CodeKataBattle

Tommaso Pasini
Elia Pontiggia
Michelangelo Stasi

Version alpha

Contents

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of CodeKata-Battle (CKB). It provides functional and non-functional requirements for the development of the system, including use cases, features, user interaction and system constraints. This document is addressed to the developers who have to implement the requirements and could be used as an agreement between the customer and the contractors.

CodeKataBattle (CKB) is a new platform that helps students improve their software development skills by training with peers on code kata. 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 and soft-skills : in fact working in a team is useful to learn soft-skills such as communication and coordination.

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.

1.1.1 Goals

Gi Allows Educators to track Students knowledge about software development

Gi Allows Educators to create coding tournaments and battles to challenge Students

Gi Allows Students to improve their software development skills by partecipating in coding tournaments

Gi Allows Students to improve their soft skills by creating teams for coding battles

redNon son sicuro siano goal che Code kata vuole risolvere

G_i Allows S to learn sw develoment divertendosi e confrontandosi con i compagni in sane competizioni con ranking

G_i At the end of the tournament, the platform shows the final ranks of each team

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1.2 B Scope

1.2.1 World Phenomena

WP_i The students fork the GitHub repository of the code kata and set up an automatic workflow

WP_i The students work on the code kata battle

WP_i The students push their work to the GitHub repository

WP_i An educator upload correct test cases and automation scripts

WP_i An educator evaluates the work done by the team at the end of the code kata battle

1.2.2 Shared Phenomena

SP_i An educator creates a tournament

SP_i An educator grants other colleagues permission to create battles

SP_i An educator creates a battle within a tournament

SP_i The students are notified of a new tournament

SP_i The students are notified of a new upcoming battle within a tournament they are subscribed to

SP_i Students use the platform to form teams

SP_i Student invite other students to join its team respecting the boundaries imposed

SP_i Student join a battle on his own

SP_i Student join a battle towards an invite -¿ Possono scegliere di essere soli oppure devono per forza

SP_i The platform sends the link of the GitHub repository to all students who are members of subscribed teams

SP_i Students are asked to fork the GitHub repository and set up an automated workflow

SP_i The forked repository's workflow notifies the platform of a new push

SP_i The platform updates the battle score of a team

SP_i The educator uses the platform to go through the sources produced by each team

SP_i The educator assigns additional score to the teams after the evaluation

SP_i The platform notifies the teams when the final battle rank becomes available

SP_i The platform updates the personal tournament score of each student

SP_i An educator closes a tournament

SP_i The platform notifies all students involved in the tournament about its end

SP_i All users see the list of ongoing tournaments as well as the corresponding tournament rank

SP_i An educator defines the gamification badges

SP_i Student visualize gained badges on its profile

1.3   C. Definitions, Acronyms, Abbreviations

1.3.1   Abbreviations

G_i = i-th goal WP_i = i-th World Phenomena
SP_i = i-th Shared Phenomena

1.3.2   Acronyms

E = educator S = students

1.4   D. Revision history

1.5   E. Reference Documents

1.6   F. Document Structure

CONTEGGIO ORE: - Condivise: i. W, G, SP, Scenari =¿ 2h30m
   - Elia i.
   - Michelangelo: i.
   - Tommaso: i.

2. Overall Description

2.1 Product Perspective

2.1.1 Scenarios

Creating a tournament

Chip, a professor of Algorithm and Data Structures at Mouseton Institute of Technology, prepared to teach the chapter on strings, launching the "Strings Operations" coding tournament on CKB. To expand participation, he allowed his colleague Dale to create challenges for his software engineering class. Students across classes would compete in string manipulation tasks, ranging from basic concatenation to advanced text analysis, fostering collaboration and learning. To make the tournament more interesting, Chip decided to award badges to the best performing students, so he created a badge for the student who participated in the most tournaments, one for the student who won the most battles and one for the student that wrote the most lines of code. All students already subscribed to CKB were notified of the new tournament, and they could join it from the tournament page.

Creating a battle

In order to get the students familiar with the CKB platform and its features, Chip created an easy battle for his students to practice on, called "word-check", that basically requires the student to implement the game wordle. He decides that the battle will last 2 weeks and that the students will be able to work in teams of 2 or 3 people; they will be able to join the battle until the last day of the battle. In addiction, he wants to give extra points to the code cleanup, so he will have to revise the code of each team at the end of the battle and assign extra points to the teams that wrote clean code. He sets all this information in the battle creation form and then he creates the battle.

Joining a battle

Huey and Dewey, two students of Chip's class, are notified of a new battle and decide to join it. Since the the more the merrier, they decide to invite their friend Louie to join them in the battle. After the registration deadline, they are notified that the battle is about to start and they are given the link to the GitHub repository of the battle to fork and set up the automated workflow in order to be able to link their GitHub account to the CKB platform. After the automated workflow is set up, they are ready to start working on the battle.

Improving the score and obtaining a badge

Donald is a warrior of the "wordcheck" battle and he is working on the battle alone. After a fitst commit, he logs in to the CKB to check his score. He sees that he is in the 3rd position and that he is 10 points behind the leader group, that is composed by Huey, Dewey and Louie. Fortunately, the battle is still open and the CKB platform allows him to improve his score by pushing new commits to the GitHub repository, so he decides to work on the battle for a couple of days and then push his work to the GitHub repository. After checking his score again, he sees that he is now in the 1st position and moreover he obtained a badge for being the first to reach 100 points in the battle and now both students and professors can see this badge when they visit Donald's profile.

Closing a battle

When the deadline for the battle created by Scrooge is reached, all participants are notified that the battle is closed and that they can't push new commits to the GitHub repository. Scrooge is notified that the battle is closed and he can now evaluate the code of each team and assign extra points for the clarity of the comments and the code, as he decided when he created the battle. After the evaluation, the final rank of the battle is available to all participants and the students are notified that they can now see the final rank of the battle.