Software Requirements Specification

Piattaforma organizzazione matrimonio

Versione 1.0

Realizzato da Daniele Capasso

<Università degli studi della Campania Luigi Vanvitelli>

Esame di Sistemi Web e Basi Dati

Tabella dei Contenuti

Tabella dei contenuti i

1. Introduzione 1

1.1 Scopo 1

1.2 Convenzioni del documento 1

1.3 Pubblico Destinatario 1

1.4 Scopo del prodotto 1

2. Descrizione Generale 2

2.1 Prospettiva del prodotto 2

2.2 Funzioni del prodotto 2

2.3 Classi e caratteristiche degli utenti 2

2.4 Ambiente operativo 2

2.5 Vincoli di progettazione e implementazione 2

3. Requisiti delle Interfacce Esterne 3

3.1 Interfacce utente 3

3.2 Interfacce hardware 4

3.3 Interfacce software 4

3.4 Interfacce di comunicazione 5

4. Funzionalità del Sistema 5

4.1 Gestione della lista degli invitati 5

4.2 Gestione della timeline degli eventi 6

4.3 Monitoraggio del budget 6

4.4 Gestione degli utenti e dei ruoli 7

4.5 Notifiche e gestione delle email 8

5. Altri Requisiti Non Funzionali 9

5.1 Requisiti di prestazione 9

5.2 Requisiti di sicurezza 9

5.3 Attributi di qualità del software 9

6. Modelli di Analisi 9

6.1 Modello E-R 10

6.2 Traduzione del modello E-R nel modello logico relazionale 10

6.3 Progettazione logica database 11

6.4 Diagramma dei casi d'uso 13

7. Test 14

7.1 Test con PHPUnit 14

7.2 Altri test 15

# Introduzione

## Scopo

Questo documento specifica i requisiti software per la **Piattaforma per Organizzazione Matrimonio**. Il sistema è una web application finalizzata ad aiutare le coppie nella pianificazione e organizzazione del proprio matrimonio. In particolare, il software offre funzionalità per la gestione della lista degli invitati, la creazione e visualizzazione di una timeline degli eventi (cerimonia, ricevimenti, ecc.) e il monitoraggio del budget, consentendo di registrare spese e controllarne l’andamento. Questo SRS copre l’intero sistema e descrive in dettaglio sia i requisiti funzionali che quelli non funzionali.

## Convenzioni del documento

In questo SRS i requisiti sono indicati mediante etichette univoche per facilitarne il riferimento. Inoltre, ad ogni requisito sarà associato un livello di priorità.

## Pubblico Destinatario

Essendo il sistema progettato per supportare l'organizzazione di matrimoni, i principali utenti dell’applicazione sono: **Coppie di sposi** che rappresentano gli utenti principali del sistema, i **Wedding Planner**  che possono essere autorizzati dalle coppie a gestire l’evento ed infine gli **invitati** al matrimonio.

## Scopo del prodotto

La **Piattaforma per Organizzazione Matrimonio** è un software web che supporta le coppie nel pianificare il proprio matrimonio. Questo software consente di avere un unico strumento centralizzato che permette di gestire invitati, eventi e budget; permette dunque il controllo in tempo reale dello stato degli inviti, di tutti gli eventi e dell’andamento delle spese.

# Descrizione generale

## Prospettiva del prodotto

La piattaforma è un nuovo prodotto autonomo ed è concepito come strumento indipendente dedicato esclusivamente all’organizzazione dei matrimoni.

Di seguito i componenti principali della piattaforma:

* **Piattaforma Organizzazione Matrimonio:** Rappresenta l’intero sistema che integra tutte le funzionalità per organizzare un matrimonio.
* **Interfaccia Utente (UI):** La parte front-end accessibile tramite browser.
* **Backend:** Il “cuore” del sistema che gestisce la logica applicativa, ricevendo le richieste dall'interfaccia utente e comunicando con gli altri componenti.
* **Database:** Il sistema relazionale (MySQL) dove vengono salvati tutti i dati relativi a utenti, invitati, eventi e budget.
* **Servizio Email:** Un servizio per l’invio di email di inviti.

## Funzioni del prodotto

Di seguito viene riportata una panoramica generale delle funzionalità.

Per una visione più dettagliata, si consulti la sezione quattro di questo documento.

Il sistema quindi dovrà consentire:

* **Gestione della Lista degli Invitati**: aggiunta, modifica, eliminazione e filtraggio degli invitati, con possibilità di raggrupparli in categorie (famiglia, amici, colleghi, ecc.).
* **Creazione e Gestione della Timeline degli Eventi**: pianificazione di cerimonie, ricevimenti e altri eventi correlati.
* **Monitoraggio del Budget**: definizione di un budget totale, registrazione delle spese e visualizzazione dell’andamento delle spese.
* **Gestione degli Utenti e dei Ruoli**: autenticazione e autorizzazione alla gestione delle funzionalità attraverso l’implementazione di ruoli differenziati (coppia, wedding planner, invitati).
* **Integrazione e Notifiche**: integrazione con un sistema di email per gestire gli inviti, oltre a notifiche interne per aggiornamenti sugli invitati in tempo reale.

## Classi e caratteristiche degli utenti

Questa piattaforma è destinata ad essere utilizzata da:

* **Coppie Sposanti e Wedding Planner**: utilizzatori principali, con necessità di un’interfaccia semplice e intuitiva per gestire l’intero evento. In particolare, il wedding planner ottiene gli stessi poteri della coppia attraverso un link di invito fornito dagli sposi.
* **Invitati**: Familiari, amici, colleghi e altri partecipanti al matrimonio, con accesso limitato per confermare la partecipazione e visualizzare informazioni relative all’evento.

## Ambiente operativo

La **Piattaforma per Organizzazione Matrimonio** opera in un ambiente web utilizzando **XAMPP 8.2.12**. In particolare:

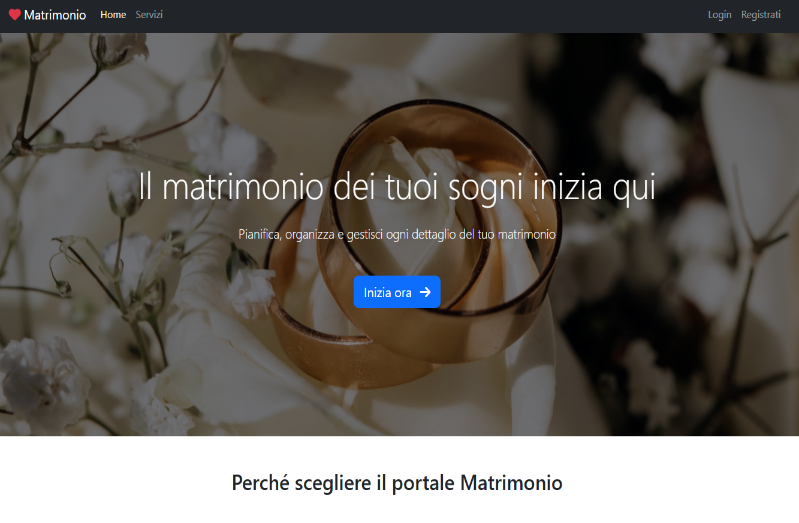
* **Server Web:**  
  Il sistema è eseguito su **Apache 2.4.58**, incluso in XAMPP, che gestisce le richieste HTTP provenienti dai client.
* **Database:**  
  Il database relazionale utilizzato è **MySQL Ver 15.1 Distrib 10.4.32-MariaDB**, in cui vengono memorizzati i dati relativi a utenti, invitati, eventi e budget.

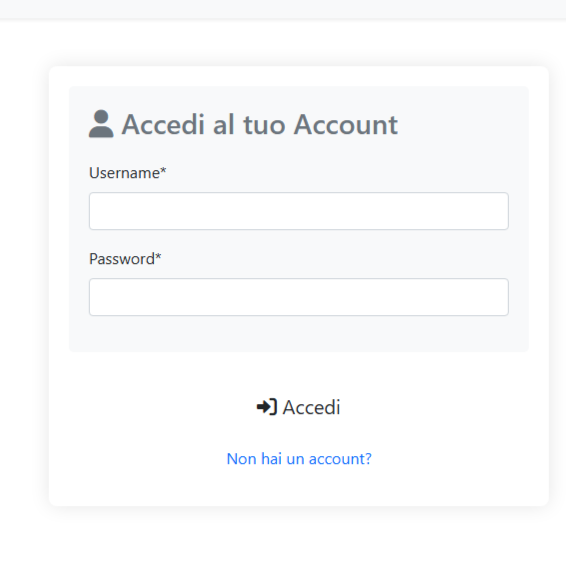
## Vincoli di progettazione e implementazione

Nella progettazione e nell’implementazione viene utilizzato il framework Bootstrap per garantire un'interfaccia coerente e intuitiva per l’utente permettendo di alleggerire lo sviluppo front-end. Inoltre, la gestione dei ruoli è basata su un campo ID\_Ruolo con valori fissi (1 per Coppia, 2 per Wedding Planner, 3 per Invitato).

# Requisiti delle Interfacce Esterne

## Interfacce utente

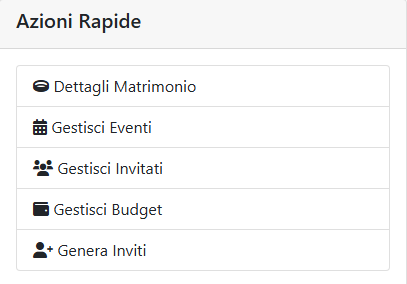
Il portale "Matrimonio" offre un'esperienza utente intuitiva adatto a tutti i tipi di utenti. La homepage iniziale offre una panoramica sulle funzionalità principali e sui benefici per l'utente e da qui si possono consultare tutti i servizi ed effettuare il login o la registrazione.

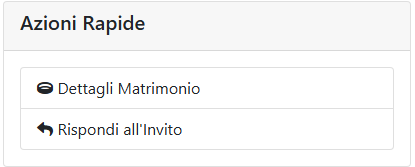
Una volta effettuato l'accesso, gli utenti possono navigare tra le diverse sezioni del portale tramite un menu di navigazione laterale differenziato per ruolo: le coppie e i wedding planner hanno accesso a tutte le funzioni di gestione del matrimonio, mentre gli invitati hanno accesso limitato che consente loro di visualizzare solo le informazioni relative al matrimonio e agli eventi ed utilizzare una sezione dedicata per rispondere all'invito.

Pagina iniziale

Dal punto di vista grafico, l'interfaccia con Bootstrap e Font Awesome garantiscono un design moderno e coerente.

login





Menu per gli invitati

Menu per coppia e wedding planner

## Interfacce hardware

Il portale "Matrimonio" è un'applicazione web-based che non richiede hardware specifico.

## Interfacce software

Il sistema si basa su tecnologie standard per garantire compatibilità; l'unico requisito minimo è l'uso di un browser moderno (es. Chrome, Firefox, Edge…) che supporti HTML5, CSS3 e JavaScript.

L’applicazione si basa su XAMPP che integra Apache per la gestione delle richieste HTTP, MySQL per l’archiviazione e la gestione dei dati, PHP per la logica lato server.

## Interfacce di comunicazione

Il sistema utilizza il protocollo standard HTTP per garantire un flusso di dati efficiente tra client e server. Inoltre, le sessioni PHP gestiscono l'autenticazione degli utenti, mantenendo lo stato tra le diverse pagine dell'applicazione.

# Funzionalità del Sistema

Di seguito vengono illustrate le principali funzionalità che il sistema deve offrire:

## Funzionalità: Gestione della lista degli invitati

4.1.1 **Descrizione e Priorità**

* **Descrizione**: Permettere all’utente (tipicamente la coppia o il wedding planner) di aggiungere, modificare, eliminare e raggruppare gli invitati per stato di conferma e categoria.
* **Priorità**: Alta

4.1.2 **Sequenze Stimolo/Risposta**

* **Sequenza Aggiunta Invitato**:
  1. L’utente seleziona “Nuovo Invitato”.
  2. Il sistema mostra un form con campi per nome, email, telefono, categoria e stato di conferma.
  3. L’utente compila il form e preme “Salva”.
  4. Il sistema registra l’invitato e aggiorna la lista.

4.1.3 **Requisiti Funzionali**

* **REQ-GI-1**: Il sistema deve consentire la creazione, modifica ed eliminazione di un invitato.
* **REQ-GI-2**: Il sistema deve permettere di raggruppare gli invitati in categorie (es. Famiglia, Amici, Colleghi).
* **REQ-GI-3**: Il sistema deve fornire una funzione di ricerca e filtro per invitati in base a categoria o stato di conferma.

## Funzionalità: Gestione della timeline degli eventi

4.2.1 **Descrizione e Priorità**

* **Descrizione**: Consente di creare e gestire una timeline degli eventi relativi al matrimonio, includendo dettagli quali nome, data, ora, luogo e descrizione.
* **Priorità**: Alta

4.2.2 **Sequenze Stimolo/Risposta**

* **Sequenza Creazione Evento**:
  1. L’utente seleziona “Nuovo Evento” dal menu della timeline.
  2. Il sistema mostra un form con campi per nome, data, orario, luogo, descrizione.
  3. L’utente inserisce i dati e conferma l’operazione.
  4. Il sistema registra l’evento e lo visualizza nella timeline.

4.2.3 **Requisiti Funzionali**

* **REQ-TE-1**: Il sistema deve consentire la creazione, modifica ed eliminazione degli eventi.
* **REQ-TE-2**: Il sistema deve permettere l’inserimento dei dettagli (nome, data, ora, luogo, descrizione) per ogni evento.
* **REQ-TE-3**: Il sistema deve offrire una visualizzazione degli eventi in lista cronologica.

## Funzionalità: Monitoraggio del budget

4.3.1 **Descrizione e Priorità**

* **Descrizione**: Consente di definire un budget totale, suddividerlo per categorie (es. catering, location, abbigliamento, decorazioni) e registrare le spese sostenute, visualizzando report e avvisi in caso di superamento delle soglie impostate.
* **Priorità**: Alta

4.3.2 **Sequenze Stimolo/Risposta**

* **Sequenza Registrazione Spesa**:
  1. L’utente accede alla sezione “ Gestisci Budget ”.
  2. Il sistema mostra un form per inserire una spesa, indicando importo, data e descrizione.
  3. L’utente inserisce le informazioni e preme “Salva Spesa”.
  4. Il sistema aggiorna il totale speso e, se necessario, invia un avviso di superamento della soglia del budget.

4.3.3 **Requisiti Funzionali**

* **REQ-BU-1**: Il sistema deve consentire di definire un budget totale.
* **REQ-BU-2**: Il sistema deve permettere l’inserimento delle spese sostenute, associandole a data, importo e descrizione.
* **REQ-BU-3**: Il sistema deve generare report che illustrino l’andamento delle spese rispetto al budget impostato.
* **REQ-BU-4**: Il sistema deve inviare un avviso in caso di superamento di soglie predefinite.

## Funzionalità: Gestione degli utenti e dei ruoli

4.4.1 **Descrizione e Priorità**

* **Descrizione**: L’applicativo fornisce un sistema di autenticazione e autorizzazione con ruoli differenziati (coppia, wedding planner, invitati), per garantire l’accesso sicuro e la gestione dei permessi.
* **Priorità**: Alta

4.4.2 **Sequenze Stimolo/Risposta**

* **Sequenza Login**:
  1. L’utente accede alla pagina di login.
  2. Il sistema verifica le credenziali e, in caso di esito positivo, reindirizza l’utente alla dashboard personalizzata in base al ruolo.
  3. In caso di credenziali errate, il sistema visualizza un messaggio di errore.

4.4.3 **Requisiti Funzionali**

* **REQ-UT-1**: Il sistema deve implementare un meccanismo di autenticazione sicura (username e password).
* **REQ-UT-2**: Il sistema deve gestire i ruoli utente e i relativi privilegi di accesso.
* **REQ-UT-3**: Il sistema deve poter invitare nuovi utenti alla registrazione alla piattaforma generando inviti univoci tramite link di invito.

## Funzionalità: Notifiche e gestione delle email

4.5.1 **Descrizione e Priorità**

* **Descrizione**: Il sistema deve integrarsi con un servizio per l’invio di email (Mercury) per l’invio di token di invito oltre a fornire notifiche in tempo reale per aggiornamenti sulla lista degli invitati. (**Nel contesto di questo progetto, l'invio delle email avverrà esclusivamente in locale, senza effettiva spedizione verso destinatari esterni).**
* **Priorità**: Media

4.5.2 **Sequenze Stimolo/Risposta**

* **Sequenza Notifica**:
  1. Al momento della creazione di un nuovo account o della risposta di partecipazione di un invitato, il sistema genera una notifica.
  2. La notifica viene inviata agli utenti interessati tramite notifiche interne.

4.5.3 **Sequenze Stimolo/Risposta**

* **Sequenza Invio Email**:
  1. Quando viene creato un invito, il sistema genera un token.
  2. L'email con il token di invito viene preparata e registrata localmente, senza effettiva spedizione verso destinatari esterni.
  3. L'utente attraverso il link nell'email può completare il processo di registrazione e di risposta della propria partecipazione.

4.5.4 **Requisiti Funzionali**

* **REQ-NO-1**: Il sistema deve visualizzare notifiche interne in tempo reale per aggiornamenti relativi alla partecipazione degli invitati.
* **REQ-NO-2**: Il sistema deve garantire che le notifiche siano immediatamente visibili nell’interfaccia utente.
* **REQ-EM-1**: Il sistema deve integrare un modulo per la creazione e l’invio locale di email contenenti i link di invito, tenendo traccia delle email inviate ai fini di test.

# Altri Requisiti Non Funzionali

## Requisiti di prestazione

La piattaforma deve essere scalabile e in grado di gestire un numero crescente di utenti senza degradare le performance.

## Requisiti di sicurezza

Per garantire maggior sicurezza, le password vengono e memorizzate in maniera sicura attraverso la funzione password\_hash() di PHP con la costante PASSWORD\_DEFAULT.

Vengono inoltre utilizzati i prepared statement dell'estensione MySQLi di PHP per rendere più sicure le query ed evitare gli attacchi basati sulle SQL injection.

## Attributi di qualità del software

* **Usabilità**: Interfaccia intuitiva e user-friendly.
* **Portabilità**: L’applicazione deve funzionare correttamente su diversi browser.

# Modelli di Analisi

* Progettazione concettuale del database mediante il modello E-R.
* Traduzione dello schema E-R nel modello logico relazionale.
* Progettazione logica del database.
* Diagramma dei casi d’uso.

## Modello E-R

## Traduzione del modello E-R nel modello logico relazionale

MATRIMONIO(**ID\_Matrimonio**, data, location, descrizione)

EVENTO(**ID\_Evento**, nome, data, ora, descrizione, location, *ID\_Matrimonio*\*)

BUDGET(**ID\_Budget**, importo\_totale, *ID\_Matrimonio*\*)

SPESA(**ID\_Spesa**, importo, descrizione, data, *ID\_Budget*\*)

INVITATO(**ID\_Invitato**, nome, cognome, email, telefono, conferma, *ID\_Utente*\**, ID\_Matrimonio\*, ID\_Categoria\*)*

CATEGORIA(**ID\_Categoria**, nome)

UTENTE(**ID\_Utente**, email, username, password)

PARTECIPAZIONE(**ID\_Partecipazione**, *ID\_Matrimonio\*, ID\_Utente\*,ID\_Ruolo\*)*

RUOLO(**ID\_Ruolo**, nome)

NOTIFICA(**ID\_Notifica**, messaggio, data\_ora, stato, *ID\_Utente\**)

## Progettazione logica del database

Di seguito viene elencata la struttura del database:

CREATE TABLE MATRIMONIO (

ID\_Matrimonio INT AUTO\_INCREMENT PRIMARY KEY,

data DATE NOT NULL,

location VARCHAR(100) NOT NULL,

descrizione VARCHAR(255) NOT NULL

);

CREATE TABLE EVENTO (

ID\_Evento INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(50) NOT NULL,

data DATE NOT NULL,

ora TIME,

descrizione VARCHAR(255),

location VARCHAR(100),

ID\_Matrimonio INT NOT NULL,

FOREIGN KEY (ID\_Matrimonio) REFERENCES MATRIMONIO(ID\_Matrimonio)

);

CREATE TABLE BUDGET (

ID\_Budget INT AUTO\_INCREMENT PRIMARY KEY,

importo\_totale DECIMAL(10,2) NOT NULL DEFAULT 0.00,

ID\_Matrimonio INT NOT NULL UNIQUE,

FOREIGN KEY (ID\_Matrimonio) REFERENCES MATRIMONIO(ID\_Matrimonio)

);

CREATE TABLE SPESA (

ID\_Spesa INT AUTO\_INCREMENT PRIMARY KEY,

importo DECIMAL(10,2) NOT NULL DEFAULT 0.00,

descrizione VARCHAR(255),

data DATE,

ID\_Budget INT NOT NULL,

FOREIGN KEY (ID\_Budget) REFERENCES BUDGET(ID\_Budget)

);

CREATE TABLE CATEGORIA (

ID\_Categoria INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(50) NOT NULL

);

CREATE TABLE INVITATO (

ID\_Invitato INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(50) NOT NULL,

cognome VARCHAR(50),

email VARCHAR(100),

telefono VARCHAR(20),

conferma ENUM('in attesa', 'confermato', 'rifiutato') NOT NULL DEFAULT 'in attesa';

ID\_Utente INT NULL UNIQUE,

ID\_Matrimonio INT NOT NULL,

ID\_Categoria INT NOT NULL,

FOREIGN KEY (ID\_Utente) REFERENCES UTENTE(ID\_Utente),

FOREIGN KEY (ID\_Matrimonio) REFERENCES MATRIMONIO(ID\_Matrimonio),

FOREIGN KEY (ID\_Categoria) REFERENCES CATEGORIA(ID\_Categoria)

);

CREATE TABLE UTENTE (

ID\_Utente INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

email VARCHAR(100) NOT NULL

);

CREATE TABLE NOTIFICA (

ID\_Notifica INT AUTO\_INCREMENT PRIMARY KEY,

messaggio VARCHAR(2000) NOT NULL,

data\_ora DATETIME DEFAULT CURRENT\_TIMESTAMP,

stato ENUM('non\_letto', 'letto') NOT NULL DEFAULT 'non\_letto',

ID\_Utente INT NOT NULL,

FOREIGN KEY (ID\_Utente) REFERENCES UTENTE(ID\_Utente)

);

CREATE TABLE RUOLO (

ID\_Ruolo INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(50) NOT NULL

);

CREATE TABLE PARTECIPAZIONE (

ID\_Partecipazione INT AUTO\_INCREMENT PRIMARY KEY,

ID\_Matrimonio INT NOT NULL,

ID\_Utente INT NOT NULL,

ID\_Ruolo INT NOT NULL,

FOREIGN KEY (ID\_Matrimonio) REFERENCES MATRIMONIO(ID\_Matrimonio),

FOREIGN KEY (ID\_Utente) REFERENCES UTENTE(ID\_Utente),

FOREIGN KEY (ID\_Ruolo) REFERENCES RUOLO(ID\_Ruolo)

);

Inoltre è presente una tabella per la gestione dei token di inviti:

CREATE TABLE INVITE\_TOKEN (

  id INT AUTO\_INCREMENT PRIMARY KEY,

  token VARCHAR(255) NOT NULL,

  wedding\_id INT NOT NULL,

  role INT NOT NULL,

  expires\_at DATETIME NOT NULL,

  used TINYINT(1) DEFAULT 0,

  FOREIGN KEY (wedding\_id) REFERENCES MATRIMONIO(ID\_Matrimonio),

  FOREIGN KEY (role) REFERENCES RUOLO(ID\_Ruolo)

);

INSERT INTO RUOLO (nome) VALUES ('Coppia'), ('Wedding Planner'), ('Invitato');

INSERT INTO CATEGORIA (nome) VALUES ('Famiglia'), ('Amici'), ('Colleghi'), ('Altro');

## Immagine che contiene testo, diagramma, schermata, linea Il contenuto generato dall'IA potrebbe non essere corretto.Diagramma dei casi d’uso

# Test

## Test con PHPUnit

In questa sezione vengono riportati i vari casi di test effettuati con PHPUnit, che permettono di verificare il corretto funzionamento delle principali funzionalità dell’applicazione. Questi test coprono le aree di registrazione, login (con username), gestione del budget, lista degli invitati e gestione degli eventi.

**Test di Registrazione e Login**  
Per quanto riguarda la registrazione, sono stati sviluppati i seguenti test:

* **testSuccessfulRegistration()**: verifica che un utente venga registrato correttamente, assicurando che tutti i dati forniti siano validi.
* **testDuplicateUsername()** verifica che il sistema non permetta la registrazione con un username già esistente, grazie al vincolo UNIQUE sul database.

Nel processo di login, l’autenticazione avviene tramite username, per cui sono stati definiti i seguenti casi di test:

* **testLoginSuccess()**: verifica che l’utente possa accedere correttamente quando vengono inseriti username e password validi.
* **testLoginFailure()**: assicura che il sistema non permetta l'accesso con un username inesistente. Ci aspettiamo che il test passi e che poi venga stampato un messaggio di conferma per comunicare che non è stato trovato nessun username.

**Test sulla Gestione dei Dati**

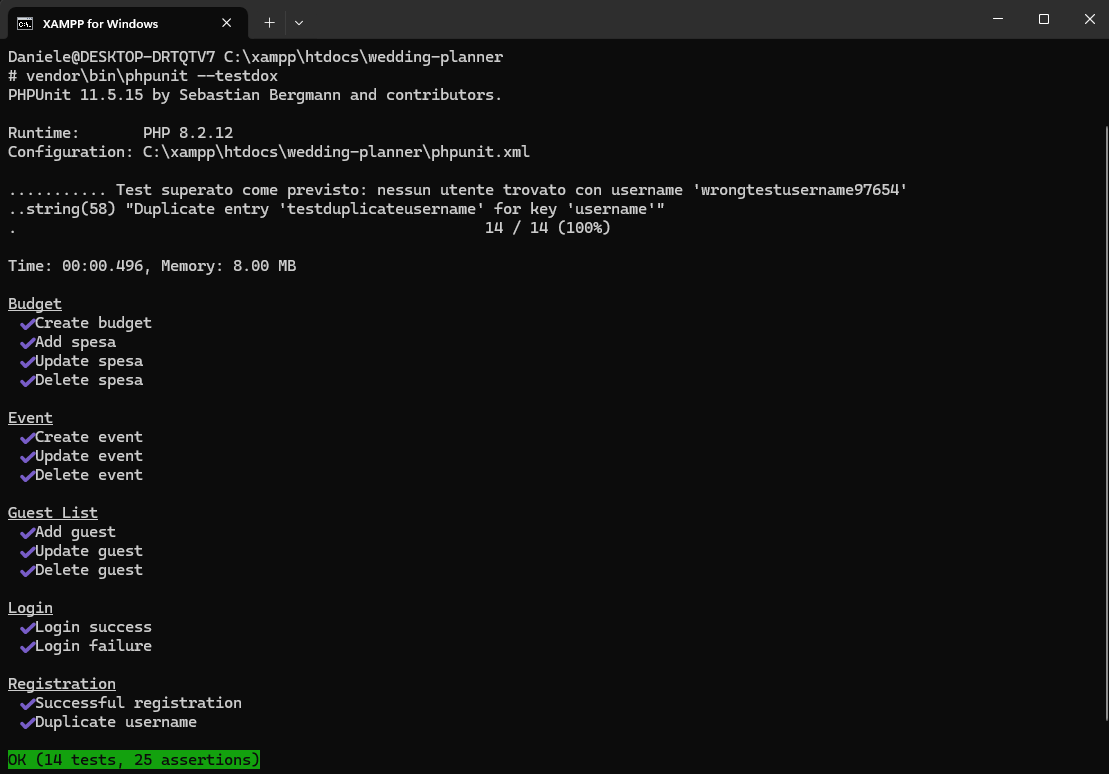
Le funzionalità relative alla gestione del budget, degli invitati e degli eventi condividono una logica di base nei test. Questi test assicurano che tutto funzioni come previsto, evitando problemi nei dati.

In particolare, vengono verificate tre operazioni fondamentali:

* Creazione: verifica che l'inserimento di un nuovo elemento (budget, spesa, invitato, evento) avvenga correttamente.
* Modifica: assicura che un elemento esistente possa essere aggiornato senza errori.
* Eliminazione: verifica che l'elemento venga rimosso dal database.

Dunque, le funzioni implementate per i test sono:

* Budget(**testCreateBudget(), testAddSpesa(), testUpdateSpesa(), testDeleteSpesa()**)
* Lista Invitati(**testAddGuest(), testUpdateGuest(), testDeleteGuest()**)
* Eventi(**testCreateEvent(), testUpdateEvent(), testDeleteEvent()**)

****

## Altri test

|  |  |  |
| --- | --- | --- |
| AZIONE | COMPORTAMENTO ATTESO | OSSERVAZIONE |
| L'utente lascia i campi vuoti durante la fase di login | Viene negato l'accesso | Il sistema nega l'accesso mostrando un messaggio di errore esplicativo del primo step dell'autenticazione |
| L'utente crea un matrimonio oppure inserisce una spesa o programma un evento con una data passata | L’utente viene avvisato di aver inserito una data nel passato | Il test verifica che l'utente venga avvisato che la data scelta non è coerente (è nel passato) permettendogli di confermare o annullare l'azione, prevenendo errori di inserimento. Viene visualizzato il messaggio "Attenzione: la data selezionata è nel passato. Sei sicuro di voler continuare?" |
| L'utente tenta di creare un matrimonio senza inserire la location, la data o la descrizione | Viene negato l’operazione e l’utente viene avvisato | Il test verifica che il sistema controlli la compilazione dei campi essenziali per la creazione di un matrimonio. Se uno dei campi obbligatori non viene inserito, l'applicazione blocca l'operazione e notifica l'utente dell'errore |
| L'utente modifica correttamente i dettagli di un matrimonio già registrato | L’aggiornamento dei dati del matrimonio avviene con successo | Il test verifica che il sistema non solo aggiorni correttamente i dati, ma fornisca anche un feedback immediato al termine dell'operazione attraverso il messaggio “Dettagli del matrimonio aggiornati con successo!” |
| L’utente inserisce una spesa o imposta un budget con importo negativo | Viene negata l’operazione e l’utente viene avvisato di aver inserito un valore negativo | Il test verifica che il sistema blocchi l'inserimento di spese o del budget con valori negativi, garantendo l'integrità dei valori inseriti. In particolare, vengono visualizzati i messaggi: “Il valore deve essere superiore o uguale a 0” per il budget oppure “ L'importo della spesa deve essere superiore o uguale a 0” per una spesa |
| L'utente inserisce una o più spese tali che, sommandosi, superino il budget pianificato | Il sistema visualizza un messaggio di avviso di superamento del budget previsto | Il test verifica che il sistema monitori dinamicamente il totale delle spese e fornisca un feedback immediato in caso di superamento del budget, aiutando l'utente a mantenere un controllo sulle spese |
| L'utente tenta di creare un invito per un Wedding Planner | Il sistema mostra un avviso di conferma e, una volta accettato, genera l'invito per il Wedding Planner | Il test verifica che il sistema mostri l'avviso di conferma "Sei sicuro di voler generare un invito per Wedding Planner? Questo ruolo ha accesso a tutte le funzionalità di gestione del matrimonio" e, una volta confermato, crei l'invito per il Wedding Planner |
| L'utente tenta di creare un invito senza inserire l’email | Il sistema impedisce l’operazione | Il test verifica che il sistema richieda il campo email come obbligatorio per la creazione dell'invito, bloccando l'operazione e notificando l'utente in caso di omissione |
| L'invitato apre la pagina per rispondere all'invito e compila correttamente tutti i campi richiesti | Il sistema registra correttamente la risposta | Il test verifica che, una volta inviato il form, il sistema salvi correttamente la risposta dell'invitato e confermi l'operazione con un messaggio di successo |
| L’utente tenta di registrarsi utilizzando un token inesistente o scaduto | Il sistema impedisce la registrazione | Il test assicura che un token possa essere utilizzato una sola volta e venga invalidato dopo l'uso.  Il sistema mostra un messaggio di errore “Token non valido o scaduto” |