UNIVERSITY OF TROMSØ UIT

FACULTY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

Ruoksat

A system for capturing, persisting and presenting the digital footprint of soccer knowledge and expertise

Kim-Edgar Sørensen

INF-3981 Master's Thesis in Computer Science
June 2013

Abstract

There are currently over 27 000 registered soccer teams in Norway. However, even with mandatory coaching licenses, there are still a vast variety of approaches to how training sessions should be planned and executed. Even in top-level professional soccer organizations, the planning of exercises involves a considerable amount of manual labor. The most common approaches include using old-fashioned pen and paper, or at best organizational tools like worksheets to plan exercises based on memory, word-to-mouth or vague guidelines. The exercises are presented to the players using sheets of paper along with verbal descriptions from the team coaches.

The digital archiving possibilities are also limited, and knowledge related to the exercises, and exercise execution, is often not organized or persisted.

This thesis designs, develops and evaluates a system for capturing, persisting and presenting expert coach knowledge in the form of exercises. The system is designed and developed with a particular focus on user interaction and user experience. The process is performed in close collaboration with coaches, medical staff and players from the elite level Norwegian soccer club Tromsø IL.

The evaluation of the system is performed through a standard pairwise comparison between the implemented system and the solutions currently available to the assessors.

Acknowledgements

I would like to thank my supervisor Professor Dag Johansen for his great advice, ideas and input throughout this entire project. Your passion is truly inspiring, and your feedback has been a great motivating factor. This project would not have been possible without you.

Further I would like to thank Agnar Christensen, Vegard Berg Johansen and Bjœrn Vidar Stenersen from Tromsø IL for your time, insight, feedback and encouraging words.

I would also like to thank Dr. Cathal Gurrin and Yang Yang from Dublin City University for providing me with great insight related to the field of human-computer interaction. Additionally, Dr. Håvard Johansen, I thank you for helping me in the process of writing this thesis. And also I thank my colleagues at the iAD group, Robert, Magnus and Jan Ove, for their input.

I thank everyone that participated in the evaluation of this project, especially Ruben Yttergård Jenssen, William Frantzen and Magnus Andersen for testing the system during this thesis.

A special thank you goes to my good friend and future colleague, Roger Hansen. You have dedicated a lot of your time to helping me with this thesis, much more than anyone could have asked. Your feedback, insight and encouragements have been essential to the entire project. I cannot thank you enough, and I really look forwarding to working with you in the future.

Finally, I would like to thank my friends and family for their support and encouraging words. I would like to direct a special thank you to my friend Melissa; you have motivated and encouraged me in moments when I needed it.

Table of Contents

List of Acronyms

AJAX Asynchronous JavaScript and XML

PTW Pre-Training Wellness

RPE Rating of Perceived Exertion

SQL Structured Query Language

NoSQL Not Only SQL

MVVM Model View ViewModel

MVC Model-View-Controller

UX User Experience

UI User Interface

GUI Graphical User Interface

XAML Extensible Application Markup Language

COM Component Object Model

HTML HyperText Markup Language

ACID Atomicity, Consistency, Isolation, Durability

RDBMS Relational Database Management System

BLOB Binary Large Object

HCI Human-Computer Interaction

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

According to the most recent statistical report from NFF [9], there are currently over 27 000 registered soccer teams in Norway, with a total of 360 000 active players divided between them. Last year alone 30 000 coaches completed the courses for entry level coaching licenses [9]. However, even with mandatory coaching licenses, there are still a vast variety of approaches to how training sessions should be planned and executed.

Even in top-level professional soccer organizations, the planning of exercises involves a considerable amount of manual labor. The most common approaches include using old-fashioned pen and paper, or at best organizational tools like worksheets to plan exercises based on memory, word-to-mouth or vague guidelines. The exercises are presented to the players using sheets of paper along with verbal descriptions from the team coaches, even when communicating to large groups of athletes. This increases the risk for miscommunication, especially with regards to new players not familiar with the exercises.

The digital archiving possibilities are also limited, and knowledge related to the exercises, and exercise execution, is often not organized or persisted. Hence, if an important member of the team leaves the organization, his or her knowledge and expertise is no longer available. Storing exercises would also allow the soccer organizations to expose the youth academy teams to exercises from the first team curriculum. This could assist in transferring expert knowledge to the youth academy coaches, and thereby instilling the first teams style-of-play[1], in the younger teams in the organization.

1.2 Problem Definition

This thesis shall design and develop a system for digitally capturing, persisting and presenting data input related to exercise execution within the sports domain. Main focus will be on Usability, user experience shall be emphasized, and design and development will be done in collaboration with end-users. An important objective will be providing an infrastructure for uploading and presenting expert coach knowledge and expertise. Particular focus will be on providing interfaces for athlete monitoring purposes. Examples include the ability for players to provide timely and accurate wellness parameters prior to training sessions, and subjective perception of exertion after training sessions. Evaluation of the system will be performed through end-user feedback.

1.3 Interpretation

Our thesis is that providing an infrastructure for capturing and presenting expert knowledge in the form of exercises, and athlete status reports, will improve planning and execution of exercises in the domain of professional soccer. We conjecture that utilizing digital tools exposed by user-friendly interfaces for exercise planning and execution will simplify and improve the planning process and presentation of soccer exercises. Additionally the expert knowledge will be retained even if key members of the organization leave the team. We also investigate the added benefits from providing a system for collecting and displaying athlete wellness and exertion reports, and exposing these reports to the coaches carrying out the training sessions.

Focus will mainly be on the interaction between the users and the system, using tools and methods from the field of Human-Computer Interaction (HCI), and well documented best practices within the realm of User Experience (UX)-design. The system will include components for collecting, organizing and displaying athlete status reports, as well as exposing interfaces for uploading data related to exercises and creating exercise plans. Finally, the system will need to provide an interface for presenting exercises by using multi-media soccer data like text and video.

The design and development processes will be performed in close

[1] http://footballspeak.com/post/2013/02/14/Ajax-Youth-System.aspx

collaboration with staff and players from the elite Norwegian soccer club Tromsø IL. An end-user comparison of the system against currently available tools will perform a final evaluation, and the result of this evaluation will be used to conclude the thesis.

1.4 Methodology

The final report of the ACM Task Force on the Core of Computer Science [5] divides the discipline of computing into three major paradigms:

- *Theory*: Rooted in mathematics, the approach is to define problems, propose theorems and seek to prove them in order to determine new relationships and progress in computing.
- *Abstraction*: Rooted in the experimental scientific method, the approach is to investigate a phenomenon by stating hypothesis, constructing models and simulations, and analyzing the results.
- *Design*: Rooted in engineering, the approach is to state requirements and specifications, design and implement systems that solve the problem, and test the systems to systematically find the best solution to the given problem.

For this thesis the design process seems to be the most suitable out of the three paradigms. However, as this thesis has its basis in HCI theory, a more specific design process is used. [7] suggests the following process for effective design that responds to human problems:

- *Problem definition*: A need or problem is identified, researched, and defined.
- *Gathering and analyzing informatuion*: Questioning assumptions, learning what is not known.
- *Generating alternatives and building prototypes*: Drawing, interviewing, modeling or evaluating statistics for the design process.
- *Implementing choices*: Prototypes are implemented and used to project content, scope and intent.
- *Evaluation outcomes*: Assessment and testing of the aforementioned prototypes.
- *Production*: The release of a final prototype, this is then refined in order to achieve the desired result.

1.4.1 HCI methods

In [6], a similar five staged user-centered design process is presented. This maps the aforementioned steps into the realm of web design, and is the process used in this thesis.

1. Requirement Analysis
The requirement analysis consists of establishing the goals of the system, both in respect to the user but also the business behind the system. One should agree on the needs and requirements of the users, and attempt to use this as a guideline throughout all stages of development. In this step, one should also attempt to analyze other systems within the same domain and evaluate the current system in place, if it exists.

2. Conceptual proposal
In this step, an outline of the design and architecture on an abstract level should be created, along with identifying essential features.

3. Prototyping
The next stage is to create mockups or prototypes of the system or application, and evaluate its usability. The stage is then reiterated to improve the design based on the evaluation until the prototypes and/or mockups indicate that the goals and requirements set in the first step is met.

4. Development
This is where the actual system is implemented based on information gathered so far in the HCI process. After the initial implementation, functionality is evaluated through different testing schemes and quality assurance methods, and this evaluation is used to improve the end result.

5. Deployment and Maintenance
The last step involves launching the system for all intended users. Feedback from users is used to maintain the functionality of the site as well as a driving force to create new requirements, and starting work on a new major release.

1.5 Context

This thesis is written as a part of the Information Access Disruption (iAD) centre. The iAD centre researches fundamental structures and concepts for large-scale information access, and the main focus areas are sport technologies, cloud computing and analytics runtimes.

In a previous project, Davvi [16], the iAD centre has conducted research into the domain of video content delivery. The Davvi project is a system for search and recommendations in existing video delivery systems, with a focus on entertainment and social aspects.

Another iAD project currently in use by TIL, and strongly related to Muithu, is Bagadus [13]. Developed by the iAD department at the University of Oslo, Bagadus is a player tracking system that uses ZXY Sports Tracking[2] to track individual player statistics, combined with a video camera array.

Research into the domain of infrastructure and virtualization technologies within the iAD centre is currently being conducted. A new operating system kernel providing fine-grained accounting, and scheduling of system resources is being developed by Kvalnes et al [19]. Light-weight alternatives to virtual machines are also being researched by Nordal et al. [29].

Oivos [38] and Cogset [37] highlights the iAD group's involvement in research into high-performance distributed data processing.

For cloud integration and deployment the research group is building and evaluating systems for supporting resource elasticity within and between both private and public clouds [17][28].In order to support this, meta-code is used. Meta-code is pre programmed code modules tailored to fit specific user need, by extending and augmenting the cloud infrastructure (Hurley et al. [17]).

With codecaps, iAD suggests a novel approach to access control for cloud-like environments, by proposing use of certificate chains embedded with executable code fragments [39].

1.5.1 Muithu

This thesis is linked the Muithu system [18], which is a light-weight video and cellular phone based notational analysis system. The system is used during sporting events, both in matches and during practice, and is comprised by

[2] http://zxy.no

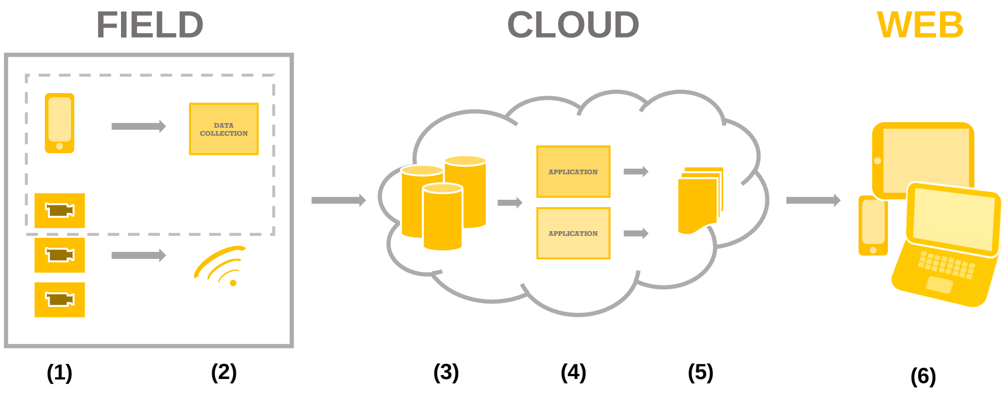


Figure 1.1: Muithu System

a multi-angle camera setup where the coaches can extract important events using a cell phone or a tablet (Figure 1.1). Muithu is currently in use by the elite level Norwegian soccer club Tromsø IL (TIL).

1.6 Outline

- *Chapter 2*: Presents some background information related to the thesis
- *Chapter 3*: Describes the requirement specification
- *Chapter 4*: Describes the general system model
- *Chapter 5*: Presents the implemented solution along with a set of visual prototypes
- *Chapter 6*: Discusses the evaluation of the system, including experiments
- *Chapter 7*: Concludes the thesis

Chapter 2

Background

In this chapter we first present some background related to Human-Computer Interaction (HCI) since this thesis has a focus on user-centered development, before presenting alternative storage solutions. We then present some background related to Windows Store apps, and then concluding the chapter by highlighting sports related topics relevant to this thesis.

2.1 HCI

Human-Computer Interaction (HCI) is a field within computer science and psychology that encapsulates the study, planning and design of interactions between the users and a computer. There are three main elements of HCI: the user, the computer and how they interact.

User

Within HCI, the user is not necessarily one single person using a computer. It is defined as whoever is performing a task using any technology [40]. This could be one individual user, or it could be a bigger group of people collaborating and/or communicating through the system. They key aspect of peoples interaction is the senses that relay information; this includes sight, hearing and touch. An example of how the sensory system should be taken into great consideration when designing a layout is the fact that even the slightest movement in the peripheral of their field of view distracts people. This in turn means only using bold and eye-catching elements in the critical path of the design flow, and keeping the outer, less important elements, clean and simplistic [7].

The Computer

In the current state of HCI, the computer is not necessarily the standard desktop computer; it could be a bigger computer system, an embedded device, clocks, sensors, or any other kind of interface.

The Interaction Between Them

With the obvious differences between humans and computers, HCI attempts to ensure interoperability. This involves analyzing the patterns of human behavior and sensory input to adapt the system to the user, and not the other way around. Ideally the user should not need to change its behavior to suit the system, even though this might be necessary in some scenarios. The user should be the main guide through the entire design process, and they should be consulted frequently through most of the development phase [40].

2.1.1 Goals

The goals of HCI are to create well functioning systems as well as increase usability. Hence the goal of the developer should be to understand the factors that determine how technology is used, develop tools to enable building such systems and achieve efficient and effective interaction. An important aspect of achieving this goal is that the interaction needs to be safe. The user should not be able to perform unsafe actions that compromise the system; this means that the interface should also be a barrier for erroneous actions. The recurring theme of HCI is the idea that the users should come first. Their preferences, needs and capabilities should determine the way developers design a system. Ideally the user should not need to change its behavior to suit the system, even though this might be necessary in some scenarios. All of the above is also applicable to web applications and web design [1].

2.1.2 Best Practice

This section discusses some fundamental principles and best practices for usable interface design.

Colors: Using color appropriately in interface design makes it easier for users to absorb information and differentiate information types and hierarchies. Colors are often used to guide users through information and highlight key data, and to avoid confusion a general rule is to use no more than three different colors. If more distinction is needed opacity can be used as a visual aid [1].

2.1.3 Page Design

Page design principles focus on readability and presentation of information in a comprehensible manner. [40] and [1] states the following elements of visual structuring for providing the users with a path and a guide through the page layout.

Chunking and Queuing: By structuring the visual field by breaking information into smaller, more manageable groups, and also queuing them in a hierarchical manner, you increase the effectiveness of the navigation by hiding elements not necessarily relevant to the users current context.

Mixing Modes: With a wide range of users, it is likely that they have different preferences in how to prefer to parse information. Some users prefer text based presentations, and others might want the information displayed as illustrations, diagrams, photos and such. This means that a good interface utilizes a combination of text and images when presenting information.

Abstracting: Making sub-elements of the layout adhere to the same principles as the layout as a whole results in a layout that is effective for the reader to parser, and also for the designer to create.

Using Grids and White-Spaces: Using white-spacing between layout elements is an extremely effective way of visually opening up a page, and also providing focus on the desired elements. It helps to combine white spaces with a grid layout, grouping together similar information, and separating the elements using white spaces. This provides the user with resting points within the visual field, and also creates the perception of simplicity and usability [7].

Field of View: Related to the grid, the field of view represents the area of the page visible by user with little or no eye movement [40]. It is important to place key elements of the layout within the users field of view, and not distract the user by placing flashing or superfluous elements in the peripheral vision. This would result in the user losing focus on the elements you wish to highlight.

2.1.4 HCI Summary

HCI is the study, planning and design of interaction between users and computers, with the goal of creating well functioning systems and increase usability. A recurring theme of HCI is that the user plays a central role, and should be considered in all stages of development.

Fundamental design principles include guidelines for page design, typography and color schemes.

Page design focuses on readability and presentation of information, guiding the users through the page layout using a clean visual structure. This includes breaking information into smaller chunks and queuing sortable content, using a recurring theme through all pages, and using a well defined grid layout, placing the content inside the users field of view.

Lastly, this section presented the appropriate color usage for highlighting key information, and that we should limit the number of main colors in the design. In order to streamline the design process, a color palette of compatible colors with a monochromatic color sub-scheme can be used.

2.2 Storage

2.2.1 Relational Databases and NoSQL

A traditional relational database is a collection of tables containing data items. The data schema and relational model must be defined in advance, and the columns and relations in the database are quite static. It allows for the use of foreign keys in order to create new relations through relating it to the primary key of another table. This relational model renders the possibility for complex queries through SQL, which is a special purpose programming language for handling data in Relational Database Management Systems (RDBMSs). ACID properties are guaranteed in traditional RDBMSs, which is manageable in centralized environments, but will result in performance loss when distributing the database across multiple storage nodes [34].

By relaxing the consistency requirements, we can achieve storage solutions that are more scalable than traditional relational databases. A more recent approach to data organization and storage is Not Only SQL (NoSQL). In NoSQL databases, the complexity of the data model is reduced, and no fixed, pre-defined schema is required, and each record can hold a varying number of attributes [31]. Replacing relations with key-value lookups, and exposing a simpler interface for accessing the data (i.e. PUT/GET), reduces the complexity of the data model. This combined with the approach of not adhering to the ACID-principles for transactions, and instead allowing for eventual consistency, makes NoSQL database systems ideal for storing large amounts of data, partitioned across multiple servers [35].

Within NoSQL databases there are several different categories of implementations, each suitable to different problem domains. Key-value databases are storages for providing a simple interface to access items (value) based on an identifier [8]. Document databases consists of one or more self-contained, semi-structured documents, where each document only contains information related to that document, and with no external relations. This structure makes document databases highly dynamic in nature [30]. For managing large amounts of structured data, Google designed and implemented a Big Table database, also known as tabular databases [4]. The database is essentially one huge table, with three dimensions, row index, column index, and a timestamp. In these tabular databases rows can hold different sets of attributes, and since they do not support relations they are able to hold millions of columns per row [4].

2.2.2 OpenStack Object Storage

OpenStack is an open cloud infrastructure platform that controls large resource pools for different aspects within datacenters, including compute, storage and network resources [10]. The storage module of OpenStack provides a cost-effective object storage for scaling out storage for use in applications or for backup, data retention and archiving [10]. It is a distributed storage system for storing static data, like videos or images, with no centralized master node for handling logic. Objects are written to multiple nodes distributed in the data center, and all logic is handled by software, and thus allowing the use of commodity hardware in the storage cluster.

Organization and access to data in the OpenStack Object Storage is done through objects, containers and account servers, with a proxy node being the interface exposed. An account server contains a list of all available containers within the account, and each container a list of objects. The objects reside in the leaf node of this hierarchical structure, and are stored as Binary Large Objects (BLOBs) by an Object Server [10].

2.2.3 MongoDB

Being one of the most widespread NoSQL databases, MongoDB provides a general purpose, open-source database with a document data model [30]. It features support for aggregation frameworks and Map-Reduce jobs, and a rich support for rich queries. The schemas can vary from document to document within the database system, while still providing functionality similar to that of relational databases, like a full query language and strong consistency.

In MongoDB, structured data is stored in JSON-like documents using dynamic schemas called BSON. In this organizational pattern, each data element is a *document*, and is stored in a collection of one to several documents. Comparable to RDBMSs, collections are like tables and documents are like records. However, in MongoDB, every document in a collection can have a completely different set of attributes compared to other documents in that same collection [30].

2.3 Windows Store Apps

Recently Microsoft launched the latest version of Microsoft Windows, Windows 8. With Windows 8, Microsoft introduced a new style of design language for applications with a touch-centric design, and a focus directed on improving User Experience (UX) on mobile, touch-based devices [11]. Applications built using this design language were originally called Metro-style apps, but are now referred to as *Windows Store apps*. These apps can be run on both desktop version of Windows 8 and Windows RT, which is a version of the operating system created for tablets with ARM processors [21].

2.3.1 Windows Runtime

A major component of the underlying platform for building Windows Store apps is the application architecture Windows Runtime or Win RT (Figure 2.1), not to be confused with the aforementioned Windows RT operating system. [11]. Win RT is a COM-based API comprised by a set of objects, rendering a level of abstraction on top of the underlying .NET and Win32 APIs. These objects are built directly on top of the Windows kernel services, and thereby improve interoperability by providing objects for handling system resources [21].

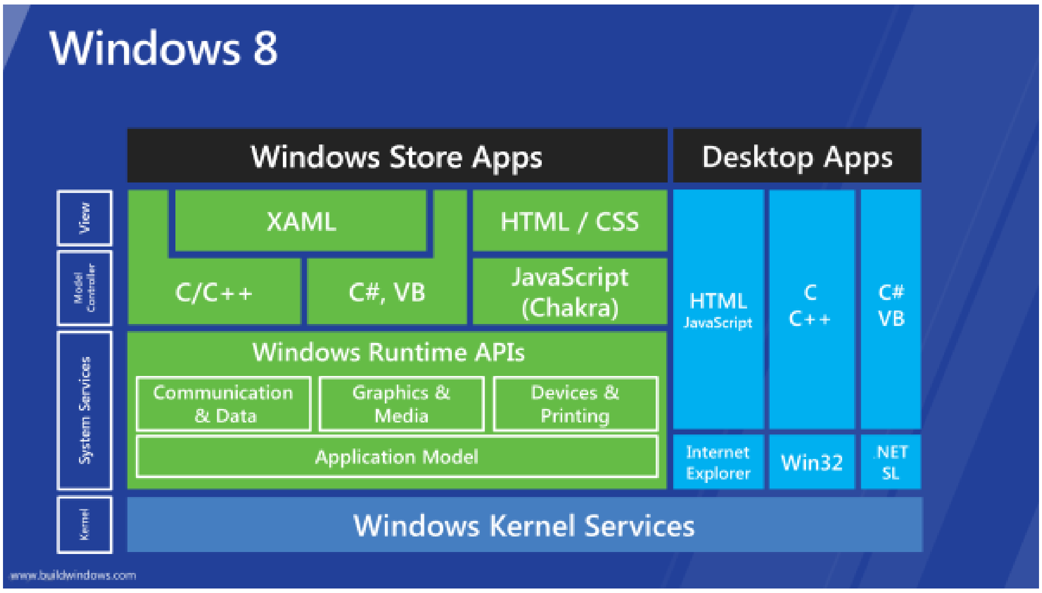


Figure 2.1: Windows Runtime architecture [20](modified)

As a result of this, Windows Runtime supports a variety of programming and markup languages, as Visual Basic, C++ or C# with XAML, or JavaScript/TypeScript alongside HTML5. Apps utilizing the Win RT application architecture also run in a sandboxed environment, where the app needs to be explicitly granted access to system resources, thus improving security and stability [2].

2.3.2 User Experience Guidelines

In order to assist in the process of layout design, Microsoft has provided developers with a set of guidelines for UX-related elements of Windows Store apps [26]. The developer is advised to use fluid layouts, which adapts a page to the current orientation and size of the device it is displayed on. The guidelines also state that the app should utilize controls designed for fluid layouts, and consider a vector-based User Interface (UI) by using Extensible Application Markup Language (XAML). Following these guidelines will also improve the usability of the application, by rendering a familiar, Windows 8-styled user interface to the users.

2.3.2.1 Page Design

The UX-guideline documentation describes a layout pattern for the page placement of UI elements. This pattern contains rules for margins, padding, header placement, and content organization. One important element of the layout pattern is the grid system used in Windows 8. A grid made of units and sub-units of square collections of pixels, allowing for uniform element alignment in all Windows Store apps. Each main grid unit consists of 20x20 pixels, and is divided into 16 sub-units of 5x5 pixels, as shown in Figure 2.2.

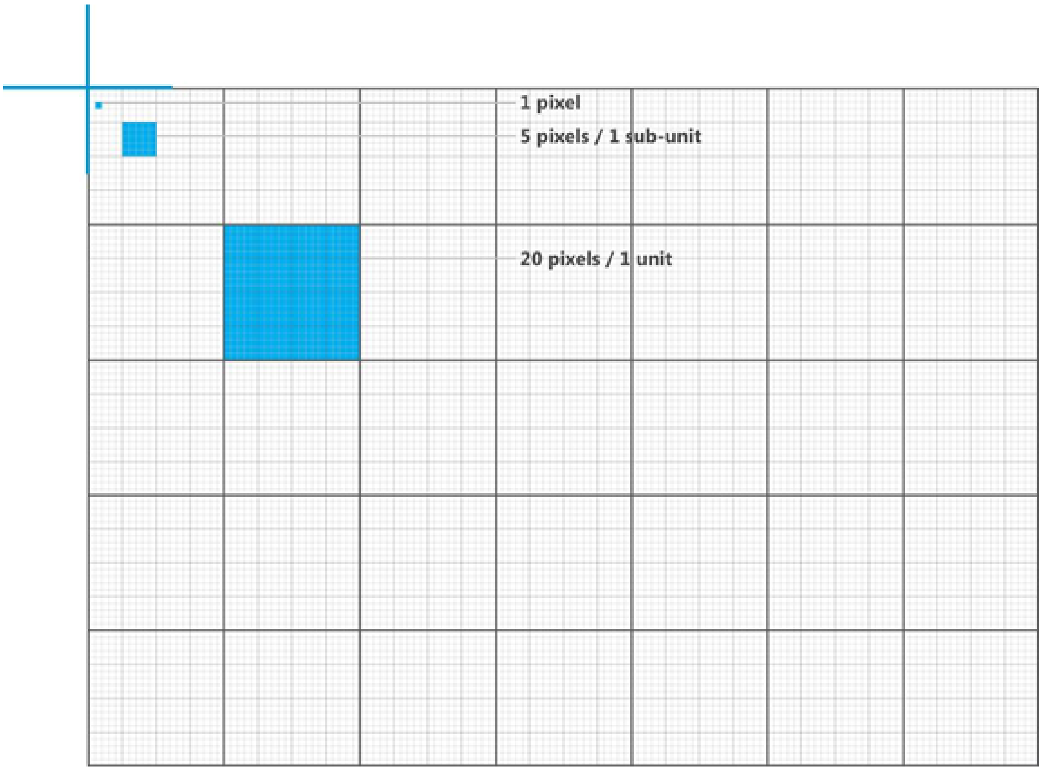
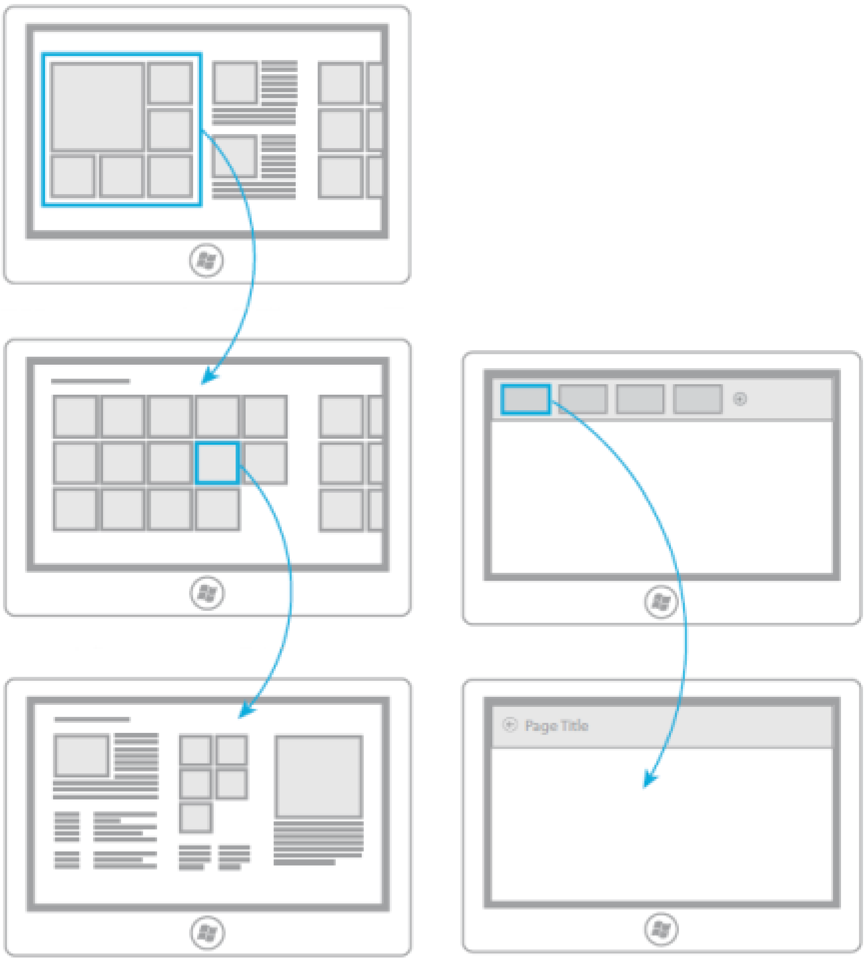


Figure 2.2: Windows 8 grid page layout [24]

The grid is used for defining accurate placement of elements like the page header, the main content canvas, and padding between content items within the content canvas. Microsoft has designed controls and data collections that are compatible with the grid system and the advised layout pattern.

2.3.2.2 Navigation Pattern

Developers are also provided with a set of navigation patterns for the application. One pattern is a hierarchical pattern, called the hub system, which focuses on separating the content into different levels of detail and pages [25].



(a) Hub navigation design (b) Flat navigation design

Figure 2.3: Windows Store app navigation patterns [25]

The navigational flow is divided into three different levels: the hub pages, the section pages and the detail pages (Figure 2.3(a)). The hub pages are the entry point of the application, and displays different categories of content, where each category corresponds to a section page. In this view, the content of the chosen category is presented as individual items leading to a detail page. This page shows details of the content, and also exposes functionality in the application. This page resides in the third level of the application, and can vary vastly depending on the nature of the application. I.e. A video application may only display the video in the detail page; whilst a detail page could also be used for a remote control application with several control elements.

For scenarios where the user will need to switch rapidly between pages, a flat navigation system is advised. This pattern closely resembles that of a modern web browser, and the ability to switch between tabs. In Windows Store apps this is done by using a UI-element called the *top app bar*, which allows the user to change the app context or page through a drop down menu, illustrated in Figure 2.3(b).

2.4 Model View ViewModel

If a system is fairly complex, it is often helpful to utilize a design pattern for organizing the code, hence making the application easier to comprehend, maintain and extend. Model View ViewModel (MVVM) is a design pattern based on the Model-View-Controller (MVC) pattern for use in software engineering, and is widely adopted. It allows for a clear separation between the back end logic of the application and the development of the user graphical user interface [33].

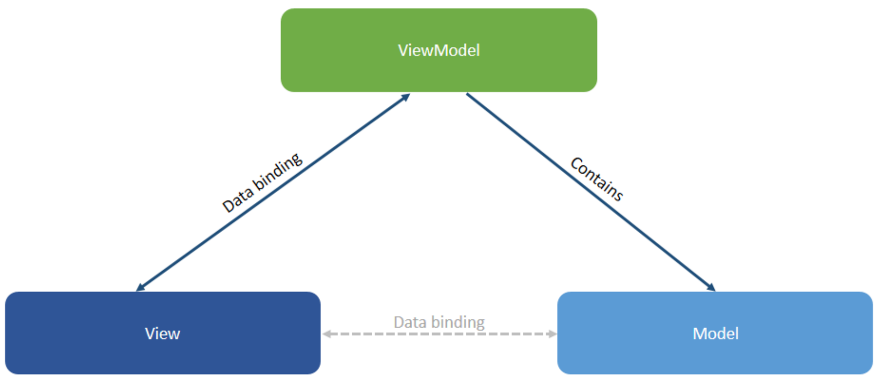


Figure 2.4: Model View ViewModel pattern

MVVM is a refinement of the MVC pattern, as it utilizes data binding to further separate the development of the view by removing all GUI-related code from view layer. The Model is the business logic and data access layer, with no dependency to the UI and only handles the processing and state of the application data, as in MVC. The view is the layer closest to the user, consisting of visual elements like buttons, labels, controls and other GUI-elements. Lastly, the ViewModel is responsible for mediating between the view and the model, and acts as an abstraction of the view. In this mediation process data binding is used. Through one-way data binding, some data can be bound directly to the Model, but with more complex operations in the UI, the ViewModel is used to perform data transformation between Model types and View types [12].

2.5 Sports Related Background

2.5.1 Exercise

In the context of this thesis, an exercise is defined as a set of videos illustrating a soccer exercise, along with some textual information describing the exercise and practical details related to the exercise. The Exercises are divided into four stages: *Preparation*, *Passing*, *Possession*, *Playing*. The preparation stage includes getting ready for the exercise (i.e. warmup). In the other three stages, the scope of the exercise widens. In the passing stage, players are focusing on a specific task, with more attention to detail. In the possession stage, more elements are combined into one exercise. Finally, in the Playing stage the exercises are put together and real life match situations are simulated.

2.5.2 Athlete Physical Status

Several studies evaluating physical performance in athletes utilizes Rating of Perceived Exertion (RPE) and Pre-Training Wellness (PTW) reports for analysis. These analyses focus on the fatigue responses and the physical status of individual athletes[22].

2.5.2.1 Rating of Perceived Exertion (RPE)

The RPE scale is a common method for determining exercise intensity in athletes, and is used both in medical studies and as a description of intensity in training sessions [22]. The scale rates a perceived degree of exertion, by self-analysis of the physical impact introduced by a workout session, and is thus a subjective measurement [32]. One of the most common definitions of the RPE scale is the one presented by Gunnar Borg called the Borg CR10

[32]. The Borg CR10 scale is divided into 10 increments, from 0 (representing rest) to 10 (indicating maximal exertion) as illustrated in Figure 2.5.

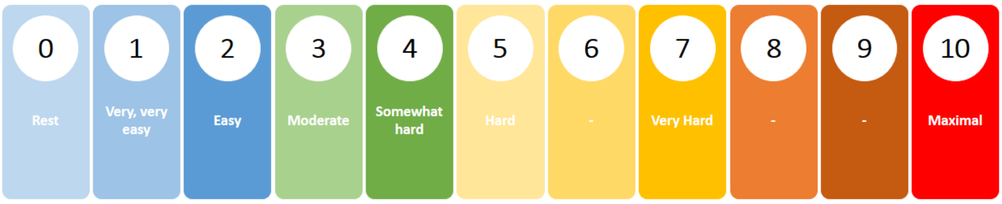


Figure 2.5: Borg CR10 RPE Scale

During or directly after the physical activity, the subject subjectively rates his or her level of exertion. The subject should not focus on singular factors like muscle pain in arms/legs or shortness of breath, but rather a total feeling of exertion. When utilizing RPE reports in the context of training sessions, the duration of the exercise is often multiplied with the exertion rating for quantifying the training load of a session [22].

2.5.2.2 Pre-Training Wellness (PTW)

Research shows that by recording athletes self-assessment of well-being, it is possible to predict overtraining based on a set of parameters of fatigue, sleep, stress and muscle soreness [22]. PTW reports have shown to be able to predict overtraining and improvement in competitive performance [15]. Recent studies have also shown that PTW reports can be used to described changes in perceptual fatigue through a team sport season [23].
A PTW questionnaire rates fatigue, sleep. Soreness and stress on a scale from 1 to 7, each value represented by a description from excellent/great to terrible/bad, shown in Figure 2.6. It also may include a value for registering illness.

	1	2	3	4	5	6	7
Fatigue	**Very Fresh**	**Fresh**	**Better Than Normal**	**Normal**	**Worse Than Normal**	**Tired**	**Always Tired**
Sleep	**Excellent**	**Very Good**	**Better Than Normal**	**Normal**	**Worse Than Normal**	**Restless**	**Terrible**
Soreness	**Feel Great**	**Feel Good**	**Better Than Normal**	**Normal**	**Worse Than Normal**	**Sore**	**Very Sore**
Stress	**Very Relaxed**	**Relaxed**	**Better Than Normal**	**Normal**	**Worse Than Normal**	**Stressed**	**Highly Stressed**
Illness	**No**	**Off Color**	**Yes**				

Figure 2.6: Pre-Training Wellness questionnaire [23]

Chapter 3

Requirement Specification

This chapter outlines the requirements of the system. In relation to the chosen user-centered design methodology, this section describes the *Requirement Analysis* and the *Conceptual Proposal* phases, both completed in close collaboration with the staff and athletes at Tromsø IL. An overview of the system and the required functionality is presented along with a conceptual architecture.

3.1 Overview

The envisioned system shall serve as a tool for capturing, persisting, presenting coach knowledge and expertise in the domain of competitive sports. The requirements of the system ranges from the ability for a coach to digitally persist sets of exercises, to collecting reports related to fatigue and exertion from individual athletes.

As the users of the system will be mostly people with non-technical backgrounds, and with a varying degree of technical competence, all features of the system must be easy to use, and present a clear and simple User Interface (UI).

At an abstract view, the system will be comprised of two main components. The first is a library that supports capturing and presenting the digital footprint of expert coach knowledge, for use as an e-learning tool for exercise drills. The other main component is a set of tools for registering and presenting reports from athletes regarding fatigue and exertion, as there is a link between exercise intensity and training benefits [22].

3.2 Coach Knowledge and Expertise

In professional sports, it is not uncommon to have players and staff members leave the organization after a period of time. During the period of engagement, the coaching team will contribute with expert knowledge, and in the case of a soccer team, instill a certain style-of-play. If this expert knowledge is undocumented, and a coach decides to leave the team for other ventures, the information and knowledge is no longer available to the organization. Another factor to consider is the new arrival of both coaches and players. Professional sports teams tend to be comprised by athletes and coaching staff with differing languages and sports culture. Today, presentation of exercises is performed using flip-overs and small, handheld whiteboards.

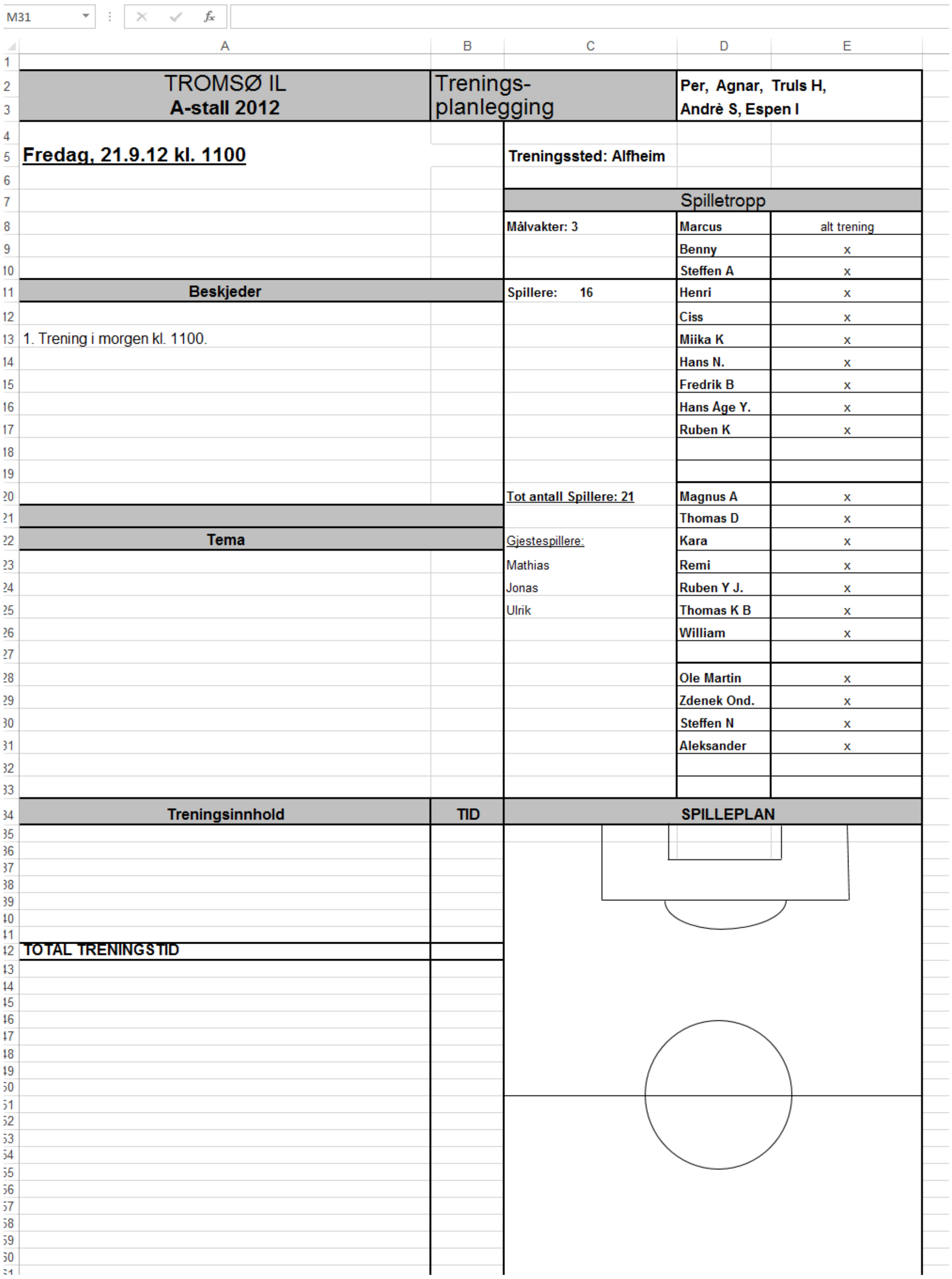
The tools for planning exercises are also quite rudimentary. Figure 3.1 shows a screenshot of the planning system currently in use by the head coach at Tromsø IL.

We suggest a system that will allow sports organizations to capture the digital footprint of this expert knowledge in the form of technical exercises and exercise plans. The system will also enable presentation of the captured content to new and existing players, coaching staff and potentially to partnering organizations. The content uploaded to the such a system would also allow for educating the organizations younger teams regarding style-of-play.

3.2.1 Capture

The content of the captured expert knowledge will be in the form of videos and textual information. The users will need to be able to upload video snippets illustrating different aspects of an exercise, along with text and images for further clarification and identification. This requires both an interface for uploading of content, as well as a back-end storage system for persisting the data.

We propose a simple web interface for performing the content uploading, as it simplifies deployment and increases client side availability. The storage solution needs to support both Binary Large Objects (BLOBs) for the video files, and persisting the exercises in a structured manner.



	A	B	C	D	E
1					
2	TROMSØ IL	Trenings-		Per, Agnar, Truls H,	
3	A-stall 2012	planlegging		Andrè S, Espen I	
4					
5	**Fredag, 21.9.12 kl. 1100**		Treningssted: Alfheim		
6					
7			Spilletropp		
8			Målvakter: 3	Marcus	alt trening
9				Benny	x
10				Steffen A	x
11	Beskjeder		Spillere: 16	Henri	x
12				Ciss	x
13	1. Trening i morgen kl. 1100.			Miika K	x
14				Hans N.	x
15				Fredrik B	x
16				Hans Åge Y.	x
17				Ruben K	x
18					
19					
20			Tot antall Spillere: 21	Magnus A	x
21				Thomas D	x
22	Tema		Gjestespillere:	Kara	x
23			Mathias	Remi	x
24			Jonas	Ruben Y J.	x
25			Ulrik	Thomas K B	x
26				William	x
27					
28				Ole Martin	x
29				Zdenek Ond.	x
30				Steffen N	x
31				Aleksander	x
32					
33					
34	Treningsinnhold	TID	SPILLEPLAN		
42	TOTAL TRENINGSTID				

Figure 3.1: Planning interface currently used by Tromsø IL

3.2.2 Plan

In professional sports, combinations exercises are constructed in order to reach specific goals, so capturing only stand-alone exercises will not be sufficient. When performing the requirement analysis for the Ruoksat system, the main coach at Tromsø IL informed us that the planning of exercises is done in weekly batches, with each daily session being comprised by multiple exercises. This means that the Ruoksat exercise library needs to implement the support for combining multiple exercises into a daily training session, and also plan and organize one session for each day of the week. This requires that the system is able to persist structured data for daily and weekly exercise plans.

As with the content capturing functionality, we propose a web interface for organizing exercise plans, as this would make the tool available to the coaches from multiple locations, with only the need for an internet connection and a web browser.

3.2.3 Present

The presentation of the content is crucial for the system to be useful. Support for video playback and rendering of corresponding textual information are needed, and the layout must be organized in a way that is easy to navigate. In addition, the exercise content displayed must be clear and simple to understand. The interface for presenting the content must be able to communicate with the systems databases, either directly or through some form of mediating service.

The example exercises we have collected from Tromsø IL show a main exercise drill along with some coaching points, or details related to the exercise. This means that the system will need to support highlighting, or hiding, elements of the exercise data, based on a desired level of detail in the presentation.

When presenting the initial plans for the head coach at Tromsø IL, he expressed a desire for linking the exercise to a match related motivational event. The ability to relate exercises to actual match incidents is helpful in conveying the importance of an exercise to participating athletes. Another feature requested during this requirement analysis was the ability to project the video from the exercise onto an external display (i.e. big screen TV), as this enables the coaches to present the exercise to the entire team simultaneously.

Since training sessions usually take place outdoors or on off-site locations, the device running the presentation of the system will need to be portable.

3.3 Perceptual Fatigue and Athlete Exertion

For top level athletes, fatigue and exertion are important factors to consider when planning and executing workout sessions. A big part of the Ruoksat system is the ability for the coaches to consider the physical state of both individual athletes, but also the entire team. The current situation is that Tromsø IL is utilizing a Pre-Training Wellness (PTW) form [23], which is delivered each morning by all of the athletes, along with a Rating of Perceived Exertion (RPE) [32] after each training session. The wellness reports are used to keep track of each players physical state, and in combination with the post-workout RPE reports, the medical staff can assess the athletes on an individual basis.

Being able to vary exercise load and intensity for each session based on physical feedback from the participating athletes is crucial [32]. Keeping track of the exertion levels and the general wellness of each individual athlete, allows the coaches to adjust exercise intensity, or rest specific athletes, to avoid serious injuries [23]. Additionally, by recording the perceived fatigue of the entire team, the weekly exercise plan can be fine-tuned to achieve peak performance and low fatigue levels on match day [22].

When defining the requirements for the Ruoksat system, Tromsø IL presented us with the current solutions for capturing and utilizing these kinds of athlete report data. Currently, the medical staff is handing out and collecting paper sheets for the players to register a perceived rating of exertion immediately after each training session. The same goes for the PTW reports. This results in a lot of paperwork and manual registration for the medical staff, and there is no dedicated system for storing, aggregating and presenting the registered data, with no means for longitudinal analysis (i.e. weekly or monthly). In addition, medical staff is receiving the PTW ratings directly before practice, giving them no time to analyze and adjust according to the reports.

After a discussion with the medical staff at Tromsø IL we presented ideas for digitally gathering the reports, and displaying the data in a simple interface. We decided upon a web interface for presenting the data, and that reporting of PTW and RPE should be supported on smartphone devices, making the submitting process of the reports accessible and convenient.

3.3.1 Athlete Status Report Application

The application that will enable athletes to report their physical status needs to support two different kinds of reports; daily pre-training wellness and a rating of perceived exertion. As the athletes should be able to use their private mobile phones for reporting, the application needs to be supported by variety of different smartphone devices. Another aspect that has to be considered is the varying level of technical competence, and that the interaction design of the application needs to be adapted to the users.

3.3.2 Athlete Status Web Interface

For presenting the report data registered by the athletes, we propose implementing a web interface, opposed to a stand-alone, platform specific application. Utilizing web browsers for the interface simplifies development and deployment, as well as significantly improving accessibility.

The need for a web interface for displaying data related to the athlete status reports introduces a set of requirements. The interface will need to present data from several reports simultaneously, and the system is of an asynchronous nature, as it would not be practical for the medical staff to receive the reports directly from each athlete, without being able to persist the data. This along with a need for keeping historical data for tracking trends and athlete development, introduces a need for a back-end storage system.

Through active discussion with the medical staff at Tromsø IL, and an iterative requirement analysis we have discovered the need for the following set of functional requirements for the interface.

Overall Team Average

Based on the collected data, the interface should display a day-to-day overall status for the entire team. This includes an average of the RPE reports from all the athletes, along with the average fatigue levels for the team. Additionally, in order to monitor the exertion and fatigue trends of the team as a whole, this section of the interface should show a chart of the most recent team reports. As this interface is part of the Ruoksat exercise system, the medical staff must be able to forward a suggestion to *change training intensity* based on the team status.

Individual Athletes

During our requirement analysis, we discovered that the most important data collected by the medical staff is the data on each individual athlete. As mentioned previously, adapting the exercise intensity to each individual athlete is crucial for achieving maximal benefit from each workout session, as well as avoiding injuries. In order to streamline the process of analyzing the reports of each athlete, the interface needs to present the data in two different views; an overview of all athletes, and additionally a more detailed profile.

The overview would need to display a basic set of information from the most recent status reports. Based on this information the medical staff can choose to inspect each athletes physical status profile individually. The individual profile would need to display complete PTW [3] and PTW reports, and show historical data for further analysis. Just as the team status section, the profile interface will need to expose functionality to enable forwarding of recommendations on exercise intensity for each athlete. This also includes the ability to suggest that an athlete should be removed from the practice altogether, based on the report data.

3.4 Non-Functional Requirements

In addition to the functional requirements described in the previous sections, the system also poses a set of non-functional properties. The following section presents the main non-functional requirements.

3.4.1 Usability

As mentioned in Section 3.1, the users of the system will be of varying technical competence, and most of them come from a non-technical background. This means that the interfaces of the system will need an intuitive layout, with simple interactions, resulting in a system with a high degree of usability.

3.4.2 Scalability

The underlying storage systems need to scale horizontally in order to serve potentially large amounts of data, to big user groups.

3.4.3 Privacy and Security

Privacy is crucial in the system, both in regards to the exercise knowledge library, and the application for reporting and monitoring athlete status. Access to the exercise data needs to be secure, as exposure of style-of-play details and exercise drills may result in losing the competitive edge towards other sports organizations. The RPE and PTW reports also needs to be held private, as it contains medical data related to identifiable individuals, and we also need the players to provide us with a consent to store this data.

3.4.4 Performance

Given the short duration of a training session, and the amount of drills needed to be performed in that short time period, the coaches and athletes do not have the time to wait for a slow UI. If the videos take too long to load, the flow of the training session is interrupted and crucial practice time is lost. The same applies to the athlete reporting. The performance of the reporting system is directly related to usability, and the user experience would suffer greatly from a slow interface.

3.5 Summary

In this requirement analysis, we have discovered a set of functional and non-functional properties. The Ruoksat system will include clients like web interfaces, external display applications, smartphones and tables. A storage system for video files and storing structured data is needed, along with a series of back-end services to handle the logic between the end-user interfaces and the underlying storage. We have also presented non-functional requirements like usability, privacy, performance and the ability to scale horizontally. A conceptual architecture is illustrated in Figure 3.2.

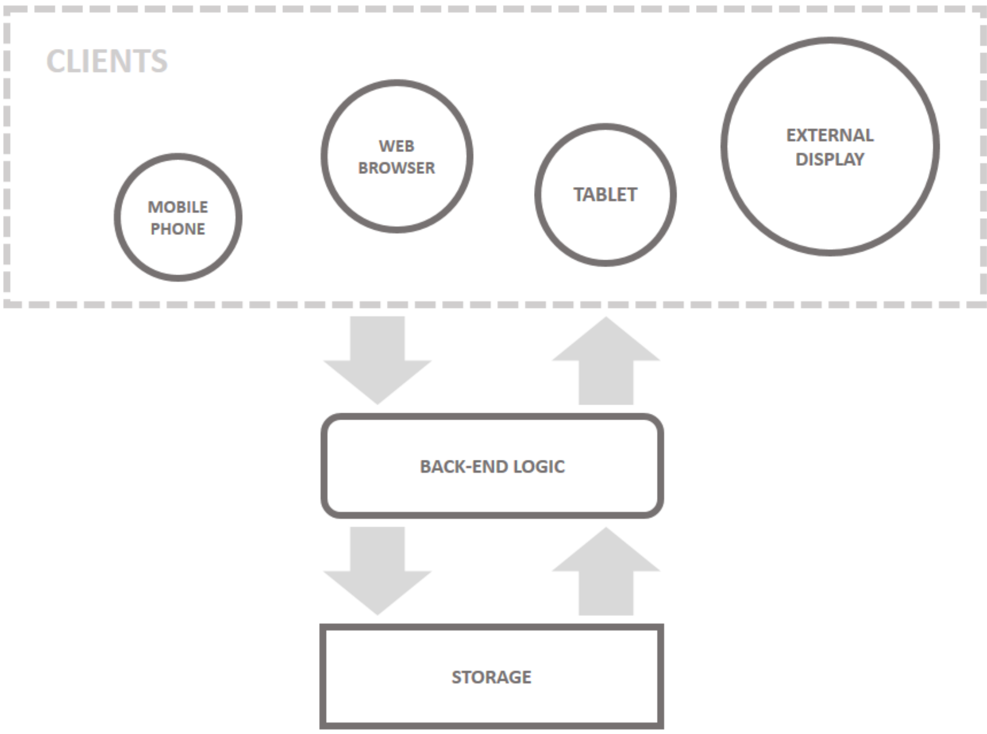


Figure 3.2: Conceptual Architecture

Chapter 4

System Model

This chapter presents the general system model. We outline an abstract view of the architecture and present the three main stages of the system.

4.1 Overview

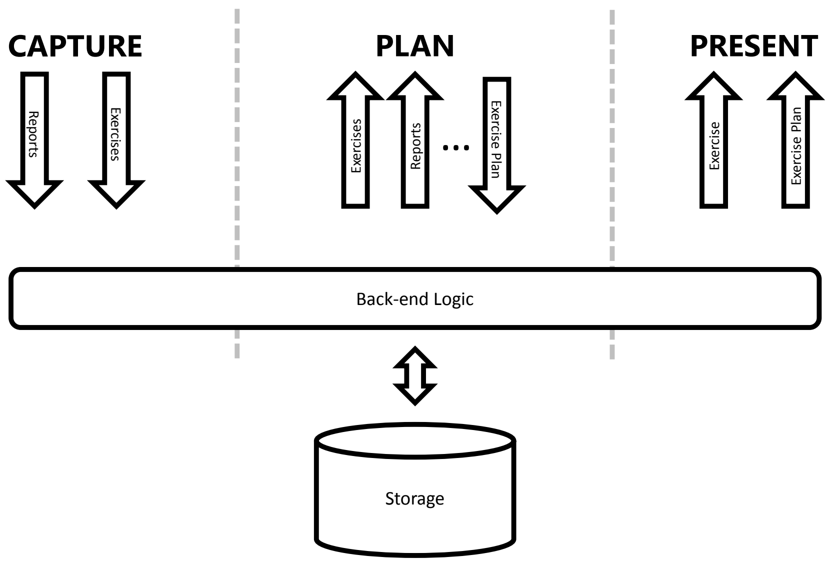


Figure 4.1: Abstract Architecture: Overview

In the requirement specification we described two different applications, a library for capturing coach knowledge and expertise in the form of exercises, and an application for collecting and displaying athlete status reports. Both of the applications will need interfaces that interact with back-end services and need persistent storage. Based on this we present a three-tier system model, divided into three different main stages: *Capture*, *Plan*, *Present* (Illustrated in Figure 4.1).

4.2 Capture

The capturing stage of the system will need to handle data input both in the form of uploaded exercises, and Rating of Perceived Exertion (RPE) and Pre-Training Wellness (PTW) reports from the athletes (Figure 4.2).

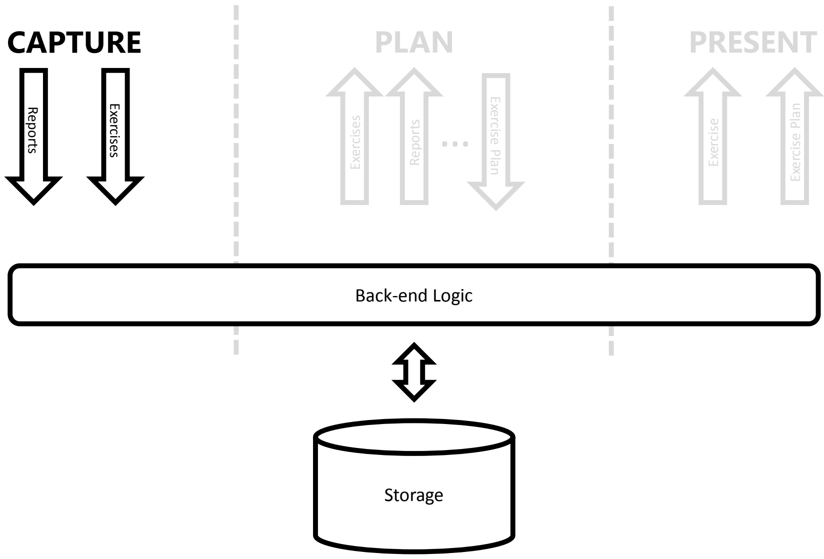


Figure 4.2: Abstract Architecture: Capture

4.2.1 Uploading Exercises

When a user uploads data related to an exercise, the system must propagate the data from the interface layer to persistent storage. This requires back-end logic for handling the input. The uploaded exercise will contain both data describing the exercise and elements related to the exercise, as well as the video files illustrating the exercise. This means that the back-end logic layer needs to organize the descriptive data in a structured manner, and forward it to a database. Additionally, the video data uploaded must be stored in a data storage suitable for Binary Large Objects (BLOBs).

4.2.2 Athlete Status Reports

The other aspect of the capturing stage is the handling of RPE and PTW reports from the individual athletes. Data is sent from the interface to the back-end, where it needs to be parsed and organized based on a suitable data model. The data must then be persisted in a database for structured data for use in the following stages of the system.

4.3 Plan

When an exercise plan is constructed, the system is in a planning stage. The exercise-planning interface will need to be presented with the currently available exercises, along with the data regarding the physical status of the athletes. In addition to downloading athlete status reports and stand-alone exercises, the planning stage of the system must also be able to upload constructed exercise plans (Figure 4.3).

4.3.1 Athlete Status Analysis Report

The first phase of the planning stage is the analysis of the athlete status reports. The medical staff needs to forward an analysis of the registered RPE and PTW to the coaching responsible for exercise planning, so this can be considered when constructing an exercise plan. This requires that the back-end layer is able to forward the analysis report to the exercise planning tools.

4.3.2 Creating Exercise Plans

In addition to the analysis report, the planning interface will need access to the exercise data from the storage layer, including both video data and the structured data describing the exercises. When an exercise plan has been constructed, it must be linked to the included exercises, along with a set of specified durations, and persisted in the underlying storage (Figure 4.4).

4.4 Present

In the presentation stage, the exercise plan and the exercises will be forwarded to a presenter interface. This means that the back-end needs to facilitate downloading of content from the storage layer.

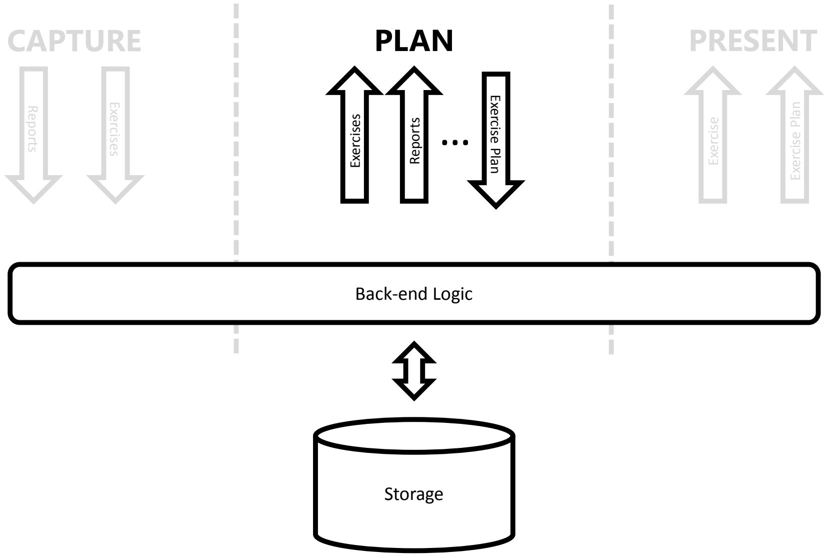


Figure 4.3: Abstract Architecture: Plan

4.4.1 Presenting Exercises

The system must support displaying of exercise related content in the presenter interface; hence, data from the underlying storage will need to be extracted by the back-end logic, and formatted to suit the data model of the interface. Additionally, in order to support video playback on a big screen interface, the system must be able to connect the presenter application to the big screen interface, through either direct communication or a mediating service.

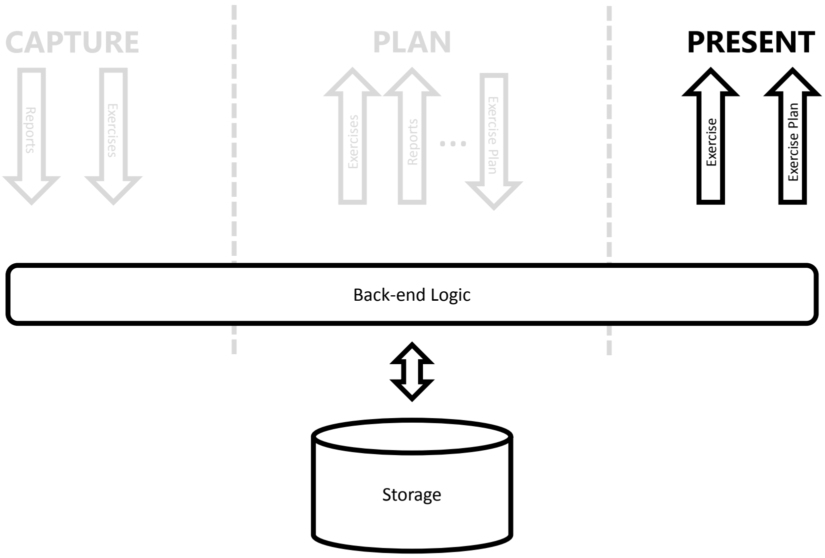


Figure 4.4: Abstract Architecture: Present

Chapter 5

Ruoksat

This chapter presents the Ruoksat system. Ruoksat is a word from the Northern Sami language meaning *red*, as in the norwegian expression *rød tråd*, which in english roughly translates to treadline. The idea behind this is that there is treadline connection between the senior first teams of a soccer organization and youth academy concerning style-of-play. It also refers to the connection between the athlete status reports and exercise planning and execution.

We begin by discussing the system architecture, before describing the mockups created as visual prototypes and the implemented solution. A series of details related to the implementation of the system concludes the chapter. A demo video of the Ruoksat system can be found at `http://site.uit.no/iad/demo`. The source code is available upon request for evaluation purposes.

5.1 System Architecture

5.1.1 Interfaces

The Ruoksat system (Figure 5.1) is comprised by a set of user interfaces connected to a back end storage layer through a set of services. Residing in the user interface layer we have several components; A smartphone for registering and submitting physical status reports (1), and two web browser interfaces; one for presenting status reports (2) and another rendering a exercise planning tool (3). Another important interface in the Ruoksat system is the presenter app, residing on a tablet device, for displaying and executing exercises (4). The last interface is a big screen display for presenting videos from the tablet device on an extended display application (5).

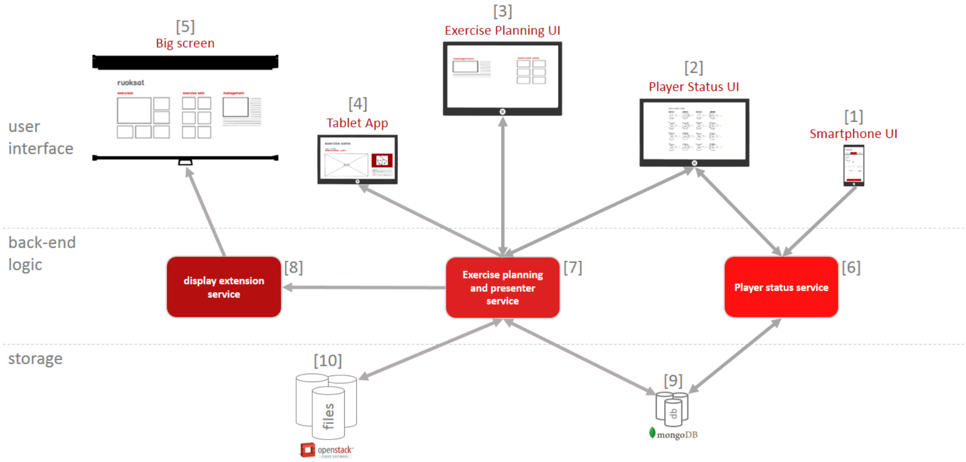


Figure 5.1: Arhcitecture: Complete System

5.1.2 Services

In the middle layer, the back-end logic resides. This layer handles the logic between the interfaces and the underlying storage systems, and consists of three different services.

The status service (6) handles incoming status reports from the athletes, and organizes and presents the persisted data for the medical staff UI (Figure 5.2).

All data to and from the exercise planning interface and the tablet presenter app is handled by the exercise planning and presenter service (7) (Figure 5.3). This service also forwards notifications from the athlete status interface to the exercise planning to, so the fatigue and exertion of the athletes can be taken into account when considering exercise intensity.

Figure 5.4 shows the architecture with a display service (8) for extending the presenter app display. This service allows the user of the tablet to forward the video to a dedicated app connected to a big screen interface.

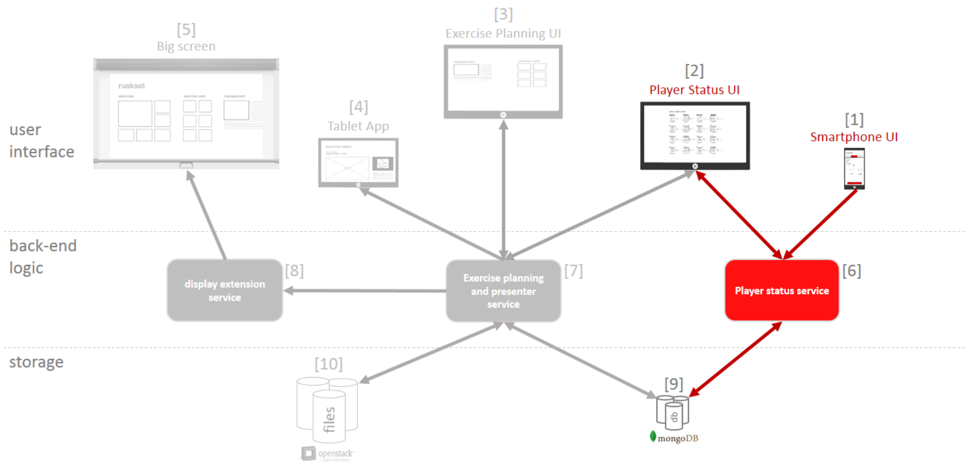


Figure 5.2: Arhcitecture: Athlete status reporting

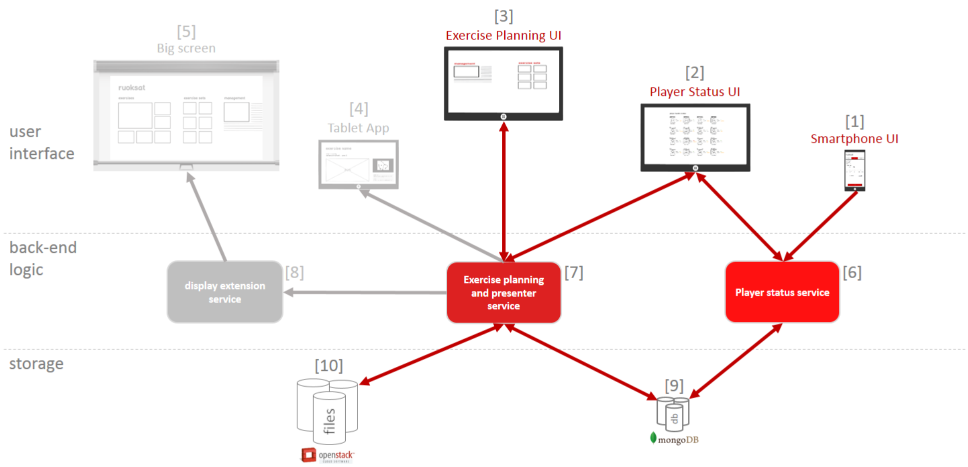


Figure 5.3: Arhcitecture: Exercise planning

5.1.3 Storage

Two different data stores comprise the storage layer, one for serving video files (or other static data) (10), and database for organizing collections of data related to the exercises and athlete status reports (9). A more detailed view of the storage layer is presented in the System Implementation section of this document (Section 5.4.1).

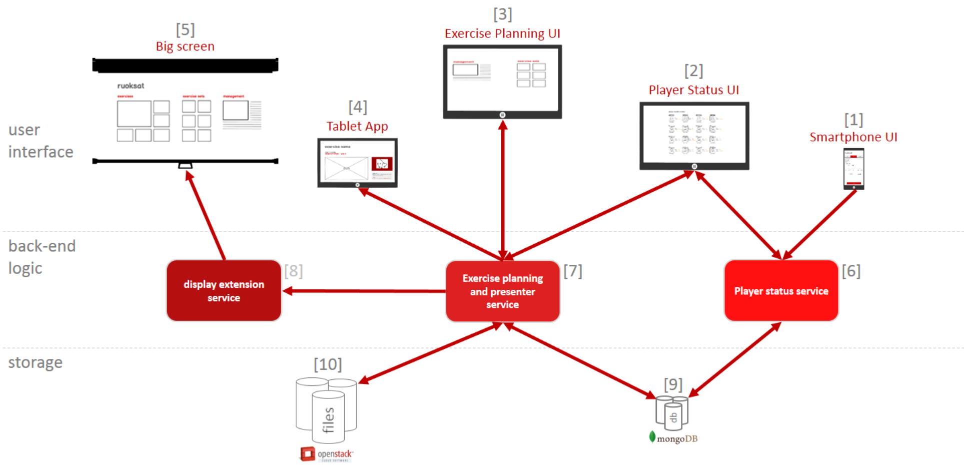


Figure 5.4: Arhcitecture: Exercise presentation/execution

5.2 Visual Prototyping

Since this system is designed and implementation in close collaboration with Tromsø IL, a set of mockups of the planned user interfaces needed to be created before the implementation could begin. The visual prototyping phase allows us to abstractly evaluate the usability of the system [7]. Additionally, by exposing these mockups to the end users of our system allowed us to re-evaluate the requirement analysis and the conceptual proposals in collaboration with both the athletes and the coaching staff at Tromsø IL.

5.2.1 Exercise Knowledge Library

5.2.1.1 Presenter

The mockups for the exercise presenter application illustrate a layout for a tablet device. More specifically, it follows the layout and navigational pattern of a Windows Store app, as presented in section 2.3. Figure 5.5 shows a visual prototype for the hub page of the application, which is the main menu of the application, working as an entry point for the user [25]. In the hub page, the user is presented with either stand-alone exercises from the system, or pre-built exercise sets.

When the user has chosen a specific exercise, he navigates to the section pages, which shows the exercise plan for that exercise, divided into sections

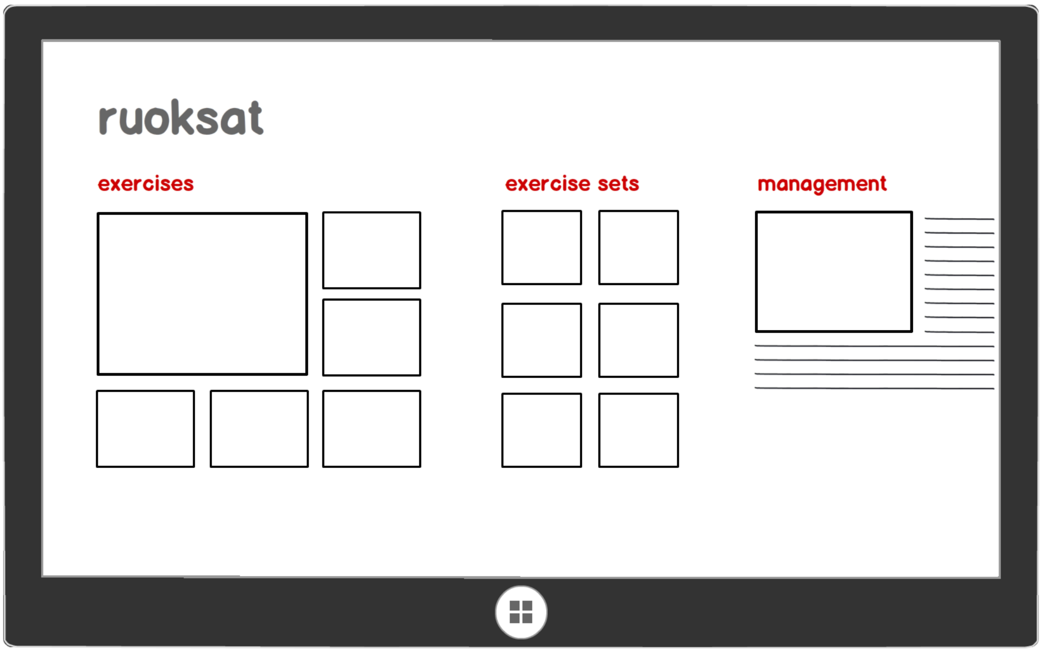
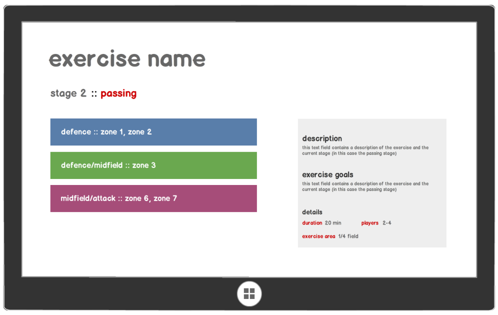


Figure 5.5: Mockup: Hub page of the exercise presenter app

or zones of the playing field. Figure 5.6 illustrates the two different layouts presented to the coaching staff at Tromsø IL during the development of the system. On the left (Figure 5.6(a)), a section page which shows a football pitch divided into sections, with some corresponding information in a grid to the right. In this layout, the user would click the desired sections, and this would in turn navigate them to the correct exercise (i.e. defensive) for that zone of the pitch. The other layout presented was Figure 5.6(b), where the sections are simply divided into a list, each describing the zone and for which playing positions the exercises is intended. This was the layout chosen by the coaches, as it was easier to interact with, and also gave enough information to the user regarding how navigate to the desired exercise.

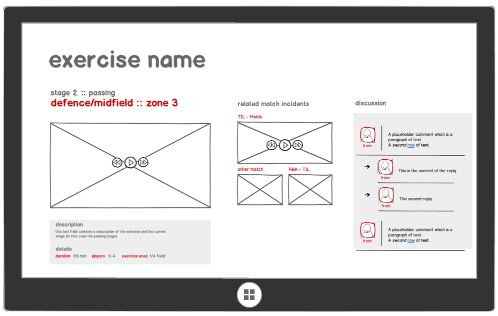
(a) Soccer field grid layout



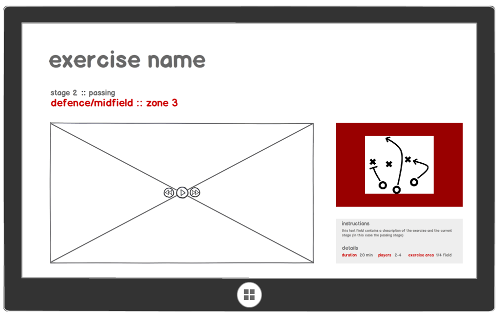
(b) List layout

Figure 5.6: Mockup: Visual prototypes for the section pages of the exercise presenter

Presentation of the each individual exercise is performed in a detail page. This page shows details related to the exercise, in addition to the exercise video. As with the section pages discussed above, we presented two different layouts for the coaching staff. One was a content rich detail page (Figure 5.7(a)), showing discussions related to the exercises, along with a set of motivational events related to the exercise. The other visual prototype, shown in (Figure 5.7(b)), shows a more simplistic layout, with only some basic information from the exercise, along with a small illustration. The layout chosen for further development was the simplistic version, as the Tromsø IL coaches indicated that this layout presented enough information, and rendered a clearer interface.



(a) Content rich detail page



(b) Simplistic detail page

Figure 5.7: Mockup: Two different layouts for the exercise detail pages

5.2.2 Athlete Status Application

As presented in the requirement analysis, the system for registering and presenting the athlete physical status is divided into two main separate tools - A reporting tool for the athletes, and an interface that presents the reports for the medical staff.

5.2.2.1 Reporting Tool

The visual prototype for the reporting tool is a smartphone adapted layout with a simple set of possible interactions (shown in Figure 5.8). The navigation is controlled by a button set of three buttons located in the top of the interface, creating a clear distinction between the navigation and the main functionality of the application [40].

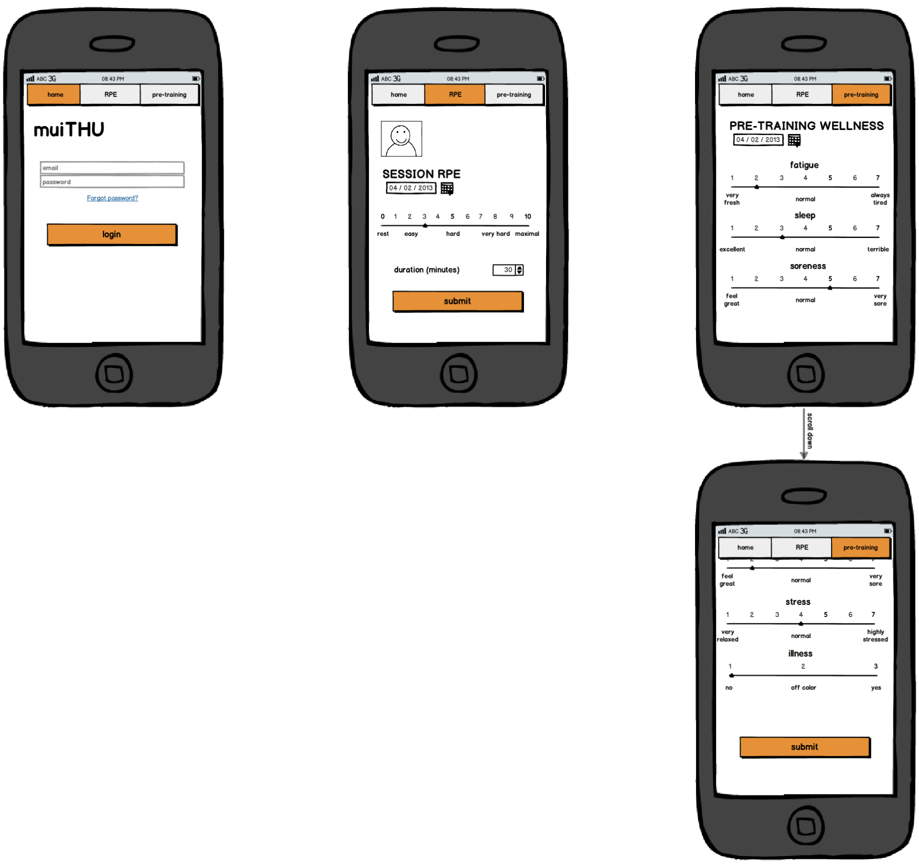


Figure 5.8: Mockup: Athlete Status mobile web app

The different views of the interface are shown in Figure 5.8. The left mockup shows a simple login screen, whilst the other mockups show the layout of the reporting tools for Rating of Perceived Exertion (RPE) and Pre-Training Wellness (PTW) reports. The visual prototype for the RPE reporting tool shows a slider bar, ranging from 1 to 10, with a description of some of the numbered values. Along with this slider bar, the interface mockup also contains a date picker tool, and an input field for registering the duration of the workout session. The duration of the exercise is, as mentioned in section 2.5.2, used in combination with the RPE value to calculate the total training load.

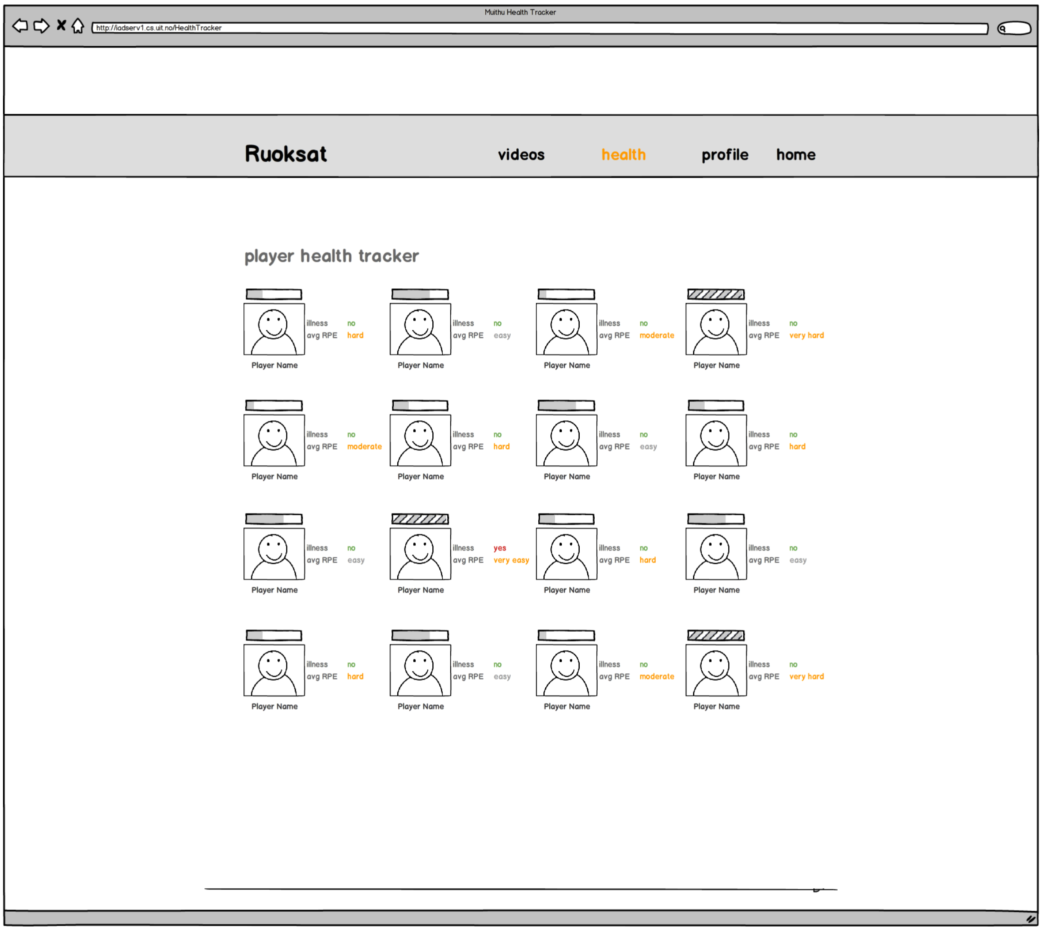


Figure 5.9: Mockup: Athlete status overview for medical staff

Similarly to the RPE interface, the view for PTW reports also utilizes a date picker and slider bars. Since the PTW reports consist of multiple values for each report, the layout presents the user with a set of slider bars, one for each category of the PTW report.

5.2.2.2 Athlete Status-Monitoring Interface

The mockups of the web interface for presenting the status reports registered from the athletes is shown in Figure 5.9 and Figure 5.10. In the visual prototypes, the web-interface is divided into two different views, one which shows an overview of all athletes and a profile view, which shows data in more detail and also historical data presented in charts.

The athlete status overview (Figure 5.9) is comprised by a grid of athletes, identified through profile images and textual data, showing the most recent report data registered. A colored bar above each profile picture indicates the overall physical status of the athlete, along with some data in regards to the wellness of the athlete.
Figure 5.10 shows the profile view, for when a user of the interface wants to do further inspection on an individual athlete. He or she is here presented with the latest reports, a set of charts to display historical data, presented by last week, month and year, and lastly a discussion forum on the top right side of the interface.

In both layouts we are using clear separation of content through chunking of information, and utizling a grid for creating a clear separation between elements [40]. We highlight important information through the use of colors, and hide what is not essential to the user.

5.3 Implemented Interfaces

This section describes the implemented interfaces, which includes the finished layout, and the connection and relation to the underlying back-end systems. Most aspects of the implementation are done in close collaboration with coaches, medical staff and athletes from Tromsø IL, in an iterative manner.

HCI methods

All interfaces are designed and implemented based on well-known principles from the field of Human-Computer Interaction (HCI) [40] [7]. We use a the contrast between grey and bright colors to guide the user through the layout, highlighting and differentiating important elements. A combination of illustrations and textual information is used to adapt to a wide variety of users [1].

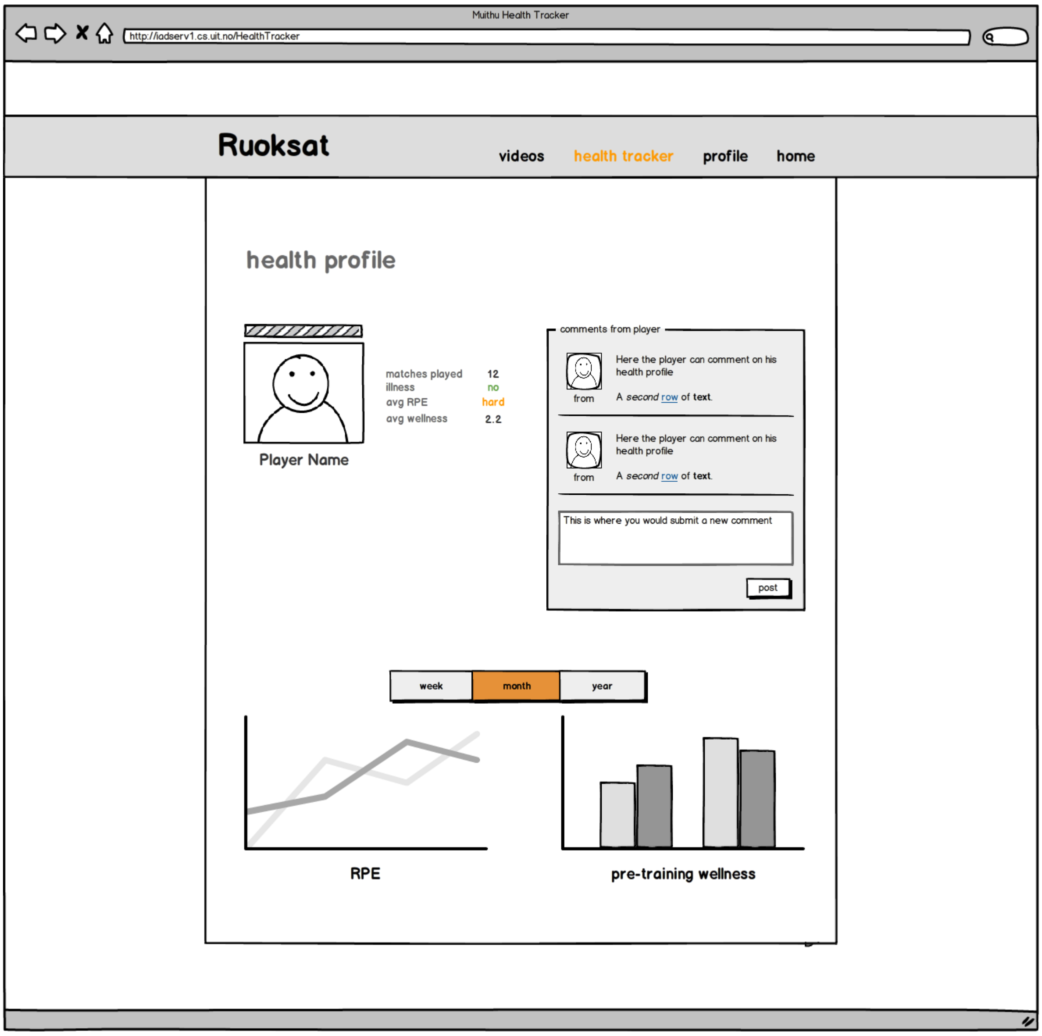


Figure 5.10: Mockup: Athlete profile for medical staff

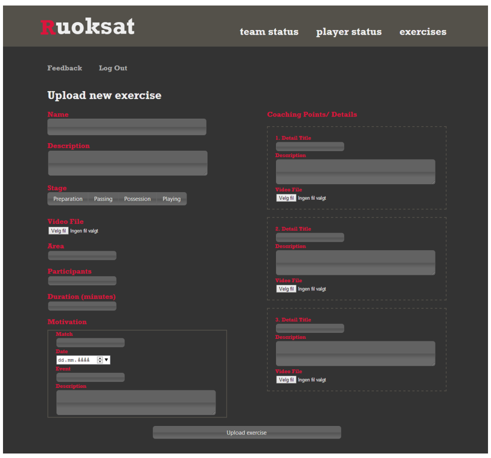
In pages of the layout where the user is presented with both an overview and a more detailed page, we re-use elements from the overview, resulting in a layout that is easy for the reader to comprehend [40]. All interfaces also use a combination of grids and white-spaces in order to create a clear separation between the elements in the layout. Finally, we use the same color palette for all interfaces in the system, creating a uniform design language throughout the system.

5.3.1 Exercise Knowledge Library

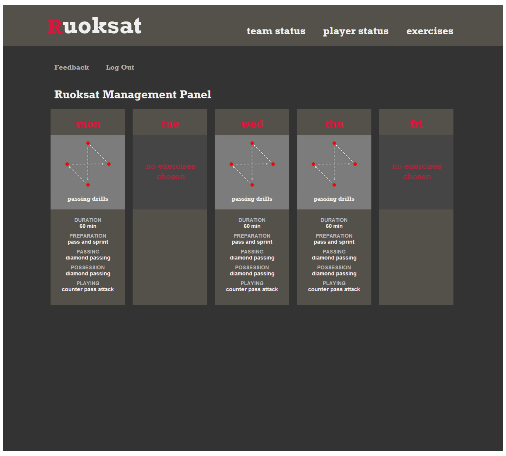
5.3.1.1 Capture

Capturing and persisting coach knowledge in the form of exercises is done through a web interface, as it does not require any advanced form of interaction, and web browser access is highly available.

Before creating a new exercise in the system, the user must have captured illustrative videos to be uploaded along with the exercise description. The layout implemented for uploading exercises is shown in Figure 5.11(a). When creating a new exercise, the user must specify a name for the exercise, along with a set of values for describing the exercise (i.e. name, number of participants, exercise area, etc.). The interface also allows the user to specify a specific match event related to the exercise, which can be uploaded directly, or fetched from the Muithu system, where match events are tagged and extracted. Finally, if there are specific technical details or focus points important to the exercise, these can be added in the form of a description and a short illustrative video snippet.



(a) Upload Exercise



(b) Week Plan

Figure 5.11: Implemented interfaces for uploading and planning exercises

5.3.1.2 Plan

When a set of exercises is captured in the system, the coaches can use the planning tool to create weekly exercise plans. Similarly to the interface for capturing an exercise, the planning of exercises is also done through a web UI.

The first view presented to the user is a week plan ranging from Monday to Friday (Figure 5.11(b)). By clicking a weekday, the exercises available in the system are presented based on which phase they belong to (Preparation, Passing, Possession and Playing. These exercises can then be combined together in order to make an exercise plan for each individual weekday, allowing the user to set the duration of each exercise chosen (Figure 5.12).

Data from the RPE and PTW reports registered by the athletes are also made available to the coach when planning the exercises. This allows for adjustments in duration and intensity based on recent and historical data on the fatigue and exertion levels of the athletes.

5.3.1.3 Present

The interface for presenting and executing the exercises is implemented on a tablet device, and follows Microsofts guidelines for page design [24] and navigational patterns [25]. All elements of the interface uses the grid system [26] for placement and alignment, for opening of the page, and simultaneously directing focus on important elements. The use of grids and white space in the canvas also create the perception of a simple and user-friendly interface [7].

When a user opens the application, she or he is met with a hub page similar to the planning interface, containing a weekly exercise plan, shown in Figure 5.13. This differs from the visual prototypes presented in Section 5.2, where the hub page instead displayed all exercises and exercise sets available to the user. The decision to change the hub page layout was made to separate the planning and presentation phases. The coaches would plan the exercises beforehand, and only the planned exercises were relevant to the coaches at the time of presentation.

By selecting a day from the exercise plan presented in the hub page, the user navigates to the section pages of the interface (Figure 5.14). The section page contains a set of exercises to be executed on the selected day, along with some information regarding duration, exercise type and other metadata related to the exercise set.

The detail page shows each individual exercise, and is the main view

Figure 5.12: Interface for editing daily exercise plans

of the presenter interface (Figure 5.15). This page contains a video player for the exercise video, some practical details related to the exercise (i.e. participants, duration, etc.), and a set of buttons organized in a grid. These buttons are made identifiable through mixing modes, combining an illustrative icon and descriptive text, for a more user-friendly interface [40].

Figure 5.13: Hub Page of the presenter app

Figure 5.14: Section page of presenter app, showing a daily exercise plan

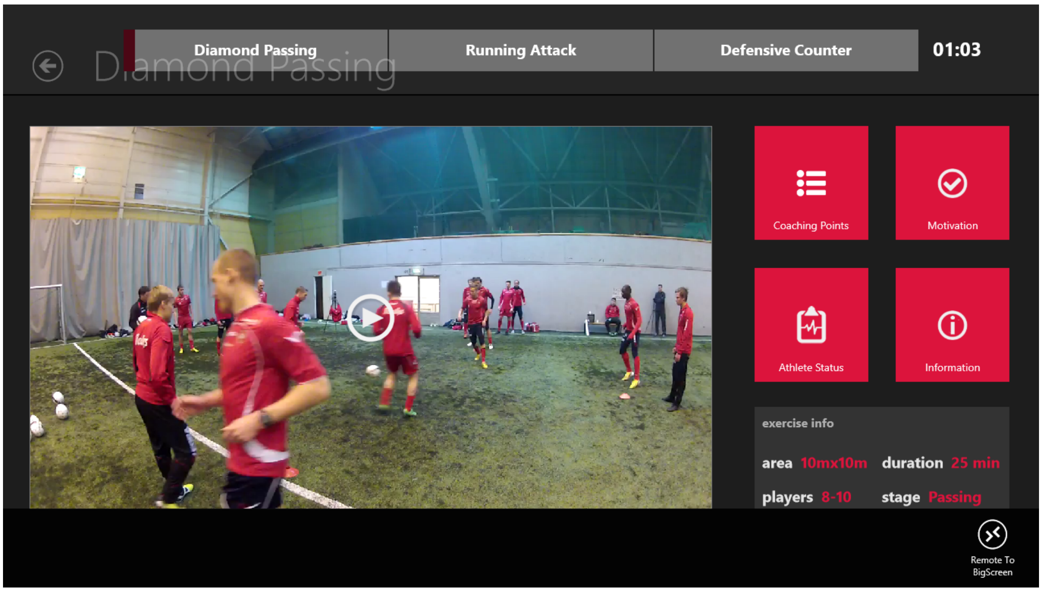
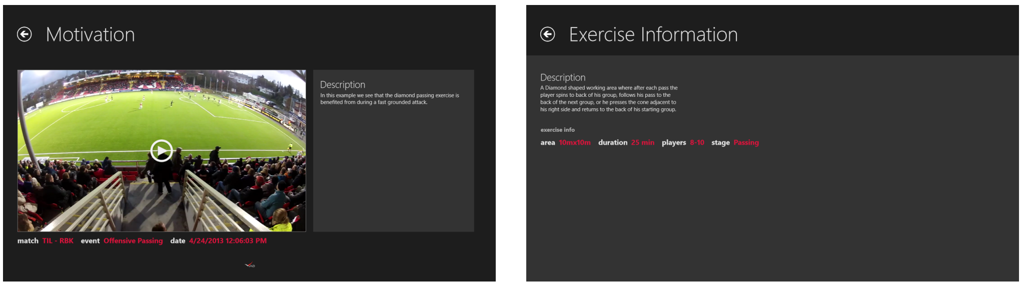


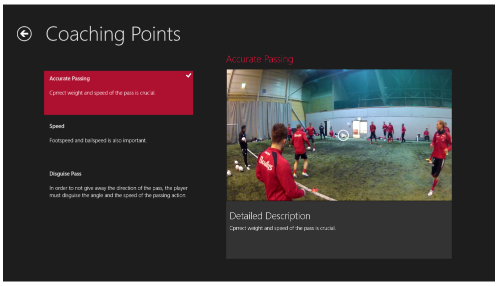
Figure 5.15: Main detail page of exercise presenter



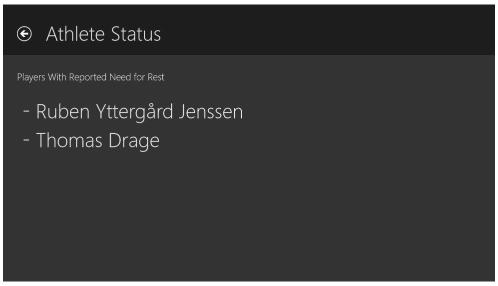
(a) Motivation Page

(b) Info Page

Figure 5.16: Screenshots of pages showing exercise motivation and info



(a) Coaching Details



(b) Athlete Status

Figure 5.17: Screenshots of pages showing

Each button leads to different pages containing information related to the exercise. These pages contain the match related motivation for the exercise, a detailed description, and a list of technical coaching details with corresponding videos, all shown in Figure 5.16 and 5.17. The last sub-detail page shows a list of athletes where the medical staff has suggested a change in training intensity based on RPE and PTW reports (Figure 5.17(b)) (the names used here are two randomly selected players from Tromsø IL, and does not reflect operational data.

These buttons allow us to hide information potentially unnecessary to the user, allowing for a cleaner interface, assisting the user by only highlighting the most important elements in the detail pages.
Another feature of the implemented presenter app is a progress bar for the planned exercises. When the first exercise is openeed in the application, a timeline is added to a hidden top bar in the interface. This bar can be accessed from any page in the app, and is displayed by swiping down on the screen (Figure 5.18. When the set duration of the first exercise has elapsed, the app automatically navigates to the next exercise. However, if the coach is not finished with the exercise when it moves on to the next, he can navigate between the exercises by tapping the progress bar.

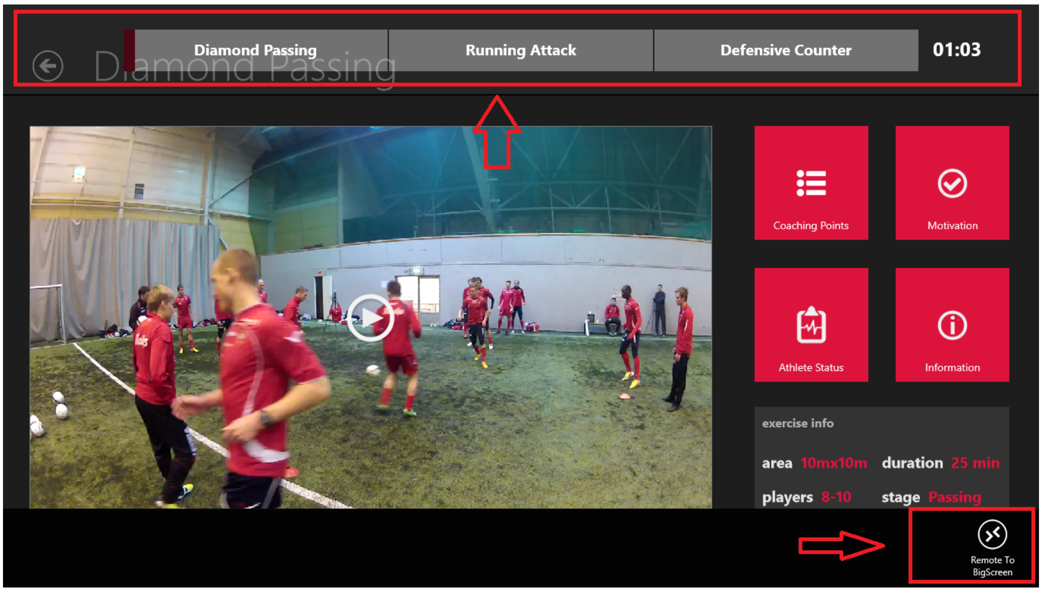


Figure 5.18: Progress bar and big-screen display forwarding for exercise presenter

In addition to the progress bar, the application also shows a bar at the bottom when a swiping motion is detected. This bottom bar exposes the button for sending the currently viewed video to a big screen display (also shown in Figure 5.18). The big screen interface is implemented through a simple Windows Store app, and when the button is pushed, the link to the current video is pushed to the big screen interface. When the user is watching an exercise video on the big screen interface, the stream can be paused and resumed by pressing the same button. Any video from the presenter can be sent to the big screen application.

5.3.2 Athlete Status Application

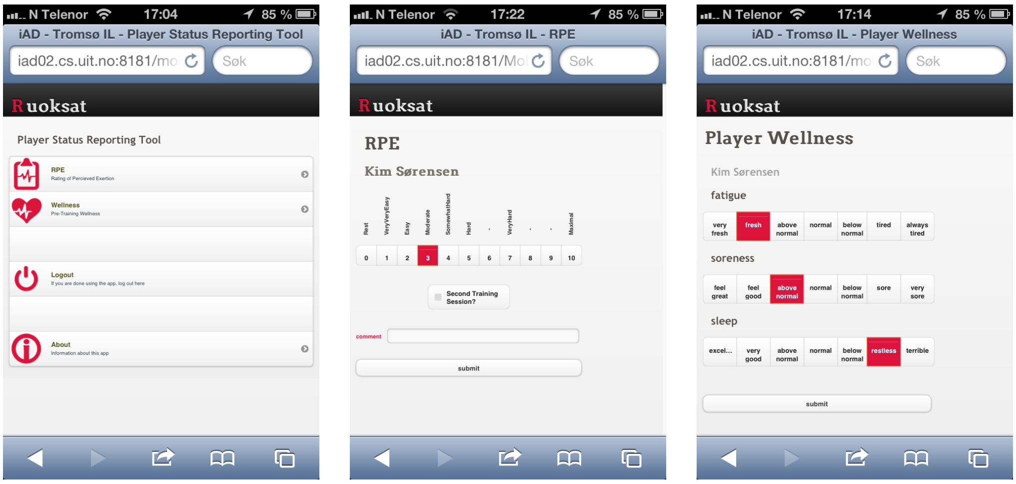
5.3.2.1 Reporting Tool

The user interface of the reporting tool used by the athletes to register RPE and PTW values is implemented as a web application adapted to support most modern smartphones. We have chosen to implement the interface as a web application as opposed to a stand-alone, platform specific, application in order to achieve easy deployment and a broad platform support.

After logging in to the interface, the user is presented with a set of choices in the main menu, as shown in Figure 5.19(a). Here the user can chose to report the RPE value after a training session, or submit a PTW report in the morning before practice. In the interface described at the prototyping phase, the user could switch between these to reports and the home page at any time through a static navigation menu in the top of the layout. This was discarded as the usage pattern showed that there were no situations where the player would register both reports in the same session, as they need to be reported at different times during the day.

When the users chooses to submit a post-workout RPE report (Figure 5.19(b)), they are met with a row of buttons, where each button represents a value between 1 and 10, along with a set of words describing the exertion intensity represented by the numerical values. The sliding bar in the visual prototypes(Figure 5.8) was replaced by this solution, as it is easier to interact with on smaller screens, improving the usability of the interface. The date picker was left out in the final implementation, as reports should be delivered within 30 minutes after a training session[32]. When discussing the functionality of the tool with Tromsø IL we also came to the conclusion that the duration of each exercise should not be reported by the athlete, but registered post-workout by the medical staff, as this simplifies the reporting process.

Through the testing phases of the interface, the athletes and the main staff member at the medical team at Tromsø IL discovered some desirable features that were missing. The first feature was the ability to add an additional RPE report, for days where the athletes had multiple workout sessions. Secondly, the athletes requested the ability to leave a comment alongside the registered values.



(a) Player Health Recording Home (b) Player Health Recording RPE (c) Player Health Recording Wellness

Figure 5.19: The player RPE and wellness reporting tool, optimized for smartphones

The interface for the PTW reports (Figure 5.19(c)) is constructed in similar fashion, except containing three different categories to report. As in [23], each value for the PTW is represented by a textual description, instead of a numeric value. The user simply selects the values corresponding to their physical status, and submits the form. When first describing the interface, it contained a set of five different PTW values to register, but two categories where removed due to issues related to storing personal data.

5.3.2.2 Athlete Status-Monitoring Interface

The athlete status-monitoring interface presents the data reported by the athletes, and is implemented as a web interface for the same reasons as the reporting tool. It enables the medical staff to inspect the reports from any device running a modern web browser, and presents the data in three different views; an overall team status average, an overview of each individual athlete, and an athlete profile view.

The view displaying the team average contains fatigue and RPE values for the entire team, along with a graph displaying the historical averages for either the last week or the last 30 days.

In the overview (depicted in Figure 5.20) the user is presented with a

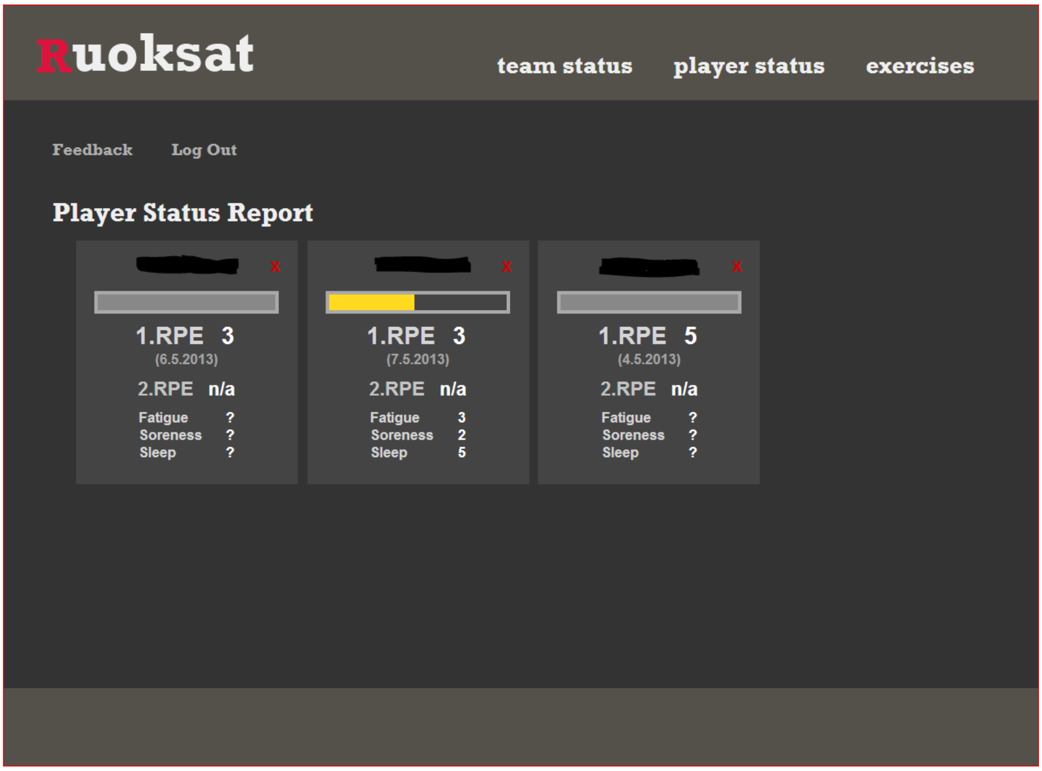


Figure 5.20: Layout: Athlete status overview for medical staff

grid of all the athletes from the team and the last reports submitted. In the figure only three players displayed, and since this is real data from operational use, the names have been been removed). It contains the name of the athlete, a colored bar indicating the average PTW values reported, and the RPE values from the most recent workout session.

The colored bar works as an easily identifiable indicator of the reported wellness of the player. The average value of the wellness report is used to decide the coloration of the bar. Through a discussion with the medical staff at TIL, we decided to display everything at normal or above from the PTW rating (Figure 2.6) as green, indicating that the player is feeling well. Scores on the lowest en of the scale are represented by a red color, which indicates that the player is not feeling well, and rest should be considered. Values between these are displayed with a yellow colored bar, as this may need further investigation, or it could that the athlete is simply having a bad day [23].

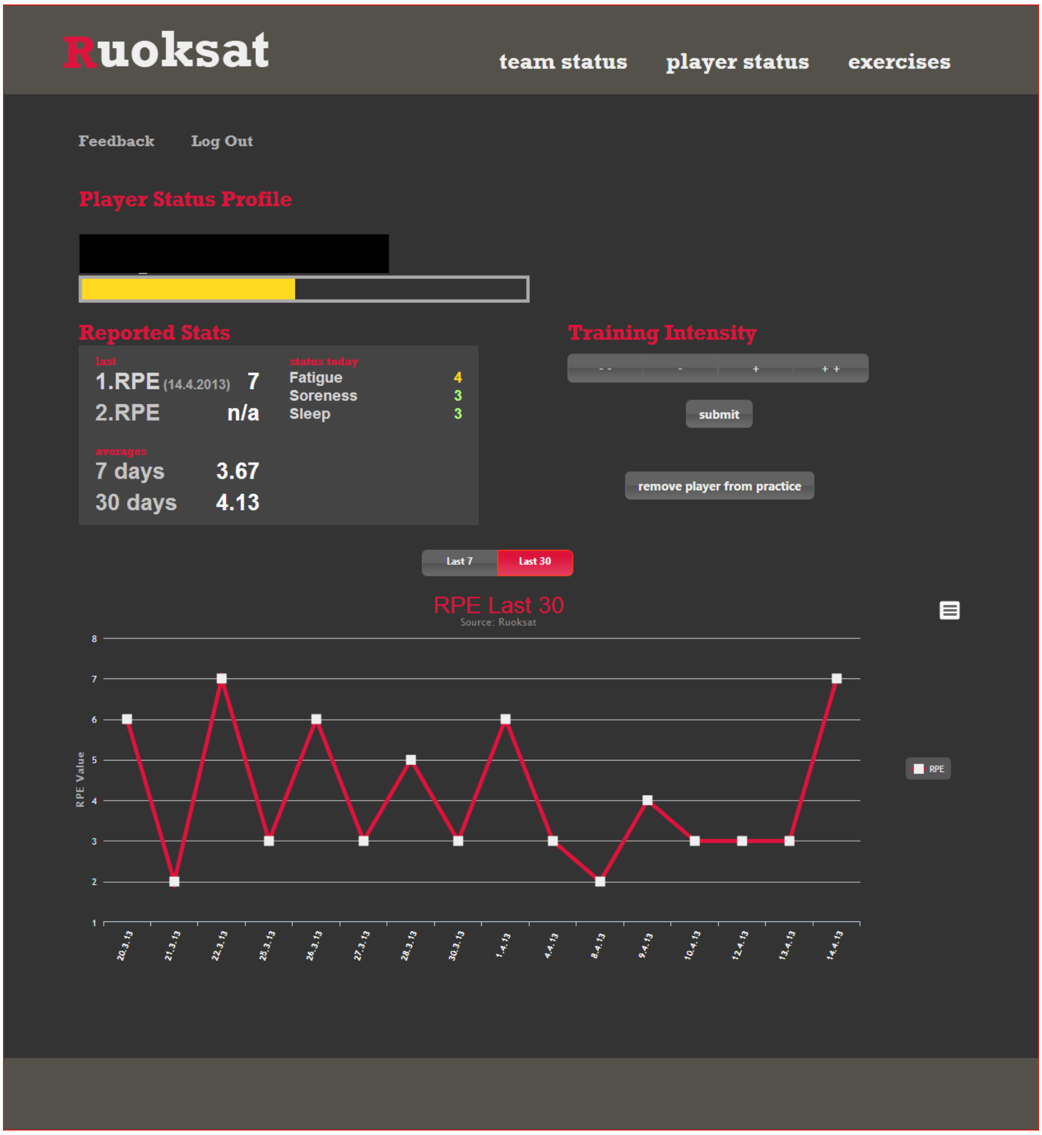


Figure 5.21: Layout: Athlete profile for medical staff

The profile picture presented in the mockups was omitted in the implemented interface, as the medical staff did not need it to aid in the identification of the athletes, and would take up valuable real estate on the page. Another change from the prototyping phase is that the interface now enables the medical staff to view a comment related to the RPE report, by clicking the RPE value in the layout.

Lastly, the user has access to a profile view for each athlete (Figure 5.21).

This view is comprised by several elements. A set of the last reported RPE and PTW values submitted by the athlete, along with the average RPE value for the last 7 and last 30 days. The same colored bar as in the overview, indicating the general wellness of the athlete, is also available in this view of the interface. Through feedback from the end-users of the system the need for historical data was expressed. A chart displaying historical RPE values is therefore presented in the profile view, enabling the user to switch between viewing data from the last 7 or last 30 days through a radio button.

A crucial part of the Ruoksat system is the ability for the medical staff to pass on a report, based on analysis of the athlete status data, to the coaching staff responsible for exercise planning. This is enabled by presenting a set of controls for suggesting a change in training intensity, or indicating that the athlete should rest during the next training session. The intensity change was initially indicated in percentages, but after a discussion with the end-users it was discovered that this would be too finely grained. Thus, the final interface allows the user to indicate a change in intensity in a more coarse grained manner. If a user reports any intensity change, this information is passed on to the exercise planning service, and is persisted in the storage layer.

Discussion between individual athletes and the medical staff is replaced by a simple comment related to the last report, as more detailed discussions would be conducted in a face to face meeting with a player, and not through the Ruoksat system.

5.4 System Implementation

5.4.1 Storage

In our requirement specification we outlined a need for two different data stores for the Ruoksat system; one for storing video files and one for structured data related to exercises and athlete status reports.

Based on the work by Hansen in [14] we have chosen to use OpenStack Object Storage for our video storage needs. He found that the consistency model and focus on unstructured binary content, makes OpenStack suitable for storing and delivering video files to multimedia applications. The storage solution was also currently deployed and available to us at the time of implementation. The video files used in the Muithu system, which was the main case study in the analysis performed by Hansen, are very similar in resolution and size as the ones suitable for the Ruoksat system. Additionally, OpenStack provides us with eventual consistency while being fault tolerant, and supports adding storage nodes on demand, with no master node, rendering a highly scalable data store [10].

For storing our structured data we have chosen to design the system using MongoDB as our document database, as it is suitable for storing enormous amounts of records and allows for aggregation and analytics through a Map-Reduce framework [30]. This makes it ideal for our RPE and PTW reports, where there are several records inserted every day and future analysis of historical data is highly likely. MongoDB is also a good match for our structured data related to exercises due to the support for embedding documents inside other documents [30]. This allows us to create one-to-many relationships between a weekly plan for exercises and each exercise included in the plan. Through this embedding scheme we can fetch the entire set of exercises with only one query.

During the development of this system a regular SQL database has been used as a temporary solution for the prototypes, as the planned iAD MongoDB storage instance has not yet been deployed. Data related to exercises and athlete reports is stored in single tables, emulating MongoDB documents, and is organized to be directly portable to a MongoDB instance.

In order for us to be able to store the athlete status reports in our system, we have had consent forms signed by all players at Tromsø IL.

5.4.2 Implementation Details

The implementation of the system utilizes several of the same software components as the Muithu system [18]. The reason for this is that Muithu is currently in operational use, and the majority of the players and coaches from Tromsø IL already have a registered account there. Sharing the Muithu database makes it more convenient for the users, as we minimize the number of different accounts they have to manage and keep track of. The software stack used when implementing the system is presented in Table 6.1.

Since Muithu does not expose any kind of external authentication methods, and the fact that motivaional events of the exercises are fetched from Muithu,

Server Operating System	Windows Server R2 Standard
Web Server	IIS 7.5
Database	MongoDB
Video Storage	OpenStack Object Storage
Web Interface	Razor Syntax (C#, HTML5, CSS3)
Tablet Interface	Windows Store apps (C#, XAML)
Back-End Logic	ASP.NET (C#)
Client Side Scripting	Javascript (jQuery, AJAX)

Table 5.1: Software Stack

we have chosen to implement the web interfaces and the services on the same web server instance as Muithu. This means that we are using Windows server as our operating system, and that IIS 7.5 is used as our web server platform. In addition to the SQL database used from the Muithu system for user authentication, we have designed the system to be implemented using MongoDB for structured storage. However, as explained in the previous section, we have implemented the prototype using a Microsoft SQL server solution for storing structured data. Based on the storage system analysis by Hansen [14], we are using the OpenStack Object Storage for managing video files in the system.

5.4.2.1 Services and Web Interfaces

With the underlying server architecture being comprised of Microsoft software, we have implemented the web interfaces using ASP.NET, utilizing the Razor View Engine[1], which allows combining HyperText Markup Language (HTML) with server code written in C#. This is advantageous for when developing systems with usability in mind, as it narrows the gap between the interface and the back-end logic.

The services used to connect the web interfaces to the back-end storage is either implemented directly into the web interface using the aforementioned Razor syntax, or is included in the same web server instance as a stand-alone ASP.NET web service.

[1] http://www.asp.net/web-pages/tutorials/basics/2-introduction-to-asp-net-web-programming-using-the-razor-syntax

5.4.2.2 Client-Side Scripting

When it comes to interactive web interfaces, client side scripting is often useful to facilitate simple patterns for interacting with the interface. In our web interfaces, we use Javascript code for implementing this client side scripting. JQuery Mobile[2] is used to style and handle interactions in the mobile web application for sending athlete status reports, and the charts showing historical data related to the RPE reports are generated using a Javascript library called Highcharts[3]. Asynchronous calls from the web interfaces to the back-end services are performed using Asynchronous JavaScript and XML (AJAX)[4].

5.4.2.3 Exercise Presenter App

The application for presenting exercises is implemented on the Windows Store App-platform. The main reasoning for this is that with the launch of the platform for Windows 8 apps, Microsoft released a clearly defined set of User Experience (UX) guidelines [26]. These guidelines focus on user-centered design, and adhere to most of the HCI principles presented in Section 2.1. By following best practices highlighted by Microsofts UX guidelines, we know that we are utilizing techniques and methods that are well documented and tested, and have been proven to render user-friendly interfaces.

With Windows Store apps we can use the same programming language as in the back-end services (C#), allowing us to create identical data structures, which simplifies the serialization and de-serialization of object instances in the system.

The Model View ViewModel (MVVM) pattern is used when developing the application, as allows for very flexible interfaces without compromising the business logic. The pattern also facilitates implementation of code that is highly maintainable and easy to comprehend.

5.4.2.4 Authentication

For authenticating users in the athlete status application and the exercise planner, we are using the same user database and methods as Muithu, which

[2] http://jquery.com/
[3] http://www.highcharts.com
[4] http://api.jquery.com/jQuery.ajax/

is a basic authentication using the WebSecurity Object[5] from ASP.NET, implemented using a WebSecurity Database.

The exercise presenter prototype does not currently offer any form of authentication, as it is still in the early stages of development. In the future we wish to utilize Codecaps[39] for discretionary access control. Codecaps is a project currently under development at the iAD-center, where capabilities containing code allow for the delegated set of rights to not be predefined, and instead evolve as needed[6].

5.4.3 Exercise Knowledge Library

The system components of the exercise are presented in Figure 5.22. The abstract flow of the system adheres to the following steps. The exercises and exercise plans are created and stored in the video storage and the structured database (1 and 2). Then, when a coach wishes to present an exercise, the data is fetched from the storage layer and populated in the presenter application (3 and 4). Finally, the presenter can forward the video to a big screen interface (5, 6 and 7).

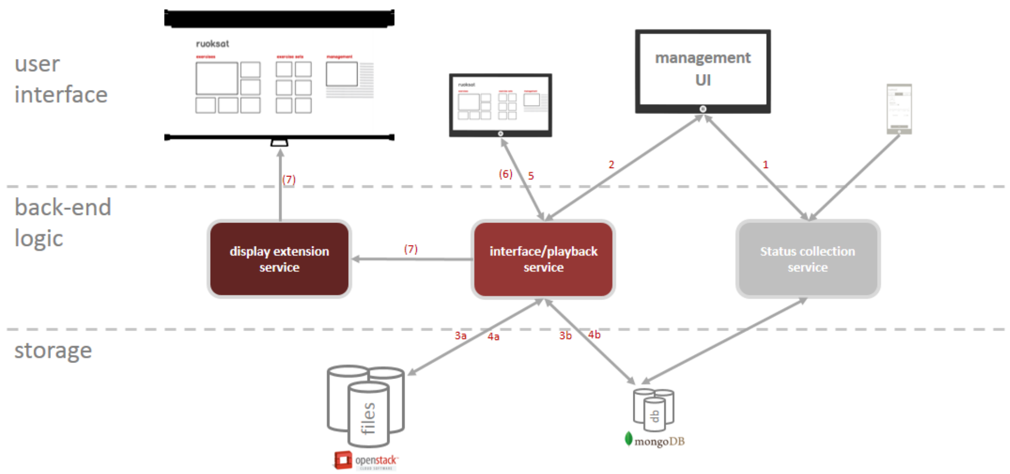


Figure 5.22: Exercise library system architecture

[5] `http://www.w3schools.com/aspnet/webpages_ref_websecurity.asp`
[6] http://site.uit.no/iad/cloud/codecaps/

5.4.3.1 Capture

The interface for capturing exercises is basically a web form. When the user has filled out the form, and chosen the video files for the exercise, the exercise planning service parses the submitted form. When the data is parsed, the video files are uploaded through the REST API exposed by OpenStack [14], and stored as three replicas in different nodes. The structured data describing the exercise is then persisted in the database, with videos being represented by a GUID referring to their location in the video storage system.

5.4.3.2 Plan

When a user is planning a week of exercises, each day is planned individually. We expose a web form for each day, where the user selects from drop down menus the exercises to include, along with the duration of the exercise. If the medical staff has suggested intensity adjustments they will be made visible to the user when planning exercises. As in the capturing process, the service parses the form and persists the exercise plan in the document storage. As explained in the storage section, each exercise included in the plan is embedded into the week plan document, allowing us to fetch the entire exercise plan in just one query.

5.4.3.3 Present

Presentation of an exercise is done through the Windows Store app. When the application is launched, a request is sent to a web service component of the exercise planning service, which returns a JSON object that can be parsed into an object representing a week plan of exercises. The exercises can then be navigated to by using the navigation pattern described in section 5.3.1.3.

```
{
"WeekPlan":{
        "MondayDuration":"25:45:25",
        "MondayThee":"Passing",
        "Monday":{
                "ExerciseData":{
                        "ID":42,
                        "Title":"Diamond Passing",
                        "Stage":2,
                        "VideoLink":"URI",
                        "Image":"URI",
                        "PlayersToBeRested":[PlayerNames],
                        "PubDate":"DATE",
                        "Info":{
                                "Duration":"25 min",
                                "Description":"TEXT"
                                "Participants":"8-10",
                                "Area":"10mx10m"
                        },
                        "ExerciseDetails":{
                                "Title":"Accurate Passing",
                                "Description":"TEXT",
                                "VideoLink":"URI",
                        },


                        "Motivation":{
                                "Match":"TIL - RBK",
                                "Event":"Offensive Passing",
                                "Description":"TEXT"
                                "CaptureTime":"DATE",
                                "VideoLink":"URI",

                        },

                }
        }
}
```

Listing 5.1: Simplified version of the exercise JSON object

The first time a video is requested from the storage system it is persisted until the application is closed. The reason for this is that the exercise videos are often displayed multiple times during one exercise, making sure all participants have understood the concepts and techniques to apply. Additionally, since the presenter app is running on a tablet device, it may be outside wireless network coverage outside on the field, and persisting videos allows the users to download the videos before starting the training session.

5.4.4 Athlete Status Application

The system architecture and components for collecting, storing and presenting the reports, are shown in Figure 5.23. The system flow is divided into the following stages. First, the report is registered and submitted by the athlete, from a smartphone device (1). Going through the status collection service, the data is stored in the database (2), before being made available to the user interface for the medical staff (3 and 4).

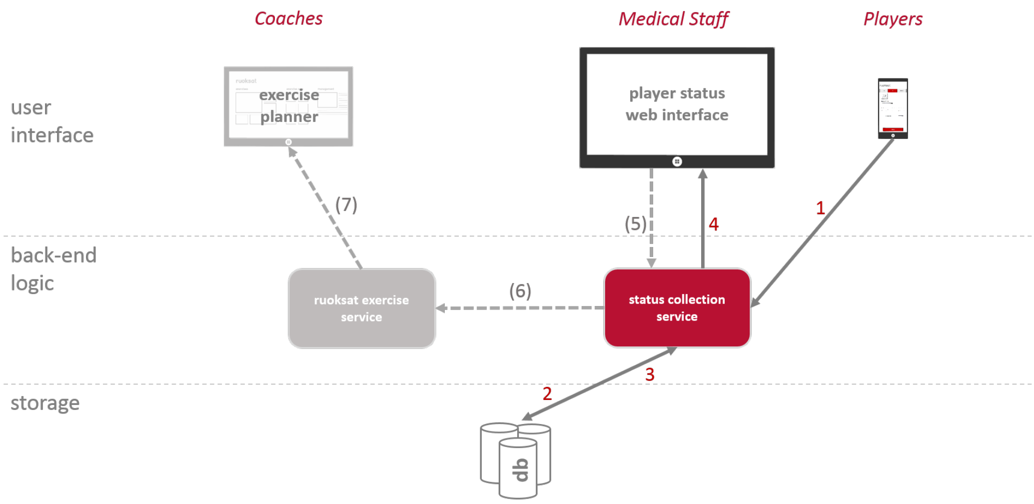


Figure 5.23: Athlete status system architecture

5.4.4.1 Reporting Tool

The smartphone adapted interface for submitting RPE and PTW reports is implemented using a combination of HTML forms and JQuery Mobile. JQuery UI allows styling the forms to suit touch interaction on a small display. When the user submits a report, it is sent to the corresponding service, which parses the report and inserts it into the document database.

5.4.4.2 Athlete Status-Monitoring Interface

In the overview section of the athlete status interface, all participating athletes are listed in a grid view. The data related to the players is fetched from the Muithu-database, and the data from Ruoksat is coordinated with the player information through the player status service, based on a player ID stored in the reports.

The profile view interface for specific athletes fetches the last registered reports from the document database along with some historical data through a web service request. When the interface has received the data, it draws a chart by plotting the historical data received, and displays the PTW report in the form of a colored bar indicating the level of fatigue reported by the athlete. If the medical staff wishes to either suggest a change in training intensity, or completely remove the player from the next training session, a dedicated record is inserted in the document database. This record is then forwarded to the exercise planning service to be exposed in the interfaces for planning and presenting exercises.

5.5 Summary

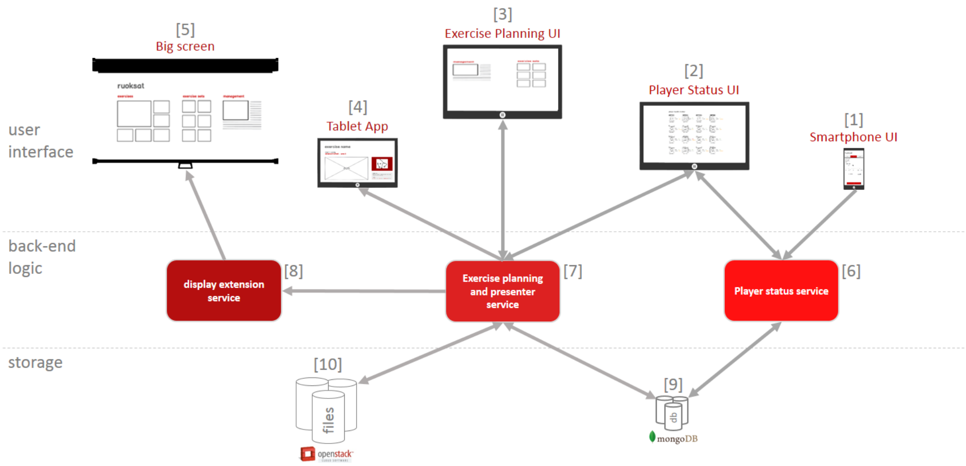


Figure 5.24: Arhcitecture: Complete System

In this chapter we have presented the complete Ruoksat architecture, along with the visual prototypes created before the corresponding interfaces were implemented. We described the implementation details, and presented the software used in the implemented system.

Chapter 6

Evaluation and Results

This chapter presents the methods used to evaluate the system, and the results gathered through the evaluation. In the evaluation process we have both conducted user surveys and some performance experiments.

6.1 Methods

6.1.1 User Survey

The Ruoksat system is designed using a user-centered design methodology and the development process has been driven by user feedback. Through the previous chapters of this document, we have described how everything from the requirements to the system implementation has been guided by feedback from both players and staff members at Tromsø IL. Since the system has been developed for, and through continuous dialogue with, the end-users, the most suitable way to evaluate the success of the project is by performing user surveys for measuring the end-user experience. Additionally, since the athlete status reporting application has been in operational use for two months, the users have a good foundation for assessing the system.

We have chosen to perform the user surveys using a set of question rated on a Likert-scale, as this is suitable for measuring user experiences[1]. The surveys are divided into two different sets of questions. The first set compares functionality of the Ruoksat system to the tools currently available to the staff and players at Tromsø IL, using a pairwise comparison method [36]. In the second section of the user surveys, the assessors are requested to rate HCI related aspects of the interfaces, rendering an assessment of the

[1] http://en.wikipedia.org/wiki/Likert-Scale

perceived end-user experience. The Likert-scale chosen is a five-point scale rating from *Poor*(1) to *Excellent*(5).

6.1.1.1 Assessors

The assessors participating in the evaluation are divided into four different groups. The medical staff and players from our collaborative partner Tromsø IL have been using the athlete reporting component of the system for two months, making them qualified to evaluate both the functionality and the user experience aspect of the system. Additionally, the reporting system is developed in direct collaboration with the medical staff, and almost all of the features are implemented upon their request. The three members of the medical staff have evaluated all aspects of the application, and the three players currently using the mobile reporting tool have rated only that element of the system.

The evaluation of the exercise components of the Ruoksat system is performed by showcasing the system for two different sets of potential users. The first group includes four coaches from the soccer youth academys in Tromsø, and another comprised by five faculty members connected to the Tromsø IL senior first team, including a specialist team coach, the head coach and the current CEO. This allows us to evaluate the system concerning both professional top-level soccer, and the posibility and benefit of early establishment of a specific style-of-play in younger teams.

6.1.2 Performance Experiments

In order to evaluate some of the non-functional requirements of the system we have conducted a small set of performance experiments on the response times of the systems most content heavy interfaces. The response times of user interfaces are directly linked to usability of the underlying applications [40]. According to [27], a user interface response times below 1000 milliseconds is fast enough for the thought-flow of the user to stay uninterrupted. Even though the users may still notice the delay, they do not lose the perception of operating directly on the data.

We have measured the response times of the three interfaces with the either the largest presented footprint or the most content rich pages to assess the performance of the user interaction with the system. The first response time we measured was the time latency of loading a video stream in the Ruoksat presenter app. Having the application connected to a Wi-Fi network, we measured the delay between navigating to a page containing a video and the video stream being available to the user. The measurements where performed by implementing a timer utility inside the application.

In addition we measured the response times of the athlete status player profile (Figure 5.21) and the exercise planning web interface (Figure 5.11(b)) using the network monitor in Google Chrome DevTools[2].

6.2 Experiments and Results

This section gives a short presentation of the results gathered, and a corresponding analysis for each presented result chart. The baselines used as a comparison in the user surveys are the tools currently in use by the assessors. An example of such tools and systems is given for each question in the user survey questionnaire. All questionnaires are available in the appendix of this document. The error bars presented in the plotted charts indicate the lowest and highest registered ratings. The Likert-scale used is a five point system with the following mapping of each rating

- 1 - Poor
- 2 OK
- 3 Good
- 4 Very Good
- 5 Excellent

6.2.1 Exercise Knowledge Library

There were a total of 9 assessors participating in the user survey of the exercise knowledge library. The participants are divided into two different classifications, one set of 4 junior coaches, and 5 assessors associated with the Tromsø IL senior first team, including the head coach and the CEO.

Ruoksat as a planning tool

The assessors in the user survey were asked to evaluate the exercise planning aspect of the Ruoksat system, along with the resources they have currently

[2]https://developers.google.com/chrome-developer-tools/?hl=no

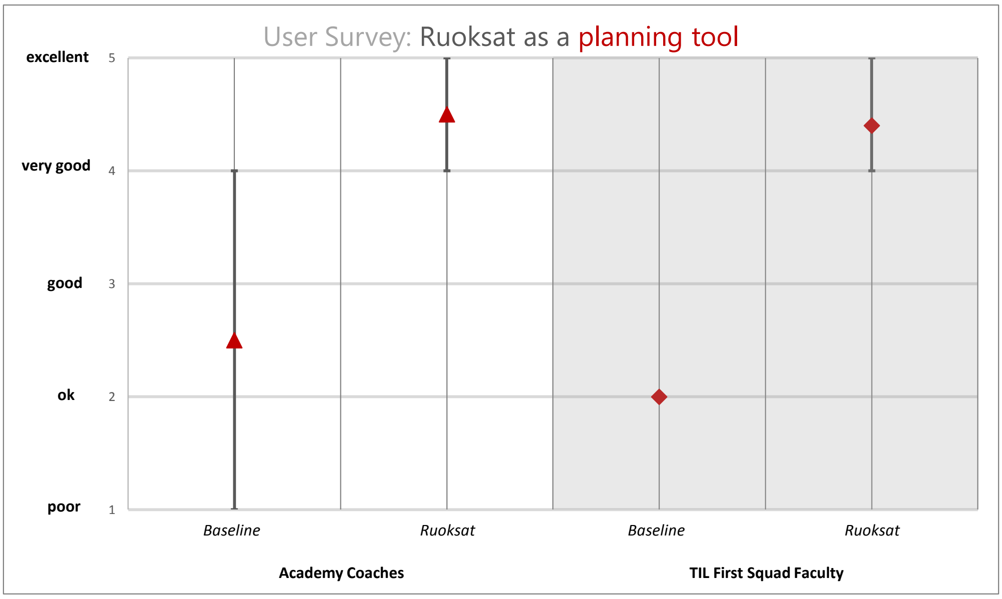


Figure 6.1: User Survey: Ruoksat as a planning tool

have available, i.e. pen and paper, calendars or other planning tools. Figure 6.1 shows the results gathered from the survey. Here we see that the youth academy coaches rate the baseline between, OK and Good, where some users indicate that the tools currently are Very Good and others rating it as Poor. The assessors from Tromsø ILs first team are unanimous in rating the baseline at OK. In both participant groups, using Ruoksat for exercise planning is rated between Very Good and Excellent, indicating that all assessors believe that the Ruoksat system for exercise planning would be an improvement to the current solutions.

Ruoksat as a presentation tool

In the second question set in the user survey, the users were asked to rate the baseline and the Ruoksat system as tools for presenting exercises. The baseline in this question could be pen and paper, overhead presentations or other similar resources. As Figure 6.2 shows, the results from both assessor groups are nearly identical. Rating the baseline between OK and Poor, whilst giving the Ruoksat system a rating close to Excellent. The variances in the answers are minimal. From these results we can conclude that the end-users view the Ruoksat system as a vast improvement to tools used today for presenting exercises.

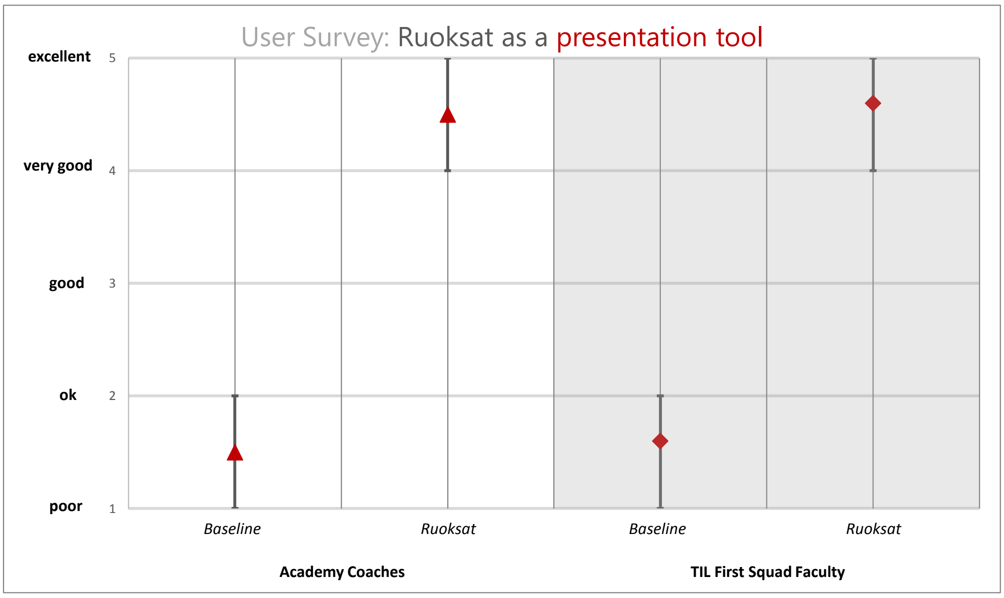


Figure 6.2: User Survey: Ruoksat as a presentation tool

Ruoksat as a coaching tool

Figure 6.3 illustrates the evalution of the Ruoksat system as a tool for coaching players during exercise execution. The baseline in this question could be flip-overs, portable white-boards or similar resources. The results from this rating show a high degree of variance, but all assessors rate the Ruoksat system as an improvement compared to other tools they are currently using.

Ruoksat exercise content

In the last pairwise comparison of the Ruoksat Exercise library we asked the users to compare the exercise data they currently have available from different sources (i.e. YouTube, Secrets2Sports[3]), to the possibility of having exercises from the Tromsø IL senior first team available. The results plotted in Figure 6.4 indicate that both groups participating in the survey would rate the potential Ruoksat exercise content Very Good to Excellent as clear improvement to the exercise database they currently have access to (OK on average).

[3]https://no-fotball.s2s.net/home.php

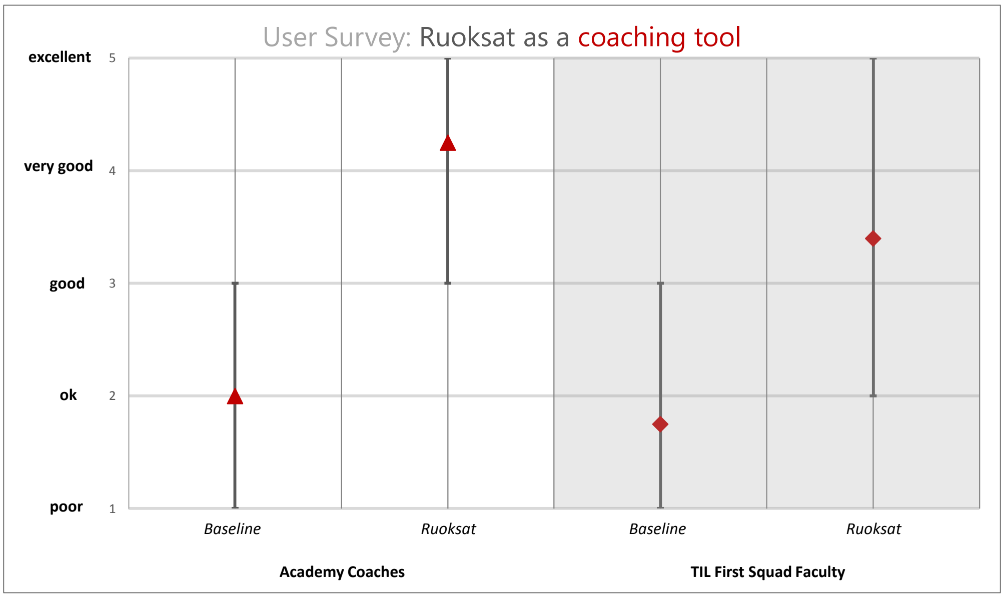


Figure 6.3: User Survey: Ruoksat as a coaching tool

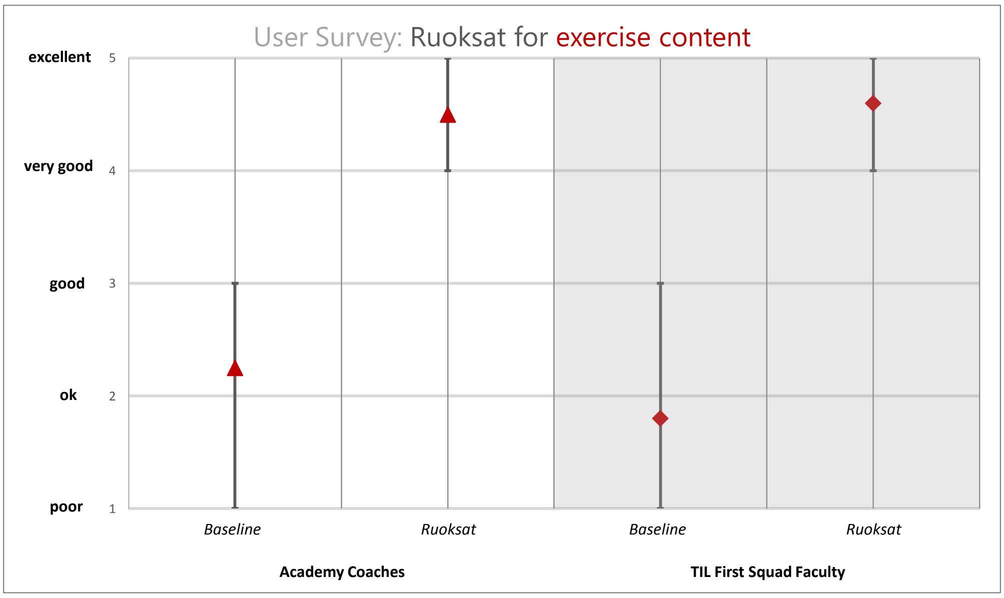


Figure 6.4: User Survey: Exercises from Ruoksat

Ruoksat Exercise Library User Experience

Finally, in the last section of the questionnaire the assessors are asked to rate the user experience of the system interfaces, based on four different

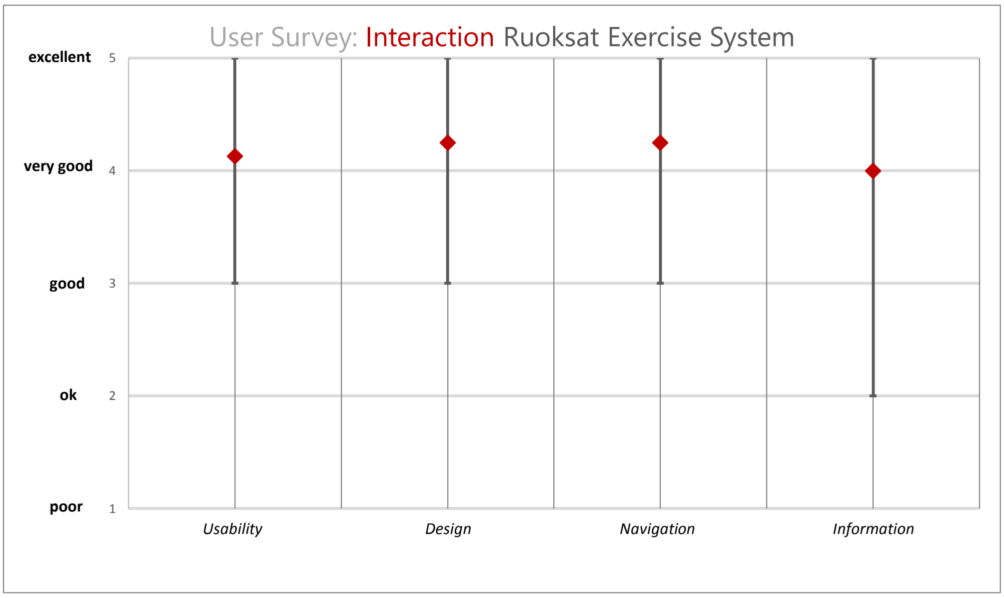


Figure 6.5: User Survey: Interaction design of Ruoksat exercise interfaces

categories: *Usability*, *Design*, *How easy it is to navigate* and *The presentation of information*. This is an important rating related to the thesis, as our main focus throughout this project have been adapting to the users in order to maximize usability and create a good user experience. The results are presented in Figure 6.5, and show that the average rating of all categories are above Very Good, indicating that we have succeeded in creating user-friendly interfaces.

6.3 Athlete Status Application

In our second user survey, we asked the medical staff at Tromsø IL to evaluate the athlete status application component of Ruoksat system. This part of the system has been in operational use for two months, with three players in a test group delivering daily Rating of Perceived Exertion (RPE) and Pre-Training Wellness (PTW) reports. The baseline in this questionnaire is the methods currently in place for collecting and evaluating these reports, which is handing out and collecting paper forms filled out by the players. With the current solution, the PTW reports are collected directly before the training session, leaving little or no time for the medical staff to analyze them before the practice begins. The assessors in this user survey are the 3 members of Tromsø ILs medical staff involved in collecting and analyzing these status reports.

Collecting Reports

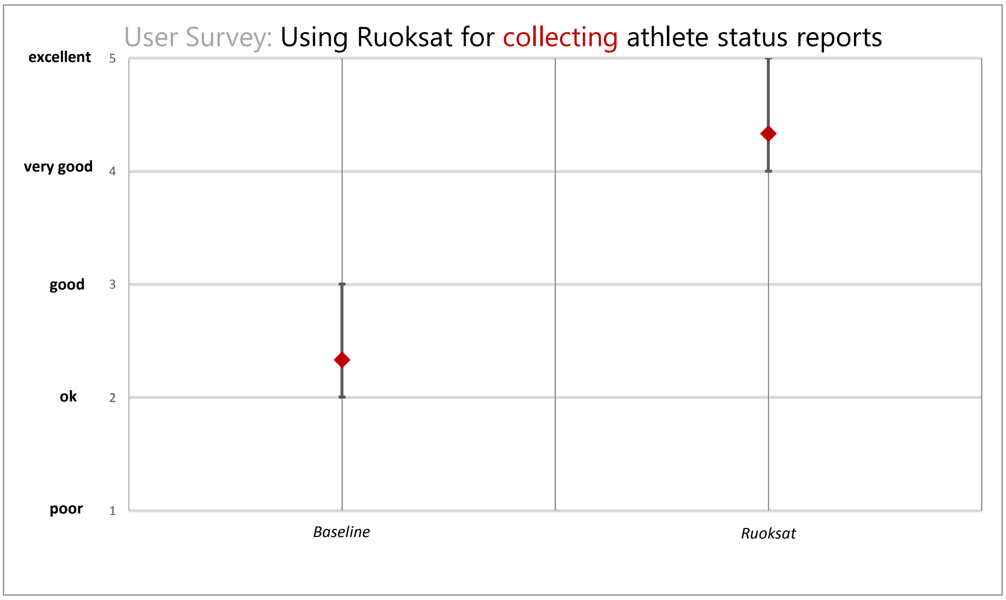


Figure 6.6: User Survey: Using Ruoksat for collecting athlete status reports

Figure 6.6 show the results of the comparison of the Ruoksat system to the baseline. Here we see a significant improvement from the baseline being rated slightly above OK, and the Ruoksat system rated as Excellent for collecting status reports from the athletes.

Archiving Reports

Todays solution for archiving the status reports means placing the paper forms somewhere the medical staff can find them at a later stage. Taking this into consideration, it is not surprising that we see a vast improvement from the rating of the baseline and the Ruoksat system in Figure 6.7. The ability to persist status reports and have historical data displayed upon request is crucial for long-time monitoring of the status of the athletes.

Viewing Reports

In the third pairwise comparison of the questionnaire, the medical staff was asked to rate the ability of viewing the collected reports. The current solution

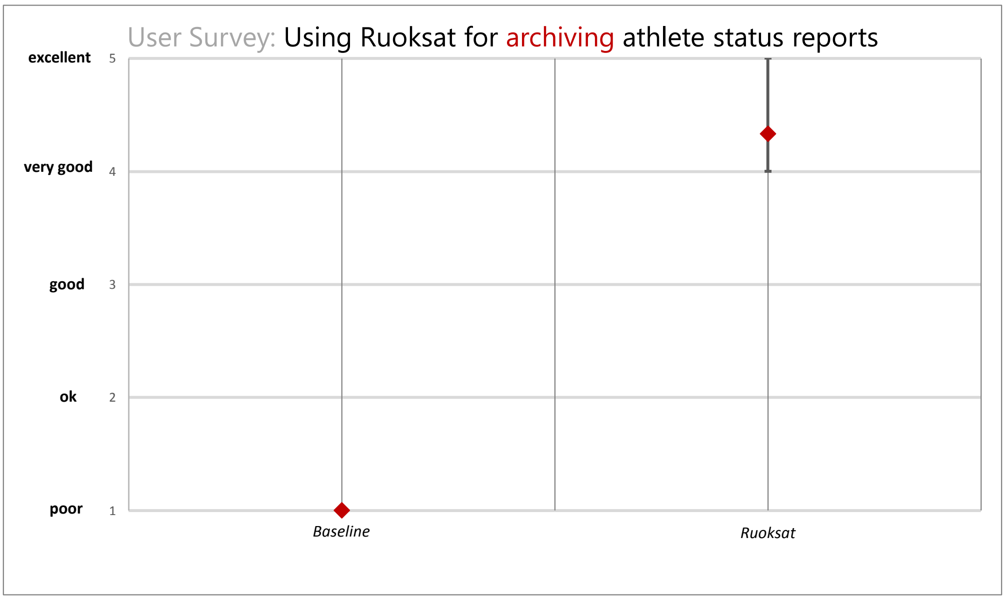


Figure 6.7: User Survey: Using Ruoksat for archiving athlete status reports

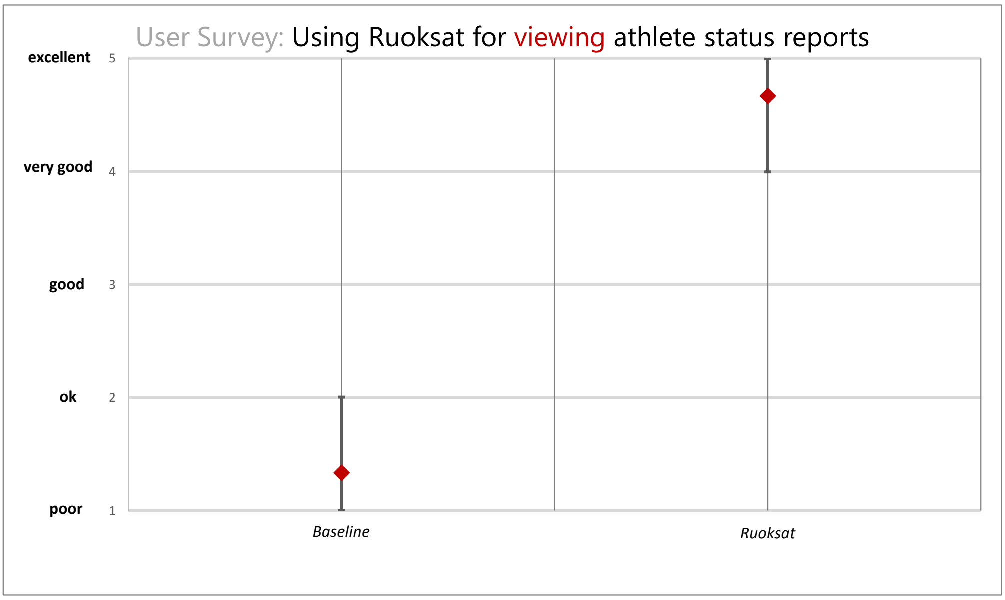


Figure 6.8: User Survey: Using Ruoksat for viewing athlete status reports

for this is reading through the last paper forms delivered by the athletes. The plot in Figure 6.8 show that the assessors rated the previous solution as Poor, whilst we see a considerable improvement when it comes to the content presentation provided by the Ruoksat system, which was rated close to Excellent.

Forwarding Reports to Coaches

In the last comparison in the user survey, the assessors gave a rating of report forwarding (Figure 6.9). More specifically, how easy it is to let the team coaches know if they have any concerns regarding player fatigue or exertion. Even though we see a noticeable improvement from the baseline to the Ruoksat rating, the improvement is still not quite as significant as with the other features of the system. We speculate this is because the receiving component of these forwarded reports is not yet in use by the Tromsø IL coaches, so there is no real way for the medical staff to assess the effectiveness of this feature.

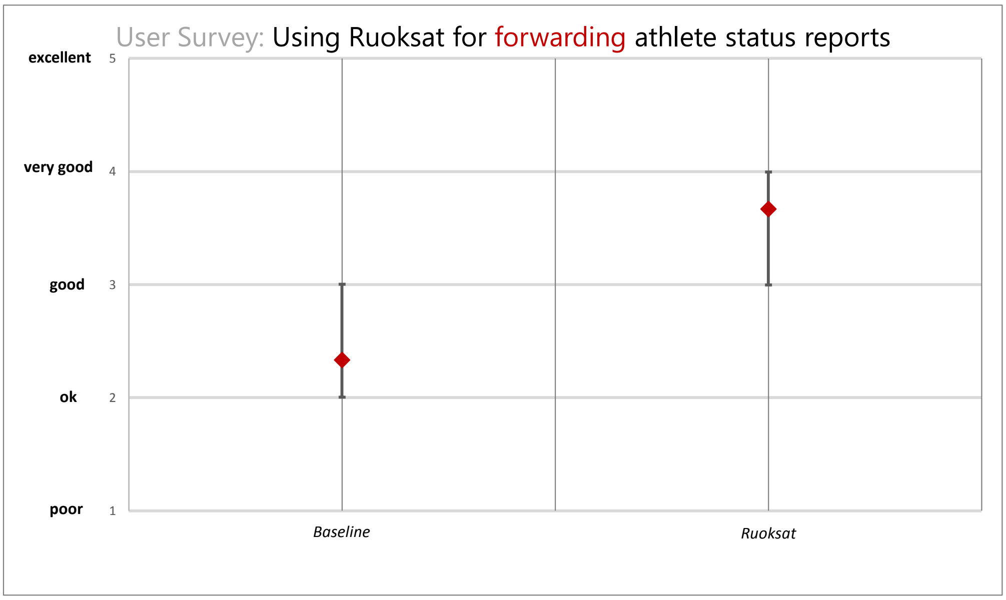


Figure 6.9: User Survey: Using Ruoksat for forwarding athlete status reports

Athlete Status Application - User Experience

As with the exercise component of the system, usability and user experience has been the main focus when designing and developing the athlete status app. Figure 6.10 shows that this focus has paid off, and that all user experience elements are rated close to Very Good or above, further indicating

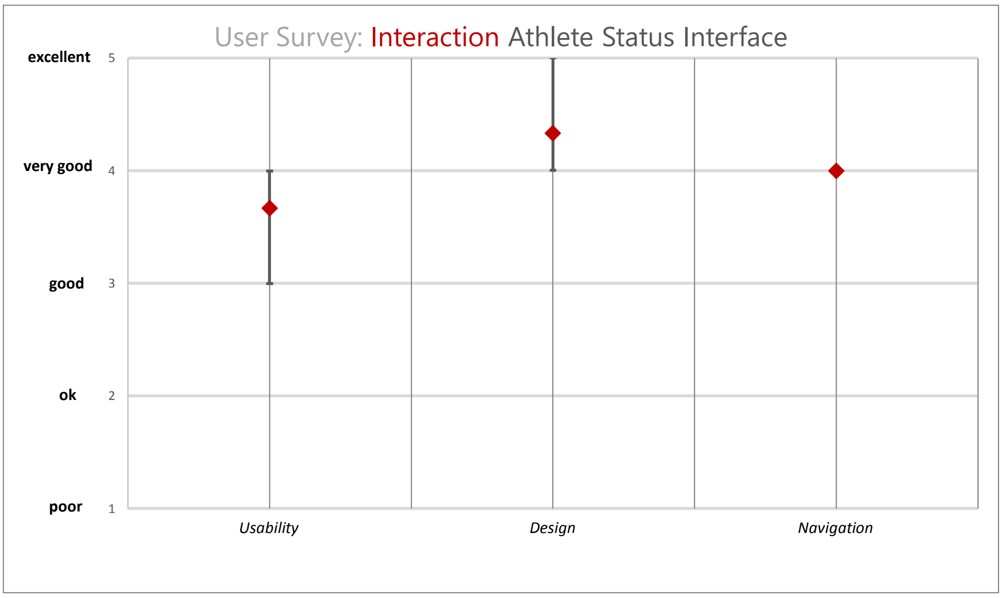


Figure 6.10: User Survey: Interaction design of athlete status web interface

that we have succeeded in creating a system which exposes user-friendly interfaces.

6.3.1 Athlete Status Reporting Tool

In order for the system to be useful for the medical staff and coaches, we need player participation. We have had three players from the first team using the system on a daily basis, and have asked them to rate the reporting tool compared to the pen and paper solution used previously.

Sending reports

In the first question, we asked the assessors to evaluate the process of sending RPE reports to the medical staff. Figures 6.11 and 6.12 show the results gathered from the questionnaire, both indicating a significant improvement from the baseline. This indicates that the players prefer the Ruoksat solution to the methods previously used by the medical staff.

Athlete Status Reporting Tool - User Experience

The user experience of the reporting tool is crucial for keeping the system in use, as a difficult and unorganized user interface could lead to the system not

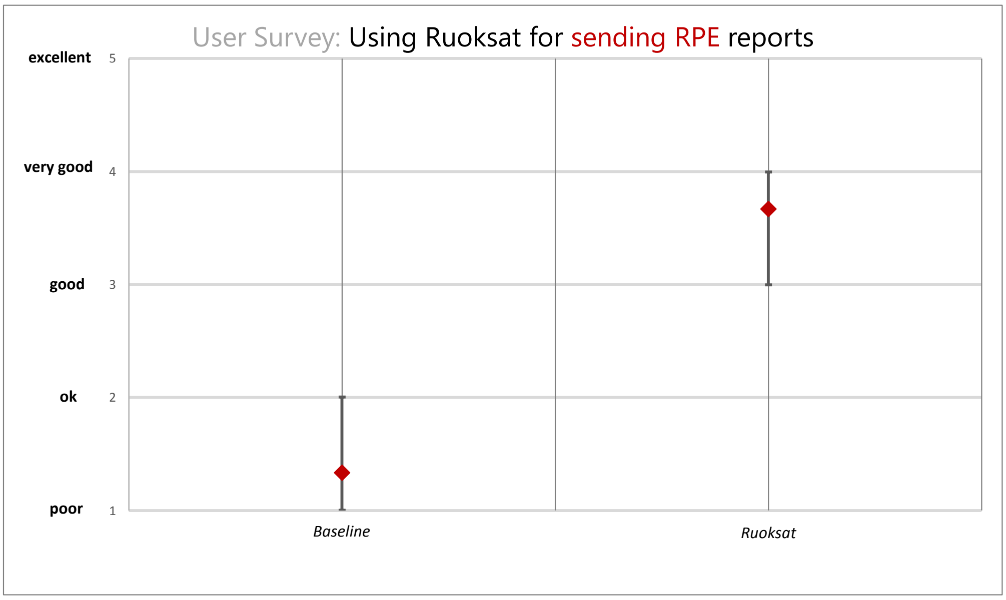


Figure 6.11: User Survey: Using Ruoksat for sending RPE reports

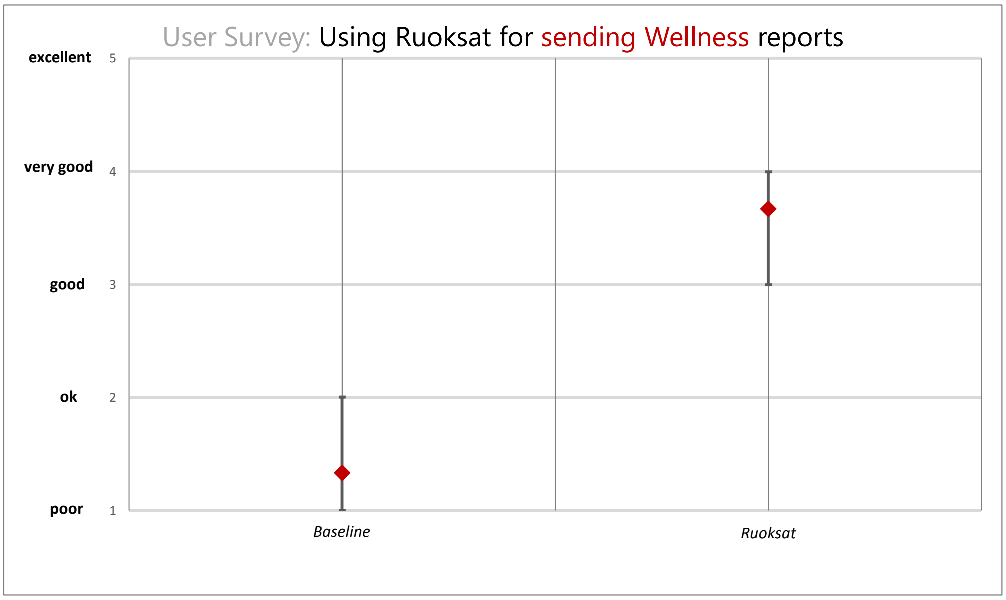


Figure 6.12: User Survey: Using Ruoksat for sending PTW reports

being used by the players, rendering the system useless. As with the other user surveys, we asked the players to rate the perceived user experience of the interfaces, and the results are presented Figure 6.13 . Again we see the

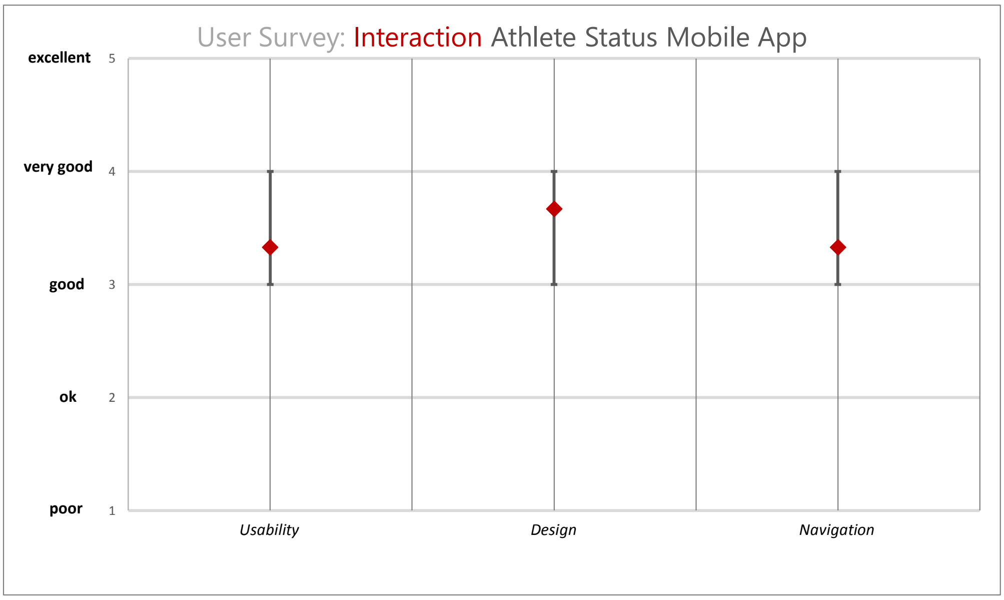


Figure 6.13: User Survey: Interaction design of status reporting tool

assessors rating the user experience as more than satisfactory.

6.3.2 UI Performance

In our performance evaluation, we measured the response time of the most content rich interfaces of the system, as these require more back-end computation and present more data to the end-user. We measured all the response times using 20 samples, and compared the results to the 1000 millisecond threshold for UI delay according to Nielsen in [27].

Interface	*Response avg (ms)*	*min (ms)*	*max (ms)*
Presenter Video Player	866	771	1041
Athlete Status Profile	426	282	670
Exercise Planner	820	691	1042

Table 6.1: UI Performance Measurements

As illustrated in Figure 6.14, the average response time for all the interfaces are well within the threshold, and even the longest measured delays are still within acceptable ranges. The videos in the presenter app is ready to be played after an average of 866 ms, with the slowest registered delay being just

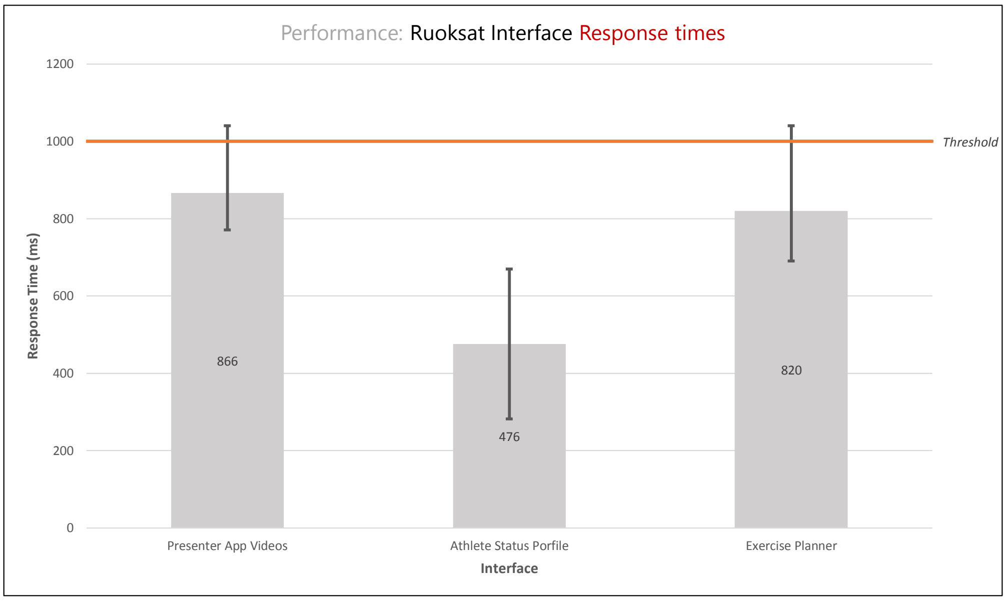


Figure 6.14: Measurements: Ruoksat interface response times

above 1 second. The measurements exercise planner web interface, which also has to stream content from the video storage, show similar results. Finally, the response time of the athlete status profile interface is well within the threshold for seamless user interaction, recording an average of 476 ms. Based on these results we can conclude that the performance of the implemented interfaces further improves the usability of the system, and renders responsive interaction, hence improving the user experience.

6.3.3 Result Summary

All results from the user survey regarding the Ruoksat exercise component were positive compared to the baseline. Both groups of assessors rated the Ruoksat system as an improvement compared to currently available tools, with regards to planning, presentation, coaching and the content of exercises. Additionally, the participants rated the user experience between Very Good and Excellent, indicating that our focus on usability has been a success.

In the evaluation of the athlete status application, we saw a significant improvement compared to the baseline when using the Ruoksat system for collecting, archiving and viewing athlete status reports. The only aspect where the baseline was rated as comparable to our system was in forwarding reports after analysis. We believe that this may be because the receiving application of these reports were not in operational, making it difficult for the assessors to see the possible benefits of the solution. This questionnaire also requested the participants to rate the user experience of the application, and all ratings were positive.

The third user survey was related to the athlete status reporting tool. Here feedback from the players stated that using the Ruoksat system was a significant improvement to the pen and paper solution used previously. The user experience of reporting tool was rated between Good and Very Good by all assessors.

Finally, we performed some performance measurements of the most content rich interfaces in the system. We found that all of the interfaces had response times well within the threshold of 1000ms, which is regarded as fast enough for the thought-flow of the user to stay uninterrupted by the delay.

Chapter 7

Conclusion

This chapter presents our achievements, gives some concluding remarks and outlines possible future work.

7.1 Achievements

This thesis designs, develops and evaluates a system for capturing, persisting and presenting expert coach knowledge in the form of exercises. The problem defintion stated the following:

This thesis shall design and develop a system for digitally capturing, persisting and presenting data input related to exercise execution within the sports domain. Main focus will be on providing an infrastructure for uploading and presenting expert coach knowledge and expertise. Intended use for such an infrastructure is practice planning and execution by exposing interfaces for managing and presenting videos and textual data related to exercises. Particular focus will be on providing interfaces for athlete monitoring purposes. Examples include the ability for players to provide timely and accurate wellness parameters prior to training sessions, and subjective perception of exertion after training sessions. Usability and user experience shall be emphasized, and design and development will be done in collaboration with end-users.

We proposed a system that exposes interfaces designed and developed using methods and guidelines from the field of Human-Computer Interaction (HCI), and best practice User Experience (UX) guidelines from Microsoft. Our thesis is that providing an infrastructure for capturing and presenting expert knowledge in the form of exercises, and athlete status reports, will improve planning and execution of exercises in the domain of professional sports.

The design and development process has been performed in close collaboration with coaches, medical staff and players from the elite level Norwegian soccer club Tromsø IL. Through this collaboration, we have defined a clear requirement analysis, followed by a prototyping phase for in-process evaluation. Through an iterative process of end-user feedback, we have designed and implemented a system that allows for uploading exercises, creating weekly exercise plans and presenting exercises on a tablet device or big screen display. We have also implemented an athlete status application, which enables the players to provide timely and accurate Rating of Perceived Exertion (RPE) and Pre-Training Wellness (PTW) reports. The reports are archived and presented to the medical staff through a web interface. The athlete status application is currently in operational use by TIL.

In order to evaluate our thesis we have conducted a user survey with assessors from the medical staff, coaches and players from Tromsø IL. We also had coaches from junior teams participating in our user surveys. The evaluation was performed through three Likert-scale based questionnaires, where all components of the system where compared to a baseline. The baseline in all user surveys were the tools currently or previously available to the assessors. In the evaluation of the exercise component of the system, we saw a significant improvement in the rating of the Ruoksat system compared to the currently available tools, regarding planning, presenting, coaching and exercise content. The results from the evaluation of the athlete status application showed similar trends, with all assessors indicating that the Ruoksat system is a considerable improvement compared to the methods and tools previously used. As our main focus in this thesis is usability and user experience, we also asked all assessors to rate the usability, design and how easily the interfaces are to navigate. The feedback stated that all exposed interfaces provided a more than satisfactory user experience, with user experience ratings between Good and Excellent.

7.1.1 Concluding Remarks

The feedback gathered through our user surveys indicate that an infrastructure for digitally capturing, capturing and presenting expert knowledge in the form of exercises can improve planning and execution of exercises within the domain of professional soccer. The evaluation of the user experience related to our implemented interfaces also indicate that using HCI methods, well defined best practice guidelines, and a constant feedback loop with the end-users has resulted in user-friendly, aesthetically pleasing interfaces.

7.2 Future Work

Even through some of the components of the Ruoksat system currently is in operational use, there are still features planned for future releases.

In the near future, we plan to populate the system with exercises from the Tromsø IL senior first team. When the system is in full operational use, we also wish to perform analysis based on statistical data from the system.

There is a great potential for using the Ruoksat exercise component for educating players and coaches from the junior teams. We wish to extend the Ruoksat system so that it will allow for sharing of exercise content, both to the youth academy for early adaption to Tromsø ILs style-of-play, but also for exchanging knowledge with partnering clubs. The athlete status application is currently being considered by other partnering soccer organizations.

We also plan to port the database of the implemented system from SQL to MongoDB, as detailed in Chapter 5. The data model designed to be directly ported from the current SQL solution to a document database. Another important aspect in getting the entire Ruoksat system production ready is adding authentication for the exercise presenter application, and possibly use codecaps [39] for all authentication purposes.

References

[1] Ashman, H., Brailsford, T., Burnett, G., Goulding, J., Moore, A., Stewart, C., and Truran, M. *HCI and the Web*. Lawrence Erlbaum, 2009.

[2] Avram, A. Design details of the windows runtime. `http://www.infoq.com/news/2011/09/Design-Details-Windows-Runtime`, Sept. 2011.

[3] Borg, G. *Borg's Perceived Exertion and Pain Scales*. Human Kinetics, 1998.

[4] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., and Gruber, R. E. Bigtable: a distributed storage system for structured data. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7* (Berkeley, CA, USA, 2006), OSDI '06, USENIX Association, pp. 15–15.

[5] Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., and Young, P. R. Computing as a discipline. *Commun. ACM 32*, 1 (Jan. 1989), 9–23.

[6] Danino, N. Hci and your site, Aug. 2012. University of Central Lancashire.

[7] Davis, M., Hawley, P., and Spilka, G. *Design as a Catalyst for Learning*. Association for Supervision & Curriculum Deve, 1998.

[8] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels, W. Dynamo: amazon's highly available key-value store. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles* (New York, NY, USA, 2007), SOSP '07, ACM, pp. 205–220.

[9] Fotballforbund", N. Statistikk breddefotball 2012 fra norges fotballforbund. http://www.fotball.no/Documents/PDF/2012/NFF/Statistikkhefte_bredde_2012.pdf, May 2013.

[10] Foundation, O. Openstack. http://www.openstack.org/, May 2012.

[11] Freeman, A. *Windows 8 Apps Revealed Using XAML and C#.* Apressus Series. Apress, 2012.

[12] Gossman, J. Introduction to model/view/viewmodel pattern for building wpf apps. http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx, Oct. 2005.

[13] Halvorsen, P., Sægrov, S., Mortensen, A., Kristensen, D. K. C., Eichhorn, A., Stenhaug, M., Dahl, S., Stensland, H. K., Gaddam, V. R., Griwodz, C., and Johansen, D. Bagadus: an integrated system for arena sports analytics: a soccer case study. In *Proceedings of the 4th ACM Multimedia Systems Conference* (New York, NY, USA, 2013), MMSys '13, ACM, pp. 48–59.

[14] Hansen, R. B. A. A video storage management system for soccer analysis. Master's thesis, University of Tromso, Dec 2012. Master of Computer Science.

[15] Hooper, S. L., Mackinnon, L. T., Howard, A., D., G. R., and Bachmann, A. W. Markers for monitoring overtraining and recovery. *Medicine & Science in Sports & Exercise 27*, 1 (1995), 106–112.

[16] Johansen, D., Halvorsen, P., Johansen, H. D., Riiser, H., Gurrin, C., Olstad, B., Griwodz, C., ge Kvalnes, Hurley, J., and Kupka, T. Search-based composition, streaming and playback of video archive content. *Multimedia Tools Appl.* (2012), 419–445.

[17] Johansen, D., and Hurley, J. Overlay cloud networking through meta-code. In *Proceedings of the 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops* (Washington, DC, USA, 2011), COMPSACW '11, IEEE Computer Society, pp. 273–278.

[18] Johansen, D., Stenhaug, M., Hansen, R. B. A., Christensen, A., and Hogmo, P.-M. Muithu: Smaller footprint, potentially larger imprint. In *ICDIM* (2012), S. Fong, P. Pichappan, S. Mohammed, P. Hung, and S. Asghar, Eds., IEEE, pp. 205–214.

[19] Kvalnes, A., Johansen, D., van Renesse, R., Schneider, F., and Valvag, S. V. Design principles for isolation kernels. Tech. Rep. Technical Report 2011-70, University of Troms, Cornell University, 2011.

[20] Lovell, M. Lap around the windows runtime. http://channel9.msdn.com/Events/BUILD/BUILD2011/PLAT-874T, Sept. 2011.

[21] Mackey, A., Tulloch, W. S., and Krishnan, M. *Introducing .NET 4.5*, 2nd ed. Apress, Berkely, CA, USA, 2012.

[22] McLean, B. D. Changes in maximal power output and perceptual fatigue responses during a division 1 female collegiate socccer season. Master's thesis, University of Texas at Austin, May 2011. Master of Science in Kinesiology.

[23] McLean, B. D., Coutts, A. J., Kelly, V., McGuigan, M. R., and Cormack, S. J. Neuromuscular, endocrine, and perceptual fatigue responses during different length between-match microcycles in professional rugby league players. *Int J Sports Physiol Perform 5*, 3 (2010), 367–83.

[24] Microsoft. Laying out an app page. http://msdn.microsoft.com/en-us/library/windows/apps/hh872191.aspx, Nov. 2012.

[25] Microsoft. Navigation design for windows store apps. http://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh761500.aspx, Mar. 2013.

[26] Microsoft. Ux guidelines for windows store apps. http://msdn.microsoft.com/en-us/library/windows/apps/hh465424.aspx, Mar. 2013.

[27] Nielsen, J. *Usability Engineering.* Interactive technologies. Morgan Kaufmann Publishers, 1993.

[28] Nordal, A., Kvalnes, Å., Hurley, J., and Johansen, D. Balava: Federating private and public clouds. IEEE Computer Society, pp. 569–577.

[29] Nordal, A., Kvalnes, A., and Johansen, D. "paravirtualizing tcp". In *Proceedings of the 6th international workshop on Virtualization Technologies in Distributed Computing Date* (New York, NY, USA, 2012), VTDC '12, ACM, pp. 3–10.

[30] Plugge, E., Hawkins, T., and Membrey, P. *The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing*, 1st ed. Apress, Berkeley, CA, USA, 2010.

[31] Pokorny, J. Nosql databases: a step to database scalability in web environment. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services* (New York, NY, USA, 2011), iiWAS '11, ACM, pp. 278–283.

[32] Singh, F., Foster, C., Tod, D., and McGuigan, M. R. Monitoring different types of resistance training using session rating of perceived exertion. *Int J Sports Physiol Perform 2*, 1 (2007), 34–45.

[33] Smith, J. Wpf apps with the model-view-viewmodel design pattern. `http://msdn.microsoft.com/en-us/magazine/dd419663.aspx`, Feb. 2009.

[34] Tanenbaum, A. S., and van Steen, M. *Distributed Systems: Principles and Paradigms*, 2nd rev. ed. ed. Prentice Hall International, 2008.

[35] Tiwari, S. *Professional NoSQL*. Wrox Programmer to Programmer. Wiley, 2011.

[36] Tsukida, K., and Gupta, M. R. How to Analyze Paired Comparison Data. Tech. rep., DTIC Document, 2011.

[37] Valvaag, S., Johansen, D., and Kvalnes, g. Cogset: a high performance mapreduce engine. *Concurrency and Computation: Practice and Experience 25*, 1 (2013), 2–23.

[38] Valvåg, S. V., and Johansen, D. Oivos: Simple and efficient distributed data processing. In *Proceedings of the 2008 10th IEEE International Conference on High Performance Computing and Communications* (Washington, DC, USA, 2008), HPCC '08, IEEE Computer Society, pp. 113–122.

[39] van Reness, R., Johansen, H. D., Naigaonkar, N., and Johansen., D. Secure abstraction with code capabilities. In *Proceedings of the 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing.* (2013).

[40] Watzman, S., and Re, M. *Visual Design Principles for Usable Interfaces*. Lawrence Erlbaum, 2009.

Appendices

Appendix A

User Surveys

A.1 Exercise Library Survey

User Experience Survey

Ruoksat

On a scale from 1 to 5 (poor to excellent), how would you rate…

1a. Using **currently available tools** (i.e. *Pen and Paper, web tools*) for *planning* soccer practices?

○ **Poor** ○ **OK** ○ **Good** ○ **Very Good** ○ **Excellent**

1b. Using ***the Ruoksat system*** for *planning* soccer practices?

○ **Poor** ○ **OK** ○ **Good** ○ **Very Good** ○ **Excellent**

2a. Using **currently available tools** (i.e. *Pen and Paper, Verbal Explanation, etc*) for *presenting exercises?*

○ **Poor** ○ **OK** ○ **Good** ○ **Very Good** ○ **Excellent**

2b. Using ***the Ruoksat Presenter App*** for *presenting exercises?*

○ **Poor** ○ **OK** ○ **Good** ○ **Very Good** ○ **Excellent**

3a. Using **currently available tools** (i.e. *Pen and Paper, Discussions, video footage*) for coaching your team during training sessions?

○	○	○	○	○
Poor	**OK**	**Good**	**Very Good**	**Excellent**

3b. Using ***the Ruoksat Presenter App*** for coaching your team during training sessions?

○	○	○	○	○
Poor	**OK**	**Good**	**Very Good**	**Excellent**

4a. The exercises **currently available** to your team? (i.e. from Youtube, memory, etc)

○	○	○	○	○
Poor	**OK**	**Good**	**Very Good**	**Excellent**

4b. The ability to present exercises from the Tromsø IL A-lag from ***the Ruoksat system*** to your team?

○	○	○	○	○
Poor	**OK**	**Good**	**Very Good**	**Excellent**

Ruoksat Interaction

On a scale from 1 to 5 (poor to excellent), how would you rate

1. The *usability* of **the Ruoksat presenter app?**

○	○	○	○	○
Poor	**OK**	**Good**	**Very Good**	**Excellent**

2. The *design* of **the Ruoksat presenter app?**

○	○	○	○	○
Poor	**OK**	**Good**	**Very Good**	**Excellent**

3. The navigational design (how it is to navigate between the pages) of **the Ruoksat presenter app**

○	○	○	○	○
Poor	**OK**	**Good**	**Very Good**	**Excellent**

4. The information available in **the Ruoksat presenter app?**

○	○	○	○	○
Poor	**OK**	**Good**	**Very Good**	**Excellent**

A.2 Athlete Status System Survey

Ruoksat Athlete Status Application User Survey

On a scale from 1 to 5 (poor to excellent), how would you rate…

1a. Using **previously available tools** (i.e. Pen and Paper) for *collecting* RPE and PTW reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

1b. Using **the Ruoksat system** for *collecting* RPE and PTW reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

2a. Using **previously available tools** (i.e. Pen and Paper) for *archiving / storing* RPE and PTW reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

2b. Using **the Ruoksat system** for *archiving / storing* RPE and PTW reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

3a. Using **previously available tools** (i.e. Pen and Paper) for *reading / viewing* RPE and PTW reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

3b. Using **the Ruoksat website** for *reading / viewing* RPE and PTW reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

4a. Using **previously available tools** (Verbal Discussion, Pen and Paper, E-mail) for *reporting fatigued players* (that should be rested or decrease training intensity)?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

4b. Using **the Ruoksat system** for *reporting fatigued players* (that should be rested or decrease training intensity)?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

Interaction

On a scale from 1 to 5 (poor to excellent), how would you rate...

5. The *Usability* of **the Ruoksat website**?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

5. The *design* of **the Ruoksat website**?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

5. The *navigation pattern* (how it is to navigate between pages) of **the Ruoksat website**?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

A.3 Athlete Reporting Tool Player Survey

Ruoksat Athlete Status Reporting Tool User Survey

On a scale from 1 to 5 (poor to excellent), how would you rate…

1a. Using **previously available tools** (i.e. Pen and Paper) for handing in *RPE* reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

1b. Using **the Ruoksat system** for handing in *RPE* reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

2a. Using **previously available tools** (i.e. Pen and Paper) for handing in *Pre-Training Wellness* reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

2b. Using **the Ruoksat system** for handing in *Pre-Training Wellness* reports?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

Interaction

On a scale from 1 to 5 (poor to excellent), how would you rate...

5. The *Usability* of **the Ruoksat reporting website**?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

5. The *design* of **the Ruoksat reporting website**?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

5. The *navigation pattern* (how it is to navigate between pages) of **the Ruoksat reporting website**?

○ Poor ○ OK ○ Good ○ Very Good ○ Excellent

Comments

[ ]

send

UNIVERSITY OF TROMSØ