



Test Plan

Versione	1.0
Data	19/01/2019
Destinatario	Prof. Andrea De Lucia
	Giovanni Vassalluzzo Luca Giaffreda Alfonso Luciani
Presentato da	Roberto De Luca



Revision History

Data	Versione	Descrizione	Autori
10/01/2019	0.1	Prima stesura	Luca Giaffreda
19/01/2019	1.0	Revisione finale	Intero Team



Sommario

Revision History	
1. Introduzione	
2. Riferimenti	
3. Oggetti da testare	
3.1. Dettagli per il testing di unità	
3.2. Dettagli per il testing di integrazione	
4. Pass/fail criteria	
5. Approccio	
6. Criteri di sospensione e ripristino	
7. Test case	

1. Introduzione

Lo scopo di questo documento è quello di pianificare il test di unità e di integrazione della piattaforma DressShop: verranno riportate le strategie adottate per il testing, gli strumenti utilizzati e le funzionalità testate. In questa fase saranno testati i bean dell'applicazione, tenendo conto anche del fatto che i possibili dati di input e output sono stati opportunamente suddivisi in categorie, rappresentanti stati validi o non validi per gli stessi; di conseguenza, durante il test ci si è accertati anche della correttezza delle variabili usate.

Successivamente si integrano le varie componenti del sistema e si verifica che il software risulti funzionante. L'approccio utilizzato sarà di tipo Bottom-up, i bean, saranno integrati nel sistema e si simulerà il comportamento dello stesso per svolgere l'integration test.

2. Riferimenti

Il documento fa riferimento alle ultime versioni dei precedenti documenti rilasciati, in particolare:

- DressShop_RAD
- DressShop_SDD
- DressShop_ODD

3. Oggetti da testare

3.1. Dettagli per il testing di unità

I sottosistemi che verranno sottoposti al test di integrazione sono i seguenti:

- Package Utente: verrà effettuato il test del bean `UtenteBean.java`;
- Package Acquisto: verrà effettuato il test dei bean `CartaDiCreditoBean.java`, e `IndirizzoBena.java`
- Package Catalogo verrà effettuato il test dei bean `OrdineBean.java`, `ProdottoBean.java`, `ProdottoInOrdine.java` e `PromozioneBean.java` ;

3.2. Dettagli per il testing di integrazione

I sottosistemi che verranno sottoposti al test di integrazione sono i seguenti:

- Package Autenticazione: verrà effettuato il test delle classi `LoginControl.java` e `LogoutControl.java`;
- Package Registrazione: verrà effettuato il test delle classi `RegistratioControl.java`;
- Package Carrello: verrà effettuato il test delle classi `cartControl.java` e `procediAcquisto.java`;
- Package Cliente: verrà effettuato il test delle classi `AggiungiCarta.java` e `RimuoviCarta.java`;
- Package Dipendenti: verrà effettuato il test della classe `MagazziniereControl.java`;

4. Pass/fail criteria

Nel caso in cui dovessero riscontrarsi errori durante la fase di testing, si procederà con la correzione dei fault intervenendo direttamente sulle porzioni di codice che generano il problema, ed iterando nuovamente con la fase di unit testing verificando che la correzione non abbia impattato altre componenti.

Nel caso in cui vengano riscontrati errori durante la fase di testing d'integrazione (e ciò è ammissibile dato che il software non si trova ancora in fase di deployment, e si stanno assemblando le varie componenti), si procederà con la correzione dei fault intervenendo direttamente sulle porzioni di codice che generano il problema.

5. Approccio

L'approccio utilizzato per eseguire il test di integrazione sarà di tipo Bottom-up (dal basso verso l'alto). Ciò comporta il test e l'integrazione delle singole componenti partendo dal livello più basso, ovvero lo Storage, per poi proseguire con l'Application Layer. Avvalendosi di opportuni driver e del software JUnit, verranno testati in una prima fase i bean dei vari packages e, solo dopo averne verificato il corretto funzionamento, si procederà al livello successivo testando i manager. Il tutto ovviamente sarà vincolato ad un database che sarà di appoggio a questa fase di testing. Inoltre, sarà utilizzato Selenium, che si occuperà di testare l'integrazione delle componenti procedendo nello sviluppo.

6. Criteri di sospensione e ripristino

Criteri di sospensione

Comprendono tutti quei casi critici di quando gli errori hanno un impatto dannoso sul progresso dell'attività di testing. Esempi possono essere:

- Crash del database
- Crash del server
- Fallimento di funzionalità interne
- Problemi relativi all'ambiente di sviluppo del testing

Criteri di ripristino

La ripresa del sistema avviene solo quando tali errori vengono risolti, ripartendo dal test case che ha causato l'errore.

7. Test Case

DressShop Prodotto

Nome classe	ProdottoBean.java
Id unit test case	Metodo
DressShop_ProdottoBean_01	testIdProdotto()
DressShop_ProdottoBean_02	testSetCodiceProdotto()
DressShop_ProdottoBean_03	testSetDescrizione()
DressShop_ProdottoBean_04	testSetMarca()
DressShop_ProdottoBean_05	testSetModello()
DressShop_ProdottoBean_06	testSetPrezzo()
DressShop_ProdottoBean_07	testSetInVendita()
DressShop_ProdottoBean_08	testSetIva()
DressShop_ProdottoBean_09	testSetFoto()
DressShop_ProdottoBean_10	testSetCategoria()
DressShop_ProdottoBean_11	testSetPromozione()

DressShop Utente

Nome classe	UtenteBean.java
Id unit test case	Metodo
DressShop_UtenteBean_01	testSetNome()
DressShop_UtenteBean_02	testSetCognome()
DressShop_UtenteBean_03	testSetEmail()
DressShop_UtenteBean_04	testSetPassword()
DressShop_UtenteBean_05	testSetTipo()
DressShop_UtenteBean_06	testSetIdUtente()
DressShop_UtenteBean_07	testSetDataNascita()

DressShop Indirizzo

Nome Classe	IndirizzoBean
Id unit test case	Metodo
DressShop_IndirizzoBean _01	testSetIndirizzo()
DressShop_IndirizzoBean _02	testSetNome()
DressShop_IndirizzoBean _03	testSetCognome()
DressShop_IndirizzoBean _04	testSetCap()
DressShop_IndirizzoBean _05	testSetProvincia()
DressShop_IndirizzoBean _06	testSetCitta()
DressShop_IndirizzoBean _07	testSetCellulare()
DressShop_IndirizzoBean _08	testSetVia()
DressShop_IndirizzoBean _09	testSetUtente()

DressShop ProdottoInOrdine

Nome classe	ProdottoInOrdineBean.java
Id unit test case	Metodo
DressShop_ProdottoInOrdineBean_01	testIdProdotto()
DressShop_ProdottoInOrdineBean_02	testSetReso()
DressShop_ProdottoInOrdineBean_03	testSetPrezzo()
DressShop_ProdottoInOrdineBean_04	testSetIva()
DressShop_ProdottoInOrdineBean_05	testSetQuantita()

DressShop Ordine

Nome classe	OrdineBean.java
Id unit test case	Metodo
DressShop_OrdineBean_01	testSetIdOrdine()
DressShop_OrdineBean_02	testSetData()
DressShop_OrdineBean_03	testSetPagato()
DressShop_OrdineBean_04	testSetCartaDiCredito()
DressShop_OrdineBean_05	testSetIndirizzo()
DressShop_OrdineBean_06	testSetutente()
DressShop_OrdineBean_07	testSetTotale()

DressShop Taglia

Nome classe	TagliaBean.java
Id unit test case	Metodo
DressShop_TagliaBean_01	testSetIdProdotto()
DressShop_TagliaBean_02	testSetTaglia()
DressShop_TagliaBean_03	testSetQuantita()

DressShop Promozione

Nome classe	PromozioneBean.java
Id unit test case	Metodo
DressShop_PromozioneBean_01	testSetIdPromozione()
DressShop_PromozioneBean_02	testSetDataInizio()
DressShop_PromozioneBean_03	testSetDataFine()
DressShop_PromozioneBean_04	testSetSconto()