



TP

Test Plan



MUSEUM MANAGEMENT **SYSTEM**

Riferimento	NC15_TP_v1
Versione	1.0
Data	06/01/2024
Destinatario	Prof. Carmine Gravino
Presentato da	Team NC15
Approvato da	--



Sommario

Team Members	3
Revision History.....	3
1. Introduzione.....	4
1.1 Obiettivo	4
2. Relazione con altri documenti	4
2.1 Relazioni con il Requirements Analysis Document (RAD)	4
2.2 Relazioni con il System Design Document (SDD)	4
3. Panoramica del sistema	5
4. Features da testare	5
5. Pass/Fail criteria	5
6. Approccio.....	6
7. Sospensione e ripristino.....	6
8. Test cases.....	7
9. Strumenti per il testing.....	12



Team Members

Nome	Cognome	Acronimo	Contatto
Salvatore	Nocera	SN	s.nocera7@studenti.unisa.it
Angelo	Barone	AB	a.barone69@studenti.unisa.it
Vincenzo	Bucciero	VB	v.bucciero2@studenti.unisa.it

Revision History

Data	Versione	Descrizione	Autori
06/01/2025	0.1	Prima stesura	VB
08/01/2025	0.2	Definizione dei Test Case	SN - VB - AB
28/1/2025	0.5	Aggiunta paragrafo 9	VB
3/2/2025	1.0	Revisione	SN – VB - AB



1. Introduzione

1.1 Obiettivo

Museum Management System si propone di semplificare le interazioni tra un museo e i rispettivi visitatori, al fine di facilitare l'organizzazione di mostre verso il pubblico e della gestione dell'inventario di opere e del personale interni alla struttura.

Il documento di Test Plan ha l'obiettivo di descrivere ed analizzare le attività di Testing per la piattaforma Museum Management System. Il fine è quello di garantire che ogni funzionalità produca i risultati attesi.

All'interno del documento sono riportate le strategie di testing adottate e gli strumenti scelti per la rilevazione degli errori, con lo scopo di presentare al cliente finale una piattaforma priva malfunzionamenti.

2. Relazione con altri documenti

Per la corretta individuazione dei test case, si fa riferimento agli altri documenti.

2.1 Relazioni con il Requirements Analysis Document (RAD)

I test case sono pianificati in relazione ai requisiti funzionali e non funzionali presentati nel RAD.

2.2 Relazioni con il System Design Document (SDD)

I test case sono pianificati in modo da rispettare la suddivisione in sottosistemi presentata nell'SDD.

3. Panoramica del sistema

Il sistema proposto basa la sua architettura sul sistema three-tier.

Verranno usati HTML5 e CSS3 per la parte di front-end e la generazione della presentation.

Per la logica applicativa e quindi il back-end sarà utilizzato Java EE.

Per la gestione del database saranno usati:

- MySQL JDBC per il collegamento al database.
- MySQL per il database su cloud in fase di deployment.

4. Features da testare

Le funzionalità da testare sono relative agli Use Case definiti nel RAD:

- **Visitatore**
 - VIS - Registrazione
- **Utente**
 - US - Acquisto Biglietto
- **Personale**
 - PER - Aggiunta Opere
- **Admin**
 - AD - Creazione avvisi

5. Pass/Fail criteria

Le attività di testing sono mirate ad identificare la presenza di faults (errori) all'interno del sistema, per effettuare un successivo intervento di eliminazione.

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.



6. Approccio

Il test che si effettuerà sul sistema si compone di 3 fasi:

- Testing di unità

6.1 Testing di Unità

Tramite Category Partition, di esattamente un metodo di una classe sviluppata. La strategia prevede di utilizzare un approccio Black-Box, attraverso l'uso del framework per il testing di classi Java JUnit.

7. Sospensione e ripristino

CRITERI DI SOSPENSIONE:

Il processo di testing proseguirà senza interruzioni fino al suo completamento, anche nel caso di rilevazione di una failure. L'eventuale sospensione temporanea del testing potrebbe verificarsi solo in presenza di un errore rilevato durante l'esecuzione nella definizione di uno dei test stessi.

CRITERI DI RIPRISTINO:

Il processo di testing riprenderà una volta che tutti i problemi legati alla sospensione saranno stati risolti. L'attività di testing ricomincerà dal Test Case che ha causato l'interruzione.

8. Test cases

L'approccio al testing verrà effettuato utilizzando il metodo Category Partition. L'approccio permette di effettuare il testing solo su combinazioni di input significative individuate a partire da requisiti ben definiti, questo permette di avere una copertura massima avendo risorse di tempo e budget limitate.

○ Registrazione - VIS

Parametro: Full Name	
Formato: /^[a-zA-Z]*\$/	
Nome Categoria	Scelte per la categoria
Lunghezza [LN]	1. Lunghezza > 50 [Error] 2. Lunghezza <= 50 [PROPERTY_LN_OK]
Formato [FNN]	1. Non rispetta il formato = false[error] 2. Rispetta il formato = true [PROPERTY_FNN_OK]
Parametro: Password	
Formato: /^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9]).{6,}\$	
Nome Categoria	Scelte per la categoria
Lunghezza [LP]	1. Lunghezza < 6 Lunghezza > 255 [Error] 2. Lunghezza >= 6 Lunghezza <= 255 [PROPERTY_LP_OK]
Formato [FNP]	1. Non rispetta il formato = false[error] 2. Rispetta il formato = true [PROPERTY_FNP_OK]
Parametro: Conferma Password	
Nome Categoria	Scelte per la categoria
Match [MCP]	1. Match con password = false [error] 2. Match con password = true [PROPERTY_MCP_OK]
Parametro: E-mail	
Formato: /^[?!\\.\[\w-_.]*[^\.])?(@\w+)(\.\w+(\.\w+)?[^\.W])\$/	
Nome	Scelte per la categoria
Lunghezza [LE]	1. Lunghezza > 100 [Error] 2. Lunghezza <= 100 [PROPERTY_LP_OK]



Formato [FNE]	3. Non rispetta il formato = false[error] 4. Rispetta il formato = true [PROPERTY FNE_OK]
Match [ME]	1. Email match in DB == true [Error] 2. Email match in DB == false [PROPERTY_ME_OK]
Parametro: Numero di telefono	
Formato: <code>/^(\+ d{1,4}[-.s]?)?(?d{1,3})?[-.s]?(d{1,4}[-.s]?)?{2,3}d{1,9}\$</code>	
Nome	Scelte per la categoria
Formato [FNT]	1. Non rispetta il formato = false[error] 2. Rispetta il formato = true [PROPERTY FNT_OK]
Parametro: Card Number	
Formato: <code>^d{16}\$</code>	
Nome	Scelte per la categoria
Formato [FNC]	1. Non rispetta il formato = false[error] 2. Rispetta il formato = true [PROPERTY FNC_OK]
Parametro: Card Expiry Date	
Formato: <code>^(0[1-9] 1[0-2])\/(0[1-9] 12)[0-9] 3[01]\$</code>	
Nome	Scelte per la categoria
Formato [FND]	1. Non rispetta il formato = false[error] 2. Rispetta il formato = true [PROPERTY FND_OK]
Parametro: Card Expiry Secret Code	
Formato: <code>^d{3}\$</code>	
Nome	Scelte per la categoria
Formato [FNS]	1. Non rispetta il formato = false[error] 2. Rispetta il formato = true [PROPERTY FNS_OK]



Test Case ID	Test frame	Esito
TC_VIS_1	LN1	ERROR: lunghezza nome non rispettata.
TC_VIS_2	LN2, FNN1	ERROR: formato nome non rispettato.
TC_VIS_3	LN2, FNN2, FNP1	ERROR: formato password non rispettato.
TC_VIS_4	LN2, FNN2, FNP2, MCP1	ERROR: la password di conferma non coincide.
TC_VIS_5	LN2, FNN2, FNP2, MCP2, FNE1	ERROR: l'e-mail non rispetta il formato.
TC_VIS_6	LN2, FNN2, FNP2, MCP2, FNE2, FNT1	ERROR: Il numero di telefono non rispetta il formato.
TC_VIS_7	LN2, FNN2, FNP2, MCP2, FNE2, FNT2, FNC1	ERROR: Il numero della carta di credito non rispetta il formato.
TC_VIS_8	LN2, FNN2, FNP2, MCP2, FNE2, FNT2, FNC2, FND1	ERROR: La data di scadenza della carta di credito non rispetta il formato.
TC_VIS_9	LN2, FNN2, FNP2, MCP2, FNE2, FNT2, FNC2, FND2, FNS1	ERROR: Il CVV non rispetta il formato.
TC_VIS_10	LN2, FNN2, FNP2, MCP2, FNE2, FNT2, FNC2, FND2, FNS2, ME1	ERROR: L'email esiste già nel database.
TC_VIS_11	LN2, FNN2, LP2, FNP2, MCP2, FNE2, FNT2, FNC2, FND2, FNS2, ME2	CORRETTO: la registrazione va a buon fine.



○ **Acquisto Biglietti - US**

Parametro: Password	
Nome	Scelte per la categoria
Match [MP]	1. Password match in DB == true [Error] 2. Password match in DB == false [PROPERTY_MP_OK]

Test Case ID	Test frame	Esito
TC_US_1	MP1	ERROR: la password non è corretta.
TC_US_2	MP2	CORRETTO: la password è corretta.

○ **Aggiunta Opere - PER**

Parametro: Art Name	
Nome categoria	Scelte per la categoria
Lunghezza[LN]	1. Lunghezza > 50 [error] 2. Lunghezza <= 50 [PROPERTY LN_OK]
Parametro: Descrizione	
Nome categoria	Scelte per la categoria
Lunghezza[LD]	1. Lunghezza >200 [error] 2. Lunghezza <= 200 [PROPERTY LD_OK]
Parametro: Artista	
Nome categoria	Scelte per la categoria
Lunghezza[LA]	1. Lunghezza > 50 [error] 2. Lunghezza <= 50 [PROPERTY LA_OK]
Parametro: Lunghezza	
Nome categoria	Scelte per la categoria
Lunghezza[LLU]	1. Lunghezza > 50 [error] 2. Lunghezza <= 50 [PROPERTY LLU_OK]
Parametro: Altezza	



Nome categoria	Scelte per la categoria
Lunghezza[LAL]	1. Lunghezza > 50 = false[error] 2. Lunghezza <= 50 = true[PROPERTY LLU_OK]
Parametro: Immagine	
Nome categoria	Scelte per la categoria
Lunghezza[LIM]	1. Lunghezza >100 = false[error] 2. Lunghezza <=100 = true[PROPERTY LIM_OK]

Test Case ID	Test frame	Esito
TC_PER_1	LN1	ERROR: il nome dell'opera è troppo lungo.
TC_PER_2	LN2, LD1	ERROR: la descrizione dell'opera è troppo lunga.
TC_PER_3	LN2, LD2, LA1	ERROR: il nome dell'autore è troppo lungo.
TC_PER_4	LN2, LD2, LA2, LLU1	ERROR: la lunghezza non rispetta il formato
TC_PER_5	LN2, LD2, LA2, LLU2, LAL1	ERROR: l'altezza non rispetta il formato
TC_PER_6	LN2, LD2, LA2, LLU2, LAL2, LIM1	ERROR: l'URL dell'immagine dell'opera è troppo lungo.
TC_PER_7	LN2, LD2, LA2, LLU2, LAL2, LIM2	CORRETTO

○ **Creazione avvisi - AD**

Parametro: Titolo	
Nome categoria	Scelte per la categoria
Lunghezza[LT]	1. Lunghezza > 50 [error] 2. Lunghezza <= 50 [PROPERTY LT_OK]
Parametro: Oggetto	
Nome categoria	Scelte per la categoria
Lunghezza[LO]	1. Lunghezza > 50 [error] 2. Lunghezza <= 50 [PROPERTY LO_OK]
Parametro: Contenuto	
Nome categoria	Scelte per la categoria
Lunghezza[LC]	1. Lunghezza > 200 [error] 2. Lunghezza <=200 [PROPERTY LC_OK]

Test Case ID	Test frame	Esito
TC_AD_1	LT1	ERROR: il titolo supera la lunghezza limite.
TC_AD_2	LT2, LO1	ERROR: l'oggetto supera la lunghezza limite.
TC_AD_3	LT2, LO2, LC1	ERROR: il contenuto supera la lunghezza limite.
TC_AD_4	LT2, LO2, LC2	CORRETTO

9. Strumenti per il testing

Per condurre l'attività di testing, è indispensabile disporre di un computer con un Browser Web e il container Tomcat. Nello svolgimento del testing è necessario: Junit.

In aggiunta, abbiamo bisogno di una simulazione del sistema che comprenda tutte e quattro le funzionalità.