Ingegneria del Software

Università degli Studi di Salerno

e-Sport

Unit Test Document

1 Introduzione

Il testing di unità rappresenta una delle fasi di testing più importanti, in quanto consiste nella verifica e nel collaudo delle singole unità software di un sistema. Essendo così importante, va affrontata con particolare cautela, altrimenti potrebbero verificarsi imprevisti con conseguente slittamento dei tempi.

Il documento di Unit Test Plan descrive l'approccio al testing ed il framework generale che guiderà i test.

Ha come scopo l’assicurarsi la rimozione dei difetti critici prima dell’inizio dei prossimi livelli di test.

Il documento introduce quindi:

- Approccio di unit testing, ovvero le regole alla base del test e la descrizione del processo di impostazione di un test valido

- Pass\fail criteria per la gestione dei test: processo per gestire la logistica del test e tutti gli eventi che si verificano durante l'esecuzione

2 Relazione con altri documenti

Dalla correlazione ad altri documenti derivano vari criteri di accettazione del test. I documenti di test case specification devono essere disponibili prima dell'inizio della fase di progettazione del test.

Per individuare correttamente i test case si terrà conto dei documenti prodotti precedentemente. Infatti, la fase di testing è strettamene legata alle fasi precedenti che costituiranno un punto di partenza indispensabile per poter effettuare un testing corretto ed adeguato e per verificare che il sistema proposto corrisponda a quello desiderato .

2.1 Relazione con RAD

La relazione tra Unit Test Plan e RAD concerne i requisiti funzionali e non funzionali del sistema, in quanto i test che saranno eseguiti su ogni funzionalità terranno conto delle specifiche espresse in esso.

2.2 Relazione con SDD

Nel System Design Document si è suddiviso il sistema in sottosistemi e l’architettura in tre livelli: Presentation Layer, Application Layer e Data Layer.

Il test dei vari componenti dovrà rimanere il più possibile fedele a tali suddivisioni.

2.3 Relazione con ODD

Il test d’integrazione dovrà attenersi quanto più possibile alle interfacce delle classi e i package definiti nell’ODD.

3 Dettagli di Unit Testing

3.1 Approccio di Unit Testing

Il testing si compone di tre fasi.

Inizialmente verranno eseguiti i test di unità dei singoli componenti in modo da testare la correttezza di ciascuna unità constatando il corretto

funzionamento delle singole unità di codice. Questa fase verrà effettuata al completamento di ogni unità realizzata per poter individuare tempestivamente gli errori presenti nel codice.

3.2 Testing di unità

Per effettuare il testing delle singoli componenti del sistema verrà utilizzata la tecnica del “Black-Box testing” attraverso il framework JUnit. In questa fase saranno analizzate le funzionalità dell’applicazione ed il comportamento delle singole componenti senza tener conto della loro struttura interna ma focalizzandosi sui comportamenti di I/O. I risultati del testing verranno analizzati e usati per correggere gli errori causa dei fallimenti del sistema. Se si dovessero verificare errori con dei risultati inattesi si interverrà tempestivamente sulla componente in modo da renderla correttamente funzionante e procedere con le fasi di testing successive.

4 Pass/fail criteria

Lo scopo del testing è quello di dimostrare la presenza di faults (errori) all’interno del sistema. Le attività di testing, infatti, saranno mirate all’identificazione dei faults e alla loro successiva correzione.

Il testing avrà successo se l’output osservato risulterà diverso dall’output atteso, quindi si parla di successo per il test se esso rileva una failure. Essa verrà poi 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.

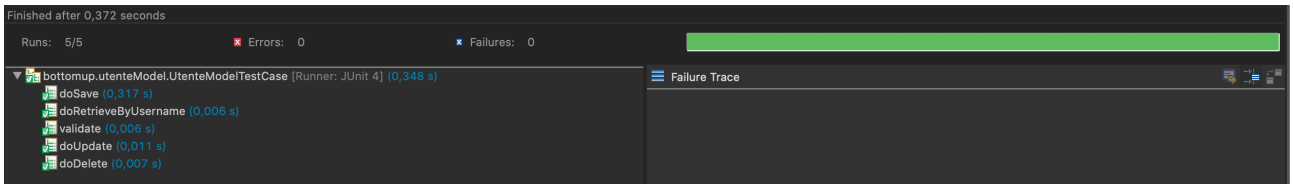
Invece, il test fallirà se non individuerà un fault.

5 Dettagli test results

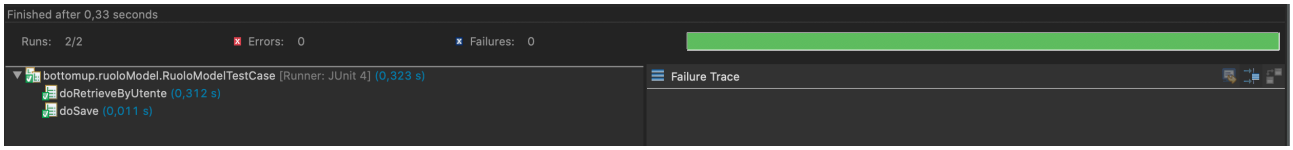
5.1 Test di Unità

Di seguito sono riportati tutti i test effettuati per le classi “Model” del sistema ClipShot tramite testing JUNIT.

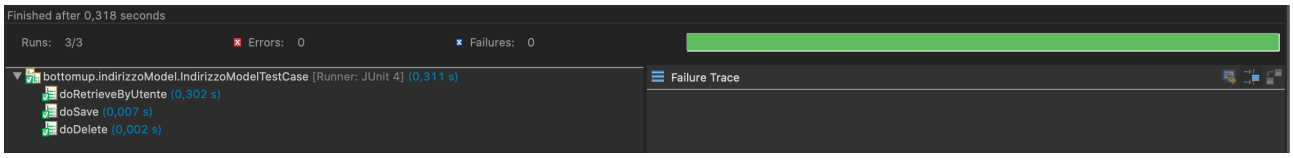
5.1.1 UtenteModel



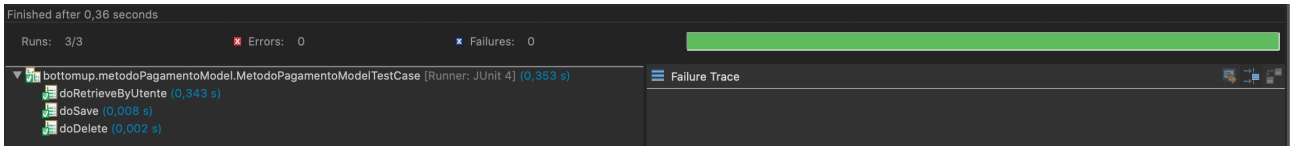
5.1.2 RuoloModel



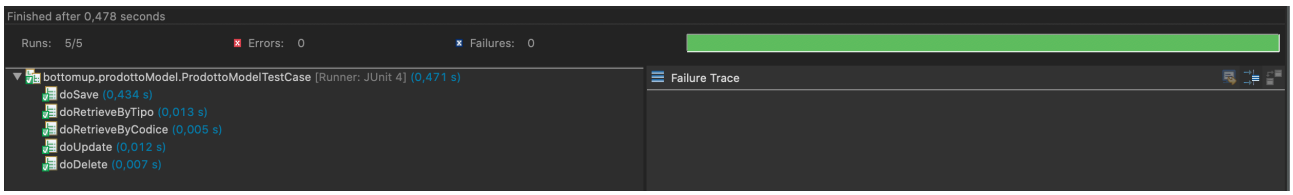
5.1.3 IndirizzoModel



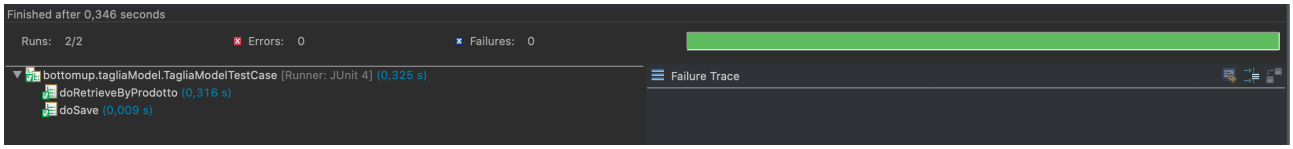
5.1.4 MetodoPagamentoModel



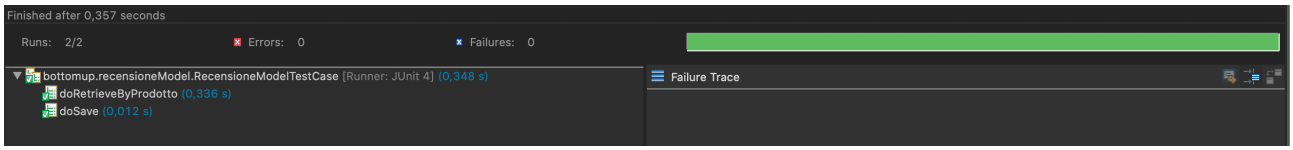
5.1.5 ProdottoModel



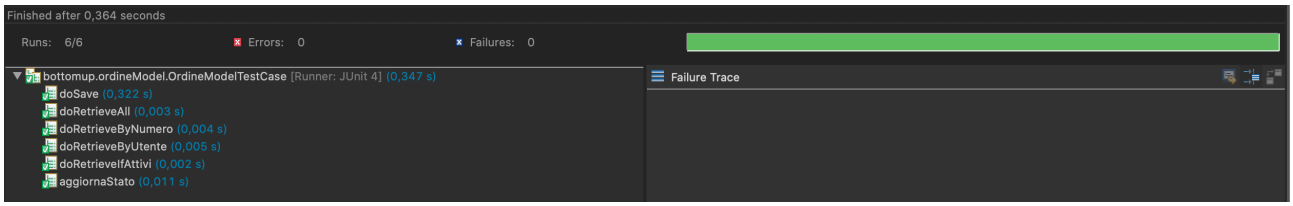
5.1.6 TagliaModel



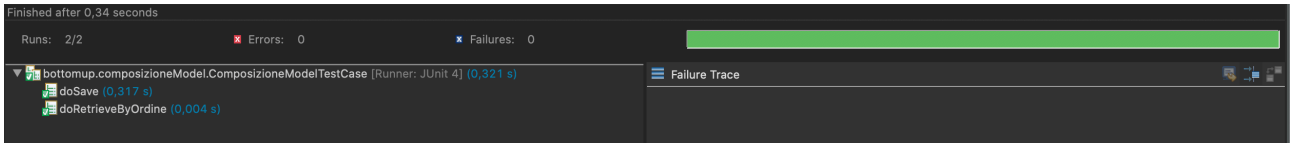
5.1.7 RecensioneModel



5.1.8 OrdineModel



5.1.9 ComposizioneModel



5.1.10 CarrelloBean