

## CAPSTONE PROJECT

### DIGITAL STREAMING SYSTEM

You are required to write a software solution for a Digital Streaming System which allows administrators to manage the movie catalogue view relevant information about the movies, actors and movie reviews (like Netflix or Amazon Prime). This application stores details of movies, actors and movie reviews. The application has to be implemented using Spring Boot REST API and Microservices Architecture for the backend. The front end is a Single page application with Angular. The DBMS is MySQL.

This software needs to address the following requirements:

#### **Business Requirements:**

The project is for administrators who use the system to perform the following operations.

The Administrator module should allow administrators to:

- A. Login and authentication.
- B. Saving details of movies, viewing, updating and deleting them.
- C. Saving details of actors, viewing, updating and deleting them.
- D. Searching for movies and displaying their details.
- E. View and add movie reviews given by film critics.
- F. New administrators to register.

#### **User Stories:**

##### ***US01: Login***

As an Administrator, I should be able to login to view and manage the movie catalogue.

*Acceptance criteria*

- Verify that the combination of adminId and password being entered by the admin is correct
- Display proper messages in case of errors or exceptions

### ***US02: View Movies***

As an Administrator, I should be able to view the available movies.

#### *Acceptance criteria*

- The movie details should be displayed in a tabular format
- If the number of movies is more than 10, then display the movie details in paginated format
- Display proper messages in case of errors or exceptions

### ***US03: Add Movies to Catalogue***

As an Administrator, I should be able to add new movies released to be displayed on the movie dashboard.

#### *Acceptance criteria*

- Movie details include image, movie title, actor, cost and year of release.
- The movie id should be autogenerated.
- Display proper messages in case of errors or exceptions

### ***US04: Modify and Delete Movies***

As an Administrator, I should be able to modify movie details and delete old movies to keep the movie dashboard updated.

#### *Acceptance criteria*

- Only the image and cost can be modified.
- Movies older than a year can be deleted.
- Display proper messages in case of errors or exceptions

### ***US05: View Actors***

As an Administrator, I should be able to view the actor details.

*Acceptance criteria*

- The actor details should be displayed in a tabular format
- Display proper messages in case of errors or exceptions

***US06: Add Actors***

As an Administrator, I should be able to add new actor details.

*Acceptance criteria*

- Actor details include first name, last name, gender and age.
- The actor id should be autogenerated.
- Display proper messages in case of errors or exceptions

***US07: Modify and Delete Actors***

As an Administrator, I should be able to modify and delete actor details.

*Acceptance criteria*

- Actors should be deleted only if there are no movie details for the actor.
- Display proper messages in case of errors or exceptions

***US08: View Movie Reviews***

As an Administrator, I should be able to view the available movies.

*Acceptance criteria*

- The movie review should be displayed in a tabular format
- If the number of movies is more than 10, then display the movie details in paginated format
- Display proper messages in case of errors or exceptions

***US09: Add Movie Reviews***

As an Administrator, I should be able to add movie reviews that are collected from movie critics.

*Acceptance criteria*

- Review details should include description, posted date and rating.
- The review id should be autogenerated.
- Display proper messages in case of errors or exceptions

***US10: New Admin Registration***

As a New Administrator, I should be able to register myself to keep track and work on the movie catalogue.

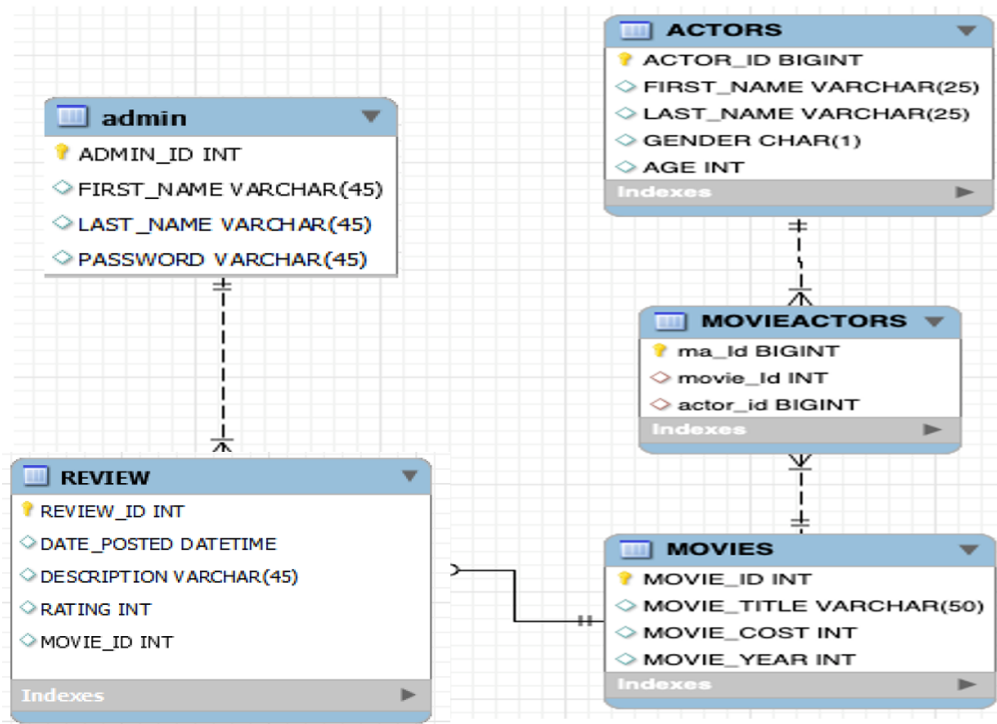
*Acceptance criteria*

- Verify that an error message is displayed if all the details are not filled
- Verify that an error message is displayed if the email id is already in use by some other user
- Verify that an error message is displayed if the name contains numbers or special characters
- Password should be a combination of at least an uppercase alphabet, lowercase alphabets, a digit, and a special character.
- Verify that an error message is displayed if the password entered by the customer is not in the expected format
- Verify that an error message is displayed if the phone number is already in use by some other user
- Verify that a new account is created if all the fields that have been entered by the user are proper.
- The user should be able to login with his/her emailId and password

**Design Documents:**

- Create the actual ER Diagram that mirror implementation of your own solution.

*Sample ER Diagram for reference*



### Functional Requirements:

- The Application should be implemented using Spring boot and Microservices architecture for the back end and a Single Page Application built with Angular as the front end.
- The microservices should be registered in a Service Registry.
- You should provide access to all microservices using a dynamic API gateway.
- Perform client-side load balancing of microservices.
- The REST API's should be tested with Postman tool.
- Perform ORM using the Spring Data JPA and Hibernate framework.
- The Angular App should use service layer to invoke REST APIs implemented with Spring boot.
- The backend to be used is MySQL.
- Appropriate validations and exception handling should be performed in the front end and the backend.

## **Best Practices**

- A. Follow proper naming conventions and ensure code quality
- B. Create Javadoc comments to explain the purpose of the class and behaviours.
- C. Use StringBuilder for concatenating large strings
- D. Enforce Compile-Time Type Checking using Generics
- E. Validate Parameter Preconditions
- F. Proper handling of NullPointerException and other relevant exceptions.
- G. Perform validations across the front-end and backend.
- H. Follow Angular Coding Style Guide.
- I. The code should have modularity.

## **Submission Requirements:**

- A. Submit a single archive containing all the artefacts
- B. The artefacts are Project Source Code, Presentation, Postman Test Collection and Database Scripts.

## **Project Presentation Requirements:**

- A. The Project should be presented individually with execution and presentation.
- B. The developer should be able to edit the code live and demonstrate as needed by the evaluator.
- C. Explain the details of API used, processes involved, workflows and frameworks utilized.