



Università degli Studi di Salerno

Corso di Ingegneria del Software

JapanWorld Test Plan Versione 1.1



Data: 17/08/2020

Progetto: JapanWorld	Versione: 1.1
Documento: Test Plan	Data: 13/07/2021

Coordinatore del progetto:

Nome	Matricola
Emanuele Patella	0512104730

Partecipanti:

Nome	Matricola
Emanuele Patella	0512104730

Scritto da:	Emanuele Patella
--------------------	------------------

Revision History

Data	Versione	Descrizione	Autore
03/07/2021	1.0	Test Plan Versione 1	Emanuele Patella
13/07/2021	1.1	Revisione Documento	Emanuele Patella

Indice

1. Introduzione.....	4
1.1. Relazione con altri documenti.....	4
2. Overview del sistema.....	4
3. Features che vengono testate/non vengono testate.....	4
4. Criteri di successo/fallimento.....	4
5. Approccio.....	4
6. Materiale di testing.....	4
7. Casi di test.....	4

1. Introduzione

Il sistema è ormai completo, quindi si è giunti alla fase cruciale dello sviluppo, il testing e, se superato correttamente, la fase di rilascio.

Il nostro obiettivo, in questo documento specifico, è quello di creare un piano di testing accurato che tenga conto di tutto ciò che è stato deciso in fase di progettazione e verifichi che le richieste dell'acquirente siano state rispettate nel modo il quanto più possibile preciso.

1.1. Relazione con altri documenti

Il piano di testing si concentrerà sul verificare se i Trade-Off decisi nell'SDD e applicati nell'ODD per lo sviluppo del software, ossia Performance, Sicurezza, Usabilità e Robustezza, sono stati sviluppati a dovere.

2. Overview del sistema

Il Sistema, come già indicato più volte negli ultimi documenti, è un sito web basato su una architettura MVC che vuole essere estremamente facile da usare per tutte le fasce d'utenza della piattaforma, senza però perdere in estetica e funzionalità. Il sistema deve anche essere estremamente sicuro, poiché il fulcro delle sue attività sono le transazioni online, di conseguenza, l'altro aspetto fondamentale su cui si è basata la progettazione è quello di rendere il sito il quanto più sicuro possibile, rendendo la struttura e le modalità di interazione dell'utente, il quanto più guidate possibile.

Quest'ultima informazione è fondamentale per definire il piano di testing, in quanto, uno degli obiettivi fondamentali del testing, sarà proprio quello di verificare se è possibile che il sistema possa essere aggirato, divenendo quindi pericolo per gli utenti.

3. Features che vengono testate/non vengono testate

Le features che verranno testate del sito differiscono in base alla componente testata:

- Per quel che riguarda la componente della View, il testing verterà sulla usabilità dell'interfaccia, quindi verranno testate:
 - Le interazioni tra le varie schermate che vengono richiamate dai link appositi;
 - L'accuratezza nella gestione degli input e degli output.
- Per quel che riguarda i Model il testing sarà incentrato sul transito efficace e preciso dei dati, poiché lo scopo del Model è solo quello di fare da tramite tra la View e il Controller.
- Per quel che riguarda il Controller, verrà testata l'accuratezza delle query e tutte le features annesse alle query, quali:
 - Acquisizione dei dati tramite il Model;
 - Update di parametri all'interno dei database in tempo reale;
 - Verifica di credenziali attraverso apposite query con speciali valori di ritorno.

4. Criteri di successo/fallimento

I successi e i fallimenti dei test verranno calcolati in base al tipo di differenze che si presenteranno tra gli oracoli, realizzati precedentemente, e i risultati effettivi del test. Se le differenze saranno sostanziali, cioè:

- Dati riportati in modo errato;
- Prodotti non inseriti nel database;
- Operazioni di transazioni economiche non effettuate;
- Output differenti dall'input immesso nell'operazione affine.

Allora il test sarà da ritenersi fallito, qualora, invece, le differenze all'interno del test, rispetto agli oracoli, consistessero solo in un diverso ordine di inserimento dei dati all'interno del database, si riterrà positivo il risultato del test.

5. Approccio

Il testing del sistema seguirà una struttura di tipo sandwich modificato per l'occorrenza poiché c'è la necessità di testare, in maniera individuale, sia le componenti della View che le componenti del Controller, perché:

- La View necessita di testing approfondito sulla sua interagibilità e capacità di acquisizione degli input da parte dell'utente.
- Il Controller necessita di testing approfondito per quanto riguarda il funzionamento corretto delle query, quindi è preferibile che le varie componenti che lo costituiscono vengano testate singolarmente.

Data l'architettura MVC su cui si basa il nostro programma non è un problema che il model non sia soggetto a testing individuale, poiché la sua unica funzione è quella di fare da tramite per i dati, formattandoli nella maniera corretta.

La modifica nel testing di tipo sandwich sta nel fatto che la fase di testing delle componenti Web-based coinvolgerà direttamente anche i Model, riducendo il numero di fasi di testing separate da 3 a 2.

6. Materiale di testing

Il testing verrà eseguito direttamente sul codice tramite i programmi:

- JUnit per quel che riguarda le componenti in Java, quindi il Controller e le query in JUnit annesse alle funzioni che interagiscono con il database;
- Selenium per quel che riguarda la componentistica Web-based del sito e le componenti di comunicazione ad essa collegate, cioè, la View e il Model;

Entrambi i linguaggi di testing sono stati scelti perché offrono funzionalità peculiari molto specifiche e, nel nostro caso, utili ad un efficace fase di testing.

7. Casi di test

I casi di testing verranno decisi in base ai requisiti più importanti del nostro sistema, cioè Performance, Sicurezza, Usabilità e Robustezza.

Il testing partirà dalle componenti del controller ad effettuare test in JUnit per verificare la loro robustezza e le loro performance.

Una volta che questa prima fase di testing sarà completata, si passerà al testing dell'interfaccia utente e del model, il quale avverrà in simultanea, in quanto le due componenti sono dipendenti l'una dall'altra. Questa serie di test verrà condotta utilizzando Selenium e verranno esplorate l'usabilità e la sicurezza del sito.