**2. Overall Description**

## **2.1 Product Perspective**

### **2.1.1 Scenarios**

1. **Tournament Creation by an Educator**

Luca Proserpio is an educator who works for a famous Italian university and is subscribed to the CKB Platform.

Luca wants to create a new tournament on the platform so that students can participate in it to improve their programming skills in Java.

Luca logs in, and the HOMEPAGE will be shown.

Luca among the various options on the homepage decides to click on the "Create New Tournament" button.   
The platform shows the CREATE\_TOURNAMENT page which contains:

* *Name field*: to set the name of the tournament
* *Collaborators list*: with all the educators in the platform
* *Deadline field*: to define the deadline date to be able to register for the tournament
* *Create Tournament button*: to proceed with the creation of the tournament

Luca enters the name "Welcome Tournament School Year 2024" in the Name field and sets the date 30/09/2023 in the Deadline field.

Next, he clicks on the list of educators selecting educators Gianpaolo Mariani, Mario Rossi, and Rosa Ballabio as he wants to allow them also to be able to create new battles for the tournament he is creating.

Accordingly, he clicks on the Create Tournament button and the platform shows a message of successful tournament creation.  
Luca will be shown the DETAIL\_TOURNAMENT page of the newly created "Welcome Tournament School Year 2024" for which, at the moment, as an empty list of available battles.

All students enrolled in the CKB Platform will be notified about the new creation of the tournament created by educator Luca.

1. **Battle Creation by and Educator for an Existing Tournament**

Gianpaolo Mariani is an educator registered with the CKB Platform and wants to create a battle for the "Welcome Tournament School Year 2024" tournament.

After logging in, the HOMEPAGE shows the list of tournaments he has created or to which he has been added by other contributors. Gianpaolo clicks on the "Welcome Tournament School Year 2024" line found in the list of tournaments mentioned earlier.

After clicking, the platform shows the DETAIL\_TOURNAMENT page of the selected tournament.

On the DETAIL\_TOURNAMENT page, Gianpaolo clicks on the "Create New Battle" button and the CREATE\_BATTLE page is shown to him.  
The CREATE\_BATTLE page contains:

* Kata Code Upload Section
  + *Battle Description field*
  + *Form to upload test cases*
  + *Form to upload build automation scripts*
* *Field group policy:* to set minimum and maximum number of students per group
* *Field registration deadline:* to define maximum date for which students can register for the battle
* *Field final submission deadline:* to define maximum date for which students can submit code to be evaluated
* Scoring Configuration section
  + *Form functional aspects*
  + *Form timeliness*
  + *Form quality level*
  + *Form optional manual evaluation*
* *Create Battle button*

Gianpaolo fills in all the fields by entering the description of the battle, the registration deadline as 6/11/2023, the final submission deadline as 23/12/2023, the minimum and maximum number of students as 3, and does not edit any of the default entries in the Scoring Configuration section as he plans to edit it later before the battle is actually started.

After that, he clicks on the Create Battle button and the platform redirect Gianpaolo to the DETAIL\_TOURNAMENT page containing the new created battle in the list of available battles.

All students who had signed up for the tournament will receive a notification of the newly added battle created by educator Gianpaolo.

1. **Student joins to an existing Tournament by receiving a notification**

Marco is an University student and thus is registered on the CKB Platform (**ASSUMPTION**). He wants to sign up for a tournament to improve his programming skills in Java as he notices that he has difficulty writing code using Object Oriented Programming. In the afternoon Marco receives a notification of a new tournament creation called "The Basics of Object-Oriented Programming in Java", he decides to take the opportunity and sign up.  
Therefore, he accesses the platform by logging in and the platform redirect him to the HOMEPAGE. Marco clicks on the record "The Basics of Object-Oriented Programming in Java" listed in the list of available Tournaments for which he is not subscribed.  
The platform shows Marco the SUBSCRIPTION\_TO\_A\_TOURNAMENT page.  
The SUBSCRIPTION\_TO\_A\_TOURNAMENT page shows information about the name and the description of the selected tournament.   
Marco clicks on the Subscribe button, and the platform enrolls Marco in the tournament showing a confirmation registration message.

1. **Studenti creano un team per una battaglia di un torneo**Marco, Stefano e Carlo sono degli studenti iscritti al torneo “The Basics of Object-Oriented Programming in Java”.  
   Vogliono partecipare a una delle battaglie disponibili del torneo creando un team.  
   In particolare, Marco effettua il login e nella HOMEPAGE clicca sul torneo dall’elenco dei tornei a cui è iscritto.  
   A Marco viene mostrata la pagina DETAIL\_TOURNAMENT e al suo interno clicca sulla prima battaglia disponibile.  
   Gli viene mostrata la pagina DETAIL\_BATTLE e clicca sul bottone “Partecipa creando un team”.  
   Marco conosce già Stefano e decide di invitarlo a partecipare al suo team.  
   Carlo, invece, effettua le stesse operazioni di Marco solo che invece di cliccare sulla voce “Partecipa creando un team” clicca sulla voce “Partecipa unendoti a un team giá esistente”.  
   Il sistema dopo aver analizzato tutti i team già esistenti decide di aggiungere Carlo allo stesso team di Marco e Stefano.
2. **Creazione della GitHub repository quando la registration deadline di una battaglia termina**Gianpaolo Mariano è un educatore della piattaforma e aveva creato una battaglia per il torneo “Welcome Tournament School Year 2024” con deadline il 6/11/2023. Il sistema alla scadenza della deadline crea la repository GitHub contenente il code kata specificato da Gianpaolo e invia il link della repository appena creata a tutti gli studenti che si sono iscritti in teams prima della scadenza della deadline.
3. **Calcolo del battle scrore di un team**Marco è uno studente e sta partecipando in una battle con il team composto da Stefano e Carlo.  
   Marco modifica il codice del progetto e effettua una commit e push sulla repository GitHub forked del progetto.   
   La piattaforma preleva il codice, lo analizza e runs the tests predisposti (test cases) per la battaglia in cui il team sta partecipando.  
   La piattaforma aggiorna il punteggio del team considerando:
   1. Il numero di test cases superati con successo
   2. Il numero di giorni che sono passati dalla fine della registration deadline
   3. Il livelo di quality delle sources estratto dallo static analysis tools

Di conseguenza, dopo l’elaborazione Marco Stefano e Carlo visualizzano il nuovo rank assegnatoli automaticamente dalla piattaforma.