



UNIVERSITÀ DEGLI STUDI DI SALERNO

Corso di Laurea in Informatica, prof. A. De Lucia, a.a. 2021-22
Progetto di Ingegneria del Software

SHOODAN

[» GitHub Repository](#)

Scadenza: 07-10-2021

› Project Proposal

Scadenza: 14-10-2021

› Problem Statement

Scadenza: 28-10-2021

› Requisiti e casi d'uso

Scadenza: 11-11-2021

› Requisiti e casi d'uso

Scadenza: 25-11-2021

› System Design Document

Scadenza: 16-12-2021

➡ **Piano di test e specifica interfacce moduli**

Scadenza: alla consegna del progetto

› Altri documenti

› Esecuzione dei test

› Object Design

› Codice sorgente

› Report sul lavoro individuale

Shodan [IS-2021/2022-DE LUCIA]

Partecipanti del progetto

Nome	Matricola	E-mail
Antonio Gravino	05121 07161	a.gravino3@studenti.unisa.it
Dario Trinchese	05121 07479	d.trinchese2@studenti.unisa.it
Raffaele Zheng	05121 09015	r.zheng@studenti.unisa.it
Carmine Fabbri	05121 07353	c.fabbri@studenti.unisa.it
Carmine Napolitano	05121 06417	c.napolitano44@studenti.unisa.it

Questo documento tratta esclusivamente il **Test Plan** del progetto.

Per ulteriore documentazione valida ai fini dell'esame, consultare la repository *docs*.

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.

Relazioni con altri documenti

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

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

Gestione Account

- Registrazione utente
- Login
- Logout
- Modifica password
- Modifica e-mail

Gestione Blog

- Aggiungi un articolo
- Rimuovi un articolo

Gestione Game

- Aggiungi un titolo
- Rimuovi un titolo

- Aggiorna un titolo
- Ricerca un titolo

Gestione Payment

- Ricarica saldo
- Aggiungi carta
- Paga ora

Gestione Cart

- Aggiungi al carrello
- Rimuovi dal carrello
- Svuota carrello

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.

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:

- Web Server 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 GESTIONE ACCOUNT

TC_Registrazione

Parametro: Username	
Categorie	Scelte
Lunghezza Username - LU	1. Lunghezza = 0 – campo vuoto [property invalidLUValue] 2. Lunghezza >= 1 [property validLUValue]
Formato Username – FU	1. Non rispetta il formato [property invalidFUValue] 2. Rispetta il formato [<i>if validLUValue</i>] [property validFUValue]
Esistenza nel database - EXUS	1. Esiste nel database [<i>if validFUValue</i>] [property EXUS] 2. Non esiste nel database [property NotEXUS]

Parametro: Password	
Categorie	Scelte
Lunghezza Password – LP	1. Lunghezza = 0 – campo vuoto [property invalidLPValue] 2. Lunghezza >= 1 [property validLPValue]
Formato Password – FP	1. Non rispetta il formato [property invalidFPValue] 2. Rispetta il formato [<i>if validLPValue</i>] [property validFPValue]

Parametro: Ripeti password	
Categorie	Scelte
Lunghezza Ripeti Password – LRP	1. Lunghezza = 0 – campo vuoto [property invalidLRPValue] 2. Lunghezza >= 1 [property validLRPValue]
Coincidenza - OVLP	1. Il campo non coincide con il campo Password [property NotOverlap] 2. il campo coincide con il campo Password [<i>if validFRPValue && validLPValue</i>] [property Overlap]

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

CODICE	COMBINAZIONE	ESITO
TC_RegistrazioneOK	LU2.FU2.EXUS2.LP2.FP2.LRP2.OVLP2.LE2.FE2.EXEM2	“Registrazione avvenuta con successo!”
TC_RegistrazioneFailed1	LU2.FU2.EXUS2.LP2.FP2.LRP2.OVLP2.LE2.FE2.EXEM1	“E-mail già esistente”
TC_RegistrazioneFailed2	LU2.FU2.EXUS2.LP2.FP2.LRP2.OVLP2.LE2.FE1	“Formato e-mail non valido”
TC_RegistrazioneFailed3	LU2.FU2.EXUS2.LP2.FP2.LRP2.OVLP2.LE1	“E-mail - campo obbligatorio”

TC_RegistrazioneFailed4	LU2.FU2.EXUS2.LP2.FP2.LRP2.OVLP1.LE2.FE2.EXEM2	"Le password non coincidono"
TC_RegistrazioneFailed5	LU2.FU2.EXUS2.LP2.FP2.LRP1.LE2.FE2.EXEM2	"Ripeti password - campo obbligatorio"
TC_RegistrazioneFailed6	LU2.FU2.EXUS2.LP2.FP1.LRP2.OVLP2.LE2.FE2.EXEM2	"Formato password non valido"
TC_RegistrazioneFailed7	LU2.FU2.EXUS2.LP1.LRP2.OVLP1.LE2.FE2.EXEM2	"Password - campo obbligatorio"
TC_RegistrazioneFailed8	LU2.FU2.EXUS1.LP2.FP2.LRP2.OVLP2.LE2.FE2.EXEM2	"Username già esistente"
TC_RegistrazioneFailed10	LU2.FU1.LP2.FP2.LRP2.OVLP2.LE2.FE2.EXEM2	"Formato username non valido"
TC_RegistrazioneFailed11	LU1.LP2.FP2.LRP2.OVLP2.LE2.FE2.EXEM2	"Username - campo obbligatorio"

TC_Login

Parametro: Username	
Categorie	Scelte
Esistenza nel database - EXUS	1. Non esiste nel database [property NotEXUS] 2. Esiste nel database [property EXUS]

Parametro: Password	
Categorie	Scelte
Associazione all'utente - AS	1. Non è correttamente associata all'username [property invalidASValue] 2. È correttamente associata all'username [<i>if EXUS</i>] [property validASValue]

CODICE	COMBINAZIONE	ESITO
TC_LoginOK	EXUS2.AS2	"Login effettuato!"
TC_LoginFailed1	EXUS2.AS1	"Password errata"
TC_LoginFailed2	EXUS1	"Utente non esistente"

TC_ModificaPassword

Parametro: Password Attuale	
Categorie	Scelte
Associazione all'utente- AS	1. La password non è associata all'utente in sessione [property invalidASValue] 2. La password è associata all'utente in sessione [property validASValue]

Parametro: Nuova password	
Categorie	Scelte
Lunghezza Nuova Password – LNP	1. Lunghezza = 0 – campo vuoto [property invalidLNPValue] 2. Lunghezza >= 1 [property validLNPValue]

Formato Nuova Password- FNP	1. Non rispetta il formato [property invalidFNPValue] 2. Rispetta il formato [<i>if validLNPValue</i>] [property validFNPValue]
-----------------------------	--

Parametro: Ripeti password	
Categorie	Scelte
Lunghezza Rip. Password - LRP	1. Lunghezza = 0 – campo vuoto [property invalidLRPValue] 2. Lunghezza >= 1 [property validLRPValue]
Coincidenza - OVLP	1. Non coincide al campo Nuova password [property NotOverlap] 2. Coincide al campo Nuova password [<i>if validLRPValue && validFNPValue</i>] [property Overlap]

CODICE	COMBINAZIONE	ESITO
TC_ModificaPassOK	AS2.LNP2.FNP2.LRP2.OVLP2	“Password modificata con successo”
TC_ModificaPassFailed1	AS2.LNP2.FNP2.LRP2.OVLP1	“Le password non coincidono”
TC_ModificaPassFailed3	AS2.LNP2.FNP1	“Formato nuova password non valido”
TC_ModificaPassFailed4	AS2.LNP1	“Nuova password: campo obbligatorio”
TC_ModificaPassFailed5	AS1	“Password errata”
TC_ModificaPassFailed6	AS2.LNP2.FNP2.LRP1	“Ripeti password: campo obbligatorio”

TC_ModificaEmail

Parametro: E-mail corrente	
Categorie	Scelte
Associazione all’utente – AS	1. Non è associata all’utente in sessione [property invalidASValue] 2. È associata all’utente in sessione [property validASValue]

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

CODICE	COMBINAZIONE	ESITO
TC_ModificaEmailOK	AS2.LNE2.FNE2.EXEM2	“Email modificata con successo”
TC_ModificaEmailFailed1	AS2.LNE2.FNE2.EXEM1	“Email già in uso”
TC_ModificaEmailFailed2	AS2.LNE2.FNE1	“Formato nuova email non valido”
TC_ModificaEmailFailed3	AS2.LNE1	“Nuova email: campo obbligatorio”
TC_ModificaEmailFailed6	AS1.LNE2.FNE2.EXEM2	“Email errata”

TEST GESTIONE BLOG

TC_AGGIUNGI_ARTICOLO

Parametro: titoloArticolo	
Categorie	Scelte
Lunghezza Titolo - LT	1. Lunghezza = 0 – campo vuoto [property invalidLTValue] 2. Lunghezza >= 1 [property validLTValue]

Parametro: sottotitoloArticolo	
Categorie	Scelte
Lunghezza Sottotitolo - LS	1. Lunghezza = 0 – campo vuoto [property invalidLSValue] 2. Lunghezza >= 1 [property validLSValue]

Parametro: contenutoArticolo	
Categorie	Scelte
Lunghezza Contenuto - LC	1. Lunghezza = 0 – campo vuoto [property invalidLCValue] 2. Lunghezza >= 1 [property validLCValue]

CODICE	COMBINAZIONE	ESITO
TC_InserimentoArtOK	LT2.LS2.LC2	“Articolo aggiunto con successo!”
TC_InserimentoArtFailed1	LT2.LS2.LC1	“Contenuto: campo obbligatorio”
TC_InserimentoArtFailed2	LT2.LS1.LC2	“Sottitolo: campo obbligatorio”
TC_InserimentoArtFailed3	LT1.LS2.LC2	“Titolo: campo obbligatorio”

TC_RIMUOVI_ARTICOLO

Parametro: idArticolo	
Categorie	Scelte
Lunghezza ID - LI	1. Lunghezza = 0 – campo vuoto [property invalidLIValue] 2. Lunghezza >= 1 [property validLIValue]
Formato ID – FI	1. Non rispetta il formato [property invalidFIValue] 2. Rispetta il formato [<i>if validLIValue</i>] [property validLIValue]
Esiste nel database - EXUS	1. Non esiste nel database [property NotEXUS] 2. Esiste nel database [<i>if validFIValue</i>] [property EXUS]

CODICE	COMBINAZIONE	ESITO
TC_RimozioneArtOK	LI2.FI2.EXUS2	“Articolo rimosso con successo”
TC_RimozioneArtFailed1	LI1	“ID mancante”

TC_RimozioneArtFailed2	LI2.FI1	"Formato ID errato"
TC_RimozioneArtFailed3	LI2.FI2.EXUS1	"Articolo non esistente"

TEST GESTIONE GAME

TC_Aggiungi_titolo

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

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

Parametro: Data	
Categorie	Scelte
Formato Data - FD	1. Non rispetta il formato 2. Rispetta il formato [property validFDValue]
Intervallo Data - ID	1. Non rispetta l'intervallo [property invalidIDValue] 2. Rispetta l'intervallo [<i>if validFDValue</i>] [property validIDValue]

Parametro: Descrizione	
Categorie	Scelte
Lunghezza Descrizione – LDS	1. Lunghezza = 0 – campo vuoto [property invalidLDSValue] 2. Lunghezza >= 1 [property validLDSValue]

Parametro: Cover	
Categorie	Scelte
Lunghezza Cover – LC	1. Lunghezza = 0 – campo vuoto [property invalidLCValue]

	2. Lunghezza >= 1 [property validLCValue]
Formato Cover – FC	1. Non rispetta il formato [property invalidFCValue] 2. Rispetta il formato [<i>if validLCValue</i>] [property validFCValue]

Parametro: Sfondo	
Categorie	Scelte
Lunghezza Sfondo – LS	1. Lunghezza = 0 – campo vuoto [property invalidLSValue] 2. Lunghezza >= 1 [property validLSValue]
Formato Sfondo – FS	1. Non rispetta il formato [property invalidFSValue] 2. Rispetta il formato [<i>if validLSValue</i>] [property validFSValue]

CODICE	COMBINAZIONE	ESITO
TC_Aggiungi_titolo1	LN2.FN2.EXN2.LP2.FP2.PP2.FD2.ID2.LDS2.LC2.FC2.LS2.FS2	“Titolo aggiunto con successo!”
TC_Aggiungi_titolo2	LN2.FN2.EXN2.LP2.FP2.PP2.FD2.ID2.LDS2.LC2.FC2.LS2.FS1	“Formato sfondo non valido”
TC_Aggiungi_titolo3	LN2.FN2.EXN2.LP2.FP2.PP2.FD2.ID2.LDS2.LC2.FC2.LS1	“Sfondo: campo obbligatorio”
TC_Aggiungi_titolo4	LN2.FN2.EXN2.LP2.FP2.PP2.FD2.ID2.LDS2.LC2.FC1.LS2.FS2	“Formato cover non valido”
TC_Aggiungi_titolo5	LN2.FN2.EXN2.LP2.FP2.PP2.FD2.ID2.LDS2.LC1.LS2.FS2	“Cover: campo obbligatorio”
TC_Aggiungi_titolo6	LN2.FN2.EXN2.LP2.FP2.PP2.FD2.ID2.LDS1.LC2.FC2.LS2.FS2	“Descrizione: campo obbligatorio”
TC_Aggiungi_titolo7	LN2.FN2.EXN2.LP2.FP2.PP2.FD2.ID1.LDS2.LC2.FC2.LS2.FS2	“Data fuori intervallo”
TC_Aggiungi_titolo8	LN2.FN2.EXN2.LP2.FP2.PP2.FD1.LDS2.LC2.FC2.LS2.FS2	“Formato data non valido”
TC_Aggiungi_titolo9	LN2.FN2.EXN2.LP2.FP2.PP1.FD2.ID2.LDS2.LC2.FC2.LS2.FS2	“Prezzo negativo”
TC_Aggiungi_titolo10	LN2.FN2.EXN2.LP2.FP1.FD2.ID2.LDS2.LC2.FC2.LS2.FS2	“Formato prezzo non valido”
TC_Aggiungi_titolo11	LN2.FN2.EXN2.LP1.FD2.ID2.LDS2.LC2.FC2.LS2.FS2	“Prezzo: campo obbligatorio”
TC_Aggiungi_titolo12	LN2.FN2.EXN1.LP2.FP2.PP2.FD2.ID2.LDS2.LC2.FC2.LS2.FS2	“Titolo già esistente”
TC_Aggiungi_titolo13	LN2.FN1.LP2.FP2.PP2.FD2.ID2.LDS2.LC2.FC2.LS2.FS2	“Formato titolo non valido”
TC_Aggiungi_titolo14	LN1.LP2.FP2.PP2.FD2.ID2.LDS2.LC2.FC2.LS2.FS2	“Titolo: campo obbligatorio”

TC_Rimuovi_titolo

Parametro: ID	
Categorie	Scelte
Lunghezza ID - LI	1. Lunghezza = 0 – campo vuoto [property invalidLIValue] 2. Lunghezza >= 1 [property validLIValue]
Formato ID – FI	1. Non rispetta il formato [property invalidFIValue] 2. Rispetta il formato [<i>if validLNValue</i>] [property validFIValue]
Esistenza nel database - EXID	1. Non esiste nel database [property NotEXI] 2. Esiste nel database [<i>if validFIValue</i>] [property EXI]

CODICE	COMBINAZIONE	ESITO
TC_Rimuovi_titolo1	LI2.FI2.EXID2	“Titolo rimosso con successo!”
TC_Rimuovi_titolo2	LI2.FI2.EXID1	“Il titolo non è presente”
TC_Rimuovi_titolo3	LI2.FI1	“Formato ID errato”
TC_Rimuovi_titolo4	LI1	“ID: campo obbligatorio”

TC_Aggiorna_titolo

Parametro: ID	
Categorie	Scelte
Esistenza nel database - EXID	1. Non esiste nel database [property NotEXI] 2. Esiste nel database [property EXI]

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

CODICE	COMBINAZIONE	ESITO
TC_Aggiungi_titolo1	EXID2.LP2.FP2.PP2	"Prezzo del titolo aggiornato con successo!"
TC_Aggiungi_titolo2	EXID2.LP2.FP2.PP1	"Prezzo negativo"
TC_Aggiungi_titolo3	EXID2.LP2.FP1	"Formato prezzo non valido"
TC_Aggiungi_titolo4	EXID2.LP1	"Prezzo: campo obbligatorio"
TC_Aggiungi_titolo6	EXID1.LP2.FP2.PP2	"Titolo non esistente"
TC_Aggiungi_titolo7	EXID1.LP2.FP1	"Titolo non esistente"

TEST GESTIONE PAYMENT

TC_AggiungiCarta

Parametro: Circuito	
Categorie	Scelte
Lunghezza Circuito - LC	1. Lunghezza = 0 – campo vuoto [property invalidLCValue] 2. Lunghezza >= 1 [property validLCValue]

Parametro: Numero Carta	
Categorie	Scelte
Lunghezza Numero - LI	1. Lunghezza = 0 – campo vuoto [property invalidLIValue] 2. Lunghezza >= 1 [property validLIValue]
Formato Numero - FI	1. Non rispetta il formato [property invalidFIValue] 2. Rispetta il formato [<i>if validLIValue</i>] [property validFIValue]
Esistenza nel database - EXID	1. Non esiste nel database [property NotEXID] 2. Esiste nel database [<i>if validFIValue</i>] [property EXID]

Parametro: Titolare	
Categorie	Scelte
Lunghezza Titolare - LT	1. Lunghezza = 0 – campo vuoto [property invalidLTValue] 2. Lunghezza >= 1 [property validLTValue]
Formato Titolare – FT	1. Non rispetta il formato [property invalidFTValue] 2. Rispetta il formato [<i>if validLTValue</i>] [property validFTValue]

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

Parametro: 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 [<i>if validFDValue</i>] [property validSValue]

CODICE	COMBINAZIONE	ESITO
TC_AggiungiCartaOK	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Carta aggiunta con successo!”
TC_AggiungiCartaFailed1	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV2.FCVV2.FD2.S1	“Carta scaduta”
TC_AggiungiCartaFailed2	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV2.FCVV2.FD1	“Formato data non valido”
TC_AggiungiCartaFailed3	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV2.FCVV1.FD2.S2	“Formato CVV non valido”
TC_AggiungiCartaFailed4	LC2.LI2.FI2.EXID2.LT2.FT2.LCVV1.FD2.S2	“CVV: campo obbligatorio”
TC_AggiungiCartaFailed5	LC2.LI2.FI2.EXID2.LT2.FT1.LCV2.FCVV2.FD2.S2	“Formato titolare non valido”
TC_AggiungiCartaFailed6	LC2.LI2.FI2.EXID2.LT1.LCV2.FCVV2.FD2.S2	“Titolare: campo obbligatorio”
TC_AggiungiCartaFailed7	LC2.LI2.FI2.EXID1.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Carta già esistente”
TC_AggiungiCartaFailed8	LC2.LI2.FI1.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Formato numero carta non valido”
TC_AggiungiCartaFailed9	LC2.LI1.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Numero carta: campo obbligatorio”
TC_AggiungiCartaFailed10	LC1.LI2.FI2.EXID2.LT2.FT2.LCVV2.FCVV2.FD2.S2	“Circuito: campo obbligatorio”

TC_RicaricaSaldo

Parametro: Quantità	
Categorie	Scelte
Formato Quantità – FQ	1. Non rispetta il formato [property invalidFQValue] 2. Rispetta il formato [property validFQValue]
Quantità - Q	1. Quantità <= 0 [property invalidQValue] 2. Quantità > 0 [<i>if validFQValue</i>] [property validQValue]

Parametro: Carta	
Categorie	Scelte
Esiste Carta - EXC	1. Non esiste carta associata [property NotEXC] 2. Esiste carta associata [property EXC]
Carta Scaduta - CS	1. Carta scaduta [property CS] 2. Carta valida [<i>if validSCValue</i>] [property NotCS]

CODICE	COMBINAZIONE	ESITO
TC_RicaricaSaldoOk	FQ2.Q2.EXC2.CS2	“Ricarica effettuata!”
TC_RicaricaSaldoFailed2	FQ2.Q2.EXC2.CS1	“Carta scaduta”
TC_RicaricaSaldoFailed3	FQ2.Q2.EXC1	“Carta non esistente”
TC_RicaricaSaldoFailed4	FQ2.Q1.EXC2.CS2	“Quantità non valida”
TC_RicaricaSaldoFailed5	FQ1.EXC2.CS2	“Formato quantità non valido”

TC_PagaOra

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

Parametro: Saldo	
Categorie	Scelte
Saldo - S	1. Saldo insufficiente [property invalidSValue] 2. Saldo sufficiente [<i>if validSCValue</i>] [property validSValue]

CODICE	COMBINAZIONE	ESITO
TC_PagaOraOk	SC2.S2	“Acquisto effettuato con successo!”
TC_PagaOraFailed1	SC2.S1	“Saldo insufficiente”
TC_PagaOraFailed2	SC1.S2	“Carrello non contiene giochi”
TC_PagaOraFailed3	SC1.S1	“Carrello non contiene giochi”