PROJECT SYNOPSIS

Project Title-

Rule based Movie Recommender System

Team Members:

S.No.	Name	AFID
1.	Ishita Sharma	AF04990651
2.	Chetna	AF04991264
3.	Divyam Gupta	AF04991712

Guided by:

Mr. Anuj Kumar

Batch Code: ANP-D2405

Course Code: ITPR

Introduction

In the modern world, users are burdened with an immense number of choices in regard to entertainment content. Finding a relevant movie or book often requires significant time and effort. This project, the Rule-Based Movie Recommender System, is a CLI application developed using Java and MySQL.

Unlike a simple management system, which would only store data, this application adds a layer of "intelligence" to it by analyzing user ratings and preferences. It allows users to rate content and provides personalized suggestions based on certain logical rules considering genre preference and global popularity. It serves as an excellent demonstration of DBMS, JDBC connectivity, and algorithmic logic in a Java environment.

Objectives

The primary objectives of this project are:

- To design a persistent storage system for movies, books, users, and their interactions (ratings) using **MySQL**.
- To implement **JDBC (Java Database Connectivity)** for seamless communication between the Java application and the database.
- To develop a **Rule-Based Algorithm** that filters and sorts data to provide meaningful recommendations.
- To create a user-friendly CLI menu for managing data (**CRUD** operations) and viewing reports.

Project Category

RDBMS (Relational Database Management System) & Desktop Application Development**.**

(Sub-category: Algorithmic Data Filtering)

Analysis

i) Modules and Description:

The project is divided into four distinct modules:

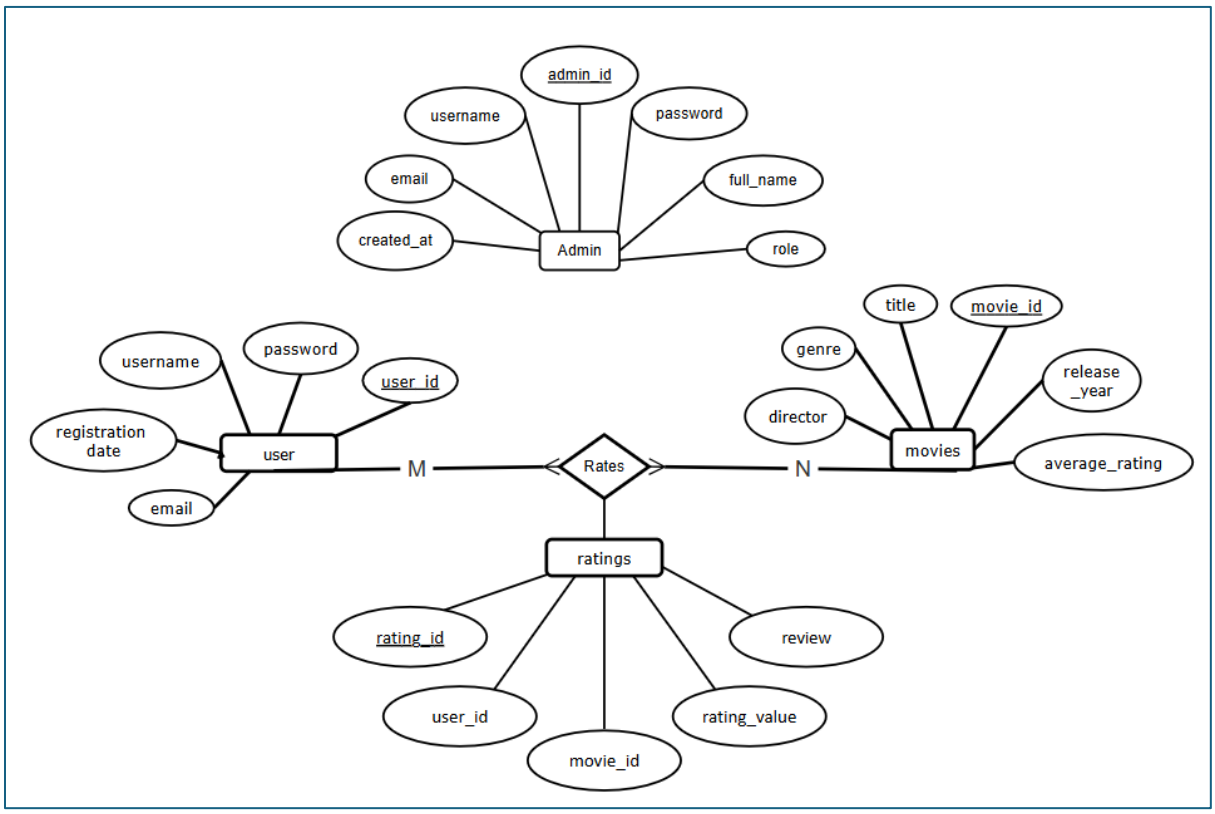
1. **Admin Module:** Responsible for system maintenance. Allows the administrator to add, update, or delete movies from the catalogue.
2. **User Module:** Handles user registration and login. Allows users to search for items and manage their profiles.
3. **Rating Module:** Captures user feedback. Users can assign a rating (1-5 stars) to any movie. This data forms the backbone of the recommendation engine.
4. **Recommendation Module:** The core intelligence of the system. It executes SQL queries to fetch:
    - Top-rated items globally.
    - Personalized suggestions based on the user's favourite genre.

ii) Database Design:

The system uses a normalized Relational Database with four main tables:

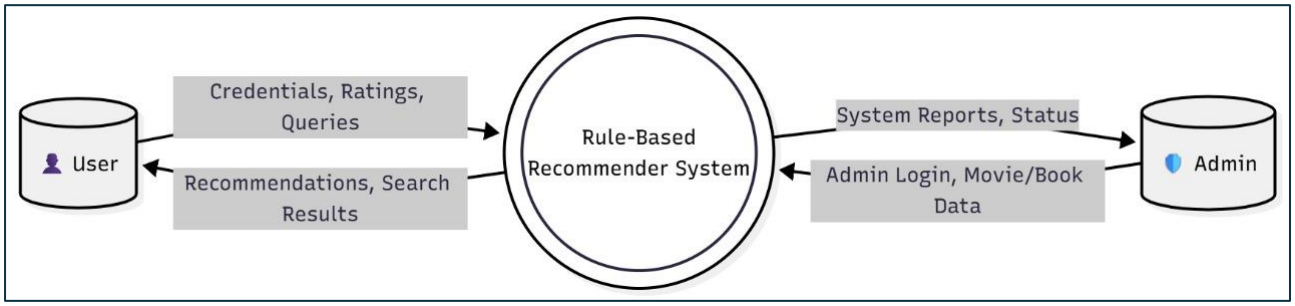
1. **movies**: Stores details of content (ID, Title, Genre, Type, Year).
2. **users**: Stores user profiles (ID, Name).
3. **ratings**: A junction table linking Users and Items (ID, User_ID, Item_ID, Rating).
4. **admins**: Stores administrator credentials (ID, Username, Password) for authorized access to the Admin Module.

iii) Entity Relationship (ER) Diagram:

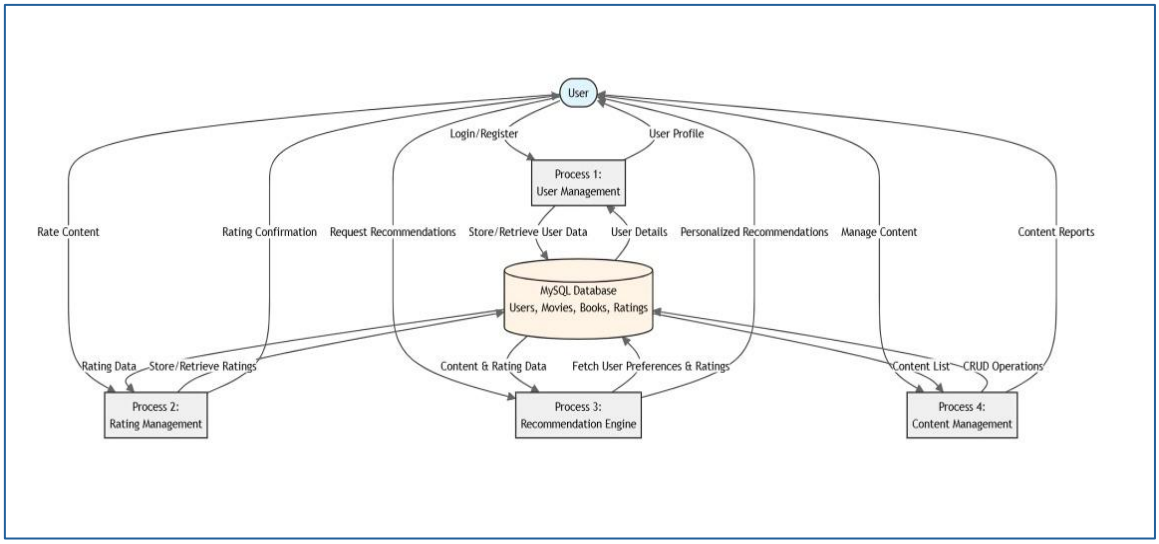


iv) Data Flow Diagram (DFD)

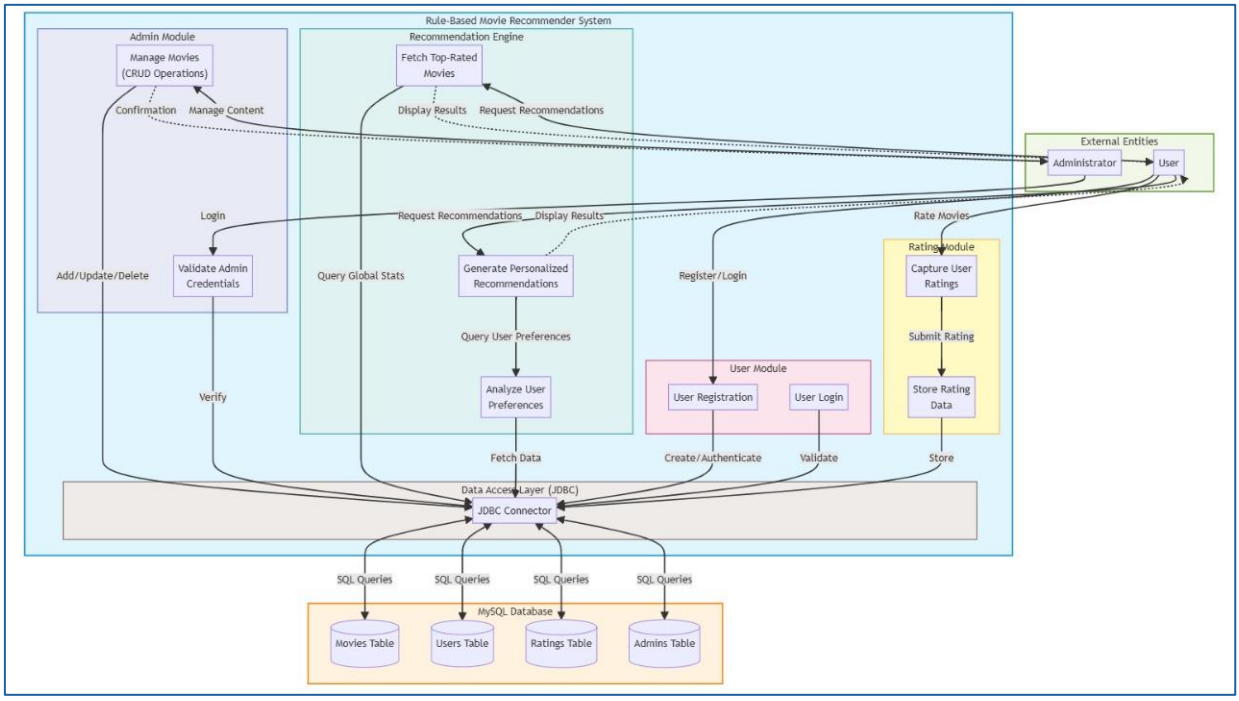
a) Zero level DFD



b) 1st level DFD

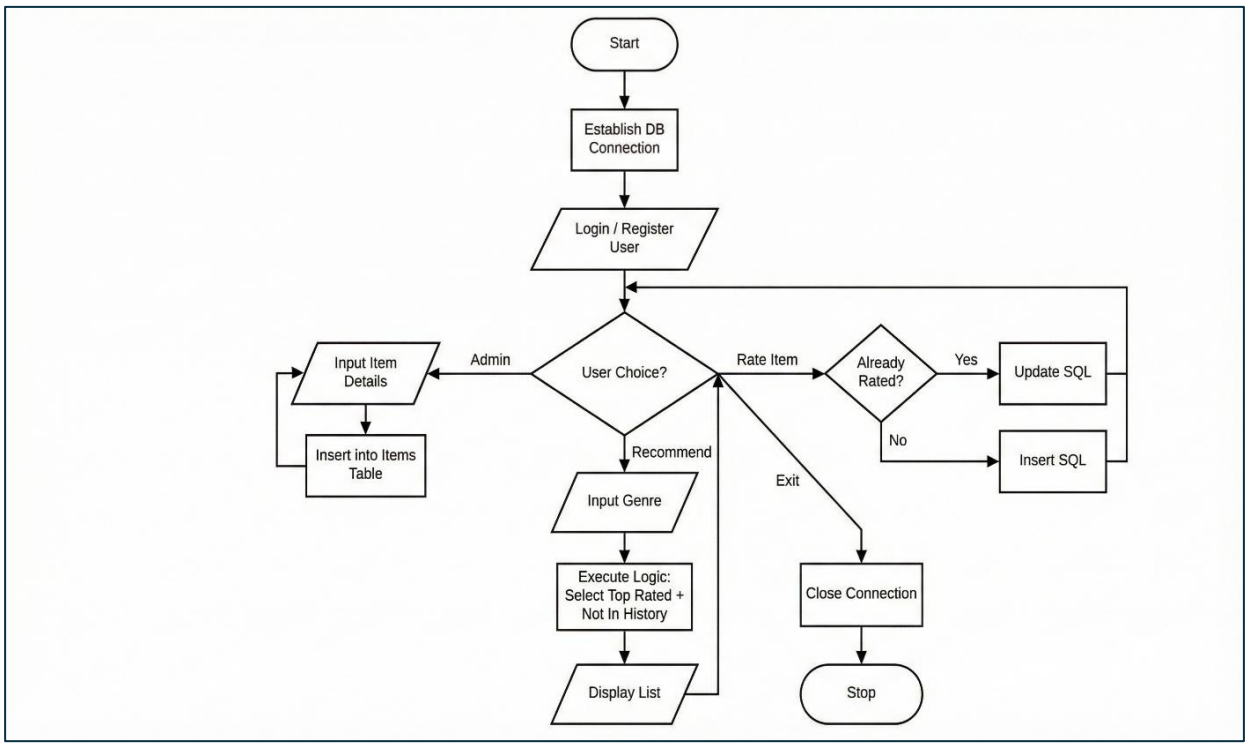


c) 2nd level DFD



Complete Structure

Process Logic Diagram:



Platform Used

Hardware Requirement

- Processor: Intel Core i3 or higher (or equivalent AMD).
- RAM: 4GB or higher.
- Storage: 500MB free disk space.

Software Requirement

- Operating System: Windows 10/11, Linux, or macOS.
- Language: Java (JDK 8 or higher).
- Database: MySQL Server (8.0 recommended).
- Development Tool (IDE): Eclipse IDE
- Drivers: MySQL Connector/J (JDBC Driver).

Future Scope

- **Web Interface:** The project can be migrated to a web-based platform using JSP/Servlets or Spring Boot.

- **Machine Learning:** The rule-based logic can be replaced with Collaborative Filtering (Matrix Factorization) for higher accuracy using Python or Java ML libraries.

- **API Integration:** Fetching real-time movie data and posters using the OMDB or IMDB API.

- **Graphical User Interface (GUI):** Implementing a desktop UI using JavaFX or Swing.

Bibliography

- Oracle Java Documentation (docs.oracle.com)
- MySQL Reference Manual (dev.mysql.com)
- GeeksForGeeks (JDBC Tutorials)
- Eclipse IDE documentation