

ITP

Integration

Test

TutoratoSmart

|  |  |
| --- | --- |
| Riferimento |  |
| Versione | 0.2 |
| Data | 16/12/2019 |
| Destinatario | Prof.ssa F. Ferrucci |
| Presentato da | Marco Delle Cave, Francesco Pagano,  Manuel Pisciotta, Alessia Olivieri |
| Approvato da |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Cambiamenti** | **Autori** |
| 14/12/2019 | 0.1 | Definizione introduzione, scelta della tecnica di test di integrazione, definizioni di pass/fail criteria | Delle Cave Marco, Pagano Francesco |
| 16/12/2019 | 0.2 | Revisione IntegrationTest | Pisciotta Manuel |

Sommario

[**1. Introduzione 3**](#_Toc27406557)

[**2. Riferimenti 4**](#_Toc27406558)

[**3. Test di Integrazione 4**](#_Toc27406559)

[**3.1 Approccio di Integration Testing 4**](#_Toc27406560)

[**3.2 Componenti da testare 4**](#_Toc27406561)

[**4. Pass/Fail Criteri 6**](#_Toc27406562)

# Introduzione

Il testing di integrazione rappresenta una delle fasi di testing più importanti, in quanto consiste nella verifica delle interazioni tra due o più componenti.

L’obiettivo del testing consiste nella **verifica della corretta interazione tra le componenti** e il rispetto delle interfacce, secondo quanto stabilito nelle Specifiche di Integrazione.

Questo documento ha il compito di identificare la strategia di testing di integrazione per il sistema TutoratoSmart (TS).

# Riferimenti

Per verificare la corretta integrazione dei sottosistemi del sistema **TutoratoSmart** sono stati predisposti dei test case basati sulla divisione in sottosistemi proposta in fase di System Design.

Il documento di riferimento è: TS\_SDD­\_0.6.

# Test di Integrazione

## 3.1 Approccio di Integration Testing

La strategia adottata per il testing di integrazione è quella di tipo “Bottom-up” la cui strategia prevede che i sottosistemi nel layer più in basso della gerarchia dopo essere stati testati individualmente, vengano testati congiuntamente ai sottosistemi nel layer di livello superiore, cioè il layer della logica; al passo successivo vengono testati i due strati sottostanti con il layer di presentazione.

La scelta dell’approccio di testing bottom-up non comporta l’utilizzo di test stub, inoltre data la natura del sistema non sarà necessario l’utilizzo di test driver.

Infatti per testare il layer dei dati è sufficiente l’esecuzione delle query e la visualizzazione dei risultati; per testare il layer di logica è sufficiente l’esecuzione dei metodi implementati.

## 3.2 Componenti da testare

La scelta delle componenti da testare segue la decisione di eseguire la strategia di testing Bottom-up.

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

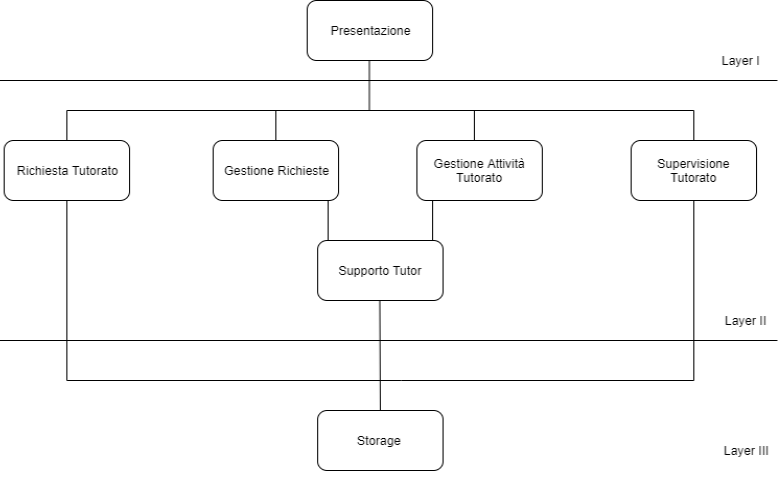
* Storage.

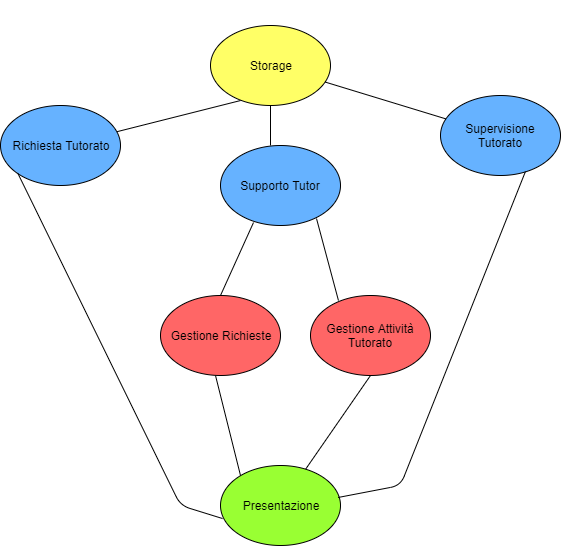
Per quanto riguarda il Logic Layer, le componenti da testare sono:

* Richieste Tutorato;
* Gestione Richieste;
* Gestione Attività Tutorato;
* Supervisione Tutorato;
* Supporto Tutorato.

Per quanto riguarda il Presentation layer, la componente da testare è:

* Presentazione.





# Pass/Fail Criteri

Il testing ha successo se l’output osservato è diverso dall’output atteso: ciò significa che parliamo di SUCCESSO se il test 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.

Viceversa parliamo di FALLIMENTO se il test non riesce ad individuare un errore.