



# UNIVERSITÀ DEGLI STUDI DI SALERNO

Ingegneria del Software



SDD – System Design Document

ANNO ACCADEMICO 2017/2018

*Versione 1.1*

# Top Manager

Nome
Prof. De Lucia Andrea
Prof. Francese Rita

# Partecipanti

Nome	Matricola
Casillo Francesco	0512103760
Ferrentino Alfredo	0512103714

# Revision History

Data	Versione	Descrizione	Autore
20/06/2018	1.0	Stesura del documento	Membri del Team
21/06/2018	1.1	Revisione del documento	Membri del team

## Sommario

<b>1. Introduzione</b>	4
1.1 Scopo del sistema	4
1.2 Obiettivi di design	4
1.3 Definizioni, Acronimi e Abbreviazioni	5
1.4 Riferimenti	5
1.5 Visione Generale	5
<b>2. Sistema Software corrente</b>	6
<b>3. Sistema Software proposto</b>	6
3.1. Visione generale	6
3.2. Decomposizione in sottosistemi	6
3.3 Mapping Hardware/Software	9
3.4. Gestione dati persistenti	11
3.5. Controllo degli accessi e sicurezza	12
3.6. Controllo globale del software	12
3.7. Condizioni di Boundary	13
3.7.1. Avvio del sistema	13
3.7.2. Terminazione del sistema	13
3.7.3. Fallimento del sistema	13
<b>4. Servizi dei Sottosistemi</b>	17
<b>5. Glossario</b>	20

# 1. Introduzione

## 1.1 Scopo del sistema

Il progetto FantaBet vuole mettere a disposizione degli utenti una piattaforma fantacalcistica nella quale possono competere con altri utenti a scopo ludico. Ogni utente potrà mettersi alla prova, scommettendo sulla formazione migliore di giornata e non solo. Infatti, agli utenti sarà permesso di richiedere l'apertura di competizione private per organizzarsi tra amici, parenti, etc.

Tuttavia, la poca competenza che alcuni utenti possono avere nell'utilizzo di portali Web implica necessariamente l'utilizzo di interfacce intuitive e di facile comprensione. Pertanto, è prevista la realizzazione di un'interfaccia intuitiva e semplice, indipendente dalle conoscenze dell'utente di uso di portali web.

## 1.2 Obiettivi di design

Tempi di risposta	Essendo Fantabet interamente online, il problema dei tempi di risposta purtroppo dipende molto dalla qualità della connessione di cui si dispone, in generale, se si dispone di una connessione di almeno 100Mbs i tempi di risposta risultano essere decisamente brevi nell'ordine di centesimi di secondi.
Memoria	Il sistema utilizza un database relazionale per memorizzare i dati. all'interno di esso sono conservati in modo sicuro tutti i dati riguardanti i vari account registrati, e i dati sulle formazioni e sulle competizioni dei vari utenti.
Robustezza	Il sito deve godere di una buona robustezza, tutti i componenti devono resistere a guasti e conservare le informazioni anche in questi casi. Inoltre, il sito reagisce in modo dinamico ad input sbagliati mostrando immediatamente messaggi di errore con la possibilità di reinserimento dell'input sbagliato.
Disponibilità	Implementando un server sempre attivo il software Fantabet può essere reperibile 24 ore su 24.
Portabilità	Essendo un software implementato interamente online, è senz'altro portabile, con la possibilità di accesso da ogni dispositivo che possa godere di una connessione. Infatti, il nostro sito è disponibile per tutti i browser (Chrome, Opera, Firefox, Safari) e nel progettarlo abbiamo fatto in modo che sia responsive per facilitarne la visualizzazione su tutti i dispositivi.
Sicurezza	Come descritto sopra, dietro l'intero sistema vi è un database che contiene tutti i dati riguardanti l'utenza; la priorità è quindi quella di difendere tale database dai guasti, ma soprattutto dagli attacchi informatici. Altre tecniche utilizzare per garantire sicurezza sono comunque prevalentemente basate sul "login", la quale permette il riconoscimento dell'utente. Tutti gli accessi al sistema avvengono tramite interrogazione ai "filtri" che appunto stabiliscono se un determinato utente ha i permessi per operare in un certo modo, al logout la sessione viene chiusa del tutto eliminando quindi la possibilità tramite glitch o bug di sistema di accedere ad aree del sito senza aver effettuato prima nuovamente il login.
Interfaccia grafica	Il sistema sarà semplificato ulteriormente dall'utilizzo di un'interfaccia user-friendly di semplice comprensione dato che il sito è rivolto a tutti i tipi di persone, sarà quindi reso semplice e minimale.

Modificabilità	Deve essere possibile intervenire sul codice esistente per correggere eventuali bug o implementare nuove funzionalità. Bisogna garantire che il codice sia leggibile per rendere agevole la modifica.
Leggibilità	Il codice sarà ben strutturato per semplificare eventuali interventi su di esso.

### 1.3 Definizioni, Acronimi e Abbreviazioni

- **JDBC:** Java DataBase Connection.
- **RAD:** Requirements Analysis Document.
- **SDD:** System Design Document. **MVC:** Model-View-Controller.
- **User-friendly:** aggettivo utilizzato per definire un software di facile utilizzo anche per persone non esperte nell'utilizzo del computer
- **Mysql:** è il più diffuso database Open Source basato sul linguaggio SQL.
- **DBMS:** Database Management System, Sistema di gestione del database.

### 1.4 Riferimenti

Il materiale di riferimento utilizzato per la stesura di questo stesso documento comprende:

- Libro di testo: B.Bruegge, A.H. Dutoit, Object Oriented Software Engineering – Using UML, Patterns and Java, Prentice Hall, 3rd edition, 2009
- Slide del docente, reperibili sulla piattaforma
- RAD

### 1.5 Visione Generale

Il documento si compone di cinque parti. In particolare, nella prima parte sono stati introdotti gli obiettivi di design e sono stati forniti i riferimenti ad altri materiali. In seguito, nella sezione sistema software corrente, verrà descritto un sistema software simile. La sezione sistema software proposto documenta il system design del nuovo sistema. In questa sezione vengono descritti i seguenti elementi:

- **decomposizione in sottosistemi:** il sistema viene suddiviso in diversi sottosistemi. Ricordiamo che un sottosistema è formato da un insieme di classi, associazioni, operazioni e vincoli che sono in relazione tra di loro. Ogni sottosistema è caratterizzato dai servizi che offre agli altri sottosistemi.
- **Mapping hardware/software:** in questa sezione vengono prese decisioni riguardo le piattaforme hardware su cui il sistema dovrà girare, una volta decise le piattaforme è necessario mappare le componenti su di esse.
- **Gestione dei dati persistenti:** descrive i dati persistenti che vengono memorizzarli dal sistema ed il tipo di infrastruttura usata per memorizzarli.
- **Controllo degli Accessi e Sicurezza:** descrive il modello degli utenti del sistema in termini di una matrice degli accessi. Questa descrive le informazioni a cui i singoli attori possono accedere e quali sono le operazioni che questi hanno la possibilità di effettuare.
- **Controllo Globale del Software:** descrive il modo in cui è implementato il controllo globale del software e come si sincronizzano i sottosistemi.
- **Condizioni di Boundary:** in questo paragrafo vengono descritte le condizioni limite del sistema come start-up e shutdown e la gestione dei fallimenti del sistema.

La sezione servizi dei sottosistemi descrive in termini di operazioni quali sono i servizi forniti da ciascun sottosistema. L'ultima parte del documento è costituita dal glossario che si occupa di elencare una serie di termini e fornire la relativa spiegazione in maniera tale da fornire supporto a coloro che leggono il documento.

## 2. Sistema Software corrente

Attualmente esiste un sistema software di gestione fantacalcistica: Fantagazzetta. Tuttavia, si occupa principalmente della gestione di fantaleghe (alias competizioni in Fantabet) senza una competizione globale. Inoltre, i regolamenti e la modalità di gioco sono del tutto differenti dal sistema proposto.

## 3. Sistema Software proposto

### 3.1. Visione generale

Il sistema proposto è un software web-based per la gestione di un fantacalcio. Le tipologie di utenti si dividono in utente e amministratore. L'utente potrà accedere alle funzionalità che permettono la gestione del proprio profilo, la partecipazione a varie competizioni e la possibilità di effettuare operazioni su di esse. L'amministratore potrà accedere alle funzionalità che riguardano la gestione degli utenti, delle competizioni, delle richieste di competizione nonché del proprio profilo.

L'architettura client server garantirà le seguenti caratteristiche:

- *Portabilità*: in quanto il sistema essendo web-based potrà essere utilizzato su una grande varietà di dispositivi diversi.
- *Performance*: il sistema garantirà reattività nel completamento delle task, anche se in realtà questo requisito è molto influenzato dalla qualità della connessione ad Internet di cui dispone l'utente.
- *Affidabilità*: il server dev'essere in grado di garantire affidabilità e consistenza dei propri dati anche in seguito al verificarsi di guasti o condizioni d'errore.

Il nostro sistema utilizza un'architettura three tier in cui ogni stato dell'applicazione ha specifici compiti:

**Presentation layer**: si occupa di presentare i risultati di una computazione agli utenti e di raccogliergli gli input.

**Application layer**: si occupa di fornire le funzionalità specifiche dell'applicazione.

**Storage layer**: si occupa della gestione dei database di sistema.

### 3.2. Decomposizione in sottosistemi

Per realizzare il sistema è stata usata un'architettura three-tier. Questo è un caso particolare di un'architettura multi tier in cui la logica dell'applicazione viene suddivisa in tre layer:

- *Presentation layer*: composto dalle interfacce grafiche e in particolare dai boundary object come le form che vengono compilate dall'utente.
- *Application layer*: composto dagli oggetti che si occupano della gestione del controllo, dell'elaborazione dati e di notificare i cambiamenti al presentation layer. Questo strato interagisce con il database sottostante tramite lo storage layer.
- *Storage layer*: si occupa della memorizzazione dei dati persistenti e del loro recupero dal database ad esempio eseguendo delle query.

Il sottosistema Presentation Layer è stato suddiviso in due sottosistemi:

- Sottosistema **Utente**. Include tutte le interfacce grafiche a cui l'utente può accedere ovvero: visualizza informazioni sul proprio profilo, visualizza informazioni su competizioni, richiede l'apertura di una nuova competizione ed inserisce la formazione di giornata per ciascuna competizione.

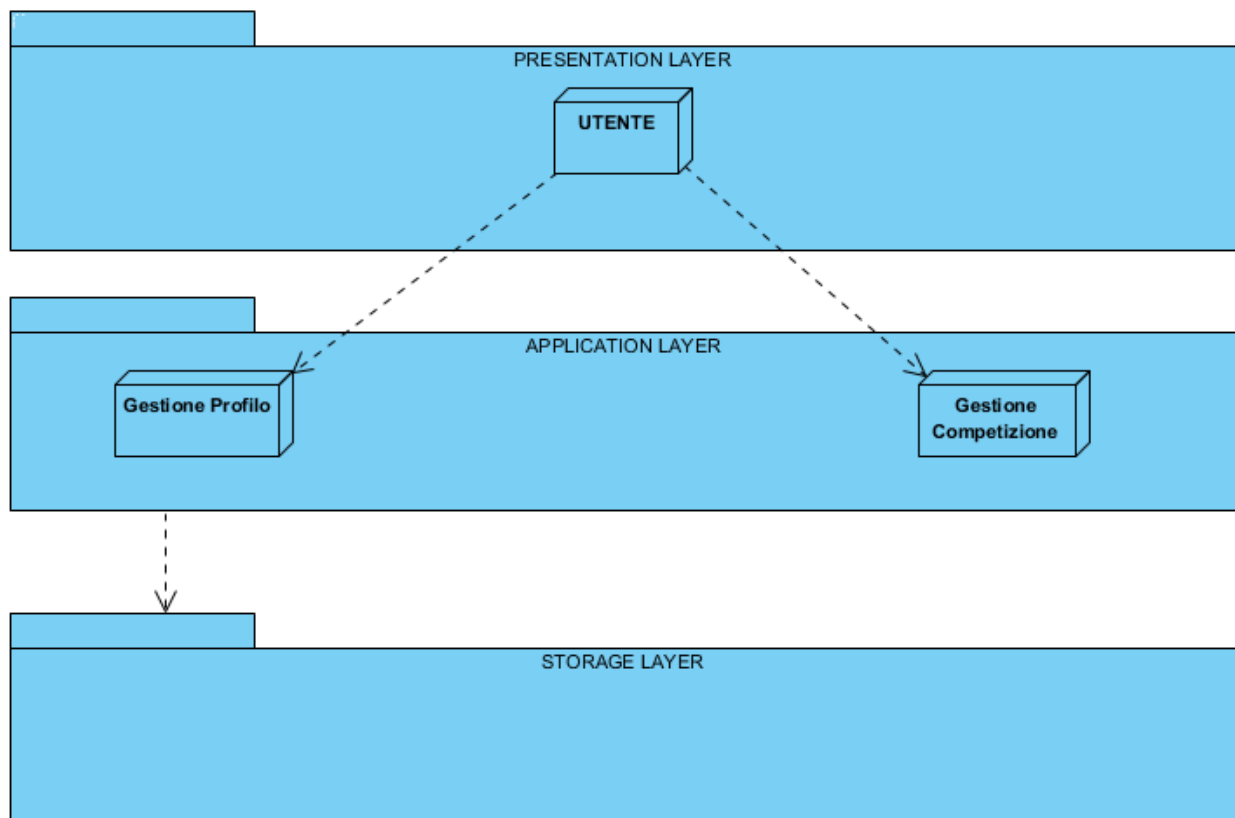
- Sottosistema **Amministratore**. Include tutte le interfacce grafiche a cui l'amministratore può accedere ovvero: elenco utenti, competizione, richieste, visualizzazione dettagli relativi ad un utente e visualizzazione informazioni personali.

Il sottosistema Application Layer è stato diviso in sottosistemi:

- Sottosistema **Gestione Profilo** che include tutte le operazioni che permettono di visualizzare le informazioni e modificare il proprio profilo.
- Sottosistema **Gestione Utenti** che include tutte le funzionalità che permettono di gestire l'utenza.
- Sottosistema **Gestione Competizione** che include tutte le funzionalità che permettono di gestire e partecipare alle competizioni.

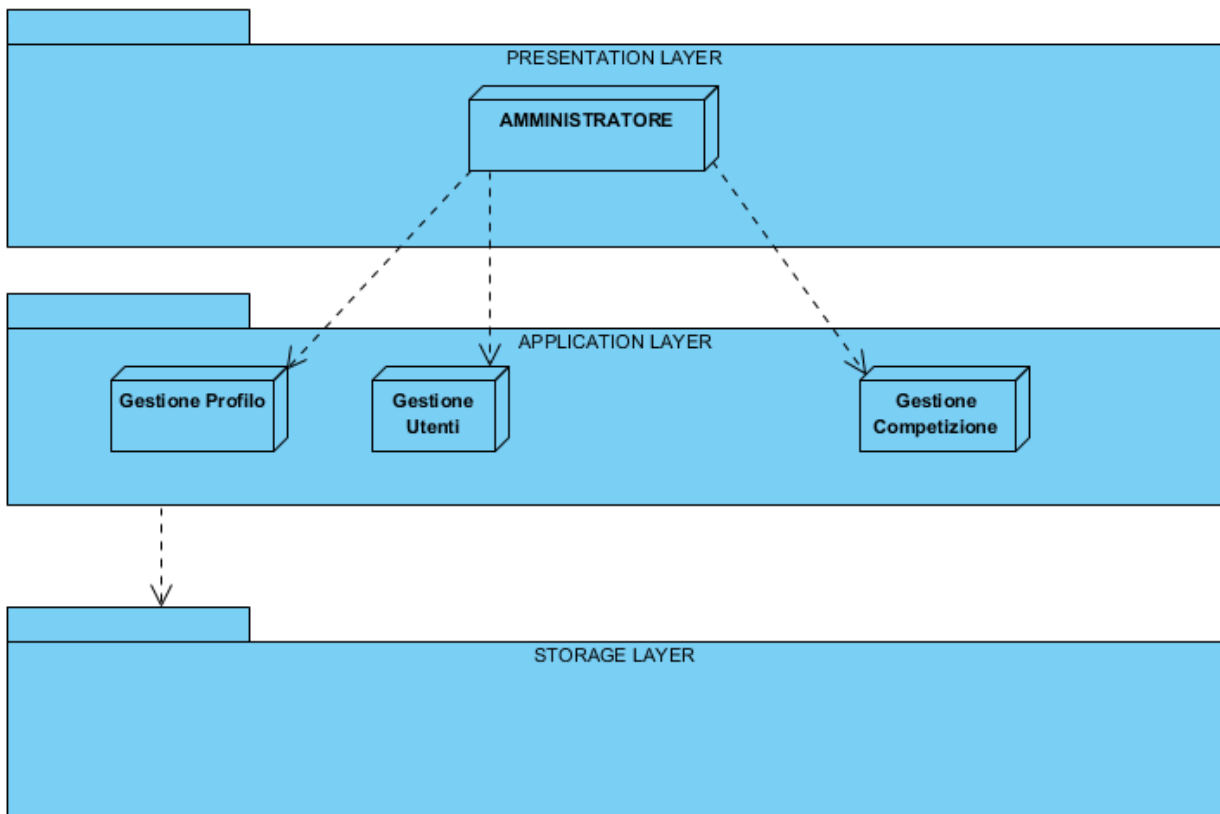
Ora mostriamo nello specifico ogni layer:

### UTENTE



- **Gestione Profilo:** consente all'utente di visualizzare e/o modificare i dati del proprio profilo.
- **Gestione Competizione:** consente all'utente di visualizzare informazioni riguardanti le competizioni a cui partecipa, consente l'inserimento di una formazione per ciascuna di esse e permette la formulazione di una richiesta di apertura di una competizione privata.

## AMMINISTRATORE



- **Gestione Profilo:** consente all'amministratore di visualizzare e/o modificare i dati del proprio profilo.
- **Gestione Competizione:** consente all'amministratore di visualizzare informazioni riguardanti le competizioni, modificarle e permette l'approvazione delle richieste di apertura di competizioni private.
- **Gestione Utenti:** consente all'amministratore di modificare e gestire le informazioni degli utenti.

In entrambi i sottosistemi di presentazione e applicazione mancano il sottosistema di **Gestione Registrazione** e il sottosistema di **Gestione Autenticazione** che sono stati omessi.

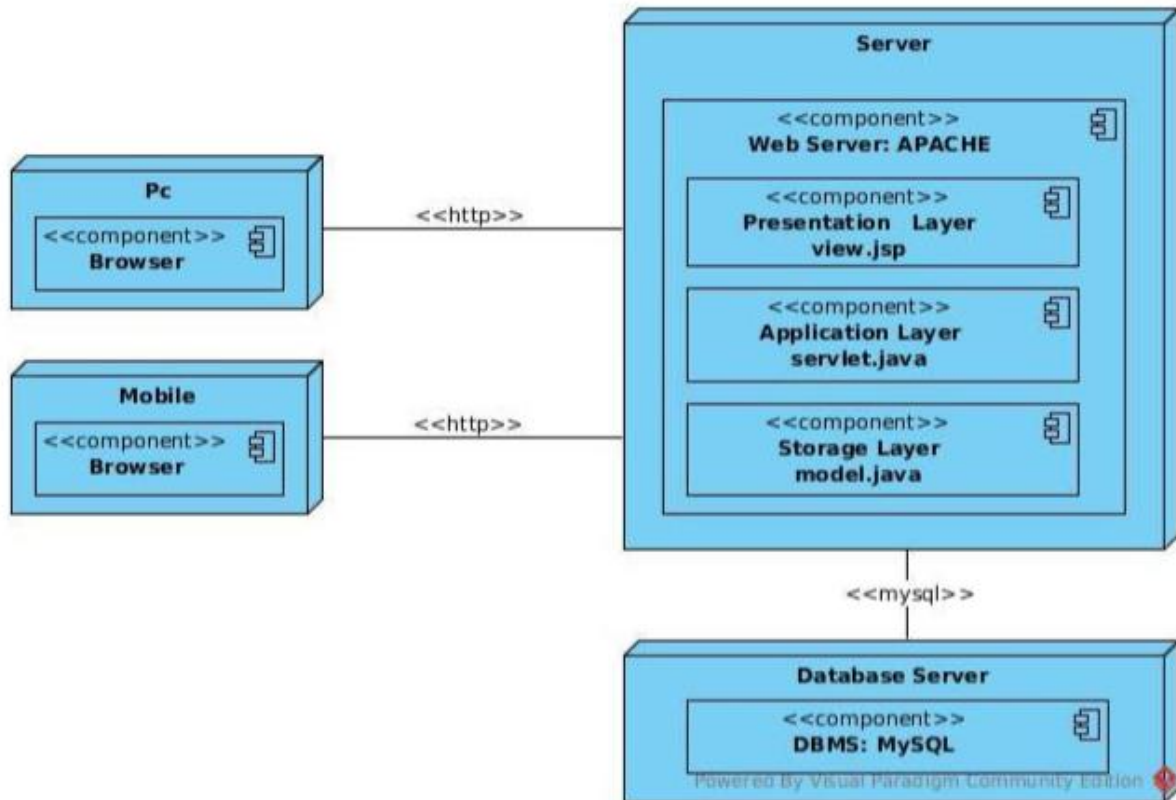
Il primo consente la registrazione al sistema ad un nuovo utente.

Il secondo consente il Login e il Logout ad un utente registrato al sistema, utente o amministratore che sia.

Entrambi i sottosistemi saranno comunque analizzati di seguito.



### 3.3 Mapping Hardware/Software



#### **Presentation layer**

L'utente utilizza il sistema mediante un qualsiasi Browser installato sul suo dispositivo sia esso uno smartphone o un PC.

#### **Application layer**

Il sistema, e quindi le funzionalità, sono implementate in linguaggio Java. Le varie servlet si occuperanno del flusso di controllo necessario all'implementazione delle varie funzioni.

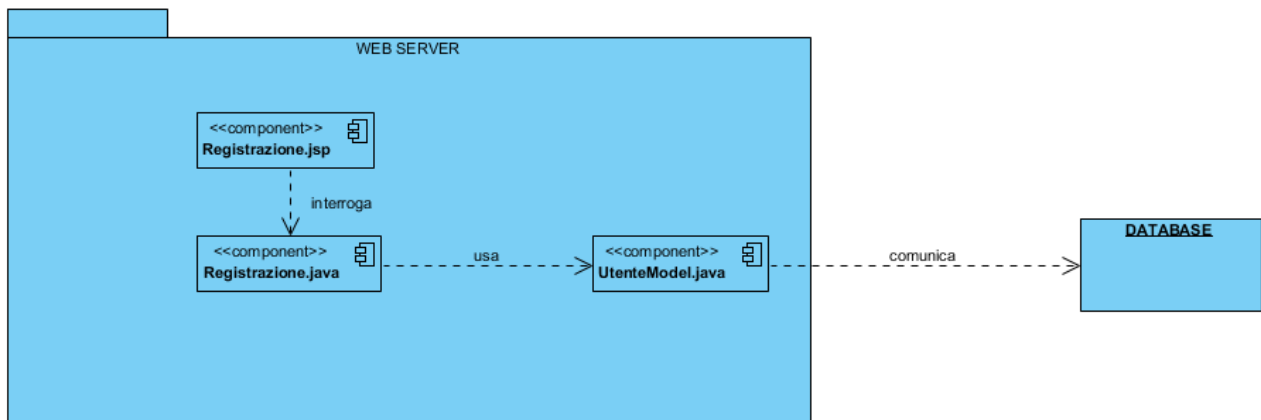
#### **Storage layer**

Questo layer rappresenta il collegamento con il database sottostante e si occupa, quindi, delle richieste di accesso e modifiche sui dati persistenti.

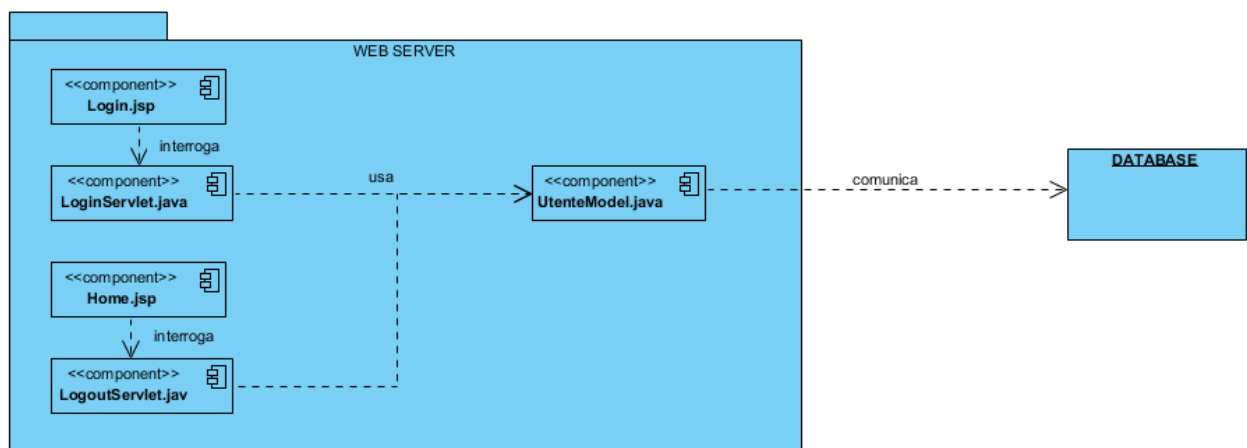
#### **Database Server**

Il DBMS usato è MySQL il quale fornisce un'API per permettere l'interazione tra sistema e database.

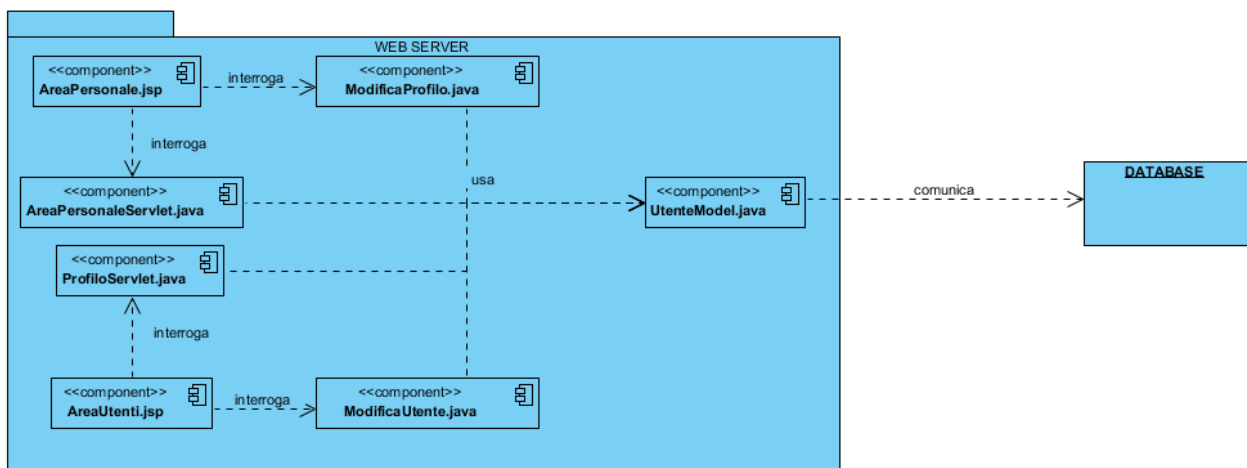
## GESTIONE REGISTRAZIONE



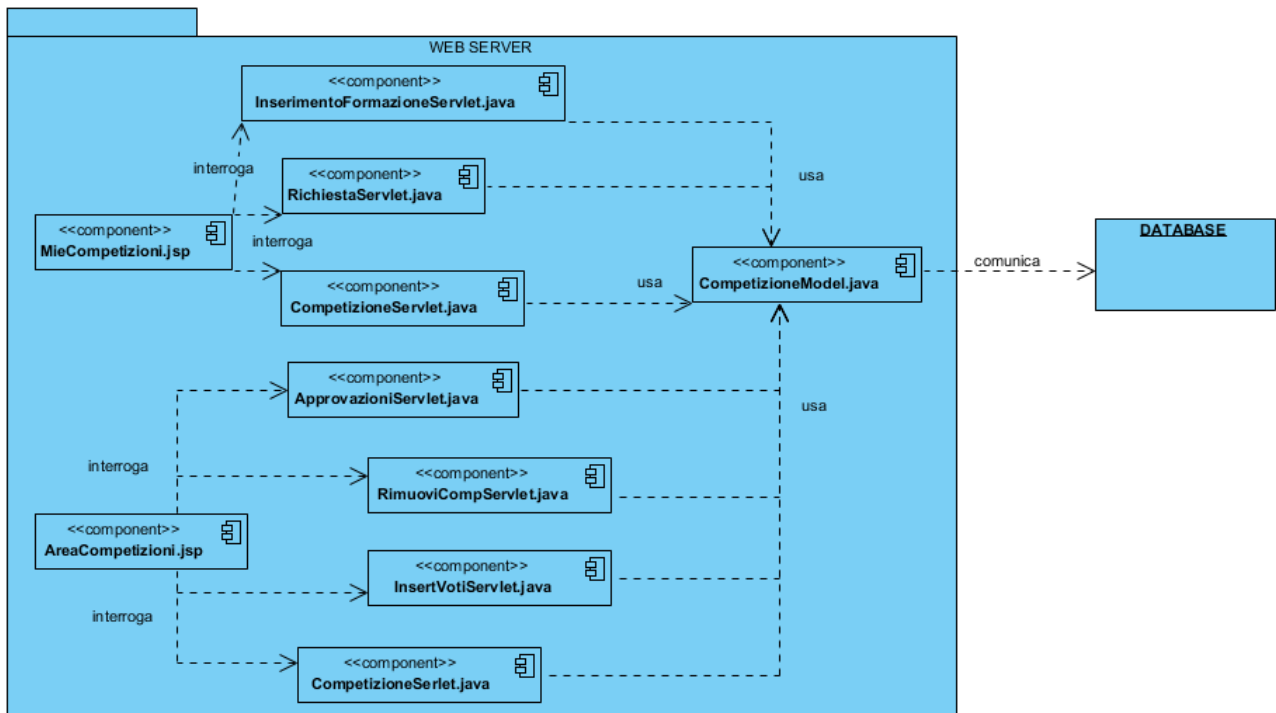
## GESTIONE AUTENTICAZIONE



## GESTIONE UTENTE



## GESTIONE COMPETIZIONE



### 3.4. Gestione dati persistenti

Analizzando la struttura del nostro sistema siamo giunti alla conclusione che per la memorizzazione delle informazioni che necessitano di essere rese persistenti, è meglio utilizzare una struttura con database. Abbiamo quindi utilizzato un database relazionale in modo da avere un'agevole gestione dell'accesso concorrente ai dati. Inoltre, questo approccio garantisce la consistenza dei dati stessi. La scelta di un DBMS è stato il giusto compromesso per essere il più possibile coerenti con i design goals stabiliti, infatti, questa soluzione porta diversi vantaggi quali:

- **Privatezza dei dati:** Un DBMS permette che utenti diversi possano accedere a diverse porzioni della base di dati e possano essere abilitati a operazioni diverse.
- **Imposizioni di vincoli di integrità sui dati:** Un DBMS permette di specificare diversi tipi di vincoli per mantenere l'integrità dei dati e controlla che tali vincoli siano soddisfatti quando vengono effettuate delle modifiche.
- **Atomicità delle operazioni:** Un DBMS permette di effettuare sequenze di operazioni in modo atomico. L'atomicità delle transazioni permette di mantenere consistente lo stato del database.

Per ulteriori dettagli si rimanda al documento di gestione dei dati persistenti.

### 3.5. Controllo degli accessi e sicurezza

All' interno del sistema i vari attori hanno il permesso di eseguire operazioni diverse sui vari sottosistemi. Per schematizzare meglio il controllo si è deciso di usare la matrice degli accessi riportata di seguito:

ATTORE SOTTOSISTEMA	AMMINISTRATORE	UTENTE	UTENTE NON REGISTRATO
GESTIONE REGISTRAZIONE			√ Registrazione
GESTIONE AUTENTICAZIONE	√ Login √ Logout	√ Login √ Logout	
GESTIONE UTENTE	√ Visualizza Utente √ Modifica Utente √ Ricerca Utente √ Modifica Profilo √ Visualizza Profilo	√ Visualizza Profilo √ Modifica Profilo	
GESTIONE COMPETIZIONE	√ Approva richieste √ Rimuovi Competizione √ Ricerca Competizione √ Visual. Competizione √ Inserimento Voti	√ Richiesta Comp. √ Visualizza Comp. √ Inserisci Formazione √ Visual. Formazione √ Aggiunta Partecipante	

### 3.6. Controllo globale del software

Il sito web ha un flusso guidato dagli eventi (event-driven): non esiste una sequenza di operazioni prestabilita, è l'utente a scegliere quale operazione desidera eseguire. L'architettura è composta da un web server che rimane in ascolto in attesa di una richiesta da parte di un web browser. Appena riceve una richiesta elabora il risultato invocando opportuni metodi sui sottosistemi coinvolti nell'operazione e invia l'output al client.

### 3.7. Condizioni di Boundary

Le condizioni limite riguardano l'accensione e lo spegnimento del sistema per quanto riguarda il lato Server. Dal lato Client si riferiscono agli errori di connessione al server.

#### 3.7.1. Avvio del sistema

Per lo start-up del sistema è necessario avviare il web server. In maniera tale che sia possibile per gli utenti e amministratori accedere al sistema tramite web.

#### 3.7.2. Terminazione del sistema

Per terminare il server è necessario l'intervento dell'amministratore. Questo dovrà effettuare la procedura di terminazione dopo la quale non sarà possibile per nessun client connettersi al sistema

#### 3.7.3. Fallimento del sistema

Nel caso si verifichi un errore dovuto all'hardware o al software si cercherà di ripristinare una configurazione del sistema precedente allo stato d'errore. Visto che i dati sono gestiti dal DBMS non c'è rischio di perderli. Tuttavia, se si verifica un guasto al supporto di memorizzazione si può avere una perdita di dati.

## Scenari

<b>Nome Scenario</b>	Startup Sistema
<b>Istanze di Attori Partecipanti</b>	Aniello: Amministratore
<b>Flusso di Eventi</b>	<ol style="list-style-type: none"><li>1. Aniello decide di voler avviare il sistema e quindi clicca sul pulsante "Avvia".</li><li>2. Il sistema, con le opportune procedure di avvio, attiva i server e i relativi servizi in remoto rendendosi disponibile ad eventuali richieste.</li><li>3. Il sistema notifica il successo della procedura.</li></ol>

<b>Nome Scenario</b>	Shutdown Sistema
<b>Istanze di Attori Partecipanti</b>	Aniello: Amministratore
<b>Flusso di Eventi</b>	<ol style="list-style-type: none"><li>1. Aniello decide di voler arrestare il sistema e quindi accede alla pagina dedicata e clicca sul pulsante "Arresta".</li><li>2. Il sistema effettua una scansione per verificare se ci sono ancora richieste in sospeso.</li><li>3. Il sistema porta a termine le eventuali richieste in sospeso</li><li>4. Tramite le opportune procedure di arresto il sistema disattiva i servizi in remoto e il server.</li><li>5. Il sistema notifica il successo della procedura.</li></ol>

Casi d'uso

ID	UC_Startup
Nome Caso Uso	Startup Server
Istanze di Attori Partecipanti	Amministratore
Condizione di Entrata	L'amministratore visualizza la console per effettuare l'avvio del sistema.
Flusso di Eventi	<div><div>Utente</div><div>Sistema</div><p>L'amministratore clicca sul pulsante "Avvia"</p><p>Il sistema attiva i propri servizi rendendosi disponibile in tutte le sue funzionalità.</p></div>
Condizione di Uscita	Il server è attivo e i relativi servizi sono disponibili
Eccezioni	Errore Startup.

<b>ID</b>	UC_Shutdown
<b>Nome Caso Uso</b>	Shutdown Server
<b>Istanze di Attori Partecipanti</b>	Amministratore
<b>Condizione di Entrata</b>	L'amministratore visualizza la console per effettuare la terminazione del sistema
<b>Flusso di Eventi</b>	<div> <div> <p>Utente</p> <p>L'amministratore clicca sul pulsante "Spegni".</p> </div> <div> <p>Sistema</p> <p>Il sistema effettua la procedura per la terminazione del server.</p> </div> </div>
<b>Condizione di Uscita</b>	Il server si è spento correttamente
<b>Eccezioni</b>	Errore shutdown.



## 4. Servizi dei Sottosistemi

### GESTIONE REGISTRAZIONE

<b>Sottosistema</b>	Gestione Registrazione
<b>Descrizione</b>	Sottosistema che permette l'iscrizione al sistema ad un nuovo utente.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Registrazione	Questa funzionalità permette all'utente non registrato di inserire i dati che andranno ad identificare il suo profilo all'interno del sistema.
Dati non corretti o esistenti	Questa funzionalità permette al sistema di restituire un messaggio d'errore nel caso in cui un utente immetta dati non corretti o già esistenti.

### GESTIONE AUTENTICAZIONE

<b>Sottosistema</b>	Gestione Autenticazione
<b>Descrizione</b>	Sottosistema che gestisce tutte le operazioni necessarie per l'autenticazione degli utenti registrati e le operazioni necessarie per controllare la correttezza dei dati inseriti.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Login	Questa funzionalità permette di effettuare l'accesso al sistema inserendo le proprie credenziali.
Logout	Questa funzionalità permette di uscire dal sistema.
Dati non esistenti	Questa funzionalità permette al sistema di restituire un messaggio d'errore nel caso in cui un utente immetta dati non esistenti.

## GESTIONE UTENTE

Sottosistema	Gestione Utente
Descrizione	Sottosistema che gestisce e raccoglie tutte le funzionalità di cui possono usufruire gli utenti del sistema ovvero: visualizzazione e modifica degli utenti del sistema e la visualizzazione dei propri profili da parte degli utenti.
Servizi Offerti	
Servizio	Descrizione
Visualizza Profilo	Questa funzionalità permette all'utente di visualizzare le informazioni inerenti al proprio profilo.
Modifica Profilo	Questa funzionalità permette all'utente di modificare i dati personali.
Visualizza Utente	Questa funzionalità permette all'amministratore di visualizzare i dati di un utente.
Ricerca Utente	Questa funzionalità permette all'amministratore di cercare un utente nel sistema.
Modifica Utente	Questa funzionalità permette di modificare un utente del sistema.
Utente inesistente	Questa funzionalità permette al sistema di restituire un messaggio d'errore nel caso in cui l'amministratore ricerchi un utente non esistente nel sistema.
Dati non corretti	Questa funzionalità permette al sistema di restituire un messaggio d'errore nel caso in cui un utente qualsiasi immetta dati non corretti nella modifica del profilo.

## GESTIONE COMPETIZIONE

Sottosistema	Gestione Competizione
Descrizione	Sottosistema che gestisce e raccoglie tutte le informazioni e le operazioni inerenti alle competizioni.
Servizi Offerti	
Servizio	Descrizione
Approva Competizione	Questa funzionalità permette all' amministratore di approvare le richieste di competizione da parte degli utenti.
Rimozione Competizione	Questa funzionalità permette all' amministratore di rimuovere una competizione esistente.
Ricerca Competizione	Questa funzionalità permette all' amministratore di cercare una competizione esistente.
Visualizza Competizione	Questa funzionalità permette all'amministratore di visualizzare una competizione esistente.
Inserimento Voto	Questa funzionalità permette all' amministratore di inserire i voti relativi ad una giornata della competizione.
Richiesta Competizione	Questa funzionalità consente all'utente di richiedere l'apertura di una competizione privata all'amministratore.
Visualizza Competizione	Questa funzionalità consente all'utente di visualizzare i dati relativi ad una competizione a cui partecipa.
Inserimento formazione	Questa funzionalità consente all'utente di inserire una formazione per una competizione per la giornata corrente.
Visualizza formazione	Questa funzionalità consente all'utente di visualizzare le formazioni inserite nelle scorse giornate.
Aggiungi partecipante	Questa funzionalità consente all'utente di aggiungere un partecipante alla competizione selezionata.
Costo Eccessivo	Questa funzionalità permette al sistema di restituire un messaggio d'errore nel caso in cui un utente immetta una formazione il cui costo è maggiore dei crediti a disposizione.
Dati errati o esistenti	Questa funzionalità permette al sistema di restituire un messaggio d'errore nel caso in cui un utente, nel richiedere l'apertura di una competizione, immetta dati non corretti o già esistenti.
Competizione Inesistente	Questa funzionalità permette al sistema di restituire un messaggio d'errore nel caso in cui l'amministratore immetta dati non esistenti nel sistema.

## 5. Glossario

- **Client:** componente che accede ai servizi o alle risorse di un'altra componente, detta server.
- **Deployment Diagram:** Schema che descrive la struttura dinamica del sistema
- **DBMS:** programma informatico (o, più frequentemente, un insieme di programmi) progettato per gestire un database, ovvero un insieme di numerosi dati strutturati. Le operazioni, normalmente, sono richieste da un gran numero di utenti.
- **Form:** finestra di dialogo incorporata in una pagina Web che consente all'utente di inserire informazioni destinate ad un server. Generalmente richiede un programma sul server che si occupi di esaminare le informazioni inviate. E' composto da spazi (campi) predefiniti, ad esempio menù a tendina, elenchi puntati o caselle di testo libero.
- **JDBC:** API per il linguaggio di programmazione Java che serve ai client per connettersi a un database. Fornisce metodi per interrogare e modificare i dati. È orientata ai database relazionali.
- **Login:** Procedura attraverso la quale ci si collega con un qualsiasi servizio in linea. All'utente viene assegnato un nome di login ed una password che vengono richiesti dal sistema ogni volta che ci si collega.
- **Layer:** E' un insieme di classi con funzionalità simile (tipicamente raggruppati in un unico package).
- **Logout:** Operazione attraverso la quale si termina un collegamento con un sistema al quale si ha accesso attraverso un nome utente e una password (vedi login).
- **MySQL:** Database management system relazionale, composto da un client con interfaccia a caratteri e un server, disponibile su molte piattaforme.
- **Password:** È un metodo di sicurezza che, mediante una stringa di caratteri, permette di identificare un utente specifico. Generalmente le password sono formate da una sequenza di lettere e numeri; digitando correttamente questi caratteri, si può avere accesso al computer o alla rete.
- **Package:** Un Package rappresenta una collezione di classi ed interfacce che possono essere raggruppate in base alla funzione comune da esse svolta
- **Server:** Programma di gestione di un servizio che invia informazioni in un particolare formato ricevuto e interpretato da un programma Client dal lato ricevente.
- **Shutdown:** fase di terminazione del sistema.
- **Start-up:** processo di accensione e di avvio di un computer, di un dispositivo o di un sistema.