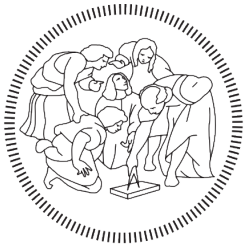
Software Engineering 2 - Prof. Di Nitto Elisabetta
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

CodeKataBattle

RASD
Requirement Analysis and Specification Document

November 1, 2023

Marco Cerino (matricola)
Mattia De Bartolomeis(matricola)



POLITECNICO
MILANO 1863

Contents

1 Introduction

1.1 Purpose

In the context of education and development in the field of programming, students often face a series of challenges. The process of improving software development skills, both for beginners and more experienced students, requires a rigorous and structured approach.

The traditional method of learning based on theoretical lessons and assigned tasks can sometimes be limited in its effectiveness, as students may not have the opportunity to concretely apply what they have learned. Theory and practice must be integrated synergistically to ensure significant growth in software development skills.

CodeKataBattle (CKB) is an innovative response to these challenges in software learning and development. The CKB platform represents a revolutionary solution for students eager to enhance their programming skills. CKB is designed to transform the learning process into a collaborative and practical experience.

Thanks to CKB, students have the opportunity to engage in real code battles, solving programming exercises and overcoming a series of specific tests. These battles allow students to apply the theoretical knowledge they have acquired, putting into practice what they have learned through a series of specific challenges.

This document represents the RASD for the CodeKataBattle (CKB) system, providing a description focused on the system's requirements and specifications. It illustrates scenarios and use cases to detail the system's features, interactions with interested actors, and the limitations it is subject to.

1.1.1 Goal

The system is characterized by the following goals:

G1 Educators can create coding tournaments and battles

G2 Students can form teams for coding battles

G3 Students can participate in coding battles

G4 Students are notified about battles and tournaments

G5 Educators are able to evaluate manually the projects of the student

G6 Projects are evaluated in an automated way based on functional aspects, timeliness, and quality level of the sources

G7 Educators and students can see the rank of the battles and tournaments

---

1.2 Scope

The main human actors in this system are students and educators. Educators use this platform to challenge students by creating competitions where groups of students compete against each other, demonstrating and improving their skills. The challenges consist of a programming exercise in a chosen language (such as Java or Python). Students must follow a "test-first" [1]approach, writing code to pass provided tests.

[1]Test-first: The "test-first" approach involves writing tests before implementing the code, promoting test-driven development.

There is also a non-human actor that plays a crucial role in the platform : Github. GitHub plays a central role in the CodeKataBattle (CKB) for hosting battles and facilitating collaboration among students. It enables automated evaluations through GitHub Actions, tracking teams' progress and updating battle scores in real-time.
Here's how the system works: a teacher creates a "battle" following specific steps. They upload the problem description and the project to CKB, set the minimum and maximum number of students per group, define deadlines for registration and project submission, and set evaluation criteria. Once enrolled in a "battle," students form teams and start working on the project. The platform integrates GitHub, a source code management service, to facilitate collaboration.
Whenever students upload a new version of their code, the platform automatically calculates the team's score, considering aspects such as the number of passed tests, timeliness of submissions, and code quality. The automated assessment also includes static code analysis to evaluate aspects like security, reliability, and maintainability. Teachers can assign personal scores based on students' performance.
At the end of each "battle," the platform updates scores and displays the updated ranking. Students and teachers can monitor progress during the challenge. After the final submission deadline, a consolidation phase takes place, which may involve manual assessment by teachers. Once this phase is complete, all involved students are notified of the final "battle" ranking.
Scores obtained in each "battle" contribute to the overall tournament score for each student. These scores are used to create an overall tournament ranking, showcasing students' performance compared to their peers.

1.2.1 World phenomena

WP1	Educators evaluate the projects
WP2	Students solve the challenges

1.2.2 Shared phenomena

ID		Controlled by
SP1	Students invite other students to join their group	world
SP2	Students push a new commit into the main branch of their repository	world
SP3	Students can view the battle rank	world
SP4	Students can view the tournament rank	world
SP5	The educator uploads the battle	world
SP6	Students register for the battle	world
SP7	Students fork GitHub repository	world
SP8	Educator close the tournament	world
SP9	The professor sets the minimum and maximum number of students per group	world
SP10	The professor sets a registration deadline	world
SP11	The professor sets a final submission deadline	world
SP12	The professor configures additional scoring parameters	world
SP13	Educator use the CKB platform to see the sources produced by each team	world
SP14	Students receive the challenge	machine
SP15	System notify user about tournament creation	machine
SP16	System notify user about the final battle rank	machine
SP17	System notify user about the creation of a new battle	machine

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- GitHub - GitHub is a web platform that provides developers with a space to host and collaborate on software projects, manage source code, and facilitate the review and integration of changes.
- Fork - A fork on GitHub is the creation of an independent copy of an existing repository, allowing developers to make changes without directly affecting the main project.
- GitHub Repository - GitHub repository is an online storage space for a software project, including files, documentation, and version history, facilitating collaboration among developers and code management.

- Commit - A commit is the act of recording changes made to files in a repository, confirming them in the Git version control system, and providing a descriptive message that traces the history of the changes.

1.3.2 Acronyms

- CKB - Code Kada Battle.
- RASD - Requirement Analysis and Specification Document.
- UI - User interface.
- UML - Unified Modelling Language.

1.3.3 Abbreviations

- WP - World Phenomena.
- SP - Shared Phenomena.