DESIGN REPORT - 1

Alexander Björnsson, Ernir Elí Ellertsson, Ívar Orri Tómasson
Nökkvi Benediktsson, Olgeir Otri Engilbertsson

HÁSKÓLINN Í REYKJAVÍK
REYKJAVÍK UNIVERSITY

26 NOVEMBER 2025

T-113-VLN1

Dr. Gylfi Þór Guðmundsson

Contents

Introduction

The premise of this design report is to get the basic wireframe of our program mapped, such as the requirements, use cases, state diagram and class diagram. The program itself is an e-Sports tournament maker managing a tournament and recording all the events of the tournament such as results, winners etc.

Requirements

- We noted down every requirement for our system for it to run, classified by severity from A to C. They are also classified by Functional and Nonfunctional requirements.
- A requirements is a list of all the requirements needed for the system to run at its core and features that must come forward in the final product such as organizers being able to create tournaments with the needed information to make it
- B requirements on other hand is a list of features that are not necessarily needed for the program but nice additions to have for more data and sophisticated results
- C requirements are mostly made up of our own ideas that are seemingly hard to implement therefore come as the least important requirement as its not necessarily to be implemented, note if any C requirements are to be implemented then all or most B requirements must already be fulfilled.

A requirement:

Functional requirements:

Number	Name (and possibly a short description)	User group(s)	Priority (A/B/C)	Additional info
Players:				
A1	Store player personal data	Organizer, Captain	A	Name, date of birth, home address, phone number, email, link, handle & team
A2	Captains can modify player information	Captain	A	modify phone number, email, address, links & handle
A3	Organizer can modify player information	Organizer	A	modify phone number, email, address, links & handle
Teams:				

A4	System can store team data	System	A	Unique name, One captain, Link to web page, ASCII logo
A5	Captain can manage players in their team	Captain	A	Add/change player information
A6	Teams created once	Organizer	A	Same team can only be created once
Organizer:				
A7	Organizer can register new teams	Organizer	A	
A8	Organizer can create tournaments	Organizer	A	Only one game per tournament
Tournament Data:				
A9	Stored tournament information	System	A	Start date, End date, Unique name, Venue, Contract, Contact person (email + phone)
A10	Create game schedule automatically	System	A	Knockout style: generated after all data is registered
Games And Results:				
A11	Organizer can input match results	Organizer	A	

A12	System updates schedule after results	System	A	Completed games show results; future matches update
Information retrieval:				
A13	Anyone can view game schedule	Users	A	Completed games show results; future games show team + time
A14	Anyone can view tournaments per team	Users	A	
A15	Anyone can view tournaments per player	Users	A	
A16	Anyone can view team rosters	Users	A	Only see names + handles; no private info
A17	Lock schedule after creation	System	A	Teams and players cannot be charged
A18	Only organizers or captains can see player private data	Organizer, Captain	A	

Nonfunctional requirements:

Number	Name (and possibly a short description)	User group(s)	Priority (A/B/C)	Additional info
A19	For all inputs, user shall know what to input	User	A	Users should know what the input possibilities are, every time.

B Requirements

Functional requirements:

Number	Name (and possibly a short description)	User group(s)	Priority (A/B/C)	Additional info
B1	Support double elimination	Organizer	B	Loser's bracket needed
B2	Allow teams to play multiple games at a tournament	Organizer	B	
B3	Support clubs (teams belong to clubs)	Organizer	B	Club has a name, colors, hometown, country
B4	Tournament reward points	System	B	Points for players, teams, clubs
B5	Player statistics	Users	B	Wins, earnings, clubs played for, tournaments played
B6	Team statistics	Users	B	Tournaments won, total wins, total earnings
B7	Club statistics	Users	B	Sort teams by earnings, wins, etc.
B8	Players can change team	Captain /		

Nonfunctional requirements:

Number	Name (and possibly a short description)	User group(s)	Priority (A/B/C)	Additional info
B9	Every UI shall be a formatted table	System	B	

C Requirements

Functional requirements:

Number	Name (and possibly a short description)	User group(s)	Priority (A/B/C)	Additional info
C1	New team created for each tournament	Organizer	C	Complicates statistics
C2	Process results files automatically	Organizer	C	Imports instead of manual entry
C3	Players can be in more than one team	System	C	
C4	Login requires player handle & password	System	C	

Nonfunctional requirements:

Number	Name (and possibly a short description)	User group(s)	Priority (A/B/C)	Additional info
C3	Club statistic page should have its colors.	System	C	The formatting on the club page should consist of the clubs' colors.

Use Cases

Our Use cases consist of every requirement and their corresponding priority, each use case is listed with a source, precondition, base flow, alternative flow post condition, actors and those who wrote the use case as authors.

- Precondition tells us what has to be true so the user can do the actions needed to reach their goal.
- Base flow shows a route the user takes to reach their goals including what they need to do on each step
- Alternative flow shows what could go wrong on the route such as an invalid input or a false statement
- Post condition is what happens after the user has reached their goals and what turns true
- Actors are the type of users going through the use case and getting to their goal

Use Cases 1 --- "Store player personal data"

Name:	Store player personal data
Number:	UC – 01
Priority:	A
Source	A1
Precondition:	The team the player belongs to already exists.
Base flow:	1. User picks role: Captain or Organizer 2. Selects "Add player". 3. The system displays a form for player information. 4. Organizer/ Captain enters name, date of birth, address, phone number, email, link, handle, and team.

	5. The system validates the data. 6. System stores the player and confirms the registration.
Alternative flow:	If something is wrong or missing (like a bad email or a missing name), the system shows an error and asks the user to fix it before saving.
Post condition	The player is added to the system with all their info. If something fails, nothing is saved.
Actors:	Organizer, Captain
Authors:	Nökkvi Benediktsson

Use Case 2 --- "Modify player info"

Use case name	Modify player
Number	UC – 02
Priority	A
Source	A2, A3
Precondition	User is Captain or organizer
Base flow	1. Select Team name 2. Selects the modifying option for their team. 3. The system prints out a list of all players to choose from. 4. User chooses a player which in turn gives a list with all his info 5. User chooses one of the following, e.g. email, phone number, etc.

Alternative flow	If User does a change on one of the players and leaves it empty the system stops him from exiting
Post condition	Changes are saved and updated to the database.
Actors	Captains
Authors	Alexander

Use case 3 --- “Store Team Data”

Use case name	Store Team Data
Number	UC – 03
Priority	A
Source	A4
Precondition	Organizer (or captain) Data layer is available Team name to be registered does not yet exist in the system
Base flow	1. Actor selects “Create new team/ “register team” in the UI. 2. System prompts for team information 3. Actor enters team name and optional fields 4. System checks that the team’s name is not already in use. 5. System prompt for 3-5 players. 6. System asks which of the players is the captain 7. User selects one of the listed players as captain. 8. system validates:

	a. Number of players is between 3 and 5. b. Team name is unique c. Each player's handle is unique in the whole system d. Captain is one of the team's players 9. If validation succeeds, logic layer sends the team object and its players to the Data layer 10. Data layer writes the team and player data to the file(s) 11. "Team successfully stored" 12. System returns to main menu or team overview
Alternative flow	1. Actor selects "Create new team/ "register team" in the UI 2. Actor enters team name 3. Team name already in use
Post condition	1. A new team record is registered 2. Unique team name 3. All data persisted to appropriate files(s)
Actor	Organizer / Team captain
Author	Ívar Orri Tómasson

Use case 4 ---" Captains can manage players on their team"

Use case name	Captains managing their players
number	UC – 04
priority	A
source	A5

Precondition	Captain has a team Captain can manage team
Base flow	1. Captain selects" Manage my team" from the menu 2. System displays team info (team name, captain, list of players with handles.) 3. Captain chooses one of the available actions. View player details, Edit player contact info (phone, address, email, link) 4. System shows players personal data 5. For edit: Captain edits fields like phone, address, email link. 6. For add player: Captain enters new player information (name, DoB, handle, etc.) 7. System checks that the handle is unique and the player is not already registered in another team. 8. System confirms the changes 9. Flow ends
Alternative flow	1. Team size limit reached 2. Captain tries to add player but then displays "Team already has the maximum number"
Post condition	1. Team data and player personal data are updated as requested 2. No player has been moved to a different team

	3. No player belongs to more than one team
Actors	Team Captain
Authors	Ívar Orri Tómasson

Use Case 5 --- “Organizers can add new teams”

Use case name	Organizer adds teams
Number	UC – 05
Priority	A
Source	A7
Precondition	User is Organizer
Base flow	1. Organizer clicks on „add team 2. Adds teams by writing in the name of said team 3. Adds players like captain adds players by name or handle 4. Team is added to the tournament
Alternative flow	Not enough players added, less than 3 makes you make more
Post condition	Changes are saved and updated to the database.
Actors	Organizers
Authors	Alexander

Use Case 6 --- “Organizers can create tournaments”

Use case name	Tournament Creation
Number	UC – 06
Priority	A
Source	A8
Precondition	User picks Organizer

Base flow	1. Organizer gets menu screen for organizers 2. Selects Create Tournament 3. Inputs all needed info for the tournament. Number of teams, Start date – End Date, Game type,
Alternative flow	Number of teams must be at least 16
Post condition	Changes are saved and updated to the database.
Actors	Captains
Authors	Alexander

Use Case 7 --- "Organizer handles schedule and match results"

Use case name	Organizer manages tournament schedule and results
Number	UC – 07
Priority	A
Source	A 10, A 11, A 12
Precondition	The Organizer is logged in. A tournament already exists. All teams for the tournament are registered.
Base flow	1. The Organizer logs into the system. 2. The Organizer selects a tournament from the list. 3. The Organizer chooses "Generate schedule". 4. The system checks if all required info is there and then creates the knock-out schedule.

	5. Later, when matches are played, the Organizer opens the same tournament again. 6. The Organizer selects a completed match from the schedule. 7. The Organizer enters the match result. 8. The system saves the result and updates the schedule with new information.
Alternative flow	If something is missing (like not enough teams or dates), the system stops the schedule creation and tells the Organizer what needs fixing.
Post condition	A full schedule exists, and all completed matches have correct result stored. The backed or schedule is updated after each match result
Actors	Organizer, System (automatically generating and updating schedule)
Authors	Nökkvi Benediktsson

Use case 8 ---" View Game Schedule"

Use case name	View Game Schedule
number	UC – 08
Priority Source	A A13

Precondition	1. Tournament and match data exist in the 2. system stored match result where applicable 3. User does not need to be logged in
Base flow	1. User chooses “View game schedule” in the menu 2. System asks the user which tournament to view (if there are multiple) 3. User selects a tournament 4. System shows a list of all matches in that tournament; For completed matches show teams and result., and for future matches show date and time 5. User finishes viewing and returns to the main menu
Alternative flow	1. In step 4, the system finds no matches for the selected tournament. 2. System shows the message “No Schedule available for this tournament” 3. User is returned to the previous menu
Postcondition	User sees a correct and up to date schedule Completed games show results Upcoming games show participating teams and date & time
Actors	All users
Authors	Ívar Orri Tómasson

Use Case 9 --- “View Tournaments for a Team”

Use Case Name	View Tournaments for a Team
Number	UC – 09
Priority	A
Source	A14
Precondition	Teams and tournaments are registered in the system User does not need to be logged in
Basic Flow	1. User Selects “View Team Information” from the menu 2. System displays a list of all registered teams 3. User selects a team 4. System shows a list of tournaments that the team has participated in 5. User finishes viewing and returns to the main menu
Alternative Flow	1. In step 4, system finds no tournaments for the selected team 2. System shows message. “This team has not participated in any tournaments” 3. User returns to the team list or main menu
Post condition	User sees the list of tournaments for a chosen team
Actors	All users
Authors	Ívar Orri Tómasson

Use case 10 --- “View tournament per player”

Use case name	View tournament per player
Number	UC – 10
Priority	A
Source	A15
Precondition	Players and tournaments are registered in the system User does not need to be logged in
Basic flow	1. User selects “View Player Information” From the menu 2. System displays a list of players (by handle) 3. User selects a player 4. System shows a list of tournaments that this player has participated in. 5. User finishes viewing and returns to the main menu
Alternative flow	Players have not played in tournament 1. In step 4, the system finds no tournaments for the selected player 2. System shows message. “This player has not participated in any tournaments 3. User returns to the player list menu
Postcondition	User has seen the list of tournaments for the chosen player Only the player’s handle is shown, not personal details
Actor	All Users
Author	Ívar Orri Tómasson

Use Case 11 --- “Anyone can view team roster”

Use case name	Anyone can view team roster
Number	UC – 11
priority	A
source	A17
Precondition	Teams and players are registered in the system
Basic flow	1. User selects “View Tean Information” from the menu 2. System shows a list of all registered teams 3. User selects team 4. System displays: Player handles for all players on team, Team name and captain handle 5. User returns to the main menu
Alternative flow	1. In step 4, the system finds that the selected team has no registered players 2. System shows: “This team has no players registered 3. User returns to the team list or main menu
Postcondition	User has viewed the team’s roster (only player handles, not personal data)
Actor	All users
Author	Ívar Orri Tómasson

Use case 12 --- “View player private data”

Use case name	View player private data

number	UC – 12
priority	A
Source	A18
Precondition	Actor is logged in as Organizer or Captain Players exist in the system with stored personal data
Basic flow	1. Actor selects “View player details” from the menu 2. System verifies that the actor is an Organizer or a captain 3. System shows a list of players (or lets actor search by handle) 4. Actor selects a player 5. System displays the player’s private data (e.g.name, address, phone, etc.) 6. Actor returns to the previous menu
Alternative flow	1. A user who is not an Organizer or Captain tries to access “View player details” or a player info screen 2. System checks user role and finds they are unauthorized 3. System does not show private data 4. System shows a message “Private player data is only visible to organizers and team captains.” 5. User is returned to the previous menu
postcondition	Actor has seen the player’s private data No unauthorized user has seen private data
Actor	Captain and organizer
Author	Ívar Orri Tómasson

Use case 13 --- “View player list”

Use case name	View player list
Number	UC – 13
Priority	A
Source	A13, A18
Precondition	Players exist in the system
Basic flow	1. User selects “View Players” from the menu 2. The system displays a list of all players and only handles and team. 3. User scrolls or searches through the list. 4. Users select a player to view more information. 5. User returns to the main menu.
Alternative flow	If no players are registered, the system shows “No players found.”
Post condition	User has seen a list of players.
Actors	All users
Authors	Nökkvi Benediktsson

Use case 14 --- “View team list”

Use case name	View team list
Number	UC – 14
Priority	A
Source	A 14, A 17
Precondition	Team exists in the system

Basic flow	1. User selects “View Teams” from the menu. 2. System displays a list of all registered teams. 3. User scrolls or searches the list. 4. Users select a team to view basic information. 5. User returns to the main menu.
Alternative flow	If no teams are registered, system shows “No teams available.”
Post condition	User has viewed the list of all teams.
Actors	All users
Authors	Nökkvi Benediktsson

Use case 15 --- “Support double elimination

Use case name	Support double elimination
Number	UC – 15
Priority	B
Source	B1
Precondition	Organizer is logged into the system Organizer has permission to create/edit tournaments The required tournament settings are available
Basic Flow	1. Organizer selects “Create tournament” or opens an existing tournament for editing 2. Organizer select “Double eliminations” as tournament type

	3. Organizer confirms number of teams and seeding 4. System generates winner and loser brackets 5. Organizer saves the tournament 6. System confirms that the double-elimination tournament is ready
Alternative flow	Number of teams is invalid system shows: "enter a valid team number" Bracket cannot be generated
Post condition	Tournament is configured as double elimination Winner's and loser's brackets are created and stored All teams are seeded into the correct bracket
Actors	Organizer
Authors	Ívar Orri Tómasson

Use case 16 --- "Support clubs (teams belong to clubs)"

Use case name	Support clubs (teams belong to clubs)
Number	UC – 16
Priority	B
Source	B3
Precondition	1. Organizer is logged in to the system. 2. There are some or teams.
Basic flow	1. Organizer creates a club. 2. Organizer selects all teams belonging to that club.
Alternative flow	1. Team doesn't exist or invalid team name. Organizer offered to fix mistake.

Post condition	Teams belong to a Club. Teams inherit club colors, hometown and country. Teams contribute to club stats
Actors	Organizer
Authors	Ernir Elí Ellertsson

Use case 17 --- “Award points “

Use case name	Award Tournament Reward points
Number	UC – 17
Priority	B
Source	B4
Precondition	1. Tournament exists and is completed 2. Points rules are defined 3. All participating players, teams and clubs are linked
Basic flow	1. Organizer opens the completed tournament 2. Organizer selects ”Calculate reward points” 3. System checks that results are final and points rules exist 4. System calculates points for: Each player, each team and each club 5. System stores the updated totals 6. 6. System confirms successful calculation and updates
Alternative flow	1. Tournament is not fully completed – system shows error and aborts 2. Points rules are missing system shows error and aborts

	3. System failure during saves
postcondition	1. No partial or duplicate point allocations are saved 2. Existing point totals remain unchanged
Actors	System (triggered by organizer)
Author	Ívar Orri Tómasson

User case 18 --- “Player statistics”

Use case name	Player statistics
Number	UC – 18
Priority	B
Source	B5
Precondition	Players exist.
Basic flow	1. Tournament finishes. 2. System calculates player statistics. 3. System adds calculated statistics to existing statistics. 4. System stores statistics
Alternative flow	1. Players have not participated in any tournaments. All player statistics are set to 0.
Post condition	All player statistics are stored and accessible
Actors	System
Authors	Ernir Elí Ellertsson

Use case 19 ---” Team statistics”

Use case name	View Team Statistics
Number	UC – 19
Priority	B
Source	B5

Precondition	2. Team exists in system 3. System has access to Teams recorded match/ tournament data
Basic Flow	1. User navigates to a Teams profile 2. System retrieves Total wins, total earnings, clubs played for, tournaments team 3. System displays the stats on the player profile 4. User uses filters (by year, competition type) 5. System refreshes the stats according to filters
Alternative flows	1. Team has no recorded stats – system shows “No statistics available yet” with zero values 2. Data retrieval error – System shows an error; suggests reloading or contact support
Post condition	1. User gets a clear error message instead of a broken page 2. No stats data is changed 3. Failure: User gets clear error message instead of a broken page
Actor	User
Author	Ívar Orri Tómasson

Use case 20 --- “Club statistics”

Use case name	Club statistics
Number	UC – 20

Priority	B
Source	B7
Precondition	Club exists
Basic flow	1. Tournament is finished. 2. System calculates Club statistics. 3. System adds calculated statistics to existing club statistics
Alternative flow	1. If club has not participated in any tournaments. All statistics are 0
Post condition	Statistics are saved and stored by system
Actors	System
Authors	Ernir Elí Ellertsson

Use case 21 --- “New Team Created for Each Tournament”

Use case name	New Team Created for each game
Number	UC – 21
Priority	C
Source	C1
Precondition	1. Organizer is logged into the system 2. Organizer has permission to manage tournaments and teams 3. The tournament exists and is open for registration
Basic flow	1. Organizer opens the tournament management page 2. System shows tournament details, including current list of teams 3. Organizer selects “Add new team” (or similar) for this tournament

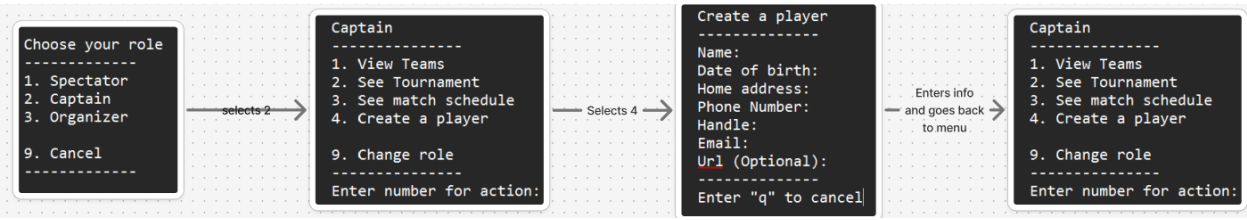
	4. System displays a “Create team for this tournament” form 5. Organizer enters required team data, example Team name, players, Club association 6. Organizer confirms and submits the form 7. System validates the input 8. System creates a new team record and links it to the current tournament 9. System updates the tournament team list and shows the new team there
Alternative flows	1. Missing or invalid data: 2. System detects invalid or incomplete data 3. System highlights errors and displays messages 4. A2- Organizer cancels 5. Organizer cancels the action; system returns to tournament page with no changes 6. A3- System error during save 7. System reports an error, and no team is created
Post conditions	1. A new team is created and linked only to the selected tournament 2. Team appears in the tournament’s team list
Actor	Organizer
Author	Ívar Orri Tómasson

Use case 22 --- “Process results files automatically”

Use case name	Process results files automatically
Number	UC – 22
Priority	C
Source	C2
Precondition	1. Results files exist. 2. Tournaments exist. 3. Games exists.
Basic flow	1. Tournament is ongoing. 2. Games are scheduled. 3. Result files come in. 4. System processes results.
Alternative flow	1. No results file? System asks for results for games.
Post condition	Results have been updated. Games update based on results.
Actors	System.
Authors	Ernir Elí Ellertsson

Happy Paths for Use Cases

UC - 01



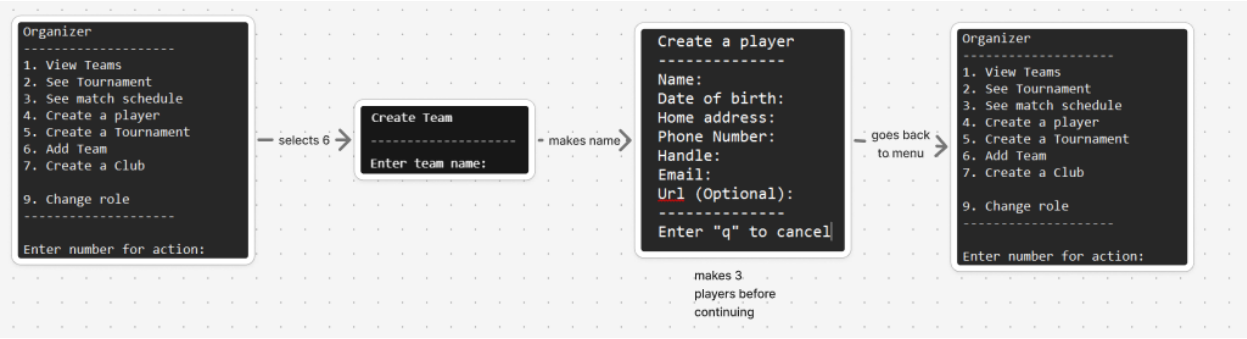
Same goes for Organizer when he tries to add a player

UC – 02, UC – 12

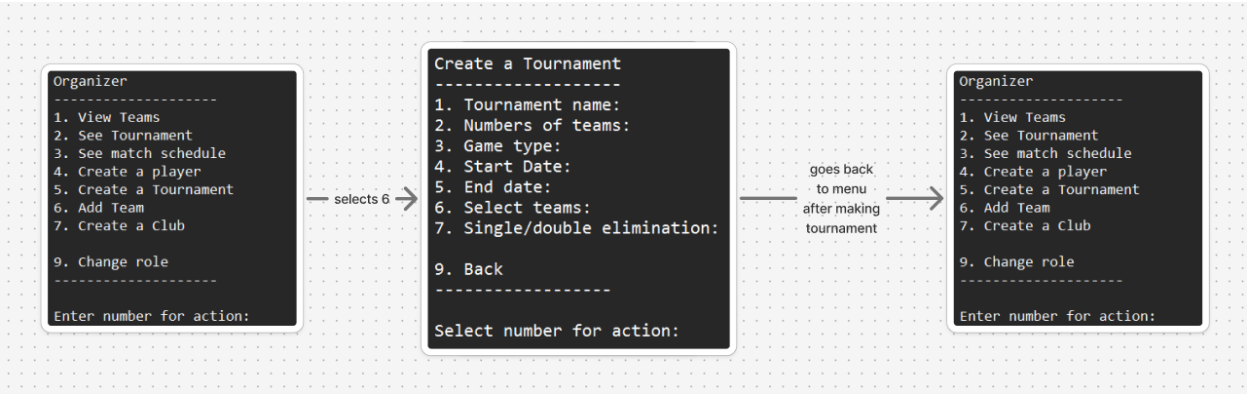
Same goes for organizer and UC - 12

UC – 04

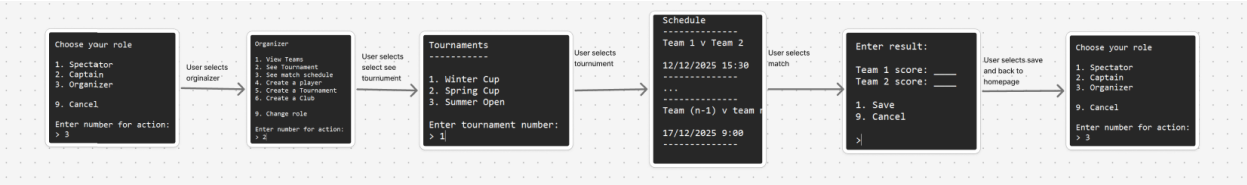
UC – 05



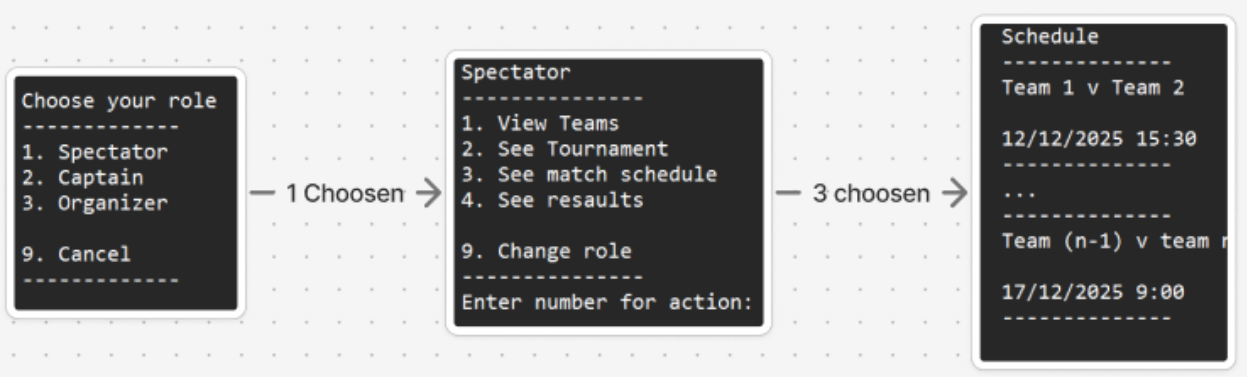
UC – 06, UC – 15



UC – 7

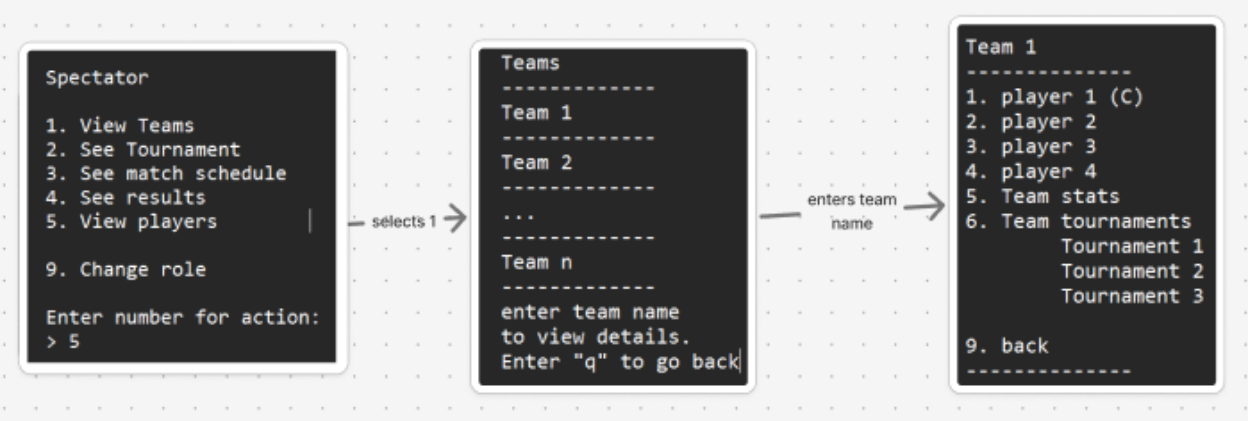


UC – 8



Same goes for captain and Organizer

UC – 9



UC – 10, UC – 11, UC – 14, UC – 18, UC – 19

For UC – 11: “Anyone can view team roster”, Stop after step 3.

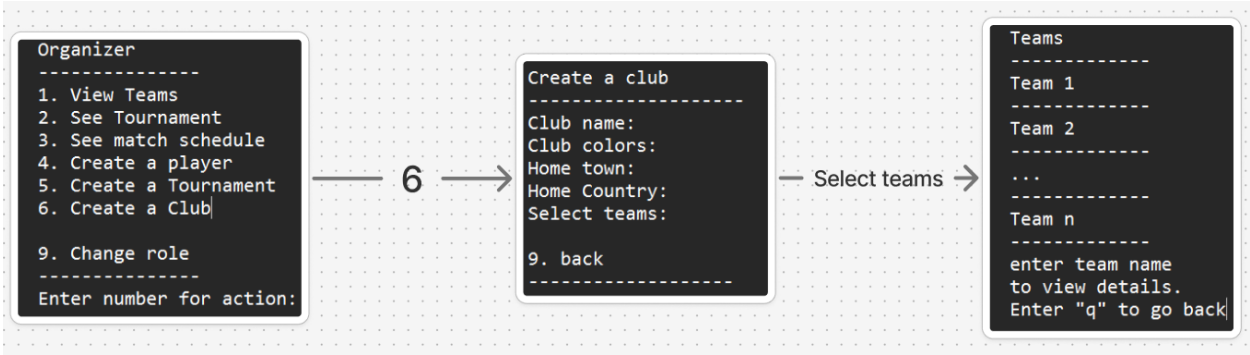
For UC – 14: “Anyone can view Team list”, stop after step 4.

For UC – 18: “View player statistics”, select 1 at last step.

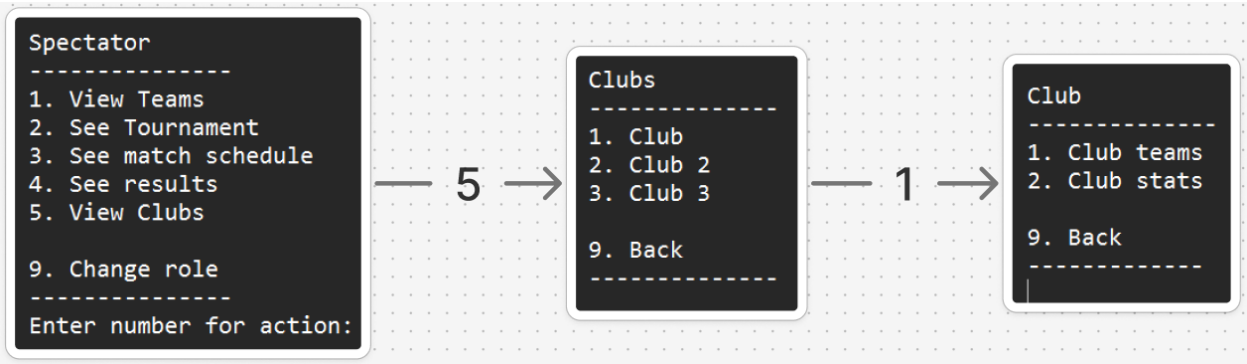
For UC – 19: “View team statistics”, select 5 after step 3

UC-13

UC – 16


Organizer
1. View Teams
2. See Tournament
3. See match schedule
4. Create a player
5. Create a Tournament
6. Create a Club
9. Change role
Enter number for action:
6
Create a club
Club name:
Club colors:
Home town:
Home Country:
Select teams:
9. back
Select teams
Teams
Team 1
Team 2
...
Team n
enter team name
to view details.
Enter "q" to go back

UC – 20


Spectator
1. View Teams
2. See Tournament
3. See match schedule
4. See results
5. View Clubs
9. Change role
Enter number for action:
5
Clubs
1. Club
2. Club 2
3. Club 3
9. Back
1
Club
1. Club teams
2. Club stats
9. Back

State Diagram

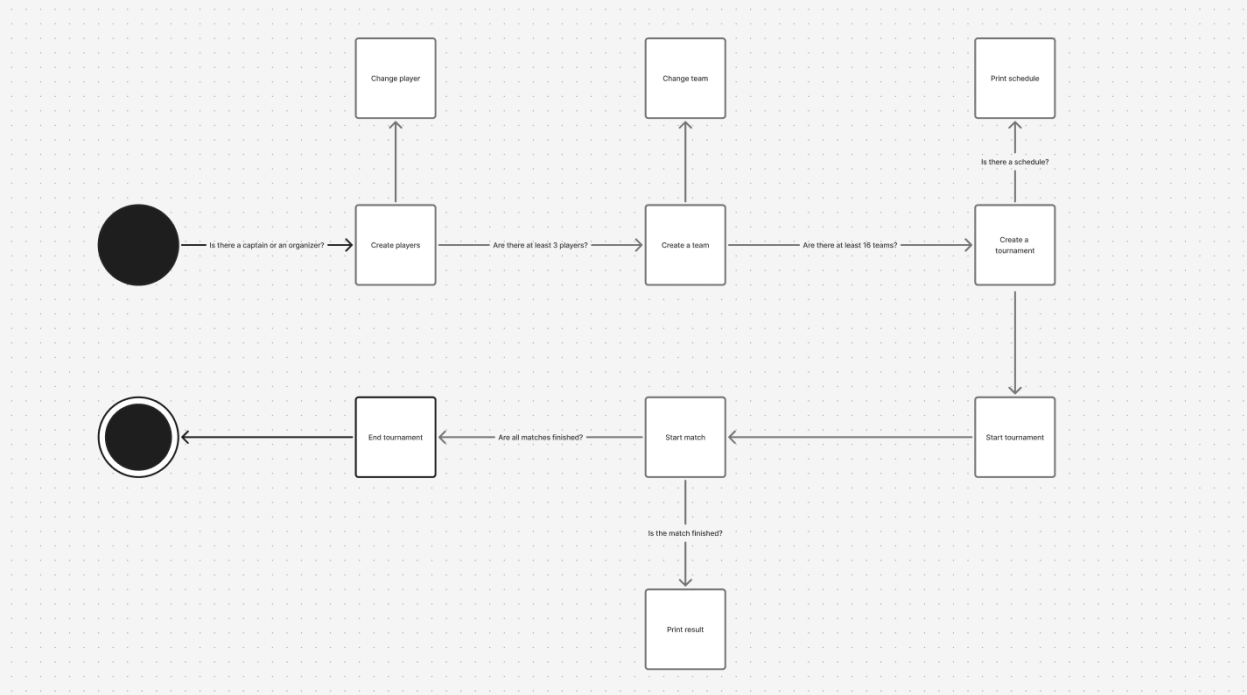
Our state diagram illustrates the flow of actions and decisions within a process. It visually represents the conditions that must be met for transitions between states to occur.

It begins in the initial state, represented by a solid black circle and progresses through various states based on specific conditions.

The transition is represented by an arrow. Each arrow may include a question. for that transition to happen, the requirements need to be fulfilled by returning true from the question on the arrow.

The squares represent states in the process. These are the states a user or system can take depending on their role and current context.

The last state is a final state and is shown as a black circle inside another circle. The final state indicates the end of the process, where no further transitions are possible.



User Case Analysis

1) Organizers / Admins

Main section	Details
NAME of group	Organizers / Admins
WHO – Background	Age: ~18–45. Gender: all genders. Education: high school or higher, often university students or staff running tournaments. Abilities/Disabilities: not known, design for mixed abilities. Computer skills: good–very good. Number: very few – usually 1 main admin
WHY – Main goals	Set up and manage tournaments, teams and players. Generate schedules/brackets. Enter and update match results. Optionally manage clubs, points and statistics.
WHAT – Equipment	Mostly laptops or desktop PCs, sometimes an external monitor. Stable internet connection. Mouse and keyboard.
WHERE – Environment	At home when planning. At the tournament venue / LAN room while the event is running. Sometimes at school or university.
WHEN – Use of system	How often: a lot in the days before a tournament and very frequently during the event. For how long each time: sessions of about 30 minutes to several hours when a tournament is running.
HOW – Skills	Comfortable with forms, tables and filters. Understands tournament formats (knock-out, double elimination). Can handle simple data validation and corrections.
HOW MUCH – Importance	Critical – without organizers the system cannot create or update tournaments.

2) Team Captains

Main section	Details
NAME of group	Team Captains
WHO – Background	Age: ~16–35. Gender: all genders. Education: mostly secondary school or university students. Abilities/Disabilities: not known, support mixed abilities. Computer skills: good; used to games, Discord, Steam, etc. Number: one per team – from a handful to dozens depending on tournament size.
WHY – Main goals	Add and maintain player information for their team. Keep contact information up to date. Check schedules and tournaments for their team.
WHAT – Equipment	Personal gaming PCs or laptops. Sometimes phones or tablets for quick checks. Home or venue Wi-Fi.
WHERE – Environment	At home. At the tournament venue between matches. Sometimes at school or university.
WHEN – Use of system	How often: a few times when setting up the team and players, then before and during tournaments. For how long each time: short sessions, usually 5–20 minutes.
HOW – Skills	Comfortable with menus, forms and tables. Can understand simple error messages (missing info, duplicates). Beginner–moderate experience with "admin" tasks.
HOW MUCH – Importance	Very important – they keep their team's data correct and are the main contact with organizers.

3) Players

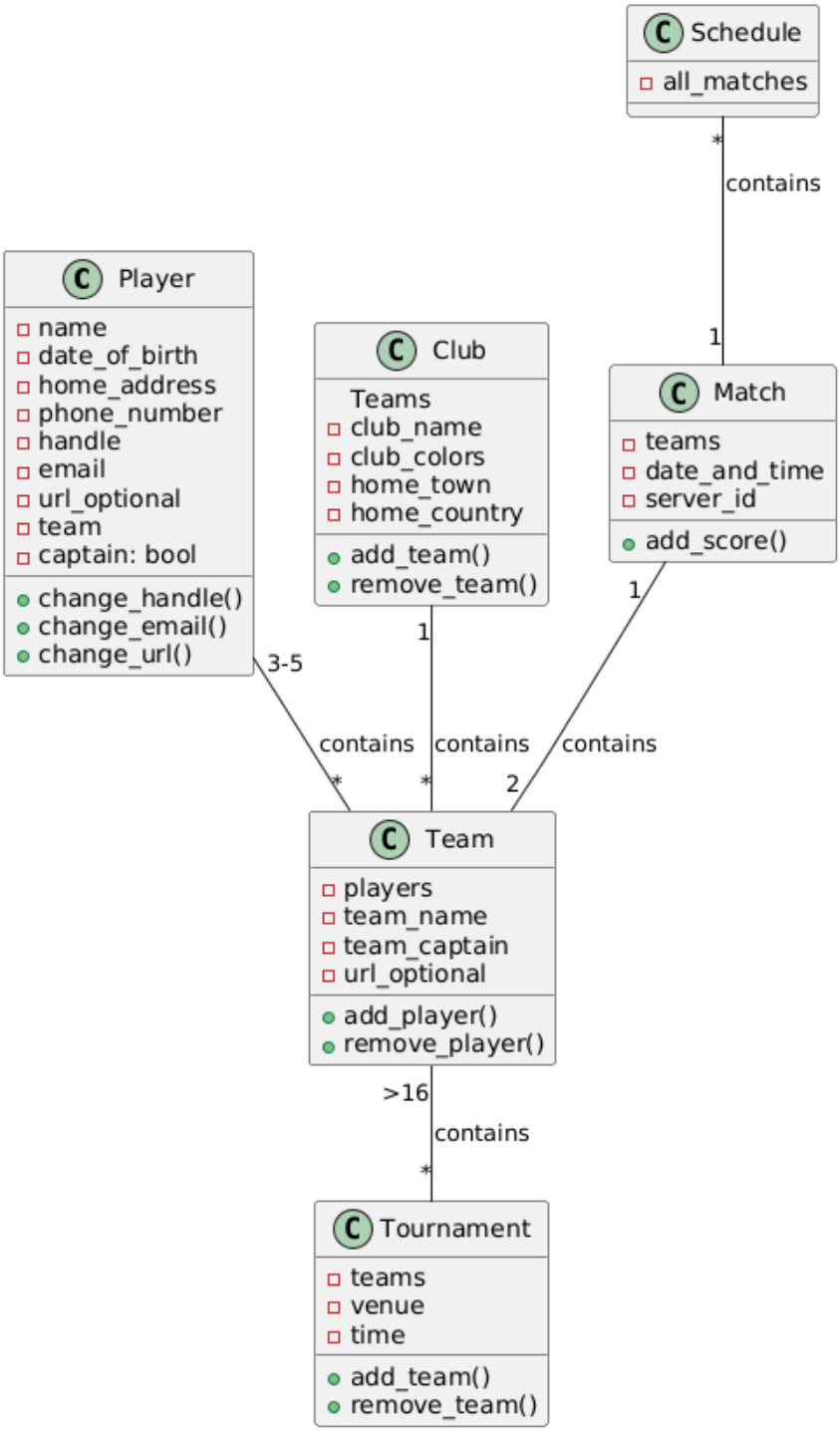
Main section	Details
NAME of group	Players (non-captain team members)
WHO – Background	Age: ~15–35. Gender: all genders. Education: mostly students, some older players. Abilities/Disabilities: not known, design for a range (e.g. colour-blind friendly). Computer skills: very good with computers and games, but may dislike complex admin UIs. Number: many – several players per team, can be tens or hundreds overall.
WHY – Main goals	See when and against whom they play. Check match results. View their own and their team's tournament history and basic statistics.
WHAT – Equipment	Gaming PCs or laptops. Very often smartphones for quick checks.
WHERE – Environment	At home before/after matches. At the venue. On the move using a phone.
WHEN – Use of system	How often: mainly around tournament days. For how long each time: very short visits (about 1–5 minutes) to check next match, results or history.
HOW – Skills	Good computer skills. Expect fast access with few clicks. Prefer simple, clear tables for times, teams and results.
HOW MUCH – Importance	Important but secondary – they don't manage data, but their experience strongly affects how good the system feels.

4) General Users / Spectators

Main section	Details
NAME of group	General Users / Spectators (friends, family, fans, journalists, teachers, etc.)
WHO – Background	Age: wide range, ~15–50+. Gender: all genders. Education: mixed – from pupils to staff/parents. Abilities/Disabilities: mixed and mostly unknown, so UI must be very clear and accessible. Computer skills: beginner to moderate. Number: potentially the largest group – anyone interested in the tournament.
WHY – Main goals	Check game schedules. See who won. Look up teams and players and which tournaments they have played in.
WHAT – Equipment	Mostly smartphones. Some laptops/desktops at school or work. Venue Wi-Fi or mobile data.
WHERE – Environment	At home. At the venue in the audience. On the go.
WHEN – Use of system	How often: mostly on match days or right after matches. For how long each time: very short sessions (under 5 minutes).
HOW – Skills	Need simple navigation and wording. Should not have to log in or understand tournament jargon. Basic click/scroll skills only.
HOW MUCH – Importance	Important for usability and visibility – they don’t run tournaments but are key for making information public and understandable.

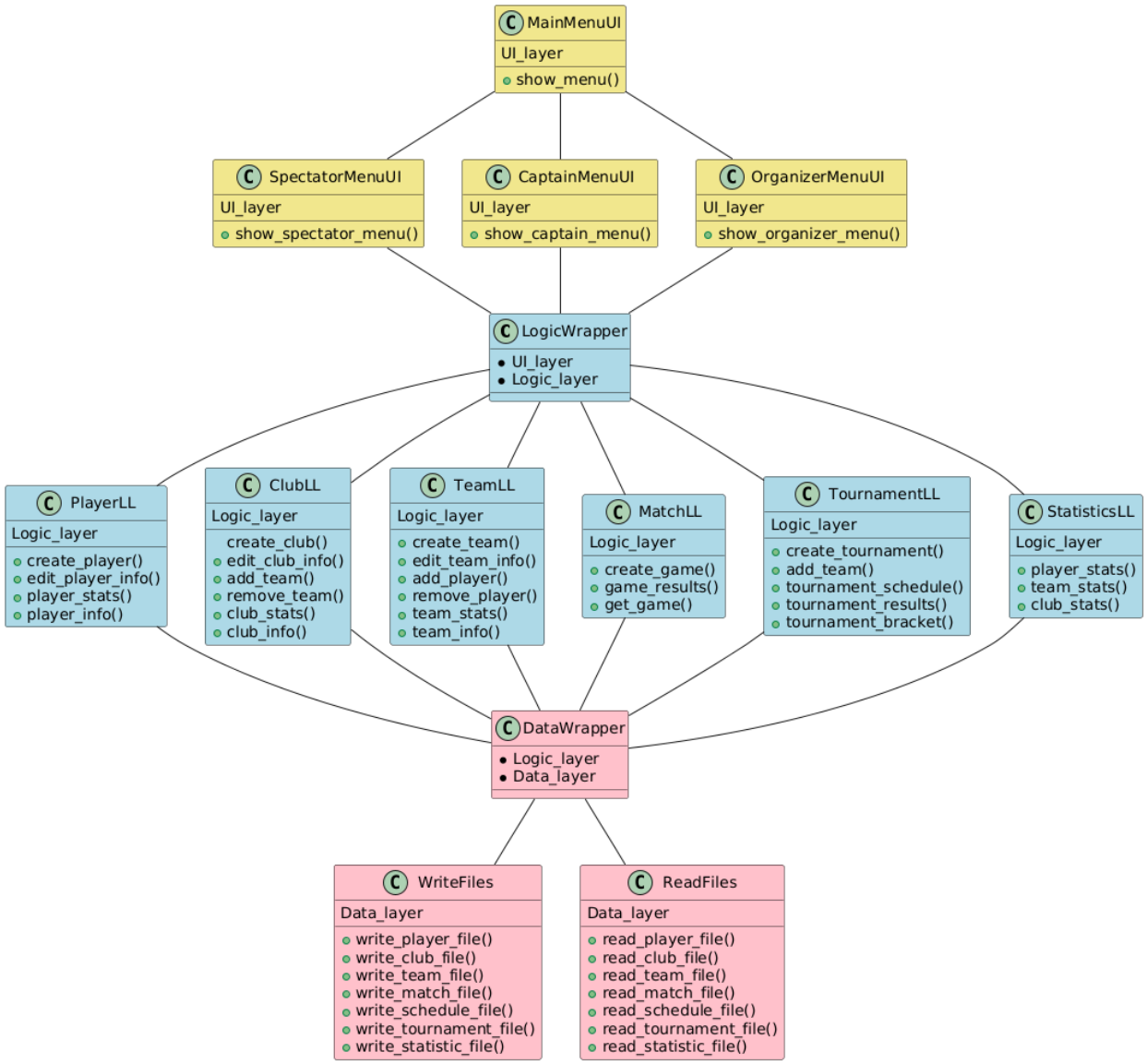
Class diagrams

Model diagram

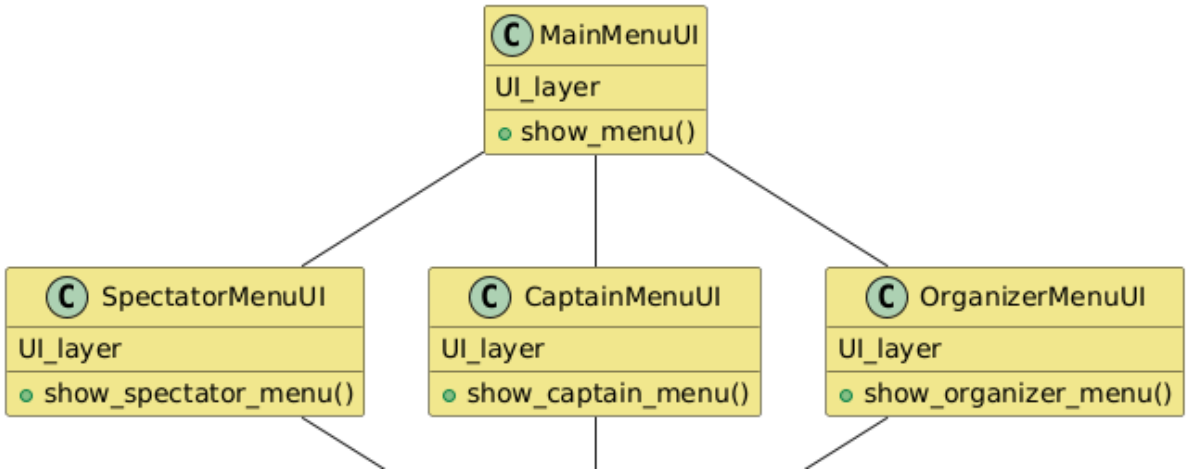


3-layer diagram

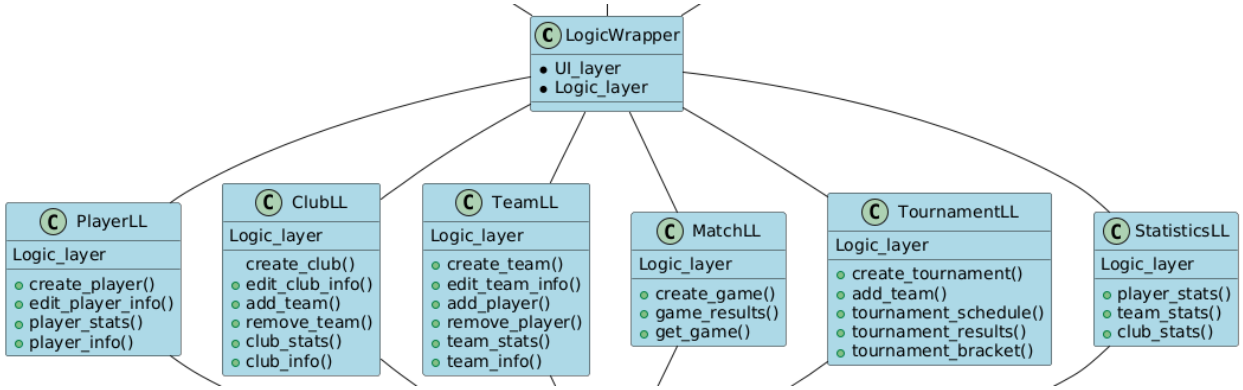
Whole



UI layer



Logic layer



Data layer