POLITECNICO
MILANO 1863

Design Document

Software Engineering 2
CodeKataBattle

Authors
Christian Rossi - 10736464
Kirolos Sharoubim - 10719510

Contents

Chapter 1

Introduction

1.1 Purpose

This document aims to provide a comprehensive insight into the CodeKataBattle system outlined in the RASD. It delves into the system's architecture, elucidating on its components, their interactions, processes, and algorithms designed to meet RASD requirements. Furthermore, it offers explicit instructions pertaining to the implementation, integration, and testing plan. Geared towards developers, testers, and project managers, this document serves as a valuable reference for the system's implementation phase.

1.2 Scope

CodeKataBattle serves as a platform utilized by educators to engage students in coding Katas - challenges designed to be addressed using a programming language specified by the challenge organizers. Educators possess the capability to establish a code Kata within a particular tournament, defining essential parameters such as the problem statement (inclusive of test cases), the minimum and maximum number of students allowed per group, and the deadlines for both code Kata registration and solution submission. Following the tournament creation, students are empowered to create groups and commence their collaborative efforts on the solution, following a test-first approach. As the ultimate deadline approaches, the system autonomously computes the final rankings, ultimately revealing the victorious participants. For a more comprehensive overview of the features accessible to end users, please consult the RASD. The architecture of the S2B is structured into three physically separated layers, each installed on distinct tiers. These layers are:

1. *Presentation Layer*: Responsible for overseeing the presentation logic and handling all interactions with end users.

2. *Business Logic Layer*: Manages the application functions provided by the S2B, ensuring seamless operation and functionality.

3. *Data Layer*: Handles secure storage and facilitates access to data, ensuring the integrity and reliability of information.

1.3 Definitions, acronyms and abbreviations

1.3.1 Definitions

Code Kata: adaptation of the concept of karate katas, where you repetitively refine a form, to the realm of software development, fostering iterative practice and improvement.
Test-first approach: software development process relies on the transformation of software requirements into test cases before the software is completely developed, and it involves monitoring the entire software development by iteratively testing the software against all these test cases.

1.3.2 Acronyms

CKB: CodeKataBattle
CK: Code Kata

1.4 Revision history

Version 1.0 - Release - date 22/12/2023

1.5 Reference documents

Document 1 - Presentation about RASD structure
Website 1 - http://codekata.com
Website 2 - https://en.wikipedia.org

1.6 Document structure

This document comprises seven sections:

1. *Introduction*: This section furnishes an overview of the Design Document (DD), encompassing the project's scope, key term definitions, references to pertinent documents, and a brief outline of the design.

2. *User Interface Design*: Outlining the design of the user interface (UI) along with user experience (UX) flowcharts, this section provides a detailed perspective on how users will interact with the system.

3. *Architectural Design*: Describing the high-level components and interactions within the system, this section includes a component view, deployment view, runtime view, and insights into selected architectural styles and patterns.

4. *Requirements Traceability*: Establishing a clear connection between the requirements specified in the Requirements and Specification Document (RASD) and the components outlined in the DD, this section ensures traceability throughout the design process.

5. *Implementation, Integration, and Test Plan*: This section details the plan for implementing, integrating, and testing the system. It includes the sequence in which subsystems and components will be implemented, providing a roadmap for the development process.

6. *Effort Spent*: Offering insights into the effort invested in the design process, this section provides information on the resources and time dedicated to the various aspects of the design.

7. *References*: This section encompasses a list of references cited within the Design Document, providing a foundation for further exploration and understanding.

Chapter 2

Architectural design

2.1 Overview

2.2 Component view

2.3 Deployment view

2.4 Runtime view

2.5 Component interfaces

2.6 Selected architectural styles and patterns

2.7 Other design decisions

Chapter 3

User interface design

Chapter 4

Requirements traceability

Chapter 5

Implementation, integration and test plan

Chapter 6

Effort spent

Chapter 7

References