

PLAN

# TESTING PLAN

**Ingegneria del Software**



**Anno Accademico 2017/2018**

**GRUPPO:**

Gaeta Francesco	05121 03566
Manzo Alessandro	05121 03628
Moliterni Marco	05121 03906
Valente Carmine	05121 03614

## Sommario:

1. Introduzione	3
2. Documenti correlati	3
2.1. Relazioni con il documento di analisi dei requisiti (RAD)	3
2.2. Relazioni con il System Design Document (SDD)	3
2.3. Relazioni con l'Object Design Document (ODD)	3
3. Panoramica del sistema	4
4. Funzionalità da testare e da non testare	4
4.1. Criteri Pass/Failed	4
4.2. Approccio	4
4.2.1. Testing di Unità	5
4.2.2. Testing d'integrazione	5
4.2.3. Testing di sistema	5
4.3. Sospensione e ripresa	6
4.3.1. Criteri di sospensione	6
4.3.2. Criteri di ripresa	6
4.3.3. Materiale per il Testing	6
5. Test Cases	6
5.1. Registrazione	7
5.2. Login	8

## 1. Introduzione

L'obiettivo del seguente documento consiste nel pianificare l'attività di testing del sistema TeachMe in modo da osservare se vi sono differenze fra il comportamento osservato e il comportamento aspettato. Lo scopo di questa fase è quello di prevenire la presenza di errori nel sistema durante l'utilizzo dell'utente finale.

Le attività di test sono state pianificate per le seguenti gestioni:

- Gestione Registrazione
- Gestione Autenticazione

Va comunque specificato, che verranno testate solamente le funzionalità implementate e descritte

nell'ODD. Per il testing, verranno utilizzati i test case, ovvero un insieme di input e di risultati attesi

che servono appunto a testare una componente per scoprirne gli errori e i fallimenti.

Verranno inoltre pianificate anche le responsabilità del team nei relativi test e lo scheduling di quest'ultimi.

## 2. Documenti correlati

Il test plan è strettamente correlato alla documentazione precedentemente prodotta, in quanto buona

parte del sistema era stata precedentemente progettata e anche implementata, e in questa fase risulta

quindi più che mai necessario utilizzare il risultato delle fasi precedenti e dell'implementazione

effettuata, in modo da rilevare correttamente le possibili differenze fra ciò che si era progettato e ciò

che il sistema realizza davvero.

### 2.1. Relazioni con il documento di analisi dei requisiti (RAD)

La relazione del test plan con il documento di analisi dei requisiti (RAD) è molto stretta in quanto in

quest'ultimo vengono descritti requisiti funzionali e non funzionali, e si ottengono gli use case sui

quali dovranno poi basarsi i test case.

### 2.2. Relazioni con il System Design Document (SDD)

La relazione del test plan con il system design document (SDD) incide particolarmente sull'organizzazione dei test in quanto, nell'ultimo documento, è stata utilizzata un'architettura a three

tiers, ovvero a tre strati, ed in particolare abbiamo: presentation layer, application layer e storage

layer. Il test delle varie componenti deve rimanere quanto più fedele possibile a questa suddivisione.

### 2.3. Relazioni con l'Object Design Document (ODD)

Il test d'integrazione farà riferimento alle classi delineate nell'ODD in modo da attenersi il più possibile a quest'ultime.

### 3. Panoramica del sistema

Il sistema TeachMe si basa su un'architettura a tre strati: presentation layer, application layer e storage layer. L'utente interagirà con il livello più alto, il presentation layer, che a sua volta mostrerà i contenuti offerti e ricavati dall'application layer che contiene la logica di business del sistema e si interfacerà con il livello storage per eseguire operazioni con i dati persistenti. Per facilitarne la gestione, il sistema è stato diviso in sottosistemi. In particolare, abbiamo:

- Gestione Autenticazione
- Gestione Utente
- Gestione Ricerca
- Gestione Preferiti
- Gestione Admin
- Gestione Connessione

### 4. Funzionalità da testare e da non testare

Di seguito saranno elencate per ogni gestione quali sono le funzionalità che saranno testate

Gestione Registrazione

- Registrazione

Gestione Autenticazione

- Login

#### 4.1. Criteri Pass/Failed

I dati di input del test saranno suddivisi in classi di equivalenza, ovvero verranno raggruppati in insiemi dalle caratteristiche comuni, per i quali sarà sufficiente testare un solo elemento rappresentativo.

Un input avrà superato un test se l'output risultante sarà quello atteso, cioè quello che è stato

specificato dal membro del team che si occuperà del testing su tale test case, il responsabile del

testing conosce quale dovrebbe essere l'output corretto.

#### 4.2. Approccio

## PLAN

Le tecniche di testing adottate riguarderanno inizialmente il testing di unità dei singoli componenti, in modo da testare nello specifico la correttezza di ciascuna unità. Seguirà il testing d'integrazione, che focalizzerà l'attenzione principalmente sul test delle interfacce delle suddette unità. Infine, verrà eseguito il testing di sistema, che vedrà come oggetto di testing l'intero sistema assemblato nei suoi componenti. Quest'ultimo servirà soprattutto a verificare che il sistema soddisfi le richieste del committente.

### **4.2.1. Testing di Unità**

Durante questa fase, verranno ricercate le condizioni di fallimento isolando i componenti ed usando test driver e stub, cioè implementazioni parziali di componenti che dipendono o da cui dipendono le componenti da testare. 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. Al fine di minimizzare il numero di test cases, i possibili input verranno partizionati in classi di equivalenza e per ogni classe verrà selezionato un test case. Gli stati erronei scovati in questa, come in qualsiasi altra fase di testing, che comporteranno un fallimento del sistema, dovranno essere tempestivamente corretti per riprendere il testing al più presto. I fallimenti identificati saranno documentati nel test incident report.

### **4.2.2. Testing d'integrazione**

In questa fase si procederà all'integrazione delle componenti di una funzionalità che verranno estate nel complesso attraverso una strategia Bottom-Up. Si passerà, poi, alla funzionalità successiva fino ad esaurire le funzionalità implementate. Quest'approccio mira principalmente a ridurre le dipendenze tra funzionalità differenti e a facilitare la ricerca di errori nelle interfacce di comunicazione tra sottosistemi.

### **4.2.3. Testing di sistema**

Lo scopo di questa fase di testing è quello di dimostrare che il sistema soddisfi effettivamente i requisiti richiesti e sia, quindi, pronto all'uso. Come per il testing di unità, si cercherà di testare le funzionalità più importanti per l'utente e quelle che hanno una maggiore probabilità di fallimento. Si noti che, come per il testing di unità, si procederà attraverso tecnica Black-Box

### **4.3. Sospensione e ripresa**

#### **4.3.1. Criteri di sospensione**

La fase di testing del sistema verrà sospesa quando si raggiungerà un compromesso tra 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.

#### **4.3.2. Criteri di ripresa**

In seguito ad ogni modifica o correzione delle componenti che genereranno errori o fallimenti, i test case verranno sottoposti nuovamente al sistema assicurandosi così di aver risolto effettivamente il problema.

#### **4.3.3. Materiale per il Testing**

L'hardware necessario per l'attività di test è un pc con connessione ad Internet (per alcune funzionalità), e con database e sito web implementati in locale.

## **5. Test Cases**

## 5.1. Registrazione

### Category Partition:

<b>Parametro :</b> Mail <b>Formato :</b> <code>^[a-zA-Z0-9.!#\$%&amp;'*/+=?^_`{ }~-]+@studenti.unisa.it\$</code>	
<b>Formato [FM]</b>	1. Rispecchia il formato 2. Contiene caratteri speciali [error]
<b>Parametro :</b> Nome <b>Formato :</b> <code>^[A-Za-z'] {1,20}\$</code>	
<b>Formato [FN]</b>	1. Rispecchia il formato 2. Lunghezza < 1 o > 20 [error] 3. Contiene caratteri speciali [error]
<b>Parametro :</b> Cognome <b>Formato :</b> <code>^[A-Za-z'] {1,20}\$</code>	
<b>Formato [FCO]</b>	1. Rispecchia il formato 2. Lunghezza < 1 o > 20 [error] 3. Contiene caratteri speciali [error]
<b>Parametro :</b> Password <b>Formato :</b> <code>^[a-zA-Z0-9\.-]{1,10}\$</code>	
<b>Formato [FP]</b>	1. Rispecchia il formato 2. Lunghezza < 1 o > 10 [error]
<b>Parametro :</b> Conferma Password <b>Formato :</b> <code>^[a-zA-Z0-9\.-]{1,10}\$</code>	
<b>Formato [FCP]</b>	1. Rispecchia il formato [if property precedente FPOK] 2. Lunghezza < 1 o > 10 [if property precedente FPOK] [error]
<b>Corrisponde [CCP]</b>	1. Parametro uguale a "Password" [if property FCPOK] 2. Parametro diverso da "Password" [if property FCPOK]
<b>Parametro :</b> Matricola <b>Formato :</b> <code>^[0-9]{4}\$</code>	

## PLAN

<b>Formato [FMA]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato</li> <li>2. Non rispecchia il formato [error]</li> </ol>
----------------------	--------------------------------------------------------------------------------------------------------------------------

### Test Cases:

Codice	Combinazione	Esito
TC_1.1_1	FN2	Errato
TC_1.1_2	FN1,FCO2	Errato
TC_1.1_3	FN1,FCO1,FU2	Errato
TC_1.1_4	FN1,FCO1,FU1	Errato
TC_1.1.5	FN1,FCO1,FU1,PDU1,FP2	Errato
TC_1.1.6	FN1,FCO1,FU1,PDU1,FP1	Errato
TC_1.1.7	FN1,FN2,FCO1,FCO2,FU1,PDU1,FP2,FMA	Corretto

## 5.2. Login

### Category Partition

<b>Parametro :</b> Username(e-mail) <b>Formato:</b> ^[a-zA-Z0-9.!#\$%&'*/=?^_`{}~]+@studenti.unisa.it\$	
<b>Formato[FU]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato</li> <li>2. Contiene caratteri speciali [error]</li> <li>3. Dopo la “@” dominio differente da studenti.unisa.it [error]</li> </ol>
<b>Presente nel database[PDU]</b>	<ol style="list-style-type: none"> <li>1. Presente nel database [if formatoFUOK and formato FPOK]</li> <li>2. Non è presente nel database [if formato FUOK and formatoFPOK] [error]</li> </ol>

<b>Parametro :</b> Password <b>Formato :</b> ^[a-zA-Z0-9\.-_]{1,10}\$
--------------------------------------------------------------------------



## PLAN

<b>Formato[FP]</b>	<ol style="list-style-type: none"> <li>1. Rispecchia il formato [if formatoFUOK]</li> <li>2. Lunghezza &lt;1 o &gt;10 [if formatoFUOK] [error]</li> </ol>
<b>Associata [A]</b>	<ol style="list-style-type: none"> <li>1. Password associata [if propertyPDUOK]</li> <li>2. Non corrisponde con la password associata [if propertyPDUOK] [error]</li> </ol>

### Test Cases:

<b>Codice</b>	<b>Combinazione</b>	<b>Esito</b>
TC_1.2_1	FU1,FP2	Errato
TC_1.2_2	FU3,FP1	Errato
TC_1.2_3	FU1,PDU1,FP1,A2	Errato
TC_1.2_4	FU1,PDU1,FP1,A1	Corretto