

Università degli Studi di Salerno
Corso di Ingegneria del Software

TrackFit
RAD_TrackFit
Versione 1.1



Data: 10/11/2025

Progetto: TrackFit	Versione: 1.1
Documento: Casid'uso_TrackFit	Data: 10/11/2025

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
22/10/2025	1.0	Creazione del documento	GDP,YF
10/10/2025	1.1	Consegna del RAD completo	GDP,YF

Indice

1. INTRODUZIONE	4
1.1. Significato del sistema	4
1.2. Scopo del sistema	4
1.3. Obiettivi e criteri di successo del sistema	4
1.4. Definizioni,acronimi e abbreviazioni	5
1.5. Riferimenti	5
1.6. Panoramica	5
2. SISTEMA ATTUALE	5
2.1. Panoramica del sistema	5
2.2. Funzioni e limiti degli strumenti attuali	6
2.3. Punti di forza e debolezza	6
2.4. Requisiti non soddisfatti o mancanze	6
3. SISTEMA PROPOSTO	7
3.1. Panoramica del sistema proposto	7
3.2. Requisiti funzionali	7
3.2.1. Gestione utenti, allenatori e gestori della palestra.....	7
3.2.2. Gestione palestre	8
3.2.3. Monitoraggio delle performance	8
3.2.4. Reportistica e analisi aggregate	9
3.2.5. Accesibilità e interfaccia intuitiva	9
3.2.6. Gestione pagamenti e abbonamenti	9
3.2.7. Controllo accessi e gestione prenotazioni.....	10
3.2.8. Interoperabilità e dati	10
3.2.9. Gestione account e sicurezza.....	11
3.3. Requisiti non funzionali	11
3.3.1. Performance.....	11
3.3.2. Reliability	11
3.3.3. Usability	12
3.3.4. Supportability	12
3.3.5. Implementation.....	13
3.3.6. Interface	13
3.3.7. Packaging.....	14
3.3.8. Legal.....	14
3.3.9. Vincoli di dominio.....	15
3.4. Modello del sistema	15
3.4.1. Scenarios.....	15
3.4.1.1. Scenario 1 corsa serale	15
3.4.1.2. Scenario 2 gestione orari da parte dell'allenatore	16
3.4.1.3. Scenario 3 autenticazione (portale business)	17
3.4.1.4. Scenario 4 prenotazione allenatore	17
3.4.1.5. Scenario 5 attivazione abbonamento annuale con coupon	18
3.4.2. Use case model.....	19
3.4.3. Object model.....	53
3.4.4. Dynamic model	54

3.4.5. Mockups e diagramma navigazionale dell'utente	58
4. GLOSSARIO	0

1. INTRODUZIONE

1.1. Significato del sistema

Il sistema **TrackFit** nasce con l'obiettivo di fornire una piattaforma digitale semplice, accessibile e personalizzabile per il monitoraggio delle attività sportive.

La finalità principale è quella di supportare l'utente nel tenere traccia dei propri allenamenti, analizzare i progressi e mantenere costanza nel tempo, favorendo un approccio consapevole e motivante al benessere personale.

TrackFit mira inoltre a superare i limiti degli strumenti attualmente disponibili, spesso frammentati o specifici per singole discipline, offrendo una visione unificata e completa delle prestazioni sportive.

1.1. Scopo del sistema

TrackFit si colloca nel settore delle applicazioni per il **fitness e il benessere personale**, con particolare attenzione al monitoraggio e all'analisi dei dati di allenamento.

Il sistema è rivolto principalmente a utenti amatoriali o semi-professionisti che desiderano gestire in modo strutturato e intuitivo le proprie attività sportive.

TrackFit sarà accessibile tramite **interfaccia web e app mobile**, per garantire all'utente la possibilità di consultare e aggiornare i propri dati in qualunque momento.

Nel medio-lungo termine, il sistema potrà essere esteso con funzionalità di **integrazione con dispositivi indossabili** e servizi esterni per la raccolta automatica dei dati.

1.2. Obiettivi e criteri di successo del progetto

Obiettivi principali

- Realizzare un sistema software per il **monitoraggio e l'analisi delle sessioni di allenamento**.
- Consentire la **registrazione personalizzata delle attività**, con dati come tipo di sport, durata, distanza e calorie.
- Fornire strumenti per la **consultazione dello storico** e l'analisi dei progressi tramite grafici e dashboard interattive.
- Offrire funzionalità di **esportazione dei dati** (PDF, CSV) per garantire all'utente un controllo completo delle informazioni.
- Progettare un'interfaccia **semplice, intuitiva**.

Criteri di successo

- L'utente deve poter registrare e consultare facilmente i propri allenamenti.
- Il sistema deve garantire una **visualizzazione chiara e personalizzabile dei dati**.
- L'interfaccia deve risultare **intuitiva anche per utenti non esperti**.
- Le funzioni di analisi e report devono essere **accurate e immediate**.
- La piattaforma deve essere **scalabile**.

1.3. Definizione, acronimi e abbreviazioni

Termine / Acronimo	Definizione
TrackFit	Nome del sistema software sviluppato.
Dashboard	Interfaccia grafica che mostra dati e statistiche in forma sintetica e interattiva.
Workout	Sessione di allenamento svolta dall'utente.
CSV (Comma-Separated Values)	Formato di file testuale utilizzato per esportare dati tabellari.
UI (User Interface)	Interfaccia utente del sistema.
API (Application Programming Interface)	Interfaccia per l'integrazione con servizi esterni o dispositivi indossabili.

1.4. Riferimenti

Linee guida di progettazione per applicazioni fitness e wellness.
Documentazione tecnica su framework web e mobile (React, Typescript).
Ricerche e articoli su sistemi di monitoraggio sportivo e user experience nel fitness.
Materiale fornito dal corso o dal progetto di riferimento (specificare se esiste).

1.5. Panoramica

Il documento descrive le finalità, gli obiettivi e l'ambito del sistema **TrackFit**, delineando la visione generale del progetto.
TrackFit si propone come una soluzione innovativa per la gestione e l'analisi delle attività sportive, con particolare attenzione alla semplicità d'uso e alla completezza delle informazioni.
Le sezioni successive forniranno dettagli su architettura, requisiti funzionali, interfaccia utente e aspetti tecnologici necessari per la realizzazione del sistema.

2. SISTEMA ATTUALE

2.1. Panoramica del sistema

Il contesto attuale è caratterizzato da una significativa e crescente attenzione verso l'attività fisica e il benessere personale. Utenti a livello amatoriale e semi-professionale sentono l'esigenza di monitorare i propri allenamenti per migliorare le prestazioni, mantenere la costanza e prevenire infortuni.
Per rispondere a questa esigenza, gli utenti attualmente utilizzano diversi strumenti digitali disponibili sul mercato. Non esiste un "sistema attuale" unificato da sostituire; piuttosto, esiste un panorama di soluzioni tecnologiche che presentano diverse problematiche.

2.2. Funzioni e limiti degli strumenti attuali

Gli strumenti che gli utenti hanno a disposizione (applicazioni mobili, piattaforme web) sono spesso specializzati e limitati a specifiche categorie di attività.

Ad esempio, molte soluzioni sono "verticali", focalizzandosi esclusivamente su sport come la corsa o il ciclismo. Sebbene questi strumenti possano eccellere nel monitoraggio di quella singola disciplina, non offrono supporto per un monitoraggio multisport che includa, ad esempio, attività da palestra (pesi, circuiti) o altri sport misti.

2.3. Punti di forza e debolezza

Punti di Forza (Impliciti):

- Esistono soluzioni mature per il tracciamento di attività specifiche (es. corsa, ciclismo).

Debolezze/Problemi:

- **Frammentazione:** Le soluzioni attuali sono "frammentate". Gli utenti sono costretti a usare più applicazioni diverse per tracciare attività diverse.
- **Complessità:** Molti strumenti disponibili risultano "complessi" da utilizzare.
- **Specializzazione Eccessiva (Verticalità):** Le soluzioni sono "limitate a specifici tipi di attività".
- **Mancanza di Visione Unificata:** La debolezza principale è la "difficoltà di avere una visione unificata e completa dei propri progressi". L'utente non può analizzare in modo aggregato i dati provenienti da sport diversi.
- **Interfacce e Reportistica:** Si rileva la necessità di interfacce più "semplici e intuitive" e di report "chiari e facilmente interpretabili", suggerendo che le soluzioni attuali siano carenti sotto questi aspetti.

2.4. Requisiti non soddisfatti o mancanze

Il panorama attuale non soddisfa l'esigenza chiave degli utenti di "centralizzare le informazioni relative all'attività fisica" in un unico sistema.

Manca una piattaforma che:

1. Supporti nativamente diverse tipologie di attività fisica, inclusi sport di resistenza, attività da palestra e sport misti.
2. Offra una "visione unificata e completa" dei progressi.
3. Stimoli l'utente a "mantenere costanza e consapevolezza nel proprio percorso sportivo".
4. Presenti le informazioni in forma "chiara, accessibile e personalizzata".

3. SISTEMA PROPOSTO

3.1. Panoramica del sistema proposto

Il sistema proposto, denominato TrackFit, si posiziona nel dominio del monitoraggio e dell'analisi degli allenamenti multisport, rivolgendosi a un target di utenti amatoriali e semi-professionisti.

La motivazione principale del progetto nasce dalla constatazione che le soluzioni software attuali risultano spesso frammentate o "verticali" (cioè, specializzate su un singolo sport, come la corsa o il ciclismo). Questa frammentazione rende difficile per l'utente ottenere una visione unificata e olistica dei propri progressi, costringendolo a utilizzare molteplici strumenti.

TrackFit intende superare queste limitazioni centralizzando le informazioni relative a diverse tipologie di attività fisica in un'unica piattaforma. L'obiettivo primario è fornire un'interfaccia utente, disponibile sia su web che su mobile, che sia semplice, intuitiva e accessibile.

Attraverso questa interfaccia, il sistema offrirà report chiari e facilmente interpretabili, permettendo all'utente di analizzare le proprie performance in modo efficace e di mantenere la costanza nel proprio percorso sportivo.

Per raggiungere questi scopi, il sistema si concentrerà sulle seguenti macro-funzionalità:

- **Registrazione e Gestione delle Sessioni:** Permettere all'utente di inserire (manualmente o tramite importazione) i dettagli di ogni "Sessione di allenamento" (es. sport, data, durata, distanza, intensità).
- **Archiviazione e Storico:** Fornire un "Profilo utente" dove ogni sessione viene salvata per costruire uno storico completo e consultabile.
- **Analisi e Reportistica:** Offrire "Dashboard" e "Report" personalizzati per visualizzare i progressi, il raggiungimento degli "Obiettivi" e le statistiche aggregate.
- **Interoperabilità (Import/Export):** Garantire la possibilità di esportare i dati in formati aperti (es. CSV, PDF) e importare dati da formati standard (es. GPX, TCX) o API esterne.

3.2. Requisiti funzionali

3.2.1 Gestione Utenti, Allenatori e Gestori della Palestra

- **FR1:** Registrazione e autenticazione di utenti, allenatori e gestori.
- **FR2:** Modifica e aggiornamento dei profili personali.
- **FR3:** Gestione dei ruoli (utente, allenatore, gestore, amministratore).
- **FR4:** Possibilità di associare utenti a specifici allenatori o corsi.
-

Note Implementative Previste (Stack Next.js/TypeScript):

- Utilizzo di token (es. JWT) per la gestione sicura delle sessioni, gestiti lato client con React Context o uno store (es. Zustand).
- Sviluppo di API endpoints RESTful (utilizzando **Next.js API Routes** o un backend dedicato) per la comunicazione tra frontend e backend.
- Interfacce e form di registrazione/login costruiti come componenti **React** modulari e *type-safe* grazie a **TypeScript**.

- Database relazionale (MySQL) per la gestione delle tabelle utenti e ruoli.
- Funzioni CRUD per l'aggiornamento dei profili, con validazione dei form gestita lato client (es. React Hook Form) e lato server.

3.2.2 Gestione Palestre

- **FR5:** Creazione, modifica e cancellazione delle palestre (da parte dell'Amministratore).
- **FR6:** Configurazione dei corsi: nome, descrizione, orario, allenatore associato, capacità massima.
- **FR7:** Gestione degli abbonamenti: definizione della tipologia, durata, prezzo e regole di rinnovo.
- **FR8:** Prenotazione dei corsi da parte degli utenti (con verifica della capacità e abbonamento attivo).

Note Implementative Previste (Stack Next.js/TypeScript):

- Struttura dati relazionale (tabelle Palestre, Corsi, Abbonamenti, Prenotazioni).
- Pannello di controllo per Gestori e Amministratori implementato come interfaccia web in **Next.js** (utilizzando pagine statiche/SSR dove appropriato).
- Utilizzo di componenti **React** e **TypeScript** per creare form complessi per la configurazione di corsi e abbonamenti, gestendo lo stato locale (useState/useReducer) e globale.
- Implementazione di un sistema di notifiche (e-mail o push) per la conferma delle prenotazioni, invocato tramite API.

3.2.3 Monitoraggio delle Performance

- **FR9:** Registrazione dei progressi individuali degli utenti (es. log allenamenti, peso, parametri fisici).
- **FR10:** Visualizzazione dei progressi tramite dashboard personali e grafici storici.
- **FR11:** Invio di notifiche automatiche per obiettivi raggiunti o per la scadenza imminente dell'abbonamento.

Note Implementative Previste (Stack Next.js/TypeScript):

- App mobile sviluppata in **React Native** (per coerenza con lo stack React) o come Progressive Web App (PWA) **Next.js**, con sincronizzazione dei dati tramite API RESTful.
- Dashboard utente costruita in **React** con grafici interattivi, utilizzando una libreria *React-friendly* (es. Recharts, Chart.js) e **TypeScript** per la tipizzazione dei dati dei grafici.
- Fetching dei dati per la dashboard tramite *data fetching hooks* di **Next.js** (es. useSWR o React Query) per una sincronizzazione efficiente.

3.2.4 Reportistica e Analisi Aggregate

- **FR12:** Generazione di report su iscrizioni, frequenza ai corsi e utilizzo delle strutture (per i Gestori).
- **FR13:** Report specifici per gli Allenatori sui progressi e la frequenza degli utenti a loro associati.
- **FR14:** Report per i Gestori relativi alla gestione complessiva della palestra (es. andamento economico, tasso di occupazione corsi).
- **FR15:** Report per l'Amministratore centrale con dati aggregati di tutte le palestre connesse al sistema.
-

Note Implementative Previste (Stack Next.js/TypeScript):

- Generazione dinamica dei report tramite query SQL aggregate (sul backend).
- Frontend dei report costruito in **React** e **TypeScript**, con componenti per filtri temporali e di categoria (es. per data, per corso).
- Valutazione dell'uso di Server-Side Rendering (SSR) di **Next.js** per la generazione di report che richiedono dati *fresh*.
- Implementazione di un accesso differenziato ai dati (RBAC) gestito sia a livello di API che a livello di componenti **React** (rendering condizionale dei componenti della UI).

3.2.5 Accessibilità e Interfaccia Intuitiva

- **FR16:** L'interfaccia utente deve essere responsiva e adattarsi a diversi formati (desktop, tablet, mobile).
- **FR17:** L'applicazione mobile deve garantire l'accesso a tutte le principali funzionalità disponibili sulla versione web.
- **FR18:** Implementazione di funzionalità di ricerca e filtraggio (es. per corsi, utenti, allenatori).
- **FR19:** Fornitura di dashboard personalizzate e ottimizzate per ogni ruolo (Utente, Allenatore, Gestore, Amministratore).

Note Implementative Previste (Stack Next.js/TypeScript):

- Sviluppo di una *Component Library* UI in **React** e **TypeScript** (es. utilizzando Storybook) per garantire la coerenza grafica e riutilizzare i componenti.
- Utilizzo di un approccio *mobile-first* con design responsive (es. CSS Modules, Tailwind CSS, o Styled Components) all'interno dell'applicazione **Next.js**.
- Routing dinamico e *layout nesting* gestiti da **Next.js** per creare le dashboard personalizzate per i vari ruoli.

3.2.6 Gestione Pagamenti e Abbonamenti

- **FR20:** Acquisto e Pagamento Abbonamenti. Il sistema deve permettere all'utente di selezionare un tipo di abbonamento (definito in FR7) ed effettuare il pagamento.
- **FR21:** Gestione Rinnovo Automatico. L'utente deve poter abilitare o disabilitare il

rinnovo automatico del proprio abbonamento.

- **FR22:** Storico Pagamenti e Fatturazione. L'utente deve poter visualizzare il proprio storico pagamenti e scaricare le relative fatture o ricevute.

Note Implementative Previste (Stack Next.js/TypeScript):

- Integrazione con un gateway di pagamento (es. Stripe) che supporti la SCA (Strong Customer Authentication).
- Le API di Next.js gestiranno i webhook del gateway per confermare i pagamenti e aggiornare lo stato dell'abbonamento sul database (MySQL).
- Creazione di una pagina "Abbonamento" nel profilo utente (React) dove gestire lo stato del rinnovo.
- Generazione PDF per le fatture (es. con pdf-lib lato server).

3.2.7 Controllo Accessi e Gestione Prenotazioni

- **FR23:** Accesso alla Struttura (Tessera Digitale). L'app mobile deve generare una tessera digitale (es. QR Code) univoca per l'utente, che il sistema di accesso della palestra (Gestore) può scansionare per validare l'abbonamento e registrare l'ingresso.
- **FR24:** Cancellazione Prenotazione Corso. L'utente deve poter cancellare una prenotazione effettuata (FR8), entro un tempo limite configurabile dal Gestore.
- **FR25:** Gestione Lista d'Attesa. Se un corso (FR6) è pieno, il sistema deve permettere all'utente di iscriversi a una lista d'attesa e notificargli se si libera un posto.
- **FR26:** Notifica Cancellazione Corso (da Gestore). Se un Gestore o Allenatore cancella una sessione di un corso, tutti gli utenti prenotati (FR8) e in lista d'attesa (FR25) devono ricevere una notifica (push o email).

Note Implementative Previste (Stack Next.js/TypeScript):

- Generazione del QR Code nell'app React (Native) o PWA.
- Creazione di un endpoint API protetto (/api/validate-access) che il sistema della palestra può interrogare per validare il QR.
- Logica di business nelle API (Next.js) per gestire le regole di cancellazione e le liste d'attesa.
- Servizio di notifiche (es. Firebase Cloud Messaging o un servizio email) invocato dalle API.

3.2.8 Interoperabilità e Dati

- **FR27:** Importazione Allenamenti Esterni. Il sistema deve permettere all'utente di importare sessioni di allenamento da file standard (es. GPX, TCX) o da API esterne (es. Strava, Garmin) per arricchire il proprio storico (FR9).
- **FR28:** Esportazione Report. Gli utenti (Gestori, Allenatori, Admin) devono poter esportare i report (FR12-FR15) in formati standard (es. CSV, PDF).

Note Implementative Previste (Stack Next.js/TypeScript):

- Sviluppo di parser lato server (nelle API Next.js) per i formati GPX/TCX.

- Gestione OAuth 2.0 per l'integrazione con API di terze parti.
- Utilizzo di librerie per la generazione di CSV/PDF (es. papaparse per CSV, pdf-lib per PDF) invocati dai componenti React della dashboard.

3.2.9 Gestione Account e Sicurezza

- **FR29: Recupero Password.** L'utente deve poter avviare una procedura di "password dimenticata" (es. tramite email) per reimpostare la propria password in sicurezza.
- **FR30: Cancellazione Account (GDPR).** L'utente deve avere la possibilità di richiedere la cancellazione permanente del proprio account e dei dati associati.

Note Implementative Previste (Stack Next.js/TypeScript):

- Implementazione di un flusso di reset password con token univoci a scadenza (gestiti via API Next.js e inviati tramite un servizio email).
- Creazione di una procedura (automatizzata o semi-automatizzata) per l'eliminazione dei dati utente dal database MySQL, nel rispetto delle normative privacy.

3.3. Requisiti non funzionali

3.3.1 Performance (Prestazioni e Scalabilità)

- **NFR-P1 (Tempi di Risposta):** Il sistema deve rispondere alle richieste utente standard (login, visualizzazione dashboard, apertura pagina corsi) in meno di 2 secondi in condizioni di carico normale (95° percentile).
- **NFR-P2 (Generazione Report):** La generazione di report complessi (es. aggregati mensili per la palestra) deve essere completata in meno di 5 secondi.
- **NFR-P3 (Scalabilità del Carico):** Il sistema deve essere in grado di supportare 1000 utenti concorrenti mantenendo i tempi di risposta (NFR-P1) senza degradazione delle prestazioni.
- **NFR-P4 (Scalabilità Orizzontale):** L'architettura deve permettere l'aggiunta di nuove risorse (es. istanze server) per gestire un aumento del numero di palestre e utenti senza richiedere una riprogettazione del sistema.
- **NFR-P5 (Aggiornamenti):** Deve essere possibile rilasciare nuove versioni del software (deploy) senza interrompere il servizio (zero-downtime deploy).

Implementazione prevista per far fronte al requisito:

- Caching lato server (es. Redis) per query frequenti e dati statici.
- Ottimizzazione delle query SQL (utilizzo di indici, analisi EXPLAIN).
- Deploy su provider Cloud (es. AWS, Vercel) con bilanciamento del carico e auto-scaling (in caso di espansione).

3.3.2 Reliability (Affidabilità e Sicurezza)

- **NFR-R1 (Disponibilità):** Il sistema deve garantire un uptime del 99,5% (esclusa la manutenzione programmata).

- **NFR-R2 (Backup e Recovery):** Devono essere eseguiti backup automatici giornalieri del database. Il sistema deve avere un RPO (Recovery Point Objective) massimo di 24 ore e un RTO (Recovery Time Objective) massimo di 4 ore in caso di disastro.
- **NFR-R3 (Autenticazione):** Tutti gli accessi al sistema devono essere autenticati. Le password devono essere salvate utilizzando un algoritmo di hashing robusto (es. bcrypt).
- **NFR-R4 (Autorizzazione):** L'accesso alle funzionalità e ai dati deve essere strettamente controllato in base al ruolo dell'utente (RBAC: Utente, Allenatore, Gestore, Amministratore).
- **NFR-R5 (Protezione Dati):** Tutte le comunicazioni tra client e server devono avvenire tramite connessione sicura (HTTPS). I dati sensibili (informazioni personali, parametri fisici) devono essere protetti.

Implementazione prevista per far fronte al requisito:

- Connessioni HTTPS e crittografia SSL/TLS.
- Controllo dei permessi (middleware) lato backend (API) per ogni richiesta.
- Strategie di backup automatico e disaster recovery gestite dal provider del database (es. AWS RDS).

3.3.3 Usability (Usabilità)

- **NFR-U1 (Intuitività):** L'interfaccia utente deve essere chiara, intuitiva e coerente in tutte le sezioni dell'applicazione. Un nuovo utente deve essere in grado di completare le attività principali (registrazione, prenotazione corso) senza formazione.
- **NFR-U2 (Supporto Multilingua):** Il sistema deve supportare come minimo le lingue Italiano e Inglese per tutte le interfacce utente.
- **NFR-U3 (Compatibilità):** Il sistema web deve essere "responsive" e garantire piena funzionalità sulle ultime due versioni dei principali browser (Chrome, Firefox, Safari) e sui sistemi operativi mobili (iOS, Android).

Implementazione prevista per far fronte al requisito:

- Test di usabilità condotti con utenti campione.
- Gestione delle traduzioni tramite librerie dedicate (es. react-i18next).
- Design system coerente per pulsanti, icone e layout.

3.3.4 Supportability (Supportabilità e Manutenibilità)

- **NFR-S1 (Modularità):** Il codice sorgente deve essere modulare, seguire i principi di design stabiliti (es. SOLID) e aderire a guide di stile definite (es. per TypeScript e React) per facilitare la manutenzione.
- **NFR-S2 (Documentazione):** Le componenti complesse e le API devono essere documentate per facilitare l'inserimento di nuovi sviluppatori nel team.
- **NFR-S3 (Logging):** Il sistema deve produrre log strutturati per eventi critici ed errori, che devono essere centralizzati per facilitare il debugging e il monitoraggio.

Implementazione prevista per far fronte al requisito:

- Adozione di una struttura di progetto chiara in Next.js.
- Utilizzo di TypeScript per la tipizzazione e la documentazione implicita del codice.
- Versionamento del codice con Git (GitHub) e adozione di un flow di lavoro (es. GitFlow).

3.3.5 Implementation (Requisiti di Implementazione)

- **NFR-I1 (Stack Frontend):** Il frontend dell'applicazione web deve essere sviluppato utilizzando **React (Next.js)** e **TypeScript**.
- **NFR-I2 (Stack Mobile):** L'applicazione mobile deve essere sviluppata utilizzando **React Native** (per coerenza con lo stack) o come **PWA** (Progressive Web App) basata su Next.js.
- **NFR-I3 (Stack Backend):** Le API di backend devono essere sviluppate in **TypeScript** (utilizzando Next.js API Routes o un framework Node.js dedicato) e seguire un'architettura RESTful.
- **NFR-I4 (Database):** Il database principale deve essere un RDBMS (Relational Database Management System) **MySQL**.
- **NFR-I5 (Controllo Versione):** Il codice sorgente deve essere gestito su un repository **Git** (es. GitHub).

3.3.6 Interface (Requisiti di Interfaccia)

- **NFR-IF1 (API Gateway Pagamenti):** Il sistema deve interfacciarsi in modo sicuro con un gateway di pagamento esterno (es. Stripe) per elaborare gli acquisti degli abbonamenti (FR20).
- **NFR-IF2 (Importazione Dati):** Il sistema deve supportare l'importazione di file in formati standard (GPX, TCX) per la registrazione degli allenamenti esterni (FR27).
- **NFR-IF3 (Esportazione Dati):** Il sistema deve fornire funzionalità di esportazione dei report (FR28) in formati standard (CSV, PDF).

3.3.7 Packaging (Requisiti di Distribuzione)

- **NFR-PK1 (Accesso Web):** L'applicazione web deve essere accessibile pubblicamente tramite un nome di dominio sicuro (HTTPS).
- **NFR-PK2 (Distribuzione Mobile):** L'applicazione mobile (se nativa) deve essere distribuibile tramite i principali store ufficiali (Apple App Store e Google Play Store).
- **NFR-PK3 (Automazione Deploy):** Il processo di rilascio (build, test e deploy) deve essere automatizzato tramite un sistema di CI/CD (es. GitHub Actions).

3.3.8 Legal (Requisiti Legali e di Conformità)

- **NFR-L1 (GDPR):** Il sistema deve essere conforme al Regolamento Generale sulla Protezione dei Dati (GDPR) per la raccolta, l'archiviazione e il trattamento dei dati personali degli utenti, inclusa la gestione del consenso e il diritto alla cancellazione (FR30).

- **NFR-L2 (Conformità Pagamenti):** Il sistema deve aderire agli standard di sicurezza PCI-DSS. L'integrazione con il gateway di pagamento (NFR-IF1) deve essere implementata in modo da ridurre al minimo l'ambito di conformità (es. utilizzando componenti UI ospitati dal provider).
- **NFR-L3 (Termini e Condizioni):** Gli utenti devono poter visualizzare e accettare esplicitamente i Termini di Servizio e l'Informativa sulla Privacy al momento della registrazione.

3.3.9 Vincoli di dominio

- **R1:** Una sessione dell'atleta è modificabile solo se l'ora di inizio dista ≥ 48 ore dal momento della modifica; altrimenti l'operazione è bloccata.
- **R2:** Ogni modifica approvata invia automaticamente un'e-mail all'atleta con riepilogo, motivazione e riferimento al piano.
- **R3:** Cancellazione gratuita ≥ 24 ore; penale 50% oltre il limite.
- **R4:** Coupon valido solo per la struttura "FitCorp — Roma Centro" e piano "Annuale"; un solo utilizzo per utente; scadenza 31/10/2025.
- **R5:** Sconto applicato solo alla quota abbonamento (non cumulabile).
- **R6:** Pagamento con Strong Customer Authentication (3-D Secure).
- **R7:** Tessera digitale valida ai tornelli solo con SUB "Attivo".
- **R8:** Unicità struttura (PIVA e CUN come chiavi globali univoche).
- **R9:** Anti-duplicato fuzzy su Denominazione + Indirizzo con punteggio di similarità.
- **R10:** Tutte le operazioni amministrative (incluse coupon, pagamento, creazione SUB, fattura) registrano un audit (utente, timestamp, azione).

3.4. System models

3.4.1. Scenarios:

3.4.1.1 Scenario 1 Corsa serale:

- **Attore principale:** Tizio Caio (Utente amatoriale)
- **Data di riferimento:** 27/10/2025
- **Obiettivo:** Svolgere una corsa serale, registrarla con l'app mobile, verificarne i risultati sul web, aggiornare gli obiettivi e generare un report.

Descrizione Tizio Caio si autentica sull'app TrackFit da smartphone e accede alla Home, dove visualizza:

- Chilometri della settimana
- Ritmo medio
- Grafico ultimi 30 giorni
- Avanzamento obiettivo

Dalla sezione Allenamenti seleziona Pianifica → Nuova sessione, con schede Dettagli,

Obiettivo, Sensori, Note/Tag.

Imposta:

- Tipo: Corsa (outdoor)
- Data/Ora: 27/10/2025, ore 19:00
- Durata stimata: 45:00
- Intensità: Media
- Obiettivo: Durata 00:45:00
- Sensori: GPS ON, Cardio OFF
- Tag: serale, parco
- Nota: “Riscaldamento 10’, poi ritmo costante”

Salva la sessione: il sistema attiva un promemoria 10 minuti prima dell’inizio. Alle 18:50 riceve una notifica push: “Allenamento corsa alle 19:00 — Obiettivo 45 min.” Dalla notifica accede alla sessione pianificata e avvia il tracciamento. Il sistema acquisisce il fix GPS.

Nel pannello Live visualizza tempo, distanza, ritmo attuale/medio, barra di progresso; sono attivi auto-lap e annunci vocali a ogni km. Durante la sessione:

- Al minuto 20 mette in pausa → cronometro fermato e evento registrato.
- Dopo 60” riprende.
- Alle 19:46 termina l’attività.

Visualizza il riepilogo provvisorio:

- Durata: 00:45:20
- Distanza: 8,05 km
- Ritmo medio: 5’38”/km
- Calorie: ≈616 kcal
- Dislivello+: 32 m

Aggiunge:

- Percezione sforzo: 7/10
- Nota: “Gambe ok, leggera fatica finale”
- Tag: serale

Il sistema sincronizza via Wi-Fi con il backend e archivia la sessione in Storico/Calendario, con mappa GPS e grafici ritmo/distanza.

La Dashboard aggiorna lo stato dell’obiettivo: 20,1/20 km (raggiunto). Tizio pubblica la sessione sulla bacheca personale, conferma, e riceve un’e-mail di conferma.

Da web accede al portale TrackFit → Area Utente / Bacheca Allenamenti, visualizza e apre il dettaglio della corsa del 27/10/2025 (riepilogo, grafici, mappa, tag, note). Il sistema:

- Segnala la sessione come candidata al conteggio obiettivo settimanale
- Verifica assenza duplicati (import GPX/TCX)
- Dopo conferma, marca la sessione come Consolidata

Infine, genera un Report PDF per ottobre 2025 (distanza totale, tempo, ritmo medio, trend settimanale, PR). Scarica il documento (archiviato in “Report salvati”, disponibile anche in CSV). Imposta un nuovo obiettivo settimanale di 25 km e aggiorna la dashboard. Da Profilo, esporta i dati come pacchetto ZIP (CSV + GPX) ricevuto via e-mail.

3.4.1.2 Scenario 2 Gestione orari da parte dell’allenatore:

- **Attore principale:** Marco Bianchi (Coach)
- **Atleta associato:** Tizio Caio
- **Data/Ora di riferimento:** 24/10/2025 – 09:00

Descrizione Il Coach accede al portale web TrackFit → Home Coach (widget “Piani attivi”, “Atleti seguiti”, “Prossime sessioni”).

Pianifica un allenamento personale (25/10/2025 – 07:00-08:00, Parco Nord, nota “Lento rigenerante”). Poi apre: Roster → Tizio Caio → Calendario Atleta e trova la sessione “Corsa medio-lunga — 27/10/2025 19:00–20:00”. Avvia una modifica oraria: propone 27/10/2025 18:00–19:00.

Il sistema calcola il preavviso (81h) → R1 soddisfatta. Inserisce motivazione: “Anticipazione per temperatura più bassa e luce diurna”. Il sistema:

- Aggiorna il calendario atleta
- Invia e-mail automatica (R2)
- Registra log audit (timestamp, utente, variazioni, motivazione)
- Mostra banner “Orario aggiornato dal Coach per 27/10/2025” in Dashboard atleta

Successivamente (26/10/2025 – 14:15) il Coach tenta un ulteriore spostamento a 17:30 → preavviso ~27h15m → R1 violata. Il sistema blocca l’operazione e mostra messaggio di errore con opzioni “Annulla” o “Proponi nota all’atleta”.

3.4.1.3 Scenario 3 Autenticazione (portale business):

- **Attore principale:** Amministratore FitCorp (admin@fitcorp.it)

Descrizione L’amministratore accede a TrackFit Business → Home Azienda. Naviga in Area Azienda → Strutture → Nuova Struttura e compila: “FitCorp — Palestra Roma Centro”, P.IVA: 12345678901, CUN: FIT-ROMA-001, Indirizzo: Via Appia 10, 00179 Roma (RM). Il sistema esegue la verifica anti-duplicato (P.IVA, CUN, Denominazione+Indirizzo):

- Nessun duplicato → creazione STR-000912 → stato Attiva
- Registra audit e invia e-mail di conferma
Poi aggiunge un allenatore: Staff → Aggiungi Allenatore → m.bianchi.coach@fitcorp.it
- Se utente esistente → associazione a STR-000912
- Se nuovo → creazione profilo e invito e-mail

Il sistema registra l’audit (“grant: role=Coach, STR-000912 → USR-00421”) e invia e-

mail di onboarding con ruolo, permessi e link di accesso.

3.4.1.4 Scenario 4 Prenotazione allenatore:

- **Attore principale:** Tizio Caio (Utente)
- **Coach:** Marco Bianchi
- **Struttura:** FitCorp — Palestra Roma Centro
- **Data/Ora di riferimento:** 26/10/2025 – 10:00

Descrizione L'utente accede a TrackFit (web/mobile) → Dashboard → Coach / Trova Allenatore. Filtra per Città=Roma, Struttura=FitCorp — Roma Centro, Disciplina=Corsa. Seleziona Marco Bianchi, legge bio, certificazioni e tariffe: “Sessione individuale 60’ — €40”. Prenota lo slot 30/10/2025 – 19:00–20:00 → nessun conflitto. Pagamento con carta VISA (SCA 3-D Secure OK, ID PAY-2025-10-26-00041). Il sistema:

- Crea prenotazione PRN-000873, stato “Confermata”
- Invia e-mail a utente e coach con allegato .ics
- Blocca slot su entrambi i calendari
- Genera ricevuta RCPT-000873.pdf

In Area Utente → Prenotazioni appare PRN-000873 con pulsanti Dettagli, Annulla, Sposta. Promemoria automatici a T-48h e T-2h; check-in rapido via QR a T-15’.

3.4.1.5 Scenario 5 Attivazione abbonamento annuale con coupon:

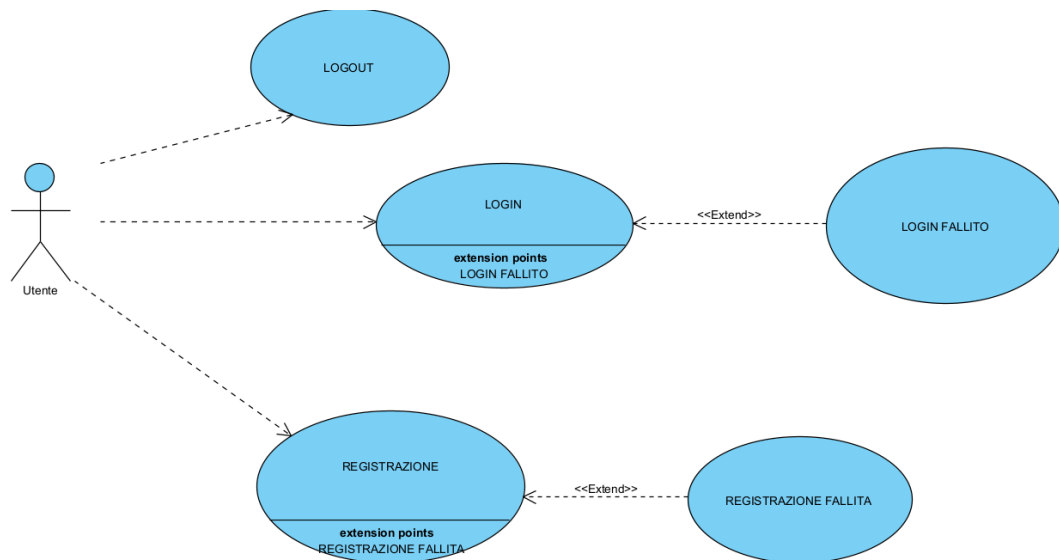
- **Attore principale:** Tizio Caio (Utente)
- **Struttura:** FitCorp — Palestra Roma Centro
- **Prodotto:** Abbonamento Annuale (12 mesi)
- **Data/Ora di riferimento:** 27/10/2025 – 18:00

Descrizione L'utente apre l'e-mail “Benvenuto in FitCorp — Roma Centro: -20% sull'abbonamento annuale” (codice ROMA-ANN20-2025). Segue il link “Attiva ora” → accede con t.caio19@trackfit.app. Il sistema verifica i vincoli (struttura, piano, validità, unicità).

Prezzo base: €480,00 → Sconto 20% (–€96,00) + Quota attivazione €20,00 → Totale €404,00. Effettua pagamento VISA (3-D Secure OK, ID PAY-2025-10-27-00421). Il sistema crea abbonamento SUB-001139, stato Attivo, validità 27/10/2025 → 26/10/2026. Coupon marcato Usato, fattura INV-2025-10-00421 (PDF) generata, tessera digitale con QR. E-mail di conferma a utente e struttura.

In Area Utente → I miei abbonamenti compare SUB-001139 con opzioni Mostra QR, Scarica fattura, Gestisci rinnovo. Alla prima visita, la reception valida il QR e registra l'ingresso.

3.4.2. Use case model:



UC 1 – Autenticazione

Attore: Utente, Coach, Amministratore

Entry Condition: L'utente è sulla schermata di autenticazione dell'app *TrackFit*

Flusso di eventi:

1. L'utente inserisce username *t.caio@example.com* e password
2. L'utente invia i dati al sistema
3. Il sistema verifica le credenziali con esito positivo
4. Il sistema mostra la *Home/Dashboard*

Exit Condition: Utente autenticato sulla Home

Alternativa: Se al punto 3 le credenziali non sono valide → UC 2

UC 2 – Autenticazione fallita

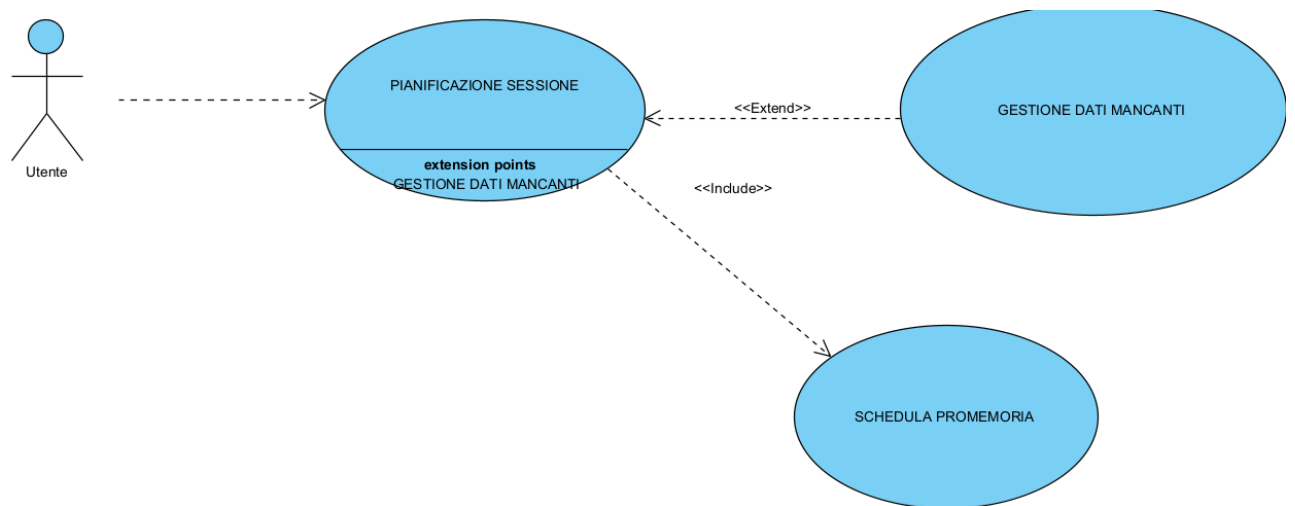
Attori: Utente, Coach, Amministratore

Entry Condition: Tentativo di autenticazione non riuscito

Flusso di eventi:

1. Il sistema mostra “username o password non corretti”
2. Il sistema ripresenta la schermata di autenticazione

Exit Condition: Attore sulla schermata di autenticazione



UC 3 – Pianificazione sessione

Attore: Utente

Entry Condition: Utente autenticato sulla Home

Flusso di eventi:

1. L'utente apre *Allenamenti* → *Pianifica*
2. Il sistema mostra “Nuova sessione” con schede: Dettagli, Obiettivo, Sensori, Note/Tag
3. L'utente compila:
 - Dettagli: Corsa (outdoor), data/ora 27/10/2025 19:00, durata stimata 45:00, intensità *Media*
 - Obiettivo: tipo *Durata*, valore 00:45:00
 - Sensori: GPS *ON*; HR *OFF*
 - Note/Tag: “serale, parco”; nota “Riscaldamento 10’, poi ritmo costante”
4. L'utente salva la pianificazione
5. Il sistema conferma e attiva promemoria 10 minuti prima

Exit Condition: Sessione pianificata con promemoria attivo

Alternativa: Se al punto 3 mancano dati obbligatori → UC 4

UC 4 – Dati pianificazione incompleti

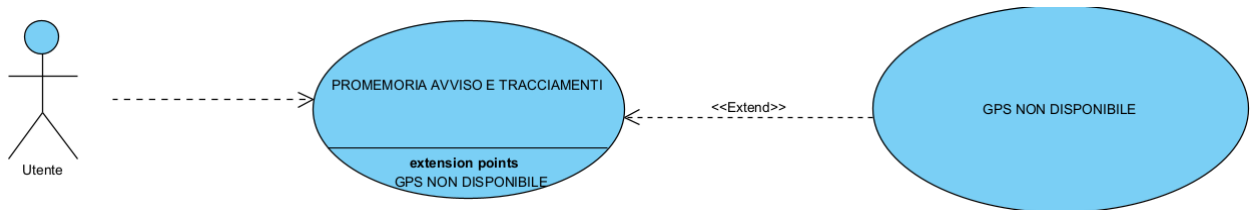
Attore: Utente

Entry Condition: Tentativo di salvataggio pianificazione con campi obbligatori mancanti

Flusso di eventi:

1. Il sistema evidenzia i campi mancanti e mostra messaggio di errore
2. L'utente completa i campi richiesti
3. L'utente salva nuovamente

Exit Condition: Pianificazione salvata



UC 5 – Promemoria avviso e tracciamenti

Attore: Utente

Entry Condition: Sessione pianificata con promemoria attivo

Flusso di eventi:

1. Alle 18:50 il sistema invia notifica “Allenamento corsa alle 19:00 — obiettivo 45 min.”
2. L'utente apre la notifica e accede a “Sessione pianificata”
3. L'utente avvia il tracciamento
4. Il sistema verifica stato GPS: ricerca → ok

Exit Condition: Sessione in stato “Live”

Alternativa: Se al punto 4 il GPS non è disponibile → UC 6

UC 6 – Gps non disponibile

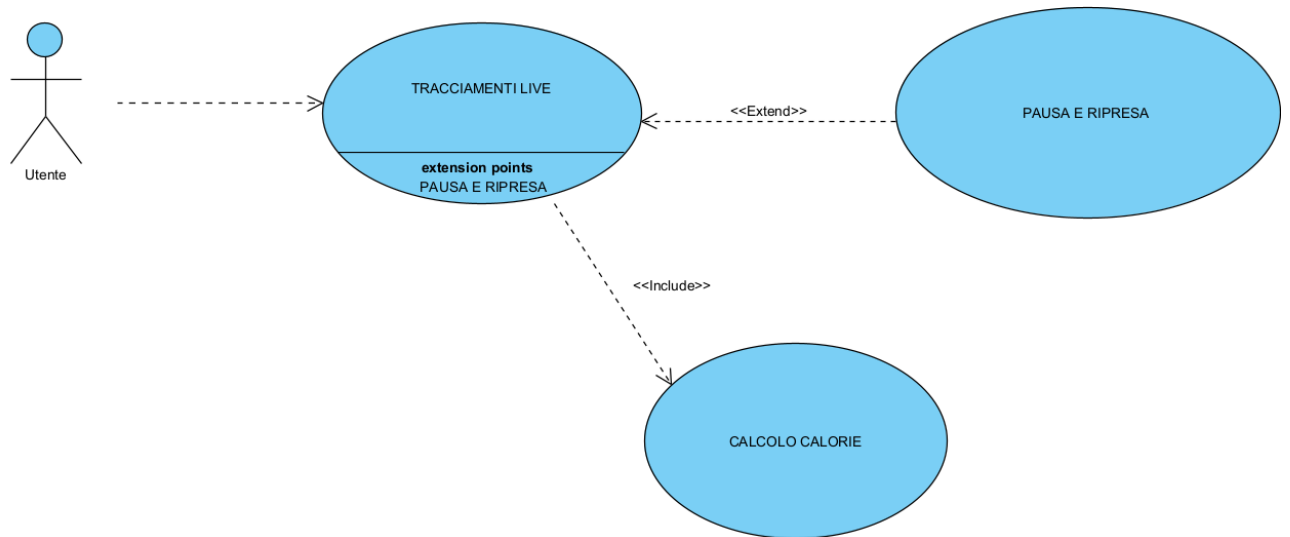
Attore: Utente

Entry Condition: Tentativo di avvio con fix GPS non acquisito

Flusso di eventi:

1. Il sistema segnala “GPS non disponibile”
2. L'utente abilita il GPS e/o attende il fix
3. Il sistema conferma fix acquisito

Exit Condition: Tracciamento avviabile



UC 7 – Tracciamenti live

Attore: Utente

Entry Condition: Sessione in stato “Live”

Flusso di eventi:

1. Il sistema mostra tempo, distanza, ritmo attuale/medio, barra progresso obiettivo
2. Il sistema emette annuncio vocale ogni 1 km con split e ritmo medio
3. Il sistema registra auto-lap e stima calorie

Exit Condition: Sessione in corso con dati registrati

Alternativa: Pausa e ripresa → UC 8

UC 8 – Pausa e ripresa

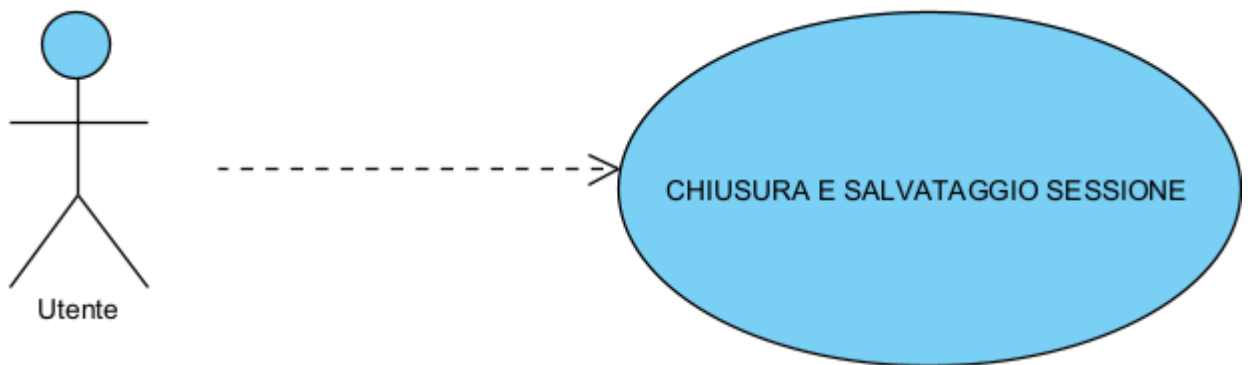
Attore: Utente

Entry Condition: Sessione in stato “Live”

Flusso di eventi:

1. Al minuto 20 l’utente mette in Pausa
2. Il sistema congela cronometro e registra l’evento
3. Dopo 60” l’utente seleziona *Riprendi*
4. Il sistema prosegue il tracciamento

Exit Condition: Sessione continua dopo pausa



UC 9 – Chiusura e salvataggio sessione

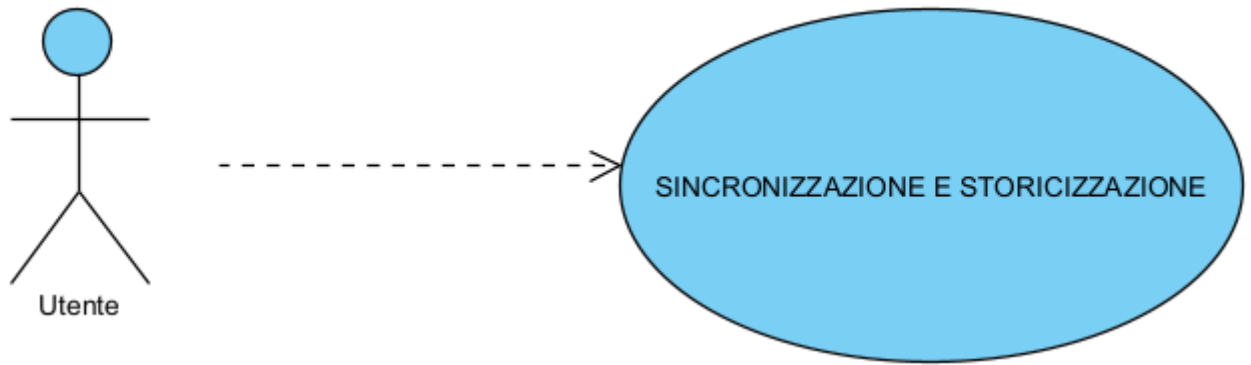
Attore: Utente

Entry Condition: Sessione in corso

Flusso di eventi:

1. Alle 19:46 l'utente seleziona *Termina*
2. Il sistema mostra riepilogo provvisorio:
 - Durata: 00:45:20
 - Distanza: 8,05 km
 - Ritmo medio: 5'38"/km
 - Calorie: ≈616 kcal
 - Dislivello+: 32 m
3. L'utente compila/modifica:
 - Percezione sforzo: 7/10
 - Note: "Gambe ok, leggera fatica finale"
 - Tag: "serale"
4. L'utente salva
5. Il sistema conferma il salvataggio

Exit Condition: Sessione salvata sul dispositivo



UC 10 – Sincronizzazione e storicizzazione

Attore: Sistema

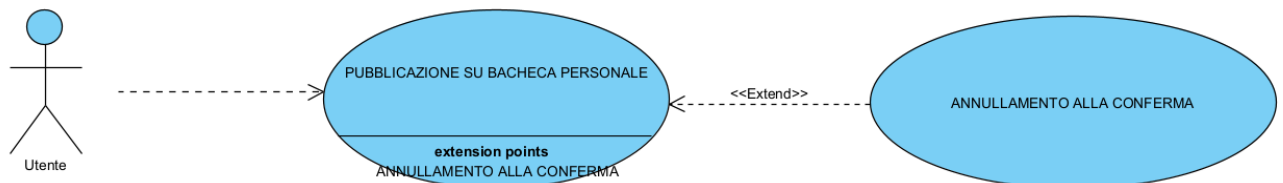
Entry Condition: Sessione salvata; connettività Wi-Fi disponibile

Flusso di eventi:

1. Il sistema sincronizza la sessione verso backend
2. Il sistema archivia in *Storico* → *Calendario*
3. Il sistema associa mappa GPS, grafici ritmo/distanza, split per km
4. Il sistema aggiorna la Dashboard: 20,1/20 km (obiettivo raggiunto)

Exit Condition: Sessione presente in Storico; Dashboard aggiornata

Alternativa: Assenza connettività → ritenta in background (stato “In attesa di sync”)



UC 11 – Pubblicazione su bacheca personale

Attore: Utente

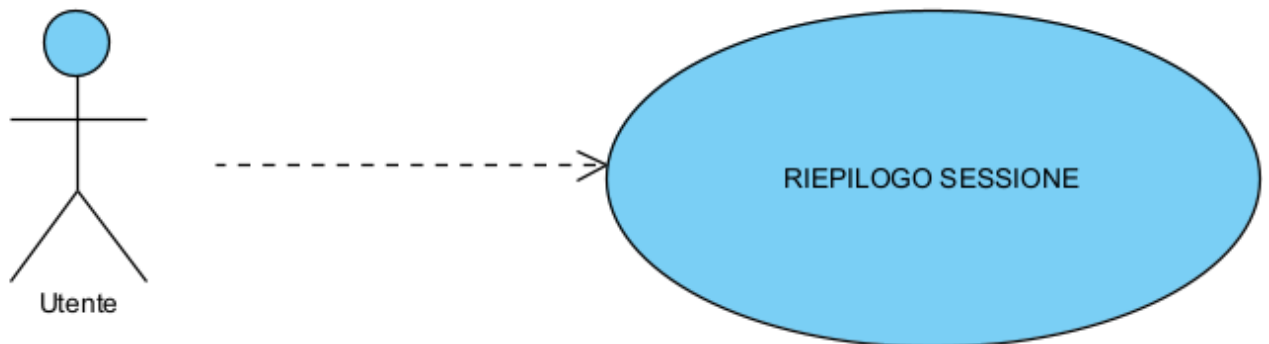
Entry Condition: Sessione presente in Storico

Flusso di eventi:

1. L'utente apre la sessione in Storico
2. L'utente seleziona “Pubblica su bacheca”
3. Il sistema richiede conferma
4. L'utente conferma
5. Il sistema imposta stato “Pubblicata” e invia email di conferma

Exit Condition: Sessione pubblicata

Alternativa: Annullamento alla conferma → stato invariato



UC 12 – Accesso via web e verifica

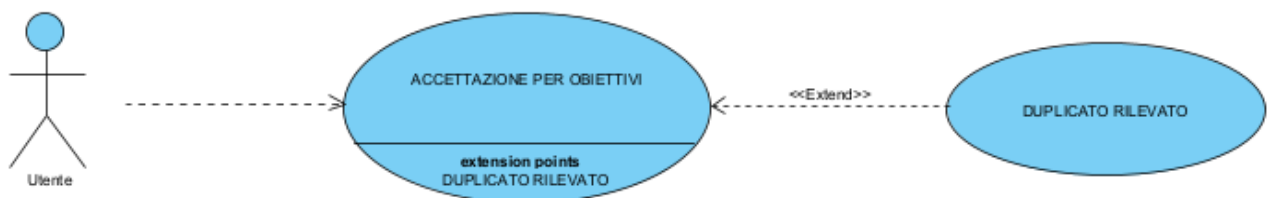
Attore: Utente

Entry Condition: Sessione pubblicata; utente con credenziali web

Flusso di eventi:

1. L'utente accede al portale *TrackFit* e si autentica
2. L'utente apre Area *Utente* / *Bacheca* Allenamenti
3. L'utente visualizza elenco sessioni Pubblicate
4. L'utente apre il dettaglio della corsa del 27/10/2025
5. Il sistema mostra riepilogo, grafici, mappa, tag, note

Exit Condition: Sessione visualizzata sul web



UC 13 – Accettazione per obiettivi e deduplica

Attore: Utente

Entry Condition: Sessione pubblicata; candidata al conteggio obiettivo settimanale

Flusso di eventi:

1. Il sistema verifica duplicati su file import (GPX/TCX)
2. Il sistema mostra esito “nessun duplicato trovato”

3. L'utente seleziona "Accetta per obiettivo"
4. Il sistema richiede conferma
5. L'utente conferma
6. Il sistema imposta stato "Consolidata"

Exit Condition: Sessione consolidata e conteggiata nell'obiettivo

Alternativa: Se al punto 1 viene rilevato un duplicato → UC 14

UC 14 – Duplicativo rilevato

Attore: Utente

Entry Condition: Esito verifica duplicati positivo

Flusso di eventi:

1. Il sistema indica la sessione duplicata e le possibili azioni
2. L'utente sceglie se unire, tenere solo una sessione o annullare
3. Il sistema applica l'azione scelta

Exit Condition: Duplicazione risolta; sessione pronta o esclusa dal conteggio



UC 15 – Generazione report

Attore: Utente

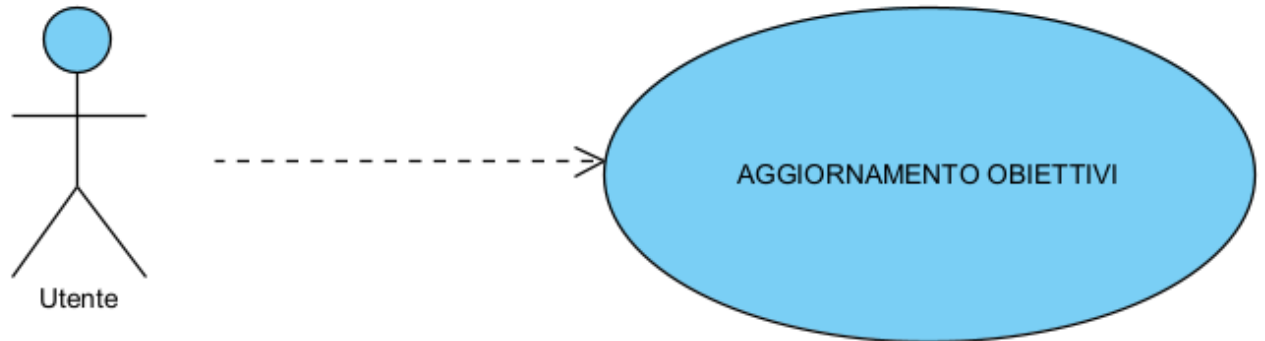
Entry Condition: Sessioni presenti in Storico

Flusso di eventi:

1. L'utente apre *Report* → *Genera PDF*
2. L'utente seleziona periodo "Ottobre 2025" e metriche:
 - Distanza totale
 - Tempo
 - Ritmo medio
 - Trend settimane
 - PR
3. L'utente avvia generazione

4. Il sistema crea il PDF
5. Il sistema rende disponibile il download e archivia in “Report salvati”
6. Il sistema offre anche “Esporta CSV”

Exit Condition: Report PDF scaricato; CSV disponibile



UC 16 – Aggiornamento obiettivi

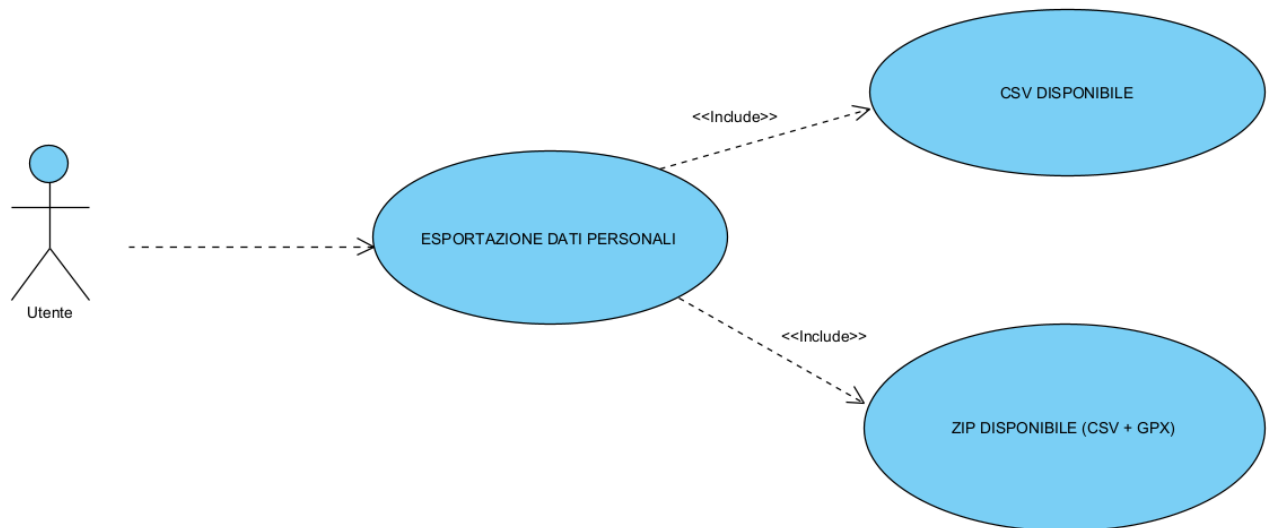
Attore: Utente

Entry Condition: Obiettivo settimanale corrente raggiunto (100%)

Flusso di eventi:

1. L'utente apre *Obiettivi*
2. L'utente imposta nuovo obiettivo 25 km per la prossima settimana
3. L'utente salva
4. Il sistema aggiorna la Dashboard con il nuovo target

Exit Condition: Nuovo obiettivo impostato



UC 17 – Esportazione dati personali

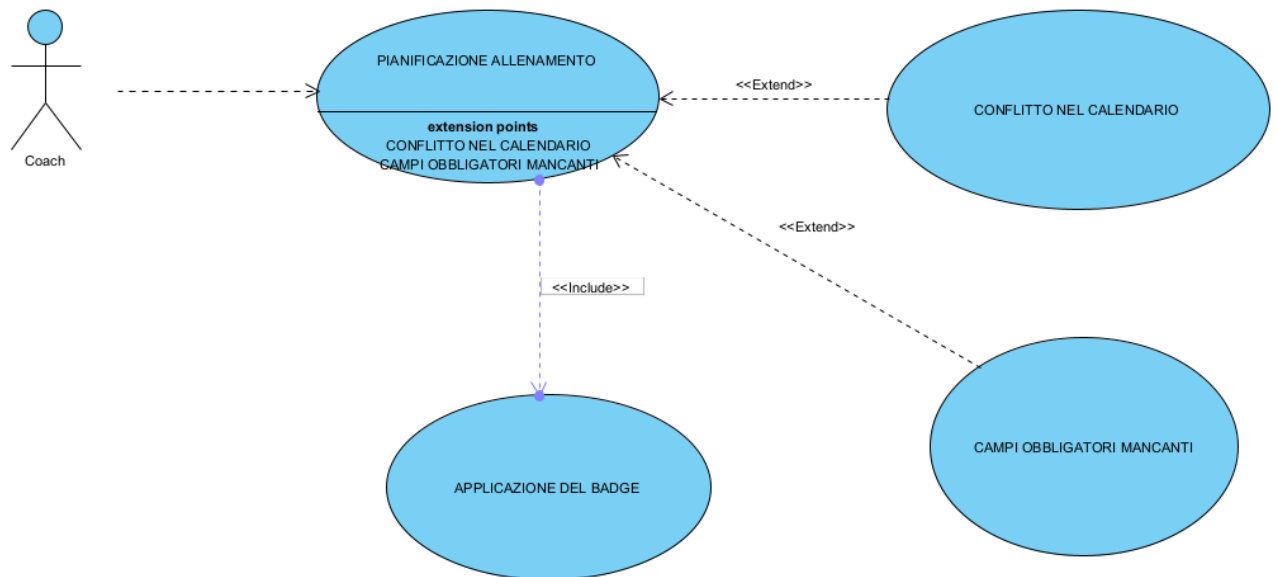
Attore: Utente

Entry Condition: Utente autenticato

Flusso di eventi:

1. L'utente apre *Profilo* → *Esporta dati*
2. L'utente seleziona *Pacchetto ZIP (CSV + GPX)*
3. L'utente conferma
4. Il sistema prepara l'archivio
5. Il sistema invia link di download via email

Exit Condition: Link di download ricevuto via email



UC 18 – Pianificazione allenamento personale del Coach

Attore: Coach

Entry Condition: Coach autenticato sulla Home Coach

Flusso di eventi:

1. Apre Calendario *Coach* → *Nuovo allenamento personale*
2. Compila:
 - Sport: Corsa
 - Data/ora: 25/10/2025 07:00-08:00
 - Luogo: Parco Nord
 - Note: “Lento rigenerante”
3. Salva
4. Il sistema registra l’evento nel Calendario Coach
5. Il sistema verifica conflitti con impegni del Coach
6. Il sistema applica badge “Personale” all’evento

Exit condition: Evento personale creato e visibile nel Calendario Coach

Alternative:

- Conflitto rilevato → **UC 21**
- Campi obbligatori mancanti → **UC 22**

UC 19 – Conflitto nel Calendario

Attore: Coach, Utente

Entry Condition: Salvataggio evento con sovrapposizione ad altro impegno

Flusso di eventi:

1. Il sistema evidenzia il conflitto (slot occupato)

2. Offre opzioni: modifica orario | salva come provvisorio | annulla
3. Il Coach seleziona un'opzione
4. Il sistema applica la scelta

Exit condition: Conflitto risolto o operazione annullata

UC 20 – Dati incompleti per allenamento personale (Coach)

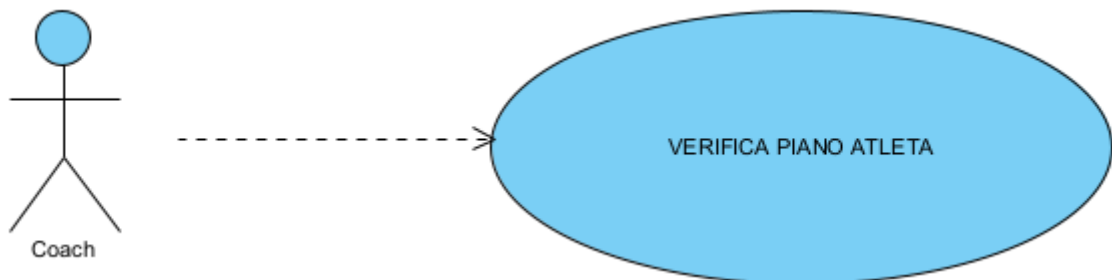
Attore: Coach

Entry Condition: Tentativo di salvataggio con campi obbligatori vuoti

Flusso di eventi:

1. Il sistema segnala i campi mancanti
2. Il Coach completa i campi
3. Salva nuovamente

Exit condition: Evento personale salvato



UC 21 – Verifica piano atleta

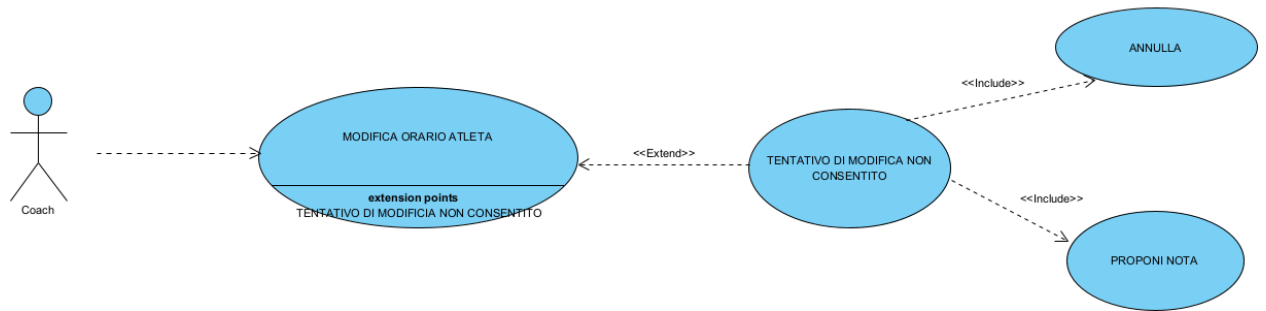
Attore: Coach

Entry Condition: Coach autenticato; atleta presente nel roster

Flusso di eventi:

1. Apre Area *Coach* → *Roster* → selezione *Tizio Caio*
2. Apre Calendario *Atleta*
3. Individua la seduta “Corsa medio-lunga – 27/10/2025 19:00-20:00”

Exit condition: Seduta dell’atleta identificata nel Calendario Atleta



UC 22 – Modifica orario atleta con preavviso valido ($\geq 48h$)

Attore: Coach

Entry Condition: Seduta atleta selezionata; data/ora corrente 24/10/2025 09:00

Flusso di eventi:

1. Selezione “Modifica orario” sulla seduta del 27/10/2025 19:00 – 20:00
2. Propone nuovo slot 27/10/2025 18:00 – 19:00
3. Il sistema calcola il preavviso 24/10/2025 09:00 \rightarrow 27/10/2025 = 81h
4. Il sistema valida R1 \rightarrow esito OK
5. Il Coach inserisce **Motivazione**: “Anticipazione per temperatura più bassa e luce diurna”
6. Conferma la modifica
7. Il sistema aggiorna il Calendario Atleta allo slot 18:00 – 19:00
8. Il sistema invia e-mail automatica all’atleta R2 con:
 - Oggetto: “TrackFit – Modifica orario allenamento 27/10/2025 18:00 – 19:00”
 - Corpo: “Ciao Tizio, il tuo allenamento Corsa medio-lunga del 27/10/2025 è stato modificato da 19:00 – 20:00 a 18:00 – 19:00. Motivazione: Anticipazione per temperatura più bassa e luce diurna. Calendario aggiornato in TrackFit – Coach: “Marco Bianchi”
 - Allegato opzionale: invito .ics con il nuovo slot
9. Il sistema registra log di audit: timestamp, utente, valori prima/dopo, motivazione

Exit condition: Seduta aggiornata; e-mail inviata; audit registrato

Alternativa: Se il vincolo R1 è violato \rightarrow UC26

UC 23 – Notifiche e controllo post-modifica lato Atleta

Attore: Sistema

Entry Condition: Modifica approvata e applicata (UC 24 completata)

Flusso di eventi:

1. Il sistema mostra banner su *Dashboard* Atleta: “Orario aggiornato dal Coach per 27/10/2025”

2. Il sistema aggiunge in *Attività recenti*: "Modifica orario da 19:00 a 18:00 – 27/10/2025"

Exit Condition: Notifiche presenti sull'account dell'atleta

UC 24 – Tentativo di modifica non consentita (< 48h)

Attore: Coach

Entry Condition: Data/ora corrente 26/10/2025 14:15; seduta atleta 27/10/2025 18:00–19:00

Flusso di eventi:

1. Il Coach tenta di spostare la seduta da 18:00 a 17:30
2. Il sistema calcola il preavviso: 27h 15m
3. Il sistema valuta R1 →violata
4. Il sistema blocca l'operazione
5. Il sistema mostra il messaggio: "Modifica orario negata: preavviso minimo 48 ore non rispettato"
6. Il sistema offre azioni: *Annulla* | *Proponi* nota all'atleta nel piano

Exit condition: Nessuna variazione a calendario; nessuna e-mail inviata

Alternativa: Selezione "Proponi nota all'atleta" → **UC 27**

UC 25 – Proposta nota al piano atleta

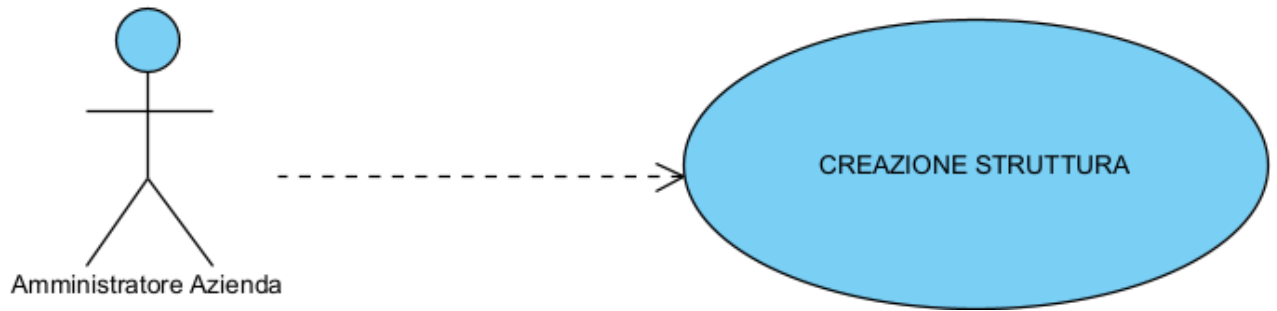
Attore: Coach

Entry Condition: Modifica bloccata per R1; Coach sceglie "Proponi nota all'atleta"

Flusso di eventi:

1. Inserisce testo nota (es. "Valuta anticipo a 17:30 se possibile; conferma via messaggi").
2. Invia la nota
3. Il sistema registra la nota nel piano atleta
4. Il sistema notifica l'atleta (notifica in-app: nessuna e-mail di modifica oraria)

Exit condition: Nota registrata; pianificazione inviata.



UC 26 – Avvio creazione Struttura

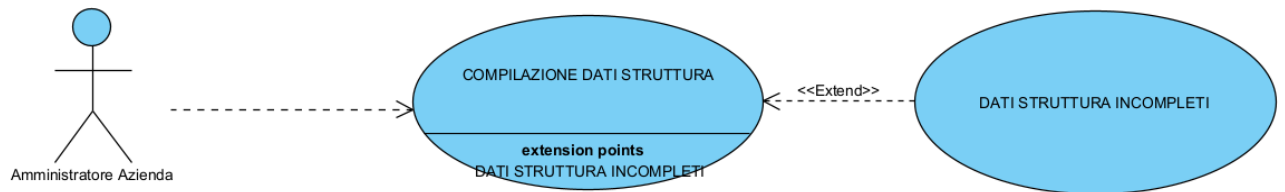
Attore: Amministratore Azienda

Entry Condition: Utente autenticato sulla Home Azienda

Flusso di eventi:

1. Apre Area Azienda → Strutture → Nuova Struttura
2. Il sistema mostra il form “Crea Struttura” con sezioni Dati anagrafici, Contatti, Localizzazione, Classificazione

Exit condition: Form di creazione Struttura aperto.



UC 27 – Compilazione dati Struttura

Attore: Amministratore Azienda

Entry Condition: Form “Crea Struttura” aperto

Flusso di eventi:

1. Compila:
 - Denominazione: “FitCorp – Palestra Roma Centro”
 - Partita IVA: 12345678901
 - Codice Struttura (CUN): FIT-ROMA-001
 - Categoria: Palestra
 - Indirizzo: Via Appia 10, 00179 Roma (RM)
 - Telefono: 06 1234567
 - E-mail: roma.centro@fitcorp.it
 - Servizi: sala pesi; percorsi cardio; spogliatoi M/F
2. Seleziona “Prosegui”
3. Il sistema valida i campi obbligatori

Exit condition: Dati pronti per verifica anti-duplicato

Alternativa: Se i dati sono incompleti → UC 32

UC 28 – Dati incompleti Struttura

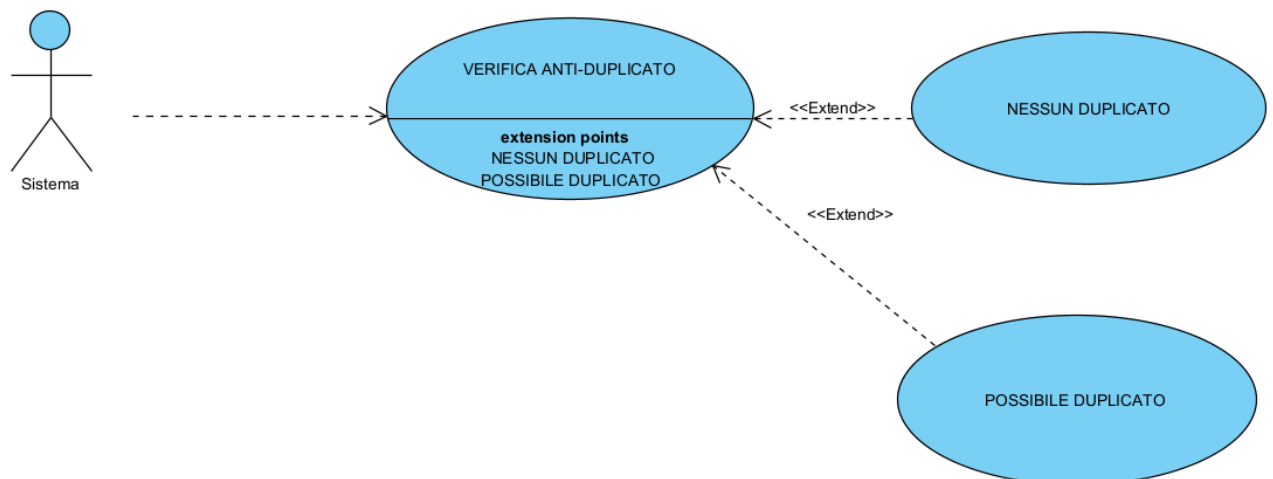
Attore: Amministratore Azienda

Entry Condition: Tentativo di proseguire con campi mancanti

Flusso di eventi:

1. Il sistema evidenzia i campi mancanti e mostra messaggio di errore
2. L'amministratore completa i campi
3. Riprova "Prosegui"

Exit condition: Dati validati



UC 29 – Verifica anti-duplicato Struttura

Attore: Sistema

Entry Condition: Dati Struttura validati

Flusso di eventi:

1. Esegue controllo su chiavi: P.IVA, CUN, Denominazione+Indirizzo
2. Se nessun duplicato certo o probabile → Esito A
3. Se rileva possibili *duplicati* → Esito B, con lista risultati
4. Applica vincoli R1-R2

Exit condition: Uno tra UC 34 o UC 35

UC 30 – Nessun duplicato

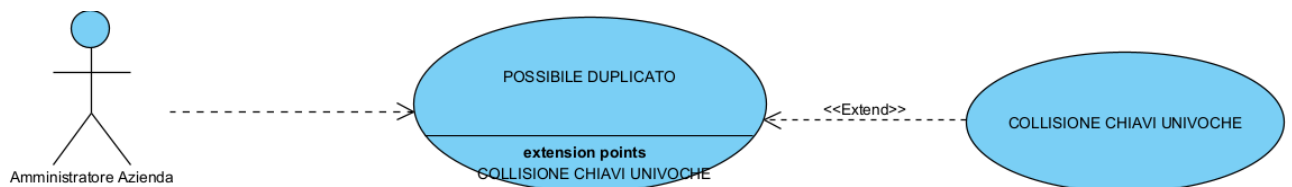
Attore: Amministratore Azienda

Entry Condition: Anti-duplicato senza corrispondenze

Flusso di eventi:

1. Il sistema mostra banner verde “Struttura non presente, puoi procedere con la creazione”
2. L’amministratore avvia *Riepilogo* → *Conferma*.

Exit condition: Passaggio alla creazione (UC 37).



UC 31 – Possibile duplicato

Attore: Amministratore Azienda

Entry Condition: Lista possibili duplicati mostrata

Flusso di eventi:

1. L’amministratore sceglie un’azione:
 - B1: *Apri scheda struttura trovata* → se Proprietario = nessuno → “Associa alla mia azienda” → *Conferma*.
 - B2: *Richiedi subentro* → invia richiesta al proprietario corrente
 - B3: *Ignora e continua* → consentito solo se nessuna chiave univoca coincide
2. Il sistema applica l’azione scelta e registra audit

Exit condition:

- B1 → Struttura associata; si prosegue a gestione Staff (UC 38)
- B2 → Stato “In attesa di subentro”; operazioni bloccate fino ad approvazione
- B3 → Passaggio alla creazione (UC 37)

Alternativa: Collisione chiavi univoche → UC 36

UC 32 – Blocco per collisione chiavi univoche

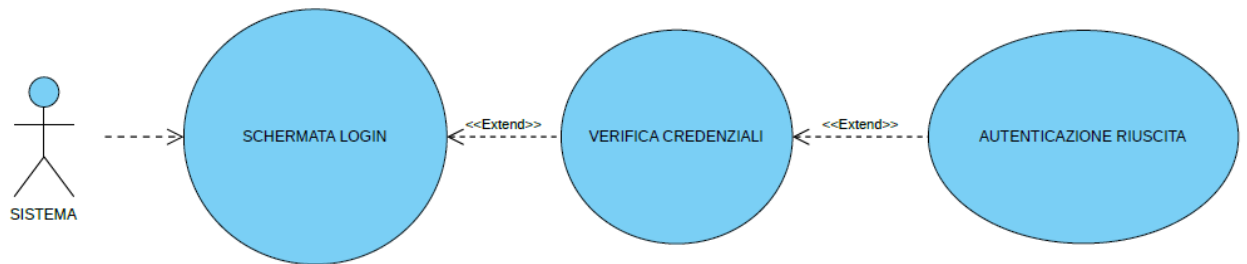
Attore: Sistema

Entry Condition: P.IVA o CUN coincidono con un record esistente

Flusso di eventi:

1. Mostra errore: “Creazione bloccata: P.IVA/CUN già presenti”
2. Offre solo azioni consentite: *Apri scheda* → *Associa* | *Richiedi subentro*

Exit condition: Creazione impedita; percorso forzato su associazione o subentro.



UC 33 – Creazione Struttura

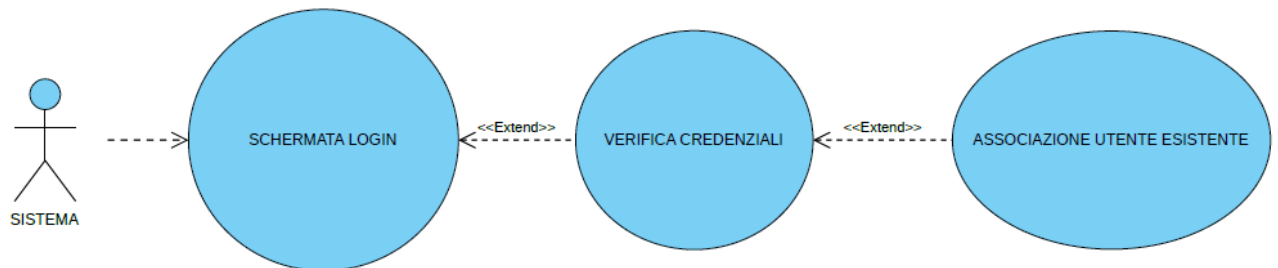
Attore: Amministratore Azienda

Entry Condition: Anti-duplicato superato (Esito A o B3)

Flusso di eventi:

1. Visualizza *Riepilogo*.
2. Conferma creazione
3. Il sistema crea la Struttura, assegna ID STR-000912
4. Il sistema registra audit “create: STR-000912, by admin@fitcorp.it,timestamp”
5. Il sistema invia e-mail di conferma a roma.centro@fitcorp.it
6. Il sistema aggiunge voce in *Attività recenti*
7. Il sistema imposta Stato struttura = Attiva

Exit condition: STR-000912 creata e attivata



UC 34 – Associazione Struttura esistente alla mia azienda

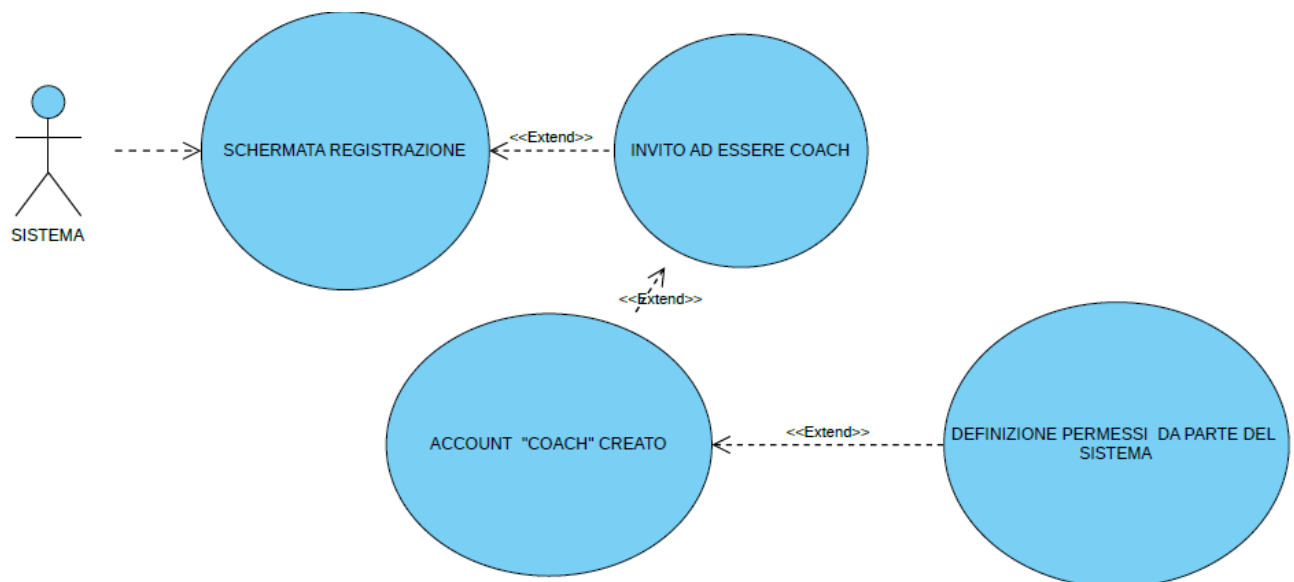
Attore: Amministratore Azienda

Entry Condition: Scelta B1 “Associa alla mia azienda”, proprietario assente

Flusso di eventi:

1. Conferma associazione
2. Il sistema associa la struttura all’azienda
3. Il sistema registra audit “associate: structure →FitCorp, by admin@fitcorp.it, timestamp”
4. Il sistema notifica in *Attività recenti*

Exit condition: Struttura associata a FitCorp, disponibile la gestione Staff.



UC 35 – Aggiunta allenatore alla struttura

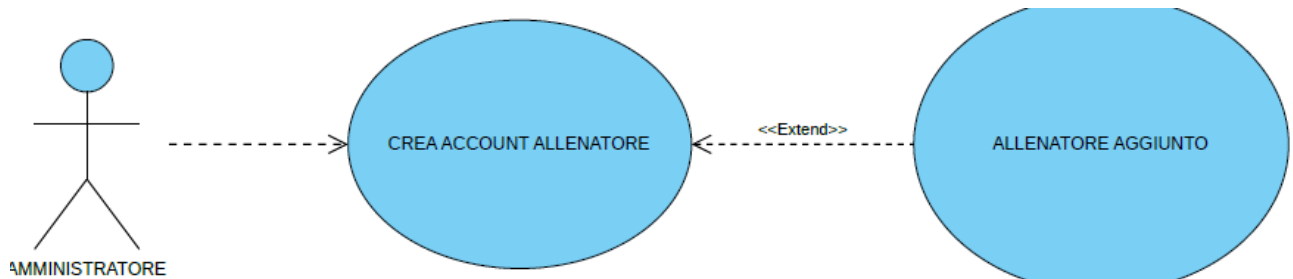
Attore: Amministratore Azienda

Entry Condition: Struttura attiva o associata all’azienda

Flusso di eventi:

1. Apre *Strutture* → “FitCorp – Palestra Roma Centro” → *Staff* → *Aggiungi allenatore*
2. Il sistema mostra “Nuovo allenatore” con modalità:
 - Invita tramite e-mail (crea utente Coach se non esiste)
 - Associa utente esistente (ricerca per e-mail/CF)

Exit condition: Form pronto per verifica allenatore



UC 36 – Verifica esistenza allenatore

Attore: Amministratore Azienda

Entry Condition: Form “Nuovo Allenatore” aperto

Flusso di eventi:

1. Inserisce e-mail m.bianchi.coach@fitcorp.it
2. Seleziona “Verifica”
3. Il sistema risponde:
 - Esito C – Utente esistente: Mostra nome, cognome, ID USR-00421, certificazioni, strutture associate
 - Esito D – Utente non trovato: richiede dati minimi per creazione

Exit condition: Percorso C o D

UC 37 – Associazione utente esistente

Attore: Amministratore Azienda

Entry Condition: Esito C — Utente esistente

Flusso di eventi:

1. Seleziona “Associa a STR-000912”
2. Conferma

Exit condition: Passaggio a definizione permessi (UC 43).

UC 38 – Creazione account coach e invito

Attore: Amministratore Azienda

Entry Condition: Esito D — Utente non trovato

Flusso di eventi:

1. Inserisce:
 - Nome: Marco
 - Cognome: Bianchi
 - E-mail: m.bianchi.coach@fitcorp.it
 - Discipline: forza; cardio
 - Certificazioni: “FIPE B-Level”
 - Ruolo: Coach
2. Seleziona “Crea e invia invito”

Exit condition: Account coach creato; passaggio a definizione permessi (UC 43)

UC 39 – Conferma permessi e associazione alla struttura

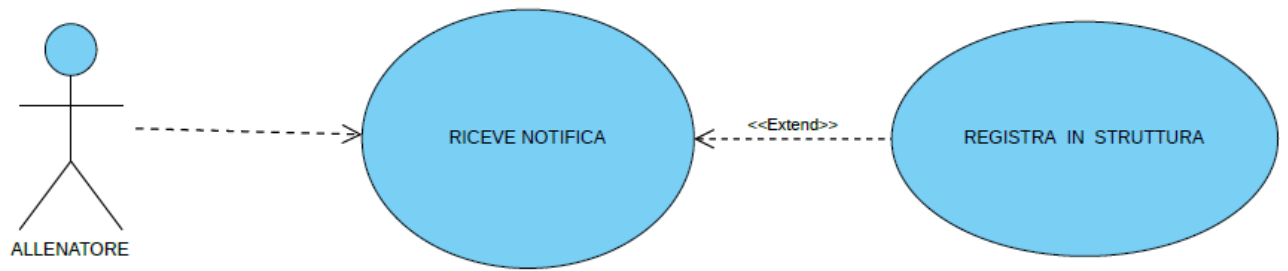
Attore: Amministratore Azienda

Entry Condition: Allenatore selezionato o creato

Flusso di eventi:

1. Rivede il riepilogo abilitazioni per STR-000912:
 - Visibilità calendario: solo struttura Roma Centro
 - Modifica orari atleti: consentita (R4)
2. Seleziona “Conferma” → “Salva”
3. Il sistema associa l’allenatore alla struttura
4. Il sistema invia l’email automaticamente all’allenatore:
 - Se nuovo: link “Attiva account”
 - Se esistente: notifica di nuova associazione
5. Il sistema registra audit “grant: role=Coach, STR-000912 → USR-00421”

Exit condition: Allenatore associato con permessi attivi



UC 40 – Notifiche e onboarding allenatore

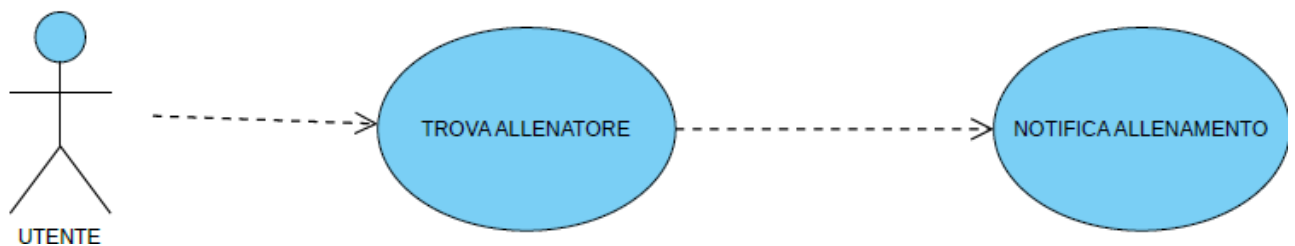
Attore: Sistema

Entry Condition: Associazione completata (UC 43)

Flusso di eventi:

1. Invia e-mail a m.bianchi.coach@fitcorp.it
 - Oggetto: “TrackFit – Assegnazione alla struttura FitCorp – Palestra Roma Centro”
 - Contenuto: Riepilogo ruolo, permessi, link *Attiva account/Accedi*
 - Allegato: invito .ics per briefing introduttivo
2. Registra voce in Attività recenti della struttura

Exit condition: Coach notificato, onboarding predisposto.



UC 41 – Ricerca allenatore

Attore: Utente

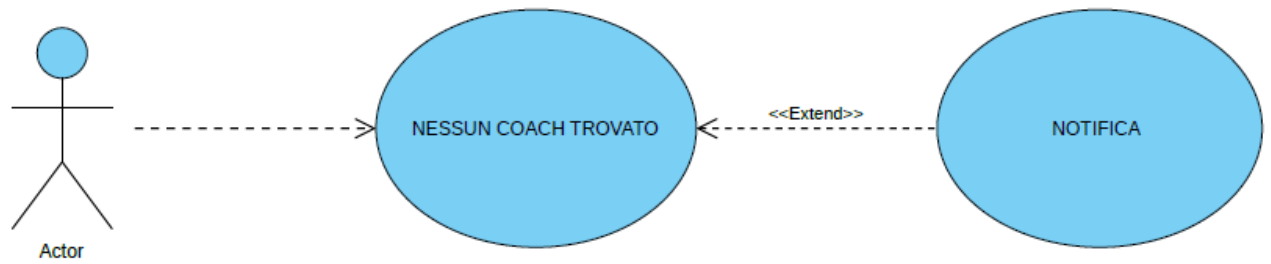
Entry Condition: Utente autenticato

Flusso di eventi:

1. Apre Area utente : Coach / Trova allenatore
2. Imposta filtri: Città=Roma; Struttura=FitCorp; Disciplina = Corsa.
3. Il sistema mostra risultati tra cui “Marco Bianchi – Coach – Disponibile lun/mer/sab – Rating 4.8/5”
4. L’utente apre il profilo del coach

Exit condition: Profilo coach aperto

Alternativa: Nessun risultato → UC 46



UC 42 – Nessun coach trovato

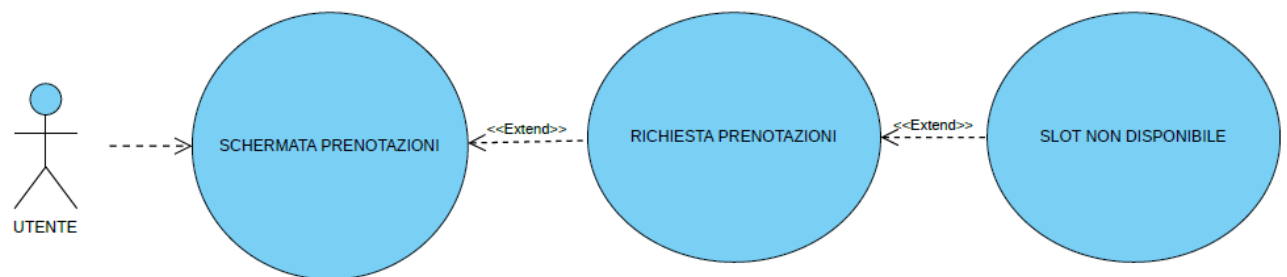
Attore: Utente

Entry Condition: Ricerca senza risultati

Flusso di eventi:

1. Il sistema mostra “Nessun coach corrispondende ai filtri”
2. Azioni disponibili: modifica filtri | annulla.

Exit condition: Utente decide se ripetere la ricerca o uscire



UC 43 – Visualizzazione profilo coach e slot disponibili

Attore: Utente

Entry Condition: Profilo del coach aperto

Flusso di eventi:

1. Il sistema mostra bio, certificazioni, struttura associata, tariffa: “Sessione individuale 60’ – 40 Euro”, politiche di modifica/cancellazione.
2. Il sistema mostra calendario integrato con slot disponibili
3. L’utente seleziona lo slot 30/10/2025 19:00 – 20:00 e preme Prenota

Exit condition: Avvio prenotazione sullo slot selezionato

Alternativa: Slot non più disponibile → UC 48

UC 44 – Slot non più disponibile

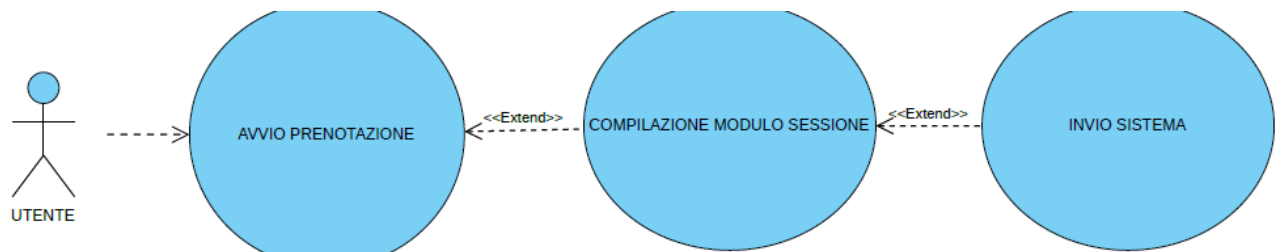
Attore: Utente

Entry Condition: Selezione slot invalidata da prenotazione concorrente

Flusso di eventi:

1. Il sistema mostra messaggio “Slot non disponibile”
2. Il sistema riporta al calendario del coach

Exit condition: Prenotazione non avviata; calendario aggiornato.



UC 45 – Compilazione modulo Prenota sessione

Attore: Utente

Entry Condition: Avvio prenotazione

Flusso di eventi:

1. Seleziona/compila:
 - Servizio: Personal Training 1:1
 - Struttura: FitCorp – Palestra Roma Centro
 - Data/ora: 30/10/2025 19:00 – 20:00
 - Note per il coach: “Focus ritmo costante”
2. Continua

Exit condition: Dati pronti per verifica

Alternativa: Campi obbligatori mancanti → UC 50

UC 46 – Dati incompleti nel modulo (Prenotazione)

Attore: Utente

Entry Condition: Tentativo di continuare con campi mancanti

Flusso di eventi:

1. Il sistema evidenzia i campi obbligatori
2. L'utente completa
3. Continua

Exit condition: Modulo valido

UC 47 – Verifica di sistema pre-riepilogo (Prenotazione)

Attore: Sistema

Entry Condition: Modulo valido

Flusso di eventi:

1. Verifica conflitti sul Calendario utente →nessuna sovrapposizione
2. Verifica disponibilità de coach sullo slot →libero
3. Applica regole R1: soddisfatta
4. Calcola preventivo: 40 Euro IVA compresa

Exit condition: Esito verifiche OK; si procede al riepilogo

Alternative:

- Conflitto utente rilevato →**UC 52**
- Slot occupato durante la verifica →**UC 53**

UC 48 – Disponibilità coach cambiata (Prenotazione)

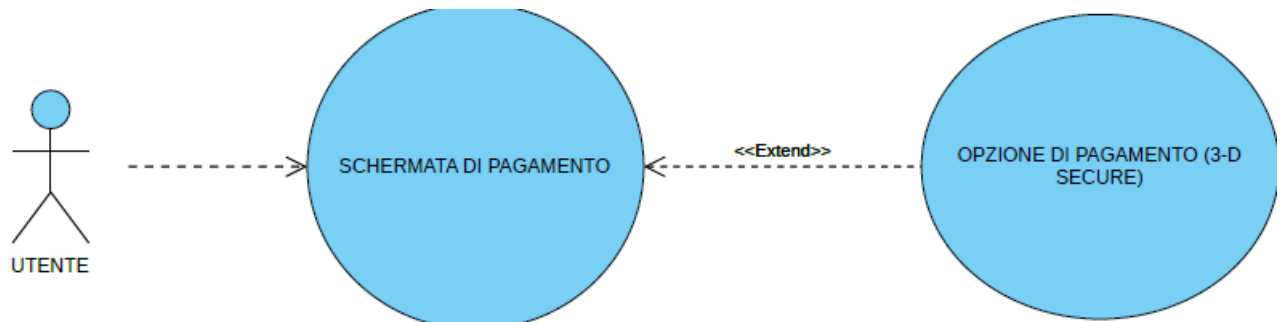
Attore: Sistema

Entry Condition: Lo slot diventa occupato

Flusso di eventi:

1. Il sistema segnala l'indisponibilità
2. Ritorno al calendario con slot aggiornati

Exit condition: Scelta di un nuovo slot o annullamento



UC 49 – Riepilogo e pagamento con SCA (Prenotazione)

Attore: Utente

Entry Condition: Verifiche OK

Flusso di eventi:

1. Il sistema mostra il riepilogo: servizio, struttura, data/ora, tariffa, politiche, note
2. L'utente seleziona metodo: Carta VISA **** 4242 (salvata)
3. “Paga e conferma” → apertura 3-D Secure dal provider
4. Immissione OTP 998877 + PIN 123456 → Conferma
5. Esito pagamento OK, il sistema genera ID transazione PAY-2025-10-26-00041

Exit condition: Pagamento riuscito

Alternativa: Pagamento rifiutato/SCA fallita → UC 55

UC 50 – Pagamento non riuscito (Prenotazione)

Attore: Utente

Entry Condition: Errore gateway, SCA fallita o fondi insufficienti

Flusso di eventi:

1. Il sistema mostra il motivo del rifiuto
2. Il sistema riporta al riepilogo

Exit condition: Prenotazione non creata; utente può riprovare con altro metodo

UC 51 – Creazione prenotazione e notifiche

Attore: Sistema

Entry Condition: Pagamento riuscito

Flusso di eventi:

1. Crea PRN-000873 con stato “Confermata”
2. Invia e-mail all’utente:
 - Oggetto: “Conferma prenotazione – Marco Bianchi – 30/10/2025 19:00”
 - Corpo: riepilogo, politiche, link Annulla/Sposta
 - Allegato: invito .ics
3. Invia e-mail al coach:
 - Oggetto: “Nuova sessione prenotata – Tizio Caio – 30/10/2025 19:00”
 - Corpo: note dell’utente, link Visualizza / Proponi modifica
4. Blocca lo slot su Calendario coach e Calendario Utente
5. Genera ricevuta RCPT-000873.pdf

Exit condition: Prenotazione confermata, notifiche inviate, calendari aggiornati

Alternativa: Policy di conferma manuale del coach → stato in “Attesa di conferma”

UC 52 – Schermata post-azione e gestione prenotazione

Attore: Utente

Entry Condition: PRN-000873 creata

Flusso di eventi:

1. Area utente → Prenotazioni mostra PRN-000873 con stato “Confermata”
2. Azioni disponibili:
 - Dettagli
 - Annulla
 - Sposta
3. Ricevuta RCPT-000873.pdf scaricabile

Exit condition: Prenotazione visibile e gestibile

UC 53 – Annullamento prenotazione

Attore: Utente

Entry Condition: PRN-000873 in stato “Confermata”

Flusso di eventi:

1. L’utente seleziona “Annulla”
2. Il sistema calcola tempo residuo:
 - Se $\geq 24h$ → penale 0% (R2)
 - Se $< 24h$ → penale 50% (R2)
3. L’utente conferma
4. Il sistema libera lo slot sui calendari
5. Il sistema invia e-mail di annullamento a utente e coach con esito economico

- (eventuale penale)
6. Il sistema registra eventuale rimborso/parziale e aggiorna la ricevuta

Exit condition: Prenotazione annullata secondo policy; calendari aggiornati

UC 54 – Spostamento prenotazione

Attore: Utente

Entry Condition: PRN-000873 in stato “Confermata”

Flusso di eventi:

1. L'utente seleziona “Sposta”
2. Il sistema mostra slot compatibili del coach
3. L'utente sceglie un nuovo slot
4. Il sistema verifica disponibilità e aggiorna la prenotazione
5. Il sistema invia e-mail di aggiornamento a utente e coach con nuovo .ics
6. Il sistema applica le policy economiche se lo spostamento implica annullamento del precedente slot a <24h

Exit condition: Prenotazione spostata o operazione annullata

UC 55 – Promemoria e check-in (Prenotazione)

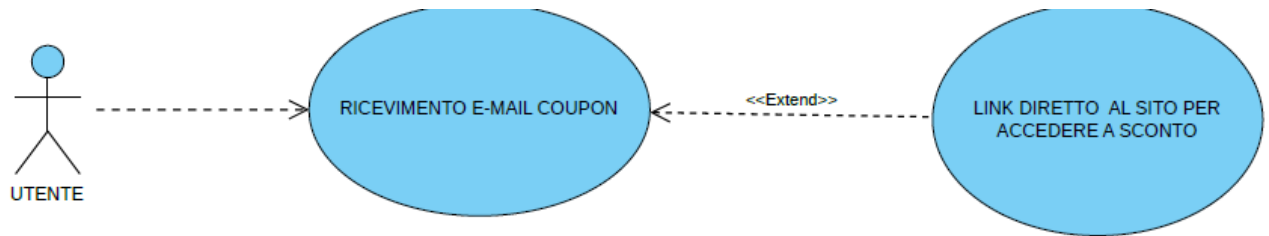
Attore: Sistema

Entry Condition: Prenotazione confermata

Flusso di eventi:

1. Invia promemoria automatici a T-48h e T-2h (push/e-mail)
2. Abilita check-in rapido in app da T-15'
3. Genera QR per accoglienza in struttura

Exit condition: Promemoria inviati; check-in disponibile



UC 56 – Apertura e-mail con coupon e deep link

Attore: Utente

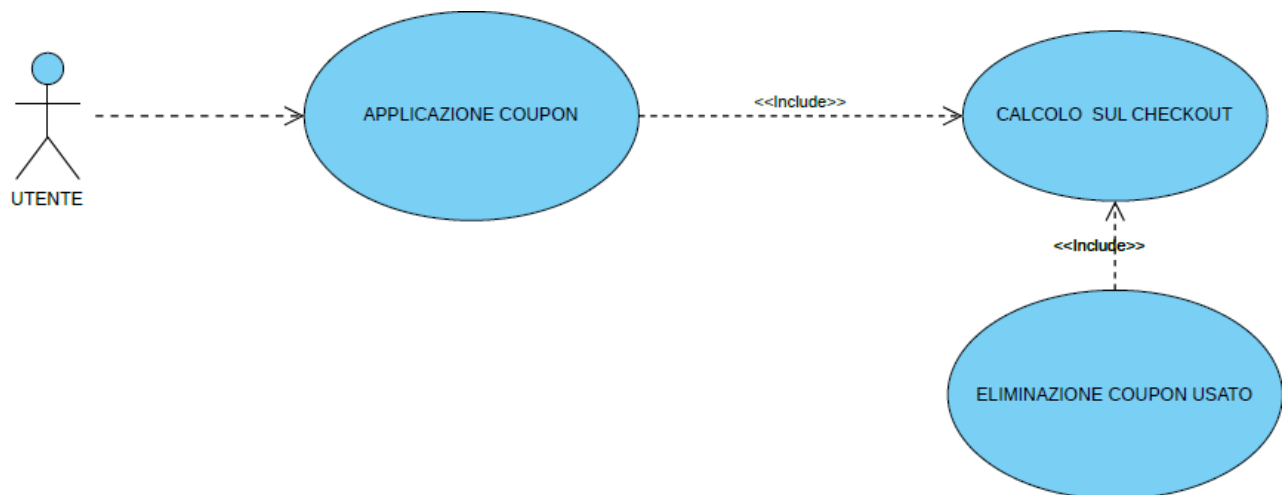
Entry Condition: E-mail promozionale ricevuta

Flusso di eventi:

1. Apre l’e-mail con oggetto “Benvenuto in FitCorp — Roma Centro: -20% sull’abbonamento annuale”
2. Visualizza codice ROMA-ANN20-2025 e seleziona “Attiva ora”
3. Il sistema apre TrackFit alla pagina Abbonamenti della struttura

Exit condition: Pagina Abbonamenti aperta via deep link

Alternativa: E-mail/codice non più valido → UC 65



UC 57 – Autenticazione e precompilazione coupon

Attore: Utente

Entry Condition: Deep link attivo; utente non autenticato

Flusso di eventi:

1. Inserisce username t.caio19@trackfit.it e password
2. Invia le credenziali
3. Il sistema autentica e reindirige alla scheda “Annuale 12 mesi” con campo coupon pre-

Exit condition: Utente autenticato su scheda prodotto con coupon valorizzato

Alternativa: Credenziali errate → UC 63

UC 58 – Autenticazione fallita (per coupon)

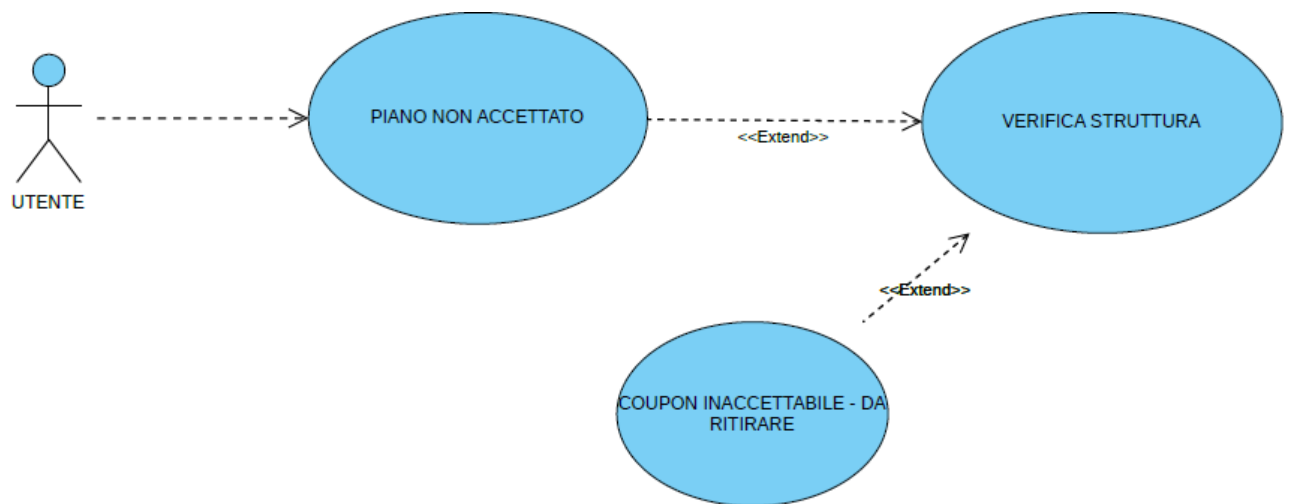
Attore: Utente

Entry Condition: Tentativo di accesso non riuscito

Flusso di eventi:

1. Il sistema mostra “username o password non corretti”
2. Rappresenta la schermata di autenticazione

Exit condition: Utente sulla schermata di accesso



UC 59 – Selezione piano e verifiche preliminari (Coupon)

Attore: Utente

Entry Condition: Scheda prodotto visibile con coupon valorizzato

Flusso di eventi:

1. Seleziona “Scegli piano”
2. Il sistema verifica:
 - Struttura target = Roma Centro →OK
 - Piano compatibile = Annuale →OK
 - Assenza abbonamenti attivi dell’utente sulla stessa struttura →OK
 - Validità coupon al tempo $T = 27/10/2025\ 18:00 < 31/10/2025\ 23:59$ →OK

- Limiti d'uso: per utente 0/1; globale < limite → OK
- Non cumulabile: assenza di altri sconti nel carrello → OK

Exit condition: Verifiche superate; passaggio al carrello

Alternative: Una qualunque verifica fallisce → UC 65

UC 60 – Coupon non valido / non applicabile

Attore: Sistema

Entry Condition: Almeno una condizione R1 fallita

Flusso di eventi:

1. Il sistema indica il motivo specifico: scaduto; struttura non corrispondente; piano non compatibile; già utilizzato; abbonamento attivo presente; sconto cumulabile vietato
2. Impedisce l'applicazione del coupon e propone proseguimento senza sconto o annullamento

Exit condition: Coupon rimosso/inefficace; l'utente decide se continuare o uscire

UC 61 – Carrello e applicazione coupon

Attore: Sistema

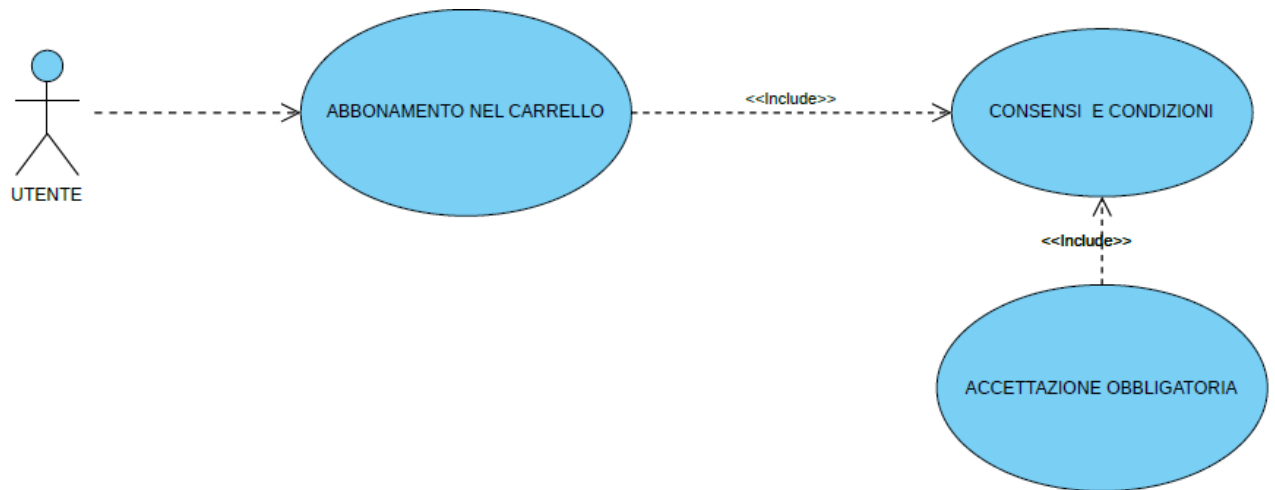
Entry Condition: Verifiche superate

Flusso di eventi:

1. Calcola il riepilogo economico secondo R2:
 - Prezzo base €480,00
 - Sconto 20% (coupon) -€96,00
 - Quota attivazione (non scontabile) €20,00
 - Totale €404,00 (IVA inclusa da listino)
2. Mostra "Procedi al pagamento"

Exit Condition: Carrello pronto al checkout

Alternativa: Presenza di un altro sconto → rimozione automatica per non cumulabilità e ricalcolo



UC 62 – Consensi e condizioni (Abbonamento)

Attore: Utente

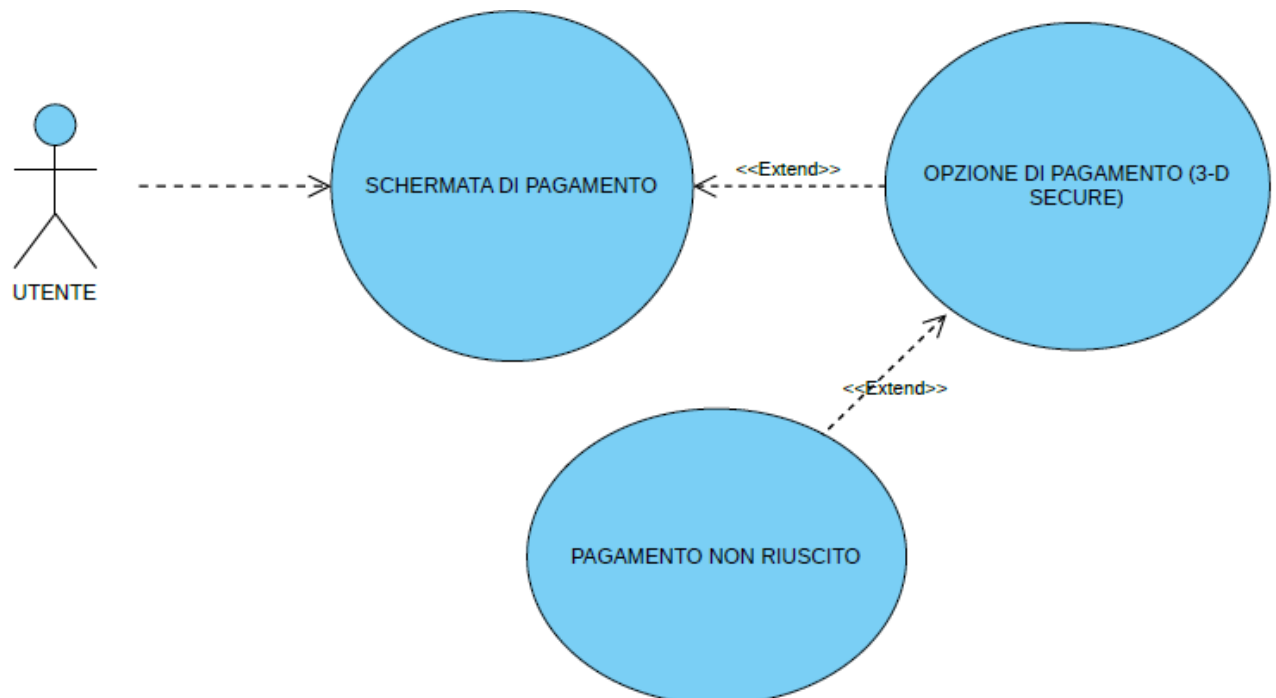
Entry Condition: Carrello pronto

Flusso di eventi:

1. Visualizza Termini e condizioni: uso struttura, privacy, rinnovo automatico (default OFF)
2. Seleziona i flag obbligatori
3. Continua

Exit Condition: Consensi registrati

Alternativa: Flag mancanti → messaggio errore; impossibile proseguire



UC 63 – Pagamento con SCA (3-D Secure) (Abbonamento)

Attore: Utente

Entry Condition: Consensi registrati; importo €404,00

Flusso di eventi:

1. Seleziona Carta VISA **** 4242
2. Clic su “Paga €404,00” → apertura 3-D Secure
3. Inserisce OTP 998877 → *Conferma*
4. Esito “Autorizzato”; il sistema genera ID transazione PAY-2025-10-27-00421

Exit Condition: Pagamento autorizzato (R6)

Alternativa: Pagamento rifiutato/SCA fallita → UC 69

UC 64 – Pagamento non riuscito (Abbonamento)

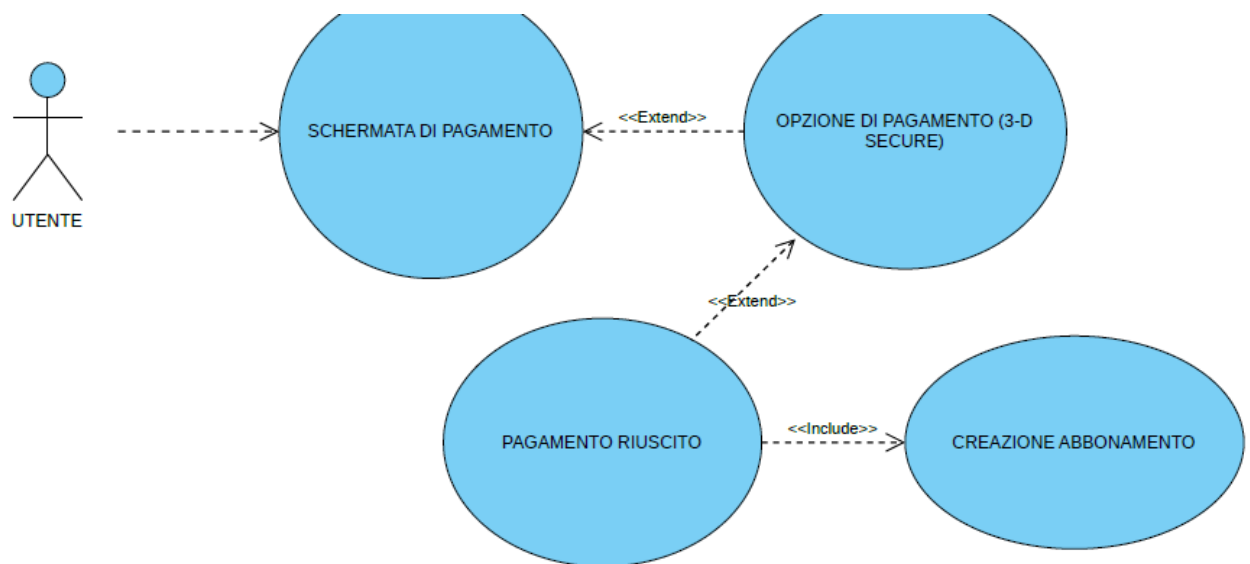
Attore: Utente

Entry Condition: Errore gateway, OTP errata, fondi insufficienti

Flusso di eventi:

1. Il sistema mostra motivo del rifiuto
2. Ritorna al riepilogo pagamento

Exit Condition: Nessuna creazione abbonamento; l’utente può riprovare



UC 65 – Creazione abbonamento

Attore: Sistema

Entry Condition: Pagamento riuscito (UC 68)

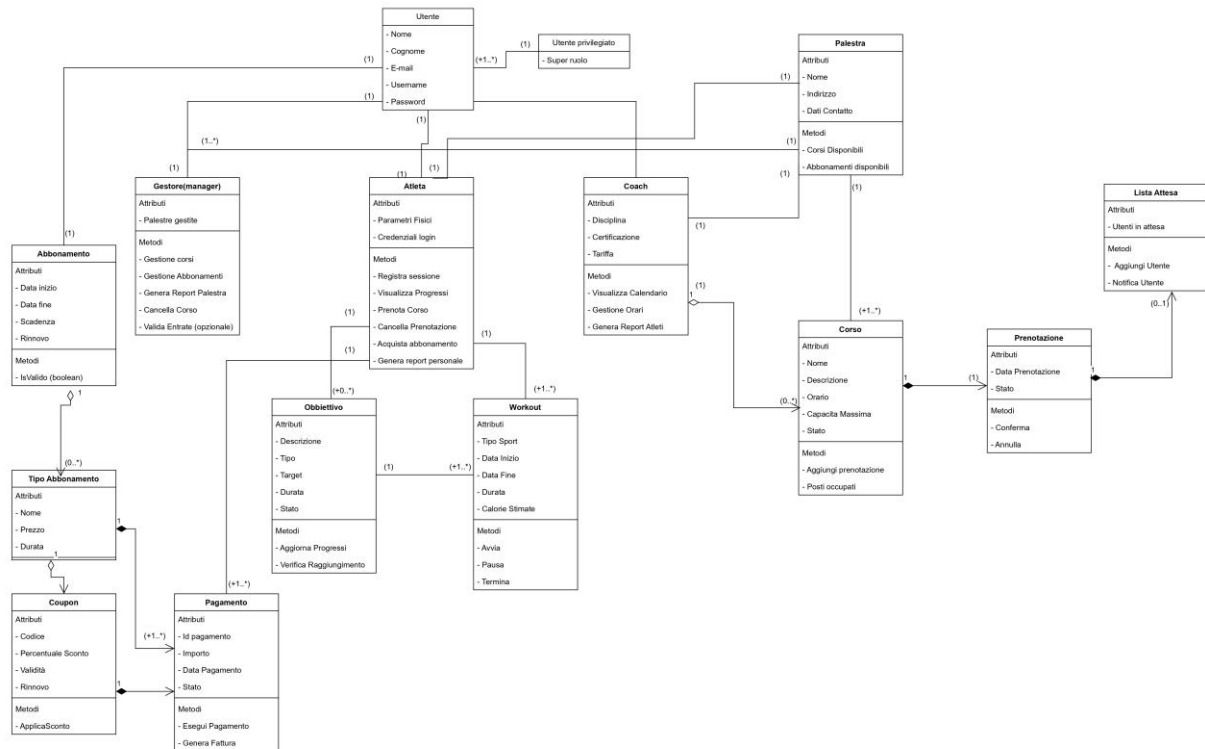
Flusso di eventi:

1. Il sistema crea l'abbonamento per l'utente t.caio19@trackfit.it.
2. Il sistema associa l'abbonamento alla struttura "FitCorp – Palestra Roma Centro".
3. Il sistema registra l'uso del coupon ROMA-ANN20-2025.
4. Il sistema invia e-mail di conferma acquisto e riepilogo abbonamento.
5. Il sistema genera la ricevuta/fattura.

Exit Condition: Abbonamento attivo e associato all'utente.

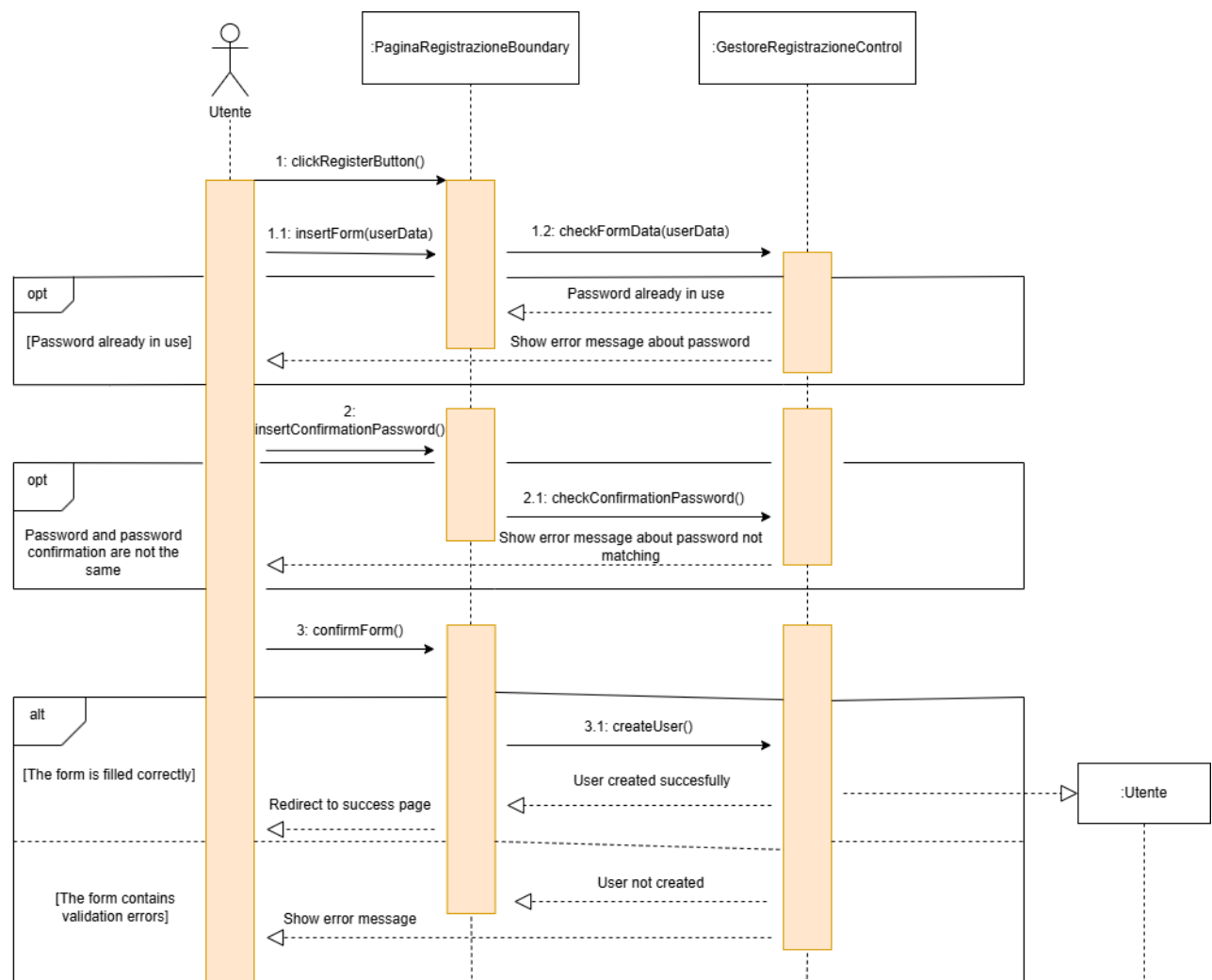
3.4.3. Object model:

Class diagram:

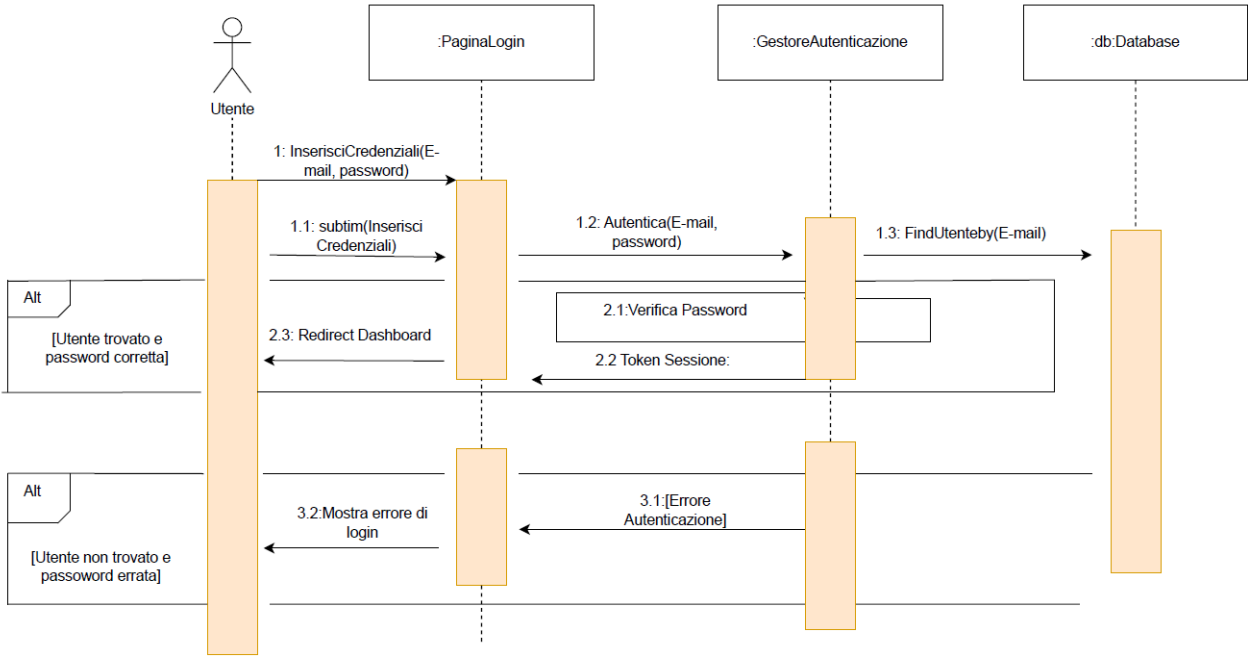


3.4.4. Dynamic model:

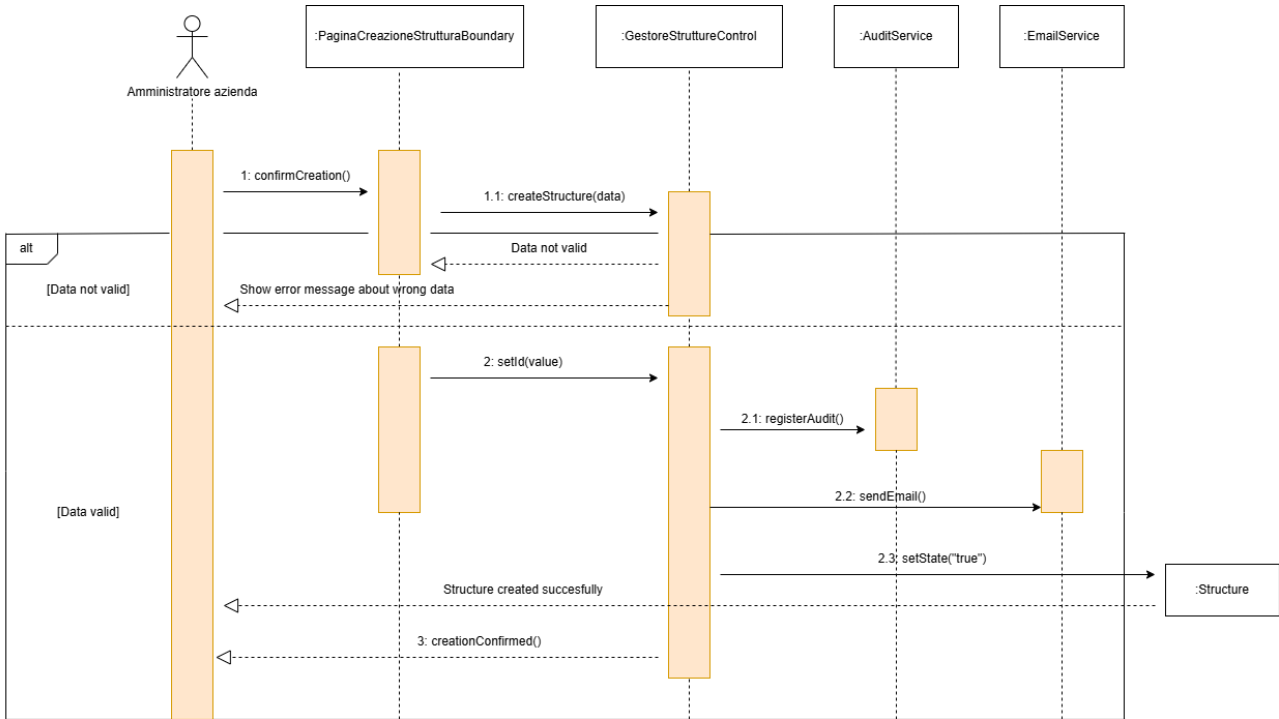
Sequence diagram 1 (riferimento ad UC 1):



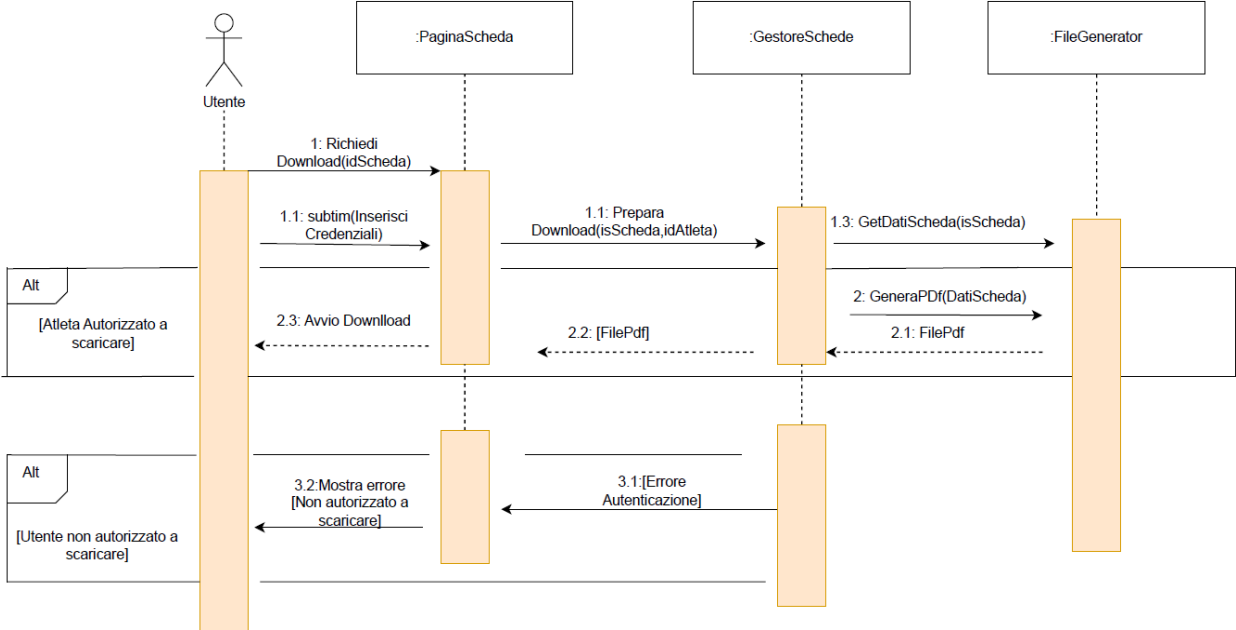
Sequence diagram 2 (riferimento ad UC 2):



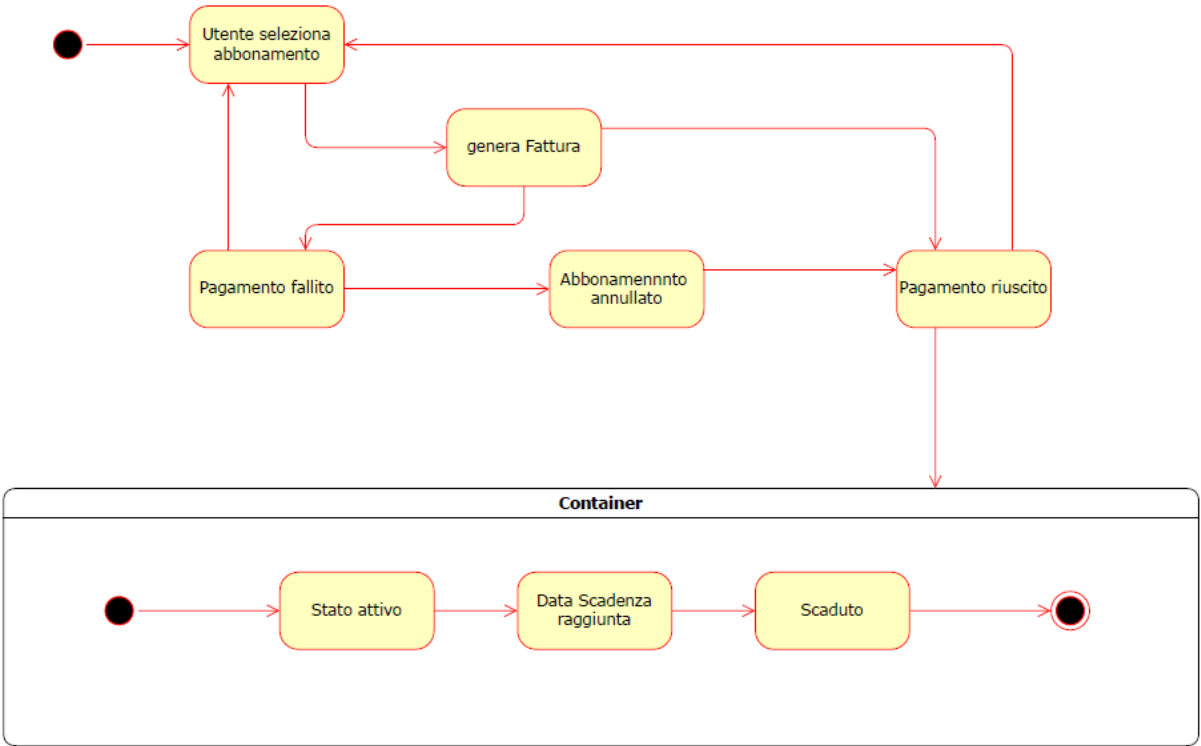
Sequence diagram 3 (riferimento ad UC 66):



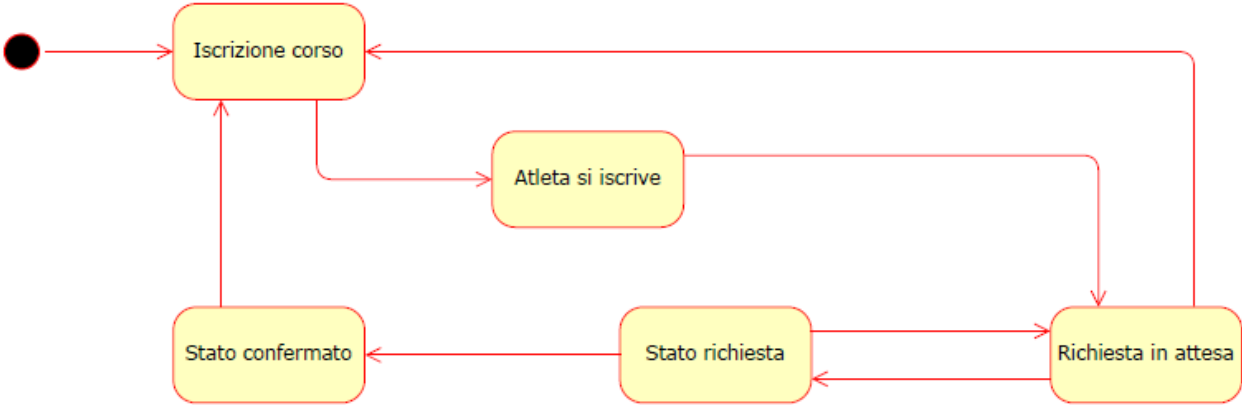
Sequence diagram 4 (riferimento ad UC 17):



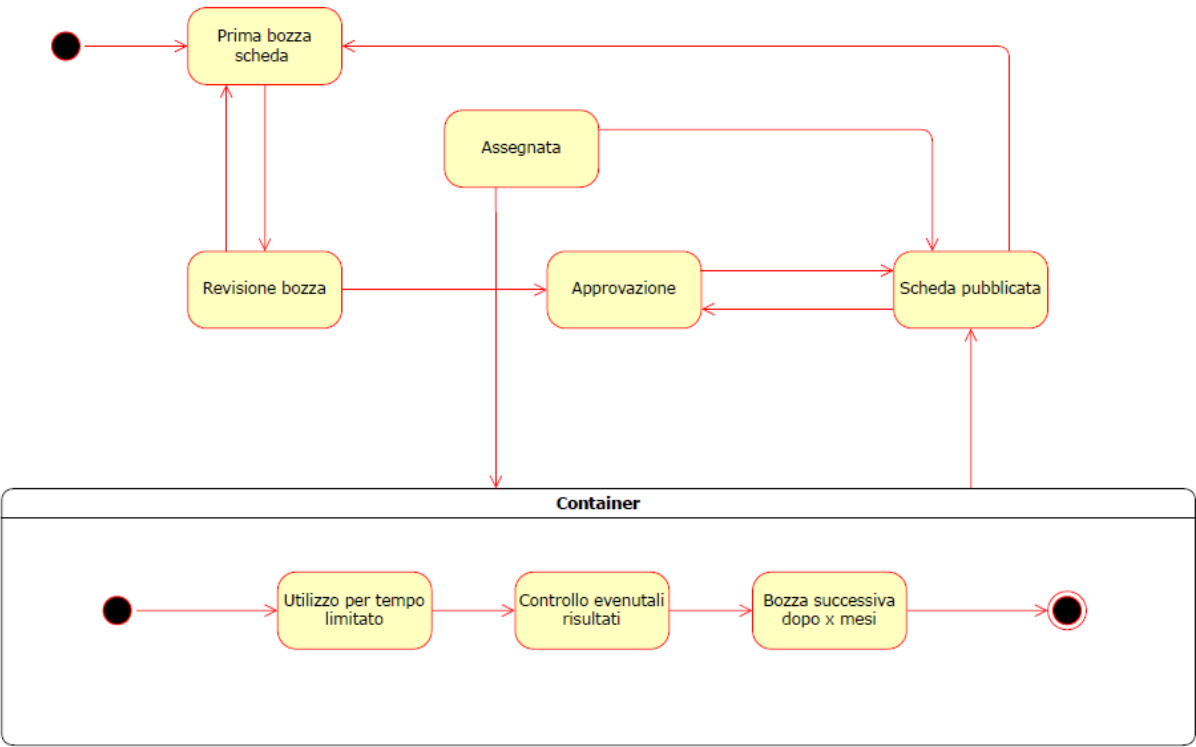
Statechart diagram 1:



Statechart diagram 2:

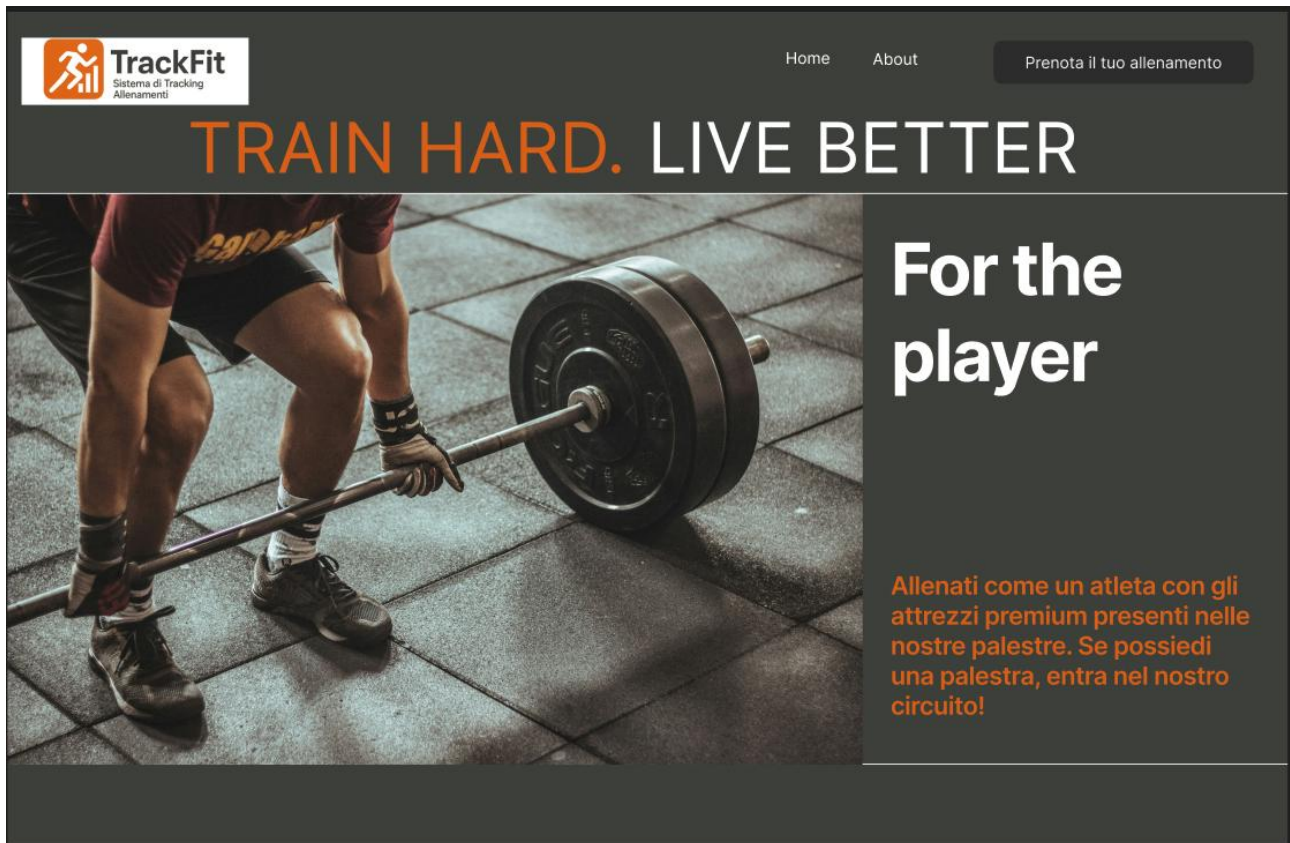


Statechart diagram 3:



3.4.5. Mockups e interfaccia navigazionale dell'utente:

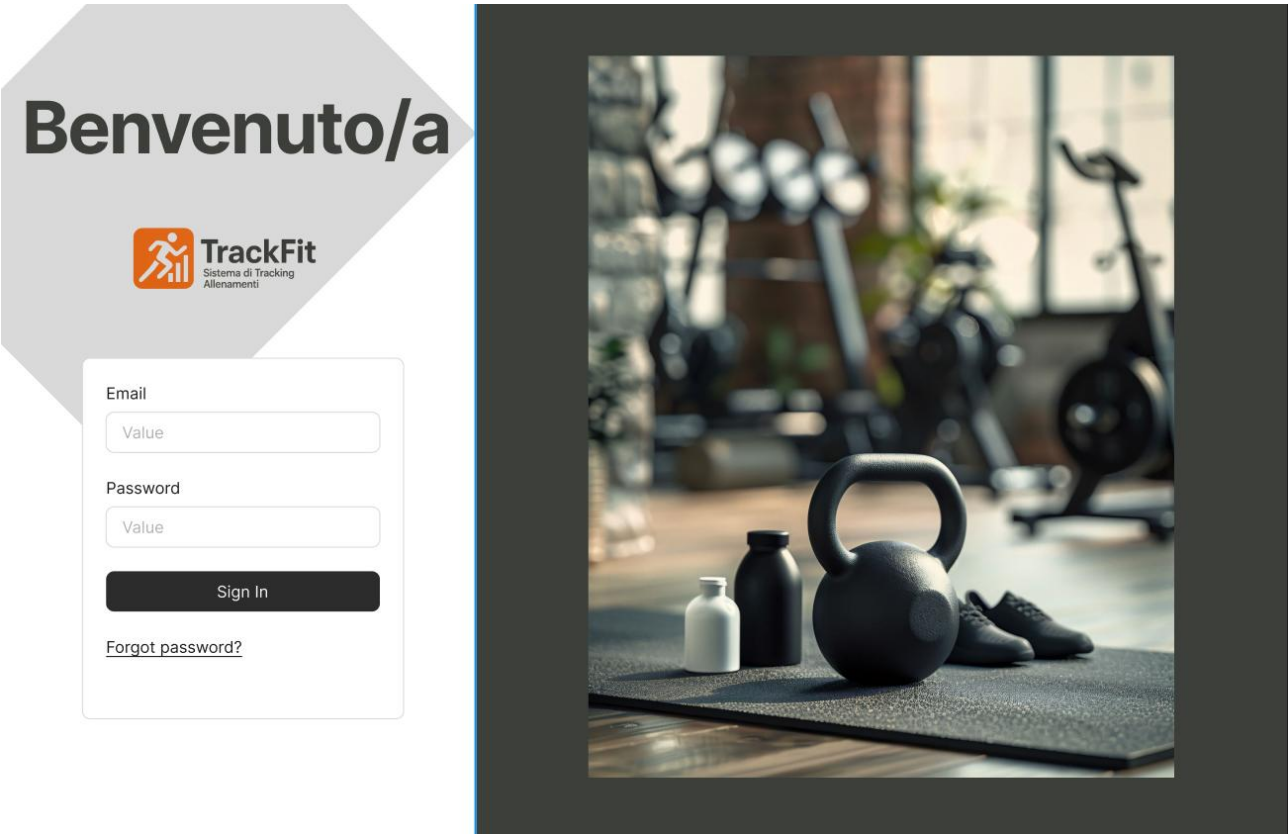
Mockup 1:



Mockup 2:



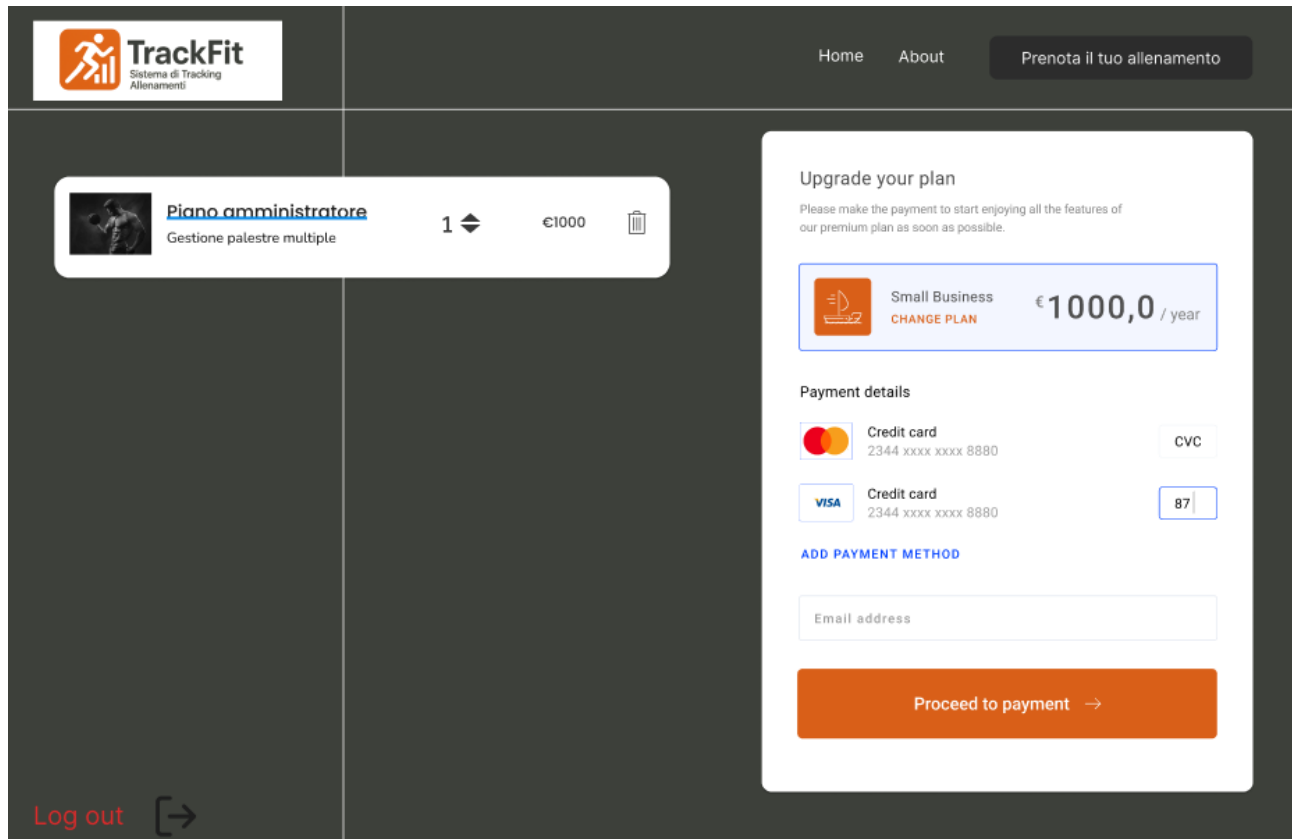
Mockup 3:



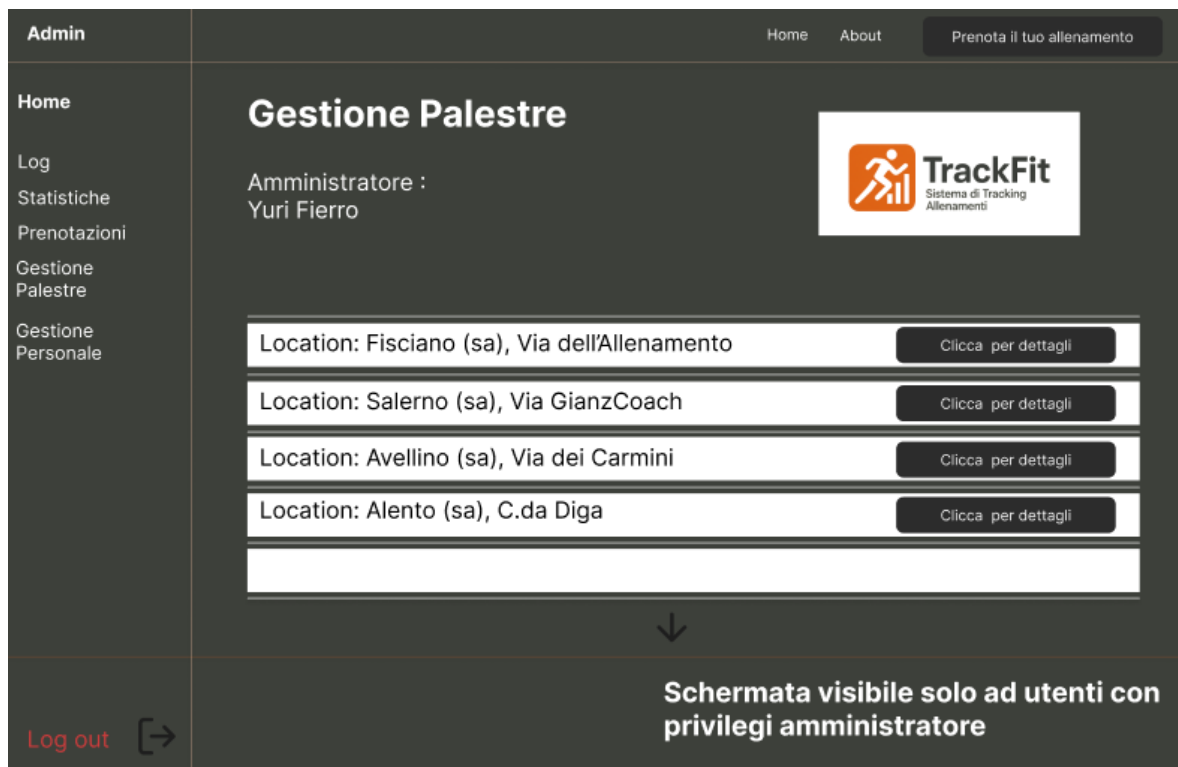
Mockup 4:



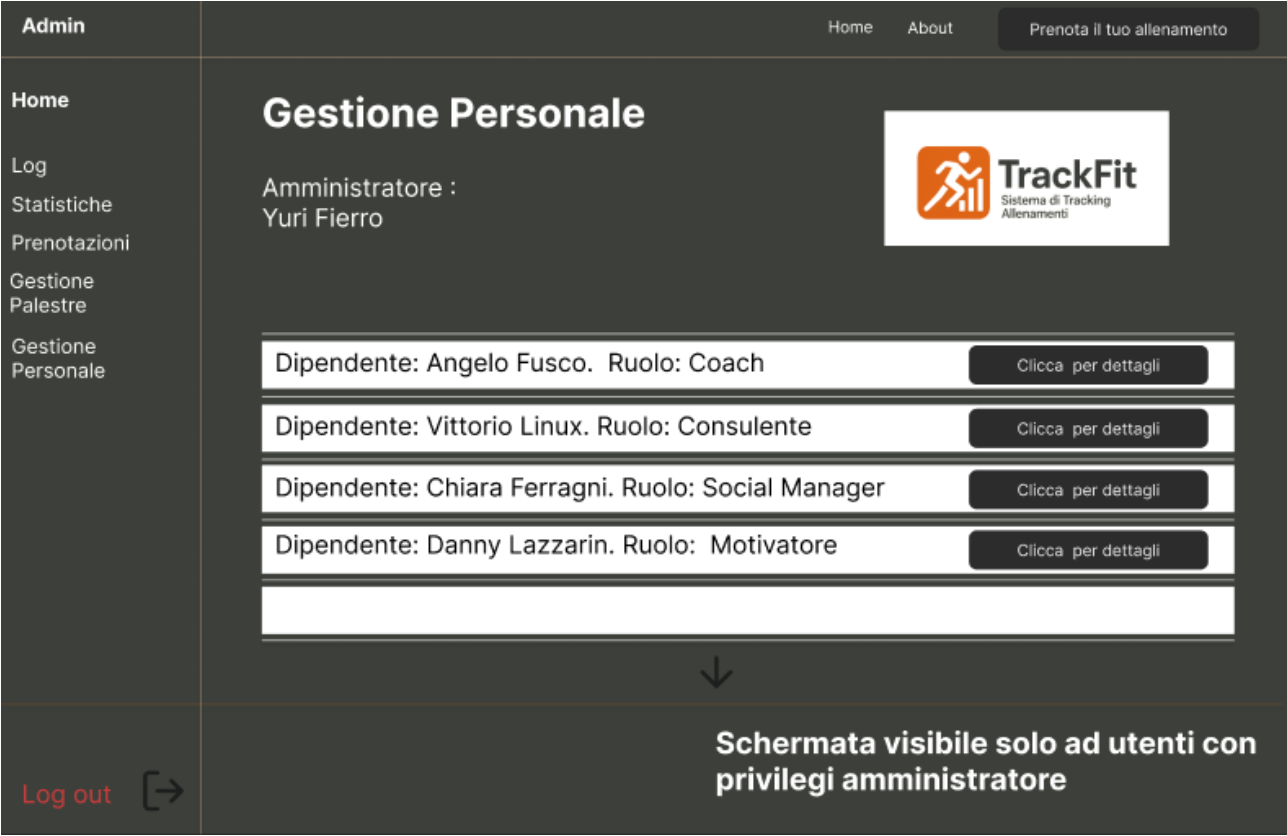
Mockup 5:



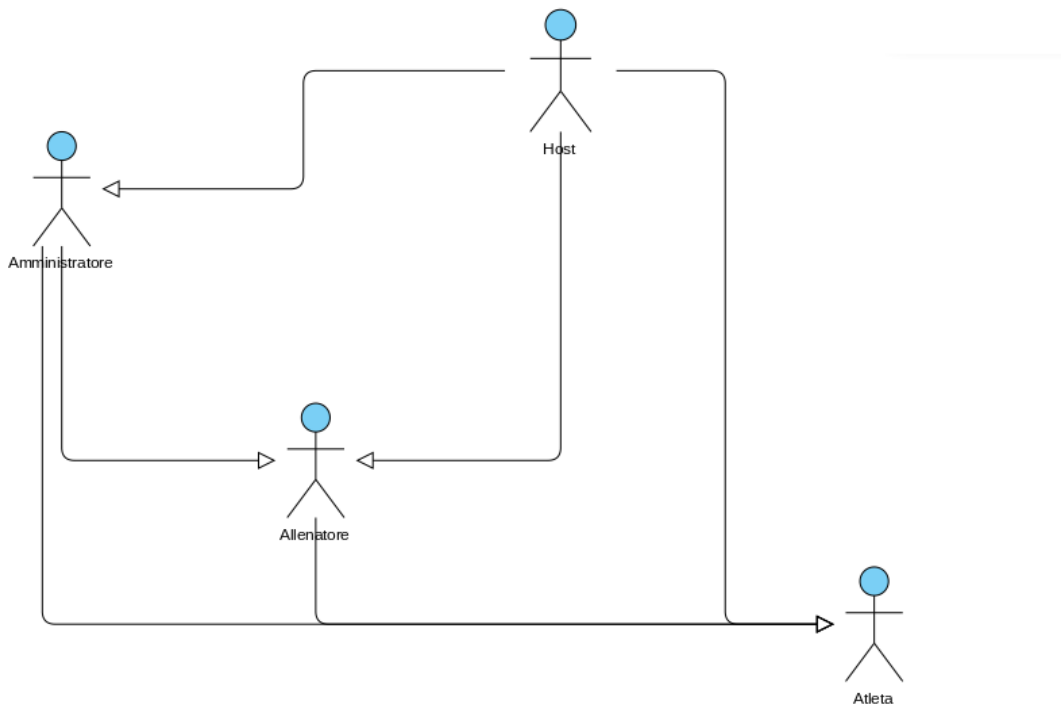
Mockup 6:



Mockup 7:

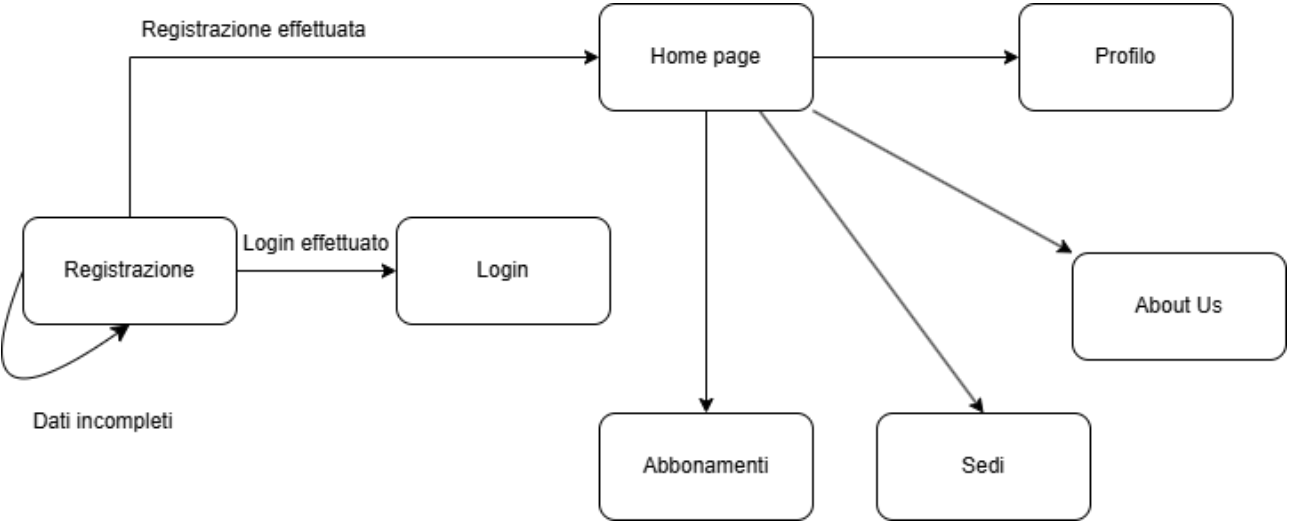


System's user hierarchy:

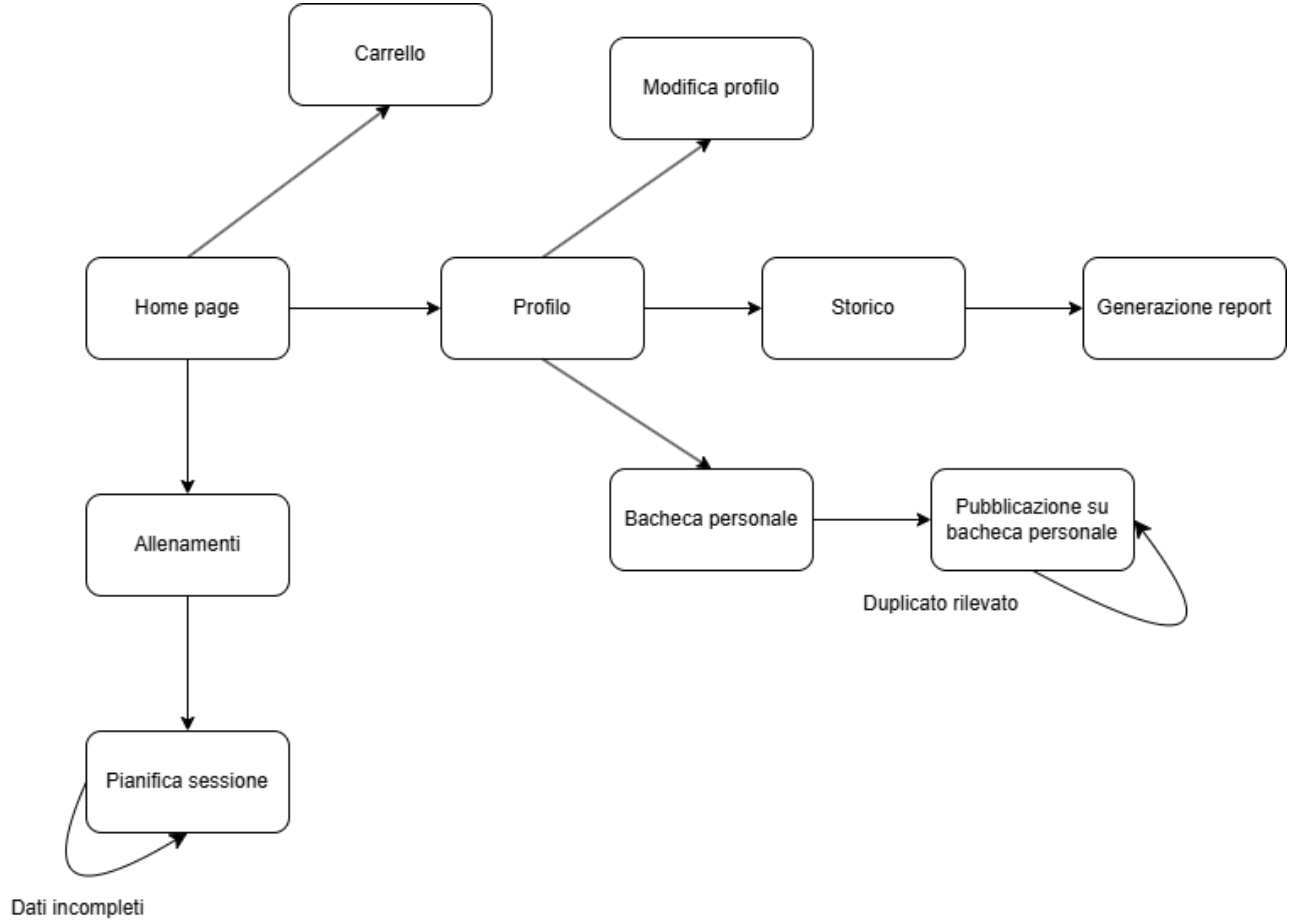


3.4.6. Navigational Diagram

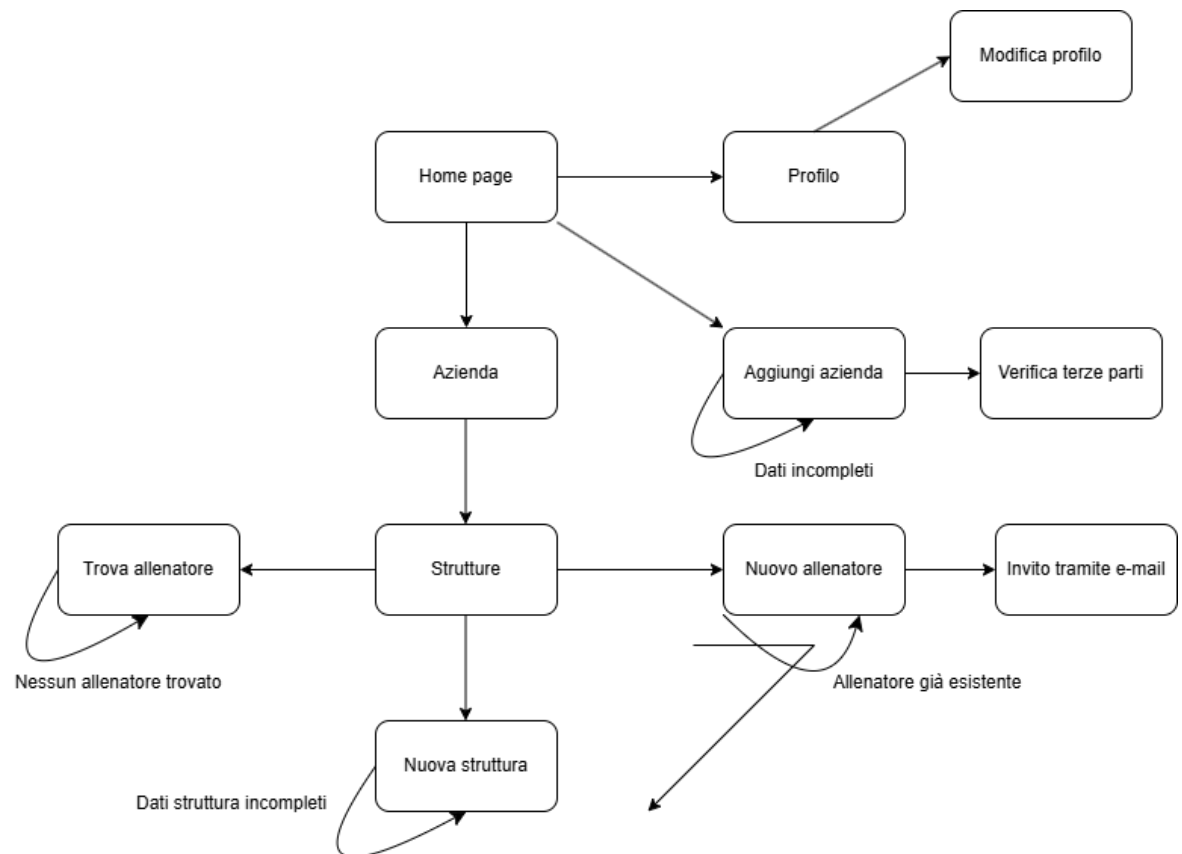
Utente guest:



Atleta:



Amministratore azienda:



Allenatore:

