

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

**TrackFit  
SystemDesignDocument\_TrackFit  
Versione 1.0**

**LOGO PROGETTO**



Data: 20/11/2025

Progetto: TrackFit	Versione: 1.0
Documento: SystemDesignDocument_TrackFit	Data: 20/10/2025

Coordinatore del progetto:

Nome	Matricola
Gabriele di Palma	0512119257

Partecipanti:

Nome	Matricola
Yuri Fierro	0512119524

Scritto da:	Gabriele di Palma
-------------	-------------------

#### Revision History

Data	Versione	Descrizione	Autore
20/10/2025	1.0	Prima stesura dell'SDD	GDP, YF
			GDP.YF

# **Indice**

1.	INTRODUZIONE .....	4
1.1.	Scopo del sistema .....	4
1.2.	Obiettivi di design.....	4
1.3.	Riferimenti .....	5
2.	SISTEMA CORRENTE .....	5
3.	ARCHITETTURA DEL SISTEMA PROPOSTO .....	6
3.1	Panoramica.....	6
3.2	Decomposizione in sottosistemi.....	6
3.3	Hardware/Software mapping.....	7
3.4	Gestione dati persistenti.....	7
3.5	Controllo di accessi e sicurezza .....	8
3.5.1	Controllo globale del software .....	9
3.6	Boundary conditions .....	9
4.	SERVIZI DEI SOTTOSISTEMI .....	10

# 1. INTRODUZIONE

## 1.1. *Scopo del sistema*

**TrackFit** si propone come una piattaforma digitale unificata per il monitoraggio e l'analisi delle attività sportive multidisciplinari. Il sistema è rivolto a 3 tipologie di utenti:

- **Utenti finali:** utenti amatoriali o semi-professionali che desiderano monitorare i propri allenamenti, visualizzare progressi attraverso dashboard personalizzate e gestire obiettivi sportivi.
- **Allenatori:** professionisti che seguono gli atleti, gestiscono piani di allenamento, monitorano le performance dei loro assistiti e offrono sessioni di coaching personalizzate.
- **Gestori di palestre:** amministratori di strutture sportive che gestiscono l'infrastruttura fisica, i corsi, gli abbonamenti, il controllo degli accessi e la reportistica sulle attività della struttura
- **Amministratori di sistema:** figure con privilegi elevati che supervisionano l'intero ecosistema TrackFit, gestiscono le strutture registrate e controllano la qualità del servizio

## 1.2. *Obiettivi di design*

Gli utenti principali di TrackFit necessitano di un'esperienza d'uso semplice, intuitiva e altamente personalizzabile. Per rispondere alle esigenze di un sistema che gestisce dati sensibili e deve garantire disponibilità continua, si identificano i seguenti obiettivi di design prioritari:

- **Disponibilità:** il sistema deve essere accessibile 24/7 per permettere agli utenti di registrare allenamenti e consultare dati in qualsiasi momento.
- **Scalabilità:** il sistema deve supportare la crescita del numero di utenti, palestre e sessioni di allenamento senza degrado delle performance.
- **Performance:** i tempi di risposta per operazioni comuni devono essere inferiori a 2 secondi
- **Sicurezza:** il sistema gestisce dati sensibili, e bisognerà garantire che questi dati siano sicuri
- **Gestione dei malfunzionamenti:** le transazioni economiche e le operazioni critiche devono rispettare le proprietà ACID
- **Interoperabilità:** il sistema deve supportare l'importazione di dati da formati standard (GPX, TCX) e l'integrazione con API esterne per arricchire lo storico degli allenamenti
- **Usabilità e Accessibilità:** l'interfaccia deve essere responsive, intuitiva anche per utenti non esperti, e supportare almeno la lingua italiana e inglese

## 1.3. *Riferimenti*

- **RAD\_TrackFit**
- **Materiale didattico del corso di Ingegneria del Software**

## 2. SISTEMA CORRENTE

Non è presente un sistema corrente per l'applicazione da sviluppare.

## 3. ARCHITETTURA DEL SISTEMA PROPOSTO

### 3.1 *Panoramica*

Nelle prossime sezioni andremo a definire l'architettura del sistema in termini di:

- Decomposizione in sottosistemi: si propone un'architettura a tre livelli con suddivisione tra interfacce, logica applicativa e logica di persistenza.
- Hardware/software mapping:
- Gestione dei dati persistenti:
- Controllo di accessi e sicurezza:
- Controllo globale del software:
- Boundary conditions:

### 3.2 *Decomposizione in sottosistemi*

Per la decomposizione in sottosistemi sono stati analizzati i requisiti funzionali e non funzionali documentati nel RAD. Considerando che TrackFit è un'applicazione che richiede elevati standard di scalabilità, sicurezza e manutenibilità, è stata adottata un'**architettura a tre livelli** (Three-Tier Architecture) in cui si distinguono:

1. **Presentation Layer (Livello di Presentazione)**: interfacce utente differenziate per ruolo.
2. **Application Logic Layer (Livello di Logica Applicativa)**: servizi di business logic e orchestrazione.
3. **Data Persistence Layer (Livello di Persistenza)**: gestione dell'accesso ai dati e storage.

#### Livello 1: Presentation Layer

Questo livello comprende le interfacce utente dedicate ai diversi attori del sistema. Tutte le interfacce comunicano con il livello applicativo sottostante tramite **API RESTful** protette:

- **UserInterface (Web Application)**: Applicazione web responsive sviluppata in **Next.js/React/TypeScript**, accessibile da browser desktop e mobile.
- **MobileApp**: Applicazione mobile nativa o Progressive Web App (PWA) ottimizzata per dispositivi iOS e Android.
- **CoachInterface**: Dashboard dedicata per gli allenatori con funzionalità specifiche quali gestione atleti, modifica piani di allenamento e reportistica.
- **GymManagerInterface**: Portale per i gestori delle palestre, focalizzato su funzionalità amministrative come la gestione dei corsi, degli abbonamenti e la reportistica della struttura.

- **AdminInterface:** Pannello di amministrazione per i supervisori del sistema, con accesso completo alle funzionalità di configurazione e monitoraggio globale.

## Livello 2: Application Logic Layer (Livello di Logica Applicativa)

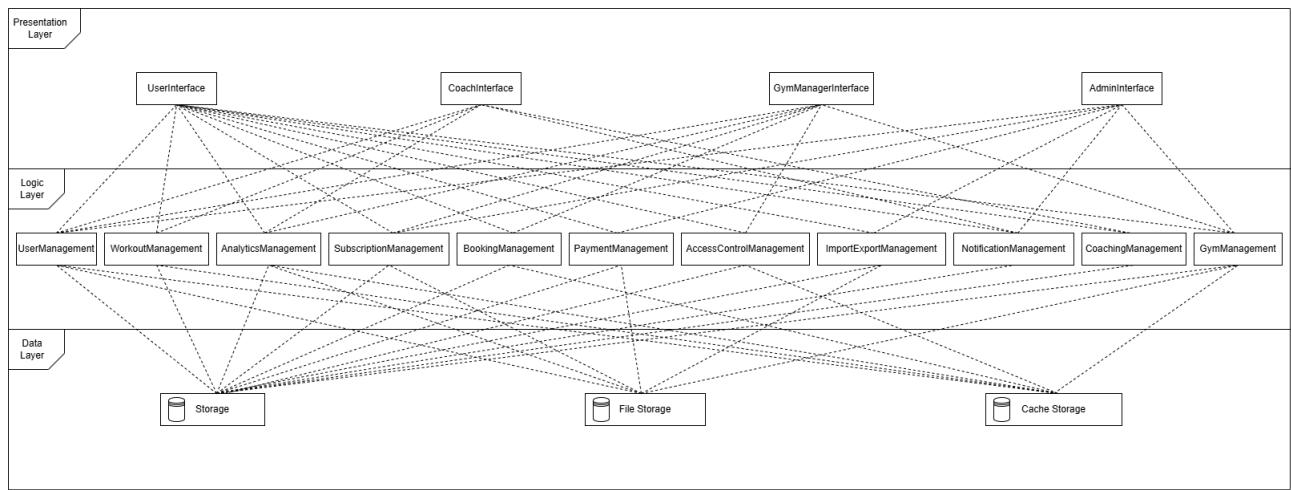
Il cuore del sistema è suddiviso in sottosistemi, ciascuno responsabile di un dominio funzionale specifico. I sottosistemi comunicano tra loro quando necessario (garantendo alta coesione funzionale) e accedono al livello di persistenza per le operazioni CRUD.

- **UserManagement:** Gestione dell'autenticazione, autorizzazione, profili utente, ruoli e permessi.
- **WorkoutManagement:** Registrazione, modifica, cancellazione e consultazione delle sessioni di allenamento.
- **AnalyticsManagement:** Generazione di dashboard, grafici, statistiche aggregate e report personalizzati.
- **GymManagement:** Gestione delle strutture sportive, definizione dei corsi, orari, capacità delle sale e configurazioni.
- **SubscriptionManagement:** Gestione degli abbonamenti, tipologie, durate, rinnovi automatici e validazioni di stato.
- **BookingManagement:** Gestione delle prenotazioni ai corsi, liste d'attesa, cancellazioni e notifiche associate.
- **CoachingManagement:** Gestione degli allenatori, disponibilità, prenotazioni di sessioni individuali e creazione piani di allenamento.
- **PaymentManagement:** Integrazione con gateway di pagamento, gestione transazioni, storico pagamenti e fatturazione.
- **AccessControlManagement:** Generazione tessere digitali (QR code), validazione degli accessi e registrazione dei log di ingresso.
- **NotificationManagement:** Servizio di invio notifiche push, email e SMS per conferme, promemoria e scadenze.
- **ImportExportManagement:** Importazione di allenamenti da file GPX/TCX e API esterne; esportazione di report in formati CSV/PDF.

## Livello 3: Data Persistence Layer (Livello di Persistenza)

Questo livello gestisce l'archiviazione e il recupero dei dati. Il pattern di accesso ai dati utilizza **JPA (Java Persistence API)** per l'astrazione dell'accesso al database, garantendo portabilità e manutenibilità del codice.

- **Storage:** Sistema di gestione del database relazionale (**MySQL**) per la memorizzazione dei dati strutturati.
- **FileStorage:** Sistema di archiviazione dedicato ai file binari (immagini profilo, documenti caricati, file GPX/TCX).
- **CacheLayer:** Layer di caching (**Redis**) implementato per migliorare le performance delle query frequenti e ridurre il carico sul database principale.

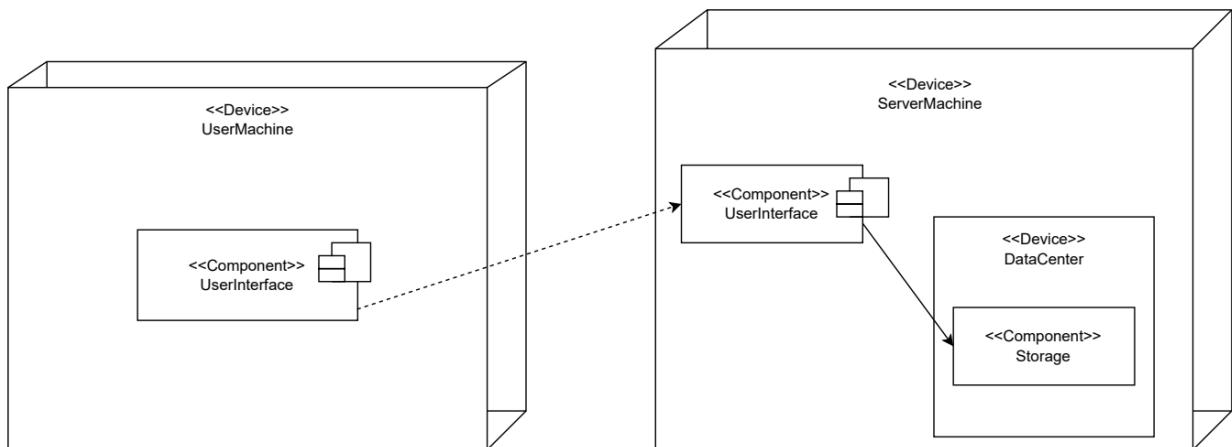


### 3.3 Hardware/Software mapping

Il sistema TrackFit è progettato come un'applicazione distribuita basata su un'architettura Client-Server. I client (le diverse tipologie di utenti) si connettono attraverso la rete internet al server centrale che elabora le richieste e gestisce la logica di business.

**- Lato Server:** Il core del sistema sarà sviluppato utilizzando il framework Spring Boot. Questa scelta permette di gestire efficacemente l'architettura a microservizi o modulare (come definito nella decomposizione), riducendo le configurazioni manuali. Spring Boot offre un server integrato (Tomcat) e una gestione ottimale delle dipendenze. Per la sicurezza e la gestione dei ruoli (Atleti, Allenatori, Gestori, Admin) verrà integrato il modulo Spring Security.

**-Lato Client:** L'interfaccia utente sarà realizzata come una Progressive Web App (PWA) ottimizzata per dispositivi mobili e desktop. Questo garantisce un'esperienza utente fluida e reattiva, comunicando con il backend tramite API REST. La PWA permette di servire tutte le interfacce specifiche (User, Coach, GymManager, Admin) attraverso un unico punto di accesso adattivo.



### 3.4 Gestione dei dati persistenti

Il sistema deve gestire la persistenza di dati eterogenei, che includono informazioni strutturate, file binari e dati ad accesso frequente.

Nello specifico, il sistema dovrà salvare in modo persistente:

- I profili degli utenti (Atleti, Allenatori, Gestori, Amministratori) e i relativi dati anagrafici.
- I piani di allenamento e le schede tecniche.
- I log delle attività e le metriche delle sessioni di allenamento.
- I dati relativi alle palestre, abbonamenti e transazioni economiche.
- File multimediali e tecnici (immagini profilo, file GPX/TCX importati).

Per garantire efficienza e integrità, la gestione dei dati è suddivisa su tre tecnologie:

**Relational Database** : Utilizzato per memorizzare tutti i dati strutturati (utenti, relazioni, allenamenti, pagamenti). L'interazione con il database avviene tramite JPA (Java Persistence API), che astrae l'accesso ai dati e garantisce la portabilità del codice. Questo assicura anche il rispetto delle proprietà ACID per le transazioni critiche (es. pagamenti).

**File Storage**: Sistema dedicato all'archiviazione di file binari di grandi dimensioni, come immagini o file di tracciamento (GPX), per non appesantire il database relazionale.

**Cache Layer** : Implementato per migliorare le performance, memorizzando temporaneamente i dati richiesti con maggiore frequenza (es. classifiche, dati di sessione attivi) e riducendo il carico sul database principale.

### 3.5 Controllo di accessi e sicurezza

La sicurezza del sistema è gestita attraverso una **Global Access Table**, che definisce rigorosamente quali operazioni ogni attore può eseguire su specifiche entità del sistema. Il componente *UserManagement* (o *AccessControlManagement*) interrogherà questa tabella per autorizzare o negare le richieste.

Attori/Oggetti	Training Plan	WorkOut	GymProfile	Subscription	UserProfile
Atleta	view, execute	create, update, import	search, view	purchase, view status	update, delete
Allenatore	create, assign, update	view (degli atleti), analyze	view	-	update info
Gestore Palestra	-	-	update, manage access	define, verify	view analytics
Admin	delete (se inapprop.)	delete	approve, ban	manage taxes	ban, manage roles

#### 3.5.1 Controllo Globale del Software

Il flusso di controllo del sistema TrackFit non risiede in un unico ciclo principale (main loop) scritto manualmente, ma è delegato al container del framework, che gestisce il ciclo di vita dell'applicazione e l'instradamento delle richieste. Il sistema adotta un modello di controllo Event-Driven (guidato dagli eventi), dove il flusso è attivato dalle azioni dell'utente sulla PWA.

Il controllo avviene secondo le seguenti modalità:

**DispatcherServlet (Front Controller):** È il componente centrale che intercetta tutte le richieste HTTP/HTTPS in arrivo dalla Progressive Web App. Agisce come "vigile urbano": analizza l'URL e il tipo di richiesta (GET, POST, PUT, DELETE) e la smista al Controller specifico del sottosistema competente (es. TrainingController per il salvataggio di un allenamento, AuthController per il login).

**Gestione della Concorrenza (Multi-threading):** Il web server integrato gestisce ogni richiesta in entrata assegnandole un thread separato prelevato da un pool. Questo permette a più utenti (Atleti, Coach, Gestori) di operare simultaneamente senza bloccarsi a vicenda.

**Elaborazione Asincrona:** Per operazioni computazionalmente onerose che potrebbero rallentare l'esperienza utente, come l'importazione e il parsing di file GPX/TCX o l'invio massivo di notifiche push, il controllo viene gestito in modo asincrono. Il sistema accetta la richiesta, restituisce immediatamente una conferma all'utente ("Elaborazione in corso...") e delega l'esecuzione effettiva a un thread separato in background, garantendo che l'interfaccia rimanga sempre reattiva.

**Gestione Globale delle Eccezioni:** Il flusso di controllo include un meccanismo centralizzato (tramite ControllerAdvice) che intercetta qualsiasi errore imprevisto o eccezione generata dai sottosistemi logici. Questo impedisce il crash dell'applicazione e

assicura che al client venga sempre restituita una risposta formattata correttamente (JSON di errore), mantenendo il sistema in uno stato consistente.

### 3.6 Boundary Conditions

Le boundary conditions rappresentano i casi limite e le situazioni eccezionali che possono verificarsi durante l'esecuzione del sistema. La gestione proattiva di queste condizioni è essenziale per garantire la robustezza e l'affidabilità di TrackFit.

Le condizioni sono:

- Picchi di carico: Durante orari di punta (es. tardo pomeriggio in palestra) o eventi promozionali, il server deve gestire numerose richieste concorrenti di "check-in" o caricamento dati.
- Integrità dei File: In caso di upload di file GPX corrotti o interrotti per caduta di connessione, il sistema deve scartare i dati parziali e notificare l'utente senza generare stati inconsistenti.
- Transazioni Fallite: Se un pagamento per un abbonamento fallisce (es. errore gateway bancario), il sistema deve garantire il rollback completo (nessun abbonamento attivato, nessun addebito parziale registrato).
- Manutenzione: L'Amministratore di sistema monitora i log di errore e interviene per il ripristino manuale in caso di guasti hardware o bug critici non gestiti automaticamente.

## 4. SERVIZI DEI SOTTOSISTEMI

In questa sezione vediamo i servizi offerti dai sottosistemi a livello di logica applicativa.

Ogni sottosistema espone un interfaccia ben definita dei servizi che implementano le funzionalità di sistema.

### User Management

Gestione degli utenti, autenticazione, autorizzazione, profili e ruoli.

Servizio	Descrizione
Autenticazione	Autenticazione utenti (login, logout, gestione sessioni)
Registrazione	Registrazione nuovi utenti con validazione dati e invio email conferma
Autorizzazione	-
Gestione del profilo	-
Gestione ruoli	-
Recupero password	tramite e-mail con token temporaneo
Eliminazione account	Cancellazione account e dati personali

## **Workout Management**

Gestione delle sessioni di allenamento

Servizio	Descrizione
Creazione Allenamento	Tramite pagina apposita
Modifica Allenamento	-
Cancellazione Allenamento	-
Importare dati sessione	Tramite pagina utente allenato

## **Gym Management**

Gestione delle strutture sportive

Servizio	Descrizione
Creazione Palestra	Tramite pagina apposita
Modifica Palestra	-
Eliminazione Palestra	-
Gestione Staff	Visualizza dati, fatturati ed ore all'interno
Informazione palestra	Visualizza struttura e dati in base ai dati prodotti dagli operatori

## **Subscription Management**

Gestione abbonamenti e validazioni

Servizio	Descrizione
Crea Iscrizione	Tramite pagina pagamento
Modifica Iscrizione	Tramite pagina apposita (Se è possibile)
Cancellazione Iscrizione	-
Storico dati	-
Applicazioni coupon	Con possibili vincoli su applicazioni multiple

## **Coaching Management**

Gestione allenatori, disponibilità e sessioni individuali.

Servizio	Descrizione
Crea Allenatore	Gestito da Manager
Elimina Allenatore	-
Gestisce disponibilità sessioni	L'utente tramite pagina apposita vede gli slot
Controlla lista atleti	L'allenatore deve vedere i suoi utenti

## **Payment Management**

Gestione pagamenti, transazioni e integrazioni con gateway esterni.

Servizio	Descrizione
Processo pagamento	Elaborazione tramite gateway esterno
Storico transazioni	Storico con filtri
Genera Fattura	-
Rimbors	Entro 15gg
Validità pagamento	Carta di credito

### **Access Control Management**

Controllo accessi fisici alle strutture tramite tessera digitale.

Servizio	Descrizione
Generazione Pass	Con qr o generazione fisica
Validazione	I tornelli devono permettere ingresso
log ingresso	Registrazione ingresso strutturate
Storico ingressi	Storico accesso fra varie palestre

### **Notification Management**

Invio notifiche multicanale(push,email,sms)

Servizio	Descrizione
Invio notifica push	A dispositivo Utente
Invio email	Invio email tramite servizio esterno
Invio sms	Solo per notifiche critiche
Notifica prenotazioni	Notifica automatica
Notifica Cancellazioni	Notifica cancellazione in corso
Notifica Scadenza	15 gg prima della scadenza
Notifica obiettivo	Notifica raggiungimento obiettivo inviabile da coach