

BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



ĐỒ ÁN  
TỐT NGHIỆP ĐẠI HỌC

Đề tài: “Website quản lý tour du lịch”

Người hướng dẫn : ThS. NGUYỄN ANH HÀO  
Sinh viên thực hiện : ĐÀO ĐỨC TÂN  
Mã số sinh viên : N18DCCN184  
Lớp : D18CQCP02-N  
Khóa : 2018-2023  
Hệ : ĐẠI HỌC CHÍNH QUY

TP.HCM, tháng 8/2022



BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



ĐỒ ÁN  
TỐT NGHIỆP ĐẠI HỌC

Đề tài: “Website quản lý tour du lịch”

Người hướng dẫn	:	ThS. NGUYỄN ANH HÀO
Sinh viên thực hiện	:	ĐÀO ĐỨC TÂN
Mã số sinh viên	:	N18DCCN184
Lớp	:	D18CQCP02-N
Khóa	:	2018-2023
Hệ	:	ĐẠI HỌC CHÍNH QUY

TP.HCM, tháng 8/2022

## LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời tri ân sâu sắc đến các thầy cô trường Học Viện Công Nghệ Bưu Chính Viễn thông cơ sở tại TP.HCM đã tận tình dẫn dắt và truyền đạt cho em rất nhiều kiến thức quý báu trong các học kỳ vừa qua.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến ThS.Nguyễn Anh Hào. Thầy đã hướng dẫn tận tình, truyền đạt kiến thức, chỉ bảo cho em trong suốt thời gian học tập và thực hiện đề tài. Kính chúc thầy và gia đình nhiều sức khoẻ và thành công trong cuộc sống.

Và xin chân thành cảm ơn tất cả các bạn đã luôn sát cánh, giúp đỡ, động viên mình trong những thời điểm khó khăn, tiếp thêm động lực và ý chí giúp vượt qua khó khăn trong suốt quá trình học tập tại trường.

Tuy nhiên, vì thời gian và kiến thức chuyên môn còn hạn chế cùng nhiều yếu tố khách quan khác nên trong quá trình thực hiện đề tài không tránh khỏi những sai sót. Kính mong được lời nhận xét và góp ý của thầy cô, để em có thể hoàn thiện bản thân mình hơn. Một lần nữa, em xin chân thành cảm ơn!

TP.HCM, tháng 8 năm 2022

Sinh viên thực hiện

Đào Đức Tân

## MỤC LỤC

<b>CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Tổng quan.....	1
1.2 Tìm hiểu “Website quản lý tour du lịch”.....	1
1.2.1. Lý thuyết.....	1
1.2.2. Thực hành.....	1
1.3 Công cụ sử dụng.....	2
1.4 Mục tiêu đề tài. ....	2
<b>CHƯƠNG 2: CƠ SỞ LÝ THUYẾT ĐỀ TÀI.....</b>	<b>3</b>
2.1 Xây dựng hệ thống .....	3
2.1.1 Giới thiệu Spring.....	3
2.1.1.1 Giới thiệu Spring Framework. ....	3
2.1.1.2 Mô hình MVC. ....	3
2.1.1.3 Lợi ích của Spring MVC.....	5
2.1.1.4 Mô hình hoạt động Spring MVC.....	6
2.1.2 Tổng quan về RESTful API.....	7
2.1.3 Tổng quan về Reactjs. ....	7
2.1.4 Giới thiệu về hệ quản trị cơ sở dữ liệu MySQL, công cụ PhpMyAdmin và XAMPP.....	8
2.1.4.1 Giới thiệu về hệ quản trị cơ sở dữ liệu MySQL: .....	8
2.1.4.2 Giới thiệu công cụ PhpMyAdmin: .....	9
2.1.4.3 Giới thiệu XAMPP: .....	9
<b>CHƯƠNG 3: KHẢO SÁT HỆ THỐNG.....</b>	<b>10</b>
3.1 Khảo sát hệ thống website quản lý tour du lịch.....	10
3.2 Các chức năng của hệ thống. ....	10
3.2.1 Yêu cầu chức năng của hệ thống.....	10
3.2.2 Yêu cầu phi chức năng.....	11
<b>CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....</b>	<b>12</b>
4.1 Mô hình chức năng nghiệp vụ (BFD – Business Function Diagram).....	12
4.2 Thiết lập mô hình thực thể ERD (ERD – Entity Relationship Diagram).....	13
4.2.1 Xác định các thực thể. ....	13
4.2.2 Mô hình thực thể kết hợp ERD.....	13
4.2.3 Mô hình dữ liệu quan hệ.....	14
4.3 Thiết kế dữ liệu.....	15
4.3.1 Mô hình Diagram.....	15
4.3.2 Từ điển dữ liệu. ....	15
4.4 Ràng buộc toàn vẹn và tầm ảnh hưởng. ....	21
4.4.1 Ràng buộc toàn vẹn miền giá trị.....	21

4.4.2 Ràng buộc toàn vẹn liên thuộc tính.....	21
4.5 Sơ đồ Use case và sơ đồ tuần tự các chức năng.....	22
4.5.1 Sơ đồ Use case .....	22
4.5.2 Sơ đồ tuần tự .....	24
<b>CHƯƠNG 5: DEMO CHƯƠNG TRÌNH.....</b>	<b>40</b>
<b>CHƯƠNG 6: KẾT LUẬN.....</b>	<b>62</b>
6.1 Kết luận .....	62
6.1.1 Kết quả đạt được.....	62
6.1.2 Hạn chế.....	62
6.2 Hướng phát triển.....	62
<b>CHƯƠNG 7: TÀI LIỆU THAM KHẢO .....</b>	<b>63</b>

## CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

### 1.1 Tổng quan.

Hiện tại, Online là một thị trường lớn và đầy tiềm năng đối với kinh doanh. Trong khi đó, người tiêu dùng đang có xu hướng “online hóa” việc mua sắm hàng hoá và dịch vụ để tiết kiệm thời gian, công sức, do đó, các doanh nghiệp cũng đang dần tiến sâu vào thị trường kinh doanh trực tuyến này với mong muốn tăng hiệu suất và doanh thu bán hàng.

Được biết, khi bán hàng theo phương pháp truyền thống rất bất cập trong vấn đề quảng cáo sản phẩm cũng như thống kê doanh thu. Việc thống kê báo cáo doanh thu hoàn toàn dựa trên phần mềm excel, việc này dẫn đến một số vấn đề khá khó khăn như thời gian xuất một báo cáo rất lâu, tốn nhiều thời gian và nhân lực, nhập liệu nếu có nhầm lẫn sẽ không được cảnh báo và xử lý, chẳng hạn như nếu xuất một sản phẩm mà trong kho số lượng tồn của sản phẩm đó không đủ để xuất hay sản phẩm đã hết hàng nhưng khách hàng không được thông báo về tình trạng này, nhân viên cũng không được nhắc để nhập thêm loại hàng ấy. Ngoài ra, khách hàng không chủ động được trong việc đặt trước và mua hàng của shop, mọi việc bán và mua phải diễn ra đồng thời giữa cả bên mua và bên bán.

Do đó, vấn đề xây dựng một shop bán hàng online rất cần thiết. Việc này giúp shop bán hàng quản lý hàng hóa tốt hơn, thống kê báo cáo doanh thu nhanh chóng, chính xác, giảm nhân lực, giảm chi phí và thời gian. Việc quảng cáo sản phẩm và thương hiệu cũng dễ dàng hơn. Về mặt khách hàng, một shop online sẽ giúp họ tiết kiệm thời gian mua sắm cũng như giúp “đặt chỗ” trước những sản phẩm mà họ yêu thích, không sợ bị mua trước hay hết hàng khi họ không đến shop, chủ động trong việc xem và đặt hàng.

Vì các vấn đề đã nêu, em chọn đề tài “**Website quản lý tour du lịch**” nhằm giúp các công ty du lịch cải thiện được vấn đề quản lý, quảng cáo dịch vụ của mình. Đồng thời phù hợp với xu hướng phát triển của thế giới hiện nay.

Chương trình này xây dựng trên nền tảng Spring boot. Với kiến thức còn hạn chế, chắc chắn không thể tránh khỏi những sai sót. Vì vậy, em rất mong nhận được mọi sự góp ý của quý thầy cô để em có thể hoàn thiện đề tài tốt hơn.

### 1.2 Tìm hiểu “Website quản lý tour du lịch”.

#### 1.2.1. Lý thuyết.

Tìm hiểu về:

- Spring boot.
- RESTful API.
- Reactjs.

#### 1.2.2. Thực hành.

Phân tích và thiết kế cơ sở lưu giữ dữ liệu.

Xây dựng website quản lý tour du lịch trực tuyến với các chức năng:

- Thêm/xóa/sửa các danh mục cho admin.
- Phân công hướng dẫn viên.
- Đăng ký, đăng nhập, tạo tài khoản.
- Tra cứu theo tiêu chí.
- Xem danh sách tour, chi tiết tour.
- Đặt tour, đặt câu hỏi, đánh giá tour.
- In hóa đơn, xuất danh sách khách hàng.

- Báo cáo doanh thu.

### 1.3 Công cụ sử dụng.

- PhpMyAdmin.
- IntelliJi 2020.1.
- Visual Studio Code.

### 1.4 Mục tiêu đề tài.

Đảm bảo việc quản lý, đặt tour được nhanh chóng, chính xác:

- Giúp nhân viên quản lý tình hình hoạt động của công ty, quản lý các danh mục, phân công hướng dẫn viên, in hoá đơn, thống kê.
- Đảm bảo khách hàng có thông tin chính xác về tour du lịch, đặt tour diễn ra đơn giản.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT ĐỀ TÀI

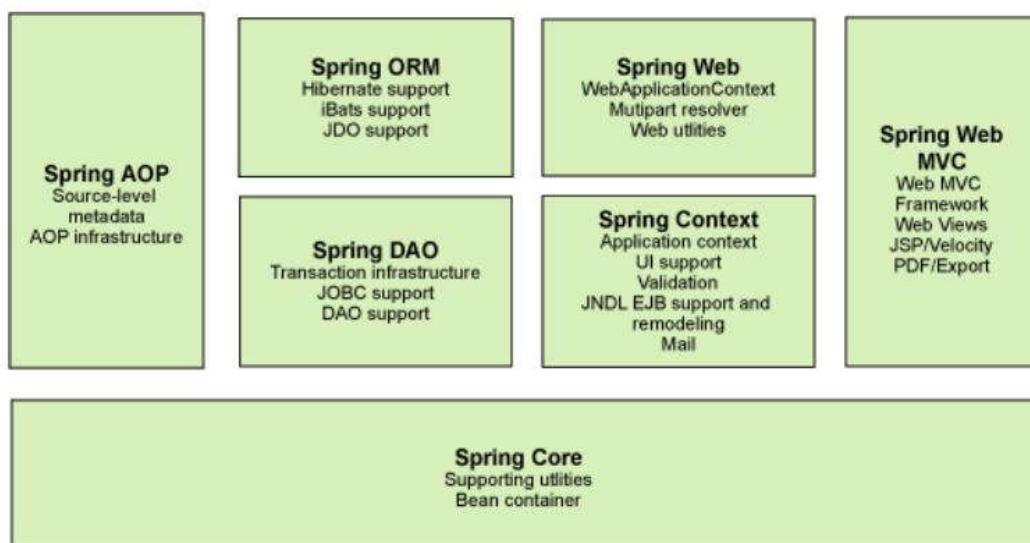
### 2.1 Xây dựng hệ thống.

#### 2.1.1 Giới thiệu Spring.

##### 2.1.1.1 Giới thiệu Spring Framework.

Spring Framework là một framework mã nguồn mở được viết bằng Java. Nó được xem như là một giải pháp kiến trúc tốt nhất của Java EE hiện nay. Theo thiết kế, framework này giảm nhẹ công việc kỹ thuật cho lập trình viên Java, để họ tập trung sâu vào các công việc nghiệp vụ của ứng dụng. Đồng thời cung cấp một giải pháp toàn diện để họ thực hiện ứng dụng một cách tiện lợi nhất, chặt chẽ nhất đồng thời dễ dàng bảo trì, bảo dưỡng sau này.

Các module chính: Spring được tổ chức thành 7 modules:



*Hình 1: Các module của Spring*

##### 2.1.1.2 Mô hình MVC.

#### Spring Core

Core package là phần lõi của framework, cung cấp những đặc tính IoC (Inversion of Control) và DI (Dependency Injection). BeanFactory đảm nhận việc sản sinh và móc nối sự phụ thuộc giữa các đối tượng trong file cấu hình.

### Spring Context/Application Context

Phía trên của Core package là Context package - cung cấp cách để truy cập đối tượng. Context package kết thừa các đặc tính từ bean package và thêm vào chức năng đa ngôn ngữ (I18N), truyền sự kiện, resource-loading,...

### Spring AOP (Aspect Oriented Programming).

Spring AOP module tích hợp chức năng lập trình hướng khía cạnh vào Spring framework thông qua cấu hình của nó. Spring AOP module cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong bất kỳ ứng dụng nào sử dụng Spring. Với Spring AOP chúng ta có thể tích hợp declarative transaction management vào trong ứng dụng mà không cần dựa vào EJB component.

Spring AOP module cũng đưa lập trình metadata vào trong Spring. Sử dụng cái này chúng ta có thể thêm annotation (chú thích) vào source code để hướng dẫn Spring và làm thế nào để thực hiện các phương thức sự kiện đã được cài đặt sẵn.

### Spring DAO (Data Access Object)

DAO package cung cấp cho tầng JDBC, bỏ bớt những coding dài dòng của JDBC và chuyển đổi mã lỗi được xác định bởi database vendor. JDBC package cung cấp cách lập trình tốt như declarative transaction management. Tầng JDBC và DAO đưa ra một cây phân cấp exception để quản lý kết nối đến database, điều khiển exception và thông báo lỗi được ném bởi vendor của database. Tầng exception đơn giản điều khiển lỗi và giảm khối lượng code mà chúng ta cần viết như mở và đóng kết nối. Module này cũng cung cấp các dịch vụ quản lý giao dịch cho các đối tượng trong ứng dụng Spring.

### Spring ORM (Object Relational Mapping)

ORM package cung cấp tầng tích hợp với object-relational mapping API bao gồm: JDO, Hibernate, iBatis. Sử dụng ORM package bạn có thể sử

dụng tất cả các object-relational mapping đó kết hợp với tất cả các đặc tính của Spring như declarative transaction management.

### **Spring Web module.**

Spring Web package cung cấp đặc tính của web như: chức năng file-upload, khởi tạo IoC container sử dụng trình lắng nghe serlvet và web-oriented application context. Nằm trên application context module, cung cấp context cho các ứng dụng web. Spring cũng hỗ trợ tích hợp với Struts, JSF và Webwork. Web module cũng làm giảm bớt các công việc điều khiển nhiều request và gắn các tham số của request vào các đối tượng domain.

### **Spring MVC Framework.**

Spring Framework là một ứng dụng mã nguồn mở phổ biến để phát triển ứng dụng Java EE dễ dàng hơn. Nó là một container gồm Web Framework (tiếp nhận và xử lý yêu cầu, chia sẻ dữ liệu...) và Web View (quản lý giao diện).

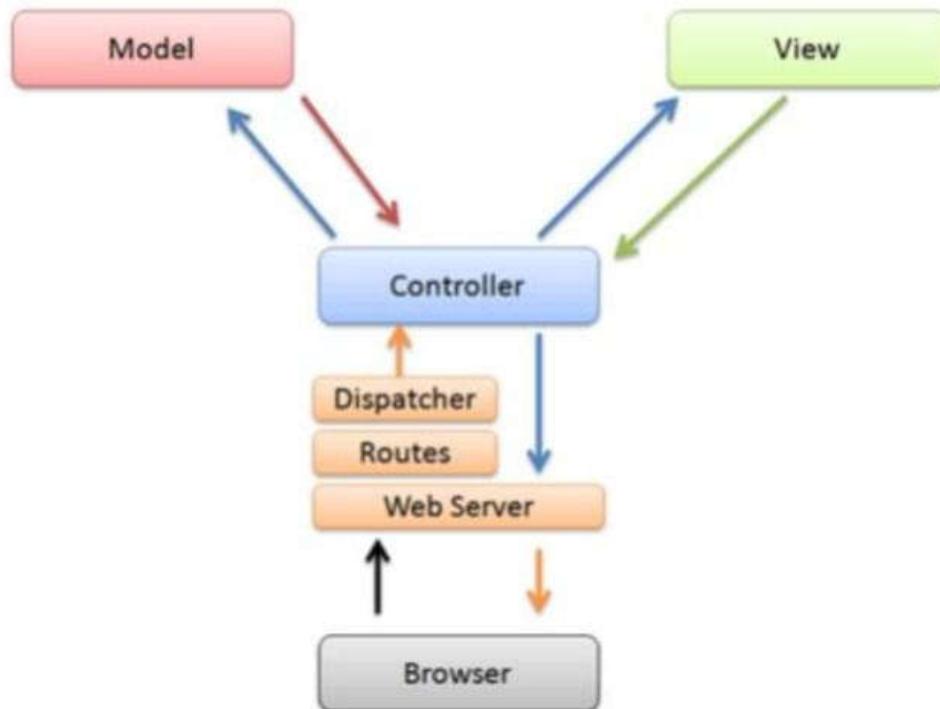
MVC Framework được cài đặt đầy đủ các đặc tính của MVC pattern để xây dựng các ứng dụng Web. Các thành phần gồm View (JSP, Velocity, Tiles và generation of PDF và Excel file...), Model (domain model) và Controller (chứa các xử lý yêu cầu).

#### **2.1.1.3 Lợi ích của Spring MVC.**

- Tất cả các framework đã được tích hợp rất tốt vào Spring.
- Hoạt động rất tốt khi áp dụng theo kiến trúc MVC.
- Sử dụng cơ chế plug-in.
- Kết hợp rất tốt với các O/R (object-relational) Mapping frameworks như là Hibernate.
- Dễ Testing ứng dụng.
- Ít phức tạp hơn so với các framework khác.

- Cộng đồng người sử dụng rất nhiều, nhiều sách mới được xuất bản.

#### 2.1.1.4 Mô hình hoạt động Spring MVC.



Hình 2: Mô hình MVC

#### Model

Model gồm các lớp java có nhiệm vụ:

- Biểu diễn data và cho phép đọc/ghi data thông qua các phương thức getter/setter theo qui ước trong JavaBean.
- Buộc dữ liệu form giao diện tức là nhận dữ liệu từ tham số và cung cấp dữ liệu để trình bày lên giao diện.
- Thi hành các yêu cầu (tính toán, kết nối CSDL ...)
- Trả về các giá trị tính toán theo yêu cầu của Controller

#### View

Bao gồm các mã tương tự như JSP, HTML, CSS, XML, Javascript, JSON... để hiển thị giao diện người dùng, các dữ liệu trả về từ Model thông qua Controller...

### **Controller**

Đồng bộ hoá giữa View và Model. Tức là với một trang JSP này thì sẽ tương ứng với lớp java nào để xử lý nó và ngược lại, kết quả sẽ trả về trang jsp nào. Nó đóng vai trò điều tiết giữa View và Model.

Như vậy, chúng ta có thể tách biệt được các mã java ra khỏi mã html. Do vậy, nó đã giải quyết được các khó khăn sự phụ thuộc nghiệp vụ lẫn nhau giữa các thành viên tham gia dự án. Người thiết kế giao diện và người lập trình java có thể mang tính chất độc lập tương đối.

Việc kiểm lỗi hay bảo trì sẽ dễ dàng hơn, việc thay đổi các theme của trang web cũng dễ dàng hơn.

#### **2.1.2 Tổng quan về RESTful API.**

REST là viết tắt của Representational State Transfer. RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau. Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản lý các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

#### **2.1.3 Tổng quan về Reactjs.**

Reactjs là một thư viện Javascript mã nguồn mở hỗ trợ xây dựng các thành phần giao diện nhanh gọn và tiện lợi. Bình thường các lập trình viên sẽ nhúng javascript vào code HTML thông qua các attribute như AngularJS nhưng với Reactjs làm việc như một thư viện cho phép nhúng HTML vào javascript thông qua JSX. Qua đó bạn có thể dễ dàng lồng các đoạn HTML vào trong JSX làm cho các component dễ hiểu và dễ sử dụng hơn.

Làm gì cũng phải có nguyên nhân, người dùng sử dụng Reactjs ngày càng nhiều thì Reactjs phải mang lại những lợi ích tốt.

- ReactJS giúp cho việc viết các đoạn code Javascript sẽ trở nên dễ dàng hơn vì nó sử dụng một cú pháp đặc biệt đó chính là cú pháp JSX. Thông qua JSX cho phép nhúng code HTML và Javascript.

- ReactJS cho phép Developer phá vỡ những cấu tạo UI phức tạp thành những component độc lập. Dev sẽ không phải lo lắng về tổng thể ứng dụng web, giờ đây Developer dễ dàng chia nhỏ các cấu trúc UI/UX phức tạp thành từng component đơn giản hơn.
- Đi kèm với ReactJS là rất nhiều các công cụ phát triển giúp cho việc debug code một cách dễ dàng hơn.
- Một trong những ưu điểm nữa của ReactJS đó là sự thân thiện với SEO. Hầu như các JS Frameworks không thân thiện với các tìm kiếm mặc dù đã được cải thiện nhiều nhưng dưới sự hỗ trợ của các render dữ liệu trả về dưới dạng web page giúp cho SEO chuẩn hơn.

#### 2.1.4 Giới thiệu về hệ quản trị cơ sở dữ liệu MySQL, công cụ PhpMyAdmin và XAMPP

##### 2.1.4.1 Giới thiệu về hệ quản trị cơ sở dữ liệu MySQL:

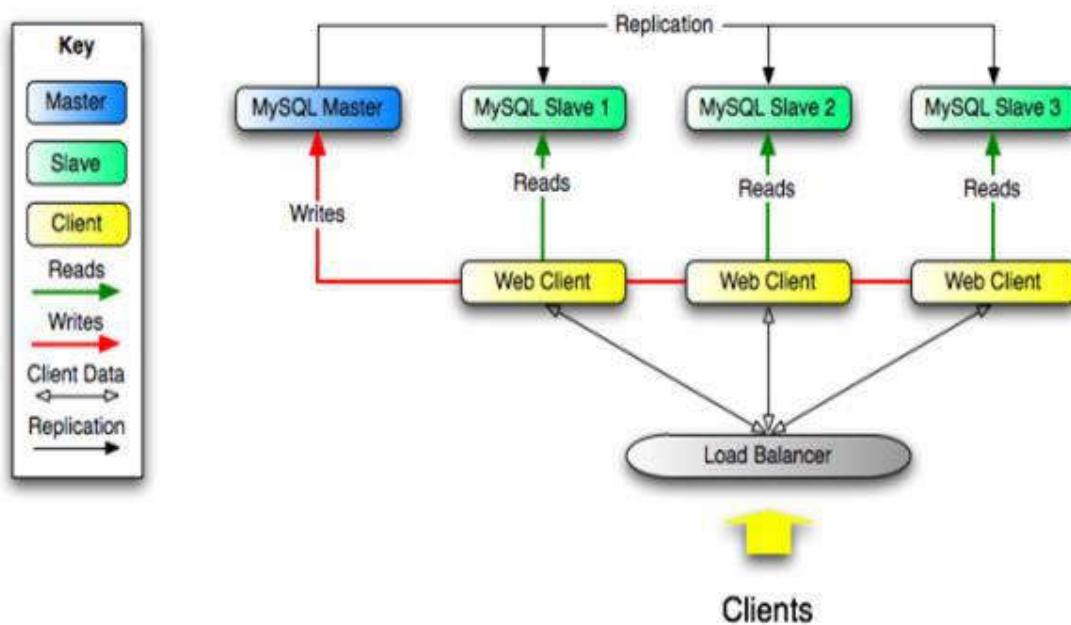
- **Giới thiệu:**

**MySQL** là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server. Với RDBMS là viết tắt của **Relational Database Management System**. MySQL được tích hợp apache, PHP. MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu. Mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. MySQL cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. MySQL được phát hành từ thập niên 90s.



Hình 4 Hệ quản trị cơ sở dữ liệu MySQL

- Cách thức hoạt động của MySQL:



Cách vận hành chính trong môi trường MySQL như sau:

- MySQL tạo ra bảng để lưu trữ dữ liệu, định nghĩa sự liên quan giữa các bảng đó.
- Client sẽ gửi yêu cầu SQL bằng một lệnh đặc biệt trên MySQL.
- Ứng dụng trên server sẽ phản hồi thông tin và trả về kết quả trên máy client.

#### 2.1.4.2 Giới thiệu công cụ PhpMyAdmin:

PhpMyAdmin là phần mềm mã nguồn mở được viết bằng ngôn ngữ PHP giúp quản trị cơ sở dữ liệu MySQL thông qua giao diện web. Sử dụng phpMyadmin người dùng có thể thực hiện được nhiều tác vụ khác nhau như khi sử dụng cửa sổ dòng lệnh. Các tác vụ này bao gồm việc tạo, cập nhật và xoá các cơ sở dữ liệu, các bảng, các trường, dữ liệu trên bảng, phân quyền và quản lý người dùng, ...

#### 2.1.4.3 Giới thiệu XAMPP:

Phần mềm XAMPP là một loại ứng dụng phần mềm khá phổ biến và thường hay được các lập trình viên sử dụng để xây dựng và phát triển các dự án website theo ngôn ngữ PHP. XAMPP được sử dụng cho mục đích nghiên cứu, phát triển website qua Localhost của máy tính cá nhân. XAMPP được ứng dụng trong nhiều lĩnh vực từ học tập đến nâng cấp, thử nghiệm Website của các lập trình viên.

## CHƯƠNG 3: KHẢO SÁT HỆ THỐNG

### 3.1 Khảo sát hệ thống website quản lý tour du lịch.

Một công ty đặt tour du lịch trực tuyến được tổ chức như sau:

- Công ty có nhiều nhân viên. Thông tin của một nhân viên bao gồm mã nhân viên, họ tên, địa chỉ, số điện thoại, email, mật khẩu, quyền....
- Khách hàng của công ty có thông tin bao gồm: mã khách hàng để phân biệt, họ tên, giới tính, số điện thoại, mật khẩu email, trạng thái ...
- Mỗi khách hàng và nhân viên đều có riêng 1 tài khoản đăng nhập là số điện thoại và mật khẩu. Mỗi nhân viên và khách hàng chỉ thuộc một quyền và một quyền sẽ có nhiều người dùng
- Công ty có nhiều tour. Một tour bao gồm những thông tin như mã tour, tiêu đề tour, hướng dẫn viên, ngày khởi hành, ngày kết thúc, giá tiền, mô tả, loại tour, ...
- Một tour có nhiều địa danh thăm quan, 1 địa danh thăm quan có thể được nhiều tour đến. Mỗi địa danh sẽ có thông tin bao gồm mã địa danh, tên địa danh, hình ảnh.
- Một tour có nhiều phương tiện di chuyển, 1 phương tiện di chuyển có thể được nhiều tour sử dụng. Mỗi phương tiện sẽ có thông tin bao gồm mã phương tiện (biển số xe), tên phương tiện, tên tài xế, loại xe.
- Một tour có nhiều dịch vụ, 1 dịch vụ có thể có trong nhiều tour. Mỗi dịch vụ sẽ có thông tin bao gồm mã dịch vụ, tên dịch vụ.
- Một tour có 1 hướng dẫn viên, 1 hướng dẫn viên có thể hướng dẫn nhiều tour.
- Một tour có thể có nhiều khách đặt, một khách có thể đặt nhiều tour khác nhau.
- Một khách hàng có thể đánh giá tour của mình đã đặt sau khi tour đã kết thúc, một tour có thể được nhiều khách hàng đánh giá.
- Một khách hàng có thể đặt câu hỏi ở nhiều tour, một tour có thể được nhiều khách hàng đặt câu hỏi.

### 3.2 Các chức năng của hệ thống.

Để tài xâ dựng website quản lý tour du lịch có các chức năng của hệ thống như sau:

#### 3.2.1 Yêu cầu chức năng của hệ thống.

Công việc của người quản lý:

- Thêm, xóa, sửa các danh mục.
- In hoá đơn.
- Phân công hướng dẫn viên.
- Khoá/mở khoá tài khoản khách hàng.
- Trả lời các câu hỏi của khách hàng.
- Báo cáo doanh thu.

Công việc của hướng dẫn viên:

- Đăng nhập
- Xem thông tin cá nhân.
- Chính sửa thông tin.
- Xem danh sách tour mà mình được phân công.
- Xem danh sách khách hàng của tour.

Công việc của người mua hàng:

- Đăng ký tài khoản/Đăng nhập.
- Xem thông tin cá nhân.
- Chính sửa thông tin cá nhân.
- Xem thông tin tour.
- Lọc tour.
- Đặt tour/ huỷ tour.
- Đánh giá tour/ đặt câu hỏi.

Đối tượng tham gia hệ thống:

- Người quản lý
- Khách hàng.
- Hướng dẫn viên.

### 3.2.2 Yêu cầu phi chức năng.

- Hệ thống đáng tin cậy, chính xác, giao diện thân thiện, truy cập dữ liệu nhanh chóng.
- Đảm bảo tính bảo mật cho người điều hành hệ thống.
- Phải có tính linh hoạt cao

## CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 4.1 Mô hình chức năng nghiệp vụ (BFD – Business Function Diagram).

Chức năng là công việc mà tổ chức cần làm và được phân theo nhiều mức từ tổng hợp đến chi tiết.

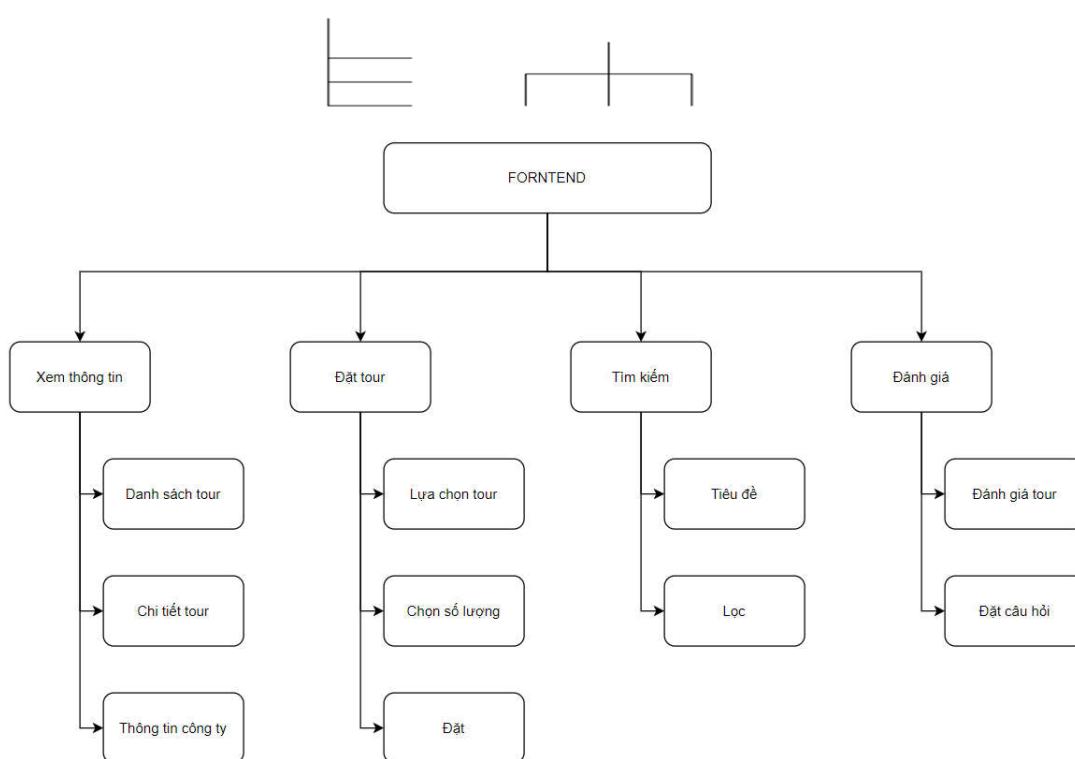
Mỗi chức năng có một tên duy nhất, các chức năng khác nhau phải có tên khác nhau. Để xác định tên cho các chức năng một cách chính xác và cụ thể, có thể bàn luận và nhất trí với người sử dụng.

Hình thức biểu diễn:

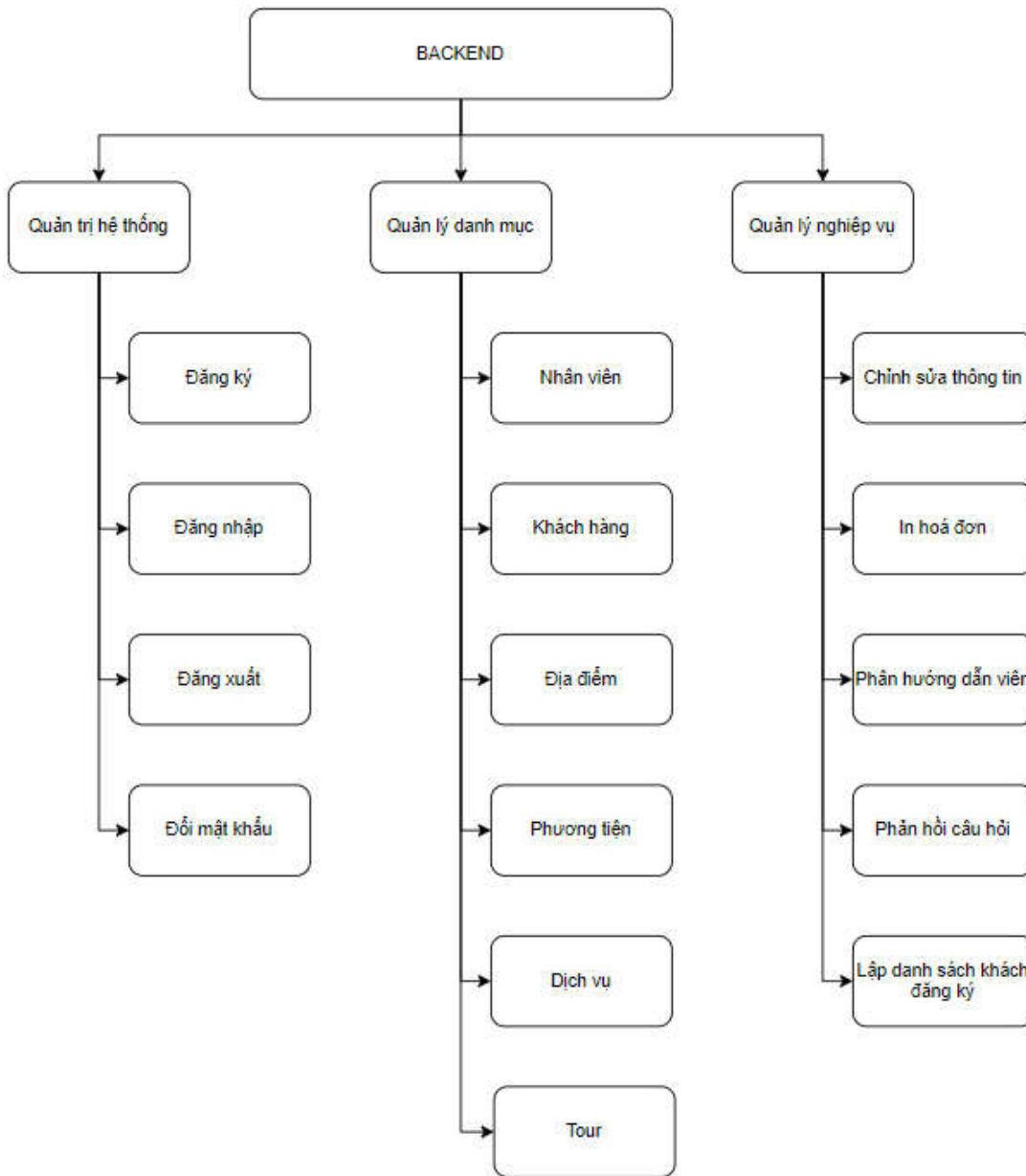
Tên chức năng

*Mỗi chức năng được phân rã thành các chức năng con. Các chức năng con có quan hệ phân cấp với chức năng cha.*

*Biểu diễn mối quan hệ phân cấp chức năng như sau:*



Hình 4.1. Mô hình phân rã chức năng phía frontend được biểu diễn thành hình cây phân cấp



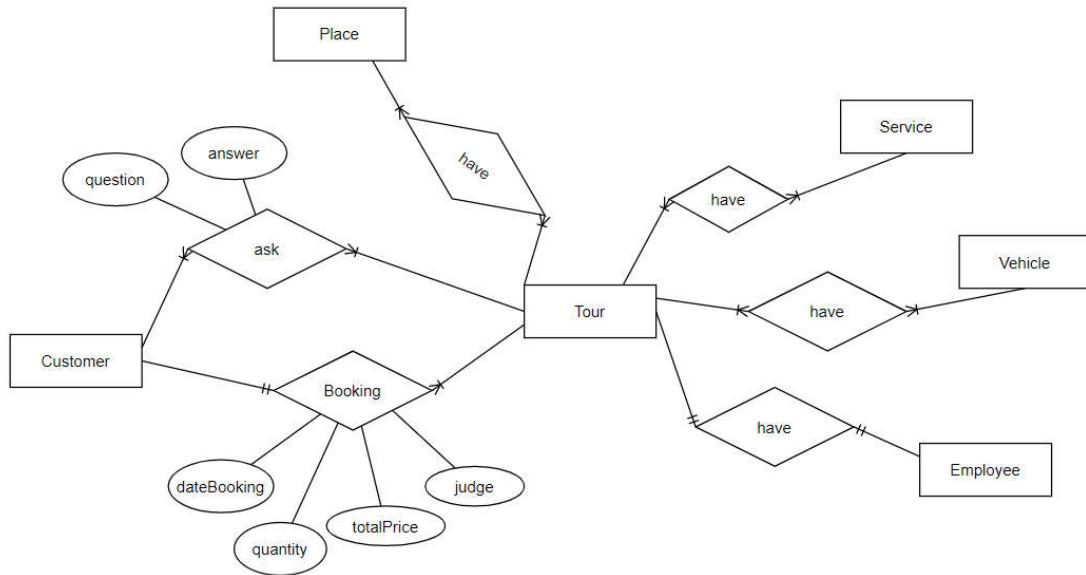
Hình 4.2. Mô hình phân rã chức năng phía backend được biểu diễn thành hình cây phân cấp

## 4.2 Thiết lập mô hình thực thể ERD (ERD – Entity Relationship Diagram).

### 4.2.1 Xác định các thực thể.

- **Tour** ([idTour](#), title, dateStart, dateEnd, price, description, type)
- **Place** ([idPlace](#), name, image)
- **Vehicle** ([idVehicle](#), name, driverName, type)
- **Employee** ([idEmployee](#), name, address, email, phone, password, role)
- **Customer** ([idCustomer](#), name, gender, email, phone, password, status)
- **Service** ([idService](#), name)

### 4.2.2 Mô hình thực thể kết hợp ERD.



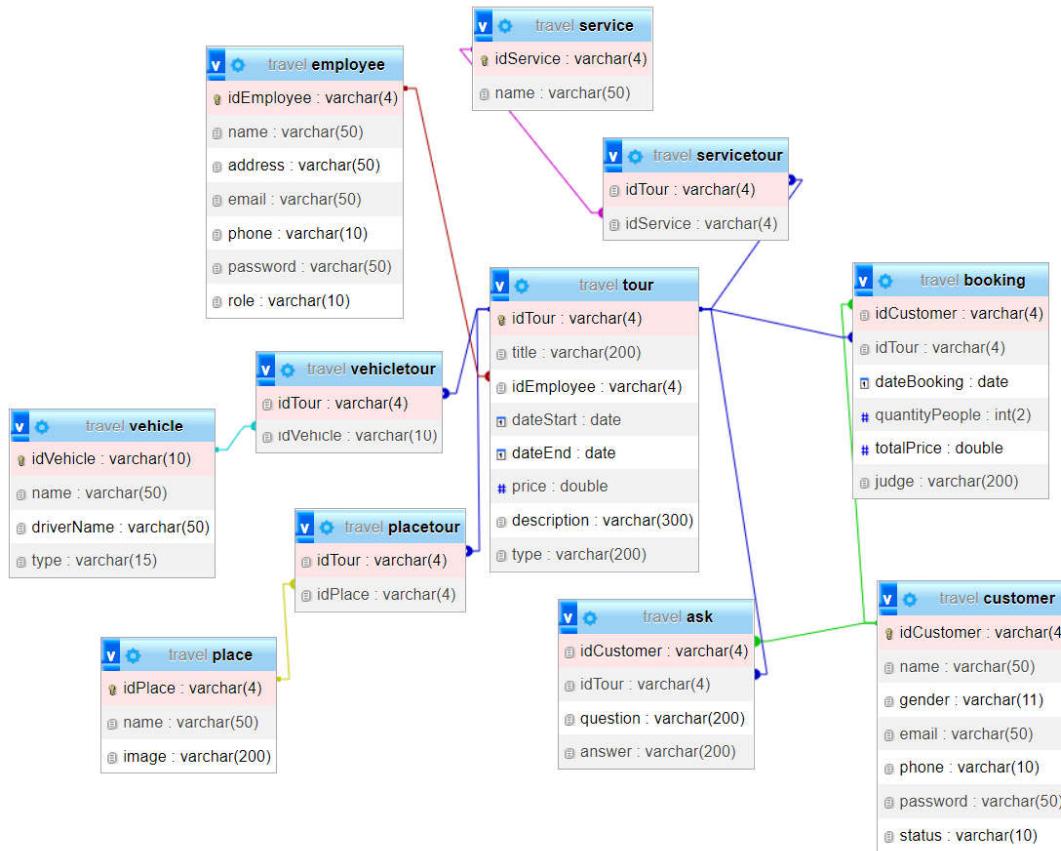
Hình 4.4. ERD.

#### 4.2.3 Mô hình dữ liệu quan hệ.

- **tour** (idTour, title, idEmployee, dateStart, dateEnd, price, description, type)
- **place** (idPlace, name, image)
- **vehicle** (lisencePlate, name, driverName, type)
- **employee** (idEmployee, name, address, email, phone, password, role)
- **customer** (idCustomer, name, gender, email, phone, password, status)
- **service** (idService, name)
- **booking** (idCustomer, idTour, dateBooking, quantity, totalPrice, judge)
- **ask** (idCustomer, idTour, question, answer)
- **placetour** (idTour, idPlace)
- **vehicletour** (idTour, idPlace)
- **servicetour** (idTour, idService)

### 4.3 Thiết kế dữ liệu.

#### 4.3.1 Mô hình Diagram.



Hình 4.5. Diagram

#### 4.3.2 Từ điển dữ liệu.

✓ Bảng: PLACE

PLACE (IDPLACE, NAME, IMAGE)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<b>IDPLACE</b>	varchar(4)	utf8mb4_general_ci		No	None		Change  Drop More	
2	<b>name</b>	varchar(50)	utf8mb4_general_ci		Yes	NULL		Change  Drop More	
3	<b>image</b>	varchar(200)	utf8mb4_general_ci		Yes	NULL		Change  Drop More	

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDPLACE	Varchar(4)	Mã địa danh		X		X
2	NAME	Varchar(50)	Tên địa danh				X

3	IMAGE	Varchar(200)	Đường dẫn đến hình ảnh				X
---	-------	--------------	------------------------------	--	--	--	---

✓ Bảng: VEHICLE

VEHICLE (IDVEHICLE, NAME, DRIVERNAME, TYPE)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
1	<b>idVehicle</b>	varchar(10)	utf8mb4_general_ci	No	None				Change	Drop More
2	<b>name</b>	varchar(50)	utf8mb4_general_ci	Yes	NULL				Change	Drop More
3	<b>driverName</b>	varchar(50)	utf8mb4_general_ci	Yes	NULL				Change	Drop More
4	<b>type</b>	varchar(15)	utf8mb4_general_ci	Yes	NULL				Change	Drop More

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDVEHICLE	Varchar(10)	Mã phương tiện		X		X
2	NAME	Varchar(50)	Tên phương tiện				X
3	DRIVERNAME	Varchar(50)	Tên tài xế				
4	TYPE	Varchar(15)	Loại xe				

✓ Bảng: SERVICE

SERVICE (IDSERVICE, NAME)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
1	<b>idService</b>	varchar(4)	utf8mb4_general_ci	No	None				Change	Drop More
2	<b>name</b>	varchar(50)	utf8mb4_general_ci	No	None				Change	Drop More

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDSERVICE	Varchar(4)	Mã dịch vụ		X		X
2	NAME	Varchar(50)	Tên dịch vụ				X

✓ Bảng: CUSTOMER

CUSTOMER (IDCUSTOMER, NAME, GENDER, EMAIL, PHONE, PASSWORD, STATUS)

# BÁO CÁO ĐẠT N ĐẠI HỌC

# CHƯƠNG 4: PHÂN TÍCH THIẾT KẾ

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/>	1	<b>idCustomer</b> 	varchar(4)	utf8mb4_general_ci		No	None		 Change	 Drop	More
<input type="checkbox"/>	2	<b>name</b>	varchar(50)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More
<input type="checkbox"/>	3	<b>gender</b>	varchar(11)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More
<input type="checkbox"/>	4	<b>email</b>	varchar(50)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More
<input type="checkbox"/>	5	<b>phone</b>	varchar(10)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More
<input type="checkbox"/>	6	<b>password</b>	varchar(50)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More
<input type="checkbox"/>	7	<b>status</b>	varchar(10)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More

STT	Tên trường	Kiểu	Điễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDCUSTOMER	Varchar(4)	Mã khách hàng		X		X
2	NAME	Varchar(50)	Họ tên				
3	GENDER	Varchar(11)	Giới tính				
4	EMAIL	Varchar(50)	Email	X			
5	PHONE	Varchar(10)	Số điện thoại	X			
6	PASSWORD	Varchar(50)	Mật khẩu				
7	STATUS	Varchar(10)	Trạng thái tài khoản				

✓ Bảng: EMPLOYEE

**EMPLOYEE (IDEMPLOYEE, NAME, ADDRESS, EMAIL, PHONE, PASSWORD, ROLE)**

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/>	1	<b>idEmployee</b> 	varchar(4)	utf8mb4_general_ci		No	None		 Change	 Drop	More
<input type="checkbox"/>	2	<b>name</b>	varchar(50)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More
<input type="checkbox"/>	3	<b>address</b>	varchar(50)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More
<input type="checkbox"/>	4	<b>email</b>	varchar(50)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More
<input type="checkbox"/>	5	<b>phone</b>	varchar(10)	utf8mb4_general_ci		No	None		 Change	 Drop	More
<input type="checkbox"/>	6	<b>password</b>	varchar(50)	utf8mb4_general_ci		No	None		 Change	 Drop	More
<input type="checkbox"/>	7	<b>role</b>	varchar(10)	utf8mb4_general_ci		Yes	NULL		 Change	 Drop	More

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDEMPLOYEE	Varchar(4)	Mã nhân viên		X		X
2	NAME	Varchar(50)	Tên nhân viên				
3	ADDRESS	Varchar(50)	Địa chỉ				
4	EMAIL	Varchar(50)	Email	X			
5	PHONE	Varchar(10)	Số điện thoại	X			
6	PASSWORD	Varchar(50)	Mật khẩu				
7	ROLE	Varchar(10)	Quyền				

✓ Bảng: BOOKING

**BOOKING** (IDCUSTOMER, IDTOUR, DATEBOOKING, QUANTITYPEOPLE, TOTALPRICE, JUDGE)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
1	<u>idCustomer</u>	varchar(4)	utf8mb4_general_ci	No	<i>None</i>					
2	<u>idTour</u>	varchar(4)	utf8mb4_general_ci	No	<i>None</i>					
3	<u>dateBooking</u>	date		No	<i>None</i>					
4	<u>quantityPeople</u>	int(2)		No	<i>None</i>					
5	<u>totalPrice</u>	double		Yes	<i>NULL</i>					
6	<u>judge</u>	varchar(200)	utf8mb4_general_ci	Yes	<i>NULL</i>					

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDCUSTOMER	Varchar(4)	Mã khách hàng		X		X
2	IDTOUR	Varchar(4)	Mã tour		X		X
3	DATEBOOKING	Date	Ngày đặt				X
4	QUANTITYPEOPLE	Int(2)	Số lượng người tham gia				X
5	TOTALPRICE	Float	Tổng tiền				X
6	JUDGE	Varchar(200)	Đánh giá				

✓ Bảng: ASK

#### ASK (IDCUSTOMER, IDTOUR, QUESTION, ANSWER)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<u>idCustomer</u>	varchar(4)	utf8mb4_general_ci		No	None		Change  Drop  More	
2	<u>idTour</u>	varchar(4)	utf8mb4_general_ci		No	None		Change  Drop  More	
3	<u>question</u>	varchar(200)	utf8mb4_general_ci		No	None		Change  Drop  More	
4	<u>answer</u>	varchar(200)	utf8mb4_general_ci		Yes	NULL		Change  Drop  More	

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDCUSTOMER	Varchar(4)	Mã khách hàng		X		X
2	IDTOUR	Varchar(4)	Mã tour		X		X
3	QUESTION	Varchar(200)	Câu hỏi				X
4	ANSWER	Varchar(200)	Câu trả lời				

✓ Bảng: PLACETOUR

#### PLACETOUR (IDTOUR, IDPLACE)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<u>idTour</u>	varchar(4)	utf8mb4_general_ci		No	None		Change  Drop  More	
2	<u>idPlace</u>	varchar(4)	utf8mb4_general_ci		No	None		Change  Drop  More	

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDTOUR	Varchar(4)	Mã tour		X		X
2	IDPLACE	Varchar(4)	Mã địa danh		X		X

✓ Bảng: VEHICLETOUR

#### VEHICLETOUR (IDTOUR, IDVEHICLE)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<u>idTour</u>	varchar(4)	utf8mb4_general_ci		No	None		Change  Drop  More	
2	<u>idVehicle</u>	varchar(10)	utf8mb4_general_ci		No	None		Change  Drop  More	

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDTOUR	Varchar(4)	Mã tour		X		X
2	IDVEHICLE	Varchar(10)	Mã phương tiện		X		X

✓ Bảng: SERVICETOUR

#### SERVICETOUR (IDTOUR, IDSERVICE)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<u>idTour</u> 🔒	varchar(4)	utf8mb4_general_ci		No	None			Change  Drop More
2	<u>idService</u> 🔒	varchar(4)	utf8mb4_general_ci		No	None			Change  Drop More

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDTOUR	Varchar(4)	Mã tour		X		X
2	IDSERVICE	Varchar(4)	Mã dịch vụ		X		X

✓ Bảng: TOUR

#### TOUR (IDTOUR, TITLE, IDEMPLOYEE, DATESTART, DATEEND, PRICE, DESCRIPTION, TYPE)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<u>idTour</u> 🔒	varchar(4)	utf8mb4_general_ci		No	None			Change  Drop More
2	<u>title</u>	varchar(200)	utf8mb4_general_ci		Yes	NULL			Change  Drop More
3	<u>idEmployee</u> 🔒	varchar(4)	utf8mb4_general_ci		Yes	NULL			Change  Drop More
4	<u>dateStart</u>	date			Yes	NULL			Change  Drop More
5	<u>dateEnd</u>	date			Yes	NULL			Change  Drop More
6	<u>price</u>	double			Yes	NULL			Change  Drop More
7	<u>description</u>	varchar(300)	utf8mb4_general_ci		Yes	NULL			Change  Drop More
8	<u>type</u>	varchar(200)	utf8mb4_general_ci		Yes	NULL			Change  Drop More

STT	Tên trường	Kiểu	Diễn giải	Khóa duy nhất	Khóa chính	Khóa ngoại	Not null
1	IDTOUR	Varchar(4)	Mã tour		X		X
2	TITLE	Varchar(200)	Tiêu đề tour				X

3	IDEMPLOYEE	Varchar(4)	Mã hướng dẫn viên			X	
4	DATESSTART	Date	Ngày khởi hành				
5	DATEEND	Date	Ngày kết thúc				
6	PRICE	Float	Giá tiền				
7	DESCRIPTION	Varchar(300)	Mô tả				
8	TYPE	Varchar(200)	Loại tour				

#### 4.4 Ràng buộc toàn vẹn và tầm ảnh hưởng.

##### 4.4.1 Ràng buộc toàn vẹn miền giá trị.

Ràng buộc 1:

- Bối cảnh: TOUR
- Mô tả:
  - Giá tiền > 0
- Nội dung:
  - $\forall t \in \text{TOUR} / t.PRICE > 0$
- Tầm ảnh hưởng

Quan hệ	Thêm	Xóa	Sửa
TOUR	+	-	+(PRICE)

Ràng buộc 2:

- Bối cảnh: BOOKING
- Mô tả:
  - số lượng >0
  - tổng tiền >0
- Nội dung:
  - $\forall b \in \text{BOOKING} / b.QUANTITYPEOPLE > 0$
  - $\forall b \in \text{BOOKING} / b.TOTALPRICE > 0$
- Tầm ảnh hưởng

Quan hệ	Thêm	Xóa	Sửa
BOOKING	+	-	+(QUANTITYPEOPLE)
BOOKING	+	-	+(TOTALPRICE)

##### 4.4.2 Ràng buộc toàn vẹn liên thuộc tính.

Ràng buộc 3:

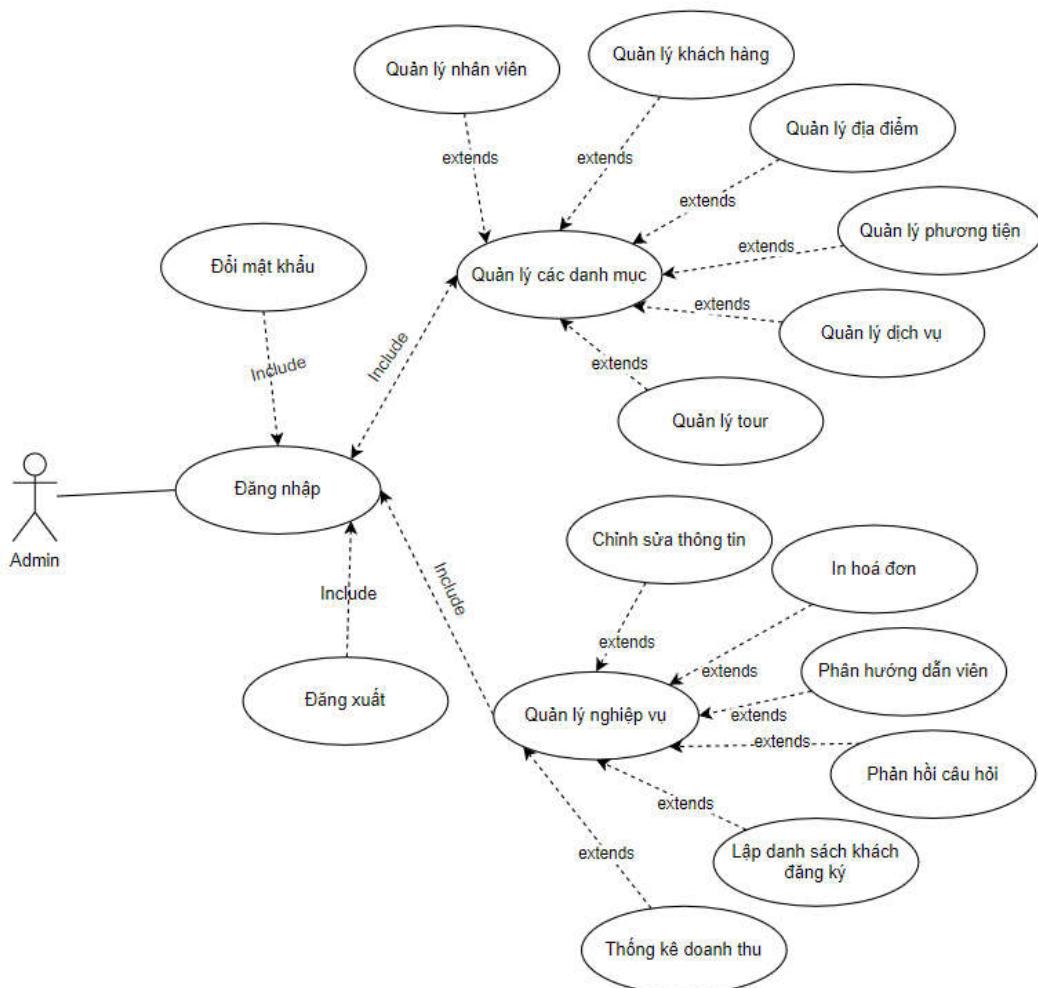
- Bối cảnh: TOUR
- Mô tả:

- Ngày kết thúc phải sau ngày bắt đầu khuyến mãi
- Nội dung:
  - $\forall t \in \text{TOUR} / t.\text{DATEEND} > d.\text{DATESTART}$
- Tâm ảnh hưởng

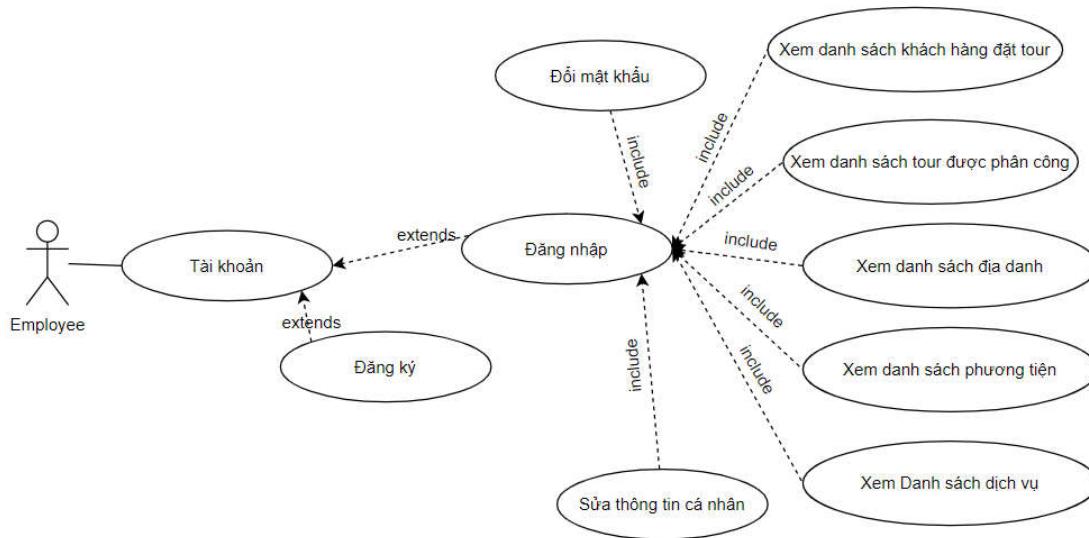
Quan hệ	Thêm	Xóa	Sửa
TOUR	+	-	+(DATESTART)
TOUR	+	-	+(DATEEND)

## 4.5 Sơ đồ Use case và sơ đồ tuần tự các chức năng.

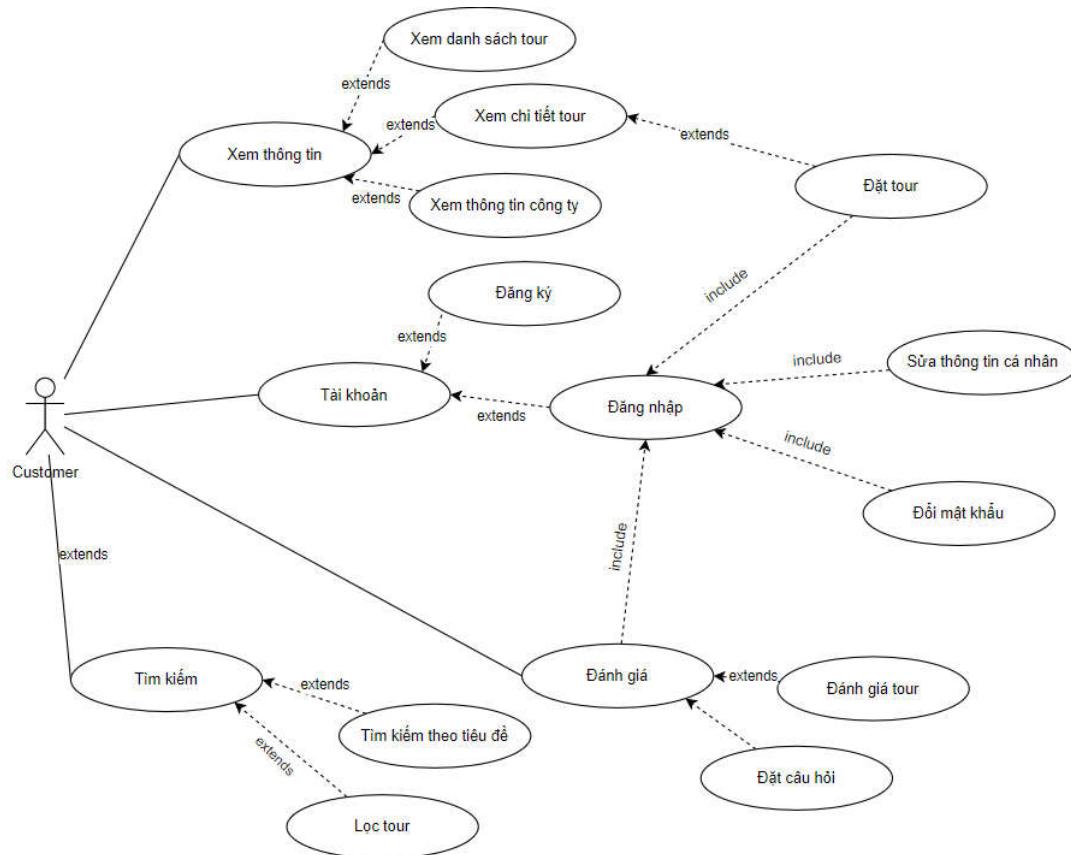
### 4.5.1 Sơ đồ Use case



Hình 4.6. Use case quản lý

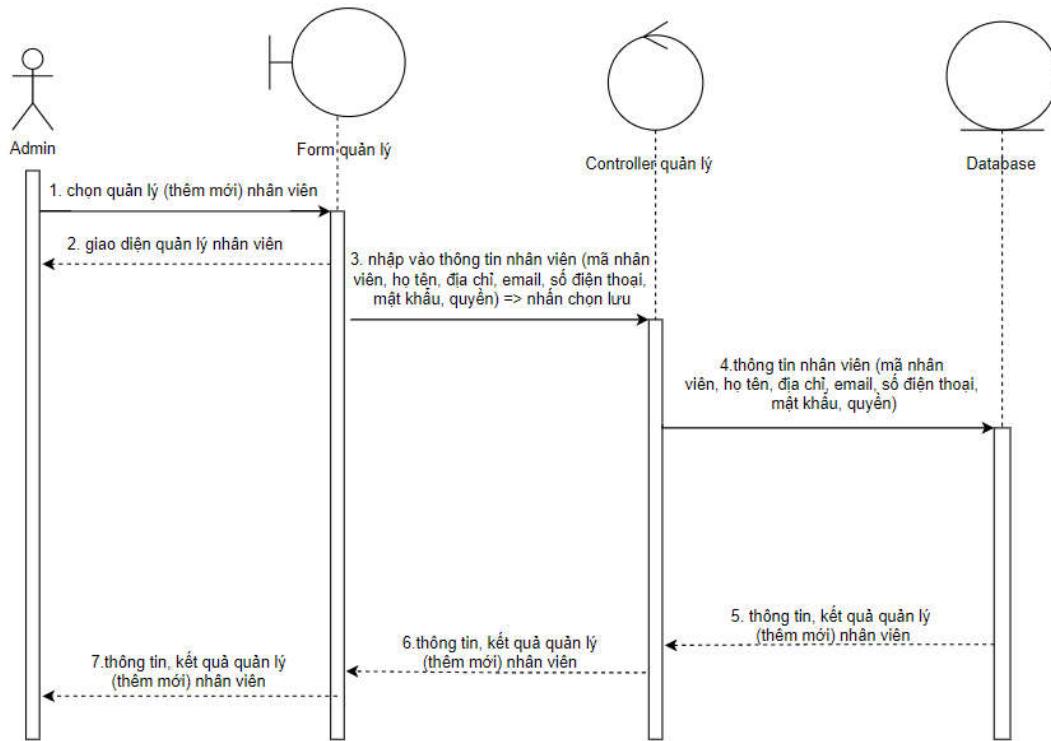


Hình 4.7. Use case nhân viên

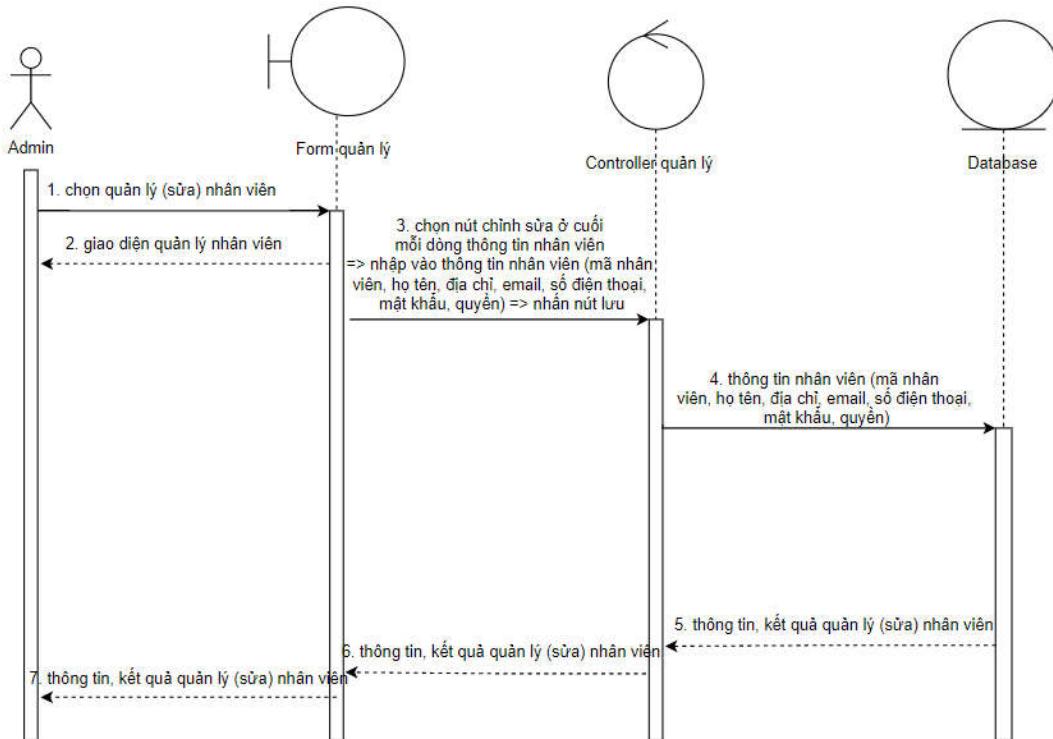


Hình 4.8. Use case khách hàng

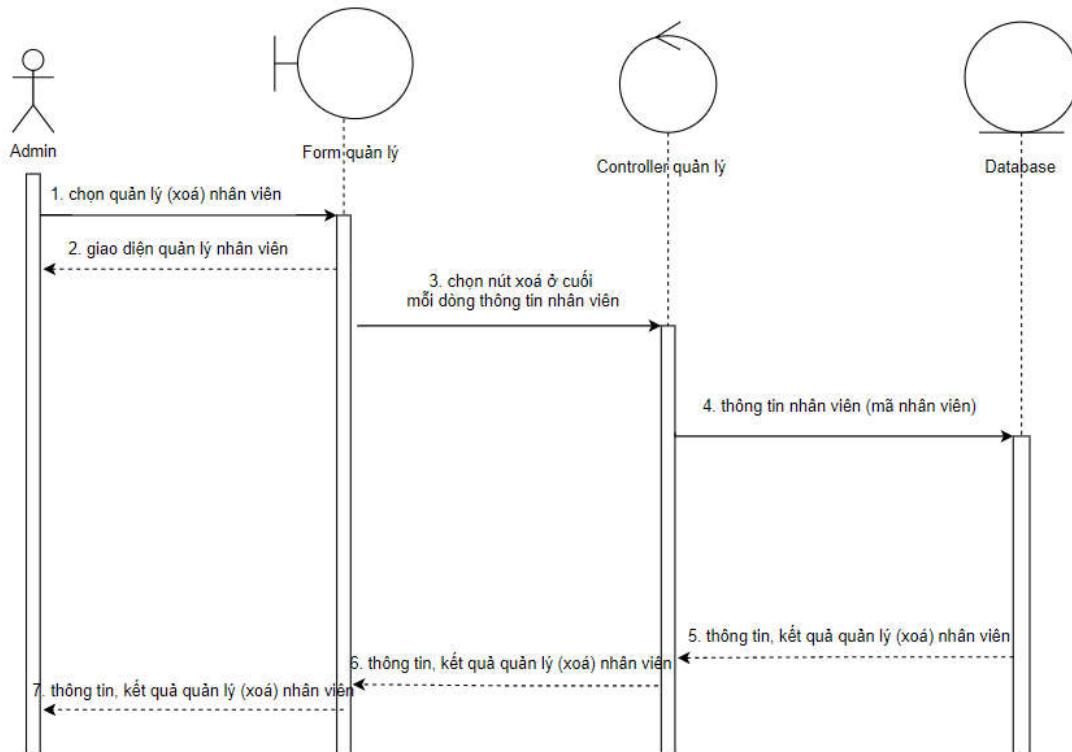
### 4.5.2 Sơ đồ tuần tự



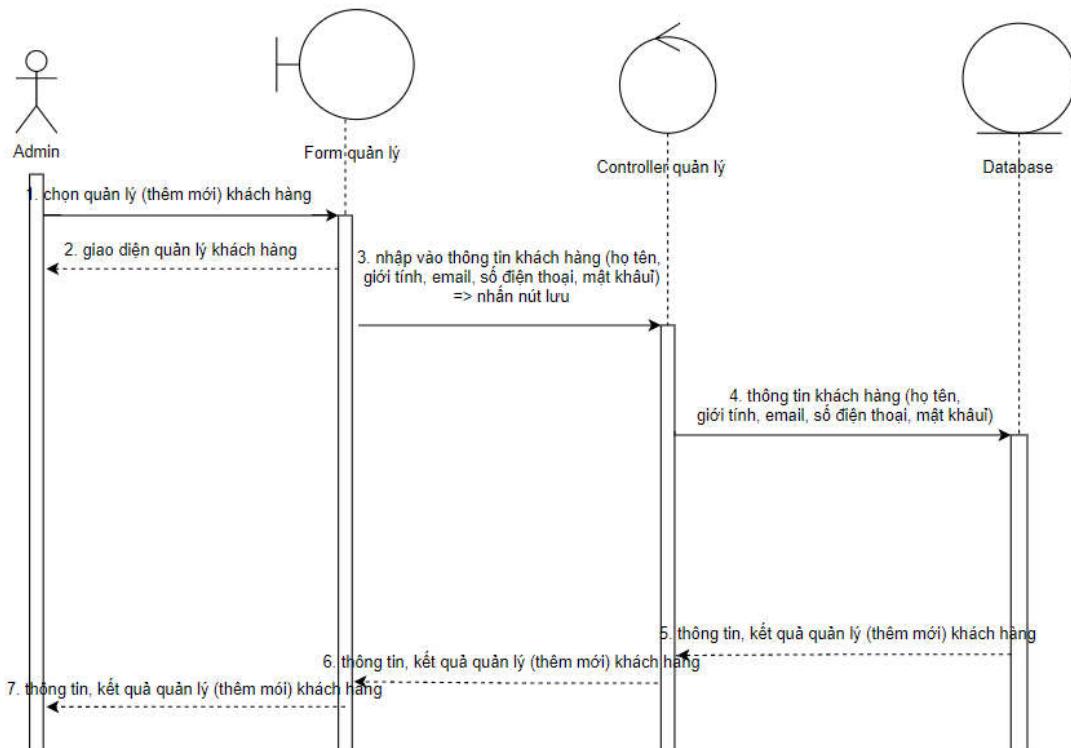
Hình 4.9: Sơ đồ tuần tự chức năng thêm mới nhân viên



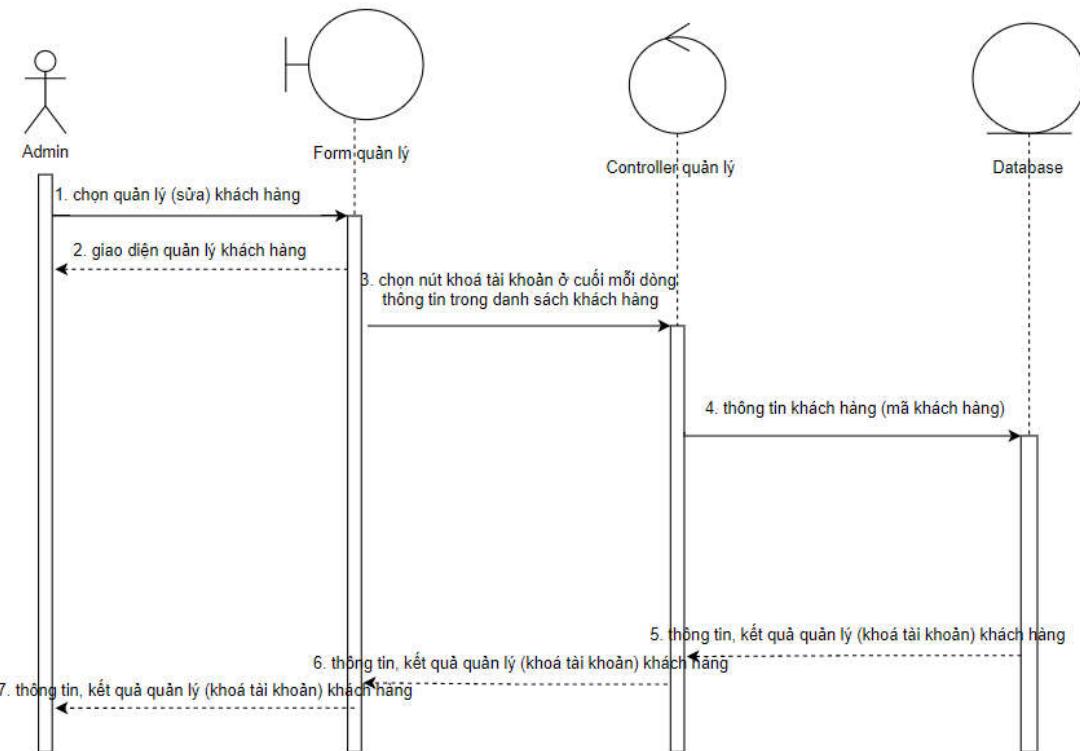
Hình 4.10: Sơ đồ tuần tự chức năng sửa nhân viên



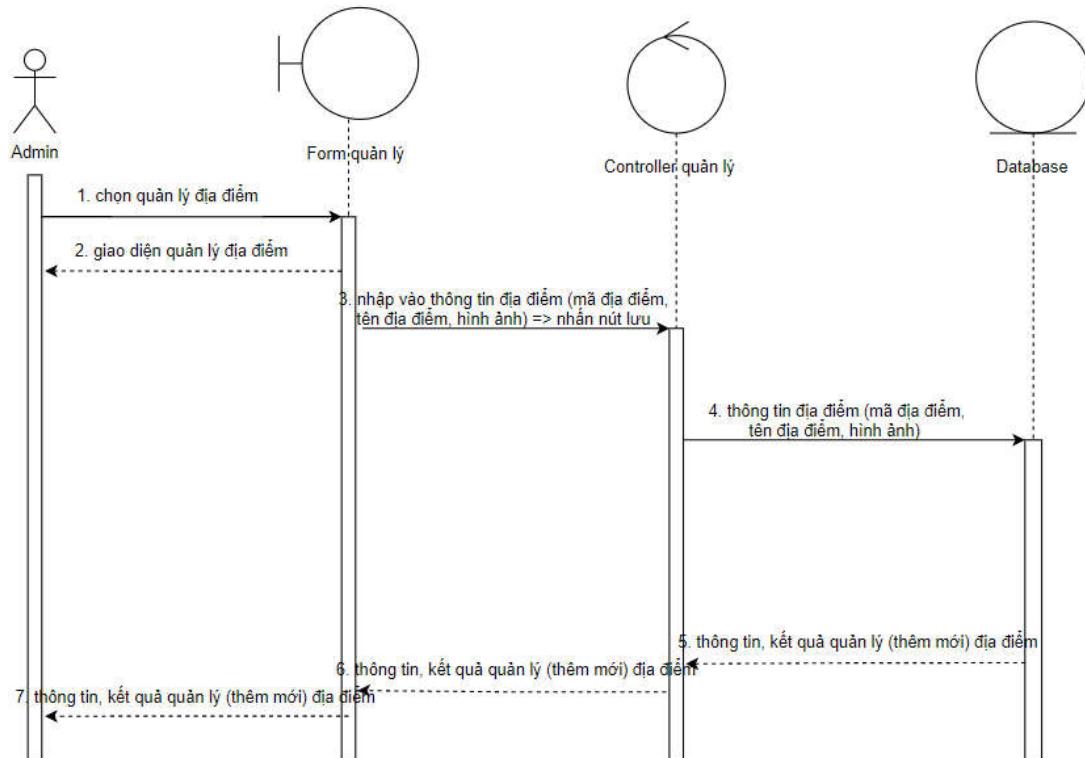
Hình 4.11: Sơ đồ tuần tự chức năng xoá nhân viên



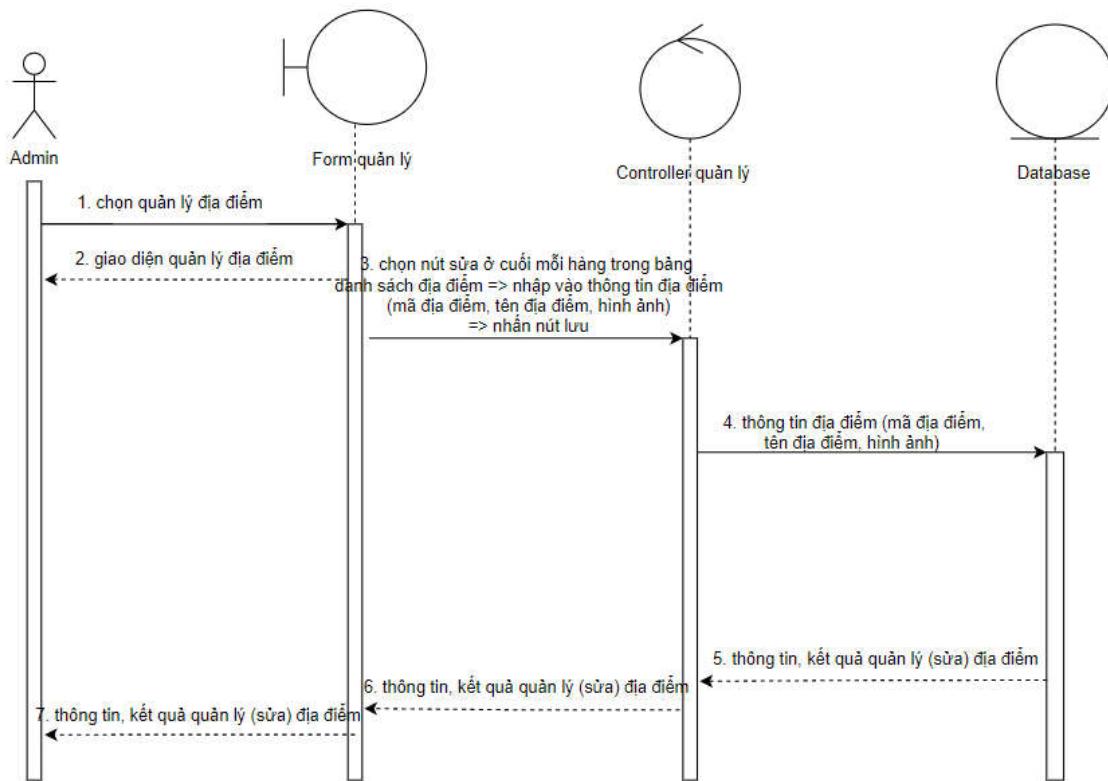
Hình 4.12: Sơ đồ tuần tự chức năng thêm mới khách hàng



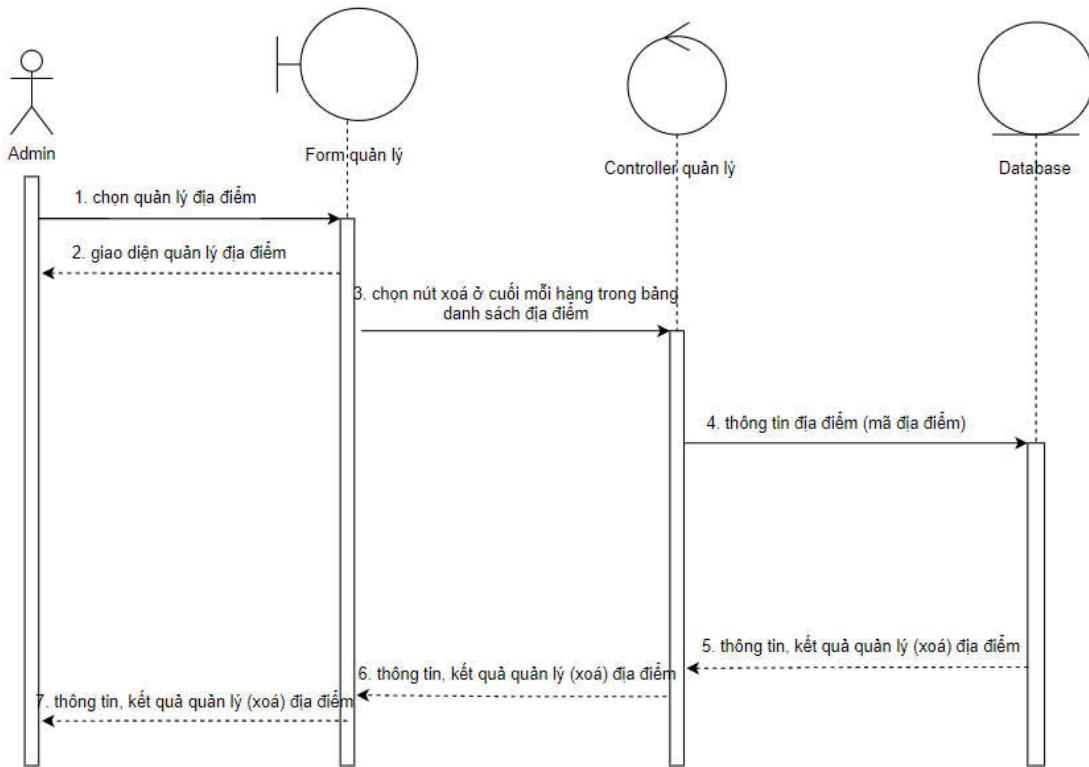
Hình 4.13: Sơ đồ tuần tự chức năng khoá tài khoản khách hàng



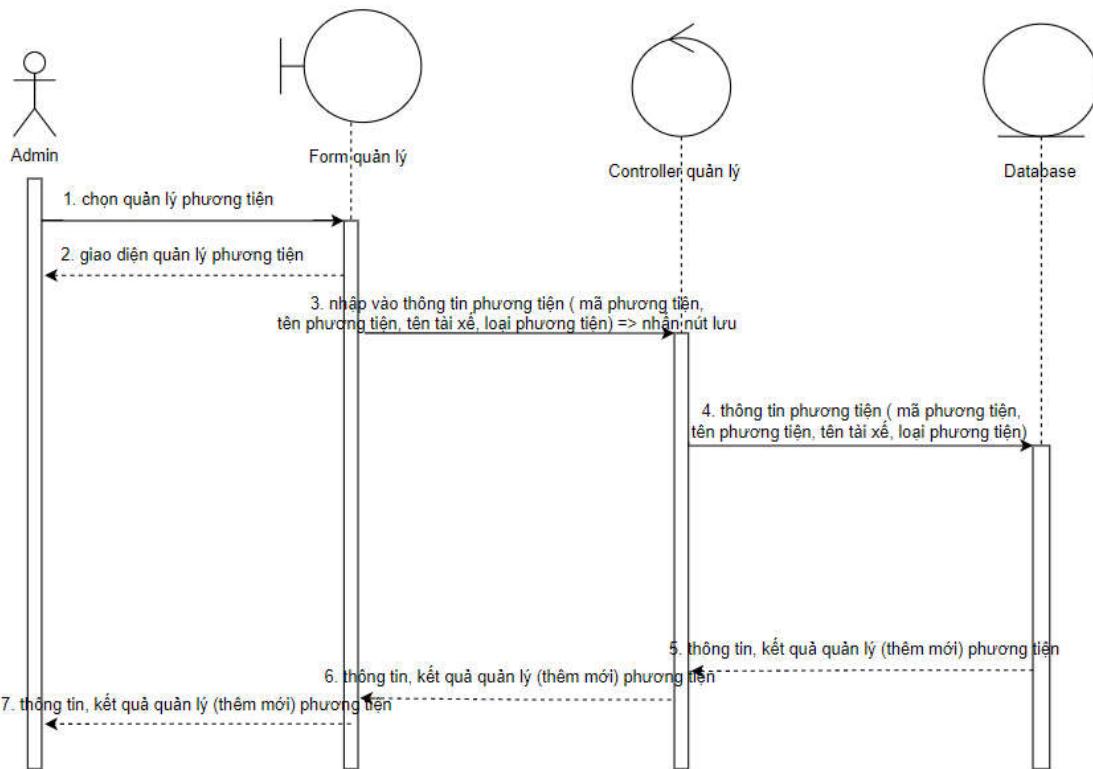
Hình 4.14: Sơ đồ tuần tự chức năng thêm mới địa điểm



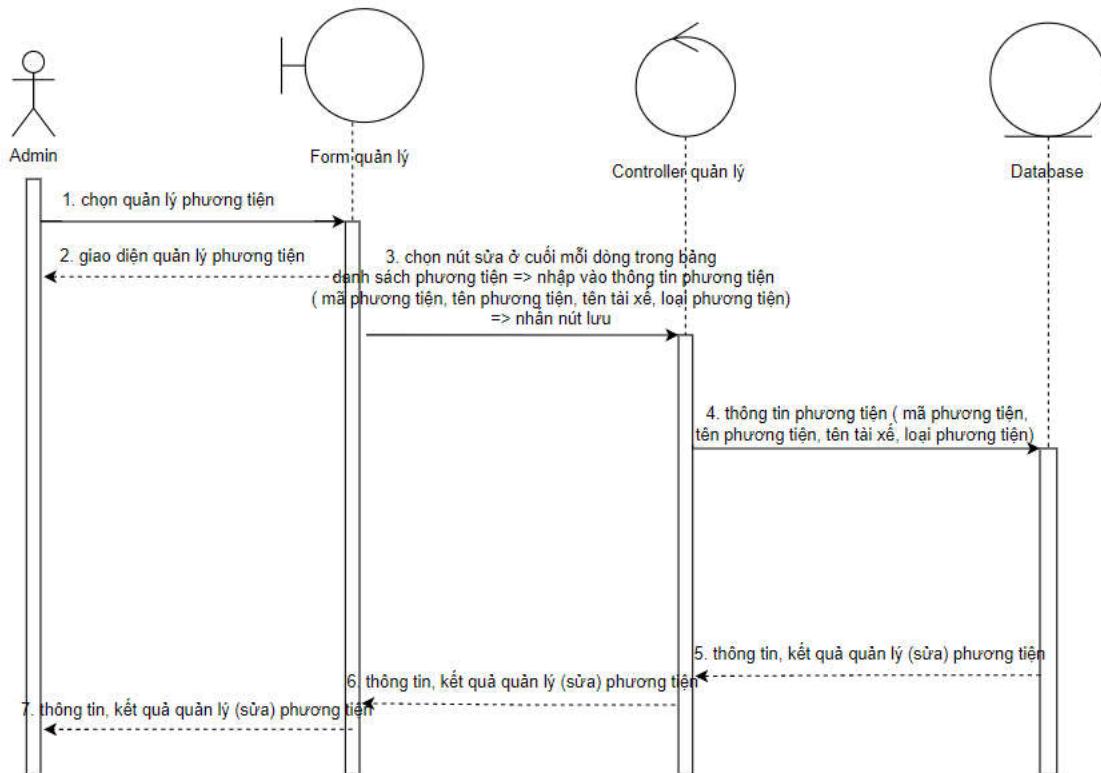
Hình 4.15: Sơ đồ tuần tự chức năng sửa địa điểm



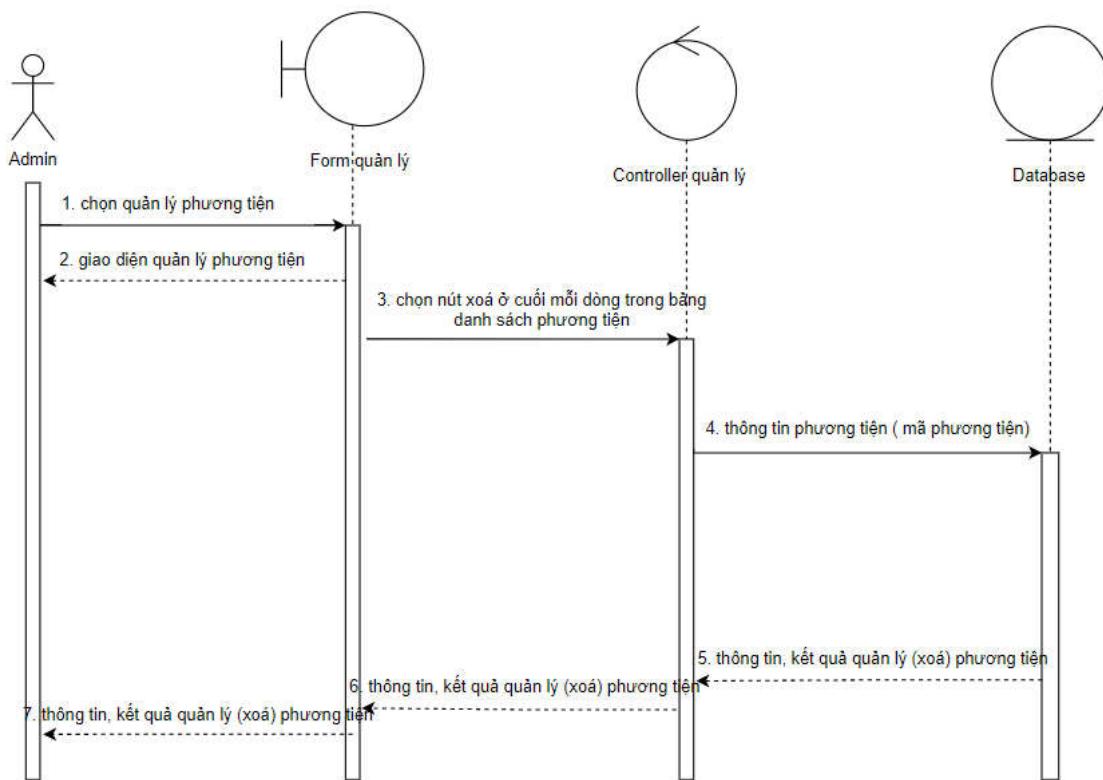
Hình 4.16: Sơ đồ tuần tự chức năng xoá địa điểm



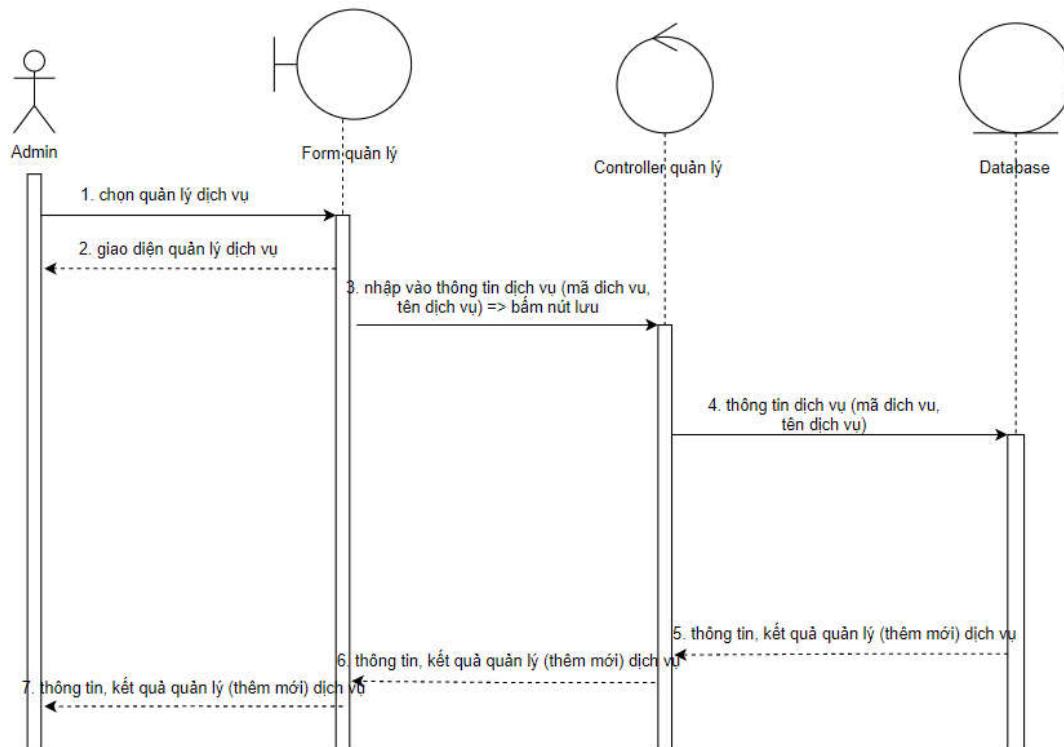
Hình 4.17: Sơ đồ tuần tự chức năng thêm mới phương tiện



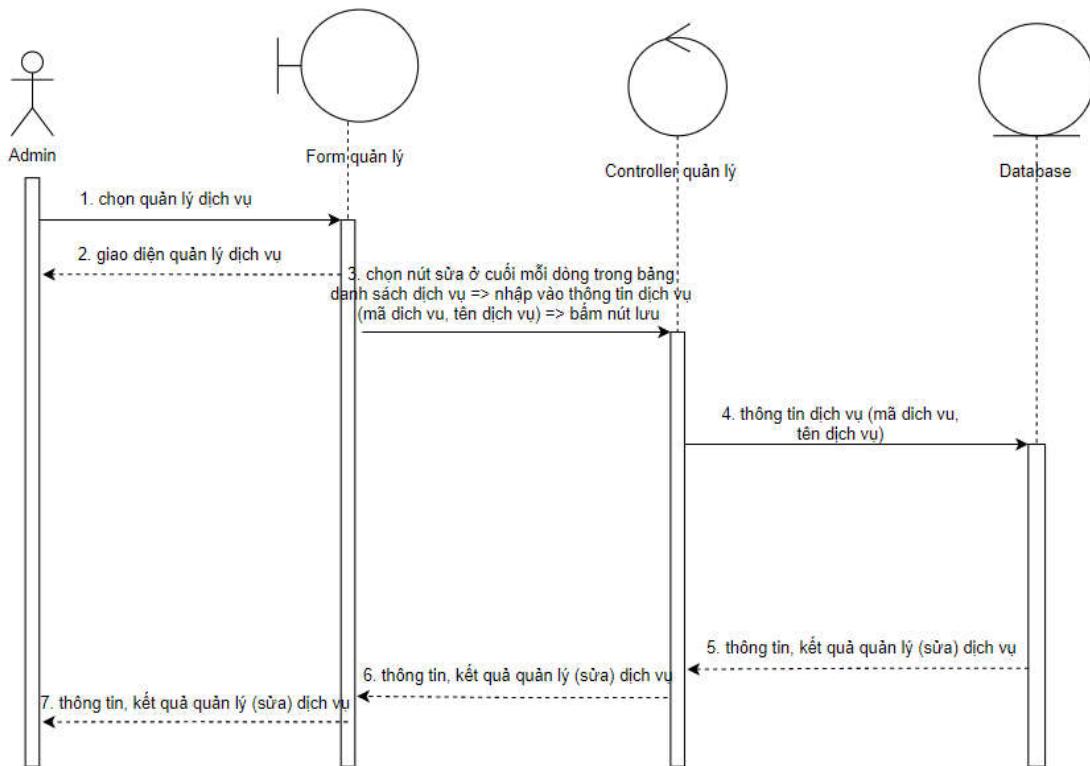
Hình 4.18: Sơ đồ tuần tự chức năng sửa phương tiện



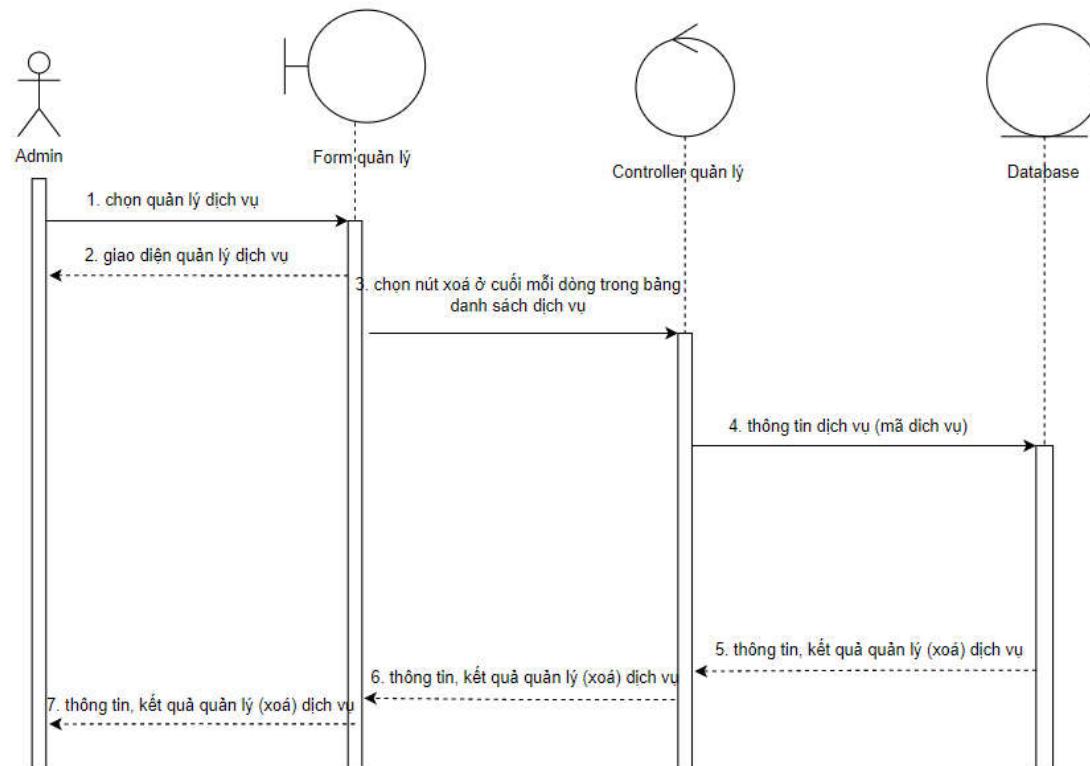
Hình 4.19: Sơ đồ tuần tự chức năng xoá phương tiện



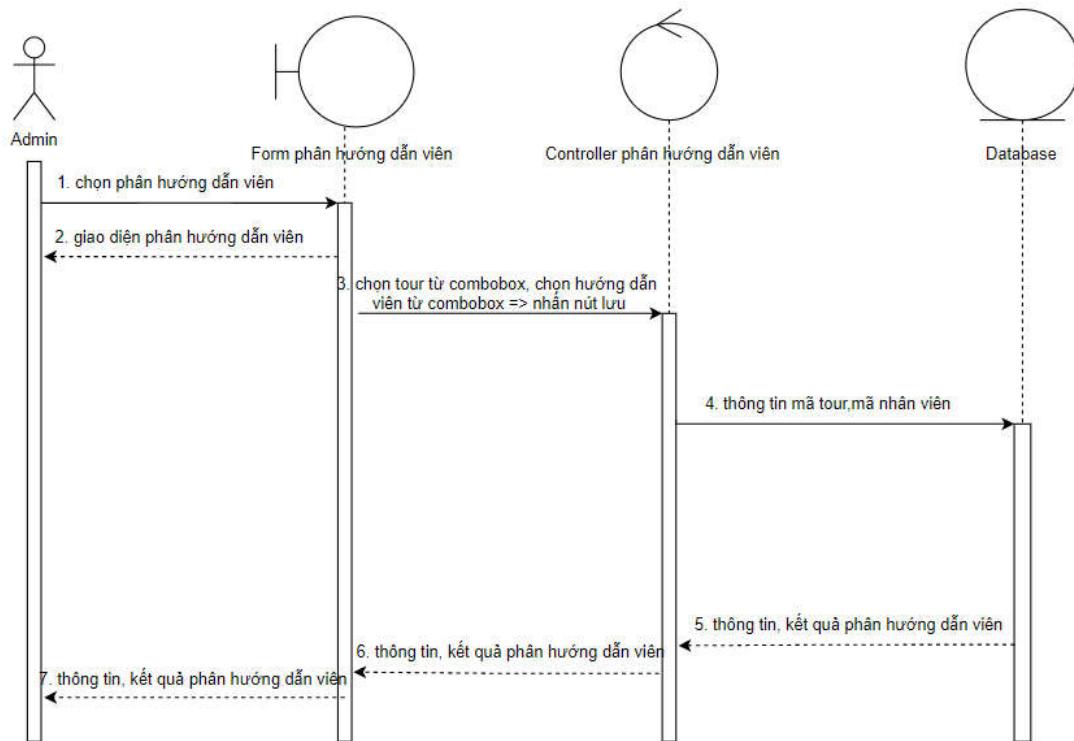
Hình 4.20: Sơ đồ tuần tự chức năng thêm mới dịch vụ



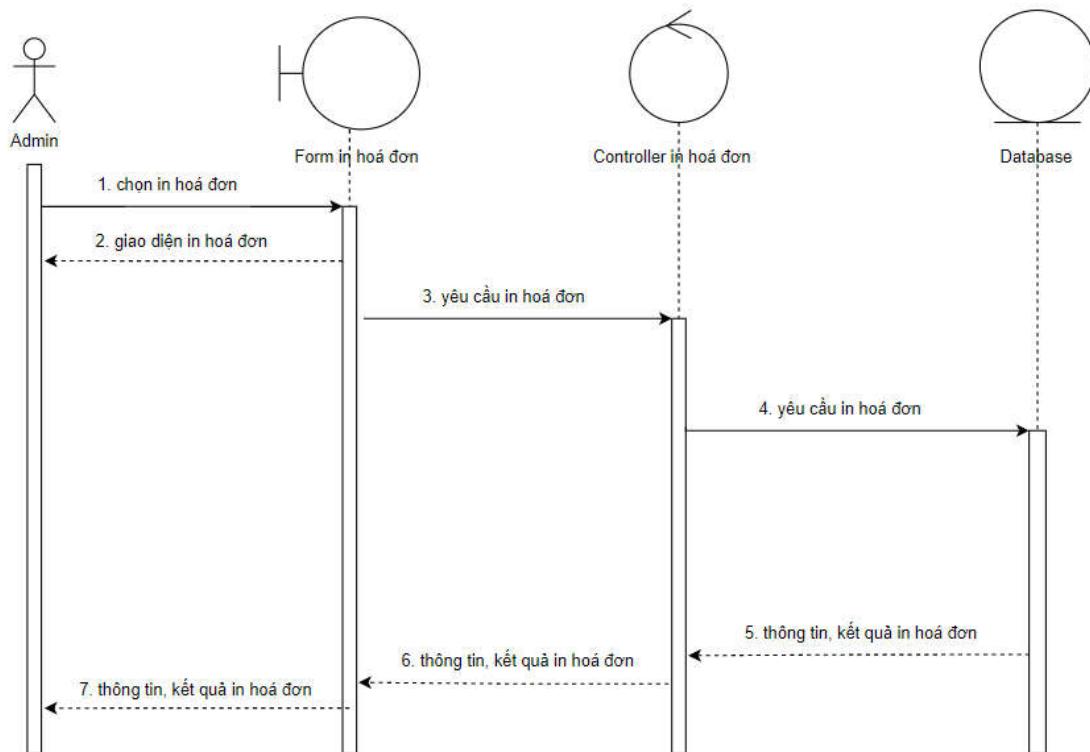
Hình 4.21: Sơ đồ tuần tự chức năng sửa dịch vụ



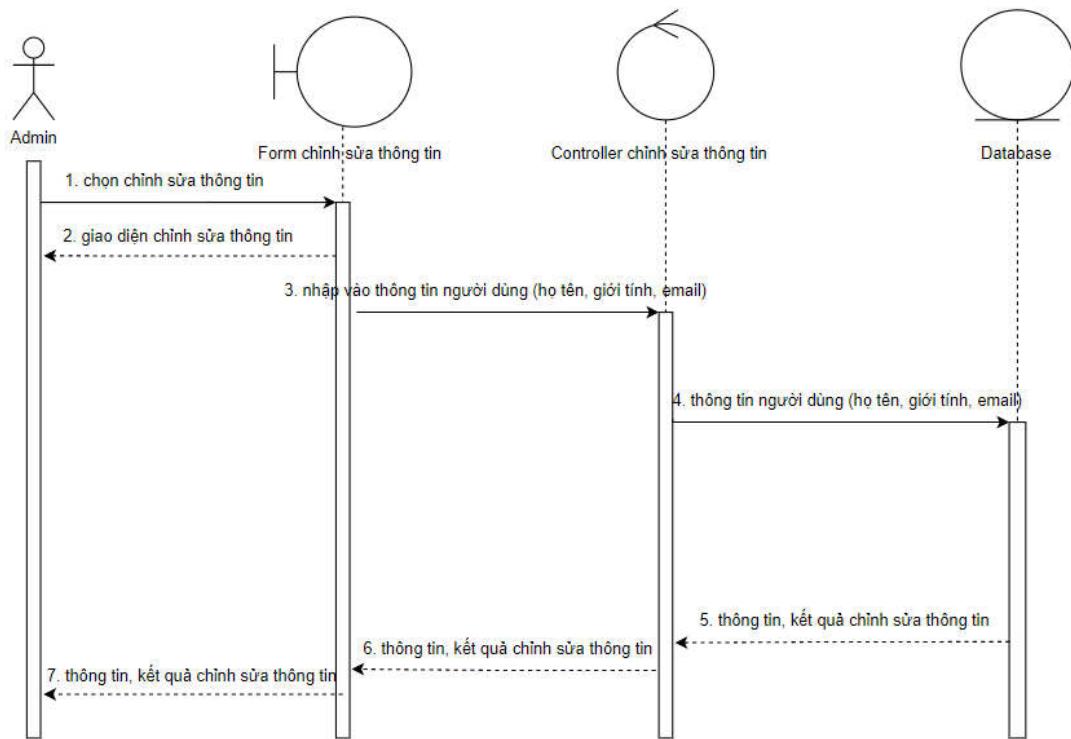
Hình 4.22: Sơ đồ tuần tự chức năng xoá dịch vụ



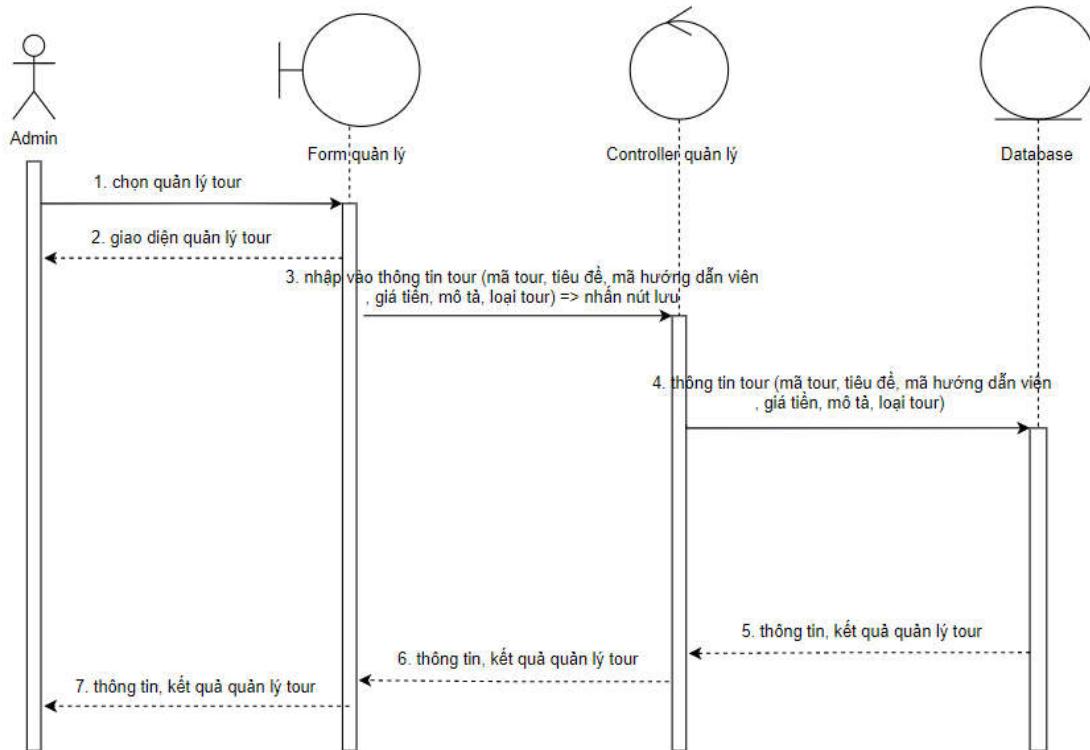
Hình 4.23: Sơ đồ tuần tự chức năng phân hướng dẫn viên



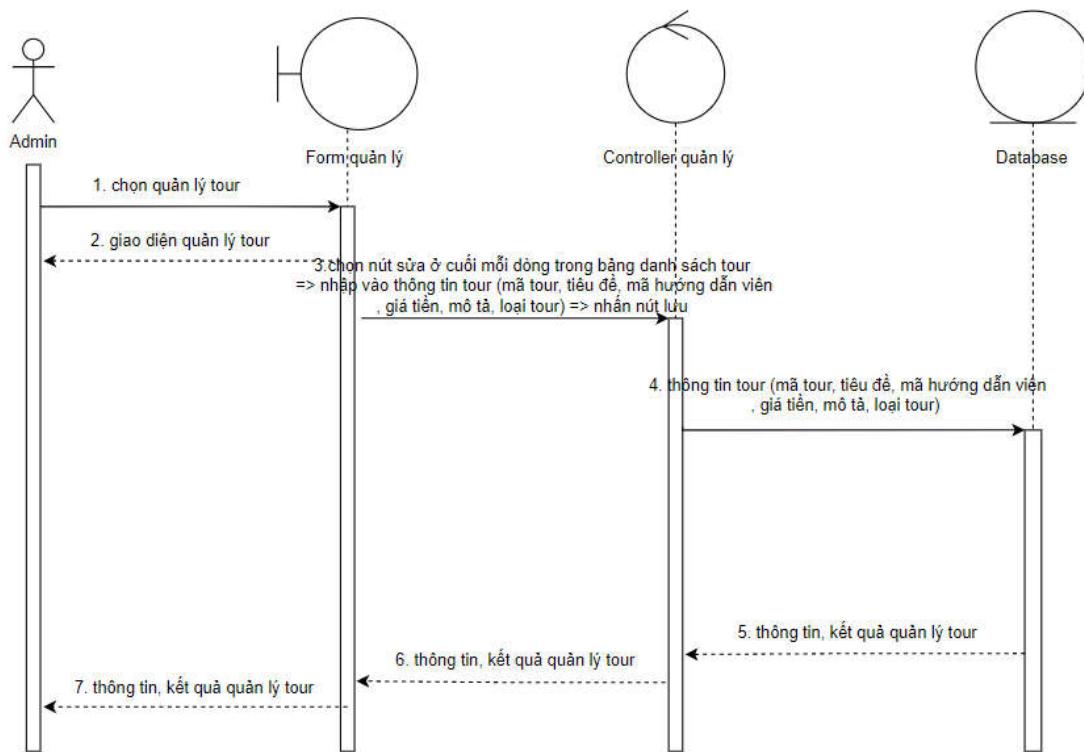
Hình 4.24: Sơ đồ tuần tự chức năng in hóa đơn



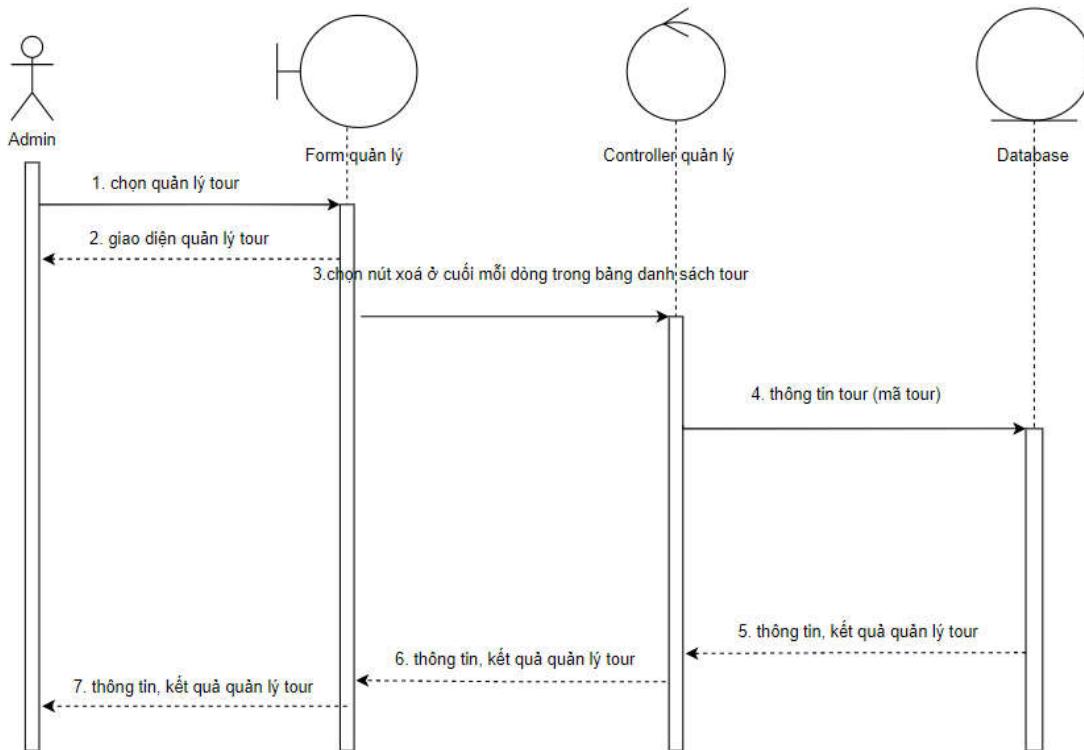
Hình 4.25: Sơ đồ tuần tự chức năng sửa thông tin



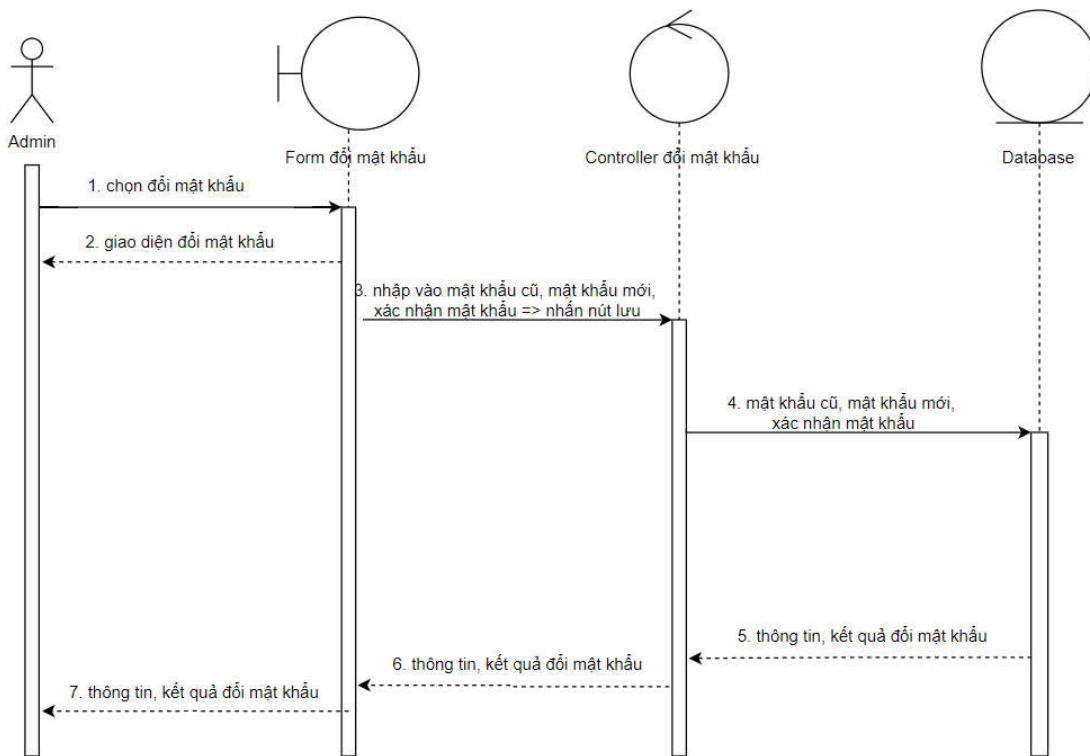
Hình 4.26: Sơ đồ tuần tự chức năng thêm mới tour



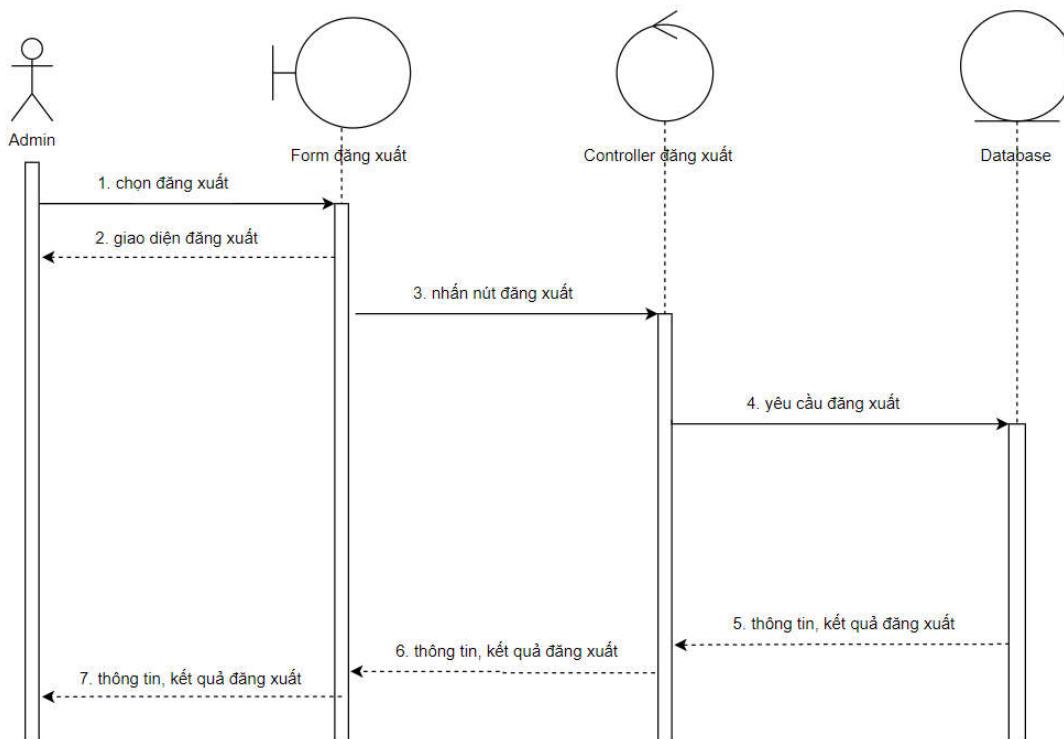
Hình 4.27: Sơ đồ tuần tự chức năng sửa tour



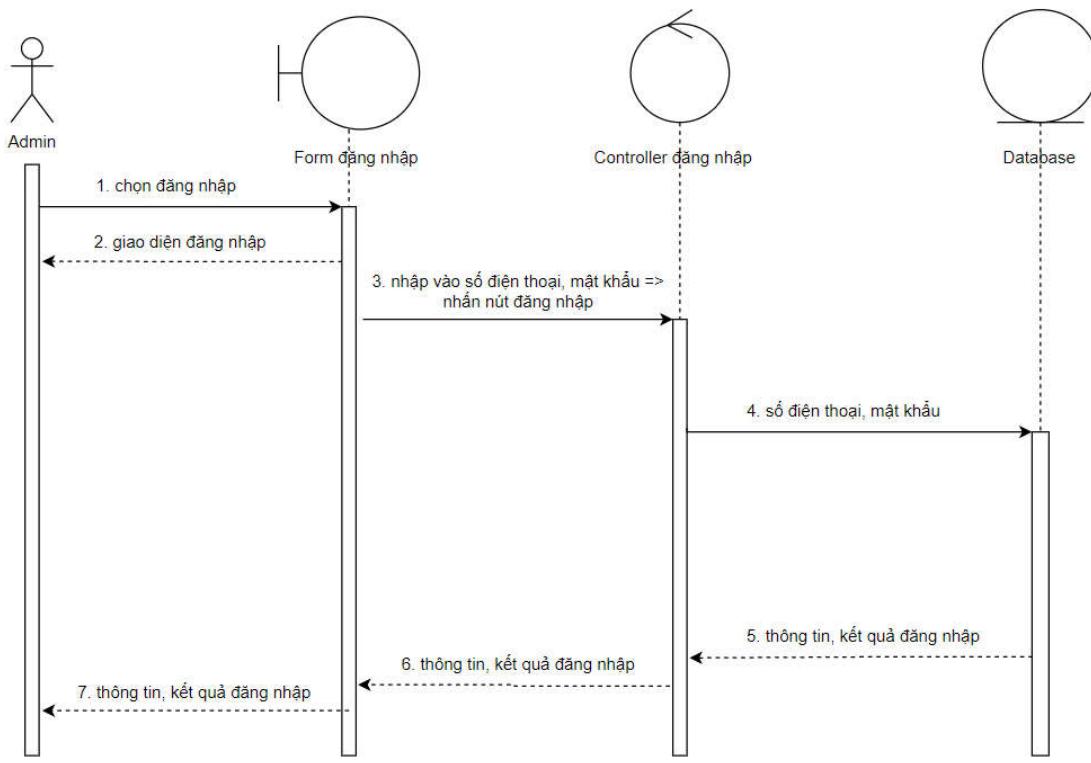
Hình 4.28: Sơ đồ tuần tự chức năng xoá tour



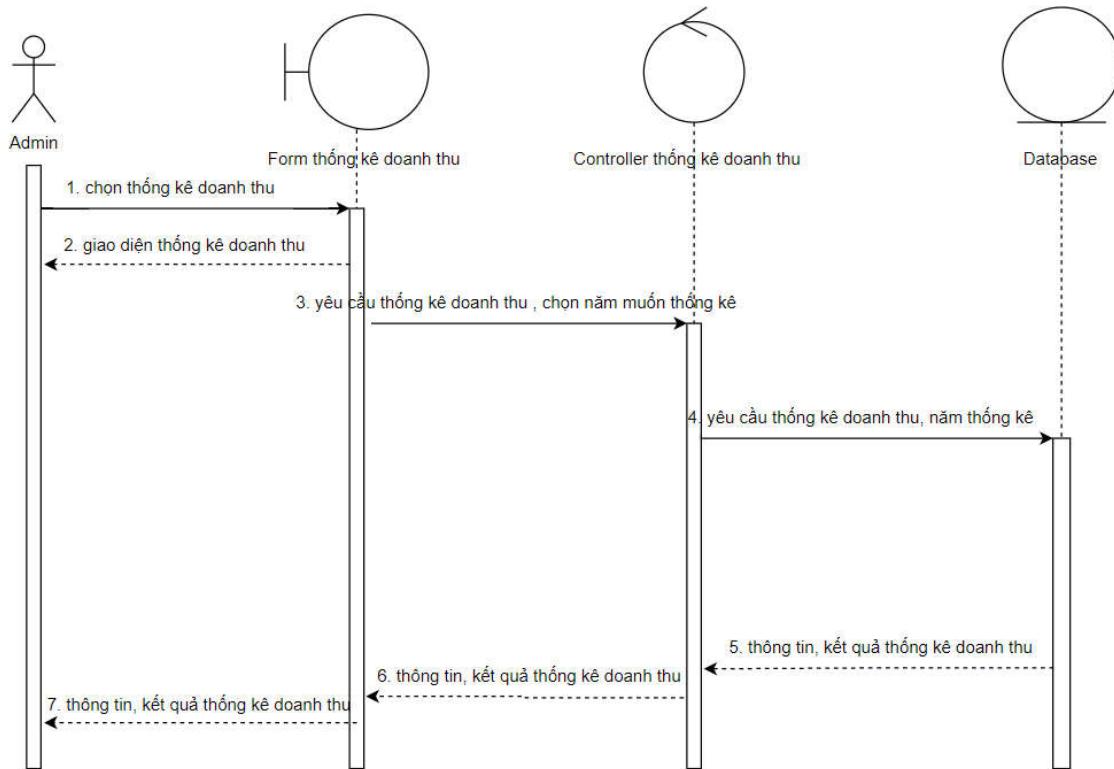
Hình 4.29: Sơ đồ tuần tự chức năng đổi mật khẩu



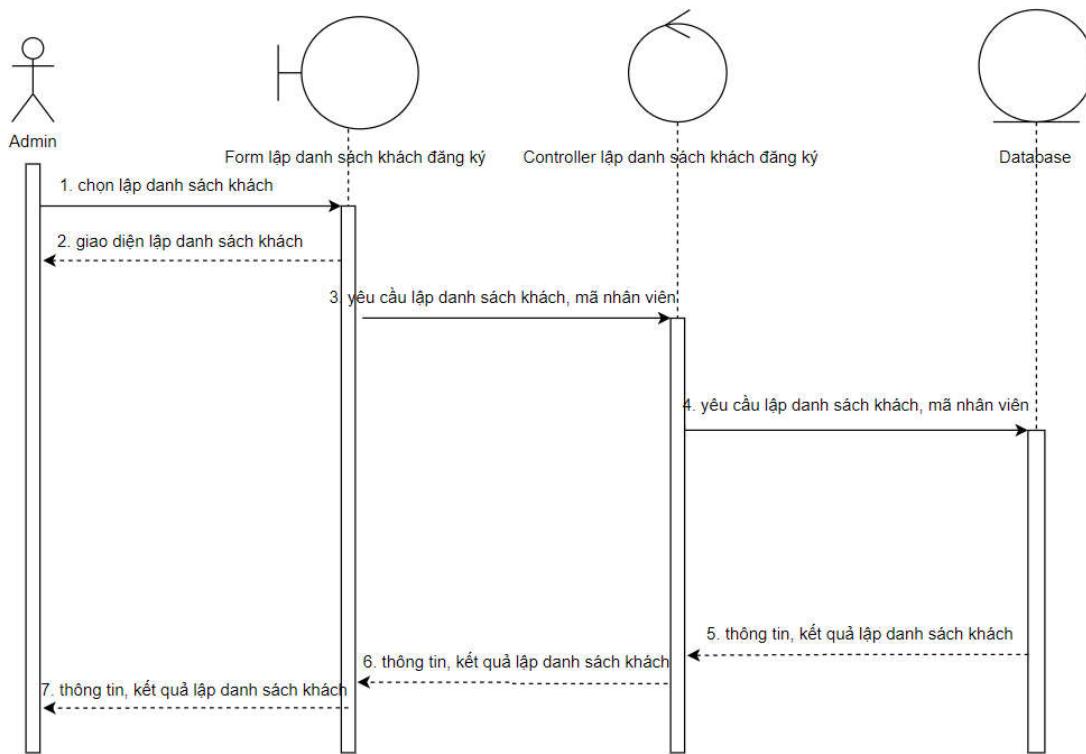
Hình 4.30: Sơ đồ tuần tự chức năng đăng xuất



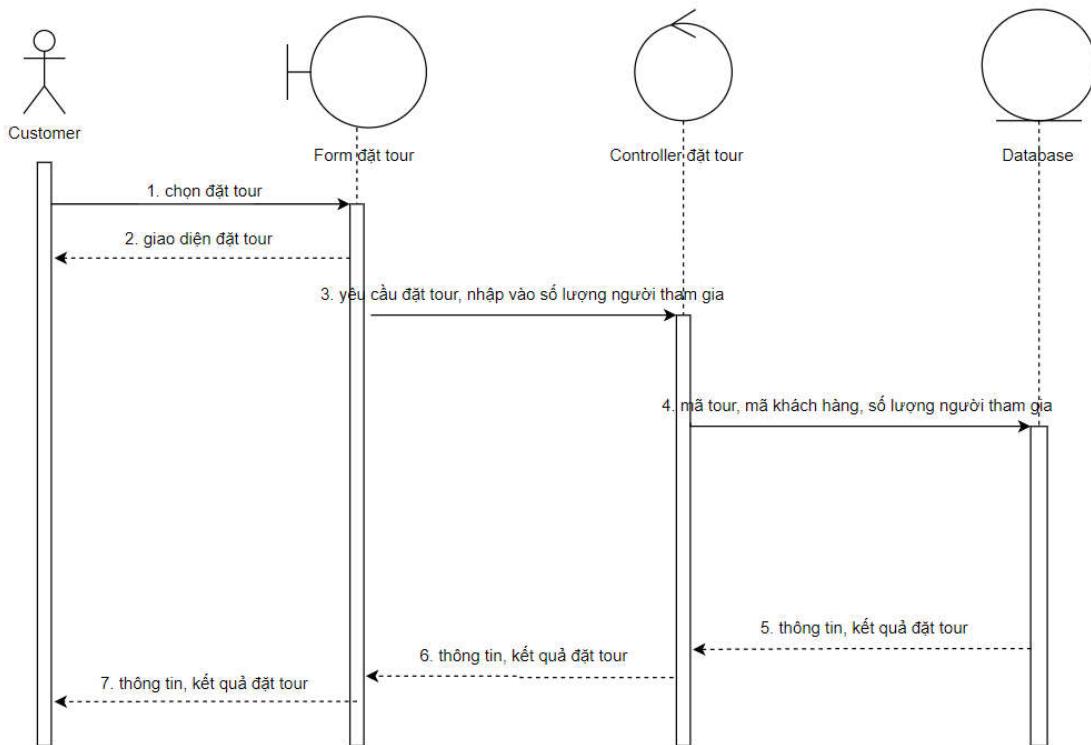
Hình 4.31: Sơ đồ tuần tự chức năng đăng nhập



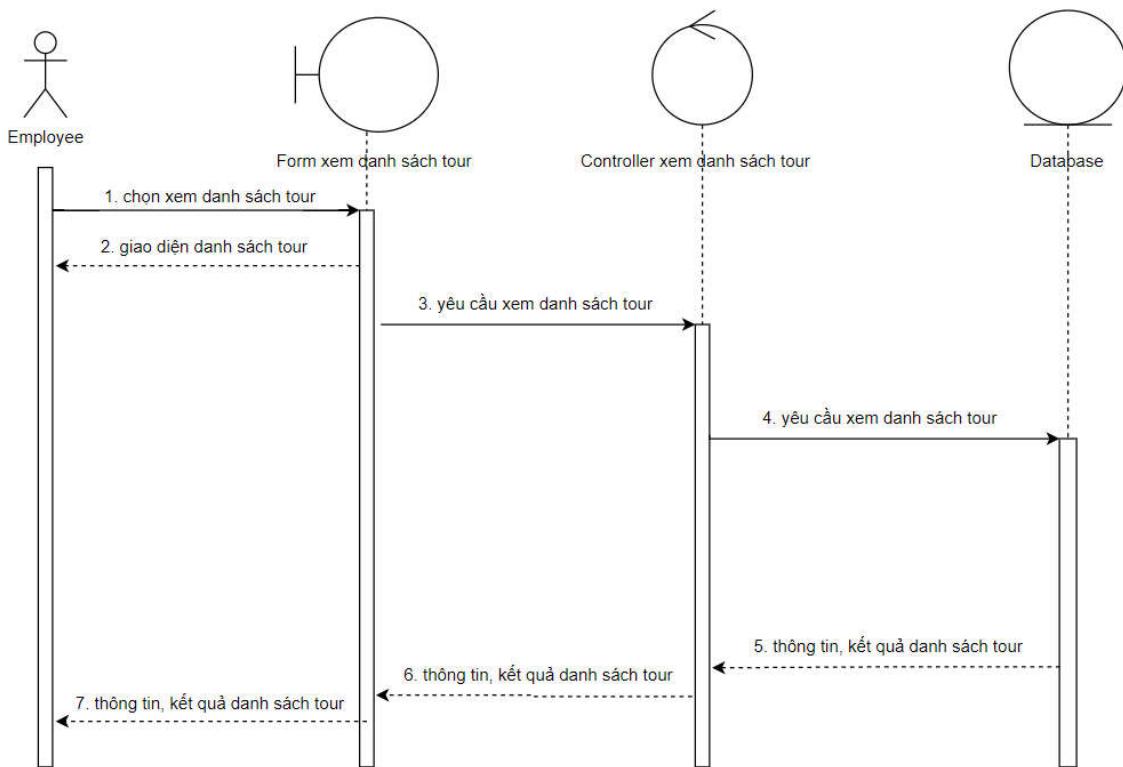
Hình 4.32: Sơ đồ tuần tự chức năng thống kê doanh thu



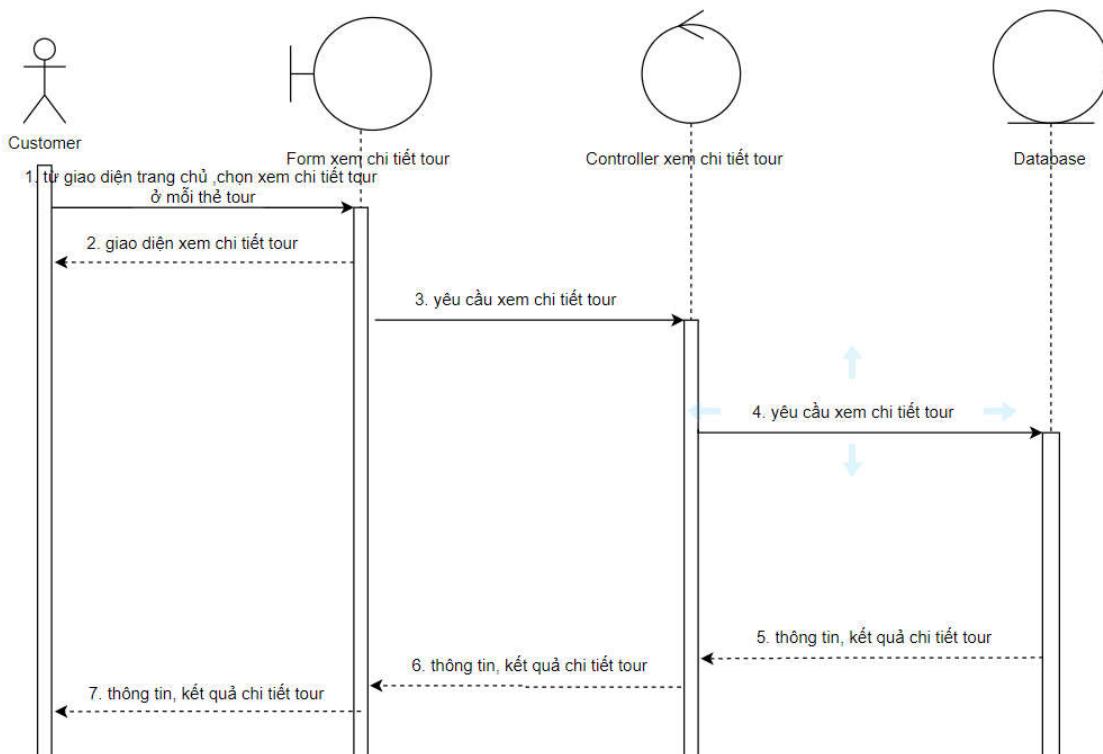
Hình 4.33: Sơ đồ tuần tự chức năng lập danh sách khách đăng kí



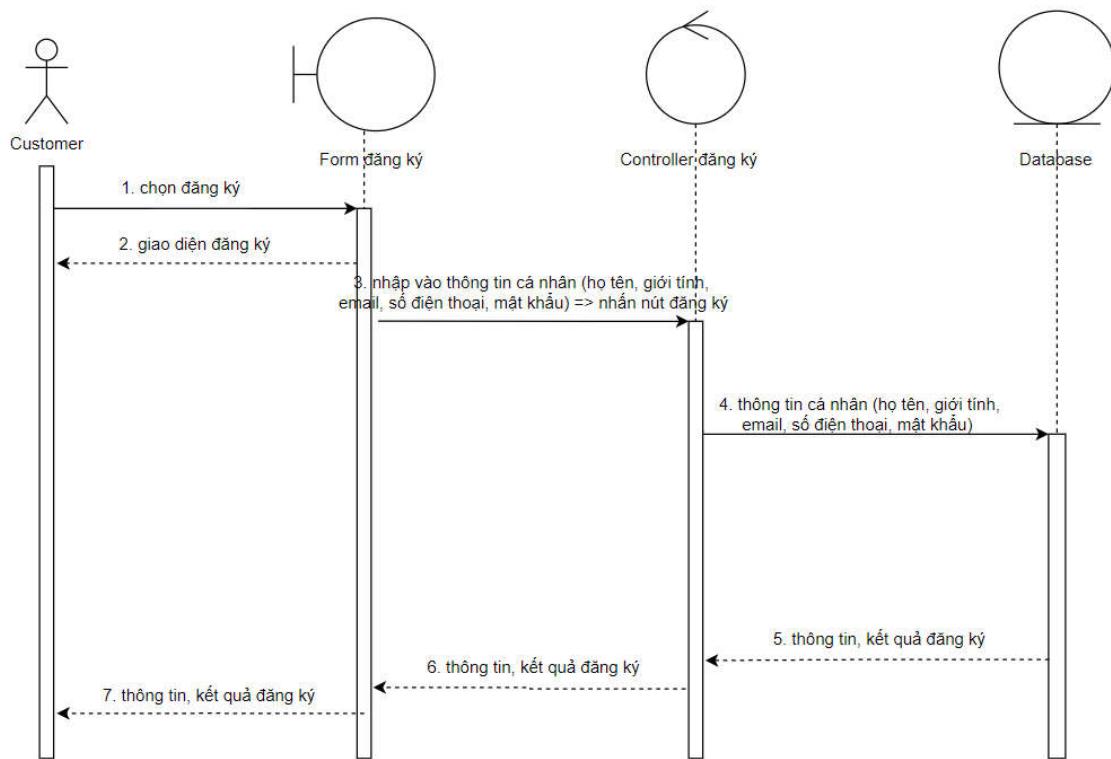
Hình 4.34: Sơ đồ tuần tự chức năng đặt tour



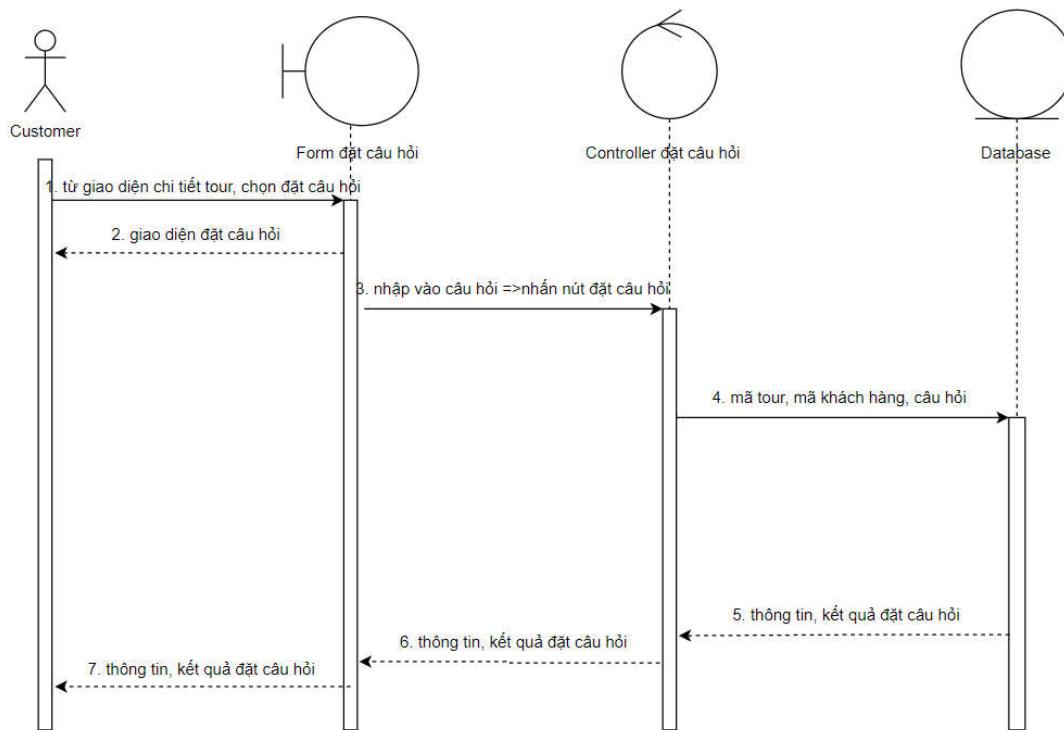
Hình 4.35: Sơ đồ tuần tự chức năng xem danh sách tour



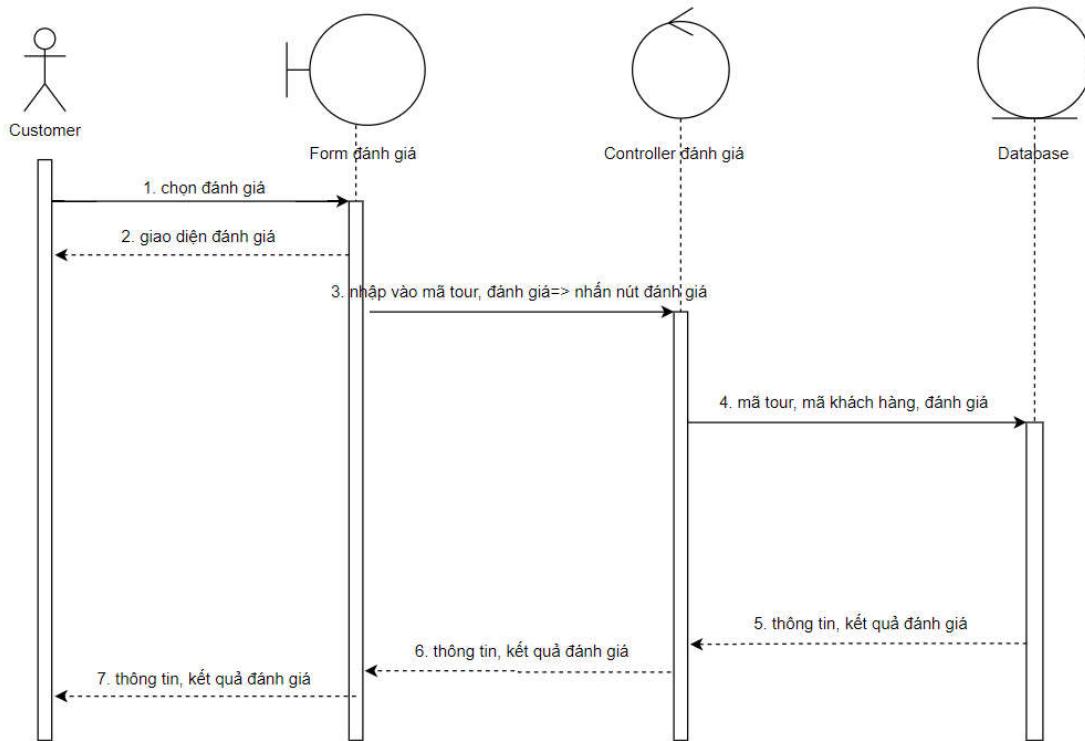
Hình 4.36: Sơ đồ tuần tự chức năng xem chi tiết tour



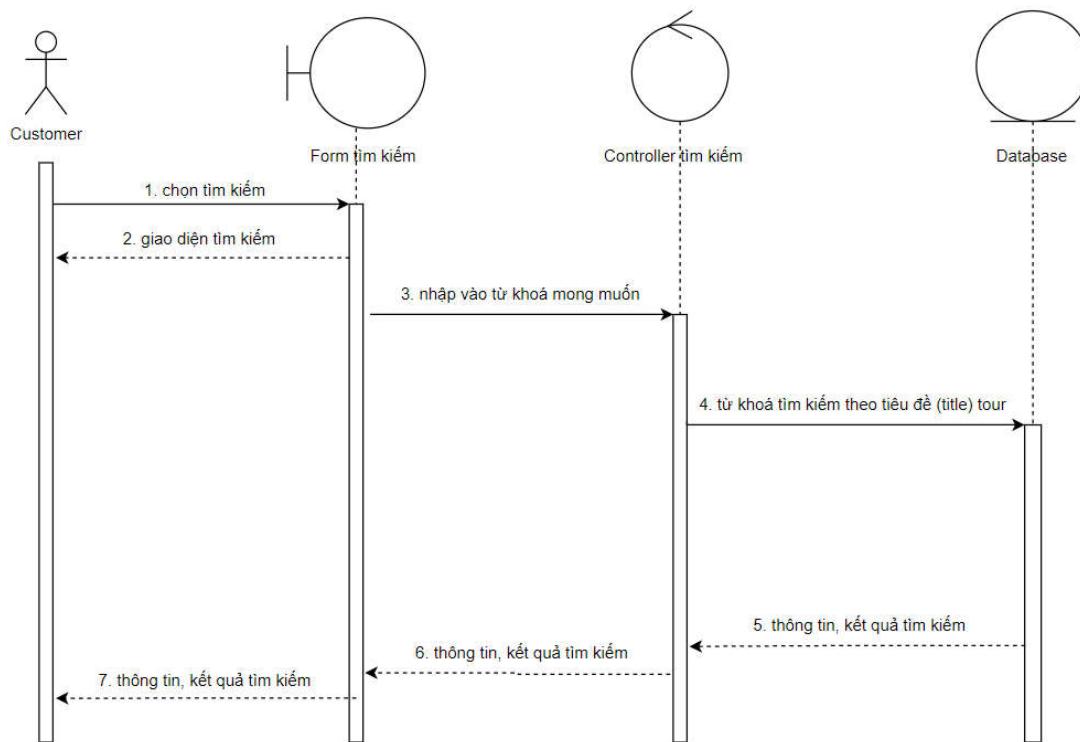
Hình 4.37: Sơ đồ tuần tự chức năng đăng ký



Hình 4.38: Sơ đồ tuần tự chức năng đặt câu hỏi

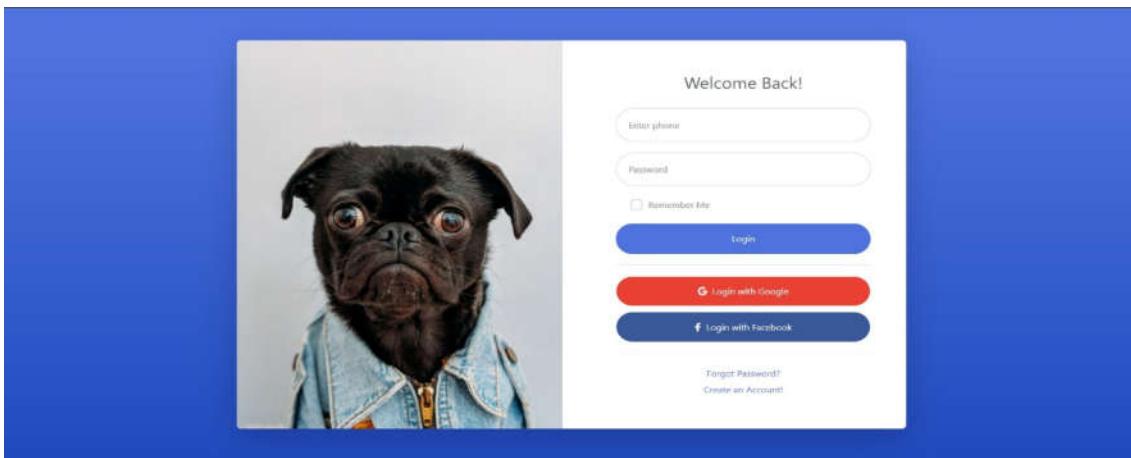


Hình 4.39: Sơ đồ tuần tự chức năng đánh giá tour



Hình 4.40: Sơ đồ tuần tự chức năng tìm kiếm

## CHƯƠNG 5: DEMO CHƯƠNG TRÌNH



Hình 5.1: Giao diện đăng nhập trang web quản lý

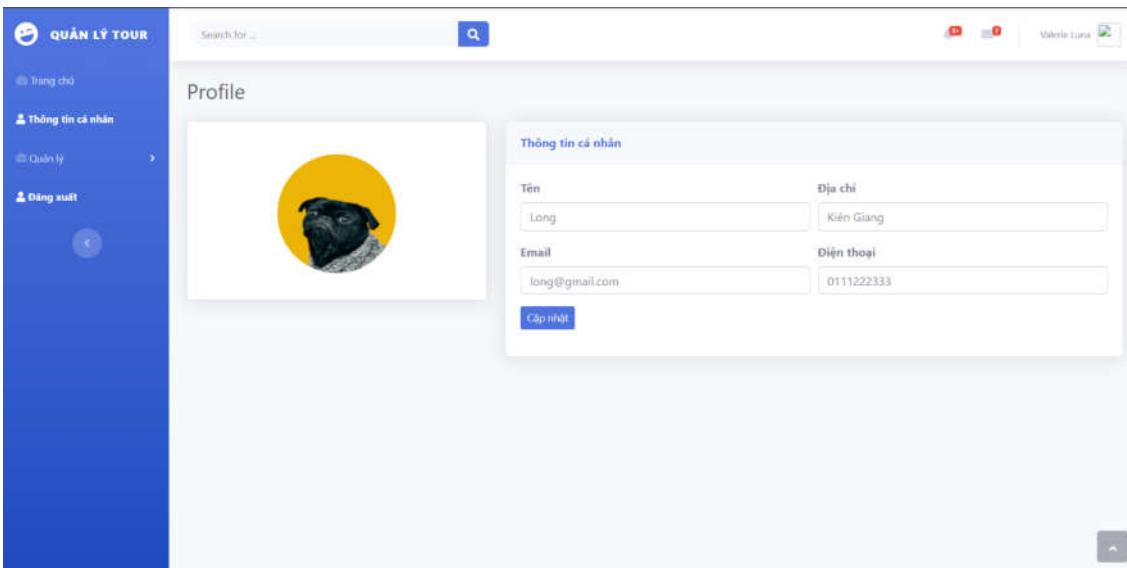
Mô tả: (form đăng nhập)

- Các ô nhập dữ liệu: số điện thoại (phone), mật khẩu (password)
- Nút xử lý: Login
- Xử lý: khi nhấn nút “Login” giao diện lấy dữ liệu từ các ô nhập dữ liệu gửi lên hệ thống, hệ thống gọi API loginEmployee, truyền vào (phone, password), API thực hiện so sánh với dữ liệu trong cơ sở dữ liệu, nếu thông tin đăng nhập chính xác thì trả về thông tin người dùng đã đăng nhập rồi chuyển sang giao diện trang thông tin cá nhân.

```
@GetMapping("/employee/Login/{phone}/{pass}")
public Employee loginEmployee(@PathVariable(name = "phone") String phone, @PathVariable(name = "pass") String pass){
    return travelService.loginEmployee(phone, pass);
}
```

```
public Employee loginEmployee(String phone, String pass){
    Employee employee= employeeRepository.checkLogin(phone, pass);
    if (employee.getIdEmployee().equals("")){
        return null;
    }
    return employee;
}
```

```
@Query("SELECT e FROM Employee e WHERE e.phone = ?1 AND e.password = ?2 ")
public Employee checkLogin(String phone, String password);
```



Mô tả: (form thông tin cá nhân)  
Hình 5.2: Giao diện thông tin cá nhân trang web quản lý

- Các ô nhập dữ liệu: tên (name), địa chỉ (address), email(email), điện thoại(phone) vừa để hiển thị thông tin cá nhân của người dùng, vừa để nhập vào thông tin để thực hiện chức năng sửa thông tin cá nhân.
- Nút xử lý: “Cập nhật” dùng để cập nhật lại thông tin cá nhân của người dùng, hệ thống gọi API updateEmployee, truyền vào (idEmployee, name, address, email), API thực hiện kiểm tra xem người dùng có tồn tại chưa, nếu chưa có thì không cập nhật, nếu có thì tiến hành cập nhật lại thông tin người dùng trong CSDL, rồi trả về thông tin người dùng sau khi đã chỉnh sửa.

```

@Data
public class EmployeeUpdateInput {
    private String idEmployee;
    private String name;
    private String address;
    private String email;
}

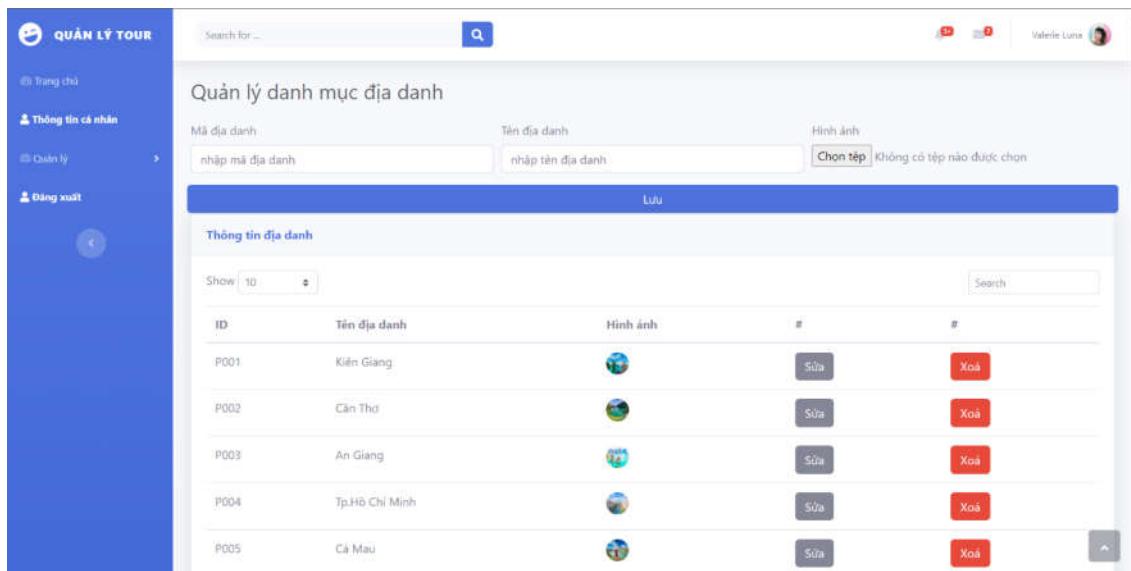
@PutMapping("/employee")
public Employee updateEmployee(@RequestBody EmployeeUpdateInput input){
    return travelService.updateEmployee( input);
}

public Employee updateEmployee(EmployeeUpdateInput input){
    Employee employee = getEmployeeById(input.getIdEmployee());

    if (employee != null){
        if (input.getName() != null) employee.setName(input.getName());
        if (input.getAddress() != null )employee.setAddress(input.getAddress());
        if (input.getEmail() != null) employee.setEmail(input.getEmail());

        return employeeRepository.save(employee);
    }
    return null;
}

```



Hình 5.3: Giao diện quản lý địa danh trang web quản lý

Mô tả: (form quản lý địa danh)

- Các ô nhập dữ liệu: mã địa danh (idPlace), tên địa danh (name), hình ảnh (image) nhập vào thông tin để thực hiện chức năng thêm mới địa danh.
- Bảng hiển thị: danh sách địa danh (idPlace, name, image)
- Nút xử lý: “Lưu” dùng để tạo mới địa danh, nút “Sửa” dùng để hiển thị popup sửa thông tin địa danh, nút “Xoá” dùng để thực hiện chức năng xoá địa danh, gọi API deletePlace, truyền vào idPlace.
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API createPlace, truyền vào (idPlace, name, image), API kiểm tra xem địa danh đó đã tồn tại chưa, nếu đã tồn tại thì không cho thêm mới, nếu chưa tồn tại thì thực hiện lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng (Place) trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách địa danh gồm các cột: ID (idPlace), Tên địa danh (name), Hình ảnh (image)

```

@Data
public class PlaceCreateInput {
    private String idPlace;
    private String name;
    private String image;
}

@PostMapping("/place")
public Place createPlace(@RequestBody PlaceCreateInput input){
    return travelService.createPlace(input);
}

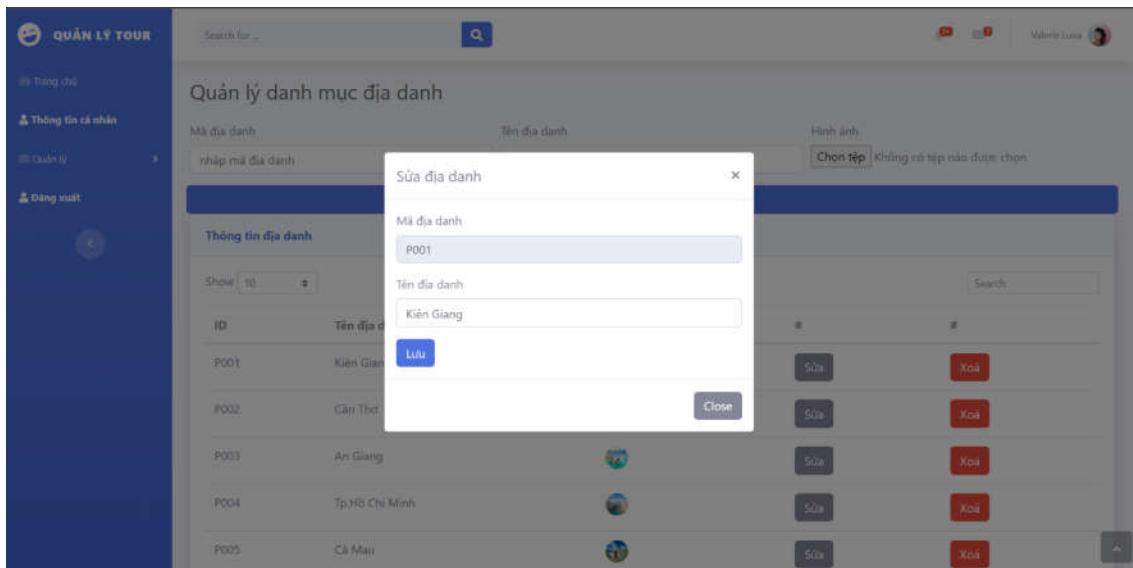
public Place createPlace(PlaceCreateInput input){
    Place checkPlace= getPlaceById(input.getIdPlace());
    if (checkPlace != null){
        return null;
    }

    Place place = new Place();

    place.setIdPlace(input.getIdPlace());
    place.setName(input.getName());
    place.setImage(input.getImage());

    return placeRepository.save(place);
}

```



Hình 5.4: Giao diện popup sửa địa danh trang web quản lý

Mô tả: (Popup sửa địa danh)

- Các ô nhập dữ liệu: mã địa danh (idPlace), tên địa danh (name) nhập vào thông tin để thực hiện chức năng sửa thông tin địa danh.
- Nút xử lý: “Lưu” dùng để cập nhật thông tin địa danh, nút “Close” dùng để thoát khỏi giao diện sửa thông tin địa danh
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API updatePlace, truyền vào (idPlace, name, image), API kiểm tra xem có tồn tại địa danh nào có mã địa danh người dùng nhập vào không, nếu không tồn tại thì không cho sửa, nếu có tồn tại thì cập nhật vào cơ sở dữ liệu, sau đó thoát khỏi giao diện sửa thông tin địa danh và hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách địa danh gồm các cột: ID (idPlace), Tên địa danh (name), Hình ảnh (image)

```

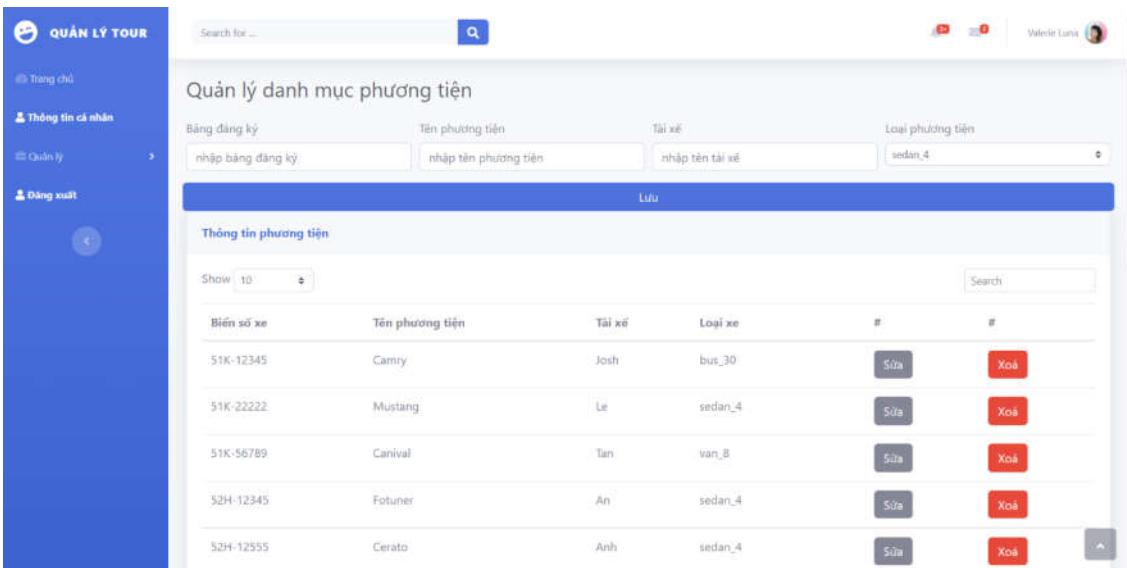
@PutMapping("/place")
public Place updatePlace(@RequestBody Place input){
    return travelService.updatePlace( input);
}

public Place updatePlace(Place input){
    Place place = getPlaceById(input.getIdPlace());

    if (place != null){
        if (input.getName() != null) place.setName(input.getName());
        if (input.getImage() != null )place.setImage(input.getImage());

        return placeRepository.save(place);
    }
    return null;
}

```



Hình 5.5: Giao diện quản lý phương tiện trang web quản lý

Mô tả: (form quản lý phương tiện)

- Các ô nhập dữ liệu: bảng đăng ký (idVehicle), tên phương tiện(name), Tài xế (driverName), Loại phương tiện (type) nhập vào thông tin để thực hiện chức năng thêm mới phương tiện.
- Bảng hiển thị: danh sách phương tiện (idVehicle, name, driverName, type)
- Nút xử lý: “Lưu” dùng để tạo mới phương tiện, nút “Sửa” dùng để hiển thị popup sửa thông tin phương tiện, nút “Xoá” dùng để thực hiện chức năng xoá phương tiện, gọi API deleteVehicle, truyền vào idVehicle.
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API createVehicle, truyền vào (idVehicle, name, driverName, type), API kiểm tra xem có tồn tại phương tiện chưa, nếu đã tồn tại thì không cho thêm mới, nếu chưa tồn tại thì lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng (Vehicle) trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách phương tiện gồm các cột: Biển số xe (idVehicle), Tên phương tiện (name), Tài xế (driverName), Loại xe (type)

```

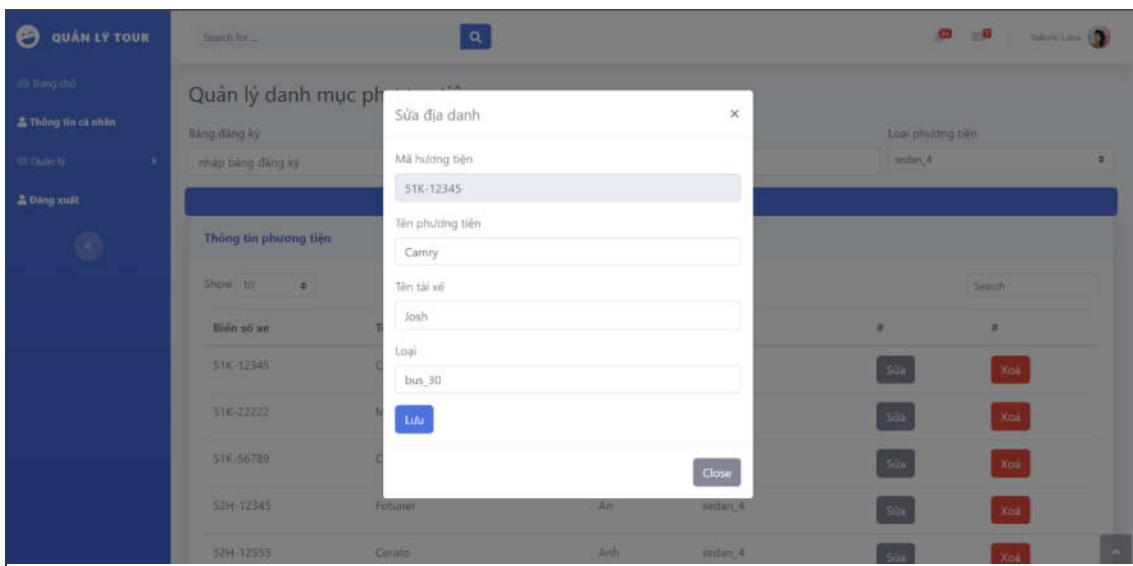
@PostMapping("/vehicle")
public Vehicle createVehicle(@RequestBody VehicleCreateInput input){
    return travelService.createVehicle(input);
}

public Vehicle createVehicle(VehicleCreateInput input){
    Vehicle checkVehicle= getVehicleById(input.getIdVehicle());
    if (checkVehicle != null){
        return null;
    }
    Vehicle vehicle = new Vehicle();

    vehicle.setIdVehicle(input.getIdVehicle());
    vehicle.setName(input.getName());
    vehicle.setDriverName(input.getDriverName());
    vehicle.setType(input.getType());

    return vehicleRepository.save(vehicle);
}

```



Hình 5.6: Giao diện popup sửa địa danh trang web quản lý

Mô tả: (Pop-up sửa phương tiện)

- Các ô nhập dữ liệu: Mã phương tiện (idVehicle), tên phương tiện(name), Tên tài xế (driverName), Loại (type) nhập vào thông tin để thực hiện chức năng sửa thông tin phương tiện.
- Nút xử lý: “Lưu” dùng để cập nhật thông tin phương tiện, nút “Close” dùng để thoát khỏi giao diện sửa thông tin phương tiện
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API updateVehicle, truyền vào (idVehicle, name, driverName, type), API kiểm tra xem có tồn tại phương tiện đó chưa, nếu chưa thì không cho phép cập nhật, nếu có tồn tại thì cập nhật vào cơ sở dữ liệu, sau đó thoát khỏi giao diện sửa thông tin phương tiện và hệ thống gửi dữ liệu từ bảng (Vehicle) trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách phương tiện gồm các cột: Biển số xe (idVehicle), Tên phương tiện (name), Tài xế (driverName), Loại xe (type)

```

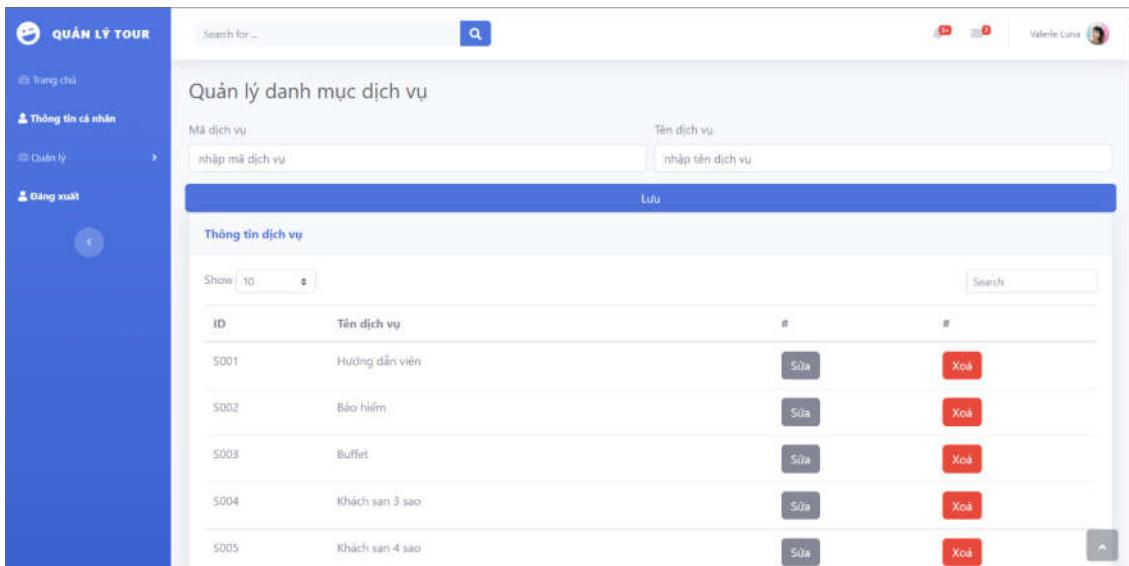
@PutMapping("/vehicle")
public Vehicle updateVehicle(@RequestBody Vehicle input){
    return travelService.updateVehicle( input);
}

public Vehicle updateVehicle(Vehicle input){
    Vehicle vehicle = getVehicleById(input.getIdVehicle());

    if (vehicle != null){
        if (input.getName() != null) vehicle.setName(input.getName());
        if (input.getDriverName() != null) vehicle.setDriverName(input.getDriverName());
        if (input.getType() != null) vehicle.setType(input.getType());

        return vehicleRepository.save(vehicle);
    }
    return null;
}

```



Hình 5.7: Giao diện quản lý dịch vụ trang web quản lý

Mô tả: (form quản lý dịch vụ)

- Các ô nhập dữ liệu: mã dịch vụ (idService), tên dịch vụ (name) nhập vào thông tin để thực hiện chức năng thêm mới dịch vụ.
- Bảng hiển thị: danh sách dịch vụ (idService, name)
- Nút xử lý: “Lưu” dùng để tạo mới dịch vụ, nút “Sửa” dùng để hiển thị popup sửa thông tin dịch vụ, nút “Xoá” dùng để thực hiện chức năng xoá dịch vụ
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API createService, truyền vào (idService, name), API tiến hành kiểm tra xem có tồn tại dịch vụ này chưa, nếu có rồi thì không cho phép thêm, nếu chưa thì lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng (Service) trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách dịch vụ gồm các cột: ID (idService), Tên dịch vụ (name)

```

@GetMapping("/service")
public List<Service> getAllService(){
    return travelService.getAllService();
}

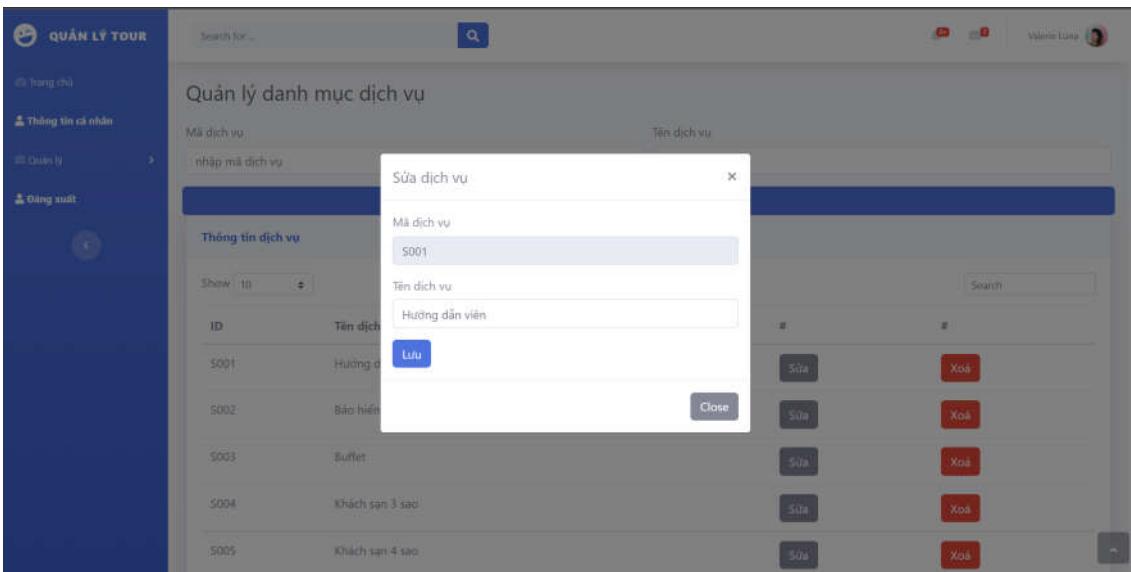
@PostMapping("/service")
public Service createService(@RequestBody ServiceCreateInput input){
    return travelService.createService(input);
}

public com.example.demo.model.Service createService(ServiceCreateInput input){
    com.example.demo.model.Service checkService= getServiceById(input.getIdService());
    if (checkService != null){
        return null;
    }
    com.example.demo.model.Service service = new com.example.demo.model.Service();

    service.setIdService(input.getIdService());
    service.setName(input.getName());

    return serviceRepository.save(service);
}

```



Hình 5.8: Giao diện popup sửa dịch vụ trang web quản lý

Mô tả: (Popup sửa dịch vụ)

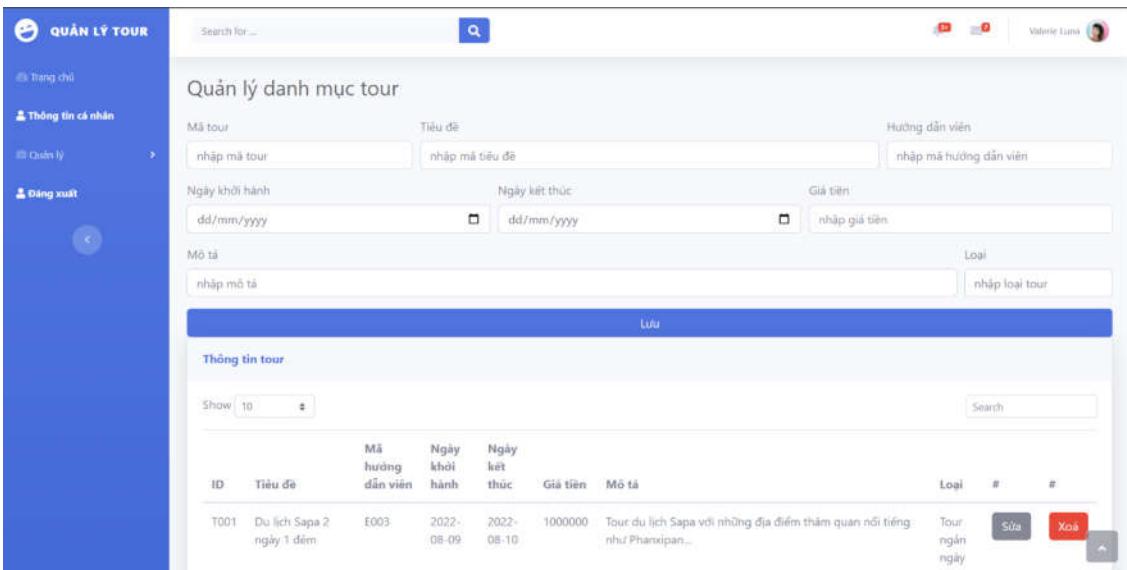
- Các ô nhập dữ liệu: Mã dịch vụ (idService), tên dịch vụ (name) nhập vào thông tin để thực hiện chức năng sửa thông tin dịch vụ.
- Nút xử lý: “Lưu” dùng để cập nhật thông tin dịch vụ, nút “Close” dùng để thoát khỏi giao diện sửa thông tin dịch vụ
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API updateService, truyền vào (idService, name), API kiểm tra xem có tồn tại dịch vụ chưa, nếu chưa thì không cho cập nhật, nếu có thì cập nhật vào cơ sở dữ liệu, sau đó thoát khỏi giao diện sửa thông tin dịch vụ và hệ thống gửi dữ liệu từ bảng (Service) trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách dịch vụ gồm các cột: Mã dịch vụ (idService), tên dịch vụ (name)

```
@PutMapping("/service")
public Service updateService( @RequestBody ServiceCreateInput input){
    return travelService.updateService( input);
}
```

```
public com.example.demo.model.Service updateService( ServiceCreateInput input){
    com.example.demo.model.Service service = getServiceById(input.getIdService());

    if (service != null){
        if (input.getName() != null) service.setName(input.getName());

        return serviceRepository.save(service);
    }
    return null;
}
```



Hình 5.9: Giao diện quản lý tour trang web quản lý

Mô tả: (form quản lý tour)

- Các ô nhập dữ liệu: mã tour (idTour), tiêu đề (title), hướng dẫn viên (idEmployee), ngày khởi hành (dateStart), ngày kết thúc (dateEnd), giá tiền (price), mô tả (description), loại (type) nhập vào thông tin để thực hiện chức năng thêm mới tour.
- Bảng hiển thị: danh sách tour (idTour, title, idEmployee, dateStart, dateEnd, price, description, type)
- Nút xử lý: “Lưu” dùng để tạo mới tour, nút “Sửa” dùng để hiển thị popup sửa thông tin tour nút “Xoá” dùng để thực hiện chức năng xoá tour
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API createTour, truyền vào (idTour, title, idEmployee, dateStart, dateEnd, price, description, type), API kiểm tra xem có tồn tại tour chưa, nếu tồn tại thì không cho thêm, nếu chưa thì lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng (Tour) trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách tour gồm các cột: ID (idTour), tiêu đề (title), hướng dẫn viên (idEmployee), ngày khởi hành (dateStart), ngày kết thúc (dateEnd), giá tiền (price), mô tả (description), loại (type)

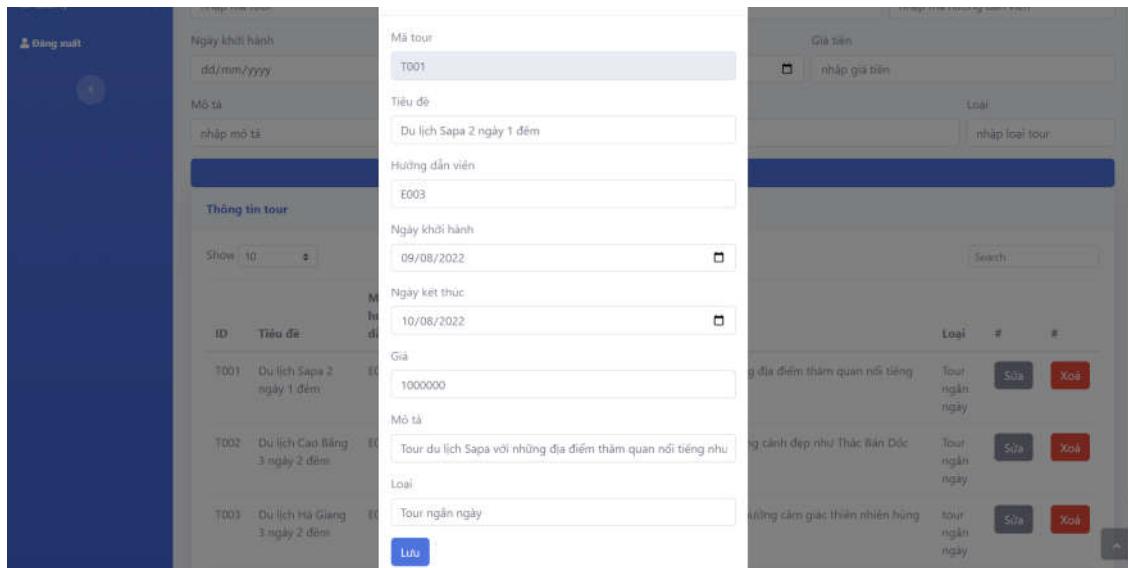
```

@GetMapping("/gettour")
public List<Tour> getAllTour(){
    return travelService.getAllTour();
}

@PostMapping("/tour")
public Tour createTour(@RequestBody Tour input){
    return travelService.createTour(input);
}

public Tour createTour(Tour input){
    Tour checkTour= getTourById(input.getIdTour());
    if (checkTour != null){
        return null;
    }
    Tour tour= new Tour();
    tour.setIdTour(input.getIdTour());
    tour.setTitle(input.getTitle());
    tour.setIdEmployee(input.getIdEmployee());
    tour.setDateStart(input.getDateStart());
    tour.setDateEnd(input.getDateEnd());
    tour.setPrice(input.getPrice());
    tour.setDescription(input.getDescription());
    tour.setType(input.getType());
    tourRepository.save(tour);
    return getTourById(tour.getIdTour());
}

```



Hình 5.10: Giao diện popup sửa tour trang web quản lý

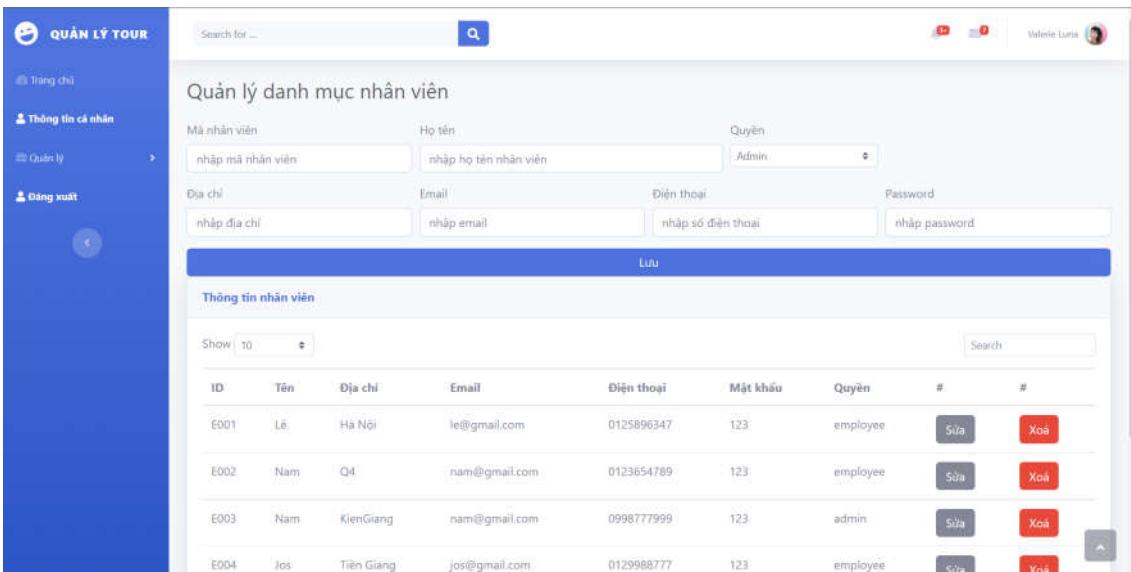
## Mô tả: (Popup sửa tour)

- Các ô nhập dữ liệu: mã tour (idTour), tiêu đề (title), hướng dẫn viên (idEmployee), ngày khởi hành (dateStart), ngày kết thúc (dateEnd), giá tiền (price), mô tả (description), loại (type) nhập vào thông tin để thực hiện chức năng sửa thông tin tour.
- Nút xử lý: “Lưu” dùng để cập nhật thông tin tour, nút “Close” dùng để thoát khỏi giao diện sửa thông tin tour
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API updateTour, truyền vào (idTour, title, idEmployee, dateStart, dateEnd, description, type), API kiểm tra xem có tồn tại tour không, nếu không thì không cho cập nhật, nếu có thì cập nhật vào cơ sở dữ liệu, sau đó thoát khỏi giao diện sửa thông tin tour và hệ thống gửi dữ liệu từ bảng (Tour) trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách tour gồm các cột: ID (idTour), tiêu đề (title), hướng dẫn viên (idEmployee), ngày khởi hành (dateStart), ngày kết thúc (dateEnd), giá tiền (price), mô tả (description), loại (type)

```
@PutMapping("/tour")
public Tour updateTour(@RequestBody Tour input){
    return travelService.updateTour( input);
}
```

```
public Tour updateTour( Tour input){
    Tour tour= getTourById(input.getIdTour());
    if (tour != null){
        if (input.getTitle() != null) tour.setTitle(input.getTitle());
        if (input.getIdEmployee() != null) tour.setIdEmployee(input.getIdEmployee());
        if (input.getDateStart() != null) tour.setDateStart(input.getDateStart());
        if (input.getDateEnd() != null) tour.setDateEnd(input.getDateEnd());
        if (input.getPrice() != null) tour.setPrice(input.getPrice());
        if (input.getDescription() != null) tour.setDescription(input.getDescription());
        if (input.getType() != null) tour.setType(input.getType());

        return tourRepository.save(tour);
    }
    return null;
}
```



Hình 5.11: Giao diện quản lý nhân viên trang web quản lý

Mô tả: (form quản lý nhân viên)

- Các ô nhập dữ liệu: mã nhân viên (idEmployee), tên nhân viên (name), quyền (role), địa chỉ (address), email (email), điện thoại (phone), password (password) nhập vào thông tin để thực hiện chức năng thêm mới nhân viên.
- Bảng hiển thị: danh sách nhân viên (idEmployee, name, address, email, phone, password, role)
- Nút xử lý: “Lưu” dùng để tạo mới nhân viên, nút “Sửa” dùng để hiển thị popup sửa thông tin nhân viên, nút “Xoá” dùng để thực hiện chức năng xoá nhân viên, gọi API deleteEmployee.
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API createEmployee lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách nhân viên gồm các cột: ID (idEmployee), tên nhân viên (name), quyền (role), địa chỉ (address), email (email), điện thoại (phone), password (password)

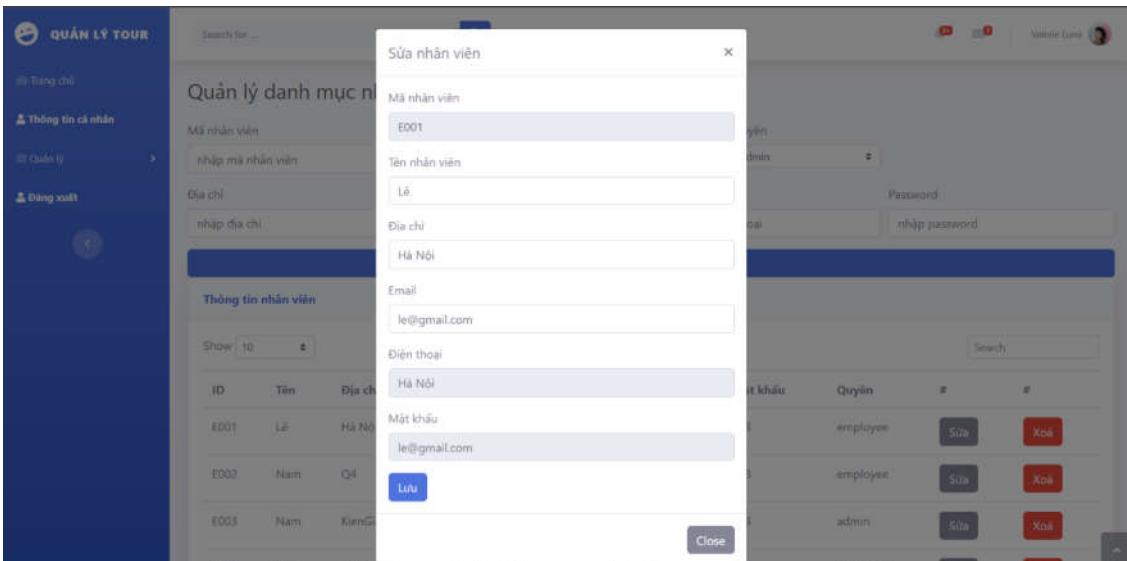
```
//Employee
@GetMapping("/employee")
public List<Employee> getAllEmployee() { return travelService.getAllEmployee(); }

@PostMapping("/employee")
public Employee createEmployee(@RequestBody EmployeeCreateInput input){
    return travelService.createEmployee(input);
}

public Employee createEmployee(EmployeeCreateInput input){
    Employee checkEmployee= getEmployeeById(input.getIdEmployee());
    if (checkEmployee != null){
        return null;
    }
    if (employeeRepository.getEmployeeByPhone(input.getPhone()) != null){
        return null;
    }
    Employee employee = new Employee();

    employee.setIdEmployee(input.getIdEmployee());
    employee.setName(input.getName());
    employee.setAddress(input.getAddress());
    employee.setEmail(input.getEmail());
    employee.setPhone(input.getPhone());
    employee.setPassword(input.getPassword());
    employee.setRole(input.getRole());

    return employeeRepository.save(employee);
}
```



Hình 5.12: Giao diện popup sửa nhân viên trang web quản lý

Mô tả: (Popup sửa nhân viên)

- Các ô nhập dữ liệu: mã nhân viên (idEmployee), tên nhân viên (name), quyền (role), địa chỉ (address), email (email), điện thoại (phone), password (password) nhập vào thông tin để thực hiện chức năng sửa thông tin nhân viên.
- Nút xử lý: “Lưu” dùng để cập nhật thông tin nhân viên, nút “Close” dùng để thoát khỏi giao diện sửa thông tin nhân viên
- Xử lý: khi nhấn nút “Lưu”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API updateEmployee cập nhật vào cơ sở dữ liệu, sau đó thoát khỏi giao diện sửa thông tin nhân viên và hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách nhân viên gồm các cột: ID (idEmployee), tên nhân viên (name), quyền (role), địa chỉ (address), email (email), điện thoại (phone), password (password)

```
@PutMapping("/employee")
public Employee updateEmployee(@RequestBody EmployeeUpdateInput input){
    return travelService.updateEmployee( input);
}

public Employee updateEmployee(EmployeeUpdateInput input){
    Employee employee = getEmployeeById(input.getIdEmployee());

    if (employee != null){
        if (input.getName() != null) employee.setName(input.getName());
        if (input.getAddress() != null )employee.setAddress(input.getAddress());
        if (input.getEmail() != null) employee.setEmail(input.getEmail());

        return employeeRepository.save(employee);
    }
    return null;
}
```

ID	Tên	Giới tính	Email	Điện thoại	Mật khẩu	Trạng thái	#
1	Jos	Male	jos@gmail.com	0123456789	123	Inactive	<button>Khoá tài khoản</button>
2	Nam	Female	nam123@gmail.com	0963963963	123	Inactive	<button>Khoá tài khoản</button>
3	Tú	Male	tu@gmail.com	0123456999	123	Active	<button>Khoá tài khoản</button>
4	Tuân	Male	tuan@gmail.com	0123963559	123	Active	<button>Khoá tài khoản</button>
5	Linh	Male	tuan@gmail.com	0189999888	123	Active	<button>Khoá tài khoản</button>
6	test	nam	test@gmail.com	0987654321	123	Inactive	<button>Khoá tài khoản</button>
7	Qué	male	que@gmail.com	0999333222	123	Inactive	<button>Khoá tài khoản</button>

Hình 5.13: Giao diện quản lý khách hàng trang web quản lý

Mô tả: (form quản lý khách hàng)

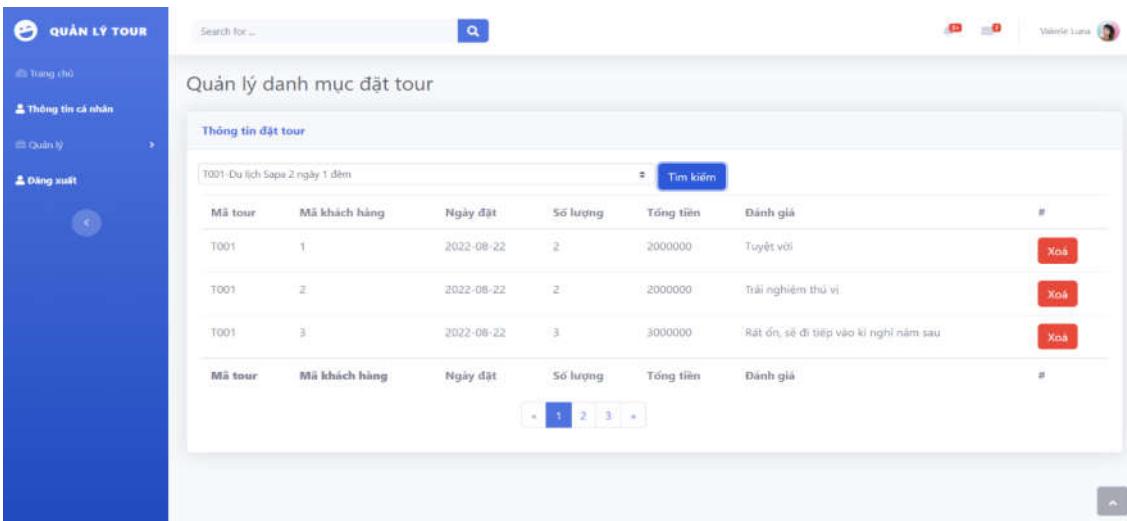
- Nút xử lý: nút “Khoá tài khoản” dùng để thực hiện chức năng khoá tài khoản khách hàng
- Bảng hiển thị: danh sách khách hàng (idCustomer, name, gender, email, phone, password, status)
- Xử lý: khi nhấn nút “Khoá tài khoản” giao diện sẽ lấy dữ liệu (idCustomer) từ dòng trong bảng gửi đến hệ thống, hệ thống gọi API blockCustomer cập nhật cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách nhân viên gồm các cột: ID (idCustomer), tên khách hàng (name), giới tính (gender), email (email), điện thoại (phone), password (password)

```
@GetMapping("/customer")
public List<Customer> getAllCustomer(){
    return travelService.getAllCustomer();
}
```

```
@PutMapping("/customer/block/{idCustomer}")
public Customer blockCustomer(@PathVariable(name = "idCustomer") int idCustomer){
    return travelService.blockCustomer(idCustomer);
}
```

```
public Customer blockCustomer(int idCustomer){
    Customer customer= getCustomerById(idCustomer);

    if (customer != null){
        customer.setStatus(ActiveStatus.Inactive);
        return customerRepository.save(customer);
    }
    return null;
}
```



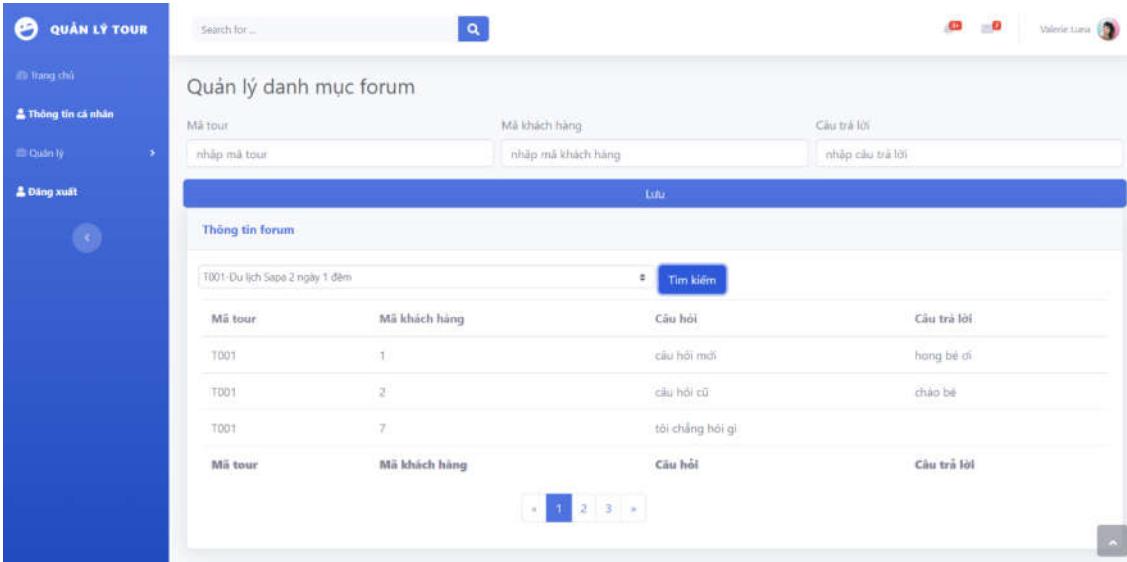
Hình 5.14: Giao diện quản lý đặt tour trang web quản lý

Mô tả: (form quản lý đặt tour)

- Các ô nhập dữ liệu: combobox chọn tour để hiển thị danh sách khách hàng của tour đó.
- Bảng hiển thị: danh sách đặt tour (idTour, idCustomer, dateBooking, quantityPeople, totalPrice, judge)
- Nút xử lý: “Tìm kiếm” dùng để hiển thị bảng danh sách đặt tour, nút “Xoá” dùng để xoá thông tin đặt tour.
- Xử lý: khi nhấn nút “Tìm kiếm”, giao diện sẽ lấy dữ liệu từ ô nhập dữ liệu (idTour) gửi đến hệ thống, gọi API getBookingByIdTour sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách đặt tour gồm các cột: mã tour (idTour), mã khách hàng (idCustomer), ngày đặt (dateBooking), số lượng (quantityPeople), tổng tiền (totalPrice), đánh giá (judge).

```
@GetMapping("/booking/tour/{idTour}")
public List<Booking> getBookingByIdTour(@PathVariable(name = "idTour") String idTour){
    return travelService.getBookingByIdTour(idTour);
}
```

```
@DeleteMapping("/booking/{idCustomer}/{idTour}")
public String deleteBooking(@PathVariable(name = "idCustomer") int idCustomer, @PathVariable(name = "idTour") String idTour){
    CustomerTourKey id= new CustomerTourKey();
    id.setIdTour(idTour);
    id.setIdCustomer(idCustomer);
    return travelService.deleteBooking(id);
}
```



Hình 5.15: Giao diện quản lý forum trang web quản lý

Mô tả: (form quản lý danh mục forum)

- Các ô nhập dữ liệu: mã tour (idTour), mã khách hàng (idCustomer), câu trả lời (question) nhập vào thông tin để cập nhật câu trả lời, combobox chọn tour để hiển thị danh sách câu hỏi của tour đó.
- Bảng hiển thị: danh sách câu hỏi (idTour, idCustomer, question, answer)
- Nút xử lý: “Tìm kiếm” dùng để hiển thị bảng danh sách câu hỏi, nút “Lưu” dùng để cập nhật câu trả lời trên forum
- Xử lý: khi nhấn nút “Tìm kiếm”, giao diện sẽ lấy dữ liệu từ ô nhập dữ liệu (idTour) gửi đến hệ thống, gọi API getAskByIdTour sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách câu hỏi gồm các cột: mã tour (idTour), mã khách hàng (idCustomer), câu hỏi (question), câu trả lời (answer). Khi nhấn nút “Lưu” giao diện sẽ lấy dữ liệu từ ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API updateAnswer lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách câu hỏi gồm các cột: mã tour (idTour), mã khách hàng (idCustomer), câu hỏi (question), câu trả lời (answer)

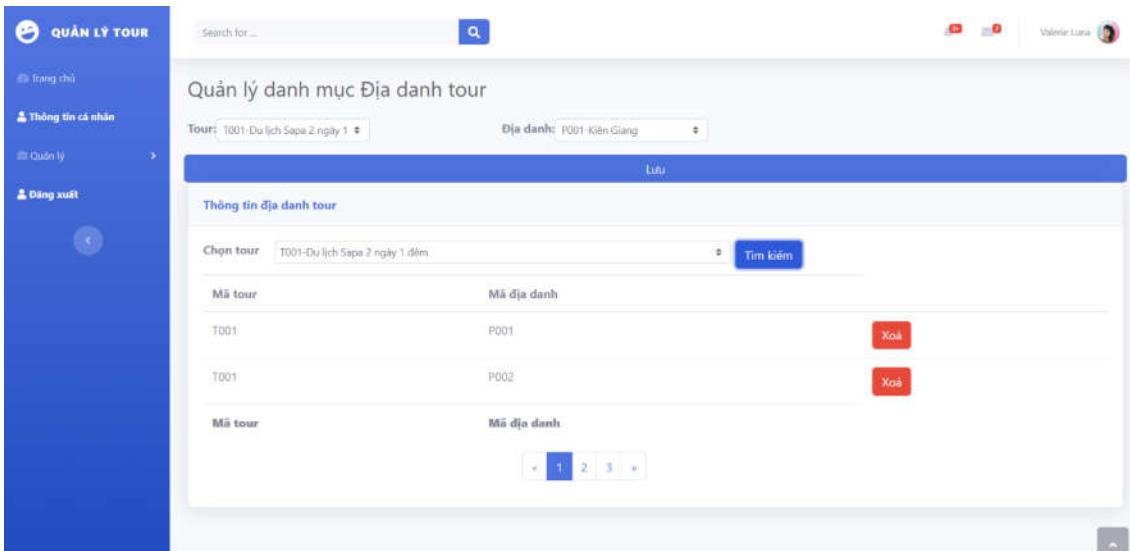
```

public List<Ask> getAskByIdTour(String idTour){
    return askRepository.getAskByIdTour(idTour);
}

public Ask updateAnswer(AnswerInput input){
    Ask ask= askRepository.getAskByIdTourAndIdCustomer(input.getIdTour(), input.getIdCustomer());

    if (ask != null){
        if (input.getAnswer() != null){
            ask.setAnswer(input.getAnswer());
            askRepository.save(ask);
        }
    }
    return null;
}

```



Hình 5.16: Giao diện quản lý địa danh-tour trang web quản lý

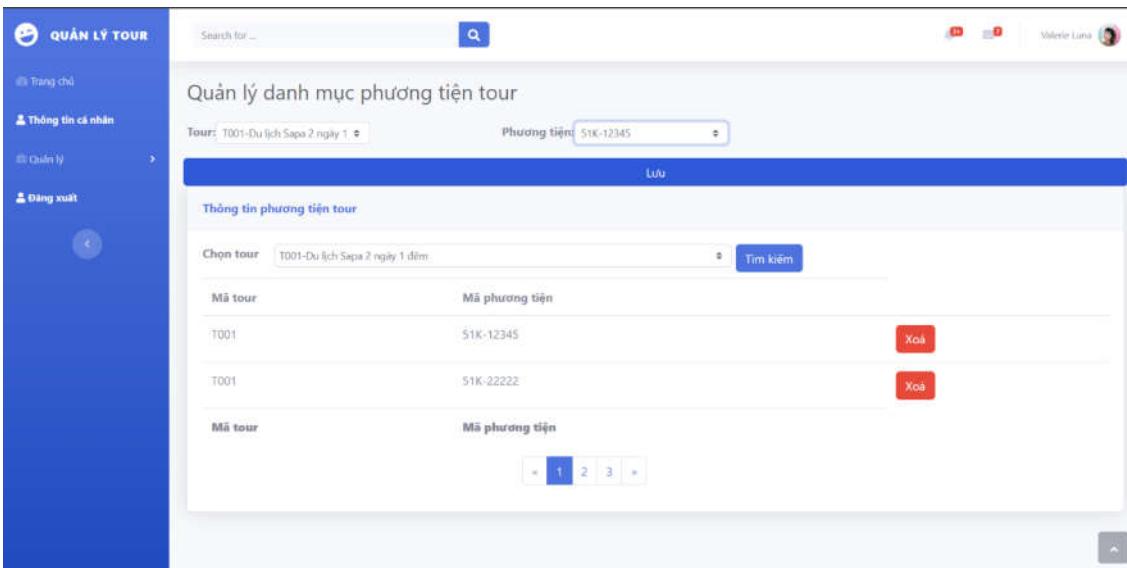
Mô tả: (form quản lý danh mục địa danh – tour)

- Các ô nhập dữ liệu: combobox tour để lấy mã tour, combobox địa danh để lấy mã địa danh combobox chọn tour để hiển thị danh sách câu hỏi của tour đó.
- Bảng hiển thị: danh sách địa danh-tour (idTour, idPlace)
- Nút xử lý: “Tìm kiếm” dùng để hiển thị bảng danh sách địa danh, nút “Lưu” dùng để tạo mới địa danh-tour, nút “Xoá” dùng để xoá địa danh-tour, gọi API deletePlaceTour
- Xử lý: khi nhấn nút “Tìm kiếm”, giao diện sẽ lấy dữ liệu từ ô nhập dữ liệu (idTour) gửi đến hệ thống, gọi API getPlaceTourByIdTour sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách địa danh-tour gồm các cột: mã tour (idTour), mã địa danh (idPlace). Khi nhấn nút “Lưu” giao diện sẽ lấy dữ liệu từ ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API createPlaceTour lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách địa danh-tour gồm các cột: mã tour (idTour), mã địa danh (idPlace).

```
@GetMapping("/placetour/{idTour}")
public List<PlaceTour> getPlaceTourByIdTour(@PathVariable(name = "idTour") String idTour){
    return travelService.getPlaceTourByIdTour(idTour);
}
```

```
@PostMapping("/placetour")
public PlaceTour createPlaceTour(@RequestBody PlaceTour input){
    return travelService.createPlaceTour(input);
}
```

```
@DeleteMapping("/placetour/{idTour}/{idPlace}")
public String deletePlaceTour(@PathVariable(name = "idTour") String idTour, @PathVariable(name = "idPlace") String idPlace){
    PlaceTourKey key= new PlaceTourKey();
    key.setIdTour(idTour);
    key.setIdPlace(idPlace);
    return travelService.deletePlaceTour(key);
}
```



Hình 5.17: Giao diện quản lý phương tiện-tour trang web quản lý

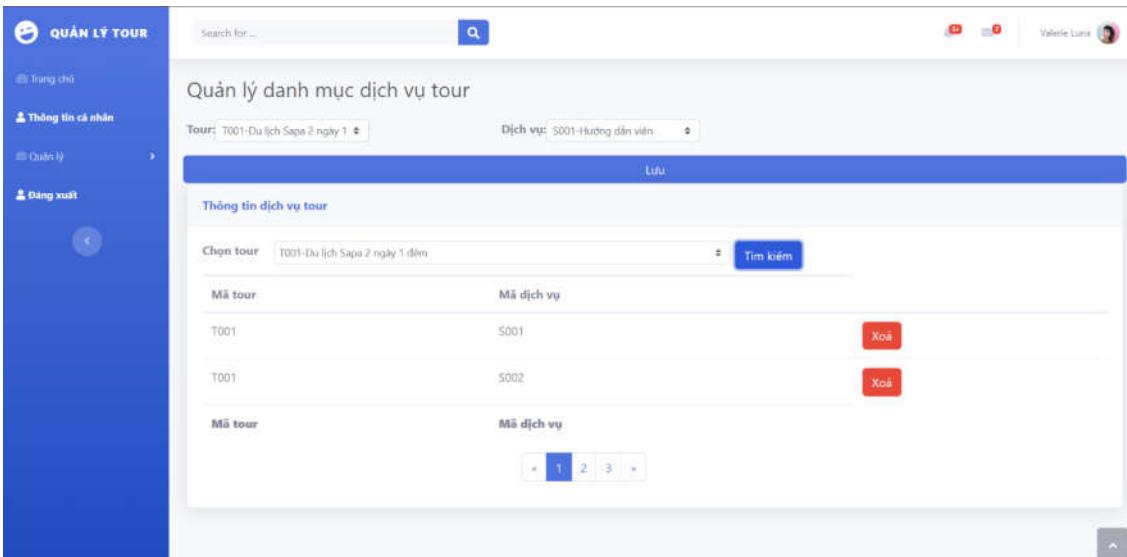
Mô tả: (form quản lý danh mục phương tiện – tour)

- Các ô nhập dữ liệu: combobox tour để lấy mã tour, combobox phương tiện để lấy mã phương tiện, combobox chọn tour để hiển thị danh sách phương tiện của tour đó.
- Bảng hiển thị: danh sách phương tiện-tour (idTour, idVehicle)
- Nút xử lý: “Tìm kiếm” dùng để hiển thị bảng danh sách phương tiện-tour, nút “Lưu” dùng để tạo mới phương tiện-tour, nút “Xoá” dùng để xoá phương tiện-tour, gọi API deleteVehicleTour
- Xử lý: khi nhấn nút “Tim kiem”, giao diện sẽ lấy dữ liệu từ ô nhập dữ liệu (idTour) gửi đến hệ thống, gọi API getVehicleTourByIdTour sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách phương tiện-tour gồm các cột: mã tour (idTour), mã phương tiện (idVehicle). Khi nhấn nút “Lưu” giao diện sẽ lấy dữ liệu từ ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API createVehicleTour lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách phương tiện-tour gồm các cột: mã tour (idTour), mã phương tiện (idVehicle).

```
@GetMapping("/vehicletour/{idTour}")
public List<VehicleTour> getVehicleTourByIdTour(@PathVariable(name = "idTour") String idTour){
    return travelService.getVehicleTourByIdTour(idTour);
}
```

```
@PostMapping("/vehicletour")
public VehicleTour createVehicleTour(@RequestBody VehicleTour input){
    return travelService.createVehicleTour(input);
}

@DeleteMapping("/vehicletour/{idTour}/{idVehicle}")
public String deleteVehicleTour(@PathVariable(name = "idTour") String idTour, @PathVariable(name = "idVehicle") String idVehicle){
    VehicleTourKey key= new VehicleTourKey();
    key.setIdTour(idTour);
    key.setIdVehicle(idVehicle);
    return travelService.deleteVehicleTour(key);
}
```



Hình 5.18: Giao diện quản lý dịch vụ-tour trang web quản lý

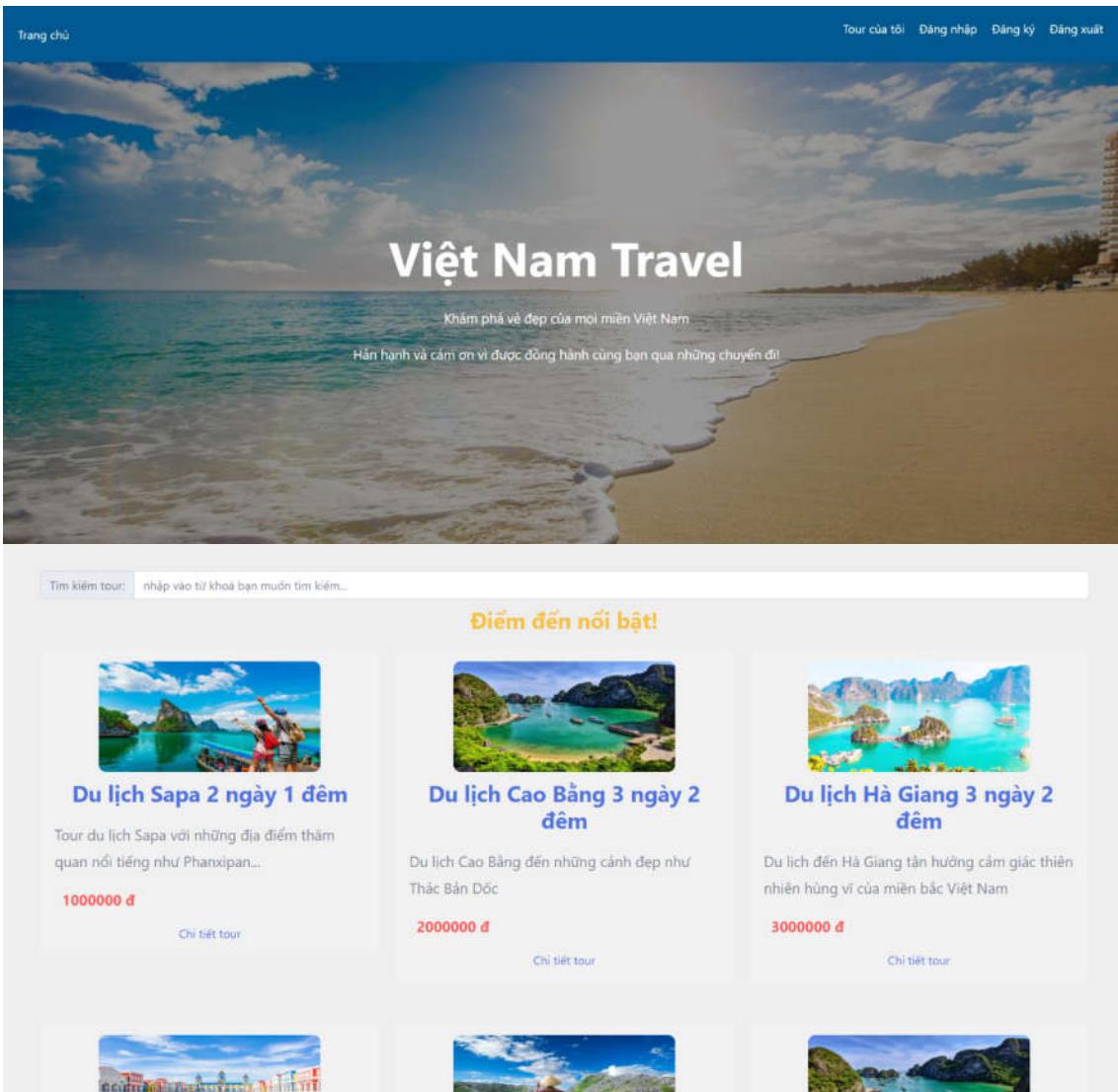
Mô tả: (form quản lý danh mục dịch vụ – tour)

- Các ô nhập dữ liệu: combobox tour để lấy mã tour, combobox dịch vụ để lấy mã dịch vụ, combobox chọn tour để hiển thị danh sách dịch vụ của tour đó.
- Bảng hiển thị: danh sách dịch vụ-tour (idTour, idService)
- Nút xử lý: “Tìm kiếm” dùng để hiển thị bảng danh sách dịch vụ-tour, nút “Lưu” dùng để tạo mới dịch vụ-tour, nút “Xoá” dùng để xoá dịch vụ-tour, gọi API deleteServiceTour
- Xử lý: khi nhấn nút “Tìm kiếm”, giao diện sẽ lấy dữ liệu từ ô nhập dữ liệu (idTour) gửi đến hệ thống, gọi API getServiceTourByIdTour sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách dịch vụ-tour gồm các cột: mã tour (idTour), mã dịch vụ (idService). Khi nhấn nút “Lưu” giao diện sẽ lấy dữ liệu từ ô nhập dữ liệu gửi đến hệ thống, hệ thống gọi API createServiceTour lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách dịch vụ-tour gồm các cột: mã tour (idTour), mã dịch vụ (idService)

```
@GetMapping("/servicetour/{idTour}")
public List<ServiceTour> getServiceTourByIdTour(@PathVariable(name = "idTour") String idTour){
    return travelService.getServiceTourByIdTour(idTour);
}
```

```
@PostMapping("/servicetour")
public ServiceTour createServiceTour(@RequestBody ServiceTour input) throws Exception {
    ServiceTour check = travelService.createServiceTour(input);
    if (check == null){
        throw new Exception("lỗi");
    }
    return check;
}

@DeleteMapping("/servicetour/{idTour}/{idService}")
public String deleteServiceTour(@PathVariable(name = "idTour") String idTour, @PathVariable(name = "idService") String idService){
    ServiceTourKey key= new ServiceTourKey();
    key.setIdTour(idTour);
    key.setIdService(idService);
    return travelService.deleteServiceTour(key);
}
```



Mô tả: (trang chủ)

- Các ô nhập dữ liệu: nhập vào từ khoá để tìm kiếm tour.
- Nút xử lý: “Chi tiết tour” dùng để chuyển qua giao diện chi tiết tour, “Tour của tôi” dùng để xem thông tin đặt tour của người dùng, “Đăng nhập” dùng để chuyển sang giao diện đăng nhập, “Đăng ký” dùng để chuyển sang giao diện đăng ký, “Đăng xuất” đăng xuất tài khoản.
- Xử lý: khi nhấn nút “Chi tiết tour”, giao diện sẽ lấy dữ liệu từ thẻ hiển thị (idTour) gửi đến hệ thống, chuyển sang giao diện chi tiết tour, sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị chi tiết tour gồm các thuộc tính tiêu đề tour (title), mô tả tour (description), ngày khởi hành (dateStart), ngày kết thúc (dateEnd), dịch vụ (services), điểm đến (places), phương tiện (vehicles), giá (price), thảo luận (ask), đánh giá (booking). Khi nhấn nút “Chi tiết tour” giao diện sẽ lấy dữ liệu từ thẻ hiển thị (idTour) gửi đến hệ thống, sau đó chuyển sang giao diện chi tiết tour.

**Du lịch Sapa 2 ngày 1 đêm**

Tour du lịch Sapa với những địa điểm thăm quan nổi tiếng như Phansipan...

Ngày khởi hành: 2022-08-09

Ngày kết thúc: 2022-08-10

Dịch vụ:

- Hướng dẫn viên
- Bảo hiểm

Điểm đến:

- Kiên Giang
- Cần Thơ

Phương tiện:

- Camry-51K-12345
- Mustang-51K-22222

Giá : 1000000 đ

Chọn số lượng:

0

**Đặt tour**

**Đánh giá:**

Khách hàng 1 :Tuyệt vời

Khách hàng 2 :Trải nghiệm thú vị

Khách hàng 3 :Rất ấn, sẽ di chuyển vào kì nghỉ năm sau

Nhập câu hỏi:

**Đặt câu hỏi**

**Thảo luận**

Khách hàng 1 :câu hỏi mới  
hóng bé ơi

Khách hàng 2 :câu hỏi cũ  
chào bé

Khách hàng 7 :tôi chẳng hỏi gì

Trang chủ

Suggestion Box

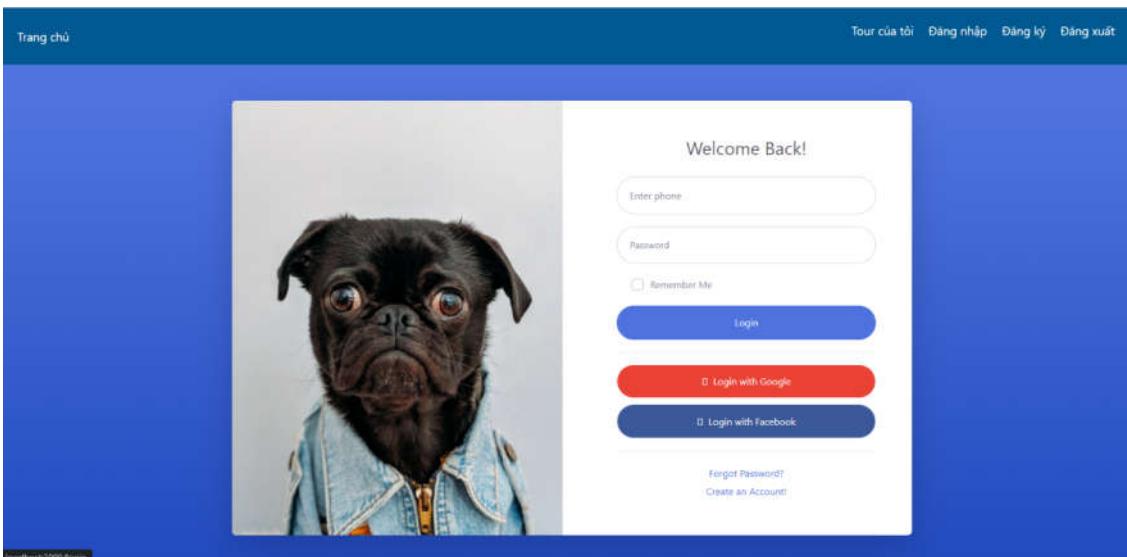
Privacy Policy

Giveaway

Hình 5.20: Trang chi tiết tour trang web đặt tour

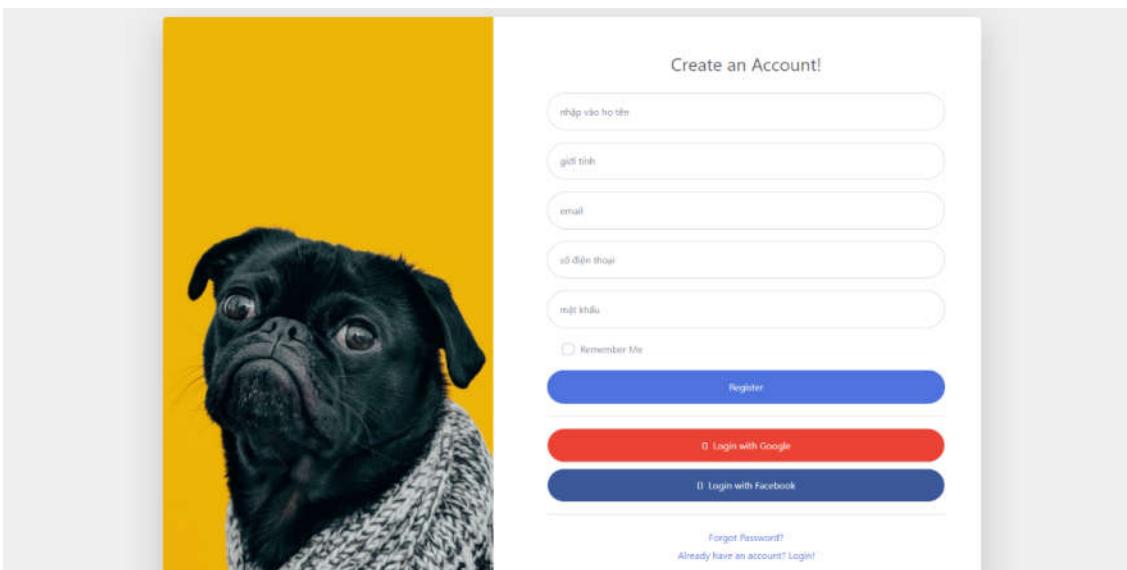
Mô tả: (trang chi tiết tour)

- Các ô nhập dữ liệu: “số lượng” nhập vào số lượng để đặt tour
- Nút xử lý: “Đặt tour” dùng để thực hiện chức năng đặt tour, nút “đặt câu hỏi” để thực hiện chức năng đặt câu hỏi trên forum.
- Xử lý: khi nhấn nút “Đặt tour” giao diện sẽ lấy dữ liệu từ các ô nhập để gửi lên hệ thống, hệ thống tiến hành lưu vào cơ sở dữ liệu, sau đó chuyển hướng về trang chủ. Khi nhấn nút “đặt câu hỏi” giao diện sẽ lấy dữ liệu từ các ô nhập để gửi lên hệ thống, hệ thống tiến hành lưu vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng lên giao diện để hiển thị câu hỏi ở trang chi tiết tour.



Mô tả: (form đăng nhập)

- Các ô nhập dữ liệu: số điện thoại (phone), mật khẩu (password)
- Nút xử lý: Login
- Xử lý: khi nhấn nút “Login” giao diện lấy dữ liệu từ các ô nhập dữ liệu gửi lên hệ thống, hệ thống tiến hành kiểm tra dữ liệu nhập vào với dữ liệu trong cơ sở dữ liệu, nếu thông tin đăng nhập chính xác thì chuyển sang giao diện trang thông tin cá nhân.



Mô tả: (form đăng ký)

- Các ô nhập dữ liệu: họ tên (name), giới tính (gender), email (email), số điện thoại (phone), mật khẩu (password)
- Nút xử lý: Register
- Xử lý: khi nhấn nút “Register” giao diện lấy dữ liệu từ các ô nhập dữ liệu gửi lên hệ thống, hệ thống tiến hành lưu vào cơ sở dữ liệu.

The screenshot shows a web-based application interface. At the top, there is a blue header bar with the text "Trang chủ" on the left and navigation links "Tour của tôi", "Đăng nhập", "Đăng ký", and "Đăng xuất" on the right. Below the header, the main content area has a title "Thông tin đặt tour". It contains two input fields: "Mã tour" (with placeholder "nhập mã tour") and "Đánh giá" (with placeholder "nhập câu đánh giá"). Below these fields is a blue button labeled "Đánh giá". Underneath the button is a table titled "Đánh giá" with the following columns: Mã tour, Mã khách hàng, Ngày đặt, Số lượng, Tổng tiền, and Đánh giá. The table contains five rows of data. Each row includes a red "Huỷ tour" button in the last column. The rows are as follows:

Mã tour	Mã khách hàng	Ngày đặt	Số lượng	Tổng tiền	Đánh giá	#
T001	1	2022-08-22	2	2000000	Tuyệt vời	<button>Huỷ tour</button>
T002	1	2022-08-23	3	6000000	Vượt ngoài sự mong đợi	<button>Huỷ tour</button>
T005	1	2022-08-23	1	1500000	chưa đánh giá	<button>Huỷ tour</button>
T008	1	2022-08-23	1	1500000	chưa đánh giá	<button>Huỷ tour</button>
T010	1	2022-08-23	4	5200000	chưa đánh giá	<button>Huỷ tour</button>

Hình 5.23: Giao diện thông tin đặt tour trang web đặt tour

Mô tả: (trang thông tin đặt tour)

- Các ô nhập dữ liệu: mã tour (idTour), đánh giá (judge) nhập vào thông tin để thực hiện chức năng đánh giá tour.
- Bảng hiển thị: danh sách đặt tour (idTour, idCustomer, dateBooking, quantityPeople, totalPrice, judge)
- Nút xử lý: “Đánh giá” dùng để cập nhật đánh giá tour, nút “Huỷ tour” dùng để thực hiện chức năng huỷ tour
- Xử lý: khi nhấn nút “Đánh giá”, giao diện sẽ lấy dữ liệu từ các ô nhập dữ liệu gửi đến hệ thống, hệ thống tiến hành cập nhật vào cơ sở dữ liệu, sau đó hệ thống gửi dữ liệu từ bảng trong cơ sở dữ liệu lên giao diện để hiển thị bảng danh sách đặt tour gồm các cột: mã tour (idTour), Mã khách hàng (iCustomer), ngày đặt (dateBooking), số lượng (quantityPeople), tổng tiền (totalPrice), đánh giá (judge).

## CHƯƠNG 6: KẾT LUẬN

### 6.1 Kết luận

#### 6.1.1 Kết quả đạt được

- Tìm hiểu về Reactjs, Spring Boot, REST API để xây dựng trang web.
- Áp dụng xây dựng trang web quản lý đặt tour.
- Xây dựng được giao diện web site cho người quản trị và người dùng.
- Thực hiện được đa số các chức năng quản lý, người dùng cơ bản.

#### 6.1.2 Hạn chế

- Giao diện chưa tối ưu.
- Còn thiếu nhiều chức năng nghiệp vụ.
- Quản lý hình ảnh còn nhiều hạn chế.
- Chưa có chức năng thống kê doanh thu.
- Phân quyền người dùng chưa đảm bảo kỹ thuật, chưa phân quyền người dùng ở mức cơ sở dữ liệu.

### 6.2 Hướng phát triển

- Tiếp tục bổ sung các chức năng còn thiếu, sửa những chức năng đang bị lỗi.
- Tiếp tục phát triển các chức năng mới để có thể áp dụng rộng rãi cho nhiều đối tượng người dùng.

## CHƯƠNG 7: TÀI LIỆU THAM KHẢO

1. TOPDEV- Giới thiệu về Reactjs- <https://topdev.vn/blog/reactjs-nhung-dieu-ban-can-phai-biet/>
2. VIBLO- Bắt đầu với Reactjs- <https://viblo.asia/p/bat-dau-voi-reactjs-phan-1-3P0lPOE8Zox>
3. W3SCHOOL- Mysql tutorial- <https://www.w3schools.com/MySQL/default.asp>
4. Spring.io- Spring boot- <https://spring.io/projects/spring-boot>
5. Baeldung- Learn Spring boot- <https://www.baeldung.com/spring-boot>
6. Stack Overflow- fix bug- <https://stackoverflow.com/>