# MỤC LỤC

[MỤC LỤC 1](#_Toc513546612)

[DANH MỤC CÁC BẢNG BIỂU 2](#_Toc513546613)

[CHƯƠNG 1 : GIỚI THIỆU 2](#_Toc513546614)

[1.1 Tổng quan 2](#_Toc513546615)

[1.2 Mục tiêu đề tài 3](#_Toc513546616)

[1.3 Mô tả yêu cầu chức năng 3](#_Toc513546619)

[1.3.1 Yêu cầu chức năng: 3](#_Toc513546620)

[1.3.2 Yêu cầu phi chức năng 3](#_Toc513546621)

[1.3.3 Cách chèn nhãn cho bảng biểu và hình vẽ 5](#_Toc513546622)

[1.1.1 Hướng dẫn chèn công thức và nhãn 7](#_Toc513546623)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 7](#_Toc513546624)

[2.1 Ngôn ngữ lập trình JAVA 7](#_Toc513546625)

[2.2 Kết nồi JDBC 8](#_Toc513546626)

[2.2.1 DriverManager: 9](#_Toc513546627)

[2.2.2 Driver: 9](#_Toc513546628)

[2.2.3 Connection : 9](#_Toc513546629)

[2.2.4 Statement : 9](#_Toc513546630)

[2.2.5 ResultSet: 9](#_Toc513546631)

[2.3 Spring MVC 9](#_Toc513546632)

[CHƯƠNG 3 : PHÂN TÍCH 11](#_Toc513546633)

[3.1 Sơ đồ Usecase 11](#_Toc513546634)

[3.1.1 Đặc tả Use Case 11](#_Toc513546635)

[CHƯƠNG 4 : THIẾT KẾ VÀ HIỆN THỰC 41](#_Toc513546636)

[4.1 Sơ đồ Class 41](#_Toc513546637)

[4.2 Sơ đồ Sitemap 41](#_Toc513546638)

[4.3 Thiết kết giao diện phác thảo 42](#_Toc513546639)

[4.3.1 Giao diện danh mục 42](#_Toc513546640)

[4.3.2 Giao diện chi tiết báo 44](#_Toc513546641)

[4.3.3 Giao diện giỏ báo 45](#_Toc513546642)

[4.3.4 Giao diện thanh toán 45](#_Toc513546643)

[4.3.5 Giao diện quản lý đợn hàng 46](#_Toc513546644)

[4.3.6 Giao diện quản lý người dùng 46](#_Toc513546645)

[4.3.7 Giao diện quản lý báo 47](#_Toc513546646)

[CHƯƠNG 5 : KẾT LUẬN 47](#_Toc513546656)

[1.2 Kết quả đạt được 47](#_Toc513546657)

[1.3 Hạn chế của đồ án 47](#_Toc513546658)

[1.4 Hướng phát triển 47](#_Toc513546659)

[TÀI LIỆU THAM KHẢO 47](#_Toc513546660)

[PHỤ LỤC 49](#_Toc513546661)

# DANH MỤC CÁC HÌNH VẼ

Hình 2‑1 Thao tác cập nhật mục lục 4

Hình 2‑2 Cách chèn nhãn cho hình 5

Hình 2‑3 Cách tạo một nhãn mới 6

Hình 2‑4 Cách tham chiếu đến một nhãn 6

# DANH MỤC CÁC BẢNG BIỂU

Bảng 2‑1 Tên bảng 10

# : GIỚI THIỆU

## Tổng quan

Thị trường nhu cầu thực tế về nhu cầu tìm kiếm thông tin hiện nay là khá cao, nhất là khoản sách báo, tạp chí giấy vẫn còn rất ổn định, nhu cầu về khoản mua sắm trực tuyến. Cùng với đó là việc tìm hiểu về Spring MVC là lý cho chúng tôi phát triển sản phẩm này.

Trang web về đặt báo online, giúp cho người dùng có thể đặt báo một cách thuận tiện, nhanh chóng, mang những giá trị tốt nhất cho khách hàng. Trang web cung cấp số lượng báo khổng lồ trong và ngoài nước, cùng với thao tác đặt báo tiện lợi, tiết kiệm thời gian thay vì phải đến các sạp báo hay những trang web tương ứng. Cùng với hệ thống quản lý online thuận lợi cho hệ thống quản trị viên, giúp nâng cao năng suất làm việc và quản lý, đẩy nhanh phát triển cho doanh nghiệp sử dụng.

## Mục tiêu đề tài

Thực diện các chúc năng chính quan trọng của hệ thống dựa trên mô hình Spring MVC và JDBC như: Chức năng thêm giỏ báo, thanh toán, đăng ký, đăng nhập, hiện danh sách báo, thêm báo, xác nhận đơn hàng.

Đạt được 90% sự hoàn thiện đối với mỗi chức năng.

## Mô tả yêu cầu chức năng

### Yêu cầu chức năng:

* Đăng nhập
* Tạo tài khoản
* Tìm kiếm
* Xem danh mục
* Xem chi tiết
* Thống kê doanh thu
* Thanh toán
* Thêm giỏ báo
* Xóa giở báo
* Quản lý giỏ báo
* Quản lý khách hàng
* Quản lý đơn hàng

### Yêu cầu phi chức năng

* Trang web đẹp mắt, thâm thiện người dùng
* Dễ dàng thao tác 1 chức năng ( tầm 15 giây 1 thao tác)
* Trang quản lý rõ ràng, dễ dàng tìm kiếm và xử lý dữ liệu trong vòng 10 giây
* : CƠ SỞ LÝ THUYẾT

**Lưu ý:** Chương này trình bày các lý thuyết mà sinh viên đã tìm hiểu, nghiên cứu và ứng dụng trong đề tài.

File mẫu này trình bày các cách định dạng trong tài liệu báo cáo, hướng dẫn cách cập nhật nội dung trong bài báo cáo theo cách soạn thảo văn bản WORD.



Hình 2‑1 Thao tác cập nhật mục lục

Phần Header: thay đổi chuyên ngành cho phù hợp

Phần Footer – phần chữ có nền màu vàng, sinh viên click chuột vào và thay đổi nội dung cho đúng với của thông tin của mình như Họ và tên, Khóa đào tạo.

MỞ ĐẦU – Phần này bắt buộc phải có, các tiêu đề mục giữ nguyên, nội dung viết theo hướng dẫn

**Heading 1** là tiêu đề gốc (cấp 1, phím tắt định dạng là **Ctrl +1**), thường sử dụng để định nghĩa cho tiêu đề của MỤC LỤC, tiêu đề CHƯƠNG, tiêu đề TÀI LIỆU THAM KHẢO, tiêu đề PHỤ LỤC và một số tiêu đề cùng cấp khác (nếu có).

Để định nghĩa một đoạn văn bản theo Style nào đó chỉ cần gõ đoạn văn bản đó rồi chọn style. Trong một Style nếu muốn cắt dòng thì đưa con trỏ đến vị trí cần cắt rồi nhấn SHIFT+ENTER thay vì nhấn ENTER.

Style cấp 1 được đánh số mặc định là Chương và số chương. Nếu sử dụng style Heading 1 cho tiêu đề Mục lục thì gõ chữ “mục lục” sau đó chọn style Heading 1. Lúc này đoạn văn sẽ thành “CHƯƠNG 1 MỤC LỤC” sau đó phải xóa phần số đi bằng cách đưa con trỏ đến đầu chữ Mục lục và nhấn Backspace. Nếu sử dụng style Heading 1 cho tiêu đề Chương thì gõ nội dung của tiêu đề chương VD “GIỚI THIỆU” sau đó chọn style Heading 1. Lúc này đoạn văn sẽ thành

“CHƯƠNG 1 GIỚI THIỆU”

Nếu muốn thành hai dòng như trình bày ở chương này thì đưa con trỏ đến đầu chữ GIỚI THIỆU.... rồi nhấn SHIFT+ENTER.

**Heading 2** là tiêu đề con (cấp 2, phím tắt định dạng là **Ctrl +2**)

**Heading 3** là tiêu đề cấp 3 (phím tắt định dạng là **Ctrl +3**)

Đối với nhãn của bảng biểu, hình vẽ và công thức nên tạo như sau:

### Cách chèn nhãn cho bảng biểu và hình vẽ

Để tạo nhãn bảng, xem kết quả ở Error: Reference source not found: Chọn bảng cần gán nhãn, Click chuột phải chọn Insert Caption và chọn mẫu của nhãn (nếu chưa có thì tạo mới và thiết lập kiểu hiển thị và kiểu đánh số)



Hình 2‑2 Cách chèn nhãn cho hình

Nếu chưa có nhãn Bảng thì thì chọn New Label rồi làm nhưn hình dưới đây



Hình 2‑3 Cách tạo một nhãn mới

Để tham chiếu đến nhãn của bảng nào đó thì chọn Insert->Cross preference và chọn đến nhãn bảng cần liên kết xem Error: Reference source not found. Cách làm tương tự với hình vẽ, công thức, mục, tài liệu tham khảo.



Hình 2‑4 Cách tham chiếu đến một nhãn

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Bảng 2‑1 Tên bảng

### Hướng dẫn chèn công thức và nhãn

|  |  |
| --- | --- |
|  | (1.1) |

Cũng tương tự như cách chèn nhãn của hình vẽ và bảng biểu, chèn nhãn cho công thức có phần hơi phức tạp. Trong văn bản này, giải pháp đơn giản là tạo một bảng có 2 cột, cột 1 là công thức, cột 2 là nhãn công thức.

Khi thêm công thức mới chỉ cần copy nhãn đã tạo ở công thức mẫu ( 1 .1 rồi paste và o cột thứ 2 của bảng chứa công thức, nhãn công thức sẽ tự động cập nật khi in hoặc có thể click chuột phải vào số cần cập nhật rồi chọn Update Field.

# : CƠ SỞ LÝ THUYẾT

## Ngôn ngữ lập trình JAVA

Là một ngôn ngữ hoàn toàn hướng đối tượng. Ngôn ngữ đa nền cho phép một chương trình có thể thực thi trên các hệ điều hành khác nhau (MS Windows, UNIS, Linux) mà không phải biên dịch lại chương trình. Phương châm của java là “Viết một lần, chạy trên nhiều nền “ (Write Once, Run Anywhere).

Ngôn ngữ đa luồng, cho phép trong một chương trình có thể có nhiều luồng điều khiển được thực thi song song nhau, rất hữu ích cho các xử lý song song.

Ngôn ngữ phân tán, cho phép trong các đối tượng của một ứng dụng được phân bố và thực thi trên các máy tính khác nhau.

Ngôn ngữ động, cho phép mã lệnh của một chương trình được tải từ một máy tính về máy của người yêu cầu thực thi chương trình.

Ngôn ngữ an toàn, tất cả các thao tác truy xuất vào các thiết bị vào ra đều được thực hiện trên máy ảo nhờ đó hạn chế các thao tác ngủy hiểm cho máy tính thật.

## Kết nồi JDBC

**JDBC** (Java Database Connectivity) là một API tiêu chuẩn dùng để tương tác với các loại cơ sở dữ liệu quan hệ. **JDBC** có một tập hợp các class và các Interface dùng cho ứng dụng **Java** có thể nói chuyện với các cơ sở dữ liệu.

http://viettuts.vn/images/java/java-jdbc/java-jdbc.png

Hình 2‑1 Sơ đồ JDBC

Các thành phần của JDBC Api về cơ bản bao gồm:

### DriverManager:

* + Là một class, nó dùng để quản lý danh sách các **Driver**(database drivers).

### Driver:

* + Là một Interface, nó dùng để liên kết các liên lạc với cơ sở dữ liệu, điều khiển các liên lạc với database. Một khi Driver được tải lên, lập trình viên không cần phải gọi nó một cách cụ thể.

### Connection :

* + Là một Interface với tất cả các method cho việc liên lạc với database. Nó mô tả nội dung liên lạc. tất cả các thông tin liên lạc với cơ sở dữ liệu là thông qua chỉ có đối tượng ***Connection***.

### Statement :

* + Là một Interface, gói gọn một câu lệnh SQL gửi tới cơ sở dữ liệu được phân tích, tổng hợp, lập kế hoạch và thực hiện.

### ResultSet:

* + ***ResultSet*** đại diện cho tập hợp các bản ghi lấy do thực hiện truy vấn.

## Spring MVC

Spring MVC là một Framework / 1 Project mã nguồn mở của Spring.

Spring MVC Framewrok cung cấp kiến truc MVC (Model-View-Controller) và các component được sử dụng để phát triển các ứng dụng web một cách linh hoạt và giảm sư

* Bất kỳ request nào tới ứng dụng web đều sẽ được gửi tới Front Controller (Dispatcher Servlet)
* Front Controller sẽ sử dụng Handler Mapping để biết được controller nào sẽ xử lý request đó
* Controller nhận request, gọi tới các class service thích hợp để xử lý yêu cầu.
* Sau khi xử lý xong, Controller sẽ nhận được model từ tầng Service hoặc tầng DAO.
* Controller gửi model vừa nhận được tới Front Controller (Dispatcher Servlet)
* Dispatcher Servlet sẽ tìm các mẫu view, sử dụng view resolver và truyền model vào nó.
* View template, model, view page được build và gửi trả lại Front Controller
* Front Controller gửi một page view tới trình duyệt để hiển thị nó cho người dùng.

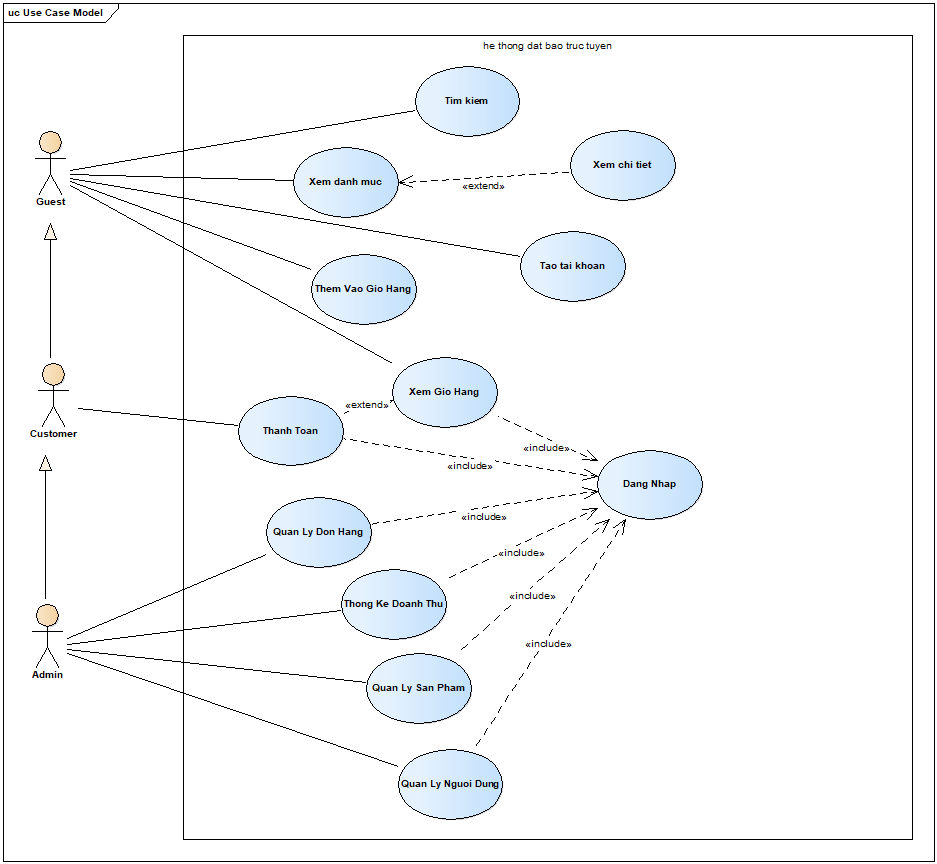
Hình 2‑1 Mô hình Spring MVC

Trong Mô hình MVC thì:

* **Model**: là các file POJO, Service, DAO thực hiện truy cập database, xử lý business
* **View**: là các file JSP, html…
* **Control**: là Dispatcher Controller, Handler Mapping, Controller – thực hiện điều hướn các request.

# : PHÂN TÍCH

## Sơ đồ Usecase



### Đặc tả Use Case

#### Đặc tả các yêu cầu chức năng

##### Đăng nhập

|  |  |
| --- | --- |
| **Use case:** *Đăng nhập* | |
| Mục đích: | Xử lí đăng nhập cho Khách hàng |
|  | Khách hàng điền thông tin tài khoản đăng nhập |
| Tác nhân: | Người dùng (Guest) |
| Điều kiện sau: | Cấp quyền cho ‘Người dùng’ theo quyền của tài khoản, trả về giao diện trước bước đăng nhập. |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng nhập tài khoản và mật khẩu 2. Khách chọn chức năng “đăng nhập” 3. Hệ thống kiểm tra thông tin đăng nhập 4. Hệ thống lấy dữ liệu và giao diện trước đó cho người dùng 5. Hệ thống cấp quyền và hiển thị giao diện trước đó cho Người dùng. |
| Luồng sự kiện phụ (Alternative Flows): | 1. Nếu thông tin đăng nhập sai hệ thống hiển thị thông báo cho khách hàng. Còn nếu trên 3 lần sai hệ thống sẽ khóa tài khoản và gởi mail cảnh báo về chủ tài khoản sau đó hiển thị giao diện cảnh báo. |





##### Tạo tài khoản

|  |  |
| --- | --- |
| **Use case:** *Tạo tài khoản* | |
| Mục đích: | Xử lí tạo tài khoản cho khách |
| Mô tả: | Khách điền thông tin tài khoản để tạo |
| Tác nhân: | Người dùng (Guest) |
| Điều kiện sau: | Khách sẽ được cấp tài khoản đăng nhập |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng chọn chức năng “tạo tài khoản” 2. Hệ thống hiển thị giao diện đăng kí 3. Khách nhập thông tin đăng kí 4. Khách chọn chức năng đăng kí 5. Hệ thống tạo tài khoản mới cho khách 6. Hệ thống gởi mail xác nhận cho khách và lưu thông tin vào CSDL 7. Hệ thống hiển thị thông báo đăng kí thành công 8. Hệ thống chuyển sang giao diện đăng nhập |
| Luồng sự kiện phụ (Alternative Flows): | 4.1 Hệ thống thông báo thông tin đăng kí thông hợp lệ  4.2 Khách sửa lại thông tin đăng kí |

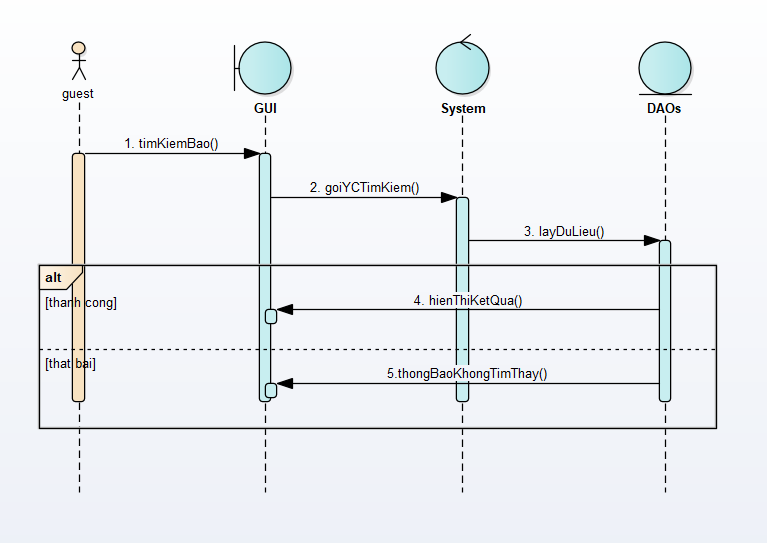




##### Tìm kiếm báo

|  |  |
| --- | --- |
| **Use case:** *Tìm kiếm báo* | |
| Mục đích: | Thao tác tìm kiếm báo thuận tiện |
| Mô tả: | Khách hàng nhập tên báo cần tìm |
| Tác nhân: | Người dùng (Guest) |
| Điều kiện sau: | Hiện thị kết quả tìm được |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng nhập thông tin cần tìm 2. Người dùng chọn chức năng “tìm kiếm” 3. Hệ thống xác nhận và tìm kiếm thông tin 4. Hệ thống hiển thị danh sách tìm được |
| Luồng sự kiện phụ (Alternative Flows): | 1. Nếu không tìm thấy hiện thông báo với khách |





##### Xem danh mục

|  |  |
| --- | --- |
| **Use case:** *Xem danh mục* | |
| Mục đích: | Khách thao tác với danh mục |
| Mô tả: | Khách tra cứu danh mục |
| Tác nhân: | Người dùng (Guest) |
| Điều kiện sau: | Hiện thị danh mục |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng chọn chức năng “Xem danh mục” 2. Hệ thống lấy dữ liệu 3. Hệ thống hiển thị danh sách báo của danh mục |

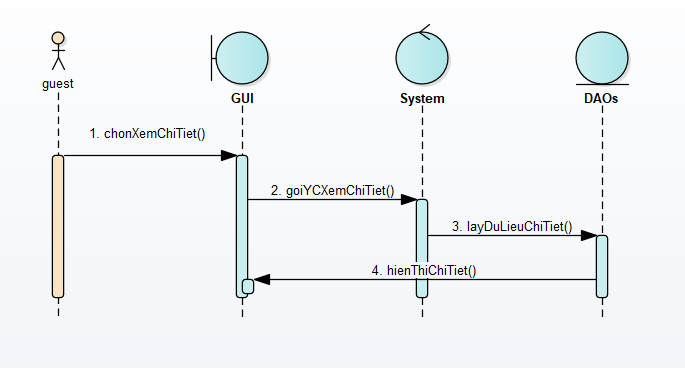




##### Xem chi tiết

|  |  |
| --- | --- |
| **Use case:** *Xem chi tiết* | |
| Mục đích: | Khách muốn xem chi tiết báo |
| Mô tả: | Khách tra cứu chi tiết giới thiệu về báo |
| Tác nhân: | Người dùng (Guest) |
| Điều kiện sau: | Hiện thị chi tiết báo |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng chọn chức năng “Xem chi tiết” 2. Hệ thống lấy dữ liệu 3. Hệ thống hiển thị chi tiết cần xem |

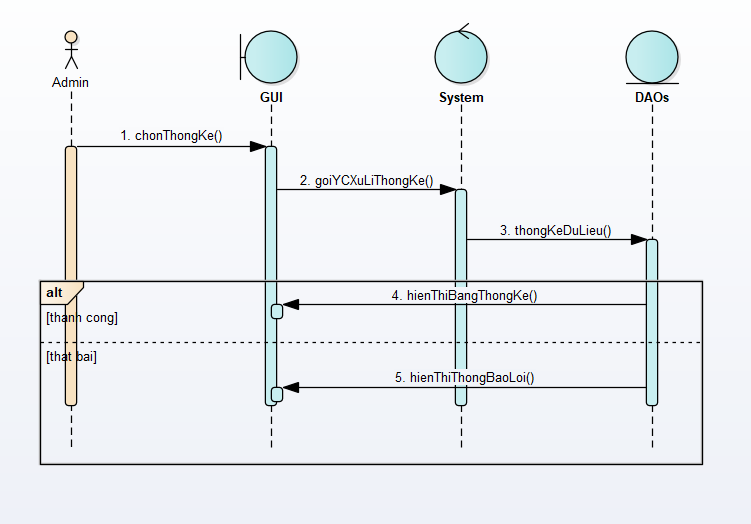




##### Thống kê doanh thu

|  |  |
| --- | --- |
| **Use case:** *Thống kê doanh thu* | |
| Mục đích: | Admin muốn thống kê doanh thu trong tháng |
| Mô tả: | Admin thao tác thống kê |
| Tác nhân: | Admin |
| Điều kiện sau: | Hiện thị bảng thống kê |
| Luồng sự kiện chính (Basic flows) | 1. Admin chọn chức năng “thống kê” 2. Hệ thống hiển thị giao diện thống kê 3. Admin chọn mục cần thống kê 4. Hệ thống xử lí dữ liệu 5. Hệ thống hiển thị bảng thống kê |





##### Thanh toán

|  |  |
| --- | --- |
| **Use case:** *Thanh toán* | |
| Mục đích: | Thanh toán dơn hàng cho thành viên |
| Mô tả: | Khách hàng điền thông tin thanh toán cho đơn hàng của mình |
| Tác nhân: | Khách hàng (Customer) |
| Điều kiện sau: | Hóa đơn sẽ được tạo và gửi yêu cầu về Admin |
| Luồng sự kiện chính (Basic flows) | 1. Khách hàng chọn chức năng “Thanh toán” 2. Hệ thống hiện giao diện GDThanhToan 3. Khách hàng chọn phương thức thanh toán. 4. Khách hàng nhập thông tin thanh toán vào mẫu. 5. Khách hàng chọn chức năng ‘Đặt báo ngay’. 6. Lưu thong tin thanh toán và thông báo thanh toán thành công 7. Hệ thống chuyển về GDTrangChu |





##### Thêm giỏ báo

|  |  |
| --- | --- |
| **Use case:** *Thêm giỏ báo* | |
| Mục đích: | Thêm báo vào giỏ hàng người dùng |
| Mô tả: | Người dùng nhập thông tin và chọn thêm giỏ hàng cho báo mình muốn. Sau đó báo sẽ vào giỏ. |
| Tác nhân: | Người dùng (Guest) |
| Điều kiện sau: | Giỏ báo sẽ có báo của người dùng vừa thêm |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng nhập thông tin đặt báo vào mẫu trong giao diện chi tiết báo 2. Người dùng chọn chức năng ‘thêm giỏ báo’ 3. Hệ thống sẽ lưu thông tin giỏ báo vào session 4. Hệ thống chuyển qua giao diện giỏ báo. |





##### Xóa giỏ báo

Nội dung tiểu mục

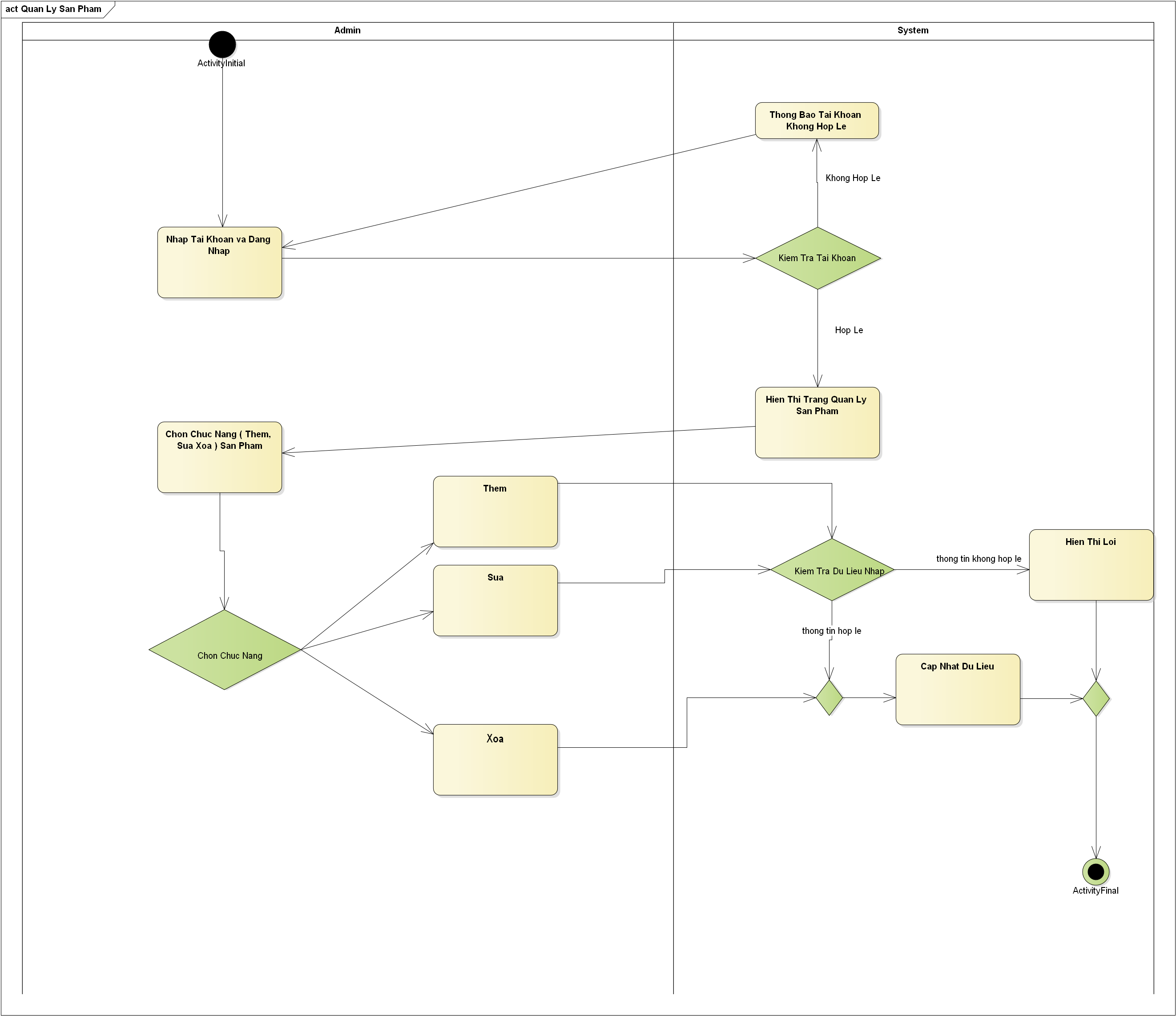
|  |  |
| --- | --- |
| **Use case:** *Xóa giỏ báo* | |
| Mục đích: | Xóa tờ báo đã đặt có trong giỏ hàng |
| Mô tả: | Người dùng chọn nút ‘xóa’ trong giao diện giỏ báo, báo sẽ biến mất khỏi giỏ báo |
| Tác nhân: | Người dùng (Guest) |
| Điều kiện sau: | Giỏ hàng sẽ mất báo của người dùng vừa xóa |
| Luồng sự kiện chính (Basic flows) | 1. Người dùng chọn chức năng ‘Xóa’ trong giao diện giỏ báo 2. Hiện thông báo xác nhận muốn xóa không. 3. Hệ thống sẵn sàng cho hành động kế tiếp tại giao diện giỏ báo |

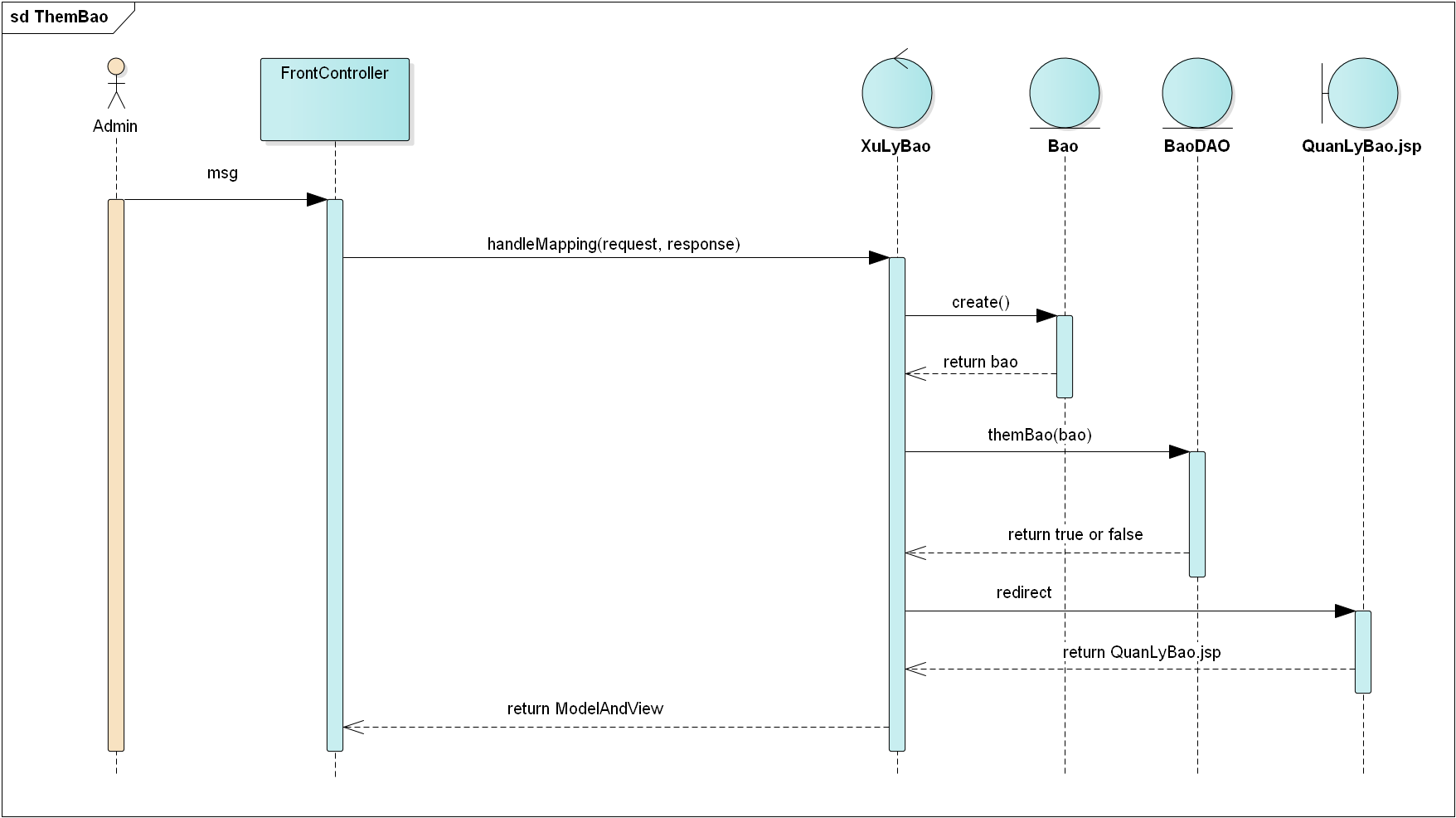


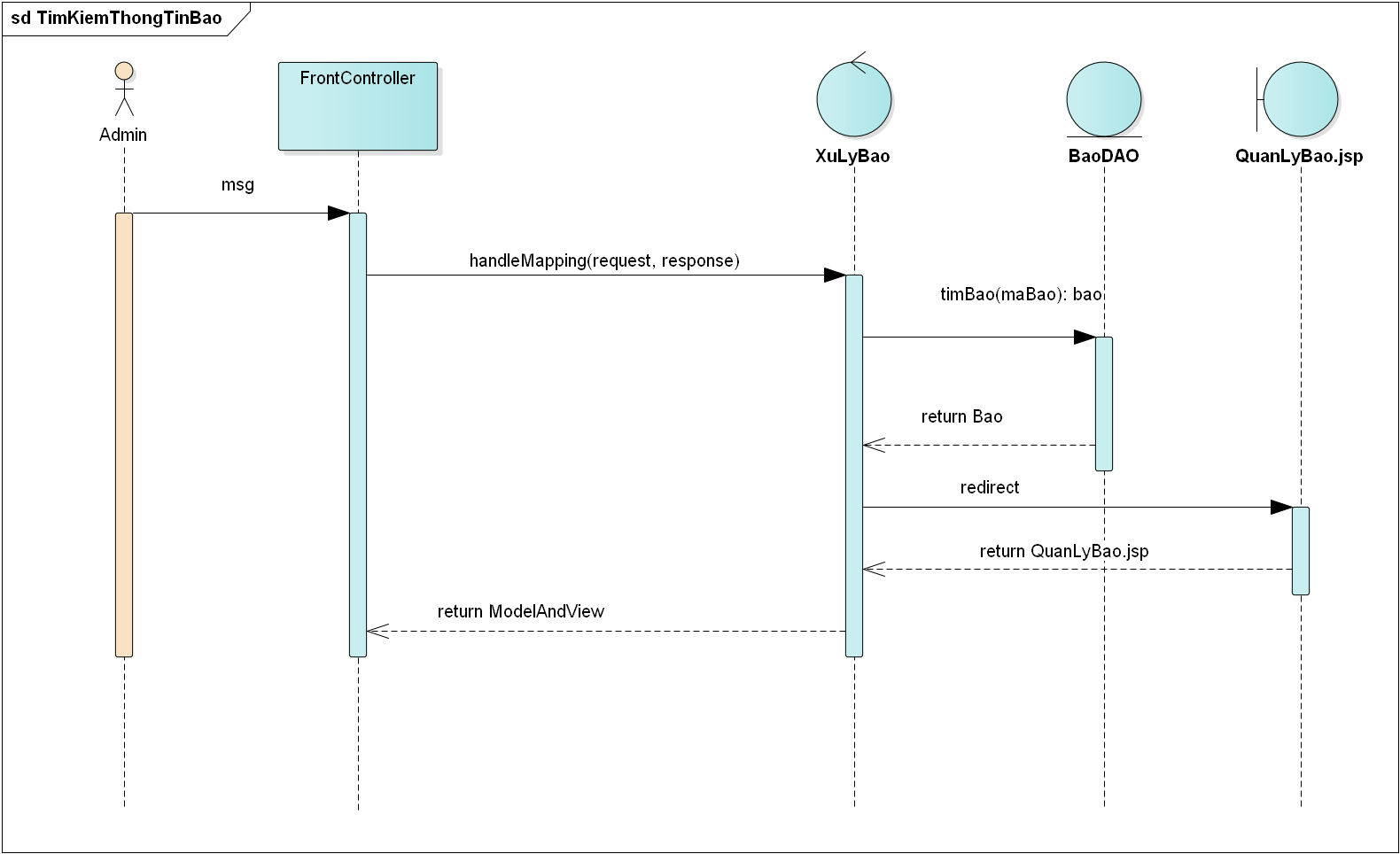


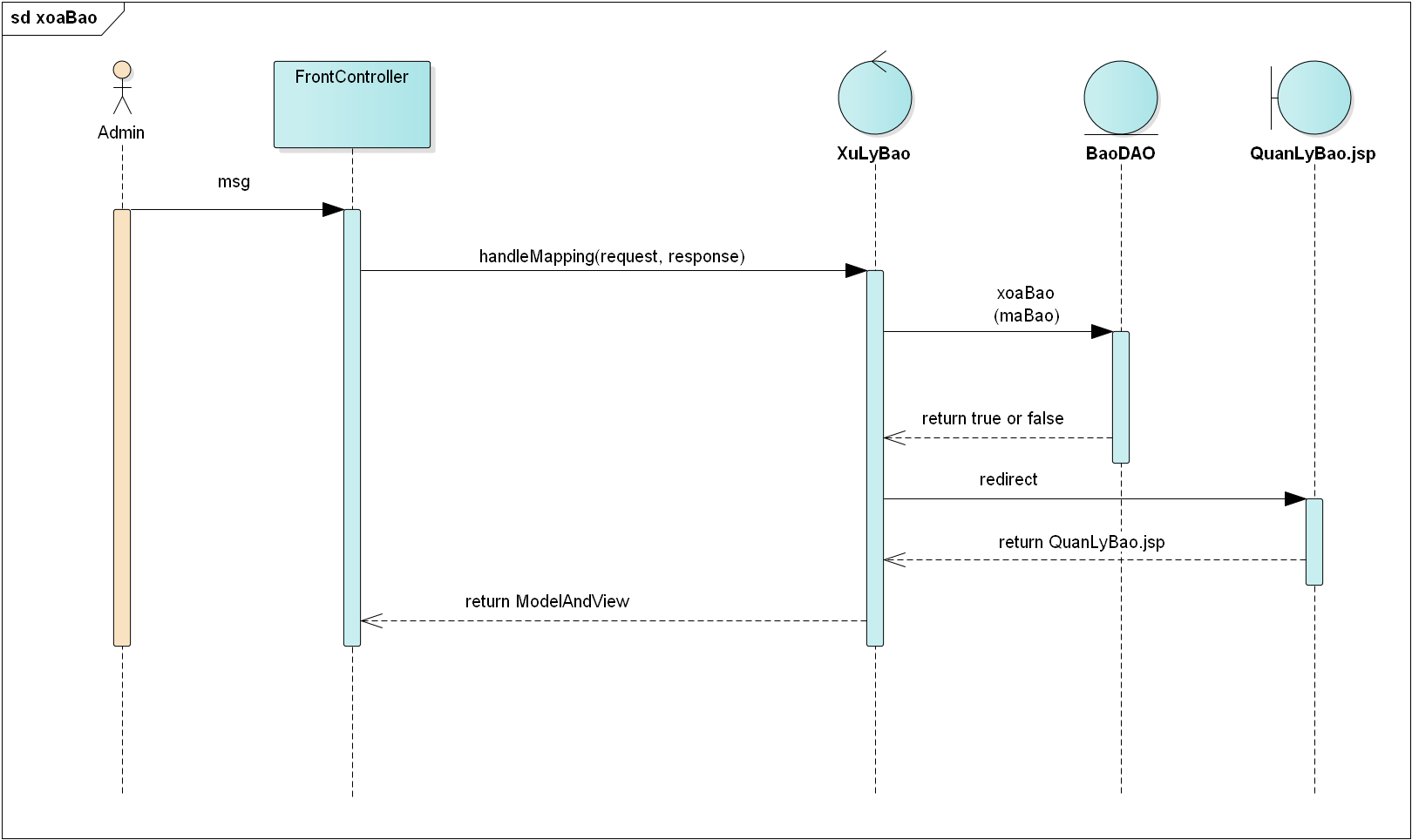
##### Quản Lý Báo

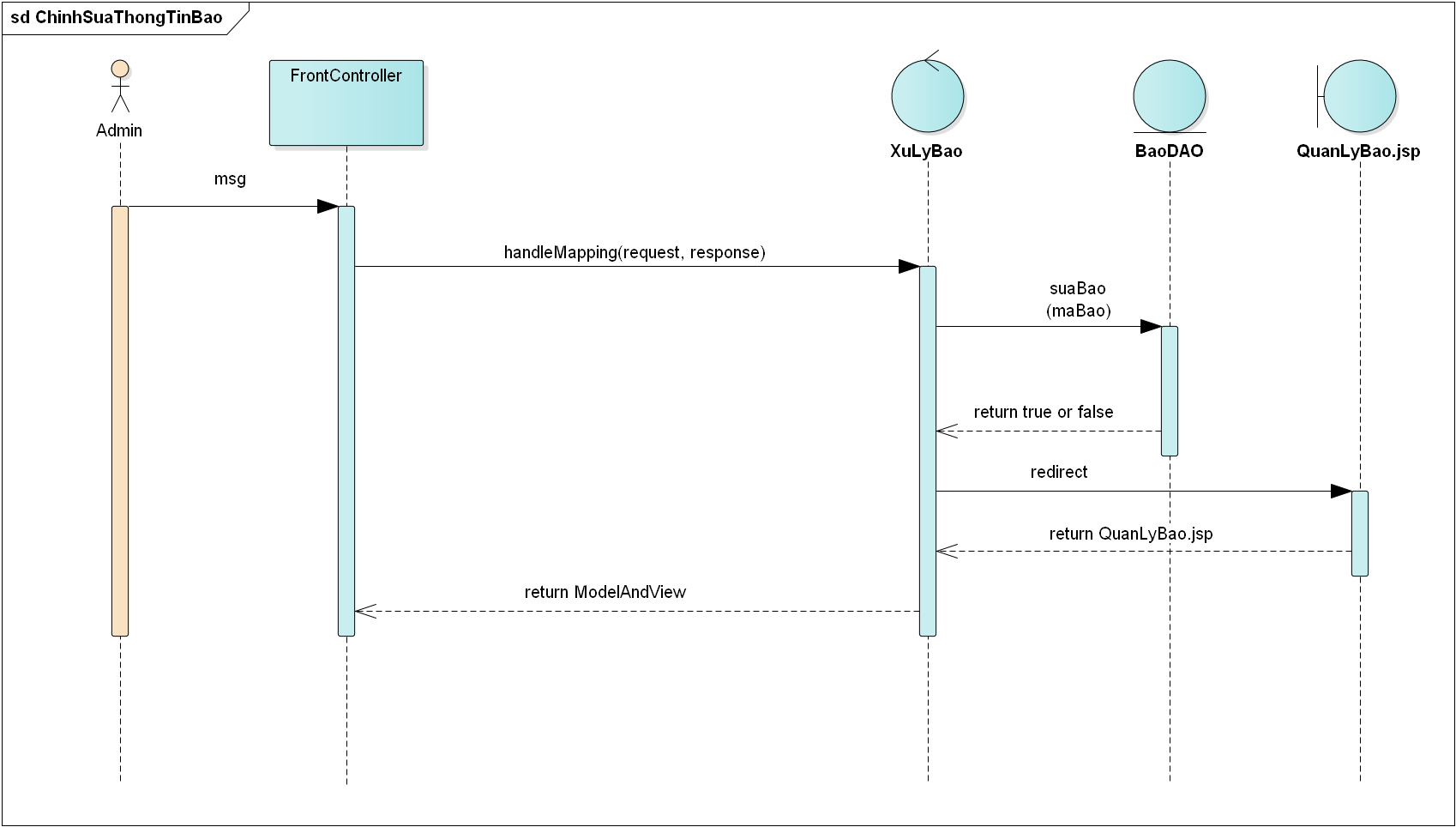
|  |  |
| --- | --- |
| **Use case: Quản Lý Báo** | |
| Mục đích: | **Admin thực hiện chức năng Quản Lý Báo** |
| Mô tả: | Admin vào trang “quản lý báo” để thực hiện thao tác thêm, xóa, sửa, cập nhật Báo |
| Tác nhân: | Admin |
| Điều kiện sau: | Cập nhật lại thông tin vào CSDL |
| Luồng sự kiện chính (Basic flows) | 1. Hệ thống hiển thị trang quản lý Báo 2. Admin chọn chức năng ( Tìm kiếm, Thêm, Sửa, Xóa ) 3. Hệ thống thực hiện chức năng theo yêu cầu. 4. Hệ thống cập nhật dữ liệu vào CSDL. 5. Hệ thống quay lại trang quản lý Báo. |





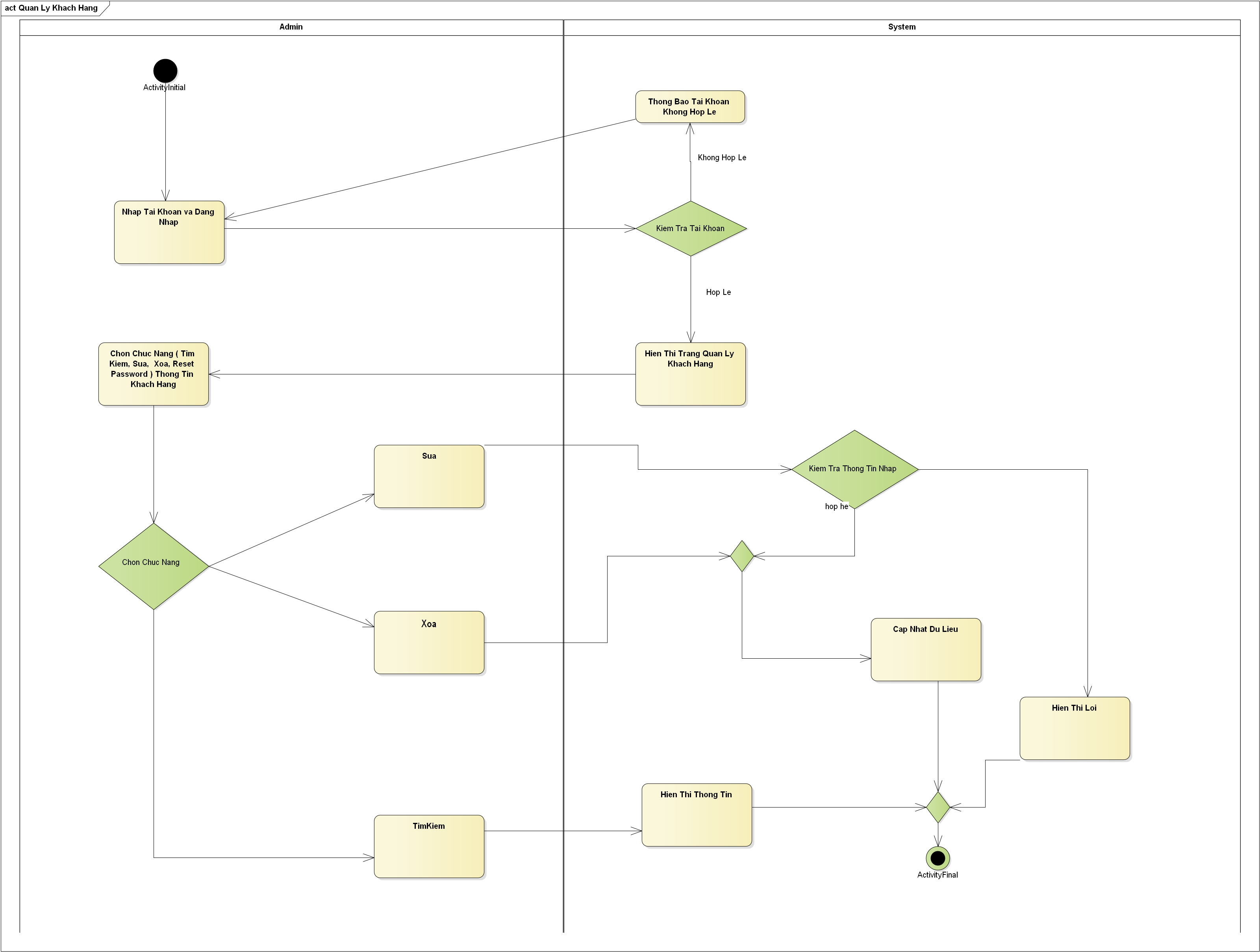


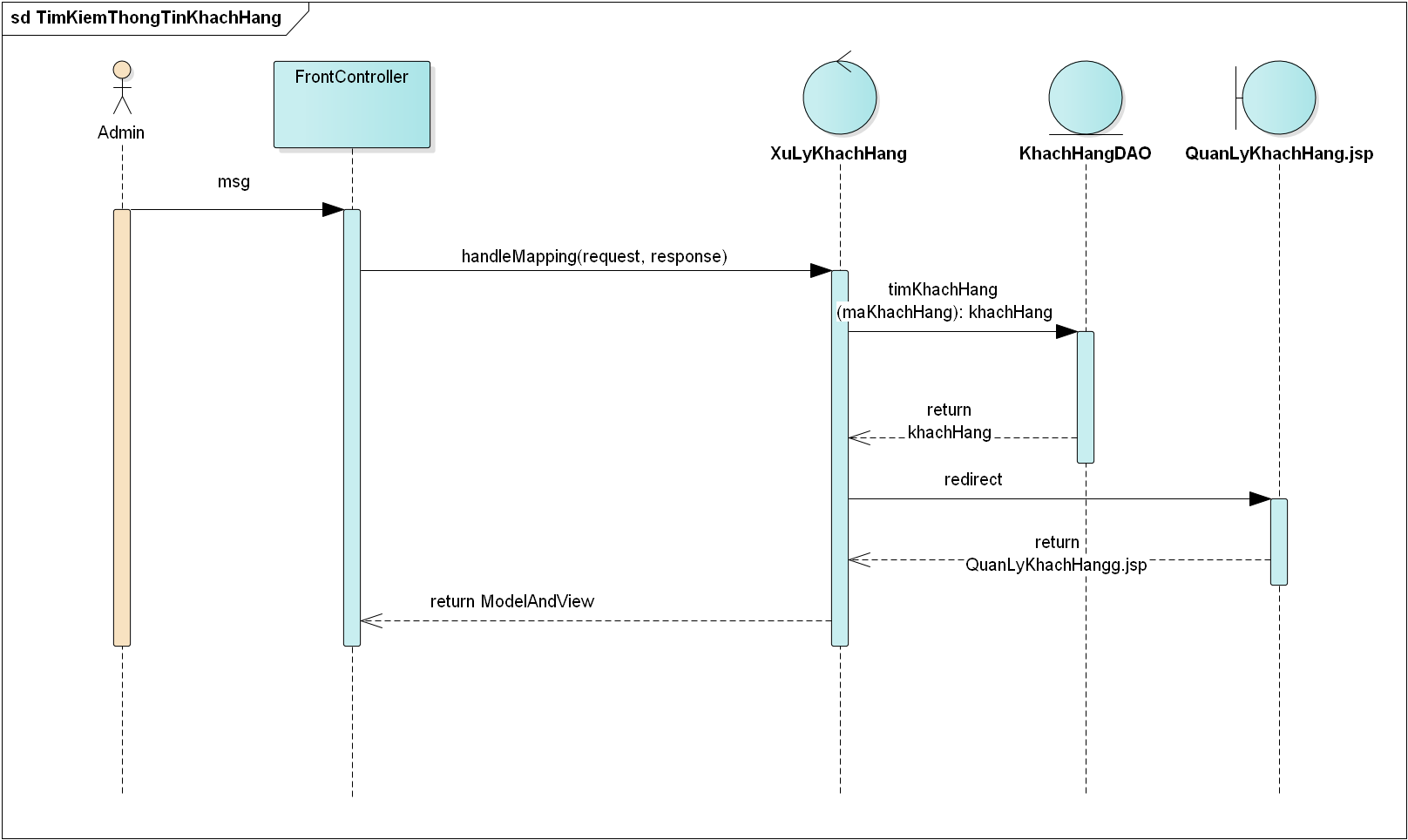
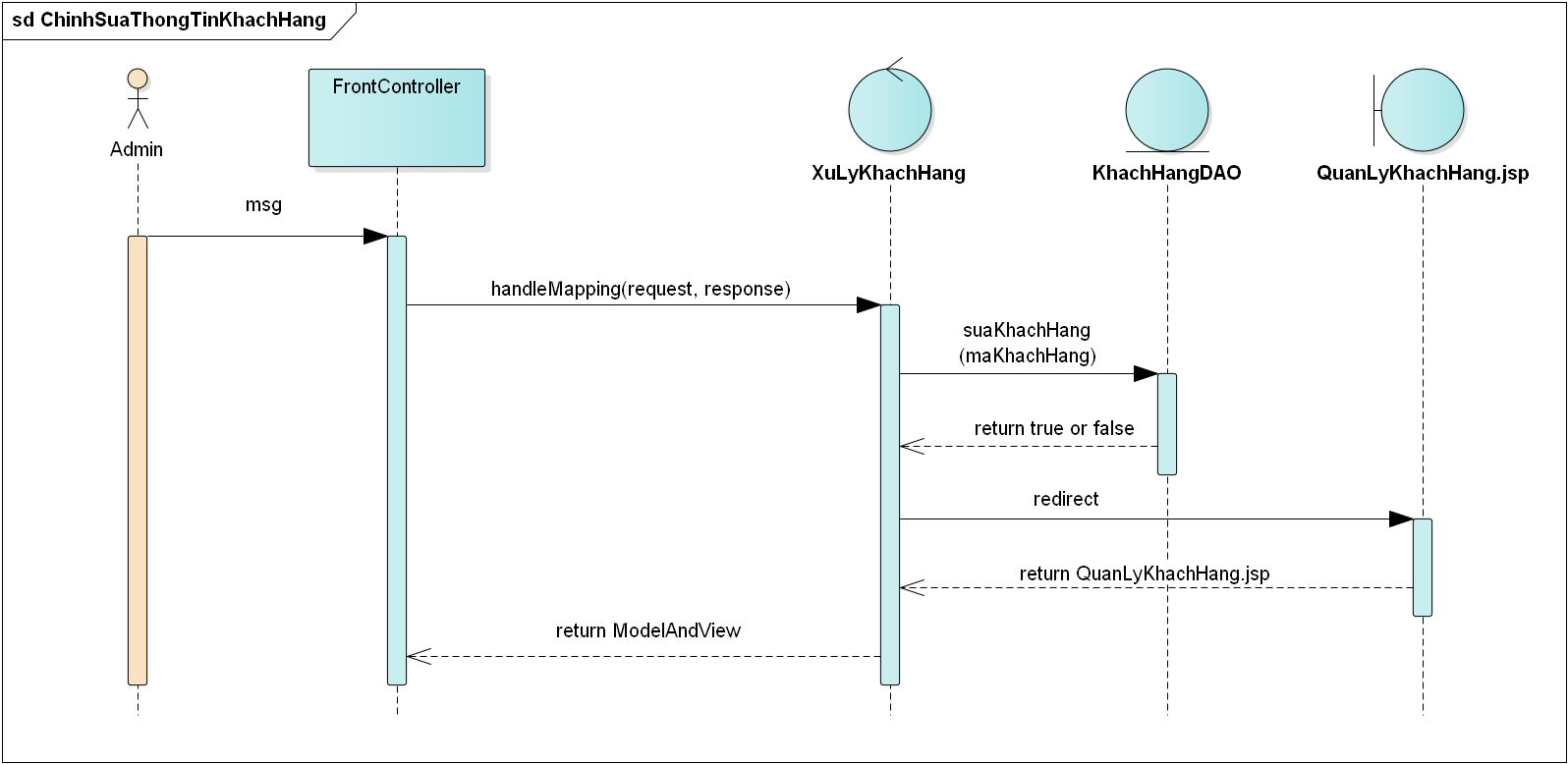


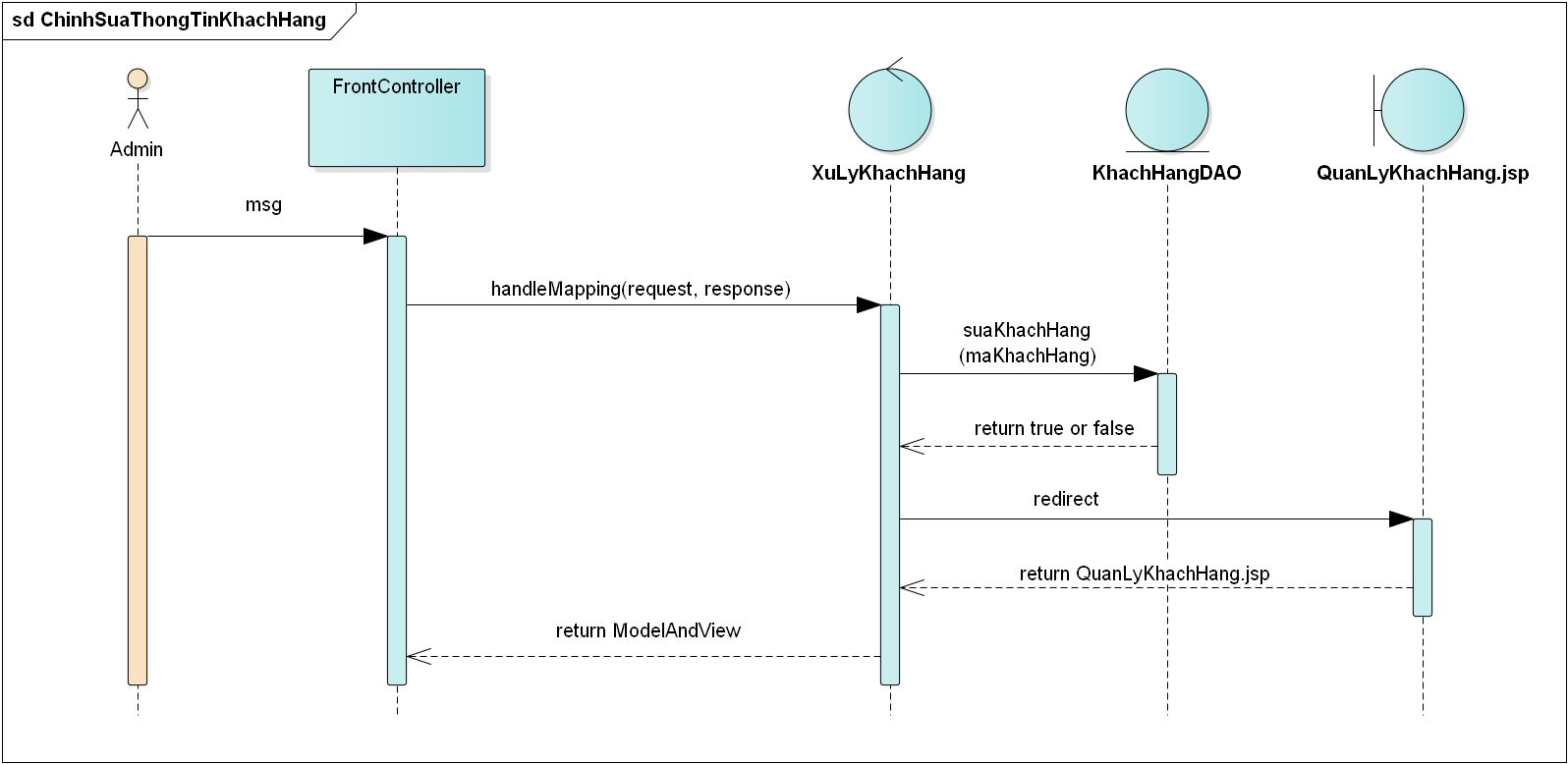


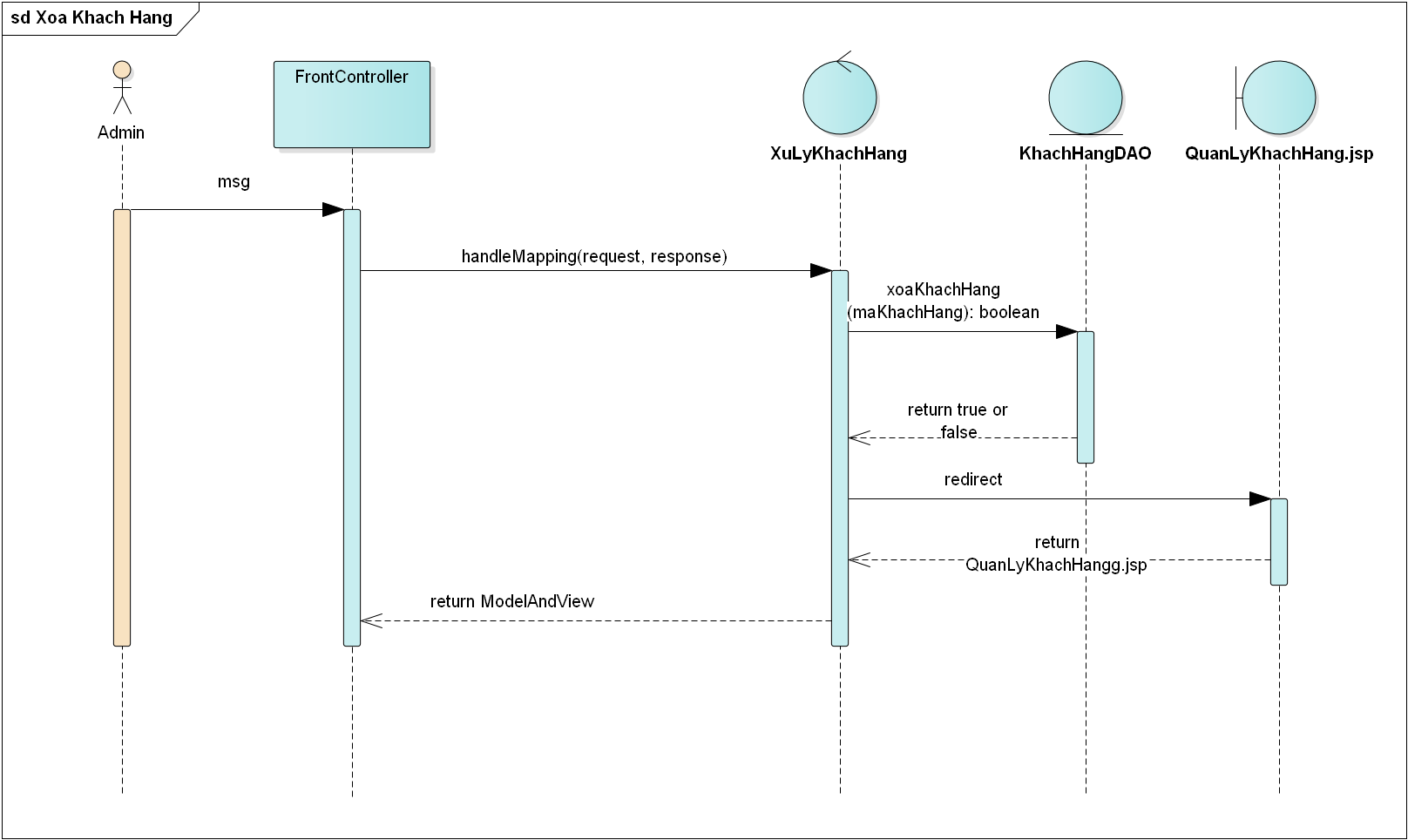
##### Quản Lý Khách hàng

|  |  |
| --- | --- |
| **Use case: Quản Lý Khách hàng** | |
| Mục đích: | **Admin thực hiện chức năng Quản Lý Khách hàng** |
| Mô tả: | Admin vào trang “quản lý **Khách hàng**” để thực hiện thao tác thêm, xóa, sửa, cập nhật Báo |
| Tác nhân: | Admin |
| Điều kiện sau: | Cập nhật lại thông tin vào CSDL |
| Luồng sự kiện chính (Basic flows) | 1. Hệ thống hiển thị trang quản lý khách hàng 2. Admin chọn chức năng ( Tìm kiếm, Sửa, Xóa ) 3. Hệ thống thực hiện chức năng theo yêu cầu. 4. Hệ thống cập nhật dữ liệu vào CSDL. 5. Hệ thống quay lại trang quản lý khách hàng. |





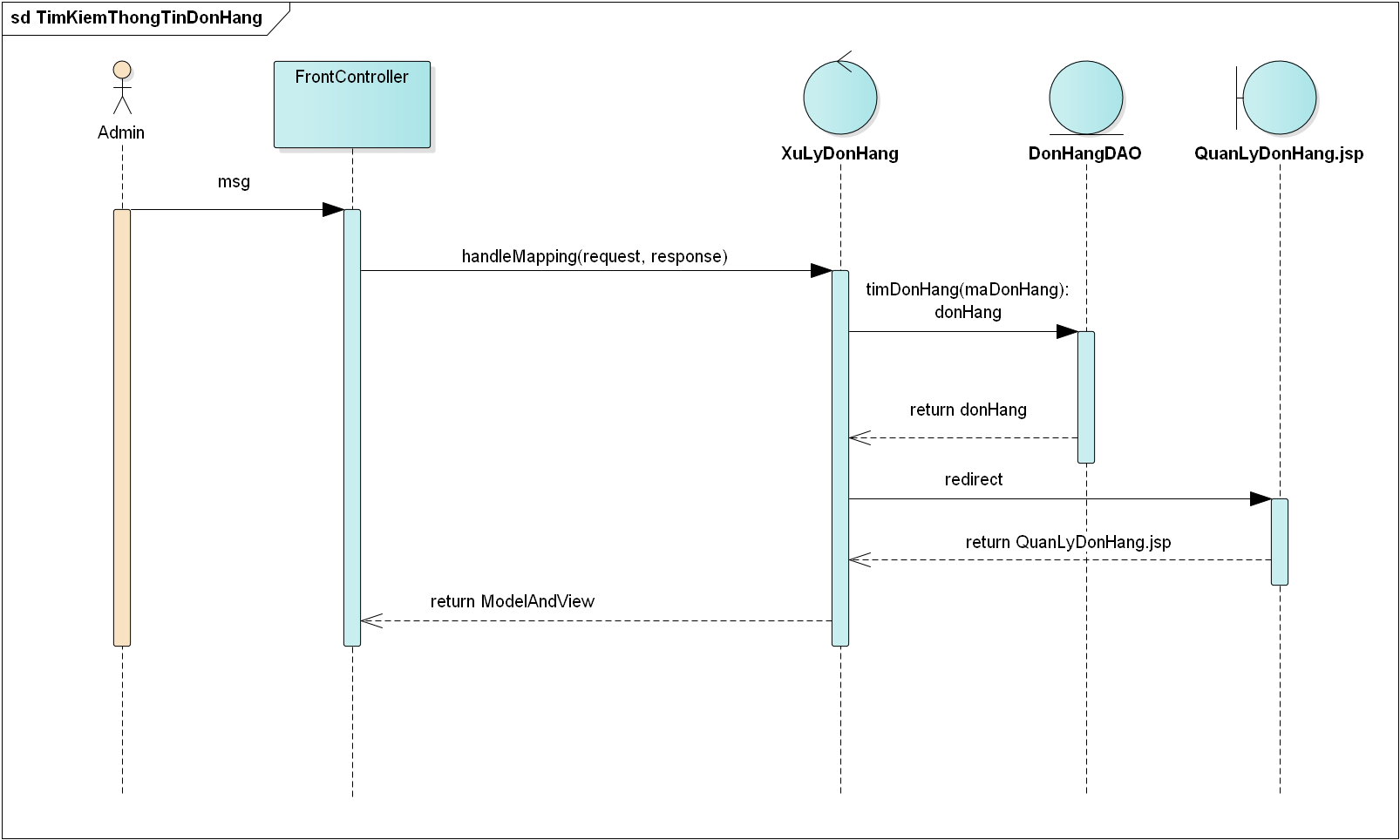


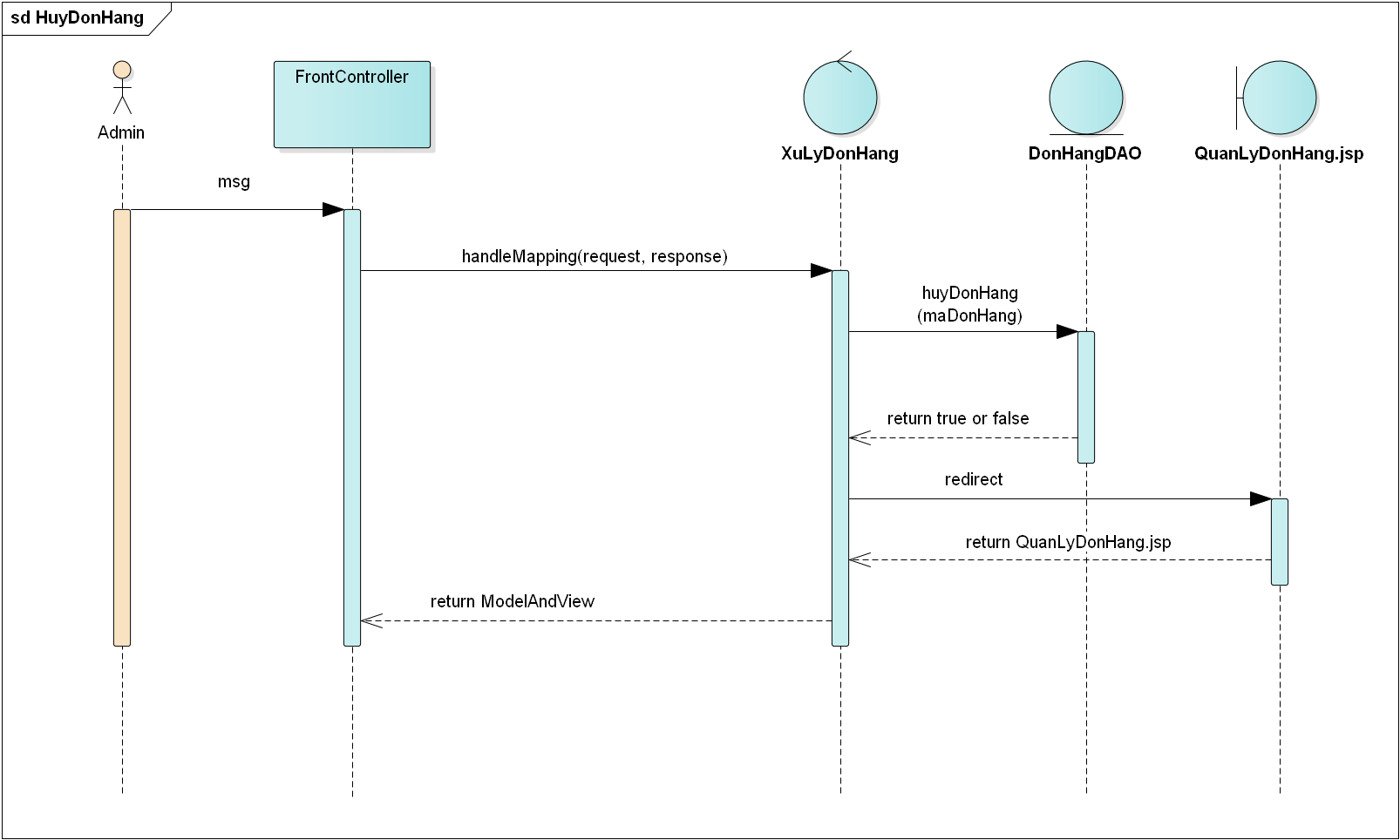


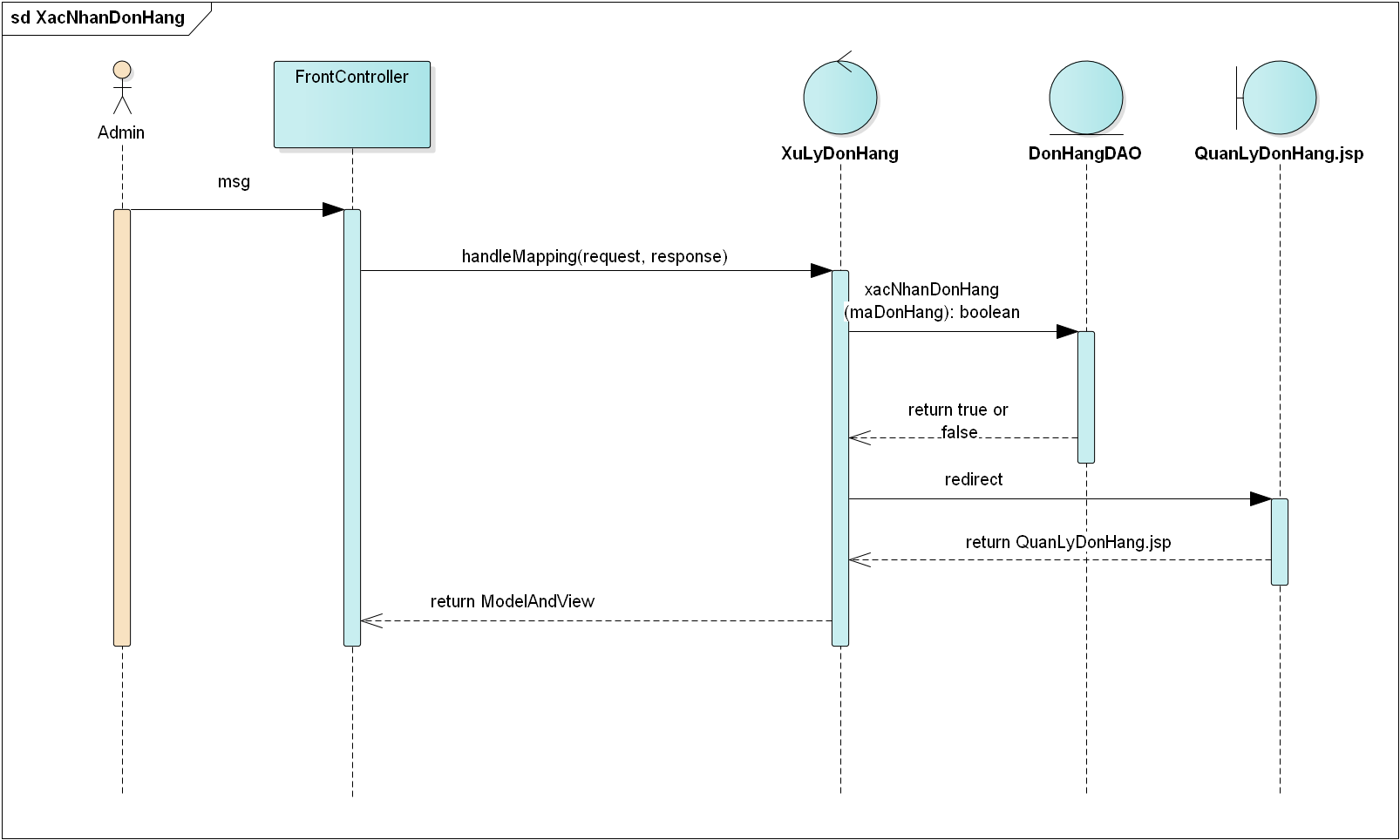
##### Quản Lý Đơn Hàng

|  |  |
| --- | --- |
| **Use case: Quản Lý Đơn Hàng** | |
| Mục đích: | **Admin thực hiện chức năng Quản Lý Đơn hàng** |
| Mô tả: | Admin vào trang “quản lý **Đơn hàng**” để thực hiện thao tác Tìm kiếm, Hủy, Xác nhận đơn hàng |
| Tác nhân: | Admin |
| Điều kiện sau: | Cập nhật lại thông tin vào CSDL |
| Luồng sự kiện chính (Basic flows) | 1. Hệ thống hiển thị trang quản lý đơn hàng 2. Admin chọn chức năng (Tìm kiếm, Hủy, Xác Nhận ) 3. Hệ thống thực hiện chức năng theo yêu cầu. 4. Hệ thống cập nhật dữ liệu vào CSDL. 5. Hệ thống quay lại trang quản đơn hàng. |









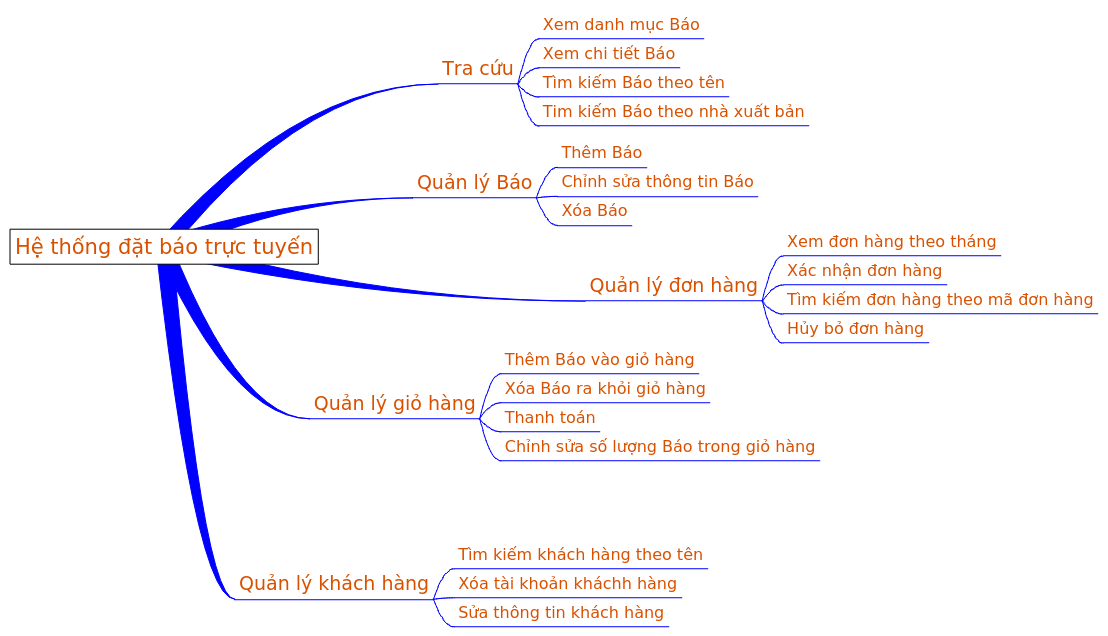
3.2.8

# : THIẾT KẾ VÀ HIỆN THỰC

## Sơ đồ Class



## Sơ đồ Sitemap



## Thiết kết giao diện phác thảo

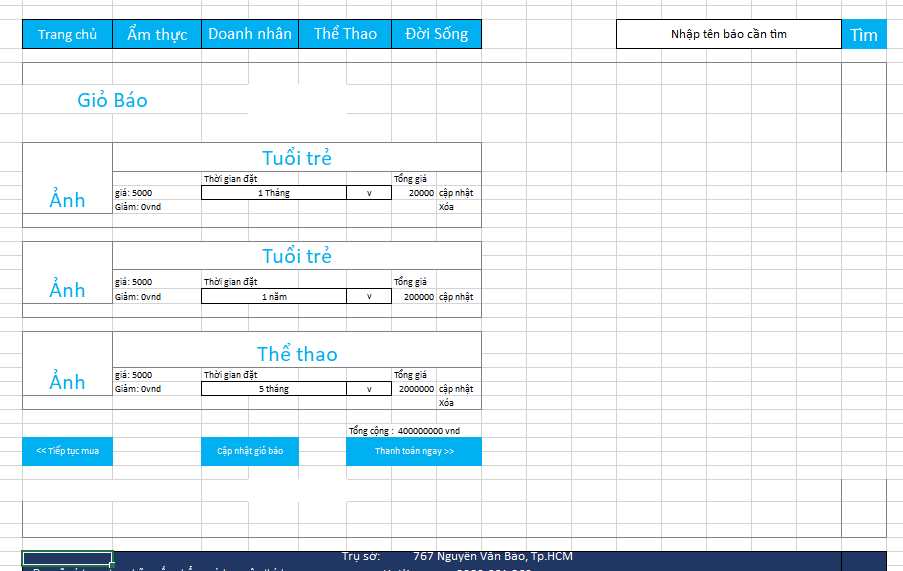
### Giao diện danh mục



### Giao diện chi tiết báo



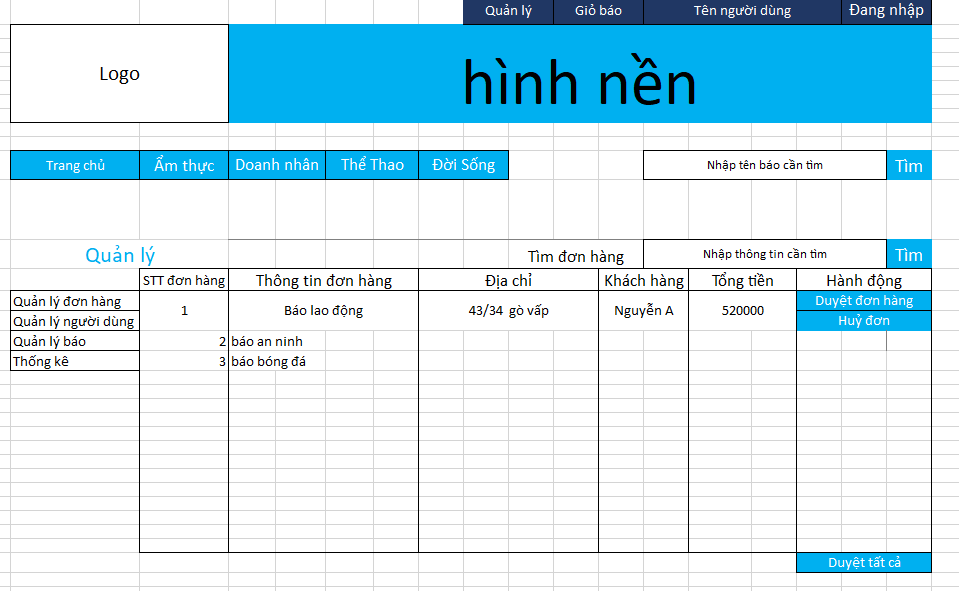
### Giao diện giỏ báo



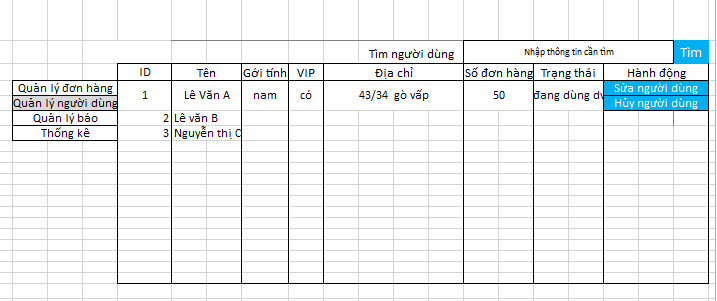
### Giao diện thanh toán



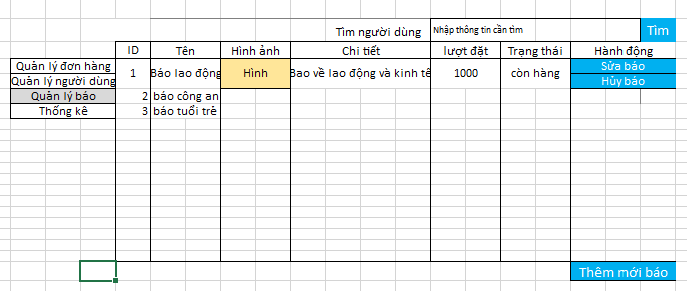
### Giao diện quản lý đợn hàng



### Giao diện quản lý người dùng



### Giao diện quản lý báo



# : KẾT LUẬN

## Kết quả đạt được

Thực đã hiện được chức năng chính của dự án gồm chức năng đăng nhập, thêm giỏ báo,đặt báo và phê duyệt đơn báo.

## Hạn chế của đồ án

Đồ án vẫn còn thiếu sót về các chức năng quản lý, các chức năng phụ trợ.

## Hướng phát triển

Nhóm sẽ phát triển cho dự án được hoàn thiện hơn và bổ sung thanh toán bằng online trong tương lai

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Việt

1. Họ và Tên tác giả (Thứ tự theo Tên). Tên sách. Tên nhà xuất bản, năm xuất bản.
2. Nguyễn Quốc Cường, Hoàng Đức Hải. Giáo trình Đồ hoạ vi tính. NXB Giáo dục, 2000.
3. Cơ sở lý thuyết JAVA: http://voer.edu.vn/m/gioi-thieu-ve-ngon-ngu-java/6a520965
4. Cơ sở lý thuyết JDBC: https://o7planning.org/vi/10167/huong-dan-su-dung-java-jdbc
5. Cơ sở lý thuyết Spring MVC: https://stackjava.com/spring/spring-mvc-phan-1-kien-truc-cua-spring-mvc.html

Các tài liệu Tiếng Anh

1. Amy Apon. Lecture for Cluster and Grid Computing. University of Arkansas, 2004.
2. Bart Jacob, Luis Ferreira, Nobert Bieberstein, Candice Gilzean, Jean-Yves, Girard, Roman Strachowski, Seong (Steve) Yu. Enabling Application for Grid Computing with Globus. IBM RedBooks, 2003.

Các tài liệu từ Internet

1. Website GIS của chính phủ: [http://gis.chinhphu.vn](http://gis.chinhphu.vn/)
2. ...

# PHỤ LỤC