|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

***SocialBook  
Documento Test Plan***



**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Barbato Alessia | 0512105858 |
| Proietto Angelica | 0512105762 |
| Russo Luca | 0512105840 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
|  |  |  |  |

Indice

1.Introduzione 4

2. Relazione con altri documenti 4

2.1 Relazioni con il Requirements Analysis Document 4

2.2 Relazioni con il System Design Document 4

2.3 Relazioni con l’Object Design Document 4

3. Panoramica del sistema 4

4. Features da testare/da non testare 5

5. Pass/fail criteria 6

6. Approccio 6

6.1 Relazioni con il System Design Document 7

6.2 Relazioni con l’Object Design Document 7

7. Sospensione e ripristino 7

7.1 Criteri di sospensione 7

7.2 Criteri di ripristino 7

8. Materiale di testing 7

9. Casi di test 7

10. Testing schedule 12

# Introduzione

Lo scopo di questo documento è l’analisi e la gestione dell’attività di testing riguardante il sistema SocialBook. Viene verificato il corretto funzionamento del sistema analizzando i singoli casi presi dai test e comparando il loro output con il presunto oracolo dei risultati. Da questa comparazione sarà possibile riscontrare gli eventuali errori e problemi da dover correggere con delle opportune modifiche. Il processo di testing viene iterato fino a che non si soddisfano in maniera accettabile i requisiti funzionali e non funzionali del sistema.

# Relazione con gli altri documenti

Per l’individuazione corretta dei casi di test si è fatto riferimento ai documenti precedentemente stilati. Infatti, ogni documento precedente è un punto di partenza indispensabile per poter effettuare un testing corretto ed adeguato e per verificare che il sistema desiderato sia simile a quello proposto.

## Relazioni con il Requirements Analysis Document (RAD)

Al fine di verificare la correttezza di SocialBook sono stati predisposti dei test basati sulle funzionalità individuate nel RAD, in particolare riguardanti i requisiti funzionali e non funzionali del sistema.

## Relazioni con il System Design Document (SDD)

La relazione tra test plan e SDD riguarda in particolare la divisione in sottosistemi. I test sono stati pianificati in base alla suddivisione eseguita durante il system design, con maggior accortezza per il sottosistema “Gestione Utente” in quanto composto dal maggior numero di requisiti funzionali.

## Relazioni con l’Object Design Document (ODD)

La relazione tra test plan e ODD si basa sulle interfacce specificate nell’ODD e delle dipendenze che vi sono tra di esse. In particolare, sono stati predisposti dei test per la verificabilità del funzionamento delle stesse.

# Panoramica del sistema

L’obiettivo del sistema è quello di offrire a tutti gli appassionati di lettura una piattaforma online sicura che permetta all’utente non solo di acquistare libri (in formato digitale), ma anche di organizzarli come meglio si preferisce, recensirli, e comunicare con gli altri utenti.

L’obiettivo della piattaforma è quella di fornire un servizio veloce e sempre aggiornato in modo da garantire l’esperienza migliore possibile agli utenti.

Inoltre la piattaforma sarà completamente gestita da tre categorie di Admin:

-Il System Manager si occupa della manutenzione del codice

-Il Costumer Manager si occupa della gestione degli utenti registrati e delle annesse recensioni

-Il Catalogue Manager si occupa della gestione del catalogo e del singolo libro

# Features da testare/da non testare

## Gestione Utente

● TC\_GU1 Registrazione

● TC\_GU2 Log-in

● TC\_GU3 Creazione Booklist

● TC\_GU4 Modifica Booklist

● TC\_GU5 Elimina Booklist

● TC\_GU6 Recensione prodotto

### NON TESTATI

● Log-out

● Visualizzazione area utente

● Modifiche utente

● Elimina recensione

● Visualizzazione Booklist personali

● Visualizzazione, inserimento e rimozione Preferiti

## Gestione Acquisto

● TC\_GA1 Inserimento prodotto in carrello

● TC\_GA2 Rimozione libro da carrello

● TC\_GA3 Acquisto

### NON TESTATI

● Visualizzazione carrello

● Visualizzazione ordini

## Gestione Visualizzazione

● TC\_GV1 Visualizzazione profilo utente

### NON TESTATI

● Visualizzazione Catalogo

● Ricerca

● Visualizzazione Libro

## Gestione Interazione

● TC\_GI1 Segui utente

### NON TESTATI

● Smetti di seguire utente

● Segui Booklist

● Smetti di seguire Booklist

● Visualizzazione seguiti e seguaci

● Visualizzazione Booklist

## Gestione Ticket

### NON TESTATI

● Interazione Ticket

● Visualizzazione Ticket

## Gestione Customer

### NON TESTATI

● Rimozione Utente

● Rimozione Recensione

● Visualizzazione Utenti

## Gestione Libri

### NON TESTATI

● Aggiunta al catalogo

● Rimozione dal catalogo

● Modifica al prezzo libro

● Creazione nuovo libro

# Pass/Fail criteria

Un test avrà successo (pass) se, dato l’input al sistema, l’output osservato sarà diverso dall’output attesto (oracolo). Al contrario, un test fallirà (fail) quando il risultato osservato è uguale a quello atteso. Il testing sarà considerato valido se tutti i seguenti vincoli saranno rispettati: testare tutti i requisiti fondamentali, effettuare test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti, infine raggiungere un branch coverage non inferiore al 75%.

# Approccio

Per il sistema SocialBook, il testing si compone tre fasi. Nella prima fase, verranno eseguiti i test di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità andando a constatarne il corretto funzionamento. Nella seconda fase, verrà effettuato il testing di integrazione in cui si andrà a testare l’integrazione dei vari sottosistemi. Infine, verrà eseguito il testing di sistema che vedrà come oggetto di testing l’intero sistema assemblato nei suoi componenti. Nella sessione di testing del sistema verrà utilizzato un approccio “BLACK-BOX” ed una strategia bottom-up: tale strategia prevede che i sottosistemi nel layer più in basso della gerarchia vengano testati individualmente, successivamente vengono testati e integrati i layer più in alto. Si ripete questo passo finché tutti i sottosistemi previsti sono testati.

## Test di unità

Il test di unità verrà effettuato sulle classi appartenenti al data access layer. Si definiranno i casi di test in modo funzionale (black-box). Per la definizione dei casi di test unitari verrà utilizzato il framework Junit (). Verrà definita una classe di test per ciascuna classe sorgente da testare.

## Test di integrazione

Il test di integrazione verrà effettuato sulle classi appartenenti all’application layer . La definizione dei casi di test avverrà tramite i framework Junit e Mockito. Verrà definita una classe di test per ciascuna classe sorgente da testare.

## Test di sistema

Per questo tipo di testing sarà utilizzato il tool Selenium IDE, che permette di registrare delle interazioni utente su un browser, e così di implementare ed eseguire i casi di test di sistema. Durante il testing di sistema, il server sarà deployato in localhost.

# Sospensione e Ripristino

In questa sezione verranno specificati i criteri di sospensione del test. Verranno inoltre specificate le attività di test che devono essere ripetute quando si riprende il test.

## Criteri di sospensione

Il testing non verrà sospeso fino alla sua terminazione anche in caso di rilevazione di una failure. Il testing verrà momentaneamente sospeso nel momento in cui l’esecuzione ritorna un errore nella definizione di uno dei test.

## Criterio di ripristino

Il testing verrà ripreso dopo aver risolto i fault individuati nella precedente esecuzione dei test.

# Materiale di testing

L’hardware necessario per l’attività di test è un computer non necessariamente avente connessione ad Internet, in quanto il sistema non è stato ancora rilasciato in produzione.

# Casi di test

## Gestione Utente

### TC\_GU1 Registrazione

|  |  |
| --- | --- |
| **Parametri** | |
| **Parametro: Nome**  **Formato: [a-zA-Z]** | |
| **Lunghezza [LN]** | 1. <1 or >15 [error] 2. >=1 or <16 [LN\_ok] |
| **Formato [FN]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FN\_ok] |
| **Parametro: Cognome**  **Formato: [a-zA-Z]** | |
| **Lunghezza [LC]** | 1. <1 or >15 [error] 2. >=1 or <16 [LC\_ok] |
| **Formato [FC]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FC\_ok] |
| **Parametro: Username**  **Formato: [a-zA-Z0-9]** | |
| **Lunghezza [LU]** | 1. <1 or >16 [error] 2. >=1 or <17 [LU\_ok] |
| **Formato [FU]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FU\_ok] |
| **Parametro: e-mail**  **Formato: \w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})** | |
| **Lunghezza [LE]** | 1. <6 or >50 [error] 2. >=6 or <51 [LE\_ok] |
| **Formato [FE]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FE\_ok] |
| **NonRegistrato [NRE]** | È già presente nel database [error]  1. Non è già presente nel database [NRE\_ok] |
| **Parametro: Password**  **Formato: ((?=.\*[A-Za-z])(?=.\*\\d)[A-Za-z\\d]{8,})** | |
| **Lunghezza [LP]** | 1. <8 or >20 [error] 2. >=8 or <21 [LP\_ok] |
| **Formato [FP]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FP\_ok] |
| **Parametro: Descrizione**  **Formato: (?=[a-zA-Z0-9- !?:.,()]{0,150}$)** | |
| **Lunghezza [LD]** | 1. >150 [error] 2. <150 [LD\_ok] |
| **Formato [FD]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FD\_ok] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| **TC\_GU1\_01** | **LN1** | [error] |
| **TC\_GU1\_02** | **LN2, FN1** | [error] |
| **TC\_GU1\_03** | **LN2, FN2, LC1** | [error] |
| **TC\_GU1\_04** | **LN2, FN2, LC2, FC1** | [error] |
| **TC\_GU1\_05** | **LN2, FN2, LC2, FC2, LU1** | [error] |
| **TC\_GU1\_06** | **LN2, FN2, LC2, FC2, LU2, FU1** | [error] |
| **TC\_GU1\_07** | **LN2, FN2, LC2, FC2, LU2, FU2, LE1** | [error] |
| **TC\_GU1\_08** | **LN2, FN2, LC2, FC2, LU2, FU2, LE2, FE1** | [error] |
| **TC\_GU1\_09** | **LN2, FN2, LC2, FC2, LU2, FU2, LE2, FE2, NRE1** | [error] |
| **TC\_GU1\_10** | **LN2, FN2, LC2, FC2, LU2, FU2, LE2, FE2, NRE2, LP1** | [error] |
| **TC\_GU1\_11** | **LN2, FN2, LC2, FC2, LU2, FU2, LE2, FE2, NRE2, LP2, FP1** | [error] |
| **TC\_GU1\_12** | **LN2, FN2, LC2, FC2, LU2, FU2, LE2, FE2, NRE2, LP2, FP2, LD1** | [error] |
| **TC\_GU1\_13** | **LN2, FN2, LC2, FC2, LU2, FU2, LE2, FE2, NRE2, LP2, FP2, LD2, FD1** | [error] |
| **TC\_GU1\_14** | **LN2, FN2, LC2, FC2, LU2, FU2, LE2, FE2, NRE2, LP2, FP2, LD2, FD2** | [ok] |

### TC\_GU2 Log-in

|  |  |
| --- | --- |
| **Parametri** | |
| **Parametro: Username**  **Formato: [a-zA-Z0-9]** | |
| **Presenza nel DB [PU]** | 1. Non è presente [error] 2. É presente [PU\_ok] |
| **Parametro: Password**  **Formato: ^(?=.\*[A-Za-z])(?=.\*\\d)[A-Za-z\\d]{8,}$** | |
| **Presenza nel DB [PP]** | 1. Non è presente [error] 2. É presente [PP\_ok] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| **TC\_GU2\_01** | **PU1** | [error] |
| **TC\_GU2\_02** | **PU2, PU1** | [error] |
| **TC\_GU2\_03** | **PU2, PU2** | [ok] |

### TC\_GU3 Creazione Booklist

|  |  |
| --- | --- |
| **Parametri** | |
| **Parametro: Nome**  **Formato: [a-zA-Z ]** | |
| **Lunghezza [LN]** | 1. <1 or >30 [error] 2. >=1 or <31 [LN\_ok] |
| **Formato [FN]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FN\_ok] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| **TC\_GU3\_01** | **LN1** | [error] |
| **TC\_GU3\_02** | **LN2, FN1** | [error] |
| **TC\_GU3\_02** | **LN2, FN2** | [ok] |

### TC\_GU4 Modifica Booklist

|  |  |
| --- | --- |
| **Parametri** | |
| **Parametro: Nome**  **Formato: [a-zA-Z ]** | |
| **Lunghezza [LN]** | 1. <1 or >30 [error] 2. >=1 or <31 [LN\_ok] |
| **Formato [FN]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FN\_ok] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| **TC\_GU4\_01** | **LN1** | [error] |
| **TC\_GU4\_02** | **LN2, FN1** | [error] |
| **TC\_GU4\_02** | **LN2, FN2** | [ok] |

### TC\_GU5 Elimina Booklist

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

### TC\_GU6 Recensione Prodotto

|  |  |
| --- | --- |
| **Parametri** | |
| **Parametro: Recensione**  **Formato: [a-zA-Z0-9- !?:.,()]** | |
| **Lunghezza [LR]** | 1. <1 or >300 [error] 2. >=1 or <301 [LR\_ok] |
| **Formato [FR]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FR\_ok] |
| **Parametro: Voto**  **Formato: [0-5]** | |
| **Valore [VV]** | 1. <1 or >5[error] 2. >=1 or <6 [VV\_ok] |
| **Formato [FV]** | 1. Non Rispetta il formato [error] 2. Rispetta il formato [FV\_ok] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| **TC\_GU6\_01** | **LR1** | [error] |
| **TC\_GU6\_02** | **LR2, FR1** | [error] |
| **TC\_GU6\_03** | **LR2, FR2, VV1** | [error] |
| **TC\_GU6\_04** | **LR2, FR2, VV2, FV1** | [error] |
| **TC\_GU6\_05** | **LR2, FR2, VV2, FV2** | [ok] |

## Gestione Acquisto

### TC\_GA1 Inserimento prodotto in carrello

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

### TC\_GA2 Rimozione libro dal carrello

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

### TC\_GA3 Acquisto

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

## Gestione Visualizzazione

### TC\_GV1 Visualizzazione profilo utente

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

## Gestione Interazione

### TC\_GI1 Segui utente

Category Partition non presente perché la funzionalità non prevede inserimento di dati da parte dell’utente.

# Test schedule

Tutte le specifiche dei Test Case verranno descritte nel documento di specifica dei Test Case.