Synthesis Assignment

User Requirement Specification Document

**by Lachezar Mitov**

**Contents**

[**Document versioning** 2](#_Toc105926614)

[**Introduction** 2](#_Toc105926615)

[**Website Address** 2](#_Toc105926616)

[**Functional requirements** 3](#_Toc105926617)

[Desktop application 3](#_Toc105926618)

[Website application 3](#_Toc105926619)

[**Use Cases** 4](#_Toc105926620)

[UC-1: Staff wants to sign-in 4](#_Toc105926621)

[UC-2: User wants to sign-out 4](#_Toc105926622)

[UC-3: Staff want to create a tournament 5](#_Toc105926623)

[UC-4: Staff wants to delete a tournament 5](#_Toc105926624)

[UC-5: Staff wants to update a tournament 6](#_Toc105926625)

[UC-6: Staff wants to make the schedule for a tournament 6](#_Toc105926626)

[UC-7: Staff wants to register a result of a match in the system 7](#_Toc105926627)

[UC-8: User want to register for a tournament 7](#_Toc105926628)

[UC-9: User wants to see information about tournaments 8](#_Toc105926629)

[UC-10 User wants to see tournament result 8](#_Toc105926630)

[UC-11: User wants to create an account 9](#_Toc105926631)

[UC-12: User wants to see their profile page 9](#_Toc105926632)

[**Database Design** 10](#_Toc105926633)

# **Document versioning**

|  |  |  |
| --- | --- | --- |
| Version | Description | Date |
| 1.0 | * Added main structure of the document * Added functional requirements * Added use cases | 10-05-2022 |
| 1.1 | * Added document versioning * Updated use cases to have pre-condition * Added new use cases ( UC-10, UC-11, UC-12) * Added reference to main document where needed | 23-05-2022 |
| 1.2 | * Updated the trigger in all use cases | 03-06-2022 |
| 1.3 | * Added website address * Added database design | 12-06-2022 |

# **Introduction**

This document will display all the functional requirements that are expected to be implemented in the application. Moreover, Use Cases with Main Success Scenario about each functional requirement are listed below.

# **Website Address**

[https://i477927.luna.fhict.nl](https://i477927.luna.fhict.nl/)

# **Functional requirements**

Because of already set names to the use cases in the program reader the naming of the use cases is not consistent.

## Desktop application

Features include:

* FR-01A: Staff should be able to create and delete tournaments
* FR-01B: Staff should be able to read and update tournaments information
* FR-03: Staff should be able to generate schedule for a tournament
* FR-04: Staff should be able to register a match result in the system
* FR-06: System should support multiple tournament types of badminton tournaments
* FR-07: Authentication and authorization

## Website application

Features include:

* FR-02: System should support registering of a player for a tournament
* FR-05: System should be able to show tournament information and results
* FR-10: System should support showing player profile information

# **Use Cases**

## UC-1: Staff wants to sign-in

* + **Actor:** Staff
  + **Trigger:** Open application
  + **Main Success Scenario:** 
    1. System requires for username and password
    2. Actor enters their username and password
    3. Actor confirms the action
    4. System approves the credentials, and the user is transferred to the staff application
  + **Extensions:**

3a: The provided information is insufficient

.1: System highlights the fields with insufficient information

.2: System displays an error message

.3: Return to MSS step 1

3b: The provided information is incorrect

.1: System displays error message

.2: Return to MSS step 1

## UC-2: User wants to sign-out

* + **Actor:** Staff/Player
  + **Trigger:** Pressing the ‘Sign Out’ button
  + **Pre-condition:** Actor should be logged in
  + **Main Success Scenario:** 
    1. Actor presses the ‘Sign out’ button.
    2. System terminates actor’s session.
    3. Return to UC-1
  + **Extensions:**

## UC-3: Staff want to create a tournament

* + **Actor:** Staff
  + **Trigger:** Pressing the ‘Add Tournament’ button
  + **Pre-condition:** Actor should be logged in
  + **Main Success Scenario:** 
    1. Actor navigates to the form for creating a tournament
    2. Actor enters the necessary information(for the necessary information please refer to 2.1 Phrase 1 in <https://fhict.instructure.com/courses/12082/pages/week-11-introduction-and-q-and-a?module_item_id=750440> )
    3. Actor confirms the action
    4. System creates a tournament successfully
  + **Extensions:**

3a: The provided information is insufficient

.1: System displays error message

.2: Return to MSS step 2

3b: The provided information is invalid

.1: System displays error message

.2: Return to MSS step 2

## UC-4: Staff wants to delete a tournament

* + **Actor:** Staff
  + **Trigger:** Pressing the ‘Update Tournament’
  + **Pre-condition:** Tournament should exist
  + **Main Success Scenario:**
    1. Actor navigates to Tournaments tab
    2. Actor chooses the tournament to be deleted
    3. Actor opens the form for deleting a tournament
    4. Actor confirms the action
    5. System performs the action and notifies the user
  + **Extensions:**

2a: Actor has not specified the tournament

.1: Systems notifies the user to specify the correct tournament

.2: Return to MSS step 2

## UC-5: Staff wants to update a tournament

* + **Actor:** Staff
  + **Trigger:** Pressing the ‘Update Tournament’
  + **Pre-condition:** Tournament should be available
  + **Main Success Scenario:** 
    1. Actor navigates to Tournaments tab
    2. Actor chooses the tournament to be updated
    3. Actor navigates to the form for updating tournaments
    4. Actor makes the changes
    5. Actor confirms the action
    6. System performs the action and notifies the user
  + **Extensions:**

2a: Actor has not specified the tournament

.1: Systems notifies the user to specify the correct tournament

.2: Return to MSS step 2

5b: Actor has filled invalid data

.1: Systems displays error message

.2: Return to MSS step 2

## UC-6: Staff wants to make the schedule for a tournament

* + **Actor:** Staff
  + **Trigger:** Pressing the ‘Create/View Schedule’
  + **Pre-condition:** Tournament should be available
  + **Main Success Scenario:**
    1. Actor navigates to Schedule tab
    2. Actor chooses a new tournament
    3. Actor confirms the action
    4. System adds and displays the schedule to the tournament
  + **Extensions:**

3a: Actor chooses a tournament which is more than 7 days in the future

.1: System displays error message

.2: Return to MSS step 1

3a: Actor chooses a tournament which was not played

.1: System displays error message

.2: Return to MSS step 1

## UC-7: Staff wants to register a result of a match in the system

* + **Actor:** Staff
  + **Trigger:** Pressing the ‘Add Result’ button
  + **Pre-condition:** Schedule consisting of matches should be available
  + **Main Success Scenario:**
    1. Actor navigates to Schedule tab
    2. Actor chooses a match from a tournament
    3. Actor opens the form for adding result
    4. Actor inputs the result of the match complying with the sport’s rules
    5. Actor confirms the action
    6. Result is registered in the system
  + **Extensions:**

5b: The input information does not comply with the rules

.1: System notifies the actor to fill in correct result

.2: Return MSS step 2

## UC-8: User want to register for a tournament

* + **Actor:** Player
  + **Trigger:** Pressing the ‘Join’ button
  + **Pre-condition:** Tournament should be available
  + **Main Success Scenario:** 
    1. Actor navigates to Tournaments page
    2. Actor chooses tournament to register in
    3. Actor confirms the action
    4. Actor is added to the tournament’s list
  + **Extensions:**

2a: Actor is not logged in

.1: System redirects actor to Login page

.2: Return MSS step 1

2b: The chosen tournament’s list is already full

.1: Actor cannot join the list

.2: Return MSS step 1

## UC-9: User wants to see information about tournaments

* + **Actor:** User
  + **Trigger:** Pressing the ‘Tournaments’ button
  + **Pre-condition:** Tournament should be available
  + **Main Success Scenario:** 
    1. Actor navigates to the Tournaments page

**Extensions:**

1a: Tournaments page is empty

.1: Page is empty

.2: Return MSS step 1

## UC-10 User wants to see tournament result

* + **Actor:** User
  + **Trigger:** Pressing the ‘See Results’ button
  + **Pre-condition:** Tournament should be available
  + **Main Success Scenario:** 
    1. Actor navigates to the Tournaments page
    2. Actor chooses a tournament and presses the button
  + **Extensions:**

1a: Tournaments page is empty

.1: Page is empty

.2: Return MSS step 1

2a: Tournament was not played

.1: Button is locked

.2: Return MSS step 1

## UC-11: User wants to create an account

* + **Actor:** User
  + **Trigger:** Pressing the ‘Register’ button
  + **Pre-condition:** Web application should be open
  + **Main Success Scenario:** 
    1. Actor navigates to the Register page
    2. Actor fills the form
    3. Actor confirms the action
    4. System creates an account
  + **Extensions:**

3a: The provided information is incorrect

.1: Systems displays error message

.2: Return MSS step 2

3b: The provided credentials are already used

.1: Systems displays error message

.2: Return MSS step 2

## UC-12: User wants to see their profile page

* + **Actor:** User
  + **Trigger:** Pressing the ‘Log In’ button
  + **Pre-condition:** Actor should be logged in
  + **Main Success Scenario:** 
    1. Actor navigates to the login page
    2. Actor logs in with their credentials
    3. Actor is navigated to the profile page
  + **Extensions:**

2a: The provided credentials are incorrect

.1: Systems displays error message

.2: Return MSS step 2

# **Database Design**

The database has table for users where both Players and Employees are stored. The Matches table has 2 foreign keys connected to Users table for the id’s of the 2 players and a foreign key connected to the Tournaments table for the id of the tournament. TournamentPlayers is a junction table between Users and Tournaments table, which stores the id of the players and id of the tournament he/she plays in. This table holds the data for every tournament and its participants.

Diagram

Description automatically generated