

Cover Page

Project Name:

IMDb: Ratings and Reviews for Movies &TV Shows

Assigned TA Name:

Akram Tarek

Team ID:

12

Team Members:

Name	ID	Department
اسماعيل محمد اسماعيل معاذ	2022170064	CS
رنا احمد محمد عاطف	2022170155	CS
شهد محمد على زينهم	2022170210	CS
اشرف خالد جبر	2022170066	CS
عبد العليم ايهاب عبد الفتاح	2022170245	CS
اسلام احمد عادل محمد	2022170057	SC
باسل محمد عبد الفتاح	2022170089	CS

(12) – IMDb: Ratings and Reviews for Movies & TV Shows

Supervisor TA: Akram Tarek

Group Members:

- اسماعيل محمد اسماعيل معاذ

- اسلام احمد عادل محمد

- عبد العليم ايهاب عبد الفتاح

- اشرف خالد جبر

- باسل محمد عبدالفتاح

- شهد محمد على زينهم

- رنا احمد محمد عاطف

25/03/2025

1. Introduction

The IMDB Movie Ratings System allows users to browse movies, rate them, and write reviews. The system provides a structured database of movies and TV shows while enabling users to engage through ratings and critiques.

2. User Requirements

- As a User, I want to search for movies so that I can find information easily.
- As a User, I want to rate movies so that I can share my opinion.
- As a User, I want to read reviews to make informed decisions.
- As an Admin, I want to moderate reviews to remove inappropriate content.
- As an Admin, I want to manage movies to keep the database updated.

3. Functional Requirements

1. User Rating Submission

- **Description/Action:** Users should be able to rate a movie on a scale of 1-10.

- **Requirements/Inputs:** User rating input.
- **Source:** User submission form.
- **Pre-condition:** User must be logged in.
- **Post-condition:** Rating is recorded in the database.
- **Output:** Updated movie rating is displayed.

2. Review Posting

- **Description/Action:** Users should be able to write and submit reviews.
- **Requirements/Inputs:** Review text input.
- **Source:** User review submission form.
- **Pre-condition:** User must be logged in.
- **Post-condition:** Review is added to the database.
- **Output:** Review is displayed under the movie.

3. Movie Search

- **Description/Action:** Users should be able to search for movies.
- **Requirements/Inputs:** Movie title, genre, or actor name.
- **Source:** Search bar.
- **Pre-condition:** User is on the search page.
- **Post-condition:** Search results are retrieved from the database.
- **Output:** List of matching movies is displayed.

4. Review Moderation

- **Description/Action:** Admin can remove inappropriate reviews.
- **Requirements/Inputs:** Admin review flag.
- **Source:** Admin dashboard.
- **Pre-condition:** Admin is logged in.
- **Post-condition:** Review is removed from public view.
- **Output:** Success message confirming review deletion.

5. Watchlist Management

- **Description/Action:** Users can add or remove movies from their watchlist.
- **Requirements/Inputs:** Movie selection.
- **Source:** Movie details page.
- **Pre-condition:** User must be logged in.
- **Post-condition:** Watchlist is updated in the database.
- **Output:** Updated watchlist is displayed

6. Report Reviews

- **Description/Action:** Users can report inappropriate reviews.
- **Requirements/Inputs:** A posted review.
- **Source:** Movie details page.
- **Pre-condition:** User must be logged in.
- **Post-condition:** Review is flagged in the system for moderation.
- **Output:** A message box confirming that the report was submitted successfully.

7. Moderate Reviews

- **Description/Action:** Admin can remove inappropriate or reported reviews.
- **Requirements/Inputs:** A flagged review.
- **Source:** Admin Dashboard.
- **Pre-condition:** Admin must be logged in.
- **Post-condition:** The review is removed from public view.
- **Output:** A message box confirming the review got deleted.

8. Manage Movies

- **Description/Action:** Admin can add, update, or remove movies from the system.
- **Requirements/Inputs:** Movie details (title, genre, release date, actors).
- **Source:** Admin dashboard.
- **Pre-condition:** Admin must be logged in.
- **Post-condition:** Movie database is updated.
- **Output:** Confirmation message indicating the movie was added, updated, or deleted successfully.

9. Manage Users

- **Description/Action:** Admin can ban or unban users violating policies.

- **Requirements/Inputs:** User account details.
- **Source:** Admin Dashboard.
- **Pre-condition:** Admin must be logged in.
- **Post-condition:** User is either banned or unbanned.
- **Output:** A message box showing the status of the user.

4. Non-Functional Requirements

1. Performance

- The system should return search results within 2 seconds.

2. Scalability

- The database should support up to a million users.

3. Security

- User authentication must encrypt sensitive data.

4. Availability

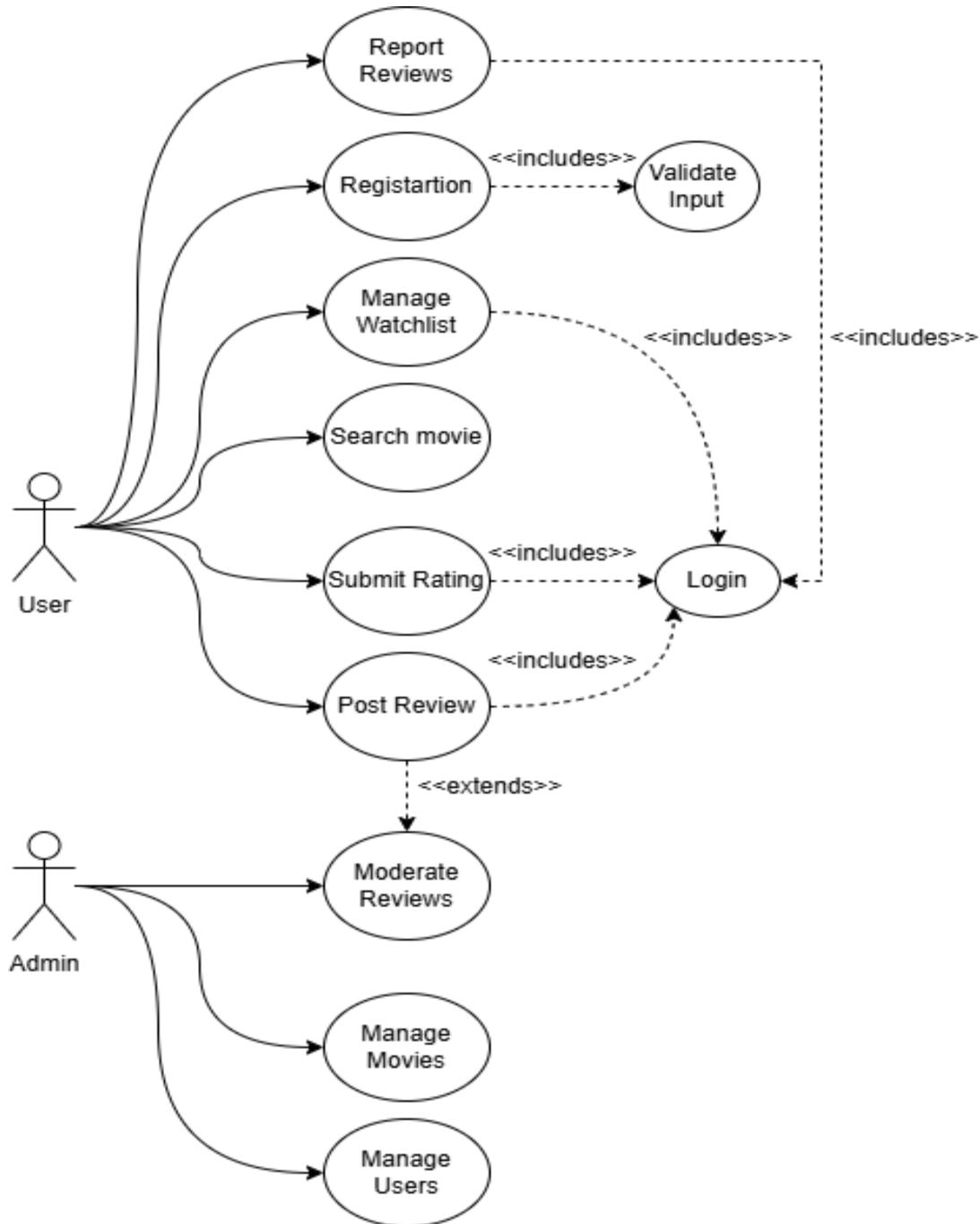
- Ensure 99.9% uptime by implementing robust error handling and automatic recovery mechanisms.

5. Usability

- The UI should be intuitive and responsive.

Use case Diagram and Sequence Diagram

Use Case Diagram:



Sequence Diagram:

