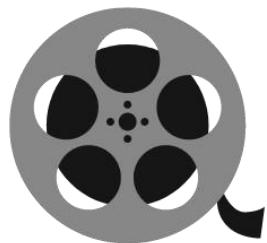




Università degli Studi di Salerno
Corso di Ingegneria del Software

Rated
Test Execution Report - Post ChangeRequest
Versione 1.0



RATED
•COMMUNITY REVIEWS•

24/01/2026

Membri del gruppo

Nome	Matricola
Francesco Rao	NF22500211
Bruno Nesticò	NF22500213
Roberto Fiorenza	NF22500212

Revision History

Data	Versione	Descrizione	Autore
24/01/2026	1.0	Prima stesura completa	Francesco Rao, Bruno Nesticò

Indice

1. Introduzione	3
2. Test di unità	3
3. Test di integrazione.....	4
4. Test di sistema	6
5. Coverage test effettuati	6

1. Introduzione

Per la definizione dei test frame si è utilizzato il metodo del category partition come specificato in TestCaseSpecification_RatedPreCR. Ogni scelta associata a una condizione di errore sarà verificata singolarmente, partendo dal presupposto che le altre categorie siano corrette. Non verranno quindi considerate combinazioni di errori tra diverse categorie. Ogni errore isolato sarà trattato come un caso di test.

Il risultato dei test verrà mostrato tramite screenshot del risultato esportato dall'IDE, al termine dell'esecuzione. Il codice utilizzato per il testing sarà disponibile sulla repository del progetto e organizzato nelle seguenti cartelle:

- **unit:** Contiene il codice dei test di unità, realizzati con JUnit e Mockito, per verificare il corretto funzionamento dei singoli metodi e delle unità di codice.
- **integration:** Contiene il codice dei test di integrazione, per validare l'interazione tra i diversi componenti del sistema.

I test documentati sono tutti quelli modificati o aggiunti in seguito ai cambiamenti del sistema legati alle Change Request.

2. Test di Unità

Profile Service Test

```
JUnit testAggiungiAllaWatchlist() (2,996 s)
JUnit testRimuoviFilmVisto() (0,012 s)
```

```
JUnit testAggiornaPreferenzeUtente_NullGeneri() (0,019 s)
JUnit testAggiornaPreferenzeUtente_WithGeneri() (0,009 s)
JUnit testIsFilmInWatchlist_True() (0,008 s)
JUnit testRetrieveWatchedFilms() (0,009 s)
JUnit testRetrieveWatchlist() (0,006 s)
JUnit testFindByUsername() (0,009 s)
JUnit testGetPreferenze() (0,004 s)
JUnit testIgnoreFilm() (0,007 s)
JUnit testAddPreferenza() (0,005 s)
JUnit testIsFilmVisto_True() (0,006 s)
JUnit testAggiungiFilmVisto() (0,003 s)
JUnit testRimuoviDallaWatchlist() (0,006 s)
```

```
JUnit testIsFilmVisto_False() (0,004 s)
```

RecensioniService Test

```
JUnit testDeleteReports() (0,008 s)
JUnit testAddValutazione_RemoveExistingLike() (0,006 s)
JUnit testAddValutazione_SwitchToDislike() (0,008 s)
JUnit testAddValutazione_RecensioneMissing() (0,008 s)
JUnit testAddRecensione_DuplicateDoesNotSave() (0,004 s)
```

```
JUnit testGetValutazioni() (0,004 s)
JUnit testGetRecensioni() (0,004 s)
JUnit testDeleteRecensione() (0,005 s)
```

CatalogoService Test

```
JUnit testGetFilmsFromRecensioni() (2,084 s)
JUnit testGetFilms() (0,007 s)
JUnit testAddFilm() (0,055 s)
```

```
JUnit testGetFilm() (0,004 s)
JUnit testRicercaFilm() (0,004 s)
JUnit testGetAllGeneri() (0,005 s)
JUnit testAggiungiFilm() (0,004 s)
```

```
JUnit testModifyFilm() (0,005 s)
JUnit testGetFilmCompatibili() (0,013 s)
JUnit testGetGeneri() (0,004 s)
JUnit testModifyFilm_UpdatesGeneri() (0,007 s)
```

3. Test di Integrazione

CatalogoService IntegrationTest

```
JUnit testFindGeneriByIdFilm() (0,000 s)
JUnit testGetFilmsByRecensioni_ShouldReturnMap() (0,000 s)
JUnit testAddFilm_AddsGeneriAssociations() (0,000 s)
JUnit testGetFilms_ReturnsAllFilms() (0,000 s)
JUnit testAggiungiFilm_ValidData_ShouldSaveToDatabase() (0,008 s)
JUnit testGetFilm_ShouldReturnSingleFilm() (0,008 s)
JUnit testGetAllGeneri_Service() (0,007 s)
JUnit testRicercaFilm_ExistingTitle_ShouldReturnResult() (0,006 s)
JUnit testModifyFilm_UpdatesGeneri() (0,017 s)
JUnit testDoRetrieveConsigliati() (0,020 s)
```

ProfileService IntegrationTest

```
JUnit testAggiornaPreferenzeUtente_Persist() (0,025 s)
JUnit testRetrieveWatchlist() (0,035 s)
JUnit testWatchlistLifecycle() (0,017 s)
JUnit testAggiungiFilmVisto_Duplicate_ShouldNotThrowAndNotDuplicate() (0,027 s)
JUnit testFilmVistoLifecycle() (0,015 s)
```

RecensioniServiceIntegrationTest

```
JUnit test GetAllRecensioniSegnalate() (0,019 s)
JUnit testAddValutazione_ChangeVote_ShouldSwitchCounters() (0,026 s)
JUnit testGetRecensione_ShouldReturnBean() (0,043 s)
JUnit testFindReportById_ShouldReturnBean() (0,025 s)
JUnit testFindRecensioniByUser() (0,021 s)
JUnit testReportAndDeleteReports() (0,036 s)
JUnit testAddValutazione_ToggleLike_ShouldRemove() (0,037 s)
JUnit testAddRecensione_AggiornaMediaFilm() (0,151 s)
JUnit testGetValutazioni_Service() (0,038 s)
```

4. Test di Sistema

Per rispettare le scadenze del progetto, come in quello originale, sono state testate manualmente e in maniera facilmente riproducibile tutte le nuove funzionalità introdotte nel sito in accordo con le Change Request. Tutti i test effettuati hanno avuto esito positivo, dimostrando che la piattaforma garantisce una corretta esperienza utente anche in seguito alle modifiche.

5. Coverage dei Test Effettuati

Considerando gli obiettivi di coverage definiti nel MasterTestPlan Document, sono stati espansi i test rispetto al progetto originale portando ai seguenti risultati:

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
C ValutazioneDAO		82%		71%	6	15	23	107	2	8	0	1
C InteresseDAO		79%		66%	5	12	19	84	1	6	0	1
C FilmDAO		89%		75%	6	18	24	151	2	10	0	1
C UtenteDAO		85%		62%	9	16	18	85	3	8	0	1
C VistoDAO		84%		75%	4	12	15	76	1	6	0	1
C ReportDAO		76%		62%	5	10	15	52	2	6	0	1
C PreferenzaDAO		75%		75%	3	8	16	54	2	6	0	1
C FilmGenereDAO		76%		75%	2	7	14	52	1	5	0	1
C GenereDAO		68%		75%	2	6	12	36	1	4	0	1
C RecensioneDAO		93%		78%	4	17	17	142	1	10	0	1
Total	442 of 2.836	84%	30 of 104	71%	46	121	173	839	16	69	0	10
sottosistemi.Gestione_Recensioni.service		88%		85%	5	26	10	105	1	12	0	1
sottosistemi.Gestione_Utenti.service		90%		92%	5	41	14	124	3	27	0	3
sottosistemi.Gestione_Catalogo.service		93%		100%	1	17	5	70	1	14	0	1

Alla luce di ciò si considerano soddisfatti gli obiettivi prefissati.