[Sommario 1](#_Toc127467311)

[1. Introduzione 1](#_Toc127467312)

[Scopo del Sistema 1](#_Toc127467313)

[Obiettivi di Design (Design Goal) 1](#_Toc127467314)

[Definizioni 3](#_Toc127467315)

[Organizzazione del Documento 3](#_Toc127467316)

[2. Architettura del Sistema Corrente 4](#_Toc127467317)

[3. Architettura del Sistema Proposto 4](#_Toc127467318)

[3.1 Panoramica sulla Sezione 4](#_Toc127467319)

[3.2 Decomposizione in Sottosistemi 4](#_Toc127467320)

[DIAGRAMMA ARCHITETTURALE 5](#_Toc127467321)

[3.3 Mapping Hardware/Software 7](#_Toc127467322)

[3.4 Class Diagram ristrutturato 8](#_Toc127467323)

[3.5 Gestione dei Dati Persistenti 9](#_Toc127467324)

[3.6 Controllo degli Accessi e Sicurezza 9](#_Toc127467325)

[3.7 Controllo Globale del Software 10](#_Toc127467326)

[3.8 Condizioni Limite 10](#_Toc127467327)

[4. Servizi dei Sottosistemi 12](#_Toc127467328)

# Introduzione

## Scopo del Sistema

Lo scopo del sistema è quello di esemplificare le interazioni tra gli utenti e lo sport, al fine di avvicinare quanto più possibile le persone a questo.

Il sistema, gestito da uno o più amministratori, permette le iscrizioni da parte di gestori di campetti e di utenti appassionati di sport.

## Obiettivi di Design (Design Goal)

Nella presente sezione si andranno a presentare i Design Goals, ovvero le qualità sulle quali il sistema deve essere focalizzato, formalizzati esplicitamente così che qualsiasi importante decisione di design può essere fatta consistentemente seguendo lo stesso insieme di design goal.

Ciascun design goal è descritto da:

* **Rank**: specifica di priorità; valori da 1 a 8;
* **ID Design Goal**: identificatore univoco e nome esplicativo;
* **Descrizione**: piccola descrizione del design goal;
* **Categoria**: categoria di appartenenza del design goal;
* **RNF di origine**: requisito non funzionale che ha originato il design goal.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RANK | ID | DESCRIZIONE | CATEGORIA | RNF DI ORIGINE |
| 5 | DG\_1  Tempi di risposta | Risposta del  server in massimo  2 secondi | Performance | RNF\_1 |
| 8 | DG\_2  Interfaccia Intuitiva | Interfaccia grafica  intuitiva e di facile  utilizzo | End User | RNF\_2 |
| 6 | DG\_3  Linguaggio Utilizzato | Utilizzo del linguaggio java e di MySQL. |  |  |
| 4 | DG\_4  Sistema Responsive | L’interfaccia grafica si deve adattare ad ogni tipo di schermo; anche mobile | End User | RNF\_6 |
| 7 | DG\_5  Disponibilità | Il sistema deve essere attivo 24/7 | Dependability | RNF\_7 |
| 3 | DG\_6  Manutenibilità | Sistema  sviluppato in  modo da essere  ben manutenibile | Manutenzione | RNF\_8 |
| 2 | DG\_7  Fallimento Sistema | Il sistema deve sapersi comportare in situazioni di fallimento notificando l’utente, tramite appositi messaggi | Dependability | RNF\_9 |

**Trade-Off**

|  |  |
| --- | --- |
| **Chiarezza del codice vs brevita’** | Si punta a garantire una migliore chiarezza e leggibilita’ del codice a scapito della brevita’ di questo. |
| **Prestazioni vs sicurezza** | Trattandosi di un progetto didattico si preferisce aumentare le performance a scapito della sicurezza dei dati. |
| **Immediatezza vs banda** | Si sceglie di caricare tutte le immagini ed i dati necessari fin da subito, evitando di utilizzare oggetti proxy che potrebbero far risparmiare banda. |

## Definizioni

Vengono di seguito delle definizioni presenti nel documento corrente:

* Sottosistema: un sottoinsieme del dominio applicativo, formato da servizi che hanno una stretta relazione funzionale.
* Design goal: le qualità sulle quali il sistema si deve focalizzare.
* Dati Persistenti: dati che vengono salvati in una base di dati.
* Mapping Hardware-Software: studio della connessione tra parti fisiche e logiche del sistema.
* SDD: System Design Document
* RAD: Requirements Analysis Document
* RNF: requisito non funzionali

## Organizzazione del Documento

Il documento si compone di quattro sezioni:

* Introduzione: Viene descritto in generale lo scopo del sistema e gli obbiettivi di design che il sistema si propone di raggiungere.
* Architettura del Sistema Corrente: Non viene descritta alcuna architettura, in quanto non esiste alcun software di questo tipo
* Architettura del Sistema Proposto: Viene descritto come il sistema sarà definito e partizionato in sottoinsiemi, il mapping Hardware/Software, la gestione dei dati persistenti e le boundary condition.
* Servizi dei Sottosistemi: Vengono indicati, per ogni sottosistema, i servizi appartenenti ad ognuno di essi.
* Glossario: contiene un chiarimento di alcuni termini o espressioni di difficile comprensione.

# Architettura del Sistema Corrente

Attualmente, non esiste un sistema che permette una tale gestione di prenotazioni di campetti. Le possibili alternative sono molto distanti dalla nostra proposta e pertanto risulta difficile il confronto.

## Architettura del Sistema Proposto

## 3.1 Panoramica sulla Sezione

Il sistema proposto è basato sullo stile architetturale Three Tier, implementato utilizzando MVC. Il motivo è che tale architettura è perfetta per lo sviluppo di WebApp, come visto e testato nel corso di TSW. In particolare la separazione dalla logica di presentazione da quella di elaborazione migliora alcuni aspetti implementativi andando a favorire lo sviluppo del sistema.

Verranno utilizzati JSP, JSTL, HTML5, CSS3, ed eventuali altri Framework come Bootstrap per la parte di front-end.

Per la logica applicativa e back-end sarà utilizzato **Java**

Per la gestione del database verrà usato **SQL**

## 3.2 Decomposizione in Sottosistemi

**Registrazione:** Gestisce la registrazione di utenti e gestori.

**Autenticazione:** Responsabile delle funzionalità di Login, Logout ed eliminazione dell’account

**Gestione Eventi:** Si occupa delle funzioni riguardanti la visualizzazione, la creazione, l’adesione, l’invito e l’eliminazione degli eventi

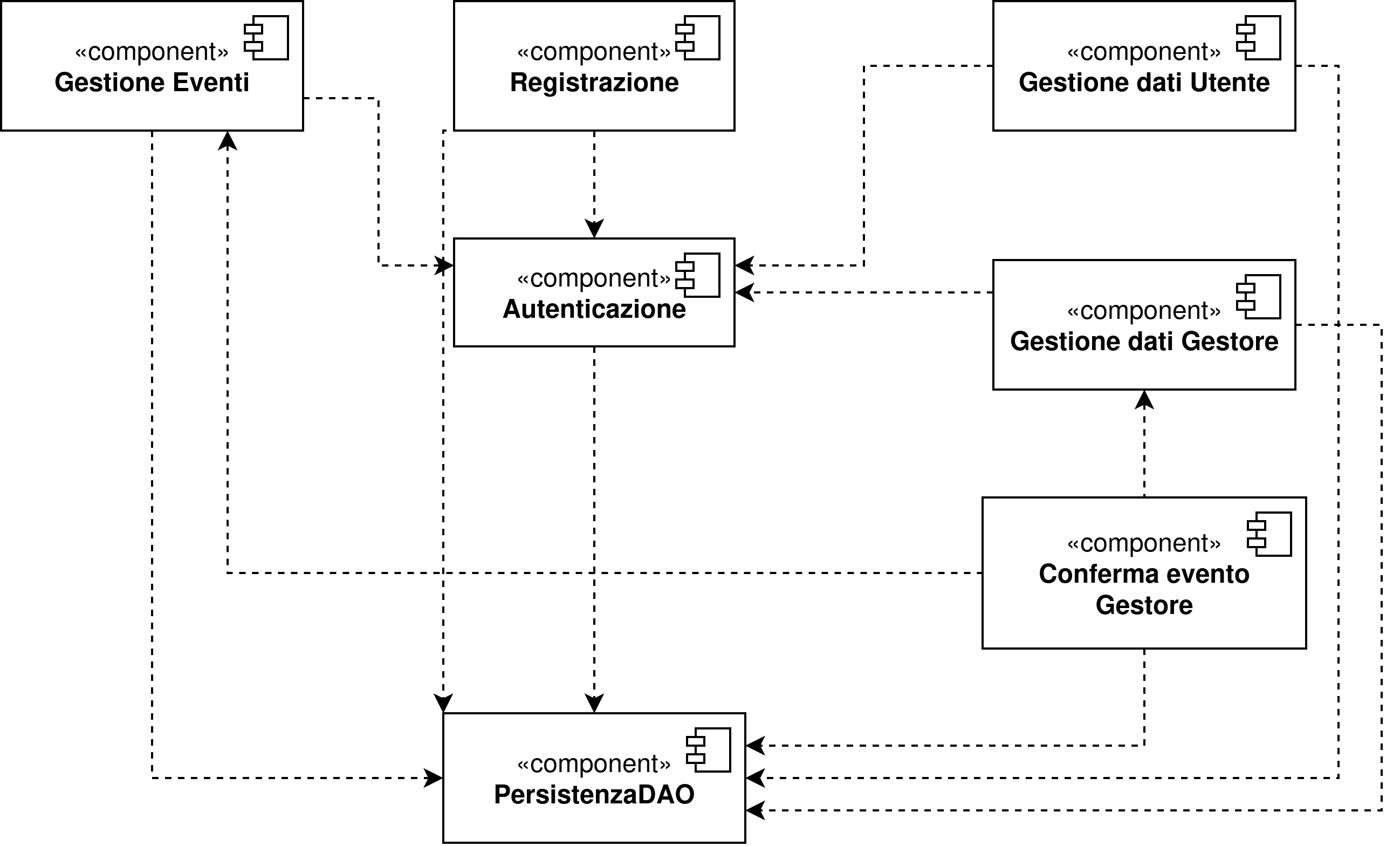
**Gestione dati utente:** Concerne la possibilità di modificare i propri dati personali e la visualizzazione degli eventi sottoscritti dall’utente

**Gestione dati gestore:** Concerne la possibilità di modificare i propri dati personali e quelli dei campi posseduti da parte del gestore

**Conferma evento Gestore:** La possibilità del gestore di accettare o rifiutare prenotazioni e, di conseguenza, di riscuotere o meno il pagamento

**Persistenza:** Si occupa di gestire la persistenza dei dati su un Database

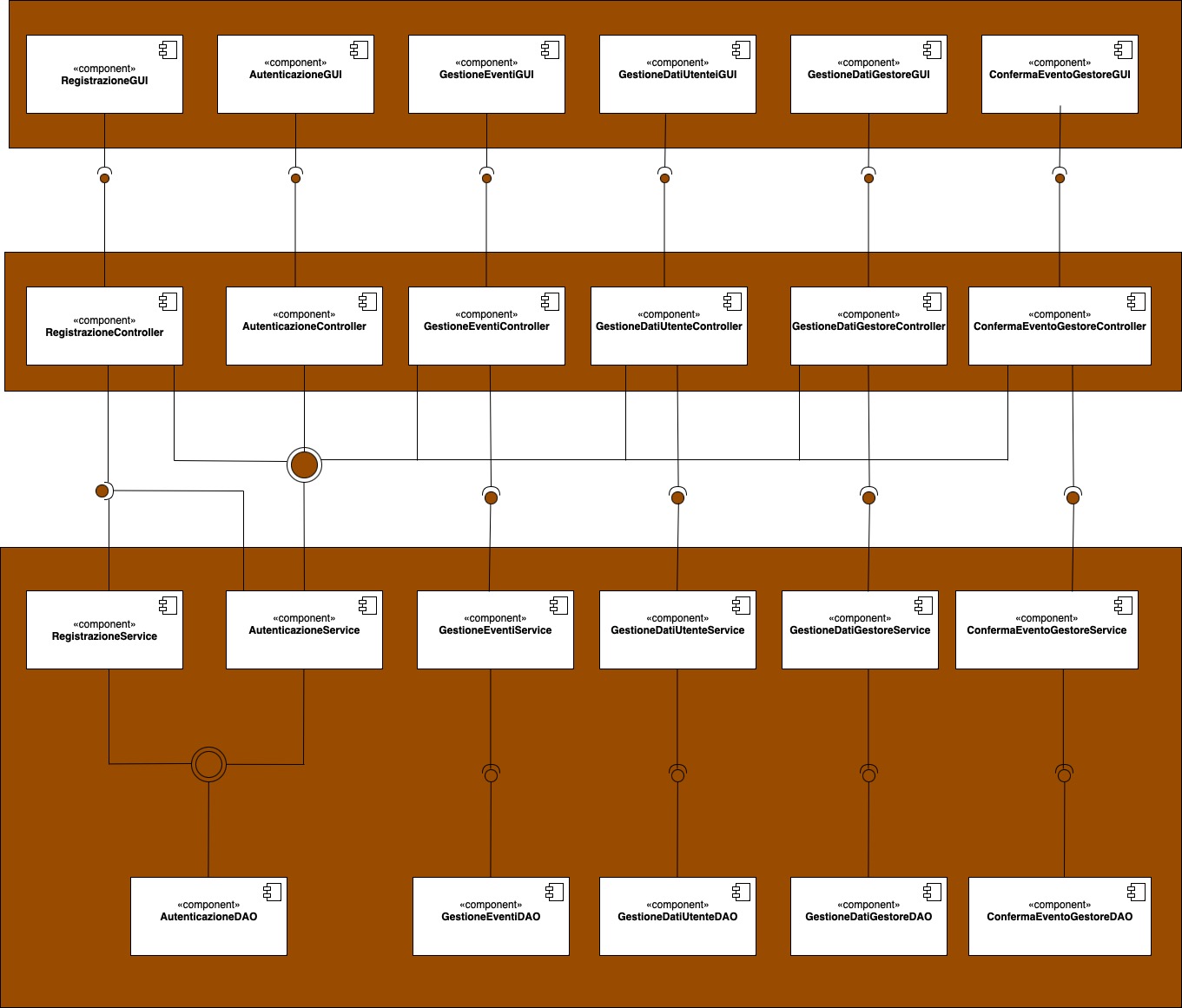
Di seguito sono mostrate le dipendenze tra i sottosistemi attraverso un Component Diagram.



## **DIAGRAMMA ARCHITETTURALE**

Di seguito una vista dettagliata di ciascun sottosistema evidenziando le componenti principale:

* **GUI**: Graphic User Interface, che contiene le varie view che saranno renderizzate per creare le pagine web da mostrare al cliente.
* **Controller**: si occupa della logica per il controllo del sistema.
* **Service**: si occupa della logica di business.
* **DAO**: Data Access Object, che si occupa di fornire accesso ai dati persistenti.

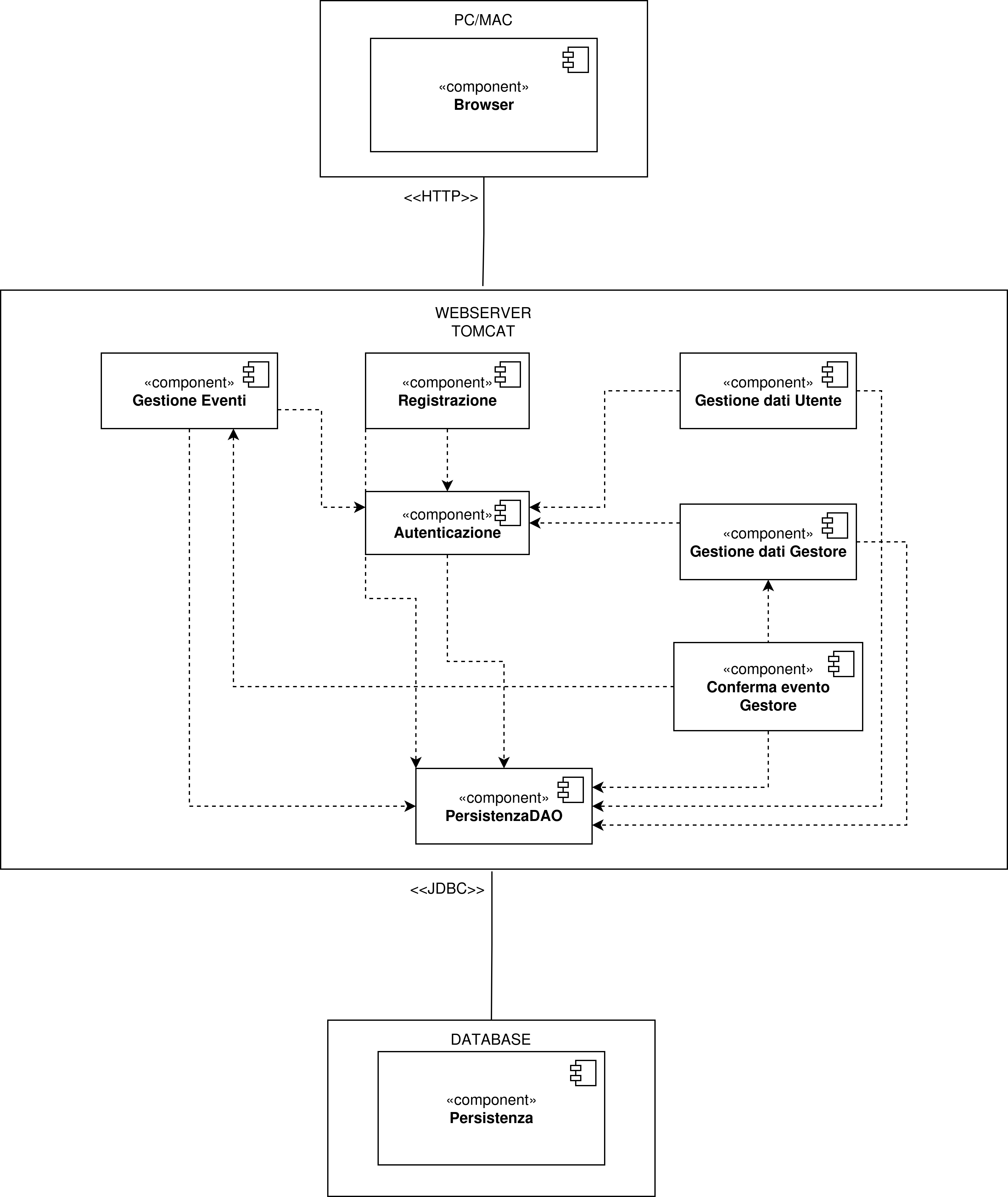


## 3.3 Mapping Hardware/Software

L’applicazione Web si basa su una piattaforma Hardware costituita da un server che risponde alle richieste effettuate dai client da una qualsiasi macchina con un browser installato ed una connessione a Internet.

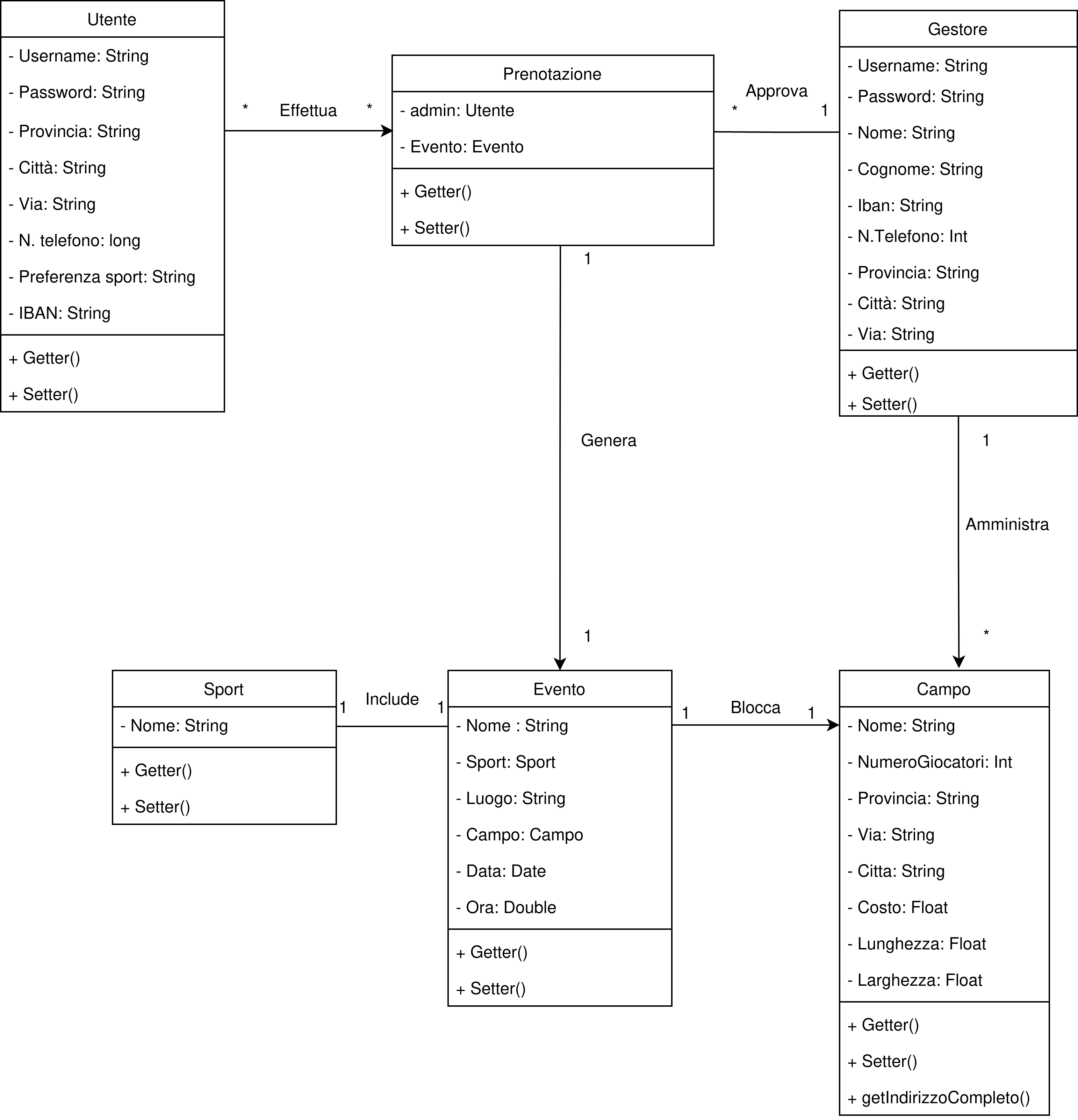
La Webapp non sarà distribuita e risiederà pertanto su un solo nodo.

Di seguito il mapping Hardware/Software mediante il Deployment Diagram.



## 3.4 Class Diagram ristrutturato

Di seguito il class diagram ristrutturato, si scegli di migliorare la leggibilità incapsulando i getter ed i setter in una sola riga.



## 

## 3.5 Gestione dei Dati Persistenti

Per la gestione del salvataggio dei dati persistenti si utilizzerà un database relazionale, al fine di gestire agevolmente l’accesso ai dati e di garantire la consistenza di questi attraverso **l’imposizione di vincoli di integrità**.

La **privatezza** e l'**affidabilità** dei dati sono garantite dalle caratteristiche stesse dei DBMS quali accesso protetto e copie ripristinabili dei dati.

Vista la precedente esperienze di tutti i membri del team, si è scelto di utilizzare MySQL come DBMS. In tale modo si potranno minimizzare gli errori relativi all’apprendimento di tecnologie nuove in tempi relativamente ristretti.

## 3.6 Controllo degli Accessi e Sicurezza

|  |  |  |  |
| --- | --- | --- | --- |
| ATTORI  =============  OGGETTI | OSPITE | UTENTE  REGISTRATO | GESTORE |
| REGISTRAZIONE | RegistrazioneForm  RegistrazioneButton  ConfermaRegistrazioneButton |  |  |
| AUTENTICAZIONE |  | Login  Logout  VisualizzaAreaUtente  ModificaDatiUtente  CancellazioneAccount | Login  Logout  VisualizzaAreaGestore  ModificaDatiGestore  ModificaDatiCampetti  CancellazioneAccount |
| GESTIONE EVENTO |  | VisualizzaBachecaEventiDisponibili  CercaEvento  CreaEvento  InvitaEvento  AderisciEvento  EliminaEvento  VisualizzaDatiEvento | VisualizzaRichieste  AccettaRichiesta  DeclinaRichiesta  VisualizzaDatiEvento |

## 

## 3.7 Controllo Globale del Software

Il flusso di eventi sarà gestito da un paradigma event-driven. Per ogni azione che l’utente desidera eseguire ci sarà un comando GUI da impartire, questo verrà gestito dal suo handler, che passerà il controllo alla corrispondente logica di controllo e a quella applicativa.

## 3.8 Condizioni Limite

Nel presente paragrafo verranno presentate le Boundary Conditions inerenti all’avvio, spegnimento e fallimento del sistema, oltre che alla gestione degli errori in fase di accesso ai dati persistenti.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificativo**  UCBC\_1 | | Avvio del Sistema | Data | 09-12-22 |
| Vers. | 0.1 |
| Autore | Antonio Renzullo |
| **Descrizione** | | Lo UC permette l’avvio del sistema | | |
| **Attore Principale** | | Utente | | |
| **Attori secondari** | | NA | | |
| **Entry Condition** | | L’utente effettua il login | | |
| **Exit Condition**  On success | | L’avvio avviene correttamente | | |
| **Exit Condition**  On failure | | Il sistema non viene avviato | | |
| **FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO** | | | | |
| 1. | Utente | L’utente entra nella sezione dedicata al login ed inserisce i propri dati. | | |
| 2. | Sistema | Il sistema verifica la correttezza dei dati inseriti e verifica la sanità dei dati persistenti, se tutto è corretto rende le sue funzionalità disponibili all’utente | | |
| **Scenario/Flusso di eventi Alternativo: Uno dei campi inseriti non risulta valido** | | | | |
| **2.a1** | Sistema | Il sistema segnala all’utente la non correttezza di uno dei campi | | |
| **2.a2** | Utente | Verifica nuovamente la correttezza dei campi | | |
| **2.a3** | Sistema | Esegue il passo 2 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificativo**  UCBC\_2 | | Spegnimento del Sistema | Data | 09-12-22 |
| Vers. | 0.1 |
| Autore | Antonio Renzullo |
| **Descrizione** | | Lo UC permette l’arresto del sistema | | |
| **Attore Principale** | | Utente | | |
| **Attori secondari** | | NA | | |
| **Entry Condition** | | L’utente effettua il login  AND il Sistema è stato precedentemente avviato  AND il Sistema non è ancora stato spento | | |
| **Exit Condition**  On success | | L’arresto avviene correttamente | | |
| **Exit Condition**  On failure | | Il sistema non viene spento | | |
| **FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO** | | | | |
| 1. | Utente | Invia un segnale di logout al Sistema | | |
| 2. | Sistema | Il sistema verifica che eventuali dati da salvare vengano resi persistenti, dopodichè termina l’esecuzione del sistema | | |
| **Scenario/Flusso di eventi Alternativo: Uno dei campi inseriti non risulta valido** | | | | |
| **2.a1** | Sistema | Il sistema segnala al lettore l'impossibilità di salvare i dati | | |
| **2.a2** | Utente | Attende qualche istante e riprova a fare il logout | | |
| **2.a3** | Sistema | Esegue il passo 2 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificativo**  UCBC\_3 | | Fallimento del sistema | Data | 09-12-22 |
| Vers. | 0.1 |
| Autore | Antonio Renzullo |
| **Descrizione** | | Lo UC definisce il comportamento del sistema in caso di fallimento | | |
| **Attore Principale** | | Utente | | |
| **Attori secondari** | | NA | | |
| **Entry Condition** | | La connessione viene terminata inaspettatamente | | |
| **Exit Condition**  On success | | I dati temporanei vengono resi persistenti ed il sistema viene riavviato | | |
| **Exit Condition**  On failure | | Il sistema non viene riavviato e vengono persi i dati non ancora resi persistenti | | |
| **FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO** | | | | |
| 1. | Utente | Include UCBC\_1 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificativo**  UCBC\_4 | | Errori di Accesso ai Dati Persistenti | Data | 09-12-22 |
| Vers. | 0.1 |
| Autore | Antonio Renzullo |
| **Descrizione** | | Lo UC definisce il comportamento del sistema qualora fosse impossibile accedere ai dati persistenti o questi risultassero corrotti | | |
| **Attore Principale** | | Utente | | |
| **Attori secondari** | | NA | | |
| **Entry Condition** | | Il sistema non può accedere ai dati persistenti  OR i dati persistenti risultano corrotti | | |
| **Exit Condition**  On success | | Il Sistema riprende il normale funzionamento | | |
| **Exit Condition**  On failure | | Il Sistema non riprende il normale funzionamento | | |
| **FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO** | | | | |
| 1. | Sistema | Notifica l’utente dell’impossibilità di accedere ai dati | | |
| 2. | Sistema | Cessa di processare eventuali richieste e risponde a tutte le richieste con un messaggio di errore | | |

# Servizi dei Sottosistemi

**Sottosistema Registrazione**

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| RegistrazioneUtente | Permette di registrarsi come utente | RegistrazioneService |
| RegistrazioneGestore | Permette di registrarsi come gestore | RegistrazioneService |

**Sottosistema Autenticazione**

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Login | Permette di effettuare l’accesso al sistema tramite le proprie credenziali | AutenticazioneService |
| Logout | Permette di disconnettersi dal sistema | AutenticazioneService |
| Cancellazione Account | Permette di eliminare i propri dati dal sistema | AutenticazioneService |

**Sottosistema Gestione Evento**

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Crea evento | Permette di creare un evento | GestioneEventiService |
| Cerca evento | Permette di cercare un evento dati determinati filtri | GestioneEventiService |
| Elimina evento | Permette di eliminare un evento | GestioneEventiService |
| Partecipa evento | Permette di partecipare ad un evento | GestioneEventiService |
| Visualizza evento | Permette di visualizzare le informazioni relative ad un evento | GestioneEventiService |

**Sottosistema Gestione dati Utente**

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Visualizza area Utente | Permette di visualizzare l’area utente | GestioneUtenteService |
| Modifica dati Utente | Permette di modificare le informazioni personali | GestioneUtenteService |
| Visualizza Eventi sottoscritti | Permette di visualizzare la lista di eventi ai quali si intende partecipare | GestioneUtenteService |

**Sottosistema Gestione dati Gestore**

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Visualizza area Gestore | Permette di visualizzare l’area gestore | GestioneGestoreService |
| Modifica dati Gestore | Permette di modificare le informazioni personali | GestioneGesotreService |
| Modifica dati Campi | Permette di modificare le informazioni dei campi posseduti | GestioneGesotreService |
| Visualizza Eventi in attesa | Permette di visualizzare la lista di eventi che potranno essere confermati oppure declinati | GestioneGesotreService |

**Sottosistema Conferma Evento Gestore**

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Accetta evento | Permette di accettare la richiesta e di incassare il pagamento | GestioneRichiesteService |
| Declina evento | Permette di rifiutare la richiesta | GestioneRichiesteService |