**Università degli Studi di Salerno**

**Corso di Ingegneria del Software**

Immagine che contiene Carattere, grafica, Elementi grafici, logo

Descrizione generata automaticamente

Test Plan

Car – Zone

Versione 0.1

13/12/2024

**Coordinatore del progetto:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Francesco Pio Cataudo | 0512116773 |

**Partecipanti**:

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Francesco Pio Cataudo | 0512116773 |
| Francesco Santoro | 0512117079 |
| Francesco Pio Bottaro | 0512118180 |
| Errico Aquino | 0512117730 |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| **13/12/2024** | 0.1 | Strutturazione documento ed informazioni iniziali. | Team |

Sommario

[Revision History 2](#_Toc184998802)

[1.Introduzione 5](#_Toc184998803)

[2. Relazione con altri documenti 5](#_Toc184998804)

[3. Overview del Sistema 6](#_Toc184998805)

[4. Feature da testare / non testare 7](#_Toc184998806)

[5. Pass / Fail criteria 7](#_Toc184998807)

[6. Approccio 7](#_Toc184998808)

[7. Suspension and resumption 7](#_Toc184998809)

[8. Testing materias (hardware/software requirements 8](#_Toc184998810)

[9. Testing cases 8](#_Toc184998811)

[10. Testing schedule 8](#_Toc184998812)

## 1.Introduzione

Questo piano di test definisce gli obiettivi, l'estensione e la strategia dei test per il sistema Car-Zone. L'obiettivo principale è garantire che il sistema soddisfi i requisiti funzionali e non funzionali definiti nel RAD e nell'SDD, identificando difetti e riducendo i rischi associati al rilascio. Questo documento rappresenta una guida chiara per gestori e tester, fornendo:

* Un quadro completo delle funzionalità del sistema.
* Un piano per identificare e risolvere i difetti prima del rilascio.
* La documentazione per monitorare l'efficacia del processo di test.

## 2. Relazione con altri documenti

Questo piano di test si riferisce ai seguenti documenti fondamentali:

* **RAD (Requirements Analysis Document)**: definisce i requisiti funzionali e non funzionali che guideranno i casi di test.
* **SDD (System Design Document)**: fornisce dettagli sull'architettura del sistema e le sue componenti principali, usati per pianificare test di integrazione e sistema.
* **ODD (Object Design Document)**: descrive le classi e le loro interfacce, essenziali per i test unitari.

Uno schema di denominazione è stato introdotto per collegare i test ai requisiti, alle classi e ai moduli: ad esempio, "Test\_R001" corrisponde al requisito funzionale "R001" e al metodo principale della classe correlata.

## 3. Overview del Sistema

Il sistema Car-Zone è suddiviso in moduli principali, ciascuno dei quali è stato progettato per essere testato separatamente:

* **Gestione Utenti**: Classi Utente, Cliente, Amministratore.
* **Gestione Catalogo Auto**: Classe CatalogoAuto per la ricerca, aggiunta, modifica e rimozione di auto.
* **Gestione Ordini**: Classe Ordine per la creazione, conferma e gestione dello stato degli ordini.
* **Interfaccia Utente**: Front-end per l'interazione con gli utenti, incluso il flusso di navigazione.

Ogni modulo dipende da altri in un'architettura stratificata, che consente una strategia di test bottom-up per i componenti di basso livello e top-down per il front-end.

## 4. Feature da testare / non testare

**Funzionalità da testare:**

1. Gestione Utenti:
   * Registrazione di nuovi utenti con email e password validi.
   * Autenticazione con credenziali valide/errate.
2. Catalogo Auto:
   * Aggiunta di nuove auto con dettagli completi (marca, modello, prezzo).
   * Ricerca e filtro nel catalogo.
   * Modifica e rimozione di auto esistenti.
3. Gestione Ordini:
   * Creazione di ordini per auto disponibili.
   * Modifica dello stato dell'ordine (confermato, annullato).
4. Interfaccia Utente:
   * Navigazione fluida tra le pagine.
   * Visualizzazione corretta delle informazioni sui cataloghi e ordini.

Funzionalità non testate:

* Integrazioni future con sistemi di pagamento.
* Ottimizzazioni per dispositivi obsoleti o browser non supportati.

## 5. Pass / Fail criteria

* **Unit Test:**
  + Il metodo deve restituire risultati corretti per input validi.
  + Deve gestire correttamente input non validi con errori espliciti.
* **Test di Integrazione:**
  + Tutti i componenti devono comunicare correttamente secondo le specifiche.
  + Eventuali errori di connessione o mancata risposta devono essere gestiti con fallback appropriati.
* **Test di Sistema:**
  + Tutte le funzionalità principali devono essere accessibili e funzionanti.
  + Le prestazioni devono rispettare i vincoli definiti (es. tempo di risposta < 2 secondi).

## 6. Approccio

**Strategia di test:**

1. **Unit Testing**:
   * Utilizzo di framework JUnit per testare metodi individuali e classi.
   * Ogni test è progettato per coprire un metodo o una funzionalità specifica.
2. **Test di Integrazione**:
   * Strategia bottom-up per verificare le interazioni tra i moduli di basso livello.
   * Strategia top-down per testare l'interfaccia utente insieme ai servizi di back-end.
3. **Test di Sistema**:
   * Verifica che l'intero sistema soddisfi i requisiti funzionali e non funzionali.
   * Esecuzione di test di carico per valutare la scalabilità.

Diagrammi UML delle dipendenze sono utilizzati per pianificare i test di integrazione e garantire una copertura completa.

## 7. Sospensione e ripresa

**Criteri di sospensione:**

* Difetti critici che impediscono ulteriori test (es. crash del sistema).
* Risorse non disponibili (es. ambienti di test non configurati).

**Criteri di ripresa:**

* Risoluzione dei difetti bloccanti.
* Verifica della disponibilità delle risorse necessarie.

## 8. Testing materias (hardware/software requirements

* **Hardware:**
  + Server dedicati per database e applicazioni.
  + Client con configurazioni standard (Windows 10).
* **Software:**
  + JUnit per test unitari.
  + Selenium per automazione dei test UI.
  + Postman per test API.
  + Database MySQL per i test del back-end.

## 9. Testing cases

**Gestione Utenti**

* Test\_R001: Verifica della registrazione utenti.
  + Input: email valida, password.
  + Passaggi:
    1. Inserire i dati nel modulo di registrazione.
    2. Premere "Registrati".
  + Output atteso: Creazione dell'account e reindirizzamento alla pagina di login.
* Test\_R002: Verifica autenticazione.
  + Input: Email e password corretti.
  + Passaggi:
    1. Inserire le credenziali nel modulo di login.
    2. Premere "Login".
  + Output atteso: Accesso al sistema con messaggio di benvenuto.
* Test\_R003: Verifica autenticazione con credenziali errate.
  + Input: Email o password non corretti.
  + Passaggi:
    1. Inserire credenziali errate nel modulo di login.
    2. Premere "Login".
  + Output atteso: Messaggio di errore "Credenziali non valide".

**Catalogo Auto**

* Test\_R004: Aggiunta di un'auto al catalogo.
  + Input: Dettagli auto (marca, modello, prezzo,ecc. ).
  + Passaggi:
    1. Accedere come amministratore.
    2. Inserire i dettagli dell'auto nel modulo di aggiunta.
    3. Premere "Aggiungi".
  + Output atteso: Auto aggiunta con successo e visibile nel catalogo.
* Test\_R005: Ricerca nel catalogo auto.
  + Input: Filtro "Marca: Fiat".
  + Passaggi:
    1. Inserire "Fiat" nel campo di ricerca.
    2. Premere "Cerca".
  + Output atteso: Visualizzazione di tutte le auto Fiat disponibili nel catalogo.
* Test\_R006: Modifica dei dettagli di un'auto.
  + Input: Nuovi dettagli auto (es. prezzo aggiornato).
  + Passaggi:
    1. Selezionare un'auto dal catalogo.
    2. Modificare i dettagli e salvare.
  + Output atteso: Dettagli aggiornati visibili nel catalogo.
* Test\_R007: Rimozione di un'auto.
  + Input: ID auto da rimuovere.
  + Passaggi:
    1. Selezionare un'auto dal catalogo.
    2. Premere "Rimuovi".
  + Output atteso: Auto rimossa dal catalogo.

Gestione Ordini

* Test\_R008: Creazione di un ordine.
  + Input: ID cliente, ID auto.
  + Passaggi:
    1. Selezionare un'auto dal catalogo.
    2. Premere "Ordina".
  + Output atteso: Ordine creato con stato "Confermato".
* Test\_R009: Annullamento di un ordine.
  + Input: ID ordine.
  + Passaggi:
    1. Selezionare un ordine dallo storico.
    2. Premere "Annulla".
  + Output atteso: Stato ordine aggiornato a "Annullato".

Interfaccia Utente

* Test\_R010: Navigazione tra le pagine.
  + Input: Nessuno.
  + Passaggi:
    1. Accedere al sistema.
    2. Navigare tra le diverse sezioni (Catalogo, Profilo, Ordini).
  + Output atteso: Navigazione fluida senza errori.

## 10. Testing schedule

**Dettagli:**

* Responsabilità:
  + Sviluppatori: Test unitari e integrazione.
  + Team QA: Test di sistema e accettazione.
* Pianificazione:
  + Settimana 1-2: Test unitari.
  + Settimana 3: Test di integrazione.
  + Settimana 4: Test di sistema e accettazione.

**Rischi e contingenze:**

* Ritardi nello sviluppo.
* Difetti critici non previsti.