



Università degli Studi di Salerno

Corso di Ingegneria del Software

JapanWorld Test Case Specification Versione 1.1



Data: 17/08/2020

Progetto: JapanWorld	Versione: 1.1
Documento: Test Case Specification	Data: 13/07/2021

Coordinatore del progetto:

Nome	Matricola
Emanuele Patella	0512104730

Partecipanti:

Nome	Matricola
Emanuele Patella	0512104730

Scritto da:	Emanuele Patella
--------------------	------------------

Revision History

Data	Versione	Descrizione	Autore
07/07/2021	1.0	Test Plan Versione 1	Emanuele Patella
13/07/2021	1.1	Revisione Documento	Emanuele Patella

Indice

1. Introduzione.....	4
2. Suddivisione casi di Testing.....	4
3. Test in JUnit: Controlli e Query associate.....	4
3.1. Test relativi al controller Prodotti.....	5
3.2. Test relativi al controller Carrello.....	7
3.3. Test relativi al controller ProfiliUtente.....	9
4. Test Selenium: Interfaccia e Model annessi.....	10
4.1. TestHomePageStart.....	11
4.2. TestInterfacciaRicerca.....	11
4.3. TestLoginLogout.....	11
4.4. TestLoginLogoutAdmin.....	11
4.5. TestRegistrazione.....	11
4.6. TestAcquistoNonLoggato.....	12
4.7. TestAcquisto.....	12

1. Introduzione

Il nostro sistema entra adesso, dopo i mesi di duro lavoro, nella fase di testing, quella che ci mostrerà se il lavoro è stato eseguito correttamente o meno.

All'interno di questo documento, partendo da tutte le decisioni e le specifiche riportare all'interno del documento di Test plan, verranno illustrati i singoli casi di test e i risultati che ci si aspetta da tali test.

Tutti i risultati, insieme ad alcune considerazioni finali e i relativi dettagli, verranno poi riportati all'interno dell'ultimo documento che verrà prodotto, il Test Execution Report.

2. Suddivisione casi di testing

I casi di testing verranno divisi in due macroaree, quella dei Test in JUnit, per i controller e le query, e quella in Selenium per l'interfaccia e per i model.

Entrambe le due tipologie di testing verranno riportate di seguito a questo paragrafo, con giustificazioni relative al tipo di test effettuato, risultato atteso, e funzionalità verificate dal test in questione.

Nota: *Tra i casi di testing in Selenium non è presente un test relativo all'aggiunta di un nuovo prodotto al database in quanto, per problemi relativi alla fase di sviluppo, non è stata creata l'interfaccia necessaria all'aggiunta del prodotto in questione.*

3. Test in JUnit: Controller e Query associate

I Test che verranno effettuati in JUnit sono tutti e soli quelli che si trovano ad interagire con le componenti del Controller e le Query in SQL ad essi associate.

Questi test verranno strutturati in singoli documenti JUnit che andranno a testare, ognuno, una componente specifica nella sua totalità.

I metodi testeranno poi ogni funzione, che sia essa di input, output o aggiornamento per i 3 tipi di input che il software potrebbe ricevere, quindi:

- **Input nullo**, quasi impossibile che si verifichi, poiché l'interfaccia impedisce che i campi necessari alle interazioni significative vengano lasciati vuoti, ma il test è comunque necessario e verrà effettuato. Ci si aspetta che il valore di ritorno, sia una Stringa vuota nella quasi totalità dei casi, poiché il 95% circa degli input e output del sito sono stringhe;
- **Input errato**, l'input con cui si dovrebbero avere il maggior numero di interazioni, principalmente scaturito da errori di battitura, mancate doppie, correttore automatico del cellulare e affini. Anche in questo caso ci si aspetta, come valore di output atteso, una Stringa vuota, poiché non ci saranno valori corrispondenti all'interno del database;
- **Input corretto**, cioè l'input atteso dal software. In questo specifico caso la Stringa attesa come output è l'elemento o la pagina cercati;

3.1. Test relativi al controller Prodotti

Il file di testing JUnit relativo al controller *Prodotti* prenderà il nome di *TestProdotti* (*tutti i file di testing avranno il nome composto dalla parola test seguita dal nome della componente che vanno a testare, di modo da renderli facilmente riconoscibili*) e si concentrerà sul testare i metodi interni del controller. I metodi testati sono:

- Ricerca: che si occupa di effettuare la ricerca attraverso l'utilizzo di un unico parametro. È la funzione che viene richiamata quando l'utente effettua una ricerca attraverso la barra di ricerca del sito web. I 3 test effettuati sono:
 - RicercaNulla: La quale verifica il valore di ritorno della funzione qualora l'input della ricerca sia nullo. Il valore atteso è la stringa "[]";
 - RicercaSbagliata: Verifica che il valore di ritorno della funzione sia "[]" qualora l'input della ricerca non sia un parametro presente nel database;
 - RicercaVera: Verifica che, qualora l'input sia un parametro presente nel database, venga restituita la stringa "[elementi

corrispondenti al parametro di ricerca nella tabella]”.

Per il testing verrà usata una riga della tabella preparata appositamente, "[{'immagine':'image/test.jpg', 'nome':'test', 'quantita':'1', 'prezzo':'1'},]"

- RicercaKey: Questa funzione ha un utilizzo molto specifico, in quanto ricerca i prodotti in base al loro codice Id e all'utente che li ha acquistati, in quanto è la funzione utilizzata per la visualizzazione dei prodotti già acquistati e serve a permetterne la visualizzazione all'interno del profilo utente dell'acquirente. Anche qui, i 3 test effettuati sono:
 - RicercaKeyNulla: Che verifica se la funzione, qualora venga richiamata con parametri nulli, restituisca un valore non nullo, in questo caso basta che il valore sia non nullo, in quanto possono essere restituite diverse eccezioni da questa funzione nello specifico;
 - RicercaKeySbagliata: Questa funzione deve, come la gemella soprariportata, verificare che il valore di ritorno sia una Stringa del tipo "[]", poiché indica la non presenza di prodotti nella cronologia degli acquisti di un dato utente;
 - RicercaKeyVera: Quest'ultima, invece, Verifica che il valore di ritorno non sia nullo e che la corrisponda al codiceId che è stato inviato in partenza, il quale viene ritornato dalla funzione come ulteriore controllo di sicurezza;
- RicercaProdotti: Questa funzione è quella utilizzata dai link nella barra di ricerca per andare alle singole pagine dei prodotti, e prende, di conseguenza, due parametri in input, la serie e il genere del prodotto in questione, di modo da poter effettuare la query in maniera corretta. I 3 test che abbiamo svolto sono:
 - RicercaProdottiNulla: In questo caso, più che negli altri, è quasi impossibile che si verifichi una ricerca con parametri nulli, tuttavia, se dovesse verificarsi, la stringa di ritorno sarebbe semplicemente una Stringa del tipo "[]" in quanto non ci sarebbe il match con alcun prodotto all'interno del database;
 - RicercaProdottiSbagliata: Idem dicasi per la ricerca di prodotti con parametri non corretti, è quasi completamente impossibile che si verifichi una ricerca con parametri non corretti ma, qualora si

verificasse, il sistema torna una Stringa del tipo “[]” per il , mancato match con i prodotti presenti nel database;

- RicercaProdottiVera: Questa ricerca, invece, restituisce, una Stringa del tipo “[Prodotti che matchano i parametri della Query]”, in maniera simile alla funzione di testing RicercaVera, ma con una differenza, per necessità di programmazione e visualizzazione, questa query formatta la Stringa con i doppi apici anziché i singoli. *Esempio*: "[{"immagine": "image/test.jpg", "nome": "test", "quantita": "1", "prezzo": "1"}]";
- RicercaProdottiAd: Questa funzione è adibita ad uso interno del software, in quanto, nonostante prenda parametri in input, non li utilizza e restituisce sempre una stringa contenente tutti i prodotti presenti nel database. La funzione viene richiamata solo e unicamente dall’area personale dell’admin per poter tenere traccia delle quantità dei prodotti nel database e aggiornarle attraverso i metodi AggiungiProdotti e RimuoviProdotti. I test, in questo caso, sono solo 2:
 - RicercaProdottiAdNulla: La quale verifica che la stringa restituita sia l’intero database.
 - RicercaProdottiAdNonNulla: Fa la stessa identica cosa di cui sopra.
- AggiungiProdotti: Effettua l’update al database se i campi non sono nulli. Il test è 1:
 - AggiuntaVera: Aggiunge il prodotto al database, se i campi non sono nulli, e verifica che sia in corrispondenza dell’Id dato dall’utente;
- Rimuovi prodotti: Effettua l’update rimuovendo il prodotto tramite il codiceId fornito dall’utente. Il test qui è solo 1, che viene eseguito insieme al test precedente:
 - RimozioneVera: Rimuove un prodotto dal database e verifica che, in corrispondenza di quel codiceId, non ci sia più nessun elemento.

3.2. Test relativi al controller Carrello

Questo controller è delegato alla gestione di tutte le interazioni che avvengono

tra il carrello istanziato nella sessione corrente e il database del sito web.

I metodi che verranno testati sono:

- InserisciProdottoCarrello: Che inserisce il prodotto nel carrello e restituisce il totale corrente. Il metodo di test è:
 - TestinserisciProdottoCarrello: e verifica semplicemente che, a operazione eseguita, il totale sia positivo, e sia quello atteso;
- Acquisto: Che prende in input un prodotto e si occupa di ridurne la quantità disponibile nel database e aggiungerlo al carrello della sessione corrente. I test che verranno effettuati sono 2:
 - TestAcquistoSchedaNulla: Verifica che il metodo restituisca una Stringa vuota, corrispondente al carrello corrente, in cui non sono stati aggiunti prodotti;
 - TestAcquistoProdottoEffettivo: Verifica che la Stringa restituita dal metodo sia esattamente il codiceId del prodotto che si è acquistato;
- Visualizza: Metodo che si occupa di permettere la visualizzazione dei prodotti già venduti ufficialmente. Questa funzione è utilizzabile solo tramite la pagina dell'admin, in quanto serve per visualizzare tutti i prodotti venduti e tenere traccia della contabilità. Il metodo di test qui è solo 1:
 - TestVisualizza: Verifica che la stringa fornita da questo metodo sia non nulla. Non viene effettuata una verifica della correttezza della stringa poiché, come sarà visibile di seguito, la query della funzione successiva è identica a quella attuale, se non per il fatto che mostra i risultati dei singoli utenti, quindi è molto più comoda per il testing mirato;
- VisualizzaUtente: Che si occupa di visualizzare i prodotti acquistati dai singoli utenti. Questo metodo può essere utilizzato solo dagli utenti all'interno della propria area utente, e serve a permettere loro di tenere traccia dei loro acquisti. La funzione usa un metodo in sessione per ottenere direttamente il nome utente dell'utente che la richiama, quindi non ci sono rischi per i dati degli altri utenti. I test sono 2.
 - TestVisualizzaUtenteEffettivo: Semplicemente si controlla che il risultato del metodo corrisponda ai prodotti acquistati dall'utente

in questione;

- TestVisualizzaUtenteNullo: Si verifica che non venga restituito alcun dato qualora il metodo venisse richiamato senza il parametro utente, e fosse quindi stato fatto partire in maniera non consona, tuttavia è una situazione che difficilmente si venga a verificare.

3.3. Test relativi al controller ProfiliUtente

Questo controller gestisce tutte le operazioni di Login, Registrazione e Logout degli utenti. I metodi che sono stati testati sono:

- Utente: Metodo che interagisce con quello precedente(VisualizzaUtente) per effettuare la visualizzazione della cronologia acquisti. I test sono 3:
 - TestUtenteNullo: Quando il valore utente non è settato, il valore di ritorno è una Stringa vuota;
 - TestUtenteSbagliato: Quando il valore utente è diverso da quello che la sessione ha memorizzato(Caso quasi impossibile) oppure l'utente non ha ancor acquistato nulla, quindi la sua cronologia acquisti è vuota. Il valor di ritorno, come sopra, è una Stringa vuota;
 - TestUtenteVero: Tutto è andato a buon fine e ci sono prodotti ordinati da questo utente. Il valore di ritorno è esattamente una Stringa con, al suo interno, tutti i prodotti acquistati dall'utente finora. Per acquistati si intende che sono stati effettivamente pagati e spediti, non anche quelli presenti nel carrello attualmente;
- Login: Metodo che gestisce tutte le interazioni tra il database e gli utenti quando vogliono loggare nel loro profilo utente. I metodi di testing qui sono 4, per via di un caso particolare:
 - TestLoginNullo: Quando non vengono inserite credenziali e si clicca direttamente sul bottone di Submit. La Striga che viene tornata è una stringa vuota;
 - TestLoginSbagliato: Quando si sbaglia ad inserire le proprie credenziali oppure non si ci è ancora registrati. Il valore di ritorno è una Stringa con testo "NoUtente" per distinguerlo dalla

situazione precedente;

- TestLoginVeroNormale: Quanto un utente generico cerca di loggare nel proprio profilo. Viene restituita una Stringa con testo “utente”, per segnalare che si sta interagendo con un utente standard;
- TestLoginVeroAdmin: Quando è l’Admin del sito ad effettuare il Login. Il valore di ritorno atteso è sempre una Stringa con, come testo, “admin”, che permette di reindirizzare l’admin sulla pagina utente dedicata e non su quelle degli utenti standard;
- Iscrizione: Che si occupa di aggiungere i nuovi utenti al database quando inseriscono i loro dati. I test sono 3:
 - TestIscrizioneNullo: Quando vengono passati a database dei campi nulli. Il risultato atteso è una Stringa vuota;
 - TestIscrizioneSbagliato: Quando vengono inseriti dei dati che corrispondono ad un utente già iscritto o che non corrispondono ai dati concessi. Il valore atteso è un Booleano di tipo False;
 - TestIscrizioneVero: Quando i dati inseriti sono validi per l’iscrizione. Il valore atteso è sempre un Booleano ma di tipo True.

4. Test Selenium: interfaccia e model annessi

Il testing dell’interfaccia svolto con Selenium si concentrerà sul verificare che tutti i processi possibili che ci si aspetta di avere in input vengano svolti in maniera corretta senza causare problemi al software e senza intaccare la sicurezza dei dati trattati dal sistema.

Le funzionalità che andranno testate sono:

- L’avvio del sito dalla sua Home Page;
- La ricerca di un prodotto;
- L’acquisto di un prodotto;
- Il Login e il Logout di un utente qualsiasi;

- Il Login e il Logout dell'Admin;
- La Registrazione di un nuovo utente.

Per alcuni di questi test verranno create più versioni per essere sicuri di coprire la totalità degli output possibili.

4.1. TestHomePageStart

Questo test si occupa semplicemente di verificare che, all'immissione del url del sito nella barra di ricerca, la pagina venga caricata correttamente.

L'output atteso è che la pagina si carichi correttamente e sia interagibile.

4.2. TestInterfacciaRicerca

Questo test effettua la ricerca di diversi prodotti sia attraverso i menù a tendina forniti dal sito, che attraverso la barra di ricerca che il sistema offre.

Il risultato atteso è che vengano visualizzate le pagine di ricerca corrispondenti alla ricerca effettuata.

4.3. TestLoginLogout

Questo test verifica che l'utente generico possa effettuare il login correttamente, accedendo alla sua area utente e potendo poi, quando lo desidera, effettuare il Logout.

Il risultato atteso è che il sistema permetta l'accesso all'area utente e che, subito dopo l'avvenuta verifica dell'accesso, permetta il Logout.

4.4. TestLoginLogoutAdmin

Questo test verifica che l'Admin possa effettuare il login correttamente, accedendo alla sua area utente e non a quella dell'utente generico. Potendo poi, quando lo desidera, effettuare il Logout.

Il risultato atteso è che il sistema permetta l'accesso all'area utente e che permetta, subito dopo l'avvenuta verifica dell'accesso, il Logout.

4.5. TestRegistrazione

Questo test verifica che il processo di Registrazione vada a buon fine.

Il test consisterà nell'effettuare la registrazione con un nuovo account e poi effettuare l'accesso per essere sicuri che tutto sia andato a buon fine.

Il risultato atteso è che l'utente riesca ad effettuare l'accesso correttamente.

4.6. TestAcquistoNonLoggato

Questo test serve a verificare che, se l'utente cerca di effettuare un acquisto senza essere loggato, viene reindirizzato alla pagina di Login invece di procedere con le pratiche dell'acquisto.

Il risultato atteso è di trovarsi sulla pagina di Login dopo aver cercato di effettuare un acquisto.

4.7. TestAcquisto

Questo test è il più lungo e complesso, ma deve testare la procedura più importante del nostro e-commerce, cioè la procedura di acquisto.

Il test effettuerà il Login con un profilo di testing, per poi provare a comprare un prodotto, effettuandone la ricerca, aggiungendolo al carrello, e poi procedendo al pagamento.

L'output atteso da questo test è quello di riuscire ad arrivare alla pagina dello scontrino e di ottenere la fattura di ciò che abbiamo acquistato.