UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

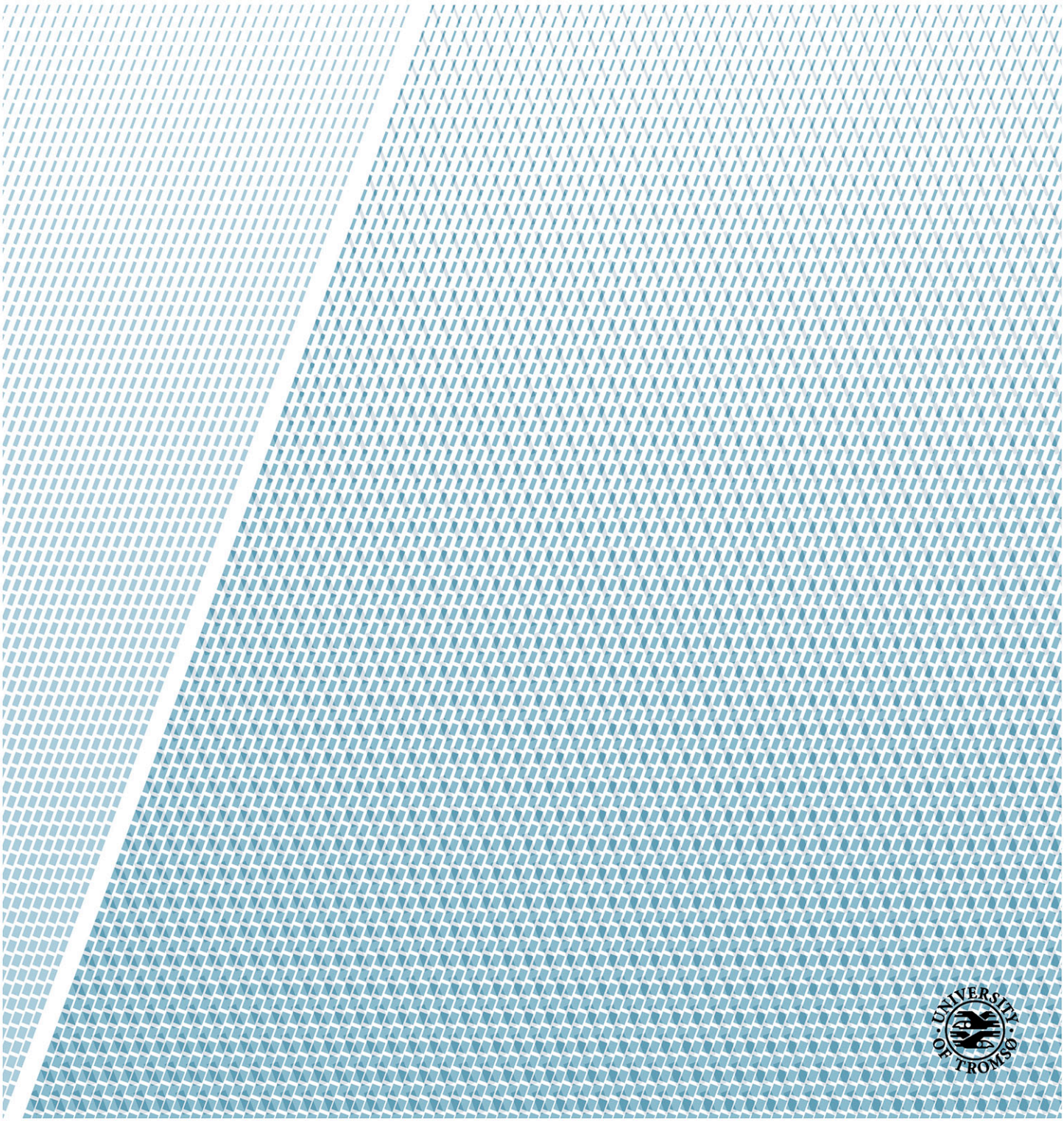
Faculty of Science and Technology
Department of Computer Science

Soccer Analytic Tool

—

Simen Lomås Johannessen

INF-3983 Capstone Project in Computer Science
December 2013



Abstract

Sports science is an increasingly hot topic and more and more soccer teams are investing time and money into strategic development of big data and sports science. All this to increase the players performance on the field and reduce the risk of injuries. Soccer teams and athletes in general are constantly looking for possibilities to gain advantages over opponents. In soccer your next opponent is analyzed down to the smallest details to find weaknesses and strengths. All this to be able to take advantage of your opponents weak points and limits their strengths.

This project creates a system for capturing events relevant for opponent analysis and an interface providing the information for analyzing soccer opponents.

Acknowledgements

Table of Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Problem definition . . . 2
- 1.3 Interpretation . . . 2
- 1.4 Methodology . . . 3
- 1.5 Outline . . . 3

2 Related work **5**
- 2.1 Types of analysis . . . 5
  - 2.1.1 Pre-match . . . 5
    - 2.1.1.1 In match . . . 6
  - 2.1.2 Post match . . . 7
- 2.2 Capturing of data . . . 7
  - 2.2.1 Opta . . . 7
  - 2.2.2 Prozone . . . 8
  - 2.2.3 Automatic capturing . . . 10
  - 2.2.4 Wrap up . . . 10
- 2.3 Presenting data . . . 11
  - 2.3.1 Prozone . . . 11
  - 2.3.2 ZXY . . . 12
  - 2.3.3 FourFourTwo . . . 13

3 Requirement Specification **15**
- 3.1 Overview . . . 15
- 3.2 Capture . . . 16
- 3.3 Present . . . 17
- 3.4 Summary . . . 19

4 Soccer Analytic toolkit **21**
- 4.1 System Architecture . . . 21

4.1.0.1 Interface . . . 22
4.1.0.2 Back-end . . . 22
4.1.0.3 Storage . . . 22
4.1.1 Domain model . . . 22
4.2 Implementation details . . . 23
4.2.1 Storage . . . 24
4.2.1.1 Indexes . . . 24
4.2.2 Back-end . . . 25
4.2.2.1 API . . . 25
4.2.3 Front-end . . . 25
4.2.3.1 Models . . . 26
4.2.3.2 Views . . . 27
4.2.3.3 Router . . . 28
4.2.4 Getting players and teams into the database . . . 29
4.2.5 Security . . . 29

5 Demo **31**
5.1 Interfaces . . . 31
5.1.1 Mockup . . . 31
5.1.2 Implemented interfaces . . . 31
5.1.2.1 Home page . . . 31
5.1.2.2 Registrating attacks . . . 32
5.1.2.3 Match view . . . 32
5.1.2.4 Team view . . . 32
5.2 Capturing process . . . 33

6 Evaluation and Results **45**
6.1 Methods . . . 45
6.1.1 User Survey . . . 45
6.2 Experiments and Results . . . 46
6.2.1 Test data . . . 46
6.2.2 SAT as a tool for opponent analytic . . . 46
6.2.3 SAT as a tool for identifying key players . . . 47
6.3 Discussion . . . 47
6.3.1 Input . . . 47

7 Conclusion **49**
7.0.2 Concluding Remarks . . . 49
7.1 Future Work . . . 49

List of Acronyms

List of Figures

2.1 A typical tweet from one of Opta's Twitter account 6
2.2 The Prozone camera system illustrated. Taken from Prozonesports.com 9
2.3 Overview of the ZXY system with receivers placed at the stadium and players wearing sensors. Image from zxy.no 11
2.4 The Prozone software - individual player analysis gives you statistics of performance over time. Image from prozonesports.com 12
2.5 The ZXY software - tracking of players gives you information of distance covered, average speed and max speed. You also get a heat map of the players movement on the field. Image from zxy.no 13
2.6 Illustration of all passes in to attacking third in the match Manchester United vs Newcastle United. Taken from fourfourtwo.com 14

3.1 Visioned illustration displaying key players and how they combine given you have queried on a team 16
3.2 A screen capture showing some data Troms IL captured in a previous analytic project 16
3.3 Illustrations of different breakthroughs we want to registrate . 17
3.4 Dividing of pitch - the first suggestion had 18 zones 18
3.5 Dividing of pitch - the second suggestion had 24 zones 18
3.6 Conceptual architecture 19

4.1 Overall architecture of the system 21
4.2 Software stack 23
4.3 Overview of the web servers API 26
4.4 Architecture of the client side 26
4.5 Shows how the passing statistic is illustrated on the client by using Highcharts.js 27

4.6 Illustrations of which zones the team has finished off their attacks from, with percent. Team - Troms IL) 28

5.1 A mockup of the main analytic page 34
5.2 All matches registrated in the database are listed on this page 35
5.3 Interface listing up all attacks for a match 36
5.4 Interface for registrating an attack 37
5.5 Interface listing all teams 38
5.6 Main interface for information about opponents - key aspects . 39
5.7 Main interface for information about opponents - offensive play 40
5.8 Main interface for information about opponents - defensive play 41
5.9 Main interface for information about opponents all players . . 42
5.10 Player view highlighting individual statistics 43
5.11 Player view highlighting individual statistics 44

6.1 Matches that have been captured and persisted into the database 46

List of Tables

Chapter 1

Introduction

1.1 Background

Sports science is an increasingly hot topic and more and more soccer teams are investing time and money into strategic development of big data and sports science [6]. All this to increase the players performance on the field and reduce the risk of injuries. One of the pioneers in this field for a long time AC Milan established in as early as 2002 the MilanLab [5]. The goal for the program was to get better and more concrete information about the players physicality by collecting data over time of players performance. As AC Milan was one of the first to use big data, the use of big data today has exploded. Services like Opta and Prozone serves player statistics, heat maps, and video analytics of players performance. Fans get exposed to these statistics by clubs and broadcasting companies that uses it actively in their coverage of soccer games. The awareness of statistics for clubs and fans has possible never been higher than now [4].

Soccer teams and athletes in general are constantly looking for possibilities to gain advantages over opponents. In soccer your next opponent is analyzed down to the smallest details to find weaknesses and strengths. All this to be able to take advantage of your opponents weak points and limits their strengths. The players should know what to expect from the opponent team. Detecting typical team plays and player movement of your opponent can help you prepare for the match.

There are several ways to gather information about your opponent; from looking through whole matches to advanced tools, which can highlight key

information for you. Some of them are expensive as Prozone [7] which is a complete system for capturing and presenting information. For small soccer clubs with relatively small budgets this can be a expensive investment. Another system is Interplay Sports, which Troms IL uses for game analytic. A video feed from any match can be used as the input letting you manually tag and describe situations in the match. In a sport where the next soccer match usually is in 3 days tools for filtering out the useless data is valuable.

In the analytic process there are two processes we will look at; first you need to gather the information and secondly is how to present the information gathered.

1.2 Problem definition

This project will develop a system complementing the Muithu and Bagadus systems. Focus will be on soccer opponent analytics, where a data repository needs to be developed capturing important events relevant for this type of analytics. Especially we want to identify the key players offensive in a team. A user interface component providing the core information about the opponent should also be developed.

1.3 Interpretation

The project is that providing an infrastructure for capturing events like attacks that leads to an attempt on the opponent goal, and presenting this information through a user interface. We are interested in how long it typically takes to capture all relevant events from a single match, as this has to be done manually. A data model that reflects what we want to capture from each attack needs to be developed for being able to get relevant information for being used as a opponent analytic system.

First a requirement specification shall be developed. Then a prototype and that fulfills the requirements. At last specialist in the domain of soccer shall evaluate the system.

The design and development processes will be performed in collaboration with staff of the Norwegian soccer club Trms 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 project.

1.4 Methodology

The final report of the ACM Task Force on the Core of Computer Science [2] divides the discipline of computing into three major paradigms:

- *Theory*: Theory: Rooted in mathematics, the approach is to define problems, propose theorems and look to prove them in order to find new relationships and progress in computing.
- *Abstraction*: Rooted in the experimental scientific method, the approach is to analyze a phenomenon by creating 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 project the design process seems to be the most suitable out of the three paradigms. The design process consists of 4 steps, which are repeated if tests reveal that the latest version of the system does not meet the requirements.

- *State requirements and specification*: A need or problem is identified, researched, and defined.
- *Design and implement the system*: Data models and a system architecture are designed. Prototypes are implemented.
- *Test the system*: Assessment and testing of the aforementioned prototypes.

1.5 Outline

- *Chapter 2*: Presents some related work to the project
- *Chapter 3*: Describes the requirement specification
- *Chapter 4*: Describes the design and implementation of the system
- *Chapter 5*: Gives a demonstration of the system
- *Chapter 6*: Evaluates and discuss the system
- *Chapter 7*: Concludes the project

Chapter 2

Related work

In this chapter we present some background related to analytic in the domain of soccer. We look at different types of analysis out there. This spans from pre-match to post-match. Then we go into how data is gathered and presented. For gathering of data there are two main approaches: manually often by human annotating it, and automatic capturing often by using sensors.

2.1 Types of analysis

In soccer there are several phases where you use analytic to help you gain insight. In this section we will look at the typical use cases of analytic. Not only soccer teams uses analytics, but also TV and fans widely uses statistics to build up under their arguments.

2.1.1 Pre-match

Pre-match you use analytic to find weaknesses and strengths in the opponent team, on an individually level or as a team. You look at your team matched up against your opponent. A typical situation is that the manager gets a video summary of the opponent highlighting the opponents strengths and weaknesses. The coaching staff may use tools like Interplay to gather information and create the video summary. Other software like Prozone**??** will also likely be used to break down the opponents offensive and defensive play.

Typically in TV you have pundits bringing you analytic of key battles during the build up to the game. A typical thing is to look at battles like wingers versus backs, midfield clashes or striker versus central defenders. The battles are often illustrated with statistics or with video clips highlighting aspects of the both players game that will be decisive.

Fans are analyzing the game all the time. Except from TV they get exposed to analytic via social media. Opta is company that has taken full advantage of social medias. Figure **??** is a screenshot of a tweet from one of the many accounts Opta control.

Figure 2.1: A typical tweet from one of Opta's Twitter account

2.1.1.1 In match

During match the coaching staff continuously analysis the match and makes adjustment. The players make all the decisions in the end, but the coach is the boss and sets the style of play. An adjustment to the formation can potentially be the tipping point that wins you the game.

Troms IL uses a system Muithu that lets you annotate sequences of a game with entity's like player, comment. This information will then be time synchronized with the video feed. Later, like in at half time or even in the game, you can search on entity's to get the corresponding video feed. As the system is available on tablets players can in the middle of the match come to sideline to see a involvement that the coaching staff would like to show. It can be anything that is tagged like a player involvement to a team move that was executed perfectly.

Using systems like ZXY or MiCoach you can get real time information about a players performance during the match. Rather than assuming that a player is tiring you can get information metrics that tells you this. You get evidence that the player in fact is fatigued. In soccer you only have 3 substitutions. Making the right ones is crucial in an even match.

Using data gathered from sports data company's like Opta you can get statistics live during the game. Its popular in TV to show statistics like ball possession percent, how far players have run or the number of passes played and so on. As mentioned earlier Opta posts many of their statistics to their social media accounts for fans and media.

2.1.2 Post match

During the post match the coach team goes through the game to evaluate the team performance. This is valuable as you get concrete clips about good and bad involvements on team or player level. You can highlight situations in the match to help players better understand tactical aspects. Interplay Sports is a system used today at Alfheim for this purpose. It is used for analysis of matches in a post-match scenario where you can tag situations in the video.

2.2 Capturing of data

2.2.1 Opta

One of the big players in the field of sport analytic Opta uses manual input to create their data repository. They have editorial teams across the world that captures data manually for the most popular soccer leagues and other sports. For example to capture statistics for a single match they have 3 humans annotating. The data is captured via an application specifically created for the purpose of capturing data as quick and easily as possible. The editorial teams of Opta need to be able to identify a player in a very short time to be able to keep up with the pace of the game. They even study things like which shoe color a player.

Opta capture all types of actions like passes, type of pass, attacks, and interceptions to mention a few. For each action they log, they add a series of description tags like pitch coordinate, timestamp, which player and team. For every single pass they add descriptors if it was a through ball, normal ball or a headed flick on from a long ball. For shots they register the foot it was kicked with, if it was a volley and so on. All this is done while the match is playing. About 1600 individual events are recorded in a standard match.

This gives them a very rich database of events and a range of possible queries to run.

```
<Event id="290575408" event_id="5" type_id="1" period_id="1"
min="0" sec="5" player_id="20856" team_id="810" outcome="1"
x="44.6"y="61.1" timestamp="2007-08-12T13:00:24.827"
last_modified="2007-08-12T13:00:25">
<Q id="1774596260" qualifier_id="141" value="91.6"/>
<Q id="1429253465" qualifier_id="140" value="49.9"/>
<Q id="1084400575" qualifier_id="56" value="Back"/>
</Event>
```

Above is an example of an event registrated in the Opta database. The event has a series of qualifiers describing it. Except from the obvious as timestamp and last modified dates we see that the player id, team id, time of event, x and y coordinates and the outcome of the event are registrated. Also we see that some extra details are included. In this example it maps to a pass from [44.6, 61.1] to [91.6, 49.9].

2.2.2 Prozone

Prozone is a video-based system that tries to track players in team sports [7]. Their data capture system incorporates 8-12 cameras, which is strategically positioned throughout the stadium to cover 100 percent of the ground, but with some redundancy in case of a faulty component. They also incorporate the TV-camera feed, which always follows the ball. All cameras are hooked into one server and uploaded at the end of the game before sent to undertake the tracking process. When they system is installed it required some setup before it is operational; cameras need to be calibrated out from the pitch size. Figure 2.2 shows the camera location on a soccer stadium.

The coding and tracking process of the player's movement and actions is a sort of manual process at present time. First each camera feed goes through each own tracking process determining image co-ordinates and continuous trajectories for each player. The output from all the cameras is then combined into one dataset. Going into the algorithm for this is out of the scope of this project. In the final stage the manual work comes into the loop. There has to be a quality control group of operators dedicated for post processing the game. First the operators has to map the players identity and with the their corresponding start location. The operators will then follow the video feed checking that the identification of all players remains constant

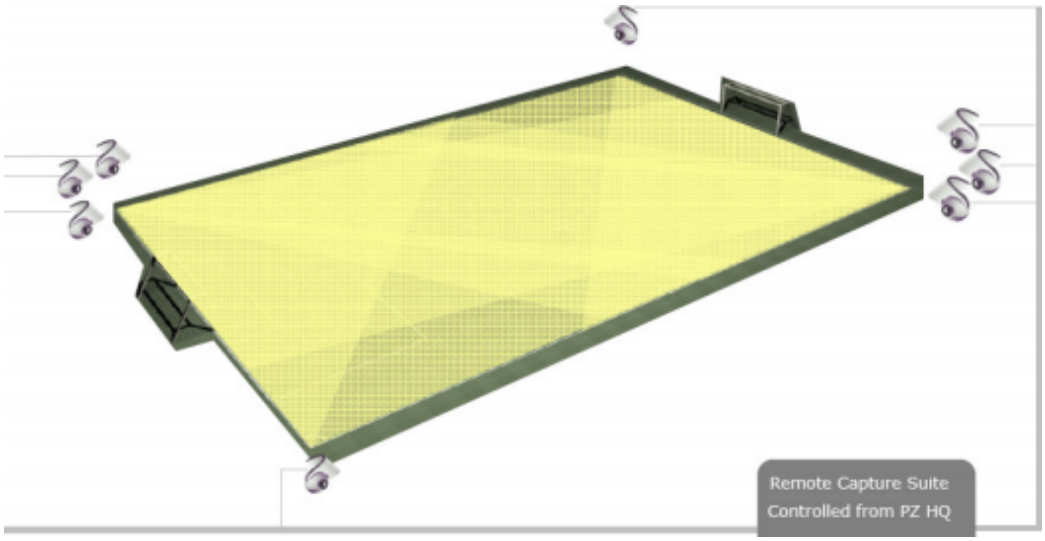


Figure 2.2: The Prozone camera system illustrated. Taken from Prozonesports.com

during the match. The tracking process may run into problems when two players collide or have any other physical contact as it becomes unclear who is who. To map video image co-ordinates to pitch co-ordinates Prozone uses computer vision homography calibration process.

The whole input process is done in a own software which helps you minimize the amount of manual work for the operators. The software follows rules created by basic machine learning algorithms to validate and verify the input. A simple example would be if the ball goes out of play the system will now that the next event will be a throw in, corner kick or goal kick.

Prozone claims to be able to track every movement of every player on the pitch with a resolution of 10th of a second. This without using any physical equipment on the players. Di Salvo et al. [1] conducted an empirical evaluation of deployed Prozone systems at Old Trafford in Manchester and Reebook Stadium in Bolton, and concluded that the video camera deployment gives an accurate and valid motion analysis. The data is after a match available through Prozone interface for analysis.

2.2.3 Automatic capturing

A system that uses sensors is the ZXY sport tracking system [1]. The system is in used by premier league soccer teams in Norway, including Troms IL and Rosenborg BK. Data captured is stored in Sybase databases with each match requiring about 500-700MB storage The players have to wear a belt around their waist for the system to be able to track their movements. The ZXY system is able to track the players movement very detailed with an accuracy of 0.5m. It has a resolution of 20 samples per second. The technology behind it relies on a radio-based signaling substrate to provide real-time high-precision positional tracking, also including acceleration and heart rate. An installation of receivers is required for the system to work. The home arena for Troms IL, Alfheim, is currently equipped with 10 receivers. A receiver tracks a specific area of the soccer field and combined they cover the whole pitch with some redundancy areas. The communication from the belt to the receivers goes on a 2.45-5.2 G Hz frequency radio signal. To compute the positional data the stationary radio receiver uses an advance vector based processing of the received radio signal. The data is aggregated and stored into a relational database. Including the positions of the players the ZXY also gives you the step frequency, speed and direction [3]

2.2.4 Wrap up

The main problem with tracking systems that uses physical sensors is that usually only one of the teams wears the sensors. This limits the functionality of the system as an opponent analysis system. You only get data for one team and if you have installed it at your stadium it captures only half of the matches.

On the other hand you have the manually systems that requires human annotation of matches. These systems have the advantage of being able to track both teams. As they rely on human annotation of some degree they get more rich data as well, but requires strict rules of the semantics of the metadata.

[1] http://www.zxy.no

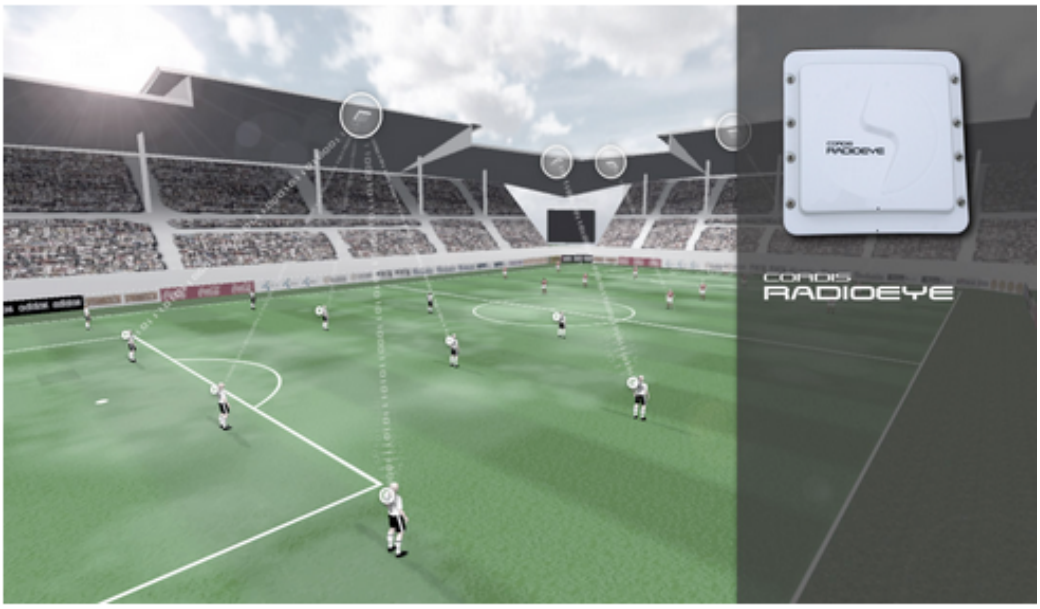


Figure 2.3: Overview of the ZXY system with receivers placed at the stadium and players wearing sensors. Image from zxy.no

2.3 Presenting data

This sections gives an short overview of how some systems presents opponent analysis.

2.3.1 Prozone

Prozone comes with several softwares to illustrate the data. The most relevant for this project is the opposition analysis system. It includes features like 2D animation, single player analysis, team analysis, pressing analysis, success/direction, player tempo, passing movements, receving the ball, player events.

On individual level you can get basic statistics like shots, total passes, passing success, crosses, shots on target, balls received, tackles, fouls.

Doing queries on the data can give you all sprints for a certain player. Players in certain areas of the field have more intensive sprints when they first are involved thus are more vulnerable to injuries. Knowing the actual physical load on players can prevent injuries by regulating the training intensity and

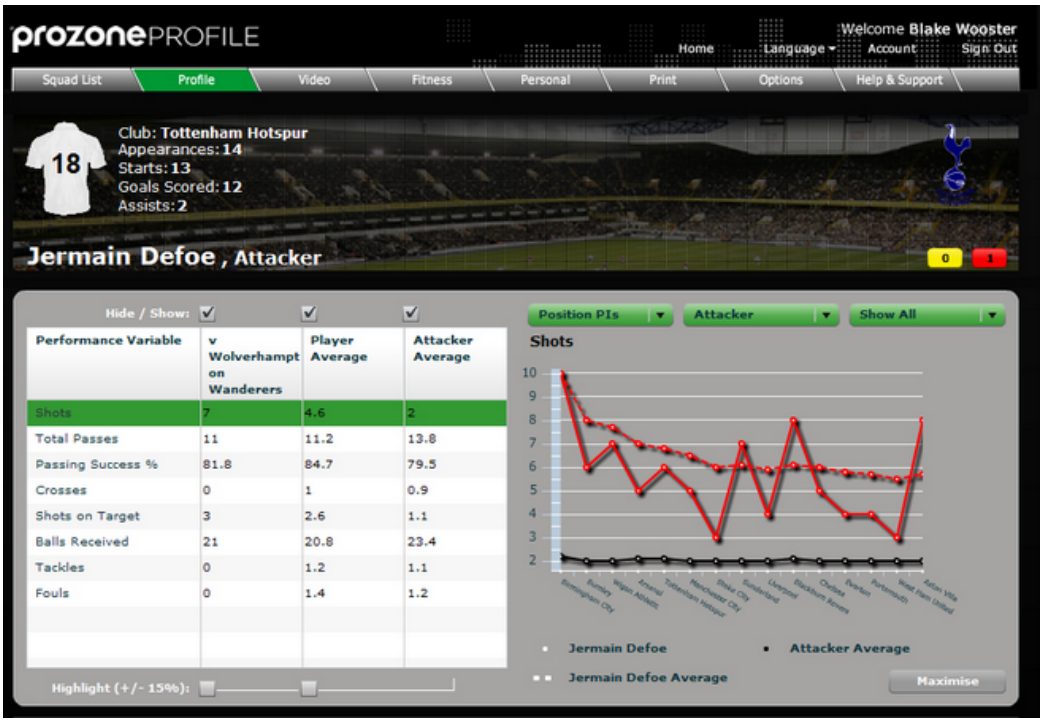


Figure 2.4: The Prozone software - individual player analysis gives you statistics of performance over time. Image from prozonesports.com

amount of time on each exercise. You can get 2D animation of matches as they have tracked every players movement.

2.3.2 ZXY

ZXY provides you with a 3D graphic interface showed in figure 2.5. This interface lets you see the players action in real time by reading the data stream to reproduce the players action. The data is streamed in real time into the database as the match goes on. While watching you can produce timestamps of different events and produce manual input which is time synchronized with the automatic data. Naturally you can build your own software on top of the Sybase database. Troms IL in collaboration with University of Troms has made several systems to complement the ZXY software. Muithu and Bagadus.

Figure 2.5: The ZXY software - tracking of players gives you information of distance covered, average speed and max speed. You also get a heat map of the players movement on the field. Image from zxy.no

2.3.3 FourFourTwo

FourFourTwo is a monthly magazine about football. They also have a web-page which they update daily. Particular they focus on analytic and has a business del with Opta giving them access to all theirs statistics. Including to own analyses on their page they have their own application Stats Zone where everyone can analyze matches with the same tools as they do self. Figure 2.6 shows a typical graphical illustration done with the Stats Zone application highlighting passes in to attacking third.

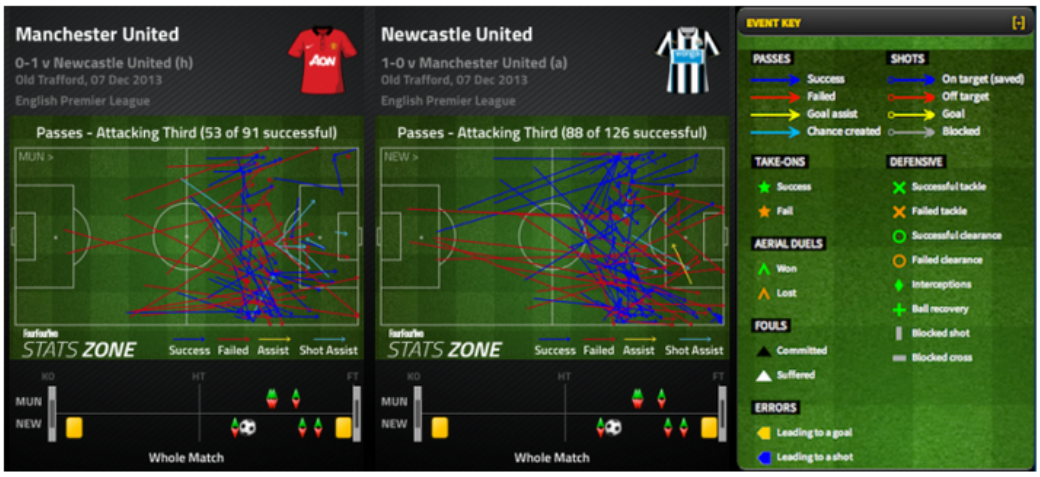


Figure 2.6: Illustration of all passes in to attacking third in the match Manchester United vs Newcastle United. Taken from fourfourtwo.com

Chapter 3

Requirement Specification

This chapter outlines the requirement specification of the system and the process to get there. The process of gathering the requirements was done in collaboration with Troms IL.

3.1 Overview

The requirements evolved during the process of developing the system. Initially a requirement specification was designed from our perspective. We looked at the different anal systems out there and what was in use at Alfheim today. Many systems looks at a single match when analyzing like FourFourTwos application. We are interested to see statistics over time to find patterns in the data. The quest of finding which players it is that creates the attack behind goal chances is the driving force behind the system. Every man can point at the striker as he has scored 20 goals for a team and say he is the teams main man. "We need to stop him!'. There is most likely other players playing in a more defensive role that has the same influence on the teams results.

Rather than going very wide providing all kind of analyses and statistics we narrowed it down to a very concrete system. A system that tries to do for many things may fell between two chairs and at the end doesn't provide anything. When you spend less time on each feature the quality of each feature is most likely reduced.

The imagined system shall give you the key players in the offensive play of a given opponent soccer team. You shall be able to search on teams

and individual players. Additionally you shall be able to see which areas of the pitch players are creating goal chances from. The system also needs a way of capturing data. This shall be an interface that enables you to store successful attacks for any match. We define key players as players that is often the breakthrough player in the attack by dribbles or passes that opens up the opponent. Figure 3.3 shows involements of a typical key player.

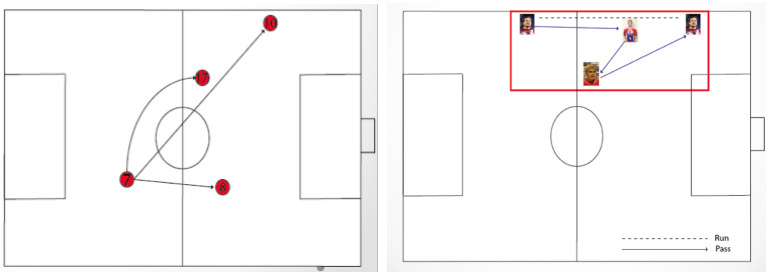


Figure 3.1: Visioned illustration displaying key players and how they combine given you have queried on a team

3.2 Capture

A domain model for the captured data is crucial to set as early as possible in the process. A change to the model slows down the development process; captured data has to be re-captured and previous queries on data may fail due to missing fields. The domain model should reflect what we want to get out of the system. It sets the boundary's for which information we can pull from the system afterwards. When defining requirements Troms IL gave us a domain model they have used for a previous analytic project. The analytic project presented a breakdown of all goals for Manchester United in the 2011/2012 season in Barclays Premier League. This domain model shown in figure 3.2 is used as our foundation for our domain model.

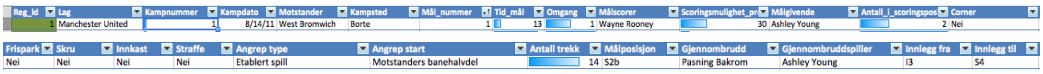


Figure 3.2: A screen capture showing some data Troms IL captured in a previous analytic project

The visioned system needs a data store. The data should be stored in database that has a rich query language for being able to support the large

range of queries on the data. Good support for aggregate functions is crucial. The data will consist of text and integers.

Additionally an interface for input will be required. This interface should ensure that the input data is correct. For example when defining what type of attack the attack is, only the predefined options should be valid as input.

The visioned system should contain database of all the matches in the Norwegian premier league. From each match every successful attack that leads to an attempt on goal should be captured. From the attack you capture: where the attack started, every pass including from/to zones, and what type of attack the attack can be categorized as. At last if there is a breaking point in the attack this should be captured. The breaking point of the attack should be stored with a breakthrough player and what type of breakthrough it was. Figure 3.3 illustrates the different breakthroughs we want to capture.

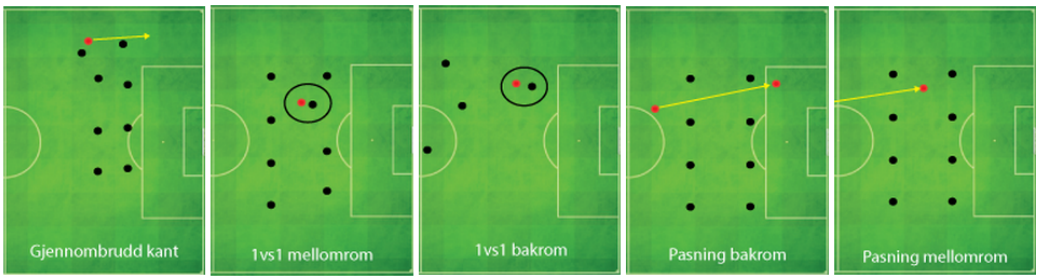


Figure 3.3: Illustrations of different breakthroughs we want to registrate

Capturing every pass in a attack leading to an attempt on goal requires a dividing of the pitch into zones. The process of defining how to divide the pitch into zones took several rounds of discussion leading to several types. These are shown in figure 3.5

3.3 Present

The presentation of the data is an important aspect of the system. This will be where the end-users will spend their time. The presentation of data needs to be as simple as possible to understand. The coaching staff is no technical experts and the system is aimed for them to use. Therefor the system will need to present statistics and other valuable information in clean and understandable way. In the end it is the 11 players the coach selects that

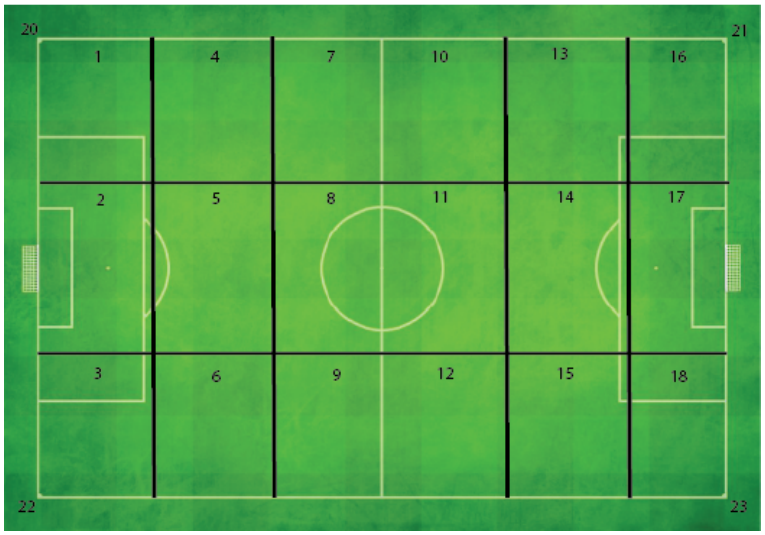


Figure 3.4: Dividing of pitch - the first suggestion had 18 zones

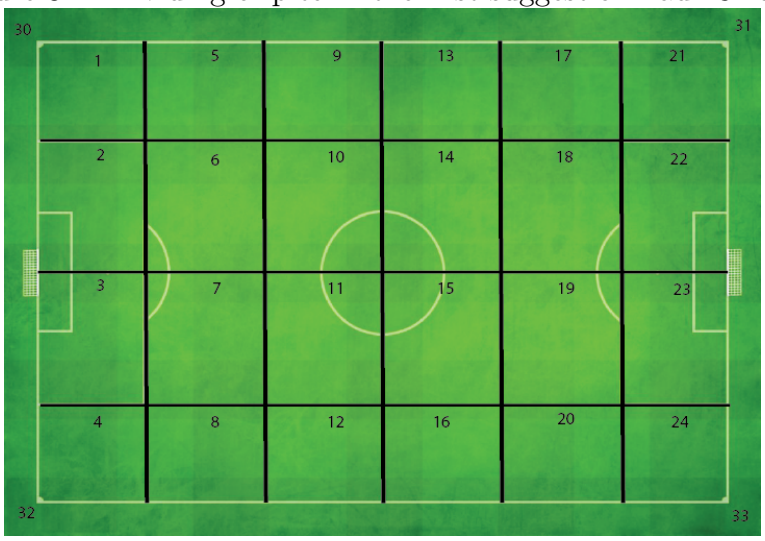


Figure 3.5: Dividing of pitch - the second suggestion had 24 zones

perform all the actions, but the coach sets the style of play and boundaries for players. The goal is to give the coach the opportunity to use the system, using both analytic illustrations and statistics, so he can better reach out to his players with his ideas. The better a coach can sell his ideas to the players the more they will believe in the philosophy, style of play and follow the game plan in games.

We purpose a web interface here as it is accessible from many devices as long as you have an Internet connection and a up to date web browser.

3.4 Summary

To summarize we need storage, back-end and an interface for capturing and presenting analytic information. Figure 3.6 shows the conceptual architecture.

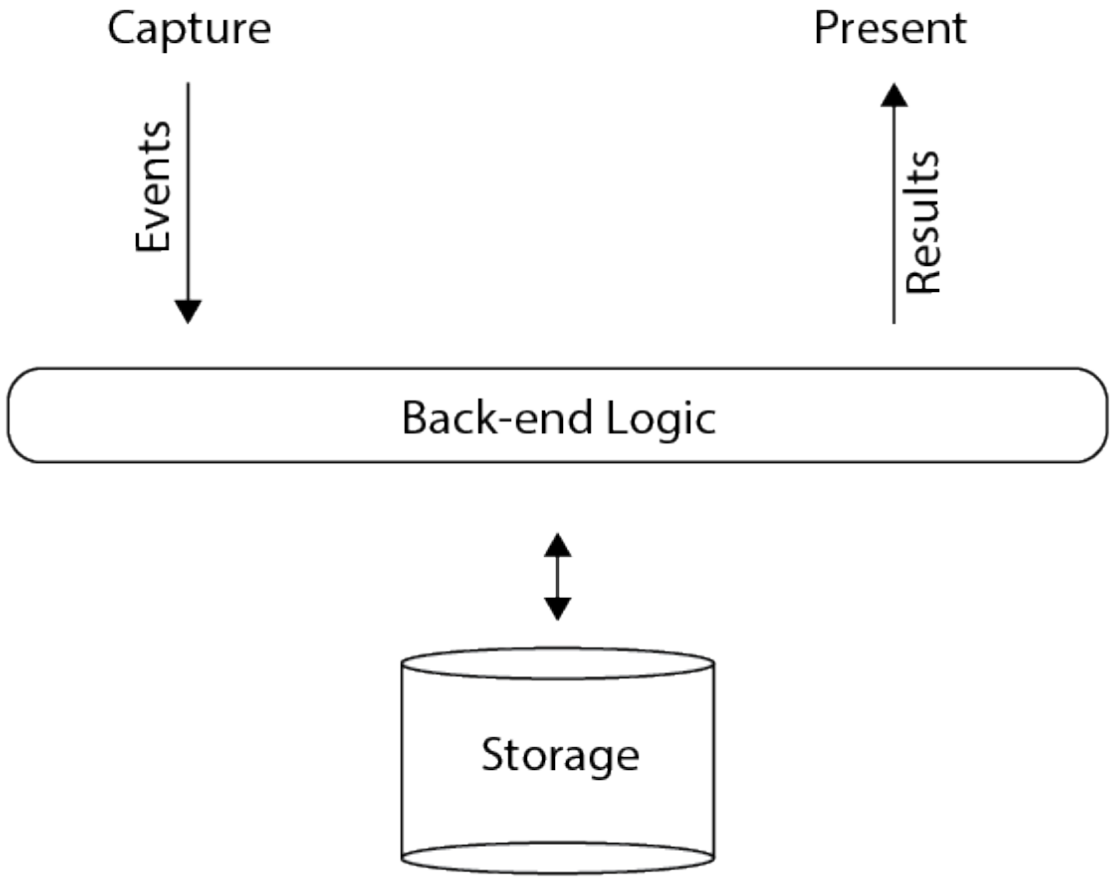


Figure 3.6: Conceptual architecture

Chapter 4

Soccer Analytic toolkit

4.1 System Architecture

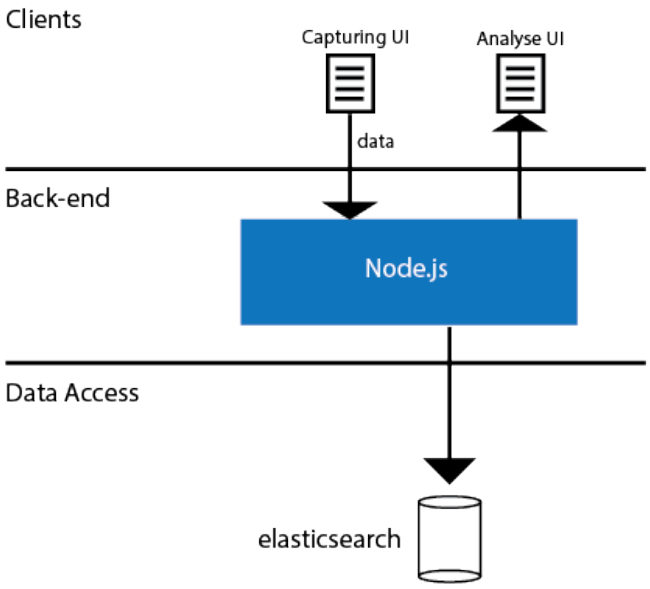


Figure 4.1: Overall architecture of the system

The system architecture can be layered in to three layers; client, back-end, database. The control flow flows from the clients requesting something or inserting data, to the back-end and to the third layer, the database, and back up again in reverse order.

4.1.0.1 Interface

The Soccer Analytic tool has one user interface - the web browser interface. Both the input and the analytic interface can be reached from this single web browser interface.

4.1.0.2 Back-end

The back-end is the middle layer between the client and the storage. Its main task is to serve static files to the clients, handle data insertions or handling web request by mapping them to database operations. New data insertions will possible be inserted into several database indexes. The back-end will then ensure that all indexes are updated before returning success to the client.

4.1.0.3 Storage

The storage layers task is to persist data and handle search queries on the data. It consist of several indexes that each store a part of domain the model.

4.1.1 Domain model

The domain model is based around matches. All attacks is wrapped into a match root. Attacks can be seen as subdocuments of the match document. Inside the attacks all passes lies with other information. This gives a easy way of understanding how all the data is related together. Below is a complete example of how it all is structured. The number of attacks and passes is stripped down to one in the example.

```
{
    "hometeam" : "Troms",
    "awayteam" : "Rosenborg",
    "score" : "1-0",
    "date" : '2013-15-09',
    "attacks": [
        {
            "time": 4,
            "touch" : 1,
```

```
            "team" : "Troms",
            "breakthrough" : "None",
            "breakthroughPlayer" : "None",
            "typeOfAttack" : "D dball",
            "attackStart" : {
                "pos" : 17,
                "typeAction" : "Frispark",
                "player" : 403,
            },
            "passes": [
                {
                    "fromPlayer": 403,
                    "toPlayer": 393,
                    "fromPos": 17,
                    "toPos": 23,
                    "action": "CROSS"
                }
            ],
            "finish" : {
                "player": 393,
                "pos": 23,
                "action": "SHOTMISS"
            }
        },
}
```

4.2 Implementation details

Software Stack	
Web server	Node.js
Database	Elasticsearch
Web Interface	HTML5, CSS3
Client Side Technology	Backbone, Bootstrap, jQuery, Mustache, Highcharts, Lightbox, async

Figure 4.2: Software stack

4.2.1 Storage

The data storage is an elasticsearch database[1]. Elasticsearch is document oriented, schema less and works well with JSON[2]. As our server is built on JavaScript working with JSON is easy. JSON-objects can be inserted right into the storage and elasticsearch will map fields and value accordingly, and make it available for search. Elasticsearch takes advantages of embedded documents meaning we can store related data together. An attack is usually made up of several passes these can be stored as an embedded document in the attack document. Then all pases can be retrieved in one query when fetching an attack.

The main reason for using Elasticsearch is its search capability. In the starting phase of the project MongoDB [3]was the storage engine. However as some aggregation queries was hard to figure out how to do a change to elasticsearch was made. It should be noted that MongoDB supports map reduce operations. With some extra work the queries causing problems could possible have been done with MongoDB. However, with elasticsearch you can in a single, simple to write query get counted how many passes all players for a team has played and received, the number of times all players has been the breakthrough-player in an attack, count type of attacks, count the most used zones for passing and finishing and so on. This makes it very easy and efficient to do queries for analyses on teams and players.

4.2.1.1 Indexes

Indexes in much like tables in a relational database in the way that they is a container for data. An index stores documents which is a bunch of key/value pairs like JSON. You can let elasticsearch automatic analyze the field data or you can use mappings. For supporting querying on players and other fields where the value is more than one word, you have to tell elasticsearch that it is a multi field. For example searching for player name "Stefan Johansen" when the field is not set as multi field will give you two results and not one as you would expect.

In our project we have 5 different indexes. Team index which stores all the teams. Player index which stores all the players. Match index stores all data from a match including all attacks. Attack index stores information about

[1]http://www.elasticsearch.org/
[3]http://www.json.org/
[3]http://www.mongodb.org/

attacks and last the pass index which stores passes. As may be noted there is data redundancy. You can argue that storage has become so cheap and if you can use a little bit extra space to gain performance you would do it. In this case it was done to be able to fully support aggregation functions on subdocuments.

4.2.2 Back-end

The back end is the middle-ware between the clients and the data layer. It exposes a RESTful[4] interface over HTTP for the client to communicate. A request coming in is transformed to a database query based on the resource it tries to access. On answer from the database the result is transformed before returning it to the client.

Similar if the client sends new data for a match the middle-ware inserts the data into the appropriate indexes. The server will respond will HTTP status code 201 if all goes well or 400 on an error. The server uses HTTP code actively to tell the client the result of requests.

In principal, since the data input is generated in the web browser (with JavaScript), it could have been inserted right away into the database without going through an extra middleware. However, this limits us as we cant combine multiple queries by going through the back-end. Cleaning of data before serving to the client would neither be possible. Normally you also do validation of the data at the back-end before inserting into your database.

4.2.2.1 API

The API of the web server is listed in figure 4.3.

4.2.3 Front-end

Front end is consist of a single page JavaScript application using Backbone.js [5]as under-supporting framework. Backbone uses a MVC model to structure the code. With Backbone your views will update automatic when data changes. In the following sections the architecture of the client and how different concepts is used will be described.

[4]http://www.ibm.com/developerworks/webservices/library/ws-restful/
[5]http://backbonejs.org/

Web server API	
GET /matches	Get all matches
GET /match/:id	Get specific match
POST /match	Post new match
GET /teams	Get all teams
GET /team/:name	Get team statistics
GET /team/:name/finalthird	Get passes into final third of the pitch
POST /team	Post new team
GET /player/:id	Get player statistics
GET /player/:id/breakthroughs	Get breakthroughs for player
GET /player/:id/finalthird	Get passes into final third of the pitch
POST /player	Post new player
POST /attack	Post new attack
POST /pass	Post new pass

Figure 4.3: Overview of the web servers API

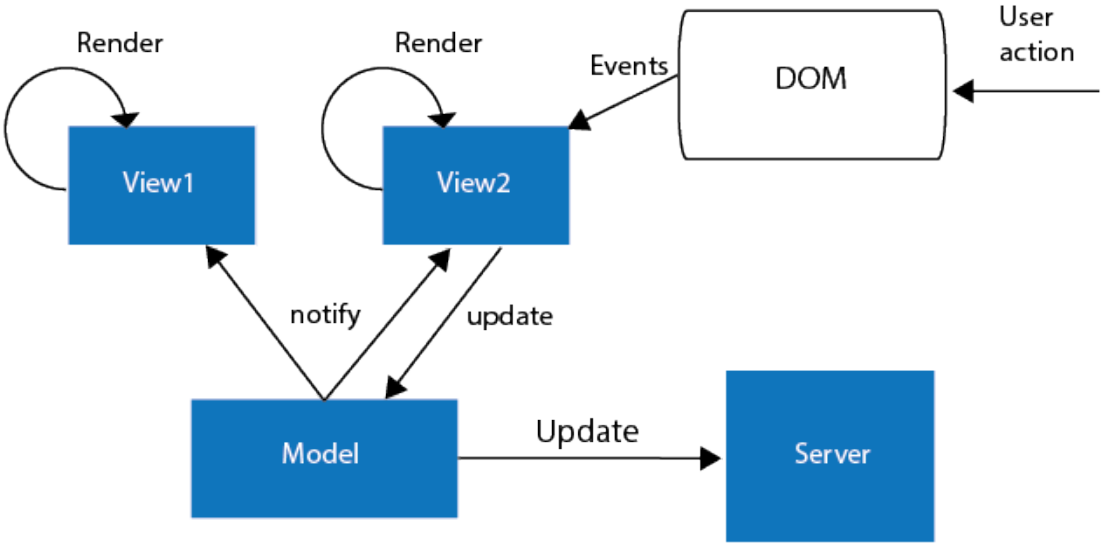


Figure 4.4: Architecture of the client side

4.2.3.1 Models

Data is represented as models in backbone. The model has mainly two responsibilities. First is whenever a update on a models data occurs the model notifies the views that has subscribed for update events for that particular

model. The second is that models is responsible for AJAX communication with the back-end. An example is when a user registrates a new attack for a match. He fills out a form and press submit. Then a new Attack Model is created. Calling save on the instance of the model will send an AJAX post request to the server with the models data in the HTTP body.

Similar to a Attack Model we have Player Model that handles everything around players. When you click yourself into a players profile the model will fetch statistics from the back-end, notify the view that the data is ready, and the view will be rendered.

There is also a Match Model (fetching and registrating matches), Team Model (viewing statistics), and a Pass Model (saving passes). They work it the same way as the models described above.

4.2.3.2 Views

For a analytic toolkit to be useful a good UI is critical. Here several helper library is used to present the data. Highcharts.js[6]is a JavaScript library for illustrating graphs. A query on team generates a lot of statistics and rather than listing them up they are presented using charts. This also gives us the advantage of displaying several numbers for each player and plot it in the same graph. In figure 4.5 the number of times a player has been involved in all attacks, number of passes into the final third of the pitch and the number of times a player has been the breakthrough-player is shown.

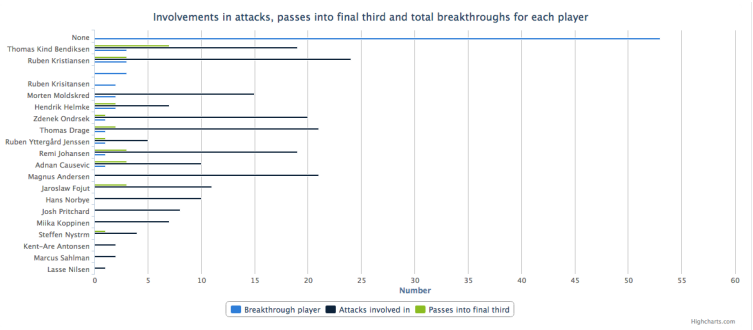


Figure 4.5: Shows how the passing statistic is illustrated on the client by using Highcharts.js

Positional data is created using the a new feature of HTML5, canvas element. It lets you draw graphics on the fly in the web page. In this system it is used

[6]http://www.highcharts.com/

to create a element that symbols the different zones in our domain model. In the Team Model you have all zones with a number that symbols shots taken from that zone. This is then plotted into the respective zones as figure 4.6 is showing.

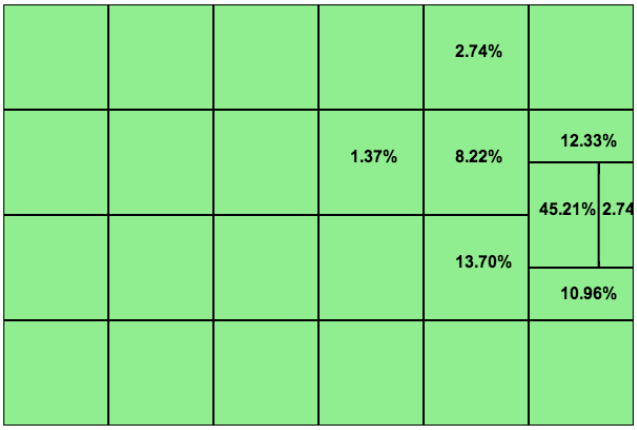


Figure 4.6: Illustrations of which zones the team has finished off their attacks from, with percent. Team - Troms IL)

Backbone comes with a library Underscore.js[7]that makes creating HTML pages with dynamic content easily. When you are rendering new content on the site you can insert data retried from a model dynamically into the HTML. In the example below the input is a an array of objects where each element in the array contains a key value pair 'name' : value. The HTML output of this will be a list of names.

```
{{#players}}
        <li>{{name}}</li>
{{/players}}
```

4.2.3.3 Router

A component not mentioned before that lies on the client is the router. The router is the glue that binds all the other moduels together. It handles the navigation between pages in the application. When users navigate on the page by clicking on links or buttons in a traditonal web page the back-end serves a new HTML page. In this page this click event is handled by a router module. The router model examins the URL request and routes the

[7]http://underscorejs.org/

request to the mapped up view. The view is then responsible for calling fetch functions on the model and render the HTML.

4.2.4 Getting players and teams into the database

Getting squads in a useful format like XML or JSON was harder than expected. On the norwegian football alliance's website you could download the squads for the current season only in PDF. Current squads is fetched from altomfotball.no, a website by the norwegian TV channel TV2. The fetching process is a own python script meant to run only once to set up the database. For each team the script basically reads the HTML document with all players listed, parse out players name, and sends in to the web server.

4.2.5 Security

Secturity is not taken into concern. This means anyone getting into the page can post new match data and add attacks. This could have been fixed by requiring a login before getting access to the site.

Chapter 5

Demo

5.1 Interfaces

5.1.1 Mockup

Figure 5.1 shows a mockup of the team analysis page that was created before starting developing. There are two main sections in the mockup: team statistics and key players.

5.1.2 Implemented interfaces

5.1.2.1 Home page

The first page you are prompted with is the listing of all matches registered in the database as figure 5.2. A click on match gives you details about that match and prompts you an interface for capturing new attacks if requested. Every field has to be submitted with a correct input value.

The whole website follows a theme throughout the pages. This theme is the standard theme in the CSS and JavaScript library Bootstrap[1]. Bootstrap makes the website by default responsive. This means the content on the site is automatically re-sized out from the size of your browser window.

[1] http://getbootstrap.com/

5.1.2.2 Registrating attacks

You registrate a new attack by pressing the New attack button. Here you fill in the result and date of the match. When you submit the match you will be sent back to the home page.

5.1.2.3 Match view

Clicking on a match gives you overview of all attacks registrated for that particular game shown in figure 5.3). This view is only meant for quality checking the data.

From the match page you can add new attack attempts, as figure 5.4 shows. This will prompt you with a fill in form for the attack. Attack start, attack end, breakthrough player, time and team needs to be filled out. The passes in the attack are added by pressing new pass. This will add a new pass to the form.

5.1.2.4 Team view

Then you have the team selection page shown in figure 5.5). This is a simple layout listing all the teams in the Norwegian premier league. From here you select the team you want to analyze.

Then there is the main view used for analyzing a team, showed in figure 5.6, figure 5.8 and figure **??**. The page design is not exactly the same as the mockup. This comes from a various things. A description of different breakthroughs has been added to enlighten users of the system. Users of the system may or may not have been involved in the process of capturing the data and therefor-extra information is needed to clarify concepts. In the mockup key players where highlighted. In the final view various statistics is presented to the user and he can then click on players he find interesting to know more about as shown in figure **??**. The whole page is divided into 3 sections; Key aspects, offensive play, defensive play.

Clicking on a player brings you to the player view shown in figure 5.10 and figure 5.11. Here individual statistics is highlighted.

5.2 Capturing process

As mentioned, the process of capturing data is manually. In the beginning it took up to 1 hour to capture all attacks for a match. When you get used to the interface and is able to quickly identify players the time used went down 15-20 minutes. Where most of the time went was getting the players id by looking up in the database manually. The time spent also depends on which match you are capturing. For example when a top team meets a team in the bottom of the table there is usually more attacks to capture.

The process of capturing has been the following:

1. Download the match video.
2. Find the match in the VGLive.no archives. VGLive.no is used to quickly find all attacks ending with a finish
3. Capture all the attack one by one via the implemented interface mentioned earlier.

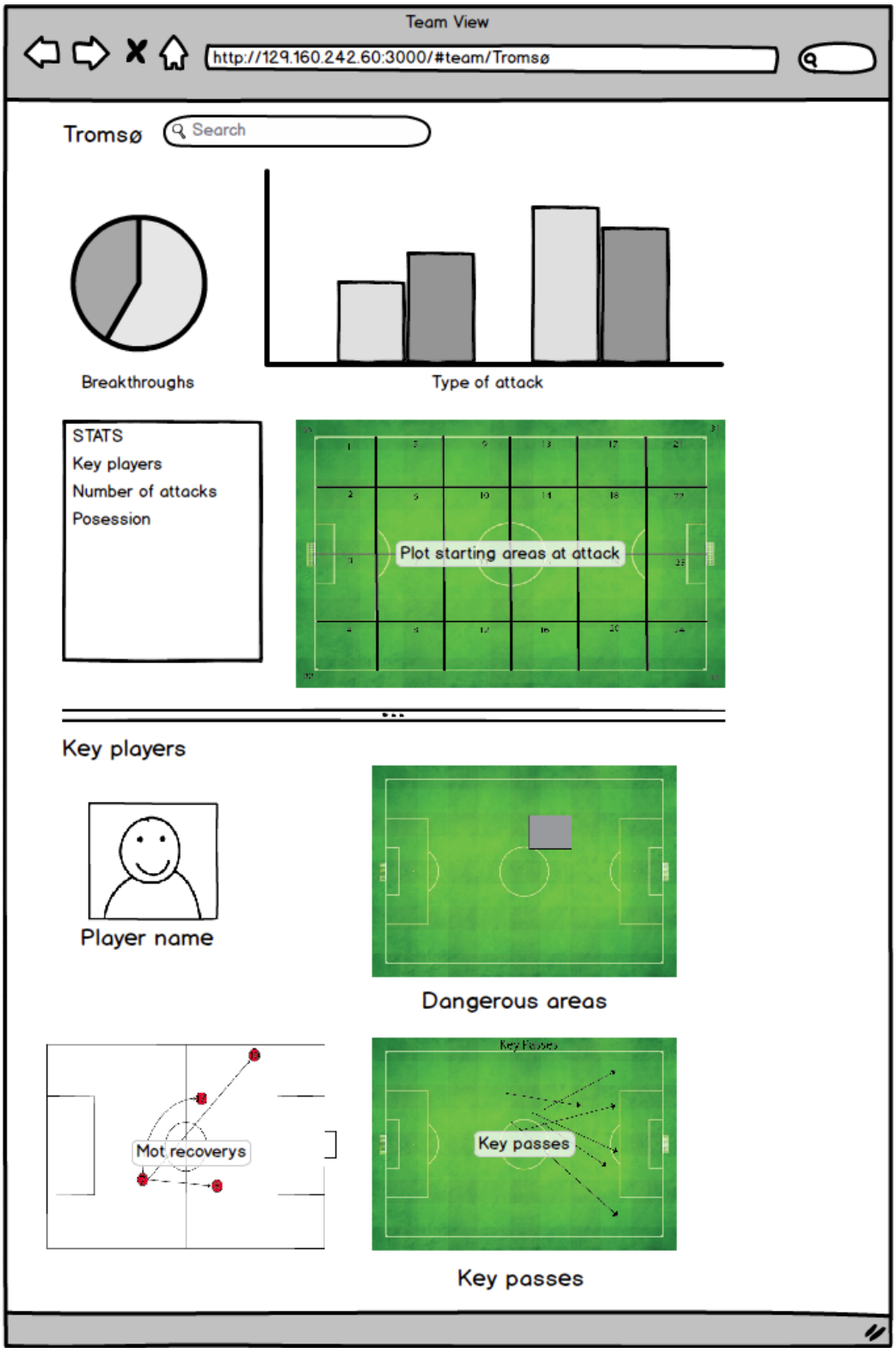


Figure 5.1: A mockup of the main analytic page

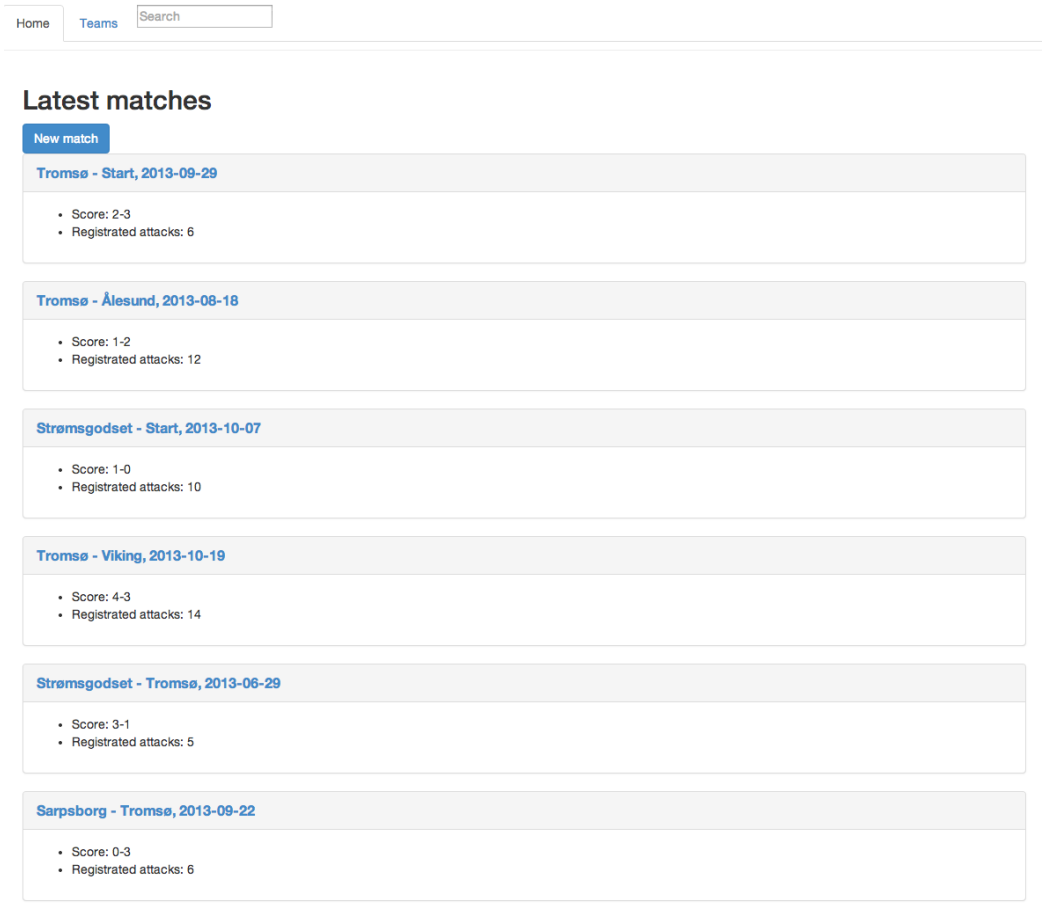


Figure 5.2: All matches registrated in the database are listed on this page

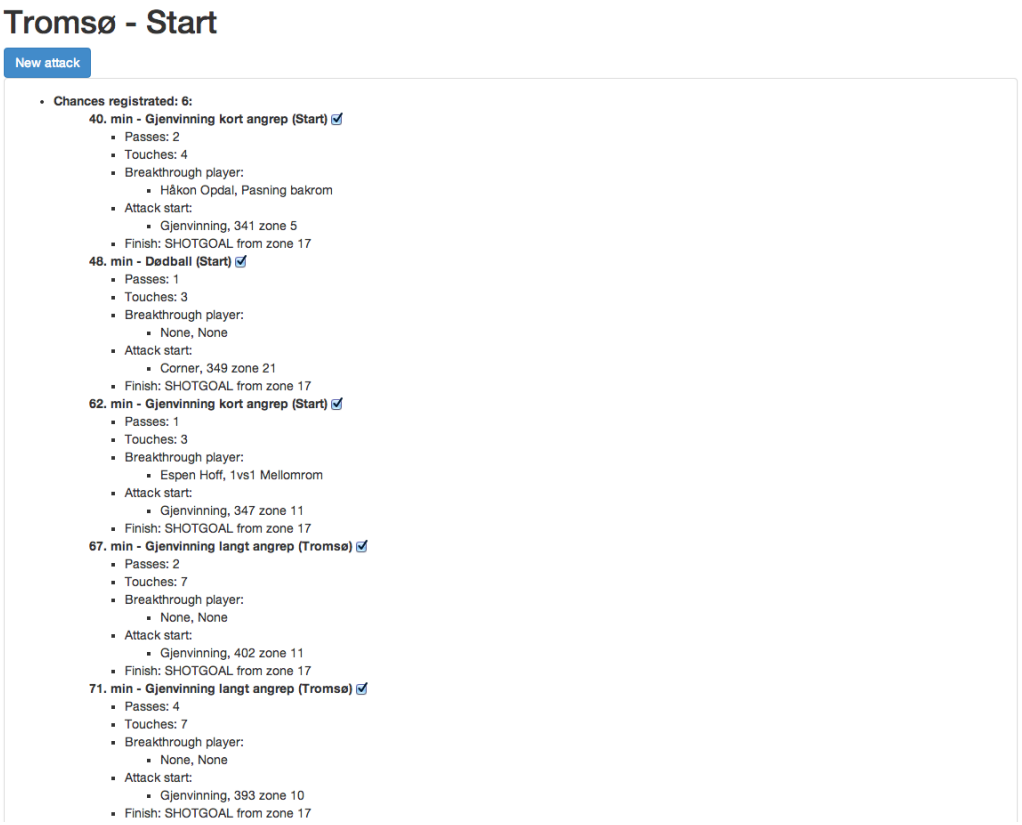


Figure 5.3: Interface listing up all attacks for a match

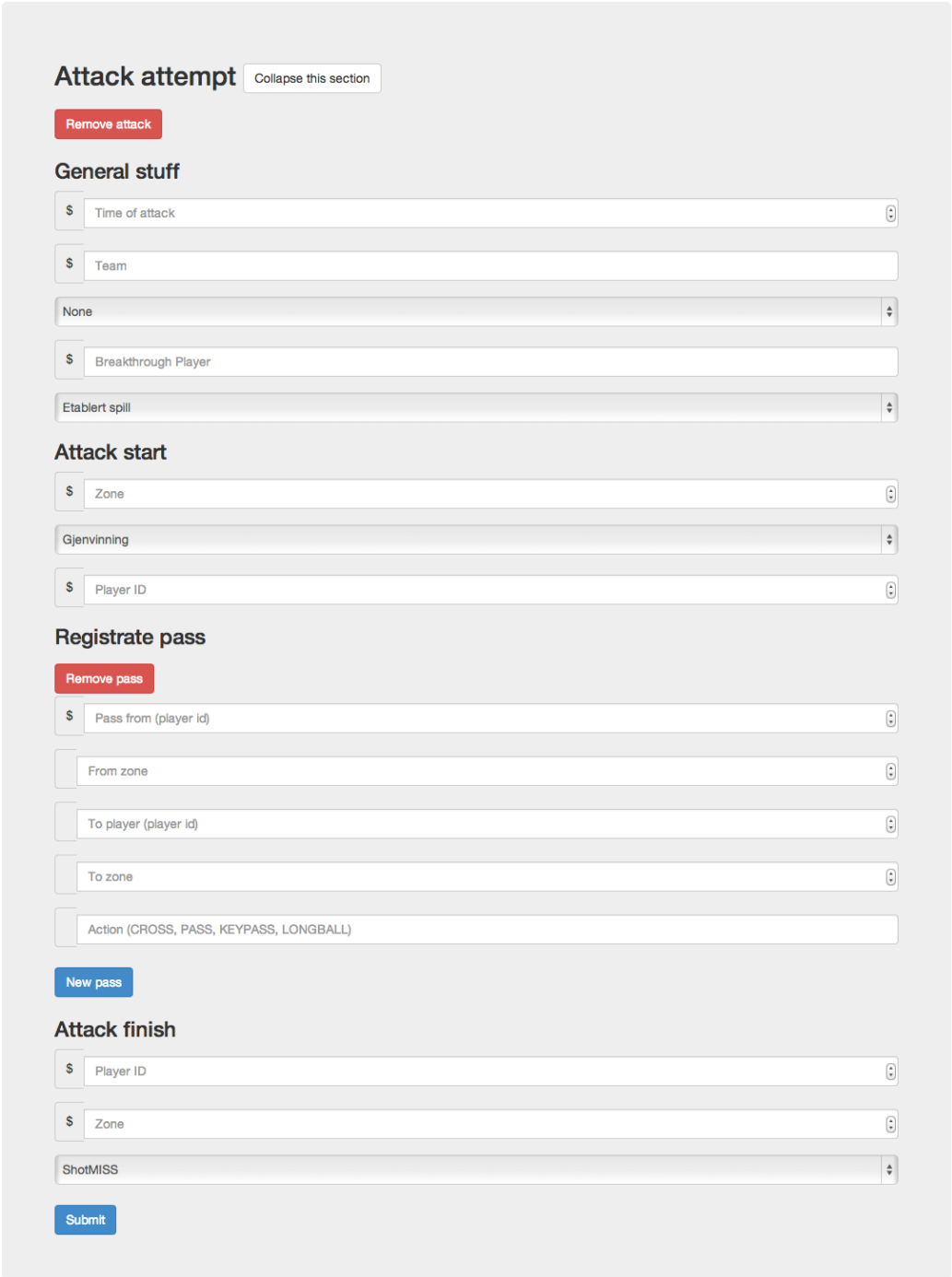


Figure 5.4: Interface for registrating an attack

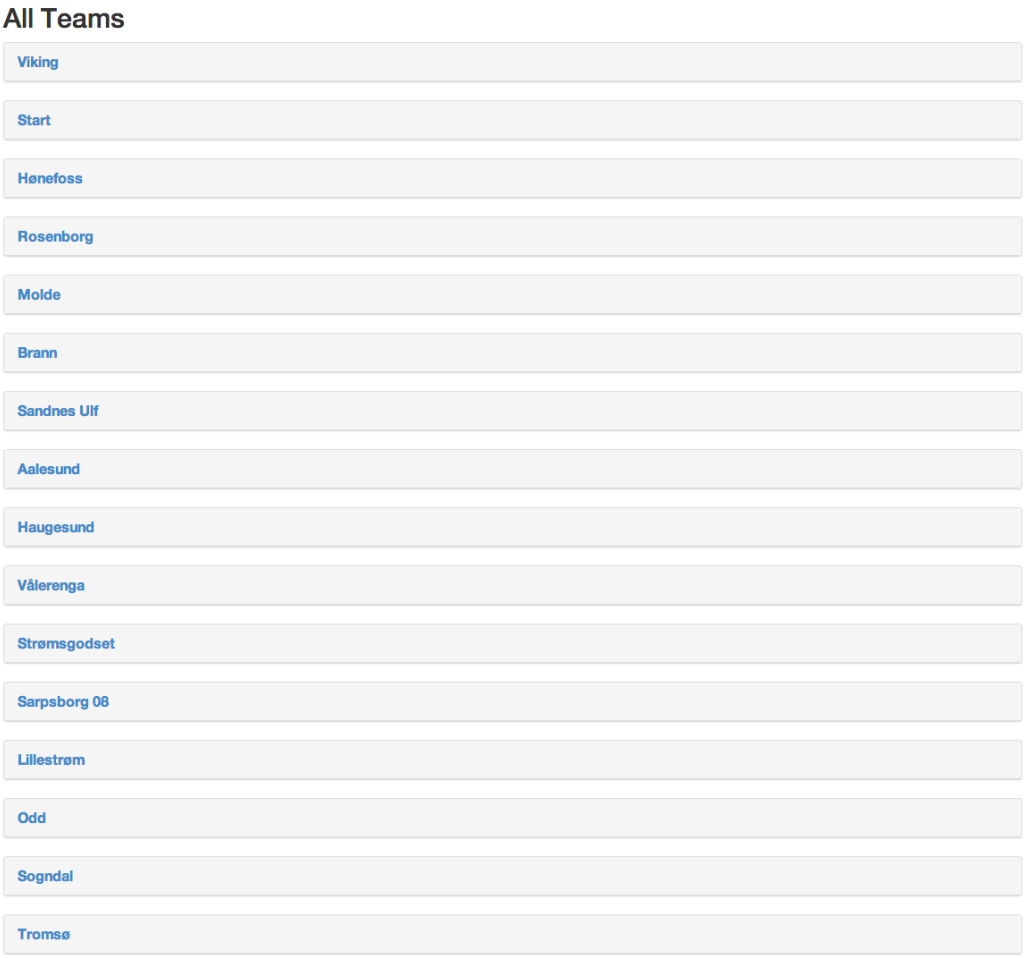


Figure 5.5: Interface listing all teams

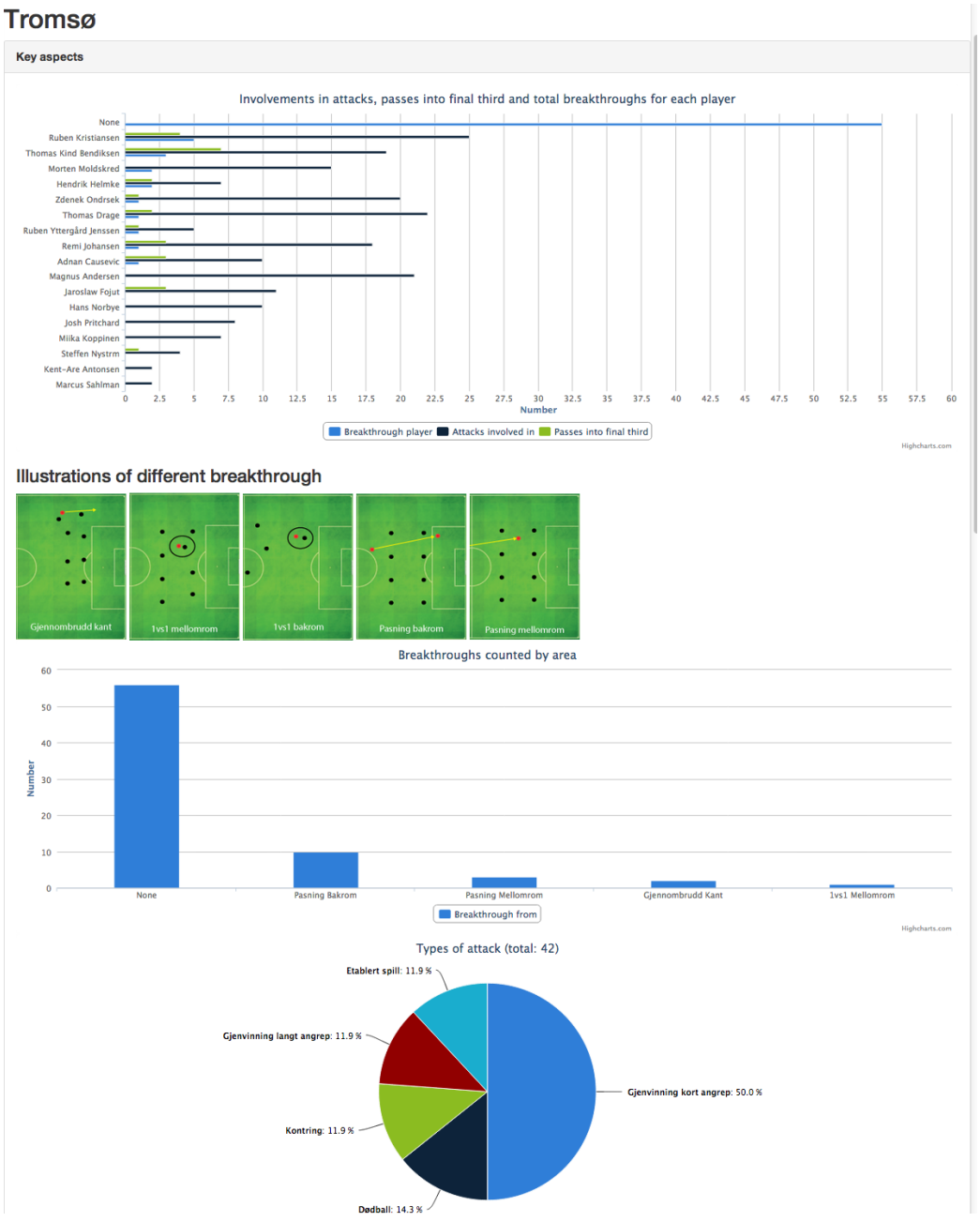


Figure 5.6: Main interface for information about opponents - key aspects

Figure 5.7: Main interface for information about opponents - offensive play

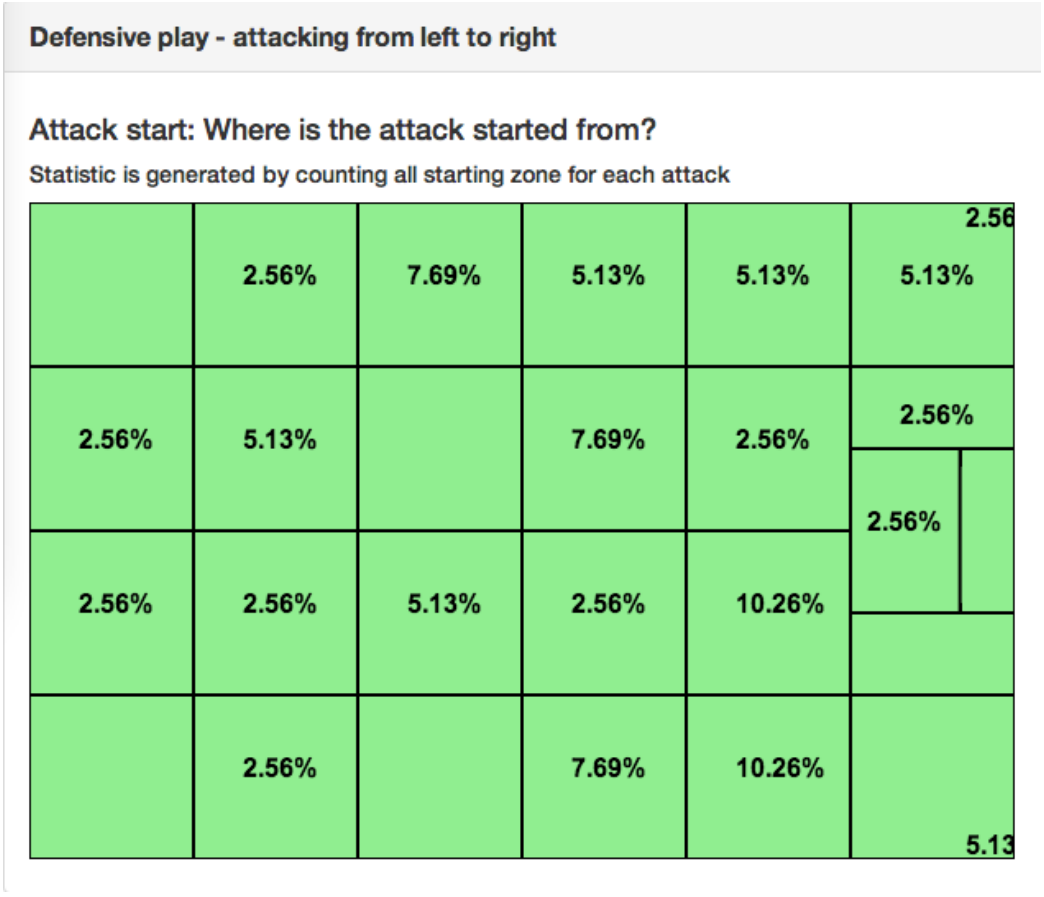


Figure 5.8: Main interface for information about opponents - defensive play

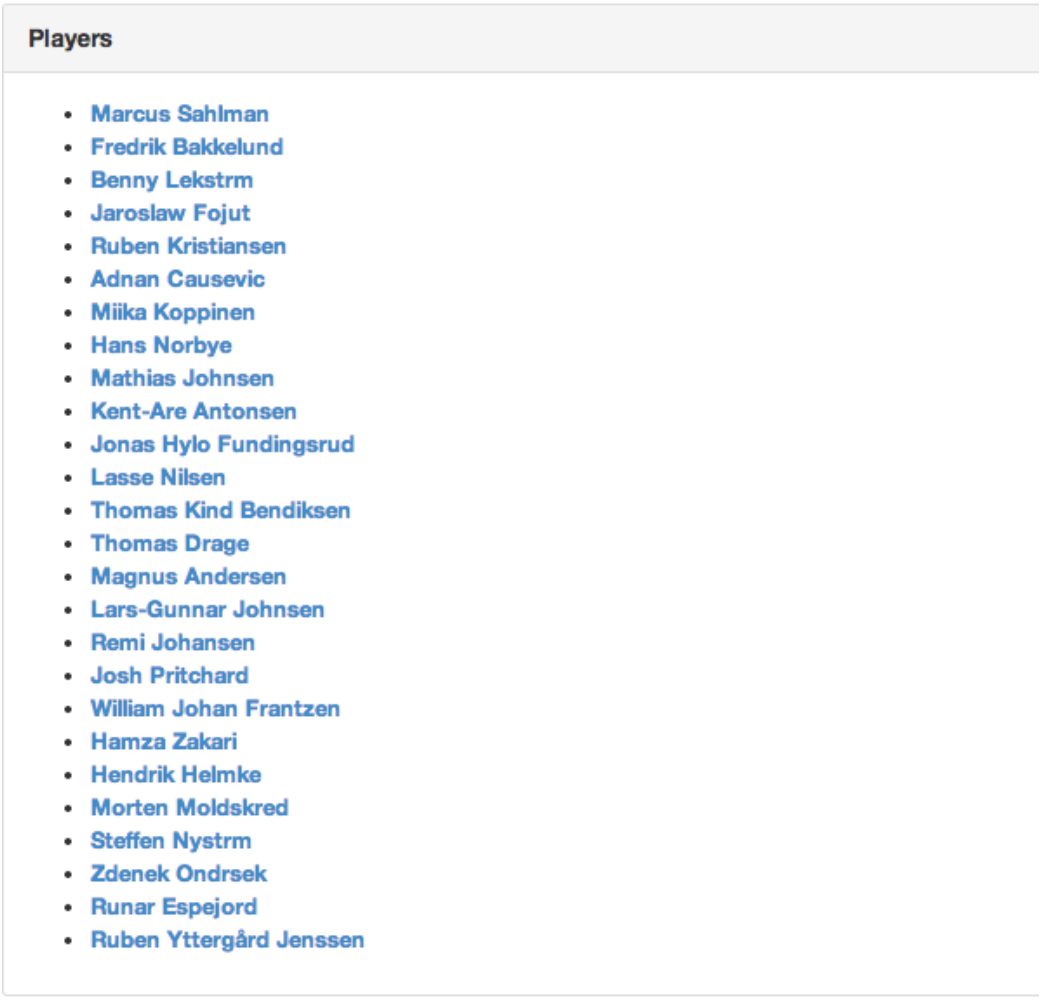


Figure 5.9: Main interface for information about opponents all players

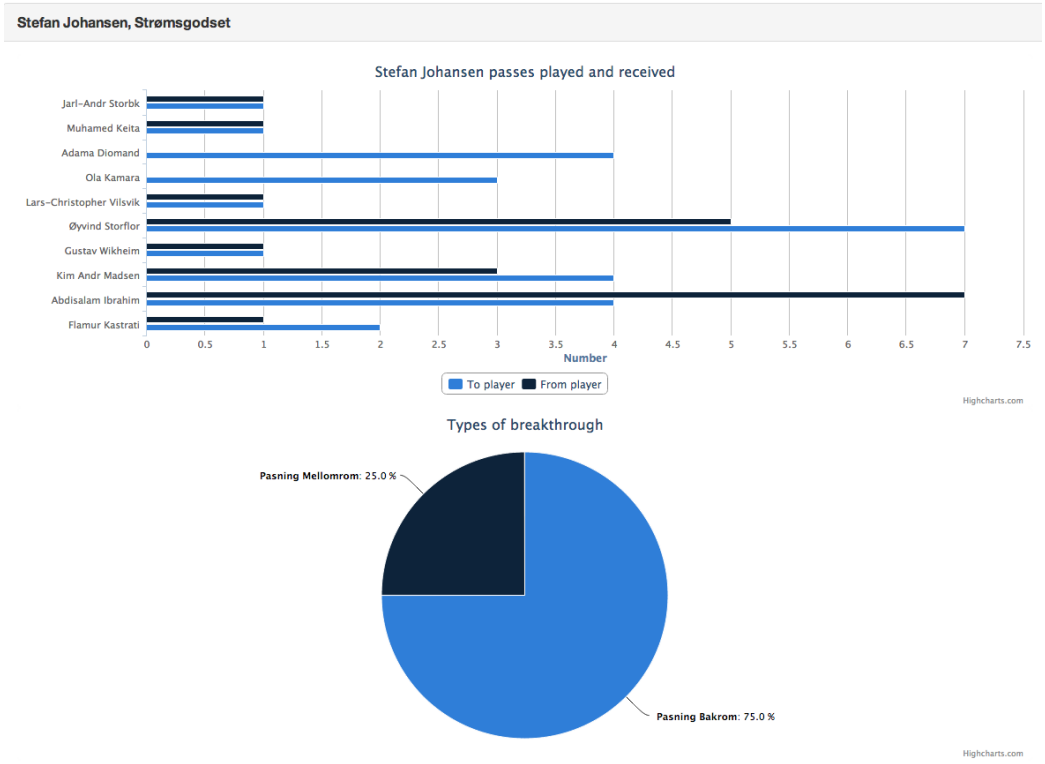


Figure 5.10: Player view highlighting individual statistics

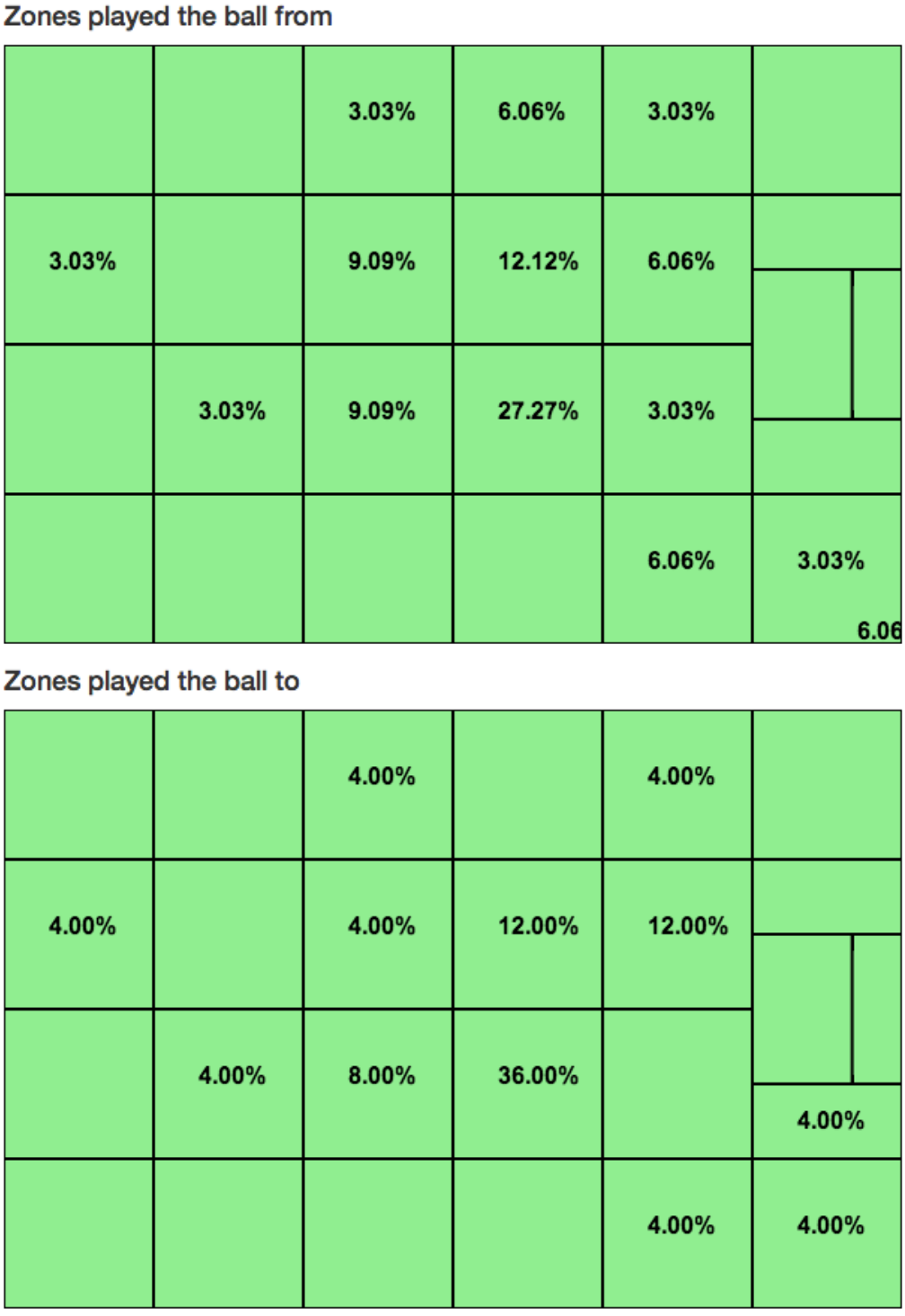


Figure 5.11: Player view highlighting individual statistics

Chapter 6

Evaluation and Results

This chapter presents methods used to evaluate the system and the results collected evaluating the system.

6.1 Methods

6.1.1 User Survey

The system is measured by doing user surveys on the end users. In our user survey we have coaches rating how much they agree with a statement on a Likert-scale. The statements compare the system against other systems in use at Alfheim today. The Likert-scale chosen is a 5 point scale from *strongly disagree* to *strongly agree*.[1]. In short it will let each individual to note how much they disagree or agree with a particular statement.

Including to this external people not in the Troms IL system. This includes Ruben Yttergrd Jenssen playing in the national soccer team for Norway and FC Kaiserslatern in 2. Bundesliga, and Lars Tjrns, a previous coach and now football expert in Canal+ also writing for 100 fotball

[1] http://www.simplypsychology.org/likert-scale.html

6.2 Experiments and Results

6.2.1 Test data

In the tests Troms IL academy coaches has been evaluating the system. In the evaluation phase the database had been populated with data from Troms IL and Strmsgodset Toppfotball matches. Only attacks from these two teams have been used in the evaluating process. A total of 34 attacks have been captured for Strmsgodset over 5 matches. For Troms a total of 42 attacks have been captured over 9 matches. Figure 6.1 lists all matches. Some matches include data from other teams than Troms and Strmsgodset. They have not been taken into consideration in the evaluating process.

Match	Attacks registrated
Viking – Tromsø, 2013-05-26	8
Strømsgodset – Tromsø, 2013-06-29	5
Molde – Tromsø, 2013-06-10	14
Strømsgodset – Vålerenga, 2013-05-10	8
Strømsgodset – Start, 2013-10-07	8
Tromsø – Vålerenga, 2013-08-04	15
Tromsø – Start, 2013-09-29	6
Tromsø – Viking, 2013-10-19	14
Tromsø – Ålesund, 2013-08-18	12
Tromsø – Strømsgodset, 2013-11-03	15
Sarpsborg – Tromsø	5
Sogndal – Strømsgodset, 2013-09-27	13

Figure 6.1: Matches that have been captured and persisted into the database

6.2.2 SAT as a tool for opponent analytic

SAT gives you valuable information about opponents that the current systems you use today doesn't provide

6.2.3 SAT as a tool for identifying key players

6.3 Discussion

6.3.1 Input

As the input is manual the current biggest limitations is humans. The system requires a lot of manually work to be operational as an opponent analytic tool. Up to one hour is the normal time spent capturing all data for a match with the current interface. As mentioned in SECTION you have to store the players by their ID. IDs are only found in the database. Instead of this a player selector interface could have been developed letting you just click on a image of the player involved. We suggest you can save up to 20 minutes by having this feature added.

There is minimal quality checking of the input data except from the match view page that lists every attack captured, meant for external operators to verify the data. Other than that you have to trust the operator that is capturing match data. The input is to some degree subjective for some data like identifying breakthroughs. In some situations one operator may say that it was a breakthrough and another wouldn't agree. This may not be the biggest problem if you set some rules to follow for the operators.

Chapter 7

Conclusion

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

In this project, develops and evaluates a system for capturing, persisting and presenting information in the field of soccer opponent analysis. Particularly we have focused on identifying key players in the opponent team

This thesis will develop a system complementing the Muithu and Bagadus systems. Focus will be on soccer opponent analytics, where a data repository need to be developed capturing important events relevant for this type of analytics. Specially we want to identify the key players in a team. A user interface component providing the core information about the opponent should also be developed.

In the requirement specification we stated what we wanted from the system; a system that identifies key players in the offensive play of a soccer team. We also wanted to be able to see where on the field the key players contribute from.

7.0.2 Concluding Remarks

7.1 Future Work

References

[1] Valter Di Salvo Adam Collins Barry McNeill Marco Cardinale. Validation of prozone ®: A new video-based performance analysis system. *mordi*, 2006.

[2] D. E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, and Paul R. Young. Computing as a discipline. *Commun. ACM*, 32(1):9–23, January 1989.

[3] Håvard D. Johansen Svein Arne Pettersen Pål Haloversen and Dag Johansen. Combining video and player telemetry for evidence-based decisions in soccer. *Regional Centre for Sport, Exercise and Health - North*, 2013.

[4] Martin Hardy. How the science of opta and prozone's statistics are changing the premier league, Nov 2011.

[5] Jonathan Howard. Sports science and big data opportunities, October 2013.

[6] Jason Turbow. Soccer embraces big data to quantify the beautiful game, June 2012.

[7] Mark Venables. Sportstech: Football, 2013. [Online; accessed 12-November-2013].