



Università degli Studi di Salerno

Corso di Ingegneria del Software

DressApp

Test Plan

Versione 1.0

Data: 20/12/2022

Coordinatore del progetto:

Nome	Matricola
Francesco Ferrara	0512109789

Partecipanti:

Nome	Matricola
Alaia Michele	0512112861
Auleta Antonio	0512106822
Ferrara Francesco	0512109789
Giammarino Josè Luis	0512107086

Scritto da:

Alaia Michele

Auleta Antonio

Ferrara Francesco

Giammarino Josè Luis

Revision History:

Data	Versione	Descrizione	Autori
13/12/2022	0.1	Inizio stesura TestPlan	Tutti i partecipanti
20/12/2022	1.0	Completamento TestPlan	Tutti i partecipanti

Indice

1. <u>Introduzione.....</u>	<u>4</u>
2. <u>Relazioni con altri documenti.....</u>	<u>4</u>
3. <u>Funzionalità da testare.....</u>	<u>4</u>
<u>3.1 Gestione Account.....</u>	<u>4</u>
<u>3.2 Gestione Cappello.....</u>	<u>4</u>
<u>3.3 Gestione Carrello.....</u>	<u>4</u>
<u>3.4 Gestione Ordini.....</u>	<u>4</u>
<u>3.5 Gestione Dettaglio Ordini.....</u>	<u>5</u>
<u>3.6 Gestione Pagamenti.....</u>	<u>5</u>
4. <u>Criteri pass / fail testing.....</u>	<u>5</u>
5. <u>Approccio.....</u>	<u>5</u>
6. <u>Sospensione e ripresa.....</u>	<u>6</u>
7. <u>Materiale per il testing</u>	<u>6</u>
8. <u>Formato input.....</u>	<u>6</u>
9. <u>Test.....</u>	<u>7</u>
<u>9.1 Test Gestione Account.....</u>	<u>7</u>
<u>9.2 Test Gestione Cappello.....</u>	<u>10</u>
<u>9.3 Test Gestione Carrello.....</u>	<u>16</u>
<u>9.4 Test Gestione Ordini.....</u>	<u>17</u>
<u>9.5 Test Gestione Dettaglio Ordini.....</u>	<u>18</u>
<u>9.6 Test Gestione Pagamenti.....</u>	<u>18</u>

1. 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 défaillance.

2. Relazioni con altri documenti

Il test planning è in stretta relazione con i documenti prodotti fino ad ora:

- I test case sono basati sulle funzionalità individuate in questi, particolarmente nel "Requirements Analysis Document" (RAD).
- La definizione dei sottosistemi ed i servizi da porre al testing sono stati individuati tramite il "System Design Document" (SDD)
- Le classi sono state definite sulla base dell'"Object Design Document" (ODD)

3. Funzionalità da testare

3.1 Gestione Account

- Registrazione
- Login
- Modifica Email
- Modifica Password

3.2 Gestione Cappello

- Aggiungi Cappello al catalogo
- Modifica Cappello
- Ricerca Cappello

3.3 Gestione Carrello

- Aggiungi prodotto
- Visualizza carrello

3.4 Gestione Ordini

- Ricerca ordini

3.5 Gestione Dettaglio Ordini

- Modifica quantità prodotto nel carrello

3.6 Gestione Pagamenti

- Aggiungi carta
- Paga ora

4. Criteri pass / fail 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.

5. 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 questa tecnica di 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 determinando 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**". Questa tecnica di 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 tramite il tool "**Selenium**".

6. 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.

7. Materiale per il testing

Gli strumenti utilizzati sono:

- Web Server Apache Tomcat in locale, il quale gira sul sistema
- Client-web per effettuare richieste al server
- MySQL
- Selenium IDE
- JUnit
- Mockito

8. Formato input

Concetto	Tipo di inserimento	Grandezza
Nome	[A-z]	[1-20]
Cognome	[A-z]	[1-20]
Email	[A-z],[@]	[1-30]
Password	[A-z],[? ,! , (,) , . ,] , [0-9]	[9-15]
Numero carta	[0-9]	[13-16]
Intestatario	[A-z]	[2-40]
Data di scadenza	[mm/aaaa]	[6]
CVV	[0-9]	[3]
Nome Cappello	[A-z]	[1-50]
Descrizione	[A-z]	[1-150]
Prezzo	[0-9]	[1-10]
Categoria	[A-z]	[1-50]
Foto	[A-z]	[1-50]

9. Test

9.1 Test Gestione Account

TC_Registrazione

Parametro: Nome	
Categorie	Scelte
Lunghezza nome - LN	1. Lunghezza = 0 - campo vuoto [property invalidLNValue] 2. Lunghezza <= 20 [property validLNValue]
Formato nome - FN	1. Non rispetta il formato [property invalidFNValue] 2. Rispetta il formato [<i>if</i> validLNValue] [property validFNValue]

Parametro: Cognome	
Categorie	Scelte
Lunghezza cognome - LC	1. Lunghezza = 0 - campo vuoto [property invalidLCValue] 2. Lunghezza <= 20 [property validLCValue]
Formato cognome - FC	1. Non rispetta il formato [property invalidFCValue] 2. Rispetta il formato [<i>if</i> validLNValue] [property validFCValue]

Parametro: Email	
Categorie	Scelte
Lunghezza email - LE	1. Lunghezza = 0 - campo vuoto [property invalidLEValue] 2. 1<= Lunghezza <= 30 [property validLEValue]
Formato email - FE	1. Non rispetta il formato [property invalidFEValue] 2. Rispetta il formato [<i>if</i> validLNValue] [property validFEValue]
Esistenza nel database - EXEM	1. Esiste nel database [<i>if</i> validFEValue] [property EXEM] 2. Non esiste nel database [property NotEXEM]

Parametro: Password	
Categorie	Scelte
Lunghezza password - LP	1. Lunghezza = 0 - campo vuoto [property invalidLNValue] 2. $9 \leq \text{Lunghezza} \leq 15$ [property validLNValue] 3. Lunghezza < 9, Lunghezza > 15 [property invalidLNValue]
Formato password - FP	1. Non rispetta il formato [property invalidFNValue] 2. Rispetta il formato [<i>if</i> validLNValue] [property validFNValue]

Codice	Combinazioni	Esito
TC_RegistrazioneOK	LN2.FN2.LC2.FC2.LE2.FE2.EXEM2.LP2.FP2	"Registrazione avvenuta con successo!"
TC_RegistrazioneEmailInUso	LN2.FN2.LC2.FC2.LE2.FE2.EXEM1.LP2.FP2	"Email già esistente"
TC_RegistrazioneEmailNonValida	LN2.FN2.LC2.FC2.LE2.FE1.LP2.FP2	"Formato email non valido"
TC_RegistrazioneEmailMancante	LN2.FN2.LC2.FC2.LE1.LP2.FP2	"E-mail - campo obbligatorio"
TC_RegistrazionePasswordMancante	LN2.FN2.LC2.FC2.LE2.FE2.EXEM2.LP1	"Password - campo obbligatorio"
TC_RegistrazionePasswordNonValida	LN2.FN2.LC2.FC2.LE2.FE2.EXEM2.LP2.FP1	"Formato password non valido"
TC_RegistrazioneLunghezzaPasswordNonValida	LN2.FN2.LC2.FC2.LE2.FE2.EXEM2.LP3.FP2	"Lunghezza password non valida"
TC_RegistrazioneNomeMancante	LN1.LC2.FC2.LE2.FE2.EXEM2.LP2.FP2	"Nome - campo obbligatorio"
TC_RegistrazioneNomeNonValido	LN2.FN1.LC2.FC2.LE2.FE2.EXEM2.LP2.FP2	"Formato nome non valido"
TC_RegistrazioneCognomeMancante	LN2.FN2.LC1.LE2.FE2.EXEM2.LP2.FP2	"Cognome - campo obbligatorio"
TC_RegistrazioneCognomeNonValido	LN2.FN2.LC2.FC1.LE2.FE2.EXEM2.LP2.FP2	"Formato cognome non valido"

TC_Login

Parametro: Email	
Categorie	Scelte
Esistenza nel database - EXEM	1. Non esiste nel database [property NotEXEM] 2. Esiste nel database [property EXEM]

Parametro: Password	
Categorie	Scelte
Associazione all'utente - AS	1. Non è correttamente associata alla email [property invalidASValue] 2. È correttamente associata alla email [if EXEM] [property validASValue]

Codice	Combinazioni	Esito
TC_LoginOK	EXEM2.AS2	"Login effettuato!"
TC_LoginPasswordErrata	EXEM2.AS1	"Password Errata"
TC_LoginEmailInesistente	EXEM1	"Email non esistente"

TC_ModificaEmail

Parametro: Nuova Email	
Categorie	Scelte
Lunghezza Nuova e-mail – LNE	1. Lunghezza = 0 – campo vuoto [property invalidLNEValue] 2. 1 <= Lunghezza <= 30 [property validLNEValue]
Formato Nuova e-mail - FNE	1. Non rispetta il formato [property invalidFNEValue] 2. Rispetta il formato [if validLNEValue] [property validFNEValue]
Esistenza nel database - EXEM	1. Non esiste nel database [property NotEXEM] 2. Esiste nel database [property EXEM]

Codice	Combinazioni	Esito
TC_ModificaEmailOK	LNE2.FNE2.EXEM1	"Email modificata con successo"
TC_ModificaEmailGiaInUso	LNE2.FNE2.EXEM2	"Email già in uso"
TC_ModificaEmailMancante	LNE1	"Nuova email: campo obbligatorio"
TC_ModificaEmailFormatoErrato	LNE2.FNE1	"Formato nuova email non valido"

TC_ModificaPassword

Parametro: Nuova Password	
Categorie	Scelte
Lunghezza Nuova password – LNP	1. Lunghezza = 0 – campo vuoto [property invalidLNPValue] 2. 9<= Lunghezza <= 15 [property validLNPValue] 3. Lunghezza < 9, Lunghezza>15 [property invalidLNPValue]
Formato Nuova password - FNP	1. Non rispetta il formato [property invalidFNPNEValue] 2. Rispetta il formato [if validLNPValue] [property validFNPValue]

Codice	Combinazioni	Esito
TC_ModificaPasswordOK	LP2.FNP2	"Password modificata con successo"
TC_ModificaPasswordMancante	LNP1	"Nuova email: campo obbligatorio"
TC_ModificaPasswordLunghezzaErrata	LNP3.FNP2	"Lunghezza nuova password non valido"
TC_ModificaPasswordFormatoErrato	LNP2.FNP1	"Formato nuova password non valido"

9.2 Test Gestione Cappello

TC_AggiungiCappelloCatalogo

Parametro: Nome	
Categorie	Scelte
Lunghezza Nome- LN	1. Lunghezza = 0 - campo vuoto [property invalidLNValue] 2. Lunghezza <= 50 [property validLNValue] 3. Lunghezza > 50 [property invalidLNValue]
Formato Nome- FN	1. Non rispetta il formato [property invalidFNValue] 2. Rispetta il formato [if validLNValue] [property validFNValue]
Esiste nel database- EXN	1. Esiste nel database [property EXN] 2. Non esiste nel database [if validFNValue] [property NotEXN]

Parametro: Descrizione	
Categorie	Scelte
Lunghezza Descrizione- LD	1. Lunghezza = 0 - campo vuoto [property invalidLDValue] 2. Lunghezza <= 150 [property validLDValue] 3. Lunghezza >150 [property invalidLDValue]
Formato Descrizione- FD	1. Non rispetta il formato [property [invalidFDValue]] 2. Rispetta il formato [/if validLDValue] [property validFDValue]

Parametro: Prezzo	
Categorie	Scelte
Lunghezza Prezzo- LP	1. Lunghezza = 0 - campo vuoto [property invalidLPValue] 2. Lunghezza >=1 [property validLPValue] 3. Lunghezza >10 [property invalidLPValue]
Formato Prezzo- FP	1. Non rispetta il formato [property [invalidFPValue]] 2. Rispetta il formato [/if validLPValue] [property validFPValue]
Positività Prezzo- PP	1. Il prezzo è negativo [property invalidPPValue] 2. Il prezzo è non negativo [if validFPValue] [property validPPValue]

Parametro: Categoria	
Categorie	Scelte
Lunghezza Categoria- LC	1. Lunghezza = 0 - campo vuoto [property invalidLCValue] 2. Lunghezza <= 50 [property validLCValue] 3. Lunghezza >50 [property invalidLCValue]
Formato Categoria- FC	1. Non rispetta il formato [property [invalidFCValue]] 2. Rispetta il formato [/if validLCValue] [property validFCValue]

Parametro: Foto	
Categorie	Scelte
Lunghezza Foto- LF	1. Lunghezza = 0 - campo vuoto [property invalidLNValue] 2. Lunghezza <= 40 [property validLNValue] 3. Lunghezza >40 [property invalidLNValue]
Formato Foto- FF	1. Non rispetta il formato [property [invalidFNValue]] 2. Rispetta il formato [/if validLNEValue] [property validFNValue]

Codice	Combinazioni	Esito
TC_AggiuntaCatalogoOk	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Aggiunta avvenuta con successo!"
TC_AggiuntaNomeInUso	LN2.FN2.EXN1.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Nome prodotto già esistente!"
TC_AggiuntaNomeNonValido	LN2.FN1.EXN1.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Formato nome non valido!"
TC_AggiuntaNomeMancante	LN1.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Nome - campo obbligatorio!"
TC_AggiuntaNomeEccessivo	LN3.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Nome supera il limite di caratteri consentiti"
TC_AggiuntaDescrizioneMancante	LN2.FN2.EXN2.LD1.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Descrizione - campo obbligatorio!"
TC_AggiuntaDescrizioneEccessiva	LN2.FN2.EXN2.LD3.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Descrizione supera il limite di caratteri consentiti"
TC_AggiuntaDescrizioneNonValida	LN2.FN2.EXN2.LD2.FD1.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Formato di descrizione non valido!"
TC_AggiuntaPrezzoMancante	LN2.FN2.EXN2.LD2.FD2.LP1.FP2.PP2.LC2.FC2.LF2.FF2	"Prezzo - Campo obbligatorio!"
TC_AggiuntaPrezzoEccessiva	LN2.FN2.EXN2.LD2.FD2.LP3.FP2.PP2.LC2.FC2.LF2.FF2	"Prezzo supera il limite consentito!"
TC_AggiuntaPrezzoNonValido	LN2.FN2.EXN2.LD2.FD2.LP2.FP1.PP2.LC2.FC2.LF2.FF2	"Formato prezzo non valido!"
TC_AggiuntaPrezzoNegativo	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP1.LC2.FC2.LF2.FF2	"Il prezzo deve essere positivo!"
TC_AggiuntaCategoriaMancante	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC1.FC2.LF2.FF2	"Categoria - Campo obbligatorio!"
TC_AggiuntaCategoriaEccessiva	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC3.FC2.LF2.FF2	"Categoria supera il limite di caratteri consentiti!"
TC_AggiuntaCategoriaNonValida	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC1.LF2.FF2	"Formato categoria non valido!"

TC_AggiuntaFotoMancante	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF1.FF2	"Foto - Campo Obbligatorio!"
TC_AggiuntaFotoEccessiva	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF3.FF2	"Foto supera il limite di caratteri consentiti!"
TC_AggiuntaFotoNonValida	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF1	"Formato Foto non valida!"

TC_ModificaCappelloCatalogo

Parametro: Nome	
Categorie	Scelte
Lunghezza Nome- LN	1. Lunghezza >50 [property invalidLNValue] 2. Lunghezza <= 50 [property validLNValue]
Formato Nome- FN	1. Non rispetta il formato [property [invalidFNValue]] 2. Rispetta il formato [if validLNValue] [property validFNValue]
Esiste nel database- EXN	1. Esiste nel database [property EXN] 2. Non esiste nel database [if validFNValue] [property NotEXN]

Parametro: Descrizione	
Categorie	Scelte
Lunghezza Descrizione- LD	1. Lunghezza >150 [property invalidLDValue] 2. Lunghezza <= 150 [property validLDValue]
Formato Descrizione- FD	1. Non rispetta il formato [property [invalidFDValue]] 2. Rispetta il formato [if validLDValue] [property validFDValue]

Parametro: Prezzo	
Categorie	Scelte
Lunghezza Prezzo- LP	1. Lunghezza >10 [property invalidLPValue] 2. Lunghezza >=1 [property validLPValue]
Formato Prezzo- FP	1. Non rispetta il formato [property [invalidFPValue]] 2. Rispetta il formato [if validLPValue] [property validFPValue]
Positività Prezzo- PP	1. Il prezzo è negativo [property invalidPPValue] 2. Il prezzo è non negativo [if validFPValue] [property validPPValue]

Parametro: Categoria	
Categorie	Scelte
Lunghezza Categoria- LC	1. Lunghezza >50 [property invalidLCValue] 2. Lunghezza <= 50 [property validLCValue]
Formato Categoria- FC	1. Non rispetta il formato [property invalidFCValue] 2. Rispetta il formato [if validLCValue] [property validFCValue]

Parametro: Foto	
Categorie	Scelte
Lunghezza Foto- LF	1. Lunghezza >40 [property invalidLNValue] 2. Lunghezza <= 40 [property validLNValue]
Formato Foto- FF	1. Non rispetta il formato [property invalidFNValue] 2. Rispetta il formato [if validLNEValue] [property validFNValue]

Codice	Combinazioni	Esito
TC_ModificaCappelloCatalogoOk	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Modifica avvenuta con successo!"
TC_ModificaNomeInUso	LN2.FN2.EXN1.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Nome prodotto già esistente!"
TC_ModificaNomeNonValido	LN2.FN1.EXN1.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Formato nome non valido!"
TC_ModificaNomeEccessivo	LN3.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Nome supera il limite di caratteri consentiti"
TC_ModificaDescrizioneEccessiva	LN2.FN2.EXN2.LD3.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Descrizione supera il limite di caratteri consentiti"
TC_ModificaDescrizioneNonValida	LN2.FN2.EXN2.LD2.FD1.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Formato di descrizione non valido!"
TC_ModificaPrezzoEccessiva	LN2.FN2.EXN2.LD2.FD2.LP3.FP2.PP2.LC2.FC2.LF2.FF2	"Prezzo supera il limite consentito!"
TC_ModificaPrezzoNonValido	LN2.FN2.EXN2.LD2.FD2.LP2.FP1.PP2.LC2.FC2.LF2.FF2	"Formato prezzo non valido!"
TC_ModificaPrezzoNegativo	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP1.LC2.FC2.LF2.FF2	"Il prezzo deve essere positivo!"
TC_ModificaCategoriaEccessiva	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC3.FC2.LF2.FF2	"Categoria supera il limite di caratteri consentiti!"
TC_ModificaCategoriaNonValida	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF2	"Formato categoria non valido!"

	FC1.LF2.FF2	
TC_ModificaFotoEccessiva	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF3.FF2	"Foto supera il limite di caratteri consentiti!"
TC_ModificaFotoNonValida	LN2.FN2.EXN2.LD2.FD2.LP2.FP2.PP2.LC2.FC2.LF2.FF1	"Formato Foto non valida!"

TC_RicercaCappello

Parametro: Nome	
Categorie	Scelte
Lunghezza Nome- LN	1. Lunghezza = 0 - campo vuoto [property invalidLNValue] 2. Lunghezza <=50 [property validLNValue] 3. Lunghezza >50 [property invalidLNValue]
Formato Nome- FN	1. Non rispetta il formato [property [invalidFNValue]] 2. Rispetta il formato [<i>if</i> validLNValue] [property validFNValue]
Esistenza nel Database-EXID	1. Non esiste nel database [property NotEXI] 2. Esiste nel database [<i>if</i> validFNValue] [property EXI]

Codice	Combinazioni	Esito
TC_RicercaAvvenutaconSuccesso	LN2.FN2.EXID2	"Ricerca avvenuta con successo!"
TC_RicercaNomeMancante	LN1	"Nome - Campo obbligatorio!!"
TC_RicercaNomeEccessivo	LN3.FN2	"Numero di caratteri inseriti eccessivo"
TC_RicercaNomeNonValido	LN2.FN1	"Il formato inserito non è valido!"
TC_RicercaFallita	LI2.FI2.EXID1	"Prodotto non presente"

9.3 Test Gestione Carrello

TC_AggiungiProdotto

Parametro: Id_Cappello	
Categorie	Scelte
Presenza nel Carrello - PC	1. Prodotto già presente nel carrello 2. Prodotto non presente nel carrello

Parametro: Id_Utente	
Categorie	Scelte
Autenticazione - A	1. Utente non autenticato [property invalidAValue] 2. Utente autenticato [property validAValue]

Codice	Combinazioni	Esito
TC_AggiungiProdottoOK	PC2.A2	"Prodotto inserito correttamente"
TC_AggiungiProdottoIncrementa	PC1.A2	"Quantità di prodotto incrementata"
TC_AggiungiProdottoGuest	A1	"Utente deve essere autenticato"

TC_VisualizzaCarrello

Parametro: Carrello	
Categorie	Scelte
Quantità Prodotti - QP	1. Carrello vuoto 2. Carrello non vuoto

Parametro: Utente	
Categorie	Scelte
Autenticazione - A	1. Utente non autenticato [property invalidAValue] 2. Utente autenticato [property validAValue]

Codice	Combinazioni	Esito
TC_VisualizzaCarrelloOK	QP2.A2	“Carrello visualizzato correttamente”
TC_VisualizzaCarrelloVuoto	QP1.A2	“Carrello vuoto”
TC_VisualizzaCarrelloNonAutenticato	QP2.A1 - QP1.A1	“Utente deve essere autenticato”

9.4 Test Gestione Ordini

TC_RicercaOrdiniUtenti

Parametro: Nome	
Categorie	Scelte
Lunghezza Nome- LN	1. Lunghezza = 0 - campo vuoto [property invalidLNValue] 2. 0 < Lunghezza < =50 [property validLNValue] 3. Lunghezza >50 [property invalidLNValue]
Formato Nome- FN	1. Non rispetta il formato [property [invalidFNValue]] 2. Rispetta il formato [<i>if validLNValue</i>] [property validFNValue]
Esistenza nel Database-EXID	1. Non esiste nel database [property NotEXI] 2. Esiste nel database [<i>if validFNValue</i>] [property EXI]

Codice	Combinazioni	Esito
TC_RicercaAvvenutaconSuccesso	LN2.FN2.EXID2	“Ricerca avvenuta con successo!”
TC_RicercaNomeMancante	LN1	“Nome - Campo obbligatorio!!”
TC_RicercaNomeEccessivo	LN3.FN2	“Numero di caratteri inseriti eccessivo”
TC_RicercaNomeNonValido	LN2.FN1	“Il formato inserito non è valido!”
TC_RicercaFallita	LI2.FI2.EXID1	“Ordine non presente”

9.5 Test Gestione Dettaglio Ordini

TC_ModificaQuantitàProdottoNelCarrello

Parametro: Quantità	
Categorie	Scelte
Quantità- Q	1. Quantità indisponibile [property invalidQValue] 2. Quantità disponibile [property validQValue]

Codice	Combinazioni	Esito
TC_QuantitàDisponibile	Q2	"Quantità modificata con successo!"
TC_QuantitàIndisponibile	Q1	"Quantità non consentita"

9.6 Test Gestione Pagamenti

TC_AggiungiCarta

Parametro: Numero	
Categorie	Scelte
Lunghezza Numero - LI	1. Lunghezza = 0 – campo vuoto [property invalidLIValue] 2. $13 \leq \text{Lunghezza} \leq 16$ [property validLIValue] 3. $\text{Lunghezza} < 13, \text{Lunghezza} > 16$ [property invalidLIValue]
Formato Numero - FI	1. Non rispetta il formato [property invalidFIValue] 2. Rispetta il formato [<i>if</i> validLIValue] [property validFIValue]
Esistenza nel database - EXID	1. Esiste nel database [property EXID] 2. Non esiste nel database [<i>if</i> validFIValue] [property NotEXID]

Parametro: Proprietario	
Categorie	Scelte
Lunghezza Titolare - LT	1. Lunghezza = 0 – campo vuoto [property invalidLTValue] 2. $1 \leq \text{Lunghezza} \leq 40$ [property validLTValue] 3. $\text{Lunghezza} > 40$ [property invalidLTValue]
Formato Titolare – FT	1. Non rispetta il formato [property invalidFTValue] 2. Rispetta il formato [<i>if</i> validLTValue] [property validFTValue]

Parametro:CVV	
Categorie	Scelte
Lunghezza CVV - LCVV	1. Lunghezza 0 – campo vuoto [property invalidLCVVValue] 2. Lunghezza =3 [property validLCVVValue] 3. Lunghezza !=3 [property invalidLCVVValue]
Formato CVV – FCVV	1. Non rispetta il formato [property invalidCVVValue] 2. Rispetta il formato [if validLCVVValue] [property validFCVVValue]

Parametro: Data di scadenza	
Categorie	Scelte
Formato Data – FD	1. Non rispetta il formato [property invalidFDValue] 2. Rispetta il formato [property validFDValue]
Scadenza - S	1. Data non rispetta la scadenza [property InvalidSValue] 2. Data rispetta la scadenza [if validFDValue] [property validSValue]

Codice	Combinazione	Esito
TC_AggiungiCartaOk	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Carta aggiunta con successo!”
TC_CartaScaduta	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV2.FCVV2.FD2.S1	“Carta scaduta”
TC_FormatoDataNonValido	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV2.FCVV2.FD1	“Formato data non valido”
TC_LunghezzaCVVNonValida	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV3.FCVV2.FD2.S2	“CVV: errore lunghezza”
TC_FormatoCVVNonValido	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV2.FCVV1.FD2.S2	“Formato CVV non valido”
TC_CVVObbligatorio	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV1.FD2.S2	“CVV: campo obbligatorio”
TC_FormatoPropNonValido	LC2.LI2.FI2.EXID2.LT2.FT1.LCV2.FCVV2.FD2.S2	“Formato proprietario non valido”
TC_ProprietarioObbligatorio	LC2.LI2.FI2.EXID2.LT1.LCV2.FCVV2.FD2.S2	“Proprietario: campo obbligatorio”
TC_CartaGiàEsistente	LC2.LI2.FI2.EXID1.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Carta già esistente”

TC_FormatoCartaNonValido	LC2.LI2.FI1.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Formato numero carta non valido”
TC_NumeroCartaObbligatorio	LC2.LI1.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Numero carta: campo obbligatorio”
TC_LunghezzaNumeroNonValido	LC2.LI3.FI2.EXID2.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Numero carta: errore lunghezza”

TC_PagaOra

Parametro: Carrello	
Categorie	Scelte
Stato Carrello - SC	1. Carrello non contiene cappelli [property invalidSCValue] 2. Carrello contiene cappelli [property validSCValue]

Codice	Combinazione	Esito
TC_AcquistoCorretto	SC2	“Acquisto effettuato con successo!”
TC_CarrelloVuoto	SC1	“Carrello non contiene cappelli”