



**Test Plan**

Data: 15/01/2021

**Coordinatore del Progetto:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Francesco Di Palma | 0512104586 |

**Partecipanti:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Giovanni Di Mauro | 0512104596 |
| Francesco Di Palma | 0512104586 |
| Maria Giuseppina Mosca | 0512106090 |
| Francesco Saviano | 0512104912 |

Sommario

[Introduzione 3](#_Toc61657135)

[Relazioni con altri documenti 3](#_Toc61657136)

[Funzionalità da testare e da non testare 3](#_Toc61657137)

[Criteri pass/fai testing 4](#_Toc61657138)

[Approccio 5](#_Toc61657139)

# Introduzione

Il test plan è un documento che si focalizza sugli aspetti manageriali del testing: gestisce lo sviluppo e le attività di testing effettuate sul sistema creato. Saranno identificati gli elementi e le funzionalità da testare, le strategie di testing, gli strumenti utilizzati per effettuarlo. Lo scopo del testing è quello di rilevare gli errori in maniera pianificata all’interno del codice realizzato, in modo che essi non si ripetano durante l’utilizzo da parte dell’utente finale. I risultati dei test servono per intervenire nei punti in cui sono presenti défaillace.

# Relazioni con altri documenti

Il test planning è in stretta relazione con i documenti prodotti fino, infatti i test case sono basati sulle funzionalità individuate in questi, particolarmente nel “documento di raccolta ed analisi dei requisiti”-(RAD).

La definizione dei sottosistemi ed i servizi da porre al testing sono stati individuati tramite il “System Design Document”-(SDD) e le classi sono state definite sulla base dell’ “Object Design Document”-(ODD).

# Funzionalità da testare e da non testare

Di seguito le funzionalità introdotte nel sistema che saranno sottoposte a test, suddivise per ogni gestione del sistema:

**GESTORE ACCOUNT**

* registrazione utente
* aggiunta indirizzo all’account
* cancellazione utente
* modifica indirizzo
* modifica profilo
* modifica profilo da parte dell’amministratore
* eliminazione di un account da parte dell’amministratore

**GESTORE CARRELLO**

* Aggiunta prodotti al carrello
* Rimozione prodotti dal carrello
* Modifica della quantità dei prodotti contenuti nel carrello
* Effettuare un acquisto

**GESTORE LOGIN**

* Effettuare login
* Effettuare logout

**GESTORE ORDINE**

* Cancellazione di un ordine da parte dell’amministratore
* Visualizzare i dettagli degli ordini
* Visualizzare tutti gli ordini dei clienti, da parte dell’amministratore

**GESTORE PRODOTTO**

* Inserimento prodotto
* Modifica prodotto
* Visualizzazione di tutti i prodotti
* Ricerca di un prodotto

# Criteri pass/fai testing

Lo scopo del testing è trovare una failure nel programma, quindi, il testing va a buon fine se l’output osservato è diverso dall’output atteso; una volta riscontrata una failure (nel caso sia presente nel sistema), si interviene per correggerla e si effettua di nuovo il testing per appurare che non abbia prodotto effetti collaterali, producendo altre failure .

# Approccio

Per il testing di unità si utilizzerà la tecnica BLACK-BOX. Con il BLACK-BOX testing ci focalizzeremo sul comportamento dell’input/output delle singole componenti senza tener 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 partizionare le “categorie” in classi di equivalenza chiamate “scelte”. Al termine saranno specificate le “combinazioni” delle scelte da testare creando delle istanze di casi di test specificando i valori dei dati effettivi per ciascuna scelta e determinare i risultati corrispondenti. Mediante il Category Partition otterremo, quindi, un test efficiente e privo di ridondanze.

# Testing di sistema

Il testing di sistema sarà effettuato tramite

# Sospensione e ripresa

La fase di testing può essere sospesa se si riscontrasse un difetto nel sistema per essere poi ripresa dopo aver risolto il problema riscontrato.

# Materiale per il testing

Gli strumenti utilizzati sono:

* WebServer Apache Tomcat 9 in locale, il quale gira sul sistema
* Client-web per effettuare richieste al server
* MySQL che gestisce il databse

.