# Specifica dei requisiti software

**per**

# <NoProfitHub>

**Versione 1.5 approvata**

**Preparato da <Luigi Imperatore>**

**<Università degli studi della Campania Luigi Vanvitelli>**

**<08/07/2025>**

# 1. Introduzione

## 1.1 Scopo

Questo documento descrive i requisiti del sistema NoProfitHub, una piattaforma per il coordinamento di volontari, associazioni e progetti no-profit.

## 1.2 Convenzioni del Documento

Tutti i requisiti sono numerati. Le parole chiave come 'deve', 'dovrebbe', 'può' seguono la convenzione RFC 2119.

## 1.3 Destinatari e Suggerimenti di Lettura

Il documento è destinato agli sviluppatori, ai gestori di progetto, alle associazioni coinvolte e agli stakeholder.

## 1.4 Ambito del Prodotto

NoProfitHub consente la gestione di progetti no-profit, la registrazione di volontari, il matching tra competenze e progetti, la gestione dei turni e la generazione di attestati.

## 1.5 Riferimenti

*Lo sviluppo del sistema conterrà le applicazioni del materiale teorico del prof Diomaiuta,le guide su w3school per l'utilzzo dei linguaggi HTML,CSS,PHP e Javascript.*

# 2. Descrizione Generale

## 2.1 Prospettiva del Prodotto

Il sistema *NoProfitHub* è una piattaforma web-based progettata per facilitare la gestione delle attività tra associazioni no profit e volontari. Si inserisce nell'ambito delle soluzioni di supporto alla collaborazione sociale, offrendo funzionalità per la gestione dei progetti, la pianificazione dei turni, il matching dinamico tra volontari e iniziative, la generazione di attestati e la redazione di report sull'impatto sociale. Il sistema si integra con un database relazionale e adotta un'architettura modulare basata su PHP e MySQL. È pensato per essere scalabile, multi-utente e facilmente estendibile, con un'interfaccia separata per volontari, associazioni e amministratori.

## 2.2 Funzioni del Prodotto

Il sistema *NoProfitHub* offre un set completo di funzionalità per la gestione di attività no profit. Le associazioni possono creare progetti, ricevere candidature da volontari, assegnare turni in base a disponibilità e competenze, e generare attestati personalizzati. I volontari possono registrarsi, aggiornare il proprio profilo con competenze e disponibilità, candidarsi ai progetti e ricevere notifiche su esiti e turni assegnati. Il sistema permette inoltre di tracciare ore svolte, visualizzare lo storico attività e ricevere attestazioni. L'amministratore può monitorare statistiche globali, validare le attività e redigere report sull'impatto sociale del volontariato coordinato attraverso la piattaforma.

## 2.3 Classi e Caratteristiche degli Utenti

Il sistema *NoProfitHub* prevede tre principali classi di utenti, ciascuna con ruoli e funzionalità distinte:

1. **Volontari**
    - Possono registrarsi e aggiornare il proprio profilo (competenze, disponibilità giornaliera e fasce orarie).
    - Possono candidarsi ai progetti aperti delle associazioni.
    - Ricevono notifiche per accettazione/rifiuto candidature e per l'assegnazione ai turni.
    - Possono visualizzare lo storico delle attività svolte, gli attestati ricevuti e l'impatto sociale generato.
2. **Associazioni**
    - Possono registrarsi e gestire un profilo organizzativo.
    - Creano e pubblicano progetti di volontariato.
    - Gestiscono le candidature dei volontari, assegnano turni rispettando le disponibilità.
    - Possono generare attestati personalizzati per i volontari.
    - Monitorano l'andamento dei progetti e le ore totali erogate.
3. **Amministratore (Admin)**

- Ha accesso a una dashboard con statistiche globali (progetti, ore, utenti attivi).
- Supervisiona l'attività della piattaforma e ne valida l'uso conforme.
- Può generare e pubblicare report sull'impatto sociale complessivo delle attività svolte.

## 2.4 Ambiente Operativo

Lo sviluppo dell'applicazione web avviene con **Visual Studio Code** come editor e **XAMPP** come ambiente server locale. Il sistema è costruito utilizzando quattro linguaggi principali:

- **HTML** per la struttura delle pagine,
- **CSS** per lo stile grafico,
- **PHP** per la logica lato server, la gestione delle sessioni e l'interazione con il database **MySQL**,
- **JavaScript** per la dinamicità dell'interfaccia, come la validazione dei dati o la visualizzazione dei calendari dei turni.

Tutti gli strumenti lavorano insieme per fornire una piattaforma efficiente per volontari, associazioni e amministratori.

## 2.5 Vincoli di Progettazione e Implementazione

Nel progetto NoProfitHub, la progettazione e implementazione sono state vincolate da alcune scelte tecnologiche e funzionali chiave. Il sistema è sviluppato principalmente in PHP per il backend, con interfaccia web basata su HTML e CSS, e utilizza JavaScript solo in forma inline per semplici interazioni lato client, senza l'impiego di file `.js` esterni o framework avanzati. È stato scelto di non utilizzare librerie o plugin esterni per mantenere la piattaforma leggera e facilmente manutenibile. Inoltre, il design è orientato alla semplicità e alla chiarezza, privilegiando l'usabilità per utenti con diversi livelli di esperienza, e limitando l'uso di funzionalità avanzate o costose. o componenti frontend complessi. Infine, la piattaforma deve garantire un'efficiente gestione di associazioni, volontari, progetti e turni, rispettando vincoli di compatibilità tra disponibilità e competenze, con attenzione a una gestione trasparente e accessibile delle funzionalità chiave come la generazione di attestati e il matching progetti-volontari.

### 2.6 Documentazione Utente

La consegna del prodotto è accompagnata da una guida all'utilizzo, nonostante la web application sia stata progettata per essere intuitiva e di facile fruizione da parte degli utenti.

### 2.7 Assunzioni e Dipendenze

Il corretto funzionamento di NoProfitHub presuppone che gli utenti dispongano di una connessione internet stabile e di un browser moderno compatibile con le tecnologie web utilizzate (HTML5, CSS3 e JavaScript inline). Si assume inoltre che gli utenti abbiano competenze informatiche di base sufficienti per interagire con un'interfaccia web standard. Dal punto di vista tecnologico, la piattaforma dipende da un ambiente server con supporto PHP e database MySQL per la gestione dei dati. Non sono previste dipendenze da librerie o framework esterni, in linea con l'obiettivo di mantenere il sistema leggero e facilmente manutenibile. Infine, si assume che le associazioni utilizzino il sistema per gestire i progetti e i turni in modo coerente, fornendo dati accurati sulle disponibilità e le competenze dei volontari.

## 3. Requisiti delle Interfacce Esterne

### 3.1 Interfacce Utente

INoProfitHub offre interfacce utente semplici e funzionali, studiate per facilitare l'interazione sia degli amministratori di associazioni sia dei volontari. Le dashboard sono personalizzate in base al ruolo dell'utente e consentono di accedere facilmente alle principali funzionalità: gestione e visualizzazione dei progetti, iscrizione e candidatura, gestione turni, consultazione degli attestati e visualizzazione delle statistiche di partecipazione. L'interfaccia è sviluppata utilizzando HTML e CSS, con l'uso limitato di JavaScript inline per migliorare l'esperienza utente attraverso semplici alert e reindirizzamenti. Particolare attenzione è stata posta alla chiarezza e all'usabilità, garantendo una navigazione intuitiva anche per utenti con competenze informatiche base.

### 3.2 Interfacce Hardware

NoProfitHub è una web application accessibile tramite dispositivi connessi a Internet, senza necessità di hardware specifico dedicato. Gli utenti possono interagire con la piattaforma utilizzando computer desktop, laptop, tablet o smartphone, purché dotati di browser compatibili. Non sono richiesti componenti hardware particolari o periferiche esterne per l'utilizzo delle funzionalità offerte.

### 3.3 Interfacce Software

NoProfitHub si basa su un'architettura client-server, con backend sviluppato in PHP e database MySQL per la gestione dei dati. L'interfaccia utente utilizza HTML e CSS, con JavaScript inline per funzionalità base lato client. La piattaforma richiede un server web compatibile con PHP e un browser aggiornato per l'accesso.

### 3.4 Interfacce di Comunicazione

NoProfitHub utilizza protocolli standard HTTP/HTTPS per la comunicazione tra client e server, garantendo sicurezza e affidabilità nello scambio dei dati. Le notifiche agli utenti, come conferme di candidatura e aggiornamenti sui turni, sono gestite tramite email. Non sono previste comunicazioni in tempo reale o tramite altri protocolli.

---

---

## 4. Funzionalità del Sistema

---

### 4.1 Gestione Progetti

#### 4.1.1 Description and Priority

Funzionalità fondamentale che consente alle associazioni di creare, modificare e cancellare progetti. Ogni progetto include titolo, descrizione, luogo, durata e competenze richieste.

#### 4.1.2 Stimulus/Response Sequences

1. L'associazione accede alla dashboard.
2. Clicca su "Crea nuovo progetto".
3. Inserisce le informazioni richieste.
4. Il sistema salva i dati nel database e mostra l'elenco aggiornato dei progetti.

#### 4.1.3 Functional Requirements

REQ-1: Il sistema deve fornire un'interfaccia per l'inserimento dei dati del progetto.
REQ-2: Il sistema deve salvare, modificare e cancellare progetti nel database.
REQ-3: Il sistema deve mostrare all'associazione solo i progetti che ha creato.

---

### 4.2 Candidatura ai Progetti

#### 4.2.1 Description and Priority

Permette ai volontari di candidarsi ai progetti visibili nella loro dashboard. Le associazioni possono accettare o rifiutare le candidature.

#### 4.2.2 Stimulus/Response Sequences

1. Il volontario visualizza i progetti disponibili.
2. Clicca su "Candidati".
3. Il sistema registra la candidatura.
4. L'associazione riceve la notifica e può accettare o rifiutare.

#### 4.2.3 Functional Requirements

REQ-1: Il sistema deve visualizzare i progetti disponibili nella dashboard del volontario.
REQ-2: Il sistema deve permettere la candidatura e il ritiro della candidatura.
REQ-3: Il sistema deve salvare lo stato della candidatura.
REQ-4: Il sistema deve permettere all'associazione di accettare o rifiutare la candidatura.

---

### 4.3 Gestione Turni

#### 4.3.1 Description and Priority

Le associazioni possono creare turni specifici per ogni progetto, indicando giorno, fascia oraria e numero di ore. Possono assegnare ai turni solo volontari disponibili e già accettati.

#### 4.3.2 Stimulus/Response Sequences

1. L'associazione seleziona un progetto.
2. Crea uno o più turni con giorno, fascia oraria e durata.
3. Il sistema salva il turno.
4. L'associazione assegna i volontari disponibili.

#### 4.3.3 Functional Requirements

REQ-1: Il sistema deve permettere la creazione, modifica e cancellazione dei turni.
REQ-2: Il sistema deve verificare la disponibilità del volontario prima dell'assegnazione.

REQ-3: Il sistema deve impedire l'assegnazione se il turno è pieno.
REQ-4: Il sistema deve salvare i turni nel database.

---

## 4.4 Matching Volontari-Progetti

### 4.4.1 Description and Priority

Il sistema propone abbinamenti tra volontari e progetti basandosi esclusivamente sulle competenze dichiarate. È una funzionalità importante per semplificare la ricerca di progetti adatti.

### 4.4.2 Stimulus/Response Sequences

1. Il volontario accede alla dashboard.
2. Il sistema mostra i progetti compatibili con le sue competenze.
3. L'associazione visualizza l'elenco di volontari compatibili con ciascun progetto.

### 4.4.3 Functional Requirements

REQ-1: Il sistema deve confrontare le competenze del volontario con i requisiti del progetto.
REQ-2: Il sistema deve visualizzare solo i progetti compatibili nella dashboard del volontario.
REQ-3: Il sistema deve visualizzare solo i volontari compatibili nella dashboard dell'associazione.

---

## 4.5 Generazione Attestati

### 4.5.1 Description and Priority

Funzionalità che consente la generazione di attestati solo se il volontario ha svolto almeno un turno. L'attestato include una descrizione, ma attualmente non viene generato in PDF né inviato per email.

### 4.5.2 Stimulus/Response Sequences

1. L'associazione accede alla sezione "Genera Attestato".

2. Seleziona un volontario.

3. Il sistema verifica che abbia completato almeno un turno.

4. Viene generato un attestato visualizzabile a schermo.

#### 4.5.3 Functional Requirements

REQ-1: Il sistema deve controllare il numero di turni svolti dal volontario.
REQ-2: Il sistema deve generare un attestato contenente nome, progetto, ore totali e descrizione.
REQ-3: Il sistema non genera ancora file PDF né invia email.

---

### 4.6 Monitoraggio Ore e Report

#### 4.6.1 Description and Priority

Tutte le ore svolte dai volontari vengono registrate. Le associazioni possono visualizzare report aggregati per progetto o volontario.

#### 4.6.2 Stimulus/Response Sequences

1. Il volontario partecipa a un turno.

2. Il sistema registra le ore svolte.

3. L'associazione accede alla sezione report per consultare i dati.

#### 4.6.3 Functional Requirements

REQ-1: Il sistema deve registrare le ore per ciascun turno completato.
REQ-2: Il sistema deve calcolare il totale ore per volontario e progetto.
REQ-3: Il sistema deve fornire una sezione report per l'associazione.

---

### 4.7 Sistema di Notifiche *(da implementare)*

#### 4.7.1 Description and Priority

Funzionalità prevista per notificare volontari e associazioni in caso di candidatura, accettazione, rifiuto o assegnazione a un turno. Attualmente non implementata.

### 4.7.2 Stimulus/Response Sequences

1. Un volontario si candida a un progetto.
2. Il sistema dovrebbe inviare una notifica o email all'associazione.
3. In caso di risposta, il volontario dovrebbe ricevere notifica.

### 4.7.3 Functional Requirements

REQ-1: Il sistema deve generare notifiche per ogni evento rilevante.
REQ-2: Il sistema deve inviare email (non ancora implementato).
REQ-3: Le notifiche devono essere visibili nella dashboard dell'utente.

---

## 4.8 Gestione Profilo Utente

### 4.8.1 Description and Priority

Ogni utente può modificare il proprio profilo, inclusi dati personali, competenze possedute e disponibilità settimanale. Funzionalità essenziale per il corretto matching con i progetti.

### 4.8.2 Stimulus/Response Sequences

1. L'utente accede alla sezione "Profilo".
2. Modifica i dati desiderati.
3. Il sistema salva le modifiche e aggiorna il profilo.

### 4.8.3 Functional Requirements

REQ-1: Il sistema deve permettere la modifica di nome, cognome, email, competenze e disponibilità.
REQ-2: Il sistema deve salvare i dati nel database.
REQ-3: Il sistema deve utilizzare competenze e disponibilità aggiornate per il matching.

## 5. Requisiti Non Funzionali

### 5.1 Requisiti di Prestazioni

- Il sistema deve garantire tempi di risposta inferiori a 2 secondi per le operazioni principali (accesso, visualizzazione dashboard, candidature).
- La gestione dei dati e delle query deve essere ottimizzata per supportare contemporaneamente almeno 100 utenti attivi senza degrado significativo delle prestazioni.

### 5.2 Requisiti di Affidabilità

- Il sistema deve mantenere l'integrità dei dati evitando perdite o corruzioni, anche in caso di interruzioni improvvise.
- I dati relativi a candidature, turni e progetti devono essere persistenti e non modificabili senza autorizzazione.

### 5.3 Requisiti di Usabilità

- L'interfaccia utente deve essere semplice e intuitiva, adatta sia ad utenti esperti che a utenti con competenze informatiche di base.
- Deve essere garantita la coerenza grafica e funzionale tra le dashboard di volontari e associazioni.

### 5.4 Requisiti di Manutenibilità

- Il codice sorgente deve essere documentato e organizzato per facilitare futuri aggiornamenti e miglioramenti.

- Il sistema deve essere modulare per consentire l'implementazione futura di nuove funzionalità come notifiche via email e generazione automatica di PDF.

---

### 5.5 Requisiti di Portabilità

- L'applicazione web deve essere compatibile con i principali browser moderni (Chrome, Firefox, Edge).

- Il sistema deve poter essere installato e configurato facilmente su server con ambiente PHP e MySQL.

---

### 5.6 Requisiti di Sicurezza

- L'accesso al sistema deve essere protetto tramite autenticazione con email e password (2FA opzionale da implementare).

- I dati sensibili, come password e dati personali, devono essere memorizzati in forma cifrata.

- Deve essere prevista una gestione dei permessi per differenziare ruoli di volontario, associazione e amministratore.

---

### 5.7 Requisiti di Scalabilità

- Il sistema deve essere progettato per poter gestire un aumento progressivo del numero di utenti, progetti e turni senza necessità di riscrittura completa.

## 6. Database

### Modello concettuale: diagramma ER

Il modello concettuale del database si basa su una serie di entità chiave che rappresentano gli elementi centrali della piattaforma NoProfitHub, organizzati per garantire coerenza e integrità dei dati.

- Utente rappresenta sia le associazioni sia i volontari, distinguibili in base al campo `tipo`. Questa scelta elimina la necessità di entità separate, semplificando la struttura e riducendo la ridondanza. Ogni utente può creare progetti (nel caso di associazioni) o candidarsi/essere assegnato a turni (nel caso di volontari).

- Progetto è l'entità che rappresenta le iniziative proposte dalle associazioni. Ogni progetto è collegato a un solo utente (associazione), ma può coinvolgere più turni e più volontari attraverso le candidature e le assegnazioni.

- Turno è associato a un singolo progetto e rappresenta un'attività pianificata in un determinato giorno e fascia oraria. Ogni turno può essere assegnato a più volontari (in base alla disponibilità e compatibilità).

- Competenze_Predefinite definisce l'elenco standard di competenze disponibili nella piattaforma. È una lista centralizzata usata per il matching tra utenti e progetti.

- Volontario_Competenze è una tabella di associazione tra `Utente` (tipo volontario) e `Competenze_Predefinite`, che permette a ciascun volontario di indicare le proprie competenze.

- Progetto_Competenze associa ogni progetto con le competenze richieste per parteciparvi, al fine di facilitare il matching con i volontari.

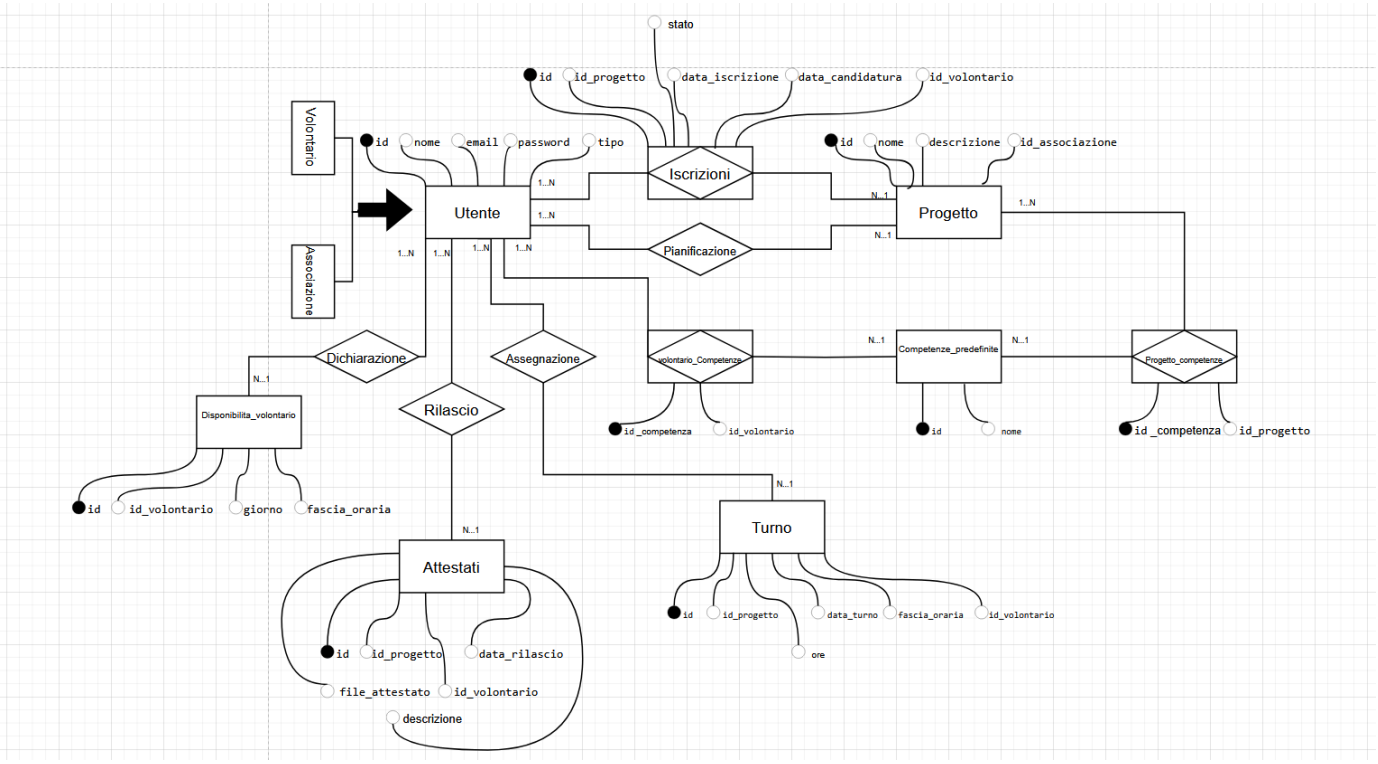
- Iscrizioni rappresenta la candidatura di un volontario a un progetto. Include anche lo stato della candidatura (in attesa, accettata, rifiutata), rendendo possibile la gestione da parte dell'associazione.

- Disponibilita_Volontario definisce i giorni e le fasce orarie in cui ogni volontario è disponibile. Viene utilizzata nel processo di assegnazione automatica ai turni.

- Attestati registra gli attestati generati per ogni volontario che ha partecipato ad almeno un turno. Contiene anche il file PDF e una descrizione dell'esperienza.

**Relazioni principali:**

- Un Utente (associazione) può creare molti Progetti, ma ogni progetto è associato a un solo utente.
- Un Progetto può avere più Turni, ma ogni turno è associato a un solo progetto.
- Un Turno può essere associato a più volontari, e ogni volontario può partecipare a più turni (relazione molti-a-molti implementata tramite le Iscrizioni e controllata tramite disponibilità).
- Le entità Volontario_Competenze e Progetto_Competenze modellano relazioni molti-a-molti tra utenti/progetti e competenze.
- Un Attestato è generato per ogni volontario che ha partecipato ad almeno un turno, ed è associato a un utente e ai relativi progetti/turni svolti.




## 7. Ristrutturazione Logica

### 7.1 Tabella dei Volumi

La tabella dei volumi indica le dimensioni previste delle principali entità del sistema per garantire scalabilità e prestazioni ottimali.

| Entità | Volume Attuale Stimato | Volume Previsto (1 anno) | Note |
|---|---|---|---|
| Utenti (Volontari + Associazioni) | 8 | 100 | Utenti attivi registrati |
| Progetti | 3 | 50 | Progetti creati dalle associazioni |
| Candidature | 3 | 200 | Candidature inviate dai volontari |
| Turni | 3 | 250 | Turni creati per i progetti |
| Attestati | 0 | 250 | Attestati generati per volontari |
| Competenze | 12 | 35 | Numero di competenze predefinite |

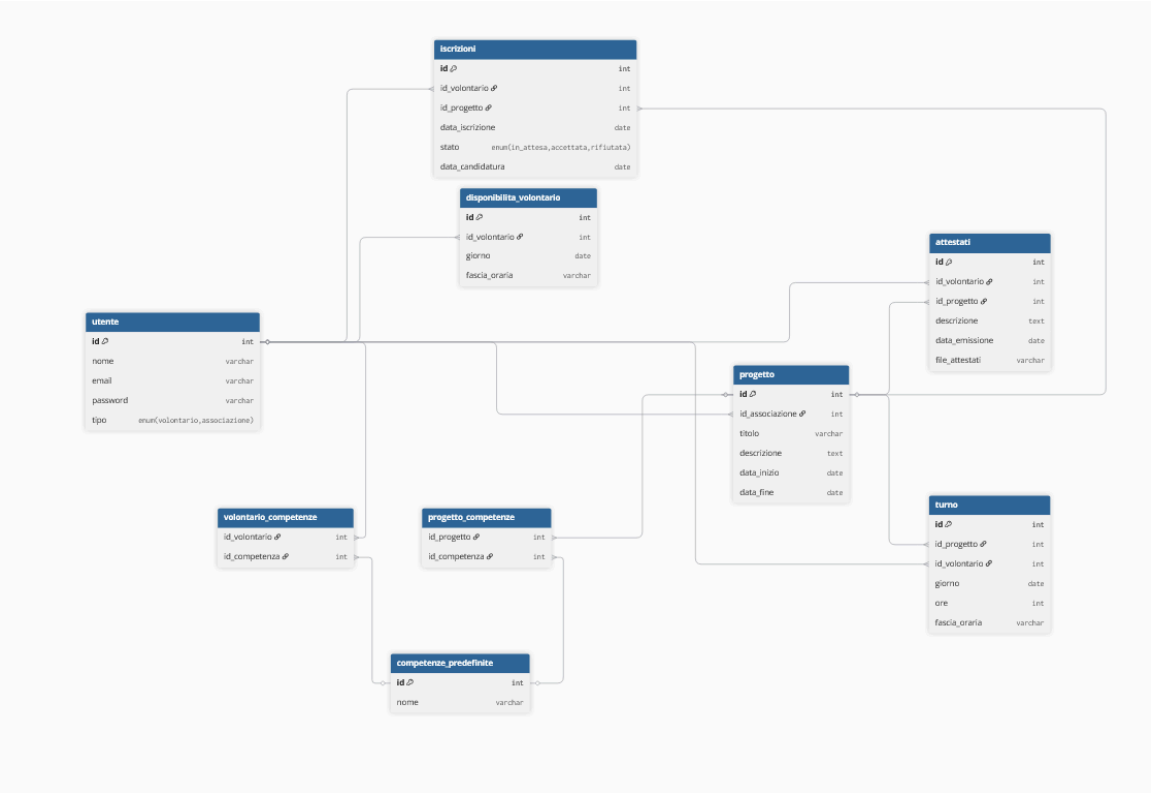
## 7.2 Ristrutturazione dello schema concettuale

### 7.2.1 Analisi delle ridondanze

Nel rifacimento del modello dati ho tolto tutte le parti superflue e spostato le informazioni in strutture più pulite: i vecchi campi di testo libero per competenze e disponibilità sono stati sostituiti da vere tabelle di associazione, così ogni relazione diventa chiara e gestibile. Ho unito il tracciamento delle ore e delle partecipazioni in un'unica tabella turni, eliminando complessità e ridondanze tra ore_volontariato e partecipazioni. In particolare, ho introdotto la tabella disponibilita_volontario che ora gestisce in autonomia giorni e fasce orarie, rendendo superflue le lookup table giorni_settimana e fasce_orarie e azzerando la necessità di tabelle di appoggio. Per mantenere il tutto in ordine ho aggiunto vincoli di chiave esterna tra le entità e creato indici mirati sulle colonne più consultate, velocizzando le query. Il risultato è un database più leggero, con query più semplici e un'architettura solida e pronta a crescere senza intoppi.

File_sqlvecchio

### 7.3 Progettazione Logica




## 8. Database implementation

```
CREATE TABLE utenti (
  id INT NOT NULL AUTO_INCREMENT,
  tipo ENUM('volontario','associazione','admin') NOT NULL,
  nome VARCHAR(100) NOT NULL,
  email VARCHAR(100) NOT NULL,
  password VARCHAR(255) NOT NULL,
  PRIMARY KEY (id)
);
```

```
CREATE TABLE competenze_predefinite (
  id INT NOT NULL AUTO_INCREMENT,
  nome VARCHAR(100) NOT NULL,
  PRIMARY KEY (id)
);

CREATE TABLE volontario_competenze (
  id_volontario INT NOT NULL,
  id_competenza INT NOT NULL,
  PRIMARY KEY (id_volontario, id_competenza)
);

CREATE TABLE progetto_competenze (
  id_progetto INT NOT NULL,
  id_competenza INT NOT NULL,
  PRIMARY KEY (id_progetto, id_competenza)
);

CREATE TABLE progetti (
  id INT NOT NULL AUTO_INCREMENT,
  id_associazione INT,
  titolo VARCHAR(150) NOT NULL,
  descrizione TEXT,
  data_inizio DATE,
  data_fine DATE,
```

```
  PRIMARY KEY (id)
);

CREATE TABLE iscrizioni (
  id INT NOT NULL AUTO_INCREMENT,
  id_progetto INT,
  id_volontario INT,
  data_iscrizione DATETIME DEFAULT CURRENT_TIMESTAMP(),
  stato ENUM('in_attesa','accettato','rifiutato') DEFAULT 'in_attesa',
  data_candidatura DATETIME DEFAULT CURRENT_TIMESTAMP(),
  PRIMARY KEY (id)
);

CREATE TABLE turni (
  id INT NOT NULL AUTO_INCREMENT,
  id_progetto INT NOT NULL,
  id_volontario INT NOT NULL,
  data_turno DATE NOT NULL,
  ore INT NOT NULL,
  fascia_oraria VARCHAR(20),
  PRIMARY KEY (id)
);

CREATE TABLE disponibilita_volontario (
  id INT NOT NULL AUTO_INCREMENT,
  id_volontario INT NOT NULL,
```

```
  giorno VARCHAR(20) NOT NULL,
  fascia_oraria VARCHAR(50) NOT NULL,
  PRIMARY KEY (id)
);

CREATE TABLE attestati (
  id INT NOT NULL AUTO_INCREMENT,
  id_volontario INT NOT NULL,
  id_progetto INT NOT NULL,
  file_attestato VARCHAR(255),
  descrizione TEXT,
  data_rilascio DATE,
  PRIMARY KEY (id)
);
```

## Descrizione sintetica

- Lo schema è basato su MySQL 8.0, InnoDB e utf8mb4 per transazioni ACID e supporto Unicode.
- Tutte le tabelle principali hanno chiavi primarie AUTO_INCREMENT; le associazioni many-to-many usano chiavi composte.
- Gli ENUM gestiscono stati e tipi (`tipo` in `utenti`, `stato` in `iscrizioni`).
- Le tabelle di join (`volontario_competenze`, `progetto_competenze`) normalizzano competenze e evitano campi free-text.
- `disponibilita_volontario` contiene direttamente giorno e fascia oraria, eliminando lookup esterne.
- `turni` centralizza ore, giorno e fascia di ciascun turno, sostituendo tabelle ridondanti.
- `attestati` collega volontari e progetti ai file dei certificati.
- Lo schema è leggero, le query restano veloci e il design si presta a futuri ampliamenti.

## 9. Test Cases – NoProfitHub

Ogni test case è presentato con:

- **ID del test**
- **Obiettivo**
- **Input**
- **Precondizioni**
- **Passi del test**
- **Risultato atteso**

---

### 9.1 TC-001: Registrazione Volontario

- **Obiettivo**: Verificare che un utente possa registrarsi come volontario.
- **Input**: Nome, email, password, conferma password.
- **Precondizioni**: L'utente non deve essere già registrato.
- **Passi**:
    1. Accedere alla pagina di registrazione.
    2. Inserire dati validi.
    3. Inviare il modulo.
- **Risultato atteso**: L'utente viene salvato nel database e riceve conferma via email(da implementare).

---

### 9.2 TC-002: Login Associazione

- **Obiettivo**: Verificare che un’associazione possa autenticarsi.
- **Input**: Email, password.
- **Precondizioni**: L’associazione è già registrata.
- **Passi**:
    1. Accedere al modulo di login.
    2. Inserire email e password corretti.
    3. Inviare il modulo.
- **Risultato atteso**: Accesso alla dashboard dell’associazione.

---

### 9.3 TC-003: Creazione Progetto (Associazione)

- **Obiettivo**: Verificare che un’associazione possa creare un nuovo progetto.
- **Input**: Titolo, descrizione, competenze richieste, date.
- **Precondizioni**: Associazione autenticata.
- **Passi**:
    1. Accedere alla dashboard.
    2. Selezionare "Crea Progetto".
    3. Inserire i dati e confermare.
- **Risultato atteso**: Il progetto è salvato nel database e visibile nella lista.

---

### 9.4 TC-004: Candidatura Volontario a Progetto

- **Obiettivo**: Verificare che un volontario possa candidarsi a un progetto.

- **Input**: Clic sul pulsante "Candidati".
- **Precondizioni**: Il progetto deve essere attivo. Il volontario deve essere loggato.
- **Passi**:
    1. Accedere alla dashboard volontario.
    2. Visualizzare i progetti compatibili.
    3. Candidarsi a uno di essi.
- **Risultato atteso**: Candidatura salvata con stato "in attesa".

---

### 9.5 TC-005: Assegnazione Volontario a Turno

- **Obiettivo**: Verificare che l'associazione possa assegnare un volontario a un turno.
- **Input**: Selezione del volontario da menu a tendina nella tabella turni.
- **Precondizioni**: Il volontario deve essere disponibile in quella fascia oraria.
- **Passi**:
    1. Aprire gestione turni.
    2. Scegliere il turno e il volontario disponibile.
    3. Cliccare "Aggiungi".
- **Risultato atteso**: Il volontario risulta assegnato e il turno aggiornato nel DB.

---

### 9.6 TC-006: Generazione Attestato

- **Obiettivo**: Verificare che l'attestato venga generato solo dopo almeno un turno svolto.

- **Input**: Click su “Genera Attestato” nella dashboard dell’associazione.
- **Precondizioni**: Il volontario ha partecipato almeno a un turno del progetto.
- **Passi**:
    1. Accedere alla gestione progetto.
    2. Selezionare volontario idoneo.
    3. Inserire i dati nella finestra
    4. Generare l’attestato.
- **Risultato atteso**: PDF generato e inviato al volontario via email.(da implementare), però il volontario può visualizzare il proprio attesto nella propria home e scaricare il file allegato.

---

### 9.7 TC-007: Matching Competenze Volontario-Progetto

- **Obiettivo**: Verificare che i progetti mostrati siano compatibili con le competenze del volontario.
- **Input**: Nessuno (il sistema mostra progetti compatibili).
- **Precondizioni**: Il volontario ha selezionato almeno una competenza.
- **Passi**:
    1. Accedere alla dashboard volontario.
    2. Verificare la lista dei progetti suggeriti.
- **Risultato atteso**: Tutti i progetti visibili richiedono almeno una competenza del volontario.

---

### 9.8 TC-008: Invio Notifica di Accettazione

- **Obiettivo**: Verificare che una notifica venga inviata quando la candidatura viene accettata.
- **Input**: Cambio di stato della candidatura da "in attesa" a "accettata".
- **Precondizioni**: Esiste una candidatura attiva.
- **Passi**:
    1. L'associazione accede alla lista candidature.
    2. Seleziona "Accetta".
- **Risultato atteso**: Email di conferma inviata al volontario.(da implementare),però il volontario può visualizzare il cambio di stato da "in attesa" a "accettata".

---

### 9.9 TC-009: Calcolo Ore Totali Volontario

- **Obiettivo**: Verificare che le ore siano correttamente calcolate.
- **Input**: Nessuno (visualizzazione nel profilo volontario).
- **Precondizioni**: Il volontario ha partecipato ad almeno un turno.
- **Passi**:
    1. Accedere al profilo.
    2. Osservare il campo "Ore totali svolte".
- **Risultato atteso**: Totale corretto (somma dei turni completati).

---

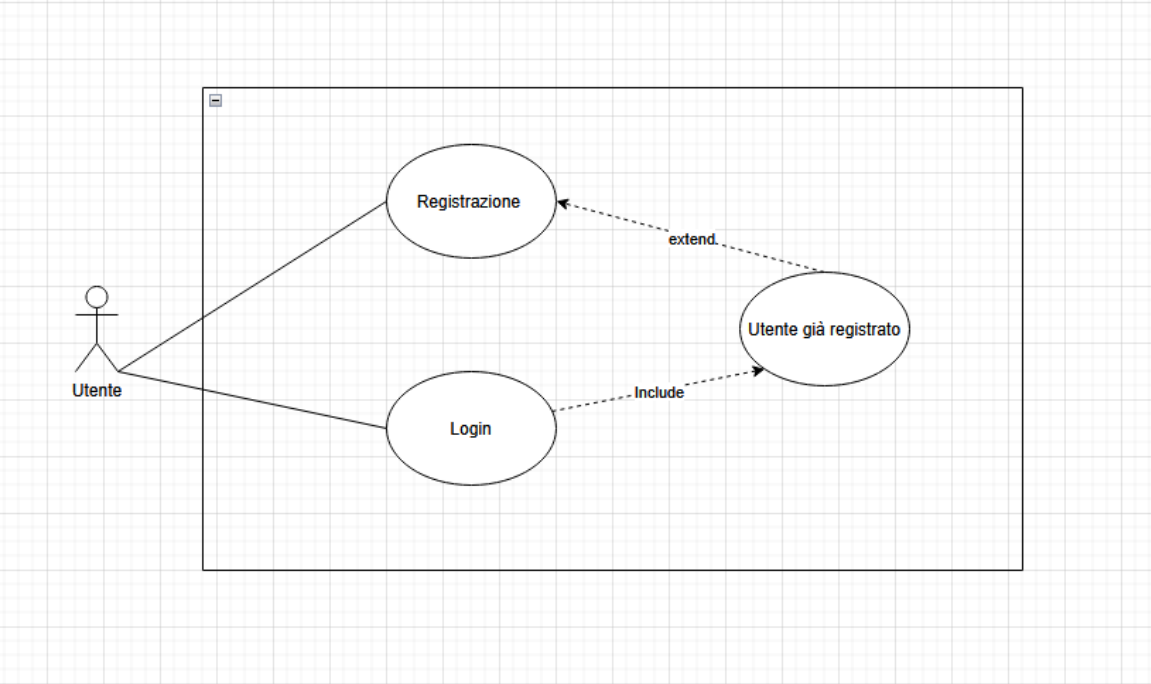
## Appendice A: Glossario

| Termine | Definizione |
| --- | --- |
| **Volontario** | Utente registrato sulla piattaforma che offre il proprio tempo e competenze per collaborare a progetti gestiti da associazioni no-profit. |
| **Associazione** | Organizzazione no-profit registrata sulla piattaforma, responsabile della creazione e gestione di progetti, turni e candidature di volontari. |
| **Progetto** | Iniziativa creata da un'associazione, con un obiettivo definito e requisiti specifici, a cui i volontari possono candidarsi per offrire il loro supporto. |
| **Turno** | Un'unità temporale (giorno + fascia oraria) all'interno di un progetto, in cui i volontari possono essere assegnati per svolgere attività pratiche. |
| **Candidatura** | Richiesta inviata da un volontario per partecipare a un progetto. Può essere in stato: *in attesa*, *accettata*, *rifiutata*. |
| **Disponibilità** | Informazioni inserite dal volontario riguardo i giorni e le fasce orarie in cui è disponibile per partecipare ai turni. |
| **Competenza** | Abilità o area di esperienza dichiarata dal volontario (es. comunicazione, logistica, primo soccorso) usata per il matching con i progetti. |
| **Matching** | Processo automatico che confronta le competenze del volontario con i requisiti dei progetti per suggerire compatibilità reciproche. |
| **Dashboard Volontario** | Interfaccia personale per il volontario dove può visualizzare progetti compatibili, candidarsi, gestire il profilo, visualizzare ore svolte e attestati. |
| **Dashboard Associazione** | Area privata dell'associazione per creare e modificare progetti, gestire candidature, assegnare turni e generare attestati. |

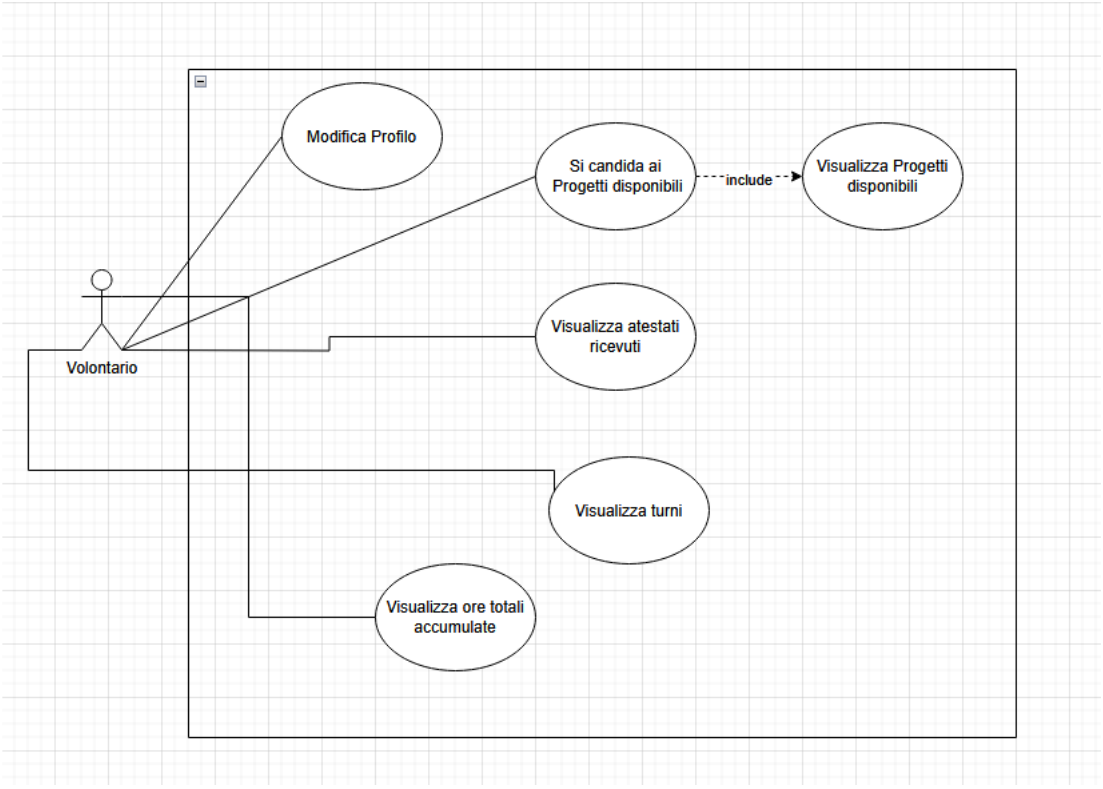
| | |
|---|---|
| **Dashboard Admin** | Interfaccia per l'amministratore di sistema, che consente di visualizzare statistiche globali, gestire utenti e monitorare le attività. |
| **Attestato** | Documento PDF generato dalla piattaforma, che certifica la partecipazione del volontario a un progetto. |
| **Ore di volontariato** | Totale delle ore svolte da un volontario nei turni completati, usato anche per generare report e attestati. |
| **Fascia oraria** | Intervallo di tempo predefinito (es. mattina, pomeriggio, sera) utilizzato per descrivere la disponibilità e la pianificazione dei turni. |
| **Requisiti del progetto** | Insieme di competenze richieste da un progetto per cui un volontario può risultare compatibile. |
| **Report impatto sociale** | Documento o sezione della piattaforma che aggrega ore, attività e progetti svolti, utile per mostrare l'efficacia dell'azione volontaria. |

## Diagrammi dei casi D’uso

### Registrazione/accesso



## Dashboard Volontario



## Dashboard Associazione