

# Movie Ticket Booking System

A CLI-based system in Python for booking movie tickets.

Developed by: Sudip Bhandari



# INTRODUCTION

- The Movie Ticket Booking System is a Python-based command-line application designed to simplify the process of browsing, booking, and managing movie tickets.
- It supports both regular users and administrators, allowing users to view available movies, check showtimes, select seats, and manage their bookings, while admins can add or remove movies and showtimes.
- The system uses a JSON file for data storage, ensuring portability and ease of use without external dependencies.
- With features like seat maps, loyalty points, and fun Easter eggs, this project demonstrates practical object-oriented programming and user-friendly CLI design.

# System Architecture

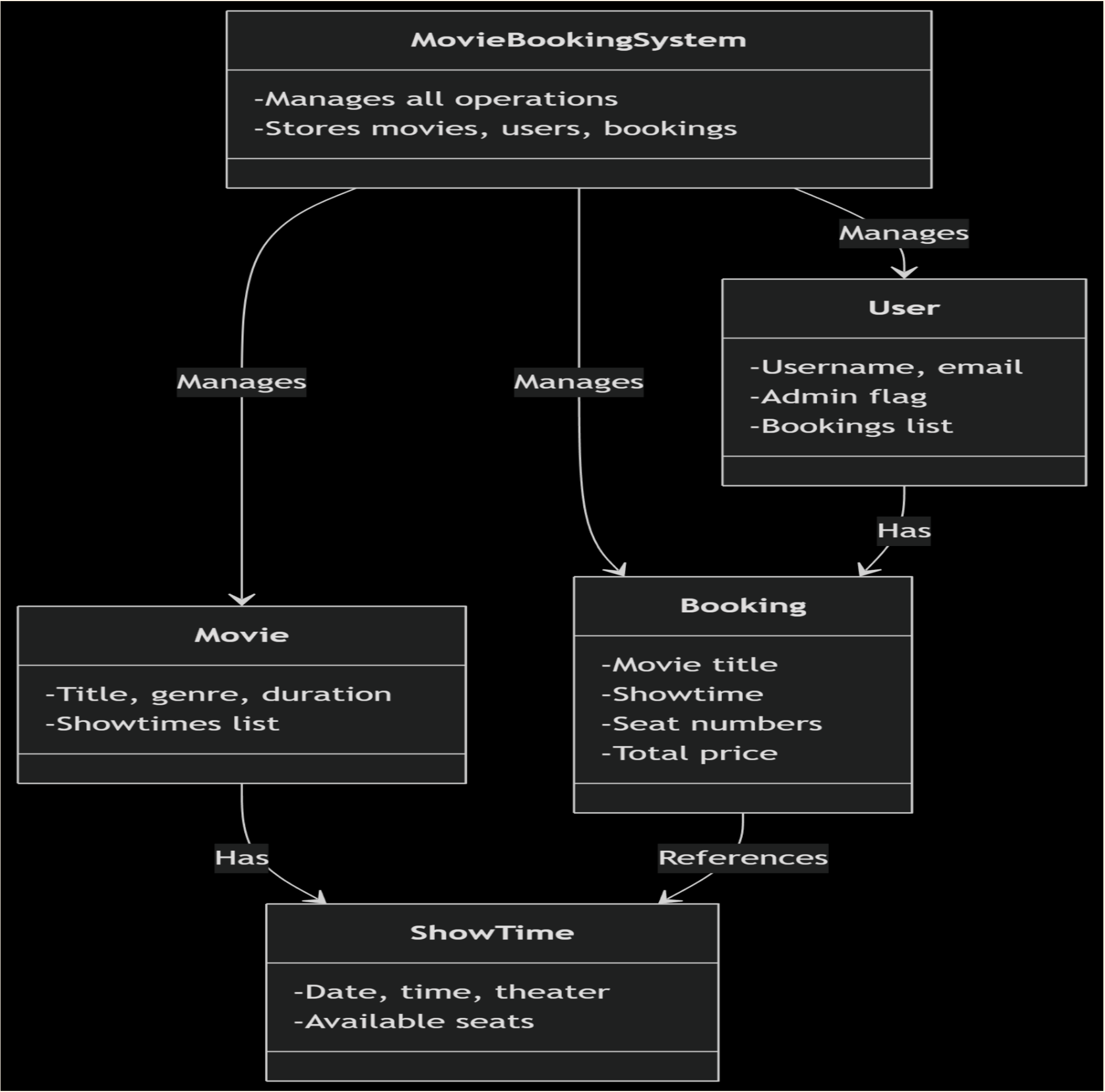
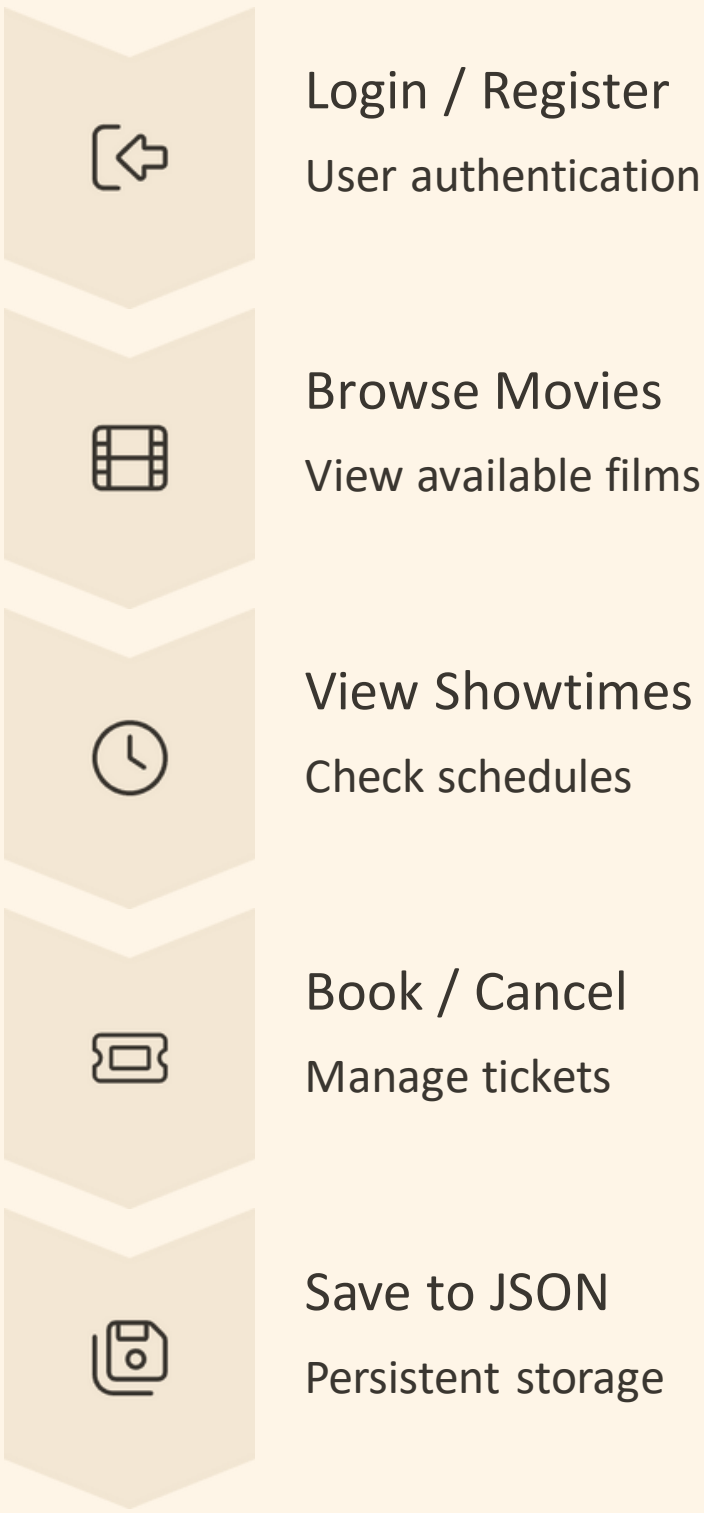


Fig: System Architecture

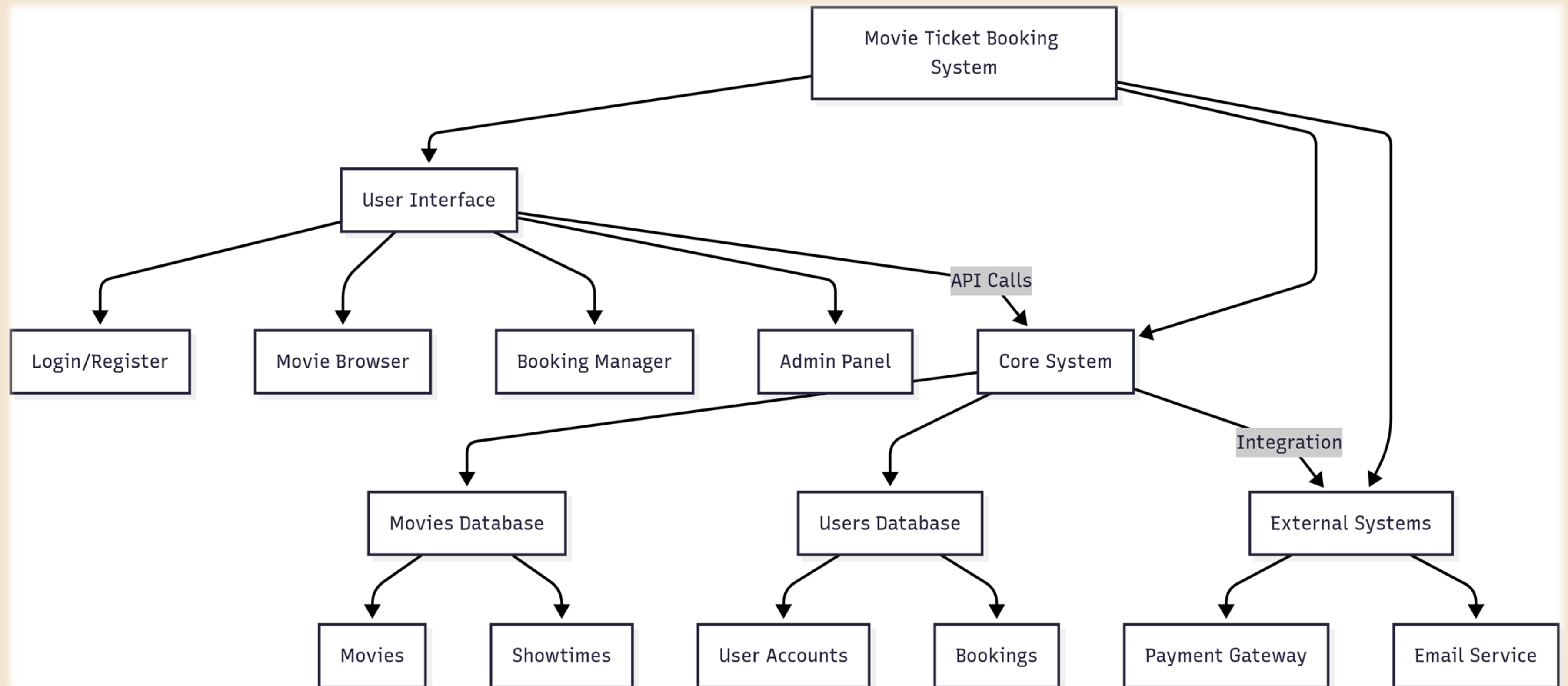


Fig: System Block Diagram

# SYSTEM ARCHITECTURE

## Modules:


- Movie: Movie details (title, genre, price, etc.)
- ShowTime: Showtimes and seat management
- Booking: User bookings and status
- User: User/admin accounts and bookings
- MovieBookingSystem: Main controller (data loading, user actions)

## Data Flow:



- User/Admin interacts via CLI
- System loads/saves data from a JSON file (movie\_system\_data.json)
- All actions (browse, book, cancel, admin ops) update in-memory objects and persist to file

# Key Features


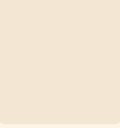
## User Features

- 



Browse Movies

Explore available films 
- 


View Showtimes

Check movie schedules 
- 


Book/Cancel Tickets

Manage reservations 
- 




Booking History & Loyalty

Review past bookings and points 



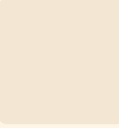
## Admin Features

- 


Add/Remove Movies

Manage film catalog  
- 

Add/Remove Showtimes

Control screening times  
- 

View All Bookings

Access user reservation data 

# User Workflows

## User Login

Simple username-based login.

Admin users require a password.

**Sample Credentials:** Admin: admin / 12345 User: sudip (no password)



## Booking Flow

Select Movie

Choose desired film.

Choose Showtime

Pick a convenient slot.

View Seat Map

Available (🟡) vs. Booked (🔴).

Enter Seats

e.g., A,1;B,2.

Confirm & Get ID

Finalize booking.

# Design Decisions

## Why JSON Files?

- Simplicity & Portability
- No external dependencies
- Easy manual inspection

## Why Class Structure?

- Object-Oriented Design
- Modularity & Reusability

## CLI Focus

- Platform-Independent
- Easy Demo & Testing
- Ideal for Learning



# IMPLEMENTATION DETAILS

## 1. Data Persistence (JSON File)

- All movies, users, and bookings are stored in 'movie\_system\_data.json'.
- Data is loaded at startup and saved after every change (add, book, cancel, etc.).
- Example:

```
```python
with open(self.data_file, 'w') as f:
    json.dump(data, f, indent=2)
```
```

## 2. Booking Logic

- User selects a showtime and seats.
- System checks seat availability and books if possible.
- Each booking is assigned a unique ID and saved.
- Example:

```
```python
if showtime.book_seats(seat_numbers):
    booking_id = str(uuid.uuid4())[:8]
    booking = Booking(...)
    self.bookings[booking_id] = booking
    self.current_user.bookings.append(booking_id)
    self.save_data()
```
```

### 3. Seat Map Display

- Visual seat map with emojis for available/booked seats.
- Example:

```
```python
if seat_num in showtime.booked_seats:
    display += "❌ "
else:
    display += "📺 "
```
```

### 4. Random Surprise (Popcorn Coupon)

- After booking, a random number is generated.
- If the number matches, a coupon message is shown:

```
```python
if random.randint(1, 5) == 3:
    print("🎲 Lucky Draw! You won a free popcorn coupon! 🍿 Use code: POPCORN2025")
```
```

### 5. Loyalty Points

- Users earn 1 point per seat booked.
- Points are displayed in the booking summary:

```
```python
points = len(booking.seat_numbers)
print(f"🏆 Loyalty Points Earned: {points}")
```
```

# Screenshots

## Visual examples of the CLI interface:

```

▼ Enter your choice: 3
🎭 Enter showtime ID: show_movie_003_003

📋 SEAT MAP (🟢 Available, 🔴 Booked):
      1  2  3  4  5  6  7  8  9 10
A : 🟢 🔴 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢
B : 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢
C : 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢
D : 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢
E : 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢
F : 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢
G : 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢
H : 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢
I : 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢
J : 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢

👉 To book, enter seats as e.g. A,1 or B,5 (row letter, column number). Multiple seats: A,1;B,2;C,3

💰 Price per seat: $12.50
📋 Enter seat(s) (e.g. A,1 or B,2;C,3): B,1
💰 Total cost: $12.50
✅ Confirm booking? (y/n): y
🎉 Booking successful! Booking ID: 0aa67d35

```

```
 MOVIE TICKET BOOKING SYSTEM   
=====
```

```
M O V I E   T I C K E T   B O O K I N G  
S Y S T E M
```

```
 Welcome to the most fun way to book your movie tickets! 
```

```
 Welcome to the Movie Ticket Booking System!
```

```
 Demo accounts: 'admin' (admin) or 'sudip' (user)
```

```
 Type 'popcorn' at any menu for a surprise!
```

```
 MAIN MENU:
```

```
1.  Login
```

```
2.  Register
```

```
3.  Exit
```

```
 Enter your choice: 1
```

```
 Enter username: admin
```

```
 Enter password for admin: 12345
```

```
 Login successful!
```

```
 Press Enter to return to menu...
```

```
 Welcome, admin! (ADMIN)
```

```
 MENU OPTIONS:
```

```
1.  Browse Movies
```

```
2.  View Movie Showtimes
```

```
3.  Book Tickets
```

```
4.  My Bookings
```

```
5. ❌ Cancel Booking
```

```
--- ADMIN OPTIONS ---
```

```
6. + Add Movie
```

```
7. - Remove Movie
```

```
8. ⏰ Add Showtime
```

```
9. 🗑️ Remove Showtime
```

```
10. 👤 View All Bookings
```

```
11. 🚪 Logout
```

# Fun Extras



## Loyalty Points

Earn points per seat booked. Users earn loyalty points every time they complete a booking.. This is shown in the "My Bookings" section, where each booking displays the number of seats (and thus points) earned.



## Lucky Draw

Random surprise on booking completion. After a successful booking, there is a mini-game: a random number is generated, and if the result matches a certain value (e.g., 3 out of 1–5), the user receives a surprise message.



## Easter Egg

Type "popcorn" for a surprise!

```
📄 MAIN MENU:
1. 🔑 Login
2. 📄 Register
3. 🖱️ Exit

📌 Enter your choice: popcorn
🎁 You found the secret popcorn! Enjoy your snack while watching the movie! 🎬
```



## Fun Facts

Discover trivia while browsing.

```
ID: movie_001
Genre: Action/Sci-Fi
Duration: 138 minutes
Rating: R
Price: $12.50
Description: Neo and the rebel leaders estimate they have 72 hours until Zion falls under siege.
💡 Fun Fact: 🎁 Popcorn was first sold in movie theaters in 1912!
```

# Data & File Handling

movie\_system\_data.json

Centralized storage for:

- Movies
- Users
- Bookings

Auto-initializes with sample data on first run.

```
{
  "movies": [
    {
      "movie_id": "movie_001",
      "title": "Avengers: Endgame",
      "genre": "Action/Adventure",
      "duration": 181,
      "rating": "PG-13",
      "description": "The Avengers assemble once more to reverse Thanos' actions.",
      "price": 15.0,
      "showtimes": [
        {
          "showtime_id": "show_movie_001_0_0",
          "movie_id": "movie_001",
          "date": "2025-07-07",
          "time": "10:00",
          "theater": "Theater A",
          "total_seats": 50,
          "booked_seats": []
        },
        {
          "showtime_id": "show_movie_001_0_1",
          "movie_id": "movie_001",
          "date": "2025-07-07",
          "time": "13:30",
          "theater": "Theater B",
          "total_seats": 50,
          "booked_seats": []
        }
      ]
    }
  ]
}
```

## I/O Operations

All input/output handled using Python's built-in json module.

Ensures data persistence between sessions.

```
import json
def load_data():
    with open('movie_system_data.json', 'r') as f:
        return json.load(f)
def save_data(data):
    with open('movie_system_data.json', 'w') as f:
        json.dump(data, f, indent=4)
```

# Requirements & Setup



3.7+ or higher Python Version Required



No External Libraries

---

## Run Instructions:

```
python "Movie_Ticket_Booking_system.py"
```

# Thank You!

Your feedback is welcome!