
Software Engineering Software Requirements Specification (SRS) Document

Knocked-Out

9/20/2022

<Final Version>

By: Matthew Mitchell, Senih Tosun, Joshua Hinojo

All work is subject to the UNCG Academic Integrity policy: <https://osrr.uncg.edu/academic-integrity/>. Academic dishonesty includes submitting for credit any software that you (or your team, in case of a team project) did not write yourself/yourselves.

Table of Contents

1. Introduction	
1. General Description (Matthew)	3
2. Functional Requirements (Joshua)	3
3. Technical Requirements (Senih)	4
4. Non-Functional Requirements (Matthew)	4
Software Architecture (Senih, Matthew)	6
Software Design (Matthew, Joshua)	7
Use Case Model (Senih)	11
Final Presentation Scenarios (Senih, Matthew, Joshua)	13

1. Introduction

1.1 Purpose: Our purpose is to provide a comprehensive set of services to fulfill our customer's needs in all steps of their tournament creation process.

1.2 Document conventions:

The purpose of this Software Requirements Document (SRD) is to describe the participant-view, organizer-view, and administrator-view requirements for the Tournament Organization System (TOS). Participant-oriented requirements describe the system from the naive-user participants perspective. These requirements include a description of the participant served by the system. Organizer-oriented requirements describe the system from a higher-level end-user perspective. These requirements include a detailed description of the of the organizer served by the system. Administrator-oriented requirements describe the system from a software developer's perspective. These requirements include a detailed description of data, performance, functional, and other pertinent requirements.

1.3 Definitions, Acronyms, and Abbreviations

Include any specialized terminology dictated by the application area or the product area. For example:

Term	Definition. Acronym, Abbreviation
.Net	A set of software technologies from Microsoft for connection information, people, and computer systems.
TOS	An abbreviation for Tournament Organization System. This is the name of the system that is being built.
C#	A programming language created by Microsoft. We will be using this language to build the ATPS.
DB	An abbreviation for Database.
MS	An abbreviation for Microsoft. Microsoft is a large software company which produces the software that will be used to implement ATPS.
Microsoft Access	A database software created by Microsoft. The campus police vehicle violation database was created using Microsoft Access.
SRD	An abbreviation for Software Requirements Document. This is the name of the document.

1.4 Intended audience: The introduction and general description in this SRS informs the users about what we are hoping to accomplish through this project, giving a general overview of the things they can do with our application. The technical, functional, and non-functional requirements are intended for our development team to have specific ideas about what exactly we need to build for the users.

1.5 Project Scope: The software goals are to provide a fully functional tournament organization tool set to our end-users. The overall business goal is to gain a customer base that can be leveraged for both paid and non-paid sponsors.

1.6 Technology Challenges: One of the main technology challenges that we came across so while building the application was implementing the relationship between the organizer and tournaments and adding teams inside the tournaments.

1.7 References:

Alred, F., Brusaw, C., and Oliu, W. (2003). *Handbook of Technical Writing* (7th ed.). Boston: Bedford/St. Martin's.

2. General Description

2.1 Product perspective: Tournament Organization System (TOS) that guides users to build and maintain complex tournaments for any competitive event.

2.2 Product features: The TOS handles participant sign-up, organizer tools for setting up different types of tournament styles while also providing a choice between random seeding and statistical seeding for participants, an administrator is necessary for verifying and maintaining the integrity of participants and organizers through moderation.

2.3 User class and characteristics: Our website application does not expect our participant users to have any prior knowledge of a computer, apart from using a web browser, or knowledge of tournament organization. Our website application has removed the need for them to have tournament knowledge and allows the user to focus on competing in a tournament. However, organizer users are required to have knowledge of which tournament style best suits their needs and/or limitations such as time, venue size, etc. Implementation of a particular tournament is handled entirely by the system. Administrators must have the same amount of knowledge as the organizer to ensure the integrity of all users.

2.4 Operating environment: Software for TOS is designed to operate in any environment that supports web browser access.

2.5 Constraints: To limit user error when entering the participant's information fields, we implemented a drop-down AJAX selection.

2.6 Assumptions and dependencies: It is assumed users have access to a web browser and an internet connection. TOS include external dependencies Spring Boot and (insert api).

3. Functional Requirements

3.1 Primary

- FR0: The system will allow the user to lookup the participants associated team and relevant data associated with that team.
- FR1: The system will allow the user to enter a new team into a specific tournament being organized.
- FR2: The system will allow the user to issue a tournament seeding. The seeding will be derived from team statistics or randomly generated.
- FR3: The system will allow the user to update the bracket once the tournament has started.
- FR4: The system will allow the user to delete a team before or after the tournament has started.
- FR5: The system will keep the user's team information and the server's tournament statistic database synchronized to within 24 hours.

4. Technical Requirements

4.1 Operating System & Compatibility 4.2 Interface requirements

4.2.1 User Interfaces

The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.

4.2.2 Hardware Interfaces

All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.

4.2.3 Communications Interfaces

Determination of all the communication standards to be utilized by the software as a part of the project

4.2.4 Software Interfaces

The interaction of the software to be developed with other software components such as frontend and the backend framework to be used, the database management system and libraries describing the need and the purpose behind each of them.

5. Non-Functional Requirements

5.1 Performance requirements

- NFR0(R): The novice user will be able to create a team and join a tournament in less than 5 minutes
- NFR1(R): The expert user will be able to generate seeding in less than 5 minutes.
- NFR2(R): The expert user will be able to generate tournament brackets in less than 1 minute. -
- NFR3(R): The novice user will be able to delete a team in less than 5 minutes.
- NFR4(R): The local copy and server-side TOS database will consume less than 20 MB of memory

5.2 Safety requirements N/A

5.3 Security requirements

The system will only be usable by authorized users.

5.4 Software quality attributes

- 5.5. Availability
- 5.6. Correctness
- 5.7. Maintainability
- 5.8. Reusability
- 5.9. Portability

5.5 Process Requirements

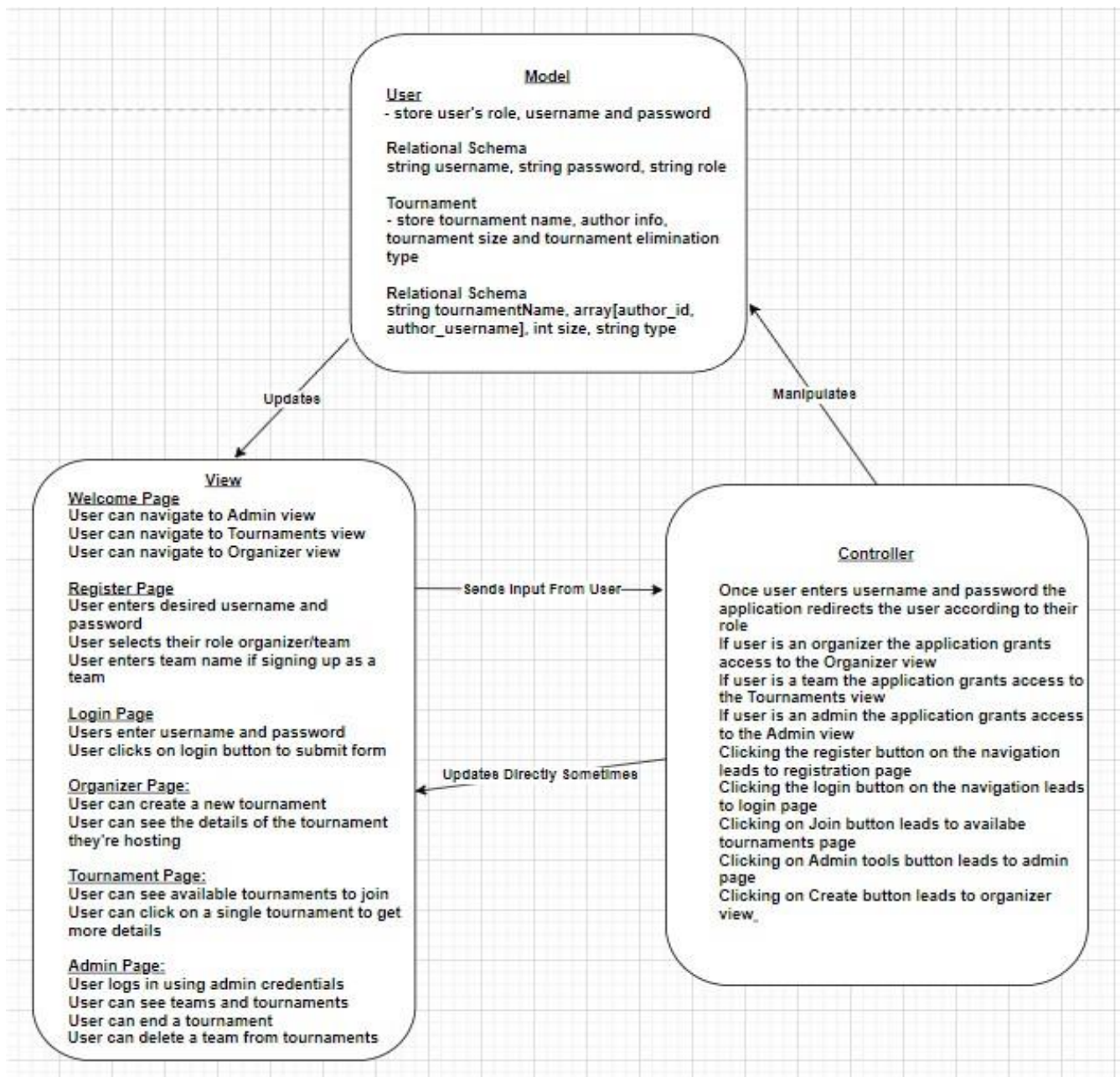
- 5.5.1. Development Process Used
- 5.5.2. Time Constraints
- 5.5.3. Cost and Delivery Date

5.6 Other requirements

All SRS/SRD should be:

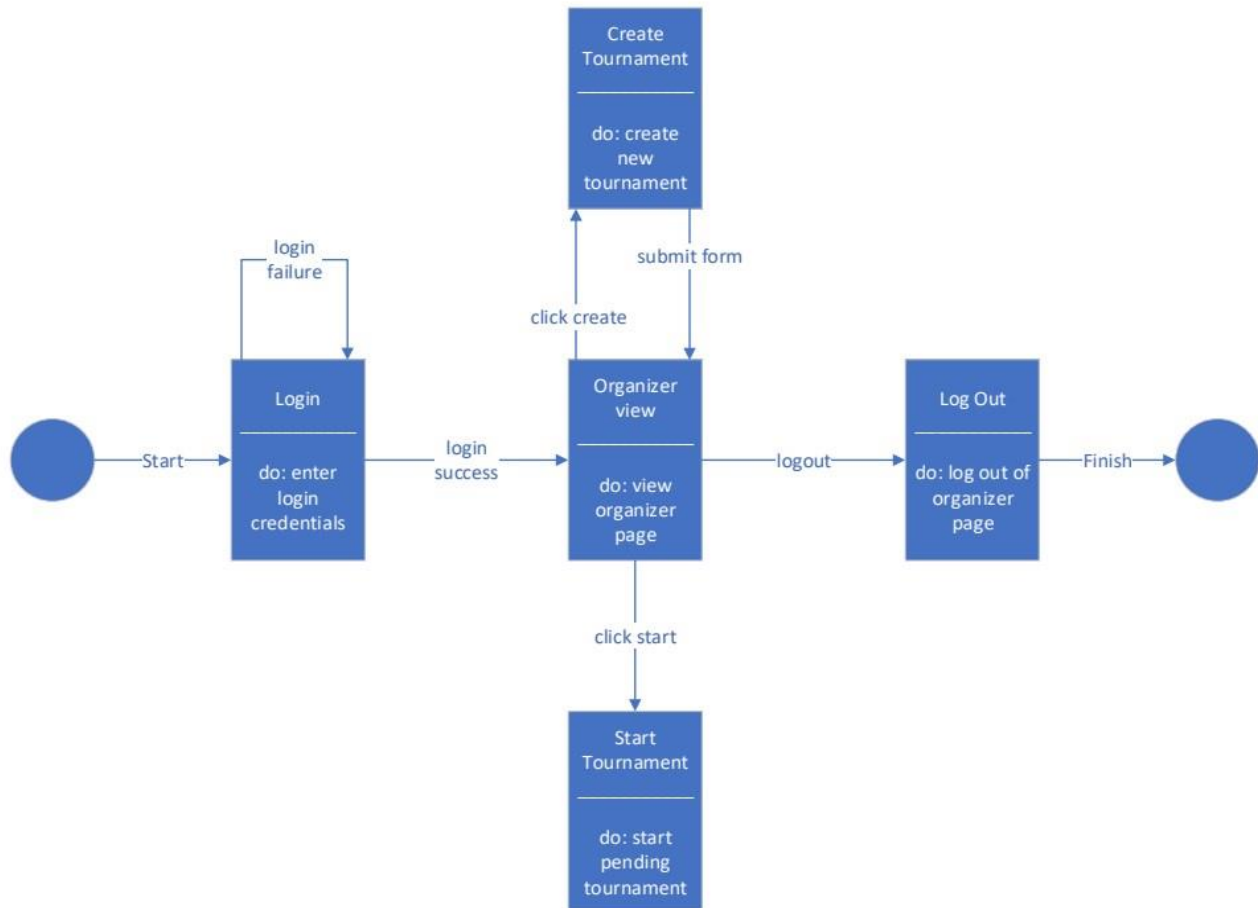
- **Correct:** A method of analysis that ensures that the software meets the requirements identified. Unambiguous: There is only one interpretation of what the software will be used for, and it is communicated in a common language.
- **Complete:** There is a representation for all requirements for functionality, performance, design constraints, attributes, or external interfaces.
- **Consistent:** Must agree with other documentation, including a systems requirements specification and other documents.
- **Ranked** for Importance and/or Stability: Since all requirements are not of equal weight, you should employ a method to appropriately rank requirements.
- **Verifiable:** Use measurable elements and defined terminology to avoid ambiguity.
- **Modifiable:** A well-defined organizational structure of the SRS document that avoids redundancies can allow easy adaptation.
- **Traceable:** Ability to trace back to the origin of development and move forward to the documents produced from the SRS.
- **Legible and Professionally Presented:** Must use a consistent font and style. Must have proper formatting of tables and charts. Must be grammatically correct. Use active tense and concise sentences.

6. Software Architecture

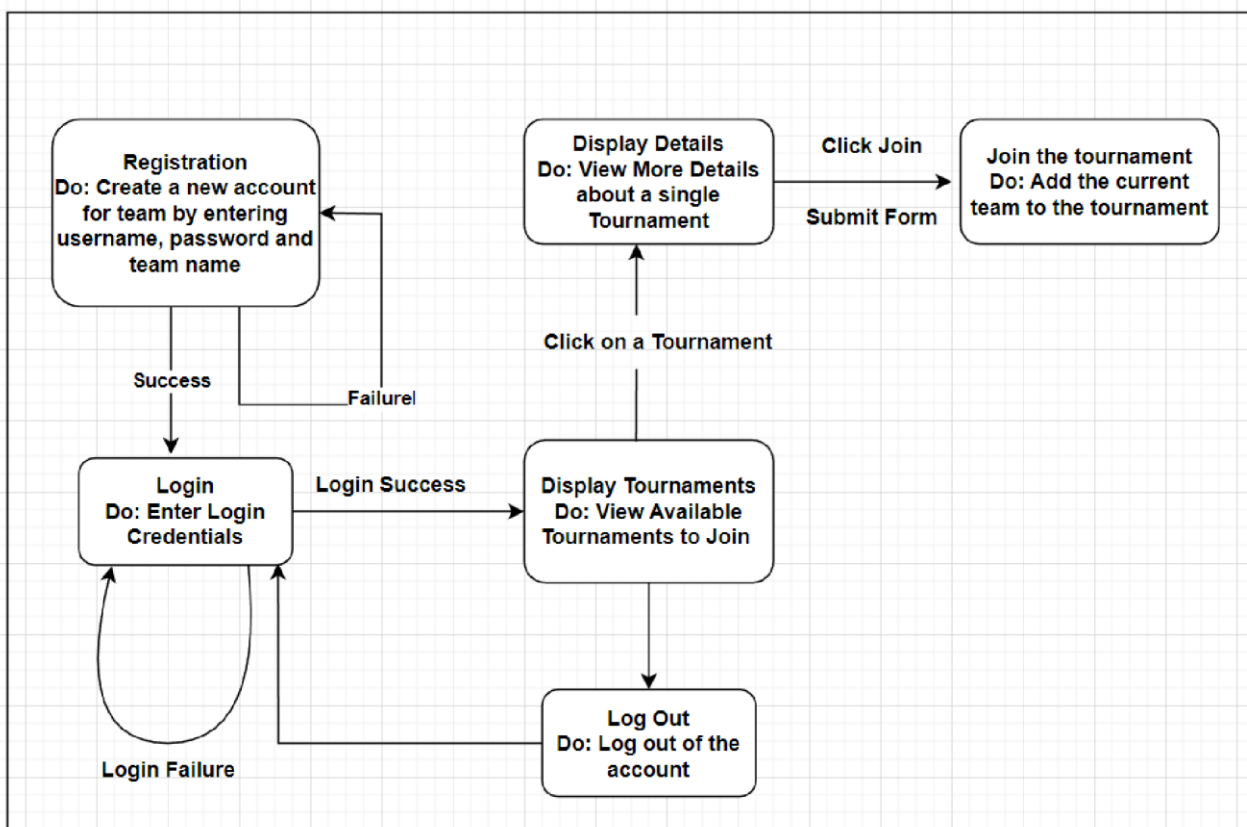


7. Software Design

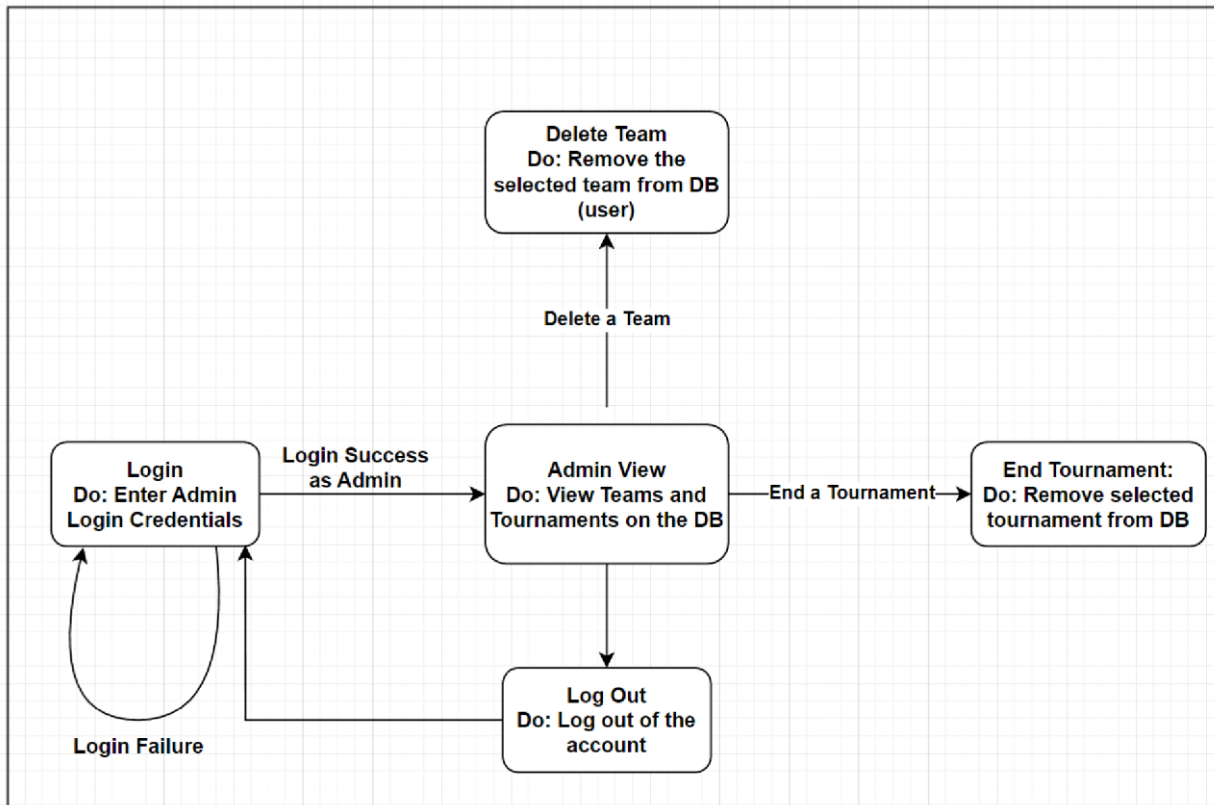
Organizer State Diagram:



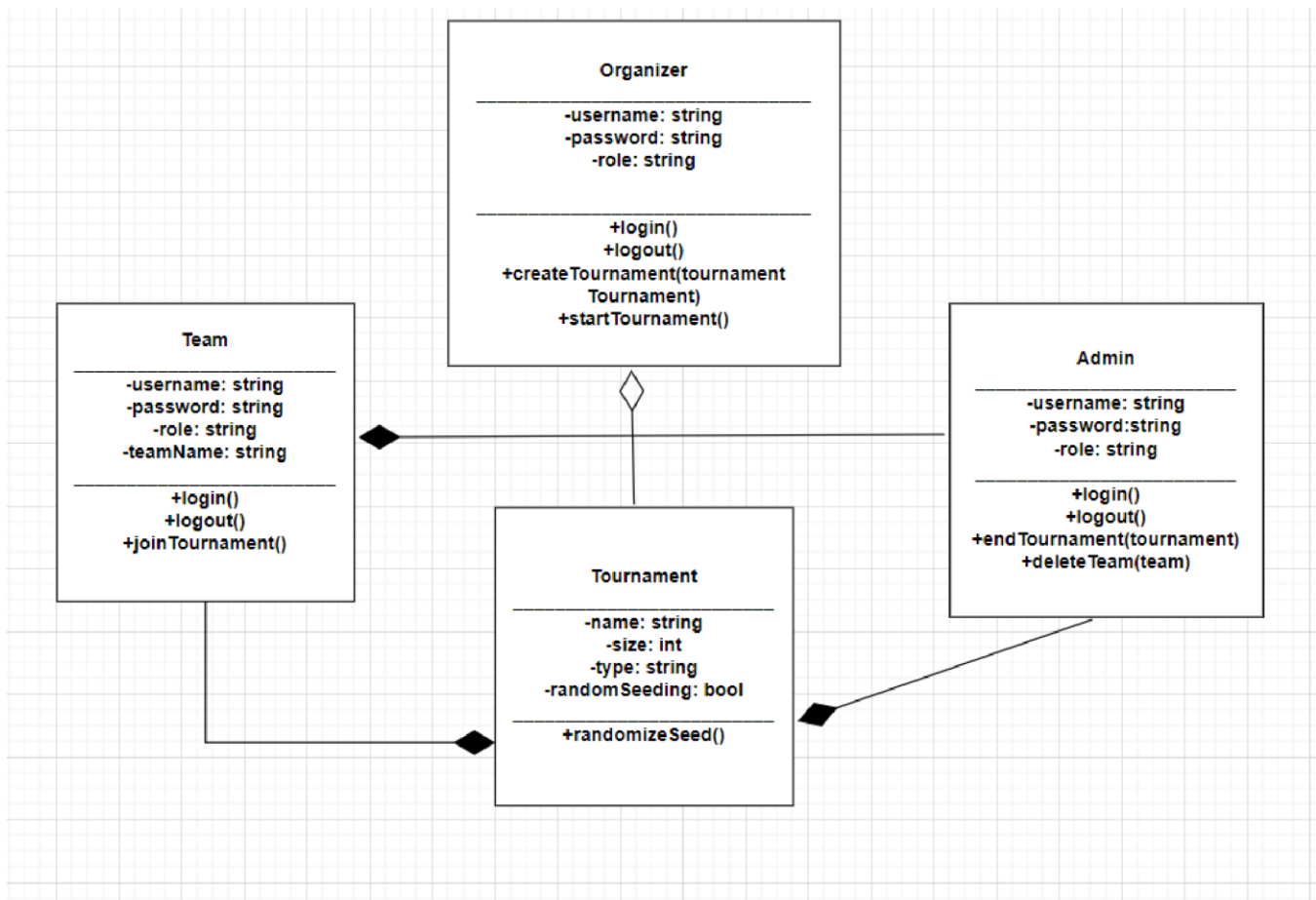
Team State Diagram:



Admin State Diagram

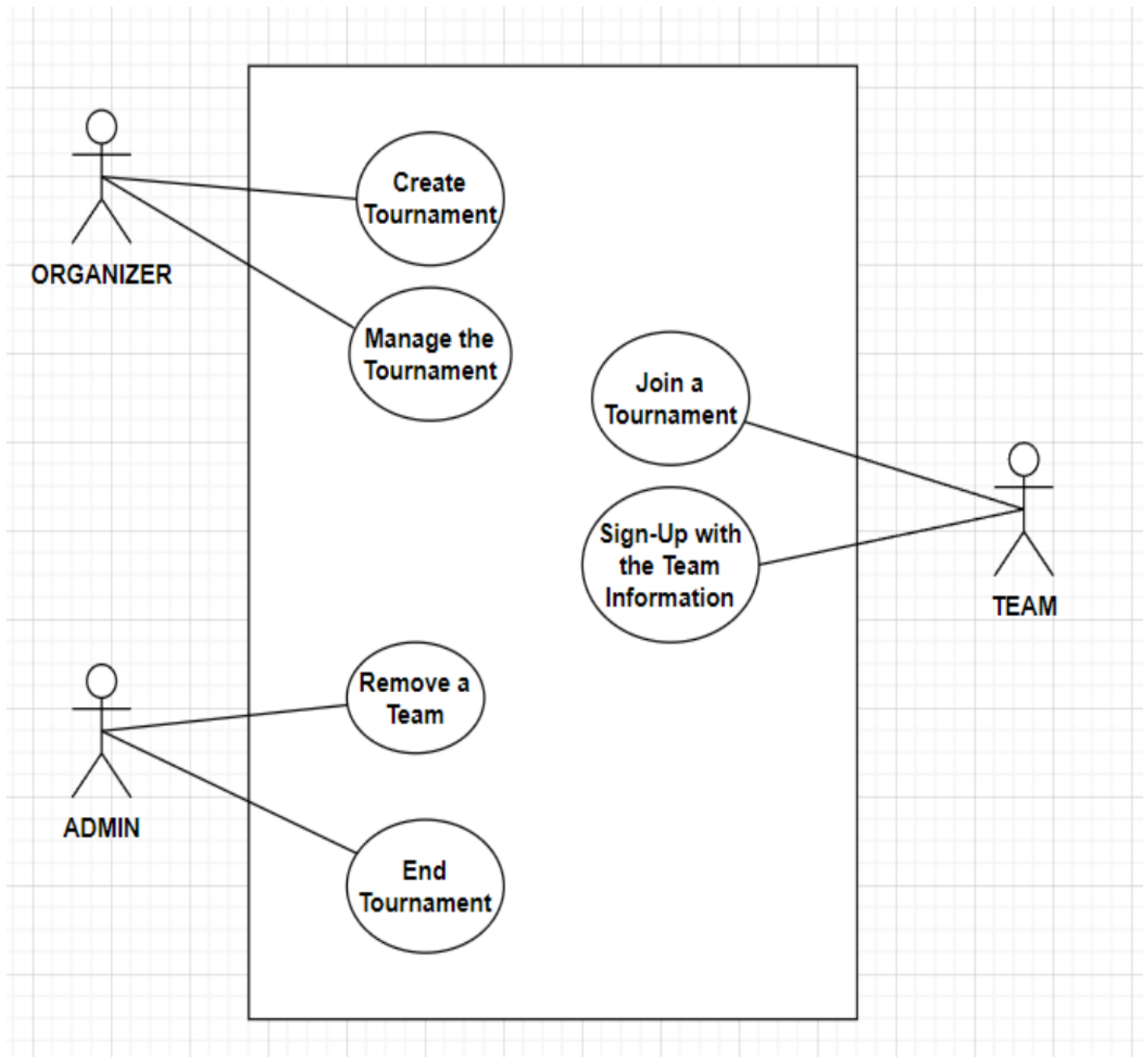


UML Diagrams



8. Appendix

USE CASE MODEL



Section I: Brief Description

Sign-Up:

User can sign-up as a team or organizer

Create Tournament:

Organizer can create a tournament upon signing up.

Join Tournament:

Teams can join any available tournaments upon signing up.

Delete a Team:

Admin can delete any teams who have cheated or violated any other rule.

End a Tournament:

Admin can end a tournament, which will remove any data associated with that tournament from the database as well.

Section II: Scenarios

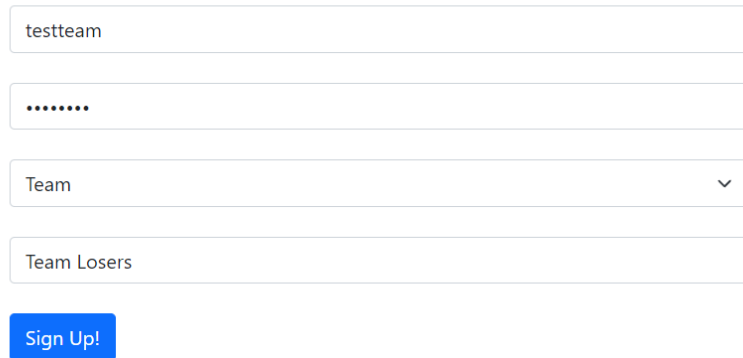
- **Sign up: (Senih)**
 - **Initial Assumption:** The tournament has been generated by the organizer and is open for sign-up.
 - **Normal:** The participant adds their team to the tournament and is automatically populated into the correct position according to the seeding.
 - **What can go wrong:** The participant adds incorrect information regarding their team which must be changed.
 - **Other activities:** Request that their team be removed by an administrator.
 - **System state on completion:** The team has been added to the system and will be displayed in the bracket.
- **Delete Team: (Joshua)**
 - **Initial Assumption:** The tournament has begun or officially ended.
 - **Normal:** The administrator deletes a team from the tournament bracket.
 - **What can go wrong:** The administrator deletes the wrong team but has been warned by the system that the changes cannot be reverted.
 - **Other activities:** View all teams currently participating in the tournament.
 - **System state on completion:** The tournament will continue where it left off without the team that was deleted.
- **Create Tournament (Matthew)**
 - **Initial Assumption:** User is logged in as an organizer.
 - **Normal:** Tournament gets created and the data is saved on the database
 - **What can go wrong:** Some fields are not properly sent to the database or user enters the same name.
 - **Other activities:** View all the previously created tournaments by the same organizer.
 - **System state on completion:** A new tournament will be created and added to the list of available tournaments for the teams to join.
- **Join a Tournament (Senih)**
 - **Initial Assumption:** User is logged in as a team
 - **Normal:** The team joins the tournament, and the database gets updated with the information.
 - **What can go wrong:** The team is already in another tournament which would prevent them from joining the current one.
 - **Other activities:** View the information on the tournaments they have joined and other teams
 - **System state on completion:** The team will join the tournament, the database gets updated, and the team will see the tournament details they have signed up.

FINAL PRESENTATION SCENARIOS

<TEAM>

Team: Sign-up as a team through the registration page. Enter the details such as username, password, and team name.

Sign Up



A sign-up form with four input fields and a button. The first field contains 'testteam'. The second field contains seven dots. The third field is a dropdown menu with 'Team' selected. The fourth field contains 'Team Losers'. Below the fields is a blue button labeled 'Sign Up!'.

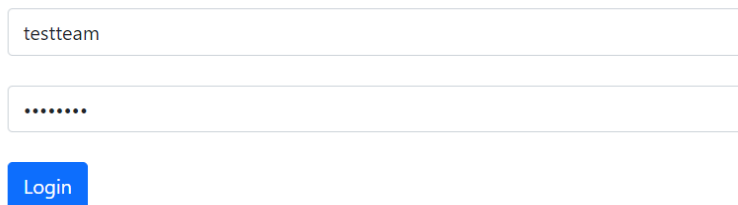
Team ▼

Already have an account? [Log In](#)

© Copyright 2022 Knocked Out

Team: Login using the credentials such as username and password.

Login



A login form with two input fields and a button. The first field contains 'testteam'. The second field contains seven dots. Below the fields is a blue button labeled 'Login'.

Not a member yet? [Sign Up](#)

© Copyright 2022 Knocked Out

Team: Join a tournament, teams can view available tournaments and join them.

Tournaments You Can Join

Refresh

#	Name	Size	Elimination Type	
1	Poker Tournament	4	single	Learn More
2	Programming Contest	4	double	Learn More

© Copyright 2022 Knocked Out

Knocked Out

Logout

Name: Poker Tournament

Size: 4

Elimination Type: single

Join

Join Tournament

Cancel

Join

#	Team Name
1	test team
2	test team 2

© Copyright 2022 Knocked Out

<ORGANIZER>

Organizer: Organizers sign up through the registration page as an organizer, entering the necessary details such as username and password and selecting their role.

Sign Up

Organizer



Sign Up!

Already have an account? [Log In](#)

© Copyright 2022 Knocked Out

Organizer: Organizers login using their credentials (username and password)

Login

Login

Not a member yet? [Sign Up](#)

© Copyright 2022 Knocked Out

Organizer: Organizers create tournaments through the create tournament route by filling the form with the necessary data, such as tournament name, tournament size, and elimination type.

The screenshot shows the 'Knocked Out' application interface. A modal titled 'Create Tournament' is open, allowing users to create a new tournament. The modal contains the following fields and options:

- Tournament Name:** A text input field with the value 'Tournament 340'.
- Size:** A dropdown menu with the value '32'.
- Type:** A dropdown menu with the value 'Single Elimination'.
- Random Seeding:** An unchecked checkbox.
- Buttons:** 'Cancel' and 'Create Tournament' buttons at the bottom right.

The background of the application shows a header with 'Knocked Out' and a 'Logout' link. The main content area has a background image of a wrench and a circuit board. The footer of the application shows '© Copyright 2022 Knocked Out'.

Organizer: Organizers can see the tournaments they have created and manage them through that page.

The screenshot shows the 'Knocked Out' application interface. The header has 'Knocked Out' and a 'Logout' link. The main content area is titled 'Your Tournaments' and displays a table of tournaments created by the user.

#	Name	Size	Elimination Type	
1	Tournament Test	4	single	<button>View</button>

The footer of the application shows '© Copyright 2022 Knocked Out'.

<ADMIN>

Admin: Admin will login through the same login page using the special admin credentials, for security purposes, there is no way a user can sign-up as an admin.

Login

Login

Not a member yet? [Sign Up](#)

© Copyright 2022 Knocked Out

Admin: The admin can view a list of current tournaments.

Knocked Out

Logout

Tournaments

Teams

Search...

Poker Tournament	View	End
Programming Contest	View	End
Tournament Test	View	End

© Copyright 2022 Knocked Out

Admin: The admin can view a list of the teams who signed up.

Knocked Out

Logout

Tournaments

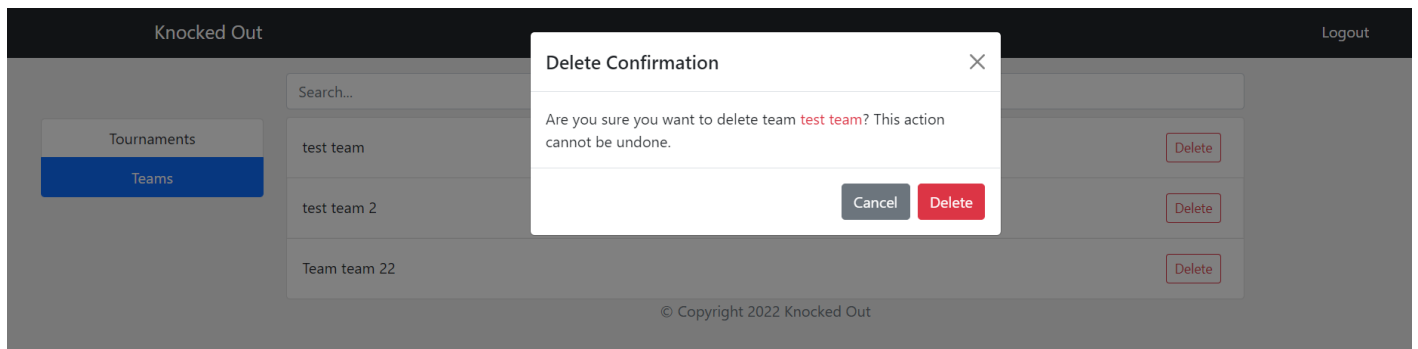
Teams

Search...

test team	Delete
test team 2	Delete
Team team 22	Delete

© Copyright 2022 Knocked Out

Admin: The admin can delete an individual team



Admin: The admin can end an individual tournament, in which the system will automatically delete the teams associated with it as well.

