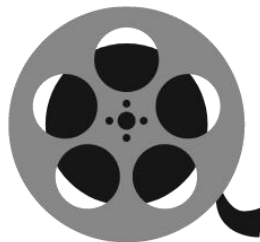




**Università degli Studi di Salerno**

**Corso di Ingegneria del Software**

**Rated  
Test Plan Document  
Versione 1.0**



**RATED**

• COMMUNITY REVIEWS •

Data: 24/11/2024

**Coordinatore del progetto:**

Nome	Matricola

**Partecipanti:**

Nome	Matricola
Francesco Rao	0512116836
Bruno Nesticò	0512117268

<b>Scritto da:</b>	Francesco Rao, Bruno Nesticò
--------------------	------------------------------

**Revision History**

Data	Versione	Descrizione	Autore
24/11/2024	1.0	Prima stesura completa	Francesco Rao, Bruno Nesticò

# Indice

1. Introduzione	4
2. Relazione con altri documenti	4
3. Feature da Testare e Non Testare	4
4. Pass/Fail Criteria	5
5. Approccio	5
6. Sospensione e ripristino	6
7. Materiale di testing	6

## 1. Introduzione

L'obiettivo del test è verificare la correttezza e il grado di affidabilità delle funzionalità principali della piattaforma **Rated**, ovvero la **gestione degli account**, la **pubblicazione e valutazione delle recensioni** e la **gestione del catalogo dei film**.

In particolare, i test riguarderanno:

- Il processo di registrazione e autenticazione degli utenti.
- La pubblicazione, visualizzazione e valutazione delle recensioni.
- Le funzionalità relative alla gestione del catalogo film da parte dei gestori.

## 2. Relazioni con altri documenti

Per la corretta individuazione dei test case, si farà riferimento ai seguenti documenti:

- Requirements Analysis Document (RAD): I test copriranno i seguenti Use Case:
  - UC01 – Login
  - UC02 – SignIn
  - UC05 – Visualizzazione Catalogo
  - UC08 – Pubblicazione di una Recensione
  - UC09 – Valutazione di una Recensione
  - UC15 – Aggiunta di un Film al Catalogo
- System Design Document (SDD).
- Object Design Document (ODD).

## 3. Feature da Testare e Non Testare

Le feature oggetto di test includono:

- Gestione Account Personale:
  - Registrazione (SignIn)
  - Login e Logout
  - Modifica profilo utente
- Pubblicazione e Visualizzazione delle Recensioni:
  - Pubblicazione di recensioni
  - Visualizzazione recensioni
  - Valutazione delle recensioni

- Gestione del Catalogo Film:
  - Aggiunta, modifica e rimozione dei film dal catalogo.

Le feature non testate includeranno la gestione avanzata della moderazione e delle segnalazioni in quanto secondarie per il ciclo di test attuale.

#### **4. Pass/Fail Criteria**

L'esito del test sarà determinato da un oracolo che confronta l'output ottenuto con i risultati attesi basati sui requisiti.

- Un test avrà esito positivo (Pass) se il risultato ottenuto coincide con quello atteso.
- Un test fallirà (Fail) se il risultato differisce da quello atteso.

Il testing sarà considerato valido solo se verranno verificati tutti i requisiti funzionali principali. In caso di errori individuati e corretti, verrà effettuato un regression testing per validare nuovamente le funzionalità precedenti.

#### **5. Approccio**

Il processo di testing sarà strutturato in tre fasi: unit testing, integration testing e system testing.

##### **Test di Unità**

- Verrà utilizzato un approccio black-box con la tecnica category partition.
- I test saranno realizzati con JUnit e Mockito per simulare componenti indipendenti (stubs e drivers).

##### **Test di Integrazione**

- Sarà adottata una strategia Sandwich, dove l'application logic layer rappresenterà il target layer.
- Il top layer (interfaccia utente) e il bottom layer (database) saranno testati singolarmente utilizzando stubs e drivers.
- Mockito sarà utilizzato per validare l'integrazione dei livelli.

##### **Test di Sistema**

- Sarà eseguito il functional testing per validare l'intera piattaforma.
- La tecnica utilizzata sarà la category partition adattata ai singoli Use Case (UC01, UC05, UC08, UC15).
- I test case di sistema saranno implementati utilizzando Selenium IDE, che consentirà di simulare le interazioni dell'utente tramite browser.

## 6. Sospensione e Ripristino

Il test sarà eseguito in modo continuo senza sospensioni, concentrandosi su singole funzionalità una alla volta.

## 7. Materiale di Testing

### Hardware

- Un computer con connessione a Internet.

### Software

- JUnit e Mockito per unit e integration testing.
- Selenium IDE per il system testing.
- Un browser compatibile (es. Google Chrome, Mozilla Firefox).
- Database MySQL per testare la persistenza dei dati.