



2024/25

TP

Test Plan Digitronics



Test Plan

CHIERCHIA PAOLO GIOVANNI, DI MICCO VINCENZO, ZOCCOLA DOMENICO

| | |
|--|-----------|
| Revision History | 1 |
| Team Members | 2 |
| 1. Introduzione | 2 |
| 2. Relazione con altri documenti | 2 |
| 3. Panoramica del sistema | 2 |
| 4. Features da testare/da non testare | 3 |
| 5. Pass/Fail criteria | 3 |
| 6. Approccio | 3 |
| 6.1 Testing di sistema | 3 |
| 6.2 Testing di integrazione..... | 3 |
| 6.3 Testing di unità | 4 |
| 6.4 Code review | 4 |
| 7. Sospensione | 4 |
| 8. Materiale di testing | 5 |
| 9. Test cases | 5 |
| 9.1 Checkout prodotti | 5 |
| 9.2 Modifica Prodotto..... | 9 |
| 9.3 Aggiunta recensione | 10 |
| 10. Testing schedule | 11 |

Revision History

| Data | Versione | Descrizione | Autori |
|-------------------|----------|-------------------------------|---------------|
| 30/12/2024 | 0.1 | Stesura capitolo da 1 a 8 | Tutto il team |
| 30/12/2024 | 0.2 | Test case Checkout | GPC |
| 30/12/2024 | 0.3 | Test case Modifica Prodotto | DZ |
| 30/12/2024 | 0.4 | Test case Aggiunta Recensione | VDM |
| 16/01/2025 | 1.0 | Revisione Finale | Tutto il team |

Team Members

| Nome | Ruolo nel Progetto | Acronimo | Informazioni di contatto |
|--------------------------|--------------------|----------|--------------------------------|
| Giovanni Paolo Chierchia | Team Member | GPC | g.chierchia8@studenti.unisa.it |
| Vincenzo Di Micco | Team Member | VDM | v.dimicco4@studenti.unisa.it |
| Domenico Zoccola | Team Member | DZ | d.zoccola3@studenti.unisa.it |

1. Introduzione

Digitronics è un e-commerce che si occupa della vendita di dispositivi elettronici come tablet, smartphone, smartwatch e relativi accessori.

Il Test Plan Digitronics ha lo scopo di descrivere e analizzare le attività di testing per la piattaforma, assicurando che ogni componente funzioni correttamente. Il documento mostra le strategie di testing adottate, le funzionalità oggetto di verifica e gli strumenti selezionati per l'identificazione dei difetti, al fine di consegnare al cliente finale una piattaforma completamente funzionante e priva di anomalie.

Sono state pianificate le attività di testing per le seguenti funzionalità:

- Aggiunta recensione
- Modifica prodotto
- Checkout prodotti

2. Relazione con altri documenti

Relazioni con il Requirements Analysis Document (RAD)

Il collegamento tra il Test Plan e il RAD si basa sui requisiti funzionali, e i test verranno eseguiti su queste funzionalità, considerando le specifiche indicate.

Relazioni con il System Design Document (SDD)

Nella fase di testing si deve tener conto della suddivisione in sottosistemi presentati nell' SDD.

3. Panoramica del sistema

Come riportato nell'SDD il nostro sistema è suddiviso secondo una struttura su tre livelli (**Three-Tier**): Interface Layer, Application Logic Layer e Storage Layer. Per la gestione del database relazionale verrà utilizzato **MySQL**.

Per il collegamento al database sarà utilizzato **JDBC**

Le interfacce utente e il front-end saranno realizzati con **JSP**, integrando **HTML5** e **CSS** per la definizione delle view.

4. Features da testare/da non testare

Le funzionalità che saranno sottoposte al testing sono:

- Aggiunta recensione
- Modifica prodotto
- Checkout prodotti

Le funzionalità per le quali non verranno eseguite attività di testing riguardano quelle non menzionate sopra.

5. Pass/Fail criteria

Il test ha successo se l'output è diverso da quello atteso; in quel caso, il testing avrà individuato un failure, il quale verrà corretto successivamente.

6. Approccio

Il testing dell'intero sistema si articola in tre fasi: testing di unità, testing di integrazione e testing di sistema.

La progettazione seguirà questo ordine: inizialmente verrà progettato il testing di sistema, basandosi sul RAD; successivamente, si procederà con il testing di integrazione, fondato sull'SDD (poichè descrive l'architettura del sistema, le interfacce e le dipendenze tra i sottosistemi); infine, il testing di unità sarà progettato durante la fase di implementazione.

L'esecuzione dei test avverrà invece nell'ordine inverso: prima il testing di unità, seguito dal testing di integrazione e, infine, dal testing di sistema.

6.1 Testing di sistema

Per il testing di sistema verrà svolto solo il testing funzionale per rispettare le tempistiche di distribuzione del progetto. Sarà utilizzato il tool Selenium IDE, che consente di registrare le azioni eseguite da un utente sul browser, facilitando l'implementazione e automatizzando l'esecuzione dei test case di sistema. Durante la fase di testing, il server sarà deployato in localhost.

6.2 Testing di integrazione

Per il testing di integrazione verrà utilizzato la strategia component-based **bottom-up**. Sarà utilizzato il framework **JUnit** per il testing delle classi Java. Il testing di integrazione sarà svolto solo per i metodi dei controller e service che si occupano di realizzare le funzionalità indicate nel primo paragrafo.

Saranno inizialmente testate le classi Service coinvolte, che si interfacciano con i DAO. Successivamente, verrà eseguito il test delle classi Controller, che si interfacciano con il rispettivo Service sottostante.

Per evitare che i test influiscano sul database MySQL è stato utilizzato il **database H2** che viene utilizzato nel testing di integrazione perché è un database leggero, veloce e facile da configurare. Permette di simulare un database reale senza dipendere da un'istanza esterna, garantendo test rapidi e isolati con dati temporanei. La chiamata ai controller verrà mockata usando Mockito. Per la continuous integration sarà utilizzato **Travis CI**. Per verificare le metriche di coverage di test è stato utilizzato lo strumento integrato **IntelliJ Code Coverage**.

6.3 Testing di unità

La strategia di unit testing prevede il test di tutti i metodi delle classi del sistema, ad eccezione dei bean e dei DAO, poiché queste componenti si limitano a fornire metodi getter e setter o a gestire l'inserimento e il recupero dei dati dal database.

I test saranno definiti seguendo un approccio black-box e documentati direttamente nel codice tramite Javadoc.

Sarà utilizzato il framework **JUnit** per il testing delle classi Java.

Per ogni classe di produzione, sarà creata una corrispondente classe di test, denominata secondo il formato NomeClasseProduzioneTest.

Le classi di test saranno sviluppate in parallelo a quelle di produzione per favorire una copertura più agevole del codice. Successivamente, queste classi saranno revisionate e, se necessario, modificate.

Durante il testing, verranno utilizzate anche altre tecnologie:

- **Mockito**, per creare gli stub e isolare le componenti sotto test.
- **IntelliJ Code Coverage** per calcolare metriche come la Branch Coverage.
- **Maven**, per gestire la build e automatizzare l'esecuzione dei test.

6.4 Code review

La code review viene gestita tramite pull request su GitHub. Ogni sviluppatore lavora su un branch separato per le proprie modifiche e, al termine, apre una pull request. Gli altri membri del team revisionano il codice, fornendo commenti e suggerimenti. Prima del merge, **Travis CI** esegue automaticamente i test per garantire la correttezza e la conformità agli standard. Una volta approvata e superati i test, la pull request viene integrata nel branch principale.

7. Sospensione

In questa sezione saranno definiti i criteri per sospendere il test e le attività che dovranno essere ripetute al momento della ripresa.

Criteri di sospensione

Il test proseguirà senza interruzioni fino al completamento, anche in caso di rilevazione di una failure. Tuttavia, il testing potrà essere temporaneamente interrotto qualora, durante l'esecuzione, emerga un errore nella definizione di uno dei test.

Criteri di ripristino

Il testing riprenderà una volta risolti i fault rilevati.

8. Materiale di testing

Gli strumenti per effettuare il testing sono un computer con un browser e una connessione ad Internet.

9. Test cases

Per definire i test frame si utilizzerà l'approccio category partition. Gli input verranno suddivisi in classi di equivalenza per ridurre al minimo il numero di test case. L'output atteso sarà determinato tramite un oracolo umano, data l'assenza di specifiche formali o semi-formali.

9.1 Checkout prodotti

Via, CAP e città saranno considerati come parametri separati per individuare eventuali anomalie nei singoli valori. Se i tre parametri sono corretti, questi saranno aggregati in un unico parametro "indirizzo", che sarà testato a parte poichè interviene la API OpenStreetMap per controllare l'esistenza dell'indirizzo in Italia. Questo approccio consente di individuare errori sia nei singoli campi sia nella combinazione complessiva, garantendo una validazione accurata.

| Parametro: Nome | |
|--|--|
| FORMATO: ^(?!\\s*\$)[a-zA-Zà-ÿÀ-ÿ\\s']{1,255}\$ | |
| Nome Categoria | Scelte per la categoria |
| Formato [FN] | <ol style="list-style-type: none">1. Rispetta il formato = false [error]2. Rispetta il formato = true [PROPERTY FN_OK] |
| Parametro: Cognome | |
| FORMATO: ^(?!\\s*\$)[a-zA-Zà-ÿÀ-ÿ\\s']{1,255}\$ | |
| Nome Categoria | Scelte per la categoria |
| Formato [FC] | <ol style="list-style-type: none">1. Rispetta il formato = false [error]2. Rispetta il formato = true [PROPERTY FCO_OK] |

| | |
|--|---|
| Parametro: Via | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza [LV] | 1. Lunghezza==0 [error] 2. Lunghezza > 255 [error] 3. Lunghezza>=1 AND lunghezza<=255 [PROPERTY LV_OK] |
| Parametro: CAP | |
| FORMATO: ^\d{5}\$ | |
| Nome Categoria | Scelte per la categoria |
| Formato [FCAP] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FCAP_OK] |
| Parametro: Città | |
| FORMATO: ^(?!\s*\$)[a-zA-Zà-ÿÀ-ÿ\s']{1,255}\$ | |
| Nome Categoria | Scelte per la categoria |
| Formato [FCI] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FCI_OK] |
| Parametro: Indirizzo (Via, CAP, Città) | |
| Nome Categoria | Scelte per la categoria |
| Esistenza [EIN] | 1. Indirizzo esiste = false [if PROPERTY LV_OK AND PROPERTY FCAP_OK AND PROPERTY FCI_OK] [error] 2. Indirizzo esiste = true AND indirizzo non è italiano [if PROPERTY LV_OK AND PROPERTY FCAP_OK AND PROPERTY FCI_OK] [error] 3. Indirizzo esiste = true AND indirizzo è italiano [if PROPERTY LV_OK AND PROPERTY FCAP_OK AND PROPERTY FCI_OK] [PROPERTY EIN_OK] |
| Parametro: N° Civico | |
| FORMATO: ^\d{1,5}(\s?(bis tris [a-zA-Z]))?\$ | |
| Nome Categoria | Scelte per la categoria |
| Formato [FNC] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FNC_OK] |

| | |
|--|--|
| Parametro: Telefono | |
| FORMATO: ^\+?[1-9]\d{1,14}\$ | |
| Nome Categoria | Scelte per la categoria |
| Formato [FT] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FT_OK] |
| Parametro: Metodo di spedizione | |
| Nome Categoria | Scelte per la categoria |
| Esistenza [EMS] | 1. Metodo di spedizione Esiste = false [errore] 2. Metodo di spedizione Esiste = true [PROPERTY EMS_OK] |
| Parametro: Numero carta | |
| FORMATO: ^(?:\d{4}){3}\d{4}\$ | |
| Nome Categoria | Scelte per la categoria |
| Formato [FNCA] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FNCA_OK] |
| Parametro: Nome intestatario | |
| FORMATO: ^(?!\\s*\$)[a-zA-Zà-ÿÀ-ÿ\\s']{1,255}\$ | |
| Nome Categoria | Scelte per la categoria |
| Formato [FNI] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FNCA_OK] |
| Parametro: Scadenza carta | |
| FORMATO: ^(0[1-9] 1[0-2])/[0-9]{2}\$ | |
| Nome Categoria | Scelte per la categoria |
| Formato [FSC] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FSC_OK] |
| Validità [VSC] | 1. Scadenza non valida [if PROPERTY FSC_OK] [error] 2. Scadenza valida [if PROPERTY FSC_OK] [PROPERTY VSC_OK] |
| Parametro: CVV | |
| FORMATO: ^[0-9]{3}\$ | |
| Nome Categoria | Scelte per la categoria |

| | |
|----------------|---|
| Formato [FCVV] | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FSC_OK] |
|----------------|---|

| Test Case ID | Test frame | Esito |
|--------------|---|--|
| TC_1.1_1 | FN1 | Errato: Il nome non rispetta il formato |
| TC_1.1_2 | FN2, FC1 | Errato: Il cognome non rispetta il formato |
| TC_1.1_3 | FN2, FC2, LV1 | Errato: La stringa della via non può essere vuota |
| TC_1.1_4 | FN2, FC2, LV2 | Errato: La stringa della via è troppo lunga |
| TC_1.1_5 | FN2, FC2, LV3, FCAP1 | Errato: Il CAP non rispetta il formato |
| TC_1.1_6 | FN2, FC2, LV3, FCAP2, FCI1 | Errato: La città non rispetta il formato |
| TC_1.1_7 | FN2, FC2, LV3, FCAP2, FCI2, EIN1 | Errato: L'indirizzo non esiste |
| TC_1.1_8 | FN2, FC2, LV3, FCAP2, FCI2, EIN2 | Errato: L'indirizzo esiste ma non è italiano |
| TC_1.1_9 | FN2, FC2, LV3, FCAP2, FCI2, EIN3, FNC1 | Errato: Il numero civico non rispetta il formato |
| TC_1.1_10 | FN2, FC2, LV3, FCAP2, FCI2, EIN3, FNC2, FT1 | Errato: Il telefono non rispetta il formato |
| TC_1.1_11 | FN2, FC2, LV3, FCAP2, FCI2, EIN3, FNC2, FT2, EMS1 | Errato: Il metodo di spedizione non esiste |
| TC_1.1_12 | FN2, FC2, LV3, FCAP2, FCI2, EIN3, FNC2, FT2, EMS2, FNCA1 | Errato: Il numero della carta non rispetta il formato |
| TC_1.1_13 | FN2, FC2, LV3, FCAP2, FCI2, EIN3, FNC2, FT2, EMS2, FNCA2, FNI1 | Errato: Il nome intestatario carta non rispetta il formato |
| TC_1.1_14 | FN2, FC2, LV3, FCAP2, FCI2, EIN3, FNC2, FT2, EMS2, FNCA2, FNI2, FSC1 | Errato: La scadenza carta non rispetta il formato |
| TC_1.1_15 | FN2, FC2, LV3, FCAP2, FCI2, EIN3, FNC2, FT2, EMS2, FNCA2, FNI2, FSC2, VSC1 | Errato: La scadenza della carta non è valida (carta scaduta) |
| TC_1.1_16 | FN2, FC2, LV3, FCAP2, FCI2, EIN3, FNC2, FT2, EMS2, FNCA2, FNI2, FSC2, VSC2, FCVV1 | Errato: Il CVV non rispetta il formato |

| | | |
|-----------|---|----------|
| TC_1.1_17 | FN2, FC2, LV3, FCAP2, FCI2, EIN3, FNC2, FT2, EMS2, FNCA2, FNI2, FSC2, VSC2, FCVV2 | Corretto |
|-----------|---|----------|

9.2 Modifica Prodotto

| Parametro: Nome Prodotto | |
|---|---|
| Nome Categoria | Scelte per la categoria |
| Lunghezza [LN] | 1. Lunghezza>255 [error] 2. Lunghezza<1 [error] 3. Lunghezza<=255 AND Lunghezza>=1 [PROPERTY LN_OK] |
| Parametro: Descrizione | |
| Lunghezza [LD] | 1. Lunghezza<1 [error] 2. Lunghezza>=1 [PROPERTY LD_OK] |
| Parametro: Prezzo Formato: ^(?!\0\00\$)\d+(\.\d{2,})\$ | |
| Lunghezza [LP] | 1. Lunghezza<1 [error] 2. Lunghezza>=1 [PROPERTY LP_OK] |
| Correttezza [CP] | 1. Prezzo<=0[errore] 2. Prezzo>0 [PROPERTY CP_OK] |
| Formato [FP] | 1. Rispetta il formato =false 2. Rispetta il formato =true [PROPERTY FP_OK] |
| Parametro: Quantità | |
| Lunghezza [LQ] | 1. Lunghezza<1 [error] 2. Lunghezza>=1 [PROPERTY LQ_OK] |
| Correttezza [CQ] | 1. Quantità<=0 [error] 2. Quantità>0 [PROPERTY CQ_OK] |
| Parametro: Categoria | |
| Esistenza [EI] | 1. Categoria Esistente = false [error] 2. Categoria Esistente = true [PROPERTY_EI_OK] |

| Test Case ID | Test frame | Esito |
|--------------|--|---|
| TC_2.1_1 | LN1 | Errato: nome prodotto troppo lungo |
| TC_2.1_2 | LN2 | Errato: nome prodotto troppo breve |
| TC_2.1_3 | LN3, LD1 | Errato: descrizione troppo breve |
| TC_2.1_4 | LN3, LD2, LP1 | Errato: lunghezza prezzo troppo breve |
| TC_2.1_5 | LN3, LD2, LP2, CP1 | Errato: prezzo minore uguale di zero |
| TC_2.1_6 | LN3, LD2, LP2, CP2, FP1 | Errato: il prezzo non rispetta il formato |
| TC_2.1_7 | LN3, LD2, LP2, CP2, FP2, LQ1 | Errato: quantità troppo breve |
| TC_2.1_8 | LN3, LD2, LP2, CP2, FP2, LQ2, CQ1 | Errato: quantità minore uguale di zero |
| TC_2.1_9 | LN3, LD2, LP2, CP2, FP2, LQ2, CQ2, EI1 | Errato: categoria non esistente |
| TC_2.1_10 | LN3, LD2, LP2, CP2, FP2, LQ2, CQ2, EI2 | Corretto |

9.3 Aggiunta recensione

| Parametro: Titolo | |
|------------------------|---|
| Nome Categoria | Scelte per la categoria |
| Lunghezza [LT] | 1. Lunghezza > 255 [error] 2. Lunghezza < 1 [error] 3. Lunghezza <= 255 AND Lunghezza >= 1 [PROPERTY LT_OK] |
| Parametro: Descrizione | |
| Nome Categoria | Scelte per la categoria |
| Lunghezza [LD] | 1. Lunghezza < 1 [error] 2. Lunghezza >= 1 [PROPERTY LD_OK] |
| Parametro: Punteggio | |

| FORMATO: ^[1-5]\$ | |
|----------------------|---|
| Nome Categoria | Scelte per la categoria |
| Formato [FP] | 1. Rispetta formato = false [error] 2. Rispetta formato = true [PROPERTY CP_OK] |

| Test Case ID | Test frame | Esito |
|--------------|---------------|--|
| TC_3.1_1 | LT1 | Errato: titolo troppo lungo |
| TC_3.1_2 | LT2 | Errato: titolo troppo corto |
| TC_3.1_3 | LT3, LD1 | Errato: descrizione troppo corta |
| TC_3.1_4 | LT3, LD2, FP1 | Errato: il punteggio non rispetta il formato |
| TC_3.1_5 | LT3, LD2, FP2 | Corretto |

10. Testing schedule

I test saranno organizzati come specificato nel paragrafo 6 di questo documento. Una volta concluso lo sviluppo, tutti i test saranno rieseguiti per garantirne il corretto funzionamento e produrre i report finali.