



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof. C. Gravino

Test Case Specification



Riferimento	
Versione	1.0
Data	15/12/2025
Destinatario	Docente Ingegneria del Software Prof. Carmine Gravino
Presentato da	Gaetano Aprile, Luigi Artuso, Alessandro De Bonis & Marco Galdi
Approvato da	



1. Test Case di Sistema

1.1 UC_1 - Registrazione Utente

Componente sotto test: *GestioneRegistrazioneService*

La funzione da testare è identificabile come:

Command: registraUtente

Syntax: registraUtente(UtenteDTO datiUtente)

Function: verificare la logica di business relativa alla creazione di un nuovo profilo sportivo nel sistema, validando i dati inseriti e garantendo l'unicità dell'account.

Input:

- **Nome:** Nome dell'utente.
- **Cognome:** Cognome dell'utente.
- **Email:** Indirizzo email (usato come username).
- **Password:** Password scelta.
- **ConfermaPassword:** Ripetizione della password per conferma.

Parametri e Categorie

Parametro: Nome	
Formato: <code>^[A-Za-z\s']{2,30}\$</code> (Lettere, spazi, apostrofi; min 2 max 30 caratteri)	
Nome Categoria	Scelta per categoria
Formato Nome [FNO]	1. Non rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FNO_OK]



Parametro: Cognome	
Formato: <code>^[A-Za-z\s']{2,30}\$</code> (Lettere, spazi, apostrofi; min 2 max 30 caratteri)	
Nome Categoria	Scelta per categoria
Formato Cognome [FCO]	1. Non rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FCO_OK]

Parametro: Email	
Formato: <code>^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,10}\$</code> (Standard email regex)	
Nome Categoria	Scelta per categoria
Formato Email [FE]	1. Non rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE_OK]
Disponibilità Email [DE]	1. Email già presente nel DB = false [error] 2. Email nuova (non presente) = true [PROPERTY DE_OK]



Parametro: Password	
Formato: $^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{8,}$$ (Almeno 8 caratteri, almeno una lettera e un numero)	
Nome Categoria	Scelta per categoria
Validità Password [VP]	1. Password debole (lunghezza o complessità errata) = false [error] 2. Password valida = true [PROPERTY VP_OK]

Parametro: Conferma Password	
Formato: Nessun formato specifico, deve coincidere con la password	
Nome Categoria	Scelta per categoria
Match Password [MP]	1. Password non coincidono = false [error] 2. Password coincidono = true [PROPERTY MP_OK]

Vincoli:

1. Se **Formato Nome [FNO]** è FNO1, il test fallisce.
2. Se **Formato Cognome [FCO]** è FCO1, il test fallisce.
3. Se **Formato Email [FE]** è FE1, il test fallisce.
4. Se **Disponibilità Email [DE]** è DE1 (Già in uso), il test fallisce.
5. Se **Validità Password [VP]** è VP1, il test fallisce.
6. Se **Match Password [MP]** è MP1, il test fallisce.



Test Case ID	Test Frame	Esito
TC_UC1_1	FNO1	Errato: Formato nome non valido
TC_UC1_2	FNO2, FCO1	Errato: Formato cognome non valido
TC_UC1_3	FNO2, FCO2, FE1	Errato: Formato email non valido
TC_UC1_4	FNO2, FCO2, FE2, DE1	Errato: Email già in uso
TC_UC1_5	FNO2, FCO2, FE2, DE2, VP1	Errato: Password debole
TC_UC1_6	FNO2, FCO2, FE2, DE2, VP2, MP1	Errato: Le password non coincidono
TC_UC1_7	FNO2, FCO2, FE2, DE2, VP2, MP2	Corretto



1.2 UC_2 - Dare una valutazione (rating) ad un partecipante

Componente sotto test: GestioneValutazioneService

La funzione da testare è identificabile come:

- **Command:** lasciaFeedback
- **Syntax:** lasciaFeedback(int idEvento, Utente valutatore, Utente valutato, RatingDTO punteggi, String descrizione)
- **Function:** verificare la logica di business relativa all'inserimento di una valutazione (Feedback) da parte di un utente verso un altro partecipante al termine di un evento controllando vincoli di partecipazione, stato evento e validità dei voti.

Input:

- **idEvento:** identificativo dell'evento sportivo.
- **valutatore:** utente che lascia il feedback (attualmente loggato).
- **valutato:** utente che riceve il feedback.
- **punteggi:** oggetto DTO contenente i voti per Abilità, Affidabilità, Sportività.
- **descrizione:** commento testuale facoltativo o obbligatorio che motiva la valutazione.

Parametro: Stato Evento	
Formato: ^\d+\$ (ID intero positivo esistente)	
Nome Categoria	Scelta per categoria
Stato Evento [SE]	1. Evento non terminato (Aperto/Futuro) = false [error] 2. Evento concluso (“Terminato”) = true [PROPERTY SE_OK]



Parametro: Partecipazione Valutato (Verifica presenza in lista iscritti)

Nome Categoria	Scelta per categoria
Partecipazione [PAR]	<p>1. Utente valutato NON presente all'evento = false [error]</p> <p>2. Utente valutato presente all'evento = true [PROPERTY PAR_OK]</p>

Parametro: Identità (Confronto tra valutatore e valutato)

Nome Categoria	Scelta per categoria
Auto-Valutazione [AV]	<p>1. Utente coincide con Valutato = true [error]</p> <p>2. Valutatore diverso da Valutato = false [PROPERTY AV_OK]</p>



Parametro: Unicità Feedback (Controllo duplicati su DB)	
Nome Categoria	Scelta per categoria
Duplicato [DUP]	1. Feedback già esistente per la coppia/evento = true [error] 2. Primo feedback per la coppia/evento = false [PROPERTY DUP_OK]

Parametro: Punteggi (Abilità, Affidabilità, Sportività)	
Formato: $^{[1-5]}\$$ (Valore numerico compreso tra 1 e 5)	
Nome Categoria	Scelta per categoria
Validità Voti[VV]	1. Almeno un voto < 1 OR voto > 5 = false [error] 2. Tutti i voti compresi tra 1 e 5 = true [PROPERTY VV_OK]

Parametro: Descrizione Feedback	
Formato: $^{[A-Za-z0-9\s.,!]\{5,255\}}\$$ (Testo alfanumerico, punteggiatura base, min 5 max 255 caratteri)	
Nome Categoria	Scelta per categoria



Descrizione [DD]	<ol style="list-style-type: none">1. Lunghezza < 5 OR Lunghezza > 255 = false [error]2. Lunghezza compresa tra 5 e 255 = true [PROPERTY LD_OK]
------------------	---

Vincoli

- Se **Stato Evento [SE]** è **SE1** (non terminato), il test fallisce.
- Se **Partecipazione [PAR]** è **PAR1** (non partecipante), il test fallisce.
- Se **Auto-Valutazione [AV]** è **AV1** (coincidono), il test fallisce.
- Se **Duplicato [DUP]** è **DUP1** (già votato), il test fallisce.
- Se **Validità Voti [VV]** è **VV1** (fuori range), il test fallisce.
- Se **Descrizione [DD]** è **DD1** (fuori range), il test fallisce.

Tabella dei test case

Test Case ID	Test Frame	Esito
TC_UC2_1	SE1	Errato: evento non ancora concluso
TC_UC2_2	SE2, PAR1	Errato: utente valutato non ha partecipato
TC_UC2_3	SE2, PAR2, AV1	Errato: tentativo di auto-valutazione
TC_UC2_4	SE2, PAR2, AV2, DUP1	Errato: feedback già inserito
TC_UC2_5	SE2, PAR2, AV2, DUP2, VV1	Errato: punteggi fuori range (1-5)



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof. C. Gravino

TC_UC2_6	SE2, PAR2, AV2, DUP2, VV2, DD1	Errato: Descrizione fuori range
TC_UC2_7	SE2, PAR2, AV2, DUP2, VV2, DD2	Corretto



1.3 UC_3 - Iscrizione Evento

Componente sotto test: *PartecipazioneEventoService* (Logica di Business)

La funzione da testare è identificabile come:

Command: *iscriviUtente*

Syntax: *iscriviUtente*(Utente utente, EventoSportivo evento)

Function: permettere ad un utente registrato di iscriversi a un evento sportivo se vi sono posti disponibili e non è già iscritto.

Input:

- **Utente:** L'oggetto che rappresenta l'utente che vuole iscriversi.
- **Evento:** L'oggetto che rappresenta la partita a cui partecipare.

Parametro: Stato Iscrizione Evento	
Formato: ^\d+\$ (ID intero positivo esistente)	
Nome Categoria	Scelta per categoria
Iscrizione [IS]	1. Utente già iscritto = false [error] 2. Utente non iscritto = true [PROPERTY IS_OK]

Parametro: Disponibilità Posti	
Nome Categoria	Scelta per categoria



Disponibilità [DP]	1. Posti disponibili = 0 = false [error] 2. Posti disponibili > 0 = true [PROPERTY DP_OK]
--------------------	--

Parametro: Stato Evento	
Nome Categoria	Scelta per categoria
Stato [SE]	1. Evento chiuso o annullato = false [error] 2. Evento aperto ("In attesa") = true [PROPERTY SE_OK]

Parametro: Validità Temporale	
Nome Categoria	Scelta per categoria
Validità Temporale [VT]	1. Data evento < Data odierna = false [error] 2. Data evento >= Data odierna = true [PROPERTY VT_OK]

Vincoli:

1. Se **Iscrizione [IS]** è **IS1** (Utente già iscritto), il test fallisce indipendentemente dagli altri parametri.
2. Se **Disponibilità [DP]** è **DP1** (Posti = 0), il test fallisce.
3. Se **Stato [SE]** è **SE1** (Evento chiuso), il test fallisce.
4. Se **Validità Temporale [VT]** è **VT1** (Data passata), il test fallisce.

Tabella dei Test Case



Test Case ID	Test Frame	Esito
TC_UC3_1	IS1	Errato: Utente già iscritto all'evento
TC_UC3_2	IS2,DP1	Errato: Posti esauriti
TC_UC3_3	IS2,DP2, SE1	Errato: Stato evento non valido
TC_UC3_4	IS2,DP2,SE2,VT1	Errato: Data evento passata
TC_UC3_5	IS2, DP2, SE2, VT2	Corretto



1.4 UC_4 - Creazione Evento

Componente sotto test: `GestioneEventoService`.

La funzione da testare è identificabile come:

Command: `creaEvento`

Syntax: `creaEvento(Utente organizzatore, String sport, DateTime dataOra, String luogo, int nPartMax)`

Function: Verificare la logica di creazione di un nuovo evento sportivo, assicurando che i vincoli temporali, numerici e di consistenza dei dati siano rispettati prima del salvataggio.

Input:

- **organizzatore:** L'utente che sta creando l'evento.
- **sport:** Tipologia di sport (Stringa).
- **dataOra:** Data e orario previsti per l'evento.
- **luogo:** Indirizzo o nome del campo da gioco.
- **nPartMax:** Numero massimo di partecipanti ammessi.

Parametro: Validità Temporale	
Formato: <code>^\d{4}-\d{2}-\d{2} \d{2}:\d{2}\$</code>	
Nome Categoria	Scelta per categoria
Validità Temporale [VT]	<div>1. Data evento < Data odierna = <code>false</code> <code>[error]</code></div> <div>2. Data evento >= Data odierna = <code>true</code> <code>[PROPERTY VT_OK]</code></div>



Parametro: Numero Partecipanti	
Formato: $^{\wedge}\backslash d+\$$	
Nome Categoria	Scelta per categoria
Numero Partecipanti [NP]	1. Numero Partecipanti ≤ 1 = <code>false</code> <code>[error]</code> 2. Numero Partecipanti ≥ 2 = <code>true</code> <code>[PROPERTY NP_OK]</code>

Parametro: Sport	
Formato: $^{\wedge}[a-zA-Z\backslash s]\{3,30\}\$$	
Nome Categoria	Scelta per categoria
Lunghezza Sport [LS]	1. Lunghezza < 3 OR Lunghezza > 30 = <code>false</code> <code>[error]</code> 2. Lunghezza ≥ 3 AND Lunghezza ≤ 30 = <code>true</code> <code>[PROPERTY LS_OK]</code>



Parametro: Luogo	
Formato: <code>^[a-zA-Z0-9\s,.-]{5,100}\$</code>	
Nome Categoria	Scelta per categoria
Lunghezza Luogo [LL]	1. Lunghezza < 5 OR Lunghezza > 100 = <code>false</code> [error] 2. Lunghezza >= 5 AND Lunghezza <= 100 = <code>true</code> [PROPERTY LL_OK]

Parametro: Stato Organizzatore	
Nome Categoria	Scelta per categoria
Stato Organizzatore [SO]	1. Organizzatore non valido (null) = <code>false</code> [error] 2. Organizzatore valido = <code>true</code> [PROPERTY SO_OK]

Vincoli:

1. Se **Validità Temporale [VT]** è **VT1** (Data passata), il test fallisce indipendentemente dagli altri parametri.
2. Se **Numero Partecipanti [NP]** è **NP1** (Insufficienti), il test fallisce.
3. Se **Lunghezza Sport [LS]** è **LS1** (Non Valido), il test fallisce.
4. Se **Luogo[LL]** è **LL1** (Non valido), il test fallisce.
5. Se **Stato Organizzatore [SO]** è **S01** (Null), il test fallisce.

Tabella dei Test Case

Test Case ID	Test Frame	Esito
--------------	------------	-------



TC_UC4_1	VT1	Errato: Data evento nel passato
TC_UC4_2	VT2, NP1	Errato: Numero partecipanti insufficiente
TC_UC4_3	VT2, NP2, LS1	Errato: Nome sport troppo breve o lungo
TC_UC4_4	VT2, NP2, LS2, LL1	Errato: Indirizzo luogo non valido
TC_UC4_5	VT2, NP2, LS2, LL2, SO1	Errato: Organizzatore non valido
TC_UC4_6	VT2, NP2, LS2, LL2, SO2	Corretto