

UNIVERSITÀ DEGLI STUDI DI SALERNO

Corso di Ingegneria del Software

A.A. 2023/2024

SecondLifeTech

TEST PLAN DOCUMENT

VERSIONE 1.1

12/03/2024



DOCENTE:

PROF. ANDREA DE LUCIA

PRESENTATO DA:

CAPRICANO DAVIDE, 05121 10422

GIARDINETTO GIUSEPPE, 05121 14655

SPERA ALFREDO JESHOVA, 05121 12240 (PROJECT MANAGER)

REVISION HISTORY

Data	Versione	Descrizione
28/02/2024	0.1	Prima versione
01/03/2024	1.0	Aggiunto test cases
12/03/2024	1.1	Integration Testing

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	5
5.3. ProductService.....	5
5.4. UserService	6

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 vengono forniti due classi di equivalenza in input, in cui si ha:

- Input corretto (tutti i campi di input sono corretti)
- Input errato (tutti i campi di input sono errati)

Per ogni input errato, verranno testati tutti i casi in cui solamente uno dei campi di input è errato. Questo per testare che venga lanciata la giusta eccezione per quel particolare campo errato.

4. Approccio

Verranno utilizzati per il testing le componenti off-the-shelf [JUnit](#) e [Mockito](#).

4.1. Unit Testing

Lo Unit Testing sarà di tipo **Black-box**, quindi ci si concentrerà solamente sull'input fornito e sull'output generato, ignorando la struttura interna dei componenti.

4.2. Integration Testing

L'Integration Testing sarà di tipo **Big Bang**, quindi tutte le componenti del sistema, dopo essere testate individualmente, verranno testate insieme.

Nell'integration testing, verrà testato tutto in un solo metodo:

- La creazione di un nuovo utente
 - La creazione di un nuovo metodo di pagamento
 - La creazione di un nuovo indirizzo di spedizione
 - La creazione di un nuovo carrello
- La creazione di un nuovo prodotto
 - La creazione di un nuovo modello
 - La creazione di una nuova variazione
- L'aggiunta di una variazione al carrello
- La finalizzazione di un ordine
 - La creazione di un nuovo ordine
 - La rimozione dal carrello degli oggetti
 - La modifica della quantità degli oggetti ordinati

5. Test Cases

5.1. CartService

Nome Feature	Input (corretto)
AddToCart	<ul style="list-style-type: none">• Carrello (deve esistere)• Id variazione prodotto (la VP deve esistere)• Quantità (tra 0 e il numero disponibile)
RemoveFromCart	<ul style="list-style-type: none">• Carrello (deve esistere)• Id prodotto nel carrello (il prodotto nel carrello deve esistere)
EditProductQuantityInCart	<ul style="list-style-type: none">• Carrello (deve esistere)• Id variazione prodotto (la VP deve esistere)• Quantità nuova (tra 0 e il numero disponibile)
FinalizeOrder	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">Ordine (deve esistere e deve avere gli oggetti all'interno)
SetOrderAsShipped	<ul style="list-style-type: none">Ordine (deve esistere)

5.3. ProductService

Nome Feature	Input (corretto)
CreateNewModel	Modello: <ul style="list-style-type: none">Nome (tra i 3 e i 50 caratteri)Brand (tra i 3 e i 50 caratteri)Categoria (deve essere "SMARTPHONE" o "TABLET")
CreateNewVariation	Variazione: <ul style="list-style-type: none">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: <ul style="list-style-type: none">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: <ul style="list-style-type: none">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	<ul style="list-style-type: none">Modello (deve esistere)
DeleteVariation	<ul style="list-style-type: none">Variazione (deve esistere)

5.4. UserService

Nome Feature	Input (corretto)
CreateNewUser	Utente: <ul style="list-style-type: none">• 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: <ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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)