Synthesis Assignment, Week 11

1. Introduction

This document describes the synthesis assignment you will be working on during weeks 11-16. This is an individual assignment where you can show your current proficiency of all seven (7) learning outcomes (LOs).

2. 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-robin*[1] 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.

2.1. Process of running a tournament

The logistics of a tournament can be divided into four (4) different phases.

Phase 1: Creating a tournament

Firstly, an association staff creates a tournament by specifying the following information:

- Sport type (i.e., *badminton*)
- Description information
- Duration information (start and end dates)
- The minimum and maximum number of players
- Location information (i.e., an address)
- Tournament system information (i.e., *round-robin*)

After creating the tournament, it should become available for players to register themselves for it.

Phase 2: Registering players

When a player is interested in participating in a tournament, they can visit the sport association's website, retrieve the list of available tournaments, and register themselves for the one they are interested in.

Note that a tournament is only available for registration when the number of maximum players has not been reached and it is one (1) week before the start date of the tournament.

---

[1] Round-robin is an all-play-all type of tournament system; see https://en.wikipedia.org/wiki/Round-robin_tournament

[2] Badminton is a racquet sport played using racquets to hit a shuttlecock across a net; see https://en.wikipedia.org/wiki/Badminton

Phase 3: The tournament

A tournament only starts when there are enough players (i.e. the minimum number of players). When the minimum number is not reached, it should be shown on the website and the remaining phases are not applicable. This also means that it then should not be possible to, for example, register results of the matches.

If the minimum number of players are met, the software should generate an overview of the games with which players play against each other (e.g. the tournament schedule). This schedule is based on the selected tournament system (i.e. round-robin).

When a given game has been played (i.e. finished) between two players, the results must be registered in the system by a *staff* member.

The official badminton scoring system must be followed/enforced when registering the result:

- A game consists of 21 points
- If the game reaches 20-20 points, the player that gains a 2-point lead wins the game
- At 29-29, the side scoring the 30th point wins the game

Phase 4: Showing tournament results

Any interested party should be able to retrieve the statistics of a given tournament. The following information should be shown:

- Tournament information
- Tournament schedule with the games between players and , if any, the results, of the played games

3. Assignment

Your assignment is to create the following deliverables:

- URS
- Test plan and test report
- UML Class diagram
- Software solution
    - Source code
    - Unit tests
    - Database

Make sure you come to an agreement with your tutor when you make assumptions; the 'openness' of the requirements are by design and are to give you the freedom/flexibility to show us your proficiency towards the LOs.

You can decide how to approach this assignment, just make sure that you can deliver everything at the predefined deadline and that your tutor approves your planning before the end of week 13 (e.g., whether it is feasible).

In the remainder of this document, you can find the constraints and requirements.

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 10th 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.

The 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

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.

Functional requirements

- *FR-01: Manage Tournaments*
  Staff must be able to manage (*CRUD* operations) the tournaments. Make sure to include the information given in *Process of running a tournament Phase 1*.

  This requirement must be implemented in a desktop application.

- *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. Make sure to at least follow the information given in *Process of running a tournament Phase 2*.

  This requirement must be implemented in a web application.

- *FR-03*: *Support generating tournament schedule*
  Staff must be able to generate the tournament's schedule. Make sure to at least follow the information given in *Process of running a tournament Phase 3*.

  This requirement must be implemented in a desktop application.

- *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*. Make sure to at least follow the information given in *Process of running a tournament Phase 3*.

  You can decide whether this should be done in a web or desktop application.

- *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. Make sure to at least follow the information given in *Process of running a tournament Phase 4*.

  This requirement must be implemented in a web application.

Major requirements

Functional requirements

- *FR-06: Support multiple tournement 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:
  - Single-elimination[3]
  - Double-elimination[4]
  - Double round-robin[5]

- *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.

- *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.

  See example given in *Figure 3 of Appendix A* for more detailed information.

Minor requirements

Functional requirements

- *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.

- *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 also can include the individual matches (games) played against different opponents.

- *FR-09: 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

[3] Single-elimination is a type of tournament where the loser is eliminated from the tournament; see: https://en.wikipedia.org/wiki/Single-elimination_tournament
[4] Double-elimination is a type of tournament where participant ceases to be eligible to win the tournament upon having lost two games or matches; see: https://en.wikipedia.org/wiki/Double-elimination_tournament
[5] Double round-robin is a type of tournament where each participant plays all others twice; see: https://en.wikipedia.org/wiki/Round-robin_tournament

can either accept or reject the challenge. If it is accepted, the result of the game can be registered by one of the players.

- *FR-11: 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.

- end of assignment -

Appendix A: An example of completed badminton tournament results

Note: <u>You are not asked</u> to generate this output. Figure 1, Figure 2 and *Figure 3 serve as an example to what information can be extracted and used in the software solution. For example: tournament information, player information, match results, how to determine ranking.*

DuelSys inc.

Tournament information

sport: Badminton
system: Round-robin

location: Badminton Club Eindhoven
start date: 02/May/2022
end date: 06/May/2022

Participants

Chou Tien Chen
Kento Momota
Anders Antonsen
Viktor Axelsen

Player	Results				Points	Rank
Chou Tien Chen		16-21	17-21	19-21	0	4
Kento Momota	21-16		22-20	19-21	2	2
Anders Antonsen	21-17	20-22		15-21	1	3
Viktor Axelsen	21-19	21-19	21-15		3	1

Results

gold: Viktor Axelsen
silver: Kento Momota
bronze: Anders Antonsen

Figure 1 Completed single-game match round-robin tournament

DuelSys inc.

Tournament information

sport: Badminton
system: Round-robin
location: Badminton Club Eindhoven

start date: 23/May/2022
end date: 28/May/2022

#	Players	Points	Rank
1	An Se-young	0	4
2	Akane Yamaguchi	2	2
3	Chen Yufei	1	3
4	Tai Tzu-ying	3	1

#	Matches	Results
1	An Se-young - Tai Tzu-ying	19-21
2	Akane Yamaguchi - Chen Yufei	22-20
3	An Se-young - Chen Yufei	17-21
4	Akane Yamaguchi - Tai Tzu-ying	19-21
5	An Se-young - Akane Yamaguchi	16-21
6	Chen Yufei - Tai Tzu-ying	15-21

Final results

gold: Tai Tzu-ying
silver: Tai Tzu-ying
bronze: Chen Yufei

Figure 2 Completed single-game match round-robin tournament

DuelSys inc.

Tournament information

sport: Badminton
system: Round-robin, two (2) game match
location: Badminton Club Veldhoven

start date: 01/Apr/2022
end date: 10/Apr/2022

#	Players	Match		Rank
		Won	Lost	
1	Chou Tien Chen	1	2	3
2	Kento Momota	2	1	1
3	Anders Antonsen	1	2	4
4	Viktor Axelsen	2	1	2

#	Matches	Results
1	Chou Tien Chen - Viktor Axelsen	21-14, 13-21
2	Kento Momota - Anders Antonsen	21-13, 21-7
3	Chou Tien Chen - Anders Antonsen	17-21, 21-19
4	Kento Momota - Viktor Axelsen	21-14, 21-15
5	Chou Tien Chen - Kento Momota	21-13, 15-21
6	Anders Antonsen - Viktor Axelsen	21-14, 21-14

Final results

gold: Kento Momota
silver: Viktor Axelsen
bronze: Chou Tien Chen

Figure 3 Completed double-game match round-robin tournament