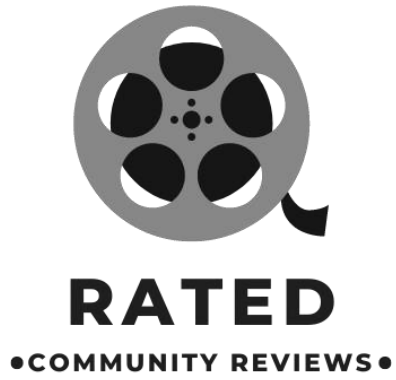
Università degli Studi di Salerno

Corso di Ingegneria del Software

Rated
Test Plan Document
Versione 1.1



Data: 12/01/2025

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Francesco Rao	0512116836
Bruno Nesticò	0512117268

Scritto da:	Francesco Rao, Bruno Nesticò

Revision History

Data	Versione	Descrizione	Autore
24/11/2024	1.0	Prima stesura completa	Francesco Rao, Bruno Nesticò
12/01/2025	1.1	Modifiche alla pianificazione ed esecuzione del testing, in accordo con le tempistiche e il codice realizzato.	Francesco Rao, Bruno Nesticò

Indice

1. INTRODUZIONE

Il seguente documento **Test Plan** definisce strategie, approcci e risorse necessari per eseguire il testing del sistema **Rated**, un'applicazione dedicata alla gestione e condivisione di recensioni di film.

L'obiettivo principale di questo documento è garantire che **Rated** soddisfi i requisiti funzionali e non funzionali, identificati nel documento RAD (Requirements Analysis Document) e progettati nel SDD (System Design Document). Il piano di test è finalizzato a individuare e correggere bug o malfunzionamenti attraverso test di **unità**, di **integrazione** e di **sistema**, fornendo un sistema finale stabile che rispetti gli obiettivi previsti in fase di progettazione.

Il sistema Rated è composto da diversi sottosistemi che coprono la gestione dei film, delle recensioni e delle relative funzionalità (come la segnalazione e la moderazione). L'attività di testing verificherà la correttezza delle funzionalità e l'efficienza delle operazioni, seguendo approcci **bottom-up** e **black-box**.

1.1. Riferimenti

- **RAD_Rated** (Requirements Analysis Document): descrive i requisiti funzionali (FR) e non funzionali (NFR) del sistema, identificati in fase di analisi.
- **SDD_Rated** (System Design Document): descrive l'architettura del sistema, i principali sottosistemi e i servizi forniti.
- **ODD_Rated** (Object Design Document): presenta la definizione delle classi, dei metodi e delle interfacce.

2. RELAZIONE CON ALTRI DOCUMENTI

Il presente Test Plan è strettamente correlato con i seguenti documenti di progetto:

- **RAD_Rated**: fornisce la base dei requisiti che saranno oggetto di verifica nelle diverse fasi di test.
- **SDD_Rated**: descrive l'architettura del sistema Rated, evidenziando i vari moduli/sottosistemi, le loro interfacce e il modo in cui essi interagiscono.
- **ODD_Rated**: contiene i dettagli di implementazione a livello di classi e metodi, utili per la definizione dei test di unità e l'organizzazione dei test di integrazione.

3. PANORAMICA DEL SISTEMA

Rated è progettato come sistema di gestione recensioni e valutazioni di film, con funzionalità che vanno dalla registrazione/login degli utenti fino alla moderazione delle recensioni e alla gestione del catalogo dei film.

L'architettura di Rated, delineata nel documento SDD_Rated, segue un modello **three-tier**:

1. **Presentation Layer**: composto da pagine JSP/HTML5/CSS/JavaScript. Gestisce l'interazione con l'utente (ricerca di film, pubblicazione di recensioni, visualizzazione del catalogo, ecc.).
2. **Application Layer**: contiene la logica di business, dove vengono elaborate le richieste degli utenti, gestite le regole di pubblicazione/moderazione delle recensioni e controllati i permessi (es. gestore del catalogo, moderatore, ecc.).
3. **Data Layer**: utilizza un database relazionale (es. MySQL) per memorizzare in modo permanente le informazioni su utenti, film, recensioni e valutazioni. Le operazioni di lettura e scrittura sul DB sono affidate a classi DAO (Data Access Object).

Tutte le interazioni tra questi strati devono essere verificate e validate per garantire che il sistema Rated offra funzionalità complete e affidabili.

4. CARATTERISTICHE DA TESTARE

4.1. Caratteristiche principali

1. **Gestione Film**
    - Visualizzazione del catalogo e relativi dettagli
    - Ricerca e ordinamento dei film
    - Aggiunta, modifica e rimozione di un film (gestore)
2. **Gestione Utente**
    - Registrazione (SignUp) e autenticazione (LogIn, LogOut)
    - Visualizzazione e modifica del profilo personale (compresa la modifica password)
    - Visualizzazione del profilo di altri utenti
3. **Gestione Recensioni**
    - Pubblicazione di nuove recensioni
    - Visualizzazione dettagliata di una recensione
    - Valutazione della recensione (like/dislike)
    - Segnalazione e rimozione di una recensione (moderazione)
    - Accesso all'area di moderazione (approvazione/eliminazione)

Per motivi di priorità e risorse, alcune funzionalità secondarie come ad esempio, alcuni filtri di accesso non critici o funzionalità di front-end non essenziali, potrebbero non essere verificate in maniera pienamente esaustiva nei test di unità. Miglioramenti ai casi di test secondari, potranno essere considerati in successivi cicli di miglioramento del sistema.

5. PASS/FAIL CRITERIA

L'obiettivo del testing è assicurare che Rated rispetti pienamente i requisiti funzionali e non funzionali:

- Un **test è considerato superato** se l'output prodotto dal sistema coincide con l'oracolo (comportamento atteso, definito nei requisiti o specifiche).
- Un **test è considerato fallito** se si riscontrano discrepanze tra il risultato atteso e il risultato effettivo.

In caso di fallimento di un test, è necessario:

1. Identificare e correggere il problema (bug fix).
2. Rieseguire i test unitari e di integrazione interessati (test di regressione).
3. Verificare che le correzioni non abbiano introdotto nuove anomalie.

6. APPROCCIO

Il testing di Rated si svolge in tre fasi principali:

1. **Testing Unitario**
    - Vengono testate le singole classi (DAO, controller, servlet) e i metodi per verificarne la correttezza rispetto alle specifiche dell'ODD_Rated.
    - Si utilizza la metodologia **black-box**: si analizzano input e output senza indagare l'implementazione interna.
    - Si applica la tecnica di **Category Partition** per identificare classi di equivalenza e ridurre i casi di test.
2. **Testing di Integrazione**
    - Verifica le interazioni tra moduli, come ad esempio la cooperazione tra controller/servlet e DAO o tra i moduli di gestione film e gestione recensioni.
    - Si adotta un approccio **bottom-up**, iniziando a integrare e testare i componenti di basso livello (DAO, modelli) per poi passare ai servizi e infine alle servlet e controller.
3. **Testing di Sistema**
    - Esegue una verifica complessiva in un ambiente il più vicino possibile a quello di produzione.

- Vengono simulati gli scenari reali (es. registrazione, login, pubblicazione recensione, moderazione, ecc.) per rilevare eventuali problematiche non emerse nelle fasi precedenti.

7. SOSPENSIONE E RIPRESA

La fase di testing può essere **sospesa** se si verificano condizioni bloccanti, come:

- **Bug critici o bloccanti** che impediscono l'esecuzione dei test pianificati (es. crash del sistema, DB non accessibile).
- **Modifiche non integrate** o non documentate, che rendono instabile l'ultima build del sistema.

Per la **ripresa** delle attività di test:

- Viene effettuata la **correzione dei bug** critici/bloccanti.
- Si eseguono test di regressione per verificare che le modifiche non abbiano introdotto nuovi errori.
- Si aggiornano eventualmente i casi di test e la documentazione, se necessario.

8. TESTING MATERIALS

Per l'esecuzione dei test su Rated, verranno utilizzati i seguenti strumenti:

- **JUnit**
  - Framework principale per i test di unità (sulle classi DAO, servlet, controller).
- **Mockito**
  - Utilizzato per creare **mock** di dipendenze nelle classi in test (es. simulazione accesso DB).
- **Selenium**
  - Impiegato per l'automazione dei test di sistema su browser.
  - Simula l'utente finale (ricerca di film, pubblicazione recensioni, moderazione, ecc.) e permette di verificare la corretta integrazione tra frontend e backend.

Gli **script SQL** di setup del database, contenente dati falsi di testing saranno disponibili all'interno della repository del progetto.

9. TEST CASES

I singoli test case, con dettagli di input e output atteso, sono documentati nel **TestCaseSpecification_Rated**.

10. TESTING SCHEDULE

Il programma dei test per Rated è pianificato in funzione delle milestone di sviluppo e delle priorità di rilascio:

1. **Testing Unitario (Tempo di sviluppo)**
    - Viene svolto in parallelo allo sviluppo. Ogni nuovo modulo o classe viene testato immediatamente.
    - Garantisce che le unità rispettino le specifiche fornite in fase di design.
2. **Testing di Integrazione (Tempo di sviluppo)**
    - Una volta consolidati i moduli di base, si integrano progressivamente le componenti (DAO, servizi, controller, interfacce).
    - Vengono eseguiti test incrociati per confermare la corretta cooperazione tra componenti.
3. **Testing di Sistema (2-3 giorni)**
    - Al termine dell'integrazione, si eseguono test end-to-end, simulando l'utilizzo tipico del sistema (registrazione utente, login, pubblicazione recensioni, moderazione, gestione film).

Al termine, si ripetono i test come **verifica finale** (test di regressione) per garantire che il sistema Rated sia conforme ai requisiti prima del rilascio. Eventuali criticità emerse vengono risolte e verificate nuovamente, assicurando un prodotto privo di bug bloccanti. I risultati del testing verranno illustrati nel dettaglio all'interno del documento TestExecutionReport_Rated, eventuali errori e le loro soluzioni verranno discussi nel documento TestIncidentReport. Nel TestSummaryReport_Rated sarà presente un breve riassunto delle performance che il sistema ha dimostrato durante i test.