
Piano di Testing del Sistema

per
EC-18

Versione 1.0

Autori:

L. Maddaluno

R. Meo

M. Vitiello

26/10/2019

blank page

Indice

3	Testing del Sistema	4
	3.1	Test Plan	4
	3.2	Unit Test	9

Data	Versione	Autore	Descrizione
26/10/2019	1.0	L. Maddaluno, R. Meo, M. Vitiello	Release iniziale

3 Testing del Sistema

Questa sezione descrive i tipi di test che sono stati eseguiti sul sistema sia periodicamente che una tantum. Ogni volta che viene effettuato un cambiamento sulla repository viene eseguita una serie di test per verificare che questi cambiamenti non abbiano introdotto errori o che possano aver portato il sistema al non soddisfare i requisiti richiesti. I test controllano i *requisiti funzionali* del sistema.

3.1 Test Plan

Test ID	1	
Nome	Registrazione	
Descrizione	Test per verificare la corretta registrazione al sistema	
Input	Risultato atteso	Risultato ottenuto
Viene premuto il tasto continua dopo aver compilato tutti i campi	<i>Registrazione effettuata</i> , viene mostrata la pagina di avvenuta registrazione	
Viene premuto il tasto continua senza aver compilato tutti i campi in modo corretto	<i>Registrazione fallita</i> , viene mostrato un messaggio di errore sui campi errati	
Viene premuto il tasto continua ma la mail è già presente nel database	<i>Registrazione fallita</i> , viene mostrato un messaggio di errore che notifica la presenza della mail nel database	
Location	Sito Web	
Note		

Test ID	2	
Nome	Registrazione	
Descrizione	Test per verificare la corretta registrazione al sistema	
Input	Risultato atteso	Risultato ottenuto
Viene premuto il tasto Registrati dopo aver compilato tutti i campi	<i>Registrazione effettuata</i> , viene mostrata la pagina di HomePage	
Viene premuto il tasto Registrati senza aver compilato tutti i campi in modo corretto	<i>Registrazione fallita</i> , viene mostrato un messaggio di errore su uno o più campi vuoti	
Viene premuto il tasto Registrati ma la mail è già presente nel database	<i>Registrazione fallita</i> , viene mostrato un messaggio di errore che notifica la presenza della mail nel database	
Location	App	
Note		

Test ID	3	
Nome	Login	
Descrizione	Test per verificare il corretto login	
Input	Risultato atteso	Risultato ottenuto
Viene premuto il tasto continua dopo aver compilato tutti i campi	<i>Login effettuato</i> , viene mostrata la pagina di benvenuto	
Viene premuto il tasto continua senza aver compilato tutti i campi in modo corretto	<i>Login fallito</i> , viene mostrato un messaggio di errore sui campi errati	
Location	Sito Web	
Note		

Test ID	4	
Nome	Login	
Descrizione	Test per verificare il corretto login	
Input	Risultato atteso	Risultato ottenuto
Viene premuto il tasto continua dopo aver compilato tutti i campi	<i>Login effettuato</i> , viene mostrato un messaggio di avvenuto login e viene visualizzata la pagina HomePage	
Viene premuto il tasto continua senza aver compilato tutti i campi in modo corretto	<i>Login fallito</i> , viene mostrato un messaggio di errore dei campi inseriti	
Location	App	
Note		

Test ID	5	
Nome	Ricerca prodotto	
Descrizione	Test per verificare la corretta ricerca di un prodotto	
Input	Risultato atteso	Risultato ottenuto
Viene inserito una parola chiave per la ricerca e viene avviata la ricerca	<i>Ricerca avvenuta con successo</i> , vengono mostrati i prodotti che soddisfano la richiesta	
Viene inserito una parola chiave per la ricerca e viene avviata la ricerca	<i>Ricerca fallita</i> , nessun prodotto soddisfa la richiesta e viene mostrato un messaggio d'informazione	
Location	Sito Web, App	
Note		

Test ID	6	
Nome	Aggiungi al carrello	
Descrizione	Test per verificare la corretta creazione del carrello	
Input	Risultato atteso	Risultato ottenuto
Viene premuto il tasto aggiungi al carrello scegliendo il colore e la spedizione	<i>Carrello creato</i> , l'icona del carrello, nella barra di stato, ci segnala la presenza di uno o più prodotti nel carrello	
Viene premuto il tasto aggiungi al carrello senza aver scelto il colore o la spedizione	<i>Carrello non creato</i> , ci viene mostrato un messaggio di errore che notifica l'assenza di alcuni campi	
Location	Sito Web, App	
Note	L'utente deve essere loggato	

Test ID	7	
Nome	Acquisto prodotto	
Descrizione	Test per verificare il corretto acquisto del prodotto	
Input	Risultato atteso	Risultato ottenuto
Viene premuto il tasto PayPal e il pagamento avviene con successo	<i>Acquisto effettuato</i> , viene creato l'ordine e viene inviata la fattura via mail	
Viene premuto il tasto PayPal e il pagamento non avviene con successo	<i>Acquisto fallito</i> , non viene creato alcun ordine e viene mostrata la pagina precedente	
Location	Sito Web, App	
Note	Il carrello non deve essere vuoto	

Test ID	8	
Nome	Lascia feedback	
Descrizione	Test per verificare il corretto rilascio di un feedback	
Input	Risultato atteso	Risultato ottenuto
Viene premuto il tasto Salva dopo aver selezionato una valutazione	<i>Feedback lasciato</i> , viene mostrato un messaggio d'informazione	
Viene premuto il tasto Salva senza aver selezionato alcuna valutazione	<i>Feedback non lasciato</i> , viene mostrata la pagina del rilascio feedback	
Location	Sito Web, App	
Note	L'utente deve aver effettuato ordini	

Test ID	9	
Nome	Aggiungi prodotto	
Descrizione	Test per verificare il corretto funzionamento dell'aggiunta di un prodotto	
Input	Risultato atteso	Risultato ottenuto
Vene premuto il tasto Aggiungi prodotto dopo aver compilato tutti i campi obbligatori	<i>Prodotto aggiunto</i> , viene mostrato il messaggio di corretta aggiunta del prodotto	
Vene premuto il tasto Aggiungi prodotto senza aver compilato tutti i campi obbligatori	<i>Prodotto non aggiunto</i> , viene mostrato un messaggio di errore sui campi errati	
Location	Sito Web	
Note		

Test ID	10	
Nome	Modifica prodotto	
Descrizione	Test per verificare il corretto funzionamento della modifica del prodotto	
Input	Risultato atteso	Risultato ottenuto
Vene premuto il tasto Conferma modifiche dopo aver compilato tutti i campi obbligatori	<i>Prodotto modificato</i> , viene mostrato il messaggio di corretta modifica del prodotto	
Vene premuto il tasto Conferma modifiche senza aver compilato tutti i campi obbligatori	<i>Prodotto non modificato</i> , viene mostrato il messaggio di errore sui campi errati	
Location	Sito Web	
Note	Il prodotto deve esistere nel database	

Test ID	11	
Nome	Elimina prodotto	
Descrizione	Test per verificare il corretto funzionamento dell'eliminazione del prodotto	
Input	Risultato atteso	Risultato ottenuto
Vene premuto il tasto Elimina prodotto	<i>Prodotto eliminato</i> , viene mostrato il messaggio di corretta eliminazione del prodotto	
Vene premuto il tasto Elimina prodotto	<i>Prodotto non viene eliminato</i> , viene mostrata la pagina precedente	
Location	Sito Web	
Note	Il prodotto deve esistere nel database	

Test ID	12	
Nome	Elimina prodotto	
Descrizione	Test per verificare il corretto funzionamento della visualizzazione delle statistiche	
Input	Risultato atteso	Risultato ottenuto
Vene premuto il tasto Vendite	<i>Visualizzazione vendite</i> , viene mostrato il grafico delle statistiche riguardanti le vendite	
Vene premuto il tasto Incassi	<i>Visualizzazione degli incassi</i> , viene mostrato il grafico delle statistiche riguardanti gli incassi	
Vene premuto il tasto Più cliccati	<i>Visualizzazione dei prodotti più cliccati</i> , viene mostrato il grafico delle statistiche riguardanti gli articoli più cliccati	
Vene premuto il tasto Feedback	<i>Visualizzazione dei feedback</i> , viene mostrato il grafico delle statistiche riguardanti i feedback rilasciati	
Location	Sito Web	
Note		

3.2 Unit Test

Gli *unit test* vengono utilizzati per verificare il corretto funzionamento dei componenti individuali quando sono isolati dal resto del sistema. I metodi testati riguardano la validità dei dati inseriti nella fase di login e il corretto calcolo del prezzo totale di un ordine.

```
1. class isValidTest {
2.
3.     @Test
4.     void testWithNullMailNullPassword() {
5.         System.out.println("testWithNullMailNullPassword");
6.         String mail=null;
7.         String password=null;
8.         assertThrows(NullPointerException.class, () -> {
9.             new CustomerDAO().isValid(mail, password);
10.        });
11.    }
12.
13.
14.    @Test
15.    void testWithWrongMailWrongPassword() {
16.        System.out.println("testWithWrongMailWrongPassword");
17.        String mail="admin@gmail.com";
18.        String password="12345";
19.        assertEquals(new CustomerDAO().isValid(mail, password), false, "Dovrebbe essere
restituito false");
20.    }
21.
22.    @Test
23.    void testWithRightMailWrongPassword() {
24.        System.out.println("testWithRightMailWrongPassword");
25.        String mail="admin@ec18";
26.        String password="12345";
27.        assertEquals(new CustomerDAO().isValid(mail, password), false, "Dovrebbe essere
restituito false");
28.    }
29.
30.    @Test
31.    void testWithWrongMailRightPassword() {
32.        System.out.println("testWithWrongMailRightPassword");
33.        String mail="admin@gmail.com";
34.        String password="adminec18";
35.        assertEquals(new CustomerDAO().isValid(mail, password), false, "Dovrebbe essere
restituito false");
36.    }
37.
38.    @Test
39.    void testWithRightMailRightPassword() {
40.        System.out.println("testWithRightMailRightPassword");
41.        String mail="admin@ec18";
42.        String password="adminec18";
43.        assertEquals(new CustomerDAO().isValid(mail, password), true, "Dovrebbe essere
restituito true");
44.    }
45.
46. }
```

```
1.  class getTotalPriceTest {
2.
3.      @Test
4.      void test() {
5.
6.          Cart cart=new Cart();
7.
8.          Item mockItem1=new Item();
9.          mockItem1.setPrice(5.0);
10.         mockItem1.setOnSale(false);
11.
12.         Item mockItem2=new Item();
13.         mockItem2.setPrice(10.1);
14.         mockItem2.setOnSale(false);
15.
16.         Item mockItem3=new Item();
17.         mockItem3.setPrice(15.0);
18.         mockItem3.setOnSale(false);
19.
20.         cart.setProducts(mockItem1);
21.         cart.setProducts(mockItem2);
22.         cart.setProducts(mockItem3);
23.
24.         cart.setShipping(0);
25.
26.         assertEquals(cart.getTotalPrice(), Double.toString(30.1));
27.     }
28.
29. }
```