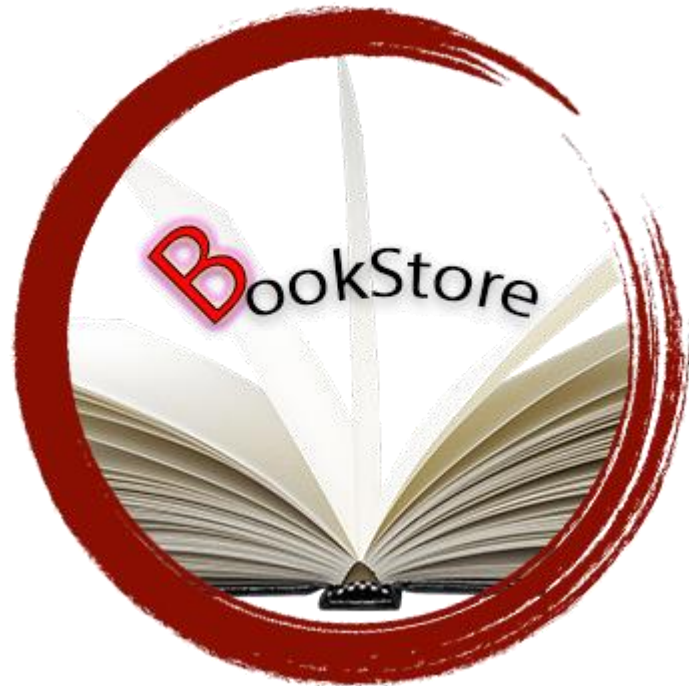




Laurea Magistrale in informatica-Università di Salerno
Corso di Gestione dei Progetti Software- Prof. Andrea DE LUCIA



Test Plan Document

Riferimento	
Versione	1.0
Data	09/03/2021
Destinatario	Prof, Andrea DE LUCIA
Presentato da	Vincenzo Raia (VR) - 0512106140 Raffaele Scarpa (RS) - 0512105708 Giuseppe Pepe (GP) - 0512105930



Revision History

Data	Versione	Descrizione	Autori
20/01/2021	0.1	Prima stesura	VR
09/03/2021	1.0	Revisione finale	VR



Sommario

Revision History	2
1. Introduzione	4
2. Documenti correlati.....	4
2.1 Relazioni con il documento di analisi dei requisiti (RAD)	4
2.2 Relazioni con il System Design Document (SDD).....	4
2.3 Relazioni con l'Object Design Document (ODD).....	4
3.Panoramica del sistema.....	4
4. Funzionalità da testare e da non testare.....	5
5. Criteri Pass/Failed	5
6. Approccio.....	6
6.1 Testing di Unità.....	6
6.2 Testing di Integrazione	6
6.3 Testing del sistema	7
7. Sospensione e ripresa.....	7
7.1 Criteri di sospensione	7
7.2 Criteri di ripresa	7
7.3 Criteri di terminazione	7
8. Materiale per il testing	7
9. Test Case.....	7
9.1 Gestione libro	7
9.1.1 Aggiungi libro.....	7
9.1.2 Cerca Libro	9
9.3 Gestione utente.....	9
9.3.1 Autenticazione.....	9
9.3.2 Modifica informazioni.....	10
9.3.3 Registrazione account	10
9.3.4 Modifica indirizzo	12
9.3.5 Modifica password	13
9.6 Gestione categoria.....	14
9.6.1 Aggiungi categoria	14
10. Riferimenti ad altri documenti di test	14



1. Introduzione

Lo scopo di questo documento è di pianificare l'attività di test del sistema BookStore per verificare se esistono differenze tra il comportamento atteso e il comportamento osservato.

Andremo a rilevare gli eventuali errori prodotti all'interno del codice, per evitare che essi si presentino nel momento in cui il sistema verrà utilizzato dall'utente finale.

Le attività di test sono state pianificate per le seguenti gestioni:

1. Gestione libro
2. Gestione carrello
3. Gestione utente
4. Gestione amministrativa
5. Gestione cliente
6. Gestione categoria

Verranno testate esclusivamente le funzionalità implementate e specificate nell'ODD.

2. Documenti correlati

Il test plan permette di rilevare le eventuali differenze tra il comportamento atteso e quello osservato del sistema.

Di seguito verranno riportate le relazioni tra il test plan e la documentazione precedente:

2.1 Relazioni con il documento di analisi dei requisiti (RAD)

La relazione tra test plan e RAD riguarda in **particolare i requisiti funzionali e non funzionali del sistema**.

2.2 Relazioni con il System Design Document (SDD)

Nel SDD abbiamo suddiviso il nostro sistema in sottosistemi e l'architettura in tre livelli:

- Presentation Layer
- Application Layer
- Storage Layer

Il test dei vari componenti deve rimanere fedele a queste suddivisioni il più possibile.

2.3 Relazioni con l'Object Design Document (ODD)

Il test d'integrazione farà quanto più riferimento possibile alle interfacce delle classi definite nell'ODD.

3. Panoramica del sistema

Il sistema è diviso secondo una architettura "Three Layers" cioè a tre livelli:

- Presentation Layer
- Application Layer
- Storage Layer



Il livello più alto interagisce con il livello applicativo che, a sua volta, si occupa di eseguire le operazioni nel database di BookStore, cercando di garantire il più possibile basso accoppiamento e alta coesione tra le varie classi.

Il sistema inoltre è stato suddiviso in sottosistemi più piccoli, in particolare è stato diviso per gestioni, definite al paragrafo 1 di questo documento.

Praticamente tutte le gestioni prevedono principalmente operazioni di inserimento, modifica, cancellazione, visualizzazione e ricerca, funzionalità che saranno testate nel corso della fase di testing del sistema.

Il sistema che proponiamo prevede tre attori principali:

- Cliente: ha la possibilità ordinare libri e visualizzare i propri ordini
- Responsabile catalogo: ha la possibilità di gestire i libri e le categorie del sito
- Manager: ha la possibilità di gestire gli utenti del sistema e gestire gli ordini dei clienti

4. Funzionalità da testare e da non testare

Di seguito sono elencate per ogni gestione quali sono le funzionalità che saranno testate, per gli stetti tempo del progetto saranno testate solo alcune parti del sistema:

1. Gestione libro
 - 1.1. Aggiungi libro
 - 1.2. Cerca Libro
2. Gestione utente
 - 3.1. Autenticazione
 - 3.2. Modifica informazioni
 - 3.3. Registrazione account
 - 3.4. Modifica indirizzo
 - 3.5. Modifica password
3. Gestione amministrativa
4. Gestione cliente
5. Gestione categoria
 - 6.1. Aggiungi categoria

Non saranno testate invece:

- Interfacce utente
- Sicurezza
- Performance

5. Criteri Pass/Failed

Abbiamo determinato un insieme di input possibili che possano aiutarci a scovare errori nel sistema.



Gli input del test come da prassi verranno suddivisi in classi di equivalenza il che permetterà di testare un solo elemento rappresentativo.

Un input super il test se l'output risultante sarà quello atteso, cioè, quello specificato dal membro del team che si occuperà del test su tale test case che conosce l'output corretto

6. Approccio

La fase di testing si compone di tre attività: una prima fase si occuperà di trovare errori in una singola componente; la seconda fase, invece, avrà come compito quello di testare le funzionalità nate dall'integrazione dei vari sottosistemi e per ultimo andremo a testare l'intero sistema assemblato al fine di verificare soprattutto che esso soddisfi i desideri del cliente.

Nel dettaglio le tecniche di testing adottate sono:

- Testing di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità.
- Testing di integrazione che si focalizzerà sul test delle interfacce delle suddette unità.
- Testing di sistema, che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti. Test utile per verificare che il sistema soddisfi le richieste del committente.

6.1 Testing di Unità

In questa fase si ricercano le condizioni di fallimento, isolando i componenti ed usando test driver e stub driver (implementazioni parziali di componenti che dipendono o da cui dipendono le componenti da testare).

La strategia utilizzata per il testing sarà basata su Black-Box.

Questa strategia si focalizza sul comportamento degli input e degli output, ignorando però la struttura interna della componente.

Al fine di minimizzare il numero di test cases, i possibili input verranno partizionati in classi di equivalenza e per ogni classe verrà selezionato un test case.

Gli stati erronei scovati che comporteranno un fallimento del sistema dovranno essere tempestivamente comunicati agli sviluppatori al fine di correggerli e ripristinare il testing.

Per comunicare i vari fallimenti identificati durante la fase di testing si utilizzerà un test incident report.

6.2 Testing di Integrazione

In questa fase si integreranno delle componenti di una funzionalità che verranno testate con una strategia **Bottom-Up**.

Con questo approccio si implementeranno prima le funzionalità di basso livello e poi si passa alla funzionalità successiva fino ad esaurire le funzionalità implementate.

Quest'approccio mira principalmente a ridurre le dipendenze tra funzionalità differenti e a facilitare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi.



6.3 Testing del sistema

Lo scopo di questa fase di testing è quello di dimostrare che il sistema soddisfi effettivamente i requisiti richiesti e sia, quindi, pronto all'uso. Come per il testing di unità, si cercherà di testare le funzionalità più importanti per l'utente e quelle che hanno una maggiore probabilità di fallimento. Si noti che, come per il testing di unità, si procederà attraverso tecnica Black-Box.

7. Sospensione e ripresa

7.1 Criteri di sospensione

Il testing del sistema verrà sospeso quando si raggiungerà un compromesso valido tra qualità del prodotto e costi dell'attività di testing. L'obiettivo è di portare avanti il testing quanto più possibile nel tempo senza però rischiare di ritardare la consegna finale del progetto.

7.2 Criteri di ripresa

In seguito alle modifiche o correzioni delle componenti che introdurranno errori o fallimenti, i test case verranno sottoposti nuovamente al sistema assicurandosi così di aver risolto effettivamente il problema.

7.3 Criteri di terminazione

Il test si considera terminato quando la totalità dei casi di test somministrati al sistema riporta esito negativo. Come da indicazione del top management, la suddetta condizione sussiste solo se il 75% dei branch sviluppati viene ricoperto in questa fase.

8. Materiale per il testing

- PC
- IDE di programmazione

L'esecuzione dei test necessita di un PC su cui è correttamente configurato Java, MySQL e Apache Tomcat.

La configurazione di MySQL deve avvenire come descritto da DBD.

Il testing è condotto utilizzando alcuni dei framework più famosi ed efficaci in ambienti Java: JUnit.e Selenium

L'utilizzo di JUnit riguarda sia il testing di unità che quello di integrazione, mentre Selenium utilizzato solo la verifica delle precondizioni per le varie operazioni CRUD.

9. Test Case

9.1 Gestione libro

9.1.1 Aggiungi libro

Category partition:

Parametro: ISBN	
Formato: / ^(?=(?:\D*\d){10}{(?:\D*\d){3}}?\$/	
Lunghezza [LI]	1. (<10 OR >13) AND <11,12> [error] 2. (=10 OR =13) [property LI_OK]
Formato[FI]	1. Non rispetta il formato [if LI_OK]



	2. Rispetta il formato [if LI_OK][property FI_OK]
--	---

Parametro: Titolo	
Formato: /^(?=[^\W_])[\w]*\$/	
Lunghezza [LT]	1. <1 OR >100 [error] 2. >=1 OR <=100 [property LT_OK]
Formato[FT]	1. Non rispetta il formato [if LT_OK] 2. Rispetta il formato [if LT_OK][property FT_OK]

Parametro: Autore	
Lunghezza [LA]	3. <1 OR >100 [error] 4. >=1 OR <=100 [property LA_OK]

Parametro: Trama	
Lunghezza [LTR]	1. <1 OR >500 [error] 2. >=1 OR <=500 [property LTR_OK]

Parametro: Data di pubblicazione	
Formato: /^(((1-9) 0[1-9] 12)\d 3[01])-(1-9 0[1-9] 1[0-2])-(12)\d{3})\$/	
Formato[FD]	1. Non rispetta il formato [errore] 2. Rispetta il formato [property FD_OK]

Parametro: Copertina	
Formato: \.(gif jpe?g tiff? png webp bmp)\$/	
Lunghezza[LC]	1. <4 [error] 2. >=4 [property LC_OK]
Formato[FC]	1. Non rispetta il formato [error] 2. Rispetta il formato [if LC_OK] 3. [property FC_OK]

Parametro: Quantità	
Lunghezza[LQ]	1. <1 [error] 2. >=1 [property LQ_OK]

Parametro: Prezzo	
Lunghezza[LP]	3. <1 [error] 4. >=1 [property LP_OK]

Test Cases:

Codice	Combinazione	Esito
TC_1.1_1	LI1	Errore



TC_1.1_2	LI2,FI1	Errore
TC_1.1_3	LI2,FI2,LT1	Errore
TC_1.1_4	LI2,FI2,LT2,FT1	Errore
TC_1.1_5	LI2,FI2,LT2,FT2,LA1	Errore
TC_1.1_6	LI2,FI2,LT2,FT2,LA2LTR1	Errore
TC_1.1_7	LI2,FI2,LT2,FT2,LA2,LTR2,FD1	Errore
TC_1.1_8	LI2,FI2,LT2,FT2,LA2,LTR2 ,FD2,LC1	Errore
TC_1.1_9	LI2,FI2,LT2,FT2,LA2,LTR2 ,FD2,LC2,FC1	Errore
TC_1.1_10	LI2,FI2,LT2,FT2,LA2,LTR2 ,FD2,LC2,FC2,LQ1	Errore
TC_1.1_11	LI2,FI2,LT2,FT2,LA2,LTR2 ,FD2,LC2,FC2,LQ2,LP1	Errore
TC_1.1_12	LI2,FI2,LT2,FT2,LA2,LTR2,FD2,LC2,FC2,LQ2,LP2	Corretto

9.1.2 Cerca Libro

Category partition:

Parametro: Parola chiave	
Formato: /^[A-Za-z0-9 !@#\$%']+\$ /	
Lunghezza[LP]	1. <1 [error] 2. >=1 [property LP_OK]
Formato[FP]	1. Non rispetta il formato [if FP_OK] 2. Rispetta il formato [property FP_OK]

Test Cases:

Codice	Combinazione	Esito
TC1.2_1	LP1	Errore
TC1.2_2	LP2,FP1	Errore
TC1.2_3	LP2,FP2	Corretto

9.3 Gestione utente

9.3.1 Autenticazione

Category partition:

Parametro: Nome utente	
Lunghezza[LN]	1. <1 [error] 2. >=1 [property LN_OK]

Parametro: Password	
Lunghezza[LP]	1. <1 [error] 2. >=1 [property LP_OK]
Match[MP]	1. Match con password = false [error] 2. Match con password = true [if LP_OK] [property MP_OK]

Test Cases:

Codice	Combinazione	Esito
TC3.1_1	LN1	Errore
TC3.1_2	LN2,LP1	Errore
TC3.1_3	LN2,LP2,MP1	Errore
TC3.1_4	LN2,LP2,MP2	Corretto



9.3.2 Modifica informazioni

Category partition:

Parametro: Nome Formato: / [A-Za-z]{2,}\$ /	
Lunghezza[LN]	1. <2 OR >128 [error] 2. >=2 AND <=128 [property LN_OK]
Formato[FN]	1. Non rispetta il formato [if LN_OK] [error] 2. Rispetta il formato [if LN_OK] [property FN_OK]

Parametro: Cognome Formato: / [A-Za-z]{2,}\$ /	
Lunghezza[LC]	1. <2 OR >128 [error] 2. >=2 AND <=128 [property LC_OK]
Formato[FC]	1. Non rispetta il formato [if LC_OK] [error] 2. Rispetta il formato [if LC_OK] [property FC_OK]

Parametro: Email Formato: / [a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}\$ /	
Formato[FE]	1. Non rispetta il formato [if LE_OK] [error] 2. Rispetta il formato [if LE_OK] [property FE_OK]

Test Cases:

Codice	Combinazione	Esito
TC3.2_1	LN1	Errore
TC3.2_2	LN2,FN1	Errore
TC3.2_3	LN2,FN2,LC1	Errore
TC3.2_4	LN2,FN2,LC2,FC1	Errore
TC3.2_5	LN2,FN2,LC2,FC2,FE1	Errore
TC3.2_6	LN2,FN2,LC2,FC2,FE2	Corretto

9.3.3 Registrazione account

Category partition:

Parametro: Password Formato: / ^(?=[0-9])(?=.*[a-z])(?=.*[A-Z]).{8,32}\$ /	
Lunghezza[LP]	1. <8 OR >32 [error] 2. >=8 AND <=128 [property LP_OK]
Formato[FP]	1. Non rispetta il formato [if LP_OK] [error] 2. Rispetta il formato [if LP_OK] [property FP_OK]



Parametro: Nome Formato: / [A-Za-z]{2,}\$/	
Lunghezza[LNO]	3. <2 OR >128 [error] 4. >=2 AND <=128 [property LNO_OK]
Formato[FNO]	3. Non rispetta il formato [if LNO_OK] [error] 4. Rispetta il formato [if LNO_OK] [property FNO_OK]

Parametro: Cognome Formato: / [A-Za-z]{2,}\$/	
Lunghezza[LC]	1. <2 OR >128 [error] 2. >=2 AND <=128 [property LC_OK]
Formato[FC]	1. Non rispetta il formato [if LC_OK] [error] 2. Rispetta il formato [if LC_OK] [property FC_OK]

Parametro: Email Formato: / [a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}\$/	
Formato[FE]	1. Non rispetta il formato [if LE_OK] [error] 2. Rispetta il formato [if LE_OK] [property FE_OK]

Parametro: Via	
Lunghezza[LV]	1. <2 OR >128 [error] 2. >=2 AND <=128 [property LE_OK]

Parametro: Comune Formato: / [A-Za-z]{2,}\$/	
Lunghezza[LCO]	1. <2 OR >128 [error] 2. >=2 AND <=128 [property LCO_OK]
Formato[FCO]	1. Non rispetta il formato [if LCO_OK] [error] 2. Rispetta il formato [if LCO_OK] [property FCO_OK]

Parametro: Provincia Formato: / [A-Za-z]{2,}\$/	
Lunghezza[LP]	1. <2 OR >128 [error] 2. >=2 AND <=128 [property LP_OK]
Formato[FP]	1. Non rispetta il formato [if LP_OK] [error]



	2. Rispetta il formato [if LP_OK] [property FP_OK]
--	--

Parametro: CAP	
Formato: / [0-9]{2,6}\$ /	
Formato[FCA]	1. Non rispetta il formato [if LCA_OK] [error] 2. Rispetta il formato [if LCA_OK] [property FP_OK]

Parametro: Username	
Lunghezza[LN]	1. <1 OR >32[error] 2. >=1 AND <=32[property LN_OK]
Match[MN]	1. Match con nome utente = true [error] 2. Match con nome utente = false [if LN_OK] [property MN_OK]

Test Cases:

Codice	Combinazione	Esito
TC3.3_1	LP1	Errore
TC3.3_2	LP2,FP1	Errore
TC3.3_3	LP2,FP2,LNO1	Errore
TC3.3_4	LP2,FP2,LNO2,FNO1	Errore
TC3.3_5	LP2,FP2,LNO2,FNO2,LC1	Errore
TC3.3_6	LP2,FP2,LNO2,FNO2, LC2,FC1	Errore
TC3.3_7	LP2,FP2,LNO2,FNO2, LC2,FC2,FE1	Errore
TC3.3_8	LP2,FP2,LNO2,FNO2, LC2,FC2,FE2,LV1	Errore
TC3.3_9	LP2,FP2,LNO2,FNO2, LC2,FC2,FE2,LV2,LCO1	Errore
TC3.3_10	LP2,FP2,LNO2,FNO2, LC2,FC2,FE2,LV2,LCO2,FCO1	Errore
TC3.3_11	LP2,FP2,LNO2,FNO2, LC2,FC2,FE2,LV2,LCO2,FCO2,LP1	Errore
TC3.3_12	LP2,FP2,LNO2,FNO2, LC2,FC2,FE2,LV2,LCO2,FCO2,LP2,FP1	Errore
TC3.3_13	LP2,FP2,LNO2,FNO2, LC2,FC2,FE2,LV2,LCO2,FCO2,LP2,FP2,FCA1	Errore
TC3.3_14	LP2,FP2,LNO2,FNO2, LC2,FC2,FE2,LV2,LCO2,FCO2,LP2,FP2,FCA2,LN1	Errore
TC3.3_15	LP2,FP2,LNO2,FNO2, LC2,FC2,FE2,LV2,LCO2,FCO2,LP2,FP2,LCA2,FCA2,LN2,MN1	Errore
TC3.3_16	LP2,FP2,LNO2,FNO2, LC2,FC2,FE2,LV2,LCO2,FCO2,LP2,FP2,LCA2,FCA2,LN2,MN2	Corretto

9.3.4 Modifica indirizzo

Category partition:

Parametro: Via	
Lunghezza[LV]	1. <2 OR >128 [error] 2. >=2 AND <=128 [property LE_OK]

Parametro: Comune



Formato: / [A-Za-z]{2,}\$ /	
Lunghezza[LC]	<ol style="list-style-type: none"> <2 OR >128 [error] >=2 AND <=128 [property LC_OK]
Formato[FC]	<ol style="list-style-type: none"> Non rispetta il formato [if LC_OK] [error] Rispetta il formato [if LC_OK] [property FC_OK]

Parametro: Provincia	
Formato: / [A-Za-z]{2,}\$ /	
Lunghezza[LP]	<ol style="list-style-type: none"> <2 OR >128 [error] >=2 AND <=128 [property LP_OK]
Formato[FP]	<ol style="list-style-type: none"> Non rispetta il formato [if LP_OK] [error] Rispetta il formato [if LP_OK] [property FP_OK]

Parametro: CAP	
Formato: / [0-9]{2,6}\$ /	
Lunghezza[LCA]	<ol style="list-style-type: none"> <2 OR >5 [error] >=2 OR <=5 [property LCA_OK]
Formato[FCA]	<ol style="list-style-type: none"> Non rispetta il formato [if LCA_OK] [error] Rispetta il formato [if LCA_OK] [property FP_OK]

Test Cases:

Codice	Combinazione	Esito
TC3.4_1	LV1	Errore
TC3.4_2	LV2,LC1	Errore
TC3.4_3	LV2,LC2,FC1	Errore
TC3.4_4	LV2,LC2,FC2,LP1	Errore
TC3.4_5	LV2,LC2,FC2,LP2,FP1	Errore
TC3.4_6	LV2,LC2,FC2,LP2,FP2,LCA1	Errore
TC3.4_7	LV2,LC2,FC2,LE2,FE2,LCA2,FCA1	Errore
TC3.4_8	LV2,LC2,FC2,LE2,FE2,LCA2,FCA2,LN1	Errore
TC3.4_9	LV2,LC2,FC2,LE2,FE2,LCA2,FCA2	Corretto

9.3.5 Modifica password

Category partition:

Parametro: Password	
Formato: / (.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}/	
Formato[FP]	<ol style="list-style-type: none"> Non rispetta il formato [error] Rispetta il formato [property FP_OK]

Parametro: Conferma Password	
Match[MC]	<ol style="list-style-type: none"> Match con password = false [error]



	2. Match con password = true [property MC_OK]
--	---

Test Cases:

Codice	Combinazione	Esito
TC3.5_1	FP1	Errore
TC3.5_2	FP2,MC1	Errore
TC3.5_3	FP2,MC2	Corretto

9.6 Gestione categoria

9.6.1 Aggiungi categoria

Category partition:

Parametro: Nome	
Lunghezza[LN]	1. <0 AND >100[error] 2. >=2 AND <=100 [property LN_OK]

Parametro: Descrizione Formato: / [A-Za-z][0-9]{2}\$ /	
Lunghezza[LD]	1. <2 [error] 2. >=2 [property LD_OK]

Test Cases:

Codice	Combinazione	Esito
TC6.1_1	LN1	Errore
TC6.1_2	LN2,LD1	Errore
TC6.1_3	LN2,LD2	Corretto

10. Riferimenti ad altri documenti di test

Le combinazioni di input che verranno somministrate al e i risultati del risultati del testing funzionale condotto tramite Selenium sono definite nel Test Execution Report

Eventuali errori rilevati verranno riportati nel Test Incident Report, mentre il riassunto dei riscontri ottenuti in questa fase verrà proposto tramite il documento di Test Summary Report.