



User Requirement Specification Document

Project: DuelSys Tournament
Management System

Rasim, Mertan M.I.

FONTYS UNIVERSITY OF APPLIED SCIENCES

Contents

Document versioning	2
Constraints.....	2
Core requirements.....	2
Functional requirements	2
Non-functional requirements.....	3
Major functional requirements	3
Minor functional requirements	3
Use Cases.....	3
UC-01: Staff member wants to login into the desktop application	3
UC-02: Staff member wants to create a tournament	3
UC-03: Player wants to create an account.....	4
UC-04: Player wants to log in to their account	4
UC-05: Player wants to join a tournament	5
UC-06: Staff wants to view the schedule of a tournament	5
UC-07: Staff wants to generate a schedule for a tournament	5
UC-08: Staff wants to register the results of a tournament	6
UC-09: Any person wants to see the details of a tournament	6
UC-10: Staff wants to view the scoreboard of a tournament	6
UC-11: Staff wants to view the match results of a tournament.....	7

Document versioning

Version	Description	Date
1.0	<ul style="list-style-type: none">Added versioningAdded use cases	15-05-2022
2.0	<ul style="list-style-type: none">Changed major requirement to FR-06Added UC-06 (View schedule)Refactored UC-07 (Generate schedule)Added extensions to UC-02Fixed typos	17-05-2022
2.1	<ul style="list-style-type: none">Added UC-10 (Scoreboard)	19-05-2022
2.2	<ul style="list-style-type: none">Added UC-11 (View results)	22-05-2022
2.3	<ul style="list-style-type: none">Added Major requirement FR-08	07-06-2022

Constraints

	Description
Mocking data	The case is simulated and there is no real data. Do make sure you make use of the any data you can find in this document. Everything else can be your own "fake" data.
UX	Make sure the software solution is as expected for modern applications; e.g. proper UX, user feedback, common practices (for example, hide sensitive data such as password), etc.
Technologies	C# with Windows Forms and ASP.Net Core Razor Pages; for a web application you may use a layout framework, such as Bootstrap, but not an ORM. The database must be a MySQL database.
Meetings	Weekly meetings with tutor are mandatory.
Deadline	Friday 10 th of June 2022 before 16.00.
Source control	FHICT GitLab (https://git.fhict.nl). Make sure you invite your OOD, WAD & WKS teachers with the appropriate access (i.e. at least Reporter access).
Website	The website should run on Luna server.
Database	The database should run on Hera server.
Submission	Final submission with all deliverables must be submitted on Canvas before the deadline.

Core requirements

Functional requirements

- FR-01: Tournament overview
 - Staff must be able to manage (CRUD operations) the tournaments.
- FR-02: Support registering players
 - When a player is interested in participating in a tournament, they can visit the sport association website, retrieve the list of available tournaments, and register themselves for it.
- FR-03: Support generating tournament schedule
 - Staff must be able to generate the tournament's schedule.
- FR-04: Support registering the results of the games
 - When a game between two players is finished, the results must be registered in the system by staff.

- FR-05: Support showing tournament information and results
 - Any interested party (e.g., a sport enthusiast, a player) must be able to retrieve information about any given tournament.

Non-functional requirements

- NFR-01: Maintainable and extendable
 - Proper OO principles must be applied to ensure good maintainability and extensibility of the code base
- NFR-02: Bug free system
 - Appropriate testing techniques must be used when implementing the system to ensure proper functioning.
- NFR-03: Secure software
 - Only authorized people may make use of the system and can only access data they are authorized for. Passwords and user input must also be handled appropriately.

Major functional requirements

- FR-06: Support multiple tournament systems
 - Double round-robin tournament
- FR-08: Support matches in a tournament

Minor functional requirements

- FR-11: Handle ties

Use Cases

UC-01: Staff member wants to login into the desktop application

- **Actor:** Staff
- **Trigger:** Start the desktop application
- **Main Success Scenario:**
 1. System requests for an email and a password.
 2. The actor enters their credentials.
 3. The actor confirms the action.
 4. System approves the credentials and opens the main form.
- **Extensions:**
 - 3a: The provided information is insufficient
 - .1: System displays an error message
 - .2: Return to MSS step 1
 - 3b: The provided information is incorrect
 - .2: System displays an error message
 - .3: Return to MSS step 1

UC-02: Staff member wants to create a tournament

- **Actor:** Staff
- **Pre-condition:** Staff is logged in (UC-01)
- **Main Success Scenario:**
 1. Actor navigates to the "Create tournament" section.

2. System requests for details – sport type, tournament type, location, description, starting date, ending date, maximum capacity, and minimum capacity.
 3. Actor fills out the information and confirms.
 4. System confirms the action.
- **Extensions:**
 - 3a: The provided information is insufficient
 - .1: System displays an error message
 - .2: Return to MSS step 1
 - 3b: Starting date is later than ending date
 - .2: System displays an error message
 - .3: Return to MSS step 1
 - 3c: Minimum capacity is greater than maximum capacity
 - .2: System displays an error message
 - .3: Return to MSS step 1
 - 3d: Minimum capacity is less than 3 players
 - .2: System displays an error message
 - .3: Return to MSS step 1

UC-03: Player wants to create an account

- **Actor:** Player
- **Main Success Scenario:**
 1. Actor visits the website and navigates to the registration page.
 2. System requests for details – name, email, and a password.
 3. Actor fills out the information and confirms.
 4. System confirms the action.
- **Extensions:**
 - 3a: The provided information is insufficient
 - .1: System displays an error message
 - .2: Return to MSS step 1
 - 3b: Email address is in incorrect format
 - .2: System displays an error message
 - .3: Return to MSS step 1
 - 3c: Email address is already in use
 - .2: System displays an error message
 - .3: Return to MSS step 1

UC-04: Player wants to log in to their account

- **Actor:** Player
- **Pre-condition:** Player already has an account (UC-03)
- **Main Success Scenario:**
 1. Actor visits the website and navigates to the login page.
 2. System requests for credentials – an email and a password.
 3. Actor fills out the information and confirms.
 4. System confirms the action.

- **Extensions:**
 - 3a: The provided information is insufficient
 - .1: System displays an error message
 - .2: Return to MSS step 1
 - 3b: The provided information is incorrect
 - .2: System displays an error message
 - .3: Return to MSS step 1

UC-05: Player wants to join a tournament

- **Actor:** Player
- **Pre-condition:** Player is logged in (UC-04)
- **Main Success Scenario:**
 1. Actor visits the website and navigates to the home page.
 2. Actor chooses a tournament of their choice.
 3. System registers the actor in the tournament.
 4. System notifies the actor for the action.
- **Extensions:**
 - 2a: The tournament is full
 - .1: System displays an error message
 - .2: End of use case
 - 2b: Player is already registered in another tournament with an overlapping schedule
 - .2: System displays an error message
 - .3: End of use case

UC-06: Staff wants to view the schedule of a tournament

- **Actor:** Staff
- **Pre-condition:** Staff is logged in (UC-01)
- **Main Success Scenario:**
 1. Actor navigates to the "Tournament overview" section.
 2. System provides a list of tournaments.
 3. User selects a tournament and selects the option to view its schedule.
 4. System displays a new section with a table of the scheduled games.
- **Extensions:**
 - 3a: The selected tournament does not have a schedule
 - .1: System suggests generating a schedule
 - .2: Actor confirms the action
 - .3: Trigger UC-07 MSS Step 4
 - .4: Resume UC-08 MSS Step 4

UC-07: Staff wants to generate a schedule for a tournament

- **Actor:** Staff
- **Pre-condition:** Staff is logged in (UC-01)
- **Main Success Scenario:**
 1. Actor navigates to the "Tournament overview" section.
 2. System provides a list of tournaments.
 3. User selects a tournament and selects the option to generate a schedule.

4. System generates a schedule and confirms the action.
- **Extensions:**
 - 3a: The selected tournament already has a generated schedule
 - .1: System provides the option to overwrite existing schedule
 - .2: Actor accepts the action
 - .3: Return to MSS Step 4
 - 3b: The number of registered users is less than the minimum required
 - .1: System displays an error message
 - .2: End of use case
 - 3c: The tournament is expired
 - .1: System displays an error message
 - .2: End of use case

UC-08: Staff wants to register the results of a tournament

- **Actor:** Staff
- **Pre-condition:** Staff is logged in (UC-01)
- **Main Success Scenario:**
 1. Actor navigates to the “Tournament overview” section.
 2. System provides a list of tournaments.
 3. Actor selects the option to register results.
 4. System displays a table of individual games and scores.
 5. Actor modifies game scores and confirms changes.
 6. System confirms the action.
- **Extensions:**
 - 3a: The selected tournament does not have a schedule
 - .1: System suggests generating a schedule
 - .2: Actor confirms the action
 - .3: Trigger UC-07 MSS Step 4
 - .4: Resume UC-08 MSS Step 4
 - 5a: A score result does not conform to the tournament’s scoring standards
 - .1: System highlights entries and displays an error message
 - .2: Return to MSS Step 2

UC-09: Any person wants to see the details of a tournament

- **Actor:** Visitor, Staff, Player
- **Main Success Scenario:**
 1. Actor visits the website.
 2. Actor selects the tournament of interest.
 3. System redirects to a page with further details of the tournament.

UC-10: Staff wants to view the scoreboard of a tournament

- **Actor:** Staff
- **Pre-condition:** Staff is logged in (UC-01)

- **Main Success Scenario:**
 1. Actor navigates to the “Tournament overview” section.
 2. System provides a list of tournaments.
 3. Actor selects the option to view the scoreboard.
 4. System displays a table of three details – place, player name, and number of matches won.
- **Extensions:**
 - 3a: The selected tournament does not have a schedule
 - .1: System suggests generating a schedule
 - .2: Actor confirms the action
 - .3: Trigger UC-07 MSS Step 4
 - .4: Resume UC-08 MSS Step 4

UC-11: Staff wants to view the match results of a tournament

- **Actor:** Staff
- **Pre-condition:** Staff is logged in (UC-01)
- **Main Success Scenario:**
 1. Actor navigates to the “Tournament overview” section.
 2. System provides a list of tournaments.
 3. User selects a tournament and selects the option to view its results.
 4. System displays a new section with a table of the scheduled games and results.
- **Extensions:**
 - 3a: The selected tournament does not have a schedule
 - .1: System suggests generating a schedule
 - .2: Actor confirms the action
 - .3: Trigger UC-07 MSS Step 4
 - .4: Resume UC-08 MSS Step 4