Advanced Hotel Management System with Enhanced OOP in Python

Project Description: This project involves developing an intricate Hotel Management System in Python, utilizing Object-Oriented Programming (OOP). The system will realistically manage various aspects of a hotel, such as staff, guests, and room bookings, with each entity characterized by unique IDs and descriptive `__str__` methods.

Enhanced Objectives and Class Structure with Specific Methods:

Detailed Class Implementation:

1. **Person (Abstract Class):**
    a. Common Attributes: `unique_id`, `name`, `contact_info`.
    b. Common Methods:
        - `__init__(self, name, contact_info)`: Initialize common attributes.
        - `update_contact_details(self, new_contact_info)`.
        - `__str__(self)`: Return a string representation of the person.
2. **Admin (Inherits from Person):**
    a. Specific Methods:
        - `create_staff_account(self, staff_details)`: Add a new staff member.
        - `remove_staff_member(self, staff_id)`: Remove an existing staff member.
        - `update_staff_role(self, staff_id, new_role)`: Change the role of a staff member.

        - `approve_maintenance_request(self, room_id, maintenance_type)`.
        - `generate_payroll_report(self)`: Create a report for staff salaries.
3. **Staff (Inherits from Person):**
    a. Specific Methods for Different Roles:
        - Receptionist: `book_guest(self, guest_id, room_id)`, `check_out_guest(self, guest_id)`.
        - Housekeeping: `mark_room_cleaned(self, room_id)`, `request_cleaning_supplies(self)`.
        - Maintenance: `report_repair_done(self, room_id)`, `order_repair_materials(self, material_list)`.
4. **Guests (Inherits from Person):**
    a. Specific Methods:
        - `request_room_booking(self, room_type, dates)`.
        - `amend_booking(self, booking_id, new_dates)`.
        - `cancel_booking(self, booking_id)`.
        - `give_feedback(self, feedback_text)`.
5. **Rooms:**
    a. Specific Methods:
        - `set_room_status(self, new_status)`: Change the status of the room.
        - `schedule_room_maintenance(self, maintenance_type)`.
        - `__str__(self)`: Return details of the room, including type and status.
6. **Hotel:**
    a. Specific Methods:
        - `list_available_rooms(self, room_type)`: Show all available rooms of a specified type.

    - `get_guest_details(self, guest_id)`: Retrieve details of a specific guest.
    - `summarize_daily_operations(self)`: Provide a summary of the day's activities and statuses.

Methods and Attributes:

- **introduce new methods and attributes that you deem necessary for the classes. This is your opportunity to tailor the system according to what you perceive as the most efficient and realistic representation of a hotel management system.**

GitHub Repository and README:

- **Upload the project to a GitHub repository.**
- **Include a detailed `README.md` file, explaining the project setup, how to run the code, and a brief overview of its functionalities.**

Detailed Project Report (PDF Format):

- **Write a comprehensive report in PDF format.**
- **The report should detail your class structures, implementation choices, and any additional functionalities you've incorporated.**

Data Storage and Management:

- **Implement JSON files to store essential data like guest information, room details, and staff records.**
- **Develop functionality to read from and write to these JSON files, integrating them with your system.**

Logging:

- **Create a log file to track all system activities.**
- **Each entry in the log should include the action performed and its timestamp.**

Error Handling:

- **Implement robust error handling throughout your system.**
- **Ensure that the system logs errors and continues to operate smoothly without crashing.**

Good luck, dear students!
Zahra Bakhshandeh