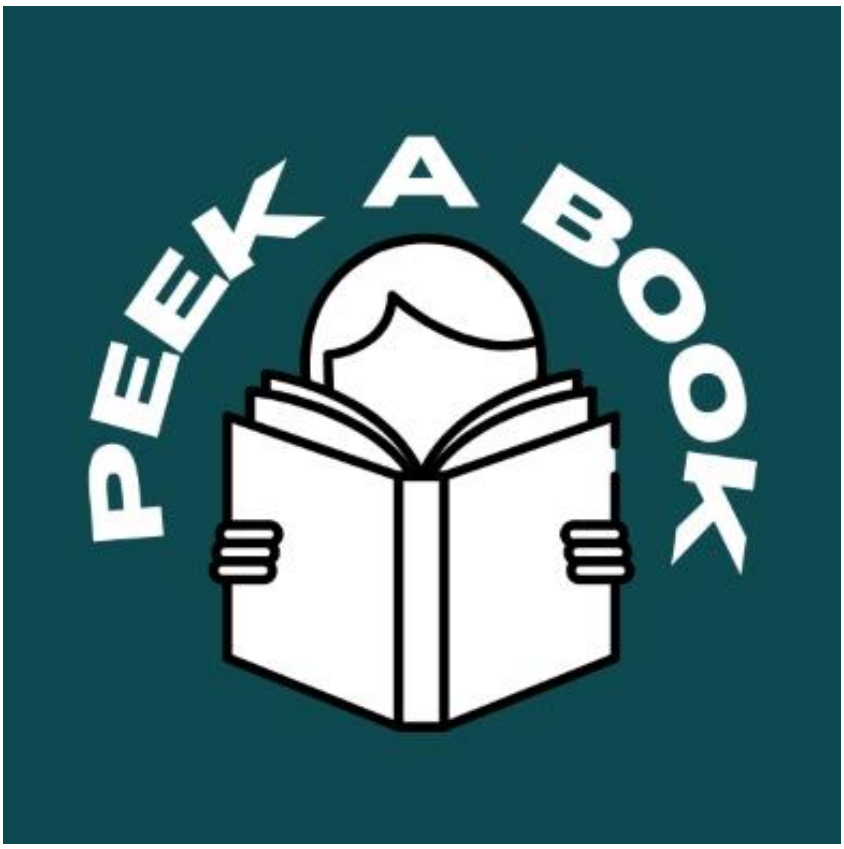
Laurea Triennale in Informatica-Università di Salerno
Corso di *Ingegneria del Software*

TEST PLAN DOCUMENT

"Peek A Book"



Versione	1.0
Data	16/08/2024
Presentato da	Iacomino Domenico, De Luca Ciro

Revision History

Data	Versione	Descrizione	Autori
01/08/2024	0.1	Prima stesura	Domenico Iacomino, Ciro De Luca
16/08/2024	1.0	Aggiunta test cases	Domenico Iacomino, Ciro De Luca

Indice

Test Plan

1. Introduction

Il gruppo Peek A Book intende sviluppare una applicazione web che mira ad essere il punto di riferimento per l'acquisto di libri in Italia ponendo l'accento sulla tempestività dell'approvvigionamento dei prodotti e sulla vicinanza al fruitore.

---

2. Relazione con altri documenti

Al fine di garantire una corretta individuazione dei casi di test, si fa riferimento a decisioni e specifiche descritte nei documenti finora prodotti:

Relazioni con il Requirements Analysis Document (RAD)

I test case pianificati in questo documento sono sviluppati in base alle caratteristiche di sistema specificate dai requisiti funzionali e non funzionali nonché dagli use case fondamentali descritti in fase di analisi.

Relazioni con il System Design Document (SDD)

Nella suddivisione e specifica dei test case si tiene conto delle scelte di design presentate nell'SDD.

3. Panoramica del sistema

Il sistema PeekABook, come già specificato all'interno del SDD, sarà sviluppato seguendo uno stile architetturale three-tier, in particolare tenendo di riferimento il pattern MVC. I sottosistemi individuati nel SDD, e di conseguenza gli oggetti specificati nell'object model presente nel RAD, saranno dunque mappati nelle 3 componenti principali dell'MVC:

- Il Model, che contiene gli oggetti che gestiscono la logica di business e il sottosistema di accesso al database.
- La View, che contiene gli oggetti di tipo boundary individuati dal sistema e che si occupa dell'interazione con l'utente.
- Il Controller, che gestisce il flusso del sistema relativo alle richieste Cliente e Admin, valida i dati in input ed effettua operazioni di rielaborazione delle risposte.

---

4. Features da testare/da non testare

In base all'attuale livello di comprensione del sistema e ai documenti redatti durante lo sviluppo, le features per le quali si effettuerà il testing sono:

Features Utente:

- Registrazione di un nuovo utente
- Login utente registrato
- Conferma di un ordine

Features Amministratore:

- Inserimento di un prodotto

Le funzionalità per le quali non è stata predisposta un'attività di testing sono quelle che sono state classificate a bassa priorità nel RAD e le funzionalità che non prevedono una forte interazione con l'utente in termini di input. In particolare, considerando l'essenza da e-commerce del sistema in sviluppo, le attività che si concentrano esclusivamente sulla visualizzazione dei dati saranno testate in fasi successive a partire dalla codifica delle funzionalità in analisi.

5. Pass/fail criteria

L'obiettivo dell'attività di testing è la scoperta di eventuali faults(errori) o ambiguità introdotte nei vari livelli di sviluppo del prodotto software. Vengono disposti, in tal proposito, dei test case che verificano l'allineamento dell'output reale con l'output atteso dalla funzionalità in analisi. L'output atteso è chiamato anche oracolo e rappresenta il risultato che ci si aspetta di ricevere dall'esecuzione del test con un dato input. Un test ha successo(pass) quando l'output del sistema diverge dall'oracolo specificato. Un test fallisce(fail) quando l'output coincide con il risultato atteso.
Il testing sarà considerato valido se i seguenti vincoli saranno rispettati:

- Testare i requisiti funzionali ad alta priorità;
- Effettuare test di regressione ogni volta che si introducono nuove caratteristiche o vengono modificate quelle presenti;
- Raggiungere un branch coverage non inferiore al 75%.

---

6. Approccio

L'approccio del testing si divide in:

- **Testing di unità**

  Per il testing di unità si utilizza la tecnica di "Black-Box testing". Si focalizza sul comportamento di I/O, senza preoccuparsi della struttura interna della componente. Se per ogni dato input, siamo in grado di prevedere l'output, allora l'unità supera il test.

- **Testing di integrazione**

  Per il testing di integrazione si utilizza la strategia "bottom-up. I sottosistemi al livello più basso della gerarchia sono testati individualmente. I successivi sottosistemi ad essere testati sono quelli che chiamano i sottosistemi testati in precedenza. Si ripete quest'ultimo passo finché tutti i sottosistemi non sono stati testati.

- **Testing di sistema**

  L'ultimo testing prima della messa in uso del sistema prevede il controllo delle funzionalità del sistema, secondo i requisiti specificati. Si utilizzerà il tool "Selenium" per simulare l'interazione da parte dell'utente.

---

7. Suspension and resumption

La fase di testing sarà sospesa quando si otterranno i risultati attesi rispettando però i tempi di consegna del progetto, dato l'elevato investimento di tempo che richiede l'attività.

La fase di testing sarà ripresa, invece se si effettueranno modifiche di sistema. Si rieseguiranno di nuovo quindi i casi di test per verificare la funzionalità del sistema anche nel caso di modifiche.

---

8. Testing materials

Il materiale richiesto per il testing saranno :

- Un dispositivo di elaborazione
- IDE Java EE
- MySQL Server
- JUnit
- Selenium

9. Test case

Sono elencati di seguito i casi di test per le funzionalità da testare :

9.1. Login

Parametro:	Username
Riscontro[R]	1.Trovato 2.Non trovato

Parametro:	Password
Riscontro[R]	1.Trovato 2.Non trovato

Identificativo	**Combinazione**	**Esito**
TC 9.1.1	R.1	Negativo
TC 9.1.2	R.2, R.2	Negativo
TC 9.1.3	R.1, R.1	Positivo

9.2. Registrazione

Parametro:	Username
Formato[FU]	1.Rispetta il formato /^\w+$/ 2.Non rispetta il formato
Riscontro[R]	1. Non esistente 2. Già esistente

Parametro:	Password
Formato[FP]	1.Rispetta il formato /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\da-zA-Z]).{8,}$/ 2.Non rispetta il formato

Parametro:	Nome
Formato[FN]	1.Rispetta il formato /^[A-Za-z]{1,100}[\s*[A-Za-z]{0,100}\s*[A-Za-z]{0,100}]{0,100}$/ 2.Non rispetta il formato

Parametro:	Cognome
Formato[FC]	1.Rispetta il formato /^[A-Za-z]{1,100}[\s*[A-Za-z]{0,100}\s*[A-Za-z]{0,100}]{0,100}$/ 2.Non rispetta il formato

Parametro:	Città
Formato[FCIT]	1.Rispetta il formato /^[A-Za-z]{1,100}[\s*[A-Za-z]{0,100}\s*[A-Za-z]{0,100}]{0,100}$/ 2.Non rispetta il formato

Parametro:	Via
Formato[FV]	1.Rispetta il formato /^[A-Za-z]{1,100}[\s*[A-Za-z]{0,100}\s*[A-Za-z]{0,100}]{0,100}$/ 2.Non rispetta il formato

Parametro:	civico
Formato[FCIV]	1.Rispetta il formato /^(\d+)$/ 2.Non rispetta il formato

Parametro:	CAP
Formato[FCAP]	1.Rispetta il formato /^(\d{5})$/ 2.Non rispetta il formato

Parametro:	email
Formato[FE]	1.Rispetta il formato /^\S{1,30}@\S{1,30}\.\S{1,20}$/ 2.Non rispetta il formato

Identificativo	**Combinazione**	**Esito**
TC 9.2.1	FN.2	Negativo
TC 9.2.2	FU.1	Negativo
TC 9.2.3	FE.2, FN.1, FC.1	Negativo
TC 9.2.4	FE.1, FN.1, FC.1, FP.2	Negativo
TC 9.2.5	FU.1, R.1, FE.1, FN.1, FC.1, FP.1	Negativo
TC 9.2.6	FU.1, R.2, FE.1, FN.1, FC.1, FP.2, FV.1, FCAP.1, FCIV.1, FICIT.1	Negativo
TC 9.2.7	FU.1, R.1, FE.1, FN.1, FC.1, FP.2, FV.1, FCAP.1, FCIV.1, FICIT.1	Positivo

9.3. Aggiunta libro nel catalogo

Parametro:	Nome
Formato[FN]	1.Rispetta il formato /^[A-Za-z]+[\s*[A-Za-z]+\s*[A-Za-z]*]*$/ 2.Non rispetta il formato

Parametro:	Prezzo
Formato[FPR]	1.Rispetta il formato /^([0-9]+\.[0-9]{2})$/ 2.Non rispetta il formato

Parametro:	Quantità
Formato[FQ]	1.Rispetta il formato /^([0-9]+)$/ 2.Non rispetta il formato

Parametro:	Anno
Formato[FA]	1.Rispetta il formato /^([0-9]{4})$/ 2.Non rispetta il formato

Parametro:	ISBN
Formato[FI]	1.Rispetta il formato /^([0-9]{10})$/ 2.Non rispetta il formato

Parametro:	Pagine
Formato[FPAG]	1.Rispetta il formato /^([0-9]+)$/ 2.Non rispetta il formato

Parametro:	Editore
Formato[FE]	1.Rispetta il formato /^[A-Za-z]+[\s*[A-Za-z]+\s*[A-Za-z]*]*$/ 2.Non rispetta il formato

Parametro:	Descrizione
Formato[FD]	1.Rispetta il formato /^(?!\s*$).+/ 2.Non rispetta il formato

Parametro:	Rating
Formato[FR]	1.Rispetta il formato /^(?!\s*$).+/ 2.Non rispetta il formato

Parametro:	Autore
Valore[VA]	1.Inserito 2. Non inserito

Parametro:	Genere
Valore[VG]	1.Inserito 2. Non inserito

Parametro:	Immagine
Formato[FIMM]	1.Rispetta il formato .jpg 2. Non rispetta il formato

Identificativo	**Combinazione**	**Esito**
TC 9.3.1	FN.2	Negativo
TC 9.3.2	FN.1	Negativo
TC 9.3.3	FN.1, FP.1, FQ.1, FA.2, FI.1, FP.1, FE.1, FD.1, FR.1, FA.1, FG.1, FIMM.1	Negativo
TC 9.3.4	FN.1, FP.1, FQ.1, FA.1, FI.1, FP.1, FE.1, FD.1, FR.1, FA.1, FG.1	Negativo
TC 9.3.5	FN.1, FP.1, FQ.1, FA.1, FI.1, FP.1, FE.1, FD.1, FR.1, FA.1, FG.1, FIMM.1	Positivo

9.4. Ricerca libro nel catalogo

Parametro:	Titolo da ricercare
Riscontro[RN]	1. Titolo con corrispondenze nel database 2. Titolo senza corrispondenze nel database

Identificativo	Combinazione	Esito
TC 9.4.1	R.2	Negativo
TC 9.4.2	R.1	Positivo