UNIVERSITÀ DEGLI STUDI DI SALERNO

Corso di Ingegneria del Software

A.A. 2023/2024

SecondLifeTech

TEST PLAN DOCUMENT

VERSIONE 1.0

01/03/2024

DOCENTE:

PROF. ANDREA DE LUCIA

PRESENTATO DA:

CAPRICANO DAVIDE, 05121 10422

GIARDINETTO GIUSEPPE, 05121 14655

SPERA ALFREDO JESHOUA, 05121 12240 (PROJECT MANAGER)

REVISION HISTORY

Data	Versione	Descrizione
28/02/2024	0.1	Prima versione
01/02/2024	1.0	Aggiunto test cases

Sommario

Sommario 2
1. Introduzione 3
2. Feature da testare e da non testare 3
   2.1. Feature da testare 3
   2.2. Feature da non testare 3
3. Criteri di Successo e Fallimento 3
4. Approccio 4
5. Test Cases 4
   5.1. CartService 4
   5.2. OrderService 4
   5.3. ProductService 4
   5.4. UserService 5

1. Introduzione

Questo documento ha lo scopo di pianificare gli aspetti gestionali del testing, per verificare se esistono delle differenze tra il comportamento osservato ed il comportamento atteso.

2. Feature da testare e da non testare

2.1. Feature da testare

Si andranno a testare tutte le interfacce dei servizi inerenti all'application layer del sistema:

- CartService
  - AddToCart
  - RemoveFromCart
  - EditProductQuantityInCart
  - FinalizeOrder
- OrderService
  - CreateAndPlaceNewOrder
  - SetOrderAsShipped
- ProductService
  - CreateNewModel
  - CreateNewVariation
  - UpdateModel
  - UpdateVariation
  - DeleteModel
  - DeleteVariation
- UserService
  - CreateNewUser
  - UpdateUser
  - DeleteUser

2.2. Feature da non testare

- Interfacce utente
- Sicurezza degli accessi (in quanto gestita da Spring Security)
- Storage Layer (in quanto gestito da JpaRepository)
- Performance

3. Criteri di Successo e Fallimento

Per ogni test case verranno forniti due classi di equivalenza in input, permettendo di testare i casi di input una sola volta nei casi in cui si ha un:

- Input corretto
- Input errato

4. Approccio

Si avrà un Unit Testing di tipo Black-box attraverso JUnit e Mockito. Quindi ci si concentrerà solamente sull'input fornito e sull'output generato, ignorando la struttura interna dei componenti.

5. Test Cases

5.1. CartService

Nome Feature	Input (corretto)
AddToCart	• Carrello (deve esistere) • Id variazione prodotto (la VP deve esistere) • Quantità (tra 0 e il numero disponibile)
RemoveFromCart	• Carrello (deve esistere) • Id prodotto nel carrello (il prodotto nel carrello deve esistere)
EditProductQuantityInCart	• Carrello (deve esistere) • Id variazione prodotto (la VP deve esistere) • Quantità nuova (tra 0 e il numero disponibile)
FinalizeOrder	• Carrello (deve esistere) • Indirizzo di spedizione (deve esistere) • Metodo di pagamento (deve esistere) • Pagamento con il sistema esterno andato a buon fine (vero)

5.2. OrderService

Nome Feature	Input (corretto)
CreateAndPlaceNewOrder	• Ordine (deve esistere e deve avere gli oggetti all'interno)
SetOrderAsShipped	• Ordine (deve esistere)

5.3. ProductService

Nome Feature	Input (corretto)
CreateNewModel	Modello: • Nome (tra i 3 e i 50 caratteri) • Brand (tra i 3 e i 50 caratteri) • Categoria (deve essere "SMARTPHONE" o "TABLET")
CreateNewVariation	Variazione: • Modello (deve esistere) • Prezzo (tra il 0 e 10000) • Anno (tra 2000 e l'anno corrente) • RAM (tra 1 e 128) • Dimensione (tra 0.1 e 20)

	• Spazio (tra 1 e 3000) • Colore (tra i 3 e i 50 caratteri) • Stato (deve essere “BUONO”, “OTTIMO” o “ACCETTABILE”)
UpdateModel	Modello con dati aggiornati: • Nome (tra i 3 e i 50 caratteri) • Brand (tra i 3 e i 50 caratteri) • Categoria (deve essere “SMARTPHONE” o “TABLET”)
UpdateVariation	Variazione con dati aggiornati: • Modello (deve esistere) • Prezzo (tra il 0 e 10000) • Anno (tra 2000 e l’anno corrente) • RAM (tra 1 e 128) • Dimensione (tra 0.1 e 20) • Spazio (tra 1 e 3000) • Colore (tra i 3 e i 50 caratteri) • Stato (deve essere “BUONO”, “OTTIMO” o “ACCETTABILE”)
DeleteModel	• Modello (deve esistere)
DeleteVariation	• Variazione (deve esistere)

5.4. UserService

Nome Feature	**Input (corretto)**
CreateNewUser	Utente: • Nome (tra i 3 e i 50 caratteri) • Cognome (tra i 3 e i 50 caratteri) • Numero di telefono (tra i 6 e i 15 caratteri) • E-mail (deve essere valida) • Ruolo (deve essere “ROLE_CLIENTE”, “ROLE_GESTORE_UTENTI”, “ROLE_GESTORE_PRODOTTI”, “ROLE_GESTORE_ORDINI”)
UpdateUser	Utente con dati aggiornati: • Nome (tra i 3 e i 50 caratteri) • Cognome (tra i 3 e i 50 caratteri) • Numero di telefono (tra i 6 e i 15 caratteri) • E-mail (deve essere valida) • Ruolo (deve essere “ROLE_CLIENTE”, “ROLE_GESTORE_UTENTI”, “ROLE_GESTORE_PRODOTTI”, “ROLE_GESTORE_ORDINI”)
DeleteUser	• Utente (deve esistere)

6. Materiali per il testing

Per eseguire il testing si ritiene necessario l'ausilio di un computer in grado di supportare la JVM utilizzata Java Spring per l'esecuzione. Non risulta quindi influente il sistema operativo della macchina.

I testing avverranno attraverso l'ausilio dell'IDE Intellij IDEA Ultimate (non risulta influente la versione, purché sia relativamente recente)