URS DUELSYS

By Arenco Meevissen

30 MEI 2022

Table of contents

Client

DuelSys inc. wants a software solution to allow their customers (sport associations) to manage their sport tournaments. The software needs to at least support a round-robin1 tournament system for badminton with possibilities to support other tournament types and sports for now.

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.

Functional requirements

ID	Requirement	Needs to be in
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. The chosen method is for the desktop to handle this
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
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-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.	

Use cases

Tournaments

Name	TC-1 Create tournaments
Actors	Employee
Conditions	Pre: The actor is on the create tournaments page Post: The actor is on the view tournaments
Scenario	1. System requires data 2. actor gives data 3. System checks data 4. System checks if the tournament can be placed 5. System adds tournament 6. System returns user to last page
Extensions	3.1 Input is incorrect 1. system describes what data parts went wrong 2. system prompts to re-enter valid data 3. The use case goes to step 2 4.1 tournament can't be placed 1. system describes why this can't be placed 2. system prompts to enter different data 3. system goes to step 2

Name	TC-2 Read tournament
Actors	Employee, Player, Anonymous
Conditions	Pre: - Post: -
Scenario	1. Actor determines which tournament to view 2. System checks on valid input and retrieves the info 3. System shows relevant info
Extensions	2.1 invalid input 1. system describes the problem and prompts actor to choose again or a different one 2. case goes to step 1 2.2 There is no tournament with the given input 1. system describes the problem and prompts actor to choose a different tournament 2. case goes to step 1 3.1 There is no/not enough info 1. System determines the problem and describes it to the actor 2. case ends

Name	TC-3 Update tournaments
Actors	Employee
Conditions	Pre: be logged in and on the view employee page Post: -
Scenario	1. Actor clicks on a tournament that needs to be updated 2. System fills already known data in for the actor 3. System prompts actor to change/add data 4. Actor fills in the needed/changed data 5. System validates input 6. System updates tournament 7. System goes to view tournaments 8. System executes TC-2
Extensions	5.1 Input is incorrect 1. system describes what data parts went wrong 2. system prompts to re-enter valid data 3. The use case goes to step 4

Name	TC-4 Delete tournaments
Actors	Employee
Conditions	Pre: Post:
Scenario	1. Actor determines which tournament to delete 2. System reads the necessary data from the selected tournament 3. System determines if it can be deleted 4. System deletes tournament 5. System executes TC-2
Extensions	2.1 system can't read necessary data 1. system gives message with the given error determined by the system 2. system ends case 3.1 system can't delete it 1. system gives message with the given error determined by the system 2. system ends case

Name	TC-5 Show finished tournaments
Actors	Employee, anonymously, players
Conditions	Pre: Post:

Scenario	1. System gets all finished tournaments 2. System determines what needs to be showed 3. System shows the tournaments
Extensions	

Name	TC-6 Show in progress tournaments
Actors	Employee, anonymously, players
Conditions	Pre: Post:
Scenario	1. System gets all in progress tournaments 2. System determines what needs to be showed 3. System shows the tournaments
Extensions	1.1 there are no in progress tournaments 1. system shows message that there are none 2. ends case

Name	TC-7 Show available tournaments
Actors	Employee, anonymously, players
Conditions	Pre: Post:
Scenario	1. System gets all available tournaments 2. System determines what needs to be showed 3. System shows the tournaments
Extensions	1.1 there are no available tournaments 1. system shows message that there are none 2. ends case

Players

Name	PC-1 Register player
Actors	Anonymously
Conditions	Pre: actor is not logged in Post: actor is logged in
Scenario	1. System requests data 2. Actor gives data 3. System validates data 4. System checks if player already exists 5. System adds player 6. System logs actor in
Extensions	3.1 actor gives invalid data 1. system gives message on what part of the data is wrong 2. case goes to step 2

	3.2 actor gives already known email 1. system gives actor a message that email is already taken 2. case goes to step 2 4.1 player already exists 1. system gives actor a message with that the player already exists 2. case goes to step 2

Name	PC-2 sign up for tournament
Actors	Player
Conditions	Pre: needs to be logged in as a player Post:
Scenario	1. Actor determines which tournament to sign up for 2. System reads the necessary info and checks if player can still sign-up 3. System request actor if it is sure to be signed up for this tournament 4. Actor determines if they want to sign up for this 5. System reads input from actor and adds player to the tournament 6. System redirects actor to last visited page
Extensions	2.1 system doesn't get info 1. system gives message with an error with the given error determined by the system 2.case ends 2.2 actor can't sign up 1. system gives message with an error with the given error determined by the system 2.case ends 2.3 actor is already signed up for this tournament 1. system gives message that the actor is already signed up for this tournament 2. case ends 4.1 actor doesn't want to sign up 1. system goes to last page 2. ends case

Name	PC-3 Login
Actors	anonymously, players
Conditions	Pre: needs to be logged out

	Post:
Scenario	1. System requests data 2. Actor gives data 3. System validates data 4. System checks given data with the data in the database 5. System logs actor in
Extensions	3.1 actor gives invalid data 1. system gives message on what part of the data is wrong 2. case goes to step 2 4.1 there are no results 1. system messages actor about wrong input 2. case goes to step 2

Name	PC-4 Login Employee
Actors	employee
Conditions	Pre: needs to be logged out Post:
Scenario	1. System requests data 2. Actor gives data 3. System validates data 4. System checks given data with the data in the database 5. System logs actor in
Extensions	3.1 actor gives invalid data 1. system gives message on what part of the data is wrong 2. case goes to step 2 4.1 there are no results 1. system messages actor about wrong input 2. case goes to step 2 4.2 account isn't an employee account 1. system messages actor on invalid account 2. case goes to step 1.

Name	PC-5 Registering an employee
Actors	employee
Conditions	Pre: needs to be logged in as employee Post: returns to last visited page
Scenario	1. Actor determines which account to "promote" to an employee account 2. System requests data 3. Actor gives data 4. System determines if data is valid and if the account is already an employee

	5. System promotes account to employee
Extensions	1.1 there are no accounts 1. system describes what is wrong 2. case ends 4.1 actor gives invalid data 1. system gives message on what part of the data is wrong 2. case goes to step 2 4.2 account is employee 1. system describes that account is already an employee 2. case stops

Name	PC-6 show account
Actors	employee
Conditions	Pre: needs to be logged in as employee Post:
Scenario	1. Actor determines which accounts to view 2. System checks on valid input and System checks for accounts with given input 3. System determines what to show and shows the tournament
Extensions	2.1 actor gives invalid data 1. system describes what went wrong 2. case goes to step 1 2.2 there are no accounts with the given data 1. system describes that there are no accounts 2. case goes to step 1.

Name	PC-7 updating an account
Actors	employee
Conditions	Pre: needs to be logged in as employee Post:
Scenario	1. Actor determines which account to update 2. System checks on valid input and requests to enter data 3. Actor gives changed data 4. System checks on valid input and updates account
Extensions	2.1 actor gives invalid data

	1. system gives message on what part of the data is wrong 2. case goes to step 1 4.1 actor gives invalid data 1. system gives message on what part of the data is wrong 2. case goes to step 3 4.2 actor gives already known email 1. system describes that email is already taken 2. case goes to step 3

Name	PC-8 deleting an account
Actors	Employee
Conditions	Pre: needs to be logged in as employee Post:
Scenario	1. Actor determines which account to delete 2. System checks on valid input 3. System request conformation from actor 4. Actor gives conformation 5. System deletes account
Extensions	2.1 actor gives invalid data 1. system gives message on what part of the data is wrong 2. case goes to step 1 4.1 actor doesn't want to delete account 1. system messages that account won't be deleted 2. case ends

Schedule

Name	SC-1 Create schedule
Actors	Employee
Conditions	Pre: needs to be logged in as employee Post:
Scenario	1. Actor determines which tournament to make a schedule for 2. System checks if the given tournament can make a schedule for 3. system makes fake schedule 4. actor checks schedule and checks if the schedule is correct

	5. system makes schedule
Extensions	1.1 There are no tournaments available 1. System messages actor that there no available tournaments. 2. Case ends 2.1 The system can't make a schedule because of not enough players 1. system describes that there are not enough players 2. case goes to step 1 2.2 The system can't make a schedule because of not reached max players and more then a week away from the tournament start date 1. system describes that the maximum hasn't been reached yet and that it's still possible that players can sign up for this tournament. 2. case goes to step 1 4.1 actor does not agree with the schedule 1. system messages actor that it will retry 2. case goes to step 2 4.2 actor doesn't want to make a schedule anymore 1. system messages actor of discontinued process 2. case ends

Result

Name	RC-1 Create result
Actors	Employee
Conditions	Pre: Post:
Scenario	1. System requires data 2. actor gives data 3. System checks data 4. System checks if the result can be placed 5. System adds result 6. System returns user to last page
Extensions	3.1 Input is incorrect 1. system describes what data parts went wrong 2. system prompts to re-enter valid data 3. The use case goes to step 2 4.1 schedule can't be placed

	1. system describes why this can't be placed 2. system prompts to enter different data 3. system goes to step 2

Name	RC-2 Read results
Actors	Employee, anonymously, players
Conditions	Pre: Post:
Scenario	1. Actor determines which tournament to view 2. System validates data 3. System gets all results with data given 4. System determines what needs to be showed 5. System shows the result(s)
Extensions	2.1 Input is incorrect 1. system describes what data parts went wrong 2. system prompts to re-enter/enter different valid data 3. The use case goes to step 1

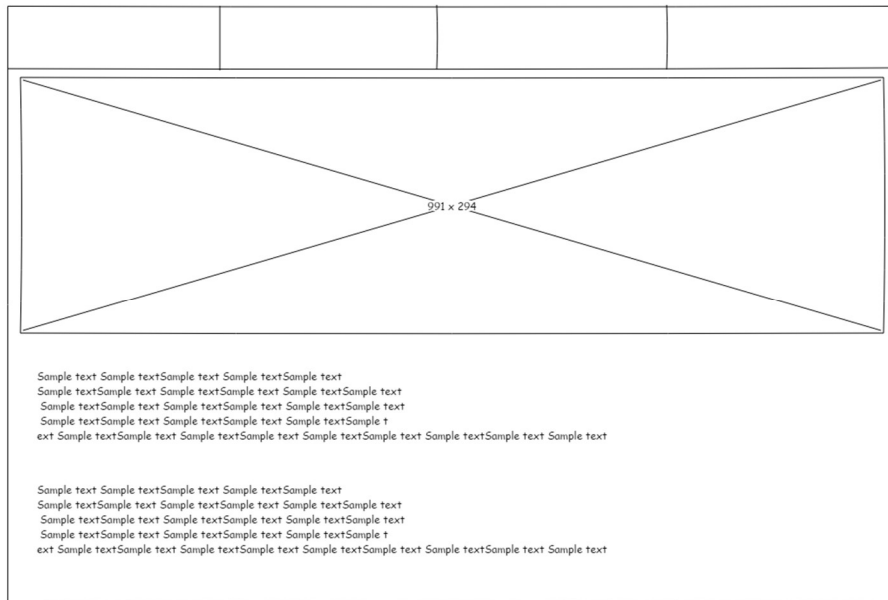| Name	RC-3 Update Results
Actors	Employee
Conditions	Pre: Post:
Scenario	1. Actor clicks on a match that needs to be updated 2. System fills already known data in for the actor 3. System prompts actor to change/add data 4. Actor fills in the needed/changed data 5. System validates input 6. System updates match 7. System goes to view matches
Extensions	5.1 Input is incorrect 1. system describes what data parts went wrong 2. system prompts to re-enter valid data 3. The use case goes to step 4 5.2 Input is a tie 1. System prompts Actor to re-enter valid data 2. The case goes to step 4

Leader board

-

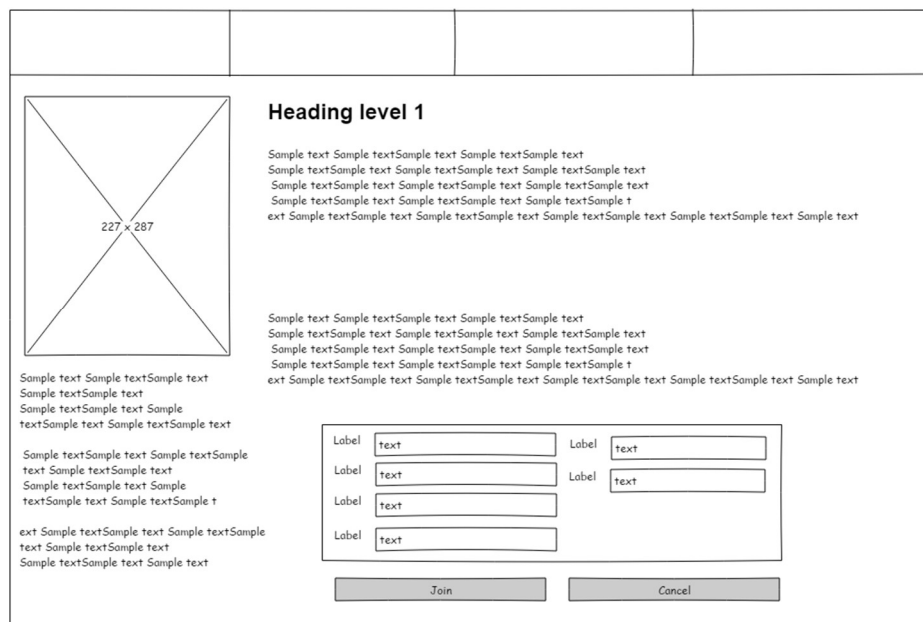| Name	LBC-1 Get leader board
Actors	Employee, anonymously, players
Conditions	Pre: Post:
Scenario	1. Actor clicks on an ongoing/finished tournament and goes to matches 2. System checks if tournament is ongoing or finished. Also, System validates data 3. System gets all results with data given 4. System determines what needs to be showed 5. System shows the result(s)
Extensions	2.1 Input is incorrect 1. system describes what data parts went wrong 2. system prompts to re-enter/enter different valid data 3. The use case goes to step 1

Leader board

-

Wireframes

Web frames

Home

Register player

Tournament info

Category ⌄	text	button

Date	Name
18-9-2017	Sample Name
18-9-2017	Sample Name
18-9-2017	Sample Name
18-9-2017	Sample Name
18-9-2017	Sample Name
18-9-2017	Sample Name
18-9-2017	Sample Name
18-9-2017	Sample Name
18-9-2017	Sample Name
18-9-2017	Sample Name

App frames

Login

Title

Input

Input

Primary

add Tournament

Title

Input

Input

Combo ▾

Secrevit fontes liquidum locoque pronaque?

Illas semine campoque declivia oppida corpora nam inter fuit discordia tellus solidumque iunctarum erat. quae terrenae ubi rerum recessit iudicis aestu fixo

Input

Input

Secrevit fontes liquidum locoque pronaque?

Illas semine campoque declivia oppida corpora nam inter fuit discordia tellus solidumque iunctarum erat. quae terrenae ubi rerum recessit iudicis aestu fixo

Primary   Cancel

Register results

Title

Person A

Person B

Input

Input

Submit