AVAFIELD

# System Design Document

| | |
|---|---|
| Riferimento | |
| Versione | 1.0 |
| Data | 1/12/2021 |
| Destinatario | Prof. Gravino |
| Presentato da | **Team AVA** |
| Approvato da | |

| DATA | Versione | Cambiamenti | Autori |
|---|---|---|---|
| **01/12/2021** | 0.1 | Prima realizzazione SDD | [tutti] |
| **02/12/2021** | 0.2 | Aggiunti capitoli 1,2,3 | [Alessio, Vincenzo] |
| **03/12/2021** | 0.3 | Aggiunti sottosistemi | [tutti] |
| **04/12/2021** | 0.4 | Raffinamenti al capitolo 3 | [tutti] |
| **06/12/2021** | 1.0 | Revisione | [tutti] |
| **21/12/2021** | 1.1 | Modificati Sottosistemi e Servizi | [Vincenzo, Alessandro] |

# Sommario

**1. Introduzione** **3**

**1.1 Obiettivi del sistema** **3**

**1.2 Design Goals** **4**

**1.2.1. Criteri di performance** 4

**1.2.2. Criteri di dependability** 4

**1.2.3. Criteri di costo** 5

**1.2.4. Criteri di manutenzione** 5

**1.2.5. Criteri di usabilità** 6

**1.3. Design Trade-off** **6**

**1.4. Definizioni, acronimi e abbreviazioni** **7**

**1.5. Riferimenti** **7**

**1.6. Panoramica** **7**

**2. Architettura del Sistema corrente** **8**

**3. Architettura del Sistema proposto** **8**

**3.1 Panoramica** 8

**3.2 Decomposizione in sottosistemi** 8

**3.3 Mapping hardware/software** 10

**3.4 Gestione dati persistenti** 11

**3.5 Controllo degli accessi e sicurezza** 12

**3.6 Controllo flusso globale del sistema** 12

**3.7 Condizione limite** 13

**4. Servizi dei Sottosistemi** **14**

# Team members

| Nome | Ruolo nel progetto | Email |
|---|---|---|
| Vincenzopio Amendola | Team Member | v.amendola15@studenti.unisa.it |
| Alessio Alfieri | Team Member | a.alfieri33@studenti.unisa.it |
| Alessandro Rusciano | Team Member | a.rusciano1@studenti.unisa.it |

## 1. Introduzione

### 1.1 Obiettivi del sistema

Il sistema da realizzare ha come obiettivo la gestione di eventi sportivi (con relative prenotazioni da parte degli utenti) ospitati all'interno delle strutture presenti sul sistema stesso.

Ulteriore obiettivo del sistema proposto è quello di rendere la prenotazione degli eventi più efficiente sfruttando un modulo di IA che consiglia all'utente eventi sportivi in base a una autovalutazione, precedentemente inserita dall'utente.
Gli attuali sistemi risultano lenti, macchinosi e talvolta di difficile comprensione: ciò scoraggia gli utenti, portandoli a preferire un approccio più semplice e diretto con le strutture.

Il sistema proposto è una web application a cui avranno accesso i seguenti utenti definiti in fase di analisi:

- **Atleta**;
- **Ospite;**
- **Admin**;

Principalmente, il sistema si occuperà della gestione persistente dei dati degli Atleti, degli Eventi, delle Strutture e delle recensioni associate. Il sistema dovrà consentire il login per gli Atleti e gli Admin del sistema. Naturalmente, per tutti gli Ospiti sarà possibile compilare l'apposito form per registrarsi alla piattaforma.
Il sistema dovrà consentire, da parte degli Ospiti, di:

- Registrarsi;
- Effettuare ricerche delle strutture e dei campi;
- Visualizzare le recensioni;

Il sistema dovrà consentire, da parte degli Atleti, di:

- Effettuare il login con username e password;
- Tenere traccia del proprio profilo;
- Aggiungere una recensione;
- Creare un nuovo evento;
- Partecipare a un evento già esistente;
- Contattare l'assistenza;

Il sistema dovrà consentire, da parte degli Admin, di:

- Aggiungere una nuova struttura;
- Modificare una struttura;
- Eliminare una struttura già esistente;

Infine, il sistema dovrà essere in grado di presentare gli eventi per ogni struttura.

## 1.2 Design Goals

I Design Goals sono organizzati in cinque categorie: Performance, Dependability, Cost, Maintenance, End User Criteria. I Design Goals identificati nel nostro sistema sono i seguenti:

### 1.2.1. Criteri di performance

- **Tempo di risposta:**
  Il sistema deve elaborare in tempi minimi una qualsiasi sottomissione effettuata dall'utente, in qualsiasi momento della giornata essa venga effettuata.

- **Memoria:**
  Per il corretto funzionamento, è richiesto uno spazio di archiviazione minimo sia in termini di dimensioni della piattaforma, sia per le dimensioni della base di dati, in quanto non è previsto un utilizzo massiccio di una memoria di massa; le immagini di presentazione delle strutture e altri possibili file di fatto, non hanno dimensioni considerevoli.

### 1.2.2. Criteri di dependability

- **Robustezza:**
  Il sistema dovrà esser in grado di sopravvivere anche ad input non validi immessi dall'utente, o a condizioni precarie dell'environment.

- **Affidabilità:**
  Il sistema deve eseguire le funzioni richieste sotto determinate condizioni per un periodo di tempo prestabilito.

- **Disponibilità:**
  Dopo aver effettuato il deploy del server, il sistema dovrà essere disponibile per un periodo significativo di tempo, che comprende qualsiasi periodo della giornata, salvo eventuali manutenzioni del server.

- **Tolleranza ai guasti:**
  Il sistema deve garantire la capacità di operare anche sotto condizioni di errore dovute, ad esempio, ad un sovraccarico di informazioni nel database.

- **Sicurezza:**
  L'accesso al sistema è previsto mediante utilizzo di username e password, di cui dovrà essere garantita la resistenza ad attacchi di malintenzionati. Per ogni tipologia di utente dovranno essere gestiti controlli, onde evitare utilizzi da parte di utenti non autorizzati.

### 1.2.3. Criteri di costo

- **Costo di sviluppo:**
  È stimato un costo complessivo di 150 ore per la progettazione e lo sviluppo del sistema (50 per ogni project member).

### 1.2.4. Criteri di manutenzione

- **Estensibilità:**
  Il sistema dovrà consentire l'aggiunta di nuove funzionalità o nuove classi a sviluppatori futuri, in base alle esigenze del cliente.

- **Modificabilità:**
  Il sistema dovrà consentire, mediante consultazione della documentazione, la modifica di funzionalità già presenti.

- **Adattabilità:**
  È previsto lo sviluppo del sistema esclusivamente per la gestione di campi sportivi, ma il software non include vere e proprie specifiche di tale ambito, per cui sarà possibile adattarlo a differenti domini applicativi.

- **Portabilità:**
  Il sistema verrà sviluppato come web application, testato principalmente sul browser Google Chrome. Tuttavia, gli standard moderni consentono al sistema di poter funzionare correttamente anche su browser differenti.

- **Tracciabilità dei requisiti:**
  Una matrice di tracciabilità consentirà il mapping semplificato del codice nei requisiti specifici.

### 1.2.5. Criteri di usabilità

- **Usabilità:**
  Il sistema sarà facile da comprendere, in modo tale da consentire all'utente di imparare ad operare agilmente, grazie ad un'interfaccia user-friendly.

- **Utilità:**
  Il sistema sarà estremamente utile, in quanto velocizza considerevolmente il processo di creazione degli eventi sportivi.

### 1.3. Design Trade-off

**Functionality vs Usability:** Il sistema deve consentire una maggiore usabilità a discapito delle funzionalità previste nel RAD. Chiaramente, in base alle esigenze del cliente verranno mantenute ed implementate le funzionalità in base alle priorità assegnatagli.

**Cost vs Robustness:** Il sistema verrà sviluppato favorendo la robustezza a discapito dei costi. Alla base di tale decisione vi è l'esigenza che il sistema sia estremamente robusto, in quanto il corretto funzionamento del sistema (anche se soggetto ad input errati o a condizioni precarie) ha un impatto concreto sugli eventi, impedendo eventualmente il corretto svolgimento.

**Efficiency vs Portability:** L'efficienza dovrà essere un punto chiave dello sviluppo del sistema. Sebbene il sistema sia "basato" sulla portabilità, nel caso in cui sarà necessario effettuare un trade-off, verrà valutata maggiormente l'efficienza, in quanto è più importante che il sistema risponda correttamente alle richieste dell'utente anziché visualizzare, ad esempio, correttamente le pagine.

**Rapid development vs Functionality:** In base al criterio di *costo di sviluppo* ed alla deadline prestabilita (ed alle priorità assegnate ai requisiti funzionali del sistema), verrà considerato maggiormente il rapido sviluppo della piattaforma a discapito di alcune funzionalità ritenute non essenziali dal cliente.

**Cost vs Reusability:** Il sistema da sviluppare essenzialmente non dispone di una versione precedente, per cui non esistono vere e proprie componenti legacy riutilizzabili. Si cercherà di mantenere i costi di produzione entro un certo limite e di riutilizzare in futuro il più possibile le componenti ad-hoc realizzate.

**Response Time vs Reliability:** Il sistema sarà implementato in modo tale da preferire l'affidabilità al tempo di risposta, onde garantire un controllo più accurato dei dati in input a discapito del tempo di risposta del sistema.

### 1.4. Definizioni, acronimi e abbreviazioni

**RAD:** Requirements Analysis Document.

**SDD:** System Design Document.

**USER-FRIENDLY:** Letteralmente "amichevole per l'utente", di facile utilizzo anche per chi non è esperto.

**DB:** DataBase, ovvero "Base di Dati".

**MySQL:** È un RDBMS Open Source basato sul linguaggio SQL, composto da un client a riga di comando e un server.

**DBMS:** DataBase Management System.

**SQL:** Structured Query Language; linguaggio standardizzato per database basati sul modello relazionale (RDBMS) progettato per: creare e modificare schemi di database.

**RESPONSIVE:** Tecnica di web design per la realizzazione di siti in grado di adattarsi graficamente in modo automatico al dispositivo coi quali vengono visualizzati, riducendo al minimo la necessità dell'utente di ridimensionare e scorrere i contenuti.

### 1.5. Riferimenti

- Slide del corso, presenti sulla piattaforma e-learning.
- *Libro di testo*: Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition. *Autori:* Bernd Bruegge & Allen H. Dutoit.

### 1.6. Panoramica

**Capitolo 1:** Contiene una panoramica del sistema software, dei suoi design goals. Sono presenti definizioni, acronimi e abbreviazioni e riferimenti ad altri documenti al fine di produrre una documentazione che possa essere chiara.

**Capitolo 2:** Vengono rese esplicite le funzionalità del sistema corrente, elencandone le criticità che hanno portato alla creazione di un nuovo sistema attualmente in progettazione.

**Capitolo 3:** Viene presentata l'architettura del sistema software proposto. Più dettagliatamente, viene descritta la decomposizione del sistema in sottosistemi e le responsabilità di ognuno di essi; inoltre, viene presentato il mapping hardware/software, la gestione dei dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, ed infine le condizioni limite.

**Capitolo 4:** Vengono presentati i servizi dei sottosistemi.

# 2. Architettura del Sistema corrente

Attualmente esistono delle piattaforme informatiche atte alla gestione di eventi sportivi ben fatte ma presentano lacune dal punto di vista delle affiliazioni di strutture su tutto il territorio nazionale. Infatti in molti punti del sud Italia non ha una copertura vasta.

# 3. Architettura del Sistema proposto

## 3.1 Panoramica

Il sistema proposto è basato sullo stile architetturale Three Tier, il motivo della presente scelta è che tale architettura è perfetta per lo sviluppo di web application come il nostro sistema, poiché la separazione della logica di presentazione da quella di elaborazione, migliora una serie di qualità, tra le quali:

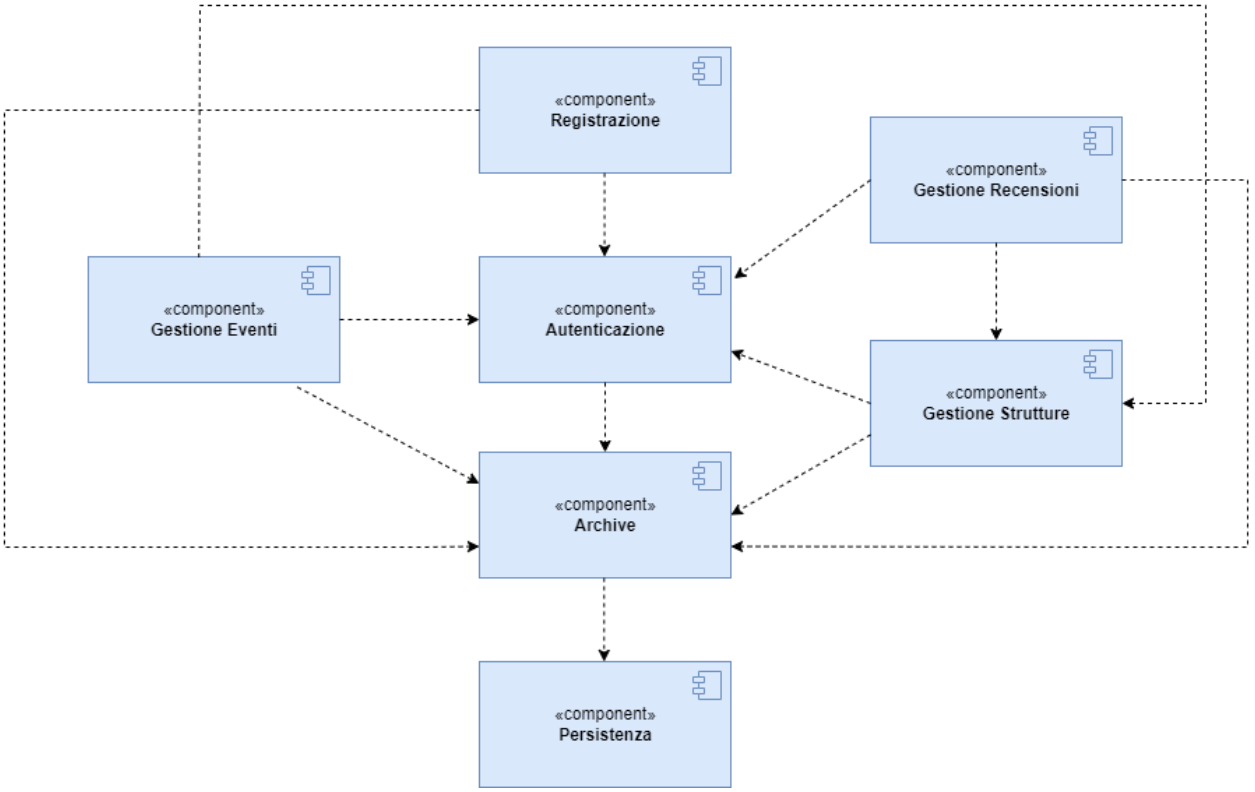
- Leggibilità
- Manutenzione
- Riuso

Il sistema da noi proposto permette la facile prenotazione presso strutture ben distribuite sul territorio. Gli utenti della piattaforma sono classificati in tre categorie: Admin, Atleta e Ospite. Di questi l'Atleta accede alla propria area utente ed ha la possibilità di prenotare e creare un nuovo Evento; L'Ospite ha la possibilità in primo luogo di registrarsi, di navigare la piattaforma, prendere visione degli Eventi attivi e delle strutture presenti.

## 3.2 Decomposizione in sottosistemi

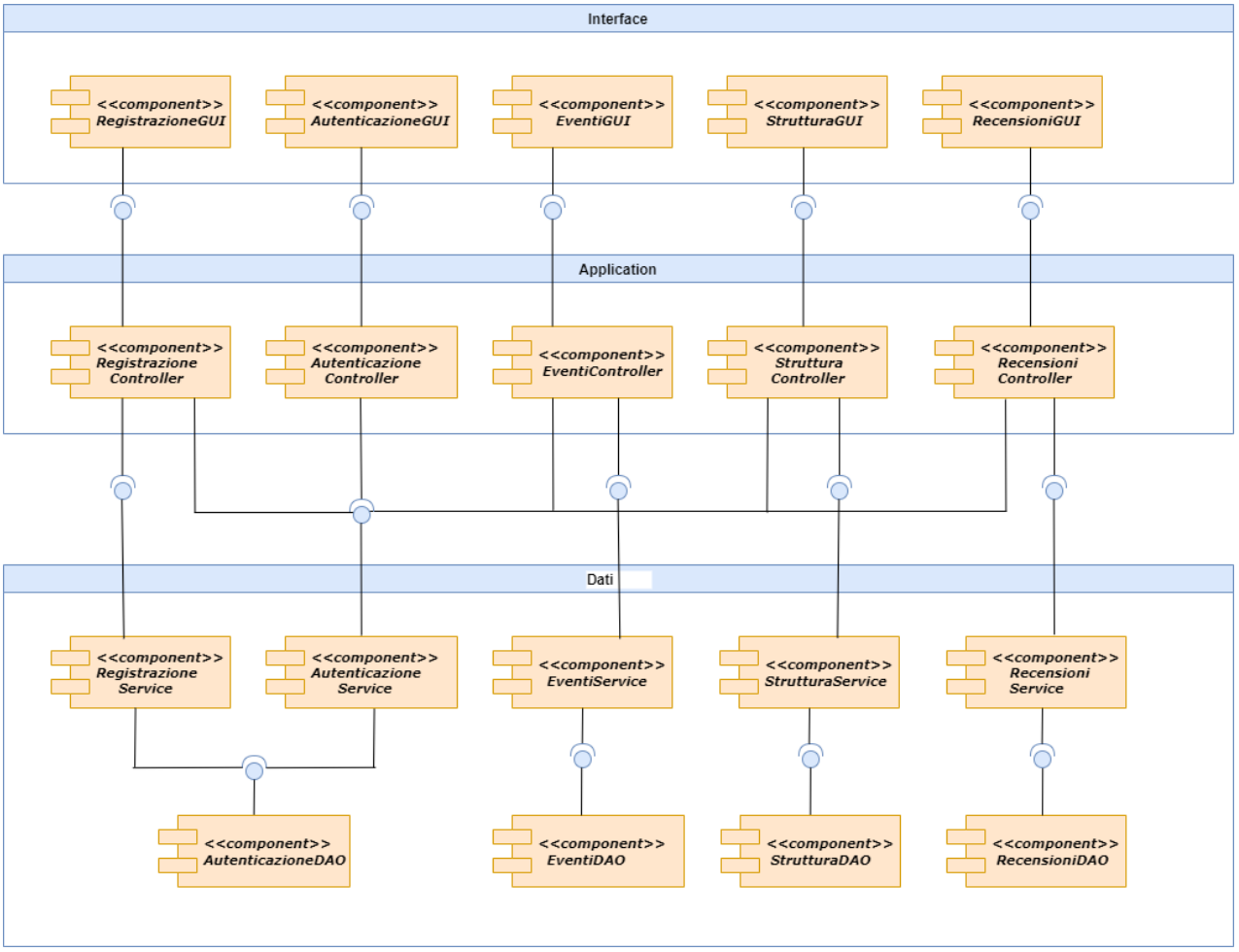
Il sottosistemi individuati sono:

- **Registrazione:** si occupa di gestire la registrazione dell'utente Ospite.
- **Autenticazione:** è responsabile delle funzionalità di Login, Logout, visualizzazione area utente e la modifica dati account da parte degli Atleti e degli Admin.
- **Gestione Eventi:** si occupa delle funzioni riguardanti la creazione, la prenotazione e la visualizzazione degli Eventi da parte degli Atleti.
- **Gestione Strutture:** si occupa delle funzioni riguardanti la creazione, visualizzazione e modifica delle Strutture da parte dell'Admin.
- **Gestione Recensioni:** si occupa di gestire le funzioni riguardanti la creazione, visualizzazione delle Recensioni da parte degli Atleti.
- **Archive:** si interpone tra i vari sottosistemi e il sottosistema di Persistenza.
- **Persistenza:** si occupa di gestire la persistenza dei dati con un database.



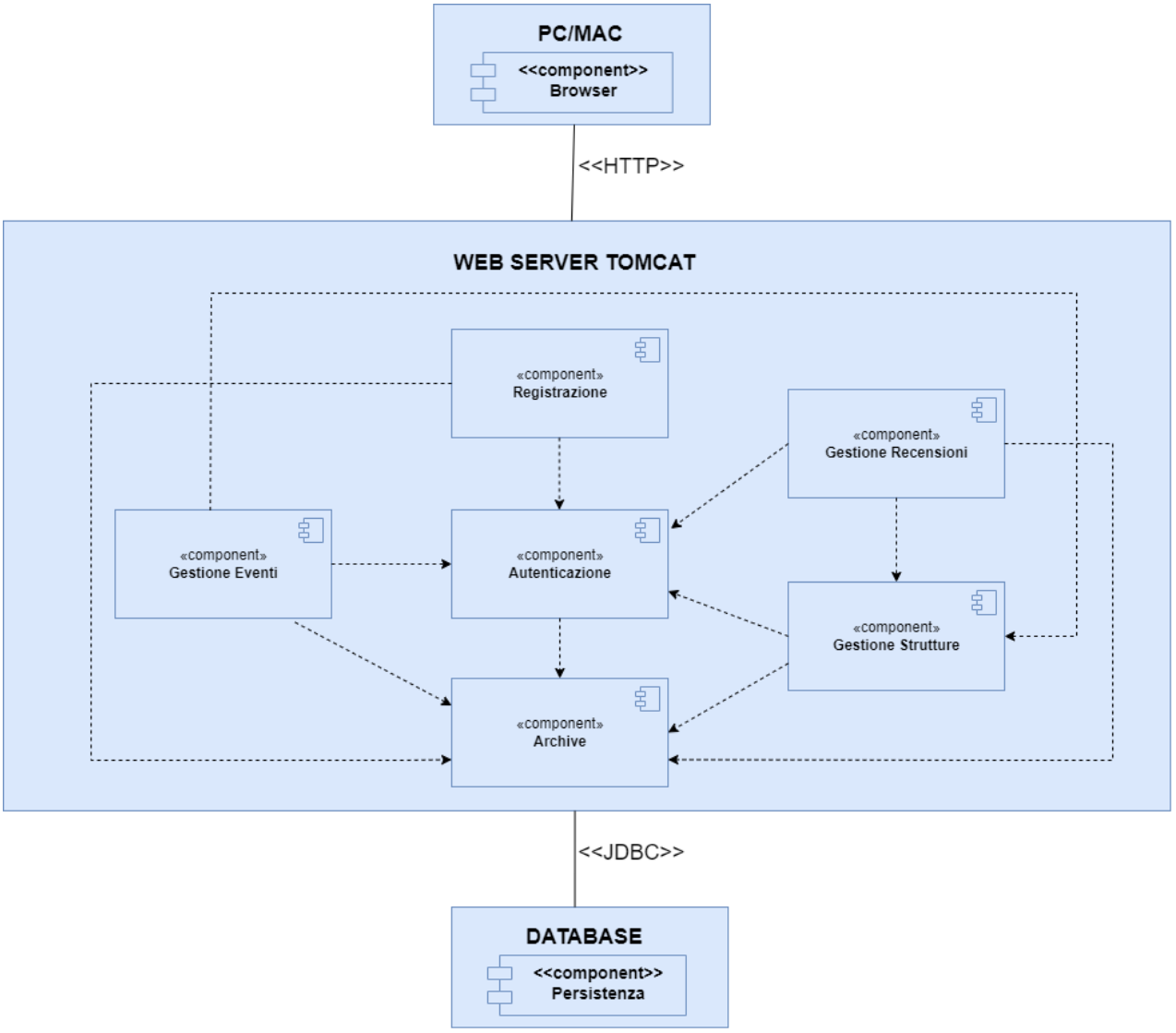

Di seguito una vista dettagliata di ciascun sottosistema evidenziando le componenti principale:

- **GUI:** Graphic User Interface, che contiene le varie view che saranno renderizzate per creare le pagine web da mostrare all'utente.
- **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.



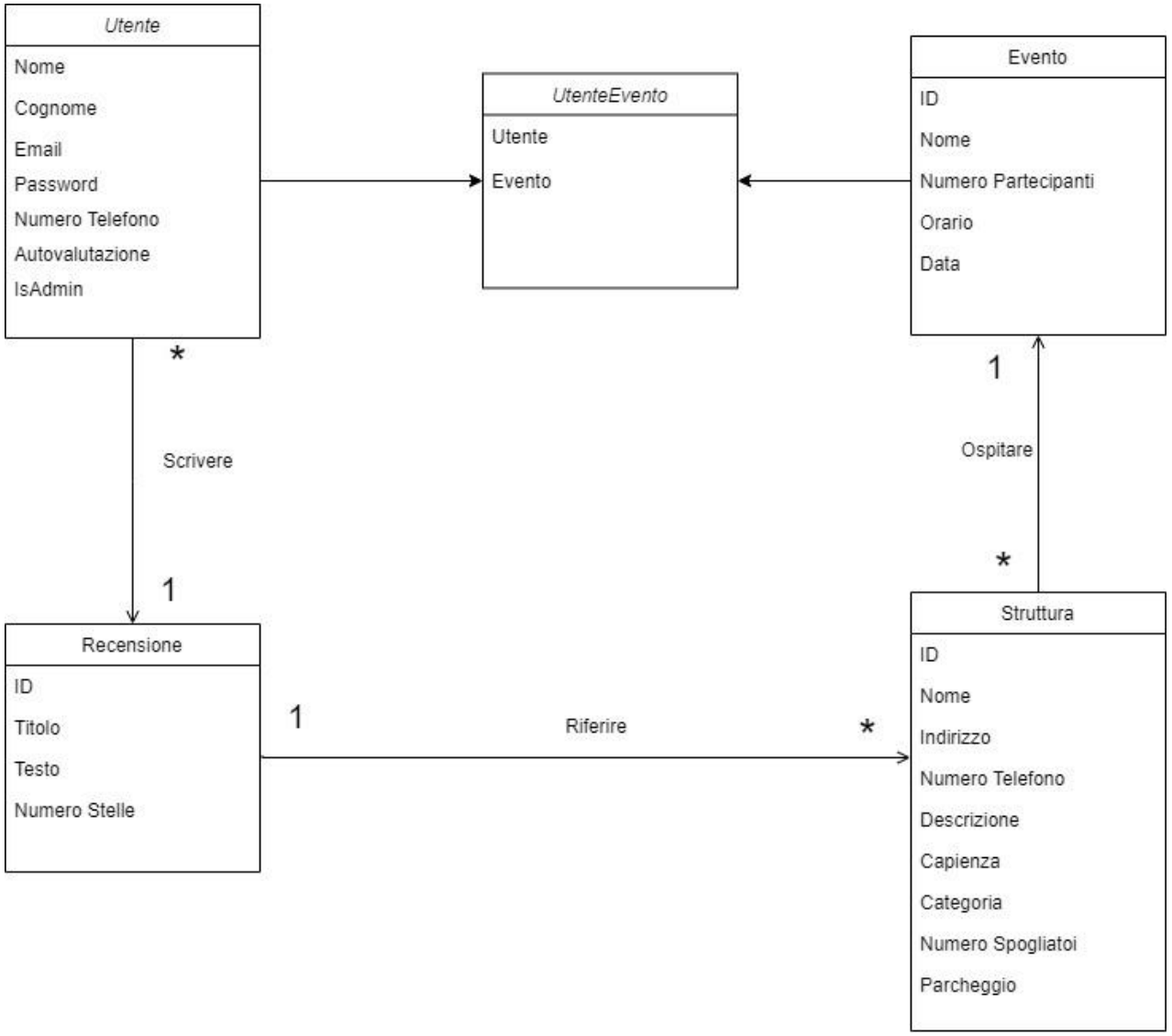

### 3.3 Mapping hardware/software

L' applicazione web che verrà sviluppata si basa su una piattaforma hardware costituita da un server che risponde alle richieste effettuate dai clienti da una qualsiasi macchina con un browser ed una connessione ad Internet.
Essendo che il nostro sistema è una web application e risiede su un web server, e che si basa su un'architettura non distribuita, risiede su un solo nodo.
Di seguito un UML deployment diagram che descrive il mapping hardware/software.




### 3.4 Gestione dati persistenti

Per la memorizzazione dei dati persistenti si è deciso di utilizzare un RDBMS (Relational Database Management System) dal momento che fornisce un accesso ai dati veloce e permette di collegare entità differenti in modo semplice. Fornisce un accesso concorrente ai dati mantenendo la coerenza dei dati anche in condizione di multiutenza e soprattutto, incorpora un meccanismo di permessi, che rende l'accesso a dati sensibili protetto e quindi utenti con operazioni diverse possono accedere a sezioni diverse della base di dati. Di seguito si allega lo schema concettuale e il mapping logico del database che si intende utilizzare per la realizzazione del sistema.


Utente
Nome
Cognome
Email
Password
Numero Telefono
Autovalutazione
IsAdmin
UtenteEvento
Utente
Evento
Evento
ID
Nome
Numero Partecipanti
Orario
Data
*
Scrivere
1
1
Ospitare
*
Recensione
ID
Titolo
Testo
Numero Stelle
1
Riferire
*
Struttura
ID
Nome
Indirizzo
Numero Telefono
Descrizione
Capienza
Categoria
Numero Spogliatoi
Parcheggio

### 3.5 Controllo degli accessi e sicurezza

Il sistema prevede un controllo degli accessi basato su autenticazione tramite credenziali per l'interazione con esso. Di seguito viene costruita una matrice degli accessi che rappresenta le operazioni concesse per ogni utente (per cui ovviamente è prevista l'autenticazione).

| **Oggetto**<br>**Attore** | **Registrazione** | **Autenticazione** | **Gestione Eventi** | **Gestione Strutture** | **Gestione Recensioni** |
|---|---|---|---|---|---|
| **Ospite** | *Registrazione* | - | *VisualizzaEventi* | *VisualizzaStrutture* | *VisualizzaRecensioni* |
| **Atleta** | - | *Login*<br>*Logout*<br>*VisualizzaAreaUtente*<br>*ModificaDati*<br>*CancellazioneAccount* | *CreaEvento*<br>*EliminaEvento*<br>*VisuallizzaEventi*<br>*PartecipaEvento* | *VisualizzaStrutture* | *CreaRecensione*<br>*EliminaRecensione*<br>*VisualizzaRecensioni* |
| **Admin** | - | *Login*<br>*Logout*<br>*VisualizzaAreaUtente*<br>*ModificaDati*<br>*CancellazioneAccount* | *VisualizzaEventi* | *VisualizzaStrutture*<br>*InserisciStruttura*<br>*ModificaStruttura*<br>*EliminaStruttura* | *VisualizzaRecensioni* |

### 3.6 Controllo flusso globale del sistema

Il sistema in realizzazione richiede per sua natura un livello sostanzialmente molto elevato di interazione con l'utente (sia esso un Atleta, Ospite o l'Admin). Non prevediamo inoltre un numero elevato di operazioni che non richiedono supervisione. Si può affermare che il sistema possa essere definito come "event-driven" in modo tale da semplificare l'approccio con l'utilizzatore. Si prevede, inoltre, data la particolarità del dominio del problema, di dover gestire in maniera concorrente interazioni del sistema con più utenti.

### 3.7 Condizione limite

**Avvio del sistema.** Al primo avvio, il sistema necessita di un web server che fornisca il servizio di accesso ad un database MySQL per la gestione dei dati persistenti. Quando un utente accede al sistema, gli verrà presentata una pagina di benvenuto dove sarà disponibile un pulsante per l'operazione di login e registrazione, uno per la visione delle strutture e uno per la creazione di un nuovo evento.

**Terminazione.** Alla chiusura dell'applicazione, il sistema termina con un logout automatico derivato dalla terminazione della sessione utente. Il server dovrà essere terminato manualmente dall'amministratore del sistema, dopodiché nessun client potrà connettersi alla piattaforma.

**Fallimento.** Si possono individuare diverse situazioni di fallimento:
Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione del server, sarà previsto un generatore di emergenza che si attiverà nell'istante in cui l'alimentazione verrà a mancare.
Un altro caso di fallimento potrebbe derivare dal software stesso che causa una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione. In questo caso non si prevedono politiche di recupero specifiche, se non il riavvio dell'intero sistema nel caso di errori fatali che ne compromettano il normale utilizzo.

## 4. Servizi dei Sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema precedentemente elencati.

### Sottosistema Registrazione

| Servizio | Descrizione | Interfaccia |
|---|---|---|
| **Registrazione** | Questa funzionalità permette di registrarsi sulla piattaforma come atleta. | RegistrazioneService |

### Sottosistema Autenticazione

| Servizio | Descrizione | Interfaccia |
|---|---|---|
| **Login** | Questa funzionalità permette di effettuare l'accesso al sistema tramite le proprie credenziali per sfruttare tutte le funzionalità che offre. | AutenticazioneService |
| **Logout** | Questa funzionalità permette di disconnettersi dal sistema. | AutenticazioneService |
| **VisualizzaAreaUtente** | Permette di visualizzare i dati relativi alla propria area utente. | AutenticazioneService |
| **ModificaDati** | Permette di modificare i dati relativi alla propria area utente. | AutenticazioneService |
| **CancellazioneAccount** | Permette di cancellare il proprio account sulla piattaforma. | AutenticazioneService |

### Sottosistema Gestione Eventi

| Servizio | Descrizione | Interfaccia |
|---|---|---|
| **CreaEvento** | Il servizio permette di creare un evento. | EventiService |
| **EliminaEvento** | Il servizio permette di eliminare un evento se l'Atleta è il creatore. | EventiService |
| **ModificaEvento** | Il servizio permette di modificare le informazioni relative ad un evento. | EventiService |
| **VisualizzaEventi** | Il servizio permette di visualizzare le informazioni relative agli eventi. | EventiService |
| **PartecipaEvento** | Il servizio permette di partecipare ad un evento. | EventiService |

### Sottosistema Gestione Strutture

| Servizio | Descrizione | Interfaccia |
|---|---|---|
| **InserisciStruttura** | Il servizio permette di inserire una struttura. | StrutturaService |
| **EliminaStruttura** | Il servizio permette di eliminare una struttura. | StrutturaService |
| **ModificaStruttura** | Il servizio permette di modificare le informazioni relative ad una struttura. | StrutturaService |
| **VisualizzaStrutture** | Il servizio permette di visualizzare le informazioni relative alle strutture. | StrutturaService |

**Sottosistema Gestione Recensioni**

| Servizio | Descrizione | Interfaccia |
|---|---|---|
| **CreaRecensioni** | Il servizio permette di creare una recensione. | RecensioniService |
| **EliminaRecensioni** | Il servizio permette di eliminare una recensione. | RecensioniService |
| **ModificaRecensioni** | Il servizio permette di modificare le informazioni relative ad una recensione. | RecensioniService |
| **VisualizzaRecensioni** | Il servizio permette di visualizzare le informazioni relative alle recensioni. | RecensioniService |