Hotel Management System - Project Report

1. Introduction

The **Hotel Management System** is a software application designed to manage hotel room bookings efficiently. It provides a user-friendly interface for guests to book rooms, view bookings, cancel reservations, and check room statuses. The system integrates C++ for the front-end interface, Python for backend data processing, and Excel for storing booking records.

Key Features:

- **Room Booking:** Guests can book rooms with personal details.
- **View Bookings:** Staff can view all booked rooms.
- **Cancel Bookings:** Guests can cancel their reservations.
- **Room Status Check:** Real-time room availability status.
- **Multi-language Support:** Available in English, French, and Spanish.
- **Excel Data Storage:** Uses 'openpyxl' to store and retrieve booking records.

---

2. System Architecture

The system follows a modular architecture with the following components:

Frontend (C++)

- User Interface: Provides menu-driven interaction.
- Language Support: Uses 'LanguageManager' for multi-language text display.
- Input Validation: Ensures correct data entry.

Backend (Python)

- **Excel Data Handling:**
  - 'write_to_excel.py' – Stores booking details.
  - 'listAvailable_rooms.py' – Lists available rooms.
  - 'cancelBooking.py' – Removes bookings.
  - 'view_room.py' – Displays all bookings.
  - 'view_room_status.py' – Checks room status.
- **Data Processing:** Manages room availability and updates records.

Data Storage (Excel)

- Stores guest details (name, mobile, address, check-in/out dates).

- Tracks room availability and types.

---

3. Implementation Details

3.1 Object-Oriented Design

The system utilizes several key OOP concepts:

- **Abstraction** through base class interfaces
- **Encapsulation** of sensitive data
- **Inheritance** for specialized room types
- **Polymorphism** for dynamic method behavior

3.2 Key Components

Language Management System

- Dynamic language switching capability
- Translation file parsing and storage
- Runtime text substitution

User Management

- Abstract base class for user types
- Specialized derived classes for different user roles
- Secure credential handling

Room Management

- Comprehensive room state tracking
- Booking validation system
- Data synchronization mechanisms

3.3 System Features

- **Interactive Console Interface:** Menu-driven navigation
- **Input Validation:** Robust data verification
- **Error Handling:** Graceful failure recovery
- **Multi-language Support:** Runtime language switching

3.4 Modules

1. **LanguageManager** Class

- Loads translations from text files (en.txt, es.txt, fr.txt).

- Provides dynamic language switching.

2. **HotelRoom Class** (Inherits from User)

- Stores guest details (name, mobile, address, check-in/check-out dates).
- Handles room booking and cancellation.

3. **HotelManagement Class**

- Core functionalities:
    - bookRoom()– Books a room and updates Excel.
    - cancelBooking() – Removes a booking.
    - viewAllRooms() – Displays all bookings.
    - viewRoomStatus() – Checks if a room is booked.
    - processPayment() – Simulates payment processing.

4. **PasswordManager Class**

- Securely manages admin passwords.

4. Integration with External Components

Python Integration

- Limited to Excel data operations
- System call-based communication
- Data marshaling between languages

Excel Data Storage

- Python-mediated data persistence
- Structured record keeping
- Basic reporting capabilities

5. Workflow

1. **User selects a language** (English, French, Spanish).
2. **Main Menu Options:**
    - **Book a Room:**
        - Checks availability → Accepts guest details → Updates Excel.
    - **View All Rooms:**
        - Displays all bookings from Excel.
    - **Cancel Booking:**
        - Removes entry from Excel.
    - **View Room Status:**
        - Checks if a room is booked.
    - **Process Payment:**

    - Simulates payment confirmation.
- **Exit:**
    - Closes the program.

---

6. Conclusion

The Hotel Management System efficiently automates room bookings, cancellations, and status checks. By combining C++ for the frontend, Python for backend logic, and Excel for storage, the system provides a seamless experience for both guests and administrators. By employing abstraction, encapsulation, inheritance, and polymorphism, the system offers a modular and maintainable structure. Additionally, the inclusion of multilingual support and user authentication reflects thoughtful design considerations for user accessibility and system security. Overall, this project highlights the practical use of C++ OOP principles in building an interactive and user-friendly hotel booking application.