



# Test Case Plan

Riferimento	
Versione	1.0
Data	4/12/2021
Destinatario	Prof. Gravino
Presentato da	<b>Team AVA</b>
Approvato da	



## Revision History

DATA	Versione	Cambiamenti	Autori
4/12/2021	0.1	Prima Stesura	[tutti]

## Sommario

1 Introduzione	3
2 Relazioni con gli altri documenti	3
3 Panoramica del sistema	4
4 Funzionalità da testare.	4
5 Criteri di successo/fallimento.	4
6 Approccio.	4
6.1 Test di unità.	4
6.2 Sospensione e ripresa	5
7 Strumenti per il testing (hardware/software).	5
8 Test Cases.	5
9 Specifica dei test cases.	5
9.1 Test di Autenticazioni	5
9.1.1 TC1 Registrazione Utente.	5
9.1.2 TC2 Login Utente.	7
9.2 Test di Gestione Eventi	7
9.2.1 TC3 Creazione Evento	8
9.3 Test di Gestione Struttura	9
9.3.1 TC4 Creazione Struttura	9



## Team members

Nome	Ruolo nel progetto	Email
Vincenzopio Amendola	Team Member	v.amendola15@studenti.unisa.it
Alessio Alfieri	Team Member	a.alfieri33@studenti.unisa.it
Alessandro Rusciano	Team Member	a.rusciano1@studenti.unisa.it

### 1. Introduzione

Col presente documento si intende raccogliere un insieme di input e risultati attesi che servono ad effettuare il testing di una componente per individuare gli errori (fault) e del fallimento (failure) a cui conducono. Si vuole fornire una panoramica sulle scelte effettuate riguardanti le componenti specifiche che saranno testate ed una descrizione sul modo in cui saranno testate.

Nella fattispecie:

- Verrà definito un Test Case Plan per l'individuazione delle componenti chiave di cui effettuare il testing
- Verrà poi definito, per ogni Test Case individuato nella fase precedente, una Test Case Specification, che individua quei valori di input che permetteranno di rendere "espressivo" il test in questione, nel senso che si individueranno valori appartenenti



- a diverse classi di equivalenza che rappresentano le varie situazioni a cui il sistema potrebbe trovarsi davanti durante la sua esecuzione in produzione.

## 2. Relazione con gli altri documenti

I documenti a cui faremo riferimento nelle prossime sezioni saranno il Requirements Analysis Document e il Object Design Document. Tali documenti assumono particolare rilievo in quanto contengono la specifica delle diverse operazioni da testare. Inoltre, in alcune sezioni verrà fatto riferimento alla suddivisione in sottosistemi presentata nel documento di System Design

## 3. Panoramica del sistema

Il sistema da noi proposto sarà una web application messa a disposizione per gli utenti (Atleti Professionisti o Dilettanti). Ogni utente può registrarsi, successivamente effettuare login e logout alla piattaforma e utilizzare le funzionalità presenti.

Durante le precedenti fasi dello sviluppo abbiamo individuato quattro sottosistemi: Gestione Utenti, Gestione Schede Tecniche, Gestione Recensione,

## 4. Funzionalità da testare e non

Per alcuni sottosistemi individuati come argomento di test specifichiamo alcune delle funzionalità di cui verificheremo il funzionamento:

- Autenticazione
  - Registrazione
  - Login
- Gestione Eventi
  - Creazione Evento
- Gestione Strutture
  - Creazione Struttura

## 5. Pass/Fail criteria

I casi di test da noi definiti verranno considerati di successo se saranno abbastanza efficaci da rilevare fault nel sistema, qualora dovessero essere presenti. A tale scopo la fase di definizione dei test case verrà curata nei dettagli in modo da far emergere il maggior numero possibile di fault. Se in seguito all'esecuzione dei test dovessero emergere dei fault, questi verranno documentati in un documento dedicato e corretti e verificati tramite test di regressione



## 6. Approccio

La fase di testing si suddivide in tre fasi principali che andremo ad eseguire per verificare il corretto funzionamento del nostro sistema.

### 6.1 Testing di unità

Il test di unità verrà eseguito in isolamento sulle singole componenti sviluppate nel nostro sistema. L'approccio utilizzato durante tale fase è quello black-box, dunque trarremo i test case dalla specifica dei requisiti e in particolare da quella in linguaggio OCL. Per effettuarli verrà utilizzato il framework JUnit

### 6.2 Sospensione e ripresa

Dal momento che il test del sistema richiede una quantità di tempo ragionevolmente bassa per essere eseguita, non è necessario definire vincoli di sospensione e ripresa del testing

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

Non è richiesta la presenza di hardware aggiuntivo per l'esecuzione del testing del sistema. È sufficiente avere un computer su cui è installato.

## 8. Test cases

Per sviluppare i test cases sarà utilizzato il metodo del Category Partition. Questo metodo consiste nell'identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore.

I test cases verranno definiti nel documento di Test Cases Specification (TCS).

## 9. Specifica dei test cases

In questo paragrafo illustreremo le specifiche dei casi di test che adotteremo per il test di sistema. Come detto in precedenza, verrà adottata una strategia category partition.

### 9.1 Test di Autenticazioni

#### 9.1.1 TC1 Registrazione Utente

Per effettuare la registrazione l'utente deve inserire diversi parametri:

- e-mail
- username
- password
- repeatPassword



vincoli che essi

devono rispettare.

<b>Parametro:</b> Email <b>Formato:</b> (^\\w+([.-]?\\w+)@\\w+([.-]?\\w+)(\\.\\w{2,3})+\$)	
Categorie	Scelte
<b>LE (Lunghezza Email)</b>	1: LE >= 8 && LE <= 50 [property LE_OK] 2: LE > 50    LE < 8 [error]
<b>FE (Formato Email)</b>	1: Rispetta il formato [if LE_OK] [property FE_OK] 2: Non rispetta il formato [error]
<b>EE (Email Esistente)</b>	1: L'email non esiste nel sistema [if FE_OK] [property EE_OK] 2: L'email esiste nel sistema [if FE_OK] [error]

<b>Parametro:</b> Username <b>Formato:</b> (\\w{4,50})	
Categorie	Scelte
<b>LU (Lunghezza Username)</b>	1: Rispetta il formato [if EE_OK] [property FU_OK] 2: Non rispetta il formato [if EE_OK] [error]

<b>Parametro:</b> Password <b>Formato:</b> (^((?=.[A-Z])(?=.[a-z])(?=.*[0-9])).{8,50}\$)	
Categorie	Scelte
<b>FP (Formato Password)</b>	1 Rispetta il formato [if FU_OK] [property FP_OK] 2 Non rispetta il formato [if FU_OK] [error]

<b>Parametro:</b> repeatPassword <b>Formato:</b> -	
Categorie	Scelte



<b>CP</b> <b>(Corrispondenza Password)</b>	1 Corrisponde a “Password” [if <b>FP_OK</b> ] [property <b>CP_OK</b> ] 2 Non corrisponde a “Password” [if <b>FP_OK</b> ] [error]
---	---

Codice	Combinazione	Esito
TC1_01	LE2	errore
TC1_02	LE1, FE2	errore
TC1_03	LE1,FE1,EE2	errore
TC1_04	LE1,FE1,EE1,FU2	errore
TC1_05	LE1, FE1, EE1, FU1, FP2	errore
TC1_06	LE1, FE1, EE1, FU1, FP1, CP2	errore
TC1_07	LE1, FE1, EE1, FU1, FP1, CP1	SUCCESSO

### 9.1.1 TC2 Login Utente

Per effettuare il Log In l’utente deve inserire diversi parametri:

- email
- password

Per ognuno dei parametri sopra elencati andremo a caratterizzare i vincoli che essi devono rispettare

Parametro: Email	
Categorie	Scelte
EV (Email Valida)	1 L’e-mail esiste nel sistema [property EV_OK] 2 L’e-mail non esiste nel sistema [error]

Parametro: Password	
Categorie	Scelte



<b>PC</b> <b>(Password Corretta)</b>	1 La Password corrisponde all'e-mail [if EV_OK] [property PC_OK] 2 La Password non corrisponde [if EV_OK] [error]
---	---

Codice	Combinazione	Esito
TC2_01	EV2	errore
TC2_02	EV1,PC2	errore
TC3_03	EV1,PC1	SUCCESSO

## 9.2 Test di Gestione Eventi

### 9.2.1 TC3 Creazione Evento

Per effettuare la creazione di un Evento l'utente deve inserire i seguenti parametri:

- Data
- Orario
- Nome
- Numero Partecipanti

Per ognuno dei parametri sopra elencati andremo a caratterizzare i vincoli che essi devono rispettare.

<b>Parametro:</b> Data <b>Formato:</b> [DD-MM-YYYY]	
Categorie	Scelte
<b>FD (Formato Data)</b>	1: Rispetta il formato [property <b>FD_OK</b> ] 2: Non rispetta il formato [error]
<b>ID (Inserimento Data)</b>	1: Data inserita rispetta il formato ed è successiva alla data corrente [if <b>FD_OK</b> && Data>DataCorrente][property <b>ID_OK</b> ] 2: Data inserita rispetta il formato ma è precedente alla data corrente [if <b>FD_OK</b> && Data<=DataCorrente][errore]

<b>Parametro:</b> Orario <b>Formato:</b> ^([ 01]?[0-9] 2[0-3])(: ([0-5][0-9]))?\$	
Categorie	Scelte





<b>FO</b> <b>(Formato Orario)</b>	<b>1:</b> Rispetta il formato [property <b>FO_OK</b> ] <b>2:</b> Non rispetta il formato [error]
--------------------------------------	---

<b>Parametro:</b> Nome <b>Formato:</b> (\w{4,50})	
Categorie	Scelte
<b>FN (Formato Nome)</b>	<b>1:</b> Rispetta il formato [property <b>FN_OK</b> ] <b>2:</b> Non rispetta il formato [error]

<b>Parametro:</b> Numero Partecipanti <b>Formato:</b> [2-10]	
Categorie	Scelte
<b>FP (Formato Partecipanti)</b>	<b>1:</b> Rispetta il formato [property <b>FP_OK</b> ] <b>2:</b> Non rispetta il formato [error]

Codice	Combinazione	Esito
<b>TC3_01</b>	FD2	errore
<b>TC3_02</b>	FD1,ID2	errore
<b>TC3_03</b>	FD1,ID1,FO2	errore
<b>TC3_04</b>	FD1,ID1,FO1,FN2	errore
<b>TC3_05</b>	FD1,ID1,FO1,FN1,FP2	errore
<b>TC3_06</b>	FD1,ID1,FO1,FN1,FP1	SUCCESSO

## 9.3 Test di Gestione Strutture

### 9.3.1 TC4 Creazione Struttura



Laurea Triennale in informatica - Università di Salerno  
Corso di Ingegneria del Software – Prof. Gravino  
Per effettuare la creazione di una Struttura l'admin deve inserire i seguenti

parametri:

- Nome
- Indirizzo
- Numero Telefono
- Numero Spogliatoi
- Parcheggio
- Categoria
- Capienza

Per ognuno dei parametri sopra elencati andremo a caratterizzare i vincoli che essi devono rispettare.

<b>Parametro:</b> Nome <b>Formato:</b> (\w{4,50})	
Categorie	Scelte
<b>FN (Formato Nome)</b>	1: Rispetta il formato [property <b>FN_OK</b> ] 2: Non rispetta il formato [error]

<b>Parametro:</b> Indirizzo <b>Formato:</b> @"^[A-Za-z]+ ?[0-9]+\$"	
Categorie	Scelte
<b>FI (Formato Indirizzo)</b>	1: Rispetta il formato [property <b>FI_OK</b> ] 2: Non rispetta il formato [error]

<b>Parametro:</b> Numero di Telefono <b>Formato:</b> ^\+[1-9]{1}[0-9]{3,14}\$	
Categorie	Scelte
<b>FT (Formato Telefono)</b>	1: Rispetta il formato [property <b>FT_OK</b> ] 2: Non rispetta il formato [error]



<b>Parametro:</b> Numero Spogliatoi <b>Formato:</b> (\w{2,20})	
Categorie	Scelte
<b>FS (Formato Spogliatoi)</b>	1: Rispetta il formato [property <b>FS_OK</b> ] 2: Non rispetta il formato [error]

<b>Parametro:</b> Parcheggio <b>Formato:</b> {si no}	
Categorie	Scelte
<b>FP (Formato Parcheggio)</b>	1: Rispetta il formato [property <b>FP_OK</b> ] 2: Non rispetta il formato [error]

<b>Parametro:</b> Categoria <b>Formato:</b> { Calcioa5   Tennis   Basket   Volley   Padel }	
Categorie	Scelte
<b>FC (Formato Categoria)</b>	1: Rispetta il formato [property <b>FC_OK</b> ] 2: Non rispetta il formato [error]

<b>Parametro:</b> Capienza <b>Formato:</b> (\w{10,100})	
Categorie	Scelte
<b>FCap(Formato Capienza)</b>	1: Rispetta il formato [property <b>FCap_OK</b> ] 2: Non rispetta il formato [error]

Codice	Combinazione	Esito
<b>TC4_01</b>	FN2	errore
<b>TC4_02</b>	FN1,FI2	errore



<b>TC4_03</b>	FN1,FI1,FT2	errore
<b>TC4_04</b>	FN1,FI1,FT1,FS2	errore
<b>TC4_05</b>	FN1,FI1,FT1,FS1,FP2	errore
<b>TC4_06</b>	FN1,FI1,FT1,FS1,FP1,FC2	errore
<b>TC4_07</b>	FN1,FI1,FT1,FS1,FP1,FC1,FCap2	errore
<b>TC4_08</b>	FN1,FI1,FT1,FS1,FP1,FC1,FCap1	SUCCESSO