



# SocialBridge

## Test Plan

## SocialBridge

<b>Riferimento</b>	C18_TP
<b>Versione</b>	0.5
<b>Data</b>	28/11/2024
<b>Destinatario</b>	Prof.ssa Filomena Ferrucci, Prof. Fabio Palomba
<b>Presentato da</b>	C18 team
<b>Approvato da</b>	

# Team Members

---

<i>Nome</i>	<i>Cognome</i>	<i>Ruolo</i>	<i>Acronimo</i>	<i>Contatto</i>
Francesco	Peluso	PM	FP	f.peluso25@studenti.unisa.it
Luciano	Bercini	PM	LB	l.bercini@studenti.unisa.it
Mariarosaria	Rossi	PM	MR	m.rossi60@studenti.unisa.it
Fabio	Di Lallo	TM	FDL	f.dilallo3@studenti.unisa.it
Michele	Di Meo	TM	MDM	m.dimeo7@studenti.unisa.it
Luca	Fiore	TM	LF	l.fiore20@studenti.unisa.it
Andrea	Senatore	TM	AS	a.senatore158@studenti.unisa.it
Francesco	De Felice	TM	FDF	f.defelice4@studenti.unisa.it
Matteo	Nocerino	TM	MN	m.nocerino8@studenti.unisa.it
Alfio	Marra	TM	AM	a.marra39@studenti.unisa.it

## Revision History

---

<i>Data</i>	<i>Versione</i>	<i>Descrizione</i>	<i>Autori</i>
28/11/2024	0.1	Prima stesura	FDL, MDM, LF, AS, FDF, MN, AM
29/11/2024	0.2	Scrittura paragrafi 1, 2, 3, 4, 5, 6	FDL, MDM, LF, AS, FDF, MN, AM
30/11/2024	0.3	Scrittura paragrafi 7, 8, 9	FDL, MDM, LF, AS, FDF, MN, AM
1/12/2024	0.4	Scrittura Unit Test Cases	FDL, MDM, LF, AS, FDF, MN, AM
2/12/2024	0.5	Revisione Test Plan	FDL, MDM, LF, AS, FDF, MN, AM

## Sommario

●	<i>Team Members</i> .....	2
●	<i>Revision History</i> .....	2
●	<i>1: Introduzione</i> .....	4
●	<i>2: System overview</i> .....	4
●	<i>3: Relationship with other documents</i> .....	4
●	<i>4: Features to be tested</i> .....	4
●	<i>5: Pass/Fail criteria</i> .....	5
●	<i>6: Test approach</i> .....	5
●	<i>7: Suspension and reinstatement criteria</i> .....	5
	○ <i>7.1: Suspension criteria</i> .....	5
	○ <i>7.2: Reinstatement criteria</i> .....	5
●	<i>8: Test deliverables</i> .....	5
●	<i>9: Testing schedule</i> .....	6
●	<i>10: Unit Test Cases</i> .....	8

## 1. Introduzione

SocialBridge è una piattaforma web inclusiva progettata per facilitare la partecipazione ad eventi sociali, garantendo accessibilità, sicurezza e usabilità. Il progetto promuove l'inclusività sociale facilitando la partecipazione a eventi per individui con esigenze particolari, come persone introversive o con disabilità.

Questo documento ha lo scopo di descrivere le attività di testing pianificate per la piattaforma SocialBridge per garantire il corretto funzionamento.

## 2. System Overview

SocialBridge è basato su un'architettura MVC, con il front-end sviluppato in React.js e il back-end realizzato in Node.js e MongoDB per la realizzazione del database. La suddivisione in moduli include Gestione Utente, Eventi, Messaggistica, Notifiche e Assistenza Utente.

## 3. Relationships with other documents

Per realizzare correttamente i test case, si fa riferimento ai seguenti documenti:

- **RAD:** in quanto i test case sono definiti sulla base dei requisiti funzionali, requisiti non funzionali e dei casi d'uso;
- **SDD:** in quanto i test case devono rispettare la suddivisione in sottosistemi descritta dal suddetto documento.

Di seguito i riferimenti ai documenti utili:

- [Requirement Analysis Document](#)
- [System Design Document](#)

## 4: Features to be Tested

La fase di testing è stata progettata per coprire tutte le funzionalità implementate nei DAO, come indicato nell'Object Design Document (ODD). Di seguito sono elencate le funzionalità che verranno testate:

- **Gestione Utente**
  - Registrazione di un nuovo utente.
  - Login utente autenticato.
  - Logout di un utente.
  - Modifica dei dati personali:
    - Nome.
    - Cognome.
    - Ruolo.
    - Interessi.
  - Recupero dei dettagli di un utente tramite ID.
  - Eliminazione di un account utente.

- **Gestione Eventi**

- Creazione di un evento con le seguenti informazioni obbligatorie:
  - Titolo.
  - Descrizione.
  - Data e ora.
  - Luogo.
  - Accessibilità.
  - Numero massimo di partecipanti.
- Modifica dei dettagli di un evento.
- Registrazione di un utente a un evento, verificando:
  - Disponibilità dei posti.
  - Validità dell'evento.
- Eliminazione di un evento.
- Ricerca di eventi tramite:
  - Titolo.
  - Descrizione.
  - Data.
  - Luogo.
  - Accessibilità.
- Recupero dei dettagli di un evento tramite ID.

- **Gestione Messaggistica**

- Invio di messaggi tra utenti:
  - Verifica degli ID del mittente e del destinatario.
  - Presenza di un contenuto testuale valido.
- Recupero della cronologia dei messaggi tra due utenti.
- Notifica di nuovi messaggi ricevuti.
- Invio di messaggi multimediali.
- Eliminazione di un messaggio specifico.
- Marcatura di un messaggio come letto.

- **Gestione Notifiche**

- Invio di notifiche per:
  - Eventi a cui l'utente è iscritto.
  - Modifiche agli eventi.
  - Aggiornamenti di sistema.
- Recupero delle notifiche ricevute.
- Marcatura di una notifica come letta.

- **Gestione Assistenza Utente**

- Invio di richieste di supporto con:
  - Dettagli del problema.
  - Allegati opzionali.
- Recupero delle richieste di supporto inviate da un utente.

- Risoluzione di una richiesta di supporto.
- Eliminazione di una richiesta di supporto.

- **Persistenza dei Dati**

- Operazioni CRUD sui DAO (Create, Read, Update, Delete), garantendo:
  - Consistenza dei dati nel database MongoDB.
  - Integrità referenziale tra le entità correlate (ad esempio, utenti ed eventi).
  - Rispetto dei formati obbligatori per dati come email, password e date.

Le funzionalità non previste per la prima release del progetto saranno escluse dal testing di unità.

## 5: Pass/Fail criteria

I test avranno l'obiettivo di individuare failure all'interno delle funzionalità implementate nei DAO, come descritto nel punto 4. Per ciascuna operazione, il criterio di successo è il seguente:

- **Pass:** Un test si considera superato se l'output effettivo corrisponde all'output atteso specificato nel Test Case, garantendo il comportamento corretto per ciascuna delle funzionalità indicate (es. registrazione utente, invio messaggi, creazione eventi).
- **Fail:** Un test fallisce se:
  - L'output effettivo non corrisponde a quello atteso.
  - Si verificano eccezioni o errori non previsti.
  - I vincoli definiti nell'ODD o SDD non sono rispettati (es. formato email non valido accettato).

Ogni failure sarà documentata nei Test Incident Report, specificando il fault individuato e suggerendo eventuali correzioni.

## 6: Test Approach

L'approccio scelto è il **bottom-up** testing, che sfrutta la tecnica del **Category Partition** per il testing funzionale black-box. Il testing riguarderà esclusivamente le unità (DAO), implementando i seguenti passaggi:

### 1. Unit Testing dei DAO

- **Tecnologia utilizzata:**
  - **Jest:** framework per il testing con supporto a MongoDB in memoria tramite *mongodb-memory-server*.
  - **Motivazione:** consente di testare direttamente i metodi DAO in un ambiente controllato, senza la necessità di una connessione a un database persistente.
- **Tecnica applicata:** Category Partition per ogni metodo dei DAO, incluso:

- Gestione utenti: registrazione, modifica, login/logout, recupero ed eliminazione.
- Gestione eventi: creazione, modifica, registrazione, ricerca, eliminazione e recupero dettagli.
- Gestione messaggistica: invio, recupero, notifiche, eliminazione e marcatura messaggi.
- Gestione notifiche: invio, recupero e marcatura.
- Gestione assistenza: invio, risoluzione, recupero ed eliminazione richieste di supporto.

## 2. Testing di Sistema (citato ma non dettagliato)

- Sebbene il sistema sarà sottoposto a testing con Selenium per verificare le interazioni tra le interfacce utente e il backend, il dettaglio di questi test non sarà incluso in questa documentazione.

## 3. Strumenti e Risorse

- **Strumenti principali:**
  - **Jest:** per il testing funzionale delle unità.
  - **mongodb-memory-server:** per simulare un database MongoDB a runtime.
- **Materiali richiesti:**
  - API per MongoDB (Mongoose) per interfacciarsi con i DAO.
  - File di configurazione per Jest e MongoDB (incluso *.env*).

## 4. Esempi di Category Partition

Per il metodo *registerUser* del DAO *GestioneUtenteDAO*:

- **Categoria: Email**
  - Partizione valida: email ben formata (es. *utente@example.com*).
  - Partizione non valida: email senza dominio (es. *utente@*) o email vuota.
- **Categoria: Password**
  - Partizione valida: password con lunghezza tra 8 e 64 caratteri.
  - Partizione non valida: password con lunghezza inferiore a 8 o priva di caratteri speciali.

Per il metodo *createEvent* di *GestioneEventiDAO*:

- **Categoria: Titolo**
  - Partizione valida: stringa con lunghezza compresa tra 5 e 50 caratteri.
  - Partizione non valida: stringa vuota o con più di 50 caratteri.
- **Categoria: Data**
  - Partizione valida: data futura rispetto alla data corrente.
  - Partizione non valida: data nel passato.

Ogni unità sarà sottoposta a test con tutte le combinazioni significative delle partizioni definite.



## **7: Suspension and reinstatement criteria**

### **7.1: Suspension criteria**

I test possono essere sospesi solo se viene mostrato un messaggio d'errore dovuto alla definizione del test altrimenti i test non possono essere sospesi finché la loro esecuzione non termina.

### **7.2: Reinstatement criteria**

I test sospesi verranno ripresi andando ad rieseguire l'intero test dopo avere corretto il fault.

## **8: Test deliverables**

I documenti di test che verranno prodotti saranno:

- Test Plan;
- Test Case Specification
- Test Execution Report
- Test Incident Report
- Test Summary Report

## **9: Testing schedule**

Le attività di testing verranno condotte come definite nei capitoli precedenti, quindi subito dopo la fase di system design.

Lo sviluppo dei test avverrà subito dopo aver ultimato la fase di sviluppo. Di conseguenza l'esecuzione dei test avverrà dopo la loro implementazione e verranno prodotti i vari report di test. Ulteriori informazioni sono indicate nei documenti di management.

## 10: Unit Test Cases

### 10.1 GestioneUtente

#### 10.1.1 UtenteDAO

<b>Metodo</b>	<i>registrazioneUtente(Utente utente)</i>
<b>Parametri</b>	<i>utente</i> : oggetto contenente nome, cognome, email, password e interessi.
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Utente</b>	<b>NUV</b> : Nuovo utente valido <b>UEX</b> : Utente già esistente <b>NMF</b> : Parametri mancanti
<b>Vincoli</b>	L'utente deve fornire una email unica.  I campi obbligatori (nome, cognome, email, password) devono essere compilati.
<b>Test Frame</b>	
<b>NUV</b>	<b>Oracolo</b> : L'utente viene salvato con successo nel database, email di conferma inviata.
<b>UEX</b>	<b>Oracolo</b> : Generata un'eccezione, "Utente già registrato". Nessuna modifica al database.
<b>NMF</b>	<b>Oracolo</b> : Generata un'eccezione, "Campi obbligatori mancanti". Nessuna modifica al database

<b>Metodo</b>	<i>loginUtente(String email, String password)</i>
<b>Parametri</b>	<i>email</i> : email dell'utente che tenta l'accesso <i>password</i> : password dell'utente.  <i>password</i> : password dell'utente.
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Utente</b>	<b>LV</b> : Login valido <b>CI</b> : Credenziali invalide <b>ND</b> : Utente non trovato nel database
<b>Vincoli</b>	La combinazione di email e password deve essere corretta.
<b>Test Frame</b>	
<b>LV</b>	<b>Oracolo</b> : Accesso consentito; sessione avviata con successo.
<b>CI</b>	<b>Oracolo</b> : Viene generata un'eccezione perchè email o la password inserita dall'utente non è valida.
<b>ND</b>	<b>Oracolo</b> : Viene generata un'eccezione perché l'utente non esiste nel database, di conseguenza non viene trovato nessun utente.

<b>Metodo</b>	<i>modificaDatiPersonali(Utente utente)</i>
<b>Parametri</b>	<i>utente</i> : oggetto contenente i dati aggiornati dell'utente inclusi email, password e interessi.
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Utente</b>	<b>MP</b> : Modifica dei dati personali valida <b>ME</b> : Modifica email non valida <b>MPW</b> : Modifica password non valida
<b>Vincoli</b>	L'email deve essere unica nel database.  La nuova password deve rispettare i requisiti

	di sicurezza.
<b>Test Frame</b>	
<b>MP</b>	<b>Oracolo:</b> I dati personali dell'utente vengono aggiornati nel database.
<b>ME</b>	<b>Oracolo:</b> Viene generata un'eccezione perché l'utente ha tentato di modificare la propria email con un valore già attualmente in uso. Non applica nessuna modifica dei dati personali dell'utente.
<b>MPW</b>	<b>Oracolo:</b> Viene generata un'eccezione perché l'utente tenta di modificare la password con una che non soddisfa i requisiti di sicurezza. Non applica nessuna modifica alla password o ad altri dati dell'utente.

<b>Metodo</b>	<i>cercaUtente(String parolaChiave, Filtro filtro)</i>
<b>Parametri</b>	<p><i>parola Chiave:</i> stringa di ricerca inserita dall'utente (es. "Marco").</p> <p><i>filtro:</i> oggetto contenente criteri aggiuntivi (es. interessi comuni, località, ecc.).</p>
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Utente</b>	<b>CU:</b> Ricerca valida con risultati <b>NS:</b> Nessun risultato trovato <b>EF:</b> Filtro non valido
<b>Vincoli:</b>	La parola chiave e i filtri devono essere correttamente formattati e non vuoti.
<b>Test Frame</b>	
<b>CU</b>	<b>Oracolo:</b> Una lista di utenti pertinenti viene restituita, ordinata in base ai criteri selezionati.
<b>NS</b>	<b>Oracolo:</b> Nessun utente trovato; messaggio restituito "Nessun utente corrispondente ai criteri".

<b>EF</b>	<b>Oracolo:</b> Generata un'eccezione, "Filtro non valido".
-----------	-------------------------------------------------------------

### 10.1.2 AffinitàDAO

<b>Metodo</b>	<i>creaAffinita(Affinita affinita)</i>
<b>Parametri</b>	<i>affinità</i>
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Affinità</b>	VA: Valutazione valida NV: Valutazione non valida NE: Utente inesistente
<b>Vincoli</b>	Entrambi gli utenti devono esistere nel database.  La valutazione (rating) deve essere un valore valido secondo i requisiti.
<b>Test Frame</b>	
<b>VA</b>	<b>Oracolo:</b> Una nuova affinità viene creata e salvata correttamente nel database.
<b>NV</b>	<b>Oracolo:</b> Viene generata un'eccezione se il rating supera il range consentito.
<b>NE</b>	<b>Oracolo:</b> Viene generata un'eccezione perché l'utente 1 vuole valutare un utente che non esiste nel database. Non avviene nessuna modifica al database.

## 10.2 GestioneEvento

### 10.2.1 EventoDAO

<b>Metodo</b>	<i>creaEvento(Evento evento)</i>
<b>Parametri</b>	<i>evento:</i> oggetto contenente titolo, descrizione, data, orario, luogo, capacità massima.

<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Evento</b>	<b>EV:</b> Evento valido <b>EC:</b> Campi obbligatori mancanti <b>EF:</b> Data evento non valida
<b>Vincoli</b>	La data deve essere futura.  Titolo e luogo sono obbligatori.
<b>Test Frame</b>	
<b>EV</b>	<b>Oracolo:</b> L'evento viene salvato con successo nel database, conferma inviata all'organizzatore.
<b>EC</b>	<b>Oracolo:</b> Generata un'eccezione, "Campi obbligatori mancanti". Nessuna modifica al database.
<b>EF</b>	<b>Oracolo:</b> Generata un'eccezione, "Campo non valido". Nessuna modifica al database.

<b>Metodo</b>	<i>modificaEvento(Evento evento)</i>
<b>Parametri</b>	<i>evento:</i> oggetto evento aggiornato con i nuovi dettagli.
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Evento</b>	<b>MEV:</b> Modifica evento valida <b>CM:</b> Campi obbligatori mancanti <b>NE:</b> Evento non trovato
<b>Vincoli</b>	Il titolo, la data e il luogo devono essere forniti per la modifica.
<b>Test Frame</b>	
<b>MEV</b>	<b>Oracolo:</b> L'evento viene aggiornato nel database.
<b>CM</b>	<b>Oracolo:</b> Generata un'eccezione, "Campi

	obbligatorie mancanti".
<b>NE</b>	<b>Oracolo:</b> Generata un'eccezione, "Evento non trovato".

<b>Metodo</b>	<i>eliminaEvento(int eventoID)</i>
<b>Parametri</b>	<i>eventoID</i> : ID dell'evento da eliminare.
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Evento</b>	<b>EV</b> : Eliminazione valida <b>NE</b> : Evento non trovato <b>NP</b> : Evento non eliminabile (evento già passato)
<b>Vincoli</b>	L'evento deve esistere e non deve avere vincoli bloccanti per l'eliminazione.
<b>Test Frame</b>	
<b>EV</b>	<b>Oracolo:</b> L'evento viene eliminato dal database, notifiche inviate ai partecipanti.
<b>NE</b>	<b>Oracolo:</b> Generata un'eccezione, "Evento non trovato".
<b>NP</b>	<b>Oracolo:</b> Generata un'eccezione, "Evento non eliminabile"

<b>Metodo</b>	<i>cercaEvento(String parolaChiave, Filtro filtro)</i>
<b>Parametri</b>	<i>parolaChiave</i> : stringa di ricerca inserita dall'utente (es. "Passeggiata").  <i>filtro</i> : oggetto contenente criteri aggiuntivi come data, luogo, accessibilità, ecc.
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Evento</b>	<b>CE</b> : Ricerca valida con risultati

	<b>NS:</b> Nessun risultato trovato <b>EF:</b> Filtro non valido
<b>Vincoli</b>	La parola chiave e i filtri devono essere correttamente formattati.
<b>Test Frame</b>	
<b>CE</b>	<b>Oracolo:</b> Una lista di eventi pertinenti viene restituita, ordinata in base ai criteri selezionati.
<b>NS</b>	<b>Oracolo:</b> Nessun evento trovato; messaggio restituito "Nessun evento corrispondente ai criteri".
<b>EF</b>	<b>Oracolo:</b> Generata un'eccezione, "Filtro non valido".

### 10.2.2 RecensioneDAO

<b>Metodo</b>	<i>creaRecensione(int eventoID, Recensione recensione)</i>
<b>Parametri</b>	<i>eventoID:</i> ID dell'evento recensito.  <i>recensione:</i> oggetto contenente valutazione e commento.
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Recensione</b>	<b>RV:</b> Recensione valida <b>NF:</b> Campi obbligatori mancanti <b>NE:</b> Evento non trovato
<b>Vincoli</b>	La valutazione e l'ID evento devono essere forniti.
<b>Test Frame</b>	
<b>EV</b>	<b>Oracolo:</b> Recensione salvata nel database e visibile nella pagina dell'evento.
<b>NE</b>	<b>Oracolo:</b> Generata un'eccezione, "Campi



	obbligatorie mancanti".
<b>NP</b>	<b>Oracolo:</b> Generata un'eccezione, "Evento non trovato".

### 10.3 GestioneAssistenzaUtente

#### 10.3.1 SupportoDAO

<b>Metodo</b>	<i>richiediAssistenza(String categoria, String descrizione)</i>
<b>Parametri</b>	<i>categoria:</i> categoria del problema (es. "Errore evento"). <i>descrizione:</i> dettagli del problema riscontrato.
<b>Oggetti dell'ambiente</b>	Database
<b>Categorie</b>	<b>Scelte</b>
<b>Supporto</b>	<b>RA:</b> Richiesta valida <b>CM:</b> Campi obbligatori mancanti
<b>Vincoli:</b>	Categoria e descrizione devono essere specificate.
<b>Test Frame</b>	
<b>RA</b>	<b>Oracolo:</b> Richiesta salvata nel database e notificata al team di supporto.
<b>CM</b>	<b>Oracolo:</b> Generata un'eccezione, "Campi obbligatori mancanti".