



leAft

leAft è l'app che ti **aiuta** ad allenarti in modo **semplice** e **personalizzato**.

Visualizza esercizi, set, ripetizioni e carico **ottimizzati** per **te**, in base alle tue **esigenze**.
Allenati meglio, raggiungi i tuoi **obiettivi** con leAft

Introduzione e obiettivi

L'applicativo sarà in grado di generare un allenamento per ogni necessità dell'utente in base alla sua esperienza.

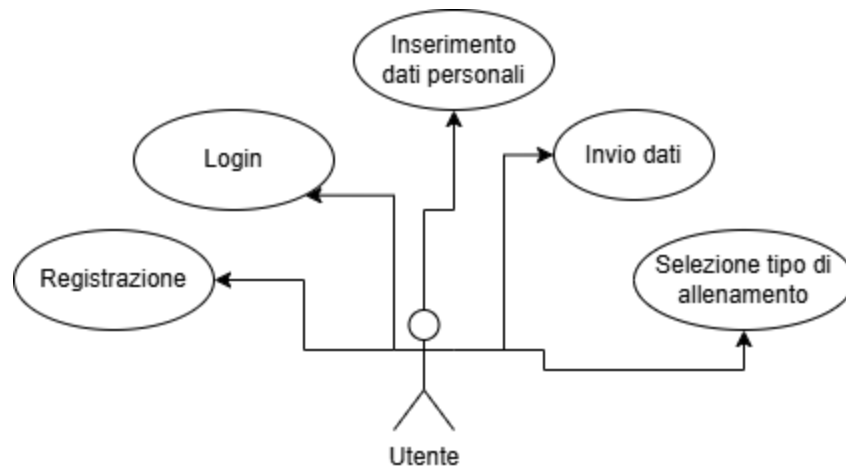
- L'utente sarà in grado di poter effettuare la registrazione all'applicativo e l'annesso accesso.
- L'utente effettuerà una fase di risposta a delle domande per avvicinarsi all'applicazione
- L'utente seguirà l'allenamento che viene fornito
-

Panoramica dei requisiti

Id	Requisito	Descrizione
F10	Accesso Utente	L'utente può accedere all'area personale

		dell'applicativo per la personalizzazione e il salvataggio dei dati
F20	Registrazione Utente	L'utente può creare un account qualora non ne avesse ancora uno a disposizione
F30	Profilazione Utente	L'utente inserirà qualche dato per la sua profilazione, l'obiettivo futuro sarà quello di garantire una maggiore personalizzazione degli esercizi e miglioramento dell'app
F40	Creazione allenamenti	L'utente avrà a disposizione 3 diverse metodologie di allenamento selezionabili singolarmente
F40.1	Beginner	Allenamento per principianti avrà lo scopo di poter introdurre l'utente nello sport e farlo sentire quanto più a suo agio e soddisfatto possibile
F40.2	Intermediate	Programma di allenamento realizzato in previsione per la maggior parte degli utenti che fruiranno dell'applicativo
F40.3	Expert	Circuito realizzato per coloro che hanno tanta esperienza alle spalle con il semplice obiettivo dell'alto mantenimento e dell'approfondimento

Use case:



Use case name	Registrazione	
Requisiti correlati		
Obiettivo nel contesto	Permettere agli utenti di registrarsi	
Pre Condizioni	L'utente deve avere accesso a un dispositivo connesso a Internet e un'email valida	
Condizione di successo	L'utente crea con successo un account e può accedere all'applicazione	
Condizione di fallimento	L'utente non riesce a creare un account a causa di un'email non valida o di una password non conforme	
Attori primari	Cliente	
Attori secondari	Sistema di autenticazione	
Trigger	L'utente seleziona l'opzione di registrazione nell'applicazione	
Flusso principale	Passo	Azione
	1	L'utente seleziona il pulsante registrati
	2	L'utente inserisce il nome
	3	L'utente inserisce il cognome
	4	L'utente inserisce nome utente
	5	L'utente inserisce password
	6	L'utente inserisce il Sesso
	7	L'utente inserisce l'età

		L'utente si registra dopo esito positivo ed accede all'applicativo
--	--	---

Obiettivi di qualità

Categoria qualità	Qualità	Descrizione
Sicurezza	Integrità	Deve essere garantito all'utente un accesso in modo tale che il suo account non possa essere preso e visualizzato da terzi
Prestazioni	Precisione (1)	Ogni componente deve funzionare in modo accurato per l'azione che deve svolgere
	Completezza	Gli algoritmi utilizzati devono essere il più possibile legalmente ineccepibili e comprensibili
	Robustezza	Il sistema deve funzionare in modo affidabile in tutte le condizioni ambientali e operative specificate
Usabilità	Facilità d'uso	Facilità d'uso per l'utente dal primo utilizzo alla fine dell'uso
	Facilità di apprendimento	Le funzioni standard devono essere il più possibile facili e intuitive da usare senza la necessità di lunghe istruzioni preliminari

Vincoli di architettura

Vincoli	Spiegazione
GitHub	Il progetto deve essere costantemente aggiornato su GitHub, mostrando tutte le collaborazioni dei membri
Layout xml	per il front end
Java	per il back end
SQLite	per il database

Vincolo	Spiegazione
Politiche di sicurezza dei dati	Garantisce la protezione dei dati sensibili degli utenti
GDPR	Dopo il rilascio del PoC il team legale dovrà controllare la conformità al GDPR.

Vincolo	Spiegazione
Linee guida di programmazione	Tutto il codice deve seguire le linee guida di programmazione stabilite per garantire la coerenza e la manutenibilità.
Convenzioni di controllo delle versioni	Tutte le modifiche devono essere committate seguendo il formato di commit standardizzato e rispettando il branching model stabilito.
Convenzioni di documentazione	La documentazione deve essere mantenuta aggiornata e seguire le convenzioni di stile e formato stabilite.

Strategia della soluzione

L' applicativo è stato sviluppato su Android Studio, sono state create tutte le classi e i file contenenti le viste che permettono all'utente di interagire con l'applicazione ed effettuare tutte le richieste.

E' stato inoltre implementato un database dove sono salvati tutte le informazioni dell'utente che vengono inserite in fase di registrazione.

Sono state quindi valutate tutte le possibili azioni che potesse effettuare l'utente per sviluppare una piattaforma all'apparenza il più facile e utilizzabile possibile per garantire un'esperienza utente eccellente.

Component

Colonna	Tipo	Attributi	Descrizione
id	INTEGER	PRIMARY KEY AUTOINCREMENT	Identificativo univoco dell'utente.
Username	TEXT	Attributi	Nome utente
Password	TEXT	NOT NULL	Password
name	TEXT		Nome dell'utente
surname	TEXT		Cognome dell'utente
gender	TEXT		Genere dell'utente
age	INTEGER		Età dell'utente

Caratteristiche del Database

Autenticazione:

- L'utente può effettuare l'accesso utilizzando username e password.
- La verifica viene effettuata attraverso query SQL che controllano le credenziali.

Gestione utenti:

- È possibile aggiungere nuovi utenti con dettagli personali.
- Ogni utente è identificato in modo univoco tramite l'id.

Recupero dati:

- La tabella users consente il recupero di tutti i dati salvati relativi agli utenti.

Manutenzione:

- In caso di aggiornamento dello schema, il metodo onUpgrade cancella la tabella precedente e ne crea una nuova.

Rischi e debiti tecnici

Rischi:

- **Sicurezza dei dati:** I dati degli utenti devono essere protetti per evitare perdite.
- **Prestazioni e scalabilità:** L'app potrebbe bloccarsi
- **Compatibilità:** L'app deve essere compatibile con diversi dispositivi Android
- **Manutenzione e aggiornamenti:** L'app deve ricevere aggiornamenti e manutenzione.

Debiti tecnici:

- **Architettura del software:** L'architettura del software deve essere ben progettata per facilitare la manutenzione e aggiornamento.
- **Gestione dei dati:** I dati degli utenti devono essere archiviati e gestiti in modo sicuro.
- **Documentazione:** L'app deve avere una sezione della documentazione per la facilitazione dell'utilizzo.