**1. Giới thiệu**

Dự án này là một ứng dụng quản lý khách sạn được phát triển bằng cách sử dụng công nghệ Spring Boot cho backend, Thymeleaf cho giao diện người dùng (frontend), và PostgreSQL làm hệ quản trị cơ sở dữ liệu. Ứng dụng nhằm mục đích cung cấp một hệ thống hiệu quả để quản lý các hoạt động chính của khách sạn, bao gồm quản lý thông tin khách hàng, phòng, khách sạn, đặt phòng, và nhân viên lễ tân.

**2. Mục tiêu Dự án**

* **Quản lý Khách hàng:** Lưu trữ và quản lý thông tin chi tiết của khách hàng như tên, địa chỉ, và số điện thoại.
* **Quản lý Phòng:** Quản lý thông tin về các phòng trong khách sạn, bao gồm số phòng, vị trí, trạng thái, và giá mỗi đêm.
* **Quản lý Khách sạn:** Quản lý thông tin cơ bản về khách sạn, bao gồm tên và vị trí.
* **Quản lý Đặt phòng:** Xử lý các yêu cầu đặt phòng, bao gồm ngày nhận phòng, ngày trả phòng, tổng giá, và trạng thái đặt phòng.
* **Quản lý Lễ tân:** Quản lý thông tin nhân viên lễ tân và các hoạt động mà họ xử lý.
* **Cung cấp giao diện người dùng thân thiện:** Sử dụng Thymeleaf để tạo ra các trang web động, dễ sử dụng cho người quản lý và nhân viên lễ tân.
* **Đảm bảo tính toàn vẹn dữ liệu:** Sử dụng PostgreSQL với các ràng buộc khóa chính và khóa ngoại để duy trì tính nhất quán và toàn vẹn của dữ liệu.

**3. Công nghệ Sử dụng**

* **Spring Boot:**
  + **Mô tả:** Một framework dựa trên Java để phát triển các ứng dụng web độc lập, sẵn sàng cho môi trường sản xuất. Spring Boot đơn giản hóa quá trình cấu hình và triển khai ứng dụng Spring.
  + **Ưu điểm trong dự án:** Tăng tốc độ phát triển với các tính năng tự cấu hình, nhúng Tomcat/Jetty, và quản lý dependency hiệu quả.
* **Thymeleaf:**
  + **Mô tả:** Một công cụ template engine phía máy chủ hiện đại cho các ứng dụng web Java. Thymeleaf được thiết kế để thay thế JSPs và cung cấp một cách an toàn và tự nhiên để tạo các trang web động.
  + **Ưu điểm trong dự án:** Hỗ trợ các template HTML tự nhiên có thể xem trực tiếp trong trình duyệt, dễ dàng tích hợp với Spring Boot, và cung cấp khả năng xử lý biểu mẫu mạnh mẽ.
* **PostgreSQL:**
  + **Mô tả:** Một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở mạnh mẽ, ổn định, và có khả năng mở rộng cao.
  + **Ưu điểm trong dự án:** Hỗ trợ đầy đủ các tính năng SQL, đáng tin cậy, hiệu suất tốt, và phù hợp cho các ứng dụng cần tính toàn vẹn dữ liệu cao.
* **Maven/Gradle (Công cụ quản lý Dependency):** (Nên đề cập nếu sử dụng)
  + **Mô tả:** Công cụ xây dựng dự án và quản lý thư viện phụ thuộc.
  + **Ưu điểm trong dự án:** Giúp quản lý các thư viện cần thiết cho Spring Boot, Thymeleaf, và PostgreSQL một cách hiệu quả.

**4. Kiến trúc Hệ thống**

Dự án tuân theo kiến trúc 3 tầng (3-tier architecture):

* **Tầng Trình bày (Presentation Layer):** Được xây dựng bằng Thymeleaf, chịu trách nhiệm hiển thị giao diện người dùng và tương tác với người dùng. Các file HTML với cú pháp Thymeleaf sẽ được sử dụng để tạo các trang web động.
* **Tầng Logic nghiệp vụ (Business Logic Layer):** Được triển khai bằng Spring Boot, chứa các service và controller xử lý logic nghiệp vụ của ứng dụng. Tầng này sẽ nhận các yêu cầu từ tầng trình bày, xử lý chúng, và tương tác với tầng truy cập dữ liệu.
* **Tầng Truy cập dữ liệu (Data Access Layer):** Sử dụng Spring Data JPA (một phần của Spring Boot) để tương tác với cơ sở dữ liệu PostgreSQL. Tầng này sẽ chứa các Repository để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên cơ sở dữ liệu.

**5. Thiết kế Cơ sở dữ liệu**

Dựa trên sơ đồ thực thể - mối quan hệ (ERD) được cung cấp, cơ sở dữ liệu bao gồm các bảng sau:

* Guest **(Khách hàng):**
  + id (Primary Key - PK)
  + name
  + address
  + phoneNo
* Hotel **(Khách sạn):**
  + id (PK)
  + name
  + location
* Room **(Phòng):**
  + roomNo (PK)
  + location
  + roomStatus
  + pricePerNight
  + hotel\_id (Foreign Key - FK, tham chiếu đến Hotel.id) - Mối quan hệ "Many-to-One" từ Room tới Hotel.
* Receptionist **(Lễ tân):**
  + id (PK)
  + name
  + phoneNo
  + password
* Booking **(Đặt phòng):**
  + id (PK)
  + checkInDate
  + checkOutDate
  + totalPrice
  + status
  + guest\_id (FK, tham chiếu đến Guest.id) - Mối quan hệ "Many-to-One" từ Booking tới Guest.
  + room\_roomNo (FK, tham chiếu đến Room.roomNo) - Mối quan hệ "Many-to-One" từ Booking tới Room.
  + receptionist\_id (FK, tham chiếu đến Receptionist.id) - Mối quan hệ "Many-to-One" từ Booking tới Receptionist.

**Mối quan hệ:**

* Gueststays inRoom **(Many-to-One):** Sơ đồ không thể hiện trực tiếp mối quan hệ này qua khóa ngoại trong Guest. Có vẻ như mối quan hệ "Guest stays in Room" được thể hiện gián tiếp thông qua Booking (Booking liên kết Guest và Room). Tuy nhiên, nếu theo đúng sơ đồ, mối quan hệ "Many-to-One" trên đường nối Guest và Room sẽ được thể hiện thông qua Booking.
* Roombelongs toHotel **(Many-to-One):** Một phòng thuộc về một khách sạn. (Thực hiện thông qua hotel\_id trong bảng Room).
* BookingguestGuest **(Many-to-One):** Một đặt phòng thuộc về một khách hàng. (Thực hiện thông qua guest\_id trong bảng Booking).
* BookingroomRoom **(Many-to-One):** Một đặt phòng dành cho một phòng cụ thể. (Thực hiện thông qua room\_roomNo trong bảng Booking).
* Bookinghandled byReceptionist **(Many-to-One):** Một đặt phòng được xử lý bởi một nhân viên lễ tân. (Thực hiện thông qua receptionist\_id trong bảng Booking).

**6. Các Chức năng Chính (Dựa trên Entities)**

* **Quản lý Khách hàng:**
  + Thêm mới khách hàng
  + Xem danh sách khách hàng
  + Cập nhật thông tin khách hàng
  + Xóa khách hàng
* **Quản lý Khách sạn:**
  + Thêm mới khách sạn
  + Xem danh sách khách sạn
  + Cập nhật thông tin khách sạn
  + Xóa khách sạn
* **Quản lý Phòng:**
  + Thêm mới phòng (liên kết với khách sạn)
  + Xem danh sách phòng
  + Cập nhật thông tin phòng (trạng thái, giá)
  + Xóa phòng
* **Quản lý Đặt phòng:**
  + Tạo đặt phòng mới (liên kết với khách hàng, phòng, lễ tân)
  + Xem chi tiết đặt phòng
  + Cập nhật trạng thái đặt phòng (ví dụ: đã nhận phòng, đã trả phòng, hủy)
  + Xóa đặt phòng
* **Quản lý Lễ tân:**
  + Thêm mới lễ tân
  + Xem danh sách lễ tân
  + Cập nhật thông tin lễ tân
  + Xóa lễ tân
  + (Có thể thêm chức năng đăng nhập/đăng xuất cho lễ tân)

**7. Hướng phát triển trong tương lai**

* **Hệ thống xác thực và phân quyền:** Triển khai Spring Security để quản lý quyền truy cập cho các vai trò khác nhau (ví dụ: quản trị viên, lễ tân).
* **Thanh toán trực tuyến:** Tích hợp với các cổng thanh toán để xử lý thanh toán đặt phòng.
* **Báo cáo và thống kê:** Phát triển các tính năng báo cáo chi tiết về doanh thu, số lượng đặt phòng, tỷ lệ lấp đầy phòng.
* **Thông báo qua email/SMS:** Gửi thông báo xác nhận đặt phòng, nhắc nhở đến khách hàng.
* **Giao diện người dùng cải tiến:** Sử dụng CSS framework (như Bootstrap) để làm cho giao diện đẹp và responsive hơn.
* **Tìm kiếm và lọc nâng cao:** Cung cấp các tùy chọn tìm kiếm và lọc dữ liệu phức tạp hơn trên các danh sách.

**8. Kết luận**

Dự án Quản lý Khách sạn sử dụng Spring Boot, Thymeleaf và PostgreSQL cung cấp một nền tảng vững chắc cho việc quản lý các hoạt động cốt lõi của khách sạn. Với kiến trúc module và các công nghệ hiện đại, dự án này có thể dễ dàng mở rộng và phát triển thêm các tính năng phức tạp trong tương lai.