

**Università degli Studi di Salerno**  
**Corso di Ingegneria del Software**

**FantaUnisa**  
**System Design Document**  
**Versione 1.0**



Data: 25/11/2025

Progetto: FantaUnisa	Versione: 1.0
Documento: System Design Document	Data: 25/11/2025

**Partecipanti:**

Nome	Matricola
Clavino Antonio	0512119692
Corona Francesco	0512119827
Sabetta Francesco	0512118990
Tiberini Monica	0512120226

<b>Scritto da:</b>	Tutti i partecipanti.
--------------------	-----------------------

## Revision History

Data	Versione	Descrizione	Autore
21/11/2025	0.1	Prima stesura esaustiva del documento	Sabetta Francesco
23/11/2025	0.2	Aggiunta matrice degli accessi, Tabella 1.3 e Glossario	Sabetta Francesco
24/11/2025	0.3	Aggiunta dei grafici	Tiberini Monica
25/11/2025	1.0	Revisione finale	Tutti i partecipanti

# **Indice**

<b>1.</b>	<b><i>INTRODUZIONE</i></b>	<b>4</b>
1.1.	Purpose .....	4
1.2.	Design Goals .....	4
1.3.	Definitions, acronyms, and abbreviations.....	5
1.4.	References .....	6
1.5.	Overview.....	6
<b>2.</b>	<b>CURRENT SYSTEM.....</b>	<b>7</b>
<b>3.</b>	<b><i>PROPOSED SYSTEM</i></b> .....	<b>7</b>
3.1.	Overview.....	7
3.2.	Subsystem decomposition.....	8
3.3.	Hardware/software mapping .....	10
3.4.	Persistent data management.....	10
3.5.	Access control and security.....	11
3.6.	Global Software Control .....	11
3.7.	Boundary Conditions.....	11
4.	Subsystem Services .....	12
<b>5.</b>	<b>GLOSSARY</b> .....	<b>13</b>

# 1. INTRODUZIONE

## 1.1 Purpose

Lo scopo di *FantaUnisa* è la progettazione e lo sviluppo di un'applicazione web che supporti i fantallenatori nella scelta della formazione ideale da schierare ogni settimana, attraverso l'elaborazione di statistiche aggiornate e l'utilizzo di un algoritmo configurabile.

Il sistema fornisce un ambiente semplice e interattivo in cui l'utente può:

- gestire la propria rosa di 25 calciatori;
- selezionare il modulo preferito;
- ricevere automaticamente una formazione consigliata in base ai dati più recenti, alla forma dei giocatori e alla difficoltà della prossima partita.

Il calcolo della formazione è affidato a un componente interno che combina statistiche di rendimento, trend stagionali e difficoltà dell'avversario, sulla base dei parametri impostati dal *Gestore degli algoritmi*.

## 1.2 Design Goals

### 1.2.1 List of design goals

- **DG\_1 – Tempo di risposta**

Il sistema deve rispondere alle richieste critiche (generazione formazione, aggiornamento dati, caricamento rosa) in meno di 2 secondi.

- **DG\_2 – Persistenza**

I dati relativi a utenti, rose, statistiche e indici di difficoltà devono essere conservati in modo sicuro e permanente su database.

- **DG\_3 – Privacy dei dati**

Solo utenti autorizzati possono accedere ai dati personali, alle rose e alle formazioni. Le password devono essere memorizzate tramite hashing.

- **DG\_4 – Robustezza**

Il sistema deve gestire file Excel non validi, dati incompleti, errori di inserimento o inconsistenze senza provocare malfunzionamenti.

- **DG\_5 – Readability**

Il codice deve essere strutturato in moduli (MVC) per favorire manutenibilità e modularità.

- **DG\_6 – Tracciabilità dei requisiti**

Ogni requisito funzionale deve essere mappato a uno o più casi d'uso e servizi.

- **DG\_7 – Usability**

L'interfaccia deve essere intuitiva, chiara e accessibile da qualsiasi dispositivo web, garantendo piena fruibilità anche da mobile tramite layout responsive.

- **DG\_8 – Scalabilità**

Il sistema deve supportare un elevato numero di utenti senza subire un considerevole degrado di prestazioni.

### **1.3 Definitions, acronyms, and abbreviations**

NOME	TIPO	DESCRIZIONE
<b>AI Engine</b>	Def.	Modulo interno che gestisce algoritmi di calcolo dell'indice di difficoltà e della formazione ottimale.
<b>Data Layer</b>	Def.	Livello architetturale responsabile della persistenza dei dati.
<b>Application Layer</b>	Def.	Livello in cui risiede la logica applicativa e i servizi del sistema.
<b>Presentation Layer</b>	Def.	Livello dell'interfaccia utente accessibile tramite browser.
<b>ORM</b>	Abbr.	Strato software che mappa le entità applicative alle tabelle del database (Object-Relational Mapping).
<b>MVC</b>	Abbr.	Architettura adottata dal sistema per separare logica, presentazione e gestione dati (Model-View-Controller).
<b>REST API</b>	Abbr.	Interfacce di comunicazione basate su architettura stateless utilizzate tra frontend, backend e moduli interni.
<b>RBAC</b>	Abbr.	Modello di autorizzazione basato sui ruoli degli utenti (Role-Based Access Control).
<b>MySQL</b>	Abbr.	DBMS relazionale utilizzato per la persistenza dei dati.
<b>Logging System</b>	Def.	Componente che registra eventi critici e diagnostici.

<b>Secure Hashing</b>	<i>Def.</i>	Tecnica utilizzata per la protezione delle password basata sulla crittografia.
<b>HTTP</b>	<i>Abbr.</i>	Protocollo di comunicazione utilizzato tra client e sistema.
<b>Session Management</b>	<i>Def.</i>	Meccanismo di gestione delle sessioni utente, incluso timeout e logout automatico. Vince chi fa più gol.

## 1.4 References

- Problem Statement FantaUnisa.
- Requirements Analysis Document FantaUnisa.

## 1.5 Overview

Il documento descrive l'architettura software proposta, la decomposizione in sottosistemi, il mapping HW/SW, la gestione della persistenza, le politiche di accesso (RBAC) e i servizi principali. Fornisce inoltre la matrice degli accessi con le operazioni autorizzate e descrive i flussi eventi principali (event-driven) tra sottosistemi.

## 2. CURRENT SYSTEM

Le piattaforme esistenti per il fantacalcio (**Fantacalcio.it**, **Leghe Fantacalcio**, **Gazzetta dello Sport**) adottano architetture monolitiche o microservizi dedicate ad ampi set di funzionalità come mercato, scambi, voti giornalieri.

Tali piattaforme presentano limiti rispetto agli obiettivi di FantaUnisa:

- non permettono la generazione automatica della formazione basata su un mix personalizzato di fantamedia + indice difficoltà;
- non permettono caricamenti personalizzati di dati tramite Excel;
- non consentono un ambiente di per analisi AI.

## 3. PROPOSED SYSTEM

### 3.1 Overview

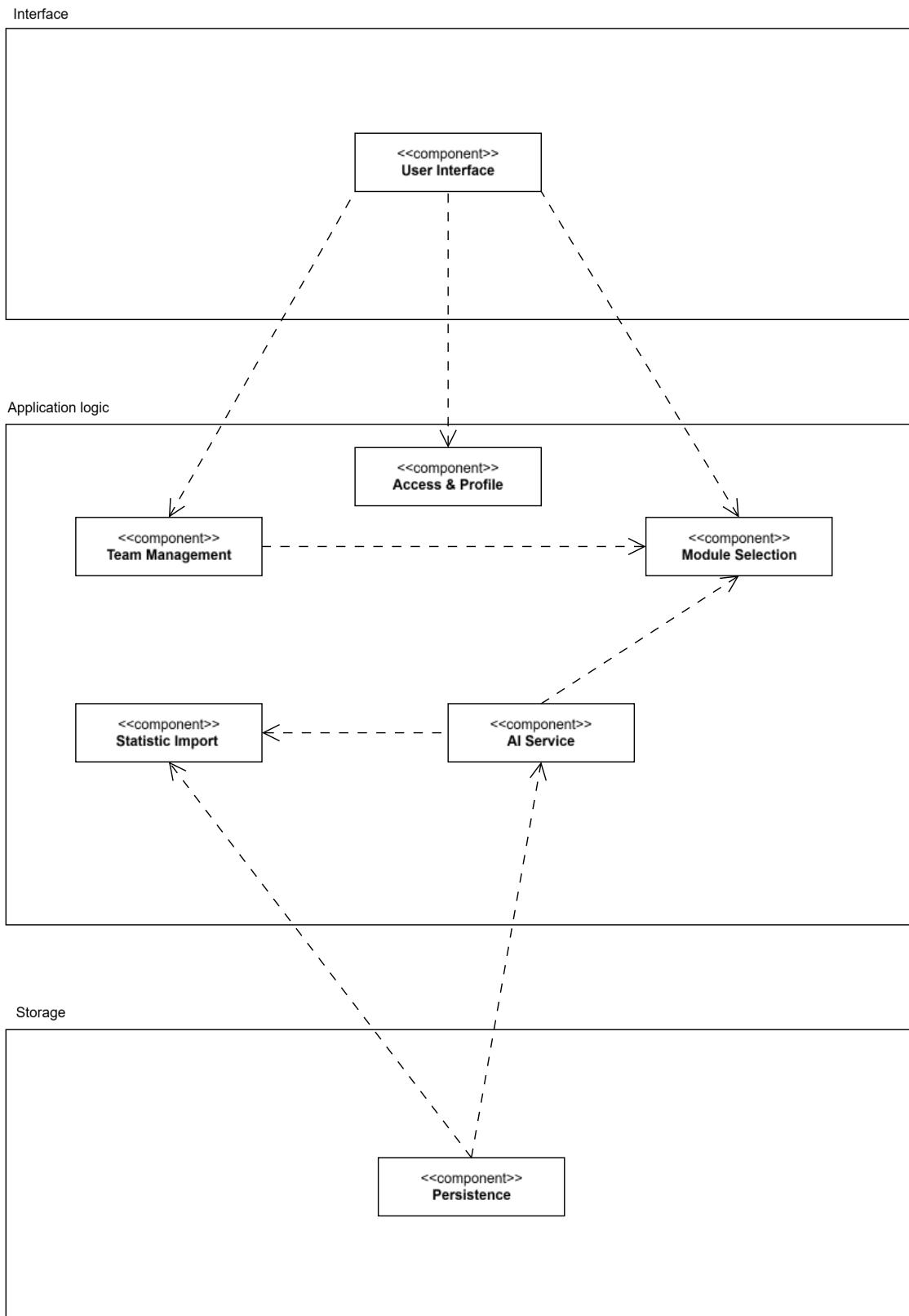
L'architettura software di FantaUnisa è basata su un modello **modulare e stratificato (MVC)**, composto da:

- **Presentation Layer** → UI Web
- **Application Layer** → Gestione algoritmi, gestione rosa, servizi utente
- **Data Layer** → DB relazionale MySQL con ORM

I moduli sono altamente separati per responsabilità e comunicano tramite API interne RESTful.

## 3.2 Subsystem decomposition

### 3.2.1 Subsystem diagram



### **3.2.2 Sottosistemi individuati**

#### **1. Access&Profile**

Gestisce registrazione, autenticazione, logout e profilo utente.

#### **2. TeamManagement**

Gestisce la creazione e modifica della rosa da parte dell'utente.

#### **3. ModuleSelection**

Gestisce la configurazione del modulo (4-3-3, 3-4-3 ecc.).

#### **4. StatisticsImport**

Gestisce il caricamento del file Excel e la validazione dei dati.

#### **5. DifficultyIndexCalculator (AI Engine)**

Calcola l'indice di difficoltà di ciascuna squadra.

#### **6. IdealFormationEngine (AI Engine)**

Genera la formazione ideale sulla base di:

- fantamedie
- indice difficoltà avversario
- modulo selezionato
- rosa utente

#### **7. StatisticsViewer**

Visualizza statistiche dei giocatori e dati carichi.

#### **8. Persistence**

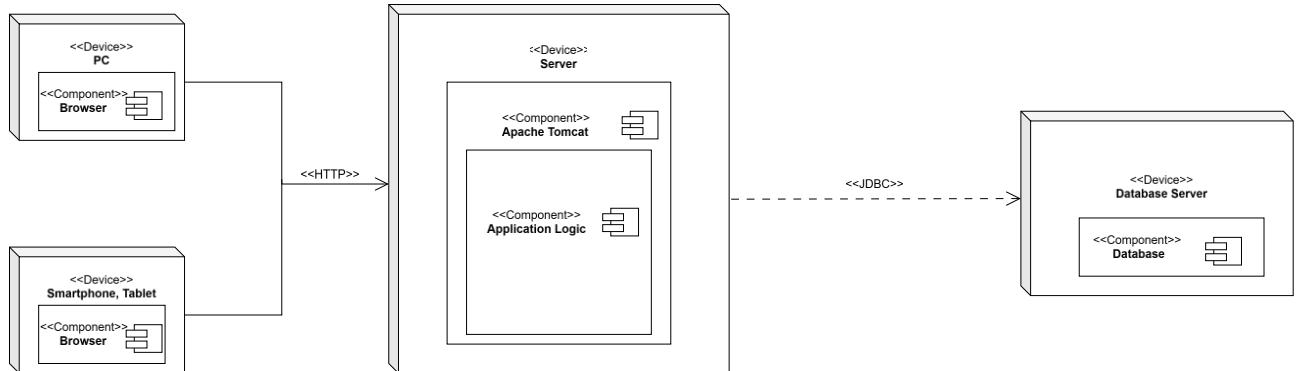
Si occupa di salvare e recuperare tutte le informazioni.

### **3.2.3 Architettura a layer**

- **Presentation Layer:** pagine HTML/JS che dialogano con l'Application Layer
- **Application Layer:** espone servizi REST e gestisce la logica
- **Data Layer:** DB MySQL, ORM, tabelle di giocatori e statistiche

### 3.3 Hardware/software mapping

#### 3.3.1 Deployment/Component Diagram



#### 3.3.2 HW/SW Specifications

##### Componenti Software

- Web Frontend (HTML/CSS/JS)
- Backend MVC (Java)
- AI Module (script dedicati)
- ORM per comunicazione DB
- MySQL Server
- Protocollo HTTP

##### Componenti Hardware

- Client (PC/mobile)
- Application Server
- DB Server

### 3.4 Persistent data management

Fare riferimento al file “**Persistent Data**” che verrà prodotto successivamente.

### **3.5 Access control and security**

Il sistema utilizza **RBAC** con i seguenti ruoli:

- **Visitatore**: può solo registrarsi
- **Utente registrato**: può gestire rosa, modulo, formazione, consultare statistiche
- **Gestore di dati statistici**: può caricare file Excel e ricalcolare statistiche

#### **3.5.1 Matrice degli Accessi**

Ruolo/ Oggetto	Account Utente	Rosa Utente	Modulo	File Excel	Statistiche Giocatore	Formazione Ideale
<b>Visitatore</b>	Registrazione	—	—	—	—	—
<b>Fanta allenatore</b>	Login Logout Modifica	Inserimento Modifica Elimina	Selezione Modifica	—	Visualizza Confronta	Genera Visualizza Salva
<b>Gestore di dati statistici</b>	Login Logout	—	—	Carica Verifica Modifica	Aggiorna Sovrascrivi	—

### **3.6 Global Software Control**

Il sistema utilizza un modello event-driven per:

- notificare l'utente del completamento del caricamento statistiche;
- avviare automaticamente i calcoli degli indici;
- avviare la generazione formazione.

### **3.7 Boundary Conditions**

#### **Inizializzazione**

- verifica connessione DB
- caricamento configurazioni
- inizializzazione logging

## **Terminazione**

- chiusura connessioni DB
- salvataggio stati pendenti

## **Fault Tolerance**

- rollback transazioni
- ritentativi su upload falliti
- logging diagnostico

## **4. Subsystem Services**

### **4.1 Access & Profile Services**

#### **Registration Service**

- verifica dati
- creazione account
- e-mail di conferma

#### **Authentication Service**

- login
- logout

#### **Profile Management Service**

- modifica credenziali
- modifica dati profilo

### **4.2 Team Management Services**

#### **Rosa Management Service**

- inserimento giocatori
- modifica giocatori
- rimozione giocatori
- validazione ruoli e quantità

### **4.3 Module Selection Services**

#### **Module Configuration Service**

- selezione modulo
- aggiornamento modulo predefinito

## 4.4 Statistics Import Services

### Excel Import Service

- upload file
- validazione struttura
- parsing dati
- aggiornamento statistiche

## 4.5 AI Services

### Difficulty Index Service

- calcolo indice difficoltà squadre
- salvataggio risultati

### Formation Optimization Service

- generazione formazione ideale
- ricalcolo su nuovo modulo

## 4.6 Statistics Viewer Services

### Statistics Visualization Service

- visualizzazione fantamedie
- storico prestazioni
- consultazione indici difficoltà

## 5. GLOSSARY

NOME	DESCRIZIONE
<b>Subsystem</b>	Componente architetturale che include più servizi o moduli con responsabilità omogenee.
<b>Componente software</b>	Elemento modulare che realizza una funzione specifica del sistema, spesso parte di un sottosistema più ampio.
<b>Endpoint</b>	URL che espone un servizio del backend tramite API REST e che può essere invocato dall'interfaccia o da altri moduli interni.
<b>Module Selection</b>	Sottosistema che gestisce la scelta e la modifica del modulo utilizzato dall'utente per generare la formazione ideale.

<b>Parsing</b>	Processo di analisi del contenuto del file Excel al fine di estrarre le informazioni e convertirle in strutture dati interne.
<b>Rollback</b>	Azione che annulla una transazione in caso di errore, riportando il sistema a uno stato consistente.
<b>Persistence Component</b>	Parte del sistema che si occupa della memorizzazione e recupero dei dati dal database, implementata tramite ORM e repository.
<b>Statistics Import</b>	Sottosistema incaricato dell'elaborazione dei file Excel contenenti dati aggiornati dei giocatori.