Applicon Fussball

Bergur Logi Lúðvíksson
Hafþór Örn Þórisson
Högni Rúnar Ingimarsson
Stefán Arnar Einarsson

Spring 2015

Teacher: Hallgrímur Arnalds
Instructor: Birgir Kaldal Kristmannsson
Examiner: Haukur Kristinsson

Table of Contents

1 Introduction

This report is meant to describe the main aspects and describe the process of the final project that was done in cooperation between the software company Applicon and Reykjavík University (hereafter “RU”). The project was done by Bergur Logi Lúðvíksson, Hafþór Örn Þórisson, Högni Rúnar Ingimarsson and Stefán Arnar Einarsson who are all students on their final year of computer science BSc at RU. Our main contact at Applicon was Gunnlaugur Einarsson. The project instructor was Birgir Kristmannsson. Applicon is a software company that develops and implements software solutions for companies and institutions of all sizes and shapes. The main focus of the company is developing solutions and creating adjustments solutions for SAP and Vigor software. The company has also been focusing on developing and maintaining the human resources and salary system called Kjarni.

Image 0: The table

2 Project description

The project, as described by Applicon, was to design and build a prototype of an application for a fussball table. They wanted an ARM based microcomputer put in their fussball table which would be capable tracking table activity via sensors. The ARM computer should gather data and be able to display it in a web application. The web application should be able to create a new game, register players, to display a live score of the current game and display statistics for all users. All data from all games and all users should be stored in a SAP HANA database.

2.1 Hardware

The only hardware needed was the Raspberry Pi ARM microcomputer along with the sensors needed to track goal scoring in the table. A Raspberry Pi Model B+ was provided to us by Applicon but how the sensors were to be made was unclear at the start of the project so Gunnlaugur left it up to us to design and build them.

Image 1: Raspberry Pi Model B+

2.2 Software

In order to build the entire system a number of programs and software was required to build the different parts of the solution. We mainly used Eclipse IDE, Sublime Text 2 and Putty (an SSH client for windows). At first Applicon wanted us to use SAPUI5 (which is SAP´s own jQuery library) and implement the website entirely using that. However once development started on the website it was changed from SAPUI5 to AngularJS which led to the switch from Eclipse to Sublime Text 2. However Eclipse was still used to create the backend on the SAP HANA platform.

2.3 Product vision

The vision of the group for the product was to once and for all determine who the best fussball player is at Applicon as they have had no solid way of telling who that is as of now. The product will also inspire more people to start playing fussball and increase the level of competition between current users.

3 Project plan

This chapter will focus on what methods and disciplines the group used to get the project done in time.

3.1 Methodology

The team decided to use the scrum methodology for the project. The reason being that all group members had used it before in a course that focused on agile methodologies anf also that a group of four is pretty well suited for scrum. Scrum offers elasticity which is good because all team members had other courses to take care of during the semester alongside this project. Scrum also supports all of the documentation required for this project.

It was decided right away that Bergur Logi would be scrum master as he very well organized and has good communication skills. Being a scrum master, Bergur had reduced work capacity planned for each sprint as some time would have to go into scrum master related work. Gunnlaugur would serve as the product owner and the rest of the team would mainly focus on working on finishing stories in the product backlog along with Bergur.

Scrum Roles	
Scrum Master	Bergur Logi Lúðvíksson
Product Owner	Gunnlaugur Einarssson (Applicon)
The Team	Hafþór Örn Þórisson, Högni Rúnar Ingimarsson, Stefán Arnar Einarsson

Product owner: Negotiate priorities, set project scope, accepts or rejects the final product
Scrum master: Remove all internal and external impediments, plan team meetings, promote that the team maintain the scrum principles, keep track of time and the stories
Team: Work on stories in the product backlog, take part in team meetings, run product tests

3.2 Attendance Times

The team decided to stick roughly to the office hours of Applicon, mainly due to the fact that the building was unaccessible very late in the evening. We knew that we had to use most of our free time on the project as all members of the team had a heavy course load for the semester.

Applicon provided us with 4 desks, monitors along with company owned computers which we were allowed to use on the project. Nearly all of our hours put in the project were when we were located at Applicon.

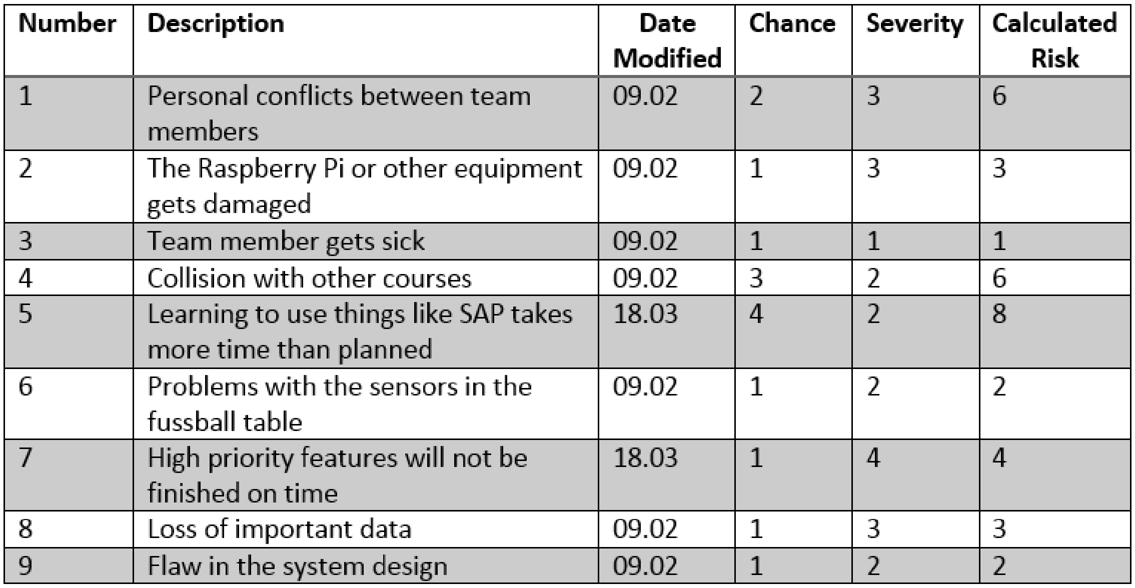
3.3 Sprints

The project was divided up into 9 sprints total. The first 5 sprints would be 2 weeks long each. During the 6th sprint we had our final exams so it was decided to make that sprint 3 weeks long in order to keep a similar capacity throughout all of the sprints. The last 3 sprints were then one week long as for the last 3 weeks none of us had any other projects due or other courses to focus on so we would be able to get in a hefty amount of work done in those weeks. After each sprint a retrospective meeting would be held in order to see what could be done better and what went well in the sprint.

3.4 Risk Analysis

During projects like this there are always some risks that the team has to be prepared for. Some of them we predicted and but other risks were unforeseen. We kept the analysis live and updated it when needed. The purpose of the risk analysis is to make precautions and a reaction plan to be ready for every risk.

Our analysis consisted of both internal and external risks. Some of them were related to the software and equipment. External risks were f.ex. collisions with other courses and when a team member gets sick.



Number	Description	Date Modified	Chance	Severity	Calculated Risk
1	Personal conflicts between team members	09.02	2	3	6
2	The Raspberry Pi or other equipment gets damaged	09.02	1	3	3
3	Team member gets sick	09.02	1	1	1
4	Collision with other courses	09.02	3	2	6
5	Learning to use things like SAP takes more time than planned	18.03	4	2	8
6	Problems with the sensors in the fussball table	09.02	1	2	2
7	High priority features will not be finished on time	18.03	1	4	4
8	Loss of important data	09.02	1	3	3
9	Flaw in the system design	09.02	1	2	2

Image 2: Risk analysis

4 Technical design

As mentioned before the system boils down to 3 main parts which are a Raspberry Pi, the website and a SAP HANA database. We will first look at the Raspberry Pi.

4.1 Raspberry Pi

The Raspberry is set up with the NOOBs Raspbian operating system. As it was very easy to set up we decided to use that operating system. The Raspberry is running two scripts one of which is for the sensors and the other is an API that communicates with the SAP HANA backend. The sensor script is written in Python 2.7.3 and has very little functionality other than calling the API when a goal is scored. The API is written in NodeJS and it takes care of all the logic for each game. It communicates with the website as to which players are selected for a game and keeps track of score and game state (whether a game is on or not).

We connected the sensors via the GPIO pins on the Raspberry and when the sensor is triggered, it sends a signal to the Raspberry which is listening to those specific pins. There are also two buttons connected to the Raspberry that do the exact opposite of the sensors but those are used to decrement goals if an error occurs in a game (e.g. an invalid goal is scored or someone accidentally hits the sensor). At first we decided to go with motion sensors to track the ball as it rolled down a small ramp within the table. However a closer examination of the table led to us noticing that it is very dark inside the table and very little room to place motion sensors along this ramp. It was then decided to use trigger sensors which triggers when weight or pressure is applied to the top of them. These were then put in between two cards to increase the surface coverage of the sensor so that the ball would trigger it as it fell out of the small ramp.

4.2 SAP HANA

The SAP HANA project has two layers, one is the database layer which holds our tables and our calculation views the other one is the xsjs layer which we use as a middleman between our database layer and our website and API, it also takes care of calculating the ELO and updating the tables.

Our SAP HANA in-memory database has three tables, one for users, one for goals and one for games which are all column store. Then we build our calculation views on top of those tables to calculate the desired statistics.

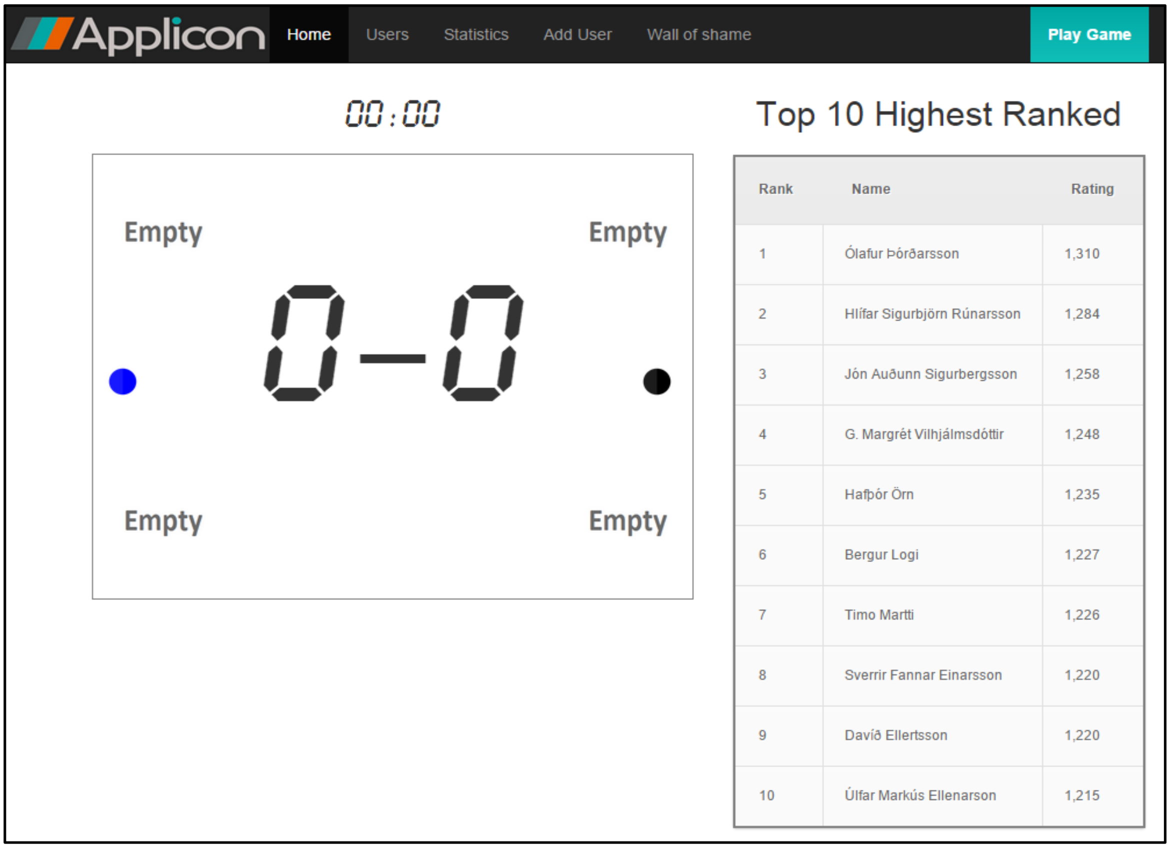
When we create a new game we do a post call from our API to a xsjs file with the game information. From there we open up a connection to the database and send in a sql query that adds the new game to the database.

4.3 Website

For the website we used HTML, AngularJS and CSS. It's main function is to provide a place for players to register for games, see a live score of the current game and see statistics for each and every player. The website queries the database for all of the data to display except for the current score of the game which is provided, as said before, by the Raspberry Pi API. On the website you can browse the list of all users and display statistics about them. You can also see a system statistics page which displays information, for example total amount of games played on the table, total amount of goals scored along with a variety of top 10 lists.

4.3.1 Home

This is the start view on the website and it is designed for spectators. When there is an ongoing game the Top10 table dissappears and new table with the goal history of game appears. The blue and black dots represent the blue and the black team.



00:00

Empty Empty

0 – 0

Empty Empty

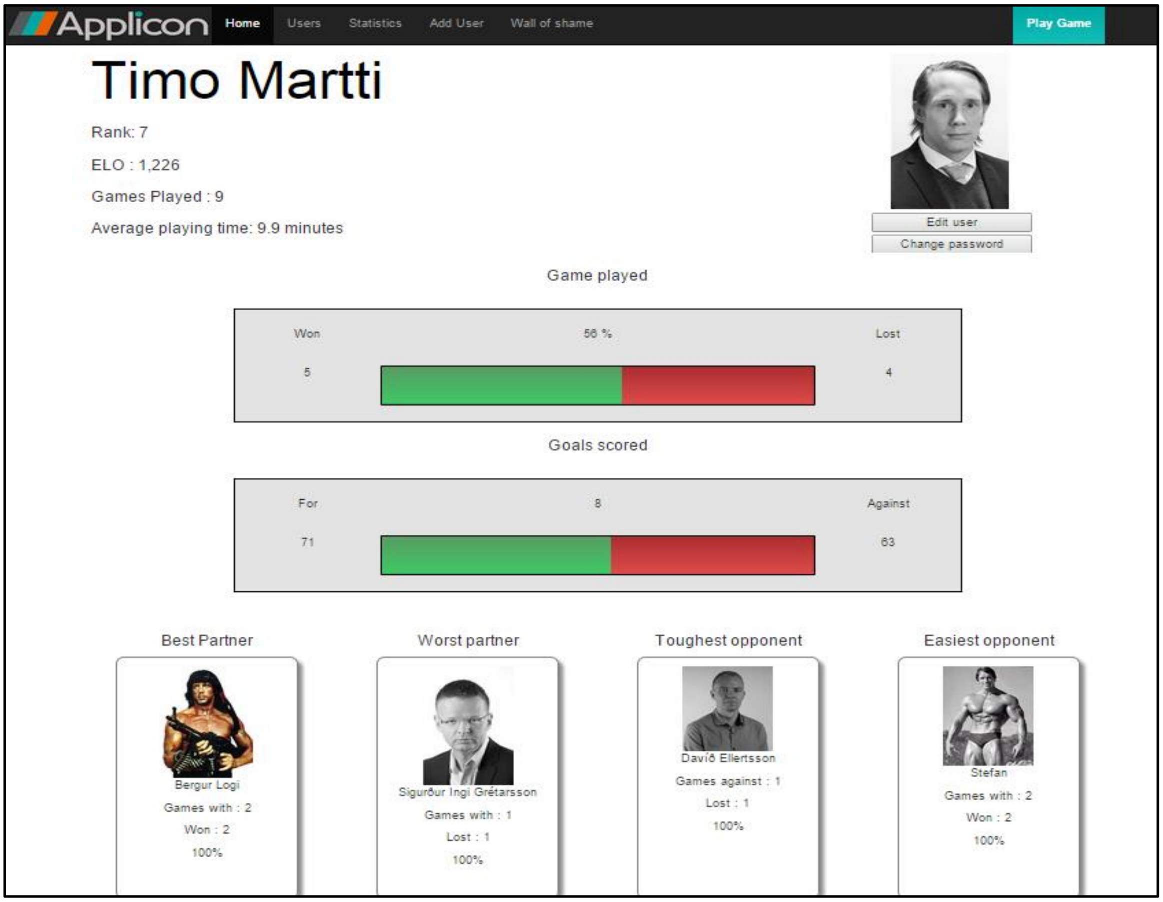
Top 10 Highest Ranked

Rank	Name	Rating
1	Ólafur Þórðarsson	1,310
2	Hlífar Sigurbjörn Rúnarsson	1,284
3	Jón Auðunn Sigurbergsson	1,258
4	G. Margrét Vilhjálmsdóttir	1,248
5	Hafþór Örn	1,235
6	Bergur Logi	1,227
7	Timo Martti	1,226
8	Sverrir Fannar Einarsson	1,220
9	Davíð Ellertsson	1,220
10	Úlfar Markús Ellenarson	1,215

Image 3: Home view

4.3.2 User

This is the view you get when you select a user. On this site there is a lot of statistics about the player, like: win ratio, game history, best partner and toughest opponent.



Timo Martti

Rank: 7

ELO : 1,226

Games Played : 9

Average playing time: 9.9 minutes

Game played

Won 5 — 56 % — Lost 4

Goals scored

For 71 — 8 — Against 63

Best Partner

Bergur Logi
Games with : 2
Won : 2
100%

Worst partner

Sigurður Ingi Grétarsson
Games with : 1
Lost : 1
100%

Toughest opponent

Davíð Ellertsson
Games against : 1
Lost : 1
100%

Easiest opponent

Stefan
Games with : 2
Won : 2
100%

Image 4: User view

4.3.3 Statistics

This site shows the table statistics and 14 Top10 lists of various statistics.

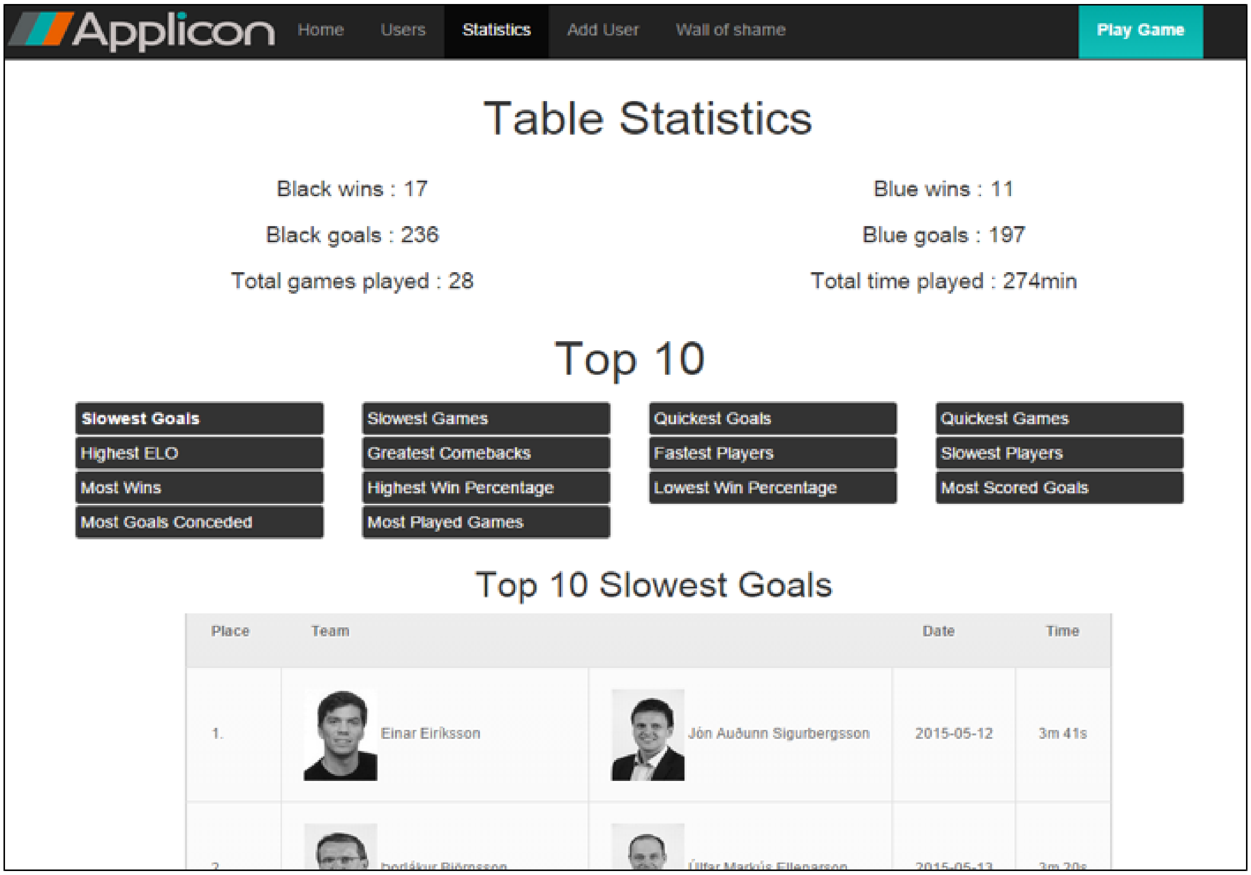


Table Statistics

Black wins : 17

Black goals : 236

Total games played : 28

Blue wins : 11

Blue goals : 197

Total time played : 274min

Top 10

Top 10 Slowest Goals

Place	Team		Date	Time
1.	Einar Eiríksson	Jón Auðunn Sigurbergsson	2015-05-12	3m 41s
2.	Þorlákur Björnsson	Úlfar Markús Ellenarson	2015-05-13	3m 20s

Image 5: Statistics view

4.3.4 Play Game

Here you can start a game and select players into the blue and black teams.

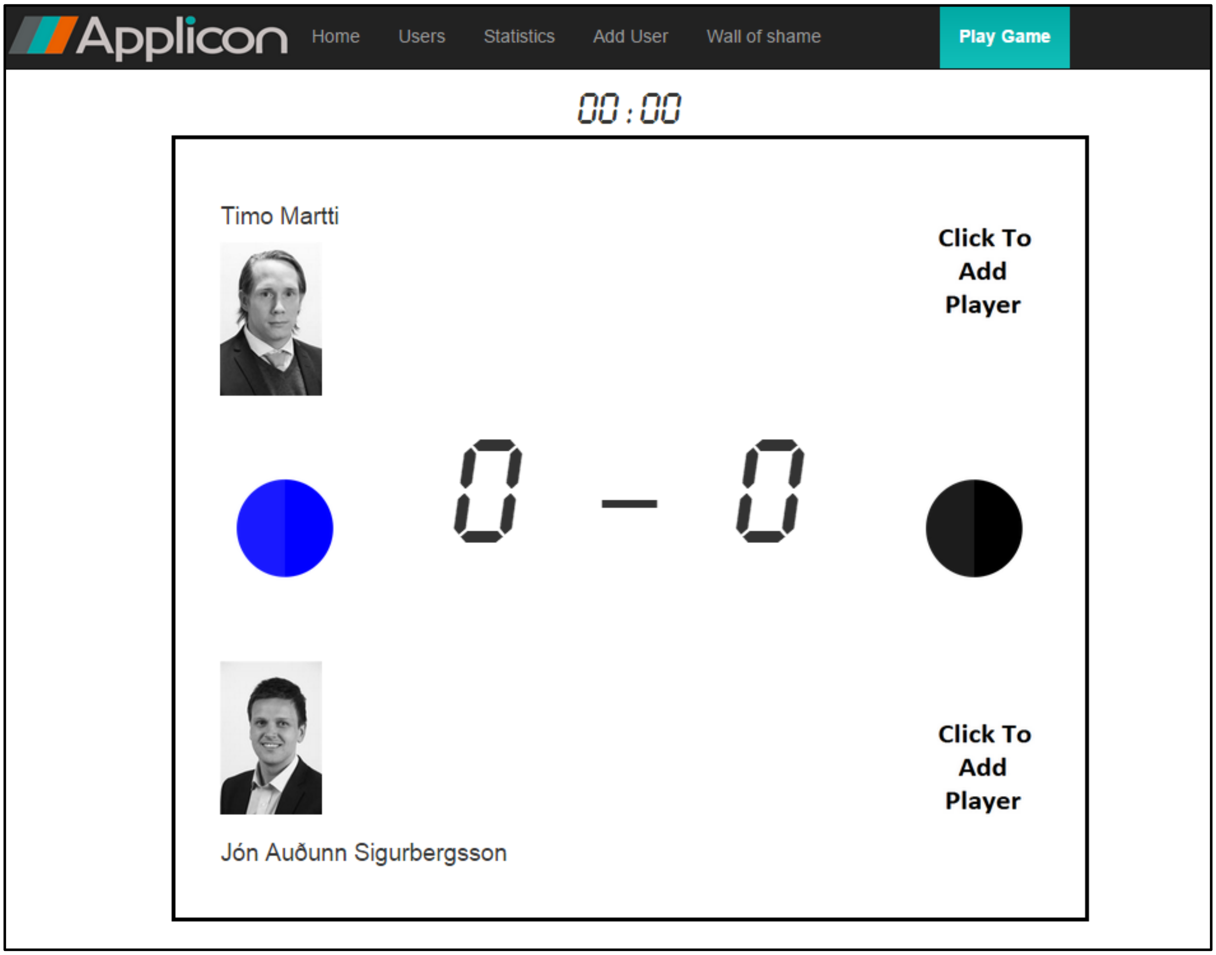


Image 6: Play Game view

5 Progress Overview

The project ended in 1352 hours and ten minutes spent by across all team members. Below is a chart that shows how many hours was spent on each spring along with burndown charts and descriptions for each sprint.

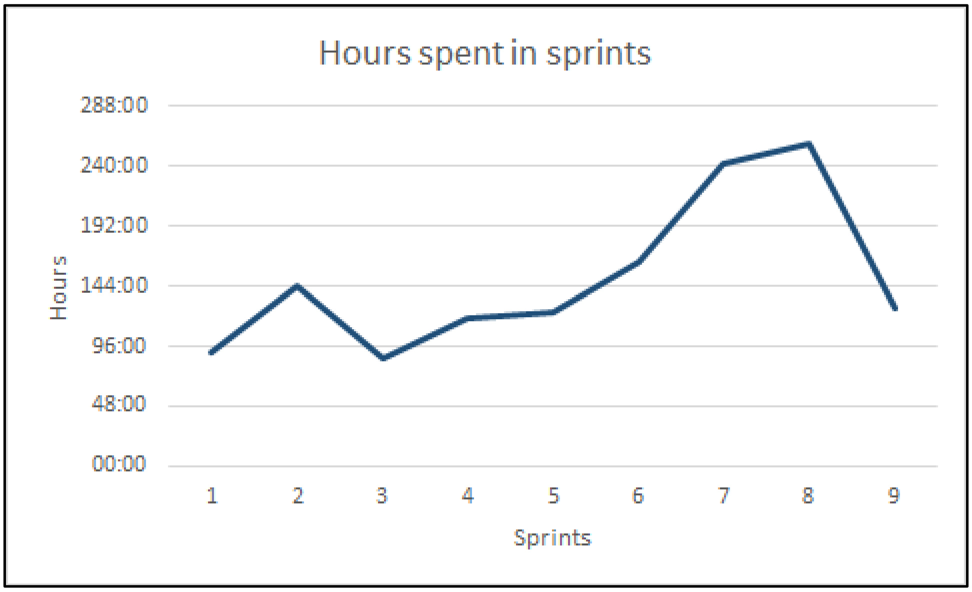


Image 7: Times per sprint graph

5.1 Sprint 1. In a galaxy far far away. 1/26/2015 - 2/8/2015

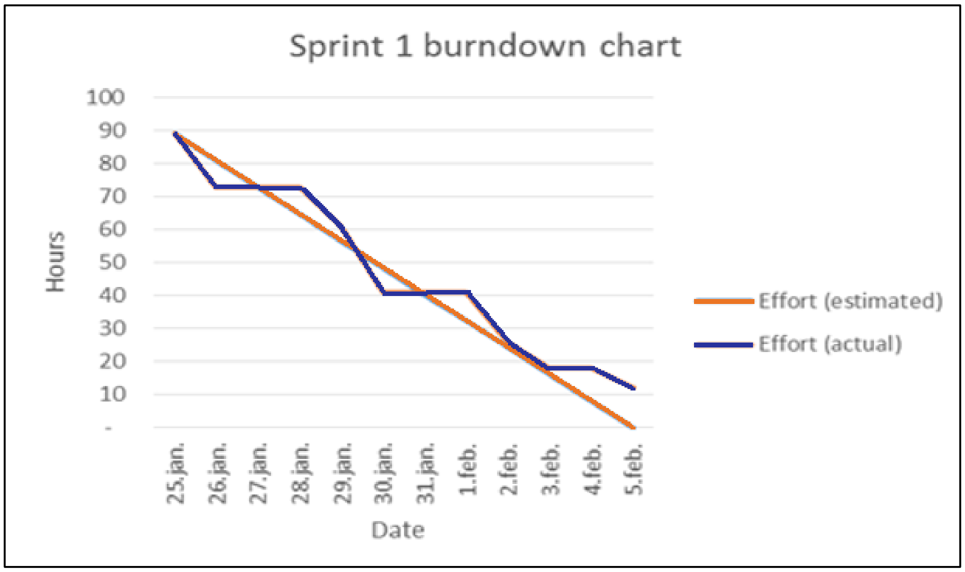


Image 8: Sprint 1 Burndown Chart

The goal was to setup an operating system on the Raspberry and get it to work with buttons and make a temporary API that sends data to a temporary database.

We managed to get the raspberry going quickly with the linux NOOB OS. The buttons worked brillianttly and displayed the score on a raw website. In this sprint we could have divided the work better between team members. After that we planned to never have more than two working on the same thing.

5.2 Sprint 2. The force is strong with this one. 2/9/2015 - 2/22/2015

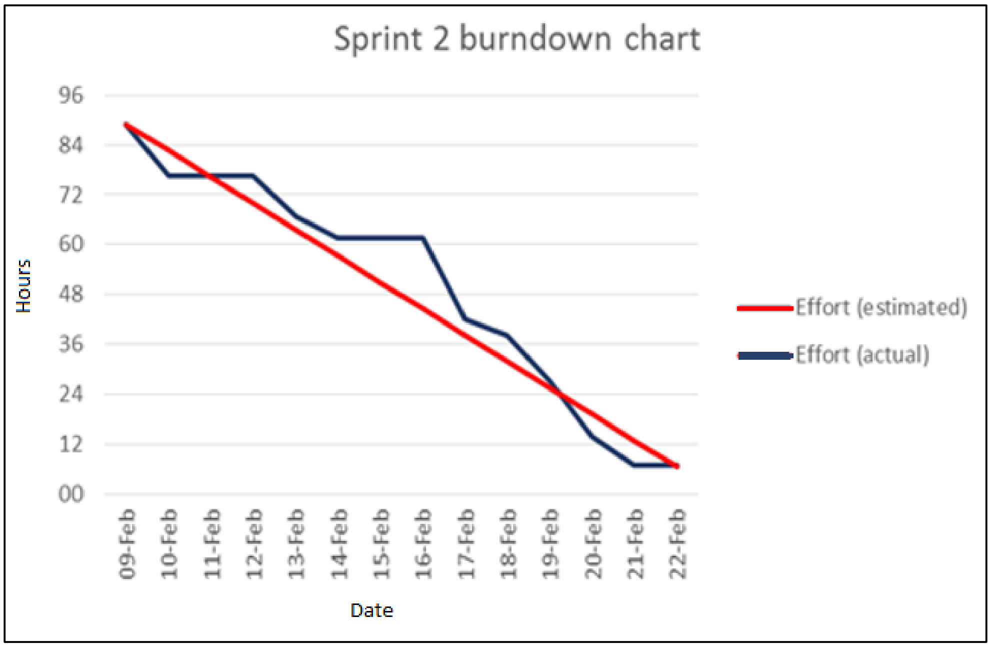


Image 9: Sprint 2 Burndown Chart

In this sprint the goal was to learn how to develop on the SAP:HANA platform and set up project for our system. Get all necessary user authorities to be able to develop on the SAP:HANA.

Learning the SAP:HANA development was made easier because of online courses on the open.sap website. However there were unforeseen problems with the access on our working computers and a connection problem between local workspaces and the SAP:HANA system. We kept on going by overcoming those obstacles in our path.

5.3 Sprint 3. The Phantom Menace. 2/23/2015 - 3/8/2015

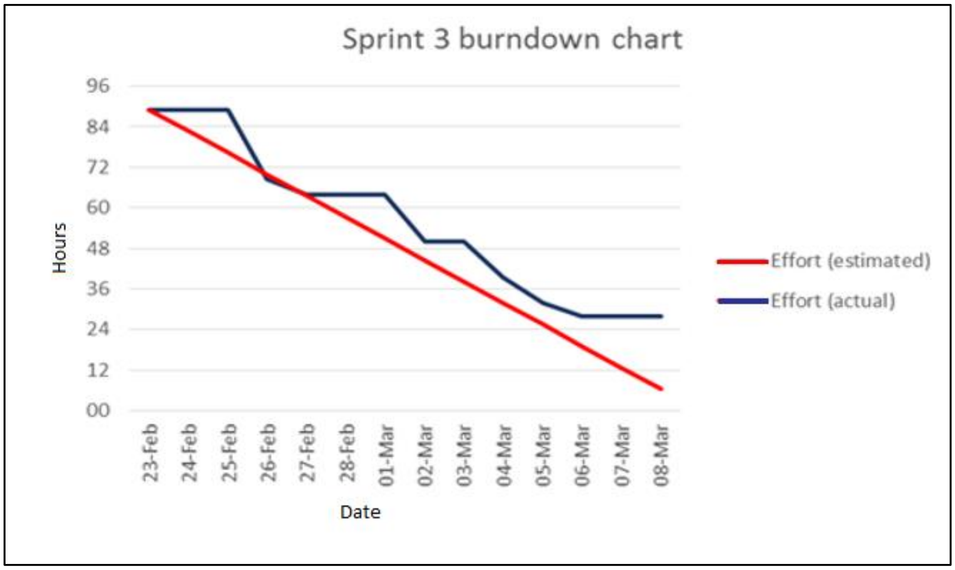


Image 10: Sprint 3 Burndown Chart

The goal was to keep on the development in the backend and create a working API on the SAP:HANA system. Create a server on the Raspberry and create necessary tables and build a connection between the server and the system.

We managed to create the xsjs files with functionality and making the sql queries for the database. But we had some problem with getting post privileges to our xsjs files. We got some help from the Applicon employees and it took some tampering to get it to work right.

5.4 Sprint 4. Attack of the Clones. 3/9/2015 - 3/22/2015

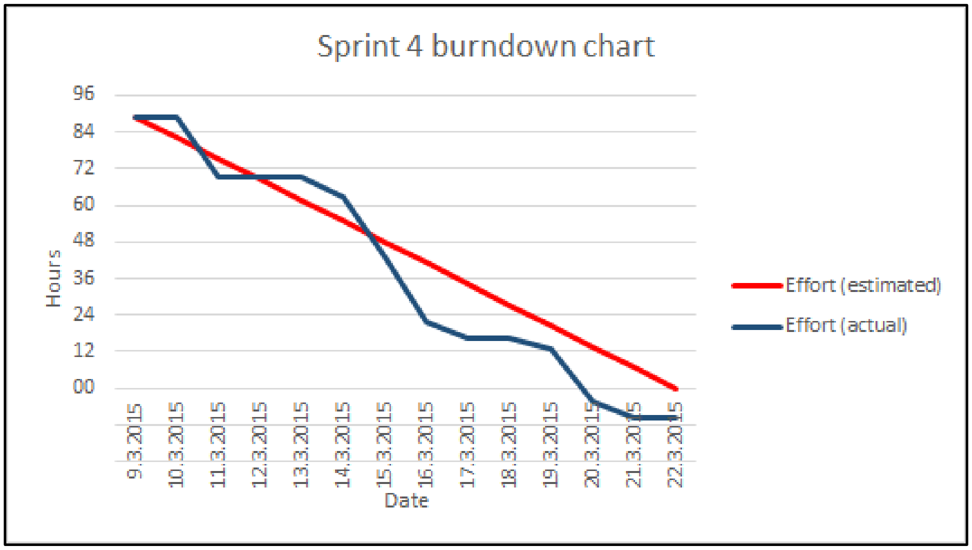


Image 11: Sprint 4 Burndown Chart

At this time the backend was at the stage we could start to develop the frontend. The goal was to get some progress in the frontend with the SAPUI5 javascript library.

After many hours of watching online video courses about the SAPUI5 the progress wasnt good enough. So we set a meeting with Gunnlaugur, the product owner, at Applicon and we came to the conclusion that the AngularJS library would be more appropriate for our website. We all had good experience in Angular so the ball started rolling at that point.

5.5 Sprint 5. Revenge of the Sith. 3/23/2015 - 4/5/2015

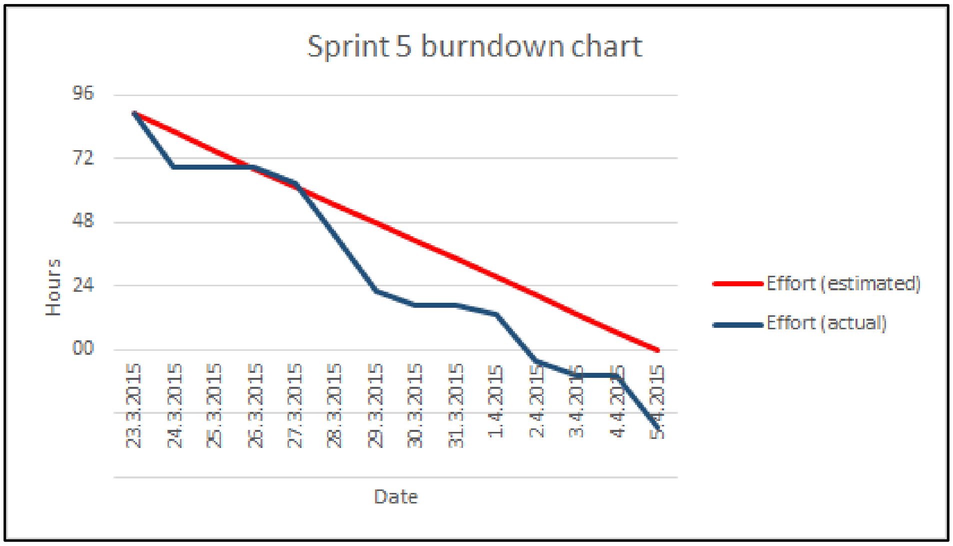


Image 12: Sprint 5 Burndown Chart

Starting the Angular website was the main priority in the sprint and it went better than expected. We managed to get a working site up with the possibility create games and watch the live score of the game. New statistic queries were added and visible on the website. Also we added a xsjs file to add new players which wasn't implemented on the website but tested and it worked with a post call.

5.6 Sprint 6. A New Hope. 4/6/2015 - 4/26/2015

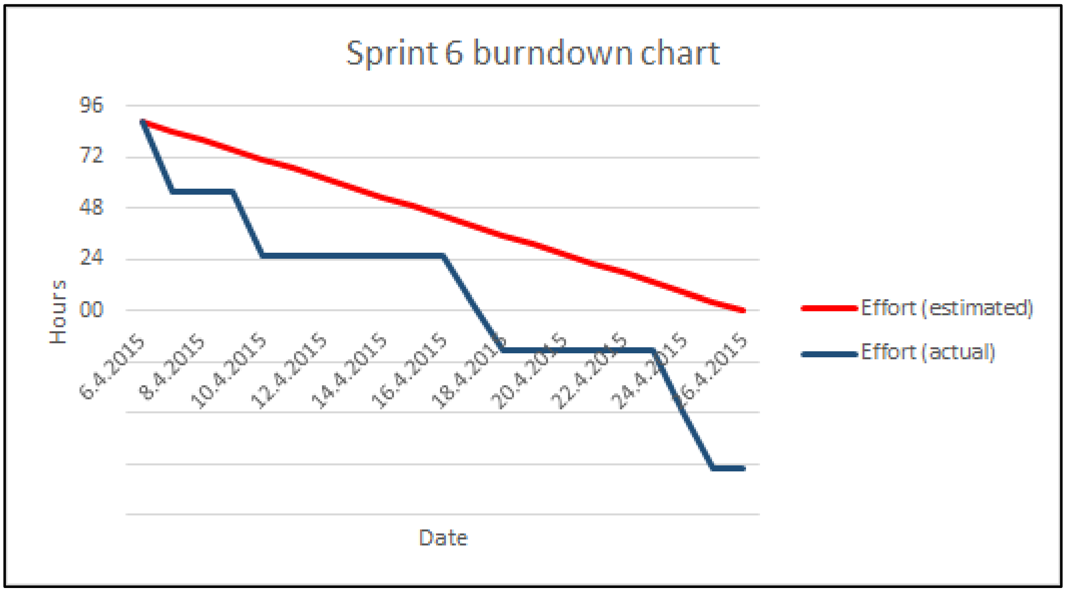


Image 13: Sprint 6 Burndown Chart

After thinking a lot about how we would register goals scored we finally came up with a good solution and made a prototype which worked well. We pursued that idea and made a final version that went into the table shortly after. With working sensors the only thing needed to set up the table was an internet connection so we set up a wifi adapter on the raspberry.

We added ELO ranking system, like is used in chess, to better determine the skill of the players. The system we used was a modified version similar to football team ranking. More statistics was added for overall player/team performance on the table, fastest goal, slowest players etc.

5.7 Sprint 7. The Empire Strikes Back. 4/27/2015 - 5/3/2015

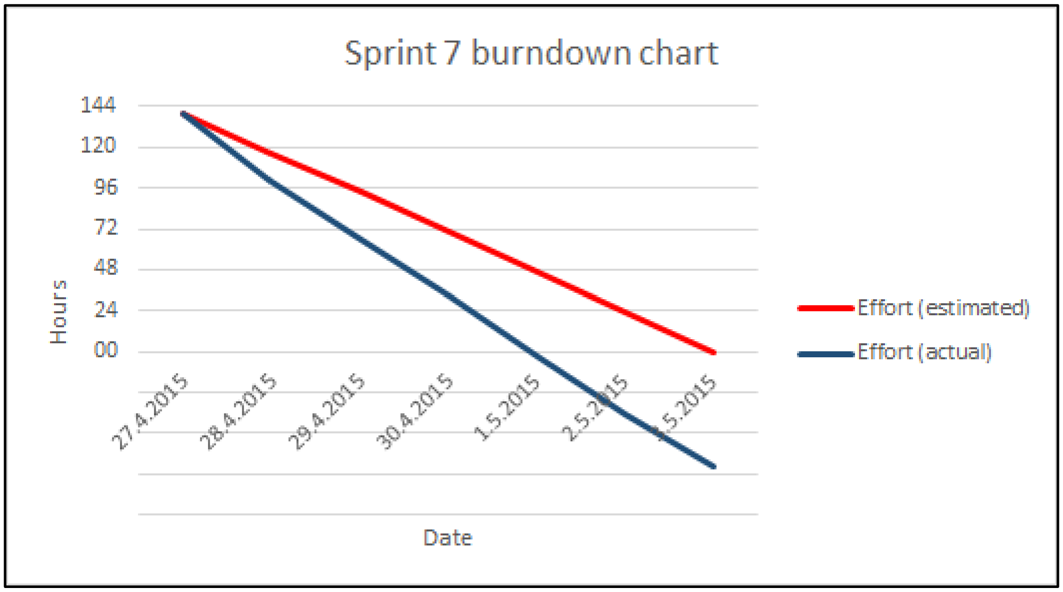


Image 14: Sprint 7 Burndown Chart

The goal was to finish the backend completely, create more functionality on the web site and setting up the raspberry on the table. We aimed at finishing the rest of the SQL queries and be done with all statistics work. With the finals over at school we managed to put all our work into the project which returned great results.

We managed to finish all planned queries and fixing others already implemented that weren't working as expected. With all the statistics available we added them to the table and user views on the website. Views for adding and editing players was also implemented and password made for all users. Overall look for website was greatly improved and made responsive to work on the tablet that was planned to be set up next to the table.

The system was set up on the table and worked as intended. The database was then reset and real data gathered while being tested by employees at Applicon.

5.8 Sprint 8. Return of the Jedi. 5/4/2015 - 5/10/2015

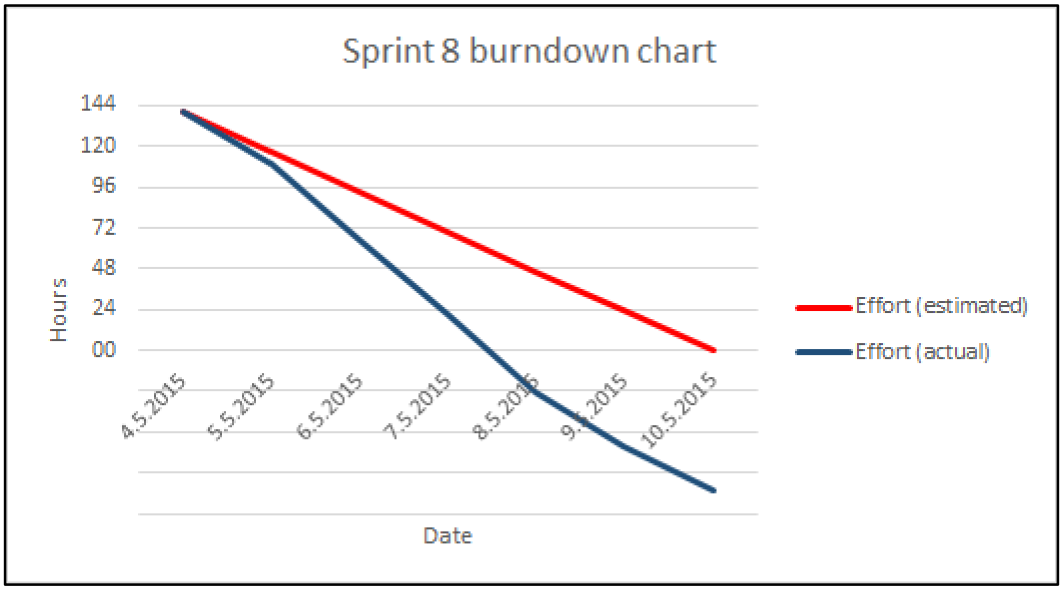


Image 15: Sprint 8 Burndown Chart

In this sprint we continued with the high work capacity and we set the goal to have our system almost finished for the 3rd status meeting. We also aimed to have some real data from played games in the database.

Gathering gamedata went well and the employees enjoyed testing our system. They gave us important feedback which we used in putting the finishing touches on website.

There were problems with wireless adapter that sometime occured during gameplay. The reason why this happened was either the wifi adapters high power consumption or that the Raspberry was randomly moving between access points on the wireless network.

5.9 Sprint 9. The Force Awakens. 5/11/2015 - 5/14/2015

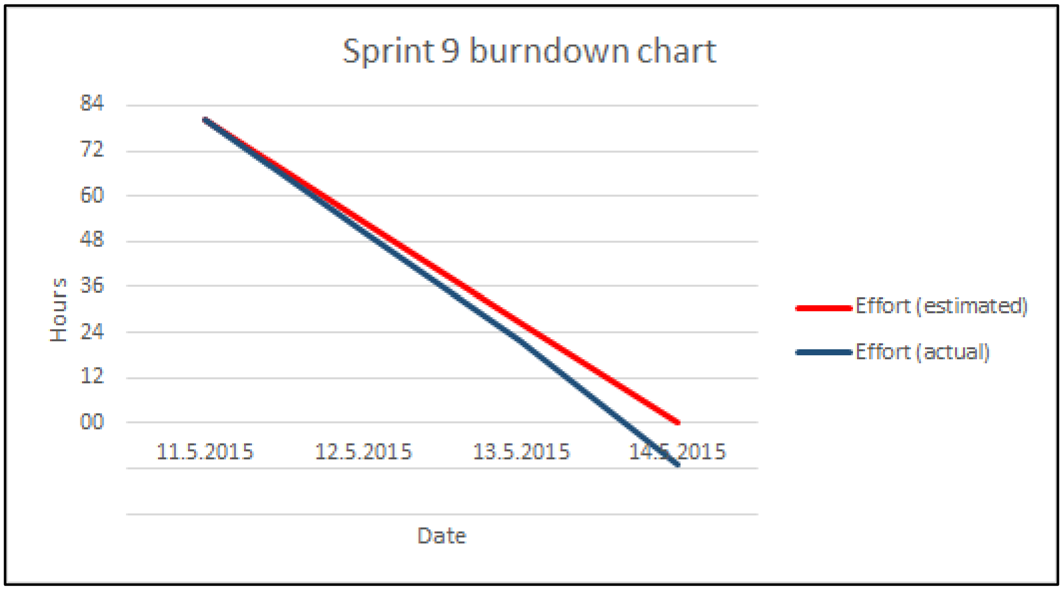


Image 16: Sprint 9 Burndown Chart

This was a 4 day sprint and the goal of this last sprint was to finish the final report and package the system into a deliverable product. A few minor bugs, mainly in the view and its responsiveness, were fixed and after that all the time went into getting the project and report ready to deliver.

6 Applicon remarks

At Applicon foosball is not just a sport, it is a part of our culture, what keeps us experimenting with new ways to outperform the opponent.

So it was not without hesitation that we took on those four undergraduates from RU to work with our HANA in-memory database, our pro-tournament foosball table and sensors to create a product that would take foosball to new heights at Applicon.

I have been so fortunate to closely follow their progress from the beginning. Working with cutting edge technology, newest trends in web design for the user interface complex database queries to create statistics and tying it together with Raspberry PI and sensors to create a fully functional product.

Of course no project is without problems but it is all about how you handle them. They never panicked, sought out assistance when needed and kept calm.

I have to say that the end result blew me away. What they delivered in the end was far beyond my expectations. They came in facing uncertainty, new unknown technology and devices, mastered it and pulled through.

Now we have a product that keeps score, produces amazing statistics and allows us to follow the game-play live. On behalf of all foosball enthusiasts at Applicon, “Thank you for doing this for us”.

Now the only thing that you need to learn is this, to play proper foosball.

Einar Eiríksson
Applicon employee

7 Conclusion

This project was a great learning experience. We got to work with hardware (which is not to common for a final project), a new database and the frontend (website). Despite a slow start we were really happy with the final product. After getting to know the database and switching from SAPUI5 to Angularjs the progress greatly improved.

The goal of the project was to gather game data to determine who was the best player along with some statistics and we believe that we we delivered that nicely. The exact statistics which should be presented wasn’t decided before hand but we implemented the ones we thought they might like, getting a lot of input from employees over the course of the project. More statistics can easily be implemented and we hope they add more if they think of something new.

It was fun working on something the company is so passionate about and seeing how happy they were seeing it in action. We would like to thank Applicon and its employees for the opportunity and help that they provided us with.

Bergur Logi Lúðvíksson
Hafþór Örn Þórisson
Högni Rúnar Ingimarsson
Stefán Arnar Einarsson

Appendix A – Product Baclog

Done:	Story points	Accomplished
Make Raspberry Pi register goals	4	**x**
Create an API that saves the registered goals to the database	4	**x**
Create an API that pushes notifications on goal scores to a frontend	8	**x**
Connect a button to Raspberry Pi that subtracts goals from current score	4	**x**
Add a function in the API that fetches data from the database	2	**x**
Go through SAP:HANA Software Development course on open.sap.com	4	**x**
Create a schema in the SAP:HANA and define access	4	**x**
Give/define access to our schema	4	**x**
Create a SAP xsjs project under our schema	2	**x**
Design the tables	2	**x**
Create columnstore database tables in sap	4	**x**
Create Sequences that increment the id/primary keys for the tables	2	**x**
Create xsjs files for each table	2	**x**
Create sql insert statements in the xsjs files	4	**x**
Create a dummy SAPUI5 frontend that allows you to add a new game to the games table	8	**x**
Create a server on that runs on the raspberry and adds goals to the goals table	8	**x**
Post request to the Raspberryserver that has the new gameId	4	**x**
Creating the needed game logic on the raspberry & in the xsjs files	12	**x**
Starting the frontend development	16	**x**
Create a frontend that displays the score of the current game	16	**x**
Users can register for a game on the frontend	8	**x**
Implement and display list of best teams	4	**x**
Implement and display list of players with best win ratio	4	**x**
Implement and display list of players with the longest winning streak	4	**x**
Implement and display list of fastest goals scored	4	**x**
Implement and display list of players with best scored-goals/conceded-goals ratio	4	**x**

Done:	Story points	Accomplished
Add a function to the API that says wether there is a game currently ongoing	4	**x**
Create database where users can register as a player	4	**x**
Show what players are playing on table	2	**x**
Configure the routing for the website	4	**x**
Connect and make sensors register goals	8	**x**
Place the raspberry and wire and connect the fussballtable	8	**x**
Set up wifi adapter for Raspberry with static IP	8	**x**
Implement ELO system	8	**x**
Create SQL queries for statistics for all players	16	**x**
Create SQL queries for top 10 list in various categories	16	**x**
Create SQL queries for wall of shame	8	**x**
Place a game tablet for the fussballtable	2	**x**
Create a view for wall of shame	4	**x**
Implement Add new player to the system	4	**x**
Implement edit existing user functionality	4	**x**
Created a password for all users needed to edit each profile	4	**x**
Created functionality in the backend for password authentication	8	**x**
Replaced temporary gui with a more refined one on the website	12	**x**
Deploy the website	8	**x**
Display more detailed statistics	12	**x**
Finalize the look of the website (for all views)	12	**x**
Be able to display history of games	8	**x**
See details of every game played	16	
Create SQL query for win ratio and other statistics if your best partner is not in the equation	12	
Add a API function that checks if there is an exciting match currently going on	2	
Add a function to the API that returns the estimated time of the table being free again	8	
Display on the website ETA of the table being free if there is a game going on	8	