

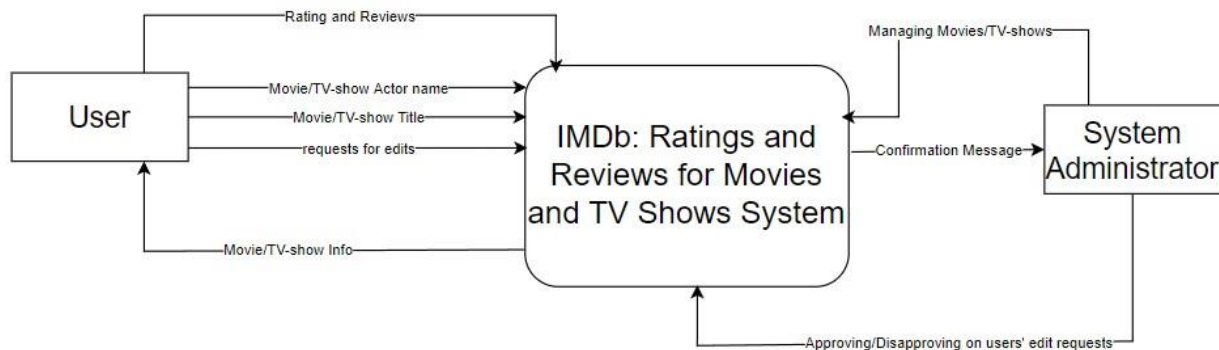
Software Requirements Specification for

IMDb: Ratings and Reviews for Movies and TV Shows

Introduction:

- An Online website that allows users to search for movies and TV Shows. The website displays all information regarding the movie/show including cast members and rewards. The website allows users to leave a rating to movies/shows and write down their reviews.

Context Diagram:



User Requirements:

- The System should be accessible for only registered users
- All users should be able to search for movies/TV-shows, view all ratings and surveys
- Movie/TV-Show page should have the movie poster, release date, all rewards, cast members pictures and names and their role in the movie/tv-show
- Users can suggest edits to existing entries. Registered users with a proven track record can add and make corrections to cast lists, credits, the addition and removal of images. Alteration of titles, cast and crew names, character names, and plot summaries are subject to an Approval Process.
- Registered users can see their approved and unapproved edits.
- Registered users can choose their username, the username should be unique
- Registered users can add reviews/surveys to a movie/tv-show
- Registered users have their own watchlist
- Registered users rate titles on a scale of 1 to 10
- Registered user profile pages show a user's registration date and, optionally their personal ratings.
- Only the Admin can add/delete/edit any data on the website.

Functional Requirements:

1. Search Movie/TV-Show:

- **Description:** Allows user to search for a movie/show using movie title or actor's name.
- **Input:** Movie name /Actor's name.
- **Source:** Users input in the search bar
- **Action:** Take the users input and searches in the database for matching and similar records or words.
- **Pre-conditions:** Is a valid movie/show name/Actor name/Category and the user should be a registered one.
- **Post-conditions:** Search results are loaded successfully without any errors
- **Output:** Display filtered movies/TV shows

2. View movie/TV show (Main page)

- **Description:** Displays all movies and TV shows available in the database
- **Input:** Not required
- **Source:** Not required
- **Action:** Goes through all database records and display them.
- **Pre-conditions:** User should be registered and must be in the main page
- **Post-conditions:** All records are displayed successfully without any errors or missing records
- **Output:** Movies/TV Shows/Top rated movies

3. Register

- **Description:** Allows guest to register their account using a unique username and password
- **Input:** User email & password & username
- **Source:** User
- **Action:** Takes the user input of user email and username and checks all registered users and see if input is unique. If it's not unique return error message.
- **Pre-conditions:** Valid and unregistered email, Valid password format & Valid username.
- **Post-conditions:** Successfully save users data in a new database record.
- **Output:** Pop up message "Account has been created successfully" // Pop up message "Email or username are already in use" "Password is not strong enough"

4. Sign in

- **Description:** User signs in into their account
- **Input:** username/email & password
- **Source:** User
- **Action:** query will be sent to database to validate if username/email & password exist and in the same record
- **Pre-conditions:** No one is signed in, Valid entries (Valid email/username & password)
- **Post-conditions:** The username and password exist in the database
- **Output:** Pop up message “Sign in successful” // “Wrong email or password”

5. Rate Movie/TV-Show

- **Description:** User will be leaving a rating on the selected movie/TV show
- **Input:** number in range [1:10]
- **Source:** User
- **Action:** System takes the users rating and adds it to the record and calculates the new average rating of the movie/tv-show using an equation.
- **Pre-conditions:** user should be Registered/Logged-in
- **Post-conditions:** Re-calculation for the new average rating is done successfully and saved in the database.
- **Output:** Display the new average rating

6. Survey Movie/TV-Show

- **Description:** Registered user write a survey that show their opinion about the movie/TV-show
- **Input:** words in range [1:1000]
- **Source:** User
- **Action:** System takes the user survey and add it to the movie/TV-show page
- **Pre-conditions:** User should be registered, survey shouldn't be a blank one
- **Post-conditions:** The survey doesn't contain more than 1000 words
- **Output:** Display the written survey in the movie/TV-show page

7. Manage watchlist (Add/Delete)

- **Description:** Registered User has his own watchlist where he can add/delete movie/TV-show from it
- **Input:** Movie/TV-Shows
- **Source:** Plus sign in Home Page/Movie Page /TV-Show Page or delete button in watchlist page
- **Action:** Movie/TV-show added/deleted to the users' watchlist
- **Pre-conditions:** User should be registered
- **Post-conditions:** Movie/TV-show is added/deleted successfully to the user's watchlist and has been saved in our database
- **Output:** Display Movie/TV-show in the user's watchlist

8. Request edits

- **Description:** A registered user fill a form that request edit in our system, these edits are related to the data (Movie/TV-show) in our website or request to add a new movie/tv-show to our website
- **Input:** Form with the user's request
- **Source:** User
- **Action:** Registered user fill a form with the suggestion that he desire, and the form is sent to the system administrator so he can approve/disapprove the user's request
- **Pre-conditions:** User should be registered
- **Post-conditions:** The form is sent successfully to our system administrator
- **Output:** Pop-up Message that the form has been recorded successfully / Error Message in case that it has not been sent

9. View edits requests

- **Description:** Every registered user has a list that contain his edits requests and their status (Approved/Pending/Disapproved)
- **Input:** Click on view edit request page
- **Source:** User
- **Action:** Edits request status should be saved to our database
- **Pre-conditions:** User should be registered, user must be in edit request page
- **Post-conditions:** The request status has been updated successfully in our database
- **Output:** Display the request status in the list page

10. Manage user profile

- **Description:** A registered user can manage his profile by change his user, his password or his Email or manage his privacy.

- **Input:** Click on user's profile page
- **Source:** User
- **Action:** The data that user change will be saved in the database.
- **Pre-conditions:** Registered user and enter valid changes, user is in user's profile page
- **Post-conditions:** Data is saved successfully in the database
- **Output:** Pop-up message that the changes have been done successfully/
Error message with invalid input.

11. Manage website data (Admin)

- **Description:** The system administrator can add/edit/delete a movie/tvshow
- **Input:** Click on Manage Website Data page
- **Source:** Admin
- **Action:** The data that admin change will be saved in the database.
- **Pre-conditions:** logged in admin with his unique username and password, and enter valid changes (Matching data type), admin is in Manage website Data page.
- **Post-conditions:** Data is saved successfully in the database
- **Output:** Pop-up message that the changes have been done successfully/
Error message with invalid input.

12. Manage Edits Requests (Admin)

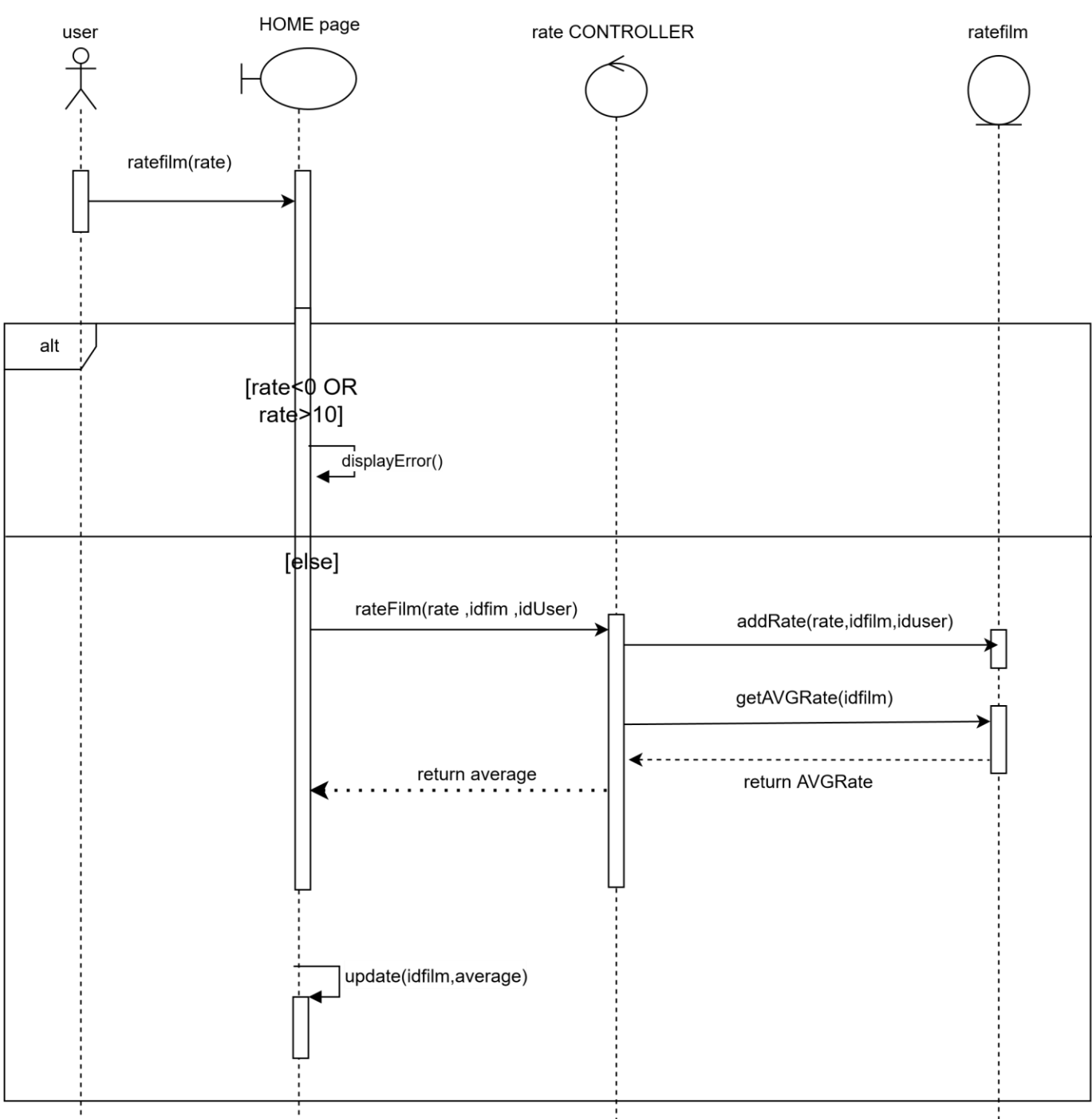
- **Description:** The system administrator View all users edits request and decide if he will make these edits into the website or not
- **Input:** Click on Manage Edits Requests page
- **Source:** Admin
- **Action:** The admin views all edits requests and change the request status.
- **Pre-conditions:** logged in admin with his unique username and password, admin is in Manage website Data page.
- **Post-conditions:** The Request status is sent to the user successfully.
- **Output:** Pop-up message that the request status is sent to the user successfully / Error message.

Non-functional Requirements:

1. Reliability ○ the percentage of the probability of failure can be a maximum 5%.
 - Number of critical failures for a function maximum 5 failures
2. Usability ○ easy to navigate through the website ○ purpose of features predicted easily
3. Capacity ○ comment of each time user write it can't be more than 1000 word
4. programming language ○ HTML, CSS, JAVA SCRIPT, PHP, and SQL.
5. Compatibility ○ The website should have the same features on all browsers
6. Response time ○ 5 second with strong internet
 - 7 second with weak internet
7. Security ○ Password protection, if the user enters his password or email wrong more than 10 times the system will ask him to answer his security question otherwise the account will be locked for 2 days
 - If the user chose to make any of his account information private it must be kept private, and no one can have access to it (ex: his ratings)
8. Scalability ○ The system should be able to handle 20 million users without performance deterioration.



add rate



search

