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, 2025*

|  |  |
| --- | --- |
| **TRƯỜNG ĐẠI HỌC PHÚ YÊN**  **KHOA KỸ THUẬT – CÔNG NGHỆ** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc** |

KẾ HOẠCH QUÁ TRÌNH THỰC TẬP CỦA SINH VIÊN

Họ và tên: Ngô Thành Đạt ; Mã sinh viên: 211CTT025

Lớp: DC21CTT01; Khóa: 2021-2025 ; Ngành: Công nghệ thông tin

Địa điểm thực tập: Công ty IMT Miền Trung

Giảng viên hướng dẫn: Trần Xuân Hiệp

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Ngày tháng năm** | **Nội dung công việc** | **Xác nhận của GVHD** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |

Đánh giá chung của giảng viên hướng dẫn:

*………………,ngày ... tháng ... năm .......*

**Giảng viên hướng dẫn**

(Ký ghi rõ họ và tên)

**MỤC LỤC**

[**1.** **Lý do chọn đề tài** 5](#_Toc196572289)

[**2. Mục tiêu đề tài** 5](#_Toc196572290)

[**3. Đối tượng và phạm vi nghiên cứu** 6](#_Toc196572291)

[**3.1. Đối tượng nghiên cứu** 6](#_Toc196572292)

[**3.2. Phạm vi nghiên cứu** 6](#_Toc196572293)

[**4. Phương pháp nghiên cứu** 6](#_Toc196572294)

[**5. Nội dung nghiên cứu** 6](#_Toc196572295)

[**CHƯƠNG 1. GIỚI THIỆU VỀ CƠ SỞ THỰC TẬP** 8](#_Toc196572296)

[**1.1.** **Thông tin về công ty** 8](#_Toc196572297)

[**1.2.** **Tổng quan về công ty** 8](#_Toc196572298)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT** 10](#_Toc196572299)

[**2.1. Sring Boot Framework** 10](#_Toc196572300)

[***2.1.1.*** ***Spring Boot là gì ?.*** 10](#_Toc196572301)

[***2.1.2.*** ***Đặc điểm nổi bật của Spring Boot.*** 10](#_Toc196572302)

[***2.1.3.*** ***Các tính năng chính .*** 10](#_Toc196572303)

[**2.2. Mô hình RESTful API trong hệ thống web** 12](#_Toc196572304)

[***2.2.1.*** ***RESTful API là gì?*** 12](#_Toc196572305)

[***2.2.2.*** ***Các thành phần chính của RESTful API*** 13](#_Toc196572306)

[***2.2.3.*** ***Các bước thiết kế RESTful API*** 14](#_Toc196572307)

[***2.2.4.*** ***Ưu điểm và nhược điểm.*** 15](#_Toc196572308)

[**CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 17](#_Toc196572309)

[**3.1.** **Đặc tả yêu cầu** 17](#_Toc196572310)

[**3.1.1.** **Yêu cầu chức năng** 17](#_Toc196572311)

[**3.1.2.** **Yêu cầu phi chức năng** 18](#_Toc196572312)

[**3.2.** **Sơ đồ Use Case** 19](#_Toc196572313)

[**3.2.1.** **Danh sách các Use Case** 19](#_Toc196572314)

[**3.2.2.** **Sơ đồ Use Case tổng quát** 19](#_Toc196572315)

[**3.2.3.** **Sơ đồ Use Case chi tiết và đặc tả.** 20](#_Toc196572316)

[**3.3.** **Sơ đồ hoạt động** 30](#_Toc196572317)

[**3.3.1** **Hoạt động Quản lý tour.** 30](#_Toc196572318)

[**3.3.2** **Hoạt động Quản lý bookings.** 33](#_Toc196572319)

[**3.3.3** **Hoạt động Quản lý khuyến mãi** 34](#_Toc196572320)

[**3.3.4** **Hoạt động Quản lý lịch trình tour** 36](#_Toc196572321)

[**3.3.5** **Hoạt động Quản lý người dùng.** 37](#_Toc196572322)

[**3.3.6** **Hoạt động Quản lý thông tin cá nhân.** 39](#_Toc196572323)

[**3.3.6** **Hoạt động Đặt tour** 42](#_Toc196572324)

[**3.4.** **Sơ đồ tuần tự** 43](#_Toc196572325)

[**3.4.1.** **Tuần tự quản lý tour** 43](#_Toc196572326)

[**3.4.2.** **Tuần tự Quản lý bookings** 46](#_Toc196572327)

[**3.4.3.** **Tuần tự quản lý người dùng** 47](#_Toc196572328)

[**3.4.5.** **Tuần tự Đặt tour** 49](#_Toc196572329)

[**3.5.** **Sơ đồ lớp** 50](#_Toc196572330)

[**3.5.1.** **Lược đồ cơ sở dữ liệu quan hệ** 50](#_Toc196572331)

[**3.5.2.** **Mô tả các bảng** 51](#_Toc196572332)

[**CHƯƠNG 4. WEBSITE ĐẶT TOUR DU LỊCH TỈNH PHÚ YÊN** 57](#_Toc196572333)

[**4.1.** **Giao diện xem danh sách tours.** 57](#_Toc196572334)

[**4.2.** **Giao diện đặt tour** 59](#_Toc196572335)

[**4.3.** **Giao diện quản lý thông tin cá nhân** 59](#_Toc196572336)

[**4.4.** **Giao diện quản lý các tour đã đặt** 59](#_Toc196572337)

[**4.5.** **Giao diện danh sách các tour** 59](#_Toc196572338)

[**4.6.** **Giao diện thêm mới tour.** 59](#_Toc196572339)

[**4.7.** **Giao diện sửa thông tin tour.** 61](#_Toc196572340)

[**4.8.** **Giao diện xem danh sách booking.** 62](#_Toc196572341)

[**4.9.** **Giao diện cập nhật trạng thái booking.** 62](#_Toc196572342)

[**4.10.** **Giao diện xem danh sách người dùng.** 62](#_Toc196572343)

[**4.11.** **Giao diện sửa thông tin người dùng.** 62](#_Toc196572344)

[**4.12.** **Giao diện xoá người dùng.** 63](#_Toc196572345)

[**4.13.** **Giao diện quản lý chương trình khuyến mãi** 65](#_Toc196572346)

[**4.14.** **Giao diện đăng ký tài khoản.** 65](#_Toc196572347)

[**4.15.** **Giao diện đăng nhập.** 66](#_Toc196572348)

[**4.16.** **Giao diện đặt tour.** 66](#_Toc196572349)

[**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.** 67](#_Toc196572350)

**LỜI 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.

Với đề tài này, tác giả mong muốn xây dựng một website hỗ trợ địa phương trong việc khai thác tiềm năng du lịch, đặc biệt là các giá trị văn hóa và làng nghề truyền thống. Đồng thời, việc áp dụng Spring Boot – công nghệ mà tác giả đang tiếp cận trong quá trình thực tập tại doanh nghiệp – cũng là cơ hội để vận dụng kiến thức vào thực tế, từ đó nâng cao kỹ năng lập trình và khả năng giải quyết bài toán hiệu quả hơn.

# **2. Mục tiêu đề tài**

* Xây dựng website đặt tour tỉnh Phú Yên chuyên nghiệp, dễ sử dụng và cung cấp chức năng đặt tour, tối ưu hóa quy trình từ việc chọn tour, đặt vé 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 đồ án.

# **3. Đối tượng và phạm vi nghiên cứu**

## **3.1. 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.

## **3.2. 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.

# **4. 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.

# **5. 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. Kiến trúc 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.**

# **CHƯƠNG 1. GIỚI THIỆU VỀ CƠ SỞ THỰC TẬP**

* 1. **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
  1. **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.

# **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1. Sring Boot Framework**

1. ***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.

1. ***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.

* ***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.
* ***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.
* ***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.

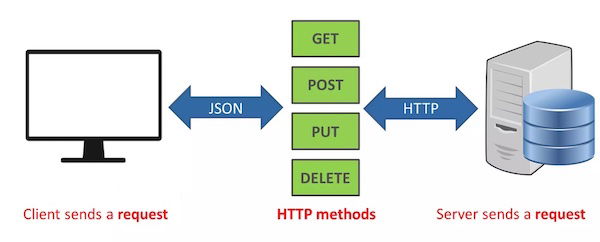
Bên cạnh đó, các framework như Quarkus hay Micronaut được tối ưu hóa cho môi trường cloud-native với thời gian khởi động nhanh, nhưng vẫn chưa thể sánh với hệ sinh thái phong phú và sự hỗ trợ toàn diện của Spring Boot. Trong khi đó, các framework non-Java như Django hay Laravel phù hợp để phát triển 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.

**2.2. Mô hình RESTful API trong hệ thống web**

1. ***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. 1.Minh hoạ 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.

1. ***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.

1. ***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.

1. ***Ưu điểm và nhược điểm.***

* ***Ưu điểm:***
* 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.

* ***Nhược điểm:***
* 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.

# **CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

1. **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 lựa chọn tour, khách hàng sẽ chọn một lịch trình cụ thể. Mỗi lịch trình có mức giá khác nhau tùy thuộc vào nội dung và thời gian tổ chức. Sau khi chọn lịch trình, hệ thống yêu cầu khách hàng nhập thông tin bao gồm: Họ tên, email, số điện thoại, ghi chú và thông tin chi tiết của tất cả những người tham gia tour.

Mỗi khách tham gia sẽ được phân loại vào một trong ba nhóm khách hàng, tương ứng với mức giá khác nhau tùy theo chính sách giá của từng tour:

* Trẻ em: dưới 12 tuổi
* Người lớn: 12-16 tuổi
* Người cao tuổi: trên 60 tuổi

Khách hàng xác nhận đơn đặt tour và thực hiện thanh toán.

***+ Quản lý các tour đã đặt***

Khách hàng có thể theo dõi, chỉnh sửa hoặc hủy các tour đã đặt, tùy theo trạng thái hiện tại của tour:

* Trạng thái “Đã thanh toán” – Khách hàng có thể chỉnh sửa/ huỷ trước ngày bắt đầu khởi hành 10 ngày (Khách hàng được hoàn trả tiền khi Huỷ).
* Trạng thái “Đã xác nhận” – Khách hàng không được phép chỉnh sửa/xoá.
* Trạng thái “Đã hoàn thành” – Cho biết khách hàng đã hoàn tất hành trình tour, chuyến đi đã diễn ra thành công và kết thúc đúng kế hoạch.
* Trạng thái “Đã huỷ” .

Trong trường hợp số lượng khách đã đặt cho một tour quá ít so với số lượng khách tối thiểu được yêu cầu cho lịch trình đó, admin sẽ liên hệ với khách hàng để thông báo tình hình và đưa ra một trong các phương án sau:

* Đề xuất huỷ tour và hoàn tiền (nếu đã thanh toán).
* Gợi ý chuyển sang một lịch trình tour khác có ngày khởi hành gần kề.

**+ Tìm kiếm thông tin tour**

* **Chức năng dành cho admin**

***+ Quản lý thông tin tour***

* Admin có thể thêm mới, chỉnh sửa hoặc xóa các tour du lịch trên hệ thống.
* Admin cập nhật trạng thái hoạt động của các tour (Đang hoạt động, Tạm dừng).
* Admin cập nhật giá cho các nhóm tuổi (Trẻ em, Người lớn, Người cao tuổi) và điều chỉnh mức giá tương ứng với từng nhóm tuổi.

***+ Quản lý lịch trình tour***

* Admin có thể thêm mới, chỉnh sửa hoặc xóa các lịch trình cho các tour du lịch, bao gồm ngày khởi hành và ngày kết thúc.
* Admin cập nhật các thông tin liên quan đến số ghế mỗi lịch tình, số ghế tối thiểu.

***+ Quản lý booking***

* Admin có thể xem danh sách các đơn đặt tour của người dùng.

***+ Quản lý thông tin người dùng***

* Admin được quyền xem, chỉnh sửa, hoặc xoá tài khoản người dùng trong trường hợp cần thiết.

***+ Quản lý chương trình khuyến mãi***

* Admin có quyền xem, thêm mới, chỉnh sửa hoặc xóa các chương trình khuyến mãi cho các lịch trình tour du lịch.
* Admin có thể cập nhật thông tin về giảm giá theo phần trăm hoặc giá trị cố định cho các 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

1. **Sơ đồ Use Case**
2. **Danh sách các Use Case**

* Đăng ký
* Đăng nhập
* Đặt tour
* Quản lý tour
* Quản lý booking
* Quản lý người dùng
* Quản lý các tour đã đặt
* Quản lý thông tin cá nhân
* Quản lý chương trình khuyến mãi

1. **Sơ đồ Use Case tổng quát**

**A group of white ovals with black text

AI-generated content may be incorrect.**

Hình 3. 1.Sơ đồ use case tổng quát

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

* **Use Case chi tiết quản lý tour**

**A group of white ovals with black text

AI-generated content may be incorrect.**

Hình 3. 2. Sơ đồ use case chi tiết quản lý tour

* **Đặc tả Use case quản lý tour**

|  |  |
| --- | --- |
| **Quản lý tour** | |
| Mục đích: | Quản lý thông tin các tour |
| Mô tả: | Thực hiện thêm, sửa, tạm dừng tour, quản lý ảnh và 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 có thể thực hiện các thao tác thêm, sửa, xoá, quản lý ảnh, quản lý lịch trình của tour.   4.1. Tour đã có khách hàng đặt (booking):   * Khi một tour đã có ít nhất một đơn đặt tour (booking), admin sẽ không được phép chỉnh sửa hoặc xóa tour đó.   →Trong giao diện quản trị, các nút "Chỉnh sửa" và "Xóa" sẽ bị ẩn đối với các tour đã có lượt đặt.   * Tour chưa có khách hàng đặt:   Nếu tour chưa có bất kỳ đơn đặt nào, admin được phép chỉnh sửa hoặc xóa thông tin của tour như bình thường.  4.2. Quản lý lịch trình tour hiển thị tất cả các lịch trình của tour đó |
| Luồng sự kiện phụ (Alternative Flows): | 5. Nếu thông tin không phù hợp hệ thống hiện đúng thông báo lỗi và yêu cầu quản trị viên nhập lại. |

* **Use Case chi tiết quản lý người dùng**

A black background with white ovals

Description automatically generated

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

* **Đặc tả Use Case quản lý người dùng**

|  |  |
| --- | --- |
| **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 sửa, xoá người dùng.   4.1. Thực hiện chỉnh sửa thông tin cá nhân người dùng hoặc chỉnh sửa phân quyền người dùng  4.2. Không được phép xoá người dùng đang có đơn đặt |
| Luồng sự kiện phụ (Alternative Flows): | 5. Nếu thông tin không phù hợp hệ thống hiện thông báo và cho quản trị viên nhập lại |

* **Use case chi tiết quản lý bookings**

A group of white ovals with black text

AI-generated content may be incorrect.

Hình 3. 4. Sơ đồ use case chi tiết quản lý bookings

* **Đặc tả Use case quản lý booking**

|  |  |
| --- | --- |
| **Quản lý booking** | |
| Mục đích: | Quản lý thông tin booking |
| Mô tả: | Thực hiện cập nhật trạng thái, xoá thông tin booking |
| 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 booking. 3. Hệ thống hiển thị danh sách booking. 4. Admin có thể thực hiện các thao tác điền vào ghi chú bookings, cập nhật trạng thái bookings, xem thông tin chi tiết.   4.1. Việc cập nhật trạng thái sẽ là:Đã xác nhận, Đã hoàn thành, Đã huỷ  4.2. Khi xem thông tin chi tiết của một đơn đặt tour, hệ thống sẽ hiển thị danh sách các thành viên tham gia kèm theo các thông tin tương ứng như họ tên, giới tính, ngày sinh, nhóm tuổi và giá tiền của từng người. |
| Luồng sự kiện phụ (Alternative Flows): | 5. Nếu thông tin không phù hợp hệ thống hiện thông báo và cho quản trị viên nhập lại |

* **Use Case chi tiết Quản lý thông tin cá nhân**

***A black and white background with white ovals

AI-generated content may be incorrect.***

Hình 3. 5. Sơ đồ use case chi tiết quản lý thông tin cá nhân

* **Đặc tả Use Case quản lý cá nhân**

|  |  |
| --- | --- |
| **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): | 5. Nếu thông tin không phù hợp hệ thống hiện thông báo nhập lại. |

* **Use Case chi tiết Quản lý chương trình khuyến mãi**

****

Hình 3. 6. Sơ đồ use case chi tiết quản lý chương trình khuyến mãi.

* **Đặc tả Use case Quản lý chương trình khuyến mãi**

|  |  |
| --- | --- |
| **Quản lý chương trình khuyến mãi** | |
| Mục đích: | Quản lý chương trình khuyến mãi các tour |
| Mô tả: | Xem, thêm mới, 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: Id tour, 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. |

* **Quản lý các tour đã đặt**

**A group of white ovals with black text

AI-generated content may be incorrect.**

Hình 3. 7. Sơ đồ use case chi tiết quản lý các tour đã đặt

* **Đặc tả Use Case quản lý các tour đã đặt**

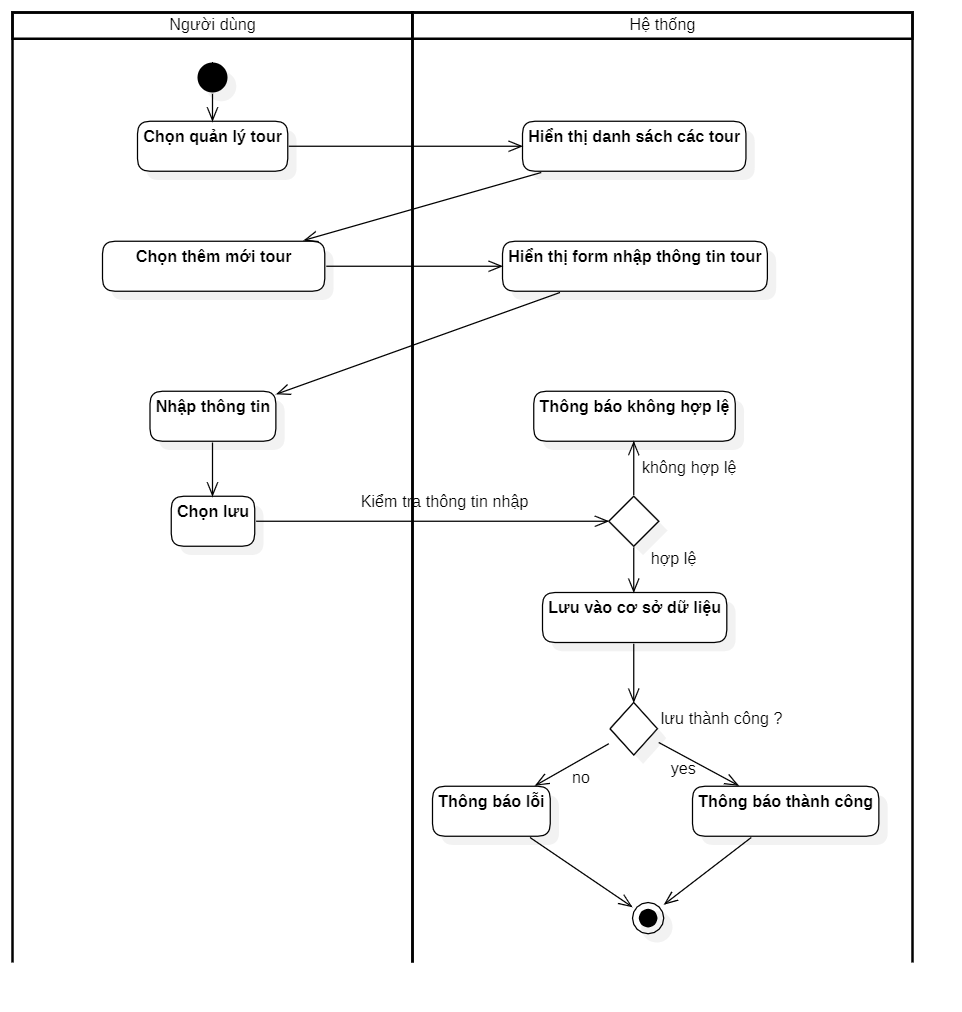
|  |  |
| --- | --- |
| **Quản lý các tour đã đặt** | |
| Mục đích: | Quản lý các tour mà khách hàng đã đặt |
| Mô tả: | Xem, sửa , huỷ các tour đã đặt |
| Tác nhân: | Khách hàng |
| Điều kiện trước: | Đăng nhập thành công, chọn chức năng quản lý các tour đã đặt |
| Điều kiện sau: | Đăng nhập thành 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”   4.1. Khách hàng chỉ có thể chỉnh sửa thông tin hoặc hủy tour khi đơn đặt ở trạng thái "Đã thanh toán".  Trạng thái này sẽ tự động chuyển thành "Đã xác nhận" sau 10 ngày trước ngày khởi hành của tour.  Khi đơn đặt chuyển sang trạng thái "Đã xác nhận", các nút "Sửa" và "Hủy" sẽ bị vô hiệu hóa, đồng nghĩa với việc khách hàng không thể thay đổi thông tin hoặc hủy tour đã đặt.. |
| Luồng sự kiện phụ (Alternative Flows): | 5. Nếu thông tin không phù hợp hệ thống hiện thông báo. |

* + **Đặc tả Use Case Đặt tour**

|  |  |
| --- | --- |
| **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. 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. |

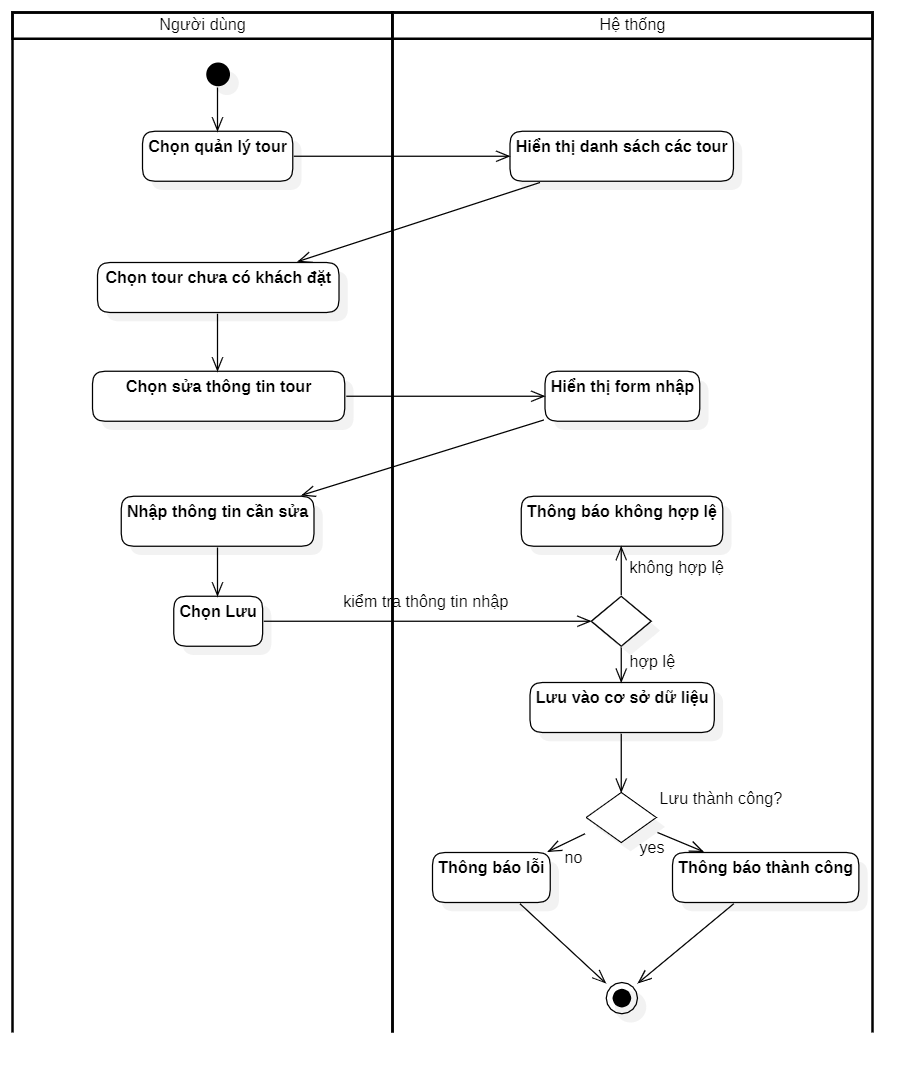
1. **Sơ đồ hoạt động**
2. **Hoạt động Quản lý tour.**

* **Thêm mới tour**

****

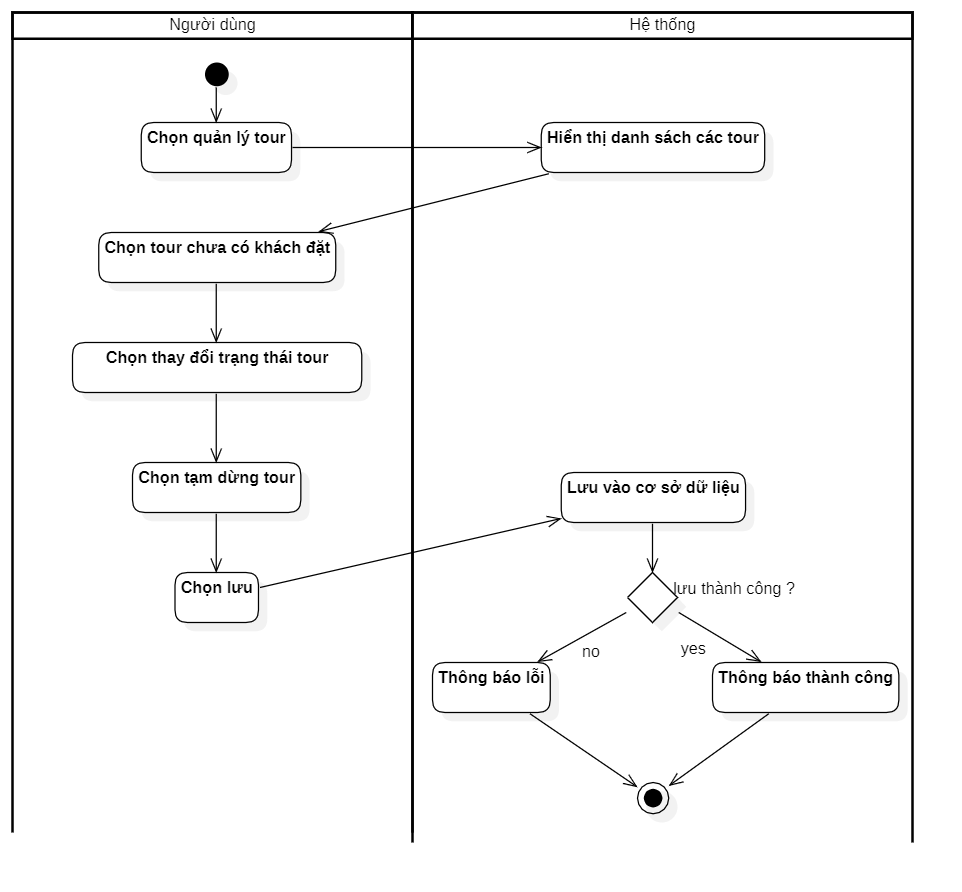
Hình 3. 8. Sơ đồ hoạt động thêm mới tour

* **Sửa thông tin tour**

****

Hình 3. 9. Sơ đồ hoạt động sửa thông tin tour.

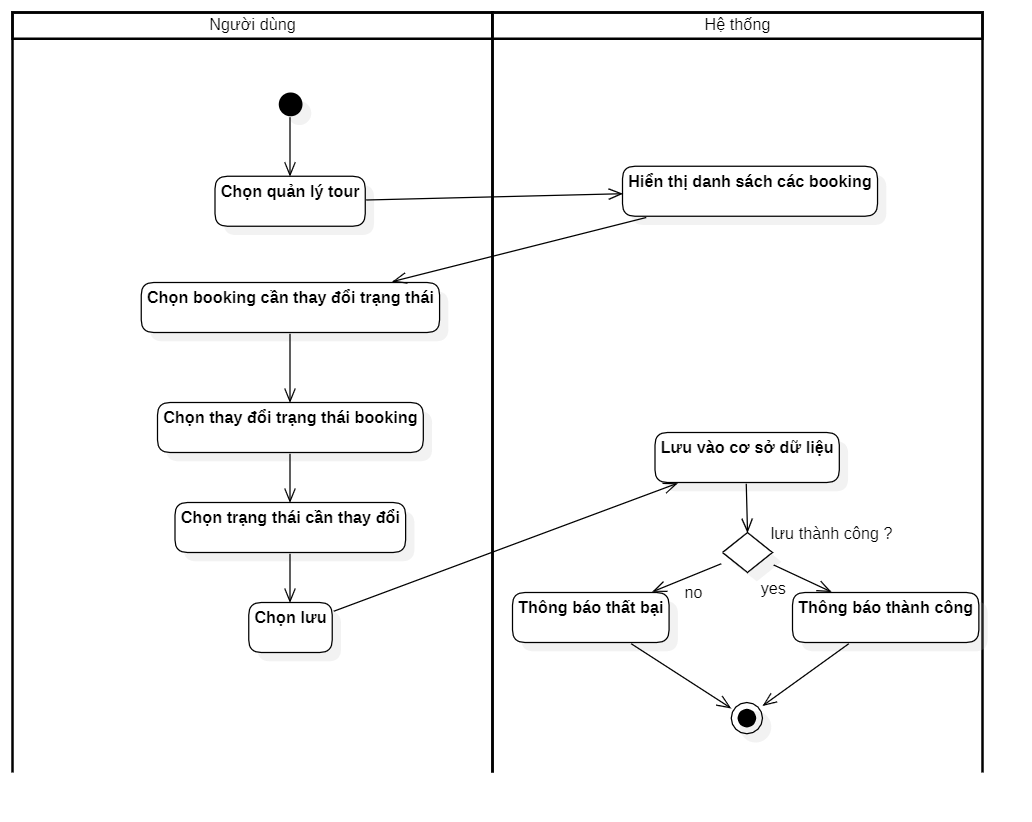
* **Tạm dừng tour**

****

Hình 3. 10. Sơ đồ hoạt động tạm dừng tour.

1. **Hoạt động Quản lý bookings.**

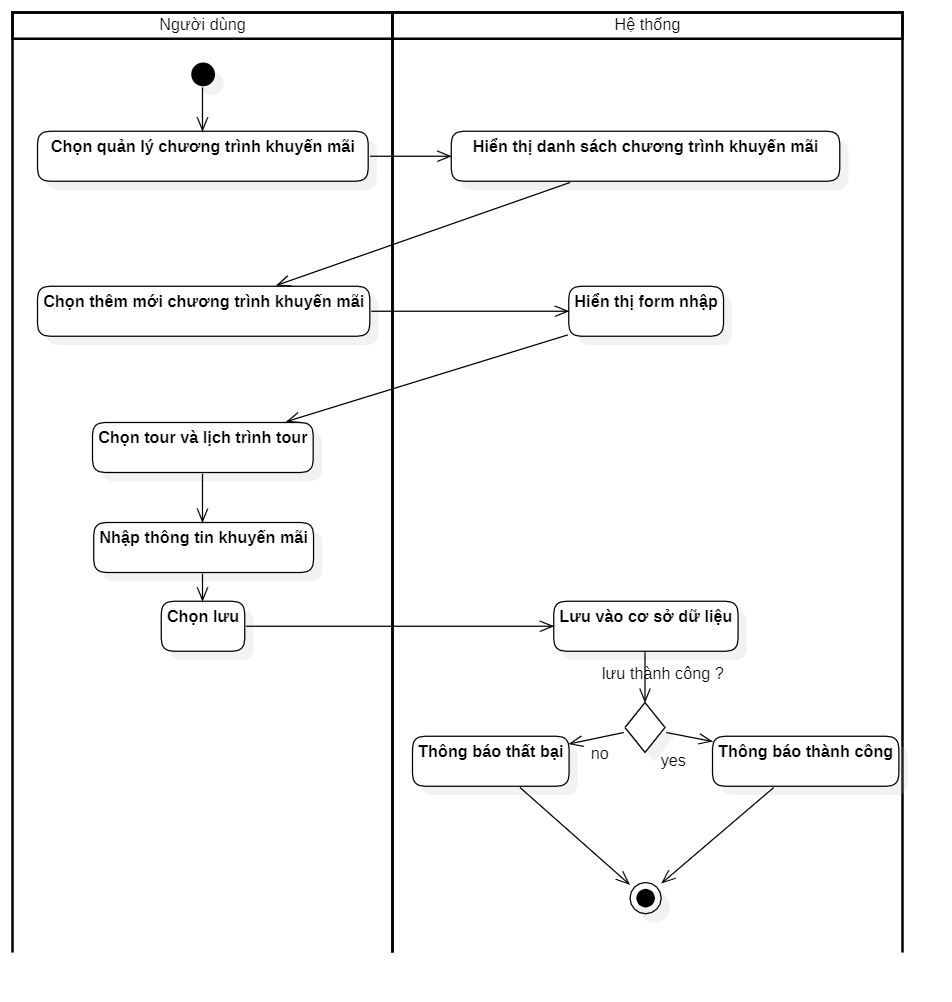
* **Cập nhật trạng thái bookings**

****

Hình 3. 11. Sơ đồ hoạt động cập nhật trạng thái booking.

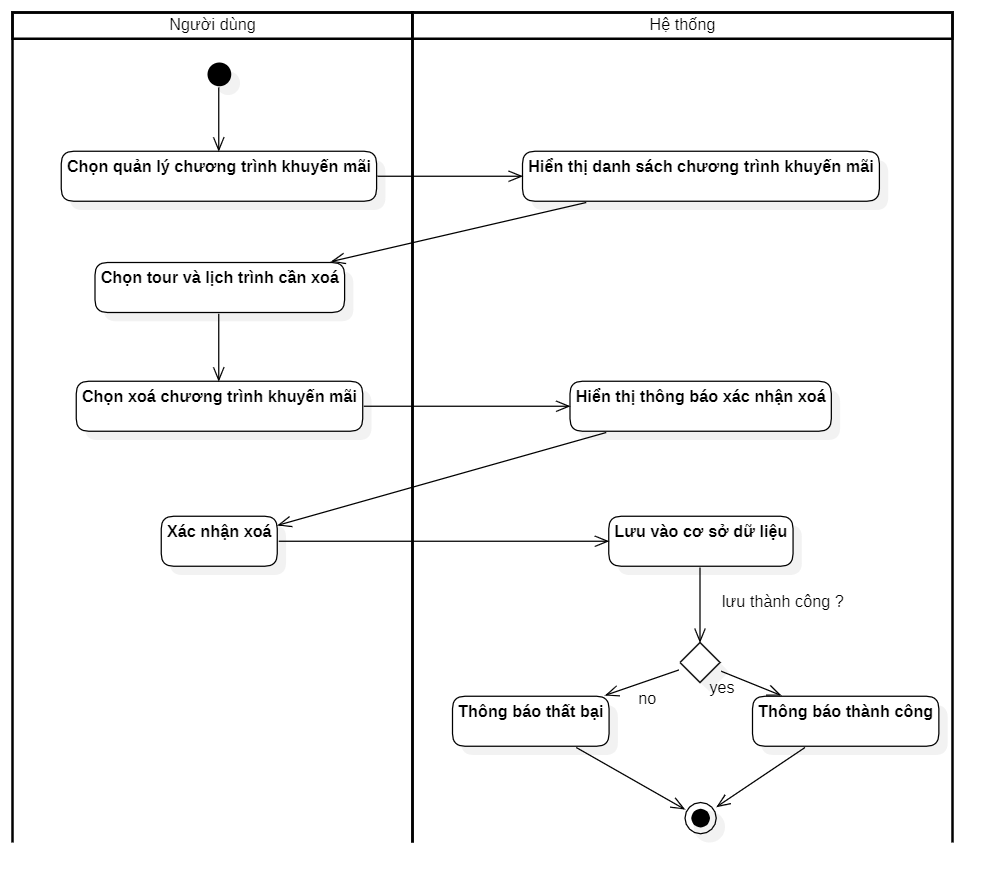
1. **Hoạt động Quản lý khuyến mãi**

* **Thêm mới chương trình khuyến mãi cho tour**

****

Hình 3. 12. Sơ đồ hạot động thêm mới chương trình khuyến mãi.

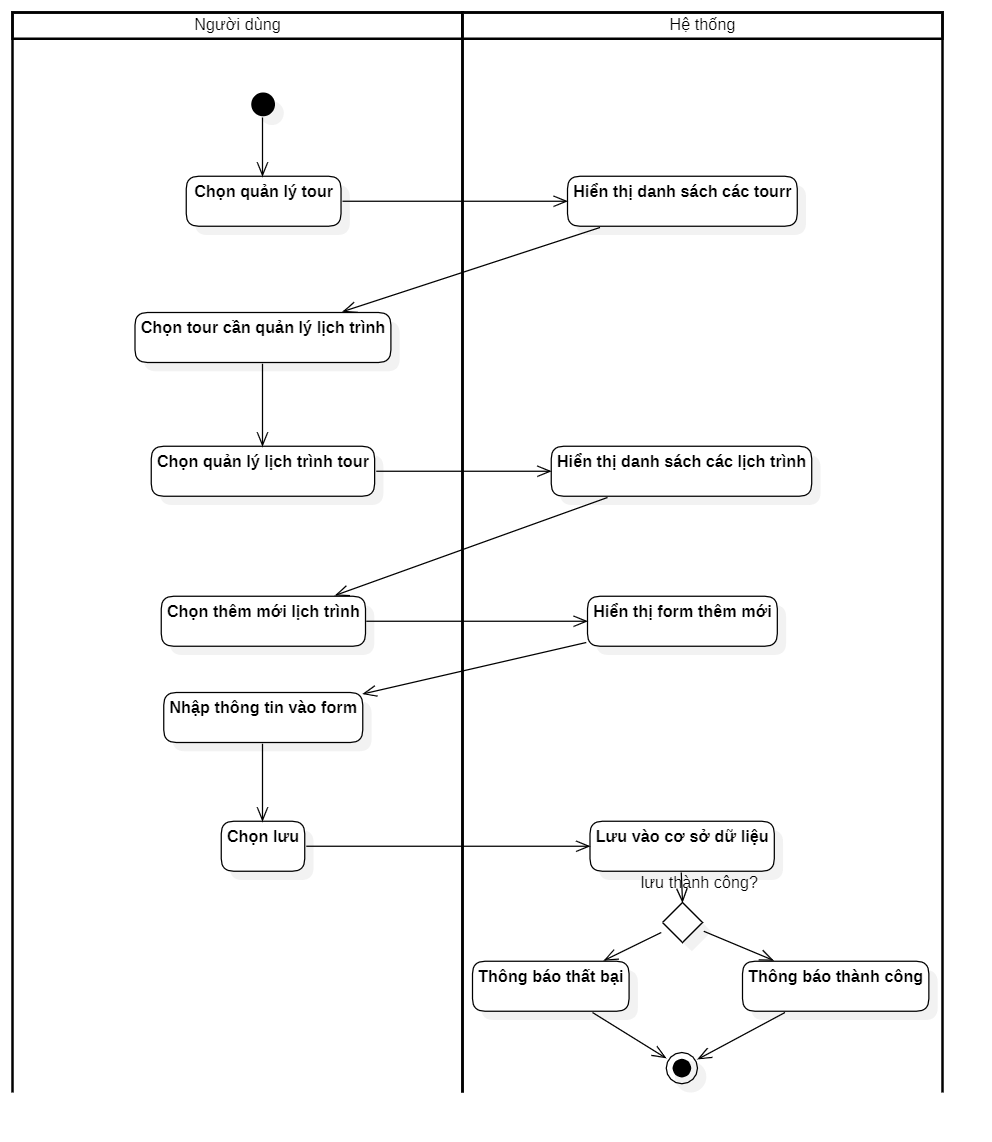
* **Xoá chương trình khuyến mãi**

****

Hình 3. 13. Sơ đồ hoạt động xoá chương trình khuyến mãi.

1. **Hoạt động Quản lý lịch trình tour**

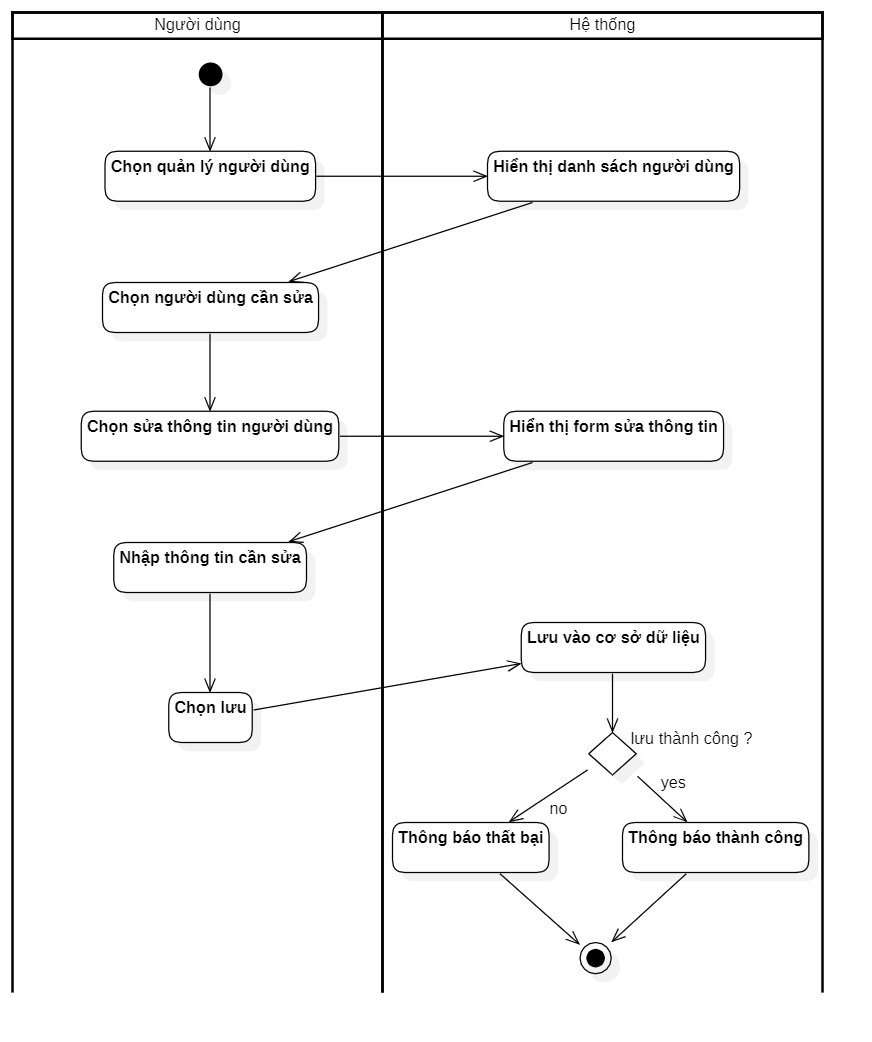
* **Thêm mới lịch trình tour**

****

Hình 3. 14. Sơ đồ hoạt động thêm mới lịch trình tour.

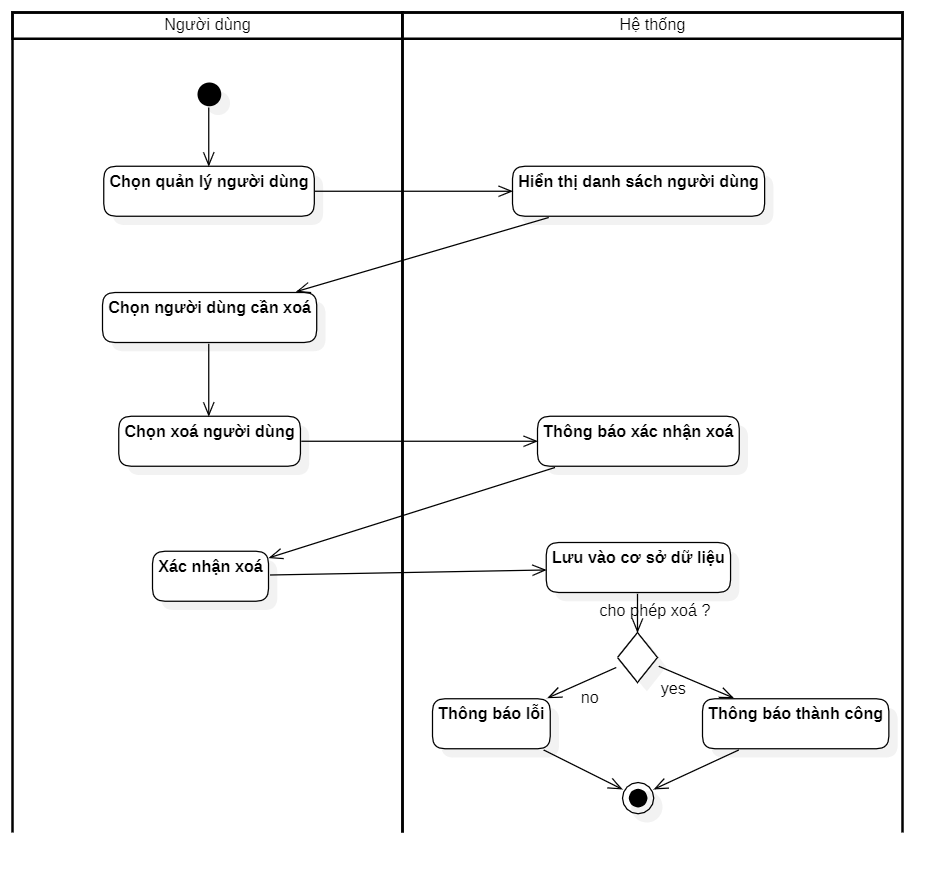
1. **Hoạt động Quản lý người dùng.**

* **Sửa thông tin người dùng**

****

Hình 3. 15. Sơ đồ hoạt động sửa thông tin người dùng.

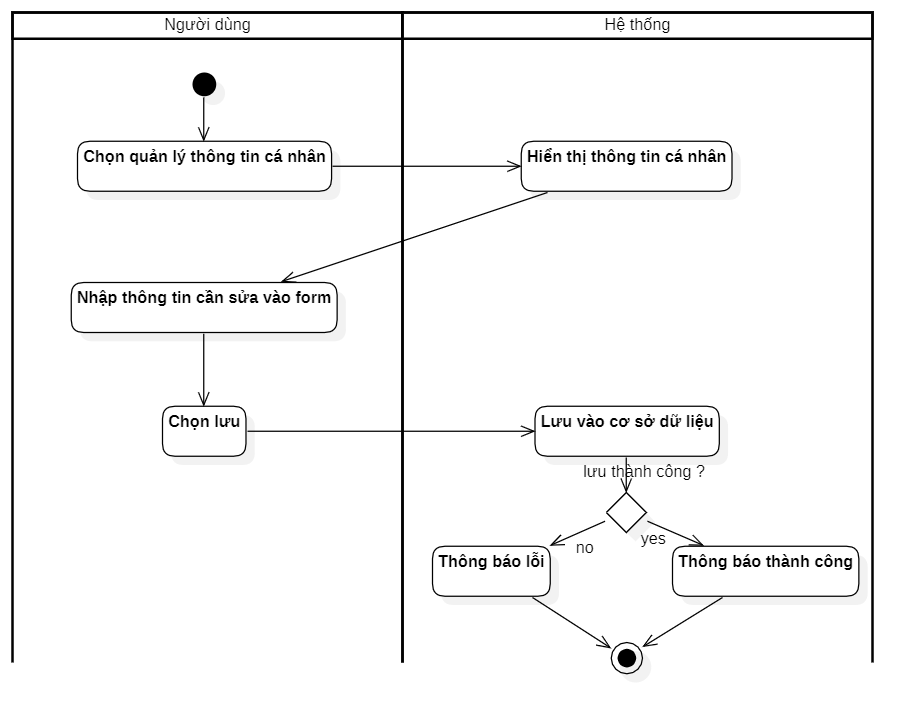
* **Xoá người dùng**

****

Hình 3. 16. Sơ đồ hoạt động xoá thông tin người dùng.

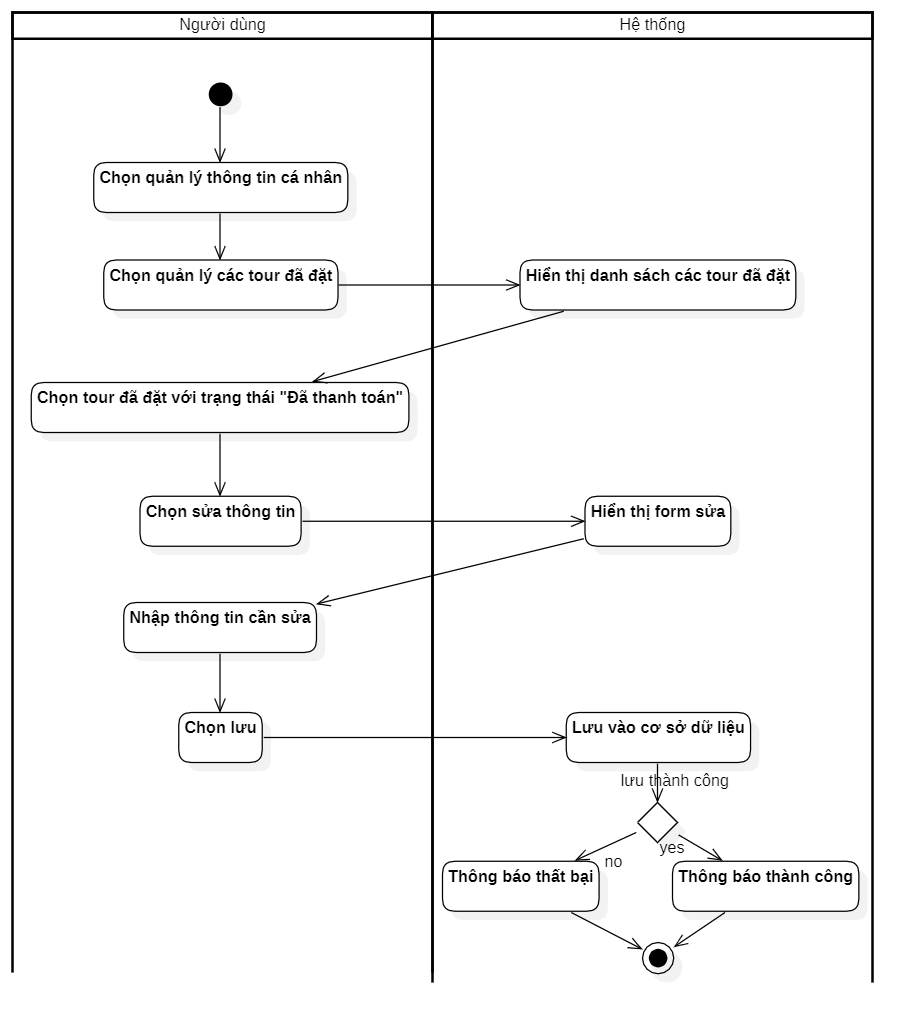
1. **Hoạt động Quản lý thông tin cá nhân.**

* **Sửa thông tin cá nhân**

****

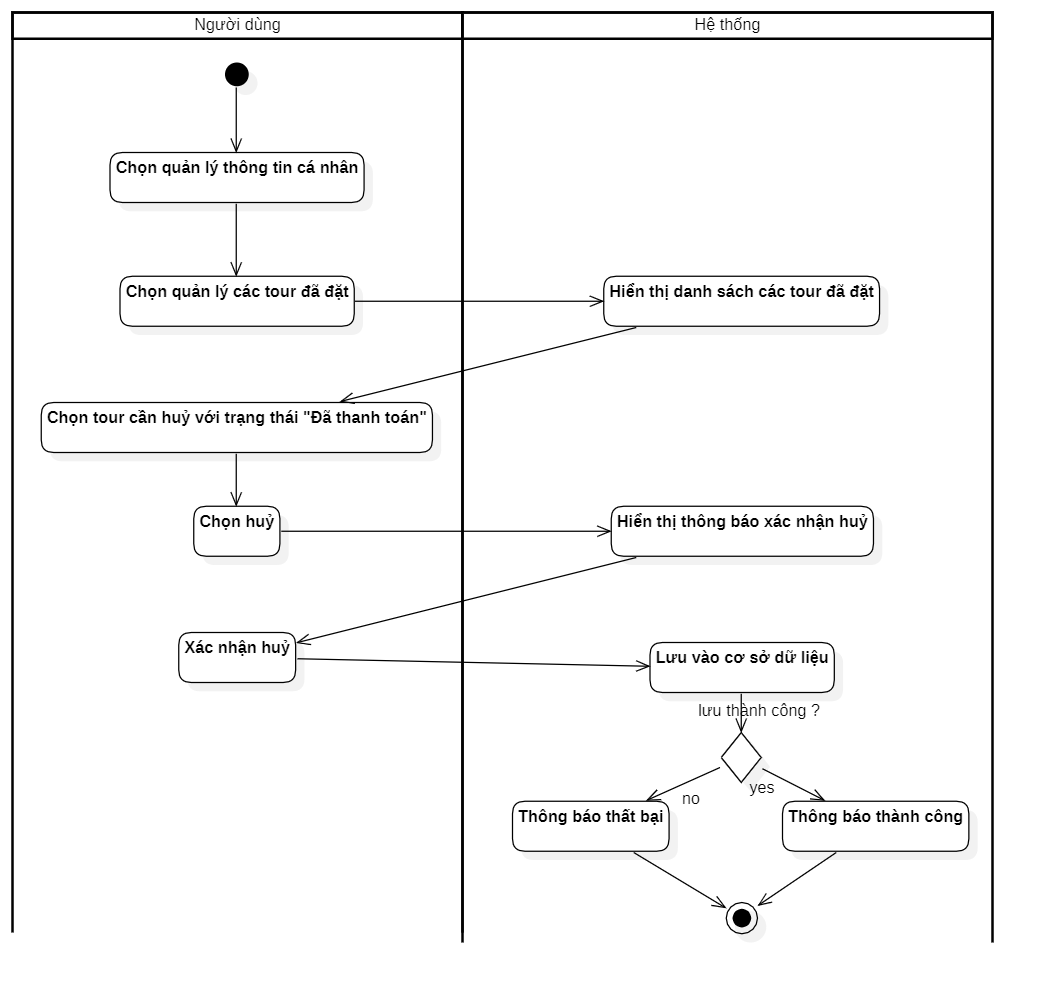
Hình 3. 17. Sơ đồ hoạt động sửa thông tin cá nhân.

* + 1. **Hoạt động quản lý các tour đã đặt**
* **Chỉnh sửa thông tin tour đã đặt**

****

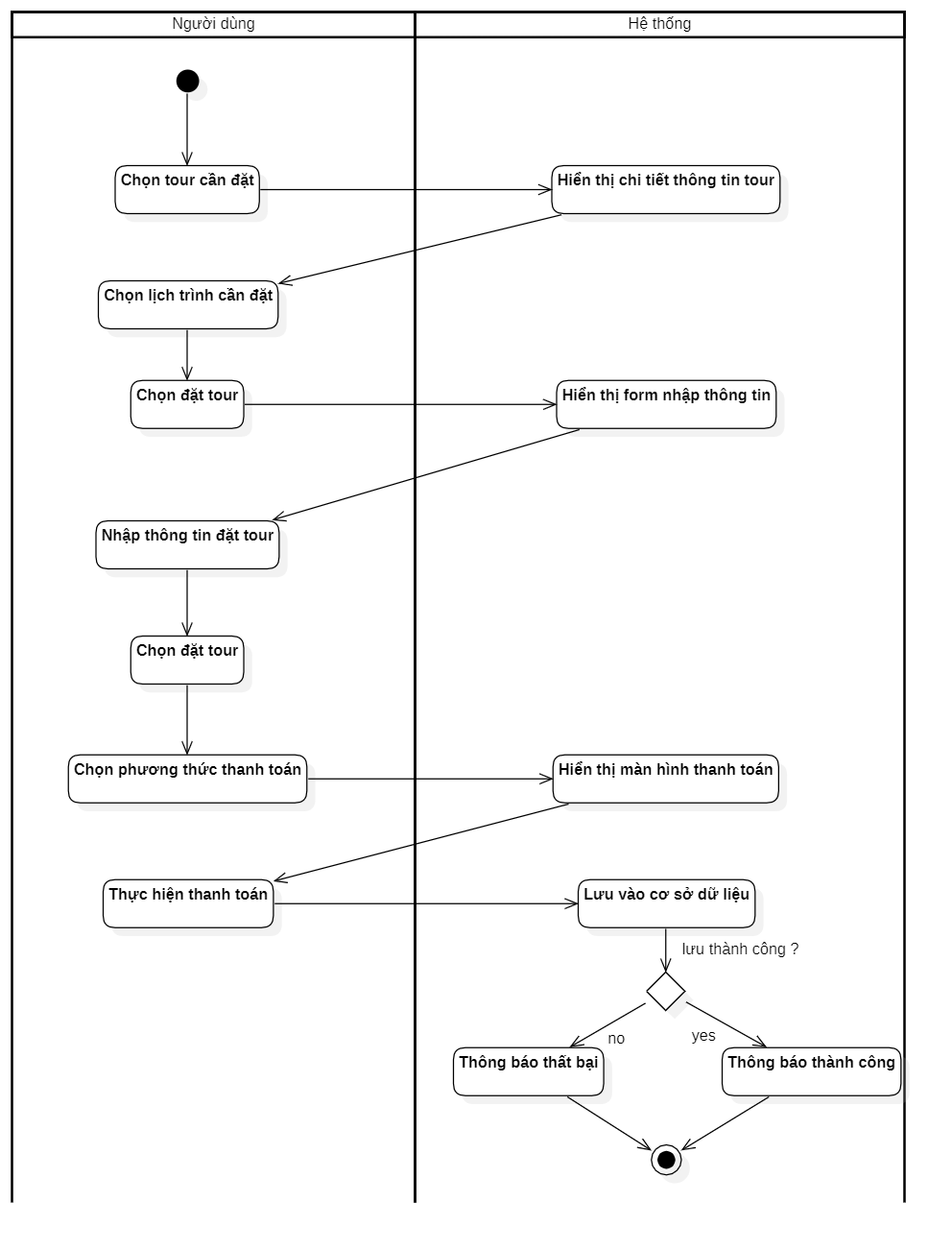
Hình 3. 18. Sơ đồ hoạt động chỉnh sửa thông tin các tour đã đặt.

* **Huỷ tour đã đặt**

****

Hình 3. 19. Sơ đồ hoạt động huỷ tour đã đặt.

* + 1. **Hoạt động Đặt tour**

****

Hình 3. 20. Sơ đồ hoạt động đặt tour.

* 1. **Sơ đồ tuần tự**

1. **Tuần tự quản lý tour**

* **Thêm mới tour**

A screenshot of a computer

Description automatically generated

Hình 3. 21. Sơ đồ tuần tự thêm mới tour.

* **Sửa thông tin tour**

A screenshot of a computer screen

Description automatically generated

Hình 3. 22. Sơ đồ tuần tự sửa thông tin tour.

* **Tạm dừng tour**

A screenshot of a computer screen

Description automatically generated

Hình 3. 23. Sơ đồ tuần tự tạm dừng tour.

1. **Tuần tự Quản lý bookings**

* **Cập nhật trạng thái bookings**

A screenshot of a computer screen

Description automatically generated

Hình 3. 24. Sơ đồ tuần tự cập nhật trạng thái booking

1. **Tuần tự quản lý người dùng**

* **Sửa thông tin người dùng**

A screenshot of a computer screen

Description automatically generated

Hình 3. 25. Sơ đồ tuần tự sửa thông tin người dùng

1. **Tuần tự quản lý thông tin cá nhân**

* **Sửa thông tin cá nhân**

A screenshot of a computer screen

Description automatically generated

Hình 3. 26. Sơ đồ tuần tự sửa thông tin cá nhân.

1. **Tuần tự Đặt tour**

A screenshot of a computer

Description automatically generated

Hình 3. 27. Sơ đồ tuần tự đặt tour.

* 1. **Sơ đồ lớp**
     1. **Lược đồ cơ sở dữ liệu quan hệ**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 3. 28. Lược đồ cơ sở dữ liệu quan hệ.

* + 1. **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**

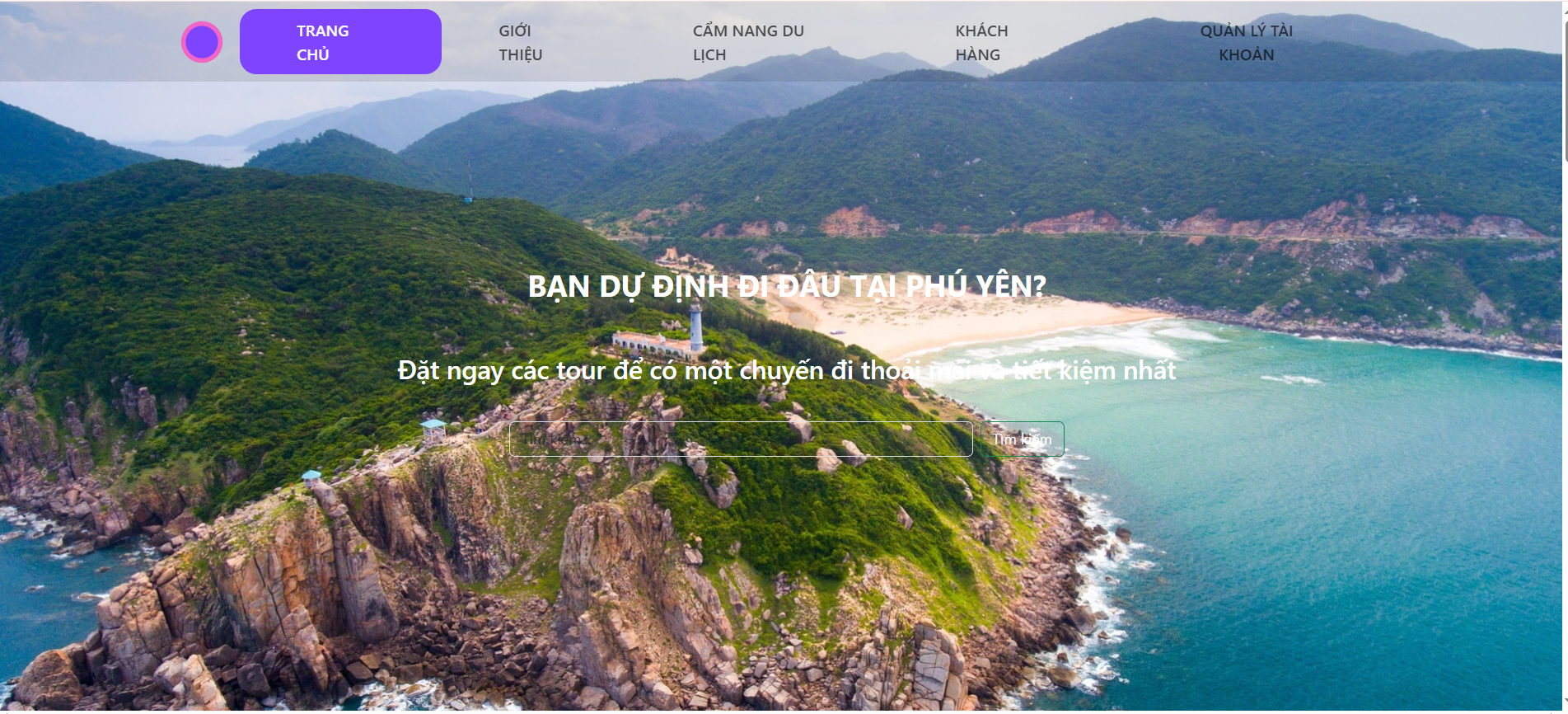
|  |  |
| --- | --- |
| **Định nghĩa** | **Thuộc tính** |
| Lưu trữ thông tin chi tiết thanh toán | * **Id:** Mã thanh toán * **Booking\_id:** Liên kết đơn đặt * **Amount:** Tổng số tiền đã thanh toán * **Payment\_date:** Ngày thanh toán * **Payment\_method:** Phương thức thanh toán * **Payment\_status:** 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 |

**CHƯƠNG 4. WEBSITE ĐẶT TOUR DU LỊCH TỈNH PHÚ YÊN**

1. **Giao diện xem danh sách tours.**

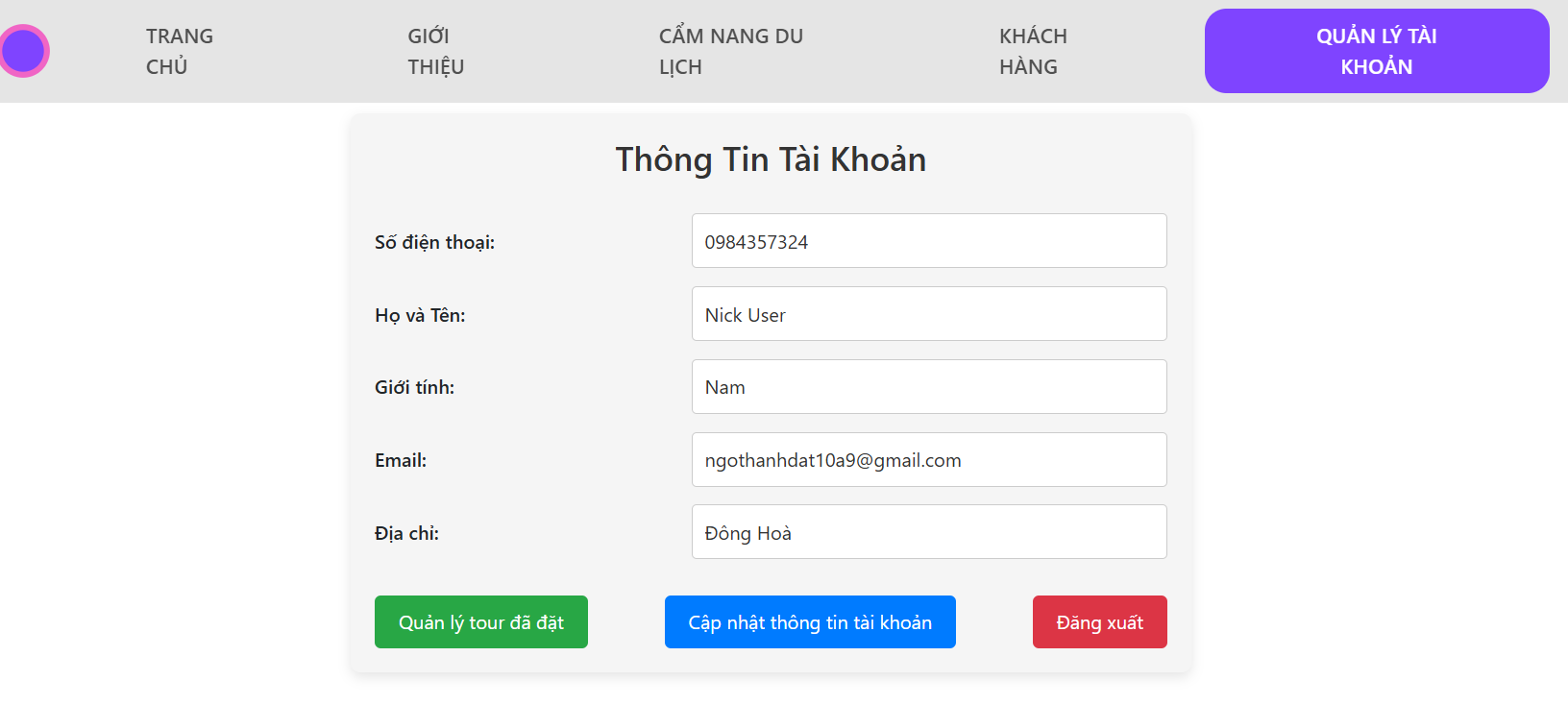


Hình 4. 1.Giao diện trang chủ



Hình 4. 2. Danh sách các tour.

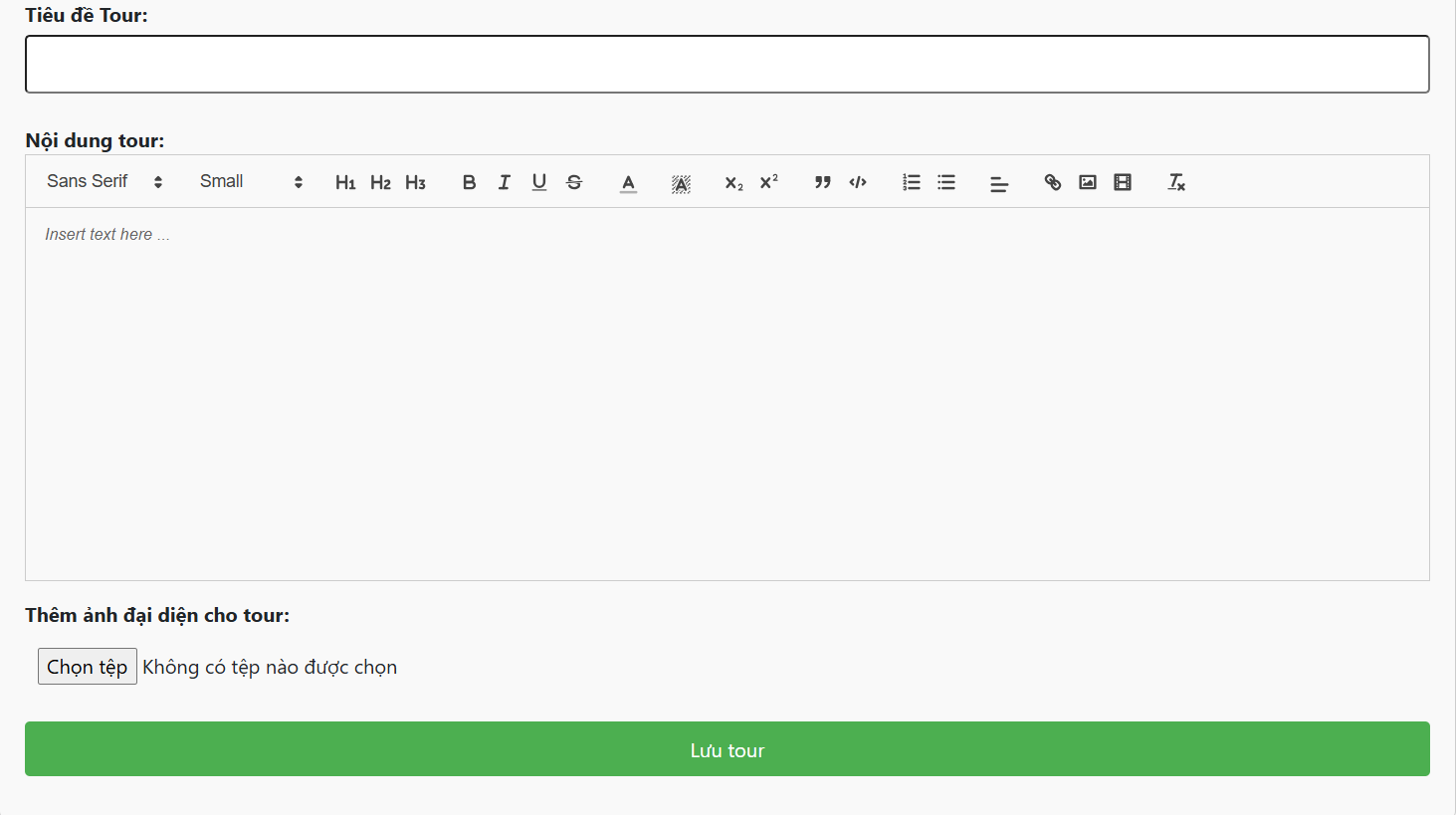
1. **Giao diện đặt tour**
2. **Giao diện quản lý thông tin cá nhân**



Hình 4. 3. Quản lý thông tin cá nhân.

1. **Giao diện quản lý các tour đã đặt**
2. **Giao diện danh sách các tour**
3. **Giao diện thêm mới tour.**

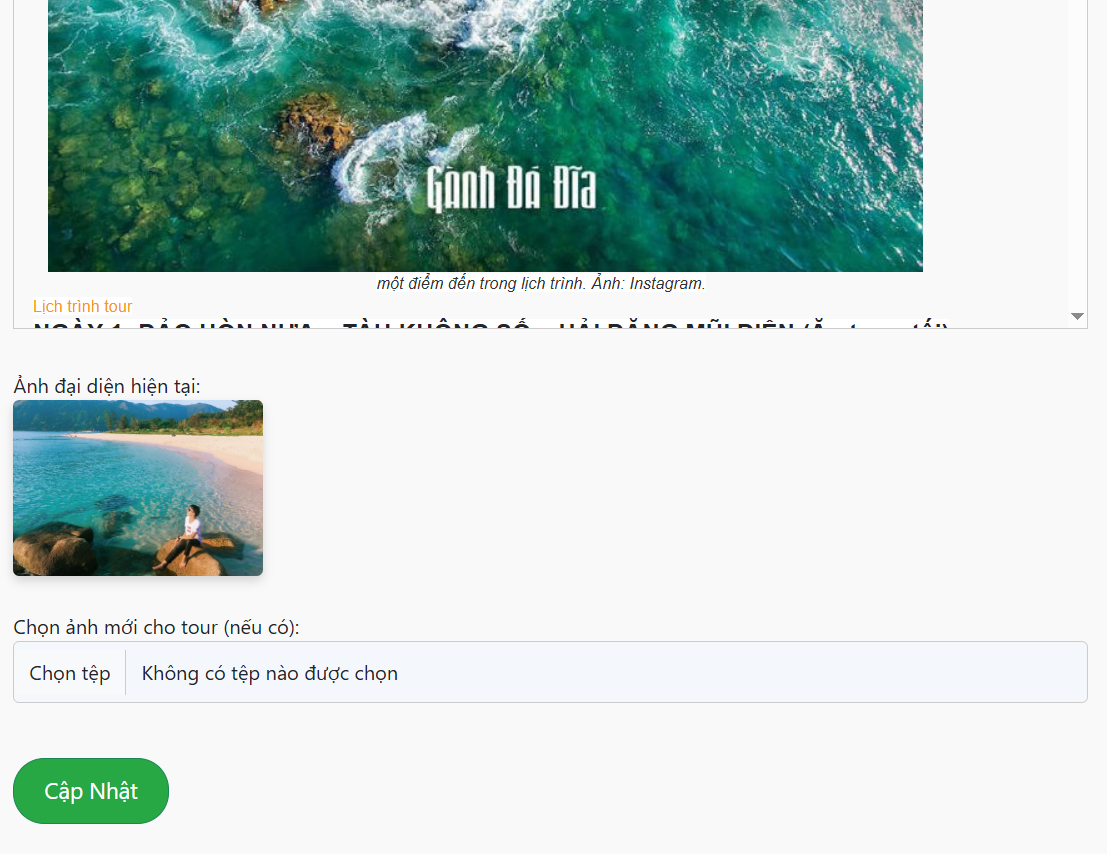




Hình 4. 5. Thêm mới tour.

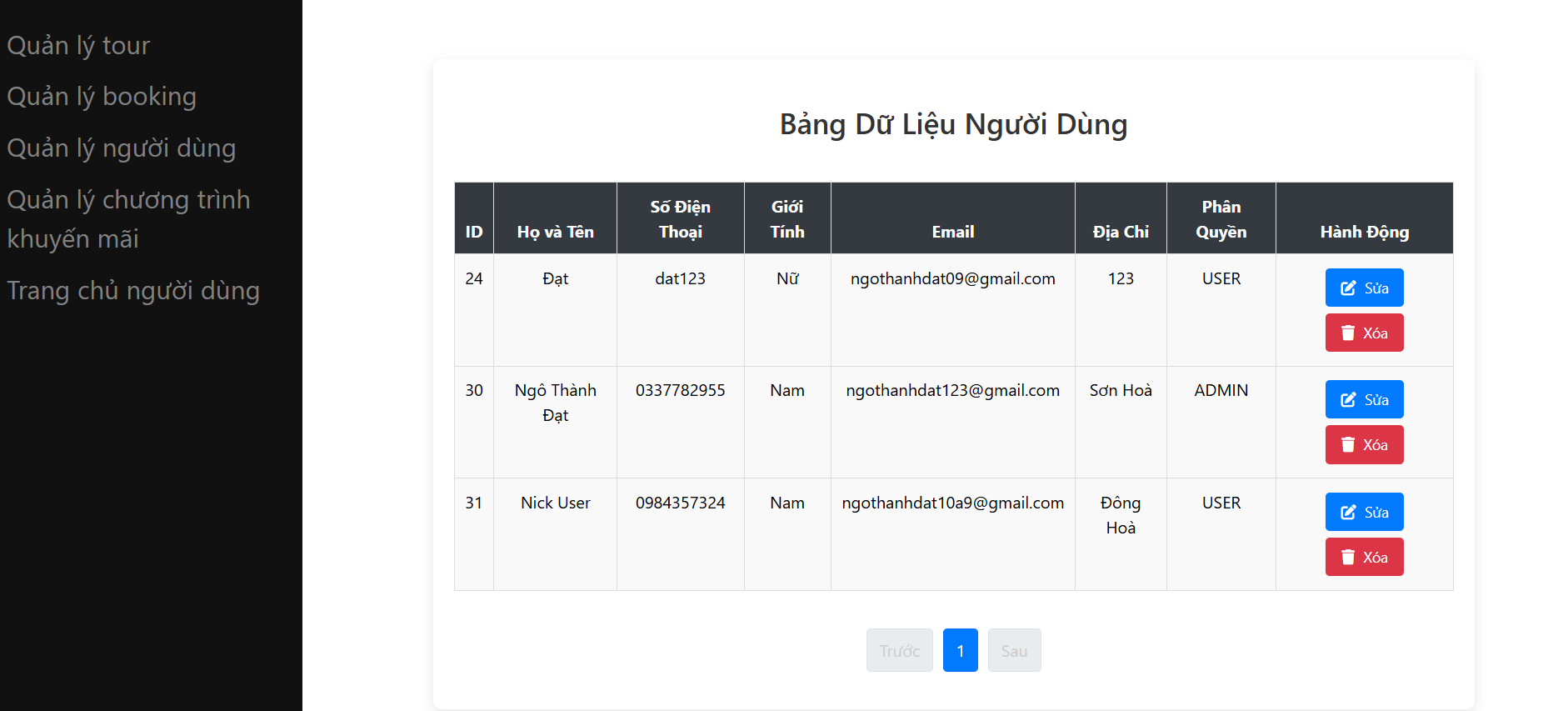
1. **Giao diện sửa thông tin tour.**



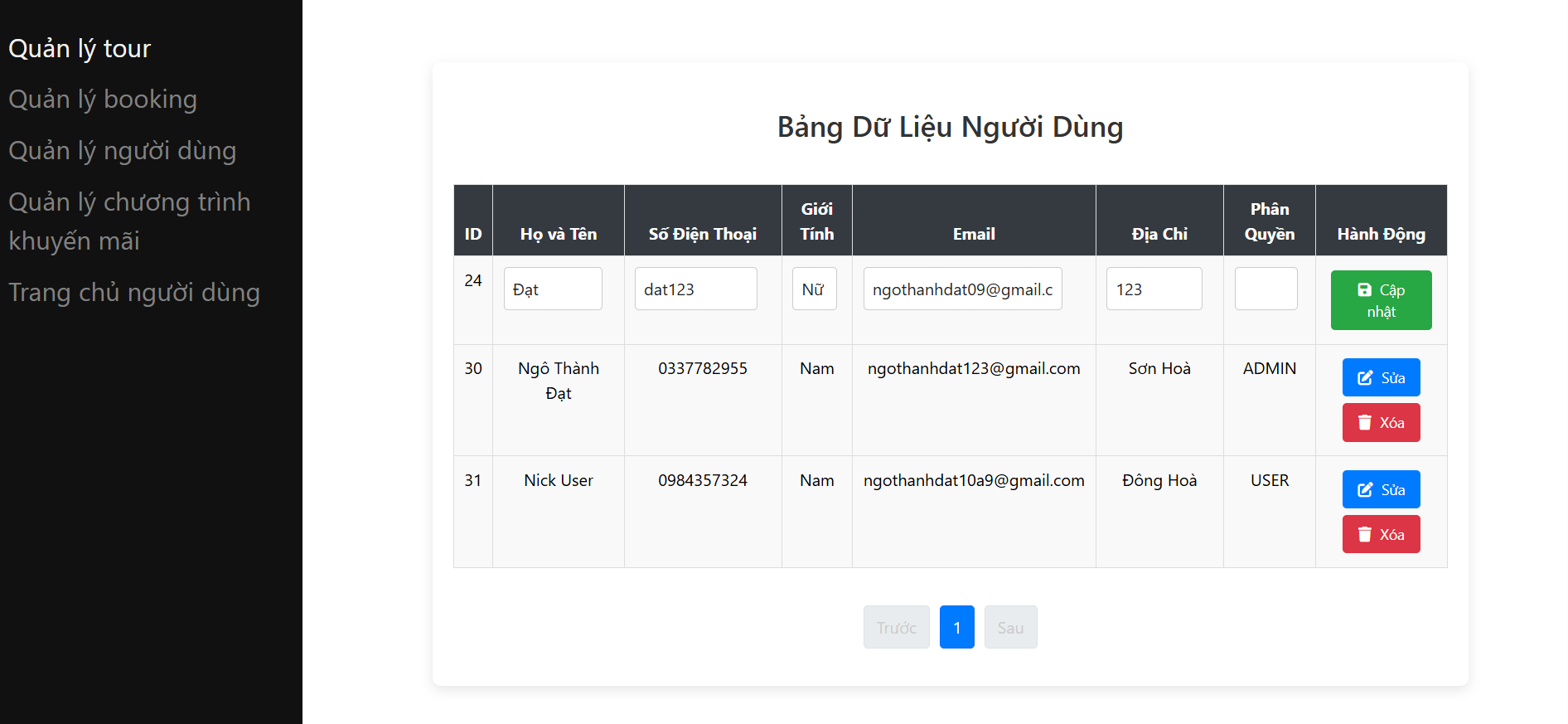


Hình 4. 6. Sửa thông tin tour.

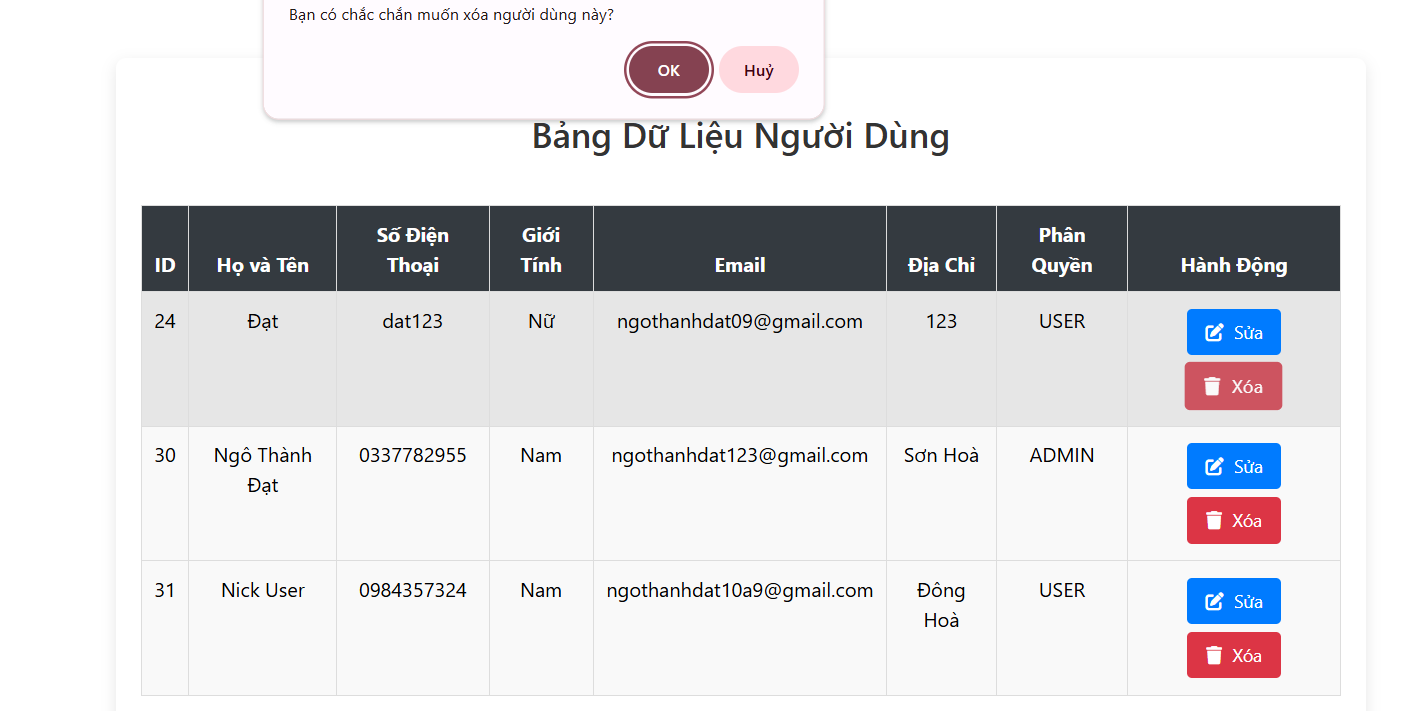
1. **Giao diện xem danh sách booking.**
2. **Giao diện cập nhật trạng thái booking.**
3. **Giao diện xem danh sách người dùng.**

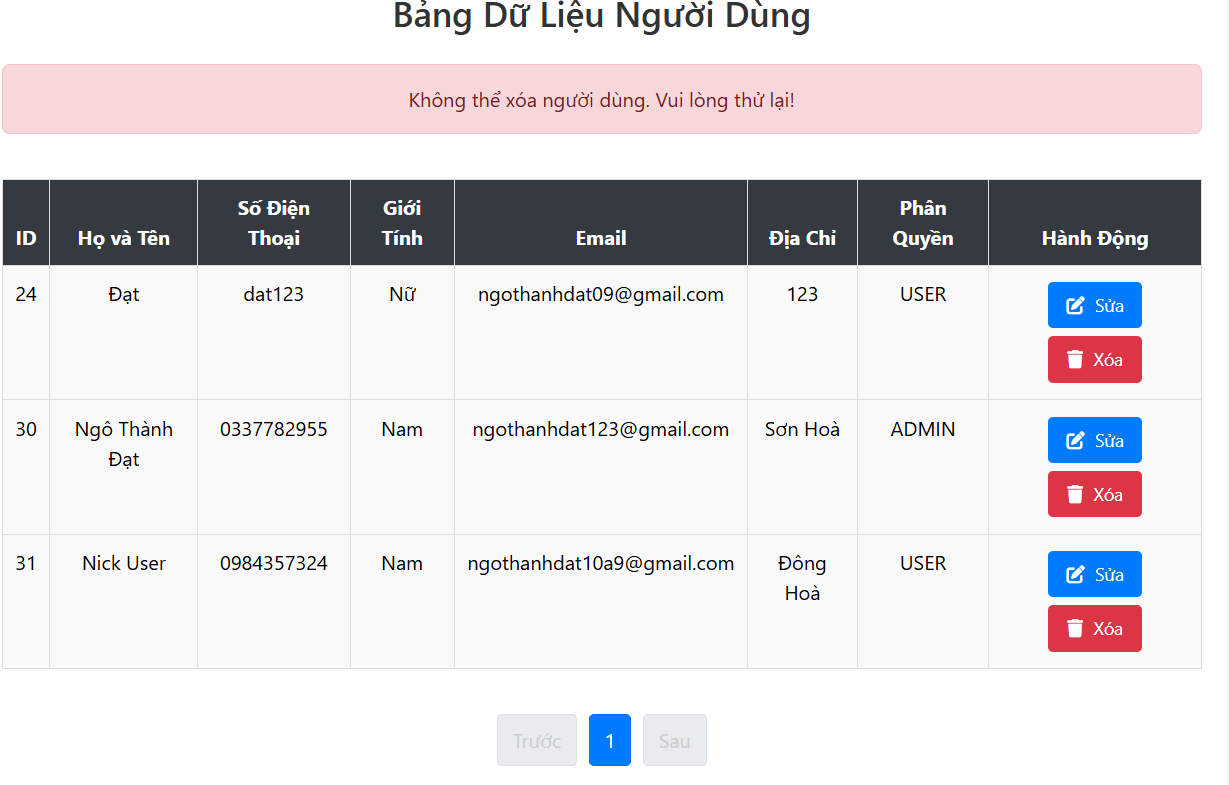


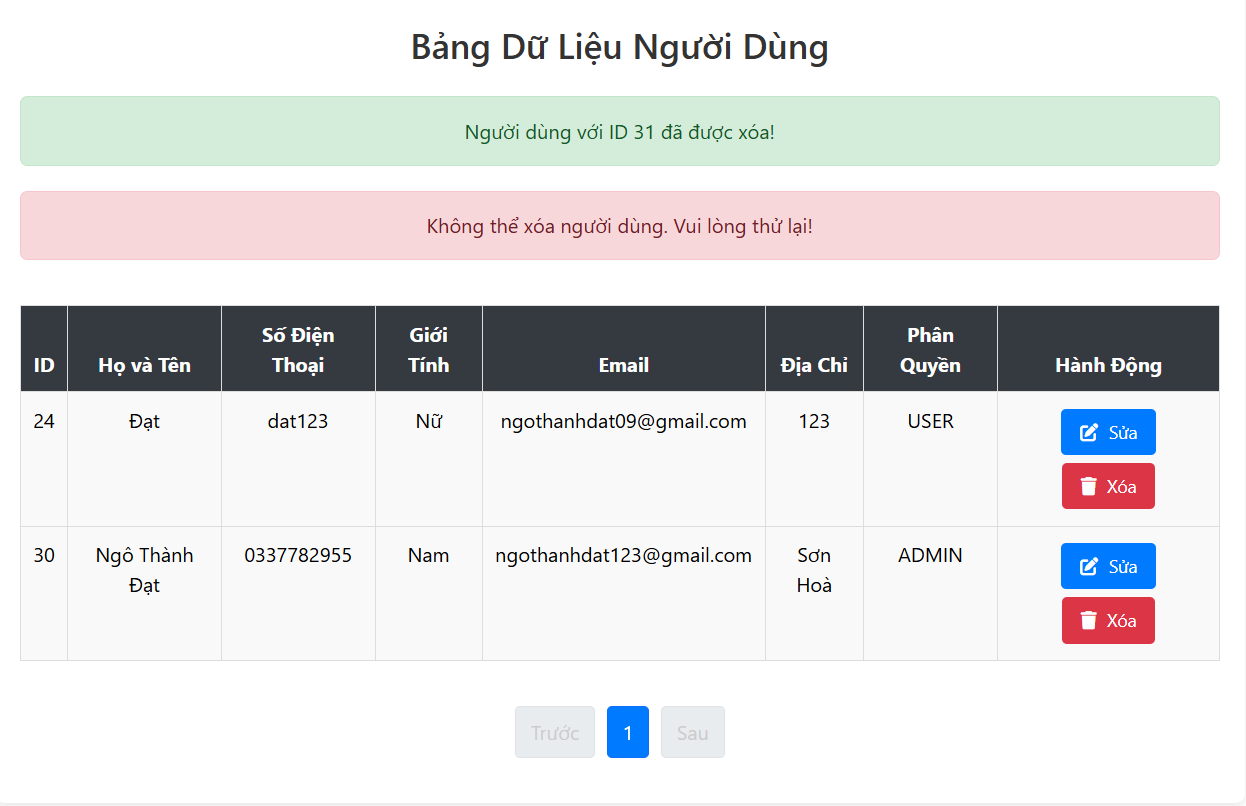
1. **Giao diện sửa thông tin người dùng.**



1. **Giao diện xoá người dùng.**







1. **Giao diện quản lý chương trình khuyến mãi**
2. **Giao diện đăng ký tài khoản.**



Hình 4. 7. Đăng ký tài khoản.

1. **Giao diện đăng nhập.**



Hình 4. 8. Giao diện đăng nhập.

1. **Giao diện đặt tour.**

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.**

Dự án 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.

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 như đặt vé xe, vé máy bay, khách sạn, homestay để cung cấp một hệ sinh thái du lịch toàn diện.

Việc tiếp tục mở rộng và nâng cấp hệ thống không chỉ giúp tăng cường tính cạnh tranh của nền tảng mà còn góp phần thúc đẩy sự phát triển của ngành du lịch trực tuyến tại Phú Yên, đáp ứng tốt hơn nhu cầu của du khách trong thời đại số.