*Corso di Laurea in Informatica, prof. A. De Lucia,*

*a.a 2021/2022*

*Progetto di Ingegneria del Software*



*Test Plan*

|  |  |
| --- | --- |
| Partecipanti | Matricola |
| Marta Coiro | 0512108154 |
| Katia Buonocore | 0512106528 |
| Rita Cuccaro | 0512109495 |
|  |  |
|  |  |
|  |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 16/11/2021 | 1.0 | Prima stesura del documento (Problem [Statement](#_top)) | Membri del team |
| 30/11/2021 | 1.0 | Requirement Analysis Document | Membri del team |
| 06/12/2021 | 1.0 | System Design Document | Membri del team |
| 20/12/2021 | 1.0 | Gestione Dati Persistenti\_MusicConsole | Membri del team |
| 27/12/2021 | 1.0 | Object Design Document | Membri del team |
|  | 1.0 | Test Plan | Membri del team |
|  |  |  |  |

**Indice**

1. INTRODUZIONE

2. DOCUMENTI CORRELATI

2.1 Relazioni con il documento di analisi dei requisiti (RAD).

2.2 Relazioni con il System Design Document (SDD).

2.3 Relazioni con l’Object Design Document (ODD).

3. PANORAMICA DEL SISTEMA.

4.FUNZIONALITA’ DA TESTARE.

5. CRITERI PASS/FAIL.

6. APPROCCIO

6.1 Test di unità

6.2 Test di integrazione

6.3 Test di sistema

7. SOSPENSIONE E RIPRESA

7.1 Criteri di sospensione

7.2 Criteri di ripresa

8. MATERIALE PER IL TESTING

9. Test Cases

Elenco vari test case

**1. INTRODUZIONE**

Lo scopo di questo documento è quello di andare ad analizzare e gestire lo sviluppo e le attività di testing che riguardano il nostro sistema ***MusicConsole***.

In particolar modo in questa sessione di lavoro, l’obiettivo sarà quello di verificare il corretto funzionamento del sistema sviluppato in diversi casi. Questi casi sono stati studiati per mettere alla prova le varie funzionalità del sistema ed effettuando questi test case saremo in grado di rivelare eventuali bug, errori e/o incongruenze tra il comportamento desiderato e quello osservato.

I risultati ci permetteranno di capire dove bisognerà correggere eventuali errori o apportare modifiche per migliorare i vari sottosistemi.

**2. DOCUMENTI CORRELATI**

Il Test Plan ha una stretta relazione con i documenti prodotti in precedenza in quanto il sistema è stato pianificato precedentemente nelle relative documentazioni.

Nella fase di testing, si verificherà se le aspettative descritte nei precedenti documenti siano state rispettate.

***2.1 RELAZIONI CON IL DOCUMENTO DI ANALISI DEI REQUISITI (RAD)***

La relazione tra Test Plan e RAD riguarda i requisiti funzionali e non funzionali del sistema, poiché i test che verranno svolti, saranno eseguiti su quelle funzionalità tendendo conto delle specifiche espresse nel precedente documento.

***2.2 RELAZIONI CON IL SYSTEM DESIGN DOCUMENT(SDD)***

La relazione tra Test Plan e SDD riguarda la suddivisione dell’architettura del sistema sviluppato in tre layer e quindi il test deve tener conto di queste suddivisioni.

***2.3 RELAZIONI CON L’OBJECT DESIGN DOCUMENT (ODD)***

L’ODD contiene i package e le class interfaces del sistema che vanno prese in considerazione durante le operazioni di testing.

**3.PANORAMICA DEL SISTEMA**

Come riportato nel SDD la struttura del nostro sistema è suddivisa secondo un’architettura three-tier: Interface Layer, Application Logic Layer e Data Layer.

Inoltre, il sistema è suddiviso in sottosistemi ed è stato diviso in gestioni quali:

* Gestione Account
* Gestione Brani
* Gestione Podcast
* Gestione Album
* Gestione Playlist
* Gestione Acquisti
* Gestione Ordini
* Gestione Rifornimento Prodotti

**4. FUNZIONALITÀ DA TESTARE**

Le funzionalità che saranno sottoposte a test sono:

* Gestione Account

1. Registrazione
2. Login

* Gestione Brani

1. Aggiunta brano
2. Modifica brano

* Gestione Podcast

1. Aggiunta podcast
2. Modifica podcast

* Gestione Album

1. Aggiungi album
2. Modifica album

* Gestione Playlist
* Gestione Acquisti

1. Visualizzazione carrello
2. Checkout carrello

* Gestione Ordini
* Gestione Rifornimento Prodotti

**5. CRITERI PASS/FAIL**

I dati in input saranno raggruppati in base a caratteristiche comuni in modo da effettuare un test su un unico elemento rappresentativo.

Il testing ha successo se l’output osservato è diverso dall’output atteso cioè la fase di testing avrà successo se individuerà una failure. Verrà quindi analizzata la failure e, se legata ad un fault, si procederà alla sua correzione.

Verrà, infine, iterata la fase di testing per verificare che la modifica apportata non abbia alcun impatto sulle altre componenti del sistema.

**6. APPROCCIO**

***6.1 TEST DI UNITÀ***

Da vedere quale modello sceglie

***6.2 TEST DI INTEGRAZIONE***

Da vedere quale modello scegliere

***6.3 TEST DI SISTEMA***

Il test di sistema serve a dimostrare che tutti i requisiti richiesti sono stati soddisfatti. Lo scopo è testare tutte le funzionalità.

Trattandosi di un sistema web-based verrà utilizzato il tool di Selenium, per fare tale testing, il quale si occupa di simulare l’interazione con il sistema dal punto di vista dell’utente.

**7. SOSPENSIONE E RIPRESA**

***7.1 CRITERI DI SOSPENSIONE***

La fase di testing del sistema verrà sospesa quando si otterranno i risultati attesi.

***7.2 CRITERI DI RIPRESA***

La fase di testing potrà riprendere dopo che ci sono state modifiche o correzioni apportate al sistema, che generano errori o fallimenti.

I test case verranno nuovamente sottoposti al sistema in modo tale da assicurarsi di aver risolto il problema effettivamente.

**8. MATERIALE PER IL TESTING**

Gli strumenti necessari per l’attività di testing sono:

* Computer sul quale è installato un browser e con una connessione ad Internet
* Selenium