

RASD

CodeKataBattles

Federica Laudizi, Antonio Marusic, Sara Massarelli

November 2023

Contents

1	Introduction	2
1.1	Purpose	2
1.1.1	Goals	2
2	Overall characteristics	3
2.1	Product functions	3
2.1.1	Visualisation of the badges	3
2.1.2	Vedere la rank	3
2.1.3	Vedere i tournament	3
2.1.4	Visualize participant	3
2.1.5	Push of the code	3
2.1.6	Signup and login same for everyone	3
2.1.7	Signup and login same for everyone	3
2.1.8	Create a tournament and Set parameters	3
2.1.9	Invite	3
2.1.10	Accept invite	3
2.1.11	Decline invite	3
2.1.12	Create repository	3
2.1.13	A ^{***} Send notification	3
2.1.14	A ^{***} Run tests	3
2.1.15	A ^{***} Assign score automatically	3
2.1.16	A ^{***} Consolidation stage: manually evaluate repo	3
2.1.17	A ^{***} Create a Kata Battle	3
2.1.18	A ^{***} Create Badge	4
3	Effort spent	4

Abstract

This Requirements Analysis and Specification Document (RASD) is going to tackle objectives, requirements (functional and non functional) for the project CodeKataBattles.

It will be presented the scope, the functionalities and the domain of such project and we will give a comprehensive overview of the interaction with users and external components and the performance expected from the system.

This document is going to be a reliable point of reference for developers since it will define clearly the functionalities of the system and it will give precise guidelines during the validation and verification process.

The profound goal of this document is to give a precise overview of the product to be in order to give the stakeholders a general understanding of usage scenarios and avoiding changes of directions during the design and implementation part of the system development.

look for
a way to
identify
spelling
mistakes

1 Introduction

1.1 Purpose

This system allows students to boost their learning by participating in code challenges called kata battles organised by teachers and independent tutors.

Such challenges will be part of tournaments so that it is possible for students to compete in multiple katas and seeing their position in a leaderbord of all participants. The system allows for the creator of the tournament to define a set of rules regarding the number of participants and the size of the groups of students and it gives the possibility to delegate the creation of the katas and the managing of the tournament also to other teachers.

The system offers a platform integrated with GitHub where teachers can publish the kata and define the test cases that the code provided by the students must pass and assess student's submissions. The students on the other hand, can submit their code and see immediately its performances. The system assigns a score to each submission based on functional and non functional aspects of the code so that students can compare their work with other participants.

Moreover, the system promotes global participation in kata battles, since it is possible to join battles organised by educators from all around the world. It is worth noticing that, not only certified teachers can create tournaments, but also independent tutors, allowing them to engage their students in productive competitions.

1.1.1 Goals

The system must satisfy this goals:

Actors definitions:

- Admin: admins are the teachers, educators or tutors that created the tournament or were accepted by the creator as collaborators. Admin users have the privilege for that tournament to manage the tournament settings and parameters and to publish new kata battles
- Participants: are the students that enroll in a tournament.

G1: Participants can participate in a tournament.

G2: Participants can submit their code through GitHub.

G3: Participants can view the results of their submission as a score from 0 to 100.

G4: Participants can see the scores of other participants and their position in the leaderboard, whether they are in the same tournament or not.

G5: Participants can see all tournaments currently running.

G6: Admins can invite collaborators to the tournament that will be considered as admins once they accept the invite.

G7: Users can request to become admins.

G8: Admins can accept user requests to become admins.

G9: Admins can accept user requests to participate in a tournament.

G10: Admins and collaborators can create kata battles.

G11: Admins can change the parameters for tournaments they created (e.g., make the tournament open, set the max number of participants, set the group size, set the duration of each kata battle).

G12: Admins can...

better explain the subdivision in groups, maybe we don't have only the concept of participant but we have the concept of group and participants are the components of the group

Better define what the parameters are.

to be continued

2 Overall characteristics

2.1 Product functions

Visualization of the system

2.1.1 Visualisation of the badges

2.1.2 Vedere la rank

2.1.3 Vedere i tournament

2.1.4 Visualize participant

Modification of the system

2.1.5 Push of the code

2.1.6 Signup and login same for everyone

2.1.7 Signup and login same for everyone

2.1.8 Create a tournament and Set parameters

2.1.9 Invite

colleague

group member

2.1.10 Accept invite

colleague

group member

2.1.11 Decline invite

2.1.12 Create repository

2.1.13 A^{***}Send notification

Send link

End of battle

Invite

2.1.14 A^{***}Run tests

2.1.15 A^{***}Assign score automatically

2.1.16 A^{***}Consolidation stage: manually evaluate repo

2.1.17 A^{***}Create a Kata Battle

The system allows admins to create a Kata Battle . The process involves asking a textual description of the problem to solve by the groups and a software project with build automation scripts whose structure depends on the programming language of choice for the kata. This project must contain a set of test cases, otherwise it won't be possible to perform the evaluation of the submissions. . After that the sytem prompts for some configurations parameters for the kata, such as minimum and maximum number of students per

define a kata battle

can katas allow for different programming languages?

group, a registration deadline, a final submission deadline and additional configurations for scoring

When the system receives in input the data required it performs some validity checks . Later, it will ask for confirmation and the kata will be created.

2.1.18 A^{oo}Create Badge

Write a formula with existing variables

3 Effort spent

Antonio Marusic

section 1: 2h

section 2: 1h

what are the
necessary
validity
checks?
what can't
be validated
and so
has to be
put in the
constraints?
so are not
Notifications
will be sent?
tournament