



Laurea Triennale in informatica - Università di Salerno

Corso di *Ingegneria del Software* - Prof. C. Gravino



# Test Plan

Stellar Library

<b>Riferimento</b>	NC10_TP_ver.1.0
<b>Versione</b>	1.0
<b>Data</b>	09/06/2026
<b>Destinatario</b>	Prof. Carmine Gravino
<b>Presentato da</b>	NC10 Team



### Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
<b>Antonio Plumitallo</b>	Team Member	AP	a.plumitallo@studenti.unisa.it
<b>Riccardo Odierna</b>	Team Member	RO	r.odierna1@studenti.unisa.it
<b>Biagio Radino</b>	Team Member	BR	b.radino@studenti.unisa.it
<b>Christian Antonio Genovese</b>	Team Member	CG	c.genovese18@studenti.unisa.it



## Revision History

Data	Versione	Descrizione	Autori
11/12/2025	0.1	Prima Stesura	CG
13/12/2025	0.2	Completamento	CG, BR ,RO
13/12/2025	0.3	Modifiche varie	CG, RO
17/12/2025	0.4	Modifiche per consistenza	CG
23/12/2025	0.5	Copertura test maggiore	CG
28/12/2025	0.6	Copertura test maggiore	CG
03/01/2026	0.7	Correzioni varie	AP
09/01/2026	1.0	Revisione	BR ,CG , AP ,RO



## 1. Introduzione

Il sistema StellarLibrary è una piattaforma e-commerce web-based progettata per supportare l'acquisto, la gestione e la scoperta di libri online. Il sistema permette agli utenti di consultare un catalogo organizzato, effettuare ricerche con filtri avanzati (titolo, autore, genere, prezzo, disponibilità), visualizzare schede prodotto complete di recensioni, gestire un carrello persistente e completare in sicurezza la procedura di checkout.

Sono inoltre disponibili funzionalità dedicate agli utenti autenticati come la consultazione della cronologia ordini e l'inserimento di recensioni.

Parallelamente, l'amministratore del sistema dispone di strumenti per la gestione del catalogo (inserimento, modifica e rimozione dei libri), per l'organizzazione degli elementi in categorie e per la moderazione delle recensioni pubblicate.

Nel presente documento vengono riportate:

- le funzionalità del sistema oggetto delle attività di test;
- l'approccio metodologico adottato per il testing, in conformità ai vincoli progettuali;
- i criteri di valutazione dell'esito dei test (pass/fail);
- i criteri di sospensione e ripristino dell'attività di testing;
- la definizione dei test case tramite tecnica Category Partition per le funzionalità selezionate;
- la pianificazione delle attività di testing di unità e di sistema.

L'obiettivo del Test Plan è identificare potenziali carenze in termini di correttezza, completezza e affidabilità delle componenti software, garantendo la consegna di un sistema conforme ai requisiti funzionali e privo di malfunzionamenti critici.



## 2. Relazione con altri documenti

Il presente documento è strettamente collegato ai materiali progettuali prodotti nell'ambito dello sviluppo del sistema *StellarLibrary*.

In particolare:

- **Requirements Analysis Document (RAD)**

I test case di sistema descritti nel Test Plan derivano direttamente dai casi d'uso individuati nel RAD.

Le funzionalità selezionate per il testing sono coerenti con i requisiti funzionali ad alta priorità definiti in tale documento.

- **System Design Document (SDD)**

La classificazione dei test case e la loro organizzazione fanno riferimento ai sottosistemi descritti nello SDD.

Per ogni test case, viene indicato il sottosistema di appartenenza per mantenere la tracciabilità tra progettazione e testing.

## 3. Panoramica del Sistema

Come definito nel *System Design Document*, il sistema *StellarLibrary* adotta un'architettura web three-tier, composta da livello di presentazione, livello di logica applicativa e livello di persistenza.

Il livello di presentazione è realizzato tramite tecnologie web standard, quali **HTML5**, **CSS3** e **JavaScript**, attraverso le quali viene gestita l'interazione con l'utente e la visualizzazione delle pagine della piattaforma.

La logica applicativa risiede nel livello intermedio, implementato tramite componenti backend che espongono servizi dedicati alla gestione delle principali funzionalità del sistema, quali autenticazione, ricerca nel catalogo, gestione del carrello, creazione degli ordini e gestione delle recensioni.

Per la persistenza dei dati, il sistema utilizza un database relazionale **MySQL**, accessibile attraverso l'adozione di **JDBC**. In esso sono memorizzate le informazioni relative agli utenti, ai libri, alle recensioni, agli ordini e ai dati del carrello.



Questa organizzazione architettuale consente una chiara separazione delle responsabilità tra i livelli e supporta efficacemente la modularità delle funzionalità descritte nel *Requirements Analysis Document*, facilitando l'individuazione delle componenti oggetto delle attività di testing.

#### 4. Funzionalità da testare

Le attività di testing di sistema previste nel presente documento riguardano un sottoinsieme delle funzionalità ad alta priorità identificate nel *Requirements Analysis Document (RAD)*.

In particolare, verranno testati i seguenti casi d'uso, scelti per rappresentare aree funzionali critiche del sistema *StellarLibrary*:

- **UC\_GCC\_7 – Checkout e creazione dell'ordine**  
Controllo dell'intero processo di completamento dell'acquisto, dalla gestione del carrello alla conferma dell'ordine, inclusa la selezione dell'indirizzo e del metodo di pagamento e la gestione dei possibili scenari di errore.
- **UC\_GA\_1 – Registrazione dell'utente**  
Verifica della correttezza del processo di registrazione, incluse la validazione dei campi obbligatori e la gestione di input non validi o email già presenti nel sistema.
- **UC\_GCR\_2 – Ricerca filtrata dei libri**  
Test delle funzionalità di ricerca e filtraggio del catalogo sulla base di criteri quali genere, prezzo, autore, titolo e disponibilità.
- **UC\_GR\_1 – Inserimento di una recensione**  
Verifica dell'inserimento di una recensione associata a un libro, compresa la validazione dei campi obbligatori (titolo e testo) e la corretta pubblicazione della recensione.



## 5. Approccio

Le attività di testing per il sistema *StellarLibrary* adottano un approccio di tipo **black-box**, basato sui requisiti funzionali definiti nel *Requirements Analysis Document (RAD)*.

Il testing è articolato in **testing di unità** e **testing di sistema**, entrambi progettati mediante la tecnica **Category Partition**.

Il **testing di unità** è finalizzato alla verifica del corretto comportamento di specifici metodi delle componenti software. Per ciascun membro del gruppo viene selezionato un metodo rappresentativo del sottosistema di competenza, del quale vengono individuati i parametri di input, le categorie rilevanti e le combinazioni significative per la definizione dei test case.

Il **testing di sistema** è orientato alla validazione del comportamento del sistema durante l'esecuzione di specifici casi d'uso del RAD. Per ogni caso d'uso selezionato, la tecnica Category Partition è utilizzata per derivare test case a partire dai flussi principali e alternativi, includendo sia scenari corretti sia condizioni di errore.

L'approccio adottato consente di garantire la tracciabilità tra requisiti, casi d'uso e test case, nonché di verificare la robustezza del sistema in presenza di input non validi o situazioni anomale.

## 6. Criteri Pass/Fail

L'esito di ciascun test case è determinato mediante un *oracolo*, inteso come il risultato atteso definito sulla base dei requisiti funzionali.

Un test case è considerato **pass** se l'output osservato durante l'esecuzione risulta conforme all'output atteso

Un test case è considerato **fail** se l'output osservato differisce da quello previsto dall'oracolo.

Il superamento della fase di testing è subordinato alla corretta esecuzione dei test case definiti, con particolare attenzione alla gestione degli input non validi e dei flussi alternativi.



## 7. Criteri di sospensione e ripristino

In questa sezione verranno specificati i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprende il test.

### Criteri di Sospensione

L'attività di testing non viene interrotta in seguito al rilevamento di una failure, al fine di garantire l'esecuzione completa dei test case pianificati.

Il testing può essere temporaneamente sospeso esclusivamente in presenza di errori nella definizione dei test case o di condizioni che impediscano la corretta esecuzione dei test stessi.

### Criteri di Ripristino

Il ripristino dell'attività di testing avviene una volta risolte le cause che hanno determinato la sospensione.

Le attività riprendono a partire dal test case che ha causato l'interruzione.

## 8. Materiale di Testing

L'hardware necessario per l'attività di test è un semplice computer, non necessariamente connesso ad internet, in quanto il sistema non è stato ancora rilasciato

## 9. Test cases

L'approccio per la definizione dei test frame sarà il category partition. Al fine di minimizzare il numero di test case, gli input saranno partizionati in classi di equivalenza. Per definire l'output atteso si userà un oracolo umano.





## 9.1 Gestione Checkout e Carrello

### 9.1.1 Confermare un ordine

Parametro: Carrello	
Nome categoria	Scelte per la categoria
Stato [SCAR]	1. Carrello non vuoto= false [error] 2. Carrello non vuoto = true [PROPERTY SCAR_OK]
Validità [VCAR]	1. Carrello null [error] 2. Carrello valido [PROPERTY VCAR_OK]
Parametro: Indirizzo di spedizione	
Nome categoria	Scelte per la categoria
Validità [VIA]	1. ID Indirizzo valido = false [error] 2. ID Indirizzo valido = true [PROPERTY VA_OK]
Seleziona [SE]	1. Indirizzo selezionato = false [error] 2. Indirizzo selezionato = true [Property SE_Ok]



Parametro: Carrello	
Nome categoria	Scelte per la categoria
Parametro: Metodo di pagamento	
Nome categoria	Scelte per la categoria
Validità [VPAY]	1. ID Metodo di pagamento valido = false [error] 2. ID Metodo di pagamento = true [PROPERTY VPAY_OK]

Test Case ID	Test Frame	Esito
TC_GCC7_1	SCAR1	Errore: carrello vuoto
TC_GCC7_2	SCAR2, SE1	Errore: indirizzo non fornito
TC_GCC7_3	SCAR2, SE2, VA1	Errore: indirizzo non valido
TC_GCC7_4	SCAR2, SE2, VA2, PAY1	Errore: metodo di pagamento non valido
TC_GCC7_5	SCAR2, SE2, VA2, PAY2	Corretto



## 9.2 Gestione Autenticazione

### 9.2.1 Registrazione

Parametro: Nome	
FORMATO: <code>^[A-Za-zÀ-ù" -]{2,30}\$</code>	
Nome categoria	Scelte per la categoria
Formato [FNO]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FNO_OK]
Completezza[CNO]	1. Nome vuoto o null = true [error] 2. Nome presente = false [PROPERTY CNO_OK]
Parametro: Utente	
Nome categoria	Scelte per la categoria
Validità [VU]	1. Utente = null [error] 2. Utente valido = [PROPERTY VU_OK]
Parametro: Cognome	
FORMATO: <code>^[A-Za-zÀ-ù" -]{2,30}\$</code>	
Nome categoria	Scelte per la categoria



Parametro: Nome	
Formato [FCO]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FCO_OK]</li> </ol>
Completezza [CCO]	<ol style="list-style-type: none"> <li>1. Cognome vuoto o null = true [error]</li> <li>2. Cognome vuoto o null = false [PROPERTY CCO_OK]</li> </ol>
Parametro: Email	
FORMATO: <code>^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,10}\$</code>	
Nome categoria	Scelte per la categoria
Formato [FE]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FE_OK]</li> </ol>
Esistenza [EE]	<ol style="list-style-type: none"> <li>1. Email già presente nel sistema = true [error]</li> <li>2. Email già presente nel sistema = false [PROPERTY EE_NEW]</li> </ol>
Completezza[CE]	<ol style="list-style-type: none"> <li>1. Email vuoto o null = true [error]</li> <li>2. Email presente = false [PROPERTY CE_OK]</li> </ol>
Parametro: Password	
Nome categoria	Scelte per la categoria



Parametro: Nome	
Lunghezza [LP]	<ol style="list-style-type: none"> <li>1. <code>len(password) &lt; 8 = false [error]</code></li> <li>2. <code>len(password) &gt;= 8 = true [PROPERTY LP_OK]</code></li> </ol>
Completezza [CP]	<ol style="list-style-type: none"> <li>1. Password vuota o null = true [error]</li> <li>2. Password presente = false [PROPERTY CP_OK]</li> </ol>
Parametro: Persistenza	
Nome categoria	Scelte per la categoria
Esito Database [ED]	<ol style="list-style-type: none"> <li>1. Errore durante il salvataggio nel DB [error]</li> <li>2. Salvataggio nel DB riuscito [PROPERTY ED_OK]</li> </ol>

r



Test Case ID	Test Frame	Esito
TC_GA_1	VU1	Errato: utente null
TC_GA_2	VU2, CNO1	Errato: nome vuoto o null
TC_GA_3	VU2, CNO2, FNO1	Errato: nome non rispetta il formato
TC_GA_4	VU2, CNO2, FNO2, CCO1	Errato: cognome vuoto o null
TC_GA_5	VU2, CNO2, FNO2, CCO2, FCO1	Errato: cognome non rispetta il formato
TC_GA_6	VU2, CNO2, FNO2, CCO2, FCO2, CE1	Errato: email vuota o null
TC_GA_7	VU2, CNO2, FNO2, CCO2, FCO2, CE2, FE1	Errato: formato email non corretto
TC_GA_8	VU2, CNO2, FNO2, CCO2, FCO2, CE2, FE2, EE1	Errato: email già registrata
TC_GA_9	VU2, CNO2, FNO2, CCO2, FCO2, CE2, FE2, EE2, CP1	Errato: password vuota o null
TC_GA_10	VU2, CNO2, FNO2, CCO2, FCO2, CE2, FE2, EE2, CP2, LP1	Errato: password troppo corta
TC_GA_11	VU2, CNO2, FNO2, CCO2, FCO2, CE2, FE2, EE2, CP2, LP2, ED1	Errato: errore durante il salvataggio nel database
TC_GA_12	VU2, CNO2, FNO2, CCO2, FCO2, CE2, FE2, EE2, CP2, LP2, ED2	Corretto



## 9.3 Gestione Catalogo e Ricerca

### 9.3.1 Filtraggio del catalogo libri

Parametro: Genere	
Nome categoria	Scelte per la categoria
Validità[PG]	<ol style="list-style-type: none"><li>1. Genere non valido (<math>id \leq 0</math> o formato non numerico) = true <b>[error]</b></li><li>2. Genere non fornito (null o vuoto) <b>[PROPERTY PG_NONE]</b></li><li>3. 3. Genere valido (<math>id &gt; 0</math>) = true <b>[PROPERTY PG_OK]</b></li></ol>
Parametro: Anno di pubblicazione	
Nome categoria	Scelte per la categoria
Validità[PY]	<ol style="list-style-type: none"><li>1. Anno non valido (formato non numerico) = true <b>[error]</b></li><li>2. Anno fuori range (<math>&lt; 1000</math> o <math>&gt;</math> anno corrente) = true <b>[error]</b></li><li>3. Anno valido (<math>1000 \leq \text{anno} \leq \text{anno corrente}</math>) = true <b>[PROPERTY PY_OK]</b></li><li>4. Anno non fornito (null o vuoto) <b>[PROPERTY PY_NONE]</b></li></ol>



Parametro: Genere	
Parametro: Prezzo minimo	
Nome categoria	Scelte per la categoria
Validità[PMIN]	<ol style="list-style-type: none"> <li>1. Prezzo minimo non valido (formato non numerico) = true <b>[error]</b></li> <li>2. Prezzo minimo negativo (<math>&lt; 0</math>) = true <b>[error]</b></li> <li>3. Prezzo minimo valido (<math>\geq 0</math>) = true <b>[PROPERTY PMIN_OK]</b></li> <li>4. Prezzo minimo non fornito (null o vuoto) <b>[PROPERTY PMIN_NONE]</b></li> </ol>
Parametro: Prezzo massimo	
Nome categoria	Scelte per la categoria
Validità[PMAX]	<ol style="list-style-type: none"> <li>1. Prezzo massimo non valido (formato non numerico) = true <b>[error]</b></li> <li>2. Prezzo massimo negativo (<math>&lt; 0</math>) = true <b>[error]</b></li> <li>3. Prezzo massimo valido (<math>\geq 0</math>) = true <b>[PROPERTY PMAX_OK]</b></li> <li>4. Prezzo massimo non fornito (null o vuoto) <b>[PROPERTY PMAX_NONE]</b></li> </ol>





Parametro: Genere	
Parametro: Relazione Prezzi	
Nome categoria	Scelte per la categoria
Coerenza[PREL]	<ol style="list-style-type: none"><li>1. Prezzo minimo &gt; prezzo massimo <b>[error]</b> [IF PMIN_OK AND PMAX_OK]</li><li>2. Prezzo minimo ≤ prezzo massimo <b>[PROPERTY PREL_OK]</b> [IF PMIN_OK AND PMAX_OK]</li></ol>
Parametro: Numero pagine	
Nome categoria	Scelte per la categoria
Validità[PM]	<ol style="list-style-type: none"><li>3. Numero pagine non valido (formato non numerico) = true <b>[error]</b></li><li>4. Numero pagine negativo (&lt; 0) = true <b>[error]</b></li><li>5. Numero pagine valido (≥ 0) = true <b>[PROPERTY PM_OK]</b></li><li>6. Numero pagine non fornito (null o vuoto) <b>[PROPERTY PM_NONE]</b></li></ol>
Parametro: Applicazione Filtri (globale)	



Parametro: Genere	
Nome categoria	Scelte per la categoria
Applicazione[AF]	<ol style="list-style-type: none"><li>1. Nessun filtro applicato = false [PROPERTY AF_NONE]</li><li>2. Uno o più filtri applicati= true [PROPERTY AF_OK]</li></ol>
Parametro: Persistenza	
Nome categoria	Scelte per la categoria
Esito DAO [ED]	<ol style="list-style-type: none"><li>1. Errore durante l'accesso al database <b>[error]</b></li><li>2. Accesso al database riuscito <b>[PROPERTY ED_OK]</b></li></ol>



Test Case ID	Test Frame	Esito
TC_GCR_2_1	AF2, PMIN2	Errore: prezzo minimo negativo
TC_GCR_2_2	AF2, PY2	Errore: anno fuori range (> anno corrente)
TC_GCR_2_3	AF2, PM2	Errore: numero di pagine negativo
TC_GCR_2_4	AF2, PG1	Errore: genere non valido ( $id \leq 0$ )
TC_GCR_2_5	AF1	Corretto: catalogo senza filtri
TC_GCR_2_6	AF2, PG3	Corretto: catalogo filtrato per genere
TC_GCR_2_7	AF2, PMIN3, PMAX3, PREL2	Corretto: catalogo filtrato per prezzo
TC_GCR_2_8	AF2, PY3	Corretto: catalogo filtrato per anno



TC_GCR_2_9	AF2, PM3	Corretto: catalogo filtrato per numero di pagine
TC_GCR_2_10	AF2, PG3, PY3, PM3, PMIN3, PMAX3, PREL2	Corretto: catalogo filtrato per più criteri
TC_GCR_2_11	AF2, PMAX2	Errore: prezzo massimo negativo
TC_GCR_2_12	AF2, PMIN3, PMAX3, PREL1	Errore: prezzo minimo > prezzo massimo
TC_GCR_2_13	AF2, PMIN1	Errore: formato prezzo minimo non valido
TC_GCR_2_14	AF2, PMAX1	Errore: formato prezzo massimo non valido
TC_GCR_2_15	AF2, PY1	Errore: formato anno non valido
TC_GCR_2_16	AF2, PY2	Errore: anno < 1000
TC_GCR_2_17	AF2, PM1	Errore: formato numero pagine non valido



TC_GCR_2_18	AF2, PG1	Errore: formato genere non valido
TC_GCR_2_19	AF2, ED1	Errore: eccezione durante ricerca nel database
TC_GCR_2_20	AF2, PMIN3, PMAX4	Corretto: solo prezzo minimo fornito
TC_GCR_2_21	AF2, PMIN4, PMAX3	Corretto: solo prezzo massimo fornito

## 9.4 Gestione Recensioni

### 9.4.1 Inserimento di una recensione

Parametro: Titolo recensione	
Nome categoria	Scelte per la categoria
Completezza [CT]	1. Titolo inserito = false [error] 2. Titolo inserito = true [PROPERTY CT_OK]
Parametro: Testo recensione	
Nome categoria	Scelte per la categoria



Parametro: Titolo recensione	
Completezza [CX]	1. Testo inserito = false [error] 2. Testo inserito = true [PROPERTY CX_OK]
Parametro: Rating	
Nome categoria	Scelte per la categoria
Selezione [CR]	1. Rating selezionato= false [error] 2. Rating selezionato (1-5) = true [PROPERTY CR_OK]

Test Case ID	Test Frame	Esito
TC_GR_1	CT1	Errato: titolo mancante
TC_GR_2	CT2, CX1	Errato: testo mancante
TC_GR_3	CT2, CX2, CR1	Errato: rating mancante
TC_GR_4	CT2, CX2, CR2	Corretto

## 10 Testing schedule

Le attività di pianificazione del testing avverranno come definito nei capitoli precedenti, cioè subito dopo la fase di design necessaria per la pianificazione.

La scrittura dei casi di test avverrà in contemporanea con lo sviluppo del codice. L'esecuzione dei test avverrà sia durante che dopo l'implementazione del sistema.

Una volta concluso lo sviluppo, tutti i test saranno rieseguiti per garantirne il corretto funzionamento e produrre i report finali.