



Test Plan



Riferimento	NC24_TP_ver.0.8
Versione	0.8
Data	08/01/2026
Destinatario	Prof. Carmine Gravino
Presentato da	Gruppo NC24
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
08/01/2026	0.1	Prima stesura	Paolo Visconti
09/01/2026	0.2	Definizione features da testare	Gabriele De Luca, Simone Sammartano
13/01/2026	0.3	Aggiunta dei primi test case	Mario Branca
14/01/2026	0.4	Aggiunta di altri test case	Mario Branca, Paolo Visconti
16/01/2026	0.5	Completamento dei test case	Gabriele De Luca
21/01/2026	0.6	Correzione di alcuni errori	Simone Sammartano
02/02/2026	0.7	Revisione dei test case	Paolo Visconti
07/02/2026	0.8	Sistemazione per la consegna finale	Tutto il team

Team Members

Nome	Ruolo	Acronimo	Informazioni di contatto
Mario Branca	Team Member	M.B.	m.branca2@studenti.unisa.it
Gabriele De Luca	Team Member	G.D.L.	g.deluca65@studenti.unisa.it
Simone Sammartano	Team Member	S.S.	s.sammartano1@studenti.unisa.it
Paolo Visconti	Team Member	P.V.	p.visconti4@studenti.unisa.it



Sommario

Revision History.....	2
Team Members.....	2
Sommario.....	3
1. Introduzione.....	4
2. Relazione con altri documenti.....	4
3. Panoramica del sistema.....	4
4. Funzionalità da testare.....	5
5. Pass/Fail criteria.....	6
6. Approccio.....	6
7. Materiale di testing.....	7
8. Test cases.....	7
8.1 Gestione Utente.....	8
8.1.1 Registrazione cliente.....	8
8.1.2 Login.....	10
8.1.3 Modifica dati account.....	11
8.2 Gestione Acquisto.....	14
8.2.1 Inserimento indirizzo di spedizione.....	14
8.2.2 Inserimento dati di pagamento.....	15
8.3 Ricerca Libro.....	16
8.3.1 Ricerca per nome (titolo).....	16
9. Testing schedule.....	17



1. Introduzione

Il presente documento di Test Plan descrive le attività di testing previste per la piattaforma Readify, con l'obiettivo di verificare il corretto funzionamento delle principali funzionalità del sistema.

Nel documento vengono illustrate le strategie di testing adottate, l'ambito delle funzionalità sottoposte a verifica e gli strumenti utilizzati per individuare eventuali malfunzionamenti. Le attività di testing sono finalizzate a garantire un comportamento affidabile del sistema e a ridurre il rischio di errori durante l'utilizzo della piattaforma. In particolare, sono state pianificate attività di testing per le seguenti aree funzionali:

- **Gestione Utente**
- **Gestione Acquisto**
- **Ricerca Libro**

2. Relazione con altri documenti

I test case descritti in questo documento sono stati definiti a partire dalla documentazione di sistema prodotta durante le fasi di analisi e progettazione.

In particolare, le attività di testing fanno riferimento ai requisiti funzionali e non funzionali descritti nel Requirements Analysis Document (RAD), che costituiscono la base per la verifica del comportamento atteso del sistema.

L'organizzazione dei test tiene inoltre conto della suddivisione del sistema in sottosistemi e componenti descritta nel System Design Document (SDD), in modo da mantenere coerenza con l'architettura complessiva dell'applicazione.

Il testing di unità e di integrazione, più strettamente legato alla struttura interna delle classi e dei package, viene documentato direttamente all'interno del codice. Per questo motivo, il presente documento si concentra principalmente sulle attività di testing a livello di sistema.

3. Panoramica del sistema

Il sistema è sviluppato secondo un'**architettura three-tier**, che separa il livello di presentazione, la logica applicativa e la gestione dei dati.



Il **livello di presentazione** è realizzato tramite pagine JSP, utilizzate per mostrare le interfacce utente, che non sono direttamente accessibili, mentre la gestione della navigazione e dei controlli di accesso è affidata alle Servlet.

La **logica applicativa** è implementata in Java e organizzata in componenti di servizio che gestiscono le principali operazioni del sistema, come la gestione degli utenti, del carrello e degli ordini. Le Servlet fungono da controller, coordinando le richieste dell'utente e l'interazione con la logica di business.

La **persistenza dei dati** è affidata a un **DBMS relazionale MySQL**, messo a disposizione dall'ateneo, e hostato in rete tramite **Microsoft Azure**, messo a disposizione gratuitamente da Microsoft, utilizzati come repository dei dati del sistema.

4. Funzionalità da testare

Le attività di testing scelte si concentrano sulle principali funzionalità del sistema che prevedono interazioni dirette con l'utente e che risultano più critiche per il corretto funzionamento della piattaforma.

Le funzionalità oggetto di testing sono le seguenti:

Gestione Utenti

- Registrazione del cliente
- Login
- Modifica dei dati dell'account

Gestione Acquisto

- Inserimento dell'indirizzo di spedizione
- Inserimento dei dati di pagamento

Ricerca Libro

- Ricerca per nome

Non sono state considerate nell'ambito del piano di test le funzionalità ritenute di minore criticità o che non prevedono input manuale da parte dell'utente, come le semplici visualizzazioni di dati o interazioni che non comportano modifiche allo stato del sistema.



5. Pass/Fail criteria

Le attività di testing sono finalizzate a individuare eventuali malfunzionamenti del sistema, al fine di consentire un intervento correttivo prima della messa in esercizio della piattaforma.

L'esito di ciascun test case è valutato confrontando il comportamento osservato del sistema con il risultato atteso, definito dall'oracolo sulla base dei requisiti.

Un **test case** è considerato **superato** (pass) se, a fronte di un determinato input, il comportamento del sistema risulta conforme all'output atteso.

Un **test case** è considerato **fallito** (fail) se il comportamento osservato del sistema differisce da quanto definito dall'oracolo.

Le attività di testing sono ritenute soddisfacenti qualora vengano rispettati i seguenti criteri:

- esecuzione di test di regressione in caso di introduzione o modifica di funzionalità del sistema;
- raggiungimento di un livello di branch coverage non inferiore al 50%.

6. Approccio

Nel piano di testing sono stati individuati due livelli principali di verifica del software:

- Il **testing funzionale** è orientato alla verifica del comportamento delle funzionalità del sistema dal punto di vista dell'utente, validando i requisiti funzionali sulla base dei casi d'uso e dei flussi di interazione previsti. Questo livello di testing consente di controllare che il sistema risponda correttamente sia in presenza di input validi sia in presenza di condizioni non corrette o inattese.
- Il **testing di unità**, invece, è focalizzato sulla verifica dei singoli metodi delle classi che implementano la logica applicativa del sistema, come i componenti di servizio e di accesso ai dati. L'obiettivo è garantire che le singole unità di codice funzionino correttamente in modo isolato, riducendo il rischio di errori durante l'integrazione.

Per la definizione dei test viene adottata la tecnica di **category partition**, applicata secondo una **metodologia black-box**, che permette di individuare insiemi significativi di



input e di stati del sistema e di derivare casi di test rappresentativi senza dipendere dai dettagli implementativi.

I test di unità sono implementati utilizzando strumenti di testing per l'ambiente Java, come **JUnit** e **Mockito**, mentre la copertura dei test viene monitorata tramite **JaCoCo**, con particolare attenzione al branch coverage.

Al fine di rendere il processo di verifica ripetibile e affidabile, l'esecuzione della suite di test è integrata in una pipeline di continuous integration, che consente di eseguire automaticamente i test a ogni aggiornamento del sistema e di individuare tempestivamente eventuali regressioni.

7. Materiale di testing

Per lo svolgimento delle attività di testing è sufficiente un computer in grado di eseguire l'applicazione e gli strumenti di test previsti. È necessaria una connessione a Internet per l'esecuzione dei test che interagiscono con i dati persistenti, poiché il database è ospitato su infrastruttura cloud. L'ambiente di test include un servlet container per l'esecuzione dell'applicazione, un database remoto gestito su Microsoft Azure per la persistenza dei dati e gli strumenti di testing utilizzati per l'esecuzione automatica dei test di unità e di sistema., I test possono essere eseguiti anche in ambiente locale utilizzando configurazioni alternative o dataset di prova, mantenendo separati i dati di test dai dati di eventuali ambienti dimostrativi.

8. Test cases

La presente sezione descrive i test case definiti per le funzionalità del sistema Readify, utilizzando la tecnica di **Category Partition** secondo un approccio **black-box**.

I test sono progettati a partire dai requisiti funzionali e dai casi d'uso descritti nella documentazione di sistema, con l'obiettivo di verificare il comportamento del sistema in presenza di input validi, input non validi e condizioni di errore significative.

Per ciascuna funzionalità vengono individuati:

- i **parametri rilevanti**;
- le **categorie** associate a ciascun parametro;
- le **scelte** (partizioni) per ogni categoria;

- una selezione di **test case rappresentativi**, derivati dalla combinazione delle partizioni più significative.

8.1 Gestione Utente

8.1.1 Registrazione cliente

Parametro: Nome	
Nome Categoria	Scelte per la categoria
Mandatory [MNO]	1. campo compilato = false[error] 2. campo compilato = true[PROPERTY MNO_OK]
Parametro: Cognome	
Nome Categoria	Scelte per la categoria
Mandatory [MCO]	1. campo compilato = false[error] 2. campo compilato = true[PROPERTY MCO_OK]
Parametro: Email	
Formato	
^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\$	
Nome Categoria	Scelte per la categoria
Formato [FEM]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FEM_OK]
Parametro: Telefono	
Formato	
^\\+?[0-9\\s-]{6,}\$	
Nome Categoria	Scelte per la categoria
Formato [FTE]	1. rispettare il formato = false[error]



	2. rispettare il formato = true[PROPERTY FTE_OK]
Parametro: Indirizzo di Spedizione	
Nome Categoria	Scelte per la categoria
Mandatory [MIN]	1. campo compilato = false[error] 2. campo compilato = true[PROPERTY MIN_OK]
Parametro: Città	
Nome Categoria	Scelte per la categoria
Mandatory [MCI]	1. campo compilato = false[error] 2. campo compilato = true[PROPERTY MCI_OK]
Parametro: CAP	
Formato $^{\wedge} \backslash d \{ 5 \} \$$	
Nome Categoria	Scelte per la categoria
Formato [FCAP]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FCAP_OK]
Parametro: Provincia	
Formato $^{\wedge} [A-Za-z] \{ 2 \} \$$	
Nome Categoria	Scelte per la categoria
Formato [FPR]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FPR_OK]
Parametro: Password	
Formato $^{\wedge} (? = . * \backslash d) (? = . * [a-z]) (? = . * [A-Z]) . \{ 8 , \} \$$	
Nome Categoria	Scelte per la categoria



Formato [FPA]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FPA_OK]
Parametro: Conferma Password	
Nome Categoria	Scelte per la categoria
Match [MCP]	1. match con password = false[error] 2. match con password = true[PROPERTY MCP_OK]

Test Case ID	Test frame	Esito
TC1.1.1	FPA	Errato: Password non rispetta il formato
TC1.1.2	MCP	Errato: Match con Password errato
TC1.1.3	FEM	Errato: Email non rispetta il formato
TC1.1.4	FCAP	Errato: CAP non rispetta il formato
TC1.1.5	FPR	Errato: Provincia non rispetta il formato
TC1.1.6	FTE	Errato: Telefono non rispetta il formato

8.1.2 Login

Parametro: Email	
Nome Categoria	Scelte per la categoria
Match [MEM]	1. match con email = false[error] 2. match con email = true[PROPERTY MEM_OK]



Test Case ID	Test frame	Esito
TC1.2.1	MEM	Errato: Match con Email errato
TC1.2.2	FEM	Errato: Email non rispetta il formato
TC1.2.3	MPA	Errato: Match con Password errato

8.1.3 Modifica dati account

Parametro: Password Attuale	
Nome Categoria	Scelte per la categoria
Match [MPAT]	<ol style="list-style-type: none">1. match con password attuale = false[error]2. match con password attuale = true[PROPERTY MPAT_OK]



Parametro: Password	
Formato $^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}$$	
Nome Categoria	Scelte per la categoria
Formato [FPA]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FPA_OK]
Parametro: Conferma Password	
Nome Categoria	Scelte per la categoria
Match [MCP]	1. match con password = false[error] 2. match con password = true[PROPERTY MCP_OK]
Parametro: Email	
Formato $^[A-Za-z0-9._%+]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$$	
Nome Categoria	Scelte per la categoria
Formato [FEM]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FEM_OK]
Parametro: CAP	
Formato $^\d{5}$$	
Nome Categoria	Scelte per la categoria
Formato [FCAP]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FCAP_OK]
Parametro: Provincia	
Formato $^[A-Za-z]{2}$$	
Nome Categoria	Scelte per la categoria



Formato [FPR]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FPR_OK]
Parametro: Telefono	
Formato <code>^\\+?[0-9\\s-]{6,}\$</code>	
Nome Categoria	Scelte per la categoria
Formato [FTE]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FTE_OK]

Test Case ID	Test frame	Esito
TC1.3.1	MPAT	Errato: Match con Password Attuale errato
TC1.3.2	FPA	Errato: Password non rispetta il formato
TC1.3.3	MCP	Errato: Match con Password errato
TC1.3.4	FEM	Errato: Email non rispetta il formato
TC1.3.5	FCAP	Errato: CAP non rispetta il formato
TC1.3.6	FPR	Errato: Provincia non rispetta il formato
TC1.3.7	FTE	Errato: Telefono non rispetta il formato

8.2 Gestione Acquisto

8.2.1 Inserimento indirizzo di spedizione

Parametro: Indirizzo di Spedizione	
Nome Categoria	Scelte per la categoria
Mandatory [MIN]	1. campo compilato = false[error] 2. campo compilato = true[PROPERTY MIN_OK]
Parametro: Città	
Nome Categoria	Scelte per la categoria
Mandatory [MCI]	1. campo compilato = false[error] 2. campo compilato = true[PROPERTY MCI_OK]
Parametro: CAP	
Formato $^{\backslash}d\{5\}$$	
Nome Categoria	Scelte per la categoria
Formato [FCAP]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FCAP_OK]
Parametro: Provincia	
Formato $^{\backslash}[A-Za-z]\{2\}$$	
Nome Categoria	Scelte per la categoria
Formato [FPR]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FPR_OK]

Test Case ID	Test frame	Esito
--------------	------------	-------



TC2.1.1	FCAP	Errato: CAP non rispetta il formato
TC2.1.2	FPR	Errato: Provincia non rispetta il formato

8.2.2 Inserimento dati di pagamento

Parametro: Titolare della Carta	
Nome Categoria	Scelte per la categoria
Mandatory [MTIC]	1. campo compilato = false[error] 2. campo compilato = true[PROPERTY MTIC_OK]
Parametro: Numero Carta	
Formato $\wedge d\{16\}\$$	
Nome Categoria	Scelte per la categoria
Formato [FNUC]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FNUC_OK]
Parametro: Scadenza	
Formato $\wedge (0[1-9] 1[0-2])\wedge d\{2\}\$$	
Nome Categoria	Scelte per la categoria
Formato [FSC]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FSC_OK]
Parametro: CVV	
Formato $\wedge d\{3\}\$$	



Nome Categoria	Scelte per la categoria
Formato [FCVV]	1. rispettare il formato = false[error] 2. rispettare il formato = true[PROPERTY FCVV_OK]

Test Case ID	Test frame	Esito
TC2.2.1	FNUC	Errato: Numero Carta non rispetta il formato
TC2.2.2	FSC	Errato: Scadenza non rispetta il formato
TC2.2.3	FCVV	Errato: CVV non rispetta il formato

8.3 Ricerca Libro

8.3.1 Ricerca per nome (titolo)

Parametro: Ricerca	
Nome Categoria	Scelte per la categoria
Contenuto Ricerca [CRI]	1. trova risultato = false[error] 2. trova risultato = true[PROPERTY CRI_OK]

Test Case ID	Test frame	Esito
TC3.1.1	CRI	Errato: Ricerca non produce risultati



9. Testing schedule

Le attività di pianificazione del testing vengono avviate subito dopo la fase di design del sistema, così da definire in anticipo l'approccio e i criteri di verifica.

La definizione e la scrittura dei casi di test procedono in parallelo allo sviluppo del codice, permettendo di verificare progressivamente le funzionalità man mano che vengono implementate.

L'esecuzione dei test viene effettuata sia durante la fase di sviluppo sia al termine dell'implementazione del sistema, al fine di individuare tempestivamente eventuali errori e prevenire regressioni.

Una volta concluso lo sviluppo, l'intera suite di test viene eseguita nuovamente per confermare il corretto funzionamento complessivo del sistema prima della messa in esercizio.