



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software* 2025/2026
Prof. Carmine Gravino



ClubConnect

Test Plan

ClubConnect

Riferimento	NC11_IP
Versione	1.0
Data	18 dic 2025
Destinatario	Prof. Carmine Gravino
Presentato da	NC11 Team
Approvato Da	

Revision History

Data	Versione	Descrizione	Autori
12/12/2025	0.1	Prima stesura	D.R A.Z. G.R. S.R.
15/12/2025	0.2	Creazione dei test case: registrazione di un utente, generare report	G.R. A.Z.
17/12/2025	0.3	Creazione restanti test case.	D.R S.R.

Membri del Team

Nome	Acronimo	Contatto
Andrea Zeno	A.Z.	a.zeno1@studenti.unisa.it
Domenico Ricciardelli	D.R.	d.ricciardelli1@studenti.unisa.it
Salvatore Pio Ruggiero	S.R.	s.ruggiero43@studenti.unisa.it
Gerardo Russo	G.R.	g.russo274@studenti.unisa.it

SOMMARIO

Revision History	2
Membri del Team	3
SOMMARIO	4
1 Introduzione	5
2 Relazione con altri documenti	5
3 Panoramica del sistema	5
4 Features da testare/da non testare	5
5 Pass/Fail criteria	6
6 Approccio	6
7 Sospensione e Ripristino	6
8 Materiale di testing	7
9 Test cases	7
9.1: Registrazione.....	7
9.1.1: Registrazione di un utente.....	7
9.1.1.1: Ospite si registra come Utente.....	7
9.2: Gestione club.....	10
9.2.1: Creare Club/Associazione.....	10
9.2.1.1: Utente crea un gruppo.....	10
9.2.2: Generare report.....	13
9.2.2.1 Utente gestore crea un report tramite Genera Report.....	13
9.3: Admin.....	15
9.3.1: Bannare un utente.....	15
9.3.1.1 Admin banna un utente.....	15

1 Introduzione

Il documento di Test Plan ha l'obiettivo di definire le strategie che verranno utilizzate per effettuare il testing del sistema

Sono state pianificate attività di testing per i seguenti use case:

- **UC_OSP:** Registrazione di un utente
- **UC_UT:** Creare Club/Associazione
- **UC_UT_GE:** Generare report
- **UC_ADM:** Bannare un utente

2 Relazione con altri documenti

Per la corretta individuazione dei test case, si fa riferimento ad altri documenti prodotti.

Relazioni con il **Requirements Analysis Document** (RAD)

I test case pianificati nel Test Plan sono elaborati in relazione ai requisiti funzionali presentati nel RAD.

Relazioni con il **Test Case Specification** (TCS)

Nel Test Case Specification sono riportati i test frame del Category Partition

Relazioni con il **Test Incident Report** (TIR)

Il TIR documenta l'andamento dei test di unità.

Relazioni con il **Test Summary report** (TSR)

Il documento riassume lo stato complessivo dei test.

3 Panoramica del sistema

Il sistema si basa sull'architettura 3-tier:

- Aiven e SQL per il database su cloud in fase di deployment.
- Per la logica applicativa (back-end) sarà utilizzato Java.
- Per il front-end sarà utilizzato un framework css(bootstrap), javascript e html.

4 Features da testare

Di seguito la lista delle features di cui si effettuerà il testing per le varie gestioni:

- **Registrazione:**
 - Registrazione di un utente
- **Gestione club:**
 - Creare Club/Associazione
 - Generare report

- **Admin:**
 - Bannare un utente

5 Pass/Fail criteria

Le attività di testing sono mirate ad identificare la presenza di faults (errori) all'interno del sistema, per effettuare un successivo intervento di eliminazione.

L'esito di un test case è valutato mediante un oracolo, inteso come il risultato atteso della sua esecuzione, basandosi sui requisiti. Un test ha successo (pass) se, dato un input al sistema, l'output ottenuto è uguale all'output dell'oracolo. Un test fallisce (fail) se, dato un input al sistema, l'output ottenuto è diverso dall'output dell'oracolo.

Tutto il testing sarà considerato valido se i vincoli saranno rispettati:

- Testare gli use case specificati (nel punto 1 di questo documento);

6 Approccio

Nel processo di testing unitario, ogni metodo inerente alle funzionalità degli use case delle classi del sistema sarà testato. I casi di test saranno progettati con un approccio black-box e documentati nel codice usando il framework JUnit per il testing delle classi Java.

Per ciascuna classe di produzione (*Production Class*) sarà creata una classe di test corrispondente, denominata con il formato **NomeTest**. Le classi di test verranno sviluppate parallelamente alle rispettive classi di produzione, in modo da garantire una buona copertura del codice.

Le tecnologie utilizzate includeranno:

- **Mockito**: Utilizzato per creare mock e stub e isolare le componenti da testare.
- **JaCoCo**: Consente di misurare metriche di copertura del codice, come la Branch Coverage.
- **Maven**: Permette di gestire la build del progetto e automatizzare l'esecuzione dei test lato server.
- **JUnit**: Permette di definire casi di test isolati, configurazioni pre e post-esecuzione.

7 Sospensione e Ripristino

Questa sezione descrive i criteri per la sospensione delle attività di test e le azioni necessarie per il ripristino del processo.

Criteri di sospensione

Le attività di testing continueranno fino al loro completamento, anche in presenza di failure. Tuttavia, il processo di testing potrà essere temporaneamente sospeso se, durante l'esecuzione, viene rilevato un errore nella definizione di uno o più test.

Criteri di ripristino

Il testing riprenderà una volta che i fault individuati saranno stati corretti.

8 Materiale di testing

L'hardware necessario per l'attività di test è un semplice computer poiché abbiamo deciso di testare solo il server. Questa scelta è stata fatta perché il server è il punto focale dell'applicazione.

9 Test cases

L'approccio per la definizione dei test frame sarà il category partition.

9.1: Registrazione

9.1.1: Registrazione di un utente

9.1.1.1: Ospite si registra come Utente

Parametro: Username	
Nome categoria	Scelte per la categoria
Valore [VU]	1. Valore = EMPTY [error] 2. Valore != EMPTY [PROPERTY_VU OK]
Registrato [RU]	1. Username Registrato = true [error] 2. Username Registrato = false [PROPERTY_RU OK]
Parametro: Nome	
Nome categoria	Scelte per la categoria
Valore [VN]	1. Valore = EMPTY [error] 2. Valore != EMPTY [PROPERTY_VN OK]
Lunghezza [LN]	1. Lunghezza > 40 [error]

	2. Lunghezza < 40 [PROPERTY_LN OK]
Parametro: Cognome	
Nome categoria	Scelte per la categoria
Valore [VC]	1. Valore = EMPTY [error] 2. Valore != EMPTY [PROPERTY_VC OK]
Lunghezza [LC]	1. Lunghezza > 40 [error] 2. Lunghezza < 40 [PROPERTY_LC OK]
Parametro: Data di Nascita	
Formato: [0-9][0-9] / [0-9][0-9] / [0-9][0-9][0-9][0-9]	
Nome categoria	Scelte per la categoria
Valore [VDN]	1. Valore = EMPTY [error] 2. Valore != EMPTY [PROPERTY_VDN OK]
Formato [FDN]	1. Formato rispettato = false [error] 2. Formato rispettato = true [PROPERTY_FDN OK]
Parametro: Email	
Formato: [a-z0-9._%+\-]+@[a-z0-9.\-]+\.[a-z]{2,}\$	
Nome categoria	Scelta per la categoria
Valore [VE]	1. Valore = Empty [error] 2. Valore != Empty [PROPERTY_VE_OK]
Formato [FE]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY_FE OK]
Registrata [RE]	1. Mail registrata = true [error] 2. Mail registrata = false

	[PROPERTY_FE OK]
Parametro: Password	
Nome categoria	Scelta per la categoria
Valore [VP]	1. Valore = EMPTY [error] 2. Valore != EMPTY [PROPERTY_VP OK]
Parametro: Cellulare	
Formato: [+0-9] {0-16}	
Nome categoria	Scelta per la categoria
Valore [VCL]	3. Valore = Empty [error] 4. Valore != Empty [PROPERTY_VCL_OK]
Formato [FCL]	3. Rispetta il formato = false [error] 4. Rispetta il formato = true [PROPERTY_FCL OK]

Test Case	Test frame	Esito
TC1.1_1	VU1	Errore: il campo username non può essere vuoto
TC1.1_2	VU2, RU1	Errore: Username già in uso
TC1.2_1	VU2, RU2, VN1	Errore: il campo nome non può essere vuoto
TC1.2_2	VU2, RU2, VN2, LN1	Errore: Nome troppo lungo
TC1.3_1	VU2, RU2, VN2, LN2, VC1	Errore: Il campo cognome non può essere vuoto
TC1.3_2	VU2, RU2, VN2, LN2, VC2, LC1	Errore: Cognome troppo lungo

TC1.4_1	VU2, RU2, VN2, LN2, VC2, LC2, VDN1	Errore: Il campo data di nascita non può essere vuoto
TC1.4_2	VU2, RU2, VN2, LN2, VC2, LC2, VDN2, FDN1	Errore: La data di nascita non rispetta il formato
TC1.5_1	VU2, RU2, VN2, LN2, VC2, LC2, VDN2, FDN2, VE1	Errore: Il campo email non può essere vuoto
TC1.5_2	VU2, RU2, VN2, LN2, VC2, LC2, VDN2, FDN2, VE2, FE1	Errore: L'email non rispetta il formato
TC1.5_3	VU2, RU2, VN2, LN2, VC2, LC2, VDN2, FDN2, VE2, FE2, RE1	Errore: L'email è già registrata
TC1.6_1	VU2, RU2, VN2, LN2, VC2, LC2, VDN2, FDN2, VE2, FE2, RE2, VP1	Errore: Il campo password non può essere vuoto
TC1.7_1	VU2, RU2, VN2, LN2, VC2, LC2, VDN2, FDN2, VE2, FE2, RE2, VP2, VCL1	Errore: Il campo cellulare non può essere vuoto
TC1.7_2	VU2, RU2, VN2, LN2, VC2, LC2, VDN2, FDN2, VE2, FE2, RE2, VP2, VCL2, FCL1	Errore: Il numero di cellulare non rispetta il formato
TC1.8	VU2, RU2, VN2, LN2, VC2, LC2, VDN2, FDN2, VE2, FE2, RE2, VP2, VCL2, FCL2	Corretto: Registrazione effettuata con successo

9.2: Gestione club

9.2.1: Creare Club/Associazione

9.2.1.1: Utente crea un gruppo

Parametro: Nome	
Nome categoria	Scelte per la categoria
Valore [VN]	1. Valore = EMPTY [error]

	2. Valore != EMPTY [PROPERTY_VN OK]
Parametro: Descrizione	
Nome categoria	Scelte per la categoria
Valore [VD]	1. Valore = EMPTY [error] 2. Valore != EMPTY [PROPERTY_VD OK]
Lunghezza [LD]	1. Lunghezza > 255 [error] 2. Lunghezza <= 255 [PROPERTY_LD OK]
Parametro: Tipo	
Nome categoria	Scelte per la categoria
Valore [VTP]	1. Valore = EMPTY [error] 2. Valore != EMPTY [PROPERTY_VTP OK]
Parametro: Settore	
Nome categoria	Scelte per la categoria
Valore [VS]	1. Valore = EMPTY [error] 2. Valore != EMPTY [PROPERTY_VS OK]
Parametro: Slogan	
Nome categoria	Scelte per la categoria
Lunghezza [LSL]	1. Lunghezza > 32 [error] 2. Lunghezza <= 32 [PROPERTY_LSL OK]
Parametro: Indirizzo sede	
Formato: [A-zÀ-ù '-'] {2,30}	
Nome categoria	Scelte per la categoria
Formato [FIS]	1. Valore = EMPTY [error] 2. Valore != EMPTY [PROPERTY_FIS OK]

	OK]
Parametro: Retta	
Nome categoria	Scelte per la categoria
Importo [IR]	1. Importo = EMPTY [error] 2. Importo != EMPTY [PROPERTY_IR OK]
Frequenza [FR]	1. Frequenza = null [error] 2. Frequenza != null [PROPERTY_FR OK]
Parametro: Logo	
Nome categoria	Scelta per la categoria
Grandezza File [GFL]	1. Grandezza file > 15 [error] 2. Grandezza file <= 15 [PROPERTY_GFL OK]
Dimensione [DL]	1. Dimensione > 320x320 [error] 2. Dimensione <= 320x320 [PROPERTY_DL OK]

Test Case	Test Frame	Esito
TC2.1_1	VN1	Errore: il campo nome non può essere vuoto
TC2.2_1	VN2, VT2, ET2, FT2, VD1	Errore: il campo descrizione non può essere vuoto
TC2.2_2	VN2, VT2, ET2, FT2, VD2, LD1	Errore: la descrizione è troppo lunga
TC2.3_1	VN2, VT2, ET2, FT2, VD2, LD2, VTP1	Errore: Il campo tipo non può essere vuoto, deve necessariamente contenere o "club" o "associazione"
TC2.4	VN2, VT2, ET2, FT2, VD2,	Errore: il campo settore

	LD2, VTP2, VS1	non può essere vuoto
TC2.5	VN2, VT2, ET2, FT2, VD2, LD2, VTP2, VS2, LSL1	Errore: Lo slogan è troppo lungo
TC2.6	VN2, VT2, ET2, FT2, VD2, LD2, VTP2, VS2, LSL2, FIS1	Errore: Il formato dell'indirizzo della sede non rispetta il formato
TC2.7_1	VN2, VT2, ET2, FT2, VD2, LD2, VTP2, VS2, LSL2, FIS2, IR1	Errore: il campo retta non può non avere un importo
TC2.7_2	VN2, VT2, ET2, FT2, VD2, LD2, VTP2, VS2, LSL2, FIS2, IR2, FR1	Errore: E' necessario inserire la frequenza del pagamento della retta
TC2.8_1	VN2, VT2, ET2, FT2, VD2, LD2, VTP2, VS2, LSL2, FIS2, IR2, FR2, GFL1	Errore: il file immagine del logo è troppo grande
TC2.8_2	VN2, VT2, ET2, FT2, VD2, LD2, VTP2, VS2, LSL2, FIS2, IR2, FR2, GFL2, DL1	Errore: il file immagine del logo non rispetta le dimensioni consentite
TC2.9	VN2, VT2, ET2, FT2, VD2, LD2, VTP2, VS2, LSL2, FIS2, IR2, FR2, GFL2, DL2	Corretto: Gruppo creato correttamente

9.2.2: Generare report

9.2.2.1 Utente gestore crea un report tramite Genera Report

Parametro: Ruolo Utente	
Nome categoria	Scelta per la categoria
Permessi [PU]	1. Ruolo != Gestore [error] 2. Ruolo = Gestore [PROPERTY PU_OK]
Parametro: Selezione Informazioni	

Nome categoria	Scelta per la categoria
Stato Selezione [SSI]	1. Selezioni = Vuota [error] 2. Selezioni != Vuota [PROPERTY SSI_OK]
Parametro: Periodo Temporale	
Nuova categoria	Scelta per la categoria
Validità Date [VDT]	1. Data Inizio > Data Fine [error] 2. Data Inizio <= Data Fine [PROPERTY VDT_OK]
Parametro: Esito	
Nuova categoria	Scelta per la categoria
Quantità Risultati [QRE]	1. Dati trovati = 0 [PROPERTY QRE_EMPTY] 2. Dati > 0 AND Tempo < 30s [PROPERTY QRE_SYNC] 3. Dati > 0 AND Tempo > 30s [PROPERTY QRE_ASYNC]

Test Case id	Test Frame	Esito
TC_3.1	PU1	Errore: L'utente non ha i permessi di gestione, accesso negato.
TC_3.2	PU2, SSI1	Errore: Nessuna categoria di informazioni selezionata per il report.
TC_3.3	PU2, SSI2, VDT1	Errore: La data di inizio è successiva alla data di fine.
TC_3.4	PU2, SSI2, VDT2, QRE1	Avviso: Nessun dato

		trovato per i parametri e il periodo specificati.
TC_3.5	PU2, SSI2, VDT2, QRE2	Corretto: Report generato e scaricato immediatamente con successo.
TC_3.6	PU2, SSI2, VDPT2, QRE3	Corretto: Report voluminoso in elaborazione, verrà inviato via e-mail.

9.3: Admin

9.3.1: Bannare un utente

9.3.1.1 Admin banna un utente

Parametro: Utente	
Nome categoria:	Scelte per le categoria:
StatoBan[SB]	1. StatoBan = true[error] 2. StatoBan = false [PROPERTY_SB_OK]
Parametro: Motivazione (Input dati)	
Formato: [a-z0-9._%+\\-]+@[a-z0-9.\\-]+\\.[a-z]{2,}\$	
Nome categoria:	Scelte per le categoria:
Lunghezza Testo[LT]	1. Lunghezza Testo=null [error] 2. Lunghezza Testo!=null [PROPERTY_LT_OK]

Test Case id	Test Frame	Esito
--------------	------------	-------

TC_4.1_1	SB1	Errato: L'utente è già bannato, non può essere bannato di nuovo
TC_4.2_1	SB2,LT1	Errato: La motivazione non può risultare vuota.
TC_4.3	SB2,LT2	Corretto: Utente bannato con successo.