

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

**FantaUnisa**  
**System Design Document**  
**Versione 2.0**



Data: 21/01/2026

Progetto: FantaUnisa	Versione: 2.0
Documento: System Design Document	Data: 21/01/2026

**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
05/12/2025	1.1	Aggiunta specifica/descrizione di operazioni e servizi	Sabetta Francesco
18/01/2026	1.2	Rinnovamento grafici e piccole modifiche generali	Sabetta Francesco
21/01/2026	2.0	Revisione finale	Tutti i partecipanti

## Indice

<b>1. INTRODUZIONE.....</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. CURRENT SYSTEM.....</b>	<b>7</b>
<b>3. PROPOSED SYSTEM.....</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 .....	12
4. Subsystem Services.....	12
<b>5. GLOSSARY.....</b>	<b>17</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 di dati statistici**.

## 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>	<i>Def.</i>	Modulo interno che gestisce algoritmi di calcolo dell'indice di difficoltà e della formazione ottimale.
<b>Data Layer</b>	<i>Def.</i>	Livello architetturale responsabile della persistenza dei dati.
<b>Application Layer</b>	<i>Def.</i>	Livello in cui risiede la logica applicativa e i servizi del sistema.
<b>Presentation Layer</b>	<i>Def.</i>	Livello dell'interfaccia utente accessibile tramite browser.
<b>ORM</b>	<i>Abbr.</i>	Strato software che mappa le entità applicative alle tabelle del database (Object-Relational Mapping).
<b>MVC</b>	<i>Abbr.</i>	Architettura adottata dal sistema per separare logica, presentazione e gestione dati (Model-View-Controller).
<b>REST API</b>	<i>Abbr.</i>	Interfacce di comunicazione basate su architettura stateless utilizzate tra frontend, backend e moduli interni.
<b>RBAC</b>	<i>Abbr.</i>	Modello di autorizzazione basato sui ruoli degli utenti (Role-Based Access Control).
<b>MySQL</b>	<i>Abbr.</i>	DBMS relazionale utilizzato per la persistenza dei dati.
<b>Logging System</b>	<i>Def.</i>	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 il **caricamento personalizzato** dei voti tramite file esterni (Excel/CSV), vincolando l'utente ai dati ufficiali.
- Non integrano strumenti nativi per il **suggerimento automatico** della formazione basato su criteri oggettivi.
- Offrono funzionalità di **community** (social) spesso scollegate dal contesto operativo della gestione squadra.

## 3. PROPOSED SYSTEM

### 3.1 Overview

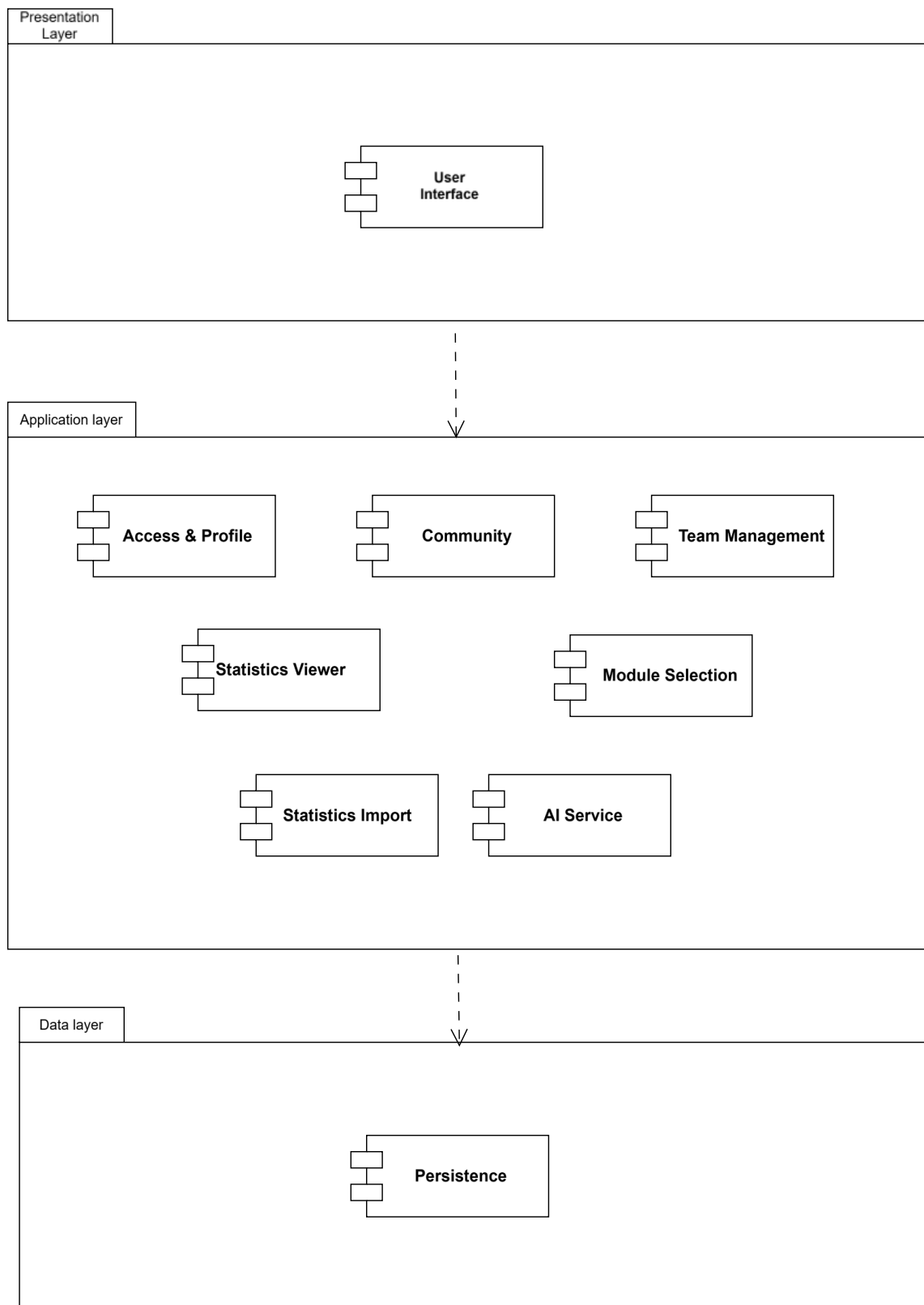
L'architettura software di **FantaUnisa** è basata su un modello **modulare e stratificato (MVC)**, composto da:

- **Presentation Layer** → UI Web per la visualizzazione dati.
- **Application Layer** → Logica di controllo e gestione servizi.
- **Data Layer** → Gestione della persistenza su DB relazionale (MySQL).

I moduli sono separati in base alle funzionalità e comunicano direttamente all'interno del sistema per garantire semplicità ed efficienza.

## 3.2 Subsystem decomposition

### 3.2.1 Subsystem diagram





### 3.2.2 Sottosistemi individuati

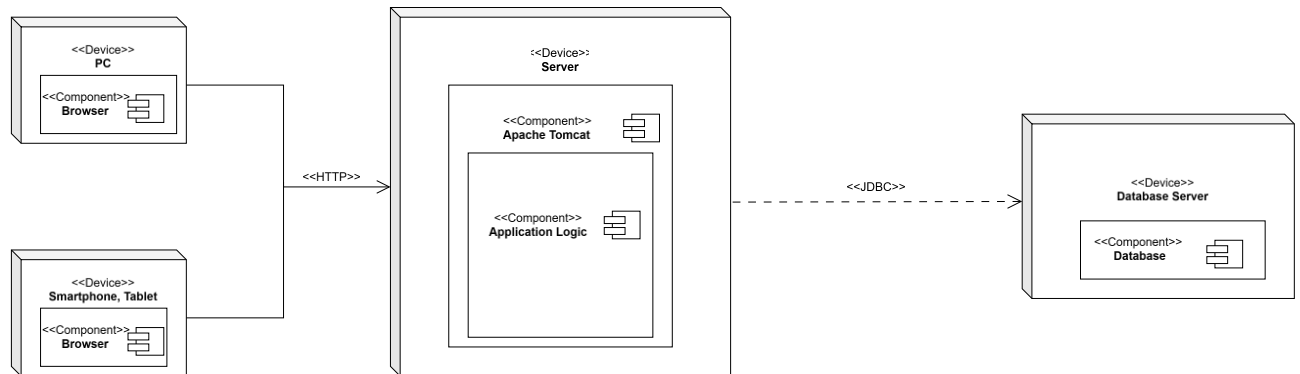
1. **Access&Profile**  
Gestisce registrazione, autenticazione, logout e profilo utente.
2. **TeamManagement**  
Gestisce la creazione e modifica della rosa da parte dell'utente.
3. **ModuleSelection**  
Gestisce la configurazione del modulo (4-3-3, 3-4-3 ecc.).
4. **StatisticsImport**  
Gestisce il 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 caricati.
8. **Community**  
Permette l'interazione tra utenti.

### 3.2.3 Architettura a layer

- **Presentation Layer:** pagine HTML/JS che dialogano con l'Application Layer
- **Application Layer:** espone servizi e gestisce la logica
- **Data Layer:** DB MySQL, DAO, 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)
- DAO 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;
- **Gestore utenti**: gestisce le interazioni nella community.

#### 3.5.1 Matrice degli Accessi

Ruolo/ Oggetto	Account Utente	Rosa Utente	Modulo	File Excel	Statistiche Giocatore	Formazione Ideale	Post	Commento	Reazione
<b>Visitatore</b>	Registrazione	—	—	—	—	—	—	—	—
<b>Fanta allenatore</b>	Login, Logout, Modifica	Inserisci, Modifica, Elimina	Selezione, Modifica	—	Visualizza, Confronta	Genera, Visualizza, Salva	Aggiungi, Rimuovi, Visualizza, Modifica	Aggiungi, Rimuovi, Visualizza, Modifica	Aggiungi, Rimuovi, Visualizza, Modifica
<b>Gestore di dati statistici</b>	Login, Logout	—	—	Carica, Verifica, Modifica	Aggiorna, Sovrascrivi	—	—	—	—
<b>Gestore utenti</b>	Blocca, Banna	—	—	—	—	—	Visualizza, Elimina	Visualizza, Elimina	Visualizza, Elimina

### 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

Qui vengono presentati i servizi che ogni sottosistema fornisce nel sistema software.

### 4.1 Access & Profile Services

Questi servizi permettono all'utente di registrarsi, autenticarsi e di modificare il proprio profilo utente.

#### Registration Service

Operazione	Descrizione
<b>verifyRegistrationData(registrationData)</b>	Verifica correttezza formale e unicità dei dati.
<b>createUserAccount(registrationData)</b>	Salva un nuovo utente nel sistema.
<b>createManagerAccount(registrationData)</b>	Salva un nuovo gestore nel sistema.
<b>sendActivationEmail(userEmail)</b>	Invia l'e-mail di conferma registrazione.
<b>confirmActivation(confirmData)</b>	Conferma la registrazione dell'utente.

## Authentication Service

Operazione	Descrizione
<b>loginUser(username, password)</b>	Verifica le credenziali e restituisce l'utente autenticato.
<b>logoutUser(sessionId)</b>	Invalida la sessione corrente.
<b>requestResetPassword(email)</b>	Avvia la procedura di reset password generando un token temporaneo e inviando una mail all'utente.
<b>resetPassword(token, newPassword)</b>	Verifica la validità del token e imposta una nuova password tramite hashing sicuro.

## Profile Management Service

Operazione	Descrizione
<b>modifyPassword(userId, currentPwd, newPwd, repeatedNewPwd)</b>	Aggiorna la password dopo aver verificato quella precedente.
<b>modifyProfile(userId, userData)</b>	Modifica i dati del profilo dell'utente.

## 4.2 Team Management Services

Questi servizi permettono la gestione della propria formazione.

### Rosa Management Service

Operazione	Descrizione
<b>addPlayerToSquad(userId, playerId)</b>	Aggiunge un giocatore alla rosa dell'utente.
<b>removePlayerFromSquad(userId, playerId)</b>	Rimuove un giocatore dalla rosa.
<b>saveSquad(userId, squadData)</b>	Salva l'intera rosa dell'utente.
<b>validateSquadComposition(squadData)</b>	Verifica che la rosa rispetti i vincoli dei ruoli.

### 4.3 Module Selection Services

Questi servizi permettono la configurazione del modulo della propria formazione.

#### Module Configuration Service

Operazione	Descrizione
<b>setModule(userId, module)</b>	Salva il modulo selezionato dall'utente.
<b>getModule(userId)</b>	Restituisce il modulo correntemente impostato.

### 4.4 Statistics Import Services

Questi servizi permettono il caricamento e la verifica del file delle statistiche dei giocatori.

#### Excel Import Service

Operazione	Descrizione
<b>uploadStatsFile(file)</b>	Carica il file Excel in piattaforma.
<b>validateStatsFile(file)</b>	Controlla che la struttura del file sia valida.
<b>parseStatsFile(file)</b>	Estrae i dati dal file.
<b>updatePlayerStats(statList)</b>	Aggiorna le statistiche dei giocatori nel database.

### 4.5 AI Services

Questi servizi permettono l'integrazione e la messa in atto del modulo IA.

#### Difficulty Index Service

Operazione	Descrizione
<b>computeTeamDifficulty(teamId)</b>	Calcola l'indice di difficoltà di una squadra.
<b>computeAllDifficulties()</b>	Ricalcola tutti gli indici di difficoltà.
<b>saveDifficultyIndex(teamId, index)</b>	Salva l'indice nel database.

### Formation Optimization Service

Operazione	Descrizione
<b>computeIdealFormation(userId, module)</b>	Calcola la formazione ideale.
<b>evaluatePlayerPotential(playerId)</b>	Stima il punteggio potenziale del giocatore.

### 4.6 Statistics Viewer Services

Questi servizi permettono la visualizzazione delle varie statistiche inerenti a giocatori e squadre.

#### Statistics Visualization Service

Operazione	Descrizione
<b>getPlayerStats(playerId)</b>	Restituisce le statistiche aggiornate del giocatore.
<b>getUserSquadStats(userId, squad)</b>	Restituisce le statistiche dei giocatori della rosa.
<b>getDifficultyRanking()</b>	Restituisce la classifica di difficoltà aggiornata.

### 4.7 Community Services

Questo sottosistema consente l'interazione sociale tra utenti della piattaforma, attraverso la pubblicazione di formazioni e la possibilità di commentarle.

#### Formation Sharing Service

Operazione	Descrizione
<b>createPost(userId, formation, description)</b>	Permette all'utente di pubblicare una formazione accompagnata da un testo descrittivo.
<b>deletePost(userId, postId)</b>	Rimuove un proprio post dalla community.
<b>getPost(postId)</b>	Restituisce i dettagli di un singolo post.
<b>getAllPosts()</b>	Restituisce l'elenco dei post pubblicati dagli utenti.

### Comment Service

Operazione	Descrizione
<b>addComment(userId, postId, text)</b>	Inserisce un commento sotto un post pubblico.
<b>deleteComment(userId, commentId)</b>	Permette all'autore o al gestore di cancellare un commento.
<b>getComments(postId)</b>	Restituisce tutti i commenti associati a un post.

### Reaction Service

Operazione	Descrizione
<b>addReaction(userId, postId, reactionType)</b>	Aggiunge una reazione (es. "like") a un post.
<b>removeReaction(userId, postId)</b>	Rimuove una reazione precedentemente inserita.
<b>getReactions(postId)</b>	Restituisce le reazioni associate al post.

### Report Service

Operazione	Descrizione
<b>checkReport(userId, postId, commentId)</b>	Visualizza il commento/post segnalato da un utente.
<b>removeComment(userId, postId)</b>	Rimuove il commento dal post
<b>banUser(userId)</b>	Blocca l'accesso al portale dell'utente
<b>validateReport(userId, reportId)</b>	L'utente che ha effettuato la segnalazione riceve un aggiornamento



## 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.