**Module 4 – Spring**

**Ứng Dụng Quản Lý Khu Nghỉ Dưỡng Furama**

# Mục đích

Học viên sử dụng những kiến thức đã học ở module Spring để viết ứng dụng quản lý Customer, Employees và các dịch vụ của khu nghỉ dưỡng Furama.

# Công nghệ phải dùng

Sau khi kết thúc khóa bài tập, học viên sẽ nắm được các kiến thức sau:

|  |
| --- |
| Spring |
| Spring MVC |
| Data binding |
| Thymeleaf |
| Spring Data JPA |
| Spring Boot |
| Spring Validation |
| Spring Security |

# Mô tả hệ thống

Xây dựng hệ thống quản lý khu nghỉ dưỡng Furama tại thành phố Đà Nẵng. Hệ thống được mô tả như sau:

1. Khu nghỉ dưỡng Furama sẽ cung cấp các dịch vụ cho thuê bao gồm Villa, House, Room.

Tất cả các dịch vụ này sẽ bao có các thông tin: Tên dịch vụ (name), Diện tích sử dụng (area), Chi phí thuê (cost), Số lượng người tối đa (max\_people), Kiểu thuê (rent\_type\_id, bao gồm thuê theo năm, tháng, ngày, giờ).

* Riêng Villa sẽ có thêm thông tin: Tiêu chuẩn phòng (standard\_room), Mô tả tiện nghi khác (description\_other\_convenience), Diện tích hồ bơi (pool\_area), Số tầng (number\_of\_floors).
* Riêng House sẽ có thêm thông tin: Tiêu chuẩn phòng (standard\_room), Mô tả tiện nghi khác (description\_other\_convenience), Số tầng (number\_of\_floors).
* Riêng Room sẽ có thêm thông tin: Dịch vụ miễn phí đi kèm (facility\_free).

Ngoài ra, resort còn cung cấp các dịch vụ đi kèm như massage, karaoke, thức ăn, nước uống, thuê xe di chuyển tham quan resort.

Dịch vụ đi kèm sẽ bao gồm các thông tin: Tên dịch vụ đi kèm (name), Đơn vị (unit), Giá tiền (cost).

1. Hệ thống quản lý còn cho phép quản lý thêm thông tin của Employee resort. Thông tin Employee sẽ bao gồm: Họ tên Employee (name), Ngày sinh (date\_of\_birth), Số CMND (id\_card), Số ĐT (phone), Email (email), Trình độ (education\_degree\_id), Vị trí (position\_id), Lương (salary).

Trình độ sẽ lưu trữ các thông tin (education\_degree): Trung cấp, Cao đẳng, Đại học và sau đại học.

Vị trí sẽ lưu trữ các thông tin (position): Lễ tân, phục vụ, chuyên viên, giám sát, quản lý, giám đốc.

Mỗi Employee sẽ thuộc một bộ phận làm việc (division): Hiện tại resort bao gồm các bộ phận: Sale – Marketing, Hành Chính, Phục vụ, Quản lý.

1. Hệ thống cho phép quản lý thêm thông tin của các Customer đến thuê và sử dụng các dịch vụ của Furama. Thông tin của Customer bao gồm: Họ tên (name), Ngày sinh (birthday), Giới tính (gender), Số CMND (id\_card), Số Điện Thoại (phone), Email (email), Loại khách (customer\_type\_id), Địa chỉ (address) (resort sẽ phân loại Customer để áp dụng các chính sách phục vụ cho phù hợp).

Loại Customer bao gồm (customer\_type): Diamond, Platinium, Gold, Silver, Member.

Customer có thể sử dụng các dịch vụ thuê Villa, House, Phòng và các dịch vụ đi kèm.

Đối với các dịch vụ thuê Villa và House thì Customer cần phải làm hợp đồng thuê với Furama.

Hợp đồng thuê sẽ bao gồm các thông tin: Số hợp đồng (id), Ngày bắt đầu (start\_date), Ngày kết thúc (end\_date), Số tiền cọc trước (deposit), Tổng số tiền thanh toán (total\_money).

Với lược đồ cơ sở dữ liệu như bên dưới:



# Yêu cầu hệ thống

## Yêu cầu 1

**Objectives**

* *Xây dựng được cấu trúc dư án Spring MVC sử dụng Spring Boot.*
* *Thymeleaf Fragment: Header, Footer.*

1. Tạo project để thực hiện các yêu cầu sau:

Khi người dùng gõ địa chỉ trên trình duyệt <http://localhost:8080> thì hiển thị ra Trang chủ theo thiết kế dưới đây:



1. Dựa vào các mô tả ở phần III, tạo prototype cho 4 chức năng sau đây:

* Danh sách dịch vụ.
* Thêm mới dịch vụ.
* Chỉnh sửa dịch vụ.
* Modal xoá dịch vụ.

Lưu ý: Màn hình Thêm mới và chỉnh sửa dịch vụ sẽ thực hiện cho cả 3 loại dịch vụ Villa, House và Room. Vì vậy các thông tin trên màn hình phải phù hợp với từng loại dịch vụ.

1. Dựa vào các prototype bên trên, tiếp tục tạo prototype cho các chức năng:

* Danh sách khách hàng.
* Thêm mới khách hàng.
* Chỉnh sửa khách hàng.
* Modal xoá khách hàng (tạo ở trang danh sách: có xác nhận việc xoá và hiển thị Tên khách hàng).

1. Dựa vào hình ảnh bên dưới để tạo các prototype sau:
   1. Danh sách hợp đồng

Với cột Các dịch vụ đi kèm sẽ bao gồm 2 button:

* Button + sẽ hiển thị modal Thêm mới hợp đồng chi tiết để thêm dịch vụ đi kèm.
* Button Danh sách các Dịch vụ đi kèm sẽ hiển thị modal danh sách các dịch vụ đi kèm.



* 1. Thêm mới hợp đồng
* Item Tổng tiền: sẽ được tự động tính toán sau khi chọn Dịch vụ hoặc thêm Dịch vụ định kèm (Gợi ý: sử dụng Ajax để không phải load lại trang).
* Button + sẽ hiển thị modal Thêm mới hợp đồng chi tiết để thêm dịch vụ đi kèm.



## Yêu cầu 2

**Objectives**

* *Biết sử dụng @Controller, @Service, @Entity.*
* *Áp dụng được Data binding.*
* *Sử dụng Spring Data JPA (Naming Query, @Query).*
* *Query như tìm kiếm, thêm mới, xoá, sửa, delete.*
* *Phân trang với Page<T>, @PageableDefault, Pageable.*

Thiết kế các chức năng với yêu cầu sau:

1. Tạo màn hình thêm mới 1 Khách hàng (sử dụng bảng **customer**).
2. Tạo màn hình liệt kê danh sách Khách hàng (Dạng bảng, có phân trang, nhảy trang, tìm kiếm, liên kết đến các chức năng xóa và chỉnh sửa thông tin của Khách hàng).
3. Tạo màn hình thêm mới 1 Dịch vụ (sử dụng bảng **facility**).
4. Tạo màn hình liệt kê danh sách Dịch vụ (Dạng bảng, có phân trang, nhảy trang, tìm kiếm, liên kết đến các chức năng xóa và chỉnh sửa thông tin của Dịch vụ).
5. Tạo màn hình thêm mới 1 Hợp đồng (sử dụng bảng **contract**).
6. Tạo màn hình thêm mới 1 Hợp đồng chi tiết khi click vào button +. (sử dụng bảng **contract\_detail**).
7. Tạo màn hình liệt kê danh sách các khách hàng đang sử dụng dịch vụ tại resort (Dạng bảng, có phân trang, nhảy trang, tìm kiếm, liên kết đến các chức năng xóa và chỉnh sửa thông tin của Dịch vụ). Nếu hợp đồng có sử dụng dịch vụ đính kèm thì hãy liệt kê các dịch vụ đính kèm đó ra ở modal.  
   (sử dụng bảng **customer**, **contract, contract\_detail, attach\_service**).

## Yêu cầu 3

**Objectives**

* Sử dụng được Validation bằng Annotation và Custom Validation.

1. Tên khách hàng **không** được chứa số. Và các kí tự đầu tiên của mỗi từ phải viết hoa.
2. Tên dịch vụ được phép chứa số. Và các kí tự đầu tiên của mỗi từ phải viết hoa.
3. Số điện thoại phải đúng định dạng 090xxxxxxx hoặc 091xxxxxxx hoặc (84)+90xxxxxxx hoặc (84)+91xxxxxxx.
4. Số CMND phải đúng định dạng XXXXXXXXX hoặc XXXXXXXXXXXX (X là số 0-9).
5. Địa chỉ email phải đúng định dạng.
6. Ngày sinh, Ngày làm hợp đồng, Ngày kết thúc sử dụng datepicker để người dùng có thể chọn ngày thuận tiện. Phải validate xem ngày giờ có hợp lệ hay không (đúng định dạng ngày giờ theo format DD/MM/YYYY).
7. Số lượng, Số tầng phải là số nguyên dương.
8. Lương, Giá, Tiền đặt cọc phải là số dương.
9. Khi click vào nút xóa (hoặc biểu tượng xóa) trên các dòng ở các màn hình liệt kê thì phải hiển thị thông báo confirm việc người dùng có chắc chắn muốn xóa dữ liệu hay không. Nếu người dùng chọn Yes thì xóa dữ liệu ở dòng đó.

## Yêu cầu 4

**Objectives**

* *Sử dụng được Spring Security để login*
* *Sử dụng được Session*
* *Sử dụng được Cookie*

1. Tạo màn hình login cho nhân viên, với giao diện mô phỏng như bên dưới:



Sau khi login thành công thì sẽ chuyển hướng sang home. Sử dụng Session để lưu họ tên nhân viên sau khi đăng nhập và hiển thị tại góc trên bên phải màn hình ở header của tất cả các trang (như giao diện ở Yêu cầu 1).

1. Sử dụng Cookie để lưu tên người dùng (username), password (mật khẩu) khi người dùng chọn checkbox “Remember me”, ngược lại thì sẽ không lưu ở Cookie.