

Problem Statement:

ExpressMovies is an application to maintain the details of released and upcoming movies. This application is helpful for knowing details about movies and their respective directors. The application requires different CRUD operations to be performed using Spring Data JPA to meet the given functionalities.

The following functionalities have to be implemented in the application:

- Add movie
- Search movie and director details
- View movies
- Update movie and director details
- Remove movie

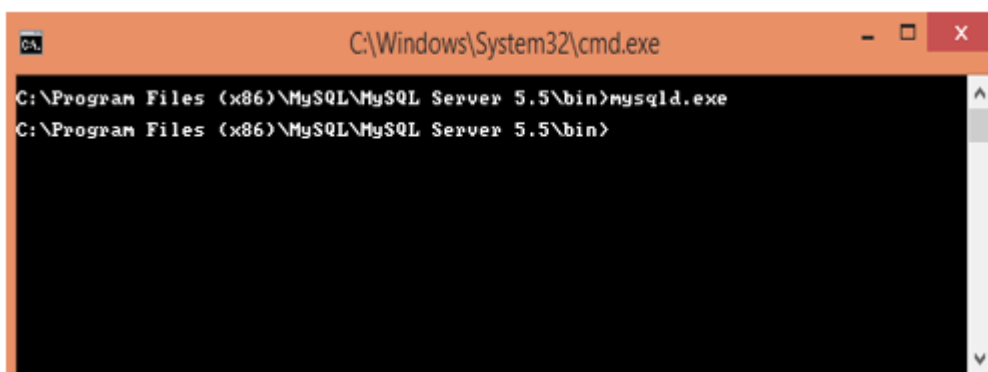
Software Requirement

1. [Spring Tool Suite\(STS\)](#) / [Eclipse IDE](#)
2. [MySQL](#)

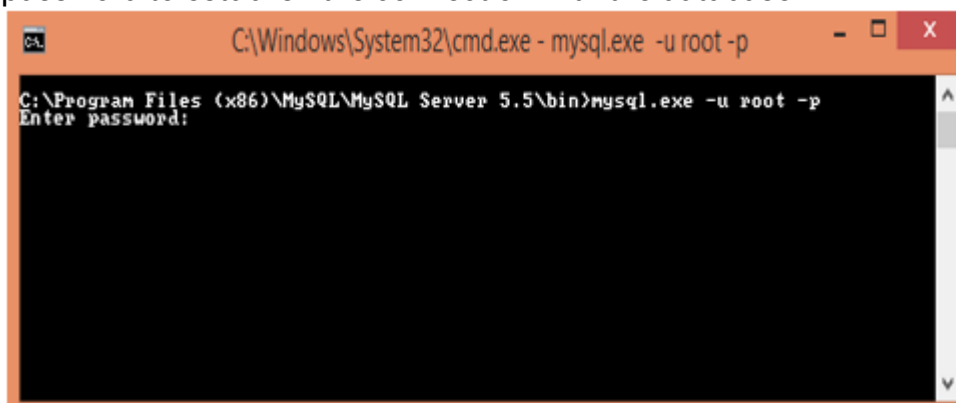
Project Setup

MySQL Database setup:

- Open the command prompt and navigate to the folder which has the MySQL server bin folder.
- Enter the command `mysqld.exe` to start the MySQL Server



- Once MySQL server is started, type the command `mysql.exe -u root -p` and key in the MySQL password to establish the connection with the database.



Your database is now ready to use. Make sure your MySQL server instance is open while running your application.

Project Implementation:

1. Database Design: Application schema and tables

- Create database schema MovieDetails in the MySQL database as shown below:

CREATE database MovieDetails;

- Create the following tables in the MovieDetails schema

Note: Ignore table creation, if your application is configured to auto-create tables using Hibernate/JPA properties

i. Create **Movie** Table

Column Name	Data Type
Movie_Id	int(Primary Key)
Movie_Title	varchar(100)
Date_Released	Date
Movie_Running_Time	DateTime

ii. Create **Director** Table

Column Name	Data Type
Director_Id	int(Primary key)
First_Name	varchar(100)
Last_Name	varchar(50)
Address	varchar(150)
Contact_Number	int
Email	varchar(100)

iii. Create **Movie_Director** Table

Column Name	Data Type
Director_Id	int(Foreign key reference to Director table)
Movie_Id	int(Foreign key reference to Movie table)

2. Application Functionalities

Implement the following functionalities as menu-based console application using Spring Data JPA with Spring Boot provide application's configuration metadata:

- Add movie
 - Insert new Movie and Director details into the database.

Note: Verify the records inserted in the database (Movie, Director and Movie_Director Table)

- Search movie and director details
 - List movies based on the movie title
 - Display movies based on the title, if not present, then display exception with the message "Invalid Movie title".
 - List movies based on director name
 - Display movie based on the director's first and last name, if the director's name is not present, then display exception with the message "Invalid Director name".
 - List director's details based on the movie title
 - Display director's details based on the movie title, if not present, then display exception with message "Invalid Movie title"
- View movies
 - Display all the movies
- Update Movie and Director details
 - Update new release date based on the movie title
- Update the new release date for the existing movie and display appropriate error messages for invalid details.
 - Update director details based on director first and last name
- Provide the director's first name and last name to update his new address, contact number and display appropriate error messages for the invalid details.
- Remove movie based on movie title

- Enter movie title to remove a movie, if the movie with the given title is not present, then display exception message as "Movie with the given title is not present"

Note- Verify the records deleted from the database tables(Movie, Director and Movie_Director Table)

Instructions for Evaluation

Refer to below instructions for getting the project evaluated by your SMEs from project or account

Assessment Instructions:

Components	Marks
Application Configuration and Implementation	40
Exception Handling	5
Code quality	5
Total	50