ATD

Federico Valentino, Nicola Zarbo

February 2023

Contents

Chapter 1

Introduction

1.1 Purpose

CodeKataBattle (CKB) is a new platform that helps students improve their software development skills by training with peers on code katas . Educators use the platform to challenge students by creating code kata battles in which teams of students can compete against each other, thus proving (and improving) their skills.
A code kata battle is essentially a programming exercise in a programming language of choice (e.g., Java, Python). The exercise includes a brief textual description and a software project with build automation scripts (e.g., a Gradle project in case of Java sources) that contains a set of test cases that the program must pass, but without the program implementation. Students are asked to complete the project with their code. In particular, groups of students participating in a battle are expected to follow a test-first approach and develop a solution that passes the required tests. Groups deliver their solution to the platform (by the end of the battle). At the end of the battle, the platform assigns scores to groups to create a competition rank.

1.2 Scope

In this document we describe how we went about testing the implementation of the group formed by Matteo Panarotto, Tommaso Tognoli and Oliver Mosgaard Stege.

- Repository URL: _Link

1.3 Definitions, Acronyms, Abbreviations

Acronym	**Definition**
CKB	CodeKataBattle
CK	Code Kata
UI	User Interface

Table 1.1: Acronyms used in the document

1.4 Revision History

1.5 Reference Documents

- RASD Document for CKB from group
- DD Document for CKB from group
- IT Document for CKB from group

1.6 Document Structure

This document is structured as follows:

- Chapter 1 contains the introduction to the document;
- Chapter 2 goes about the installation process;
- Chapter 3 explains how the testing went;
- Chapter 4 contains any additional points worth raising;
- Chapter 5 contains the effort spent by each group member;
- Chapter 6 lists tools and references used;

Chapter 2

Installation Setup

For the installation phase we followed the *Installation instruction* chapter in the IT document.

2.1 Application Server

Since the given JAR files were not working and returned a security exception we decided to run the Application Server executables directly from source inside the Intellij IDEA IDE. Doing this they didn't raise any issues. The problematic part during the installation was the MongoDB and PostgreSQL parts. While they were was relatively easy to install the instructions lacked of any terminal command or whatsoever info on what the exact steps were to create the internal DBs and in case of Postgre how to configure the user.

2.2 Webserver

Installing the Webserver instead was pretty straight forward, the terminal commands were described accurately. Unfortunately due to an untreated CORS policy we couldn't access the web page directly from our default browser and following the guide tips we had to install Google Chrome and launch it while disabling web security. Doing this gave us access to the CKB web platform.

2.2.1 Platform

- Intel Core i5-9600k
- 16 GB RAM
- O.S.: Ubuntu 22.04

Chapter 3

Acceptance Test Cases

3.1 Test Scenarios

3.1.1 Register

1. Goal: Register a new user
2. Steps to reproduce:
   - Open the app
   - Click on ”New user? Sign Up!”
   - Fill out first name, second name, email, password and choose account type
   - Click Sign up button
3. Issues:
   - If a user signs up with the same email as another user then the entry for that user is lost.

3.1.2 Login

1. Goal: Log into platform
2. Steps to reproduce:
   - Open the app
   - Fill out email and password
   - Click Login button
3. Issues:
   - The main page of the application contains information even when the user is completely new to the platform
   - If a students logs in they can’t see or interact with anything

3.1.3 Create Tournament

1. Goal: Create a tournament
2. Steps to reproduce:
   - Login
   - Click on manage tournaments
   - Click on + symbol
   - Fill out the form
   - Create the tournament
3. Issues:
   - Badge creation and scoring function not working
   - Sometimes the confirm button doesn't work and doesn't redirect you to the Tournaments Page
   - The newly created tournament contains information that shouldn't be there

3.1.4 Invite Educator

1. Goal: Invite an educator during tournament creation
2. Steps to reproduce:
   - Login
   - Click on manage tournaments
   - Click on + symbol
   - Fill out the form
   - Insert educator email to invite
   - Create the tournament
3. Issues:
   - Invitations don't work, the other educator can't see their invitations and accept them

3.1.5 Create Battle

1. Goal: Create a battle
2. Steps to reproduce:
   - Login
   - Create a tournament and go to the tournament page

- Fill out the battle creation form
- Click on create battle

3. Issues:
   - Battle creation doesn't work, no new battle gets added to the GUI

3.1.6 Evaluation of Battle Sources

1. Goal: Manually evaluate sources
2. Steps to reproduce:
   - Login
   - Go to a tournament page
   - Click on evaluate battle sources
   - Click on download sources
   - Add a score
   - Click on finalize score
3. Issues:
   - No button works on the evaluate battle sources page
4. Additional Points:
   - This testing was only done since in the tournament page there is always a single battle that needs manual evaluation, otherwise the testing of this aspect wouldn't have happened

3.1.7 Join tournament

1. Goal: As a student join a tournament
2. Issues:
   - Couldn't be tested due to the lack of any real tournament on the main page

3.1.8 Join Battle

1. Goal: As a student join a battle
2. Issues:
   - Couldn't be tested due to the lack of any real tournament on the main page

Chapter 4

Additional Points

Since the Web Application wasn't working as expected we decided to instead run the various test the group made through Postman. This way we would at least ensure the functionalities worked in the backend. The main problem was that the tests were written using particular tokens given at runtime on their machines. Our machines reported different tokens so no tests aside from the login and registering were working. No other form of test were given so, as of today we can't actually say that the implemented features reported on the IT document were really implemented.

Chapter 5

Effort spent

Student	**Time for Implementation**
Valentino	5
Zarbo	5

Chapter 6

References

6.1 Used tools

- GitHub for project versioning
- Overleaf as LaTeX editor
- IntelliJ Idea as Java IDE