# Design Goals

## Criteri di performance

* **Response Time**: FantaFootball deve garantire un tempo di risposta massimo di 2 secondi alle richieste degli utenti.

## Criteri di affidabilità

* **Fault tolerance**: FantaFootball deve consentire il normale svolgimento delle leghe anche quando l’account di un allenatore partecipante viene cancellato.
* **Robustness**: FantaFootball deve gestire eventuali input non validi da parte degli utenti.

## Criteri di costo

## Criteri di manutenzione

* **Portability**: FantaFootball deve essere accessibile da qualsiasi piattaforma che ha a disposizione un browser.
* **Availability**: tutte le funzionalità di FantaFootball devono essere utilizzabili in qualsiasi momento.
* **Modifiability**: FantaFootball deve essere modificabile per consentire di aggiornare l’elenco dei calciatori a ogni nuova stagione del campionato di Serie A.

## Criteri end user

* **Usabilità**: il sistema sarà molto semplice da apprendere anche senza la consultazione della documentazione associata, che verrà comunque fornita.

# Architettura del sistema proposto

## Panoramica

Il sistema proposto è un’applicazione web che consente a diversi utenti di registrarsi e creare leghe per giocare al “Fantacalcio”. Il sistema fornisce supporto automatizzato per gestione aste e calcolo dei punteggi, inoltre consente ai diversi allenatori di scambiare giocatori.

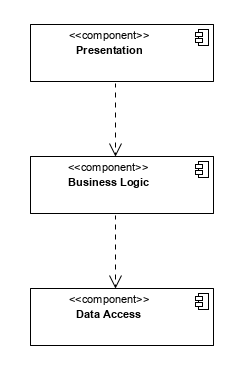
Il sistema verrà suddiviso in client e server, il client gestirà la parte di presentazione e la parte di logica direttamente connessa all’interfaccia grafica, il server invece gestirà la parte di logica relativa ai dati e i dati stessi che saranno salvati in un database salvato sul server. Le funzionalità saranno divise in layer logici in base alle differenti funzionalità: presentazione, business logic e sistema di memorizzazione.

## Decomposizione in sottosistemi

### Decomposizione in Layer

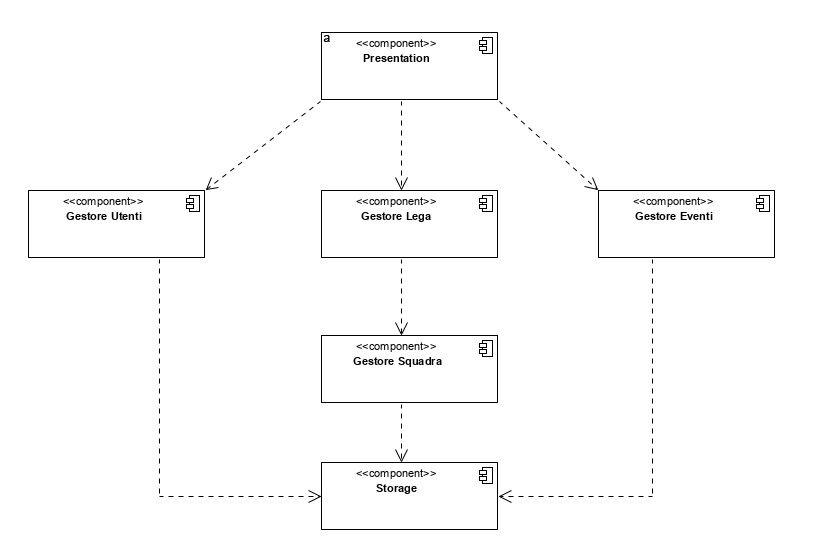
La decomposizione prevista per il sistema è composta da 3 layer che si occupano di funzionalità differenti:

* **Presentation**: gestisce la visualizzazione dei dati e, più in generale, la rappresentazione dei controlli (forms, controlli di input, labels, ecc.) necessari per l'interfaccia utente.
* **Business Logic**: rappresenta la parte principale dell'applicazione, definendo il domain model dell'applicazione, ovvero le entità (ad esempio: allenatore, lega, asta ecc.), le loro relazioni e le logiche applicative.
* **Data Access**: contiene tutto quello che concerne la persistenza dei dati (database, tabelle, record, file system, ecc.).



### Decomposizione in sottosistemi

Il sistema si compone di sette componenti che si occupano di gestirne aspetti e funzionalità differenti:



Il livello di Presentation è composto da un unico sottosistema che gestisce l’interfaccia grafica.

Il livello di Business Logic è composto da 4 sottosistemi:

* Gestore Utenti gestisce la registrazione di nuovi utenti nel sistema, login, eventuali modifiche e cancellazione del profilo;
* Gestore Lega si occupa di funzionalità quali creazione di una nuova lega, invito degli allenatori, organizzazione di aste e gestione delle partite della lega;
* Gestore Squadra gestisce le funzionalità riguardanti acquisto di giocatori, scambi e schieramento della formazione;
* Gestore Bacheca riguarda la pubblicazione, modifica e cancellazione di eventi (o altro tipo di post) pubblicati dallo scout.

Il livello di Data Access prevede il sottosistema Storage che immagazzina e gestisce i dati persistenti.

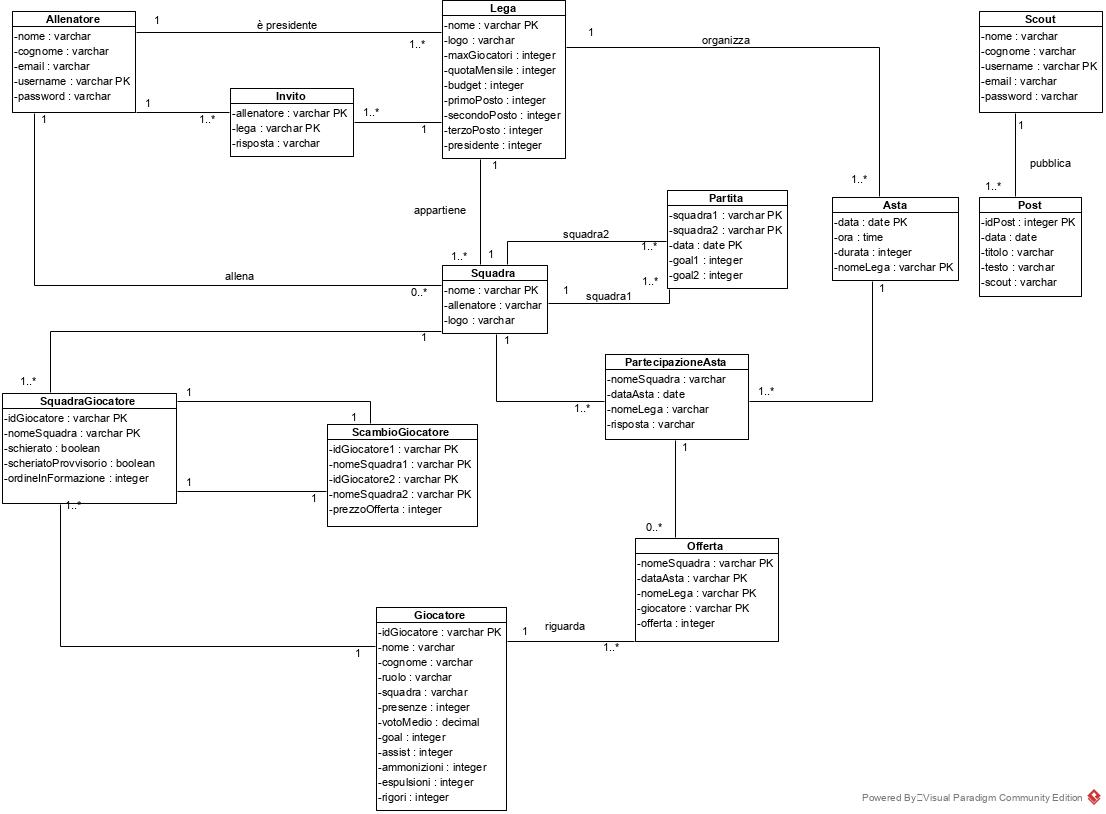
### Diagramma di Deployment

## Mapping hardware/software

Il sistema che sarà realizzato si basa su un’architettura Web-based. La struttura hardware è composta da un server centrale e vari client che possono collegarsi. Sul server ci sarà un DBMS per la gestione dei dati persistenti. Il client conterrà presentation layer e business logic (JavaScript), il server conterrà la logica di business (servlet e JSP) e il layer di data storage.

## Gestione dei dati persistenti

Per gestire i dati persistenti è stato preferito l’utilizzo di un database relazionale poiché serve un ampio spazio di memorizzazione. Inoltre, i database consentono l’accesso concorrente da parte di più utenti.



## Controllo d’accesso

Nel sistema si hanno 3 tipologie di utenti: allenatore, presidente e scout. Ogni utente può accedere al sito tramite un sistema di login composto da username e password. Ogni attore può accedere a diverse funzionalità del sistema ma con diritti di accesso regolamentati sulla base delle differenti tipologie di utenza. Per documentare i diritti di accesso e per tenerne traccia all’interno del sistema usufruiamo di una tabella di controllo degli accessi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sottosistema  Attore | Gestore | | | |
| Utente | Lega | Squadra | Bacheca |
| Presidente | * Login |  |  |  |

## Controllo flusso globale del sistema

Il controllo del flusso globale del sistema è di tipo event-driven in quanto fornisce funzionalità che richiedono una continua interazione con l’utente.

## Condizione di limite (Boundary conditions)

### Start-up

### Start-up (a seguito di un fallimento)

### Terminazione

### Fallimento

# Servizi dei sottosistemi

Gestore Utenti: registrazione

Gestore Lega

Gestore Squadra

Gestore Bacheca