TRƯỜNG ĐẠI HỌC PHÚ YÊN

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

A red blue and white circle with white text

AI-generated content may be incorrect.

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

*Tên đề tài:*

**XÂY DỰNG WEBSITE ĐẶT TOUR DU LỊCH CÁC ĐỊA ĐIỂM VĂN HOÁ VÀ LÀNG NGHỀ TẠI TỈNH PHÚ YÊN**

**Ngành học : Công nghệ thông tin**

**Khóa : 2021-2025**

**Sinh viên thực hiện : Ngô Thành Đạt**

**Giảng viên hướng dẫn : Trần Xuân Hiệp**

*Phú Yên, 5/2025*

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU VỀ CƠ SỞ THỰC TẬP 8](#_Toc197726096)

[1.1. Thông tin về công ty 8](#_Toc197726097)

[1.2. Tổng quan về công ty 8](#_Toc197726098)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 9](#_Toc197726099)

[2.1. Sring Boot Framework 9](#_Toc197726100)

[2.1. Mô hình RESTful API trong hệ thống web 11](#_Toc197726101)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 16](#_Toc197726102)

[3.1. Mô tả bài toán. 16](#_Toc197726103)

[3.2. Đặc tả yêu cầu 16](#_Toc197726104)

[3.3. Sơ đồ Use Case 18](#_Toc197726105)

[3.4. Sơ đồ hoạt động 27](#_Toc197726106)

[3.5. Sơ đồ tuần tự 41](#_Toc197726107)

[3.6. Sơ đồ lớp 56](#_Toc197726108)

[CHƯƠNG 4. WEBSITE ĐẶT TOUR DU LỊCH CÁC ĐỊA ĐIỂM VĂN HOÁ VÀ LÀNG NGHỀ TẠI TỈNH PHÚ YÊN 63](#_Toc197726109)

[4.1. Giao diện trang chủ 64](#_Toc197726110)

[4.2. Giao diện xem chi tiết và đặt tour 65](#_Toc197726111)

[4.3. Giao diện quản lý thông tin cá nhân 66](#_Toc197726112)

[4.4. Giao diện quản lý lịch sử đặt tour 67](#_Toc197726113)

[4.5. Giao diện quản lý tour 69](#_Toc197726114)

[4.6. Giao diện quản lý lịch trình tour 70](#_Toc197726115)

[4.7. Giao diện quản lý các đơn đặt tour 71](#_Toc197726116)

[4.8. Giao diện quản lý chương trình khuyến mãi 71](#_Toc197726117)

[4.9. Giao diện quản lý người dùng. 73](#_Toc197726118)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN. 74](#_Toc197726119)

[TÀI LIỆU THAM KHẢO 75](#_Toc197726120)

**DANH MỤC HÌNH ẢNH**

[Hình 2. 1. Minh hoạt về RESTful API 11](#_Toc197704109)

[Hình 3.1. Sơ đồ use case tổng quát. 18](#_Toc197726046)

[Hình 3.2. Sơ đồ use case quản lý tour. 19](#_Toc197726047)

[Hình 3. 3. Sơ đồ use case quản lý các đơn đặt tour 20](#_Toc197726048)

[Hình 3. 4. Sơ đồ use case quản lý chương trình khuyến mãi. 21](#_Toc197726049)

[Hình 3. 5. Sơ đồ use case quản lý người dùng 23](#_Toc197726050)

[Hình 3. 6. Sơ đồ use case quản lý thông tin cá nhân. 24](#_Toc197726051)

[Hình 3. 7. Sơ đồ use case quản lý lịch sử đặt tour 25](#_Toc197726052)

[Hình 3. 8. Sơ đồ hoạt động thêm mới tour. 27](#_Toc197726053)

[Hình 3. 9. Sơ đồ hoạt động sửa thông tin tour. 28](#_Toc197726054)

[Hình 3. 10. Sơ đồ hoạt động thay đổi trạng thái tour 29](#_Toc197726055)

[Hình 3. 11. Sơ đồ hoạt động thêm mới lịch trình tour 30](#_Toc197726056)

[Hình 3. 12. Sơ đồ hoạt động xoá lịch trình tour 31](#_Toc197726057)

[Hình 3. 13. Sơ đồ hoạt động thay đổi trạng thái đơn đặt tour 32](#_Toc197726058)

[Hình 3. 14. Sơ đồ hoạt động thêm mới chương trình khuyến mãi cho tour. 33](#_Toc197726059)

[Hình 3. 15. Sơ đồ hoạt động xoá chương trình khuyến mãi. 34](#_Toc197726060)

[Hình 3. 16. Sơ đồ hoạt động sửa thông tin người dùng. 35](#_Toc197726061)

[Hình 3. 17. Sơ đồ hoạt động xoá người dùng. 36](#_Toc197726062)

[Hình 3. 18. Sơ đồ hoạt động sửa thông tin cá nhân. 37](#_Toc197726063)

[Hình 3. 19. Sơ đồ hoạt động chỉnh sửa thông tin tour đã đặt. 38](#_Toc197726064)

[Hình 3. 20. Sơ đồ hoạt động huỷ tour đã đặt. 39](#_Toc197726065)

[Hình 3. 21. Sơ đồ hoạt động đặt tour. 40](#_Toc197726066)

[Hình 3. 22. Sơ đồ tuần tự thêm mới tour. 41](#_Toc197726067)

[Hình 3. 23. Sơ đồ tuần tự sửa thông tin tour. 42](#_Toc197726068)

[Hình 3. 24. Sơ đồ tuần tự thay đổi trạng thái tour 43](#_Toc197726069)

[Hình 3. 25. Sơ đồ tuần tự thêm mới lịch trình tour 44](#_Toc197726070)

[Hình 3. 26. Sơ đồ tuần tự xoá lịch trình tour. 45](#_Toc197726071)

[Hình 3. 27. Sơ đồ tuần tự cập nhật trạng thái đơn đặt tour. 46](#_Toc197726072)

[Hình 3. 28. Sơ đồ tuần tự thêm mới chương trình khuyến mãi 47](#_Toc197726073)

[Hình 3. 29. Sơ đồ tuần tự cập nhật trạng thái chương trình khuyến mãi. 48](#_Toc197726074)

[Hình 3. 30. Sơ đồ tuần tự sửa thông tin người dùng. 49](#_Toc197726075)

[Hình 3. 31. Sơ đồ tuần tự xoá tài khoản người dùng. 50](#_Toc197726076)

[Hình 3. 32. Sơ đồ tuần tự sửa thông tin cá nhân. 51](#_Toc197726077)

[Hình 3. 33. Sơ đồ tuần tự chỉnh sửa thông tin đơn đặt tour 52](#_Toc197726078)

[Hình 3. 34. Sơ đồ tuần tự huỷ tour 53](#_Toc197726079)

[Hình 3. 35. Sơ đồ tuần tự đặt tour 54](#_Toc197726080)

[Hình 3. 36. Sơ đồ tuần tự đặt tour. 55](#_Toc197726081)

[Hình 3. 37. Sơ đồ lớp 56](#_Toc197726082)

[Hình 3. 38. Lược đồ cơ sở dữ liệu quan hệ. 57](#_Toc197726083)

[Hình 4. 1. Sơ đồ phân cấp trang web - vai trò: khách hàng 63](#_Toc197726084)

[Hình 4. 2. Giao diện trang chủ 64](#_Toc197726085)

[Hình 4. 3. Giao diện đặt tour 65](#_Toc197726086)

[Hình 4. 4. Giao diện quản lý thông tin cá nhân. 66](#_Toc197726087)

[Hình 4. 5. Giao diện quản lý các tour đã đặt 67](#_Toc197726088)

[Hình 4. 6. Sơ đồ phân cấp trang web - Vai trò: Admin 68](#_Toc197726089)

[Hình 4. 7. Giao diện quản lý tour 69](#_Toc197726090)

[Hình 4. 8. Giao diện quản lý lịch trình tour 70](#_Toc197726091)

[Hình 4. 9. Giao diện quản lý booking 71](#_Toc197726092)

[Hình 4. 10. Giao diện quản lý chương trình khuyến mãi 72](#_Toc197726093)

[Hình 4. 11. Giao diện thêm chương trình khuyễn mãi 72](#_Toc197726094)

[Hình 4. 12. Giao diện quản lý người dùng 73](#_Toc197726095)

**LỜI CẢM ƠN**

Trước tiên, em xin được bày tỏ lòng biết ơn chân thành và sâu sắc đến thầy Trần Xuân Hiệp – người đã tận tình hướng dẫn, đồng hành và hỗ trợ em trong suốt quá trình thực hiện đề tài tốt nghiệp “Xây dựng website đặt tour du lịch các địa điểm văn hoá và làng nghề tại tỉnh Phú Yên”. Thầy không chỉ giúp em định hướng rõ ràng về nội dung nghiên cứu, mà còn luôn sẵn sàng góp ý, hỗ trợ tận tình, giúp em từng bước hoàn thiện đề tài một cách tốt nhất.

Em xin chân thành cảm ơn quý thầy cô khoa Kỹ thuật – Công nghệ, Trường Đại học Phú Yên vì đã truyền đạt cho em những kiến thức quý báu trong các học phần liên quan đến lập trình, phân tích thiết kế hệ thống, phát triển ứng dụng web,… Nhờ đó, em đã có được nền tảng vững chắc để áp dụng vào việc xây dựng và triển khai hệ thống. Những bài học từ thầy cô không chỉ hữu ích trong đề tài này mà còn là hành trang quan trọng cho con đường nghề nghiệp của em sau này.

Em cũng xin gửi lời cảm ơn chân thành đến công ty IMT Miền Trung, nơi em có cơ hội thực tập, học hỏi kinh nghiệm thực tế và được tạo điều kiện tiếp cận môi trường làm việc chuyên nghiệp. Sự hỗ trợ từ các anh chị trong công ty đã giúp em hiểu rõ hơn về quy trình phát triển phần mềm, đồng thời góp phần quan trọng vào việc hoàn thiện đề tài này.

Một lần nữa, em xin chân thành cảm ơn tất cả sự quan tâm, giúp đỡ và những đóng góp quý báu mà em đã nhận được. Em xin kính chúc quý thầy cô luôn mạnh khỏe và thành công trong sự nghiệp giáo dục.

Em xin chân thành cảm ơn

Phú Yên, ngày 26 tháng 4 năm 2025

Sinh viên thực hiện

Ngô Thành Đạt

**MỞ ĐẦU**

1. **Lý do chọn đề tài**

Trong bối cảnh công nghệ thông tin ngày càng phát triển, việc ứng dụng các giải pháp số vào lĩnh vực du lịch đang trở thành xu hướng tất yếu. Ngành du lịch hiện không chỉ cần nâng cao chất lượng dịch vụ mà còn phải chú trọng đến việc tối ưu hóa quá trình tiếp cận khách hàng, hỗ trợ du khách trong việc tra cứu thông tin, đặt tour và nâng cao trải nghiệm một cách nhanh chóng, thuận tiện.

Phú Yên là một trong những điểm đến du lịch hấp dẫn tại Việt Nam, nổi bật với nhiều danh lam thắng cảnh như Ghềnh Đá Đĩa, Bãi Xép, Mũi Điện,... Bên cạnh những địa danh đã quen thuộc, nơi đây còn sở hữu nhiều địa điểm văn hóa và làng nghề truyền thống mang đậm bản sắc địa phương. Tuy nhiên, các giá trị này vẫn chưa được khai thác hiệu quả trên các nền tảng du lịch số. Xuất phát từ thực tế đó, đề tài được triển khai nhằm không chỉ cung cấp thông tin về các tour du lịch phổ biến mà còn hướng đến phát triển các tour trải nghiệm mang tính bản địa, góp phần gìn giữ và phát huy văn hóa địa phương.

Spring Boot được lựa chọn làm nền tảng phát triển backend của hệ thống nhờ khả năng đơn giản hóa việc xây dựng ứng dụng web, hiệu suất cao, dễ mở rộng và tích hợp tốt với các công nghệ hiện đại. Kiến trúc RESTful API được áp dụng nhằm đảm bảo sự phân tách rõ ràng giữa frontend và backend, tạo điều kiện thuận lợi cho quá trình phát triển và bảo trì hệ thống.

Thông qua đề tài không chỉ giúp em rèn luyện và củng cố những kiến thức đã học, mà còn áp dụng được những kiến thức mới và kinh nghiệm từ quá trình thực tập - điều này giúp em cải thiện đáng kể kỹ năng lập trình và khả năng giải quyết vấn đề một cách hiệu quả hơn.

1. **Mục tiêu đề tài.**

* Xây dựng website đặt tour tỉnh Phú Yên chuyên nghiệp, dễ sử dụng và cung cấp chức năng đặt tour và các chức năng quản lý.
* Ứng dụng công nghệ Spring Boot để phát triển hệ thống hiệu suất cao, dễ mở rộng, tận dụng kiến thức thực tập vào đề tài.

1. **Đối tượng và phạm vi nghiên cứu**
   1. **Đối tượng nghiên cứu**

* Các địa điểm du lịch tại Phú Yên.
* Quy trình hoạt động của hệ thống đặt tour du lịch.
* Spring Boot Framework.
  1. **Phạm vi nghiên cứu**
* Các danh lam thắng cảnh, địa điểm văn hóa và làng nghề truyền thống tại Phú Yên.
* Tài liệu về Spring Framework.

1. **Phương pháp nghiên cứu**

* Phương pháp thu thập tài liệu: Nghiên cứu các tài liệu về xây dựng webiste sử dụng framework Sping Boot và các hệ thống đặt tour trực tuyến của các nền tảng khác.
* Phương pháp thực nghiệm: Thực hiện xây dựng website tiến hành kiểm thử tính năng, đánh giá hiệu suất và tối ưu hệ thống.

1. **Nội dung nghiên cứu**

**Chương 1: Giới thiệu về cơ sở thực tập.**

* 1. Thông tin về công ty
  2. Tổng quan về công ty

**Chương 2: Cơ sở lý thuyết.**

2.1. Sring Boot Framework

2.2. Mô hình RESTful API trong hệ thống web

**Chương 3: Phân tích và thiết kế hệ thống.**

3.1. Đặc tả yêu cầu

3.2. Sơ đồ use case

3.3. Sơ đồ hoạt động

3.4. Sơ đồ tuần tự

3.5. Sơ đồ lớp

**Chương 4: Website đặt tour du lịch tỉnh Phú Yên.**

**Kết luận.**

**Tài liệu tham khảo.**

# GIỚI THIỆU VỀ CƠ SỞ THỰC TẬP

## Thông tin về công ty

* Công ty cổ phần phần mềm IMT Miền Trung
* Địa chỉ: 10 Nguyễn Mỹ, Phường 9, TP Tuy Hoà, Tỉnh Phú Yên, Việt Nam
* Số điện thoại: 0905576046
* Ngành nghề chính: Lập trình máy vi tính

## Tổng quan về công ty

IMT Solutions là một trong những công ty dẫn đầu trong lĩnh vực dịch vụ phần mềm và chuyển đổi số, được thành lập vào năm 2008 bởi các kỹ sư người Việt Nam có nhiều kinh nghiệm quốc tế. Trụ sở chính của IMT Solutions đặt tại TP. Hồ Chí Minh, Việt Nam, với nhiều chi nhánh phát triển trên toàn quốc, trong đó nổi bật là IMT Central tại Tuy Hòa, Phú Yên.

*IMT Central - Hướng Tới Sự Phát Triển Bền Vững*

Nhằm mở rộng quy mô hoạt động và phát triển nguồn nhân lực chất lượng cao, IMT Solutions đã thành lập IMT Central tại Tuy Hòa, Phú Yên. Đây không chỉ là một trung tâm phát triển phần mềm mà còn là một cầu nối quan trọng giữa công ty và các trường đại học, cao đẳng tại khu vực miền Trung. IMT Central tập trung vào nghiên cứu và phát triển các giải pháp công nghệ, ứng dụng tự động hóa quy trình (RPA), trí tuệ nhân tạo (AI), kiểm thử phần mềm tự động, và các dự án chuyển đổi số phục vụ khách hàng trong và ngoài nước.

*Thành Tựu và Định Hướng Phát Triển*

Trong những năm qua, IMT Solutions đã đạt được nhiều thành tựu đáng kể, bao gồm các giải thưởng danh giá như VNITO’s Leading IT Outsourcing Award, Top 50 Vietnam IT Company Award, giải Sao Khuê cho dịch vụ gia công phần mềm năm 2016, chứng chỉ ISO 27001:2013 và danh hiệu Đối tác Vàng của Microsoft trong 10 năm liên tiếp.

IMT Central, với vai trò là một trung tâm quan trọng trong chiến lược phát triển của công ty, tiếp tục mở rộng quy mô và tập trung vào các dự án công nghệ cao. Bên cạnh đó, trung tâm còn góp phần thúc đẩy quá trình chuyển đổi số tại khu vực miền Trung, giúp các doanh nghiệp địa phương tiếp cận công nghệ hiện đại và nâng cao năng lực cạnh tranh.

*Đóng Góp Cộng Đồng và Trách Nhiệm Xã Hội*

Không chỉ chú trọng vào hoạt động kinh doanh, IMT Central cũng tích cực tham gia các hoạt động cộng đồng, như hỗ trợ sinh viên, hợp tác đào tạo với các trường đại học, tổ chức các chương trình hướng nghiệp và hỗ trợ chuyển đổi số cho các doanh nghiệp vừa và nhỏ tại khu vực miền Trung.

# CƠ SỞ LÝ THUYẾT

**2.1. Sring Boot Framework**

***Spring Boot là gì ?.***

Spring Boot là một framework giúp phát triển ứng dụng Java nhanh chóng, đơn giản bằng cách tự động cấu hình (Auto-Configuration) và cung cấp các tính năng sẵn có. Được xây dựng trên nền tảng Spring Framework phát triển và ra mắt phiên bản đầu tiên (Spring Boot 1.0) vào tháng 4 năm 2014.

Một trong những điểm đột phá của Spring Boot là công cụ Spring Initializr, cho phép các nhà phát triển khởi tạo dự án nhanh chóng với các thư viện cần thiết chỉ trong vài thao tác. Nhờ những cải tiến này, Spring Boot đã trở thành nền tảng mạnh mẽ, giúp tối ưu quá trình phát triển và triển khai ứng dụng Java hiện đại [1].

***Đặc điểm nổi bật của Spring Boot.***

Tối ưu hóa quá trình phát triển: Spring Boot cung cấp cấu hình mặc định thông minh và tự động, giúp giảm thiểu việc thiết lập thủ công, từ đó rút ngắn thời gian phát triển ứng dụng Java.

Tích hợp mạnh mẽ: Hệ sinh thái Spring Boot cho phép dễ dàng tích hợp với các thư viện và module trong Spring Framework, giúp việc kết nối các dịch vụ và hệ thống trở nên đơn giản hơn mà không cần cấu hình phức tạp.

Máy chủ nhúng (Embedded Server): Spring Boot đi kèm với các máy chủ nhúng như Tomcat, Jetty hoặc Undertow, giúp triển khai ứng dụng dễ dàng mà không cần cấu hình thêm máy chủ bên ngoài.

Cấu hình tự động: Cơ chế tự động cấu hình thông minh của Spring Boot giúp ứng dụng tự nhận diện và thiết lập các thành phần cần thiết dựa trên thư viện và module được sử dụng.

Quản lý phụ thuộc hiệu quả: Spring Boot hỗ trợ quản lý phụ thuộc thông qua Maven hoặc Gradle, giúp đơn giản hóa quá trình thêm, cập nhật và kiểm soát thư viện trong ứng dụng.

Giám sát và quản lý hệ thống: Spring Boot cung cấp các công cụ mạnh mẽ như Spring Boot Actuator, giúp giám sát, thu thập thông tin và quản lý hiệu suất ứng dụng dễ dàng [2]

***Các tính năng chính .***

*Spring Application*

Tiện ích Spring Application chịu trách nhiệm cho việc cấu hình và khởi động ứng dụng Spring Boot, cụ thể như:

Spring Application tận dụng tính chất cấu hình tự động của Spring Boot để giảm thiểu sự phức tạp trong cấu hình ứng dụng. Hệ thống tự động quét các gói chứa các lớp và component của ứng dụng, giúp ứng dụng tự động phát hiện và cấu hình các bean cần thiết.

Tiện ích đem lại các cấu hình mặc định cho các ứng dụng web. Kèm theo đó là khả năng kích hoạt khả năng tích hợp với các máy chủ nhúng như Tomcat, Jetty, hoặc Undertow giúp việc triển khai ứng dụng web trở nên dễ dàng hơn [3].

*Externalized Configuration*

Trong Spring Boot, tính năng này cho phép người dùng thực thi quá trình cấu hình ứng dụng bằng các tập tin cấu hình bên ngoài, biến môi trường mà không cần phải chỉnh sửa mã nguồn. Đây là thành phần nâng cao sự linh hoạt cho ứng dụng, cho phép cấu hình thay đổi mà không cần phải biên dịch lại mã nguồn. Một số tiện ích trên công cụ cần được kể đến chính là:

External Configuration Files: Cho phép cấu hình ứng dụng bằng cách sử dụng các tệp tin cấu hình như YAML, Properties, hoặc JSON. Người dùng có thể chỉ định vị trí của các tệp tin này bằng cách sử dụng tham số dòng lệnh, biến môi trường hoặc vị trí mặc định.

Setting Profiles: Cho phép sử dụng các profiles để cấu hình ứng dụng cho các môi trường khác nhau như development, testing, staging và production. Profiles cho phép bạn chỉ định cấu hình riêng biệt cho từng môi trường mà không cần phải thay đổi mã nguồn.

Environment Variables: Spring Boot hỗ trợ cấu hình bằng biến môi trường, cho phép bạn điều chỉnh ứng dụng một cách dễ dàng mà không cần phải sửa đổi tập tin cấu hình [3].

*Profiles*

Profiles trong Spring Boot là một cơ chế cho phép định nghĩa và quản lý cấu hình ứng dụng theo từng môi trường cụ thể, chẳng hạn như development, testing, staging và production, mà không làm thay đổi mã nguồn. Cơ chế này hỗ trợ việc triển khai ứng dụng trên nhiều môi trường khác nhau mà không yêu cầu sửa đổi trực tiếp trong mã lập trình, từ đó tăng tính linh hoạt và khả năng bảo trì của hệ thống.

Yếu tố này đã giúp ứng dụng hoàn toàn có thể chuyển đổi việc triển khai ở các môi trường khác nhau. Các tính năng chính của Profiles trong Spring Boot bao gồm:

Định nghĩa Profiles: Spring Boot cho phép tạo ra các tệp cấu hình riêng biệt cho từng môi trường bằng cách sử dụng tiền tố "application-" kết hợp với tên profile. Ví dụ: "application-dev.yml" cho môi trường development, "application-prod.yml" cho môi trường production và cũng có thể sử dụng các properties tương ứng.

Áp dụng Profiles: Có thể chỉ định profile được sử dụng thông qua các cấu hình hoặc tham số dòng lệnh khi khởi chạy ứng dụng. Spring Boot sẽ tải cấu hình từ tập tin phù hợp với profile được chỉ định.

Overriding: Profiles cho phép override các cấu hình mặc định được chỉ định trong tệp cấu hình chung. Điều này giúp bạn tuỳ chỉnh cấu hình mà không cần phải sửa đổi tập tin gốc [3].

*Logging*

Logging là một thành phần quan trọng trong quá trình phát triển phần mềm, giúp theo dõi và phân tích hoạt động của hệ thống. Spring Boot cung cấp hệ gứng dụng.

So với các framework khác, Spring Boot nổi bật nhờ khả năng đơn giản hóa quá trình phát triển trong khi vẫn đảm bảo tính linh hoạt và hiệu suất cao. Nếu Jakarta EE yêu cầu cấu hình phức tạp và phụ thuộc nhiều vào máy chủ ứng dụng, thì Spring Boot với cơ chế cấu hình tự động và máy chủ nhúng giúp việc triển khai trở nên nhanh chóng, thuận tiện hơn.

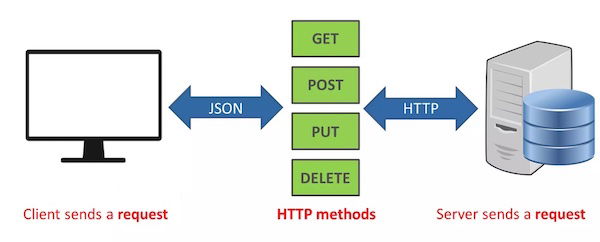
Bên cạnh đó, các framework như Quarkus hay Micronaut được tối ưu hóa cho môi trường cloud-native với thời gian khởi động nhanh, nhưng vẫn chưa thể sánh với hệ sinh thái phong phú và sự hỗ trợ toàn diện của Spring Boot. Trong khi đó, các framework non-Java như Django hay Laravel phù hợp để phát triển các ứng dụng web đơn giản nhưng chưa đủ mạnh mẽ để xây dựng các hệ thống lớn, đặc biệt là kiến trúc microservices.

Dự án sẽ kết hợp sử dụng Spring Boot cho backend, Angular cho frontend và MySQL làm cơ sở dữ liệu, tạo nên một hệ thống web hiện đại với kiến trúc rõ ràng, hiệu suất cao và khả năng mở rộng linh hoạt [3].

## **Mô hình RESTful API trong hệ thống web**

***RESTful API là gì?***

RESTful API (Representational State Transfer Application Programming Interface) là một kiểu thiết kế giao diện lập trình ứng dụng dựa trên kiến trúc REST. Đây là một phương pháp tiêu chuẩn để xây dựng các dịch vụ web, cho phép các hệ thống giao tiếp với nhau thông qua giao thức HTTP. RESTful API giúp ứng dụng web dễ dàng gửi và nhận dữ liệu từ máy chủ mà không phụ thuộc vào nền tảng hay công nghệ cụ thể.



Hình 2. . Minh hoạa về RESTful API

RESTful API hoạt động theo mô hình client-server, trong đó client gửi yêu cầu đến server thông qua các phương thức HTTP tiêu chuẩn. Server xử lý yêu cầu, thực hiện các thao tác cần thiết và phản hồi kết quả về cho client dưới định dạng dữ liệu phổ biến như JSON hoặc XML [4].

***Các thành phần chính của RESTful API***

*Resource (Tài nguyên)*

Mọi đối tượng trong hệ thống (chẳng hạn như người dùng, sản phẩm, đơn hàng) được xem như một resource và có một URI (Uniform Resource Identifier) duy nhất để truy cập. Việc tổ chức các resource theo nguyên tắc REST giúp hệ thống dễ dàng quản lý và mở rộng.

*HTTP Methods (Phương thức HTTP)*

RESTful API sử dụng các phương thức HTTP để thực hiện các thao tác trên resource:

* GET: Truy xuất thông tin từ server.
* POST: Tạo mới một resource.
* PUT: Cập nhật resource hiện có.
* DELETE: Xóa resource khỏi hệ thống.

Các phương thức này tuân theo nguyên tắc CRUD (Create - Read - Update - Delete), giúp API hoạt động một cách nhất quán và dễ hiểu.

*Status Code (Mã trạng thái HTTP)*

Sau khi xử lý yêu cầu, server phản hồi bằng mã trạng thái HTTP để chỉ ra kết quả:

* 200 OK: Thành công
* 201 Created: Resource được tạo thành công.
* 400 Bad Request: Yêu cầu không hợp lệ.
* 404 Not Found: Không tìm thấy resource.
* 500 Internal Server Error: Lỗi từ phía server.

*Hypermedia*

Hypermedia là các liên kết giữa các resource giúp cho việc điều hướng giữa các resource dễ dàng hơn, cũng như giúp cho client có thể tự động tìm kiếm và truy xuất các resource liên quan. Hypermedia có thể được định dạng bằng các định dạng như HTML, Atom hay JSON [5].

***Các bước thiết kế RESTful API***

*Bước 1. Xác định tài nguyên (Resource).*

Tài nguyên là đối tượng chính trong mô hình RESTful API, có thể là người dùng, sản phẩm, đơn hàng, v.v. Mỗi tài nguyên được định danh bằng một Uniform Resource Identifier (URI) duy nhất. Việc xác định tài nguyên giúp xây dựng một hệ thống API có cấu trúc rõ ràng, dễ hiểu và dễ mở rộng.

Ví dụ: Với tài nguyên người dùng trong hệ thống có thể định danh bằng URL : GET /users/{id} – Lấy thông tin một người dùng theo ID

*Bước 2. Xác định các phương thức HTTP.*

Sau khi đã xác định các tài nguyên, cần xác định các phương thức HTTP tương ứng để thao tác trên tài nguyên đó. RESTful API tuân theo các quy tắc chuẩn của giao thức HTTP với bốn phương thức chính: GET, POST, PUT, DELETE.

*Bước 3. Xác định định dạng dữ liệu.*

Dữ liệu được truyền giữa máy khách (Client) và máy chủ (Server) cần được định dạng theo một tiêu chuẩn chung để đảm bảo khả năng tương thích. Các định dạng phổ biến được sử dụng trong RESTful API bao gồm:

* JSON (JavaScript Object Notation) – Định dạng phổ biến nhất nhờ tính đơn giản, dễ đọc và hỗ trợ rộng rãi trên nhiều nền tảng.
* XML (Extensible Markup Language) – Thường được sử dụng trong các hệ thống cũ hoặc yêu cầu tính mô tả chi tiết hơn.

*Bước 4. Thiết kế URL*

Việc thiết kế URI đóng vai trò quan trọng trong RESTful API, giúp hệ thống có cấu trúc rõ ràng, dễ sử dụng và tuân theo các tiêu chuẩn chung. Khi thiết kế URI, cần tuân theo một số nguyên tắc như:

* Sử dụng danh từ thay vì động từ (ví dụ: /users thay vì /getUsers).
* Sử dụng dạng số nhiều cho tài nguyên (ví dụ: /products thay vì /product).
* Tránh sử dụng dấu gạch dưới (\_) hoặc chữ in hoa trong đường dẫn, thay vào đó sử dụng chữ thường và dấu gạch ngang (-).

*Bước 5. Hoàn thiện và tài liệu hoá API.*

Sau khi đã hoàn thành các bước trên, cần xây dựng cuối cùng của RESTful API. Cuối cùng sẽ là một tài liệu thiết kế được viết bằng ngôn ngữ thông dụng như Swagger hoặc RAML.

***Ưu điểm của mô hình RESTful API trong hệ thống web.***

*Tính dễ sử dụng và thuận tiện trong truy xuất dữ liệu*

RESTful API được thiết kế nhằm đơn giản hóa quá trình truy xuất dữ liệu và xử lý yêu cầu từ phía máy khách. Nhờ vào kiến trúc linh hoạt, các nhà phát triển không cần phải hiểu sâu về cấu trúc của cơ sở dữ liệu hay sử dụng nhiều ngôn ngữ lập trình khác nhau để xây dựng API. Thay vào đó, họ chỉ cần tập trung vào thiết kế các giao diện lập trình ứng dụng (API) có khả năng tương tác với các hệ thống bên ngoài một cách dễ dàng và hiệu quả.

*Tính mở và khả năng mở rộng cao*

Với việc hoạt động dựa trên các giao thức truyền tải dữ liệu phổ biến như HTTP và HTTPS, RESTful API có tính mở cao, cho phép các hệ thống có thể kết nối và tương tác với nhau một cách linh hoạt. Nhờ đặc điểm này, RESTful API không bị giới hạn trong một nền tảng cụ thể mà có thể triển khai trên nhiều môi trường khác nhau, hỗ trợ tích hợp giữa các hệ thống một cách hiệu quả.

*Tối ưu chi phí và thời gian phát triển*

Việc áp dụng RESTful API giúp các nhà phát triển tiết kiệm đáng kể thời gian và chi phí trong quá trình xây dựng ứng dụng. Nhờ vào khả năng tái sử dụng, mở rộng và tương thích với nhiều công nghệ hiện có, RESTful API giúp đẩy nhanh quá trình phát triển phần mềm, giảm thiểu các công đoạn cấu hình phức tạp, đồng thời tạo điều kiện thuận lợi để ứng dụng có thể tích hợp với nhiều nền tảng khác nhau một cách nhanh chóng và hiệu quả [2].

***Nhược điểm của mô hình RESTful API trong hệ thống web***

*Khả năng mở rộng*

Đây không phải là giải pháp tốt nhất cho các ứng dụng có yêu cầu khả năng mở rộng cao. Khi số lượng người dùng và lưu lượng truy cập tăng lên, RESTful API có thể gặp phải vấn đề về hiệu suất và tốc độ xử lý yêu cầu.

*Bảo mật*

RESTful API cung cấp cho các ứng dụng một giao diện lập trình dễ sử dụng, tuy nhiên điều này cũng đồng nghĩa với việc có thể gây ra những vấn đề về bảo mật. Các nhà phát triển cần phải đảm bảo rằng RESTfulAPI của họ được bảo vệ an toàn để tránh các cuộc tấn công từ bên ngoài và tình trạng lỗ hổng bảo mật.

*Khó khăn trong việc quản lý phiên*

RESTful API không hỗ trợ quản lý phiên, điều này gây ra khó khăn trong việc xác thực và quản lý người dùng. Khi sử dụng, các nhà phát triển cần phải có chính sách xác thực phù hợp để giữ cho thông tin của người dùng được bảo vệ an toàn.

RESTful API sẽ được sử dụng làm nền tảng cho hệ thống đặt tour du lịch tại Phú Yên, đóng vai trò trung gian kết nối giữa frontend và backend, đảm bảo việc giao tiếp dữ liệu diễn ra một cách hiệu quả và thống nhất. Việc áp dụng RESTful API giúp frontend có thể dễ dàng gửi yêu cầu và nhận phản hồi từ backend để thực hiện các chức năng quan trọng như tìm kiếm tour, đặt tour, quản lý thông tin khách hàng, thanh toán và theo dõi đơn đặt hàng [2].

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Mô tả bài toán.

Trong bối cảnh ngành du lịch ngày càng phát triển mạnh mẽ, nhu cầu của người dân về việc tìm kiếm và đặt tour du lịch một cách nhanh chóng, thuận tiện và đáng tin cậy ngày càng gia tăng. Cả khách hàng và các công ty du lịch đều mong muốn có một nền tảng trực tuyến hiện đại, dễ sử dụng để kết nối, đặt tour và quản lý lịch trình du lịch. Bài toán đặt ra là làm thế nào để thiết kế một hệ thống đặt tour du lịch trực tuyến có giao diện trực quan, dễ sử dụng và phù hợp với nhu cầu của cả khách hàng lẫn quản trị viên, đồng thời giải quyết được các vấn đề sau:

Khách hàng có thể:

* Xem thông tin chi tiết của tour.
* Đăng ký tài khoản và đăng nhập để thực hiện các chức năng cần thiết.
* Đặt tour và tiến hành thanh toán.
* Quản lý lịch sử đặt tour với khả năng theo dõi, chỉnh sửa hoặc hủy

Đối với tính năng quản trị thì hệ thống cung cấp đầy đủ các chức năng quản lý để hỗ trợ quản trị viên điều hành hoạt động du lịch hiệu quả.

## Đặc tả yêu cầu

***Yêu cầu chức năng***

*Chức năng dành cho người dùng (khách hàng):*

* Xem thông tin tour

Khách hàng vào ứng dụng để xem thông tin các tour du lịch, bao gồm: tên tour, mô tả chi tiết, lịch trình, thời gian khởi hành , dịch vụ đi kèm, giá tour và hình ảnh minh họa.

* Đăng ký

Tạo tài khoản cho khách hàng thực hiện các chức năng trong hệ thống

* Đăng nhập

Khách hàng phải đăng nhập vào hệ thống trước khi thực hiện đặt tour.

* Đặt tour

Sau khi chọn tour mong muốn, người dùng điền thông tin cá nhân và xác nhận đơn đặt chỗ và thực hiện thanh toán.

* Quản lý lịch sử đặt tour

Khách hàng có thể theo dõi, chỉnh sửa hoặc hủy các tour đã đặt.

* Tìm kiếm thông tin tour

*Chức năng dành cho admin*

* Quản lý thông tin tour

Thêm mới, chỉnh sửa các tour du lịch trên hệ thống và xem thống kê

* Quản lý lịch trình tour

Thêm mới, chỉnh sửa hoặc xóa các lịch trình mỗi tour.

* Quản lý các đơn đặt tour

Xem, cập nhật trạng thái các đơn đặt tour của khách hàng.

* Quản lý thông tin người dùng

Xem, chỉnh sửa, hoặc xoá tài khoản người dùng.

* Quản lý chương trình khuyến mãi

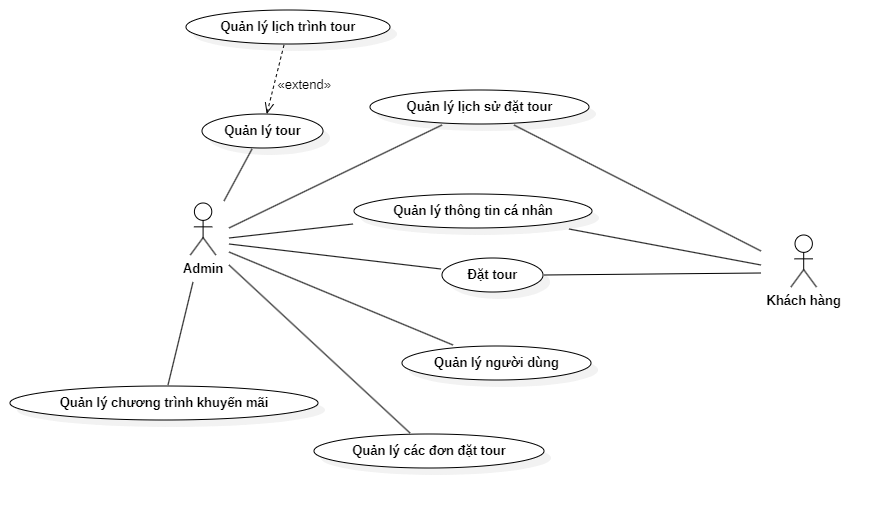
Xem, thêm mới, chỉnh sửa hoặc xóa các chương trình khuyến mãi cho các lịch trình tour.

***Yêu cầu phi chức năng***

* Giao diện bắt mắt mang lại trải nghiệm trực quan , dễ sử dụng
* Khả năng dễ bảo trì , dễ dàng nâng cấp, chỉnh sửa

## Sơ đồ Use Case

***Sơ đồ Use Case tổng quát***

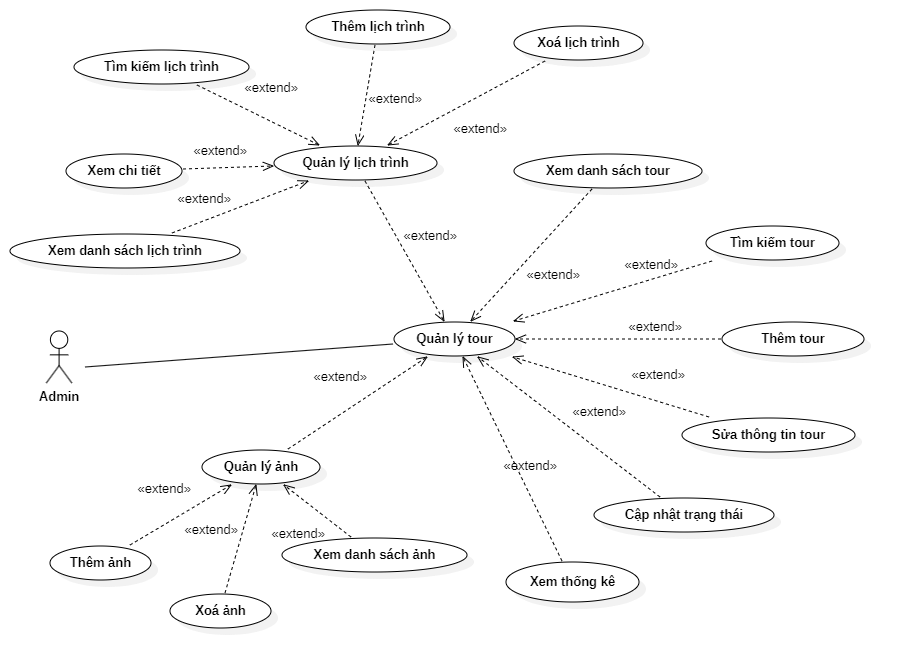


Hình 3.. Sơ đồ use case tổng quát.

***Sơ đồ Use Case chi tiết và đặc tả.***

Các sơ đồ use case chi tiết và đặc tả dành cho admin

* Sơ đồ use case quản lý tour

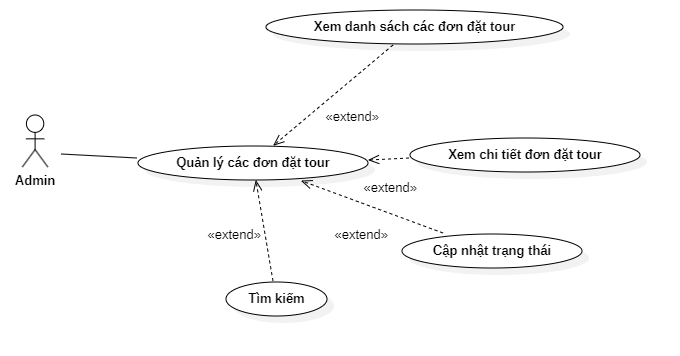


Hình 3.. Sơ đồ use case quản lý tour.

* Đặc tả Use case quản lý tour

|  |  |
| --- | --- |
| **Use case quản lý tour** | |
| Mục đích | Quản lý thông tin các tour |
| Mô tả | Thực hiện xem danh sách, tìm kiếm, thêm, sửa, cập nhật trạng thái , xem thống kê, quả lý ảnh, quản lý lịch trình tour |
| Tác nhân | Admin |
| Điều kiện trước | Đăng nhập thành công, chọn chức năng quản lý tour |
| Điều kiện sau | Đăng nhập thành công thì kích hoạt hệ thống, ngược lại thì báo lỗi đăng nhập lại. |
| Luồng sự kiện chính (Basic flows) | 1. Đăng nhập vào hệ thống với phân quyền admin. 2. Chọn chức năng quản lý tour. 3. Hệ thống hiển thị danh sách các tour được quản lý. 4. Admin thực hiện xem danh sách, tìm kiếm, thêm, sửa, cập nhật trạng thái , xem thống kê, quả lý ảnh, quản lý lịch trình tour |
| Luồng sự kiện phụ (Alternative Flows) | 1. Nếu thông tin không phù hợp hệ thống hiện thông báo lỗi. |

* Sơ đồ use case quản lý các đơn đặt tour

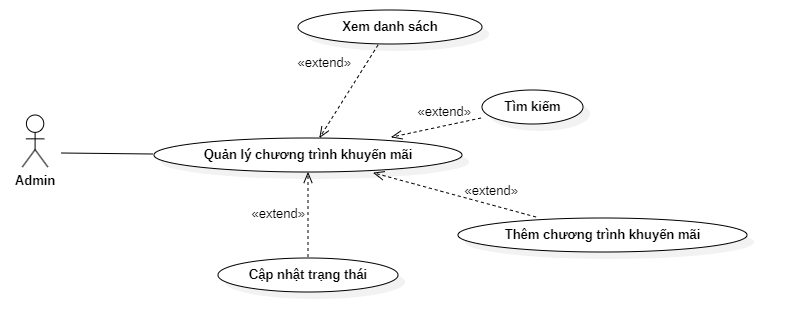


Hình 3. . Sơ đồ use case quản lý các đơn đặt tour

* Đặc tả use case quản lý các đơn đặt tour

|  |  |
| --- | --- |
| **Use case quản lý các đơn đặt tour** | |
| Mục đích | Quản lý thông tin booking |
| Mô tả | Thực hiện xem danh sách, tìm kiếm, xem chi tiết đơn đặt, cập nhật trạng thái đơn đặt tour |
| Tác nhân | Admin |
| Điều kiện trước | Đăng nhập thành công, chọn chức năng quản lý booking |
| Điều kiện sau | Đăng nhập thành công thì kích hoạt hệ thống, ngược lại thì báo lỗi đăng nhập lại. |
| Luồng sự kiện chính (Basic flows) | 1. Đăng nhập vào hệ thống với phân quyền admin. 2. Chọn chức năng quản lý các đơn đặt tour. 3. Hệ thống hiển thị danh sách. 4. Admin có thể thực hiện các thao tác như: tìm kiếm, xem chi tiết đơn đặt tour, cập nhật trạng thái đơn đặt tour |
| Luồng sự kiện phụ (Alternative Flows) | 1. Nếu thông tin không phù hợp hệ thống hiện thông báo lỗi. |

* Use case quản lý chương trình khuyến mãi

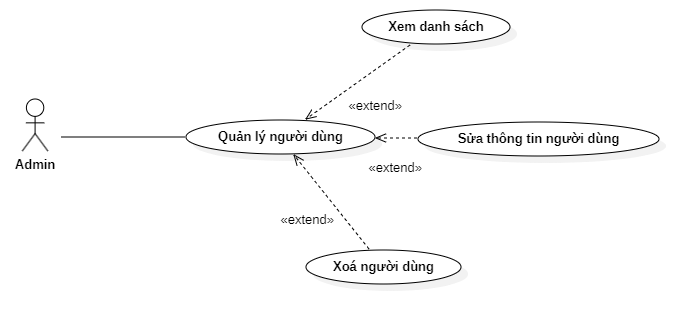


Hình 3. . Sơ đồ use case quản lý chương trình khuyến mãi.

* Đặc tả use case quản lý chương trình khuyến mãi

|  |  |
| --- | --- |
| **Use case quản lý chương trình khuyến mãi** | |
| Mục đích | Quản lý chương trình khuyến mãi các tour |
| Mô tả | Thực hiện xem, tìm kiếm, thêm mới, cập nhật trạng thái, xoá chương trình khuyến mãi |
| Tác nhân | Admin |
| Điều kiện trước | Đăng nhập thành công, chọn chức năng quản lý chương trình khuyến mãi |
| Điều kiện sau | Đăng nhập thành công thì kích hoạt hệ thống, ngược lại thì báo lỗi đăng nhập lại. |
| Luồng sự kiện chính (Basic flows) | 1. Đăng nhập vào hệ thống với phân quyền admin 2. Chọn chức năng quản lý chương trình khuyến mãi 3. Hệ thống hiển thị danh sách các chương tình khuyến mãi 4. Admin có thể thực hiện cập nhật trạng thái, tìm kiếm các chương trình khuyến mãi theo: tên tour, trạng thái. 5. Admin được phép xoá các chương trình khuyến mãi tạm ngưng |
| Luồng sự kiện phụ (Alternative Flows) | 6. Nếu thông tin không phù hợp hệ thống hiện thông báo lỗi. |

* Use case quản lý người dùng



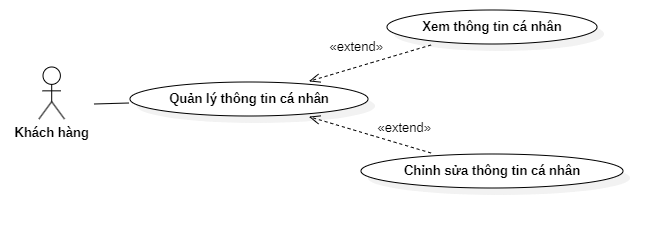
Hình 3. . Sơ đồ use case quản lý người dùng

* Đặc tả Use Case quản lý người dùng

|  |  |
| --- | --- |
| **Use case quản lý người dùng** | |
| Mục đích | Quản lý thông tin người dùng |
| Mô tả | Thực hiện sửa, xoá người dùng |
| Tác nhân | Admin |
| Điều kiện trước | Đăng nhập thành công, chọn chức năng quản lý người dùng |
| Điều kiện sau | Đăng nhập thành công thì kích hoạt hệ thống, ngược lại thì báo lỗi đăng nhập lại. |
| Luồng sự kiện chính (Basic flows) | 1. Đăng nhập vào hệ thống với phân quyền admin. 2. Chọn chức năng quản lý người dùng. 3. Hệ thống hiển thị danh sách người dùng. 4. Admin có thể thực hiện các thao tác xem, sửa, xoá người dùng. |
| Luồng sự kiện phụ (Alternative Flows) | 1. Nếu thông tin không phù hợp hệ thống hiện thông báo lỗi. |

Các sơ đồ use case chi tiết và đặc tả dành cho khách hàng

* Use Case quản lý thông tin cá nhân

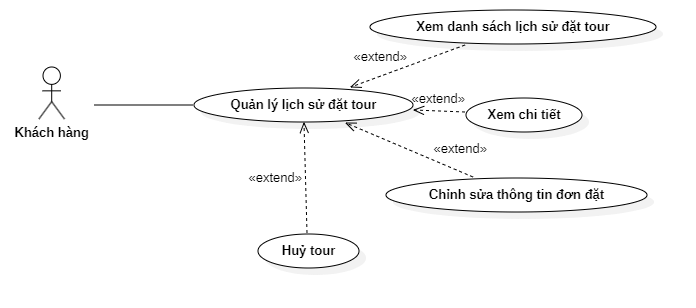


Hình 3. . Sơ đồ use case quản lý thông tin cá nhân.

* Đặc tả Use Case quản lý cá nhân

|  |  |
| --- | --- |
| **Use case quản lý thông tin cá nhân** | |
| Mục đích | Quản lý thông tin cá nhân |
| Mô tả | Xem, sửa thông tin cá nhân |
| Tác nhân | Khách hàng |
| Điều kiện trước | Đăng nhập thành công, chọn chức năng quản lý thông tin cá nhân |
| Điều kiện sau | Đăng nhập thành công thì kích hoạt hệ thống, ngược lại thì báo lỗi đăng nhập lại. |
| Luồng sự kiện chính (Basic flows) | 1. Đăng nhập vào hệ thống với phân quyền khách hàng. 2. Chọn chức năng quản lý thông tin cá nhân. 3. Hệ thống hiển thị thông tin cá nhân 4. Người dùng có thể thực hiện thao tác sửa thông tin |
| Luồng sự kiện phụ (Alternative Flows) | 1. Nếu thông tin không phù hợp hệ thống hiện thông báo lỗi. |

* Quản lý lịch sử đặt tour



Hình 3. . Sơ đồ use case quản lý lịch sử đặt tour

* Đặc tả use case quản lý lịch sử đặt tour.

|  |  |
| --- | --- |
| **Use case quản lý lịch sử đặt tour** | |
| Mục đích | Quản lý các tour mà khách hàng đã đặt |
| Mô tả | Thực hiện xem danh sách, xem chi tiết, chỉnh sửa thông tin và huỷ đơn đặt tour |
| Tác nhân | Khách hàng |
| Điều kiện trước | Đăng nhập thành công, chọn chức năng quản lý các tour đã đặt |
| Điều kiện sau | Đăng nhập thành công thì kích hoạt hệ thống, ngược lại thì báo lỗi đăng nhập lại. |
| Luồng sự kiện chính (Basic flows) | 1. Đăng nhập vào hệ thống với phân quyền khách hàng. 2. Chọn chức năng quản lý các tour đã đặt. 3. Hệ thống hiển thị danh sách các tour đã đặt 4. Người dùng có thể thực hiện thao tác sửa thông các đơn đặt hoặc huỷ với trạng thái “Đã thanh toán” |
| Luồng sự kiện phụ (Alternative Flows): | 1. Nếu thông tin không phù hợp hệ thống hiện thông báo. |

* + Đặc tả Use Case Đặt tour

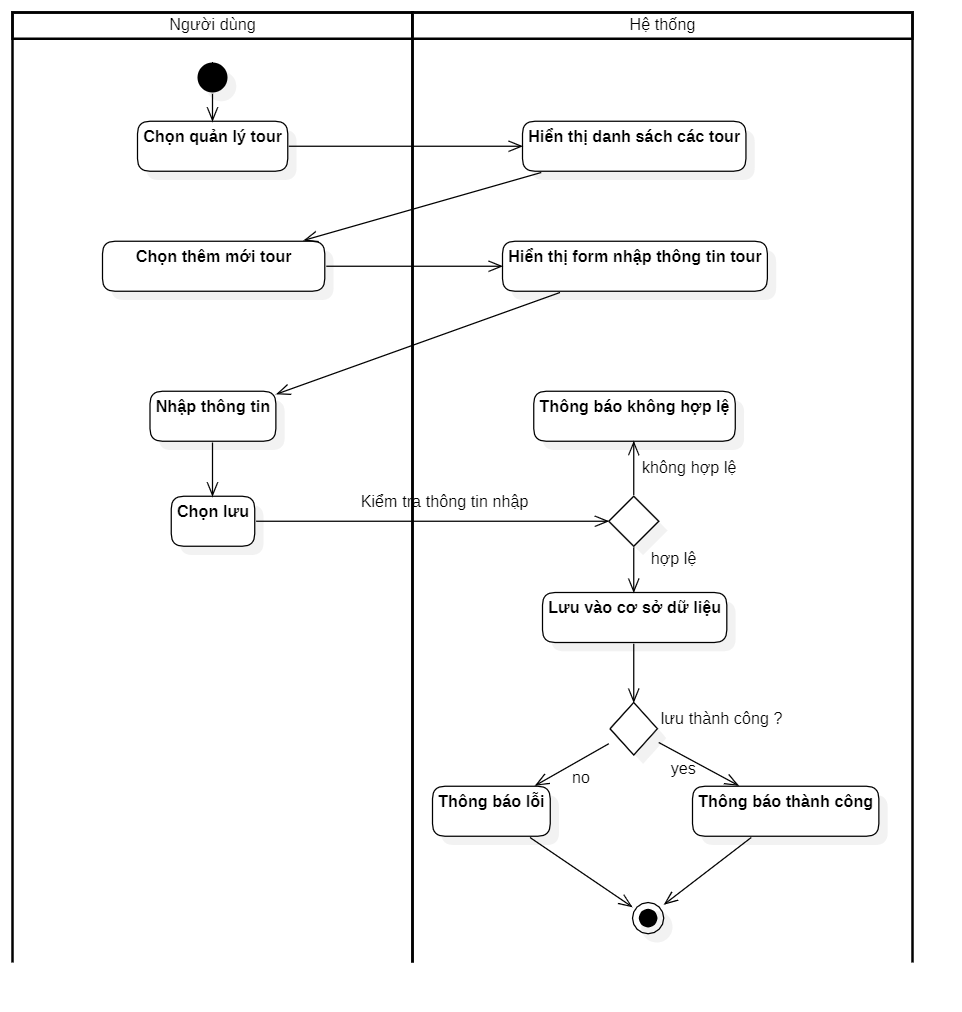
|  |  |
| --- | --- |
| **Use case đặt tour** | |
| Mục đích | Cho phép khách hàng thực hiện đặt tour du lịch |
| Mô tả | Đặt tour |
| Tác nhân | Khách hàng |
| Điều kiện trước | Đăng nhập thành công |
| Điều kiện sau | Đăng nhập thành công thì kích hoạt hệ thống, ngược lại thì báo lỗi đăng nhập lại. |
| Luồng sự kiện chính (Basic flows) | 1. Khách hàng đăng nhập vào hệ thống 2. Khách hàng chọn tour mong muốn từ danh sách tour đang mở bán 3. Hệ thống hiển thị form nhập thông tin đặt tour 4. Khách hàng nhập thông tin cá nhân, danh sách người đi cùng (nếu có), và xác nhận nhóm khách phù hợp theo độ tuổi 5. Khách hàng tiến hành xác nhận đặt tour và chuyển sang bước thanh toán 6. Khách hàng thực hiện thanh toán. 7. Hệ thống hiển thị thông báo đặt tour thành công |
| Luồng sự kiện phụ (Alternative Flows) | 7. Nếu thông tin không phù hợp hệ thống hiện thông báo. |

## Sơ đồ hoạt động

Các sơ đồ hoạt động dành cho admin

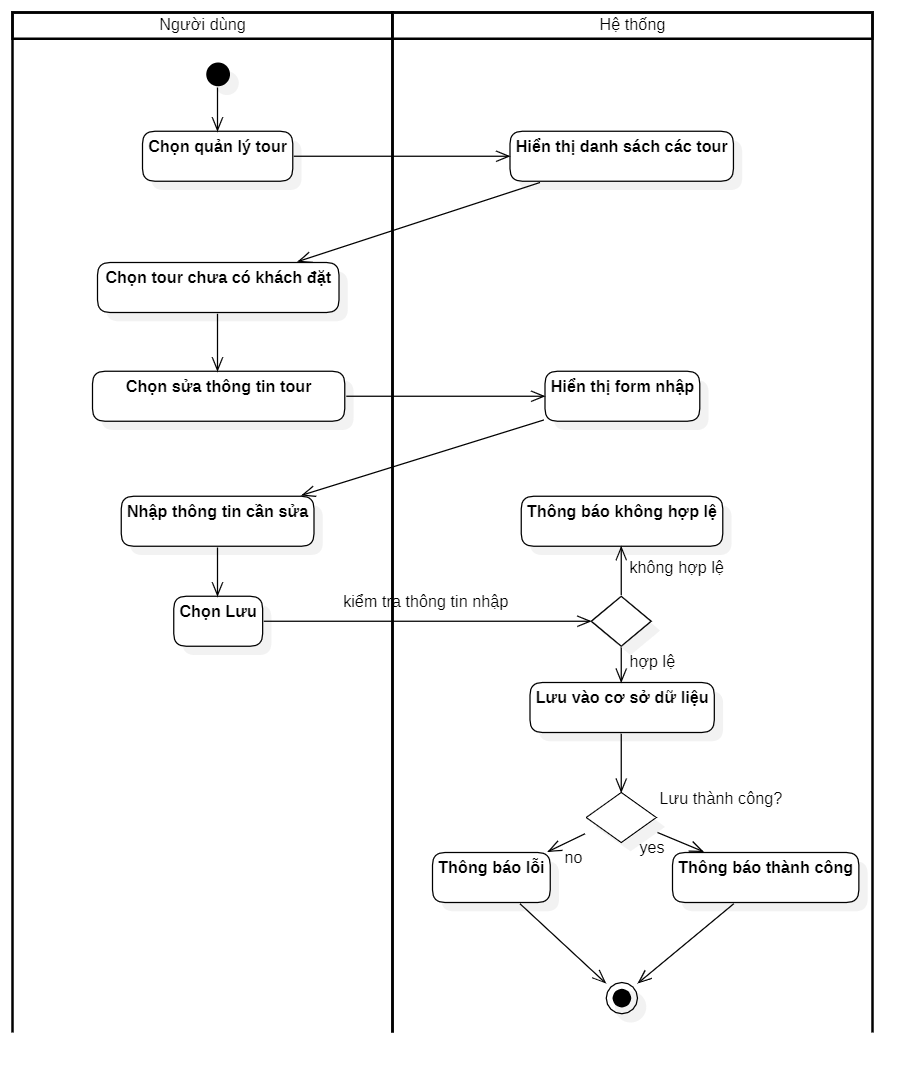
Hoạt động quản lý tour.

* Thêm mới tour

****

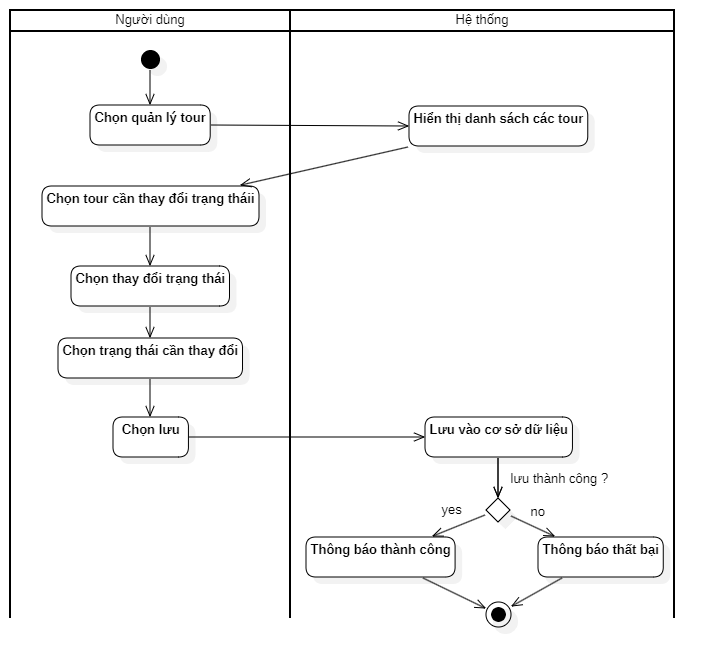
Hình 3. . Sơ đồ hoạt động thêm mới tour.

* Sửa thông tin tour

****

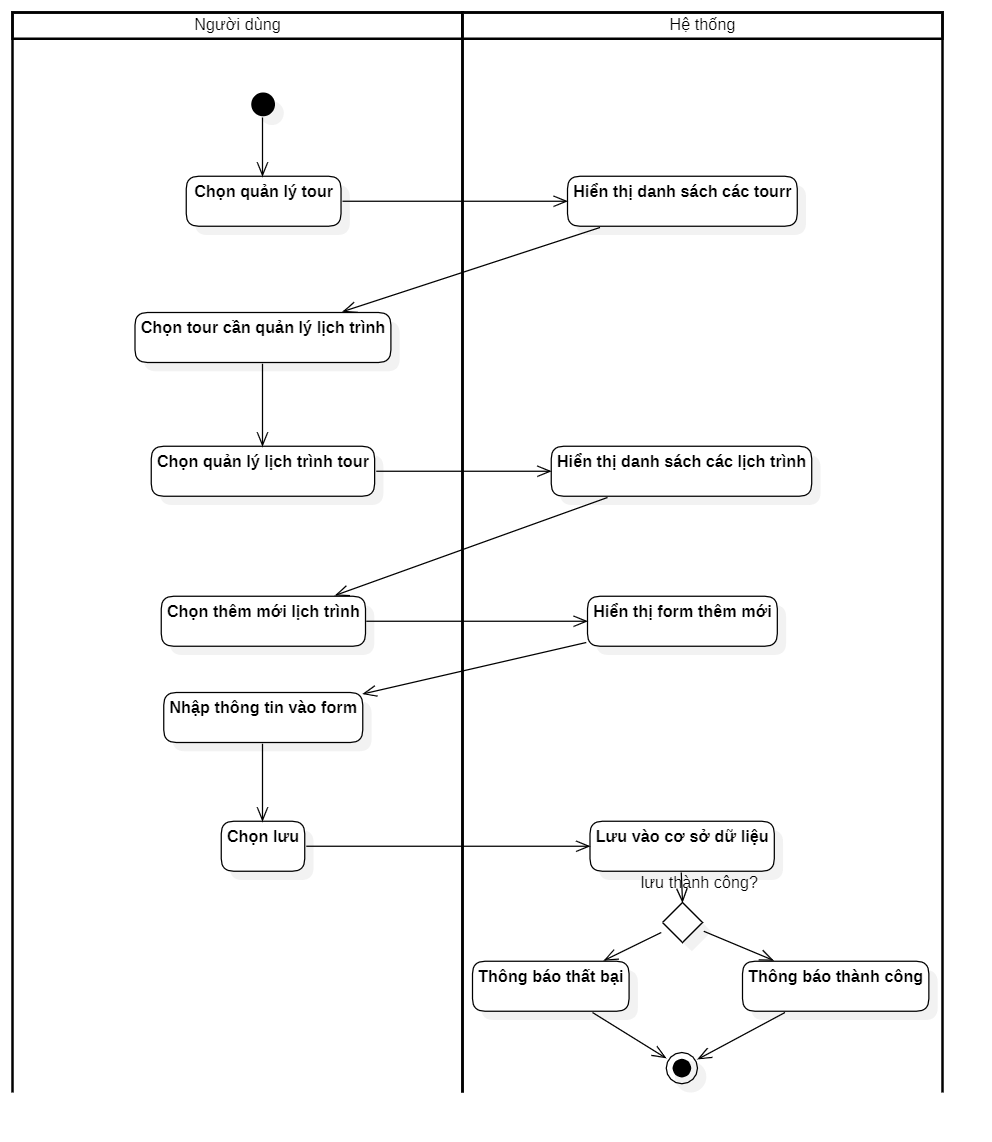
Hình 3. . Sơ đồ hoạt động sửa thông tin tour.

* Thay đổi trạng thái tour



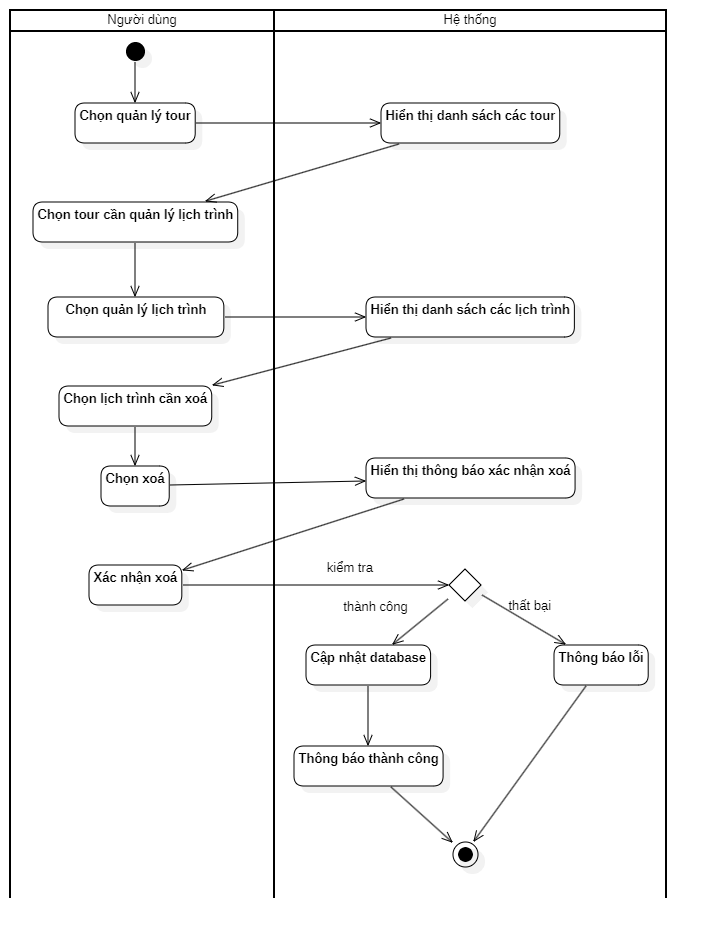
Hình 3. . Sơ đồ hoạt động thay đổi trạng thái tour

* Thêm mới lịch trình tour

****

Hình 3. . Sơ đồ hoạt động thêm mới lịch trình tour

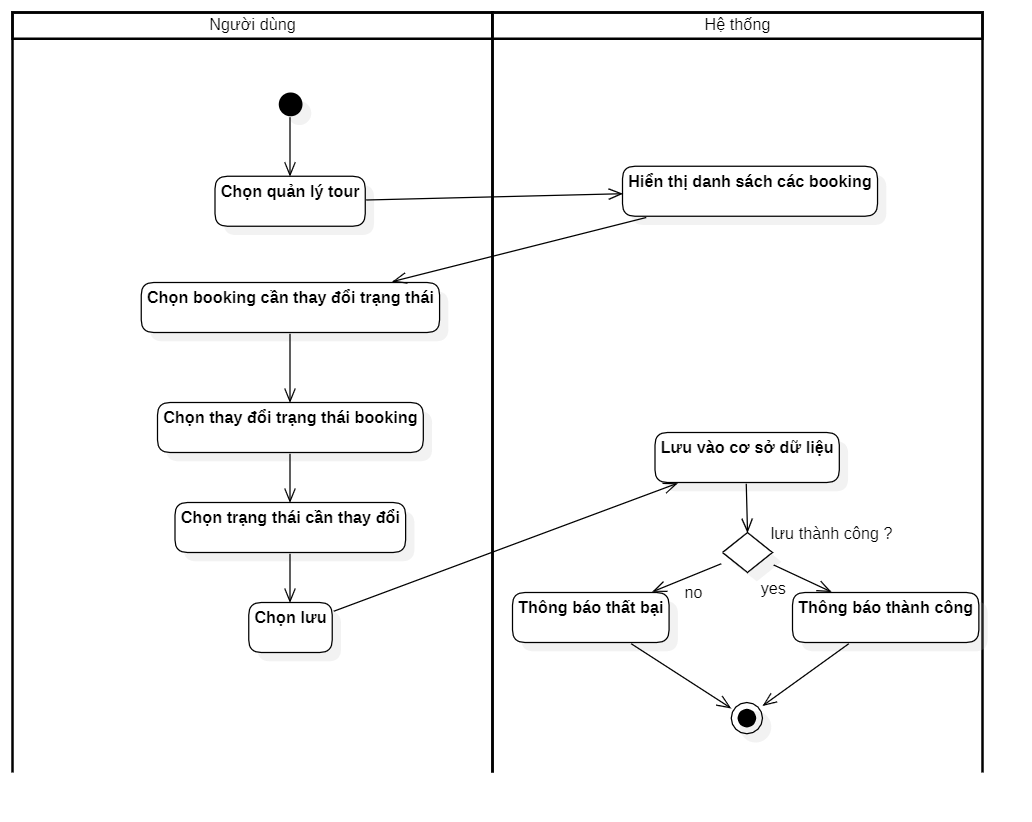
* Xoá lịch trình tour



Hình 3. . Sơ đồ hoạt động xoá lịch trình tour

Hoạt động quản lý các đơn đặt tour

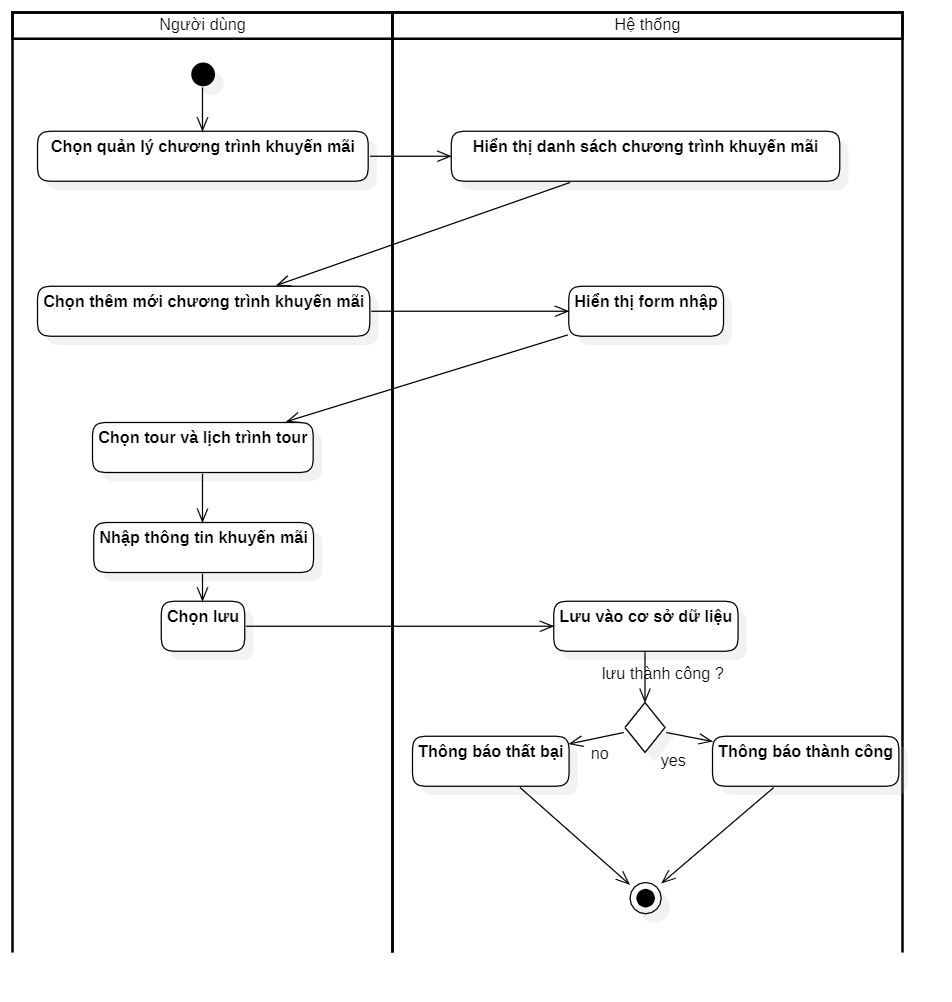
* Cập nhật trạng thái

****

Hình 3. . Sơ đồ hoạt động thay đổi trạng thái đơn đặt tour

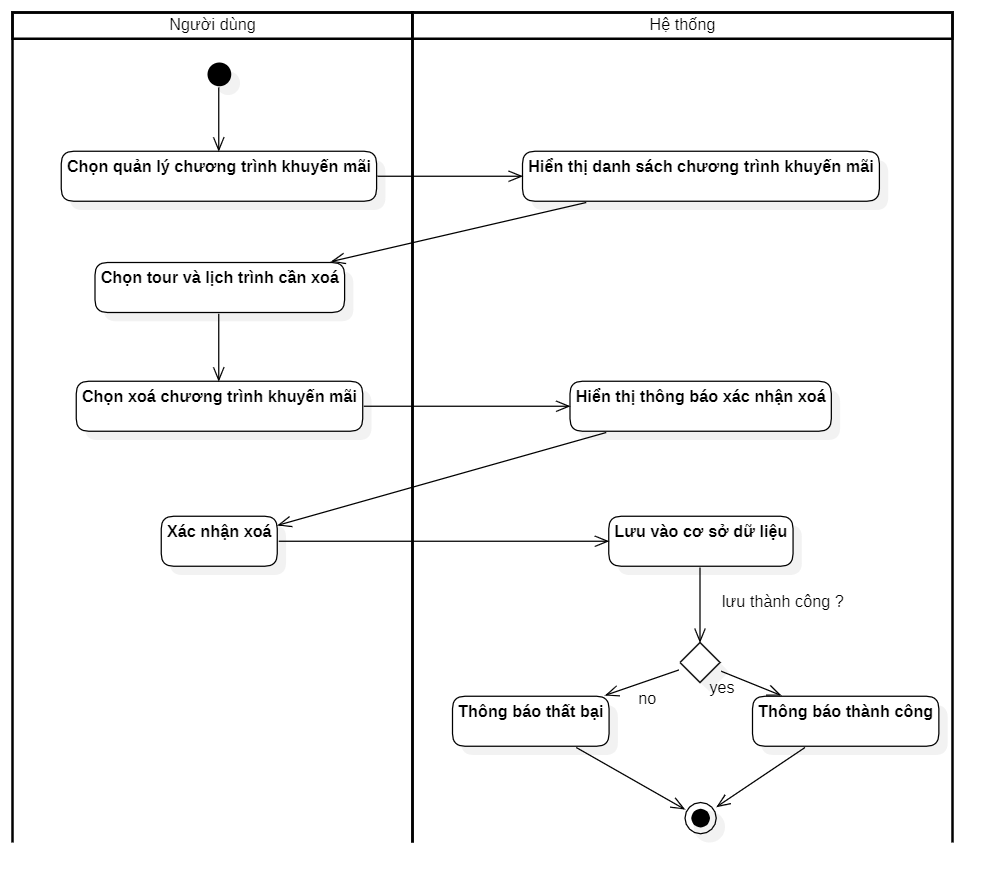
Hoạt động quản lý chương trình khuyến mãi

* Thêm mới chương trình khuyến mãi cho tour

****

Hình 3. . Sơ đồ hoạt động thêm mới chương trình khuyến mãi cho tour.

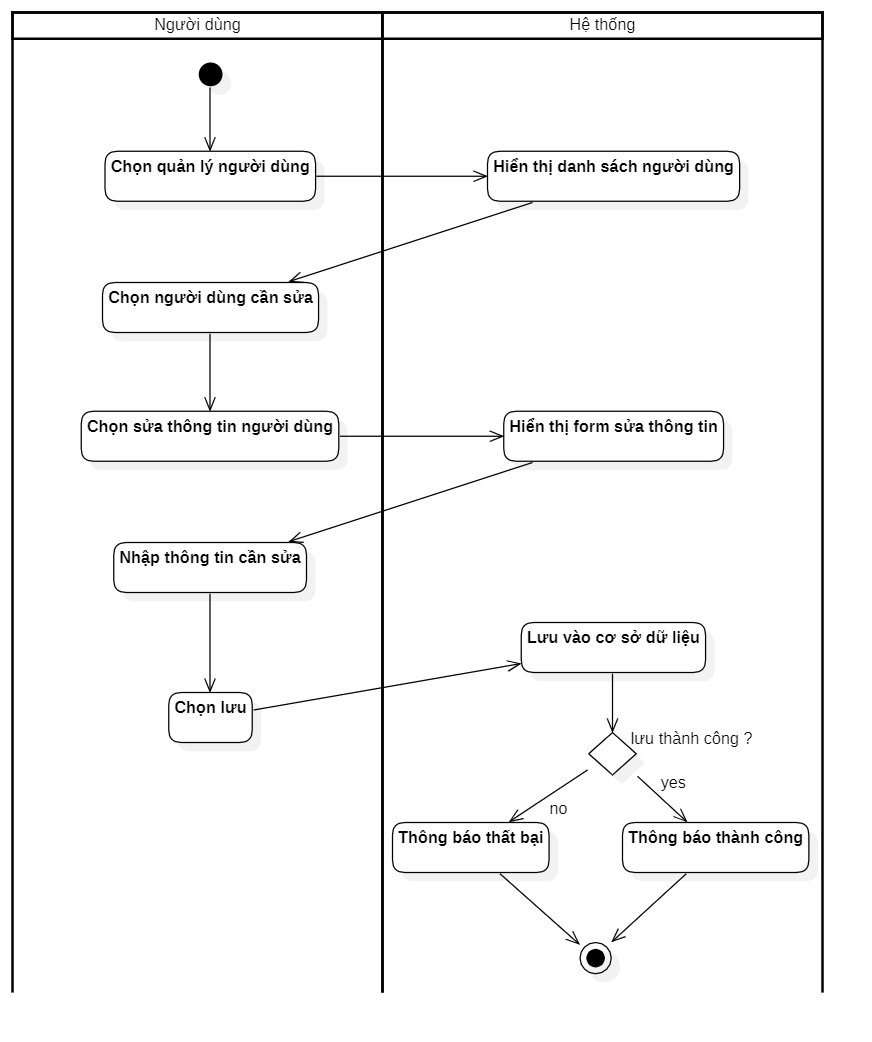
* Xoá chương trình khuyến mãi

****

Hình 3. . Sơ đồ hoạt động xoá chương trình khuyến mãi.

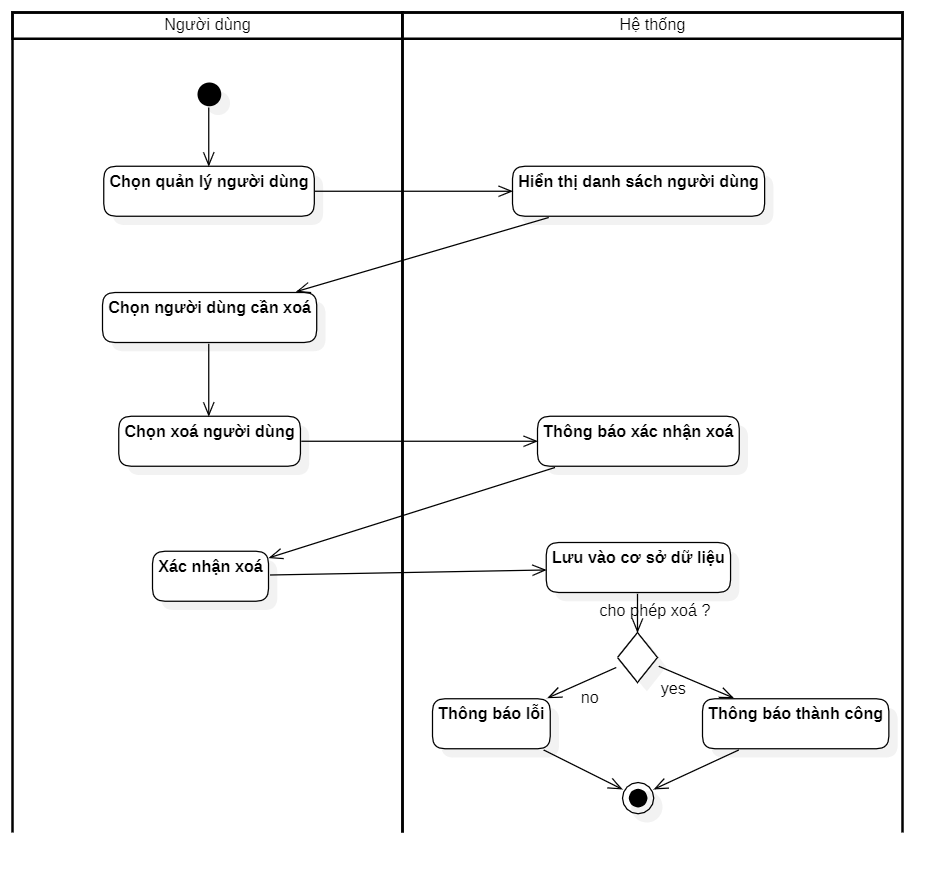
Hoạt động quản lý người dùng.

* Sửa thông tin người dùng

****

Hình 3. . Sơ đồ hoạt động sửa thông tin người dùng.

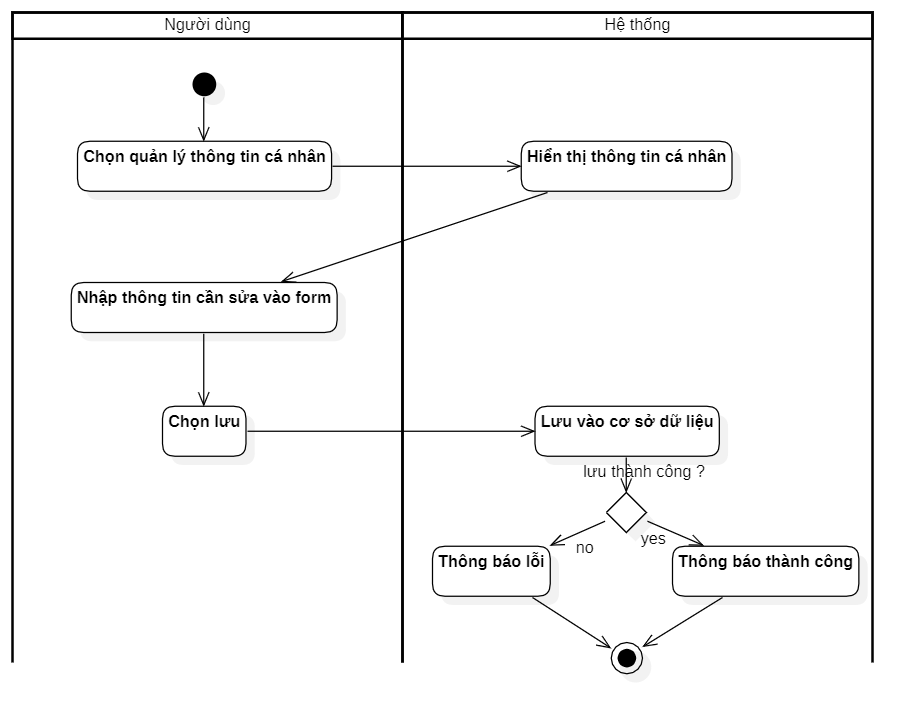
* Xoá người dùng

****

Hình 3. . Sơ đồ hoạt động xoá người dùng.

Hoạt động quản lý thông tin cá nhân.

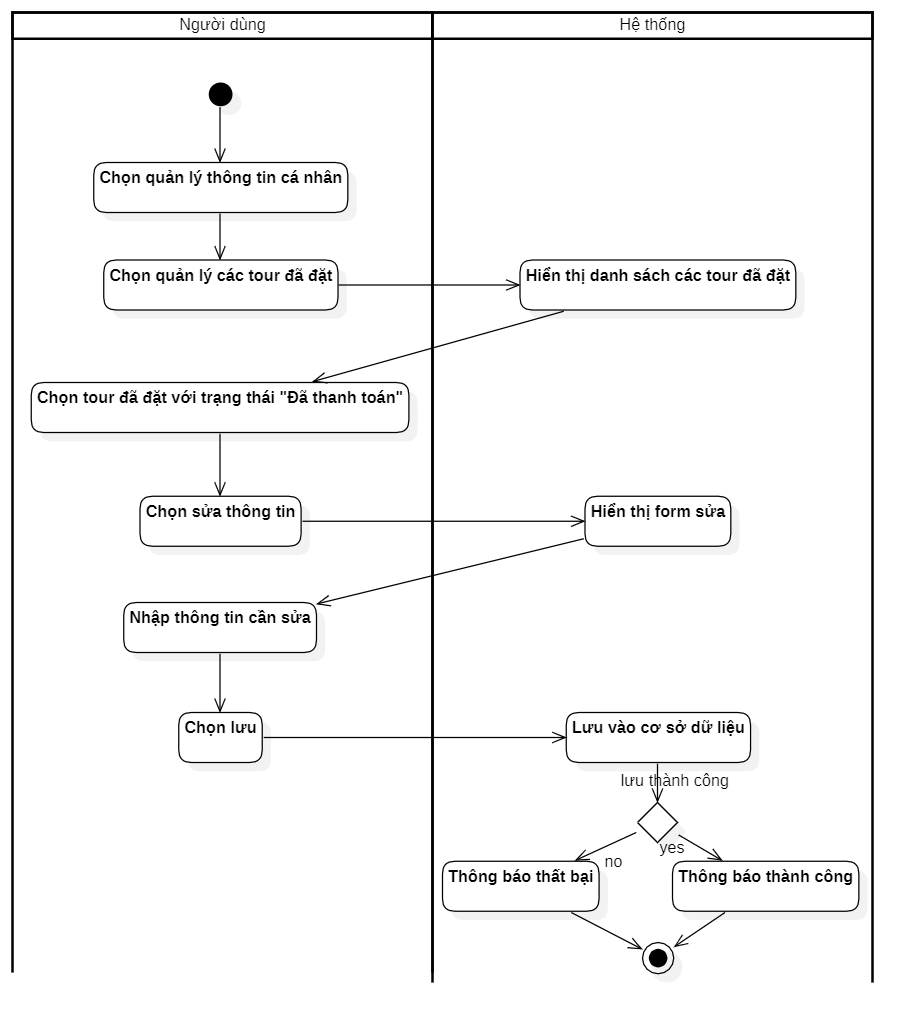
* Sửa thông tin cá nhân

****

Hình 3. . Sơ đồ hoạt động sửa thông tin cá nhân.

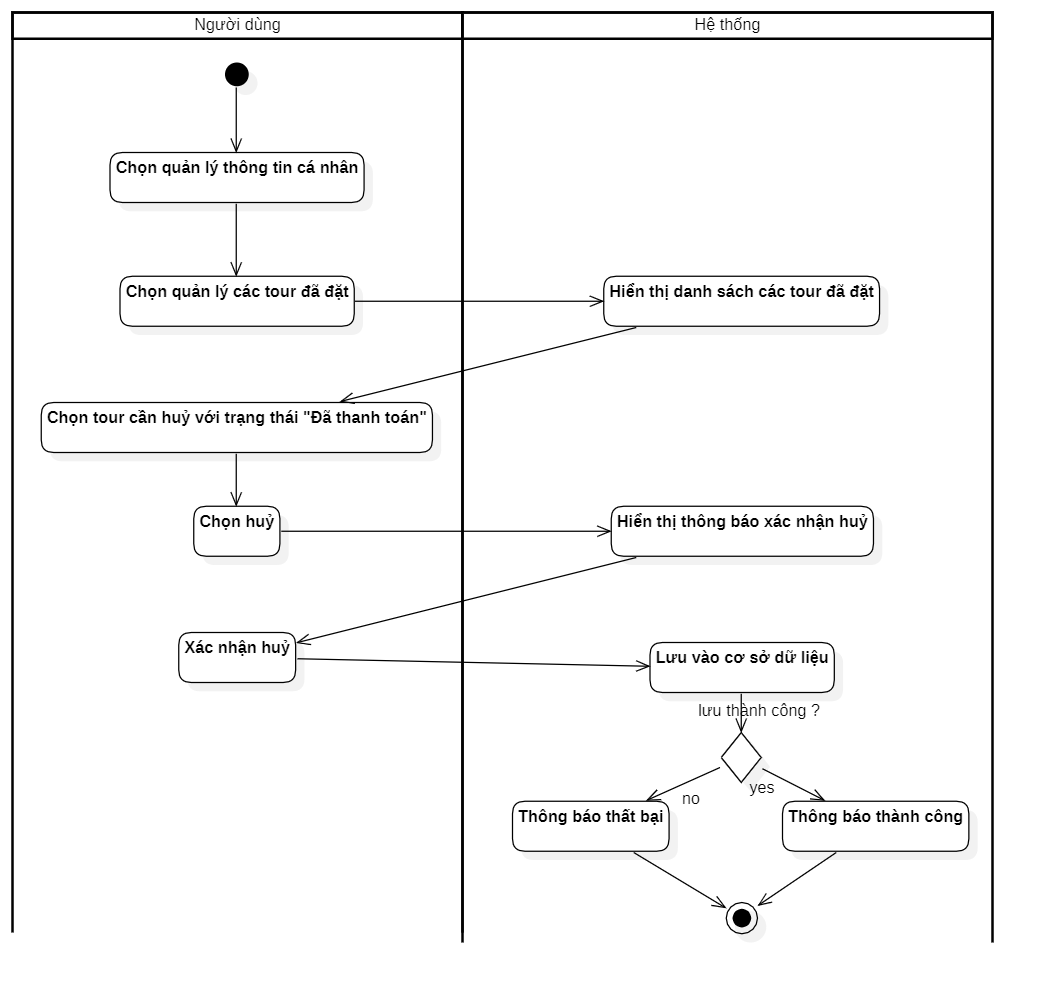
Hoạt động quản lý lịch sử đặt tour

* Chỉnh sửa thông tin tour đã đặt

****

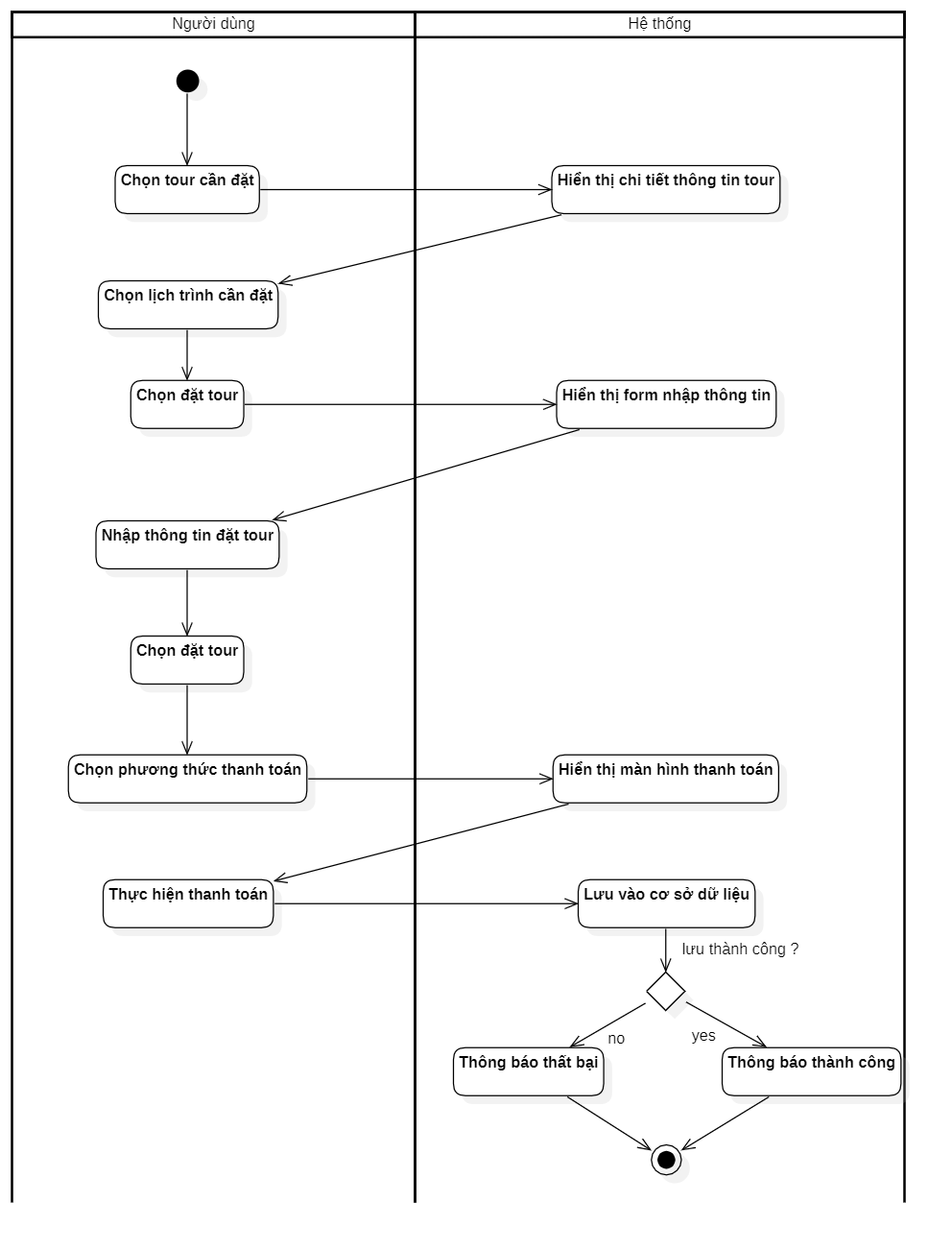
Hình 3. . Sơ đồ hoạt động chỉnh sửa thông tin tour đã đặt.

* Huỷ tour đã đặt

****

Hình 3. . Sơ đồ hoạt động huỷ tour đã đặt.

Hoạt động Đặt tour

****

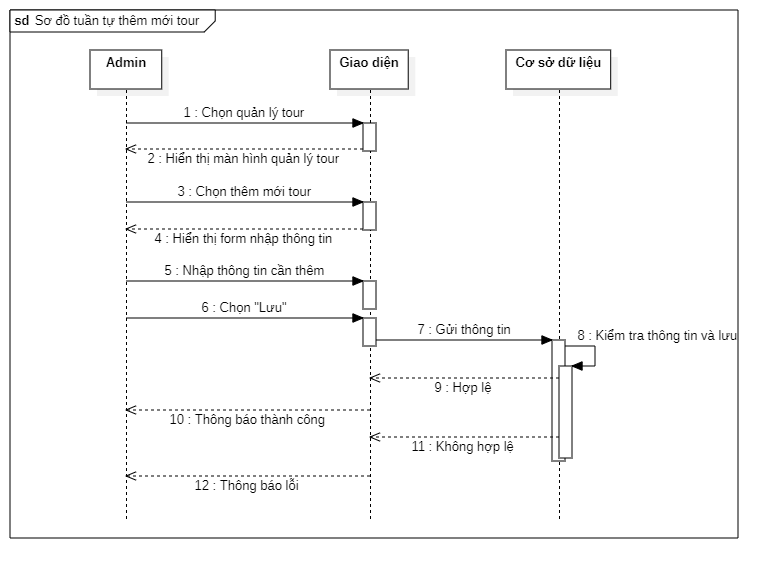
Hình 3. . Sơ đồ hoạt động đặt tour.

## Sơ đồ tuần tự

Các sơ đồ tuần tự dành cho admin

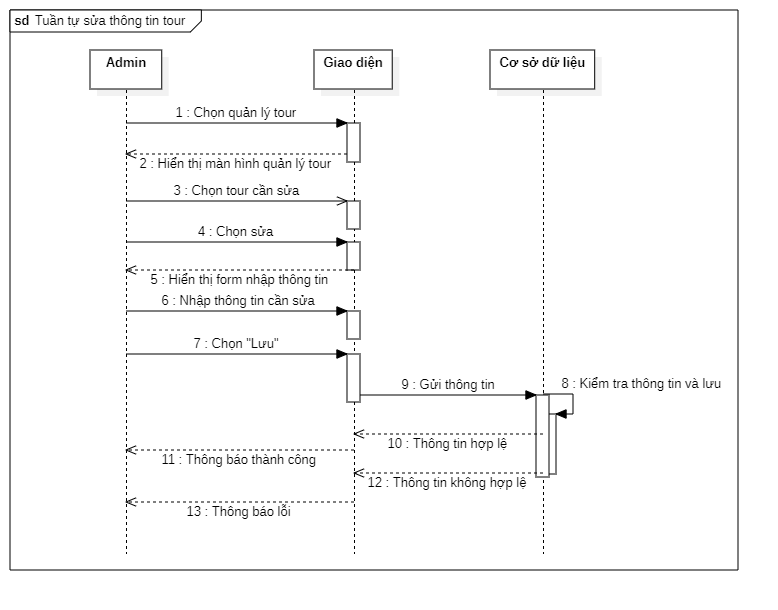
Tuần tự quản lý tour

* Thêm mới tour



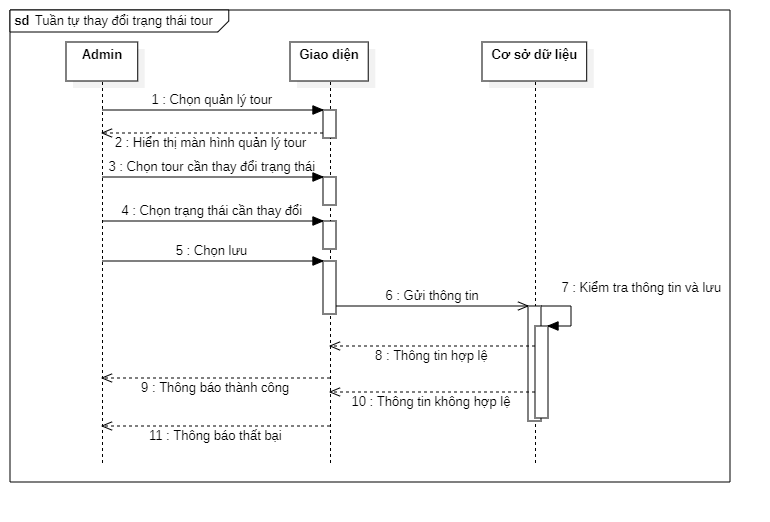
Hình 3. . Sơ đồ tuần tự thêm mới tour.

* Sửa thông tin tour



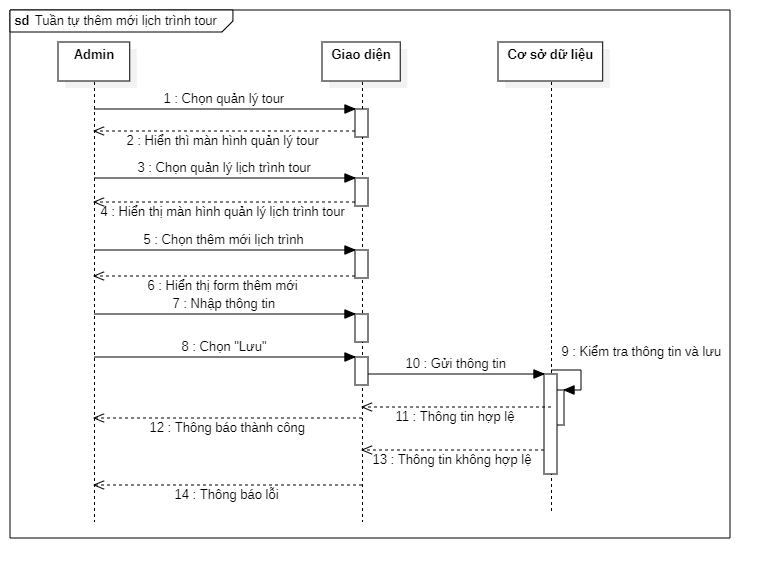
Hình 3. . Sơ đồ tuần tự sửa thông tin tour.

* Thay đổi trạng thái tour



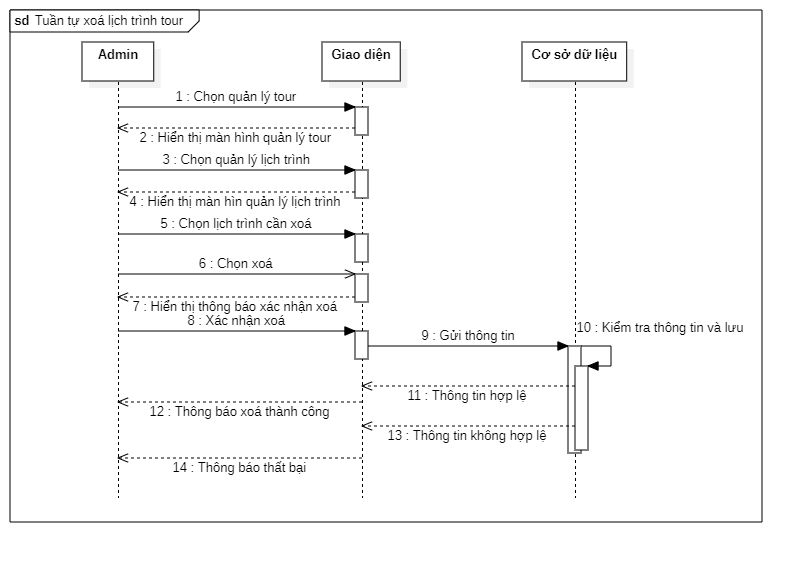
Hình 3. . Sơ đồ tuần tự thay đổi trạng thái tour

* Thêm mới lịch trình tour



Hình 3. . Sơ đồ tuần tự thêm mới lịch trình tour

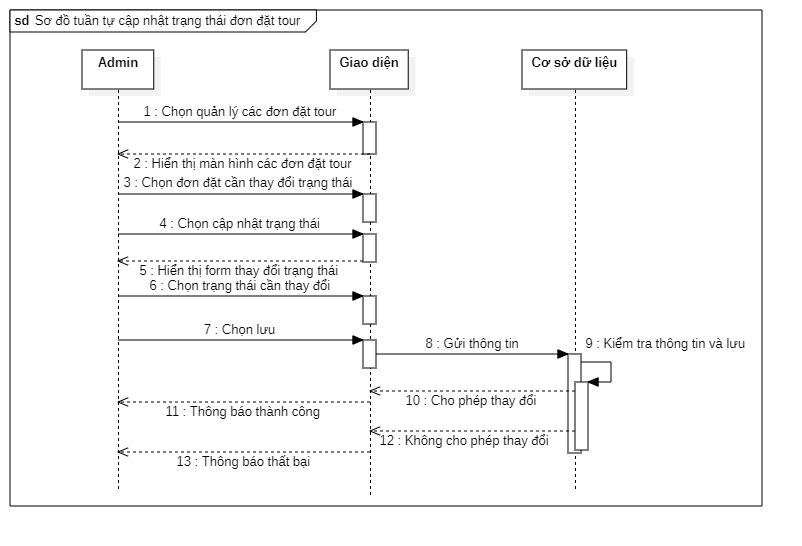
* Xoá lịch trình tour



Hình 3. . Sơ đồ tuần tự xoá lịch trình tour.

Tuần tự quản lý các đơn đặt tour

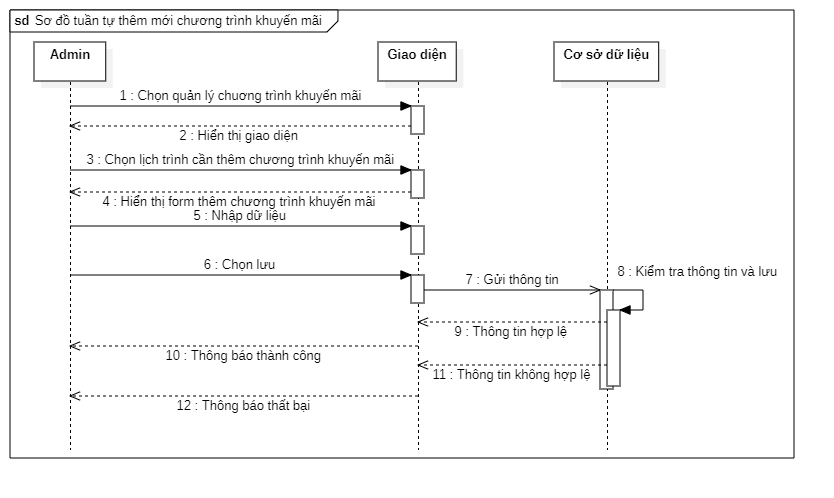
* Cập nhật trạng thái



Hình 3. . Sơ đồ tuần tự cập nhật trạng thái đơn đặt tour.

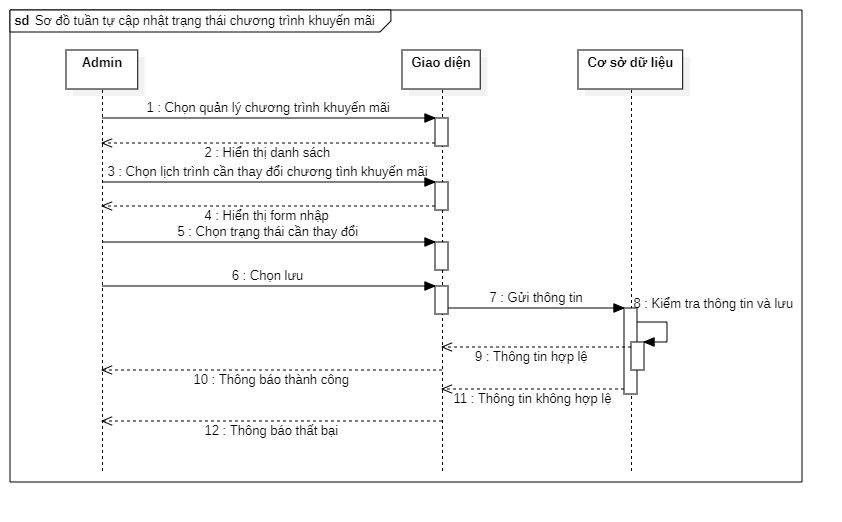
Tuần tự quản lý chương trình khuyến mãi

* Thêm mới chương trình khuyến mãi



Hình 3. . Sơ đồ tuần tự thêm mới chương trình khuyến mãi

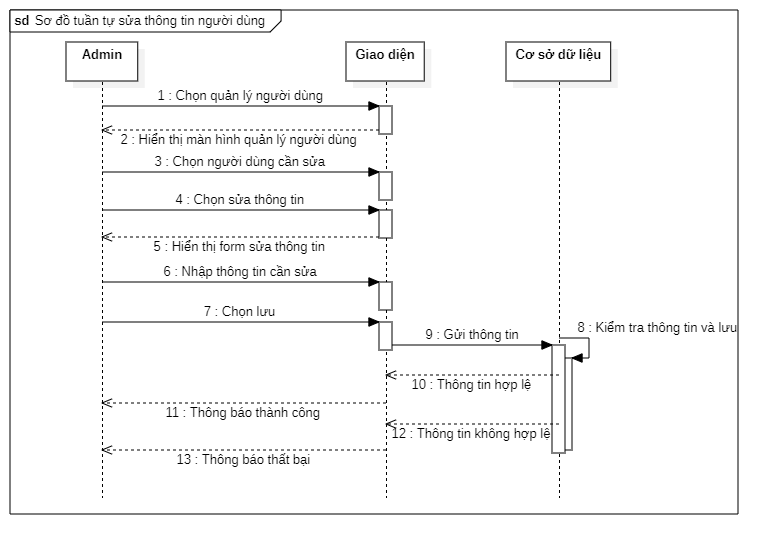
* Cập nhật trạng thái chương trình khuyến mãi



Hình 3. . Sơ đồ tuần tự cập nhật trạng thái chương trình khuyến mãi.

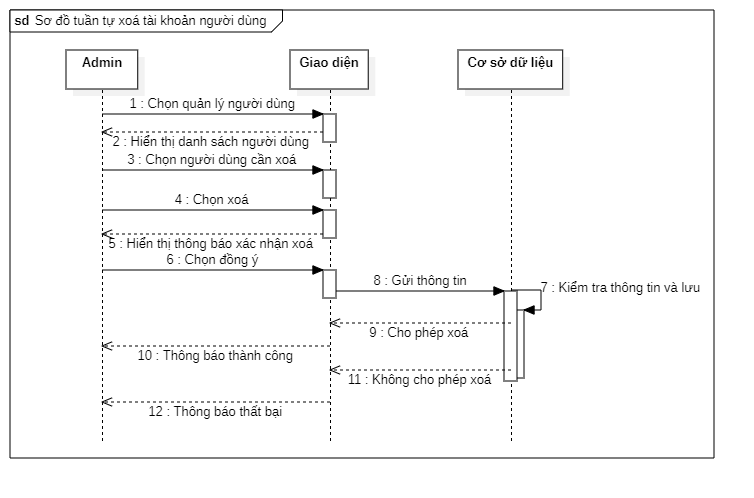
Tuần tự quản lý người dùng

* Sửa thông tin người dùng



Hình 3. . Sơ đồ tuần tự sửa thông tin người dùng.

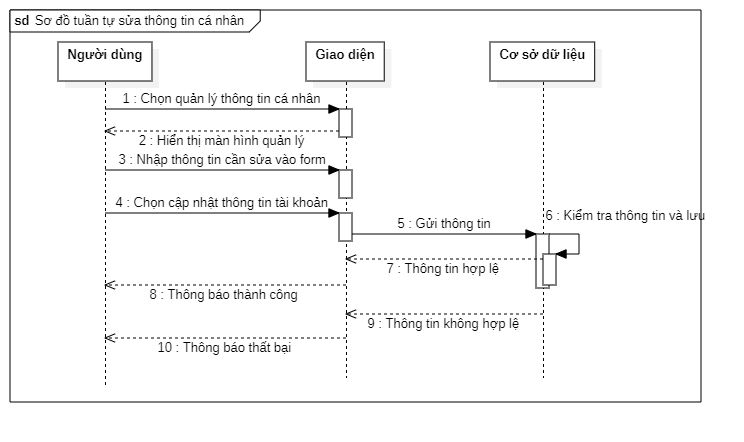
* Xoá tài khoản người dùng



Hình 3. . Sơ đồ tuần tự xoá tài khoản người dùng.

Tuần tự quản lý thông tin cá nhân

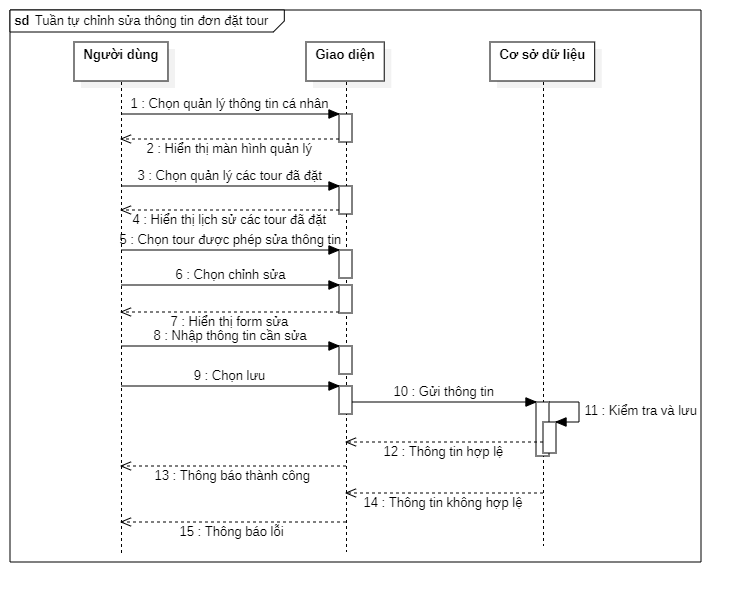
* Sửa thông tin cá nhân



Hình 3. . Sơ đồ tuần tự sửa thông tin cá nhân.

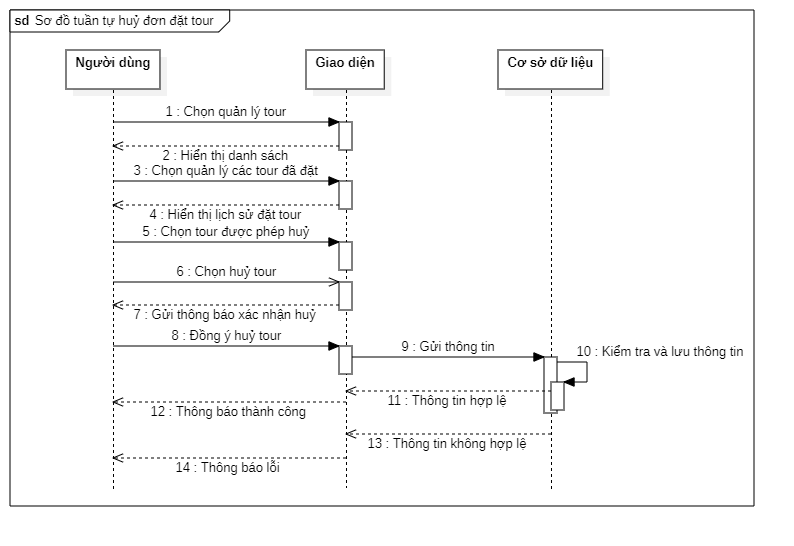
Tuần tự quản lý lịch sử đặt tour

* Chỉnh sửa thông tin đơn đặt



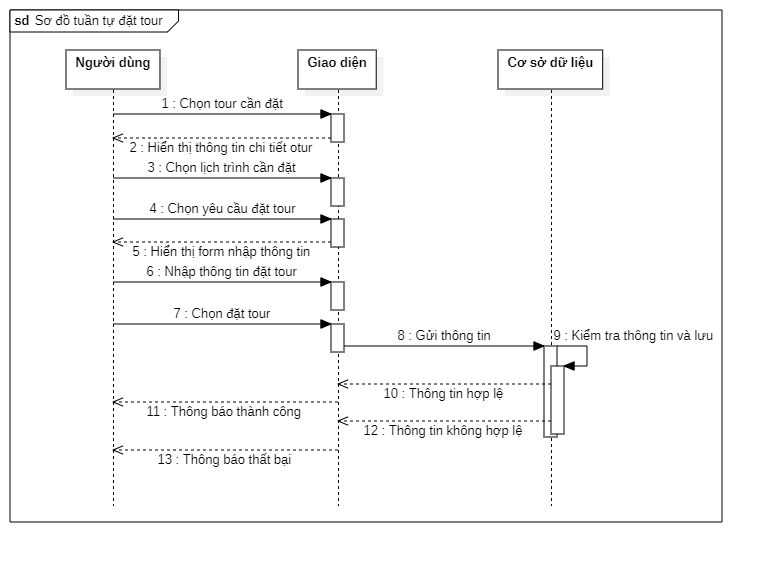
Hình 3. . Sơ đồ tuần tự chỉnh sửa thông tin đơn đặt tour

* Huỷ tour



Hình 3. . Sơ đồ tuần tự huỷ tour

Tuần tự đặt tour



Hình 3. . Sơ đồ tuần tự đặt tour

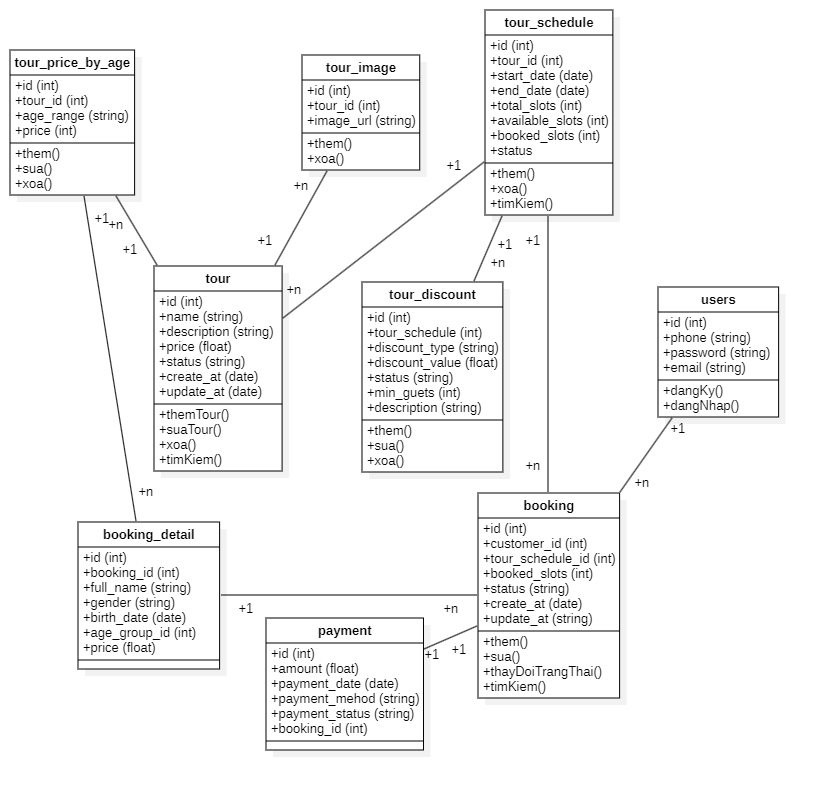
A screenshot of a computer

Description automatically generated

Hình 3. . Sơ đồ tuần tự đặt tour.

## Sơ đồ lớp

Sơ đồ lớp



Hình 3. . Sơ đồ lớp

Lược đồ cơ sở dữ liệu quan hệ

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 3. . Lược đồ cơ sở dữ liệu quan hệ.

**Mô tả các bảng**

* Bảng users

|  |  |
| --- | --- |
| Định nghĩa | Thuộc tính |
| Lưu trữ thông tin của người dùng | * Id: Khóa chính, định danh duy nhất cho mỗi người dùng * Username: Tên đăng nhập của người dùng * Password: Mật khẩu (đã mã hóa) của người dùng * Email: Địa chỉ email của người dùng * Phone: Số điện thoại * Role\_id: Liên kết đến bảng roles – xác định quyền của người dùng (admin, khách hàng) * Status: Trạng thái hoạt động: active, inactive * Created\_at: Thời gian tạo tài khoản. * Update\_at: Thời gian cập nhật thông tin tài khoản gần nhất |

* Bảng roles

|  |  |
| --- | --- |
| Định nghĩa | Thuộc tính |
| Vai trò người dùng | * Id: Khoá chính * Name: Tên quyền (Admin, Customer) * Description: Mô tả chi tiết về vai trò * Create\_at: Thời gian tạo * Update\_at: Thời gian cập nhật gần nhất |

* Bảng tours

|  |  |
| --- | --- |
| Định nghĩa | Thuộc tính |
| Lưu trữ thông tin tour | * Id: Mã định danh tour * Name: Tên tour * Description: Mô tả chi tiết tour * Price: Giá gốc(chưa tính theo độ tuổi, chưa giảm giá) * Status: Trạng thái (active, inactive) * Created\_at: Ngày tạo tour * Updated\_at: Ngày cập nhật tour |

* Bảng tour\_price\_by\_age

|  |  |
| --- | --- |
| Định nghĩa | Thuộc tính |
| Lưu trữ giá tour theo độ tuổi | * Id: Mã giá theo độ tuổi * Tour\_id: Tham chiếu đến bảng tours * Age\_range: Khoảng độ tuổi (<2, 2-10, >10). * Price: Giá tương ứng * Created\_at: Ngày tạo * Updated\_at: Ngày cập nhật gần nhất |

* Bảng tour\_schedule

|  |  |
| --- | --- |
| Định nghĩa | Thuộc tính |
| Lưu trữ thông tin lịch trình tour | * Id: Mã lịch trình * Tour\_id: Tham chiếu đến bảng tours * Start\_date: Ngày khởi hành * End\_date: Ngày kết thúc * Total\_slots: Tổng số ghế có thể đặt * Available\_slots: Số ghế còn lại * Booked\_slots: Số ghế đã được đặt * Status: Trạng thái (available, full, canceled) * Created\_at: Ngày tạo * Updated\_at: Ngày cập nhật gần nhất |

* Bảng Booking

|  |  |
| --- | --- |
| Định nghĩa | Thuộc tính |
| Đơn đặt tour | * Id: Mã đơn đặt * Customer\_id: Người đặt (Liên kết đến bảng users) * Tour\_schedule\_id: Lịch trình đã chọn * Booked\_slots: Số ghế đặt * Status: Trạng thái (onfirmed, canceled, completed) * Created\_at: Ngày đặt * Updated\_at: Ngày cập nhật đơn * Note: ghi chú |

* Bảng booking\_detail

|  |  |
| --- | --- |
| Định nghĩa | Thuộc tính |
| Lưu trữ thông tin chi tiết hành khách | * Id: Mã chi tiết * Booking\_id: Đơn đặt tương ứng * Full\_name: Người đi (khác người đặt) * Gender: Giới tính * Birth\_date: Sinh nhật * Age\_group\_id: Nhóm tuổi * Price: Giá tiền tính theo nhóm tuổi tại thời điểm đặt |

* Bảng payment

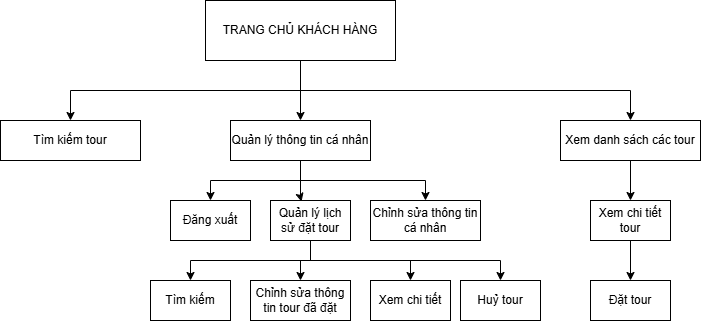
|  |  |
| --- | --- |
| Định nghĩa | Thuộc tính |
| Lưu trữ thông tin chi tiết thanh toán | * Id: Mã thanh toán * Booking\_id: Liên kết đơn đặt * Amount: Tổng số tiền đã thanh toán * Payment\_date: Ngày thanh toán * Payment\_method: Phương thức thanh toán * Payment\_status: Trạng thái thanh toán * Created\_at: Ngày tạo bảng ghi * Update\_at: Ngày cập nhật bảng ghi |

* Bảng tour\_discount

|  |  |
| --- | --- |
| Định nghĩa | Thuộc tính |
| Lưu thông tin tour ưu đãi | * Id: Mã ưu đãi * Tour\_id: Tour được áp dụng ưu đãi * Discount\_type: Percentage (giảm %) hoặc fixed (giảm tiền) * Discount\_value: Giá trị giảm * Status: active, expired * Create\_at: ngày tạo * Update\_at: ngày cập nhật gần nhất * Min\_guests: số khách tối thiểu để áp dụng khuyến mãi * Description: Mô tả |

# WEBSITE ĐẶT TOUR DU LỊCH CÁC ĐỊA ĐIỂM VĂN HOÁ VÀ LÀNG NGHỀ TẠI TỈNH PHÚ YÊN

**Sơ đồ phân cấp trang web – Vai trò: Khách hàng**

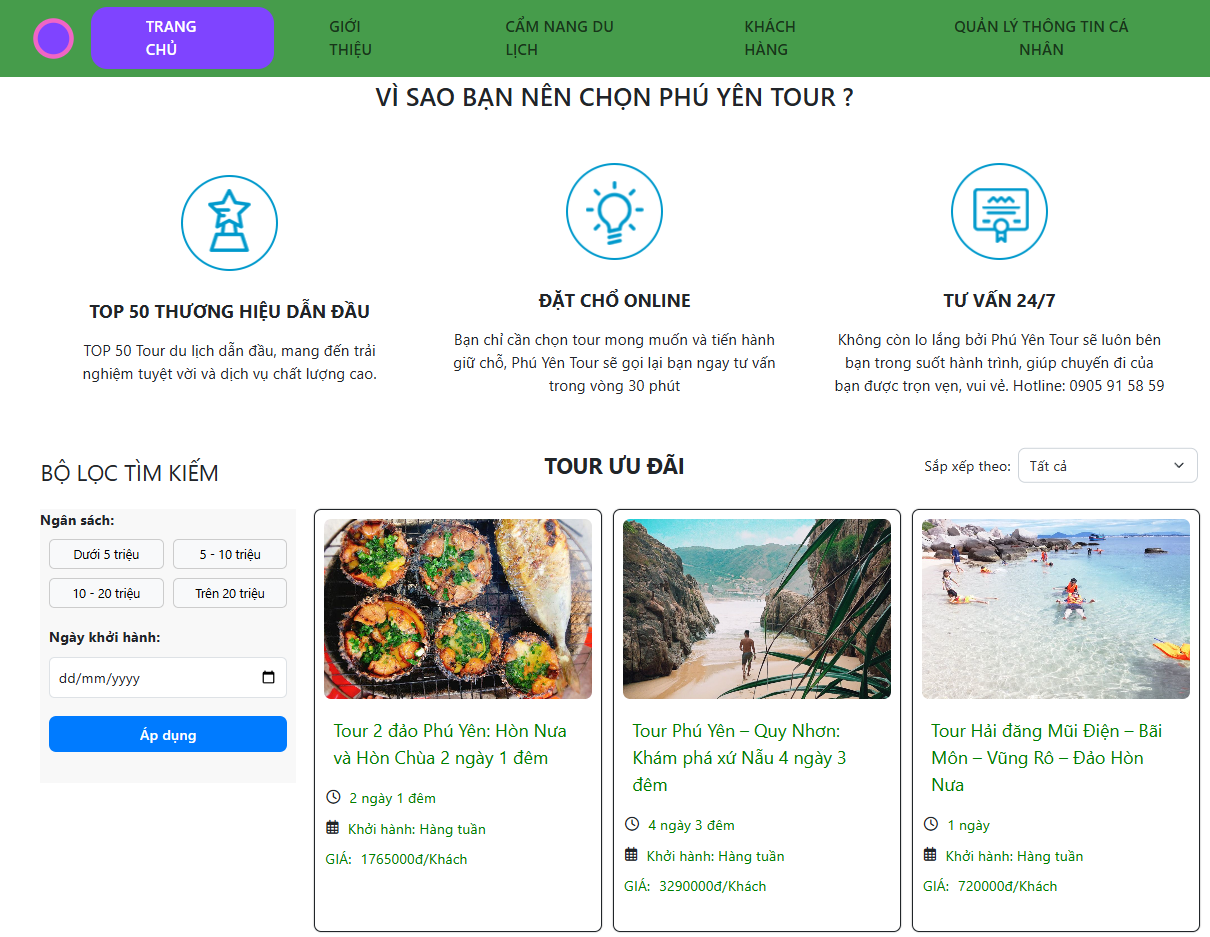


Hình 4. . Sơ đồ phân cấp trang web - vai trò: khách hàng

## Giao diện trang chủ

Trang chủ là nơi hiển thị tổng quan hệ thống đặt tour, giúp người dùng dễ dàng khám phá các tour nổi bật, tour khuyến mãi và bắt đầu hành trình tìm kiếm tour phù hợp. Giao diện được thiết kế trực quan với danh sách các tour du lịch hấp dẫn được trình bày dưới dạng thẻ, đi kèm hình ảnh, tên tour, giá và ngày khởi hành. Bên cạnh đó, trang chủ tích hợp một bộ lọc tìm kiếm thông minh, cho phép khách hàng dễ dàng lọc và lựa chọn tour theo các tiêu chí cá nhân như:

* Ngân sách (ví dụ: dưới 5 triệu đồng),
* Ngày khởi hành cụ thể,
* Sắp xếp theo giá tăng/giảm hoặc độ phổ biến.



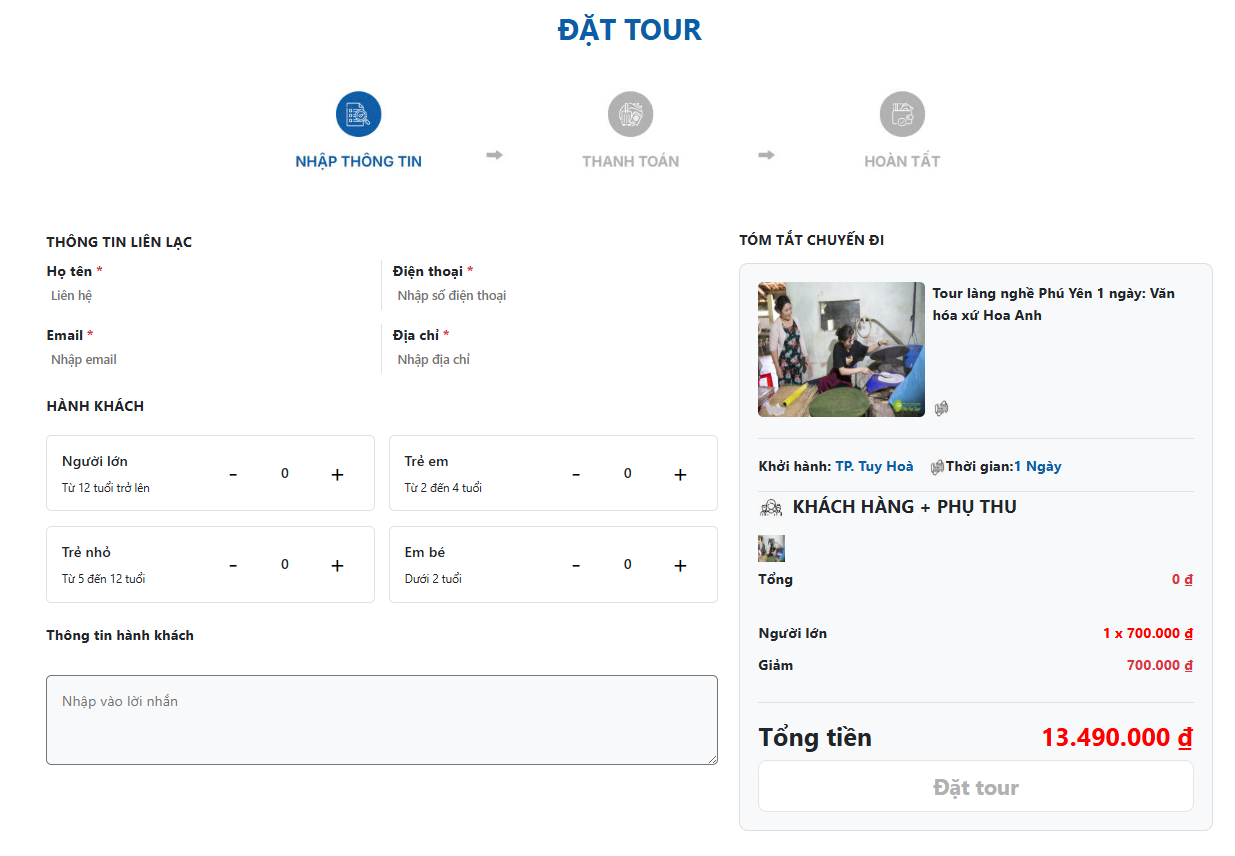
Hình 4. . Giao diện trang chủ

## Giao diện xem chi tiết và đặt tour

Giao diện này là bước đầu tiên trong quy trình đặt tour, xuất hiện sau khi người dùng chọn một tour cụ thể từ danh sách hoặc từ trang chi tiết của tour. Giao diện bao gồm thanh tiến trình với ba bước: Nhập thông tin, Thanh toán và Hoàn tất, trong đó bước hiện tại được làm nổi bật.

Ở bên trái, người dùng điền thông tin liên hệ như họ tên, số điện thoại, email và địa chỉ. Dưới đó là phần chọn số lượng hành khách, được chia theo độ tuổi: người lớn, trẻ em, trẻ nhỏ và em bé. Một ô ghi chú cũng được cung cấp để người dùng thêm yêu cầu đặc biệt.

Bên phải là phần tóm tắt chuyến đi, hiển thị tên tour, hình ảnh, thời gian, điểm khởi hành và bảng tính chi phí theo số lượng hành khách. Tổng số tiền được làm nổi bật rõ ràng. Nút “Đặt tour” nằm bên dưới, cho phép người dùng tiếp tục sang bước thanh toán nếu đã nhập đầy đủ thông tin.



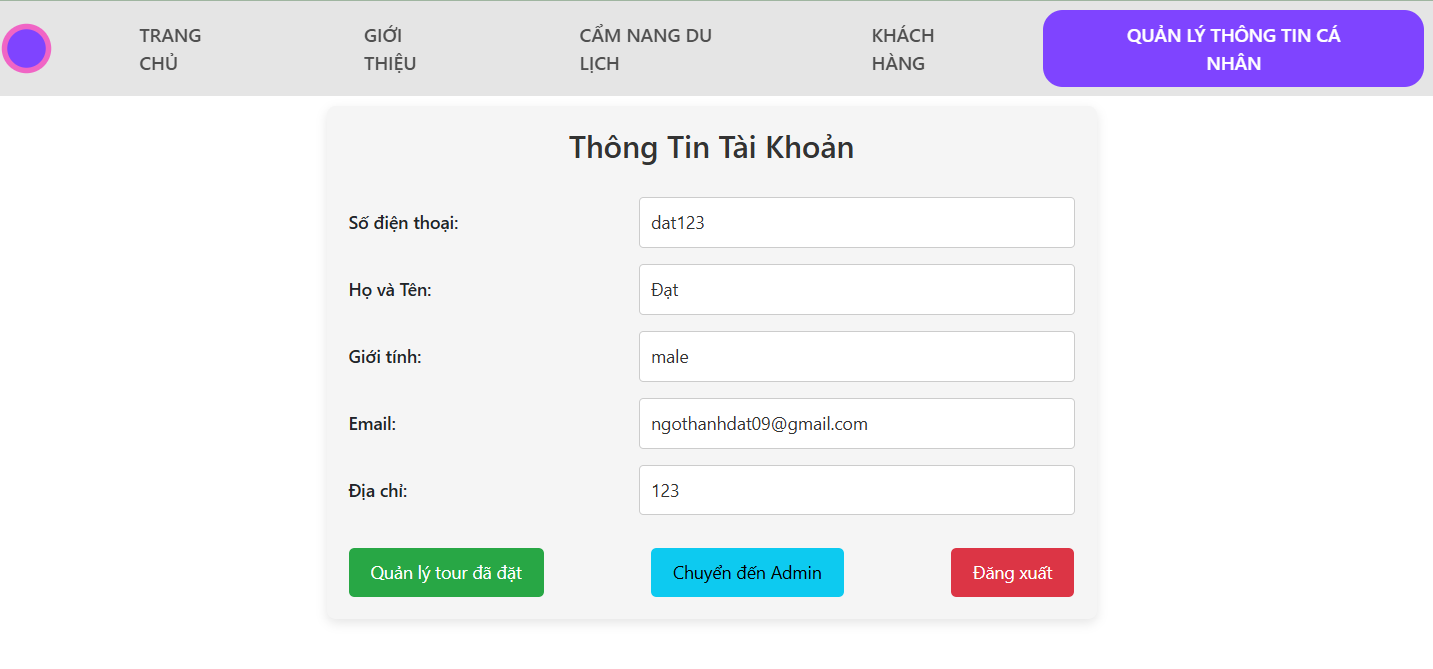
Hình 4. . Giao diện đặt tour

## Giao diện quản lý thông tin cá nhân

Đây là giao diện quản lý thông tin cá nhân, cho phép người dùng xem và chỉnh sửa các thông tin cơ bản đã đăng ký với hệ thống.

Phần nội dung trung tâm hiển thị biểu mẫu bao gồm các trường: số điện thoại, họ và tên, giới tính, email và địa chỉ. Tất cả các trường đều có thể chỉnh sửa trực tiếp, giúp người dùng dễ dàng cập nhật thông tin khi cần thiết.

Bên dưới là ba nút chức năng: “Quản lý tour đã đặt” để chuyển đến danh sách các tour mà người dùng đã đăng ký; “Cập nhật thông tin tài khoản” để lưu lại các thay đổi; và nút đỏ “Đăng xuất” để thoát khỏi tài khoản hiện tại.



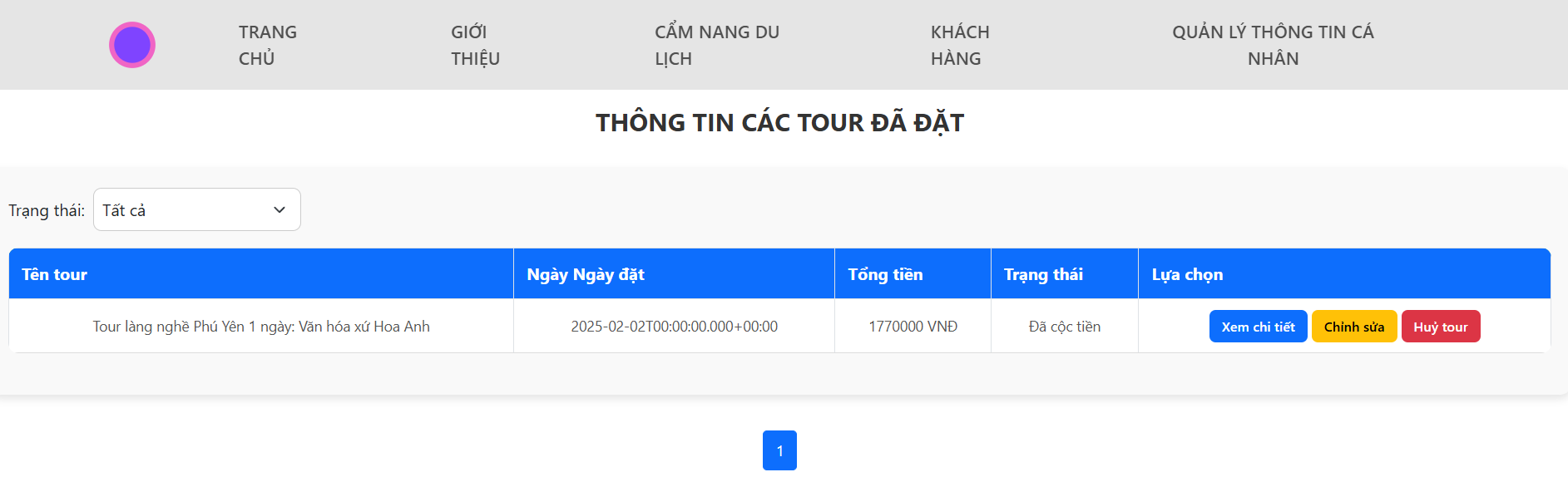
Hình 4. . Giao diện quản lý thông tin cá nhân.

## Giao diện quản lý lịch sử đặt tour

Đây là giao diện quản lý các tour đã đặt, nơi người dùng có thể theo dõi lịch sử đặt tour và thực hiện các hành động liên quan.

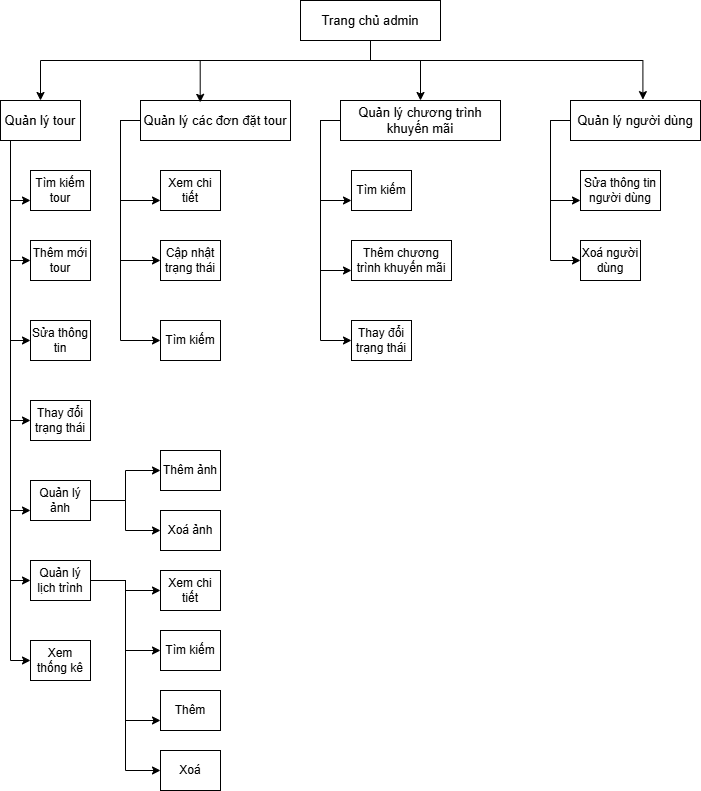
Phía trên bảng dữ liệu là bộ lọc trạng thái, cho phép người dùng lọc các tour theo trạng thái như “Tất cả”, “Đã xác nhận”, “Đã thanh toán”, “Đã hoàn thành”, hoặc “Đã huỷ”. Bên dưới là bảng hiển thị thông tin chi tiết của từng tour đã đặt, bao gồm: tên tour, ngày đặt, tổng tiền, trạng thái hiện tại và các lựa chọn hành động.

Cột “Lựa chọn” cung cấp ba nút thao tác: “Xem chi tiết” để xem thông tin cụ thể về tour đã đặt, “Chỉnh sửa” để cập nhật thông tin đặt tour, và “Huỷ tour” nếu người dùng muốn huỷ chuyến đi.



Hình 4. . Giao diện quản lý các tour đã đặt

**Sơ đồ phân cấp trang web – Vai trò: Admin**

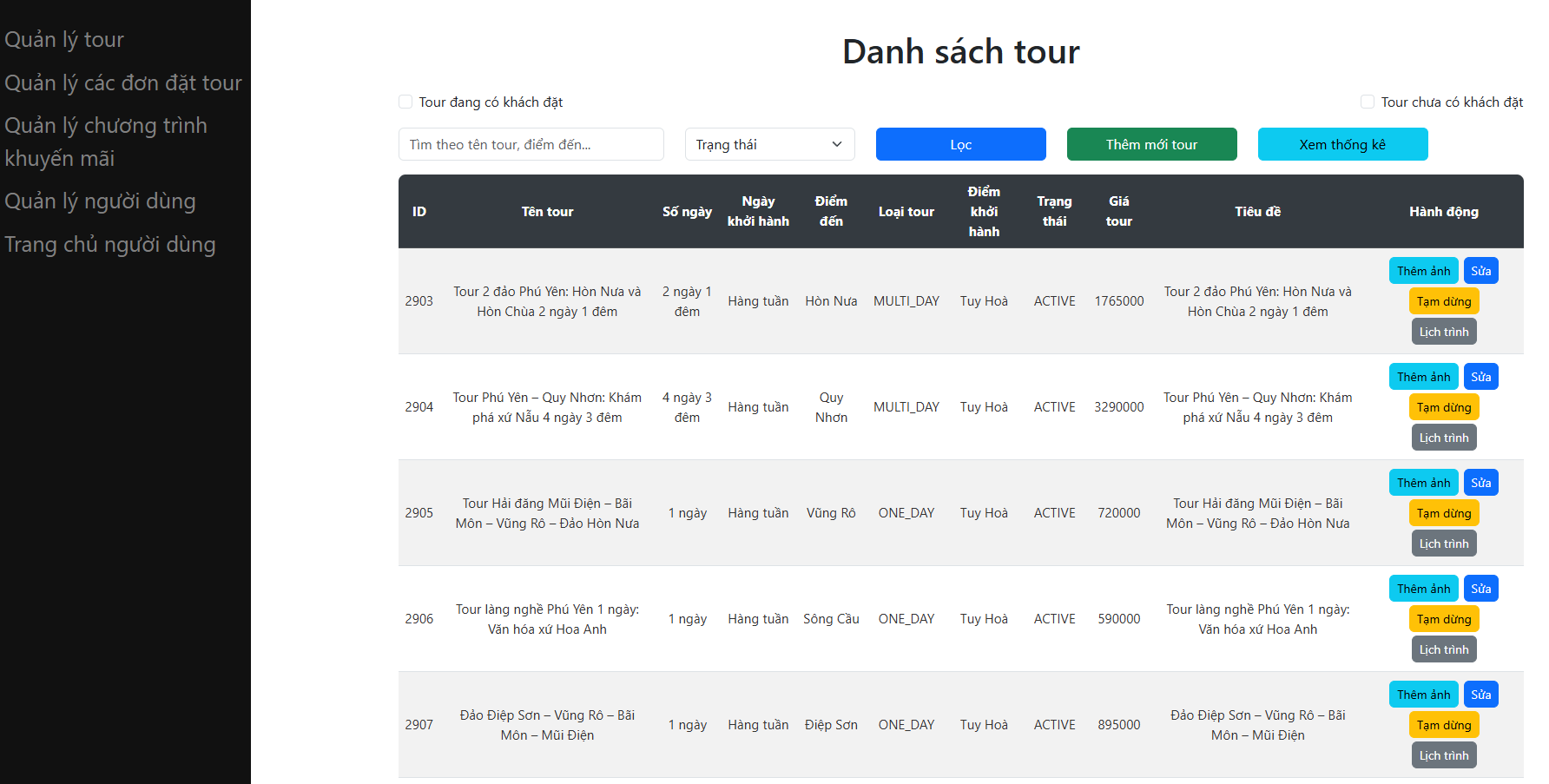
****

Hình 4. . Sơ đồ phân cấp trang web - Vai trò: Admin

## Giao diện quản lý tour

Giao diện “Danh sách tour” hiển thị danh sách các tour du lịch đang có trong hệ thống và cho phép quản trị viên dễ dàng tìm kiếm, lọc và thao tác với từng tour. Phía trên là tiêu đề trang cùng khu vực lọc, bao gồm ô tìm kiếm theo tên tour hoặc điểm đến, bộ lọc trạng thái hoạt động, checkbox lọc theo tình trạng khách đặt, và các nút “Lọc”, “Thêm mới tour” và “Xem thống kê”. Trong đó, “Xem thống kê” giúp truy cập nhanh vào báo cáo tổng hợp các số liệu liên quan đến tour.

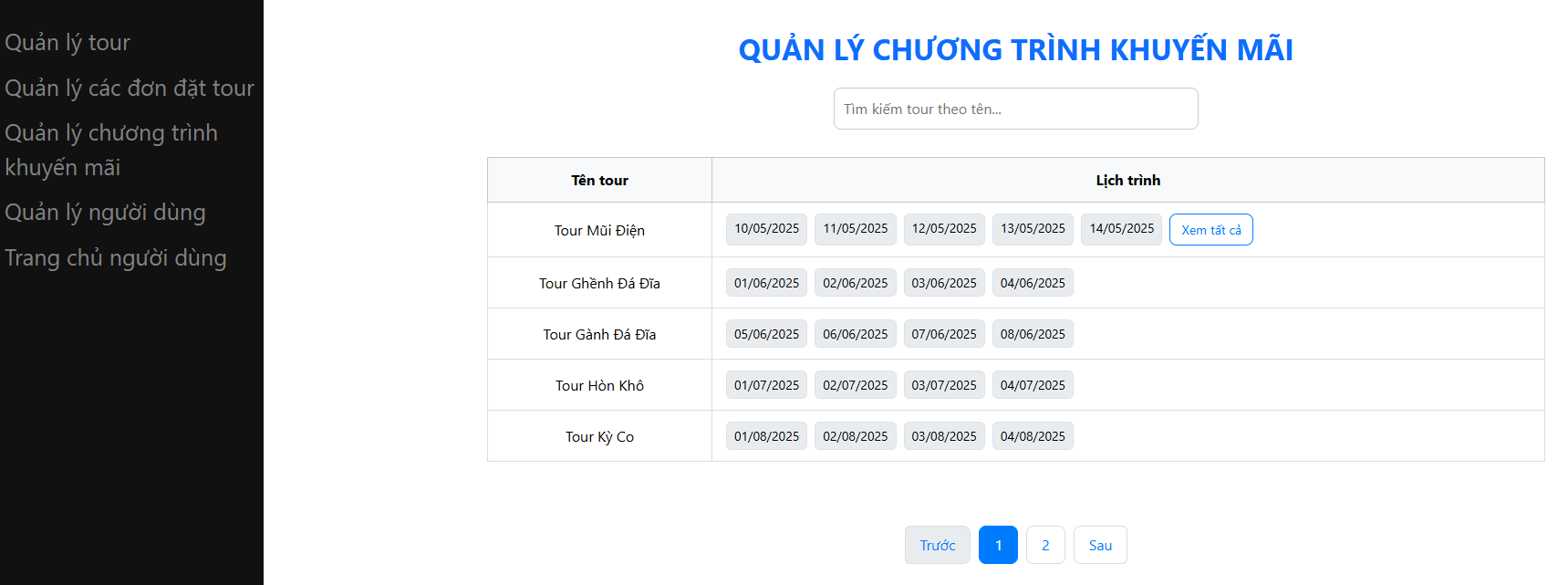
Bên dưới là bảng danh sách tour được trình bày theo dạng lưới, hiển thị các thông tin như ID, tên tour, số ngày, điểm đến, giá tour, trạng thái và tiêu đề mô tả. Mỗi dòng tương ứng với một tour cụ thể và đi kèm các nút thao tác như “Thêm ảnh”, “Sửa”, “Tạm dừng” và “Lịch trình”.



Hình 4. . Giao diện quản lý tour

## Giao diện quản lý lịch trình tour

Giao diện “Quản lý lịch trình tour” hiển thị bảng danh sách các lịch trình đã được thiết lập, bao gồm thông tin về ngày khởi hành, ngày kết thúc, tổng số ghế, số ghế còn lại và trạng thái hoạt động của từng tour. Ngoài ra, mỗi dòng dữ liệu đều đi kèm với các nút hành động như “Xem chi tiết”, “Xoá” và “Cập nhật trạng thái”, giúp người quản trị dễ dàng thao tác. Màn hình còn hỗ trợ lọc theo trạng thái để tiện theo dõi các tour đang hoạt động hay đã kết thúc. Giao diện trực quan, dễ sử dụng, đóng vai trò quan trọng trong việc theo dõi tình trạng vận hành và điều phối các tour du lịch đến các địa điểm văn hoá và làng nghề tại tỉnh Phú Yên.



Hình 4. . Giao diện quản lý lịch trình tour

## Giao diện quản lý các đơn đặt tour

Giao diện “Danh sách các đơn đặt tour” là nơi quản trị viên có thể theo dõi toàn bộ thông tin liên quan đến các đơn đặt tour từ người dùng. Giao diện hiển thị bảng danh sách các đơn đặt đã được thực hiện với các cột thông tin chi tiết như: ID đơn hàng, họ và tên khách hàng, số điện thoại, tên tour đã đặt, số lượng người tham gia, tổng số tiền, trạng thái thanh toán và ghi chú từ khách hàng. Mỗi đơn đặt tour đều có hai nút chức năng: “Cập nhật trạng thái” cho phép quản trị thay đổi tình trạng thanh toán hoặc xử lý đơn, và “Xem chi tiết” để xem đầy đủ nội dung đơn hàng. Màn hình này giúp quản lý hiệu quả hoạt động đặt tour, hỗ trợ xử lý các yêu cầu khách hàng một cách nhanh chóng và chính xác.

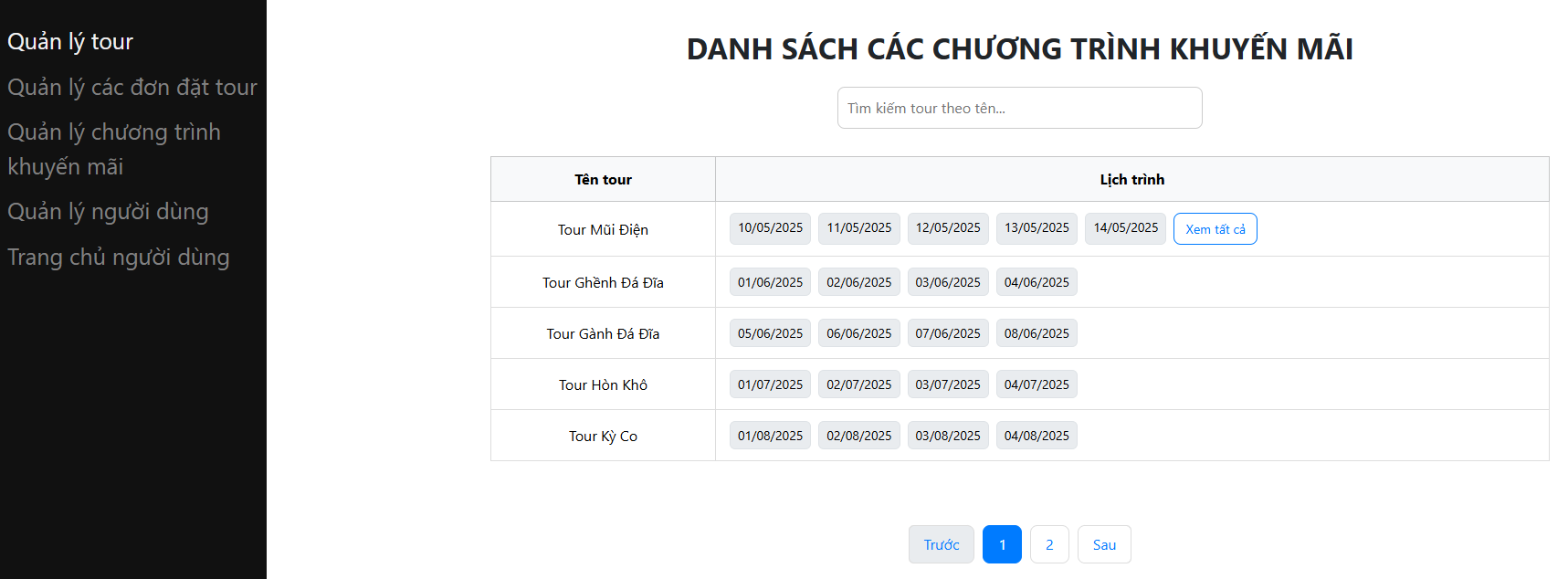


Hình 4. . Giao diện quản lý booking

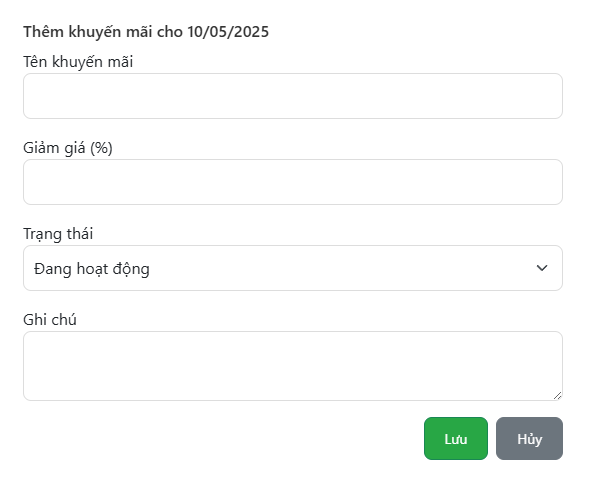
## Giao diện quản lý chương trình khuyến mãi

Giao diện “Danh sách các chương trình khuyến mãi” cho phép quản trị viên theo dõi các tour du lịch đang được áp dụng ưu đãi theo từng lịch trình cụ thể. Tại đây, hệ thống hiển thị bảng thông tin gồm hai cột chính: Tên tour và lịch trình áp dụng khuyến mãi. Với mỗi tour, người quản trị có thể xem nhanh các ngày khởi hành được ưu đãi hoặc nhấn “Xem tất cả” để mở rộng toàn bộ danh sách lịch trình tương ứng.

Khi chọn vào 1 lịch trình cụ thể sẽ hiển thị ra form nhập chương trình khuyến mãi vào lịch trình. Giao diện cũng hỗ trợ thanh tìm kiếm theo tên tour và phân trang để dễ dàng theo dõi khi có số lượng lớn dữ liệu. Màn hình này đóng vai trò quan trọng trong việc quản lý và triển khai các chiến dịch giảm giá, kích cầu du lịch đến các địa điểm văn hoá và làng nghề tại tỉnh Phú Yên.



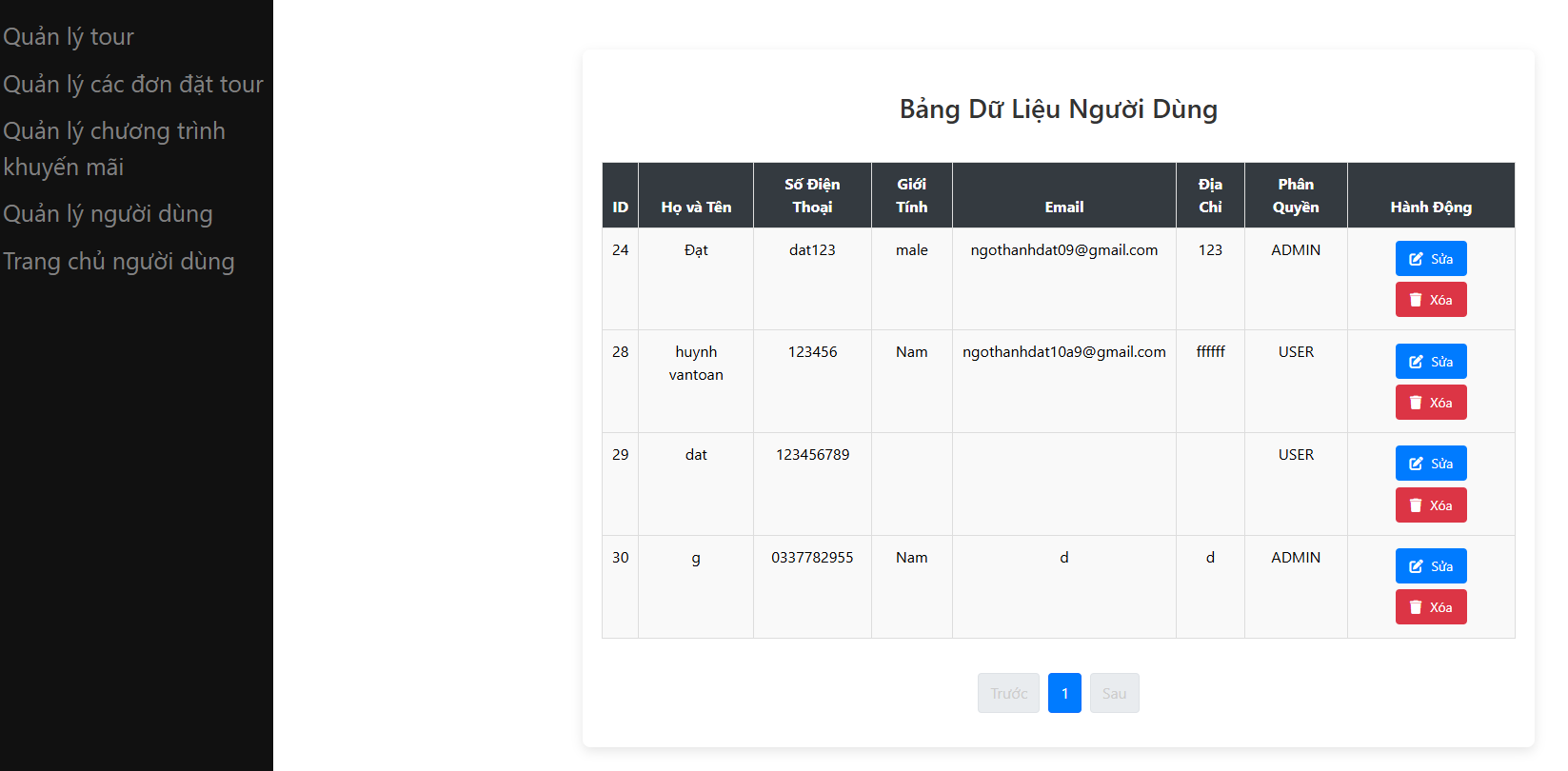
Hình 4. . Giao diện quản lý chương trình khuyến mãi



Hình 4. . Giao diện thêm chương trình khuyến mãi

## Giao diện quản lý người dùng.

Giao diện “Quản lý người dùng” là nơi quản trị viên có thể theo dõi, chỉnh sửa và xoá thông tin tài khoản của tất cả người dùng đã đăng ký hệ thống. Giao diện hiển thị dưới dạng bảng, bao gồm các cột: ID, Họ và Tên, Số Điện Thoại, Giới Tính, Email, Địa Chỉ, Phân Quyền và Hành Động. Trong đó, cột “Phân quyền” giúp phân biệt người dùng thường (USER) và quản trị viên (ADMIN). Mỗi dòng dữ liệu có hai nút chức năng: “Sửa” để chỉnh sửa thông tin người dùng và “Xoá” để xoá khỏi hệ thống. Giao diện còn hỗ trợ phân trang để quản lý hiệu quả số lượng lớn người dùng.



Hình 4. . Giao diện quản lý người dùng

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.**

* 1. **Kết luận**

Đề tài xây dựng website đặt tour du lịch đã đạt được mục tiêu đề ra, cung cấp một nền tảng trực tuyến tiện lợi giúp khách hàng dễ dàng tìm kiếm và đặt tour du lịch một cách nhanh chóng và hiệu quả. Hệ thống được phát triển dựa trên kiến trúc RESTful API, sử dụng Spring Boot cho backend và Angular cho frontend, đảm bảo tính linh hoạt, hiệu suất cao và khả năng mở rộng trong tương lai. Cơ sở dữ liệu MySQL được tích hợp chặt chẽ, hỗ trợ tốt cho việc quản lý thông tin tour, đơn đặt tour và dữ liệu người dùng.

Website không chỉ đáp ứng nhu cầu tìm kiếm và đặt tour mà còn hướng đến trải nghiệm người dùng tối ưu thông qua giao diện trực quan, thân thiện, hỗ trợ đa nền tảng trên cả máy tính và thiết bị di động. Các tính năng quan trọng như quản lý tour, theo dõi đơn đặt hàng và xác nhận thông tin khách hàng được xây dựng một cách hệ thống, giúp nâng cao hiệu quả hoạt động và sự thuận tiện cho người dùng.

* 1. **Hướng phát triển**

Trong tương lai, hệ thống có thể được phát triển thêm nhiều tính năng nâng cao nhằm nâng cao trải nghiệm khách hàng, bao gồm:

* Gợi ý tour thông minh dựa trên sở thích và hành vi của người dùng.
* Tích hợp đánh giá, phản hồi từ khách hàng để cải thiện chất lượng dịch vụ.
* Mở rộng dịch vụ liên quan: đặt vé xe, vé máy bay, khách sạn, homestay để cung cấp một hệ sinh thái du lịch toàn diện.

# TÀI LIỆU THAM KHẢO

[1] LCDUNG. (2017). Giới thiệu về Spring Boot. Spring Boot là gì?. Từ:  [https://lcdung.top/](%20https://lcdung.top/) [Truy cập 4/4/2025].

[2] Nhựt Liên. (2024). Spring boot là gì? Cách sử dụng Spring boot làm dự án như thế nào nhanh nhất?. Từ:  [https://fptshop.com.vn/](%20https://lcdung.top/) [Truy cập 4/4/2025].

[3] Spring.io. (n.d.). Core Features. Từ:  [https://spring.io/](%20https://spring.io/) [Truy cập 4/4/2025].

[4] Lokesh Gupta (2025). What is REST?. Từ:  [https://restfulapi.net/](%20https://spring.io/) [Truy cập 5/4/2025].

[5] Ngọc Thuý. (2024). Rest API là gì? Mách bạn tất tần tật những điều cần biết để thiết kế Rest API hiệu quả. Từ:  [https://fptshop.com.vn/](%20https://lcdung.top/) [Truy cập 4/4/2025].