

Hotel Room Reservation System

Objective: Design and implement a user-friendly console-based hotel room reservation system that allows efficient management of bookings, user profiles, and room details using file storage.

Functional Requirements:

1. Hotel Model:

- Define various room types such as Deluxe, Suite, Single, Double in a textual format (e.g., optional read JSON or CSV files).
- Each room type should have attributes like amenities, maximum occupancy, etc., defined in the file.

2. Room Management:

- Create a file for room instances: e.g., Deluxe Room 101, Suite 201.
- Attributes for each room should include:
 - Room number
 - Type (e.g., Deluxe, Suite)
 - Price per night
 - Cancellation fee
 - Status (e.g., available, booked)

3. User Interactions:

- On startup, provide a text menu with options like "1. View Rooms", "2. Book a Room", "3. Cancel Booking", etc.
- For user registration and profile management:
 - Store user profiles in a file with details like username, password, and booking history.
 - Allow users to register, log in, and view their profile.
- Booking:
 - Ask the user for desired dates and room type.
 - Display available rooms and their prices.
 - On booking, update the room's status in the file and store the booking details.
- Cancellation:
 - Request the reservation ID.



- **Compute any cancellation fees, update the room status, and log the cancellation in the user's profile.**

4. Booking Summary and Reports:

- **Provide an admin mode (accessed via a password or special command).**
- **Allow administrators to:**
 - **View all bookings.**
 - **View total income and total cancellation fees.**
 - **Add or remove rooms or modify room details.**

Non-functional Requirements:

1. Data Persistence:

- **Use file operations to read and write data.**
- **Implement error-handling to manage potential file reading/writing issues, ensuring data integrity.**

2. User Input Validation:

- **Ensure input validation for all user inputs to prevent errors and ensure system stability.**

3. Scalability:

- **Design the file structure and code in a modular way to easily add more room types or other features in the future.**

Bonus Features:

1. Special Offers:

- **Allow administrators to set promotional codes that users can apply for discounts.**

2. Search Feature:

- **Provide options to search for bookings based on username, room number, or date range.**

3. Backup System:

- **Implement a feature that allows administrators to create backups of all data files.**



Evaluation Criteria:**1. Code Quality:**

- **Ensure your code is clean, structured, well-commented, and follows best practices. Use meaningful variable and method names.**

2. Functionality:

- **Ensure all functionalities are implemented and are bug-free.**

3. File Management:

- **Ensure efficient, error-free file operations and maintain data consistency.**

4. User Experience:

- **Aim for a user-friendly console interface with clear instructions and feedback.**
- **Keep it simple!!!**

Good luck! Remember to regularly test your application with different scenarios to catch and resolve potential issues.

NB! You can try to make a Booking system with multiple Hotels.

