System Design for Spheremovie

Group 3

EECS 3311

Murtaza, Karishma, Salan, Allen

Table of Contents

Class Name: Movie
Parent Class (if any): N/A
Subclasses (if any): N/A
Responsibilities:

- Model a document representing a movie stored in the Mongodb collection of movies
- Attributes will include the movie's unique ID, its IMDb ID, its title, the release date, a link to a trailer, a poster, backdrops, and genres
- Keep a reference to associated reviews

Collaborators:

- Review class: it has a list of review objects that are needed to link or keep track of movie reviews.

Class Name: Movie Controller
Parent Class (if any): N/A
Subclasses (if any): N/A
Responsibilities:

- It manages and responds HTTP requests associated with the movie data
- Makes an endpoint to retrieve all the movies and a single movie by the imdb ids

Collaborators:

- Optional: Holds the movie data for when a response is not available
- Spring Framework: for annotations such as @CrossOrigin
- MovieService class: handles the logic necessary to access movie data
- Movie class: holds the movie details provided in the responses

Class Name: Movie Repository
Parent Class (if any): Mongo Repository
Subclasses (if any): N/A
Responsibilities:

- It is an interface that interacts with the Mongodb for the operation to create, read, update, and delete(CRUD) on the Movie class documents
- Implements method to retrieve a movie using the imdb id

Collaborators:

- Optional
- Mongo Repository
- Movie class

Class Name: Movie Service
Parent Class (if any): N/A
Subclasses (if any): N/A
Responsibilities:

- Provides service layer which interacts with the repository for managing the movie data
- Retrieves all the movies from the repository and any specific movie by the imdb id

Collaborators:

- Movie class
- Optional
- MovieRepository class

Class Name: Review
Parent Class (if any): N/A
Subclasses (if any): N/A
Responsibilities:

- This is a document for review which is stored in the collection of mongodb reviews
- Includes a constructor to initialize a review with the time details and content

Collaborators:

- LocalDateTime
- ObjectId
- Movie class

Class Name: Review Controller
Parent Class (if any): N/A
Subclasses (if any): N/A
Responsibilities:

- It manages and responds HTTP requests associated with the movie reviews
- Makes an API endpoint to create new reviews where it receives the review data and associates with a movie using the imdb id

Collaborators:

- Spring Framework
- Map
- ReviewService class
- Review class

Class Name: Review Repository
Parent Class (if any): MongoRepository
Subclasses (if any): N/A
Responsibilities:

- Provides access of database stored in mongodb to the Review documents
- Reviews are handled by the operation to create, read, update, and delete(CRUD)

Collaborators:

- MongoRespository
- ObjectId
- Review class

Class Name: Review Service
Parent Class (if any): N/A
Subclasses (if any): N/A
Responsibilities:

- New reviews are created and saved in the database
- For database operations, the mongodb tools are used
- Adding new reviews using the imdb id to the list of movie reviews

Collaborators:

- MongoTemplate
- Review class
- Update
- LocalDateTime
- ReviewRepository
- Criteria

- Movie

Class Name: Spheremovie Application
Parent Class (if any): N/A
Subclasses (if any): N/A

Responsibilities:

- Starts the sphere movie application
- The spring boot application is started
- Manages HTTP requests with the RestController default

Collaborators:

- Spring Framework
- SpringApplication
- RestController

Software Architecture Diagram

User Interaction

Users interact with the website to perform actions like browsing movies, watching trailers, adding to a watchlist, giving star ratings, searching for specific movie trailers, and selecting a language.

API Requests and Responses

Displays data to the users

Frontend (React JS)

- Search bar functionality for finding specific movie trailers
- Multilingual language options for user preferences
- Adding or removing movie trailers from the watchlist
- Star rating input for reviews
- Smooth page transitions and responsiveness

HTTP requests

Backend (Spring Boot)

- Processing search queries
- Storing or retrieving watchlist data for users
- Storing or retrieving star ratings alongside reviews
- Serving multilingual content based on user language preferences

Makes API call to the backend

Controllers and Services

Stores or Retrieve data

MongoDB (Movies, Reviews, Ratings, Watchlist)

- Movie details and reviews
- User-specific watchlists
- Star ratings linked to reviews
- Language preferences for multilingual content