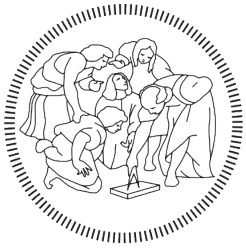
Software Engineering 2 - Prof. Di Nitto Elisabetta
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

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.

1.4 Revision history

1.5 Reference Documents

The creation of this document is based on :

- Slides of Software Engineering 2 course

1.6 Document Structure

This document is composed of six sections:

1. **Introduction**: In this section, we present an overview of the document's content and structure. We outline the problem at hand and articulate the goals that the system aims to achieve. The subsection on scope delves into a comprehensive description of various world and shared phenomena relevant to our subject matter. Additionally, essential information is provided to facilitate proper understanding, including definitions and abbreviations.
2. **Overall Description**: This section offers a holistic description of the system. We provide insights into the system's architecture and functionality, accompanied by an outline of the users and their primary functionalities. Domain diagrams are introduced to illustrate key components, and various scenarios are detailed to enhance comprehension. Finally, we present the fundamental assumptions within the system's domain.
3. **Specific Requirements**: In this section, we delve into the specific requirements of the system. Both functional and non-functional requirements are elucidated to provide a comprehensive understanding. Use case diagrams are included to visually represent system interactions, accompanied by detailed descriptions of each use case and related sequence diagrams. The section concludes with a mapping of requirements onto both overarching goals and individual use cases.
4. **Formal Analysis Using Alloy**: Here, we conduct a formal analysis of the system using Alloy. This includes a rigorous examination of the system's specifications and constraints to ensure coherence and reliability.
5. **Effort Spent**: In this section, we provide an estimate of the effort invested by each member of the group. This allows for a transparent understanding of the distribution of work within the team.
6. **References**: The document concludes with a comprehensive list of references used during the research and development phases. This section serves as a valuable resource for readers seeking additional information or verification of the document's content.

2 Overall Description

2.1 Product perspective

2.1.1 Scenarios

A. Educator Registration
Marco, docente presso l'Università di Napoli, ha recentemente scoperto CKB tramite un collega. Attratto dalle potenzialità della piattaforma nel migliorare le competenze di sviluppo software degli studenti, Marco ha deciso di iscriversi come educatore. Recatosi sulla home page di CKB ha selezionato l'opzione "Registrati" e ha inserito il suo indirizzo email, scelto una password e fornito dettagli di base sul suo profilo, inclusi nome e cognome. Successivamente, ha completato la procedura selezionando l'opzione "Iscriviti come educatore". Una volta registrato con successo, Marco è stato reindirizzato alla sua dashboard personale all'interno di CKB. Qui, ha avuto accesso a una panoramica delle funzionalità per educatori, inclusa la possibilità di creare nuovi tornei e battaglie.

B. Student Registration
Andrea, uno studente dell'Università di Bologna, ha conosciuto l'applicazione CKB grazie al consiglio del suo professore. Affascinato dall'opportunità di affinare le sue competenze di sviluppo software attraverso sfide di code kata, Andrea ha deciso di seguire il suggerimento del docente e di iscriversi alla piattaforma. Recatosi sulla home page di CKB ha selezionato l'opzione "Registrati" e successivamente ha fornito il suo indirizzo email, creando una password. Inoltre, ha inserito il suo nome e cognome, selezionando l'opzione "Iscriviti come studente" per completare la registrazione. Tuttavia, durante il processo di registrazione, ha ricevuto un messaggio di errore. Il sistema CKB ha segnalato che l'indirizzo email inserito era già presente nel sistema. Andrea, prontamente, ha corretto l'indirizzo email, assicurandosi di utilizzare un account valido. Una volta completato con successo il processo di registrazione, Andrea è stato reindirizzato alla sua dashboard personale sulla piattaforma CKB. Qui, ha potuto esplorare le varie funzionalità dedicate agli studenti, come i tornei attivi, le competizioni imminenti e le informazioni sul suo progresso nella piattaforma.

C. Tournament Creation
Daniele, professore di informatica presso il Politecnico di Milano, desidera creare un torneo sulla piattaforma CKB per permettere ai suoi studenti di sviluppare e migliorare le loro competenze di programmazione attraverso la partecipazione a battaglie di code kata.

Daniele accede alla piattaforma utilizzando le sue credenziali di accesso come professore e naviga alla sezione dedicata alla creazione di tornei sulla piattaforma CKB.

Successivamente, avvia il processo di creazione di un nuovo torneo selezionando l'opzione appropriata dalla dashboard principale. Il sistema richiede a Daniele di inserire informazioni sul torneo, come il nome, una data di scadenza per la registrazione al torneo da parte degli studenti e la lista di colleghi che possono creare battaglie all'interno di tale torneo.

Il professore completa il modulo e, dopo aver inserito le informazioni richieste, conferma la creazione del torneo. Il sistema verifica la validità delle informazioni fornite e, nel caso il nome del torneo risulti già esistente, invierà il seguente warning: "Il nome del torneo è gia esistente". Altrimenti, in caso di verifica positiva, il sistema registra il nuovo torneo all'inteno del sistema. Tutti gli studenti iscritti alla piattaforma ricevono notifiche sulla creazione del nuovo torneo.

D. Battle Creation
Giovanni, un professore con privilegi di creazione di battaglie all'interno del torneo "Algorithms and data structures" sulla piattaforma CKB, decide di creare una nuova battaglia. A tale scopo,dopo aver effettuato l'accesso, naviga alla sezione dedicata alla creazione di battaglie e avvia il processo di creazione di una nuova battaglia. Il sistema richiede a Daniele di inserire informazioni essenziali sulla battaglia, tra cui una breve descrizione testuale, il progetto software con gli script di automazione della build, il numero minimo e massimo di studenti per gruppo, la data di scadenza per la registrazione e per la consegna del progetto. Inoltre Daniele imposta informazioni aggiuntive che andranno ad incidere sulla valutazione del punteggio, come sicurezza, mantenimento e affidabilità. Infine, il sistema integra la nuova battaglia nella piattaforma. Gli studenti iscritti al torneo pertinente ricevono notifiche riguardo alla prossima battaglia.

E. Tournament registration
Sarah, una studentessa di ingegneria informatica, si ritrova desiderosa di una nuova sfida per

elevare le sue abilità di programmazione. Decide allora di collegarsi alla piattaforma CKB. Dopo aver effettuato l'accesso, esplora nella sezione dedicata i tornei disponibili e ne individua uno in particolare - "Python Challenge". Sarah clicca sul torneo e legge la sua descrizione. Senza esitazione, decide di partecipare alla Python Challenge. Cliccando su 'Iscriviti al torneo' Sarah fa ufficialmente parte del torneo. A questo punto può visualizzare tutte le battaglie in programma all'interno del torneo e verrà anche notificata alla creazione di battaglie future .

F. Battle registration
Leonardo, studente iscritto al torneo "Super Tournament" su CKB, riceve una notifica che cattura la sua attenzione: "Chiamata alla Battaglia - Python Coding Challenge". Volentoroso di mettere alla prova le sue abilità di programmazione, Leonardo, dopo aver effettuato l'accesso, entra nella sezione dedicata alla battaglia, legge con attenzione i dettagli forniti e decide di iscriversi alla battaglia cliccando su "Iscriviti alla battaglia". Successivamente la piattaforma permette a Leonardo di scegliere se partecipare individualmente o invitare altri colleghi a formare un team per la battle, rispettando il numero minimo e massimo di partecipanti consentiti. Dopo la scadenza della fase di registrazione, la piattaforma CKB genera automaticamente una repository dedicata per il progetto della "Python Coding Challenge". Successivamente, il link diretto alla repository viene consegnato ai membri del team e reso disponibile nella sezione "Le tue battaglie" dell'applicazione.

G. Face the battle
Il team "CodeCrafters", composto da appassionati studenti desiderosi di immergersi nella battle, riceve tramite mail il link alla repository su GitHub dedicata alla sfida "Algoritmi Avanzati" su CKB. Con un semplice click, procedono con il fork di tale repository, contenente il codice kata, dando così vita al loro spazio di lavoro virtuale. Per garantire una valutazione continua e precisa del loro lavoro, i CodeCrafters configurano GitHub Actions, che assicura una tempestiva comunicazione con la piattaforma. Immersi nella stimolante sfida degli "Algoritmi Avanzati", i CodeCrafters avviano il processo di sviluppo adottando l'approccio test-first. Con creatività, creano le prime implementazioni, le sottopongono a test e le committano nella repository principale, tracciando così ogni passo del loro iterativo percorso. Ogni push prima della scadenza della battaglia attiva la piattaforma, che valuta automaticamente gli aspetti funzionali e la tempestività del lavoro dei CodeCrafters. Successivamente, la piattaforma calcola e aggiorna il loro punteggio di battaglia, offrendo feedback in tempo reale sulla qualità del loro prezioso contributo.

H. Ranking display
Matteo, durante la sua partecipazione a una battaglia su CKB, accedendo alla sezione del Torneo e della battaglia d'interesse può monitorare in tempo reale la posizione della sua squadra attraverso una classifica dinamica. Al termine della battaglia, si trova di fronte a una fase di consolidamento, durante la quale l'educatore può decidere se assegnare un punteggio personale al progetto o affidarsi esclusivamente al punteggio automatico. Alla conclusione di questa fase, Matteo può consultare la classifica finale recandosi nella sezione dedicata alla battaglia appena conclusa, ottenendo così una panoramica completa delle prestazioni della sua squadra.

Il punteggio ottenuto in questa specifica battaglia si somma a quelli accumulati nelle battaglie precedenti, contribuendo a formare il suo punteggio complessivo nel torneo. Matteo e tutti gli utenti hanno la possibilità di esplorare la classifica nella sezione dedicata al torneo stesso , accessibile a tutti gli utenti interessati.

Infine, una volta che l'educatore chiude definitivamente il torneo, Matteo e gli altri iscritti a CKB possono consultare la classifica finale del torneo sempre nella sezione relativa a quel torneo.

I. Evaluates project
Una volta scaduta la fase di sottomissione di una battaglia del torneo "Algoritmi Avanzati" su CKB, il Professore Bianchi, creatore della sfida, si prepara ad eseguire una valutazione manuale dei progetti presentati dai team partecipanti.Accede alla piattaforma, seleziona il torneo e la battaglia appena conclusa. All'interno dell'interfaccia, trova una lista completa dei link alle repository degli studenti con la quale può accedere ai diversi progetti. Inizia esaminando attentamente il codice sorgente prodotto da ciascun team, analizzando gli aspetti che non possono essere completamente valutati in modo automatico. Durante questa fase, il Professore Bianchi attribuisce un punteggio personale a ciascun team in base alla sua esperienza e conoscenza approfondita, in particolare premiando la creatività, l'ingegnosità e l'approccio strategico dei team partecipanti.