Dưới đây là yêu cầu chi tiết cho dự án "Hotel Booking System" sử dụng Spring Boot. Dự án này sẽ bao gồm các chức năng quản lý khách sạn cơ bản, cho phép người dùng tìm kiếm và đặt phòng, quản lý đặt chỗ, và quản lý phòng.

### **1. Yêu cầu chức năng**

#### **Người dùng:**

1. **Đăng ký và Đăng nhập**
   * Người dùng có thể đăng ký tài khoản mới.
   * Người dùng đã đăng ký có thể đăng nhập vào hệ thống.
2. **Tìm kiếm và Đặt phòng**
   * Người dùng có thể tìm kiếm phòng khách sạn dựa trên các tiêu chí như thành phố, ngày đến, ngày đi, và số lượng khách.
   * Xem thông tin chi tiết về phòng bao gồm hình ảnh, mô tả, tiện nghi, giá, và tình trạng phòng.
   * Người dùng có thể đặt phòng và nhận được xác nhận đặt phòng.
3. **Quản lý Đặt chỗ**
   * Người dùng có thể xem và quản lý các đặt chỗ của họ.
   * Người dùng có thể hủy đặt chỗ trước thời hạn quy định.
4. **Đánh giá và Bình luận**
   * Người dùng có thể để lại đánh giá và bình luận về khách sạn sau khi đã lưu trú.

#### **Quản trị viên:**

1. **Quản lý Khách sạn**
   * Quản trị viên có thể thêm, chỉnh sửa, và xóa thông tin khách sạn.
   * Quản trị viên có thể quản lý các tiện nghi và dịch vụ của khách sạn.
2. **Quản lý Phòng**
   * Quản trị viên có thể thêm, chỉnh sửa, và xóa thông tin phòng.
   * Quản trị viên có thể quản lý tình trạng phòng (còn trống, đã đặt, đang sửa chữa).
3. **Quản lý Đặt chỗ**
   * Quản trị viên có thể xem tất cả các đặt chỗ và thay đổi trạng thái đặt chỗ.
4. **Quản lý Người dùng**
   * Quản trị viên có thể xem danh sách người dùng, xóa hoặc chặn người dùng.

### **2. Yêu cầu phi chức năng**

1. **Bảo mật**
   * Sử dụng Spring Security để bảo vệ các endpoint.
   * Dữ liệu nhạy cảm (như mật khẩu) phải được mã hóa.
2. **Hiệu suất**
   * Hệ thống cần phải phản hồi nhanh và có khả năng xử lý nhiều yêu cầu đồng thời.
3. **Khả năng mở rộng**
   * Thiết kế hệ thống để có thể dễ dàng mở rộng và thêm tính năng mới.
4. **Tính sẵn sàng**
   * Hệ thống phải có khả năng khôi phục nhanh chóng sau khi xảy ra sự cố.

### **3. Công nghệ và Công cụ**

1. **Back-end:**
   * **Spring Boot:** Framework chính cho ứng dụng.
   * **Spring Data JPA:** Quản lý tương tác với cơ sở dữ liệu.
   * **Spring Security:** Quản lý bảo mật và xác thực người dùng.
   * **MySQL hoặc PostgreSQL:** Cơ sở dữ liệu chính.
   * **Hibernate:** ORM cho Spring Data JPA.
   * **REST API:** Cung cấp các dịch vụ API cho front-end.
2. **Front-end:**
   * **Angular/React/Vue.js:** Framework front-end cho giao diện người dùng.
   * **Bootstrap:** Thư viện CSS cho giao diện responsive.
3. **Khác:**
   * **Swagger:** Tạo và quản lý tài liệu API.
   * **Docker:** Đóng gói ứng dụng và môi trường phát triển.
   * **Jenkins:** CI/CD, tự động hóa quá trình build và deploy.

### **4. Cấu trúc cơ bản của dự án**

* **Model:** Chứa các lớp đại diện cho dữ liệu và tương tác với cơ sở dữ liệu.
* **Repository:** Tương tác với cơ sở dữ liệu.
* **Service:** Chứa logic nghiệp vụ.
* **Controller:** Xử lý các yêu cầu HTTP và trả về phản hồi.
* **DTOs (Data Transfer Objects):** Đối tượng truyền tải dữ liệu giữa các tầng trong ứng dụng.

### **5. Các bước triển khai**

1. **Thiết kế CSDL:** Xác định các bảng và mối quan hệ giữa chúng.
2. **Cài đặt các chức năng Back-end:** Triển khai các API RESTful cho các chức năng người dùng và quản trị viên.
3. **Cài đặt Front-end:** Xây dựng giao diện người dùng và kết nối với API back-end.
4. **Bảo mật:** Thiết lập Spring Security và quản lý phân quyền.
5. **Kiểm thử:** Thực hiện kiểm thử đơn vị, kiểm thử tích hợp và kiểm thử hệ thống.
6. **Triển khai:** Đưa ứng dụng lên server (sử dụng Docker và CI/CD nếu có thể).

### **6. Các yêu cầu bổ sung**

* **Xử lý ngoại lệ:** Đảm bảo hệ thống xử lý tốt các lỗi và cung cấp thông tin hữu ích cho người dùng.
* **Log và giám sát:** Ghi lại các hành động và lỗi xảy ra trong hệ thống để dễ dàng theo dõi và sửa lỗi.