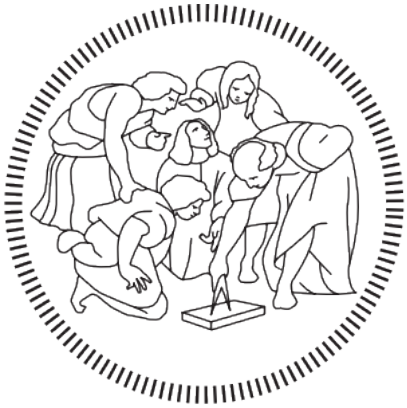

POLITECNICO
MILANO 1863

CodeKataBattle

Implementation and Test deliverable documentation

Software Engineering 2 project
Academic year 2023 - 2024

27 december 2023
Version 0.0

Authors:
Tommaso Pasini
Elia Pontiggia
Michelangelo Stasi

Professor:
Matteo Camilli

Contents

1. Description of the software

This document contains the description of the implementation of the CodeKataBattle platform.

Its purpose is to provide a complete and detailed explanation of the system and its structure to the developers who will work on the project (in our case, the members of other teams and our professors). It is also a useful reference for the developers who will maintain the system in the future.

1.1 Links

In the following list are provided the links to all the documents that are part of the CodeKataBattle project's implementation

- Source Code
- executable ready to be run

1.2 Implemented features

In the following, we list the features and the requirements that have been implemented in the system.

Since the group that has implemented the system is composed of three people, as the assignment required, we have not implemented the features regarding the ganmification aspects of the system

1.3 Adopted development framework

Backend

The system has been developed using the **Spring Boot** framework, which is a framework that allows to create web applications in Java.
The main reason for this choice is that it is a framework often used in this context since it allows to create a web application with REST APIs in a simple way.
In addiction, there are a lot of libraries that can be used to integrate the framework with other technologies, such as the database, which is very useful.

Frontend

For the development of the frontend, we have implemented a single page application simply using **HTML, SCSS and JavaScript**.
Even it is a dated approach, we have chosen this solution because it is the one that we are most familiar with and because it is the one that we have used in the previous projects. Moreover, since the application is very simple, we have not felt the need to use a more complex framework.

Database

The database has been implemented using **MySQL**.
The main reason for this choice is that it is the most used database in the world and it can be easily integrated with the Spring Boot framework thanks to the **Spring Data JPA** library.

1.4 Structure of the source code

The source code of the system can be found in the ITD folder of the repository.
Its structure is the classical one of a Spring Boot application, which is the following:

- `src/main/java/ckb/platform/`: contains the source code of the spring application, divided in the following subfolders:
  - `advices/`: contains the classes that are used to insert into the response the error messages in case of exceptions
  - `controllers/`: contains the classes that are used to handle the requests and are the responsible of the communication between the APIs and the database
  - `entities/`: contains the JPA classes that represent the entities of the database
  - `exceptions/`: contains the classes that are used to handle the exceptions
  - `repositories/`: contains the interfaces that are used to communicate with the database
- `src/main/resources/static/`: contains the files of the frontend application, i.e. the HTML, SCSS and JavaScript files

2. Performed test

3. Installation instructions

3.1 Static Analysis Tool - SonarQube

In order to use this tool, we need to install SonarQube Community Edition at this link. The next step is to install a local instance of SonarQube by following these instructions. In our case we simply run from the zip file. **Java 17 is required to start the local instance of SonarQube**.
Our project details are :

- **Project-name and Project-key** : CKB-platform
- **Host-url** : http://localhost:9000
- **Token** : the token is generated once the server is running

Now we need to install the SonarScanner for CLI and we can find the download here. Once we have installed everything and run the SonarQube server, we can modify our file **sonar-project.properties** with the right token; to analyse our code we have to open a terminal in our root directory and run the following command : `sonar-scanner.bat`. Now we can see our analysis at our host-url.

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
Frontend	21h
Writing of the document	3h

Table 4.3: Effort Spent by Elia Pontiggia

Michelangelo Stasi

Topic	Time
Backend	32 h

Table 4.4: Effort Spent by Michelangelo Stasi