Laurea in Informatica - Università degli Studi di Salerno Corso di Ingegneria del Software - Prof. A. De Lucia


Data: 08/10/2024

TEAM

Gianluca Fusco 0512116485

Antonio Giorgio 0512106036

Cristian Di Popolo 0512105370

# Revision History

| Data | Versione | Descrizione | Autore |
|---|---|---|---|
| 14/12/2024 | 1.0 | Test Plan | G. Fusco<br>A. Giorgio<br>C. Di Popolo |

# Indice

# 1. Introduzione

All'interno del presente documento vengono riportate le strategie di testing adottate, le funzionalità che saranno testate e gli strumenti scelti per individuare eventuali errori, con l'obiettivo di fornire al cliente finale una piattaforma stabile e priva di malfunzionamenti. Sono state pianificate attività di testing per i seguenti raggruppamenti di funzionalità:

- User Management
- Merchandising
- Cart
- Orders

# 2. Relazione con altri documenti

## 2.1 Relazione con il Requirements Analysis Document (RAD)

I test case pianificati nel Test Plan sono elaborati in relazione ai requisiti funzionali e non funzionali definiti nel RAD.

## 2.2 Relazione con il System Design Document (SDD)

I test case pianificati nel Test Plan rispettano la suddivisione in sottosistemi presentata nell'SDD, verificando che ciascun modulo funzioni in modo isolato e integrato.

## 2.3 Relazione con il Object Design Document (ODD)

Il testing di unità e di integrazione, maggiormente legato all'ODD e alla divisione in package del sistema, è documentato all'interno del codice sorgente del progetto e non viene descritto dettagliatamente in questo documento.

# 3. Funzionalità da testare / non testare

## 3.1 Funzionalità da testare

I seguenti casi di test sono stati pianificati per le funzionalità principali:

### User

- Visualizzazione dell'area personale

### Login

- Registrazione dell'utente

### Magazzino

- Aggiunta prodotto

### Modifica prodotto

- Rimozione prodotto
- Visualizzazione pagina prodotto

### CARRELLO

- Visualizzazione carrello
- Modifica quantità dei prodotti nel carrello
- Eliminazione prodotti dal carrello
- Creazione ordine

### ORDINI

- Modifica ordine
- Cancellazione ordine

## 3.2 Funzionalità da non testare

Per ragioni di tempo e costi, non verranno testate le restanti funzionalità.

# 4. Criteri di accettazione / rifiuto

L'obiettivo del testing è identificare e correggere i fault nel sistema.
I criteri di successo o fallimento sono:

Pass: Il sistema si comporta come previsto nei casi di test specificati.

Fail: Qualsiasi deviazione dai comportamenti attesi provoca una fase di debugging e successiva ripetizione dei test, comprese eventuali verifiche su componenti correlate.

# 5. Approccio al test

## 5.1 Testing di Integrazione

Approccio:
È stato adottato un approccio di integrazione bottom-up, che facilita l'identificazione di fault a livello dei componenti di basso livello.

Strumenti:

- JUnit: Per la definizione e l'esecuzione dei test.
- Mockito: Per la generazione di stub e simulazioni delle dipendenze.

Processo:

- Test delle classi DAO in isolamento, utilizzando un database in-memory per le operazioni sui dati.
- Test delle componenti di controllo (Controller) in isolamento, simulando le dipendenze.
- Test dell'integrazione tra Controller e DAO.

Esempio di test:

- Per la funzionalità di aggiunta prodotto:
- Test della classe Product in isolamento.
- Test della classe ProductService in isolamento, simulando il comportamento della Product.

- Test integrato tra ProductService e Product utilizzando un database in-memory.

## 5.1 Testing di Unità

Approccio:

- Ogni metodo delle classi del sistema verrà testato in isolamento.

Strumenti:

- JUnit: Per la definizione dei casi di test.
- Mockito: Per simulare dipendenze e creare stub.
- DBUnit: Per la gestione di un database in-memory equivalente a quello di produzione.

Processo:

- I test di unità seguono un approccio black-box, documentato direttamente nel codice.

Tecnologie:

- Mock di dipendenze tramite Mockito.
- Verifica dei dati persistenti con DBUnit.

# 6. Test Cases

## USER

- Verifica che un utente possa registrarsi con credenziali valide.
- Test di login con credenziali valide/errate.
- Visualizzazione delle informazioni personali corrette per un utente autenticato.

## MAGAZZINO

- Aggiunta di un nuovo prodotto con dati validi.
- Modifica delle informazioni di un prodotto esistente.
- Eliminazione di un prodotto dal catalogo.
- Verifica della visualizzazione di un prodotto specifico.

## CARRELLO

- Visualizzazione corretta dei prodotti nel carrello.
- Modifica della quantità di un prodotto nel carrello.
- Rimozione di un prodotto dal carrello.
- Creazione di un ordine a partire dal contenuto del carrello.

## ORDINI

- Modifica dei dettagli di un ordine esistente.
- Cancellazione di un ordine e verifica della rimozione dal database.