UTP

UNIT TEST PLAN

Sommario

1. Introduzione

Il testing di unità rappresenta una delle fasi di testing più importanti, in quanto consiste nella verifica e il collaudo di singole unità software di un sistema. Tuttavia, questa fase è anche molto delicata e può essere causa di imprevisti e slittamenti dei tempi a causa di errori e malfunzionamento del sistema.

Questo Unit Test Plan scrive quindi l'approccio di test ed il framework generale che guiderà i test della piattaforma Tutto-Elettronica. Ha come scopo l'assicurarsi che i difetti critici vengano rimossi prima che possano iniziare i prossimi livelli di test.

Il documento introduce quindi:

• approccio di unit testing, ovvero le regole su cui basare il test e la descrizione del processo per impostare un test valido;

• Pass\fail criteria 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

Essendo chiaramente in relazione con altri documenti, ne 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, in quanto saranno un punto di partenza indispensabile per poter effettuare un testing corretto ed adeguato e per verificare che il sistema desiderato sia simile a quello proposto.

2.1 Relazione con RAD

La relazione tra Unit Test Plan e RAD riguarda in particolare i requisiti funzionali e non funzionali del sistema poiché i test che saranno eseguiti su ogni funzionalità terranno conto delle specifiche espresse in tale documento.

2.2 Relazione con SDD

Nel System Design Document abbiamo suddiviso il nostro sistema in sottosistemi e l'architettura in tre livelli: Presentation Layer, Application Layer e Storage Layer. Il test dei vari componenti deve rimanere fedele a queste suddivisioni il più possibile.

2.3 Relazione con ODD

Il test d'integrazione farà quanto più riferimento possibile alle interfacce delle classi e i package definiti nell'ODD.

Inoltre, lo sviluppo deve essere completato ed i risultati condivisi tra il team di test per evitare i difetti duplicati. Infine, l'ambiente di test deve essere installato, configurato e pronto per l'uso dell'applicazione.

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 nello specifico la correttezza di ciascuna unità andando a constatare il corretto funzionamento di tutte le 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 di ogni singola componente del sistema verrà utilizzata la tecnica "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. I risultati del testing verranno analizzati e usati per correggere gli errori che causano il fallimento del sistema. Se si verifica un errore con dei risultati inattesi si interviene in maniera tempestiva 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 di questi faults e ad un successivo intervento per eliminarne la presenza. Il testing avrà successo se l'output osservato è diverso dall'output atteso: ciò significa che parliamo di successo se il test rileva un 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.

Viceversa, si parlerà di fallimento se il test non riesce ad individuare un errore.

5.Overview test results

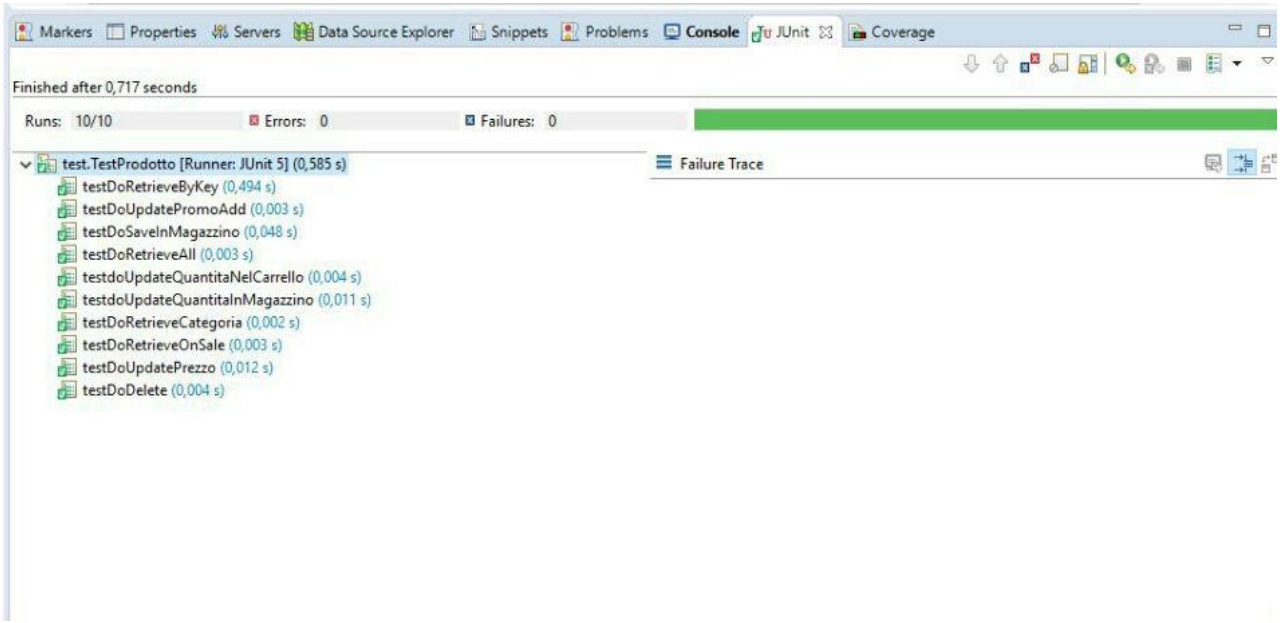
Per informazioni dettagliate sugli item testati e sui risultati si faccia riferimento al paragrafo successivo.

5.1.Dettagli test results

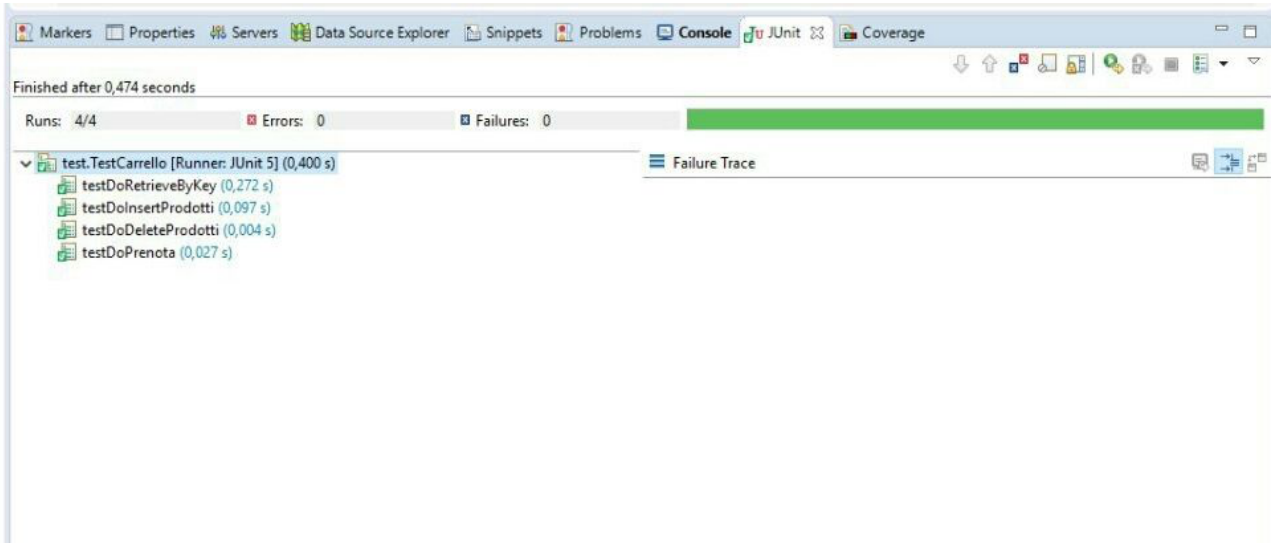
5.1.1. Test di Unità

Di seguito sono riportati tutti i test effettuati per le classi 'Manager' del sistema Tutto-Elettronica tramite testing JUNIT**.**

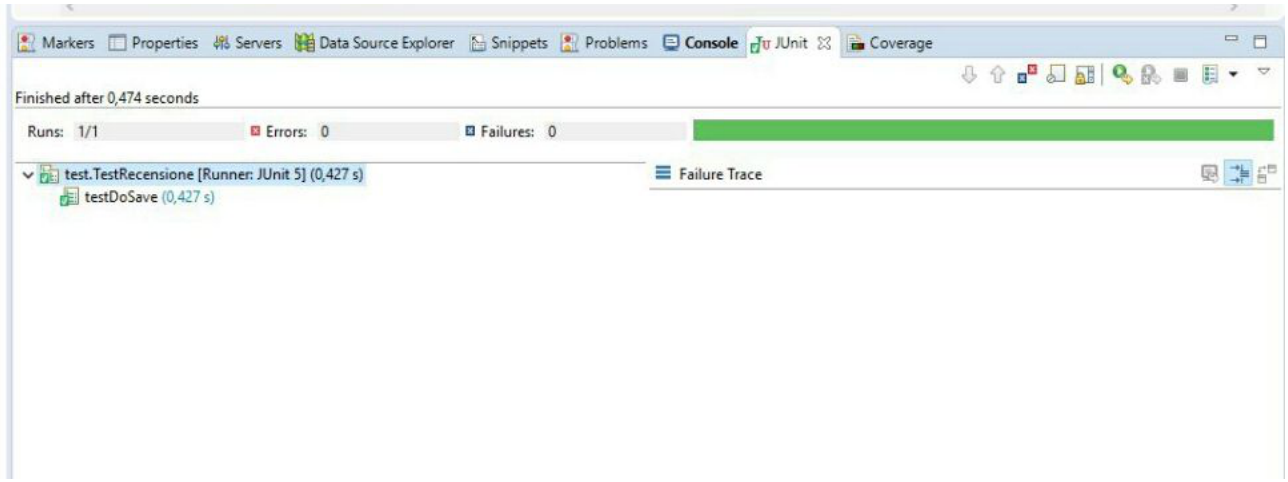
-Test ProdottoManager



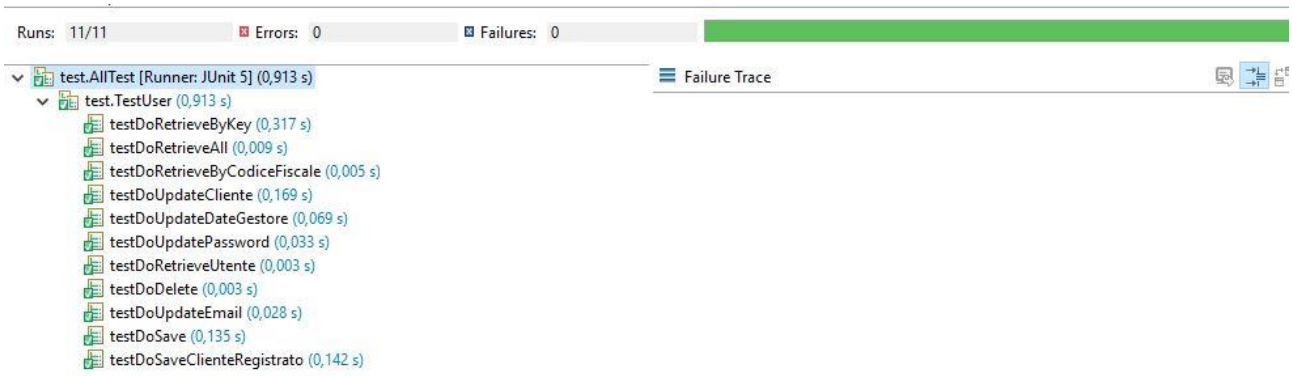
-Test CarrelloManager



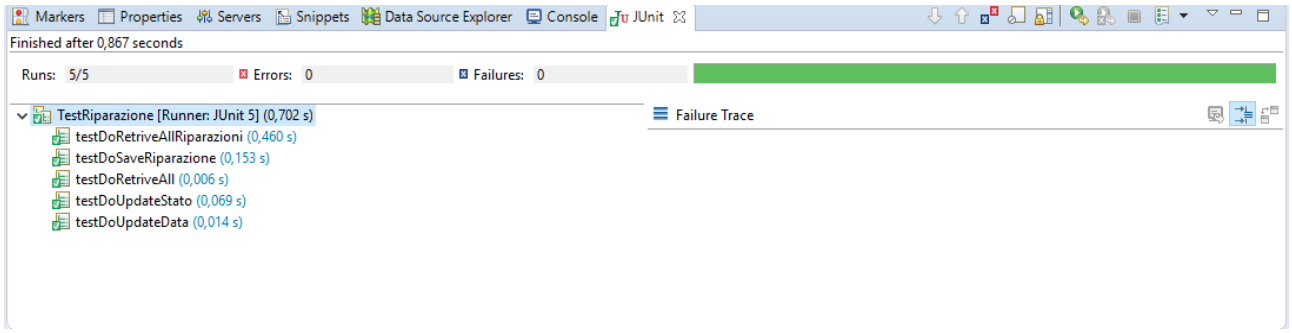
-Test RecensioneManager



-Test UserManager



-Test RiparazioneManager



6. Conclusioni

La fase di test si è quindi confermata una delle parti più importanti della realizzazione di un progetto software; in quanto, pur rimanendo fedeli ai tempi di consegna e alla realizzazione ed implementazione del codice, ci ha sicuramente consentito di migliorare la qualità dello stesso e di rendere più affidabile la piattaforma Tutto-Elettronica.