



UNIT TEST PLAN

Riferimento	
Versione	1.0
Data	08/01/2019
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Mario De Cicco, Mario Greco, Giovanni Di Nocera, Anna Maria Raffaella Riccio



Revision History

Data	Versione	Descrizione	Autori
08/01/2019	Draft 1.0	Strutturazione Documento	Tutti



Sommario

1. Introduzione	5
2. Relazione con gli altri documenti	5
3. Dettagli del Level Testing Design	5
3.1 Approccio di Unit Testing	5
3.2 Componenti da testare	6
4. Pass/Fail Criteria	6



Laurea Magistrale in informatica-Università di Salerno
Corso di *Gestione dei Progetti Software*- Prof.ssa F.Ferrucci



1. Introduzione

Il testing di unità rappresenta la fase di testing in cui si assicura che le singole unità sviluppate funzionino nel modo corretto. Per unità si intende normalmente il minimo componente di un programma dotato di funzionamento autonomo.

Questo documento ha il compito di identificare la strategia di testing di unità per il sistema. In particolare, saranno specificate le componenti da testare e il modo in cui il testing dovrà essere eseguito.

2. Relazione con gli altri documenti

Per verificare se l'implementazione delle varie componenti del sistema è corretta, sono stati effettuati dei test cases basati sui requisiti funzionali identificati nella fase di analisi e specifica dei requisiti. I documenti ai quali si fa riferimento sono:

- Test Plan
- RAD
- SDD
- ODD

3. Dettagli del Level Testing Design

3.1 Approccio di Unit Testing

La prima modalità per valutare le funzionalità di un sistema è quella di verificare che le componenti sviluppate funzionino singolarmente, quindi, all'interno del sistema dovranno essere testati innanzitutto gli oggetti Entity.

Nel nostro caso per il testing di unità è necessario utilizzare un approccio di tipo White-Box. In particolare, lo sviluppo dei casi di test sarà effettuato utilizzando il framework JUnit.



3.2 Componenti da testare

Di seguito vengono elencati le componenti del sistema da testare, nello specifico gli oggetti Entity:

- AziendaBean.java
- ConvenzioneBean.java
- ImpiegatoBean.java
- QuestionarioAziendaBean.java
- QuestionarioStudianteBean.java
- StudenteBean.java
- TirocinioBean.java
- TutorBean.java

4. Pass/Fail Criteria

Il testing ha “successo” quando il risultato reale non corrisponde al risultato atteso, altrimenti il test fallisce e in questo caso il risultato osservato è in linea con l’oracolo. Quindi la fase di testing avrà successo se individuerà una failure. In tal caso questa verrà analizzata e, se legata ad un fault, si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica non abbia impattato su altri componenti del sistema. In particolare:

- Ogni test eseguito è richiesto poiché ogni minimo cambio di implementazione ha bisogno di essere testato almeno una volta.
- Non sono accettati difetti critici.

Il responsabile dei testing, quindi, dovrà conoscere e prevedere gli output corretti.