­

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Plan  Progetto  SalernArte   |  |  | | --- | --- | | Versione | 0.1 | | Data | 28/11/2022 | | Destinatario | Studenti di Ingegneria del Software 2021/22 | | Presentato da | Martino Lucia [0512105234], Longo Marco [0512105945], Della Pepa Alessia [0512105720] | | Approvato da |  | |

Membri del Team

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Matricola** | **Acronimo** | **Informazioni di contatto** |
| Della Pepa Alessia | 0512105720 | DPA | a.dellapepa5@studenti.unisa.it |
| Longo Marco | 0512105945 | LM | m.longo36@studenti.unisa.it |
| Martino Lucia | 0512105234 | ML | l.martino11@studenti.unisa.it |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 28/11/2022 | 0.1 | Prima stesura: struttura del documento | DPA, ML, LM |

Sommario

[Membri del Team 2](#_Toc120544981)

[RevisionHistory 2](#_Toc120544982)

[1. Introduzione 3](#_Toc120544983)

[2. Relazione con gli altri documenti 3](#_Toc120544984)

[3. Panoramica del sistema 3](#_Toc120544985)

[4. Funzionalità da testare e non testare 4](#_Toc120544986)

[4.1Gestione utente 4](#_Toc120544987)

[4.2 4](#_Toc120544988)

[5. Pass/Fail criteria 4](#_Toc120544989)

[6. Approccio 4](#_Toc120544990)

[6.1. Testing di unità 4](#_Toc120544991)

[6.2. Testing funzionale 4](#_Toc120544992)

[7. Strumenti per il testing (hardware/software) 4](#_Toc120544993)

[8. Test cases 4](#_Toc120544994)

[9. Specifica dei test cases 4](#_Toc120544995)

[9.1. Gestione cicli di dottorato 4](#_Toc120544996)

[10. Pianificazione del testing e assegnazione dei ruoli 4](#_Toc120544997)

1. Introduzione

SalernArte si propone come obiettivo principale di realizzare un’agenzia online specializzata nella vendita di biglietti riguardanti mostre d’arte ed eventi teatrali e culturali nel salernitano. L’obiettivo è quello di facilitare a tutti i cittadini la ricerca di iniziative culturali salernitane, raggruppando queste ultime in un unico ambiente semplice ed intuitivo, e di ottimizzare il lavoro di organizzatori di eventi che si interfacciano ai cittadini.

Il documento di Test Plan ha l’obiettivo di descrivere ed analizzare le attività di Testing per la piattaforma SalernArte. Il fine è quello di garantire che ogni aspetto funzioni in modo corretto.

Dal momento che il sistema è stato diviso in tre gestioni, per facilitare la divisione degli incarichi e la leggibilità tra i membri del team, abbiamo deciso di mantenere la suddivisione anche nelle attività di testing che verranno divise nelle seguenti gestioni:

* Gestione Utente
* Gestione Eventi
* Gestione Acquisti

1. Relazione con gli altri documenti

Per la stesura e l’individuazione dei test case si fa riferimento agli altri documenti che sono stati prodotti:

* [Requirements Analysis Document](2-Requirements%20Analysis%20Document.docx) (RAD), da cui abbiamo ricavato tramite l’analisi dei requisiti funzionali e non funzionali i test case che verranno presentati nel seguente documento
* [System Design Document](3-System%20Design%20Document%20.docx) (SDD), da cui verrà presa la suddivisione in sottosistemi che dovrà essere rispettata all’interno dei test case.
* [Object Design Document](4-Object%20Design%20Document.docx) (ODD), da rivedere.

1. Panoramica del sistema

Il Sistema proposto è basato sullo stile architetturale Three Tier. Questo tipo di architettura è stata scelta poiché si sposa perfettamente per lo sviluppo di Web Application, come previsto dal nostro progetto.

Per la parte del Front-end saranno utilizzati HTML5 e CSS3.Per quanto riguarda il Back-end verranno utilizzati Java e Javascript. Per la gestione del database saranno usati MySQL e JDBC.

1. Funzionalità da testare e non testare

Di seguito la lista delle features di cui si effettuerà il testing per le varie gestioni:

4.1Gestione utente

4.2 Gestione Eventi

4.3 Gestione Acquisti

Le funzionalità di cui non si andrà ad effettuare le attività di testing delle funzionalità che non prevedono input manuale da parte dell’utente - ad esempio attività riguardanti esclusivamente visualizzazioni di dati.

1. Pass/Fail criteria

La fase di testing viene effettuata per identificare la presenza di faults (errori) nel sistema al fine di effettuare una loro successiva rimozione.

L’esito di un test case è valutato mediante un oracolo, inteso come il risultato atteso della sua esecuzione, basandosi sui requisiti.

Un test ha successo (pass) se, dato un input al sistema, l’output ottenuto è diverso dall’output atteso dall’oracolo.

Un test fallisce (fail) se, dato un input al sistema, l’output ottenuto è uguale all’output atteso dall’oracolo.

Tutto il testing sarà considerato valido se tutti i seguenti vincoli saranno rispettati:

* Testare tutti i requisiti funzionali di alta e media priorità
* Effettuare un test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti;
* Raggiungere un branch coverage non inferiore al 75%

1. Approccio
   1. Testing di unità
   2. Testing funzionale
   3. Testing di Sistema
   4. Testing di integrazione
2. Sospensione e ripristino

In questa sezione verranno specificati i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprende il test.

**Criteri di sospensione**

Il testing non verrà sospeso fino alla sua terminazione, anche in caso di rilevazione di una failure. Il testing potrà essere momentaneamente sospeso nel caso venga restituito, al momento dell’esecuzione, un errore nella definizione di uno dei test stessi.

**Criteri di ripristino**

Il testing verrà ripreso dopo aver risolto i fault individuati

1. Strumenti per il testing (hardware/software)

L’hardware necessario per l’attività di test è un semplice computer, non necessariamente connesso ad internet, in quanto il sistema non è stato ancora rilasciato.

1. Test cases

L’approccio per la definizione dei test frame sarà il category partition. Al fine di minimizzare il numero di test case, gli input saranno partizionati in classi di equivalenza. Per definire l'output atteso si userà un oracolo umano, per via dell'assenza di specifiche formali/semi-formali.

9.1 Gestione Utente

|  |  |  |
| --- | --- | --- |
| **Parametro: Username** | | |
| **FORMATO:**  **^[A-Za-z0-9-\_]\*$** | | |
| **Nome Categoria** | **Scelte per categorie** | |
| Lunghezza [LU] | 1. Lunghezza>30 = false[error] 2. Lunghezza<1 = false[error] 3. 1<Lunghezza<30 = true[PROPERTY LU OK] | |
| Formato [FU] | 1. Rispetta il formato = false [error]  2. Rispetta il formato = true [PROPERTY FU\_OK] | |
| [CU] | 1. Username non presente nel database = false[error] 2. Username non presente nel database = true[PROPERTY CU\_OK] | |
| **Parametro: Nome** | | |
| **FORMATO:**  **/^[A-Za-z \']\*$** | | |
| **Nome Categoria** | **Scelte per categorie** | |
| Lunghezza [LN] | 1. Lunghezza>30 = false[error] 2. Lunghezza<1 = false[error] 3. 1<Lunghezza<=30= true[PROPERTY LN OK] | |
| Formato [FN] | 1. Rispetta il formato = false [error]  2. Rispetta il formato = true [PROPERTY FN\_OK] | |
| **Parametro: Cognome** | | |
| **FORMATO:**  **/^[A-Za-z \']\*$** | | |
| Lunghezza [LC] | | 1. Lunghezza>30 = false[error] 2. Lunghezza<1 = false[error] 3. 1<Lunghezza<=30= true[PROPERTY LC OK] |
| Formato [FC] | | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FC\_OK] |

9.2 Gestione Eventi

|  |  |  |
| --- | --- | --- |
| **Parametro: Username** | | |
| **FORMATO:**  **^[A-Za-z0-9-\_]\*$** | | |
| **Nome Categoria** | **Scelte per categorie** | |
| Lunghezza [LU] | 1. Lunghezza>30 = false[error] 2. Lunghezza<1 = false[error] 3. 1<Lunghezza<30 = true[PROPERTY LU OK] | |
| Formato [FU] | 1. Rispetta il formato = false [error]  2. Rispetta il formato = true [PROPERTY FU\_OK] | |
| [CU] | 1. Username non presente nel database = false[error] 2. Username non presente nel database = true[PROPERTY CU\_OK] | |
| **Parametro: Nome** | | |
| **FORMATO:**  **/^[A-Za-z \']\*$** | | |
| **Nome Categoria** | **Scelte per categorie** | |
| Lunghezza [LN] | 1. Lunghezza>30 = false[error] 2. Lunghezza<1 = false[error] 3. 1<Lunghezza<=30= true[PROPERTY LN OK] | |
| Formato [FN] | 1. Rispetta il formato = false [error]  2. Rispetta il formato = true [PROPERTY FN\_OK] | |
| **Parametro: Cognome** | | |
| **FORMATO:**  **/^[A-Za-z \']\*$** | | |
| Lunghezza [LC] | | 1. Lunghezza>30 = false[error] 2. Lunghezza<1 = false[error] 3. 1<Lunghezza<=30= true[PROPERTY LC OK] |
| Formato [FC] | | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FC\_OK] |

|  |  |  |
| --- | --- | --- |
| **Test case ID** | **Test Frame** | **Esito** |
| TC\_3.3\_1 | EDBL1 | Errore: L’email inserita non è presente nel database |
| TC\_3.3\_2 | EDBL2,FEL1 | Errore: Il formato dell’e-mail non è corretto |
| TC\_3.3\_3 | EDBL2,FEL2,LEL1 | Errore: L’e-mail non è stata inserita |
| TC\_3.3\_4 | EDBL2,FEL2,LEL2 | Errore: L’e-mail e troppo lunga |
| TC\_3.3\_5 | EDBL2,FEL2,LEL3,PDBL1 | Errore: La password non è presente nel database |
| TC\_3.3\_6 | EDBL2,FEL2,LEL3,PDBL2,LPL1 | Errore: La password non è stata inserita |
| TC\_3.3\_7 | EDBL2,FEL2,LEL3,PDBL2,LPL2 | Errore: La password è troppo lunga |
| TC\_3.3\_8 | EDBL2,FEL2,LEL3,PDBL2,LPL3 | Corretto |

9.3 Gestione Acquisti

|  |  |  |
| --- | --- | --- |
| **Parametro: Username** | | |
| **FORMATO:**  **^[A-Za-z0-9-\_]\*$** | | |
| **Nome Categoria** | **Scelte per categorie** | |
| Lunghezza [LU] | 1. Lunghezza>30 = false[error] 2. Lunghezza<1 = false[error] 3. 1<Lunghezza<30 = true[PROPERTY LU OK] | |
| Formato [FU] | 1. Rispetta il formato = false [error]  2. Rispetta il formato = true [PROPERTY FU\_OK] | |
| [CU] | 1. Username non presente nel database = false[error] 2. Username non presente nel database = true[PROPERTY CU\_OK] | |
| **Parametro: Nome** | | |
| **FORMATO:**  **/^[A-Za-z \']\*$** | | |
| **Nome Categoria** | **Scelte per categorie** | |
| Lunghezza [LN] | 1. Lunghezza>30 = false[error] 2. Lunghezza<1 = false[error] 3. 1<Lunghezza<=30= true[PROPERTY LN OK] | |
| Formato [FN] | 1. Rispetta il formato = false [error]  2. Rispetta il formato = true [PROPERTY FN\_OK] | |
| **Parametro: Cognome** | | |
| **FORMATO:**  **/^[A-Za-z \']\*$** | | |
| Lunghezza [LC] | | 1. Lunghezza>30 = false[error] 2. Lunghezza<1 = false[error] 3. 1<Lunghezza<=30= true[PROPERTY LC OK] |
| Formato [FC] | | 1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FC\_OK] |

1. Testing schedule

Le attività di pianificazione del testing avverranno successivamente alla stesura dei documenti di System Design e Object Design.

La scrittura dei casi di test avverrà in contemporanea con lo sviluppo del codice.

L’esecuzione dei test verrà svolta durante l’implementazione del sistema e successivamente verranno rieseguiti tutti i casi di test quando l’implementazione del sistema sarà terminata per garantirne il corretto funzionamento.