



Spotibase

Test Plan

Document

Spotibase

Docente

Gravino Carmine

Studenti

Silvio Venturino	0512108837
Umberto Della Monica	0512107742
Gagliarde Nicolapio	0512106980
Alessandro Aquino	0512106527



Revision History

Data	Versione	Descrizione	Autori
26/11/21	0.1	Creazione copertina, sommario. Prima stesura dei punti da 1 a 7	Tutto il team
27/11/21	0.2	Prima stesura di test case e test frame	Tutto il team
02/12/21	0.3	Continuazione test case e revisione regex	Tutto il team

Team Members

Nome	Ruolo nel progetto	Email	Acronimo
Alessandro Aquino	Team member	a.aquino33@studenti.unisa.it	AA
Nicolapio Gagliarde	Team member	n.gagliarde@studenti.unisa.it	NG
Umberto Della Monica	Team member	u.dellamonica2@studenti.unisa.it	UM
Silvio Venturino	Team member	s.venturino@studenti.unisa.it	SV



Sommario

1. Introduzione	4
2. Relazione con gli altri documenti	4
3. Panoramica del sistema	4
4. Funzionalità da testare e non	5
5. Pass/Fail criteria	6
6. Approccio	6
7. Suspension and resumption	7
8. Strumenti per il testing (hardware/software)	7
9. Test cases	8



1.Introduzione

Lo scopo di questo documento è quello di pianificare l'attività di testing del sistema Spotibase. L'attività di testing è fondamentale per verificare la correttezza dell'implementazione del sistema e dunque evitare che si presentino errori durante l'utilizzo da parte dell'utente finale.

Il documento presenta le funzionalità da testare e da non testare, le strategie di testing, i vari casi di test.

2.Relazione con gli altri documenti

- Relazione con RAD

Il seguente documento è in stretta relazione con il documento di Analisi dei Requisiti prodotto precedentemente, poiché' l'attività di testing viene effettuata sulle funzionalità individuate nel RAD.

Riferimento : Spotibase_RAD

- Relazione con SDD

Nel documento di System Design sono indicati i servizi dei sottosistemi e l'architettura del sistema da testare.

Riferimento : Spotibase_SDD

3.Panoramica del sistema

Il sistema segue l'architettura three-tier dove il Data tier si occupa della gestione del dominio applicativo, il Presentation tier si occupa di dare in output delle informazioni sotto forma di testo o grafica, il Logic tier gestisce l'interazione tra gli altri tier.

Il Logic tier è dunque suddiviso in sottosistemi quali :

- Gestione Canzone
- Gestione Carrello
- Gestione Playlist
- Gestione di Autenticazione
- Gestione Preferenza



4. Funzionalità da testare e non

Saranno testate tutte le funzionalità del sistema, ad esclusione di :

- Funzionalità di visualizzazione, poiché non hanno nessun input
- Oggetti Bean, poiché hanno soltanto metodi di get e set
- Funzionalità con priorità MEDIA e BASSA

Carrello, Canzone e Preferenza non verranno ulteriormente ampliate nel seguito del documento in quanto non presentano input dall'utente ma sono presenti visto che verranno ugualmente testate con un approccio white box.

Carrello:

- Acquistare Carrello;

Canzone:

- Riproduzione Canzone
- Inserimento Canzone In Carrello

Preferenza:

- Inserisci Preferenza

Autenticazione e Playlist invece verranno ampliati all'interno del documento perchè prendono input dall'utente e quindi si può utilizzare il category partition.

Autenticazione:

- Registrazione
- Login

Playlist:

- Creazione Playlist



5. Pass/Fail criteria

Il criterio per il quale il testing avrà successo sarà quando verrà individuata una failure, quindi quando l'output risultato sarà diverso dall'oracolo (output atteso). In tal caso si procederà poi all'individuazione e correzione dell'errore. Inoltre bisogna raggiungere il 75% di branch coverage.

6. Approccio

L'approccio del testing si divide in :

Testing di unità

Per il testing di unità si utilizza la tecnica di “White-Box testing”, utilizzando la tecnica del branch testing, ovvero la ricerca di tutti i possibili branch all'interno del codice che si sta testando.

Per il testing di unità utilizzeremo la libreria Java Junit per effettuare il mocking dei DAO. Per mock di un oggetto si intende oggetti simulati che riproducono il comportamento degli oggetti reali in modo controllato. Un programmatore crea un oggetto mock per testare il comportamento di altri oggetti, reali, ma legati ad un oggetto inaccessibile o non implementato; quest'ultimo verrà sostituito da un mock. Inoltre si utilizzerà Spring per la request e la response e JaCoco per la branch coverage.

Per eseguire è stato creato un database di testing per simulare il database reale.



Testing di integrazione

Non effettueremo il testing di integrazione ma forniremo una descrizione chiara del concetto. La scelta è dovuta dal nostro sistema, essendo già sviluppato in precedenza, il testing di integrazione è stato effettuato ma non documentato. Per completezza verrà implementato un test di integrazione, in specifico sulla servletPlaylist del logic tier.

I test di integrazione sono test che coinvolgono l'accesso al disco, il servizio dell'applicazione e testano la comunicazione tra il logic e il data tier.

Dopo aver eseguito il testing di unità, verrà effettuato il testing di integrazione, testando le funzionalità delle servlet utilizzando le implementazioni vere dei DAO (non Mock) che andranno a scrivere nel database reale.

Per il testing di integrazione avremmo potuto utilizzare una 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. Per questo testing utilizzeremo Selenium IDE.

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 al sistema. Si eseguiranno nuovamente, quindi, i casi di test per verificare la funzionalità del sistema anche nel caso di modifiche.



8. Strumenti per il testing (hardware/software)

Il materiale richiesto per il testing saranno :

- Un dispositivo di elaborazione
- IDE Java
- MySQL Server
- JUnit
- Jacoco
- Spring
- Selenium IDE

9. Test cases

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

1.1 Accedere alla piattaforma

Parametro: email Formato: [a-zA-Z0-9.!#\$%&'*\+/=?^_`{ }~-]{1,63}@[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?)*\$	
Categorie	Scelte
formato fe	1: rispetta il formato [property formatoFEok] 2: non rispetta il formato [errore]
esiste ee	1: è associata a un utente [if formatoFEok][property esisteEEok] 2: non è associata nessun utente[if formatoFE][errore]



Parametro: password

Categorie	Scelte
esiste eep	1: è associata a un utente [property esisteEEPok] 2: non è associata a nessun utente [errore]

Codice	Combinazione	Esito
TC_1.1_01	fe2	errore
TC_1.1_02	fe1.ee2	errore
TC_1.1_03	fe1.ee1.eep2	errore
TC_1.1_04	fe1.ee1.eep1	accesso



1.2 Registrazione

Parametro: email

Formato:

[a-zA-Z0-9.!#\$%&'*/=\?^_`{|}~]{1,63}@[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?)*\$

Categorie	Scelte
lunghezzaPrefisso lipr	<ol style="list-style-type: none">1. lunghezzaPrefisso == 0 [errore]2. lunghezzaPrefisso >= 1 && lunghezzaPrefisso <= 64 [property lunghezzaPrefissoLIPRok]3. lunghezzaPrefisso > 64 [errore]
lunghezzaDominio lido	<ol style="list-style-type: none">1. lunghezzaDominio == 0 [if lunghezzaPrefissoLIPRok [errore]2. lunghezzaDominio >=1 and lunghezzaDominio <= 254 [if lunghezzaPrefissoLIPRok] [property lunghezzaDominioLIDOOK]3. lunghezzaDominio > 254 [if lunghezzaPrefissoLIPRok [errore]
formato fe	<ol style="list-style-type: none">1. rispetta il formato [if lunghezzaPrefissoLIPRok and lunghezzaDominioLIDOOK] [property formatoFEok]2. NON rispetta il formato [if lunghezzaPrefissoLIPRok and lunghezzaDominioLIDOOK] [errore]
Esiste ee	<ol style="list-style-type: none">1. è associata a un utente [if lunghezzaPrefissoLIPRok and lunghezzaDominioLIDOOK and formatoFEok] [errore]2. non è associata a nessun utente [if lunghezzaPrefissoLIPRok and lunghezzaDominioLIDOOK and formatoFEok]



	[property EsisteEEok]
--	-------------------------------

Parametro: username

Formato: $^{[a-zA-Z0-9_\-]\{1,40\}}\$$

Categorie	Scelte
formato fu	<ol style="list-style-type: none">1. rispetta il formato [property formatoFUok]2. NON rispetta il formato [errore]
nonEsiste eu	<ol style="list-style-type: none">1. è associata a un utente [if formatoFUok] [errore]2. non è associata a nessun utente [if formatoFUok] [property nonEsisteEUok]

Parametro: Password

Formato: $^{(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).\{8,20\}}\$$

Categorie	Scelte
formato fp	<ol style="list-style-type: none">1. rispetta il formato [property formatoFPok]2. NON rispetta il formato [errore]



Parametro: Conferma Password

Categorie	Scelte
Corrisponde cp	1. NON corrisponde [errore] 2. corrisponde [property corrispondeCPok]

Codice	Combinazione	Esito
TC_1.2_01	lipr1	errore
TC_1.2_02	lipr3	errore
TC_1.2_03	lipr2.lido1	errore
TC_1.2_04	lipr2.lido3	errore
TC_1.2_05	lipr2.lido2.fe2	errore
TC_1.2_06	lipr2.lido2.fe1.ee1	errore
TC_1.2_07	lipr2.lido2.fe1.ee2.fu2	errore
TC_1.2_08	lipr2.lido2.fe1.ee2.fu1.eu1	errore
TC_1.2_09	lipr2.lido2.fe1.ee2.fu1.eu2 .fp2	errore
TC_1.2_10	lipr2.lido2.fe1.ee2.fu1.eu2 .fp1.cp1	errore
TC_1.2_11	lipr2.lido2.fe1.ee2.fu1.eu2 .fp1.cp2	inserimento



1.3 Creazione Playlist

Parametro: Titolo Formato: $^{\wedge}[a-zA-Z0-9_.-.]\{0,100\}\backslash w\$$	
Categorie	Scelte
Formato ft	1.rispetta il formato [property FormatoFTok] 2. non rispetta il formato [errore]
Esiste et	1. è associato a una playlist [if FormatoFTok] [errore] 2.non esiste nel DB [if FormatoFTok] [property EsisteETok]

Parametro: Note Formato: null $^{\wedge}[a-zA-Z0-9_.-.]\{0,100\}\backslash w\$$	
Categorie	Scelte
Formato ftn	1. rispetta il formato [property FormatoFTNok] 2. non rispetta il formato [errore]

Codice	Combinazione	Esito
TC_1.3_01	ft2	errore
TC_1.3_02	ft1.et1	errore
TC_1.3_03	ft1.et2.ftn2	errore
TC_1.3_04	ft1.et2.ftn1	inserimento