



## Group Assignment 3

T-103-STST, Discrete Mathematics for Engineers  
Reykjavík University — Department of Computer Science, Reykjavík, Iceland

Lecturer: Ilham Qasse

Deadline: Thursday, October 30th, 2025, at 23:59

The goal of this project is to propose, interrogate, design, create a bookkeeping system for hosting and managing an e-sport tournament event, followed by user testing of the software. The software allows organizers to create and manage tournaments and input match results.

# Requirement List

## 0.1 Functional Requirements

Nr	Functional requirements	User group	Priority	Completed
1	The user can login as organizer	Organizer	A	
2	The user can register as a player	Player	A	
3	The user can login as a player	Player	A	
4	The user can enter as a spectator without signing in	General	A	
5	Organizer can create a tournament	Organizer	A	
6	Organizer can add a team to a tournament	Organizer	A	
7	When a team captain leaves a team, he assigns a team member to be the new captain	Team Captain	A	
8	The Organizer archives tournaments when they finish	Organizer	A	
10	Organizer can publish a tournament	Organizer	A	
11	Organizer can input match results	Organizer	A	
12	Player can create a team	Player	A	
13	Player becomes team captain	Player	A	
14	Player can leave a team	Player	A	
15	Team captain can add a player to his team	Team Captain	A	
16	Team captain can remove a player from his team	Team Captain	A	
17	Team captain can leave his team	Team Captain	A	
18	The user can view tournament	General	A	
19	The user can view tournament schedule	General	A	
20	The user can view all teams	General	A	
21	The user can view all players	General	A	
22	Organizer can manage tournament	Organizer	A	
23	Organizer can create a club	Organizer	B	
24	When the organizer creates a tournament, he chooses between double elimination and normal	Organizer	B	
25	Teams are assigned to a club on creation	Team Captain	B	
26	The user can view all clubs	Player	B	
27	Organizer can modify time of a tournament before it has started	Organizer	B	
28	Organizer can delete a tournament before it has been published	Organizer	C	
29	Organizer can edit clubs	Organizer	C	
30	Team captains can transfer his authority to a player in the team	Team Captain	C	
31	Team captain can change what club the team is associated with	Team Captain	C	
32	Organizer can cancel an ongoing tournament	Organizer	C	
33	Team captain can add ascii art to his team	Team Captain	C	
34	Organizer can edit specific match scheduling	Organizer	C	
35	The user can type in shortcuts for windows to travel to specific window	General	C	

## 0.2 Non-functional Requirements

Nr	Non-Functional requirements	User group	Priority	Completed
1	System automatically shuffles and arranges teams in a tournament match, based on quantity of teams and time frame	System	A	
2	When team captain leaves a team if he is alone then the team is archived	System	A	
3	Each Player can only be in one team at the same time	System	A	
4	The system should not crash due to invalid data	System	A	
5	The player personal data should not be visible to the spectator	System	A	
6	Only the team captain can modify the team	System	A	
7	Only the organizer can create tournaments	System	A	
8	Code must follow Python type hinting everywhere	System	A	
9	Code must follow snake caseing naming convention	System	A	
10	Name and Username should be the lenght of 3-30	System	A	
11	Tournament have must have 16 teams	System	A	
12	Player, Team, Tournament, Servers, Matches, Clubs generate UUID when stored in file	System	A	
13	The user can see a breadcrumb navigation	System	B	
14	Only the organizer can modify clubs	System	B	
15	Give users options to cancel while inputting information	System	B	
16	Tournament can have more than 16 teams	System	B	
17	The Club Name will be printed in its color	System	C	
18	Show tournaments brackets visually	System	C	

# Use Cases

User logs in as organizer	
Number	1
Priority	A
Precondition	Program is running
Base flow	<ol style="list-style-type: none"> <li>1. User selects log in</li> <li>2. Inputs the details for the Organizer           <ul style="list-style-type: none"> <li>• Organizer username</li> </ul> </li> </ol>
Alternative flow	User is now an Organizer and can create and manage tournaments.
Post condition	User/Organizer
Actors	
Author	Elmar Sigmarsson

Organizer can create a tournament	
Number	5
Priority	A
Precondition	Logged in as organizer, use case(1)
Base flow	<ol style="list-style-type: none"> <li>1. Organizer selects Create Tournament</li> <li>2. Organizer inputs tournament information           <ul style="list-style-type: none"> <li>• Unique name</li> <li>• Start date</li> <li>• End date</li> <li>• Start time frame</li> <li>• End time frame</li> <li>• Max servers</li> <li>• Venue</li> <li>• Contact email</li> <li>• Contact number</li> </ul> </li> </ol>
Alternative flow	3. User is now registered as a player
Post condition	Organizer has now created and has access to a new unique tournament
Actors	Organizer
Author	Elmar Sigmarsson

User registers as player	
Number	2
Priority	A
Precondition	Program is running
Base flow	<ol style="list-style-type: none"> <li>1. User selects Sign up</li> <li>2. User inputs his information           <ul style="list-style-type: none"> <li>• Full name</li> <li>• Date of birth</li> <li>• Address</li> <li>• Phone</li> <li>• Email</li> <li>• Link (optional)</li> <li>• Handle</li> </ul> </li> </ol>
Alternative flow	User is now registered as a player <ul style="list-style-type: none"> <li>• User tries to sign up with an existing account. User needs to login to his account. See use case(3)</li> </ul>
Post condition	User is now a player and cannot be registered as a player again.
Actors	User/Player
Author	Elmar Sigmarsson

Organizer adds team to tournament	
Number	6
Priority	A
Precondition	Organizer has already created a tournament, use case (5)
Base flow	<ol style="list-style-type: none"> <li>1. Organizer selects tournament</li> <li>2. Organizer selects View Teams.</li> <li>3. Organizer selects add team</li> <li>4. Organizer inputs team name           <ul style="list-style-type: none"> <li>• No team has been made, a player needs to make a team, use case (10)</li> <li>• The team doesn't have enough players, the team captain needs to add players to his team, use case (11)</li> </ul> </li> </ol>
Alternative flow	That team is now apart of the tournament
Post condition	Organizer
Actors	
Author	Elmar Sigmarsson

A player logs in	
Number	3
Priority	A
Precondition	Program is running
Base flow	<ol style="list-style-type: none"> <li>1. Player selects log in</li> <li>2. Player inputs his Handle</li> <li>3. Player is now logged in           <ul style="list-style-type: none"> <li>• User doesn't have a player account. User needs to sign up as a player. Use case(2)</li> </ul> </li> </ol>
Alternative flow	The player is now logged in and can access information about himself
Post condition	Player
Actors	
Author	Elmar Sigmarsson

Organizer removes a team from a tournament	
Number	7
Priority	A
Precondition	A team has to have been added to the Tournament, use case (6)
Base flow	<ol style="list-style-type: none"> <li>1. Organizer selects tournament</li> <li>2. Organizer selects view team</li> <li>3. Organizer selects remove team</li> <li>4. Organizer inputs team name</li> </ol>
Alternative flow	That team is no longer apart of the tournament
Post condition	Organizer
Actors	
Author	Elmar Sigmarsson

User enters as spectator	
Number	4
Priority	A
Precondition	Program is running
Base flow	<ol style="list-style-type: none"> <li>1. The user selects the Spectator option</li> <li>2. Now the user is a spectator</li> </ol>
Alternative flow	The user is now a spectator and can view details from the tournament (schedule, match results, clubs, teams and players)
Post condition	User
Actors	
Author	Elmar Sigmarsson

Organizer starts a tournament	
Number	8
Priority	A
Precondition	The tournament needs to have enough teams.
Base flow	<ol style="list-style-type: none"> <li>1. Organizer is logged in, case (1)</li> <li>2. Organizer selects manage tournaments</li> <li>3. Organizer inputs a tournament name</li> <li>4. Organizer is now able managing a tournament           <ul style="list-style-type: none"> <li>• Tournament doesn't have enough teams in it, organizer needs to add teams to the tournament, use case(6).</li> </ul> </li> </ol>
Alternative flow	The organizer is now viewing the tournament and its info and can edit/manage it (publish, manage teams in the tournament, change name/date)
Post condition	Organizer
Actors	
Author	Elmar Sigmarsson

Organizer can publish a tournament	
Number	9
Priority	A
Precondition	The tournament needs to have 16 teams
Base flow	<ol style="list-style-type: none"> <li>Organizer selects a tournament to manage, use case (8)</li> <li>Organizer selects publish tournament</li> <li>And confirms <ul style="list-style-type: none"> <li>Tournament doesn't have enough teams in it, organizer needs to add teams to tournament, use case (6)</li> </ul> </li> </ol>
Alternative flow	<ul style="list-style-type: none"> <li>Needs post condition!!!</li> </ul>
Post condition	Needs post condition!!!
Actors	Organizer
Author	Elmar Sigmarsdóttir

Team captain removes a player from their team.	
Number	13
Priority	A
Precondition	Team captain must have added a player to their team, use case(11)
Base flow	<ol style="list-style-type: none"> <li>Captain logs in, use case (3)</li> <li>Captain selects view team</li> <li>Captain selects remove player</li> <li>Captain inputs Player's handle</li> <li>Player has been removed from the team</li> </ol>
Alternative flow	<ul style="list-style-type: none"> <li>That player removes himself from the team, use case (14)</li> </ul>
Post condition	Team no longer has That Player in their team
Actors	Team captain, Player
Author	Elmar Sigmarsdóttir

Organizer inputs result of a match	
Number	10
Priority	A
Precondition	A tournament needs to have been published
Base flow	<ol style="list-style-type: none"> <li>Organizer selects a tournament to manage, use case (8)</li> <li>Organizer selects input match results</li> <li>Organizer gets a list of all matches and selects matches to input results</li> <li>Organizer is asked who of the two teams won the match and inputs the team name</li> <li>Player is now team captain of the newly created team.</li> </ol>
Alternative flow	
Post condition	Results of a specific match have been made so the winner of that match goes to the next round
Actors	Organizer
Author	Elmar Sigmarsdóttir

Player leaves their team	
Number	14
Priority	A
Precondition	Player needs to be in a team.
Base flow	<ol style="list-style-type: none"> <li>Player logs in, use case (3)</li> <li>Player selects view team</li> <li>Player selects leave team</li> <li>(Error message/ confirmation)</li> <li>Player is no longer a part of any team</li> </ol>
Alternative flow	<ul style="list-style-type: none"> <li>Team captain removes player from team, case (12)</li> </ul>
Post condition	The player is no longer a part of any team and can now be picked onto a new team or create his own
Actors	Player
Author	Elmar Sigmarsdóttir

Player creates a team	
Number	11
Priority	A
Precondition	
Base flow	<ol style="list-style-type: none"> <li>Player logs in, use case (3)</li> <li>Captain selects Add player to team.</li> <li>Player inputs team name</li> <li>Player inputs Club (no input = no club)</li> <li>(Now the player is the team captain of an empty team)</li> </ol>
Alternative flow	<ul style="list-style-type: none"> <li>Player is already a part of a team. Player needs to leave his existing team, see case (14) and then create a team</li> </ul>
Post condition	Player is now a team captain of a team
Actors	Player
Author	Elmar Sigmarsdóttir

Team captain leaving his team.	
Number	15
Priority	A
Precondition	Player needs to be a team captain.
Base flow	<ol style="list-style-type: none"> <li>Captain logs in, use case (3)</li> <li>Captain selects view team</li> <li>Captain selects leave team</li> <li>(Error message/ are you sure)</li> <li>Captain selects a Player from team to become new Captain</li> <li>The Captain has left the team</li> </ol>
Alternative flow	<ul style="list-style-type: none"> <li>If the team the Captain is leaving has no other players the team is archived, but still has its statistics (wins, etc.)</li> </ul>
Post condition	The Captain is no longer a part of the team and is now has the status of just Player (not in team), and the player he selected has the status team Captain of that team
Actors	Captain, player
Author	Elmar Sigmarsdóttir

Team captain adds a player to their team	
Number	12
Priority	A
Precondition	Player must have created a team, use case(10)
Base flow	<ol style="list-style-type: none"> <li>Captain logs in, use case (3)</li> <li>Captain selects My Team</li> <li>Captain selects Add Player to Team</li> <li>Captain inputs Players Handle</li> <li>That Player is now a part of team</li> </ol>
Alternative flow	<ul style="list-style-type: none"> <li>A player is already in another team. That player needs to leave his team, see case (14), then the Captain can add that player to his team</li> </ul>
Post condition	Team captain has now added a player to his team
Actors	Team captain, Player
Author	Elmar Sigmarsdóttir

Any user can see tournaments	
Number	16
Priority	A
Precondition	
Base flow	<ol style="list-style-type: none"> <li>User selects Tournaments</li> <li>User selects which tournament</li> <li>User is now viewing a specific tournament</li> </ol>
Alternative flow	<ul style="list-style-type: none"> <li>There are no tournaments available, the organizer needs to create a tournament, use case (5)</li> <li>A tournament is not published, so the organizer needs to start the tournaments, use case (9)</li> </ul>
Post condition	User can now look through that tournament
Actors	Any user
Author	Elmar Sigmarsdóttir

Any user can see tournament schedule	
Number	17
Priority	A
Precondition	
Base flow	<p>1. User selects a tournament</p> <p>2. User selects Schedule</p> <p>3. User is now viewing the schedule of a tournament</p>
Alternative flow	User can view the schedule of the next round of that tournament (or whole tournament, date)
Post condition	
Actors	Any user
Author	Elmar Sigmarsson

Organizer can edit time frame of tournament	
Number	22
Priority	B
Precondition	Tournament has to created and NOT started
Base flow	<p>1. Organizer selects a tournament to manage, use case (8)</p> <p>2. Organizer selects Edit Time Frame</p> <p>3. Organizer input new:</p> <ul style="list-style-type: none"> <li>• Start date</li> <li>• End date</li> </ul> <p>4. Organizer selects Edit Time Frame</p>
Alternative flow	
Post condition	The Time frame has been updated of that tournament
Actors	Organizer
Author	Elmar Sigmarsson

Any user can view teams	
Number	18
Priority	A
Precondition	
Base flow	<p>1. User selects teams (from main menu)</p> <p>2. User is viewing a list of all teams</p> <ul style="list-style-type: none"> <li>• User can select a tournament see use case (XXX), and in that menu select teams and then the user is viewing only the teams in that tournament</li> </ul>
Alternative flow	<ul style="list-style-type: none"> <li>• User can select a tournament see use case (XXX), and in that menu select teams and then the user is viewing only the teams in that tournament</li> </ul>
Post condition	The user is viewing a list of teams that he can select from see that teams information (players, statistics)
Actors	Any user
Author	Elmar Sigmarsson

Organizer can delete a tournament	
Number	23
Priority	B
Precondition	A tournament can NOT be published
Base flow	<p>1. Organizer selects a tournament to manage, use case (8)</p> <p>2. Organizer selects Delete</p> <p>3. Organizer confirms deletion</p>
Alternative flow	
Post condition	Now the tournament has been deleted and no records of it are left
Actors	Organizer
Author	Elmar Sigmarsson

Any user can view players	
Number	19
Priority	A
Precondition	
Base flow	<p>1. User selects players (in main menu)</p> <p>2. User is viewing a list of all registered players</p>
Alternative flow	<ul style="list-style-type: none"> <li>• User can select a tournament see use case (XXX), and in that menu select players and then the user is viewing only the players in that tournament</li> <li>• User selects a team, use case (XX), and from the team menu the user can view a list of that teams player</li> </ul>
Post condition	The user is viewing list of players that he can select from to see that players information (handle, statistics)
Actors	Any user
Author	Elmar Sigmarsson

Player can edits their information.	
Number	24
Priority	B
Precondition	
Base flow	<p>1. Player logs in, use case (3)</p> <p>2. Player selects Edit Information</p> <p>3. Player selects the info he wants to edit (all info)</p> <p>4. Player has changed his info</p> <p>3. Player has changed their information.</p>
Alternative flow	
Post condition	Player has updated their information.
Actors	Player
Author	Elmar Sigmarsson

Organizer can create clubs	
Number	20
Priority	B
Precondition	
Base flow	<p>1. Organizer selects View Clubs.</p> <p>2. Organizer selects add club.</p> <p>3. Organizer inputs information.</p> <ul style="list-style-type: none"> <li>• Name</li> <li>• Club color</li> <li>• Hometown</li> <li>• Country</li> </ul>
Alternative flow	Organizer has added a new club.
Post condition	
Actors	Organizer
Author	Elmar Sigmarsson

Team Captain can make team Player a Captain	
Number	25
Priority	B
Precondition	
Base flow	<p>1. Player logs in, use case (3)</p> <p>2. Captain selects My Team</p> <p>3. Captain selects transfer authority</p> <p>4. Captain inputs players' name to transfer authority to</p>
Alternative flow	
Post condition	Now that Player is the Team Captain, and the old Team Captain is just a Player on the team
Actors	Captain, Player
Author	Elmar Sigmarsson

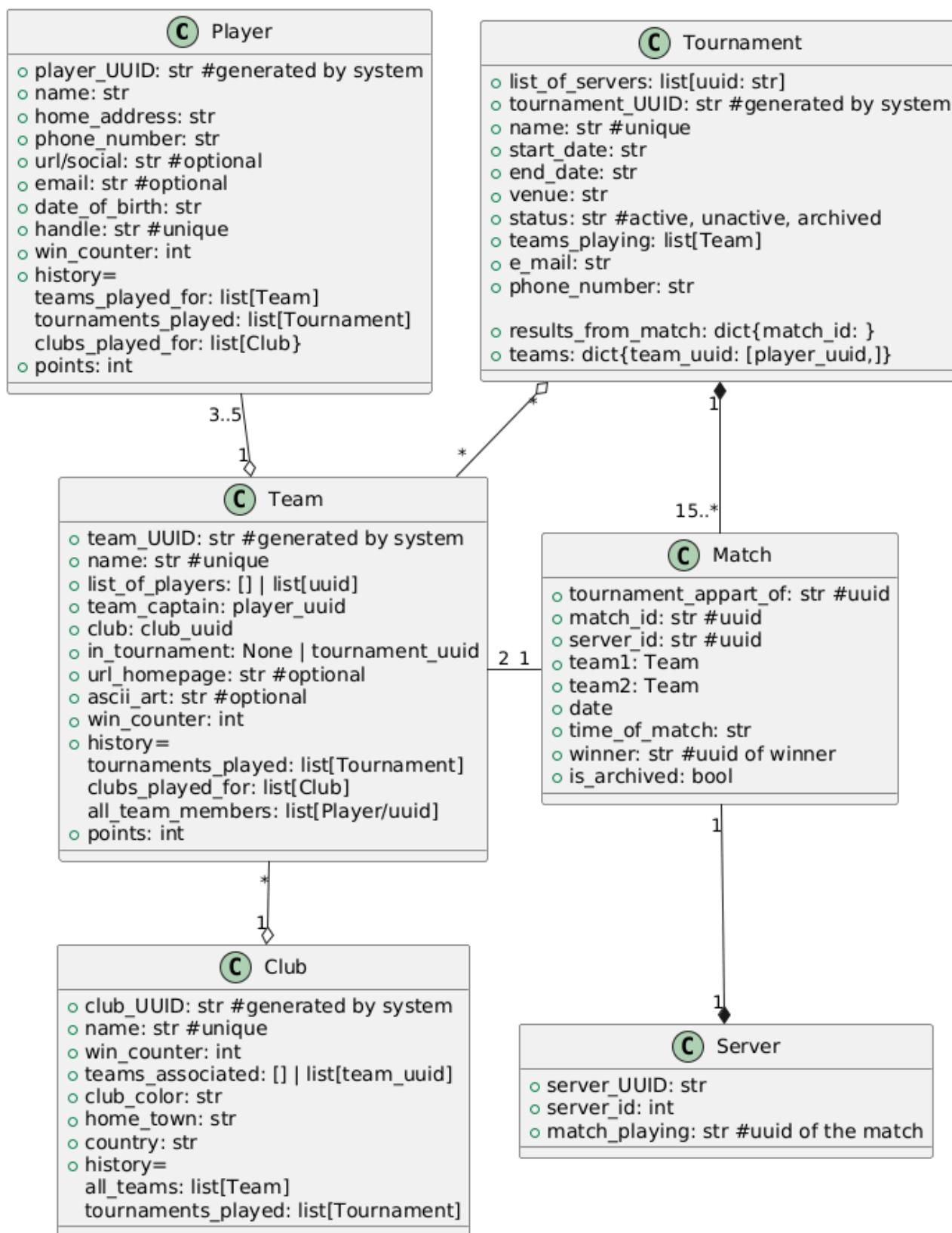
Organizer can choose type of tournament when creating one	
Number	21
Priority	B
Precondition	
Base flow	<p>1. While Organizer is creating a tournament, use case (5)</p> <p>2. Now there is an extra section which asks the organizer what type of tournament it is (knockout or double elimination)</p>
Alternative flow	Organizer has added selected what type of tournament
Post condition	
Actors	Organizer
Author	Elmar Sigmarsson

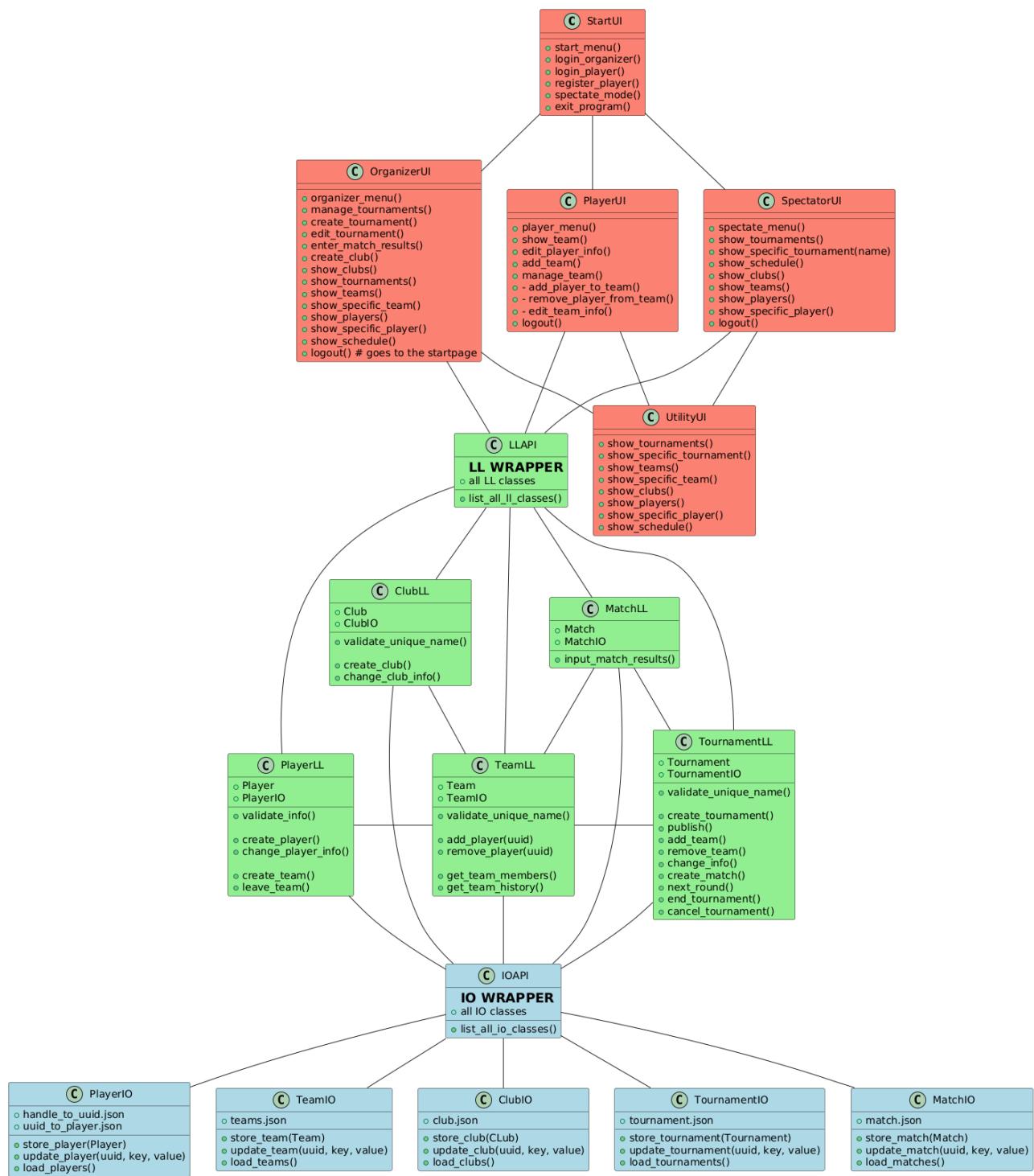
Any user can view clubs	
Number	26
Priority	B
Precondition	
Base flow	<p>1. User selects spectator</p> <p>2. User selects clubs</p>
Alternative flow	User is now viewing a list of clubs
Post condition	
Actors	Any user
Author	Elmar Sigmarsson

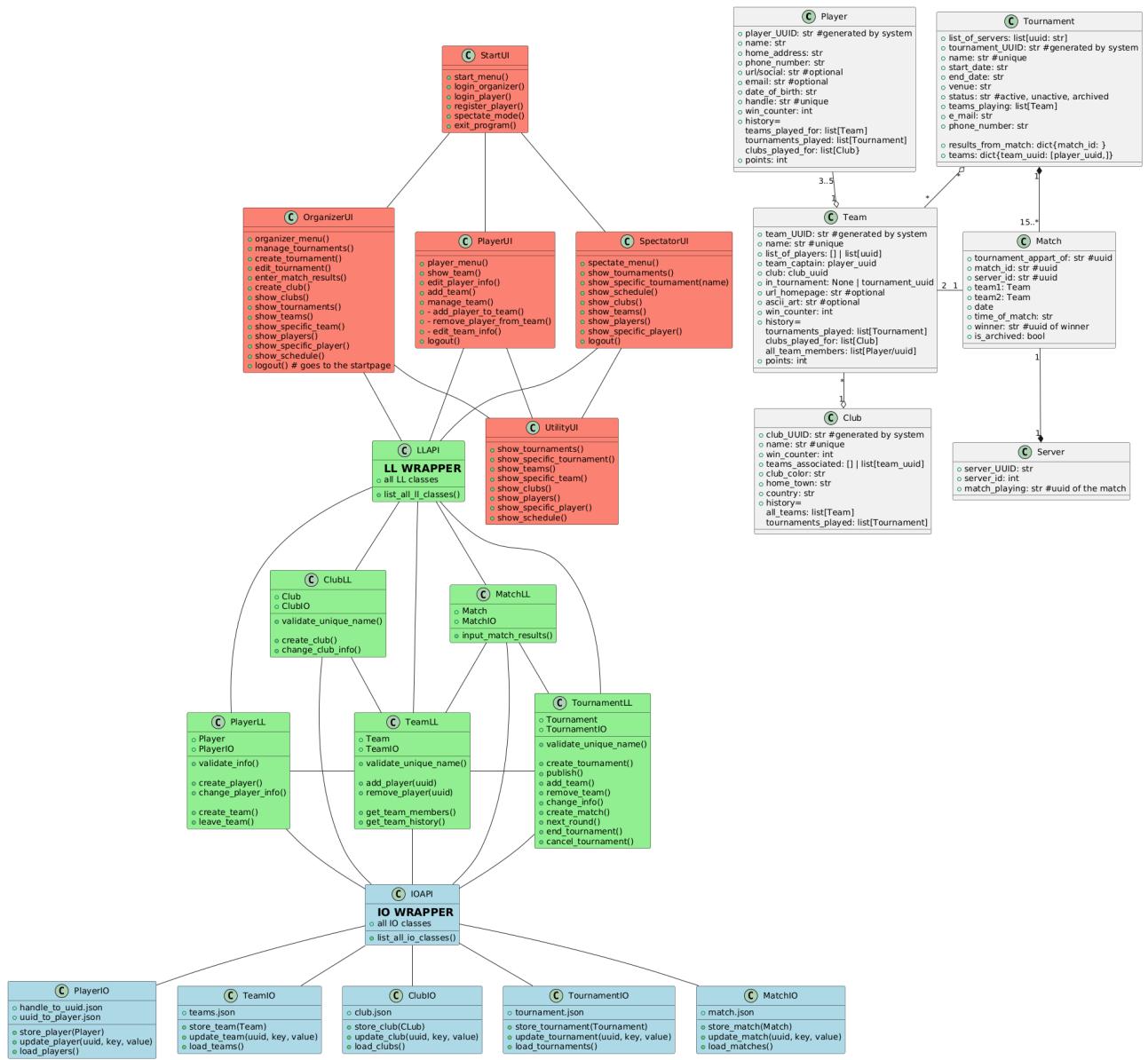
Organizer can cancel a tournament		Team captain can change the teams club	
Number	27	Number	29
Priority	C	Priority	C
Precondition	A tournament has to have been published	Precondition	
Base flow	1. Organizer selects a tournament 2. Organizer selects Cancel tournament 3. Tournament is now canceled	Base flow	1. Team Captain is logged in, and selects View Team 2. From View Team menu Captain selects Change Club 3. Captain has a list of the clubs and can select what his teams new club is
Alternative flow		Alternative flow	
Post condition	The tournament is cancelled, all records are not kept, the statistics for the teams is reverted to the state before the tournament started	Post condition	Now the Captains team is in a new club, so the team is listened in the their new club and no longer in the old club
Actors	Organizer	Actors	Team Captain
Author	Elmar Sigmarsen	Author	Elmar Sigmarsen
Team captain can edit team name		Team Captain can add ASCII art to his team (logo)	
Number	28	Number	30
Priority	C	Priority	C
Precondition		Precondition	
Base flow	1. Captain is logged in, and selects View Team 2. From View Team menu Captain selects Change Team Name 3. Captain inputs the new team name (has to be unique)	Base flow	1. Captain is logged in, and selects View Team 2. Captain selects add ASCII art 3. Captain inputs the ASCII art
Alternative flow		Alternative flow	
Post condition	Now that teams name has a new name, and new or other teams cannot use that name, but can use the previous team name	Post condition	Now the team has ASCII art that can be viewed from any user viewing the team
Actors	Team Captain	Actors	Team Captain
Author	Elmar Sigmarsen	Author	Elmar Sigmarsen

# Class Diagrams

Class diagrams are the main building block in object-oriented modeling. They are used to show the different objects in a system, their attributes, their operations and the relationship between them. The two class diagrams are with model classes and 3 tier design classes. Model classes are a class (bucket) that is just the data, with few functions. While the 3-tier design splits the classes into the user interaction layer, logic layer and data layer, which are connected by so called wrappers that make it so that each layer doesn't impact the rest of the layers to make it easier refactor or change classes or functions without the other layers getting affected. The diagrams below show model classes, 3-tier design and model classes combined with 3-tier design.







# User Group Analysis

Team sheet		
Goalkeeper	GK	Paul Robinson
Defenders	LB	Lucas Radebe
	DC	Michael Dubarry
	DC	Dominic Matteo
	RB	Didier Domi
Midfielders	MC	David Batty
	MC	Eirik Bakke
	MC	Jody Morris
Forward	FW	Jamie McMaster
Strikers	ST	Alan Smith
	ST	Mark Viduka

Name of group	Player	Spectators
---------------	--------	------------