*Testing Plan*

*Progetto*

*OctoPlus*



**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 07/01/2024 | **0.1** | **Prima stesura** | **Tutto il team** |

# Introduzione

Lo scopo di questo documento è delineare le strategie adottate per testare il sistema OctoPlus al fine di garantire un funzionamento privo di errori, procederemo all'analisi dettagliata dei vari componenti, indicando le specifiche strategie di testing e i casi di test corrispondenti.

# Relazioni con altri documenti

# Panoramica del sistema

Il sistema sarà un e-commerce per la vendita di prodotti per scuba-diving; offrirà una serie di funzionalità che gli permetteranno di competere sul mercato.

# Funzionalità da testare

* Funzionalità utente:
* Registrazione
* Autenticazione
* Ricerca Prodotto
* Filtro Prodotti
* Aggiunta di un prodotto al carrello
* Rimozione di un prodotto dal carrello
* Effettuare un ordine
* Modifica password
* Funzionalità gestore degli ordini:
* Gestione dello stato dell’ordine (cancellazione, cambio stato)
* Funzionalità gestore del catalogo:
* Aggiunta di un prodotto
* Rimozione prodotto
* Modifica prodotto
* Funzionalità gestore amministratore:
* Aggiunta di un amministratore
* Rimozione amministratore
* Modifica password amministratore

# Criterio Pass/Fail

Il testing è un approccio intenzionale volto a creare situazioni di fallimento o errori in modo pianificato. Questa pratica è preziosa poiché consente a noi sviluppatori di individuare difetti nel sistema prima del rilascio al cliente. Riducendo i difetti, miglioriamo la soddisfazione dell’utente durante l'utilizzo del sistema.

Per valutare se un caso di test ha rilevato un difetto, utilizziamo un "oracolo", cioè il risultato atteso dell'esecuzione, confrontandolo con il risultato effettivo del test. I criteri di pass/fail del testing si suddivideranno come segue:

* Fail: indica che il test ha avuto successo nel trovare un difetto, il risultato del test differisce dal suo oracolo.
* Pass: indica che il test non ha avuto successo nel trovare difetti, il risultato del test corrisponde al suo oracolo.

# Approccio

Per il testing del sistema, adotteremo un approccio bottom-up, focalizzandoci sulle tecniche di testing black-box permettendoci di focalizzarci sull’esperienza utente. Inizieremo con i test di unità, per poi procedere ai test di sistema. Ogni fase di testing sarà accompagnata da un report di esecuzione dei test, consentendoci di identificare e affrontare eventuali problematiche emerse durante il processo di verifica.

# Strumenti per il testing

Gli strumenti che andremo ad utilizzare per effettuare il testing sono:

* DBUnit
* Selenium
* Mockito

# Test case