Table of Contents

---

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to provide a comprehensive overview of the requirements for Cinevea—a web application offering personalized movie recommendations and review aggregation. The platform integrates a chatbot to enhance user interaction and decision-making, consolidating data from platforms like IMDb and Rotten

Tomatoes. The system serves as a one-stop solution for movie enthusiasts seeking reliable reviews, personalized suggestions, and user-friendly interactions.

1.2 Document Conventions

Standard typographical conventions are used throughout this document:

- **Bold Text** for section headings.
- *Italicized Text* for emphasis or important points.
- Unique identifiers such as REQ-1 for requirements for easy traceability.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- **Developers:** For understanding the functional and non-functional requirements.
- **Project Managers:** To track project scope and deliverables.
- **QA Testers:** To ensure system requirements are met during testing phases.
- **Stakeholders:** To validate business and user expectations.

Reading Suggestions: Start with the Overview (Section 2) for context, then proceed to System Features (Section 3) for specific functionality.

1.4 Project Scope

Cineveamvie aims to provide a comprehensive platform for:

- Personalized movie recommendations based on user mood, genre preferences, and history.
- Aggregating and displaying reviews from trusted platforms such as IMDb, Rotten Tomatoes, and Metacritic.
- Enhancing user engagement through a responsive chatbot.

The project aligns with business goals of increasing user retention and engagement by 20% within the first six months post-launch.

1.5 References

- IMDb API Documentation
- Rotten Tomatoes API Documentation
- Natural Language Processing (NLP) Libraries
- Cloud Hosting and Scalability Guides

---

2. Overall Description

2.1 Product Perspective

Cinevea is a standalone web application with mobile compatibility. It integrates with external review platforms via APIs and utilizes AI for natural language processing to provide an interactive user experience.

Context:

- A new product in the movie recommendation domain.
- Competes with existing platforms by offering a unified experience of recommendations, reviews, and chat-based interaction.
- Incorporates user-generated data for continuous improvement of recommendations.

Diagram: A context diagram depicting integrations with external platforms and user interactions will be included in Appendix B.

2.2 Product Features

- **Personalized Recommendations:** AI-driven suggestions based on user input.
- **Aggregated Reviews:** Displaying consolidated reviews from multiple platforms.
- **Chatbot:** Seamless interaction for recommendations and queries.
- **User Contributions:** Ability to rate movies and create collaborative lists.
- **Insights and Trends:** Visual stats on user's viewing habits and trending themes.

2.3 User Classes and Characteristics

1. **End-Users:**
    - Characteristics: Movie enthusiasts with varied technical expertise.
    - Needs: Simple, intuitive interface and personalized features.
2. **Content Providers:**
    - Platforms supplying review and rating data.
    - Needs: Clear integration guidelines and data usage compliance.
3. **Developers:**
    - Characteristics: Technical team implementing and maintaining the system.
    - Needs: Comprehensive documentation and modular code structure.

2.4 Operating Environment

- **Web:** Supported on Chrome, Firefox, Safari, and Edge (latest versions).
- **Mobile:** iOS (13+) and Android (8.0+).
- **Backend:** Cloud-based hosting with scalable architecture.

2.5 Design and Implementation Constraints

- Adherence to third-party API usage terms.
- Compliance with GDPR and other data privacy regulations.
- Chatbot responses within 2 seconds.

2.6 User Documentation

- User manuals.
- Tutorials for chatbot and advanced features.
- FAQs addressing common queries.

2.7 Assumptions and Dependencies

- APIs from IMDb, Rotten Tomatoes, and Metacritic remain accessible.
- Stable internet connection for real-time data retrieval.

---

3. System Features

3.1 Feature 1: Movie Recommendation

3.1.1 Description and Priority

- **High Priority:** Provides personalized suggestions based on user mood, genre, and watch history.

3.1.2 Stimulus/Response Sequences

- **Stimulus:** User specifies mood or genre preferences.
- **Response:** System suggests movies using collaborative filtering and content-based algorithms.

3.1.3 Functional Requirements

- **REQ-1:** Implement AI algorithms for personalized recommendations.
- **REQ-2:** Display trending and top-rated movies.
- **REQ-3:** Allow users to provide feedback on recommendations.

3.2 Feature 2: Review Aggregation

3.2.1 Description and Priority

- **High Priority:** Displays consolidated reviews and ratings from trusted platforms.

3.2.2 Stimulus/Response Sequences

- **Stimulus:** User searches for a movie.
- **Response:** System fetches and displays reviews from IMDb, Rotten Tomatoes, and Metacritic.

3.2.3 Functional Requirements

- **REQ-4:** Integrate APIs for review aggregation.
- **REQ-5:** Present user-friendly summaries of reviews.

3.3 Feature 3: Chatbot Integration

3.3.1 Description and Priority

- **Medium Priority:** Ensures engaging and natural language-based interaction.

3.3.2 Stimulus/Response Sequences

- **Stimulus:** User queries the chatbot.
- **Response:** Chatbot responds with personalized recommendations or movie details.

3.3.3 Functional Requirements

- **REQ-6:** Implement NLP for chatbot queries.
- **REQ-7:** Handle up to 1,000 concurrent users.

3.4 Feature 4: User Ratings and Reviews

3.4.1 Description and Priority

- **High Priority:** Allows users to post ratings and reviews for movies.

3.4.2 Stimulus/Response Sequences

- **Stimulus:** User submits a rating or writes a review for a movie.
- **Response:** System stores and displays the rating/review on the movie's page.

3.4.3 Functional Requirements

- **REQ-8:** Enable users to submit ratings on a scale (e.g., 1-5 stars).
- **REQ-9:** Allow users to write and edit reviews.
- **REQ-10:** Implement a moderation system for reviews.

3.5 Feature 5: Watchlists and Collaborative Lists

3.5.1 Description and Priority

- **Medium Priority:** Provides functionality for users to create and share watchlists.

3.5.2 Stimulus/Response Sequences

- **Stimulus:** User adds movies to a watchlist or creates a shared list with friends.
- **Response:** System stores and retrieves the watchlist or collaborative list.

3.5.3 Functional Requirements

- **REQ-11:** Allow users to create and manage personal watchlists.
- **REQ-12:** Enable users to invite friends to collaborate on shared lists.
- **REQ-13:** Provide privacy settings for lists (public, private, friends-only).

---

4. External Interface Requirements

4.1 User Interfaces

- Intuitive design with advanced search and filtering options.
- Mobile-friendly interface.
- Dark and light themes for user comfort.

4.2 Hardware Interfaces

- Support for standard desktop and mobile devices.

4.3 Software Interfaces

- API integration with IMDb, Rotten Tomatoes, and Metacritic.
- Use of cloud services for storage and processing.

4.4 Communications Interfaces

- HTTPS for secure data transmission.
- WebSocket for real-time chatbot communication.

---

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Response time: < 2 seconds.
- Support: 1,000 concurrent users.

5.2 Safety Requirements

- Minimize risks of data breaches by using secure protocols.
- Regularly update software to mitigate vulnerabilities.

5.3 Security Requirements

- Encrypt all sensitive user data.
- Implement multi-factor authentication for account access.

5.4 Software Quality Attributes

- **Usability:** Intuitive interfaces with minimal learning curve.
- **Reliability:** Ensure 99.9% uptime.
- **Scalability:** Easily support increased user load.
- **Maintainability:** Modular design for easy updates.

Appendices

A. Glossary

- **NLP:** Natural Language Processing.
- **API:** Application Programming Interface.

B. Analysis Models

- ER diagram, use-case diagram, kano diagram, class diagram and state diagram.

C. Issues List

- Tracking unresolved points and pending decisions.