

Documentazione del Progetto: Visite Guidate

Felappi Giorgio (740733)

Barbetti Daniel (740851)

Folgoni Leonardo (740821)

9 giugno 2025

Indice

1	Introduzione	2
1.1	Overview applicativo: Visite Guidate	2
1.2	Ruoli e responsabilità	2
2	Casi d'Uso	3
2.1	Casi d'uso per ogni utente	4
2.2	Casi d'uso per Configuratori	7
2.3	Casi d'uso per Volontari	12
2.4	Casi d'uso per Fruitori	13
3	Infrastruttura informatica	16
3.1	Diagramma ER database	16
4	Guida all'installazione dell'applicazione	16
4.1	Prerequisiti	16
4.2	Installazione dipendenze PHP	17
4.3	Configurazione variabili d'ambiente	17
4.4	Migrazioni e seed iniziali	18
4.5	Installazione dipendenze front-end	18
4.6	Avvio dell'applicazione	18
4.7	Production deployment	18

1 Introduzione

Il presente documento illustra l'intero percorso di sviluppo dell'elaborato finale del corso di *Programmazione Web* per l'anno scolastico 2024/2025. Vengono analizzate fasi progettuali, scelte architetture e tecniche adottate, nonché le funzionalità chiave dell'applicativo web. A corredo sono inclusi diagrammi, procedure di installazione e linee guida operative, così da fornire al lettore una base completa per comprendere e utilizzare efficacemente l'applicativo.

1.1 Overview applicativo: Visite Guidate

L'obiettivo della piattaforma è ottimizzare l'incontro tra **domanda** e **offerta** di visite guidate, valorizzando siti di pregio naturale, storico e culturale all'interno di un territorio predefinito. Il sistema può essere adottato da varie organizzazioni locali, prive di scopo di lucro, intenzionate a promuovere la conoscenza del territorio avvalendosi di volontari che prestano la loro opera a titolo gratuito.

1.2 Ruoli e responsabilità

Configuratore Funzione di governance interna all'organizzazione.

- Inserimento e manutenzione di: luoghi, tipologie di visita, parametri di sistema (capienza, periodi di iscrizione, perimetro geografico).
- Gestione del calendario: pubblicazione visite, assegnazione guide, conferme o cancellazioni in base a soglie di partecipazione.
- Supervisione operativa con dashboard riepilogative

Volontario Guida abilitata dall'organizzazione.

- Dichiarazione ricorrente delle disponibilità tramite agenda online.
- Consultazione delle visite assegnate con dettaglio logistico (data, ora, punto di ritrovo, partecipanti).

Fruitore Utente finale, singolo o gruppo.

- Consultazione del catalogo visite
- Prenotazione autonoma e gestione delle proprie iscrizioni.

2 Casi d'Uso

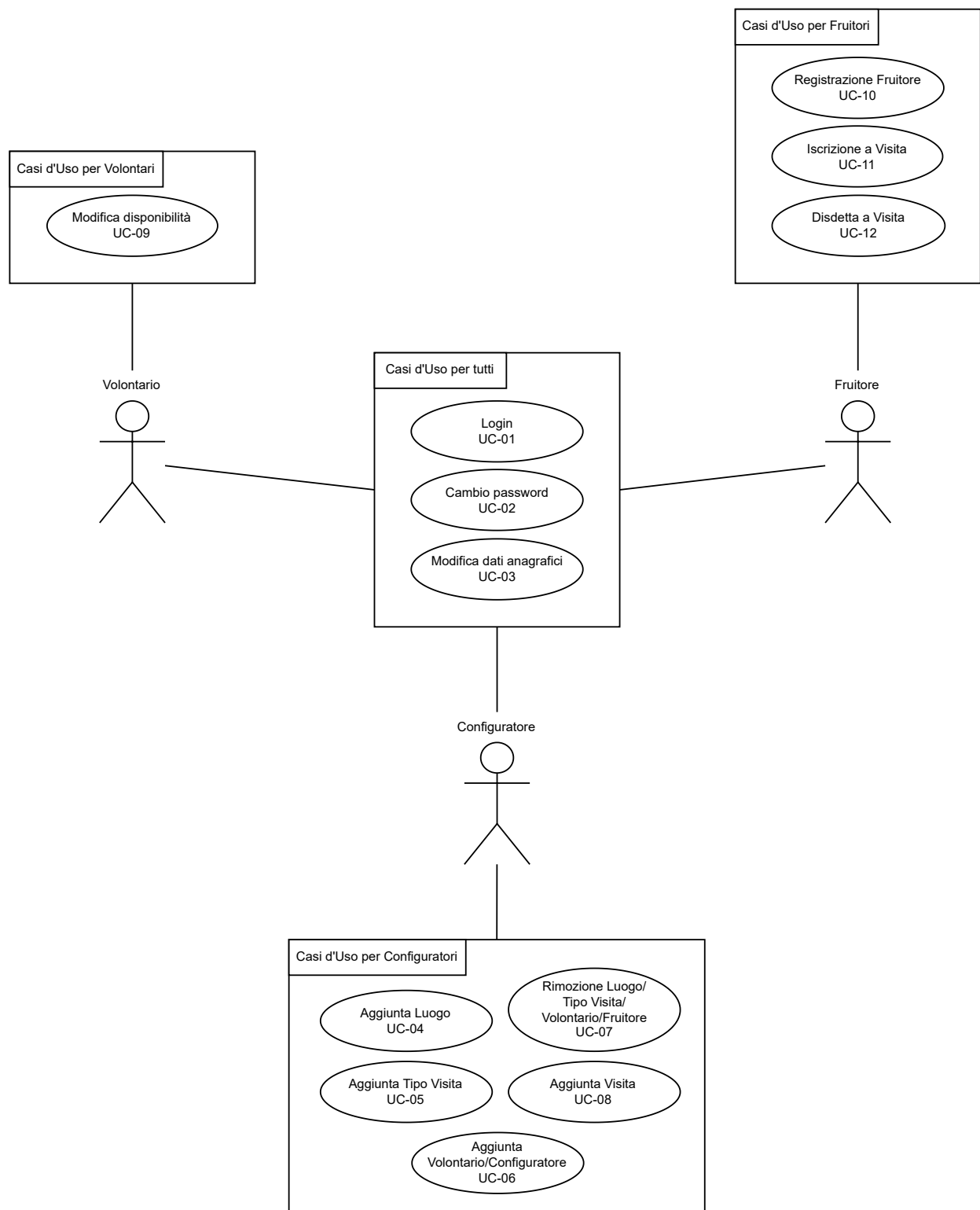


Figura 1: Diagramma dei Casi d'Uso

2.1 Casi d'uso per ogni utente

Use Case: Login (ogni categoria di utente)	
ID	UC-01
Attore(i)	Configuratore, Volontario oppure Fruitore (non autenticato)
Obiettivo	Un utente (configuratore, volontario o fruitore) accede al sistema fornendo le proprie credenziali.
Precondizioni	<ul style="list-style-type: none">• L'utente non è autenticato.• Esistono credenziali per gli utenti. I loro username sono univoci tra tutti gli altri utenti.
Scenario Principale	<ol style="list-style-type: none">1. L'utente avvia la procedura di login.2. L'utente inserisce username e password.3. Il sistema cerca l'utente tra Configuratori, Volontari e Fruitori.4. Se l'username esiste e la password è corretta, il sistema autentica l'utente con il relativo ruolo. <p>Fine.</p>
Scenario Alternativo	4.a. Se si verifica un errore, il sistema lo notifica e torna al passo 2.
Postcondizioni	<ul style="list-style-type: none">• L'utente è autenticato e il suo ruolo (Configuratore/Volontario/Fruitore) è noto al sistema.• L'utente accede al menu principale.

Tabella 1: UC 01

Use Case: Cambio Password (ogni categoria di utente)	
ID	UC-02
Attore(i)	Configuratore, Volontario oppure Fruitore
Obiettivo	L'utente cambia la propria password di accesso personale.
Precondizioni	<ul style="list-style-type: none"> • L'utente è autenticato.
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente seleziona l'opzione per cambiare la password. 2. L'utente inserisce la vecchia password e quella nuova (due volte). 3. Se la password vecchia è corretta, le due password nuove coincidono e soddisfano i criteri di sicurezza, i dati vengono inseriti nel sistema. 4. Il sistema conferma l'avvenuto cambio. <p>Fine.</p>
Scenario Alternativo	<p>4.a. Se una qualsiasi delle condizioni del punto 3 non è verificata, nessun dato è modificato.</p> <p>Fine.</p>
Postcondizioni	<ul style="list-style-type: none"> • L'utente rimane nella schermata di cambio password.

Tabella 2: UC 02

Use Case: Modifica dati anagrafici (ogni categoria di utente)	
ID	UC-03
Attore(i)	Configuratore, Volontario oppure Fruitore
Obiettivo	L'utente aggiunge, rimuove o modifica le proprie informazioni personali nel suo profilo.
Precondizioni	<ul style="list-style-type: none"> • L'utente è autenticato.
Scenario Principale	<ol style="list-style-type: none"> 1. L'utente seleziona l'opzione per modificare i propri dati anagrafici. 2. Il sistema presenta una schermata opportuna. 3. Il volontario inserisce i propri dati: nome, cognome e data di nascita. 4. Il volontario salva le modifiche. 5. Il sistema salva e conferma l'aggiunta. <p>Fine.</p>
Postcondizioni	<ul style="list-style-type: none"> • L'utente rimane nella schermata di modifica dati anagrafici.

Tabella 3: UC 03

2.2 Casi d'uso per Configuratori

Use Case: Aggiunta Luogo	
ID	UC-04
Attore(i)	Configuratore
Obiettivo	Il Configuratore aggiunge un nuovo Luogo visitabile al sistema.
Precondizioni	<ul style="list-style-type: none">• Il Configuratore è autenticato.
Scenario Principale	<ol style="list-style-type: none">1. Il Configuratore seleziona l'opzione per aggiungere un Luogo.2. Il sistema richiede i dati del Luogo:<ul style="list-style-type: none">• Nome/identificativo univoco (es. "Castello del Buonconsiglio").• Descrizione aggiuntiva (opzionale, es. "Antica dimora dei vassalli...").• Collocazione geografica (es. indirizzo o coordinate).3. Il Configuratore inserisce i dati richiesti.4. Il sistema verifica i dati (nome univoco e non vuoto, collocazione non vuota).5. Se corretti, il sistema salva il nuovo Luogo in modo persistente.6. Il sistema conferma l'aggiunta. <p>Fine.</p>
Scenario Alternativo	<ol style="list-style-type: none">5.a. Se una qualsiasi delle condizioni del punto 4 non è verificata, nessun dato è modificato.6.a. Il sistema notifica l'errore.
Postcondizioni	<ul style="list-style-type: none">• Il Configuratore rimane nella schermata di aggiunta Luogo.

Tabella 4: UC 04

Use Case: Aggiunta Tipo Visita	
ID	UC-05
Attore(i)	Configuratore
Obiettivo	Il Configuratore aggiunge un nuovo Tipo Visita al sistema.
Precondizioni	<ul style="list-style-type: none"> Il Configuratore è autenticato.
Scenario Principale	<ol style="list-style-type: none"> Il Configuratore seleziona l'opzione per aggiungere un Tipo Visita. Il sistema richiede i dati del Tipo Visita: <ul style="list-style-type: none"> Luogo a cui collegare il tipo di visita, tra quelli già presenti nel sistema. Titolo univoco (es. "Le segrete del castello"). Descrizione aggiuntiva (opzionale, es. "Un'emozionante esperienza...). Punto di ritrovo (es. indirizzo o coordinate). Ora di inizio e durata. Periodo in cui è istanziabile una visita di questo tipo. Numero minimo e massimo di partecipanti. Se si necessita o meno di comprare un biglietto. Il configuratore inserisce i dati richiesti. Il sistema verifica i dati: nessun campo vuoto tranne eventualmente la descrizione, dati inseriti corretti dal punto di vista logico e secondo il senso comune. Se corretti, il sistema salva il nuovo Tipo Visita in modo persistente. Il sistema conferma l'aggiunta. <p>Fine.</p>
Scenario Alternativo	<ol style="list-style-type: none"> Se una qualsiasi delle condizioni del punto 4 non è verificata, nessun dato è modificato. Il sistema notifica l'errore.
Postcondizioni	<ul style="list-style-type: none"> Il Configuratore rimane nella schermata di aggiunta Tipo Visita.

Tabella 5: UC 05

Use Case: Aggiunta nuovo Volontario/Configuratore	
ID	UC-06
Attore(i)	Configuratore
Obiettivo	Il Configuratore aggiunge un nuovo volontario o configuratore al sistema.
Precondizioni	<ul style="list-style-type: none"> • Il Configuratore è autenticato.
Scenario Principale	<ol style="list-style-type: none"> 1. Il Configuratore seleziona l'opzione per aggiungere un Volontario o Configuratore. 2. Il sistema richiede username e password da assegnare al nuovo utente. 3. Il Configuratore inserisce i dati richiesti. 4. Il sistema verifica i dati: username non già in uso, password sufficientemente sicura e inserita due volte correttamente. 5. Se corretti, il sistema salva il nuovo utente, con il rispettivo ruolo, in modo persistente. I dati potranno essere usati per effettuare il login da parte del nuovo utente. 6. Il sistema conferma l'aggiunta. <p>Fine.</p>
Scenario Alternativo	<p>5.a. Se una qualsiasi delle condizioni del punto 4 non è verificata, nessun dato è modificato.</p> <p>6.a.: Il sistema notifica l'errore.</p>
Postcondizioni	<ul style="list-style-type: none"> • Il Configuratore rimane nella schermata di aggiunta.

Tabella 6: UC 06

Use Case: Rimozione Luogo/Tipo Visita/Volontario/Fruitore	
ID	UC-07
Attore(i)	Configuratore
Obiettivo	Il Configuratore rimuove un dato dal sistema.
Precondizioni	<ul style="list-style-type: none"> • Il Configuratore è autenticato.
Scenario Principale	<ol style="list-style-type: none"> 1. Il Configuratore seleziona l'opzione per rimuovere un Luogo/Tipo Visita/Volontario/Fruitore. 2. Il sistema controlla se l'eliminazione del dato comporta inconsistenze nel sistema. 3. Se non ci sono inconsistenze, il dato viene correttamente rimosso. 4. Il sistema conferma la rimozione. <p>Fine.</p>
Scenario Alternativo	<ol style="list-style-type: none"> 3.a. Se vengono rilevate inconsistenze, vengono rimossi i dati interessati: <ul style="list-style-type: none"> • L'eliminazione di un Luogo comporta la rimozione dei Tipi Visita associati. • L'eliminazione di un Tipo Visita comporta la rimozione delle Visite associate. • L'eliminazione di un Volontario comporta la rimozione delle Visite associate. • L'eliminazione di un Fruitore comporta la rimozione delle iscrizioni a lui associate. 4.a. Il sistema notifica l'errore.
Postcondizioni	<ul style="list-style-type: none"> • Il Configuratore rimane nella schermata di rimozione.

Tabella 7: UC 07

Use Case: Aggiunta Visita	
ID	UC-08
Attore(i)	Configuratore
Obiettivo	Il Configuratore aggiunge una nuova Visita al sistema.
Precondizioni	<ul style="list-style-type: none"> • Il configuratore è autenticato.
Scenario Principale	<ol style="list-style-type: none"> 1. Il Configuratore seleziona l'opzione per aggiungere una nuova Visita. 2. Il sistema richiede i dati della Visita: Tipo Visita, data e Volontario designato. 3. Il Configuratore inserisce i dati richiesti. 4. Il sistema verifica che non ci siano campi vuoti. 5. Se corretti, il sistema salva la nuova Visita. 6. Il sistema conferma l'aggiunta. <p>Fine.</p>
Scenario Alternativo	<ol style="list-style-type: none"> 5.a. Se si verifica un errore nel punto 4, nessun dato è modificato. 6.a. Il sistema notifica l'errore.
Postcondizioni	<ul style="list-style-type: none"> • Il Configuratore rimane nella schermata di aggiunta Visita.

Tabella 8: UC 08

2.3 Casi d'uso per Volontari

Use Case: Modifica disponibilità	
ID	UC-09
Attore(i)	Volontario
Obiettivo	Il Volontario aggiunge o rimuove giorni di disponibilità per il mese successivo.
Precondizioni	<ul style="list-style-type: none">• Il Volontario è autenticato.
Scenario Principale	<ol style="list-style-type: none">1. Il Volontario seleziona l'opzione per modificare le proprie disponibilità per il mese successivo.2. Il sistema presenta una schermata opportuna con i giorni selezionabili.3. Il Volontario inserisce i propri dati.4. Il Volontario salva le modifiche.5. Il sistema salva e conferma l'aggiunta. Fine.
Postcondizioni	<ul style="list-style-type: none">• Il configuratore rimane nella schermata di aggiunta Visita.

Tabella 9: UC 09

2.4 Casi d'uso per Fruitori

Use Case: Registrazione Fruitore	
ID	UC-10
Attore(i)	Fruitore (non autenticato)
Obiettivo	Un Fruitore aggiunge al sistema le proprie nuove credenziali, con cui potrà effettuare il login.
Precondizioni	<ul style="list-style-type: none">• Il Fruitore non è autenticato.
Scenario Principale	<ol style="list-style-type: none">1. Il Fruitore avvia la procedura di registrazione.2. Il Fruitore inserisce i propri username e password (due volte per conferma).3. Se le due password coincidono e soddisfano i criteri di sicurezza, e lo username non appartiene già ad un altro utente, i dati vengono inseriti nel sistema. <p>Fine.</p>
Scenario Alternativo	3.a. Se una qualsiasi delle condizioni del punto 3 non è verificata, torna al passo 2.
Postcondizioni	<ul style="list-style-type: none">• Il Fruitore può poi effettuare il login.

Tabella 10: UC 10

Use Case: Iscrizione a Visita	
ID	UC-11
Attore(i)	Fruitore
Obiettivo	Il Fruitore iscrive se stesso ed eventualmente altre persone ad una Visita.
Precondizioni	<ul style="list-style-type: none"> • Il Fruitore è autenticato.
Scenario Principale	<ol style="list-style-type: none"> 1. Il Fruitore seleziona l'opzione per iscriversi ad una visita proposta. 2. Il sistema richiede il numero di persone da iscrivere (compreso il titolare della prenotazione). 3. Il Fruitore inserisce il dato richiesto. 4. Il sistema verifica che il numero sia maggiore o uguale a uno e che non ecceda la capacità rimanente della Visita. 5. Se corretto, il sistema salva la prenotazione e consegna un codice univoco associato ad essa. <p>Fine.</p>
Scenario Alternativo	<ol style="list-style-type: none"> 5.a. Se una delle condizioni del punto 4 non è verificata, nessun dato è modificato. 6.a. Il sistema notifica l'errore.
Postcondizioni	<ul style="list-style-type: none"> • Il Fruitore rimane nella schermata delle Visite disponibili.

Tabella 11: UC 11

Use Case: Disdetta a Visita	
ID	UC-12
Attore(i)	Fruitore
Obiettivo	Il Fruitore elimina una propria iscrizione ad una Visita.
Precondizioni	<ul style="list-style-type: none"> • Il fruitore è autenticato.
Scenario Principale	<ol style="list-style-type: none"> 1. Il Fruitore seleziona l'opzione per disdire una sua prenotazione. 2. Il sistema chiede conferma. 3. Se confermato, il sistema elimina la prenotazione e aggiorna di conseguenza il numero di iscritti alla Visita in questione. <p>Fine.</p>
Scenario Alternativo	3.a. Se non confermato, nessun dato è modificato.
Postcondizioni	<ul style="list-style-type: none"> • Il Fruitore rimane nella schermata delle Visite.

Tabella 12: UC 12

3 Infrastruttura informatica

L'applicazione è stata sviluppata utilizzando il framework **Laravel** in versione 12, con il supporto del package **spatie/laravel-permission** per la gestione di ruoli e permessi. L'ambiente di sviluppo utilizza **PHP 8.2** e si basa su un database **SQLite**, il cui file è posizionato localmente all'interno della directory `database/database.sqlite`. L'autenticazione, la gestione delle sessioni e i driver di cache sono configurati per utilizzare il file system locale.

Per l'interfaccia utente, è stato adottato **Bootstrap 5** con il sistema di build e asset management gestito tramite **Vite**. La configurazione comprende anche **bootstrap-icons** e il supporto per **Sass** per la compilazione degli stili.

3.1 Diagramma ER database

Di seguito viene riportato il diagramma ER delle tabelle peculiari all'applicazione, omettendo quelle amministrative secondarie (permessi, cache...).

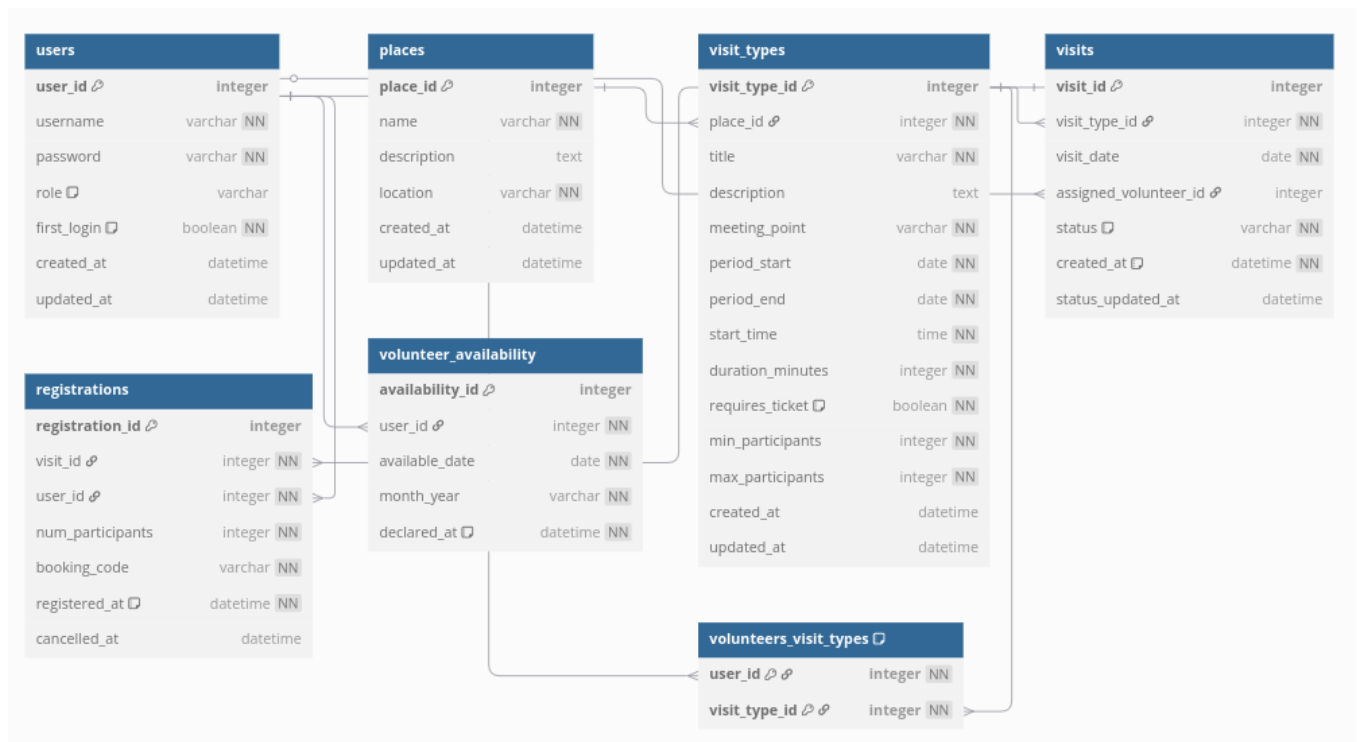


Figura 2: Diagramma ER

4 Guida all'installazione dell'applicazione

4.1 Prerequisiti

L'applicazione è stata sviluppata in ambiente UNIX, di conseguenza la guida all'installazione farà riferimento a tale ambiente.

- **PHP** ≥ 8.2
- **Composer** ≥ 2.7 (dependency manager PHP).
- **Database**: creare il file vuoto `database/database.sqlite`
- **Node.js** ≥ 18 LTS e **npm** ≥ 9 (build asset front-end con Vite).

4.2 Installazione dipendenze PHP

```
composer install --prefer-dist --no-interaction --optimize-autoloader
```

4.3 Configurazione variabili d'ambiente

1. Copiare il file esempio:

```
cp .env.example .env
```

2. Generare la chiave d'applicazione:

```
php artisan key:generate
```

3. Aprire .env e impostare le configurazioni desiderate. Di seguito un esempio di configurazione per esecuzione locale:

```
APP_NAME=VisiteGuidate
APP_ENV=local
APP_KEY=base64:<YOUR_KEY_HERE>
APP_DEBUG=false
APP_URL=http://localhost

APP_LOCALE=en
APP_FALLBACK_LOCALE=en
APP_FAKER_LOCALE=en_US

APP_MAINTENANCE_DRIVER=file

PHP_CLI_SERVER_WORKERS=4

BCRYPT_ROUNDS=12

LOG_CHANNEL=stack
LOG_STACK=single
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=sqlite
DB_DATABASE=database/database.sqlite

SESSION_DRIVER=file
SESSION_LIFETIME=120
SESSION_ENCRYPT=false
SESSION_PATH=/
SESSION_DOMAIN=null

BROADCAST_CONNECTION=log
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync

CACHE_STORE=file

MEMCACHED_HOST=127.0.0.1

VITE_APP_NAME="${APP_NAME}"
```

4.4 Migrazioni e seed iniziali

Se si volesse popolare il database con dei dati predefiniti, eseguire il comando:

```
php artisan migrate --seed
```

4.5 Installazione dipendenze front-end

```
npm ci          # installa package-lock esistente
npm run build    # oppure 'npm run dev' per sviluppo con HMR
```

4.6 Avvio dell'applicazione

Ambiente locale

```
php artisan serve --host=0.0.0.0 --port=8000
# ora disponibile su http://localhost:8000
```

4.7 Production deployment

Nel caso si volesse preparare l'applicazione a un ambiente di produzione, adeguare il file di *environment* con le proprie configurazioni ed eseguire i seguenti comandi:

```
php artisan config:cache
php artisan route:cache
php artisan view:cache
php artisan optimize          # combina i passaggi precedenti
```

Risulta ora possibile servire l'applicazione tramite un server HTTP quale Nginx.