

**Test Plan**

[Versione 1]

**Sommario**

[**Informazioni sul documento** 2](#_Toc177983745)

[**Generalità** 2](#_Toc177983746)

[**Team Project** 2](#_Toc177983747)

[**Revision History** 3](#_Toc177983748)

[**1. Introduzione** 5](#_Toc177983749)

[**2. Relazione con altri documenti** 5](#_Toc177983750)

[**3. Panoramica del sistema** 6](#_Toc177983751)

[**4. Features da testare/da non testare** 6](#_Toc177983752)

[**5. Pass/Fail criteria** 6](#_Toc177983753)

[**6. Approccio** 6](#_Toc177983754)

[**7. Sospensione e ripristino** 6](#_Toc177983755)

[**8. Materiale di testing** 6](#_Toc177983756)

[**9. Test cases** 6](#_Toc177983757)

[**10. Testing schedule** 6](#_Toc177983758)

**Informazioni sul documento**

**Generalità**

* **Progetto**: TechHeaven
* **Versione**: [Versione 1]
* **Documento**: Test plan
* **Data**: [23/09/2024]

**Team Project**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome Membro | Matricola | Ruolo | Contatti |
| Dorotea Serrelli | 0512113740 | Project manager | d.serrelli1@studenti.unisa.it |
| Raffaella Sabatino | 0512115114 | Team member | r.sabatino17@studenti.unisa.it |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 23/09/2024 | 0.1 | Stesura delle sezioni del Test plan seguenti : Introduzione, Relazione con altri documenti, Panoramica del sistema. | Dorotea Serrelli |
| 23/09/2024 | 0.2 | Stesura sezioni: Features da testare/non testare, Pass/Fail criteria, Sospensione e ripristino | Raffaella Sabatino |
|  |  |  |  |

**1. Introduzione**

Il sistema software TechHeaven, commissionato dalla società TechHeavenSrl, si propone di fornire alla clientela informazioni sulla società , sul punto vendita “TechHeaven – Il paradiso digitale” e sui prodotti trattati.

La società committente, infatti, è responsabile dall’anno 2000 della gestione del negozio “TechHeaven – Il paradiso digitale”, specializzato nella vendita di prodotti elettronici, elettrodomestici, telefonia.

Tale negozio è, attualmente, un punto vendita di riferimento nella zona per lo smercio, la qualità e il prezzo dei prodotti, al punto che soddisfa un grande bacino di utenza, quasi esclusivamente residente nella provincia.

La società intende espandere i confini della propria attività ed ampliare la clientela, avvalendosi di un sistema software che consenta, sotto il profilo soggettivo, una maggiore conoscibilità della società e dell’affidabilità della stessa; sotto il profilo oggettivo, favorisca l’incremento della vendita dei prodotti.

In questo documento verranno descritte ed analizzate le attività di Testing per il sistema software TechHeaven, in modo da garantire che ogni aspetto funzioni in modo corretto.

All’interno del documento sono riportate le strategie di testing adottate, le funzionalità testate e gli strumenti scelti per la rilevazione degli errori, con lo scopo di presentare al cliente finale una piattaforma priva di malfunzionamenti.

Sono state pianificate attività di testing per le seguenti gestioni:

* Gestione utente;
* Gestione ordini commissionati al negozio;
* Gestione catalogo dei prodotti del negozio.

**2. Relazione con altri documenti**

Di seguito si riportano I documenti di progetto prodotti, i quali sono stati utilizzati per l’individuazione dei test case:

* **Relazioni con il Requirements Analysis Document (RAD)**

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

* **Relazioni con il System Design Document (SDD)**

I test case pianificati nel Test Plan devono rispettare la suddivisione in sottosistemi presentata nell’SDD.

* **Relazioni con l’Object Design Document (ODD)**

A partire da questo documento si elaborano i test di unità e di integrazione poiché quest’ultimi sono legati alla divisione del sistema in packages.

Pertanto, il test di unità e di integrazione saranno scritti e documentati unicamente all’interno del codice dell’applicativo.

Per tale motivo, nel presente documento non vi saranno riferimenti al loro design.

**3. Panoramica del sistema**

Lo stile architetturale del sistema software proposto è Three-tier, adatto per i sistemi interattivi poiché consente di gestire interazioni complesse tra i componenti del sistema e l’utente.

La separazione della logica di presentazione da quella relativa all’elaborazione ed archiviazione dei dati offre una serie di vantaggi come leggibilità, manutenzione e riuso.

Nello sviluppo del sistema si sono utilizzati come componenti off-the-shelf:

* HTML, CSS e Javascript per la presentazione dei dati e l'interfaccia utente;
* Java e JSP (JavaServer Pages) per la logica applicativa;
* un database relazionale con DBMS MySQL, che utilizza il linguaggio SQL per gestire e interrogare i dati persistenti. L’utilizzo del database relazionale per l’archiviazione e gestione dei dati persistenti è motivata dai seguenti vantaggi: scalabilità ed offerta dei servizi per la gestione della concorrenza, il controllo degli accessi, il recupero da arresto.

**4. Features da testare/da non testare**

Si delineano le funzionalità del sistema software, divise per gestioni, sulle quali si concentrerà il testing:

* **Gestione utente**
* Registrazione utente;
* Autenticazione utente;
* Modifica dati personali;
* Reimpostazione della password;
* Ricerca nel sito per menù di navigazione;
* Ricerca nel sito per barra di ricerca;
* Modifica dei prodotti nel carrello virtuale;
* Modifica dei prodotti nella lista dei desideri (wishlist);
* Acquisto dei prodotti nel carrello virtuale (creazione ordine).
* **Gestione degli ordini**
* Visualizzazione degli ordini evasi;
* Visualizzazione degli ordini da evadere e preparazione di un ordine alla spedizione;
* Creazione richiesta di approvvigionamento di un prodotto non disponibile.
* **Gestione del catalogo**
* Inserimento di un prodotto;
* Cancellazione di un prodotto;
* Modifica delle specifiche di un prodotto.

**5. Pass/Fail criteria**

L’attività di testing ha l’obiettivo di verificare la correttezza nel funzionamento del prodotto software realizzato. Ciò avviene identificando eventuali errori (faults) all’interno del sistema, per effettuarne un successivo intervento di eliminazione prima che l’intero sistema venga rilasciato.

L’esito di un test case è valutato mediante un oracolo, inteso come il risultato atteso dell’esecuzione di un test case, basandosi sui requisiti.

Un test case ha successo (PASS) se, dato un input al sistema, l’output ottenuto è diverso dall’oracolo.

Al contrario, un test case fallisce (FAIL) se, dato un input al sistema, l’output ottenuto è uguale all’output atteso dall’oracolo.

Il testing sarà considerato valido se tutti i seguenti vincoli saranno rispettati:

* raggiungere una branch coverage non inferiore al 75%;
* effettuare test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti.

**6. Approccio**

**7. Sospensione e ripristino**

In questa sezione si specificano i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprende il test.

**Criteri di sospensione**

Il testing non verrà sospeso fino alla sua terminazione, anche in caso di rilevazione di una failure. Il testing potrà essere momentaneamente sospeso nel caso venga restituito, al momento dell’esecuzione, un errore nella definizione di uno dei test stessi.

**Criteri di ripristino**

Il testing verrà ripreso dopo aver risolto gli errori (faults) individuati.

**8. Materiale di testing**

**9. Test cases**

In questa sezione si definiscono i test cases individuati, a seguito dell’applicazione del Category partition, per le funzionalità da testare.

Per ciascun test case individuato, è stato elaborato un documento a parte, denotato *NomeTestCase\_TestCase\_TechHeaven*, nel quale si specificano nel dettaglio i parametri di input individuati con le classi e scelte corrispondenti, nonché

**9.1 Test cases per Gestione utente**

…

**9.1.1 Registrazione utente**

|  |  |
| --- | --- |
| Parametro : Username | |
| Formato : ^[a-Za-Z]{5, }$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FUS] | 1. Username corretto [property FUS\_OK] |
| 2. Username non corretto [Error] |
| Lunghezza [LUS] | 1. Username di lunghezza corretta >= 5 corretto [property LUS\_OK]  2. Username di lunghezza non corretta < 5 [Error] |
| Username presente nel database [UP] | 1. Username presente nel database [if FUS\_OK AND LUS\_OK]  2. Username non presente nel database [property UP\_OK] |
| Parametro : Password | |
| Formato: ^(?=.\*[a-zA-Z])(?=.\*[0-9]).{5,}$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FPW] | 1. Password con formato corretto [property FPW\_OK]  2. Password con formato non corretto [Error] |
| Lunghezza [LPW] | 1. Password con lunghezza corretta >= 5 [property LPW\_OK]  2. Password con lunghezza non corretta < 5 [Error] |
| Parametro : Nome | |
| Formato: ^[A-Za-z\s]+$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FNO] | 1. Nome con formato corretto [property FNO\_OK]  2. Nome con formato non corretto [Error] |
| Parametro : Cognome | |
| Formato: ^[A-Za-z\s]+$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FCOG] | 1. Cognome con formato corretto [property FCOG\_OK]  2. Cognome con formato non corretto [Error] |
| Parametro : Sesso | |
| Formato: {F, M} | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FGEN] | 1. Genere con formato corretto [property FGEN \_OK]  2. Genere con formato non corretto [Error] |
| Parametro : Email | |
| Formato: ^\S+@\S+\.\S+$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FMAIL] | 1. Email con formato corretto [property FMAIL \_OK]  2. Email con formato non corretto [Error] |
| Email presente nel database [MAILP] | 1. Email presente nel database (if FMAIL\_OK)  2. Email non presente nel database [property MAILP\_OK] |
| Parametro : Telefono | |
| Formato: ^[0-9]{3}-[0-9]{3}-[0-9]{4}$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FTEL] | 1. Telefono con formato corretto [property FTEL \_OK]  2. Telefono con formato non corretto [Error] |
| Parametro : Via | |
| Formato: ^[A-Za-z\s]+$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FVIA] | 1. Via con formato corretto [property FVIA \_OK]  2. Via con formato non corretto [Error] |
| Parametro : Civico | |
| Formato: ^(([0-9])|(([0-9]+|\w)(\w|[0-9]+)))$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FNCIV] | 1. Numero civico con formato corretto [property FNCIV \_OK]  2. Numero civico con formato non corretto [Error] |
| Parametro : Città | |
| Formato: ^[A-Za-z\s]+$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FCITTA] | 1. Città con formato corretto [property FCITTA \_OK]  2. Città con formato non corretto [Error] |
| Parametro : Cap | |
| Formato: ^\d{5}$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FCAP] | 1. CAP con formato corretto [property FCAP \_OK]  2. CAP con formato non corretto [Error] |
| Parametro : Provincia | |
| Formato: ^[A-Za-z]{2}$ | |
| Nome categoria | **Scelte per la categoria** |
| Formato [FPROV] | 1. Provincia con formato corretto [property FPROV \_OK]  2. Provincia con formato non corretto [Error] |

|  |  |  |
| --- | --- | --- |
| Test Case ID | Test frame | Esito |
| TC\_1.1\_1 | FUS2 | L’username deve avere almeno lunghezza pari a 5 e contenere solo lettere. |
| TC\_1.1\_2 | FUS1, LUS2 | L’username deve avere almeno lunghezza pari a 5 e contenere solo lettere. |
| TC\_1.1\_3 | FUS1, LUS1, UP1 | Non e\' possibile associare l'username inserita al tuo account.  Riprova la registrazione inserendo un'altra username. |
| TC\_1.1\_4 | FUS1, LUS1, UP2, FPW2 | La password deve avere almeno 5 caratteri che siano lettere e numeri. |
| TC\_1.1\_5 | FUS1, LUS1, UP2, FPW1, LPW2 | La password deve avere almeno 5 caratteri che siano lettere e numeri. |
|  |  |  |

**10. Testing schedule**