



UNIVERSITÀ DEGLI STUDI DI SALERNO

*Corso di Laurea in Informatica, prof. A. De Lucia,  
a.a 2021/2022  
Progetto di Ingegneria del Software*



*Test Plan*

Partecipanti	Matricola
Marta Coiro	0512108154
Katia Buonocore	0512106528
Rita Cuccaro	0512109495

## Revision History

Data	Versione	Descrizione	Autore
16/11/2021	1.0	Prima stesura del documento (Problem <u>Statement</u> )	Membri del team
30/11/2021	1.0	Requirement Analysis Document	Membri del team
06/12/2021	1.0	System Design Document	Membri del team
20/12/2021	1.0	Gestione Dati Persistenti_MusicConsole	Membri del team
27/12/2021	1.0	Object Design Document	Membri del team
15/01/2022	1.0	Test Plan	Membri del team
28/01/2022	1.0	Test Execution Report	Membri del team
18/02/2022	1.0	Test Summary Report	Membri del team

# **Indice**

## **1. INTRODUZIONE**

## **2. DOCUMENTI CORRELATI**

2.1 Relazioni con il documento di analisi dei requisiti (RAD).

2.2 Relazioni con il System Design Document (SDD).

2.3 Relazioni con l'Object Design Document (ODD).

## **3. PANORAMICA DEL SISTEMA.**

## **4.FUNZIONALITA' DA TESTARE.**

## **5. CRITERI PASS/FAIL.**

## **6. APPROCCIO**

6.1 Test di unità

6.2 Test di integrazione

6.3 Test di sistema

## **7. SOSPENSIONE E RIPRESA**

7.1 Criteri di sospensione

7.2 Criteri di ripresa

## **8. MATERIALE PER IL TESTING**

## **9. TEST CASES**

### **9.1 Gestione Account**

9.1.1 Registrazione

9.1.2 Login Utente/Amministratore

9.1.3 ModificaAccount

### **9.2 Gestione Acquisti**

9.2.1 EmissioneOrdine

### 9.3 Gestione Carrello

9.3.1 AggiuntaProdottiCarrello

9.3.2 EliminazioneProdottoCarrello

9.3.3 Checkout

### 9.4 Gestione Prodotto

9.4.1 ModificaAlbum

9.4.2 ModificaBrano

9.4.3 ModificaPodcast

9.4.4 CreazionePlaylist

9.4.5 AggiuntaBraniAllaPlaylist

9.4.6 RifornimentoProdotti

9.4.7 CercaProdotti

# 1. INTRODUZIONE

Lo scopo di questo documento è quello di andare ad analizzare e gestire lo sviluppo e le attività di testing che riguardano il nostro sistema ***MusicConsole***.

In particolar modo in questa sessione di lavoro, l'obiettivo sarà quello di verificare il corretto funzionamento del sistema sviluppato in diversi casi. Questi casi sono stati studiati per mettere alla prova le varie funzionalità del sistema ed effettuando questi test case saremo in grado di rivelare eventuali bug, errori e/o incongruenze tra il comportamento desiderato e quello osservato.

I risultati ci permetteranno di capire dove bisognerà correggere eventuali errori o apportare modifiche per migliorare i vari sottosistemi.

## 2. DOCUMENTI CORRELATI

Il Test Plan ha una stretta relazione con i documenti prodotti in precedenza in quanto il sistema è stato pianificato precedentemente nelle relative documentazioni.

Nella fase di testing, si verificherà se le aspettative descritte nei precedenti documenti siano state rispettate.

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

La relazione tra Test Plan e RAD riguarda i requisiti funzionali e non funzionali del sistema, poiché i test che verranno svolti, saranno eseguiti su quelle funzionalità tenendo conto delle specifiche espresse nel precedente documento.

### ***2.2 RELAZIONI CON IL SYSTEM DESIGN DOCUMENT(SDD)***

La relazione tra Test Plan e SDD riguarda la suddivisione dell'architettura del sistema sviluppato in tre layer e quindi il test deve tener conto di queste suddivisioni.

### ***2.3 RELAZIONI CON L'OBJECT DESIGN DOCUMENT (ODD)***

L'ODD contiene i package e le class interfaces del sistema che vanno prese in considerazione durante le operazioni di testing.

### **3. PANORAMICA DEL SISTEMA**

Come riportato nel SDD la struttura del nostro sistema è suddivisa secondo un'architettura three-tier: Presentation Layer, Application Logic Layer e Data Layer.

Inoltre, il sistema è suddiviso in sottosistemi ed è stato diviso in gestioni quali:

- Gestione Account
- Gestione Acquisti
- Gestione Carrello
- Gestione Prodotto

### **4. FUNZIONALITÀ DA TESTARE**

Le funzionalità che saranno sottoposte a test sono:

- Gestione Account
  1. Registrazione
  2. Login Utente/Amministratore
  3. Modifica Account
- Gestione Acquisti
  1. Emissione Ordine
- Gestione Carrello
  1. AggiuntaProdottiCarrello
  2. EliminazioneProdottoCarrello
  3. Checkout
- Gestione Prodotti
  1. Modifica Album
  2. Modifica Brano
  3. Modifica Podcast
  4. Creazione Playlist
  5. Aggiungi Brani Alla Playlist
  6. Rifornimento Prodotti
  7. Cerca Prodotti

## **5. CRITERI PASS/FAIL**

I dati in input saranno raggruppati in base a caratteristiche comuni in modo da effettuare un test su un unico elemento rappresentativo.

Il testing ha successo se l'output osservato è diverso dall'output atteso cioè la fase di testing avrà successo se individuerà una failure. Verrà quindi analizzata la failure e, se legata ad un fault, si procederà alla sua correzione.

Verrà, infine, iterata la fase di testing per verificare che la modifica apportata non abbia alcun impatto sulle altre componenti del sistema.

## **6. APPROCCIO**

L'approccio della fase di testing si compone di tre fasi. Si inizia con il test di Unità, si procede con il test di Integrazione e si termina con il test di Sistema.

### **6.1 TESTING DI UNITÀ**

Il testing di Unità ha lo scopo di testare singolarmente le componenti del sistema.

Per questo testing si utilizzerà la tecnica "BLACK-BOX". Con il "BLACK-BOX" testing ci focalizzeremo sul comportamento dell'input/output delle singole componenti senza tenere 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 partizionarle in classi di equivalenza chiamate "scelte". Al termine saranno specificate le "combinazioni" delle scelte da testare e determinare i risultati corrispondenti. Mediante il "Category Partition" otterremo, quindi, un test efficiente e privo di ridondanze.

### **6.2 TESTING DI INTEGRAZIONE**

Il testing di Integrazione si occupa di combinare le componenti e testarle come un unico gruppo. Per questa tipologia di testing andremo ad utilizzare la tecnica "BOTTOM-UP", ritenuta più adatta per i software basati sul paradigma Object Oriented. Si farà utilizzo del framework JUnit per la definizione dei test case in Java.

### **6.3 TEST DI SISTEMA**

L'obiettivo del test di Sistema è quello di dimostrare che tutti i requisiti richiesti sono stati soddisfatti. Tutte le funzionalità verranno testate.

Trattandosi di un sistema web-based verrà utilizzato il tool di Selenium, il quale si occuperà di simulare l'interazione con il sistema dal punto di vista dell'utente.

## **7. SOSPENSIONE E RIPRESA**

### **7.1 CRITERI DI SOSPENSIONE**

La fase di testing del sistema verrà sospesa quando si otterranno i risultati attesi.

### **7.2 CRITERI DI RIPRESA**

La fase di testing potrà riprendere dopo che ci sono state modifiche o correzioni apportate al sistema, che generano errori o fallimenti.

I test case verranno nuovamente sottoposti al sistema in modo tale da assicurarsi di aver risolto il problema effettivamente.

## **8. MATERIALE PER IL TESTING**

Gli strumenti necessari per l'attività di testing sono:

- Computer sul quale è installato un browser e con una connessione ad Internet
- Selenium



## 9. TEST CASE

### 9.1 Gestione Account

#### 9.1.1 Registrazione

Parametro: <b>Nome</b> Formato: [A-Za-z]	
Lunghezza [LN]	1. Lunghezza=0 o campo vuoto [property invalidLN] 2. Lunghezza>=1 [property validLN]
Formato [FN]	1. Non rispecchia il formato [property invalidFN] 2. Rispecchia il formato [if validLN] [property validFN]

Parametro: <b>Cognome</b> Formato: [A-Za-z]	
Lunghezza [LC]	1. Lunghezza=0 o campo vuoto [property invalidLC] 2. Lunghezza>=1 [property validLC]
Formato [FC]	1. Non rispecchia il formato [property invalidFC] 2. Rispecchia il formato [if validLC] [property validFC]

Parametro: <b>Numero di telefono</b> Formato: [0-9{10}]	
Lunghezza [LNT]	1. Lunghezza=0 o campo vuoto [property invalidLNT] 2. Lunghezza=10 [property validLNT] 2.1 Lunghezza>10 [property invalidLNT]
Formato [FNT]	1. Non rispecchia il formato [property invalidFNT] 2. Rispecchia il formato [if validLNT] [property validFNT]

Parametro: <b>Email</b> Formato: [A-Za-z09._%+-]@ [A-Za-z] . [A-Za-z]	
Lunghezza [LE]	1. Lunghezza=0 o campo vuoto [property invalidLE] 2. Lunghezza>=1 [property validLE]
Formato [FE]	1. Non rispecchia il [property invalidFE] 2. Rispecchia il formato [if validLE] [property validFE]

Parametro: <b>Username</b>	
Lunghezza [LU]	1. Lunghezza=0 o campo vuoto [property invalidLU] 2. Lunghezza>=1 [property validLU]
Presente [PU]	1. L'username inserito è già presente nel database [property invalidPU] 2. L'username inserito non è presente nel database [if validFU] [property validPU]

Parametro: <b>Password</b>	
Lunghezza [LP]	1. Lunghezza=0 o campo vuoto o Lunghezza>8 [property invalidLP] 2. 1<= Lunghezza <=8 [property validLP]

Codice	Combinazione	Esito
TC_1.1.1	LN2, FN2, LC2, FC2, LNT2, FNT2, LE2, FE2, LU2, PU2, LP2	Registrazione avvenuta con successo
TC_1.1.2	LN2, FN2, LC2, FC2, LNT2, FNT2, LE2, FE2, LU2, PU2, LP1	Campo password vuoto o lunghezza>8
TC_1.1.3	LN2, FN2, LC2, FC2, LNT2, FNT2, LE2, FE2, LU2, PU1, LP2	Username già presente
TC_1.1.4	LN2, FN2, LC2, FC2, LNT2, FNT2, LE2, FE2, LU1, PU2, LP2	Campo username vuoto

TC_1.1.5	LN2, FN2, LC2, FC2, LNT2, FNT2, LE2, FE1, LU2, PU2, LP2	Formato email non valido
TC_1.1.6	LN2, FN2, LC2, FC2, LNT2, FNT2, LE1, FE2, LU2, PU2, LP2	Campo email vuoto
TC_1.1.7	LN2, FN2, LC2, FC2, LNT2, FNT1, LE2, FE2, LU2, PU2, LP2	Formato numero di telefono non valido
TC_1.1.8	LN2, FN2, LC2, FC2, LNT1, FNT2, LE2, FE2, LU2, PU2, LP2	Campo numero di telefono vuoto
TC_1.1.9	LN2, FN2, LC2, FC1, LNT2, FNT2, LE2, FE2, LU2, PU2, LP2	Formato cognome non valido
TC_1.1.10	LN2, FN2, LC1, FC2, LNT2, FNT2, LE2, FE2, LU2, PU2, LP2	Campo cognome vuoto
TC_1.1.11	LN2, FN1, LC2, FC2, LNT2, FNT2, LE2, FE2, LU2, PU2, LP2	Formato nome non valido
TC_1.1.12	LN1, FN2, LC2, FC2, LNT2, FNT2, LE2, FE2, LU2, PU2, LP2	Campo nome vuoto

### 9.1.2 Login Utente/Amministratore

<b>Parametro: Username</b>	
Lunghezza [LU]	1. Lunghezza=0 o campo vuoto [property invalidLU] 2. Lunghezza>=1 [property validLU]
Presente [PU]	1. Username presente nel database [if validLU] [property existPU] 2. Username non presente nel database [property notExistPU]

<b>Parametro: Password</b>	
Lunghezza [LP]	1. Lunghezza=0 o campo vuoto o Lunghezza>8 [property invalidLP] 2. 1<= Lunghezza <=8 [property validLP]

Combacia [CP]	1. Password non associata all'username [property invalidCP] 2. Password associata all'username [if validLP] [property validCP]
---------------	---

Codice	Combinazione	Esito
TC_1.2.1	CP2, LP2, PU2, LU2	Login effettuato
TC_1.2.2	CP2, LP2, PU2, LU1	Campo username vuoto
TC_1.2.3	CP2, LP1, PU2, LU2	Campo password vuoto
TC_1.2.4	LP1, LU2	Campi username e password vuoti

### 9.1.3 Modifica Account

Parametro: <b>Città</b>	
Modificata [MCI]	1. La città non è stata modificata [property validMCI] 2. La città è stata modificata [property validMCI]

Codice	Combinazione	Esito
TC_1.3.1	MCI2	Città modificata
TC_1.3.2	MCI1	Città non modificata

## **9.2 Gestione Acquisti**

### 9.2.1 Emissione Ordine

Parametro: <b>Data Spedizione</b>	
Disponibile [DDS]	1. La data di spedizione non è disponibile [property invalidDDS] 2. La data di spedizione è disponibile [property validDDS]

Codice	Combinazione	Esito
TC_2.1.1	DDS2	Ordine spedito con successo
TC_2.1.2	DDS1	Data di spedizione non disponibile

## 9.3 Gestione Carrello

### 9.3.1 AggiuntaProdottiCarrello

Parametro: <b>Prodotti</b>	
Aggiunto [AAPC]	1. Prodotto aggiunto nel carrello [property validAAPC] 2. Prodotto già presente nel carrello [property invalidAAPC]

Codice	Combinazione	Esito
TC_3.3.1	AAPC2	Prodotto non aggiunto al carrello in quanto già presente
TC_3.3.2	AAPC1	Prodotto aggiunto al carrello

### 9.3.2 EliminazioneProdottoCarrello

Parametro: <b>Prodotti</b>	
Eliminato [EAPC]	1. Prodotto eliminato dal carrello [property validEAPC] 2. Prodotto non eliminato dal carrello [property invalidEAPC]

Codice	Combinazione	Esito
TC_3.2.1	EAPC2	Prodotto non eliminato dal carrello
TC_3.2.2	AAPC1	Prodotto eliminato dal carrello

### 9.3.3 Checkout

Parametro: <b>Nome Carta</b> Formato: [A-Za-z]	
Lunghezza [LNC]	1. Lunghezza=0 o campo vuoto [property invalidLNC] 2. Lunghezza>=1 [property validLNC]
Formato [FNC]	1. Non rispetta il formato [property invalidFNC]

	2. Rispecchia il formato [if validLNC] [property validFNC]
--	---

Parametro: <b>Cognome Carta</b> Formato: [A-Za-z]	
Lunghezza [LCC]	1. Lunghezza=0 o campo vuoto [property invalidLCC] 2. Lunghezza>=1 [property validLCC]
Formato [FCC]	1. Non rispecchia il formato [property invalidFCC] 2. Rispecchia il formato [if validLCC] [property validFCC]

Parametro: <b>Numero carta di credito</b> Formato: ([0-9]{4})-([0-9]{4})-([0-9]{4})-([0-9]{4})	
Lunghezza [LNCC]	1. Lunghezza=0 o campo vuoto [property invalidLNCC] 2. Lunghezza>=1 [property validLNCC]
Formato [FNCC]	1. Non rispecchia il formato [property invalidFNCC] 2. Rispecchia il formato [if validLNCC] [property validFNCC]

Parametro: <b>Data Scadenza Carta</b>	
Lunghezza [LDSC]	1. Lunghezza=0 o campo vuoto [property invalidLDSC] 2. Lunghezza>=1 [property validLDSC]
Disponibile [DDSC]	1. La data di scadenza della carta non è disponibile [property invalidDDSC] 2. La data di scadenza della carta è disponibile [if property validLDSC] [property validDDSC]

Parametro: <b>CVV</b> Formato: [0-9 {3}]	
Lunghezza [LCVV]	1. Lunghezza=0 o campo vuoto [property invalidLCVV] 2. Lunghezza>=1 [property validLCVV]

Formato [FCVV]	1. Non rispecchia il formato [property invalidFCVV] 2. Rispecchia il formato [if validLCVV] [property validFCVV]
Presente [PCVV]	1. Codice della carta già presente nel database [if validFCVV] [property existPCVV] 2. Codice della carta non presente nel database [property notExistPCVV]

Codice	Combinazione	Esito
TC_3.3.1	LNC2, FNC2, LCC2, FCC2, LNCC2, FNCC2, LDSC2, DDSC2, LCVV2, FCVV2, PCVV2	Ordine inviato
TC_3.3.2	LNC2, FNC2, LCC2, FCC2, LNCC2, FNCC2, LDSC2, DDSC2, LCVV2, FCVV2, PCVV1	CVV già presente nel database
TC_3.3.3	LNC2, FNC2, LCC2, FCC2, LNCC2, FNCC2, LDSC2, DDSC2, LCVV2, FCVV1, PCVV2	Formato CVV non valido
TC_3.3.4	LNC2, FNC2, LCC2, FCC2, LNCC2, FNCC2, LDSC2, DDSC2, LCVV1, FCVV2, PCVV2	Campo CVV vuoto
TC_3.3.5	LNC2, FNC2, LCC2, FCC2, LNCC2, FNCC2, LDSC2, DDSC1, LCVV2, FCVV2, PCVV2	Data scadenza carta non disponibile
TC_3.3.6	LNC2, FNC2, LCC2, FCC2, LNCC2, FNCC2, LDSC1, DDSC2, LCVV2, FCVV2, PCVV2	Campo data scadenza carta vuoto
TC_3.3.7	LNC2, FNC2, LCC2, FCC2, LNCC2, FNCC1,	Formato numero carta non valido

	LDSC2, DDSC2, LCVV2, FCVV2, PCVV2	
TC_3.3.8	LNC2, FNC2, LCC2, FCC2, LNCC1, FNCC2, LDSC2, DDSC2, LCVV2, FCVV2, PCVV2	Campo numero carta vuoto
TC_3.3.9	LNC2, FNC2, LCC2, FCC1, LNCC2, FNCC2, LDSC2, DDSC2, LCVV2, FCVV2, PCVV2	Formato cognome carta non valido
TC_3.3.10	LNC2, FNC2, LCC1, FCC2, LNCC2, FNCC2, LDSC2, DDSC2, LCVV2, FCVV2, PCVV2	Campo cognome carta vuoto
TC_3.3.11	LNC2, FNC1, LCC2, FCC2, LNCC2, FNCC2, LDSC2, DDSC2, LCVV2, FCVV2, PCVV2	Formato nome carta non valido
TC_3.3.12	LNC1, FNC2, LCC2, FCC2, LNCC2, FNCC2, LDSC2, DDSC2, LCVV2, FCVV2, PCVV2	Campo nome carta vuoto



## 9.4 Gestione Prodotti

### 9.4.1 Modifica Album

Parametro: <b>Nome Album</b>	
Lunghezza [LNA]	1. Lunghezza=0 o campo vuoto [property invalidLNA] 2. Lunghezza>=1 [property validLNA]

Parametro: <b>Nome Artista</b>	
Lunghezza [LNAR]	1. Lunghezza=0 o campo vuoto [property invalidLNAR] 2. Lunghezza>=1 [property validLNAR]

Parametro: <b>Tipo</b>	
Modificato [MT]	1. Il tipo non è stato modificato [property notModifiedMT] 2. Il tipo è stato modificato [property modifiedMT]

✚ È possibile passare da un tipo album -> tipo brano e tipo album-> tipo podcast e viceversa. (Si rimanda a “Modifica Brano” e “Modifica Podcast”)

Parametro: <b>Prezzo Streaming, Vinile, CD</b>	
Formato: [0-9{2}.0-9{2}]	
Lunghezza [LP]	1. Lunghezza=0 o campo vuoto [property invalidLP] 2. Lunghezza>=1 [property validLP]
Formato [FP]	1. Non rispecchia il formato [property invalidFP] 2. Rispecchia il formato [if validLP] [property validFP]

Codice	Combinazione	Esito
TC_4.1.1	LNA2, LNAR2, MT1, LP2, FP2	L'album è stato modificato correttamente
TC_4.1.2	LNA2, LNAR2, MT1, LP2, FP1	Formato prezzi non valido
TC_4.1.3	LNA2, LNAR2, MT2, LP2, FP2	È stato modificato il tipo dell'album (si riporti alla

		modifica del corrispondente tipo selezionato)
--	--	---

#### 9.4.2 Modifica Brano

Parametro: <b>Nome Brano</b>	
Lunghezza [LNB]	1. Lunghezza=0 o campo vuoto [property invalidLNB] 2. Lunghezza>=1 [property validLNB]

Parametro: <b>Nome Artista</b>	
Lunghezza [LNAR]	1. Lunghezza=0 o campo vuoto [property invalidLNAR] 2. Lunghezza>=1 [property validLNAR]

Parametro: <b>Tipo</b>	
Modificato [MT]	1. Il tipo non è stato modificato [property notModifiedMT] 2. Il tipo è stato modificato [property modifiedMT]

✚ È possibile passare da un tipo brano -> tipo album e tipo brano-> tipo podcast e viceversa. (Si rimanda a “Modifica Album” e “Modifica Podcast”)

Parametro: <b>Prezzo Brano</b>	
Formato: [0-9{2}.0-9{2}]	
Lunghezza [LPB]	1. Lunghezza=0 o campo vuoto [property invalidLPB] 2. Lunghezza>=1 [property validLPB]
Formato [FPB]	1. Non rispecchia il formato [property invalidFPB] 2. Rispecchia il formato [if validLPB] [property validFPB]

Parametro: <b>Genere</b>	
--------------------------	--

Lunghezza [LG]	1. Lunghezza=0 o campo vuoto [property invalidLG] 2. Lunghezza>=1 [property validLG]
Disponibile [DG]	1. Tipo di genere non disponibile tra quelli presenti sul sito [property invalidDG] 2. Tipo di genere disponibile tra quelli presenti sul sito [if validLG] [property validDG]

Codice	Combinazione	Esito
TC_4.2.1	LNB2, LNB2, MT1, LPB2, FPB2, LG2, DG2	Brano modificato con successo
TC_4.2.2	LNB2, LNB2, MT1, LPB2, FPB2, LG2, DG1	Tipo di genere non disponibile
TC_4.2.3	LNB2, LNB2, MT1, LPB2, FPB2, LG1, DG2	Campo genere vuoto
TC_4.2.4	LNB2, LNB2, MT1, LPB2, FPB1, LG2, DG2	Formato prezzo brano non valido
TC_4.2.5	LNB2, LNB2, MT1, LPB1, FPB2, LG2, DG2	Campo prezzo brano vuoto
TC_4.2.6	LNB2, LNB2, MT2, LPB2, FPB2, LG2, DG2	Il tipo è stato modificato (si riporti alla modifica del corrispondente tipo selezionato)
TC_4.2.7	LNB2, LNB1, MT1, LPB2, FPB2, LG2, DG2	Campo nome artista vuoto
TC_4.2.8	LNB1, LNB2, MT1, LPB2, FPB2, LG2, DG2	Campo nome brano vuoto

#### 9.4.3 Modifica Podcast

Parametro: <b>Ideatore</b>	
Lunghezza [LI]	1. Lunghezza=0 o campo vuoto [property invalidLI] 2. Lunghezza>=1 [property validLI]

Parametro: <b>Tipo</b>	
Modificato [MT]	1. Il tipo non è stato modificato [property notModifiedMT] 2. Il tipo è stato modificato [property modifiedMT]

- ✚ È possibile passare da un tipo podcast -> tipo album e tipo podcast-> tipo brano e viceversa. (Si rimanda a “Modifica Album” e “Modifica Brano”)

Parametro: <b>Prezzo Podcast</b>	
Lunghezza [LPP]	1. Lunghezza=0 o campo vuoto [property invalidLPP] 2. Lunghezza>=1 [property validLPP]
Formato [FPP]	1. Non rispecchia il formato [property invalidFPP] 2. Rispecchia il formato [if validLPP] [property validFPP]

Codice	Combinazione	Esito
TC_4.3.1	LI2, MT1, LPP2, FPP2	Podcast modificato con successo
TC_4.3.2	LI2, MT1, LPP2, FPP1	Formato prezzo podcast non valido
TC_4.3.3	LI2, MT1, LPP1, FPP2	Campo prezzo podcast vuoto
TC_4.3.4	LI2, MT2, LPP2, FPP2	Il tipo è stato modificato (si riporti alla modifica del corrispondente tipo selezionato)
TC_4.3.5	LI1, MT1, LPP2, FPP2	Campo ideatore obbligatorio

#### 9.4.4 Creazione Playlist

Parametro: <b>Nome Playlist</b>	
Lunghezza [LNP]	1. Lunghezza=0 o campo vuoto [property invalidLNP] 2. Lunghezza>=1 [property validLNP]
Esiste [ENP]	1. Esiste già una playlist con questo nome [property existENP] 2. Non esiste una playlist con questo nome [if validLNP] [property notExistENP]

Codice	Combinazione	Esito
--------	--------------	-------

TC_4.4.1	LNP2, ENP2	Playlist creata con successo
TC_4.4.2	LNP2, ENP1	Esiste già una playlist con questo nome
TC_4.4.3	LNP1, ENP2	Campo nome playlist obbligatorio

#### 9.4.5 Aggiunta Brani Alla Playlist

Parametro: <b>Aggiunta Brano Playlist</b>	
Presente [PABP]	1. Brano già presente nella playlist [property invalidPABP] 2. Brano non presente nella playlist [property validPABP]

Codice	Combinazione	Esito
TC_4.5.1	PABP2	Brano aggiunto alla playlist
TC_4.5.2	PABP1	Brano già presente nella playlist

#### 9.4.6 Rifornimento Prodotti

Parametro: <b>Quantità Rifornimento Prodotti</b>	
Quantità [QRP]	1. Quantità non disponibile in magazzino [property invalidQRP] 2. Quantità disponibile in magazzino [validQRP]

Codice	Combinazione	Esito
TC_4.6.1	QRP2	La quantità del prodotto è stata rifornita con successo
TC_4.6.2	QRP1	La quantità di prodotti che si vuole rifornire non è disponibile in magazzino

#### 9.4.7 Cerca prodotto

Parametro: <b>Cerca</b>	
-------------------------	--

Presente [PC]	1. Il prodotto ricercato non è presente sul sito [property invalidPC] 2. Il prodotto ricercato è presente sul sito [property validPC]
---------------	--

Codice	Combinazione	Esito
TC_4.7.1	PC1	Il prodotto ricercato non è presente sul sito
TC_4.7.2	PC2	Il prodotto ricercato è presente sul sito