POLITECNICO
MILANO 1863

CodeKataBattle project by Russo Mario
and Picone Paolo

Requirement Analysis and Specification Document

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Picone Paolo and Russo Mario
Version:	1.0
Date:	20-November-2023
Download page:	https://github.com/piconepaolo/PiconeRusso
Copyright:	Copyright © 2023, Picone Paolo and Russo Mario – All rights reserved

Contents

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

List of Figures

List of Tables

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

1 Introduction

1.1 Purpose

The motivation behind the existence of the CodeKataBattle platform is to provide students with a dedicated platform for practicing their programming skills. The platform aims to create a competitive environment where teams of students can participate in programming tournaments and solve programming challenges.

By offering a platform specifically designed for programming practice, CodeKataBattle aims to provide students with a structured and engaging way to improve their coding abilities. The competitive nature of the platform adds an extra layer of motivation and excitement, encouraging students to push their limits and strive for excellence.

Overall, the motivations behind the existence of the CodeKataBattle platform are to create a dedicated space for programming practice, foster healthy competition among students, and provide a means for tracking and improving programming skills.

1.1.1 Goals

G1 Educator can create a tournament

G2 Educator can set up battles for a tournament

G3 Student can compete in a tournament

G4 Team of students can compete in a battle

G5 Educator can evaluate the performance of a student in a battle

G6 CKB notify the student about upcoming tournaments

G7 CKB notify the student about upcoming battles

1.2 Scope

The scope of CKB is to provide a platform for programming practice. The platform will allow students to participate in programming tournaments and solve programming challenges. The torunaments consist of a series of battles. In order to participate to a battle a student must be subscribed to the tournament in which the battle takes place. In each battle the students have to form teams by inviting other students to join. The platform will allow educators to create tournaments. For the single torunament, the educator is able to invite other educators to join the tounament as educator. The educator is able to create battles for the torunament by uploading the code kata composed by the description and software project. In addition, he should be able to upload a set of configuration for scoring. For each battle the educator is able to set restrictions such as the minimum and maximum number of team members, the registration deadline and the submission deadline. The educator is also able to set a time limit for the battle. CKB will assign the scoring for each team of the battle based on some automated tests regarding the functional aspects, timeliness, quality of the code and the set of configuration for scoring provided by the educator for the battle. In addition to this automated scoring the educator will also be able to assign a manual scoring for each team of the battle. In order to submit the solution for a battle, CKB provides a GitHub repository. The teams must fork the repository and work on their solution in the forked repository. For each push to the remote repository CKB will be informed through an API call made with GitHub Actions which teams must set up.

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

1.2.1 World phenomena

WP1 Student wants to compete in a tournament

WP2 Educator wants to create a tournament

WP3 Educator wants to manually evaluate the performance of a student in a battle

WP4 Educator wants to set up a battle for a tournament

1.2.2 Shared phenomena

Phenomena controlled by the world and observed by the machine:

SP1 Student push the solution of the battle to the forked repository

SP2 Educator sets deadlines for tournaments and battles

SP3 Educator sets the configuration for scoring

SP4 Educator uploads the code kata for a battle to CKB

SP5 Educator sets the restrictions for a battle

Phenomena controlled by the machine and observed by the world:

SP6 CKB notifies the student about upcoming tournaments and battles

SP7 CKB notifies about the students about the results of the battle and the tournament

SP8 CKB shows information about the scoring of the battle and the tournament

SP9 CKB provides a GitHub repository for each battle

1.3 Definitions, Acronyms, 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

This document is structured as follows:

- **Introduction**: it provides a general description of the product, its purpose and the goals that the project aims to achieve. It also contains the scope of the product, the phenomena that the product will interact with and the shared phenomena between the product and the world. Finally, it contains the definitions, acronyms and abbreviations used in the document.
- **Overall Description**: TODO
- **Specific Requirements**: TODO
- **Formal Analysis**: TODO
- **Effort Spent**: TODO
- **References**: it contains the list of the documents used to redact this document.

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

2 Overall Description

2.1 Product Perspective

2.1.1 Scenarios

1. **Educator creates a battle for a tournament**
   The educator who wants to create a battle in a tournament for which he is the creator, or for which he has the permission to create battles. The creation of a battle consists of in:

   - Set the deadline for registration of teams
   - Set the deadline for the submission of solutions
   - Set the minimum and maximum number of students per team
   - Upload the code kata to the CKB platform

   Once the battle is created, the students registered at that tournament are notified by CKB about the upcoming battle. The CKB platform will also create a new repository on Github for the battle, and will invite the students to fork it.

2. **Student invite other students to join the team**
   Roberto is a student who wants to participate in a battle. He has already registered in the tournament in which the battle has been created, and he has been notified about the upcoming battle. He wants to invite other students to join his team for the battle. He can do this by sending an invitation to the other students, who will receive a notification about the invitation. The other students can accept or decline the invitation. If they accept, they will be added to the team and they will be able to compete as a team in a battle. Even if the students invited have accepted the invitation the partecipation to the battle is not guaranteed since the educator could have set a maximum number of students per team.

3. **Student push the solution of the battle to the forked repository**
   The team A have been working on the solution for the battle for a while and it seems to be passing all the tests provided in the code kata. Since the time being is before the submission deadline the team push the solution to the forked repository. This will trigger the CKB platform to evaluate the solution and update the score of the team in the battle. In the case that the submission date is after the submission deadline the push will not trigger another evaluation of the solution by the CKB platform.

4. **Educator manually updates the score of a team**
   Giacomo is an educator who has created a battle for a tournament in which he has been invited by another collegue. Once the submission deadline is expired he wants to manually assess the score of a team to cover all the aspects that the CKB platform cannot evaluate. To do this he access through the CKB platform the battle page and he can see the list of teams that have submitted a solution. He can then manually update the score of a team by clicking on the ‘Update score’ button. This will open a form where he can modify the current score of the team previously evaluated by the platform. Once he is satisfied with the score he can click on the ‘Save’ button to save the new score for the team. The new score will be visible to the team and to the other students subscribed to the tournament.

5. **CKB notifies students about the result of the tournament**
   When an Educator closes a tournament, the CKB platform will notify all the students subscribed to the tournament about the result of the tournament. The notification will contain the list of the teams that have participated in the tournament and their final score.

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

2.1.2 Domain class Diagram

In Figure 1 is shown the domain class diagram of the CKB platform. This diagram shows the main entities of the system and their relationships.

- **User**: is the main entity of the system. It represents a user of the CKB platform. It can be either a student or an educator.
- **Student**: is a user of the CKB platform. It can participate in tournaments and battles. It can also create teams and invite other students to join them.
- **Educator**: is a user of the CKB platform. It can create and manage tournaments and battles. He can also invite other educators to join a tournament. He also uploads the code kata to the CKB platform for a battle.
- **Tournament**: is a competition between teams of students. It is created by an educator and it can be joined by students. It can contain multiple battles.
- **Battle**: is a competition between teams of students. It is created by an educator and it can be joined by students. It is part of a tournament and it can contain multiple teams.
- **Team**: is a group of students that compete together in a battle. It is created by a student and it can be joined by other students.
- **Invitation**: is a request sent by a student to another student to join a team. It can be accepted or declined by the receiver.
- **Code Kata**: is the description of the problem that the student have to solve in a battle. It contains the description of the software project to be implemented, the test cases that the solution must pass and the build automation scripts.
- **Score**: is the score of a team in a battle. It is calculated by the CKB platform based on the evaluation of the submission of the team. It also adds up the manual score given by the educator. The score assigned to the team, for each battle, is also the score of the students in the tournament.
- **Ranking**: is the ranking of the students in a tournament. It is calculated by the CKB platform based on the score of the students in the battles of the tournament in which they have participated.
- **Repository**: is the repository on Github that contains the code kata and the build automation scripts for a battle. It is created by the CKB platform when a battle is created. It is forked by the teams that want to compete in the battle.
- **Forked Repository**: is the repository forked by a team of students to compete in a battle. It contains the code kata and the build automation scripts for the battle. Each push to the forked repository will trigger the CKB platform to evaluate the submission of the team and update the score of the team.
- **Submission**: is the code that a team of students have implemented for a battle or a subsequent version of a previous submission. It is pushed to the forked repository of the battle by a team member. The CKB platform will evaluate the submission and update the score of the team.
- **Evalutation**: is the evaluation of a submission. It is done by the CKB platform based on the code kata provided by the educator. It is done automatically by the CKB platform when a submission is pushed to the forked repository of the battle. It is also done manually by the educator when he updates the score of a team.

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

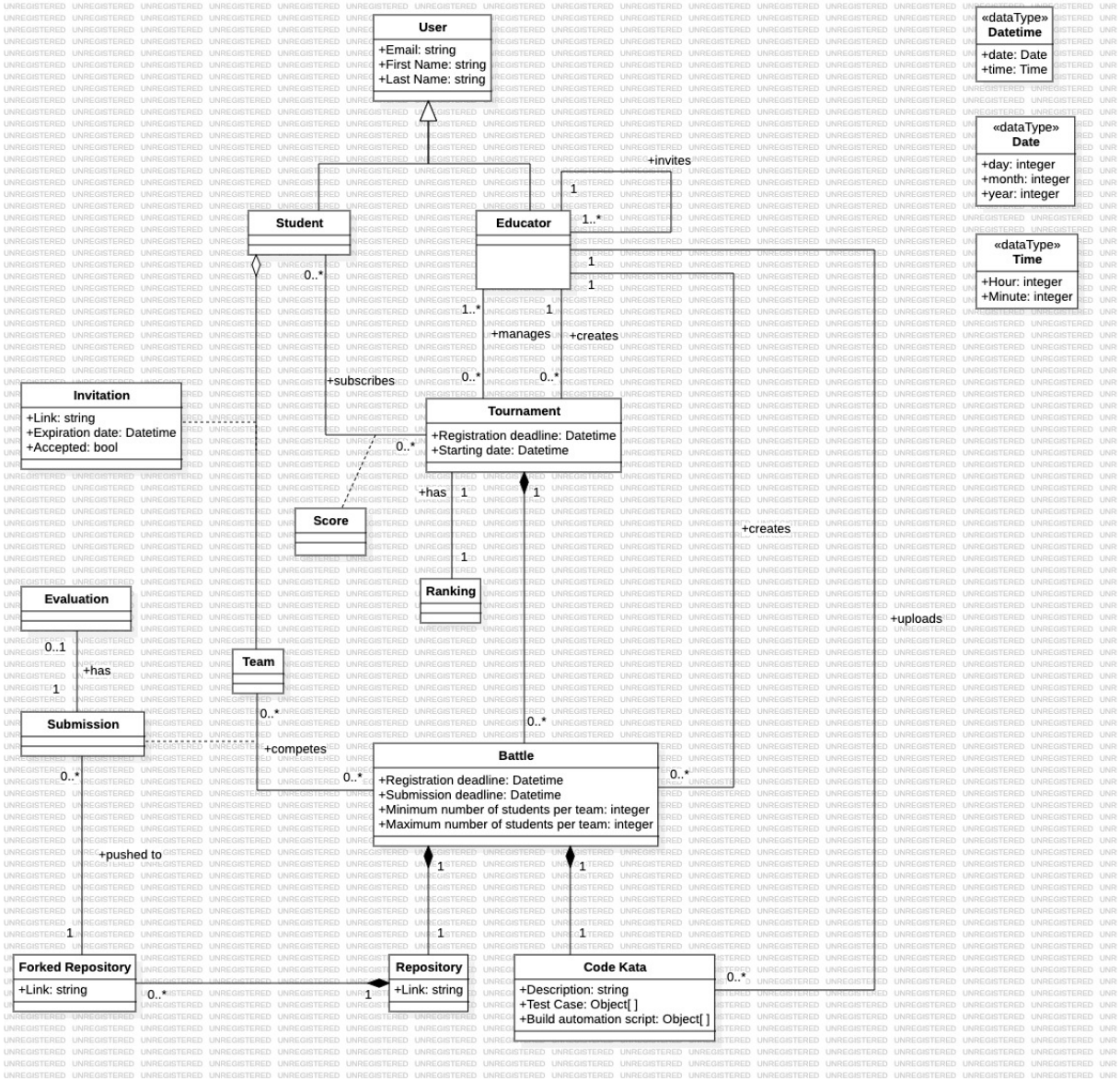

Figure 1: Domain class diagram

2.1.3 Statecharts

Statechart of a Tournament

The statechart in Figure 2 shows the possible states of a tournament. A tounament can be in two principal states: “In Progress” or “Closed”. A tournament is in the “In Progress” state when it is created by an educator. When the educator closes the tournament, it will be in the “Closed” state.

Figure 2: Statechart of a Tournament

Statechart of a Battle

The statechart in Figure 3 shows the possible states of a battle. When the educator creates a battle it

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

will be in initial “Created” state. When the date of the registration deadline is expired, the battle will be in “Registration closed” state. When registration is closed, the platform sends the link of the repository to the students that have registered to the battle. After the link is sent the battle will be in “Accepting submissions” state where the teams can submit their solutions. When the date of the submission deadline is expired, the battle will be in “Consolidation” state. In this state the educator can manually update the score of a team and any additional submission from a team will not be considered. When the educator closes the tournament and finalizes the evaluations, it will be in the “Closed” state.

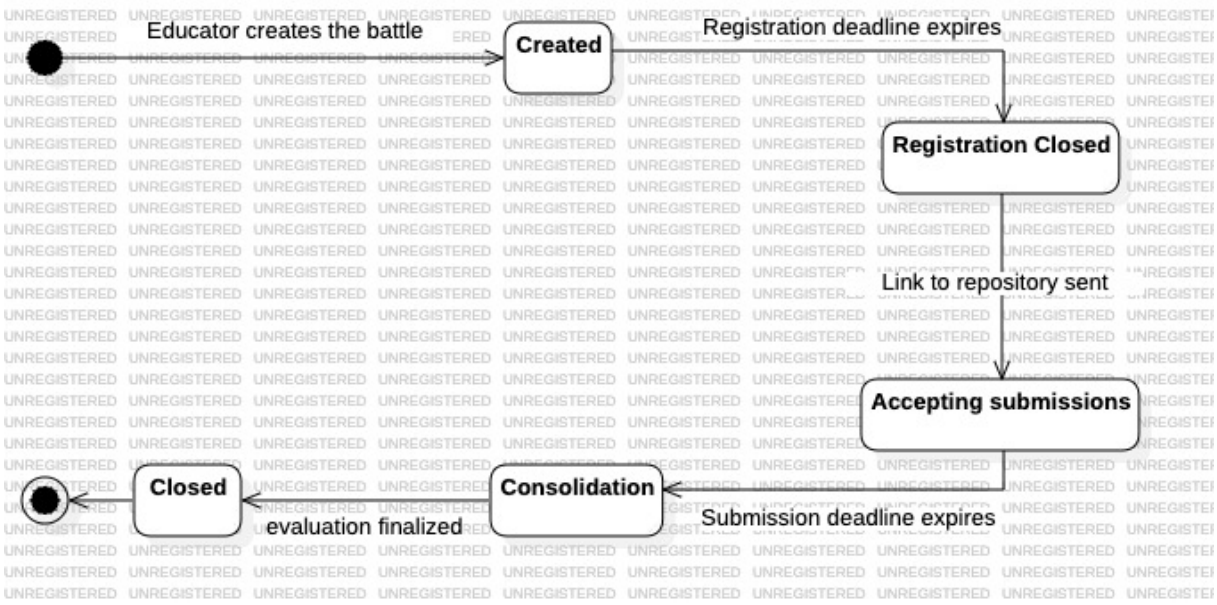

Figure 3: Statechart of a Battle

2.2 Product Functions

2.2.1 User registration and login

The CKB platform allows the users to register and login to the platform. The registration is required to access the platform and to use its functionalities. The registration process requires the user to provide a valid email address and a password. The email address will be used to identify the user in the platform i.e.User or Educator. The password will be used to authenticate the user when he wants to login to the platform. The CKB platform will send a confirmation email to the user to verify the email address. The user will be able to login to the platform only after the email address has been verified.

2.2.2 Creation of a tournament

The CKB platform allows the educators to create tournaments. The Educator will be prompted to provide the name of the tournament and the deadline for the registration. When the tournament is created, the CKB platform will send a notification to all the students registered to the platform. The Educator, that has done the creation, will be able to invite other educators to join the tournament.

2.2.3 Creation of a battle

The CKB platform allows the educators to create battles. The Educator will be prompted to provide the name of the battle, the deadline for the registration, the deadline for the submission of the solutions, the minimum and maximum number of students per team and the code kata. When the battle is created,

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

the CKB platform will send a notification to all the students subscribed to the tournament. The CKB platform will also create a new repository on Github for the battle, and will invite the students to fork it.

2.2.4 Creation and joining of a team

The students in order to compete in a battle must form or join a team. The CKB platform allows the students to create teams by inviting other students to join their team. The student will be prompted to provide the name of the team, as well as the list of the emails of the students that he wants to invite. The CKB platform will send a notification to the invited students. The invited students will be able to accept or decline the invitation. If they accept, they will be added to the team as long as the minimum and maximum number of students per team (set by the educator) is respected.

2.2.5 Submission and evaluation of a solution

The students of a team that are competing for a battle can submit their solution to the CKB platform in order to be evaluated. The submission consists of pushing the code that the team has implemented for the battle to the forked repository of the battle previously created by the CKB platform and forked by the team. The CKB platform will evaluate the submission using the code kata provided by the educator and will update the score of the team.

2.2.6 Manual update of the score

The CKB platform allows the educators to manually update the score of a team. After accessing the battle page on the CKB platform, educators can manually adjust a team's score as long as the battle is in the *Consolidation stage*. This feature proves beneficial when evaluating aspects beyond the scope of automatic assessment already done by the platform. On the battle page, educators can view the list of teams that have submitted solutions. By clicking the Update Score' button, a form is opened, allowing the educator to modify the team's current score as assessed by the platform. Once satisfied with the adjustments, clicking the Save' button finalizes the new score.

2.2.7 Notification of the result of a tournament

Once an educator closes a tournament, the CKB platform will notify all the students subscribed to the tournament about the result of the tournament. The notification will contain the list of the teams that have participated in the tournament and their final score.

2.2.8 Creation of the repository to sumbit the solution

The CKB platform will create a new repository on Github for the battle, and will invite the students to fork it by sending the link of the repository to the students. The repository will contain the code kata provided by the educator and the build automation scripts. The students will be able to push their solution once they have forked the repository and set up the actions to trigger the evaluation of the solution by the CKB platform.

2.3 User Characteristics

The CKB platform is designed to be used by two types of users: *Educators* and *Students*.

2.3.1 Educator

The Educator is the user that manages tournaments and battles. He is able to invite other educators to join a tournament in order to grant them the permission to create battles for that tournament. The educator is

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

the user that creates and closes the tournaments and battles (for which he has the permissions to do so in the case he was invited). He can set restrictions for the battles such as the deadline for the registration of teams, the deadline for the submission of solutions, the minimum and maximum number of students per team. He provides the code kata for the battle. He can also manually update the score of a team. The educator can be looked at as the administrator for a tournament and battles for that tournament. Nevertheless he cannot compete like a student in a tournament or battle.

2.3.2 Student

The Student is the user that participates in tournaments and battles. He can team up with other students to compete in a battle by forming teams. He can submit the solution of a battle with his team. He can also compete in multiple battles and multiple tournaments with multiple teams at the same time. The student can be looked at as the competitor in a tournament since the ranking of the tournament is per student. Nevertheless he cannot manage tournaments and battles like an educator.

2.4 Assumptions, Dependencies and Constraints

2.4.1 Regulatory policies

The CKB platform will store the informations provided by the user, such as the name, surname and email address and informations on forked repositories. This information will not be used for commercial purposes and they will not be shared with third parties.

2.4.2 Domain assumptions

This paragraph contains the assumptions that the CKB platform makes about the domain in which it operates. These assumptions are out of the scope of the CKB platform and they are not verified by the platform.

D1: User must have an internet connection

D2: The Student must have a Github account

D3: The Educator must have a Github account

D4: User must have a valid educational email address in order to be verified by the CKB platform

D5: Teams must setup the Github actions to trigger the evaluation of the solution by the CKB platform

D6: Student needs to accept the invitation to join a team

D7: Student must be registered to CKB to receive notifications about tournaments

D8: Student must be registered to the tournament to receive notifications about upcoming battles

D9: Educator must upload a solvable code kata to the CKB platform for a battle

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

3 Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces

Figure 4: Student Dashboard

3.1.2 Hardware Interfaces

Since there is no particular hardware requirement for the system, the system will be able to run on any hardware that can run a web browser and has an internet connection.

3.1.3 Software Interfaces

The system in order to perform its functions does not require any interaction with other software systems.

3.1.4 Communication Interfaces

The system provides the functionality of automatic evaluation. This function requires the system to communicate with an external client that triggers this automatic actions. In particular, the system needs to expose an API that the GitHub Action set up by the students will call when a new commit is pushed.

3.2 Functional Requirements

This section specifies all the requirements that the system must satisfy in order to be considered complete and functional.

R1: The system shall allow the user to register to the system

R2: The system shall allow the user to login to the system

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

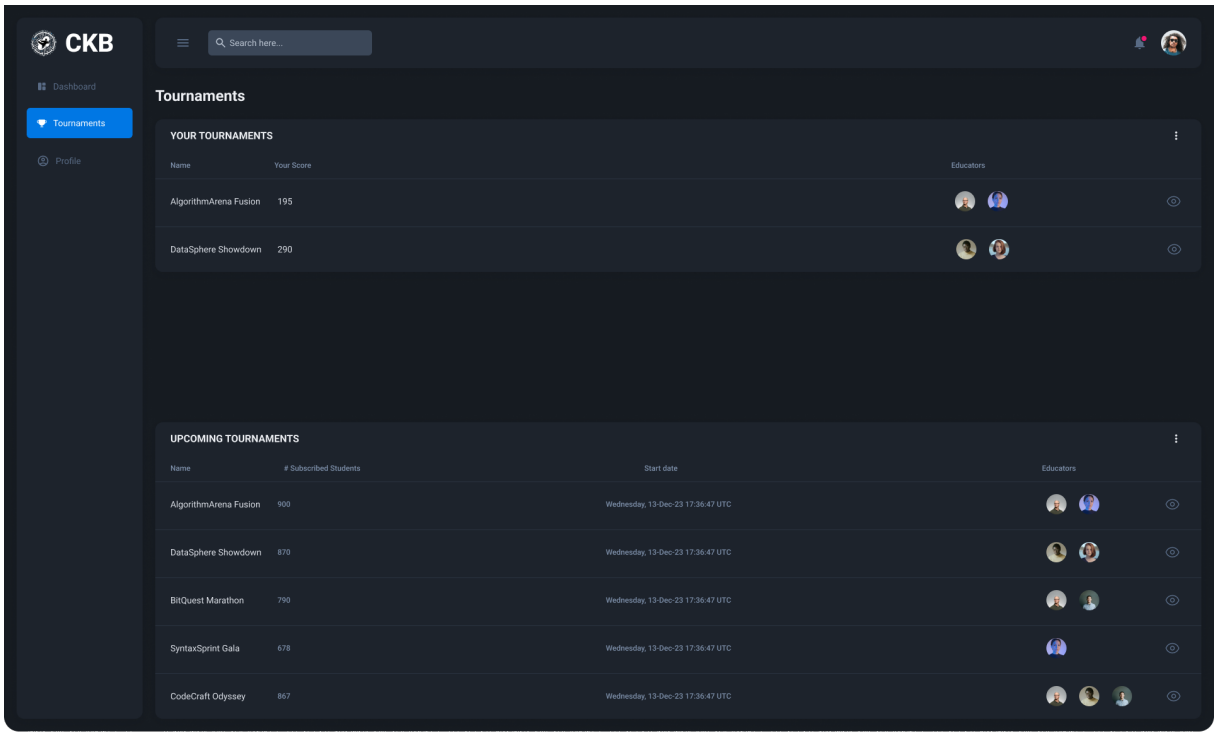

Figure 5: Student Dashboard Tournaments Tab

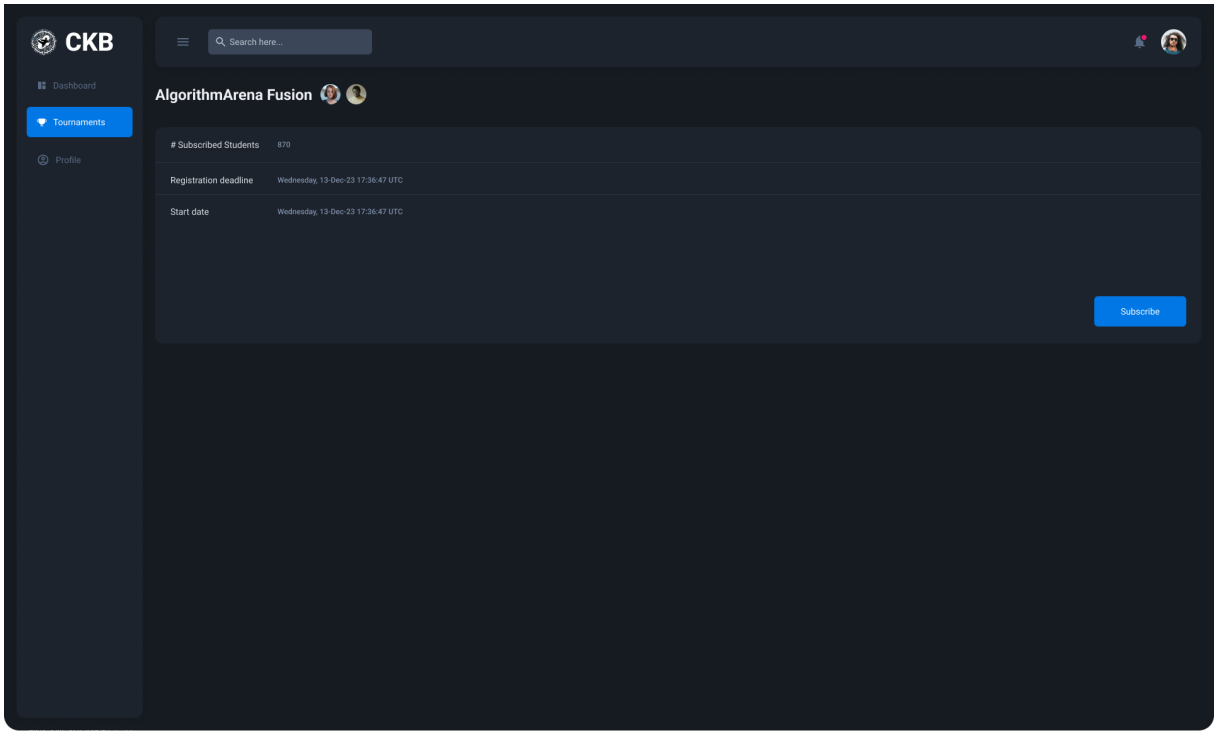

Figure 6: Student Dashboard Tournament OverView

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

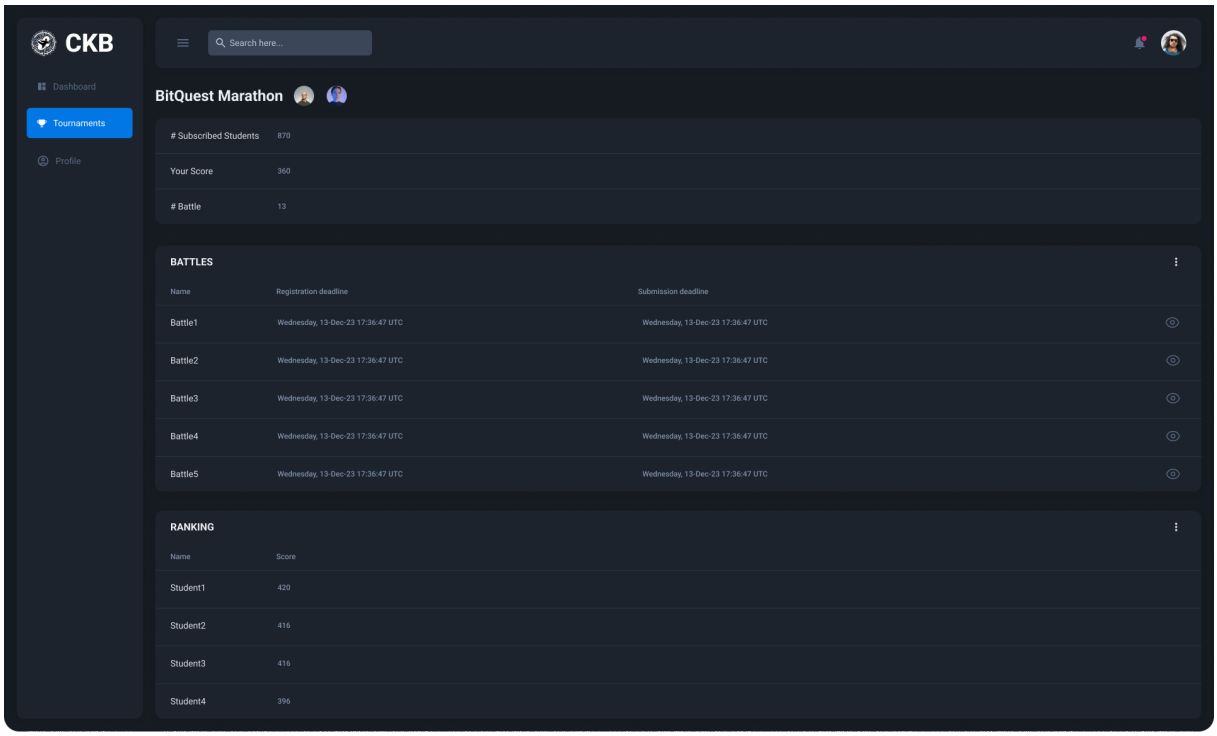

Figure 7: Student Dashboard Tournament Ongoing

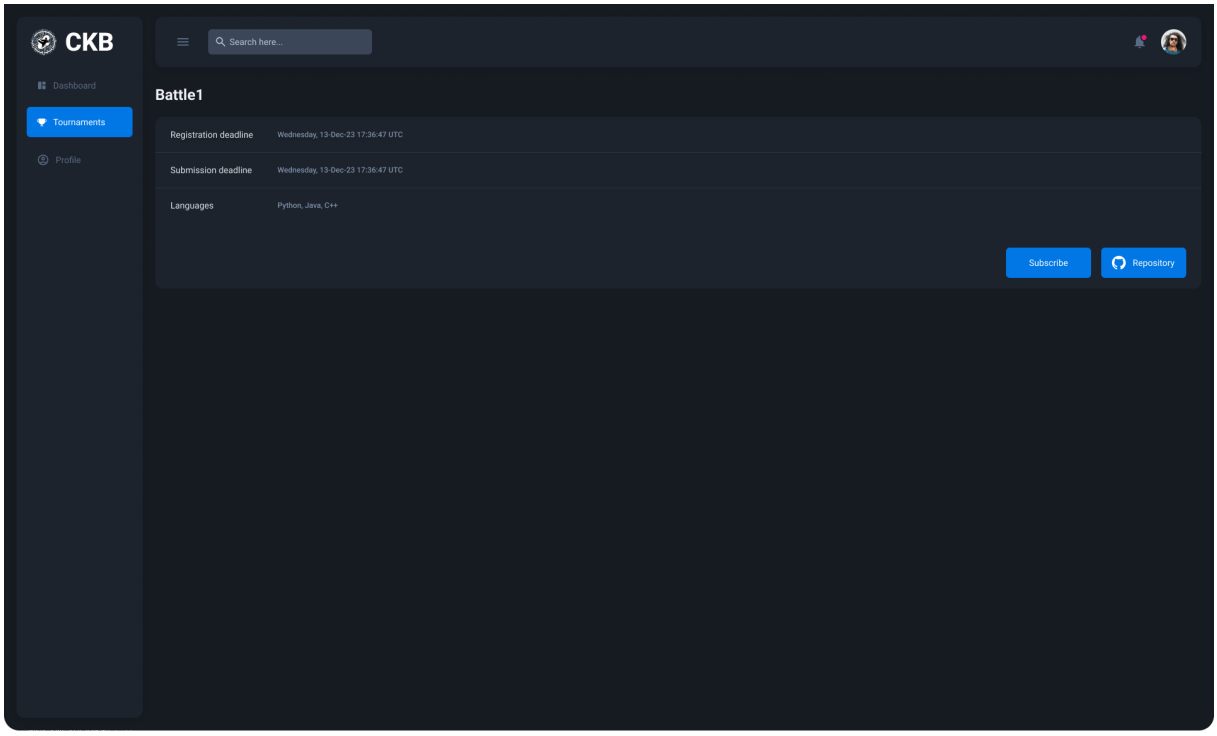

Figure 8: Student Dashboard Battle Overview

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

R3: The system shall allow the educator to create a new tournament

R4: The system should allow the educator to create a new battle for a tournament

R5: The system shall allow the student to subscribe to a tournament

R6: The system shall allow the student to subscribe to a battle

R7: The system shall create a forkable repository for each battle

R8: The system shall allow the student to create a team for a battle

R9: The system should allow the student to invite other students to join a team

R10: The system should notify students subscribed to the platform when a new tournament is created

R11: The system should notify students subscribed to a tournament when a new battle is created

R12: The system should notify students subscribed to a tournament when the system publishes the ranking of the tournament

R13: The system shall guarantee that the restrictions for a battle set by the educator are respected

R14: The system shall allow the educator to set the restrictions for a battle

R15: The system shall allow the educator that created a tournament to invite other educators to the tournament

R16: The system shall allow the educator to close a tournament

R17: The system should allow the educator to manually update the score of a team for a battle

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

4 Formal Analysis Using Alloy

Organize this section according to the rules defined in the project description.

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

5 Effort Spent

Provide here information about how much effort each group member spent in working at this document. We would appreciate details here.

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved

References

Copyright © 2023, Russo Mario and Picone Paolo – All rights reserved