



# Maintenance Report

MyBomber

**Team:** Gaetano Mauro, Umberto Loria

**Riferimento:** MB\_MaintenanceReport

**Versione:** 1.0

**Data:** 18/07/2022



# Sommario

Sommario	2
Panoramica sistema attuale	5
Descrizione sistema attuale	5
Activity Diagram	5
Attori del sistema	5
Requisiti funzionali	6
Architettura del sistema	8
Gestione dati persistenti	10
<b>Statechart evento</b>	<b>15</b>
<b>Attività di manutenzione</b>	<b>16</b>
Change request 1 - Cifratura password	16
Analisi richiesta modifica	16
Impact Analysis	17
Identificazione SIS (Starting Impact Set)	17
Identificazione CIS (Candidate Impact Set)	18
Identificazione AIS (Actual Impact Set)	19
Identificazione FPIS (False Positive Impact Set)	21
Identificazione DIS (Discovered Impact Set)	21
Metriche Impact Analysis	21
Change request 2 - Entità squadra	22
Analisi richiesta modifica	22
Impact Analysis	23
Identificazione SIS (Starting Impact Set)	23
Identificazione CIS (Candidate Impact Set)	26
Identificazione AIS (Actual Impact Set)	29
Identificazione FPIS (False Positive Impact Set)	33
Identificazione DIS (Discovered Impact Set)	33



Metriche Impact Analysis	33
Change request 3 - Aggiornamento ID Evento	34
Analisi richiesta modifica	34
Impact Analysis	34
Identificazione SIS (Starting Impact Set)	34
Identificazione CIS (Candidate Impact Set)	35
Identificazione AIS (Actual Impact Set)	36
Identificazione FPIS (False Positive Impact Set)	37
Identificazione DIS (Discovered Impact Set)	37
Metriche Impact Analysis	38
Change request 4 - Evento annullato, iniziato e finito	38
Analisi richiesta modifica	38
Impact Analysis	38
Identificazione SIS (Starting Impact Set)	38
Identificazione CIS (Candidate Impact Set)	39
Identificazione AIS (Actual Impact Set)	39
Identificazione FPIS (False Positive Impact Set)	40
Identificazione DIS (Discovered Impact Set)	40
Metriche Impact Analysis	40
<b>Testing Execution Report</b>	<b>41</b>
CR 1 - Cifratura password	41
Tests di unità e integrazione	41
Test di sistema	42
CR 2 - Entità squadra	42
Tests di unità e integrazione	42
Test di sistema	43
CR 3 - Aggiornamento ID Evento	44
Tests di unità e integrazione	44



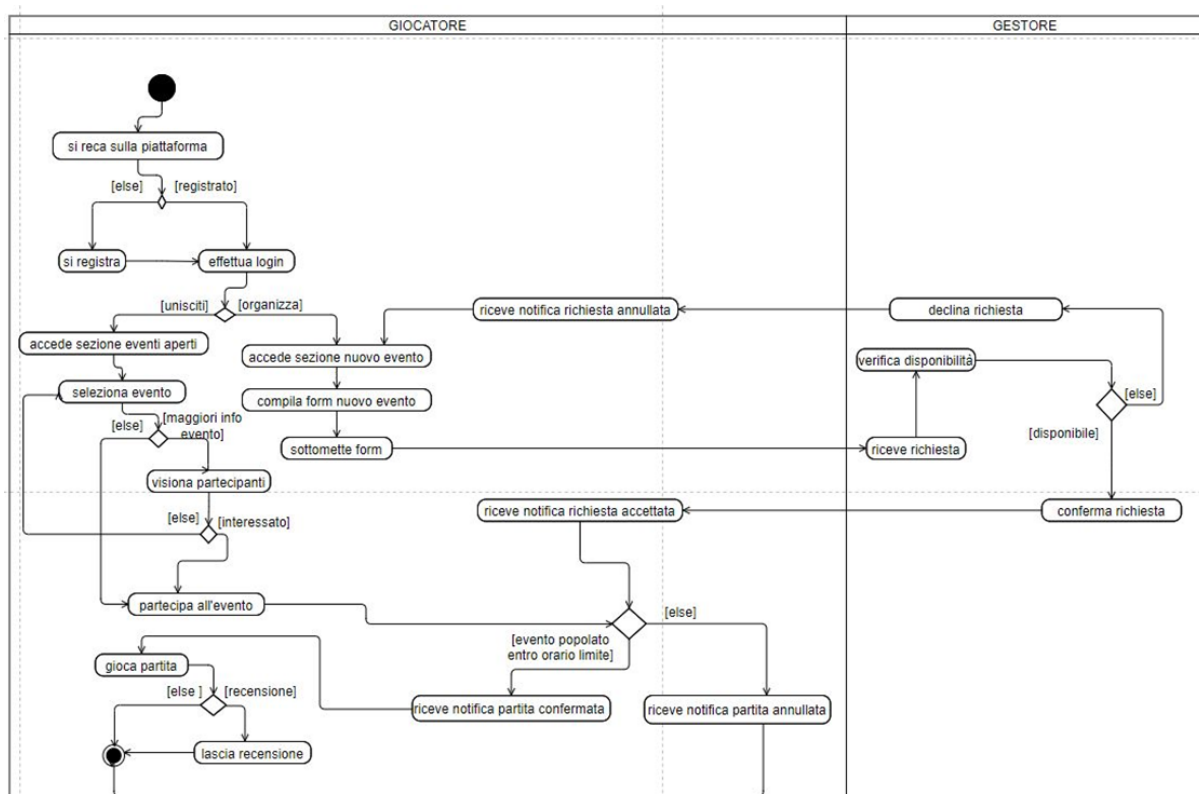
Test di sistema	45
CR 4 - Evento annullato, iniziato e finito	45
Tests di unità e integrazione	45
Test di sistema	45
<b>Panoramica sistema modificato</b>	<b>46</b>
Descrizione sistema modificato	46
Attori del sistema	46
Requisiti funzionali	46
Architettura del sistema	50
Gestione dati persistenti	51
Statechart evento	52

# Panoramica sistema attuale

## Descrizione sistema attuale

MyBomber intende fornire agli appassionati di calcio e ai gestori degli impianti sportivi un sistema per migliorare e rendere più semplice l'organizzazione e/o prenotazione di match, tra amici, conoscenti o sconosciuti, con lo scopo di creare una vera e propria rete che permetta di collegare tutti gli appassionati tra di loro. Gli utenti potranno scambiarsi dei feedback sulla prestazione tra di loro.

## Activity Diagram



## Attori del sistema

Nel sistema attuale si hanno i seguenti attori:

- **Giocatore**
  - Può partecipare agli eventi o crearne uno
  - Gestire gli eventi a cui partecipa o ha partecipato



- Può lasciare una recensione ad un altro giocatore
- **Gestore**
  - Accettare o rifiutare eventi presso la propria struttura
  - Controlla gli eventi da svolgere o che sono stati già svolti presso la propria struttura

## Requisiti funzionali

### Gestione Utente

*RF\_GU\_1* - Il sistema dovrà permettere ad un giocatore di effettuare la registrazione, **priorità alta**.

*RF\_GU\_2* - Il sistema dovrà permettere ad un gestore di effettuare la registrazione, **priorità alta**.

*RF\_GU\_3* - Il sistema dovrà permettere ad un gestore o un giocatore di effettuare il login, **priorità alta**.

*RF\_GU\_4* - Il sistema dovrà permettere ad un utente loggato di effettuare il logout, **priorità alta**.

*RF\_GU\_5* - Il sistema dovrà permettere ad un utente loggato di visualizzare l'area utente, **priorità alta**.

*RF\_GU\_6* - Il sistema dovrebbe permettere ad un utente loggato di modificare i dati personali attraverso l'area utente, **priorità bassa**.

*RF\_GU\_7* - Il sistema dovrebbe permettere ad un utente registrato di recuperare la propria password, **priorità bassa**.

### Gestione Evento

*RF\_GE\_1* - Il sistema dovrà permettere ad un giocatore di sottoporre una richiesta per un nuovo evento, **priorità alta**.

- *RF\_GE\_1\_US* (versione user story): Come giocatore voglio poter creare un nuovo evento per pianificare una nuova partita.

*RF\_GE\_2* - Il sistema dovrà permettere ad un giocatore di visualizzare gli eventi disponibili, **priorità alta**.

*RF\_GE\_3* - Il sistema dovrà permettere ad un giocatore di partecipare ad un evento, **priorità alta**.

- *RF\_GE\_3\_US* (versione user story): Come giocatore voglio poter partecipare ad un evento attraverso la selezione del medesimo evento nella lista degli eventi disponibili.



*RF\_GE\_4* - Il sistema dovrà permettere ad un giocatore di visualizzare i dettagli di un evento, **priorità bassa**.

*RF\_GE\_5* - Il sistema dovrà permettere ad un giocatore di visualizzare gli eventi in programma, **priorità alta**.

*RF\_GE\_6* - Il sistema dovrà permettere ad un giocatore di visualizzare gli eventi recenti, **priorità alta**.

*RF\_GE\_7* - Il sistema dovrà permettere ad un gestore di visualizzare gli eventi attivi della propria struttura, **priorità alta**.

- *RF\_GE\_7\_US* (versione user story): Come gestore voglio poter visualizzare gli eventi attivi nella mia struttura per coordinare al meglio la stessa.

*RF\_GE\_8* - Il sistema dovrebbe permettere ad un giocatore di annullare un evento da lui organizzato, **priorità bassa**.

*RF\_GE\_9* - Il sistema dovrà permettere ad un utente di visualizzare la cronologia eventi, **priorità alta**.

*RF\_GE\_10* - Il sistema dovrà permettere ad un gestore di visualizzare le richieste di nuovi eventi, **priorità alta**.

*RF\_GE\_11* - Il sistema dovrà permettere ad un gestore di accettare una richiesta evento tramite la pagina di gestione delle richieste, **priorità alta**.

- *RF\_GE\_11\_US* (versione user story): Come gestore voglio poter accettare la richiesta di un nuovo evento per gestire la struttura.

*RF\_GE\_12* - Il sistema dovrà permettere ad un gestore di declinare una richiesta evento tramite la pagina di gestione delle richieste, **priorità alta**.

*RF\_GE\_13* - Il sistema dovrà permettere al giocatore di visualizzare la sua valutazione, **priorità alta**.

*RF\_GE\_14* - Il sistema dovrebbe permettere ad un gestore di aggiungere un nuovo evento, **priorità bassa**.

*RF\_GE\_15* - Il sistema dovrebbe permettere ad un utente di ricercare un evento, **priorità bassa**.

*RF\_GE\_16* - Il sistema dovrebbe permettere ad un gestore di modificare un evento, **priorità bassa**.

*RF\_GE\_17* - Il sistema dovrebbe permettere ad un utente di visualizzare gli eventi disponibili, **priorità bassa**.



*RF\_GE\_18* - Il sistema dovrebbe inviare una notifica al giocatore se un evento a cui partecipa viene annullato, **priorità bassa**.

*RF\_GE\_19* - Il sistema dovrebbe inviare una notifica al giocatore se una richiesta per un nuovo evento viene accettata, **priorità bassa**.

*RF\_GE\_20* - Il sistema dovrebbe inviare una notifica al gestore se una richiesta per un nuovo evento viene sottomessa, **priorità bassa**.

*RF\_GE\_21* - Il sistema dovrebbe permettere ad un giocatore di abbandonare un evento, **priorità bassa**.

### **Gestione Recensione**

*RF\_GR\_1* - Il sistema dovrà permettere ad un giocatore di recensire un altro giocatore con cui ha giocato, **priorità alta**.

- *RF\_GR\_1\_US* (versione user story): Come giocatore voglio poter recensire un altro giocatore per tenere aggiornata la sua valutazione.

*RF\_GR\_2* - Il sistema dovrà permettere ad un giocatore di rimuovere una recensione da lui inserita, **priorità alta**.

*RF\_GR\_3* - Il sistema dovrebbe permettere ad un giocatore di modificare una recensione da lui inserita, **priorità bassa**.

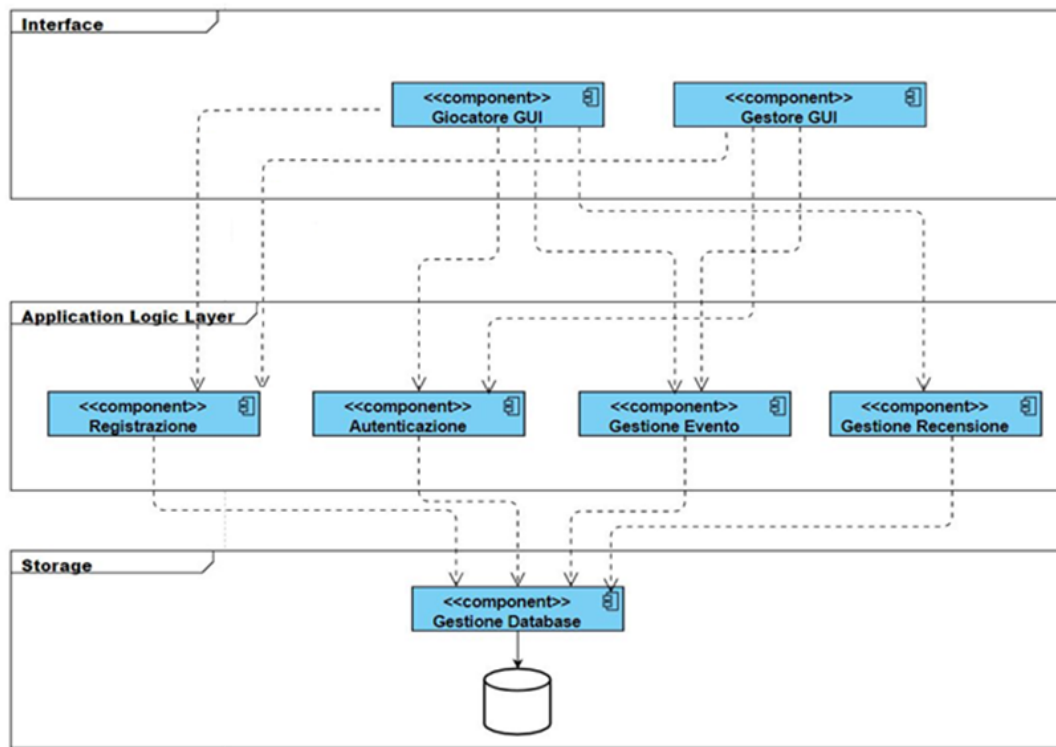
*RF\_GR\_4* - Il sistema dovrebbe permettere ad un giocatore di visualizzare la cronologia delle recensioni, **priorità bassa**.

## Architettura del sistema

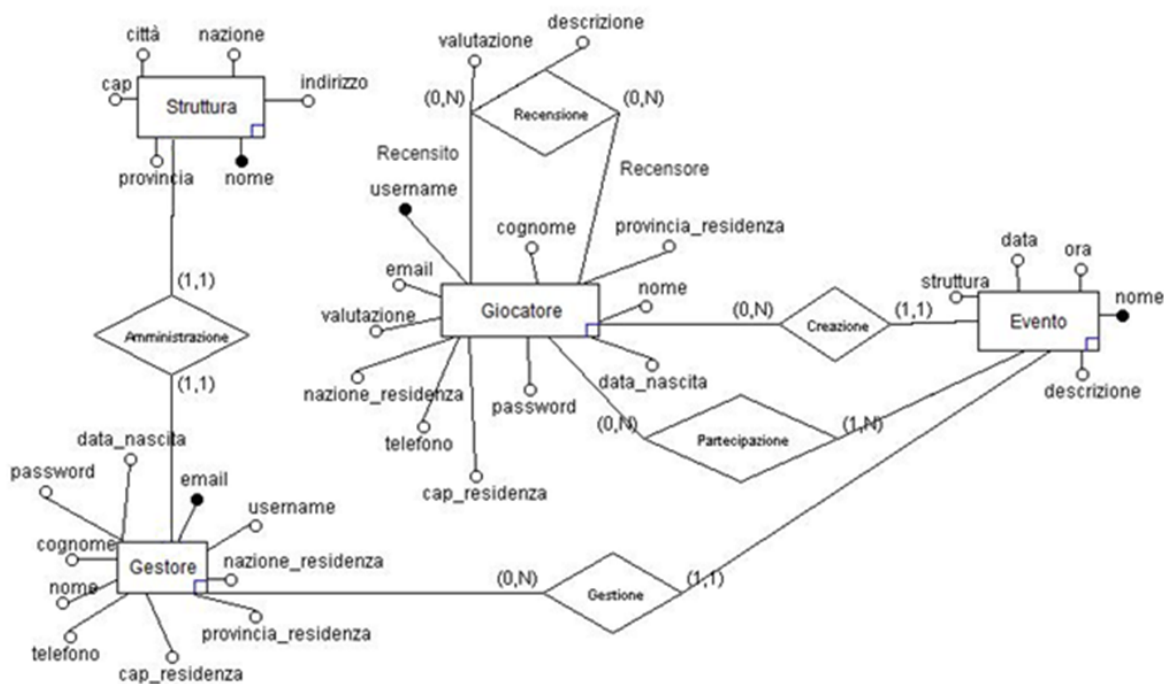
Il nostro sistema è basato su un'architettura Three-tier. Quindi, i sottosistemi sono organizzati in tre strati:

- L'interface layer (include tutti i boundary object che interfacciano con l'utente);
- L'application logic layer (include tutti gli oggetti relativi al controllo e alle entità che realizzano l'elaborazione, le regole di verifica e la notifica richieste dall'applicazione);
- Lo storage layer (effettua la memorizzazione, il recupero e l'interrogazione di oggetti persistenti).





## Gestione dati persistenti



Nome Entità		Giocatore		
Descrizione		Contiene i dati relativi ad un giocatore		
Nome campo		Tipo	Vincolo di chiave	Altri vincoli
E-mail		Varchar(30)	PRIMARY KEY	NOT NULL
Nome		Varchar(30)		NOT NULL

<b>Cognome</b>	Varchar(30)		NOT NULL
<b>Password</b>	Byte(32)		NOT NULL
<b>Telefono</b>	Varchar(10)		NOT NULL
<b>Valutazione</b>	float		NOT NULL

Nome Entità		Gestore	
Descrizione	Contiene i dati relativi ad un gestore		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
Nome	Varchar(30)		NOT NULL
Cognome	Varchar(30)		NOT NULL
E-mail	Varchar(30)	PRIMARY KEY	NOT NULL
Password	Byte(32)		NOT NULL

<b>Telefono</b>	Varchar(10)		NOT NULL
<b>Cap_residenza</b>	Int(10)		NOT NULL

Nome Entità		Evento	
Descrizione	Contiene i dati relativi ad un gestore		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
Nome	Varchar(30)	PRIMARY KEY	NOT NULL
Struttura	Varchar(30)		NOT NULL
Data	Date		NOT NULL
Ora	Time		NOT NULL
E-mail_utente	Varchar(30)	FOREIGN KEY	NOT NULL
E-mail_gestore	Varchar(30)	FOREIGN KEY	NOT NULL

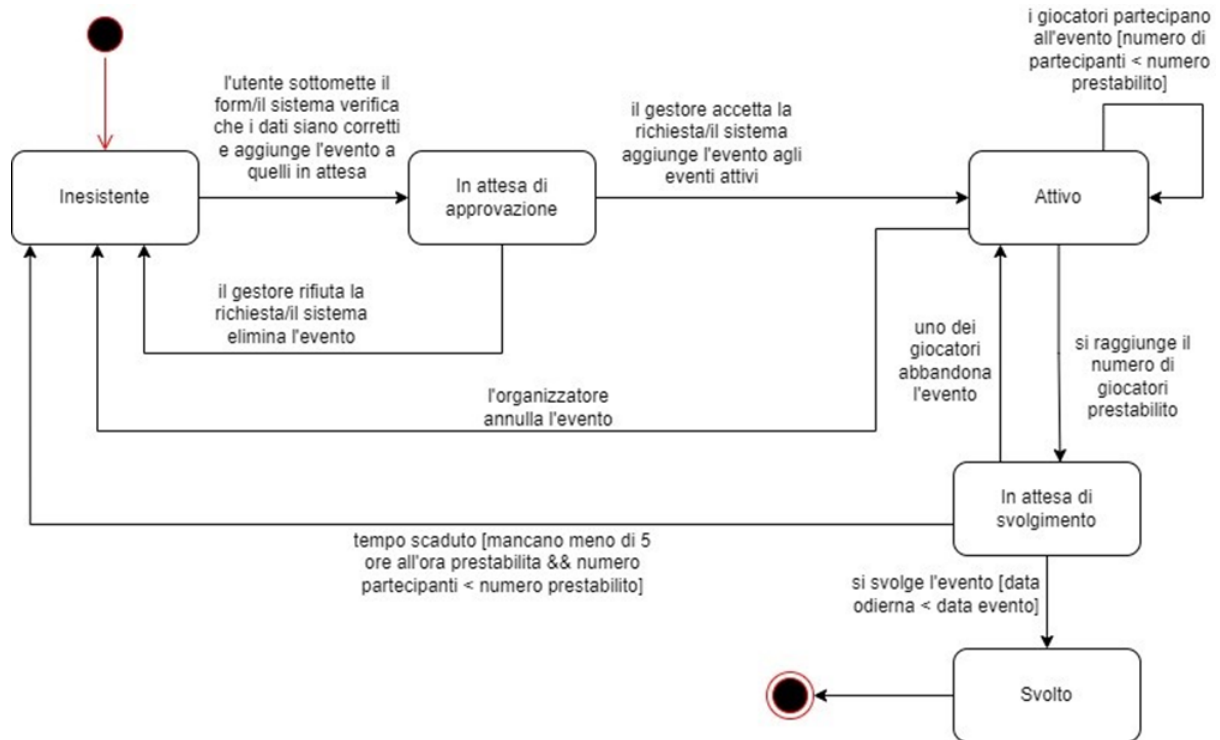
Nome Entità		Struttura	
Descrizione	Contiene i dati relativi ad un gestore		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
E-mail	Varchar(30)	PRIMARY KEY FOREIGN KEY	NOT NULL
Nome	Varchar(30)		NOT NULL
Indirizzo	Varchar(30)		NOT NULL
Città	Varchar(30)		NOT NULL
Nazione	Varchar(30)		NOT NULL
Provincia	Varchar(30)		NOT NULL

Nome Entità		Partecipazione
Descrizione	Contiene i dati relativi ad un gestore	

Nome campo	Tipo	Vincolo di chiave	Altri vincoli
E-mail	Varchar(30)	PRIMARY KEY FOREIGN KEY	NOT NULL
Nome_evento	Varchar(30)	PRIMARY KEY FOREIGN KEY	NOT NULL

Nome Entità	Recensione		
Descrizione	Contiene i dati relativi ad un gestore		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
E-mail	Varchar(30)	PRIMARY KEY FOREIGN KEY	NOT NULL
Nome_evento	Varchar(30)	PRIMARY KEY FOREIGN KEY	NOT NULL
Recensione	Int		NOT NULL

## Statechart evento



## Attività di manutenzione

Nella sezione seguente vengono illustrate le attività di manutenzione svolte. È stata svolta una specifica attività di impact analysis per ognuna delle change request.

Le change request sono state svolte in maniera sequenziale, quindi al termine dell'analisi e dell'implementazione della prima, si è proceduti ad analizzare la seconda e ad implementarne le modifiche, e così via. Per il candidate impact set ci siamo serviti della matrice di tracciabilità (all'interno della quale sono stati mappati, per ogni requisito funzionale implementato, le classi e i test case relativi), matrice delle dipendenze e class diagram (questi ultimi due generati da IntelliJ).

A supporto dell'impact analysis, sarà anche effettuata una ricerca nel codice basata attraverso tecniche di information retrieval integrate in IntelliJ, come find in files e find usages.

Per quanto riguarda la granularità delle componenti da considerare come un'unità di modifica, ogni metodo o variabile modificata/aggiunta/rimossa viene incrementato il contatore delle componenti di 1.

### Change request 1 - Cifratura password

#### Analisi richiesta modifica

La change request 1 prevede una manutenzione perfezionativa, dato che si vanno a migliorare la sicurezza e l'affidabilità del sistema passando da password in chiaro a password cifrate.

Nel sistema attuale quando un nuovo utente (sia gestore che giocatore) si registra viene salvata direttamente la password in chiaro e quindi quando un utente già registrato tenta di accedere al sistema, verrà fatto il confronto direttamente con la password in chiaro.

Per applicare questa modifica, quindi, si dovrà procedere tenendo in considerazione questi 3 step:

1. Quando un utente si registra, la password deve essere convertita in una stringa tramite algoritmi SHA-256 e successivamente salvata su db;
2. Quando un utente registrato effettua login, bisogna convertire la password tramite algoritmi SHA-256 e poi successivamente fare il confronto con quella sul db;
3. Si dovrebbe poi fare in modo che anche le password degli utenti già registrati al sistema siano convertite tramite uno script.

Per aggiungere questa modifica sarà quindi necessario analizzare le componenti che dovrebbero essere impattate. Sarà adottato un approccio top-down, quindi si parte dall'analisi della documentazione fino ad arrivare al codice.



## Impact Analysis

### Identificazione SIS (Starting Impact Set)

Componente	Tipo	Modifica	Azione
HashTool	Util	Aggiunta classe per gli algoritmi SHA-256 <ul style="list-style-type: none"> <li>Aggiunta del metodo hashSHA256() che prende in input una password in chiaro e restituisce il corrispondente cifrato.</li> </ul>	Aggiunta
Giocatore	Tabella	Modifica dell'attributo "password_giocatore", da varchar(30) a varchar(64) (sia per db di produzione che di testing)	Modifica
Gestore	Tabella	Modifica dell'attributo "password_gestore", da varchar(30) a varchar(64) (sia per db di produzione che di testing)	Modifica
GestoreDAO	DAO	<ul style="list-style-type: none"> <li>Passare nel doSave() la password cifrata piuttosto che quella in chiaro</li> <li>Passare nel doUpdate() la password cifrata piuttosto che quella in chiaro</li> <li>Creazione nuovo metodo doRetrieveByAuth()</li> </ul>	Modifica/Aggiunta
GiocatoreDAO	DAO	<ul style="list-style-type: none"> <li>Passare nel doSave() la password cifrata piuttosto che quella in chiaro</li> <li>Passare nel doUpdate() la password cifrata piuttosto che quella in chiaro</li> <li>Creazione nuovo metodo doRetrieveByAuth()</li> </ul>	Modifica/Aggiunta
LoginServlet	Servlet	<ul style="list-style-type: none"> <li>Utilizzo del metodo doRetrieveByAuth() (sia per gestore che giocatore) piuttosto che doRetrieveByKey()</li> <li>Eccezione NoSuchAlgorithmException</li> </ul>	Modifica
TestHashTool	Test	Test della classe HashTool	Aggiunta
TestLoginServlet	Test	Modifica del test in base alle modifiche apportate in LoginServlet	Modifica
TestIntegrazioneLogin	Test	Modifica del test in base alle modifiche apportate in LoginServlet	Modifica
TestGestoreDAO	Test	<ul style="list-style-type: none"> <li>Passare una password cifrata nel mock del GestoreBean</li> <li>throw exception da aggiungere dove si usa doupdate</li> <li>Aggiungere test per doRetrieveByAuth()</li> </ul>	Modifica/aggiunta
TestGiocatoreDAO	Test	<ul style="list-style-type: none"> <li>Passare una password cifrata nel mock del GiocatoreBean</li> <li>Aggiungere test per doRetrieveByAuth()</li> </ul>	Modifica/aggiunta
RecensioneServlet	Servlet	Aggiungere throw exceptions	Modifica

## Identificazione CIS (Candidate Impact Set)

Componente	Tipo	Modifica	Azione
HashTool	Classe/Utilità	Aggiunta classe per gli algoritmi SHA-256 <ul style="list-style-type: none"> <li>• Aggiunta del metodo hashSHA256() che prende in input una password in chiaro e restituisce il corrispondente cifrato.</li> </ul>	Aggiunta
Giocatore	Tabella	Modifica dell'attributo "password_giocatore", da varchar(30) a varchar(64) (sia per db di produzione che di testing)	Modifica
Gestore	Tabella	Modifica dell'attributo "password_gestore", da varchar(30) a varchar(64) (sia per db di produzione che di testing)	Modifica
GestoreDAO	DAO	<ul style="list-style-type: none"> <li>• Passare nel doSave() la password cifrata piuttosto che quella in chiaro</li> <li>• Passare nel doUpdate() la password cifrata piuttosto che quella in chiaro</li> <li>• Creazione nuovo metodo doRetrieveByAuth()</li> </ul>	Modifica/Aggiunta
GiocatoreDAO	DAO	<ul style="list-style-type: none"> <li>• Passare nel doSave() la password cifrata piuttosto che quella in chiaro</li> <li>• Passare nel doUpdate() la password cifrata piuttosto che quella in chiaro</li> <li>• Creazione nuovo metodo doRetrieveByAuth()</li> </ul>	Modifica/Aggiunta
LoginServlet	Servlet	<ul style="list-style-type: none"> <li>• Utilizzo del metodo doRetrieveByAuth() (sia per gestore che giocatore) piuttosto che doRetrieveByKey()</li> <li>• Eccezione NoSuchAlgorithmException</li> </ul>	Modifica
TestHashTool	Test	Test della classe HashTool	Aggiunta
TestLoginServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestIntegrazioneLogin	Test	Modifica del test in base alle modifiche apportate in LoginServlet	Modifica
TestAreaUtenzeServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestCreazioneEventoServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestCronologiaEventiServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestLogoutServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestRegistrazioneServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestRichiesteEventiServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica

TestGestoreDAO	Test	<ul style="list-style-type: none"> <li>Passare una password cifrata nel mock del GestoreBean</li> <li>throw exception da aggiungere dove si usa doupdate</li> <li>Aggiungere test per doRetrieveByAuth()</li> </ul>	Modifica/aggiunta
TestEventiRecentiServlet	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestParticipaEventiServlet	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestRecensioneServlet	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestGiocatoreDAO	Test	<ul style="list-style-type: none"> <li>Passare una password cifrata nel mock del GiocatoreBean</li> <li>Aggiungere test per doRetrieveByAuth()</li> </ul>	Modifica/aggiunta
TestIntegrazioneCreaEvento	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestIntegrazioneEventiRecenti	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestIntegrazioneParticipaEventi	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestIntegrazioneRecensione	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestIntegrazioneRegistrazione	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestIntegrazioneRichiesteEventi	Test	Passare una password cifrata nel mock del GestoreBean	Modifica
TestIntegrazioneCronologiaEventi	Test	Passare una password cifrata nel mock del GestoreBean	Modifica
RecensioneServlet	Servlet	Aggiungere throw exceptions	Modifica

#### Identificazione AIS (Actual Impact Set)

Componente	Tipo	Modifica	Azione
HashTool	Classe/Utilità	Aggiunta classe per gli algoritmi SHA-256 <ul style="list-style-type: none"> <li>Aggiunta del metodo hashSHA256() che prende in input una password in chiaro e restituisce il</li> </ul>	Aggiunta



		corrispondente cifrato.	
Giocatore	Tabella	Modifica dell'attributo "password_giocatore", da varchar(30) a varchar(64) (sia per db di produzione che di testing)	Modifica
Gestore	Tabella	Modifica dell'attributo "password_gestore", da varchar(30) a varchar(64) (sia per db di produzione che di testing)	Modifica
GestoreDAO	DAO	<ul style="list-style-type: none"><li>Passare nel doSave() la password cifrata piuttosto che quella in chiaro</li><li>Passare nel doUpdate() la password cifrata piuttosto che quella in chiaro</li><li>Creazione nuovo metodo doRetrieveByAuth()</li></ul>	Modifica/Aggiunta
GiocatoreDAO	DAO	<ul style="list-style-type: none"><li>Passare nel doSave() la password cifrata piuttosto che quella in chiaro</li><li>Passare nel doUpdate() la password cifrata piuttosto che quella in chiaro</li><li>Creazione nuovo metodo doRetrieveByAuth()</li></ul>	Modifica/Aggiunta
LoginServlet	Servlet	<ul style="list-style-type: none"><li>Utilizzo del metodo doRetrieveByAuth() (sia per gestore che giocatore) piuttosto che doRetrieveByKey()</li><li>Eccezione NoSuchAlgorithmException</li></ul>	Modifica
TestHashTool	Test	Test della classe HashTool	Aggiunta
TestLoginServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestAreaUtenteServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestCreazioneEventoServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestCronologiaEventiServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestLogoutServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestRegistrazioneServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestRichiesteEventiServlet	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestGestoreDAO	Test	<ul style="list-style-type: none"><li>Passare una password cifrata nel mock del GestoreBean</li><li>throw exception da aggiungere dove si usa doUpdate</li><li>Aggiungere test per doRetrieveByAuth()</li></ul>	Modifica/aggiunta
TestEventiRecentiServlet	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestPartecipa	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica

EventiServlet			
TestRecensioneServlet	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestGiocatoreDAO	Test	<ul style="list-style-type: none"> <li>Passare una password cifrata nel mock del GiocatoreBean</li> <li>Aggiungere test per doRetrieveByAuth()</li> </ul>	Modifica/aggiunta
TestIntegrazioneCreaEvento	Test	Passare una password cifrata nel mock del GestoreBean e GiocatoreBean	Modifica
TestIntegrazioneEventiRece	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestIntegrazionePartecipaEventi	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestIntegrazioneRecensione	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestIntegrazioneRegistrazione	Test	Passare una password cifrata nel mock del GiocatoreBean	Modifica
TestIntegrazioneRichiesteEventi	Test	Passare una password cifrata nel mock del GestoreBean	Modifica
TestIntegrazioneCronologiaEventi	Test	Passare una password cifrata nel mock del GestoreBean	Modifica
RecensioneServlet	Servlet	Aggiungere throw exceptions	Modifica

#### Identificazione FPIS (False Positive Impact Set)

Componente	Tipo	Modifica	Azione
TestIntegrazioneLogin	Test	<ul style="list-style-type: none"> <li>Modifica del test in base alle modifiche apportate in LoginServlet</li> </ul>	Modifica

#### Identificazione DIS (Discovered Impact Set)

Il discovered impact set è risultato vuoto, in quanto non sono state scoperte modifiche da effettuare durante l'attività di modifica del sistema.

#### Metriche Impact Analysis

Numero di componenti nel SIS: 20

Numero di componenti nel CIS: 36



Numero di componenti nel AIS: 35

Numero di componenti nel FPIS: 1

Numero di componenti nel DIS: 0

**Recall:**  $|CIS \cap AIS| \div |AIS| = 35 \div 35 = 1$ . Questo è ovvio perché il DIS è 0.

**Precision:**  $|CIS \cap AIS| \div |CIS| = 35 \div 36 = 0,97$ .

## Change request 2 - Entità squadra

### Analisi richiesta modifica

La CR2 prevede un'evoluzione del sistema, aggiungendo una nuova entità al sistema: squadra. Questa modifica comporta l'aggiunta dell'entità "squadra" nel db, la creazione dei model corrispondenti e l'aggiunta di nuovi requisiti funzionali:

#### Gestione Squadra

*RF\_GS\_01* Il sistema dovrà permettere ad un giocatore di poter creare una squadra di calcio, **priorità alta**.

*RF\_GS\_02* Il sistema dovrà permettere ad un capitano di poter eliminare la squadra che ha creato, **priorità bassa**.

*RF\_GS\_03* Il sistema dovrà permettere ad un giocatore di poter visionare le informazioni di una squadra di calcio, **priorità alta**.

*RF\_GS\_04* Il sistema dovrà permettere ad un giocatore di aggiungersi ad una squadra, **priorità alta**.

*RF\_GS\_05* Il sistema dovrà permettere ad un giocatore di abbandonare la sua squadra, **priorità alta**.

*RF\_GS\_06* Il sistema dovrà permettere al giocatore capitano della squadra di accettare o declinare la richiesta di un giocatore nella sua squadra, **priorità bassa**.

*RF\_GS\_07* Il sistema dovrà permettere al giocatore capitano della squadra di invitare giocatori nella sua squadra, **priorità bassa**.

*RF\_GS\_08* Il sistema dovrà permettere al capitano di una squadra di creare degli eventi di tipo squadra, **priorità alta**.

*RF\_GS\_09* Il sistema dovrà permettere al capitano di partecipare a degli eventi di tipo squadra, **priorità alta**.

*RF\_GS\_10* Il sistema dovrà permettere ai giocatori di una squadra di visionare gli eventi squadra a cui dovranno partecipare, **priorità alta**.

*RF\_GS\_11* Il sistema dovrà permettere a un capitano di poter recensire un'altra squadra, **priorità bassa**.

*RF\_GS\_12* Il sistema dovrà permettere al capitano di poter eliminare la recensione fatta ad un'altra squadra, **priorità bassa**.

*RF\_GS\_13* Il sistema dovrà permettere ad un giocatore di poter visionare le recensioni di una squadra, **priorità bassa**.

*RF\_GS\_14* Il sistema dovrà permettere ai giocatori di visionare gli ultimi match di una squadra, **priorità bassa**.

*RF\_GS\_15* Il sistema dovrà permettere ai giocatori di visionare tutte le squadre presenti sul sistema, **priorità alta**.

*RF\_GS\_16* Il sistema dovrà permettere al giocatore capitano di poter modificare le informazioni della propria squadra, **priorità bassa**.

## Impact Analysis

### Identificazione SIS (Starting Impact Set)

Componente	Tipo	Modifica	Azione
Squadra	Table	Aggiunta entità squadra nel db.	Aggiunta
SquadraBean	Bean	Aggiunta del bean della squadra.	Aggiunta
SquadraDAO	DAO	Classe per interrogare l'entità squadra.	Aggiunta
AreaUtenteServlet	Servlet	Aggiunta delle info da passare alla view AreaUtente.jsp	Modifica
CreaSquadra.jsp	View	View in cui è possibile aggiungere le info per creare una squadra.	Aggiunta
CreaSquadraServlet	Servlet	Servlet per salvare le info della squadra di calcio	Aggiunta
creasquadra-validation.js	Script	Validazione dei campi passati durante la creazione della squadra.	Aggiunta
SquadraPage.jsp	View	View che permette di visualizzare le informazioni specifiche della squadra.	Aggiunta
MostraSquadraServlet	Servlet	Servlet che ritorna i dati da mostrare su SquadraPage.	Aggiunta
UniscitiSquad	Servlet	Servlet che aggiunge un giocatore in una squadra.	Aggiunta

raServlet			
AbbandonaSquadraServlet	Servlet	Servlet che rimuove un giocatore da una squadra.	Aggiunta
Evento	Table	Aggiunta dell'attributo "tipologia" per distinguere i due tipi di eventi.	Modifica
EventoBean	Bean	Aggiunta dell'attributo "tipologia".	Modifica
CreaEventoServlet	Servlet	Gestione della logica per stabilire se l'evento è di tipo squadra o normale.	Modifica
TestCreaEventoServlet	Test	Aggiunta del metodo creaEventoSquadra().	Modifica
EventoDAO	DAO	<ul style="list-style-type: none"> <li>doRetrieveEventiGestore(), aggiungere il nuovo attributo "tipologia".</li> <li>doRetrieveEventiRecenti(), aggiungere il nuovo attributo "tipologia".</li> <li>doRetrieveEventi(), aggiungere il nuovo attributo "tipologia".</li> <li>doRetrieveByKey(), aggiungere il nuovo attributo "tipologia".</li> <li>aggiungere metodo doRetrieveEventiRecentiSquadra()</li> </ul>	Modifica
PartecipaSquadra	Table	Aggiunta della tabella che contiene le squadre associate all'evento.	Aggiunta
PartecipaEventiServlet	Servlet	Aggiunta della logica che aggiunge una squadra ad un evento.	Modifica
PartecipaEventi.jsp	View	Modifica del pulsante in "Partecipa come squadra" nel caso sia un evento squadra.	Modifica
PartecipazioneSquadraBean	Bean	Bean di partecipaSquadra.	Aggiunta
PartecipazioneSquadraDAO	DAO	DAO di partecipaSquadra.	Aggiunta
TestIntegrazionePartecipaEventi	Test	<ul style="list-style-type: none"> <li>Aggiunta metodo partecipaEventoSquadra()</li> <li>AggiuntaMetodo partecipaEventoCompletoSquadra()</li> </ul>	Modifica
header.jsp	View	Aggiunta delle voci inerenti alla squadra.	Modifica
Squadre.jsp	View	Aggiunta della pagina contenente tutte le squadre.	Aggiunta
MostraSquadreServlet	Servlet	Restituisce tutte le squadre presenti sul sistema.	Aggiunta
TestMostraSquadreServlet	Test	Test MostraSquadreServlet	Aggiunta
TestIntegrazioneMostraSquadre	Test	Test integrazione MostraSquadreServlet	Aggiunta





adre			
TestSquadraDAO	Test	Test SquadraDAO	Aggiunta
TestMostraSquadraServlet	Test	Test MostraSquadraServlet	Aggiunta
TestIntegrazioneMostraSquadra	Test	Test MostraSquadraServlet	Aggiunta
TestAbbandonaSquadraServlet	Test	Test AbbandonaSquadraServlet	Aggiunta
TestIntegrazioneAbbandonaSquadra	Test	Test AbbandonaSquadraServlet	Aggiunta
TestPartecipazioneSquadraDAO	Test	Test PartecipazioneSquadraDAO	Aggiunta
giocatore	table	Aggiunta attr "id_squadra".	Modifica
GiocatoreBean	Bean	Aggiunta attr "idSquadra".	Modifica
LoginServlet	Servlet	<ul style="list-style-type: none"> <li>• Aggiunta in sessione della squadra del giocatore</li> <li>• Aggiunta in sessione dei loghi</li> </ul>	Modifica
TestLoginServlet	Test	<ul style="list-style-type: none"> <li>• test giocatore con squadra</li> </ul>	Modifica
TestIntegrazioneLogin	Test	<ul style="list-style-type: none"> <li>• test giocatore con squadra</li> </ul>	Modifica
logo_squadra	table	tabella contenente i loghi delle squadre	Aggiunta
LogoSquadraBean	Bean	Bean di logo_squadra	Aggiunta
LogoSquadraDAO	DAO	DAO di logo_squadra	Aggiunta
TestLogoSquadraDAO	Test	test del dao	Aggiunta
GiocatoreDAO	DAO	Aggiunte del metodo doRetrieveBySquadra()	Modifica
TestGiocatoreDAO	Test	Test del metodo doRetrieveBySquadra()	Modifica
EliminaSquadraServlet	Servlet	Eliminazione squadra	Aggiunta
TestEliminaSquadraServlet	Test	Test della servlet elimina squadra	Aggiunta

TestIntegrazioneEliminaSquadra	Test	Test della servlet elimina squadra	Aggiunta
TestIntegrazioneCreaEvento	Test	<ul style="list-style-type: none"> <li>Modifica del test per il controllo della tipologia dell'evento</li> <li>Aggiunta test per tipologia squadra</li> </ul>	Modifica
TestIntegrazioneRichiesteEventi	Test	<ul style="list-style-type: none"> <li>Aggiunta tipologia ai mock evento</li> <li>Aggiunta test accettaRichiesteSquadra()</li> </ul>	Modifica
TestCreaSquadraServlet	Test	Test di CreaSquadraServlet	Aggiunta
TestIntegrazioneCreaSquadra	Test	Test di CreaSquadraServlet	Aggiunta

#### Identificazione CIS (Candidate Impact Set)

Componente	Tipo	Modifica	Azione
Squadra	Table	Aggiunta entità squadra nel db.	Aggiunta
SquadraBean	Bean	Aggiunta del bean della squadra.	Aggiunta
SquadraDAO	DAO	Classe per interrogare l'entità squadra.	Aggiunta
AreaUtente.jsp	View	<ul style="list-style-type: none"> <li>Aggiunta campo "squadra" tra le info</li> <li>Aggiunta sezione per creare una squadra</li> </ul>	Modifica
AreaUtenteServlet	Servlet	Aggiunta delle info da passare alla view AreaUtente.jsp	Modifica
TestAreaUtenteServlet	Test	Modifica al metodo mostraGiocatore().	Modifica
CreaSquadra.jsp	View	View in cui è possibile aggiungere le info per creare una squadra.	Aggiunta
CreaSquadraServlet	Servlet	Servlet per salvare le info della squadra di calcio	Aggiunta
creasquadra-validation.js	Script	Validazione dei campi passati durante la creazione della squadra.	Aggiunta
SquadraPage.jsp	View	View che permette di visualizzare le informazioni specifiche della squadra.	Aggiunta
MostraSquadraServlet	Servlet	Servlet che ritorna i dati da mostrare su SquadraPage.	Aggiunta
UniscitiSquadraServlet	Servlet	Servlet che aggiunge un giocatore in una squadra.	Aggiunta
AbbandonaSq	Servlet	Servlet che rimuove un giocatore da una squadra.	Aggiunta



uadraServlet			
Evento	Table	Aggiunta dell'attributo "tipologia" per distinguere i due tipi di eventi.	Modifica
EventoBean	Bean	Aggiunta dell'attributo "tipologia".	Modifica
CreaEventoServlet	Servlet	Gestione della logica per stabilire se l'evento è di tipo squadra o normale.	Modifica
CreaEvento.jsp	View	Aggiunta di uno switch per selezionare il tipo di evento.	Modifica
TestCreaEventoServlet	Test	<ul style="list-style-type: none"><li>• Aggiunta del metodo creaEventoSquadra().</li><li>• <b>Modifica dell'EventoBean per aggiungere l'attributo "tipologia".</b></li></ul>	Modifica
TestEventiRecentiServlet	Test	<ul style="list-style-type: none"><li>• Modifica dell'EventoBean per aggiungere l'attributo "tipologia".</li><li>• Aggiunta nuovo test per testare le logiche squadra</li></ul>	Modifica
TestPartecipaEventiServlet	Test	<ul style="list-style-type: none"><li>• Modifica dell'EventoBean per aggiungere l'attributo "tipologia".</li><li>• Aggiunta metodo partecipaEventoSquadra()</li><li>• Aggiunta Metodo partecipaEventoCompletoSquadra()</li></ul>	Modifica
TestRichiesteEventiServlet	Test	<ul style="list-style-type: none"><li>• Modifica dell'EventoBean per aggiungere l'attributo "tipologia".</li><li>• Aggiunta del metodo accettaRichiesteSquadra()</li></ul>	Modifica
EventoDAO	DAO	<ul style="list-style-type: none"><li>• doRetrieveEventiGestore(), aggiungere il nuovo attributo "tipologia".</li><li>• doRetrieveEventiRecenti(), aggiungere il nuovo attributo "tipologia".</li><li>• doRetrieveEventi(), aggiungere il nuovo attributo "tipologia".</li><li>• doRetrieveByKey(), aggiungere il nuovo attributo "tipologia".</li><li>• aggiungere metodo doRetrieveEventiRecentiSquadra()</li></ul>	Modifica
TestEventoDAO	Test	<ul style="list-style-type: none"><li>• Modificare gli EventoBean aggiungendo il nuovo attributo "tipologia".</li><li>• Aggiungere test per nuovo metodo doRetrieveEventiRecentiSquadra()</li></ul>	Modifica
PartecipaSquadra	Table	Aggiunta della tabella che contiene le squadre associate all'evento.	Aggiunta
PartecipaEventiServlet	Servlet	Aggiunta della logica che aggiunge una squadra ad un evento.	Modifica
PartecipaEventi.jsp	View	Modifica del pulsante in "Partecipa come squadra" nel caso sia un evento squadra.	Modifica
PartecipazioneSquadraBean	Bean	Bean di partecipaSquadra.	Aggiunta



PartecipazioneSquadraDAO	DAO	DAO di partecipaSquadra.	Aggiunta
TestIntegrazionePartecipazioneEventi	Test	<ul style="list-style-type: none"><li>• Aggiunta metodo partecipEventoSquadra()</li><li>• AggiuntaMetodo partecipEventoCompletoSquadra()</li></ul>	Modifica
header.jsp	View	Aggiunta delle voci inerenti alla squadra.	Modifica
Squadre.jsp	View	Aggiunta della pagina contenente tutte le squadre.	Aggiunta
MostraSquadreServlet	Servlet	Restituisce tutte le squadre presenti sul sistema.	Aggiunta
TestMostraSquadreServlet	Test	Test MostraSquadreServlet	Aggiunta
TestIntegrazioneMostraSquadre	Test	Test integrazione MostraSquadreServlet	Aggiunta
TestSquadraDAO	Test	Test SquadraDAO	Aggiunta
TestMostraSquadraServlet	Test	Test MostraSquadraServlet	Aggiunta
TestIntegrazioneMostraSquadra	Test	Test MostraSquadraServlet	Aggiunta
TestAbbandonaSquadraServlet	Test	Test AbbandonaSquadraServlet	Aggiunta
TestIntegrazioneAbbandonaSquadra	Test	Test AbbandonaSquadraServlet	Aggiunta
TestPartecipazioneSquadraDAO	Test	Test PartecipazioneSquadraDAO	Aggiunta
giocatore	table	Aggiunta attr "id_squadra".	Modifica
GiocatoreBean	Bean	Aggiunta attr "idSquadra".	Modifica
LoginServlet	Servlet	<ul style="list-style-type: none"><li>• Aggiunta in sessione della squadra del giocatore</li><li>• Aggiunta in sessione dei loghi</li></ul>	Modifica
TestLoginServlet	Test	<ul style="list-style-type: none"><li>• test giocatore con squadra</li></ul>	Modifica
TestIntegrazioneLogin	Test	<ul style="list-style-type: none"><li>• test giocatore con squadra</li></ul>	Modifica
logo_squadra	table	tabella contenente i loghi delle squadre	Aggiunta

LogoSquadra Bean	Bean	Bean di logo_squadra	Aggiunta
LogoSquadra DAO	DAO	DAO di logo_squadra	Aggiunta
TestLogoSquadraDAO	Test	test del dao	Aggiunta
GiocatoreDAO	DAO	Aggiunte del metodo doRetrieveBySquadra()	Modifica
TestGiocatoreDAO	Test	Test del metodo doRetrieveBySquadra()	Modifica
EliminaSquadraServlet	Servlet	Eliminazione squadra	Aggiunta
TestEliminaSquadraServlet	Test	Test della servlet elimina squadra	Aggiunta
TestIntegrazioneEliminaSquadra	Test	Test della servlet elimina squadra	Aggiunta
TestIntegrazioneCreaEvento	Test	<ul style="list-style-type: none"> <li>Modifica del test per il controllo della tipologia dell'evento</li> <li>Aggiunta test per tipologia squadra</li> </ul>	Modifica
TestIntegrazioneRichiesteEventi	Test	<ul style="list-style-type: none"> <li>Aggiunta tipologia ai mock evento</li> <li>Aggiunta test accettaRichiesteSquadra()</li> </ul>	Modifica
TestCreaSquadraServlet	Test	Test di CreaSquadraServlet	Aggiunta
TestIntegrazioneCreaSquadra	Test	Test di CreaSquadraServlet	Aggiunta

#### Identificazione AIS (Actual Impact Set)

Componente	Tipo	Modifica	Azione
Squadra	Table	Aggiunta entità squadra nel db.	Aggiunta
SquadraBean	Bean	Aggiunta del bean della squadra.	Aggiunta
SquadraDAO	DAO	Classe per interrogare l'entità squadra.	Aggiunta
AreaUtente.jsp	View	<ul style="list-style-type: none"> <li>Aggiunta campo "squadra" tra le info</li> <li>Aggiunta sezione per creare una squadra</li> </ul>	Modifica
AreaUtenteServlet	Servlet	Aggiunta delle info da passare alla view AreaUtente.jsp	Modifica
TestAreaUtenteServlet	Test	Modifica al metodo mostraGiocatore().	Modifica



CreaSquadra.jsp	View	View in cui è possibile aggiungere le info per creare una squadra.	Aggiunta
CreaSquadraServlet	Servlet	Servlet per salvare le info della squadra di calcio	Aggiunta
creasquadra-validation.js	Script	Validazione dei campi passati durante la creazione della squadra.	Aggiunta
SquadraPage.jsp	View	View che permette di visualizzare le informazioni specifiche della squadra.	Aggiunta
MostraSquadraServlet	Servlet	Servlet che ritorna i dati da mostrare su SquadraPage.	Aggiunta
UniscitiSquadraServlet	Servlet	Servlet che aggiunge un giocatore in una squadra.	Aggiunta
AbbandonaSquadraServlet	Servlet	Servlet che rimuove un giocatore da una squadra.	Aggiunta
Evento	Table	Aggiunta dell'attributo "tipologia" per distinguere i due tipi di eventi.	Modifica
EventoBean	Bean	Aggiunta dell'attributo "tipologia".	Modifica
CreaEventoServlet	Servlet	Gestione della logica per stabilire se l'evento è di tipo squadra o normale.	Modifica
CreaEvento.jsp	View	Aggiunta di uno switch per selezionare il tipo di evento.	Modifica
TestCreaEventoServlet	Test	<ul style="list-style-type: none"><li>• Aggiunta del metodo creaEventoSquadra().</li><li>• Modifica dell'EventoBean per aggiungere l'attributo "tipologia".</li></ul>	Modifica
TestEventiRecentiServlet	Test	<ul style="list-style-type: none"><li>• Modifica dell'EventoBean per aggiungere l'attributo "tipologia".</li><li>• Aggiunta nuovo test per testare le logiche squadra</li></ul>	Modifica
TestPartecipaEventiServlet	Test	<ul style="list-style-type: none"><li>• Modifica dell'EventoBean per aggiungere l'attributo "tipologia".</li><li>• Aggiunta metodo partecipaEventoSquadra()</li><li>• AggiuntaMetodo partecipaEventoCompletoSquadra()</li></ul>	Modifica
TestRichiesteEventiServlet	Test	<ul style="list-style-type: none"><li>• Modifica dell'EventoBean per aggiungere l'attributo "tipologia".</li><li>• Aggiunta del metodo accettaRichiesteSquadra()</li></ul>	Modifica
EventoDAO	DAO	<ul style="list-style-type: none"><li>• doRetrieveEventiGestore(), aggiungere il nuovo attributo "tipologia".</li><li>• doRetrieveEventiRecenti(), aggiungere il nuovo attributo "tipologia".</li><li>• doRetrieveEventi(), aggiungere il nuovo attributo "tipologia".</li><li>• doRetrieveByKey(), aggiungere il nuovo attributo "tipologia".</li><li>• aggiungere metodo</li></ul>	Modifica

		doRetrieveEventiRecentiSquadra()	
TestEventoDAO	Test	<ul style="list-style-type: none"> <li>• Modificare gli EventoBean aggiungendo il nuovo attributo "tipologia".</li> <li>• Aggiungere test per nuovo metodo doRetrieveEventiRecentiSquadra()</li> </ul>	Modifica
PartecipaSquadra	Table	Aggiunta della tabella che contiene le squadre associate all'evento.	Aggiunta
PartecipaEventiServlet	Servlet	Aggiunta della logica che aggiunge una squadra ad un evento.	Modifica
PartecipaEventi.jsp	View	Modifica del pulsante in "Partecipa come squadra" nel caso sia un evento squadra.	Modifica
PartecipazioneSquadraBean	Bean	Bean di partecipaSquadra.	Aggiunta
PartecipazioneSquadraDAO	DAO	DAO di partecipaSquadra.	Aggiunta
TestIntegrazionePartecipaEventi	Test	<ul style="list-style-type: none"> <li>• Aggiunta metodo partecipaEventoSquadra()</li> <li>• Aggiunta Metodo partecipaEventoCompletoSquadra()</li> </ul>	Modifica
header.jsp	View	Aggiunta delle voci inerenti alla squadra.	Modifica
Squadre.jsp	View	Aggiunta della pagina contenente tutte le squadre.	Aggiunta
MostraSquadreServlet	Servlet	Restituisce tutte le squadre presenti sul sistema.	Aggiunta
TestMostraSquadreServlet	Test	Test MostraSquadreServlet	Aggiunta
TestIntegrazioneMostraSquadre	Test	Test integrazione MostraSquadreServlet	Aggiunta
TestSquadraDAO	Test	Test SquadraDAO	Aggiunta
TestMostraSquadraServlet	Test	Test MostraSquadraServlet	Aggiunta
TestIntegrazioneMostraSquadra	Test	Test MostraSquadraServlet	Aggiunta
TestAbbandonaSquadraServlet	Test	Test AbbandonaSquadraServlet	Aggiunta
TestIntegrazioneAbbandonaSquadra	Test	Test AbbandonaSquadraServlet	Aggiunta

TestPartecipazioneSquadraDAO	Test	Test PartecipazioneSquadraDAO	Aggiunta
giocatore	table	Aggiunta attr "id_squadra".	Modifica
GiocatoreBean	Bean	Aggiunta attr "idSquadra".	Modifica
LoginServlet	Servlet	<ul style="list-style-type: none"> <li>• Aggiunta in sessione della squadra del giocatore</li> <li>• Aggiunta in sessione dei loghi</li> </ul>	Modifica
TestLoginServlet	Test	<ul style="list-style-type: none"> <li>• test giocatore con squadra</li> </ul>	Modifica
TestIntegrazioneLogin	Test	<ul style="list-style-type: none"> <li>• test giocatore con squadra</li> </ul>	Modifica
logo_squadra	table	tabella contenente i loghi delle squadre	Aggiunta
LogoSquadraBean	Bean	Bean di logo_squadra	Aggiunta
LogoSquadraDAO	DAO	DAO di logo_squadra	Aggiunta
TestLogoSquadraDAO	Test	test del dao	Aggiunta
GiocatoreDAO	DAO	Aggiunte del metodo doRetrieveBySquadra()	Modifica
TestGiocatoreDAO	Test	Test del metodo doRetrieveBySquadra()	Modifica
EliminaSquadraServlet	Servlet	Eliminazione squadra	Aggiunta
TestEliminaSquadraServlet	Test	Test della servlet elimina squadra	Aggiunta
TestIntegrazioneEliminaSquadra	Test	Test della servlet elimina squadra	Aggiunta
TestIntegrazioneCreaEvento	Test	<ul style="list-style-type: none"> <li>• Modifica del test per il controllo della tipologia dell'evento</li> <li>• Aggiunta test per tipologia squadra</li> </ul>	Modifica
TestIntegrazioneRichiesteEventi	Test	<ul style="list-style-type: none"> <li>• Aggiunta tipologia ai mock evento</li> <li>• Aggiunta test accettaRichiesteSquadra()</li> </ul>	Modifica
TestCreaSquadraServlet	Test	Test di CreaSquadraServlet	Aggiunta
TestIntegrazioneCreaSquadra	Test	Test di CreaSquadraServlet	Aggiunta





TestEventiRecentiServlet	Test	Aggiunta nuovo test per testare le logiche squadra	Modifica
TestRichiesteEventiServlet	Test	Aggiunta del metodo accettaRichiesteSquadra()	Modifica
EventoDAO	DAO	Aggiungere metodo doRetrieveEventiRecentiSquadra()	Modifica
TestEventoDAO	Test	Aggiungere test per nuovo metodo doRetrieveEventiRecentiSquadra()	Modifica
logo_squadra	table	tabella contenente i loghi delle squadre	Aggiunta
LogoSquadraBean	Bean	Bean di logo_squadra	Aggiunta
LogoSquadraDAO	DAO	DAO di logo_squadra	Aggiunta
TestLogoSquadraDAO	Test	test del dao	Aggiunta

#### Identificazione FPIS (False Positive Impact Set)

Non sono stati introdotti falsi positivi.

#### Identificazione DIS (Discovered Impact Set)

Componente	Tipo	Modifica	Azione
TestEventiRecentiServlet	Test	Aggiunta nuovo test per testare le logiche squadra	Modifica
TestRichiesteEventiServlet	Test	Aggiunta del metodo accettaRichiesteSquadra()	Modifica
EventoDAO	DAO	Aggiungere metodo doRetrieveEventiRecentiSquadra()	Modifica
TestEventoDAO	Test	Aggiungere test per nuovo metodo doRetrieveEventiRecentiSquadra()	Modifica
logo_squadra	table	tabella contenente i loghi delle squadre	Aggiunta
LogoSquadraBean	Bean	Bean di logo_squadra	Aggiunta
LogoSquadraDAO	DAO	DAO di logo_squadra	Aggiunta
TestLogoSquadraDAO	Test	test del dao	Aggiunta

#### Metriche Impact Analysis

Numero di componenti nel SIS: 59



Numero di componenti nel CIS: 73

Numero di componenti nel AIS: 81

Numero di componenti nel FPIS: 0

Numero di componenti nel DIS: 8

**Recall:**  $|CIS \cap AIS| \div |AIS| = 73 \div 81 = 0,9$ .

**Precision:**  $|CIS \cap AIS| \div |CIS| = 73 \div 73 = 1$ . Questo è ovvio perché il FPIS è 0.

## Change request 3 - Aggiornamento ID Evento

### Analisi richiesta modifica

La CR3 prevede una correzione dello schema DB realizzato nel sistema attuale. Gli identificativi degli eventi del sistema sono delle stringhe e non hanno nessuna limitazione, un esempio di chiave di un evento è "Evento di Luglio". Si è deciso di disciplinare gli ID inseriti in DB inserendo una fase di "traduzione" dei nomi degli eventi in stringhe contenenti simboli alfanumerici e trattini (per esempio, un evento di nome "Evento anno 2023" avrà come codice "evento-anno-2023"). Si è inoltre deciso di simulare di avere già un database in esercizio, e quindi si è prevista una fase di migrazione di schema e dati di un database preesistente. In seguito i nuovi requisiti funzionali:

### Impact Analysis

#### Identificazione SIS (Starting Impact Set)

Componente	Tipo	Modifica	Azione
EventoBean	Bean	Aggiunta codice come variabile di istanza, getter & setter.	Aggiunta
EventoDAO	DAO	<ul style="list-style-type: none"><li>Utilizzo codice evento come ID invece di nome</li><li>Utilizzo Slug Generator</li></ul>	Modifiche
PartecipazioneBean	Bean	Utilizzo codice evento invece di nome.	Modifica
PartecipazioneDAO	DAO	Utilizzo codice evento invece di nome.	Modifica
RecensioneBean	Bean	Utilizzo codice evento invece di nome.	Modifica
RecensioneDAO	DAO	Utilizzo codice evento invece di nome.	Modifica
SlugGenerator	Classe	Generatore di slug.	Aggiunta
EventiRecenti.jsp	JSP	Utilizzo codice evento invece di nome.	Modifica



PartecipaEventi.jsp	JSP	Utilizzo codice evento invece di nome.	Modifica
RichiesteEventi.jsp	Bean	Aggiunta dell'attributo "tipologia".	Modifica
TestEventoDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestPartecipazioneDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestPartecipazioneSquadraDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestRecensioneDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestSlugGenerator	Test	Test unit di SlugGenerator.	Aggiunta
Evento	Tabella	<ul style="list-style-type: none"><li>• Aggiunta campo codice</li><li>• Nome chiave candidata non più primaria</li></ul>	Aggiunta
Partecipazione	Tabella	Utilizzo codice evento invece di nome come chiave esterna.	Modifica
Recensione	Tabella	Utilizzo codice evento invece di nome come chiave esterna.	Modifica

#### Identificazione CIS (Candidate Impact Set)

Componente	Tipo	Modifica	Azione
CreaEventoServlet	Servlet	Lettura evento esplicitamente via nome.	Modifica
PartecipaEventiServlet	Servlet	Lettura codice evento da parametri, non nome.	Modifica
RecensioneServlet	Servlet	Lettura codice evento da parametri, non nome.	Modifica
EventoBean	Bean	Aggiunta codice come variabile di istanza, getter & setter.	Aggiunta
EventoDAO	DAO	<ul style="list-style-type: none"><li>• Utilizzo codice evento come ID invece di nome</li><li>• Utilizzo Slug Generator</li></ul>	Modifiche
PartecipazioneBean	Bean	Utilizzo codice evento invece di nome.	Modifica
PartecipazioneDAO	DAO	Utilizzo codice evento invece di nome.	Modifica
RecensioneBean	Bean	Utilizzo codice evento invece di nome.	Modifica
RecensioneDAO	DAO	Utilizzo codice evento invece di nome.	Modifica
SlugGenerator	Classe	Generatore di slug.	Aggiunta
EventiRecenti.jsp	JSP	Utilizzo codice evento invece di nome.	Modifica
PartecipaEventi.jsp	JSP	Utilizzo codice evento invece di nome.	Modifica
RichiesteEventi.jsp	Bean	Utilizzo codice evento invece di nome.	Modifica

<b>TestCreaEventoServlet</b>	Test	Utilizzo codice evento invece di nome.	Modifica
<b>TestPartecipaEventiServlet</b>	Test	Utilizzo codice evento invece di nome.	Modifica
<b>TestRecensioneServlet</b>	Test	Utilizzo codice evento invece di nome.	Modifica
TestEventoDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestPartecipazioneDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestPartecipazioneSquadraDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestRecensioneDAO	Test	Utilizzo codice evento invece di nome.	Modifica
<b>AllTestUnitUtil</b>	Test suite	Aggiunta test di Slug Generator.	Aggiunta
TestSlugGenerator	Test	Test unit di SlugGenerator.	Aggiunta
Evento	Tabella	<ul style="list-style-type: none"> <li>• Aggiunta campo codice</li> <li>• Nome chiave candidata non più primaria</li> </ul>	Aggiunta
Partecipazione	Tabella	Utilizzo codice evento invece di nome come chiave esterna.	Modifica
Recensione	Tabella	Utilizzo codice evento invece di nome come chiave esterna.	Modifica

#### Identificazione AIS (Actual Impact Set)

Componente	Tipo	Modifica	Azione
CreaEventoServlet	Servlet	Lettura evento esplicitamente via nome.	Modifica
PartecipaEventiServlet	Servlet	Lettura codice evento da parametri, non nome.	Modifica
RichiesteEventiServlet	Servlet	Lettura codice evento da parametri, non nome.	Modifica
RecensioneServlet	Servlet	Lettura codice evento da parametri, non nome.	Modifica
EventoBean	Bean	Aggiunta codice come variabile di istanza, getter & setter.	Aggiunta
EventoDAO	DAO	<ul style="list-style-type: none"> <li>• Utilizzo codice evento come ID invece di nome</li> <li>• Utilizzo Slug Generator</li> </ul>	Modifiche
PartecipazioneBean	Bean	Utilizzo codice evento invece di nome.	Modifica
PartecipazioneDAO	DAO	Utilizzo codice evento invece di nome.	Modifica
RecensioneBean	Bean	Utilizzo codice evento invece di nome.	Modifica



RecensioneDAO	DAO	Utilizzo codice evento invece di nome.	Modifica
SlugGenerator	Classe	Generatore di slug.	Aggiunta
DaiRecensione.jsp	JSP	Utilizzo codice evento invece di nome.	Modifica
EventiRecenti.jsp	JSP	Utilizzo codice evento invece di nome.	Modifica
PartecipaEventi.jsp	JSP	Utilizzo codice evento invece di nome.	Modifica
RichiesteEventi.jsp	Bean	Utilizzo codice evento invece di nome.	Modifica
TestCreaEventoServlet	Test	Utilizzo codice evento invece di nome.	Modifica
TestPartecipaEventiServlet	Test	Utilizzo codice evento invece di nome.	Modifica
TestRecensioneServlet	Test	Utilizzo codice evento invece di nome.	Modifica
TestRichiesteEventiServlet	Test	Utilizzo codice evento invece di nome.	Modifica
TestEventoDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestPartecipazioneDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestPartecipazioneSquadraDAO	Test	Utilizzo codice evento invece di nome.	Modifica
TestRecensioneDAO	Test	Utilizzo codice evento invece di nome.	Modifica
AllTestUnitUtil	Test suite	Aggiunta test di Slug Generator.	Aggiunta
TestSlugGenerator	Test	Test unit di SlugGenerator.	Aggiunta
Evento	Tabella	<ul style="list-style-type: none"><li>• Aggiunta campo codice</li><li>• Nome chiave candidata non più primaria</li></ul>	Aggiunta
Partecipazione	Tabella	Utilizzo codice evento invece di nome come chiave esterna.	Modifica
Recensione	Tabella	Utilizzo codice evento invece di nome come chiave esterna.	Modifica

#### Identificazione FPIS (False Positive Impact Set)

Non sono stati introdotti falsi positivi.

#### Identificazione DIS (Discovered Impact Set)

Componente	Tipo	Modifica	Azione
RichiesteEventiSer	Servlet	Lettura codice evento da parametri, non nome.	Modifica



vlet			
DaiRecensione.jsp	JSP	Utilizzo codice evento invece di nome.	Modifica
TestRichiesteEvent iServlet	Test	Utilizzo codice evento invece di nome.	Modifica

### Metriche Impact Analysis

Numero di componenti nel SIS: 20

Numero di componenti nel CIS: 27

Numero di componenti nel AIS: 30

Numero di componenti nel FPIS: 0

Numero di componenti nel DIS: 3

**Recall:**  $|CIS \cap AIS| \div |AIS| = 27 \div 30 = 0,9$ .

**Precision:**  $|CIS \cap AIS| \div |CIS| = 27 \div 27 = 1$ . (perché FPIS=0)

## Change request 4 - Evento annullato, iniziato e finito

### Analisi richiesta modifica

La CR3 prevede l'aggiunta di tre stati alla entità Evento. In particolare, gli stati aggiuntivi sono:

- annullato: un evento diventa annullato se non ha raggiunto il numero di giocatori necessari entro dieci minuti dopo il suo inizio previsto;
- iniziato: se è in stato completato e scatta l'inizio pianificato;
- - finito: se è in stato iniziato e scatta la fine dell'evento pianificato (tempo di inizio più un'ora di durata).

### Impact Analysis

#### Identificazione SIS (Starting Impact Set)

Componente	Tipo	Modifica	Azione
CronJob	Classe	Aggiunta di logiche di timer per gestire aggiornamento automatico stati eventi.	Aggiunta
EventStateUpdater	Classe	Permette di calcolare un nuovo potenziale stato per un evento.	Aggiunta



InitializeListener	Classe	Utility per avviare il timer all'inizio dell'esecuzione del software nell'Application Server.	Aggiunta
MB_CR4_Regression	Test suite	Test suite di regressione della CR4.	Aggiunta
AllTestUnitaModel	TestSuite	Aggiunta test unit TestStatoEventoBean	Modifica
TestStatoEventoBean	Test	Verifica funzionamento cambiamenti stati evento.	Aggiunta

#### Identificazione CIS (Candidate Impact Set)

Componente	Tipo	Modifica	Azione
CronJob	Classe	Aggiunta di logiche di timer per gestire aggiornamento automatico stati eventi.	Aggiunta
EventStateUpdater	Classe	Permette di calcolare un nuovo potenziale stato per un evento.	Aggiunta
InitializeListener	Classe	Utility per avviare il timer all'inizio dell'esecuzione del software nell'Application Server.	Aggiunta
MB_CR4_Regression	Test suite	Test suite di regressione della CR4.	Aggiunta
AllTestUnitaModel	TestSuite	Aggiunta test unit TestStatoEventoBean	Modifica
TestStatoEventoBean	Test	Verifica funzionamento cambiamenti stati evento.	Aggiunta

#### Identificazione AIS (Actual Impact Set)

Componente	Tipo	Modifica	Azione
CronJob	Classe	Aggiunta di logiche di timer per gestire aggiornamento automatico stati eventi.	Aggiunta
EventStateUpdater	Classe	Permette di calcolare un nuovo potenziale stato per un evento.	Aggiunta
InitializeListener	Classe	Utility per avviare il timer all'inizio dell'esecuzione del software nell'Application Server.	Aggiunta
MB_CR4_Regression	Test suite	Test suite di regressione della CR4.	Aggiunta
AllTestUnitaModel	TestSuite	Aggiunta test unit TestStatoEventoBean	Modifica
TestEventoBean	Test	Dismissione controlli precedenti allo stato finito.	Rimozione
TestStatoEventoBean	Test	Verifica funzionamento cambiamenti stati evento.	Aggiunta
EventiRecenti.jsp	JSP	Controllo stato finito di eventi.	Aggiunta



### Identificazione FPIS (False Positive Impact Set)

Non sono stati introdotti falsi positivi.

### Identificazione DIS (Discovered Impact Set)

Componente	Tipo	Modifica	Azione
TestEventoBean	Test	Dismissione controlli precedenti allo stato finito.	Rimozione
EventiRecenti.jsp	JSP	Controllo stato finito di eventi.	Aggiunta

### Metriche Impact Analysis

Numero di componenti nel SIS: 6

Numero di componenti nel CIS: 6

Numero di componenti nel AIS: 8

Numero di componenti nel FPIS: 0

Numero di componenti nel DIS: 2

**Recall:**  $|CIS \cap AIS| \div |AIS| = 6 \div 8 = 0,75$ .

**Precision:**  $|CIS \cap AIS| \div |CIS| = 6 \div 6 = 1$ . (perché FPIS=0)



# Testing Execution Report

In questa sezione vengono riportati i risultati dell'esecuzione dei test per ogni CR.

## CR 1 - Cifratura password

### Tests di unità e integrazione

Cover: MB\_CR1\_Regression x

Tests passed: 46 of 46 tests – 7 sec 27 ms

Test	Time
MB_CR1_Regression (SuitesMaintenance)	7 sec 27 ms
TestLoginServlet	3 sec 745 ms
loginGestoreOk	2 sec 248 ms
notGioLogged	8 ms
notGioLoggedErrorPass	1 sec 256 ms
loginGiocatoreOk	103 ms
notGesLogged	8 ms
doGet	5 ms
notGesLoggedErrorPass	10 ms

IntelliJ IDEA coverage runner

include patterns:

```
control\utente\.LoginServlet(\$.*)*
control\utente\.LogoutServlet(\$.*)*
control\utente\.RegistrazioneServlet(\$.*)*
model\utente\.gestore\.GestoreDAO(\$.*)*
```

Element	Class, %	Method, %	Line, %	Branch, %
all				
control.utente.LoginServlet	100% (...)	85% (6/7)	89% (41/46)	92% (13/14)
control.utente.LogoutServlet	100% (...)	50% (1/2)	81% (9/11)	83% (5/6)
control.utente.RegistrazioneServlet	100% (...)	100% (6/6)	96% (119/...	96% (29/30)
model.utente.gestore.GestoreDAO	100% (...)	100% (7/7)	83% (138/...	29% (17/58)
model.utente.giocatore.GiocatoreDAO	100% (...)	90% (9/10)	82% (221/...	25% (24/96)

Cover: TestHashTool x

Tests passed: 1 of 1 test – 53 ms

Test	Time
TestHashTool (unita.util)	53 ms
testHashSHA256	53 ms

IntelliJ IDEA coverage runner

tracing and tracking per test coverage ...

include patterns:

```
unita\util\.*
util\.HashTool(\$.*)*
```

exclude patterns:

Class transformation time: 0.0427578s for 351 classes

Element	Class, %	Method, %	Line, %	Branch, %
util.HashTool	100% (...)	100% (3/3)	100% (11/...	100% (4/4)

## Test di sistema

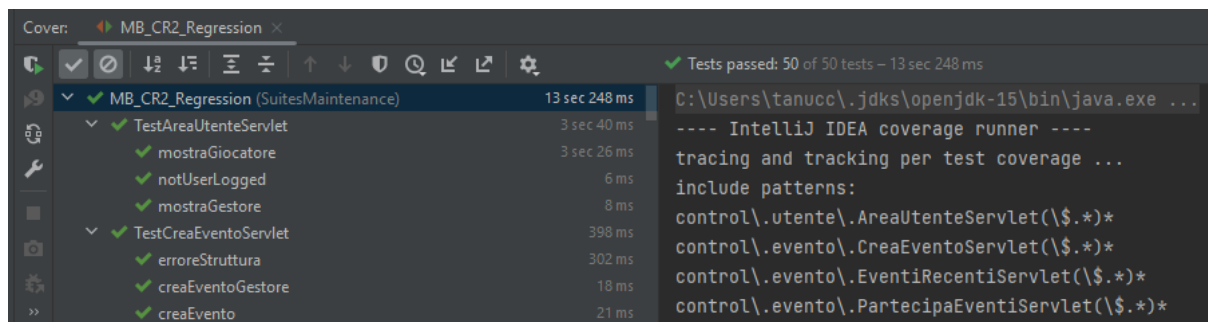
Sono stati rieseguiti i test dei sottosistemi:

1. Gestione giocatore (TC\_4.1\_X)
2. Gestione gestore (TC\_4.2\_X)
3. Gestione autenticazione (TC\_4.3\_X)

Tutti i test sono andati a buon fine e quindi non è stato individuato nessun errore.

## CR 2 - Entità squadra

### Tests di unità e integrazione



Cover: MB\_CR2\_Regression

Tests passed: 50 of 50 tests – 13 sec 248 ms

Test	Duration
MB_CR2_Regression (SuitesMaintenance)	13 sec 248 ms
TestAreaUtenteServlet	3 sec 40 ms
mostraGiocatore	3 sec 26 ms
notUserLogged	6 ms
mostraGestore	8 ms
TestCreaEventoServlet	398 ms
erroreStruttura	302 ms
creaEventoGestore	18 ms
creaEvento	21 ms

IntelliJ IDEA coverage runner

tracing and tracking per test coverage ...

include patterns:

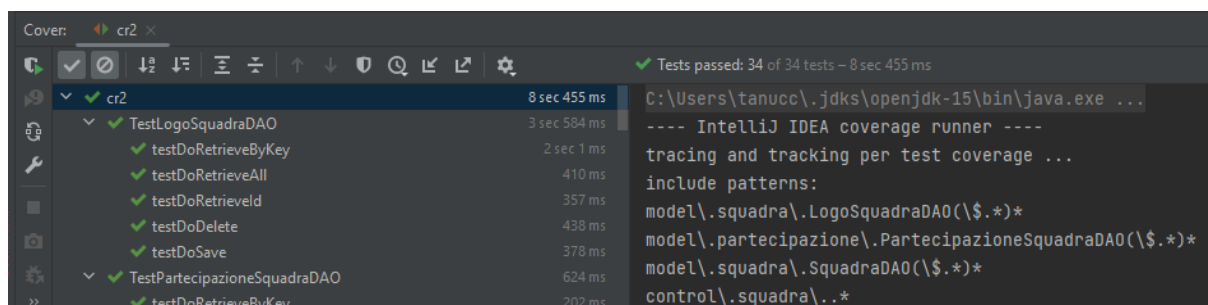
control\.\utente\.\AreaUtenteServlet(\\$.\*)\*

control\.\evento\.\CreaEventoServlet(\\$.\*)\*

control\.\evento\.\EventiRecentiServlet(\\$.\*)\*

control\.\evento\.\PartecipaEventiServlet(\\$.\*)\*

Element	Class, %	Method, %	Line, %	Branch, %
all				
control.evento.CreaEver	100% (...)	80% (4/5)	94% (49/...	100% (18...
control.evento.EventiRe	100% (...)	100% (2/2)	90% (18/...	100% (6/6)
control.evento.Partecip	100% (...)	100% (5/5)	92% (51/...	85% (17/...
control.evento.Richieste	100% (...)	83% (5/6)	94% (53/...	92% (24/...
control.utente.AreaUter	100% (...)	66% (2/3)	94% (17/...	100% (4/4)
model.evento.EventoDA	100% (...)	100% (12/...	91% (317...	27% (42/...



Cover: cr2

Tests passed: 34 of 34 tests – 8 sec 455 ms

Test	Duration
cr2	8 sec 455 ms
TestLogoSquadraDAO	3 sec 584 ms
testDoRetrieveByKey	2 sec 1 ms
testDoRetrieveAll	410 ms
testDoRetrieveVeld	357 ms
testDoDelete	438 ms
testDoSave	378 ms
TestPartecipazioneSquadraDAO	624 ms
testDoRetrieveByKey	202 ms

IntelliJ IDEA coverage runner

tracing and tracking per test coverage ...

include patterns:

model\.\squadra\.\LogoSquadraDAO(\\$.\*)\*

model\.\partecipazione\.\PartecipazioneSquadraDAO(\\$.\*)\*

model\.\squadra\.\SquadraDAO(\\$.\*)\*

control\.\squadra\.\*

Element ▲	Class, %	Method, %	Line, %	Branch, %
▼ all	100% (...)	76% (20/26)	87% (182...	97% (66/...
▼ control.squadra	100% (...)	76% (20/26)	87% (182...	97% (66/...
⊙ AbbandonaSquadra!	100% (...)	33% (1/3)	60% (14/...	50% (2/4)
⊙ CreaSquadraServlet	100% (...)	66% (4/6)	92% (70/...	100% (34...
⊙ EliminaSquadraServlet	100% (...)	66% (2/3)	87% (21/...	100% (6/6)
⊙ MostraSquadreServlet	100% (...)	100% (4/4)	91% (21/...	100% (6/6)
⊙ SquadraSpecificaServlet	100% (...)	100% (6/6)	94% (34/...	100% (10...
⊙ UniscitiSquadraServlet	100% (...)	75% (3/4)	81% (22/...	100% (8/8)
⊙ model.partecipazione.Pi	100% (...)	100% (3/3)	82% (48/...	28% (8/28)
⊙ model.squadra.LogoSql	100% (...)	100% (5/5)	87% (85/...	23% (19/...
⊙ model.squadra.Squadra	100% (...)	83% (5/6)	71% (99/...	23% (13/...

## Test di sistema

Sono stati rieseguiti i test dei sottosistemi:

1. Gestione evento (TC\_4.4\_X)
2. Gestione autenticazione (TC\_4.3\_X)

Tutti i test sono andati a buon fine e quindi non è stato individuato nessun errore.

Inoltre, sono stati effettuati nuovi test di sistema:

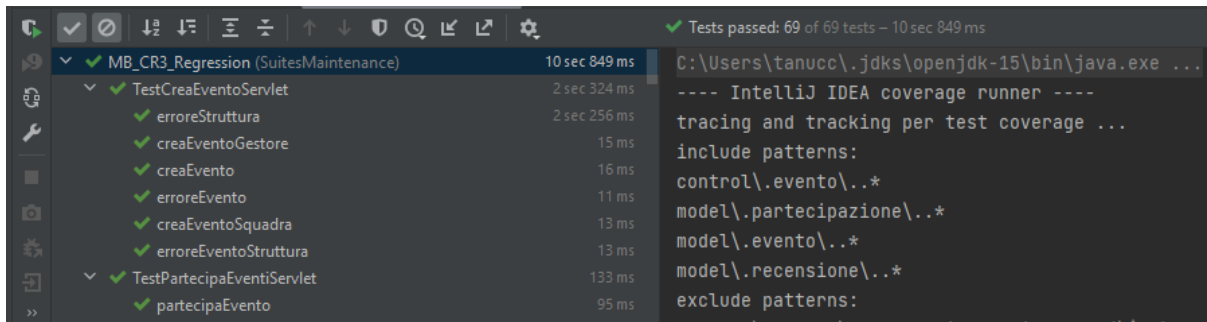
1. Gestione squadra (TC\_4.6\_X)

Durante il test del TC\_4.6\_13 è stata individuata un'eccezione nell'apertura della pagina della squadra specifica dopo la creazione. L'errore è stato risolto nel commit "Bug fix error TC\_4.6\_13" (cod. 3247f773c604394d3ef44033a908cd4ac59d24bf).

Tutti gli altri test sono andati a buon fine.

## CR 3 - Aggiornamento ID Evento

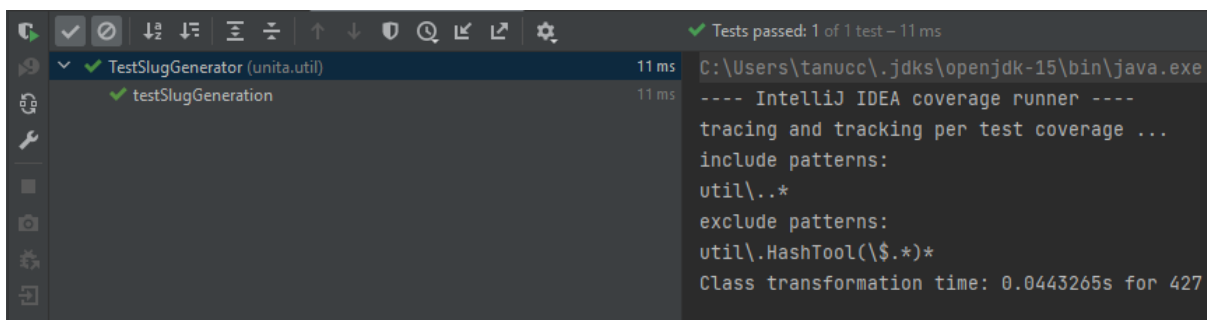
### Tests di unità e integrazione



Tests passed: 69 of 69 tests - 10 sec 849 ms

Test	Duration
TestCreaEventoServlet	2 sec 324 ms
erroreStruttura	2 sec 256 ms
creaEventoGestore	15 ms
creaEvento	16 ms
erroreEvento	11 ms
creaEventoSquadra	13 ms
erroreEventoStruttura	13 ms
TestPartecipaEventiServlet	133 ms
partecipaEvento	95 ms

Element	Clas...	Metho...	Line, %	Branch...
control.evento	100...	87% (14...	93% (15...	92% (59...
CreaEventoSe	100...	80% (4/5)	94% (49...	100% (1...
PartecipaEven	100...	100% (5...	92% (52...	85% (17...
RichiesteEven	100...	83% (5/6)	94% (52...	92% (24...
model.evento	100...	86% (39...	89% (38...	28% (44...
EventoBean	100...	81% (26...	71% (27...	0% (0/2)
EventoDAO	100...	100% (1...	91% (35...	28% (44...
model.partecipaz	100...	100% (1...	82% (16...	32% (24...
Partecipazioni	100...	100% (5...	100% (6...	100% (0...
Partecipazioni	100...	100% (6...	80% (10...	34% (16...
Partecipazioni	100...	100% (5...	100% (6...	100% (0...
Partecipazioni	100...	100% (3...	82% (48...	28% (8/...
model.recensione	100...	100% (1...	88% (16...	28% (24...
RecensioneBe	100...	100% (1...	100% (1...	100% (0...



Tests passed: 1 of 1 test - 11 ms

Test	Duration
testSlugGeneration	11 ms

Element	Clas...	Metho...	Line, %	Branch...
util	100...	100% (1...	100% (7...	100% (0...
SlugGenerator	100...	100% (1...	100% (7...	100% (0...

## Test di sistema

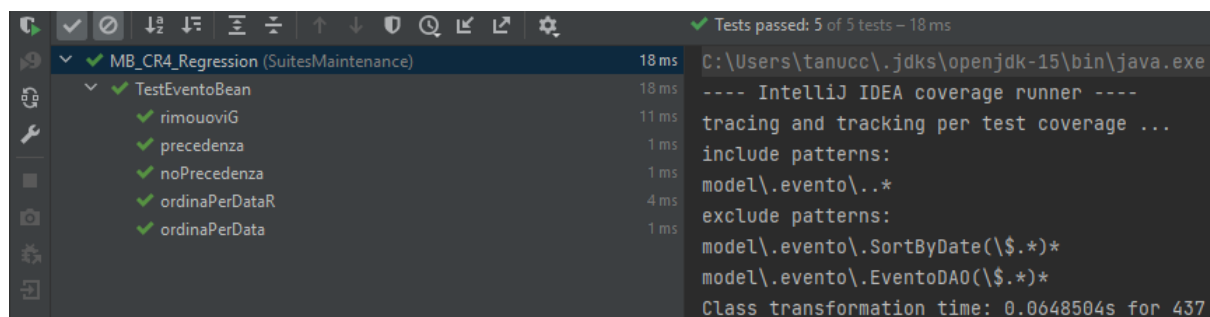
Sono stati rieseguiti i test dei sottosistemi:

1. Gestione evento (TC\_4.4\_X)
2. Gestione recensione (TC\_4.5\_X)

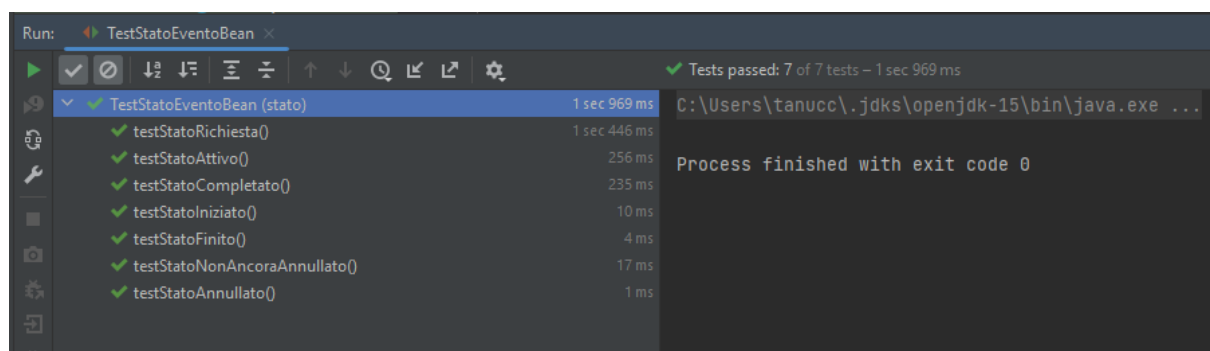
Tutti i test sono andati a buon fine e quindi non è stato individuato nessun errore.

## CR 4 - Evento annullato, iniziato e finito

### Tests di unità e integrazione



Element	Clas...	Metho...	Line, %	Branch...
model.evento	100...	53% (17...	60% (23...	100% (...)
EventoBean	100...	53% (17...	60% (23...	100% (...)



## Test di sistema

Sono stati rieseguiti i test dei sottosistemi:

1. Gestione evento (TC\_4.4\_X)

Tutti i test sono andati a buon fine e quindi non è stato individuato nessun errore.

# Panoramica sistema modificato

## Descrizione sistema modificato

MyBomber intende fornire agli appassionati di calcio e ai gestori degli impianti sportivi un sistema per migliorare e rendere più semplice l'organizzazione e/o prenotazione di match, tra amici, conoscenti o sconosciuti, con lo scopo di creare una vera e propria rete che permetta di collegare tutti gli appassionati tra di loro. Gli utenti potranno scambiarsi dei feedback sulla prestazione tra di loro o unirsi alle squadre presenti sul sistema.

## Attori del sistema

Nel sistema attuale si hanno i seguenti attori:

- **Giocatore**
  - Può partecipare agli eventi o crearne uno
  - Gestire gli eventi a cui partecipa o ha partecipato
  - Può lasciare una recensione ad un altro giocatore
- **Gestore**
  - Accettare o rifiutare eventi presso la propria struttura
  - Controlla gli eventi da svolgere o che sono stati già svolti presso la propria struttura
- **Capitano**
  - Creare o partecipare a eventi di tipo squadra
  - Eliminare la propria squadra

## Requisiti funzionali

### Gestione Utente

*RF\_GU\_1* - Il sistema dovrà permettere ad un giocatore di effettuare la registrazione, **priorità alta**.

*RF\_GU\_2* - Il sistema dovrà permettere ad un gestore di effettuare la registrazione, **priorità alta**.

*RF\_GU\_3* - Il sistema dovrà permettere ad un gestore o un giocatore di effettuare il login, **priorità alta**.

*RF\_GU\_4* - Il sistema dovrà permettere ad un utente loggato di effettuare il logout, **priorità alta**.



*RF\_GU\_5* - Il sistema dovrà permettere ad un utente loggato di visualizzare l'area utente, **priorità alta**.

*RF\_GU\_6* - Il sistema dovrebbe permettere ad un utente loggato di modificare i dati personali attraverso l'area utente, **priorità bassa**.

*RF\_GU\_7* - Il sistema dovrebbe permettere ad un utente registrato di recuperare la propria password, **priorità bassa**.

### **Gestione Evento**

*RF\_GE\_1* - Il sistema dovrà permettere ad un giocatore di sottomettere una richiesta per un nuovo evento, **priorità alta**.

- *RF\_GE\_1\_US* (versione user story): Come giocatore voglio poter creare un nuovo evento per pianificare una nuova partita.

*RF\_GE\_2* - Il sistema dovrà permettere ad un giocatore di visualizzare gli eventi disponibili, **priorità alta**.

*RF\_GE\_3* - Il sistema dovrà permettere ad un giocatore di partecipare ad un evento, **priorità alta**.

- *RF\_GE\_3\_US* (versione user story): Come giocatore voglio poter partecipare ad un evento attraverso la selezione del medesimo evento nella lista degli eventi disponibili.

*RF\_GE\_4* - Il sistema dovrà permettere ad un giocatore di visualizzare i dettagli di un evento, **priorità bassa**.

*RF\_GE\_5* - Il sistema dovrà permettere ad un giocatore di visualizzare gli eventi in programma, **priorità alta**.

*RF\_GE\_6* - Il sistema dovrà permettere ad un giocatore di visualizzare gli eventi recenti, **priorità alta**.

*RF\_GE\_7* - Il sistema dovrà permettere ad un gestore di visualizzare gli eventi attivi della propria struttura, **priorità alta**.

- *RF\_GE\_7\_US* (versione user story): Come gestore voglio poter visualizzare gli eventi attivi nella mia struttura per coordinare al meglio la stessa.

*RF\_GE\_8* - Il sistema dovrebbe permettere ad un giocatore di annullare un evento da lui organizzato, **priorità bassa**.

*RF\_GE\_9* - Il sistema dovrà permettere ad un utente di visualizzare la cronologia eventi, **priorità alta**.





*RF\_GE\_10* - Il sistema dovrà permettere ad un gestore di visualizzare le richieste di nuovi eventi, **priorità alta**.

*RF\_GE\_11* - Il sistema dovrà permettere ad un gestore di accettare una richiesta evento tramite la pagina di gestione delle richieste, **priorità alta**.

- *RF\_GE\_11\_US* (versione user story): Come gestore voglio poter accettare la richiesta di un nuovo evento per gestire la struttura.

*RF\_GE\_12* - Il sistema dovrà permettere ad un gestore di declinare una richiesta evento tramite la pagina di gestione delle richieste, **priorità alta**.

*RF\_GE\_13* - Il sistema dovrà permettere al giocatore di visualizzare la sua valutazione, **priorità alta**.

*RF\_GE\_14* - Il sistema dovrebbe permettere ad un gestore di aggiungere un nuovo evento, **priorità bassa**.

*RF\_GE\_15* - Il sistema dovrebbe permettere ad un utente di ricercare un evento, **priorità bassa**.

*RF\_GE\_16* - Il sistema dovrebbe permettere ad un gestore di modificare un evento, **priorità bassa**.

*RF\_GE\_17* - Il sistema dovrebbe permettere ad un utente di visualizzare gli eventi disponibili, **priorità bassa**.

*RF\_GE\_18* - Il sistema dovrebbe inviare una notifica al giocatore se un evento a cui partecipa viene annullato, **priorità bassa**.

*RF\_GE\_19* - Il sistema dovrebbe inviare una notifica al giocatore se una richiesta per un nuovo evento viene accettata, **priorità bassa**.

*RF\_GE\_20* - Il sistema dovrebbe inviare una notifica al gestore se una richiesta per un nuovo evento viene sottomessa, **priorità bassa**.

*RF\_GE\_21* - Il sistema dovrebbe permettere ad un giocatore di abbandonare un evento, **priorità bassa**.

### **Gestione Recensione**

*RF\_GR\_1* - Il sistema dovrà permettere ad un giocatore di recensire un altro giocatore con cui ha giocato, **priorità alta**.

- *RF\_GR\_1\_US* (versione user story): Come giocatore voglio poter recensire un altro giocatore per tenere aggiornata la sua valutazione.





*RF\_GR\_2* - Il sistema dovrà permettere ad un giocatore di rimuovere una recensione da lui inserita, **priorità alta**.

*RF\_GR\_3* - Il sistema dovrebbe permettere ad un giocatore di modificare una recensione da lui inserita, **priorità bassa**.

*RF\_GR\_4* - Il sistema dovrebbe permettere ad un giocatore di visualizzare la cronologia delle recensioni, **priorità bassa**.

### **Gestione Squadra**

*RF\_GS\_01* Il sistema dovrà permettere ad un giocatore di poter creare una squadra di calcio, **priorità alta**.

*RF\_GS\_02* Il sistema dovrà permettere ad un capitano di poter eliminare la squadra che ha creato, **priorità bassa**.

*RF\_GS\_03* Il sistema dovrà permettere ad un giocatore di poter visionare le informazioni di una squadra di calcio, **priorità alta**.

*RF\_GS\_04* Il sistema dovrà permettere ad un giocatore di aggiungersi ad una squadra, **priorità alta**.

*RF\_GS\_05* Il sistema dovrà permettere ad un giocatore di abbandonare la sua squadra, **priorità alta**.

*RF\_GS\_06* Il sistema dovrà permettere al capitano della squadra di accettare o declinare la richiesta di un giocatore nella sua squadra, **priorità bassa**.

*RF\_GS\_07* Il sistema dovrà permettere al capitano della squadra di invitare giocatori nella sua squadra, **priorità bassa**.

*RF\_GS\_08* Il sistema dovrà permettere al capitano di una squadra di creare degli eventi di tipo squadra, **priorità alta**.

*RF\_GS\_09* Il sistema dovrà permettere al capitano di partecipare a degli eventi di tipo squadra, **priorità alta**.

*RF\_GS\_10* Il sistema dovrà permettere ai giocatori di una squadra di visionare gli eventi squadra a cui dovranno partecipare, **priorità alta**.

*RF\_GS\_11* Il sistema dovrà permettere a un capitano di poter recensire un'altra squadra, **priorità bassa**.

*RF\_GS\_12* Il sistema dovrà permettere al capitano di poter eliminare la recensione fatta ad un'altra squadra, **priorità bassa**.

RF\_GS\_13 Il sistema dovrà permettere ad un giocatore di poter visionare le recensioni di una squadra, **priorità bassa**.

RF\_GS\_14 Il sistema dovrà permettere ai giocatori di visionare gli ultimi match di una squadra, **priorità bassa**.

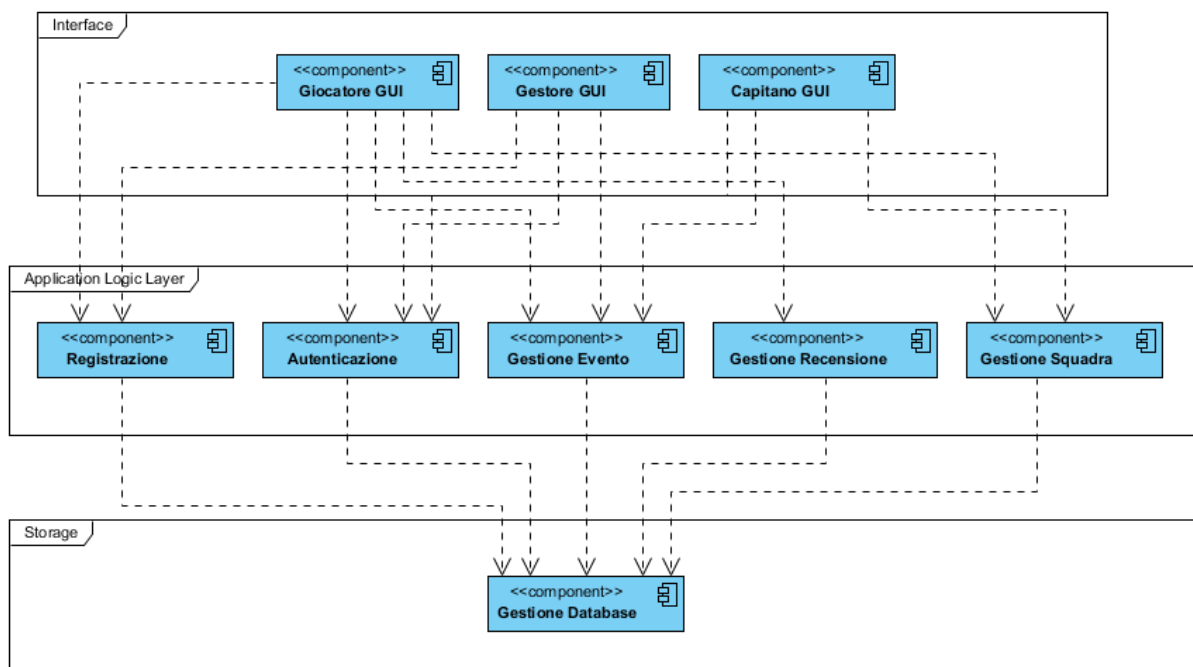
RF\_GS\_15 Il sistema dovrà permettere ai giocatori di visionare tutte le squadre presenti sul sistema, **priorità alta**.

RF\_GS\_16 Il sistema dovrà permettere al capitano di poter modificare le informazioni della propria squadra, **priorità bassa**.

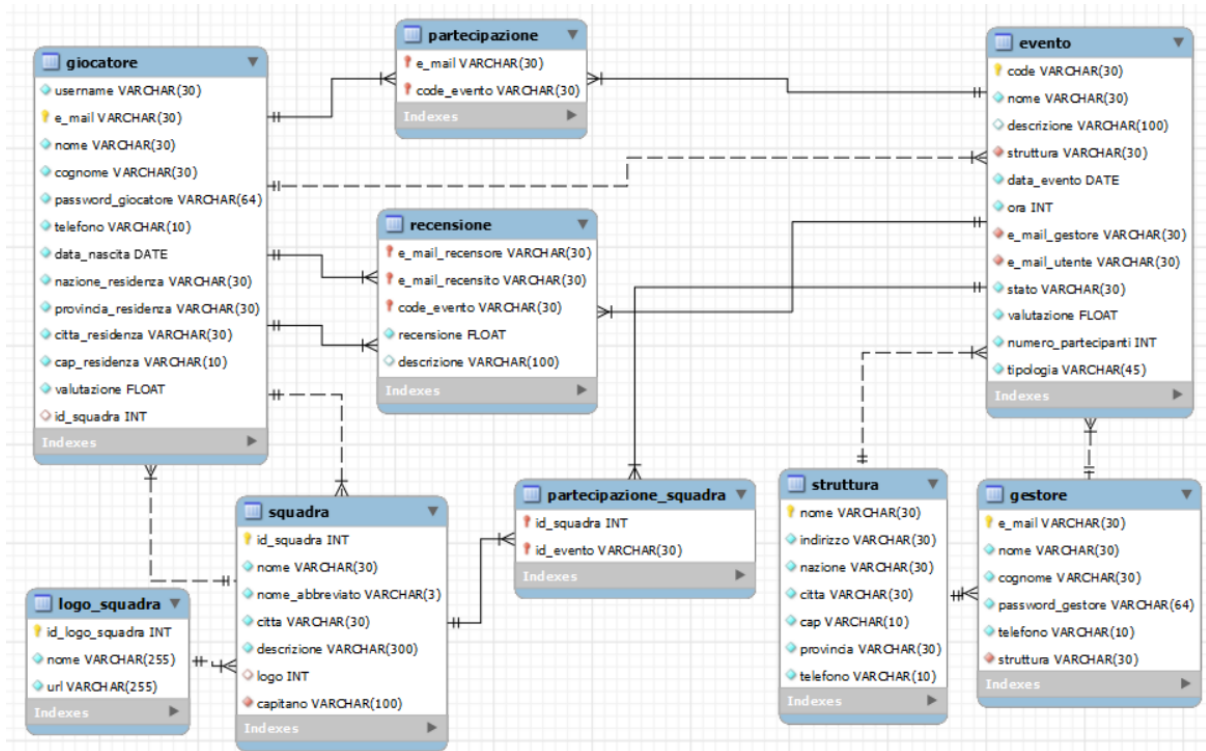
## Architettura del sistema

Il nostro sistema è basato su un'architettura Three-tier. Quindi, i sottosistemi sono organizzati in tre strati:

- L'interface layer (include tutti i boundary object che interfacciano con l'utente);
- L'application logic layer (include tutti gli oggetti relativi al controllo e alle entità che realizzano l'elaborazione, le regole di verifica e la notifica richieste dall'applicazione);
- Lo storage layer (effettua la memorizzazione, il recupero e l'interrogazione di oggetti persistenti).



## Gestione dati persistenti



## Statechart evento

