Software Engineering - CSCI-313

Movie recommendation System

SRS Document

Eslam Ahmed Mohamed	202000039
Esraa Negm Sayed	202000799
Mariam Amr Barakat	202000210
Mohamed Abdel-Maged Essawey	202000440
Rawan Mohammed Elframawy	202001762

Dr, Walaa Medhat

Teaching assistant: Shaymaa Sayed ElKaliouby

Table of contents

Contents

1. Introduction

1.1 Purpose

The objective of this documentation is to illustrate the platform's key features. It will discuss a general description of the website's user experience design and interface (UI/UX), the performance requirements, the use case diagram, and the Entity–relationship diagram (ERD). Furthermore, we will talk about the limits the platform must work under and how the website will respond to external factors.

1.2 Scope

Movie Recommender is a recommendation system, which provides users with movies which they may like, based on the movies that they previously saw. Every logged-in user should have access to the recommender system. It will be a system that has a database of movies with information related to the movies, along with a database that will store user preferences. The recommendation system will use mathematical models that relate to linear algebra to find which movies the user will like. And it will output a set of recommendations with pictures and information regarding the movies. Furthermore, it will provide users with a voting system so that they can judge the recommendations, so that we improve our algorithms and formulas used.

1.3 Technologies used

- **Flask**: A python back-end framework that allows building securable, clean, and maintainable websites.
- **HTML-CSS-Java Script**: Will be used in the front-end
- **Streamlit**: An open-source app framework in Python language. It helps us create web apps for data science and machine learning in a brief time. It will be used in the prototype phase.
- **MySQL**: Database management system for the users and movies data.

1.4 Intended Audience

This Software Requirements document is intended for:

- **Developers:** To have ease of implementation.
- **Software Testers:** Use this document as a base for their testing strategy as some bugs are easier to find using a requirements document.
- **Scrum Master:** The project manager that manages developers and testers in each sprint.
- **Customer:** To follow up with the project requirements.
- **End users of this application:** If they want to read about what this project can do.

1.5 Overview

The recommendation system is a website that helps users to find their next movie to watch. This is done by allowing the user to search for movies that they want to learn more about, as the website will provide the users with the movie description, trailer and other people's comments. Also, there is a feed which will contain recommended movies that the user can add to their watch later or their liked movies. Moreover, users can rate movies that they have watch and write their review. In the next chapters, we will give an overall description of the product including product perspective, user characteristics, constraints…etc. Then we will mention all the functional and non-functional requirements of the system. After that, we will discuss the use cases of the system. Finally, we will show the diagrams of the system as sequence diagram and class diagram.

2. Overall Description

2.1 Product Perspective

- "Movie Recommendation System" is a website that helps users who like movies to have movies recommended for them based on the data given from the other users and the user's previous information.
- The recommendation system analyzes the past preferences of the user concerned, and then it uses this information to try to find similar movies. This information is available in the database (e.g., lead actors, director, genre, etc.). After that, the system provides movie recommendations for the user.
- The website has two types of users: website administrators and website users interested in movies. Each of them has their own unique functionalities on the website.
- The website is compatible with any device that has a functioning web browser.

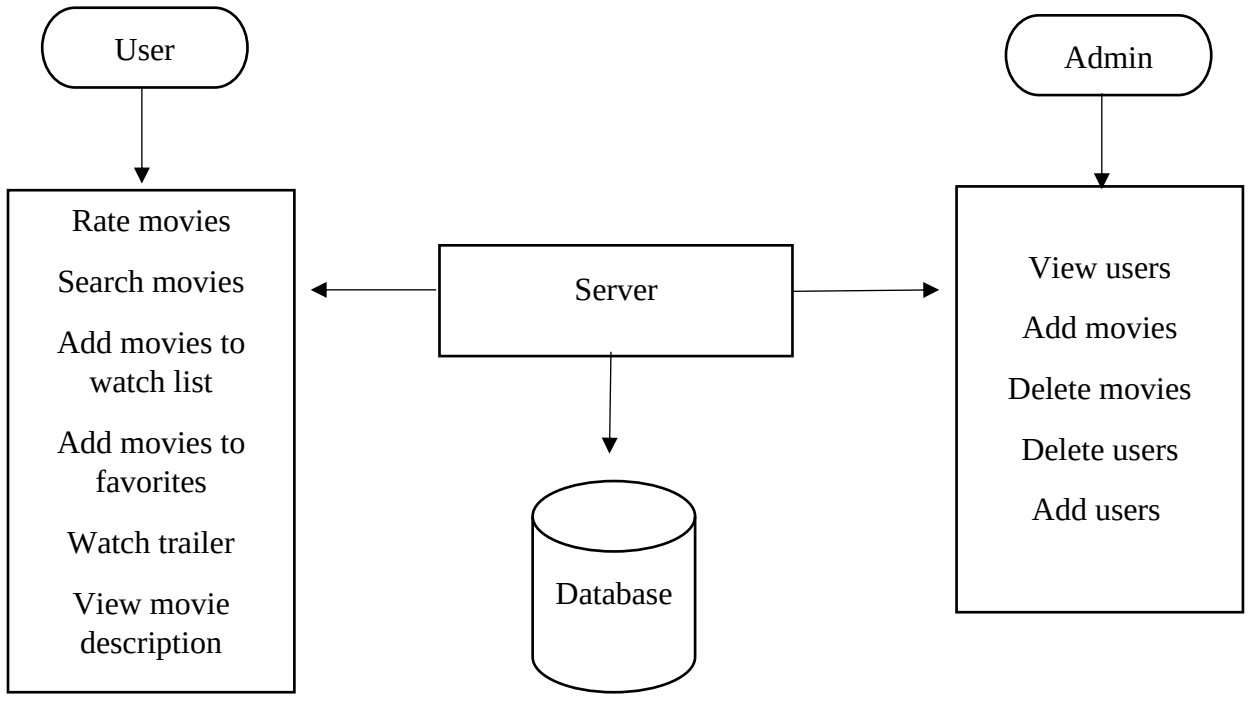

Figure 1 Block Diagram for recommendation system

2.2 User Characteristics

There are essentially two types of users that can interact with the system, website users and the administrators, surely each has different privileges.

- The user of our recommendation system can search for movies, add the movies to a watchlist, add movies to favorite's folder, watch the movies' trailer, and view the movie description.
- The administrator will interact with the website and the database; he will be able to view users, add users, delete users, add movies, delete movies, and view movies.

2.3 Operating Environment

The product is going to be a website that runs on any browser. For development we used windows OS.

2.4 Constraints

- The system filtration process depends on the comparison between the movies database and the user; therefore, all these processes happen via the Internet, so the user must have stable internet connection.
- Sign-in and password are used for the identification of the user.
- The system shall use the current standard MySQL database engine.

2.5 Assumptions and Dependencies

The following assumptions and dependencies were identified during the development of this plan:

- It is assumed that all users of this system have internet access.
- The user should have an installed web browser on his device.
- The user should have an account registered in the database

3. Interface

3.1 System Interface

The user will start by registering, this will include adding his full name, username, email, phone number, gender, and password. However, if the user already has an account he will click “I have an account”.

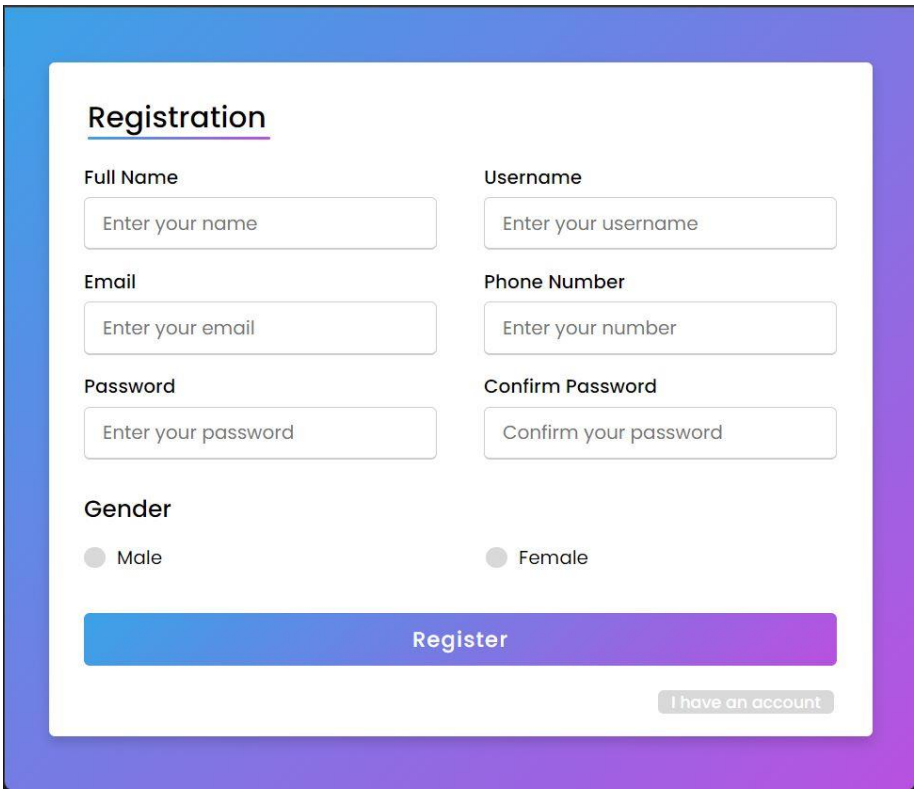

Figure 2 Registration Page

If the user is already registered, he will just be prompted to login using his username and password as seen in figure 2 below.

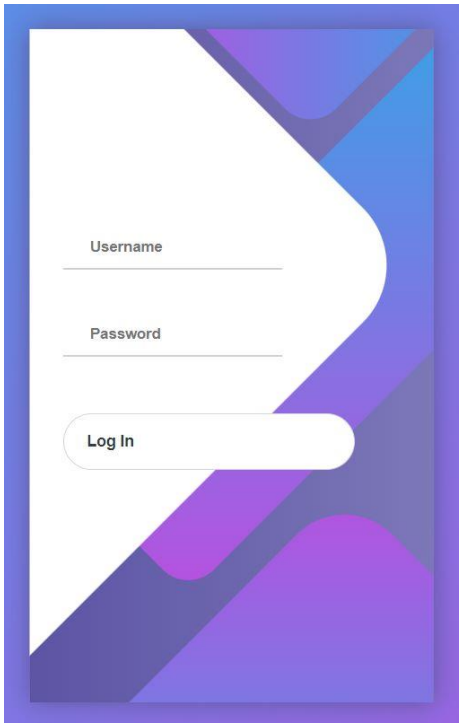

Figure 3 Login Page

This will lead the user to the home page which will contain all recommended movies. It will also allow the user to search for movies using the search bar, add to favorites and give his rating. This is shown in figure 4.

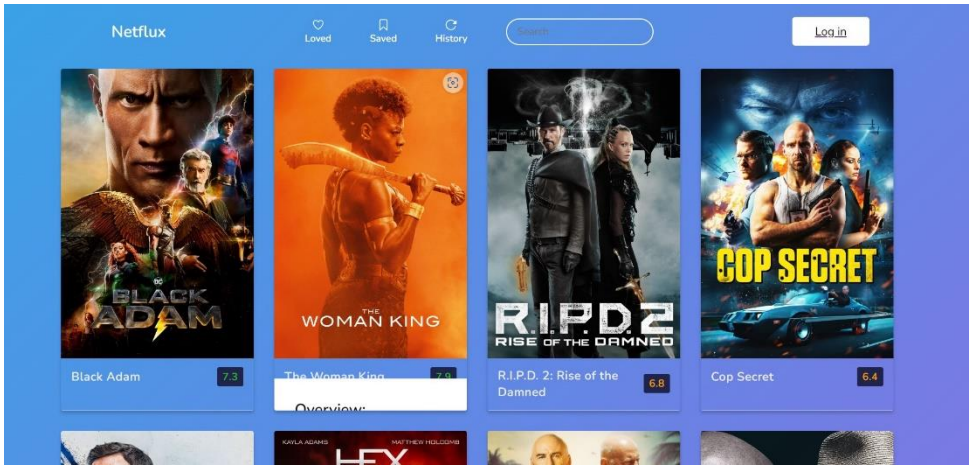

Figure 4 Movie Recommendation Page

Finally, when the user clicks on the movie the movie description with the trailer will be shown as in figure 5.

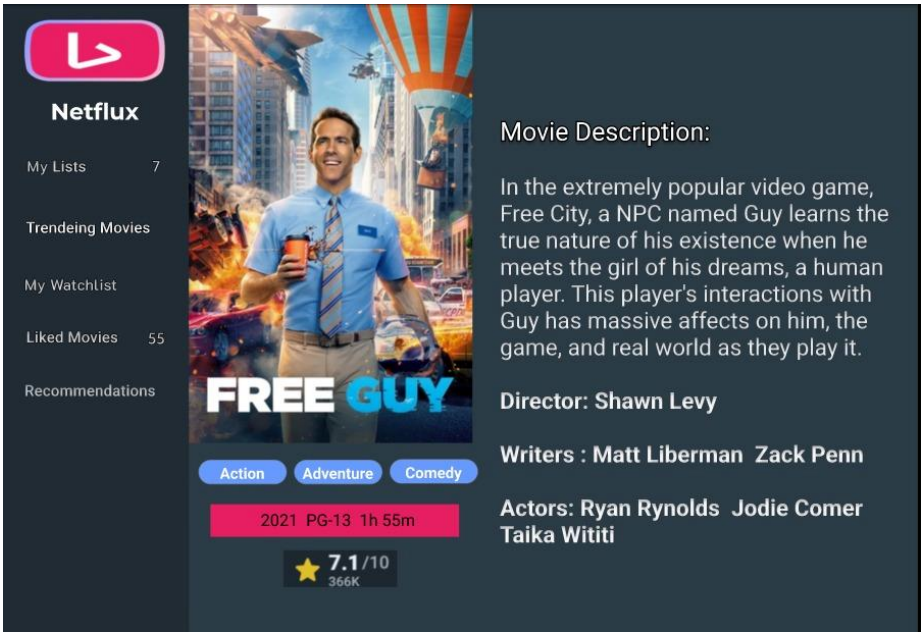

Figure 5 Movie Description Page

3.2 Software Interface

The movie recommendation system is under a web-based environment using HTML, CSS, and JavaScript. The system will need to have the browser installed to open the website and load the webpages. System shall use MySQL to store users and movies data. Moreover, it shall use flask framework to handle requests and responses between the system and backend.

4. Functional Requirement

4.1 User Class 1 – The User

4.1.1 Functional Requirements

Title: Login

Desc: System shall allow the existing users who have an account in the system to provide their username and password to enjoy website features.

4.1.2 Functional Requirements

Title: Sign up

Desc: System shall allow the users who are new to the system to make an account using personal data to register in the system.

4.1.3 Functional Requirements

Title: Add to favorites

Desc: System shall allow the users to add their favorite movies to a specific list called Favorites.

4.1.4 Functional Requirements

Title: Add to watchlist

Desc: System shall allow the users to save movies to a specific list called watchlist.

4.1.5 Functional Requirements

Title: Open favorites

Desc: System shall allow the users to open their favorite movies and view them.

4.1.6 Functional Requirements

Title: Open watchlist

Desc: System shall allow the users to access and view their watchlist.

4.1.7 Functional Requirements

Title: Search for a movie

Desc: System shall allow the users to search for movies by providing movie name.

4.1.8 Functional Requirements

Title: Open movie description

Desc: System shall allow the users to view details about movies if they exist.

4.1.9 Functional Requirements

Title: Rate a movie

Desc: System shall allow the users to give a rate for any movie.

4.1.10 Functional Requirements

Title: Edit profile

Desc: System shall allow each user to edit his personal data in his profile such as his name, username or reset password.

4.1.11 Functional Requirements

Title: Open history

Desc: System shall allow the users to view the movies they watched before as each watched movie is added to the history list.

4.1.12 Functional Requirements

Title: Get Recommendations

Desc: System shall suggest movies as a recommendation to user based on the dataset which is considered as collaborative approach.

4.1.13 Functional Requirements

Title: Filter Movies

Desc: System shall allow the user to click on the filter button, making the user choose which criteria to filter movies on.

4.2 User Class 2 – The Admin

4.2.1 Functional Requirements

Title: Add movies to the database of the system

Desc: System shall allow the admin staff to add or remove movies to the system.

4.2.2 Functional Requirements

Title: Edit movies

Desc: System shall allow the admin staff to edit movies details in the database to the system.

4.2.3 Functional Requirements

Title: Remove users.

Desc: System shall allow the admin staff to remove users if there is unexpected behavior.

5. Non- Functional Requirement

5.1 Reliability

- The average time of failures should not exceed a maximum of 2 times per day.

5.2 Recoverability

- After a system failure, it should take approximately 5 minutes to recover.

5.3 Performance

- The full response time for the main login page to load is two seconds.
- The system generates all possible recommendations for the movies in one second.

5.4 Usability

- The interface should be simple to understand and should enable users to successfully accomplish their goals, it takes the user 5 minutes to be able to navigate about 80% of the pages and use them correctly.

5.5 Availability

- The system should be capable of working 24/7.

5.6 Maintainability

- The system is being refactored weekly to make it easier to maintain the code base and correct bugs as they appear.
- The System should use continuous integration so that features and bug fixes can be deployed quickly without downtime.
- Used the command Design pattern.

5.7 Security

- The system is safe against any SQL injection methods.
- Database must be reached securely, and its data should not be broken.
- User’s passwords are encrypted in the database.

6. Diagrams

6.1 Use Case Diagram

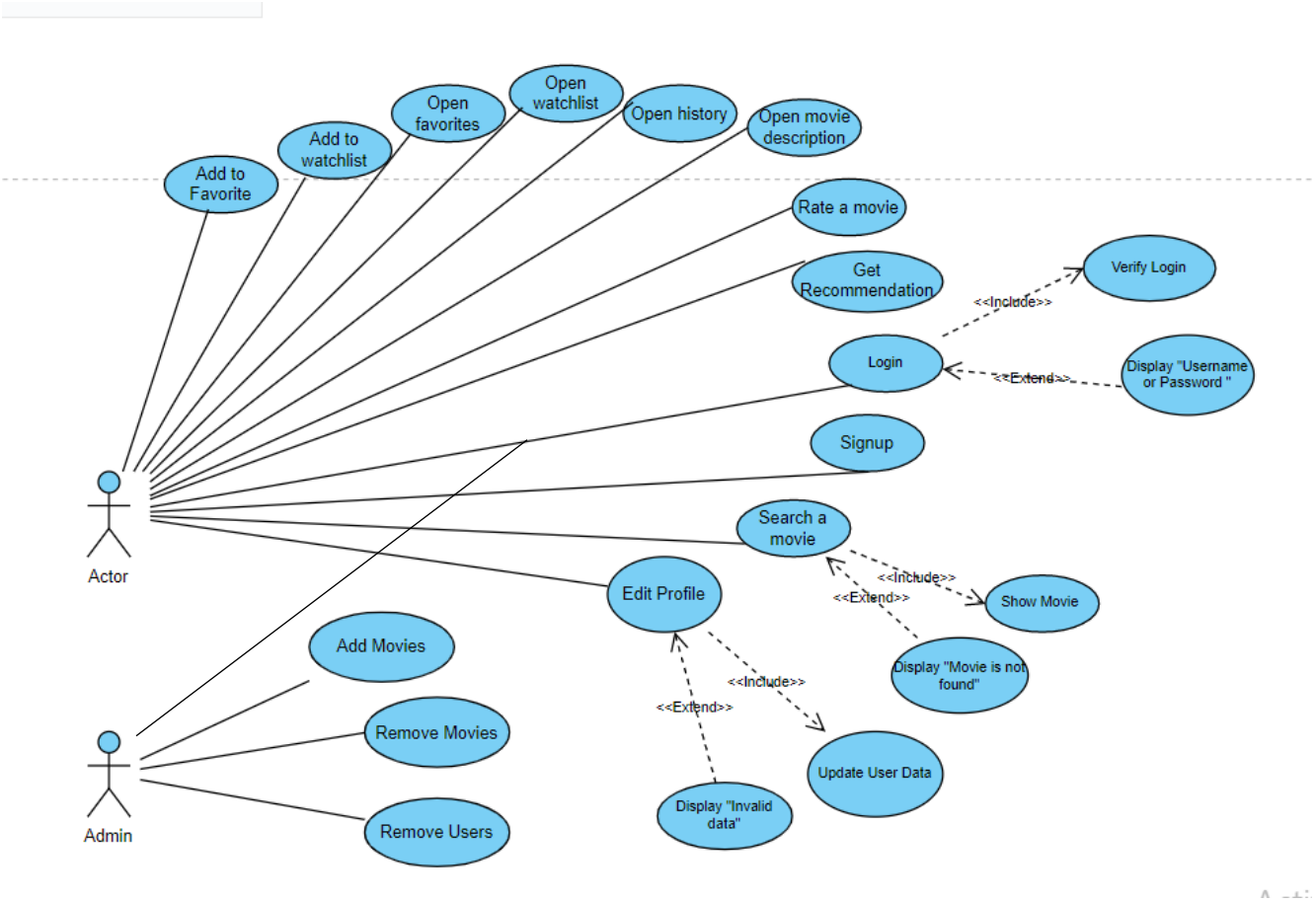

Figure 6 Use Case Diagram

6.1.1 Use Case Scenarios

Use case 1

Use case Name	**Login**
Actors	User
Main Success Scenario:	1. System displays login page for the user. 2. System requires user to type their username and password. 3. User clicks on login button. 4. The system validates the user data. 5. System logs the user into the system.
Exceptions	3a. User inputs invalid username or password. 3b. User logs in with missing fields.
Actions	3a. The system displays a warning message saying, “incorrect username or password, please try again”. And the fields are emptied, and user is required to retype his data. 3b. The system displays an error message saying, “please fill all fields”, and user is required to complete missing data.
Pre-Condition	User has an account already
Post-Condition	User will be logged into the system successfully

Use case 2

Use case Name	**Sign Up**
Actors	User
Main Success Scenario:	1. System displays Signup page for the user. 2. User required to fill full name, username, email, phone number, password, and gender. 3. Users click on register. 4. The system validates the user input. 5. System creates an account for user. 6. System logs in the user into the system.
Exceptions	3a. User inputs a username or email that are already in the system. 3b. User leaves any of the fields empty.
Actions	3a. System displays “this account is already available, please login instead”. User required to click “I have an account”. 3b. System displays “Please fill all fields”, the user is required to fill all missing data.
Pre-Condition	User opens the sign-up page successfully
Post-Condition	User will have a new account and logged in

Use case 3

Use case Name	**Add to favorites**
Actors	User
Main Success Scenario:	1. The user logs into his/her account successfully. 2. The user clicks the “Loved” button beside the movie desired. 3. The button changes color from white to red.
Exceptions	1a. User wants to add to favorites before logging in. 2a. User clicks on favorite icon, when it was already on.
Actions	1a. The system will display “Please login first”. 2a. System displays “Are you sure you want to unfavorite this movie?”
Pre-Condition	User is logged into his account
Post-Condition	The movie that is loved will be added to the “Loved” section

Use case 4

Use case Name	**Add to watchlist**
Actors	User
Main Success Scenario:	1. User is logged into his/her account successfully. 2. The user clicks the watchlist button beside the desired movie. 3. The button changes to a full white saved icon
Exceptions	1a. User is not logged in. 2b. User clicks a movie that is already in the saved movies.
Actions	1a. The system displays “Please login into your account.” 2b. The system displays “This movie is already saved”.
Pre-Condition	User is successfully logged in.
Post-Condition	The movie will be saved in the “watchlist” section.

Use case 5

Use case Name	**Open favorites**
Actors	User
Main Success Scenario:	1. User is logged in his account. 2. User clicks the loved button at the top bar. 3. The system opens the loved section, and all movies the user loved will appear in a gallery.
Exceptions	1a. User isn't logged in into his account.
Actions	1a. The system displays "please login into your account". User is expected to click login.
Pre-Condition	The user is successfully logged into his account
Post-Condition	The loved movies page is opened successfully

Use case 6

Use case Name	**Open watchlist**
Actors	User
Main Success Scenario:	1. User is logged in his account 2. User clicks the watchlist button at the top bar 3. The system opens the watchlist section, and all movies the user saved will appear in a gallery
Exceptions	1a. User isn't logged in
Actions	1a. The system displays "Please login first"
Pre-Condition	The user is successfully logged into his account
Post-Condition	The watchlist page is opened successfully

Use case 7

Use case Name	**Search for a movie**
Actors	User
Main Success Scenario:	1. User clicks the search button. 2. User types a movie name. 3. The system will search for the movie. 4. The movie appears and similar movies.
Exceptions	2a. User types a movie that isn't in the database
Actions	2a. The system will display "No such movie found"
Pre-Condition	1. The user is logged in successfully. 2. User types a correct movie name.
Post-Condition	A gallery of the searched movie and similar movies appear

Use case 8

Use case Name	**Open movie description**
Actors	User
Main Success Scenario:	1. The user hovers over a movie 2. The description appears
Exceptions	None
Actions	None
Pre-Condition	User is successfully logged into his account
Post-Condition	The description of a movie appears

Use case 9

Use case Name	**Rate a movie**
Actors	User
Main Success Scenario:	1. User is logged in. 2. User clicks on a movie. 3. The system opens the video player. 4. User scrolls down. 5. User clicks on rate and comment. 6. System displays the star rating and an editable box. 7. User is prompted to assign the star rating 8. User is prompted to type his comment. 9. User clicks on save. 10. The comment is posted on page.
Exceptions	7a. User doesn't assign a star rating and clicks save 8a. User's comment has offensive speech 8b. User doesn't type anything and clicks on save button 9a. User doesn't click the saved button
Actions	7a. System prompts "Please give your star rating" 8a. System displays "Offensive speech is not allowed" 8b. System prompts "Please write a comment" 9a. System displays "Your rating isn't saved, are you sure you want to exit?"
Pre-Condition	User is logged in successfully into his account
Post-Condition	The user's rating and comment is posted successfully

Use case 10

Use case Name	**Edit Profile**
Actors	User
Main Success Scenario:	1. User is logged into his account. 2. User clicks profile button. 3. User clicks edit profile button. 4. User edits by adding/changing profile picture, username, password, or description. 5. User clicks save. 6. System saves changes to the database.
Exceptions	4a. User wants to change his password and wrote wrong old password 4b. User uploads a picture that is too large 4c. New username is already available
Actions	4a. System displays a warning message saying, "Please re-enter your old password correctly". The user is expected to re-write his old password. 4b. System displays a warning message saying, "Picture is too large". The user is expected to re-upload a new picture. 4c. System displays a warning message saying, "Please enter another username, this is already taken". User then is required to retype a new username.
Pre-Condition	User is logged in into his account
Post-Condition	Changes to his account is successful

Use case 11

Use case Name	**Open History**
Actors	User
Main Success Scenario:	1. User is logged into his/her account 2. User clicks on history button 3. The system opens a page with all the movies the user has watched or checked
Exceptions	1a. User isn't logged in
Actions	1a. The system will prompt the user to log in, a message is displayed saying "Please login first." User is expected to click the login button.
Pre-Condition	User is logged into his or her account.
Post-Condition	The history of the user appears on the screen.

Use case 12

Use case Name	**Get Recommendation**
Actors	User
Main Success Scenario:	1. User is logged into his account. 2. User clicks the homepage. 3. A gallery of movies is displayed and are a series of recommended movies.
Exceptions	1a. User not logged in.
Actions	1a. The system displays “Please login first”
Pre-Condition	The user must have logged in successfully
Post-Condition	A gallery of recommended movies appears

Use case 13

Use case Name	**Filter Movies**
Actors	Users
Main Success Scenario:	1. User clicks the filter icon. 2. User selects what criteria to filter upon, date, genre, actors, directors, or production companies. 3. User clicks done. 4. The page loads movies that obey the given criteria.
Exceptions	3a. User doesn't click on done, and clicks elsewhere
Actions	3a. The system displays "Your filtered attributes will be discarded. Press filter to save or exit otherwise."
Pre-Condition	User has opened the website home page successfully and logged in.
Post-Condition	A gallery of filtered items is displayed.

Use case 14

Use case Name	**Add movie**
Actors	Admin
Main Success Scenario:	1. Admin logged in as an admin. 2. Admin clicks on the add movie icon. 3. Admin adds details of movie. 4. Admin clicks “add.”
Exceptions	1a. The person logging in doesn’t have the admin privilege. 2a. Admin added a movie that already exists.
Actions	1a. The system displays a “You don’t have admin privilege”. 2a. The system displays “This movie already exists”.
Pre-Condition	The user must be logged in as an admin.
Post-Condition	The admin successfully adds a new movie to the system, and it is revealed to the user.

Use case 15

Use case Name	**Remove movie**
Actors	Admin
Main Success Scenario:	1. Admin logged in as an admin. 2. Admin clicks on the remove movie icon. 3. Admin clicks on the movie to remove. 4. Admin reconfirms the deleting process.
Exceptions	1a. The person logging in doesn't have the admin privilege 4a. Admin doesn't click on confirm and wants to open another page
Actions	1a. The system displays a "You don't have admin privilege" message. 4a. The system displays "Please click confirm to save changes"
Pre-Condition	The user must be logged in as an admin.
Post-Condition	The admin successfully deletes a movie to the system, and it is revealed to the user.

Use case 16

Use case Name	**Remove User**
Actors	Admin
Main Success Scenario:	1. Admin logged in as an admin. 2. Admin clicks on the “remove” user icon. 3. Admin types of the username of the user. 4. Admin clicks the user to delete 5. Admin reconfirms the deleting process.
Exceptions	1a. The person logging in doesn’t have the admin privilege 3a. Admin writes an incorrect username 5a. Admin doesn’t confirm and wants to navigate elsewhere in the system
Actions	1a. The system displays a “You don’t have admin privilege” message. 3a. System displays “No such user to be able to delete” 5a. The system displays “Please click confirm to save changes”
Pre-Condition	The user must be logged in as an admin.
Post-Condition	The admin successfully deletes a user to the system.

6.3 Sequence Diagram

6.2.1 Filter a movie

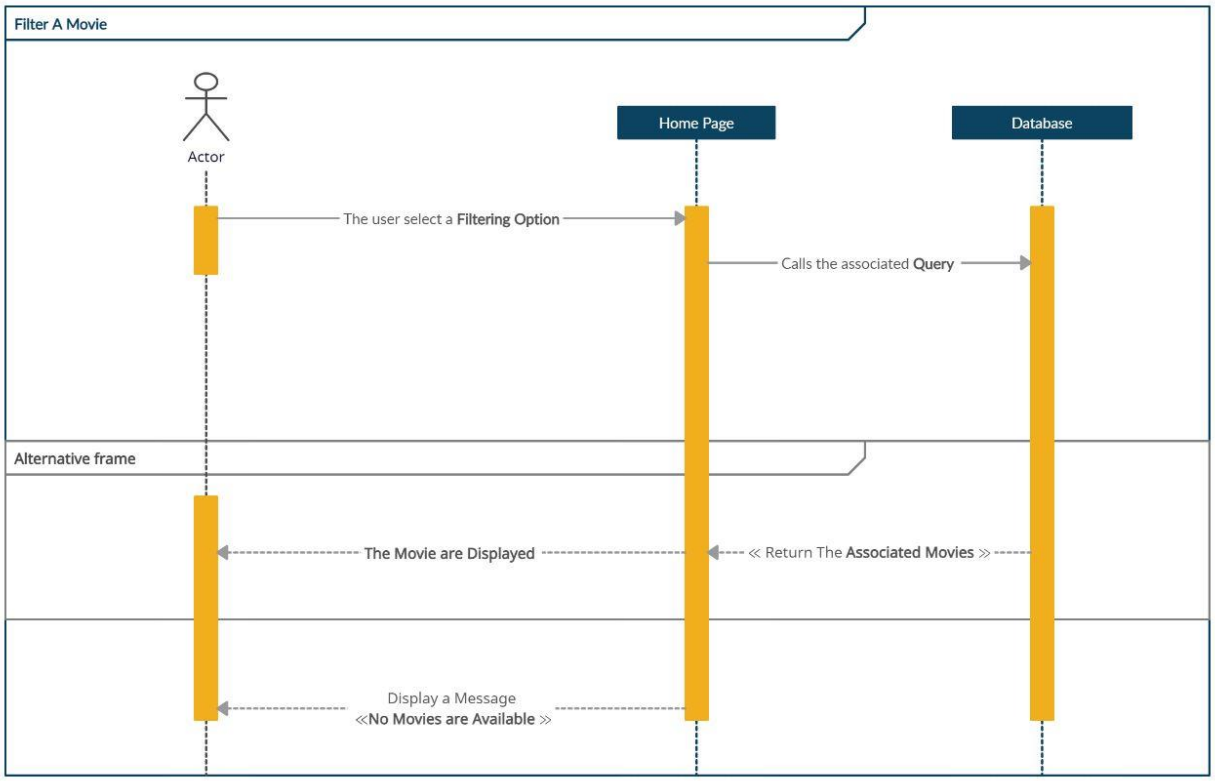

Figure 7 Filter Movie Sequence Diagram

6.2.2 Rate a movie

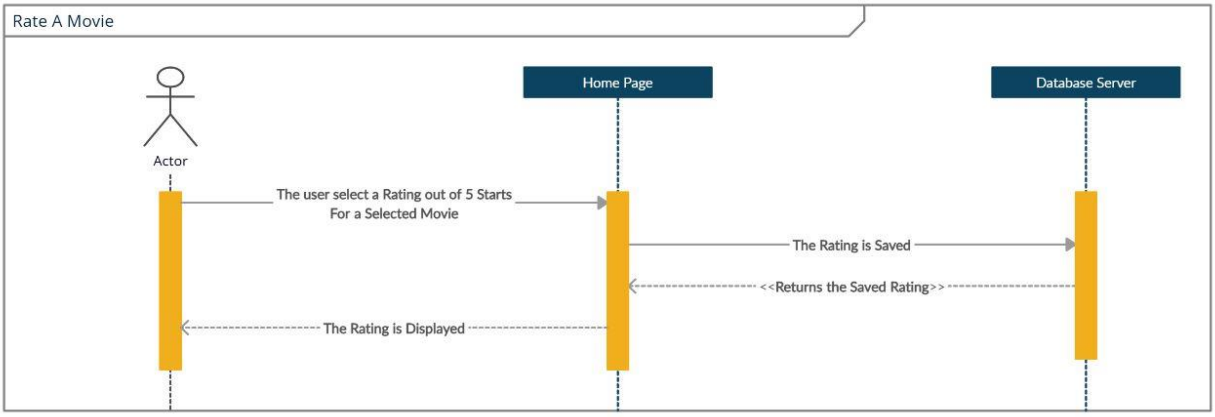

Figure 8 Rate a Movie Sequence Diagram

6.2 Class Diagram