

Software Requirements Specifications

Tournament Master

Contents

[Contents](#)

[List of Figures](#)

[Revision History](#)

[List of Changes](#)

[1 Introduction](#)

[1.1 Purpose](#)

[1.2 Project Scope](#)

[1.3 Glossary of Terms](#)

[1.4 References](#)

[1.5 Overview](#)

[2 Overall Description](#)

[2.1 Product Perspective](#)

[2.2 Product Features](#)

[2.3 User Classes and Characteristics](#)

[2.4 Operating Environment](#)

[2.5 Design and Implementation Constraints](#)

[2.6 Assumptions and dependencies](#)

[3 System Features](#)

[3.1 Sign In and Out](#)

[3.1.1 Description and Priority](#)

[3.1.2 Stimulus/Response Sequences](#)

[3.1.3 Functional Requirements](#)

[3.1.4 Use Cases](#)

[3.2 User Registration](#)

[3.2.1 Description and Priority](#)

[3.2.2 Stimulus/Response Sequences](#)

[3.2.3 Functional Requirements](#)

[3.2.4 Use Cases](#)

[3.3 Tournament Creation](#)

[3.3.1 Description and Priority](#)

[3.3.2 Stimulus/Response Sequences](#)

[3.3.3 Functional Requirements](#)

[3.3.4 Use Cases](#)

[3.4 Joining Tournaments](#)

[3.4.1 Description and Priority](#)

- [3.4.2 Stimulus/Response Sequences](#)
 - [3.4.3 Functional Requirements](#)
 - [3.4.4 Use Cases](#)
- [3.5 Tournament Roster and Moderator Delegation](#)
 - [3.5.1 Description and Priority](#)
 - [3.5.2 Stimulus/Response Sequences](#)
 - [3.5.3 Functional Requirements](#)
 - [3.5.4 Use Cases](#)
- [3.6 Tournament Updating](#)
 - [3.6.1 Description and Priority](#)
 - [3.6.2 Stimulus/Response Sequences](#)
 - [3.6.3 Functional Requirements](#)
 - [3.6.4 Use Cases](#)
- [3.7 Tournament Viewing](#)
 - [3.7.1 Description and Priority](#)
 - [3.7.2 Stimulus/Response Sequences](#)
 - [3.7.3 Functional Requirements](#)
 - [3.7.4 Use Cases](#)
- [3.8 Tournament Seeding](#)
 - [3.8.1 Description and Priority](#)
 - [3.8.2 Stimulus/Response Sequences](#)
 - [3.8.3 Functional Requirements](#)
 - [3.8.4 Use Cases](#)
- [3.9 Tournament Bracketing](#)
 - [3.9.1 Description and Priority](#)
 - [3.9.2 Stimulus/Response Sequences](#)
 - [3.9.3 Functional Requirements](#)
 - [3.9.4 Use Cases](#)
- [3.10 Tutorials](#)
 - [3.10.1 Description and Priority](#)
 - [3.10.2 Stimulus/Response Sequences](#)
 - [3.10.3 Functional Requirements](#)
 - [3.10.4 Use Cases](#)
- [3.11 Settings](#)
 - [3.11.1 Description and Priority](#)
 - [3.11.2 Stimulus/Response Sequences](#)
 - [3.11.3 Functional Requirements](#)
 - [3.11.4 Use Cases](#)
- [3.12 Error Message Examples](#)
 - [3.12.1 Description and Priority](#)
 - [3.12.2 Stimulus/Response Sequences](#)
 - [3.12.3 Functional Requirements](#)
 - [3.12.4 Use Cases](#)

- [4 External Interface Requirements](#)
 - [4.1 User Interfaces](#)
 - [4.2 Hardware Interfaces](#)
 - [4.3 Software Interfaces](#)
 - [4.4 Communications Interfaces](#)
 - [4.5 Domain Model](#)
 - [4.5.1 Level 0 Diagram](#)
 - [4.5.2 Level 1 Diagram](#)
 - [4.5.3 Level 2 Diagram](#)
 - [4.5.4 Entity-Relationship Diagram](#)
- [5 Other Nonfunctional Requirements](#)
 - [5.1 Performance Requirements](#)
 - [5.2 Safety Requirements](#)
 - [5.3 Security Requirements](#)
 - [5.4 Software Quality Attributes](#)
- [Appendix A: Analysis Models](#)
 - [A.1 First Elicitation Meeting Minutes](#)

List of Figures

- Figure 1: Use case sign in
- Figure 2: Use case sign out
- Figure 3: Use case user registration
- Figure 4: Use case tournament creation
- Figure 5: Use case joining tournaments when invited
- Figure 6: Use case joining tournaments when not invited
- Figure 7: Use case of tournament owner adding a moderator
- Figure 8: Use case of Updating tournament details
- Figure 9: Use case of editing a team roster
- Figure 10: Use case of viewing a tournament's details
- Figure 11: Use case of describing the manual seeding option
- Figure 12: Use case describing a tournament moderator editing tournament details
- Figure 13: Use case describing a subset of the tutorials
- Figure 14: Use case describing the different settings a user can change
- Figure 15: Use case Showing the error messages a user could generate
- Figure 16: A general layout of Tournament Master
- Figure 17: Use case diagram for Tournament Master
- Figure 18: Level 0 Data Flow Diagram for Tournament Master
- Figure 19: Level 1 Data Flow Diagram for Tournament Master
- Figure 20: Level 2 Data Flow Diagram for Tournament Master
- Figure 21: Entity-Relationship Diagram for Tournament Master

Revision History

Name	Date	Reason for Changes	Version
Initial	2015/01/22	N/A	1.0
Clarification	2015/01/27	Issues were clarified over email with Tournament Co.	1.1
Final Version 1 Revision	2015/01/28	Touch-ups and final clarification	1.2
Version 2 Draft	2015/02/26	New information and revised requirements	2.0
Version 2 Final Revision	2015/02/26	Final changes	2.1

List of Changes

Change	Section	Description
Public/Private Rules	Glossary	Created glossary terms for public, private, and hidden tournaments to accurately depict the different levels of privileges.
Invite Notifications	3.5.2, 5.3	Notifications should only occur for invitations to tournaments, permissions updates for moderators, and anything the user is involved in.
Static Team Rosters	3.4.2	Rosters cannot be changed after a tournament has started and team rosters are kept as plain text.
Data Requirements	5.1	The application should not function without an internet connection.
Safety Requirements	5.2	A disclaimer should be displayed in application removing all liability from injury.
Flow Diagram	4.1	Add information to the flow diagram for guest users.
Tournament Lock Time	3.8.3 REQ 2	Tournament information will lock upon the beginning of the first match.

Seeding Option	3.8.1	Added "N/A" as a seeding option.
Tournament Information	3.7.3 REQ 1	Tournament directors must be able to edit tournament information from the viewing page.
Tournament Viewing - no homepage	3.7.1	On the viewing page, users can view the tournament bracket and match information. Users can only view a hidden tournament if they have been invited to it. This is an important feature.
Moderator/ Director Features	3.6.2, Glossary	Add changed to Invite. Director gained moderator removal privilege.
Project Scope	1.2	Changed scope declaration and allowed for further usage than what's listed.
Tournament Byes	3.6.2	Moderators and Directors handle the special cases of byes.
Team Updating	3.6.2	A team leader should be able to edit the team members .
Use Cases	3.x.4	Added possible use cases for all system features.
Domain Model	4.5.x	Added Data Flow Diagrams for levels 0, 1, and 2
Stakeholders	1.6	Stakeholders have been added for Tournament Master.
Order Change	3.2.4	Swapped order of use case 1 and 2.
Start and End Date and Time	3.3.4	Added start and end date and time to the user interface.
Player Count	3.3.4	Remove the minimum player count for a team in a tournament.
Use Cases	3.4.4 Use case 2	Specified in use cases that invite is needed for non-public tournaments and that users can be invited to any tournament
Player Count	3.4.4 Use case 2	Removed player count from alt flow B.
Label Change	3.6.3	The first REQ-3 had label changed to REQ-2.
Forfeit	3.6.4	Added forfeit option the user interface.
Seed Changing	3.8.3 REQ-2	Elaborated on explanation of seeding specifications.
Random Seeding	3.8.3 REQ-5	Random seeding can be modified manually after random selection.

Seeding type Lock	3.8.3 REQ-6	Added requirement to specify seeding type locking.
Tutorial Interface	3.10.3 REQ-3 REQ-4	Added requirements for re-opening the tutorial interface.
Password Changing	3.11.4 Use case 1	Corrected main flow and alternative flow to require the user's current password when changing their password.
Precondition Change	3.11.4	Added precondition for a user being signed in.
Disclaimer	5.2	Modified the disclaimer statement to focus more on the layout and not the contents of the disclaimer.
Elicitation Notes	Appendix A.2	Added RS1.0 elicitation notes.
Elicitation Notes	Appendix A.3	Added RS2.0 Draft elicitation notes.

1 Introduction

1.1 Purpose

This document specifies the requirements for the creation of a tournament organization and management system titled Tournament Master. Tournament Master will allow users to create, view, and maintain tournament information in a simple and manageable way.

1.2 Project Scope

Managing tournaments is difficult because of the problems associated with structuring rounds and communicating information to many parties. Tournament Master will be a simple, self-contained application to ease tournament complexity for both organizers and players. It will be a complete tournament handling solution with features such as initial seeding, bracketing, and teams. Users with different roles in a tournament will be able to have different levels of authorization.

1.3 Glossary of Terms

Tournament Director	User who initially creates a tournament and has full moderator privileges. The director is the only user with the privilege of assigning/removing moderators and deleting the tournament.
Tournament Moderator	User who has been granted moderator privilege for a tournament by the tournament director. Moderators have the ability to modify tournament information.
User	Person who is using the software.
Guest User	Unregistered user who can only see publicly viewable information.
Team Leader	User who creates and represents a team.
Seeding	Process of matching opponents based on tournament style for the initial bracket.
Bracketing	The placement and progression style of competitors in a tournament.
Public Tournament	Tournament that is viewable and joinable by any user of the application.

Private Tournament	Tournament that is viewable by any user of the application, but requires an invitation to join.
Hidden Tournament	Tournament that requires an invitation to join and is only visible by participants.

1.4 References

- [1] Tournament Systems, Wikipedia," [online] 2013, http://en.wikipedia.org/wiki/Category:Tournament_systems (Accessed: January 2015).
- [2] jQuery Bracket Library, [online], <http://www.aropupu.fi/bracket/> (Accessed: January 2015).
- [3] PhoneGap Development Made Easier, Telerik, [online] <http://www.telerik.com/campaigns/platform/phonegap?gclid=CK-EvvPHtcMCFciEfgodBbEAFQ> (Accessed: January 2015).
- [4] imLeagues, [online] <http://www.imleagues.com/Default.aspx> (Accessed: January 2015).
- [5] Challonge, [online] 2015 http://challonge.com/tournament/bracket_generator (Accessed: January 2015).

1.5 Overview

This document begins with a general overview of the product, including product perspective, features, types of users, and operational environments. There is a detailed discussion of the system features and external interface along with their requirements. The specification finishes with a review of the non-functional requirements, including performance, safety, security, and software quality attributes.

1.6 Stakeholders

Stakeholders involved in the Tournament Co. application Tournament Master are:

- Players
- Team Leaders
- Spectators
- Tournament Organizers
- Tournament Co. Employees

2 Overall Description

2.1 Product Perspective

Tournament Master is a response to a request from Tournament Co. for an innovative and cross-platform application. The application will resemble an existing product known as IMLeagues. Similarly, it is for managing tournaments and is targeted towards non-professional users. In contrast to IMLeagues, Tournament Master will handle public, private, and hidden tournaments.

2.2 Product Features

The following are the features that Tournament Master will offer:

User Registration

The system will allow users to register for an account with Tournament Master. Registering allows users to manage, view, and update both tournament and personal information.

Team Creation

A registered user can create a team and add team members.

Tournament Creation

Users will be faced with multiple options when creating a tournament such as choosing the visibility permissions, seeding, and bracketing.

Tournament Sign Up

Users can view and sign up for tournaments if its public/private/hidden status allows.

Seeding

There are a variety of seeding options available (Reference: 3.8).

Bracketing and Results

Users will be able to see bracketing and updated results.

2.3 User Classes and Characteristics

Guest users have read-only access, which means they can view public and private tournament rosters. Registered users can have the status of director, moderator, team leader, or participant. The permissions are tournament specific; for example, a user can be a player in one tournament and the director of another.

The creator of a tournament is the director. Tournament directors can change settings, schedule matches, modify rankings, delete tournaments, and assign moderators. Moderators have all the same permissions as the director except that they are not able to assign other moderators, change tournament privacy, or delete the tournament.

2.4 Operating Environment

The application will support Android 4.1+, iOS 7+, Google Chrome, Firefox, and Internet Explorer 8+. The application can be used on any operating system that runs the supported web browsers. The main development effort will be focused towards the responsive website. The website is important because it will be converted to software with a web application framework.

2.5 Design and Implementation Constraints

The libraries available for development will be constrained to packages that are compatible with the specified operating environments. Users can choose how to be notified of certain events; these communication types are limited to email and application notifications.

2.6 Assumptions and dependencies

The tournament rankings screen will depend on jQuery and the jQuery Bracket plugin. The application will rely on email and notifications to keep users updated.

3 System Features

3.1 Sign In and Out

3.1.1 Description and Priority

Users must sign in to gain access to user-specific features. An account is needed to join, create, or moderate tournaments. A notable exception is to view non-hidden tournaments which do not need the user to be signed in. A user signs out to become a guest. A guest will be able to view tournaments but will not be able to modify any data. A user may sign out to switch accounts or to protect information from changing. This feature is high priority and mandatory to complete.

3.1.2 Stimulus/Response Sequences

Every page contains a sign in button if the user is not signed in. Pressing the button brings the user to the sign in page. The sign in page contains fields for a username and password. Once signed in, the user is returned to the previous page that they were viewing. The sign in button will transform into a sign out button. The user will be able to press the button anywhere in the interface and the session will be terminated.

3.1.3 Functional Requirements

REQ-1

The system must allow the user to enter a username and password using validated inputs.

REQ-2

The system must notify the user if the username does not exist.

REQ-3

The system must notify the user if the password is incorrect.

REQ-4

The system proceeds to the homepage if the entered information matches an account's.

REQ-5

The system terminates the session if the user signs out.

3.1.4 Use Cases

Use Case #1

Sign In

Description

This use case describes how a user will Sign In.

Actors

Registered User

Preconditions

The user must be registered.

Main Flow

1. The user clicks sign in from any page in the application.
2. **<Enter Sign In Info>** A new page is opened and the user enters their username and password, then presses "Sign In".
3. **<User Authentication>** The system confirms the user exists and information is correct.
4. The system brings the user back to the page they were on.
5. The use case ends.

Postconditions

The user now has access to additional features.

Alternative Flows

- A. At **<User Authentication>**, if the user enters a non existent username-password combination, then
 - a. The system displays the message "Username or password was incorrect".
 - b. Return to **<Enter Sign In Info>**.

User Interface

The user presses the “Sign In” button, and is brought to the sign in page. The user enters a username and password. The user presses “Sign In”, is signed in, and is brought back to the previously viewed page.



Figure 1: Use case sign in

Use Case #2

Sign Out

Description

This use case describes how a user will sign out

Actors

Registered User

Preconditions

The user must be registered.

The user must be signed in.

Main Flow

1. The user clicks sign out button in the header.
2. System responds by signing the user out and opening the index page.

Postconditions

The system ends the user's session.

Alternative Flows

There are no alternative flows for this use case.

User Interface

The user presses the sign out button. The user is signed out, and is brought to the homepage.



Figure 2: Use case sign out

3.2 User Registration

3.2.1 Description and Priority

User-specific features require an account. Registration allows users to be granted increased privileges such as the ability to create, join, and modify tournaments. This feature is high priority and mandatory to complete.

3.2.2 Stimulus/Response Sequences

The sign in page contains an option to register. The option opens the registration page. The registration page contains fields for a username, full name, password, and email address. An email address is required to register, but is not verified. The application continues to the homepage once registration is complete.

3.2.3 Functional Requirements

REQ-1

The system must allow the user to register for the application using validated inputs.

REQ-2

The system must notify the user if registration fails.

REQ-3

The system must notify the user if registration succeeds.

3.2.4 Use Cases

Use Case #1

User Registration

Description

This use case describes the registration process for a user.

Actors

Unregistered User

Preconditions

The user must not be signed in.

Main Flow

1. The user opens the application and clicks “Sign In” button.
2. The user clicks either the “Register” or “Create an account” buttons.
3. **<Registration>** The user enters a username, full name, email address, password, and a confirmed password.
4. **<Registration Info Verification>** The system checks to see if the input fields are valid.
5. The system automatically signs in the user and brings them to the last page they were on.
6. The use case ends.

Postconditions

The new user information is added to the database to ensure identical accounts cannot be created.

Alternative Flows

- A. At **<Registration Info Verification>**, if the user enters an email or username that is already used by another user, then
 - a. The system displays the message “Username or email address already taken”.
 - b. The existing fields are highlighted.
 - c. Return to **<Registration>**.
- B. At **<Registration Info Verification>**, if the input fields are not valid, then
 - a. The system displays the message “Highlighted fields are invalid”.
 - b. The invalid fields are highlighted.
 - c. Return to **<Registration>**.

User Interface

The user presses the “Sign In” button to be brought to the sign in page. The user then presses either the “Register” or “Create an account” button to be brought to the registration page. The user fills out the fields, presses “Register” to create an account, and then is brought back to the previously viewed page.

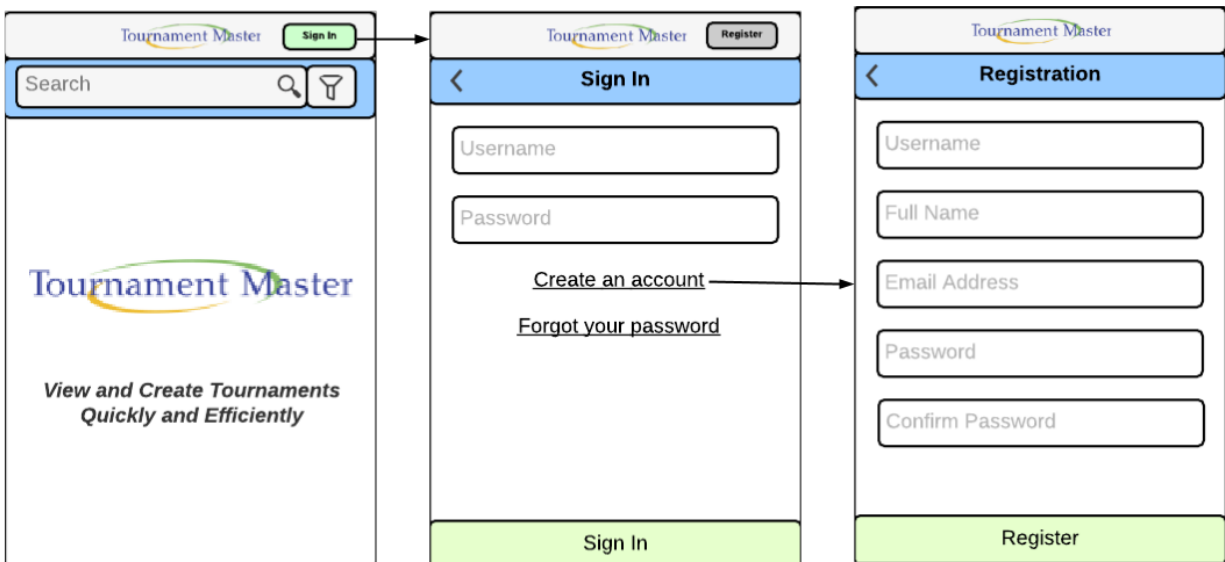


Figure 3: Use case user registration

3.3 Tournament Creation

3.3.1 Description and Priority

Tournament directors may delegate moderator responsibilities to other registered users. Users who are on a roster for a tournament but are not registered for an account within the system are unable to receive moderator responsibility for a tournament. In order for a user to create a tournament, a tournament name, a sport category, a maximum number of registered teams, a seeding type, start/end date and time, location, privacy settings(public/private/hidden), and a required number of players per team must be specified. Creating tournaments has very high priority and must be completed.

3.3.2 Stimulus/Response Sequences

The tournament creation option will be available when a user signs in to the application. The user can then fill out the available fields and click a button to create the tournament. An error message will be displayed if any required fields are empty which notifies the user to fix their input.

3.3.3 Functional Requirements

REQ-1

The system must allow users to create tournaments.

REQ-2

The system must notify the user if any required tournament creation input fields are incomplete.

REQ-3

The system must allow tournament directors and moderators to invite registered users to tournaments.

3.3.4 Use Cases

Use Case #1

Create Tournament

Description

This use case describes how a registered user can create a tournament.

Actors

Registered User

Preconditions

The user must be registered.

The user must be signed in.

Main Flow

1. The user clicks create tournament from the index page.
2. **<Tournament Info Input>** The user enters the tournament name, sport category, maximum team count, a seeding type, the start/end date and time, location, privacy setting, and hidden setting.
3. **<Tournament Verification>** The system checks to see if the tournament information is valid.
4. The tournament is created.
5. The user is assigned the status of tournament Director.
6. The system grants the tournament Director additional privileges.
7. The use case ends.

Postconditions

The tournament director is given director privileges until the tournament is deleted.

Alternative Flows

- A. At **<Tournament Verification>**, if the user enters invalid info or does not fill out required fields, then
 - a. The system displays the message “Highlighted fields were incomplete or invalid”.
 - b. The invalid fields are highlighted.
 - c. Return to **<Tournament Info Input>**.

User Interface

The homepage contains a “Create Tournament” when the user is signed in. Pressing it brings the user to the tournament creation page.

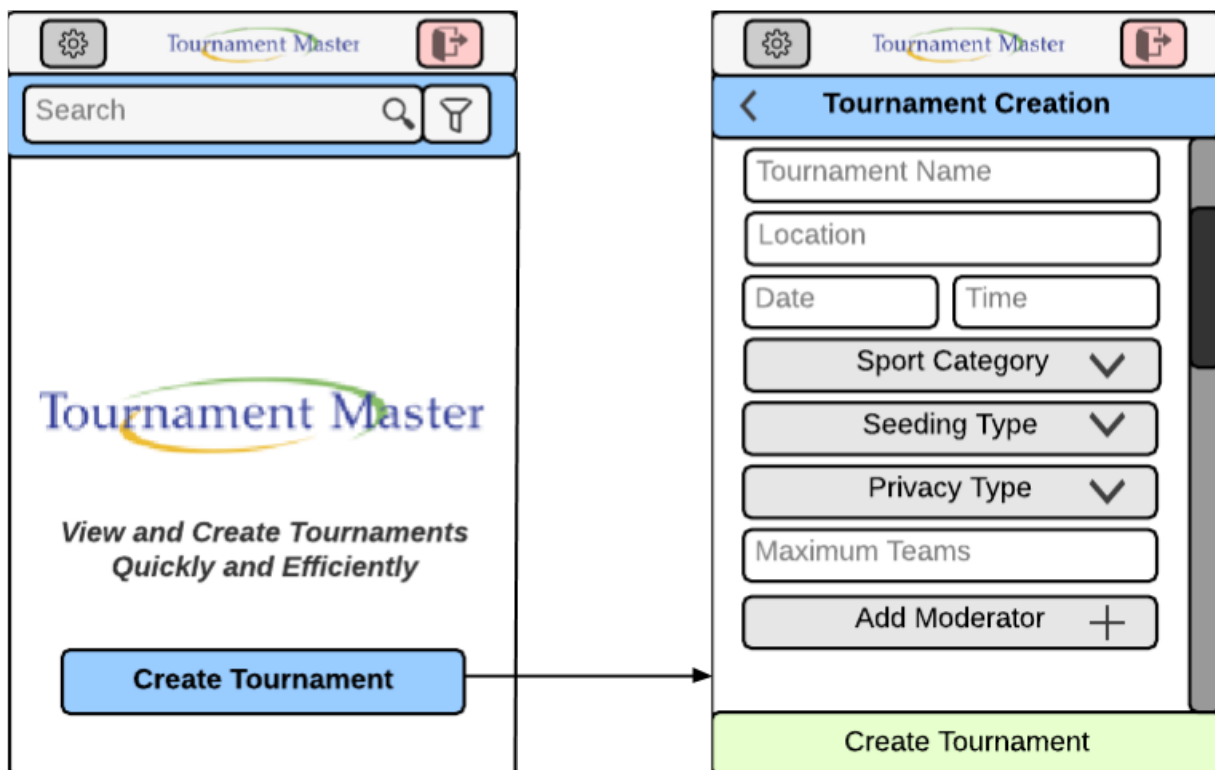


Figure 4: Use case tournament creation

3.4 Joining Tournaments

3.4.1 Description and Priority

Users may register a roster into public/private team tournaments or themselves into public/private single tournaments. Users may also be invited to join any public/private/hidden tournaments. Joining tournaments is a high priority feature as teams and players are required for tournaments to occur.

3.4.2 Stimulus/Response Sequences

The team roster should meet the minimum and maximum number of players requirement. The team roster is listed in plain text format upon joining as a team.

3.4.3 Functional Requirements

REQ-1

The system must allow users to register for an existing public/private tournament.

REQ-2

The system must allow users who have been invited to a hidden tournament to register in that tournament.

REQ-3

The system must allow users to register for an existing tournament to which they have been invited.

REQ-4

The system must notify the user if required team information is incomplete for the registration of a team-based tournament.

3.4.4 Use Cases

Use Case #1

Join Tournament with Invitation

Description

This use case describes how a user can join a tournament as a single or as a team when invited.

Actors

Registered User

Tournament Director/Moderator

Preconditions

The user must be registered.

Main Flow

1. Do use case titled **View Tournament** as a director or moderator
2. Click “Invite Player/Team” and enter the registered user's email
3. The system sends an email to the user with all tournament information and lets them know it will appear in tournament search.
4. Do use case titled **View Tournament** as the registered user and click on the tournament labeled Invited
5. The user accepts the invitation to the tournament
6. **<Register Team>** The user registers their team and adds all team information.
7. **<Team Verification>** The system verifies the team information is complete and valid.
8. The user or team is now part of the tournament.
9. The user case ends.

Postconditions

If the registered user joined a hidden tournament it will now be viewable in view tournaments.
The user is now considered a team leader/ tournament participant.

Alternative Flows

- A. At **<Team Verification>**, if the teams player count does not meet requirements, then
 - a. The system displays the message “Team sizes can only be between size x-x, please meet this requirement”.
 - b. Go back to step **<Register Team>**.
- B. At **<Team Verification>**, if any team registration fields are incomplete, then
 - a. The system displays the message “Please fill in all fields for your team”.
 - b. Go back to step **<Register Team>**.

User Interface

An “Invite player/team” button is available to the director and moderators of a tournament on the viewing page. Pressing it opens a dialog window where a user’s email address can be entered. Invited users will be emailed with information about the tournament. The tournament will highlighted in search results with an invited label. They press “Join” on the tournament viewing page and a popup will open with fields required to join. They fill out the fields and press the “Join” button on the popup to join the tournament.

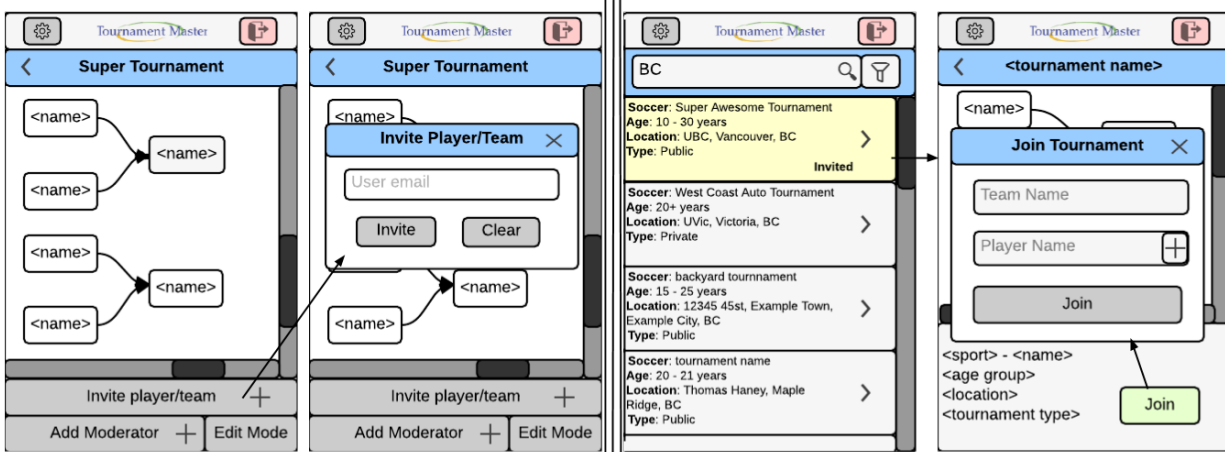


Figure 5: Use case joining tournaments when invited

Use Case #2

Join Tournament without Invitation

Description

This use case describes how a user can join a tournament as a single or as a team without being invited.

Actors

Registered User

Preconditions

The user must be registered.

The tournament must be public.

Main Flow

1. Do use case titled **View Tournament**.
2. **<Join Tournament>** The user clicks join tournament.
3. **<Join Verification>** The system verifies the tournament is not full.
4. **<Register Team>** The user registers their team and adds all team information.
5. **<Team Verification>** The system verifies the team information is complete and valid.
6. The user or team is now part of the tournament.
7. The use case ends.

Postconditions

The user is now considered a team leader/ tournament participant.

Alternative Flows

- A. At **<Join Verification>**, if the tournament is at its max participants, then
 - a. The system displays the message “This tournament is full and cannot be joined”.
 - b. Go back to **<Join Tournament>**
- B. At **<Team Verification>**, if any team registration fields are incomplete, then
 - a. The system displays the message “Please fill in all fields for your team”.
 - b. Go back to step **<Register Team>**.

User Interface

The “Join” button opens a popup with fields for team information. The popup could be expanded upon in future builds. Filling out the fields, and pressing “Join,” closes the popup, and makes the join tab get replaced by more tournament viewing space.

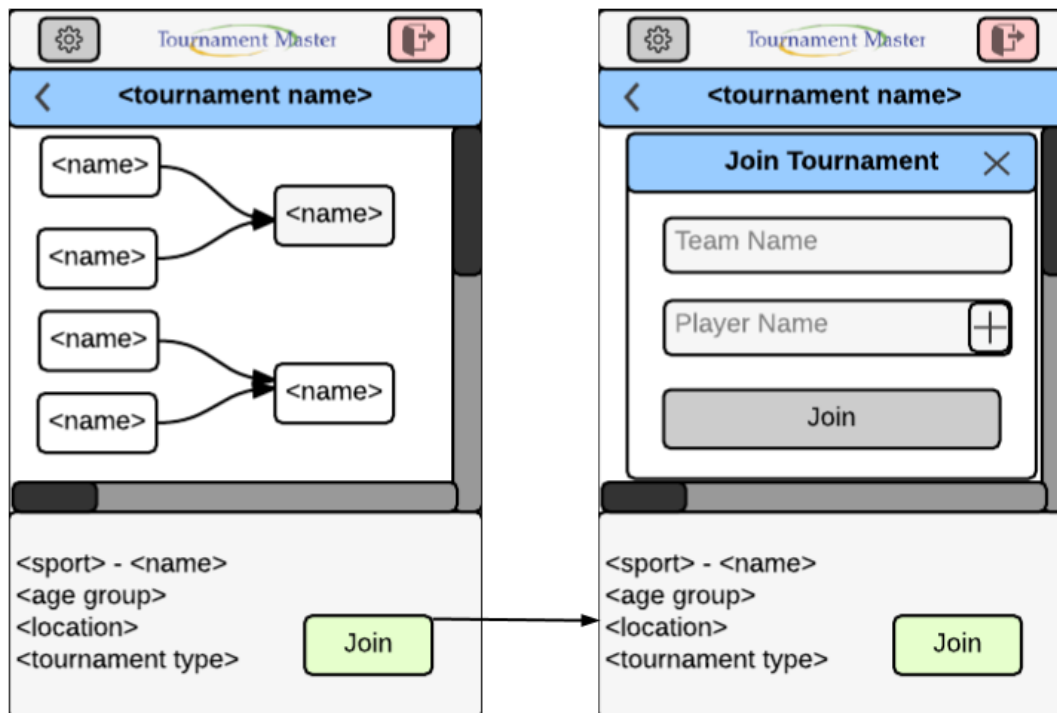


Figure 6: Use case joining tournaments when not invited

3.5 Tournament Roster and Moderator Delegation

3.5.1 Description and Priority

Tournament directors delegate moderators for their tournaments. Tournament delegation has high priority and must be completed.

3.5.2 Stimulus/Response Sequences

Tournament directors will have the option to add a moderator. When added, the user will receive a notification of the permissions update.

3.5.3 Functional Requirements

REQ-1

Tournament directors must have the ability to assign moderator permissions to any registered or unregistered user. Moderators can be assigned when a director or moderator adds them by email through the tournament viewing page. An email will be sent with instructions on how to register if no account exists for a given email address. Otherwise, if the user is already registered accepting the invitation in the email will automatically update them as moderator status.

REQ-2

The system must notify a tournament director if the user they invite to be a moderator is not registered.

REQ-3

Users which are invited to be a moderator for a tournament must have the ability to deny the position.

3.5.4 Use Cases

Use Case #1

Tournament Moderator Delegation

Description

This use case describes how a tournament director may delegate administrative privileges.

Actors

Tournament Director

Registered Users

Preconditions

Only tournament directors can delegate administrative privileges.

Only registered users can be appointed moderators.

The user must be signed in.

The user must be on the View Tournaments page.

Main Flow

1. The Tournament Director must click the “Add Moderator” button.
2. **<Add Moderator>** The tournament director enters a registered user’s email.
3. **<Moderator Verification>** The system verifies the email corresponds to a registered user.
4. An invitation is sent to that registered user by email.
5. **<Accept Notification>** The user accepts the invitation.
6. The use case ends.

Postconditions

The user can now edit tournament details and is assigned status of tournament moderator.

Alternative Flows

- A. At, **<Moderator Verification>**, if the email is already a moderator, then
 - a. The system will display the message “This user is already a moderator”.
 - b. The user is brought back to **<Add Moderator>**.
- B. At, **<Moderator Verification>**, if the email does not exist, then
 - a. The system will display the message “That user does not exist, please try again”.
 - b. The user is brought back to **<Add Moderator>**.

User Interface

An “Add Moderator” button will be available to the director of a tournament on the viewing page. Pressing it opens a popup with a field for an email address. Pressing “Invite” sends an email to the user with an invitation to become a moderator.

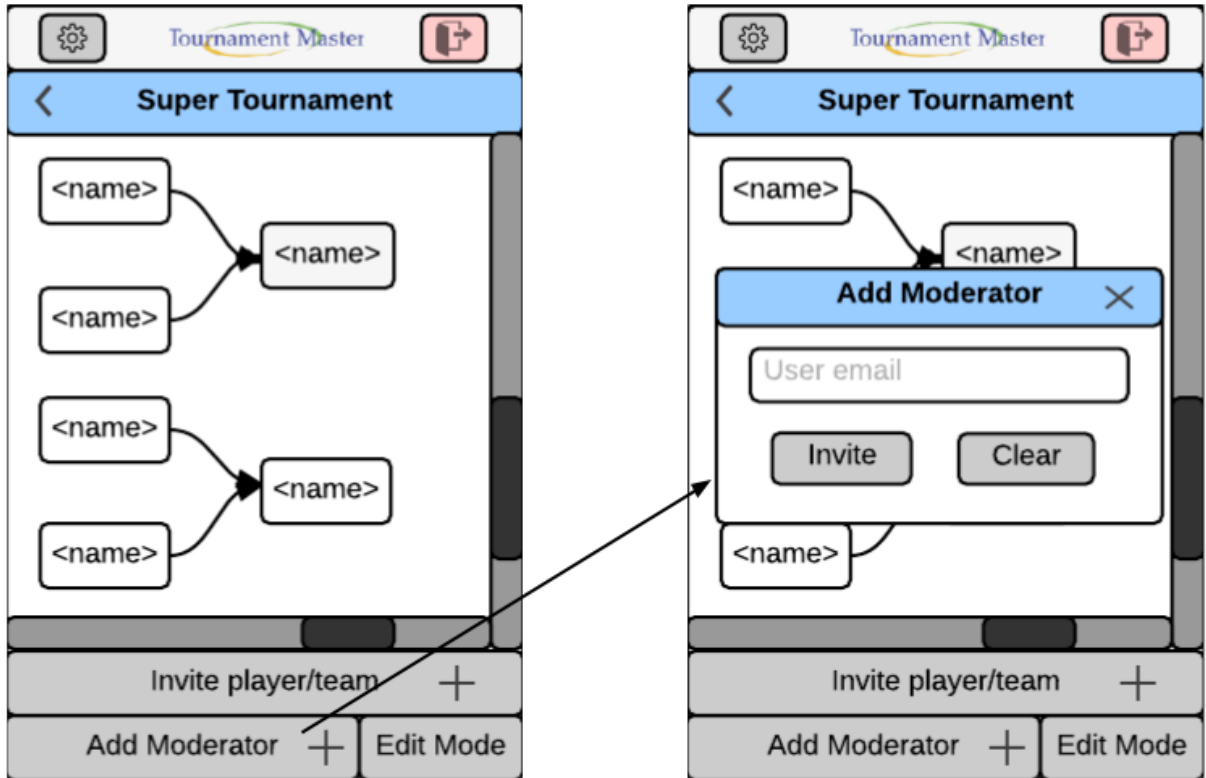


Figure 7: Use case of tournament owner adding a moderator

3.6 Tournament Updating

3.6.1 Description and Priority

Tournament bracket and setting updates are only handled by the director and moderators of the tournament. Team rosters can be edited by team leaders before tournaments begin. This system feature has very high priority and must be completed.

3.6.2 Stimulus/Response Sequences

The tournament Director and Moderators can:

- Update the scores
- Update tournament and decide wins or losses
- Manage draws, forfeits, disqualifications, and special conditions such as byes
- Invite players or teams to appropriate tournaments

The directors and moderators will have a similar view of the tournament bracket to standard users but they will be able to edit various fields and options throughout the page.

Team leaders can edit:

- Team Name
- Team Roster

The team leaders can only edit this information before a tournament begins.

3.6.3 Functional Requirements

REQ-1

Updating scores of matches in progress and inputting the outcome of a match within a tournament must be available to tournament directors and moderators.

REQ-2

A director must be able to change a private tournament into a public tournament. Modifying a public tournament to a private tournament is not possible.

REQ-3

A director must be able to change a hidden tournament into a private tournament. Modifying a private tournament to a hidden tournament is not possible.

3.6.4 Use Cases

Use Case #1

Tournament Updating

Description

This use case describes how tournament details are updated.

Actors

Tournament Director/Moderator

Preconditions

Only tournament directors and moderators can edit tournament details.

Main Flow

1. Do the use case titled **View Tournament** as a moderator or director.
2. The user now clicks the “edit” button.
3. The user now clicks the edit symbol for a specific game.
4. **<Update Tournament>** The user can now update all game details and clicks “submit”.
5. **<Update Verification>** The system verifies information is complete and correct.
6. The user can now click save to save changed information.
7. Tournament details are updated.
8. The use case ends.

Postconditions

The tournament details will be updated for all other viewers.

Alternative Flows

- A. At, **<Update Verification>**, if tournament information is left incomplete or invalid, then
 - a. The system will display the message “Tournament details are incomplete”.
 - b. Incomplete fields will be highlighted.
 - c. The user will be brought back to **<Update Tournament>**.

User Interface

When the user clicks the Edit Mode button the format of the application will change so games can be editable. When the user clicks the edit button for a game a pop-up will appear to input game information. When the information is submitted and verified the bracket will automatically update.

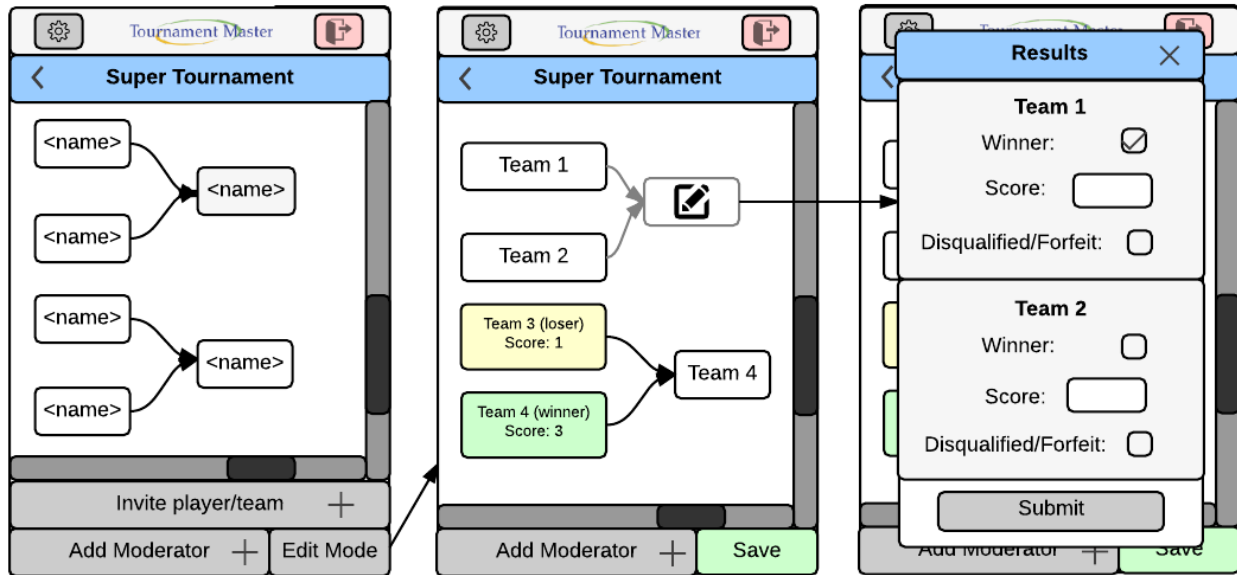


Figure 8: Use case of updating tournament details

Use Case #2

Team editing

Description

The use case describes how a user can edit their team roster.

Actors

Any Registered user that is a team leader.

Preconditions

The user must be registered for the application.

The user must have created a team and gained the team leader status.

Main Flow

- 1) Do use case titled **View Tournament** as a team leader.
- 2) The user double clicks on their team name.
- 3) The user clicks the edit button on their team info pop-up.
- 4) **<Edit Team Info>** The user can now add and remove players from their team roster and clicks submit.
- 5) **<Info Verification>** The system checks to make sure the team information is still valid.
- 6) The team information is now updated for everyone.
- 7) The use case ends.

Postconditions

All other users will now see the updated team roster.

Alternative Flows

- A. At, **<Info Verification>** if the team name is not complete, then
 - a. The system displays the message "Team name is not complete"
 - b. The user is brought back to **<Edit Team Info>**
- B. At, **<Info Verification>** if the minimum player count does not meet requirements, then
 - a. The system displays the message "Team size is too small"
 - b. The user is brought back to **<Edit Team Info>**
- C. At, **<Info Verification>** if the maximum player count does not meet requirements, then
 - a. The system displays the message "Team size is too large"
 - b. The user is brought back to **<Edit Team Info>**

User Interface

As a team leader, double clicking on a team created by this type of user will open a popup with information about the team. Clicking the edit button results in a different popup that allows the user to edit the team name and team members. After editing, the save button will bring the user back to the previous page to view the new changes.

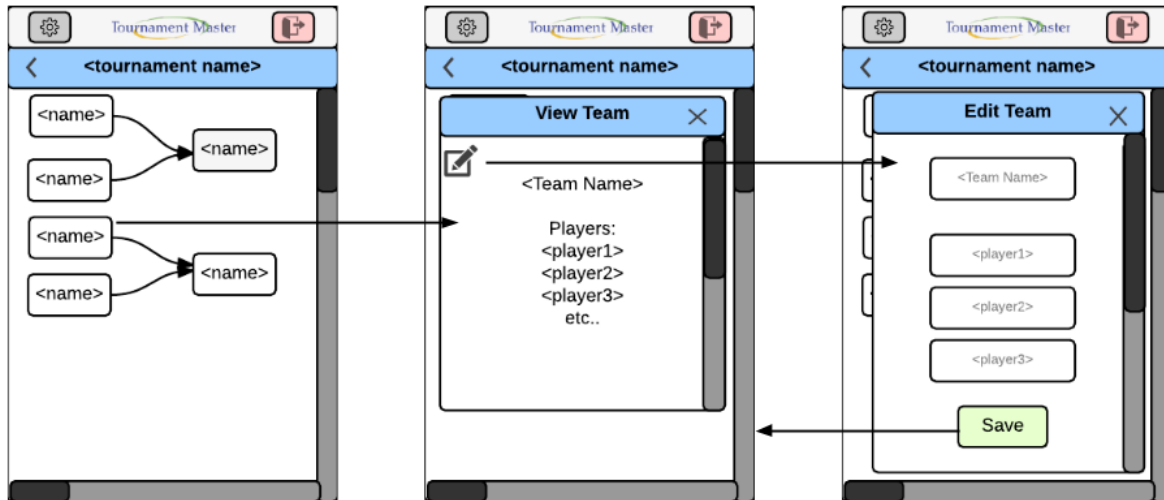


Figure 9: Use case of editing a team roster

3.7 Tournament Viewing

3.7.1 Description and Priority

On the viewing page, users can view the tournament bracket and match information. Users can only view a hidden tournament if they have been invited to it. This is an important feature.

3.7.2 Stimulus/Response Sequences

Users may search for tournaments by desired tournament details. Tournaments can be selected and information can be viewed on another page.

3.7.3 Functional Requirements

REQ-1

Tournament directors must be able to edit tournament information from the tournament viewing page. They cannot edit brackets or seeding if the tournament has been started.

REQ-2

Users must be able to search for and view hidden tournaments they are apart of and all public/private tournaments

REQ-3

The system must notify the user if required input fields are incomplete when searching

3.7.4 Use Cases

Use Case #1

Tournament Viewing

Description

The use case describes how a user can view a tournament.

Actors

Any User

Preconditions

If it is a hidden tournament, the user must be invited to the tournament to view its details.

Main Flow

1. **<Tournament Search>** The user searches for tournaments in the search bar on the index page.
2. **<Tournament Results>** The system updates the results in real time.
3. The user selects the tournament they desire.
4. The system opens the tournament in a new page.
5. The use case ends.

Postconditions

No postconditions are performed.

Alternative Flows

- A. At, **<Tournament Results>** if there are no results, the user will have no options to select, so
 - a. Go back to step **<Tournament Search>**.

User Interface

Typing text into the search field will continuously update matching tournaments. Selecting a tournament opens its viewing page.

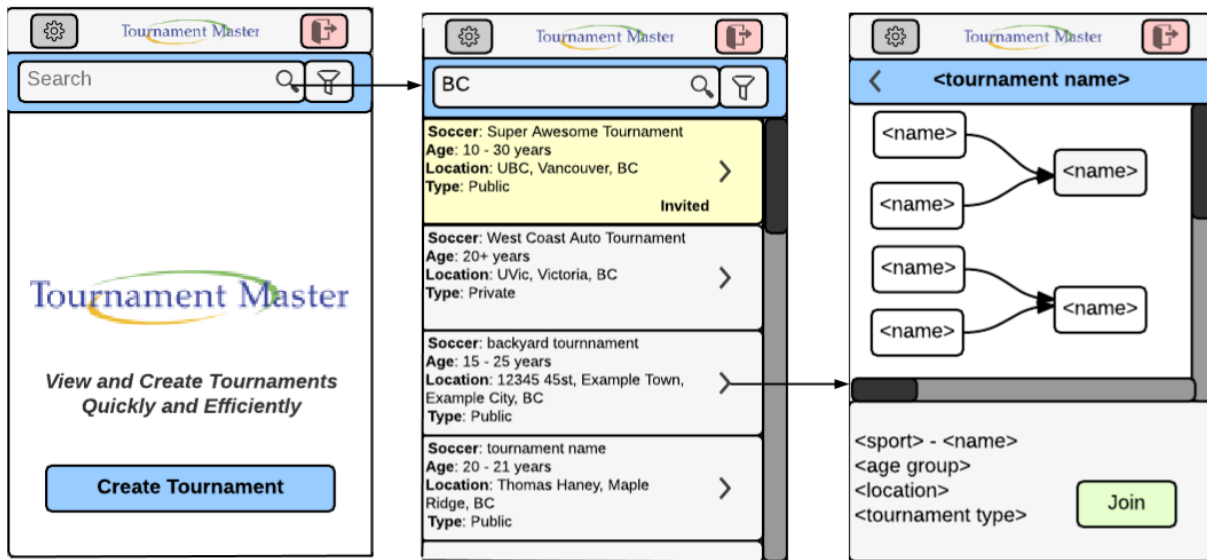


Figure 10: Use case of viewing a tournament's details

3.8 Tournament Seeding

3.8.1 Description and Priority

Tournaments set up opponents based on a seeding system. Usually opponents are matched up based on the greatest possible divergence between skill. Seeding intends to quickly distill tournament rounds into their most competitive versions involving evenly matched players/teams.

Some bracketing formats are restricted to specific seeding options. The following table show compatible seeding and bracketing formats:

Seeding Type Option	Bracketing Type Option
N/A	Round Robin
Round Robin	Single Elimination Elimination
Manual Selection	Single Elimination Elimination
Random Selection	Single Elimination Elimination
Swiss System	Swiss System

Seeding is a critical priority feature and must be completed.

3.8.2 Stimulus/Response Sequences

Within the tournament creation process the Director must select a seeding type and bracketing format based on the set of compatible pairs.

3.8.3 Functional Requirements

REQ-1

Multiple seeding options must be available to the tournament director.

REQ-2

The system allows the user to switch between manual, random, and red robin seeding types before the tournament has started.

REQ-3

Manual seeding can be reorganized by the team moderator as teams join a tournament. It is configurable until the first match starts, at which point it becomes locked for the duration of the tournament.

REQ-4

The system locks the round robin seeding outcome on tournament creation and decides it automatically. Note: REQ-2 is still applicable.

REQ-5

The system allows the random seeding type to be modified by manually after the tournament creation.

REQ-6

The system does not allow the user to change seeding types once the tournament has started.

3.8.4 Use Cases

Use Case #1

Tournament Seeding

Description

This use case describes the seeding system for Tournament Master

Actors

Tournament Director

Tournament Moderator

Preconditions

The Tournament Seeding is selected in tournament creation.

Main Flow

- 1) Do use case titled **Sign In**.
- 2) See use case **Tournament Creation**.
- 3) See use case **Tournament Viewing** as a tournament director.
- 4) **<Tournament Seeding>** A tournament director or moderator can edit the seeding if the seeding option is manual by dragging and dropping teams.
- 5) **<Seeding Verification>** The system confirms the seeding process is complete and valid.
- 6) Seeding is now setup for the tournament.
- 7) The use case ends.

Postconditions

The system adds the tournament seeding type (selected in **<Tournament Seeding >**) to the created tournament.

Alternative Flows

- A. At, **<Seeding Verification>**, if the current seeding is not complete, then,
 - a. The system displays the message "Tournament seeding is currently incomplete".
 - b. The user continues at step **<Tournament Seeding>**.

User Interface

Clicking “Edit Mode” makes more screen space available. When in edit mode, the director or moderators can drag and drop teams into different positions for manual seeding.

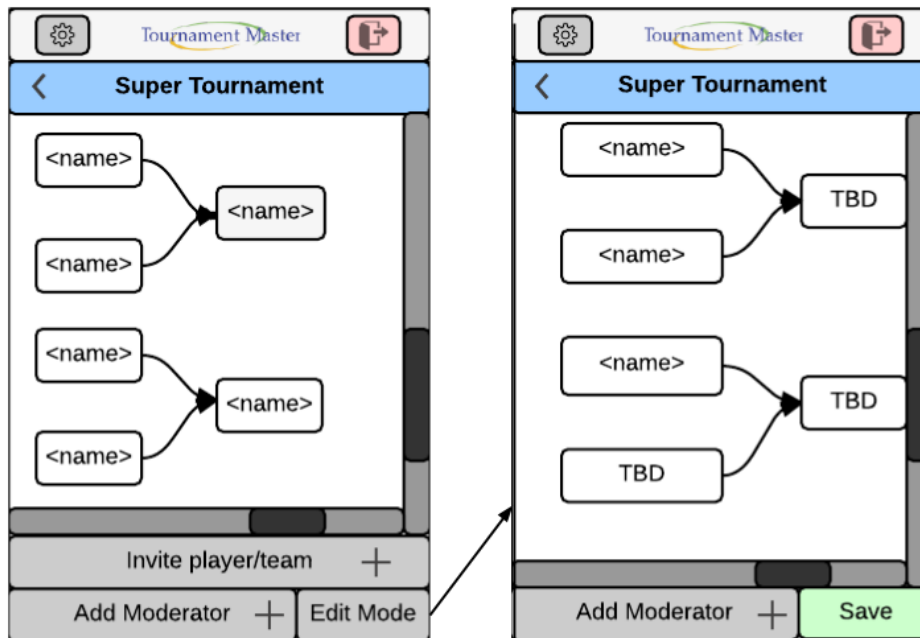


Figure 11: Use case describing the manual seeding option

3.9 Tournament Bracketing

3.9.1 Description and Priority

Tournament brackets support Swiss and single-elimination (otherwise known as a knockout). The bracketing format can be selected as an option on the tournament creation page and will affect every match and elimination. Supporting all common bracketing formats is essential for a tournament application to be useful and therefore is a high priority and must be completed.

3.9.2 Stimulus/Response Sequences

The tournament format can be chosen on the tournament creation page. The format selected will determine the matches that take place, along with the elimination of players or teams. The diagram of rounds on the tournament page will be updated accordingly.

3.9.3 Functional Requirements

REQ-1

The bracketing status will only be updated once the tournament director has updated match outcomes.

3.9.4 Use Cases

Use Case #1

Tournament Bracketing

Description

This use case describes updating the bracketing system for Tournament Master

Actors

Tournament Director

Tournament Moderator

Preconditions

There are no preconditions for this use case.

Main Flow

- 1) Do use case titled **Sign In**.
- 2) See use case **Tournament Creation**.
- 3) See use case **Tournament Viewing**.
- 4) See use case **Tournament Updating**.
- 5) Once the user updates game results, the system updates bracketing for all tournament viewers.
- 6) The use case ends.

Postconditions

There are no postconditions for this use case.

Alternative Flows

There are no alternative flows for this use case.

User Interface

The interface format switches when a director or moderator clicks “Edit Mode.” In edit mode, the match results can be updated by pressing one of the edit buttons that appear. The buttons open a popup with fields for information about the match. Once the information is submitted, the tournament bracket on the screen will automatically update.

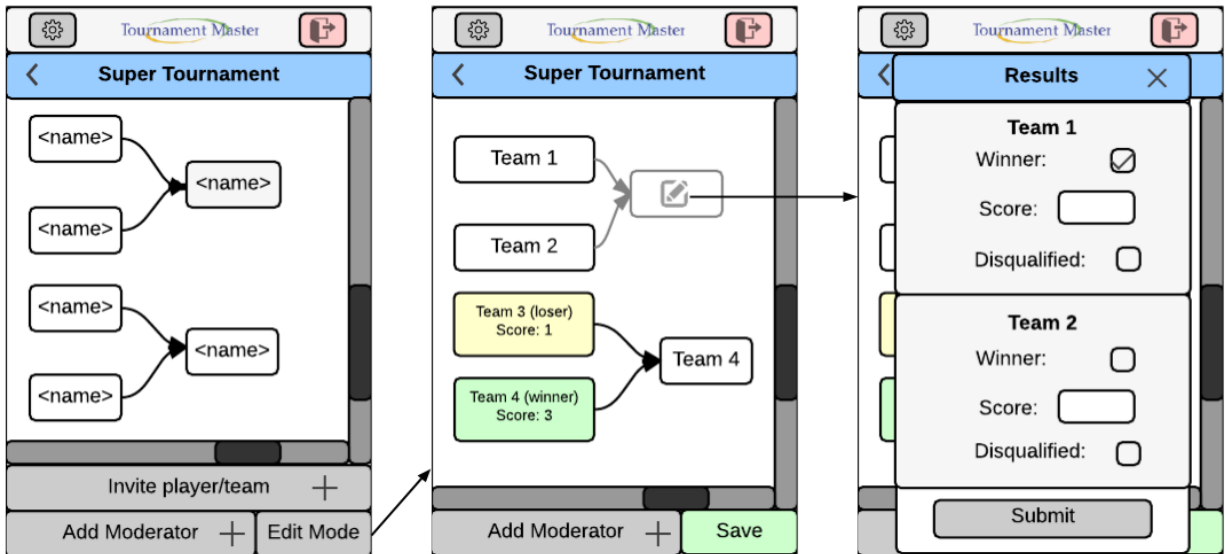


Figure 12: Use case describing a tournament moderator editing tournament details

3.10 Tutorials

3.10.1 Description and Priority

When a user first uses the software, the tutorials will be displayed to introduce each feature. The tutorials can be configured in the application settings where there will be an option to reinitialize the tutorial interface. The tutorials will help with the overall usability of the application and assist new users. Tutorials are a low priority feature whose completion is not mandatory.

3.10.2 Stimulus/Response Sequences

Helpful bubbles containing text will appear over a darkened screen to describe functionality. An option will be available to close the tutorial.

3.10.3 Functional Requirements

REQ-1

The system prompts the user with text describing the current screen.

REQ-2

The system must allow the user to exit a tutorial prematurely.

REQ-3

The settings for that user must have the tutorial mode on.

REQ-4

The system has the tutorial mode setting default to 'on' when the application is first launched.

3.10.4 Use Cases

Use Case #1

Tutorials

Description

This use case describes the tutorials for tournament master

Actors

Registered and non-registered users

Preconditions

The tutorial will show up the first time the application is opened and if chosen on the options page

Main Flow

- 1) The user opens the application for the first time
- 2) **<Tutorial>** The tutorial opens on the front page and can be used to step through the entire program. The tutorial can be closed at anytime.
- 3) **<Settings Tutorial>** The user opens the settings page and selects tutorial.
- 4) The use case ends.

Postconditions

The tutorial will never appear automatically again after the first time the application is opened.

Alternative Flows

- A. At, **<Settings Tutorial>**, if the user selects the tutorial option, then
 - a. The system opens up the tutorial
 - b. Go back to step **<Tutorial>**

User Interface

A first-time user will be presented with tutorial messages, explaining features of the application. The user can close the tutorial at any point, and the tutorial option will be turned off in the settings. Tutorials can be turned on again from the settings page.

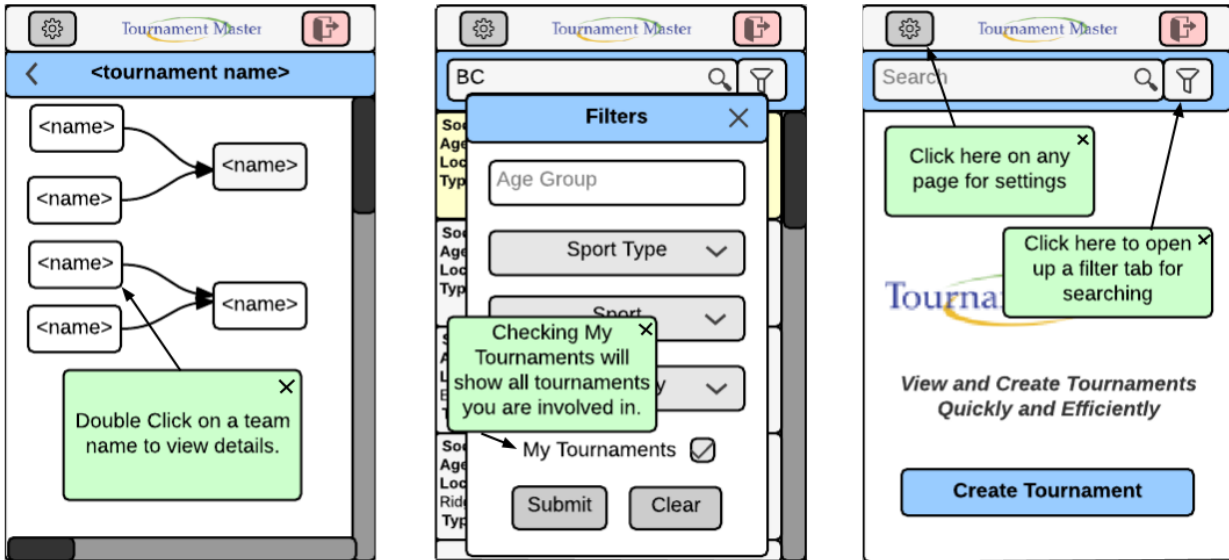


Figure 13: Use case describing a subset of the tutorials

3.11 Settings

3.11.1 Description and Priority

Users should be able to edit standard profile information and system settings. Users can modify their email address, password, and notification preferences. This is a mandatory feature for all registered users.

3.11.2 Stimulus/Response Sequences

The personal settings will be accessible via a button. When the button is pressed a new page will appear with a listing of all the configurable changes.

3.11.3 Functional Requirements

REQ-1

Users are able to update their account information and system settings.

REQ-2

The system must notify the user if updated settings fields are incomplete.

3.11.4 Use Cases

Use Case #1

Password Change

Description

This use case describes the user changing their password

Actors

Registered Users

Preconditions

Users must be registered.

User must be signed in.

Main Flow

1. The user navigates to the settings page.
2. The user can view their account information and edit the password.
3. **<Current Password Inputted>** The user enters their current password.
4. **<New Password Inputted>** The user enters their new password.
5. **<Information Verification>** The system confirms the user's password.
6. A notification stating the new information has been saved appears.
7. The use case ends.

Postconditions

The system updates the database with the user's new information.

Alternative Flows

- A. At **<Information Verification>**, if the user enters an incorrect password, then
 - a. The system displays the message " Password is incorrect, please try again".
 - b. Return to **<Enter Sign In Info>**

User Interface

The user interface for this use case can be found below in Use Case #2.

Use Case #2

Toggle Tutorials

Description

This use case describes how the user turns the tutorials on/off.

Actors

Registered Users

Preconditions

Users must be registered

Main Flow

1. Do use case titled **Sign In**.
2. The user navigates to the settings page.
3. The user activates the toggle switch.
4. The system saves the current state of the toggle switch.
5. The user can see the new change.
6. The use case ends.

Postconditions

The system updates the database with the current state of the toggle switch.

Alternative Flows

There are no alternative flows for this use case.

User Interface

A settings button is available when the user is signed in. Pressing the button will bring the user to the settings page. The settings page contains fields to update user information and preferences.

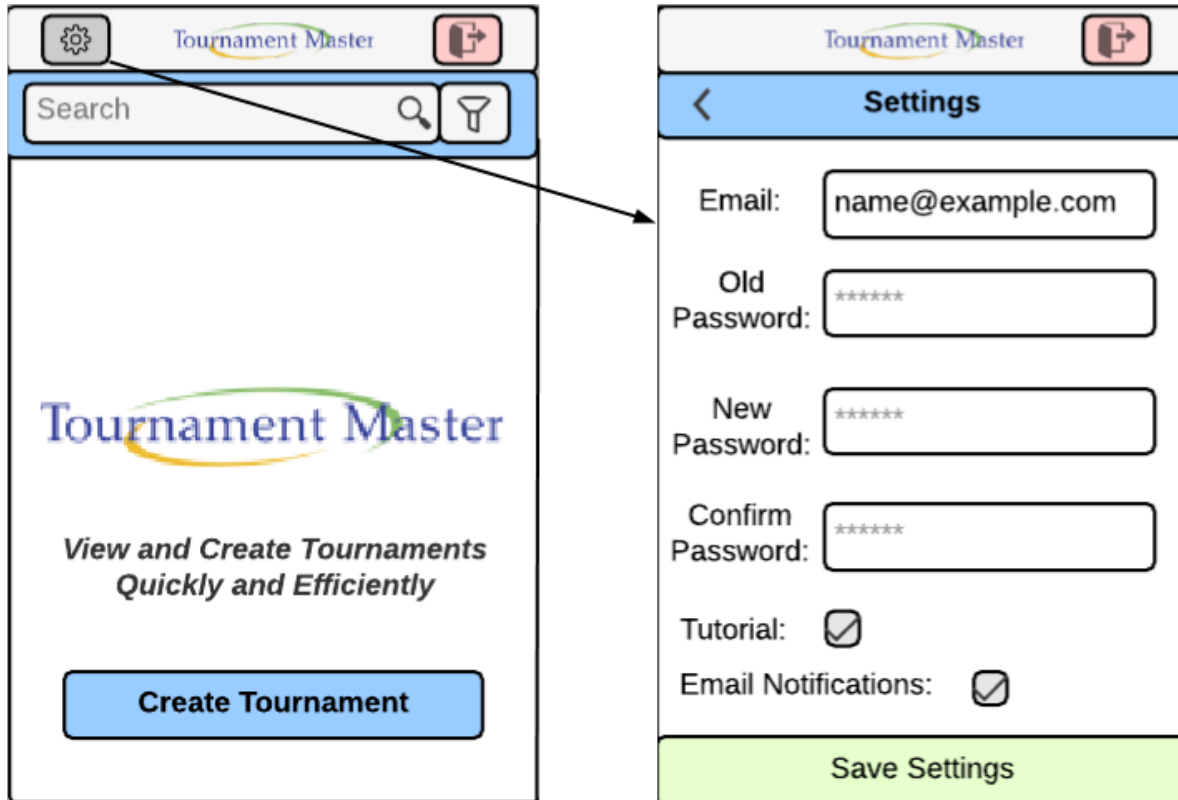


Figure 14: Use case describing the different settings a user can change

3.12 Error Message Examples

3.12.1 Description and Priority

The descriptions for this section were discussed in previous sections

3.12.2 Stimulus/Response Sequences

The stimulus/response sequences for this section were discussed in previous sections

3.12.3 Functional Requirements

The descriptions for this section were discussed in previous sections

3.12.4 Use Cases

Depending on the availability of space error messages can be displayed with red text on screens or boxes can appear and slowly fade away.

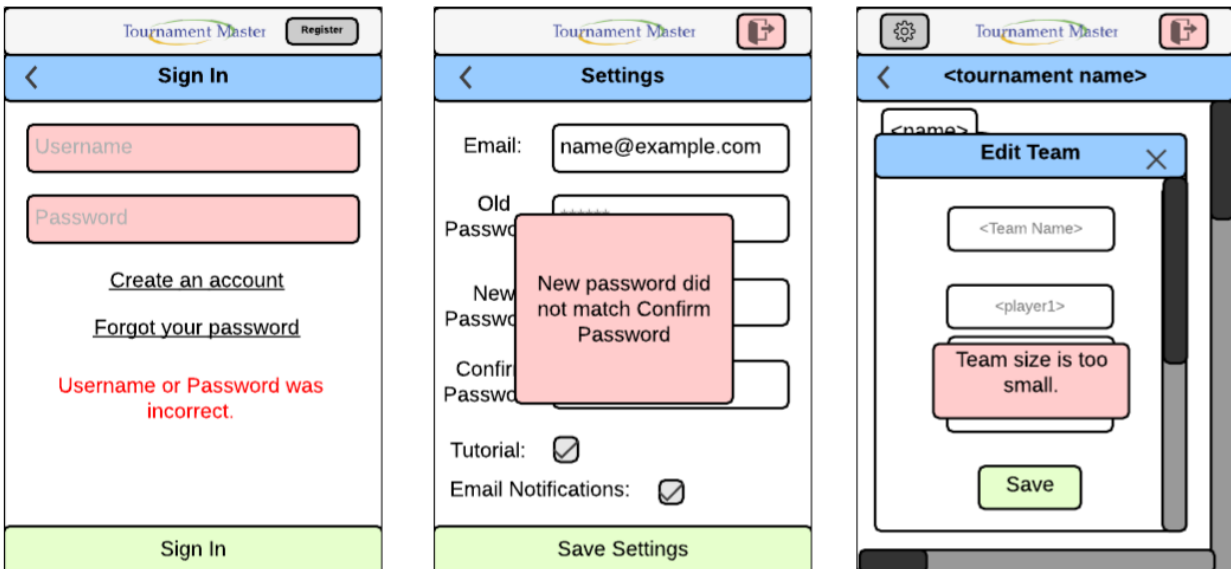


Figure 15: Use case showing the error messages a user could generate

4 External Interface Requirements

4.1 User Interfaces

The application should be accessible for users with little to no experience with similar applications. The sections and transitions in the application should be clear, and information should be well organized so that new users have no problems finding their way through the application.

Users are able to access all features of the application from the home page once logged in.

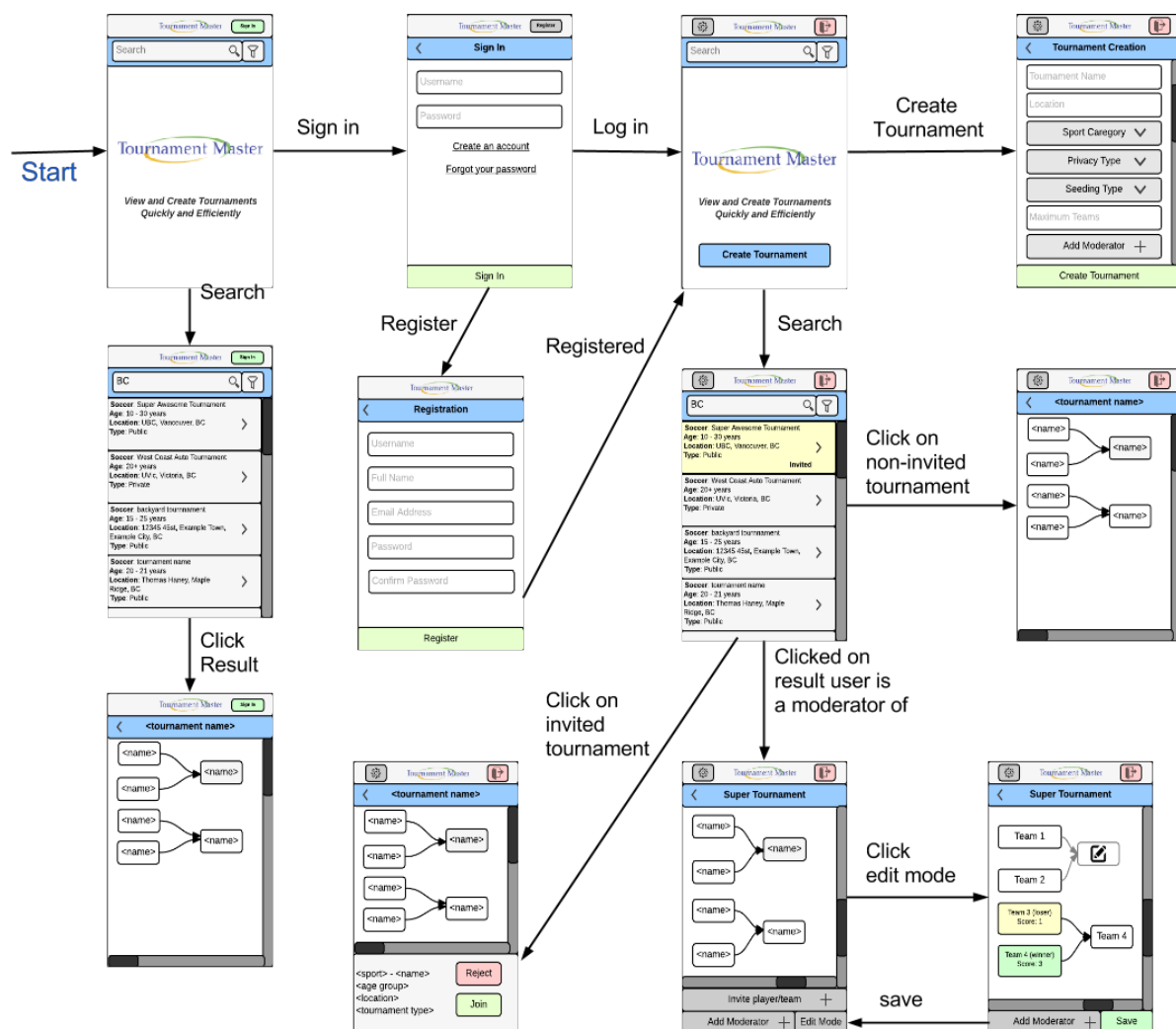


Figure 16: A general layout of Tournament Master

4.2 Use Case Diagram

The following diagram is a use case diagram that provides a simple representation of a user's interaction with the system and depicting use case specifications.

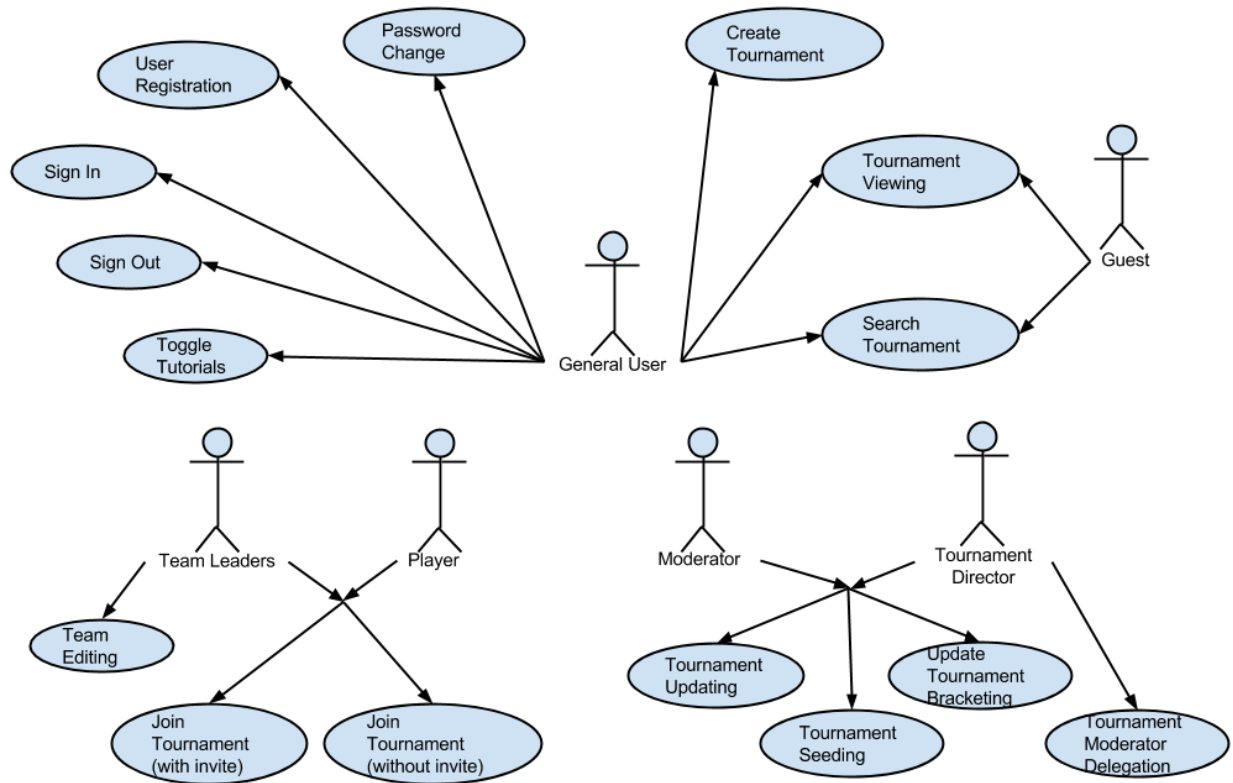


Figure 17: Use case diagram for Tournament Master

4.3 Hardware Interfaces

The application will be web-based and as such any supported device must have an internet connection in to be used. If the application is opened without internet connection an error message should be displayed.

4.4 Software Interfaces

Data will be transmitted to and from a Tournament Co. hosted server and stored in a MySQL database. This data includes user accounts, user permissions, tournaments, and rankings. Email addresses and passwords will be transmitted over HTTPS to ensure against network snooping. PhoneGap will be used to port our initial web-based application to a semi-native application for phones. Notifications to users will be sent via e-mail for tournament invites and other tournament-related changes that the user is involved in.

4.5 Communications Interfaces

The communication methods used in the application include application notifications, and email. Any administrative position (Tournament Director, Tournament Moderators, Team Leaders) requires an account within the application. Players that are listed in a team can also have an account but there is no link between the team and the player. Email notifications will only occur when a user is invited to a tournament or a user is invited to be a moderator of tournament.

4.6 Domain Model

4.6.1 Level 0 Diagram

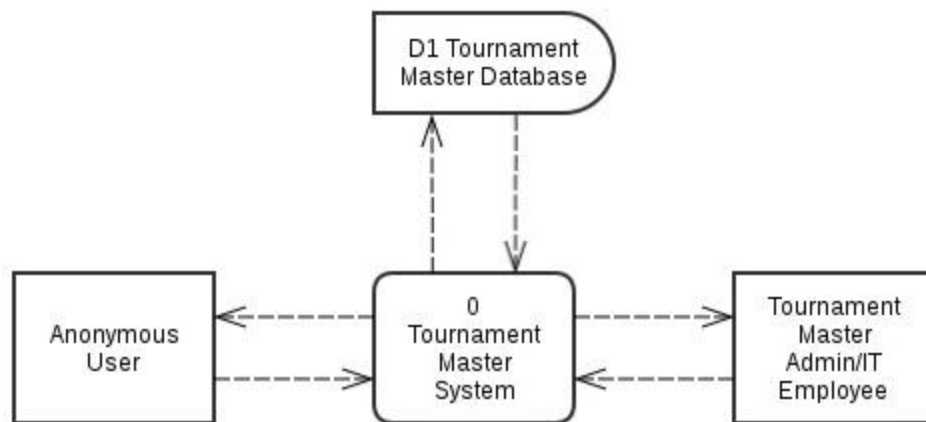


Figure 18: Level 0 Data Flow Diagram for Tournament Master

A high level view of the Tournament Master System. The only external entities are Tournament Master System Admins and the Users themselves as there are no payment details requiring outside interaction.

4.6.2 Level 1 Diagram

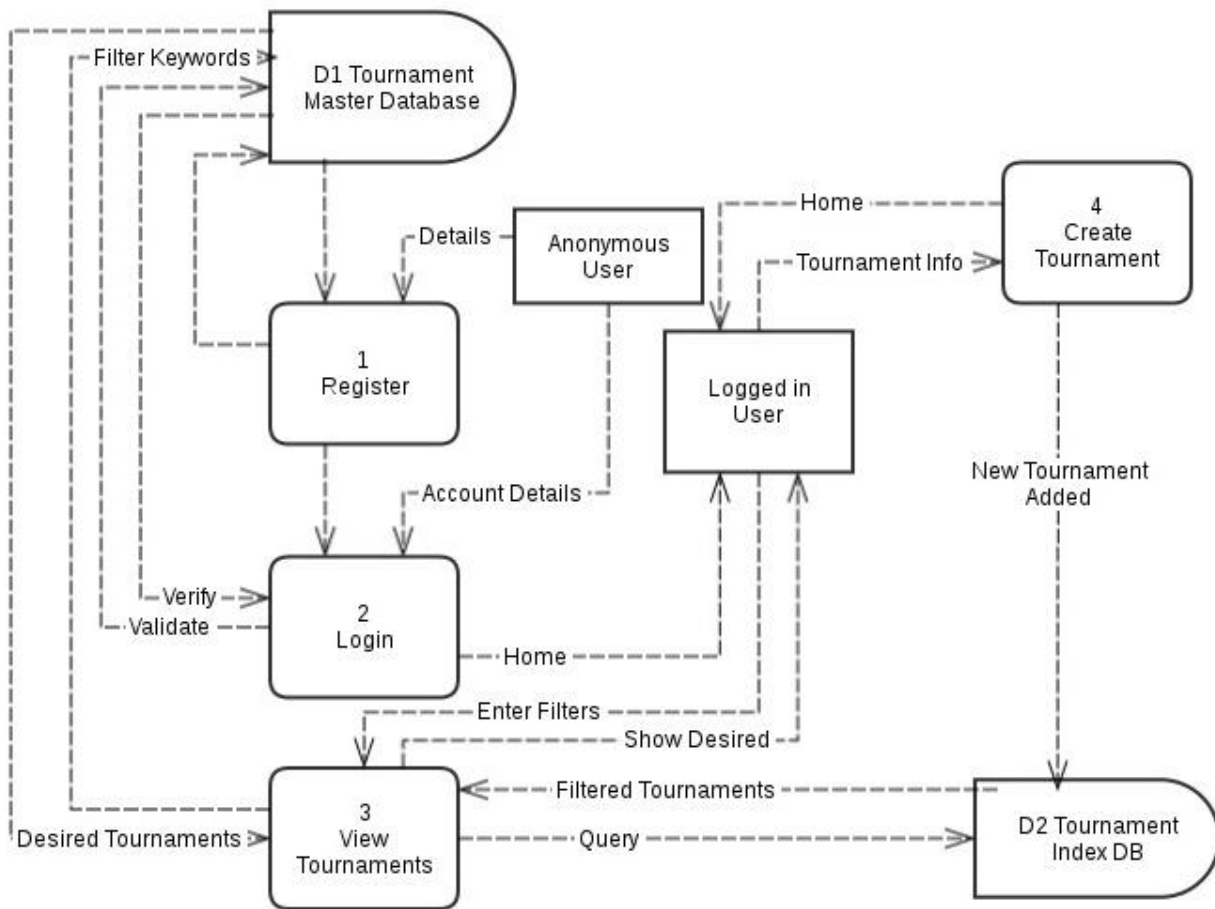


Figure 19: Level 1 Data Flow Diagram for Tournament Master

The Fundamental Functions of the system are the ability to register for the system, login to an account, view a tournament list, or create a new tournament. The data source D2 here is functionally equivalent to the D1 Tournament Master Database and has been duplicated for readability.

4.6.3 Level 2 Diagram

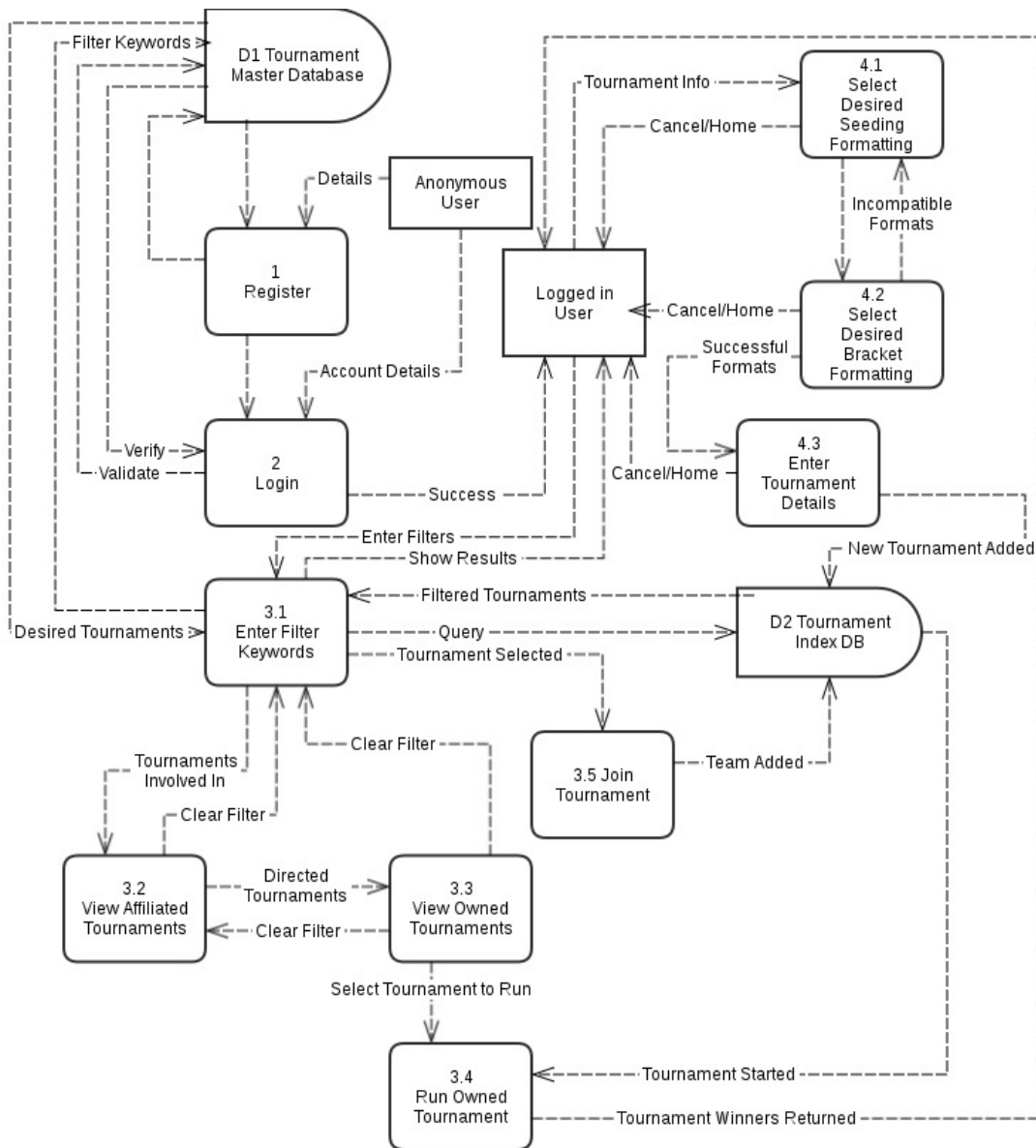


Figure 20: Level 2 Data Flow Diagram for Tournament Master

A closer look at the 3.0 View Tournaments and 4.0 Create Tournaments processes. Within the View Tournaments process is the ability to filter results to only the tournaments that a user is involved with (including only tournaments a user has created as a Director), the ability to join a tournament in the list, and the ability to run a tournament that was created as a Director. The Create Tournament Process includes selecting compatible Seeding and Bracket Formats and other relevant tournament information such as maximum number of teams competing and public or private designated status.

4.6.4 Entity-Relationship Diagram

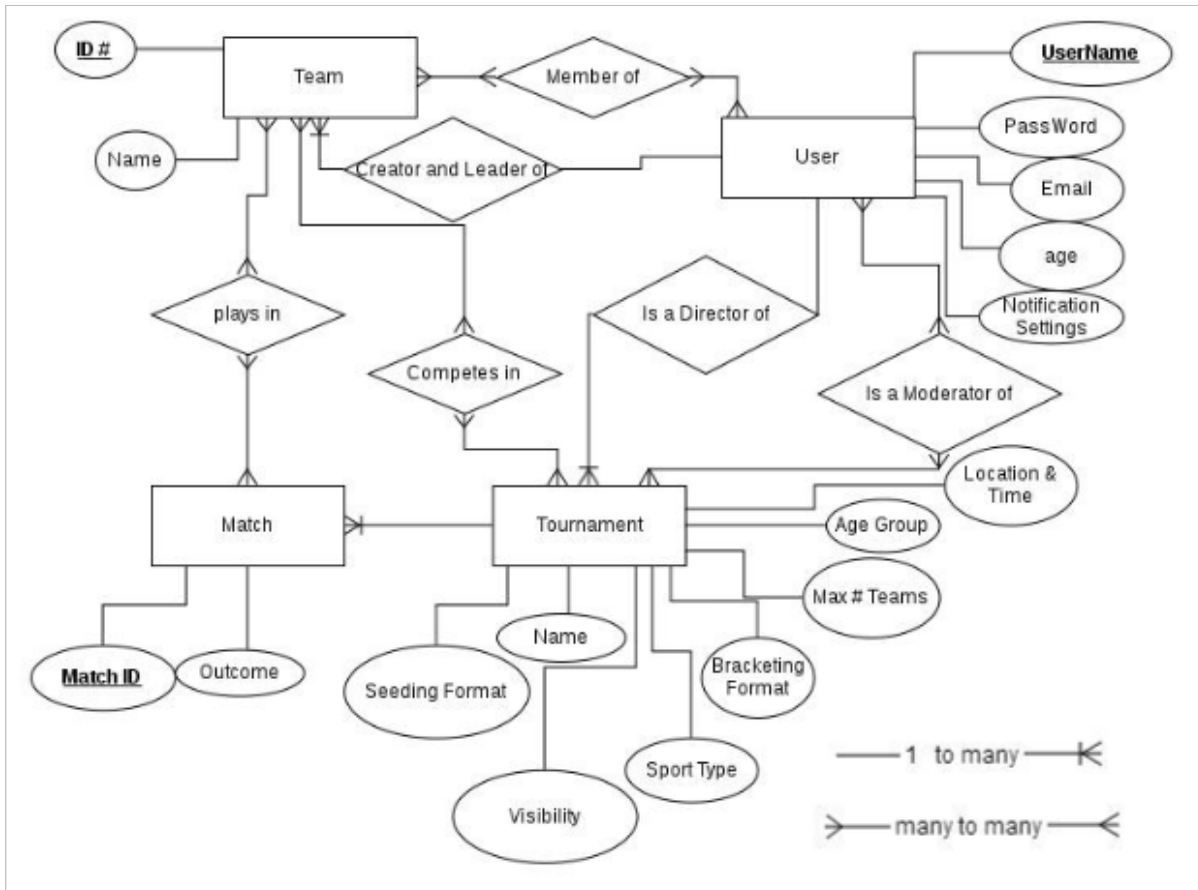


Figure 21: Entity-Relationship Diagram for Tournament Master

A user may be a director, moderator, team leader, or team member. A tournament may have multiple moderators but only a single director. Similarly, a team may only have one leader but can have many members. A team may play many matches within a tournament and a match may only belong to a single tournament.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

The application must be reasonably responsive across all supported devices. Users should not notice any delay when transitioning between screens, and loading times should be minimal. A cache and prefetched data will be used to give the appearance of increased network speed. When the user does not have an internet connection, the application will not function and display an error.

5.2 Safety Requirements

Tournament Co. will provide a disclaimer that will be shown on the application's first launch. The application will become available for use once the user agrees. The disclaimer will not be shown again unless it is changed. It will remove all liability that may be assigned to Tournament Co. for tournaments managed with the application.

5.3 Security Requirements

Login and registration requires an encrypted connection using a security certificate. Passwords will be cryptographically hashed in the database and are never displayed to the user. Email addresses will never be shared with other users without explicit permission. Operations to do with sensitive information, such as authentication and passwords changes, will be transmitted over HTTPS. All email notifications will have the option of being disabled from the settings screen.

5.4 Software Quality Attributes

Tournament Co. is a startup company that plans to use this application as a foundation to build on, so the application should be adaptable and robust. Planned expansions in version 2.0 of this application include:

- A tournament history for each player
- Pre-configured teams
- Integrated live streaming and chat
- More tournament types
- Player statistics (wins/losses, profile picture, etc.)
- Integration with social media

These features will be kept in mind when planning and designing the application. Tournament Co. is not sure of the product's future direction, which means the application has to be extensible for modifications and large upgrades. As such, the system allows for the expansion user tiers, which could include coaches, captains, or referees. The abstract architecture allows for additional sport types to be easily added.

Described in section 4.2, the application will display an error message when there is no internet connection.

Appendix A: Analysis Models

A.1 First Elicitation Meeting Minutes

Date: January 22, 2015

Time: 10:00 AM

Place: ECS 104

Present: Sarah, Jakob

Recorder: Sarah, Jakob

1. Tournament Specifics

- Tournament formats
- Tournament scope (commercial vs intramural)
- Participant limits
- Disqualification parameters

2. Clarifications

- Hierarchy of privileges, directors can raise moderators
- Private tournament access
- Private vs. public declarations
- Compatible platforms
- Tournament searching/browsing

3. User Classes and Privileges

- Director
- Leader
- User/player
- Guest

A.2 RS1.0 Elicitation Meeting Minutes

Date: February 26, 2015

Time: 10:00AM

Place: ECS 104

Present: Sarah, Jakob

Recorder: Sarah, Jakob

1. Tournament Specifics

- specify the public/private rules for joining and viewability
- notifications for invite only
- static team rosters
- app should not function if there is no data connection

- safety requirements: have a single general disclaimer that removes all liability from the app based on injury etc..
- fix picture and add information for unregistered users
- tournaments lock only once the results of the first match of the tournament has starte
- make N/A a seeding option on a pure round robin tournament

2. Clarifications

- 3.6.3 REQ-1: only if the tournament hasn't started
- 3.6.1: homepage for viewing tournaments
- 3.4.1 REQ-1: mention moderator removal feature, change add to invite
- 1.2: remove "The application will handle tournament creations, initial seeding, and bracketing." not covering all of scope
- 3.5: list and explain how system handles byes

A.3 RS2.0 Elicitation Meeting Minutes

Date: February 26, 2015

Time: 10:00AM

Place: ECS 104

Present: Sarah, Jakob

Recorder: Sarah, Jakob

1. Clarifications

- 3.2.4: change order of use cases 1 and 2
- 3.3.4: change main flow in use case to assign start date and time in UI
- 3.3.4: remove minimum player counts, forfeit if not enough
- 3.4.4: use case 2, can only join public tournaments without an invite
- 3.4.4: use case 2, remove alt flow B about player count
- 3.6.3 REQ-3: fix header
- 3.8.3 REQ-2: elaborate seed locking requirements
- 3.8.3 REQ-4: random seeding to be modified manually if required
- 3.10.4: add requirement to re-open tutorial interface
- 3.11.4: use case 1, add main flow step for password confirmation
- 3.11.4: add "signed in" precondition
- 5.2: create a more general disclaimer that does not speak of disclaimer contents
- add forfeit option to UI