



Laurea Triennale in informatica-Università di Salerno  
Corso di Ingegneria del Software- Prof. C. Gravino

# Test Summary Report

## Progetto

# SalernArte

Versione	0.1
Data	17/02/2023
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

## Revision History

Data	Versione	Descrizione	Autori
17/02/2023	0.1	Prima stesura: struttura del documento	LM
17/02/2023	0.2	Revisione documento	LM,DPA

## Sommario

Membri del Team .....	2
Revision History .....	2
1. Introduzione .....	2
2. Relazione con gli altri documenti .....	3
3. Testing Unitario e di Integrazione .....	3
4. Testing di sistema .....	4

## 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.

Nel documento di Test Plan abbiamo descritto le modalità di Testing che sono state scelte e che sono state utilizzate per la piattaforma SalernArte.

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 Carrello

## 2. Relazione con gli altri documenti

---

Viene riportata di seguito le relazioni che questo documento ha relativamente agli altri documenti di Testing prodotti:

- [Test Plan](#) , all'interno del documento di Test Plan sono state definite le modalità di esecuzione dei test e tramite la rappresentazione dei Test Frame relativi al Category Partition è stato possibile generare l'esecuzione dei test di sistema da cui il report presente in questo documento dipende.
- [Test Case Specification](#) , all'interno di questo documento sono state descritte nello specifico le esecuzioni dei Test di Sistema che sono state implementate tramite SeleniumIDE e la cui analisi ci ha permesso di generare il documento di report.
- [Test Incident Report](#) , effettua una panoramica riassuntiva del testing di sistema effettuato e che viene ripreso dal Test Summary Report.

## 3. Testing Unitario e di Integrazione

---

Durante la stesura delle classi di test abbiamo proceduto a suddividere il lavoro da svolgere nelle due tipologie di classi di test. Ogni membro del team aveva il compito di lavorare nella Gestione assegnata e implementare le classi di testing che gli appartenevano; si procedeva con lo sviluppo delle classi di test d'unità e infine con quelle d'integrazione.

Si procedeva poi alla verifica del testing delle classi sviluppate: nel caso in cui le classi di test implementate contenessero degli errori, il team member proprietario della classe procedeva alla correzione degli errori e infine effettuava il push sulla repository GitHub.












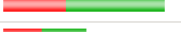





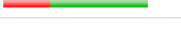






















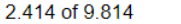

Quando si effettuava la pull request il sistema eseguiva i test che erano stati caricati per verificarne la correttezza.

Nel percorso "DOCUMENTAZIONE\Test d'unità e integrazione" sono reperibili tutti i report dell'esecuzione dei test; il report della coverage presentata di seguito è reperibile all'interno del file [index.html](#).

Per dimostrare la raggiunta della Branche Coverage abbiamo utilizzato il tool JaCoCo che ha restituito i seguenti risultati:



## SalernArteWebsite

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
 <a href="#">model.dao</a>		84%		75%	39	153	197	1.149	4	74	0	9
 <a href="#">model.entity</a>		60%		100%	37	132	68	293	37	131	0	10
 <a href="#">gestioneEventi.controller</a>		73%		55%	28	41	38	168	0	5	0	1
 <a href="#">autenticazione.service</a>		72%		61%	50	84	39	130	4	13	0	1
 <a href="#">autenticazione.controller</a>		70%		54%	27	47	39	139	2	10	0	2
 <a href="#">gestioneEventi.service</a>		81%		67%	38	89	37	160	6	25	0	1
 <a href="#">gestioneAcquisti.utils</a>		0%		n/a	6	6	39	39	6	6	2	2
 <a href="#">gestioneAcquisti.controller</a>		43%		18%	18	24	33	62	1	5	0	1
 <a href="#">gestioneAcquisti.service</a>		77%		57%	37	56	22	112	2	11	0	1
 <a href="#">gestioneEventi.utils</a>		0%		0%	5	5	18	18	3	3	1	1
 <a href="#">registrazione.service</a>		85%		68%	27	53	11	69	1	7	0	1
 <a href="#">utils</a>		0%		n/a	5	5	14	14	5	5	2	2
 <a href="#">registrazione.controller</a>		85%		54%	9	16	10	69	2	5	0	1
 <a href="#">singleton</a>		94%		100%	1	3	1	15	1	2	0	1
Total	2.414 of 9.814	75%	313 of 823	61%	327	714	566	2.437	74	302	5	34

## 4. Testing di sistema

Terminata la fase di testing d'unità e di integrazione abbiamo effettuato un test completo del sistema delle funzionalità testate utilizzando il tool Selenium IDE, per Chrome. Per ogni Gestione individuata e assegnata in precedenza ad ogni team member abbiamo generato una Test Suite dove all'avvio del sistema si settava l'ambiente in modo che la pre-condizione fosse verificata. Al termine dell'esecuzione, una volta controllato l'Assert assegnato, è stato resettato il sistema alla formazione originale per non intaccare i successivi test case.

Ogni volta che si verificava un errore del test case si procedeva ad individuare l'esecuzione errata e, una volta modificata, si procedeva con l'esecuzione della Test Suite corrispondente.

Per motivi di leggibilità il report effettivo di tutte le esecuzioni e gli esiti del test di sistema sono stati riportati in forma tabellare all'interno di un file excel disponibile insieme al resto della documentazione e reperibile anche tramite il seguente link:

- [Test Incident Report](#)