Group 1 - Asura Tournament Wiki

Asura Tournament System

DCST1008 Systemutvikling - Group 1

Members: Felix Albrigtsen, Kristoffer Longva Eriksen, Jonas Jødestøl Haugland, Kristoffer Juelsen

Introduction

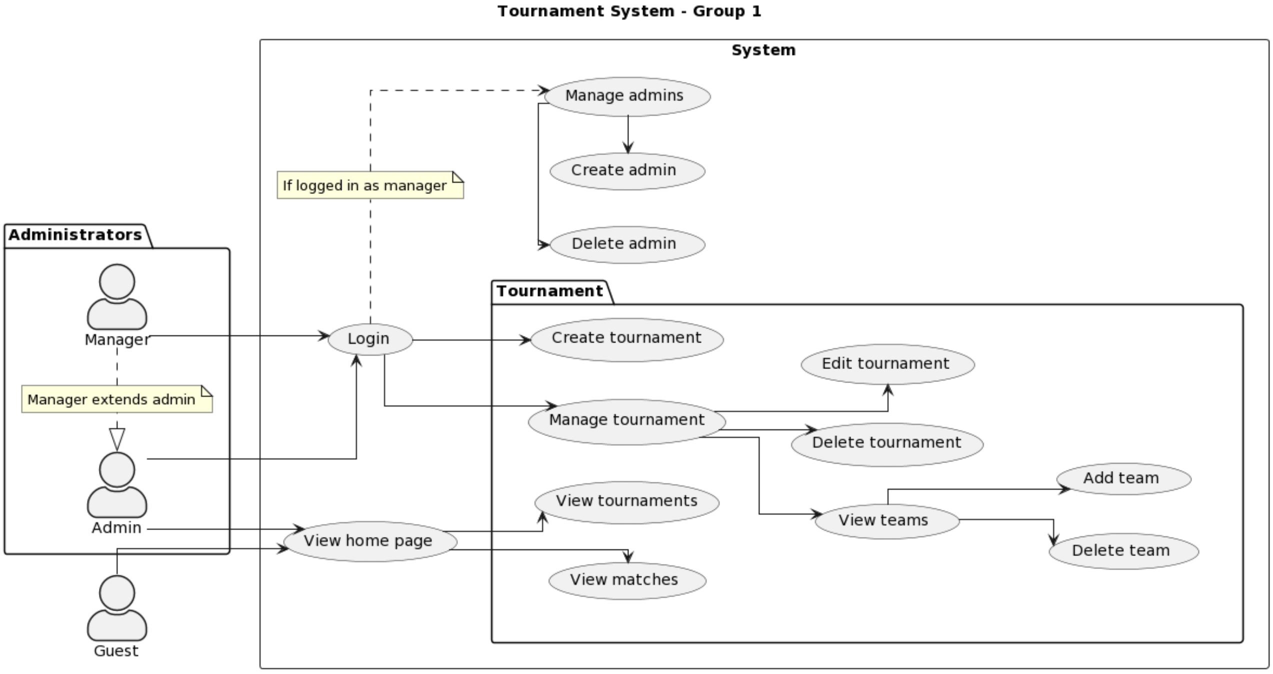
A complete description of our assignment, customer, planning, and methodology is described in the Vision document (PDF). The wiki page is intended to be read online on NTNU-IDI GitLab and *not* in PDF or any other non-interactive format.

Wiki Contents

Documentation

System

Use Case Diagram



Use Cases

Use Case 1 – Create admin

Action	**Create admin**
Actor	Manager
Pre-conditions	Logged in as manager
Post-conditions	A new admin is created
Basic path	1. Manager enters “View Admins” page
	2. Manager enters “Create new admin” page
	3. Manager fills inn necessary information and submits the form
Exception	3.1 Manager doesn’t fill in admin name
	3.2 Manager doesn’t fill in initial password
	3.3 Manager doesn’t submit
Additional information	The following must be entered to create an admin:
	- Admin username
	- Initial password

Use case 2 – Create tournament

Action	**Create tournament**
Actor	Admin
Pre-conditions	Logged in as admin or manager
Post-conditions	A new tournament is created
Basic path	1. Admin enter “Create New Tournament” page
	2. Admin fills inn necessary information
	3. Admin submits form
Exception	2.1 Admin doesn’t fill in “Tournament name” field
	2.2 Admin doesn’t fill in start and end time
	3.1 Admin doesn’t submit the form
Additional information	The following must be entered to create a tournament:
	- Tournament name
	- Tournament start and end time

Use-case 3 – Create team

Action	**Create team**
Actor	Admin
Pre-conditions	Logged in as an administrator
Post-conditions	A new team has been created
Basic path	1. Admin enters “View teams” page
	2. Admin enters “Manage teams” page
	3. Admin enters “Create new team” page
	4. Admin fills inn necessary information
	5. Admin submits form
Exception	4.1 Admin doesn’t fill in “Team name” field
	4.2 Admin doesn’t fill in any team members
	5.1 Admin doesn’t submit the form
Additional information	The following must be entered to create a team:
	- Team name
	- Team members

Use-case 4 – Delete admin

Action	Delete admin
Actor	Manager
Pre-conditions	Logged inn as manager
Post-conditions	An admin is deleted
Basic path	1. Manager enters “View Admins” page
	3. Manager enters “Remove Admin”
	4. Manager selects the intended admin(s) to remove
	5. Manager confirms choice
Exception	3.1 Manager can’t remove another manager
	3.2 Admin no longer exists, already deleted from database
	5.1 Manager doesn’t confirm choice
Additional information	The following must be true to delete an admin
	- Selected admin must still exist in the database
	- Manager must confirm choice

Use-case 5 – Manage tournament

Action	Manage tournament
Actor	Admin
Pre-conditions	Logged in as an administrator
Post-conditions	Selected tournament has been edited and saved
Basic path	1. Admin enters “Manage Tournament” page
	2. Admin fills inn desired information
	3. Admin submits changes
Exception	3.1 Admin doesn’t submit
Additional information	To save the edited information at least one field must be filled out

Use-case 6 – Delete tournament

Action	Delete tournament
Actor	Admin
Pre-conditions	Logged in as an administrator
Post-conditions	The selected tournament is deleted
Basic path	1. Admin enters “Manage Tournament”
	2. Admin deletes tournament
Exception	There are no active tournaments
Additional information	

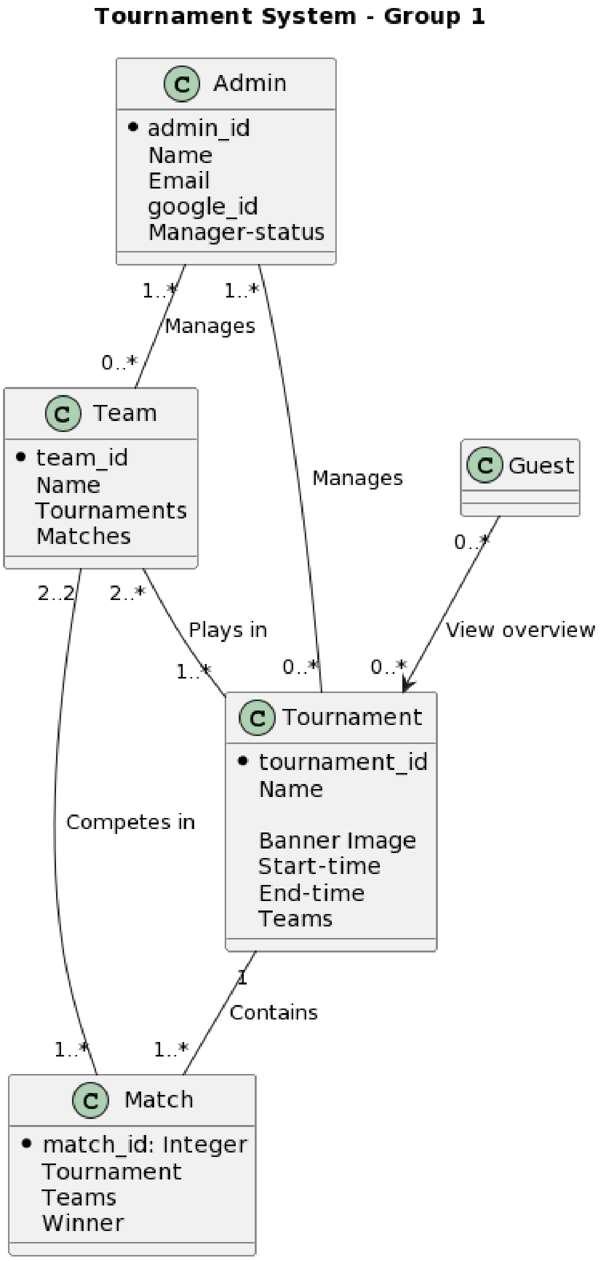
Use-case 7 – Manage team

Action	Manage team
Actor	Admin
Pre-conditions	Logged in as an administrator
Post-conditions	Selected team has been edited and saved
Basic path	1. Admin enters “View teams” page
	2. Admin enters “Manage team” page
	3. Admin fills inn desired information
	4. Admin submits changes
Exception	3.1 There are no teams created
	4.1 Admin doesn’t submit
Additional information	To save the edited information at least one field must be filled out

Use-case 8 – Delete team

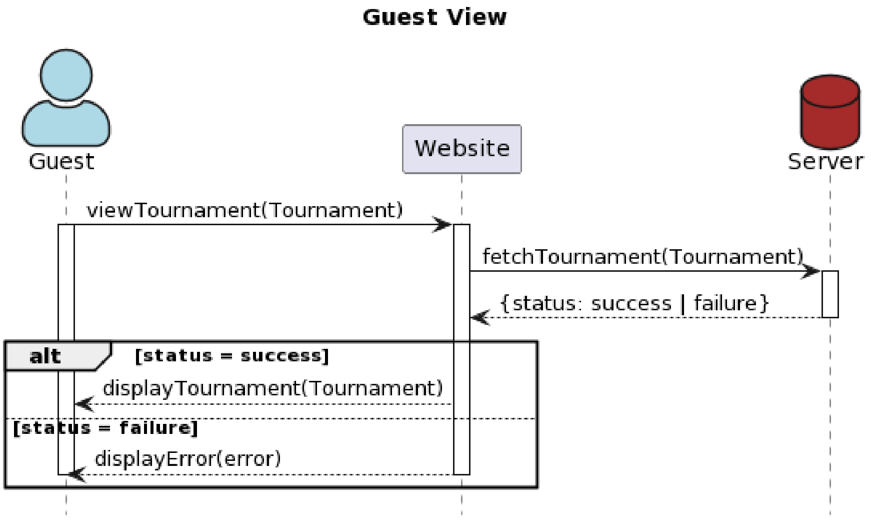
Action	Delete team
Actor	Admin
Pre-conditions	Logged in as an administrator
Post-conditions	Selected team has been deleted
Basic path	1. Admin enters “View teams” page
	2. Admin enters “Manage team” page
	3. Admin deletes team
Exception	2.1 There are no teams created
Additional information	

Domain Model

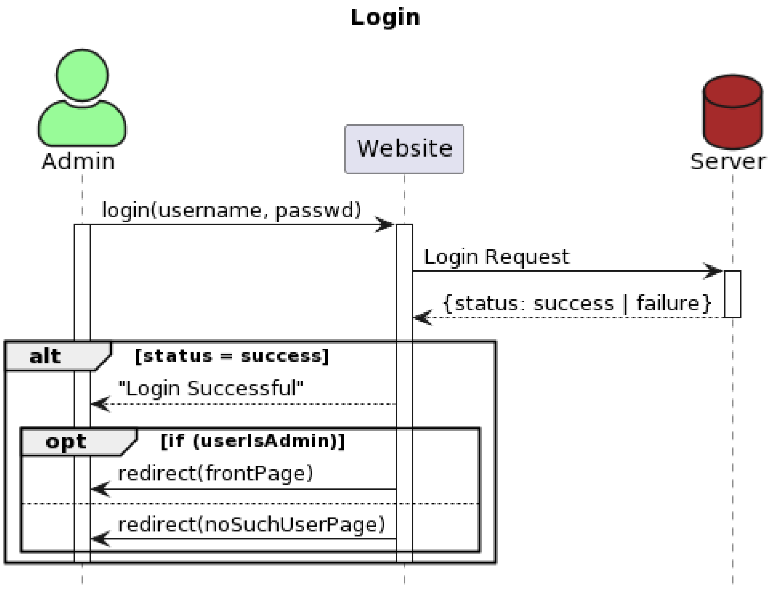


Sequence diagrams

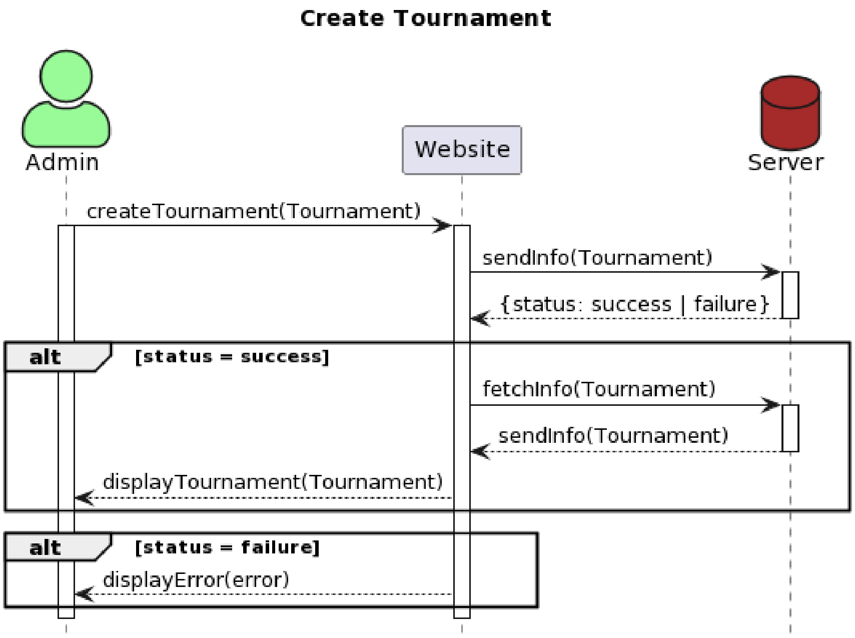
Guest Sequence Diagram


Guest View
Guest
Website
Server
viewTournament(Tournament)
fetchTournament(Tournament)
{status: success | failure}
alt
[status = success]
displayTournament(Tournament)
[status = failure]
displayError(error)

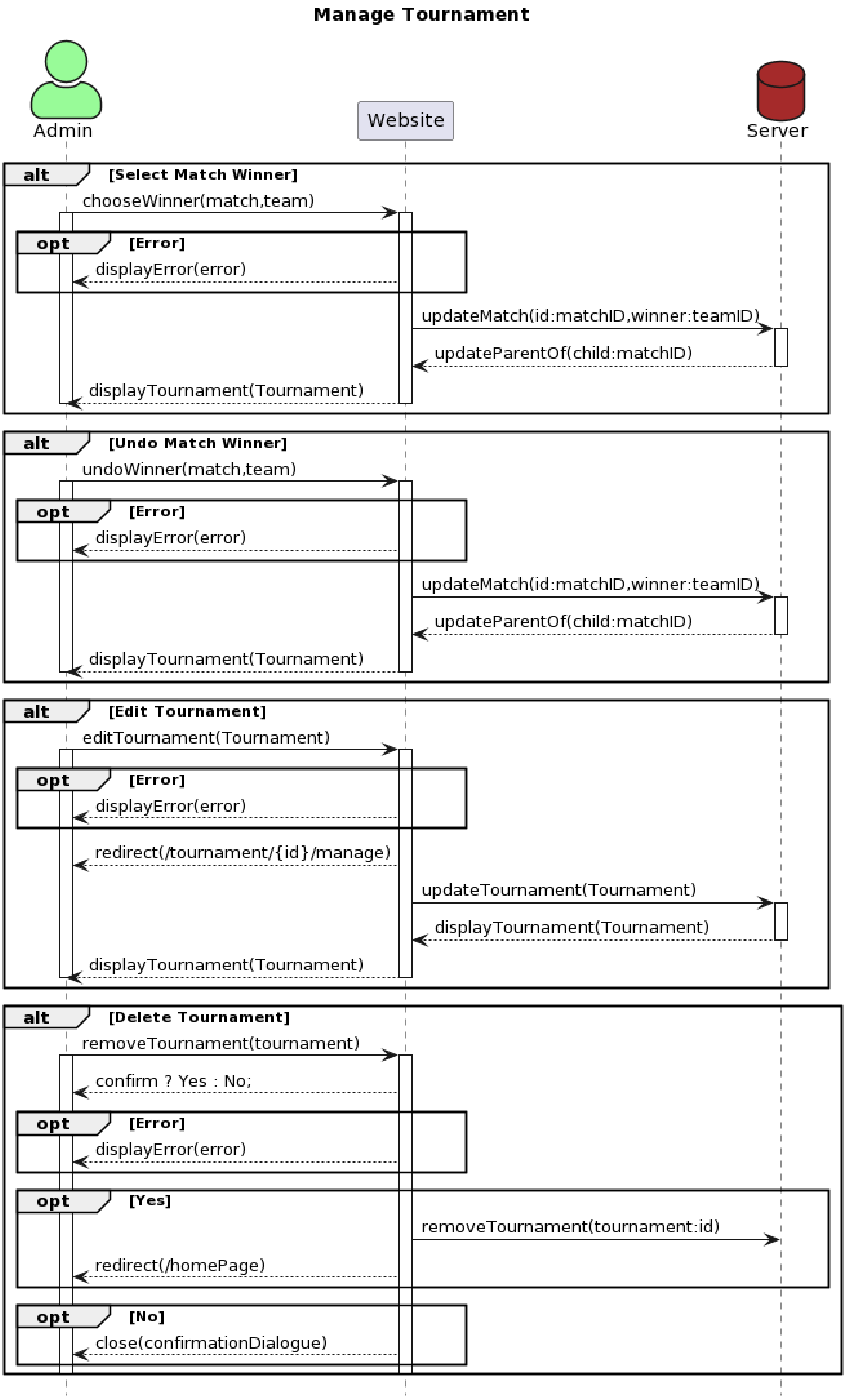
Login Sequence Diagram



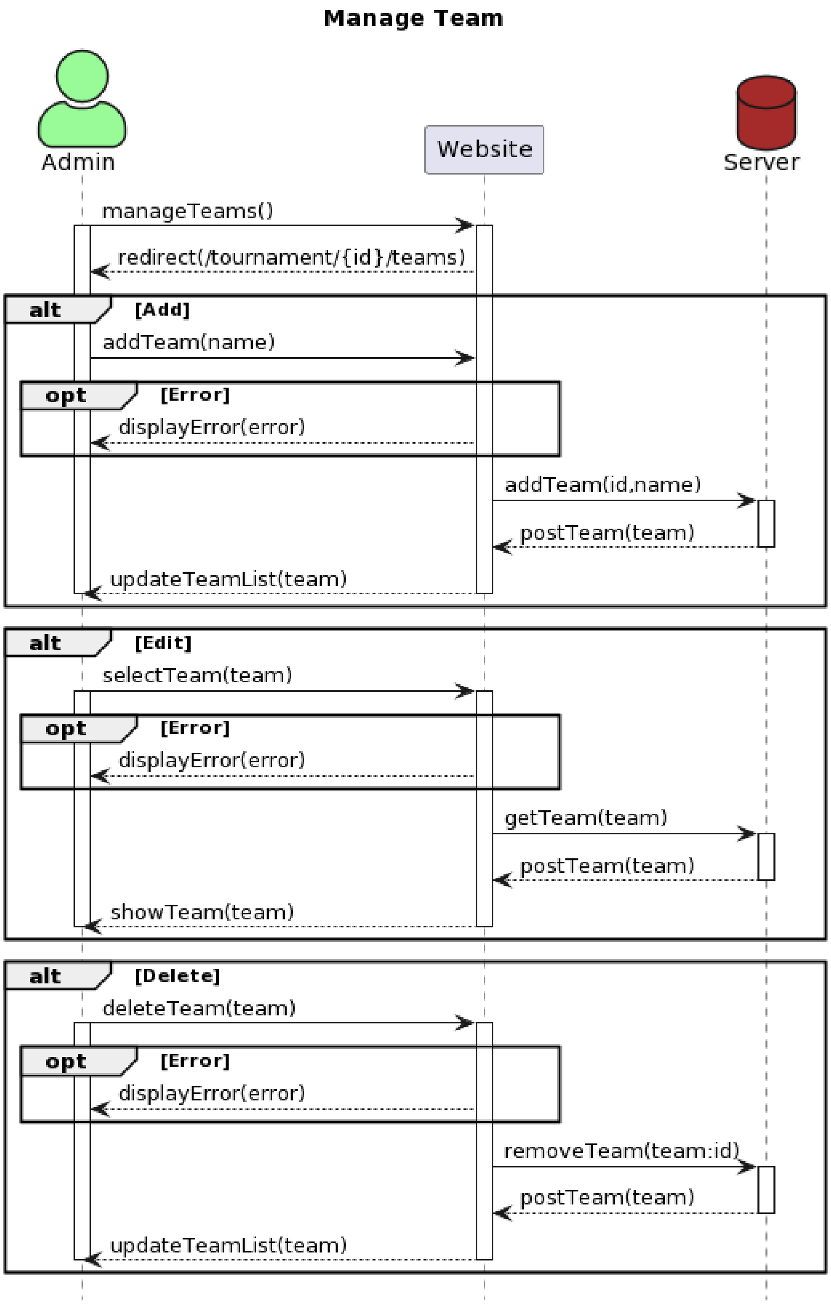
Create Tournament Sequence Diagram



Manage Tournament Sequence Diagram

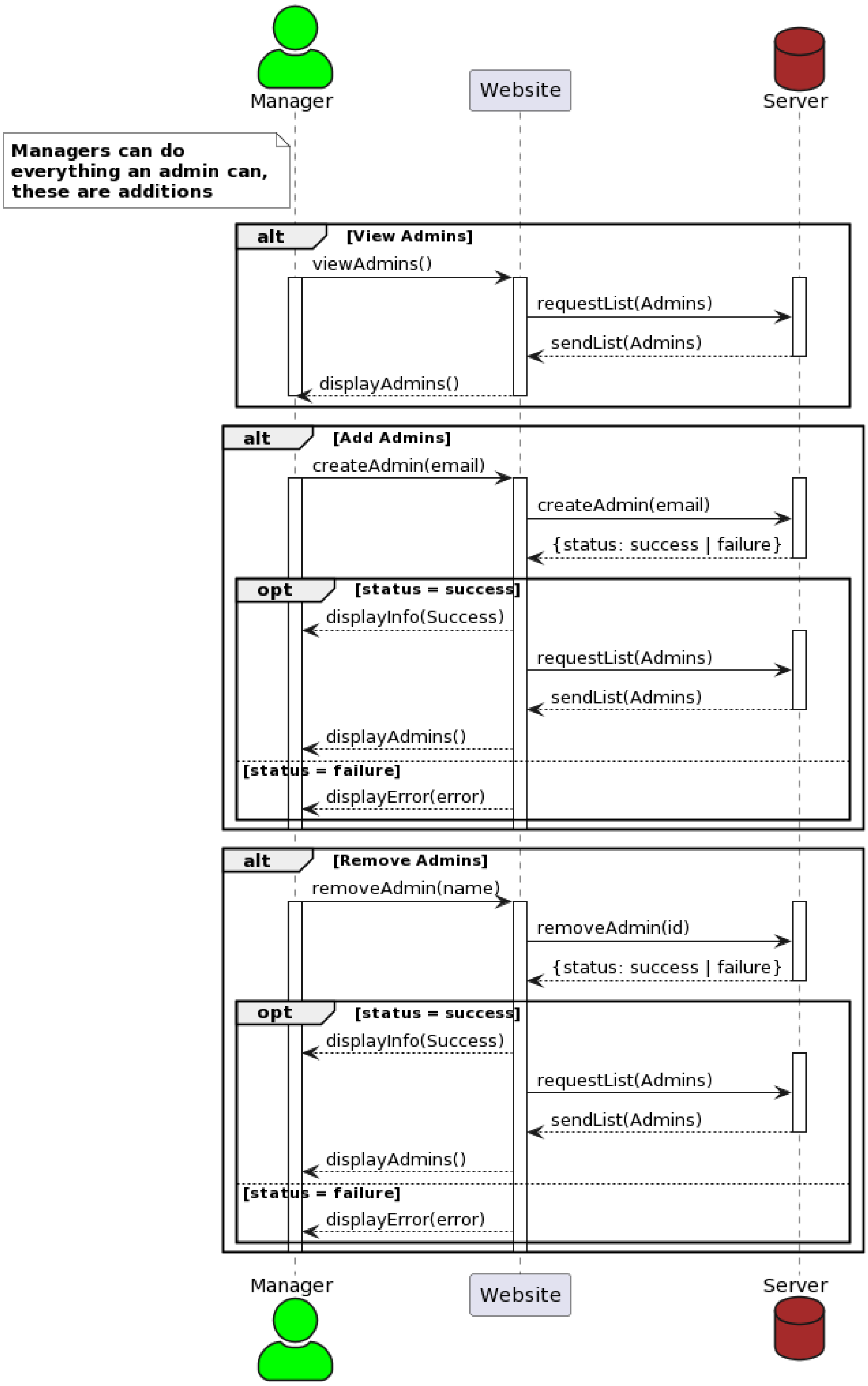

Manage Tournament
Admin
Website
Server
alt [Select Match Winner]
chooseWinner(match,team)
opt [Error]
displayError(error)
updateMatch(id:matchID,winner:teamID)
updateParentOf(child:matchID)
displayTournament(Tournament)
alt [Undo Match Winner]
undoWinner(match,team)
opt [Error]
displayError(error)
updateMatch(id:matchID,winner:teamID)
updateParentOf(child:matchID)
displayTournament(Tournament)
alt [Edit Tournament]
editTournament(Tournament)
opt [Error]
displayError(error)
redirect(/tournament/{id}/manage)
updateTournament(Tournament)
displayTournament(Tournament)
displayTournament(Tournament)
alt [Delete Tournament]
removeTournament(tournament)
confirm ? Yes : No;
opt [Error]
displayError(error)
opt [Yes]
removeTournament(tournament:id)
redirect(/homePage)
opt [No]
close(confirmationDialogue)

Manage Team Sequence Diagram



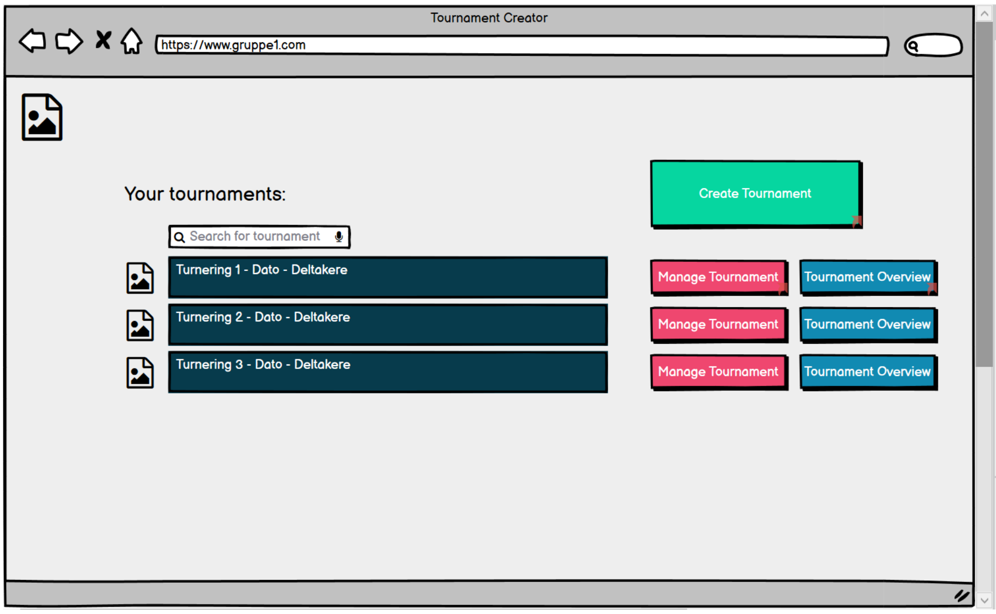
Sequence Diagram for Manager View and Actions

Manager View



Wireframe

Tournaments



Tournament Creator

https://www.gruppe1.com

Your tournaments:

Search for tournament

Turnering 1 - Dato - Deltakere

Turnering 2 - Dato - Deltakere

Turnering 3 - Dato - Deltakere

Start Page

Notes

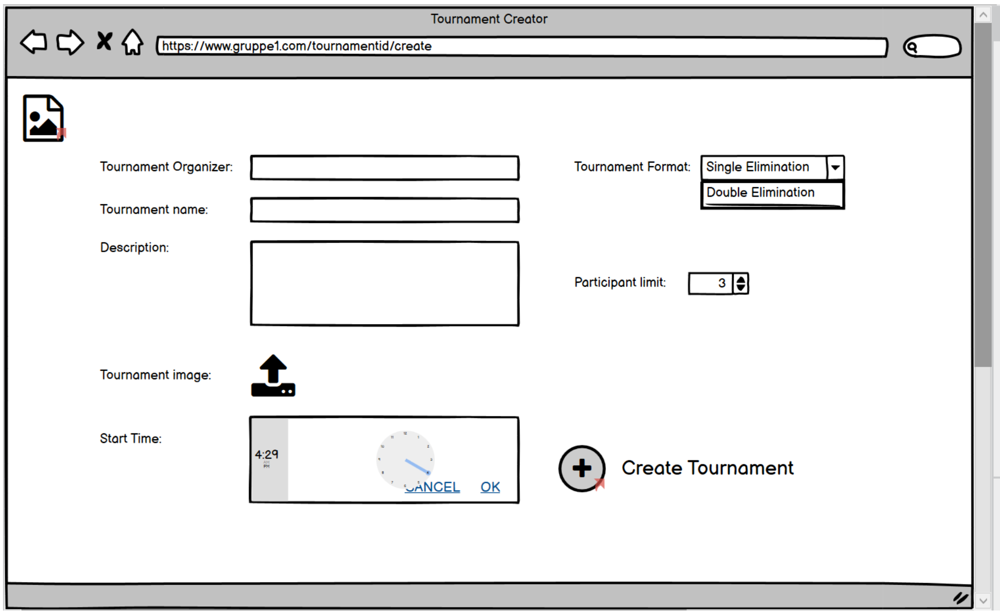
This Wireframe shows the basics of how or product will look. On the start page, the user can see a list of tournaments they are already administrating. They can choose to check on or manage one of these tournaments or create a new tournament.

The colour pallet of the start page is an example, and subject to change based on feedback from our customer.

Alternate Versions

Click '+' to create an alternate version of this wireframe

Create Tournaments



Tournament Creator

https://www.gruppe1.com/tournamentid/create

Tournament Organizer:

Tournament name:

Description:

Tournament image:

Start Time: 4:29

Tournament Format: Single Elimination

Double Elimination

Participant limit: 3

Create Tournament

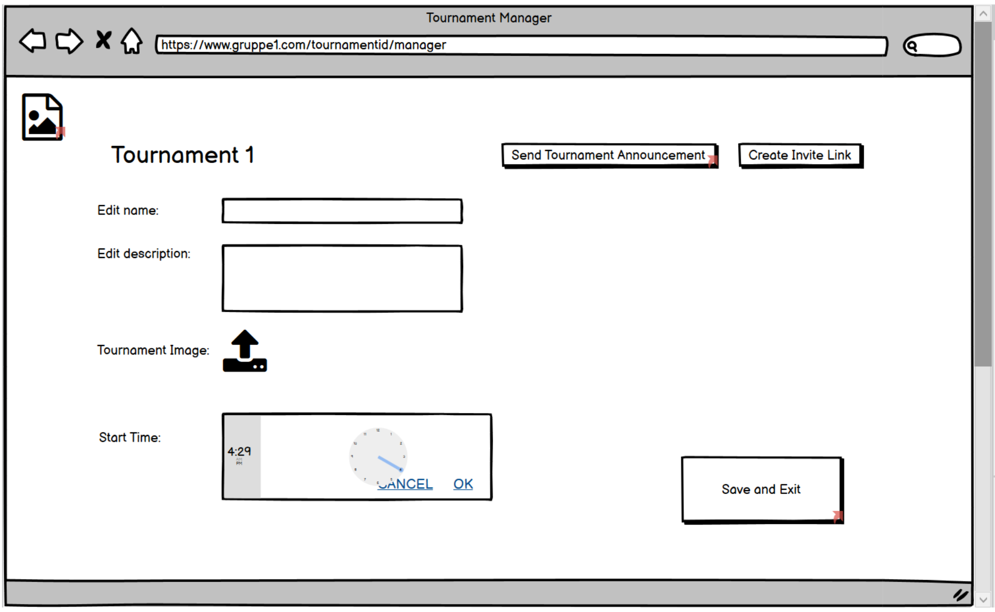
Notes

If the user wants to create a new tournament they are sent to this page. Here they have the ability to change how the tournament will appear and function.

Alternate Versions

Click '+' to create an alternate version of this wireframe

Manage Tournement



Manage Tournament

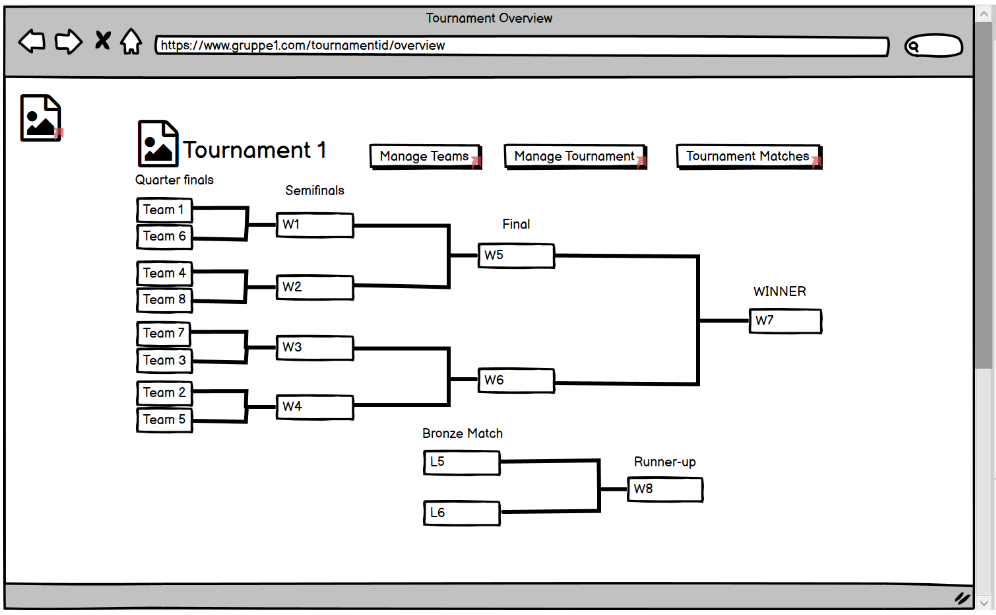
Notes

If the user wants to manage an existing tournament they will be sent to this page. Here they can change the appearance of the tournament, set a different start time or send a tournament announcement. Once they save and exit they will be sent back to the page they came from (start page or tournament overview).

Alternate Versions

Click '+' to create an alternate version of this wireframe

Tournament Overview



Tournament Overview

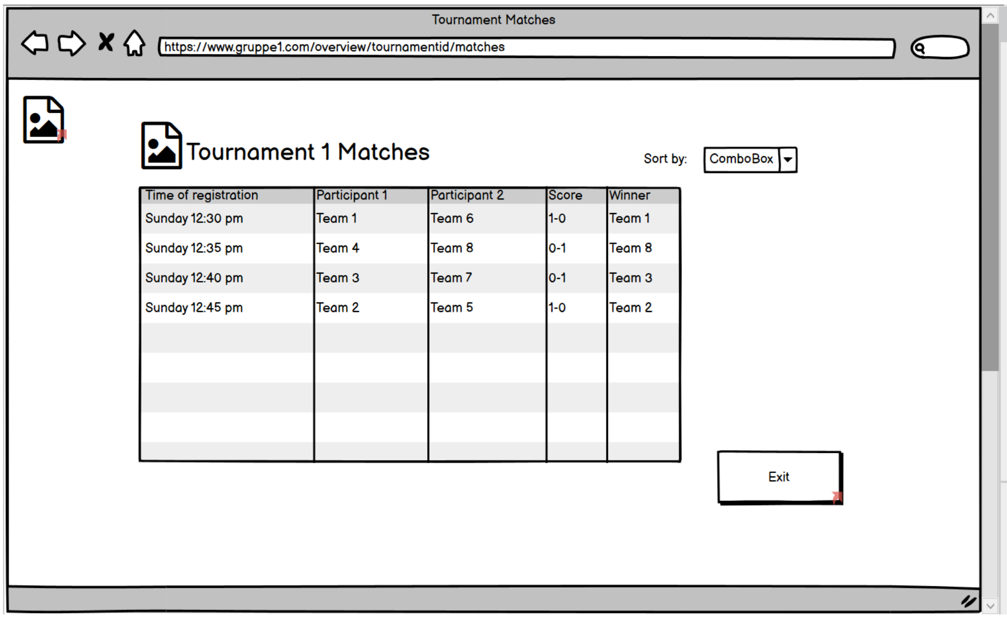
Notes

This is an example of a tournament overview with a single elimination format. Here, the user has the ability to see the status of their tournament, view tournament matches, manage teams or enter the tournament management page.

Alternate Versions

Click '+' to create an alternate version of this wireframe

Tournament Matches



Tournament Matches

https://www.gruppe1.com/overview/tournamentid/matches

Tournament 1 Matches

Sort by: ComboBox

Time of registration	Participant 1	Participant 2	Score	Winner
Sunday 12:30 pm	Team 1	Team 6	1-0	Team 1
Sunday 12:35 pm	Team 4	Team 8	0-1	Team 8
Sunday 12:40 pm	Team 3	Team 7	0-1	Team 3
Sunday 12:45 pm	Team 2	Team 5	1-0	Team 2

Tournament Matches

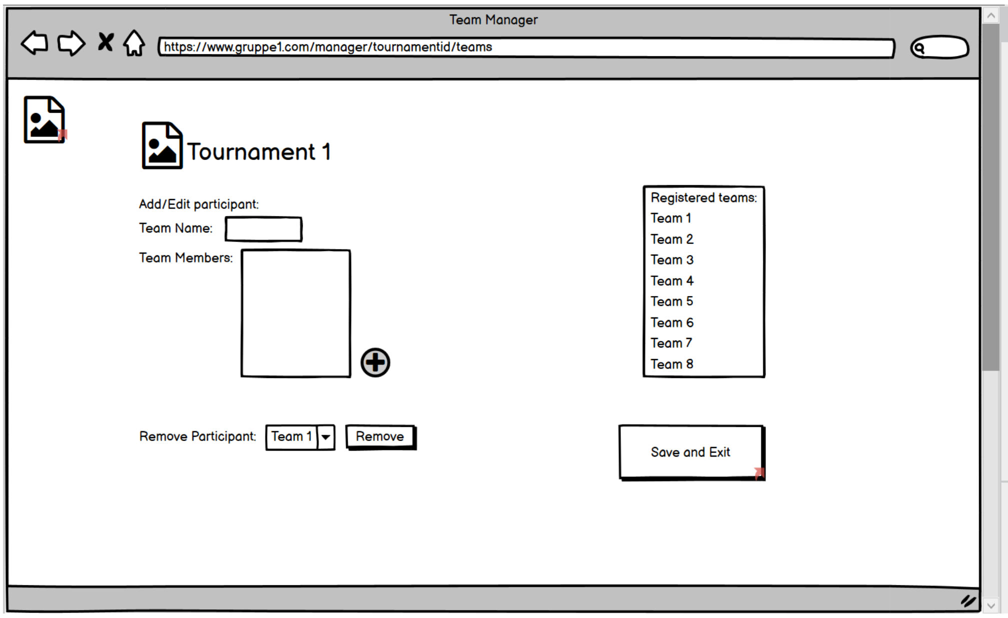
Notes

This page has a simple overview of finished matches in the current tournament.

Alternate Versions

Click '+' to create an alternate version of this wireframe

Team Manager



Team Manager

https://www.gruppe1.com/manager/tournamentid/teams

Tournament 1

Add/Edit participant:

Team Name:

Team Members:

Registered teams:
Team 1
Team 2
Team 3
Team 4
Team 5
Team 6
Team 7
Team 8

Remove Participant: Team 1

Team Manager

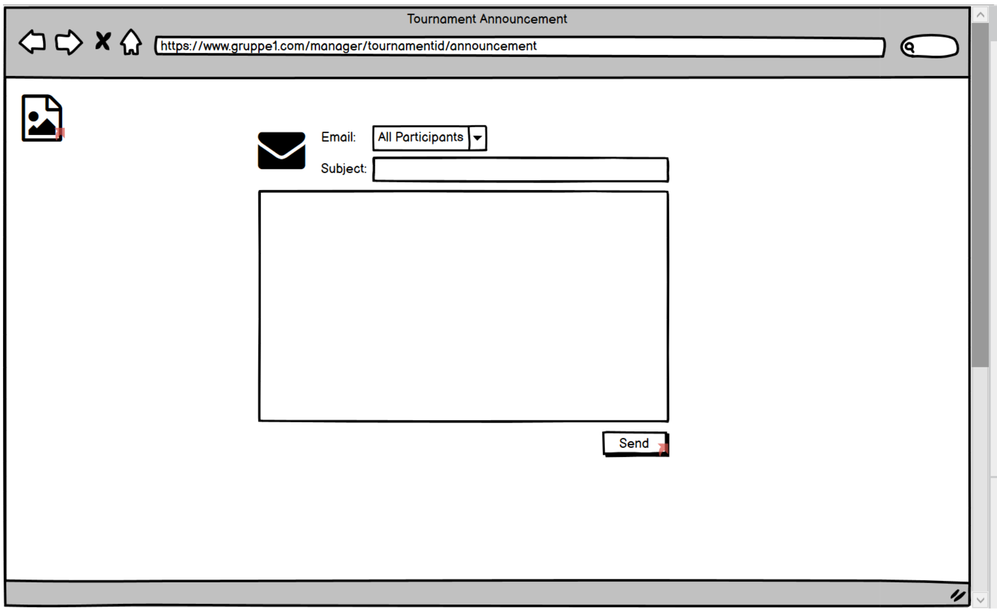
Notes

On this page the user can add, edit or remove teams. They can select a team from the registered teams to edit, or add a new team. The bracket will be updated when a new team is created.

Alternate Versions

Click '+' to create an alternate version of this wireframe

Announcements



Tournament Announcement

https://www.gruppe1.com/manager/tournamentid/announcement

Email: All Participants

Subject:

Announcements

Notes

On this page the user can send critical information to all or some of the tournament participants. It has yet to be decided whether this will be done over email or discord.

Alternate Versions

Click '+' to create an alternate version of this wireframe

Usability Tests

Usability test 28.02.2022 – Initial test (Wireframes)

PURPOSE

- Test UI and UX of the wireframe prototype to find improvements to implement in the MVP.
- Test color scheme and other elements that needs to be WCAG 2.1 - 1 compliant.

GIVEN TASKS

1. Create Tournament and manage it
2. Create and manage teams and players
3. See tournament overview and overview of finished matches
4. Give feedback on color scheme/contrasts and button sizes etc.

RESULTS

1. The tester successfully created a tournament and navigated to the manage window.
   - *Comments:* User could not quickly find the “Create Tournament” button.
2. The tester successfully navigated to the team and player management window.
   - *Comments:* The tester was satisfied with the UI.
3. Tester was presented with the tournament overview after creation, overview of tournaments was also a part of the home page.
4. Briefly commented on a button, with green background color and white text, which had too little contrast. This was expected and part of our color scheme testing.

CONCLUSION

The customer was quite happy with our MVP but had a few things they wanted to be changed.

- The brackets need to show winners, and how to select and change them if the wrong team is picked more clearly.
- The Navigation between pages should be improved. We implemented a navigation bar to fix this.

Usability test 28.03.2022 – 1st Iteration – internal (MVP)

PURPOSE

- Test UI and UX for the MVP to find improvements to implement in the final product.
- Test color scheme and other elements that need to be WCAG 2.1 - 1 compliant.

GIVEN TASKS

1. Create tournament
2. Manage tournament
3. Delete tournament
4. Create team
5. Edit team
6. Delete team
7. Select winners of matches
8. Maneuver around the web application without issues

RESULTS

1. Tournament was created with expected size
2. Tournament details were altered and saved, as expected
3. Tournament was deleted, as expected
4. Teams created with given name. Feature for adding members was not yet implemented.
5. Team details were altered and saved as expected
6. Team was deleted as expected
7. The selected winner was picked and automatically filled in the next tiered match.
   - *Comment*: No intuitive way of knowing how to select winners, and no obvious option for undoing selections. Brackets will be changed to make this more intuitive.
8. All buttons lead to the expected page.

    - *Comment*: None of the pages had a "back" button which made navigating back to the previous page cumbersome. A tournament bar was added to fix this.

CONCLUSION

The customer was quite happy with our Wireframe but had a few things they wanted to be changed.

- The contrast ratio of some of the colors we picked for the prototype was to low, new colors needs to be picked.
- Buttons needs to be more visible and obvious.

Usability test 29.03.2022 – 2nd Iteration (MVP)

PURPOSE

- Test UI and UX for the MVP and discover improvements for final iteration.
- Test navigation of the site for an external user.
- Gather feedback of overall feel and look of the website, including color scheme and layout.
- Gather comments and concerns from the customer regarding future implementations.

GIVEN TASKS

1. Maneuver around the website
2. Create a tournament
3. View a tournament
4. Manage a tournament
5. Delete a tournament
6. Manage teams
7. Create a team
8. Edit a team
9. Delete a team
10. Select winners of matches and "complete" a tournament
11. Comment on the overall feel of the product

RESULTS

1. The users maneuvered the site quite well, with minor hiccups
    - *Comment*: We had a menu button that the users clicked quite frequently, at the time it did not do anything, so we removed it for the MVP showcase to the supervisor. It will be implemented it with function later in the production. Comment: The first user had issue returning to the front page, as the "Home" icon did not have any complementary text to guide the user to click on it, this will be added in future iterations of the product.
2. The users were able to create a tournament smoothly at the expected size with the expected values
    - *Comment*: When choosing the time for the tournament the user rarely pressed the calendar button to easily select a time, and rather wrote it down manually, leading to us finding a "bug" with the time selection input, which will be fixed in future iterations. We will also overhaul the input field for time & date to prevent manual input, while still making it accessible.
3. The users had no trouble viewing the tournaments
    - *Comment*: The user had comments on how the brackets were presented and did not like the feel of it as it were nowhere near perfect at the time of review. We will overhaul the bracket system completely to make the user experience better.
4. The users were easily able to go into the manage tournament and navigate within the system
5. The users were able to delete a tournament, as expected
6. The users navigated to the "manage teams" section without minor issues
    - *Comment*: The user tried adding a team by clicking on the bracket directly at first, which they were then prompted with a "no team selected" alert, which confused them at first. This is something we will improve on and perhaps add in future iterations of the bracket system.
7. The users were able to create a team without issue
    - *Comment*: They liked that they couldn't add more teams than they chose for the tournament
8. The users were able to edit a team without issue
    - *Comment*: The users were pleasantly surprised that the bracket updated automatically and found the edit process very intuitive
9. The users were able to delete a team successfully and without issue
10. The users were confused about how to complete a tournament to begin with, but after a while they were able to complete it, although they found it not very intuitive
    - *Comment*: We will make it more clear how to complete a tournament with buttons, as well as making it clearer who has won and proceeded in the tournament.
11. The users were very happy with the overall layout
    - *Comment*: They did not like the brackets, which will be changed.

CONCLUSION

The overall impression the customer had after the user test was very positive, and they had more ideas for what could be added next rather than complaints. The things that they felt needed improvement have been added as comments under the given tasks, for future development they gave us quite a few ideas to work with.

- Bracket system needs to be completely overhauled.
- Minor additions need to be added to certain elements, mostly specified in the result section above.

We will discuss further implementation of functionality both internally and externally with the customer, to ensure they will be happy with the product. They also gave an impression of things they want added or changed that did not directly influence the product but are more Quality-of-Life changes.

- Better art or find a different cover for the tournaments
- The possibility to add an image or map in the tournament description
- Tournament history tab or page to view completed tournaments
    - Needs to be searchable and if on the main page should be quite small
- Tournament archiving should be flexible as they could drag on, that way they don't get archived the minute they "expire" according to the date and time set.
- A nice addition would be a log that the managers can overlook to view who changes or creates what on the tournament, not a necessary addition but a nice one for sure.
- The possibility for adding prices to the tournament, as most have them and the participants are always asking what they can win or get by participating.
- A login system
    - This is something that is already planned to be implemented
- A possible feedback page, not very high priority
- A countdown until a specific tournament starts

All these comments and suggestions are things that will be taken into consideration when continuing development of the project. Some of the suggestions might not be achievable in the time we have to complete the project but we will do our best to adhere to the customers wishes.

Universal Design

Good universal design is very important for any product. Particularly with web applications like ours, it can be critical to achieving good user experiences.

To achieve great Universal Design, we followed Don Norman's seven fundamental design principles.

Don Norman's seven design principles

1. DISCOVERABILITY (VISIBILITY)

Discoverability offered a certain challenge to us, as we set ambitious goals for our product and wanted to include all functionality we and the user deemed necessary. This meant that the front page needed options for editing or viewing each tournament, creating new tournaments, viewing tournament history, user profiles and logging out. For discoverability we could have added all these functions to the front page of the application, however, this would go against other principles. We decided that the most important functions that would see the most use were creating, editing and viewing tournaments. For this reason, we added obvious functionality for performing these functions on the front page, and placed the history, profile and logout functions in a hamburger menu. We feel that this gives good discoverability as it is easy to understand what the web application is meant to be used for, while also having all the functionality available. All the other pages also have good discoverability, with all functionalities clearly visible and labeled as soon as the page is open. The user will also have an expectation of what a page is used for, as they have clicked a clearly labeled button to get there.

2. FEEDBACK

We are using Material UI snackbars throughout the web application to provide continuous feedback to users. The snackbars have been color-coded (i.e., red for error), to provide the user with an instant understanding of the purpose of the message. When tournaments are created successfully, the user will be notified, and when a tournament creation form is submitted with errors, the user will be informed about this. When a user copies the discord invite link through the edit tournament page, they are notified that the link has been copied to their clipboard, because without this information it would be impossible to see if the button had any function.

3. AFFORDANCE

On each functional component in our web application, there is text to describe its function. This is our way of achieving high affordance. For certain components, we have added symbols to further describe their functions. We also use colors to describe what components do. For delete buttons, we have trash can symbols and red color. For the create tournament, there is a green button with a plus symbol. We have text in addition to symbols on any component with a symbol to avoid unfortunate cultural misunderstandings.

4. MAPPING

Our best example of natural mapping exists on the view tournament page, where each time a winner of a match is selected, it automatically updates the bracket and you can see the changes in real-time. For other changes to tournaments, such as editing tournaments and managing teams, a user is immediately directed to the tournament page where they can see their changes. On the front page, we have clearly placed edit and view tournament buttons in the same visual cards as the tournament they belong to, to distinguish them from edit and view buttons for different tournaments.

5. CONSTRAINTS

Our way of providing the user with constraints is closely linked to what was mentioned in part 1; discoverability. We have avoided cluttering our pages with too many functional components at the same time, to make clear to the user what options they have. On each page except the front page and create tournament page we have a navigation page that greys out the page the user is currently on. This informs the user about which page they are on and which pages they can navigate to while remaining in the same tournament. Each button that leaves a page clearly states so, to avoid leaving pages you don't wish to leave.

6. CONSISTENCY

We have used React components with material styling to ensure that components with similar functionalities have similar styles throughout the application. Buttons for generation or creation are green, while buttons for deleting are red. Neutral buttons that navigate to pages within a tournament are blue. Big, filled buttons consistently represent more significant actions while smaller, outlined buttons represent smaller changes. All tournaments on the front page have the same size and style, no matter what it contains. All teams look the same, both on the team manager page and tournament overview. All input components also have the same styling with titles and placeholders.

7. SIGNIFIERS

Our solution to providing the user with information about where they should perform specific actions, goes hand in hand with our solution for affordance. All buttons have text, and the most significant ones also have symbols. The most important buttons are also bigger and clearly located on a card with a specific team or tournament or at the bottom of a form.

In addition to following Don Norman's seven fundamental design principles, we know that following principle 1 of WCAG 2.1 (perceivable) can significantly improve certain users' experiences. Our steps to follow this principle include, but are not limited to:

- Testing of each page in the web application using Microsoft's Accessibility Insights for Web.
- Sticking to the same colors and color scheme to ensure the contrast ratio stays the same throughout the web application.
- Ensuring all navigation is intuitive and obvious for the user through multiple internal and external user tests.

UD Testing with Microsoft's "Accessibility Insights for Web"

All test summaries can be found in documentation/UD_Tests

First test of web application using Microsoft "Accessibility Insights for Web" – MVP

PURPOSE

- Check that each page meets the requirements for WCAG 2.1 – 1 Perceivable

METHOD

Used the "Accessibility Insights for Web" tool by Microsoft to run automatic, partially-automatic and manual tests on each page.

RESULTS

Tournaments page

- All checks passed

Create tournament page

Issues:

- The "Upload" button on the page did not meet the contrast ratio requirement
  - Color-contrast – Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds
  - Resources: WCAG 1.4.3
- Select-name – The select (dropdown) element did not have an accessible name
  - Ensures select element has an accessible name
  - Recourses: WCAG 1.3.1 And 4.1.2
- All other tests passed

Solutions:

- Changed background color to increase contrast
- Removed select element in favor of a slider

Manage tournaments page

- All checks passed

View tournament page Issue:

- The “Manage teams” button on the page did not meet the contrast ratio requirement
  - Color-contrast – Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds
  - Resources: WCAG 1.4.3

Solution:

- Changed button color to increase contrast

Manage teams page

- All checks passed

Edit teams page Issues:

- Duplicate-id-aria – two elements had the aria id of “teamNameInput”
  - Ensures every id attribute value used in ARIA and in labels is unique
  - Recourses: WCAG 4.1.1
- All other tests passed

Solutions:

- Changed the ARIA id attribute on one of the elements

Second test of web application using Microsoft “Accessibility Insights for Web” – Near-final product

PURPOSE

- Check that each page meets the requirements for WCAG 2.1 – 1 Perceivable

METHOD

Used the “Accessibility Insights for Web” tool by Microsoft to run automatic, partially-automatic and manual tests on each page.

RESULTS

Tournament Overview page

Issues:

- Listitem – some list elements remained from an old revision
  - Ensures `<li>` elements are used semantically
  - WCAG 1.3.1

Solution:

- To ensure a responsive design, all default `<ul>` and `<li>` elements were turned into MUI elements that had has corresponding `<ul>` and `<li>` as its component, to keep the already set up system using `<li>` and `<ul>`. The error was a result of only some elements being changed into these MUI boxes with their corresponding components, and has since been resolved.

Admins page

Issues:

- Aria-required-children
  - Ensures elements with an ARIA role that requires child roles contain them
  - WCAG 1.3.1
- Aria-required-parent
  - Ensures elements with an ARIA role that requires parent roles are contained by them
  - WCAG 1.3.1
- Duplicate-id-aria – multiple elements had the aria id of “rankSelect”
  - Ensures every id attribute value used in ARIA and in labels is unique
  - WCAG 4.1.1

There was not enough time left to properly address these issues

Project Structure

Our tournament manager is a full stack project with a REST-API. The backend is written in javascript, using nodejs and express. The frontend uses react with Material UI.

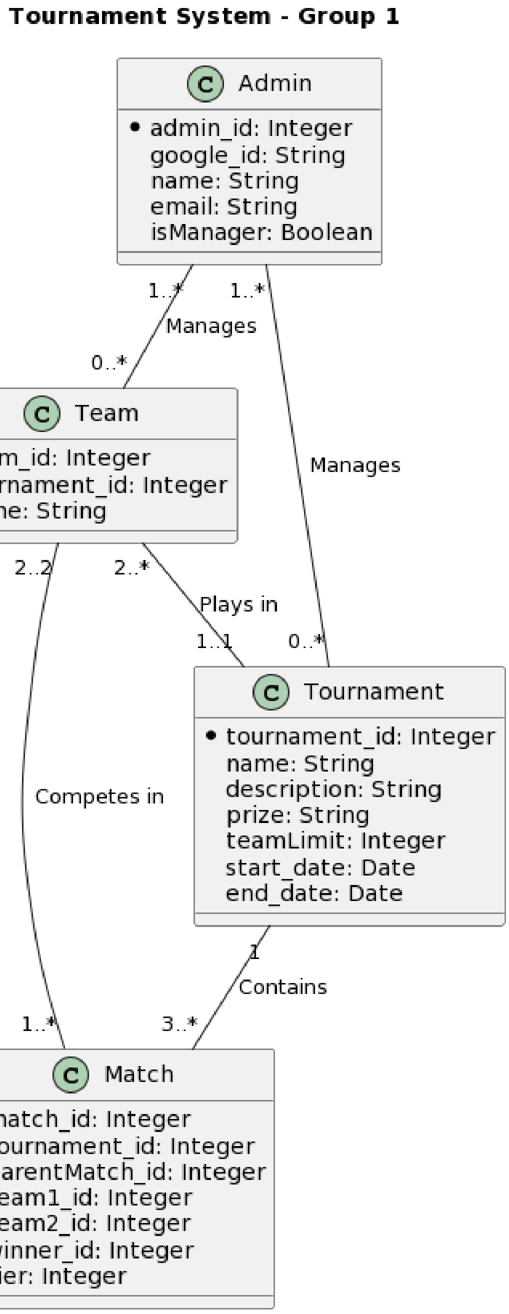
We have chosen these tools to build on knowledge and experiences from our programming courses.

Each web request goes to the server, that requests information from the MySQL database and sends it to the user. When received by the browser, it is presented dynamically on the react page.

The server source code can be found in `src/server` and consists mainly of these two files:

- `index.js` handles web request and user login
- `tmdb.js` handles database interaction

Class Diagram



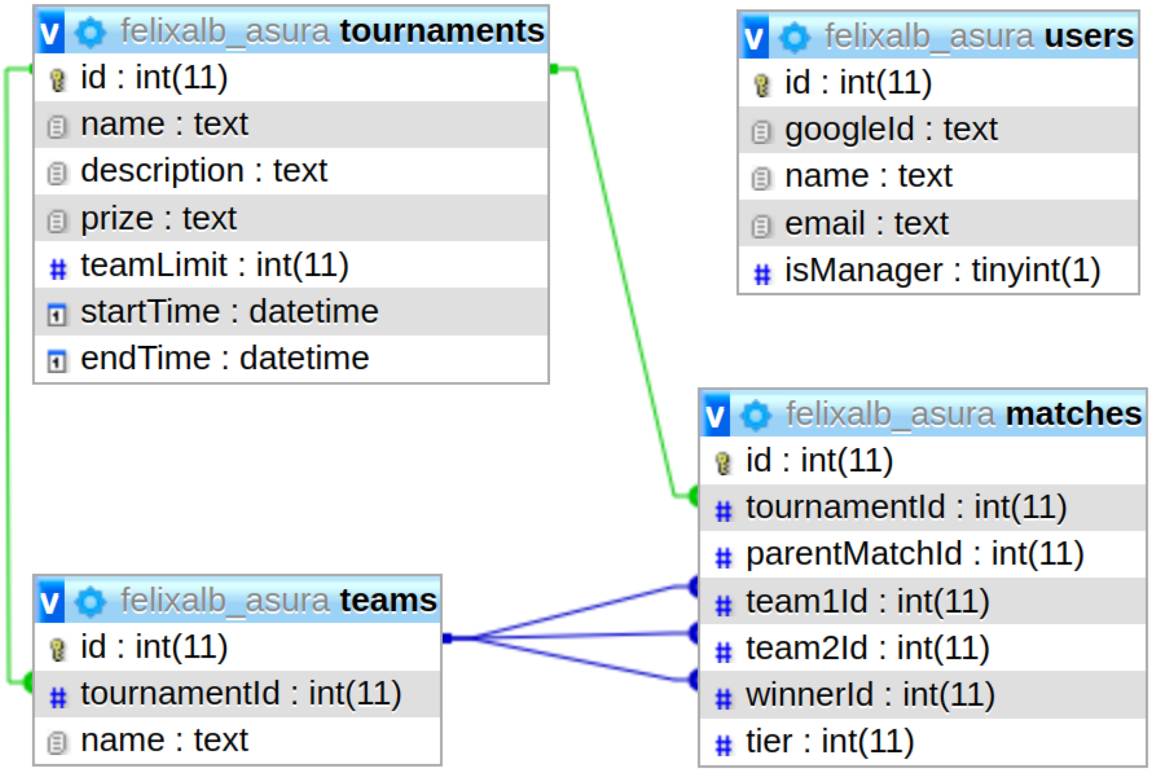
Persistence / Database

All data is stored in the MySQL-database. User sessions are stored in server memory.

- Persistence / Database

```
- [Entity-Relationship / ER-Diagram](#entity-relationship--er-diagram)
  - [The SQL calls to create the database tables needed:](#the-sql-calls-to-create-the-database-tables-needed)
```

ENTITY-RELATIONSHIP / ER-DIAGRAM



felixalb_asura **tournaments**
id : int(11)
name : text
description : text
prize : text
teamLimit : int(11)
startTime : datetime
endTime : datetime

felixalb_asura **users**
id : int(11)
googleId : text
name : text
email : text
isManager : tinyint(1)

felixalb_asura **matches**
id : int(11)
tournamentId : int(11)
parentMatchId : int(11)
team1Id : int(11)
team2Id : int(11)
winnerId : int(11)
tier : int(11)

felixalb_asura **teams**
id : int(11)
tournamentId : int(11)
name : text

THE SQL CALLS TO CREATE THE DATABASE TABLES NEEDED:

```sql
-- WARNING: Will delete EVERYTHING in the database!

DROP TABLE IF EXISTS matches;
DROP TABLE IF EXISTS teams;
DROP TABLE IF EXISTS tournaments;
DROP TABLE IF EXISTS users;

-- Create the tables
CREATE TABLE tournaments (
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
    name TEXT NOT NULL,
    description TEXT,
    prize TEXT,
    teamLimit INTEGER NOT NULL,
    startTime DATETIME NOT NULL,
    endTime DATETIME NOT NULL
);

CREATE TABLE teams (
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
    tournamentId INTEGER NOT NULL,
    name TEXT NOT NULL,

    FOREIGN KEY (tournamentId) REFERENCES tournaments (id) ON DELETE CASCADE
);

CREATE TABLE matches (
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
    tournamentId INTEGER NOT NULL,
    parentMatchId INTEGER,
    team1Id INTEGER,
    team2Id INTEGER,
    winnerId INTEGER,
    tier INTEGER,

    FOREIGN KEY (tournamentId) REFERENCES tournaments (id) ON DELETE CASCADE,
    FOREIGN KEY (team1Id) REFERENCES teams (id) ON DELETE SET NULL,
    FOREIGN KEY (team2Id) REFERENCES teams (id) ON DELETE SET NULL,
    FOREIGN KEY (winnerId) REFERENCES teams (id) ON DELETE SET NULL
);

CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTO_INCREMENT,
    googleId TEXT,
    name TEXT,
    email TEXT NOT NULL,
    isManager BOOLEAN NOT NULL
);
```

Source Code

The project source is developed as two separate components, one in `/src/client/` and one in `src/server/`.

The folder containing both of these can be found in the repository files: https://gitlab.stud.idi.ntnu.no/felixalb/dcst1008-2022-group1/-/tree/main/src

Design choices and underlying theory is described in the final report. Usage and installation are described in their respective manuals.

Installation manual

This is guide describes the process of installing Asura Tournament Manager on your own server. At the time of writing, you can reach our production server at https://asura.feal.no/.

Before installing

Before installing Asura Tournament Manager, you need the following:

- A suitable server to host the software. There are many ways to deploy the server, for example
    - Cloud hosting provider
    - Virtual Machine
    - Dedicated server
- A domain name to reach the server over the internet
- A MySQL database to store persistent data
    - This can be either on the same server as the application, or on a separate database server
- A Google account to sign in to the application
- A Google API key for authenticating users
    - Acquiring this is both simple and free, and is described in the Google API documentation
- [Recommended] A reverse proxy. Software proxies like Nginx will greatly improve the performance, reliability and security of the application.
    - The tournament manager does not handle SSL/HTTPS by itself
    - The server is susceptible to DoS attacks
    - Although the application logs requests to the console, they are not saved on disk
    - Setting up a reverse proxy is not required, but it is *strongly* recommended. Configuration is out of scope of this manual, but it can be found in the Nginx documentation

Download the source

Before you can run the tournament system, you must download the source code. At the time of writing, it is available on NTNU-IDI Gitlab.

Download the entire repository to your machine:

```
git clone git@gitlab.stud.idi.ntnu.no:felixalb/dcst1008-2022-group1.git
```

Configure the client

The client is a web application that runs in the clients browser. After this step is finished, the entire build consists of a single html file, a few javascript files and some static assets.

Install dependencies

```
cd src/client
npm install
```

This step fetches all the required libraries into the node_modules folder.

Configure environment variables

```
cp dotenv-template .env
nano .env
```

Edit the `.env` -file containing a few simple settings for the client, using nano or any other editor. Only the two first lines have to be changed, and just requires the server URL.

Build the client

```
npm run build
```

This step will build the client, optimize it and package it into just a few files. The resulting files are placed in the `build` folder. You are now done with the client and everything else, like serving the build files, are handled by the server.

Configure the server

The server is a node application that both serves the client files and handles the database and tournament logic. The installation process is similar to the client.

Install dependencies

```
cd src/server
npm install
```

Configure environment variables

```
cp dotenv-template .env
nano .env
```

Edit the `.env` -file containing all the server options. This includes:

- The server URL
- The MySQL database credentials
- The Google API credentials
- A cookie secret. This can be any random string of text, as long as it's secret.
- The remaining options should be left as default

Initialize the database

To create the required tables, start with an empty database.

This can be done with a mysql cli utility:

```
mysql -h mysql.stud.ntnu.no -u felixalb_sysut -p felixalb_asura < ./management/initDB.sql
```

Or by pasting the content of `src/server/management/initDB.sql` into another MySQL manager like PHPMyAdmin.

After initializing the empty tables, you will have to insert your own email address. Insert email=youraddress and isManager=1 into the `users` table to register your user. All other users can be added in the graphical user interface.

Check the list of requirements

Read the list of preparations above to see all the requirements. When you have configured everything, you are ready to start the server.

Start the server

When all the steps above have been completed, you can start the server with:

In `src/server` :

```
npm start
```

You should see a message like this:

```
> tournament-server@1.0.0 start
> node index.js

Listening on port 3000
```

Access logs, error messages and other useful messages will be printed to the screen. The database connection is handled automatically, and will reconnect if the connection is lost.

The server can be stopped by pressing Ctrl+C, as nodejs will handle terminate the process cleanly, freeing its resources.

User Manual

Table of Contents

Introduction

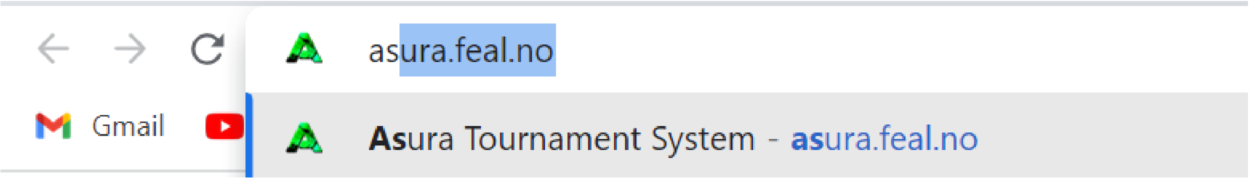
The purpose of the Asura Tournament System is to make it easier for administrators and tournament hosts in the Asura Community to host and manage tournaments. Our system allows administrators to view created tournaments, create new tournaments and edit or manage tournaments. The user interface at asura.feal.no is intended to provide sufficient guidance for understanding how to use the system, however, if there are questions or concerns; see the “Use the Web Application” section of this manual. We hope the manual is clear and informative and thank you for using our product.

User Access Levels

There are three user access levels to the Asura Tournament System. These are users, administrators and managers. Users only have the ability to view tournament brackets and results and are intended to be the participants of the tournament. Administrators are intended to be the tournament hosts and have access to close to the full functionality of the system. They can create, view, manage and edit tournaments. They can also manage teams and players and select winners of tournament matches. Managers have the same access as administrators. In addition, they can create new and remove old administrators.

Using the Web Application

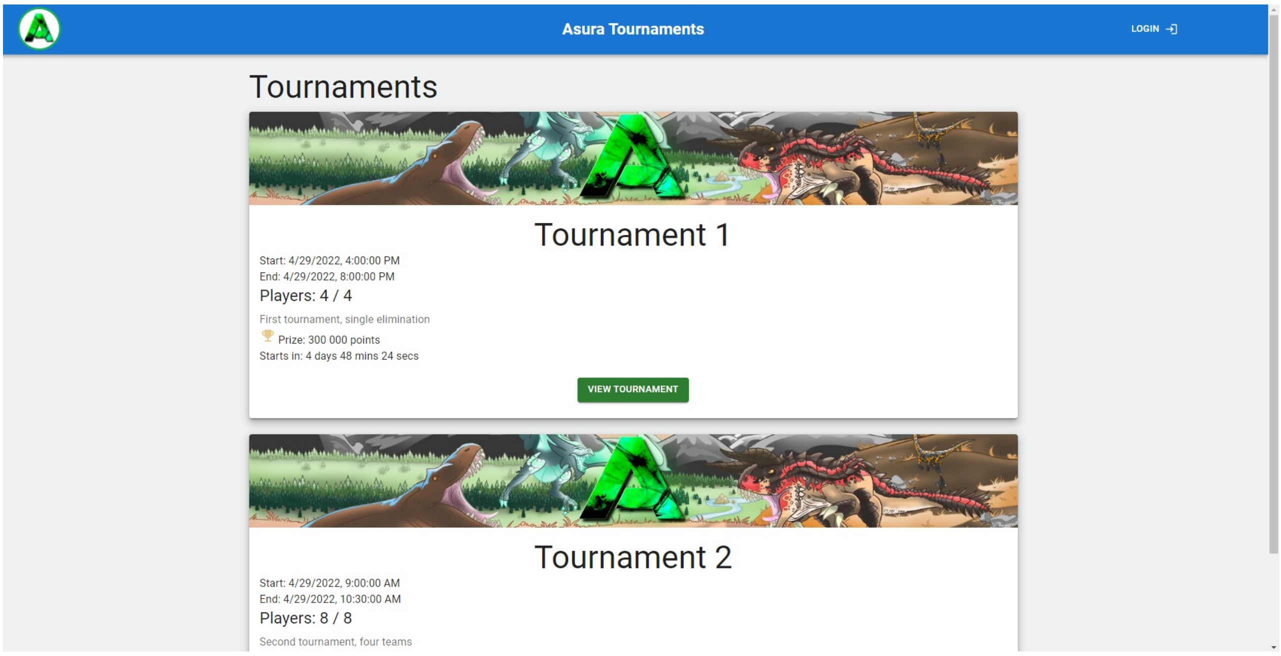
OPENING THE WEB APPLICATION



- Type *asura.feal.no* into your browser and press Enter.
- You should now see the web application

USERS' PERSPECTIVE

NAVIGATING THE WEB APPLICATION



Tournaments

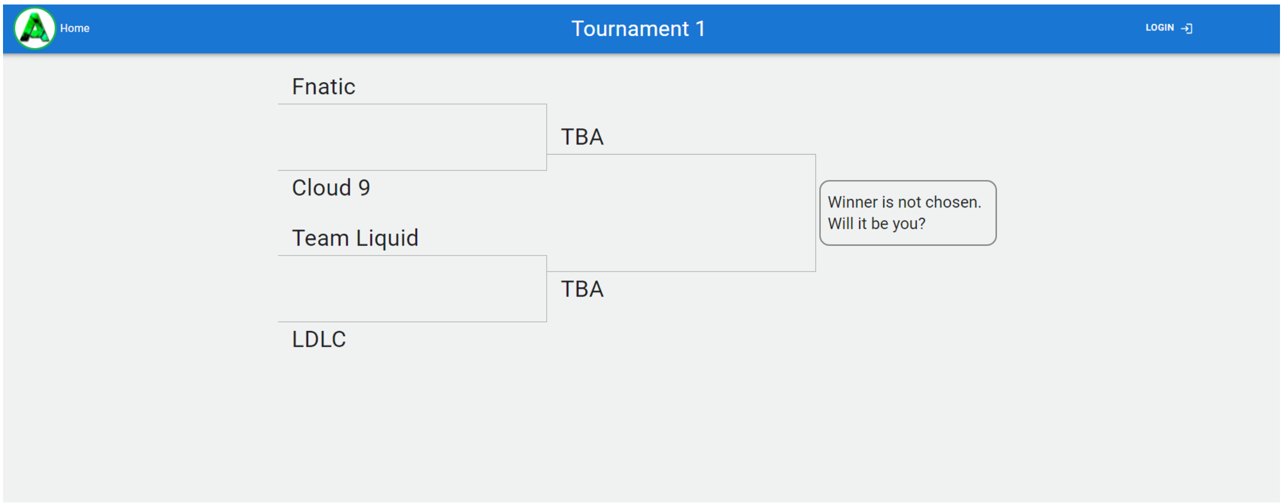
Tournament 1

Start: 4/29/2022, 4:00:00 PM
End: 4/29/2022, 8:00:00 PM
Players: 4 / 4
First tournament, single elimination
Prize: 300 000 points
Starts in: 4 days 48 mins 24 secs

Tournament 2

Start: 4/29/2022, 9:00:00 AM
End: 4/29/2022, 10:30:00 AM
Players: 8 / 8
Second tournament, four teams

- In the top left of the web application, you can always see the asura logo that will, if clicked, take you to the home page.
- In the user view, you can see a list of tournaments down the middle of the page.
- For each tournament you can click the view tournament button at the bottom of each tournament card to view the brackets of the specific tournament.

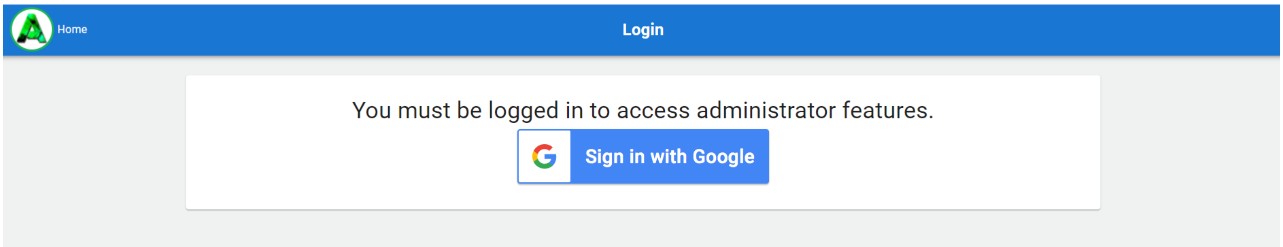


Winner is not chosen. Will it be you?

ADMINISTRATORS' PERSPECTIVE

HOW TO LOG IN

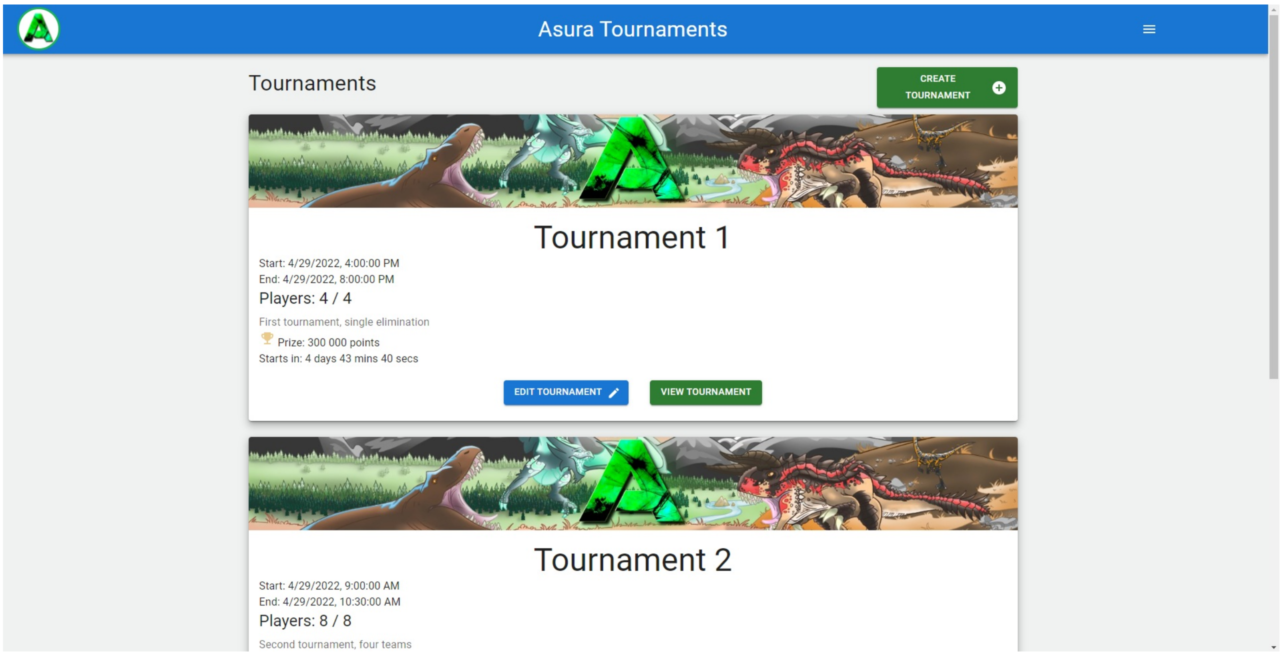
- Click the Login button in the top right of the front page.



You must be logged in to access administrator features.

- Click the Sign in with Google button, and follow the instructions to log in.

NAVIGATING THE WEB APPLICATION



Tournament 1

Start: 4/29/2022, 4:00:00 PM
End: 4/29/2022, 8:00:00 PM
Players: 4 / 4

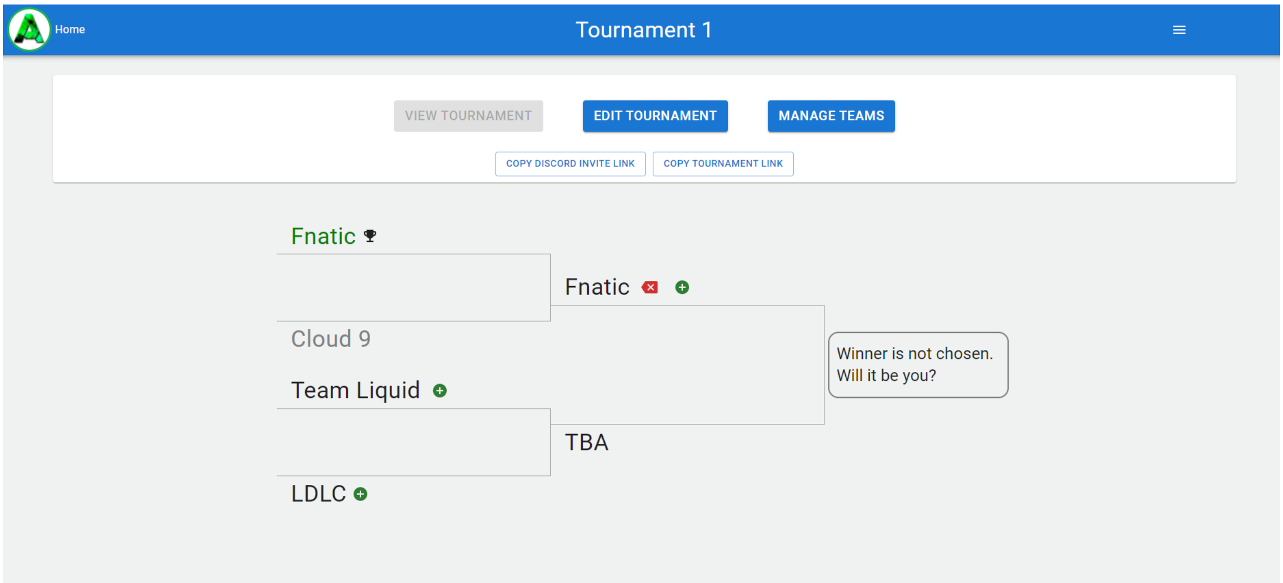
First tournament, single elimination
Prize: 300 000 points
Starts in: 4 days 43 mins 40 secs

Tournament 2

Start: 4/29/2022, 9:00:00 AM
End: 4/29/2022, 10:30:00 AM
Players: 8 / 8

Second tournament, four teams

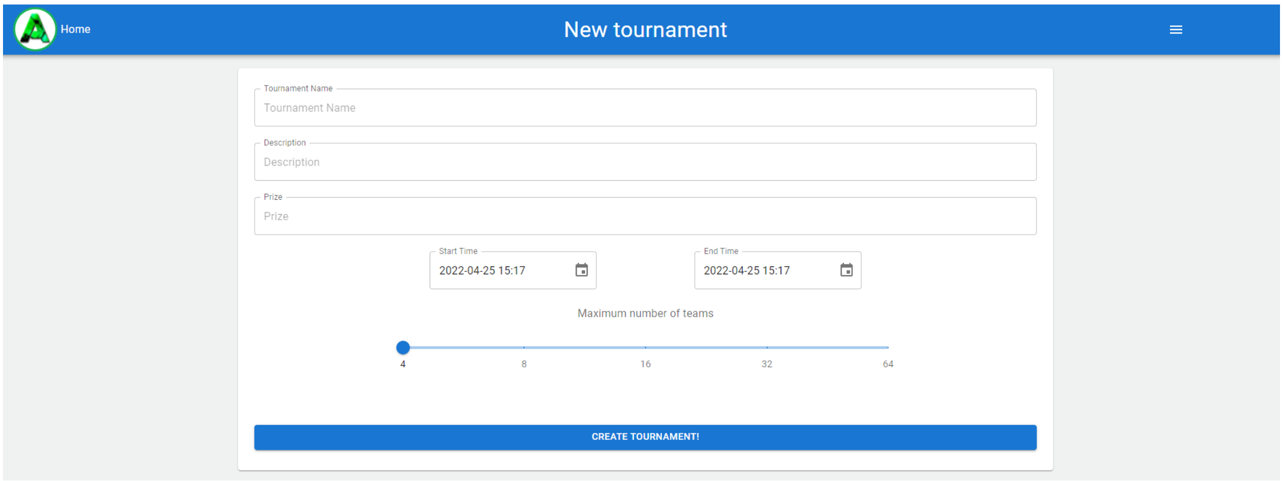
- In the top left of the web application, you can always see the asura logo that will, if clicked, take you to the home page.
- In the top right there is a menu containing options for viewing your profile, tournament history or logging out.
- Below the menu icon there is a large create tournament button for creating tournaments.
- The rest of the page contains a list of current tournaments with an option to edit or view each of them.



Winner is not chosen. Will it be you?

- In addition to navigating to pages from the front page; once you have entered a specific tournament, it is also possible to navigate between pages in the navigation bar near the top of the page.
- In the navigation bar you also have options to copy an invite link to the asura discord, or a link to share the tournament.

HOW TO CREATE A TOURNAMENT



Tournament Name: Tournament Name

Description: Description

Prize: Prize

Start Time: 2022-04-25 15:17

End Time: 2022-04-25 15:17

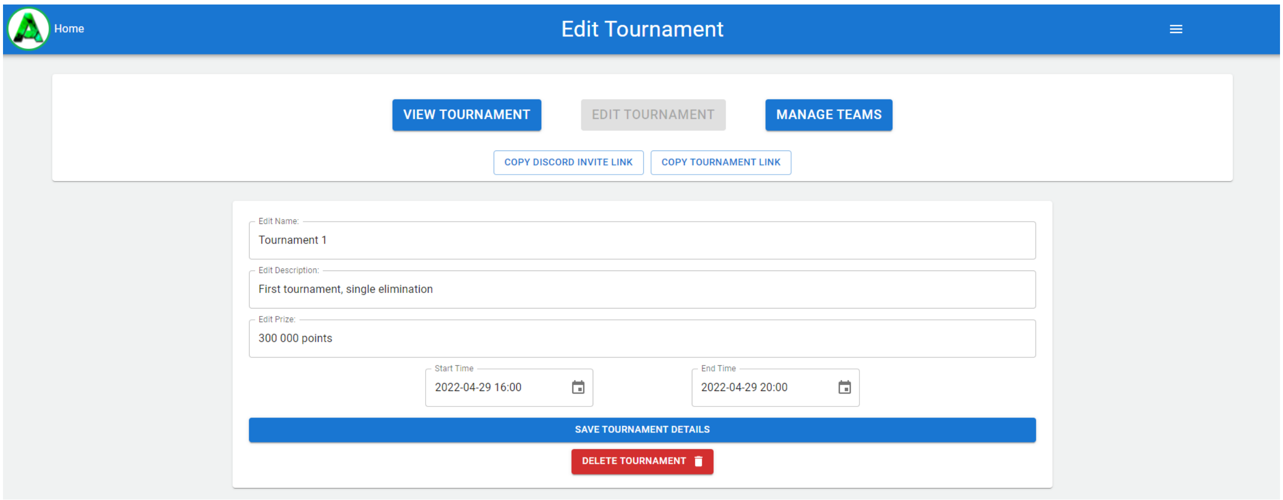
- Begin by clicking the big Create Tournament button on the front page.
- You will then see this window, where you should enter the desired tournament details.
- Fill out the tournament name, description, prize, start and end time and the maximum number of teams.
- All fields must be filled before creating the tournament.
- Once done, click Create Tournament near the bottom of the page and your tournament will be created.

HOW TO VIEW A TOURNAMENT

- On the front page click the view tournament button belonging to the tournament you would like to view.
- If already on the edit tournament or manage teams page of the correct tournament, you can view the tournament through the navigation bar as described in “Navigating the web application”.

HOW TO EDIT TOURNAMENT DETAILS

- On the front page click the edit tournament button belonging to the tournament you would like to edit.



Edit Name: Tournament 1

Edit Description: First tournament, single elimination

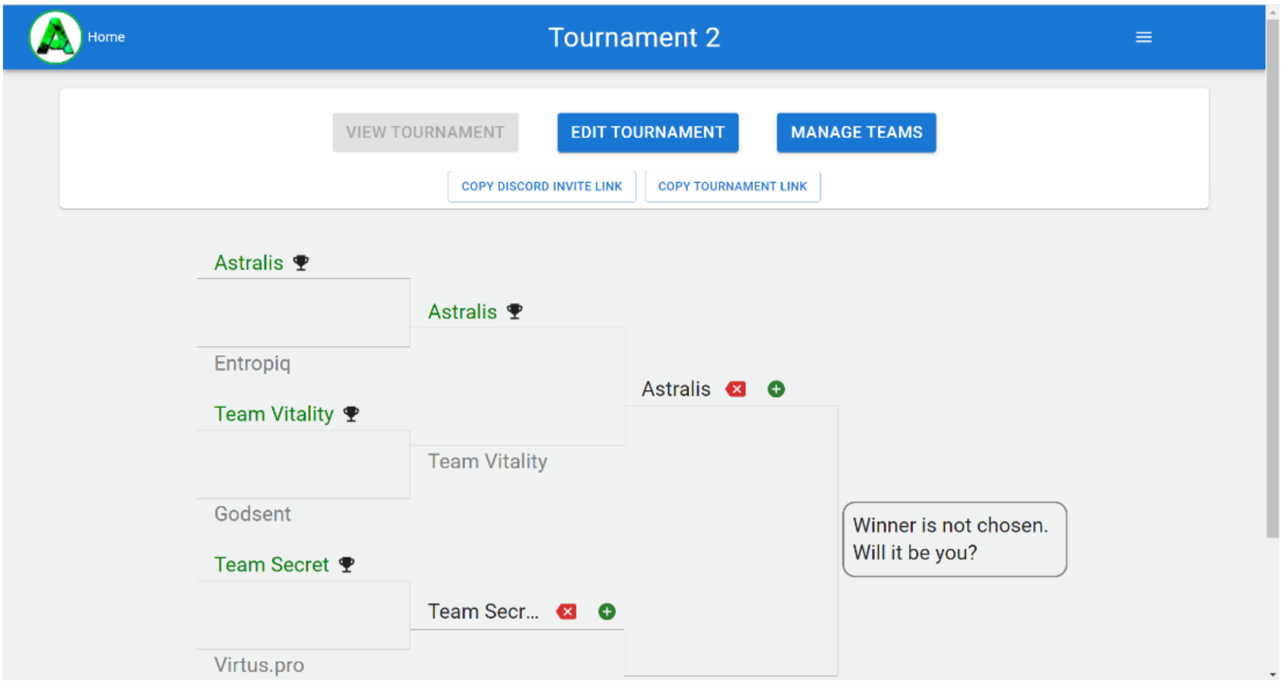
Edit Prize: 300 000 points

Start Time: 2022-04-29 16:00

End Time: 2022-04-29 20:00

- As you can see the current tournament information is already entered in the edit tournament page.
- You can edit as much or little of this as you want, and then click the Save Tournament Details button below the filled-out form.
- To delete a tournament, click the Delete Tournament button at the bottom of the page.

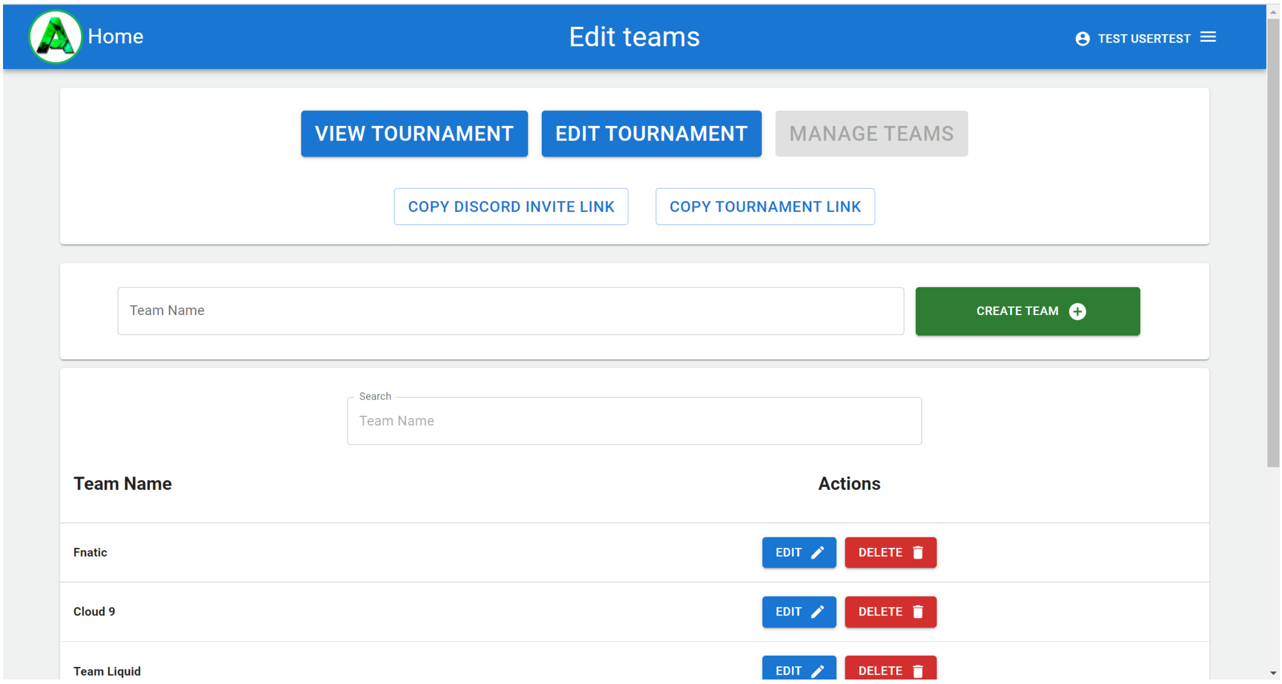
HOW TO MANAGE THE TOURNAMENT BRACKET



Winner is not chosen. Will it be you?

- Enter the View Tournament Page per the instructions in “How to view a tournament”.
- To select a team as the winner of a match, click the green plus next to the team's name.
- If a mistake has been made, click the red arrow next to a team's name to demote them to the previous stage of the bracket.

HOW TO MANAGE TEAMS

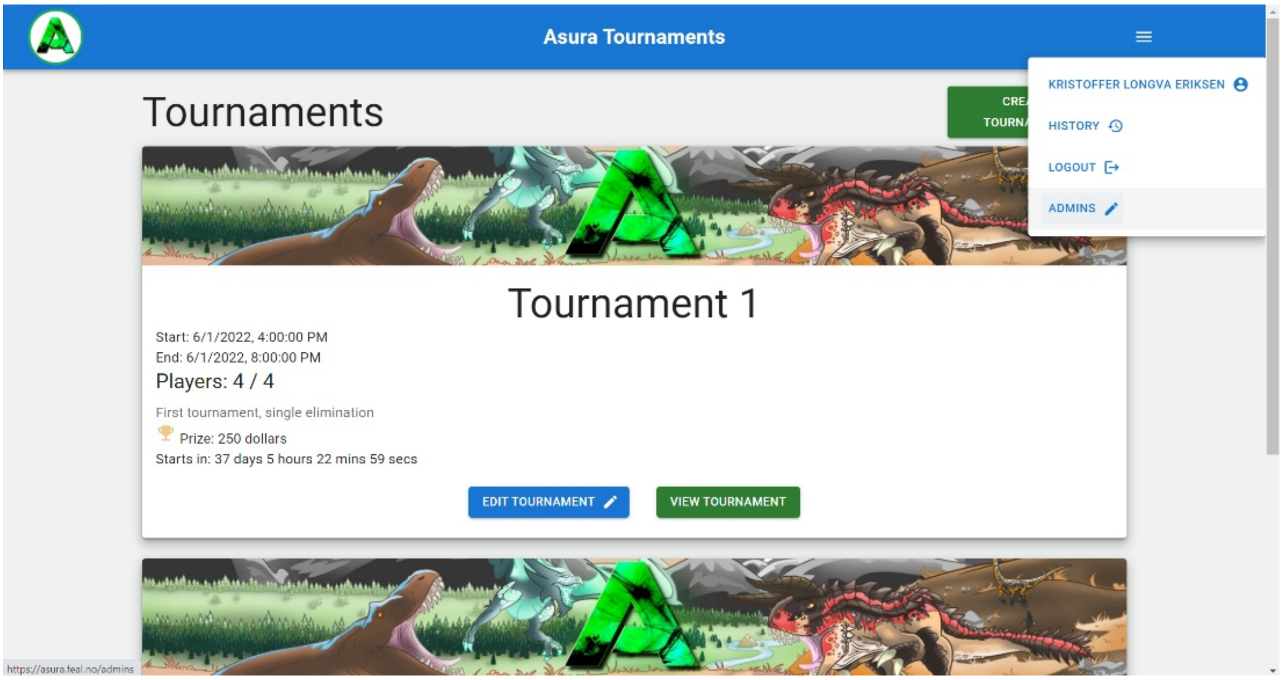


Team Name	Actions
Fnatic	EDIT DELETE
Cloud 9	EDIT DELETE
Team Liquid	EDIT DELETE

- On the manage teams page, you can create teams by filling in the Team Name field and clicking the Create Team button, if the maximum number of teams has not already been reached.
- To delete a team, click the Delete button next to the team you want to delete.
- To edit a team, click the Edit button next to the team you want to edit.
- Once a team has been selected for editing, an edit field will appear at the bottom of the page, where you can edit and save a new name.

MANAGERS’ PERSPECTIVE

HOW TO ADD OR REMOVE ADMINISTRATORS

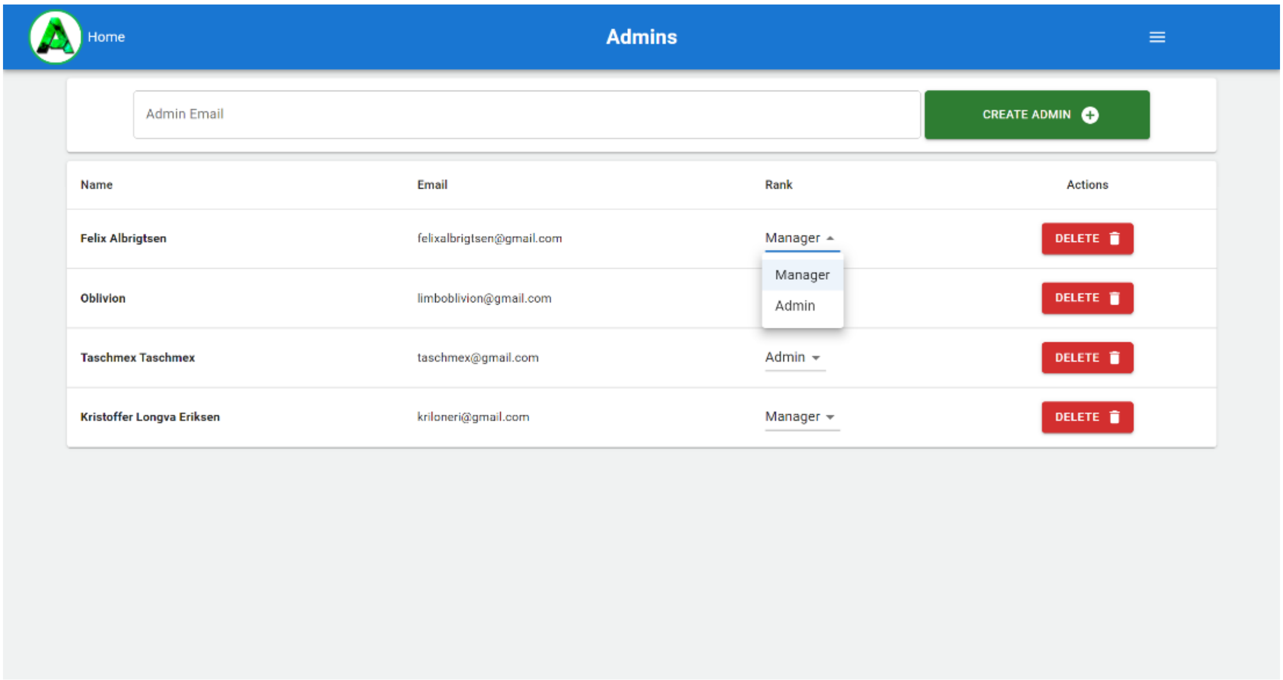


Tournaments

Tournament 1

Start: 6/1/2022, 4:00:00 PM
End: 6/1/2022, 8:00:00 PM
Players: 4 / 4
First tournament, single elimination
Prize: 250 dollars
Starts in: 37 days 5 hours 22 mins 59 secs

- Click the menu in the top right of any page and click on Admins.



Name	Email	Rank	Actions
Felix Albrigtsen	felixalbrigtsen@gmail.com	Manager	DELETE
Oblivion	limboblivion@gmail.com		DELETE
Taschmex Taschmex	taschmex@gmail.com	Admin	DELETE
Kristoffer Longva Eriksen	kriloneri@gmail.com	Manager	DELETE

- Here you can add new admins by adding their Gmail address in the Admin Email field and clicking the Create Admin button.
- You can delete admins/managers by clicking the Delete button next to the admin you want to delete.
- You can change a person's rank by using the drop-down menu next to their name, as shown above.

CONCLUSION

This manual should provide a clear overview of how to use the Asura Tournament System to host and manage tournaments in an easier way. The web application is tested in Google Chrome, if you have any issues, switching to Chrome could resolve them. We thank you for choosing our product and hope it works well for you. If there are questions or concerns past this manual, contact Oblivion#4483 on Discord.

Vision Document

Vision in PDF form

Group 1

Asura Tournament System

Vision

Version 1.0

Revision History

Date	Version	Description	Author
06/03/22	0.1	Preliminary Draft	Felix Albrigtsen, Jonas Jødestøl Haugland, Kristoffer Juelsen, Kristoffer Longva Eriksen
16/03/22	0.2	Changes according to supervisor meeting	Jonas Jødestøl Haugland, Kristoffer Juelsen
18/03/2022	1.0	Final Draft	Felix Albrigtsen, Jonas Jødestøl Haugland, Kristoffer Juelsen

Table of Contents

Vision

1 **Introduction**

This document provides and describes the requirements and features of the "Asura Tournament System", which is the project for the course "DCST1008 – Systemutvikling". The assignment consists of developing a tournament system for a specified stakeholder. The assignment is well defined and is presented as a task to create a management system for multi-team tournaments, where the primary user should be the tournament administrator.

The Introduction includes relevant information that will be used and referenced throughout the document.

1.1 Purpose and scope

The purpose of the document is to define these needs and features, as well as focusing on the capabilities required by the stakeholders and target users, including why these exist. Further explanations of each feature and design goal is described in their corresponding sections in the document.

The scope of the document includes only the specifications for the tournament system previously described, and does not include any other projects, there are no external dependencies. Any supplemental information not directly regarding the project will be clearly marked

1.2 Definitions, Acronyms, and Abbreviations

The Isle – The Isle is the game that will be played in our tournaments, and is intended to be a gritty, open-world survival horror game. It is a game where you go through the life cycles of a Dinosaur and grow up, fight, hunt and survive. Upon dying you need to start over, making it quite competitive for some people to become the best.

Asura – A European based gaming community that hosts game servers for The Isle, as well as other games, and regularly hosts tournaments and events for its members. This group will serve as our customer.

UD – Universal Design

1.3 References

[1] WCAG 2.1

This is the standard which we will base our development on, to include Universal Design and accommodate for anyone who wishes to use or explore our products features. https://www.w3.org/TR/WCAG21/

Don Norman's principles of Interaction design

https://www.educative.io/edpresso/what-are-normans-design-principles

1.4 Overview

The rest of the Vision document is organized in different sections where the information within those sections mostly correlate to each other, but also is used to define features for the rest of the document, and some sections are elaborated later in the document. The first section includes positioning in regard to business opportunities and different statements. The second section describes the project goals, and the next thereafter includes stakeholder and user descriptions. The rest of the document includes Product Overview, Product Features, Constraints, Quality Ranges, Precedence and Priority as well as other product requirements, such as standards. Finally, it includes documentation requirements. The document is organized in these sections to provide an easier reading experience and to be able to quickly navigate through the document.

2 Positioning

2.1 Business Opportunity

In the ever-increasing landscape of competitions and tournaments, there is a growing need for a simple and effective system to handle many users without the need for end users to worry about the infrastructure and organization of any tournament they would want to host. With this project we wish to decrease the workload put on hosts, and let the administrators focus on the game rather than the technology.

2.2 Problem Statement

The problem of	**Easily hosting tournaments without the need for much manual labor**
affects	The Asura gaming community
the impact of which	Creates lots of extra work for management and moderators that wish to host a tournament or event
a successful solution would be	Easily managed Have a fast-paced setup and completion Accessible and easily implemented

2.3 Product Position Statement

For	**The management of a large, international, gaming community that hosts events and tournaments.**
Who	Wishes to more easily create online tournaments and lessen the workload for the organizers
The (product name)	Is a web application that can be accessed by the specified user
That	Provides a good and accessible solution that is upheld to international standards and implemented in such a way that anyone would be able to utilize the product given the right documentation and manuals.
Unlike	"Challonge", another tournament hosting website
Our product	Should be accessible to anyone who needs to use the system and correspond to international standards regarding Universal Design, as well as being straight-forward and simple to use in addition to having smart and useful features.

3 Project goals

3.1 Impact goals

- Ensure readability and maintainability of our code and project
  - To keep our project readable, we will use descriptive Git commit messages and peer review to ensure good quality. For our code specifically we will use descriptive variable names that are easy to understand, as well as commenting our code where necessary.
- Reduce the efforts needed to host a tournament
  - We will measure this by measuring the time it takes for the customer currently and aiming for a set amount of time needed to have a successful setup of a tournament that is shorter than the current time.
- Improve the working environment when hosting these events by developing for helpful and creative uses
  - Not a thing easily measured, but by getting feedback from the customer and reviewing the success of the tournaments hosted we will secure these goals are met. By also adhering to standards and not developing for our own winnings, but rather the improvement of the community we provide the service for, we will ensure these goals are met.

3.2 Result goals

- Hand in deliverables on time
- Develop a good product that meets the customers' expectations and needs
- Create a product that follows necessary standards such as UD, WCAG 2.1[1] and provides accessibility for anyone who needs to use the system

3.3 Process goals

- Learn the process of developing a complete system
- Improve communication with other students and lectures for guidance and better development in the future
- Pass the subject

4 Stakeholder and User Descriptions.

4.1 Market Demographics

The organization is one of the biggest communities and providers of game servers in the market that it focuses on. It has a good reputation and is the only server that provides the unique style of gameplay in the game servers it hosts of its size. With well over 30,000 users that is an active part as well as being an ever-increasing place of gathering, the organization is one of the lead places in its category. With an active player base and moderation team, it is in good shape and is ever growing.

We would like the market demographic to continue growing as the customer expands and extends its services to other platforms and markets. It is already a very large market in comparison to other potential smaller, local customers, so we are happy with servicing the established environment.

By creating this project, we can increase the member retention and also member engagement, resulting in increased activity and growth. This would also in some cases lead to an increase in turnover for the customer.

4.2 Stakeholder Summary

Name	Description	Responsibilities
Community Management	End user	Add and manage Community Moderators Approve tournaments and manage tournaments Organize and create tournaments
Community Members	Will not be able to manage the system, only view the tournaments Is the group of people who will participate in the tournaments	None
Community Moderators	The people who primarily organize the tournaments with approval from management End user	Organize and create tournaments
Course Lecturer	End user	View the final product and guide the production through iterations
Teaching Assistant	End user	View the product during production and comment on potential improvements
Project Developers	End user and developer of product	Create and manage the product Responsible for maintenance and further development Communication with the customer, lecturers, and teaching assistants

4.3 User Summary

Name	Description	Responsibilities	Stakeholder
Community Managers	Leaders of the customer organizations	Manage tournaments Manage tournament organizers Manage archive	Self-represented
Community Moderators	Moderators of the community	Organize tournaments Who this is can change and it is a varied group of people	Community Managers
Community Members	Guests	View tournaments if they wish	Self-represented
System Managers	The creators of the project	In charge of maintenance Fix bugs and errors that might display themselves down the line	Self-represented

4.4 User Environment

Currently, there are four people who are considered managers, and 22 moderators actively pursuing the improvement of the community. The number of people in either of these positions change because of people retiring, or being hired, so it is not a set value. There is one owner, who should have some superior level of control regarding certain cases.

There are no set tasks, as most of the work is voluntary coming from the moderators. The things that have a set schedule are the events (tournaments), but the time slot of these vary depending on the organizer and content.

This is an online operation, so no environmental constraints will be liable, other than some moderators or users being unavailable due to local environmental problems. Current platforms are three different games, two of which are located on steam, the other on an individual platform. One of these games is a plan for future development but is close to being implemented.

The only other application in use that could be directly utilized by us as system developers is Discord and integrating this with either Discord to login or to be able to format messages to be sent to the communities' channels. This, however, is not a priority for us and will only be implemented by necessity.

4.5 Key Stakeholder or User Needs

Need	Priority	Concerns	Current Solution	Proposed Solutions
Create a Tournament	High		None	Web application UI
Manage Tournaments	High	Cooperation, multiple admins	None	Web application UI
View Tournaments & standings	High		None	Web application UI
Complete tournament and announce a winner	Medium		None	Web application and Discord
Archive tournaments	Low		None	Export summary
User notifications	Medium	Integration with discord	None	Discord or Email
Login system for organizers	High	Security, privacy		Web interface with cookies/browser storage

4.6 Alternatives and Competition

4.6.1 Challonge

Challonge is an established easy-to-use web application that is free to use for anyone.

Its major strengths are that it is already in use by the customer as it is a simple solution and straightforward in most cases.

A weakness it has is that it is not built for the customer specifically, but rather built for anyone who would wish to host a tournament, and therefore must provide a lot of additional elements rather than focusing on one specific customer and their needs.

5 Product Overview

5.1 Product Perspective

The product aims to be a targeted alternative to the bigger, broader, tournament hosting websites like Challonge. The larger websites are accommodated to host tournaments for a lot of different games which makes them bloated for users wanting to host infrequent tournaments for a single game.

The product is a simple website aimed towards simple navigation and needing fewer clicks to create the initial tournament as well as managing them. It is a self-contained system and consists of the client interface (the Web GUI), a server running the application, and a database server which holds the team and player information.

5.2 Summary of Capabilities

The tournament management system we produce should enable all administrators in our customer group to create and manage tournament plans for their given games. The back-end of the application will host the database storing all tournament data, but this should not be directly visible to the user. Administrators should have a simple web interface where they can create, edit, and delete tournaments. Players, not just administrators, should also have access to a web link where they can see the tournament status. This ensures that the system is accessible to everyone without requiring any installation or additional tools. The management pages should be fault-tolerant and handle any conceivable input from the user and must be resilient to user mistakes. Normal usage of the application should be well documented and explained in the supplied user manual.

5.3 Assumptions and Dependencies

The tournament system is naturally based on the type of game being played. In the specified game, every match exists between exactly two teams or players. If this were to change, for example by requiring a match between three different teams, our requirements and build process would have to adapt.

As previously stated, our product will be hosted on a web server. That means that all usage requires a web browser and an internet connection. Without access to the internet, the user will not be able to send and receive data between their computer and the web server. The server application will be hosted on one of our private servers but require little in terms of software and hardware performance. Any computer with a good internet connection and the ability to run Node.JS will suffice.

5.4 Risk analysis

Here we have listed possible events and analyzed the probability and consequence for each.

- Project delays
  - The project could at any time become delayed due to illness or other unforeseen events.
  - **Aversion:**
    - Team communication: The entire team is kept updated, and workloads can be shifted to other team members if someone is unable to participate, for example when ill.
- Late delivery or delivery of unfinished product
  - The project could be delivered late or in an unfinished state due to underestimation of work hours needed to finish the product.
  - **Aversion** :
    - Active use of the Gantt-diagram and GitLab-board ensures that we stay with the pre-defined plan. By keeping eye on these diagrams, deviations can be easily spotted and corrected.
- Loss of work
  - Program code or documentation that has not yet been committed or saved otherwise may be lost due to a power outage, application crash or other unforeseen events.
  - **Aversion:**
    - All team members will follow a policy of working directly in the git repository, with separate commits for each feature or issue.
    - We will keep daily backups of our git repository on a personal server.
    - Text documents and similar work not included in our git will be stored in Microsoft OneDrive.

Server downtime

- The server hosting the tournament system may experience internal or external issues regarding uptime and availability, making the product become temporarily unavailable.

Aversion:

- Configuration files will be backed up in git and locally on our personal computers.
- The NTNU-IDI MySQL database seems stable, but we can easily change to any other MySQL instance.
- The installation manual will be devised for easy reinstallation if we must.

What	Probability	Consequence	Risk factor
Project delay	3	2	6
Late delivery	2	5	10
Delivery of unfinished product	2	5	10
Loss of work	1	4	4
Server downtime	1	2	2

Probability and consequence values are given on a scale of 1-5. Risk factor is probability times consequence.

5.5 Installation

Because we use a cloud-like model where no work is being done on the user's computer, no special installation concerns are required. All users are only required to have a web browser and an internet connection.

The actual software will be built in JavaScript on top of the node.js and React frameworks. These tools will be required on the server for building and running the project. The basic mechanics will build on tools and techniques used in our earlier programming courses.

6 Product Features

6.1 Intuitive web interface

The software should be accessible through a normal web browser. The user will be able to interact with all components of the system by using a mouse and keyboard in the browser. By complying with established guidelines for universal design and clean web development, the menus should be intuitive, simple, and understandable. This minimizes installation time and costs, training time and allows the software to be used by people with a disability.

6.2 User management

Our customer has two levels of administrator access, "Administrators" and "Management". Users in the management group should be able to access and modify all tournaments in the system, as well as creating or removing administrators.

6.3 Easy Collaboration

As opposed to smaller, simpler, tournament management systems, our product will allow multiple administrators to manage the same tournament at once. This will be important to our customers, as the administration consists of many different people in all parts of the world. Relying on single people might be a bad idea, and this system will still be fully functional if one or more administrators disappear or fail to fulfill their tasks, as a manager or other administrator can take over the project.

6.4 Automatic Bracketing

To create a new tournament bracket, an administrator can simply create an empty tournament and insert the competing teams. When the list of teams is saved, pairs of teams will be combined randomly into matches. This makes the process of creating new tournaments both simple and fair. If the administrator is not happy with the distribution, they can either manually edit the order, or randomize again.

6.5 Scheduling

Each tournament is created with a duration and starting time and date. When teams are invited, they can receive reminders about upcoming tournaments ahead of time. This can improve the workflow of the administrators, as they can prepare the tournament in advance so the games can start exactly when scheduled, without having to wait for configuration.

7 Constraints

The project assignment constraints how much time we are able to spend on the assignment itself, as well as on what technologies that are allowed to be utilized. Knowledge could also be a constraint in some cases, as well as the need for expertise to create elements of the project.

8 Quality Ranges

We have not defined specific performance requirements in terms of numbers, but usability and a feeling of responsiveness is highly prioritized. We will not accept excessive delays and hiccups in the finished product, and the software should leave a professional impression on the user.

The system should be able to handle our expected workload with an excessive margin. This is a part of robustness, which also includes proper error handling when something goes wrong at runtime. By writing efficient back-end code with little resource overhead, and by logging important actions, we can ensure that the system is robust.

Usability is an important aspect of modern web development. By sticking to established guidelines concerning Universal Design and availability, we will ensure that the tournament system is easy to use. This also includes the page structure being accessible to screen readers, and that the visual elements can be distinguished clearly.

9 Precedence and Priority

Here we have listed different system features and prioritized them. | **Feature** | **Priority** | **Precedence** | |-------------|--------------|----------------| | Intuitive Web Interface | 3 | Intuitive Web Interface is the most important feature of the program, and it precedes any other, because unless the interface is intuitive there is little reason to improve on other areas. | | User Management | 3 | User management should be one of the core features of the program to ensure easy access and usage, | | Automatic Bracketing | 2 | Automatic Bracketing is a Quality-of-Life improvement which reduces the amount of effort and clicking needed by the end user to set up a bracket. With just simple inputs of names and teams, the system will create a bracket on its own without the need of an end user moving slots around and manually assigning each slot with a name. | | Easy Collaboration | 1 | Easy Collaboration by multiple administrators is a "nice to have" feature, but it is more important to improve the workflow of each administrator rather than allowing multiple administrators to enter the workflow. Therefore, it is preceded by scheduling. | | Scheduling | 1 | Scheduling of when and where precedes easy collaboration, as its more important to have a functioning system where everyone is able to view the correct info, rather than having multiple people being able to edit and manage a single tournament. |

Priority values are given on a scale of 1-3. Precedence compares the features with the same priority and ranks them accordingly within the priority range.

10 Other Product Requirements

10.1 Applicable Standards

WCAG 2.1[1]

The system should work on any platform, including Windows and *NIX-based Operating Systems

TCP/IP standard is the basis for our web communication

10.2 System Requirements

The server needs to be able to run Node to present the web application.

The user needs a computer with internet access to access the application.

10.3 Performance Requirements

The tournament system should be a fairly simple application without any specific hardware requirements. Although the system may be simple, we might expect several hundred web requests in a short period of time if all contestants are loading the dashboard to check the status of their tournaments. We will not set any hard limits on performance, like measuring response times, but we will ensure that the page is perceived as responsive and stable. The user should not have to wait for pages to load or actions to occur, but these should be integrated in a seamless web interface with little latency. Downtime is unacceptable, and the page should be perceived as smooth and professional.

11 Documentation Requirements.

11.1 User Manual

The user manual includes a basic guide on how to use the system and will also include an email address to contact the system developers if the need arises. It should be relatively short, no more than two A4 pages so that it can be printed on a single, double-sided A4 paper if necessary.

11.2 Internal Documentation

To aid in team collaboration, planning, maintenance and documentation for the subject assignment, all code, models, and internal structures must be documented and presented on our wiki page hosted on GitLab. Source code should be written as clearly, readable, and concisely as possible, and be commented inline where appropriate. Documentation will be done in parallel with development, and not afterwards.

11.3 Installation Guides, Configuration, and Readme File

The product will be hosted by us, the developers, and maintained by us in a web-based environment, so there is no need for any installation guides for this specific project. Should be need arise for others to host this environment themselves installation guides can be procured.