**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**



**BÁO CÁO THỰC NGHIỆM**

**HỌC PHẦN: LẬP TRÌNH JAVA**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ CHO CÔNG TY DU LỊCH VIET TOURIST BẰNG JAVA SWING**

**NHÓM: 8**

***CBHD: TS. Hà Mạnh Đào***

**Sinh viên:**

1. **Ngô Xuân Tùng – 2022601704 (Nhóm trưởng)**
2. **Nguyễn Hữu Thắng – 2022601164**
3. **Nguyễn Đào Quốc Việt – 2022602428**

**Hà Nội, 2024**

***LỜI MỞ ĐẦU***

*Trong bối cảnh kinh tế phát triển mạnh mẽ đi cùng đó ngành du lịch cũng ngày càng phát triển. Tuy nhiên, nhiều người có kinh tế không có thời gian tìm hiểu booking cho bản thân mình những chuyến du lịch thư giãn sau nhiều ngày làm việc vất vả. Bên cạnh đó các công ty du lịch lại khó quản lý các chuyến du lịch, khách hàng, nhân viên…Chính bởi lý do đó việc ứng dụng công nghệ thông tin để nâng cao hiệu quả quản lý và phục vụ khách hàng đã trở thành yếu tố then chốt. Với nhu cầu đó, phần mềm quản lý cho công ty du lịch đóng vai trò quan trọng trong việc tối ưu hóa quy trình vận hành, từ quản lý thông tin khách hàng, tour du lịch, đến lập lịch trình và xử lý các yêu cầu dịch vụ.*

*Đề tài "Xây dựng phần mềm quản lý cho công ty du lịch Viet Tourist bằng Java Swing" được thực hiện nhằm giải quyết bài toán quản lý phức tạp, mang đến một giải pháp tiện lợi, dễ sử dụng, và hiệu quả cho công ty. Phần mềm này sẽ giúp tối ưu hóa các quy trình quản lý, từ cài đặt chuyến tham quan, quản lý khách hàng, cho đến quản lý nhân sự và tài chính. Việc sử dụng ngôn ngữ lập trình Java kết hợp thư viện giao diện đồ hoạ Swing, một công nghệ cài đặt giao diện người dùng mạnh mẽ và linh hoạt, phần mềm hướng đến việc tạo ra một hệ thống thân thiện, trực quan, đồng thời đáp ứng tốt các yêu cầu nghiệp vụ.*

*Qua bài tập lớn này, nhóm không chỉ hướng tới việc thực hành và ứng dụng các kiến thức đã học, mà còn mong muốn khám phá thêm về cách xây dựng và phát triển phần mềm thực tiễn, đồng thời nâng cao kỹ năng lập trình cũng như khả năng giải quyết vấn đề trong lĩnh vực công nghệ thông tin.*

*Bên cạnh đó, Nhóm 8 chúng em cảm ơn đến Giảng viên TS. Hà Mạnh Đào đã giúp đỡ nhóm hoàn thành báo cáo này và chúng em xin ghi nhận sự đóng góp từ thầy, các thành viên trong lớp cũng như các thầy cô trong khoa CNTT để chúng em hoàn thiện báo cáo hơn!*

MỤC LỤC

Chương 1. Tổng quan đề tài 4

1.1 Giới thiệu đề tài 4

1.2 Ý nghĩa thực tiễn của đề tài 4

1.3 Lý do chọn đề tài 5

1.4 Công cụ thực hiện đề tài 5

Chương 2. Phân tích thiết kế 0

2.1 Tài liệu đặc tả yêu cầu người dùng 0

**2.1.1 Yêu cầu chức năng** 0

**2.1.2 Yêu cầu phi chức năng** 1

2.2 Phân tích và thiết kế hệ thống 1

**2.2.1 Mô hình hoá chức năng** 1

**2.2.2 Mô tả chi tiết các use case** 1

2.2.2.1 Use case Đăng nhập (Ngô Xuân Tùng) 1

2.2.2.2 Use case Quản lý Tour (Ngô Xuân Tùng) 2

2.2.2.3 Use case Quản lý xe (Ngô Xuân Tùng) 3

2.2.2.4 Use case Quản lý nhân viên (Ngô Xuân Tùng) 4

2.2.2.5 Use case Quản lý khách sạn (Ngô Xuân Tùng) 5

# **Chương 1. Tổng quan đề tài**

## 1.1 Giới thiệu đề tài

Công ty Cổ phần Dịch vụ Thương mại Du lịch VietTourist luôn nỗ lực hết mình và không ngừng sáng tạo, đổi mới, phấn đấu trở thành công ty quy mô và chuyên nghiệp nhất trong lĩnh vực du lịch lữ hành.

* Địa chỉ:  91-93 Lê Quốc Hưng, Phường 13, Quận 4, Thành phố Hồ Chí Minh; 58 Nguyễn Trường Tộ - Ba Đình, Hà Nội
* Điện thoại: 19001868 – 0909886688; 1900 1868 - Nhánh 2 - 0931883688

Với đề tài xây dựng phần mềm quản lý chuyến du lịch công ty Du lịch VietTourist, nhiệm vụ và công việc chính đề ra gồm có:

* Khảo sát sơ bộ: tìm hiểu các yếu tố cơ bản tạo tiền đề xây dựng phần mềm phù hợp với công ty.
* Khảo sát chi tiết: thu thập thông tin chi tiết của hệ thống (chức năng xử lý, thông tin được phép nhập và xuất khỏi hệ thống, ràng buộc, giao diện cơ bản, nghiệp vụ) phục vụ cho việc phân tích và thiết kế.

Hình thức sản phẩm: Phần mềm ứng dụng

Kết quả đạt được: Xây dựng phần mềm quản lý du lịch có đầy đủ chức năng cần thiết cho người quản lý và khách hàng sử dụng, giao diện thân thiện với người dùng, dễ dàng sử dụng và hạn chế lỗi xảy ra.

## 1.2 Ý nghĩa thực tiễn của đề tài

* Những hoạt động quản lý du lịch ở các công ty du lịch như kiểm soát những vé đã đặt của du khách, quản lý nhân viên hay quản lý xe có thể trong quá trình làm những công việc đó gặp phải nhầm lẫn, sai sót.
* Quản lý du lịch đòi hỏi sự chính xác, khoa học. Nếu không kiểm soát được tối đa hệ thống quản lý du lịch, sẽ gây ra rất nhiều thất thoát, tốn kém.
* Việc quản lý các thông tin cần thiết của công ty có vai trò quan trọng trong sự phát triển của cửa hàng đó. Giúp cho độ uy tín của công ty được đẩy lên cao và sẽ thu hút du khách hơn.
* Việc quản lý du lịch bằng phần mềm giúp quản lý một cách tổng quát, đảm bảo tính chính xác trong quá trình quản lý, giúp cho việc quản lý trở nên dễ dàng và khoa học hơn.

## 1.3 Lý do chọn đề tài

Giúp Công ty Du lịch VietTourist có một giải pháp hiệu quả để quản lý thông tin khách hàng, đặt tour và thanh toán do việc quản lý thủ công không còn đáp ứng được yêu cầu. Xây dựng phần mềm quản lý bằng Java Swing sẽ giúp tự động hóa quy trình, nâng cao hiệu suất và giảm thiểu sai sót. Đồng thời, dự án này cũng tạo cơ hội để áp dụng kiến thức về Java và Swing vào thực tế, mang lại giá trị thực tiễn cho công ty.

## 1.4 Công cụ thực hiện đề tài

* **Eclipse**

**Eclipse** là một môi trường phát triển tích hợp (IDE – Integrated Development Environment), mã nguồn mở dùng cho lập trình máy tính. Nó chứa một không gian làm việc cơ sở và một hệ thống plug-in để mở rộng để tùy chỉnh môi trường. Eclipse được viết chủ yếu bằng Java và nó được dùng chủ yếu cho lập trình ứng dụng Java, nhưng nó cũng có thể dùng để lập trình ứng dụng bằng các ngôn ngữ khác thông qua plug-ins, bao gồm C, C++, C#, Fortran, Groovy, Haskell, HTML, JavaScript, PHP, Python, Ruby (Bao gồm Ruby on Rails framework). Nó cũng có thể dùng để phát triển các tài liệu bằng LaTeX (thông qua một plug-in TeXlipse) và các gói tin cho phần mềm Mathematica. Môi trường phát triển bao gồm Eclipse Java development tools (JDT) cho Java và Scala, Eclipse CDT cho C/C++, và Eclipse PDT for PHP, và những gói khác. Eclipse cung cấp bộ công cụ phong phú giúp các nhà phát triển có thể viết, biên dịch, gỡ lỗi và triển khai ứng dụng một cách hiệu quả.

Các tính năng chính của Eclipse bao gồm:

* Hỗ trợ Java SE, Java EE và JavaFX: Eclipse cung cấp công cụ mạnh mẽ để phát triển ứng dụng desktop, web và di động bằng Java.
* Hỗ trợ nhiều ngôn ngữ lập trình: Ngoài Java, Eclipse còn hỗ trợ các ngôn ngữ khác như PHP, C/C++, HTML5, JavaScript, Groovy, và nhiều ngôn ngữ khác thông qua các plugin.
* Giao diện người dùng thân thiện: Eclipse có giao diện trực quan, dễ sử dụng với nhiều công cụ kéo thả giúp tăng tốc độ phát triển ứng dụng.
* Công cụ quản lý dự án: Eclipse hỗ trợ quản lý dự án, kiểm soát phiên bản (như Git, SVN), và tích hợp tốt với các hệ thống xây dựng (như Maven, Ant).
* Biên tập mã thông minh: Eclipse cung cấp tính năng gợi ý mã, kiểm tra lỗi cú pháp, và hỗ trợ tự động hoàn thành mã, giúp lập trình viên viết mã nhanh chóng và chính xác hơn.
* Cộng đồng và tài liệu phong phú: Eclipse có một cộng đồng phát triển mạnh mẽ và nhiều tài liệu, plugin giúp người dùng dễ dàng mở rộng và tùy chỉnh IDE theo nhu cầu.

Eclipse thường được sử dụng trong môi trường giáo dục và công nghiệp để phát triển các ứng dụng từ nhỏ đến lớn, nhờ vào sự linh hoạt và mạnh mẽ của nó.

* **MySQL**

**MySQL** là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, …MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

* **Draw.io**

**Draw.io** là các mô hình hoặc bản mẫu trực quan của một sản phẩm, thường được sử dụng trong thiết kế giao diện người dùng (UI/UX) và phát triển sản phẩm kỹ thuật số. Draw.io thể hiện cấu trúc, bố cục và thẩm mỹ của sản phẩm, giúp hình dung rõ ràng về cách nó sẽ trông như thế nào sau khi hoàn thiện.

*Các đặc điểm chính của Draw.io:*

* Mô tả chi tiết giao diện: Draw.io thường hiển thị các yếu tố giao diện như nút bấm, hình ảnh, văn bản, và bố cục tổng thể của trang hoặc ứng dụng. Chúng giúp truyền tải một cách chi tiết hơn về hình thức và bố trí của sản phẩm so với wireframes (khung sườn).
* Không có chức năng thực sự: Draw.io chỉ là các hình ảnh tĩnh hoặc bản mẫu, không có chức năng hoạt động như sản phẩm hoàn thiện. Chúng không thể được tương tác như một nguyên mẫu (prototype) có thể có chức năng cơ bản.
* Tập trung vào thiết kế và thẩm mỹ: Các Draw.io thường tập trung vào màu sắc, kiểu chữ, hình ảnh, và các yếu tố thiết kế khác để thể hiện giao diện cuối cùng. Điều này giúp các nhà thiết kế, nhà phát triển và khách hàng có thể đánh giá và điều chỉnh các yếu tố thẩm mỹ trước khi chuyển sang giai đoạn phát triển hoặc lập trình.
* Dùng trong giai đoạn sớm của phát triển: Draw.io thường được sử dụng ở giai đoạn đầu của quá trình thiết kế để thảo luận và lấy ý kiến từ các bên liên quan trước khi bắt đầu phát triển chi tiết sản phẩm.

Draw.io là một bước quan trọng trong quá trình phát triển sản phẩm kỹ thuật số vì chúng giúp truyền đạt ý tưởng thiết kế và đảm bảo rằng tất cả các bên liên quan đều hiểu rõ và đồng ý về giao diện trước khi bắt đầu công việc phát triển chi tiết hơn.

* + - **CASE Studio**

**Case Studio** là một công cụ phần mềm được sử dụng để thiết kế, mô hình hóa và quản lý cơ sở dữ liệu. Đây là một công cụ phổ biến trong lĩnh vực phát triển phần mềm và quản trị cơ sở dữ liệu, đặc biệt là trong việc tạo ra các mô hình dữ liệu trực quan, thường được gọi là Entity-Relationship Diagrams (ERD) hoặc Database Schemas.

*Các tính năng chính của Case Studio bao gồm:*

* Mô hình hóa cơ sở dữ liệu: Case Studio cho phép người dùng thiết kế cơ sở dữ liệu bằng cách sử dụng các mô hình ERD. Các thực thể (entity), thuộc tính (attributes), và mối quan hệ (relationships) giữa các thực thể có thể được định nghĩa rõ ràng.
* Hỗ trợ nhiều hệ quản trị cơ sở dữ liệu: Case Studio hỗ trợ nhiều hệ quản trị cơ sở dữ liệu khác nhau như MySQL, PostgreSQL, Oracle, Microsoft SQL Server, v.v. Điều này cho phép người dùng thiết kế mô hình dữ liệu tương thích với nhiều nền tảng khác nhau.
* Tạo mã SQL tự động: Một trong những lợi ích chính của Case Studio là khả năng tự động tạo mã SQL từ mô hình cơ sở dữ liệu. Điều này giúp giảm thiểu sai sót khi chuyển từ thiết kế sang triển khai thực tế.
* Phân tích và tối ưu hóa: Case Studio cung cấp các công cụ để phân tích và tối ưu hóa mô hình dữ liệu, đảm bảo rằng cơ sở dữ liệu được thiết kế hiệu quả và hoạt động tốt.
* Quản lý phiên bản: Case Studio có thể giúp theo dõi và quản lý các phiên bản khác nhau của mô hình cơ sở dữ liệu, giúp người dùng dễ dàng theo dõi các thay đổi và kiểm soát quá trình phát triển.
* Tài liệu hóa tự động: Công cụ này có khả năng tạo ra tài liệu tự động từ mô hình dữ liệu, giúp đơn giản hóa quá trình tài liệu hóa cơ sở dữ liệu.
  + - Lợi ích của việc sử dụng Case Studio:
  + Hiệu quả: Tăng tốc quá trình thiết kế và phát triển cơ sở dữ liệu nhờ các công cụ tự động hóa và hỗ trợ đa nền tảng.
  + Chính xác: Giảm thiểu rủi ro và sai sót trong thiết kế cơ sở dữ liệu bằng cách sử dụng các mô hình trực quan và mã tự động.
  + Hợp tác: Dễ dàng chia sẻ và làm việc nhóm với các mô hình dữ liệu, cải thiện sự phối hợp giữa các nhà phát triển và quản trị viên cơ sở dữ liệu.

Mặc dù Case Studio từng là một công cụ phổ biến, nhưng hiện tại, nhiều công cụ mới hơn và tiên tiến hơn đã thay thế nó, chẳng hạn như MySQL Workbench, ER/Studio, Microsoft Visio, hoặc Oracle SQL Developer Data Modeler.

* + - **IBM Rational Rose**

**IBM Rational Rose** là một công cụ lập mô hình trực quan mạnh trợ giúp bạn phân tích và thiết kế các hệ thống phần mềm hướng đối tượng. Nó được dùng để lập mô hình hệ thống trước khi bạn viết mã lệ. Với việc sử dụng mô hình, bạn có thể giảm bớt những thiếu sót về thiết kế. Mô hình Rose là một bức tranh về một hệ thống từ nhiều góc nhìn khác nhau. Nó bao gồm tất cả các sơ đồ UML, các actor, các use case, các đối tượng, các lớp, các thành phần… Nó mô tả chi tiết nội dung mà hệ thống sẽ gộp và cách nó sẽ làm việc. Có thể xem một mô hình Rose tương tự như bản thiết kế mẫu. Một mô hình Rose chứa đựng các sơ đồ khác nhau cho phép các thành viên trong nhóm đề án xem hệ thống từ các góc nhìn khác nhau như: khách hàng, nhà thiết kế, quản trị đề án, … Khi đã có được bản thiết kế thì sẽ giảm bớt một số vấn đề phiền phức như: khi lập trình theo truyền thống thì khi hoàn tất đề án, sau một thời gian sử dụng khách hàng yêu cầu thêm một vài chức năng nào đó vì có cập nhật mới thì người lập trình phải xem lại toàn bộ hệ thống rồi sau đó mới cập nhật. Điều này tốn rất nhiều thời gian. Nay nhờ có bản thiết kế thì chỉ cần xem cập nhật đó nằm ở phần nào và chỉnh sửa, nâng cấp hệ thống. Điều đó sẽ linh hoạt và giảm rất nhiều thời gian…

Có ba phiên bản khác nhau của Rose:

* Rose Modeler: cho phép bạn tạo mô hình cho hệ thống, nhưng không hỗ trợ tiến trình phát sinh mã hoặc thiết kế kỹ thuật đảo ngược.
* Rose Professional: cho phép bạn phát sinh mã trong một ngôn ngữ.
* Rose Enterprise: cho phép bạn phát sinh mã cho C++, Java, Ada, Corba, Visual Basic, Oracle … Một mô hình có thể có các thành phần được phát sinh bằng các ngôn ngữ khác nhau.

# **Chương 2. Phân tích thiết kế**

## 2.1 Tài liệu đặc tả yêu cầu người dùng

### **2.1.1 Yêu cầu chức năng**



*Hình … : Các yêu cầu chức năng của hệ thống*

* Đối với người dùng:
* Đăng nhập: Cho phép người dùng đăng nhập vào tài khoản đã đăng ký trước đó bằng cách cung cấp thông tin đăng nhập như tên tài khoản và mật khẩu để thực hiện các chức năng tương ứng với quyền của mình.
* Xem tour hiện hành: Cung cấp cho du khách xem thông tin chi tiết về các tour du lịch trong hệ thống.
* Đặt tour: Cho phép người dùng đặt tour mong muốn một cách nhanh chóng.
* Xem tour đã đặt: Cung cấp cho người dùng thông tin chuyến du lịch du khách đã đặt như kinh phí, địa điểm, phương tiện .... của tour du lịch mà khách hàng đã đặt.
* Đổi mật khẩu: Cho phéo người dùng đổi mật khẩu tài khoản khi
* Đăng xuất: Cho phép người dùng đăng xuất khỏi tài khoản hiện tại, kết thúc phiên làm việc và trở về trạng thái người dùng chưa đăng nhập.
* Đối với quản trị viên:
* Đăng nhập: Hệ thống cho phép quản trị viên đăng nhập để thực hiện các chức năng tương ứng với quyền của mình.
* Đăng xuất: Hệ thống cho phép quản trị viên đăng xuất tài khoản, kết thúc phiên làm việc và trở về màn hình chưa đăng nhập.
* Quản lý (nhân viên, xe, khách sạn, tour, vé, tài khoản): Quản lý và cập nhật thông tin (nhân viên, xe, khách sạn, tour, vé, tài khoản của admin và user): xem danh sách, thêm, sửa, xoá.

### **2.1.2 Yêu cầu phi chức năng**

* Tính bảo mật: Nhân viên quản lý chỉ đăng nhập được bằng một tài khoản duy nhất được công ty cung cấp.
* Tính dễ sử dụng: Giao diện hài hoà thân thiện, ngôn ngữ đa dạng dễ sử dụng.
* Tính ổn định: Hệ thống xử lý nhanh gọn, thuận tiện, 24/24 và có thể thêm chức năng cần thiết sau này. Xử lý lỗi ngay khi ghi nhận được ý kiến đóng góp, phản ánh của khách hàng.
* Độ tin cậy: Đảm bảo thông tin chính xác, không bị rò rỉ thông tin của khách hàng, nhân viên và công ty.
* Sử dụng được trên nhiều hệ điều hành như: Windows, MacOS, Androids, IOS...

## 2.2 Phân tích và thiết kế hệ thống

### **2.2.1 Mô hình hoá chức năng**

* Các tác nhân của hệ thống: Người dùng – Du khách (User), Quản trị viên (Admin).
* Các use case chính:
* Đối với Quản trị viên (Admin).

Use case: Đăng nhập, Quản lý tour, Quản lý tài khoản, Quản lý khách sạn, Quản lý nhân viên, Quản lý xe, Quản lý vé, Thống kê vé.

* Đối với Người dùng – Du khách (User).

Use case: Đăng nhập, Đổi mật khẩu, Tìm kiếm tour theo tên, Đặt tour Xem tour đã đặt.

**Hình …Biểu đồ use case tổng quát**

### **2.2.2 Mô tả chi tiết các use case**

#### *Use case Đăng nhập (Ngô Xuân Tùng)*

1. Mô tả vắn tắt

Use case này cho phép quản trị viên đăng nhập vào hệ thống

1. Luồng sự kiện
2. Luồng cơ bản
3. Use case này bắt đầu khi hệ thống khởi chạy, hệ thống sẽ hiển thị một màn hình yêu cầu nhập các thông tin tên đăng nhập, mật khẩu.
4. Quản trị viên nhập thông tin về tài khoản của mình tên đăng nhập, mật khẩu và kích vào nút “Đăng nhập”. Hệ thống sẽ hiển thị màn hình trang chủ admin quản lý du lịch. Use case kết thúc.
5. Luồng rẽ nhánh

2.2.1 Đăng nhập thất bại

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 2 trong luồng cơ bản, nếu người dùng điền sai thông tin tên đăng nhập hoặc mật khẩu, hệ thống sẽ hiển thị một thông báo “Tên đăng nhập hoặc mật khẩu không chính xác” và use case kết thúc.
3. Tại bước 1 khi người dùng để trống tên tài khoản hoặc mật khẩu, hệ thống sẽ hiển thị thông báo và yêu cầu nhập lại và use case kết thúc.
4. Tại bước 2 trong luồng cơ bản, nếu người quản lý kích nút “Hủy” trên màn hình. Hệ thống sẽ thoát khỏi màn hình đăng nhập. Use case kết thúc.

2.2.2. Quản trị viên thoát khỏi trang đăng nhập:

Quản trị viên nhấn “Huỷ” ở bước 2 luồng cơ bản trên màn hình. Hệ thống sẽ thoát khỏi màn hình đăng nhập. Use case kết thúc.

1. Các yêu cầu đặc biệt

Không có

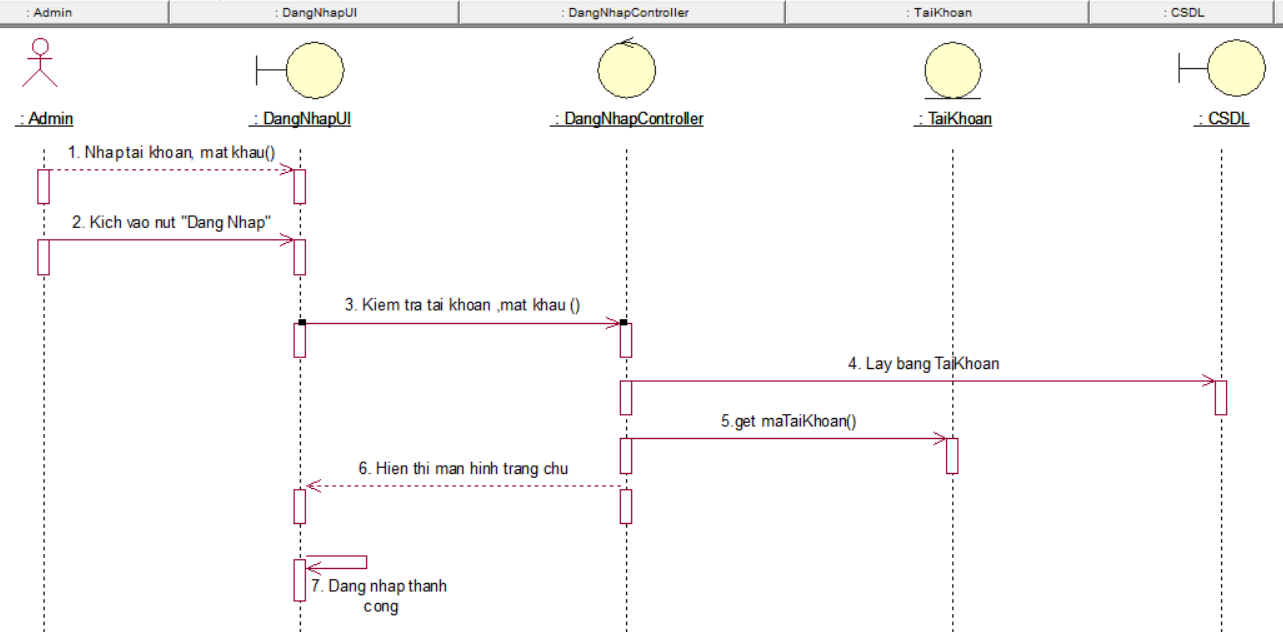
1. Tiền điều kiện

Quản trị viên chưa đăng nhập

1. Hậu điều kiện
2. Nếu đăng nhập thành công người quản trị được chuyển đến màn hình chính.
3. Nếu đăng nhập thất bại người dùng được chuyển đến trang chủ của giao diện đăng nhập.
4. Điểm mở rộng

Không có.

1. Biểu đồ trình tự



1. *Use case Quản lý Tour (Ngô Xuân Tùng)*
   1. Mô tả vắn tắt Use case cho phép nhân viên quản lý tìm kiếm, thêm, sửa, xóa các tour trong hệ thống.
   2. Luồng sự kiện
2. Luồng cơ bản
   * 1. Use case này bắt đầu khi nhân viên quản lý đăng nhập vào tài khoản Admin. Hệ thống sẽ hiển thị:

* Các trường nhập thông tin gồm: Mã tour, tên tour, thời gian, địa điểm, mã xe, mã khách sạn.
* Thông tin dữ liệu tour gồm: Mã tour, tên tour, thời gian, địa điểm, mã xe, mã khách sạn được được hiển thị dưới dạng bảng.
  + 1. Thêm:

Nhân viên quản lý nhập thông tin của tour vào các trường nhập. Sau đó kích vào nút “Thêm”. Hệ thống sẽ tạo ra một bản ghi mới trong bảng Tour và hiển thị lại thông tin Tour lên màn hình

* + 1. Cập nhật:
* Nhân viên quản lý bấm vào một dòng dữ liệu Tour. Hệ thống sẽ lấy dữ liệu nhập lên các trường dữ liệu, nhân viên sẽ sửa những trường muốn sửa.
* Sau đó kích vào nút “Sửa”. Hệ thống sẽ cập nhật lại thông tin của tour đó vào bảng Tour và hiển thị danh sách lên màn hình.
  + 1. Xóa:

Nhân viên quản lý bấm vào một dòng dữ liệu Tour. Sau đó bấm nút “Xóa”. Hệ thống sẽ hiển thị lên màn hình một thông báo “Xác nhận xóa”. Bấm “Đồng ý”. Hệ thống sẽ xóa dữ liệu tour và cập nhật lại bảng Tour và hiển thị danh sách lên màn hình.

1. Luồng rẽ nhánh
   * + 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
       2. Tại bước 2 trong luồng cơ bản nếu nhân viên quản lý không điền đầy đủ thông tin vào các trường thì hệ thống sẽ hiển thị thông báo “Bạn chưa nhập đầy đủ thông tin” và yêu cầu nhập lại và use case kết thúc.
   1. Các yêu cầu đặc biệt

Use case này chỉ cho phép nhân viên quản lý thực hiện

* 1. Tiền điều kiện

Nhân viên quản lý cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

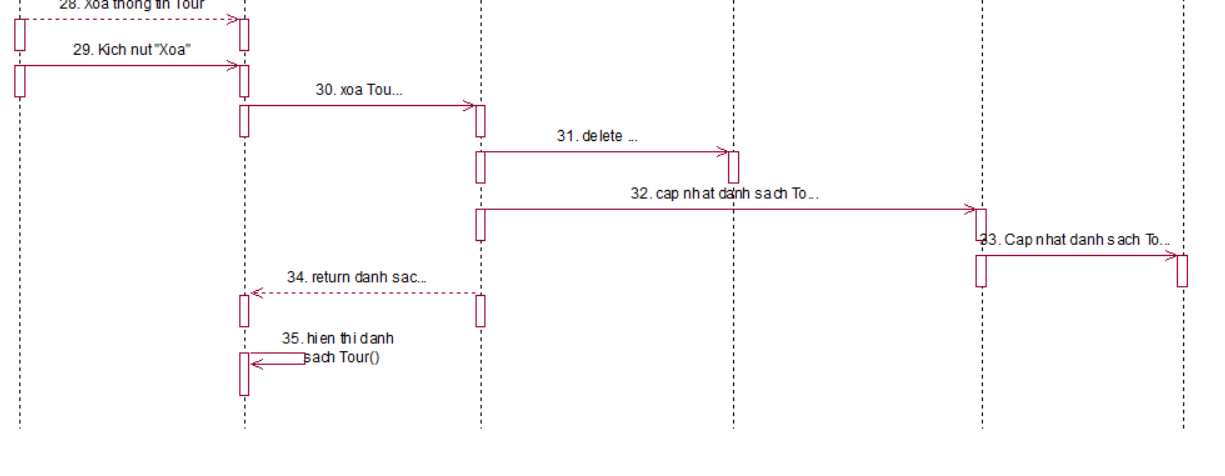
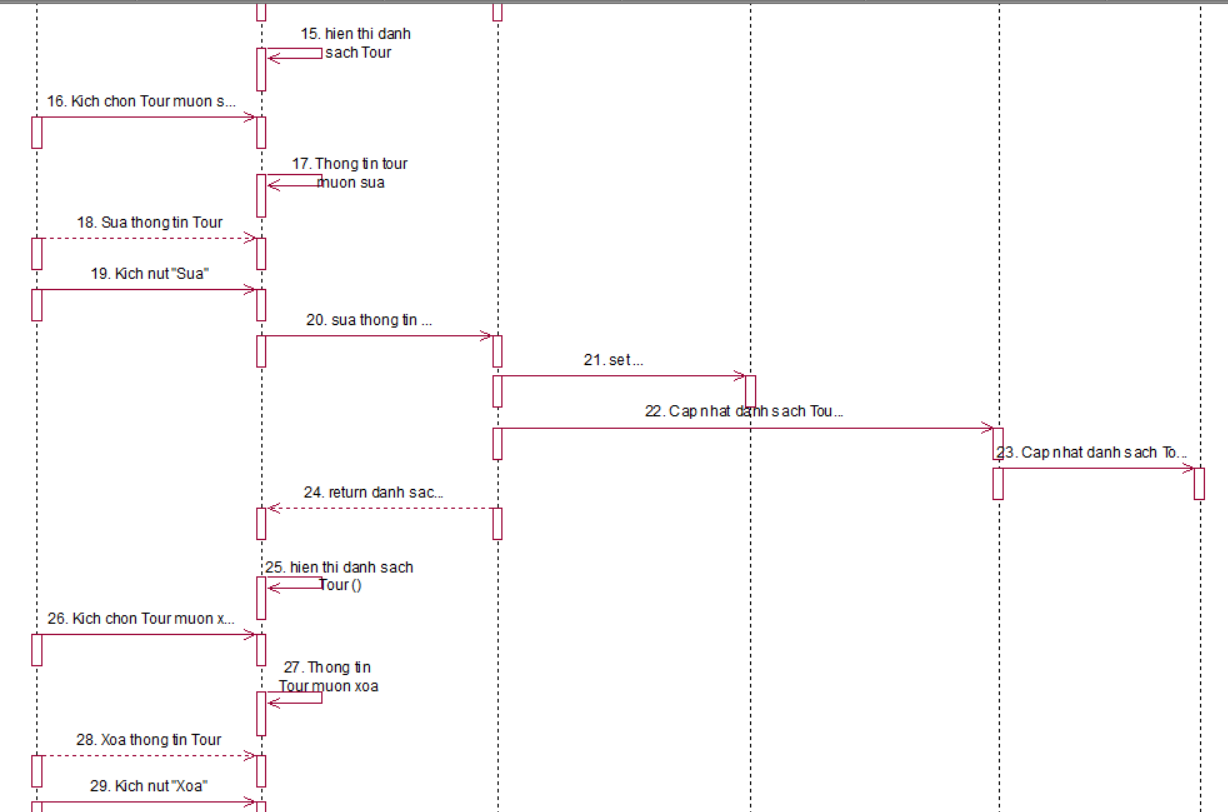
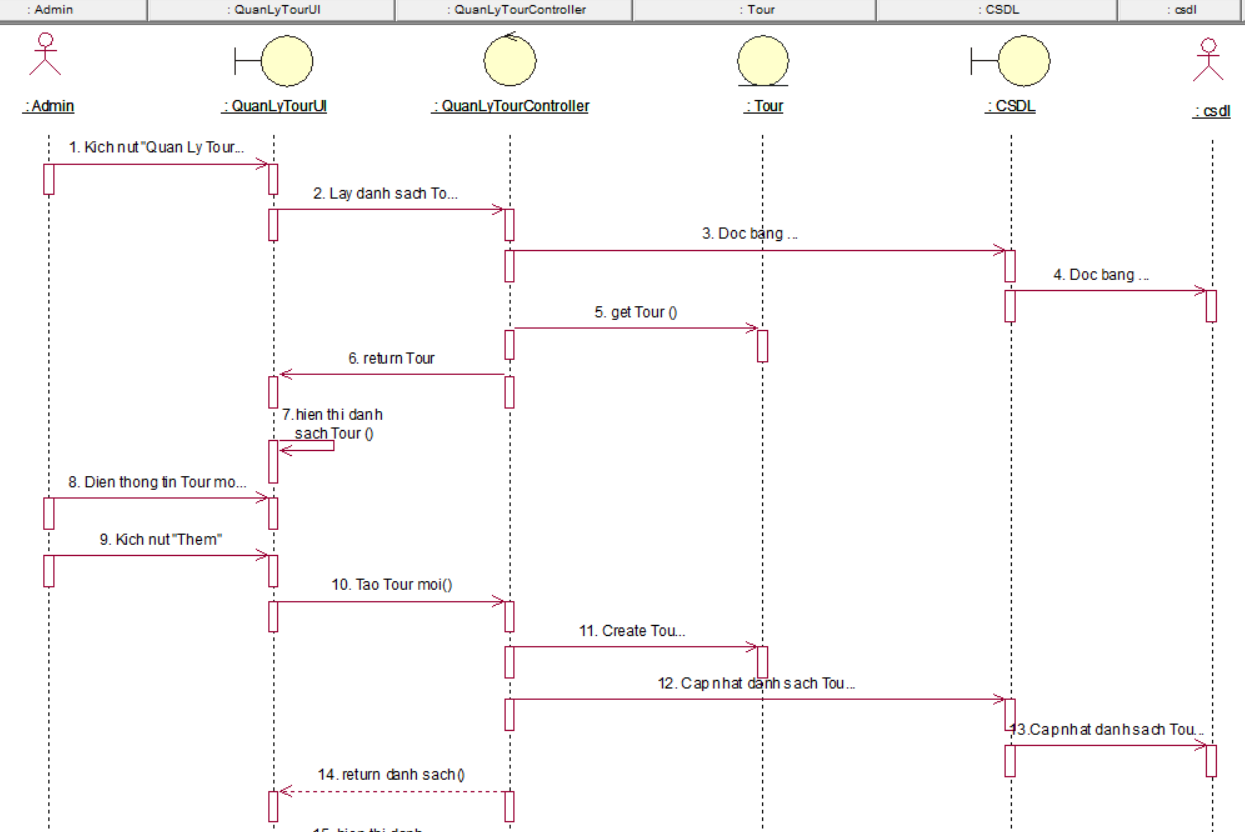
* 1. Hậu điều kiện

Không có

* 1. Điểm mở rộng

Không có

* 1. Biểu đồ trình tự



1. *Use case Quản lý xe (Ngô Xuân Tùng)*
   * + - 1. Mô tả vắn tắt
         2. Luồng sự kiện
2. Luồng cơ bản
3. Use case này bắt đầu khi người quản lý kích vào “Quản lý xe” trên thanh menu quản lý.

* Hệ thống sẽ lấy dữ liệu từ bảng Xe thông tin: Mã xe, loại xe, biển số.
* Hệ thống sẽ hiển thị trường nhập thông tin cho phép người quản lý nhập: Mã xe, loại xe, biển số.

1. Thêm:

Nhân viên quản lý nhập thông tin của xe vào trường nhâp, sau đó kích nút “Thêm”. Hệ thống sẽ tạo ra một bản ghi mới trong bảng Xe và hiển thị lại thông tin lên màn hình.

1. Sửa:

Nhân viên quản lý bấm vào một dòng dữ liệu của xe. Hệ thống sẽ lấy dữ liệu nhập lên các trường dữ liệu, sau đó nhân viên bấm nút “Sửa”. Hệ thống sẽ cập nhật lại thông tin và hiển thị lên màn hình

1. Xóa:

* Nhân viên quản lý kích vào một dòng dữ liệu trong danh sách xe, sau đó bấm nút “Xóa”. Hệ thống sẽ hiển thị một màn hình yêu “Xác nhận xóa”
* Nhân viên quản lý kích vào nút “Đồng ý”. Hệ thống sẽ xóa xe khỏi bảng Xe và hiển thị danh sách các phòng lên màn hình. Use case kết thúc.

1. Luồng rẽ nhánh

1) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

2) Tại bước 4b trong luồng cơ bản nếu nhân viên quản lý kích vào nút “Hủy” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các Xe và use case kết thúc.

* + - * 1. Các yêu cầu đặc biệt.

Use case này chỉ cho phép nhân viên quản lý thực hiện

* + - * 1. Tiền điều kiện.

Quản trị viên cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case

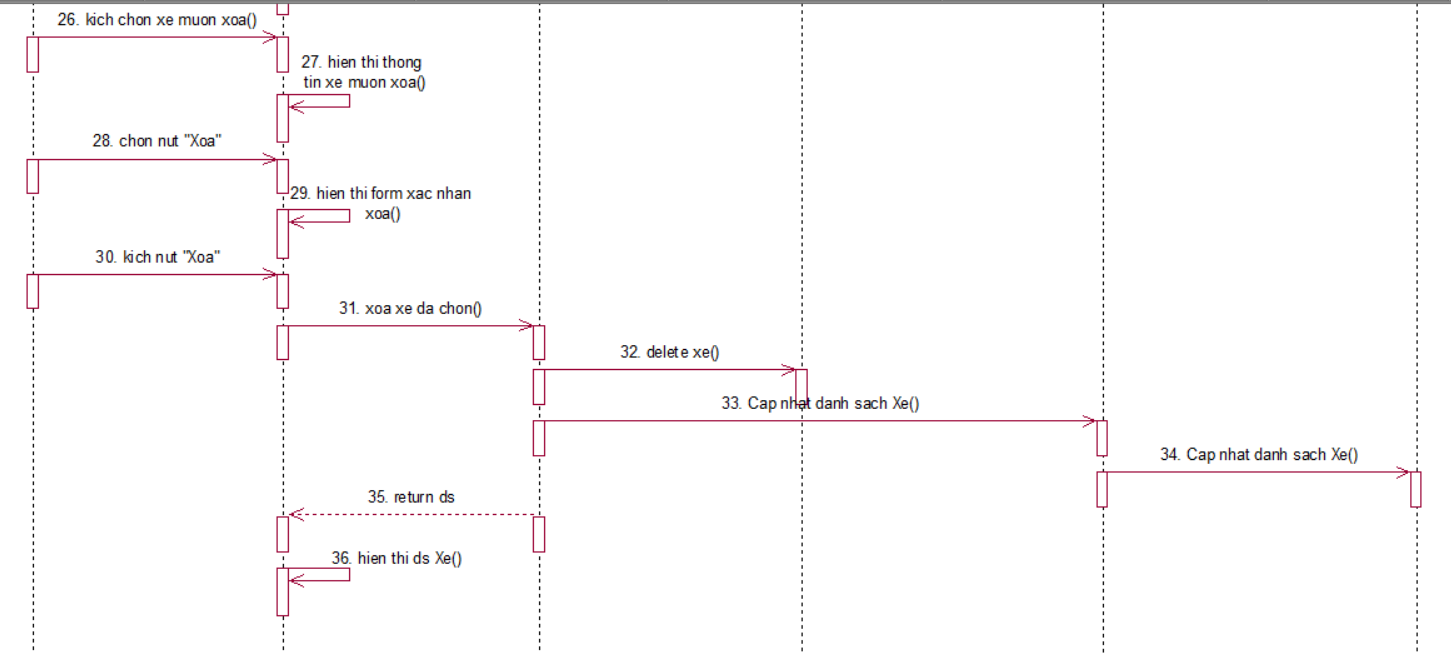
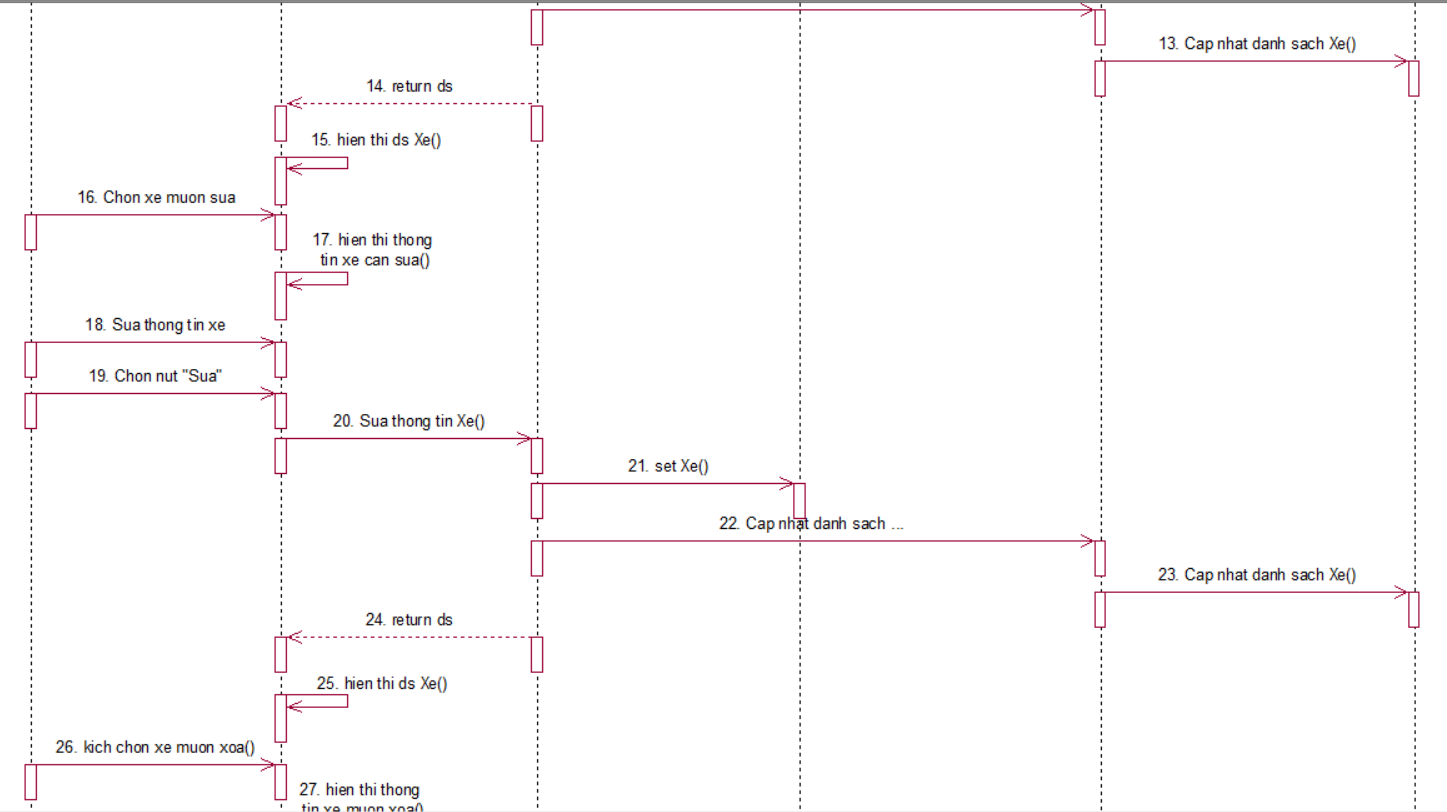
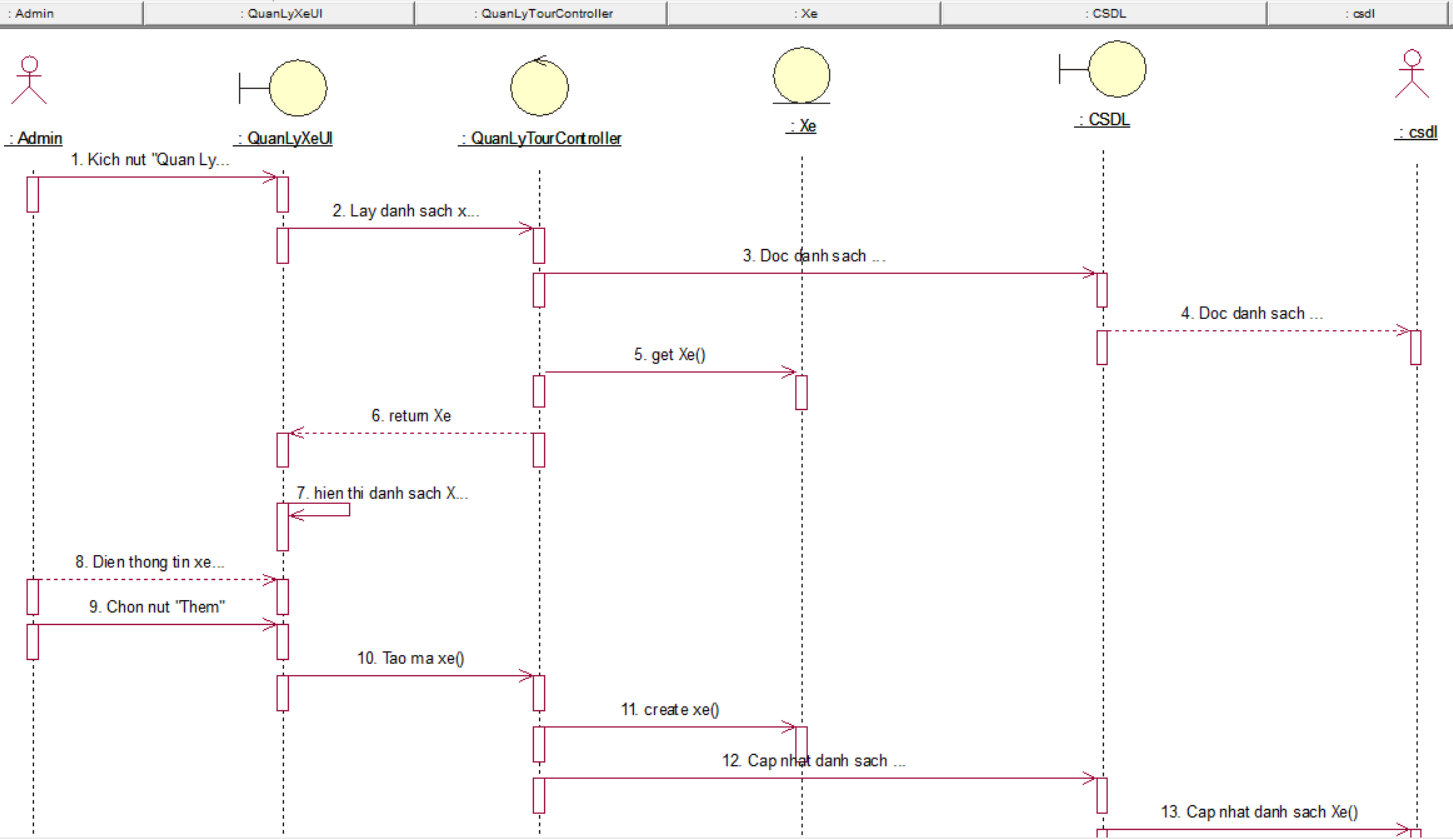
* + - * 1. Hậu điều kiện.

Không có

* + - * 1. Điểm mở rộng.

Không có

* + - * 1. Biểu đồ trình tự



1. Use case Quản lý nhân viên (Ngô Xuân Tùng)

Mô tả vắn tắt

Use case này cho phép nhân viên quản lý xem, thêm, sửa, xóa các nhân viên có trong hệ thống.

Luồng sự kiện

* 1. Luồng cơ bản

1. Use case này bắt đầu khi người quản lý kích vào “Quản lý nhân viên” trên thanh menu quản lý.

* Hệ thống sẽ lấy dữ liệu từ bảng NhanVien thông tin: Mã nhân viên, tên nhân viên, số điện thoại, địa chỉ, ngày sinh, giới tính, email và hiển thị ra màn hình.
* Hệ thống sẽ hiển thị trường nhập thông tin cho phép người quản lý nhập: Mã nhân viên, tên nhân viên, số điện thoại, địa chỉ, ngày sinh, giới tính, email.

1. Thêm:

Nhân viên quản lý nhập thông tin của nhân viên vào trường nhập, sau đó kích nút “Thêm”. Hệ thống sẽ tạo ra một bản ghi mới trong bảng NhanVien và hiển thị lại thông tin lên màn hình.

1. Sửa:

Nhân viên quản lý bấm vào một dòng dữ liệu của nhân viên. Hệ thống sẽ lấy dữ liệu nhập lên các trường dữ liệu, sau đó nhân viên bấm nút “Sửa”. Hệ thống sẽ cập nhật lại thông tin và hiển thị lên màn hình

1. Xóa:

* Nhân viên quản lý kích vào một dòng dữ liệu trong danh sách nhân viên, sau đó bấm nút “Xóa”. Hệ thống sẽ hiển thị một màn hình yêu “Xác nhận xóa”
* Nhân viên quản lý kích vào nút “Đồng ý”. Hệ thống sẽ xóa nhân viên khỏi bảng NhanVien và hiển thị danh sách các nhân viên lên màn hình. Use case kết thúc.

1. Luồng rẽ nhánh:
   1. Luồng rẽ nhánh
2. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
3. Tại bước 4b trong luồng cơ bản nếu nhân viên quản lý kích vào nút “Hủy” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các nhân viên và use case kết thúc.

Các yêu cầu đặc biệt:

Use case này chỉ cho phép nhân viên quản lý thực hiện

Tiền điều kiện:

Nhân viên quản lý cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case

Hậu điều kiện:

Không có

Điểm mở rộng:

Không có

Biểu đồ trình tự

A screenshot of a computer

Description automatically generatedA diagram of a company

Description automatically generated with medium confidenceA diagram of a computer

Description automatically generated

1. Use case Quản lý khách sạn (Ngô Xuân Tùng)

Mô tả vắn tắt

Use case này cho phép nhân viên quản lý Xem, Thêm, Sửa, Xóa đối với các khách sạn có trong hệ thống.

Luồng sự kiện

* 1. Luồng cơ bản

1. Use case này bắt đầu khi nhân viên quản lý kích vào “Quản lý khách sạn” trong menu quản lý. Hệ thống sẽ:

* Lấy dữ liệu khách sạn từ bảng KhachSan gồm: Mã khách sạn, tên khách sạn, địa điểm, và hiển thị ra màn hình.
* Hiển thị ra các trường dữ liệu nhập gồm: Mã khách sạn, tên khách sạn, địa điểm.

1. Thêm:

Nhân viên quản lý nhập thông tin của khách sạn vào trường nhập, sau đó kích nút “Thêm”. Hệ thống sẽ tạo ra một bản ghi mới trong bảng KhachSan và hiển thị lại thông tin lên màn hình.

1. Sửa:

Nhân viên quản lý bấm vào một dòng dữ liệu của các khách sạn. Hệ thống sẽ lấy dữ liệu nhập lên các trường dữ liệu, sau đó nhân viên bấm nút “Sửa”. Hệ thống sẽ cập nhật lại thông tin và hiển thị lên màn hình

1. Xóa:

* Nhân viên quản lý kích vào một dòng dữ liệu trong danh sách khách sạn, sau đó bấm nút “Xóa”. Hệ thống sẽ hiển thị một màn hình yêu “Xác nhận xóa”
* Nhân viên quản lý kích vào nút “Đồng ý”. Hệ thống sẽ xóa khách sạn khỏi bảng KhachSan và hiển thị danh sách các khách sạn lên màn hình. Use case kết thúc.
  1. Luồng rẽ nhánh

1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.
2. Tại bước 2b của luồng cơ bản nếu nhân viên quản lý chưa nhập số liệu hoặc nhập số liệu không hợp lệ thì hệ thống sẽ thông báo và yêu cầu nhập lại và use case kết thúc.
3. Tại bước 2c của luồng cơ bản nếu nhân viên quản lý kích vào nút “Hủy” hệ thống sẽ bỏ qua thao tác chuyển và quay trở lại màn hình trang chủ quản trị và use case kết thúc.

Các yêu cầu đặc biệt

Use case này chỉ cho phép nhân viên quản lý thực hiện

Tiền điều kiện

Nhân viên quản lý cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case

Hậu điều kiện

Không có

Điểm mở rộng

Không có

Biểu đồ trình tự

