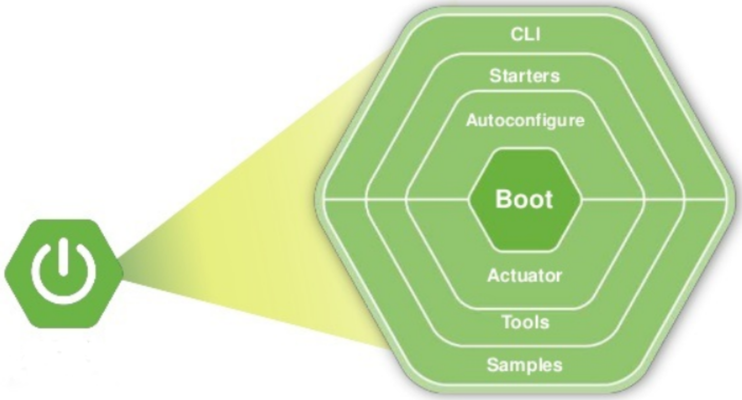

Figure 5.1: Spring Boot components, adopted from [10]

Hibernate ORM

A persistance layer of the app consists of Java Persistence API compliant *Hibernate ORM*[4] framework. Hibernate performs an object mapping, that maps Java classes to database tables using annotations in this app. For executing queries, it is used *Hibernate Query Language* **(HQL)**, which is comparable with SQL, but HQL provides full support for polymorphic queries. One of the main advantage of using Hibernate is database independent and subsequently a potential migration to a new database.

[4] <https://hibernate.org/>

Build Tool

For dependency management, project life-cycle, project modularization and project building is used Maven[5].

5.1.2 Authentication

Authentication of users, storing login data and securing endpoints are assured by *Spring Security*[6]. Spring Security is a framework focusing on providing authentication and authorization to Java apps. It provides a protection against attacks, such as clickjacking, cross site request forgery or session fixation. The password should not be ever saved as plain text to the database. Accordingly, the password is hashed using the *BCrypt* algorithm, before it is saved to the database. For the authentication of users is used *JSON Web Tokens*[7].

5.1.3 Model

Domain *models*, that are annotated with a *@Entity* in the app contains classes, that defines database entities. The persistence logic should by extracted from the service layer. The domain model contains a declaration of *Constraints* for validation domain objects in the form of annotations as well. The following source code 5.1 contains a snippet of the implementation of the entity `Activity`.

```
@Entity
@Table(name = "activity")
public class Activity extends BaseEntity<Long> {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "activity_type")
    private String activityType;

    @NotBlank(message = "Activity title is required")
```

[5] <https://maven.apache.org/>
[6] <https://spring.io/projects/spring-security>
[7] <https://jwt.io/>

```
    @Size(max = 50, message = "Please use less than 50
        characters")
    @Column(name = "title", nullable = false)
    private String title;
}
```

Source code 5.1: An example a part of the implementation the entity `Activity`

5.1.4 Repository

A *repository*, annotated with a *@Repository*, is a mechanism that encapsulates the behavior for storage, retrieval, search, update and delete operation on objects. In Source code 5.2, the JPA query method selects a list of activities based on the given email.

```
@Repository
public interface ActivityRepository extends BaseRepository<
    Activity>{

    List<Activity> findAllByPerformingPerson(String email);
}
```

Source code 5.2: An example of the implementation JPA query method

5.1.5 Service

Services are annotated with a *@Service* and they contains business-related logic of the service layer. Each service class implements an interface and therefore it inherits the abstract methods of the interface. A mapping between services and controllers is in a one-to-one relationship. Implementation of the service is shown in Source code 5.3.

```
@Override
public Activity createActivity(Activity activity, String email)
    {
    Person person = personRepository.findByEmail(email).
        orElseThrow(()
                -> new EntityNotFoundException("Email not found"
                    ));
    activity.setPerformingPerson(person.getEmail());
    person.addActivity(activity);

        return activityRepository.save(activity);
```

```
}
```

Source code 5.3: An example of the implementation the service for creating an activity

5.1.6 Controller

A *controller*, annotated with *@Controller*, implements the service layer of the app and this service layer is exposed via RESTful API. Considering that the future development of the app can include a mobile application, using the REST API is very suitable in this way. As the data exchange format, JSON is chosen. To create a RESTful API, using *Spring Data REST* is provided. The controller is by default a singleton and it should not execute business logic. The following snippet of the code 5.4 shows an implementation of the controller for adding an activity.

```
@PostMapping()
 public ResponseEntity<?> addActivity(@Valid @RequestBody
     Activity activity, BindingResult result, Principal
     principal) {
      LOG.debug("Request to create a new activity: {}", activity
          );
      ResponseEntity<?> errorMap = fieldValidator.validateFields
          (result);
      if (errorMap != null) return errorMap;

      Activity activity1 = activityService.createActivity(
          activity, ((Person) ((
          UsernamePasswordAuthenticationToken) principal).
          getPrincipal()).getEmail());
      return new ResponseEntity<>(activity1, HttpStatus.CREATED)
          ;
}
```

Source code 5.4: An example of the implementation the controller for adding an activity

5.1.7 View

A view of the back-end is represented only by a template of the email, that is sent in order to reset the user password. The server uses *Java Mail library*[8] to send out emails to the users.

[8] <https://javaee.github.io/javamail/>

5.2 Front-end

In Section 5.2.1, the technology choice is explained. Following sections describe classes that define an action - Section 5.2.2, reducer - Section 5.2.3, store - Section 5.2.4 and view - Section 5.2.5. Finally, Section 5.2.6 explains an implementation of the addressed heuristics for UI design.

The project structure of the front-end is arranged by the pattern *Rails-style*, that contains separate folders for `actions`, `components`, `reducers` and other miscellaneous configuration files, such as package.json [41] [39].

5.2.1 Technology choice

For an implementation of the front-end, the choice of suitable technologies is focused on javascript-based frameworks or libraries. The advantages of using the framework is described in Section 5.1.1. There are many popular front-end frameworks or libraries, such as Angular, Backbone, Ember, React or VueJS. As library, there is chosen for *React*[9] of the version 16, that works with a JavaScript runtime *Node.js*[10] in this app. The reasons for the selection are described further.

React

React is a popular JavaScript library for user interfaces and it does in a declarative way. It is component-based and the components are independent and reusable. This application defines the react component by using the *ECMAScript 6* JavaScript class, which makes the code cleaner. React utilize the *Virtual Document Object Model* **(DOM)** for selective re-rendering of the UI. The Virtual DOM is a lightweight copy of the DOM and contrary to that, handling of the Virtual DOM is considerably faster. After the Virtual DOM is updated, React compares it to a snapshot. Consequently, React comprehends which parts have been modified and only these parts are updated to the real DOM. This process is called *reconciliation* [18].

[9] <https://reactjs.org/>
[10] <https://nodejs.org/en/>

Redux

Redux[11] is a predictable state container for JavaScript apps. Basically, it is a library for working with the state of the app. It enables writing apps that behave consistently and they can run in different environments, such as a client, server or native. It develops the ideas of *Flux*[12], that is the application architecture for building UI. It provides a *unidirectional data flow*. Redux is characterized by the three primary principles:

1. **Store** is an object that holds the application's state tree. Only a single store should be used in a redux-based app.

2. **Action** - given the *state* is read-only, the only way to change the state is to emit an action. The action is defined as a plain object and it is the only way to receive data into the store.

3. **Reducer** is a function that computes a new state given the previous state and an action. It should be *pure functions*. Pure functions return the same result for given set of arguments.

The following Figure 5.2 depicts React and Redux architecture and their components.

[11] <https://redux.js.org/>
[12] <https://facebook.github.io/flux/>

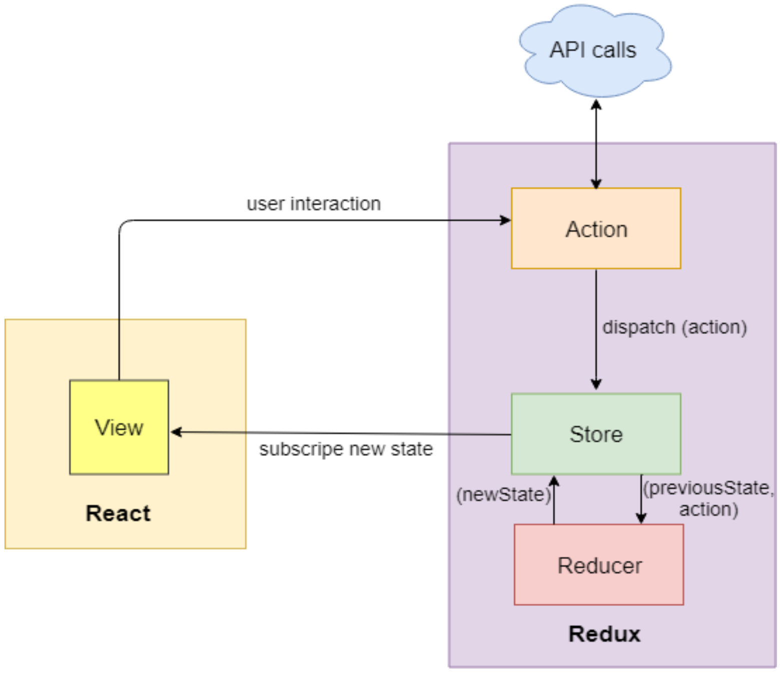

Figure 5.2: React and Redux architecture

React Router

There are many solutions available for routing in react. For this application, *React Router*[13] is used. It provides an entry point of the application.

Axios

Axios[14] is a promised-based JavaScript HTTP client library for both Node.js and the browser. Axios works asynchronously and it enables to make HTTP calls to REST endpoints and consume JSON data from the third-party REST API.

[13] <https://github.com/ReactTraining/react-router>
[14] <https://github.com/axios/axios>

Bootstrap

This application is designed using *Bootstrap 4*[15], which is very popular front-end component library for building responsive, mobile-first project on the web.

Chart.js

For the implementation F9 requirement, an analysis of performed PA, a visualization library is intended. The choice is concentrated on javascript-based libraries, which contains D3.js, Chart.js or Recharts library. For this application, *Chart.js*[16] is chosen. It is a popular library of open source HTML 5 charts for responsive web apps that applies the canvas element. It includes 8 basic chart types.

Package management

For the package management on the front-end side is used *NPM*[17] which is preinstalled in Node.js runtime.

5.2.2 Action

Action Creators are functions that create and return actions in Redux. *Actions* are plain JavaScript objects sending data to the store from the app and they are only the store's source of information [40]. To be able to do asynchronous operations, such as to delay the dispatch of an action or to dispatch only if certain conditions are fulfilled, *Redux Thunk* middleware[18] is used. In Source code 5.5, the implementation of the action, which get activities from the back-end and consequently, use this data with an action, is provided.

[15] <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
[16] <https://www.chartjs.org/>
[17] <https://www.npmjs.com/tool>
[18] <https://github.com/reduxjs/redux-thunk>

```
export const getActivities = () => async dispatch => {
    const res = await axios.get("/api/activity");
  dispatch({
    type: GET_ACTIVITIES,
    payload: res.data
  });
};
```

Source code 5.5: An example of the implementation the action

5.2.3 Reducer

A *reducer* specifies how the state of the application changes when responding to the action, that is sent to the store. The following snippet of the code 5.6 displays the implementation of the `errorReducer`.

```
const initialState = {};

export default function(state = initialState, action) {
  switch (action.type) {
    case GET_ERRORS:
      return action.payload;

    default:
      return state;
  }
}
```

Source code 5.6: An example of the implementation `errorReducer`

5.2.4 Store

The *store* is the object that connects actions and reducers. There is a single store in the application and it can be described as a model, because it stores the whole state of the app.

5.2.5 Component

React components can be described as views. They render HTML content with the data getting from the state. React uses *JSX*[19], which is a HTML-based syntax extension to JavaScript.

5.2.6 Usability: addressing heuristics for user interface design

An app's usability is a key criterion for users to keep using an app. Consequently, Jakob Nielsen's ten usability heuristics, presented in Section 3.2.1, has been taken into account during the development of the app. These principles are subjective. Therefore, an objective binary decision as to whether this app satisfies a certain heuristic is not simple to make. Nonetheless, it is described further how the app has been tried to take into consideration these heuristics. The previews of the user interface is shown in Appendix B.

1. **Visibility of system status**
   Before deletion the particular PA, a window is displayed with the warning message to the user, that makes an attempt to do this action.
2. **Match between system and the real world**
   The app is available in English. Given that the user understands English, terminology, general messages and error messages are formulated and displayed to the user in that way that they could be understood.
3. **User control and freedom**
   The user can move smoothly through the various app pages and perform the potential actions by using the navigation menu. If the user choose some action by mistake, it is possible to cancel it.
4. **Consistency and standards**
   Referring to particular actions or situations, no distinct terms are used. A certain term or message implies the same thing in different situations.
5. **Error prevention**
   Error prevention in this app is realized to make the primary action noticeable with a larger click area whereas secondary actions are just displayed as a link. Next, the validation of registration form, login form and activity log is checked before saving them. If an issue is identified,

[19] <https://reactjs.org/docs/introducing-jsx.html>

the user has to resolve it before he can save the item again. Lastly, the actions to prevent crashing the app are accomplished by Java exceptions on the back-end side.

6. **Recognition rather than recall**
   The app contains an activity dictionary, that the user can choose from.

7. **Flexibility and efficiency of use**
   Given that, logging of an executed PA is the key app function. The user can do this using the floating action button placed on the top left corner or through a navigation menu.

8. **Aesthetic and minimalist design**
   The app does not contain information, which is not relevant. It uses icons instead of textual explanation in some cases. UI follows respect the principles of visual design, such as contrast, alignment or repetition and it uses the same color palette throughout the app.

9. **Help users recognize, diagnose, and recover from errors**
   The app provides prompt feedback to the user, when an error occurs in the app. It is realized by means of an alert dialog or showing the exact error.

10. **Help and documentation**
    As it is not feasible within the scope, documentation has not been created.

5.3 Database

As SQL database, it is used *PostgreSQL 11*[20]. It meets the criteria layed out in Section 4.3. The main advantage of using PostgreSQL contrary to MySQL is SQL compliance. PostgreSQL is largely SQL compliant as it meets almost all core features of the SQL standard. Compared to MySQL, it is a partially SQL compliant [1]. Since Hibernate is used in the app, it is possible to change database, like MySQL or Oracle. The database model is depicted in Figure **??**.

[20] <https://www.postgresql.org/docs/11/index.html>

Chapter 6

Evaluation

In this chapter, the evaluation of the web application is discussed. In Section 6.1, the user testing is explained.

6.1 User testing

In Section 6.1.1, the methodology of the user testing is introduced. In section 6.1.2, a set of test scenarios is created. Lastly, the results of the user testing are discussed in Section 6.1.3.

6.1.1 Methodology

The tree participants took part in the user testing of the web application. The user testing itself is realized using created test scenarios, that contain both positive and negative test cases and it is performed on a single device. The aim of the user testing is verification the usability of the designed UI, covering the created use cases, that are developed from functional requirements and finding out errors in the implementation of the web app. The following Figure 6.1 describes profiles of these participants.

Participant 1	
Age	26 years
Gender	Male
Occupation	Software developer
Participant 2	
Age	24 years
Gender	Female
Occupation	Auditor
Participant 3	
Age	57 years
Gender	Male
Occupation	Teacher of the informatics subjects

Table 6.1: Profiles of the participants for the user testing

6.1.2 Test scenarios

The table 6.2 contains a summary of the test scenarios, use cases, that are covered and test cases included, that are derived from a particular test scenario. Test cases contains both positive and negative testing. The following test scenarios are created based on the identified use cases and they assure covering these use cases. Each test case of the particular test scenario is described in detail below.

Test Scenario ID	Test Scenario Name	Use Case Covered	Test Cases Included
TS1	Check a user sign up	UC1	TC1, TC2
TS2	Check a user login	UC2	TC3, TC4
TS3	Check a user logout	UC3	TC5
TS4	Check resetting a forgotten password	UC4	TC6
TS5	Check a user profile	UC5	TC7
TS6	Check creating a physical activity	UC6	TC8
TS7	Check a search in the activity dictionary	UC7	TC9
TS8	Check updating a physical activity	UC8	TC10
TS9	Check deleting a physical activity	UC9	TC11
TS10	Check a history of performed physical activities	UC10	TC12
TS11	Check statistic graphs of performed physical activities	UC11	TC13

Table 6.2: Test scenarios

TC1: User sign up with valid data

- **Test scenario ID and test scenario name:**
  TS1 - Check a user sign up
- **Preconditions:**
  The user is unauthenticated.

- **Test data:**
Valid data are provided.

- **Test steps:**

1. Navigate to *http://localhost:3000/register*.
2. Fill fields, such as first name, last name, email address, password, confirm password, age, height, weight and gender.
3. Click on *Create Account* button.

- **Expected result:**
A new account is created and the user is redirected to the login page.

TC2: User sign up with invalid data

- **Test scenario ID and test scenario name:**
TS1 - Check a user sign up

- **Preconditions:**
The user is unauthenticated.

- **Test data:**
Invalid data are provided.

- **Test steps:**

1. Navigate to *http://localhost:3000/register*.
2. Fill fields, such as last name, email address, age, height, weight and gender.
3. Click on *Create Account* button.

- **Expected result:**
The error messages *"First name is required"* is displayed for the first name field and *"Password must be at least 8 characters"* is displayed for the password field.

- **Note:**
The user needs to fill up all the required fields correctly and after that, the account is created.

TC3: User login with valid data

- **Test scenario ID and test scenario name:**
  TS2 - Check a user login
- **Preconditions:**
  The user is unauthenticated.
- **Test data:**
  Valid data are provided.
- **Test steps:**

  1. Navigate to *http://localhost:3000/login*.
  2. Fill the personal email address and password. Login data is entered from the successful registration process.
  3. Click on *Log In* button.

- **Expected result:**
  The user is logged in to the app and redirected to the dashboard.

TC4: User login with invalid data

- **Test scenario ID and test scenario name:**
  TS2 - Check a user login
- **Preconditions:**
  The user is unauthenticated.
- **Test data:**
  Invalid data are provided.
- **Test steps:**

  1. Navigate to *http://localhost:3000/login*.
  2. Fill a user email address and password, that are different from the successful registration process.
  3. Click on *Log In* button

- **Expected result:**
  The error messages *"Invalid Email"* and *"Invalid Password"* are displayed for the email address field and the password field respectively.

- **Note:**
  The user needs to fill up the required fields correctly and after that, it is possible to log in.

TC5: User logout

- **Test scenario ID and test scenario name:**
  TS3 - Check a user logout
- **Preconditions:**
  The user is authenticated.
- **Test steps:**

  1. Click on *Logout* link.
- **Expected result:**
  The user is successfully logged out from the app and redirected to the landing page.

TC6: Resetting a forgotten password with valid email

- **Test scenario ID and test scenario name:**
  TS4 - Check resetting a forgotten password
- **Preconditions:**
  The user is unauthenticated.
- **Test data:**
  Valid email is entered.
- **Test steps:**

  1. Navigate to *http://localhost:3000/login*.
  2. Click on *Forgot your password?* link.
  3. Enter a valid email address.
  4. Check an email box and click on the link.
  5. Enters a new password and a confirm password.
- **Expected result:**
  The password is changed.

TC7: View and update the user profile

- **Test scenario ID and test scenario name:**
  TS5 - Check a user profile
- **Preconditions:**
  The user is authenticated.
- **Test data:**
  Data, that was filled up in the successful registration process, is loaded to the personal profile page.
- **Test steps:**
  1. Click on *User* icon.
  2. The data, filled up in the registration process, is viewable.
  3. Fill a resting heart rate and body fat in the body statistics section.
- **Expected result:**
  The personal profile and body statistics are update. The user is redirected to the dashboard.

TC8: Creating a physical activity

- **Test scenario ID and test scenario name:**
  TS6 - Check creating a physical activity
- **Preconditions:**
  The user is authenticated.
- **Test data:**
  Title, date, duration, distance and heart rate are provided.
- **Test steps:**
  1. Navigate to *http://localhost:3000/dashboard*.
  2. Click on *Log activity* button or *Log activity* link in the header.
  3. Fill a date, title, that are required, duration, distance and heart rate and other desired fields.

- **Expected result:**
  If the user fills up a duration, distance and heart rate fields, there are automatically calculated average pace, average speed, calories and steps fields. A new physical activity is created and it is displayed in the dashboard and the user is redirected to that dashboard.

TC9: Search in the activity dictionary

- **Test scenario ID and test scenario name:**
  TS7 - Check a search in the activity dictionary

- **Preconditions:**
  The user is authenticated.

- **Test data:**
  The select menus *Activity type* and *Activity subtype* are filled by the data from the activity dictionary.

- **Test steps:**

  1. Navigate to *http://localhost:3000/dashboard*.
  2. Click on *Log activity* button.
  3. Choose an activity type in the select menu.
  4. Choose an activity subtype in the second select menu from options that are generated based on the value from the activity type select menu.

- **Expected result:**
  The chosen activity type and consequently the activity subtype are displayed in the select menus.

TC10: Updating a physical activity

- **Test scenario ID and test scenario name:**
  TS8 - Check updating a physical activity

- **Preconditions:**
  The user is authenticated.

- **Test data:**
  Created physical activity logs are shown in the dashboard. Data of the each PA log, that was filled up in the successful creating a PA operation, is loaded to the update page.

- **Test steps:**

  1. Navigate to *http://localhost:3000/dashboard*.
  2. Click on *Update* button of the particular PA.
  3. The data, filled up in the creating a PA , is viewable.
  4. Fill any other desired fields.

- **Expected result:**
  If the user fills up a duration, distance and heart rate fields, there are automatically calculated average pace, average speed, calories and steps fields. The filled data are updated and the user is redirected to the dashboard.

TC11: Deleting a physical activity

- **Test scenario ID and test scenario name:**
  TS9 - Check deleting a physical activity

- **Preconditions:**
  The user is authenticated.

- **Test data:**
  Created physical activity logs are shown in the dashboard.

- **Test steps:**

  1. Navigate to *http://localhost:3000/dashboard*.
  2. Click on *Delete* button of the particular PA.
  3. A confirm window with the message "Do you want to delete this activity" is thrown.
  4. Click on *OK*

- **Expected result:**
  The particular PA is deleted from the app.

TC12: View a history of performed physical activities

- **Test scenario ID and test scenario name:**
TS10 - Check a history of performed physical activities
- **Preconditions:**
The user is authenticated.
- **Test data:**
Each physical activity shows a detail information about title of PA and its attributes, such as calories, date, distance and heart rate.
- **Test steps:**

  1. Navigate to *http://localhost:3000/dashboard*.
  2. Information of the each created PA, such as title, calories, date, distance and heart rate, is viewable.

- **Expected result:**
The list of created PAs are shown in the dashboard.

TC13: View statistic graphs of performed physical activities

- **Test scenario ID and test scenario name:**
TS11 - Check statistic graphs of performed physical activities
- **Preconditions:**
The user is authenticated.
- **Test data:**
Created physical activity logs are shown in the dashboard.
- **Test steps:**

  1. Click on *Analysis* link.

- **Expected result:**
The following parameters for the certain day for the past month in the 2D bar graph are displayed:

  1. The total number of the calories.
  2. The total number of the distances.
  3. The total number of the steps.

6.1.3 Results

The results of the user testing reveals this error in the implementation and change requests on the user interface in the application:

- When the user enters a value in the heart rate field. This value was not saved to the database.
- Improving a description of legends in the charts.
- Improving a description of physical activity attributes in the activity history.

These suggestions and the error were fixed. The participants admires usability, understandability, design and easy orientation through the application.

Chapter 7

Conclusion

The goal of this thesis was to analyze and implement the software for recording physical activity and subsequently, to analyze the performed physical activities. The development of this application includes the several steps that correspond to the goals defined in Section 1.2. Firstly, the literature research of the physical activity has been presented and these objectives has been described: a construct of physical activities, subjective and objective assessment tools to measure a physical activity. Afterwards, features of diary application, that should be included in PA diaries, have been created. These features has been pursued in detail in existing solutions on the market and a feature comparison has been provided for these apps. Based on that, applications that promote a physical activity is possible to divide into three groups that are focused on intense cardio, weight loss and tracking and analyzing workouts.

In regard to outcomes of literature and market research, functional and non-functional requirements have been assembled. In accordance with functional requirements, the user scenarios has been created. As a non-functional requirement, the focus has been mainly aimed on the usability of the application.

The application enables to log a various kinds of physical activities and for that, the activity dictionary has been created. Data calculations, such as average pace, average speed, calories ans steps, are provided from entered attributes of the physical activity. It is possible to manage personal profile and fill up body statistics. On the basis of logged physical activities, it enables to watch statistics for a certain day in the past month and therefore the user awares how much the user spent with a physical activity.

The application architecture comprises of the back-end, front-end and PostgreSQL database. The back-end and the front-end side is designed by MVC architectural pattern. For the implementation of the back-end is used Spring Boot. The back-end is linked to the front-end via REST API. The front-end is implemented in React library and Redux. Chart.js has been chosen for a data visualization.

Finally, the evaluation stage has been performed based on created test scenarios. The user testing reveals some minor errors in the application, that has been fixed. The participants appreciated usability, aesthetics and easy orientation through the app.

7.1 Future development

The topic provides many ideas for future development. One of this ideas can be a development of the mobile application, as REST API is implemented in the application. Another idea is integration with Google Fit. The Google Fit platform is a framework, which incorporates the Google Fitness Store component, a central repository comprising user fitness data. Apps on different platforms and devices can store fitness data from wearables or sensors and can access data created by other apps that is stored within to Google Fit. This requires user consent before an app can read and store fitness data [13].

Bibliography

[1] *Postgresql vs mysql - 2ndquadrant | postgresql*, <`https://www.2ndquadrant.com/en/postgresql/postgresql-vs-mysql/`>, [Online; accessed 09-May-2019].

[2] Barbara Ainsworth, Lawrence Cahalin, Matthew Buman, and Robert Ross, *The current state of physical activity assessment tools*, Progress in Cardiovascular Diseases **57** (2015), no. 4, 387 – 395, 2013 Global Congress on Physical Activity - All Hearts Need Exercise: A Global Call to Action by the AHA.

[3] B. E. Ainsworth, C. J. Caspersen, C. E. Matthews, L. C. Masse, T. Baranowski, and W. Zhu, *Recommendations to improve the accuracy of estimates of physical activity derived from self report*, J Phys Act Health **9 Suppl 1** (2012), 76–84.

[4] B. E. Ainsworth, W. L. Haskell, S. D. Herrmann, N. Meckes, D. R. Bassett, C. Tudor-Locke, J. L. Greer, J. Vezina, M. C. Whitt-Glover, and A. S. Leon, *2011 Compendium of Physical Activities: a second update of codes and MET values*, Med Sci Sports Exerc **43** (2011), no. 8, 1575–1581.

[5] ———, *2011 Compendium of Physical Activities: a second update of codes and MET values*, Med Sci Sports Exerc **43** (2011), no. 8, 1575–1581.

[6] Behavioral Risk Factor Surveillance System state coordinators and Centers for Disease Control and Prevention staff, *2018 Behavioral Risk Factor Surveillance System Questionnaire*, <`https://www.cdc.gov/brfss/questionnaires/pdf-ques/2018_BRFSS_English_Questionnaire.pdf`>, January 2018, [Online; accessed 20-October-2018].

[7] N. F. Butte, U. Ekelund, and K. R. Westerterp, *Assessing physical activity using wearable monitors: measures of physical activity*, Med Sci Sports Exerc **44** (2012), no. 1 Suppl 1, 5–12.

[8] Ariane Bélanger-Gravel, Gaston Godin, and Steve Amireault, *A meta-analytic review of the effect of implementation intentions on physical activity*, Health Psychology Review 2011 **1–32** (2011).

[9] C. L. Craig, A. L. Marshall, M. Sjostrom, A. E. Bauman, M. L. Booth, B. E. Ainsworth, M. Pratt, U. Ekelund, A. Yngve, J. F. Sallis, and P. Oja, *International physical activity questionnaire: 12-country reliability and validity*, Med Sci Sports Exerc **35** (2003), no. 8, 1381–1395.

[10] CodesJava, *Spring boot architecture diagram - spring example*, <`https://codesjava.com/spring-boot-architecture-diagram`>, 2019, [Online; accessed 03-May-2019].

[11] Logan T. Cowan, Sarah A. Van Wagenen, Brittany A. Brown, Riley J. Hedin, Yukiko Seino-Stephan, P. Cougar Hall, and Joshua H. West, *Apps of steel: Are exercise apps providing consumers with realistic expectations?: A content analysis of exercise apps for presence of behavior change theory*, Health Education & Behavior **40** (2013), no. 2, 133–139, PMID: 22991048.

[12] D. E. Conroy, C. H. Yang, and J. P. Maher, *Behavior change techniques in top-ranked mobile apps for physical activity*, Am J Prev Med **46** (2014), no. 6, 649–652.

[13] Google Developers, *Platform overview | google fit | google developers*, <`https://developers.google.com/fit/overview`>, 2018, [Online; accessed 20-March-2019].

[14] Ramesh Fadatare, *Standard project structure for spring boot projects*, <`https://www.javaguides.net/2019/01/standard-project-structure-for-spring-boot-projects.html`>, Feb 2019, [Online; accessed 08-May-2019].

[15] Federico Guede-Fernández, Bernat Carbonés, Lluís Ortís, Miguel Angel García-González, Juan Jose Ramos-Castro, and M Fernández-Chimen, *Assessment of energy expended in physical activity by a smartphone-based system*, IFMBE Proceedings **45** (2015), 893–896.

[16] Regina Guthold, Gretchen A Stevens, Leanne M Riley, and Fiona C Bull, *Worldwide trends in insufficient physical activity from 2001 to 2016: a pooled analysis of 358 population-based surveys with 1·9 million participants*, The Lancet Global Health **6** (2018), no. 10, e1077 – e1086.

[17] J. P. Higgins, *Smartphone Applications for Patients' Health and Fitness*, Am. J. Med. **129** (2016), no. 1, 11–19.

[18] Juha Hinkula, *Hands-on full stack development with spring boot 2.0 and react : build modern and scalable full stack applications using the java-based spring framework 5.0 and react*, pp. 106–107, Packt Publishing, Birmingham, UK, 2018.

[19] Andrew P. Hills, Najat Mokhtar, and Nuala M. Byrne, *Assessment of physical activity and energy expenditure: An overview of objective measures*, Frontiers in Nutrition **1** (2014), 5.

[20] Joseph Ingeno, *Software architect's handbook : Become a successful software architect by implementing effective architecture concepts*, Packt Publishing Ltd, Birmingham, 2018.

[21] Jakob Nielsen, *10 heuristics for user interface design: Article by jakob nielsen*, <`https://www.nngroup.com/articles/ten-usability-heuristics/`>, 2005, [Online; accessed 20-March-2019].

[22] ______, *Usability 101: Introduction to usability*, <`https://www.nngroup.com/articles/usability-101-introduction-to-usability/`>, Janurary 2012, [Online; accessed 20-March-2019].

[23] R. P. Joseph, N. H. Durant, T. J. Benitez, and D. W. Pekmezi, *Internet-Based Physical Activity Interventions*, Am J Lifestyle Med **8** (2014), no. 1, 42–68.

[24] LR Keytel, JH Goedecke, TD Noakes, H Hiiloskorpi, R Laukkanen, L van der Merwe, and EV Lambert, *Prediction of energy expenditure from heart rate monitoring during submaximal exercise*, Journal of Sports Sciences **23** (2005), no. 3, 289–297.

[25] A. McTiernan, *Cancer prevention and management through exercise and weight control*, Nutrition and Disease Prevention, pp. 18–19, CRC Press, 2016.

[26] Jakob Nielsen, *Usability engineering*, Academic Press, Boston, 1993.

[27] Chris Northwood, *The full stack developer*, p. 73, Apress, 2018.

[28] World Health Organization, *Global Recommendations on Physical Activity for Health*, <http://apps.who.int/iris/bitstream/handle/10665/44399/9789241599979_eng.pdf?sequence=1>, 2010, [Online; accessed 15-October-2018].

[29] ———, *Medical Dictionary for the Health Professions and Nursing*, <https://medical-dictionary.thefreedictionary.com/energy+expenditure>, 2012, [Online; accessed 11-October-2018].

[30] ———, *Global action plan on physical activity 2018–2030: more active people for a healthier world*, <http://apps.who.int/iris/bitstream/handle/10665/272722/9789241514187-eng.pdf>, 2018, [Online; accessed 14-October-2018].

[31] ———, *Physical activity*, <http://www.who.int/news-room/fact-sheets/detail/physical-activity>, February 2018, [Online; accessed 11-October-2018].

[32] ———, *What is Moderate-intensity and Vigorous-intensity Physical Activity?*, <https://www.who.int/dietphysicalactivity/physical_activity_intensity/en/>, 2018, [Online; accessed 15-October-2018].

[33] Sherry Pagoto and Gary G. Bennett, *How behavioral science can advance digital health*, Translational Behavioral Medicine **3** (2013), no. 3, 271–276.

[34] John P. Porcari, Cedric X. Bryant, and Fabio Comana, *Exercise physiology*, F.A. Davis Company, 2015.

[35] A. C. Pinheiro Volp, F. C. Esteves de Oliveira, R. Duarte Moreira Alves, E. A. Esteves, and J. Bressan, *Energy expenditure: components and evaluation methods*, Nutr Hosp **26** (2011), no. 3, 430–440.

[36] Shanhu Qiu, Xue Cai, Xiang Chen, Bingquan Yang, and Zilin Sun, *Step counter use in type 2 diabetes: a meta-analysis of randomized controlled trials*, BMC Medicine **12** (2014), no. 1.

[37] G.Karthik Reddy and K. Lokesh Achari, *A non invasive method for calculating calories burned during exercise using heartbeat*, 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO).

[38] Carolyn Rabin and Beth Bock, *Desired features of smartphone applications promoting physical activity*, Telemedicine and e-Health **17** (2011), no. 10, 801–803, PMID: 22010977.

[39] *File structure - react*, <`https://reactjs.org/docs/faq-structure.html`>, 2019, [Online; accessed 09-May-2019].

[40] *Actions - redux*, <`https://redux.js.org/basics/actions`>, 2018, [Online; accessed 09-May-2019].

[41] *Code structure - redux*, <`https://redux.js.org/faq/code-structure`>, 2018, [Online; accessed 09-May-2019].

[42] Nicola Ridgers and Stuart Fairclough, *Assessing free-living physical activity using accelerometry: Practical issues for researches and practitioners*, European Journal of Sport Science - EUR J SPORT SCI **11** (2011), 205–213.

[43] Miriam Reiner, Christina Niermann, Darko Jekauc, and Alexander Woll, *Long-term health benefits of physical activity – a systematic review of longitudinal studies*, BMC Public Health **13** (2013), no. 1, 813.

[44] L. G. Sylvia, E. E. Bernstein, J. L. Hubbard, L. Keating, and E. J. Anderson, *Practical guide to measuring physical activity*, J Acad Nutr Diet **114** (2014), no. 2, 199–208.

[45] *Spring boot reference guide*, <`https://docs.spring.io/spring-boot/docs/2.0.8.RELEASE/reference/htmlsingle/#using-boot-structuring-your-code`>, 2018, [Online; accessed 08-May-2019].

[46] U.S. Department of Health and Human Services, *2018 Physical Activity Guidelines Advisory Committee Scientific Report*, <`https://health.gov/paguidelines/second-edition/report/pdf/PAG_Advisory_Committee_Report.pdf`>, February 2018, [Online; accessed 4-November-2018].

Appendix A

Use Cases

UC1: Sign in to the app

- **Description:**
  A user creates a new account in the application.
- **Trigger:**
  The user presses the "Sign Up" link or button.
- **Preconditions:**
  The user has not created the account.
- **Normal Flow:**
  1. The user fills up all required fields, such as first name, last name, email address, password, confirm password, age, weight and gender.
  2. The user submits the filled registration form.
- **Postconditions:**
  The account is created and the user is redirected to the login page.

UC2: Login to the app

- **Description:**
  A user logs in to the application.

- **Trigger:**
  The user presses the "Log In" link or button.
- **Preconditions:**
  The user is unauthenticated.
- **Normal Flow:**

  1. The user enters the valid email address and password.
  2. The user submits the login.
- **Postconditions:**
  The user is already logged to the app and redirected to the dashboard.

UC3: Logout from the app

- **Description:**
  A user logs out from the application.
- **Trigger:**
  The user presses the "Logout" link.
- **Preconditions:**
  The user is already logged in to the app.
- **Normal Flow:**

  1. The user submits the logout.
- **Postconditions:**
  The user is logged out from the app and redirected to the landing page.

UC4: Reset a forgotten password

- **Description:**
  The user forgot the password. After the user enters a valid email address, that was filled up in the registration form, the link with the reset token is sent to that user's email address. This link enables to reset the password and the user can enter a new password, that is stored to the database.
- **Trigger:**
  The user presses the "Forgot your password?" link.

- **Preconditions:**
  The user is unauthenticated.

- **Normal Flow:**

  1. The user enters the valid email address.
  2. The reset link is sent on that email address and the user clicks on this link.
  3. The user fills up a new password and confirm password.

- **Postconditions:**
  A new password is stored to the database and the user can log with new password to the app.

UC5: View and update the user profile

- **Description:**
  The user views and updates the personal profile and body statistics.

- **Trigger:**
  The user presses the user icon.

- **Preconditions:**
  The user is already logged in to the app.

- **Normal Flow:**

  1. The user views the filled personal data from the registration process.
  2. The user submits these personal data.

- **Alternative Flow:**

  1. The user updates some of the following fields, such as first name, last name, age and gender or adds the body statistics, like height, weight, resting heart rate or body fat.
  2. The user submits the updated personal data.

- **Postconditions:**
  The data, that the user decides to update, are changed.

UC7: Search in an activity dictionary

- **Description:**
  The user searches a desired physical activity in an activity dictionary during creating the physical activity.
- **Trigger:**
  The user presses the "Activity type" select menu.
- **Preconditions:**
  The user is already logged in to the app.
- **Normal Flow:**

  1. The user chooses an activity type in the select menu.
  2. Based on the chosen activity type, they are generated activity subtypes in the second select menu and the user chooses the activity subtype.

- **Postconditions:**
  The chosen activity type and consequently the activity subtype are displayed in the select menus.

UC8: Update an existing physical activity

- **Description:**
  The user updates an existing physical activity. The data for the particular PA is loaded to the form. If the user fills up a duration, distance and heart rate fields, there are automatically calculated average pace, average speed, calories and steps fields.
- **Trigger:**
  The user presses the "Update" button.
- **Preconditions:**
  The user is already logged in to the app.
- **Normal Flow:**

  1. The user searches a physical activity type in an "Activity Type" select menu.
  2. Based on the selected activity type, the app generates other advanced fields.

3. The user searches a desired physical activity in "Activity Subtype" select menu and fill out generated fields.

- **Alternative Flow:**

  1. The user did not find a desired physical activity in an "Activity Subtype" select menu.
  2. The user type the name of the physical activity to the "Title" field.
  3. The user fill out default fields.

- **Postconditions:**
  The physical activity is updated and the updated data is displayed in a history of the performed physical activities.

UC9: Delete an existing physical activity

- **Description:**
  The user deletes an existing physical activity.

- **Trigger:**
  The user presses the "Delete" button.

- **Preconditions:**
  The user is already logged in to the app.

- **Normal Flow:**

  1. A confirm window with the message "Do you want to delete this activity" is thrown.
  2. The user submits this window message.

- **Postconditions:**
  The physical activity is deleted from the dashboard and in the database.

Appendix B

Previews of the user interface

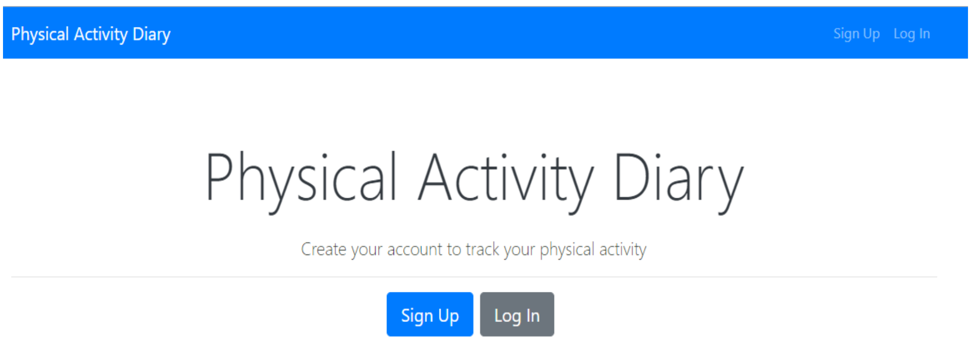

Figure B.1: Landing page

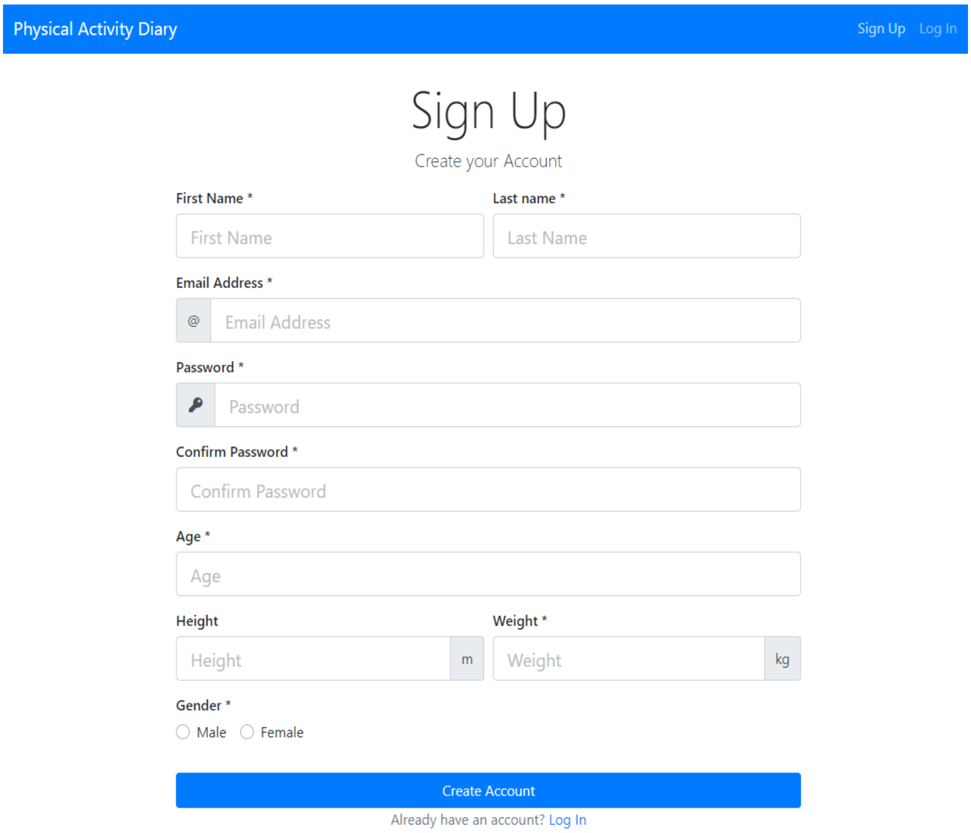

Figure B.2: Sign up page

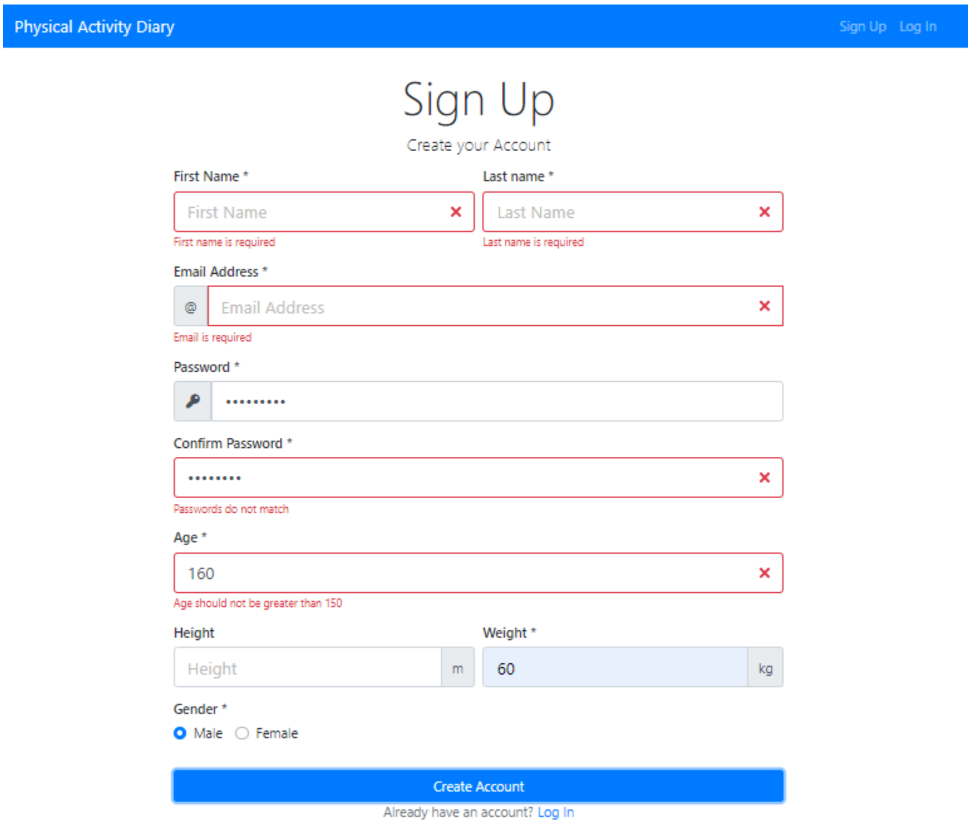

Figure B.3: Validation of the sign up form

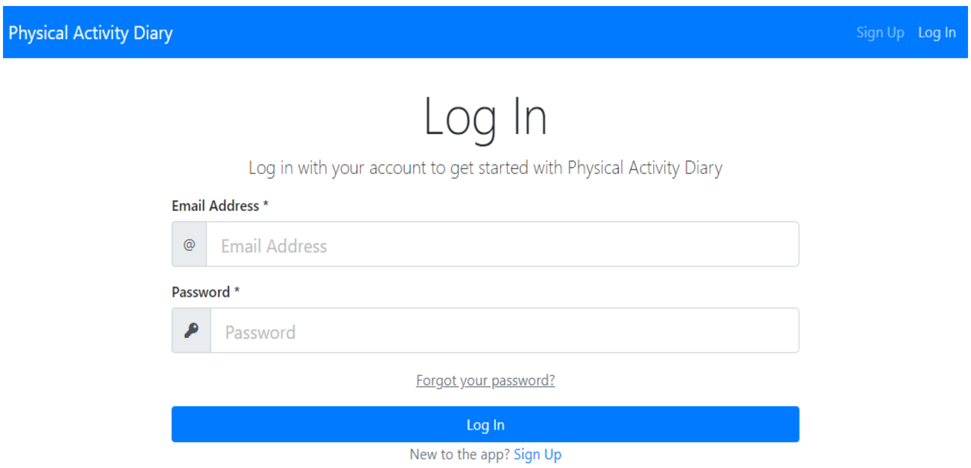

Figure B.4: Log in page

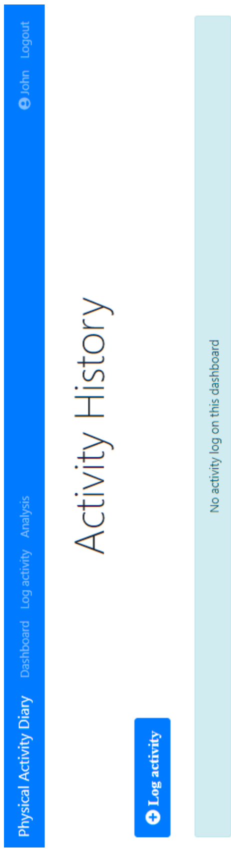

Figure B.5: Welcome dashboard

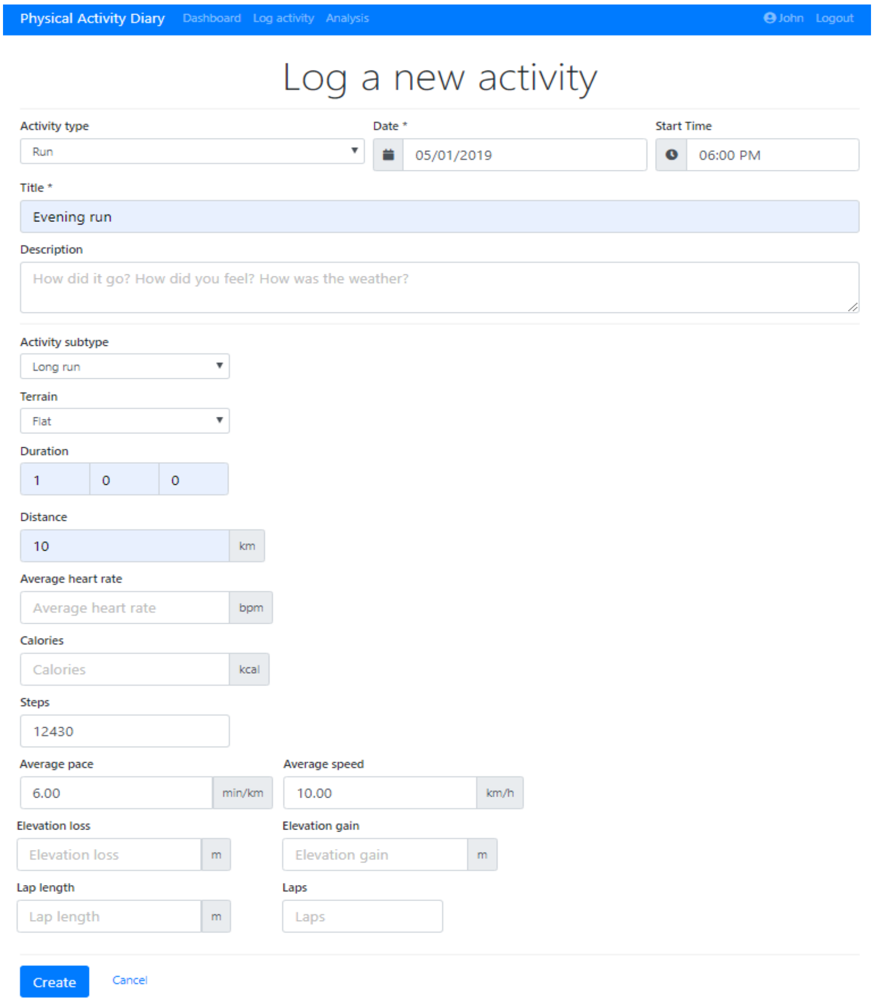

Figure B.6: Logging a running activity with automatic data calculations

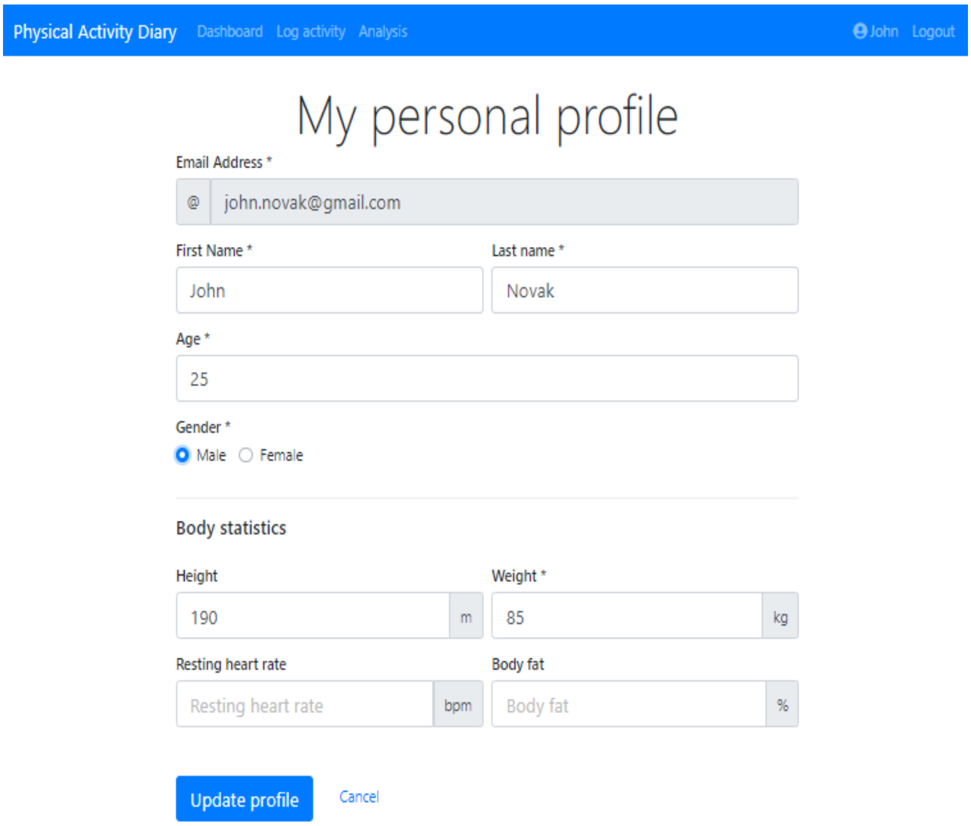

Figure B.7: Personal profile with body statistics

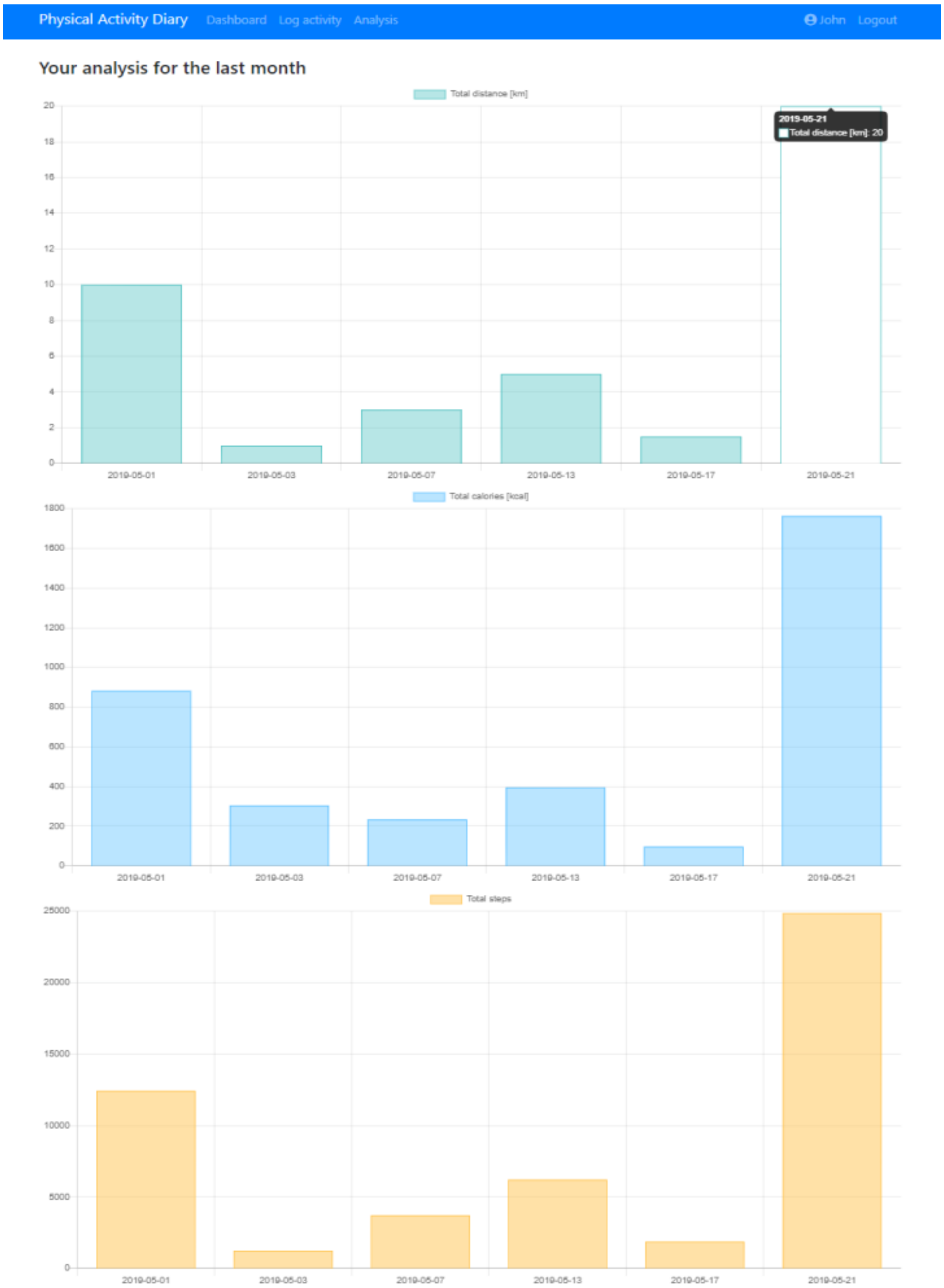

Figure B.8: Personal analysis

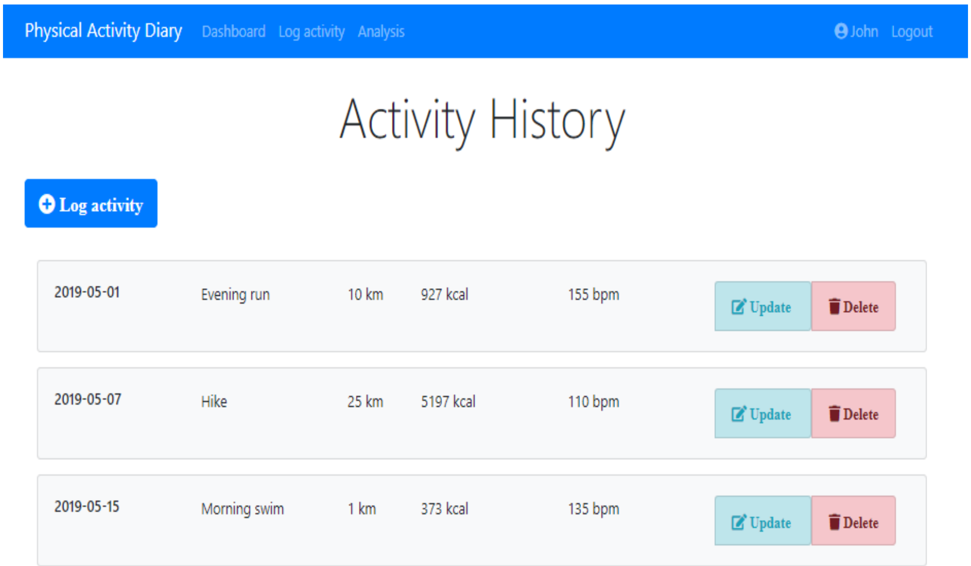

Figure B.9: Dashboard with the activity history

Appendix C

Contents of the enclosed CD

```
|--readme.txt..............the file with the CD content description
|
|--source.........................the directory of the source codes
|  |-- pa_diary..................the implementation of the PA diary
|  |-- thesis............the directory of LaTeX thesis source codes
|
|--text.................................the thesis text directory
|  |--DT_Barbora-Rolandova-2019.pdf...the thesis text in PDF format
```