

**Thiết Kế Phần Mềm**

**cho**

**Website Thông tin nhà trọ**

**Phiên bản 1.1 được phê chuẩn**

**Được chuẩn bị bởi:**

**Lương Đức Duy**

**Ngô Minh Phương**

**Nguyễn Hoàng Đông**

**Trường Đại học Cần Thơ**

**Ngày tạo ra bản đặc tả: 06/09/2014**

## Theo dõi phiên bản tài liệu

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Ngày** | **Lý do thay đổi** | **Phiên bản** |
| Lương Đức Duy | 06/09/2014 | Tạo ra tài liệu đặc tả | 1.0 |
| Lương Đức Duy | 06/09/2014 | Tổng quan hệ thống.  Thiết kế dữ liệu.  Thiết kế chức năng:  Tìm kiếm nhà trọ, Yêu cầu đăng thông tin nhà trọ, xem nhà trọ, like hà trọ, bình luận nhà trọ, sửa thông tin nhà trọ. | 1.1 |

Mục lục

[Theo dõi phiên bản tài liệu ii](#_Toc398591405)

[1 Giới thiệu 1](#_Toc398591406)

[1.1 Mục tiêu 1](#_Toc398591407)

[1.2 Phạm vi 1](#_Toc398591408)

[1.3 Bảng chú giải thuật ngữ 1](#_Toc398591409)

[1.4 Tài liệu tham khảo 2](#_Toc398591410)

[1.5 Tổng quan về tài liệu 2](#_Toc398591411)

[2 Tổng quan hệ thống 3](#_Toc398591412)

[3 Kiến trúc hệ thống 4](#_Toc398591413)

[3.1 Thiết kế kiến trúc 4](#_Toc398591414)

[3.2 Mô tả sự phân rã 5](#_Toc398591415)

[3.3 Cơ sở thiết kế 5](#_Toc398591416)

[4 Thiết kế dữ liệu 6](#_Toc398591417)

[4.1 Mô tả dữ liệu 7](#_Toc398591418)

[4.2 Từ điển dữ liệu 10](#_Toc398591419)

[5 Thiết kế theo chức năng 10](#_Toc398591420)

[5.1 Đăng nhập 10](#_Toc398591421)

[5.2 Tìm kiếm nhà trọ 10](#_Toc398591422)

[5.3 Yêu cầu đăng thông tin nhà trọ 12](#_Toc398591423)

[5.4 Xem nhà trọ 15](#_Toc398591424)

[5.5 Like nhà trọ 16](#_Toc398591425)

[5.6 Bình luận trên trang nhà trọ 18](#_Toc398591426)

[5.7 Sửa thông tin nhà trọ 20](#_Toc398591427)

[6 Bảng tham khảo tới các yêu cầu 24](#_Toc398591428)

[7 Các phụ lục 24](#_Toc398591429)

# Giới thiệu

## Mục tiêu

Tài liệu thiết kế phần mềm này được tạo ra nhằm mô tả thiết kế kiến trúc và thiết kế chi tiết về **Website thông tin nhà trọ** cho nhóm phát triển phần mềm và các đối tác. Cụ thể:

* Tài liệu giúp nhóm phát triển phần mềm có thể hiểu rõ các vấn đề về thành phần chức năng và cách thiết kế và cài đặt hệ thống.
* Tài liệu giúp cho khách hàng hình dung về thiết kế các chức năng phần mềm xem có phù hợp với các yêu cầu đã đặt ra hay không, từ đó có thể chỉnh sửa cho đúng ý muốn khách hàng
* Tài liệu được lưu trữ dùng làm tài liệu tham khảo trong quá trình bảo trì phần mềm sau này.
* Tài liệu được sử dụng để làm tài liệu tham khảo trong quá trình kiểm thử phần mềm sau này.

## Phạm vi

* Mục tiêu chính của **“Website thông tin nhà trọ”** là hỗ trợ sinh viên  cũng như người lao động ở xa tìm kiếm nhà trọ trong suốt quá trong quá trình học tập và làm việc ở thành phố Cần Thơ.
* Mục tiêu cụ thể:
  + Phân tích và thiết kế website cung cấp thông tin, tìm kiếm nhà trọ tại thành phố Cần Thơ. Trong đó chú trọng tính tối ưu của cơ sở dữ liệu, tối ưu giao diện nhanh hơn. Thiết kế giao diện làm sao cho người xem tập trung vào chức năng tìm kiếm thông tin, không bị loãng thông tin.
  + Cài đặt website tìm kiếm nhà trọ thành phố Cần Thơ. Sử dụng các công nghệ hoặc dịch vụ mới hiện nay bao gồm: Google Maps, Sping, Hibernate.

## Bảng chú giải thuật ngữ

|  |  |  |
| --- | --- | --- |
| STT | Thuật ngữ / Từ viết tắt | Định nghĩa / Mô tả |
| 1 | MVC | Model-View-Controller |
| 2 | RAM | Random Access Memory |
| 3 | CPU | Central Processing Unit |
| 4 | PowerDesigner | Công cụ thiết kế các mô hình dữ liệu |
| 5 | HDD | Hard Disk Drive |
| 6 | Spring | Spring Framework. |
| 7 | Hibernate | Framework hỗ trợ kết nối cơ sở dữ liệu. |
| 8 | Admin | Người quản trị hệ thống |
| 9 | csdl | Cơ sở dữ liệu |

## Tài liệu tham khảo

[1] Giáo trình Phân tích và thiết kế hệ thống thông tin.

[2] Đoàn Văn Ban (2003), Phân tích thiết kế hướng đối tượng bằng UML – NXB Thống kê.

## Tổng quan về tài liệu

Tài liệu được nhóm thiết kế tạo ra trong quá trình thiết kế hệ thống. Tài liệu mô tả từ tổng quan đến chi tiết về kiến trúc hệ thống, các chức năng, giao diện, đến cơ sở dữ liệu

Nội dung tài liệu bao gồm:

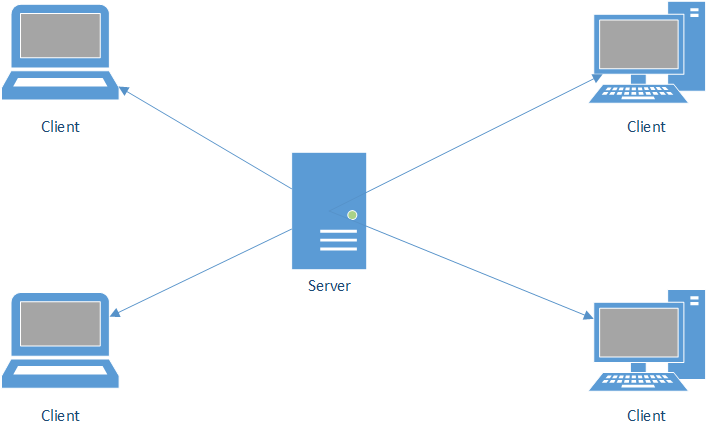
1. Giới thiệu
   1. Mục tiêu.
   2. Phạm vi.
   3. Bảng chú giải thuật ngữ.
   4. Tài liệu tham khảo.
2. Tổng quan hệ thống.
3. Kến trúc hệ thống.
   1. Thiết kế kiến trúc.
   2. Mô tả sự phân rả.
   3. Cơ sở thiết kế.
4. Thiết kế dữ liệu.
   1. Mô tả dữ liệu
   2. Từ điển dữ liệu.
5. Thiết kế theo chức năng.
   1. Đăng nhập.
6. Bảng tham khảo.
7. Các phụ lục.

# Tổng quan hệ thống

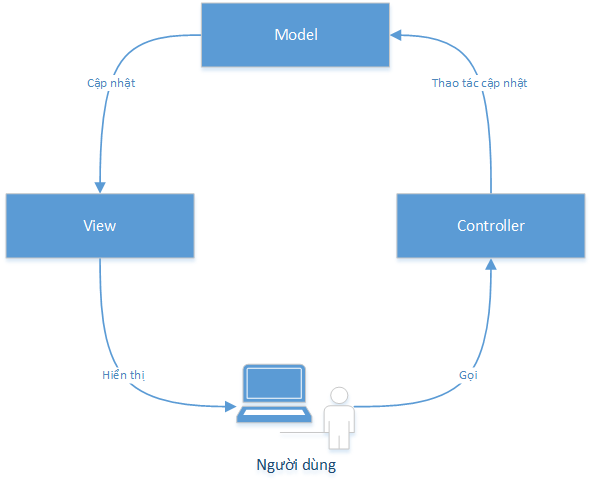
* Ngày này, xã hội phát triển, nhu cầu học tập và lao động ngày càng tăng. Xu hướng chung mọi người đều muốn đến các thành phố lớn để tìm kiếm môi trường, cơ hội làm việc và học tập tiên tiến hơn, hiện đại hơn, dễ dàng cập nhật và tiếp cận các thông tin mới.
* Khó khăn đầu tiên khi mà đi xa nhà để học tập, làm việc đó là nơi ở. Nắm bắt được nhu cầu nhà ở đó, nhiều nhà trọ được xây dựng lên để phục vụ nhu cầu nhà trọ rất lớn, mặt khác đem lại nguồn thu nhập dồi dào cho các chủ nhà trọ.
* Có một hệ quả xảy ra, có quá nhiều nhà trọ, chủ nhà trọ lại muốn quảng cáo về nhà trọ của mình. Những người xa nhà, lần đầu tiên đặt chân đến thành phố lại không thông thuộc đường đi, có quá nhiều thông tin về nhà trọ, như vậy rất khó để cho mọi người tìm kiếm cũng như chủ nhà trọ quảng cáo về nhà trọ của mình.
* Từ vấn đề trên, nhóm để xuất xây dựng “**Website thông tin nhà trọ**” để giải quyết vấn đề tìm kiếm nhà trọ cho người dùng không thông thuộc đường đi tại thành phố Cần Thơ.
* Website Thông tin nhà trọ cung cấp chức năng tìm kiếm thông tin về nhà trọ ở TP Cần Thơ. Người dùng có thể tương tác trực tiếp với trang web thông qua Google Maps được tích hợp ở trang chủ. Ngoài ra, website còn cho phép các thành viên có thể đăng thông tin nhà trọ của mình lên để quảng bá. Thông tin nhà trọ sẽ được nhóm quản trị xác nhận để tránh tình trạng nhà trọ “ma”. Website Thông tin nhà trọ sẽ mang tính xã hội cao, trong đó cho phép người dùng là thành viên của hệ thống có thể vote cho nhà trọ mình thích, hoặc bình luận về nhà trọ có trong dữ liệu của Website.
* Website Thông tin nhà trọ được xây dựng dựa trên nền tảng Java, sử dụng Spring Framework và Hibernate để phát triển. Chạy trên Tomcat.

# Kiến trúc hệ thống

## Thiết kế kiến trúc



*Hình 1: Kiến trúc Client-server.*

**

*Hình 2: Mô hình MVC.*

## Mô tả sự phân rã

## Cơ sở thiết kế

Lựa chọn kiến trúc client-server, do hệ thống chạy trên nền tảng Intenet.

Lựa chọn mô hình MVC để thiết kế do:

* Tính dễ hiểu và phát triển của mô hình cho ứng.
* Dễ dàng mở rộng ứng dụng sau này.
* Mô hình MVC rất phổ biến hiện nay, nên rất dễ tìm kiếm tài liệu và nhận được sự giúp đỡ từ cộng đồng.
* Spring Framework dựa dùng phát triển trên mô hình MVC.

# Thiết kế dữ liệu



*Hình 4.1: Mô hình CDM.*

**Nhà trọ**: lưu thông tin về một nhà trọ, một nhà trọ sẽ thuộc về một loại nhà trọ: Nhà trọ độc lập, nhà riêng, khu nhà trọ, ở cùng chủ nhà. Thông tin về loại nhà trọ được chứa trong bảng **Loại**.

Nhà trọ sẽ có nhiều loại phòng, mỗi loại phòng có thông tin về: diện tích, số người, giá, số lượng, và mô tả thêm các đặc tính khác, thông tin này được lưu trong bảng **loaiphong**.

Nhà trọ chứa tập hợp các hình ảnh dùng để quảng bá cho nhà trọ. Thông tin về hình ảnh lưu trong bảng **Hinh**.

Mỗi người dùng sẽ có một tài khoản trong hệ thống. Thông tin về tài khoản người dùng được lưu trong bảng **User**, trong đó thuộc tính **enabled** để chỉ ra tài khoản có bị khóa hay không. Mỗi người dùng sẽ có các quyền tương ứng trong hệ thống, được lưu trữ trong bảng **Userrole**. Một người có thể có nhiều quyền. Trong hệ thống hiện tại, có 3 quyền chính: Admin, user, khách tự do. Quyền khách tự do không cần lưu vào trong bảng **Userrole**.

Một người dùng là thành viên của hệ thống (có quyền User) có thể đăng một hoặc nhiều nhà trọ. Khi nhà trọ mới được đăng sẽ được chuyển sang trạng thái chờ, thuộc tính **trangthai** trong bảng **nhatro** lưu giữ thông tin trạng thái, cụ thể: 0 – đang chờ xác nhận, 1-Đã được xác nhận, 2-Không được xác nhận.

Người dùng là thành viên của hệ thống có thể like và bình luận trên các trang xem nhà trọ, thông tin này được lưu trong bảng **thich** và **comment**.

Hệ thống cung cấp chức năng remember me cho người dùng khi đã đăng nhập vào hệ thống 1 lần, lần sau có thể tự động đăng nhập mà không cần nhập tài khoản và password. Thông tin để phục vụ cho chức năng này được lưu tron bảng **persistenlogin**.

## Mô tả dữ liệu



*Hình 4.1: Cơ sở dữ liệu.*

**Pesistentlogin**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| series | SERIES | varchar(64) | 64 |  | X |  | X |
| usernametemp | USERNAMETEMP | varchar(200) | 200 |  |  |  | X |
| token | TOKEN | varchar(64) | 64 |  |  |  | X |
| last\_used | LAST\_USED | timestamp |  |  |  |  |  |

**Toado**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| toadoid | TOADOID | int |  |  | X |  | X |
| X | X | float |  |  |  |  | X |
| Y | Y | float |  |  |  |  | X |

**Hinh**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| hinhid | HINHID | int |  |  | X |  | X |
| nhatroid | NHATROID | int |  |  |  | X | X |
| duongdan | DUONGDAN | varchar(200) | 200 |  |  |  | X |

**Nhatro**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| nhatroid | NHATROID | int |  |  | X |  | X |
| loaiid | LOAIID | int |  |  |  | X | X |
| username | USERNAME | varchar(100) | 100 |  |  | X | X |
| toadoid | TOADOID | int |  |  |  | X | X |
| sdt | SDT | varchar(12) | 12 |  |  |  | X |
| diachi | DIACHI | varchar(100) | 100 |  |  |  |  |
| email | EMAIL | varchar(50) | 50 |  |  |  |  |
| trangthai | TRANGTHAI | int |  |  |  |  | X |
| motanhatro | MOTANHATRO | varchar(200) | 200 |  |  |  |  |
| ngaydang | NGAYDANG | datetime |  |  |  |  |  |
| ngayyeucau | NGAYYEUCAU | datetime |  |  |  |  |  |

**Loai**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| loaiid | LOAIID | int |  |  | X |  | X |
| tenloai | TENLOAI | varchar(20) | 20 |  |  |  | X |

**Loaiphong**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| loaiphongid | LOAIPHONGID | int |  |  | X |  | X |
| nhatroid | NHATROID | int |  |  |  | X | X |
| dientich | DIENTICH | int |  |  |  |  | X |
| songuoi | SONGUOI | int |  |  |  |  | X |
| gia | GIA | float(8,2) | 8 | 2 |  |  | X |
| soluong | SOLUONG | int |  |  |  |  | X |
| motaloaiphong | MOTALOAIPHONG | varchar(200) | 200 |  |  |  |  |

**Comment**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| commentId | COMMENTID | int |  |  | X |  | X |
| nhatroid | NHATROID | int |  |  |  | X | X |
| username | USERNAME | varchar(100) | 100 |  |  | X | X |
| comment | COMMENT | varchar(200) | 200 |  |  |  | X |
| datecomment | DATECOMMENT | timestamp |  |  |  |  |  |

**Thich**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| thichid | THICHID | int |  |  | X |  | X |
| username | USERNAME | varchar(100) | 100 |  |  | X | X |
| datethich | DATETHICH | timestamp |  |  |  |  |  |
| nhatroid | NHATROID | int |  |  |  | X | X |

**User**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| username | USERNAME | varchar(100) | 100 |  | X |  | X |
| password | PASSWORD | varchar(10) | 10 |  |  |  | X |
| email | EMAIL | varchar(50) | 50 |  |  |  |  |
| enabled | ENABLED | bool |  |  |  |  |  |

**Userrole**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Data Type | Length | Precision | P | F | M |
| userroleid | USERROLEID | int |  |  | X |  | X |
| username | USERNAME | varchar(100) | 100 |  |  | X | X |
| role | ROLE | varchar(45) | 45 |  |  |  | X |

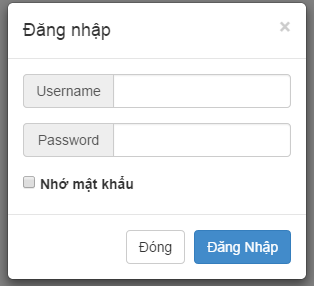
## Từ điển dữ liệu

# Thiết kế theo chức năng

## Đăng nhập

**- Mục đích**: Cho phép người dùng đăng nhập vào hệ thống.

**- Giao diện**:



**- Các thành phần trong giao diện**: ghi vào bảng sau các mô tả về từng thành phần (đã

được đánh số) của giao diện.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Textbox User | Null |  |
| 2 | Textbox Password | Null | Phân biệt hoa thường. |
| 3 | Button Đăng nhập |  |  |
| 4 | Button đóng |  |  |
| 5 | Label Username |  |  |
| 6 | Label Password |  |  |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | User |  |  |  | x |
| 2 | UserRole |  |  |  | x |

**- Cách xử lý:**

1. Người dùng nhập username và password.
2. Người dùng ấn đăng nhập.
3. Thành công hệ thống quay về trang chủ.
4. Nếu thông tin nhập vào không đầy đủ hoặc sai thì quay lại trang đăng nhập.

**- Hàm/ sự kiện:**

**Procedure**: **login**

public ModelAndView login(

@RequestParam(value = "error", required = false) String error,

@RequestParam(value = "logout", required = false) String logout)

Returns ModelAndView

**Pseudocode**

Create new ModelAndView

If error is not equal to null

Call method model.addObject with "error", "Invalid username and password!"

EndIf

If logout is not equal to null

Return result of ModelAndView with "redirect:/"

EndIf

Call method model.setViewName with "login"

Return model

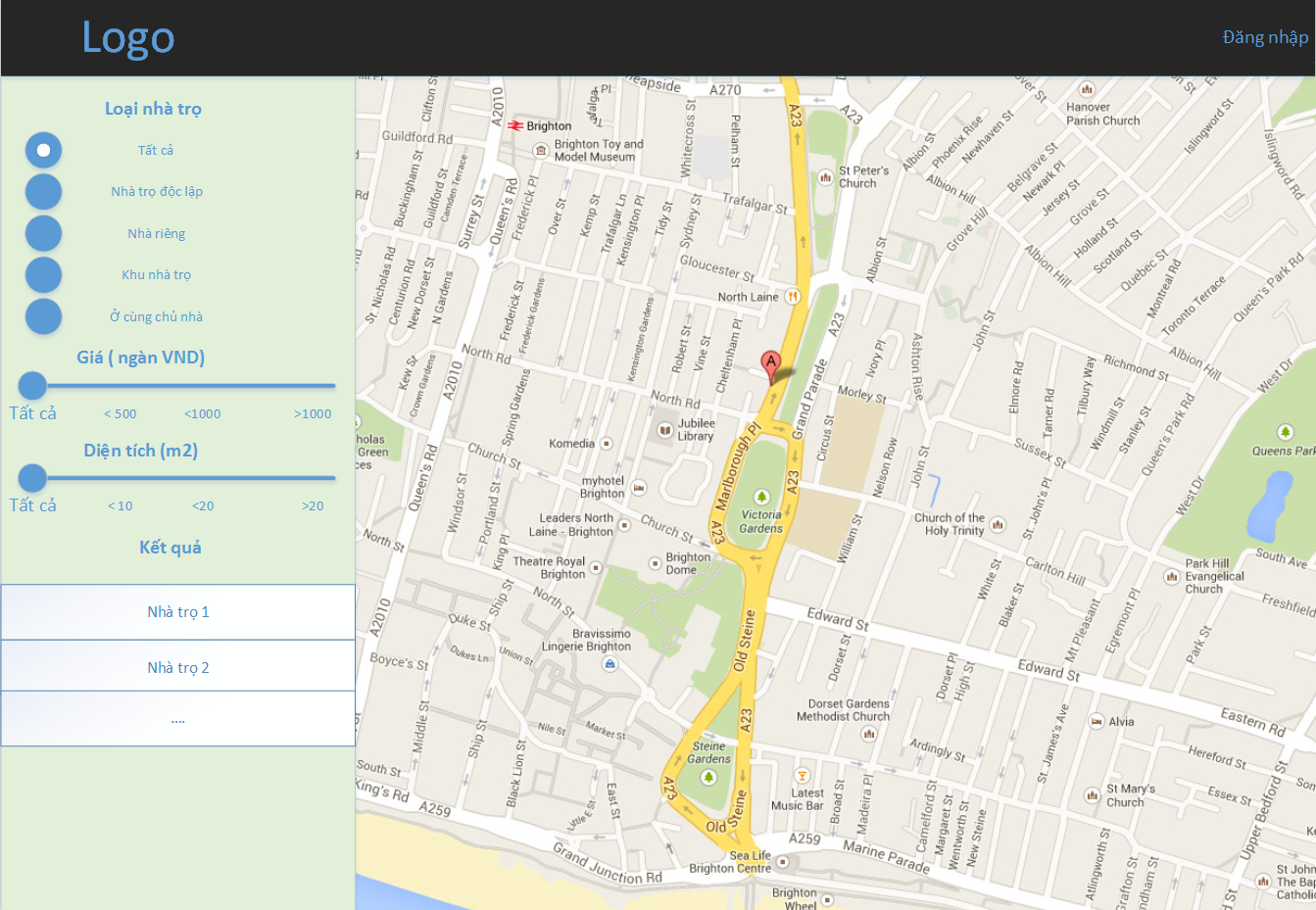
**- Các ràng buộc:**

1. Phần này sử dụng Spring security

## Tìm kiếm nhà trọ

**- Mục đích**: Cho phép người dùng tìm nhà trọ dựa trên các tiêu chí tìm kiếm và thao tác trên Google Maps.

**- Giao diện**:



**- Các thành phần trong giao diện**:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Radio Loại nhà trọ | Tất cả |  |
| 2 | Slider Giá (ngàn VND) | Tất cả |  |
| 3 | Slider Diện tích(m­2) | Tất cả |  |
| 4 | Liên kế Nhà trọ (1, 2, …) |  | Liên kết đến trang xem nhà trọ |
| 5 | Marker |  | Liên kết đến trang xem nhà trọ |

**- Dữ liệu được sử dụng**: liệt kê các bảng trong cơ sở dữ liệu hoặc các cấu trúc dữ liệu

được cần đến bởi chức năng này.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | Nhatro |  |  |  | X |
| 2 | Loai |  |  |  | x |

**- Cách xử lý:**

1. Người dùng truy cập vào website.
2. Người dùng chọn các tùy chọn trên giao diện để tìm kiếm.
3. Hệ thống truy xuất các nhà trọ thỏa điều kiện.
4. Hệ thống hiển thị kết quả ra danh sách các liên kết và các marker trên bản đồ.

**- Hàm/ sự kiện** (nếu có):

Load trang khi vừa truy cập vào website.

**Procedure: home**

public String home(Model model)

**Parameters**

* model - Model

Returns String

**Pseudocode**

Return "home"

Xử lý yêu cầu tìm kiếm (trả về danh sách các đối tượng Marker thông qua Ajax, Jquery xử lý thêm vào trên bản đồ và danh sách liên kết)

**Procedure: searchMotel**

<Marker> searchMotel(HttpServletRequest request)

**Parameters**

request - HttpServletRequest

Returns Marker

**Pseudocode**

Get loaiid

Get mucgia

Get mucdientich

Get toado on Maps

Create new List<Nhatro> nhatros;

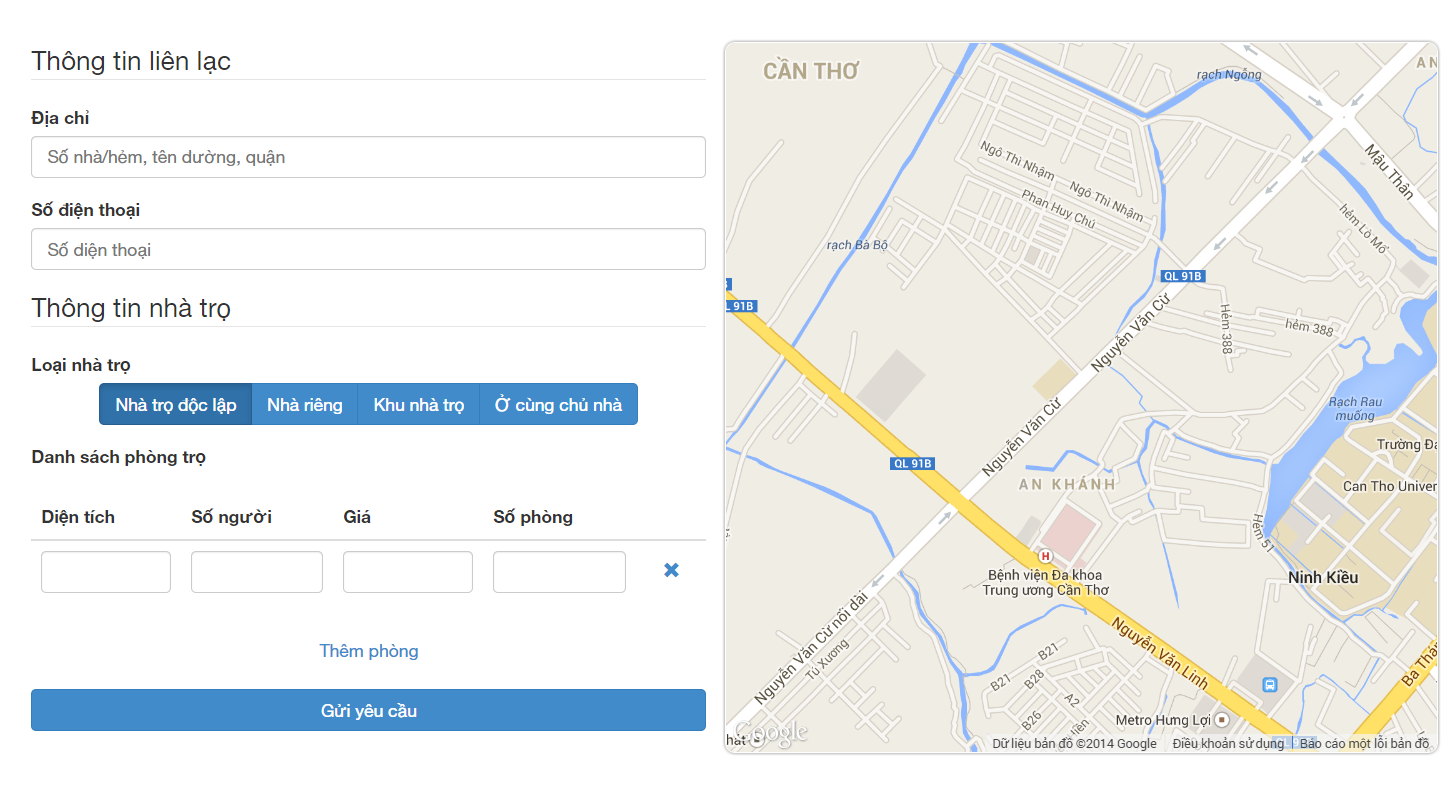
Find nhatro have loaid and gia < mucgia and distance between toado and nhatro.toado <= 10

Return markers

## Yêu cầu đăng thông tin nhà trọ

**- Mục đích**: Cho phép người dùng đã đăng nhập vào hệ thống đăng tin nhà trọ để quảng cáo.

**- Giao diện**:



**- Các thành phần trong giao diện**:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Textbox diachi |  | Đặt placeholder như trong hình để gợi ý người dùng nhập đúng địa chỉ |
| 2 | Textbox Số điện thoại |  | Kiểm tra số diện thoai từ 10 tới 11 số |
| 3 | Radio Loại nhà trọ | Khu nhà trọ |  |
| 4 | Number dientich | 5 | Giá trị nhỏ nhất = 5 |
| 5 | Number songuoi | 1 | Giá trị nhỏ nhất = 1 |
| 6 | Number gia | 200 000 | Giá trị nhỏ nhất = 200 000 |
| 7 | Number sophong | 1 | Giá trị nhỏ nhất = 1 |
| 8 | Liên kết Thêm phòng |  | Dùng Jquery xử lý việc thêm 1 dòng trên danh sách phòng trọ |
| 9 | Submit button Gửi yêu cầu |  |  |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | Nhatro | X |  |  |  |
| 2 | Loaiphong | X |  |  |  |
| 3 | User |  |  |  | X |
| 4 | Loai |  |  |  | X |

**- Cách xử lý:**

1. Người dùng truy cập vào chức năng yêu cầu đăng nhà trọ.
2. Nếu người dùng chưa đăng nhập thì chuyển về đăng nhập để đăng nhập.
3. Hệ thống hiển thị form yêu cầu.
4. Người dùng nhập các thông tin như trong giao diện.
5. Người dùng nhấn nút Gửi yêu cầu.
6. Hệ thống xử lý việc thêm yêu cầu.
7. Quay trở về trang chủ.

**- Hàm/ sự kiện**:

Xử lý hiện form yêu cầu đăng thông tin nhà trọ

**Procedure: yeuCau**

public String yeuCau(Model model)

**Parameters**

* model - Model

Returns String

**Pseudocode**

Return "yeucaudang"

Xử lý yêu cầu đăng thông tin nhà trọ

**Procedure: yeuCauHandling**

public String yeuCauHandling(@ModelAttribute YeuCauDangNhaTro yeuCauDangNhaTro)

Returns String

**Pseudocode**

Create new Nhatro

Call method nhatro.setDiachi with yeuCauDangNhaTro.getDiaChi

Call method nhatro.setSdt with yeuCauDangNhaTro.getSoDt

Call method nhatro.setLoai with loaiDao.findById yeuCauDangNhaTro.getLoaiid

Call method nhatro.setToado with yeuCauDangNhaTro.getToaDo

Call method nhatro.setTrangthai with 0

Call method nhatro.setUser with userDao.findByUserName "current User"

Call method nhatroDao.save with nhatro

Initialise phongs to yeuCauDangNhaTro.getPhongs

For i is 0, i is less than phongs.size, i increments by 1

Call method phongs.get with i

Call method loaiphongDao.save with phongs.get i

EndFor

Return "redirect:/"

## Xem nhà trọ

**- Mục đích**: Cho phép người dùng xem thông tin nhà trọ.

**- Giao diện**:



**- Các thành phần trong giao diện**:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Heading diachi |  | Hiển thị thông tin địa chỉ |
| 2 | Heading gia |  | Hiển thị giá thấp nhất trong các giá phòng thuộc nhà trọ. |
| 3 | Button Like |  | Đổi sang màu đỏ khi người dùng đã like ở trang nhà trọ này |
| 4 | Textarea comment |  | Đặt placeholder “Viết bình luận … Nhấn enter để gửi”  Tự động gửi comment khi người dùng nhấn enter |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | Nhatro |  |  |  | X |
| 2 | Principal |  |  |  | X |
| 3 | User |  |  |  | X |
| 4 | Thich |  |  |  | X |
| 5 | Loaiphong |  |  |  | X |
| 6 | Comment |  |  |  | X |
| 7 | Hinh |  |  |  | X |

**- Cách xử lý:**

1. Người dùng truy cập vào trang nhà trọ cần xem.
2. Hệ thống truy xuất vào cơ sở dữ liệu và load các thông tin cần thiết.

**- Hàm/ sự kiện**:

**Procedure: xemNhaTro**

public String xemNhaTro(@PathVariable("nhatroid") Integer nhaTroId,

Model model, Principal principal)

Returns String

**Pseudocode**

Initialise nhatro to nhatroDao.findById with nhaTroId

Initialise isLike to false

If principal is not equal to null

Initialise user to userDao.findByUserName with principal.getName

Initialise thich to thichDao.findByNhaTroUser with nhatro, user

If thich is not equal to null

Set isLike to true

EndIf

EndIf

Create new ArrayList

Initialise gia to Float.MAX\_VALUE

For (Loaiphong loaiPhong : loaiPhongs)

Set gia to Math.min with gia, loaiPhong.getGia

Call method System.out.println with loaiPhong

EndFor

If gia is equal to Float.MAX\_VALUE

Set gia to 0 as float

EndIf

Create new ArrayList

Create new ArrayList

Call method model.addAttribute with "gia", gia

Call method model.addAttribute with "loaiPhongs", loaiPhongs

Call method model.addAttribute with "nhatro", nhatro

Call method model.addAttribute with "numberOfLikes", nhatro.getThiches

Call method model.addAttribute with "hinhs", hinhs

Call method model.addAttribute with "isLike", isLike

For i is 0, i is less than comments.size, i increments by 1

For j is i plus 1, j is less than comments.size, j increments by 1

If comments.get with i

Call method Collections.swap with comments, i, j

EndIf

EndFor

EndFor

Call method model.addAttribute with "comments", comments

Return "xemNhaTro"

## Like nhà trọ

**- Mục đích**: Cho phép người like nhà trọ.

**- Giao diện**:



**- Các thành phần trong giao diện**:

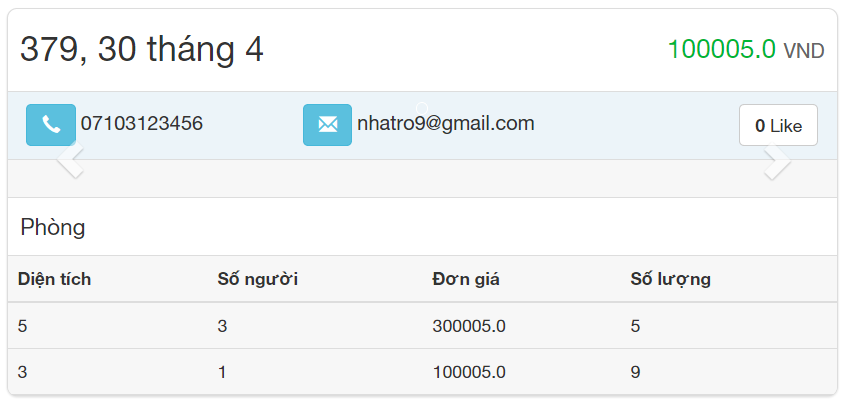
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Button like |  | Màu trắng nếu người dùng chưa like nhà trọ này, ngược lại hiển thị màu đỏ. |

**- Dữ liệu được sử dụng**: liệt kê các bảng trong cơ sở dữ liệu hoặc các cấu trúc dữ liệu

được cần đến bởi chức năng này.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | Nhatro |  |  |  | X |
| 2 | User |  |  |  | X |
| 3 | Thich | x |  | x | x |

**- Cách xử lý:**





Nếu người dùng chưa đăng nhập thì chuyển về trang đăng nhập.

**- Hàm/ sự kiện**:

**Procedure: likeNhaTro**

Public String likeNhaTro(@PathVariable("nhatroid")Integer nhaTroId, Principal principal)

Returns String

**Pseudocode**

Initialise nhatro to nhatroDao.findById with nhaTroId

Initialise name to principal.getName

Initialise user to userDao.findByUserName with name

Initialise thich to thichDao.findByNhaTroUser with nhatro, user

If thich is equal to null

Create new Thich

Call method thich.setNhatro with nhatro

Call method thich.setUser with user

Call method thichDao.add with thich

Else

Call method thichDao.delete with thich

EndIf

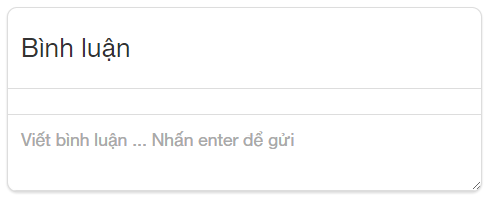
Return "redirect:/nhatro/" plus nhaTroId

**- Các ràng buộc :** người dùng phải đăng nhập vào hệ thống mới được like.

## Bình luận trên trang nhà trọ

**- Mục đích**: Cho phép người bình luận trên các trang nhà trọ.

**- Giao diện**:



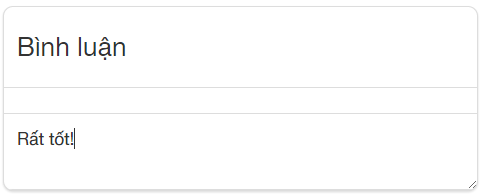
**- Các thành phần trong giao diện**:

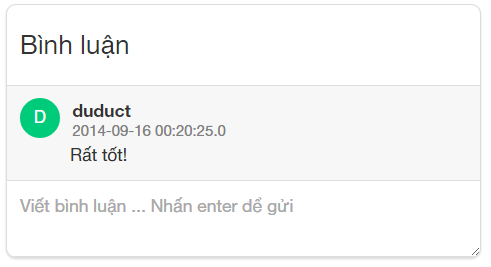
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Textarea comment |  | Đặt placeholder như trong hình, tự động đăng comment khi người dùng nhấn nút enter |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | Nhatro |  |  |  | X |
| 2 | User |  |  |  | X |
| 3 | Comment | x |  |  |  |

**- Cách xử lý:**





**- Hàm/ sự kiện**:

**Procedure: commentNhaTro**

Public String commentNhaTro(@PathVariable("nhatroid") Integer nhaTroId, Principal principal, HttpServletRequest request)

Returns String

**Pseudocode**

Initialise nhatro to nhatroDao.findById with nhaTroId

Initialise user to userDao.findByUserName with principal.getName

Create new Comment

Call method comment.setComment with request.getParameter "comment"

Call method comment.setNhatro with nhatro

Call method comment.setUser with user

Call method commentDao.add with comment

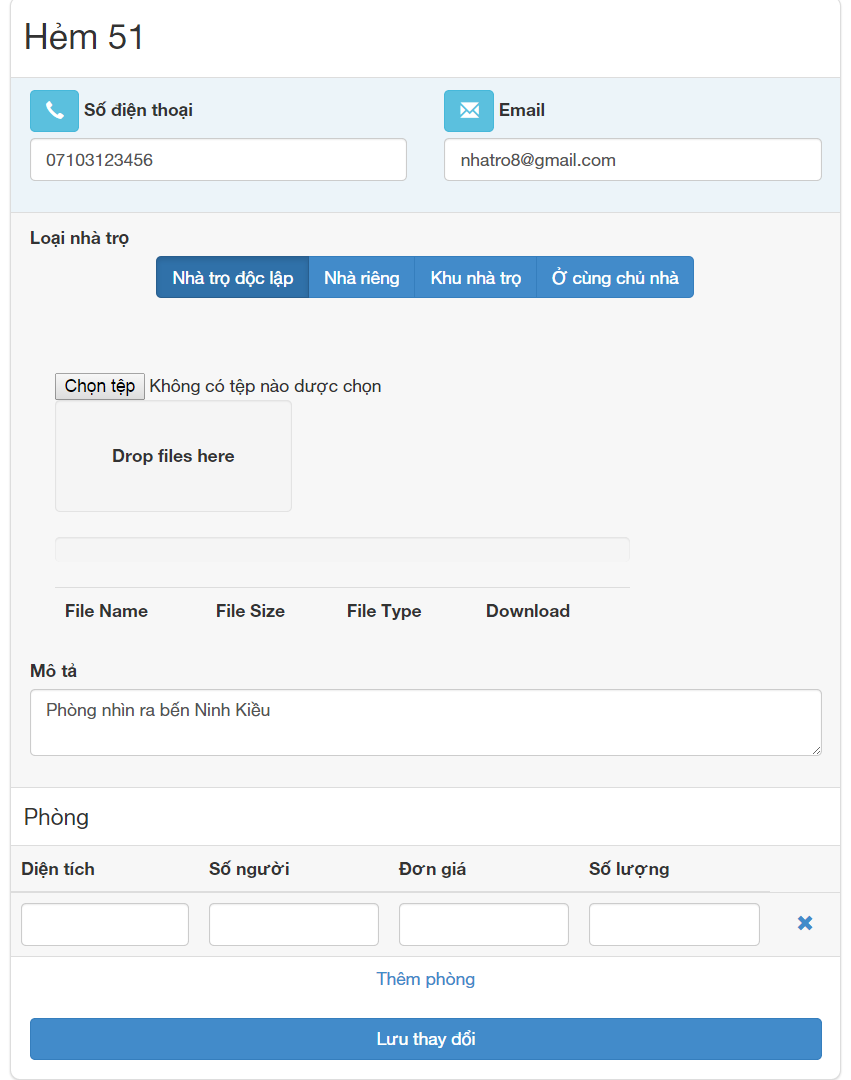
Return "redirect:/nhatro/" plus nhaTroId

**- Các ràng buộc**: người dùng phải đăng nhập vào hệ thống mới được comment.

## Sửa thông tin nhà trọ

**- Mục đích**: Cho phép người dùng là chủ của nhà trọ sửa thông tin nhà trọ.

**- Giao diện**:



**- Các thành phần trong giao diện**: ghi vào bảng sau các mô tả về từng thành phần (đã

được đánh số) của giao diện.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
|  | <Mỗi thành phần trong giao diện có thể là button hay textbox hay combo box…> |  | Viết lưu ý cho những thành  phần trong giao diện có cách  xử lý đặc biệt hoặc các quy  định mà lập trình viên phải  thực hiện. |
| 1 | Textbox sdt |  | Kiểm tra tính hợp lệ của số điện thoại |
| 2 | Textbox email |  | Kiểm tra tính hợp lệ của email. |
| 3 | Radio loainhatro |  |  |
| 4 | fileUpload file[] |  | Cho phép kéo thả hình ảnh để upload |
| 5 | Textarea motanhatro |  |  |
| 6 | Number dientich | 5 | Giá trị nhỏ nhất = 5 |
| 7 | Number songuoi | 1 | Giá trị nhỏ nhất = 1 |
| 8 | Number gia | 200 000 | Giá trị nhỏ nhất = 200 000 |
| 9 | Number sophong | 1 | Giá trị nhỏ nhất = 1 |
| 10 | Liên kết Thêm phòng |  | Dùng Jquery xử lý việc thêm 1 dòng trên danh sách phòng trọ |
| 11 | Submit button Lưu thay đổi |  |  |

**- Dữ liệu được sử dụng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | Nhatro |  | x |  | X |
| 2 | user |  | x |  |  |
| 3 | Loai |  |  |  | X |
| 4 | Loaiphong | X |  | X |  |
| 5 | Hinh | X |  | X |  |

**- Cách xử lý:**

1. Người dùng truy cập vào tính năng sửa nhà trọ.
2. Nếu người dùng chưa đăng nhập thì chuyển về trang đăng nhập.
3. Hệ thống hiển thị form nhập các thông tin như trong hình.
4. Người dùng chỉnh sửa các thông tin trong form.
5. Người dùng nhấn nút Lưu thay đổi.
6. Hệ thống nhận thông tin từ form và cập nhật vào csdl.

**- Hàm/ sự kiện** :

Xử lý yêu cầu hiển thị form sửa thông tin nhà trọ tương ứng với nhatroid

**Procedure: suaNhaTro**

public String suaNhaTro(@PathVariable("nhatroid") Integer nhaTroId,

Model model)

Returns String

**Pseudocode**

Initialise nhatro to nhatroDao.findById with nhaTroId

Call method model.addAttribute with "nhatro", nhatro

Create new ArrayList

Call method model.addAttribute with "loaiPhongs", loaiPhongs

Return "suaNhaTro"

Xử lý thông tin đã sửa từ form.

**Procedure: suaNhaTroHandling**

public String suaNhaTroHandling(

@ModelAttribute YeuCauDangNhaTro yeuCauDangNhaTro,

HttpSession session)

Returns String

**Pseudocode**

Create new Nhatro

Set nhatro to nhatroDao.findById with yeuCauDangNhaTro.getNhatroid

Call method nhatro.setSdt with yeuCauDangNhaTro.getSoDt

Call method nhatro.setEmail with yeuCauDangNhaTro.getEmail

Call method nhatro.setLoai with loaiDao.findById yeuCauDangNhaTro.getLoaiid

Call method nhatro.setUser with userDao.findByUserName "admin"

// Remove all old loaiphong

For (Loaiphong phong : phongs)

Call method loaiphongDao.delete with phong

EndFor

// Add new loai phong

For i is 0, i is less than phongs.size, i increments by 1

Call method phongs.get with i

Call method loaiphongDao.save with phongs.get i

EndFor

Update image list

For (Hinh hinh : hinhs)

Call method hinhDao.delete with hinh

EndFor

Add new hinh

Initialise count to 0

For (FileMeta newHinh : newHinhs)

Create new Hinh

Call method hinh.setNhatro with nhatro

Initialise absoluteFinalDirectory to fileDirectory.getSaveDirectoryImage plus "hinh"

Initialise dynamicFinalDirectory to "hinh"

Call method hinh.setDuongdan with dynamicFinalDirectory

Try

Call method FileCopyUtils.copy with newHinh.getBytes

Catch FileNotFoundException e

Call method System.out.println with "File" plus absoluteFinalDirectory plus " not found !"

Call method e.printStackTrace

Catch IOException e

TODO Auto-generated catch block

EndTry

Call method hinhDao.add with hinh

Call method fileMetaDao.delete with newHinh

Count++;

EndFor

Call method session.removeAttribute with "imagesList"

Return "redirect:/"

**- Các ràng buộc**: Người dùng phải là chủ nhà trọ và đã đăng nhập vào hệ thống mới được sửa thông tin nhà trọ.

## Quản lý yêu cầu đăng tin

**- Mục đích**: Cho phép người dùng quản lý các yêu cầu đăng tin từ thành viên.

**- Giao diện**:

1

2

3

4

5

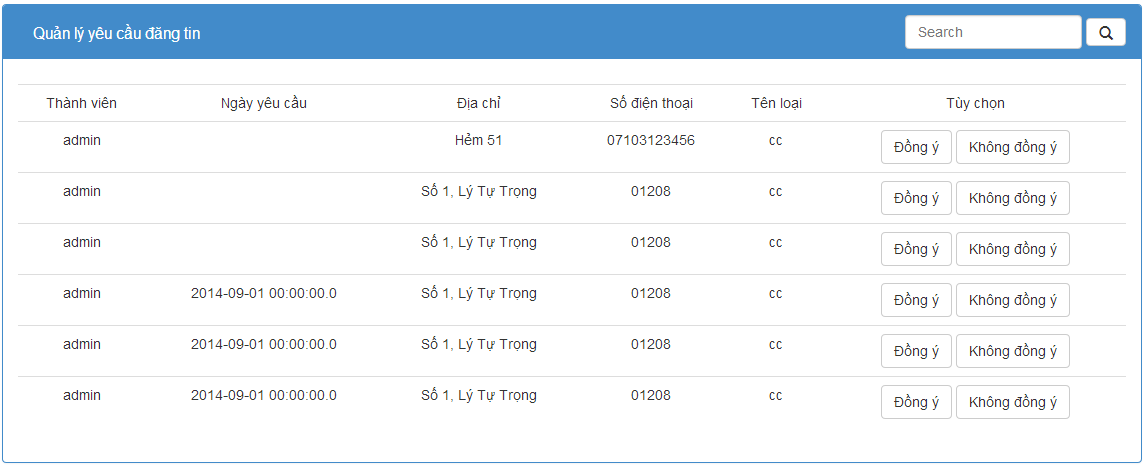
6

7

8

9

10



**- Các thành phần trong giao diện**: ghi vào bảng sau các mô tả về từng thành phần (đã

được đánh số) của giao diện.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Label Quản lý yêu cầu đăng tin |  |  |
| 2 | Textbox Search | Null | Tìm kiếm theo giá trị tương đương **VD**: Nhập “ad” kết quả có thể là “admin”. |
| 3 | Button Search |  | Khi được nhấn lấy thông tin từ Textbox Search rồi thực hiện tìm kiếm. |
| 4 | Label Thành viên |  | Tên thành viên gửi yêu cầu đăng tin. |
| 5 | Label Ngày yêu cầu |  | Hiển thị ngày yêu cầu, có thể hiển thị đầy đủ ngày giờ hoặc chỉ hiển thị ngày. |
| 6 | Label Địa chỉ |  | Địa chỉ của thông tin nhà trọ mà thành viên gửi yêu cầu. |
| 7 | Label Số điện thoại |  | Số điện thoại của người gửi yêu cầu. |
| 8 | Label Tên loại |  | Tên loại nhà trọ. |
| 9 | Button Đồng ý |  | Đồng ý cho phép thông tin nhà trọ được đăng (hiển thị) trên website. |
| 10 | Button Không đồng ý |  | Không đồng ý cho phép thông tin nhà trọ được đăng (hiển thị) trên website. |

**- Dữ liệu được sử dụng**: liệt kê các bảng trong cơ sở dữ liệu hoặc các cấu trúc dữ liệu

được cần đến bởi chức năng này.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | nhatro | dongy |  | khongdongy | listWaiting |

**- Cách xử lý:**

. **Trường hợp 1:**

1. Hiển thị danh sách các yêu cầu đăng tin.
2. Người dùng chọn button Đồng ý để chấp nhận cho yêu cầu đăng tin.
3. Người dùng chọn button không đồng ý để không chấp nhận cho yêu cầu đăng tin.

**Trường hợp 2:**

1. Hiển thị danh sách các yêu cầu đăng tin.
2. Người dùng nhập nội dùng cần tìm kiếm vào Textbox Search.
3. Hệ thống hiển thị kết quả (Nếu không có thì hiện thông báo “Nội dung tìm kiếm không tồn tại”).
4. Người dùng chọn button Đồng ý để chấp nhận cho yêu cầu đăng tin.
5. Người dùng chọn button không đồng ý để không chấp nhận cho yêu cầu đăng tin.

**- Hàm/ sự kiện** (nếu có): mô tả giải thuật cho từng biến cố bằng sơ đồ hoặc bằng ngôn

ngữ giả.

Hàm hiển thị các yêu cầu đăng tin.

Create new ModelAndView

Call method model.setViewName with "managerRequirement"

Initialise lists to nhatroDao.listWaiting

Initialise showNull to "KhÃ´ng cÃ³ yÃªu cáº§u Ä‘Äƒng tin"

If lists.isEmpty

Call method model.addObject with "showNull", showNull

Else

Call method model.addObject with "listUser", lists

EndIf

Return model

Hàm đồng ý yêu cầu đăng tin

Call method nhatroDao.dongy with nhatroid

Return "redirect:/admin/quanlydangtin"

Hàm không đồng ý yêu cầu đăng tin

Call method nhatroDao.khongdongy with nhatroid

Return "redirect:/admin/quanlydangtin"

Hàm tìm kiếm yêu cầu đăng tin

Create new ModelAndView

Initialise lists to nhatroDao.searchYeuCau with searchContent

Initialise searchNull to "ThÃ´ng tin tÃ¬m kiáº¿m khÃ´ng tá»“n táº¡i."

If searchContent.isEmpty or searchContent.equals with " "

Call method model.setViewName with "redirect:/admin/quanlydangtin"

Else

Call method model.setViewName with "managerRequirement"

If lists.isEmpty

Call method model.addObject with "searchNull", searchNull

Else

Call method model.addObject with "lists", lists

EndIf

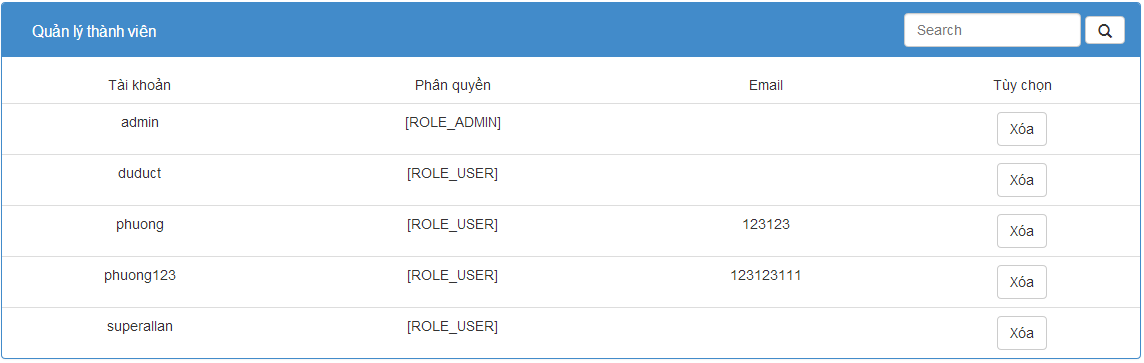
EndIf

Return model

## Quản lý thành viên

**- Mục đích**: Cho phép người dùng quản lý các thành viên.

**- Giao diện**:



1

2

3

4

5

6

7

**- Các thành phần trong giao diện**: ghi vào bảng sau các mô tả về từng thành phần (đã

được đánh số) của giao diện.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Label Quản lý thành viên |  |  |
| 2 | Textbox Search | Null | Tìm kiếm theo giá trị tương đương **VD**: Nhập “ad” kết quả có thể là “admin”. |
| 3 | Button Search |  | Khi được nhấn lấy thông tin từ Textbox Search rồi thực hiện tìm kiếm. |
| 4 | Label Tài khoản |  | Tên tài khoản của người dùng. |
| 5 | Label Phân quyền |  | Cho biết người dùng có quyền gì trong hệ thống. |
| 6 | Label Email |  | Email của người dùng. |
| 7 | Button Xóa |  | Xóa người dùng ra khỏi hệ thống. |

**- Dữ liệu được sử dụng**: liệt kê các bảng trong cơ sở dữ liệu hoặc các cấu trúc dữ liệu

được cần đến bởi chức năng này.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | User và userrole |  |  | xoaThanhVien | listUser |

**- Cách xử lý:**

**Trường hợp 1:**

1. Hiển thị danh sách các thành viên.
2. Người dùng chọn button Xóa để xóa thành viên ra khỏi CSDL.
3. Hệ thống hiển thị thông báo “Yes” “No” cho người dùng chọn
4. Người dùng chọn Yes hệ thống sẽ cập nhật lại trang Quản lý thành viên.
5. Người dùng chọn No hệ thống trở lại bước 1.

**Trường hợp 2:**

1. Hiển thị danh sách các thành viên.
2. Người dùng nhập nội dùng cần tìm kiếm vào Textbox Search.
3. Hệ thống hiển thị kết quả (Nếu không có thì hiện thông báo “Nội dung tìm kiếm không tồn tại”).
4. Người dùng chọn button Xóa để xóa thành viên ra khỏi CSDL.
5. Hệ thống hiển thị thông báo “Yes” “No” cho người dùng chọn
6. Người dùng chọn Yes hệ thống sẽ cập nhật lại trang Quản lý thành viên.
7. Người dùng chọn No hệ thống trở lại bước 1.

**- Hàm/ sự kiện** (nếu có): mô tả giải thuật cho từng biến cố bằng sơ đồ hoặc bằng ngôn

ngữ giả.

Hàm truy xuất thành viên từ CSDL.

Create new ModelAndView

Call method model.setViewName with "managerUser"

Initialise lists to userDao.listUser

Initialise showNull to "Không tồn tại thành viên nào.”

Create new ArrayList

Create new ArrayList

If lists.isEmpty

Call method model.addObject with "showNull", showNull

Else

Initialise size to lists.size

For i is 0, i is less than size, i increments by 1

Initialise s to lists.get with i as UserRole

Create new HashSet

Call method set.add with s.getRole

Call method lists.get with i

EndFor

Call method model.addObject with "listUser", lists

EndIf

Return model

Hàm xóa thành viên.

Call method userDao.xoaThanhVien with username

Return "redirect:/admin/quanlythanhvien"

Hàm tìm kiếm thành viên.

Create new ModelAndView

Initialise lists to userDao.searchThanhVien with content

Initialise searchNull to "ThÃ´ng tin tÃ¬m kiáº¿m khÃ´ng tá»“n táº¡i."

If content.isEmpty or content.equals with " "

Call method model.setViewName with "redirect:/admin/quanlythanhvien"

Else

Call method model.setViewName with "managerUser"

If lists.isEmpty

Call method model.addObject with "searchNull", searchNull

Else

Initialise size to lists.size

For i is 0, i is less than size, i increments by 1

Initialise s to lists.get with i as UserRole

Create new HashSet

Call method set.add with s.getRole

Call method lists.get with i

EndFor

Call method model.addObject with "lists", lists

EndIf

EndIf

Return model

## Đăng xuất

* Mục Đích: cho phép người dùng đăng xuất khỏi hệ thống.
* Giao diện:
* Các thành phần trong giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 3 | Button Đăng xuất |  |  |

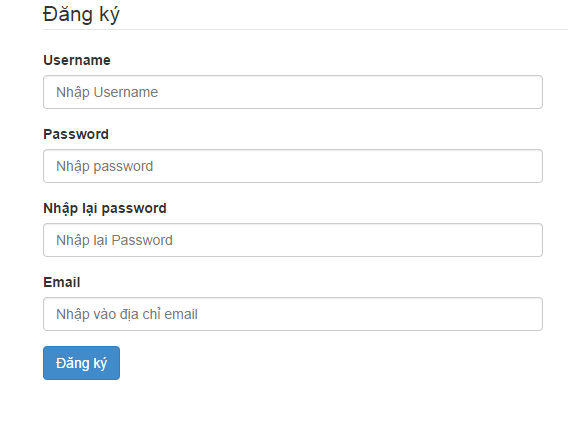
* Cơ sở dữ liệu được sử dụng:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/  Cấu trúc dữ liệu | Phương thúc | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | User |  |  |  | x |

* Cách xử lý:
  1. Người dùng ấn nút thoát
  2. Tài khoản sẽ được đăng xuất ra khỏi hệ thống.
* Hàm/ sự kiện:
* Các ràng buộc nếu có:

## Đăng ký

* Mục Đích: cho phép người dùng đăng ký một tài khoản trong hệ thống
* Giao diện:



* Các thành phần trong giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Textbox User | Null |  |
| 2 | Textbox Password | Null | Phân biệt giữa viết hoa và viết thường. |
| 3 | Textbox Nhập lại password | Null | Phân biệt chữ hoa và chữ thường |
| 4 | Textbox Email |  | Viết đúng định dạng xxx@xxx.xx |
| 5 | Button Đăng ký |  |  |

* Cơ sở dữ liệu được sử dụng:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/  Cấu trúc dữ liệu | Phương thúc | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | User | x |  |  | x |

* Cách xử lý:

1. Hệ thống hiển thị form đăng ký.
2. Người dùng điền thông tin vào form.
3. Người dùng ấn đăng ký.
4. Nếu thông tin sai hoặc nhập không đầy đủ thì hệ thống quay lại trang đăng ký.
5. Thành công! Hệ thống lưu thông tin vào csdl.

* Hàm/ sự kiện:

**Procedure: dangKy**

public String dangKy(@ModelAttribute User user)

Returns String

**Pseudocode**

Call method user.setEnabled with true

Call method userDao.add with user

Create new UserRole

Call method userRole.setUser with user

Call method userRole.setRole with "ROLE\_USER"

Call method userroleDao.add with userRole

Return "home"

**Procedure**: **Constructor**

@RequestMapping(value = "/dangkyHandling", method = RequestMethod.POST)

**Parameters**

* value

Returns RequestMapping

**Pseudocode**

N/A

* Các ràng buộc nếu có:
  1. Các trường trong form không được bỏ trống.

## Xóa nhà trọ

* Mục Đích: cho phép người dùng đăng xuất khỏi hệ thống.
* Giao diện:



* Các thành phần trong giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 3 | Button Đăng xuất |  |  |

* Cơ sở dữ liệu được sử dụng:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/  Cấu trúc dữ liệu | Phương thúc | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | User |  |  |  | x |
| 2 | Nhatro |  |  | x | x |
| 3 | Duong |  |  | x | x |

* Cách xử lý:
  1. Người dùng đăng nhập vào hệ thống.
  2. Người dùng chọn trang quản lý thành viên, vào mục nha trọ của tôi.
  3. Hệ thống hiển thị danh sách nhà trọ do người dùng đăng.
  4. Người dùng bấm nút xóa.
  5. Hệ thống hiển thị thông báo cảnh cáo.
  6. Chọn hủy: hệ thống trở về trang trước đó.
  7. Chọn đồng ý: hệ thống xóa nhaaf trọ ra khỏi cơ sở dữ liệu.
* Hàm/ sự kiện:

**Procedure**: **deleteById**

@Transactional

public void deleteById(Integer id)

**Parameters**

* id - Integer

Returns Transactional

**Pseudocode**

Initialise nhaTro to findById with id

Call

* Các ràng buộc nếu có:

## Xem thông tin nhà trọ đã đăng

* Mục Đích: cho phép người dùng xem thông tin nhà trọ đã đăng.
* Giao diện:



* Các thành phần trong giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Button Search |  |  |
| 2 | Textbox Search | Null |  |
| 3 | Button Xem |  |  |
| 4 | Button Xóa |  |  |

* Cơ sở dữ liệu được sử dụng:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | Nhatro |  |  |  | X |
| 2 | Principal |  |  |  | X |
| 3 | User |  |  |  | X |
| 4 | Thich |  |  |  | X |
| 5 | Loaiphong |  |  |  | X |
| 6 | Comment |  |  |  | X |
| 7 | Hinh |  |  |  | X |

* Cách xử lý:
  1. Đăng nhập vào hệ thống.
  2. Vào trang quản lý thành viên.
  3. Vào trang nhà trọ đã đăng
  4. Hệ thống hiển thị danh sách nhà trọ đã đăng.
  5. Người dùng ấn nút xem
  6. Hệ thống chuyển sang xem nhà trọ.
* Hàm/ sự kiện:

**Procedure**: **Constructor**

@RequestMapping(value = "/danhsachnhatro", method = RequestMethod.GET)

Parameters

* - value

Returns RequestMapping

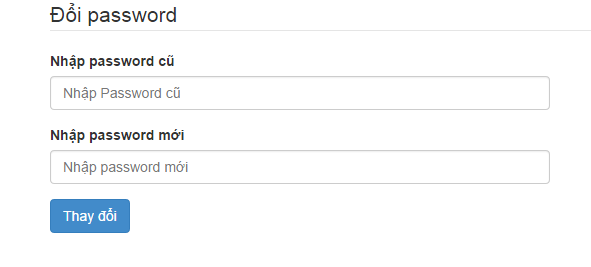
**Pseudocode**

N/A

* Các ràng buộc nếu có:

## Sửa thông tin (password)

* Mục Đích: cho phép người dùng đăng đổi password của tài khoản trong hệ thống.
* Giao diện:
* Các thành phần trong giao diện:



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Loại điều khiển | Giá trị mặc định | Lưu ý |
| 1 | Label Nhập password cũ |  |  |
| 2 | Label Nhập password cũ |  |  |
| 3 | Textbox oldpass | Null | Phân biệt hoa thường |
| 4 | Textbox newpass | Null | Phân biệt hoa thường |
| 5 | Button Thay đổi |  |  |

* Cơ sở dữ liệu được sử dụng:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên bảng/  Cấu trúc dữ liệu | Phương thúc | | | |
| Thêm | Sửa | Xóa | Truy vấn |
| 1 | User |  | x |  | x |

* Cách xử lý:
  1. Người dùng đăng nhập vào hệ thống.
  2. Người dùng đến trang sửa thông tin.
  3. Hệ thống hiển thị trang đổi passsword.
  4. Người dùng điền đầy đủ thông tin vào.
  5. Người dùng ấn nút hoàn thành.
  6. Hệ thống hiển thị thông báo thành công.
  7. Nếu thông tin sai hay nhập thiếu hệ thống sẽ thông báo lỗi.
* Hàm/ sự kiện:

**Procedure**: **suaThongTin**

public String suaThongTin(HttpServletRequest request, Principal principal)

**Parameters**

* request - HttpServletRequest
* principal - Principal

Returns String

**Pseudocode**

Initialise oldPassword to request.getParameter with "oldPass"

Initialise newPassword to request.getParameter with "newPass"

Initialise user to userDao.findUserbyUserName with principal.getName

If oldPassword is equal to user.getPassword

Call method user.setPassword with newPassword

Call method userDao.save with user

Else

Return "redirect:/thanhvien/suathongtin"

EndIf

Return "redirect:/login"

* Các ràng buộc nếu có:

# Bảng tham khảo tới các yêu cầu

|  |  |  |
| --- | --- | --- |
| STT | Thiết kế chức năng | Đáp ứng yêu cầu chức năng |
| 1 | 5.2 Tìm kiếm nhà trọ | 4.3 Tìm kiếm nhà trọ |
| 2 | 5.3 Yêu cầu đăng thông tin nhà trọ | 4.4 Yêu cầu đăng thông tin nhà trọ |
| 3 | 5.4 Xem nhà trọ | 4.5 Xem thông tin nhà trọ |
| 4 | 5.5 Like nhà trọ |
| 5 | 5.6 Bình luận nhà trọ |
| 6 | 5.7 Sửa thông tin nhà trọ | 4.6 Sửa thông tin nhà trọ |
| 7 | 5.8 Quản lý yêu cầu đăng tin | 4.7 Quản lý yêu cầu đăng thông tin nhà trọ |
| 8 | 5.9 Quản lý thành viên | 4.9 Quản lý thông tin nhà trọ |

# Các phụ lục