

Università degli Studi di Salerno

Corso di Ingegneria del Software

CERICinema Unit Test Design (UTD) 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 DESIGN DEL TEST DI UNITA'	5
2.1.	<i>Funzionalità da testare</i>	5
2.2.	<i>Approccio</i>	6
2.3.	<i>Identificazione del test</i>	6
2.4.	<i>Criteri di successo/ fallimento</i>	11
2.5.	<i>Prodotti del test</i>	11
3.	GENERALI	12
3.1.	<i>Glossario</i>	12

1. INTRODUZIONE

Lo scopo di questo documento è quello di pianificare il design del test di unità del software CERICinema: verranno riportate le strategie adottate per il testing, gli strumenti utilizzati e le funzionalità testate. Nello specifico saranno riportati tutti i metodi esercitati sulle classi prese in considerazione.

1.1. *Identificativo del documento*

Il seguente documento è identificato in modo univoco dalle seguenti proprietà: data di rilascio, nomi di chi ha partecipato alla stesura e alla revisione del suddetto; tali informazioni sono riportate nel frontespizio.

1.2. *Scopo*

Il documento fa riferimento alla pianificazione del test di unità per il software CERICinema, all'interno del quale si verificano le singole unità di una classe e si controlla che i metodi e le classi abbiano il comportamento atteso. I membri del team hanno testato esclusivamente i Bean delle classi implementate nel sistema.

1.3. *Riferimenti*

- RAD_CERICinema.doc
- SDD_CERICinema.doc
- ODD_CERICinema.doc
- TPP_CERICinema.doc
- TCS_CERICinema.doc

2. DETTAGLI PER IL DESIGN DEL TEST DI UNITA'

In questa sezione del documento verranno forniti i dettagli relativi alle funzionalità ed all'approccio utilizzati per eseguire lo unit test design.

2.1. *Funzionalità da testare*

Il documento si propone di verificare il test di unità effettuato sui Bean presenti nel sistema; in particolare saranno presi in considerazione per ogni package:

- Verrà effettuato il test della classe Proiezione.java
- Verrà effettuato il test della classe Utente.java
- Verrà effettuato il test della classe Prenotazione.java
- Verrà effettuato il test della classe Film.java

2.2. *Approccio*

L'approccio utilizzato per eseguire il test di unità è stato del tipo black-box, in quanto l'attenzione è stata posta sul comportamento I/O del sistema. I test si sono basati, quindi, solo su ciò che il sistema poteva mostrare, senza conoscere l'implementazione delle classi stesse. I Bean sono stati esercitati tenendo conto anche del fatto che i possibili dati di input e output sono stati opportunamente suddivisi in categorie rappresentanti stati validi o non validi per gli stessi; di conseguenza, durante il test ci si è accertati anche della correttezza delle variabili usate. Avendo verificato il corretto funzionamento di queste classi, è possibile procedere al livello successivo; il framework di supporto utilizzato è JUnit.

2.3. *Identificazione del test*

UTC_PRZ: Proiezione.java

Nome Classe	Proiezione.java
ID Unit Test Case	Metodo
UTC_PRZ_1	testGetId()
UTC_PRZ_2	testSetId()
UTC_PRZ_3	testGetNomeProiezione()
UTC_PRZ_4	testSetNomeProiezione()
UTC_PRZ_5	testGetSala()
UTC_PRZ_6	testSetSala()
UTC_PRZ_7	testGetDataProiezione()
UTC_PRZ_8	testSetFilm()

UTC_U: Utente.java

Nome Classe	Utente.java
ID Unit Test Case	Metodo
UTC_U_1	testGetIdUtenti ()
UTC_U_2	testSetIdUtenti()
UTC_U_3	testGetUsername()
UTC_U_4	testSetUsername()
UTC_U_5	testGetNomeUtente()
UTC_U_5	testSetNomeUtente()
UTC_U_6	testGetCognomeUtente()
UTC_U_7	testSetCognomeUtente()
UTC_U_8	testGetResidenza()
UTC_U_9	testSetResidenza()
UTC_U_19	testGetDataNascita()
UTC_U_20	testSetDataNascita()

UTC_PRN: Prenotazione.java

Nome Classe	Prenotazione.java
ID Unit Test Case	Metodo
UTC_PRN_1	testGetIdPrenotazioni()
UTC_PRN_2	testSetIdPrenotazioni()
UTC_PRN_3	testGetProiezione()
UTC_PRN_4	testSetProiezione()
UTC_PRN_5	testGetSala()
UTC_PRN_6	testSetSala()

UTC_PRN_7	testGetPosto()
UTC_PRN_8	testSetPosto()
UTC_PRN_9	testGetUtente
UTC_PRN_10	testSetUtente

UTC_FM: Film.java

Nome Classe	Film.java
ID Unit Test Case	Metodo
UTC_FM_1	testGetIdFilm()
UTC_FM_2	testSetIdFilm()
UTC_FM_3	testGetNomeFilm()
UTC_FM_4	testSetNomeFilm()
UTC_FM_5	testGetDataUscita()
UTC_FM_6	testSetDataUscita()
UTC_FM_7	testGetGenere()
UTC_FM_8	testSetGenere()
UTC_FM_9	testGetDescrizione()
UTC_FM_10	testSetDescrizione()
UTC_FM_11	testGetLocandina()
UTC_FM_12	testSetLocandina()

2.4. *Criteri di successo/ fallimento*

Nel caso in cui siano stati riscontrati errori durante la fase di testing di unità, si è proceduto con la correzione dei fault intervenendo direttamente sulle porzioni di codice che generano il problema. Si è iterata, dunque, nuovamente la fase di testing per verificare che la modifica non avesse impattato su altre componenti del sistema, ma, al contrario, ne avesse migliorato il comportamento.

2.5. *Prodotti del test*

Si faccia riferimento alla sezione 2.2 di UTR_CERICinema.doc.

3. GENERALI

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;
- **SPMP:** Software Project Management Plan;
- **TCS:** Test case specification;
- **TP:** Test Plan;