



TP

Test Plan

ReVamp Ascent

<b>Riferimento</b>	C07_TP_ver.1
<b>Versione</b>	1.0
<b>Data</b>	20/12/2025
<b>Destinatario</b>	Prof. C.Gravino

Revision History

Data	Versione	Descrizione	Autori
------	----------	-------------	--------



18/11/2025	0.1	Prima stesura	VM, AA
02/12/2025	0.2	Creazione Test Frame	VM
03/12/2025	0.3	Stesura parti testuali	VM
03/12/2025	0.4	Completamento Test Frame	VM
04/12/2025	0.5	Prima revisione	Tutto il team
06/12/2025	1.0	Revisione finale e correzione degli errori	VM
11/12/2025	1.1	Aggiunta sezione per il Test di Unità	VM
20/12/2025	1.2	Scrittura dei capitoli riguardanti il test di integrazione e dello schedule	VM
20/01/2025	1.3	Modifiche sezione 9.3.2 Accettazione ticket prestito	VM
21/01/2025	2.0	Revisione	Tutto il team

## Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Antonio Aliberti	Team Member	AA	a.aliberti56@studenti.unisa.it



Vincenzo Martucci	Team Member	VM	v.martucci5@studenti.unisa.it
Raffaella Di Pasquale	Team Member	RDP	r.dipasquale5@studenti.unisa.it

## Sommario

Revision History .....	1
Team members .....	2
1 Introduzione .....	4
2 Relazione con altri documenti.....	4
3 Panoramica del sistema .....	5
4 Features da testare/da non testare .....	6
5 Pass/Fail criteria .....	7
6 Approccio .....	8
7 Sospensione e ripristino .....	12
8 Materiale di testing.....	12
9 Test cases.....	12
9.1.4 Nuovo metodo di pagamento .....	18
10 Testing schedule .....	33



## 1 Introduzione

---

ReVamp Ascent è una piattaforma e-commerce pensata per valorizzare la vendita di prodotti di arredo e design, offrendo un'esperienza di acquisto online moderna, curata e altamente usabile, ispirata agli standard dei grandi player del settore. L'obiettivo è quello di semplificare e rendere più intuitiva l'interazione tra utente e catalogo prodotti, guidando il cliente lungo l'intero processo d'acquisto: dalla navigazione del catalogo, alla gestione del carrello, fino al pagamento e al post-vendita.

Il documento di Test Plan ha l'obiettivo di descrivere e analizzare le attività di Testing previste per la piattaforma ReVamp Ascent, al fine di garantire che ogni funzionalità implementata operi in modo corretto, affidabile e conforme ai requisiti individuati in fase di analisi.

All'interno del documento sono riportate le strategie di testing adottate, le funzionalità oggetto di verifica e gli strumenti utilizzati per la rilevazione e la gestione degli errori, con lo scopo di presentare all'utente finale una piattaforma priva, per quanto possibile, di malfunzionamenti e in grado di supportare un flusso d'acquisto completo.

Sono state pianificate attività di testing per le seguenti gestioni principali:

- **Gestione Utente** (registrazione, login, profilo, dati personali)
- **Gestione Catalogo Prodotti** (inserimento e modifica prodotti nel Catalogo, filtri, ricerca, dettaglio prodotto)
- **Gestione Carrello e Ordine** (aggiunta/rimozione prodotti nel carrello, riepilogo, conferma ordine)
- **Gestione Recensioni**

## 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 e non funzionali presentati nel RAD.

### **Relazioni con il System Design Document (SDD)**

I test case pianificati nel Test Plan devono rispettare la suddivisione in sottosistemi presentata nell'SDD.



## Relazioni con il Object Design Document (ODD)

Per ciò che concerne il test di unità e di integrazione, maggiormente legati allo ODD e alla divisione in package del sistema, essi saranno scritti e documentati unicamente all'interno del codice dell'applicativo. Per tale motivo, nel presente documento, non vi saranno riferimenti al loro design.

## 3 Panoramica del sistema

---

Il sistema **ReVamp Ascent** è realizzato secondo un'architettura **three-tier**, con una chiara separazione tra:

- **Presentation Tier (Front-end)**: responsabile dell'interazione con l'utente e della visualizzazione delle pagine.
- **Buisness Tier (Back-end)**: contiene la logica applicativa e gestisce le richieste.
- **Data Tier (Database)**: si occupa della persistenza e della gestione dei dati.

### *Presentation Tier (Front-end)*

La parte di presentazione è sviluppata utilizzando tecnologie web standard e pagine dinamiche, in particolare:

- **HTML** e **CSS** per struttura e stile dell'interfaccia.
- **JSP** per la generazione dinamica delle pagine lato server.
- **JavaScript** per validazioni lato client, interazioni dinamiche e miglioramento dell'usabilità.

La comunicazione tra browser e applicazione avviene tramite protocollo **HTTP**, attraverso richieste e risposte gestite dal server.

### *Buisness Tier (Back-end)*

La logica applicativa è implementata in **Java**, tramite un'architettura basata su:

- **Servlet** per la gestione delle richieste HTTP, il controllo del flusso applicativo e l'instradamento verso le JSP.
- Componenti di supporto (es. DAO/Bean) per l'accesso ai dati e l'organizzazione della logica di business.

Questo livello ha il compito di elaborare le operazioni principali della piattaforma (catalogo, carrello, ordine, recensioni, gestione utente), orchestrando correttamente la comunicazione tra interfaccia e database.



## Data Tier (Database)

La persistenza dei dati è affidata a un database relazionale basato su:

- **SQL** per interrogazioni, aggiornamenti e gestione delle entità (utenti, prodotti, ordini, recensioni, ecc.).

Il back-end accede ai dati eseguendo operazioni CRUD e query specifiche, garantendo coerenza e integrità delle informazioni memorizzate.

## 4 Features da testare/da non testare

---

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

- Gestione Identità e Accessi:

- o Registrazione Cliente
- o Login
- o Modifica dati account

- Gestione Catalogo Prodotti

- o Inserimento di un prodotto
- o Modifica di un prodotto

- Gestione Acquisti e

informazioni Cliente:

- o Acquisto del Carrello
- o Inserimento metodo di pagamento
- o Modifica di metodo di pagamento
- o Inserimento indirizzo



- Amministrazione:
  - Inserimento Articolo
  - Modifica Articolo
  - Visualizza lista utenti
  
- Gestione Recensioni:
  - Inserimento Recensione

Le funzionalità di cui non si andrà ad effettuare le attività di testing riguardano requisiti funzionali di bassa o media priorità; sono inoltre escluse le funzionalità che non prevedono input manuale da parte dell'utente - ad esempio attività riguardanti esclusivamente visualizzazioni di dati.

## 5 Pass/Fail criteria

---

Le attività di testing sono mirate ad identificare la presenza di faults (errori) all'interno del sistema, per effettuarne 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 diverso dall'output atteso dall'oracolo.

Un test fallisce (fail) se, dato un input al sistema, l'output ottenuto è uguale all'output atteso dall'oracolo.

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



- Testare tutti i requisiti funzionali ad alta priorità;
- Effettuare test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti;
- Raggiungere un branch coverage non inferiore al 75% sul codice back-end (Servlet/DAO/logica applicativa)

## 6 Approccio

---

Il testing dell'intero sistema si compone di tre fasi: testing di sistema, testing di integrazione e testing di unità. Verranno progettati nell'ordine appena definito, ma verranno eseguiti in ordine inverso.

Prima della fase di implementazione del sistema, avverrà la progettazione dei casi di test di sistema, perfezionati in seguito nella loro fase di esecuzione; durante la fase implementativa avverrà la progettazione dei casi di test di unità.

Durante lo sviluppo saranno eseguite periodiche attività di revisione sul codice prodotto.

Poiché la progettazione è organizzata seguendo un modello simile al modello a V, il testing di sistema è stato pianificato in seguito alla stesura del documento Requirements Analysis Document, mentre la pianificazione del testing di integrazione avverrà dopo la stesura del System Design Document.

### Testing di Sistema

Per il testing di sistema sarà utilizzato il tool Selenium IDE, che permette di registrare le azioni che un utente può intraprendere sul browser, in modo da poter implementare ed eseguire i test case di sistema. Il server, per la fase di testing, verrà deployato in localhost.

#### Functional testing

Il functional testing ha il fine di validare i requisiti funzionali. Consiste nell'individuare i possibili faults generati dagli input degli utenti.

#### Performance Testing

A causa del basso budget a disposizione, non si assicura l'esecuzione del performance testing.

#### Pilot Testing





A causa del basso budget a disposizione, non si assicura l'esecuzione del pilot testing.

## Acceptance Testing

L'acceptance testing verrà effettuato solo sul functional testing, ed il Project Manager simulerà la figura del cliente.

## Installation Testing

A causa del basso budget a disposizione, non si assicura l'esecuzione dell'installation testing.

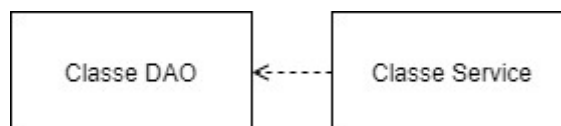
## Testing di integrazione

Verrà utilizzato un approccio bottom-up, metodo ritenuto più adatto per un software basato sul paradigma Object Oriented. La definizione dei test case avverrà tramite il framework *JUnit*, mentre verrà usato *Mockito* per il mocking. Verrà valutato l'utilizzo di Github Actions per realizzare la Continuous Integration, in alternativa a Travis CI. L'automatizzazione del run dei test sarà gestita da *Maven*, ed infine come tool di misurazione e report coverage sarà utilizzato *JaCoCo*.

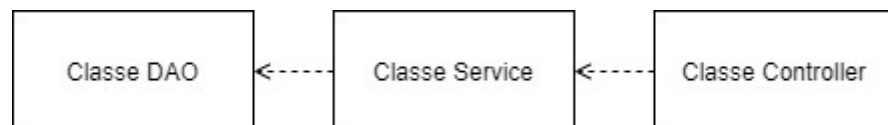
Il test di integrazione sarà il medesimo per tutte le componenti da testare. Nello specifico, si procederà prima con il test delle classi Service, e successivamente con il test delle classi Controller. Durante questa seconda esecuzione, la chiamata al controller sarà mockata usando Mockito.

Di seguito viene presentato un esempio grafico di test di integrazione diviso nei due steps.

### Primo step

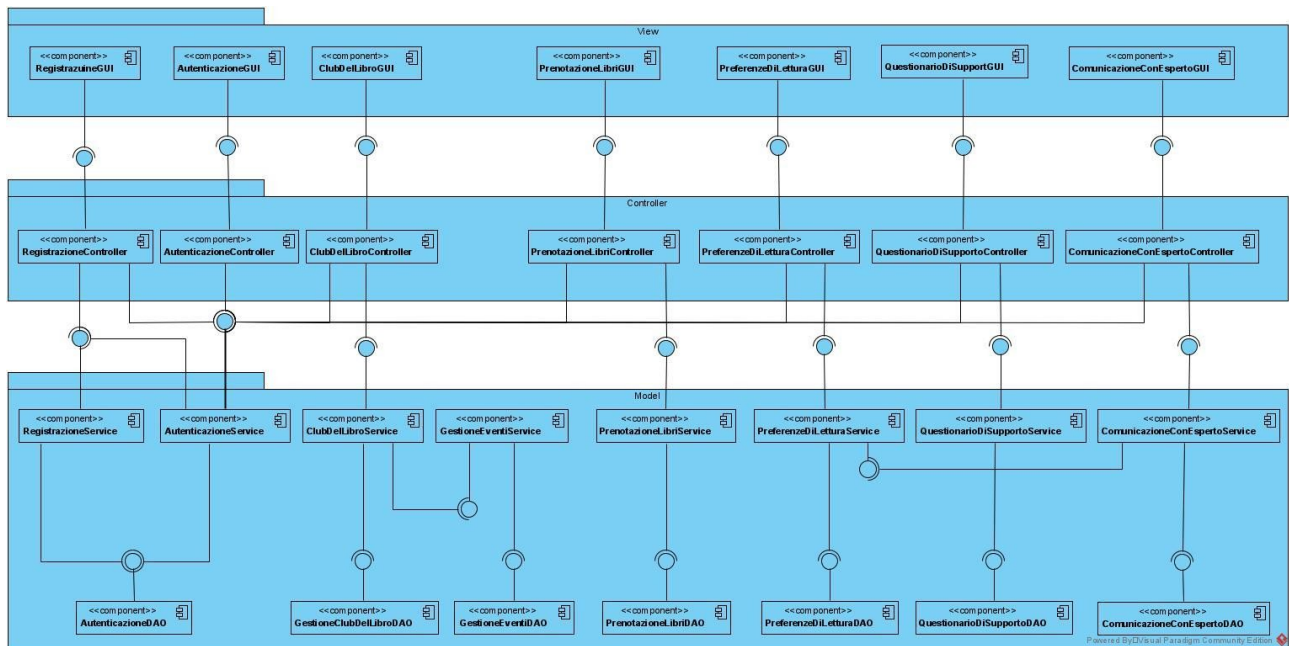


### Secondo step



In generale, il test di integrazione sarà compreso nella stessa classe di test di quelli unitari. Solo per il sottosistema club del libro, si è scelto un approccio differente, separandoli.

Per ciò che concerne le dipendenze tra i sottosistemi, si riporta di seguito il diagramma architetturale.



Il test di integrazione verrà fatto a partire dal layer model fino a salire al layer controller.

## Testing di Unità

Per il testing di unità la strategia prevista consiste nel testare ogni metodo delle classi del sistema. Da esse, sono escluse le interfacce e le classi entity, poiché quest'ultime presentano solo metodi getters e setters. I casi di test saranno definiti attraverso un approccio black-box e saranno documentati direttamente nel codice, attraverso l'uso del framework per il testing di classi Java *JUnit*.

Per ogni Production Class sarà definita una Test Class che rispetterà il formato *NomeProductionClassTest*. Tali classi saranno scritte in parallelo alle Production class, per garantire una più facile copertura del codice. Le stesse classi saranno poi revisionate e modificate da sviluppatori differenti.

Altre tecnologie usate in tale fase saranno:

- *Mockito*: per la costruzione degli stub e l'isolamento della componente testata.
- *JaCoCo*: per il calcolo di metriche tra le quali la Branch Coverage.
- *Maven*: per la build e l'esecuzione automatica dei tests.



## Ispezione del codice

Per aumentare la qualità del codice ci si affiderà principalmente ai controlli del tool Checkstyle e della CI/CD con Maven. Sebbene vi sia l'intenzione di fare ispezione del codice, anche se non costante e approfondita, a causa del basso budget non si può assicurare tale pratica.



## 7 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

Il testing non verrà sospeso fino alla sua terminazione, anche in caso di rilevazione di una failure. Il testing potrà essere momentaneamente sospeso nel caso venga restituito, al momento dell'esecuzione, un errore nella definizione di uno dei test stessi.

### Criteri di ripristino

Il testing verrà ripreso dopo aver risolto i fault individuati.

## 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, per via dell'assenza di specifiche formali/semi-formali.

### 9.1 Gestione Utente

#### 9.1.1 Registrazione Utente

**Parametro: Nome**



FORMATO: $^{\wedge}[A-z\grave{A}-\grave{u}}\prime - ]\{2,30\}\$$	
Nome Categoria	Scelte per la categoria
Formato [FNO]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FNO_OK]
Parametro: Cognome	
FORMATO: $^{\wedge}[A-z \grave{A}-\grave{u}}\prime - ]\{2,30\}\$$	
Nome Categoria	Scelte per la categoria
Formato [FCO]	1. Rispetta il formato = false [error]
	2. Rispetta il formato = true [PROPERTY FCO_OK]

Parametro: Username	
Nome Categoria	Scelte per la categoria
Lunghezza [LN]	1. Lunghezza >30 = false[error] 2. Lunghezza <=30 = true [PROPERTY LN_OK]
Parametro: Email	



FORMATO:	
^[A-z0-9._%+-]+@[A-z0-9.-]+\.[A-z]{2,10}\$	
Nome Categoria	Scelte per la categoria
Formato [FE]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE_OK]
Parametro: Password	
Nome Categoria	Scelte per la categoria
Lunghezza [LP]	1. Lunghezza < = false [error] 2. Lunghezza >= 8= true [PROPERTY LP_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
TC_1.1_1	LN1	Errato: cognome non inserito
TC_1.1_2	LN2	Errato: nome non inserito



TC_1.1_3	LN2	Errato: sesso non inserito
TC_1.1_4	LN2	Errato: email non inserito
TC_1.1_5	LN2	Errato: username non inserito
TC_1.1_6	LN2 , LN6	Errato: Password non inserito
TC_1.1_7	LN2	Errato: Conferma Password non inserito
TC_1.1_8	LN8, LN6	Errato: Password e Conferma non coincidono
TC_1.1_9	LN9, LN2	Errato: nome non corretto
TC_1.1_10	LN9, LN1	Errato: cognome non corretto
TC_1.1_11	LN11, LN14	Errato: username già presente
TC_1.1_12	LN12	Errato: e-mail già presente
TC_1.1_13	LN11	Errato: e-mail non corretta



TC_1.1_14	LN11	Errato: password non corretta
TC_1.1_15	LN15	Corretto

### 9.1.2 Login

Parametro: Nome utente	
Nome Categoria	Scelte per la categoria
Lunghezza [LN]	<ol style="list-style-type: none"><li>1. Lunghezza &gt;30 = false[error]</li><li>2. Lunghezza &lt;=30 = true [PROPERTY LN_OK]</li></ol>
Parametro: Password	
Nome Categoria	Scelte per la categoria
Lunghezza [LP]	<ol style="list-style-type: none"><li>1. Lunghezza &lt; 8 = false [error]</li><li>2. Lunghezza &gt;= 8 = true [PROPERTY LP_OK]</li></ol>

Test Case ID	Test frame	Esito
TC_1.2_1	LN2 ,LU1	Errato: nome utente non inserito
TC_1.2_2	LN2, LU1	Errato: password non inserita
TC_1.2_3	LN9,LU2	Errato: password non corretta





TC_1.2_4	LU3, LN9	Errato: username non corretta
TC_1.2_5	LN2, LU2	Corretto

### 9.1.3 Login Amministratore

Parametro: Nome utente	
Nome Categoria	Scelte per la categoria
Lunghezza [LN]	3. Lunghezza >30 = false[error] 4. Lunghezza <=30 = true [PROPERTY LN_OK]
Parametro: Password	
Nome Categoria	Scelte per la categoria
Lunghezza [LP]	3. Lunghezza < 8 = false [error] 4. Lunghezza >= 8 = true [PROPERTY LP_OK]

Test Case ID	Test frame	Esito
TC_1.3_1	LN2, LU1	Errato: username non inserito
TC_1.3_2	LN2, LU1	Errato: password non inserita



TC_1.3_3	LN9, LU2	Errato: password non corretta
TC_1.3_4	LA1	Errato: username non autorizzato
TC_1.3_5	LA1, LU2, LN2	Corretto

#### 9.1.4 Nuovo metodo di pagamento

Parametro: Numero carta	
FORMATO: /^\d{16}\$/	
Nome Categoria	Scelte per la categoria
Formato [FNO]	<ol style="list-style-type: none"><li>1. Rispetta il formato = false [error]</li><li>2. Rispetta il formato = true [PROPERTY FNO_OK]</li></ol>

Parametro: Intestatario	
FORMATO: /^\d{16}\$/	
Nome Categoria	Scelte per la categoria
Lunghezza [LP]	<ol style="list-style-type: none"><li>5. Lunghezza &lt; 30 = false [error]</li><li>6. Lunghezza &gt;= 30 = true [PROPERTY LP_OK]</li></ol>



Parametro: Scadenza	
<b>FORMATO:</b> <code>/^[0[1-9]   1[0-2]]\d{2}\$</code>	
Nome Categoria	Scelte per la categoria
Formato [FNO]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FNO_OK]

Parametro: CVV		
FORMATO: /^\d{3}\$/		
Nome Categoria	Scelte per la categoria	
Formato [FNO]	<div>1. Rispetta il formato = false [error]</div> <div>2. Rispetta il formato = true [PROPERTY FNO_OK]</div>	
Test Case ID	Test frame	Esito
TC_1.4_1	MP1	Errato: Numero di Carta non inserito
TC_1.4_2	MP1	Errato: Scadenza non inserita



TC_1.4_3	MP1	Errato: Intestatario non inserito
TC_1.4_4	MP1	Errato: CVV non inserito
TC_1.4_5	MP1	Errato: Numero Carta non corretto
TC_1.4_6	MP1	Errato: Scadenza già scaduta
TC_1.4_7	MP2	Errato : Scadenza non corretta
TC_1.4_8	MP2	Errato: CVV non corretto
TC_1.4_9	MP3	Errato: carta già registrata
TC_1.4_10	MP4	Corretto

### 9.1.5 Aggiungi indirizzo

Parametro: Via	
FORMATO: $^{\wedge}[\emptyset-9A-z\grave{A}-\grave{u}}\prime-\{2,30\}\$$	
Nome Categoria	Scelte per la categoria
Formato [FVN]	<ol style="list-style-type: none"><li>1. Rispetta il formato = false [error]</li><li>2. Rispetta il formato = true [PROPERTY FVN_OK]</li></ol>

Parametro: Città	
Nome Categoria	Scelte per la categoria



Lunghezza [LN]	<ol style="list-style-type: none"> <li>1. Lunghezza &gt;30 = false [error]</li> <li>2. Lunghezza &lt;=30 = true [PROPERTY LN_OK]</li> </ol>
----------------	---

Parametro: CAP	
FORMATO: ^\d{5}\$	
Nome Categoria	Scelte per la categoria
Formato [FVN]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FVN_OK]</li> </ol>

Parametro: Provincia	
Nome Categoria	Scelte per la categoria
Lunghezza [LN]	<ol style="list-style-type: none"> <li>3. Lunghezza &gt;2 = false [error]</li> <li>4. Lunghezza &lt;=2 = true [PROPERTY LN_OK]</li> </ol>
Parametro: Nazione	
FORMATO: ^[A-zÀ-ù ' -]{2,30}\$	
Nome Categoria	Scelte per la categoria
Match [MCP]	<ol style="list-style-type: none"> <li>1. Match con Nazione = false [error]</li> <li>2. Match con Nazione = true [PROPERTY MCP_OK]</li> </ol>



Parametro: indirizzo preferito	
FORMATO: ^/^(true   false   1   0)\$/	
Nome Categoria	Scelte per la categoria
Formato [FNO]	3. Rispetta il formato = false [error] 4. Rispetta il formato = true [PROPERTY FNO_OK]

Test Case ID	Test frame	Esito
TC_1.5_1	ME1	Errato: Via non inserita
TC_1.5_2	ME1	Errato: Nazione non inserita
TC_1.5_3	ME1	Errato: Città non inserita
TC_1.5_4	ME1	Errato: CAP non inserito
TC_1.5_5	ME1	Errato: Provincia non inserito



TC_1.5_6	ME1	Errato: Nazione non inserita
TC_1.5_7	ME2	Errato: Inserimento di numeri in Città
TC_1.5_8	ME3	Errore: CAP non corretto
TC_1.5_9	ME3	Errato: Provincia non corretta
TC_1.5_10	ME3	Errato: Via non corretta
TC_1.5_11	ME3	Errato: Città non corretta
TC_1.5_12	ME3	Errato: Nazione non corretta
TC_1.5_13	ME4	Corretto

## 9.2 Modifica credenziali

Parametro: Modifica Username	
Nome Categoria	Scelte per la categoria
Lunghezza [LN]	1. Lunghezza > 30 = false [errore] 2. Lunghezza <= 30 = true [PROPERTY LN_OK]



Parametro: Password attuale		
Nome Categoria		Scelte per la categoria
Match [MCP]		1. Match con password attuale = false [error] 2. Match con password attuale = true [PROPERTY MCP_OK]
Parametro: Nuova Password		
Nome Categoria		Scelte per la categoria
Lunghezza [LN]		3. Lunghezza > 8 = false [errore] 4. Lunghezza <= 8 = true [PROPERTY LN_OK]
Parametro:Confermata Nuova Password		
Match [MCP]		1. Match con password = false [error] 2. Match con password = true [PROPERTY MCP_OK]
Test Case ID	Test frame	Esito
TC_1.6_1	LN1	Errato: Password non inserita
TC_1.6_2	LN2, LD1	Errato: username non inserito
TC_2.1_3	LN2, LD2, GFC1	Errato: username non corretta





TC_1.6_4	LN2, LN6	Errato: Password non corretta
TC_1.6_5	LN6	Errato: Conferma Password non corretta
TC_1.6.6	MC1, LN2	Errato: nuova password non corretta
TC_1.6_7	MC2	Password Modificata
TC_1.6_8	MC3	Username Modificato

## 9.2 Gestione Catalogo

### 9.2.1 Ricerca Catalogo

Parametro: Barra di ricerca	
Nome Categoria	Scelte per la categoria
Lunghezza [LN]	1. Lunghezza > 30 = false [errore] 2. Lunghezza <= 30 = true [PROPERTY LN_OK]

Test Case ID	Test frame	Esito
TC_2.1_1	RP1	Corretto
TC_2.1_2	RP2	Errato: Prodotto non esistente
TC_2.1_4	RP3	Errato: Barra di ricerca non corretto



TC_2.1_5	RP4	Corretto
----------	-----	----------

## 9.2.2 Gestione Recensioni

Parametro: Valutazione	
Nome Categoria	Scelte per la categoria
<b>FORMATO:</b> $\wedge(?:[1-5])\backslash s+stelle\$$	
Range Stelle [RS]	<ol style="list-style-type: none"> <li>1. Lunghezza &gt; 30 OR Lunghezza &lt; 1 = false [errore]</li> <li>2. Lunghezza &lt;= 30 AND Lunghezza &gt;= 1 = true [PROPERTY LN_OK]</li> </ol>
Parametro: Titolo	
Nome Categoria	Scelte per la categoria
Lunghezza [LD]	<ol style="list-style-type: none"> <li>1. Lunghezza &gt; 30 = false [errore]</li> <li>2. Lunghezza &lt;= 30 = true [PROPERTY LD_OK]</li> </ol>
Parametro: Descrizione	
Nome Categoria	Scelte per la categoria
Lunghezza [LD]	<ol style="list-style-type: none"> <li>3. Lunghezza &gt; 255 = false [errore]</li> <li>4. Lunghezza &lt;= 255 = true [PROPERTY LD_OK]</li> </ol>

Test Case ID	Test frame	Esito
TC_3.1_1	GR1	Errato: Valutazione non inserita



TC_3.1_2	GR1	Errato: Titolo non inserito
TC_3.1_3	GR1	Errato: Descrizione non inserita
TC_3.1_4	GR2	Errato: Titolo non corretto
TC_3.1_5	GR2	Errato: Descrizione non corretta
TC_3.1_6	GR3	Corretto

#### 9.4.1 Amministrazione

##### Parametro: Codice Articolo

###### Nome Categoria

###### Scelte per la categoria

Lunghezza [LN]

1. Lunghezza > 30 OR Lunghezza < 1 = false [errore]
2. Lunghezza <= 30 AND Lunghezza >= 1 = true [PROPERTY\_LN\_OK]

##### Parametro: Nome

###### Nome Categoria

###### Scelte per la categoria

Lunghezza [LN]

3. Lunghezza > 30 OR Lunghezza < 1 = false [errore]
4. Lunghezza <= 30 AND Lunghezza >= 1 = true [PROPERTY\_LN\_OK]

##### Parametro: Descrizione

###### Nome Categoria

###### Scelte per la categoria



Lunghezza [LD]	<ol style="list-style-type: none"><li>1. Lunghezza &gt; 255 = false [errore]</li><li>2. Lunghezza &lt;= 255 = true [PROPERTY_LD_OK]</li></ol>
----------------	---

#### Parametro: Sconto

##### Nome Categoria

##### Scelte per la categoria

**FORMATO:** ^(?:0|0\.\d{1,2}|1(?:\.\d{0,2})?)\$

Range [SCT]

Sconto nel range = false [error]  
Sconto nel range = true [PROPERTY SCT\_OK]

Decimali [PRD]

Max 2 decimali = false [error]  
Max 2 decimali = true [PROPERTY PRD\_OK]

#### Parametro: Prezzo

##### Nome Categoria

##### Scelte per la categoria

**FORMATO:** ^(?:0|[1-9]\d\*)(?:\.\d{1,2})?\$

Positivo [PRP]

Prezzo > 0 = false [error]  
Prezzo > 0 = true [PROPERTY PRP\_OK]

Decimali [PRD]

Max 2 decimali = false [error]  
Max 2 decimali = true [PROPERTY PRD\_OK]

#### Parametro: Peso

##### Nome Categoria

##### Scelte per la categoria

**FORMATO:** ^(?:0|[1-9]\d\*)(?:\.\d{1,3})?\$



Positivo [PRP]

Peso > 0 = false [error]

Peso > 0 = true [PROPERTY PRP\_OK]

Decimali [PRD]

Max 2 decimali = false [error]

Max 2 decimali = true [PROPERTY PRD\_OK]

#### Parametro: Dimensione

Nome Categoria

Scelte per la categoria

**FORMATO:**  $^{[1-9]}\backslash d\{0,3\}x[1-9]\backslash d\{0,3\}$

Formato [DIM]

Rispetta "NxM" = false [error]

Rispetta "NxM" = true [PROPERTY  
DIM\_OK]PRP\_OK]

#### Parametro: Dimensione

Nome Categoria

Scelte per la categoria

**FORMATO:**  $^{(mobili | sedie | divani | letti | illuminazione | tavolini | accessori | esterni | bambini)}$

Dominio [CAT]

Categoria non nel set = false [error] 2.

Categoria nel set = true [PROPERTY  
CAT\_OK]

Parametro: Colore		
Nome Categoria	Scelte per la categoria	
Lunghezza [LD]	1. Lunghezza > 30 = false [errore] 2. Lunghezza <= 30 = true [PROPERTY LD_OK]	
Test Case ID	Test frame	Esito
TC_4.1_1	IA1	Errato: Codice articolo non inserito



TC_4.1_2	IA1	Errato: Nome non inserito
TC_4.1_3	IA1	Errato: Descrizione non inserita
TC_4.1_4	IA1	Errato: Prezzo non inserito
TC_4.1_5	IA1	Errato: categoria non inserita
TC_4.1_6	IA2	Errato: codice non corretto
TC_4.1_7	IA2	Errato: Nome non corretto
TC_4.1_8	IA2	Errato: Prezzo non corretto
TC_4.1_9	IA2	Errato: sconto non corretto
TC_4.1_10	IA3	Errato: Articolo già esistente
TC_4.1_11	IA4	Corretto

## 9.2.4 Modifica articolo

### Parametro: Codice Articolo

#### Nome Categoria

#### Scelte per la categoria

Lunghezza [LN]

5. Lunghezza > 30 OR Lunghezza < 1 = false [errore]

6. Lunghezza <= 30 AND Lunghezza >= 1 = true [PROPERTY\_LN\_OK]

### Parametro: Nome



Nome Categoria	Scelte per la categoria
Lunghezza [LN]	<p>7. Lunghezza &gt; 30 OR Lunghezza &lt; 1 = false [errore]</p> <p>8. Lunghezza &lt;= 30 AND Lunghezza &gt;= 1 = true [PROPERTY_LN_OK]</p>
<b>Parametro: Descrizione</b>	
Nome Categoria	Scelte per la categoria
Lunghezza [LD]	<p>3. Lunghezza &gt; 255 = false [errore]</p> <p>4. Lunghezza &lt;= 255 = true [PROPERTY_LD_OK]</p>

#### Parametro: Sconto

Nome Categoria	Scelte per la categoria
----------------	-------------------------

**FORMATO:** ^(?:0|0\.\d{1,2}|1(?:\.\d{0,2})?)\$

Range [SCT]

Sconto nel range = false [error]  
Sconto nel range = true [PROPERTY SCT\_OK]

Decimali [PRD]

Max 2 decimali = false [error]  
Max 2 decimali = true [PROPERTY PRD\_OK]

#### Parametro: Prezzo

Nome Categoria	Scelte per la categoria
----------------	-------------------------

**FORMATO:** ^(?:0|[1-9]\d\*)(?:\.\d{1,2})?\$



Positivo [PRP]

Prezzo > 0 = false [error]  
Prezzo > 0 = true [PROPERTY PRP\_OK]

Decimali [PRD]

Max 2 decimali = false [error]  
Max 2 decimali = true [PROPERTY PRD\_OK]

#### Parametro: Peso

Nome Categoria

Scelte per la categoria

**FORMATO:**  $^{(?:(0|[1-9]\d*)?(?:\.\d{1,3}))?}$$

Positivo [PRP]

Peso > 0 = false [error]  
Peso > 0 = true [PROPERTY PRP\_OK]

Decimali [PRD]

Max 2 decimali = false [error]  
Max 2 decimali = true [PROPERTY PRD\_OK]

#### Parametro: Dimensione

Nome Categoria

Scelte per la categoria

**FORMATO:**  $^{[1-9]\d{0,3}x[1-9]\d{0,3}}$$

Formato [DIM]

Rispetta "NxM" = false [error]  
Rispetta "NxM" = true [PROPERTY  
DIM\_OK]PRP\_OK]

#### Parametro: Dimensione

Nome Categoria

Scelte per la categoria

**FORMATO:**  $^{(mobili|sedie|divani|letti|illuminazione|tavolini|accessori|esterni|bambini)}$$

Dominio [CAT]

Categoria non nel set = false [error] 2.  
Categoria nel set = true [PROPERTY  
CAT\_OK]





Parametro: Colore	
Nome Categoria	Scelte per la categoria
Lunghezza [LD]	<ul style="list-style-type: none"><li>3. Lunghezza &gt; 30 = false [errore]</li><li>4. Lunghezza &lt;= 30 = true [PROPERTY LD_OK]</li></ul>

Test Case ID	Test frame	Esito
TC_4.2_1	IA1, MA1	Errato: Nome non inserito
TC_4.2_2	IA1, MA1	Errato: Descrizione non inserita
TC_4.2_3	IA1, MA1	Errato: Categoria non inserita
TC_4.2_4	IA1, MA1	Errato: prezzo non inserito
TC_4.2_5	IA2, MA2	Errato: nome non corretto
TC_4.2_6	IA2, MA2	Errato: prezzo non corretto
TC_4.2_7	IA2, MA2	Errato: sconto non corretto
TC_4.2_8	IA2, MA2	Errato: peso non corretto
TC_4.2_9	MA3	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.

Per altre informazioni si rimanda ai documenti di management sullo schedule.