MOVIE TICKET RESERVATION SYSTEM

CS23333-OBJECT ORIENTED PROGRAMMING USING JAVA

Submitted by

FURDEEN.F -231001043

ARIVAZHAGAN	-231001017

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE,
THANDALAM
(An Autonomous Institution)

An Autonomous Institution
Affiliated to ANNA UNIVERSITY, Chennai

RAJALAKSHMI ENGINEERING COLLEGE BONAFIDE CERTIFICATE

Certified that this project titled “Movie Ticket Reservation System” is the Bonafide work of “FURDEEN.F(231001043)”,”ARIVAZHAGAN(231001017)” who carried out the project work under my supervision.

SIGNATURE
Dr. Valarmathi
HEAD OF THE DEPARTMENT
Information Technology
Rajalakshmi Engineering College,
Rajalakshmi Nagar, Thandalam
Chennai – 602105

SIGNATURE
Mrs. Usha S
COURSE INCHARGE
Information Technology
Rajalakshmi Engineering College
Rajalakshmi Nagar, Thandalam
Chennai – 602105

This project is submitted for IT19341 – Introduction to Oops and Java held on

INTERNAL EXAMINAR **EXTERNAL EXAMINAR**

TABLE OF CONTENTS

- **Abstract:**

The Movie Ticket Reservation System is a Java-based application that uses JDBC to connect to an MySQL database for storing and analyzing user tickets. The system includes features for inserting user tickets, calculating ticket availability metrics

such as mean and standard deviation, and identifying top and lowest performers. A secure login mechanism ensures that only authorized users can access the system, providing a robust and efficient solution for entertainment experience booking..

- **Introduction:**

The Movie Ticket Reservation System is an innovative application designed to streamline the management and booking of user experience data. Utilizing Java and JDBC to interface with an MySQL database, this system enables the efficient storage, retrieval, and ticket availability evaluation of user tickets. Key features include the ability to input user tickets, compute vital statistics such as average and standard deviation, and identify top and lowest performers. Enhanced with a secure login mechanism, the system ensures that only authorized users can access and manage sensitive entertainment data, making it a robust tool for academic experience booking.

- **Purpose:**

The purpose of this project is to create an efficient and user-friendly user mark booking system that benefits both users and teachers. The system aims to:

- Provide a comprehensive tool for educators to manage user experience data.
- Facilitate detailed ticket availability booking to identify trends and areas for improvement.
- Enable the generation of experience reports and comparative booking.
- Support data-driven decision-making to enhance the entertainment experience.

- **Scope of the Project:**

The scope of the Movie Ticket Reservation System encompasses a range of functionalities designed to enhance the entertainment process by leveraging data management and booking tools. The system aims to streamline the process of recording, storing, and analyzing user tickets. It provides educators with a robust platform to input user tickets, calculate various statistics, and generate comprehensive experience reports.

One of the core components of this project is its integration with a MySQL database using JDBC (Java Database Connectivity). JDBC enables seamless interaction between the application and the MySQL database, allowing for efficient data storage and retrieval. This connectivity ensures that user tickets are securely stored and can be accessed for real-time

booking. The system supports multiple features such as the insertion of user tickets, calculation of ticket availability measures like mean and standard deviation, and the identification of top and lowest performers. Additionally, it offers a user-friendly interface for educators to view and analyze user experience data, facilitating informed decision-making and targeted entertainment interventions.Software Requirement

1.5 Specification:

1. **Database Integration**:
   - **MySQL Database**: The system uses a MySQL database to store user data, including names and tickets. This database is connected using JDBC (Java Database Connectivity), ensuring seamless interaction between the application and the database.
   - **Schema Design**: The database schema includes a table named `users` with columns for user names and their respective tickets.

2. **Backend Development**:
   - **Server Setup**: The backend server is implemented using Java with the `HttpServer` class, handling various endpoints for different functionalities.
   - **Endpoints**:
     - **/api/insert**: Handles the insertion of user tickets.
     - **/api/calculateStatistics**: Computes and returns ticket availability measures such as mean, variance, and standard deviation.
     - **/api/topPerformer**: Retrieves the user with the highest tickets.
     - **/api/lowestPerformer**: Retrieves the user with the lowest tickets.
     - **/api/login**: Manages user authentication for accessing the system.

3. **Frontend Development**:
   - **HTML and CSS**: The frontend is developed using HTML for structure and CSS for styling, providing a user-friendly interface for inputting user tickets, viewing booking results, and generating experience reports.
   - **JavaScript**: Utilizes JavaScript and Chart.js to dynamically update the frontend, handle form submissions, and render ticket availability charts.

4. **Functionality**:
   - **Data Input**: Teachers can input user names and tickets through a web form.
   - **Statistical Analysis**: The system calculates key ticket availability measures and presents them in a comprehensible format.
   - **Performance Reports**: Generates detailed reports highlighting top and lowest performers, along with comparative booking.
   - **User Authentication**: Secure login mechanism to ensure that only authorized personnel can access the system.

5. **Technology Stack**:
   - **Frontend**: HTML, CSS, JavaScript, Chart.js
   - **Backend**: Java, MySQL, JDBC
   - **Tools and Libraries**: JSON handling libraries for data exchange, HttpServer for handling HTTP requests and responses.

This comprehensive system aims to streamline the process of user mark booking, providing educators with valuable insights into user experience through efficient data management and booking tools.

References and Acknowledgement:

- **https://www.javatpoint.com/java-awt**
- **https://www.javatpoint.com/java-swing**

Overall Description

The Movie Ticket Reservation System offers a streamlined solution for managing and analyzing user experience data. It provides authorized users, such as teachers and administrators, with tools to input, view, and analyze user tickets, thereby simplifying the process of academic experience evaluation.

Product Perspective

The system is built using a client/server architecture, compatible with various operating systems. The frontend is developed with HTML, CSS, and JavaScript, incorporating Chart.js for data visualization. The backend is powered by Java with the HttpServer class for handling HTTP requests, and MySQL for efficient data management using JDBC (Java Database Connectivity).

Product Functionality

a) Insert Student Marks: Allows users to input and store user tickets in the database.
b) Calculate Statistics: Computes ticket availability measures such as mean, variance, and standard deviation of user tickets.
c) View Top Performer: Retrieves and displays the user with the highest tickets.
d) View Lowest Performer: Retrieves and displays the user with the lowest tickets. e) Generate Performance Report: Provides detailed reports on user experience. f) User Authentication: Ensures secure access for authorized users.

User and Characteristics

- Qualification: Users should have a basic understanding of entertainment processes and a basic qualification equivalent to matriculation.
- Experience: Familiarity with user experience booking and basic ticket availability concepts is beneficial.
- **Technical Skills**: Users are expected to have elementary knowledge of computers and the ability to interact with web-based applications.

Operating Environment:

Hardware Requirements:

- ***Processor****: Any processor over i3*
- ***Operating System****: Windows 8, 10, 11*
- ***Processor Speed****: 2.0 GHz*
- ***RAM****: 4GB*
- ***Hard Disk****: 500GB*

Software Requirements:

- Database: MySQL (utilizing JDBC)
- ***Frontend****: HTML, CSS, JavaScript (with Chart.js)*
- ***Backend****: Java*

Constraints

- *System access is limited to authorized users such as administrators and teachers.*
- *The delete operation is restricted to administrators without additional checks for simplicity.*
- *Administrators must exercise caution during deletion to maintain data consistency.*

Assumptions and Dependencies

- *System administrators are responsible for creating and securely communicating login IDs and passwords to users.*
- *Users are assumed to have basic knowledge of using web-based applications.*

Specific Requirements

User Interface: The Movie Ticket Reservation System provides user-friendly, menu-driven interfaces for:

a) **Login**: Secure login for authorized users.

b) Insert Student Marks: Input and store new user tickets

. c) View Statistics: Display ticket availability booking of user tickets.

d) Top Performer: Retrieve and display the user with the highest tickets.

e) Lowest Performer: Retrieve and display the user with the lowest tickets.

f) Generate Report: Create and view experience reports.

g) **User Authentication**: Secure management of user access.

Hardware Interface:

- Screen resolution of at least 640 x 480 or above.
- Compatible with any version of Windows 8, 10, 11.

Software Interface for Movie Ticket Reservation System

- **Operating System**: MS-Windows (Windows 8, 10, 11)
- **Frontend Development**: HTML, CSS, JavaScript (with Chart.js)
- **Backend Development**: Java
- Database: MySQL (integrated with JDBC for data management)
- **Integrated Development Environment (IDE)**: NetBeans

Functional Requirements for Movie Ticket Reservation System

- **Login Module (LM)**:
  - Users (admins) can access the Login Module via a secure login page.
  - The system supports login using a username and password.
  - Passwords are masked to ensure security.
  - Only authorized admins, whose credentials match those in the database, are granted access to the system.
- **Registered Users Module (RUM)**:
  - After successful login, users (admins) are granted access to the main features of the application.
  - Users can view detailed information about users and their tickets.
  - Admins have the ability to update user tickets, insert new data, and perform various administrative tasks related to data management.
- **Administrator Module (AM)**:
  - After logging in successfully, the system displays the administrative functions.
  - Admins can manage user data: adding new records, updating existing records, and deleting unused data.
  - The "Add" function allows admins to input new user details and tickets, while the "Update" function allows modifications to existing user information.
  - All add, update, or delete actions trigger communication with the backend (via the Server Module) to make necessary changes in the database.
- **Server Module (SM)**:
  - The Server Module acts as an intermediary between the frontend modules and the database.
  - It receives and processes requests from various modules, ensures proper formatting of data, and manages the system's functionality.
  - It handles communication with the database to validate and execute requests, ensuring data consistency and integrity, especially when dealing with user tickets and booking.

Non-functional Requirements:

Performance:

The system must efficiently handle user data booking requests, ensuring that calculations like mean, standard deviation, and other

statistics are completed in under 2 seconds.
The system should handle a large number of concurrent requests for data input and report generation without significant delays, ensuring high responsiveness for administrators.

Reliability:
The system must be robust and capable of recovering gracefully from failures. In case of data corruption or abnormal shutdown, the system should provide mechanisms for data recovery and ensure minimal loss of data.
The system should be thoroughly tested to handle edge cases such as incorrect or missing data entries, ensuring smooth operation under various conditions.

Availability:
The system should be available 24/7 for administrators to perform tasks like adding, updating, or deleting user data and generating reports.
It should maintain high availability and perform critical operations, like querying user tickets, with minimal downtime or service interruptions.Security:
• A robust security mechanism must be in place on the server side to prevent unauthorized access, safeguard user payment information, and ensure the integrity of the reservation system.
• User privacy, including personal details, must be securely stored and managed to maintain confidentiality.
Security:
- The system must implement strong authentication mechanisms to ensure that only authorized administrators can access the backend and modify sensitive user data.
- Passwords and sensitive user data must be encrypted in both transit (using SSL/TLS) and at rest in the database.
- The system should regularly audit user activities to prevent unauthorized access or modifications to user records.

Maintainability:
The system should be available 24/7 for administrators to perform tasks like adding, updating, or deleting user data and generating reports.
It should maintain high availability and perform critical operations, like querying user tickets, with minimal downtime or service interruptions

- **<u>Module description:</u>**

1. Register:
- The admin can register an account by providing a unique username and a secure password. The system stores these credentials in a secure manner, ensuring only authorized administrators can access the

backend.

2. Login:
- Admins log in using their username and password. The system verifies the credentials against the database, and upon successful login, grants access to the administrative functions.

3.After Login:

- ***Add Movie Tickets:*
  - The admin can input user details including name and tickets. The system saves the data to the database, allowing for future booking and report generation.

- ***View Empty Tickets:*
  - Admins can view detailed information about users' tickets. The data is displayed in a user-friendly format, and the admin can update or modify the records as needed.

- ***Update Tickets:*
  - Admins have the ability to modify a user's tickets. This allows for corrections or updates to be made to any record if necessary.

- ***Delete Resereved Tickets:*
  - Admins can delete user tickets from the system if they are no longer needed. This function is restricted to prevent accidental deletion of important data.

4.Impleme ntation Design:

Home Page: LOGIN

- **Database Design:**

For the Movie Ticket Reservation System, the data is stored and retrieved from a MySQL database, which is chosen for its ability to handle structured data efficiently. The system uses a database to store user tickets, perform ticket availability analyses, and manage administrator activities.

Database Design:

- **Data Elements and Structures: At the booking stage, the required data elements are identified, such as user names, tickets, and ticket availability booking results. These elements are structured and organized to facilitate storage and retrieval.**
- **Normalization:** The database schema undergoes normalization to ensure internal consistency, minimize redundancy, and optimize data storage. This process helps avoid unnecessary duplication of data and ensures that the database is scalable and efficient.
- **Data Integrity:** Relationships between various data items are established, ensuring that data integrity is maintained. The normalization process helps in minimizing the chances of data inconsistencies and allows for easier updates.
- **MySQL Database:** MySQL is selected due to its widespread use, efficiency, and flexibility in handling

relational data. It supports quick, reliable, and flexible access to the data for various users (administrators) while minimizing data inconsistencies.

The database is designed to be efficient, minimizing storage requirements and ensuring high stability.

//MovieTicketsBooking.JAVA

```
import java.util.Scanner;

public class MovieTicketBooking {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Displaying movie options
        System.out.println("Welcome to the Movie Ticket Booking System!");
        System.out.println("Select a movie:");
        System.out.println("1. Movie A - $10");
        System.out.println("2. Movie B - $12");
        System.out.println("3. Movie C - $8");

        // Taking the movie choice from the user
        System.out.print("Enter the number of the movie you want to watch: ");
        int movieChoice = scanner.nextInt();

        // Checking if the movie choice is valid
        if (movieChoice < 1 || movieChoice > 3) {
            System.out.println("Invalid movie selection.");
            return;  // Exit if invalid
        }

        // Taking the number of tickets
        System.out.print("Enter the number of tickets you want to book: ");
        int numTickets = scanner.nextInt();

        // Defining ticket prices based on the movie
        int ticketPrice = 0;
        String movieName = "";

        switch (movieChoice) {
            case 1:
                movieName = "Movie A";
```

```
                ticketPrice = 10;
                break;
            case 2:
                movieName = "Movie B";
                ticketPrice = 12;
                break;
            case 3:
                movieName = "Movie C";
                ticketPrice = 8;
                break;
        }

        // Calculating the total cost
        int totalCost = numTickets * ticketPrice;

        // Displaying the booking details
        System.out.println("\nBooking Summary:");
        System.out.println("Movie: " + movieName);
        System.out.println("Number of tickets: " + numTickets);
        System.out.println("Total cost: $" + totalCost);

        scanner.close();
    }
}
```

Conclusion:

The “Movie Ticket Reservation System” project, developed with attention to detail, the design emphasizes a seamless user experience while ensuring efficient functionality for administrators. Key features include adding, viewing, and managing user tickets, with the ability to perform ticket availability booking and generate reports. The system maintains robust security, especially in sensitive operations like modifying and removing administrator accounts, to ensure data integrity and safeguard against unauthorized access. This system is designed to meet the current needs of entertainment institutions while being adaptable for future improvements, ensuring long-term effectiveness and flexibility in handling user experience data.

Reference links:

- https://www.javatpoint.com/java-awt
  https://www.javatpoint.com/java-swi