



# **Progetto di Ingegneria del Software 2025/26**

**Università Ca' Foscari Venezia**

## **Documento dei Requisiti**

**1.0**

**Eco Group**

**11/11/2025**



### Document Informations

NomeProgetto	Acronimo	
Deliverable	Documento dei Requisiti	
Data di Consegna	11/11/2025	
Team Leader	Enrico Sforza	902600@stud.unive.it
Team members	Enrico Rampazzo 901141@stud.unive.it Alberto Ferragosti 895936@stud.unive.it	

### Document History

Version	Issue Date	Stage	Changes	Contributors
1.0	11 / 11 / 2025	Draft	Aggiunti caso d'uso: profilo	902600
1.1	11 / 11 / 2025	Draft	Aggiunti casi d'uso: login	895936
1.2	11 / 11 / 2025	Draft	Aggiunto caso d'uso: quest	901141
2.0	25/01/2026	Final	Revisione pre consegna	895936, 902600



# Indice

<b>1. Introduzione</b>	<b>5</b>
1.1. Scopo del documento	5
1.2. Descrizione generale del sistema	5
1.3. Coinvolgimento degli stakeholder	5
<b>2. Modelli del sistema</b>	<b>6</b>
<b>2.1. Casi d'uso</b>	<b>6</b>
Profilo	6
<b>3. Definizione dei requisiti funzionali</b>	<b>13</b>
3.1. Requisiti funzionali principali	13
Profilo	14
Quest	14
Login	15
Registrazione	17
3.2. Possibili evoluzioni del sistema (requisiti aggiuntivi)	19
Profilo	19
Quest	20
Login	20
Registrazione	21
3.3. Priorità complessiva	22
Profilo	22
Quest	22
Login	22
Registrazione	22
<b>4. Definizione dei requisiti non funzionali</b>	<b>23</b>
4.1. Requisiti di prodotto	23
Profilo	23
Quest	24
Login	24
Registrazione	25
4.2. Requisiti di processo	26
Profilo	26
Quest	27
Login	27
Registrazione	27
4.3. Requisiti esterni	28
<b>5. Specifica dei requisiti funzionali</b>	<b>29</b>
Profilo	29
Quest	30
Login	31



Registrazione

33

## **6. Tabella di tracciabilità**

**36**



# 1. Introduzione

## 1.1. Scopo del documento

Il presente documento ha l'obiettivo di descrivere in modo dettagliato i requisiti funzionali e non funzionali dell'applicazione EcoApp, un sistema gamificato per la promozione di comportamenti sostenibili nelle abitudini quotidiane degli utenti.

## 1.2. Descrizione generale del sistema

EcoApp è un sistema **hardware + software** composto da:

- **Software:** applicazione mobile Android (versione minima 8+) sviluppata in Java con Android Studio.
- **Hardware:** smartphone dell'utente (sensori GPS, accelerometro e rete mobile).

L'app consente agli utenti di:

- Creare un profilo personale;
- Visualizzare i propri progressi ambientali (CO<sub>2</sub> evitata, punti ottenuti, ecc.);
- Partecipare a missioni;
- Interagire con altri utenti tramite una sezione "Amici".

## 1.3. Coinvolgimento degli stakeholder

Gli stakeholder principali sono:

- **Utenti finali:** cittadini interessati a migliorare le proprie pratiche sostenibili.
- **Team di sviluppo (EcoGroup):** responsabile di analisi, progettazione, implementazione e testing.
- **Docente del corso:** revisore tecnico e valutatore dei deliverable.

Durante la fase di analisi, gli stakeholder sono stati coinvolti tramite:

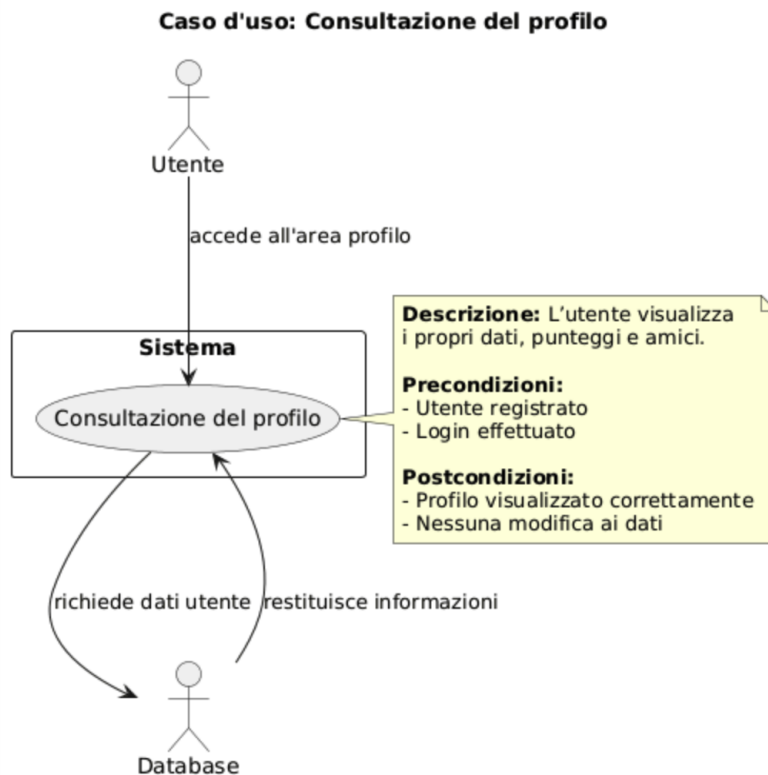
- feedback del docente;
- questionari;
- confronto interno nel team per la definizione delle priorità funzionali.



## 2. Modelli del sistema

### 2.1. Casi d'uso

#### Profilo



- **Attori e Descrizioni:**

- **Utente:** Persona registrata nell'applicazione che accede al proprio profilo, visualizza dati, punteggi e lista amici.
- **Database:** Componente software responsabile della memorizzazione e del recupero dei dati utente (anagrafici, punteggi, relazioni di amicizia, ecc.).

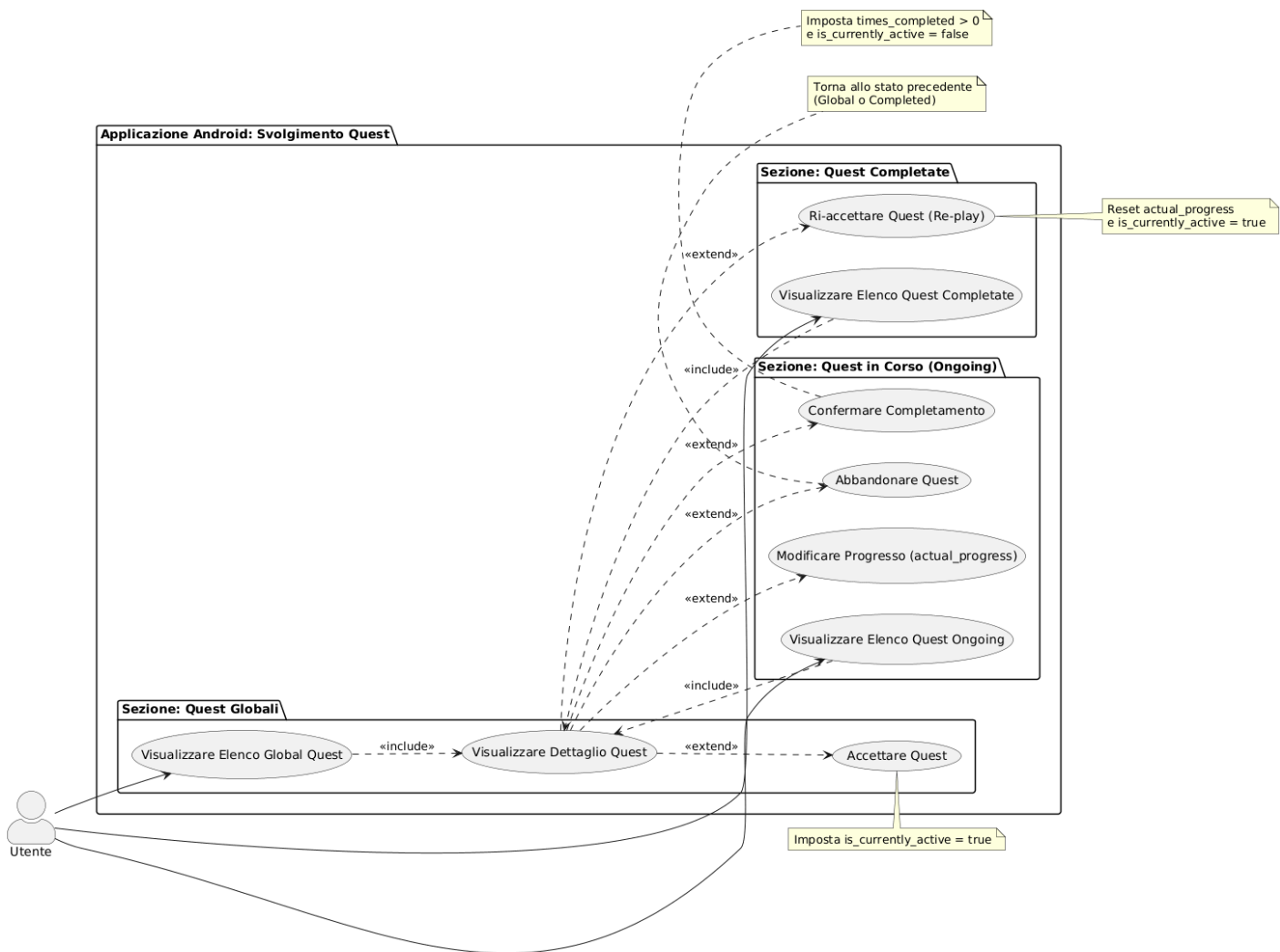
- **Caso d'uso: Consultazione del profilo**

- **Attori principali:** Utente, Database
- **Descrizione sintetica:** L'utente accede alla sezione "Profilo" per visualizzare le proprie informazioni personali, i punteggi accumulati e la lista dei propri amici.
- **Precondizioni:** L'utente deve essere già registrato ed autenticato nel sistema.
- **Trigger (evento scatenante):** L'utente seleziona l'icona o la voce di menu "Profilo" all'interno dell'applicazione.
- **Flusso principale degli eventi:**
  - L'utente accede alla sezione "Profilo".
  - Il sistema invia una richiesta al database per ottenere i dati associati all'account dell'utente.
  - Il database restituisce i dati richiesti (nome, email, punteggio, amici, missioni completate, ecc.).
  - Il sistema mostra i dati all'utente in un'interfaccia chiara e ordinata.



- L'utente può scorrere la schermata per visualizzare tutte le informazioni disponibili.
- **Flussi alternativi:**
  - 2a. In caso di errore di connessione al database, il sistema mostra un messaggio di errore ("Impossibile caricare i dati del profilo").
  - 4a. Se alcuni dati non sono disponibili, vengono mostrati con valori predefiniti o placeholder.
- **Postcondizioni:** Il profilo viene visualizzato correttamente, e l'utente può consultare i propri dati aggiornati.

## Quest





Nome del caso d'uso	Svolgimento di quest
Attori principali	Utente
Descrizione sintetica	L'utente accede alla sezione Quest dell'applicazione per visualizzare le quest disponibili, accettarle, monitorarne il progresso, completarle o riattivarle se già concluse.
Precondizioni	L'utente deve essere già registrato ed autenticato nel sistema.
Trigger (evento scatenante)	L'utente seleziona la voce di menu "Quest" all'interno dell'applicazione.
Flusso principale degli eventi	<ol style="list-style-type: none"><li>1. L'utente accede alla sezione "Quest".</li><li>2. Il sistema mostra le tre sezioni: Quest Globali, Quest in corso, Quest completate.</li></ol> <p>QUEST GLOBALI</p> <ol style="list-style-type: none"><li>3. L'utente visualizza l'elenco delle quest globali disponibili.</li><li>4. L'utente seleziona una quest.</li><li>5. Il sistema mostra i dettagli della quest (descrizione, step, punti).</li><li>6. L'utente accetta la quest.</li><li>7. La quest viene spostata tra le quest in corso.</li></ol> <p>QUEST IN CORSO</p> <ol style="list-style-type: none"><li>8. L'utente visualizza le quest attualmente attive.</li><li>9. L'utente seleziona una quest in corso.</li><li>10. L'utente aggiorna il progresso della quest.</li></ol>



11. Al raggiungimento del progresso massimo, l'utente conferma il completamento.

12. La quest viene segnata come completata e spostata tra le quest completate.

Flussi alternativi 6a. L'utente non accetta la quest → il sistema ritorna all'elenco delle quest globali.

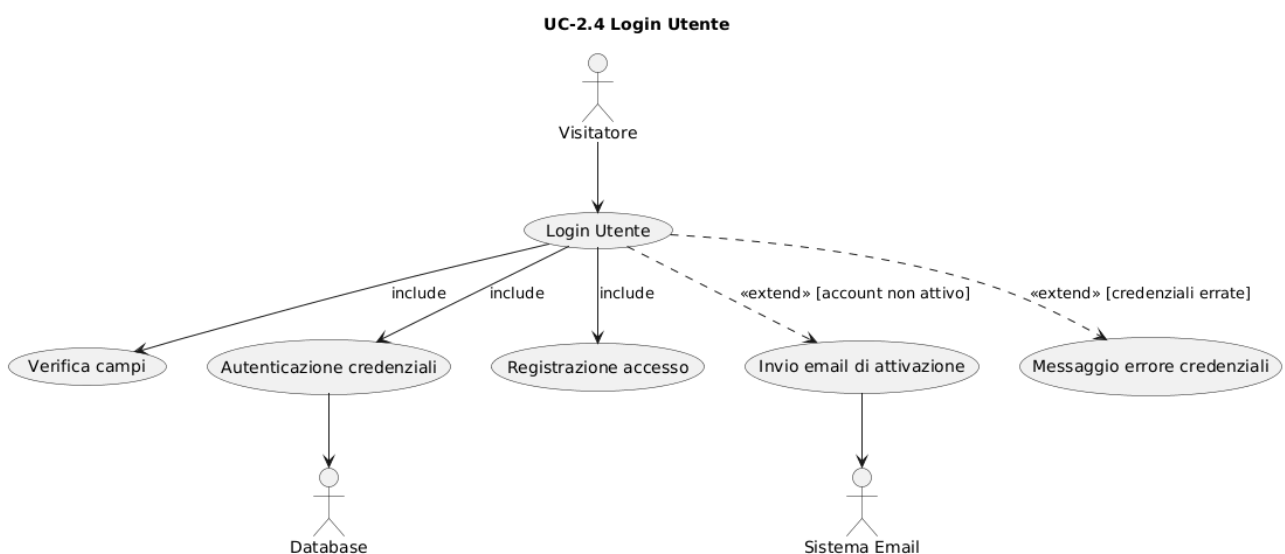
10a. L'utente abbandona la quest → la quest ritorna allo stato precedente (non attiva).

11a. L'utente non conferma il completamento → la quest rimane nello stato in corso. 12a. Dalla sezione Quest completate, l'utente riaccetta una quest → la quest viene nuovamente spostata tra le quest in corso.

Postcondizioni Lo stato delle quest viene aggiornato correttamente, e l'utente può visualizzare le quest svolte dal profilo.

## Nome del caso d'uso

Login Utente



## Attori principali



Visitatore

Database

Sistema Email

### **Descrizione sintetica**

Un visitatore si autentica nel sistema per accedere alle funzionalità riservate, diventando un utente autenticato.

### **Precondizioni**

Il visitatore deve avere un account registrato nel sistema.

Il database e il sistema email devono essere attivi e funzionanti.

### **Trigger (evento scatenante)**

Il visitatore seleziona la voce "Accedi" e inserisce le proprie credenziali.

### **Flusso principale degli eventi**

1. Il visitatore apre la schermata di login.
2. Inserisce l'indirizzo email e la password.
3. Il sistema verifica che i campi siano stati compilati correttamente.
4. Il sistema invia la richiesta di autenticazione al database.
5. Il database confronta le credenziali con quelle registrate.
6. Se la corrispondenza è positiva, il sistema autentica l'utente.
7. Il sistema aggiorna lo stato del visitatore in "Utente autenticato".
8. Il sistema registra data e ora dell'accesso.
9. L'utente accede alle funzionalità riservate.

### **Flussi alternativi**

3a. Campi mancanti: il sistema mostra un messaggio che richiede la compilazione obbligatoria.

5a. Credenziali errate: il sistema mostra un messaggio di errore e chiede di riprovare.

5b. Account non attivo: il sistema mostra un messaggio e invia un'email con le istruzioni per l'attivazione.

### **Postcondizioni**

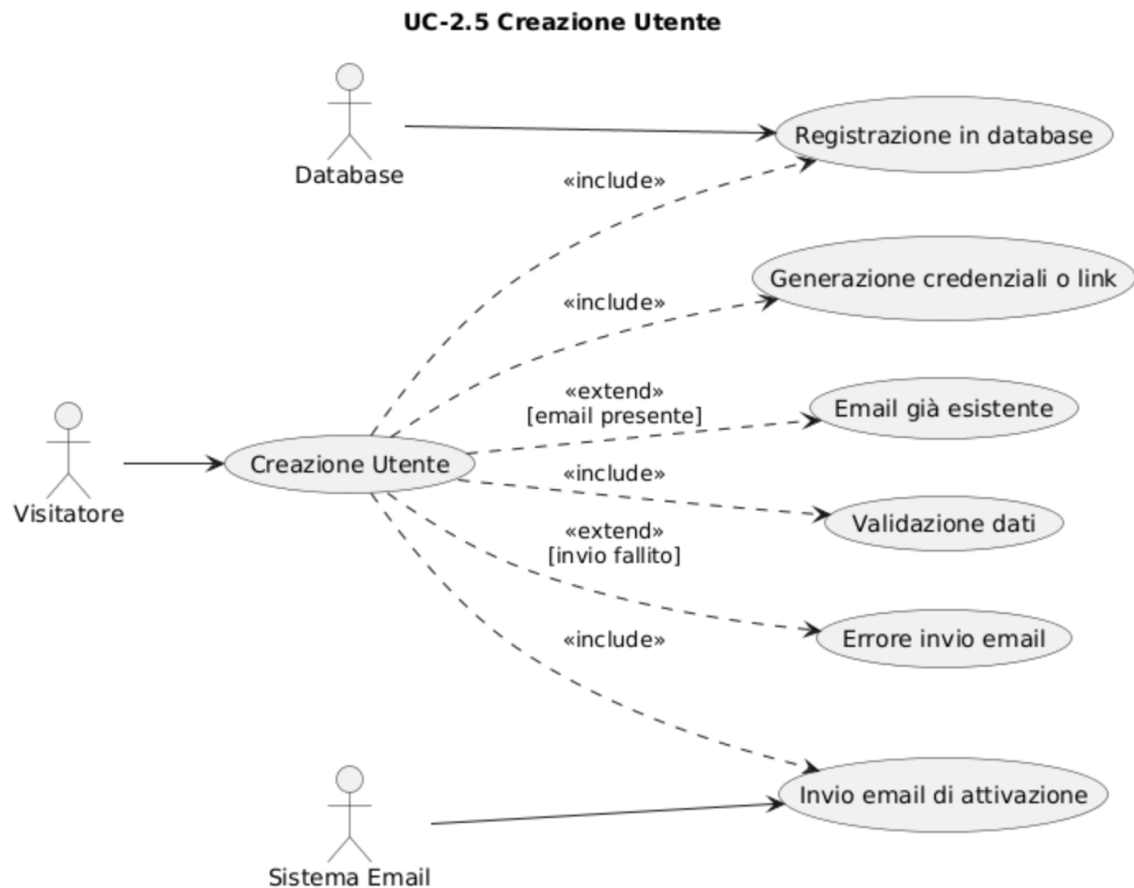
L'utente è autenticato e può accedere alle aree riservate del sistema.

Il sistema registra l'esito del tentativo di accesso.



## Nome del caso d'uso

Creazione Utente



## Attori principali

Visitatore

Database

Sistema Email



## **Descrizione sintetica**

Un visitatore/utente si registra autonomamente nel sistema inserendo i propri dati. Il sistema salva le informazioni nel database e invia un'email di conferma all'utente.

## **Precondizioni**

Il database e il sistema email devono essere attivi e funzionanti.

## **Trigger (evento scatenante)**

Un visitatore/utente seleziona la voce "Crea nuovo utente" nella schermata di accesso.

## **Flusso principale degli eventi**

1. Visitatore accede alla schermata di registrazione.
2. Visitatore seleziona l'opzione "crea account/registrati"
3. Visitatore Inserisce i dati richiesti (nome, cognome, email, psw, ecc.).
4. Il sistema valida i dati inseriti (campi obbligatori, formato, ecc.).
5. Il sistema verifica che l'email non sia già registrata
6. Il sistema registra il nuovo utente nel database.
7. Il database salva le informazioni del nuovo utente.
8. Il sistema notifica all'utente tramite mail che la creazione è avvenuta con successo.

## **Flussi alternativi**

4a. Dati non validi: il sistema evidenzia i campi errati e chiede la correzione.

5a. Email già esistente: il sistema mostra un messaggio di errore e interrompe il processo.

8a. Errore nell'invio dell'email: il sistema registra un avviso ma conferma comunque la creazione dell'utente.

## **Postcondizioni**

Il nuovo utente è registrato nel database.

L'utente riceve un'email di conferma di attivazione.

Il sistema aggiorna la lista degli utenti attivi.



### 3. Definizione dei requisiti funzionali

In questa sezione vengono descritti i servizi software che il sistema deve offrire all'utente per consentire la corretta fruizione dell'applicazione. I requisiti funzionali specificano cosa il sistema deve fare, senza entrare nel dettaglio tecnico dell'implementazione.

Tutti i requisiti sono stati classificati per priorità (Alta, Media, Bassa) e accompagnati da una breve motivazione.

#### 3.1. Requisiti funzionali principali

##### Profilo

###### RF1: Accesso al profilo

- **Descrizione:** L'utente deve poter accedere all'area "Profilo" dal menu principale dell'applicazione.
- **Priorità:** Alta
- **Motivazione:** Permette di avviare la consultazione dei propri dati.

###### RF2: Recupero dati utente

- **Descrizione:** Il sistema deve interrogare il database per ottenere le informazioni associate all'account (nome, email, punteggio, amici, missioni completate).
- **Priorità:** Alta
- **Motivazione:** Fondamentale per la visualizzazione del profilo.

###### RF3: Visualizzazione dati personali

- **Descrizione:** Il sistema deve mostrare all'utente i propri dati personali (nome, email, immagine del profilo).
- **Priorità:** Alta
- **Motivazione:** Core della funzionalità di profilo.

###### RF4: Visualizzazione punteggi

- **Descrizione:** Il sistema deve mostrare i punti accumulati dall'utente e le missioni completate.
- **Priorità:** Alta
- **Motivazione:** Rende visibile il progresso dell'utente.

###### RF5: Visualizzazione lista amici

- **Descrizione:** Il sistema deve mostrare la lista degli amici.
- **Priorità:** Media
- **Motivazione:** Favorisce l'aspetto sociale dell'app.

###### RF6: Aggiornamento dinamico dati



- **Descrizione:** Il profilo deve aggiornarsi automaticamente in caso di modifiche.
- **Priorità:** Media
- **Motivazione:** Garantisce la coerenza delle informazioni.

#### **RF8: Sicurezza e privacy**

- **Descrizione:** I dati personali devono essere recuperati e mostrati solo all'utente autenticato.
- **Priorità:** Alta

### **Quest**

#### **RF9: Assegnazione quest**

- **Descrizione:** L'utente sceglie quest legate a comportamenti ecologici.
- **Priorità:** Alta.
- **Motivazione:** È il cuore dell'app, guida l'attività.

#### **RF10: Tracciamento completamento**

- **Descrizione:** L'utente segna la quest come completata con un click.
- **Priorità:** Alta.
- **Motivazione:** Serve a misurare i progressi.

#### **RF11: Sistema punti/ricompense**

- **Descrizione:** Ogni quest completata assegna punti.
- **Priorità:** Alta.
- **Motivazione:** Serve a misurare i progressi.

#### **RF12: Rinnovo automatico**

- **Descrizione:** Le quest ripetibili si rigenerano allo scadere senza intervento manuale.
- **Priorità:** Alta.
- **Motivazione:** Riduce gestione e mantiene continuità.

#### **RF13: Contatore progressi**

- **Descrizione:** Conteggio delle volte in cui l'utente ripete la stessa quest(in caso di quest ripetibili).
- **Priorità:** Media.
- **Motivazione:** Evidenzia costanza e premia la regolarità.

#### **RF15: Storico e statistiche**

- **Descrizione:** L'utente visualizza quest passate e punteggio.
- **Priorità:** Bassa/Media.
- **Motivazione:** Feedback e motivazione a lungo termine.

### **Login**

#### **RF16: Schermata di accesso**



- **Descrizione:** Il sistema deve fornire una schermata di login con campi email e password e pulsante "Accedi".
- **Priorità:** Alta
- **Motivazione:** Punto di ingresso alla funzionalità di autenticazione.

#### **RF17: Validazione form lato client**

- **Descrizione:** Il sistema deve validare il formato dell'email e la presenza della password prima dell'invio.
- **Priorità:** Alta
- **Motivazione:** Riduce errori e chiamate inutili al backend.

#### **RF18: Autenticazione credenziali**

- **Descrizione:** Il sistema deve verificare email e password confrontandole con i dati nel database.
- **Priorità:** Alta
- **Motivazione:** Autenticazione corretta dell'utente.

#### **RF19: Account Sempre Attivo**

- **Descrizione:** Tutti gli account registrati nel sistema sono immediatamente attivi e utilizzabili dopo la registrazione, senza necessità di verifica email.
- **Priorità:** Media
- **Motivazione:** L'architettura implementata non prevede un sistema di verifica email per attivazione account. Questa scelta semplifica l'esperienza utente per un MVP, riducendo l'attrito nella fase di onboarding. La verifica email potrà essere implementata in versioni future.

#### **RF20: Blocco per credenziali errate**

- **Descrizione:** Dopo un numero configurabile di tentativi falliti, il sistema deve applicare un blocco temporaneo dell'account.
- **Priorità:** Media
- **Motivazione:** Mitiga attacchi di forza bruta.

#### **RF21: Creazione sessione sicura**

- **Descrizione:** In caso di esito positivo, il sistema deve creare una sessione sicura con scadenza e protezione da hijacking.
- **Priorità:** Alta
- **Motivazione:** Protegge l'accesso alle aree riservate.

#### **RF22: Remember me opzionale**

- **Descrizione:** Il sistema deve permettere il login persistente opzionale con token a scadenza estesa.
- **Priorità:** Media
- **Motivazione:** Migliora l'esperienza utente.

#### **RF23: Tracciamento Accessi Base**



- **Descrizione:** Il sistema deve registrare data, ora, esito e identificativo utente del tentativo di login.
- **Priorità:** Bassa
- **Motivazione:** Non è implementato un log dettagliato degli accessi con IP, device, geolocalizzazione. Il token JWT contiene intrinsecamente informazioni temporali sufficienti per tracciabilità base. Un sistema di audit log completo potrà essere considerato in evoluzioni future.

#### RF24: Messaggistica di errore chiara

- **Descrizione:** Il sistema deve mostrare messaggi specifici per campi mancanti, credenziali errate, account non attivo, problemi di rete.
- **Priorità:** Alta
- **Motivazione:** Guida efficace alla risoluzione dell'errore.

#### RF25: Logout e invalidazione sessione

- **Descrizione:** Il sistema deve permettere il logout e invalidare immediatamente la sessione e i token associati.
- **Priorità:** Alta
- **Motivazione:** Sicurezza e controllo dell'accesso.

#### RF26: Supporto Utente per problemi di accesso

- **Descrizione:** In caso di problemi di accesso (password dimenticata), l'utente può contattare il supporto. La funzionalità "Password dimenticata" con reset via email è prevista come evoluzione futura
- **Priorità:** Bassa
- **Motivazione:** Non esistendo verifica email, non esiste nemmeno il reinvio email di attivazione. Il requisito originale è stato trasformato in una nota per evoluzioni future, mantenendo tracciabilità della funzionalità prevista ma non implementata.

### Registrazione

#### RF27: Accesso schermata di registrazione

- **Descrizione:** Il sistema deve esporre una schermata di registrazione accessibile ai visitatori non autenticati tramite il link "Registrati" presente nella schermata di login
- **Priorità:** Alta
- **Motivazione:** L'architettura implementata prevede registrazione self-service senza intervento di amministratori. La schermata è accessibile pubblicamente per permettere a nuovi utenti di creare autonomamente il proprio account.

#### RF28: Avvio procedura di registrazione

- **Descrizione:** Il visitatore può accedere alla schermata di registrazione tramite il pulsante "Registrati" presente nella schermata di login
- **Priorità:** Alta



- **Motivazione:** Innesca il flusso principale.

#### **RF29: Inserimento dati utente**

- **Descrizione:** Il sistema deve richiedere i seguenti dati obbligatori per la registrazione: nome completo, indirizzo email valido, password (minimo 6 caratteri)
- **Priorità:** Alta
- **Motivazione:** Il modello User implementato in MongoDB prevede questi tre campi obbligatori. Il cognome è stato unificato nel campo "name", il ruolo non è necessario poiché tutti gli utenti hanno gli stessi permessi.

#### **RF30: Validazione dati lato server**

- **Descrizione:** Il sistema deve validare unicità email, formato campi e consistenza del ruolo prima del salvataggio.
- **Priorità:** Alta
- **Motivazione:** Integrità dei dati e correttezza logica.

#### **RF31: Registrazione nel database**

- **Descrizione:** In caso di validazione positiva, il sistema deve creare il record utente con stato "invitato" o "attivo" secondo policy.
- **Priorità:** Alta
- **Motivazione:** Persistenza dell'account.

#### **RF32: Scelta Password Utente**

- **Descrizione:** L'utente deve inserire la password desiderata durante la registrazione. La password deve rispettare i requisiti minimi di sicurezza (almeno 6 caratteri) e viene hashata con bcrypt prima del salvataggio
- **Priorità:** Alta
- **Motivazione:** L'implementazione prevede che l'utente scelga direttamente la propria password, senza generazione automatica di credenziali temporanee. Questo semplifica il flusso eliminando la necessità di un sistema di attivazione via email.

#### **RF33: Invio Email di Benvenuto**

- **Descrizione:** Al completamento della registrazione, il sistema invia un'email di benvenuto all'indirizzo fornito tramite il servizio Resend. L'email è informativa e non contiene link di attivazione
- **Priorità:** Media
- **Motivazione:** L'email inviata è di benvenuto, non di attivazione. L'account è già attivo al momento dell'invio. Il servizio Resend è configurato con API key in variabile d'ambiente per sicurezza.

#### **RF34: Gestione duplicati email**



- **Descrizione:** Se l'email è già presente, il sistema deve bloccare la creazione e segnalare il conflitto all'amministratore.
- **Priorità:** Alta
- **Motivazione:** Evitare inconsistenze e sovrapposizioni.

#### RF35: Conferma Registrazione all'utente

- **Descrizione:** Il sistema deve mostrare un messaggio di conferma all'utente al completamento della registrazione. L'utente accede.
- **Priorità:** Media
- **Motivazione:** Non esistendo un ruolo amministratore, la conferma viene mostrata direttamente all'utente che ha effettuato la registrazione. Viene inoltre inviata un'email di benvenuto tramite il servizio Resend.

#### RF36: Permessi Utente Standard

- **Descrizione:** Tutti gli utenti registrati nel sistema hanno lo stesso livello di permessi base (utente standard) con accesso a tutte le funzionalità dell'app
- **Priorità:** Bassa
- **Motivazione:** L'architettura attuale non prevede un sistema di ruoli differenziati. Tutti gli utenti possono: completare quest, gestire amicizie, visualizzare statistiche. Un sistema di ruoli/permessi potrà essere considerato in evoluzioni future.

#### RF37: Registrazione Data Creazione Account

- **Descrizione:** Il sistema deve registrare automaticamente la data e ora di creazione dell'account nel campo `createdAt` del documento utente in MongoDB
- **Priorità:** Bassa
- **Motivazione:** L'audit completo con tracciamento dell'amministratore creatore non è applicabile in un sistema self-service. Il timestamp di creazione è sufficiente per tracciabilità base ed è gestito automaticamente da Mongoose.

#### RF38: Resilienza invio email

- **Descrizione:** In caso di fallimento nell'invio dell'email di benvenuto, la registrazione viene comunque completata con successo. L'errore viene loggato nel server ma non blocca il flusso utente.
- **Priorità:** bassa
- **Motivazione:** L'email di benvenuto è una feature "nice-to-have", non critica. L'utente può usare l'app anche senza riceverla. Non è implementato un sistema di retry automatico per semplicità del MVP.



### 3.2. Possibili evoluzioni del sistema (requisiti aggiuntivi)

Nel corso dello sviluppo, il sistema potrebbe essere esteso con funzionalità aggiuntive come:

#### Profilo

##### RF39: Modifica dati profilo

- **Descrizione:** Permettere all'utente di aggiornare la propria immagine, nome o descrizione.
- **Motivazione:** Aumenta la personalizzazione.

##### RF40: Condivisione profilo

- **Descrizione:** Consentire all'utente di condividere i propri risultati tramite social o link diretto.
- **Motivazione:** Incentiva la partecipazione e la visibilità dell'app.

##### RF41: Statistiche avanzate

- **Descrizione:** Fornire grafici e indicatori sull'andamento settimanale/mensile dei punteggi.
- **Motivazione:** Supporta l'auto-valutazione e la motivazione.

#### Quest

##### RF42: Quest collaborative

- **Descrizione:** Squadre o gruppi completano quest insieme.
- **Priorità:** Media.
- **Motivazione:** Aumenta socialità e coinvolgimento.

##### RF43: Quest geolocalizzate

- **Descrizione:** Attività legate a punti specifici (es. parco vicino).
- **Priorità:** Media.
- **Motivazione:** Rende le quest più concrete e locali.

##### RF44: Personalizzazione avanzata

- **Descrizione:** L'app propone quest in base allo stile di vita e storico dell'utente.
- **Priorità:** Media.
- **Motivazione:** Migliora la rilevanza delle quest.

##### RF45: Verifica automatica

- **Descrizione:** L'app usa foto/scan e riconoscimento semplice per convalidare le quest.
- **Priorità:** Bassa/Media.
- **Motivazione:** Riduce spam e aumenta affidabilità.

##### RF46: Integrazione con partner

- **Descrizione:** Quest speciali sponsorizzate da aziende/associazioni ecologiche.
- **Priorità:** Bassa.



- **Motivazione:** Introduce premi reali e partnership.

## Login

### RF47: Autenticazione a due fattori

- **Descrizione:** Implementare un secondo livello di verifica tramite codice OTP inviato via email o app di autenticazione.
- **Motivazione:** Aumenta il livello di sicurezza e protegge da accessi non autorizzati.

### RF48: Accesso tramite social login

- **Descrizione:** Permettere l'autenticazione tramite provider esterni come Google, Microsoft o Apple.
- **Motivazione:** Migliora l'esperienza utente e riduce la necessità di gestire password.

### RF49: Accesso biometrico

- **Descrizione:** Integrare l'autenticazione tramite riconoscimento facciale o impronta digitale su dispositivi compatibili.
- **Motivazione:** Rende il login più rapido e sicuro.

### RF50: Monitoraggio accessi sospetti

- **Descrizione:** Il sistema deve segnalare tentativi di login da posizioni o dispositivi insoliti.
- **Motivazione:** Migliora la sicurezza e permette azioni preventive contro accessi fraudolenti.

### RF51: Modalità offline con sincronizzazione successiva

- **Descrizione:** Consentire il login temporaneo in modalità offline, con sincronizzazione al ripristino della connessione.
- **Motivazione:** Garantisce continuità d'uso in contesti con connettività limitata.

## Registrazione

### RF52: Importazione massiva utenti

- **Descrizione:** Permettere all'amministratore di caricare più utenti contemporaneamente tramite file CSV o integrazione API.
- **Motivazione:** Riduce i tempi di inserimento e migliora l'efficienza operativa.

### RF53: Inviti con registrazione autonoma

- **Descrizione:** Consentire all'amministratore di inviare link di invito che permettano all'utente di completare autonomamente la registrazione.
- **Motivazione:** Snellisce il flusso di creazione e responsabilizza l'utente finale.

### RF54: Gestione avanzata dei ruoli



- **Descrizione:** Consentire la definizione di ruoli personalizzati con permessi granulari su moduli e funzionalità.
- **Motivazione:** Aumenta la flessibilità e l'adattabilità a diverse strutture organizzative.

#### **RF55: Integrazione con sistemi HR o CRM**

- **Descrizione:** Collegare la creazione utenti con sistemi esterni (es. HRM, CRM) per sincronizzare dati anagrafici e ruoli.
- **Motivazione:** Garantisce coerenza tra i sistemi aziendali e riduce duplicazioni di dati.

#### **RF56: Scadenza automatica account**

- **Descrizione:** Introdurre la possibilità di definire una data di scadenza per gli account temporanei, con disattivazione automatica.
- **Motivazione:** Migliora la sicurezza e il controllo sugli accessi nel tempo.

### **3.3. Priorità complessiva**

#### **Profilo**

- I requisiti **RF1–RF4, RF8** sono classificati come core e devono essere implementati nella prima versione del sistema.
- I requisiti **RF5–RF7** sono considerati secondari ma necessari per la piena usabilità del sistema.
- I requisiti **RF39–RF41** rappresentano possibili estensioni future.

#### **Quest**

- I requisiti **RF9–RF12** sono classificati come core e devono essere implementati nella prima versione del sistema.
- I requisiti **RF13–RF15** sono considerati secondari ma necessari per la piena usabilità del sistema.
- I requisiti **RF42–RF46** rappresentano possibili estensioni future.

#### **Login**

- I requisiti **RF16–RF18, RF20–RF22, RF24–RF25** sono classificati come core e devono essere implementati nella prima versione del sistema.
- Nota: **RF19** (account sempre attivo) è implicito nell'architettura e non richiede implementazione specifica
- I requisiti **RF23** (tracciamento accessi base) sono considerati secondari ma necessari per la piena usabilità del sistema.
- I requisiti **RF26** (recupero password), **RF47–RF51** rappresentano possibili estensioni future.

#### **Registrazione**

- I requisiti **RF27–RF32, RF34–RF36** sono classificati come core e devono essere implementati nella prima versione del sistema.



- Nota: **RF27-RF29, RF32, RF35-RF36** sono stati formulati per architettura self-service (visitatore si registra autonomamente)
- I requisiti **RF33** (email benvenuto), **RF36-RF37** (tracciabilità, resilienza email) sono considerati secondari ma necessari per la piena usabilità del sistema.
- I requisiti RF52-RF56 rappresentano possibili estensioni future.

## 4. Definizione dei requisiti non funzionali

In questa sezione vengono definiti i Requisiti Non Funzionali (RNF). Questi requisiti non descrivono *cosa* il sistema deve fare (già coperto dai requisiti funzionali), ma *come* deve farlo.

Specificano i vincoli e le qualità del sistema, come le prestazioni, l'usabilità, la sicurezza e l'affidabilità, che sono cruciali per garantire un'esperienza utente positiva e un sistema robusto.

### 4.1. Requisiti di prodotto

#### Profilo

##### RNF1: Usabilità

- **Descrizione:** l'interfaccia deve essere semplice, chiara e coerente con il resto dell'applicazione.
- **Priorità:** Alta
- **Motivazione:** migliora l'esperienza utente e riduce gli errori di navigazione.
- **Metrica:** percentuale di utenti che completano la consultazione senza assistenza.
- **Valore soglia:**  $\geq 95\%$ .

##### RNF2: Prestazioni

- **Descrizione:** il sistema deve caricare le schermate in tempi rapidi anche con connessioni di rete moderate.
- **Priorità:** Alta
- **Motivazione:** garantisce fluidità e reattività dell'applicazione.
- **Metrica:** tempo medio di caricamento.
- **Valore soglia:**  $\leq 2$  secondi.

##### RNF3: Affidabilità

- **Descrizione:** il modulo profilo deve essere costantemente disponibile e funzionante, anche con più accessi simultanei.
- **Priorità:** Alta
- **Motivazione:** evita interruzioni e migliora la continuità del servizio.
- **Metrica:** percentuale di uptime del servizio.



- **Valore soglia:**  $\geq 99\%$ .

#### RNF4: Sicurezza

- **Descrizione:** i dati personali devono essere accessibili solo all'utente autenticato, con connessione cifrata e controllo delle sessioni.
- **Priorità:** Alta
- **Motivazione:** protegge la privacy e previene accessi non autorizzati.
- **Metrica:** numero di violazioni registrate.
- **Valore soglia:** 0.

#### Quest

##### RNF5: Affidabilità tracciamento

- **Descrizione:** Il completamento di una quest non deve andare perso.
- **Priorità:** Alta.
- **Motivazione:** Mantiene la fiducia dell'utente.
- **Metrica:** Percentuale errori di salvataggio.
- **Valore soglia:**  $< 0.1\%$ .

##### RNF6: Usabilità interfaccia quest

- **Descrizione:** Le quest devono essere leggibili e intuitive.
- **Priorità:** Media.
- **Motivazione:** Aumenta engagement.
- **Metrica:** Tasso di completamento spontaneo.
- **Valore soglia:**  $> 80\%$ .

#### Login

##### RNF8: Usabilità

- **Descrizione:** la schermata di login deve essere chiara, intuitiva e coerente con lo stile generale dell'applicazione.
- **Priorità:** Alta
- **Motivazione:** facilita l'accesso e riduce gli errori di inserimento credenziali.
- **Metrica:** percentuale di utenti che completano correttamente il login al primo tentativo.
- **Valore soglia:**  $\geq 95\%$ .

##### RNF9: Prestazioni autenticazione

- **Descrizione:** il tempo di risposta del sistema tra l'invio delle credenziali e l'esito del login deve essere minimo.
- **Priorità:** Alta
- **Motivazione:** migliora la percezione di efficienza e reattività del sistema.



- **Metrica:** tempo medio di risposta del server durante l'autenticazione.
- **Valore soglia:**  $\leq 2$  secondi.

#### RNF10: Affidabilità accesso

- **Descrizione:** la funzione di login deve essere disponibile e funzionante anche in condizioni di carico elevato.
- **Priorità:** Alta
- **Motivazione:** garantisce la continuità del servizio e l'accessibilità per tutti gli utenti.
- **Metrica:** percentuale di uptime del modulo login.
- **Valore soglia:**  $\geq 99\%$ .

#### RNF11: Sicurezza delle credenziali

- **Descrizione:** le credenziali devono essere trasmesse solo tramite connessioni cifrate (HTTPS) e mai memorizzate in chiaro.
- **Priorità:** Alta
- **Motivazione:** previene il furto di dati e garantisce la protezione delle informazioni sensibili.
- **Metrica:** numero di violazioni o vulnerabilità rilevate.
- **Valore soglia:** 0.

#### RNF12: Gestione errori e feedback

- **Descrizione:** il sistema deve fornire messaggi di errore chiari e non ambigui (senza indicare quale campo è errato).
- **Priorità:** Media
- **Motivazione:** migliora l'esperienza utente e mantiene la sicurezza del processo di autenticazione.
- **Metrica:** percentuale di messaggi di errore conformi agli standard di sicurezza.
- **Valore soglia:** 100%.

### Registrazione

#### RNF13: Usabilità form amministratore

- **Descrizione:** Il form di registrazione deve permettere il completamento in un tempo medio  $\leq 1$  minuto, con campi chiari (nome, email, password) e feedback immediato sulla validazione.
- **Priorità:** Media
- **Motivazione:** riduce errori di compilazione e velocizza il processo di registrazione.
- **Metrica:** tempo medio di completamento del form di creazione.
- **Valore soglia:**  $\leq 1$  minuto.

#### RNF14: Prestazioni registrazione

- **Descrizione:** il salvataggio di un nuovo utente nel database deve completarsi in tempi rapidi.
- **Priorità:** Alta
- **Motivazione:** garantisce fluidità e efficienza nell'amministrazione del sistema.
- **Metrica:** tempo medio di completamento della registrazione.
- **Valore soglia:**  $\leq 3$  secondi.



#### RNF15: Affidabilità invio email

- **Descrizione:** il sistema di invio automatico delle email di attivazione deve garantire la consegna e il tracciamento dell'esito.
- **Priorità:** Alta
- **Motivazione:** l'email è essenziale per completare l'attivazione dell'utente.
- **Metrica:** percentuale di email consegnate con successo.
- **Valore soglia:**  $\geq 98\%$ .

#### RNF16: Sicurezza dati registrazione

- **Descrizione:** i dati inseriti durante la creazione utente devono essere validati, cifrati e memorizzati in modo sicuro.
- **Priorità:** Alta
- **Motivazione:** protegge i dati personali e garantisce la conformità alle norme di sicurezza informatica.
- **Metrica:** numero di incidenti di data breach.
- **Valore soglia:** 0.

#### RNF17: Tracciabilità creazione account

- **Descrizione:** Il sistema deve registrare automaticamente la data/ora di creazione di ogni account nel campo `createdAt` del documento MongoDB.
- **Priorità:** Media
- **Motivazione:** Non essendoci un sistema admin, non è necessario tracciare "chi" ha creato l'utente (è sempre l'utente stesso). Il timestamp di creazione gestito da Mongoose è sufficiente per audit base.
- **Metrica:** percentuale di operazioni tracciate con successo.
- **Valore soglia:** 100%.

## 4.2. Requisiti di processo

### Profilo

#### RNF18: Manutenibilità

- **Descrizione:** il codice deve essere facilmente aggiornabile, documentato e coperto da test automatici.
- **Priorità:** Media
- **Motivazione:** facilita la gestione evolutiva del sistema.
- **Metrica:** percentuale di copertura dei test.
- **Valore soglia:**  $\geq 80\%$ .

#### RNF19: Documentazione del codice



- **Descrizione:** tutte le funzionalità relative alla consultazione del profilo devono essere descritte in modo chiaro nella documentazione tecnica.
- **Priorità:** Media
- **Motivazione:** agevola la manutenzione e la collaborazione tra sviluppatori.
- **Metrica:** copertura della documentazione.
- **Valore soglia:** 100%.

## Quest

### RNF20: Coerenza aggiornamenti quest

- **Descrizione:** Aggiornamenti regolari.
- **Priorità:** Alta.
- **Motivazione:** Routine.
- **Metrica:** Rispetto pianificazione.
- **Valore soglia:** > 90%.

## Login

### RNF22: Manutenibilità modulo autenticazione

- **Descrizione:** il codice relativo al login deve essere strutturato in moduli riutilizzabili, commentato e coperto da test automatici.
- **Priorità:** Media
- **Motivazione:** facilita gli aggiornamenti e la gestione di bug.
- **Metrica:** percentuale di copertura dei test.
- **Valore soglia:**  $\geq 85\%$ .

### RNF23: Monitoraggio e logging

- **Descrizione:** il sistema deve generare log di tentativi di accesso, errori e anomalie per consentire analisi successive.
- **Priorità:** Media
- **Motivazione:** supporta la diagnosi dei problemi e il monitoraggio della sicurezza.
- **Metrica:** percentuale di eventi registrati correttamente.
- **Valore soglia:** 100%.

## Registrazione

### RNF24: Documentazione Tecnica Registrazione

- **Descrizione:** Le funzionalità di registrazione self-service devono essere documentate nel manuale tecnico, includendo: endpoint API, validazioni, flusso di creazione account, gestione errori
- **Priorità:** Bassa
- **Motivazione:** Agevola la manutenzione del codice e l'onboarding di nuovi sviluppatori nel team



- **Metrica:** Copertura della documentazione tecnica
- **Valore soglia:** 100% degli endpoint documentati

#### **RNF25: Scalabilità del Sistema di Registrazione**

- **Descrizione:** Il sistema deve poter gestire un numero crescente di utenti registrati senza degrado delle prestazioni, sfruttando MongoDB Atlas come database cloud scalabile
- **Priorità:** Media
- **Motivazione:** Garantisce la crescita del sistema. MongoDB Atlas offre scaling automatico e il backend su Render.com può essere scalato orizzontalmente
- **Metrica:** Numero massimo di utenti gestibili con tempi di risposta stabili ( $\leq 2$  secondi per operazioni di registrazione/login)
- **Valore soglia:**  $\geq 10.000$  utenti

### **4.3. Requisiti esterni**

#### **RNF26: Compatibilità**

- **Descrizione:** il sistema deve funzionare correttamente sui dispositivi Android 8+
- **Priorità:** Media
- **Motivazione:** garantisce l'accesso al profilo a tutti gli utenti.
- **Metrica:** percentuale di compatibilità testata.
- **Valore soglia:**  $\geq 95\%$ .

#### **RNF27: Conformità GDPR**

- **Descrizione:** tutti i dati personali visualizzati devono essere trattati nel rispetto della normativa sulla privacy (GDPR).
- **Priorità:** Alta
- **Motivazione:** obbligatorio per la protezione dei dati e la conformità legale.
- **Metrica:** rispetto delle linee guida GDPR.
- **Valore soglia:** compliance totale.

#### **RNF28: Scalabilità**

- **Descrizione:** il sistema deve mantenere prestazioni stabili anche in caso di un numero elevato di utenti simultanei.
- **Priorità:** Media
- **Motivazione:** garantisce continuità di servizio con l'aumento del traffico.
- **Metrica:** numero massimo di utenti simultanei supportati.
- **Valore soglia:**  $\geq 1000$  utenti.



## 5. Specifica dei requisiti funzionali

### Profilo

#### RF1 – Accesso al profilo

- **Descrizione:** l'utente accede all'area *Profilo* dal menu principale.
- **Input:** selezione dell'icona o voce "Profilo".
- **Output:** apertura schermata con dati personali.
- **Pre-condizione:** utente autenticato.
- **Post-condizione:** profilo visualizzato correttamente.
- **Effetti collaterali:** nessuno.

#### RF2 – Recupero dati utente

- **Descrizione:** il sistema interroga il database per ottenere i dati associati all'account.
- **Input:** richiesta automatica al database.
- **Output:** dati restituiti (nome, email, punteggio, amici, missioni completate).
- **Pre-condizione:** utente autenticato.
- **Post-condizione:** dati caricati nella schermata profilo.
- **Effetti collaterali:** possibile latenza di rete.

#### RF3 – Visualizzazione dati personali

- **Descrizione:** mostra nome, email e immagine del profilo.
- **Input:** dati utente dal database.
- **Output:** interfaccia popolata con informazioni personali.
- **Pre-condizione:** dati disponibili.
- **Post-condizione:** informazioni correttamente visualizzate.
- **Effetti collaterali:** nessuno.

#### RF4 – Visualizzazione punteggi

- **Descrizione:** mostra punteggio totale e missioni completate.
- **Input:** dati punteggio dal database.
- **Output:** tabella punteggi visibile all'utente.
- **Pre-condizione:** dati aggiornati disponibili.
- **Post-condizione:** punteggi corretti in visualizzazione.
- **Effetti collaterali:** aggiornamento dinamico.

#### RF5 – Visualizzazione lista amici

- **Descrizione:** mostra elenco amici e relativi punteggi.
- **Input:** richiesta elenco dal database.
- **Output:** lista ordinata visualizzata.
- **Pre-condizione:** connessione disponibile.



- **Post-condizione:** amici visibili.
- **Effetti collaterali:** uso di rete.

#### RF6 – Aggiornamento dinamico dati

- **Descrizione:** il profilo si aggiorna automaticamente a ogni modifica.
- **Input:** trigger interno di variazione dati.
- **Output:** schermata aggiornata.
- **Pre-condizione:** connessione attiva.
- **Post-condizione:** dati sincronizzati.
- **Effetti collaterali:** leggero consumo di rete.

#### RF8 – Sicurezza e privacy

- **Descrizione:** mostra i dati solo all'utente autenticato tramite connessione cifrata.
- **Input:** richiesta autenticata.
- **Output:** visualizzazione sicura.
- **Pre-condizione:** sessione valida.
- **Post-condizione:** nessun accesso non autorizzato.
- **Effetti collaterali:** gestione token o sessione.

### Quest

#### RF9: Assegnazione quest

- **Descrizione:** L'utente sceglie quest ecologiche.
- **Input:** Selezione da lista.
- **Output:** Quest assegnata all'utente.
- **Pre-condizione:** L'utente è loggato.
- **Post-condizione:** Quest aggiunta alle quest attive.
- **Effetti collaterali:** Nessuno.

#### RF10: Tracciamento completamento

- **Descrizione:** L'utente segna la quest come completata.
- **Input:** Click o upload prova.
- **Output:** Stato quest aggiornato.
- **Pre-condizione:** Quest assegnata e attiva.
- **Post-condizione:** Quest marcata come completata.
- **Effetti collaterali:** Aggiornamento statistica interna.

#### RF11: Sistema punti/ricompense

- **Descrizione:** Ogni quest completata assegna punti.
- **Input:** Evento completamento quest.
- **Output:** Punti aggiunti.



- **Pre-condizione:** Quest completata.
- **Post-condizione:** Punteggio utente aumentato.
- **Effetti collaterali:** Aggiornamento classifica/ricompense.

#### **RF12: Rinnovo automatico**

- **Descrizione:** Le quest ripetibili si rigenerano alla scadenza.
- **Input:** Niente.
- **Output:** Quest rigenerate.
- **Pre-condizione:** Timer/periodo scaduto.
- **Post-condizione:** Nuove quest disponibili.
- **Effetti collaterali:** Nessuno.

#### **RF13: Contatore progressi**

- **Descrizione:** Conteggio ripetizioni della stessa quest.
- **Input:** Evento completamento.
- **Output:** Contatore aggiornato.
- **Pre-condizione:** Quest ripetibile attiva.
- **Post-condizione:** Contatore incrementato.
- **Effetti collaterali:** Nessuno.

#### **RF15: Storico e statistiche**

- **Descrizione:** Visualizzazione storico quest.
- **Input:** Richiesta pagina storico.
- **Output:** Dati statistici mostrati.
- **Pre-condizione:** Utente loggato.
- **Post-condizione:** Storico visualizzato.
- **Effetti collaterali:** Nessuno.

### **Login**

#### **RF16 – Apertura schermata di login**

- **Descrizione:** il sistema mostra la schermata di autenticazione con campi email e password e pulsante “Accedi”.
- **Input:** richiesta di accesso alla sezione “Login”.
- **Output:** interfaccia di login visibile.
- **Pre-condizione:** utente non autenticato.
- **Post-condizione:** modulo di autenticazione disponibile per l’inserimento dati.
- **Effetti collaterali:** nessuno.

#### **RF17 – Inserimento credenziali**

- **Descrizione:** l’utente inserisce email e password nei campi previsti.
- **Input:** digitazione dei dati nei campi email e password.
- **Output:** valori caricati nei campi di input.



- **Pre-condizione:** schermata di login aperta.
- **Post-condizione:** dati pronti per la validazione.
- **Effetti collaterali:** nessuno.

#### RF18 – Validazione dati

- **Descrizione:** il sistema verifica che entrambi i campi siano compilati e che l'email sia in formato corretto.
- **Input:** dati digitati dall'utente.
- **Output:** messaggio di errore o conferma di validazione.
- **Pre-condizione:** campi compilati.
- **Post-condizione:** dati validati o richiesta di correzione.
- **Effetti collaterali:** possibile messaggio di alert.

#### RF19 – Autenticazione utente

- **Descrizione:** il sistema confronta le credenziali con i dati presenti nel database.
- **Input:** richiesta di verifica con email e password.
- **Output:** risposta di autenticazione (successo o errore).
- **Pre-condizione:** connessione attiva e database raggiungibile.
- **Post-condizione:** utente autenticato o segnalazione di errore.
- **Effetti collaterali:** accesso registrato nei log.

#### RF20 – Gestione errori di accesso

- **Descrizione:** In caso di credenziali errate, il sistema mostra un messaggio di errore generico. Dopo 5 tentativi falliti consecutivi, l'account viene temporaneamente bloccato per 15 minuti (rate limiting).
- **Input:** Esito negativo del controllo credenziali.
- **Output:** Messaggio "Credenziali non valide" o "Troppi tentativi, riprova tra X minuti".
- **Pre-condizione:** Autenticazione fallita.
- **Post-condizione:** Utente informato; eventuale blocco temporaneo attivato.
- **Effetti collaterali:** Contatore tentativi incrementato in memoria server.
- **Priorità:** Alta
- **Motivazione:** Non esistono "account non attivi" poiché non c'è verifica email. Il rate limiting protegge da attacchi brute force.

#### RF21 – Creazione sessione sicura

- **Descrizione:** in caso di accesso riuscito, il sistema genera una sessione utente con token sicuro e scadenza configurata.
- **Input:** esito positivo del login.
- **Output:** sessione attiva e token associato.
- **Pre-condizione:** autenticazione avvenuta con successo.
- **Post-condizione:** utente autenticato con sessione attiva.
- **Effetti collaterali:** memorizzazione token temporaneo.



## RF22 – Tracciamento Accessi Base

- **Descrizione:** Il sistema traccia implicitamente gli accessi attraverso i token JWT che contengono timestamp di creazione (iat) e scadenza (exp).
- **Input:** Evento di login completato.
- **Output:** Token JWT con metadati temporali.
- **Pre-condizione:** Autenticazione positiva.
- **Post-condizione:** Token emesso con informazioni temporali.
- **Effetti collaterali:** Nessun log persistente dettagliato.
- **Priorità:** Bassa
- **Motivazione:** Un sistema di audit log completo non è implementato nel MVP. Il JWT fornisce tracciabilità base sufficiente.
- 

## RF23 – Logout e chiusura sessione

- **Descrizione:** il sistema deve consentire all'utente di terminare la sessione e invalidare il token di accesso.
- **Input:** comando "Logout".
- **Output:** chiusura sessione e ritorno allo stato "Visitatore".
- **Pre-condizione:** sessione attiva.
- **Post-condizione:** utente disconnesso.
- **Effetti collaterali:** cancellazione cookie/sessione.

## Registrazione

### RF24: Accesso Schermata Registrazione

- **Descrizione:** Il visitatore non autenticato accede alla schermata di registrazione tramite il link "Registrati" presente nella schermata di login.
- **Input:** Click su link/pulsante "Registrati" in LoginActivity.
- **Output:** Apertura RegisterActivity con form di registrazione.
- **Pre-condizione:** Utente non autenticato.
- **Post-condizione:** Form di registrazione visualizzato.
- **Effetti collaterali:** Nessuno.
- **Priorità:** Alta
- **Motivazione:** L'architettura prevede registrazione self-service, non amministrativa.

### RF25: Visualizzazione Form Registrazione

- **Descrizione:** Il sistema mostra un form con i campi: nome completo, email, password, conferma password.
- **Input:** Apertura schermata registrazione.
- **Output:** Form con campi editabili visualizzato.
- **Pre-condizione:** Schermata registrazione aperta.
- **Post-condizione:** Campi pronti per l'inserimento.
- **Effetti collaterali:** Nessuno.



- **Priorità:** Alta
- **Motivazione:** Il form richiede solo i dati essenziali (name, email, password) come da modello User in MongoDB.

#### **RF26: Inserimento Dati Registrazione**

- **Descrizione:** L'utente compila i campi nome, email e password.
- **Input:** Digitazione nei campi del form.
- **Output:** Valori caricati nei campi di input.
- **Pre-condizione:** Form registrazione visualizzato.
- **Post-condizione:** Dati pronti per validazione.
- **Effetti collaterali:** Nessuno.
- **Priorità:** Alta
- **Motivazione:** Non sono richiesti cognome separato né ruolo (tutti utenti standard).

#### **RF27: Validazione Dati Lato Client**

- **Descrizione:** Il sistema verifica lato client che: email sia in formato valido, password abbia almeno 6 caratteri, tutti i campi obbligatori siano compilati.
- **Input:** Dati inseriti dall'utente.
- **Output:** Abilitazione pulsante "Registrati" o messaggio di errore inline.
- **Pre-condizione:** Campi compilati.
- **Post-condizione:** Dati validati localmente o errori segnalati.
- **Effetti collaterali:** Feedback visivo immediato.
- **Priorità:** Alta
- **Motivazione:** Validazione client-side migliora UX evitando chiamate server per errori banali.

#### **RF28: Invio Richiesta al Server**

- **Descrizione:** Il sistema invia una richiesta POST a `/api/auth/register` con i dati validati (name, email, password).
- **Input:** Dati validati lato client.
- **Output:** Richiesta HTTP inviata al backend.
- **Pre-condizione:** Validazione client superata, connessione attiva.
- **Post-condizione:** Richiesta in attesa di risposta server.
- **Effetti collaterali:** Utilizzo rete.
- **Priorità:** Alta
- **Motivazione:** Comunicazione via API REST con backend Node.js su Render.com.

#### **RF29: Validazione Dati Lato Server**

- **Descrizione:** Il server valida i dati ricevuti: email formato corretto, email non già registrata, password minimo 6 caratteri.
- **Input:** Dati ricevuti dal client.
- **Output:** Conferma validazione o errore specifico (es. "Email già registrata").
- **Pre-condizione:** Richiesta ricevuta dal server.
- **Post-condizione:** Dati validati o errore restituito.
- **Effetti collaterali:** Query al database per verifica duplicati.
- **Priorità:** Alta
- **Motivazione:** Doppia validazione (client + server) garantisce integrità dati.

#### **RF30: Creazione Account nel Database**



- **Descrizione:** Il server crea un nuovo documento nella collection Users di MongoDB con: name, email, password (hashata con bcrypt), campi default (totalPoints: 0, co2Saved: 0, level: "Eco-Novizio", badges: [badge iniziale]).
- **Input:** Dati validati.
- **Output:** Documento utente creato con \_id generato.
- **Pre-condizione:** Validazione server superata.
- **Post-condizione:** Utente registrato e immediatamente attivo.
- **Effetti collaterali:** Scrittura su MongoDB Atlas.
- **Priorità:** Alta
- **Motivazione:** L'account è subito attivo senza necessità di verifica email.

#### RF31: Invio Email di Benvenuto

- **Descrizione:** Il sistema invia un'email di benvenuto all'indirizzo registrato tramite il servizio Resend. L'email è informativa e NON contiene link di attivazione.
- **Input:** Email utente appena registrato.
- **Output:** Email inviata (o errore loggato).
- **Pre-condizione:** Utente creato con successo.
- **Post-condizione:** Email inviata o fallimento gestito silenziosamente.
- **Effetti collaterali:** Chiamata API esterna a Resend.
- **Priorità:** Media
- **Motivazione:** Email di benvenuto migliora engagement ma non è bloccante per la registrazione.

#### RF32: Conferma Registrazione all'Utente

- **Descrizione:** Il sistema mostra un messaggio di successo all'utente ("Registrazione completata!") e lo reindirizza alla schermata di login.
- **Input:** Risposta positiva dal server.
- **Output:** Toast/messaggio di conferma + redirect a LoginActivity.
- **Pre-condizione:** Registrazione completata con successo.
- **Post-condizione:** Utente informato e pronto per il login.
- **Effetti collaterali:** Navigazione a nuova Activity.
- **Priorità:** Alta
- **Motivazione:** Feedback immediato all'utente che ha completato l'azione.

#### RF33: Gestione Errori di Registrazione

- **Descrizione:** In caso di errore (email duplicata, errore server, timeout), il sistema mostra un messaggio specifico all'utente senza perdere i dati inseriti.
- **Input:** Errore restituito dal server o dalla rete.
- **Output:** Messaggio di errore contestuale (es. "Email già registrata", "Errore di connessione").
- **Pre-condizione:** Errore durante la registrazione.
- **Post-condizione:** Utente informato, form preservato per correzione.
- **Effetti collaterali:** Nessuna modifica al database.
- **Priorità:** Alta
- **Motivazione:** Gestione errori user-friendly migliora l'esperienza di onboarding.



## 6. Tabella di tracciabilità

ID (RF)	R0 1	R0 2	R0 4	R0 5	R0 8	R0 9	R1 1	R1 3	R1 4	R1 5	R1 6	R2 3	R2 6	R2 7	R2 8
<b>RF01</b> (Accesso Profilo)	X	X											X		X
<b>RF02</b> (Recupero Dati)		X	X										X	X	X
<b>RF03</b> (Vis. Dati Pers.)	X		X										X	X	
<b>RF04</b> (Vis. Punteggi)	X	X											X		X
<b>RF05</b> (Vis. Amici)	X	X											X	X	X
<b>RF06</b> (Agg. Dinamico)		X											X		X
<b>RF08</b> (Sicurezza Profilo)			X										X	X	
<b>RF09</b> (Assegn. Quest)													X		X
<b>RF10</b> (Tracc. Quest)				X									X		X



<b>RF11</b> (Sistema Punti)				X									X		X
<b>RF12</b> (Rinnovo Quest)													X		X
<b>RF13</b> (Contatore Prog.)				X									X		X
<b>RF15</b> (Storico)	X	X											X		X
<b>RF16</b> (Schermata Login)					X								X		
<b>RF17</b> (Valid. Client)					X								X		
<b>RF18</b> (Autenticazione)						X	X					X	X	X	X
<b>RF19</b> (Acc. sempre attivo)					X					X			X	X	
<b>RF20</b> (Blocco errati)					X		X					X	X		
<b>RF21</b> (Sessione Sicura)						X	X						X		



<b>RF22</b> (Remember me)					X		X						X		
<b>RF23</b> (Registro Login)													X	X	
<b>RF24</b> (Errore Login)					X								X		
<b>RF25</b> (Logout)					X		X						X		
<b>RF26</b> (supporto pb accesso)					X								X	X	
<b>RF27</b> (Accesso schermata registrazione)							X						X		
<b>RF28</b> (visualizzazione form)								X					X		
<b>RF29</b> (Inser. Dati)								X			X		X	X	
<b>RF30</b> (Valid. Server)									X		X		X	X	
<b>RF31</b> (Registraz. DB)									X		X		X	X	X



<b>RF32</b> (creazione account in db)										X		X		
<b>RF33</b> (Invio Email benvenuto)									X			X	X	
<b>RF34</b> (conferma registrazione)								X				X	X	
<b>RF35</b> (gestione errore registrazione)														
<b>RF36</b> (permessi utente standard)														
<b>RF37</b> (tracciabilità creazione)														
<b>RF38</b> (Resilienza Email)									X			X		

## 7. Tabella di dipendenza dei requisiti

Requisito (ID)	Descrizione Sintetica	Dipende da (Prerequisito)
<b>RF2</b>	Recupero dati utente	<b>RF18, RF21</b>
<b>RF3</b>	Visualizzazione dati	<b>RF2</b>
<b>RF4</b>	Visualizzazione punteggi	<b>RF2, RF11</b>



<b>RF9</b>	Assegnazione quest	<b>RF18, RF21</b>
<b>RF10</b>	Tracciamento compl.	<b>RF9</b>
<b>RF11</b>	Sistema punti/ric.	<b>RF10</b>
<b>RF16</b>	Schermata login	
<b>RF17</b>	Inserimento credenziali client	<b>RF16</b>
<b>RF18</b>	Autenticazione cred.	<b>RF17, RF31</b>
<b>RF19</b>	Account sempre attivo	<b>RF31</b>
<b>RF20</b>	Blocco gestione cred errate	<b>RF18</b>
<b>RF21</b>	Creazione sess. sicura	<b>RF18</b>
<b>RF22</b>	Remember me	<b>RF21</b>
<b>RF23</b>	Tracciamento login	<b>RF18</b>
<b>RF24</b>	Messaggistica errore login	<b>RF17, RF18, RF20</b>
<b>RF25</b>	Logout + invalidazione sessione	<b>RF21</b>
<b>RF26</b>	Supporto problemi accesso	<b>RF16</b>
<b>RF27</b>	Accesso scherm. registrazione	<b>RF16</b>
<b>RF28</b>	Avvio procedura registrazione	<b>RF27</b>
<b>RF29</b>	Inserimento Dati Utente	<b>RF28</b>
<b>RF30</b>	Validazione lato server	<b>RF29</b>



<b>RF31</b>	Registrazione nel DB	<b>RF30</b>
<b>RF32</b>	Scelta psw utente	<b>RF29</b>
<b>RF33</b>	Invio mail benvenuto	<b>RF31</b>
<b>RF34</b>	Gestione duplicati mail	<b>RF30</b>
<b>RF35</b>	Conferma registrazione	<b>RF31</b>
<b>RF36</b>	Permessi utenti standard	<b>RF31</b>
<b>RF37</b>	Registrazione data creazione account	<b>RF31</b>
<b>RF38</b>	Resilienza invio mail	<b>RF33</b>
<b>RNF5</b>	Affidabilità tracc.	<b>RF10</b>
<b>RNF6</b>	Usabilità interf. quest	<b>RF9</b>