First Iteration Report

PROG1004

Group nr: 3

Members:

- Sebastian Instanes Skylstad
- Jan Olav Lyche Aspelund
- Daniel Hinderaker
- Arvid Moemeni
- Andreas Blakli

Date: 06.03.2022

1 Table of Contents

Maximus Cup Arranger Vision

Version 1.0

Revision History

Date	Version	Description	Author
11.03.2022	V1	First iteration of the Vision document.	All team members

Vision

2 Introduction

The purpose of this document is to collect, analyze, and define high-level needs and features of the Maximus Cup Arranger. It focuses on the capabilities needed by the stakeholders and the target users, and why these needs exist. The details of how the Maximus Cup Arranger fulfills these needs are detailed in the use-case and supplementary specifications.

The goal for Maximus Cup Arranger, a football tournament management application, is outlined in this document. The project's purpose is to design and construct an app that organizes cups and teams, allowing end users to easily establish and run their own competitions. We want to create a project that is simple to use while also being comprehensive and effective.

2.1 Purpose and scope

The purpose and goals for our tournament application Maximus Cup Arranger are outlined in this vision document. We state what we aim to accomplish and how we plan to do it in this section.

Our team has focused on developing an easy-to-use application for consumers and small-scale tournaments. The finished result, however, cannot expect a large market share or the high levels of sophistication required to compete in the professional tournament management space due to the relatively inexperienced programming team and a three-month timeframe.

2.2 References

[1] StatCounter (2022, 06 March). *Desktop Windows Version Market Share Worldwide*. Retrieved 06.03.2022 from https://gs.statcounter.com/os-version-market-share/windows/desktop/worldwide/

[2] Cup Manager (2022, 06 March). *Cup Manager*. Retrieved 06.03.2022 from https://cupmanager.net/no/

[3] thekush7 (2022, 06 March). *Football-game.* Retrieved 06.03.2022 from https://github.com/thekush7/Football-game-Football-club-management-system

[4] Seyed Ali Amirsahi (2022, 06 March). *Blackboard.* Retrieved 06.03.2022 from https://ntnu.blackboard.com/ultra/courses/_30664_1/cl/outline

2.3 Overview

The rest of the vision document outlines our project's objectives and how we plan to achieve them. It also includes a risk analysis and cost estimates. The vision document describes the end product we want to achieve, as well as the various requirements set.

3 Positioning

3.1 Business Opportunity

The current offerings by other Football tournament applications are aimed at professional teams with a high budget. After being contacted to develop a tournament application**,** we have set on a mission to create a full-featured, easy to use application that is within range for hobbyist football teams and managers wishing to automate the matchmaking experience. We believe that our focus on this user-group can make it a successful contender among other matchmaking application offerings.

3.2 Problem Statement

Currently, there are not a lot of budget options available for hobbyists and volunteer-based tournament applications. This makes the process of matchmaking cumbersome and time-consuming for the end user. Which as a result makes creating tournaments and contests across the volunteer-driven space much harder, resulting in fewer competitions and tournaments. A solution to this problem is an application that is both easy to use and affordable, which is what Maximus Cup Arranger aims to do.

3.3 Product Position Statement

This product is made for a corporation who had a demand for tournament-based matchmaking but can be of use to any individuals wishing to create tournaments. Today, this applies to many volunteer-based clubs that wish to make the matchmaking experience less difficult. Unlike Cup Manager, our primary competitor, we aim to create a less expensive, more privacy-perceiving solution. Our application stores all data locally and does not collect any telemetry.

4 Project goals

4.1 Impact goals

- Optimize our client’s workflow and work process.
- Reduce resources and time spent on tournament administration.
- Make tournament arrangement as simple as possible by automating the process.

4.2 Result goals

- Create a functional football tournament application, which simplifies tournament arrangement.
- Make sure that our application is user-friendly and effective.
- Finish all documentation and the software in time.
- Weekly meetings with teacher assistant and group.
- Each team member should work 110 hours to achieve this.

4.3 Process goals

- Increase understanding of software engineering methodology.
- Get to know each other within the group
- Improve our ability to work in teams and learn about different team roles.
- Improve communication skills.
- Learn about new technologies and tools like Git, and different C++ libraries.
- Get a good grade in PROG1004.

5 Stakeholder and User Descriptions

5.1 Market Demographics

Maximus Cup Arranger aims to reach a wide audience for our product by implementing an easy-to-use user-interface with well-known, trusted technologies. We will focus on primarily making the application available on Microsoft Windows, as Windows 10 currently has a market share of approximately 75% *(StatCounter, 2022)*, making it the most popular platform. Our target audience is small volunteer-based football teams looking for a simple, powerful and privacy respecting tournament matchmaking application. As our organization does not have experience or reputation in these markets, the goal is to focus on a slow and steady growth, earning customer trust along the way, further expanding our userbase. Having a stable and solid program will help us reach these goals.

5.2 Stakeholder Summary

Name	Description	Responsibilities
Developer	The developer oversees development, maintainability, and documentation of the application.	- Ensures that the system will be maintainable - Ensures that there will be a market demand for the product's features - Development of the product - Meet the user requirements
Client	This is the person or organization that has commissioned this project.	- Paying the developers - Setting requirements for the product - Approves funding - Monitors the project's progress

5.3 User Summary

Name	Description	Responsibilities	Stakeholder
User	End user. The users that will be using the application.	- Getting familiar with the application, to be able to utilize it an effective manner. - Provide feedback for future development and improvements. - Use the platform on a regular basis so it becomes a part of an overall effective workflow. - Maintaining appropriate security measures so no sensitive data is lost or compromised.	Not applicable

5.4 User Environment

The application will be developed by a team of five members for the entirety of this project. Each task cycle is around one week, depending on the amount of required for each activity. E.g., more time is required for implementing a feature than writing the manual. Goals will be set at the start of each task cycle, meaning the amount of time spent on each task may change. In total, the team members aim to spend around 110 hours each on the project.

The program is developed as a simple terminal-based application in C++, making it not reliant on internet-connectivity to function. This would mean that supported platforms would not be subject to environmental constraints. Currently, the plan is to focus development on Microsoft Windows, as that is the most popular widely used platform.

5.5 Key Stakeholder or User Needs

Need	**Priority**	**Concerns**	**Current Solution**	**Proposed Solutions**
Tournament format feature	High	Could be hard to implement.	N/A	TBD
Update Tournament	High	No concerns	N/A	TBD
Adding multiple teams	High	No concerns	N/A	TBD
Edit or delete teams	High	No concerns	N/A	TBD
View teams and tournament	High	No concerns	N/A	TBD
Persistent storage	High	User-friendliness	N/A	Database

5.6 Alternatives and Competition

Identify alternatives the stakeholder perceives as available. These can include buying a competitor’s product, building a homegrown solution, or simply maintaining the status quo. List any known competitive choices that exist or may become available. Include the major strengths and weaknesses of each competitor as perceived by the stakeholder or end user.

App	Cup Manager	Football-Game
Pros	1) Integrates with payment solutions 2) Advanced features such as visualizations, website-generator 3) Available in browser or as mobile app	1) Open source. 2) Privacy respecting
Cons	1) Pricy, subscription-based service. 2) Privacy-concerns	1) Not compiled. 2) Hard to use. 3) Missing user guide

6 Product Overview

The product will consist of a program able to run any small/ medium sized soccer tournaments. The programs capabilities are:

- Make tournament brackets following soccer cup rules (single elimination)
- Add teams and players
- Removing teams and players
- Store data on teams and players in the tournament
- Update stored data

6.1 Product Perspective

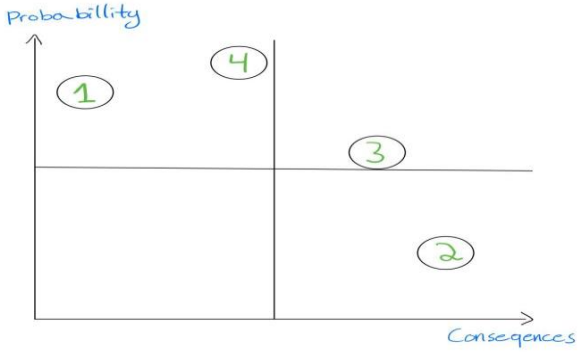
There are a lot of variants to our product already on the market. Many of these are online and free to use. One of these is “Tournament bracket generator” which allows anyone to make a single elimination tournament bracket for most games. Most of these generators are easy to use and flexible, meaning that its fast to remove and start over. The difference, however, is that in addition to be able to output a bracket for the tournament our program will store teams and player data locally and allow for updates. This program is self-contained and is not connected to other larger systems.

6.2 Summary of Capabilities

The program is capable of creating multiple tournaments, editing and storing data locally. Each player on a team will also be saved and the user is able to edit these. By being able to store teams on the program, the user will be able to add them to new tournaments, thus automating a tedious process of adding players and teams each time a new tournament is made.

6.3 Risk analysis

1. Project slowed down as a result of team members getting sick.
2. Corona situation might develop and result in university closing down.
3. Uncertainty of risk connected to project team's schedule estimate compared to the scope of the project.
4. Unforeseen problems with the code.



Risk factor description:

1. Probability is high for one of the team members to get sick while we're working on this project. The consequences can vary depending on the severity of the situation, but we will still add this in the low-risk group. This is because we as a group can take on the extra workload while our team member recuperates.
2. Probability seems to be low for the moment, but the situation is always changing. The reason the consequences are so high is because it will affect all steps in this project. Team meetings, user tests and meetings with client will all be affected. This will in our opinion highly affect our rate of progress. If such a situation occurs, we'll handle with it by working online which is a decent substitute at best.

3. As a team we have a rough idea of how much work it will take to make our product. Since our group has never worked together and not developed a product on this scale before, there might be more work than we realize. If this is the case, we would consider limiting the scope of our project to have enough time to develop and deliver a product within the final deadline. This is something we are closely following up each meeting to cultivate a better understanding of the work needed to finish the project on time.
4. This is our first time making a program from scratch and we are not expecting everything to work on the first try. Unforeseen problems will most likely occur, and we will deal with them accordingly. Most of the usual problems with coding we are experienced with are on the low end of the consequences scale, however most of us have never made a whole program without templates or guidance. This could lead to progression loss on the project trying to figure out how to solve new coding issues.

6.4 Cost, Pricing, and benefits

Cost and Pricing	Number of people	Hourly Salary	Hours per person	Total salary NOK
Planning	5	1000	35	175000
Administration	5	1500	35	262500
Programming	5	1200	70	420000
User test	5	0	2	0
Total Cost				857500

6.5 Quantifiable and non-quantifiable benefits

Quantifiable benefits:

- The estimated NOK value of potential new customers
- The estimated NOK value of potential investors
- New customers

Non-quantifiable benefits

- The value of a better reputation in the market
- Increased teamwork
- Improved communication

6.6 Estimated costs

The project team consists of five members with an estimate of 110 hours of work per worker. To finish this project the cost will reach 857.500 NOK if everything goes as planned.

6.7 Licensing and Installation

The program will be a .exe file that can be downloaded and run locally. No preinstalled programs or specific requirements are needed for the user to be able to run the program on a windows operating system.

7 Product Features

Tournament Feature

- Purpose: Setup a tournament and generate matches manually or randomly.

Update Feature

- Purpose: Update ongoing tournament, which team won/lost.

Add Feature

- Purpose: Add and set name for a team.

Edit or delete Feature

- Purpose: Edit team name or delete team

View Feature

- Purpose: View all teams added, and tournament statistics (Wins/Losses).

Database Feature

- Purpose: Import team from file, write finished tournament to file.

8 Constraints

- Since most of our group members have minimal experience with GUI design, we will make the software command line (CLI) based.
- Our software will be shipped as a compiled executable file, not with an installation manager. This reduces our ability to send software updates and new features to our client. Our client will be notified when new updates/features are available, and then they must manually download the updated version.

9 Quality Ranges

As the program will be small and client based, performance will not be an issue. It should reach a point of robustness where it will not crash randomly or from typing wrong commands and handle at the very least 10 teams and tournaments at once. If feasible, it should also be usable enough to be used in a real tournament without any problems happening.

10 Precedence and Priority

Our feature priorities are listed as such:

1. Tournament format feature
2. Adding multiple teams
3. View tournament
4. Update tournament
5. View team
6. Edit or delete teams
7. Database

11 Other Product Requirements

11.1 Performance and System Requirements

Performance and system requirements are minimal. Any computer running windows 8 and above will be able to run our software.

12 Documentation Requirements

User manual, Online help and installation guide will be available when we have the final product.

13 A Feature Attributes

13.1 A.1 Status

Some of the features implemented in the prototype are not yet as we envisioned.

Proposed						Database
Approved	Tournament					
Implemented		Update	Add	Edit/Delete	View	

13.2 A.2 Effort

Most of our features are simple. Some will still require more time and resources, but not by a large margin. Everyone in the group will contribute code, some more than others, and some will focus more on documentation.

Tournament format feature

- The effort needed to code this feature, as we envisioned it, is higher than most of our features. The reason being we need to learn a new C++ library called “random”. This library is needed for random generation of numbers, which will come in handy for random generation of matches between different teams. This is also the core feature of the software and relies on the Adding teams feature.

Update tournament

- This feature complexity is low. It is dependent on other features in our program, but it is needed for our software to work as intended.

Adding multiple teams feature

- All features in our program rely on this feature, multiple teams are needed to arrange a tournament. It is still a simple feature which should not require too much effort.

Edit or delete team(s) feature

- This will also be one of the features that require more effort than others. It requires mechanisms for when you are allowed to edit or delete a team.

View team and tournament feature

- The complexity of this feature is quite low. It is only supposed to show the different ongoing tournaments and the teams.

Database feature

- This feature, as envisioned, should be simple. It does not rely on other features.

13.3 A.3 Stability

Most of our software features rely on other features, but not in the sense that they are not changeable. We planned our software features to be flexible in case we need to make changes, without changing the entire software.

13.4 A.4 Target Release

Our prototype will contain the core features of our envisioned software. We will focus on the core components and features of the software before we add the extras. Our software will be fully complete and operational before 29.04.2022

13.5 A.5 Assigned To

All feature development, programming tasks and developer roles will be delegated within the team.

13.6 A.6 Reason

The requirements for our project can be found on Blackboard (Project Description). These requirements were set by our professor, Seyed Ali Amirshahi, in PROG1004 - Software Engineering. Our client might request additional requirements for our software that we will implement.

14 Prototypes

Note: We created our first paper prototype with a **GUI** design. Later that evening our TA told us we needed to create a user interface for CLI. We invested a good amount of time on the paper GUI design, so it is added here as documentation as it is completed work. We also based our CLI user interface on the GUI paper version. See the appendix 17.2 Paper GUI prototype Figure 16, Figure 17, Figure 18, Figure 19, Figure 20, Figure 21, Figure 22, Figure 23 and appendix 17.3 Paper prototype CLI.

Second note: Our first designs in paper GUI and paper CLI prototype were written in Norwegian. The final program prototype is written in English. The final program prototype was the version we did the user testing on.

14.1 CLI football tournament application first rapid prototype:

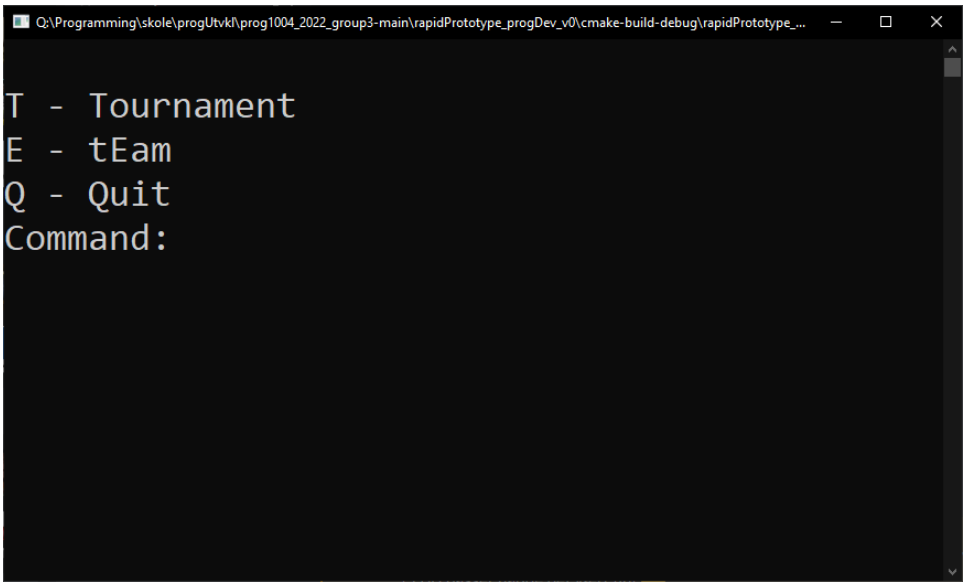


Figure 1

Figure 2

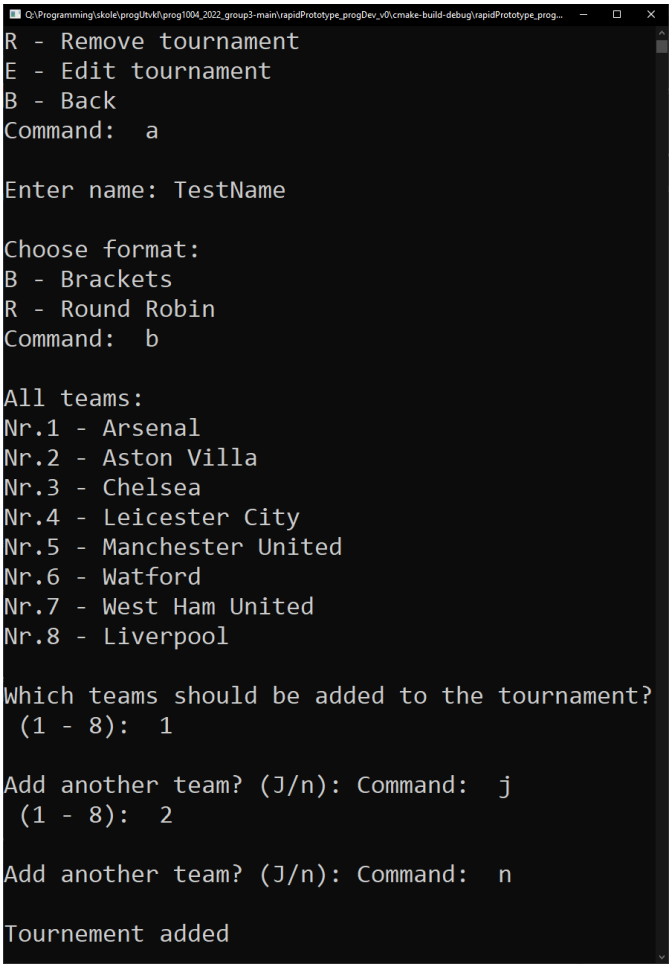

Q:\Programming\skole\progUtvkl\prog1004_2022_group3-main\rapidPrototype_progDev_v0\cmake-build-debug\rapidPrototype_prog...
R - Remove tournament
E - Edit tournament
B - Back
Command: a
Enter name: TestName
Choose format:
B - Brackets
R - Round Robin
Command: b
All teams:
Nr.1 - Arsenal
Nr.2 - Aston Villa
Nr.3 - Chelsea
Nr.4 - Leicester City
Nr.5 - Manchester United
Nr.6 - Watford
Nr.7 - West Ham United
Nr.8 - Liverpool
Which teams should be added to the tournament?
(1 - 8): 1
Add another team? (J/n): Command: j
(1 - 8): 2
Add another team? (J/n): Command: n
Tournement added

Figure 3

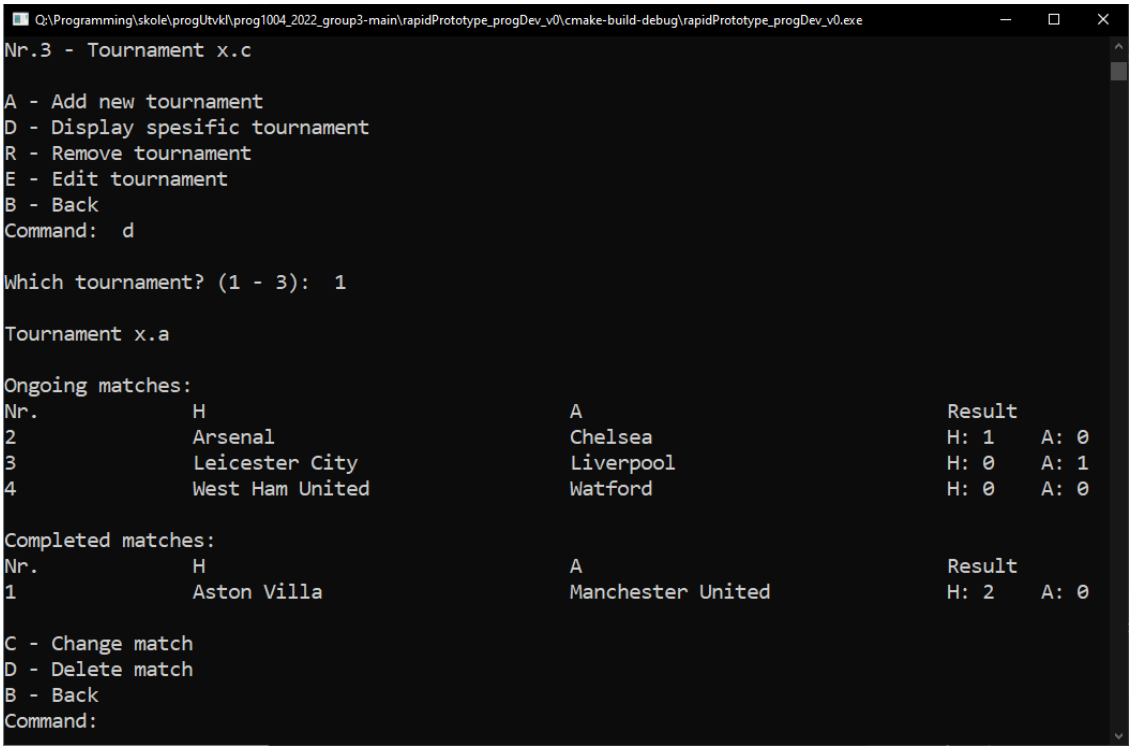

Q:\Programming\skole\progUtvkl\prog1004_2022_group3-main\rapidPrototype_progDev_v0\cmake-build-debug\rapidPrototype_progDev_v0.exe
Nr.3 - Tournament x.c
A - Add new tournament
D - Display spesific tournament
R - Remove tournament
E - Edit tournament
B - Back
Command: d
Which tournament? (1 - 3): 1
Tournament x.a
Ongoing matches:
Nr. H A Result
2 Arsenal Chelsea H: 1 A: 0
3 Leicester City Liverpool H: 0 A: 1
4 West Ham United Watford H: 0 A: 0
Completed matches:
Nr. H A Result
1 Aston Villa Manchester United H: 2 A: 0
C - Change match
D - Delete match
B - Back
Command:

Figure 4

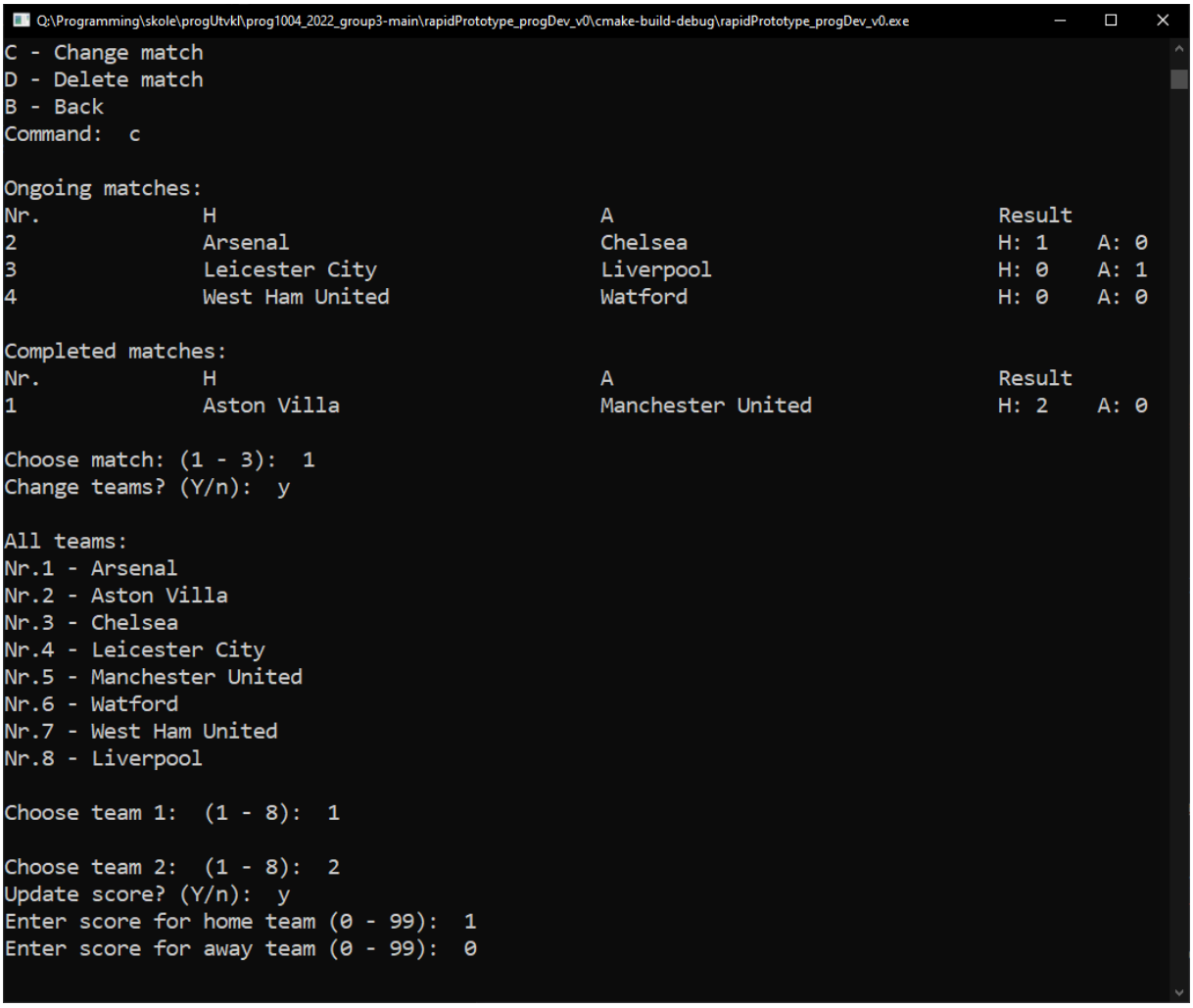

Q:\Programming\skole\progUtvkl\prog1004_2022_group3-main\rapidPrototype_progDev_v0\cmake-build-debug\rapidPrototype_progDev_v0.exe
C - Change match
D - Delete match
B - Back
Command: c
Ongoing matches:
Nr. H A Result
2 Arsenal Chelsea H: 1 A: 0
3 Leicester City Liverpool H: 0 A: 1
4 West Ham United Watford H: 0 A: 0
Completed matches:
Nr. H A Result
1 Aston Villa Manchester United H: 2 A: 0
Choose match: (1 - 3): 1
Change teams? (Y/n): y
All teams:
Nr.1 - Arsenal
Nr.2 - Aston Villa
Nr.3 - Chelsea
Nr.4 - Leicester City
Nr.5 - Manchester United
Nr.6 - Watford
Nr.7 - West Ham United
Nr.8 - Liverpool
Choose team 1: (1 - 8): 1
Choose team 2: (1 - 8): 2
Update score? (Y/n): y
Enter score for home team (0 - 99): 1
Enter score for away team (0 - 99): 0

Figure 5

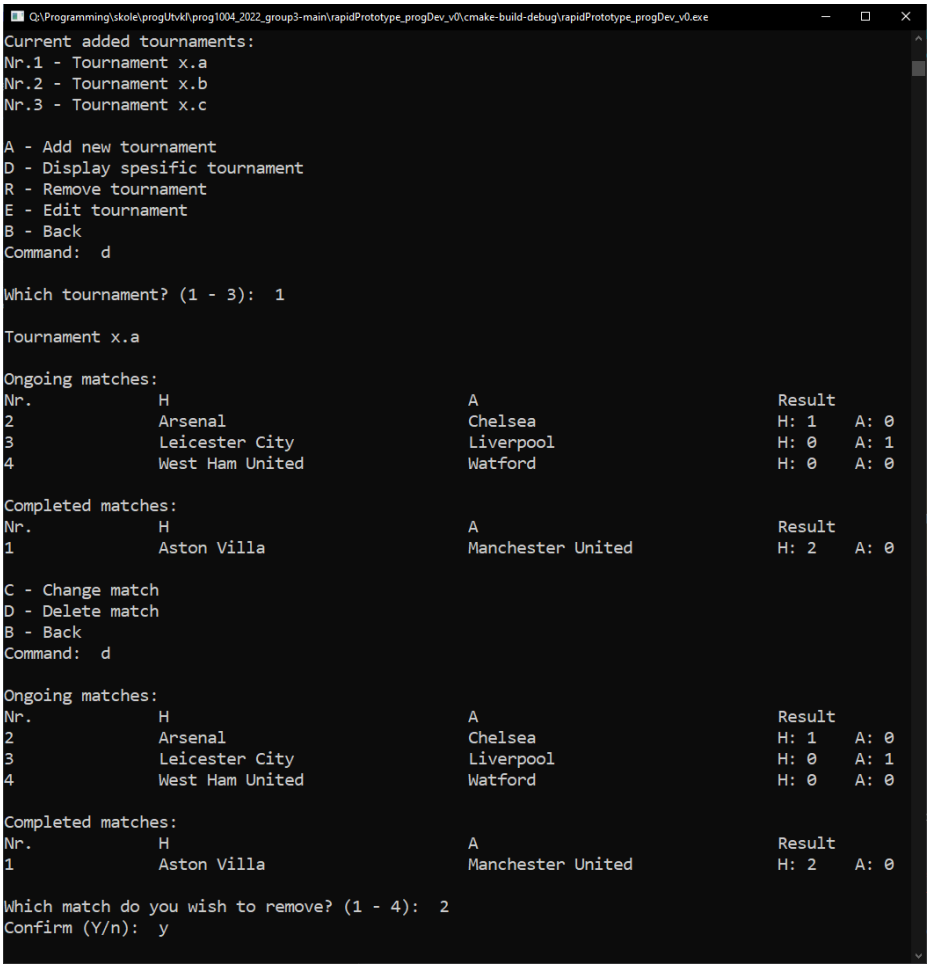


Figure 6

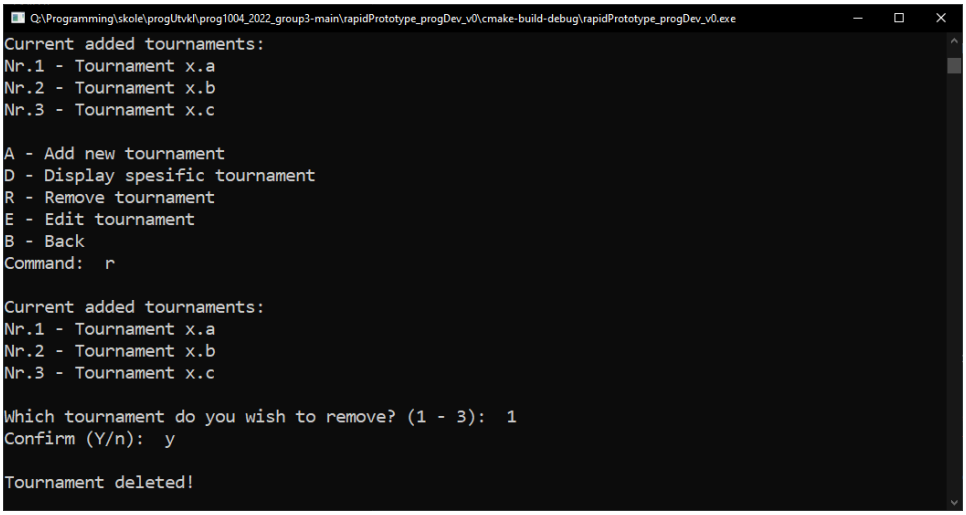


Figure 7

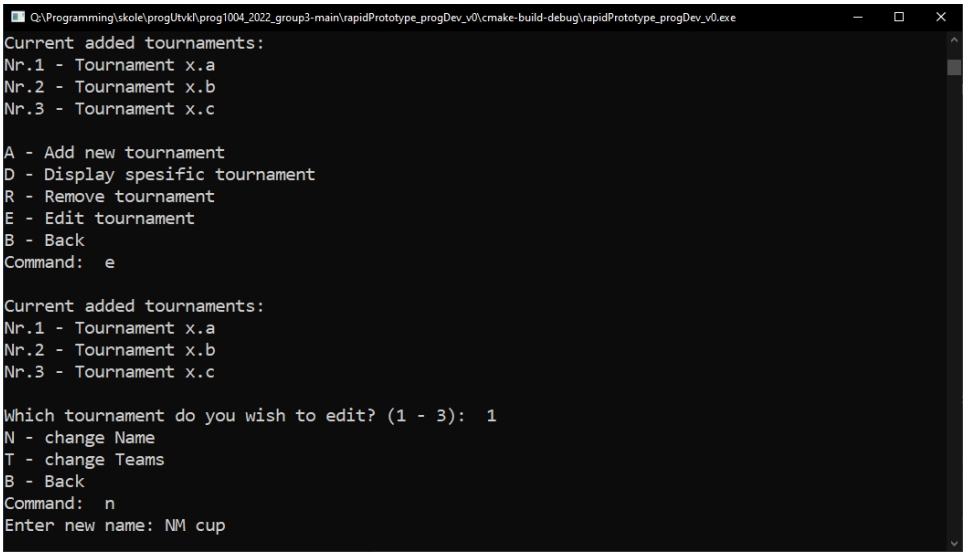


Figure 8

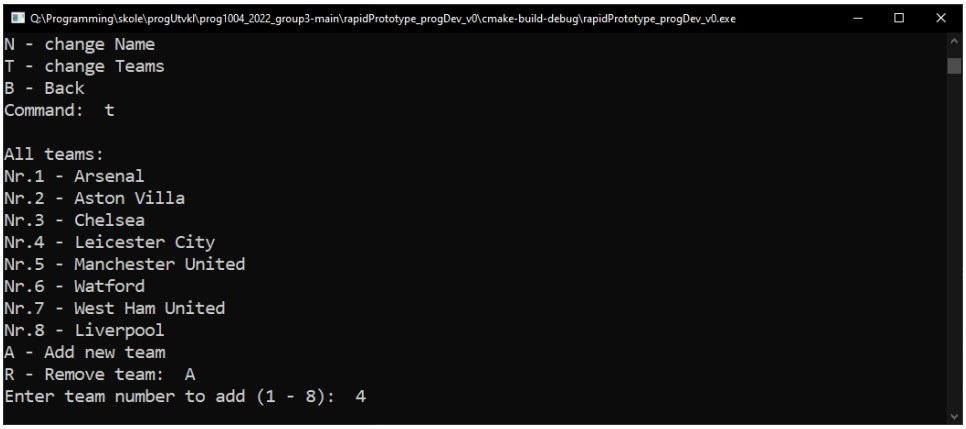


Figure 9

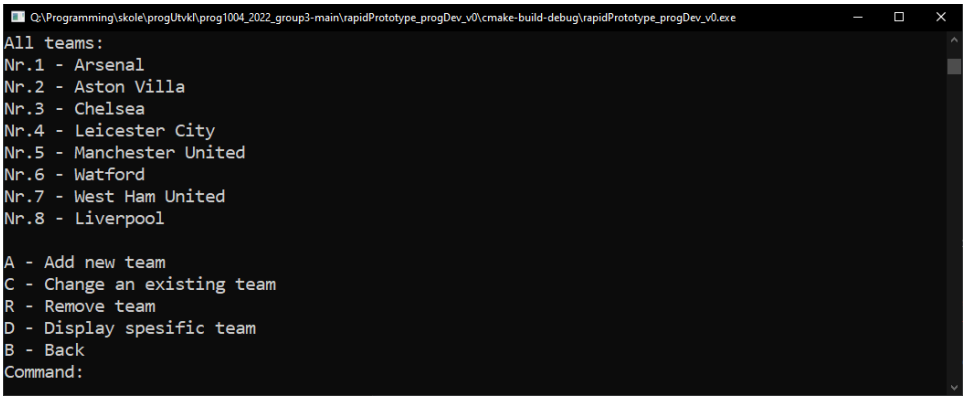


Figure 10

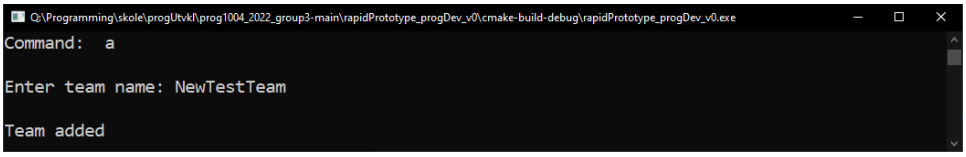


Figure 11

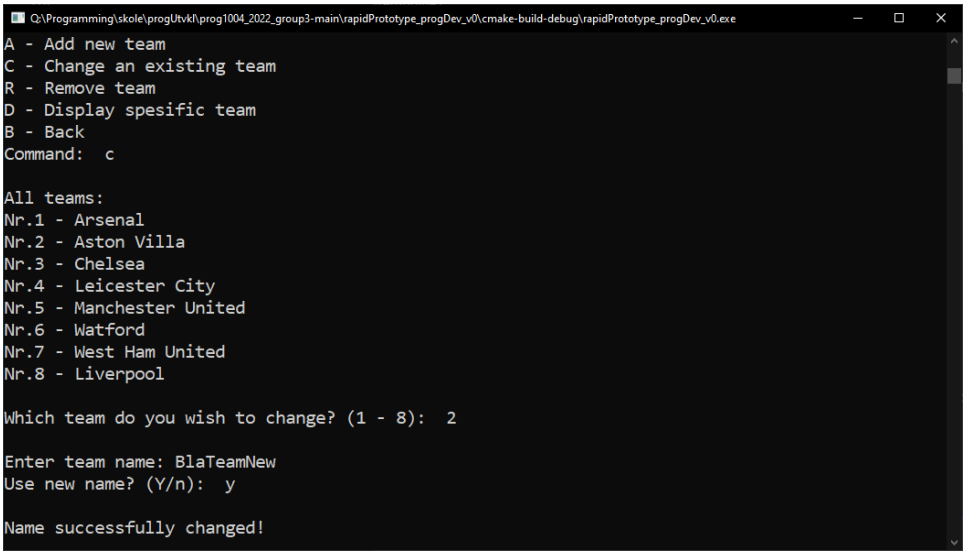


Figure 12

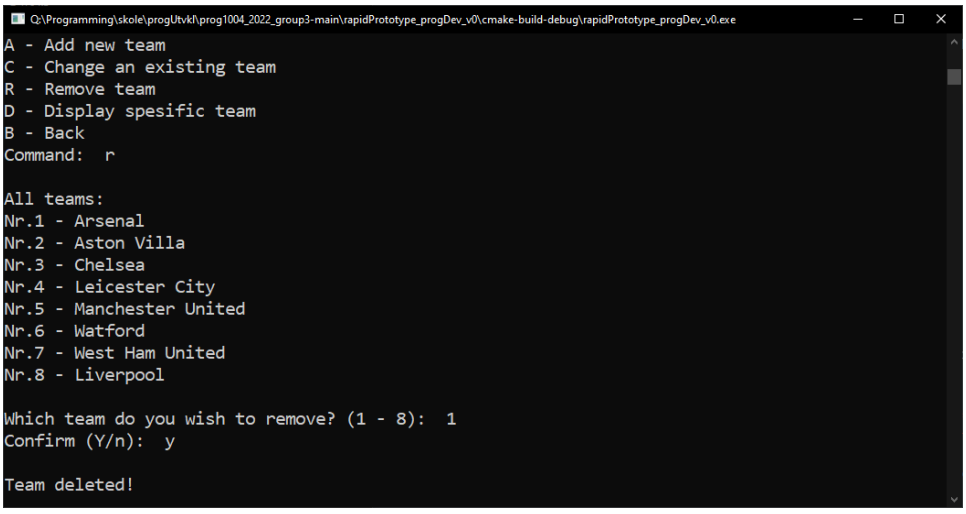


Figure 13

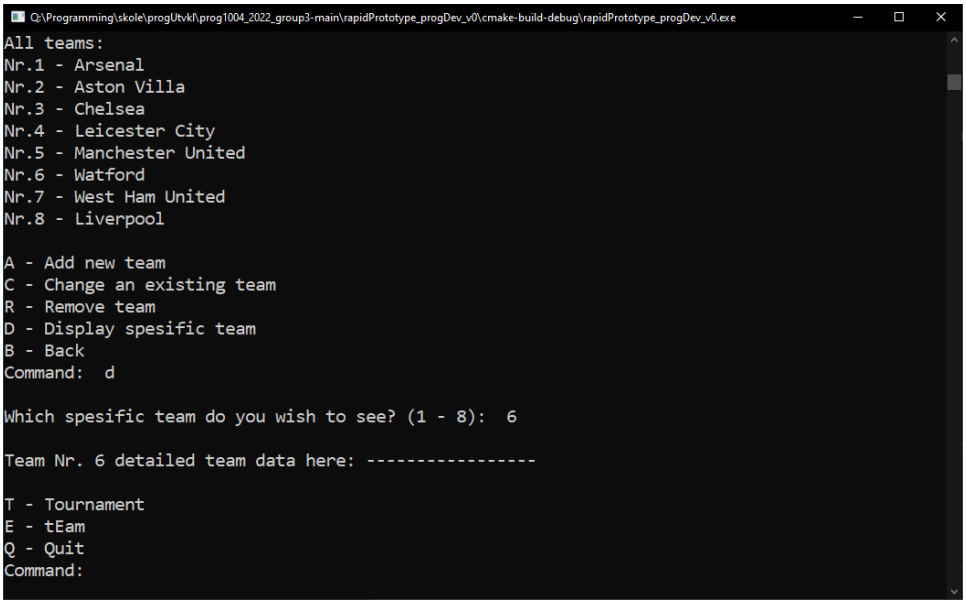

C:\Programming\skole\progUtvkl\prog1004_2022_group3-main\rapidPrototype_progDev_v0\cmake-build-debug\rapidPrototype_progDev_v0.exe
All teams:
Nr.1 - Arsenal
Nr.2 - Aston Villa
Nr.3 - Chelsea
Nr.4 - Leicester City
Nr.5 - Manchester United
Nr.6 - Watford
Nr.7 - West Ham United
Nr.8 - Liverpool
A - Add new team
C - Change an existing team
R - Remove team
D - Display spesific team
B - Back
Command: d
Which spesific team do you wish to see? (1 - 8): 6
Team Nr. 6 detailed team data here: ----------
T - Tournament
E - tEam
Q - Quit
Command:

Figure 14

15 First User Test Report

15.1 Introduction

Maximus Cup Arranger aims to make the process around adding, configuring and updating football cup tournaments easy. It serves as a football cup tournament manager in which a user can easily configure their cup tournaments.

15.1.1 Summary

The main takeaway from the user test were that the program for the most part was intuitive to use, but it lacked clarity in some areas which made the user's expectations differ from what a program command did and what happened e.g., when the commands 'T' then 'A' was used to get to add teams in a tournament, and the page with ongoing and completed matches were unclear.

15.1.2 People Involved

There was a total of four participants:

1. An interviewer.
2. A person who took notes.
3. An observer.
4. The user.

15.1.3 Demographic

The sample size of the user test is limited to one user due to time constraints and difficulties in finding multiple participants which had high amount of experience in the same tournament systems. The age group of the demographic is late 50's with a decent amount of experience in computer and application use. Our expectations were that the user would most likely find some functionality unintuitive which we as system designers might consider "self-explanatory".

15.2 Tasks

The user had to complete predetermined sequences of interactions with the program in the form of commands, numerical and string inputs to explore the application. This ensured that as many aspects as possible would be explored. See 17.1.1 Interview questions and steps to go through with user for the first user testing session 02.03.2022. in the appendix.

15.3 Results

Most features are good, and the program is for the most part intuitive to use. Some aspects the user displayed their concerns about were:

- How the adding teams to a tournament were unclear and confusing
- The ongoing and completed match log provided poor readability.
- Confusion regarding tournament types.
- When adding team or tournament the user expected their respective results to be printed automatically.

See 17.1.2 Minutes 02.03.22 User test 1 in the appendix

15.4 Conclusion

The feedback received from the user was valuable. It shed light on flaws the program had and areas which needed improvement. Some of feedback was on functionality we knew beforehand were not optimal solutions, but most importantly the user mentioned some areas of the program which we had not considered. This test provided us with useful data which we can use to improve the next application iteration.

16 Domain Model

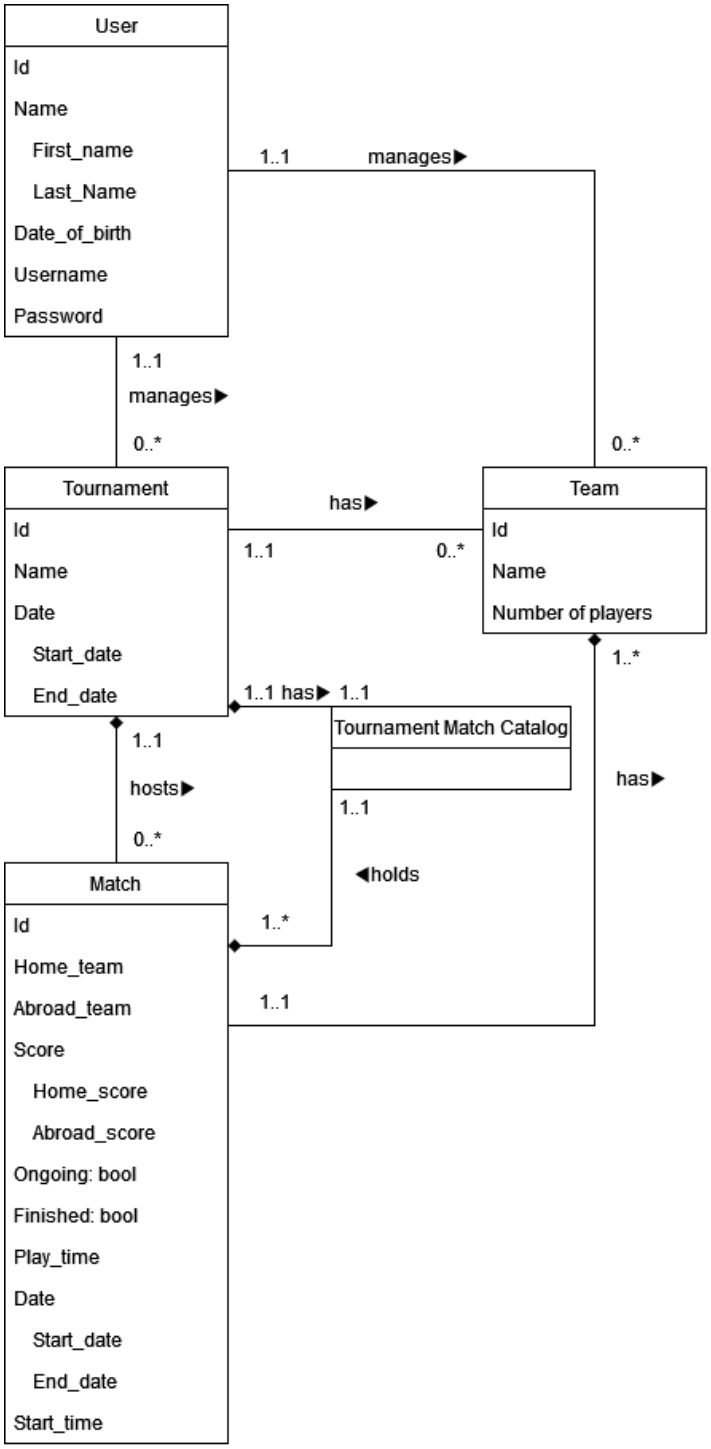


Figure 15

17 Appendix

17.1 User Testing Documentation

17.1.1 Interview questions and steps to go through with user for the first user testing session 02.03.2022.

General guidelines to follow when testing with a user:

- Let the user try to figure it out for themselves, only guide/intervene the user if stuck.
- Do not interrupt the user.
- Make the user comfortable.
- Do not stress the user.
- Try not to influence the user, i.e., limit our influence over the user.
- Encourage feedback, even if the received feedback is irrelevant or useless.
- Stay on track with the questions and the outlaid testing steps below.

Steps to go through with user:

1. What would you do to add a tournament?
2. What would you do to see all the current tournaments?
3. What would you do to remove a tournament?
4. ---------- edit a tournament?
5. ---------- change the name of a tournament?
6. ---------- withdraw or add a team to an ongoing tournament?
7. ---------- display the current matches in a tournament?
8. Could you try to display all the football teams?
9. ---------- add a new team?
10. ---------- change an existing team?
11. ---------- remove a team?
12. ---------- display extended information about a specific team?
13. ---------- go back to main menu?
14. Could you please update the score of an ongoing match?

15. ---------- remove a match from an ongoing tournament?

16. ---------- quit the program?

Questions:

1. Does the user interface feel intuitive?

2. Do the selected operations do what the user expected?

3. How was the feedback from the program after doing an operation? (e.g. change existing team, were the user confused if an operation had been fully/successfully completed after entering a command?)

4. Does the user have any suggestions for improvements that could be made to the interface?

5. Was there anything that the program did well?

6. Does the user have any other personal comments? (Anything goes).

17.1.2 Minutes 02.03.22 User test 1

User test 1

For this first user test, we have let the user test our first iteration of the program. Along the way Arvid was writing down the first impressions and comments given by the user. Andreas has the role of observer, making sure that we ask the user all the questions we have agreed upon beforehand. Jan Olav has the main responsibility of interacting with the user, asking questions, and making sure to show every aspect of the program.

Medium used: Discord with screensharing to show the program to the user.

Key takeaways

- User not familiar with keywords used in the program like round robin and brackets.
    - Addressing this next meeting.
- Confusion related to what information is presented to the user on the program.
    - Addressing this by presenting more intuitive information to the user.
- User expected mostly happy with the program, but had some expectations that weren't met
    - User expected tournament information to be shown automatically when he chose to look at a tournament or player group.
    - Would like to be notified if a tournament is already stored on the program.
    - User thinks the program should have a way to show the entire tournament history. The development from game one to the finals.
    - User wants there to be more clear information about restrictions when he gets the option to choose values in the program.

Summary

Mostly good feedback from the user, but there was a lot of good feedback we can work on before showcasing the next iteration of the program. When the first iteration of the program was made, the group focused heavily on making the presentation of the different functions the program offered. The functions themselves aren't written yet, but we wanted the user to get an insight to what the program offers. Before the next user test we are focusing on building a more robust program, but also use the feedback to improve already existing code.

List of questions after user tests program

Follow up questions:

1. Does the user interface feel intuitive?

- Yes, for the most part I think, but I got confused about the teams, some of the presentation in "pågående kamper" seems odd that there already are stored tournament.
    - To show the user how stored tournaments were showcased in the program we made some examples, but this confused the user.
        - For next time we will remove these examples and let the program show tournaments that the user makes in the user test.

2. Do the selected operations do what the user expected?

- Yes, but I'm confused on what cup and round robin rules mean when I'm given the option to create a new tournament.
    - When the user makes a tournament, he will get the option to make one of two types of tournament brackets. These seems to be confusing and is something we will address in the next meeting.
    - Suggestion from the user: Leagues and cup rules
        - Soccer thermology

3. How was the feedback from the program after doing an operation? (e.g. change existing team, were the user confused if an operation had been fully/successfully completed after entering a command?)

- When I added a team or tournaments, I expected their results would be printed automatically, at least the title of the tournament or something.

17.2 Paper GUI prototype

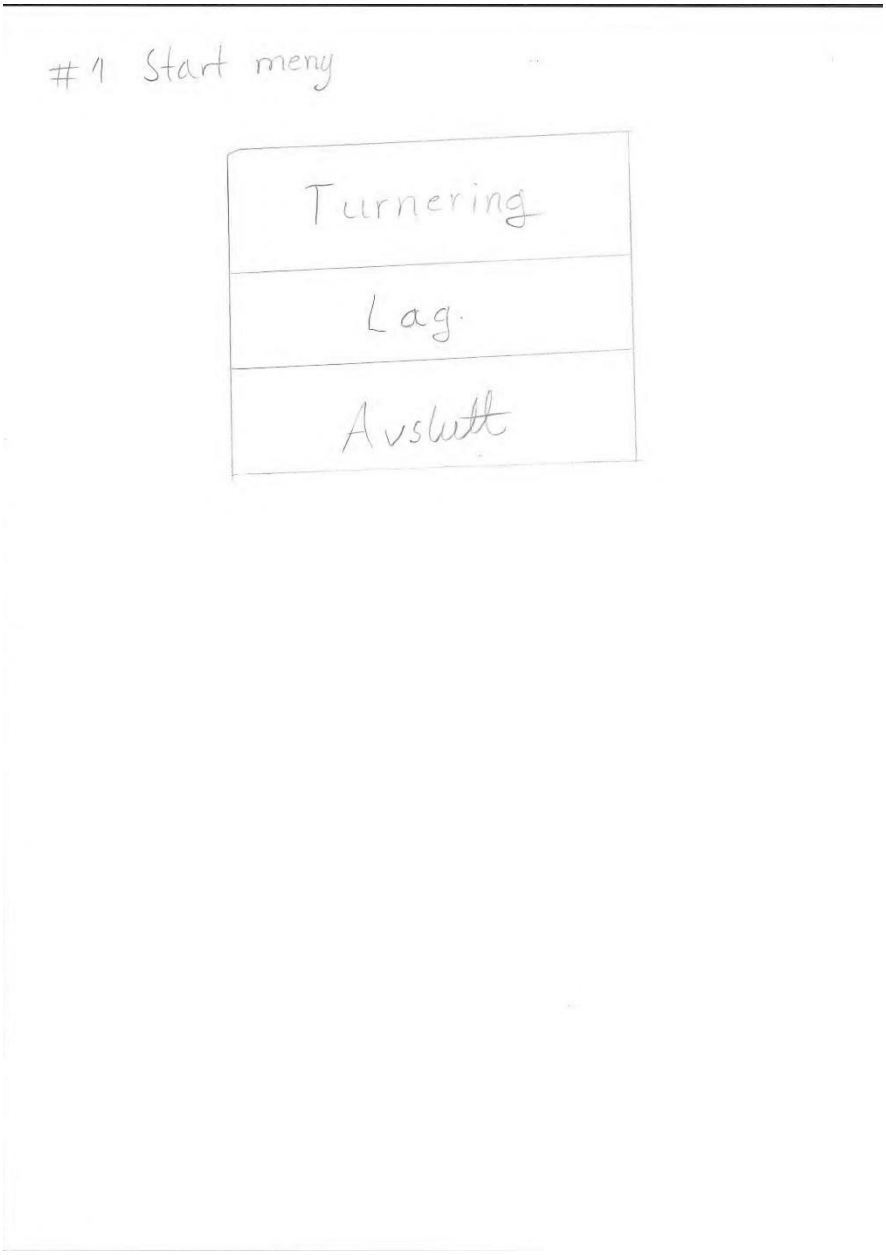


Figure 16

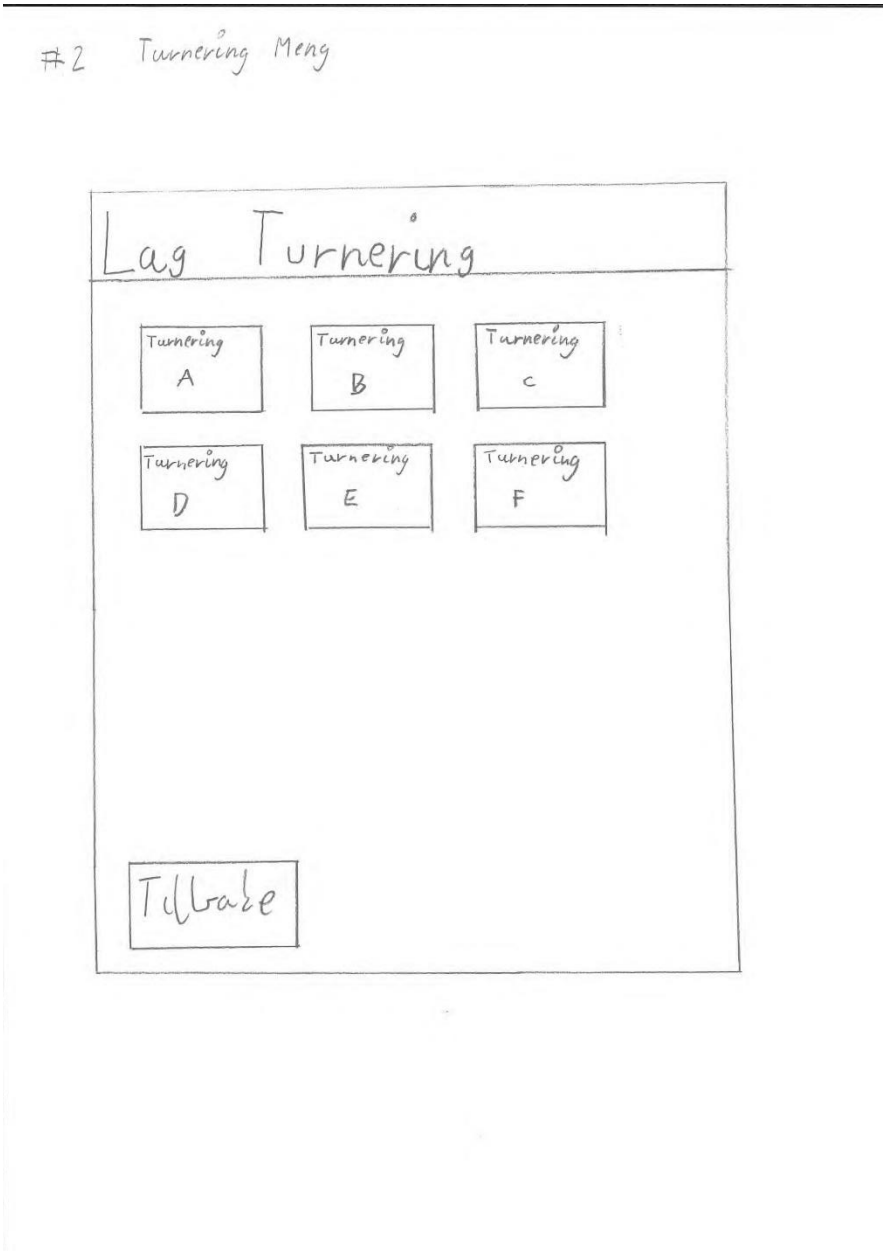

#2 Turnering Meny
Lag Turnering
Turnering A
Turnering B
Turnering C
Turnering D
Turnering E
Turnering F
Tilbake

Figure 17

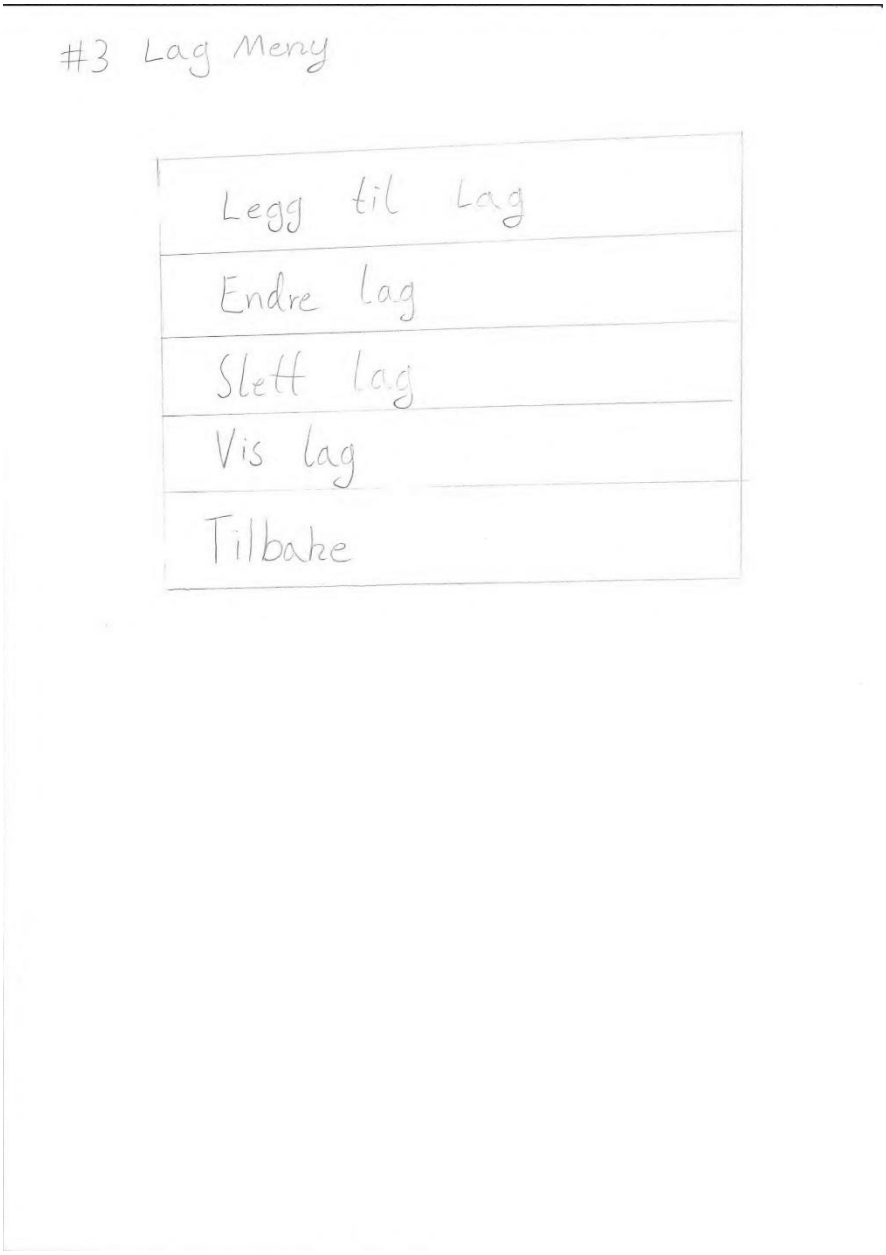


Figure 18

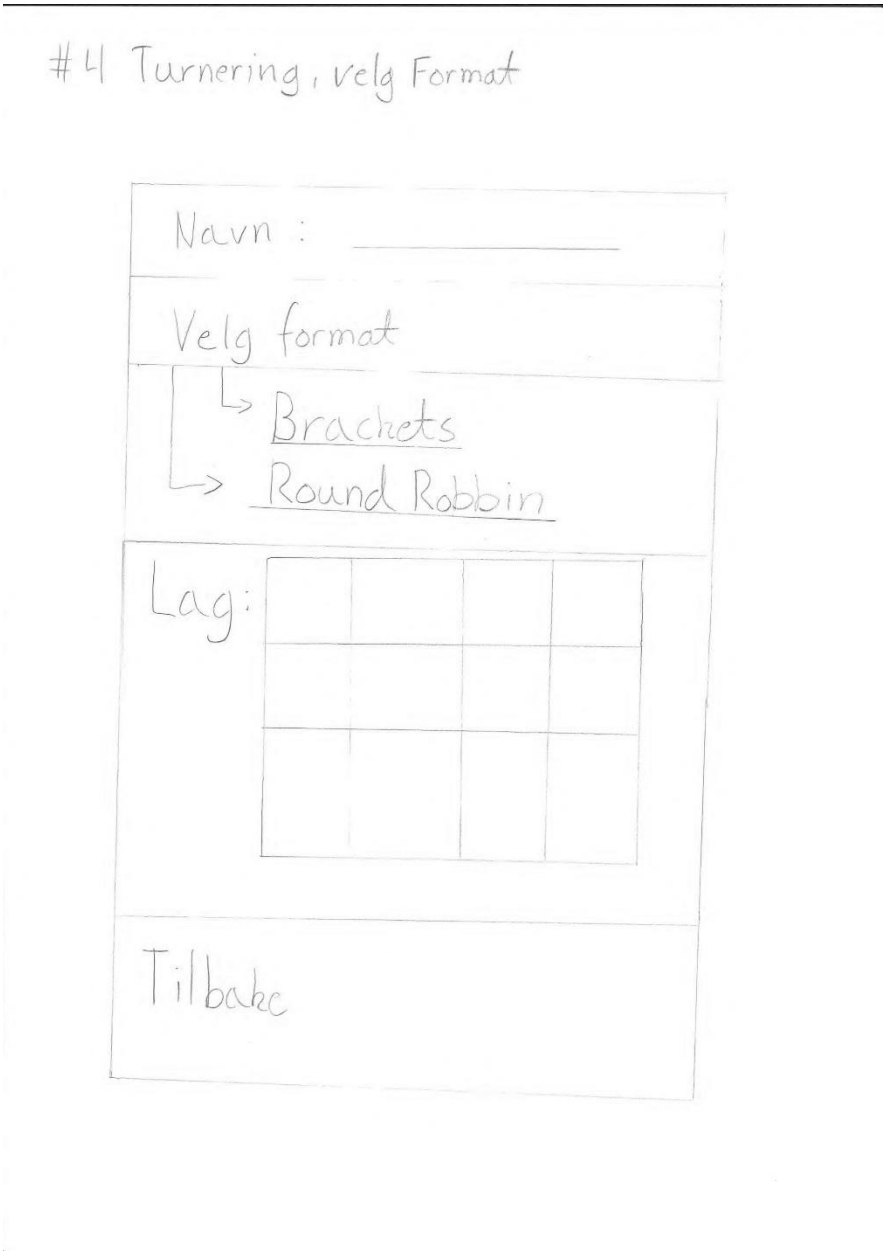

#4 Turnering, velg Format
Navn :
Velg format
Brackets
Round Robbin
Lag:
Tilbake

Figure 19

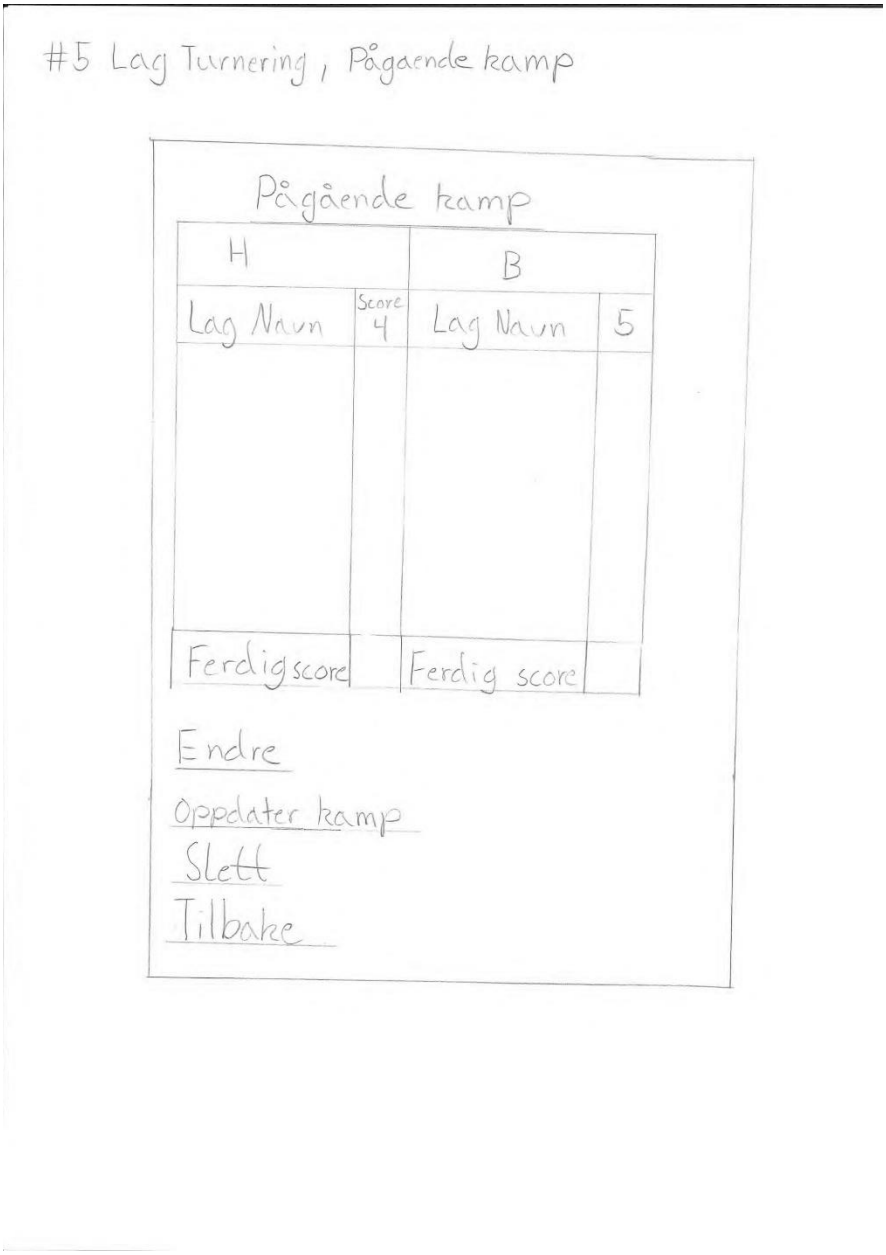

#5 Lag Turnering, Pågående kamp
Pågående kamp
H
B
Lag Navn
Score 4
Lag Navn
5
Ferdigscore
Ferdig score
Endre
Oppdater kamp
Slett
Tilbake

Figure 20

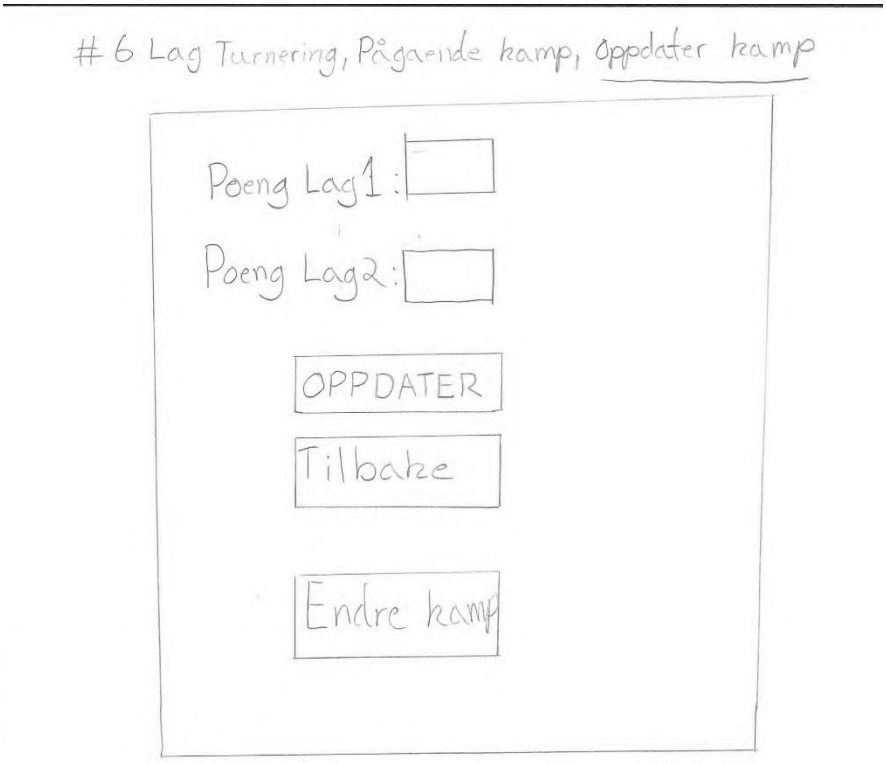

#6 Lag Turnering, Pågående kamp, oppdater kamp
Poeng Lag1:
Poeng Lag2:
OPPDATER
Tilbake
Endre kamp

Figure 21

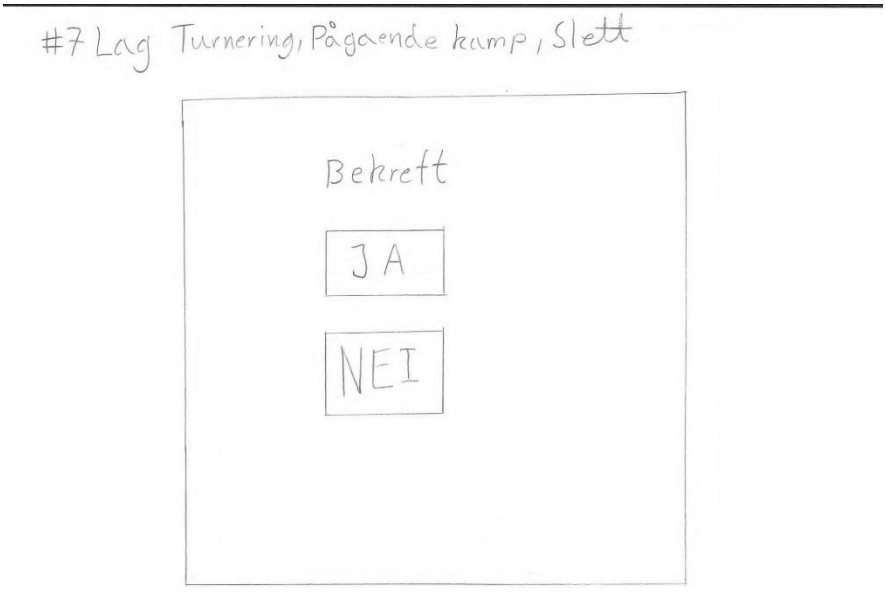


Figure 22

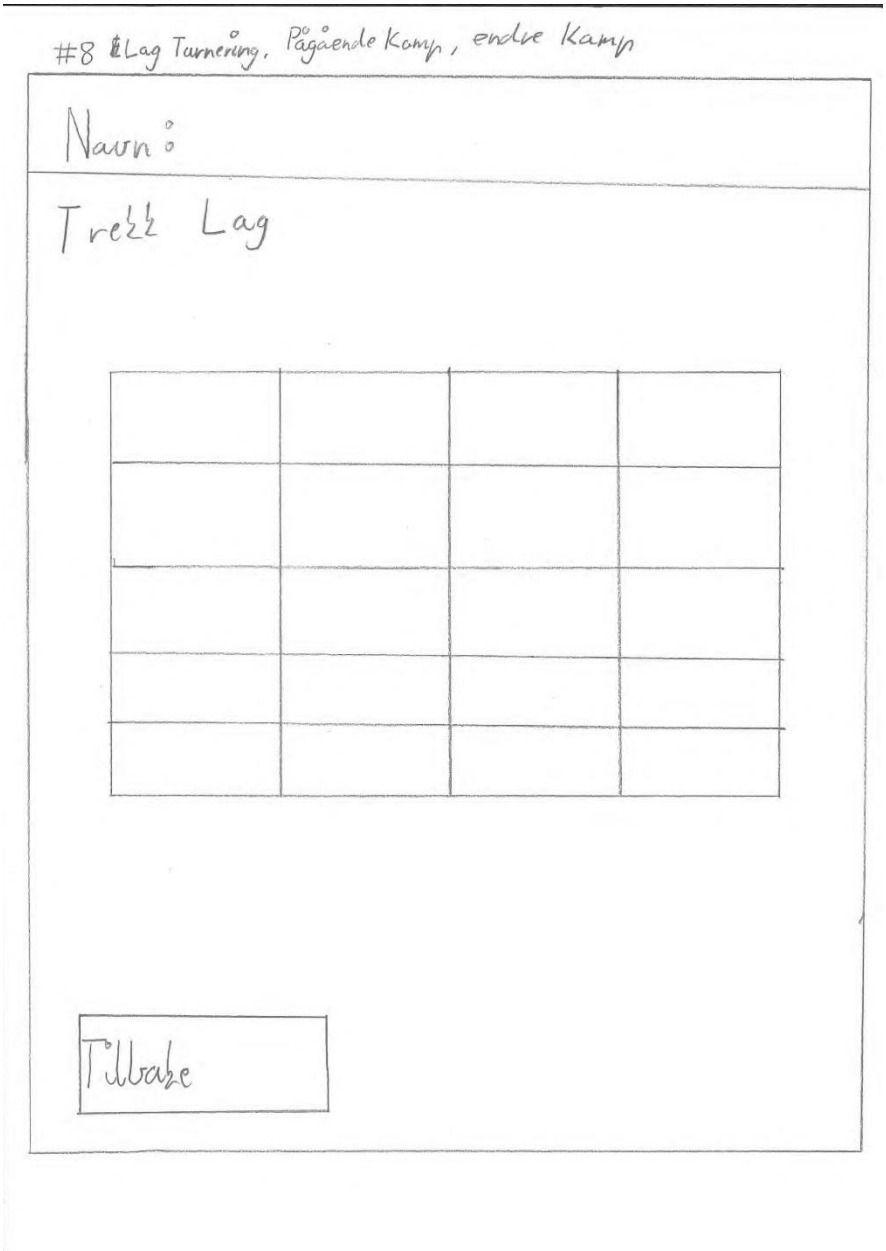

#8 8Lag Turnering, Pågående Kamp, endre Kamp
Navn:
Trekk Lag
Tilbake

Figure 23

17.3 Paper prototype CLI

Wireframe turneringsprogram

Turnering – 1

Lag – 2

Avslutt – 0

Legg til lag

Vis lag

Endre lag

Slett lag

Tilbake - 0

Lag turnering – N

Turnering A - 1

Turnering B - 2

Turnering C - 3

Turnering D - 4

Tilbake – 0

Action: Skriv navn: *Userinput*

↓

Brackets – 1

Round Robin – 2

Avbryt - 0

Action: Velg format: *Userinput*

↓

Lag 1 – 1
Lag 2 – 2
Lag 3 – 3
Lag 4 – 4

Avbryt - 0

Action: Velg første laget som er med: *Userinput*

↓

Lag 1 – 1
Lag 2 – 2
Lag 3 – Valgt
Lag 4 – 4
Ferdig - F

Avbryt – 0

Action: Velg andre laget som er med: *Userinput*

Osv til alle lag er valgt

↓

Printout: Turnering satt opp.

Tilbake til turnering meny.

Pågående kamper:

Hjemme - Borte - Kamp nr

Lag 1 4p - Lag 3 3p - 1

Lag 2 2p - Lag 4 1p - 2

Endre score – 1

Trekk lag (fra kamp) - 2

Slett kamp – 3

Tilbake – 0

Hvis 1: Hvilken kamp vil du endre på: *Userinput*

Lag 1 4p - Lag 3 3p - 1

Hvilket lag vil du endre på (1 eller 3): *Userinput*

↓

Skriv mengden poengene i lag *** økes: *Userinput*
(f.eks 3 eller -5)

Lag 1 – 1
Lag 2 – 2
Lag 3 – 3
Lag 4 – 4

Avbryt - 0

Skriv laget du vil trekke: *Userinput*

↓

Lag *** er trukket fra pågående kamp.

Tilbake til pågående kamper.

Pågående kamper:

Hjemme	-	Borte	-	Kamp nr
Lag 1	-	Lag 3	-	1
Lag 2	-	Lag 4	-	2

Avbryt - 0

Skriv kampen du vil slette: *Userinput*

↓

Kamp *** er slettet.

Tilbake til pågående kamper.