



Object Oriented Analysis & Design with Java

UE20CS352

Movie Review System

Team info(K section)

- **PES1UG20CS657** - Kushagra Singh
- **PES1UG20CS667** - Nikhil Raju Mohite
- **PES1UG20CS670** - Pavan Kumar Nuthi
- **PES1UG20CS673** - Pranav R

Submission Date: 28-04-2023

GitHub Repository Link -

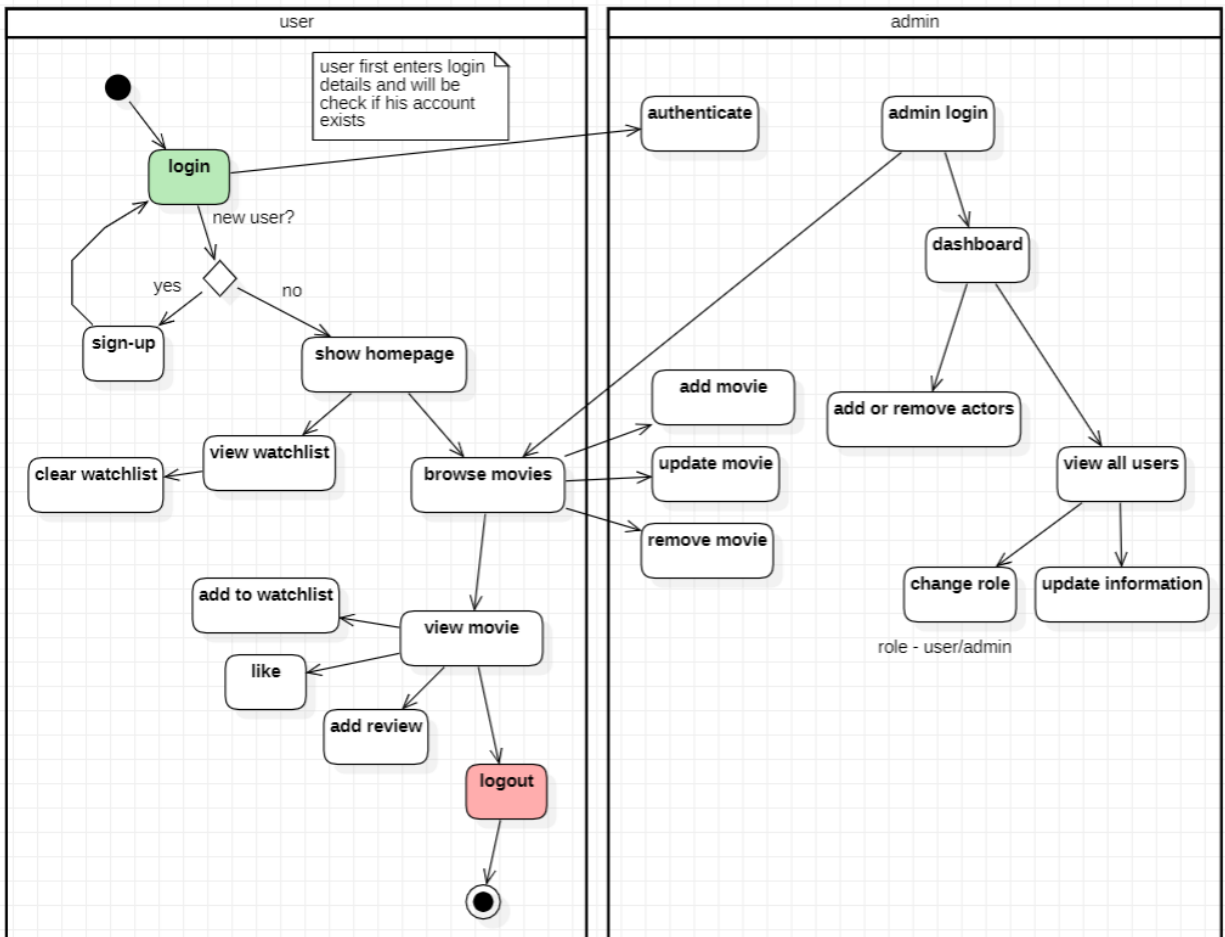
<https://github.com/nkilm/movie-review-system-spring-boot>

Abstract

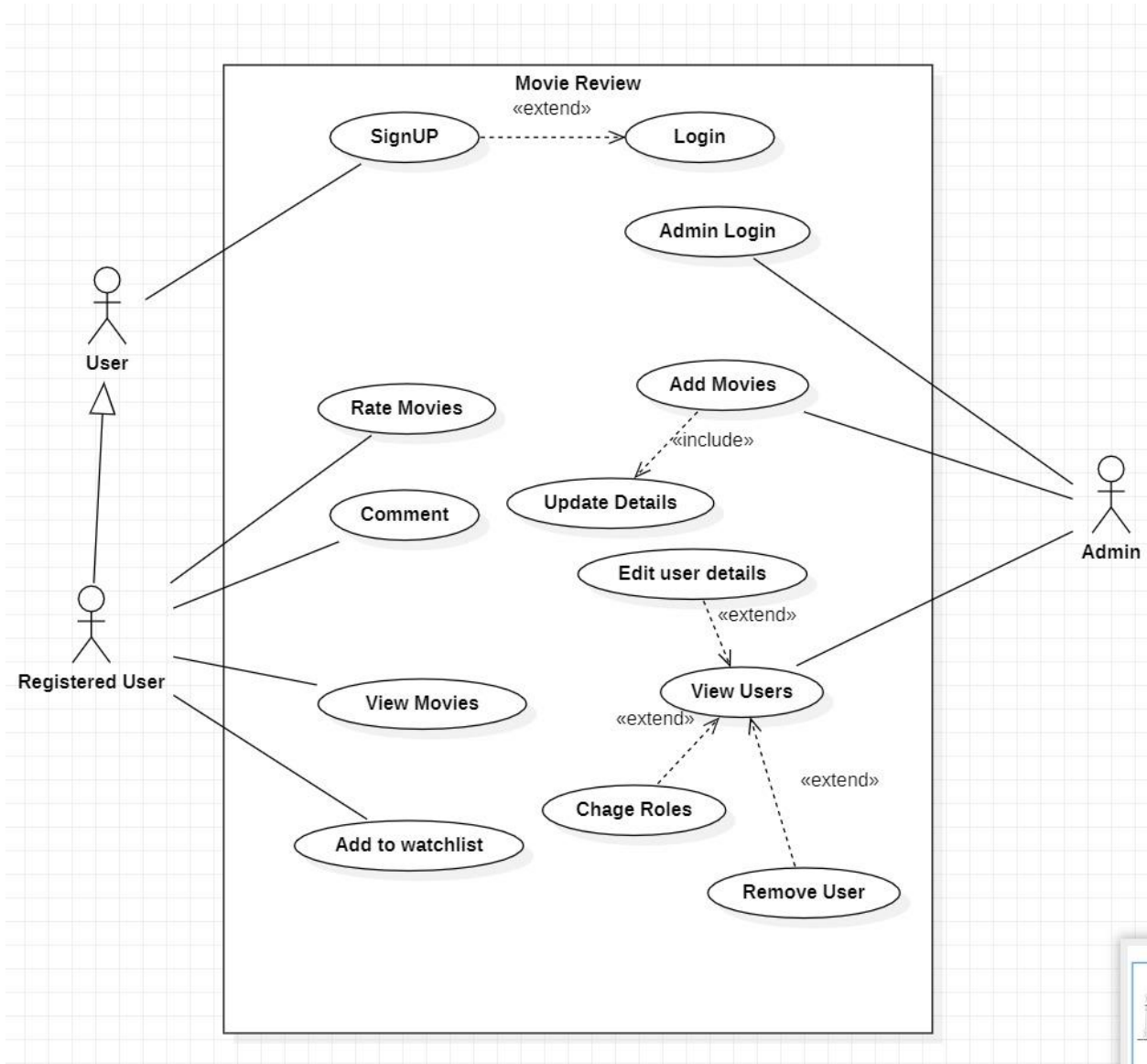
The Movie Review System is a project developed using Spring Boot framework. The system provides a platform for movie enthusiasts to view and post reviews about various movies. The system features user registration and login, movie browsing, review posting, and searching functionalities. The application allows users to rate and comment on movies and also provides a recommendation system based on the user's previous review history. The project is designed with a responsive user interface and follows a RESTful API architecture. The implementation of the system involved the use of technologies such as Spring Boot, MySQL, HTML/CSS, and Bootstrap. Overall, the Movie Review System aims to provide an enjoyable and informative experience for movie lovers.

DIAGRAMS

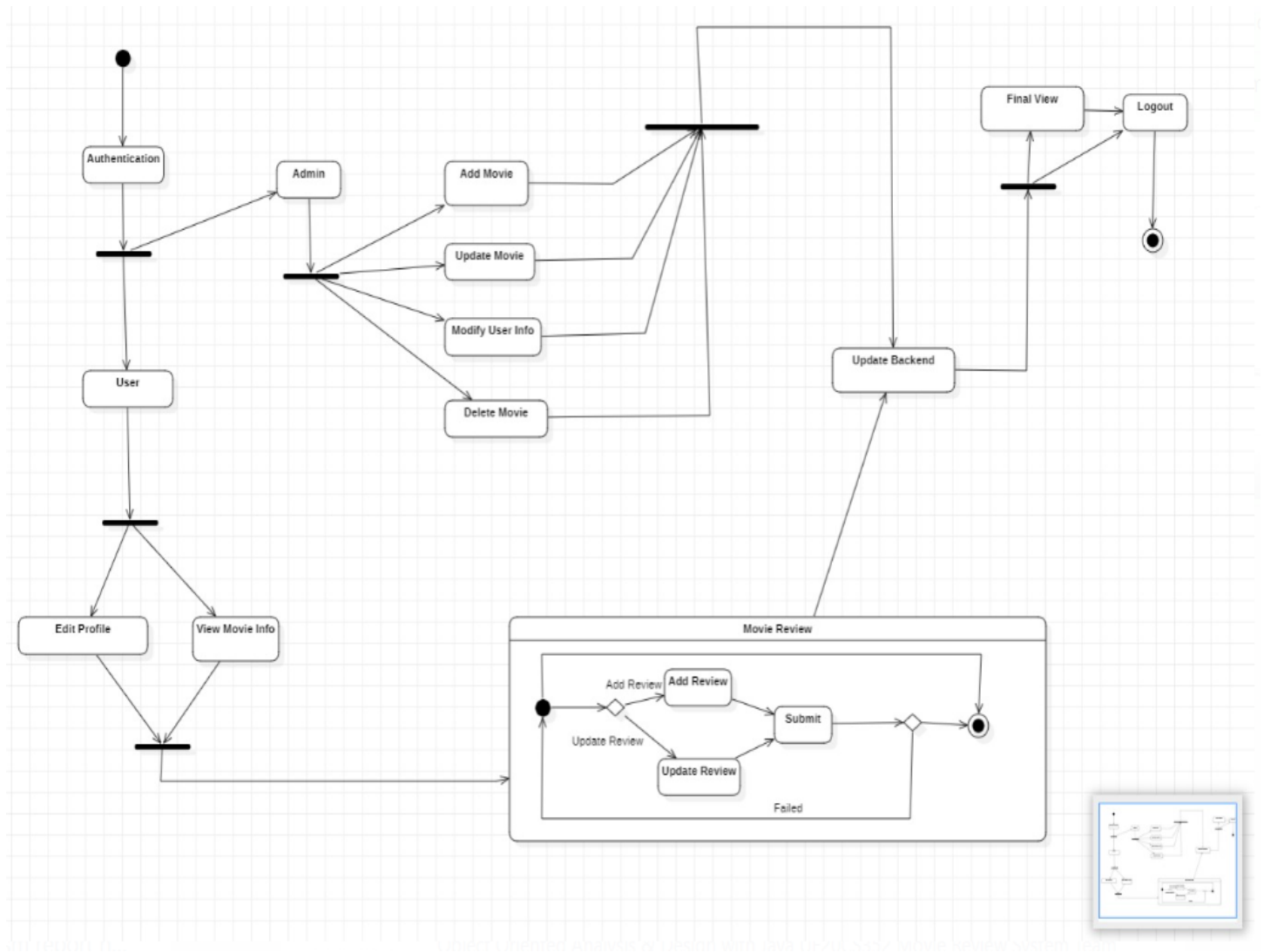
Activity Diagram



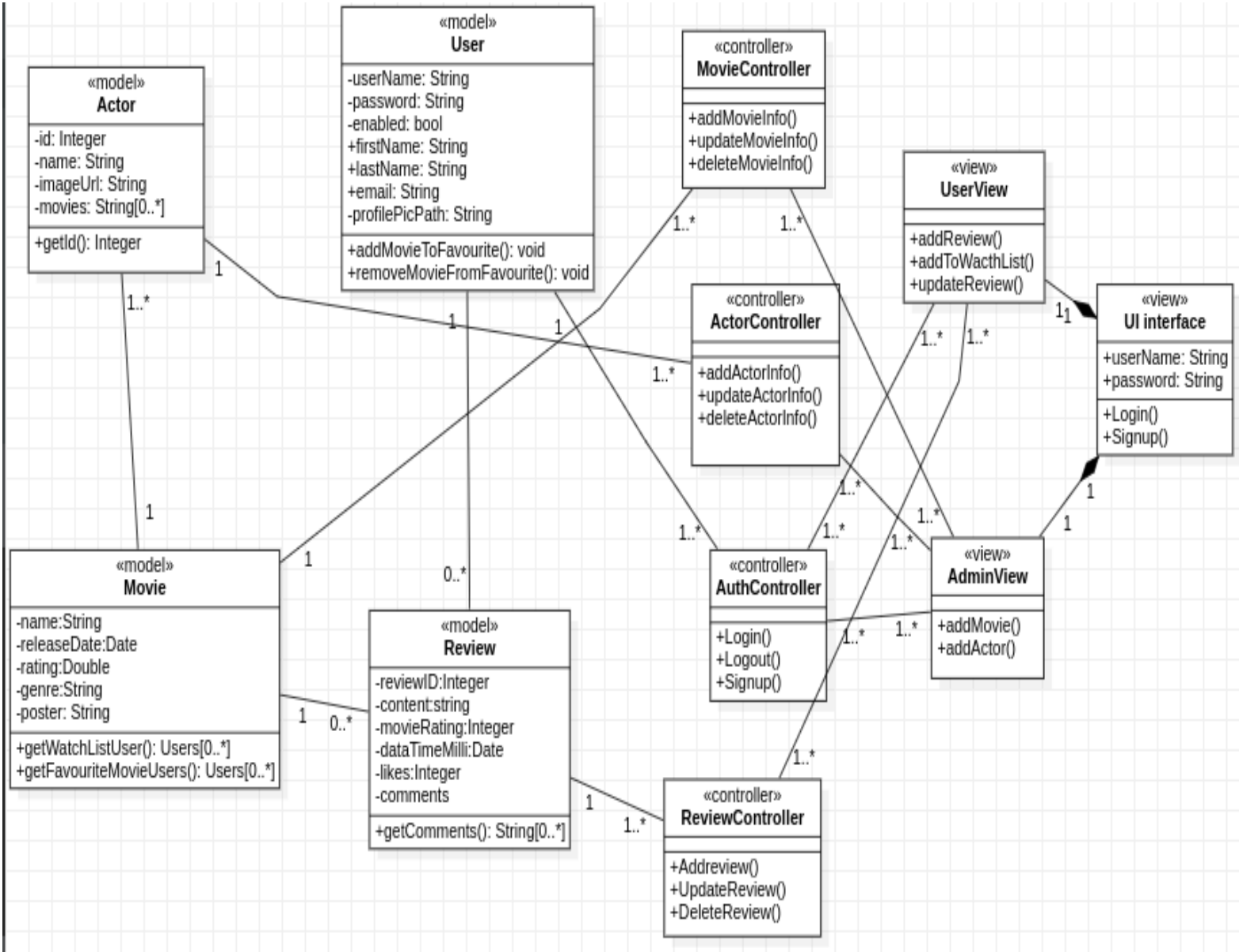
Use Case diagram



STATE DIAGRAM



CLASS DIAGRAM



ARCHITECTURE PATTERN:

The MVC (Model View Controller) pattern divides the application logic into three interconnected components: the model, the view, and the controller.

- **MODEL:** The Model component represents the data and business logic of the application. It includes classes such as Student, Query, and User for login. These classes are responsible for retrieving and manipulating data, as well as performing any necessary calculations or algorithms.
- **VIEW:** The View component is responsible for rendering the data from the model and presenting it to the user in a user-friendly way. This component includes web pages and interfaces that users interact with, such as login screens, dashboards, and queries.
- **CONTROLLER:** The Controller component acts as an intermediary between the model and the view. In the project's context, a single controller is used for all user categories. The controller receives requests from users through the view and uses the model to perform necessary operations to fulfill those requests. This includes handling user inputs, validating user data, and invoking appropriate methods on the model to update or retrieve data. The controller then returns the results to the view for display to the user.

DESIGN PRINCIPLES :

The project adheres to the following principles:

- **Single Responsibility Principle (SRP):** This principle suggests that each module or class should have only one responsibility or reason to change. In this project, the Student, Faculty, Coordinator, and IT Staff modules are responsible only for their own functionalities and should not overlap with other modules.
- **Open-Closed Principle (OCP):** This principle suggests that a class or module should be open for extension but closed for modification. In this project, the Elective Allocation Algorithm is designed in a way that it can be extended in the future if there are changes in the allocation rules, without modifying the existing code.
- **Dependency Inversion Principle (DIP):** This principle suggests that high-level modules should not depend on low-level modules, but rather on abstractions. In this project, the Student, Faculty, Coordinator, and IT Staff modules depend on abstractions, such as interfaces, rather than concrete implementations.
- **Interface Segregation Principle (ISP):** This principle suggests that a class should not be forced to implement interfaces that it does not use. In this project, the modules only implement the interfaces that are relevant to their functionalities.
- **Don't Repeat Yourself (DRY):** This principle suggests that code should not be duplicated but instead be abstracted into reusable components. In the context of this project, common functionalities such as database access, authentication, and logging is abstracted into separate components to avoid code duplication.

DESIGN PATTERNS:

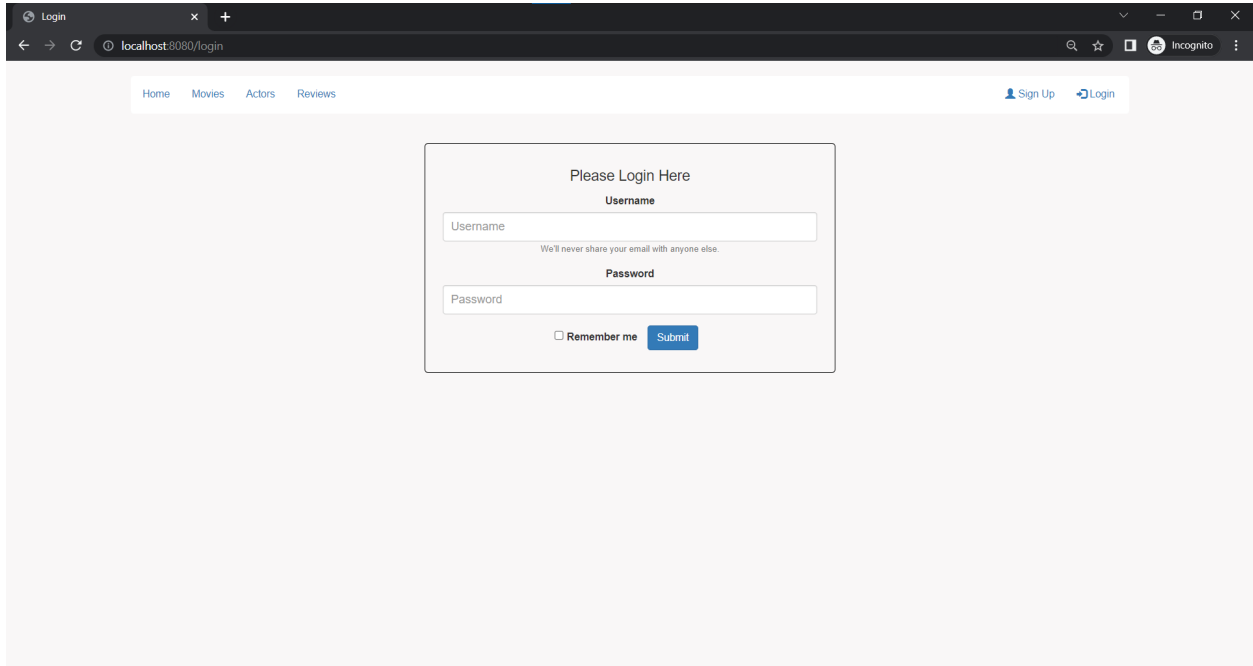
- **Singleton Pattern:** This pattern ensures that only one instance of a class is created and provides a global point of access to it. In the Elective Automation System, a singleton pattern is used to ensure that only one instance of the algorithm for allocation of students and faculty is created.
- **Factory Pattern:** This pattern provides a way to create objects without exposing the creation logic to the client. A factory interface provides methods to create objects, while a factory implementation provides the actual implementation of these methods. In the Elective Automation System, a factory pattern is used to create instances of the different modules (Student, Faculty, Coordinator, and IT staff).
- **Builder Pattern:** The Builder pattern is a creational design pattern that separates the construction of a complex object from its representation, allowing the same construction process to create various representations. In the Elective Automation System, a builder pattern is used to update a user's profile. We can update any attribute we want without having to remember the values of other attributes.
- **Repository Pattern:** This pattern separates the application's data access logic from the rest of the application. A repository interface provides methods to access and manipulate data, while a repository implementation provides the actual implementation of these methods. The repository pattern is used to handle database operations in the Elective Automation System.

GitHub Repository Link -

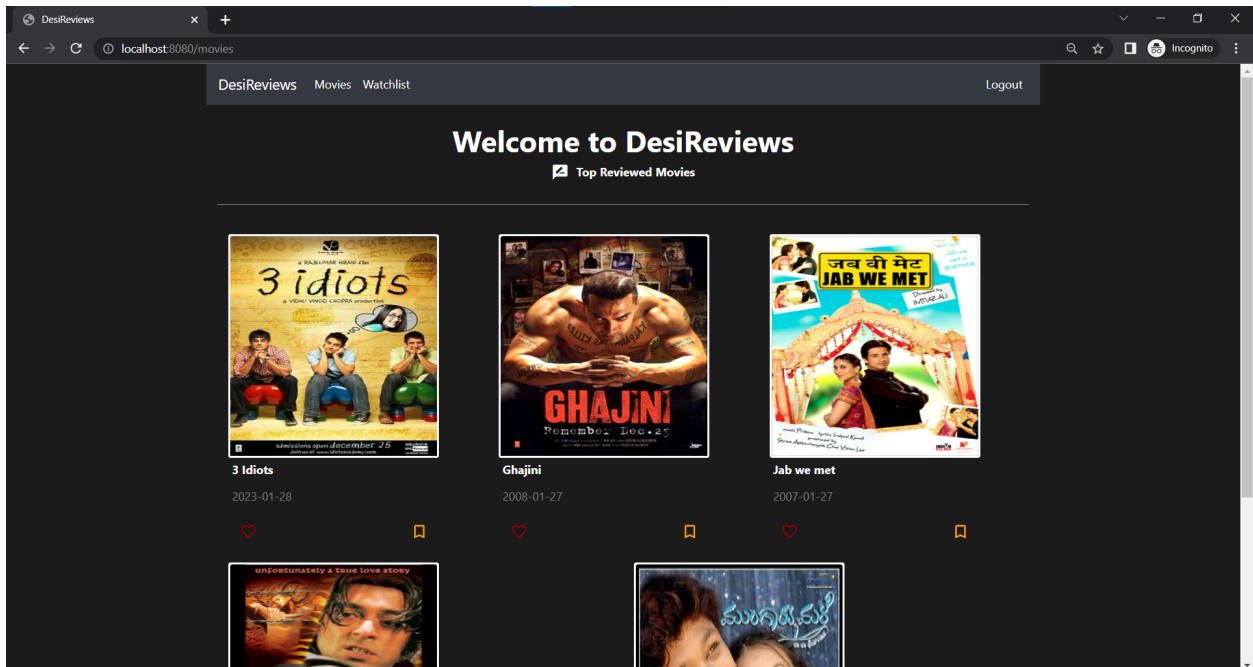
<https://github.com/nkilm/movie-review-system-spring-boot>

Visuals of the webpage/Output

login/signup page



homepage/movies



Viewing individual movie

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/movies/3'. The page has a dark theme and a navigation bar with links for 'DesiReviews', 'Movies', 'Watchlist', and a 'Logout' button. The main content area features the movie poster for '3 Idiots' on the left. To the right of the poster, the movie title '3 Idiots' is displayed in large white text. Below the title, the 'Rating:' field is empty, and the 'Genre: Comedy' and 'Release date: 2023-01-28' are shown. A 'Description:' section contains the text: 'The story follows the friendship of three students at an Indian engineering college and is a satire about the social pressures under the Indian education system.' Below the description is a 'Cast:' section with a placeholder. At the bottom of the movie information section, there is a 'Select Rating' dropdown menu set to '1' and a 'Detailed Review' text area with the placeholder 'Write a review...'. A blue 'Submit Review' button is located at the bottom right of the review form. Below the review form, the username 'amoghnrao' and the timestamp '28-04-2023 12:49' are displayed. At the very bottom, a star rating of '10 /10' is shown.

Reviews given by users

This screenshot shows the same movie page for '3 Idiots' as the previous one, but with two user reviews displayed below the review form. The first review is from 'testuser' on '28-04-2023 12:49', with a rating of '10 /10' and the text 'best bachlor movie'. The second review is from 'test' on '01-05-2023 13:22', also with a rating of '10 /10' and the text 'Really good movie!'. The review form and movie information section remain the same as in the previous screenshot.

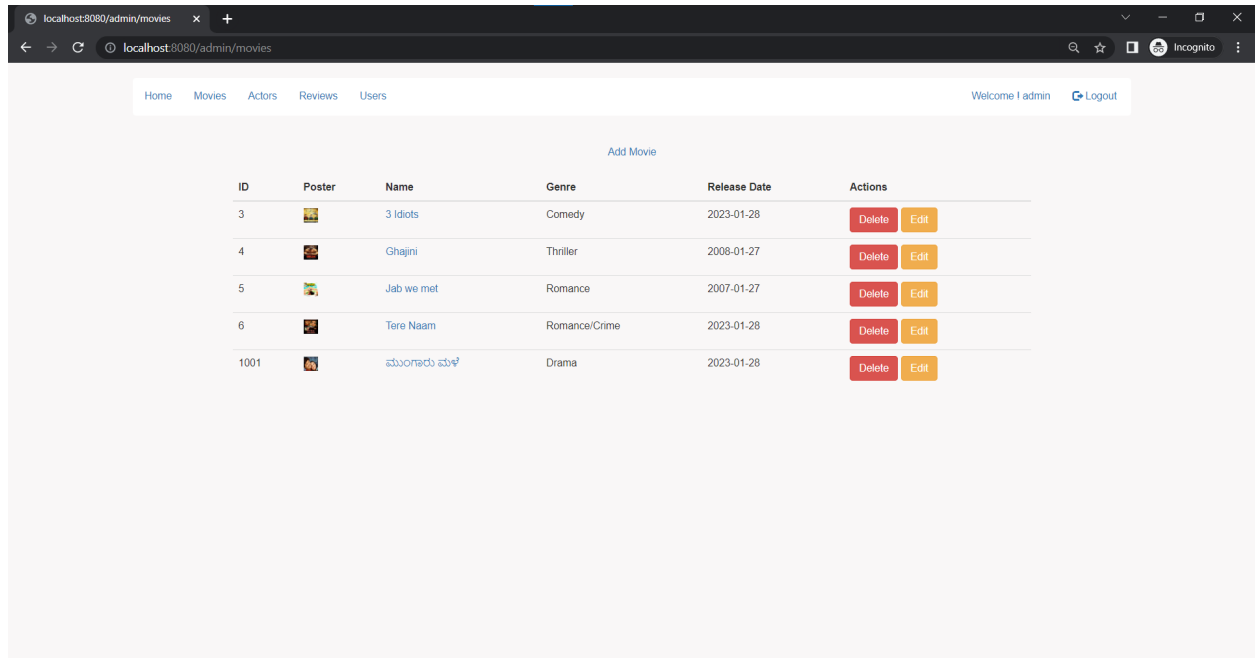
Updating the review - the user can update only his reviews.






The screenshot shows a web browser window with the address bar displaying `localhost:8080/movies/3/review/edit/7`. The browser's Incognito mode is active. The website's header is dark gray and includes the text 'DesiReviews', 'Movies', 'Watchlist', and a 'Logout' link. The main content area is light gray and features a yellow box titled 'Edit Your Review'. Inside this box, there is a 'Select Rating' dropdown menu currently set to '10', a 'Detailed Review' text area containing the text 'Really good movie!', and a blue 'Update Review' button.

Admin functions - only admin user can perform these

Admin can perform CRUD operation(s) on Movie, Actor, and user.

Admin dashboard



Add Movie					
ID	Poster	Name	Genre	Release Date	Actions
3		3 Idiots	Comedy	2023-01-28	Delete Edit
4		Ghajini	Thriller	2008-01-27	Delete Edit
5		Jab we met	Romance	2007-01-27	Delete Edit
6		Tere Naam	Romance/Crime	2023-01-28	Delete Edit
1001		ಮುನ್ಸೂರು ರಾತ್ರಿ	Drama	2023-01-28	Delete Edit

MOVIE

Adding movie

DesiReviews

localhost:8080/admin/movies/add

DesiReviews AdminActorsMoviesMango People View

Add a New Movie

Name

Release Date

New Movie

02-05-2023

Genre

Drama

Poster

Choose Filebg-image.jpg

Casts

Selmon Bhai

Description

A movie about 21st century.

Add

Editing Movie

DesiReviews

localhost:8080/admin/movies/edit/3

DesiReviews AdminActorsMoviesMango People View

Add a New Movie

Name

Release Date

3 Idiots

dd-mm-yyyy

Genre

Comedy

Poster

Choose FileNo file chosen

Casts

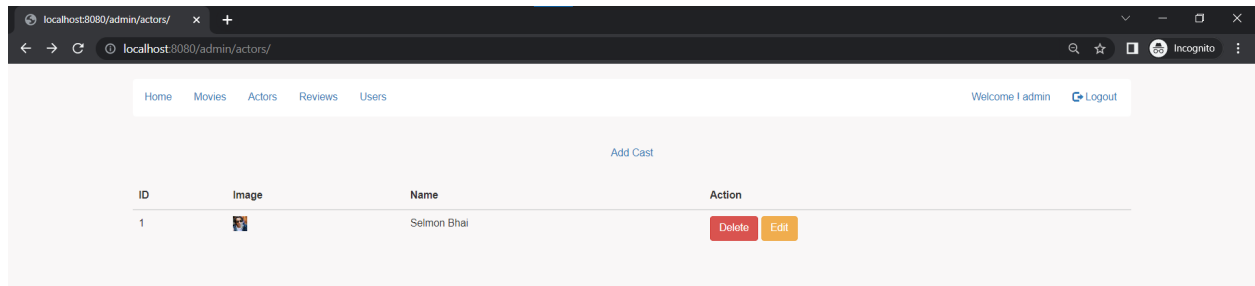
Selmon Bhai

Description

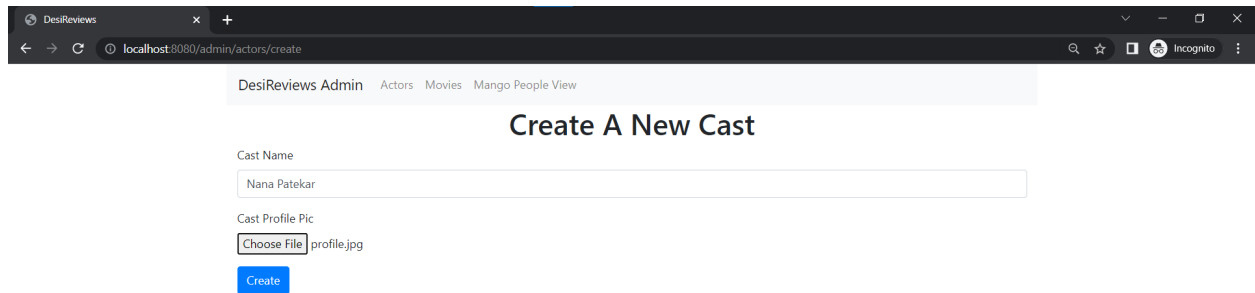
The story follows the friendship of three students at an Indian engineering college and is a satire about the social pressures under the Indian education system.

Update

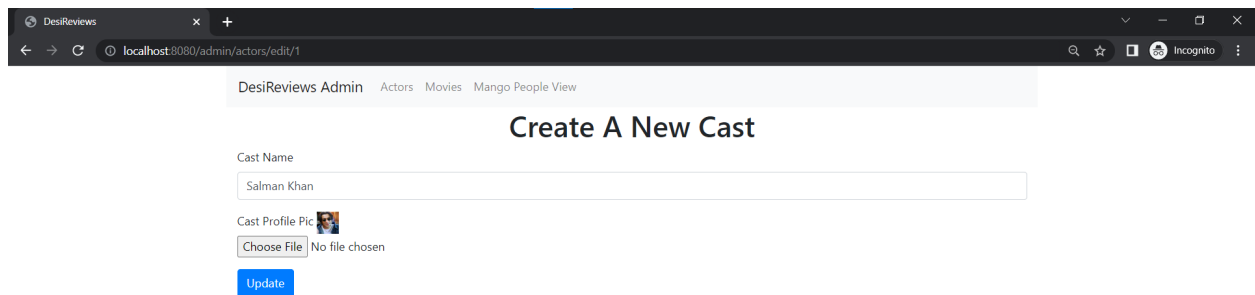
View all cast



Adding cast

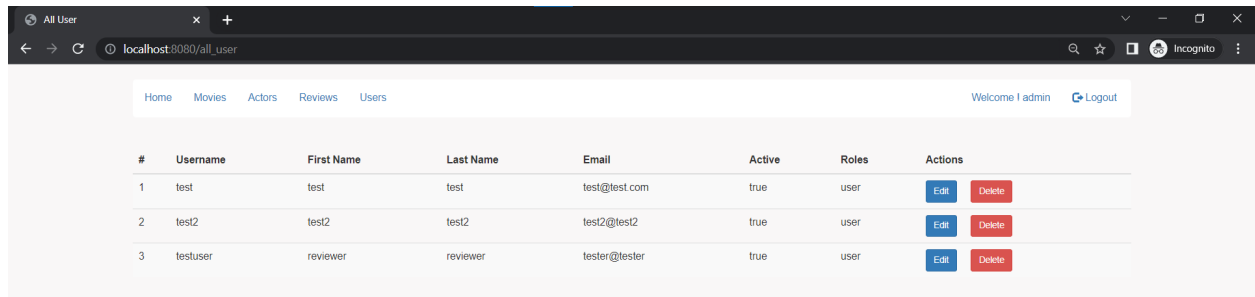


Editing cast



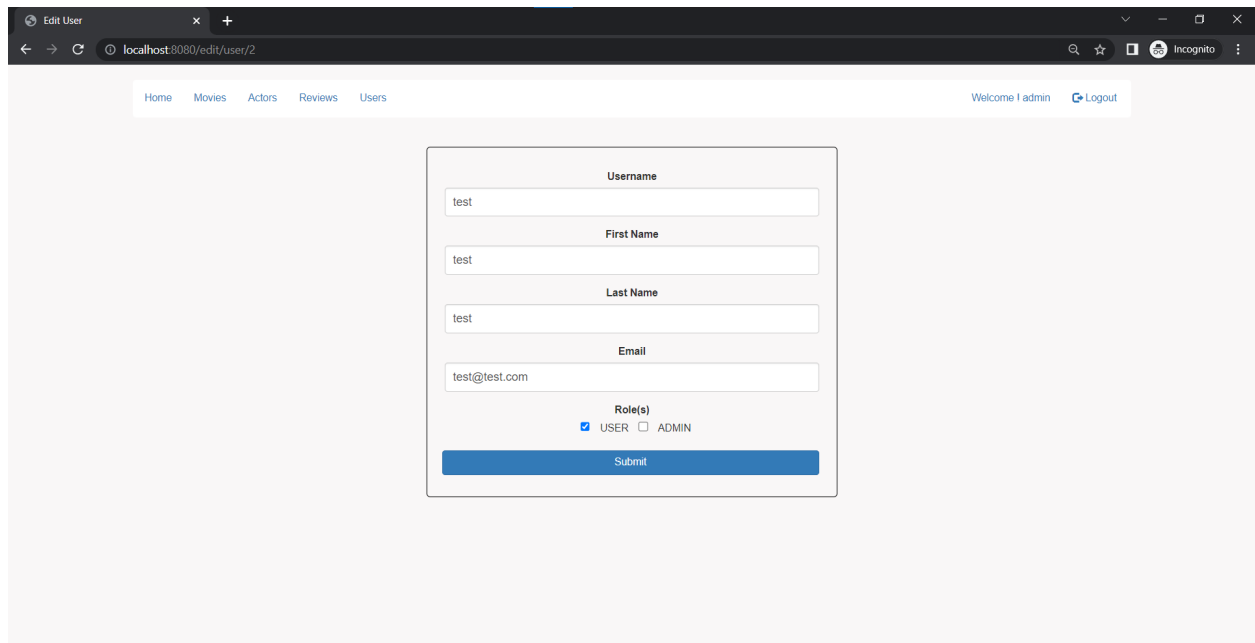
USER

View all users



#	Username	First Name	Last Name	Email	Active	Roles	Actions
1	test	test	test	test@test.com	true	user	Edit Delete
2	test2	test2	test2	test2@test2	true	user	Edit Delete
3	testuser	reviewer	reviewer	tester@tester	true	user	Edit Delete

Update user info/change role



Username

test

First Name

test

Last Name

test

Email

test@test.com

Role(s)

☒ USER ☐ ADMIN

Submit

Individual Contributions :

- ❖ Kushagra : User and Admin Authentication ,Authentication service implementation,AuthController,security,UserDao,User Model,Login and signup Frontend
- ❖ Nikhil : Actor Controller,Actor Service implementation,Actor Dao,Editing actors information Frontend,Actor Model,Adding watchlist
- ❖ Pavan : Movie Controller,movie service implementation,Movie Dao,Movie Model,security,List of movies frontend
- ❖ Pranav : Review controller,review service implementation,Review Dao,Review Model,Role Management,frontend to add movie reviews