



# UTP

# Unit Test Plan

M.T.O. - Modulo di Tirocinio Online

Auriemma Antonio  
Napolitano Maddalena

Peluso Maurizio  
Scavone Francesca

Data	Versione	Descrizione	Autori
18/01/2017	0.1	Prima stesura	AA, NM, PM, SF
21/01/2017	0.9	Completamento del documento	SF, AA
28/01/2018	1.0	Revisione finale	AA, NM, PM, SF

# Sommario

I.	<b>1.Introduzione</b>	4
II.	<b>2.Relazione con gli altri documenti</b>	4
III.	<b>3.Dettagli del level testing</b>	5
	3.1 Approccio di Unit Testing	5
	3.2 Componenti da testare	5
IV.	<b>4.Pass/fail criteria</b>	6

# 1. Introduzione

---

La stesura del documento di Unit Test Plan ha come obiettivo quello di identificare la strategia di testing di unità riguardanti il software M.T.O. - Moduli di Tirocinio Online.

Quest'attività sarà svolta in modo tale da testare il funzionamento di ogni classe, metodo, funzionalità di MTO, in modo da verificare che non vi sia alcun fault all'interno dell'implementazione e che il comportamento atteso corrisponda al comportamento effettivo. Gli eventuali errori identificati saranno analizzati, in modo da poterli prontamente correggere, o, più in generale, apportare modifiche per il miglioramento dei vari sottosistemi.

All'interno del documento saranno descritte, in particolare, le componenti da testare e il modo attraverso il quale deve essere eseguito.

# 2. Relazione con gli altri documenti

---

Lo unit test plan presenta una stretta connessione con il resto dei documenti che fino ad oggi sono stati prodotti, in quanto descrivono le generali funzionalità che devono essere osservate all'interno del sistema. Per tale ragione permetterà di rilevare eventuali differenze tra quanto veniva descritto all'interno di essi, ovvero ciò che si desiderava realizzare, e ciò che effettivamente il sistema esegue. In particolare, per verificare il corretto funzionamento del sistema, saranno utilizzati i test cases che sono stati individuati.

Di seguito vengono riportate le relazioni tra lo unit test plan e la documentazione precedente:

- **Requirements Analysis Document (RAD):** i requisiti funzionali e non funzionali del sistema sono ciò che collegano i due documenti, ciò perché i test verranno eseguiti su ogni funzionalità tenendo conto delle specifiche espresse all'interno del RAD;
- **System Design Document (SDD):** la divisione in sottosistemi e l'architettura definita all'interno di tale documento saranno oggetto del testing d'integrazione;
- **Object Design Document (ODD):** i packages e le varie classi individuate all'interno del documento sarà oggetto principale del testing.
- **Test Plan (TP):** i test cases individuati.

## 3. Dettagli del level testing

---

### 3.1 Approccio di Unit Testing

Il primo passo per valutare le funzionalità di un sistema è quello di verificare che le componenti che sono state prodotte, singolarmente, funzionino come progettato. Per questo motivo saranno sottoposte alla prima fase di testing le classi che compongono il sistema.

Per essere in linea con il budget, la tecnica che verrà utilizzata per il testing di unità è quella "Big-Bang": essa prevede che tutte le componenti vengano prima testate individualmente e poi insieme, come un unico sistema. Essendo il sistema MTO non molto ampio ed essendo il budget a disposizione del team limitato, si è scelto questo tipo di approccio, nonostante sia costoso in termini di rilevazione di fault.

### 3.2 Componenti da testare

Le componenti da testare, in termini di classi, sono elencati di seguito.

Gli oggetti di tipo Entity che verranno sottoposti al testing sono:

- DocumentoRichiesta.java;
- Firma.java;
- Tirocinio.java;
- Utente.java;
- UtenteType.java;

Le componenti DAO che saranno sottoposte al testing sono:

- AccountDAO.java;
- TirociniDAO.java;
- GestioneRichiesteTirocinioDAO.java.

Per gli oggetti di tipo Control, le componenti da testare sono:

- GestioneAccount.java;
- GestioneRichiesteTirocinio.java;
- GestioneTirocinio.java;
- Facade.java.

Per il livello di Presentazione, le componenti che saranno sottoposte al testing sono:

- LoginServlet.java
- LogoutServlet.java
- AggiungiTirocinioServlet.java
- EliminaListaTirocinioServlet.java
- ListaTirocinioServlet.java
- ModificaListaTirocinioServlet.java
- RegistrazioneServlet.java
- FirmaRichiestaServlet.java

- ListaRichiesteTirociniServlet.java
- VisualizzaStatoRichiestaServlet.java

## 4. Pass/fail criteria

---

Il testing darà esito positivo se l'output che verrà prodotto sarà diverso dall'output atteso, ciò sta a significare che la fase di testing avrà successo se verranno individuati degli errori. Una volta individuati sarà attuata una fase di revisione del codice sorgente ed effettuate le corrette modifiche. A questo punto verrà iterata la fase di testing per verificare che le correzioni apportate hanno risolto gli errori e che ciò non abbia avuto alcun impatto su gli altri componenti del sistema.

Nel caso in cui l'output osservato sarà uguale all'oracolo definito, allora il testing darà esito negativo.