A picture containing icon

Description automatically generated

REVISION HISTORY

Done by: Stefan Popescu (4186354)

Tutor: Tim Kurvers

|  |  |
| --- | --- |
| Version | Changes |
| 0.1 | Created the document |
|  |  |

Contents

[1. Case description 3](#_Toc103521165)

[2. User requirements 3](#_Toc103521166)

[2.1 Functional requirements 3](#_Toc103521167)

[2.1.1 Core requirements 4](#_Toc103521168)

[2.1.2 Major Requirements 5](#_Toc103521169)

[2.1.3 Minor requirements 6](#_Toc103521170)

[2.1.4 My Requirements 7](#_Toc103521171)

[2.2 Nonfunctional requirements 8](#_Toc103521172)

[3. Use cases 9](#_Toc103521173)

[1. C# Application 9](#_Toc103521174)

[UC-01: Adding a new employee to the system 9](#_Toc103521175)

[UC-02: Removing an employee from the system 9](#_Toc103521176)

[UC-03: Updating an employee’s information 10](#_Toc103521177)

[UC-04: Viewing all the employees of the company 11](#_Toc103521178)

[UC-05: Viewing all the available tournaments 11](#_Toc103521179)

[UC-06: Adding a new tournament to the system 11](#_Toc103521180)

[UC-07: Removing a tournament from the system 12](#_Toc103521181)

[UC-08: Updating a tournament’s information 12](#_Toc103521182)

# Case description

The company *DuelSys inc.* wants a software solution to allow their customers (sport associations) to manage their sport tournaments. For now, the software must support a *round-robin1* tournament system for *badminton*2, but *DuelSys inc.* also wants the software to have the potential to support *other types of* tournament systems and sports.

A tournament has multiple *players* competing in *badminton* games to determine who is the best (e.g., gold, silver, and bronze medal). To determine this, the purpose of the software is to register all the results of each game.

This software solution will be used by sport association *staff* (*staff*) to organize tournaments and by *players* to find information about the tournament(s) they want to participate in.

# User requirements

## Functional requirements

Below you can find the requirements for this assignment, and they are divided into three (3) categories. Your submission must contain all the *Core requirements*, at least one (1) of the *Major requirements* and at least one (1) of the *Minor requirements*.

During the first tutor meeting in week 13, you can propose and *get approval* from your tutor which Major and Minor functional requirements you would like to implement. Make sure that your selection gives you an opportunity to cover all LOs.

In addition, you can still propose and discuss with your tutor which *other* functional requirements you want to include.

### Core requirements

|  |  |
| --- | --- |
| FR | Priority (MSCW) |
| *FR-01: Manage Tournaments Staff* must be able to manage (*CRUD* operations) the tournaments. | M |
| *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. | M |
| *FR-03*: *Support generating tournament schedule Staff* must be able to generate the tournament’s schedule. | M |
| *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*. | M |
| *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. | M |

### Major Requirements

|  |  |
| --- | --- |
| FR | Priority (MSCW) |
| * 1. *FR-06: Support multiple tournament systems*   Extend the software solution to also support different tournament systems. It should be possible, for a *staff* member, to specify what tournament system should be used when creating a new tournament. For now, at least one of the following tournament systems (in addition to round-robin) is required:   * 1. o Single-elimination3   2. o Double-elimination4   3. o Double round-robin5 | M |
| *FR-07: Support multiple sport types*  Extend the software solution to also support different sport types (e.g., basketball, tennis, quidditch, league of legends, chess, etc.). It should be possible, for a *staff* member, to specify which sport type when creating new tournament. Make sure that when registering the result of a game the official scoring rules are followed. | C |
| *FR-08: Support matches in a tournament*  Extend the software solution to support multi-game matches in a tournament. For now, every match will consist of three (3) games that must be played by the same players and the player with most one games has won the match. | C |

### Minor requirements

|  |  |
| --- | --- |
| FR | Priority (MSCW) |
| *FR-09: Support leader board* Extend the software solution to also support a leader board. When there is an ongoing tournament, any interested party (e.g., a sport enthusiast, a player) can retrieve the list of players participating in the tournament, ordered based on their current position/rank in the tournament. | C |
| *FR-10: Generate player profile* Extend the software solution to also show player profile information. The information should include player’s general information, information related to participation in different tournaments with the ranking and can include the individual matches (games) played against different opponents. | M |
| *FR-11: Support challenge games* Extend the software solution to also support challenge games. It should be possible for a player to challenge another player for a game. When the challenge is created, the opponent  can either accept or reject the challenge. If it is accepted, the result of the game can be registered by one of the players. | C |
| *FR-12: Handle ties*  At the end of a tournament, when all matches are played that the software only have three winners. The software handles any ties, i.e., two or more players have won the same number of games/matches, so that is always only one player per for the first, the second and the third places.  As an example, given in Figure 3 of Appendix A, Kento Momota and Viktor Axelsen won two (2) games/matches each, while the other opponents just won one (1). The system decides that Kento Momota should score the first place, because he won both games played against Viktor Axelsen.  For handling ties, you can re-use and extend the strategy given in the example above or define your own strategy. | C |

### 2.1.4 My Requirements

|  |  |
| --- | --- |
| FR | Priority (MSCW) |
| *FR-12: Administrator Staff* must be able to manage (*CRUD* operations) the *Staff.* | C |

## Nonfunctional 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.

# 3. Use cases

## 1. C# Application

### UC-01: Adding a new employee to the system

* FR: 12
* Actor: Administrator, System
* Pre-condition:
* Trigger:
* Main success scenario:
  1. The actor navigates to the employee’s management tab.
  2. The actor selects to the “Add employee”.
  3. The system displays a new form “Add employee”.
  4. The actor fills in the fields with the required information.
  5. The actor navigates to the “Add” button.
  6. The system sends a success message.
  7. The system announces the employee has been successfully added to the system.
* Extensions:

3a. There is an option where the actor can stop the action by pressing the “X” sign at the top of the new popped up form.

1. Go back to MSS step 1.

5a. There is one or more fields with missing of syntactically incorrect information.

1. The system shows a notification informing the actor which information is not appropriate to the required format.
2. Return to MSS step 4.

### UC-02: Removing an employee from the system

* FR: 12
* Actor: Administrator, System
* Pre-condition:
* Trigger:
* Main success scenario:
  1. The actor navigates to the employee’s management tab.
  2. The actor navigates to the employee he wants to remove.
  3. The actor navigates to the “Delete employee” button.
  4. The system removes the selected employee from the system.
  5. The system displays a success message, implying the action was successful.
* Extensions:

3a. The system sends a message stating that no employee has been selected.

1. Return to MSS step 2.

### UC-03: Updating an employee’s information

* FR: 12
* Actor: Administrator, System
* Pre-condition:
* Trigger:
* Main success scenario:

1. The actor navigates to the employee’s management tab.
2. The actor selects the employee he wants to update the information on.
3. The actor navigates to the “Edit employee” button.
4. A new form “Edit employee” pops up.
5. The actor makes the wanted modifications.
6. The actor navigates to the “Edit” button.
7. The system sends a success message.
8. The employee has been successfully modified.

* Extensions:

4a. There is an option where the actor can stop the action by pressing the “X” sign at the top of the new popped up form.

1. Return to MSS step 1.

6a. There is one or more fields with missing of syntactically incorrect information.

1. The system shows a notification informing the actor which information is not appropriate to the required format.
2. Return to MSS step 5.

### UC-04: Viewing all the employees of the company

* FR: 12
* Actor: Administrator, Employee, System
* Pre-condition:
* Trigger:
* Main success scenario:

1. The actor navigates to the employee’s management tab.
2. The actor sees all the information about the employees of the company.

### UC-05: Viewing all the available tournaments

* FR: 01
* Actor: Administrator, Employee, System
* Pre-condition:
* Trigger:
* Main success scenario:
  1. The actor navigates to the tournaments’ management tab.
  2. The actor sees all the information about the available tournaments.

### UC-06: Adding a new tournament to the system

* FR: 01
* Actor: Administrator, Employee, System
* Trigger:
* Main success scenario:
  1. The actor navigates to the tournaments’ management tab.
  2. The actor navigates to the “Add tournament” button.
  3. A new form “Add tournament” pops up.
  4. The actor fills in the fields with the required information.
  5. The actor presses the “Add” button.
  6. The software sends a success message.
  7. The tournament has been successfully added to the system.
* Extensions:

3a. There is an option where the administrator can stop the action by pressing the “X” sign at the top of the new popped up form.

5a. There is one or more fields with missing of syntactically incorrect information.

1. The system shows a notification informing the actor which information is not appropriate to the required format.
2. Return to MSS step 4.

### UC-07: Removing a tournament from the system

* FR: 01
* Actor: Administrator, Employee, System
* Pre-condition:
* Trigger:
* Main success scenario:
  1. The actor navigates to the tournaments’ management tab.
  2. The actor selects the tournament he wants to remove.
  3. The actor navigates to the “Delete tournament” button.
  4. The system removes the selected tournament from the system.
  5. The system displays a success message, implying the action was successful.
* Extensions:

3a. The system sends a message stating that no tournament has been selected.

1. Return to MSS step 2.

### UC-08: Updating a tournament’s information

* FR: 01
* Actor: Administrator, Employee, System
* Pre-condition:
* Trigger:
* Main success scenario:
  1. The actor navigates to the tournaments’ management tab.
  2. The actor selects the tournaments he wants to make the modifications on.
  3. The actor navigates to the “Edit tournament” button.
  4. A new form “Update tournament” pops up.
  5. The actor fills in the fields with the required information.
  6. The actor navigates to the “Edit” button.
  7. The system sends a success message.
  8. The tournament has been successfully modified.
* Extensions:

4a. There is an option where the administrator can stop the action by pressing the “X” sign at the top of the new popped up form.

6a. There is one or more fields with missing of syntactically incorrect information.

1. The system shows a notification informing the actor which information is not appropriate to the required format.
2. Return to MSS step 5.

### UC-09: Generate a tournament’s structure

* FR: 03,06
* Actor: Administrator, Employee, System
* Pre-condition:
* Trigger:
* Main success scenario:
  1. The actor navigates to the tournaments’ management tab.
  2. The system displays all the tournaments.
  3. The actor navigates to the “Generate Tournament Structure” button.
  4. The system displays a new form with all the rounds generated for that tournament.
* Extensions:

4a. The system announces the actor that the generation cannot be done due to the “Status” of the tournament

1. Enroll more players so the tournament locks itself

2. Go back to MSS step 3.

### UC-10: Add scoring to all matches of a tournament

* FR: 04
* Actor: Administrator, Employee, System
* Pre-condition:
* Trigger:
* Main success scenario:
  1. The actor navigates to the tournaments’ management tab.
  2. The system displays all the tournaments.
  3. The actor navigates to the “Generate Tournament Structure” button.
  4. The system displays a new form with all the rounds generated for that tournament.
  5. The actor presses the first round.
  6. The system displays a drop-down box with all matches.
  7. The actor selects the match he wants to score.

8. The system displays the names of the 2 parties involved in the match.

9. The actor fills in the match results and navigates the “DONE” button.

10. The system displays the matches excluding the already scored ones.

11. The actor repeats the action for all the matches and rounds.

* Extensions:

4a. The system announces the actor that the generation cannot be done due to the “Status” of the tournament

1. Enroll more players so the tournament locks itself

2. Go back to MSS step 3.

### UC-11: Calculate rankings based on the match results

* FR: 01
* Actor: Administrator, Employee, System
* Pre-condition:
* Trigger:
* Main success scenario:
  1. The actor navigates to the tournaments’ management tab.
  2. The system displays all the tournaments.
  3. The actor navigates to the “Generate Tournament Structure” button.
  4. The system displays a new form with all the rounds generated for that tournament.
  5. The actor presses the first round.
  6. The system displays a drop-down box with all matches.
  7. The actor selects the match he wants to score.

8. The system displays the names of the 2 parties involved in the match.

9. The actor fills in the match results and navigates the “DONE” button.

10. The system displays the matches excluding the already scored ones.

11. The actor repeats the action for all the matches and rounds.

12. The system shows that no more matches are to be scored.

13. The actor navigates the “Calculate results” button.

14. The system displays a message stating that the calculation is complete

* Extensions:

4a. The system announces the actor that the generation cannot be done due to the “Status” of the tournament

1. Enroll more players so the tournament locks itself

2. Go back to MSS step 3.

14a. The system announces the actor that the calculation cannot be completed without all the matches being scored.

1. Go back to MSS step 9.

### UC-12: View tournament results for a finished tournament

* FR: 05
* Actor: Player, System
* Pre-condition: The actor is logged on the website
* Trigger: Actor wants to see information about a finished tournament
* Main success scenario:
  1. The actor navigates to the “Past tournaments”
  2. The system displays all the finished tournaments.
  3. The actor navigates the “View details” button for a tournament displayed
  4. The system displays the results of that tournament.

### UC-13: View tournament results for a player’s tournament

* FR: 10
* Actor: Player, System
* Pre-condition: The actor is logged on the website
* Trigger: Actor wants to see information about one of his finished tournaments
* Main success scenario:
  1. The actor navigates to the “My tournaments”
  2. The system displays all his finished tournaments.
  3. The actor navigates the “View details” button for a tournament displayed
  4. The system displays the results of that tournament.

### UC-13: Enroll for a tournament

* FR: 02
* Actor: Player, System
* Pre-condition: The actor is logged on the website
* Trigger: Actor wants to register for a tournament
* Main success scenario:
  1. The actor navigates to the “Tournaments”
  2. The system displays all the unfinished tournaments.
  3. The actor navigates the “Enroll” button
  4. The system shows the actor has been enrolled as the button changes to disabled.
* Extensions:

3a. The system announces the actor that he cannot register for that tournament by disabling the button

1. Go back to MSS step 1