



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

# 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

## Revision History

Data	Versione	Descrizione	Autori
28/11/2022	0.1	Prima stesura: struttura del documento	DPA, ML, LM
30/11/2022	0.2	Funzionalità da testare e non: gestione utente	ML
1/12/2022	0.3	Gestione Eventi: Funzionalità da testare e non, Test cases: richiesta inserimento evento, richiesta modifica evento e ricerca evento	DPA
02/12/2022	0.4	Gestione Carrello: funzionalità da testare e non Gestione Utente: funzionalità da testare e non, Test Cases: registrazione organizzatore ed utente, login, modifica profilo organizzatore	LM
03/12/2022	0.5	Gestione Eventi: Terminati tutti i test cases	DPA
05/12/2022	0.6	Modifica profilo amministratore: funzionalità da testare e non Registrazione scolaresca: funzionalità da testare e non	ML
08/01/2023	0.7	Modificata tabella modifica profilo amministratore, creazione tabella modifica profilo utente	ML
30/01/2023	0.8	Revisione del documento ed aggiunta delle sezioni mancanti e di autori all'interno delle tabelle	DPA



06/02/2023	0.9	Revisione del documento ed aggiunta delle sezioni mancanti e di autori all'interno delle tabelle	LM
17/02/2023	1.0	Revisione documento	LM, DPA

## Sommario

Membri del Team .....	2
Revision History .....	2
1. Introduzione .....	4
2. Relazione con gli altri documenti .....	4
3. Panoramica del sistema .....	5
4. Funzionalità da testare e non testare .....	5
4.1 Gestione Utente .....	5
4.2 Gestione Eventi .....	5
4.3 Gestione Carrello .....	6
5. Pass/Fail criteria .....	6
6. Approccio .....	6
6.1 Testing di unità .....	7
6.2 Testing funzionale .....	<b>Errore. Il segnalibro non è definito.</b>
6.3 Testing di Sistema .....	<b>Errore. Il segnalibro non è definito.</b>
6.4 Testing di integrazione .....	<b>Errore. Il segnalibro non è definito.</b>
7. Sospensione e ripristino .....	9
8. Strumenti per il testing (hardware/software) .....	9
9. Test cases .....	9
9.1 Gestione Utente .....	9
9.1.1 Registrazione Utente .....	9
9.1.2 Modifica profilo utente .....	12
9.1.3 Login .....	15
9.1.4 Modifica Profilo Organizzatore .....	16
9.1.5 Registrazione Organizzatore .....	19
9.1.6 Modifica Profilo Amministratore .....	22
9.1.7 Registrazione Scolaresca .....	24
9.1.8 Modifica profilo scolaresca .....	25



9.2 Gestione Eventi.....	27
9.2.1 Richiesta inserimento Evento.....	27
9.2.2 Richiesta modifica Evento .....	31
9.2.3 Ricerca Evento .....	35
9.3 Gestione Carrello .....	35
9.3.1 Aggiungi al carrello .....	35
9.3.2 Modifica Carrello.....	36
10. Testing schedule.....	36

## 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.

Questo documento ha come obiettivo quello di descrivere le modalità di Testing che sono state scelte e che verranno utilizzate per la piattaforma SalernArte. Nello specifico andremo quindi a descrivere i criteri che sono stati utilizzati per scegliere le funzionalità da testare, in che modo queste attività sono state divise tra i componenti del gruppo, descriveremo l'approccio scelto per il Testing della piattaforma per poi entrare nel dettaglio dei TestCase individuati e di vari criteri che andranno seguiti.

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

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

- [Requirements Analysis Document](#) (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](#) (SDD), da cui verrà presa la suddivisione in sottosistemi che dovrà essere rispettata all'interno dei test case.



- [Object Design Document](#) (ODD), per quanto riguarda i test di unità e di integrazione la cui suddivisione è maggiormente legata al documento di ODD e nello specifico alla suddivisione del sistema in sottosistemi, essi saranno scritti e documentati all'interno del codice applicativo.

### 3. 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.

### 4. Funzionalità da testare e non testare

---

Le funzionalità che non verranno testate sono quelle che non prevedono l'inserimento di input da parte dell'utente - ad esempio attività riguardanti esclusivamente visualizzazioni di dati oppure l'utilizzo di un bottone. I requisiti funzionali di priorità alta che non vengono testati in questa fase per i motivi sopracitati, verranno testati successivamente con diverse tecniche e saranno quindi trattati in una diversa documentazione.

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

#### 4.1 Gestione Utente

Le funzionalità che saranno testate relative alla Gestione Utente sono le seguenti:

- Registrazione utente
- Registrazione scolaresca
- Registrazione organizzatore
- Modifica profilo utente
- Modifica profilo organizzatore
- Modifica profilo scolaresca
- Modifica profilo amministratore
- Login

#### 4.2 Gestione Eventi

Le funzionalità che saranno testate relative alla Gestione Eventi sono le seguenti:

- Richiesta inserimento evento



- Richiesta modifica evento
- Ricerca evento

### 4.3 Gestione Carrello

Le funzionalità che saranno testate relative alla Gestione Carrello saranno:

- Aggiungi al carrello
- Modifica carrello

## 5. Pass/Fail criteria

La fase di testing ha come scopo quello di provare il sistema al fine di controllare che i requisiti stabiliti con il cliente siano stati rispettati e di individuare quanti più errori possibili che saranno quindi successivamente corretti. Ricordiamo che è possibile provare la presenza di errori in un sistema, ma non la loro assenza.

Sulla base dei requisiti individuati nella fase di analisi verranno scritti i casi di test (Test cases), che saranno valutati utilizzando un oracolo che rappresenta il comportamento atteso che dovrà essere confrontato con il risultato effettivo dell'esecuzione.

Un test è considerato un successo (pass) se, per un determinato input, il risultato atteso è diverso da quello osservato ed è stato quindi rilevato un malfunzionamento.

Un test verrà considerato un fallimento (fail) se, per un determinato input, il risultato atteso e quello osservato coincidono e non è stato rilevato un malfunzionamento.

Il testing sarà considerato valido nel momento in cui i vincoli elencati di seguito saranno stati tutti rispettati:

- Testare tutti i requisiti funzionali di alta e media priorità
- Raggiungere una branch coverage non inferiore al 75%

## 6. Approccio

Possiamo dividere le fasi di Test del sistema rappresentato dalla piattaforma SalernArte in 3 fasi: Test di Sistema, Testi di Integrazione e Test di Unità.

La progettazione delle attività di test avverrà seguendo l'ordine in cui sono state presentate le fasi di test, ma i test verranno successivamente eseguiti seguendo l'ordine inverso: ossia effettuando prima Testing di Unità, poi di Integrazione ed infine Test di Sistema.

Successivamente alla stesura del Requirements Analysis Document verrà progettato il Test di Sistema tramite l'analisi dei Requisiti individuati.

Successivamente alla stesura del System Design Document ed alla conseguente divisione dell'architettura del sistema in sottosistemi e layer, verranno prese decisioni relative alla progettazione dei Test di Unità e di Integrazione.

Durante la fase di implementazione verranno eseguite periodiche attività di revisione del codice eseguite dall'intero team tramite meet settimanali organizzati sulla piattaforma Teams. Per ogni sottosistema



implementato verranno prodotte prima le classi che effettueranno il Test di Unità e successivamente quelle relative al Test di Integrazione.

Dopo la fase di implementazione e del Testing di Unità e di Integrazione, verrà eseguito il Test di Sistema .

## 6.1 Testing di Sistema

Il Test di Sistema sarà eseguito con il supporto del tool Selenium IDE, che permette di automatizzare l'esecuzione del test di sistema registrando le azioni che un utente può eseguire all'interno del browser e permettendo di definire un oracolo per ogni esecuzione. Verrà fatto il deploy del sistema in localhost per permettere di eseguire i test di sistema.

A causa del basso budget a disposizione, non si assicura l'esecuzione delle seguenti categorie di Testing di Sistema:

- Performance Testing
- Pilot Testing
- Acceptance Testing
- Installation Testing

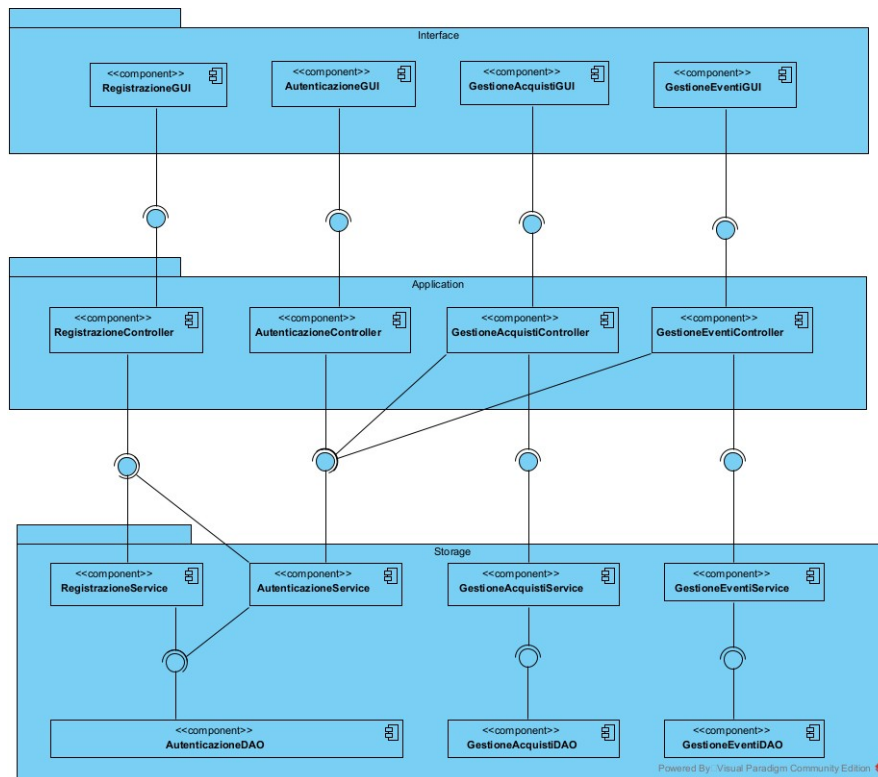
## 6.1 Testing funzionale

L'obiettivo di del test Funzionale è la validazione dei requisiti funzionali individuati in fase di Analisi Dei Requisiti. Tramite l'analisi dei requisiti funzionali vengono individuati i Test Case che andremo a definire in questo documento, che saranno scelti tra le funzionalità chiave del sistema e tra quelle che hanno più probabilità di generare fault, ossia quelle che prevedono l'inserimento di input da parte dell'utente.

## 6.2 Testing di Integrazione

Per l'esecuzione del Test di Integrazione verrà utilizzato un approccio di tipo Bottom Up, scelto perché è la strategia che meglio si adatta a sistemi di tipo Object-oriented come il nostro. La scrittura delle classi di test avverrà tramite l'utilizzo del framework *JUnit*, mentre per effettuare lo Stub delle componenti si utilizzerà la funzionalità di Mock fornita dalla libreria *Mockito* . Verrà utilizzato *Maven* per l'esecuzione automatizzata delle classi di test ed infine per il calcolo della coverage del sistema verrà utilizzato *JaCoCo*.

Facendo riferimento al Diagramma Architettuale definito all'interno del System Design Document e che verrà riportato di seguito:



Il Test di Integrazione verrà effettuato a partire dal layer Storage dove verrà effettuata l'integrazione della classe service con la rispettiva componente DAO da cui dipende, dopodichè si salirà fino al layer Application dove verrà testata la classe controller integrando la componente service da cui dipende. La chiamata alla classe controller (e quindi la generazione della request che chiama la classe controller) verrà simulata utilizzando Mockito.

Il nome delle classi di test seguiranno la seguente regola: Presa una classe di nome *NomeClasse*, la rispettiva classe che effettuerà il test di integrazione sarà nominata *NomeClasseTestIntegrazione*.

### 6.3 Testing di Unità

Per l'esecuzione del Test di Unità si è deciso di utilizzare un approccio di tipo BlackBox, eseguendo quindi il testing delle operazioni che implementano le funzionalità individuate durante l'analisi dei requisiti e nello specifico dando priorità alle funzionalità individuate precedentemente tramite i Test Case che, come abbiamo detto, sono state scelte tra quelle funzionalità con il più alto rischio di generare fault a causa della necessità dell'inserimento di input da parte dell'utente. Successivamente si proseguirà con il test delle restanti funzionalità di priorità alta e media.

Nello specifico i metodi di test che faranno riferimento ai Test Case individuati, avranno come nome del metodo di test il TestCaseID corrispondente, sostituendo al carattere “.”, non utilizzabile all'interno del nome di un metodo, il carattere “p”.

Di seguito un esempio:

- TestCaseID: TC\_1.1\_1
- firma metodo: void TC\_1p1\_1();





Il nome delle classi di test seguiranno la seguente regola: Presa una classe di nome *NomeClasse* , la rispettiva classe che effettuerà il test di Unità sarà nominata *NomeClasseTest*.

Seguendo sempre la logica del Testing di tipo Bottom Up, verranno testate per unità le classi all'interno di ogni layer, prima di essere integrate. Verrà escluso il testing delle interfacce e delle classi di tipo Entity, queste ultime poiché fornite solo di metodi come getters, setters e toString.

Anche in questa fase come nella precedente è previsto:

- l'utilizzo del framework *JUnit*;
- l'utilizzo di *Mockito* per isolare la componente testata generando opportuni stub e driver;
- L'utilizzo di *Maven* e *JaCoCo* per l'esecuzione automatizzata dei test e la generazione del file di report e della branch coverage del sistema

## 7. Sospensione e ripristino

### Criteri di sospensione

Il testing verrà eseguito fino alla sua completa terminazione anche nel caso in cui venissero generati dei failure. Solo nel caso in cui un errore nella definizione della classe di test dovesse generare un'interruzione del test al momento dell'esecuzione, allora il test potrà essere momentaneamente sospeso per procedere alla correzione della classe di test.

### Criteri di ripristino

Nel momento in cui gli errori individuati nelle classi di test sono stati risolti, il test dovrà riprendere la sua esecuzione.

## 8. Strumenti per il testing (hardware/software)

L'hardware utilizzato per il testing è un computer, in quanto il sistema non è stato ancora rilasciato.

## 9. Test cases

Per la definizione dei Test Frame sarà utilizzato l'approccio di Category Partition studiato. Verranno partizionati gli input in classi di equivalenza che permetteranno di ridurre il numero di test case trattati. L'output atteso sarà definito da un oracolo umano, derivato dall'analisi delle specifiche dei requisiti e dai casi d'uso.

### 9.1 Gestione Utente

#### 9.1.1 Registrazione Utente

Autore	LM
Parametro: Nome	



FORMATO: ^[a-zA-Z\u00C0-\u00ff]{1,50}\$	
Nome Categoria	Scelte per categorie
Formato [FN]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FN_OK]
Parametro: Cognome	
FORMATO: ^[a-zA-Z\u00C0-\u00ff]{1,50}\$	
Nome Categoria	Scelte per categorie
Formato [FC]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FC_OK]
Parametro: Sesso	
FORMATO: "Uomo" "Donna" "Altro"	
Nome Categoria	Scelte per categorie
Formato [FS]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FS_OK]
Parametro: Data di nascita	
FORMATO: [GG/MM/AAAA]	
Nome Categoria	Scelte per categorie
Formato [FD]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FD_OK]
Inserimento [DS]	1. Precedente alla data odierna [if <b>formatoFDok</b> && dataDiNascita < dataOdierna] [property <b>inserimentoDSok</b> ] 2. Uguale alla data odierna [if <b>formatoFDok</b> && dataDiNascita == dataOdierna] [error]



	3. Successiva alla data odierna [if <b>formatoFDok</b> && dataDiNascita > dataOdierna] [error]
<b>Parametro: E-mail</b>	
<b>FORMATO:</b> <code>^\w+([\.\-]? \w+)*@\w+([\.\-]? \w+)*(\.\w+){1,100}\$</code>	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Formato [FE]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE_OK]
<b>Parametro: Password</b>	
<b>FORMATO:</b> <code>^(?=.*[A-Z])(?=.*\d)(?=.*\W){6,30}\$</code>	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Formato [FP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FP_OK]
<b>Parametro: ConfermaPassword</b>	
<b>FORMATO:</b> <code>^(?=.*[A-Z])(?=.*\d)(?=.*\W){6,30}\$</code>	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Match [MCP]	1. Match con password= false [error] 2. Match con password= true [PROPERTY MP OK]
Formato [FCP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FPC OK]

Test case ID	Test Frame	Esito
TC_1.1_1	FN1	Errore: Nome non valido
TC_1.1_2	FN2 , FC1	Errore: Cognome non



		valido
TC_1.1_3	FN2, FC2, FS1	Errore: Sesso non valido
TC_1.1_4	FN2, FC2, FS2, FD1	Errore: Data di nascita non valida
TC_1.1_5	FN2, FC2, FS2, FD2, DS2	Errore: Data di nascita uguale alla data odierna
TC_1.1_6	FN2, FC2, FS2, FD2, DS3	Errore: Data di nascita successiva alla data odierna
TC_1.1_7	FN2, FC2, FS2, FD2, DS1, FE1	Errore: E-mail non valida
TC_1.1_8	FN2, FC2, FS2, FD2, DS1, FE2, FP1	Errore: Password non valida
TC_1.1_9	FN2, FC2, FS2, FD2, DS1, FE2, FP2, MCP1	Errore: Le password non coincidono
TC_1.1_10	FN2, FC2, FS2, FD2, DS1, FE2, FP2, MCP2, FCP1	Errore: conferma password non valido
TC_1.1_11	FN2, FC2, FS2, FD2, DS1, FE2, FP2, MCP2, FCP2	Corretto

### 9.1.2 Modifica profilo utente

Autore		ML
Parametro: Nome		
FORMATO: ^[ a-zA-Z\ u00C0-\ u00ff]{1,50}\$		
Nome Categoria	Scelte per categorie	
Formato [FN]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FN_OK]	
Parametro: Cognome		
FORMATO: ^[ a-zA-Z\ u00C0-\ u00ff]{1,50}\$		
Nome Categoria	Scelte per categorie	
Formato [FC]	3. Rispetta il formato = false [error]	



	4. Rispetta il formato = true [PROPERTY FC_OK]
<b>Parametro: Sesso</b>	
<b>FORMATO:</b> "Uomo" "Donna" "Altro"	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Formato [FS]	3. Rispetta il formato = false [error] 4. Rispetta il formato = true [PROPERTY FS_OK]
<b>Parametro: Data di nascita</b>	
<b>FORMATO:</b> [GG/MM/AAAA]	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Formato [FD]	3. Rispetta il formato = false [error] 4. Rispetta il formato = true [PROPERTY FD_OK]
Inserimento [DS]	4. Precedente alla data odierna [if <b>formatoFDok</b> && dataDiNascita < dataOdierna] [property <b>inserimentoDSok</b> ] 5. Uguale alla data odierna [if <b>formatoFDok</b> && dataDiNascita == dataOdierna] [error] 6. Successiva alla data odierna [if <b>formatoFDok</b> && dataDiNascita > dataOdierna] [error]
<b>Parametro: E-mail</b>	
<b>FORMATO:</b> <b>^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w+{1,100}\$</b>	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Formato [FE]	3. Rispetta il formato = false [error] 4. Rispetta il formato = true [PROPERTY FE_OK]



Parametro: VecchiaPassword	
FORMATO: $\wedge(?=.*[A-Z])(?=.*\d)(?=.*\W).\{6,30\}\$$	
Nome Categoria	Scelte per categorie
Formato [FP]	3. Rispetta il formato = false [error] 4. Rispetta il formato = true [PROPERTY FP_OK]
Parametro: NuovaPassword	
FORMATO: $\wedge(?=.*[A-Z])(?=.*\d)(?=.*\W).\{6,30\}\$$	
Nome categoria	Scelte per categorie
Formato [FG]	5. Rispetta il formato = false [error] Rispetta il formato = true [PROPERTY FP_OK]
Parametro: ConfermaNuovaPassword	
FORMATO: $\wedge(?=.*[A-Z])(?=.*\d)(?=.*\W).\{6,30\}\$$	
Nome Categoria	Scelte per categorie
Match [MCP]	3. Match con password= false[error] 4. Match con password= true [PROPERTY MP OK]
Formato [FCP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FPC OK]

Test case ID	Test Frame	Esito
TC_1.2_1	FN1	Errore: Nome non valido
TC_1.2_2	FN2, FC1	Errore: Cognome non valido



TC_1.2_3	FN2, FC2, FS1	Errore: Sesso non valido
TC_1.2_4	FN2, FC2, FS2, FD1	Errore: Data di nascita non valida
TC_1.2_5	FN2, FC2, FS2, FD2, DS2	Errore: Data di nascita uguale alla data odierna
TC_1.2_6	FN2, FC2, FS2, FD2, DS3	Errore: Data di nascita successiva alla data odierna
TC_1.2_7	FN2, FC2, FS2, FD2, DS1, FE1	Errore: E-mail non valida
TC_1.2_8	FN2, FC2, FS2, FD2, DS1, FE2, FP1	Errore: VecchiaPassword non valida
TC_1.2_9	FN2, FC2, FS2, FD2, DS1, FE2, FP2, FG7	Errore: NuovaPassword non valida
TC_1.2_10	FN2, FC2, FS2, FD2, DS1, FE2, FP2, FG8, MCP1	Errore: Le password non coincidono
TC_1.2_11	FN2, FC2, FS2, FD2, DS1, FE2, FP2, FG8, MCP2, FCP1	Errore: conferma password non valido
TC_1.2_12	FN2, FC2, FS2, FD2, DS1, FE2, FP2, FG8, MCP2, FCP2	Corretto

### 9.1.3 Login

Autore		LM
Parametro: E-mail		
FORMATO:		
^\\w+([\\ \\. -]?\\w+)*@\\w+([\\ \\. -]?\\w+)*([\\ \\. \\w+){1,100}\$		
Nome Categoria	Scelte per categorie	
Match [ME]	1. Match con email = false [error] 2. Match con email = true [PROPERTY ME OK]	
Parametro: Password		



FORMATO: ^(?=.*[A-Z])(?=.*\d)(?=.*\W){6,30}\$	
Nome Categoria	Scelte per categorie
Match [MP]	1. Match con password= false[error] 2. Match con password= true [PROPERTY MP OK]

Test case ID	Test Frame	Esito
TC_1.3_1	ME1	Errore: Email non corretta o non valida
TC_1.3_2	ME2, MP1	Errore: La password non è corretta o non valida
TC_1.3_3	ME2, MP2	Corretto

#### 9.1.4 Modifica Profilo Organizzatore

Autore		LM
Parametro: Nome		
FORMATO: ^[ a-zA-Z\ u00C0-\ u00ff]{1,50}\$		
Nome Categoria	Scelte per categorie	
Formato [FN]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FN_OK]	
Parametro: Cognome		
FORMATO: ^[ a-zA-Z\ u00C0-\ u00ff]{1,50}\$		
Nome Categoria	Scelte per categorie	
Formato [FC]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FC_OK]	
Parametro: Sesso		
FORMATO:		





“Uomo” “Donna” “Altro”	
Nome Categoria	Scelte per categorie
Formato [FS]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FS_OK]
Parametro: Data di nascita	
FORMATO: [GG/MM/AAAA]	
Formato [FD]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FD_OK]
Inserimento [DS]	1. Precedente alla data odierna [if <b>formatoFDok</b> && dataDiNascita < dataOdierna] [property <b>inserimentoDSok</b> ] 2. Uguale alla data odierna [if <b>formatoFDok</b> && dataDiNascita == dataOdierna] [errore] 3. Successiva alla data odierna [if <b>formatoFDok</b> && dataDiNascita > dataOdierna] [errore]
Parametro: E-mail	
FORMATO: ^\\w+([\\ \\.-]?\\w+)*@\\w+([\\ \\.-]?\\w+)*([\\ \\.-]\\w+){1,100}\$	
Nome Categoria	Scelte per categorie
Formato [FE]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE_OK]
Parametro: Password	
FORMATO: ^(?=.*[A-Z])(?=.*\\d)(?=.*\\W){6,30}\$	
Nome Categoria	Scelte per categorie
Formato [FP]	1. Rispetta il formato = false [error]



	2. Rispetta il formato = true [PROPERTY FP_OK]
<b>Parametro: Biografia</b>	
<b>FORMATO:</b>	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Lunghezza [LB]	1. Lunghezza<1 = false[error] 2. Lunghezza>=1 = true [PROPERTY LE OK]
<b>Parametro: Conferma Password</b>	
<b>FORMATO:</b>	
$\wedge(?=.*[A-Z])(?=.*\d)(?=.*\W).\{6,30\}$$	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Match [MCP]	1. Match con password= false[error] 2. Match con password= true [PROPERTY LCP OK]
Formato [FCP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FCP_OK]
<b>Parametro: IBAN</b>	
<b>FORMATO:</b>	
$\wedge(it IT)[0-9]{2}[A-Za-z][0-9]{10}[0-9A-Za-z]{12}$$	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Formato [FI]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE_OK]

Test case ID	Test Frame	Esito
TC_1.4_1	FN1	Errore: Nome utente non valido
TC_1.4_2	FN2, FC1	Errore: Cognome non valido



TC_1.4_3	FN2, FC2, FS1	Errore: Sesso non valido
TC_1.4_4	FN2, FC2, FS2, FD1	Errore: Data di nascita non valida
TC_1.4_5	FN2, FC2, FS2, FD2, DS2	Errore: Data di nascita uguale alla data odierna
TC_1.4_6	FN2, FC2, FS2, FD2, DS3	Errore: Data di nascita successiva alla data odierna
TC_1.4_7	FN2, FC2, FS2, FD2, DS1, FE1	Errore: Email non valida
TC_1.4_8	FN2, FC2, FS2, FD2, DS1, FE2, FP1	Errore: Password non valida
TC_1.4_9	FN2, FC2, FS2, FD2, DS1, FE2, FP2, LB1	Errore: La biografia non è valida
TC_1.4_10	FN2, FC2, FS2, FD2, DS1, FE2, FP2, LB2, MCP1	Errore: Le password non coincidono
TC_1.4_11	FN2, FC2, FS2, FD2, DS1, FE2, FP2, LB2, MCP2, FCP1	Errore: Il conferma password non è valido
TC_1.4_12	FN2, FC2, FS2, FD2, DS1, FE2, FP2, LB2, MCP2, FCP2, FI1	Errore: IBAN non valido
TC_1.4_13	FN2 FC2 FS2 FD2 DS1 FE2 FP2 LB2 MCP2 FCP2 FI2	Corretto

### 9.1.5 Registrazione Organizzatore

Autore		LM
Parametro: Nome		
FORMATO: ^[ a-zA-Z\ u00C0-\ u00ff]{1,50}\$		
Nome Categoria	Scelte per categorie	
Formato [FN]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FN_OK]	
Parametro: Cognome		
FORMATO: ^[ a-zA-Z\ u00C0-\ u00ff]{1,50}\$		
Nome Categoria	Scelte per categorie	



Formato [FC]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FC_OK]</li> </ol>
<b>Parametro: Sesso</b>	
<b>FORMATO:</b> <b>“Uomo” “Donna” “Altro”</b>	
Formato [FS]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FS_OK]</li> </ol>
<b>Parametro: Data di nascita</b>	
<b>FORMATO:</b> <b>[GG/MM/AAAA]</b>	
Formato [FD]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FD_OK]</li> </ol>
Inserimento [DS]	<ol style="list-style-type: none"> <li>1. Precedente alla data odierna [if <b>formatoFDok</b> &amp;&amp; dataDiNascita &lt; dataOdierna] [property <b>inserimentoDSok</b>]</li> <li>2. Uguale alla data odierna [if <b>formatoFDok</b> &amp;&amp; dataDiNascita == dataOdierna] [errore]</li> <li>3. Successiva alla data odierna [if <b>formatoFDok</b> &amp;&amp; dataDiNascita &gt; dataOdierna] [errore]</li> </ol>
<b>Parametro: E-mail</b>	
<b>FORMATO:</b> <b>^\\w+([\\ \\.]?\\w+)*@\\w+([\\ \\.]?\\w+)*([\\ \\.\\w+){1,100}\$</b>	
Formato [FE]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FE_OK]</li> </ol>
<b>Parametro: Password</b>	
<b>FORMATO:</b> <b>^(?=.*[A-Z])(?=.*\d)(?=.*\W){6,30}\$</b>	
Formato [FP]	<ol style="list-style-type: none"> <li>1. Rispetta il formato = false [error]</li> <li>2. Rispetta il formato = true [PROPERTY FP_OK]</li> </ol>



Parametro: Biografia	
FORMATO:	
Lunghezza [LB]	1. Lunghezza<1 = false[error] 2. Lunghezza>=1 = true [PROPERTY LE OK]
Parametro: Conferma Password	
FORMATO: $\wedge(?.*[A-Z])(?.*\d)(?.*\W).\{6,30\}\$$	
Match [MCP]	1. Match con password= false[error] 2. Match con password= true [PROPERTY LCP OK]
Formato [FCP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FCP_OK]
Parametro: IBAN	
FORMATO: $\wedge(it IT)[0-9]\{2\}[A-Za-z][0-9]\{10\}[0-9A-Za-z]\{12\}\$$	
Formato [FI]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE_OK]

Test case ID	Test Frame	Esito
TC_1.5_1	FN1	Errore: Nome utente non valido
TC_1.5_2	FN2, FC1	Errore: Cognome non valido
TC_1.5_3	FN2, FC2, FS1	Errore: Sesso non valido
TC_1.5_4	FN2, FC2, FS2, FD1	Errore: Data di nascita non valida
TC_1.5_5	FN2, FC2, FS2, FD2, DS2	Errore: Data di nascita uguale alla data odierna



TC_1.5_6	FN2, FC2, FS2, FD2, DS3	Errore: Data di nascita successiva alla data odierna
TC_1.5_7	FN2, FC2, FS2, FD2 , DS1, FE1	Errore: Email non valida
TC_1.5_8	FN2, FC2, FS2, FD2 , DS1, FP1	Errore: Password non valida
TC_1.5_9	FN2, FC2, FS2, FD2 , DS1, FP2, LB1	Errore: La biografia non è valida
TC_1.5_10	FN2, FC2, FS2, FD2 , DS1, FP2, LB2, MCP1	Errore: Le password non coincidono
TC_1.5_11	FN2, FC2, FS2, FD2 , DS1, FP2, LB2, MCP2, FCP1	Errore: Il conferma password non è valido
TC_1.5_12	FN2, FC2, FS2, FD2 , DS1, FP2, LB2, MCP2, FCP2, FI1	Errore: IBAN non valido
TC_1.5_13	FN2 FC2 FS2 FD2 DS1 FE2 FP2 LB2 MCP2 FCP2 FI2	Corretto

#### 9.1.6 Modifica Profilo Amministratore

Autore		ML
Parametro: Nome		
FORMATO: ^[ a-zA-Z\ u00C0- \ u00ff]{1,50}\$		
Nome Categoria	Scelte per categorie	
Formato [FN]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FN_OK]	
Parametro: Cognome		
FORMATO: ^[ a-zA-Z\ u00C0- \ u00ff]{1,50}\$		
Nome Categoria	Scelte per categorie	
Formato [FC]	5. Rispetta il formato = false [error] 6. Rispetta il formato = true [PROPERTY FC_OK]	
Parametro: E-mail		



FORMATO: $\wedge \backslash \backslash w+([\backslash \backslash .-]? \backslash \backslash w+)^* @ \backslash \backslash w+([\backslash \backslash .-]? \backslash \backslash w+)^*(\backslash \backslash . \backslash \backslash w+)\{1,100\} \$$	
Nome Categoria	Scelte per categorie
Formato [FE]	5. Rispetta il formato = false [error] 6. Rispetta il formato = true [PROPERTY FE_OK]
Parametro: Password	
FORMATO: $\wedge (?=.*[A-Z])(?=.*\d)(?=.*\W).\{6,30\} \$$	
Nome Categoria	Scelte per categorie
Formato [FP]	6. Rispetta il formato = false [error] 7. Rispetta il formato = true [PROPERTY FP_OK]
Parametro: ConfermaPassword	
FORMATO: $\wedge (?=.*[A-Z])(?=.*\d)(?=.*\W).\{6,30\} \$$	
Nome Categoria	Scelte per categorie
Match [MCP]	5. Match con password= false[error] 6. Match con password= true [PROPERTY MP OK]
Formato [FCP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FPC OK]

Test case ID	Test Frame	Esito
TC_1.6_1	FN1	Errore: Nome non valido
TC_1.6_2	FN2, FC3	Errore: Cognome non valido
TC_1.6_3	FN2, FC4, FS4, FD4, DS4, FE3	Errore: E-mail non valida
TC_1.6_4	FN2, FC4, FS4, FD4, DS4, FE4, FP3	Errore: Password non valida
TC_1.6_5	FN2, FC4, FS4, FD4, DS4, FE4, FP4, MCP3	Errore: Le password non coincidono



TC_1.6_6	FN2, FC4, FS4, FD4, DS4, FE4, FP4, MCP4, FCP1	Errore: Il conferma password non è valido
TC_1.6_7	FN2, FC4, FS4, FD4, DS4, FE4, FP4, MCP4, FCP2	Corretto

### 9.1.7 Registrazione Scolaresca

Autore		ML
Parametro: Istituto		
FORMATO: ^[\ a-zA-Z\u00C0-\u00ff]{1,100}\$		
Nome Categoria	Scelte per categorie	
Formato [FI]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FS_OK]	
Parametro: E-mail		
FORMATO: ^\ w+([\ \.-]? \ w+)*@\ w+([\ \.-]? \ w+)*([\ \.-] \ w+){1,100}\$		
Nome Categoria	Scelte per categorie	
Formato [FE]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE_OK]	
Parametro: Password		
FORMATO: ^(?=.*[A-Z])(?=.*\ d)(?=.*\ W){6,30}\$		
Nome Categoria	Scelte per categorie	
Formato [FP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FP_OK]	
Parametro: ConfermaPassword		
FORMATO: ^(?=.*[A-Z])(?=.*\ d)(?=.*\ W){6,30}\$		
Nome Categoria	Scelte per categorie	





Match [MCP]	1. Match con password= false [error] 2. Match con password= true [PROPERTY MP OK]
Formato [FCP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FPC OK]

Test case ID	Test Frame	Esito
TC_1.7_1	FI1	Errore: Istituto non valido
TC_1.7_2	FI2, FE1	Errore: Email non valida
TC_1.7_3	FI2, FE2, FP1	Errore: Password non valida
TC_1.7_4	FI2, FE2, FP2, MCP1	Errore: Le password non coincidono
TC_1.7_5	FI2, FE2, FP2, MCP2, FCP1	Errore: Conferma password non valida
TC_1.7_6	FI2, FE2, FP2, MCP2, FCP2	Corretto

#### 9.1.8 Modifica profilo scolaresca

Autore		ML
Parametro: Istituto		
FORMATO: ^[ a-zA-Z\ \u00C0-\u00ff]{1,100}\$		
Nome Categoria	Scelte per categorie	
Formato [FI]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FS_OK]	
Parametro: E-mail		
FORMATO: ^\\ \w+([\\ \.-]?\\ \w+)*@\\ \w+([\\ \.-]?\\ \w+)*([\\ \.-]\\ \w+){1,100}\$		
Nome Categoria	Scelte per categorie	
	1. Rispetta il formato = false [error]	



Formato [FE]	2. Rispetta il formato = true [PROPERTY FE_OK]
<b>Parametro: Password</b>	
<b>FORMATO:</b> $\wedge(?.*[A-Z])(?.*\d)(?.*\W).\{6,30\}\$$	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Formato [FP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FP_OK]
<b>Parametro: ConfermaPassword</b>	
<b>FORMATO:</b> $\wedge(?.*[A-Z])(?.*\d)(?.*\W).\{6,30\}\$$	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
Match [MCP]	1. Match con password= false[error] 2. Match con password= true [PROPERTY MP OK]
Formato [FCP]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FPC OK]

Test case ID	Test Frame	Esito
TC_1.8_1	FI1	Errore: Istituto non valido
TC_1.8_2	FI2, FE1	Errore: Email non valida
TC_1.8_3	FI2, FE2, FP1	Errore: Password non valida
TC_1.8_4	FI2, FE2, FP2, MCP1	Errore: Le password non coincidono
TC_1.8_5	FI2, FE2, FP2, MCP2, FCP1	Errore: Conferma password non valida
TC_1.8_6	FI2, FE2, FP2, MCP2, FCP2	Corretto



## 9.2 Gestione Eventi

### 9.2.1 Richiesta inserimento Evento

Autore		DPA
Parametro: Tipo Evento		
FORMATO: è una di queste stringhe: “mostra”, “teatro”		
Nome Categoria	Scelte per categorie	
FORMATO [FTE]	1. Rispetta il formato = False [error] 2. Rispetta il formato= True [PROPERTY FTE_OK]	
Parametro: Nome Evento		
FORMATO: ^[0-9°A-zÀ-ù ‘-]{1,50}\$		
Nome Categoria	Scelte per categorie	
FORMATO [FNE]	1. Rispetta il formato = False [error] 2. Rispetta il formato= True [PROPERTY FNE_OK]	
Parametro: Descrizione Evento		
FORMATO: ^[0-9°A-zÀ-ù ‘-]{1,320}\$		
Nome Categoria	Scelte per categorie	
FORMATO[FDE]	1. Rispetta il formato = False [error] 2. Rispetta il formato= True [PROPERTY FDE_OK]	
Parametro: Foto Evento		
Nome Categoria	Scelte per categorie	
Grandezza File [GFFE]	1. Grandezza > 15MB = False [error] 2. Grandezza <= 15MB = True [PROPERTY GFFE_OK]	



Dimensione [DFE]	<ol style="list-style-type: none"><li>1. Dimensione &gt; 1024x768 pixel = False [error]</li><li>2. Dimensione &lt;= 1024x768 pixel= True [PROPERTY DFE_OK]</li></ol>
<b>Parametro: Numero Biglietti</b>	
<b>FORMATO: ^([1-9]+[.][0-9]+)\$</b>	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
FORMATO [FNB]	<ol style="list-style-type: none"><li>1. Rispetta il formato= False [error]</li><li>2. Rispetta il formato= True [PROPERTY FNB_OK]</li></ol>
<b>Parametro: Prezzo Biglietti</b>	
<b>FORMATO: ^([0-9]+[.][0-9]*)\$</b>	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
FORMATO [FPB]	<ol style="list-style-type: none"><li>1. Rispetta il formato= False [error]</li><li>2. Rispetta il formato= True [PROPERTY FPB_OK]</li></ol>
<b>Parametro: Data inizio</b>	
<b>FORMATO: gg/mm/aaaa</b>	
<b>Nome Categoria</b>	<b>Scelte per categorie</b>
FORMATO [FDI]	<ol style="list-style-type: none"><li>1. Rispetta il formato= False [error]</li><li>2. Rispetta il formato= True [PROPERTY FDI_OK]</li></ol>
VINCOLO [VDI]	<ol style="list-style-type: none"><li>1. DataInizio&lt;=DataAttuale =False [error]</li><li>2. DataInizio &gt;DataAttuale = True [PROPERTY VDI_OK]</li></ol>
<b>Parametro: Data fine</b>	



FORMATO: gg/mm/aaaa	
Nome Categoria	Scelte per categorie
FORMATO [FDF]	1. Rispetta il formato= False [error] 2. Rispetta il formato= True [PROPERTY FDF_OK]
VINCOLO [VDF]	1. DataFine <= DataInizio =False [error] 2. DataFine >DataInizio = True [PROPERTY VDF_OK]
Parametro: Sede	
FORMATO: ^[0-9°A-zÀ-ù '-]{1,100}\$	
Nome Categoria	Scelte per categorie
FORMATO [FSE]	1. Rispetta il formato= False [error] 2. Rispetta il formato= True [PROPERTY FSE_OK]
Parametro: Indirizzo	
FORMATO: ^[0-9°A-zÀ-ù '-]{2,30}\$	
Nome Categoria	Scelte per categorie
FORMATO [FIND]	1. Rispetta il formato= False [error] 2. Rispetta il formato=True [PROPERTY FIND_OK]

Test case ID	Test Frame	Esito
TC_2.1_1	FTE1	Errore: il Tipo Evento non rispetta il formato
TC_2.1_2	FTE2, FNE1	Errore: il Nome Evento non rispetta



		il formato
TC_2.1_3	FTE2, FNE2, FDE1	Errore: la Descrizione Evento non rispetta il formato
TC_2.1_4	FTE2, FNE2, FDE2, GFFE1	Errore: il file Foto Evento ha una grandezza superiore ai 15MB
TC_2.1_5	FTE2, FNE2, FDE2, GFFE2, DFE1	Errore: il file Foto Evento ha una dimensione maggiore di 1024x768 pixel
TC_2.1_6	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB1	Errore: il Numero Biglietti non rispetta il formato
TC_2.1_7	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB1	Errore: il Prezzo Biglietti non rispetta il formato
TC_2.1_8	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI1	Errore, la Data Inizio non rispetta il formato
TC_2.1_9	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, VDI1	Errore, la Data Inizio inserita è precedente o contemporanea alla Data Attuale
TC_2.1_10	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, VDI2, FDF1	Errore, la Data Fine non rispetta il formato
TC_2.1_11	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, VDI2, FDF2, VDF1	Errore, la Data Fine inserita è precedente o contemporanea alla Data Inizio
TC_2.1_12	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, VDI2, FDF2, VDF2, FSE1	Errore, la Sede non rispetta il formato
TC_2.1_13	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, VDI2, FDF2, VDF2, FSE2, FIND1	Errore, l'indirizzo non rispetta il



		formato
TC_2.1_14	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, VDI2, FDF2, VDF2, FSE2, FIND2	Corretto

### 9.2.2 Richiesta modifica Evento

Autore		DPA
Parametro: Tipo Evento		
FORMATO: è una di queste stringhe: “mostra”, “teatro”		
Nome Categoria	Scelte per categorie	
FORMATO [FTE]	1. Rispetta il formato = False [error] 2. Rispetta il formato= True [PROPERTY FTE_OK]	
Parametro: Nome Evento		
FORMATO: ^[0-9°A-zÀ-ù ‘-]{1,50}\$		
Nome Categoria	Scelte per categorie	
FORMATO [FNE]	1. Rispetta il formato = False [error] 2. Rispetta il formato= True [PROPERTY FNE_OK]	
Parametro: Descrizione Evento		
FORMATO: ^[0-9°A-zÀ-ù ‘-]{1,320}\$		
Nome Categoria	Scelte per categorie	
FORMATO[FDE]	1. Rispetta il formato = False [error] 2. Rispetta il formato= True [PROPERTY FDE_OK]	
Parametro: Foto Evento		
Nome Categoria	Scelte per categorie	



Grandezza File [GFFE]	1. Grandezza > 15MB = False [error] 2. Grandezza <= 15MB = True [PROPERTY GFFE_OK]
Dimensione [DFE]	1. Dimensione > 1024x768 pixel = False [error] 2. Dimensione <= 1024x768 pixel= True [PROPERTY DFE_OK]
Parametro: Numero Biglietti	
FORMATO: ^[0-9]+\$	
Nome Categoria	Scelte per categorie
FORMATO [FNB]	1. Rispetta il formato= False [error] 2. Rispetta il formato= True [PROPERTY FNB_OK]
Parametro: Prezzo Biglietti	
FORMATO: ^([0-9]+[.])*+\$	
Nome Categoria	Scelte per categorie
FORMATO [FPB]	1. Rispetta il formato= False [error] 2. Rispetta il formato= True [PROPERTY FPB_OK]
Parametro: Data inizio	
FORMATO: gg/mm/aaaa	
Nome Categoria	Scelte per categorie
FORMATO [FDI]	1. Rispetta il formato= False [error] 2. Rispetta il formato= True [PROPERTY FDI_OK]
Parametro: Data fine	





FORMATO: gg/mm/aaaa	
Nome Categoria	Scelte per categorie
FORMATO [FDF]	1. Rispetta il formato= False [error] 2. Rispetta il formato= True [PROPERTY FDF_OK]
VINCOLO [VDF]	1. DataFine <= DataInizio =False [error] 2. DataFine >DataInizio = True [PROPERTY VDF_OK]
Parametro: Sede	
^[0-9°A-zÀ-ù '-]{1,100}\$	
Nome Categoria	Scelte per categorie
FORMATO [FSE]	1. Rispetta il formato= False [error] 2. Rispetta il formato= True [PROPERTY FSE_OK]
Parametro: Indirizzo	
FORMATO: ^[0-9°A-zÀ-ù '-]{2,30}\$	
Nome Categoria	Scelte per categorie
FORMATO [FIND]	1. Rispetta il formato= False [error] 2. Rispetta il formato=True [PROPERTY FIND_OK]

Test case ID	Test Frame	Esito
TC_2.2_1	FTE1	Errore: il Tipo Evento non rispetta il formato
TC_2.2_2	FTE2, FNE1	Errore: il Nome Evento non rispetta il formato



TC_2.2_3	FTE2, FNE2, FDE1	Errore: la Descrizione Evento non rispetta il formato
TC_2.2_4	FTE2, FNE2, FDE2, GFFE1	Errore: il file Foto Evento ha una grandezza superiore ai 15MB
TC_2.2_5	FTE2, FNE2, FDE2, GFFE2, DFE1	Errore: il file Foto Evento ha una dimensione maggiore di 1024x768 pixel
TC_2.2_6	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB1	Errore: il Numero Biglietti non rispetta il formato
TC_2.2_7	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB1	Errore: il Prezzo Biglietti non rispetta il formato
TC_2.2_8	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI1	Errore, la Data Inizio non rispetta il formato
TC_2.2_9	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, FDF1	Errore, la Data Fine non rispetta il formato
TC_2.2_10	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, FDF2, VDF1	Errore, la Data Fine inserita è precedente o contemporanea alla Data Inizio
TC_2.2_11	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, FDF2, VDF2, FSE1	Errore, la Sede non rispetta il formato
TC_2.2_12	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, FDF2, VDF2, FSE2, FIND1	Errore, l'indirizzo non rispetta il formato
TC_2.2_13	FTE2, FNE2, FDE2, GFFE2, DFE2, FNB2, FPB2, FDI2, FDF2, VDF2, FSE2, FIND2	Corretto



### 9.2.3 Ricerca Evento

Autore		DPA
Parametro: Testo Ricerca		
FORMATO: ^[0-9°A-zÀ-ù '-]*\*\$		
Nome Categoria	Scelte per categorie	
FORMATO [FTR]	1. Rispetta il formato= False [error] 2. Rispetta il formato=True [PROPERTY FTR_OK]	

Test case ID	Test Frame	Esito
TC_2.3_1	FTR1	Errore: il Testo Ricerca non rispetta il formato
TC_2.3_2	FTR2	Corretto

## 9.3 Gestione Carrello

### 9.3.1 Aggiungi al carrello

Autore		LM
Parametro: Quantità		
FORMATO		
Nome Categoria	Scelte per categorie	
Quantità [QC]	1. Quantità < 1 = false [error] 2. 1<Quantità<numBiglietti = true [PROPERTY QA_OK]	

Test case ID	Test Frame	Esito
--------------	------------	-------



TC_3.1_1	QC1	Errore: Quantità non valida
TC_3.1_2	QC2	Corretto

### 9.3.2 Modifica Carrello

Autore		LM
Parametro: Quantità		
FORMATO:		
Nome Categoria	Scelte per categorie	
Quantità [QC]	1. $Quantità < 1 = \text{false}$ [error] 2. $1 < Quantità < \text{numBiglietti} = \text{true}$ [PROPERTY QC_OK]	

Test case ID	Test Frame	Esito
TC_3.2_1	QC1	Errore: Quantità non valida
TC_3.2_2	QC2	Corretto

## 10. 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.