Requirement Analysis and Specification Document

Christian Rossi Kirolos Sharoubim

Academic Year 2023-2024

Contents

1	Intr	$\operatorname{roduction}$ 3								
	1.1	Purpose								
	1.2	Scope								
	1.3	Definitions, acronyms and abbreviations								
		1.3.1 Definitions								
		1.3.2 Acronyms								
		1.3.3 Abbreviations								
	1.4	Revision history								
	1.5	Reference documents								
	1.6	Document structure								
2	Ove	erall description								
	2.1	Product perspective								
		2.1.1 Scenarios								
		2.1.2 Domain class diagrams								
		2.1.3 State diagrams								
	2.2	Product functions								
		2.2.1 First function								
	2.3	User characteristics								
		2.3.1 Student								
		2.3.2 Educator								
	2.4	Assumptions, dependencies and constraints								
3	Spe	pecific requirements 10								
	3.1	External interface requirements								
		3.1.1 User interfaces								
		3.1.2 Hardware interfaces								
		3.1.3 Software interfaces								
		3.1.4 Communication interfaces								
	3.2									
	3.3									
	3.4	Design constraints								
		3.4.1 Standards compliance								
		3.4.2 Hardware limitations								
		3.4.3 Any other constraint								
	3.5	Software system attributes								
		3.5.1 Reliability								
		3.5.2 Availability								

	3.5.3	Security	. 10
	3.5.4	Maintainability	. 10
	3.5.5	Portability	. 10
4	Formal an	nalysis with Alloy	11
5	Effort spe	ent	12
6	Reference	es	13

Introduction

1.1 Purpose

CodeKataBattle is a platform employed by educators to challenge students on code Katas, problems meant to be tackled using a programming language selected by the organizers of the challenge.

Specifically, educators have the capability to establish a code Kata within a particular tournament by defining the following parameters: the problem to be solved (with some test cases), the minimum and maximum number of students per group, and the deadlines for code Kata registration and solution submission.

After the creation of the tournament, students can form groups and start their work on the solution, employing a test-first approach. When the ultimate deadline approaches, the system autonomously calculates the final rankings and unveils the winners.

1.2 Scope

To ensure the proper development of this application, it is crucial to examine all imaginable phenomena linked to the system.

These phenomena can be categorized into three distinct categories: world phenomena (occurring in the external environment beyond the machine's control), machine phenomena (taking place within the machine and beyond the influence of the external world), and shared phenomena (interactions and events involving both the system and the external world).

Regarding CodeKataBattle, we encounter the following external phenomena:

- 1. The educator generates a code Kata problem description.
- 2. The educator formulates a series of test cases for the code Kata problem.
- 3. The students develop a program to solve the code Kata.
- 4. The educator assigns a personal score based on code quality.

Additionally, the following internal phenomena arise within the system:

- 1. The system creates a GitHub repository.
- 2. The system establishes an automated workflow through GitHub Actions to notify CKB about new commits.

- 3. The system analyzes the repositories at every commit.
- 4. The system conducts automated evaluation and updates the score according to the following criteria:
 - (a) Number of passed test cases.
 - (b) Timeliness between the start of the challenge and the last commit on the main branch.
 - (c) Static code analysis for security, reliability, and maintainability.

Lastly, the following shared interactions and events emerge:

- 1. The educator initiates a battle on the platform, including the following steps:
 - (a) Uploading the code Kata.
 - (b) Specifying the minimum and maximum number of students per group.
 - (c) Setting a registration deadline.
 - (d) Setting a final submission deadline.
 - (e) Configuring additional scoring parameters.
- 2. Students submit their implementations to the platform via GitHub commits.
- 3. The educator creates a tournament.
- 4. The educator grants permissions to other educators to create code Kata within a specific tournament.
- 5. Students subscribe to specific tournaments.
- 6. The student joins an already existing group.
- 7. The student forms a new group and extends invitations to other students.
- 8. The educator concludes the tournament, and the system notifies the students.

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

1.3.3 Abbreviations

1.4 Revision history

Version 1.0 - Release - date TBD

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 contains the following elements:

- 1. **Introduction**: this section offers a general overview of CodeKataBattle and its objectives.
- 2. **Overall description**: this section provides comprehensive information about the system, including interfaces, constraints, domain assumptions, software dependencies, and user characteristics.
- 3. **Specific requirements**: this section outlines the system's functionalities through scenarios and use case diagrams.
- 4. **Formal analysis with alloy**: this section contains the Alloy employed to verify the platform's correctness.
- 5. **Effort spent**: this section details the amount of time (in hours) contributed by each group member.
- 6. **References**: this section lists the tools used in the document's development.

Overall description

2.1 Product perspective

2.1.1 Scenarios

1. New user joins the platform

Maria is a Computer science student. Maria wrote a simple program in Java for a school project a month ago, and realised she really enjoys it. Thus, she is looking for a chace to improve her Java skills as she didn't feel always confident about all the code she wrote. While searching online, Maria finds the CodeKataBattle platform and, as some challenges were exactly what she was looking for, she signs up: she inserts her email and a password and procedes as a Student. A verification email is sent and, once the mail is verified by clicking the link in the verification email, she can start browsing the available tournaments looking for an interesting battle.

2. Educator creates a new tournament

Anne is a professor in an engineering university. She teaches a course about optimization of problems' solution. One day she hears from a colleague of her, who's a computer science professor, about a platform called CodeKataBattle, which allows students to compete solving programming problems. Anne enjoys writing some python scripts to solve simple problems of everyday life and, while writing the scripts, she noticed that they required a deeper understanding of the solution. Since she knows that her students enjoy programming, as herself, Anne decides to create a tournament on CodeKataBattle, with the objective to make her students implement algorithms for the methods of problem solving taught in the course. At first, she was not sure that she could make something so complex but the fact that the platform could assign automatically the ranking needing only some test cases and that multiple programming languages were supported made Anne change her mind. Thus, Anne creates a nwe tournament by setting a name and a deadline. All the students subscribed to CodeKataBattle are notified by the platform and can now subscribe.

3. Educator adds a CK to a specific tournament

Jamie is a high school programming teacher; he is a Python and JavaScript expert but also knows Java. Jamie has been using CodeKataBattle for a while, since it helps his students to challenge themselves and provides an interctive programming exercise.

Every year a CodeKataBattle tournament is held by the professor and based on the final ranking the students get some extra points in the exam grade.

For this year, Jamie just found out an interesting problem that can be solved in various ways, each of them exploiting different proprieties of python. Thus, he decides to upload the problem in the tournament, not allowing students to join in groups since he wants to test each student individually.

Therefore Jamie creates five test cases, sets the deadline date and procedes to upload the codekata, the registration deadline, the submission deadline and an additional scoring criteria on cleanliness of the code.

4. Student join a group for a specific CK in a tournament

Donald is a carwash employe, even if things are going great his dream was to beacome a web developer. Donald already learned a lot JavaScript and is always in search of new challenges. Donald knows that in programming environment team working is a must. Since he always studied and exercised alone on CodeKataBattle, he now wants to practice some team work. So Donald opens CodeKataBattle and looks for some tournaments were battles in numerous groups are allowed. Once he found some battles about topics he's interested in, where groups are allowed, he create a group and waits for other students to join.

5. Student does a commit

Charlotte and Matt are working on the solution of a battle in their teacher tournament. They thought every test was passed, but then, they realised there might be a particular case in wich their algorithm fails. Thus, they fix the bug and, once they think the algorithm is done, the changes are committed and pushed. At this point the CodeKata-Battle platform performs the tests, the static analysis and the time ranking, giving the two students a new higher rank, confirming that there was a bug in their code.

6. Educator closes the tournament

Jeremiah is an Educator on CodeKataBattle, each month he creates a tournament about a simple problem, with a couple battles, one is usually a simple exercise, while the second is a more complex variation of the first one. As the month is coming to its end Jeremiah opens CodeKataBattle platform and checks how the partecipants are doing and decides to close the tournament. The platform then procedes to assign the final ranking and score of each student subscribed to the tournament and makes them available for everyone to consult.

2.1.2 Domain class diagrams

The domain class diagram for CodeKataBattle is presented below, covering all the elements within the system's operational environment and illustrating their interactions.

DIAGRAM HERE

Description of the elements in the class diagram HERE.

2.1.3 State diagrams

Introduction to the state diagrams and why we choose to represent those scenarios.

First diagram

DIAGRAM

Description of the diagram.

Second diagram

2 DIAGRAM

Description of the 2 diagram.

2.2 Product functions

Introduction to functions offered and specific description of each one.

2.2.1 First function

2.3 User characteristics

The platform accommodates interaction from two user categories: students and educators. The initial user category engages in Code Kata tournaments, while educators, on the other hand, serve as the organizers of these tournaments.

2.3.1 Student

The student, as an individual aiming to secure victory in a specific tournament, may need to either join an existing team or establish one with fellow students, considering that each tournament has a predetermined group size.

In particular, a student requires access to the following features within the system:

- Student login.
- Personalized user interface.
- Access to the list of available tournaments.
- The ability to join a desired tournament if it's open for registration.
- Capability to create and join groups.
- Invitation of other students to their group.
- Access to the GitHub repository link for a specific tournament.
- Access to both interim and final tournament rankings.

2.3.2 Educator

The educator's role encompasses the creation and management of tournaments, with the potential involvement of other educators designated by the tournament's owner, who is typically the initiating educator. All educators should possess the capability to create code Katas for the tournaments they manage and provide additional points for evaluation.

To facilitate these responsibilities, educators need access to the following features within the system:

- Educator login.
- Personalized user interface.

- Access to a list of tournaments under their management.
- The ability to create new tournaments with customized rules.
- Within a managed tournament:
 - The option to invite other educators to participate.
 - The capability to add new code Katas.
 - The authority to close the tournament.
 - The ability to assign extra points if specified in the tournament rules.
- Access to the GitHub repository for every student in the active tournament.
- Access to both interim and final tournament rankings.

2.4 Assumptions, dependencies and constraints

Specific requirements

3.1	External	inte	rface	requi	iremen	\mathbf{ts}

- 3.1.1 User interfaces
- 3.1.2 Hardware interfaces
- 3.1.3 Software interfaces
- 3.1.4 Communication interfaces
- 3.2 Functional Requirements
- 3.3 Performance requirements
- 3.4 Design constraints
- 3.4.1 Standards compliance
- 3.4.2 Hardware limitations
- 3.4.3 Any other constraint
- 3.5 Software system attributes
- 3.5.1 Reliability
- 3.5.2 Availability
- 3.5.3 Security
- 3.5.4 Maintainability
- 3.5.5 Portability

Chapter 4
Formal analysis with Alloy

Effort spent

The table below offers a concise overview of the hours invested by each group member, along with a brief description of their contributions. Dates where the same amount of time was dedicated by all members usually indicates collaborative efforts.

Date	Rossi	Sharoubim	Description
22-10-2023	1	1	Repository setup and file structure
28-10-2023	3	3	First chapter initial content
XX-XX-XXXX	-	-	
Total	4	4	-

References