Application Development Frameworks

Assignment 2 Document

Gearoid Sheehan (R00151523)

For this assignment Spring Boot and JPA were used to create the application. The application revolved around two entities, the film entity and the director entity. One director can have many films. Fragments are used for the header and footer of the html page. The header consists of link to the home page, directors page, films page and edit page. The footer consists of links which alternate the applications language in the html between English and French

In the home page, a simple welcome message is displayed. The directors page returns a list of all the directors in the database in ascending order by their name. Each of these director's names can be clicked, and the link will take the user to a page which displays all the films by that director. The films page showcases all the films in the database with their title, year of release and name of director.

The edit page consists of several links to other sub-pages. The first link to add a director requires an authentication of at least type 'user' to login. Once logged in, the user can enter a name into the input box and add a director of that name to the database. If the add is successful, a success message is displayed. If the director already exists in the database, an error message is displayed. The second link in the edit page also requires authentication of at least type 'user' to login. Once logged in, the user can add a film to the database by entering the new films title, year of release and by selecting a director from the dropdown. Duplicate names of films are allowed as per the specification. They year must be at least 1888 and will show an error message when a lower year is attempted. I did not have time to implement the max year functionality. The third link in the edit page also requires authentication of at least type 'user' to login. Once logged in, the user can select one of the films from the dropdown list and enter a new title for that film in the text input. When the films title is updated successfully, a success message is shown.

The fourth and fifth links require admin access. If logged in under the user credentials already, an error message is shown stating that access is forbidden for the current logged in account, and a link to logout is provided.

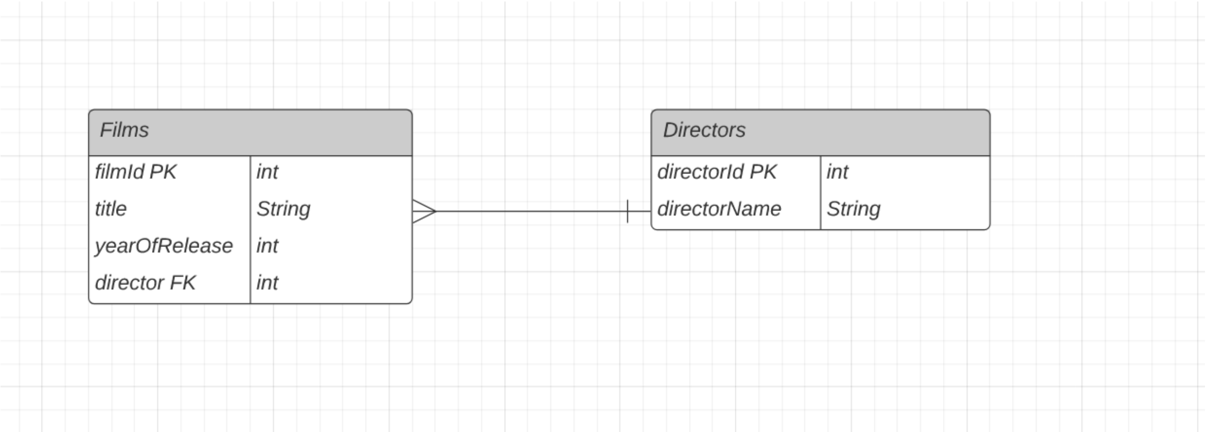
The fourth link goes to a page where a dropdown list is displayed. The user can select a film from this dropdown list and delete that film from the database. When deleted a success message is shown. The fifth link goes to a page which also provides a dropdown. This dropdown includes all the directors in the database, and the user can select one of these directors to delete them and all their films from the database. A success message is shown when done.

It is worth noting I have modified the security so that admins have access to all of the edit features, including those that require 'user' login.

I was unable complete the end of the specification due to time constraints. This feature was the API endpoints. I began creating the second project but did not get very far. It is included in my project upload regardless.

All CSS was done by me. While I managed to get a decent website layout, ideally if I had more time, I would have put more effort into making the page a bit less bland.

Database Diagram



Films	
filmId PK	int
title	String
yearOfRelease	int
director FK	int

Directors	
directorId PK	int
directorName	String