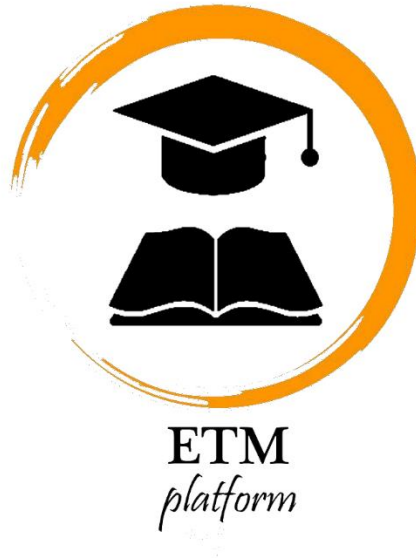




Laurea Magistrale in informatica - Università di Salerno  
Corso di *Ingegneria Gestione ed Evoluzione del Software*



# ITD

# Integration Test Design

Riferimento	
Versione	1.1
Data	28/08/2019
Destinatario	Prof. Andrea De Lucia
Presentato da	Marco della Medaglia Giuseppe D'Avanzo Luigi Cerreto



## Sommario

Revision History .....	<b>Errore. Il segnalibro non è definito.</b>
1. Introduzione .....	3
1.1 Riferimenti .....	3
2. Dettagli per il testing di integrazione .....	4
2.1 Features da testare.....	4
2.2 Features da non testare .....	4
2.3 Approccio.....	4
2.4 Pass/fail criteria .....	5
2.5 Criteri di sospensione e ripristino.....	5
3. Integration Test Case.....	6
3.1 ITC_AUT - AutenticazioneManager .....	6
3.2 ITC_USER - UtenteManager.....	6
3.3 ITC_TESI – PropostaTesiManager .....	7
3.4 ITC_CONSEGNA – ConsegnaManager .....	8
3.5 ITC_ADMIN - AmministratoreManager .....	8
3.6 ITC_ATTIVITA – AttivitàManager .....	9
3.7 ITC_FILE – FileManager.....	9
3.8 ITC_PARTECIPA – PartecipaManager .....	10
4. Glossario .....	11



# 1. Introduzione

---

Lo scopo di questo documento è quello di pianificare il test di integrazione della piattaforma ETM: verranno riportate le strategie adottate per il testing, gli strumenti utilizzati e le funzionalità testate.

Nel test si integrano le varie componenti del sistema e si verifica che il software risulti funzionante. L'approccio utilizzato sarà di tipo Bottom-up, ed ogni membro del team è responsabile del testing di alcune componenti del sistema e procederà partendo dal testare i bean delle varie classi implementate per poi procedere con i relativi manager.

Sarà fatto utilizzo di Travis per il testing delle nuove funzionalità integrate, di cui si è effettuato il commit su GitHub.

## 1.1 Riferimenti

Il documento fa riferimento alle ultime versioni dei precedenti documenti rilasciati, in particolare:

- ETM\_RAD\_v1.3
- ETM\_SDD\_v1.0
- ETM\_ODD\_v1.0
- ETM\_SPMP\_v1.0
- ETM\_TCS\_v1.0



## 2. Dettagli per il testing di integrazione

---

### 2.1 Features da testare

I sottosistemi che verranno sottoposti al test di integrazione sono i seguenti:

- Package Autenticazione: verrà effettuato il test della classe AutenticazioneManager.java;
- Package Utente: verrà effettuato il test della classe UtenteManager.java;
- Package Proposta Tesi: verrà effettuato il test della classe PropostaTesiManager.java;
- Package Consegna: verrà effettuato il test della classe ConsegnaManager.java;
- Package Area Condivisa: verrà effettuato il test della classe AreaCondivisaManager.java;
- Package Amministratore: verrà effettuato il test della classe AmministratoreManager.java;
- Package Attività: verrà effettuato il test della classe AttivitàManager.java;

### 2.2 Features da non testare

Non verranno prodotte classi di test per le servlet dei vari package, ma esse verranno comunque testate per poterne verificare l'effettivo funzionamento e per poter poi realizzare il test di sistema.

### 2.3 Approccio

L'approccio utilizzato per eseguire il test di integrazione sarà di tipo Bottom-up (dal basso verso l'alto). Ciò comporta il test e l'integrazione delle singole componenti partendo dal livello più basso, ovvero lo Storage, per poi proseguire con l'Application Layer.

Avvalendosi di opportuni driver e del software JUnit, verranno testati in una prima fase i bean dei vari packages e, solo dopo averne verificato il corretto funzionamento, si procederà al livello successivo testando i manager. Il tutto ovviamente sarà vincolato ad un database che sarà di appoggio a questa fase di testing.



Inoltre, sarà utilizzato Travis, che si occuperà di testare l'integrazione delle componenti procedendo nello sviluppo.

## 2.4 Pass/fail criteria

Nel caso in cui vengano riscontrati errori durante la fase di testing d'integrazione (e ciò è ammissibile dato che il software non si trova ancora in fase di deployment, e si stanno assemblando le varie componenti), si procederà con la correzione dei fault intervenendo direttamente sulle porzioni di codice che generano il problema.

## 2.5 Criteri di sospensione e ripristino

- Criteri di sospensione

Comprendono tutti quei casi critici di quando gli errori hanno un impatto dannoso sul progresso dell'attività di testing. Esempi possono essere:

- o Crash del database
- o Crash del server
- o Fallimento di funzionalità interne
- o Problemi relativi all'ambiente di sviluppo del testing

- Criteri di ripristino

La ripresa del sistema avviene solo quando tali errori vengono risolti, ripartendo dal test case che ha causato l'errore.



## 3. Integration Test Case

### 3.1 ITC\_AUT - AutenticazioneManager

Nome classe	AutenticazioneManager.java
Layer	Application Layer
ID Unit Test Case	Metodo
ITC_AUT_01	testSetValidazione()
ITC_AUT_02	testGetUtente()
ITC_AUT_03	testRegistraUtente()
ITC_AUT_04	testGetAdmin()
ITC_AUT_05	testGetPassword()

### 3.2 ITC\_USER - UtenteManager

Nome classe	UtenteManager.java
Layer	Application Layer
ID Unit Test Case	Metodo
ITC_USER_01	testGetInfo()
ITC_USER_02	testCercaUtente()
ITC_USER_03	testModificaPassword()
ITC_USER_04	testModificaUtente()
ITC_USER_05	testCercaListaUtenteNome()
ITC_USER_06	testCercaListaUtenteCognome()



### 3.3 ITC\_TESI – PropostaTesiManager

Nome classe	PropostaTesiManager.java
Layer	Application Layer
ID Unit Test Case	Metodo
ITC_TESI_01	testInserisciPropostaTesi()
ITC_TESI_02	testArchiviaPropostaTesi ()
ITC_TESI_03	testGetPropostaTesi ()
ITC_TESI_04	testChiudiPropostaTesi ()
ITC_TESI_05	testRimuoviPropostaTesi ()
ITC_TESI_06	testGetProposteTesiAttive ()
ITC_TESI_7	testAccettaRichiestaPartecipazione()
ITC_TESI_8	testRifiutaRichiestaPartecipazione()
ITC_TESI_9	testCercaRichiestePartecipazione()
ITC_TESI_10	testInserisciRichiestaPartecipazione()
ITC_TESI_11	testGetProposteTesiDocente()
ITC_TESI_12	testModificaPropostaTesi()
ITC_TESI_13	testGetRichiestaStudente()
ITC_TESI_14	testGetInsegnamenti()
ITC_TESI_15	testGetTesiRecenti()



### 3.4 ITC\_CONSEGNA – ConsegnaManager

Nome classe	ConsegnaManager.java
Layer	Application Layer
ID Unit Test Case	Metodo
ITC_CONSEGNA_01	testAggiungiConsegna ()
ITC_CONSEGNA_02	testModificaConsegna ()
ITC_CONSEGNA_03	testEliminaConsegna ()
ITC_CONSEGNA_04	testGetConsegna ()
ITC_CONSEGNA_05	testGetListaConsegne()

### 3.5 ITC\_ADMIN - AmministratoreManager

Nome classe	AmministratoreManager.java
Layer	Application Layer
ID Unit Test Case	Metodo
ITC_ADMIN_01	testGetListaUtente ()
ITC_ADMIN_02	testEliminaUtente ()
ITC_ADMIN_03	TestGetUtente()





### 3.6 ITC\_ATTIVITA – AttivitàManager

Nome classe	AmministratoreManager.java
Layer	Application Layer
ID Unit Test Case	Metodo
ITC_ATTIVITA_01	testAggiungiAttivita()
ITC_ATTIVITA_02	testGetListaAttivita()
ITC_ATTIVITA_03	testGetNotifiche()
ITC_ATTIVITA_04	testLeggiNotifiche()

### 3.7 ITC\_FILE – FileManager

Nome classe	AmministratoreManager.java
Layer	Application Layer
ID Unit Test Case	Metodo
ITC_FILE_01	testAggiungiFile()
ITC_FILE_02	testModificaFile()
ITC_FILE_03	testEliminaFile()
ITC_FILE_04	testGetFile()
ITC_FILE_05	testScaricaFile()
ITC_FILE_06	testGetListaFile()



### 3.8 ITC\_PARTECIPA – PartecipaManager

Nome classe	AmministratoreManager.java
Layer	Application Layer
ID Unit Test Case	Metodo
ITC_PARTECIPA_01	testInserisciPartecipazione()
ITC_PARTECIPA_02	testGetListaPartecipazione()



## 4. Glossario

---

**Proposta tesi:** corrisponde al tirocinio interno di uno studente; caricata da un docente, la proposta di tesi consiste in uno specifico lavoro per una problematica data.