



**Università degli Studi di Salerno**  
Corso di Ingegneria del Software



*Enoteca  
Il Gocciolatoio*

*Bevi poco... Ma bevi bene!*

## **Test Plan**

Data: 15/01/2021

## Coordinatore del Progetto:

Nome	Matricola
Francesco Di Palma	0512104586

## Partecipanti:

Nome	Matricola
Giovanni Di Mauro	0512104596
Francesco Di Palma	0512104586
Maria Giuseppina Mosca	0512106090
Francesco Saviano	0512104912

## Sommario

Introduzione .....	3
Relazioni con altri documenti .....	3
Funzionalità da testare e da non testare .....	3
Criteri pass/fai testing .....	4
Approccio .....	5
Sospensione e ripresa .....	5
Materiale per il testing .....	5
TEST GESTORE ACCOUNT .....	6
TEST GESTORE PRODOTTO .....	18

## Introduzione

Il test plan è un documento che si focalizza sugli aspetti manageriali del testing: gestisce lo sviluppo e le attività di testing effettuate sul sistema creato. Saranno identificati gli elementi e le funzionalità da testare, le strategie di testing, gli strumenti utilizzati per effettuarlo. Lo scopo del testing è quello di rilevare gli errori in maniera pianificata all'interno del codice realizzato, in modo che essi non si ripetano durante l'utilizzo da parte dell'utente finale. I risultati dei test servono per intervenire nei punti in cui sono presenti défaillances.

## Relazioni con altri documenti

Il test planning è in stretta relazione con i documenti prodotti fino, infatti i test case sono basati sulle funzionalità individuate in questi, particolarmente nel "documento di raccolta ed analisi dei requisiti"-(RAD).

La definizione dei sottosistemi ed i servizi da porre al testing sono stati individuati tramite il "System Design Document"-(SDD) e le classi sono state definite sulla base dell' "Object Design Document"-(ODD).

## Funzionalità da testare e da non testare

Di seguito le funzionalità introdotte nel sistema che saranno sottoposte a test, suddivise per ogni gestione del sistema:

### **GESTORE ACCOUNT**

- registrazione utente
- aggiunta indirizzo all'account
- cancellazione utente
- modifica indirizzo
- modifica profilo

- modifica profilo da parte dell'amministratore
- eliminazione di un account da parte dell'amministratore
- Effettuare login
- Effettuare logout

#### **GESTORE CARRELLO**

- Aggiunta prodotti al carrello
- Rimozione prodotti dal carrello
- Modifica della quantità dei prodotti contenuti nel carrello
- Effettuare un acquisto

#### **GESTORE ORDINE**

- Cancellazione di un ordine da parte dell'amministratore
- Visualizzare i dettagli degli ordini
- Visualizzare tutti gli ordini dei clienti, da parte dell'amministratore

#### **GESTORE PRODOTTO**

- Inserimento prodotto
- Modifica prodotto
- Visualizzazione di uno specifico prodotto
- Ricerca di un prodotto

## Criteri pass/fai testing

Lo scopo del testing è trovare una failure nel programma, quindi, il testing va a buon fine se l'output osservato è diverso dall'output atteso; una volta riscontrata una failure (nel caso sia presente nel sistema), si interviene per correggerla e si effettua di nuovo il testing per appurare che non abbia prodotto effetti collaterali, producendo altre failure.

## Approccio

L'approccio della fase di testing si compone di tre fasi. Si inizia con il testing di unità che ha lo scopo di testare le componenti del sistema singolarmente, poi si passa al testing di integrazione in cui le componenti del sistema verranno combinate e testate come un unico gruppo ed, infine, si effettua il testing di sistema che mira a verificare il funzionamento dell'intero sistema.

Per il testing di unità si utilizzerà la tecnica "BLACK-BOX". Con il "BLACK-BOX" testing ci focalizzeremo sul comportamento dell'input/output delle singole componenti senza tener conto della loro struttura interna. A causa della mancanza di fattibilità di effettuare un test esaustivo per l'ingente quantitativo di dati di input, verrà utilizzata la strategia del "Category Partition", che consente di decomporre lo spazio di input in categorie per poi partizionare le "categorie" in classi di equivalenza chiamate "scelte". Al termine saranno specificate le "combinazioni" delle scelte da testare creando delle istanze di casi di test specificando i valori dei dati effettivi per ciascuna scelta e determinare i risultati corrispondenti. Mediante il "Category Partition" otterremo, quindi, un test efficiente e privo di ridondanze.

Per il testing d'integrazione si utilizzerà la tecnica "WHITE-BOX". La "WHITE-BOX" testing viene utilizzata per testare la logica interna del sottosistema o dell'oggetto considerato, infatti, per trovare un errore nel codice bisogna usare dei dati che percorrono la parte errata del programma. Per testare una parte di programma si introduce il concetto di "cammino", ovvero, una sequenza di istruzioni attraversata durante un'esecuzione. Non esiste un criterio capace di testare ogni singolo cammino (dato l'elevato numero di questi ultimi), è possibile trovare un numero finito di cammini indipendenti che combinati tra loro forniscono la maggior parte dei restanti cammini.

Per determinare se il sistema rispecchia tutti i requisiti funzionali e globali allora sarà effettuato il testing di sistema tramite il tool "Selenium".

## Sospensione e ripresa

La fase di testing può essere sospesa se si riscontrasse un difetto nel sistema per essere poi ripresa dopo aver risolto il problema riscontrato.

## Materiale per il testing

Gli strumenti utilizzati sono:

- WebServer Apache Tomcat 9 in locale, il quale gira sul sistema
- Client-web per effettuare richieste al server
- MySQL che gestisce il database

- Selenium IDE per il test di sistema
- JUnit per il test di unità ed il test di integrazione

## TEST GESTORE ACCOUNT

## TC\_AggiungiUtente

Parametro nome:	
CATEGORIA	SCELTE
Lunghezza nome-LN	1.Lunghezza = 0 - campo vuoto <b>[invalidLNvalue]</b> 2. lunghezza >=1 – <b>[propety validLNvalue]</b>
Formato nome - FN	1. Non rispetta il formato [if <b>validLNvalue</b> ] <b>[invalidFNvalue]</b> 2. Rispetta il formato [if <b>validLNvalue</b> ] [property <b>validFNvalue</b> ]

Parametro cognome:	
CATEGORIA	SCELTE
Lunghezza cognome-LC	1.Lunghezza = 0 - campo vuoto <b>[invalidLCvalue]</b> 2. lunghezza >=1 – <b>[propety validLCvalue]</b>
Formato cognome - FC	1. Non rispetta il formato [if <b>validLCvalue</b> ] <b>[invalidFCvalue]</b> 2. Rispetta il formato [if <b>validLCvalue</b> ] <b>[property validFNvalue]</b>

Parametro username:	
CATEGORIA	SCELTE
Lunghezza username-LU	1.lunghezza = 0 - campo vuoto <b>[invalidLUvalue]</b> 2. lunghezza >=1 –

	[propety <b>validLUvalue</b> ]
Formato username - FU	1. Non rispetta il formato [if <b>valideLUvalue</b> ] [ <b>invalidFUvalue</b> ] 2. Rispetta il formato [if <b>validLUvalue</b> ] [property <b>validFUvalue</b> ]
Esiste username - EU	1. Esiste nel DB [if <b>validLUvalue</b> AND <b>validFUvalue</b> ] [ <b>invalidEUvalue</b> ] 2. Non esiste nel DB [if <b>validLUvalue</b> AND <b>validFUvalue</b> ] [propety <b>validEUvalue</b> ]

Parametro e-mail: Formato: “/^[0-9A-Za-z]+ \$/”	
CATEGORIA	SCELTE
Lunghezza e-mail-LE	1.lunghezza = 0 - campo vuoto [ <b>invalidLEvalue</b> ] 2. lunghezza >=1 – [propety <b>validLEvalue</b> ]
Formato e-mail - FE	1. Non rispetta il formato [if <b>valideLEvalue</b> ] [ <b>invalidFEvalue</b> ] 2. Rispetta il formato [if <b>validLEvalue</b> ] [property <b>validFEvalue</b> ]
Esiste e-mail - EE	1. Esiste nel DB [if <b>validLEvalue</b> AND <b>validFEvalue</b> ] [ <b>invalidEEvalue</b> ] 2. Non esiste nel DB [if <b>validLEvalue</b> AND <b>validFEvalue</b> ] [propety <b>validEEvalue</b> ]

Parametro password: Formato: “/^[0-9A-Za-z]+ \$/”
--

CATEGORIA	SCELTE
Lunghezza password-LP	1.lunghezza = 0 - campo vuoto [invalidLPvalue] 2. lunghezza >=1 – [propety validLPvalue]
Formato password - FP	1. Non rispetta il formato [if valideLPvalue] [invalidFPvalue] 2. Rispetta il formato [if validLPvalue] [property validFPvalue]
Esiste password - EP	1. Esiste nel DB [if validLPvalue] AND validFPvalue] [invalidEPvalue] 2. Non esiste nel DB [if validLPvalue AND validFPvalue] [propety validEPvalue]

Parametro data: Formato: “^\\d{4}\\-(0[1-9] 1[012])\\-(0[1-9] [12][0-9] 3[01])\$”	
CATEGORIA	SCELTE
Lunghezza data-LD	1.lunghezza = 0 - campo vuoto [invalidLPvalue] 2. lunghezza >=1 – [propety validLPvalue]
Formato data – FD	1. Non rispetta il formato [if valideLPvalue] [invalidFPvalue] 2. Rispetta il formato [if validLPvalue] [property validFPvalue]
Esiste data - ED	1. Esiste nel DB [if validLPvalue] AND validFPvalue] [invalidEPvalue] 2. Non esiste nel DB [if validLPvalue AND validFPvalue] [propety validEPvalue]

CODICE	COMBINAZIONE	ESITO
TC_AggiungiUtente	LN2.FN2.LU2.FU2.EU2.LE2.FE2.EE2.LP2.FP2.EP2.LD2.FD2.ED2	CORRETTO
TC_AggiungiUtenteNomeVuoto	LN1	ERROR
TC_AggiungiUtenteCogomeVuoto	LC1	ERROREE
TC_AggiungiUtenteUsernameVuoto	LU1	ERRORE
TC_AggiungiUtenteEmailVuoto	LE1	ERRORE
TC_AggiungiUtentePassVuoto	LP1	ERRORE







[illegible]

Parametro password: Formato: “/^[0-9A-Za-z]+\$/”	
CATEGORIA	SCELTE
Lunghezza password-LP	1.lunghezza = 0 - campo vuoto [invalidLPvalue] 2. lunghezza >=1 – [propety validLPvalue]
Corrispondenza password e-mail- CPE	1. c'è corrispondenza tra password e e-mail nel DB [if validLPvalue] [propety invalidCPEvalue] 2. Non esiste nel DB [if validLPvalue] [propety validCPEvalue]

CODICE	COMBINAZIONE	ESITO
TC_GestioneLoginUtentetrovato	LE2.EE2.LP2.CPE2	CORRETTO
TC_GestioneLoginUtenteUsernameVuoto	LE1	ERRATO
TC_gestioneLoginUtentePasswordVuoto	LP1	ERRATP

TC GestioneModificaIndirizzo







CODICE	COMBINAZIONE	ESITO
TC_GestioneModificaUtente	LN2.FN2.LU2.FU2.EU2.LE2.FE2.EE2.LP2.FP2.EP2.LD2.FD2.ED2	CORRETTO
TC_GestioneModificaUtenteNomeVuoto	LN1	ERROR
TC_GestioneModificaUtenteCogomeVuoto	LC1	ERROREE
TC_GestioneModificaUtenteUsernameVuoto	LU1	ERRORE
TC_GestioneModificaUtenteEmailVuoto	LE1	ERRORE
TC_GestioneModificaUtentePassVuoto	LP1	ERRORE
TC_GestioneModificaUtenteDateVuoto	LD1	ERRORE
TC_GestioneModificaUtenteNomeErrato	FN1	ERRORE
TC_GestioneModificaUtenteCognomeErrato	FC1	ERRORE
TC_GestioneModificaUtenteUsernameErrato	FU1	ERRORE
TC_GestioneModificaUtenteEmailErrato	FE1	ERRORE
TC_GestioneModificaUtentePasswordErrato	FP1	ERRORE
TC_GestioneModificaUtenteDateErrato	FD1	ERRORE

## TC\_GestioneModificaUtenteByAdmin

Parametro username:	
Formato: "[~@#\^\\$\&\\*\ \(\)\_\\+=\[\]\{\}\ \ \\\\\\.\\?]*+\$"	
CATEGORIA	SCELTE
Lunghezza username-LU	1.lunghezza = 0 - campo vuoto <b>[invalidLUvalue]</b> 2. lunghezza >=1 – <b>[propety validLUvalue]</b>
Formato username - FU	1. Non rispetta il formato [if <b>valideLUvalue</b> ] <b>[invalidFUvalue]</b> 2. Rispetta il formato [if <b>validLUvalue</b> ] [property <b>validFUvalue</b> ]
Esiste username - EU	1. Esiste nel DB [if <b>validLUvalue</b> ] <b>AND validFUvalue</b> <b>[validEUvalue]</b> 2. Non esiste nel DB [if <b>validLUvalue AND</b> <b>validFUvalue</b> ] [propety <b>invalidEUvalue</b> ]















## TC\_ModificaProdotto

Parametro nome:	
Formato: "[~@#\\^\\\$&\\* \\(\\)_\\+=\\[\\]\\ \\{\\}\\ \\ \\ \\ \\ \\.\\ ?]*+\$"	
CATEGORIA	SCELTE
Lunghezza nome-LN	1. Lunghezza = 0 - campo vuoto <b>[invalidLNvalue]</b> 2. lunghezza >=1 – <b>[propety validLNvalue]</b>
Formato nome - FN	1. Non rispetta il formato [if <b>validLNvalue</b> ] <b>[invalidFNvalue]</b> 2. Rispetta il formato [if <b>validLNvalue</b> ] <b>[property validFNvalue]</b>

[illegible]

Parametro descrizione: Formato: “/^ [A-Za-z0-9]+ \$/”	
CATEGORIA	SCELTE
Lunghezza descrizione-LD	1. Lunghezza = 0 - campo vuoto [invalidLDvalue] 2. lunghezza >=1 – [propety validLDvalue]
Formato descrizione- FD	1. Non rispetta il formato [if validLDvalue] [invalidFDvalue] 2. Rispetta il formato [if validLDvalue] [property validFDvalue]

<b>Parametro prezzo:</b>	
<b>Formato: "[~@#\\^\\\$&amp;\\ * \\(\\)_\\ =\\ \\[\\]\\ \\{\\}\\ \\ \\ \\ \\.\\ ?]*+\$"</b>	
CATEGORIA	SCELTE
Lunghezza prezzo-LP	1.Lunghezza = 0 - campo vuoto [invalidLPvalue]







## TC\_RicercaProdotto

Parametro: nome	
Formato: "[~@#\^\\\$&\\* \\/\\_\\+=\\[\\]\\{\\}\\ \\ \\ \\.\\ \\?]*+\$"	
CATEGORIA	SCELTE
Lunghezza nome-LN	1. lunghezza = 0 - campo vuoto <b>[invalidLNvalue]</b> 2. lunghezza >=1 – <b>[property validLNvalue]</b>
Formato nome- FN	1. Non rispetta il formato [if <b>valideLNDvalue</b> ] <b>[invalidFNvalue]</b> 2. Rispetta il formato [if <b>validLNvalue</b> ] <b>[property validFNvalue]</b>
Esiste nome - EN	1. Esiste nel DB [if <b>validLNvalue</b> <b>AND validFNvalue</b> <b>[validENvalue]</b> 2. Non esiste nel DB [if <b>validLNvalue AND</b> <b>validFNvalue]</b> <b>[property invalidENvalue]</b>