Università degli Studi di Salerno
Corso di Ingegneria del Software

E-store Libri&Musica

Test Plan
Versione 0.1

Data: 19/11/2016

Progetto: E-store Libri&Musica | Versione: 0.2

Documento: TestPlan | Data: 16/02/2017

Partecipanti:

Nome	Matricola
Fimiani Luca	0512103180

Scritto da: Fimiani Luca

Revision History

Data	Versione	Descrizione	Autore
02/01/2017	0.1	Inizio stesura documento	Fimiani Luca
16/02/2017	0.2	Revisione	Fimiani Luca

INDICE DEGLI ARGOMENTI

INDICE DELLE TABELLE

1. Introduzione

Il testing d'unità è la fase di test durante la quale ci si assicura che le componenti sviluppate funzionino correttamente anche se eseguite in condizioni di isolamento.
Questo documento identifica e descrive la strategia di testing da utilizzare per il presente sistema. Saranno infatti specificate le componenti da testare e il modo in cui il testing dovrà essere eseguito.

2. Relazione con altri documenti

Per verificare la corretta implementazione delle varie componenti del sistema sono stati predisposti dei test cases basati sui requisiti funzionali identificati nella fase di analisi e specifica dei requisiti. Il documento a cui si fa riferimento è "E-StoreLibri&Musica_RAD".

3. Dettagli del Level Test Design

3.1 Approccio di Unit Testing

Il primo modo di valutare le funzionalità di un sistema è quello di verificare che le componenti sviluppate funzionino come dovrebbero nonostante siano isolate dal resto del sistema.

Gli oggetti verranno testati nel seguente ordine :

1. Testing degli oggetti **Entity**
2. Testing degli oggetti **Control**
3. Testing degli oggetti **Boundary (Test non effettuato)**

Per testare gli oggetti Boundary è necessario fare uso di test stub.

Verrà utilizzato un approccio White-Box e si farà uso del framework JUnit per lo sviluppo dei casi di test.

3.2 Componenti da testare

Sono stati testati solo alcuni metodi. Sotto sono riportati i metodi di test, utilizzati in JUnit :

- TestMyOrder()
- TestInsertNewOrder()
- TestGetAllProdotti()
- TestSearchProdottoDaTitolo()
- TestSearchProdottoByCodice()
- TestInsertNewProdotto()

Oggetti Entity
CartBean.java
OrdineBean.java
ProdottiBean.java
UtenteBean.java
OrderDM.java
ProdottiDM.java
UtenteDM.java

Oggetti Control
GestioneOrdineAdmin.java
GestioneProdottoAdmin.java
GestioneUtenteAdmin.java
GestioneProdotti.java
GestioneUtente.java
ImageBuffer.java
OperazioniOrdine.java

4. Pass / Fail Criteria

Lo scopo del testing è di rilevare eventuali errori (faults), ovvero comportamenti inattesi da parte del sistema. Per fare ciò si analizzeranno gli input tenendo conto che si potrà superare il test solamente se l'output non corrisponderà ai risultatati attesi (risultati di fallimento) . Facendo riferimento all'oracolo si intende il risultato che ci si aspetta di ottenere affinchè il test non sia superato. Nel momento in cui l'output non produrrà risultati attesi (failure results) il test avrà fallito e dunque non ci sarà bisogno di analizzare il sottosistema corrispondente.