

**DEPARTMENT OF COMPUTER SCIENCE**

***SEMESTER PROJECT***

***MOVIE TICKET BOOKING SYSTEM***

|  |  |
| --- | --- |
| **SUBMITTED TO** | **DR. RUBINA ADNAN** |
| **SUBMITTED BY** | **RABEEA CHUGHTAI SP22-BCS-079****SADIA IMRAN SP22-BCS-087** |
| **COURSE TITLE** | **CSC270 - DATABASE SYSTEM** |
| **DATED** | **DEC 29 2023** |

Table of Contents

[**Introduction** 4](#_Toc154700961)

[**Description** 4](#_Toc154700962)

[**Entities and Attributes** 5](#_Toc154700963)

[**1.** **MOVIES** 5](#_Toc154700964)

[**2.** **GENRE** 6](#_Toc154700972)

[**3.** **MOVIEGENRE** 6](#_Toc154700975)

[**4.** **SHOW** 7](#_Toc154700978)

[**5.** **SEAT** 7](#_Toc154700984)

[**6.** **BOOKING** 8](#_Toc154700989)

[**7.** **CUSTOMER** 9](#_Toc154700995)

[**Modules** 9](#_Toc154701000)

[**I.** **ADMIN** 9](#_Toc154701001)

[**II.** **CUSTOMER** 10](#_Toc154701008)

[**Functional Requirements** 12](#_Toc154701013)

[ **User Registration and Authentication** 12](#_Toc154701014)

[ **Movie Selection and Filtering** 12](#_Toc154701015)

[ **Reservation Process** 13](#_Toc154701016)

[ **User Profile Management** 13](#_Toc154701017)

[ **Administrator Control:** 13](#_Toc154701018)

[ **Booking Confirmation** 13](#_Toc154701019)

[**Non-Functional Requirements:** 13](#_Toc154701020)

[ **Performance** 13](#_Toc154701021)

[ **Reliability** 13](#_Toc154701022)

[**Tools and Technology:** 13](#_Toc154701023)

[ **Tools:** 14](#_Toc154701024)

[ **Technology:** 14](#_Toc154701025)

# **Introduction**

Online Movie Ticket Booking System, a user-friendly platform lets you effortlessly reserve and purchase movie tickets online. With a visually appealing interface and intuitive filters, you can easily find and book your favorite movies. User accounts ensure a personalized experience, and administrators have the power to manage the movie database.

# **Description**

Online Movie Ticket Booking System is a comprehensive and user-friendly web application designed to streamline the process of reserving and purchasing movie tickets online. Upon visiting the website, users are greeted with a visually appealing home dashboard showcasing an extensive list of movies currently available in our database. With a user-friendly filter sidebar, patrons can tailor their movie selections based on personal preferences, including PG ratings, duration, and genre.

To engage with our system fully, users are required to log in to their personalized accounts. For those without an account, a seamless registration process is available, ensuring accessibility for all. Once logged in, users can easily navigate through the movie listings, select their preferred film, and transition to a dedicated movie page providing details about the movie. The reservation process involves a simple 'Book Me’ click, redirecting users to a Booking page where they can effortlessly reserve a specific number of seats for a chosen showtime. Upon successful completion, the reservation details are promptly displayed on the user's profile. Our administrators wield control over the system, enabling them to add or delete movies. Users can access additional features like password reset and a feedback forum for sharing valuable insights on the website experience. Our project aims to create a movie ticket booking system that combines efficiency, accessibility, and user-centric features for a seamless and enjoyable movie-going experience.

# **Entities and Attributes**

## **MOVIES**

### *MOVIE\_ID (Primary Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute uniquely identifying each movie.

### *DURATION:*

Data Type: NUMBER (1,0)

Represents the duration of the movie

### *TITLE:*

Data Type: VARCHAR2(25 BYTE)

Stores the title of the movie.

### *PG\_RATING:*

Data Type: NUMBER (2,1)

Represents the rating of the movie, particularly the Parental Guidance (PG) rating.

### *INDUSTRY:*

Data Type: VARCHAR2(255 BYTE)

Indicates the industry associated with the movie, such as Hollywood or Bollywood.

### *PRICE:*

Data Type: NUMBER (4,0)

Represents the price or cost associated with the movie.

### *STORY:*

Data Type: VARCHAR2(80 BYTE)

Describes the storyline or plot of the movie.

# **GENRE**

### *GENRE\_ID (Primary Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute represents the unique identifier for each genre in the "genres" table.

### *GENRE\_NAME:*

Data Type: VARCHAR2(50 BYTE)

Stores the name or label of the genre.

## **MOVIEGENRE**

### *MOVIE\_ID (Foreign Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute establishes a link between the "MovieGenre" table and the "movies" table, allowing you to associate genres with specific movies.

### *GENRE\_ID (Foreign Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute establishes a link between the "MovieGenre" table and the "genres" table, allowing you to associate movies with specific genres.

## **SHOW**

### *SHOW\_ID (Primary Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute uniquely identifying each show.

### *START\_TIME:*

Data Type: TIMESTAMP (6)

Represents the start time of the show.

### *END\_TIME:*

Data Type: TIMESTAMP (6)

Represents the end time of the show.

### *MOVIE\_ID (Foreign Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute establishes a link between the "shows" table and the "movies" table, indicating which movie is being shown during the specified time.

### *GENRE\_ID (Foreign Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute establishes a link between the "shows" table and the "genres" table, indicating the genre associated with the movie being shown.

## **SEAT**

### *SEAT\_NUMBER:*

Data Type: VARCHAR2(10 BYTE)

Represents the seat number or identifier for each seat in the venue.

### *STATUS:*

Data Type: VARCHAR2(20 BYTE)

Indicates the status of the seat, such as "available," "reserved," or "booked."

### *BOOKING\_ID (Foreign Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute establishes a link between the "seats" table and the table representing bookings, indicating which booking is associated with a particular seat.

### *SHOW\_ID (Foreign Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute establishes a link between the "seats" table and the "shows" table, indicating the show during which the seat is located.

## **BOOKING**

### *BOOKING\_ID (Primary Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute uniquely identifying each booking.

### *NOOFSEATS:*

Data Type: NUMBER (3,0)

Represents the number of seats booked in the booking.

### *SHOW\_ID (Foreign Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute establishes a link between the "booking" table and the "shows" table, indicating the show for which the seats are booked.

### *CUSTOMER\_ID (Foreign Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute establishes a link between the "booking" table and the table representing customers, indicating the customer who made the booking.

### *PRICE:*

Data Type: NUMBER (4,0)

Represents the total price associated with the booking.

## **CUSTOMER**

### *CUSTOMER\_ID (Primary Key):*

Data Type: VARCHAR2(4 BYTE)

This attribute, uniquely identifying each customer.

### *CUSTOMER\_NAME:*

Data Type: VARCHAR2(25 BYTE)

Stores the name of the customer.

## *CUSTOMER\_EMAIL:*

Data Type: VARCHAR2(40 BYTE)

Represents the email address of the customer.

### *PASSWORD:*

Data Type: VARCHAR2(20 BYTE)

Stores the password associated with the customer's account.

# **Modules**

## **ADMIN**

The admin module facilitates streamlined data management, offering functions for tasks like deleting, updating, inserting, and searching records.

### **Display Movies**

The display function retrieves and displays all records from the "movies" table, offering a comprehensive overview of the available movie data. This function is useful for getting a quick snapshot of the existing movies in the system.

### **Insert**

The insert function allows the admin to add new data to the database. By selecting a table (Movies, Genre, Show, or Movies&Genre), the admin can input information for insertion. This function is crucial for keeping the database up to date with new entries.

### **Delete**

The delete function allows the admin to remove data from the database based on user selection. It presents a panel where the admin can choose the type of data (Movie, Genre, Show, or Booking) to delete. The selected option triggers the appropriate actions for data deletion.

### **Update**

The update function enables the admin to modify existing data in the database. By selecting a table (Show, Movies, or Genre), the admin is prompted to make updates to the chosen data set. This function facilitates the dynamic management of information within the system.

### **Search**

The search function provides the admin with the capability to find specific data in the database. In this case, the admin can search for movies by entering the movie name. The search results help in quickly locating and retrieving relevant information.

## **CUSTOMER**

The customer-oriented functions in the code provide a user-friendly platform for exploring and booking movie tickets.

### **Display Movies**

Triggered by the "Display" button, this function retrieves and displays all records from the "movies" table. It provides customers with a comprehensive overview of available movies, facilitating quick access to essential information.

### **Search Movies**

Activated by the "Search" button, the search function allows customers to find specific movies by entering the movie name. The search results help customers quickly locate and access detailed information about a particular movie.

### **Filter Movies**

The filter function, initiated by the "Filter" button, empowers customers to refine their movie selections based on various criteria. Users can choose from a range of filters such as Duration, Industry, Ratings, Price, and Genre.

* **Duration Filter**

Customers can select the desired movie duration (e.g., 1 hour, 2 hours) using radio buttons, narrowing down the list of movies based on their preferred duration.

* **Industry Filter**

Users can filter movies by industry (e.g., Bollywood, Hollywood, Lollywood) using radio buttons, enabling a more targeted search based on industry preferences.

* **Genre Filter**

The genre filter allows customers to refine their movie choices based on specific genres. This feature enhances the browsing experience, catering to individual taste and preferences.

* **Ratings Filter**

Customers can filter movies by ratings using a slider, enabling a dynamic search based on preferred rating ranges.

* **Price Filter**

The price filter empowers customers to dynamically adjust movie prices based on their preferences. Movies meeting the specified price criteria are instantly displayed.

### **Booking**

* **Choose Movie and Show Timing**

Customers can select a movie from a database-generated list, and the system updates show timings for that movie, allowing them to choose their preferred time.

* **Pick Seats**

The form displays available seats based on the selected show timing. Customers can select multiple seats using checkboxes presented in a grid layout.

* **Confirm Booking and Generate Receipt**

After choosing seats, customers can click the "Book Now" button to confirm their booking. Upon confirming the booking, a receipt is generated, displaying essential. Customers can review the receipt before closing the booking process.

# **Functional Requirements**

## **User Registration and Authentication**

Users should be able to create accounts with unique usernames and passwords. The system should authenticate users during login to ensure secure access.

## **Movie Selection and Filtering**

Users must be able to browse a comprehensive movie catalog with details.

Filtering options, such as genre, duration, and PG rating, should be available for personalized movie selection.

## **Reservation Process**

The system should facilitate the booking of seats for a selected movie and showtime. Users must be able to view and choose available seats in real-time.

## **User Profile Management**

Users should have the ability to view and manage their profile information. The system must display a history of previous reservations for reference.

## **Administrator Control:**

Administrators should have access to functionalities for adding, updating, and deleting movies from the database. The system should provide tools for administrators to manage user accounts.

## **Booking Confirmation**

Users should receive confirmation of their booking, including details like movie title, showtime, and seat numbers.

# **Non-Functional Requirements:**

## **Performance**

The system must efficiently handle concurrent user requests, ensuring responsive interactions even during peak hours, and have a response time of 2 seconds for user interactions.

## **Reliability**

The application should guarantee 99.5% uptime for user availability.

# **Tools and Technology:**

## **Tools:**

***SQL Developer****:* DBMS

***MongoDB Compass*:** Non-Relational Database

***IntelliJ IDEA:*** IDE

## **Technology:**

***JAVA:*** Programming Language

***SQL***: Query Language

***MongoDB:*** NoSQL Query Language

# **Graphical User Interface**

**A movie theater with red seats

Description automatically generated**

**Login & Signup**

**A screen shot of a login form

Description automatically generated**

**A screen shot of a login screen

Description automatically generated**

A screenshot of a login page

Description automatically generated

**Admin Panel**

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Customer Panel**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

# **A diagram of a company Description automatically generatedER Diagram**