

## Università degli Studi di Ferrara Dipartimento di Ingegneria

# Corso di Laurea Triennale in Ingegneria Elettronica e Informatica

Documentazione Ingegneria del Software

## Progetto: Gestione Attività Sportive

Studente: Michele Vaccari Matricola 121955

Docente:

Prof. Fabrizio Luglio

## INDICE

```
SPECIFICHE PROGETTO
        Caratteristiche del sistema
        Utenti del sistema
2 MODELLO DEI CASI D'USO
        Attore amministratore
               Caso d'uso: Logout
        2.1.1
        2.1.2
               Caso d'uso: Inserisci arbitro
                                                 4
               Caso d'uso: Modifica arbitro
        2.1.3
               Caso d'uso: Visualizza arbitri
                                                   6
        2.1.4
               Caso d'uso: Visualizza utenti
        2.1.5
               Caso d'uso: Visualizza gestori squadra
                                                            8
        2.1.6
               Caso d'uso: Inserisci gestore squadra
        2.1.7
                                                          9
        2.1.8
               Caso d'uso: Modifica gestore squadra
                                                          10
               Caso d'uso: Elimina utente
        2.1.9
        2.1.10 Caso d'uso: Inserisci torneo eliminazione
                                                             12
        2.1.11 Caso d'uso: Visualizza tornei
        2.1.12 Caso d'uso: Inserisci torneo italiana
                                                        14
        2.1.13 Caso d'uso: Visualizza storico referti
                                                         15
        2.1.14 Caso d'uso: Assegna arbitro a partita
   2.2 Attore gestore della squadra
               Caso d'uso: Compila dati squadra
        2.2.1
                                                      19
        2.2.2
               Caso d'uso: Visualizza squadra
                                                   20
        2.2.3
               Caso d'uso: Modifica dati squadra
                                                      21
               Caso d'uso: Inserisci giocatore
        2.2.4
                                                  22
               Caso d'uso: Modifica giocatore
        2.2.5
                                                   24
               Caso d'uso: Visualizza giocatori
        2.2.6
                                                    25
               Caso d'uso: Elimina giocatore
        2.2.7
               Caso d'uso: Compila formazione
        2.2.8
                                                     26
               Caso d'uso: Logout
        Attore arbitro
               Caso d'uso: Compila referto
        2.3.1
        2.3.2
               Caso d'uso: Visualizza lista dei referti
                                                          30
               Caso d'uso: Logout
        2.3.3
                                       31
   2.4 Attore utente pubblico
                                    35
               Caso d'uso: Visualizza pagine statiche
                                                          35
               Caso d'uso: Effettua login
        2.4.2
   MODELLO DEL DOMINIO
        Diagramma delle classi
                                    38
   MODELLO DI DESIGN
                            39
```

4.1	Diagramma delle classi		
	4.1.1	Classi Boundary	39
	4.1.2	Classi Control	41
	4.1.3	Classi Entity 4	4
4.2	Diagr	ammi di sequenza	46

## ELENCO DELLE FIGURE

Figura 1	Casi d'uso dell'attore amministratore 3		
Figura 2	Diagramma delle attività della creazione di un torneo		
	all'italiana 18		
Figura 3	Casi d'uso dell'attore gestore della squadra 19		
Figura 4	Diagramma delle attività di modifica della squadra 22		
Figura 5	Casi d'uso dell'attore arbitro 29		
Figura 6	Diagramma delle attività della compilazione di un		
т.	referto 33		
Figura 7	Diagramma delle attività del logout di un utente 34		
Figura 8	Casi d'uso dell'attore utente pubblico 35		
Figura 9	Diagramma delle attività del login di un utente pub-		
	blico 37		
Figura 10	Diagramma delle classi di dominio 38		
Figura 11	Classi della sotto-cartella databaseservice 41		
Figura 12	Classi della sotto-cartella errorservice 42		
Figura 13	Classi della sotto-cartella logservice 42		
Figura 14	Classi della sotto-cartella sessionservice 42		
Figura 15	Classi del package bflows 43		
Figura 16	Classi service del package blogics 44		
Figura 17	Classi del package blogics 45		
Figura 18	Diagramma di sequenza della creazione della forma-		
	zione di una squadra 46		

# SPECIFICHE PROGETTO

#### 1.1 CARATTERISTICHE DEL SISTEMA

Il sito *SportFerrara* è una piattaforma per la gestione delle attività sportive attraverso una comunità di utenti.

L'applicazione web ha le seguenti caratteristiche:

- 1. Possibilità di creare gli utenti per i due ruoli principali: arbitri e gestori di squadra;
- 2. Possibilità di creare i diversi tornei, di disegnarne lo schema. Ogni torneo ha: nome, descrizione e tipo (all'italiana o ad eliminazione).
- 3. Un torneo è costituito da diversi gironi o fasi, in base alla tipologia di torneo. Un girone ha un nome (ad esempio: girone 1) ed è costituito da un insieme di gare. Una fase ha un nome (ad esempio: semifinale 1) ed è costituita da una sola gara;
- 4. Una gara è descritta da: data e ora, luogo, nomi delle squadre che si sfidano, uno o più arbitri;
- 5. Per ogni torneo è associata una classifica generale che viene aggiornata sulla base dei referti di gara compilati dagli arbitri;
- 6. Possibilità per il gestore di una squadra di creare una squadra con una rosa di massimo 36 giocatori. Ogni squadra ha un nome e può avere uno sponsor. Per ogni giocatore il gestore può specificare: nome, cognome, luogo e data di nascita, numero di maglia e foto;
- 7. Per ogni gara cui la squadra è assegnata, il gestore della squadra deve fornire la formazione della squadra composta da nome e cognome dei giocatori e rispettivi ruoli;
- 8. A gara terminata l'arbitro dovrà compilare un referto in cui annoterà: l'orario effettivo di inizio e di fine della gara, risultato finale, numero di reti con i rispettivi giocatori che li hanno realizzati, i giocatori espulsi per ogni squadra e i giocatori ammoniti per ogni squadra;
- 9. Possibilità di selezionare il torneo desiderato e visualizzare:
  - La pagina relativa ad una gara con: nomi delle squadre, formazioni e referti;
  - La pagina relativa ad una squadra con la rosa dei giocatori e il calendario delle partite;
  - La pagina relativa ad un giocatore con le statistiche di gioco (punti realizzati, espulsioni e ammonizioni) per quel torneo.

#### 1.2 UTENTI DEL SISTEMA

Il sistema prevede che le categorie di utenti sia così rappresentata:

AMMINISTRATORI Possono effettuare i punti dal 1 al 5 compresi.

GESTORI DI SQUADRA Possono effettuare i punti dal 6 al 7 compresi.

**Arbitri** Possono effettuare solamente il punto 8.

UTENTI PUBBLICI Possono effettuare solamente il punto 9.

# 2 | MODELLO DEI CASI D'USO

Si esegue l'analisi dei casi d'uso nel dominio degli attori che interessano i casi d'uso medesimi suddividendo lo schema completo in sottoschemi, uno per ogni attore, al fine di mantenere una maggior leggibilità del progetto. Si documenta ogni caso d'uso al fine di descrivere nel dettaglio il comportamento del sistema senza riferirsi ad una particolare implementazione.

#### 2.1 ATTORE AMMINISTRATORE

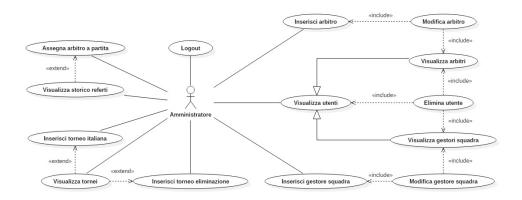


Figura 1: Casi d'uso dell'attore amministratore

## 2.1.1 Caso d'uso: Logout

## Descrizione

Questa funzionalità permette all'amministratore di uscire dalla sua pagina e di iniziare a navigare come utente pubblico. Il pulsante di Logout è presente in ogni pagina dell'applicazione web, in alto a destra.

## Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo.

#### Pre-condizioni

È richiesto l'accesso al sistema tramite la funzionalità di login.

#### Post-condizioni

Termina la sessione di accesso al sistema in qualità di amministratore.

## Flusso principale

- 1. L'amministratore seleziona nella pagina in cui si trova il pulsante di Logout;
- 2. Il sistema scollega l'amministratore dalle pagine a lui riservate;
- 3. Il sistema redireziona l'amministratore nella schermata di login e da quel momento può navigare come utente pubblico.

## Flussi alternativi

Nel caso di operazione non riuscita si notifica all'amministratore il tipo di errore che si è verificato.

## Diagramma delle attività

Il diagramma delle attività per il caso d'uso "Logout" è illustrato nella figura 7 a pagina 34.

#### 2.1.2 Caso d'uso: Inserisci arbitro

## Descrizione

Questa funzionalità permette all'amministratore di creare un nuovo utente di tipo arbitro.

#### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore interagisce direttamente con il sistema compilando i campi richiesti.

Arbitro, partecipazione non attiva in quanto si procede solo alla creazione di tale attore.

#### Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter inserire un arbitro.

## Post-condizioni

Viene aggiornato lo stato sul database con l'inserimento di un nuovo arbitro. L'arbitro è associato all'amministratore.

## Flusso principale

- 1. L'amministratore seleziona la voce "Nuovo Arbitro" nella propria home page oppure seleziona la voce "Inserisci Nuovo Arbitro"nella pagina di visualizzazione degli utenti;
- 2. L'amministratore inserisce per l'arbitro i seguenti dati: nome, cognome, email, password, indirizzo, telefono, data di nascita, nazionalità, foto e carriera;
- 3. L'amministratore invia la richiesta di creazione del nuovo utente al
- 4. In caso di email già presente nel database viene notificato con un messaggio l'indisponibilità della email inserita invitando l'amministratore ad inserire una nuova mail;
- 5. Se tutti i dati sono stati inseriti correttamente, il sistema inserisce l'arbitro nel database;
- 6. Il sistema visualizza i dati dell'arbitro appena inserito dando la possibilità all'amministratore di modificare i dati.

#### Flussi alternativi

Nel caso di inserimento errato dei dati da parte dell'amministratore o di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

## 2.1.3 Caso d'uso: Modifica arbitro

#### Descrizione

Questa funzionalità permette all'amministratore di modificare i dati di un arbitro presente nel database.

## Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore interagisce direttamente con il sistema modificando alcuni dei campi richiesti.

Arbitro, partecipazione non attiva in quanto si procede solo alla modifica di tale attore.

## Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter modificare un arbitro.

#### Post-condizioni

Viene aggiornato lo stato sul database con la modifica di un arbitro.

## Flusso principale

- 1. L'amministratore seleziona la voce "Utenti" nella pagina in cui si trova;
- 2. L'amministratore seleziona la voce "Arbitro" per visualizzare gli utenti di tipo arbitro;
- 3. L'amministratore seleziona la voce "Modifica", per l'arbitro che desidera modificare, nella lista degli arbitri;
- 4. L'amministratore può modificare per l'arbitro i seguenti dati: nome, cognome, email, password, indirizzo, telefono, data di nascita, nazionalità, foto e carriera;
- 5. L'amministratore invia la richiesta di modifica dell'arbitro al sistema;
- 6. In caso di modifica dell'email con l'inserimento di una email già presente nel database, viene notificato, con un messaggio, l'indisponibilità della email inserita invitando l'amministratore ad inserire una nuova mail;
- 7. Se tutti i dati sono stati inseriti correttamente, il sistema modifica l'arbitro nel database;
- 8. Il sistema visualizza i dati dell'arbitro appena modificato dando la possibilità all'amministratore di effettuare ulteriori modifiche.

## Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

## Punti di inclusione

Si può accedere a questo caso d'uso quando si effettua l'inserimento di un nuovo arbitro, vedi caso d'uso 2.1.2 a pagina 4, oppure quando si visualizza la lista di gli arbitri creati dall'amministratore, vedi caso d'uso 2.1.4.

## 2.1.4 Caso d'uso: Visualizza arbitri

#### Descrizione

Questa funzionalità permette di visualizzare la lista di tutti gli arbitri creati dall'amministratore.

#### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo.

#### Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter visualizzare i propri arbitri.

## Post-condizioni

Nessuna post-condizione in quanto la visualizzazione degli arbitri non contribuisce a cambiare lo stato del sistema.

## Flusso principale

- 1. L'amministratore seleziona la voce "Utenti" nella pagina in cui si trova;
- 2. L'amministratore selezione la voce "Arbitro" nella pagina utenti;
- 3. Il sistema visualizza tutti gli arbitri in ordine alfabetico mostrandone: id, cognome, nome, email, indirizzo e telefono. Il sistema, inoltre, visualizza per ogni utente un pulsante per eliminare l'utente e un pulsante per modificare l'utente.

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

## 2.1.5 Caso d'uso: Visualizza utenti

## Descrizione

Questa funzionalità permette di visualizzare la lista di tutti gli utenti creati dall'amministratore.

#### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo.

## Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter visualizzare i propri utenti.

#### Post-condizioni

Nessuna post-condizione in quanto la visualizzazione degli utenti non contribuisce a cambiare lo stato del sistema.

## Flusso principale

- 1. L'amministratore seleziona la voce "Utenti" nella pagina in cui si trova;
- 2. Il sistema visualizza tutti gli utenti in ordine alfabetico mostrandone: id, cognome, nome, email, indirizzo e telefono. Il sistema, inoltre, visualizza per ogni utente un pulsante per eliminare l'utente.

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

## Punti di specializzazione

Questo caso d'uso si può specializzare in Visualizza Arbitri, vedi caso d'uso 2.1.4 a pagina 6, oppure Visualizza gestori squadra, vedi caso d'uso 2.1.6.

## 2.1.6 Caso d'uso: Visualizza gestori squadra

#### Descrizione

Questa funzionalità permette di visualizzare la lista di tutti i gestori di squadra creati dall'amministratore.

## Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesi-

#### Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter visualizzare i propri gestori di squadra.

#### Post-condizioni

Nessuna post-condizione in quanto la visualizzazione dei gestori di squadra non contribuisce a cambiare lo stato del sistema.

- 1. L'amministratore seleziona la voce "Utenti" nella pagina in cui si trova;
- 2. L'amministratore selezione la voce "Gestore" nella pagina utenti;

3. Il sistema visualizza tutti gli gestore di squadra in ordine alfabetico mostrandone: id, cognome, nome, email, indirizzo e telefono. Il sistema, inoltre, visualizza per ogni utente un pulsante per eliminare l'utente e un pulsante per modificare l'utente.

## Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

## 2.1.7 Caso d'uso: Inserisci gestore squadra

## Descrizione

Questa funzionalità permette all'amministratore di creare un nuovo utente di tipo gestore squadra.

#### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore interagisce direttamente con il sistema compilando i campi richiesti.

Gestore squadra, partecipazione non attiva in quanto si procede solo alla creazione di tale attore.

#### Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter inserire un gestore di una squadra.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'inserimento di un nuovo gestore di squadra. Il gestore di squadra è associato all'amministratore.

- 1. L'amministratore seleziona la voce "Nuovo Gestore Squadra" nella propria home page oppure seleziona la voce "Inserisci Nuovo Gestore Squadra"nella pagina di visualizzazione degli utenti;
- 2. L'amministratore inserisce per il gestore squadra i seguenti dati: nome, cognome, email, password, indirizzo, telefono e nome della squadra che andrà a gestire;
- 3. L'amministratore invia la richiesta di creazione del nuovo utente al sistema;

- 4. In caso di email già presente nel database viene notificato con un messaggio l'indisponibilità della email inserita invitando l'amministratore ad inserire una nuova mail;
- 5. Se tutti i dati sono stati inseriti correttamente, il sistema inserisce nel database il gestore della squadra e una nuova squadra associato ad esso;
- 6. Il sistema visualizza i dati del gestore appena inserito dando la possibilità all'amministratore di modificare i dati.

#### Flussi alternativi

Nel caso di inserimento errato dei dati da parte dell'amministratore o di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

#### 2.1.8 Caso d'uso: Modifica gestore squadra

#### Descrizione

Questa funzionalità permette all'amministratore di modificare i dati di un gestore squadra presente nel database.

#### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore interagisce direttamente con il sistema modificando alcuni dei campi richiesti.

Gestore squadra, partecipazione non attiva in quanto si procede solo alla modifica di tale attore.

## Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter modificare un gestore di una squadra.

#### Post-condizioni

Viene aggiornato lo stato sul database con la modifica di un gestore di una squadra.

- 1. L'amministratore seleziona la voce "Utenti" nella pagina in cui si trova;
- 2. L'amministratore seleziona la voce "Gestore" per visualizzare gli utenti di tipo gestore squadra;

- 3. L'amministratore seleziona la voce "Modifica", per il gestore squadra che desidera modificare, nella lista degli arbitri;
- 4. L'amministratore può modificare per l'arbitro i seguenti dati: nome, cognome, email, password, indirizzo, telefono e squadra;
- 5. L'amministratore invia la richiesta di modifica dell'arbitro al sistema;
- 6. In caso di modifica dell'email con l'inserimento di una email già presente nel database, viene notificato, con un messaggio, l'indisponibilità della email inserita invitando l'amministratore ad inserire una nuova mail:
- 7. Se tutti i dati sono stati inseriti correttamente, il sistema modifica il gestore squadra nel database;
- 8. Il sistema visualizza i dati del gestore squadra appena modificato dando la possibilità all'amministratore di effettuare ulteriori modifiche.

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

#### Punti di inclusione

Si può accedere a questo caso d'uso quando si effettua l'inserimento di un nuovo gestore squadra, vedi caso d'uso 2.1.7 a pagina 9, oppure quando si visualizza la lista di tutti i gestori squadra creati dall'amministratore, vedi caso d'uso 2.1.6 a pagina 8.

## 2.1.9 Caso d'uso: Elimina utente

#### Descrizione

Questa funzionalità permette all'amministratore di eliminare un utente registrato e, conseguentemente, di invalidare l'accesso delle credenziali dell'utente eliminato all'applicazione web.

#### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore interagisce direttamente con il sistema eliminando l'utente selezionato.

#### Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter eliminare i propri utenti.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'eliminazione dell'utente selezionato.

## Flusso principale

- 1. L'amministratore seleziona la voce "Elimina", per l'utente che desidera eliminare, nella lista degli utenti;
- 2. Il sistema chiede la conferma di esecuzione dell'operazione;
- 3. Se l'amministratore conferma l'operazione l'utente viene eliminato altrimenti l'operazione viene annullata.

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

#### Punti di inclusione

Si può accedere a questo caso d'uso quando si effettua la visualizzazione di tutti gli utenti, vedi caso d'uso 2.1.5 a pagina 7, quando si visualizza la lista di tutti gli arbitri creati dall'amministratore, vedi caso d'uso 2.1.4 a pagina 6, quando si visualizza la lista di tutti i gestori squadra creati dall'amministratore, vedi caso d'uso 2.1.6 a pagina 8.

## 2.1.10 Caso d'uso: Inserisci torneo eliminazione

#### Descrizione

Questa funzionalità permette all'amministratore di creare un nuovo torneo ad eliminazione diretta.

#### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore interagisce direttamente con il sistema compilando i campi richiesti.

#### Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter creare un torneo ad eliminazione.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'inserimento di un nuovo torneo ad eliminazione.

## Flusso principale

- 1. L'amministratore seleziona la voce "Nuovo Torneo Eliminazione" nella propria home page o nella pagina di visualizzazione dei tornei;
- 2. Il sistema restituisce la pagina web dove l'amministratore inserirà: il nome del torneo, il numero di squadre partecipanti e una descrizione relativa al torneo;
- 3. L'amministratore prosegue con la creazione cliccando sul pulsante "Continua";
- 4. Il sistema controlla che tutti i dati siano stati inseriti correttamente e visualizza la lista delle squadre complete presenti sul sistema (una squadra è completa quando il gestore di squadra ha compilato tutti i campi relativi alla propria squadra altrimenti la squadra non può partecipare ai tornei).
- 5. L'amministratore seleziona le squadre da iscrivere al torneo;
- 6. Il sistema controlla che sia stato inserito il giusto numero di squadre;
- 7. Il sistema chiede la conferma all'amministratore e, in caso affermativo, il sistema crea il torneo;
- 8. Il sistema notifica il successo dell'operazione all'amministratore visualizzando il tabellone degli incontri.

## Flussi alternativi

Nel caso di inserimento errato dei dati da parte dell'amministratore o di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

## Punti di estensione

Si può accedere a questo caso d'uso quando si effettua la visualizzazione della lista di tutti i tornei creati dall'amministratore, vedi caso d'uso 2.1.11, ma si tratta di un accesso opzionale.

## 2.1.11 Caso d'uso: Visualizza tornei

#### Descrizione

Questa funzionalità permette all'amministratore di vedere i tornei da lui creati e di monitorarne l'andamento.

#### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore richiede la visualizzazione delle pagine.

#### Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login.

#### Post-condizioni

Nessuna post-condizione in quanto la visualizzazione dei tornei non contribuisce a cambiare lo stato del sistema.

## Flusso principale

- 1. L'amministratore seleziona la voce "Tornei" nella pagina in cui si trova;
- 2. Il sistema invia la pagina richiesta con la lista dei tornei creati dall'amministratore.

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

#### 2.1.12 Caso d'uso: Inserisci torneo italiana

#### Descrizione

Questa funzionalità permette all'amministratore di creare un nuovo torneo all'italiana.

### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore interagisce direttamente con il sistema compilando i campi richiesti.

## Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter creare un torneo all'italiana.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'inserimento di un nuovo torneo all'italiana.

## Flusso principale

1. L'amministratore seleziona la voce "Nuovo Torneo Italiana" nella propria home page o nella pagina di visualizzazione dei tornei;

- 2. Il sistema restituisce la pagina web dove l'amministratore inserirà: il nome del torneo, il numero di squadre partecipanti e una descrizione relativa al torneo;
- 3. L'amministratore prosegue con la creazione cliccando sul pulsante "Continua";
- 4. Il sistema controlla che tutti i dati siano stati inseriti correttamente e visualizza la lista delle squadre complete presenti sul sistema (una squadra è completa quando il gestore di squadra ha compilato tutti i campi relativi alla propria squadra altrimenti la squadra non può partecipare ai tornei).
- 5. L'amministratore seleziona le squadre da iscrivere al torneo;
- 6. Il sistema controlla che sia stato inserito il giusto numero di squadre;
- 7. Il sistema chiede la conferma all'amministratore e, in caso affermativo, il sistema crea il torneo;
- 8. Il sistema notifica il successo dell'operazione all'amministratore visualizzando il tabellone delle varie giornate.

#### Flussi alternativi

Nel caso di inserimento errato dei dati da parte dell'amministratore o di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

#### Punti di estensione

Si può accedere a questo caso d'uso quando si effettua la visualizzazione della lista di tutti i tornei creati dall'amministratore, vedi caso d'uso 2.1.11 a pagina 13, ma si tratta di un accesso opzionale.

#### Diagramma delle attività

Il diagramma delle attività per il caso d'uso "Inserisci torneo all'italiana" è illustrato nella figura 2 a pagina 18.

## 2.1.13 Caso d'uso: Visualizza storico referti

## Descrizione

Questa funzionalità permette all'amministratore di vedere tutti i referti che l'amministratore ha assegnato ai rispettivi arbitri.

## Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore richiede la visualizzazione delle pagine.

#### Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter visualizzare tale pagina.

#### Post-condizioni

Nessuna post-condizione in quanto la visualizzazione dello storico dei referti non contribuisce a cambiare lo stato del sistema.

## Flusso principale

- 1. L'amministratore seleziona la voce "Referti" nella pagina in cui si tro-
- 2. Il sistema invia la pagina richiesta con la lista dei referti e dei rispettivi arbitri a cui sono stati assegnati dall'amministratore.

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

## 2.1.14 Caso d'uso: Assegna arbitro a partita

#### Descrizione

Questa funzionalità permette all'amministratore di assegnare un arbitro ad una partita di un determinato torneo, associando all'arbitro un referto che dovrà essere compilato al termine della partita.

#### Attori coinvolti

Amministratore, partecipazione attiva dell'attore verso il caso d'uso medesimo. L'amministratore interagisce direttamente con il sistema compilando i campi richiesti.

Arbitro, partecipazione non attiva in quanto si procede solo all'assegnazione di un referto a tale attore.

### Pre-condizioni

È richiesto l'accesso al sistema come amministratore, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter assegnare un arbitro ad una partita di un torneo.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'inserimento di un nuovo referto e con l'associazione del referto appena creato con un arbitro presente nel database.

## Flusso principale

- 1. L'amministratore seleziona la voce "Assegna arbitro" nella propria home page oppure seleziona la voce "Assegna Arbitro a Partita" nella pagina di visualizzazione dei referti;
- 2. L'amministratore seleziona il torneo;
- 3. L'amministratore seleziona la giornata/fase del torneo;
- 4. Se per tale fase/giornata sono disponibili incontri non ancora assegnati, l'amministratore può selezionare la partita altrimenti il sistema visualizza un messaggio di indisponibilità delle partite;
- 5. L'amministratore sceglie un arbitro ed inserisce per la partita: data, orario e luogo;
- 6. L'amministratore preme il pulsante di "Conferma assegnazione".
- 7. Il sistema chiede conferma all'amministratore dell'assegnazione dell'arbitro alla partita. Se l'amministratore acconsente il sistema elabora i dati, altrimenti non avviene alcuna operazione e permette all'amministratore di modificare i dati inseriti;
- 8. Il sistema modifica lo stato del database aggiornando i dati della partita e inserendo un nuovo referto;
- 9. Il sistema notifica l'amministratore l'esito positivo dell'operazione, visualizzando il referto appena inserito nello storico dei referti.

## Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'amministratore il tipo di errore che si è verificato.

## Punti di estensione

Si può accedere a questo caso d'uso quando si effettua la visualizzazione della lista dello storico dei referti creati dall'amministratore, vedi caso d'uso 2.1.13 a pagina 15, ma si tratta di un accesso opzionale.

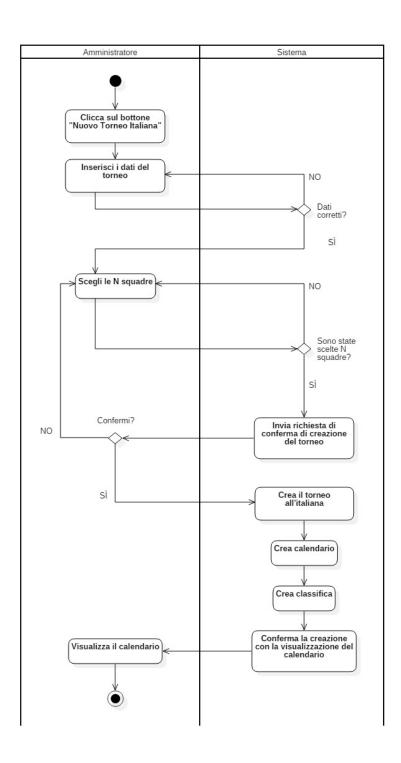


Figura 2: Diagramma delle attività della creazione di un torneo all'italiana

#### ATTORE GESTORE DELLA SQUADRA 2.2

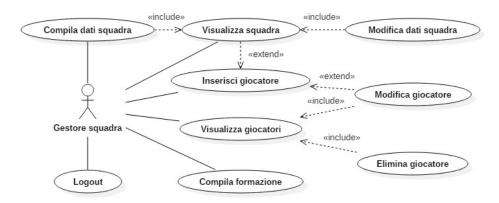


Figura 3: Casi d'uso dell'attore gestore della squadra

## Caso d'uso: Compila dati squadra

#### Descrizione

Questa funzionalità permette al gestore squadra di compilare i dati della propria squadra.

## Attori coinvolti

Gestore squadra, partecipazione attiva dell'attore verso il caso d'uso medesimo. Il gestore squadra interagisce direttamente con il sistema compilando i campi richiesti.

## Pre-condizioni

È richiesto il primo accesso al sistema come gestore squadra, tramite la funzionalità di login, in quanto solo lui può modificare i dati della sua squadra.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'aggiornamento dei dati di una squadra.

## Flusso principale

1. Il gestore della squadra seleziona la voce "Compila la tua squadra" nella pagina propria home page oppure seleziona la voce "Squadra" nella pagina in cui si trova;

- 2. Il gestore della squadra inserisce per la squadra i seguenti dati: nome, sede, foto della squadra, logo della squadra, nome dello sponsor, immagine dello sponsor e descrizione della squadra;
- 3. Il gestore della squadra seleziona la voce "Conferma Dati Squadra";
- 4. Il sistema controlla che il gestore della squadra abbia inserito correttamente tutti i dati. Se i dati sono corretti il sistema invia una richiesta di conferma dei dati inseriti al gestore della squadra;
- 5. Se il gestore della squadra conferma la modifica allora il sistema visualizza la pagina della squadra con i dati aggiornati;

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica al gestore della squadra il tipo di errore che si è verificato.

## 2.2.2 Caso d'uso: Visualizza squadra

#### Descrizione

Questa funzionalità permette al gestore della squadre di visualizzare la propria squadra.

#### Attori coinvolti

Gestore squadra, partecipazione attiva dell'attore verso il caso d'uso medesimo.

## Pre-condizioni

È richiesto l'accesso al sistema come gestore squadra, tramite la funzionalità di login.

## Post-condizioni

Nessuna post-condizione in quanto la visualizzazione della squadra non contribuisce a cambiare lo stato del sistema.

- 1. Il gestore della squadra seleziona la voce "Visualizza Squadra" nella sua home page oppure seleziona nella pagina in cui si trova la voce "Squadra";
- 2. Il sistema visualizza i dati della squadra mostrandone: nome, logo, immagine della squadra, sede, descrizione, nome dello sponsor e logo dello sponsor.

## Flussi alternativi

Nel caso il gestore della squadra non abbia compilato i dati della squadra si attiva il caso d'uso Compila squadra, vedi caso d'uso 2.2.1 a pagina 19.

Nel caso di operazione non riuscita, si notifica al gestore della squadra il tipo di errore che si è verificato.

#### Punti di inclusione

Si può accedere a questo caso d'uso quando si compilano i dati della squadra, vedi caso d'uso 2.2.1 a pagina 19, oppure quando si modificano i dati della squadra, vedi caso d'uso 2.2.3.

## 2.2.3 Caso d'uso: Modifica dati squadra

## Descrizione

Questa funzionalità permette al gestore squadra di modificare i dati della propria squadra.

## Attori coinvolti

Gestore squadra, partecipazione attiva dell'attore verso il caso d'uso medesimo. Il gestore squadra interagisce direttamente con il sistema compilando i campi richiesti.

#### Pre-condizioni

È richiesto l'accesso al sistema come gestore squadra, tramite la funzionalità di login, in quanto solo lui può modificare i dati della sua squadra.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'aggiornamento dei dati di una squadra.

- 1. Il gestore della squadra seleziona la voce "Modifica" nella pagina di visualizzazione della squadra oppure seleziona la voce "Inserisci Nuovo Giocatore" nella pagina di visualizzazione dei giocatori;
- 2. Il gestore della squadra inserisce per la squadra i seguenti dati: nome, sede, foto della squadra, logo della squadra, nome dello sponsor, immagine dello sponsor e descrizione della squadra;
- 3. Il gestore della squadra seleziona la voce "Conferma Modifiche";
- 4. Il sistema controlla che il gestore della squadra abbia inserito correttamente tutti i dati. Se i dati sono corretti il sistema invia una richiesta di conferma dei dati inseriti al gestore della squadra;

5. Se il gestore della squadra conferma la modifica allora il sistema visualizza la pagina della squadra con i dati aggiornati;

## Flussi alternativi

Nel caso di operazione non riuscita, si notifica al gestore della squadra il tipo di errore che si è verificato.

## Diagramma delle attività

Il diagramma delle attività per il caso d'uso "Modifica squadra" è illustrato nella figura 4.

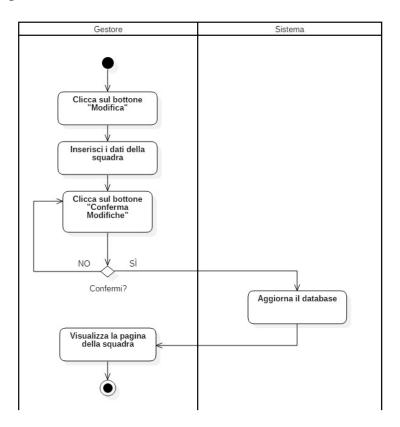


Figura 4: Diagramma delle attività di modifica della squadra

## Caso d'uso: Inserisci giocatore

## Descrizione

Questa funzionalità permette al gestore squadra di inserire nella rosa della propria squadra un nuovo giocatore.

#### Attori coinvolti

Gestore squadra, partecipazione attiva dell'attore verso il caso d'uso medesimo. Il gestore squadra interagisce direttamente con il sistema compilando i campi richiesti.

#### Pre-condizioni

È richiesto l'accesso al sistema come gestore squadra, tramite la funzionalità di login, in quanto solo lui può inserire un nuovo giocatore nella sua squadra. Il gestore squadra può inserire al massimo 36 giocatori.

## Post-condizioni

Viene aggiornato lo stato sul database con l'inserimento di un nuovo giocatore.

## Flusso principale

- 1. Il gestore squadra seleziona la voce "Nuovo Giocatore" nella propria home page oppure seleziona la voce "Inserisci Nuovo Giocatore" nella pagina di visualizzazione della squadra oppure seleziona la voce "Inserisci Nuovo Giocatore" nella pagina di visualizzazione dei giocatori;
- 2. Il gestore squadra inserisce per il giocatore i seguenti dati: nome, cognome, data di nascita, nazionalità, numero di maglia, ruolo, foto e descrizione;
- 3. Il gestore squadra invia la richiesta di creazione del nuovo giocatore al sistema;
- 4. Il sistema controlla che il gestore della squadra abbia inserito correttamente tutti i dati. Se i dati sono corretti il sistema crea un nuovo giocatore;
- 5. Il sistema visualizza i dati del giocatore appena inserito dando la possibilità al gestore della squadra di modificarli;

## Flussi alternativi

Nel caso di operazione non riuscita, si notifica al gestore della squadra il tipo di errore che si è verificato.

## Punti di estensione

Si può accedere a questo caso d'uso quando si effettua la visualizzazione dei dati della squadra, vedi caso d'uso 2.2.2 a pagina 20, oppure quando si modificano i dati di un giocatore della squadra, vedi caso d'uso 2.2.5 nella pagina successiva, ma si tratta in entrambi i casi di un accesso opzionale.

## 2.2.5 Caso d'uso: Modifica giocatore

#### Descrizione

Questa funzionalità permette al gestore squadra di modificare un giocatore della propria squadra.

#### Attori coinvolti

Gestore squadra, partecipazione attiva dell'attore verso il caso d'uso medesimo. Il gestore squadra interagisce direttamente con il sistema compilando i campi richiesti.

#### Pre-condizioni

È richiesto l'accesso al sistema come gestore squadra, tramite la funzionalità di login, in quanto solo lui può modificare un giocatore della sua squadra.

## Post-condizioni

Viene aggiornato lo stato sul database con l'aggiornamento dei dati di un giocatore.

## Flusso principale

- 1. Il gestore squadra seleziona la voce "Modifica" corrispondente al giocatore che vuole modificare, nella pagina di visualizzazione dei giocatori della squadra;
- 2. Il gestore squadra può modificare per il giocatore i seguenti dati: nome, cognome, data di nascita, nazionalità, numero di maglia, ruolo, foto e descrizione;
- 3. Il gestore squadra invia la richiesta di modifica del giocatore al sistema;
- 4. Il sistema controlla che il gestore della squadra abbia inserito correttamente tutti i dati. Se i dati sono corretti il sistema modifica i dati del giocatore;
- 5. Il sistema visualizza i dati del giocatore appena inserito dando la possibilità al gestore della squadra di modificarli;

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica al gestore della squadra il tipo di errore che si è verificato.

## Punti di inclusione

Si può accedere a questo caso d'uso quando si visualizzano i giocatori della squadra, vedi caso d'uso 2.2.6 nella pagina seguente.

## 2.2.6 Caso d'uso: Visualizza giocatori

## Descrizione

Questa funzionalità permette di visualizzare la lista di tutti gli i giocatori creati da un gestore di una squadra.

#### Attori coinvolti

Gestore squadra, partecipazione attiva dell'attore verso il caso d'uso medesimo.

#### Pre-condizioni

È richiesto l'accesso al sistema come gestore squadra, tramite la funzionalità di login.

#### Post-condizioni

Nessuna post-condizione in quanto la visualizzazione dei giocatori non contribuisce a cambiare lo stato del sistema.

## Flusso principale

- 1. Il gestore della squadra seleziona la voce "Giocatori" nella pagina in cui si trova;
- 2. Il sistema visualizza tutti i giocatori in ordine alfabetico mostrandone: foto, nome, cognome, numero maglia e ruolo. Il sistema, inoltre, visualizza per ogni utente un pulsante per eliminare l'utente e il pulsante per modificare l'utente.

## Flussi alternativi

Nel caso di operazione non riuscita, si notifica al gestore della squadra il tipo di errore che si è verificato.

## 2.2.7 Caso d'uso: Elimina giocatore

#### Descrizione

Questa funzionalità permette al gestore della squadra di eliminare un giocatore dalla rosa della squadra che gestisce.

## Attori coinvolti

Gestore squadra, partecipazione attiva dell'attore verso il caso d'uso medesimo. Il gestore squadra interagisce direttamente con il sistema eliminando il giocatore selezionato.

#### Pre-condizioni

È richiesto l'accesso al sistema come gestore squadra, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter eliminare i propri giocatori.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'eliminazione del giocatore selezionato.

## Flusso principale

- 1. Il gestore della squadra seleziona la voce "Elimina", per il giocatore che desidera eliminare, nella lista dei giocatori;
- 2. Il sistema chiede la conferma di esecuzione dell'operazione;
- 3. Se il gestore della squadra conferma l'operazione il giocatore viene eliminato altrimenti l'operazione viene annullata.

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica al gestore della squadra il tipo di errore che si è verificato.

## Punti di inclusione

Si può accedere a questo caso d'uso quando si visualizzano i giocatori della squadra, vedi caso d'uso 2.2.6 nella pagina precedente.

## 2.2.8 Caso d'uso: Compila formazione

#### Descrizione

Questa funzionalità permette al gestore squadra di scegliere i giocatori che faranno parte ad una formazione in una determinata partita.

#### Attori coinvolti

Gestore squadra, partecipazione attiva dell'attore verso il caso d'uso medesimo.

Il gestore squadra interagisce direttamente con il sistema scegliendo 11 giocatori titolari e 7 riserve.

#### Pre-condizioni

È richiesto l'accesso al sistema come gestore squadra, tramite la funzionalità di login, in quanto solo lui può compilare la formazione per una partita.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'inserimento della formazione scelta dal gestore della squadra.

## Flusso principale

- 1. Il gestore della squadra seleziona la voce "Nuova Formazione" nella propria home page oppure seleziona la voce "Formazioni" nella pagina di visualizzazione dei giocatori;
- 2. Il sistema visualizza la lista di tutte le partite a cui il gestore deve fornire la formazione dove per ogni partita vengono visualizzate le seguenti informazioni: id del referto, nome del torneo, le due squadre sfidanti e la data della partita. Il sistema, inoltre, visualizza per ogni partita un pulsante per compilare la formazione della squadra per la partita corrispondente;
- 3. Il gestore della squadra seleziona una partita;
- 4. Il sistema visualizza la lista di tutti i giocatori che fanno parte della squadra con a lato la possibilità di scegliere se far giocare un giocatore come titolare o come riserva;
- 5. Il gestore della squadra sceglie 11 giocatori titolari e 7 giocatori riserve;
- 6. Il gestore seleziona la voce "Conferma Formazione";
- 7. Il sistema controlla che il gestore della squadra abbia selezionato correttamente i giocatori. Se i giocatori sono stati selezionati correttamente il sistema invia una richiesta di conferma della formazione inserita al gestore della squadra altrimenti invia una notifica dell'errore commesso al gestore della squadra;
- 8. Se il gestore della squadra conferma la formazione allora il sistema aggiorna il database inserendo la formazione appena compilata e visualizza la lista di tutte le partite rimanenti a cui il gestore deve fornire la formazione;

## Flussi alternativi

Nel caso di operazione non riuscita, si notifica al gestore della squadra il tipo di errore che si è verificato.

## 2.2.9 Caso d'uso: Logout

## Descrizione

Questa funzionalità permette al gestore della squadra di uscire dalla sua pagina e di iniziare a navigare come utente pubblico. Il pulsante di Logout è presente in ogni pagina dell'applicazione web, in alto a destra.

## Attori coinvolti

Gestore della squadra, partecipazione attiva dell'attore verso il caso d'uso medesimo.

#### Pre-condizioni

È richiesto l'accesso al sistema tramite la funzionalità di login.

## Post-condizioni

Termina la sessione di accesso al sistema in qualità di gestore della squadra.

## Flusso principale

- 1. Il gestore della squadra seleziona nella pagina in cui si trova il pulsante di Logout;
- 2. Il sistema scollega l'amministratore dalle pagine a lui riservate;
- 3. Il sistema redireziona il gestore della squadra nella schermata di login e da quel momento può navigare come utente pubblico.

#### Flussi alternativi

Nel caso di operazione non riuscita si notifica all'amministratore il tipo di errore che si è verificato.

## Diagramma delle attività

Il diagramma delle attività per il caso d'uso "Logout" è illustrato nella figura 7 a pagina 34.

#### ATTORE ARBITRO 2.3

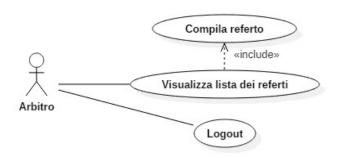


Figura 5: Casi d'uso dell'attore arbitro

## Caso d'uso: Compila referto

#### Descrizione

Questa funzionalità permette all'arbitro di compilare un referto che gli è stato assegnato relativo ad una partita.

#### Attori coinvolti

Arbitro, partecipazione attiva dell'attore verso il caso d'uso medesimo.

Amministratore, partecipazione non attiva dell'attore verso il caso d'uso medesimo in quanto l'amministratore è responsabile solo della creazione della partita che l'arbitro dovrà dirigere.

Gestore squadra, partecipazione non attiva dell'attore verso il caso d'uso medesimo in quanto il gestore della squadra ha fornito la formazione della squadra per la partita a cui riferisce il referto.

#### Pre-condizioni

È richiesto l'accesso al sistema come arbitro, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter compilare il referto di una partita. È richiesto che l'arbitro sia stato assegnato alla partita dall'amministratore. È necessario che entrambe le squadre abbiano fornito la formazione per la partita a cui fa riferimento il referto.

#### Post-condizioni

Viene aggiornato lo stato sul database con l'aggiornamento del referto e viene aggiornata la classifica delle due squadre che hanno disputato l'incontro. In caso di torneo ad eliminazione diretta si provvede a qualificare al turno successivo la squadra che ha vinto.

## Flusso principale

- 1. L'arbitro seleziona la voce "Compila" nella pagina di visualizzazione dei referti;
- 2. Il sistema visualizza la pagina per l'inserimento dell'orario di inizio e l'orario di fine della partita;
- 3. L'arbitro inserisce gli orari effettivi di inizio e fine della partita e successivamente seleziona la voce "Conferma Orari";
- 4. Il sistema controlla i dati inseriti e se sono corretti visualizza le due formazioni con i giocatori di entrambe le squadre. Per ogni giocatore il sistema visualizza: foto del giocatore, numero maglia, nome e cognome del giocatore, campo per inserire il numero di goal effettuati e campo per inserire il numero di ammonizioni. Il sistema, inoltre, visualizza il numero di autogol effettuati per ciascuna squadra;
- 5. L'arbitro inserisce i campi richiesti e conferma l'inserimento selezionando la voce "Conferma";
- 6. Il sistema chiede conferma all'arbitro della compilazione del referto della partita. Se l'arbitro acconsente il sistema elabora i dati, altrimenti non avviene alcuna operazione e permette all'arbitro di modificare i dati inseriti;

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'arbitro il tipo di errore che si è verificato.

#### Diagramma delle attività

Il diagramma delle attività per il caso d'uso "Compila referto" è illustrato nella figura 6 a pagina 33.

## Punti di inclusione

Si può accedere a questo caso d'uso quando si visualizza la lista dei referti, vedi caso d'uso 2.3.2.

## 2.3.2 Caso d'uso: Visualizza lista dei referti

#### Descrizione

Questa funzionalità permette all'arbitro di vedere tutti i referti che deve compilare o che ha già compilato.

## Attori coinvolti

Arbitro, partecipazione attiva dell'attore verso il caso d'uso medesimo.

#### Pre-condizioni

È richiesto l'accesso al sistema come arbitro, tramite la funzionalità di login, in quanto solo lui ha il privilegio di poter visualizzare tale pagina.

#### Post-condizioni

Nessuna post-condizione in quanto la visualizzazione dello dei referti non contribuisce a cambiare lo stato del sistema.

## Flusso principale

- 1. L'arbitro seleziona la voce "Nuovo Referto" nella propria home page oppure seleziona la voce "Referti" nella pagina in cui si trova;
- 2. Il sistema visualizza tutti i referti in ordine di id mostrandone: id del referto, nome del torneo, squadre sfidanti e data della partita. Il sistema, inoltre, visualizza per ogni referto un pulsante per compilarlo oppure visualizza "Compilato" se il referto è già stato compilato.

#### Flussi alternativi

Nel caso di operazione non riuscita, si notifica all'arbitro il tipo di errore che si è verificato.

## 2.3.3 Caso d'uso: Logout

## Descrizione

Questa funzionalità permette all'arbitro di uscire dalla sua pagina e di iniziare a navigare come utente pubblico. Il pulsante di Logout è presente in ogni pagina dell'applicazione web, in alto a destra.

#### Attori coinvolti

Arbitro, partecipazione attiva dell'attore verso il caso d'uso medesimo.

#### Pre-condizioni

È richiesto l'accesso al sistema tramite la funzionalità di login.

#### Post-condizioni

Termina la sessione di accesso al sistema in qualità di arbitro.

- 1. L'arbitro seleziona nella pagina in cui si trova il pulsante di Logout;
- 2. Il sistema scollega l'arbitro dalle pagine a lui riservate;

3. Il sistema redireziona l'arbitro nella schermata di login e da quel momento può navigare come utente pubblico.

## Flussi alternativi

Nel caso di operazione non riuscita si notifica all'arbitro il tipo di errore che si è verificato.

## Diagramma delle attività

Il diagramma delle attività per il caso d'uso "Logout" è illustrato nella figura 7 a pagina 34.

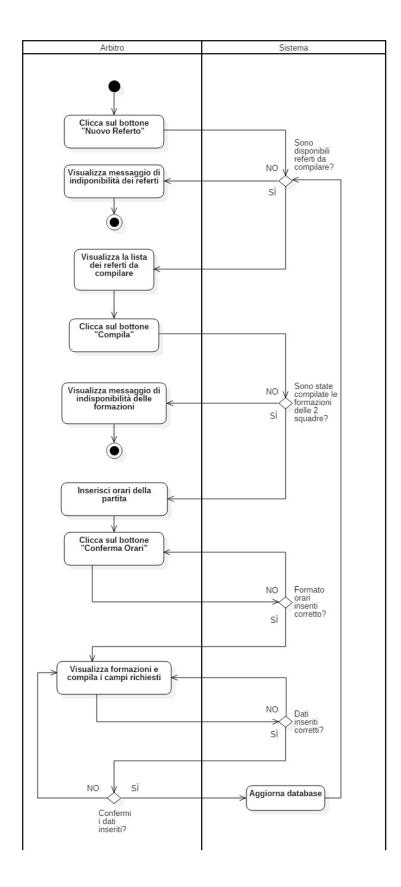


Figura 6: Diagramma delle attività della compilazione di un referto

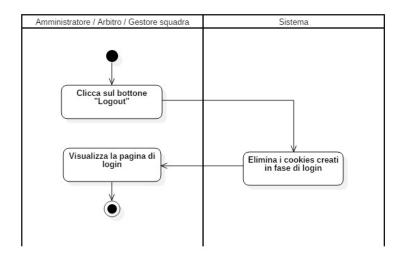


Figura 7: Diagramma delle attività del logout di un utente

#### 2.4 ATTORE UTENTE PUBBLICO



Figura 8: Casi d'uso dell'attore utente pubblico

# Caso d'uso: Visualizza pagine statiche

#### Descrizione

Questa funzionalità permette all'utente pubblico di visualizzare tutte le pagine richieste con contenuto non dinamico.

#### Attori coinvolti

Utente pubblico, partecipazione attiva, richiedendo la visualizzazione delle pagine.

## Pre-condizioni

Nessuna precondizione, in quanto non serve uno stato particolare per visualizzare una pagina statica.

#### Post-condizioni

Nessuna postcondizione, in quanto la visualizzazione di una pagina statica non altera lo stato dell'applicazione.

## Flusso principale

- 1. Richiesta di qualsiasi pagina che abbia contenuto statico da parte dell'utente pubblico
- 2. Risposta del server con la pagina statica richiesta da parte dell'utente pubblico;

#### Flussi alternativi

Non presenti.

# 2.4.2 Caso d'uso: Effettua login

#### Descrizione

Questa funzionalità permette all'utente pubblico di farsi riconoscere dal sistema, accedendo quindi alle funzionalità che sono a lui riservate.

#### Attori coinvolti

Utente pubblico, partecipazione attiva dell'attore che passa da Utente Pubblico ad Amministratore o Gestore squadra o Arbitro.

#### Pre-condizioni

Compilazione del modulo di login.

## Post-condizioni

L'utente passa da attore Utente Pubblico ad Amministratore o Gestore squadra o Arbitro.

# Flusso principale

- 1. L'utente pubblico seleziona la voce "Login" nella pagina in cui si trova;
- 2. L'utente pubblico compila il modulo inserendo la propria email e la propria password;
- 3. L'utente pubblico seleziona la voce "Accedi";
- 4. Il sistema effettua verifica se le informazioni inserite nel campo email sono corrette dal punto di vista sintattico inviando all'utente pubblico, in caso di errore, una notifica con l'errore riscontrato;
- 5. Il sistema dopo aver ricevuto i dati il server controlla se nella base dati è presente un utente con indirizzo email e password ricevuti;
- 6. Se le informazioni esistono sulla base dati l'utente pubblico viene riconosciuto dal sistema. L'utente visualizza il suo nome e cognome con un messaggio di benvenuto che lo informa della sua tipologia. L'utente selezionando la voce "MyHome" visualizza la propria home page e un menù con tutte le operazioni che può eseguire sul sistema.

#### Flussi alternativi

Nel caso di inserimento di email e password non presenti nel database vengono notificate all'utente pubblico le possibili cause di errore.

# Diagramma delle attività

Il diagramma delle attività per il caso d'uso "Effettua login" è illustrato nella figura 9 nella pagina seguente.

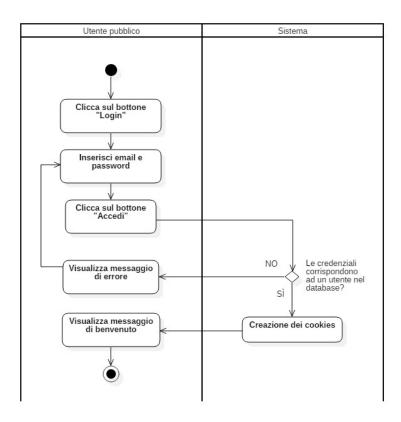


Figura 9: Diagramma delle attività del login di un utente pubblico

# 3 | MODELLO DEL DOMINIO

# 3.1 DIAGRAMMA DELLE CLASSI

Si presenta ora il modello delle classi di dominio. Il modello delle classi di dominio è un modello statico dello spazio del problema che fornisce una visione strutturale "logica" del sistema da realizzare. Il modello delle classi di dominio descrive entità e concetti propri dell'area funzionale che il sistema dovrà implementare.

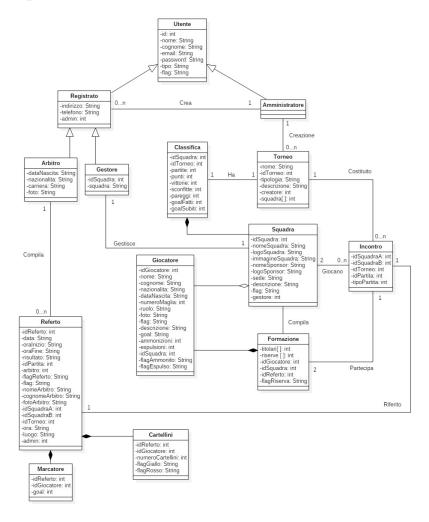


Figura 10: Diagramma delle classi di dominio

N.B. Al fine della leggibilità del diagramma delle classi si è scelto di non inserire i metodi delle varie classi. I metodi delle classi sono specificati nel modello di design delle classi *Entity* a pagina 44.

# 4 MODELLO DI DESIGN

# 4.1 DIAGRAMMA DELLE CLASSI

Il raffinamento del modello delle classi di dominio avviene attraverso il modello di design suddividendo tutte le classi del progetto in tre categorie:

- 1. Classi Boundary;
- 2. Classi Control;
- 3. Classi Entity.

Di seguito vengono presentate le categorie sopracitate.

# 4.1.1 Classi Boundary

Sono le classi che si interfacciano con l'utente.

Le classi boundary rappresentano un punto di interfaciamento bidirezionale tra gli attori ed il sistema.

Nel caso dell'applicazione web, queste sono le singole pagine JSP che in fase di compilazione vengono convertite in Servlet (classi Java che estendono la classe HTTPServlet).

Gli attori sono: Admin (amministratore), Arbitro, Gestore (gestore di una squadra) e Guest (utente pubblico).

Si procede con l'elenco delle classi Boundary:

# Admin:

- Calendario.jsp
- dettagliTorneo.jsp
- dettagliUtente.jsp
- homeA.jsp
- newReferto.jsp
- newTorneo.jsp
- referti.jsp
- refertiView.jsp
- selezionaSquadre.jsp

- tornei.jsp
- utenti.jsp

# Arbitro:

- dettagliReferto.jsp
- homeR.jsp
- myReferti.jsp
- referti.jsp

# Gestore:

- dettagliGiocatore.jsp
- giocatori.jsp
- homeG.jsp
- newFormazione.jsp
- sceltaGiocatori.jsp
- squadre.jsp

# Guest:

- listaGiocatori.jsp
- listaSquadre.jsp
- listaTornei.jsp
- vediGiocatore.jsp
- vediReferto.jsp
- vediSquadra.jsp
- vediTorneo.jsp

# Tutti gli utenti:

- ErrorPage.jsp
- login.jsp
- home.jsp

#### Classi Control 4.1.2

Sono le classi che validano le interazioni dell'utente e si interfacciano con la base dati.

Queste classi incapsulano la logica applicativa dei casi d'uso sopra elencati. Il compito di queste classi è quello di coordinare il lavoro svolto da istanze delle classi Boundary ed Entity.

Nel caso dell'applicazione web, queste sono le classi della *Business Logic*, i Services associati e le classi che governano il flusso delle conversazioni.

# Package bflows:

Il package bflows (acronimo di BusinessFlows) si occupa del controllo del flusso delle conversazioni. Le classi di questo package usano le classi del package blogics (acronimo di *BusinessLogics*) che si occupano di mappare l'oggetto del database e di fornire i servizi interessati dalla conversazione medesima. Si nota che una conversazione può avere più classi Boundary associate.

# Package blogics:

Il package blogics si occupa di rendere disponibili tutte le funzioni e i servizi che l'applicazione web può offrire. I servizi presenti in questo package non sono vincolati in alcun modo a particolari interfacce o a determinate conversazioni, ma possono essere utilizzate in più contesti.

## Package service:

Le classi contenute in questo package sono classi di servizio. Infatti si può notare che il costruttore di tali classi è privato e che tutti i metodi delle classi sono statici.

In particolare si osserva nella classe DBService.java l'implementazione del pattern Singleton che gestisce la connessione al database.

databaseservice Le classi contenute in questa sottocartella della classe services, permettono di gestire la connessione al database.

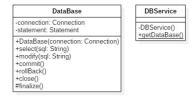


Figura 11: Classi della sotto-cartella databaseservice

errorservice Le classi contenute in questa sottocartella della classe services, permettono di gestire gli errori che si possono generare durante l'esecuzione dell'applicazione.



Figura 12: Classi della sotto-cartella errorservice

logservice Le classi contenute in questa sottocartella della classe services, permettono di scrivere i messaggi di errore in uno specifico file denominato log.



Figura 13: Classi della sotto-cartella logservice

sessionservice La classe contenute in questa sottocartella della classe services, permette di gestire la sessione tramite il protocollo HTTP.

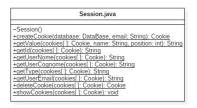


Figura 14: Classi della sotto-cartella sessionservice

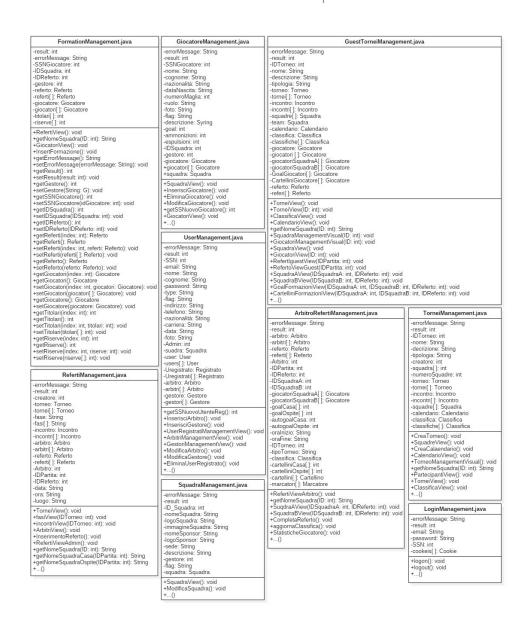


Figura 15: Classi del package bflows

N.B. Il simbolo +...() sta indica tutti i metodi getter e setter utili per far funzionare correttamente l'applicazione. Si è scelto di non inserirli per non aggiungere ulteriore complessità ai diagrammi.

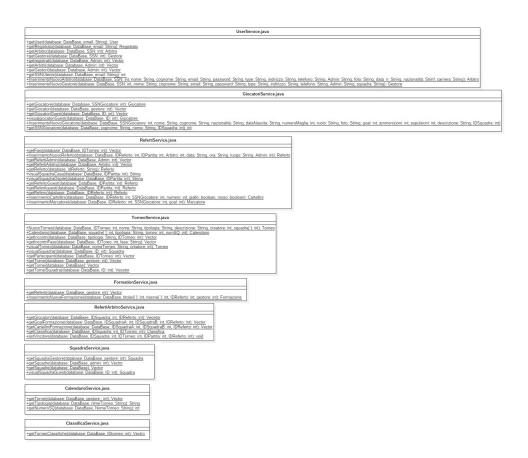


Figura 16: Classi service del package blogics

# Classi Entity

Sono le classi che mappano la struttura della base dati.

Le classi Entity modellano le informazioni gestite dal sistema e formalizzano la struttura logica dei dati, in modo da isolare i dettagli della memorizzazione dei dati e di limitare l'impatto di eventuali modifiche alla struttura del database.

Nel caso dell'applicazione web, queste sono le classi della Business Logic che implementano i metodi setter, getter e i costruttori per creare l'oggetto mediante passaggio di dati di tipo primitivo od oggetti quali RecordSet.

La figura 17 nella pagina successiva rappresenta le classi di questo tipo.

Signature in Process of the Control	Classifica.java					User.java		
Forestitute in Section in the Sectio	+IDSquadra; int			+SSN: int				
Forestitute in Section in the Sectio	+IDTomeo; int +Partite: int			+nome: String				
Secretary of Hospitals and Distance or Parties of Parties of North Control (1994) and Parties of Pa	+Punti: int			+email: St +passwore	tring d: Strina			
Counted in Change and Counted in Counter of Particle in Plants, or Victoria, in Scientific in Particle, and Scientific in Particle in Counter in Particle in P	+Sconfitte: int			+type: String +flog: String				
Casacles and Organics and Organ	+GoalFatti: int			+User(int SSN, String nome, String cognome, String email, String password, String type)				
**Causalon-Branches Colorisan London Horizon, broken propeging) and  **Causalon-Branches Colorisan London Horizon, broken propeging) and  **Causalon-Branches Colorisan London Horizon, Strong Section Strong Section London Horizon, Strong Section London Horizon, Strong Section London Horizon, Strong Section London Horizon, Strong Section London London Horizon, Strong Section London Lond	+Classifica(int IDSquadra, int IDTorneo, int Partite, int Punti, int Vittorie, int Sconfitte, int Pareggi, int GoalFatti, int GoalSubiti)			+User(ResultSet resultSet)				
Address jours  Addres	+Classifica(ResultSet resultSet)	, int Scotting, int I taleggi, int	oour au, iii oououbii)	- delete(D	undusc dum	nooy, voia		
Transport Strong Common Strong corporate, Strong practices, Strong	1 20 7	Arbitro in	ava.					
Althorized St. String comes. String controls. String stands. String passeord. String passeo	+data n: String	Arbitro.ja	ıva					
**Ameny Control Service Servic	+nazionalità: String +carriera: String							
Centron jam Control jam Contro	+Arbitro(ResultSet resultSet) +insert(DataBase database): void	sword, String type, String inc	firizzo, String telefono, int	Admin, Strin	g foto, String	data_n, String nazionalità, String carriera)		
## ## ## ## ## ## ## ## ## ## ## ## ##								
	+squadra: String					+riserve[]: int		
Projection (Particle database) void  **Registrato, juna  **Torneo, jun	+Gestore(int SSN, String nome, String cognome, String email, String password, String type, String indirizzo, String telefono, int Adn +Gestore(ResultSet resultSet)					+IDSquadra: int		
## Comparison of	+insert(DataBase database): void							
**International Sting **International Still Control Co						+Formazione(int[] titolari, int[] riserve, int IDSquadra, int IDRefe		
**Admin the Company of		.java				+hormazione(ResultSet resultSet) +insert(DataBase database): void		
Registance (SSL) String come, String captions, String email. String passwordt. String types, String indicates, String bedonin; int Admini Registance (String String	+telefon: String +Admin: int							
Tomore String ones String ones String tomore, Strin		password, String type, String	indirizzo, String telefono,	int Admin)				
Information in Common String nome, String topologia, String descrizione, int creatione, int Common String Information in Common String Information Information in Common String Information Informati					Incente	o iava		
**Ipologia Siring **Control Residence (I) int **Tomocylin Direce, Siring nome, Siring tipologia, Siring descrizone, int creative, int] squaled) **Tomocylin Direce, Siring nome, Siring tipologia, Siring descrizone, int creative, int] squaled) **Tomocylin Direce, Siring nome, Siring tipologia, Siring descrizone, int creative, interpretation interpretat	+nome: String							
**Complete Difference, Sirring nome, Sirring speciega, String descrizione, int creatore, intil sepadation in the control of th	+tipologia: String		+IDSquadraB: int					
Transcript ID Squadta, in IDSquadta, in IDSq	+creatore: int		+IDPartita: int					
** Innocentifications of interest and incompression of interest and interest	+squadra[]: int				tra A int IDS aundra B int IDT armon int IDP artita int name Acc)			
## Squadra.java  ## Squ	+Torneo(ResultSet resultSet)	int creatore, int[] squadra)	+Incontro(ResultSet res	sultSet)		,		
**ID. Squarks in it **	+insert(DataBase database): void		+upuale(Database data	abase, int ib	oquaura, iii i	Draittabopo, boolean casa, int nomeAss). Void		
**Constanting String **Constan		Squadra.ja	ava					
Hopsphonous String Hopsphonous S	+nomeSquadra: String							
**Construction in the Spring of String Control String Industrial Control String Industrial Control String Industrial Control In	+logoSquadra: String +immenine Squadra: String							
+socies String -fedestizations: Fedestizations: Fedestizations	+nomeSponsor: String							
reflage Sirring rejectore. Int  - Squadrafier ID. Squadra, Sirring nomeSquadra, String logoSponsor, int gestore, String flag) - Squadrafieration results of  - Squadrafieration results of  - Squadrafieration results of  - Squadrafieration results of  - Sponsor for the squadrafieration of  - String seed, String nomeSponsor, String logoSponsor, int gestore, String flag) - String results of  - String results	+sede: String							
**Special Control Department of the Septiment of Septimen								
*Squadra(ResulSet resulSet)  *SpiNisocatore int ** **nome String ** **capacitals. String ** **tuncerollagia, int ** **tuncerol	+gestore: int		- Ji	Ctrime	64	in land state of the state of t		
## Glocatore java  ## Glocatore	+Squadra(ResultSet resultSet)	uring immaginesquadra, suri	ig descrizione, string sede	, suring non	iesporisor, su	ing togosponsor, int gestore, string riag)		
**SSNiciocatore in nome String **cognome. String	-upunie(Danabase Unitabase). You		Giocatore iava					
**Congrome String **Congrome S	+SSNGiocatore; int		Giocatore.java					
-tdatalwascia: String -truster String -truster String -truster String -tdescrizione: str	+nome: String							
**ImmerChalgalis* int* **Lock String **Hore	+cognome: String	+cognome: String +nazionaliti's String						
-flore String -f	+cognome: String +nazionalità: String							
-flags String -f	+cognome: String +nazionalità: String +dataNascita: String +numeroMaglia: int							
**Space in the **space of the state of the s	+cognome: String +hazionalital: String +dataNascita: String +dutaNascita: String +ruolo: String +ruolo: String							
+Enginements String  +Disquadra. String ruolo, String foto, int goal, int ammonizioni, int espulsioni, String descrizione, int IDSquadra  *Coccatore(first SNGiocatore, String nome, String cognome, String nazionalità, String dataNascita, int numeroMaglia, String ruolo, String foto, int goal, int ammonizioni, int espulsioni, String descrizione, int IDSquadra  *Coccatore(first) espulsioni, String descrizione, int IDSquadra  *Coccatore(first) espulsioni, int espulsioni, String descrizione, int IDSquadra  *Coccatore(first) espulsioni, int es	+cognome. String +nazonalità: String +dataNasorta: String +dataNasorta: String +dataNasorta: String +dataNasorta: String +dataNasorta: String +loto String +loto String +loto String							
FilagEspulso String  *Giocatore(Filas String nome, String nome, String cognome, String nazionalità, String dataNascita, int numeroMaglia, String foto, int goal, int ammonizioni, int espulsioni, String descrizione, int (IDSquadra* Giocatore(FeathStef resultSef)  *Insert(DiataBase database) void  *dioleta ChataBase database) void  *dioleta ChataBase database) void  *dioleta ChataBase database) void  *updateStatistica(DataBase database) int gool, int amm, int esp) void  *UpdateStatistica(DataBase database) void  *updateStatistica(DataBase database) void  *IDReferto int  *dioleta intervention of the string tipologia, int	-cognone: String -nazionalità: String -dataflascata: String -dataflascata: String -dataflascata: String -trusico: String -trusico: String -trusico: String -descrizzione: String -descrizzione: String -goal: int							
*Ciocatore(Facts SNGiocatore, String nome, String cognome, String adataNascita, int numeroMaglia, String ruolo, String foto, int goal, int ammonizioni, int espulsioni, String descrizione, int IDSquadra finiser(DataBase database) void *update(DataBase dat	-cognome String -nazionalità String -datal·lascità String -datal·lascità String -datal·lascità String -fation S							
*Giocatore (ResultSet resultSet) *Insert(DataBase database): void *delete(DataBase database): void *delete(DataBase database): void *tepolate(DataBase database): void *tepolate(DataBase database): void *tepolate(DataBase database): void *tepolate(DataBase database): void *delete(DataBase database): void *delete(DataBase): void *delete(DataBase database): void *	-cognome-String -nazionalità-String -nazionalità-String -datalisascità-String -datalisascità-String -datalisascità-String -flotto-String -flo							
+doletic DataBase database): void  -typidate (DataBase database); void	-cognone String -trazionalità String -thatal fascota String -thatal fascota String -thatal fascota String -thatal fascota String -those String -those String -those String -these String							
**UpdateStatistica(DataBase database, int gool, int amm, int esp): void  **UpdateStatistica(DataBase database, int gool, int amm, int esp): void  **UpdateStatistica(DataBase database, int gool, int amm, int esp): void  **UpdateStatistica(DataBase database): void  **UpdateStatistica(DataBase database): void  **Cartellini, good, int International Engagement of the Cartellini, good, g	-cognome String -razionalità String -razionali	ionalità, String dataNascita, i	nt numeroMaglia, String ru	iolo, String f	foto, int goal, ii	nt ammonizioni, int espulsioni, String descrizione, int l'OSquadra		
Calendario_java  +iploopigis String +squadrie(] int +ipDremo: int +ind Torneo: int +calendario(int] squadre(, String tipologia, int IDtorneo, int numSQ) +calendario(int) [ data, int add): int +shift-filt(int) [ da	-cognone: String -thatchlascuta: String -thatchlascuta: String -thatchlascuta: String -thatchlascuta: String -thatchlascuta: String -thor: String -thor: String -thor: String -thescritione: String -thescritione: String -thescritione: String -thescritione: String -thescritione: String -thescritione: String -thatchlascuta	ionalità, String dataNascita, i	nt numeroMaglia, String ru	iolo, String f	foto, int goal, it	nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
**IpDeferto: int  -thDraferto:	-cognome String -racordalità. Sering -racordalità. Sering -racordalità. Sering -racordalità. Sering -racordalità. Sering -racord. String -raco	ionalità, String dataNascita, i	nt numeroMaglia, String ru	iolo, String f	foto, int goal, it	nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
- Squared [: int   HI former, int   HI f	-cognome String -razoralital. String -data/lascuta. String -doscuta. String -filage. String -doscuta. String -filage. Strin	ionalità, String dataNascita, i	nt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int IDSquadra,		
**ImmSQ; int **Calendarior(int)   squadre, String typologia, int   IDtorneo, int numSQ) **shift.leff(int)  clata, int add), int **ImmSqr(int)  challendarior   squadre, String typologia, int   IDtorneo, int numSQ) **ImmSqr(int)  challendarior   string   **ImmSqr(int)  challendarior   **ImmSqr(int)  challendarior   string   **ImmSqr(int)  challendarior   *	-cognome. String -trazionalità. String -tdatal fascula: String -tdot: String -tdot: String -tdot: String -tdoscrizone: S		nt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
+shifte(fird) data, int add), int -shift(shift(ind) data data data data data data data da	-cognome String -razonaltal. String -razonalta	+IDReferto: int +data: String	nt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
**Abriton: int **Abri	-cognome String -razionalità.	+IDReferto: int +data: String +Oralinzio: String +OraFine: String	mt numeroMaglia, String n.			nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
#Reference in the Cartellini_java  #IRReference in the Hispoianic String  #IRREGISTION STRING	-cognome. String -razionalità. Sering -razionalità.	+IDReferto: int +data: String +Oralnizio: String +OraFine: String +risultato: String	nt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int l'DSquadra		
+1DReferto: int +1DReferto: int +1Bug Sallos: String +1Bug Sallos: TesultSelf +1Bug Sallos: String +1Bug Sallos: S	-cognome. String -chard lascolar. String -chost String -cho	+IDReferto: int +data: String +Oralnizio: String +OraFine: String +risultato: String +IDPartita: int +Arbitro: int	nt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int (DSquadra		
+florArbitro: String +florArbitro: String +flagCalio: String +flagCalio: String +flagCalio: String +flagCalio: String +cartelling(rti DRefero, int SSNGiccatore, int numCart) +theretion till +theretion till +theretion till +Refero, firti DRefero, string data, string oralnizio, String oralnizio, String oralnizio, string orange interfloration till DPartita, int Arbitro, String ora, String luogo, int Admin) +theretion till +Refero, firti DRefero, int SSNGiccatore, int goal) +theretion till +th	-cognome String -recognisms Stri	+IDReferto int +data. String +Oralizero. String +Oralizero. String +risultato. String +IDPartita int +Arbitro int +BagReferto. String +Bag. String	nt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
+flumCart int Hagicalos. String Hagicalos. Hagical	-cognome. String -razionalità. Sering -razionalità.	+IDReferto: int +data: String +Oralinizo: String +Oralinizo: String +Insulation: String +IDPartita: int +Arbitro: int +IlagReferto: String +llag: String +romeArbitro: String +llag: String	mt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
+flagflosos String  -tradeflin(inf) DRefetro, int SSNGocatore, int numCart)  -tradeflin(inf) DRefetro, int SSNGocatore, int numCart)  -tradeflin(inf) Exelector, int numCart	-cognome. String -trustorials.	+IDReferto: int +data: String +Oralnizo: String +Oralnizo: String +Oralnizo: String +Oralnizo: int +IDPartia: int +Arbitro: int +IlagReferto: String +Ilag: String +oromeArbitro: String +cognomeArbitro: Nome +COGNOMEAR: int	nt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int (DSquadra		
+Admin: nt ** Cardelini(Flexit/Set result/Set) Marcatore,java IDReferro int IDReferro, String drata, String oralnizio, String oraFine, String risultato, int IDPartita, int Arbitro, String ora, String luogo, int Admin) Referro(mt IDReferro, String drata, String oralnizio, String oraFine, String risultato, int IDPartita, int Arbitro, String ora, String luogo, int Admin) Referro(mt IDReferro, int SSNGiocatore, int goal) Marcatore/fersulfSet resulfSet)	-cognome String -racionalità.	+IDRefero: int +IDRefero: int +Iditat String +Oralinize: String +Oralinize: String +raulato: String +raulato: String +raulato: String +raulato: String +Rag: String +Rag: String +Rag: String +Rag: String +NomeArbitro: String +Oralinize: String +IDSquadraB: int	nt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
+Insert(DataBase database, boolean giallo, boolean rosso): void  Marcatore,java  +IDReferto. int 45SN(Gocatore; int 4-goal: int 4-fibro, String data, String oralizio, String ora/Fine, String risultato, int IDPartita, int Arbitro, String luogo, int Admin)  +Referto/ResulfSet resulfSet)  +IDReferto. int 4-sSN(Gocatore; int 4-goal: int 4-fibro, String data, String oralizio, String ora/Fine, String risultato, int IDPartita, int Arbitro, String luogo, int Admin)  +Referto/ResulfSet resulfSet)  +Insert(DataBase database): void  +Update(DataBase database): void  +Marcatore/ResulfSet resulfSet)	-cognome String -racionalità.	*IDReferto. int *data. String *double String *OraFine. String *risultato. String *risultato. String *risultato. String *lagFerlato. String *lagFerlato. String *nomeArbitro. String *nomeArbitro. String *nomeArbitro. String *IDSquadraA. int *IDSquadraA. int *IDSquadraA. int *IDSquadraA. int *IDSquadraB. int *IDSq	nt numeroMaglia, String n.			nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
#IDRefertor. int #SNS(iocatore; int goal) #Marcator(int IDReferto, int SSNGiocatore; int goal) #Marcator(int IDReferto, int SSNGiocatore; int goal) #Marcator(int IDReferto, int SSNGiocatore; int goal)	-cognome String -inacronalitis Smring -inacronalitis Smring -inacronalitis Smring -inacronalitis Smring -inacronalitis Smring -indice String	+IDReferto: int +data: String +Oralnizo: String +Oralnizo: String +Oralnizo: String +IDParitte: int +InagPeferto: String +IlagPeferto: String +IlagPeferto: String +IlagPeferto: String +IlagPeferto: String +IlagPederto:	nt numeroMaglia, String ru			nt ammonizioni, int espulsioni, String descrizione, int IDSquadra		
+Update(DataBase database): void +SSNGiocatore: int +goal: int -Marcatore(firt IDFeferto, int SSNGiocatore, int goal) -Marcatore(set)USet resultSet)	-cognome String -inacronalita String -inacronalita String -inacronalita String -inacronalita String -inacronalita String -index String -inacronality -inacro	+IDReferto: int +data. String +Oralnizio: String +Oralnizio: String +Oralnizio: String +Oralnizio: String +Arbitro: int +lagReferto: String +lagReferto: String +lagReferto: String +louoArbitro: String +cognomeArbitro: String +cognomeArbitro: String +louoArbitro: String +louoArbitro: String +louoArbitro: String +Louon: String +Louon: String +Admin: int +Refertor(int IDReferto, String +Admin: int	ing data. String grafnizio, o		Referto.java			
+goal: int +Marcatore(int IDRefero, int SSNGiocatore, int goal) +Marcatore(ResultSet resultSet)	-cognome String -inacronalital String -inacronalital String -inacronalital String -inacronalital String -inacronalital String -index String -i	+IDReferto: int +IDReferto: int -data: String -Oralinizo: String -Oralinizo: String -resultato: String -resultato: String -resultato: String -resultato: String -resultato: String -resultation: String -resultation: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -resultation:	ing data, String oralnizio, i		Referto.java			
+Marcatore(ResultSet resultSet)	-cognome String -inacronalital Smring -inacronalital Smring -inacronalital Smring -inacronalital Smring -inacronalital Smring -inacronalital Smring -index String -index String -index String -index String -index String -inacronality -inacron	+IDReferto: int +IDReferto: int -data: String -Oralinizo: String -Oralinizo: String -resultato: String -resultato: String -resultato: String -resultato: String -resultato: String -resultation: String -resultation: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -resultation:	ing data, String oralnizio, i		Referto.java			
+insert(DataBase database): void	-cognome. String -recognish. Sering -recognish. Sering -recognish. Sering -recognish. Sering -record Sering -recognish. Sering	+IDReferto: int +IDReferto: int -data: String -Oralinizo: String -Oralinizo: String -resultato: String -resultato: String -resultato: String -resultato: String -resultato: String -resultation: String -resultation: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -nome-Arbitro: String -resultation:	ing data, String oralnizio, i		Referto.java			

Figura 17: Classi del package blogics

#### 4.2 DIAGRAMMI DI SEQUENZA

Si riporta a titolo di esempio un Sequence Diagram relativo alla creazione della formazione di una squadra per una partita.

Il gestore di una determinata squadra si occupa della creazione della formazione per una specifica partita di un torneo.

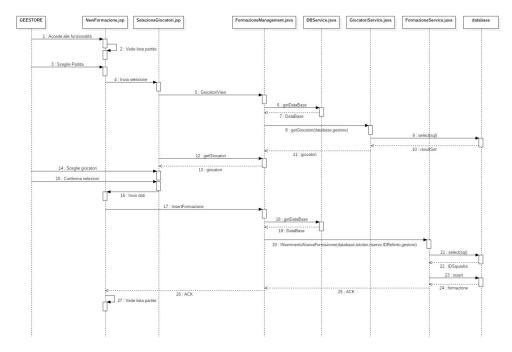


Figura 18: Diagramma di sequenza della creazione della formazione di una squadra