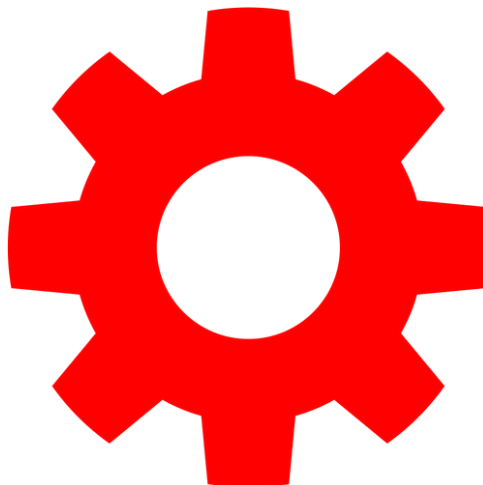


**SHOP-AUTORICAMBI**  
**SYSTEM TEST PLAN E SPECIFICA DEI CASI DI TEST**  
**Versione 2.0**



## **INDICE**

### **1. Introduzione**

### **2. Riferimenti e relazioni**

#### **2.1. Relazioni con il RAD**

#### **2.2. Relazioni con l'SDD**

#### **2.3. Relazioni con l'ODD**

### **3. Panoramica del sistema**

### **4. Funzionalità testate e non testate**

#### **4.1. View**

#### **4.2. Controller**

#### **4.3. Model**

### **5. Pass/Fail**

#### **5.1. Pass criteria**

#### **5.2. Failcriteria**

### **6. Approcci**

#### **6.1. Test delle unità**

#### **6.2. Test di integrazione**

### **7. Sospensione e Ripristino**

#### **7.1. Criteri di sospensione**

#### **7.2. Criteri di ripristino**

### **8. Strumenti per il testing**

#### **8.1 Strumenti hardware**

#### **8.2 Strumenti software**

### **9. Test Case**

#### **9.1. Registrazione**

#### **9.2. Login**

**9.3. Ricerca prodotto**

**9.4. Caricamento prodotto (gestore)**

**9.5. quantità (prodotto)**

**10. Testing schedule**

**10.1. Gestione dei rischi**

**10.2. Organizzazione delle attività**

## **1. INTRODUZIONE**

Il testing consiste nel trovare le differenze tra il comportamento atteso specificato attraverso il modello di sistema e il comportamento che si ha dopo l'implementazione del sistema.

Lo scopo principale di questa attività è quello di testare il sistema e di rilevare gli errori e i problemi che si vengono a verificare in fase di implementazione. Questo vuol dire che nella fase di testing andiamo a cercare di creare il massimo numero di errori in modo tale che questi possano essere corretti in questa suddetta fase. Lo scopo del documento è quello di andare a definire i test case su cui verranno testate le funzionalità del sistema. Per ogni funzionalità saranno forniti un numero sufficiente di istanze di input in modo tale da fornire almeno un test case composto da dati corretti e quindi appartenenti a classi valide per ogni campo di input che rispettano le condizioni definite nel documento di test plan. Ci sarà inoltre un test case per ogni input che prevede una classe non valida e almeno un test case che non soddisfa le condizioni definite nel test plan.

## **2. RIFERIMENTI E RELAZIONI**

In questa parte vengono descritte le relazioni con gli altri documenti prodotti durante la fase di sviluppo del progetto software SHOP-AUTORICAMBI. Tali relazioni sono necessarie al fine di produrre una fase di testing che sia coerente con tutte le specifiche dedotte durante l'analisi ed il raffinamento dei requisiti e delle funzionalità che il sistema dovrà fornire. Le relazioni che andiamo a prendere in considerazione sono con:

- RAD – documento di analisi dei requisiti
- SDD – documento di system design
- ODD – documento di object design

### **2.1. RELAZIONE CON IL DOCUMENTO DI ANALISI DEI REQUISITI (RAD)**

Tali relazioni riguardano i requisiti funzionali e non funzionali del sistema. I test delle funzionalità terranno conto delle specifiche espresse nel RAD. I test devono tener presente dei diversi attori descritti nel documento di analisi (RAD):

- Cliente
- Gestore
- Prodotti

Per ciascun cliente e gestore dovrà essere testata la validità dei campi associati.

Per i prodotti dovranno essere testati inserimento, cancellazione, ricerca, quantità.

Inoltre deve essere considerato il flusso di controllo descritto nei diagrammi delle sequenze di operazioni, in modo da testare determinate funzionalità all'interno e all'esterno del proprio flusso standard.

## **2.2. RELAZIONI CON IL DOCUMENTO DI SYSTEM DESIGN (SDD)**

La pianificazione dei test delle componenti rispecchia la decomposizione in sottosistemi specificata nel documento del System Design.

## **2.3. RELAZIONI CON IL DOCUMENTO DI OBJECT DESIGN (ODD)**

La fase di testing deve considerare il documento di Object Design in quanto esso fornisce la base per realizzare l'implementazione. Pertanto è necessario effettuare il testing delle unità per individuare le differenze tra il comportamento atteso e quello specificato.

## **3. PANORAMICA DEL SISTEMA**

Il sito di commercio elettronico che si vuole realizzare è concepito sull'idea di dare la possibilità, a dei clienti, di acquistare pezzi di ricambio per automobili. Il sito, inoltre, intende pubblicizzare le componenti di ricambio da esso vendute permettendo, inoltre, ai clienti la possibilità di affidarsi a delle officine presso le quali effettuare il montaggio, in presenza di personale qualificato, delle componenti acquistate

Il sistema SHOP-AUTORICAMBI segue il modello MVC per la suddivisione dei livelli:

- Model: che rappresenta i dati dell'applicazione;
- View: che rappresenta la visualizzazione degli oggetti;
- Controller: che rappresenta l'insieme di regole che permettono la trasformazione degli input sulle viste in modifiche del modello.

Ogni livello, come descritto nel documento SDD, è stato suddiviso in sottosistemi in modo da raggruppare le componenti.

Ciascun sottosistema, appartenente a ciascun livello, deve essere testato.

Il presente test plan relativo al sistema SHOP-AUTORICAMBI si pone i seguenti obiettivi:

- Dettagliare le attività richieste per preparare e condurre il testing;
- Definire le fonti usate per preparare la pianificazione;
- Affinchè il testing riesca a raggiungere al meglio i suoi obiettivi, si è ritenuto necessario non condurre tale processo in modo casuale, cioè guidato solo dalla fantasia e dalla esperienza dei test-engineer, ma definendo i punti generali al quale far riferimento:

- **Individuare quando deve partire il processo di testing e quale è il criterio di terminazione da utilizzare**

Identificare a priori lo stato di inizio e di fine dell'interno di un processo è fondamentale per evitare che il processo cominci quando ancora non sono presenti tutte le informazioni necessarie o che si prolunghi eccessivamente.

- **Definire per ogni test i risultati attesi**

La descrizione dei risultati ottenuti dall'esecuzione di un test case è determinante poiché, senza una precisa connotazione degli output attesi, si corre il rischio di interpretare come correnti dei risultati che viceversa non lo sono.

- **Analizzare a fondo i risultati di ciascun test**

- **Non pianificare il testing assumendo tacitamente che nessun errore sarà individuato**

Questo è un comportamento che deriva direttamente da un'errata interpretazione del documento di testing, laddove si crede che il testing sia la verifica delle funzionalità del sistema, mentre il vero processo di testing è la ricerca del maggior numero di errori.

- **Il miglior test case è quello che ha la più alta probabilità di scoprire errori non ancora rilevati.**

Questo principio è poco pragmatico, in quanto non è possibile a priori valutare la probabilità di scoprire errore da parte di un test case, ma possiamo affermare che si deve sempre tendere ad una produzione limitata il più possibile di test case di qualità, al fine di massimizzare il rapporto tra gli errori scoperti ed i test utilizzati.

Ovviamente non sempre è possibile riuscire a rispettare tutti i principi guida elencati, per cui si cerca, all'interno delle metodologie, di seguire il più possibile le linee guida che essi tracciano.

#### 4. FUNZIONALITA' TESTATE E NON

Le componenti prese in considerazione nella fase di testing rappresentano la maggior parte delle funzionalità di SHOP-AUTORICAMBI. In particolare saranno testate:

- Gestione autenticazione (cliente e gestore): sarà testata la funzionalità di log-in dei vari attori del sistema.
- Gestione prodotti: saranno testate le funzionalità di inserimento, "\*\*\*\*\*"modifica\*\*\*\*\* e ricerca di un prodotto.
- Gestione gestore: caricamento dei vari prodotti indicando per ognuno di essi il codice, il nome, la descrizione, il prezzo e una eventuale quantità.

Tutte queste funzionalità richiedono meccanismi dinamici che in base a dati persistenti permettono le diverse operazioni attraverso elaborazioni server.

#### **4.1. FUNZIONALITA' DA TESTARE**

Le funzionalità da testare riguardano i tre livelli dell'architettura proposta, come di seguito descritto.

##### **4.1.1 VIEW**

Le funzionalità che devono essere testate per il livello di visualizzazione sono:

- areaCli.jsp
- areaAmmi.jsp
- carrello.jsp
- cerca\_prodotti.jsp
- contatti.jsp
- footer.jsp
- login.jsp
- menu.jsp
- prodotti.jsp
- registrazione.jsp
- shop\_autoricambi.jsp

##### **4.1.2 CONTROLLER**

Tutte le funzionalità del livello di visualizzazione che devono essere testate implicano il test parallelo delle componenti dipendenti nel livello di controllo, che forniscono l'elaborazione necessaria affinché il livello di visualizzazione possa fornire all'utente, che ha richiesto le operazioni, l'insieme dei risultati associati.

Di conseguenza, le funzionalità che devono essere testate sono:

- Acquista\_Control.java
- Carrello\_Control.java
- Login\_Control.java
- Logout\_Control.java
- Registrazione\_Control.java
- AmministratoreProdotti\_Control.java

- Prodotti\_Control.java

#### **4.1.3 MODEL**

Tutte le funzionalità del livello di controllo che devono essere testate implicano il test parallelo delle componenti dipendenti nel livello dati, che forniscono l'interazione con l'insieme dei dati persistenti presenti all'interno del sistema SHOP-AUTORICAMBI.

### **5. PASS/FAIL CRITERIA**

La fase di testing necessita di criteri formali per la determinazione del successo o dell'insuccesso di un determinato test. I dati di ogni input relativi ai test verranno divisi in classi di equivalenza: un input appartenente ad una classe specifica supera il test se l'output ottenuto corrisponde ai risultati attesi, specificati nell'oracolo.

#### **5.1 PASS CRITERIA**

I pass criteria determinano l'insuccesso del test e quindi la correttezza del comportamento della componente testata. I pass criteria per la fase di testing del sistema SHOP-AUTORICAMBI sono essenzialmente raggruppati in due categorie principali, che possono essere raffinate qualora il test necessiti di raffinamento. Le categorie individuate sono le seguenti:

- Comportamento atteso:
  - o L'oracolo rileva che il comportamento della componente rispetti il comportamento atteso
- Nessun errore rilevato dalla componente:
  - o La componente non genera alcuna eccezione non prevista.

#### **5.2 FAIL CRITERIA**

I failcriteria determinano il successo del test e quindi la presenza di errori nel comportamento della componente testata. I failcriteria per la fase di testing del sistema SHOP-AUTORICAMBI sono essenzialmente raggruppati in 2 categorie principali che possono essere raffinate nei casi specifici. Le categorie individuate sono le seguenti:

- Errore rilevato dalla componente:
  - o La componente genera una eccezione non prevista.
- Comportamento non atteso:



- L'oracolo rileva che il comportamento della componente si discosti dal comportamento atteso rispetto agli input.

## **6. APPROCCI**

Nella sessione di testing di SHOP-AUTORICAMBI sarà utilizzato un approccio di tipo "BLACK BOX", il quale prevede che i test siano effettuati ad un livello di astrazione più alto, in modo da non entrare nei dettagli del codice, ma basandosi sulle specifiche delle funzionalità da testare.

L'approccio alla fase di testing si compone di tre fasi:

- Testing di unità, che controlla i singoli componenti (classi, metodi)
- Testing di integrazione, che testa l'integrazione dei vari sottosistemi
- Testing funzionale, che verifica la funzionalità dell'intero sistema assemblato.

### **6.1 TESTING DELLE UNITA'**

Con il testing di unità sarà effettuato un controllo delle varie classi e metodi del sistema, quindi saranno ricercate le condizioni di fallimento evidenziando gli errori. Il testing di unità sarà eseguito utilizzando il framework JUnit. In particolare, per ogni classe che esegue operazioni complesse, sarà sviluppata la relativa classe JUnit.

### **6.2 TESTING DI INTEGRAZIONE**

Con il testing di integrazione si effettuerà un controllo sull'integrazione delle varie componenti del sistema. Si adotterà una strategia di tipo "Bottom-up".

Per effettuare questi test di integrazione, spesso sarà necessario l'utilizzo di driver, dato che tale strategia va ad integrare man mano i sottosistemi partendo dal layer che si trova più in basso nella scala gerarchica. Dall'applicazione del metodo del Category Partition si ricaverà la lista dei test cases.

## **7. SOSPENSIONE E RIPRISTINO**

La fase di testing del sistema SHOP-AUTORICAMBI può essere interrotta e ripresa più volte, se il fine ultimo è di rendere il sistema corretto e ogni funzionalità completa. Di seguito sono riportati i

criteri secondo cui è necessario sospendere la fase di testing e le modalità secondo cui poi deve essere ripresa.

### **7.1. CRITERI DI SOSPENSIONE**

La sospensione della fase di testing deve avvenire qualora un test abbia esito positivo ovvero si è riscontrato un errore all'interno di una componente.

Questi criteri comprendono tutti gli errori che hanno un impatto dannoso sul progresso dell'attività di testing, come il crash del database, il crash del sistema operativo, il fallimento di intere funzionalità ed anche problemi relativi all'ambiente di sviluppo del testing.

### **7.2. CRITERI DI RIPRISTINO**

La ripresa del test avviene soltanto quando tutti i problemi relativi alla sospensione del test sono stati risolti. L'attività di testing riprenderà a partire dal test case che ha causato la sospensione.

## **8. STRUMENTI PER IL TESTING**

Gli strumenti per il testing verranno divisi in due tipi differenti:

- Strumenti hardware
- Strumenti software

### **8.1. STRUMENTI HARDWARE**

Il sistema sarà testato su un personal computer con la necessaria dotazione software. Non sono richieste particolari specifiche hardware se non quelle adatte a far funzionare il DBMS server e il Web server apache.

### **8.2. STRUMENTI SOFTWARE**

Per i test delle unità e di integrazione si cercherà di adoperare JUnit. JUnit è un framework interamente sviluppato in Java Open-Source. Fornisce un insieme di API che assistono gli sviluppatori nel crearti in modo semplice ed automatico dei test per il proprio software. Permette di separare i test dal codice. Fornisce una struttura di asserzioni per confrontare i risultati attesi (oracolo) con quelli ottenuti. Fornisce una interfaccia grafica molto elementare per la valutazione dei risultati dei test: barra verde se i test case in esecuzione falliscono, barra rossa se qualche test case della test suite fallisce.

Consente di definire:

- Test case: singolo test su di una specifica funzionalità (un metodo, una classe, etc.).
- Test suite: una collezione di test case che possono essere raggruppati in base a caratteristiche omogenee e possono essere eseguite in blocco.

In particolare i software che si utilizzeranno per l'attività di testing sono i seguenti:

- JUnit
- MySQL
- Apache Tomcat 7 o superiore
- NetBean IDE 8.1
- Java Runtime Environment (JRE) 7 o superiore
- Java Development kit (JDK) 7 o superiore
- Java Enterprise Edition (J2EE)
- Selenium IDE per l'elecuazione dei test case;
- Browser (Google Chrome, FirefoxMozilla, Safari, Opera, Internet Explorer, etc.).

## 9. TEST CASE

I test case sono diretti a scoprire a eventuali malfunzionamenti o comportamenti errati da parte del sistema. Per fare ciò è necessario testare il sistema su diverse istanze di input, ognuna diretta a testare comportamenti del sistema in determinate condizioni, un tipo o classe di input piuttosto di un'altra. Il testing viene quindi strutturato in base alle funzionalità fornite dal sistema. Per ognuna di queste verranno fornite delle istanze di input che costituiscono i test case dei vari scenari. I test case potranno appartenere a tre diverse categorie di input in base al tipo di dati da cui sono costituiti relativamente allo use case in esame che possono essere:

- Grammaticalmente non validi: fanno parte di questa categoria gli input che contengono caratteri non validi, che sono a loro volta suddivisi in classi di equivalenza.
- Logicamente non validi: fanno parte di questa categoria i dati che potrebbero essere grammaticalmente validi ma che nel contesto non lo sono, ad esempio in un form che richiede l'inserimento di due numeri per la definizione di un range di valori, due input logicamente non validi possono essere un lowerbound (il valore inferiore del range) più alto dell'upperbaund (valori superiori del range).
- Validi: fanno parte di questa categoria i dati validi, cioè che sono grammaticalmente validi e hanno senso nel contesto in cui sono utilizzati.

Quindi per ogni funzionalità verranno presentati un insieme di test. Ognuno di questi insiemi è diviso a sua volta in tre sottoinsiemi:

- Input grammaticalmente non validi, ma logicamente validi
- Input logicamente non validi, ma grammaticalmente validi
- Input grammaticalmente e logicamente validi.

## 9.1. REGISTRAZIONE

Le funzionalità da testare riguardano una componente del sottosistema ACCESSO, in particolare la funzionalità di registrazione.

### 9.1.1. CLASSI DI EQUIVALENZA

INPUT	Nome, cognome, via, città	
CLASSI VALIDE	C_0001	Stringa alfanumerica con lunghezza compresa tra 1 e 40 caratteri.
CLASSI NON VALIDE	C_0002	Stringa maggiore di 40 caratteri.
	C_0003	Stringa minore di 1 carattere.

INPUT	Codice fiscale	
CLASSI VALIDE	C_0004	Stringa alfanumerica con lunghezza fissa di 16 caratteri.
CLASSI NON VALIDE	C_0005	Stringa alfanumerica minore di 16 caratteri.
	C_0006	Stringa alfanumerica maggiore di 16 caratteri.
	C_0007	Stringa alfanumerica discorde con la forma standard di un codice fiscale.

INPUT	Data di nascita	
CLASSI VALIDE	C_0008	Stringa del tipo gg/mm/aaaa.
CLASSI NON VALIDE	C_0009	Stringa discorde col tipo gg/mm/aaaa.

INPUT	Telefono, cellulare	
CLASSI VALIDE	C_0010	Stringa numerica con lunghezza fissa di 10 caratteri
CLASSI NON VALIDE	C_0011	Stringa alfanumerica di lunghezza variabile.
	C_0012	Stringa numerica minore di 10 caratteri.
	C_0013	Stringa numerica maggiore di 10 caratteri.

INPUT	e-mail	
CLASSI VALIDE	C_0014	Stringa alfanumerica con formato "nomeUtente"@provider"
CLASSI NON VALIDE	C_0015	Stringa alfanumerica discorde dal formato.

INPUT	Password	
CLASSI VALIDE	C_0016	Stringa alfanumerica presa tra 7 e 40 caratteri, contenente almeno una lettera maiuscola e almeno una cifra.

CLASSI NON VALIDE	C_0017	Stringa alfanumerica minore di 7 caratteri.
	C_0018	Stringa alfanumerica maggiore di 40 caratteri.
	C_0019	Stringa alfanumerica compresa tra 7 e 40 caratteri che non contiene almeno una lettera maiuscola.
	C_0020	Stringa alfanumerica compresa tra 7 e 40 caratteri che non contiene almeno una cifra.

INPUT	CAP	
CLASSI VALIDE	C_0021	Stringa numerica di 5 cifre.
CLASSI NON VALIDE	C_0022	Stringa numerica minore di 5 cifre.
	C_0023	Stringa numerica maggiore di 5 cifre.
	C_0024	Stringa contenente almeno una lettere o un carattere speciale.

INPUT	Password confermata	
CLASSI VALIDE	C_0025	Stringa alfanumerica presa tra 7 e 40 caratteri, contenente almeno una lettera maiuscola e almeno una cifra, che coincide con la password.
CLASSI NON VALIDE	C_0026	Stringa incoerente con la password.

### 9.1.2. FORMAL TEST SPECIFICATION

INPUT	Nome, cognome, via, città	
ID_CODICE		SPECIFICA FORMALE
C_0001		[propertynome_cognome_via_cittàOK]
C_0002		[errore]
C_0003		[errore]

INPUT	Codice fiscale	
ID_CODICE		SPECIFICA FORMALE
C_0004		[propertycodice_fiscaleOK]
C_0005		[errore]
C_0006		[errore]
C_0007		[errore]

INPUT	Data di nascita	
ID_CODICE		SPECIFICA FORMALE
C_0008		[propertydata_nascitaOK]
C_0009		[errore]

INPUT	Telefono, cellulare	
ID_CODICE		SPECIFICA FORMALE
C_0010		[propertytelefono_cellulareOK]
C_0011		[errore]
C_0012		[errore]

C_0013	[errore]
--------	----------

INPUT	Email
ID_CODICE	SPECIFICA FORMALE
C_0014	[propertyEmailOK]
C_0015	[errore]

INPUT	Password
ID_CODICE	SPECIFICA FORMALE
C_0016	[propertyPasswordOK]
C_0017	[errore]
C_0018	[errore]
C_0019	[errore]
C_0020	[errore]

INPUT	CAP
ID_CODICE	SPECIFICA FORMALE
C_0021	[propertyCAPOK]
C_0022	[errore]
C_0023	[errore]
C_0024	[errore]

INPUT	Conferma Password
ID_CODICE	SPECIFICA FORMALE
C_0025	[propertyConferma_PasswordOK]
C_0026	[errore]

### 9.1.3. TEST CASE

TEST CASE	TC_Registrazione_01	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0004	SSSGPP95E10H703E
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0008	10/05/1995
Telefono	C_0010	3921248463
Cellulare	C_0010	089562389
Email	C_0014	giuseppe@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0001	Giovanni battaglia
CAP	C_0021	84084
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_02	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703EAA
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0008	10/05/1995
Telefono	C_0010	3921248463
Cellulare	C_0010	089562389
Email	C_0014	giuseppe@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0001	Giovanni battaglia
CAP	C_0021	84084
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_03	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0007	SSSGPP95E10H703EAA
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0009	10/05/95
Telefono	C_0010	3921248463
Cellulare	C_0010	089562389
Email	C_0014	giuseppe@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0001	Giovanni battaglia
CAP	C_0021	84084
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_04	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0004	SSSGPP95E10H703E
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0008	10/05/1995
Telefono	C_0011	39212484637868t7
Cellulare	C_0010	089562389
Email	C_0014	giuseppe@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0001	Giovanni battaglia
CAP	C_0021	84084
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_04	
PARAMETRO	SCELTA	VALORE

Codice fiscale	C_0004	SSSGPP95E10H703E
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0008	10/05/1995
Telefono	C_0010	3921248463
Cellulare	C_0010	089562389
Email	C_0014	giuseppe@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0001	Giovanni battaglia
CAP	C_0021	84084
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_05	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0004	SSSGPP95E10H703E
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0008	10/05/1995
Telefono	C_0010	3921248463
Cellulare	C_0010	089562389
Email	C_0015	giusepp@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0001	Giovanni battaglia
CAP	C_0021	84084
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_06	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0004	SSSGPP95E10H703E
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0008	10/05/1995
Telefono	C_0010	3921248463
Cellulare	C_0010	089562389
Email	C_0014	giuseppe@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0026	Universita
Via	C_0001	Giovanni battaglia
CAP	C_0021	84084
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_07	
PARAMETRO	SCELTA	VALORE
Codice fiscale	C_0004	SSSGPP95E10H703E
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0008	10/05/1995



Telefono	C_0010	3921248463
Cellulare	C_0010	089562389
Email	C_0014	giuseppe@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0002	Giovanni battaglia fturifgrtdturnfctrydvornntyfgjjgk
CAP	C_0021	84084
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_08	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0005	SSSGPP95E1
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0009	105/1995
Telefono	C_0010	3921248463
Cellulare	C_0010	089562389
Email	C_0014	giuseppe@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0001	Giovanni battaglia
CAP	C_0024	8408h478
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_09	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0004	SSSGPP95E10H703E
Nome	C_0003	
Cognome	C_0001	Sessa
Data nascita	C_0008	10/05/1995
Telefono	C_0010	3921248463
Cellulare	C_0010	0895f62389
Email	C_0014	giuseppe@hotmail.it
Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0001	Giovanni battaglia
CAP	C_0021	84084
Città	C_0001	Fisciano

TEST CASE	TC_Registrazione_10	
PARAMETRO	SCelta	VALORE
Codice fiscale	C_0007	SSSGPP95E10H70RR
Nome	C_0001	Giuseppe
Cognome	C_0001	Sessa
Data nascita	C_0008	10/05/1995
Telefono	C_0010	3921248463
Cellulare	C_0010	089562389
Email	C_0014	giuseppe@hotmail.it

Password	C_0016	Universita95
Conferma password	C_0025	Universita95
Via	C_0001	Giovanni battaglia
CAP	C_0022	8484
Città	C_0001	Fisciano

## 9.2. LOGIN

Le funzionalità da testare riguardano una componente del sottosistema ACCESSO, in particolare la funzionalità di login.

### 9.2.1. CLASSI DI EQUIVALENZA

INPUT	Username	
CLASSI VALIDE	C_0027	Stringa alfanumerica con formato "nomeUtente.@provider".
CLASSI NON VALIDE	C_0028	Stringa alfanumerica discorde dal formato.
	C_0029	Stringa alfanumerica uguale a 0.

INPUT	Password	
CLASSI VALIDE	C_0030	Stringa alfanumerica presa tra 7 e 40 caratteri, contenente almeno una lettera maiuscola e almeno una cifra.
CLASSI NON VALIDE	C_0031	Stringa alfanumerica minore di 7 caratteri.
	C_0032	Stringa alfanumerica maggiore di 40 caratteri.
	C_0033	Stringa alfanumerica compresa tra 7 e 40 caratteri che non contiene almeno una lettera maiuscola.
	C_0034	Stringa alfanumerica compresa tra 7 e 40 caratteri che non contiene almeno una cifra.

### 9.2.2. FORMAL TEST SPECIFICATION

INPUT	Username	
ID_CODICE	SPECIFICA FORMALE	
C_0027	[propertyusernameOK]	
C_0028	[errore]	
C_0029	[errore]	

INPUT	Password	
ID_CODICE	SPECIFICA FORMALE	
C_0030	[propertyPasswordOK]	
C_0031	[errore]	

C_0032	[errore]
C_0033	[errore]
C_0034	[errore]

### 9.2.3. TEST CASE

TEST CASE	TC_Login_01	
PARAMETRO	SCelta	VALORE
Username	C_0027	Giuseppe@hotmail.it
Password	C_0030	Universita95

TEST CASE	TC_Login_02	
PARAMETRO	SCelta	VALORE
Username	C_0028	Giuseppe.hotmail.it
Password	C_0030	Universita95

TEST CASE	TC_Login_03	
PARAMETRO	SCelta	VALORE
Username	C_0027	Giuseppe@hotmail.it
Password	C_0034	Universita

TEST CASE	TC_Login_04	
PARAMETRO	SCelta	VALORE
Username	C_0029	
Password	C_0033	universita95

## 9.3. RICERCA PRODOTTO

Le funzionalità da testare riguardano una componente del sottosistema RICERCA, in particolare la funzionalità di ricerca di un prodotto.

### 9.3.1. CLASSI DI EQUIVALENZA

INPUT	Codice pezzo	
CLASSI VALIDE	C_0035	Stringa alfanumerica compresa tra 1 e 20 caratteri.
CLASSI NON VALIDE	C_0036	Stringa alfanumerica minore di 1 caratteri.
	C_0037	Stringa alfanumerica maggiore di 20 caratteri.

INPUT	Marchio	
CLASSI VALIDE	C_0038	Stringa alfanumerica compresa tra 1 e 30 caratteri.
CLASSI NON VALIDE	C_0039	Stringa alfanumerica minore di 1 caratteri.
	C_0040	Stringa alfanumerica maggiore di 30 caratteri.

INPUT	Modello	
CLASSI VALIDE	C_0041	Stringa alfanumerica compresa tra 1 e 30 caratteri.
CLASSI NON VALIDE	C_0042	Stringa alfanumerica minore di 1 caratteri.
	C_0043	Stringa alfanumerica maggiore di 30 caratteri.

### 9.3.2. FORMAL TEST SPECIFICATION

INPUT	Codice pezzo	
ID_CODICE	SPECIFICA FORMALE	
C_0035	[propertycodice_pezzoOK]	
C_0036	[errore]	
C_0037	[errore]	

INPUT	Marchio	
ID_CODICE	SPECIFICA FORMALE	
C_0038	[propertymarchioOK]	
C_0039	[errore]	
C_0040	[errore]	

INPUT	Modello	
ID_CODICE	SPECIFICA FORMALE	
C_0041	[propertymodelloOK]	
C_0042	[errore]	
C_0043	[errore]	

### 9.3.3. TEST CASE

TEST CASE	TC_RicercaProdotto_01	
PARAMETRO	SCELTA	VALORE
CodicePezzo	C_0035	00000000A
Marchio	C_0038	Alfa
Modello	C_0041	147

TEST CASE	TC_RicercaProdotto_02	
PARAMETRO	SCELTA	VALORE
CodicePezzo	C_0036	
Marchio	C_0038	Alfa
Modello	C_0041	147

TEST CASE	TC_RicercaProdotto_03	
PARAMETRO	SCELTA	VALORE
CodicePezzo	C_0037	00001234567891415161223065120650
Marchio	C_0038	Alfa
Modello	C_0041	147

TEST CASE	TC_RicercaProdotto_04	
PARAMETRO	SCELTA	VALORE

CodicePezzo	C_0035	00000000A
Marchio	C_0039	
Modello	C_0041	147

TEST CASE	TC_RicercaProdotto_05	
PARAMETRO	SCELTA	VALORE
CodicePezzo	C_0036	
Marchio	C_0039	
Modello	C_0042	

#### 9.4. CARICAMENTO PRODOTTO (GESTORE)

Le funzionalità da testare riguardano una componente del sottosistema CARICAMENTO, in particolare la funzionalità di Caricamento di un prodotto da parte di un gestore.

##### 9.4.1 CASI DI EQUIVALENZA

INPUT	Codice pezzo	
CLASSI VALIDE	C_0044	Stringa alfanumerica compresa tra 1 e 20 caratteri.
CLASSI NON VALIDE	C_0045	Stringa alfanumerica minore di 1 caratteri.
	C_0046	Stringa alfanumerica maggiore di 20 caratteri.

INPUT	Marchio	
CLASSI VALIDE	C_0047	Stringa alfanumerica compresa tra 1 e 30 caratteri.
CLASSI NON VALIDE	C_0048	Stringa alfanumerica minore di 1 caratteri.
	C_0049	Stringa alfanumerica maggiore di 30 caratteri.

INPUT	Modello	
CLASSI VALIDE	C_0050	Stringa alfanumerica compresa tra 1 e 30 caratteri.
CLASSI NON VALIDE	C_0051	Stringa alfanumerica minore di 1 caratteri.
	C_0052	Stringa alfanumerica maggiore di 30 caratteri.

INPUT	Prezzo	
CLASSI VALIDE	C_0053	Numero decimale con due numeri dopo la virgola.
CLASSI NON VALIDE	C_0054	Carattere alfabetico all'interno del prezzo.

INPUT	Prezzo scontato	
CLASSI VALIDE	C_0055	Numero decimale con due numeri dopo la virgola.
CLASSI NON VALIDE	C_0056	Carattere alfabetico all'interno del prezzo.

INPUT	Descrizione	
CLASSI VALIDE	C_0057	Stringa alfanumerica di lunghezza arbitraria.

CLASSI NON VALIDE	C_0058	Stringa alfanumerica con lunghezza minore di 1 carattere.
-------------------	--------	---

INPUT	Quantità	
CLASSI VALIDE	C_0059	Numero intero maggiore o uguale a 0.
CLASSI NON VALIDE	C_0060	Numero intero minore di 0.
	C_0061	Carattere alfabetico all'interno della quantità.

#### 9.4.2. FORMAL TEST SPECIFICATION

INPUT	Codice pezzo	
ID_CODICE		SPECIFICA FORMALE
C_0035		[propertycodice_pezzoOK]
C_0036		[errore]
C_0037		[errore]

INPUT	Marchio	
ID_CODICE		SPECIFICA FORMALE
C_0038		[propertymarchioOK]
C_0039		[errore]
C_0040		[errore]

INPUT	Modello	
ID_CODICE		SPECIFICA FORMALE
C_0041		[propertymodelloOK]
C_0042		[errore]
C_0043		[errore]

INPUT	Prezzo	
ID_CODICE		SPECIFICA FORMALE
C_0053		[propertyPrezzoOK]
C_0054		[errore]

INPUT	Prezzo Scontato	
ID_CODICE		SPECIFICA FORMALE
C_0055		[propertyPrezzo_ScontatoOK]
C_0056		[errore]

INPUT	Descrizione	
ID_CODICE		SPECIFICA FORMALE
C_0057		[propertyDescrizioneOK]
C_0058		[errore]

INPUT	Quantità	
-------	----------	--

ID_CODICE	SPECIFICA FORMALE
C_0059	[propertyQuantitàOK]
C_0060	[errore]
C_0061	[errore]

### 9.4.3. TEST CASE

TEST CASE	TC_Caricamento_01	
PARAMETRO	SCELTA	VALORE
CodicePezzo	C_0044	10000A
Marchio	C_0047	Alfa
Modello	C_0050	147
Prezzo	C_0053	20.05
Prezzo scontato	C_0055	15.00
Descrizione	C_0057	Pattini interiori per alfa 147
Quantità	C_0059	10

TEST CASE	TC_Caricamento_03	
PARAMETRO	SCELTA	VALORE
CodicePezzo	C_0044	10000A
Marchio	C_0047	
Modello	C_0050	147
Prezzo	C_0053	20.05
Prezzo scontato	C_0055	15.00
Descrizione	C_0057	Pattini interiori per alfa 147
Quantità	C_0059	10

TEST CASE	TC_Caricamento_04	
PARAMETRO	SCELTA	VALORE
CodicePezzo	C_0044	10000A
Marchio	C_0047	Alfa
Modello	C_0050	147
Prezzo	C_0053	20.05
Prezzo scontato	C_0055	15.00
Descrizione	C_0057	Pattini interiori per alfa 147
Quantità	C_0059	10

TEST CASE	TC_Caricamento_05	
PARAMETRO	SCELTA	VALORE
CodicePezzo	C_0044	10000A
Marchio	C_0047	Alfa
Modello	C_0050	147
Prezzo	C_0053	20.05
Prezzo scontato	C_0055	15.00
Descrizione	C_0057	
Quantità	C_0059	10

TEST CASE	TC_Caricamento_06	
PARAMETRO	SCELTA	VALORE
CodicePezzo	C_0044	10000A
Marchio	C_0047	Alfa
Modello	C_0050	147
Prezzo	C_0053	20.05
Prezzo scontato	C_0055	15.00
Descrizione	C_0057	Pattini interiori per alfa 147
Quantità	C_0059	10ab

## 9.5. QUANTITA' (PRODOTTO)

Le funzionalità da testare riguardano una componente del sottosistema PRODOTTO, in particolare la funzionalità di inserire la quantità di un prodotto che si vuole acquistare.

### 9.5.1. CLASSI DI EQUIVALENZA

INPUT	Quantità	
CLASSI VALIDE	C_0062	Numero intero maggiore o uguale a 0.
CLASSI NON VALIDE	C_0063	Numero intero minore di 0.
	C_0064	Carattere alfabetico all'interno della quantità.

### 9.5.2. FORMAL TEST SPECIFICATION

INPUT	Quantità	
ID_CODICE	SPECIFICA FORMALE	
C_0062	[propertyQuantitàOK]	
C_0063	[errore]	
C_0064	[errore]	

### 9.5.3. TEST CASE

TEST CASE	TC_Quantità_01	
PARAMETRO	SCELTA	VALORE
Quantità	C_0062	10

TEST CASE	TC_Quantità_02	
PARAMETRO	SCELTA	VALORE
Quantità	C_0063	-5

TEST CASE	TC_Quantità_03	
PARAMETRO	SCELTA	VALORE
Quantità	C_0064	15a



## **10. TESTING SCHEDULE**

Di seguito sono elencate la gestione dei rischi che occorre seguire durante la fase di testing, l'organizzazione delle attività di testing e la schedulazione delle attività durante il periodo stabilito per il testing.

### **10.1. GESTIONE DEI RISCHI**

I possibili rischi generati dalle attività di testing sono stati minimizzati diminuendo le componenti del sistema da implementare e, quindi, testare. Inoltre, effettuando un testing di tipo funzionale viene limitato lo sviluppo di stub e driver per il testing delle singole componenti e quindi l'introduzione di nuovi errori nel nuovo codice di cui sarebbero composti. D'altro canto il testing di funzionalità rallenta l'individuazione di errori qualora un caso di test avesse esito positivo, poiché l'utilizzo di più componenti per il test di una singola funzionalità viene eseguita ma non in modo completamente corretto. Qualora la fase di testing evidenziasse un numero di errori maggiore rispetto alla media attesa, viene pianificato un impegno maggiore dei membri del team sulle attività di testing ed in casi estremi l'abbandono delle altre attività fino a quando errori gravi (funzionalità non corretta, risultati errati, modifiche apportate in modo errato) non vengano risolti.

### **10.2. ORGANIZZAZIONE DELLE ATTIVITA'**

Le attività di testing devono svolgersi sulle singole funzionalità divise nei livelli di suddivisione del sistema, rispettando le direttive indicate dal documento di system design.