

CERICinema
Integration Test Plan (ITP)
Versione 1.0



Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Edilio Massaro	0512103218
Ildo Tiberio	0512103134
Raffaele Squeglia	0512103122
Chiara Bufalino	0512102894

Scritto da:	
-------------	--

Revision History

[illegible]

Indice

1.	INTRODUZIONE	4
1.1.	<i>Identificativo del documento</i>	4
1.2.	<i>Scopo</i>	4
1.3.	<i>Riferimenti</i>	4
2.	DETTAGLI PER IL SEGUENTE LIVELLO DI TEST PLAN	5
2.1.	<i>Features da testare</i>	5
2.2.	<i>Approccio</i>	5
2.3.	<i>Pass/fail criteria</i>	5
2.4.	<i>Criteri di sospensione e ripristino</i>	5
3.	TEST MANAGEMENT	7
3.1.	<i>Ambiente di sviluppo e strumenti utilizzati</i>	7
4.	GENERAL	8
4.1.	<i>Test coverage</i>	8
4.2.	<i>Glossario</i>	8

1. INTRODUZIONE

Lo scopo di questo documento è quello di pianificare il test di integrazione del software CERICinema: verranno riportate le strategie adottate per il testing, gli strumenti utilizzati e le funzionalità testate. Verrà inoltre verificato il livello di affidabilità di CERICinema e apportate poi eventuali modifiche per correggere eventuali fault che si presenteranno.

1.1. *Identificativo del documento*

Il seguente documento è identificato in modo univoco con le seguenti proprietà: data di rilascio, i nomi di chi ha partecipato alla stesura e alla revisione del suddetto.

1.2. *Scopo*

Lo scopo del documento è quello di fornire una presentazione dettagliata della pianificazione del test di integrazione per il software CERICinema. 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 gruppo è responsabile del testing di alcune componenti del sistema e procederà sempre partendo dal testare i Bean delle varie classi implementate per poi procedere con i relativi Manager.

1.3. *Riferimenti*

- RAD_CERICinema.docx
- SDD_CERICinema_SDD.docx
- ODD_CERICinema.docx
- TPP_CERICinema.docx
- TCS_CERICinema.docx

2. DETTAGLI PER IL SEGUENTE LIVELLO DI TEST PLAN

In questa sezione del documento verranno forniti i dettagli relativi alle funzionalità ed agli strumenti utilizzati per eseguire il test di integrazione. Sarà inoltre fornita una matrice di tracciabilità per la fase di testing.

2.1. Features da testare

I sottosistemi che verranno sottoposti al test di integrità in questa fase sono i seguenti:

- Verrà effettuato il test della classe DBConnector.java

2.2. 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 beans 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.

2.3. 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.

2.4. 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:

- Crash del database
- Crash del server
- Fallimento di funzionalità interne

- 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. TEST MENAGEMENT

3.1. Ambiente di sviluppo e strumenti utilizzati

Le risorse software utilizzate per il testing sono:

- Eclipse: ambiente di sviluppo per Java.
- Microsoft Office: editor per la stesura dei documenti.
- JUnit: software per la realizzazione delle classi di test.

4. GENERAL

4.1. *Test coverage*

Oltre al test di unità, al test di integrazione e al test di sistema non sono previste ulteriori coperture per il software CERICinema.

4.2. *Glossario*

Definizioni:

- **Bean:** componente software riutilizzabile, usata per rappresentare le informazioni di interesse.
- **ODD:** Documento che riporta e analizza gli oggetti che compongono il sistema analizzando le componenti a più basso livello, riportandole così come saranno implementate.
- **RAD:** Documento di Raccolta e analisi dei Requisiti che contiene l'elenco dei requisiti funzionali e non funzionali individuati in fase di individuazione dei stessi e la loro analisi sotto forma di scenari e casi d'uso. I mock-up mostrano una possibile implementazione dell'interfaccia del sistema.
- **SDD:** Documento che riporta la progettazione del sistema come risultato di una prima fase di modellazione: contiene una suddivisione ad alto livello del sistema nei sottosistemi che lo comporranno.
- **TCS:** Documento che specifica i casi di test in tutti i loro dettagli.
- **TP:** Documento che descrive il piano di testing adottato nel progetto e la definizione dei casi di test.

Acronimi:

- **ODD:** Object Design Document;
- **RAD:** Requirement Analysis Document;
- **SDD:** System Design Document;
- **TCS:** Test case specification;
- **TP:** Test Plan;