UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Computer Science

Sáhka

Architecting of a Soccer Performance Development Software Toolkit

Markus Bergum Hilbertsen

INF-3981 Master's Thesis in Computer Science - June 2022

“If you want to look like an athlete, you have to train like an athlete.”
– Jeff Cavaliere — ATHLEAN-X

Abstract

Elite football today is infused with technologies for collecting data on players' performance and health. Technologies such as wearable health and fitness trackers, full-body medical scans, and positional systems mounted around a stadium give coaches a wide range of accurate data about their players. However, coaches do not have the time or ability to analyze all this data for each player to give them individualized training schedules. Hence, coaches need tools to collect, analyze, summarize, and present the data to them in a much more consumable format.

Existing systems within sports technology are isolated, only collecting their own data and using it for their own purposes. Hence, Sáhka will break out of this norm and create a novel system within this domain.

Sáhka is a system that federates relevant data from several different sources. The data is processed, stored, analyzed, and presented visually to the users. Additionally, by collecting a large quantity and variety of data, Sáhka acts as a Big Data repository for real-time sports data. In the future, Sáhka will be used as a platform for training Machine Learning algorithms.

Contents

List of Figures

List of Abbreviations

2FA Two-Factor Authentication

30-15IFT 30-15 Intermittent Fitness Test

AI Artificial Intelligence

API Application Programming Interface

BJ Broad Jump

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart

CHO Carbohydrates

CMJ Countermovement Jump

CSG Cyber Security Group

DAG Directed Acyclic Graph

DNS Domain Name System

DOS Denial-of-Service

DXA Dual-Energy X-Ray Absorptiometry

GDPR General Data Protection Regulations

GPS Global Positioning System

HTTP Hypertext Transfer Protocol

IOS iPhone Operating System

IOT Internet of Things

ISM Industrial, Scientific and Medical

JSON JavaScript Object Notation

LSTM Long Short-Term Memory

MB Megabyte

MDCN Microsoft Development Centre Norway

PC Personal Computer

REST Representational State Transfer

RPC Remote Procedure Call

SJ Squat Jump

SRPE Session Rating of Perceived Exertion

TCP Transmission Control Protocol

TIL Tromsø Idrettslag

UiT University of Tromsø

URL Uniform Resource Locator

XML Extensible Markup Language

1 Introduction

Football is currently the most popular sport in the world. Naturally, this has led to fierce competition, where teams are always looking for something to give them an edge over the others. Technology is one area where this competition has been a huge driving force, leading to the development of everything from wearable sensors to mounted positional systems to full-body scans. These systems help coaches make more informed decisions about their players by measuring and keeping track of their performance and health. Specifically, these systems are valuable when researching training methods and amounts for the best performance increase.

However, with the incoming abundance of systems for tracking performance and health, coaches have difficulty keeping up with the amount of available data. Coaches are, after all, still human and, as such, have a limited amount of time and ability to process data. Hence, there has been a void in the industry for systems that aid coaches in this process. Coaches now need systems that can help them by collecting, analyzing, and presenting all the available data to them in one location. This should let coaches quickly get the vital information they need on each player to make an informed decision about what they should focus on going forward.

This thesis is based on some work performed in a Capstone thesis [1]. The Capstone thesis consisted of exploring this research domain and developing an initial prototype of Sáhka[1]. This thesis will refine and expand upon the initial prototype, integrating several more data sources and adding an analysis component to the system.

Furthermore, by collecting such a high quantity and variety of data, Sáhka becomes a Big Data platform for real-time multimodal sports data. This avenue will be explored further in the future, where several theses will use Sáhka as a platform for training Machine Learning algorithms. Additionally, two theses have already been defined, which will continue working with Sáhka, focusing on different solutions and integration of video.

This thesis is written in collaboration with Tromsø Idrettslag (TIL) Men's A-team, a professional Norwegian football team. They used Sáhka throughout its initial development during the Capstone thesis and have continued to use it during the development of this thesis. While using Sáhka, their coaches have provided valuable feedback on what type of information is essential to them and how they want it visualized. Due to this collaboration, the data sources integrated with Sáhka are limited by TIL and the systems they use.

1.1 Problem Definition

The majority of sports tools available today are focused on collecting and presenting one type of performance or health data to coaches. However, using several different tools to get an overview of a player's performance and health is disorganized and time-consuming for a coach with over twenty players in a team and limited time.

This thesis aims to rectify this problem by developing a tool that collects player data from several different sources and presents it to coaches in one application. Hence, the following problem definition is formalized:

This thesis will investigate the design and implementation of a system that collects and analyses multimodal data about football players from several different sources to aid coaches in providing intervention of training load and practices personalized to individual athletes. Particular focus will be on developing a proof-of-concept prototype that provides coaches with as much relevant information as possible in an easily consumed format.

1. Sáhka means *conversation* in the Northern Sámi language

1.2 Limitations

The scope of this thesis will be to continue the work on Sáhka by refining and improving the prototype developed and integrating more data sources. Specifically, this thesis will explore if and how six different data sources can be integrated; DXA, physical performance tests, nutrition, video clips, player objectives, and a journal.

Furthermore, this thesis will expand upon the previous prototype by analyzing the data collected to notify coaches if one of their players meets specific conditions. This analysis component will resemble a publish-subscribe model, where coaches can subscribe to specific events and be notified when they occur.

1.3 Method

The final report of the ACM Task Force on the Core of Computer Science [2] presents three paradigms by which computing as a discipline should be divided:

> **Theory** is rooted in mathematics and consists of defining a problem, hypothesizing whether possible relationships are true, and interpreting the results.
>
> **Abstraction** is rooted in the experimental scientific method and consists of forming a hypothesis, constructing a model and making a prediction, designing an experiment and collecting data, and analyzing the results.
>
> **Design** is rooted in engineering and consists of defining requirements and specifications, designing and implementing the system, and testing the system.

This thesis is rooted in the design paradigm. Given a problem definition with specific requirements, a prototype system will be designed, implemented, and evaluated.

1.4 Research Context

This thesis is written as a part of the Cyber Security Group[2] (CSG) at The Arctic University of Norway[3] (UiT). CSG undertakes high-impact inter-disciplinary and inter-faculty research and innovation at the intersection of computer science, health informatics, statistics, medicine, sports science, and law. The long-term objective of CSG is to provide new knowledge, research tools, and innovative, disruptive technologies in this convergence area.

CSG carries out experimental computer science systems research related to the architecture and construction of scalable, efficient, fault-tolerant, privacy-preserving, and secure distributed systems. Moreover, CSG evaluates their engineered systems by putting them through actual deployment with realistic use.

CSG works on a wide range of fundamental distributed systems problems related to trust and dependability. Trust-by-design is essential when a distributed system is initially architected, designed, and built. CSG's research focuses on building and investigating robust and efficient software infrastructures, trusted data management and storage, secure networking, multi-media support, and support for privacy-preserving algorithmic analysis.

CSG's chosen methodology to solve a concrete research problem is primarily an experimental systems approach where prototypes are constructed and experimentally evaluated. However, a more theoretical approach might also be used if it is more natural for a specific problem.

One of CSG's main partners is the Corpore Sano Centre[4], which undertakes high-impact inter-disciplinary and inter-faculty research and innovation at the intersection of computer science, sports science, and medicine. Concerning sports, such as football, this research investigates how performance data from players can be collected and analyzed to provide insights for injury prevention, training personalization, and evidence-based decisions for team performance improvements.

The consortium affiliated with the Corpore Sano Centre aims to innovate on medicine and the use of technology, where Microsoft Development Centre Norway (MDCN) and Forzasys[5] are key industrial partners.

2. https://site.uit.no/arcsecc/
3. https://en.uit.no/startsida
4. https://site.uit.no/corporesano/
5. https://forzasys.com/

In the context of the Corpore Sano Centre, CSG investigates structuring techniques for future-generation large-scale information access applications. This includes fundamental research issues like how to best partition an application into a set of cooperating modules, how to optimize interaction among them, how and where to deploy them, how to interact with the users, how to provide integrity, security, and auditing, and how to ensure fault-tolerance and privacy.

CSG has extensive experience developing systems for the entire technology stack. In the low-level end, Vortex [3, 4] is an entirely event-driven multiprocessor operating system that efficiently and automatically balances load across processors. In the DNS domain, Jovaku [5] reduces latency by introducing a truly global caching infrastructure built on the existing DNS system. Further out in the cloud, Balava [6] manages computations that span multiple clouds, public and private, and involve data with confidentiality constraints. Diggi [7, 8], incorporating Arm TrustZone [9] and SGX [10, 11] enables IOT devices, smartphones, computers, and cloud solutions from different vendors to seamlessly connect and integrate in a privacy-preserving and secure manner.

Specific to the sports technology domain, CSG has contributed systems for over a decade. Muithu [12, 13] is a real-time activity annotation system designed to be intuitive while imposing minimum effort on the user. Coaches use a smartphone interface to tag events on the pitch, and Muithu uses hindsight recording to extract a video of the event, which is made available to the coaches during half-time or after the session. Bagadus [14, 15] integrates a sensor system, an analytics annotation system, and a video processing system that uses a video camera array. The system can play events from matches, follow individual players, and create video summaries based on queries to the sensor system. DAVVI [16, 17] delivers multi-quality video content in a torrent-similar way. Through applied search, personalization, and recommendations, users can efficiently search for and retrieve specific football events from extensive video archives for almost immediate playback.

In a more pure sports science context, Ivan Baptista, using Metrix [18], which provides real-time analysis of physical performance parameters, showed that physical demands in official football matches vary greatly across playing positions [19]. Consequently, a higher level of training specificity is required to prepare players of different positions to meet their match demands.

Finally, Sigurd Pedersen, who will be mentioned throughout this thesis due to his current position as TIL's physical coach, has made several contributions to studying the effects of various strength training methods on performance in high-level male and female football players [20].

1.5 Contributions

This thesis develops a system prototype that collects and analyses football player data to aid coaches in monitoring, intervening, and personalizing training programs for individual athletes. The system integrates several different types of data, including, but not limited to, positional data, wellness, body composition, nutrition, and videos, to provide coaches with as much relevant information as possible. Furthermore, this thesis demonstrates the possibility of running algorithmic analysis, customized by coaches, on the data collected to notify them of specific events regarding their players.

1.6 Outline

The rest of this thesis is structured as follows.

Chapter 2 will provide some context for this thesis, including technical background information and other related work.

Chapter 3 will define the specific requirements for this thesis, including functional and non-functional requirements.

Chapter 4 will detail the design and implementation of Sáhka and why these design choices were made.

Chapter 5 will present some experiments performed and their results.

Chapter 6 will discuss some of the relevant topics relevant to this thesis.

Chapter 7 will conclude this thesis with some final words and present suggestions for possible future work.

2

Background & Related Work

This chapter will give some background on how and why training monitoring is used in elite football today. Additionally, some background into the different data sources explored in this thesis will be presented. Finally, the contributions of the Capstone thesis Sáhka is based on will be described.

2.1 Training Monitoring in Elite Football

Training monitoring is the process of monitoring an athlete's activity and load during a match or training session [21]. This monitoring is essential to determine whether athletes adapt to their training programs, find the most effective training methods, measure fatigue and the corresponding need for recovery, and minimize the risk of injuries [22]. Additionally, even though a team may perform the same training session, it is crucial to measure the *individual* athletes because they may respond differently to the same load [23].

With the influx of new technology, training monitoring has become more accessible and accurate than ever. During training, these technologies provide data on external load factors, such as speed, distance, and power, and internal load factors, such as heart rate, oxygen uptake, and lactate levels [24]. Additionally, advances in smartwatches enable them to track sleep and rest outside of training, which is essential for making sure athletes can recover enough.

The data collected through these technologies can be divided into quantitative and qualitative data.

2.1.1 Quantitative

Quantitative data gives an objective numerical measure of something, often a player's performance or health in football. One of football's most common quantitative data types comes from positional systems. These systems track a player's position on the pitch during a match or training session, giving insights such as the player's speed, distance traveled, turns, and accelerations. Positional systems today primarily use one of two technologies to track the players: radio and GPS.

GPS systems, such as STATSport[1], use satellites to determine the position of the sensor worn by players. Sample the players' position often, and it can be used to calculate speed, distance, and acceleration. GPS systems benefit from being simple to use, requiring only the sensor itself to be worn because the satellites are already in orbit around the earth. However, depending on the number of satellites available and the sensor's sampling rate, these GPS systems may suffer in terms of accuracy and reliability, especially for assessing short high-speed straight-line running and efforts involving a change of direction [25].

Radio-based systems, such as ZXY Sports Tracking[2], use radio-based signals to provide positional tracking of players. Like GPS systems, radio-based systems use sensors worn by the players to track their positions. However, this also means that GPS systems are limited to outdoor locations with good satellite coverage. Additionally, the radio-based sensors communicate with stationary radio receivers mounted around the pitch, which computes positional data based on these signals. Due to higher sampling rates and being much closer to the radio receivers, radio-based systems provide significantly more accurate data compared to GPS systems [26, 27]. However, these systems are costly to construct and not portable because they require radio receivers mounted around the pitch.

1. `https://statsports.com/`
2. `https://tracab.com/`

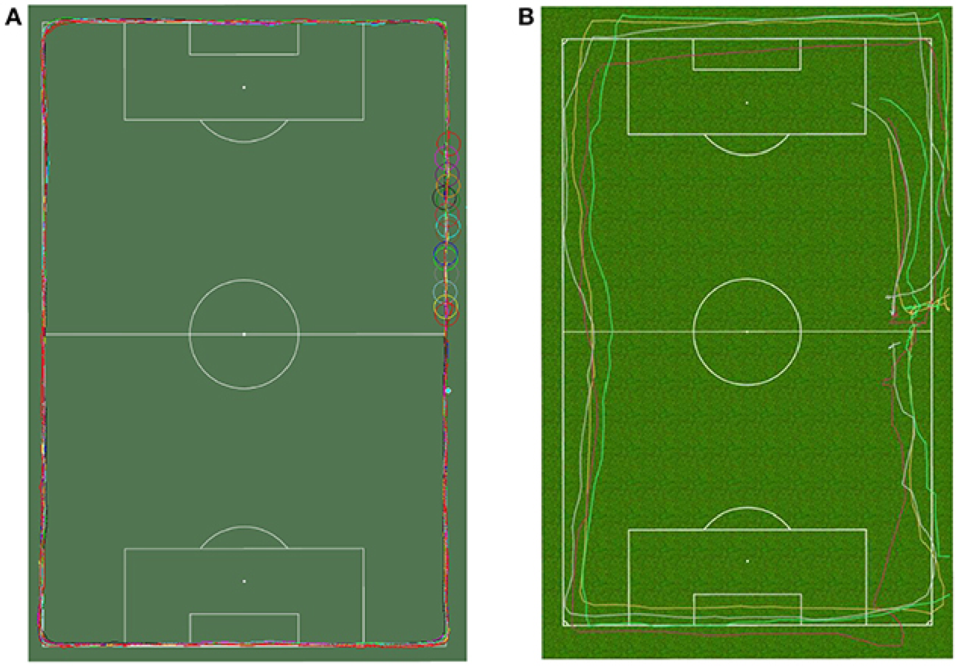

Figure 2.1: Comparison of tracking technologies with 12 players running at the side and endlines of the pitch at Alfheim Stadium [28]. (A) ZXY tracking tracking results, 12 players shown. (B) Polar Team Pro GPS tracking results (5 of 12 players shown).

Figure 2.1 shows the result of a study that compared the accuracy of radio waves against GPS in sampling tracking data [28]. The study had twelve football players jog clockwise around the pitch at Alfheim, following the side- and end-lines, wearing both the Polar Team Pro 10 GHz GPS system and ZXY. Figure 2.1 (B) shows that the GPS tracking deviates significantly from the actual trajectory of the players, whereas ZXY's tracking in Figure 2.1 (A) follows the side- and end-lines much more closely.

Other quantitative data sources include DXA scans, physical performance tests, and nutrition, which will be explained more in later sections.

2.1.2 Qualitative

Qualitative data is non-numeric, subjective data often collected through, for example, questionnaires, interviews, or video footage. A popular qualitative measure in football is Session Rating of Perceived Exertion (SRPE), where a player reports how much effort they feel they exerted during a session. This type of data may provide helpful situational information that quantitative data does not include. For example, two training sessions with the same quantitative metrics may have affected the player differently if, for example, the player was already tired or had little sleep before one of the sessions, which only a difference in the SRPE ratings would reflect.

Qualitative data can also be used in combination with quantitative data. SportSence [29], for example, combines qualitative video data with quantitative tracking data to improve data analytics.

2.2 Data Sources

2.2.1 Positional Data

ZXY Sports Tracking is a radio-based positional system for tracking players' positions on the pitch. The system consists of sensor belts worn by the players and radio receivers mounted around the pitch [26] depicted in Figures 2.2 and 2.3. They use this system instead of GPS systems due to its higher accuracy.

The ZXY installation at Alfheim is based on the 2.45 GHz ISM band for radio communication and signal transmissions [26]. Furthermore, each radio receiver has an approximately 90-degree field of view, and they are arranged with overlapping zones to provide high immunity to occlusions and signal blocking, which increases reliability. Each radio receiver uses the radio signals received from the belts to compute their positional data. The positional sampling rate is currently set to 20 Hz. The sensor belts also come equipped with an accelerometer, gyroscope, heart-rate sensor, and compass to provide additional data.

Figure 2.2: ZXY sensor belt worn by players [26].

Figure 2.3: ZXY radio receivers mounted around the pitch [26].

2.2.2 PmSys

PmSys is a digital monitoring system that collects, analyses, and presents health data [30]. TIL has been using PmSys on and off for years as a tool for coaches to monitor their players. The data is collected through an app where the athletes answer questions about their perceived training load, wellness, and injuries.

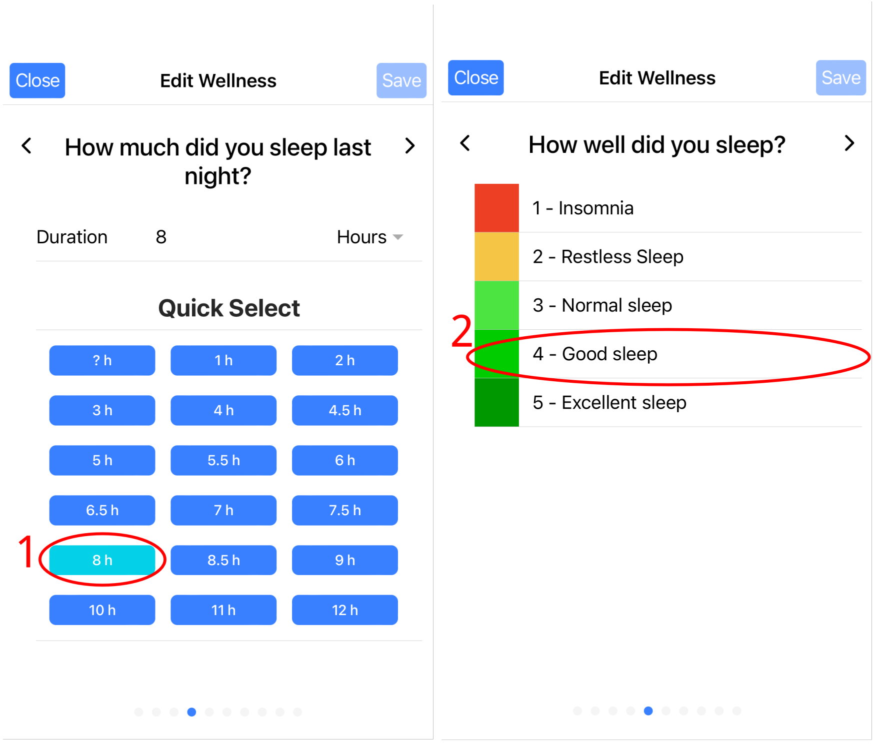

Figure 2.4: Pages in PmSys app where players log their sleep amount and quality.

Figure 2.4 shows two of the questions asked in the PmSys app. The first question, on the left, asks how many hours the player slept the previous night. The player then chooses the closest option from the list of one to twelve hours in half-hour increments. In this example, the player selected eight hours, as highlighted by the red circle.

The second question, on the right, asks the player to rate the quality of their sleep on a scale from one to five, one being the worst quality and five the best. For example, selecting option four, highlighted by the red circle, would indicate that the player had a good sleep but not the best.

PmSys also includes a web portal for coaches to monitor their athletes [30]. The web portal includes visualization and analysis capabilities for a single player or all team members [31]. The visualization includes wellness, injuries, illnesses, session participation, and essential training load indicators such as daily and weekly load, acute load, chronic load, acute chronic workload ratio, monotony, and strain.

Additionally, the portal includes a messaging system, allowing coaches to send messages that the players receive as push notifications in the PmSys app. These messages can be individual messages to a single player or scheduled messages, for example, reminding players to answer the survey every day.

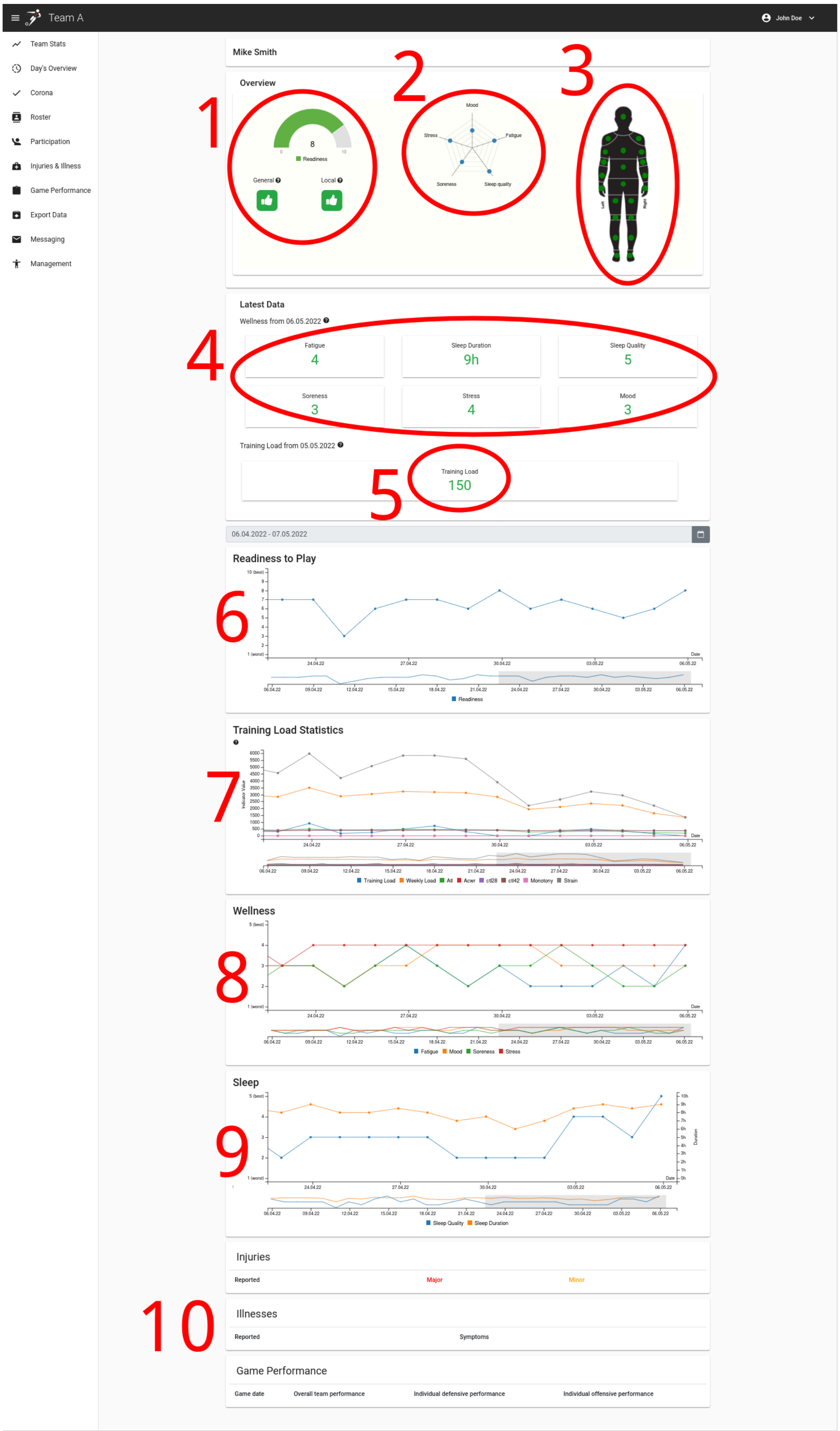

Figure 2.5: PmSys trainer portal showing a single player's data.

Figure 2.5 shows a player's profile page in PmSys' web portal.

1. Shows the player's current readiness to train on a scale from one to ten, one being not ready and ten being completely ready.

2. This spider diagram shows some of the player's latest wellness metrics, including mood, fatigue, sleep quality, soreness, and stress.

3. This human figure contains a circle on each body part. The circles will be red if the player is currently injured on that body part and green otherwise. Since all the circles of this figure are green, the player is not injured anywhere.

4. This section shows the player's latest wellness metrics in numeric form, including fatigue, sleep duration, sleep quality, soreness, stress, and mood.

5. This number shows the player's training load the previous day. The training load is a sum of several different load metrics, including acute training load (ATL), chronic training load (CTL), acute chronic workload ratio (ACWR), monotony, and strain.

6. This graph shows the player's readiness to play over time.

7. This graph shows the different training load metrics over time, including daily training load, weekly training load, ATL, CTL, ACWR, monotony, and strain.

8. This graph shows some of the player's wellness metrics over time, including fatigue, mood, soreness, and strain.

9. This graph shows the player's sleep duration and quality over time.

10. This section lists any of the player's injuries, illnesses, and game performances. Currently, this player has no data here.

2.2.3 Abel

Abel[3] is a digital personal trainer that assists with nutrition and training. Abel plans users' nutrition and training programs according to their goals, such as losing weight or gaining muscle. For nutrition, Abel contains recipes for thousands of dishes and a grocery list feature. For training, Abel can plan

3. https://abel.fit/

training sessions, and every exercise it suggests comes with an instructional video demonstrating how to perform the exercise. Additionally, Abel contains features for coaches to plan their athletes' nutrition and training and monitor their progress.

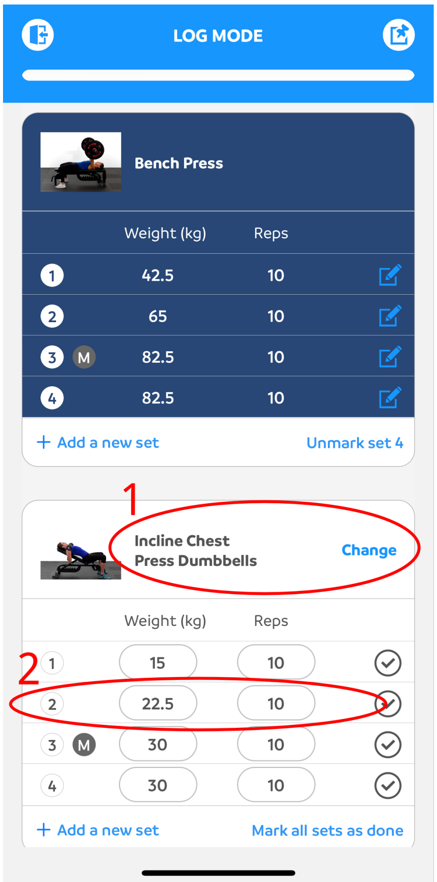

Figure 2.6: page in abel app where players log training sessions.

Figure 2.6 shows a page in the Abel app for logging a strength training session. The first highlight shows where the user can select the exercise performed. In the second highlight, the user enters the weight lifted and the number of reps performed.

2.2.4 DXA

Dual-Energy X-ray Absorptiometry (DXA) is an x-ray scan that measures body composition in terms of bone mineral content, bone mineral density, lean mass, and fat mass [32]. Body composition is highly relevant in sports regarding athletes' performance and health. Lean mass contributes positively to an athlete's strength and performance, whereas excess body fat negatively impacts performances such as sprinting, jumping, and aerobic capacity. Hence, having an accurate measure of an athlete's muscle and fat mass can be helpful when planning the athlete's training program going forward.

DXA scans also measure the composition of individual body parts. This is useful to detect muscle asymmetry between, for example, the left and right leg, which is crucial to detect as early as possible due to the increased risk of injury associated with muscle imbalances [33].

2.2.5 Performance Tests

Football is a physically demanding sport, requiring the players to be capable in several aspects of fitness, including aerobic and anaerobic power, muscle strength, flexibility, and agility [34]. Each player will, and should, perform differently in these aspects, depending on their playing position and the team's style of play. Hence, coaches need to consider each player's current and ideal performance in these aspects when planning their short- and long-term training programs.

Measuring a player's current performance is often done using several physical tests. Each test objectively measures some aspect of the player's physical performance, and when combined, they give an overview of the player's strengths and weaknesses. The coaches can then use this information to create optimal individualized training programs. Additionally, re-testing an athlete after a training program to compare with the results from before the program can give an objective measurement of how well the program works on specific fitness aspects.

The specific tests integrated with Sáhka are CMJ, Squat Jump (SJ), Broad Jump (BJ), Benchpress, Pull-Up, 15-meter sprint, and the 30-15 Intermittent Fitness Test ($30\text{-}15_{IFT}$) [35]. These tests, specifically, were chosen by Sigurd Pedersen, TIL's physical coach, because the players perform them regularly.

2.2.6 Nutrition

Football players can run on average 12 km during one match [36]. These feats take an enormous toll on the body and are not feasible without adequate fuel. Fuel that the body can only get through food. Hence, nutrition is a critical factor for football players to perform at their best.

Carbohydrates (CHO) provide the energy the body uses. It is stored within the muscles and liver and used up during physical activity. The amount of CHO needed each day depends on body composition (height and weight) and activity levels. Naturally, since football players are very active, they require a lot of CHO, especially before a match.

Proteins are the building blocks of the body's muscles. They are necessary for the body to make new muscle tissue and repair old or damaged tissue. Hence, if football players do not maintain a net positive protein balance, it is likely to result in a decrease in muscle mass, which is detrimental to a player's performance [37].

Fat, perhaps contrary to popular belief, is also essential to the body. While excess fat, especially when stored as subcutaneous fat, is unhealthy, the right amount of good fat is critical to protect the body's organs, support cell growth, regulate hormones, maintain the immune system, and reduce inflammation. Furthermore, fat is required to absorb fat-soluble vitamins, such as A, D, E, and K, which the body needs to function normally.

How much of these three nutrients the body needs depends on several factors, including age, gender, and activity level. Many systems are available today for tracking nutrition, for example, MyFitnessPal[4] and Lifesum[5]. These systems track meals and break them down into calories and micro- and macro-nutrients. This lets users keep track of and plan their meals according to their specific requirements.

2.2.7 Video Clips

Analyzing player and team performance after a match or training is one of the coaches' many jobs. One of the methods used for this is to view video clips of critical events such as, for example, goals, shots, and tackles. This allows the coach to replay the event multiple times to catch more details and identify what was good and what could be improved.

Systems such as Eliteserien Highlights[6] use a lot of manpower to find and tag events manually after a match. However, this is not feasible during training, where manpower is more limited than during matches. Systems, such as Muithu [12], aim to make recording critical events so simple that it can be used during training. It does this by allowing the head coach to tag events as they happen in real-time through a few clicks on their phone. The system then records

4. `https://www.myfitnesspal.com/`
5. `https://lifesum.com/`
6. `https://highlights.eliteserien.no/`

a specified amount of time before and after the tag and saves it as a video clip.

2.3 Directed Acyclic Graph

In graph theory, a field of mathematics, a *graph* is defined as two sets: a finite set of vertices and a set finite of edges, where each edge is connected to two vertices [38]. Graphs are often represented by circles as vertices and lines between them as edges. *Directed* means that the edges have a specific direction, often represented by an arrow instead of a line. Finally, a directed graph is *acyclic* if there are no cycles in the graph. This means that once a vertex has been visited, it can never be visited again. An example of a directed acyclic graph (DAG) is presented in Figure 2.7.

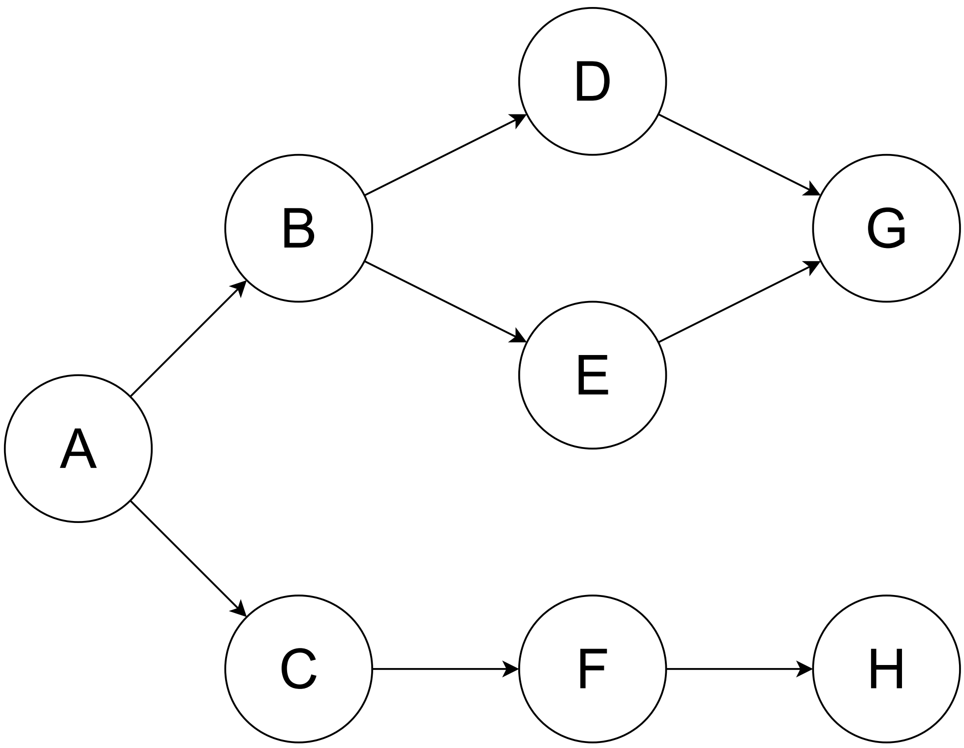

Figure 2.7: Example of a DAG.

DAGs are often used in computer science to represent the flow of execution or communication in a system. For example, in a system with multiple processes, a DAG is often used to represent communication, where the processes are vertices, and the edges represent communication between them through something like TCP or RPC. Machine Learning and AI frameworks, such as Tensorflow [39] and Ray [40], often use DAGs to describe execution flow.

DAGs can also be helpful in modeling publish-subscribe systems, such as SIENA [41]. Each node in the DAG would represent a component in the system which subscribes from its previous nodes and publishes to its next nodes. For example, in Figure 2.7, node A would publish to nodes B and C, and node B would subscribe to node A and publish to nodes D and E.

2.4 Capstone

The contributions of the Capstone thesis this system is based on were to implement the initial system model and explore integration with three data sources; ZXY Sports Tracking, PmSys, and Abel. This section will recap those contributions.

2.4.1 Database

The database is used to store user data persistently. For internal system data, such as login information and team memberships, this is required to have data persist between user sessions. Data from third-party systems is not necessarily required to be stored in Sáhka's database since it can be fetched when a user queries it. However, storing third-party data in the database has some benefits and detriments.

First, storing the data in Sáhka's database increases query performance. This is because the database is physically closer to the back-end, being hosted on the same server, which reduces the time the data has to travel. Furthermore, data from third-party systems require some form of processing before being presented to the users. Doing the processing in the background and storing the already-processed data in the database eliminates this processing time from the users' query time.

The second benefit to consider is availability. Fetching data from third-party systems in real-time requires that they are available. Hence, if a third-party system is unavailable, the part of Sáhka using that data is also unavailable. Conversely, if all fetched data is stored in the database, the data will still be available if the third-party systems are unavailable. In the extreme case of a third-party system becoming permanently unavailable, by becoming discontinued or rescinding access, the data becomes completely lost if it is not stored in the database. Furthermore, many systems rate-limit their APIs, which would reduce Sáhka's availability if data was fetched in real-time and users exceeded the limits.

On the other hand, there are some detriments to storing third-party data in Sáhka's database. The main downside is the increased storage demands. Depending on the type of data, storing it in the database may significantly increase storage demands, consequently impacting the cost of scaling the system. As will be discussed in Section 4.1.5, some data is too costly to store in the database and is instead fetched in real-time. Additionally, there is a consistency issue. Currently, Sáhka only fetches new data from third-party systems, which means that if data is updated in a third-party system, it will not be reflected in Sáhka. However, this is not a significant concern since the data fetched from third-party systems is primarily raw and immutable.

When choosing which database to use, the first consideration was whether to use a relational or non-relational database. Since the schema of the data from third-party systems is unknown, some of the data may not fit a relational model or a strict schema. Hence, the choice was made to use a non-relational, schemaless database. Additionally, having a schemaless database will also simplify the process of adding or removing data sources in the future.

After deciding the type of database to use, the actual choice was somewhat arbitrary. Some of the most popular databases that fit these criteria include MongoDB[7], Redis [42], Cassandra [43], and Dynamo [44]. However, MongoDB was chosen because it was a familiar database with built-in support for concurrency and vertical and horizontal scaling. Additionally, the database can be upgraded to MongoDB Enterprise[8], which supports encryption at rest, which means the data is stored encrypted on disk, for extra security.

2.4.2 Back-end

The back-end of Sáhka has two primary responsibilities: processing requests from the front-end and fetching data from third-party systems. The requests from the front-end are sent to a REST API, where the back-end performs actions based on the request. These actions include responding with data to the front-end or storing new data in the database.

The other responsibility of the back-end is to collect data from third-party systems. This is done in a separate process that constantly runs in the background. The process periodically queries the third-party systems and fetches all new data, if there is any. In a perfect world, the third-party systems would

7. https://www.mongodb.com/
8. https://www.mongodb.com/try/download/enterprise

be able to notify Sáhka when new data is available, eliminating the need for these periodic requests. However, since none of the third-party systems currently integrated has this option, periodic queries are the second-best option. Pseudo-code for this process is presented in Figure 2.8.

```
function CollectXXXData
    data ← FetchData()
    data ← ProcessData(data)
    DatabaseInsert(data)
end function

loop
    CollectZXYData()
    CollectPmSysData()
    Sleep()
end loop
```

Figure 2.8: Data collection algorithm [1]

The process sleeps for a specified amount of time between each round of collecting data. How long the process sleeps determines how long users might have to wait for new data in the third-party systems to be reflected in Sáhka. However, there may be a minimum amount of time necessary to sleep if the third-party systems rate-limit their APIs. Furthermore, if some third-party systems have much lower rate-limits, sleeping a different amount of time for each may be worthwhile.

When choosing a programming language to implement the back-end, two factors were considered: ease of development and performance. Ease of development includes language features, libraries, specifically for MongoDB and REST APIs, and familiarity. Performance was a consideration because the system would be processing a lot of data, and it would reduce query times for the users. With these considerations in mind, C# was chosen because it is a relatively high-level language with libraries for MongoDB and REST APIs while maintaining relatively good performance.

2.4.3 Front-end

A front-end is the interface with which users interact with a system. For Sáhka, the front-end has two functionalities: visualizing data to users and collecting data from users. Visualizing data consists of fetching data from the back-end and then displaying it to the user in, for example, a graph or table. Collecting data involves taking some user inputted data and sending it to the back-end for processing and storage. Consequently, any implementation chosen for the front-end needs to support these features.

With this in mind, there are three primary approaches to implementing a front-end: a web application, a native application, or a hybrid application. Web applications are run in browsers, which means they have the advantage of being available on any platform or device as long as it has a browser, with just one implementation of the application. This is especially beneficial for Sáhka because it lets the coaches use the application on their PC, tablet, or phone.

Contrarily, native applications are explicitly implemented for one platform, such as IOS, Android, or Windows. This comes with the benefit of often having better performance than web applications because they are implemented in faster languages than the JavaScript running in browsers, such as Swift or Kotlin. Additionally, native applications can be optimized for their specific platform since they will only run there. Finally, native applications can provide some functionalities that web applications cannot, such as push notifications and offline mode. The downside to native applications is that one version of the application has to be implemented for each platform it wants to support.

Hybrid applications are the middle ground between web applications and native applications. They are implemented in frameworks that allow them to be run in a browser and as native applications. This gives them the versatility of running on any platform with just one implementation and the improved user experience of native applications with push notifications and offline mode. However, implementing hybrid applications is more work than web applications because of the added functionalities, and they have lower performance than native applications because they are still using JavaScript.

With all these considerations in mind, the front-end of Sáhka was implemented as a web application because it has the lowest implementation overhead while making the applications available on all the necessary devices. Specifically, Sáhka's front-end was implemented in Angular[9] because it was a suitable and familiar framework, which eased the development process. Angular can also be ported to a hybrid framework like Ionic[10] without too much effort in the future.

9. `https://angular.io/`
10. `https://ionicframework.com/`

2.4.4 Data Sources

ZXY Sports Tracking

As stated in Section 3.2, Sáhka must collect positional data from the ZXY system. The ZXY systems, and the servers that run the system, are installed locally at a stadium or pitch. Consequently, each installation of ZXY will need its own connection to Sáhka. Since Sáhka is developed in collaboration with TIL, their ZXY server, running at Alfheim, will be integrated.

The ZXY server has a REST API that will be used to fetch the data. Specifically, two endpoints will be used. The first endpoint provides a list of every training session available in the system. The second endpoint provides an XML file containing the data from one particular session. These endpoints will be used in conjunction with each other to fetch the XML file of each session.

As mentioned in the functional requirement for ZXY, the data should be stored locally in the database. The alternative would be to not store the data in the database and instead fetch it from the ZXY server when needed, for example, when a user accesses one of the ZXY pages on the front-end. However, this would significantly increase the query time since the data would need to be fetched from the ZXY server, processed, then returned to the front-end. Instead, if the data is fetched from the ZXY server, processed, and stored in the database, it would only have to be fetched from the database and returned to the front-end. Additionally, if the data is not stored in the database and the ZXY server is unavailable, then Sáhka would be unable to provide ZXY data to the users. However, if the data is stored in the database, it would still be available even if the ZXY server is not. Hence, the data will be fetched from the ZXY server, processed, and stored in the database to reduce query time and increase availability.

The processing that is performed before storing the data consists of separating the XML document by the players so that each player has one collection of data per session. This is done because the front-end shows data for one player at a time, so it would be inefficient to store the entire session in one document and parse it during a query. The format of the documents stored in the database is shown in Figure 2.9.

```
{
    "_id": ObjectId("6177d973354af39d9f2b0342"),
    "session_id": "76456",
    "player_id": "2364",
    "start_time": ISODate("2021-12-06T17:02:30.104Z"),
```

```
    "stop_time": ISODate("2021-12-06T18:51:57.104Z"),
    "teams": [ "6546", "3246" ],
    "periods": {
        "period_id": "1",
        "start_time": ISODate("2021-12-06T17:02:30.104Z"),
        "stop_time": ISODate("2021-12-06T17:47:53.104Z"),
        "period_name": "First half",
        "total_distance": Decimal128("5739"),
        "walk_distance": Decimal128("1656"),
        "jogg_distance": Decimal128("2542"),
        "run_distance": Decimal128("1189"),
        "accelerations": [
            {
                "start_time":
↪   ISODate("2021-10-24T17:02:43.416Z"),
                "stop_time":
↪   ISODate("2021-10-24T17:02:44.216Z"),
                "distance": Decimal128("2.3689999580383"),
                "max_acceleration":
↪   Decimal128("2.566999912262")
            },
            ...
        ],
        "decelerations": [
            {
                "start_time":
↪   ISODate("2021-10-24T17:07:06.465Z"),
                "stop_time":
↪   ISODate("2021-10-24T17:07:07.865Z"),
                "distance": Decimal128("5.0939998626709"),
                "max_acceleration":
↪   Decimal128("3.6129999160767")
            },
            ...
        ],
        "runs": [
            {
                "start_time_run":
↪   ISODate("2021-10-24T17:03:29.266Z"),
                "start_time_fastrun":
↪   ISODate("2021-10-24T17:03:30.166Z"),
                "max_speed_time":
↪   ISODate("2021-10-24T17:03:30.866Z"),
```

```
                "stop_time_fastrun":
↪   ISODate("2021-10-24T17:03:32.166Z"),
                "stop_time_run":
↪   ISODate("2021-10-24T17:03:34.166Z"),
                "max_speed": Decimal128("6.289999961853"),
                "distance": Decimal128("25.930000305176"),
                "offensive_run": false
            },
        ],
        "minutes": [
            {
                "minute": 1,
                "total_distance":
↪   Decimal128("124.58000183105"),
                "walk_distance":
↪   Decimal128("34.349998474121"),
                "jogg_distance":
↪   Decimal128("65.669998168945"),
                "run_distance": Decimal128("19.739999771118"),
                "fastrun_distance":
↪   Decimal128("4.8200001716614"),
                "sprint_distance": Decimal128("0")
            },
        ]
    }
}
```


Figure 2.9: Example of a player's ZXY data from a football match.

The front-end contains multiple pages for visualizing the ZXY data in different ways. The information coaches want on their players from ZXY is primarily; the distance they run, divided into the total distance, high intensity, and sprint distance; how fast they run; and their accelerations, both distance and speed. The data does include more details, such as jogging and walking distance; however, feedback from TIL's coaches indicates this level of detail is unnecessary and unwanted. In particular, feedback from Bjørn Vidar Stenersen, a previous physical coach at TIL and currently for the Norwegian national football team, has provided feedback on how coaches want the ZXY data to be visualized.

Figure 2.10: A coach's overview of a team's ZXY data over a period of one week.

The first page, shown in Figure 2.10, is exclusive to coaches because it shows ZXY data for an entire team. On this page, the coach selects a period, for example, the last seven days, and the graphs show the sum of each player's ZXY data for that period.

Hence, the bar highlighted by the first circle shows the distance traveled by one player for the specified period. The bar is separated into three colors, blue, orange, and green, which denotes the total distance traveled, distance traveled during high intensity runs, and distance traveled during sprints, respectively.

The second highlighted bar shows the fastest the player has run during the specified period. Additionally, each bar in this plot has a red dotted line above it, which shows what 90% of the player's top speed ever is. One of TIL's coaches explicitly requested this line because it is helpful to see how hard the players are exerting themselves.

The third highlighted bar shows how many accelerations and decelerations the player has had during the specified period. The blue part of the bar represents accelerations, whereas the orange part represents decelerations.

The final highlighted bar shows the distance the player has accelerated and decelerated during the specified period. The blue and orange parts of the bar separate the data into accelerations and decelerations, respectively.

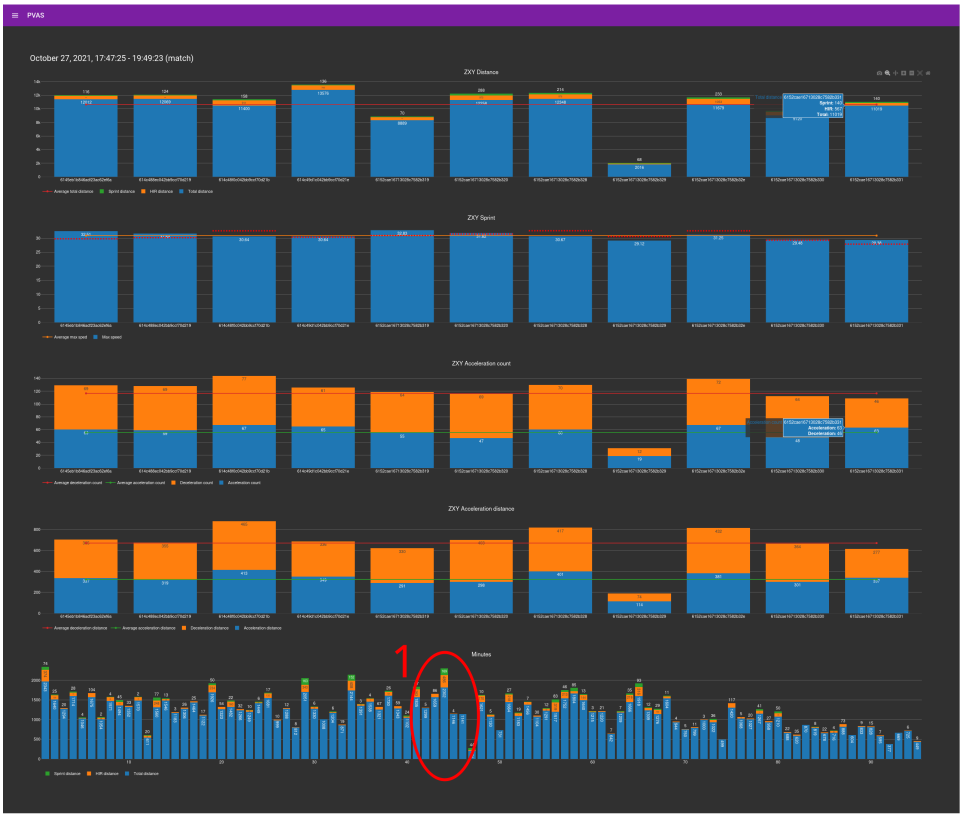

Figure 2.11: An overview of a team's ZXY data during one session.

The second page, shown in Figure 2.11 and also exclusive to coaches, displays a team's ZXY data for one specific session. The first four graphs show the same data as the previous page; distance traveled, max speed, and acceleration and deceleration. Additionally, these graphs contain horizontal lines showing the team's average value. For example, the line in the first graph shows the team's average total distance traveled.

The new addition to this page is the last graph, which shows the team's distance traveled minute by minute. For example, the bar in the middle of the highlight shows the distance traveled by the team during the 44th minute of the session. These bars are also separated into blue, orange, and green, representing total distance, distance during high-intensity runs, and sprint distance, respectively.

The last two pages are available to coaches and players and show ZXY data for just one player. Coaches can then change which player to view data from, whereas players can only view their own data.

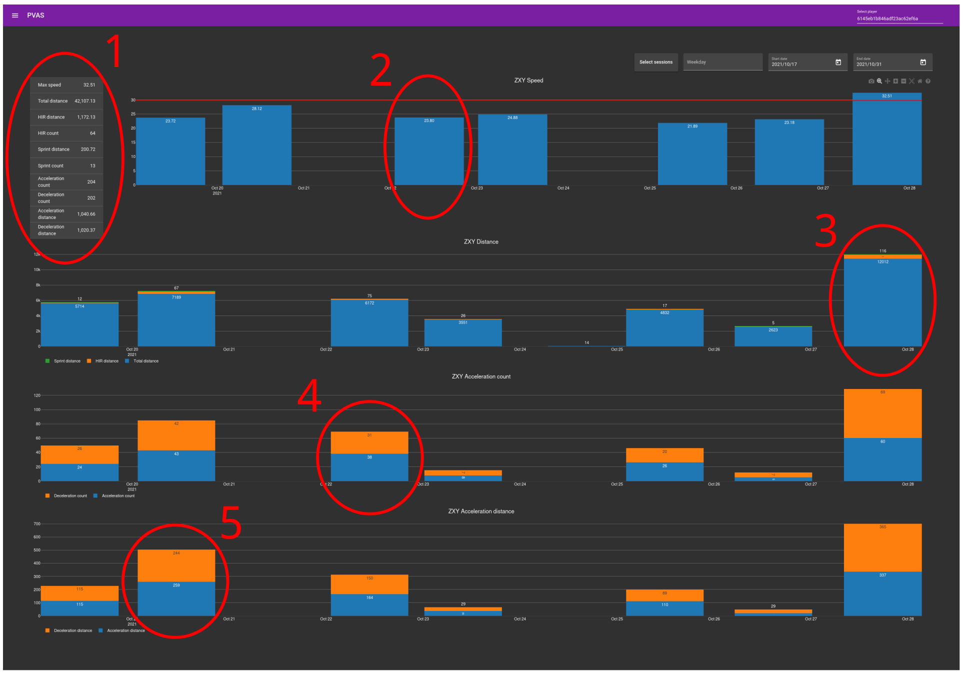

Figure 2.12: An overview of a player's ZXY data over a period of one month.

The page in Figure 2.12 gives an overview of the player's sessions for a specified amount of time, for example, one month. The table in circle one shows the sum of different metrics during the specified period, including max speed, total distance, high-intensity distance, sprint distance, number of sprints, number of accelerations and decelerations, and distance accelerated and decelerated.

The first graph shows the player's max speed for each session in the specified period. Hence, the bar in circle two shows the player's max speed during one of the sessions. This graph also contains the red horizontal line representing 90% of the player's max speed ever run.

The second graph shows the player's distance traveled during each session of the specified period. The bar in circle three shows the distance traveled during one session separated into blue, orange, and green, representing total distance, high-intensity distance, and sprint distance, respectively.

The third graph shows the player's number of accelerations and decelerations for each session in the specified period. The bar in circle four shows the number of accelerations in blue and decelerations in orange during one session.

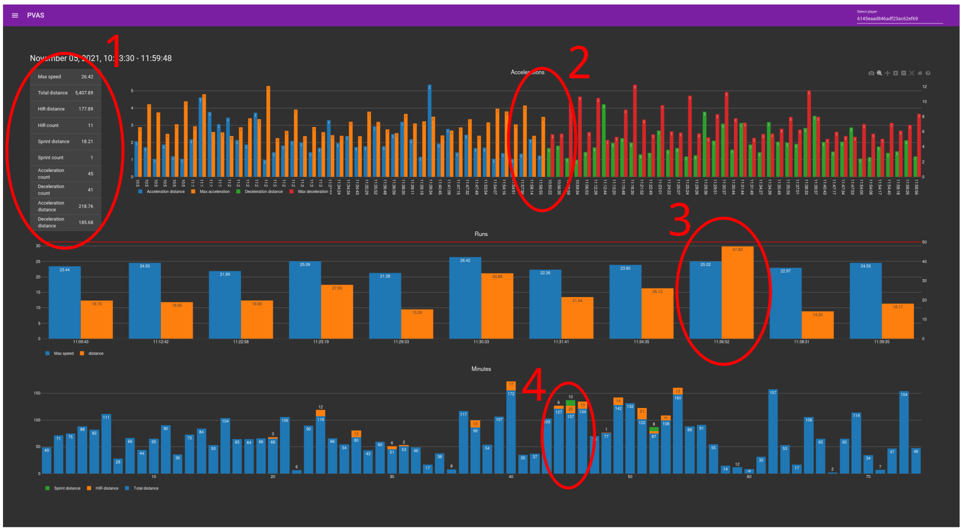

Figure 2.13: An overview of a player's ZXY data during one session.

The last page, shown in Figure 2.13, shows a more detailed view of one player's ZXY data during one session. The table in circle one summarizes the player's metrics during the session, showing max speed, total distance, high-intensity distance, number of high-intensity runs, sprint distance, number of sprints, number of accelerations and decelerations, and distance accelerated and decelerated.

The first graph shows each of the player's accelerations and decelerations during the session. Each blue and orange bar is a pair that represents one acceleration event, where the blue bar represents the distance accelerated and the orange bar represents the max acceleration rate during the event. Likewise, the red and green bars represent deceleration events, green being the distance decelerated and red being the max deceleration rate.

The second graph shows each of the player's runs during the session. A movement classifies as a run if it has a top speed >= 19.8 km/h and lasts more than one second. Circle three shows a blue and orange bar pair, representing one run event. The blue bar shows the top speed during the run, and the orange bar shows the distance traveled during the run. This graph also has the red horizontal line showing 90% of the player's max speed ever run.

The final graph shows the player's distance traveled minute by minute. Hence, the bar in the middle of circle four shows the player's distance traveled during the 45th minute. These bars are separated into blu, orange, and green, representing total distance, high-intensity distance, and sprint distance, respectively.

PmSys

As stated in Section 3.2, Sáhka must collect wellness data from the PmSys. Since PmSys is a product of ForzaSys, the data is fetched from their servers using their REST API. Specifically, one endpoint will be used, which provides all wellness data for one player.

The same options for storing ZXY data in the previous section also apply to storing PmSys data. The data can either be fetched from the PmSys servers on-demand or stored locally in Sáhka's database. Consequently, the same reasoning for storing ZXY data in the database also applies here; it reduces query time and increases availability at the cost of additional storage requirements.

When fetching the PmSys data, one request has to be sent for each player. Additionally, by specifying the date of the latest data already stored in Sáhka, only new data is fetched from PmSys. The data received includes some unnecessary fields, which can be filtered out before being stored in the database to reduce disk usage. Figure 2.14 shows an example of a JSON response from PmSys, and Figure 2.15 shows the data stored in the database.


```
{
    "header": {
        "id": "7aa9f6eb-953b-4cb3-a5fe-35d77a1863e4",
        "creation_date_time": "2018-05-23T07:10:33.759Z",
        "schema_id": {
            "namespace": "corporesano",
            "name": "wellness",
            "version": "1.0.0"
        },
        "acquisition_provenance": {
            "source_name": "",
            "source_creation_date_time": "",
            "modality": "self-reported"
        },
        "user_id": "xxxxxxxxxxxxxxxxxxxx",
        "additional_properties": "{}"
    },
    "body": {
        "effective_time_frame": {
```

```
            "date_time": "2018-05-23T07:10:33.759Z"
        },
        "fatigue": 3,
        "mood": 4,
        "readiness": 5,
        "sleep": {
            "duration": {
                "unit": "h",
                "value": 8
            },
            "quality": 4
        },
        "soreness": 2,
        "soreness_area": [
            32696007
        ],
        "stress": 4
    }
}
```


Figure 2.14: Example of PmSys response body [1].


```
{
    "_id": ObjectId("619627f93501b3cc6b3558e8"),
    "user_id": "6451793a930d3433cb2a1b25",
    "creation_date": ISODate("2018-05-23T07:10:33.759Z"),
    "date": ISODate("2018-05-23T07:10:33.759Z"),
    "fatigue": 3,
    "mood": 4,
    "readiness": 5,
    "soreness": 2,
    "stress": 4,
    "soreness_area": [
        32696007
    ],
    "sleep": {
        "duration": {
            "unit": "h",
            "value": 8
        },
        "quality": 4
    }
}
```

Figure 2.15: PmSys data after being processed [1].

The front-end for visualizing PmSys data is relatively straightforward, with only one page, shown in Figure 2.16. The page shows the PmSys data for one player for the specified period, for example, the last seven days. Coaches can select which of their player's data to view, whereas players can only view their own data.

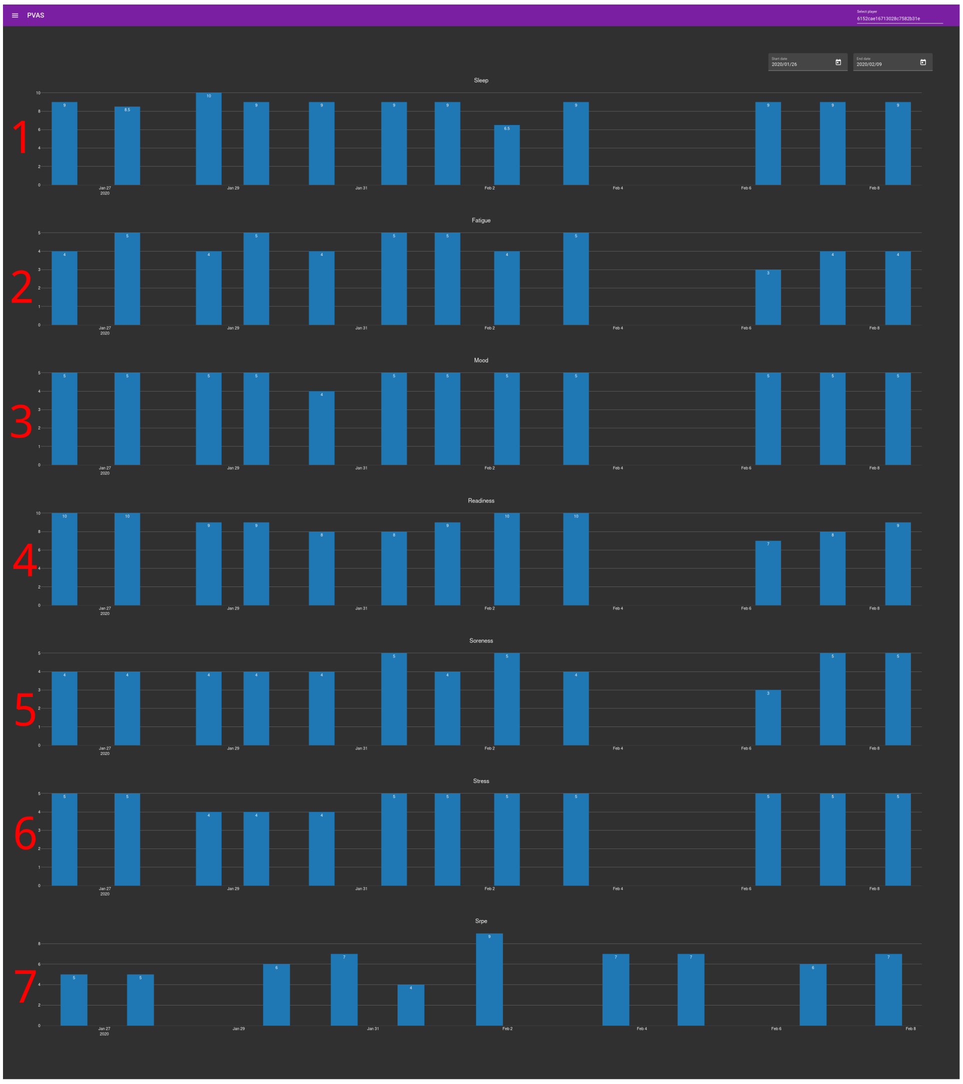

Figure 2.16: An overview of a player's PmSys data for a given period.

Each graph displays one of the wellness metrics the player reports in the PmSys app; the time of the report along the x-axis and the value along the y-axis. The first graph shows the player's sleep metrics. Each bar shows how long the player slept, and hovering over the bar will also reveal the sleep quality in a tooltip. After that, the graphs show, in order, the player's fatigue, mood, readiness, soreness, stress, and SRPE.

Abel

As defined in Section 3.2, Sáhka must be able to collect physical training data equivalent to the data Abel collects; the system TIL has been using for logging and monitoring strength training sessions. The best solution would be to collect this data from Abel in the same way as the two previous data sources. However, Abel does not have any public methods available for Sáhka to collect this data. Hence, this data must come from somewhere else. The two options available were for TIL to switch to another system with a public API, or to reimplement this functionality in Sáhka. Both of these options come with some benefits and downsides.

The first benefit of reimplementing Abel is reducing the number of third-party dependencies. This is good because third-party dependencies may increase the cost of maintaining the system in the future if they change, become unavailable, or become discontinued. However, reimplementing the features of a third-party system may add more complexity to developing the system. Abel is relatively simple to reimplement, with just a few pages for collecting user data; however, other systems may be too much to be worth it. For example, ZXY requires sensor belts and radio receivers, which is too far outside the scope of what Sáhka is.

Another consideration is the added control and customizability of reimplementation. This allows Sáhka to customize what data to collect and how to collect it to fit the users' needs best. Comparatively, third-party systems are often more generalized to fit a larger audience and might not fit the needs of Sáhka perfectly.

Since TIL would have to move away from Abel regardless, and the cost of reimplementing it in Sáhka was comparable to integrating a different system, it was decided to reimplement the required features in Sáhka.

The primary feature reimplemented from Abel was for players to log physical training sessions. Each session includes information about which exercises were performed, the number of repetitions, and the weight lifted. The data is entered into a form in Sáhka's front-end by either the player or their coach, as shown in Figure 2.17.

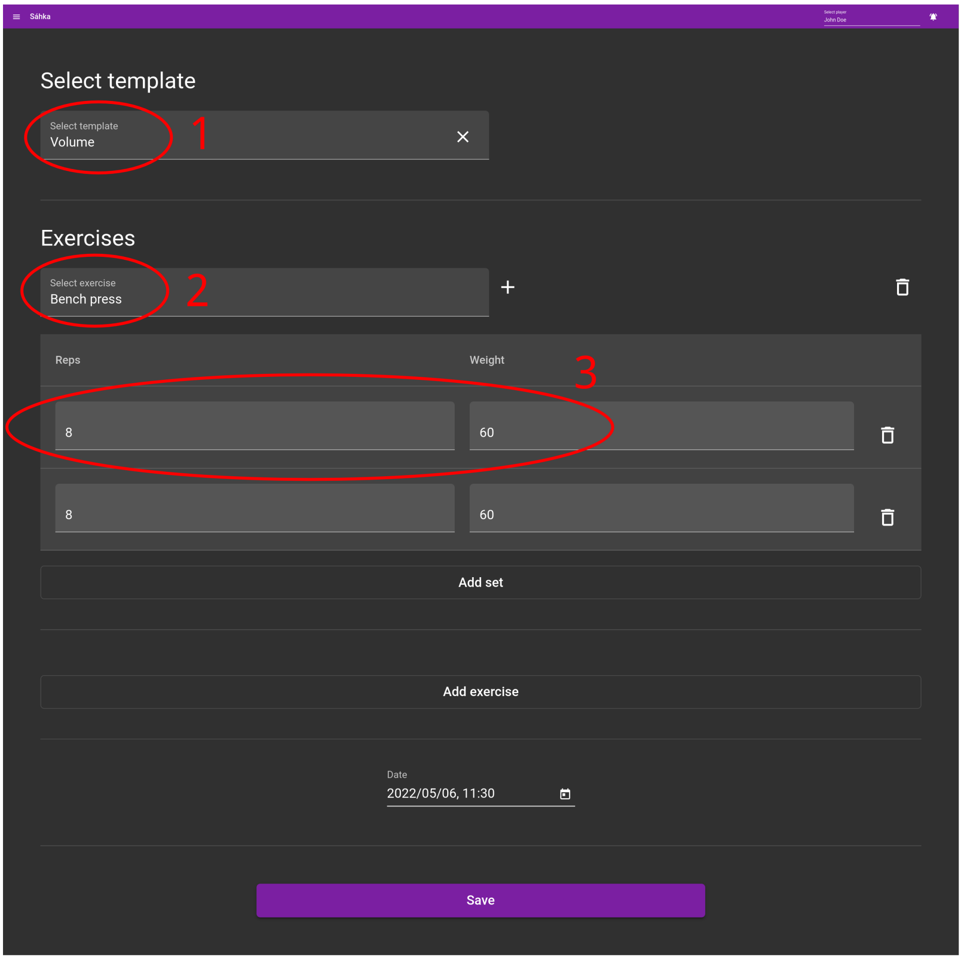

Figure 2.17: The page where users log training sessions.

In the input highlighted by circle one, the player can select a template training session to follow. The template, which a coach has created, fills out the exercises and the number of sets and reps the player should perform.

If the player does not select a template or wants to change or add an exercise, the input in circle two allows them to select which exercise they performed. Subsequently, the player inputs the number of reps performed and the weight lifted for each set of the exercise, as shown in circle three.

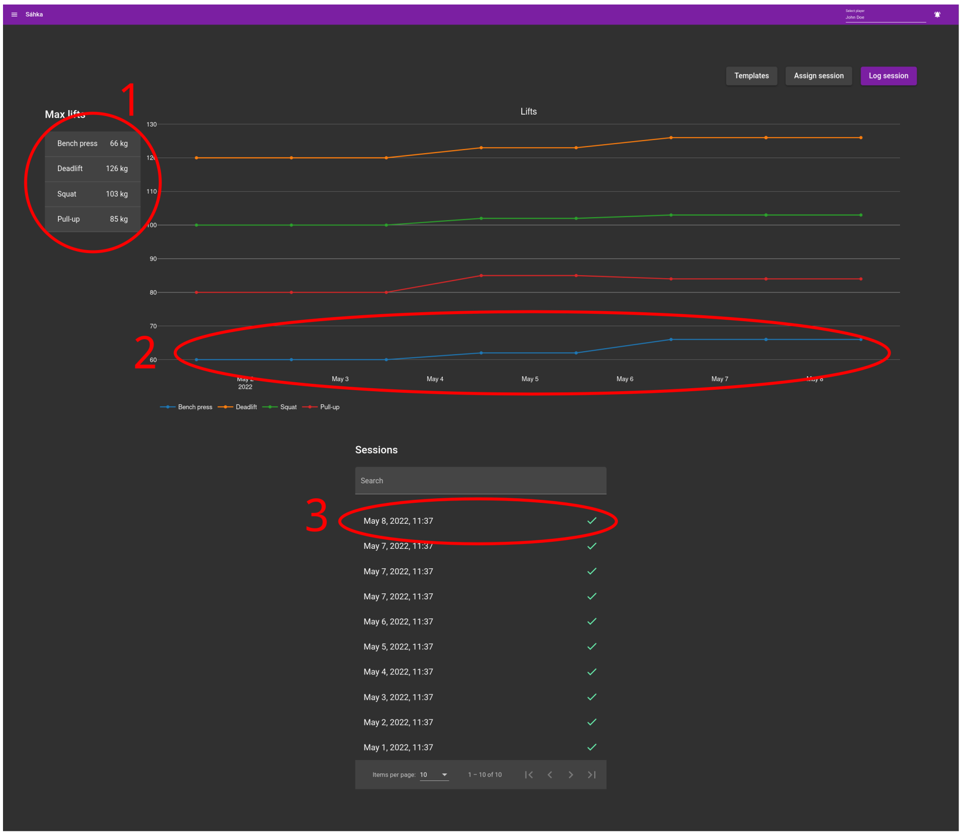

Figure 2.18: An overview of a player's training data.

Figure 2.18 shows the page that visualizes a player's training data. The table in circle one shows a list of every exercise the player has performed. Next to each exercise is the heaviest they have lifted in that exercise.

The graph shows every exercise the player performs, where the x-axis is the time the exercise was performed, and the y-axis is the heaviest weight lifted during the exercise. Hence, the blue line inside circle two shows what the player lifted in bench press each session.

Finally, at the bottom of the page is a list of every training session the player has performed. Clicking one of the items, for example, the one inside circle three, will open the page from Figure 2.17, where the user can view, modify, or delete the session.

2.5 Summary

This chapter has given some background into some of the methods and purposes of monitoring the training load of athletes. Additionally, the data sources explored in this thesis have been presented.

3

Requirement Specification

This chapter outlines the requirements of the system based on the problem definition from Section 1.1. A model for the system will be proposed, and functional and non-functional requirements will be specified.

3.1 System Model

Based on the problem definition stated in Section 1.1, the system needs to have four core functionalities:

1. *Collect* data on players
2. *Store* data persistently
3. *Present* data to authorized users
4. *Algorithmically analyze* data

Presenting data to authorized users requires that the system has a front-end where users can view and interact with said data. The front-end should also be able to collect data from users when third-party systems are not available or satisfactory for a specific type of data. Subsequently, the front-end will require a back-end to send and receive the data from. This back-end will also be responsible for collecting player data from third-party systems. Finally, all data collected will need to be stored persistently.

Based on these requirements, a 4-tiered system model is proposed, with the four tiers being a front-end, back-end, database, and the external third-party systems. The system model is illustrated in Figure 3.1.

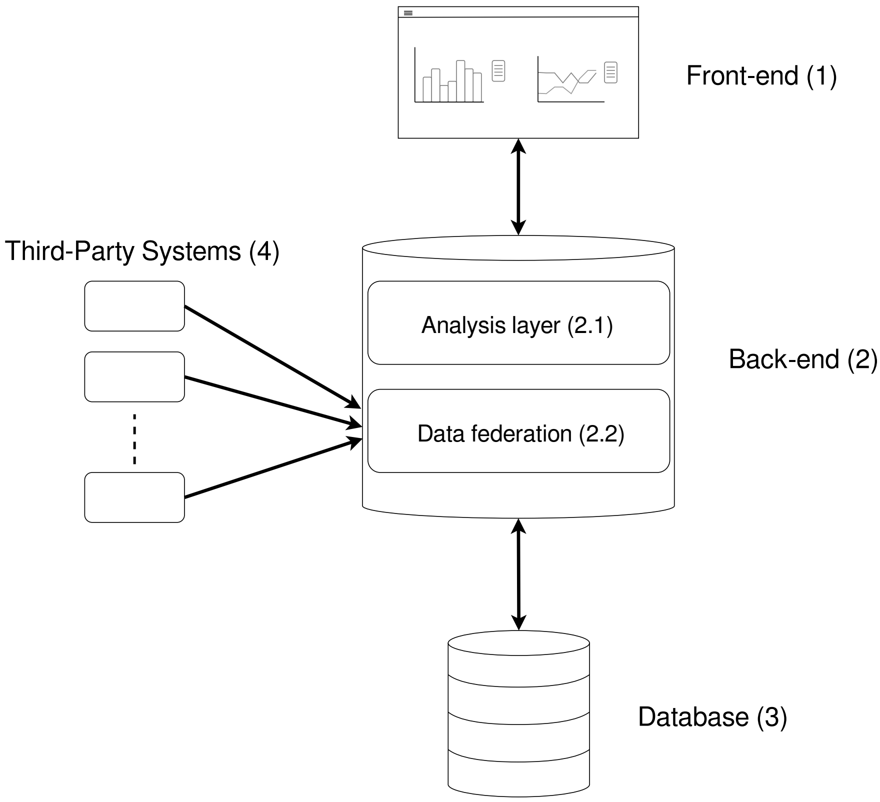

Figure 3.1: Proposed 4-tiered system model.

The front-end (1) is the interface users use to interact with the system, and it has two main functionalities: collecting data from users and presenting data to authorized users. The front-end sends the data collected to the back-end (2) and fetches the data to present from the back-end. The back-end is divided into two layers. The analysis layer (2.1) runs all the analysis algorithms that provide notifications to the coaches. The data federation layer (2.2) is responsible for processing all requests from the front-end and collecting data from the third-party systems (4). Finally, the back-end stores all data in the database (3) for persistence.

3.2 Functional Requirements

Functional requirements are statements of services a system should provide, how the system should react to particular inputs, how the system should behave in particular situations, and what the system should not do [45]. The following functional requirements are defined for Sáhka:

- **ZXY:** The system must be able to collect positional data from the ZXY Sports Tracking system. The data should be processed and stored locally to reduce query time and increase availability. Finally, the data should be visualized to the users as graphs and tables, representing distance traveled, speed, and acceleration.

- **PmSys:** The system must be able to collect wellness data from PmSys. The data should be processed and stored locally to reduce query time and increase availability. Finally, the data should be visualized to the users as graphs representing the various wellness metrics, such as sleep and mood.

- **Abel:** The system must be able to collect, store, and visualize physical training data directly from the users, similar to the Abel system. The data should be visualized to the users as graphs and tables, representing the various training metrics, such as the number of sets, reps, and weight lifted.

- **DXA:** The system must be able to collect, store, and visualize DXA data. The data should be visualized to the users as graphs representing the various DXA metrics, such as weight and body fat percentage.

- **Performance tests:** The system must be able to collect, store, and visualize different types of performance tests, including CMJ, SJ, BJ, bench press, pull-up, 15-meter sprint, and $30\text{-}15_{\text{IFT}}$. The data should be visualized to the users as graphs representing the results of these tests over time.

- **Nutrition:** The system must be able to collect, store, and visualize nutritional data from the users in the form of images. The images should be visualized to the users in reverse chronological order.

- **Video clips:** The system must be able to collect video clips from various sources, including directly from the users and Eliteserien Highlights. The videos should not be stored locally, when possible, to reduce disk and bandwidth usage. Finally, the videos should be presented to the users in reverse chronological order, and the users must be able to play them.

- **Objectives:** The system must be able to collect, store, and visualize data of players' objectives for given periods. Additionally, coaches must be able to write private notes for each objective.

- **Journal:** The system must be able to keep a journal for each user where they can write personal notes. Additionally, the journal must be private for each user.

- **Analysis:** The system must be able to run algorithmic analyses on the data collected to detect specific events, such as a player sleeping too little. The analyses must be run proactively whenever new relevant data is available. The system must then notify the coaches of that player that the event has occurred. Additionally, coaches must be able to customize the analyses by choosing and combining the events they want in a DAG format.

- **Visualization:** The system shall be developed in close collaboration with TIL, specifically their coaches. Throughout the development, the coaches will provide feedback on how they want the data to be visualized. The system must continuously adapt to reflect what the coaches want. This includes all the different types of coaches, such as head coach, physical coach, and mental coach.

3.3 Non-Functional Requirements

Non-functional requirements are constraints on the services or functions offered by a system, including timing constraints, constraints on the development process, and constraints imposed by standards [45]. This section outlines the non-functional requirements defined for Sáhka.

3.3.1 Privacy

The system will be collecting a lot of personal data from its users, some of which may be highly sensitive. For example, DXA data contains information about a user's body composition, such as weight, body fat percentage, and muscle mass. This introduces a critical requirement that the system ensures its users' privacy through access control, utilizing authentication and authorization, to only allow authorized users to view a player's data.

Additionally, the system is subject to the General Data Protection Regulations [46] (GDPR) and must comply with these regulations. GDPR compliance will be discussed further in Section 6.4.

3.3.2 Security

The system should take security measures to ensure that users can trust it with their privacy. This includes authentication, access control, and protection against malicious attacks. How Sáhka deals with security will be discussed further in Section 6.2.

3.3.3 Scalability

The performance demands of the system are dependent on the number of users and the amount of data collected, both of which are expected to increase over time. Hence, the system should be scalable to accommodate these demands. As the number of users increases, the number of concurrent requests will increase proportionally. Consequently, the system should scale up its processing power to maintain low response times. An increase in users, the number of data sources, and time will also increase the storage demands on the system, and its storage capacity should be scaled accordingly.

3.3.4 Availability

Availability requires that the system, and all its features, are available for the users when they want to access their data. However, since the system depends on third-party systems for some of its data, their availability is outside of its control. Therefore, data fetched from third-party systems should be stored locally to remain available even if the third-party system is unavailable. Additionally, locally stored data must be backed up to ensure no data loss in the event of failures.

3.3.5 Maintainability

The system developed is being used in production and will continue to be so. Hence, the system should be easy to maintain for any developer. This is especially important for the integrations with third-party systems since they may change their data format, API, or become unavailable in the future. Additionally, the system should be designed such that adding new data sources is a seamless process.

3.3.6 Usability

The primary user group of this system is coaches and their athletes. Currently, the system is specialized towards football; however, the system can be used by any related sport since many of the data sources are sport agnostic. Hence, the system should be designed with the target group in mind. The user interface should be easy and intuitive to use, while presenting as much relevant information as possible. Feedback from TIL during development will be highly relevant to this requirement.

3.4 Other Issues

This thesis was developed in a five-month period, limited by the deadline imposed by UiT. Consequently, some limitations and simplifications were made when developing the system to save time. Additionally, although the system developed is a prototype, TIL's coaches and players had to use the system during development to populate data and provide feedback. Hence, the system needed to be operational during development. This involved setting up and maintaining a server to host the database, back-end, and front-end.

3.5 Summary

This chapter has proposed a 3-tiered model for the system with a front-end, back-end, and database. Additionally, requirements for the system's performance and behavior have been specified.

4

Design & Implementation

This chapter will describe the design and implementation of the system based on the problem definition formalized in Section 1.1 and the requirements specified in Section 3.

4.1 Data Sources

4.1.1 ZXY

As mentioned in Section 1, TIL's coaches have been using Sáhka during its development and provided feedback on what information they want and how they want it presented. One crucial piece of feedback from Sigurd Pedersen, one of TIL's physical coaches, and Gaute Uglestad Helstrup, the head coach at TIL, was that they wanted more ways to compare and filter ZXY data. Specifically, they wanted to filter and compare players based on their playing position, view a team's data for each day of a given period, or view a team's performance for each type of training session, such as endurance, speed, or activation, for a given period.

To implement filtering and comparison based on playing position, the players must first have their position associated with their account. This can be entered by a coach or the players themselves.

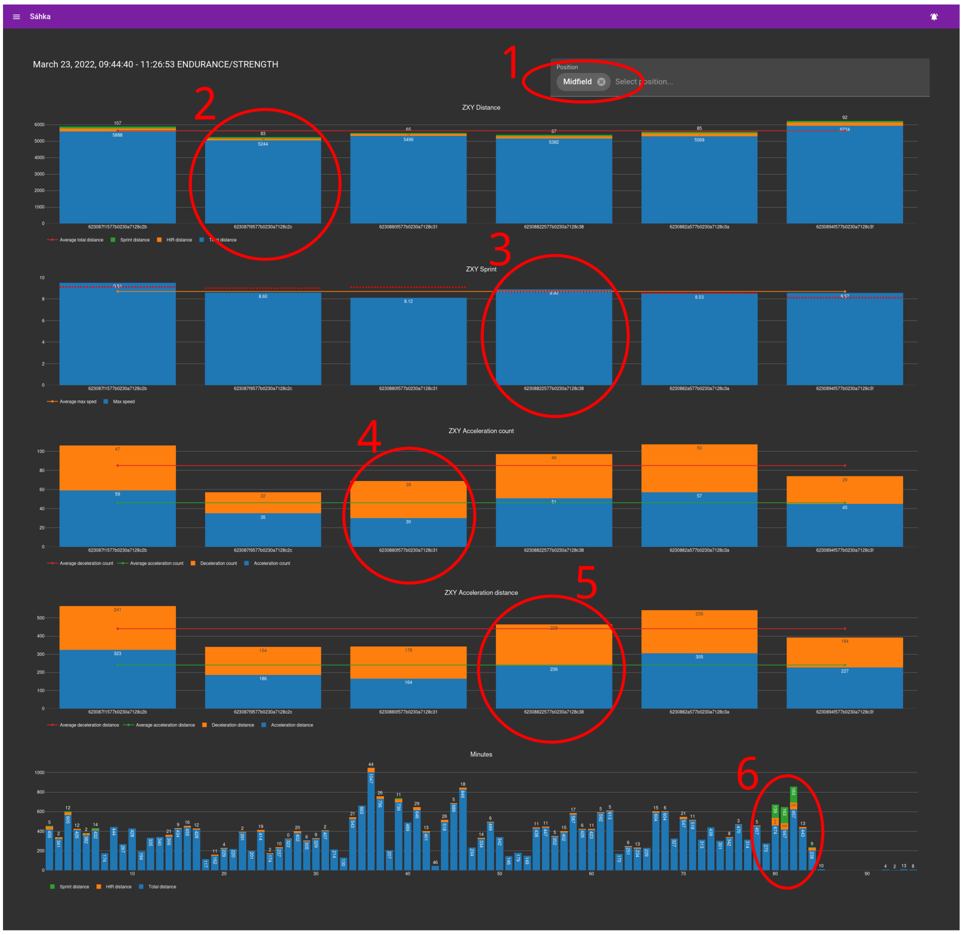

Figure 4.1: The page showing ZXY data for players in the Wingback position.

Figure 4.1 shows the same page as Figure 2.11 after the new modifications. Circle one highlights an input where the coach can select a playing position, either defender, wingback, midfield, or striker. After selecting a position, only the players in that position will be shown in the graphs. The graphs still show the same information as before; the first graph shows distance traveled, divided into the total distance, high-intensity distance, and sprint distance; the second graph shows max speed, and the red dotted line that shows 90% of the player's max speed ever run; the third graph shows the number of accelerations and decelerations; the fourth graph shows the distance accelerated and decelerated; and the final graph shows the distance traveled by every player in the selected position minute by minute, divided into total, high-intensity, and sprint distance.

In addition to filtering based on playing position, the coaches wanted to compare the different positions against each other. This is done by selecting multiple positions in the highlighted input, as shown in Figure 4.2.

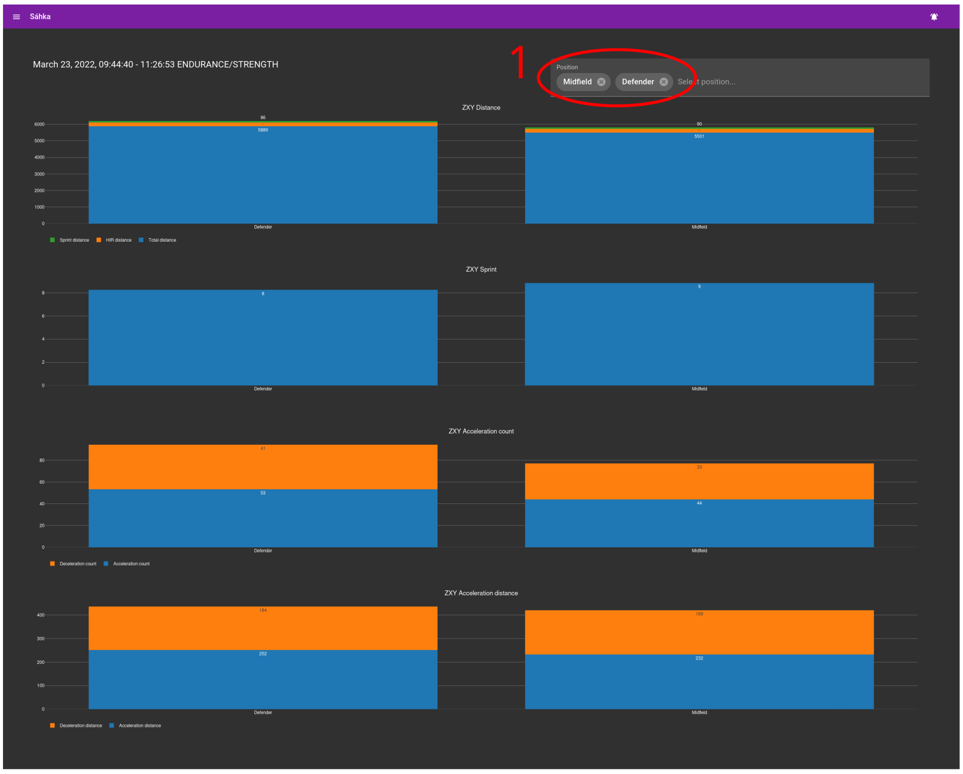

Figure 4.2: The page comparing ZXY data between players in the Wingback position against players in the Striker position.

Figure 4.2 shows the page when the defender and midfield positions are selected. In the first graph, the left bar shows the average total distance traveled by the defenders, and the right bar shows the average for the midfielders. The bars are divided into blue, orange, and green, corresponding with total distance, high-intensity distance, and sprint distance, respectively.

The second graph shows the average max speed of the players in each position; the left bar shows the defenders' speed, and the right shows the midfielders'.

In the third graph, the left bar shows the average number of accelerations and decelerations for the defenders and the right bar for the midfielders. The bars are divided into blue and orange, representing accelerations and decelerations, respectively. Likewise, the final graph shows each position's average distance accelerated and decelerated.

The second feature the coaches wanted, viewing a team's ZXY data for each day of a given period, was added to the page from Figure 2.10. The coaches can switch back and forth between the old version of the page and the new.

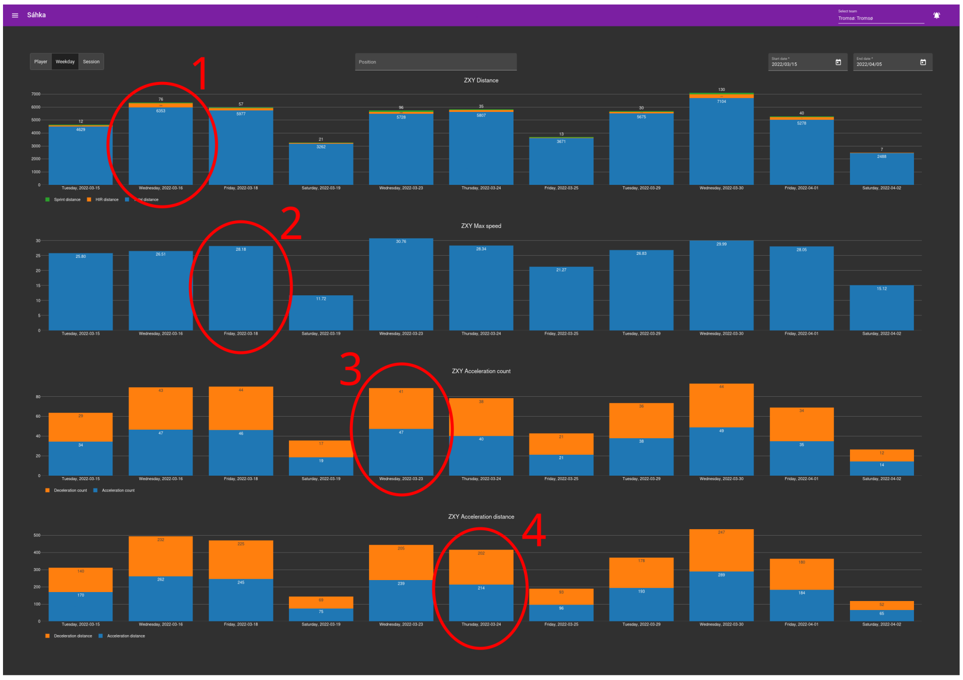

Figure 4.3: The page showing a team's average ZXY data per day.

Figure 4.3 shows the new version of the page. The graphs now show the average metrics of the team for each day of the specified period. Hence the bar in circle one shows the team's average distance traveled on Wednesday the 16th of March 2022. The bar in circle two shows the team's average max speed on the subsequent Friday. The bar in circle three shows the team's average number of accelerations and decelerations the following Wednesday. Finally, the bar in circle four shows the team's average distance accelerated and decelerated the following Thursday.

The final addition, viewing a team's ZXY data based on the type of training session, required an additional data source to be added to Sáhka because the ZXY data itself does not contain any information about the type of session it represents. Ideally, the coaches could add this information as metadata to each ZXY session; however, this was not possible in the current ZXY system. Hence, this data would need to be sourced from somewhere else.

The initial proposal was to have the coaches manually tag each session after they had been fetched into Sáhka. However, after further discussion, a solution with more benefits was discovered. One of TIL's coaches currently has a calendar of the team's schedule, including the type of training sessions for each day, in an excel sheet. Hence, implementing a calendar would let the coaches use Sáhka for managing their schedules instead of an excel sheet, and it would provide the information needed to match ZXY data with the type of session it was.

The calendar implemented consists of one page that shows a team's schedule in one-week intervals. Each team has a unique calendar to which only coaches can view and add events.

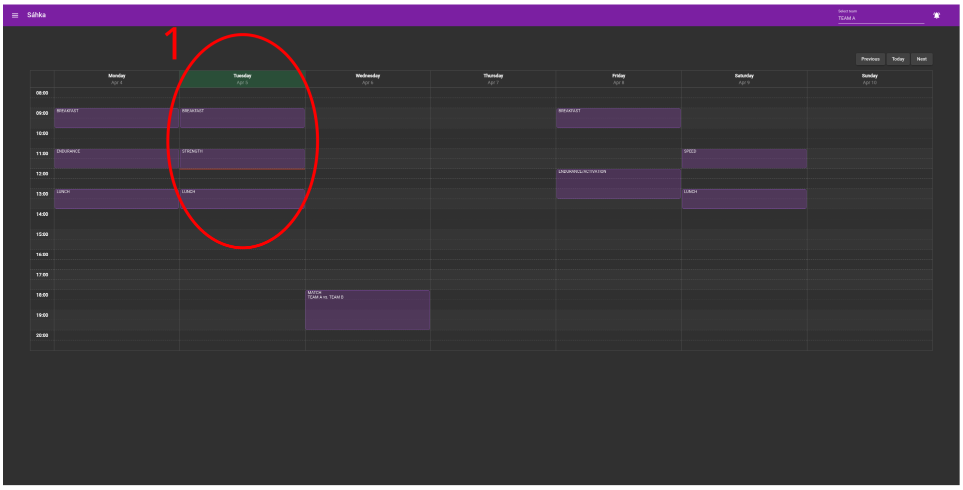

Figure 4.4: The page showing a team's calendar.

Figure 4.4 shows a team's calendar during week fourteen of 2022. Each purple square represents an event and shows when it starts and ends. Hence, the three purple squares in the red circle show that on Tuesday of that week, the team will have breakfast at 09:00, strength training at 11:00, and lunch at 13:00.

With the information available in the calendar, it is possible to determine what type of training session ZXY data belongs to, such as endurance, strength, or speed. The page from Figure 4.3 was further extended to show the average values of a team for each type of training session in a given period.

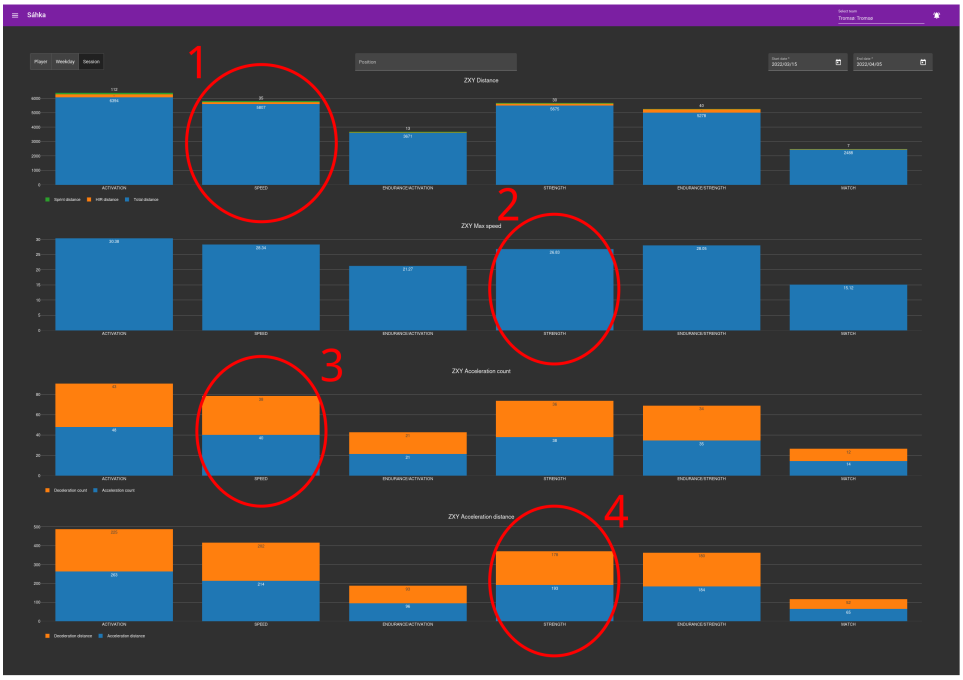

Figure 4.5: The page showing a team's average ZXY data per type of session.

In Figure 4.5, the graphs show the team's average ZXY metrics for each type of training session of the specified period. Hence, the bar in circle one shows the team's average distance traveled every speed session of the specified period. The bar in circle two shows the team's average max speed during all strength sessions of the specified period. The bar in circle three shows the team's average number of accelerations and decelerations every speed session. Finally, the bar in circle four shows the team's average distance accelerated and decelerated during every strength session.

4.1.2 DXA

As stated in Section 3.2, Sáhka must be able to collect, store, and visualize DXA data. The DXA scanner used by TIL is located at Alfheim and run by a computer there. Due to the highly sensitive nature of the DXA data, this computer is password protected and not connected to any networks, internal or external. This significantly reduces the risk of unauthorized parties gaining access to the DXA data by requiring physical access to the machine. However, keeping the computer offline also means that Sáhka's back-end cannot fetch the DXA data automatically.

There were two possible solutions at this point. Either accept the extra work and risk of connecting the DXA computer to the internet or have the coaches manually enter the data into Sáhka. Given the infrequency of the DXA scans and the relatively low effort of manually inputting this data, TIL wanted the second option implemented. However, it is still possible to implement the first option in the future if anyone has their DXA data online. This would allow teams to choose which method suits them best.

The DXA data is collected through a form on Sáhka's front-end, where coaches enter the results after a DXA scan (Figure 4.6). The data is visualized through graphs showing the player's DXA results over time (Figure 4.7).

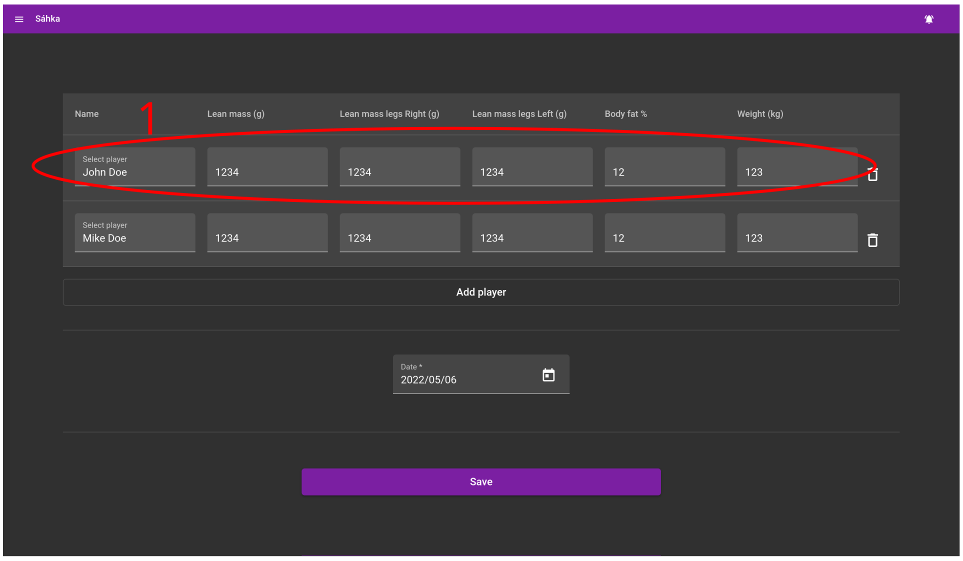

Figure 4.6: Form for inputting DXA data.

Figure 4.6 shows the form coaches use to enter DXA data. The red circle highlights that the coach has selected a player named John Doe and entered the results from his scan, including lean mass, lean mass in the right leg, lean mass in the left leg, body fat percentage, and weight.

Another page visualizes the DXA data through graphs showing the player's results over time.

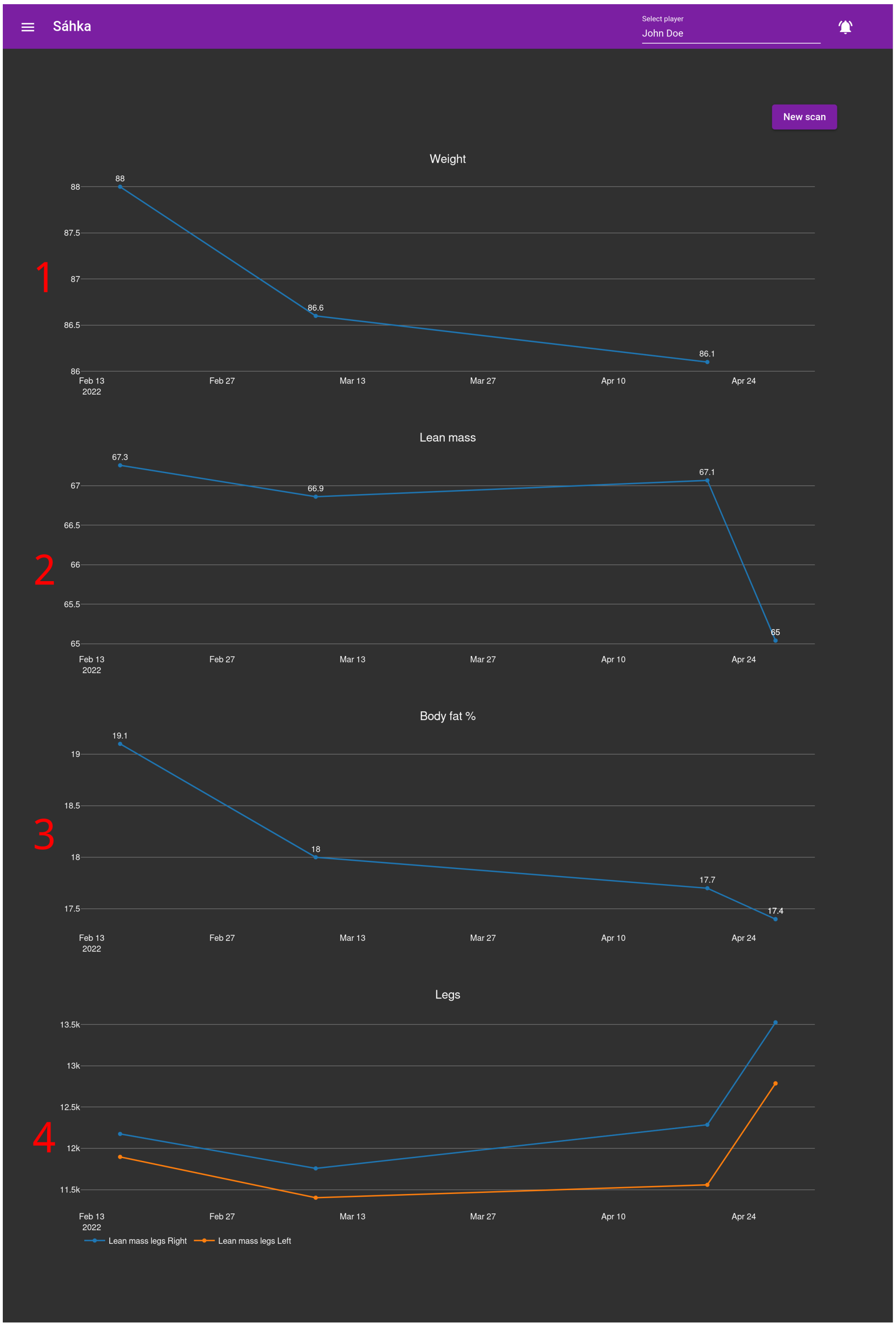

Figure 4.7: The page for visualizing the player's DXA data.

Figure 4.7 shows the page that visualizes DXA data. Graph one shows the player's weight over time. The second graph shows the player's lean mass over time. Graph three shows the player's body fat percentage over time. Lastly, the last graph shows the player's lean mass in each leg over time.

4.1.3 Performance Tests

As defined in Section 3.2, Sáhka must be able to collect, store and visualize data from different performance tests. Currently, TIL does not use any designated systems to track their players' physical performance tests. Instead, the coach performing the tests writes the results in an Excel sheet on their computer. However, this method is not privacy-preserving, not very organized, and if a coach wants to share the document with someone else, they must do it manually. Hence, they wanted to integrate these performance tests with Sáhka to manage this data better, ensure privacy, and manage access control.

Like the DXA, Sáhka has to collect the performance test data directly from the users since there are no third-party systems to fetch from automatically. The data is collected through a form on Sáhka's front-end, where coaches can enter the test results, similar to the form for DXA data.

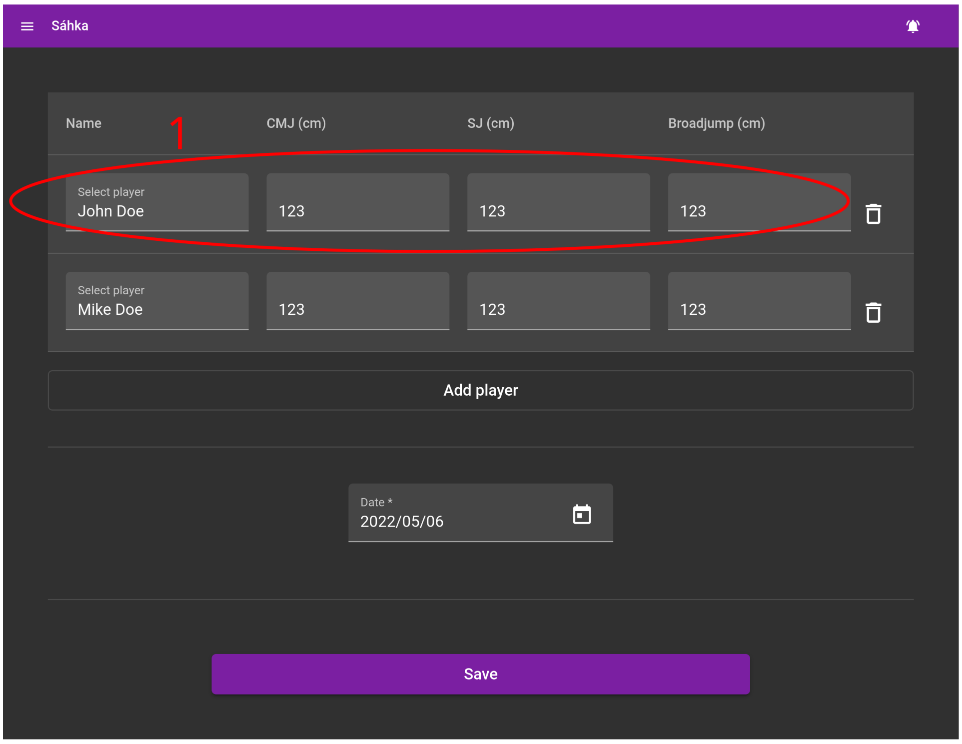

Figure 4.8: Example of form for performance tests.

Figure 4.8 shows the form for entering test results. The red circle highlights that the coach has selected a player named John Doe and entered the results from the three jump tests: CMJ, SJ, and BJ.

Another page visualizes the test data through graphs showing the test results over time.

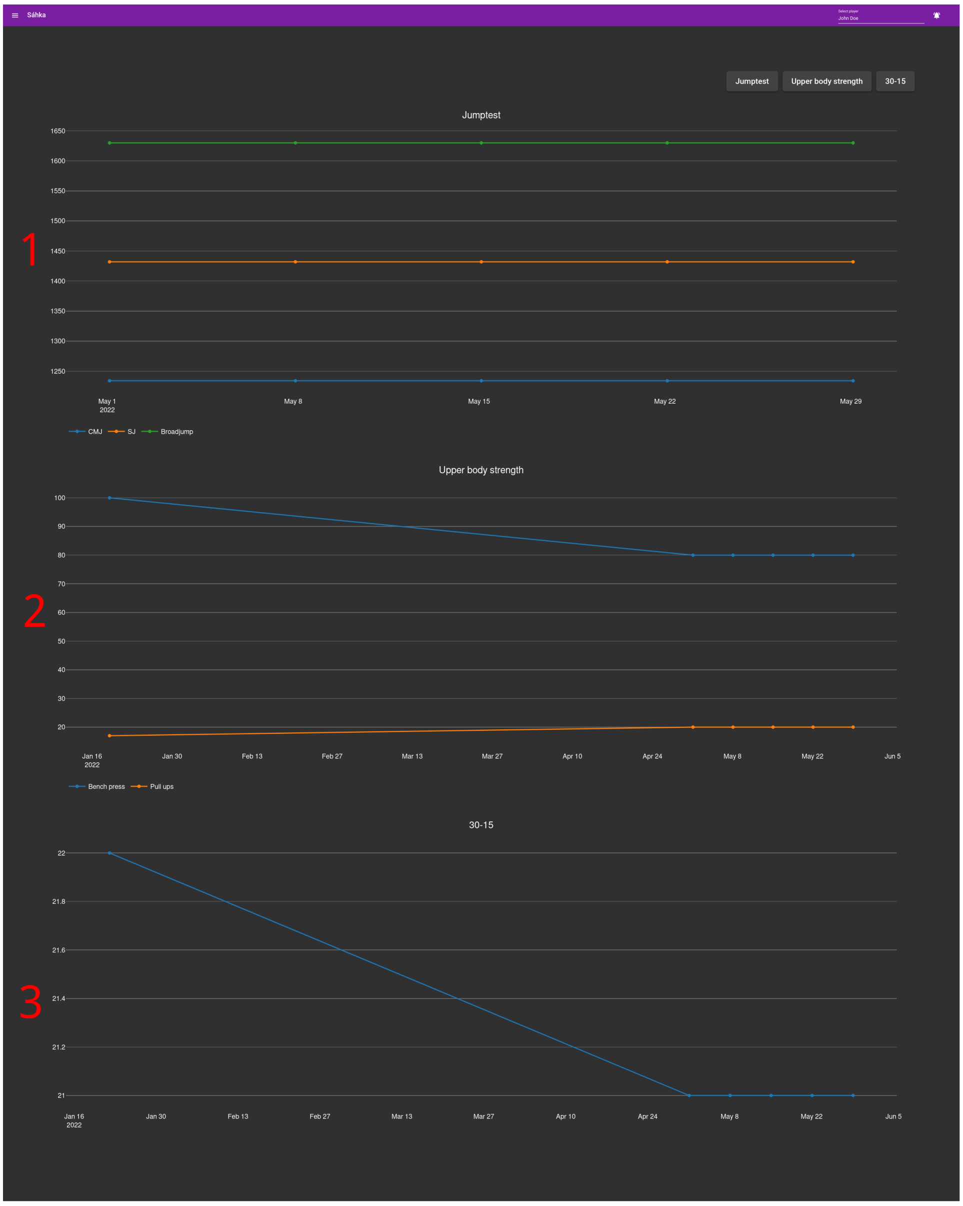

Figure 4.9: The page for visualizing a player's performance tests.

Figure 4.9 shows the page that visualizes test data. The first graph shows the 15-meter sprint results. Graph two shows the results from all the jump tests: CMJ, SJ, and BJ. The third graph shows the results from the strength tests: bench press and pull-up. Finally, the last graph shows the results from the $30\text{-}15_{\text{IFT}}$ tests.

4.1.4 Nutrition

As defined in Section 3.2, Sáhka must be able to collect, store and visualize nutritional data from the users. Several systems are available for tracking nutrition, for example, MyFitnessPal and Lifesum. However, these systems are relatively tedious to use because they require the amount of each ingredient in a meal to be specified. This is why TIL is currently not using any such system.

Instead, the coaches at TIL wanted a system that is quick and easy to use to increase the likelihood of the players using the feature. A simple solution would be to have the players write the contents of their meals or select them from a list. However, this would still require a lot of typing or looking through an extensive list of foods. To maximize the likelihood of the players actually using this feature, it needs to require as little effort as possible. The best solution for this would be to have the players take a picture of their meals. A picture provides a thousand words worth of information while requiring only a few clicks by the user. This would give coaches the primary information they wanted, what and when their players eat, while requiring minimal effort from the players. Figure 4.10 shows the form used to upload a meal image, and Figure 4.11 shows the page visualizing a player's meals.

As mentioned in Section 2.2.6, several systems are available today for tracking nutrition. However, the downside that these systems have in common, and the reason TIL is not using any of them, is the amount of complexity and effort required to record meals.

Instead, the coaches wanted a system that is quick and easy to use to increase the likelihood of the players using the feature. Hence, considering a picture is worth a thousand words, the solution implemented was for the players to take and upload a picture of their meals. This would give coaches the primary information they wanted, what and when their players eat, while requiring minimal effort from the players.

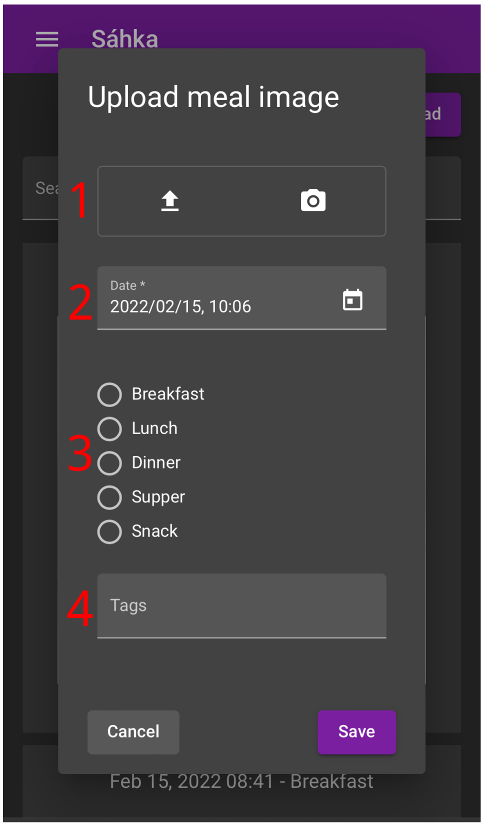

Figure 4.10: The interface where users can take and upload images of a meal.

Figure 4.10 shows the form used to upload images of a meal. Using input one, users can either upload an image from their device by clicking the icon on the left or open the camera and take a picture by clicking the camera icon on the right. In input two, the user can select what time the meal was consumed, which defaults to the current time. Input three is an optional input where the user can select what type of meal this was, either breakfast, lunch, dinner, supper, or a snack. Finally, input four is another optional input where the user can write some tags of what the meal contains, for example, meat, potatoes, and vegetables.

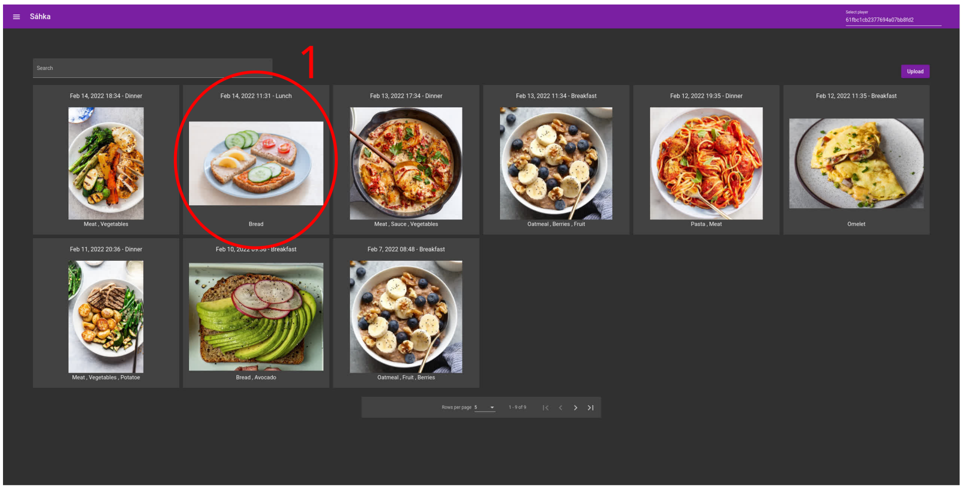

Figure 4.11: The nutrition page, showing a player's most recent meals.

Figure 4.11 shows the page that visualizes a player's meals. The meals are listed in order from most to least recent. The red circle highlights what the player had for lunch on the 14th of February 2022.

This method of tracking nutrition does not give details on the amount of micro- and macro-nutrients the players eat, though they can be estimated based on the contents of the meal, considering food such as meat is high in protein and pasta is high in CHO. This lack of detail is intentional since the coaches at TIL have reported that they do not want that level of detail; they only need a broad sense of what and when the players are eating.

4.1.5 Video Clips

As stated in Section 3.2, Sáhka must be able to collect video clips from multiple sources, including direct uploads from the users and Eliteserien Highlights.

Upload

Collecting videos directly from users is a relatively straightforward process. Users upload their videos to Sáhka's front-end using the interface shown in Figure 4.12, which then sends them to the back-end to be stored in the database. Also stated in the requirement specification for collecting videos was that they should not be stored locally when possible to save disk and bandwidth usage. However, since these videos come from the users, not other third-party systems, they might not be stored anywhere else and must be stored in the database to be persistent.

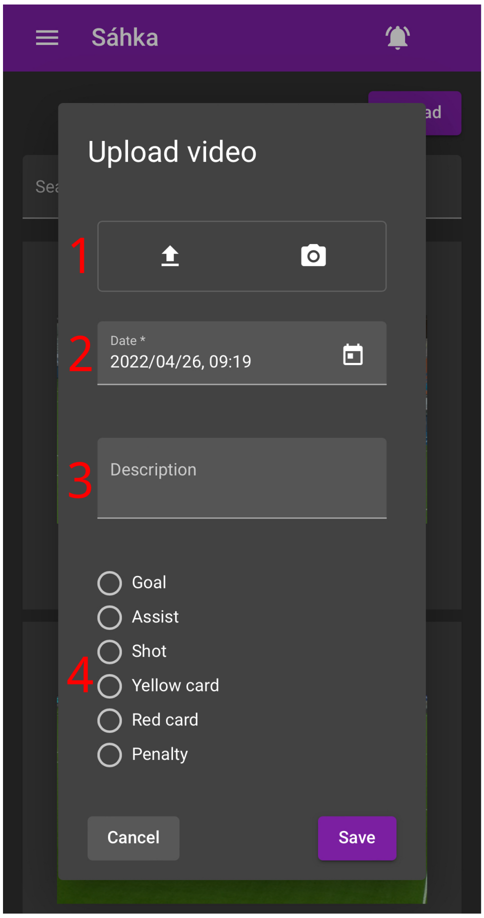

Figure 4.12: The interface where users can take and upload video clips.

Figure 4.12 shows the form used to upload video clips. Using input one, the user can upload a video from their device or open the camera and film a video directly. In input two, the user can set the time the video was taken, which defaults to the current time. Input three is an optional field where users can write a video description. Finally, in input four, the user can select a category to put the video in, either a goal, assist, shot, yellow card, red card, or penalty.

Eliteserien Highlights

Eliteserien Highlights is a product of ForzaSys, and the videos are accessible through their REST API. This means that these videos can be fetched automatically by Sáhka's back-end. Hence, the process that fetches data from third-party systems shown in Figure 2.8 was extended to also fetch data from Eliteserien Highlights, as shown by the line highlighted green in Figure 4.13.

```
function CollectXXXData
    data ← FetchData()
    data ← ProcessData(data)
    DatabaseInsert(data)
end function

loop
    CollectZXYData()
    CollectPmSysData()
    CollectEliteserienHighlightsData()
    Sleep()
end loop
```

Figure 4.13: Data collection algorithm [1]

The API used to fetch videos from Eliteserien Highlight contains an endpoint for fetching every video of one player. This endpoint also accepts a date as a parameter and will only return videos created after that date. Hence, giving the date of the latest video already stored in Sáhka ensures that only new videos are fetched. An example of the JSON response from this endpoint is shown in Figure 4.14:


```
{
  "playlists": [
    {
      "creator": {
        "id": "xxxxxxxxx",
        "name": "tags-video",
        "profile_image": null
      },
      "date": "2018-10-22T05:09:18.653686Z",
      "description": "Gult kort A-Team: John Doe.",
      "duration_ms": 25000,
      "events": [
        {
          "content_source": null,
          "description": "Gult kort A-Team: John Doe.",
          "from_timestamp": 3428000,
```

```
        "recording_timestamp":
↪  "2018-10-21T19:52:08.000000Z",
        "tags": [
          {
            "action": "yellow card",
            "player": {
              "id": 1575,
              "type": "player",
              "value": "John Doe"
            },
            "team": { "id": 7, "type": "team", "value":
↪  "A-Team" }
          }
        ],
        "to_timestamp": 3453000,
        "video_asset_id": 1863,
        "video_asset_name":
↪  "https://d22hh18o76pkhl.cloudfront.net/forzify-backend-ti⌋
↪  ppeligaen/xxxxxxxxx/xxxxxxxxx/thumb/hd"
      }
    ],
    "frontend_url":
↪  "https://highlights.eliteserien.no/playlist/xxxxxxxxx",
    "game": {
      "attendance": null,
      "date": "2018-10-21",
      "end_of_1st_half": "2018-10-21T18:45:33.000000Z",
      "end_of_1st_overtime": null,
      "end_of_2nd_half": "2018-10-21T19:52:42.000000Z",
      "end_of_2nd_overtime": null,
      "everysport_id": 3091526,
      "extra_referees": null,
      "finished_time": "2018-10-21T19:52:42.000000Z",
      "home_team": {
        "id": 11,
        "import_id": 269,
        "import_source": "FIKS",
        "logo_url": "https://cdn.forzasys.com/team-logos/ti⌋
↪  ppeligaen/xxxxxxx.png",
        "name": "B-Team",
        "short_name": null
      },
      "home_team_goals": 1,
```

```
        "id": 782,
        "import_id": "6903455",
        "import_source": "FIKS",
        "is_available_publicly": true,
        "officially_closed": true,
        "phase": "finished",
        "public_release_timestamp":
↪   "2018-10-21T20:30:00.000000Z",
        "referee_external_id": null,
        "referee_name": null,
        "round": 26,
        "stadium_name": null,
        "start_of_1st_half": "2018-10-21T18:00:21.000000Z",
        "start_of_1st_overtime": null,
        "start_of_2nd_half": "2018-10-21T19:02:39.000000Z",
        "start_of_2nd_overtime": null,
        "start_of_penalties": null,
        "start_time": "2018-10-21T18:00:00.000000Z",
        "stop_periods": [],
        "tournament_id": 3,
        "tournament_name": "Eliteserien",
        "visiting_team": {
          "id": 7,
          "import_id": 298,
          "import_source": "FIKS",
          "logo_url": "https://cdn.forzasys.com/team-logos/ti
↪   ppeligaen/A-Team.png",
          "name": "A-Team",
          "short_name": null
        },
        "visiting_team_goals": 2
      },
      "hd_thumbnail_override_url": null,
      "hd_thumbnail_url":
↪   "https://d22hh18o76pkhl.cloudfront.net/forzify-backend-ti
↪   ppeligaen/xxxxxxxxx/xxxxxxxxx/thumb/hd/xxxxxx.jpg",
      "id": "xxxxxxxxx",
      "is_accessible": true,
      "is_live": false,
      "is_placeholder": false,
      "is_private": false,
      "liked_count": 0,
```

```
      "minified_frontend_url": "http://highlights.eliteserien
↪   .no/webview/playlist/xxxxxxxxx",
      "rating": 1,
      "thumbnail_override_url": null,
      "thumbnail_url":
↪   "https://d22hh18o76pkhl.cloudfront.net/forzify-backend-ti
↪   ppeligaen/xxxxxxxxx/xxxxxxxxx/thumb/xxxxxx.jpg",
      "user_liked": false,
      "video_url": "https://api.forzasys.com/eliteserien/play
↪   list.m3u8/xxxxx:xxxxx:xxxxx/Manifest.m3u8",
      "view_count": 8
    },
    ...
  ],
  "total": 45
}
```


Figure 4.14: Example of Eliteserien Highlights response body.

As shown in Figure 4.14, the field "frontend_url" is the URL to the video file on their servers for this video clip. Since these videos are fetched from a third-party system, where they are already stored, it is not necessary to download and store them locally; instead, only the URLs need to be stored so the videos can be played from the front-end. However, both of these solutions come with benefits and downsides.

One benefit of only storing the URL is that it is much cheaper in terms of storage and bandwidth compared to downloading a video. Additionally, there may be legal issues with downloading and distributing videos from other systems.

On the other hand, only storing the URLs of videos makes Sáhka reliant upon Eliteserien Highlights being online for users to play the videos. Should Eliteserien Highlights' servers be unavailable, the users will see every video without being able to play them. By contrast, downloading the videos will make them available to the users so long as Sáhka itself is online.

Furthermore, being reliant upon a third-party system can be detrimental in the long run if, for example, the system becomes discontinued or makes its services unavailable. Additionally, if Eliteserien decides to only store videos for a certain period, Sáhka becomes limited to the same period.

Due to the increased storage and bandwidth costs associated with downloading the videos, only the URL is stored in Sáhka's database. The response from Figure 4.14 also contains some information that is irrelevant for Sáhka and therefore filtered out. An example of how one video is stored in the database is presented in Figure 4.15.

```
{
  "_id": ObjectId("620b8098f633e7eb621aeba5"),
  "Player": ObjectId("61fbc1cb2377694a07bb8fd2"),
  "Description": "Gult kort A-Team: John Doe.",
  "Action": "Yellow card",
  "Date": ISODate("2019-05-19T18:00:00.000Z"),
  "Source": "Eliteserien",
  "Url": "https://highlights.eliteserien.no/embed/playlist/xx⌋
↪   xxxxxx",
  "ThumbnailUrl":
↪   "https://d22hh18o76pkhl.cloudfront.net/forzify-backend-ti⌋
↪   ppeligaen/xxxxxxxx/xxxxxxxx/thumb/xxxxxxxx.jpg",
  "CreationDate": ISODate("2019-05-19T20:19:33.175Z")
}
```

Figure 4.15: Eliteserien Highlights data after being processed.

The videos are presented to the users as a list of thumbnails of each video (Figure 4.16). When playing a user-uploaded video, it is fetched from the back-end and played in the front-end. When playing a video from Eliteserien Highlights, the URL is used in an iframe[1], which plays the video from their servers.

1. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe

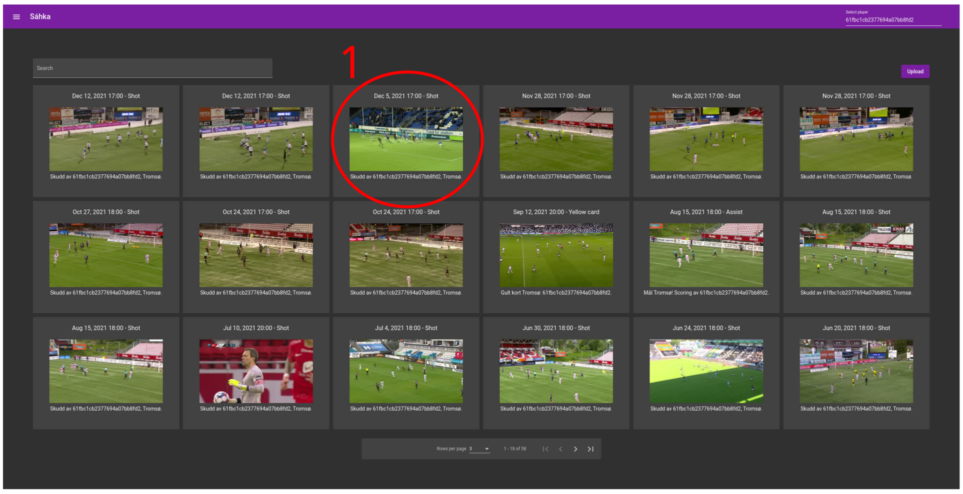

Figure 4.16: The video clips page, showing a player's most recent clips.

Figure 4.16 shows the list of all the player's videos in order from most to least recent. For example, the red circle highlights a video from Eliteserien Highlights where the player had a shot against the opponent. Clicking this video will redirect to the page in Figure 4.17, where the user can play the video.

Figure 4.17: Playing the video highlighted in Figure 4.16.

4.1.6 Objectives

As stated in Section 3.2, Sáhka must be able to collect, store, and visualize players' objectives. Setting objectives for players on what they should work on and improve is a crucial aspect of a coach's job. During a coach-player meeting, they discuss and identify areas that the player should focus on improving for a period. After that period, they meet again to review and discuss the player's progress.

Currently, TIL does not have a designated system to keep track of what is discussed during these meetings. Each coach just writes their notes in files kept in OneDrive[2] or something similar. Not only is this poor organization, but it also means that players have to ask their coach or take their own notes if they want to review what they have discussed with their coach in the past.

Instead, the coaches wanted this integrated into Sáhka as a place they can write down the objectives discussed during a coach-player meeting. This would allow both the coach and player to review the objectives throughout the period. Additionally, the coaches wanted to write private notes on each objective that they could then discuss with the player at the end of the period.

These features were implemented as a page in Sáhka's front-end, shown in Figure 4.18. Coaches can write down the objectives discussed during a meeting, which are sent to the back-end and stored in the database. These objectives are accessible by the coaches and the player. Additionally, the coaches can write notes next to each objective. The notes are only accessible by the coaches, not the player.

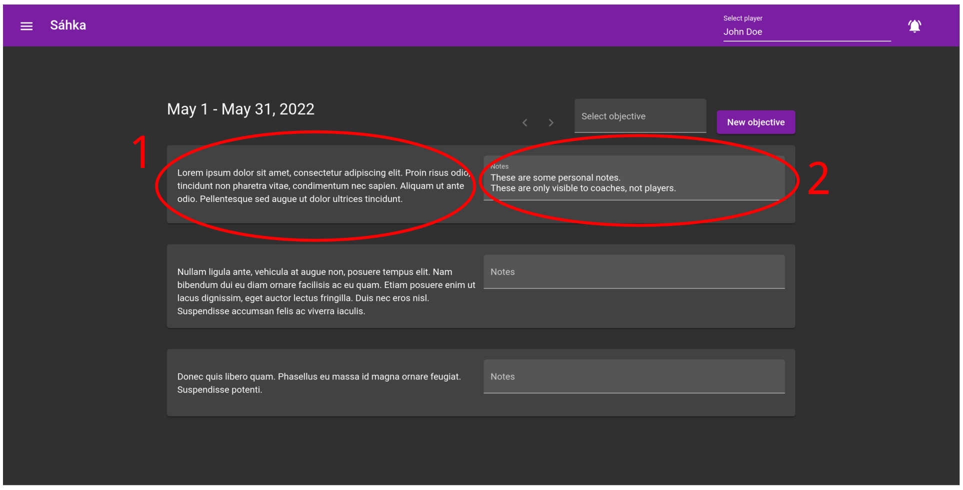

Figure 4.18: The objectives page, showing a player's current objectives and the coach's notes.

Figure 4.18 shows the objectives page from a coach's perspective. The page

2. `https://www.microsoft.com/en-us/microsoft-365/onedrive/online-cloud-storage`

shows that the player has three objectives, the first objective being highlighted by circle one. Next to each objective is an input field where coaches can write notes. Circle two highlights the notes for the first objective.

Also stated in Section 3.2 is that Sáhka must adapt to feedback received during development. In the initial version of the objectives page, there was only one text field for coaches to write notes per objective, and they were shared amongst the coaches. This meant there was no way to determine who wrote what in the notes and when. Instead, they wanted the notes to be divided into entries. Each entry would then provide information about which coach wrote it and when. Hence, the implementation was modified to accommodate this.

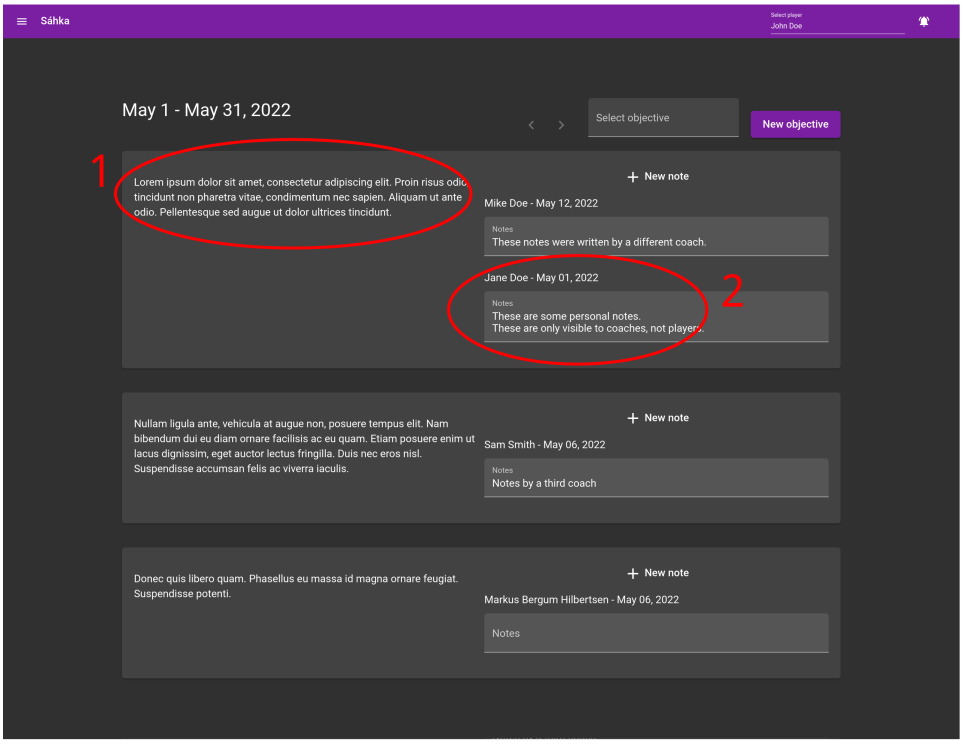

Figure 4.19: The objectives page after adapting to the feedback from TIL's coaches.

Figure 4.19 shows the new version of the objectives page. The page still shows that the player has three objectives, the first objective being highlighted by circle one. However, coaches can now add a new note next to each objective when they want to write something. The notes are then listed with the name of the coach and the time it was written. For example, circle two highlights a note written by Jane Doe on the 1st of May 2022.

4.1.7 Journal

As defined in Section 3.2, Sáhka must be able to keep a private journal for each user. This was inspired by the medical industry, where doctors keep a detailed medical history for patients containing information such as illnesses, test results, treatments, and allergies. Reviewing a patient's medical history is crucial and will often directly impact the patient's treatment [47]. Similarly, coaches should consider a player's history when planning their training and workload. For example, players with a history of specific injuries may need a different training regiment than players who have never been injured.

There already exist many types of medical journals and templates for how they should look [48, 49, 50]. However, these templates often contain a lot of questions and checkmarks about, for example, medications or conditions a patient may have. This level of detail and specificity is unnecessary for the journal a football coach needs. Hence, a more straightforward journal format is adequate, with only text fields for taking notes.

The journal implemented allows users to write entries containing any notes they have about, for example, performance, health, training, or injuries. Each entry also contains a date, representing when it was created. Players only have one journal, whereas coaches have one journal per player for better organization. Additionally, every journal is private and only accessible by the user who wrote it.

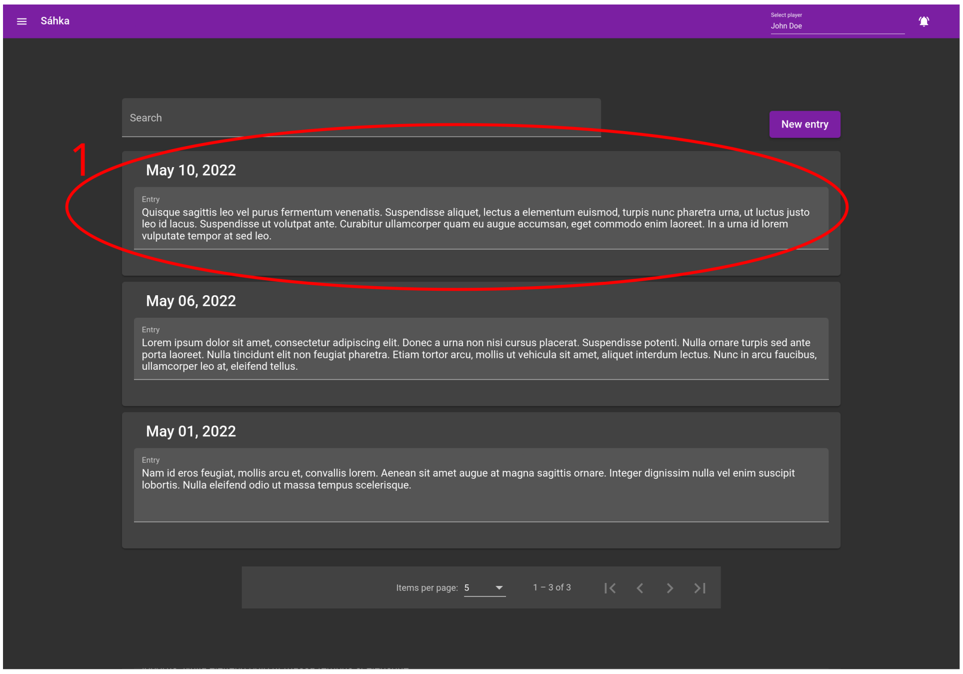

Figure 4.20: Journal page, showing the coach's journal entries.

Figure 4.20 shows the user's journal as a list of entries from most to least recent. For example, circle one highlights a journal entry written on the 10th of May 2022.

4.2 Analysis

One of the coaches' jobs is to monitor their players and intervene if they detect any dips in their performance or health, such as reduced performance, too little sleep, or not eating enough. The initial functionality of Sáhka was to aid coaches in this process by collecting player data from different sources and presenting it to them in one application. However, regularly checking every player on a team is still tedious and takes time away from the coaches to do their many other jobs. Hence, instead of manually monitoring their players, the coaches wanted a system that automatically analyses the data and notifies them if they need to intervene with a player. This expanded the functional requirements in Section 3.2 to state that Sáhka must be able to analyze the data collected to detect specific events and notify coaches when they occur.

The analysis works by passing relevant player data through a series of filters assembled in a DAG. Starting at the leaf nodes, if the data passes through all the filters and reaches the root of the DAG, then the coaches are notified. Each filter is one factor the coaches want to be notified about; for example, a filter can be "if a player has slept less than five hours on average the last week." The filters can then be combined in the DAG using boolean operators to create more specific notifications. Using the DAG from Figure 4.21, the coaches would be notified if a player has slept less than five hours on average the last week and had an average mood of less than two the last week.

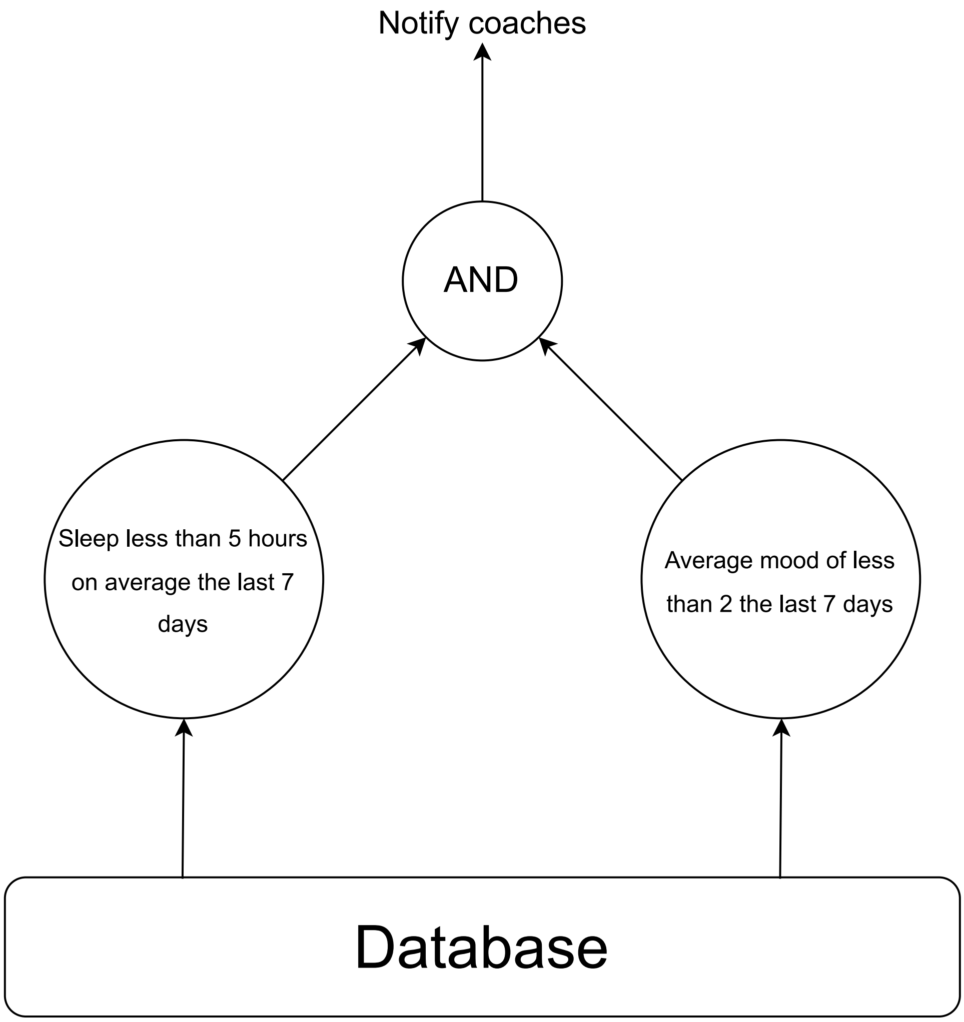

Figure 4.21: DAG showing the structure of an analysis graph.

Figure 4.21 shows an example of an analysis graph. The database is an active component, which means when new data is available, it will be published to the bottom nodes where it is relevant. For example, if a player reports how long they slept the previous night, the database will send the sleep data to the bottom left node. Each node processes its incoming data according to its filter and sends the result to the next node. At the root of the graph, the result will be to either notify the coaches that the event has occurred or not.

Running an analysis only when new relevant data is inserted ensures no redundant analyses are performed. Furthermore, it allows the coaches to be notified as soon as possible, compared to only periodically if the analyses were run at an interval. However, because Sáhka's front-end is implemented as a web application, it is not possible to send an actual push notification to the coaches. Instead, they have to check the website for any new events. This could be worked around by sending the notifications through email since the coaches' emails are known through authentication. However, this has not been implemented in the current prototype, and implementing push notifications, either through email or a native application, is left for future work.

This data flow, where the data comes from the database, passes through a series of filters, and possibly reaches the end-user, is similar to stream processing systems like Johka [51], which accepts a stream of data from a source and passes it through a series of pipes before combining it into a single result. Additionally, using DAGs to model computation flow is popular in distributed systems like Dryad [52], where the vertices represent programs or processes, and the edges are channels between them.

4.3 Summary

This chapter has presented the design and relevant implementation details of Sáhka. Based on the system model from Section 3.1, the system was designed in a four-tiered hierarchy with a database, back-end, front-end, and external third-party systems. Additionally, the system has been integrated with several data sources, some of which are third-party systems, and some come from the users directly.

5 Experiments

This chapter presents an evaluation of Sáhka, including the validity of some of the design choices made and a final user evaluation of the whole system.

5.1 Video Disk Footprint

As discussed in Section 4.1.5, the primary reason for only storing the URL of third-party videos is the storage cost of downloading and storing the actual videos. This experiment will quantify that cost and compare it to the current solution of only storing the URL.

Sixteen videos from Eliteserien Highlights were used for this experiment. The videos were first stored using the current solution with only the URL and some metadata. Then the videos were downloaded and stored along with the same metadata.

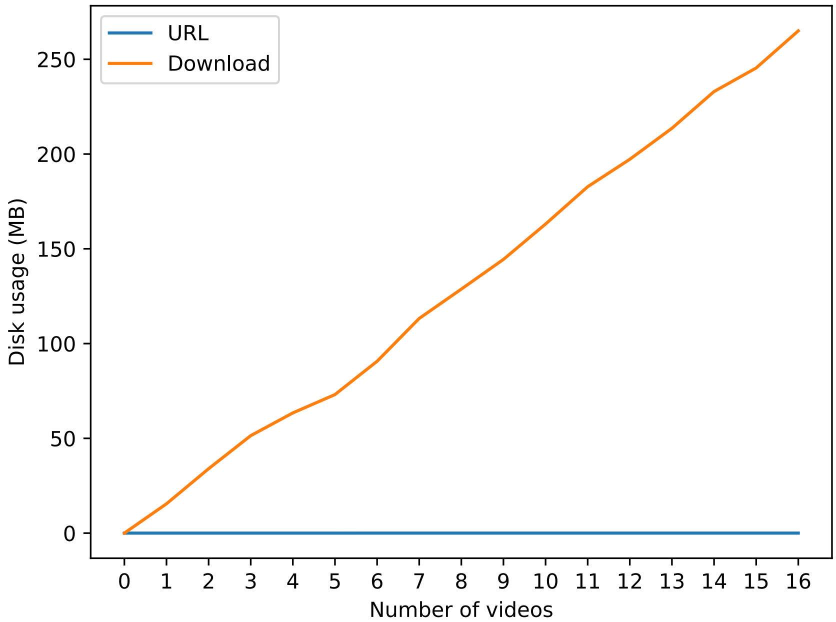

Figure 5.1: Disk usage when storing the URL's to videos compared to storing the actual videos.

Figure 5.1 shows that downloading and storing videos is significantly more expensive than storing URLs. Storing the sixteen videos required more than 250MB of disk usage, whereas, with the URLs, it only took about 5,6KB. That is a difference of five orders of magnitude. This significant difference is unsurprising, considering the URLs used in this experiment were around sixty characters long, whereas the videos were, on average, 16MB each.

5.2 Local vs. Remote Access

One of the main reasons for storing third-party data locally in Sáhka was to reduce query times for users. The claim was that fetching data from the database is faster than fetching and processing data from its remote source. Hence, users would have to wait a shorter time for a response from the back-end. This experiment aims to measure the latency for fetching data locally and compare it to fetching the same data from its remote source.

This experiment was performed three times, once for each third-party system.

In each experiment, the latency was measured fifty times for fetching the same data from the database and then from the third-party system. For PmSys, the data fetched was one year's worth of one player's data. For ZXY, the data from one player's training session was fetched. Finally, for Eliteserien Highlights, all the videos of one player were fetched. The latency measured was the time it took for the back-end to receive a request for data until the data was sent back.

The computer running this experiment had the following specifications:

Machine: HP Pavilion Power Laptop 15-cb0xx

OS: Arch Linux x86_64 (5.17.5-arch1-1)

CPU: Intel i7-7700HQ (8) @ 3.800GHz

GPU: NVIDIA GeForce GTX 1050 Mobile

Memory: 16GB

Additionally, since the remote data had to be fetched through HTTP requests, the experiment was performed at UiT with a one Gigabit Ethernet connection.

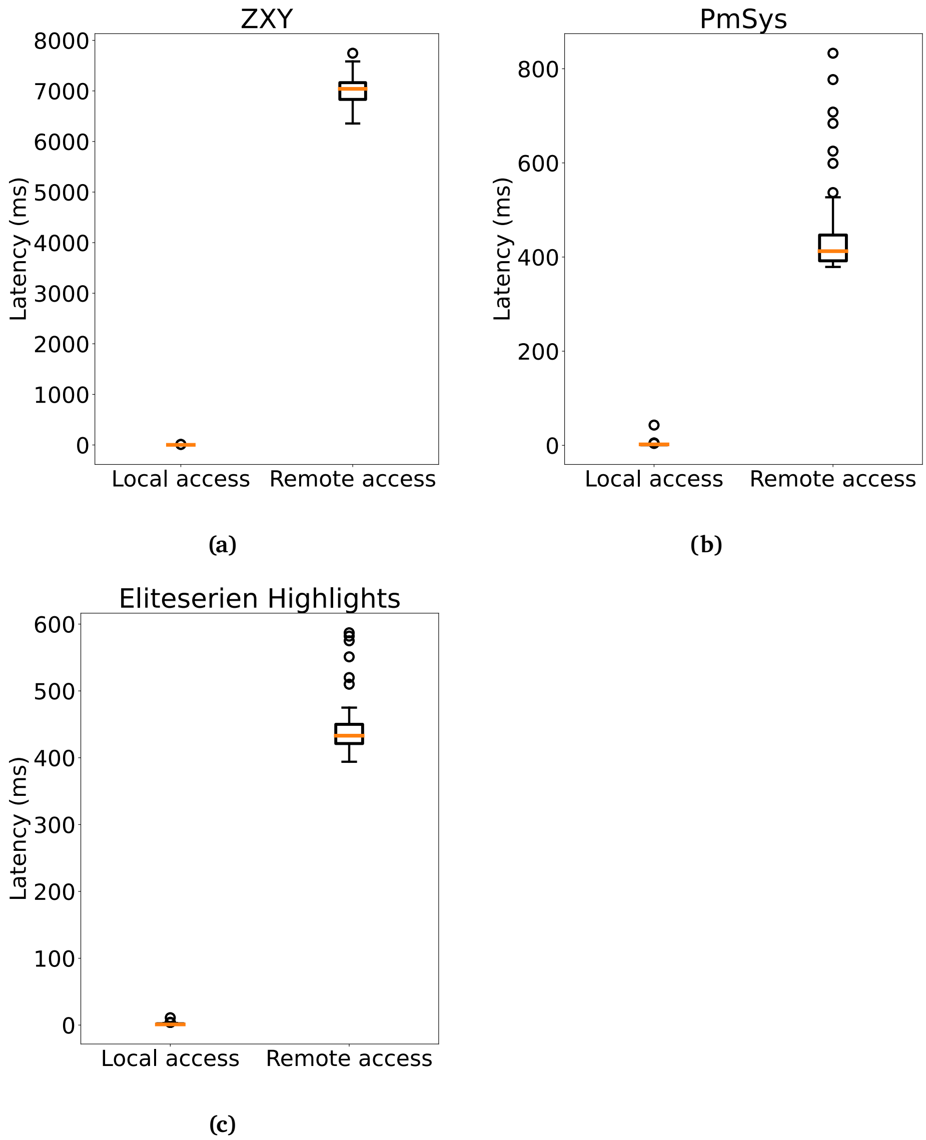

Figure 5.2: Latency of fetching (a) PmSys, (b) ZXY, and (c) Eliteserien Highlights data locally compared to fetching it from the remote source.

Figure 5.2 shows the results from this experiment. Figure 5.2 (a) shows that the local latency for fetching ZXY data is, on average, 3,6 milliseconds, whereas the remote latency is 7 seconds. For PmSys (b), the local latency is around 2,6 milliseconds, and the remote latency is 4,5 seconds. Lastly, fetching videos (c) from the database has an average latency of 1,7 milliseconds, while fetching from Eliteserien Highlights has a 4,4-second latency.

The takeaway is that, as suspected, fetching data from third-party systems is significantly slower than fetching from the local database. Hence, third-party data should be stored locally to reduce the time users have to wait for data.

5.3 User Evaluation

Section 3.2 stated that the system shall be developed in close collaboration with TIL and must continuously adapt to feedback received from their coaches. This is a critical requirement because the system is, after all, developed for the end-users: coaches and players.

This experiment will evaluate the usability of Sáhka by having coaches and players at TIL answer a survey. Additionally, some of the questions will evaluate which avenues are worthwhile exploring more in the future.

Seven educated coaches and seven elite male football players at TIL participated in the survey. The coaches and players answered different surveys, though some questions overlapped. All questions, with one exception, were answered using the Likert Scale [53], with one being the worst option and five the best.

5.3.1 Coaches

The first two questions will evaluate the coaches' satisfaction with the current status quo - having multiple tools for collecting and visualizing data - and the need for a system that collects all the different data into one platform. These questions will show whether there is a market for a system like Sáhka among coaches or whether they are satisfied with the tools available.

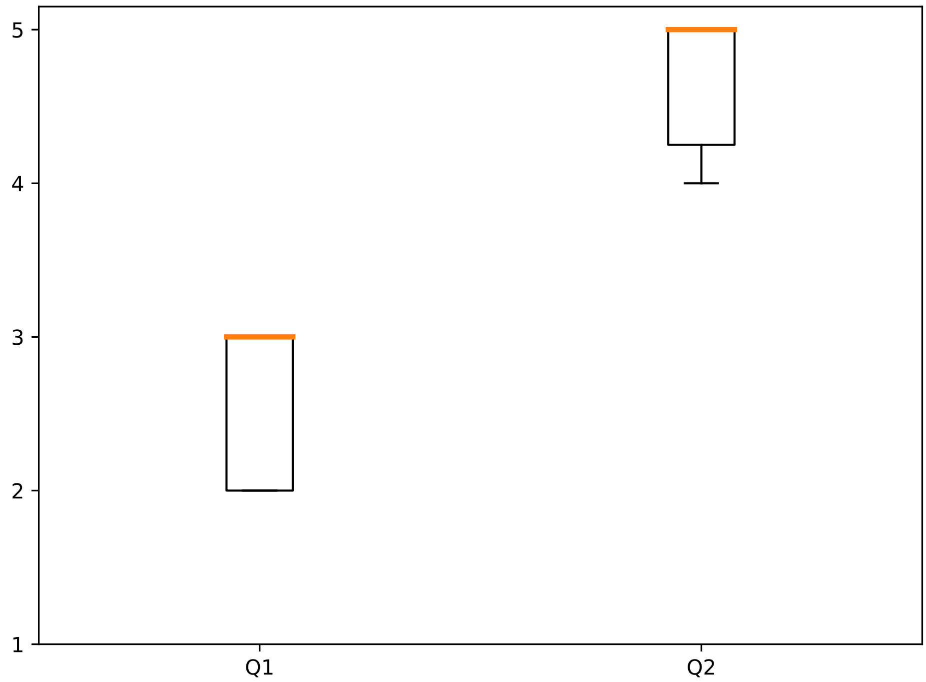

Figure 5.3: Q1: Today you have multiple services where you can fetch and visualize football data. How satisfied are you with having this in multiple different services?
Q2: How satisfied would you be if you could have all the data collected into a straightforward platform?

Figure 5.3 shows that the coaches are not very satisfied with using several different tools, and would rather have all data collected into one system. This demonstrates that there is a definite market for a system like Sáhka among coaches.

The following two questions will evaluate the coaches' satisfaction with how their current tools visualize data compared to how Sáhka visualizes data. To incentivize coaches to use Sáhka, it must be better than, or at least as good as, the current tools at visualizing data. Otherwise, they may want to keep using the current tools because they present the data better.

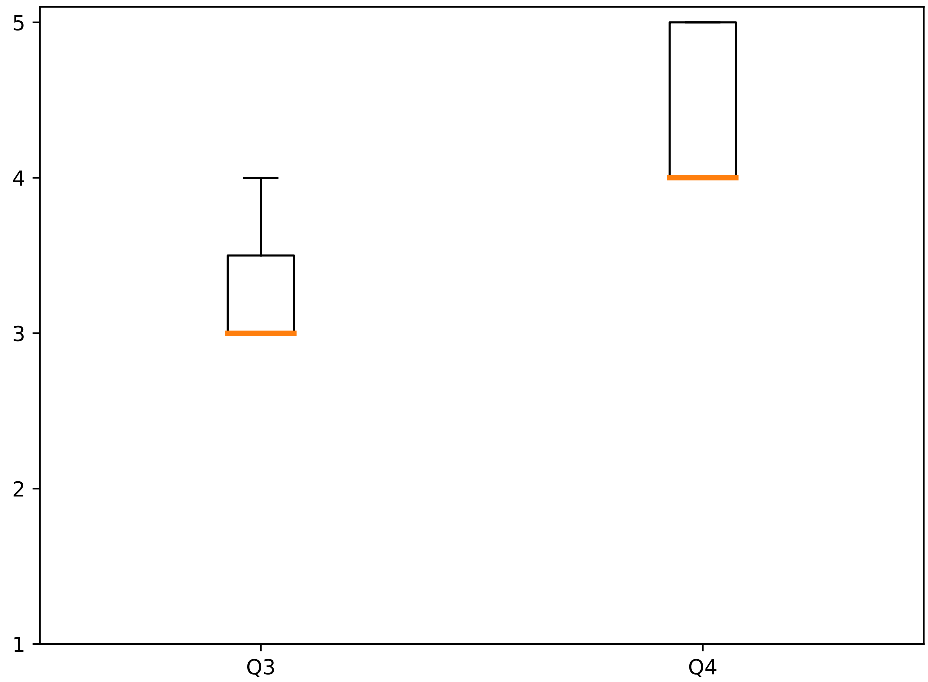

Figure 5.4: Q3: How satisfied are you with how existing services visualize data?
Q4: How satisfied are you with how Sáhka integrates and visualizes data in one straightforward platform?

Figure 5.4 shows that the coaches are moderately satisfied with how their current tools visualize data, and more satisfied with how Sáhka visualizes data. This will incentivize the coaches to use Sáhka if for no other reason than to get better visualization of the data collected.

The following three questions evaluate the importance of coaches following up with each individual player throughout a season, for example, through weekly coach-player meetings. Further, they evaluate whether the coaches have sufficient tools available for these follow-ups, and whether Sáhka could be a more helpful tool.

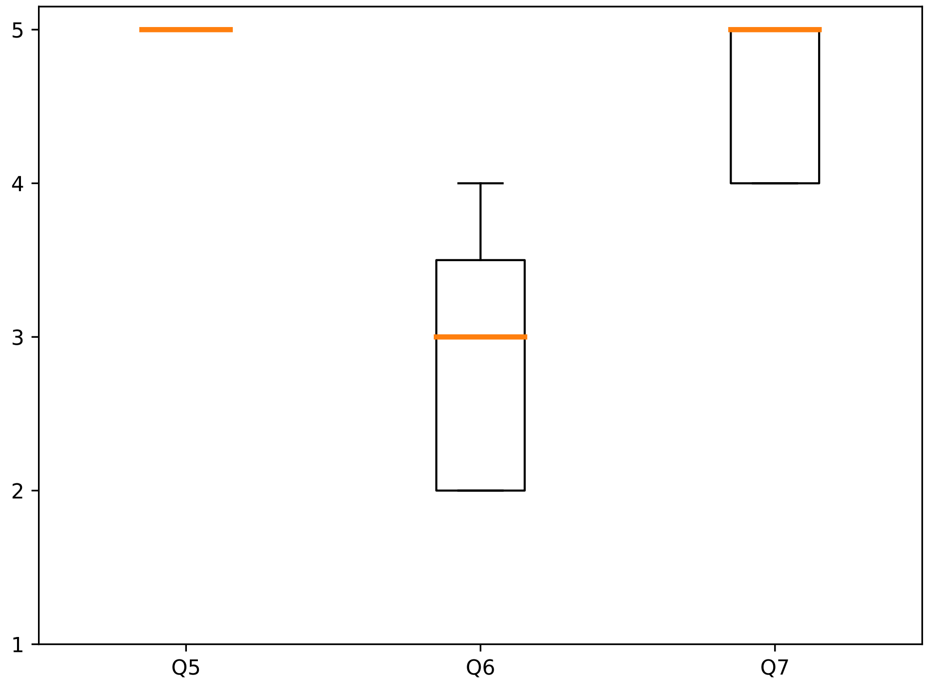

Figure 5.5: Q5: How important is it for you to follow up with each individual player (and relations between individual players) throughout a season?
Q6: How sufficient do you feel current tools are for following up with each individual player throughout a season (for example, through weekly coach-player meetings)?
Q7: How much do you think Sáhka would help you with following up with each individual player (for example, through weekly coach-player meetings)?

The results from Figure 5.5 show that the coaches unanimously agree that following up with each individual player is crucial. Additionally, they feel that the currently available tools are moderately sufficient for this task and that Sáhka could significantly help them follow up with individual players.

Question eight will evaluate which data sources coaches consider most important. This information is crucial for the future when deciding what type of data to focus on integrating and improving.

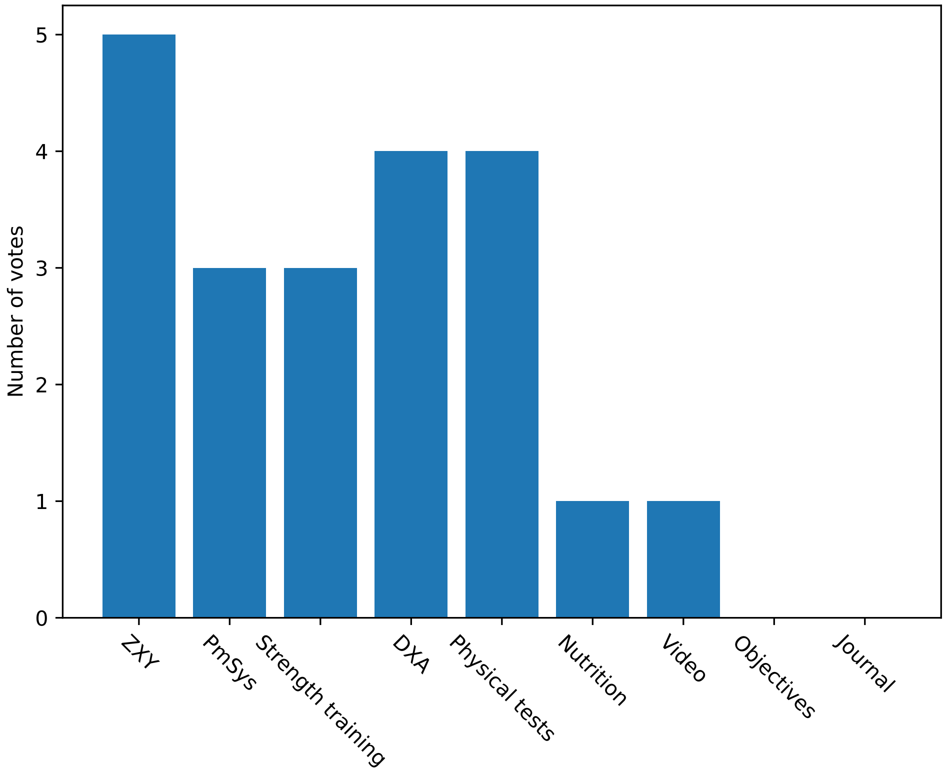

Figure 5.6: Q8: Which of the data sources in Sáhka do you think is most important (choose three)?

Figure 5.6 shows the results from question eight. The coaches are relatively varied in which data sources they find important, with ZXY being the top pick, followed by DXA and physical tests.

Question nine evaluates whether coaches would find it helpful to have algorithms that analyze data and notifies them of specific events. Currently, Sáhka analyses the data and notifies based on specific criteria set by the coaches. This question will evaluate whether it is worthwhile to continue this avenue and expand it with Machine Learning algorithms in the future.

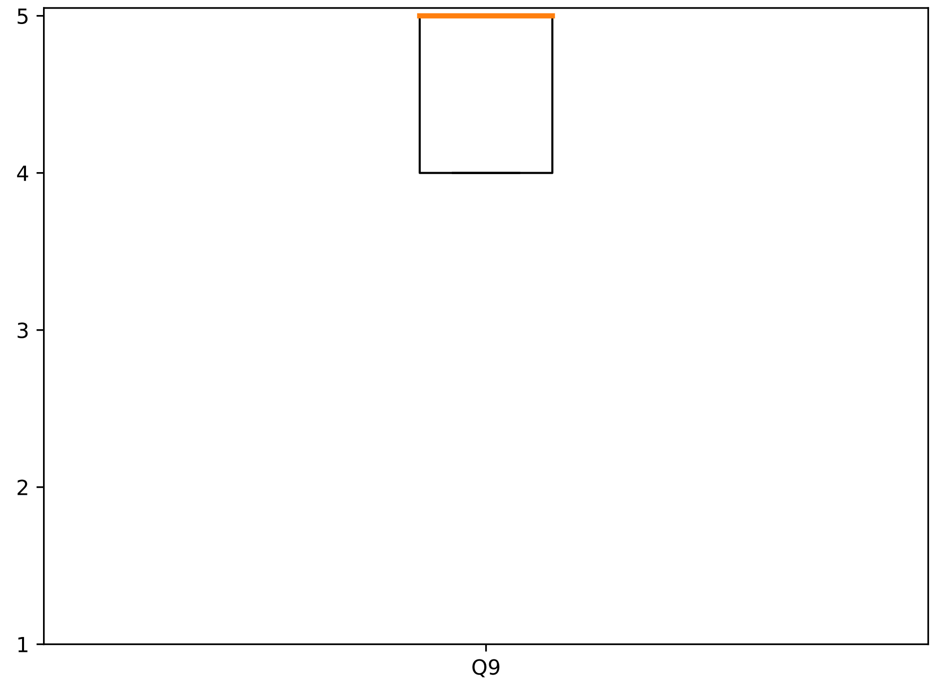

Figure 5.7: Q9: How much do you think it would help you to have algorithms that analyze data and notify you about specific events?

The result from Figure 5.7 shows that the coaches would find it very helpful with algorithmic analyses that can notify them of events. Hence, it is worthwhile exploring more analysis and Machine Learning solutions, which, as mentioned, will be done in the future.

As mentioned in Section 1, future work around Sáhka will focus a lot on video and different solutions therein. This question will evaluate whether the coaches find this a worthwhile direction to expand.

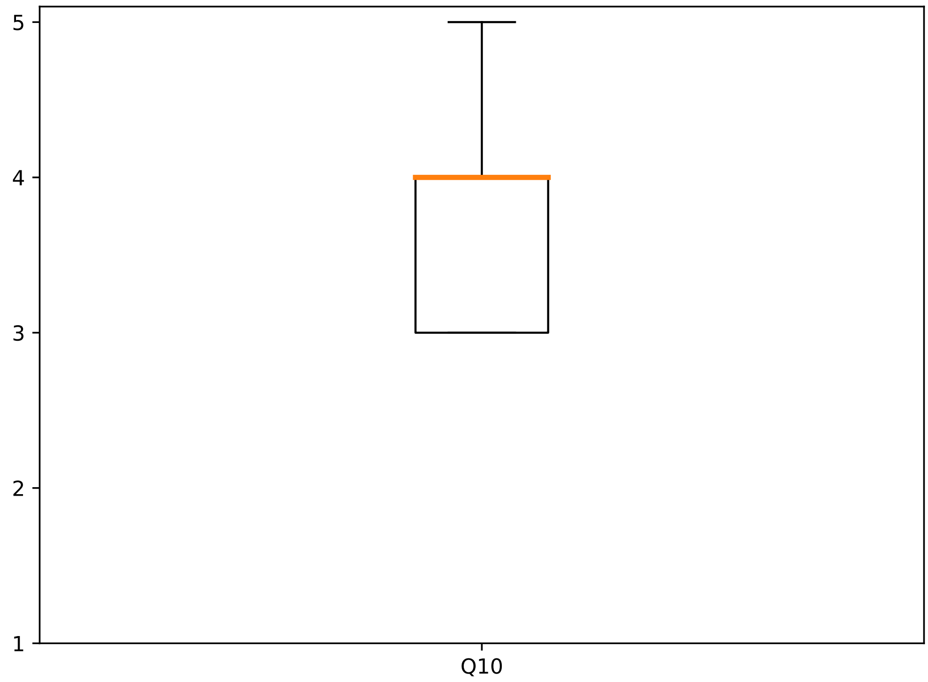

Figure 5.8: Q10: We are planning to focus on solutions and integration of video in the future. How important would it be for you to integrate and simplify different video solutions?

Figure 5.8 shows that the coaches would find more video solutions and integration quite important. This is a good indication that the two theses that will explore different video solutions are worthwhile.

The last question for the coaches will evaluate their overall satisfaction with Sáhka and how likely they are to use it. This will indicate whether this thesis successfully developed a new tool for coaches, which will help them and improve their ability to coach.

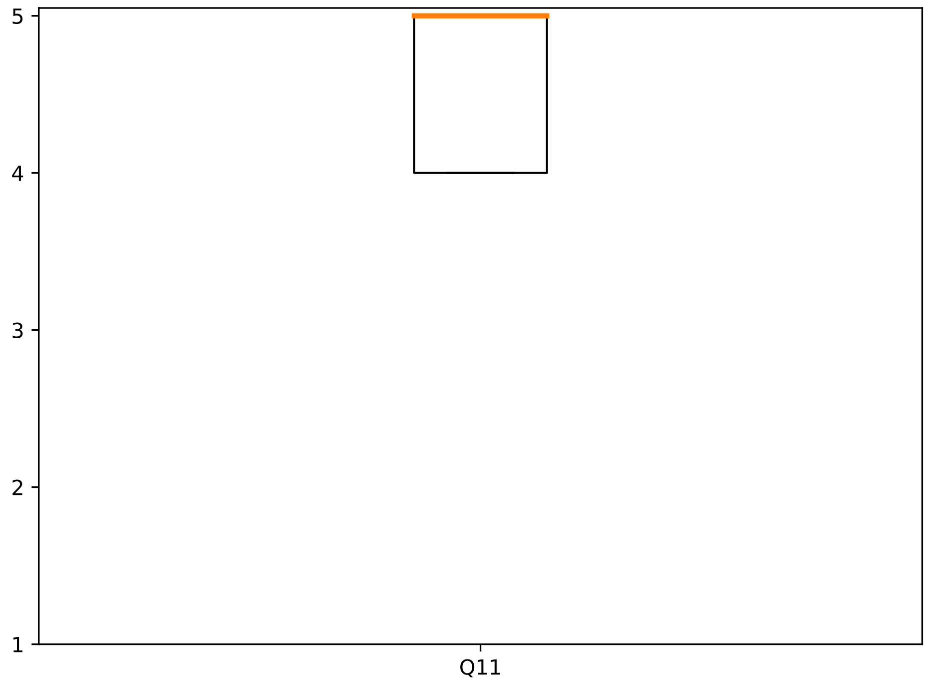

Figure 5.9: Q11: How likely is it that you will use Sáhka if available?

Figure 5.9 shows that the coaches are highly likely to use Sáhka. Some of the coaches have already been using Sáhka actively throughout its development, and this result indicates that they, and the rest of the coaches, want to continue doing so.

5.3.2 Players

For the players, the first three questions evaluate whether they find it important to have easy access to all the data collected about them. Further, they evaluate whether the players are satisfied with their current access to said data, and how satisfied they would be with a system that collects their data into one platform. This will indicate whether there is a market for a system like Sáhka among the players or not.

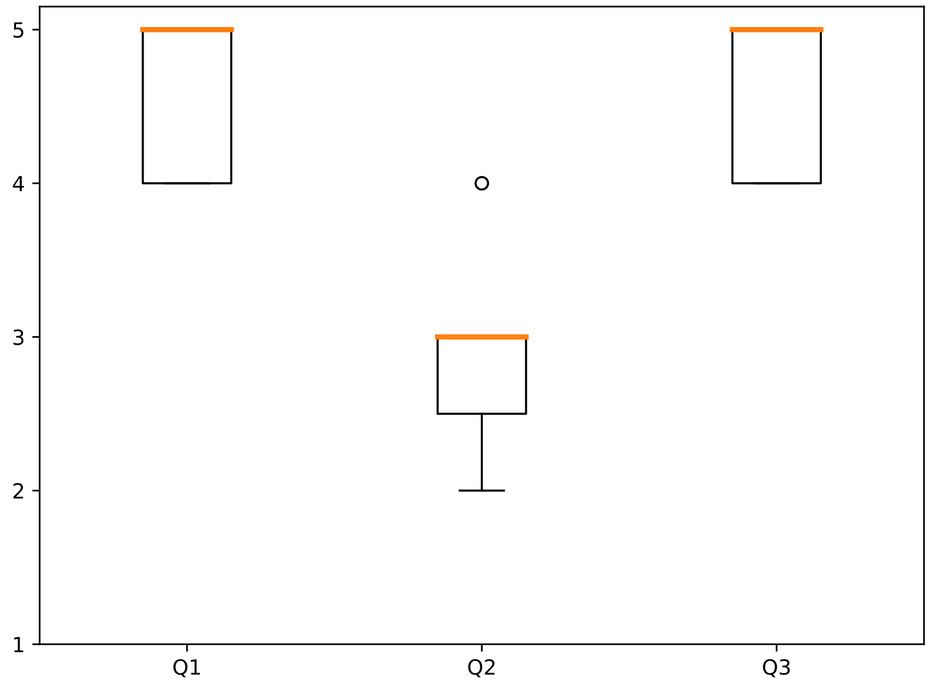

Figure 5.10: **Q1:** Today a lot of different data about you is collected. How important is it that you have access to your personal data?
Q2: How satisfied are you with how good and easy access you have to your personal data?
Q3: How satisfied would you be if you could have all your personal data collected in one simple and available platform (Sáhka)?

Figure 5.10 shows that the players find it very important to have access to the data collected on them. Additionally, it shows that they are relatively satisfied with the current tools for accessing their data, but they would be significantly more satisfied using one system that gives them access to all their data. Hence, there is a definite market for a system like Sáhka among the players.

The following three questions evaluate how important it is for players to be personally followed up by their coaches. Additionally, they evaluate whether the players currently have adequate tools for preparing and conducting personal follow-up meetings with a coach. Lastly, they will evaluate if the players think a system that collects all their personal data in one platform would help prepare and conduct personal meetings with a coach.

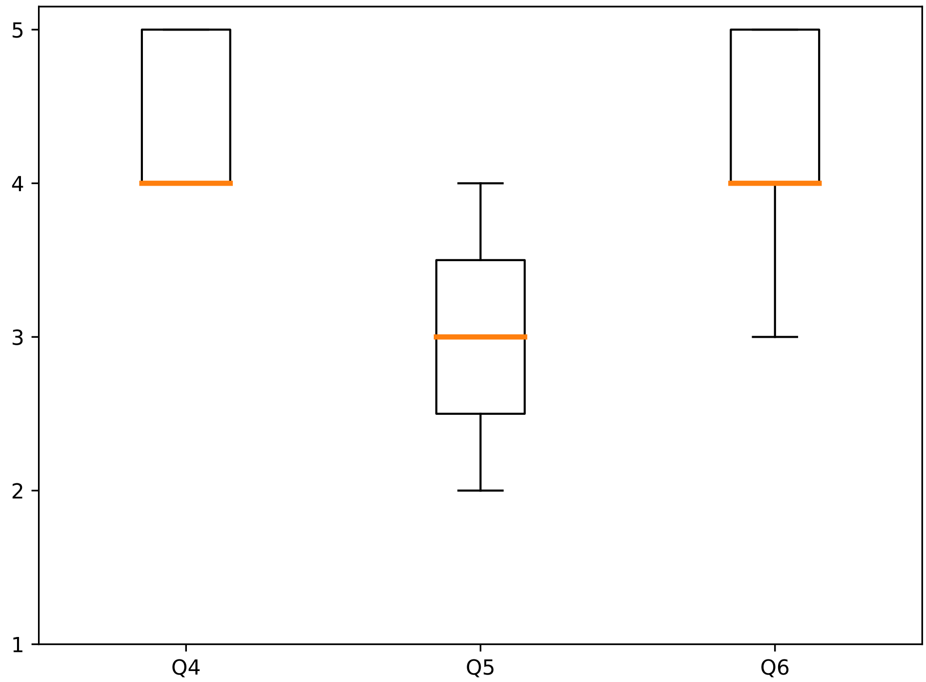

Figure 5.11: Q4: How important is it for you to be personally followed up by your coaches?
Q5: Do you have sufficient tools to prepare and conduct personal meetings with a coach?
Q6: We have developed a system that collects all your personal data into one system (Sáhka). How much do you think this system would help you prepare and conduct personal meetings with a coach?

The results from Figure 5.11 show that the players find it very important to be personally followed up by their coaches. Additionally, the tools they currently have are only moderately adequate for preparing and conducting personal meetings with a coach, and a system like Sáhka would be more helpful. This, combined with the results from Figure 5.5, demonstrates that a system like Sáhka would be a beneficial tool for both coaches and players in enabling follow-up of each individual player, which they agree is extremely important.

Question seven will evaluate which data sources the players consider most important. This will complement the results from Figure 5.6, which evaluated which data sources the coaches considered most important.

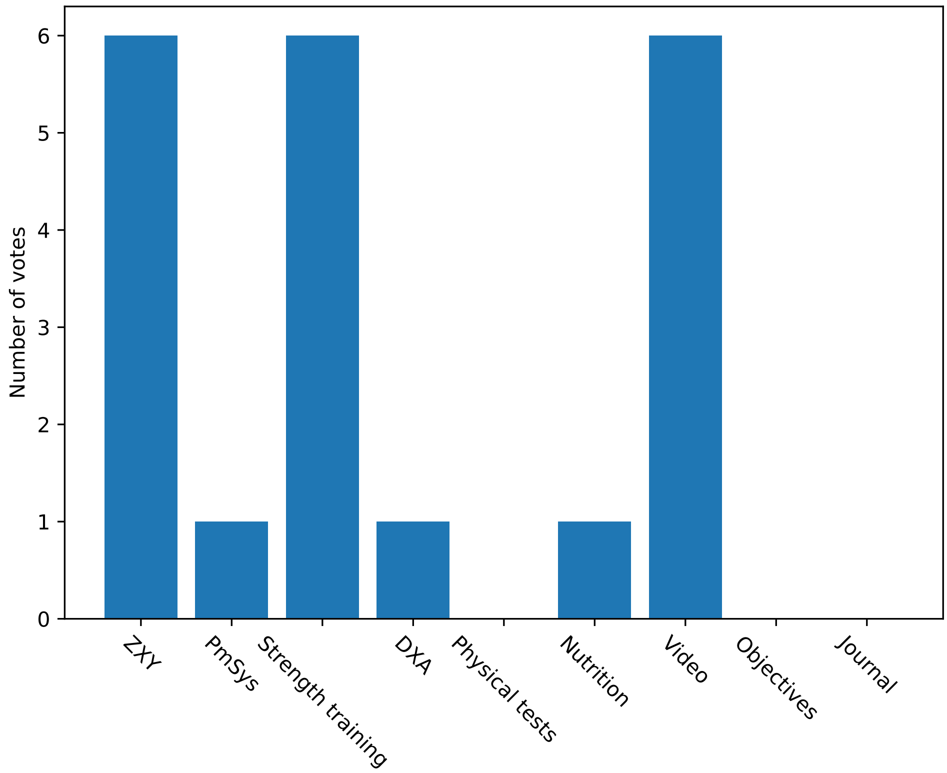

Figure 5.12: Q7: Which of the data sources in Sáhka do you think is most important (choose three)?

Figure 5.12 shows the players' most important data sources. Unlike the coaches, where the votes were relatively spread out, the players almost unanimously consider ZXY, strength training, and videos the most important data sources.

Question eight will evaluate how important the players think a system like Sáhka will be for their personal development as football players. This is important because if they do, they will want to use the system for themselves and their growth as a football player, not just because their coaches want them to use the system. Lastly, the final question evaluates how likely the players are to use Sáhka if available.

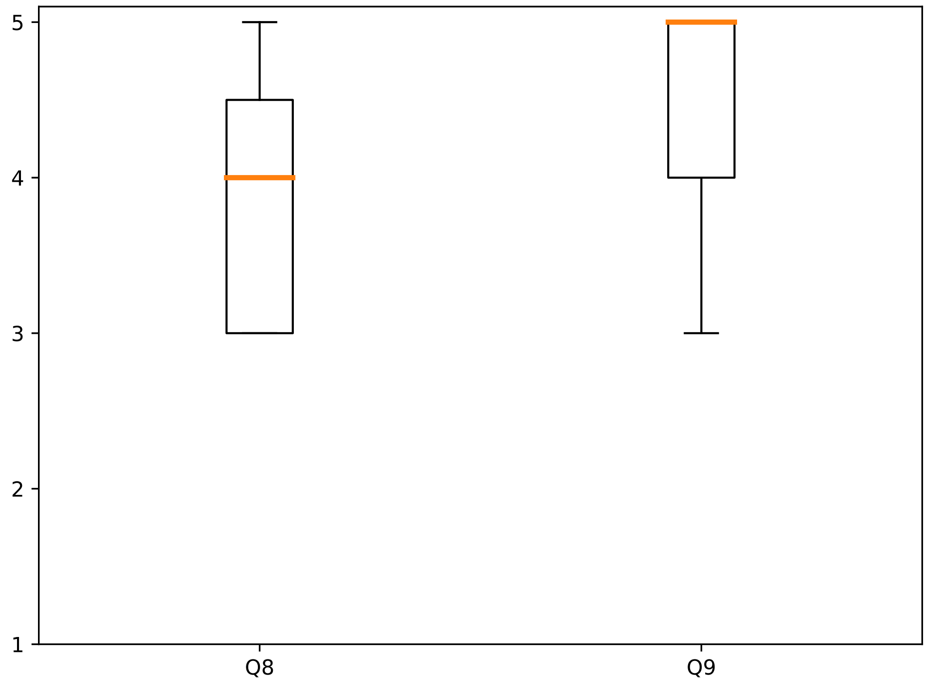

Figure 5.13: **Q8:** If you have a system available that collects your personal data (Sáhka), how valuable do you think such a system will be for your personal development as a football player?
Q9: How likely is it that you will use the system we have developed (Sáhka) if available?

The results from Figure 5.13 show that the players would find a system that collects and visualizes their personal data as valuable to their personal development as football players, and they would be likely to use such a system.

5.4 Summary

This chapter has presented some experiments that evaluate Sáhka's usability and cost of operation. The first experiment compared the storage cost of using a different method for storing videos from third-party systems against the current method. The second experiment evaluated the latency of fetching data from the local database compared to fetching it from third-party systems. Finally, some of TIL's players and educated coaches answered a user survey that evaluated Sáhka's usability and user satisfaction.

6 Discussion

This chapter will discuss some of the relevant topics surrounding this thesis.

6.1 Third-Party Systems

Sections 2.4.1 and 2.4.4 discussed some implications of relying on third-party systems. Sections 4.1.5 and 5.1 elaborated further on this by comparing the storage costs when storing the videos from Eliteserien Highlights against the availability issues when storing the URLs. However, when deciding whether to use a third-party system, the system's trustworthiness regarding user data should also be considered.

The third-party systems that are relevant for Sáhka deal primarily with user data in some shape or form. This raises an issue of who has control over that data and whether the third-party system ensures their users' privacy. For systems like ZXY, this is not an issue because the system is installed locally, which means the data is also stored locally. For TIL, this means that their ZXY data is stored at Alfheim, and they have complete control of the data and can ensure their users' privacy. On the other hand, for systems that only deal with public data, like Eliteserien Highlights, privacy is not an issue that needs to be considered.

6.2 Security

Security is a critical factor for systems like Sáhka due to the highly sensitive nature of the data in circulation. This section will discuss some of the security issues and solutions relevant to Sáhka.

6.2.1 Authentication

Authentication in Sáhka is handled using email and password. Currently, the email is not used for anything other than authentication; however, it can be used to, for example, notify users, recover accounts, or reset passwords. Before doing any of this, though, the emails should be verified. Verifying an email ensures that the user has access to the email and can use it to, for example, reset their password if they forget it.

Although passwords can, in theory, be secure enough for authentication, they are, in practice, quite vulnerable for one underlying reason: people are notoriously bad at remembering passwords. Because people are bad at remembering passwords, they often choose ones that are simple to remember, such as "12345678," "password," and "qwerty." Unfortunately, these passwords are also extremely easy to crack using, for example, a dictionary attack, which tries all the most common passwords until it finds the correct one. Furthermore, since people often reuse the same password, an attacker only needs to crack their password once to access all their accounts.

Consequently, two-factor authentication (2FA) is often employed to increase security. 2FA is an extra layer of security used during the authentication process to ensure users are who they say they are. One of the most common methods of 2FA, though not a very secure method, is text messages. This method ensures that the user trying to authenticate has access to the phone linked to the account by sending a unique code via text message, which the user must enter to authenticate. There are many other forms of 2FA, some of which are more secure than others; however, the more secure a method is, the more effort it usually requires to use.

6.2.2 Access Control

Access control is used to regulate who can access what in a system. This ensures that unauthorized parties cannot access confidential information, such as other users' data, or perform actions they are not permitted to, for example, purchasing something from another user's account.

In Sáhka, the first step of access control is authentication. Without authenticating, none of Sáhka's data is accessible. After authenticating, the system can determine what data the user has access to; players only have access to their own data, and coaches have access to the data of the players on their team.

This type of access control ensures that data is only accessible by authenticated users with the correct privileges. However, it does not guarantee anything about the actual person that is authenticated. For example, if an attacker gains access to a user's computer, which is already authenticated, they will circumvent the access control and have access to the same data as the user.

6.2.3 System Attacks

In addition to targeting users and their data, attackers can also target a system in several ways. One of the most common attacks against a system is denial-of-service (DOS) attacks, which flood the system with traffic to make it unavailable to other users. Actions can be taken to try and block these types of attacks; however, it adds a layer of complexity to the system. Hence, there are also third-party services available today that can be used to prevent these types of attacks.

Another similar type of attack is the use of automated bots to create spam in a system. For example, a bot that continuously creates spam accounts. This is a common problem for sites such as YouTube[1], where bots flood videos' comment sections with spam. A popular method of stopping bots is the use of CAPTCHA tests. These tests ask users to identify, for example, distorted letters or items in images, trivial tasks for humans but difficult for bots to complete.

6.3 Fault Tolerance

Fault tolerance is a system's ability to maintain proper operation in the event of one or more failures within some of its components. This is achieved by expecting likely failures and designing the system to be able to cope with them. Consequently, the more fault tolerant a system should be, the more complex it needs to be to handle more types of failures. This complexity is worthwhile for critical systems such as airplanes or hospital equipment because it is crucial that those systems remain operational. For other systems, where it is not catastrophic if it is unavailable for a period, too much fault tolerance

1. `https://www.youtube.com/`

may not be cost-effective.

Since Sáhka is hosted on a single server, meaning the server is a single point of failure, a server crash is the most likely failure to occur. If the server only goes down for a period, Sáhka would only be unavailable for that period. Worse would be if the server disk crashes and the data becomes unrecoverable. This would result in a loss of Sáhka's database and all the user data stored there. Currently, this issue is prevented by backing up the database periodically to another server. This means that only the data stored between two backups can be lost. The system is currently set to backup once a day, which means that, in the worst case, one day's worth of data is lost.

A better solution would be to utilize MongoDB's built-in replication functionality. MongoDB would then keep each replica up to date, meaning no data would be lost so long as one replica remains. Moreover, since Sáhka's front-end and back-end can be trivially replicated, Sáhka could be entirely distributed across the servers hosting a replica. This would let Sáhka be available in the event of server failures as long as one of the servers remain alive.

6.4 Privacy

Ensuring privacy is essential for systems that deal with personal data. Hence, the European Union created GDPR, a set of principles that dictate how personal data can be stored and processed, and what measures must be taken to ensure the subjects' privacy. This section will discuss what these principles are and how they relate to Sáhka.

Lawfulness, fairness, and transparency
This principle states that all personal data must be processed lawfully, fairly, and transparently in relation to the data subject. This means that users must be informed of what data is being collected on them and its use, and they must consent to this. For Sáhka, all data collected is used to present the coaches with as much relevant information as possible. Additionally, before TIL started using Sáhka, each player signed a consent form, allowing their data to be used.

Purpose limitation
This principle states that data should only be collected for specified, explicit, and legitimate purposes and should not be processed in a way that is incompatible with those purposes. Sáhka only collects and processes data for one purpose: to visualize it for the users.

Data minimization

This principle states that only data that is relevant for the specified purposes can be collected. For Sáhka, this means that only data that is relevant to the information coaches want can be collected. Hence, Sáhka cannot collect, for example, banking information or tracking data outside the pitch.

Accuracy
This principle states that all data must be accurate and either updated or deleted when inaccurate. As mentioned in Section 2.4.1, data collected from third-party systems are not updated. Consequently, if users edit their data in the third-party systems, the change will not be reflected in Sáhka. This needs to be rectified in the future to be compliant with this principle.

Storage limitation
This principle states that personal data should only be identifiable to its subject as long as necessary for its processing. Since the purpose of Sáhka is to present the data to the users, they must know to whom the data belongs. Hence, the data is identifiable as long as the subject has an account in Sáhka.

Integrity and confidentiality
This principle states that all personal data must be processed in a way that ensures security against unauthorized or unlawful processing and accidental loss, destruction, or damage. How Sáhka deals with security and loss of data was discussed in Sections 6.2 and 6.3.

Accountability
This final principle states that the controller of the personal data must be able to demonstrate compliance with the aforementioned principles. Consequently, Sáhka must be able to prove that it is compliant with these principles.

6.5 Usability

Section 5.3 presented a user evaluation conducted on coaches and players at TIL. The first set of questions confirmed that both the coaches and players were not content with the current set of tools available. The main issue was that they had to use multiple tools to collect and view data, and the players did not even have access to most of their data. Instead, they wanted one system that could collect and visualize all different types of data into one cohesive platform.

The second set of questions evaluated how Sáhka managed to fill this role of a single system that collects and visualizes disparate types of data. The coaches answered that Sáhka visualizes data better than the current tools, which is a bonus to having the data collected in one system. Additionally, both coaches and players agreed that Sáhka would be an excellent tool for facilitating individual follow-up of each player through, for example, regular coach-player meetings, which they all agreed is extremely important.

Lastly, the coaches and players were asked whether they would use Sáhka if it were available. The response here was positive, especially among the coaches. This corresponds well with feedback received at Alfheim, where the coaches have exclaimed excitement at using a novel system like Sáhka.

6.6 Summary

This chapter has discussed some of the relevant topics surrounding this thesis, including; some implications of relying on third-party applications; security concerns in systems that deal with sensitive data; why fault tolerance is essential, and ways to achieve it; how Sáhka deals with privacy with regards to GDPR; and the usability of Sáhka.

7 Conclusion

The problem definition of this thesis stated the following:

> *This thesis will investigate the design and implementation of a system that collects and analyses multimodal data about football players from several different sources to aid coaches in providing intervention of training load and practices personalized to individual athletes. Particular focus will be on developing a proof-of-concept prototype that provides coaches with as much relevant information as possible in an easily consumed format.*

Based on this problem definition, this thesis has Sáhka, a system that collects and analyses multimodal data on football players and presents the information visually to their coaches. Sáhka is currently integrated with three third-party systems: ZXY, PmSys, and Eliteserien Highlights. Additionally, Sáhka collects data directly from the users, including DXA, strength training, physical tests, nutrition, videos, objectives, and a journal. Feedback from coaches using the system has heavily influenced how and what data is visualized. Lastly, Sáhka uses algorithms to analyze the data collected and notify coaches if specific events occur, for example, if a player is not sleeping enough.

This thesis has been developed in close collaboration with TIL, who have been using Sáhka throughout its development, and will continue using it henceforth. Their educated coaches have provided valuable feedback, which has helped tailor Sáhka to their specific needs.

In the future, several contributions will be made based on the work performed in this thesis. Two theses will focus on integrating and simplifying different video solutions with Sáhka. Additionally, several theses will use Sáhka as a platform for training different Machine Learning algorithms.

7.1 Future work

7.1.1 Data Sources

One clear objective for future work is the addition of more data sources, both from third-party systems and directly from the users. Collecting more data will give a broader picture of the players' performance and health, letting coaches make more informed decisions. Additionally, it would be worthwhile to integrate multiple data sources for the same type of data. This has already been done for videos, which are collected directly from the users and Eliteserien Highlights. Supporting multiple ways of collecting data will allow users to choose their preferred method, or use multiple methods to collect even more data.

7.1.2 Machine Learning

The current analysis technique is relatively simplistic, combining conditional operations in DAGs to detect certain events. One downside to this approach is that events can only be detected after they occur. It would be better if the system could predict events before they happen, such as predicting a high risk of injury, so steps can be taken to avoid them. This is where more complex Machine Learning algorithms, especially Neural Networks, are more suitable. For example, using Long Short-Term Memory (LSTM) [54] on PmSys data has been shown to be able to predict future peeks in a football player's readiness to train [55]. Hence, exploring different algorithms on Sáhka's wide range of data could provide valuable models for predicting future events.

7.1.3 Native Application

As mentioned in Section 2.4.3, Sáhka was developed as a web application instead of a native or hybrid application. This would enable Sáhka to be available on any platform with just one implementation. However, with nearly every person having a smartphone these days, more and more systems are being developed as mobile apps to take advantage of features like push notifications and offline mode. This is something Sáhka could benefit from as well, letting coaches send notifications to players and accessing data without an internet connection.

Bibliography

[1] Markus Hilbertsen. *PVAS: Player Visual Analysis System*. Dec. 2021.

[2] D. E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, Paul R. Young, and Peter J. Denning. “Computing as a Discipline.” In: *Commun. ACM* 32.1 (Jan. 1989), pp. 9–23. ISSN: 0001-0782. DOI: 10.1145/63238.63239. URL: https://doi.org/10.1145/63238.63239.

[3] Robbert van Renesse, Aage Kvalnes, Dag Johansen, and Audun Arnesen. *Vortex. An event-driven multiprocessor operating system supporting performance isolation*. 2003.

[4] ÅGE KVALNES, D Johansen, ROBBERT VAN RENESSE, FRED B SCHNEIDER, and STEFFEN VIKEN VALVÅG. “Scheduler control over all resource consumption.” In: *Submitted to a journal* (2013).

[5] Robert Pettersen, Steffen Viken Valvåg, Åge Kvalnes, and Dag Johansen. “Jovaku: Globally Distributed Caching for Cloud Database Services Using DNS.” In: *2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*. 2014, pp. 127–135. DOI: 10.1109/MobileCloud.2014.20.

[6] Audun Nordal, Åge Kvalnes, Joseph Hurley, and Dag Johansen. “Balava: Federating Private and Public Clouds.” In: *2011 IEEE World Congress on Services*. 2011, pp. 569–577. DOI: 10.1109/SERVICES.2011.21.

[7] Anders Tungeland Gjerdrum. “Diggi: A Distributed Serverless Runtime for Developing Trusted Cloud Services.” MA thesis. 2020.

[8] Anders Tungeland Gjerdrum, Håvard Dagenborg Johansen, Lars Brenna, and Dag Johansen. “Diggi: A Secure Framework for Hosting Native Cloud Functions with Minimal Trust.” In: *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. 2019, pp. 18–27. DOI: 10.1109/TPS-ISA48467.2019.00012.

[9] Robert Pettersen, Håvard Dagenborg Johansen, and Dag Johansen. “Secure Edge Computing with ARM TrustZone.” In: *IoTBDS*. 2017.

[10] Anders T. Gjerdrum, Robert Pettersen, Håvard D. Johansen, and Dag Johansen. “Performance Principles for Trusted Computing with Intel SGX.” In: *Cloud Computing and Service Science*. Ed. by Donald Ferguson, Vìctor Mèndez Muñoz, Jorge Cardoso, Markus Helfert, and Claus Pahl.

Cham: Springer International Publishing, 2018, pp. 1–18. ISBN: 978-3-319-94959-8.

[11] Eleanor Birrell, Anders Gjerdrum, Robbert van Renesse, Håvard Johansen, Dag Johansen, and Fred B. Schneider. "SGX Enforcement of Use-Based Privacy." In: *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*. WPES'18. Toronto, Canada: Association for Computing Machinery, 2018, pp. 155–167. ISBN: 9781450359894. DOI: 10.1145/3267323.3268954. URL: https://doi.org/10.1145/3267323.3268954.

[12] Dag Johansen, Magnus Stenhaug, Roger B. A. Hansen, Agnar Christensen, and Per-Mathias Høgmo. "Muithu: Smaller footprint, potentially larger imprint." In: *Seventh International Conference on Digital Information Management (ICDIM 2012)*. 2012, pp. 205–214. DOI: 10.1109/ICDIM.2012.6360105.

[13] Magnus Stenhaug, Yang Yang, Cathal Gurrin, and Dag Johansen. "Muithu: A Touch-Based Annotation Interface for Activity Logging in the Norwegian Premier League." In: *Proceedings of the 20th Anniversary International Conference on MultiMedia Modeling - Volume 8326*. MMM 2014. Dublin, Ireland: Springer-Verlag, 2014, pp. 365–368. ISBN: 9783319041162. DOI: 10.1007/978-3-319-04117-9_37. URL: https://doi.org/10.1007/978-3-319-04117-9_37.

[14] Håkon Kvale Stensland, Vamsidhar Reddy Gaddam, Marius Tennøe, Espen Helgedagsrud, Mikkel Næss, Henrik Kjus Alstad, Asgeir Mortensen, Ragnar Langseth, Sigurd Ljødal, Østein Landsverk, Carsten Griwodz, Pål Halvorsen, Magnus Stenhaug, and Dag Johansen. "Bagadus: An Integrated Real-Time System for Soccer Analytics." In: *ACM Trans. Multimedia Comput. Commun. Appl.* 10.1s (Jan. 2014). ISSN: 1551-6857. DOI: 10.1145/2541011. URL: https://doi.org/10.1145/2541011.

[15] Pål Halvorsen, Simen Sægrov, Asgeir Mortensen, David K. C. Kristensen, Alexander Eichhorn, Magnus Stenhaug, Stian Dahl, Håkon Kvale Stensland, Vamsidhar Reddy Gaddam, Carsten Griwodz, and Dag Johansen. "Bagadus: An Integrated System for Arena Sports Analytics: A Soccer Case Study." In: *Proceedings of the 4th ACM Multimedia Systems Conference*. MMSys '13. Oslo, Norway: Association for Computing Machinery, 2013, pp. 48–59. ISBN: 9781450318945. DOI: 10.1145/2483977.2483982. URL: https://doi.org/10.1145/2483977.2483982.

[16] Dag Johansen, Håvard Johansen, Tjalve Aarflot, Joseph Hurley, Åge Kvalnes, Cathal Gurrin, Sorin Zav, Bjørn Olstad, Erik Aaberg, Tore Endestad, Haakon Riiser, Carsten Griwidz, and Pål Halvorsen. "DAVVI: A Prototype for the next Generation Multimedia Entertainment Platform." In: *Proceedings of the 17th ACM International Conference on Multimedia*. MM '09. Beijing, China: Association for Computing Machinery, 2009, pp. 989–990. ISBN: 9781605586083. DOI: 10.1145/1631272.1631482. URL: https://doi.org/10.1145/1631272.1631482.

[17] Dag Johansen, Pål Halvorsen, Håvard Johansen, Håkon Riiser, Cathal Gurrin, Bjørn Olstad, Carsten Griwodz, Aage Kvalnes, Joseph Hurley, and Tomas Kupka. "Search-based composition, streaming and playback of video archive content." In: *Multimedia Tools and Applications - MTA* 61 (Nov. 2012), pp. 1–27. DOI: 10.1007/s11042-011-0847-5.
[18] Kim Hartvedt Andreassen. "Metrix: Real-time Analysis of Physical Performance Parameters in Elite Soccer." MA thesis. 2018.
[19] Ivan Baptista. "Football training specificity - Training individualization within the collective periodization." PhD Dissertation. The Arctic University of Norway, 2020.
[20] Sigurd Pedersen. "Training and influence of maximal strength in football players - With specific emphasis on females." PhD Dissertation. The Arctic University of Norway, 2021.
[21] Carl Foster, Jose Rodriguez-Marroyo, and Jos de Koning. "Monitoring Training Loads: The Past, the Present, and the Future." In: *International journal of sports physiology and performance* 12 (May 2017), pp. 2–8.
[22] Pitre Bourdon, Marco Cardinale, Andrew Murray, Paul Gastin, Michael Kellmann, Matthew Varley, Tim Gabbett, Aaron Coutts, Darren Burgess, Warren Gregson, and Nigel Cable. "Monitoring Athlete Training Loads: Consensus Statement." In: *International Journal of Sports Physiology and Performance* 12 (May 2017), p. 161. DOI: 10.1123/IJSPP.2017-0208.
[23] Paolo Gaudino, F. Iaia, Giampietro Alberti, Richard Hawkins, A Strudwick, and Warren Gregson. "Systematic Bias between Running Speed and Metabolic Power Data in Elite Soccer Players: Influence of Drill Type." In: *International journal of sports medicine* 35 (Oct. 2013). DOI: 10.1055/s-0033-1355418.
[24] Shona L. Halson. "Monitoring Training Load to Understand Fatigue in Athletes." In: *Sports Medicine (Auckland, N.z.)* 44 (2014), pp. 139–147.
[25] Liam C Hennessy and Ian Jeffreys. "The Current Use of GPS, Its Potential, and Limitations in Soccer." In: *Strength & Conditioning Journal* (2018).
[26] Håvard Johansen, Svein Pettersen, Pål Halvorsen, and Dag Johansen. "Combining Video and Player Telemetry for Evidence-Based Decisions in Soccer." In: *icSPORTS 2013 - Proceedings of the International Congress on Sports Science Research and Technology Support* (Sept. 2013).
[27] Wouter G.P. Frencken, Koen A.P.M. Lemmink, and Nico J. Delleman. "Soccer-specific accuracy and validity of the local position measurement (LPM) system." In: *Journal of Science and Medicine in Sport* 13.6 (2010), pp. 641–645. ISSN: 1440-2440. DOI: https://doi.org/10.1016/j.jsams.2010.04.003. URL: https://www.sciencedirect.com/science/article/pii/S1440244010001131.
[28] Svein A. Pettersen, Håvard D. Johansen, Ivan A. M. Baptista, Pål Halvorsen, and Dag Johansen. "Quantified Soccer Using Positional Data: A Case Study." In: *Frontiers in Physiology* 9 (2018). ISSN: 1664-042X. DOI:

10.3389/fphys.2018.00866. URL: https://www.frontiersin.org/article/10.3389/fphys.2018.00866.

[29] Philipp Seidenschwarz, Adalsteinn Jonsson, Fabian Rauschenbach, Martin Rumo, Lukas Probst, and Heiko Schuldt. "Combining Qualitative and Quantitative Analysis in Football with SportSense." In: *Proceedings Proceedings of the 2nd International Workshop on Multimedia Content Analysis in Sports*. MMSports '19. Nice, France: Association for Computing Machinery, 2019, pp. 34–41. ISBN: 9781450369114. DOI: 10.1145/3347318.3355519. URL: https://doi.org/10.1145/3347318.3355519.

[30] Kennet Vuong. "PmSys: a monitoring system for sports athlete load, wellness & injury monitoring." MA thesis. 2015.

[31] Håvard D. Johansen, Dag Johansen, Tomas Kupka, Michael Riegler, and Pål Halvorsen. "Scalable Infrastructure for Efficient Real-Time Sports Analytics." In: *Companion Publication of the 2020 International Conference on Multimodal Interaction* (2020). Ed. by Khiet Truong, Dirk Heylen, Mary Czerwinski, Nadia Berthouze, Mohamed Chetouani, and Mikio Nakano. DOI: 10.1145/339503510.1145/3395035.3425300. URL: https://dl.acm.org/doi/proceedings/10.1145/3395035https://dl.acm.org/doi/10.1145/3395035.3425300https://dl.acm.org/doi/pdf/10.1145/3395035.3425300.

[32] Chiara Milanese, Valentina Cavedon, Giuliano Corradini, Francesco De Vita, and Carlo Zancanaro. "Seasonal DXA-measured body composition changes in professional male soccer players." In: *Journal of Sports Sciences* 33.12 (2015). PMID: 25773172, pp. 1219–1228. DOI: 10.1080/02640414.2015.1022573. eprint: https://doi.org/10.1080/02640414.2015.1022573. URL: https://doi.org/10.1080/02640414.2015.1022573.

[33] Jiri Dvorak and Astrid Junge. "Football Injuries and Physical Symptoms." In: *The American Journal of Sports Medicine* 28.5_suppl (2000), pp. 3–9. DOI: 10.1177/28.suppl_5.s-3. eprint: https://doi.org/10.1177/28.suppl_5.s-3. URL: https://doi.org/10.1177/28.suppl_5.s-3.

[34] M Svensson and B Drust. "Testing soccer players." In: *Journal of Sports Sciences* 23.6 (2005). PMID: 16195009, pp. 601–618. DOI: 10.1080/02640410400021294. eprint: https://doi.org/10.1080/02640410400021294. URL: https://doi.org/10.1080/02640410400021294.

[35] Martin Buchheit. *The 30-15 Intermittent Fitness Test: 10 year review*.

[36] *Taste of the Liverpool Way*. English. Hardcover. ISBN: 978-1911613961.

[37] D. Bonnici, M. Greig, Ibrahim Akubat, Andy Sparks, David Bentley, and Lars Mcnaughton. "Nutrition in Soccer: A Brief Review of the Issues and Solutions." In: *Journal of Science in Sport and Exercise* 1 (June 2019). DOI: 10.1007/s42978-019-0014-7.

[38] K. Thulasiraman and M.N.S. Swamy. *Graphs: Theory and Algorithms*. A Wiley interscience publication. Wiley, 1992. ISBN: 9780471513568. URL: https://books.google.no/books?id=4kHN6uSinQoC.

[39] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. "TensorFlow: A System for Large-Scale Machine Learning." In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*. OSDI'16. Savannah, GA, USA: USENIX Association, 2016, pp. 265–283. ISBN: 9781931971331.

[40] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. "Ray: A Distributed Framework for Emerging AI Applications." In: *Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation*. OSDI'18. Carlsbad, CA, USA: USENIX Association, 2018, pp. 561–577. ISBN: 9781931971478.

[41] Antonio Carzaniga, David S. Rosenblum, and Alexander L. Wolf. "Design and Evaluation of a Wide-Area Event Notification Service." In: *ACM Trans. Comput. Syst.* 19.3 (Aug. 2001), pp. 332–383. ISSN: 0734-2071. DOI: 10.1145/380749.380767. URL: https://doi.org/10.1145/380749.380767.

[42] JL Carlson. *Redis in Action. Media. johnwiley. com. au*. 2013.

[43] Avinash Lakshman and Prashant Malik. "Cassandra: A Decentralized Structured Storage System." In: *SIGOPS Oper. Syst. Rev.* 44.2 (Apr. 2010), pp. 35–40. ISSN: 0163-5980. DOI: 10.1145/1773912.1773922. URL: https://doi.org/10.1145/1773912.1773922.

[44] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. "Dynamo: Amazon's Highly Available Key-Value Store." In: *Proceedings of Twenty-First ACM SIGOPS Symposium on Operating Systems Principles*. SOSP '07. Stevenson, Washington, USA: Association for Computing Machinery, 2007, pp. 205–220. ISBN: 9781595935915. DOI: 10.1145/1294261.1294281. URL: https://doi.org/10.1145/1294261.1294281.

[45] Ian Sommerville. *Software Engineering*. Pearson, 2011. ISBN: 9780137035151.

[46] *REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. Official Journal of the European Union (OJ) L 119/1, 4.5.2016, Article 5. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN#d1e1807-1-1.

[47] Nichol JR, Sundjaja JH, and Nelson G. "Medical History." In: *StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing* (Jan. 2022). URL: https://www.ncbi.nlm.nih.gov/books/NBK534249/.

[48] E. Nygren and P. Henriksson. "Reading the medical record. I. Analysis of physician's ways of reading the medical record." In: *Computer Methods and Programs in Biomedicine* 39.1 (1992), pp. 1–12. ISSN: 0169-2607. DOI: https://doi.org/10.1016/0169-2607(92)90053-A.

[49] Clement J. McDonald, J.Marc Overhage, William M. Tierney, Paul R. Dexter, Douglas K. Martin, Jeffrey G. Suico, Atif Zafar, Gunther Schadow, Lonnie Blevins, Tull Glazener, Jim Meeks-Johnson, Larry Lemmon, Jill Warvel, Brian Porterfield, Jeff Warvel, Pat Cassidy, Don Lindbergh, Anne Belsito, Mark Tucker, Bruce Williams, and Cheryl Wodniak. "The Regenstrief Medical Record System: a quarter century experience." In: *International Journal of Medical Informatics* 54.3 (1999), pp. 225–253. ISSN: 1386-5056. DOI: https://doi.org/10.1016/S1386-5056(99)00009-X.

[50] Samiran Nundy, Atul Kakar, and Zulfiqar A. Bhutta. "Medical Records." In: *How to Practice Academic Medicine and Publish from Developing Countries? A Practical Guide*. Singapore: Springer Singapore, 2022, pp. 429–440. ISBN: 978-981-16-5248-6. DOI: 10.1007/978-981-16-5248-6_45. URL: https://doi.org/10.1007/978-981-16-5248-6_45.

[51] Lars Brenna, Johannes Gehrke, Dag Johansen, and Mingsheng Hong. *Distributed Event Stream Processing with Non-deterministic Finite Automata*. 2009.

[52] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, and Dennis Fetterly. "Dryad: Distributed Data-parallel Programs from Sequential Building Blocks." In: *Proceedings of the 2007 Eurosys Conference*. Association for Computing Machinery, Inc., Mar. 2007. URL: https://www.microsoft.com/en-us/research/publication/dryad-distributed-data-parallel-programs-from-sequential-building-blocks/.

[53] R. Likert. *A Technique for the Measurement of Attitudes*. A Technique for the Measurement of Attitudes nr. 136-165. Archives of Psychology, 1932. URL: https://books.google.no/books?id=9rotAAAAYAAJ.

[54] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory." In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[55] Theodor Wiik, Håvard D. Johansen, Svein-Arne Pettersen, Ivan Baptista, Tomas Kupka, Dag Johansen, Michael Riegler, and Pål Halvorsen. "Predicting Peek Readiness-to-Train of Soccer Players Using Long Short-Term Memory Recurrent Neural Networks." In: *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. 2019, pp. 1–6. DOI: 10.1109/CBMI.2019.8877406.