POLITECNICO
MILANO 1863

CodeKataBattle

Acceptance test deliverable documentation

Software Engineering 2 project
Academic year 2023 - 2024

6 january 2024
Version 1.0

Authors:
Tommaso Pasini
Elia Pontiggia
Michelangelo Stasi

Professor:
Matteo Camilli

Contents

1. Introduction

This document has the purpose of showing the results of the acceptance tests performed on the CodeKataBattle platform developed by another team. The tests are performed on the system to verify that the system meets the requirements and the expectations of the customer.

In the following pages it is reported a complete and detailed description of the tests performed, the results obtained and the bugs found during the tests.

1.1 Scope

The project analyzed was developed by the team composed by:

- Maria Lucia Porfido
- Rosanna Iannaccone
- Federica Persico

Team's repository: `https://github.com/marialucia54/IannacconePersicoPorfido`

2. Installation and setup

Although the installation instruction reported on the ITD document are complete and exhaustive, the code presents a README file that briefly explains how to install and run the system (if the machine is already equipped with python), and the major dependencies are automatically installed by simply running the `pip install -r requirements.txt` command.

All the installation went smoothly, and the system was up and running in a few minutes.
The only thing that should be added is a default location for the application, because if the user directly connects to `localhost:8000` as suggested by the server log, no page is shown, and the user has to manually navigate to `localhost:8000/login` to access the login page.

3. Acceptance test cases

The test cases are taken entirely from the use cases reported in the RASD document. The test cases are reported in the following tables:
GITHUB

3.1 Test cases

Name	Registration on CKB platform
Input	The user inserts its credentials and clicks on the registration button
Expected	The user is registered on the platform and can access the main page
Pass/Fail	Pass

Name	Discovering and Participating in Tournaments
Input	The STU navigates on the plaform and selects and joins a tournament
Expected	The STU is now part of the tournament
Pass/Fail	Incomplete
Notes	Once the student selects the tournament on the main page, the system will automatically add the student to the tournament; it would be better to add a confirmation message to the user (as it is expected to happen according to what is written on the RASD use case). Furthermore, nowhere it is specified the submission deadline to the tournament

Name	Creating a Tournament
Input	The EDU creates a tournament and fills in the required fields, then clicks on the create button
Expected	The tournament is created
Pass / Fail	Pass

Name	Granting permissions
Input	In a tournament, created by edu A, A can search for another EDU B and grant him the permission to manage the tournament
Expected	EDU B is now a manager of the tournament
Pass / Fail	Fail: it is not clear what to insert in the text box (email, just name, name and surname, github username...), and by trying to put any ot these it doesn't work anyway

Name	Joining a battle
Input	The STU selects a tournament and joins a battle within It
Expected	The STU is now part of the battle
Pass / Fail	Pass

Name	Viewing tournament rankings
Input	The User selects a tournament and selects the rankings button
Expected	The User can see the rankings of the tournament
Pass / Fail	Pass

Name	Closing a tournament
Input	The EDU selects a tournament and closes it
Expected	The tournament is closed
Pass / Fail	Pass

3.2 Bug reports

There are quite a few recurring bugs in the system, which are listed below:

- The system does not show the message to confirm that a general action (e.g. subscription to a tournament, creation / closure of a tournament, etc.) has been completed: this is as simple as important to the user experience, as it is expected to happen according to what is written on the RASD use case.
- Whenever an error occurs, the system does not show any message to the user, but it simply dumps the python error page on the screen: this is not only bad for the user experience, but it is also a security issue, as it could reveal sensitive information to the user
  The stack dump is very useful for the developers, but it should be removed after the development phase.
- The students aren't always notified relevant events occur (the *Notification* section doesn't show the new entries)
- Qualcuno riesce a connetersi a fucking github?!

3.3 Final Comments

In general, we found the application usable and quite user friendly, with a well-structured code and sufficiently commented in large files

We think that the only problems we found out are due to the short time the project was developed and we are quite sure that, in a real world scenario, after out revision the developers would immediatly fix the bugs we found out; we are not so familiar with python but we still think that it is just a matter of a bunch of lines of code

4. Effort Spent

Team

Topic	**Time**

Table 4.1: Effort Spent during team meetings

Tommaso Pasini

Topic	**Time**

Table 4.2: Effort Spent by Tommaso Pasini

Elia Pontiggia

Topic	**Time**
First part of the use cases	3h

Table 4.3: Effort Spent by Elia Pontiggia

Michelangelo Stasi

Topic	**Time**

Table 4.4: Effort Spent by Michelangelo Stasi