MOVIE TICKET BOOKING SYSTEM

A MINIPROJECT FOR UHS UCS 2303 - OBJECT ORIENTED PROGRAMMING

Submitted by

PIRANOW C - 3122235001096

NITHIS KUMAR J- 3122235001096

SABARITHAN P – 3122235001108

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING,

KALAVAKKAM- 603 110

DECEMBER 2024

INDEX

Problem Statement

Design and implement a Movie Ticket Booking System in Java that provides a streamlined and interactive experience for users to browse, select, and book movie tickets at various theaters across multiple cities. The system caters to two primary user roles—Admin and Customer—and is divided into several modules to handle specific functionalities, such as managing movie listings, viewing showtimes, and processing bookings and payments.

MOTIVATION FOR PROBLEM STATEMENT

The motivation for developing this Movie Ticket Booking System stems from the need for a convenient, efficient, and user-friendly solution to streamline movie ticket reservations. Modern users seek the ease of browsing, selecting, and booking tickets online without the inconvenience of visiting theaters in person. At the same time, theater administrators require a simple, scalable system to manage listings, showtimes, and seating across multiple locations with minimal effort and error. By integrating features like secure admin login, UPI payment, and file-based data management, this system not only enhances user experience but also empowers theater staff with robust control over scheduling and availability. Ultimately, the system addresses the increasing demand for an accessible and secure platform that benefits both moviegoers and theater operators alike.

SCOPE

- Users can browse and select movies based on location, date, theater, and showtime.
- The system allows seat selection with real-time availability, marking seats as unavailable once booked.
- UPI payment integration enables secure and cashless booking transactions.
- Admins can add or remove cities, theaters, movies, and update showtimes through a file-based data management system.
- Admin access is secured by a login system to ensure only authorized users can make changes.
- The system uses a simple file structure for data storage, allowing easy maintenance and scalability without requiring a database.
- Modular design allows for future enhancements, such as loyalty programs, user reviews, and additional payment methods.
- The system is intended to streamline ticket booking for users and simplify listing management for administrators across multiple locations.

LIMITATIONS

- Snacks cannot be ordered as part of the booking process.

- Detailed movie ticket information, such as seat numbers, theater location, or booking

- confirmation, cannot be displayed comprehensively within the system.

- The current UPI payment module only supports one payment metho Object Oriented Features d, limiting flexibility for users who may prefer other payment options

- The file-based data storage system may struggle with performance and scalability if the data grows significantly, making it less efficient than a database for handling large volumes.

- Seat availability management depends on text files, which may lead to inconsistencies if files are not updated properly during bookings.

Object Oriented Features

- **Inheritance**

  o Core classes such as Admin and Customer inherit common properties from an abstract User class. This provides a shared structure for all user-related attributes and methods while allowing specialized behavior for each role.

- **Packages**

  o All classes are organized into packages, such as admin, customer, theater, and utilities, separating functionalities logically. For instance, admin stores classes related to admin operations, while theater manages classes for handling theater and showtime information.

- **Interfaces**

  - Java interfaces such as Comparable and Serializable are implemented where necessary for sorting or file handling.

  - User-defined interfaces are also used for modularity. For instance, a BookingInterface could be created to define essential booking methods that both Admin and Customer classes implement.

- **Exception Handling**

  - Exception handling is integrated throughout, especially in file operations for managing location and movie files, and during the UPI payment process to handle invalid inputs or payment failures.

  - Custom exceptions are defined for specific scenarios, such as InvalidBookingException when a user attempts to book unavailable seats, ensuring robust error management
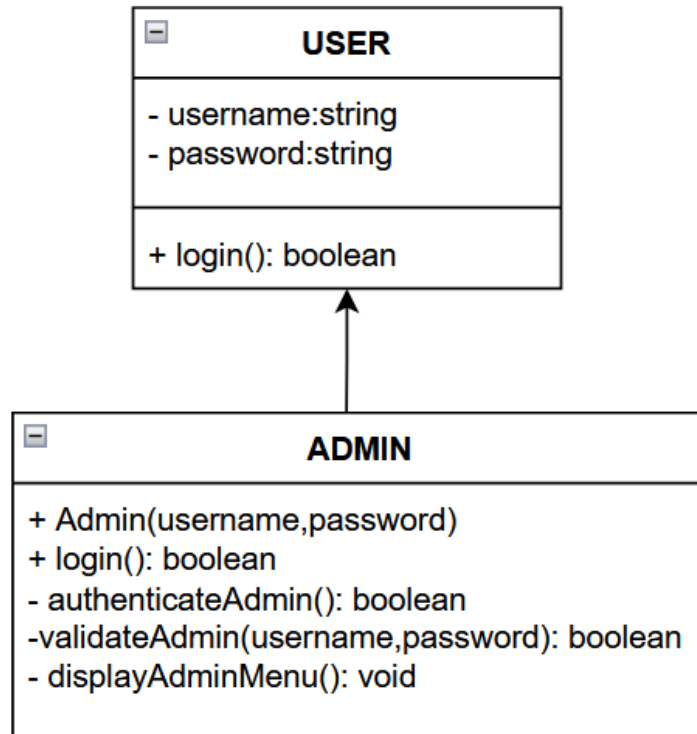
Design of the solution and Explanation of Modules

The Movie Ticket Booking System is structured to provide a smooth, interactive experience for both users and administrators by dividing the program into several core modules. The system follows object-oriented principles, utilizing classes, inheritance, and encapsulation to maintain a modular design. The use of file-based storage simplifies data management, while packages separate functionality, making the system scalable and easy to maintain.

ADMIN CLASS:

The **Admin Class** in the Java-based Movie Ticket Booking System manages the admin login process and provides essential functionalities for maintaining movie listings, theaters, and showtimes. It extends an abstract User class, inheriting basic attributes like username and password, and implements a secure login system that validates credentials from a file (admin.txt). After successful login, the admin accesses a menu that offers options to add or remove movies, manage theaters and showtimes, update or delete theaters, and exit. Each option interacts with the AdminFeatures class, which handles the actual modifications to the data stored in files. This design keeps the system organized and allows efficient management of movie and theater information.
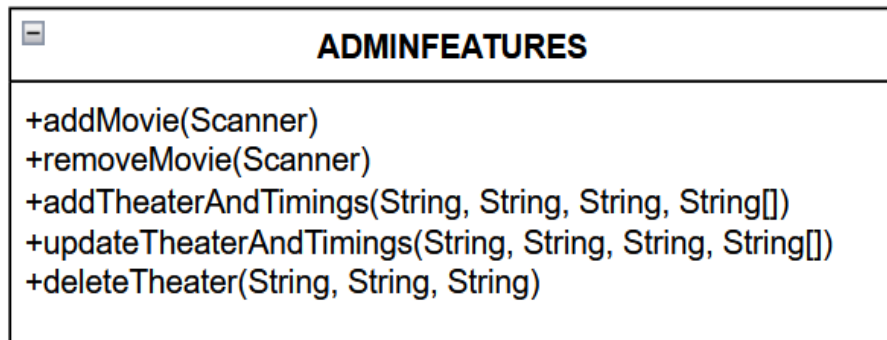
CLASS DIAGRAM:

ADMINFEATURES:

The AdminFeatures Class in the Java-based Movie Ticket Booking System provides various administrative functionalities essential for managing movies, theaters, and showtimes within specific locations. This class allows administrators to add or remove movies by updating location-specific files, each representing available movies in that location. When a movie is added, a dedicated file for the movie-location combination is created to store associated theater names and timings. Admins can further add or update theaters with showtimes within these files, enabling precise management of available showtimes for each theater. Additionally, they can delete specific theaters for a movie, with all operations handling file reading and writing to ensure data consistency. This modular setup simplifies the process of managing movie and theater listings across locations, enhancing flexibility and control over the system's data.

CLASS DIAGRAM:

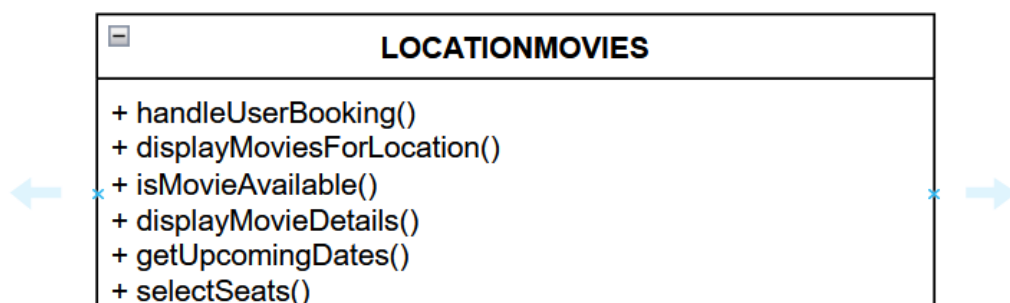| ▭ **ADMINFEATURES** |
| --- |
| +addMovie(Scanner) |
| +removeMovie(Scanner) |
| +addTheaterAndTimings(String, String, String, String[]) |
| +updateTheaterAndTimings(String, String, String, String[]) |
| +deleteTheater(String, String, String) |

LOCATIONMOVIES CLASS:

The LocationMovies class defines a movie ticket booking system that allows users to book tickets by selecting a city, movie, theater, date, and seats. It first checks if the user's location is valid by reading a city-specific file containing available movies. Once a movie is selected, users can choose from upcoming dates and available theaters, followed by selecting seats in a visual seating layout. The system validates inputs and updates seat availability, storing the changes in a file. Finally, it calculates the total price based on seat selection and processes the payment via UPI. The class integrates file I/O, error handling, and seat selection logic to facilitate a smooth booking process.

CLASS DIAGRAM:


```
┌─────────────────────────────────────────────┐
│ ⊟          LOCATIONMOVIES                     │
├─────────────────────────────────────────────┤
│ + handleUserBooking()                         │
│ + displayMoviesForLocation()                  │
│ + isMovieAvailable()                          │
│ + displayMovieDetails()                       │
│ + getUpcomingDates()                          │
│ + selectSeats()                               │
└─────────────────────────────────────────────┘
```
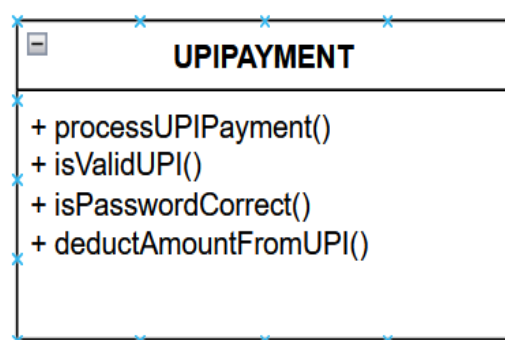
UPIPAYMENT CLASS:

The UPIPayment class is responsible for processing UPI-based payments in a movie ticket booking system. It allows users to enter their UPI ID and password for authentication, ensuring that the provided UPI ID exists in the system and the entered password is correct. Once verified, it checks if the user has sufficient balance by reading from a file (upi.txt), deducting the specified payment amount if the balance is sufficient. The updated balance is then written back to the file. If any issues arise during verification or balance deduction (such as insufficient funds or incorrect credentials), appropriate error messages are displayed. The class integrates file I/O operations for managing UPI details and balances, and employs error handling for file reading and writing processes.

CLASS DIAGRAM:

```
┌──────────────────────────────────┐
│ ▣         UPIPAYMENT              │
├──────────────────────────────────┤
│ + processUPIPayment()            │
│ + isValidUPI()                   │
│ + isPasswordCorrect()            │
│ + deductAmountFromUPI()          │
│                                  │
└──────────────────────────────────┘
```

IMPLEMENTATION AND OUTPUT SCREENSHOTS

Admin Login and Authentication:

Admins will authenticate with a username and password. If the login fails, they will be prompted to retry or contact the head office.

```
Are you a user or admin? (user/admin)
ADMIN
Enter username: SABARI
Enter password: 121
Incorrect credentials. Remaining attempts: 2
1. Retry
2. Forgot password
1
Enter username: sabari
Enter password: 1111
Login successful!
```

Admin Management of Movies and Showtimes:

Admin can add, update, or remove movies and showtimes.

```
Admin Menu:
1. Add Movie
2. Remove Movie
3. Add Theater and Showtimes
4. Update Theater and Showtimes
5. Delete Theater
6. Exit
Choose an option: 1
Adding a movie:
Enter the location: chennai
Enter the movie name to add: GANGUVA

Choose an option: 2
Removing a movie:
Enter the location: chennai
Enter the movie name to remove: GOAT
```

```
Choose an option: 3
Enter the location: chennai
Enter the movie name: GANGUVA
Enter the theater name: IMAX
Enter showtimes (comma-separated): 9.00AM,12.00PM,4.00PM,7.00PM
```

```
Choose an option: 4
Enter the location: chennai
Enter the movie name: GANGUVA
Enter the theater name to update: IMAX
Enter new showtimes (comma-separated): 11.00AM,5.00PM
```

```
Choose an option: 5
Enter the location: chennai
Enter the movie name: GANGUVA
Enter the theater name to delete: IMAX
```

Displaying Movies by Location:

After login, admins can manage movie listings by location. For customers, movies are displayed based on the location selected.

```
Are you a user or admin? (user/admin)
user
Enter your City: chennai
Movies available in chennai:
AMARAN
good bad ugly
lubber pandu
GANGUVA
Do you want to book a ticket or exit? (book/exit)
book
```

Payment Integration:

A simple UPI payment prompt, asking for the user's UPI ID and confirming the payment.

```
Total price: 400.0 INR
Enter your UPI ID for payment: piranow@sbi
Enter your password for UPI ID: 2003
Payment of 400.0 from UPI ID piranow@sbi successful.
```

Booking and Seat Selection:

Once the user selects a movie, they can choose a showtime and select their seats. The seats are marked as unavailable after booking.

```
Enter the name of the movie you want to book: amaran
Available theaters for amaran:
1. ROHINI MULTIPLEX
2. PVR MARINA MALL
3. INOX
4. KS THEATER
5. MAYAJAL
6.
Please select a date for your booking:
1. 10 Nov 2024
2. 11 Nov 2024
3. 12 Nov 2024
Choose a date (1-3): 2
You selected the date: 11 Nov 2024
Choose a theater by entering the serial number: 2
Available showtimes for PVR MARINA MALL:
12.00PM,5.00PM,10.00PM
Enter the desired showtime or type 'exit' to cancel: 5.00PM
No previous seat bookings found. All seats are available.
Enter the number of seats you want to book: 2

                    SCREEN THIS WAY

     1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
A    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
B    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
C    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
D    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
E    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
F    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
G    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
H    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
I    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
J    O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O  O
Enter the seat numbers you want to book (e.g., A1, B2): g12,g13
You have successfully booked seats: G12, G13
Total price: 400.0 INR
```

INFERENCE

The Movie Ticket Booking System provides a functional and effective solution for managing movie bookings and theater operations, particularly for small to medium-scale applications. It allows for easy movie browsing, secure ticket booking, and administrative control over listings and showtimes. However, the system's reliance on file-based data storage and limited payment options may present challenges as the user base or data volume grows. Additionally, the lack of real-time synchronization and advanced security features may affect performance and data integrity in larger, high-traffic environments. Despite these limitations, the system offers a simple yet scalable foundation that could be further enhanced with features like a web or mobile interface, multi-payment support, and a database-driven backend for improved scalability and security.

THANK YOU