

Università degli Studi di Salerno

Corso di Ingegneria del Software

**Auto Shop
Test Plan
Versione 1.0**



Data: 02/02/2018

Cognome Nome	Matricola
Corrado Mancino Alfredo	0512102506
Carbè Daniele	0512102326
Caloia Gennaro	0512102332

Indice

1. Introduzione
2. Relazione con altri documenti
3. Panoramica del sistema
4. Funzionalità da testare e non
5. Pass/Fail criteria
6. Approccio
 - 6.1 Testing di unità
 - 6.2 Testing di integrazione
 - 6.3 Testing di sistema
7. Test Cases
8. Specifica dei Test Cases

1. Introduzione

Lo scopo di questo documento è quello di analizzare e gestire lo sviluppo e le attività di testing riguardanti il software Auto shop. Questa sessione di lavoro deve verificare il corretto funzionamento del sistema in diversi casi, studiati appositamente per mettere alla prova ogni singola funzionalità e caratteristica del sistema, al fine di ottenere un corretto funzionamento. I risultati di questi test saranno utilizzati per capire dove bisognerà intervenire, e quindi correggere eventuali errori o apportare modifiche per il miglioramento dei vari sottosistemi. Il processo verrà iterato fino a che non si otterranno i risultati attesi in accordo con i tempi di sviluppo previsti.

2. Relazione con gli altri documenti

Per verificare il corretto funzionamento del sistema sono stati predisposti dei test cases basati sulle funzionalità individuate sia nel documento inerente gli scenari che su quello di analisi dei requisiti (RAD). Il documento a cui facciamo riferimento è quindi:
RAD 2.0;

3. Panoramica del sistema

Il sistema ha come obiettivo la creazione di un sito di vendita di T-shirt.

Il suo sviluppo è stato concepito in maniera modulare e pertanto è stato possibile dividerlo in sottosistemi (si veda System Design Document); questo approccio apporta vantaggi dal punto di vista della manutenzione e sarà fondamentale nella fase di testing, in quanto si potrà individuare un errore nel singolo sottosistema effettuando un numero limitato di test. Una volta testati individualmente tutti i sottosistemi andremo a testare la loro integrazione utilizzando un approccio di tipo "Bottom-up". Il sottosistema a livello gerarchico più basso sarà testato individualmente. Successivamente, sarà testata la sua integrazione con la componente a livello di gerarchia più alto. Questo procedimento sarà iterato finché non avremo integrato tutti i sottosistemi per verificare che l'intero sistema funzioni correttamente.

4. Funzionalità da testare e non

Le componenti prese in considerazione nella fase di testing rappresentano le funzionalità core del sistema, ovvero:

● **Gestione Account**

Questa funzionalità permette ad un utente di effettuare l'accesso e di fare il logout dal sistema; permette agli admin, invece, di autenticarsi ed accedere alle funzionalità a loro consentite.

➤ Saranno testate le funzionalità di login.

● **Gestione Carrello e Checkout**

Tale gestione consente di visualizzare, inserire ed eliminare dati relativi agli articoli presenti nel carrello e procedere successivamente al checkout per terminare l'acquisto.

➤ Saranno testate le funzionalità di inserimento di un nuovo elemento, della sua eliminazione e del procedimento per la terminazione dell'acquisto.

● **Gestione Articoli**

Tale gestione consente di visualizzare, inserire, modificare ed eliminare dati relativi alle T-shirt.

➤ Sarà testata la funzionalità di inserimento delle quantità da parte di un utente di una determinata T-shirt.

● **Gestione Acquisti**

Tale gestione consente di visualizzare ed evadere gli ordini in entrata ed in uscita.

➤ Sarà testata la funzionalità di visualizzazione dei dettagli di un ordine e della corretta evasione di un ordine.

● **Gestione Dipendenti**

Tale gestione consente di visualizzare, inserire, modificare ed eliminare dati relativi ai dipendenti.

➤ Sarà testata la funzionalità di inserimento, di eliminazione e di modifica dei dati di un dipendente.

5. Pass/Fail criteria

Il testing ha successo se l'output osservato è diverso dall'output atteso: ciò significa che la fase di testing avrà successo se individuerà un fallimento. In tal caso questa verrà analizzata e, se

legata ad un fault, si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica non abbia impattato su altri componenti del sistema. Al contrario, il testing fallirà se l'output osservato sarà uguale all'oracolo.

6. Approccio

Nella sessione di testing del sistema verrà utilizzato un approccio di tipo “BLACK BOX”, che

prevede che i test vengano effettuati in maniera da non scendere nei dettagli del codice,

ma basandosi sulle specifiche delle funzionalità da testare.

L'approccio alla fase di testing si compone di tre fasi:

- Testing di unità, che controlla i singoli componenti (classi, metodi)
- Testing di integrazione, che va a testare l'integrazione dei vari sottosistemi
- Testing di sistema: test funzionale, che andrà a verificare la funzionalità dell'intero sistema assemblato.

6.1 Testing di unità

Con il testing di unità verrà effettuato un controllo delle varie classi e metodi del

sistema, quindi saranno ricercate le condizioni di fallimento andando ad evidenziare gli

errori. Il testing di unità, sarà eseguito dal team di sviluppo attraverso l'implementazione di classi di test utilizzando il framework JUnit. In particolare, per

ogni classe che esegue operazioni complesse sarà sviluppata la relativa classe JUnit.

6.2 Testing di integrazione

Con il testing di integrazione si effettuerà un controllo sull'integrazione delle varie

componenti del sistema. Si adotterà una strategia di tipo “Bottom-up”.

Per effettuare

questi test di integrazione, spesso saranno necessari l'utilizzo di driver dato che tale

strategia va ad integrare passo passo i sottosistemi partendo dal layer che si trova più in

basso nella scala gerarchica. La lista dei casi di test verrà fuori dall'applicazione del

metodo del Category Partition.

6.3 Testing di sistema

Con il testing di sistema verrà effettuato un controllo della correttezza dell'intero

sistema. E' da considerare il testing più critico, in quanto può risultare molto complesso

andare alla ricerca di eventuali errori, essendo impegnati tutti i sottosistemi. Questo test

sarà effettuato utilizzando il framework Selenium, che mette a disposizione strumenti per il controllo di sistemi web-based.

7. Test cases

Per sviluppare i test cases sarà utilizzato il metodo del Category Partition. Questo metodo consiste nell'identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore. I test cases verranno definiti nel documento di Test Cases Specification (TCS).

8.2 Gestione Autenticazione

8.2.1 Login

<p>Parametro -> Email</p> <p>Formato fe: (?:[a-z0-9!#\$%&'*/+=?^_`{ }~-]+(?:\.[a-z0-9!#\$%&'*/+=?^_`{ }~-]+)*)\"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f] \\"[\x01-\x09\x0b\x0c\x0e-\x7f])*)\"@(?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])? \\[(?:(?:(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\\.){3}(?:25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?) [a-z0-9-]*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f] \\"[\x01-\x09\x0b\x0c\x0e-\x7f])+)\\])</p> <p>Categorie</p>	<p>Scelte</p>
---	----------------------

Lunghezza Email	1: lunghezza < 5 [errore]
formato email	1: rispetta il formato [if lunghezza<5] e rispetta il formato fe 2: non rispetta il formato [if lunghezza<5] o non rispetta il formato fe [errore]
Parametro -> Password Categorie	Scelte
Lunghezza lp	1: lunghezza < 8 [errore] 2: lunghezza 8-16 [ok]
Esiste ee	1: esiste nel DB [ok] 2: non esiste nel DB [errore]

Codice	Combinazione	Esito
TC_Login_01	le2ee1lp2pp1	Accesso
TC_Login_02	le2ee2	Errore
TC_Login_03	le2ee1lp2pp2	Errore
TC_Login_04	le2ee2lp2pp2	Errore

8.2 Gestione Dipendente

8.2.2 Inserimento Dipendente

Parametro -> Nome Formato [A-Z,a-z] Categorie	Scelte
Lunghezza ln	1: lunghezza < 2 [errore] 2: lunghezza 2-25 [ok] 3: lunghezza > 25 [errore]
Formato fn	1: rispetta il formato [A-Z,a-z]

	[ok]
	2: non rispetta il formato [errore]

Parametro -> Cognome Formato [A-Z,a-z] Categorie	Scelte
Lunghezza lc	1: lunghezza < 2 [errore]
	2: lunghezza 2-25 [ok]
	3: lunghezza > 25 [errore]
Formato fc	1: rispetta il formato [A-Z,a-z] [ok]
	2: non rispetta il formato [errore]

Parametro -> Username Formato [A-Z,a-z,0-9] Categorie	Scelte
Lunghezza lu	1: lunghezza < 2 [errore]
	2: lunghezza 2-25 [ok]
	3: lunghezza > 25 [errore]
Formato fu	1: rispetta il formato [A-Z,a-z,0-9] [ok]
	2: non rispetta il formato [errore]

Parametro -> Password Formato [A-Z,a-z,0-9] Categorie	Scelte
Lunghezza lp	1: lunghezza < 8 [errore]
	2: lunghezza 8-16 [ok]
	3: lunghezza > 16 [errore]
Formato fp	1: rispetta il formato [A-Z,a-z,0-9] [ok]
	2: non rispetta il formato [errore]

Codice	Combinazione	Esito
TC_InsDipendente_01	ln2fn1lc2fc1lu2fu1lp2fp1	Accesso

TC_InsDipendente_02	ln2fn1lc2fc1lu2fu1lp2fp2	Errore
TC_InsDipendente_03	ln2fn1lc2fc1lu2fu2	Errore
TC_InsDipendente_04	ln2fn1lc2fc2	Errore

8.2.3 Inserimento ricambio

Parametro -> Marca	
Formato: [A-Z,a-z,0-9]	
Categorie	Scelte
Lunghezza lm	1: lunghezza < 2 [errore]
	2: lunghezza 2-25 [ok]
	3: lunghezza > 25 [errore]
Formato fm	1: rispetta il formato [A-Z,a-z,0-9] [ok]
	2: non rispetta il formato [errore]

Parametro -> Quantità	
Formato: [0-9]	
Categorie	Scelte
Lunghezza lq	1: lunghezza < 1 [errore]
	2: lunghezza 1-4 [ok]
	3: lunghezza > 4 [errore]
Formato fq	1: rispetta il formato [0-9] [ok]
	2: non rispetta il formato [errore]

Parametro -> Tipo	
Formato: [A-Z,a-z,0-9]	
Categorie	Scelte
Lunghezza lt	1: lunghezza < 2 [errore]
	2: lunghezza 2-25 [ok]
	3: lunghezza > 25 [errore]
Formato ft	1: rispetta il formato [A-Z,a-z,0-9] [ok]
	2: non rispetta il formato [errore]

Parametro -> Prezzo	
Formato: [0-9]	
Categorie	Scelte
Lunghezza lp	1: lunghezza < 1 [errore]
	2: lunghezza 1-4 [ok]
	3: lunghezza > 4 [errore]
Formato fp	1: rispetta il formato [0-9] [ok]
	2: non rispetta il formato [errore]

Codice	Combinazione	Esito
TC_InsRicambio_ 01	lm2fm1lq2fq1lt2ft1lp2fp1	Accesso
TC_InsRicambio_ 02	lm2fm1lq2fq1lt2ft1lp2fp2	Errore
TC_InsRicambio_ 03	lm2fm1lq1fq1lt2ft1	Errore
TC_InsRicambio_ 04	lm2fm1lq2fq2	Errore