

**UNIVERSITÀ DEGLI STUDI DI  
ROMA TOR VERGATA**



**Documentazione progetto ISPW  
Roberto Di Muro 0327974**

## Sommario

<b>1.</b>	<b><i>Specifiche e requisiti del progetto</i></b>	<b>3</b>
1.1.1	Introduzione .....	3
1.1.2	Panoramica del sistema .....	3
1.1.3	Requisiti Hardware e Software.....	3
1.1.4	Sistemi correlati .....	3
<b>1.2</b>	<b>User stories</b>	<b>4</b>
1.2.1	Us-1 .....	4
1.2.2	Us-2.....	4
1.2.3	Us-3.....	4
<b>1.3</b>	<b>Functional requirement</b>	<b>4</b>
1.3.1	Fr-1 .....	4
1.3.2	Fr-2 .....	4
1.3.3	Fr-3 .....	4
<b>1.4</b>	<b>Use case</b>	<b>4</b>
1.4.1	Use case diagram .....	4
1.4.2	Use case internal steps.....	5
<b>2.</b>	<b>Storyboards</b>	<b>6</b>
<b>3.</b>	<b>Design</b>	<b>9</b>
3.1	VOPC .....	9
3.2	Design level.....	10
3.3	Design patterns .....	10
3.4	Activity diagram .....	11
3.5	Sequence diagram.....	13
3.6	State diagram.....	13
<b>4.</b>	<b>Link utili</b> .....	<b>14</b>

# 1. Specifiche e requisiti del progetto

## 1.1.1 Introduzione

Questo documento ha lo scopo di presentare gli obiettivi e la progettazione dietro l'applicazione EzGym, un app utile per la comunicazione tra atleta e personal trainer se non si ha la possibilità di comunicare di persona. Verranno trattate le funzionalità principali ed il processo di sviluppo e di progettazione che c'è dietro EzGym.

## 1.1.2 Panoramica del sistema

L'applicazione è stata progettata per facilitare le interazioni tra atleta e personal trainer. In particolare, ha due funzionalità principali:

- **Richiesta della scheda:** l'applicazione permette agli atleti che si registrano nel sistema di inserire i propri dati personali attraverso un questionario mirato e di richiedere al personal trainer una scheda di allenamento a personalizzata.
- **Creazione della scheda:** l'applicazione permette al personal trainer di visualizzare le richieste degli atleti, di visualizzare il loro dati personali e di creare una scheda personalizzata inserendo per ogni giorno un numero arbitrario di esercizi.

## 1.1.3 Requisiti Hardware e Software

**Requisiti software:** java SDK, IDE: IntelliJ IDEA e database MySQL.

**Requisiti hardware:** nessun requisito particolare, basta un computer con un processore moderno.

## 1.1.4 Sistemi correlati

Nel panorama delle applicazioni dedicate al fitness e alla gestione degli allenamenti esistono diverse soluzioni che offrono funzionalità simili a quelle proposte da EzGym, pur differenziandosi per obiettivi e modalità di utilizzo.

- **MyFitnessPal** è una piattaforma orientata principalmente al monitoraggio dell'alimentazione e dell'attività fisica quotidiana. Permette agli utenti di tenere traccia delle calorie, dei macronutrienti e degli allenamenti svolti, offrendo un'interfaccia semplice e accessibile. Tuttavia, non fornisce un supporto strutturato alla creazione di schede di allenamento personalizzate da parte di personal trainer.
- **Trainerize** è una piattaforma più orientata ai professionisti del settore fitness, consentendo ai personal trainer di creare e assegnare programmi di allenamento ai propri clienti e di monitorarne i progressi. Sebbene offra funzionalità avanzate, l'interfaccia risulta meno immediata per l'utente meno esperto e maggiormente focalizzata su un contesto professionale.

**EzGym** si colloca in questo contesto come una soluzione che unisce semplicità d'uso e personalizzazione, permettendo agli atleti di richiedere schede di allenamento e ai personal trainer di crearle e gestirle in modo strutturato. L'applicazione mira a semplificare la comunicazione tra atleta e personal trainer, centralizzando la gestione degli allenamenti e offrendo un'esperienza user-friendly sia per utenti alle prime armi sia per professionisti del settore.

## 1.2 User stories

### 1.2.1 Us-1

Come atleta, voglio compilare ed inviare i miei dati fisici per richiedere una scheda di allenamento, in modo che il personal trainer possa creare un piano adatto ai miei obiettivi.

### 1.2.2 Us-2

Come atleta, voglio visualizzare la scheda di allenamento creata dal personal trainer, in modo che possa eseguire i miei allenamenti senza la sua presenza.

### 1.2.3 Us-3

Come personal trainer, voglio visualizzare i dati e la richiesta dell'atleta e creare la sua scheda di allenamento, in modo da creare un programma personalizzato.

## 1.3 Functional requirement

### 1.3.1 Fr-1

Il sistema deve fornire al Personal Trainer autenticato la possibilità di visualizzare un riepilogo dell'atleta che ha richiesto la scheda di allenamento, includendo nome, cognome, e-mail, genere, età (anni), peso (kg), altezza (cm), obiettivo, livello di attività, giorni di allenamento e BMI calcolato da peso e altezza con arrotondamento a due decimali.

### 1.3.2 Fr-2

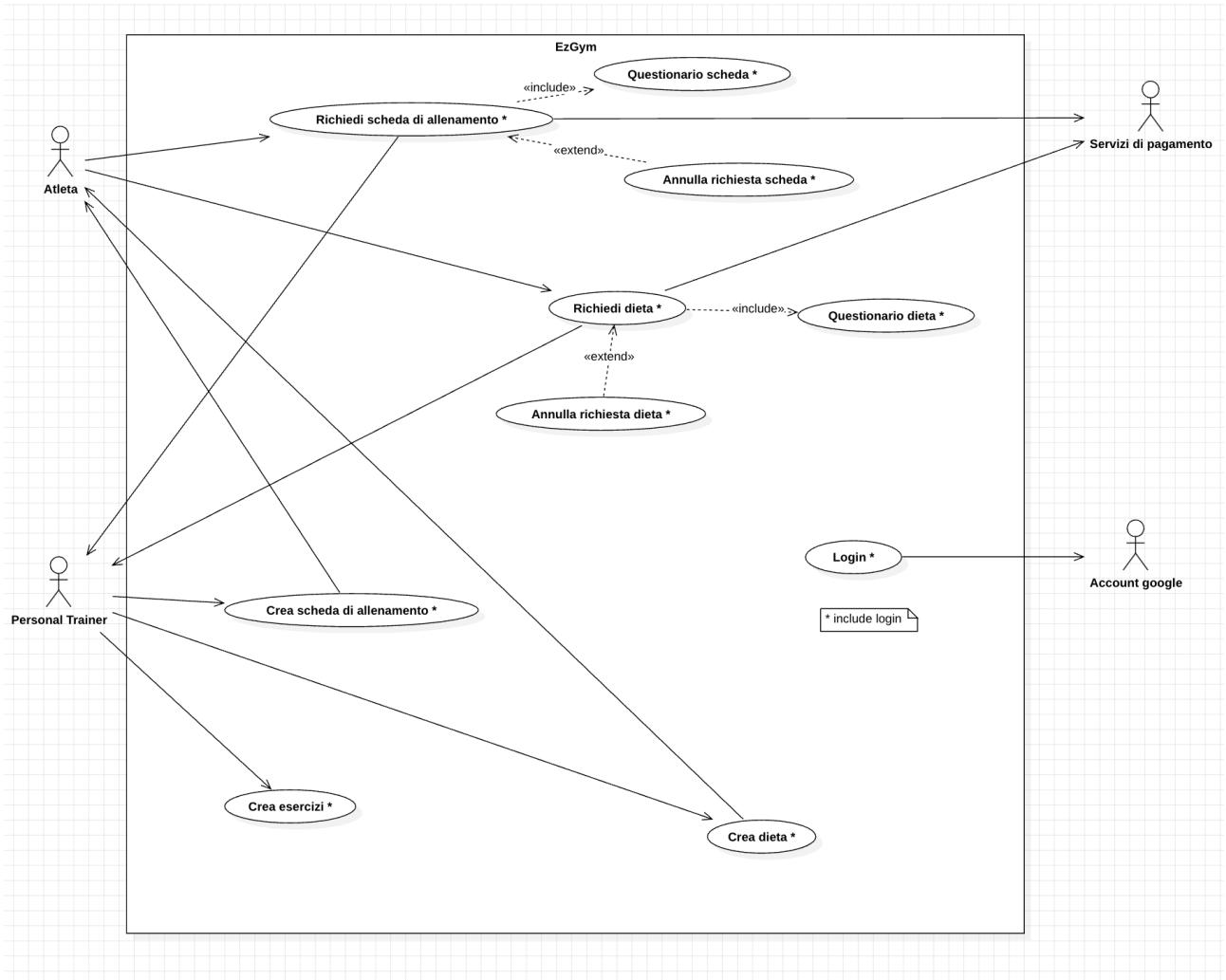
Il sistema deve fornire al Personal Trainer autenticato la possibilità di creare e salvare una scheda di allenamento associata all'atleta richiedente, che contenga almeno una sessione, in cui per ogni esercizio il sistema memorizza nome esercizio, serie, ripetizioni, tempo di recupero, tipologia e note.

### 1.3.3 Fr-3

Il sistema deve fornire all'atleta autenticato la possibilità di richiedere una scheda di allenamento, inserendo genere, età, peso, altezza, obiettivo, livello di attività attuale e quanti giorni ci si vuole allenare.

## 1.4 Use case

### 1.4.1 Use case diagram



## 1.4.2 Use case internal steps

Qui vengono mostrati gli internal steps relativi al caso d'uso **Richiedi scheda**.

### Flusso principale:

1. L'atleta effettua il login
2. Il sistema verifica le credenziali
3. Il sistema rende disponibili le funzionalità per l'atleta autenticato
4. L'atleta avvia una richiesta di scheda di allenamento
5. Il sistema avvia la compilazione di un questionario
6. L'atleta inserisce i dati richiesti
7. L'atleta conferma il questionario
8. Il sistema registra la richiesta di scheda associata all'atleta
9. Il sistema notifica la richiesta al personal trainer
10. Il sistema notifica l'avvenuta richiesta di scheda di allenamento

### Estensioni:

#### 2.a) Credenziali non valide:

- 2.a1) Sistema informa l'atleta dell'errore di autenticazione.
- 2.a2) Sistema richiede nuove credenziali. **Goto 1.**

#### 4.a) Questionario incompleto o dati non validi:

- 4.a1) Il sistema segnala quali campi non sono stati compilati o non sono validi
- 4.a2) Il sistema richiede l'aggiornamento dei dati. **Goto 3**

#### 5.a) Notifica al personal trainer non riuscito:

- 5.a1) Sistema registra la richiesta con stato “da inoltrare”.
- 5.a2) Sistema informa l’atleta che la richiesta è stata registrata ma non inoltrata.
- 5.a3) Sistema termina il caso d’uso. **Goto Exit.**

## 2. Storyboards

Qui vengono mostrate alcune funzionalità di EzGym sotto forma di storyboards.

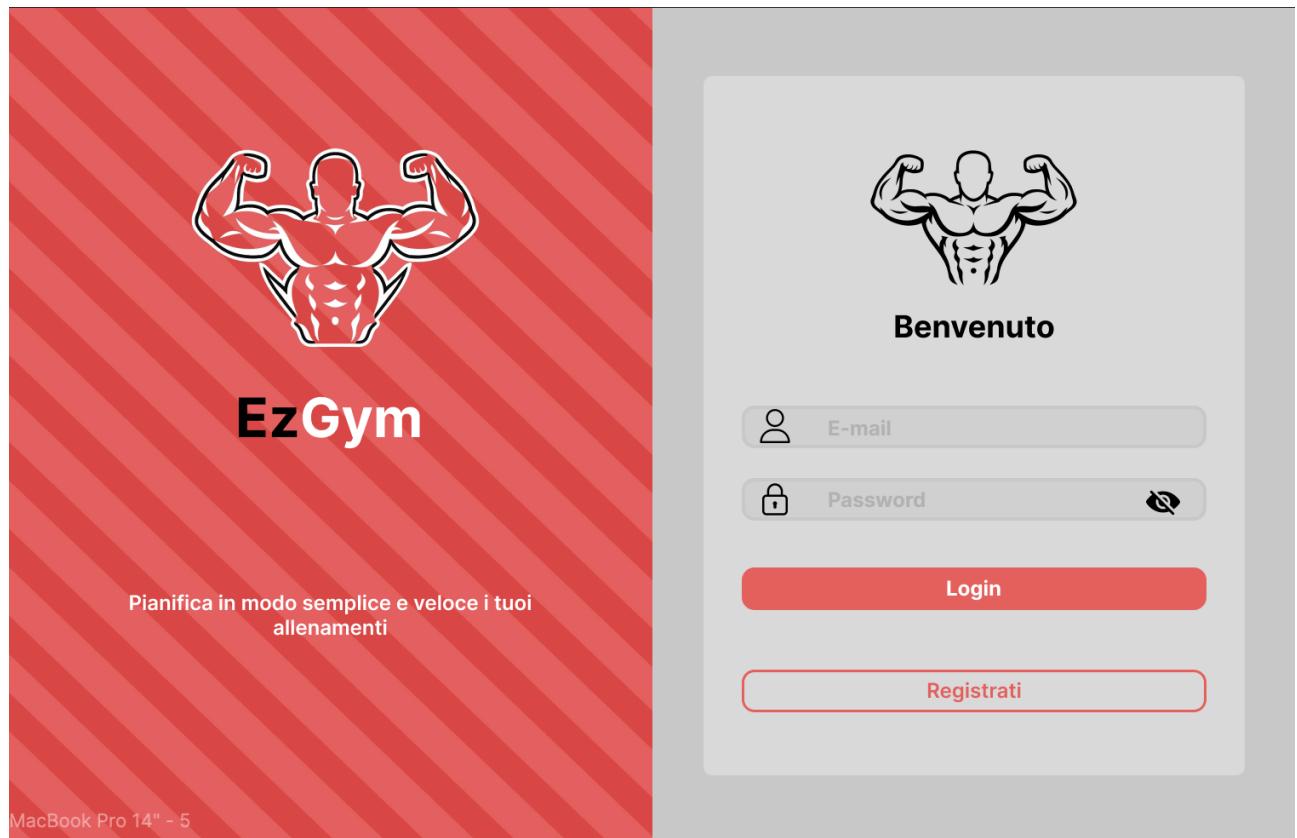


Figura 1: schermata di login

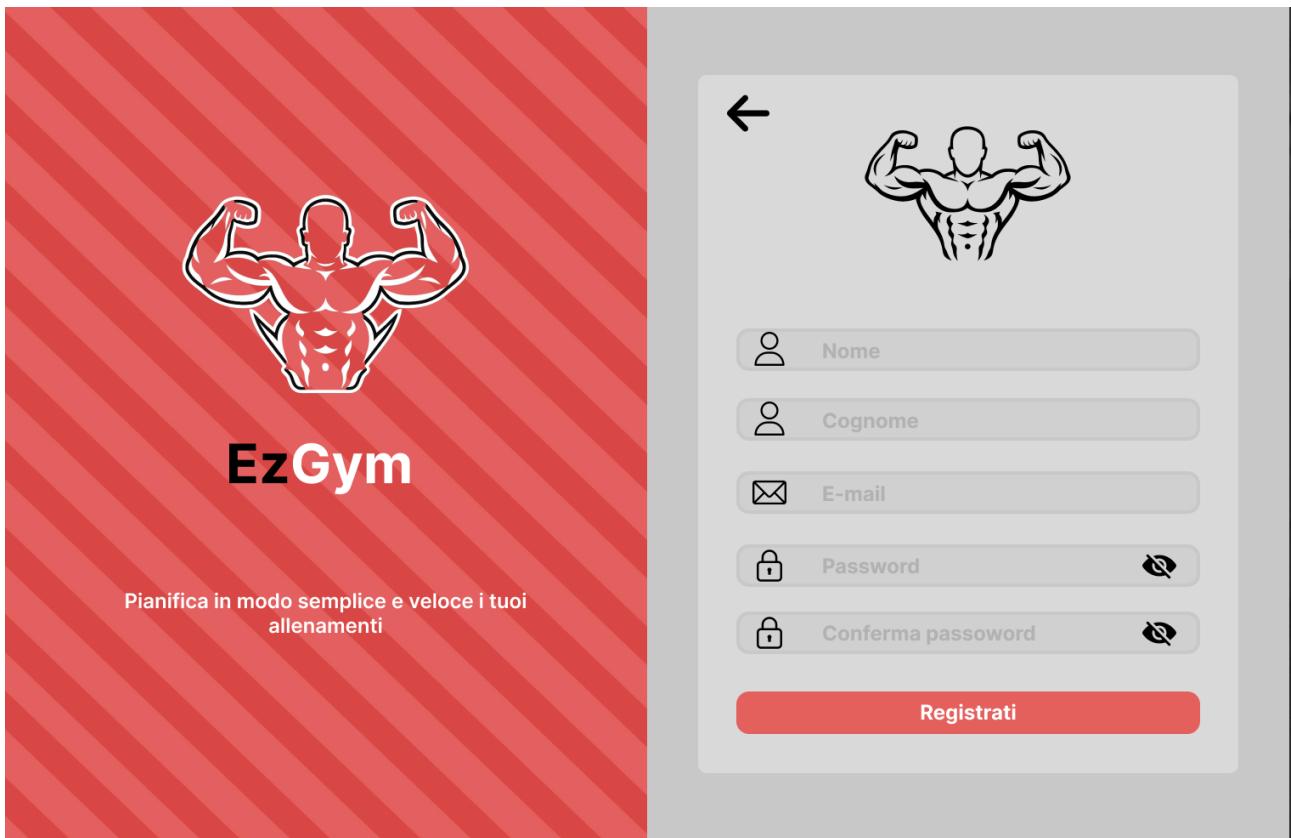


Figura 2: schermata di registrazione



Figura 3: schermata home dell'atleta



Figura 4: schermata home del persona trainer

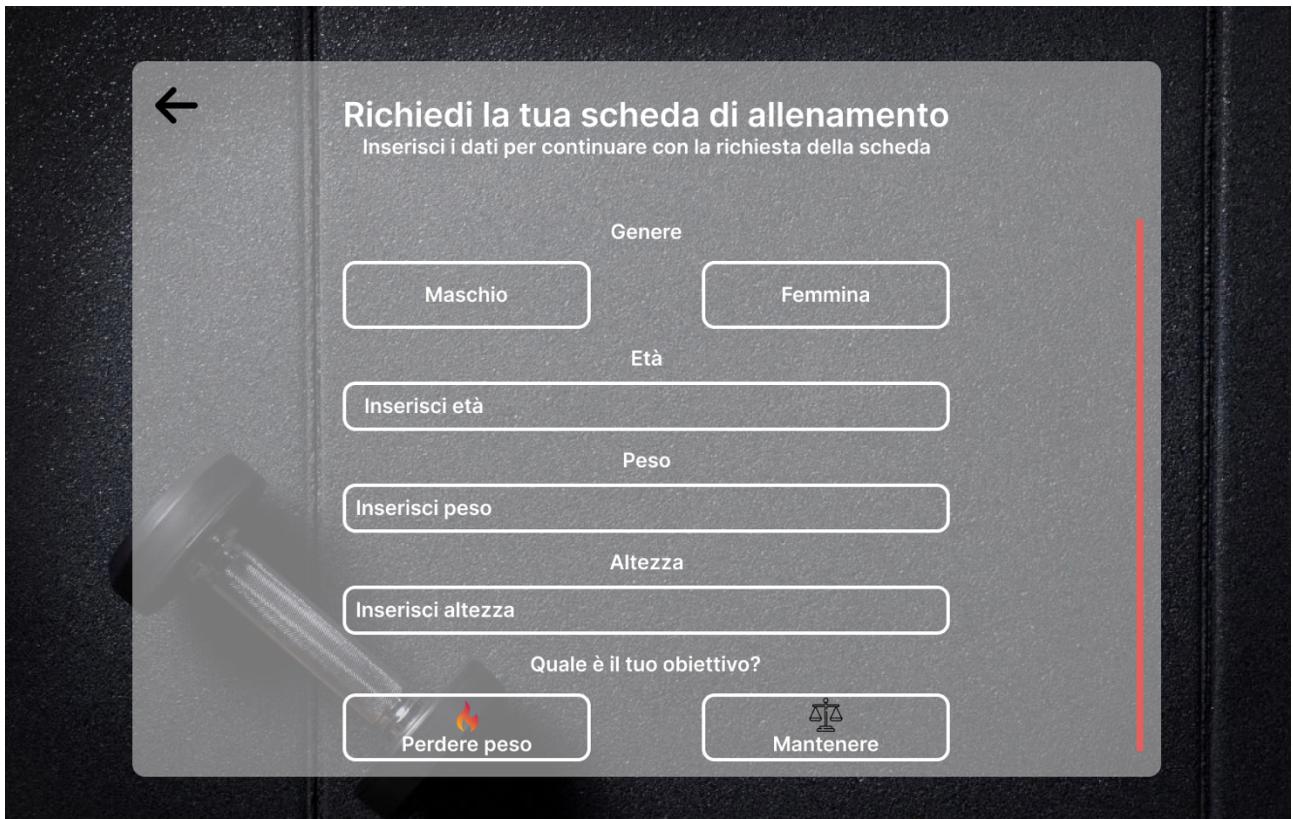


Figura 5: schermata questionario

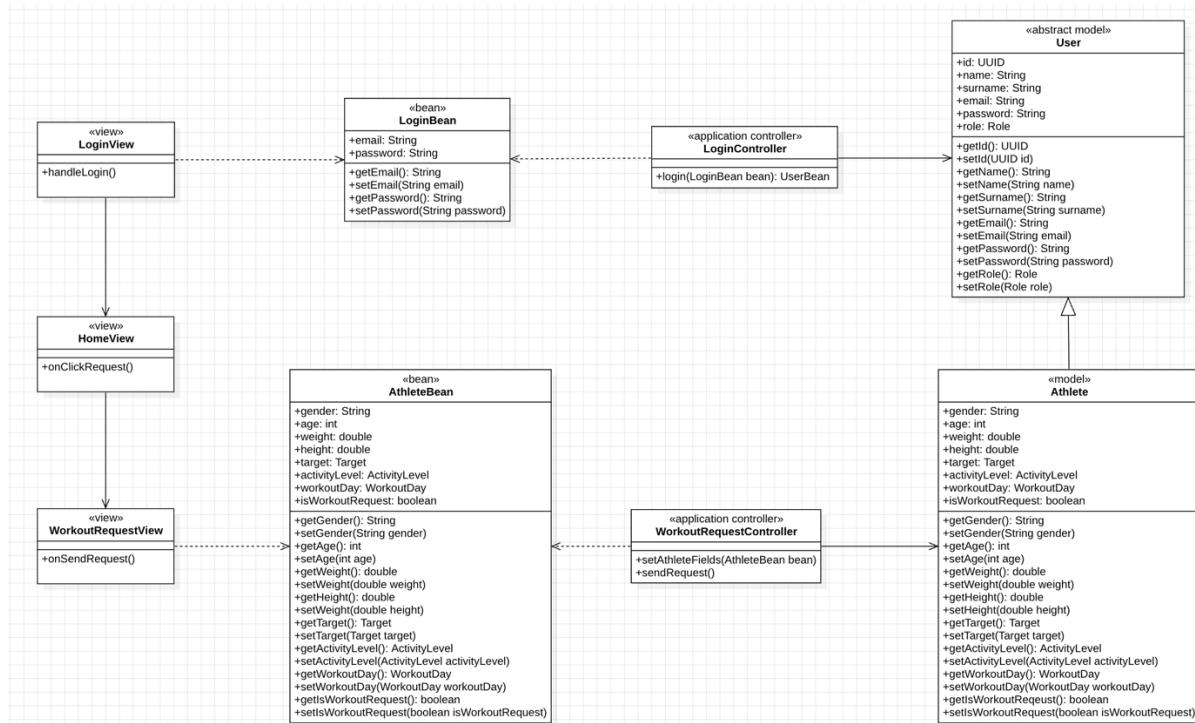


Figura 6: schermata creazione scheda

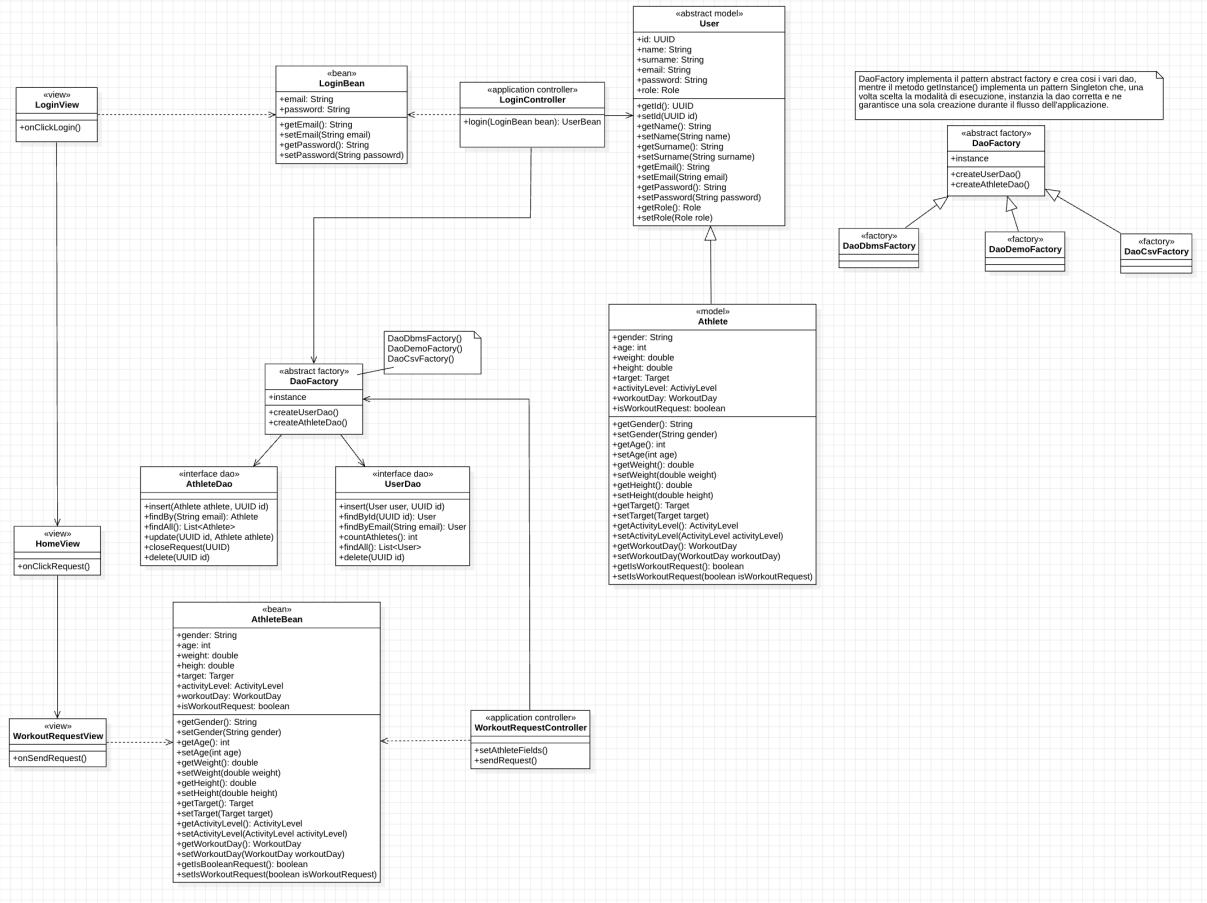
### 3. Design

I diagrammi che seguono sono presenti nella cartella *deliverables* del progetto.

#### 3.1 VOPC



## 3.2 Design level



## 3.3 Design patterns

Nell'applicazione EzGym sono stati adottati in particolare i pattern Abstract Factory e Singleton, al fine di garantire flessibilità, coerenza e una gestione centralizzata dei componenti di persistenza e configurazione.

È stato utilizzato il pattern Abstract Factory per la creazione delle famiglie di DAO in base alla modalità di esecuzione dell'applicazione. La modalità viene scelta dall'utente all'avvio dell'applicazione ed è gestita dal file di configurazione Setup.java, che determina quale implementazione concreta dei DAO debba essere istanziata. In questo modo, il resto dell'applicazione rimane indipendente dalla specifica tecnologia di persistenza utilizzata.

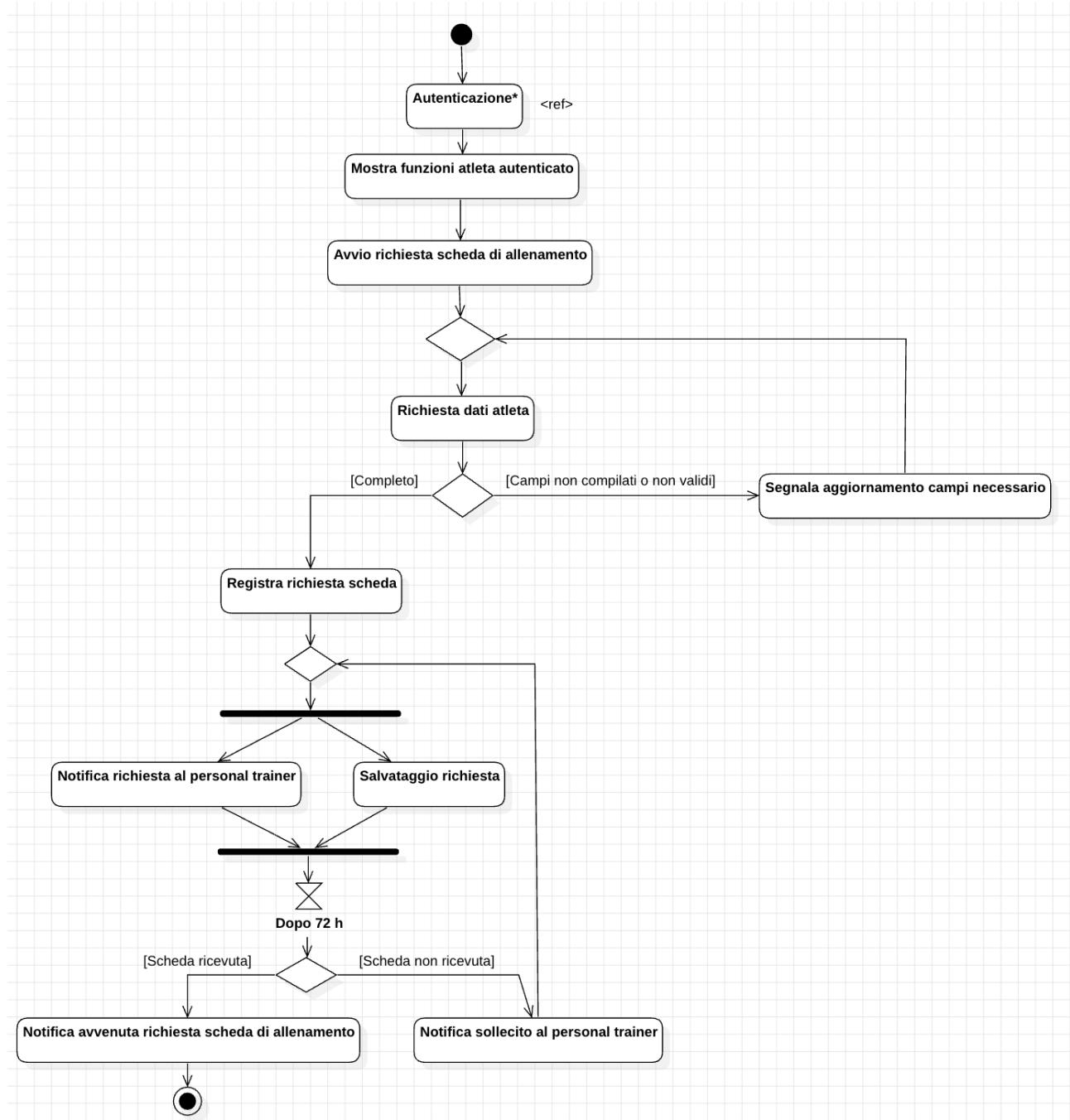
Contestualmente, è stato adottato il pattern Singleton per garantire l'esistenza di un'unica istanza sia della factory dei DAO sia dei singoli DAO concreti. Questo approccio evita creazioni multiple non necessarie, assicura un punto di accesso globale e coerente agli oggetti di persistenza.

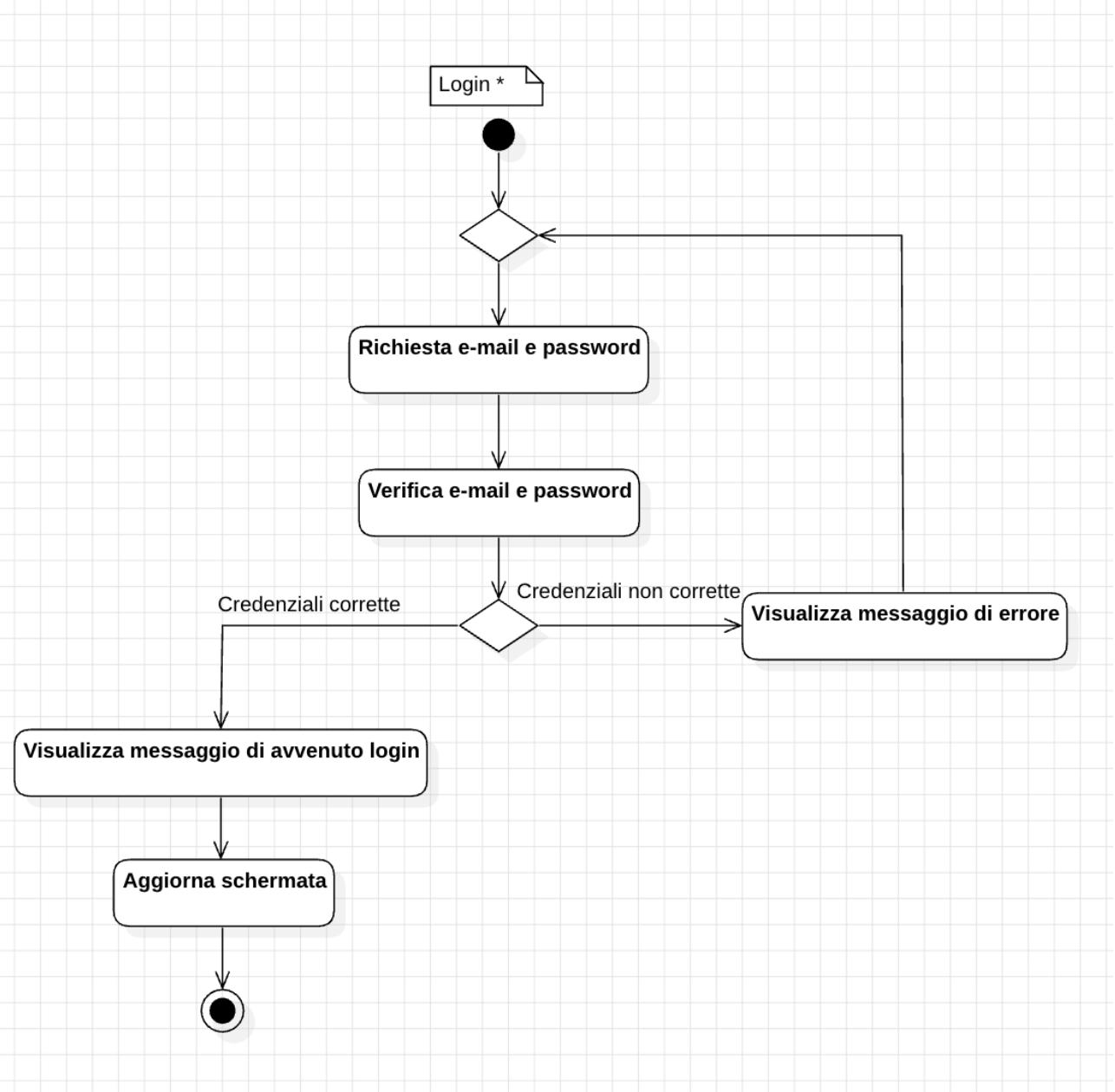
Il pattern Singleton è stato applicato anche ad altri componenti fondamentali dell'architettura, non limitandosi ai DAO, ma estendendosi alla gestione delle risorse condivise. In particolare, è stato utilizzato in:

- **DbmsConnector:** per garantire un unico punto di accesso alla connessione con il DBMS, centralizzando la configurazione e la gestione della connessione al database;

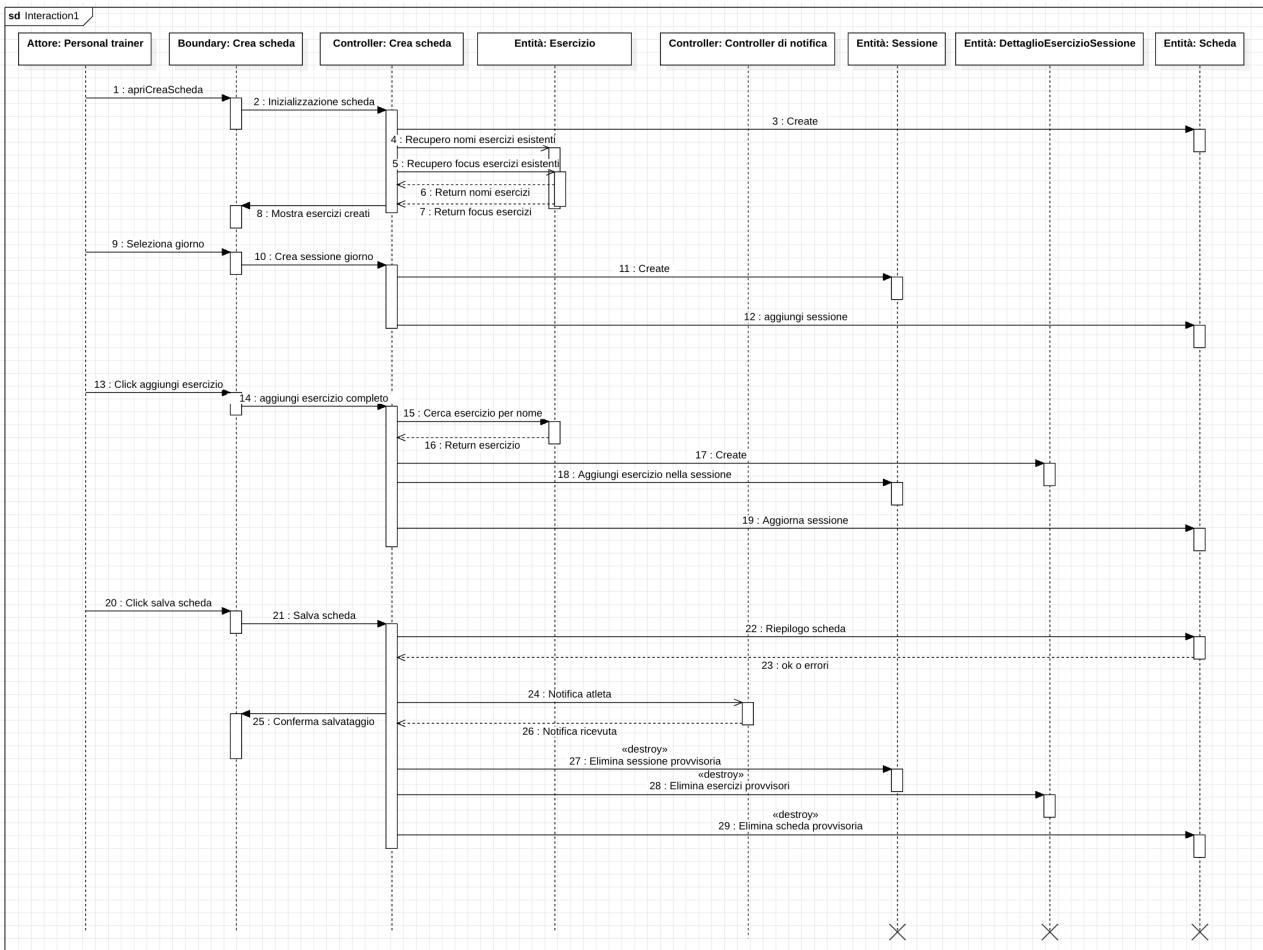
- **DemoMemory**: per condividere un'unica struttura di memoria in modalità demo tra tutti i DAO demo, assicurando coerenza dei dati e comportamento uniforme durante l'esecuzione.

### 3.4 Activity diagram

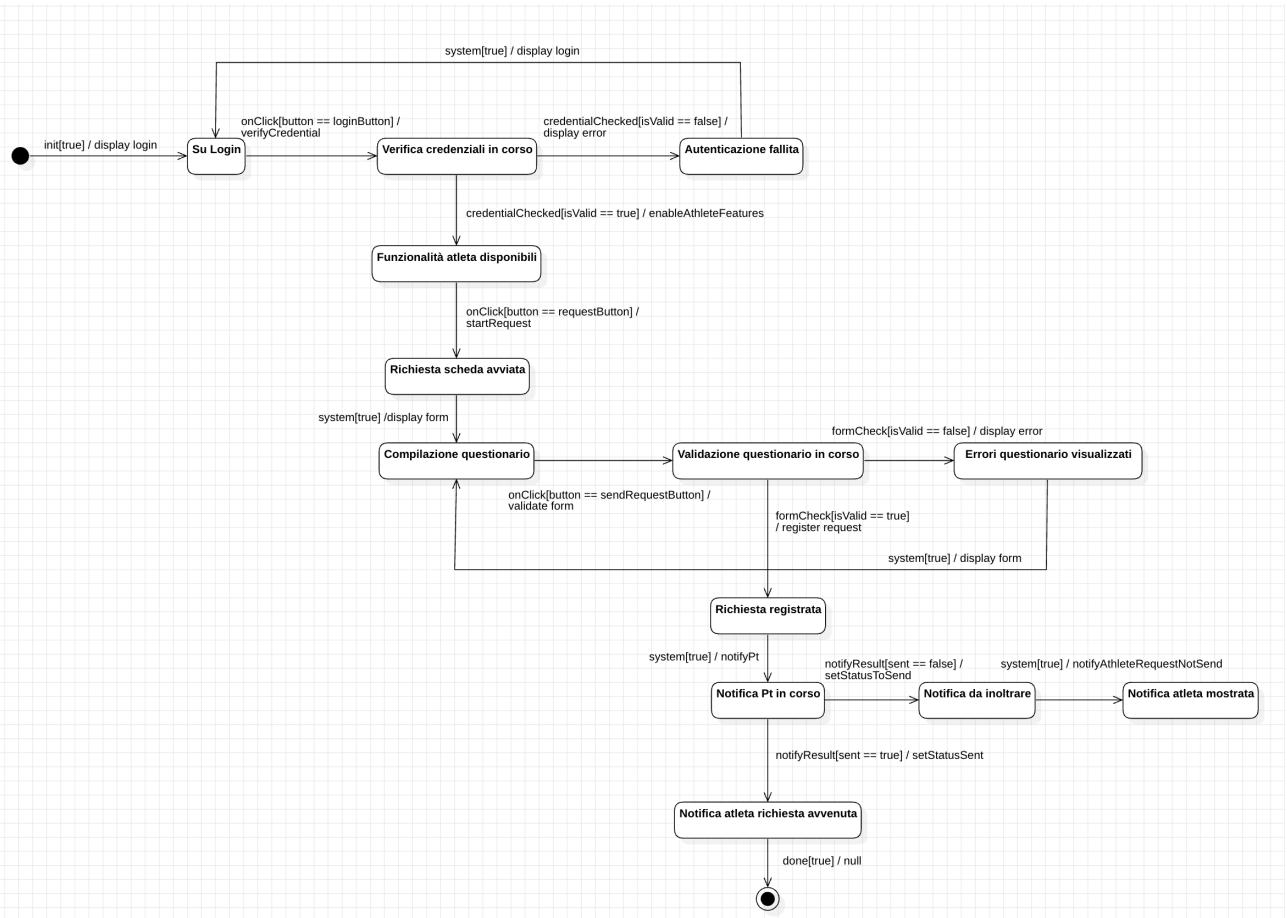




### 3.5 Sequence diagram



### 3.6 State diagram



## 4. Link utili

Sonar cloud: <https://sonarcloud.io/organizations/robertodimuro/projects>

GitHub: <https://github.com/RobertoDiMuro/Progetto.git>

Video YouTube: [https://youtu.be/d2\\_qEso2oSc](https://youtu.be/d2_qEso2oSc)