Maximus Cup Arranger Vision

Version 2.0

Innholdsfortegnelse

Revision History

Date	Version	Description	Author
11.03.2022	V1	First iteration of the Vision document.	All team members
22.03.2022	V2	Second iteration of the Vision document.	All team members

Table 1

Vision

1 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.

1.1 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.

1.2 References

[1] StatCounter (2022, 01. February). *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, 08. January). *Football-game.* Retrieved 06.03.2022 from https://github.com/thekush7/Football-game-Football-club-management-system

[4] cplusplus.com (2021). <*random*>. Retrieved 06.03.2022 from https://www.cplusplus.com/reference/random/

[5] Logitech Europe S.A (2022). *Tournament Bracket Generator*. Retrieved 06.03.2022 from https://challonge.com/no/tournaments/bracket_generator.html

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

1.3 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.

2 Positioning

2.1 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.

2.2 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.

2.3 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 (Cup Manager, 2022), 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.

3 Project goals

3.1 Impact goals

- Expanding the company.
- Increase profits.
- Increase client/customer base.
- Brand recognition.

3.2 Result goals

- Deliver a user-friendly football tournament application, which allows the user to manage multiple teams and tournaments.
- Increase loyalty to our customers.

- Increase user productivity.
- Increase the user's profit.
- 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.

3.3 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.

4 Stakeholder and User Descriptions

4.1 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 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.

4.2 Stakeholder Summary

Shown below is a summary of current identified stakeholders, their descriptions, and responsibilities.

Name	Description	Responsibilities
Developer	The developer oversees development, maintainability, and documentation of the application.	- Ensures that the system will be maintainable. - 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. - Ensures that there will be a market demand for the product's features.

Table 2

4.3 User Summary

Shown below is a list of all current identified users with descriptions and their associated responsibilities.

Name	Description	Responsibilities

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.

Table 3

4.4 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.

4.5 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	CLI
Adding multiple teams	High	No concerns	N/A	CLI
Edit or delete teams	High	No concerns	N/A	CLI
View teams and tournament	High	No concerns	N/A	CLI
Persistent storage	High	User-friendliness	N/A	Database

Table 4

4.6 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

Table 5

5 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

5.1 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* (Logitech Europe S.A, 2022). 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 being 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.

5.2 Summary of Capabilities

The program should be 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.

5.3 Risk analysis

1. Project slowed down, as a result of team members getting sick.
2. Corona situation might develop and result in university closing.
3. Project scope.
4. Coding issues.
5. Final phase project dependencies.
6. Loss of data.

Risk factor	**Probability**	**Severity**	**Consequence**	**Action**
Illness	High	Low	Will vary. May end up causing delays in certain tasks.	Delegate outstanding tasks to other members.
Covid-19 Lockdown	Low	High	May end up causing delays	Collaborate as efficiently as

				possible online.
Lacking time management	Medium	Medium	Work could end up not being finished by set deadlines.	Set realistic goals and delegate tasks based on each other's strongest roles.
Broken code	Medium	High	May end up causing major project delays.	Ensure good time management, and only push working code to the main branch
Loss of data	Low	High	May end up needing to do work over again	Delegate missing tasks to other members.

Table 6

Risk Matrix		Consequences				
		Insignificant	Minor	Moderate	Major	Severe
Probability	Almost Certain	Mid	High	Very High	Very High	Very High
	Likely	1	High	High	Very High	Very High
	Possible	Low	Mid	3	4	Very High
	Unlikelyh	Low	Low	Mid	2,5	High
	Rare	Low	Low	Low	Low	Mid

Figure 1

Risk factor description:

1. The 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 to the low-risk group with a high probability. This is because we as a group can take on the extra workload while our team member recuperates.
2. Probability is 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 clients will all be affected. This will highly affect our rate of progress. If such a situation occurs, we will handle it by working online, which is a decent substitute at best.

3. As a team we have an estimate 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. For now, our opinion is that the probability is small/medium, and the consequences are medium.
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 program without templates or guidance. This could lead to progression loss on the project trying to figure out how to solve new coding issues. Because of our limited time on this project, we are considering this risk of having medium probability and high consequences.
5. Loss of data is something the group considered when the project started. It is something that happens from time to time but working with Git and pushing up new data after every work session will hopefully minimize the risk of big data losses. Using Git we are also able to go back to older versions of files in our repository if needed. Because of this we have judged data loss to be of low probability with low consequences. This can vary depending on what data is lost. If we find ourselves in a situation where one of the team members has lost data, we will first consider how much work it will take to get it back and then distribute the work within the team. Every team member is also working proactively to minimize the chance of losing data by saving their work locally as often as they judge sensible.

5.4 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 testing	5	0	2	0
Total cost				857500

Table 7

5.5 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

5.6 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.

5.7 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 Microsoft Windows operating system.

6 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.

7 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.

8 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.

9 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

10 Other Product Requirements

10.1 Performance and System Requirements

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

11 Documentation Requirements

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

12 A Feature Attributes

12.1 A.1 Status

Features added to our MVP are listed below. Some of these features still need to be tweaked, changed or optimized, based on user-test feedback.

Proposed						
Approved		Update		Edit/Delete		
Implemented	Tournament		Add		View	Database

Table 8

12.2 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” (cplusplus.com, 2021). 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.

12.3 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. Changes to certain features will still require us to re-engineer parts of other features.

12.4 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.

12.5 A.5 Assigned To

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

12.6 A.6 Reason

The requirements for our project can be found on Blackboard. 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.