|  |
| --- |
| A logo with numbers and letters  Description automatically generated  **Università degli Studi di Salerno Corso di Ingegneria del Software** |

**Rated  
System Design Document  
Versione 1.1**

**A black and white logo

Description automatically generated**

Data: 8/01/2025

**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 |
| 08/01/2025 | 1.1 | Modifiche alle sezioni descrittive del testo e ai diagrammi | Francesco Rao, Bruno Nesticò |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Indice**

1. Introduzione 4

1.1 Scopo del Sistema 4

1.2 Obiettivi di progettazione 4

1.2.1 Criteri di usabilità 4

1.2.2 Criteri di affidabilità 4

1.3 Riferimenti 4

2. Architettura Software Proposta 5

2.1 Decomposizione in Sottosistemi 5

2.2 Mapping Hardware/Software 6

2.3 Gestione dei dati persistenti 7

2.4 Controllo degli accessi e sicurezza 8

2.5 Controllo Software 9

2.6 Condizioni di confine 10

3. Servizi dei Sottosistemi 11

**1. Introduzione**

**1.1. Scopo del sistema**

Lo scopo del sistema è permettere la gestione di una piattaforma web dedicata agli appassionati di cinema. Si intende gestire le recensioni dei film, le valutazioni dei contenuti pubblicati dagli utenti e il sistema reputazionale che favorisce la visibilità dei recensori più attivi e apprezzati. Queste operazioni verranno eseguite rispettivamente dai gestori del catalogo, responsabili dell’aggiornamento dell’offerta di film, e dai moderatori, incaricati di garantire il rispetto delle linee guida e dei contenuti pubblicati. Inoltre, gli utenti avranno la possibilità di registrarsi, accedere, pubblicare recensioni, valutare i contributi della community e interagire con altri membri.

**1.2. Obiettivi di progettazione**

**1.2.1 Criteri di usabilità**

Per garantire un’esperienza utente ottimale, la piattaforma adotterà i seguenti criteri di usabilità:

***NFR1 Validazione degli input:*** saranno implementati meccanismi per prevenire errori durante l’inserimento di dati. Messaggi di errore chiari guideranno l’utente nella correzione dei valori errati.

***NFR2 Design responsive*:** l’interfaccia sarà ottimizzata per adattarsi a diversi dispositivi (PC, tablet, smartphone), rendendo l’esperienza uniforme e accessibile.

***NFR3 Navigazione intuitiva*:** ogni pagina presenterà una barra di navigazione per facilitare l’accesso rapido alle diverse sezioni.

**1.2.2 Criteri di affidabilità**

Il sistema garantirà robustezza e protezione attraverso:

***NFR4 Controllo avanzato degli input:*** oltre alla validazione primaria, saranno effettuati ulteriori controlli per gestire scenari non previsti e prevenire errori critici.

***NFR5 Sicurezza dei dati:*** saranno adottati protocolli di crittografia per proteggere le informazioni sensibili, come le credenziali degli utenti. Le password dovranno rispettare requisiti di complessità (es. lunghezza minima e inclusione di caratteri speciali).

**1.2.3 Criteri di prestazioni**

***NFR6 Prestazioni:*** Il sistema deve garantire tempi di caricamento inferiori a 2 secondi per ogni pagina o funzione principale.

**1.2.4 Criteri di supportabilità**

***NFR7 Supportabilità:*** Il codice deve essere strutturato in modo da permettere futuri aggiornamenti senza compromettere l’integrità del sistema.

**1.3 Riferimenti**

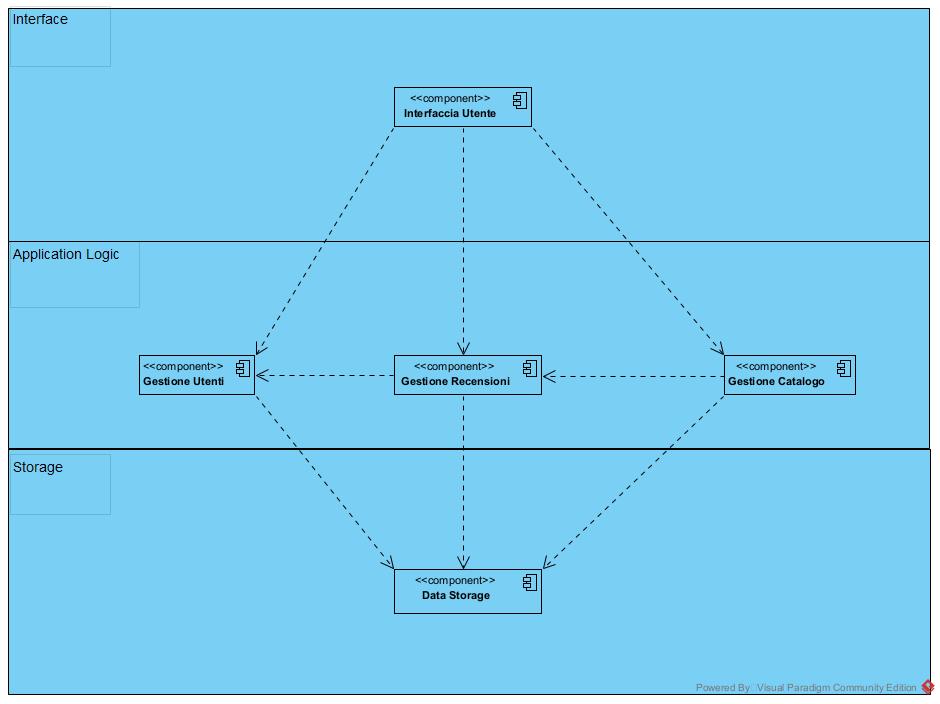
RequirementAnalysisDocument\_Rated

**2. Architettura Software Proposta**

**2.1 Decomposizine in Sottosistemi**

l sistema prodotto sarà caratterizzato da una architettura a tre livelli: Livello Interface, Livello Application Logic e Livello Storage.

*Component Diagram*



**Interface**

L’Interfaccia Utente è il sottosistema responsabile dell’interazione con l’utente finale. Fornisce le classi e le operazioni necessarie per acquisire input, visualizzare output e interagire con i servizi della piattaforma. Si occupa inoltre di:

*Coordinare la logica applicativa,* delegando le operazioni principali ai sottosistemi del livello sottostante.

*Validare i dati in ingresso*, assicurando la conformità alle specifiche del sistema e restituendo eventuali messaggi di errore in caso di input errati.

**Application Logic**

La Logica Applicativa è suddivisa in tre sottosistemi principali:

*Gestione Utenti:* si occupa dell’autenticazione, della creazione e dell’eliminazione degli account. Garantisce la protezione delle credenziali attraverso l’adozione di protocolli sicuri per la gestione dei dati sensibili.

*Gestione Catalogo:* consente ai Gestori di inserire, aggiornare e rimuovere film dal catalogo, mantenendo un’offerta costantemente aggiornata e coerente con le preferenze della community.

*Gestione Recensioni:* supporta l’aggiunta, la valutazione e la moderazione delle recensioni. Questo include meccanismi per segnalare contenuti inappropriati e l’assegnazione di punteggi che influenzano il sistema reputazionale.

**Storage**  
Il Data Storage rappresenta il sottosistema responsabile della memorizzazione e gestione degli oggetti persistenti, come:

*Dati utente:* informazioni personali, credenziali e preferenze.

*Dati di catalogo:* elenco dei film disponibili con relative informazioni.

*Dati delle recensioni:* contenuti testuali, valutazioni e metadati associati.

La persistenza dei dati è garantita attraverso l’utilizzo di un database relazionale ottimizzato per supportare operazioni frequenti e simultanee.

Nel progettare il sistema, abbiamo deciso di adottare un'architettura semi-chiusa con l'obiettivo principale di mantenere un'elevata coesione all'interno dei nostri moduli. 1. Gestione di Operazioni Multi-Contesto: Alcune funzionalità del nostro sistema, come ad esempio la rimozione di una recensione che viola le linee guida e il warn dell’utente, operano su più ambiti contemporaneamente. Questa funzione, per sua natura, potrebbe appartenere sia al modulo di gestione utenti sia a quello di gestione recensioni. Un'architettura semi-chiusa consente di gestire tali operazioni in modo efficace senza compromettere la coesione di ciascun modulo. 2. Separazione delle Responsabilità: Adottando un'architettura semi-chiusa, riusciamo a separare chiaramente la logica delle funzioni che interagiscono con diversi domini. Questo approccio permette di isolare le responsabilità, riducendo le dipendenze incrociate tra i moduli e facilitando l'individuazione e la risoluzione di eventuali problemi

Il sistema adotta un’**architettura semi-chiusa** con l'obiettivo principale di mantenere un'elevata coesione all'interno dei nostri moduli.

1. *Gestione di Operazioni Multi-Contesto*: Alcune funzionalità del nostro sistema, come ad esempio la rimozione di una recensione che viola le linee guida e il warn dell’utente, operano su più ambiti contemporaneamente. Questa funzione, per sua natura, potrebbe appartenere sia al modulo di gestione utenti sia a quello di gestione recensioni. Un'architettura semi-chiusa consente di gestire tali operazioni in modo efficace senza compromettere la coesione di ciascun modulo.

2. *Separazione delle Responsabilità*: Adottando un'architettura semi-chiusa, riusciamo a separare chiaramente la logica delle funzioni che interagiscono con diversi domini. Questo approccio permette di isolare le responsabilità, riducendo le dipendenze incrociate tra i moduli e facilitando l'individuazione e la risoluzione di eventuali problemi.

**2.2 Mapping Hardware/Software**

Il sistema *Rated* è, per sua natura, un sistema distribuito, poiché gli utenti interagiscono con esso da macchine diverse, in remoto. Si distinguono tre tipi principali di componenti:

* Nodo Utente, che fornisce l’interfaccia utente. Questo nodo esegue il sottosistema Interface tramite un web browser.
* Application Server, che gestisce la logica applicativa e i controlli. Esegue il sottosistema Application Logic.
* Database Server, che gestisce la persistenza dei dati e i servizi offerti. Esegue il sottosistema Storage.

Il sistema sarà implementato utilizzando Apache Tomcat 9 in combinazione con Java JDK 17.0. Tomcat è un contenitore di Servlet che consente il deployment e l’esecuzione di applicazioni web. La logica applicativa sarà scritta in Java per sfruttare la sua modularità e robustezza.

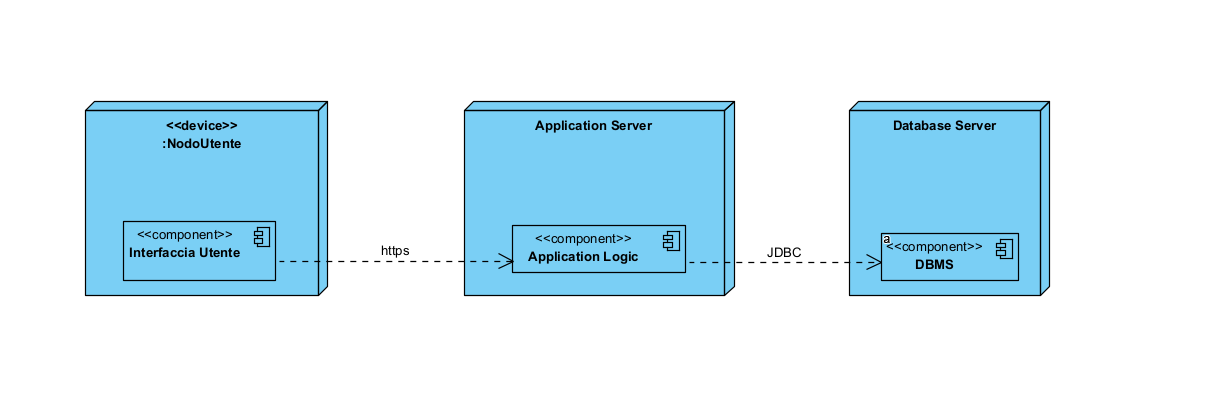
*Tecnologie utilizzate*:

* HTML5, CSS: strumenti utilizzati per la creazione di interfacce utente moderne e responsive, che garantiscono una fruibilità ottimale su dispositivi diversi.
* JavaScript: utilizzato per fornire interattività e validazione dinamica dei dati inseriti dagli utenti.
* JDBC: utilizzato per gestire la comunicazione tra il backend e il database relazionale.
* AJAX: implementato per richieste asincrone, migliorando l’esperienza utente con aggiornamenti dinamici senza ricaricare la pagina.

In particolare, per implementare i sottosistemi eseguiti sul Nodo Server, verranno utilizzate le Servlet Java per gestire le richieste HTTP inviate dal browser e rispondere dinamicamente.

*Deployment Diagram*

Il componente Application Logic racchiude i sottosistemi relativi al layer logico. Il componente Storage, invece, si occupa della gestione dei dati persistenti attraverso MySQL, mentre il componente Interface è responsabile della presentazione e interazione utente.



**2.3 Gestione dei dati persistenti**

*Identificazione dei dati persistenti*

Il sistema gestisce il seguente insieme di oggetti da memorizzare in maniera persistente:

* Utente: informazioni personali, credenziali e preferenze.
* Film: dettagli sui film disponibili nel catalogo (titolo, regista, descrizione, ecc.).
* Recensione: contenuti testuali, votazioni e metadati relativi agli utenti e ai film.
* Valutazione: like/dislike fatto da un’utente verso una recensione
* Report: segnalazione di un’utente verso una recensione

Per memorizzare questi oggetti, verrà utilizzato un database relazionale, MySQL, per le seguenti ragioni:

* Supporto per query concorrenti.
* Meccanismi di transazione per garantire l’integrità e la consistenza dei dati (proprietà ACID).
* Capacità di eseguire backup e ripristino in modo efficiente.

*Tecnologie utilizzate*

* MySQL 8.0: DBMS relazionale per la gestione dei dati persistenti.
* JDBC: utilizzato per interfacciarsi tra il backend e il database in modo sicuro ed efficiente.

La logica applicativa interagirà con il database mediante oggetti Java, utilizzando query SQL ottimizzate e verifiche di consistenza per garantire la robustezza del sistema.

**2.4 Controllo degli accessi e sicurezza**

La piattaforma Rated più essere utlizzata da diversi tipi di utenti, ognuno con permessi e privilegi differenti.

*Matrice di controllo degli accessi*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Registro\_Utenti** | **Moderation\_Service** | **Catalogo** | **Catalogo\_Recensioni** |
| **Utente Guest** | SingUp() |  | GetFilms() GetFilm() RicercaFilm() | FindRecensioni() GetRecensioni() |
| **Utente Recensore** | LogIn() LogOut() |  | GetFilms() GetFilm() RicercaFilm() | RimuoviRecensione() ValutaRecensione() GetValutazioni() FindRecensioni() GetRecensioni() |
| **Moderatore** | LogIn() LogOut() | Warn() | GetFilms() GetFilm() RicercaFilm() | GetAllRecensioniSegnalate() ApprovaRecensione() RimuoviRecensione() FindRecensioni() GetRecensioni() |
| **Gestore Del Catalogo** | LogIn() LogOut() |  | AggiungiFIlm() RimuoviFilm() ModificaFilm() GetFilms() GetFilm() RicercaFilm() | FindRecensioni() GetRecensioni() |

Vincoli non esprimibili tramite la tabella: utente\_recensore può usare RimuoviRecensione() solo sulle sue recensioni.

**2.5. Controllo Software**

Osservando il **dynamic model** prodotto in fase di analisi dei requisiti, è stato scelto un tipo di controllo centralizzato. Inquanto il controllo nei diagrammi di sequenza è sempre affidato ad un’oggetto “centrale”.

In particolare, **Rated** adotta un flusso di controllo **event-driven**, integrando il modello **three-tier** per garantire modularità, separazione delle responsabilità e una migliore gestione delle risorse.

**Funzionamento generale**

Il sistema Rated segue un flusso di controllo centralizzato strutturato su tre livelli logici principali, come descritto di seguito:

1. **Presentazione (Presentation Tier)**:
   * Questo livello è rappresentato dal browser dell’utente e dal Web Server, che gestisce l’interfaccia utente e la ricezione delle richieste HTTP.
   * Il Web Server riceve le richieste dal client e le inoltra al livello di logica applicativa per l'elaborazione.
2. **Logica applicativa (Application Tier)**:
   * Include la gestione delle regole aziendali e della logica applicativa.
   * Una volta ricevuta una richiesta dal Web Server, il sistema elabora i dati e coordina le interazioni con il livello di accesso ai dati.
   * Per ogni richiesta, il Web Server avvia un thread dedicato, consentendo la gestione concorrente di più richieste senza compromettere le prestazioni.
3. **Accesso ai dati (Data Tier)**:
   * Questo livello si occupa del recupero, della scrittura e della manipolazione dei dati.
   * La logica applicativa interagisce con il livello dei dati attraverso un'interfaccia standard, garantendo un accesso sicuro e ottimizzato al database o ad altre risorse di archiviazione persistente.

**Flusso di controllo centralizzato**

1. **Ricezione della richiesta**: Il Web Server attende richieste dal browser dell’utente e le inoltra alla logica applicativa.
2. **Elaborazione della richiesta**: Il sistema interpreta la richiesta, esegue la logica applicativa e interagisce con il livello di accesso ai dati per recuperare o aggiornare le informazioni necessarie.
3. **Generazione della risposta**: Dopo aver elaborato la richiesta, i risultati vengono restituiti al Web Server.
4. **Risposta al client**: Il Web Server invia la risposta al browser dell’utente, completando il ciclo.

Grazie all'architettura **three-tier**, il sistema Rated garantisce una chiara separazione tra la presentazione, la logica applicativa e l’accesso ai dati, migliorando la manutenibilità, la scalabilità e la sicurezza complessiva del sistema.

**2.6 Condizioni di Confine**

Il ruolo dell'amministratore nel sistema Rated si concentra sulla gestione e manutenzione del server. Di seguito sono descritte le principali condizioni limite relative alle fasi di avvio, spegnimento e gestione dei fallimenti.

**Fase di Avvio**

Durante l’avvio, il server del sistema Rated esegue una serie di operazioni per preparare l’ambiente di esecuzione. Questa fase include:

* Avvio del server Apache Tomcat, che carica i file di configurazione necessari per il deploy dell’applicazione.
* Connessione al database MySQL, utilizzando il driver JDBC per garantire l’accesso ai dati essenziali (utenti, prenotazioni, programmazione dei film, sale, posti).
* Avvio della JVM, che legge i file di configurazione e carica i moduli applicativi per la logica di business.

Al termine di questa sequenza, il server è operativo e pronto a gestire le richieste degli utenti. L’intera fase di avvio è gestita dal web container.

**Fase di Spegnimento**

Lo spegnimento del server viene eseguito per manutenzioni, aggiornamenti o interruzioni del servizio. Questa fase prevede:

* Chiusura delle connessioni attive e completamento o annullamento corretto delle transazioni in corso, garantendo l’integrità dei dati.
* Arresto della JVM.
* Disconnessione dal database MySQL.

Completate queste operazioni, il server può essere spento in sicurezza. Anche questa fase è completamente gestita dal web container.

**Fase di Fallimento**

Eventuali guasti hardware, blackout o crash software possono interrompere il funzionamento del sistema. In queste situazioni, l’amministratore deve intervenire prontamente per il ripristino. Le principali attività includono:

* Analisi dei log di sistema per diagnosticare le cause del fallimento.
* Esecuzione delle procedure necessarie per ripristinare il servizio.

L’amministratore è responsabile di monitorare e mitigare tali eventi per garantire la continuità del servizio e minimizzare i tempi di inattività.

**3 Servizi dei Sottosistemi**

Considerando i sottosistermi del livello Application Logic: Gestione Utenti, Gestione Film e Gestione Recensioni, si elencano i servizi che offrono.

*Gestione Utenti*

* Servizio di **Autenticazione**, per la registrazione dell’utente e per la verifica delle credenziali di accesso a tempo di Login.  
  Operazioni: login(), logout(), register()
* Servizio di **Moderazione**, per gestire gli utenti che violano le linee guida.  
  Operazioni: warn()
* Servizio di **Gestione del Profilo**, per permettere all’utente di poter aggiornare il proprio profilo.  
  Operazioni: ProfileUpdate(), PasswordUpdate(), findByUsername(), getUtenti

*Gestione Catalogo*

* Servizio di **Gestione del Catalogo** per l’aggiunta, modifica e rimozione dei film.  
  Operazioni: getFilms(), aggiungiFilm(), rimuoviFilm(), ricercaFilm(), getFilm(), getFilms(), addFilm(), modifyFilm(), removeFilm()

*Gestione Recensioni*

* Servizio di **Gestione dele recensioni**, per permettere di gestione di recensioni, valutazioni e report.

