

Progetto IS – D&D



TP – Test Plan

TEAM – DAVIDE ALBANESE, DAVIDE SCUTIERO

1.Introduzione

Andiamo ad affrontare una delle fasi più delicate della progettazione, ovvero la fase di testing. Essa consiste nel verificare l' interazione tra due o più componenti e ne controlla il corretto funzionamento.

Questo documento ha il compito di individuare la strategia di testing di integrazione per il sistema Devices and Developments (D&D).

2. Riferimenti

Per verificare la corretta integrazione dei sottosistemi del sistema D&D sono stati predisposti dei test case basati sui vari sottosistemi realizzati in fase di Requirements Analysis .

Il documento di riferimento è quindi RAD- D&D .

3. Test di integrazione

3.1 Approccio di Integration Test

La strategia scelta di testing è di tipo “bottom-up” che prevede che i sottosistemi per i layer più bassi siano testati individualmente e che siano testati congiuntamente ai layer di livello superiore. Abbiamo scelto di utilizzare questo tipo di strategia per il semplice motivo che il nostro sistema è stato implementato allo stesso modo. E’ nostra intenzione rendere il più semplice possibile la struttura di base per renderla più comprensibile e analizzabile. Abbiamo optato per tre diversi layer , quali: Storage, Logic e Interface. Per testare lo strato di storage è sufficiente eseguire le query progettate, mentre per il logic layer sarà sufficiente eseguire tutti i metodi implementati.

3.1 Componenti da testare

La scelta delle componenti da testare segue la strategia Bottom-up.

Per quanto riguarda lo Storage Layer, quindi, la componente da testare è:

1. Database

Come abbiamo detto in precedenza, per testare questo layer sarà sufficiente eseguire le query che abbiamo progettato per il nostro sistema.

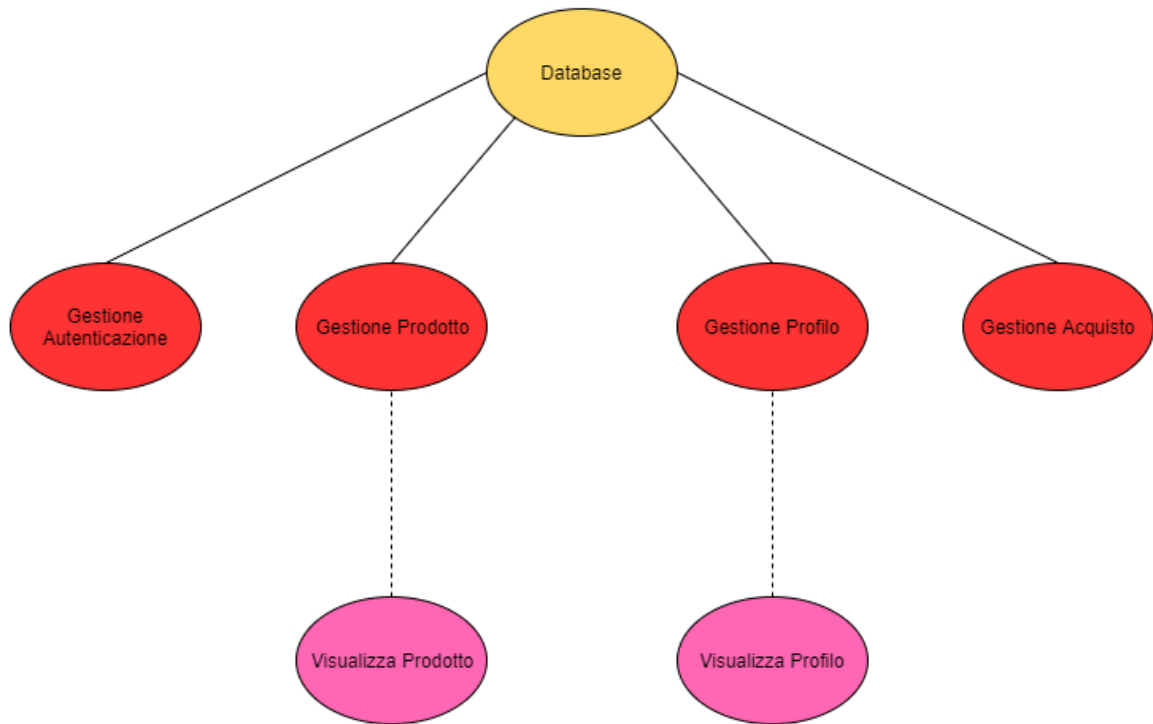
Per quanto riguarda il Logic Layer le componenti da testare sono:

1. Gestione Autenticazione
2. Gestione Prodotto
3. Gestione Acquisto
4. Gestione Profilo

Si effettuerà il testing per ogni tipo di gestione singolarmente e successivamente si simulerà la comunicazione di ogni singolo componente con la gestione dell’autenticazione per verificarne il corretto funzionamento. Successivamente si verificherà l’integrazione delle singole componenti con il database e in seguito rapportate alla gestione autenticazione.

Per quanto riguarda l’Interface Layer:

1. Visualizza Ordine
2. Visualizza Profilo



4. Unit Testing

L'obiettivo dello Unit Testing è che ogni sottosistema sia codificato e che svolga le funzionalità previste.

Le unità candidate per il testing sono prese dal modello a oggetti e dalla decomposizione dei sottosistemi che abbiamo realizzato. I sottosistemi dovranno essere testati prima individualmente e una volta appurata l'integrità del sistema, essi andranno testati simultaneamente. Il test verrà effettuato solo dopo che le classi sono state ritenute "stabili" ovvero , nessuna modifica verrà apportata, e tutte le funzionalità sono state implementate.

5. Pass/Fail - Criteri di valutazione

Il testing ha successo se l'output osservato è diverso dall'output ottenuto : ciò significa che si parla di SUCCESSO se il test rileva un failure.

Se si verifica un successo quindi, saremo in grado di analizzare il problema e di correggerlo.

Dopo le opportune correzioni il test verrà ripetuto per verificare l'integrità del sistema;insomma ci accerteremo che le altre classi funzionino correttamente.

Si parla invece di FALLIMENTO come si può immaginare, se il test non rileva alcun tipo di errore.