

Project Proposal

on

Theatre Ticket Booking System

Guided By-Mr. Anuj Kumar

Created By-

Name-Yuvraj patel

AFID-AF04990756

Batch Code-ANP-D2406

Course Code-ITPR

Name-Shreyash Maurya

AFID- AF04990621

Batch Code-ANP-D2406

Course Code-ITPR

Movie & Theatre Ticket Booking System using Java, JDBC, and MySQL (CLI-Based Application)

Introduction

The Movie & Theatre Ticket Booking System is a command-line application developed using **Java**, **JDBC**, and **MySQL**. The project aims to replicate the core functionalities of real-world multiplex and online ticketing platforms such as BookMyShow, PVR Cinemas, and INOX, but with a simplified and educational design suitable for offline and academic use.

This system allows users to browse movies, explore show timings, view screens with auto-generated seats, and book tickets seamlessly. The admin panel provides capabilities to manage movies, halls, screens, and shows in a structured manner. Advanced features such as **automatic seat creation**, **booking cancellation**, **refund processing**, **user booking history**, **detailed booking confirmation**, and an **admin revenue dashboard** make the system realistic and feature-rich.

The application's architecture follows a modular layered design:

- **Model Layer** represents data objects.
- **DAO Layer** handles database interactions.
- **Util Layer** manages database connectivity.
- **Main Controller** orchestrates menus, logic, and user interactions.

All functionalities operate through Java's CLI (Command Line Interface), making the system lightweight, platform-independent, and easy to run in academic environments without GUI frameworks.

This project demonstrates the integration of database-driven applications using JDBC, proper SQL schema design with relational connections, clean object-oriented structure, and practical concepts like transactions, validation, reporting, and structured workflow logic.

Objective

The primary objective of this project is to build a fully functional **console-based movie ticket booking system** that showcases real-world software engineering concepts such as database connectivity, data management, multi-user flow, and admin-controlled configuration.

Specific objectives include:

1. To simulate real-world booking workflows

The project recreates the complete cycle of theatre management:

- a. Adding movies
- b. Creating halls and screens
- c. Scheduling shows
- d. Generating seats automatically
- e. Allowing users to browse and book tickets
- f. Generating tickets and storing booking records

2. To implement structured database management using MySQL & JDBC

The system uses:

- a. Multiple related tables (movies, screens, shows, bookings, payments, tickets, seats, users)
- b. Foreign key relationships
- c. SQL joins for reporting, history, and real-time seat availability

- d. Safe insertion and query patterns through DAOs

3. **To demonstrate OOP design and modularity**

Each entity is modeled as a class, while DAOs handle database interactions.

This separation improves readability, maintainability, and scalability.

4. **To enhance user experience with meaningful features**

- a. Auto seat generation upon creating a screen
- b. Booking confirmation with movie, hall, screen & showtime details
- c. Booking cancellation with refund calculation rules
- d. User booking history
- e. Admin revenue dashboard (show-wise, movie-wise, total revenue, ticket count)

5. **To provide a strong academic example**

The project serves as a comprehensive demonstration of:

- a. Java programming
- b. JDBC connectivity
- c. SQL database design
- d. CLI-based interaction
- e. Real-world business logic simulation

6. **To create a foundation for future expansion**

This system can easily evolve into:

- a. A GUI-based app
- b. A web application
- c. A microservice
- d. A full stack web portal

Project Category

Database Management & Application Development Using Java (JDBC)

This project falls under the following academic categories:

- **Database Application Project**

The system heavily relies on MySQL relational models, CRUD operations, SQL joins, transactions, and reporting.

- **Core Java Application**

Built using Java 8+ principles such as OOP, exception handling, collections, modularity, and command-line interaction.

- **JDBC-Based CRUD & Transaction System**

The project demonstrates professional-level JDBC usage:

- Prepared statements
- Connection pooling (via utility class)
- Error handling
- Multi-table queries

- **Software Engineering & System Design**

Architecture follows DAO pattern, layered code structure, and real-case logical flows.

- **Console-Based Management System**

A full end-to-end ticketing system without GUI, ideal for academic demonstration.

Analysis

➤ Modules and Description

The Movie & Theatre Ticket Booking System is divided into multiple modules, each responsible for a specific domain of the application. This modular design makes the project scalable, maintainable, and easier to understand.

1. Admin Management Module

This module handles all administrative operations required to configure the theatre system.

Key Functions:

- Admin login authentication
- Adding new movies
- Creating halls (theatre branches)
- Creating screens under each hall
- Auto-generation of seats for each screen
- Scheduling movie show timings
- Viewing lists of movies, halls, screens, and shows
- Accessing the revenue dashboard

Description:

The admin ensures that users always see updated movies, screens, and showtimes. The system connects to MySQL to store and fetch movie details, hall info, screen capacity, and schedules. The admin is the core controller for the entire booking environment.

2. User Management Module

Handles user-related activities such as registration, login, and viewing personal booking records.

Key Functions:

- Creating a new user account
- Login authentication
- Maintaining user profile
- Viewing booking history

Description:

This module ensures secure login and personalized experience for each user. Every user action is tracked and stored in multiple tables, enabling features like booking history and cancellation reporting.

3. Movie Management Module

Admin-controlled module to maintain movie information.

Key Functions:

- Adding movie title, duration, language, and description
- Listing all movies

Description:

Stores movie metadata in the database. Movies act as parent entities linked to shows and bookings.

4. Hall & Screen Management Module

Handles the structure of the theatre complex.

Key Functions:

- Adding halls (theatre branches)
- Adding screens under halls
- Auto-generating seats for each screen
- Listing available halls & screens

Description:

Screens belong to halls. Each screen has a configurable number of seats. Seat generation ensures each screen gets proper labeled seats (A1, A2... etc) with seat types (REGULAR / PREMIUM).

5. Show Scheduling Module

Controls when and where movies are played.

Key Functions:

- Assign movie to screen
- Set show timings
- Set ticket price
- List all upcoming shows

Description:

Each show is linked to a movie + screen + time. Users book tickets based on available shows.

6. Ticket Booking Module

This is the core interactive module for users.

Key Functions:

Listing available shows

- Showing available seats
- Selecting seats
- Creating booking entry
- Creating tickets
- Payment simulation
- Booking confirmation email-style summary
- Automatic seat blocking

Description:

This module performs seat selection, booking creation, payment, and ticket generation. All data is inserted into bookings, tickets, and payments tables.

7. Payment & Refund Module

Handles all financial transactions.

Key Functions:

- Payment via CARD, CASH, or UPI
- Recording payment in database
- Applying refund rules based on how close to the show time
- Creating refund transaction as negative payment

Description:

Supports partial and full refunds:

- 24 hours → 100%
- 3-24 hours → 50%
- <3 hours → 0%

8. Booking History & Cancellation Module

Allows users to manage bookings after purchase.

Key Functions:

- Viewing past and ongoing bookings
- Showing ticket seat numbers for each booking
- Cancelling future bookings
- Automatic ticket status update
- Refund calculation

Description:

This module ensures transparency and user control. It adds convenience features similar to BookMyShow.

9. Admin Revenue Dashboard Module

Provides financial insights.

Key Functions:

- Total revenue
- Movie-wise revenue
- Show-wise earnings

- Ticket sales count
- Booking status distribution

Description:

This module helps administrators understand earning performance and manage business decisions through data analysis.

➤ Database Design

The database design follows **relational schema principles** and uses **8+ normalized tables** with proper foreign keys to maintain data integrity. The system uses MySQL for storage and JDBC for access.

1. Users Table

Stores customer accounts.

Column	Type	Description
id	INT	Unique user ID
name	VARCHAR	User full name
email	VARCHAR	Login credential
password	VARCHAR	Account password
phone	VARCHAR	Contact number

2. Admin Table

Stores admin login credentials.

Column	Type	Description	Key
id	INT	Unique admin ID	Primary Key (PK)
name	VARCHAR	Admin full name	
email	VARCHAR	Login credential	
password	VARCHAR	Account password	

3. Movies Table

Represents movies available for screening.

Column	Type	Description	Key
id	INT	Unique movie ID	Primary Key (PK)
title	VARCHAR	Movie title	
duration_minutes	INT	Movie runtime	
language	VARCHAR	Movie language (e.g., English, Hindi)	
description	TEXT	Plot summary or synopsis	

4. Halls Table

Represents theatre branches.

Column	Type	Description	Key
id	INT	Unique hall/branch ID	Primary Key (PK)
name	VARCHAR	Name of the theatre/branch	
location	VARCHAR	Physical location or address of the hall	

5. Screens Table

Represents screens under halls.

Column	Type	Description	Key
id	INT	Unique screen ID	Primary Key (PK)
hall_id	INT	Hall this screen belongs to	ForeignKey (F--> halls.id)
name	VARCHAR	Name of the screen (e.g., Screen 1, Gold Class)	
total_seats	INT	Total number of seats in this screen	

6. Seats Table

Auto-generated seats for each screen.

Column	Type	Description	Key
id	INT	Unique seat ID	Primary Key (PK)
screen_id	INT	Screen this seat is located in	Foreign Key (FK--> screens.id)
seat_label	VARCHAR	The alphanumeric label (e.g., A1, B5)	
seat_type	VARCHAR	Type of seat (REGULAR / PREMIUM)	
is_active	TINYINT	Status (1=Active, 0=Inactive/Broken)	

7. Shows Table

Represents movie screenings.

Column	Type	Description	Key
id	INT	Unique show ID	Primary Key (PK)
movie_id	INT	Movie being screened	Foreign Key (FK--> movies.id)
screen_id	INT	Screen where the movie is shown	Foreign Key (FK --> screens.id)
show_time	DATETIME	Date and time of the screening	
price	DECIMAL	Base price of a ticket for this show	

8. Bookings Table

Records booking transactions.

Column	Type	Description	Key
id	INT	Unique booking ID	Primary Key (PK)
user_id	INT	Customer who made the booking	Foreign Key (FK --> users.id)
show_id	INT	Show that was booked	Foreign Key (FK --> shows.id)
total_amount	DECIMAL	Final total amount charged	
status	VARCHAR	Booking status (PENDING / CONFIRMED / CANCELLED)	
booking_time	TIMESTAMP	Time the booking was created (Default: CURRENT_TIMESTAMP)	

9. Tickets Table

One row per seat booked.

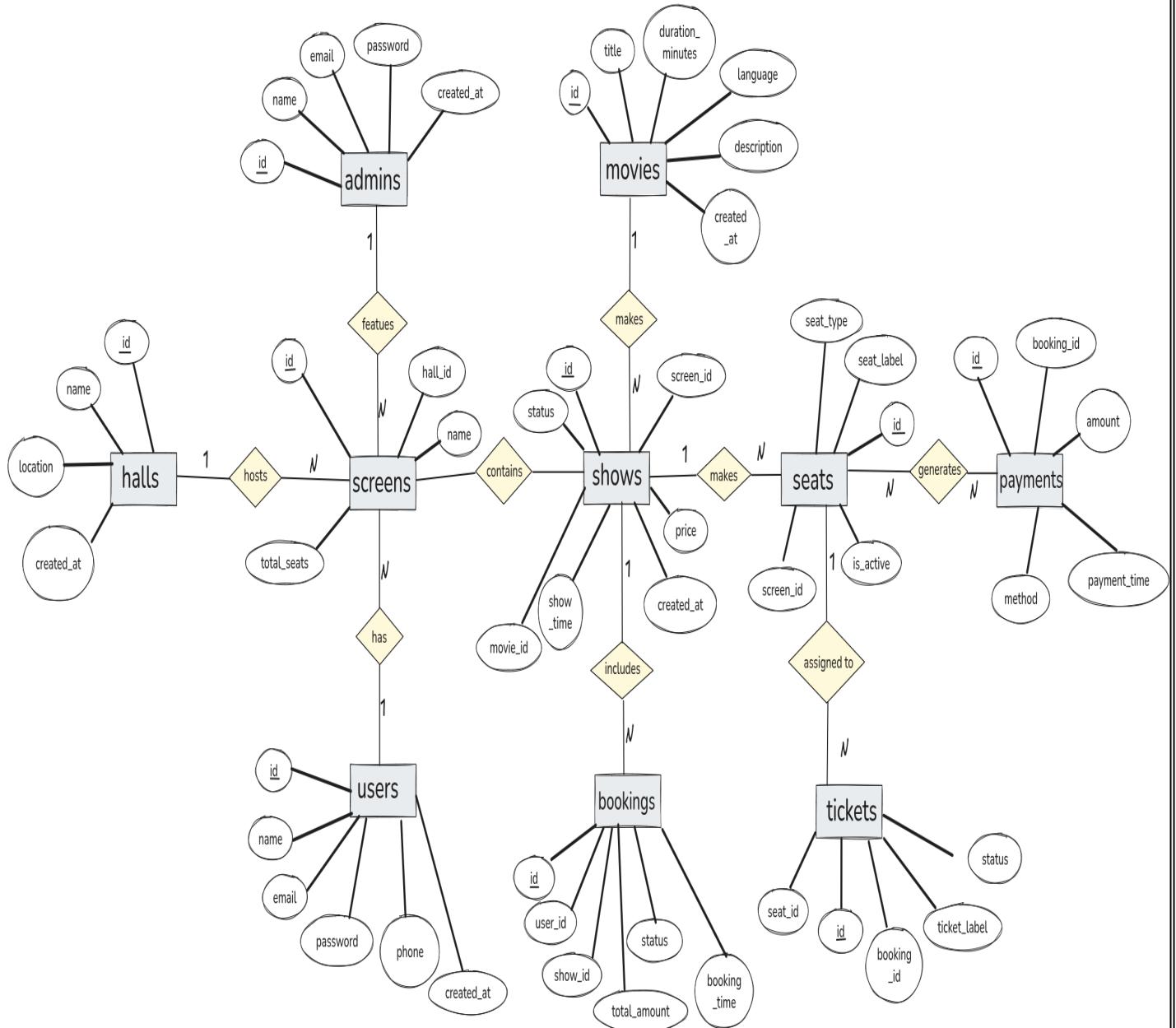
Column	Type	Description	Key
id	INT	Unique ticket ID	Primary Key (PK)
booking_id	INT	Booking this ticket belongs to	Foreign Key (FK --> seats.id)
seat_id	INT	The specific seat booked	Foreign Key (FK --> seats.id)
ticket_label	VARCHAR	Generated ticket identification code	
status	VARCHAR	Ticket status (ACTIVE / CANCELLED)	

10. Payments Table

Stores payment + refund transactions.

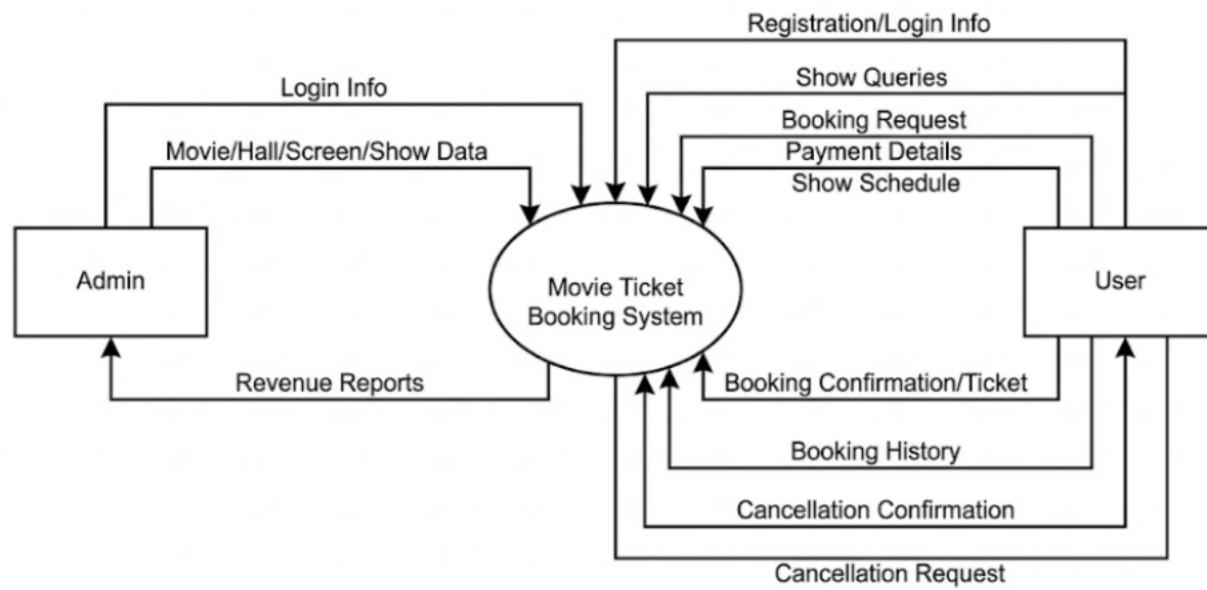
Column	Type	Description	Key
id	INT	Unique payment ID	Primary Key (PK)
booking_id	INT	Booking the payment is for	Foreign Key (FK--> bookings.id)
amount	DECIMAL	Transaction amount (negative for a refund)	
method	VARCHAR	Payment method (UPI / CARD / CASH / REFUND)	
status	VARCHAR	Transaction status (SUCCESS / FAILED)	
payment_time	TIMESTAMP	Time the payment was recorded (Default: CURRENT_TIMESTAMP)	

> ER Diagram

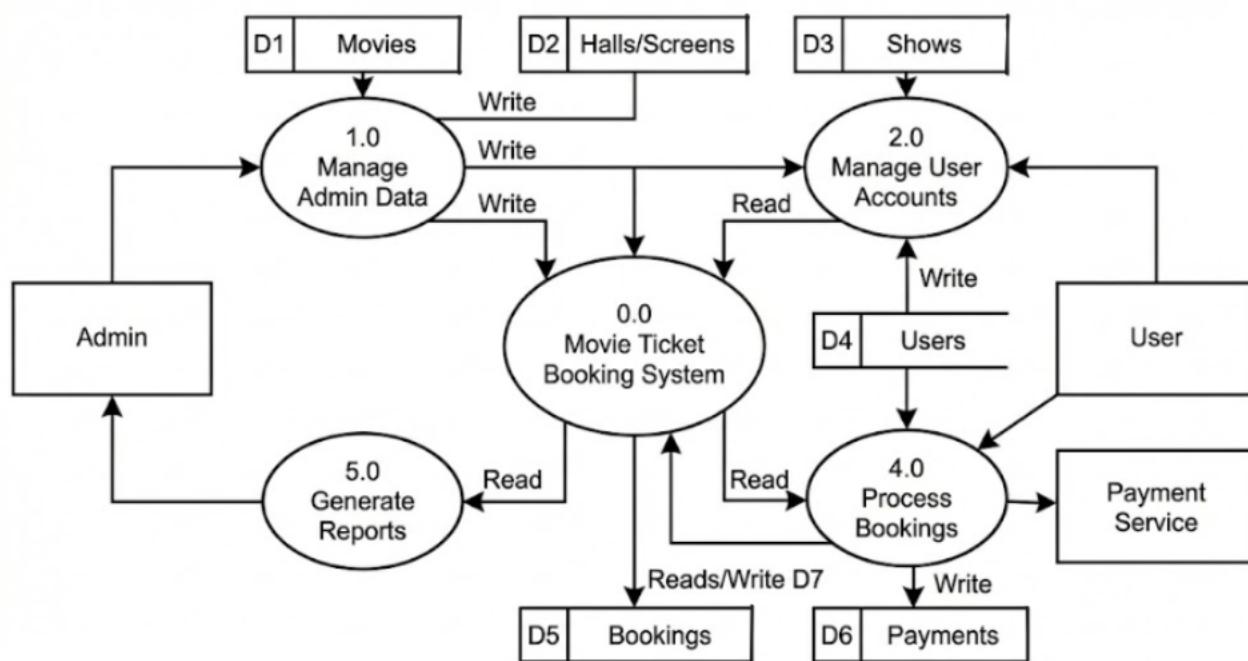


> Data Flow Diagram

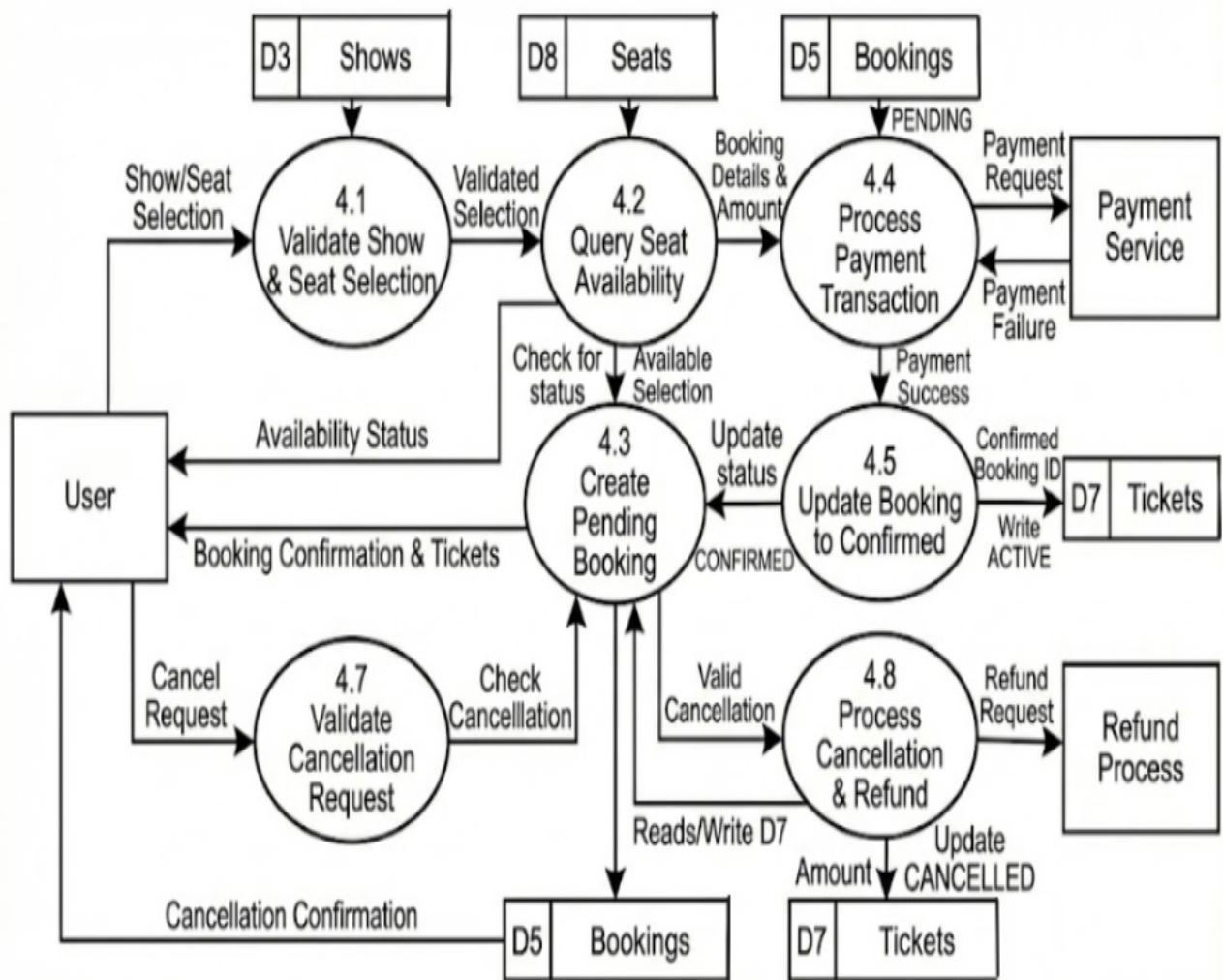
Data Flow Diagram Level 0



Data Flow Diagram Level 1



Data Flow Diagram Level 2



Complete Structure

The Movie & Theatre Ticket Booking System follows a **layered architecture**, ensuring clean separation of concerns, modular design, and easy maintainability. The structure is divided into four primary layers: **Model**, **DAO**, **Utility**, and **Main Controller**. These layers work together to deliver a complete end-to-end booking experience through a robust command-line interface.

1. Model Layer (Data Structure Layer)

This layer contains Java classes that represent database entities. Each class models a real-world component such as movies, halls, screens, bookings, payments, etc.

Entities:

- Admin
- User
- Movie
- Hall
- Screen
- Seat
- Show
- Booking
- Ticket
- Payment

Purpose:

This layer ensures:

- Data encapsulation
- Type safety
- Clean mapping between objects and database tables
- Easy passing of data across system modules

2. DAO Layer (Data Access Layer)

The DAO (Data Access Object) layer handles all communication with the **MySQL database**. Every CRUD operation (Create, Read, Update, Delete) is implemented here using **JDBC prepared statements**.

Primary DAO Classes:

- AdminDAO
- UserDAO
- MovieDAO
- HallDAO
- ScreenDAO
- SeatDAO
- ShowDAO
- BookingDAO
- TicketDAO
- PaymentDAO

Responsibilities:

- Secure interaction with the database
- Managing foreign key relations (e.g., bookings → shows → movies)
- Executing SELECT/INSERT/UPDATE queries
- Ensuring error handling for SQL exceptions

This structure supports scalable, reusable database logic.

3. Utility Layer

This layer contains helper classes used system-wide.

The core utility class:

DBUtil.java

- Establishes MySQL connection
- Loads JDBC driver
- Returns connection objects to DAO classes
- Ensures centralized and maintainable DB credentials

This layer acts as the backbone for all database operations.

4. Controller Layer (Main.java)

The **Main** class controls the entire application flow.

Handles:

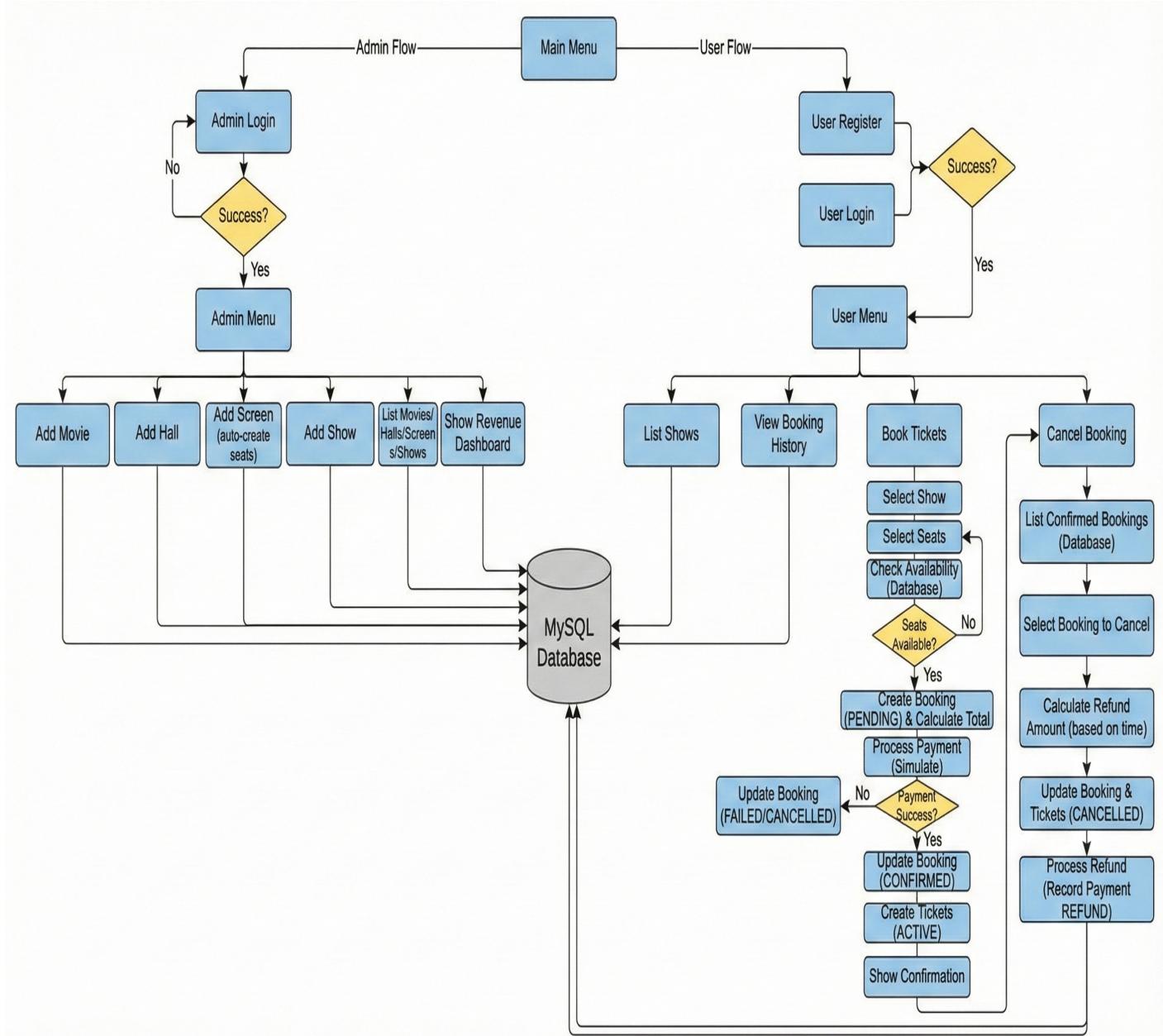
- Admin menus (add movie, hall, show, etc.)
- User menus (login, booking, cancellation, history)
- Seat availability and seat selection
- Booking creation and payment workflow
- Refund calculations
- Reporting dashboard for admin
- CLI-based input/output operations

UI Flow:

- Main Menu → Admin/User workflows
- Admin operations → database update
- User operations → booking → payment → ticket generation

The controller layer is the “brain” of the system, directing how each module interacts.

➤ Process Logical Diagram



Platform Used

The Movie & Theatre Ticket Booking System is built using a simple yet powerful technology stack designed for portability, reliability, and ease of execution in academic environments. The system runs entirely from the command line, making it suitable for low-resource machines and offline laboratory environments.

Platform Overview:

The project is executed on a **Java Runtime Environment (JRE)** with a **MySQL relational database** connected through the **JDBC (Java Database Connectivity)** API. The application follows a modular architecture and runs as a standalone console program without requiring any external UI libraries or servers.

➤ Hardware Requirement

The system is lightweight and can run on any standard computer used in computer science laboratories. Minimal hardware specifications include:

Minimum Requirements:

- **Processor:** Dual-Core CPU (Intel/AMD)
- **RAM:** 2 GB
- **Storage:** 200 MB free disk space
- **Display:** Standard text-compatible terminal

- **Keyboard:** Required for CLI inputs

Recommended Requirements:

- **Processor:** Intel i3 or higher
- **RAM:** 4 GB or more
- **Storage:** 1 GB free disk space
- **Operating System:** Linux / Windows / macOS
- **Stable MySQL installation**

Since this project is command-line based, it runs efficiently even on older machines with very low hardware resources.

➤ Software Requirement

1. Operating System

- Windows 7/8/10/11
- Linux (Ubuntu / Fedora / Arch / Debian)
- macOS

2. Development Tools

- **Java Development Kit (JDK 8 or above)**
Required to compile and run Java code.

- **MySQL Server (8.0 or above)**
Used as the relational database backend.
- **MySQL Connector/J (JDBC Driver)**
Required for connectivity between Java and MySQL.
- **Terminal / Command Prompt / Bash Shell**
Used for executing the CLI application.

3. Optional Tools

- VS Code / IntelliJ / Eclipse (for editing code)
- ERD tools for database diagram representation
- Git for version control

The project is platform-independent and can run anywhere Java and MySQL are installed.

Future Scope

The current version of the Movie & Theatre Ticket Booking System is a fully functional, reliable CLI-based application. However, it provides a strong foundation for several enhancements that can transform it into a production-grade platform.

Potential Future Enhancements:

1. Web-Based Frontend

Develop a full web application using:

- HTML / CSS / JavaScript
- React / Angular / Vue
- Spring Boot backend

This will allow users to browse shows visually and book tickets online.

2. Mobile Application Integration

Create:

- Android App
- iOS App

Both connecting to the same backend using REST APIs.

3. Online Payment Gateway Integration

Replace simulated payments with:

- Razorpay
- Paytm
- Stripe

This would make the system commercially usable.

4. Email or SMS Ticket Delivery

Send:

- PDF tickets
- QR-code based tickets
- Transaction confirmation notifications

5. Dynamic Pricing

Implement seat pricing logic similar to real theatres:

- Peak hours → higher prices
- Weekdays → discounted prices
- Premium rows → higher value

6. Seat Layout UI

Graphical seat map for:

- Better visualization
- Real-time seat blocking

7. Multi-Theatre & Multi-City Support

Extend the system to manage:

- Multiple branches
- City-wise booking
- Interconnected dashboards

8. Cloud Deployment

Host the system on:

- AWS
- Google Cloud
- Azure

Enable scalability and real-time multiuser access.

9. Advanced Analytics

Add:

- Heatmaps of bookings
- Movie performance trends
- Revenue projections
- AI-based show scheduling

Bibliography

Below is the reference list for the tools, technologies, and concepts used in the project.

Books and Study Material

1. **Herbert Schildt**, *Java: The Complete Reference* – Oracle Press
2. **Cay S. Horstmann**, *Core Java Volume I & II* – Pearson Education
3. **Thomas Connolly & Carolyn Begg**, *Database Systems: A Practical Approach to Design, Implementation, and Management*
4. **JDBC™ API Guide**, Oracle Documentation

Online Resources

1. Oracle Java Documentation – <https://docs.oracle.com/javase>
2. MySQL Official Documentation – <https://dev.mysql.com/doc/>
3. JDBC MySQL Connector –
<https://dev.mysql.com/downloads/connector/j>
4. GeeksforGeeks – Java & JDBC tutorials
5. StackOverflow – Code troubleshooting discussions

Tools Used

- OpenJDK / Oracle JDK
- MySQL Community Server
- MySQL Workbench / CLI
- VS Code / IntelliJ IDEA
- Linux Bash Terminal