



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: 28/08/2020

Coordinatore del progetto:

Nome	Matricola
Emanuele Patella	0512104730

Partecipanti:

Nome	Matricola
Emanuele Patella	0512104730

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

Revision History

Data	Versione	Descrizione	Autore
20/08/2020	1.0	Test Plan Versione 1	Emanuele Patella
28/08/2020	1.1	Revisione Documento	Emanuele Patella

Indice

- 1. Introduzione.....4
- 2. Objective & Task.....4
 - 2.1. Objective.....4
 - 2.2. Task.....4
- 3. Testing strategy.....4
 - 3.1. Alpha testing.....4
 - 3.2. System and Integration Testing.....5
 - 3.3. Performance and Stress Testing.....6

1. Introduzione

Con questo piano di testing cercherò di evidenziare quali operazioni possano portare a criticità o situazioni di inconsistenza dei dati.

2. Task e Obiettivi

2.1. Obiettivi

L'obiettivo principale dei miei test sarà quello di dimostrare la robustezza del mio software a livello di permanenza dei dati e di robustezza propria dell'infrastruttura creata.

2.2. Task

I task di questo test plan sono pochi, ma estremamente importanti per la buona riuscita del progetto, in quanto dovranno mostrare se l'infrastruttura ha delle criticità da risolvere o dei punti in cui è probabile dovranno essere effettuati degli upgrade in futuro.

I Task si divideranno in due macrocategorie:

- Quelli necessari a controllare la stabilità e l'affidabilità del sito (Effettuati già in sede di sviluppo, poiché il progetto fa capo a un vecchio lavoro risalente all'anno accademico 2017/2018);
- Quelli necessari a capire quali siano i punti che possono essere più facilmente soggetti ad upgrade futuri senza intaccare la struttura sviluppata fin'ora.

3. Strategia di testing

3.1. Fase di Alpha Testing

Per la fase di Testing preliminare delle unità sono partito con il testing di tutta l'interfaccia grafica, prima in singoli moduli, e poi di tutto il sito.

Lo scopo di questi test era, principalmente, verificare la presenza di eventuali bug dovuti ad errori nella codifica o a link inseriti in maniera erranea.

L'output di questo test è stato il primo scoglio da affrontare, poiché, finché non si fosse presentata una interfaccia funzionante ed, allo stesso tempo, accattivante, non avrei potuto procedere nello sviluppo della parte back-hand del sito.

Dopo aver completato lo sviluppo dell'interfaccia ho proseguito iniziando con lo sviluppo dei model, in quanto non dipendono direttamente dall'interfaccia, quindi possono essere testati e controllati separatamente dall'interfaccia grafica.

In questo caso il grosso del test è consistito nel compiere le query all'interno di MySQL workbench, così da poter controllare se gli output erano quelli attesi oppure no, portando a galla eventuali criticità sia nella struttura stessa del database che delle query che intendevo effettuare. Una volta verificate le query non ho dovuto far altro che creare i metodi dei model con i giusti dati in input e il giusto output, prima di inserirvi le query con i parametri variabili.

Ultimo passaggio è stato quello di scrivere i controller, cioè i moduli che formattano i dati presi in input da interfaccia grafica e li re-inviano ai model. Il testing di queste componenti non è stato effettuato poiché il loro funzionamento è fortemente influenzato dagli input che gli vengono dati, quindi sono stati implementati e testati durante la fase di "System and Interaction Testing".

3.2. Testi di sistemi e interazione

In questa fase sono state testate tutte le componenti in macro-blocchi, permettendo a interfaccia grafica, controller e model, di interfacciarsi e verificando se ci fossero problemi non rilevati durante la fase di Alpha Testing.

Qui si sono palesati eventuali problemi dovuti all'interazione dei moduli, i quali sono stati subito risolti e corretti.

Questa fase di testing serve anche a portare a galla eventuali problemi dovuti all'idea di design che si ha per il sito, in quanto avere la piattaforma "completa" davanti ti permette di riscontrare eventuali errori logici non rilevabili in fase di sviluppo, i quali sono stati, a loro volta, risolti e se ne è preso nota, per eventuali futuri sviluppatori che dovessero avere a che fare con il progetto in questione.

3.3. Test di stress e performance

I test di stress e performance sono stati eseguiti aprendo il maggior numero di sessioni possibili sul singolo device in mio possesso per verificare eventuali crash e limiti del sistema. I risultati sono stati incoraggianti, in quanto il software è riuscito a reggere un migliaio di schede aperte in contemporanea, nonostante non sia stato possibile simulare l'attività simultanea di utenti, data la presenza di un unico individuo a fare tutto manualmente.

I rallentamenti sono stati minimi, e, in parte, anche dovuti ai limiti tecnici del device.