

SwaGGed
Test Case Specification Document
Versione 4.0



Data: 09/01/2025

<u>Progetto: SwaGGed</u>	Versione: 4.0
<u>Documento: Test Case Specification</u>	Data: 09/01/2025

Coordinatore del progetto

Nome	Matricola

Partecipanti:

Nome	Matricola
Choaib Goumri	0512118390
Mattia Gallucci	0512116893

Scritto da:	Choaib Goumri
	Mattia Gallucci

Revision History

Data	Versione	Descrizione	Autore
19/12/2024	1.0	Inizio test case specification	Choaib Goumri, Gallucci Mattia
26/12/2024	2.0	Cases riguardo le aggiunte	Choaib Goumri, Gallucci Mattia
02/01/2025	3.0	Cases riguardo le eliminazioni	Choaib Goumri, Gallucci Mattia
08/01/2025	4.0	Aggiunta del caso di ricerca	Choaib Goumri, Gallucci Mattia

Sommario

1. TEST CASES.....	4
TCA Autenticazione.....	5
TCR Registrazione	11
TCP Aggiunta Post.....	26
TCC Aggiunta di un commento	33
TCL Aggiunta apprezzamento	38
TCCM Aggiunta di una community	43
TCEP Eliminazione Post	49
TCEC Eliminazione Commento.....	53
TCEM Eliminazione Community.....	59
TCRL Rimozione apprezzamento	64
TCRF Ricerca contenuti con filtro	69
TCIC Iscrizione Community	79
TCDC Disiscrizione Community	83
TCIU Inibisci Utente	87

1. TEST CASES

La descrizione fornisce un approccio sistematico alla progettazione dei test basato sul **category partition method**, una tecnica strutturata che permette di generare casi di test chiari, mirati ed efficaci. Questo metodo si fonda sull'identificazione e la suddivisione delle possibili categorie di input, condizioni o parametri di un sistema. Ogni categoria viene ulteriormente suddivisa in valori rappresentativi o partizioni, assicurando che i casi di test coprano una varietà di scenari con un approccio esaustivo ma ottimizzato.

Principali Passaggi del Category Partition Method

1. **Identificazione dei Parametri:** Analisi dei requisiti o delle specifiche del sistema per individuare i fattori rilevanti (input, condizioni o configurazioni) che influenzano il comportamento del sistema.
2. **Creazione delle Categorie:** Ogni parametro identificato viene suddiviso in categorie logiche che rappresentano i possibili stati o condizioni del parametro stesso.
3. **Partizionamento in Valori:** Le categorie vengono ulteriormente raffinate in partizioni o valori rappresentativi, includendo casi limite, valori tipici e scenari anomali.
4. **Definizione dei Vincoli:** Si stabiliscono regole e vincoli che limitano combinazioni non valide o irrilevanti tra categorie.
5. **Generazione dei Test Case:** Utilizzando le combinazioni valide dei valori, si crea un set completo di casi di test che coprono le diverse condizioni del sistema.

TCA Autenticazione

Il documento descrive un caso d'uso per l'autenticazione di un utente tramite e-mail e password. Viene definito un insieme di parametri da validare, come la presenza dell'e-mail nel database, il formato e la lunghezza sia dell'e-mail che della password, e la verifica che i campi non siano vuoti.

Il formato dell'e-mail è considerato valido se contiene un dominio, il simbolo "@" e una estensione, come ".com" o ".it". Qualsiasi e-mail priva di queste caratteristiche è ritenuta non valida. Analogamente, il formato della password è considerato valido solo se contiene una combinazione di lettere, numeri e caratteri speciali; una password composta esclusivamente da numeri è ritenuta non valida.

Queste regole garantiscono che i dati immessi siano conformi agli standard richiesti e riducono il rischio di errori o falle nella sicurezza.

Inoltre, vengono identificati vari test case che esplorano situazioni di autenticazione riuscita e fallita. Gli errori di autenticazione possono derivare da vari motivi, come e-mail non valida, formato errato della password, campo vuoto o lunghezza non conforme.

Ogni test case descrive i parametri da testare, l'input fornito e il risultato atteso, che può essere il successo dell'autenticazione o il fallimento con un messaggio di errore specifico.

L'obiettivo del documento è garantire che il sistema gestisca correttamente tutti i casi di input, assicurando sia la sicurezza che l'usabilità del processo di login.

Caso d'uso	UC 05 Autenticazione
Parametri	Username, Password
Oggetti dell'ambiente	Database
Categorie	Scelte
Username	U1: Username presente nel database U2: E Username non presente nel database
Campo username	CU1: Campo username vuoto CU2: Campo username inserito
Password	P1: Password associata ad username nel database P2: Password non associata ad username nel database
Campo Password	CP1: Campo password vuota CP2: Campo password inserito
Properties e selettori	U1: Username presente nel database [property username_presente]
	CE1: campo username vuoto [ERROR] CE2: campo username inserito [property CE_VALIDO]
	P1: Password associata ad e-mail nel database [if username_presente] [property P_VALIDO]
	CP1: campo password vuoto [ERROR] CP2: campo password inserito [property CP_VALIDO]

Test Case	Test Frame	Esito atteso
TCA1	U1, CU2, P1, CP2	Autenticazione riuscita: utente autenticato
TCA2	U2, CU2, P1, CP2,	Autenticazione fallita: username non presente
TCA3	U1, CU2, P2, CP2	Autenticazione fallita: password non associata a nessun username
TCA4	U2, CU1, P1, CP2	Autenticazione fallita: campo username vuoto
TCA5	U1, CU2, P2, CP1	Autenticazione fallita: campo password vuoto

Test Case ID	TCA1	Test Frame	U1, CU2, P1, CP2
Input			
Parametri		Valore della scelta	
E-mail		“swagged@gmail.com”	
Password		“SwaGGed07\$”	
Oracolo			
Autenticazione riuscita: utente autenticato			

Test Case ID	TCA2	Test Frame	U2, CU2, P1, CP2
Input			
Parametri		Valore della scelta	
E-mail		“swarreddd”	
Password		“Pass07\$”	
Oracolo			
Autenticazione fallita: username non presente			

Test Case ID	TCA3	Test Frame	U1, CU2, P2, CP2,
Input			
Parametri		Valore della scelta	
E-mail		“swagged”	
<u>Password</u>		“Pass007\$”	
Oracolo			
Autenticazione fallita: password non associata a username			

Test Case ID	TCA5	Test Frame	U1, CU2, P2, CP1
Input			
Parametri		Valore della scelta	
E-mail		“swagged”	
<u>Password</u>		“”	
Oracolo			
Autenticazione fallita: campo password vuoto			

Test Case ID	TCA4	Test Frame	U2, CU1, P1, CP2
Input			
Parametri		Valore della scelta	
E-mail		“”	
<u>Password</u>		“ValidPassword123”	
Oracolo			
Autenticazione fallita: campo username vuoto			

TCR Registrazione

Il documento descrive un caso d'uso per la registrazione di un utente tramite username, e-mail, password e conferma della password. Viene definito un insieme di parametri da validare, come la disponibilità dello username nel database, la validità e la presenza dell'e-mail, il formato e la lunghezza sia della password che della conferma password, e la verifica che i campi non siano vuoti.

Il formato dell'e-mail è considerato valido se contiene un dominio, il simbolo "@" e una estensione, come ".com" o ".it". Qualsiasi e-mail priva di queste caratteristiche è ritenuta non valida. Analogamente, il formato della password è considerato valido solo se contiene una combinazione di lettere, numeri e caratteri speciali; una password composta esclusivamente da numeri è ritenuta non valida.

Queste regole garantiscono che i dati immessi siano conformi agli standard richiesti e riducono il rischio di errori o falle nella sicurezza.

Inoltre, vengono identificati vari test case che esplorano situazioni di registrazione riuscita e fallita. Gli errori di registrazione possono derivare da vari motivi, come username già in uso, e-mail già registrata, formato errato della password, lunghezza non conforme della password, campo vuoto, o mancato match tra password e conferma password.

Ogni test case descrive i parametri da testare, l'input fornito e il risultato atteso, che può essere la riuscita della registrazione o il fallimento con un messaggio di errore specifico. L'obiettivo del documento è garantire che il sistema gestisca correttamente tutti i casi di input, assicurando sia la sicurezza che l'usabilità del processo di registrazione.

Caso d'uso	UC O4 Registrazione
Parametri	E-mail, Password, Conferma password
Oggetti dell'ambiente	Database
Categorie	Scelte
Username	U1: Username presente nel database U2: Username non presente nel database
Campo Username	CU1: Campo username vuoto CU2: Campo username inserito
E-mail	E1: E-mail non presente nel database E2: E-mail già presente nel database
Formato E-mail	FE1: Formato e-mail non valido FE2: Formato e-mail valido
Campo E-mail	CE1: Campo e-mail vuoto CE2: Campo e-mail inserito
Lunghezza E-mail	LE1: $5 \leq \text{Lunghezza e-mail} \leq 30$ LE2: $\text{Lunghezza e-mail} < 5 \vee \text{Lunghezza e-mail} > 30$
Formato Password	FP1: Formato password non valido FP2: Formato password valido
Campo Password	CP1: Campo password vuota CP2: Campo password inserito
Lunghezza Password	LP1: $\text{lunghezza} \geq 7$ LP2: $\text{lunghezza} < 7$
Matching con campo password 2	MA1: match MA2: non match
Properties e selettori	CU1: Campo username vuoto [ERROR]

	E1: E-mail presente nel database [property e-mail_presente]
	FE1: formato e-mail non valido [ERROR]
	FE2: formato e-mail valido [if CE_VALIDO] [property FE_VALIDO]
	CE1: campo e-mail vuoto [ERROR]
	CE2: campo e-mail inserito [property CE_VALIDO]
	LE1: $5 \leq \text{Lunghezza e-mail} \leq 30$ [if CE_VALIDO] [property LE_VALIDO]
	LE2: Lunghezza e-mail < 5 Lunghezza e-mail > 30 [ERROR]
	FP1: formato password non valido [ERROR]
	FP2: formato valido [if CP_VALIDO] [property FP_VALIDO]
	CP1: campo password vuoto [ERROR]
	CP2: campo password inserito [property CP_VALIDO]
	LP1: lunghezza password ≥ 5 [if CP_VALIDO] [property LP_VALIDO]
	LP2: lunghezza password < 5 [ERROR]
	MA2: non match. [ERROR]

Test Case	Test Frame	Esito Atteso
TCR1	U2, E1, FE2, CE2, LE1, FP2, CP2, LP1, MA1, CU2	Registrazione riuscita: utente registrato
TCR2	U1, E1, FE2, CE2, LE1, FP2, CP2, LP1, MA1, CU2	Registrazione fallita: username già in uso
TCR3	U2, E2, FE2, CE2, LE1, FP2, CP2, LP1, MA1, CU2	Registrazione fallita: e-mail già in uso
TCR4	U2, E1, FE2, CE2, LE2, FP2, CP2, LP1, MA1, CU2	Registrazione fallita: lunghezza e-mail non valida
TCR5	U2, E1, FE2, CE2, LE1, FP1, CP2, LP2, MA1, CU2	Registrazione fallita: password troppo corta
TCR6	U2, E1, FE2, CE2, LE1, FP2, CP2, LP1, MA2, CU2	Registrazione fallita: password e conferma non corrispondono
TCR7	U2, E1, FE1, CE2, LE1, FP2, CP2, LP1, MA1, CU2	Registrazione fallita: formato e-mail non valido
TCR8	U2, E1, FE2, CE1, LE1, FP2, CP2, LP1, MA1, CU2	Registrazione fallita: campo e-mail vuoto
TCR9	U2, E1, FE2, CE2, LE1, FP1, CP2, LP1, MA1, CU2	Registrazione fallita: formato password non valido
TCR10	U2, E1, FE2, CE2, LE1, FP2, CP1, LP1, MA1, CU2	Registrazione fallita: campo password vuoto
TCR11	U2, E1, FE2, CE2, LE1, FP2, CP2, LP1, MA1, CU1	Registrazione fallita: campo username vuoto

Test Case	TCR1	Test Frame	U2, E1, FE2, CE2, LE1, FP2, CP2, LP1, MA1
Input			
Parametri		Valore della scelta	
Username		“nuovoUtente”	
E-mail		“nuovo.email@esempio.it”	
Password		Password123	
Conferma password		Password123	
Oracolo			
Registrazione riuscita: utente registrato			

Test Case	TCR2	Test Frame	U1, E1, FE2, CE2, LE1, FP2, CP2, LP1, MA1
Input			
Parametri		Valore della scelta	
Username		“nuovoUtente”	
E-mail		“nuovo.email2@esempio.it”	
Password		Password123	
Conferma password		Password123	
Oracolo			
Registrazione fallita: username già in uso			

Test Case	TCR3	Test Frame	U2, E2, FE2, CE2, LE1, FP2, CP2 LP1, MA1
Input			
Parametri		Valore della scelta	
Username		“nuovoUtente2”	
E-mail		“nuovo.email3@esempio.it”	
Password		Password123	
Conferma password		Password123	
Oracolo			
Registrazione fallita: e-mail già in uso			

Test Case	TCR4	Test Frame	U2, E1, FE2, CE2, LE2, FP2, CP2 LP1, MA1
Input			
Parametri		Valore della scelta	
Username		nuovoUtente3	
E-mail		emailmolto.lunga.che.non.vale@esempio.com	
Password		Password123	
Conferma password		Password123	
Oracolo			
Registrazione fallita: lunghezza e-mail non valida			

Test Case	TCR5	Test Frame	U2, E1, FE2, CE2, LE1, FP1, CP2, LP2, MA1
Input			
Parametri		Valore della scelta	
Username		nuovoUtente4	
E-mail		nuovo.email4@esempio.it	
Password		ps	
Conferma password		ps	
Oracolo			
Registrazione fallita: password troppo corta			

Test Case	TCR6	Test Frame	U2, E1, FE2, CE2, LE1, FP2, CP2, LP1, MA2
Input			
Parametri		Valore della scelta	
Username		nuovoUtente5	
E-mail		nuovo.email5@esempio.it	
Password		Password123	
Conferma password		Password1234	
Oracolo			
Registrazione fallita: password e conferma non corrispondono			

Test Case	TCR7	Test Frame	U2, E1, FE1, CE2, LE1, FP2, CP2, LP1, MA1
Input			
Parametri		Valore della scelta	
Username		nuovoUtente6	
E-mail		nuovo.email6esempio.it	
Password		Password123	
Conferma password		Password123	
Oracolo			
Registrazione fallita: formato e-mail non valido			

Test Case	TCR8	Test Frame	U2, E1, FE2, CE1, LE1, FP2, CP2, LP1, MA2
Input			
Parametri		Valore della scelta	
Username		nuovoUtente7	
E-mail			
Password		Password123	
Conferma password		Password123	
Oracolo			
Registrazione fallita: campo e-mail vuoto			

Test Case	TCR9	Test Frame	U2, E1, FE2, CE2, LE1, FP1, CP2, LP1, MA1
Input			
Parametri		Valore della scelta	
Username		“nuovoUtente8”	
E-mail		“nuovo.emai8l@esempio.it”	
Password		1234567	
Conferma password		1234567	
Oracolo			
Registrazione fallita: utente formato password non valido			

Test Case	TCR10	Test Frame	U2, E1, FE2, CE2, LE1, FP2, CP1, LP1, MA1
Input			
Parametri		Valore della scelta	
Username		“nuovoUtente8”	
E-mail		“nuovo.emai8l@esempio.it”	
Password		1234567	
Conferma password		1234567	
Oracolo			
Registrazione fallita: campo password vuoto			

Test Case	TCR11	Test Frame	U2, E1, FE2, CE2, LE1, FP2, CP2, LP1, MA1, CU1
Input			
Parametri		Valore della scelta	
Username			
E-mail		“swagged@gmail.com”	
Password		“Pass07\$”	
Conferma password		“Pass07\$”	
Oracolo			
Registrazione fallita: campo username vuoto			

TCP Aggiunta Post

Il documento descrive un caso d'uso per la creazione di un post, convalidando diversi parametri per garantire che l'operazione sia eseguita correttamente solo quando tutti i criteri vengono soddisfatti.

I parametri considerati includono titolo, corpo del post, immagine associata e lo stato di autenticazione dell'utente.

Vengono identificate diverse proprietà e selettori per valutare ciascun parametro, come la presenza del titolo, il formato e la lunghezza del titolo, la lunghezza del corpo del post, la dimensione dell'immagine, e l'autenticazione dell'utente.

Ogni test case esplora scenari che simulano condizioni di successo o di fallimento nella creazione del post.

Gli errori possono derivare da titolo mancante, formato non valido, lunghezza non conforme del titolo o del corpo, immagine troppo grande, oppure utente non autenticato.

I test case includono dettagli sul test frame, i valori di input forniti e l'oracolo, che specifica l'esito atteso dell'operazione, sia essa il successo o il fallimento con una descrizione dell'errore riscontrato.

L'obiettivo è assicurare che il sistema gestisca correttamente tutti i casi di input, garantendo una gestione sicura e funzionale della creazione del post.

Caso d'uso	UC 26 – Creare Post
Parametri	Titolo, Corpo, Immagine, Utente
Categorie	Scelte
Titolo	T1: titolo vuoto T2: titolo inserito
Formato titolo	FT1: formato non valido FT2: formato valido
Lunghezza titolo	LT1: lunghezza = 0 OR lunghezza > 256 LT2: 0 < lunghezza <= 256
Lunghezza corpo	LC1: lunghezza > 512 LC2: lunghezza <= 512
Dimensione immagine	DI1: dimensione > 10 MB DI2: dimensione <= 10MB
Utente	U1: Utente autenticato U2: Utente non autenticato
Properties e selettori	T1: titolo vuoto [ERROR] T2: titolo inserito [property INSERITO] FT1: formato non valido [ERROR] FT2: formato valido [if INSERITO && utente_autenticato] [property T_VALIDO] LT1: lunghezza = 0 OR lunghezza > 256 [ERROR] LT2: 0 < lunghezza <= 256 [if INSERITO AND T_VALIDO] LC1: lunghezza > 512 [ERROR] LC2: lunghezza <= 512 DI1: dimensione > 10 MB [ERROR] DI2: dimensione <= 10MB U1: Utente non autenticato [ERROR] U1: Utente autenticato [property utente_autenticato]

Test Case ID	Test Frame	Esito atteso
TCP1	T2, FT2, LT2, LC2, DI2, U1	Post creato
TCP2	T1, FT2, LT2, LC2, DI2, U1	Post non creato: titolo vuoto
TCP3	T2, FT1, LT2, LC2, DI2, U1	Post non creato: formato titolo non valido
TCP4	T2, FT2, LT1, LC2, DI2, U1	Post non creato: lunghezza titolo non valida
TCP5	T2, FT2, LT2, LC1, DI2, U1	Post non creato: lunghezza corpo non valida
TCP6	T2, FT2, LT2, LC2, DI1, U1	Post non creato: dimensione immagine non valida
TCP7	T2, FT2, LT2, LC2, DI2, U2	Post non creato: utente non autenticato

Test Case	TCP1	Test Frame	T2, FT2, LT2, LC2, DI2, U1
Input			
Parametri		Valore della scelta	
Titolo		“Miglior artista al mondo”	
Corpo		“Kanye West”	
Immagine		Null	
Oracolo			
Post creato			

Test Case	TCP2	Test Frame	T1, FT2, LT2, LC2, DI2, U1
Input			
Parametri		Valore della scelta	
Titolo		null	
Corpo		“Kanye West”	
Immagine		null	
Oracolo			
Post non creato: titolo vuoto			

Test Case	TCP3	Test Frame	T2, FT1, LT2, LC2, DI2, U1
Input			
Parametri		Valore della scelta	
Titolo		“ ”	
Corpo		“Kanye West”	
Immagine		null	
Oracolo			
Post non creato: formato titolo non valido			

Test Case	TCP4	Test Frame	T2, FT2, LT1, LC2, DI2, U1
Input			
Parametri		Valore della scelta	
Titolo		“”	
Corpo		“Kanye West”	
Immagine		null	
Oracolo			
Post non creato: lunghezza titolo non valida			

Test Case	TCP5	Test Frame	T2, FT2, LT2, LC1, DI2, U1
Input			
Parametri		Valore della scelta	
Titolo		“Miglior artista al mondo”	
Corpo		*lunghezza corpo >= 256*	
Immagine		null	
Oracolo			
Post non creato: lunghezza corpo non valida			

Test Case	TCP6	Test Frame	T2, FT2, LT2, LC2, DI1, U1
Input			
Parametri		Valore della scelta	
Titolo		“Miglior artista al mondo”	
Corpo		“Kanye West”	
Immagine		*dimensione immagine >= 10 MB*	
Oracolo			
Post non creato: dimensione immagine non valida			

Test Case	TCP7	Test Frame	T2, FT2, LT2, LC2, DI2, U2
Input			
Parametri		Valore della scelta	
Titolo		“Miglior artista al mondo”	
Corpo		“Kanye West”	
Immagine		Null	
Oracolo			
Post non creato: utente non autenticato			

TCC Aggiunta di un commento

Il documento descrive un caso d'uso per la creazione di un commento associato a un post da parte di un utente autenticato. Sono definiti i parametri da validare, tra cui l'esistenza dell'ID del post, la presenza e la lunghezza del contenuto del commento, e l'autenticazione dell'utente.

Il commento può essere aggiunto solo se il post esiste, il contenuto del commento è valido (non vuoto e non supera i 300 caratteri), e l'utente è autenticato.

Vengono inoltre forniti test case che verificano diverse combinazioni di input, come commenti su post inesistenti, commenti vuoti o troppo lunghi, e tentativi di commento da parte di utenti non autenticati, garantendo così la corretta gestione di tutti i possibili scenari.

Caso d'uso	UC 07 Creare commento
Parametri	ID post, Contenuto del commento, Utente
Oggetti dell'ambiente	Database
Categorie	Scelte
ID Post	ID1: Post esistente ID2: Post non esistente
Contenuto Commento	CC1: CCommento vuoto CC2: Commento non vuoto CC3: Commento troppo lungo (>300 caratteri)
Utente	U1: Utente autenticato U2: Utente non autenticato
Properties e selettori	ID1: Post esistente [property post_esistente] CC1: Commento vuoto [ERROR] CC2: Commento non vuoto [if post_esistente && utente_autenticato] [property CC_VALIDO] CC3: Commento troppo lungo [ERROR] U1: Utente autenticato [property utente_autenticato]

Test Case	Test Frame	Esito Atteso
TCC1	ID1, CC2, U1	Commento aggiunto con successo
TCC2	ID2, CC2, U1	Aggiunta fallita: post non esistente
TCC3	ID1, CC1, U1	Aggiunta fallita: commento vuoto
TCC4	ID1, CC3, U1	Aggiunta fallita: commento troppo lungo
TCC5	ID1, CC2, U2	Aggiunta fallita: utente non autenticato

Test Case	TCC1	Test Frame	ID1, CC2, U1
Input			
Parametri		Valori	
ID Post		“123”	
Contenuto commento		“Ben fatto”	
Utente		“utente123 “	
Oracolo			
Commento aggiunto con successo			

Test Case	TCC2	Test Frame	ID2, CC2, U1
Input			
Parametri		Valori	
ID Post		“999”	
Contenuto commento		“Ben fatto”	
Utente		“utente123 “	
Oracolo			
Aggiunta fallita: post non esistente			

Test Case	TCC3	Test Frame	ID1, CC1, U1
Input			
Parametri		Valori	
ID Post		“123”	
Contenuto commento		“”	
Utente		“utente123 “	
Oracolo			
Aggiunta fallita: commento vuoto			

Test Case	TCC4	Test Frame	ID1, CC2, U1
Input			
Parametri		Valori	
ID Post		“123”	
Contenuto commento		“[Un testo lungo oltre 300 caratteri qui...]”	
Utente		“utente123 “	
Oracolo			
Aggiunta fallita: commento troppo lungo			

Test Case	TCC5	Test Frame	ID1, CC2, U2
Input			
Parametri		Valori	
ID Post		“123”	
Contenuto commento		“Ben fatto”	
Utente			
Oracolo			
Aggiunta fallita: utente non autenticato			

TCL Aggiunta apprezzamento

Il documento descrive un caso d'uso che consente a un utente autenticato di apprezzare i contenuti attraverso l'aggiunta di un like associato a un post.

Perché l'operazione abbia successo, è necessario verificare l'esistenza del post nel database, l'assenza di un like precedente da parte dello stesso utente sul medesimo contenuto e l'autenticazione dell'utente.

Se una di queste condizioni non viene rispettata, il sistema non consente di aggiungere il like e restituisce un errore specifico.

L'obiettivo è garantire che l'interazione sia gestita correttamente in ogni scenario possibile, compresi i tentativi di aggiunta di un like a un post inesistente, l'aggiunta di un like già presente o l'azione da parte di un utente non autenticato. I test case individuati esplorano tutte le combinazioni rilevanti di input per assicurare la validità e l'affidabilità del processo.

Caso d'uso	UC 06 Apprezzare i Contenuti
Parametri	ID Post, Like, Utente
Oggetti dell'ambiente	Database
Categorie	Scelte
ID Post	ID1: Post non esistente ID2: Post esistente
Like	L1: Like già presente sul post da parte dell'utente L2: Like non presente sul post da parte dell'utente
Utente	U1: Utente autenticato U2: Utente non autenticato
Properties e selettori	ID1: Post non esistente [ERROR] ID2: Post esistente [property POST_VALIDO]
	L1: Like già presente sul post da parte dell'utente [ERROR: Like già presente] L2: Like non presente sul post da parte dell'utente [property LIKE_VALIDO]
	U1: Utente autenticato [property AUTENTICATO] U2: Utente non autenticato [ERROR]

Test Case	Test Frame	Esito Atteso
TCL1	ID2, U1, L2	Like aggiunto con successo
TCL2	ID1, U1, L2	Like non aggiunto: post inesistente
TCL3	ID2, U2, L2	Like non aggiunto: utente non autenticato
TCL4	ID2, U1, L1	Like non aggiunto: like già presente

Test Case	TCL1	Test Frame	ID2, U1, L2
Input			
Parametri		Valori	
ID Post		“123”	
Like		113	
Utente		“utente123 “	
Oracolo			
Like aggiunto con successo			

Test Case	TCL2	Test Frame	ID1, U1, L2
Input			
Parametri		Valori	
ID Post		“999”	
Like		123	
Utente		“utente123 “	
Oracolo			
Like non aggiunto: post inesistente			

Test Case	TCL3	Test Frame	ID2, U2, L2
Input			
Parametri		Valori	
ID Post		“123”	
Like		“123”	
Utente			
Oracolo			
Like non aggiunto: utente non autenticato			

Test Case	TCL4	Test Frame	ID2, U1, L1
Input			
Parametri		Valori	
ID Post		“123”	
Like		“123”	
Utente		“utente123 “	
Oracolo			
Like non aggiunto: like già presente			

TCCM Aggiunta di una community

Il documento descrive un caso d'uso relativo alla creazione di una community da parte di un utente autenticato. L'operazione richiede la validazione di alcuni parametri fondamentali come il nome della community, la descrizione associata e l'identità dell'utente. Il nome della community deve essere unico nel database, non può essere vuoto e deve essere inserito correttamente.

La descrizione della community deve essere valida, non vuota e non eccedere i 200 caratteri. Inoltre, l'utente deve essere autenticato per completare la creazione della community. Se una o più di queste condizioni non vengono rispettate, l'operazione fallisce e viene restituito un errore specifico che ne spiega il motivo.

I test case delineati esplorano tutte le combinazioni di input rilevanti, compresi scenari in cui il nome della community è già presente, la descrizione è vuota o troppo lunga, il nome della community non è inserito o l'utente non è autenticato.

Questi test garantiscono un'implementazione robusta e la corretta gestione di tutte le possibili situazioni, assicurando che solo community valide possano essere create e aggiunte al sistema.

Caso d'uso	UC 16 Creare community
Parametri	Nome Community, Descrizione Community, Utente
Oggetti dell'ambiente	Database
Categorie	Scelte
Nome Community	NC1: Nome già presente nel database NC2: Nome non presente nel database
Campo Nome Community	CNC1: Campo nome community vuoto CNC2: Campo nome community inserito
Descrizione Community	DC1: Descrizione valida DC2: Descrizione vuota DC3: Descrizione troppo lunga (>200 caratteri)
Utente	U1: Utente autenticato U2: Utente non autenticato
Properties e selettori	NC2: Nome non presente nel database [property community non_presente] CNC1: Campo nome community vuoto [ERROR] CNC2: Campo nome community inserito [property inserito] U1: Utente autenticato [property utente_autenticato] DC1: Descrizione valida [if inserito && community_non_presente && utente autenticato] [property DC_VALIDO] DC3: Descrizione troppo lunga (>200 caratteri) [ERROR]

Test Case	Test Frame	Esito Atteso
TCCM1	NC2, CNC2, DC1, U1	Community creata con successo
TCCM2	NC1, CNC2, DC1, U1	Creazione fallita: nome già presente
TCCM3	NC2, CNC1, DC1, U1	Creazione fallita: campo nome community vuoto
TCCM4	NC2, CNC2, DC2, U1	Creazione fallita: descrizione vuota
TCCM5	NC2, CNC2, DC3, U1	Creazione fallita: descrizione troppo lunga
TCCM6	NC2, CNC2, DC1, U2	Creazione fallita: utente non autenticato

Test Case	TCCM1	Test Frame	NC2, CNC2, DC1, U1
Input			
Parametri		Valori	
Nome Community		“TechEnthusiasts”	
Descrizione Community		“Community per amanti della musica”	
Utente		“utente123 “	
Oracolo			
Community creata con successo			

Test Case	TCCM2	Test Frame	NC2, CNC1, DC1, U1
Input			
Parametri		Valori	
Nome Community		“TechEnthusiasts “	
Descrizione Community		“Community per amanti della musica”	
Utente		“utente123 “	
Oracolo			
Creazione fallita: nome già presente			

Test Case	TCCM3	Test Frame	NC1, CNC2, DC1, U1
Input			
Parametri		Valori	
Nome Community		“ “	
Descrizione Community		“Community per amanti della musica”	
Utente		“utente123 “	
Oracolo			
Creazione fallita: campo nome community vuoto			

Test Case	TCCM4	Test Frame	NC2, CNC2, DC2, U1
Input			
Parametri		Valori	
Nome Community		“Enthusiasts”	
Descrizione Community		“ ”	
Utente		“utente123 “	
Oracolo			
Creazione fallita: descrizione vuota			

Test Case	TCCM5	Test Frame	NC2, CNC2, DC3, U1
Input			
Parametri		Valori	
Nome Community		“TechEnthusiasts”	
Descrizione Community		“[Un testo lungo oltre 200 caratteri qui...]”	
Utente		“utente123 “	
Oracolo			
Creazione fallita: descrizione troppo lunga			

Test Case	TCCM6	Test Frame	NC2, CNC2, DC1, U2
Input			
Parametri		Valori	
Nome Community		“TechEnthusiastssssss”	
Descrizione Community		“[Un testo lungo oltre 200 caratteri qui...]”	
Utente		“ “	
Oracolo			
Creazione fallita: utente non autenticato			

TCEP Eliminazione Post

Il documento descrive un caso d'uso per l'eliminazione di un post da parte di un utente, specificando i requisiti e le condizioni necessarie per completare con successo l'operazione.

Per eliminare un post, è necessario che l'utente sia autenticato e proprietario del post stesso, e che l>ID del post corrisponda a un elemento esistente nel database. Se il post non esiste, l'eliminazione non può avvenire.

Se l'utente è autenticato ma non è il proprietario del post, l'operazione viene rifiutata per mancanza di autorizzazione. Allo stesso modo, l'utente non autenticato non può procedere all'eliminazione di alcun post.

I test case presentano varie combinazioni di input per verificare che il sistema gestisca correttamente scenari diversi, tra cui l'eliminazione di un post valido da parte del proprietario, tentativi di eliminazione di post inesistenti, accesso negato a utenti non proprietari e rifiuto per utenti non autenticati.

Questi test assicurano che solo utenti autorizzati possano interagire con il sistema nel rispetto delle regole definite, garantendo la sicurezza e l'integrità del processo di gestione dei contenuti.

Caso d'uso	UC 21 Eliminare Post
Parametri	ID Post, Utente
Oggetti dell'ambiente	Database
Categorie	Scelte
ID Post	ID1: Post non esistente ID2: Post esistente
Utente	U1: Utente autenticato e proprietario post U2: Utente autenticato ma non proprietario post U3: Utente non autenticato
Properties e selettori	ID1: Post non esistente [ERROR] ID2: Post esistente [property post_esistente]] U1: Utente autenticato e proprietario post [property autorizzato] U2: Utente autenticato ma non proprietario post [ERROR] U3: Utente non autenticato [ERROR]

Test Case	Test Frame	Esito Atteso
TCEP1	ID2, U1	Post eliminato con successo
TCEP2	ID1, U1	Post non eliminato: post inesistente
TCEP3	ID2, U2	Post non eliminato: utente non autorizzato
TCEP4	ID2, U3	Post non eliminato: utente non autenticato

Test Case	TCEP1	Test Frame	ID2, U1
Input			
Parametri		Valori	
ID Post		“123”	
Utente		“utente123 “	
Oracolo			
Post eliminato con successo			

Test Case	TCEP2	Test Frame	ID1, U1
Input			
Parametri		Valori	
ID Post		“999”	
Utente		“utente123 “	
Oracolo			
Post non eliminato: post inesistente			

Test Case	TCEP3	Test Frame	ID2, U2
Input			
Parametri		Valori	
ID Post		“123”	
Utente		“utente1234 “	
Oracolo			
Post non eliminato: utente non autorizzato			

Test Case	TCEP4	Test Frame	ID2, U3
Input			
Parametri		Valori	
ID Post		“1234”	
Utente		“ “	
Oracolo			
Post non eliminato: utente non autenticato			

TCEC Eliminazione Commento

Il documento descrive un caso d'uso per la gestione dell'eliminazione di un commento associato a un post. Per procedere con successo, il sistema verifica che il post esista, che il commento sia valido e associato al post indicato, e che l'utente sia autenticato e proprietario del commento.

Se il post non è presente nel database, l'operazione non può essere eseguita. Analogamente, l'assenza del commento all'interno del post impedisce l'eliminazione. Gli utenti autenticati ma non proprietari del commento ricevono un errore per mancanza di autorizzazione. Anche gli utenti non autenticati non sono abilitati a eseguire l'operazione.

I test case coprono tutti i possibili scenari, verificando la corretta eliminazione del commento quando i parametri sono validi, e garantendo che l'operazione venga bloccata in caso di post inesistenti, commenti assenti o condizioni di autenticazione e autorizzazione non soddisfatte. Questi controlli assicurano la sicurezza e la coerenza nell'interazione con il sistema, rispettando i diritti degli utenti e preservando l'integrità dei contenuti.

Caso d'uso	UC 23 Eliminare Commento
Parametri	ID Post, Commento, Utente
Oggetti dell'ambiente	Database
Categorie	Scelte
ID Post	ID1: Post non esistente ID2: Post esistente
ID Commento	C1: Commento non esistente nel post indicato C2: Commento esistente nel post indicato
Utente	U1: Utente autenticato e proprietario del commento U2: Utente autenticato ma non proprietario del commento U3: Utente non autenticato
Properties e selettori	ID1: Post non esistente [ERROR] ID2: Post esistente [property POST_VALIDO] C1: Commento non esistente nel post indicato [ERROR] C2: Commento esistente nel post indicato [property COMMENTO_VALIDO] U1: Utente autenticato e proprietario del commento [property AUTORIZZATO] U2: Utente autenticato ma non proprietario del commento [ERROR] U3: Utente non autenticato [ERROR]

Test Case	Test Frame	Esito Atteso
TCEC1	ID2, C2, U1	Commento eliminato con successo
TCEC2	ID1, C1, U1	Commento non eliminato: post inesistente
TCEC3	ID2, C1, U1	Commento non eliminato: commento inesistente
TCEC4	ID2, C2, U2	Commento non eliminato: utente non autorizzato
TCEC5	ID2, C2, U3	Commento non eliminato: utente non autenticato

Test Case	TCEC1	Test Frame	ID2, C2, U1
Input			
Parametri		Valori	
ID Post		“123”	
Commento		“La tua musica è emozionante”	
Utente		“utente123 “	
Oracolo			
Commento eliminato con successo			

Test Case	TCEC2	Test Frame	ID1, C1, U1
Input			
Parametri		Valori	
ID Post		“999”	
Commento		“La tua musica è bella”	
Utente		“utente123 “	
Oracolo			
Commento non eliminato: post inesistente			

Test Case	TCEC3	Test Frame	ID2, C1, U1
Input			
Parametri		Valori	
ID Post		“123”	
Commento		“”	
Utente		“utente123 “	
Oracolo			
Commento non eliminato: commento inesistente			

Test Case	TCEC4	Test Frame	ID2, C2, U2
Input			
Parametri		Valori	
ID Post		“1234”	
Commento		“La tua musica è bella”	
Utente		“utente1234 “	
Oracolo			
Commento non eliminato: utente non autorizzato			

Test Case	TCEC5	Test Frame	ID2, C2, U3
Input			
Parametri		Valori	
ID Post		“1234”	
Commento		“La tua musica è bella”	
Utente		“ “	
Oracolo			
Commento non eliminato: utente non autenticato			

TCEM Eliminazione Community

Il documento descrive un caso d'uso per la gestione dell'eliminazione di una community da parte di un utente.

Per garantire che l'operazione venga eseguita correttamente, il sistema verifica che la community esista nel database e che l'utente sia autenticato e abbia il ruolo di proprietario della community. Se la community non esiste, l'operazione viene bloccata poiché non c'è nulla da eliminare.

Gli utenti autenticati ma non proprietari non hanno l'autorizzazione a eseguire questa operazione, e viene loro notificato un errore di mancata autorizzazione.

Gli utenti non autenticati non possono in alcun caso eliminare una community, essendo privi delle credenziali necessarie per identificarsi e verificare il possesso.

I test case assicurano che l'eliminazione della community avvenga correttamente solo quando tutte le condizioni sono soddisfatte, e che venga gestito ogni scenario che coinvolge community inesistenti, utenti non autorizzati o non autenticati. Questo approccio garantisce sicurezza e integrità nell'interazione con il sistema e salvaguarda i diritti degli utenti e la coerenza dei dati archiviati.

Caso d'uso	UC 22 Eliminare Community
Parametri	Community, Utente
Oggetti dell'ambiente	Database
Categorie	Scelte
Community	C1: Community non esistente C2: Community esistente
Utente	U1: Utente autenticato e proprietario della community U2: Utente autenticato ma non proprietario della community U3: Utente non autenticato
Properties e selettori	C1: Community non esistente [ERROR] C2: Community esistente [property COMMUNITY_VALIDA]
	U1: Utente autenticato e proprietario [property AUTORIZZATO] U2: Utente autenticato ma non proprietario [ERROR] U3: Utente non autenticato [ERROR]

Test Case	Test Frame	Esito Atteso
TCEM1	C2, U1	Community eliminata con successo
TCEM2	C1, U1	Community non eliminata: community inesistente
TCEM3	C2, U2	Community non eliminata: utente non autorizzato
TCEM4	C2, U3	Community non eliminata: utente non autenticato

Test Case	TCEM1	Test Frame	C2, U1
Input			
Parametri		Valori	
Community		“MusicEnthusiasts”	
Utente		“utente123 “	
Oracolo			
Community eliminata con successo			

Test Case	TCEM2	Test Frame	C1, U1
Input			
Parametri		Valori	
Community		“Enthusiastsssssss”	
Utente		“utente123 “	
Oracolo			
Community non eliminata: community inesistente			

Test Case	TCEM3	Test Frame	C2, U2
Input			
Parametri		Valori	
Community		“TechEnthusiasts”	
Utente		“utente1234 “	
Oracolo			
Community non eliminata: utente non autorizzato			

Test Case	TCEM4	Test Frame	C2, U3
Input			
Parametri		Valori	
Community		“MusicEnthusiasts”	
Utente		“ “	
Oracolo			
Community non eliminata: utente non autenticato			

TCRL Rimozione apprezzamento

Il caso d'uso descritto riguarda la rimozione di un apprezzamento (like) associato a un post. L'operazione può essere effettuata solo da un utente autenticato, e solo se il post esiste e il like che si intende rimuovere è stato effettivamente registrato da parte dello stesso utente.

Qualora il post non esista nel database, il sistema blocca l'operazione notificando che il like non può essere rimosso. Nel caso in cui l'utente non sia autenticato, l'operazione viene respinta poiché l'identità dell'utente non può essere verificata.

Se il like non è stato registrato sul post indicato, la rimozione non è necessaria e viene restituito un messaggio di errore relativo a questa situazione.

I test case coprono tutti i possibili scenari, garantendo la corretta gestione delle operazioni e la protezione dell'integrità dei dati. Questo sistema assicura che le operazioni di rimozione avvengano solo quando tutte le condizioni sono soddisfatte e fornisce risposte precise e comprensibili in caso di errori o tentativi non validi.

Caso d'uso	UC 24 Togliere Apprezzamento
Parametri	ID Post, Like, Utente
Oggetti dell'ambiente	Database
Categorie	Scelte
ID Post	ID1: Post non esistente ID2: Post esistente
Like	L1: Like già presente sul post da parte dell'utente L2: Like non presente sul post da parte dell'utente
Utente	U1: Utente autenticato U2: Utente non autenticato
Properties e selettori	ID1: Post non esistente [ERROR] ID2: Post esistente [property POST_VALIDO]
	L1: Like già presente sul post da parte dell'utente [ERROR: Like già presente] L2: Like non presente sul post da parte dell'utente [property LIKE_VALIDO]
	U1: Utente autenticato [property AUTENTICATO] U2: Utente non autenticato [ERROR]

Test Case	Test Frame	Esito Atteso
TCRL1	ID2, U1, L2	Like rimosso con successo
TCRL2	ID1, U1, L2	Like non rimosso: post inesistente
TCRL3	ID2, U2, L2	Like non rimosso: utente non autenticato
TCRL4	ID2, U1, L1	Like non rimosso: like già presente

Test Case	TCRL1	Test Frame	ID2, U1, L2
Input			
Parametri		Valori	
ID Post		“123”	
Like		113	
Utente		“utente123 “	
Oracolo			
Like rimosso con successo			

Test Case	TCRL2	Test Frame	ID1, U1, L2
Input			
Parametri		Valori	
ID Post		“999”	
Like		123	
Utente		“utente123 “	
Oracolo			
Like non rimosso: post inesistente			

Test Case	TCRL3	Test Frame	ID2, U2, L2
Input			
Parametri		Valori	
ID Post		“123”	
Like		“123”	
Utente			
Oracolo			
Like non rimosso: utente non autenticato			

Test Case	TCRL4	Test Frame	ID2, U1, L1
Input			
Parametri		Valori	
ID Post		“123”	
Like		“123”	
Utente		“utente123 “	
Oracolo			
Like non rimosso: like già presente			

TCRF Ricerca contenuti con filtro

Il caso d'uso descritto riguarda la ricerca di contenuti con l'applicazione di filtri, consentendo agli utenti di individuare Post, Community o Utenti specifici attraverso una stringa di ricerca.

L'operazione è valida solo se l'utente inserisce una stringa di ricerca, e il filtro selezionato determina il tipo di contenuto ricercato. Il sistema verifica, inoltre, l'esistenza dei contenuti richiesti nel database e restituisce i risultati solo se tutte le condizioni specificate vengono soddisfatte.

Se la stringa di ricerca non viene inserita, il sistema blocca immediatamente l'operazione e restituisce un errore per indicare che questo parametro è obbligatorio.

Analogamente, qualora il filtro selezionato non porti a risultati corrispondenti (ad esempio, un Post o una Community inesistente), il sistema segnala che il contenuto non è stato trovato.

Caso d'uso	UC O2 Ricerca contenuti con filtro
Parametri	Stringa di ricerca, Filtro, Post, Utenti, Community
Oggetti dell'ambiente	Database
Categorie	Scelte
Stringa di ricerca	SR1: Stringa di ricerca non inserita SR2: Stringa di ricerca inserita
Filtro	F1: Post F2: Community F3: Utenti
Post	P1: Post non esistente P2: Post esistente
Utente	U1: Utente non esistente U2: Utente esistente
Community	C1: Community non esistente C2: Community esistente
Properties e selettori	SR1: Stringa di ricerca non inserita [ERROR] SR2 Stringa di ricerca inserita [property RICERCA_VALIDA]
	F1: Filtro Post selezionato [property FILTRO_POST] F2: Filtro Community selezionato [property FILTRO_COMMUNITY] F3: Filtro Utenti selezionato [property FILTRO_UTENTI]
	P1: Post non esistente [ERROR] P2: Post esistente [if RICERCA_VALIDA && FILTRO_POST] [property POST_TROVATO]
	U1: Utente non esistente [ERROR] U2: Utente esistente [if RICERCA_VALIDA && FILTRO_UTENTE] [property UTENTE_TROVATO]
	C1: Community non esistente [ERROR] C2: Community esistente [if RICERCA_VALIDA && FILTRO_COMMUNITY] [property COMMUNITY_TROVATA]

Test Case	Test Frame	Esito Atteso
TCRF1	SR2, F1, P2, U1, C1	Post trovato con successo
TCRF2	SR2, F1, P1, U1, C1	Post non trovato
TCRF3	SR2, F2, P1, U1, C2	Community trovata con successo
TCRF4	SR2, F2, P1, U1, C1	Community non trovata
TCRF5	SR2, F2, P1, U1, C1	Utente trovato con successo
TCRF6	SR2, F3, P1, U1, C1	Utente non trovato
TCRF7	SR1, F1, <u>P1</u> , U1, C1	Errore: "Stringa di ricerca è obbligatoria"

Test Case	TCRF1	Test Frame	SR2, F1, P2, U1, C1
Input			
Parametri		Valori	
Stringa di Ricerca:		“GeeKed”	
Filtro		Post	
Post		“GeeKed”	
Utente		Non rilevante per il filtro Post	
Community		Non rilevante per il filtro Post	
Oracolo			
Post trovato con successo			

Test Case	TCRF2	Test Frame	SR2, F1, P1, U1, C1
Input			
Parametri		Valori	
Stringa di Ricerca:		“PostInesistente123”	
Filtro		Post	
Post			
Utente		Non rilevante per il filtro Post	
Community		Non rilevante per il filtro Post	
Oracolo			
Post non trovato			

Test Case	TCRF3	Test Frame	SR2, F2, ID1, U1, C2
Input			
Parametri		Valori	
Stringa di Ricerca:		“Music Innovators”	
Filtro		Community	
ID Post		Non rilevante per il filtro Community	
Utente		Non rilevante per il filtro Community	
Community		“Music Innovators”	
Oracolo			
Community trovata con successo			

Test Case	TCRF4	Test Frame	SR2, F2, ID1, U1, C1
Input			
Parametri		Valori	
Stringa di Ricerca:		“CommunityInesistente789”	
Filtro		Community	
ID Post		Non rilevante per il filtro Community	
Utente		Non rilevante per il filtro Community	
Community			
Oracolo			
Community non trovata			

Test Case	TCRF5	Test Frame	SR2, F3, ID1, U2, C1
Input			
Parametri		Valori	
Stringa di Ricerca:		“MarioRossi”	
Filtro		Utenti	
ID Post		Non rilevante per il filtro Utenti	
Utente		"MarioRossi"	
Community		Non rilevante per il filtro Utenti	
Oracolo			
Utente trovato con successo			

Test Case	TCRF6	Test Frame	SR2, F3, ID1, U1, C1
Input			
Parametri		Valori	
Stringa di Ricerca:		“UtenteInesistente456”	
Filtro		Utenti	
ID Post		Non rilevante per il filtro Utenti	
Utente			
Community		Non rilevante per il filtro Utenti	
Oracolo			
Utente non trovato			

Test Case	TCRF7	Test Frame	SR1, F1, ID1, U1, C1
Input			
Parametri		Valori	
Stringa di Ricerca:		“”	
Filtro		Post	
ID Post		Non rilevante	
Utente		Non rilevante	
Community		Non rilevante	
Oracolo			
Errore: "Stringa di ricerca è obbligatoria"			

TCIC Iscrizione Community

Il caso d'uso descritto riguarda l'iscrizione degli utenti a una community, una funzionalità che consente agli utenti autenticati di unirsi a community esistenti all'interno del sistema. Perché l'operazione sia valida e completata con successo, devono essere soddisfatte alcune condizioni essenziali.

In primo luogo, la community a cui si intende iscriversi deve esistere nel database. Se la community indicata non è presente, il sistema non può completare la richiesta e restituisce un messaggio di errore che avvisa l'utente dell'impossibilità di procedere perché la community non è valida o non esistente.

Un altro requisito fondamentale è che l'utente che richiede l'iscrizione sia autenticato. Solo gli utenti che hanno effettuato correttamente l'accesso al sistema possono unirsi a una community. Se l'utente non è autenticato, la richiesta di iscrizione viene immediatamente respinta, poiché il sistema non può verificare l'identità dell'utente né associare correttamente l'iscrizione al suo profilo. In questo caso, l'utente riceve un messaggio che lo invita a effettuare l'accesso prima di procedere.

Se entrambe le condizioni – l'esistenza della community e l'autenticazione dell'utente – sono soddisfatte, l'iscrizione viene eseguita con successo. L'utente viene aggiunto alla lista dei membri della community e riceve una notifica che conferma l'avvenuta iscrizione.

Questa gestione delle iscrizioni assicura un funzionamento chiaro e strutturato, evitando errori e garantendo che solo utenti verificati possano accedere alle community valide. Inoltre, il sistema fornisce messaggi espliciti e comprensibili in caso di problemi, migliorando l'esperienza dell'utente e mantenendo l'integrità e l'ordine all'interno del database delle community.

Caso d'uso	UC 11 Iscrivarsi alle Community
Parametri	Community, Utente
Oggetti dell'ambiente	Database
Categorie	Scelte
Community	C1: Community non esistente C2: Community esistente
Utente	U1: Utente non autenticato U2: Utente autenticato
Properties e selettori	C1: Community non esistente [ERROR] C2: Community esistente [if UTENTE_AUTENTICATO] [property ISCRIZIONE_VALIDA]
	U1: Utente non autenticato [ERROR] U2: Utente autenticato [property UTENTE_AUTENTICATO]

Test Case	Test Frame	Esito Atteso
TCIC1	U2, C2	Iscrizione avvenuta con successo
TCIC2	U1, C2	Iscrizione fallita: utente non autenticato
TCIC3	U2, C1	Iscrizione fallita: community non esistente

Test Case	TCIC1	Test Frame	U2, C2
Input			
Parametri		Valori	
Utente		“utente123”	
Community		MusicEnthusiast	
Oracolo			
Iscrizione avvenuta con successo			

Test Case	TCIC2	Test Frame	U1, C2
Input			
Parametri		Valori	
Utente			
Community		MusicEnthusiast	
Oracolo			
Iscrizione fallita: utente non autenticato			

Test Case	TCIC3	Test Frame	U2, C1
Input			
Parametri		Valori	
Utente		“utente123”	
Community		MusiccccEnthusiast	
Oracolo			
Iscrizione fallita: community non esistente			

TCDC Disiscrizione Community

Il caso d'uso descritto riguarda la disiscrizione degli utenti da una community, una funzionalità pensata per consentire agli utenti autenticati di lasciare una community a cui si erano precedentemente iscritti. Per garantire che l'operazione venga eseguita correttamente, devono essere soddisfatte alcune condizioni fondamentali.

Innanzitutto, la community da cui si intende disisciversi deve esistere nel database. Qualora la community non sia presente, il sistema blocca immediatamente la richiesta, notificando che la community non è valida e non può essere trovata. Allo stesso modo, l'utente che richiede la disiscrizione deve essere autenticato.

Se l'utente non ha effettuato l'accesso al sistema, la sua identità non può essere verificata e l'operazione non viene autorizzata. In questo caso, il sistema restituisce un messaggio di errore che sottolinea la necessità di autenticarsi per poter procedere.

Un ulteriore vincolo riguarda lo stato dell'utente rispetto alla community: l'utente deve risultare iscritto alla community da cui vuole disisciversi. Se l'utente non è membro della community, la richiesta di disiscrizione non ha senso e viene respinta. Il sistema, in tal caso, informa l'utente che non risulta iscritto e che quindi non può effettuare l'operazione richiesta.

Se tutte queste condizioni sono soddisfatte – ovvero la community è valida ed esistente, l'utente è autenticato e risulta effettivamente iscritto – l'operazione di disiscrizione viene completata con successo. Il sistema rimuove l'utente dalla lista dei membri della community e notifica che la disiscrizione è avvenuta correttamente.

Questo processo assicura che solo gli utenti autorizzati possano interagire con le community in modo controllato, preservando l'integrità del sistema e offrendo messaggi chiari e coerenti in caso di errori. In questo modo, la gestione delle disiscrizioni risulta trasparente ed efficiente, evitando ambiguità e garantendo un'esperienza utente ottimale.

Caso d'uso	UC 25 Disiscriversi alle Community
Parametri	Community, Utente
Oggetti dell'ambiente	Database
Categorie	Scelte
Community	C1: Community non esistente C2: Community esistente
Utente	U1: Utente non autenticato U2: Utente autenticato
Properties e selettori	C1: Community non esistente [ERROR] C2: Community esistente [if UTENTE_AUTENTICATO] [property DISISCRIZIONE_VALIDA]
	U1: Utente non autenticato [ERROR] U2: Utente autenticato [property UTENTE_AUTENTICATO]

Test Case	Test Frame	Esito Atteso
TCDC1	U2, C2	Disiscrizione avvenuta con successo
TCDC2	U1, C2	Disiscrizione fallita: utente non autenticato
TCDC3	U2, C1	Disiscrizione fallita: community non esistente

Test Case	TCDC1	Test Frame	U2, C2
Input			
Parametri		Valori	
Utente		“utente123”	
Community		MusicEnthusiast	
Oracolo			
Disiscrizione avvenuta con successo			

Test Case	TCIC2	Test Frame	U1, C2
Input			
Parametri		Valori	
Utente			
Community		MusicEnthusiast	
Oracolo			
Disiscrizione fallita: utente non autenticato			

Test Case	TCDC3	Test Frame	U2, C1
Input			
Parametri		Valori	
Utente		“utente123”	
Community		MusiccccEnthusiast	
Oracolo			
Disiscrizione fallita: community non esistente			

TCIU Inibisci Utente

Il caso d'uso descritto riguarda la funzionalità di inibizione di un utente registrato, un'operazione che consente a un moderatore di limitare l'accesso o le azioni di un determinato utente all'interno del sistema. Questa operazione è strettamente regolamentata e può essere effettuata solo quando tutte le condizioni necessarie sono soddisfatte.

Perché l'inibizione avvenga con successo, è indispensabile che il moderatore sia un utente con i privilegi necessari. Il sistema verifica che l'utente che avvia l'operazione sia effettivamente riconosciuto come moderatore nel database. Se questa condizione non è rispettata e l'utente non è un moderatore, il sistema rifiuta l'operazione e restituisce un messaggio di errore, impedendo qualsiasi azione non autorizzata.

Inoltre, l'operazione richiede che l'utente da inibire sia un utente registrato nel sistema. Questo significa che il profilo dell'utente deve essere presente nel database e che sia possibile identificare con precisione il suo account. Se l'utente non è registrato, l'inibizione fallisce, e il sistema restituisce un messaggio di errore che indica chiaramente l'impossibilità di procedere.

Se entrambe le condizioni sono soddisfatte – cioè, se l'operazione viene avviata da un moderatore e l'utente da inibire è registrato – il sistema esegue l'inibizione con successo. In questo caso, il profilo dell'utente viene aggiornato per riflettere lo stato di inibizione, e il moderatore riceve una conferma che l'azione è stata completata correttamente.

Questa gestione assicura che solo gli utenti con le autorizzazioni appropriate possano limitare altri utenti e che le azioni di moderazione siano sempre tracciabili e basate su criteri chiaramente definiti. Inoltre, il sistema fornisce messaggi espliciti e informativi in caso di errore, mantenendo trasparenza e controllo sull'operazione di inibizione.

Caso d'uso	<u>UC 19 Inibire Utente</u>
Parametri	<u>Utente Moderatore, Utente registrato</u>
Oggetti dell'ambiente	<u>Database</u>
Categorie	<u>Scelte</u>
Utente Moderatore	<u>UM1: Utente non Moderatore</u> <u>UM2: Utente Moderatore</u>
Utente registrato	<u>UR1: Utente non registrato</u> <u>UR2: Utente registrato</u>
Properties e selettori	<u>UM1: Utente non Moderatore [ERROR]</u> <u>UM2: Utente Moderatore [property</u> <u>MODERATORE]</u>
	<u>UR1: Utente non registrato [ERROR]</u> <u>UR2 Utente registrato</u> <u>[IfMODERATORE][property</u> <u>INIBIZIONE_AVVENUTA]</u>

Test Case	Test Frame	Esito Atteso
TCIU1	UM2, UR2	Inibizione avvenuta con successo
TCIU2	UM2, UR1	Inibizione fallita: utente non registrato
TCIU3	UM1, UR2	Inibizione fallita: utente non moderatore

Test Case	TCIU1	Test Frame	UM2, UR2
Input			
Parametri		Valori	
Utente Moderatore		"moderatore123"	
Utente Registrato		"utente456"	
Oracolo			
Inibizione avvenuta con successo			

Test Case	TCIU2	Test Frame	UM2, UR1
Input			
Parametri		Valori	
Utente Moderatore		"moderatore123"	
Utente Registrato			
Oracolo			
Inibizione fallita: utente non registrato			

Test Case	TCIU3	Test Frame	UM1, UR2
Input			
Parametri		Valori	
Utente Moderatore			
Utente Registrato		"utente456"	
Oracolo			
Inibizione fallita: utente non moderatore			