



# Università degli Studi di Salerno

Corso di Laurea in Informatica, a.a. 2022-23

Progetto del corso di Ingegneria del Software

prof. A. De Lucia

GitHub: /is-iPlay-22-23/



# iPlay

## > System Design Document

Versione 0.8.1

## Coordinatore del progetto

<i>Nome</i>	<i>Matricola</i>
Andrea De Lucia Manuel De Stefano	0512100000

## Partecipanti

<i>Nome</i>	<i>Matricola</i>
Vincenzopaolo Esposito	0512105337
Francesco Festa	0512106189
Samantha Iudici	0512110197

## Revision History

<i>Data</i>	<i>Versione</i>	<i>Descrizione</i>	<i>Autore</i>
10/11/2022	0.1.0	Creazione documento, creazione capitolo 1 (Introduzione con obiettivi del sistema, design goal, riferimenti e panoramica).	Vincenzopaolo Esposito Francesco Festa Samantha Iudici
15/11/2022	0.2.0	Creazione capitolo 2 e 3 (Architettura del sistema corrente e Architettura del sistema proposto); Creazione paragrafi: Panoramica, Decomposizione in sottosistemi e Mapping hardware e software.	Vincenzopaolo Esposito Francesco Festa Samantha Iudici
18/11/2022	0.3.0	Creazione paragrafi: Gestione dei dati persistenti e Controllo degli accessi e sicurezza.	Vincenzopaolo Esposito Francesco Festa Samantha Iudici
22/11/2022	0.4.0	Creazione paragrafo: Controllo del sistema globale.	Vincenzopaolo Esposito Francesco Festa Samantha Iudici
25/11/2022	0.5.0	Creazione capitolo 4 (Servizi dei sottosistemi).	Vincenzopaolo Esposito Francesco Festa Samantha Iudici
30/11/2022	0.5.1	Correzione Revision History usando il versionamento semantico.	Vincenzopaolo Esposito Francesco Festa Samantha Iudici
1/12/2022	0.6.0	Sostituzione degli schemi E-R nel paragrafo "Gestione dei dati persistenti" con il class diagram.	Vincenzopaolo Esposito Francesco Festa Samantha Iudici
9/12/2022	0.6.1	Correzione Revision History (shifting versione).	Vincenzopaolo Esposito Francesco Festa

			Samantha Iudici
13/12/2022	0.7.0	Aggiunto mail server nel paragrafo "Mapping Hardware/Software" e sostituito lo schema.	Vincenzopaolo Esposito Francesco Festa Samantha Iudici
16/12/2022	0.8.0	Sostituzione class diagram nel paragrafo 3.4 "Gestione dei dati persistenti".	Vincenzopaolo Esposito Francesco Festa Samantha Iudici
21/12/2022	0.8.1	Correzione class diagram nel paragrafo 3.4 "Gestione dei dati persistenti".	Vincenzopaolo Esposito Francesco Festa Samantha Iudici

## Indice

1. INTRODUZIONE.....	5
1.1 Obiettivi del sistema.....	5
1.2 Design Goal .....	5
1.2 Riferimenti .....	6
1.4 Panoramica .....	6
2. ARCHITETTURA DEL SISTEMA CORRENTE .....	6
3. ARCHITETTURA DEL SISTEMA PROPOSTO.....	6
3.1 Panoramica .....	6
3.2 Decomposizione in sottosistemi.....	7
3.3 Mapping Hardware/Software.....	8
3.4 Gestione dei dati persistenti .....	8
3.5. Controllo degli accessi e sicurezza .....	10
3.6. Controllo del sistema globale .....	11
4. SERVIZI DEI SOTTOSISTEMI.....	11

# 1. INTRODUZIONE

## 1.1 Obiettivi del sistema

Questo sistema nasce dall'esigenza di voler riunire sotto un unico portale le persone che praticano sport (sia a livello agonistico che amatoriale) e le strutture che sono adibite ad ospitare eventi di genere sportivo (in base poi allo sport che vi si può praticare) affinché possano essere organizzati eventi sportivi, in modo rapido e concreto.

Le figure principali che saranno presenti all'interno del sistema sono:

-L'utente: colui che ha intenzione di organizzare un evento sportivo in una determinata struttura e in una specifica data, decidendo se tale evento sarà pubblico o privato (decidendovi quindi il tipo di accesso).

-Il proprietario della struttura sportiva: colui che ospiterà gli eventi sportivi, atta a fornire le proprie informazioni sui servizi che mette a disposizione (campi da calcio, tennis, basket e ecc.), di recapito e disponibilità, oltre che al tipo di sport che vi si può praticare.

L'obiettivo principale del sistema è quello di rendere più semplice e veloce la creazione e di vari eventi sportivi agevolando la ricerca delle varie strutture sportive.

## 1.2 Design Goal

### Criteri di Performance

- Tempo di Risposta:  
Gli standard sui tempi di risposta devono essere:
  - 5 secondi per caricare una qualsiasi pagina.
  - 8 secondi per aggiungere/modificare/eliminare un elemento tra cui:  
Evento/campo/utente/fascia oraria.

### Criteri di Affidabilità

- Robustezza:  
Il sistema sarà in grado di mostrare alert di errori dovuti ad una non corretta immissione dei dati da parte degli utenti.
- Tolleranza ai guasti:  
In caso di problematiche lato utente, iPlay garantirà il salvataggio delle modifiche effettuate in qualsiasi momento.
- Sicurezza:  
Il sistema deve prevedere contromisure per evitare attacchi SQL injection e i dati sensibili quali le credenziali di accesso devono essere codificate in caso di accessi non desiderati ai database di gestione.

### Criteri di Manutenzione

- Manutenzione:  
iPlay sarà facilmente modificabile a seguito della distribuzione e vi sarà la possibilità di estenderlo a nuovi scenari.

### Criteri Utenti Finali

- Usabilità:  
La piattaforma sarà responsive, così da poter garantire la visione su dispositivi con

risoluzione diverse. Il sistema sarà costruito utilizzando il concetto del Flat Design Website e, inoltre, avrà un'interfaccia minimal per permettere all'utente di concentrarsi sulle funzionalità del sito.

## 1.2 Riferimenti

Problem Statement: Requisiti non funzionali

Requirements Analysis Document: Use Case

## 1.4 Panoramica

Al secondo punto del documento verrà presentata l'architettura del sistema corrente.

Al terzo punto sarà presentata l'architettura del sistema proposto, in cui si tratterà:

- Decomposizione del sistema in sottosistemi
- Mapping Hardware/Software
- Gestione dei dati persistenti
- Controllo degli accessi e sicurezza
- Controllo del sistema globale

Al quarto punto saranno presentati i servizi dei vari sottosistemi individuati.

## 2. ARCHITETTURA DEL SISTEMA CORRENTE

Siccome non esiste un software già in uso, è richiesto sviluppare un sistema nuovo da zero (Greenfield Engineering). I requisiti sono ottenuti dall'utente finale e dal cliente.

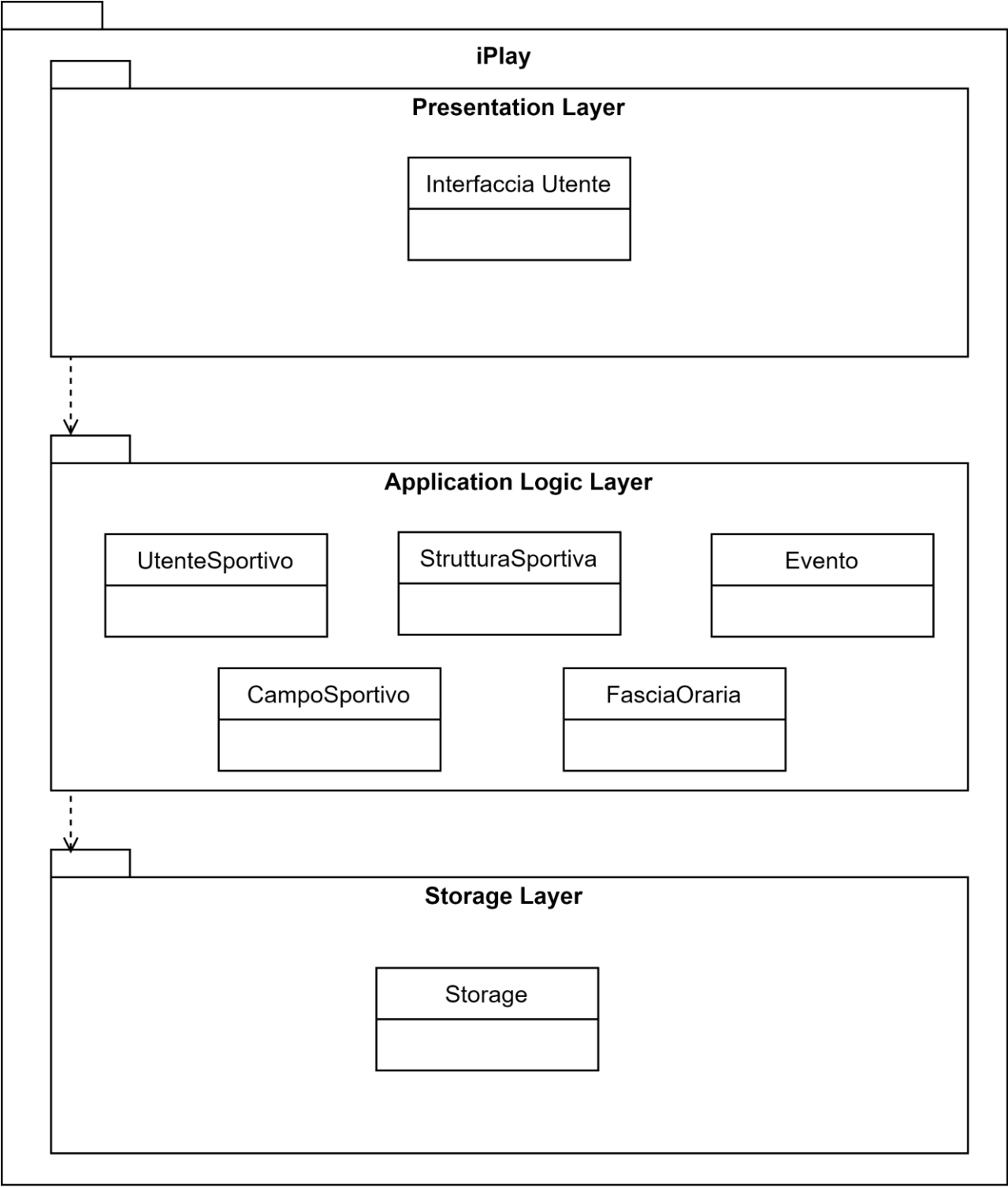
## 3. ARCHITETTURA DEL SISTEMA PROPOSTO

### 3.1 Panoramica

Il Sistema iPlay verrà realizzato usando l'architettura three-tier, divisa in: Presentation Layer, Application Logic Layer e Storage Layer. Ogni Layer si occupa rispettivamente di:

- Presentation Layer: include tutto ciò che riguarda l'interfaccia con l'utente (UI)
- Application Logic Layer: gestisce tutte le funzionalità del sistema
- Storage Layer: gestisce la memorizzazione dei dati tramite DBMS (MySQL)

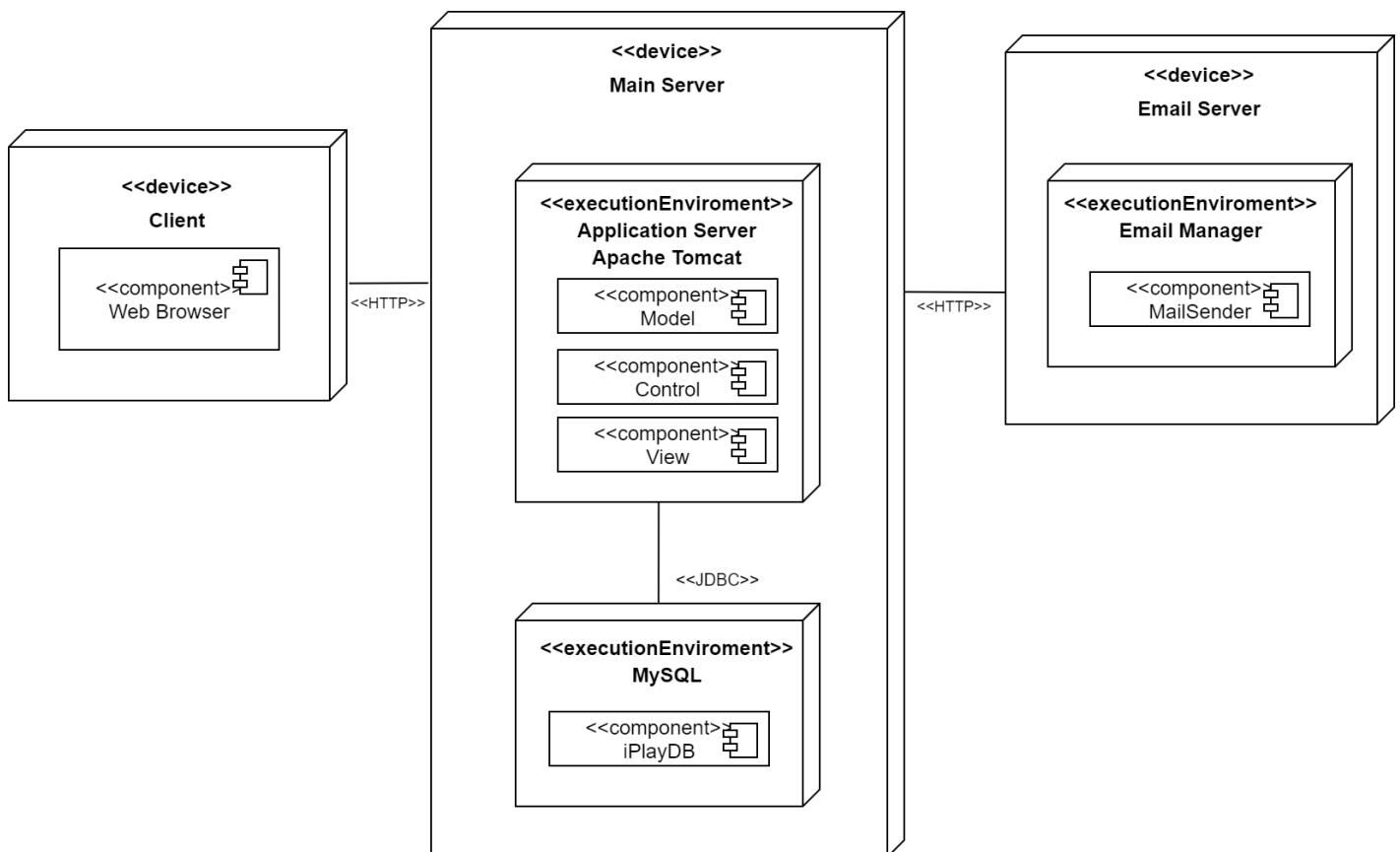
3.2 Decomposizione in sottosistemi



### 3.3 Mapping Hardware/Software

Il sistema che si intende sviluppare inizialmente prevederà l'utilizzo di due server.

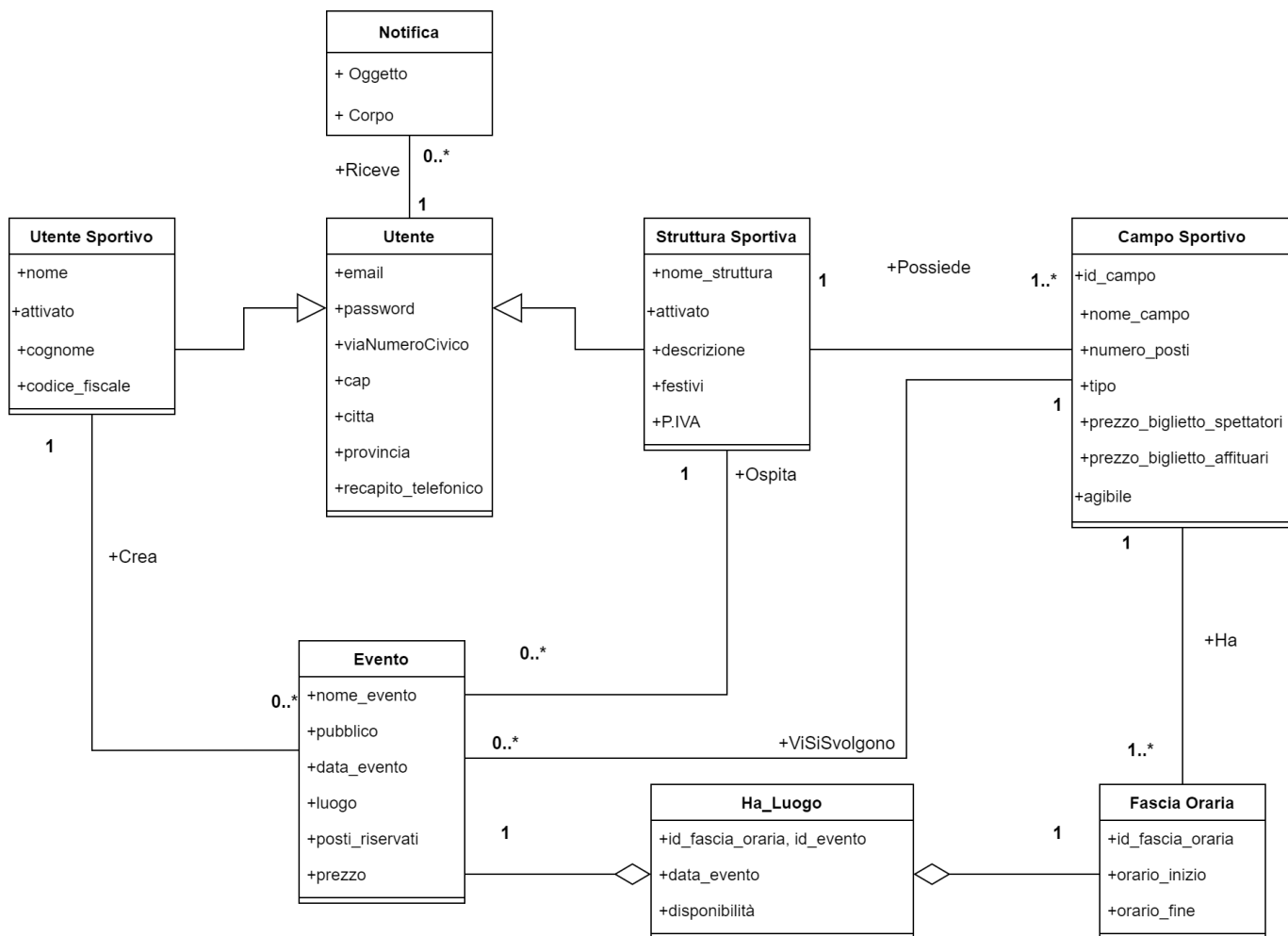
- Il Main Server su cui sarà installato un database DBMS Mysql per la gestione dei dati persistenti e un web server Apache Tomcat che servirà per la gestione delle: Java Servlet, JavaServer Pages, Java Expression Language e Java WebSocket. Il Client è un qualunque dispositivo che abbia un web browser installato. Il protocollo usato per trasferire i dati tra client e server sarà HTTP, mentre per le comunicazioni tra il server Database e quello Web si utilizzano le API JDBC.
- L'Email Server su cui sarà installato un email manager per l'invio di email di conferma, o di recupero password. Comunica con il Main Server tramite richieste HTTP.



### 3.4 Gestione dei dati persistenti

I dati persistenti saranno memorizzati attraverso il DBMS Mysql seguendo questo Class Diagram (Notazione UML)





Mapping:

UTENTI SPORTIVI: (codiceFiscale, email, passwordU, viaNumeroCivico, città, provincia, cap, recapitoTelefonico, nome, cognome, attivato)

PROPRIETARI STRUTTURE SPORTIVE: (pIva, email, passwordP, viaNumeroCivico, città, provincia, cap, recapitoTelefonico, nomeStruttura, festivi, descrizione, attivato)

CAMPI: (idCampo, numeroPosti, tipo, nomeCampo, prezzoBigliettoSpettatori, prezzoBigliettoAffittuari, agibile, pIva\*)

EVENTI: (idEvento, pubblico, nomeEvento, stato, postiRiservati, prezzoEvento, pIva\*, codiceFiscale\*, idCampo\*)

FASCE ORARIE: (idFasciaOraria, orarioInizio, orarioFine, idCampo\*)

HA\_LUOGO: ((idFasciaOraria, idEvento), disponibilit , dataEvento)

Durante l'analisi del sistema   stato deciso di utilizzare un DBMS relazionale (MYSQL) per le seguenti motivazioni:

- Accesso ai dati tramite un linguaggio universale (SQL).
- Accesso efficiente ai dati. Un DBMS ha molti modi per ottimizzare l'accesso all'informazione. La base di dati   solitamente memorizzata in memoria secondaria. Un DBMS permette di creare dei file ausiliari (indici) che permettono l'accesso veloce ai dati su disco

- Indipendenza dei dati. Un DBMS mantiene diversi livelli di astrazione dei dati, permettendo di accedere ai dati logici indipendentemente dalla loro rappresentazione fisica.
- Controllo della ridondanza dei dati. Ogni dato logico dovrebbe essere memorizzato in un solo posto nella base di dati. Avere più copie della stessa informazione ha i seguenti svantaggi:
  - o Maggior uso di memoria;
  - o Le modifiche della stessa informazione debbono essere effettuate diverse volte;
  - o Ci possono essere fenomeni di inconsistenza dei dati qualora gli aggiornamenti dei dati vengano eseguiti in modo indipendente;
- Atomicità delle operazioni. Un DBMS permette di effettuare sequenze di operazioni in modo atomico. Ciò significa che l'intera sequenza di operazioni viene eseguita con successo oppure nessuna di queste operazioni ha alcun effetto sui dati della base.
- Accesso concorrente ai dati. Un DBMS permette a più utenti di accedere contemporaneamente alla base di dati. Più utenti possono accedere nello stesso istante a dati diversi.
- Privatezza dei dati. Un DBMS permette un accesso protetto ai dati. Utenti diversi possono avere accesso a diverse porzioni della base di dati e possono essere abilitati a diverse operazioni su di esse.
- Affidabilità dei dati. Un DBMS offre dei metodi per salvare copie dei dati (backup) e per ripristinare lo stato della base di dati in caso di guasti software e hardware (recovery).

Per effettuare la ristrutturazione dello schema E-R si è proceduto innanzitutto all'eliminazione della generalizzazione Utente: Utente Sportivo/Proprietario Struttura Sportiva. L'alternativa più adatta alla nostra situazione, in quanto generalizzazione totale, è stata trasferire gli attributi dell'entità padre ai figli. La scelta è dovuta ai vantaggi che questa soluzione porta in termini di tempi di accesso alle informazioni e migliora la gestione della memoria.

In seguito, si è proceduto all'individuazione delle chiavi primarie per ogni entità e qualora un'entità ne era sprovvista abbiamo creato degli id(stringhe). Infine, si è provveduto ad effettuare un'analisi delle ridondanze e degli attributi.

- Sicurezza: Uno dei target principali dell'implementazione di una base di dati è riuscire a mantenere la consistenza dei dati a fronte di qualsiasi problema che sia il crash di sistema o del database, in modo che i dati siano sempre coerenti con la logica dell'applicazione. Cosa succede se il sistema va in crash mentre è in corso l'operazione di inserimento nel database? Potremmo aver caricato il dato all'interno del filesystem ma non il riferimento nel database. Inoltre, se dovessimo caricare i documenti nel filesystem i file stessi potrebbero essere esposti (accessibili al di fuori del sito), mentre caricandoli nel database i dati non possono essere acceduti se non si hanno i permessi necessari.
- Backup: Effettuando backup centralizzati in un solo punto si diminuiscono i punti di failure, inoltre, effettuando backup separati di filesystem e database possiamo riscontrare incongruenze quando si va ad effettuare il ripristino dei dati perché quest'ultimi non sono effettuati in contemporanea.

### 3.5. Controllo degli accessi e sicurezza

Essendo il sistema formato da tre tipologie di attori, ognuno con funzionalità differenti, si è organizzato all'interno della tabella di accesso globale, in seguito riportata, le operazioni che ogni utente può usufruire

Attori	Utente non Loggato	Utente Sportivo	Proprietario Struttura Sportiva

Oggetti			
Utente	Registrazione() Login()	Logout() VisualizzaProfilo() ModificaProfilo() DisattivaProfilo() RecuperoPassword()	Logout() VisualizzaProfilo() ModificaProfilo() DisattivaProfilo() RecuperoPassword()
Campi			CreazioneCampoSportivo() ModificaCampoSportivo() EliminaCampoSportivo() MostraCampiSportivi()
Evento	RicercaEventi() RicercaStrutturaSportiva Ospitante()	CreazioneEventoSportivo() ModificaEventoSportivo() EliminaEventoSportivo() MostraEventoSportivo() RicercaEventi() RicercaEventiVicinanze() RicercaStrutturaSportivaOspitante()	MostraEventoSportivo() ModificaStatoEventoSportivo() RicercaEventi() RicercaEventiVicinanze() RicercaStrutturaSportivaOspitante()
Fascia Oraria			CreazioneFasciaOrariaCampo() ModificaFasciaOrariaCampo() EliminaFasciaOrariaCampo() MostraFasciaOrariaCampo()

### 3.6. Controllo del sistema globale

Il sistema è caratterizzato da un sito web accessibile dal browser. Quando un utente effettua l'accesso (Login), avviene un accesso al database per controllare se effettivamente l'utente esiste. Una volta avuta la conferma, l'utente può accedere alle diverse funzionalità messe a disposizione da parte del sistema.

L'utente può richiedere una risorsa attivando una URL. Il browser costruisce una HTTP Request con le informazioni prese dall'URL e viene inviato al server HTTP, in particolare viene catturata da un Controller.

Il Controller riceve l'HTTP Request tradotta in oggetto Java, dopo averlo letto utilizza i dati estratti per eseguire delle operazioni, opzionalmente utilizzando i Model di riferimento dell'oggetto desiderato. Termina con la realizzazione di una risposta inviata sotto forma di Response ad una View, che avrà il compito di leggere i dati e formattarli in pagina Web.

## 4. SERVIZI DEI SOTTOSISTEMI

- **Utente**
  - Registrazione - consente all'utente di potersi registrare in iPlay

- Login - consente all'utente di effettuare il login
  - Logout - consente all'utente di effettuare il logout
  - Recupero Password - consente all'utente di poter recuperare la password
  - Modifica Profilo - consente all'utente di poter modificare i dati del proprio profilo
  - Visualizza Profilo – consente all'utente di poter visualizzare i dati del proprio profilo
  - Disattiva Profilo – consente all'utente di poter disattivare il proprio profilo
- **Campi**
    - Creazione Campo Sportivo – consente al proprietario della struttura sportiva di creare un nuovo campo sportivo
    - Modifica Campo Sportivo – consente al proprietario della struttura sportiva di poter modificare i dati di un campo sportivo
    - Mostra Campi Sportivi – consente al proprietario della struttura sportiva di poter visualizzare i dati del campo sportivo
    - Elimina Campo Sportivo – consente al proprietario della struttura sportiva di poter eliminare un campo sportivo dal sistema
- **Evento**
    - Creazione Evento Sportivo – consente all'utente sportivo di poter creare un nuovo evento sportivo
    - Modifica Evento Sportivo – consente all'utente sportivo di poter modificare i dati di un evento sportivo
    - Elimina Evento Sportivo – consente all'utente sportivo di poter eliminare un evento sportivo
    - Mostra Evento Sportivo – consente all'utente sportivo/proprietario struttura sportiva di poter visualizzare i dati di un evento sportivo
    - Ricerca Evento Sportivo – consente all'utente sportivo/proprietario struttura sportiva di poter ricercare un evento sportivo presente nel sistema
    - Ricerca Evento Sportivo Nelle Vicinanze – consente all'utente sportivo/proprietario struttura sportiva di poter ricercare un evento sportivo nelle vicinanze presente nel sistema
    - Ricerca Struttura Sportiva Ospitante – consente all'utente sportivo/proprietario struttura sportiva di poter ricercare la struttura sportiva che ospita l'evento
    - Modifica Stato Evento Sportivo – consente al proprietario struttura sportiva di poter modificare lo stato di un evento presente nel sistema
- **Fascia Oraria**
    - Crea Fascia Oraria Campo Sportivo – consente al proprietario struttura sportiva di poter creare una nuova fascia oraria per un determinato campo sportivo
    - Modifica Fascia Oraria Campo Sportivo – consente al proprietario struttura sportiva di poter modificare i dati di una determinata fascia oraria riferita a un campo sportivo
    - Elimina Fascia Oraria Campo Sportivo – consente al proprietario struttura sportiva di poter eliminare una determinata fascia oraria riferita a un campo sportivo
    - Mostra Fascia Oraria Campo Sportivo – consente al proprietario struttura sportiva di poter visualizzare la fascia oraria riferita a un campo sportivo

- **Storage**
  - Salvare Dati
  - Aggiornare Dati
  - Eliminare Dati