

|  |  |
| --- | --- |
| Riferimento |  |
| Versione | 1.0 |
| Data | 16/12/2018 |
| Destinatario | Filomena Ferrucci – Carmine Gravino |
| Presentato Da | Daniele Cerardi – Antonio Junior Citarella – Antonio Vitale – Francesco Di Lascio – Simone Pirozzi |
| Approvato Da |  |

ITP Integration Test Plan

DInfBoard

|  |  |
| --- | --- |
| Nome | Matricola |
| Daniele Cerardi | 0512104555 |
| Antonio Junior Citarella | 0512104543 |
| Francesco Di Lascio | 0512103219 |
| Simone Pirozzi | 051214723 |
| Antonio Vitale | 0512104819 |

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autori |
| 15/12/2018 | 0.1 | Stesura del template. | Antonio Junior Citarella |
| 16/12/2018 | 0.2 | Stesura documento completo. | Daniele Cerardi |

Sommario

1. [Introduzione 0](#_Toc436918708)
2. [Riferimenti 0](#_Toc436918709)

[Approccio di Integration Testing](#_Toc436918710)

[Componenti da testare](#_Toc436918711)

1. [Pass/Fail Criteri 2](#_Toc436918712)

# 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 DINFBoard(DIB).

# Riferimenti

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

Il documento di riferimento è: DIB\_SDD.

# Test di Integrazione

## 3.1 Approccio di Integration Testing

La strategia adottata per il testing di integrazione è quella di tipo “Bottom-up”, la quale prevede che i sottosistemi dei layer di livello più basso vengano testati prima in maniera indipendente e poi in relazione ai livelli successivi nella gerarchia.

La scelta dell’approccio di testing bottom-up non comporta l’utilizzo di test stub; inoltre, non sarà necessario l’utilizzo di test driver.

Il testing delle operazioni sui dati persistenti richiederà la semplice sottomissione e memorizzazione di query contenenti i dettagli di un’entità del sistema.

La logica funzionale sarà testata sulla base dell’esecuzione dei metodi associati alle varie operazioni del sistema.

## 3.2 Componenti da testare

Il testing di integrazione è basato su una strategia di tipo bottom-up e l’architettura del sistema è di tipo MVC.

Per quanto riguarda la gestione dei dati persistenti, ovvero il Model, la componente da testare è:

* Storage.

Per quanto riguarda la logica funzionale e di business,ovvero il Controller le componenti da testare sono:

* UtenteSottosistema;
* AnnuncioSottosistema;
* AmministratoreSottosistema;
* ControlloSottosistema.

Per quanto riguarda il lato di della presentazione, ovvero la View, la componente da testare è:

* GUI

# 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.

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