



POLITECNICO

MILANO 1863

SOFTWARE ENGINEERING II

CKB – CodeKataBattle

Acceptance Test

Version 1.0

Feraboli Alessandro, Filippini Marco, Lucca Simone

February 12, 2024

Identification of the project

Authors: Ignazio Iacono, Sara Imparato, Simona Mastroberardino

Repository: <https://github.com/Semproniia/IaconoImparatoMastroberardino>

documents references

Rasd: <https://github.com/Semproniia/IaconoImparatoMastroberardino/blob/main/RASD/RASDv2.pdf>

DD: <https://github.com/Semproniia/IaconoImparatoMastroberardino/blob/main/DD/DDv1.pdf>

Installation setup

In the link to onedrive there were two zip files including the backend and frontend code for the application.

After unzipping the files, it was necessary to download postgres, pgAdmin4 and angular.

Overall, the process was intuitive, even though we could not use the docker-compose file found in the root of the project to start the Postgres server.

Acceptance test cases applied and the corresponding outcome.

1. Registration

Success tests:

- Check registration idempotence:
The registration is idempotent, as it should be.

Problems:

- The system does not accept a short password, but no error message appears to indicate it.

Comment:

- The graphic looks well done, but the errors are not handled in the client.

2. Login

Success tests:

- Login successful

3. Create new tournament.

Success tests:

- New tournament created successfully.

4. Create battle

Problems:

- A battle can be created in a closed tournament.
- A battle can't be created if the minimum number of participants and the maximum number of participants is 1.
- The language of the program is always set to html.

Success tests:

- New battle created successfully under standard circumstances.

5. Register student to battle

Individual participation in a battle when a student is already registered to it.

Test objective: verify that a student cannot register to a battle if he/she was already registered to it.

Precoditions:

1. User is logged in.
2. User is registered to the tournament within the battle.
3. User is already registered in the battle.

Test steps:

1. User navigates to the "My Tournament Section"
2. Student selects the desired tournament
3. Student views the battle for that tournament and finds the desire one
4. Student selects to join it individually
5. Student clicks on the "Confirm" button

Expected result: Student gets an error message saying he/she is already registered in the selected battle.

Actual result: Student gets a confirmation alert saying he/she has successfully joined the battle. In the database, another entry, with the exact same values of another one already created, is generated in the student battle entity so the registration to the battle is now duplicated. To see the consequences of this duplication, other tests should be done regarding quitting battles or student evaluation, to see whether the systems registers in the right way these kinds of event but these features have not been implemented.

Problems: The creation of a battle is not idempotent

6. register to a tournament.

Test objective: verify that a student cannot register to a tournament he/she was already registered to it.

Precoditions:

1. User is logged in
2. User is registered to the tournament

Test steps:

1. User navigates to the “All Tournaments Section”
2. Student selects the desired tournament
3. Student clicks on the “Confirm” button

Expected result: Student gets an error message saying he/she is already registered in the selected tournament.

Actual result: Student gets a confirmation alert saying he/she has successfully joined the tournament. In the database, another entry, with the exact same values of another one already created, is generated in the stud_tournament entity so the registration to the tournament is now duplicated. To see the consequences of this duplication, other tests should be done regarding quitting tournaments or student evaluation, to see whether the systems registers in the right way these kinds of event but these features have not been implemented.

Problems: the registration process is not idempotent.

Additional point on the quality of the documentation and code:

- The main difficulty found by testing this software was the lack of error handling from the client side.
- Many requirements were not implemented.

Effort Spent

| | Feraboli | Filippini | Lucca |
|--------------|-----------------|------------------|--------------|
| <i>hours</i> | 2 | 2 | 2 |