

# User Requirement Specifications

Anton Yordanov | 4095995

# Table of content

<b>Table of content</b>	<b>2</b>
<b>Agreements with client</b>	<b>3</b>
<b>Functional requirements</b>	<b>4</b>
<b>Use cases</b>	<b>5</b>
Use case #1: Create Tournament	5
Use case #2: Registering Players	5
Use case #3: Registering Results	6
Use case #4: Logging in	6
Use case #5: Updating information	7
Use case #6: Viewing player profile	7

## Agreements with client

- The solution has to be delivered before Friday 10th of June 2022 before 16.00
- OOP principles have to be followed
- No data is provided so mock data will be used
- Website should be hosted on the Luna server
- The major requirement I will develop is FR-07: Support multiple sport types
- The minor requirement I will develop is FR-10: Generate player profile

## Functional requirements

- NFR-01: Maintainable and extendable
  - NFR-02: Bug free system
  - NFR-03: Secure software
- 
- FR-01: Manage Tournaments
  - FR-02: Support registering players
  - FR-03: Support generating tournament schedule
  - FR-04: Support registering the results of the games
  - FR-05: Support showing tournament information and results
  - FR-06: Support multiple tournament systems
  - FR-07: Support multiple sport types
  - FR-08: Support matches in a tournament
  - FR-09: Support leaderboard
  - FR-10: Generate player profile
  - FR-11: Support challenge games
  - FR-12: Handle ties

Must	Should	Could	Won't
FR-01	FR-10	FR-06	FR-11
FR-02	FR-07	FR-08	
FR-03		FR-09	
FR-04		FR-12	
FR-05			

# Use cases

## Use case #1: Create Tournament

Actor : Staff

Main Success Scenario:

1. Actor opens the tournament section
2. System displays creation form
3. Actor enters the needed information
4. System confirms addition
5. End of MSS

Extension:

- 3a. Wrong information is entered
- 4a. System displays that the information is incorrect
- 5a. Return to step 2 in MSS

## Use case #2: Registering Players

Actor : Player

Main Success Scenario:

1. Actor goes to registration page
2. System displays registration form
3. Actor enters the required data
4. System creates account
5. End of MSS

Extension:

- 3a. Wrong information is entered
- 4a. System displays that the information is incorrect
- 5a. Return to step 2 in MSS

## Use case #3: Registering Results

Actor : Staff

Main Success Scenario:

1. Actor opens the game section
2. System displays game section
3. Actor selects game to enter the results
4. System displays the game form
5. Actor enters the results of the game
6. System registers the results
7. End of MSS

Extension:

- 5a. Wrong information is entered
- 6a. System displays that the information is incorrect
- 7a. Return to step 4 in MSS

## Use case #4: Logging in

Actor : Player

Main Success Scenario:

1. Actor goes to login page
2. System displays login form
3. Actor enters the required data
4. System logs player in
5. End of MSS

Extension:

- 3a. Wrong information is entered
- 4a. System displays that the information is incorrect
- 5a. Return to step 2 in MSS

## Use case #5: Updating information

Actor : Player

Main Success Scenario:

1. Actor goes to account page
2. System displays account information
3. Actor Enters new information
4. System updates information
5. End of MSS

Extension:

- 3a. Wrong information is entered
- 4a. System displays that the information is incorrect
- 5a. Return to step 2 in MSS

## Use case #6: Viewing player profile

Actor : Player

Main Success Scenario:

1. Actor selects another player account
2. System displays account information
3. End of MSS