



UNIVERSITÀ DEGLI STUDI  
DI SALERNO

**Ingegneria del Software**

A.A. 2019/2020



**Test Plan**

**Versione 1.1**

**Top Manager:**

Professore
Andrea De Lucia

**Partecipanti:**

Nome	Matricola
Vittorio Ventura	05121 5766

# Test Plan Document

## 1. Introduzione

Lo scopo di questo documento è definire quelli che sono gli aspetti gestionali del test tenendo traccia di quelle che sono le risorse e il programma dell'attività di test. L'obiettivo sarà quindi quello di cercare le differenze tra il comportamento previsto specificato all'interno dei modelli di sistema e il comportamento osservato dal sistema implementato. Sulla base dei risultati ottenuti verranno quindi individuati eventuali fault, che verranno corretti in un secondo momento. Verranno testate quindi le funzionalità descritte in precedenza all'interno dei seguenti documenti:

- RAD
- ODD

## 2. Relazione con gli altri documenti

Questo documento è chiaramente in stretta relazione con gli altri documenti: Nel RAD sono specificati i requisiti funzionali e non funzionali che ci aiuteranno nell'esecuzione dei test; Nell'SDD il sistema in uso è stato decomposto seguendo il modello MVC, quindi ottenendo 3 livelli, View, Control e Model. Cercheremo dunque di mantenere il testing quanto più fedele a quest'architettura. Infine, Il test d'integrazione farà riferimento alle interfacce delle classi definite nell'ODD.

## 3. Panoramica del sistema

Verrà testato principalmente ogni Management con una particolare enfasi sulle operazioni di inserimento e aggiornamento. Verranno effettuati inizialmente test sui singoli sottosistemi, in maniera tale da ridurre la complessità dell'attività di test, semplificare l'individuazione e la correzione dei guasti e garantire lo svolgimento in parallelo delle diverse attività di test. Si passerà successivamente all'Integration testing in modo da individuare eventuali errori che durante la prima fase di test descritta in precedenza non sono stati individuati mediante le opportune strategie. Il test verrà quindi svolto nella maniera descritta concentrandosi sulle operazioni descritte nella prima parte del punto 3.

## 4. Funzionalità da testare

- Login
- Registrazione
- Inserimento annuncio
- Inserimento allegato
- Aggiungi Università
- Aggiungi Categoria
- Elimina Annuncio
- Elimina Utente
- Modifica Ruolo

## 5. Criteri di Successo/Fallimento

I dati di input verranno raggruppati in insiemi con caratteristiche comuni, per i quale testeremo solo un elemento rappresentativo e non tutti i possibili input. Un input supererà un test se l'output è tra quelli attesi, cioè quello che è stato specificato in precedenza dal membro che si occuperà del testing su quel determinato test case.

## 6. Approccio

Abbiamo deciso di dividere l'attività di testing in tre categoria: test di unità, test d'integrazione e test di sistema

### 6.1 Testing d'unità

Durante questa fase ci concentriamo sul testing degli elementi che costituiscono il sistema software utilizzando test driver e stub ove necessario per verificare la componente. La strategia utilizzata per il testing si baserà esclusivamente sulla tecnica Black-box, che si focalizza sul comportamento Input/Output, ignorando la struttura interna della componente. Utilizzando il test d'unità possiamo semplificare l'individuazione e la correzione di fault consentendoci di concentrarci su piccole parti del sistema, ridurre la complessità dell'attività di testing e infine parallelizzare l'attività di test delle componenti. Per effettuare il testing di unità di ogni singola componente del sistema UniAds verrà utilizzata la tecnica del "Black-Box Testing" attraverso il framework JUnit; Per minimizzare il numero di test case e migliorare la qualità di essi, gli input saranno partizionati in classi di equivalenza. Tali classi serviranno a definire le categorie da usare, come testing funzionale, il Category Partition; quindi per ogni classe di equivalenza verranno individuati dei vincoli a cui saranno associate specifiche scelte.

Gli errori che verranno individuati durante questa fase iniziale di testing verranno ovviamente corretti nel più breve tempo possibile e verranno riportati i vari fallimenti nell'apposito documento "Test incident report".

## 6.2 Testing d'integrazione

In questa fase passiamo all'integrazione dei componenti precedentemente testati con l'obiettivo di identificare quelli che sono fault non identificati fino a questa fase. Siccome l'ordine in cui i componenti vengono integrati può influenzare lo sforzo totale richiesto dal test d'integrazione si è scelto di focalizzare la parte principale sull'integrazione dei seguenti sottosistemi:

ArticoloManagement, AllegatoManagement e NotificaManagement. Si è scelto di utilizzare una strategia di test di integrazione orizzontale in particolare una strategia Bottom-up quindi testare prima i componenti del layer inferiore (fatto già nella fase precedente di testing) e poi integrarli con quelli del layer superiore. Si è scelto di utilizzare questa strategia in modo da poter individuare eventuali errori di interfaccia

## 6.3 Testing di sistema

Mediante questo test cerchiamo di garantire che l'intero sistema sia conforme ai requisiti funzionali e non funzionali. Verranno eseguite le diverse funzionalità del sistema passando di volta in volta alla funzionalità successiva fino al termine di esse. Per questa fase utilizzeremo Selenium.

## 7. Sospensione e riassunzione

La fase di testing verrà sospesa quando si raggiungerà un compromesso tra la qualità del prodotto e costi dell'attività di testing. Il testing verrà quindi portato avanti quanto più possibile nel tempo senza però rischiare di ritardare la consegna finale del progetto e rispettando i budget del progetto. Inoltre, in seguito alle modifiche o correzioni delle componenti che introdurranno errori o fallimenti, i test case verranno sottoposti nuovamente al sistema facendo così in modo di risolvere definitivamente il problema.

## 8. Materiale di testing

Il testing verrà effettuato mediante l'utilizzo di un pc ,dotato degli opportuni software, e non sarà neanche necessaria una connessione ad internet visto che il pc sarà dotato di una copia locale e fornito di un'apposita API.

## 9. Casi di Testing

### TC\_0.1 Login

<b>Parametro:</b> email	Formato: ^[a-zA-Z0-9_+~]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-~]+\$
<b>Formato[FE]</b>	<ol style="list-style-type: none"><li>1 Rispecchia il formato [FEOK]</li><li>2 Non rispecchia il formato [error]</li></ol>
<b>Esiste [EE]</b>	<ol style="list-style-type: none"><li>1 L'email non esiste nel DB [error] [if FEOK]</li><li>2 L'email è registrata nel DB [EEOk] [if FEOK]</li></ol>

<b>Parametro:</b> password	Formato: ^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[#\$^+=!*()@%&]).{8,30}\$
<b>Formato [FP]</b>	<ol style="list-style-type: none"><li>1. Rispecchia il formato [FEOK]</li><li>2. Non rispecchia il formato [error]</li></ol>
<b>Match[MP]</b>	<ol style="list-style-type: none"><li>1. match con password utente = false [error]</li><li>2. match con password utente = true [property MP_OK]</li></ol>

<b>Parametro:</b> Ruolo	Selezione : Utente , Gestore Piattaforma
<b>MatchRuolo[MR]</b>	<ol style="list-style-type: none"><li>1. Match con ruolo utente = false [error]</li><li>2. Match con ruolo utente = true [property MR_OK]</li></ol>

**Test case**

Codice	Combinazione	Esito
TC_0.1_01	FE2	ERROR
TC_0.1_02	FE1, EE1	ERROR
TC_0.1_03	FE1, EE2, FP2	ERROR
TC_0.1_04	FE1, EE2, FP1, MP1	ERROR
TC_0.1_05	FE1, EE2, FP1, MP2,MR1	ERROR
TC_0.1_06	FE1, EE2, FP1, MP2,MR2	SUCCESS

## TC\_1.1 Registrazione

<b>Parametro:</b> nome	Formato: ^[A-Z][a-z][^#!@&<>\\[\]\"~;\$^%{}?]{0-9}]{0,30}\$
<b>Formato [FN]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [FNOK]</li> <li>2. Non rispecchia il formato [error]</li> </ol>

<b>Parametro:</b> cognome	Formato: ^[A-Z][a-z][^#!@&<>\\[\]\"~;\$^%{}?]{0-9}]{0,30}\$
<b>Formato [FC]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [FCOK]</li> <li>2. Non rispecchia il formato [error]</li> </ol>

<b>Parametro:</b> email	Formato: ^[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+.\$
<b>Formato[FE]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [FEOK]</li> <li>2. Non rispecchia il formato [error]</li> </ol>
<b>Esiste [EE]</b>	<ol style="list-style-type: none"> <li>3. L'email non esiste nel DB [EEOK] [if FEOK]</li> <li>4. L'email è già registrata nel DB [error][if FEOK]</li> </ol>

<b>Parametro:</b> password	Formato: ^(?=[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[#\$^+=!*()@%&]).{8,30}\$
<b>Formato[FP]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [FP]</li> <li>2. Non rispecchia il formato [error]</li> </ol>

<b>Parametro:</b> Conferma Password	Formato: ^(?=[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[#\$^+=!*()@%&]).{8,30}\$
<b>Formato[FCP]</b>	<ol style="list-style-type: none"> <li>1. Non rispecchia il formato [error]</li> <li>2. Non corrisponde con FP [error]</li> <li>3. Corrisponde con FP [FCPOK]</li> </ol>



## Test case

Codice	Combinazione	Esito
TC_1.1_01	FE2	ERROR
TC_1.1_02	FE1, EE2	ERROR
TC_1.1_03	FE1, EE1, FN2	ERROR
TC_1.1_04	FE1, EE1, FN1, FC2	ERROR
TC_1.1_05	FE1, EE1, FN1, FC1, FU2	ERROR
TC_1.1_06	FE1, EE1, FN1, FC1, FU1, EU2	ERROR
TC_1.1_07	FE1, EE1, FN1, FC1, FU1, EU1, FP2	ERROR
TC_1.1_08	FE1, EE1, FN1, FC1, FU1, EU1, FP1	ERROR
TC_1.1_09	FE1, EE1, FN1, FC1, FU1, EU1, FP1, FCP1	ERROR
TC_1.1_10	FE1, EE1, FN1, FC1, FU1, EU1, FP2	ERROR
TC_11_11	FE1, EE1, FN1, FC1, FU1, EU1, FP1, FP3	SUCCESS

## TC\_2.1 Aggiungi Università

<b>Parametro:</b> Sigla Università	Formato: ^[A-Z][^#!@&<>\\[\]\"~;,\$^%{}]?{0-9}]{0,100}\$
<b>Formato[FSU]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [FSUOK]</li> <li>2. Non rispecchia il formato [error]</li> </ol>
<b>Esiste [SUE]</b>	<ol style="list-style-type: none"> <li>1. L'università non esiste nel DB [SUEOK] [if FSUOK]</li> <li>2. L'università è già registrata nel DB [error][if FSUOK]</li> </ol>

<b>Parametro:</b> Località	Formato: ^[A-Z][^#!@&<>\\[\]\"~;,\$^%{}]?{0-9}]{0,30}\$
<b>Formato[FL]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [FLOK]</li> <li>2. Non rispecchia il formato [error]</li> </ol>
<b>Esiste [LE]</b>	<ol style="list-style-type: none"> <li>1. La regione esiste nel DB [LEOK] [if FLOK]</li> <li>2. La regione non è registrata nel DB [error][if FLOK]</li> </ol>

Codice	Combinazione	Esito
TC_2.1_01	FSU2	ERROR
TC_2.2_02	FSU1, SUE2	ERROR
TC_2.3_03	FSU1, SUE1, FL2	ERROR
TC_2.4_04	FSU1, SUE1, FL1, LE2	ERROR
TC_2.5_05	FSU1, SUE1, FL1, LE1	SUCCESS

### TC\_2.2 Elimina Categoria

<b>Parametro:</b> Categoria	Formato: ^[A-Z][^#!@&<>\\\"~;\$^%{}?]{0-9}}{0,30}\$
<b>Formato[FC]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [FCOK]</li> <li>2. Non rispecchia il formato [error]</li> </ol>
<b>Esiste [CE]</b>	<ol style="list-style-type: none"> <li>1. La categoria già esiste nel DB [error] [if FLOK]</li> <li>2. La categoria non è registrata nel DB [CEOK][if FLOK]</li> </ol>

### Test case

Codice	Combinazione	Esito
TC_2.2_01	FC2	ERROR
TC.2.2_02	FC1, CE1	ERROR
TC_2.2_03	FC1, CE2	SUCCESS

### TC\_2.3 Elimina Annuncio

<b>Parametro:</b> email	Formato: ^[a-zA-Z0-9_+~]+@[a-zA-Z0-9~]+.[a-zA-Z0-9~]+.\$
<b>Formato[FE]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [FEOK]</li> <li>2. Non rispecchia il formato [error]</li> </ol>
<b>Esiste [EE]</b>	<ol style="list-style-type: none"> <li>1. L'email non esiste nel DB [error] [if FEOK]</li> <li>2. L'email è registrata nel DB [EEOK] [if FEOK]</li> </ol>

<b>Parametro:</b> titolo	Formato: [A-Z][a-zA-Z0-9][^#@&<>\\\"~;\$^%{}?]{4,50}\$
<b>Formato [FT]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [FTOK]</li> <li>2. Non rispecchia il formato [error]</li> </ol>
<b>Esiste [ET]</b>	<ol style="list-style-type: none"> <li>1. Il titolo è presente nel database [ETOK] [if FTOK]</li> <li>2. Il titolo non è presente nel database [error] [if FTOK]</li> </ol>
<b>Match [MT]</b>	<ol style="list-style-type: none"> <li>1. Il titolo non corrisponde con email dell'utente [error] [IF ETOK]</li> <li>2. Il titolo corrisponde con email dell'utente [MTOK] [IF ETOK]</li> </ol>

**Test case**

Codice	Combinazione	Esito
TC_2.3_01	FE2	ERROR
TC_2.3_02	FE1, EE1	ERROR
TC_2.3_03	FE1, EE2, FT2	ERROR
TC_2.3_04	FE1, EE2, FT1, ET2	ERROR
TC_2.3_05	FT1, EE2, FT1, ET1, MT1	ERROR
TC_2.3_07	FT1, EE2, FT1, ET1, MT2	SUCCESS

#### TC\_2.4 Elimina Utente

<b>Parametro:</b> email	Formato: ^[a-zA-Z0-9_+~]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+\.\$
<b>Formato[FE]</b>	<ol style="list-style-type: none"><li>1. Rispecchia il formato [FEOK]</li><li>2. Non rispecchia il formato [error]</li></ol>
<b>Esiste [EE]</b>	<ol style="list-style-type: none"><li>1. L'email esiste nel DB [EEOK] [if FEOK]</li><li>2. L'email non esiste nel DB [error][if FEOK]</li></ol>

#### Test case

Codice	Combinazione	Esito
TC_2.4_01	FE2	ERROR
TC_2.4_02	FE1, EE2	ERROR
TC_2.4_03	FE1, EE1	SUCCESS

#### TC\_2.5 Modifica Ruolo

<b>Parametro:</b> email	Formato: ^[a-zA-Z0-9_+~]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+\.\$
<b>Formato[FE]</b>	<ol style="list-style-type: none"><li>1. Rispecchia il formato [FEOK]</li><li>2. Non rispecchia il formato [error]</li></ol>
<b>Esiste [EE]</b>	<ol style="list-style-type: none"><li>1. L'email non esiste nel DB [error] [if FEOK]</li><li>2. L'email è registrata nel DB [EEOK] [if FEOK]</li></ol>

<b>Parametro:</b> Ruolo[SR]	Selezione : Utente , Gestore Piattaforma
-----------------------------	--

#### Test case

Codice	Combinazione	Esito
TC_2.5_01	FE2	ERROR
TC_2.5_02	FE1, EE1	ERROR
TC_2.5_03	FE1, EE2,SR1	SUCCESS

### TC\_3.1 Crea Annuncio

<b>Parametro:</b> titolo	Formato: [A-Z][a-zA-Z0-9][^#@<>\"~;\$^%{}?]{4,50}\$
<b>Formato[FT]</b>	3. Rispecchia il formato [FTOK] 4. Non rispecchia il formato [error]
<b>Esiste [ET]</b>	3. Il titolo è già presente nel database [if FTOK][error] 4. Il titolo non è presente nel database [if FTOK][ETOK]

<b>Parametro:</b> immagini	Formato: ([a-z_\-s0-9\.]++)\.(jpeg/ png)
<b>Formato[FIMG]</b>	1. Rispecchia il formato [FIMG] 2. Non rispecchia il formato [error]

<b>Parametro:</b> Università	
<b>Selezione [FU]</b>	1. Un valore selezionato [FUOK] 2. Nessun valore selezionato [error]

<b>Parametro:</b> categoria	
<b>Selezione [FC]</b>	1. Un valore selezionato [FCOK] 2. Nessun valore selezionato [error]

## Test case

Codice	Combinazione	Esito
TC_3.1_01	FT2	ERROR
TC_3.1_02	FT1, ET1	ERROR
TC_3.1_03	FT1, ET2, FIMG2	ERROR
TC_3.1_04	FT1, ET2, FIMG1, FU2	ERROR
TC_3.1_05	FT1, ET2, FIMG1, FU1, FC2	ERROR
TC_3.1_07	FT1, ET2, FDOC1, FC1, EC1, FC1	SUCCESS

## 10. Testing schedule

L'attività di testing richiede agli sviluppatori di trovare fault nei componenti del sistema ed è meglio che il test venga eseguito da uno sviluppatore che non è stato coinvolto nello sviluppo della componente o da un utente esperto del sistema. Tuttavia in questo caso il test è stato svolto dagli stessi autori del sistema. Testando inizialmente le singole componenti abbiamo cercato di individuare quante più fault possibili nel sistema e quindi successivamente correggerli. Dopo aver corretto gli eventuali fault abbiamo nuovamente testato il sistema per verificare che la modifica non abbia introdotto nuovi fault. Infine siamo passati ai test d'integrazione ripetendo ovviamente lo stesso procedimento precedentemente descritto. L'attività di testing è fondamentale per lo sviluppo corretto di un software, in quanto una tale mancanza potrebbe causare il fallimento dell'intero sistema. Per tale motivo, è fondamentale schedulare il testing.