Group 2E-8. First Deliverable. Conceptual Design.

Made by:

Joan Matarredona Coloma

Javier Alfaro Martínez

Enrique Gallifa Tronch

INDEX

1

Identify the task objects from **all previous cases and the case study.** Compile them in tables.

Slides: 3-6

2

Identify the containers of the **use cases assigned to your group** (register coach, delete coach, create tournament, register a pair in a tournament).

Slides: 7-11

3

Create the content diagram of the previous use cases, starting from the main container and adding all the navigation links needed (if needed, also conditions).

Slides: 7-11

Task object	Attributes	Actions
Coach	Name	Register
0.5	Telephone Number	Modify
	Email Address	Delete
	Teaching level accredited	~~Assign a group~~
	Password	
	Login name	
	~~Training group~~ only in group	

Task object	Attributes	Actions
Training Group	Coach	Create
	Court	Delete
0.4	Students list of	Close
0.25	Training shift	Modify coach

level
age interval

2,15 of 3

Task object	Attributes	Actions
Manager	Password	Accredited
0,25	Login name	Modify
	Email address	

Task object	Attributes	Actions
Pair	Pupil1	~~Add to tournament~~ in tournament
0.1	Pupil2	~~Delete from tournament~~
		~~Oversee~~ create
		~~Organize~~
		Modify

Task object	Attributes	Actions
Tournament 0.25	~~Contact address of the~~ list of participant~~s~~	Create / Register pair
	Results list of court	Send results
other object	← Matches	~~Oversee~~
	Number of pairs	Enter results

Task object	Attributes	Actions
Court	Number a identifies them	~~Assign training group~~
0.1	Outdoor/Indoor	~~Assign coach~~
	~~Training group~~	create
	~~Coach~~	modifi
	~~Busy~~	

Task object	Attributes	Actions
Students	Name	Register
0.3	Telephone Number	Delete
	Emai Address	~~Add to training group~~
	Age	~~Remove from training group~~ re training group
	Play Level	~~Add to a pair~~ re pair
	Training shift list of	~~Delete to a pair~~
	Training hour list of	Update
	~~Enrolled in a tournament~~	

Main Container

It only has user functions, which correspond with the use cases

- *Name*: Main
- *Purpose*: Allow user to select among the main features
- *Functions*:
  - ~~Search resources~~
  - ~~View current updates~~
  - ~~Contact support team~~
  - Register coach
  - Delete coach
  - Create tournament
  - Register a pair in a tournament
- *Links*:
  - ▶ Insert coach data
  - ▶ Select coach
  - ▶ Enter date
  - ▶ Select tournament
- *Objects*:
- *Contraints*:

Exactly the same names

Use Case3. Register Coach

Coach was not created !!!

You must respect uc actions

coach

Name: Insert data

Objective: Insert coach data

Functions:

- ■ Requests coach details
- ● Enters coach details

you must specify the fields

Links:

- ▶ Check data

Objects: Coach

Constraints: None

coach

Name: Check data

Objective: Check data is correct

Functions:

- ■ Verifies coach details

Links:

- ▶ ~~Notify~~ insert data
- ▶ notify

Objects: Coach

Constraints: None

Name: Notify

Objective: Notify the user and creates coach

Functions:

- ■ Notifies an error or a successful operation
- ■ creates login and pass
- ■ sends email to the coach with login and pass
- ■ inserts coach

Links:

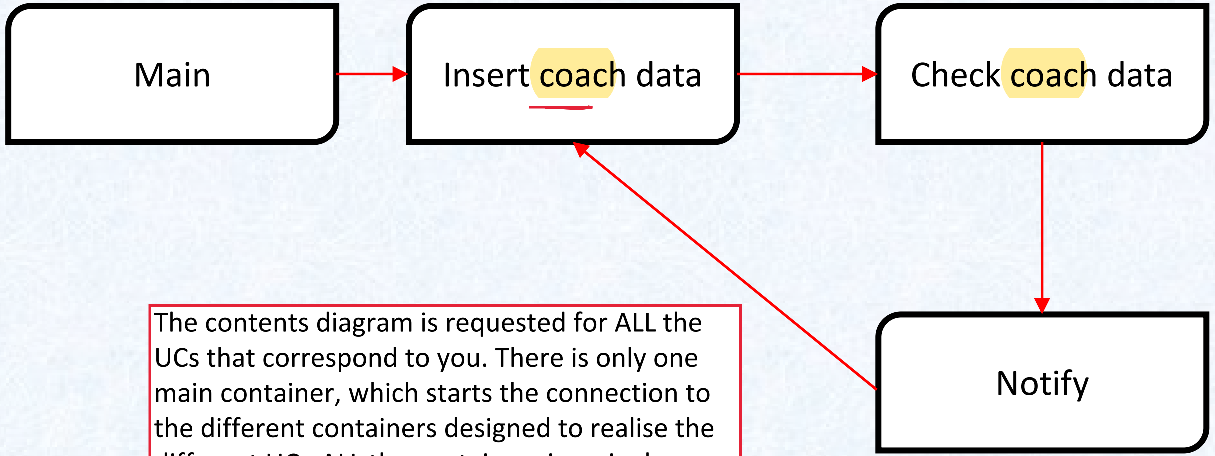
- ▶ ~~Insert data~~

Objects: ~~None~~

Constraints:

- ~~In case of error: it returns to “Insert data”.~~
- ~~If everything is correct : sends an email to the coach with login and password and informs the user.~~

→ ■ functions

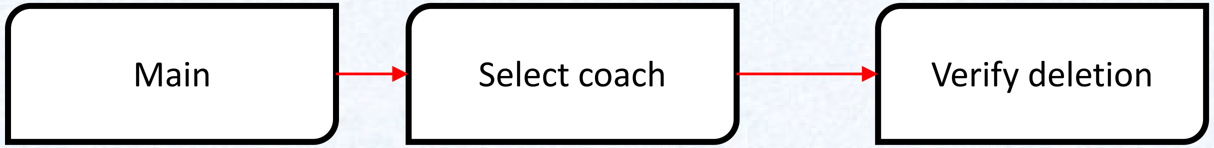


The contents diagram is requested for ALL the UCs that correspond to you. There is only one main container, which starts the connection to the different containers designed to realise the different UC. ALL the containers in a single diagram, for all use cases.
The content diagram is a prototype of the application, so it contains all uc

0.8 of 1.25

Use Case4. Delete Coach

1.25 of 1.25



Name: Select coach

Objective: Select coach to delete

Functions:

- Displays list of coaches not training a group
- Select coach to delete

Links:

- Verify deletion

Objects: Coach

Constraints: None

Name: Verify deletion

Objective: Verifies the coach to be deleted

Functions:

- Requests confirmation
- Verifies that the selected coach is the one the user wants to delete
- Removes the coach

Links: None

Objects: Coach

Constraints: None

Use Case11. Create Tournament

Name: Enter date

Objective: Enter the date and start time of tournament

Functions:

- ■ Asks for the date and star time
- ● Enters the date and time

Links:

- ▶ Validate the date

Objects: Tournament

Constraints: None

Name: Enters number of pairs

Objective: Enters the number of pairs that can register

Functions:

- □ Register number of pairs
- ● Enters the number of pairs

Links:

- ▶ Select the courts

Objects: Tournament

Constraints: None

Name: Validate the date

Objective: Checks date is correct

Functions:

- ■ Verifies date is correct

Links:

- ▶ Enter number of pairs
- ▶ Enter date

Objects: Tournament

Constraints:

- In case of error: it returns to "Enter date".
- If everything is correct : goes to the next step, "Enters number of pairs".

Name: Select the courts

Objective: Select the courts on which the tournament will take places

Functions:

- ■ Informs about free courts for the tournament date
- ● Select the court for tournament

Links:

- ▶ Creates the tournament

Objects: Court

Tournament

Constraints: None

Name: Create the tournament

Objective: Create the tournament confirmed by the user

Functions:

- ■ Calculates and displays the tournament schedule
- ● Confirms thew tournament
- ■ Creates and saves the tournament
- □ warns user

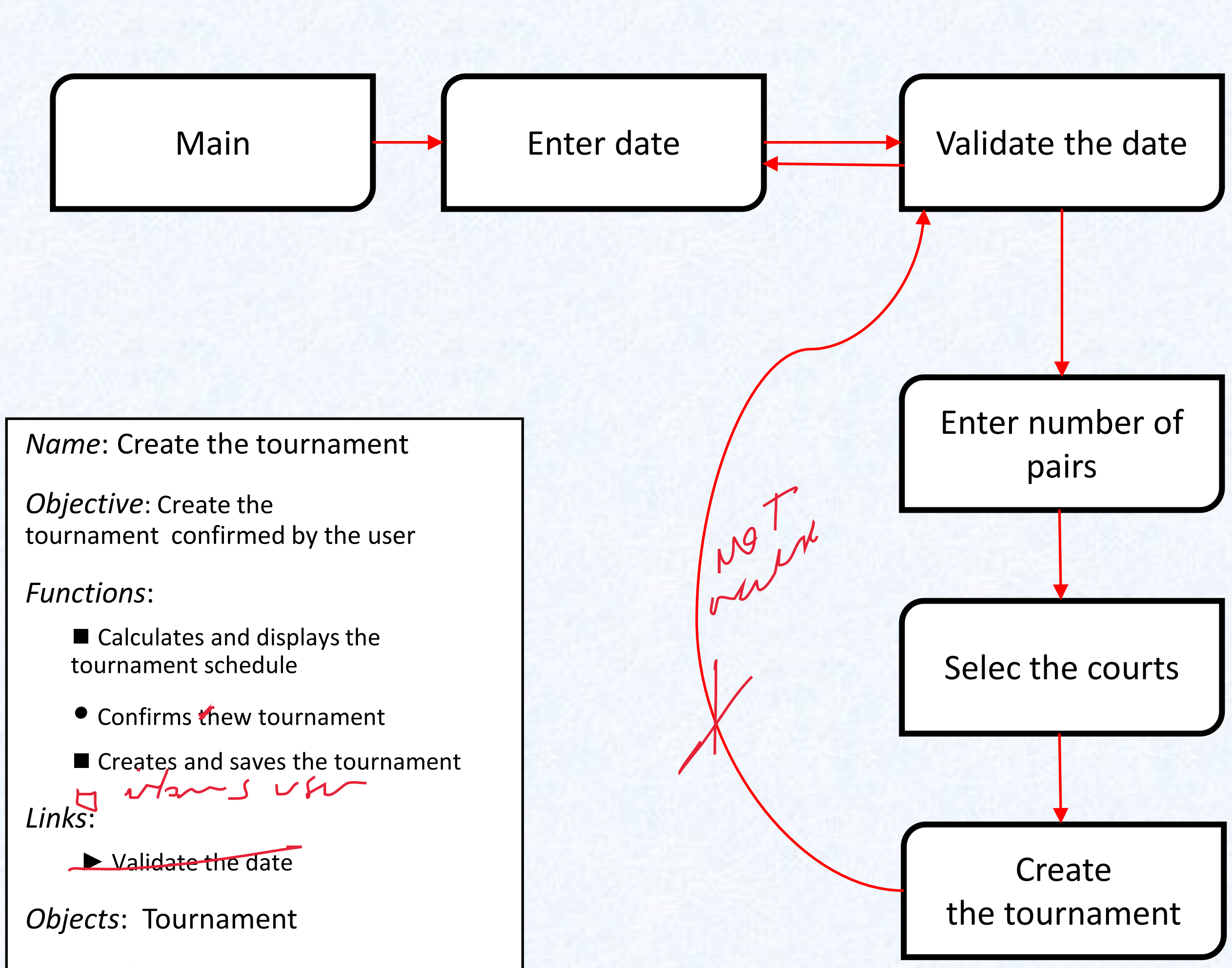
Links:

- ~~▶ Validate the date~~

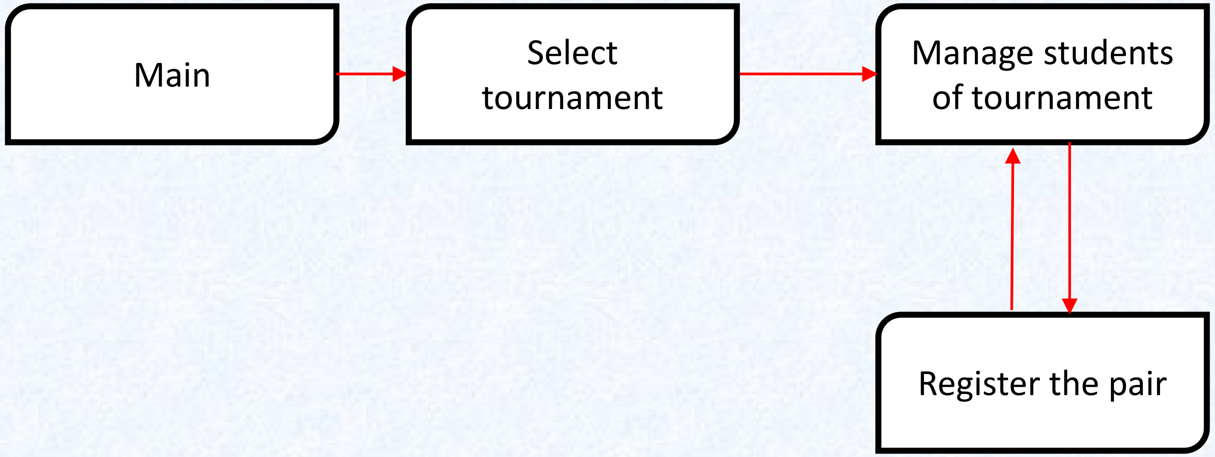
Objects: Tournament

Constraints:

- ~~If not confirmed goes to Validate the date~~
- ~~If confirmed continues~~



Use Case12. Register a Pair in a Tournament



Main

Select tournament

Manage students of tournament

Register the pair

1.25 / 1.25

Name: Select tournament

Objective: Selecting the tournament to register in

Functions:

- ■ Offers all active tournaments and their details
- ● Selects one tournament of the list of all available tournaments

Links:

- ▶ Manage students of tournament

Objects: Tournament

Constraints:

- The tournament must have available places

functional

Name: Manage students of tournament

Objective: Check data is correct

Functions:

- ■ Checks places available
- ■ Offers students not enrolled in the tournament
- ■ Request the selection of two students
- ● Selects two students and ask to register the pair

Links:

- ▶ Register the pair

Objects: Students, Pair, Tournament

Constraints: None

Name: Register the pair

Objective: Checks that two students have been selected

Functions:

- ■ Checks if two students have been selected
- ■ Registers the pair
- □ Request if wants to enter another pair
- ● Confirms to add or not

Links:

- ▶ Manage students of tournament

Objects: Students, pair

Constraints:

- ~~If there are not students selected go back to “Manage students of tournament”.~~

functional