



TSR

# TEST SUMMARY REPORT

## ModuLink

Riferimento	NC08_TSR_ver.1.0
Versione	1.0
Data	19/01/2026
Destinatario	Studenti di Ingegneria del Software 2025/26
Presentato da	Buzi Arjel, Carpentieri Daniele, Chikviladze Aleksandre, Cito Roberto.
Codice Gruppo	NC08
Approvato da	



## Revision History

Data	Versione	Descrizione	Autori
19/01/2026	0.1	Prima stesura	Aleksandre
20/01/2026	0.2	Correzioni vari	Aleksandre
20/01/2026	1.0	Stesura Definitiva e Revisione	Aleksandre



## Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
<b>Roberto Cito</b>	Team Member	RC	<a href="mailto:r.cito@studenti.unisa.it">r.cito@studenti.unisa.it</a>
<b>Daniele Carpentieri</b>	Team Member	DC	<a href="mailto:d.carpentieri8@studenti.unisa.it">d.carpentieri8@studenti.unisa.it</a>
<b>Aleksandre Chikviladze</b>	Team Member	AC	<a href="mailto:a.chikviladze@studenti.unisa.it">a.chikviladze@studenti.unisa.it</a>
<b>Arjel Buzi</b>	Team Member	AB	<a href="mailto:a.buzi@studenti.unisa.it">a.buzi@studenti.unisa.it</a>



## Summary

Revision History .....	2
Team members .....	3
1: INTRODUZIONE.....	5
1.1 Scopo del Documento .....	5
1.2 Scopo del Prodotto .....	5
1.3 Acronimi e Definizioni.....	5
1.4 Documenti di Riferimento .....	6
2: TESTING DI UNITÀ .....	6
2.1 Metodologia e Strumenti.....	6
2.2 Analisi della Code Coverage.....	7
3: TESTING DI SISTEMA.....	7
3.1 Ambiente ed Esecuzione .....	7
3.2 Riepilogo Risultati per Sottosistema.....	8
3.3 Analisi dei Difetti Risolti .....	8
4: VALUTAZIONE .....	9
4.1 Varianze (Differenze dal Piano) .....	9
4.2 Valutazione Complessiva .....	9
5: CONCLUSIONI .....	10
5.1 Riepilogo delle Attività .....	10
5.2 Valutazione Finale .....	10



# 1: INTRODUZIONE

## 1.1 Scopo del Documento

Il presente documento costituisce il **Test Summary Report (TSR)** del progetto **ModuLink**. Lo scopo di questo rapporto è riassumere i risultati delle attività di test eseguite sul sistema software (versione 1.0), fornendo una valutazione oggettiva della qualità del prodotto finale. Il documento analizza l'esito dei **Test di Unità** (White-box) e dei **Test di Sistema** (Black-box), documentando la copertura del codice, gli incidenti rilevati e risolti, e la conformità rispetto ai requisiti definiti nel RAD (*Requirements Analysis Document*). Questo report serve come base formale per la decisione di rilascio del software.

## 1.2 Scopo del Prodotto

ModuLink è una piattaforma web modulare di tipo SaaS (*Software as a Service*) progettata per la gestione integrata dei processi aziendali. Il sistema permette la configurazione dinamica di funzionalità tramite l'installazione di **Moduli** (es. GTM - Task Manager, GDM - Magazzino, GCA - Calendario) e gestisce l'accesso multi-tenant sicuro per diverse aziende, regolando i permessi tramite un sistema RBAC (*Role-Based Access Control*).

## 1.3 Acronimi e Definizioni

ACRONIMO	DEFINIZIONE
<b>TSR</b>	Test Summary Report.
<b>TIR</b>	Test Incident Report.
<b>TPD</b>	Test Plan Document.
<b>TCSD</b>	Test Case Specification Document.
<b>SUT</b>	System Under Test (Il sistema oggetto del test).
<b>LOC</b>	Lines Of Code (Linee di codice).
<b>Coverage</b>	Percentuale di codice eseguito durante i test automatici.
<b>JUnit</b>	Framework per il testing di unità in ambiente Java.
<b>Selenium</b>	Framework per l'automazione dei test funzionali web.



## 1.4 Documenti di Riferimento

La stesura di questo report fa riferimento alla seguente documentazione approvata:

- [RAD] Requirements Analysis Document v1.4
- [SDD] System Design Document v1.2
- [TPD] Test Plan Document v0.5
- [TCSD] Test Case Specification Document v0.5

## 2: TESTING DI UNITÀ

L'attività di testing di unità (*Unit Testing*) ha avuto l'obiettivo di verificare il corretto funzionamento delle singole componenti software (classi e metodi) in isolamento, prima della loro integrazione nel sistema completo. L'approccio adottato è di tipo **White-box**, basato sulla conoscenza della struttura interna del codice sorgente.

### 2.1 Metodologia e Strumenti

Per l'automazione e l'esecuzione dei test sono stati utilizzati i seguenti strumenti, integrati nell'ambiente di sviluppo (IDE IntelliJ IDEA) e nella pipeline di Continuous Integration (GitHub Actions):

- **JUnit 5:** Framework principale utilizzato per la scrittura dei Test Case e la gestione delle asserzioni (verifiche *Expected* vs *Actual*).
- **Mockito:** Libreria utilizzata per la creazione di oggetti simulati (*Mock*). Data l'architettura modulare di ModuLink, Mockito è stato fondamentale per testare i singoli Service (es. TaskService) isolandoli dalle dipendenze esterne (es. Database o altri Service).
- **JaCoCo (Java Code Coverage):** Strumento utilizzato per misurare la copertura del codice, ovvero la percentuale di istruzioni ed esecuzioni condizionali (branch) sollecitate durante i test.

Metrica	Valore Raggiunto	Obiettivo TPD	Esito
Line Coverage (Linee di codice eseguite)	87%	> 80%	Soddisfatto
Branch Coverage (Rami condizionali if/else coperti)	79%	> 70%	Soddisfatto



**Method Coverage (Metodi invocati almeno una volta)**

92%

> 90%

Soddisfatto

## 2.2 Analisi della Code Coverage

L'obiettivo prefissato nel TPD era il raggiungimento di una *Line Coverage* superiore all'80% per i package critici (Service e Controller).

Di seguito sono riportati i risultati finali rilevati tramite JaCoCo sulla versione v1.0-RELEASE:

### Dettaglio per Package

Package	Line Coverage	Note
com.modulink.service	94%	Copertura quasi totale della Business Logic critica (GTM, GDM, GDU).
com.modulink.controller	82%	Testati gli endpoint REST e la gestione delle richieste HTTP.
com.modulink.model	98%	Copertura elevata su Entity e DTO (Getters, Setters, Costruttori).
com.modulink.security	85%	Verifica delle catene di filtri di Spring Security.
com.modulink.util	76%	Classi di utilità e helper.

I risultati evidenziano un'ottima stabilità del codice sorgente. Le parti non coperte riguardano principalmente gestori di eccezioni rare (es. errori di connessione al DB irreversibili) difficilmente riproducibili in ambiente di test standard.

## 3: TESTING DI SISTEMA

L'attività di Testing di Sistema ha avuto l'obiettivo di validare i requisiti funzionali dell'intera applicazione ModuLink, simulando le interazioni reali dell'utente finale tramite l'interfaccia web (GUI). L'approccio adottato è di tipo **Black-box**.

### 3.1 Ambiente ed Esecuzione

Le sessioni di test sono state eseguite nel seguente ambiente:

- **Browser:** Google Chrome (v120+) e Mozilla Firefox.
- **Database:** MySQL 8.0 (con dataset di test pre-popolato).



- **Strumenti:** Esecuzione manuale dei casi d'uso complessi e automazione tramite script **Selenium IDE** per i flussi ripetitivi (es. Login/Logout).

Ogni Test Case definito nel documento *TCSD* è stato eseguito. In caso di discrepanza tra il risultato atteso e quello ottenuto, è stato aperto un incidente nel *TIR* (Test Incident Report), seguito dalla correzione del codice e dalla riesecuzione del test (*Regression Testing*).

### 3.2 Riepilogo Risultati per Sottosistema

Di seguito sono riportati i risultati aggregati per ogni modulo funzionale.

Test Suite (Sottosistema)	# Test Case Totali	# Fallimenti Iniziali	# Fallimenti Finali	Note e Incidenti Rilevati
GDU (Gestione Utenti)	22	1	0	Rilevato incidente sulla lunghezza minima password ( <b>TC2_GDU.1</b> ). Bug risolto e test passato alla seconda esecuzione.
GTM (Task Manager)	9	1	0	Rilevato incidente sulla validazione data scadenza nel passato ( <b>TC5_GTM.1</b> ). Bug risolto.
GDM (Magazzino)	11	1	0	Rilevato incidente sull'inserimento di prezzi negativi ( <b>TC7_GDM.1</b> ). Aggiunta validazione backend.
GDE (Calendario)	11	0	0	Nessuna anomalia riscontrata nella gestione eventi.
<b>TOTALE</b>	<b>53</b>	<b>3</b>	<b>0</b>	<b>Tasso di Successo Finale: 100%</b>

### 3.3 Analisi dei Difetti Risolti

Durante la prima iterazione di test sono emerse alcune non conformità critiche che sono state tracciate e risolte prima del rilascio finale:

1. **Incidente su GDU (Sicurezza):** Il sistema accettava password inferiori a 8 caratteri.
  - *Azione Correttiva:* Aggiornata la logica di validazione nel RegisterController. Il test TC2\_GDU.1 ora restituisce correttamente l'errore [ERR] Lunghezza insufficiente.
2. **Incidente su GDM (Integrità Dati):** Era possibile inserire prodotti con prezzo negativo o quantità non numeriche.



- *Azione Correttiva:* Implementato controllo rigoroso nel MagazzinoService. Il test TC7\_GDM.1 ora impedisce il salvataggio e mostra un alert all'utente.
- 3. **Incidente su GTM (Logica Business):** Il sistema permetteva la creazione di Task con scadenza antecedente alla data di creazione.
  - *Azione Correttiva:* Aggiunto controllo if (date.isBefore(now)) nel metodo di creazione.

Alla data del presente documento, **tutti i Test Case passano con esito positivo** (Passed).

## 4: VALUTAZIONE

In questa sezione viene analizzata la qualità complessiva del processo di testing e del prodotto software, evidenziando eventuali scostamenti rispetto al piano originale (*Test Plan Document*).

### 4.1 Varianze (Differenze dal Piano)

Rispetto a quanto pianificato nel documento *TPD (Test Plan Document v0.5)*, le attività di test hanno seguito la schedulazione prevista. È stata tuttavia introdotta una variazione procedurale positiva:

- **Test Esplorativi Aggiuntivi:** Oltre ai Test Case formali descritti nel TCSD, sono state eseguite sessioni non pianificate di *Exploratory Testing* manuale per verificare la responsività dell'interfaccia grafica (Dashboard e Sidebar) su diverse risoluzioni di schermo. Questa attività ha permesso di affinare l'esperienza utente (UX), pur non essendo strettamente legata a requisiti funzionali critici.

Non sono state registrate variazioni negative (es. test saltati per mancanza di tempo); tutti i moduli previsti (GDU, GTM, GDM, GCA, GDR) sono stati coperti.

### 4.2 Valutazione Complessiva

Sulla base dei dati raccolti nel Capitolo 2 (Coverage > 80%) e nel Capitolo 3 (Zero difetti residui), la valutazione della qualità del software **ModuLink v1.0** è positiva.

- **Robustezza:** Il sistema gestisce correttamente gli input non validi (es. date passate, prezzi negativi) senza crash o comportamenti imprevisti, grazie ai meccanismi di validazione implementati nel Backend.
- **Sicurezza:** I test sui ruoli (RBAC) confermano che gli utenti non autorizzati non possono accedere a moduli o dati riservati.
- **Affidabilità:** Durante l'intera fase di regression testing, il server non ha mostrato perdite di connessione o corruzione dei dati.

Gli incidenti rilevati durante la prima fase (cfr. *Test Incident Report*) sono stati tutti chiusi con successo. Allo stato attuale, **non esistono "Known Bugs" (Bug Noti) di severità Alta o Critica**.



## 5: CONCLUSIONI

### 5.1 Riepilogo delle Attività

Le attività di verifica e validazione si sono concluse il 22/01/2026. Di seguito i dati riepilogativi dello sforzo di testing:

- **Durata Totale Testing:** Circa 30 ore-uomo (distribuite tra Unit Test, Integrazione e Sistema).
- **Test Case Eseguiti:** 60 Test di Sistema + suite completa di Unit Test automatizzati.
- **Copertura Codice:** 87% (Line Coverage).
- **Ambiente di Riferimento:** Server Apache Tomcat locale, Database MySQL 8.0, Client Chrome/Firefox.

### 5.2 Valutazione Finale

Il team di sviluppo **NC08**, analizzati i risultati del presente rapporto, dichiara che il software **ModuLink**:

1. Soddisfa tutti i **Requisiti Funzionali (RF)** prioritari definiti nel RAD.
2. Rispetta i criteri di accettazione qualitativa stabiliti nello Statement of Work (SOW).
3. È privo di difetti bloccanti che ne impediscono l'utilizzo in ambiente di produzione.

Pertanto, il sistema viene considerato **IDONEO AL RILASCIO (Ready for Release)** nella sua versione **1.0**.