|  |  |
| --- | --- |
| LogoUniversitàDiSalerno.jpg | **Università degli Studi di Salerno**  **Corso di Ingegneria del Software** |

**eRockOn**

**Test Plan**

**Versione 2.0**



|  |  |
| --- | --- |
| **Progetto**: eRockOn | **Versione** : 2.0 |
| **Documento**: Test plan | **Data** : 25/02/2016 |

**Partecipanti:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Luigi Russo [LR] | 0512102280 |
| Roberto Prisco [PR] | 0512101718 |

|  |  |
| --- | --- |
| **Scritto da:** | Luigi Russo |

Indice

1. Introduzione

2. Relazione con altri documenti

3. Panoramica del Sistema

4. Funzionalità da Testare e non

5. Pass/fail criteria

6. Approccio

6.1. Testing di Unità

6.2. Testing di accettazione

7. Test cases

8. Specfica dei test Cases

Introduzione

Il test è un procedimento che fa parte del ciclo di vita del software utilizzato per individuare le carenze di correttezza, completezza ed affidabilità; Ai fini di ciò presentiamo questo documento con lo scopo di gestire e analizzare le attività di testing riguardanti eRockOn e i relativi sviluppi. L’obiettivo primario è quello di trovare possibili bug nelle componenti del sistema. In base a questi risultati sarà possibile capire dove bisognerà intervenire , e quindi apportare eventuali modifiche e correggere eventuali errori, così da migliorare i vari sottosistemi e rendere efficiente il prodotto. L’iterazione del processo sarà effettuata fino al momento in cui non si otterranno gli esiti richiesti rispettando i tempi di sviluppo previsti.

Relazione con altri documenti

Per la documentazione di ogni test descritto si rimanda al documento TCS\_eRockOn , esso contiene gli input e gli output attesi.

Panoramica del sistema

eRockOn è una piattaforma web-based per il noleggio di strumenti musicali e sale di registrazione, che permette ad ogni utente del servizio di accedere al sistema previa autenticazione, e di controllare tutti gli annunci da lui inseriti e gli annunci inseriti dagli altri utenti del sistema. Inoltre, dopo aver visualizzato i dati dello strumento/sala a cui è interessato, l’utente può inviare una richiesta di noleggio al proprietario dello strumento/sala in questione, che deciderà se accettare o meno. Ogni annuncio verrà supervisionato da un amministratore che deciderà se approvarlo oppure no. Inoltre l’amministratore potrà in qualunque momento eliminare un annuncio già approvato o un utente.

Funzionalità da testare e non

Le componenti che saranno testate sono:

* Login
* Inserimento annuncio strumento
* Inserimento annuncio sala
* Invio richiesta di noleggio

Le componenti che non saranno testate sono :

* Area amministratore
* Area utente
* Pagina di ricerca

Pass/fail criteria

L’obiettivo della fase di test è quello di progettare i test in modo tale da riscontrare delle discordanze fra il comportamento atteso e quello implementato, quindi si cerca di dimostrare che l’implementazione sia inconsistente con il modello del sistema.

Quindi in questa fase devono essere trovati il maggior numero di errori che poi dovranno essere corretti dagli sviluppatori.

Approccio

I casi di test sono progettati sulla base della descrizione del sistema, ovvero partendo dal documento di specifiche del sistema; l’approccio che verrà utilizzato è ***Black box testing*** *,* quindi non sarà controllata la struttura interna del sistema ma il comportamento di esso in base agli input e output.

Dall’insieme dei dati di input possibili si individua il sottoinsieme che può rivelare la presenza di difetti nel sistema in modo da progettare casi di test efficaci.

Individuati i bug e successivamente corretti, le componenti saranno pronte per essere integrate in sottosistemi più grandi. Utilizzando un approccio ***Big-bang*** le componenti corrette saranno testate individualmente e poi tutte insieme come un unico sistema,nel caso in cui si dovessero riscontrare nuovi bug, non individuati durante il test di unità (black box), verranno aggiunte nuove componenti per correggerli.

Dopodichè il sistema sarà testato nella sua completezza per assicurarci che sia conforme ai requisiti funzionali.

Test Case

Per selezionare gli input per il test del sistema verrà effettuata la partizione in classi equivalenti. Gli input e output saranno raggruppati in classi in cui tutti i membri di una classe sono relazionati , cioè il programma effettua elaborazioni simili per ciascun membro della stessa classe.

Specifica dei Test cases

## TC\_1.0 Login

|  |  |
| --- | --- |
| **Parametri**: Username  **Formato**: [az-AZ-09]{30} |  |
| **Categorie** | **Scelte** |
| **Lunghezza Lu** | 1:Lunghezza ==0 [errore]  2:lunghezza==30 [property lunghezzaLuOk]  3:Lunghezza>30 [errore] |
| **Formato Fu** | 1:rispetta il formato  [if lunghezzaLuOk][property formatoFuOk]  2: non rispetta il formato  [if !lunghezzaLuOk][errore] |
| **Esiste Eu** | 1: esiste nel DB [if lunghezzaLuOk and formatoFuOk][proprety EsisteEuOk]  2: non esiste nel DB [if lunghezzaLuOk and !formatoFuOk][errore] |

|  |  |
| --- | --- |
| **Parametri**: Password  **Formato**: [az-AZ-09]{30} |  |
| **Categorie** | **Scelte** |
| **Lunghezza Lp** | 1:Lunghezza ==0 [errore]  2:lunghezza==30 [property lunghezzaLpOk]  3:Lunghezza>30 [errore] |
| **Formato Fp** | 1:rispetta il formato  [if lunghezzaLpOk][property formatoFpOk]  2: non rispetta il formato  [if !lunghezzaLpOk][errore] |
| **Esiste Ep** | 1: esiste nel DB [if lunghezzaLpOk and formatoFpOk][proprety EsisteEpOk]  2: non esiste nel DB [if lunghezzaLpOk and !formatoFpOk][errore] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| TC\_1.0\_1 | Lu1 | Errore |
| TC\_1.0\_2 | Lu3 | Errore |
| TC\_1.0\_3 | Lp1 | Errore |
| TC\_1.0\_4 | Lp3 | Errore |
| TC\_1.0\_5 | Lu2 Fu1 Eu2 | Errore |
| TC\_1.0\_6 | Lp2 Fp2 Ep2 | Errore |
| TC\_1.0\_7 | Lu2 Fu1 Eu1 Lp1 Fp2 Ep2 | Errore |
| TC\_1.0\_8 | Lu2 Fu1 Eu1 Lp3 Fp2 Ep2 | Errore |
| TC\_1.0\_9 | Lu1 Lu2 Eu2 Lp1 Fp2 Ep2 | Errore |
| TC\_1.0\_10 | Lu1 Lu2 Eu2 Lp3 Fp2 Ep2 | Errore |
| TC\_1.0\_11 | Lu2 Fu1 Eu1 Lp2 Fp1 Ep1 | Login Effettuato |

## TC\_1.1 Inserimento Annuncio Strumento

|  |  |
| --- | --- |
| **Parametri**: Descrizione  **Formato**: [az-AZ-1-9]{200} |  |
| **Categorie** | **Scelte** |
| **Lunghezza Ld** | 1:Lunghezza ==0 [errore]  2:lunghezza==200 [property lunghezzaLdOk]  3:Lunghezza>200 [errore] |
| **Formato Fd** | 1:rispetta il formato  [if lunghezzaLdOk][property formatoFdOk]  2: non rispetta il formato  [if !lunghezzaLdOk][errore] |

|  |  |
| --- | --- |
| **Parametri**: NomeStrumento  **Formato**: [az-AZ-1-9]{20} |  |
| **Categorie** | **Scelte** |
| **Lunghezza Ln** | 1:Lunghezza ==0 [errore]  2:lunghezza==20 [property lunghezzaLnOk]  3:Lunghezza>20 [errore] |
| **Formato Fn** | 1:rispetta il formato  [if lunghezzaLnOk][property formatoFnOk]  2: non rispetta il formato  [if !lunghezzaLnOk][errore] |

|  |  |
| --- | --- |
| **Parametri**: TipoStrumento  **Formato**: [az-AZ]{20} |  |
| **Categorie** | **Scelte** |
| **Lunghezza Lt** | 1:Lunghezza ==0 [errore]  2:lunghezza==20 [property lunghezzaLtOk]  3:Lunghezza>20 [errore] |
| **Formato Ft** | 1:rispetta il formato  [if lunghezzaLtOk][property formatoFtOk]  2: non rispetta il formato  [if !lunghezzaLtOk][errore] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| TC\_1.1\_1 | Ld1 | Errore |
| TC\_1.1\_2 | Ld3 | Descrizione tagliata. |
| TC\_1.1\_3 | Ln1 | Errore |
| TC\_1.1\_4 | Ln3 | Nome tagliato. |
| TC\_1.1\_5 | Lt1 | Errore |
| TC\_1.1\_6 | Lt3 | Tipo tagliato. |
| TC\_1.1\_7 | Ld1,Fd2,Ln1,Fn2,Lt1,Ft2 | Errore |
| TC\_1.1\_8 | Ld1,Fd2,Ln2,Fn1,Lt1,Ft2 | Errore |
| TC\_1.1\_9 | Ld1,Fd2,Ln1,Fn2,Lt2,Ft1 | Errore |
| TC\_1.1\_10 | Ld3,Fd1,Ln3,Fn1,Lt2,Ft1 | Descrizione e Nome tagliati. |
| TC\_1.1\_11 | Ld3,Fd1,Ln2,Fn1,Lt3,Ft1 | Descrizione e Tipo tagliati. |
| TC\_1.1\_12 | Ld2,Fd1,Ln3,Fn1,Lt3,Ft1 | Nome e Tipo tagliati. |
| TC\_1.1\_13 | Ld3,Fd1,Ln3,Fn1,Lt3,Ft1 | Nome, Tipo e Descrizione tagliati. |
| TC\_1.1\_14 | Ld2,Fd1,Ln2,Fn1,Lt2,Ft1 | Annuncio Inserito |

## TC\_1.2 Inserimento Annuncio Sala

|  |  |
| --- | --- |
| **Parametri**: Descrizione  **Formato**: [az-AZ-1-9]{200} |  |
| **Categorie** | **Scelte** |
| **Lunghezza Ld** | 1:Lunghezza ==0 [errore]  2:lunghezza==200 [property lunghezzaLdOk]  3:Lunghezza>200 [errore] |
| **Formato Fd** | 1:rispetta il formato  [if lunghezzaLdOk][property formatoFdOk]  2: non rispetta il formato  [if !lunghezzaLdOk][errore] |

|  |  |
| --- | --- |
| **Parametri**: NomeSala  **Formato**: [az-AZ-1-9]{20} |  |
| **Categorie** | **Scelte** |
| **Lunghezza Ln** | 1:Lunghezza ==0 [errore]  2:lunghezza==20 [property lunghezzaLnOk]  3:Lunghezza>20 [errore] |
| **Formato Fn** | 1:rispetta il formato  [if lunghezzaLnOk][property formatoFnOk]  2: non rispetta il formato  [if !lunghezzaLnOk][errore] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| TC\_1.2\_1 | Ld1 | Errore |
| TC\_1.2\_2 | Ld3 | Descrizione tagliata |
| TC\_1.2\_3 | Ln1 | Errore |
| TC\_1.2\_4 | Ln3 | Nome tagliato |
| TC\_1.2\_5 | Ld1,Fd2,Ln1,Fn2 | Errore |
| TC\_1.2\_6 | Ld1,Fd2,Ln2,Fn1 | Errore |
| TC\_1.2\_7 | Ld1,Fd2,Ln3,Fn1 | Errore |
| TC\_1.2\_8 | Ld2,Fd1,Ln1,Fn2 | Errore |
| TC\_1.2\_9 | Ld3,Fd1,Ln3,Fn1 | Descrizione e Nome tagliati |
| TC\_1.2\_10 | Ld2,Fd1,Ln2,Fn1 | Annuncio Inserito |

## TC\_1.3 Invio richiesta di noleggio

|  |  |
| --- | --- |
| **Parametri**: Data Inizio  **Formato**: [gg/mm/aaaa] |  |
| **Categorie** | **Scelte** |
| **Formato Fi** | 1:rispetta il formato  [property formatoFiOk]  2: non rispetta il formato  [errore] |

|  |  |
| --- | --- |
| **Parametri**: Data Fine  **Formato**: [gg/mm/aaaa] |  |
| **Categorie** | **Scelte** |
| **Formato Ff** | 1:rispetta il formato  [property formatoFfOk]  2: non rispetta il formato  [errore] |

|  |  |  |
| --- | --- | --- |
| **Codice** | **Combinazione** | **Esito** |
| TC\_1.31\_1 | Fi2 | Errore |
| TC\_1.3\_2 | Ff2 | Errore |
| TC\_1.3\_3 | Fi1,Ff1 | Richiesta inviata |