App Development Frameworks

Project 2 - Zac Dair & Aoise Warner

Main Project - localhost:8080 ***API Consumer - localhost:8081***

Application Architecture

The assignment has the following Java packages:

Configurations, Controllers, DAO, Entities, Forms, Service

There are three application files, one main, one for the dev profile which populates the in-memory database with Directors, Films, and users, and a stub for a MySQL implementation of the project.

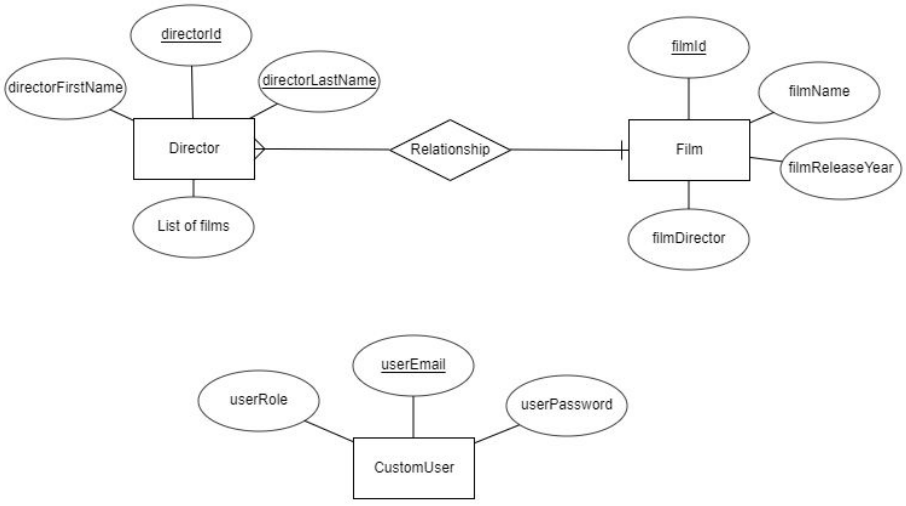
The assignment has the following resources:

static/css/styles.css (overall styling for the project)

*templates/** containing all views required

Three properties files, *application.properties* for general project-wide configurations, *application-dev.properties* for h2 configurations, and finally *application-MySQL* for MySQL configurations.

Database Design



Any user can view a list of the directors in the database in alphabetical order (by surname)	✔️ - Only ordered by surname as required
Any user can select a director from the list to view that director's films	✔️
Any user can view a list of the films in the database in chronological order showing all information you have about that film	✔️
Any user can register by providing an email and password	✔️ - no pattern checking is implemented meaning the user can use a username or email
Authenticated users can add a director to the database	✔️
Authenticated users can add a film providing the film's title, year of release, and a director, which must be selected from a list provided. • Two films can have the same title. • A film's title cannot be left blank. • The film's year of release between 1888 and the current year	✔️ - For validating the current year requirement, this is handled in the service layer, using a calendar object
Authenticated users edit the title of a film • requires that the method's signature created in the repository be annotated as @Transational (facilitating rollback if necessary) and @Modifying	✔️
Administrators can delete a film	✔️
Administrators can delete a director along with their films	✔️
Two REST API endpoints which you write yourself, to allow authenticated users to, • access (in JSON format) films released in a particular year • delete a director (and associated films) given their id	✔️ - The custom API path starts with /myapi/ /film/by/year/{releaseYear} /director/delete/{directorId} Both of which can be consumed using the secondary project (API consumer), with authenticated requests
Internationalisation • Can the user change language? • On all pages or perhaps just some of them?	✔️ - The user can navigate the entire website in both English and French
Form Validation	✔️ - Implemented for all forms including registering, with custom error messages
JPA	✔️ - Queries handled by JPA
Security	✔️