Software Requirements Specification

for

Online database of information related to films

Version 1.0 approved

Prepared by Zheleznova Valeriia

Institute for Applied System Analysis

October 6, 2021

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Zheleznova Valeriia | October 6, 2021 | Initial revision | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

The aim of this project is to create an online database for film information, including film descriptions, trailers, ratings and comments. This document describes the basic requirements for this system.

## Document Conventions

There are no special designations in this document.

## Intended Audience and Reading Suggestions

This project is an online database of information related to films. This project is done as a learning project. The main audience of the system is film fans and film critics.

## Product Scope

The main purpose of the system is to make it as easy as possible for any user to find information about the film they would like to see and to see the rating of the films by different genres. For special film lovers, there is an opportunity to write a review about the film. We want to create a database of thousands or millions of films, information about them, ratings and reviews by those who have seen them.

## References

https://www.imdb.com/

# Overall Description

## Product Perspective

The system created will include features such as:

* **Viewing information** about the film, namely its description, including writer, actors, director, year, budget, etc., rating, trailer
* **Viewing film ratings** based on genres
* **Write users own review** of the movie and rate it.

## Product Functions

The main functions of the system are represented as Use Case UML diagram:

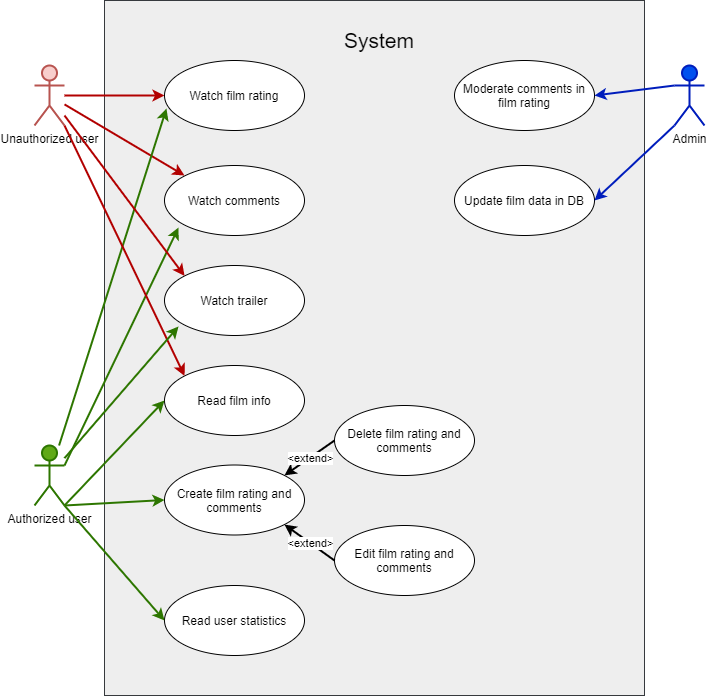


Figure 2.1 Use Case UML diagram

## User Classes and Characteristics

The user should be able to view full details of the film, write their own review and view the rating of the films by genre.

The system will support 3 types of users: anonymous user, authorized user, and administrator.

Functions of anonymous user:

* Read film information
* Read film reviews
* Watch film rating
* Watch rating of films based on genre

Functions of authorized user:

* Read film information
* Read film reviews
* Watch film rating
* Watch rating of films based on genre
* Rate the film
* Write film review
* Watch user statistics in personal profile

Functions of administrator:

* Add film and its information to database
* Delete film and its information from database
* Change film information in database
* Moderate reviews

## Operating Environment

* Environment: Amazon Web Services (AWS)
* Database: MongoDB
* Client/Server system
* Client side: PWA build with ReactJS
* Server side: microservice build with NodeJS

## Design and Implementation Constraints

The service should be as resilient and automatically scalable as possible, to ensure smooth operation of the service. The service should process requests quickly, with a minimum of 99 percent of requests in 250 milliseconds. When analyzing the service on Google Lighthouse, the minimum score should be 95.

## User Documentation

The system should have its own help system, contained on a separate page with frequently asked questions and an opportunity to write to support.

## Assumptions and Dependencies

The service presented has a direct dependency on AWS, such as an AWS crash, which could disrupt the smooth operation of the system.

# External Interface Requirements

## User Interfaces

For this system, you need to use a perfectly engineered design for a comfortable user experience.

## Hardware Interfaces

As we use ReactJS, we need to be able to run the server and client side on existing platforms.

## Software Interfaces

Integration of viewing the trailer directly from YouTube, without redirecting to the service (but for the user's convenience, such a function is also provided). Also integrated is authorization via Google account, Twitter account, Facebook account.

## Communications Interfaces

The service should communicate with users by sending them emails, which will contain both authorization information and information on innovations. Communication with support will also take place via email.

# System Features

## Movie Database

### Description and Priority

Database that will contain information about films. Priority: High

## Rating system

### Description and Priority

User must be able to watch the rating of films related on genre. Priority: High

## Review system

### Description and Priority

User must be able to watch and write reviews on film. Priority: High

# Other Nonfunctional Requirements

## Performance Requirements

The system must compete in terms of speed with the other major players in the market. The maximum allowable download speed of the service on the user's device is 2 seconds, no matter the user's device.

## Safety Requirements

There are no special requirements for data integrity.

## Security Requirements

User data should be protected as much as possible from being obtained and processed by unauthorized persons, which will be ensured by sophisticated data encryption.

## Software Quality Attributes

The system should be broken down into modules that can be reused.

## Business Rules

No special business requirements.