

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI: XÂY DỰNG WEBSITE DU LỊCH

Giảng viên hướng dẫn: PHẠM THỊ MIÊN

Sinh viên thực hiện: BIÊN THANH PHONG

Lớp : CQ.CNTT.K59

Khoá : 59

Tp. Hồ Chí Minh, năm 2022

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI: XÂY DỰNG WEBSITE DU LỊCH

Giảng viên hướng dẫn: PHẠM THỊ MIÊN

Sinh viên thực hiện: BIÊN THANH PHONG

Lớp : CQ.CNTT.K59

Khoá : 59

Tp. Hồ Chí Minh, năm 2022

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP
BỘ MÔN: CÔNG NGHỆ THÔNG TIN

-----***-----

Mã sinh viên: 5951071073

Họ tên SV: BIỆN THANH PHONG

Khóa: 59

Lớp: CQ.CNTT.K59

1. Tên đề tài: XÂY DỰNG WEBSITE DU LỊCH

2. Mục đích, yêu cầu:

Hiểu được các khía cạnh, các vấn đề của du lịch trong đời sống từ đó xây dựng thành công website du lịch.

3. Nội dung và phạm vi đề tài

a. Nội dung:

Tổng quan về bài toàn quản lý du lịch.

Phân tích thiết kế phần mềm quản lý du lịch.

Xây dựng website du lịch.

b. Phạm vi

Xây dựng các tour trong nước, tour ngoài nước.

Chat bot hỗ trợ tư vấn khách hàng.

Nghiên cứu sử dụng công nghệ Reactjs và Nodejs vào website.

4. Công nghệ, công cụ và ngôn ngữ lập trình:

- Frontend: Reactjs + Redux. HTML/CSS-scss.
- Backend: Node.js(Express) + MySql(Sequelize)

5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng

- Báo cáo tốt nghiệp
- Xây dựng hoàn chỉnh trang web quản lý du lịch

6. Giáo viên và cán bộ hướng dẫn

Họ tên: Phạm Thị Miên

Đơn vị công tác: Trường Đại học GTVT Phân hiệu tại thành phố Hồ Chí Minh

Điện thoại: 0961170638

Email: ptmien@utc2.edu.vn

Ngày tháng 10 năm 2021
Trưởng BM Công nghệ Thông tin

Đã giao nhiệm vụ TKTN
Giáo viên hướng dẫn

ThS. Trần Phong Nhã

Đã nhận nhiệm vụ TKTN

Sinh viên: Biện Thanh Phong

Điện thoại: 0339993446

Ký tên:

Email: bienthanhphong212@gmail.com

LỜI CẢM ƠN

Để hoàn thành đề tài tốt nghiệp này trước hết em xin gửi đến quý thầy, cô Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông Vận tải tại Thành phố Hồ Chí Minh lời cảm ơn chân thành vì đã truyền đạt cho em những kiến thức không chỉ từ sách vở, mà còn những kinh nghiệm quý giá từ cuộc sống trong khoảng thời gian học tập tại trường. Đặc biệt em xin gửi đến cô Trần Thị Miên lời cảm ơn sâu sắc nhất, cô đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình thực hiện đề tài. Tuy đề tài không được lớn lao nhưng nếu không được sự hướng dẫn chỉ bảo tận tình của thầy và cô thì đề tài tốt nghiệp này khó có thể hoàn thành được. Vì thời gian làm đồ án có hạn cũng như hiểu biết cá nhân còn hạn chế, em cũng đã nỗ lực hết sức để hoàn thành bài báo cáo đồ án một cách tốt nhất, nhưng chắc chắn vẫn sẽ có những thiếu sót không thể tránh khỏi. Em kính mong nhận được sự thông cảm và những ý kiến đóng góp chân thành từ quý thầy cô.

Sau cùng, em xin kính chúc Quý Thầy Cô trong Bộ môn Công nghệ thông tin đang công tác tại Bộ phận một cửa – Phòng Tổ chức hành chính luôn mạnh khỏe, hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm
Giáo viên hướng dẫn

Phạm Thị Miên

MỤC LỤC

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

LỜI CẢM ƠN

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

CHƯƠNG I. MỞ ĐẦU..... 1

- 1. Lý do chọn đề tài 1
- 2. Mục tiêu 1
- 2.1 Cấu trúc báo cáo thực tập tốt nghiệp 2

CHƯƠNG II. TÌM HIỂU REACTJS, REDUX, NODE.JS(EXPRESS), MYSQL(SEQUELIZE)..... 3

- 1. Tổng quan về Reactjs..... 3
 - 1.1 Reactjs là gì? 3
 - 1.2 Thành phần cơ bản của Reactjs..... 3
 - 1.3 Lợi ích khi sử dụng Reactjs..... 3
- 2. Tổng quan về Redux 4
 - 2.1 Redux là gì?..... 4
 - 2.2 Những lợi ích của Redux 4
 - 2.3 Cấu trúc của Redux 5
 - 2.3.1 Actions 5
 - 2.3.2 Reducers 6
 - 2.3.3 Store 6
- 3. Tổng quan về NodeJS 6
 - 3.1 Khái niệm NodeJS..... 6
 - 3.2 Đặc điểm của NodeJS 7
 - 3.3 Express là gì 7

CHƯƠNG III. BÀI TOÁN QUẢN LÝ DU LỊCH 20

- 1. Mô tả bài toán 20
- 2. Đặc tả yêu cầu bài toán 20
- 3. Thực trạng 21
- 4. Giải quyết..... 21

CHƯƠNG IV. PHÂN TÍCH THẾT KẾ HỆ THỐNG 22

- 1. Biểu đồ Use case..... 22
 - 1.1 Xác định tác nhân..... 22
- 2. Xác định các ca sử dụng 22
- 3. Biểu đồ use case chính..... 23

4. Đặc tả use case	23
4.1 Đặc tả UC “ Đăng ký tour du lịch”	23
4.2 Đặt tả use case “Xem và quản lý thông tin du lịch”	24
4.3 Đặc tả use case “tìm kiếm”	25
4.4 Đặc tả use case quản lý thông tin nhân viên	25
4.5 Đặc tả use case đăng nhập.....	26
4.6 Đặc tả use case quản lý thông tin khách hàng.....	27
4.7 Đặc tả use case tư vấn khách hàng.....	28
4.8 Đặc tả use case tạo phiếu đăng ký tour	29
5. Biểu đồ tuần tự.....	30
5.1 Biểu đồ tuần tự quản lý nhân viên	30
5.2 Biểu đồ tuần tự tạo phiếu đăng ký tour	31
5.3 Biểu đồ tuần tự quản lý thông tin khách hàng	32
5.4 Biểu đồ tuần tự tư vấn khách hàng.....	33
5.5 Biểu đồ tuần tự đăng ký tour.....	34
5.6 Biểu đồ tuần tự xem thông tin khách hàng	35
5.7 Biểu đồ tuần tự khách hàng tìm kiếm	36
5.8 Biểu đồ tuần tự quản lý tour du lịch.....	37
6. Biểu đồ hoạt động.....	38
6.1 Biểu đồ hoạt động khách hàng đăng ký tour du lịch.....	38
6.2 Biểu đồ hoạt động khách hàng xem tour du lịch	39
6.3 Biểu đồ hoạt động tạo phiếu đăng ký tour	40
6.4 Biểu đồ hoạt động quản lý thông tin khách hàng.....	41
6.5 Biểu đồ hoạt động tư vấn khách hàng.....	41
6.6 Biểu đồ hoạt động quản lý nhân viên.....	42
6.7 Biểu đồ hoạt động quản lý tour du lịch.....	43
7. Biểu đồ lớp.....	44
7.1 Tìm lớp.....	44
7.2 Biểu đồ lớp chi tiết.....	46
CHƯƠNG V. THIẾT KẾ GIAO DIỆN	47
1. Một số giao diện hệ thống.....	47
1.1 Giao diện tìm kiếm.....	47
1.2 Giao diện trang Chủ	47
1.3 Giao diện điền thông tin đặt tour	48
1.4 Giao diện tour du lịch.....	49
CHƯƠNG VI. KẾT QUẢ VÀ KIẾN NGHỊ	50
TÀI LIỆU THAM KHẢO	51

SƠ ĐỒ, HÌNH VẼ

Hình 1. Use case chính	23
Hình 2. Use case đăng ký tour du lịch	24
Hình 3. Use case quản lý thông tin tour du lịch	25
Hình 4. Use case tìm kiếm.....	25
Hình 5. Use case quản lý thông tin nhân viên	26
Hình 6. Use case đăng nhập	27
Hình 7. Use case quản lý thông tin khách hàng	28
Hình 8. Use case tư vấn khách hàng.....	29
Hình 9. Use case tạo phiếu đăng ký tour	29
Hình 10. BDTT quản lý nhân viên	30
Hình 11. BDTT tạo phiếu đăng ký tour.....	31
Hình 12. BDTT quản lý thông tin khách hàng	32
Hình 13. BDTT tư vấn khách hàng	33
Hình 14. BDTT đăng ký tour	34
Hình 15. BDTT xem thông tin khách hàng	35
Hình 16. BDTT khách hàng tìm kiếm	36
Hình 17. BDTT quản lý tour du lịch	37
Hình 18. BDHD đăng ký tour du lịch.....	38
Hình 19. BDHD xem tour du lịch	39
Hình 20. BDHD tạo phiếu đăng ký tour.....	40
Hình 21. BDHD quản lý thông tin khách hàng	41
Hình 22. BDHD tư vấn khách hàng	41
Hình 23. BDHD quản lý nhân viên	42
Hình 24. BDHD quản lý tour du lịch	43
Hình 25. Biểu đồ lớp	46
Hình 26. Giao diện tìm kiếm	47
Hình 27. Giao diện đặt tour	48
Hình 28. Giao diện tour du lịch	49

CHƯƠNG I. MỞ ĐẦU

1. Lý do chọn đề tài

Lý do chọn đề tài Xã hội phát triển thì nhu cầu đi tham quan du lịch của mỗi người là rất lớn. Hiện nay du lịch được đánh giá là ngành công nghiệp vàng, cung cấp nhiều lợi nhuận cho quốc gia. Ngành du lịch phát triển mạnh mẽ trở thành một lĩnh vực hoạt động kinh tế hàng đầu. Bên cạnh đó, du lịch là một cách tiếp cận thực tế nhất tạo nên ấn tượng khó quên đối với con người khi muốn tìm hiểu một địa danh, một vùng đất, hay một dân tộc,...

Một trong những yếu tố giúp du khách tiếp cận và giới thiệu nhanh nhất các giá trị đó là hình thức thông tin quảng bá. Đó là những thông tin được giới thiệu trên hệ thống truyền thông đại chúng, hệ thống internet và các hình thức khác. Đối với du khách các thông tin về giá trị điểm đến, điều kiện đi lại, ăn ở, điều kiện về an ninh an toàn nơi mình sẽ đến du lịch luôn là những vấn đề cần được quan tâm. Chính vì thế, quảng bá du lịch có vai trò rất quan trọng để du khách biết về một điểm đến hấp dẫn, lý thú; hay một đất nước tươi đẹp giàu truyền thống văn hóa - lịch sử, với những danh lam thắng cảnh nổi tiếng, với hình ảnh về đất nước, con người, môi trường...từ đó làm tăng mối quan hệ hữu nghị giữa các dân tộc, các quốc gia.

Hơn nữa, mạng Internet hiện nay đang phát triển một cách mạnh mẽ. Ngày càng có nhiều doanh nghiệp Việt Nam đã tận dụng khai thác mạng Internet nhằm nâng cao hiệu quả kinh doanh và đạt được những thành công không nhỏ. Thương mại điện tử đã và đang là thách thức cho các doanh nghiệp trong thời kỳ cạnh tranh và hội nhập khắc nghiệt như hiện nay. Và du lịch cũng không phải là ngoại lệ. Nắm bắt được vấn đề đó, em đã tiến hành “Xây dựng website du lịch”, giúp quý khách hàng có thể hiểu biết thêm về các danh thắng cảnh của Việt Nam, và thế giới để lựa chọn được các địa điểm giải trí, thư giãn và đặt tour qua mạng internet.

2. Mục tiêu

Tìm hiểu công nghệ reactjs và nodejs với các thư viện liên quan.

Xây dựng website du lịch sử dụng công nghệ reactjs và nodejs.

Hỗ trợ tìm kiếm tour du lịch trong nước ở các tỉnh, và các chuyến tham quan ngoài nước.

Trang quản lý chuyến tham quan, và đặt tour du lịch.

2.1 Cấu trúc báo cáo thực tập tốt nghiệp

2.1.1 Chương I: Mở đầu

2.1.2 Chương II: Tìm hiểu reactjs, redux, node.js(express), mysql(sequelize)

2.1.3 Chương III: Bài toán quản lý du lịch

2.1.4 Chương IV: Phân tích thiết kế hệ thống

2.1.5 Chương V: Thiết kế giao diện

2.1.6 Chương VI: Kết luận và kiến nghị

CHƯƠNG II. TÌM HIỂU REACTJS, REDUX, NODE.JS(EXPRESS), MYSQL(SEQUELIZE)

1. Tổng quan về Reactjs

1.1 Reactjs là gì?

Reactjs là một thư viện Javascript mã nguồn mở hỗ trợ xây dựng các thành phần giao diện nhanh gọn và tiện lợi. Bình thường các lập trình viên sẽ nhúng javascript vào code HTML thông qua các attribute như AngularJS nhưng với Reactjs làm việc như một thư viện cho phép nhúng HTML vào javascript thông qua JSX. Qua đó bạn có thể dễ dàng lồng các đoạn HTML vào trong JSX làm cho các component dễ hiểu và dễ sử dụng hơn. Trong Reactjs thường dùng javascript để thiết kế bố cục cho trang web, nhưng nhược điểm là cấu trúc khá là khó. Thay vào đó sử dụng JSX và nhúng các đoạn HTML vào javascript, ta thấy cú pháp dễ hiểu hơn và JSX cũng có thể tối ưu code khi biên soạn. Vừa dễ cho người lập trình mà vừa tiện cho việc biên dịch.

1.2 Thành phần cơ bản của Reactjs

Thành phần cơ bản của React được gọi là components. Syntax để viết HTML sử dụng Javascript để render. Bạn có thể tạo ra một component bằng các gọi phương thức `createClass` của đối tượng React, điểm bắt đầu khi tiếp cận với thư viện này. Có thể lồng nhiều component vào nhau thông qua lệnh `return` của phương thức `render`.

Trong một chương trình thì có rất nhiều các component, để đơn giản việc quản lý các component đó người ta sử dụng `redux`, `redux` giống như 1 cái kho chứa các component và khi dùng component nào thì chỉ cần gọi nó ra.

Virtual DOM không được tạo ra bởi Reactjs nhưng lại được sử dụng rất nhiều. Đây là một chuẩn của W3C được dùng để truy xuất code HTML hoặc XML. Các Virtual DOM sẽ được tạo ra khi chạy chương trình, đó là nơi chứa các component. Sử dụng DOM sẽ tiết kiệm được hiệu suất làm việc, khi có thay đổi gì Reactjs đều tính toán trước và việc còn lại chỉ là thực hiện chúng lên DOM.

1.3 Lợi ích khi sử dụng Reactjs

- ReactJS giúp cho việc viết các đoạn code Javascript sẽ trở nên dễ dàng hơn vì nó sử dụng một cú pháp đặc biệt đó chính là cú pháp JSX. Thông qua JSX cho phép nhúng code HTML và Javascript.
- ReactJS cho phép Developer phá vỡ những cấu tạo UI phức tạp thành những component độc lập. Dev sẽ không phải lo lắng về tổng thể ứng dụng web, giờ đây

Developer dễ dàng chia nhỏ các cấu trúc UI/UX phức tạp thành từng component đơn giản hơn.

- Đi kèm với ReactJS là rất nhiều các công cụ phát triển giúp cho việc debug code một cách dễ dàng hơn. Một trong những ưu điểm nữa của ReactJS đó là sự thân thiện với SEO. Hầu như các JS Frameworks không thân thiện với các tìm kiếm mặc dù đã được cải thiện nhiều nhưng dưới sự hỗ trợ của các render dữ liệu trả về dưới dạng web page giúp cho SEO chuẩn hơn.

2. Tổng quan về Redux

2.1 Redux là gì?

Redux là một predictable state management tool cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. Redux ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ Elm và kiến trúc Flux của Facebook. Do vậy Redux thường dùng kết hợp với React.

2.2 Những lợi ích của Redux

- Hỗ trợ dự đoán trạng thái

Redux có chức năng dự đoán và quản lý trạng thái và chúng sẽ không bao giờ thay đổi. Lợi ích này có thể giúp người dùng thực hiện các nhiệm vụ phức tạp như hoàn tác hoặc làm lại vô thời hạn. Đồng thời, Redux còn có chức năng luân chuyển linh hoạt giữa các trạng thái để kiểm tra hiệu quả trong thời gian thực.

- Khả năng bảo trì

Redux có một hệ thống code cực kỳ nghiêm ngặt, nhưng với những người đã sử dụng và hiểu về Redux sẽ tiếp cận dễ dàng hơn. Cũng chính việc này đã giúp cho Redux có thể được bảo trì một cách dễ dàng.

Bên cạnh đó, lợi ích này còn góp phần giúp người dùng tách biệt logic nghiệp vụ khỏi sơ đồ thành phần. Trong khi đó, mục tiêu quan trọng của các ứng dụng có quy mô lớn hiện nay đều là hướng cho phần mềm của mình có thể dự đoán cũng như bảo trì được.

- Gỡ lỗi một cách dễ dàng

Redux cho phép người dùng gỡ lỗi dễ dàng bằng cách lưu lại những Action và trạng thái để dễ nhận diện đối với những trường hợp lỗi mã hóa, lỗi mạng và một số lỗi khác khi định dạng trong quá trình triển khai chương trình. Việc gỡ lỗi thông thường sẽ cần nhiều thời gian và phức tạp nhưng với Redux DevTools của Redux sẽ hỗ trợ người dùng thực hiện thao tác gỡ lỗi dễ dàng hơn.

- Lợi ích về hiệu suất

React Redux thực hiện nhiều tối ưu hóa hiệu suất bên trong để thành phần được kết nối của riêng bạn chỉ hiển thị khi thực sự cần.

- Dễ kiểm tra

Tương đối dễ dàng để kiểm tra các ứng dụng Redux vì các chức năng được sử dụng để thay đổi trạng thái của các chức năng thuần túy.

- Trạng thái bền bỉ

Bạn có thể duy trì một số trạng thái của ứng dụng trong bộ nhớ cục bộ và khôi phục chúng sau khi làm mới. Điều này thực sự tiện lợi.

- Kết xuất phía máy chủ

Trên máy chủ có thể để Redux được hiển thị, người dùng có thể xử lý các kết xuất ban đầu của chương trình bằng cách truyền tải những trạng thái đến các máy chủ và đợi phản hồi từ nó.

2.3 Cấu trúc của Redux

Redux có cấu trúc gồm 3 phần: Actions, Reducers và Store.

2.3.1 Actions

Actions có thể hiểu là các sự kiện trong chương trình. Đây là cách duy nhất để người dùng truyền tải data từ ứng dụng đến Store trong Redux, dữ liệu được truyền tải có thể là các yêu cầu từ người dùng, ứng dụng, API call hoặc form submission.

Các sự kiện sẽ được gửi với phương thức `store.dispatch()`. Cần phải thể hiện rõ loại sự kiện thông qua loại giá trị thì yêu cầu mới có thể được thực hiện. Nơi chứa những thông tin đó được gọi là “payload” và sự kiện được tạo thông qua Action Creator. Khi các phép tính thực hiện các Actions sẽ hoàn trả một state mới hoàn toàn, vì thế tính dự đoán của trạng thái này rất cao và dễ dùng.

2.3.2 Reducers

Nếu Actions có nhiệm vụ mô tả những gì xảy ra nhưng lại không chỉ rõ phần trạng thái nào của response đã thay đổi và thay đổi như thế nào, khi đó Reducer sẽ đảm nhiệm việc này. Reducers được xem là các hàm thuần túy lấy trạng thái hiện tại của ứng dụng sau đó thực hiện một Action và trả về trạng thái mới. Các trạng thái này sẽ được lưu trữ dưới dạng đối tượng và đồng thời chúng xác định trạng thái của ứng dụng thay đổi như thế nào để đáp ứng với Action được gửi đến Store.

Reducers dựa trên hàm “reduce” của JavaScript, trong đó một giá trị sẽ được tính từ nhiều giá trị sau khi thực hiện chức năng gọi lại.

2.3.3 Store

Store là chương trình duy nhất tại Redux hỗ trợ lưu trữ trạng thái ứng dụng hay có thể nói chỉ có một Store trong bất kỳ ứng dụng Redux nào. Store cho phép người dùng có thể tiếp tục truy cập vào những chương trình đã được lưu và có thể can thiệp vào chúng thông qua những phương pháp hỗ trợ gồm cập nhật, đăng ký hoặc hủy.

Trong Redux, Store đóng vai trò quan trọng nhất bởi chức năng và nhiệm vụ của mình, để kích hoạt các Action được thực hiện phải sử dụng đến các phân tử dispatcher sau đó gửi đến reducer.

2.4 Nguyên lý vận hành cơ bản của React Redux

Trong quá trình xây dựng Redux, các chuyên gia đã dựa vào 3 nguyên lý cơ bản sau đây:

- Sử dụng nguồn dữ liệu đảm bảo tin cậy. Các state của tất cả ứng dụng đều nằm cùng một object tree trong một Store duy nhất.
- Chỉ được phép đọc trạng thái. Hiểu đơn giản, để thay đổi trạng thái của ứng dụng, cách duy nhất là thực hiện một Action.
- Sử dụng các hàm thuần túy để thay đổi với mục đích chỉ ra cách state được biến đổi từ Action. Vì thế, bạn mới cần sử dụng các pure function được gọi là Reducer.

3. Tổng quan về NodeJS

3.1 Khái niệm NodeJS

- NodeJS là một nền tảng (platform) phía Server side được xây dựng, vận hành trên V8 JavaScript runtime của Chrome giúp xây dựng và phát triển các ứng dụng mạng một cách nhanh chóng và có khả năng mở rộng dễ dàng.

- Node.js ra mắt vào năm 2009, bởi Ryan Dahl và nhận sự bảo trợ từ công ty Joyent (Hoa Kỳ).
- Phần core bên dưới của Node.js hầu hết được viết bằng C++. Chính vì thế mà tốc độ xử lý và hiệu năng khá cao.
- Bên cạnh đó, Node.js còn tạo ra ứng dụng có tốc độ xử lý nhanh với thời gian thực (real-time).
- Node.js cũng được áp dụng để tạo các sản phẩm có lượng truy cập lớn, mở rộng nhanh và đổi mới công nghệ... hoặc tạo dự án startup tiết kiệm thời gian nhất.

3.2 Đặc điểm của NodeJS

- Nodejs có nhiều đặc điểm nổi bật, vượt trội. Nắm được các đặc điểm này sẽ giúp bạn hiểu sâu hơn về Nodejs, đồng thời đưa ra quyết định sử dụng nền tảng này chính xác.
- Bất đồng bộ và phát sinh sự kiện (Non-blocking and Event Driven): Tất cả các APIs của thư viện Node.js đều bất đồng bộ (non-blocking), NodeJS không cần đợi một API trả về dữ liệu. Server chuyển sang một API khác sau khi gọi nó và có cơ chế riêng để gửi thông báo và nhận phản hồi về các hoạt động của Node.js và API đã gọi.
- Tốc độ nhanh: Phần core phía dưới được viết gần như toàn bộ bằng C++ kết hợp Chrome V8 Engine nên tốc độ xử lý công việc của Node.js cực nhanh, nhưng vẫn đảm bảo được tính chuẩn xác.
- Đơn giản – Hiệu năng cao: Node.js sử dụng một mô hình luồng đơn luồng (single thread) và các sự kiện lặp (event-loop). Cơ chế sự kiện cho phép phía Server trả về phản hồi theo non-blocking, đồng thời tăng hiệu quả sử dụng. Các luồng đơn cung cấp dịch vụ cho nhiều request hơn hẳn Server truyền thống.
- Không lưu bộ nhớ đệm (non buffer): Nền tảng Node.js không có vùng nhớ đệm, tức không cung cấp khả năng lưu trữ dữ liệu buffer.

3.3 Express là gì

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

Một số chức năng chính của Expressjs như sau:

Thiết lập các lớp trung gian để trả về các HTTP request.

- Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số.

4.MySQL(Sequelize):

4.1 Sequelize là gì ?

Sequelize là một ORM dành cho Node.js và io.js. Nó hỗ trợ bạn truy cập một cách dễ dàng đến PostgreSQL, MySQL, MariaDB, SQLite và MSSQL cùng với các tính năng như là relations, transaction, replication ...

4.2 Tính năng cơ bản

4.2.1 Model

Để tạo 1 ánh xạ giữa 1 model và 1 table ta sử dụng phương thức define. Sequelize sẽ tự động tạo các attributes là createdAt và updatedAt (nếu bạn không muốn sử dụng hoặc chỉ sử dụng 1 trong 2 attributes thì có thể tham khảo ở phần config sau đây).

```
var Project = sequelize.define('Project', {  
  title: Sequelize.STRING,  
  description: Sequelize.TEXT  
})
```

Ngoài ra có thể thêm các option cho các attributes:

```
var Book = sequelize.define('Book', {  
  title: { type: Sequelize.STRING, allowNull: false, defaultValue: true },  
  author: { type: Sequelize.STRING, allowNull: false },  
  ISBN: { type: Sequelize.STRING, unique: true },  
});
```

Một số kiểu dữ liệu phổ biến của Sequelize:

Sequelize.STRING	// VARCHAR(255)
Sequelize.STRING(1234)	// VARCHAR(1234)
Sequelize.STRING.BINARY	// VARCHAR BINARY
Sequelize.TEXT	// TEXT
Sequelize.INTEGER	// INTEGER
Sequelize.BIGINT	// BIGINT
Sequelize.BIGINT(11)	// BIGINT(11)
Sequelize.FLOAT	// FLOAT
Sequelize.FLOAT(11)	// FLOAT(11)
Sequelize.FLOAT(11, 12)	// FLOAT(11,12)
Sequelize.REAL	// REAL Chỉ hỗ trợ PostgreSQL.
Sequelize.REAL(11)	// REAL(11) Chỉ hỗ trợ PostgreSQL.
Sequelize.REAL(11, 12)	// REAL(11,12) Chỉ hỗ trợ PostgreSQL.
Sequelize.DOUBLE	// DOUBLE
Sequelize.DOUBLE(11)	// DOUBLE(11)
Sequelize.DOUBLE(11, 12)	// DOUBLE(11,12)
Sequelize.DECIMAL	// DECIMAL
Sequelize.DECIMAL(10, 2)	// DECIMAL(10,2)

Config

```
var Bar = sequelize.define('Bar', { /* bla */ }, {  
  // Không tự động thêm timestamp attributes (updatedAt, createdAt)  
  timestamps: false,  
  
  // softDelete, Sequelize sẽ tự động thêm attribute deletedAt, chỉ hoạt động khi bạn  
  // enable timestamps  
  paranoid: true,  
  
  // Sử dụng underscore style thay cho camel style  
  // updatedAt sẽ là updated_at...  
  underscored: true,  
  
  // chỉ định tên table  
  tableName: 'my_very_custom_table_name'  
})
```

Sử dụng model:

```
// Tạo model  
  
var Task = sequelize.define('Task', {  
  title: Sequelize.STRING,  
  rating: { type: Sequelize.INTEGER, defaultValue: 3 }  
})
```

```

// Tìm kiếm

// Tìm kiếm theo Id

var task = Task.findById(123);


// Tìm kiếm theo attributes

var task = Task.findOne({ where: {title: 'aTask'} }).then(function(task) {

})


var task = Task.findOne({

    where: {title: 'aProject'},

    attributes: ['id', ['title', 'rating']]

}).then(function(task) {

})


// Tìm kiếm nhiều bản ghi

var task = Task.findAll().then(function(tasks) {

})


// Giới hạn số lượng kết quả của query

var task = Task.findAll({ limit: 10 })

// Bỏ qua 10 kết quả đầu tiên

var task = Task.findAll({ offset: 10 })


//Đếm số lượng result

var task = Task.count().then(function(c) {

    console.log("There are " + c + " tasks!")

})

```

```

// Thêm mới

    // 1. Tạo một thể hiện của model (chưa được lưu vào database)

    var task = Task.build({

        title: 'specify the project idea',

        rating: 2

    })

    task.title // ==> 'specify the project idea'

    task.rating // ==> 2

    // Để lưu vào database chúng ta sử dụng function save

    task.save()

// 2. Tạo bản ghi trong database

    Task.create({ title: 'foo', rating: 5});

// Sửa

    // Có 2 cách để cập nhật dữ liệu

    // Cách 1:

    task.title = 'a very different title now'

    task.save().then(function() {}))

    // Cách 2

    task.update({

        title: 'a very different title now'

    }).then(function() {}))

// Xóa

    task.destroy({ force: true })

```

4.2.2 Query

- Để select một vài attribute, bạn có thể sử dụng attributes option

```
Model.findAll({
  attributes: ['foo', 'bar']
});

// => SELECT foo, bar from ...
```

- Mệnh đề where

```
Post.findAll({
  where: {
    authorId: 2
  }
});

// SELECT * FROM post WHERE authorId = 2

Post.findAll({
  where: {
    authorId: 12,
    status: 'active'
  }
});

// SELECT * FROM post WHERE authorId = 12 AND status = 'active';

Post.destroy({
  where: {
    status: 'inactive'
  }
});
```

```
// DELETE FROM post WHERE status = 'inactive';

Post.update({
  updatedAt: null,
}, {
  where: {
    deletedAt: {
      $ne: null
    }
  }
});
```

- Các toán tử bạn có thể sử dụng cùng với where

```
$and: {a: 5}      // AND (a = 5)

$or: [{a: 5}, {a: 6}] // (a = 5 OR a = 6)

$gt: 6,          // > 6

$gte: 6,         // >= 6

$lt: 10,         // < 10

$lte: 10,        // <= 10

$ne: 20,         // != 20

$between: [6, 10], // BETWEEN 6 AND 10

$notBetween: [11, 15], // NOT BETWEEN 11 AND 15

$in: [1, 2],     // IN [1, 2]

$notIn: [1, 2],  // NOT IN [1, 2]
```

```

$like: '%hat',      // LIKE '%hat'

$notLike: '%hat'    // NOT LIKE '%hat'

$iLike: '%hat'      // ILIKE '%hat' (Phân biệt hoa thường) (chỉ hỗ trợ PG)

$notILike: '%hat'   // NOT ILIKE '%hat' (chỉ hỗ trợ PG)

$like: { $any: ['cat', 'hat']}

                // LIKE ANY ARRAY['cat', 'hat']

$overlap: [1, 2]    // && [1, 2] (PG array overlap )

$contains: [1, 2]   // @> [1, 2] (PG array contains )

$contained: [1, 2]  // <@ [1, 2] (PG array contained )

$any: [2,3]         // ANY ARRAY[2, 3]::INTEGER (chỉ hỗ trợ PG)

```

- Raw Query: Đôi khi bạn muốn excute một raw query, khi đó bạn có thể sử dụng `sequelize.query`

```
sequelize.query("SELECT * FROM `users`");
```

4.2.3 Relations

Các loại relation gồm: One-To-One associations, One-To-Many associations, Belongs-To-Many associations.

- One-To-One associations

- BelongsTo:

```

var Player = this.sequelize.define('Player', { /* attributes */})

, Team = this.sequelize.define('Team', { /* attributes */});

Player.belongsTo(Team);

```

- HasOne:


```
var User = sequelize.define('User', { /* ... */ })

var Project = sequelize.define('Project', { /* ... */ })

// One-way associations

Project.hasOne(User)
```

- One-To-Many associations

```
var User = sequelize.define('User', { /* ... */ })

var Project = sequelize.define('Project', { /* ... */ })

Project.hasMany(User, { as: 'Workers' })
```

- Belongs-To-Many associations

```
Project.belongsToMany(User, { through: 'UserProject' });

User.belongsToMany(Project, { through: 'UserProject' });
```

4.2.4 Migration

Để sử dụng tính năng migration bạn cần cài thêm package sequelize-cli

```
$ npm install --save sequelize-cli
```

Các command của sequelize-cli

```
$ sequelize db:migrate // Chạy migration.

$ sequelize db:migrate:undo // Rollback lần cuối chạy migration.

$ sequelize help // Hiện thị help.

$ sequelize init // Khởi tạo project.
```

```
$ sequelize migration:create // Tạo 1 migration mới.
```

```
$ sequelize version
```

Bằng cách sử dụng queryInterface bạn có thể sử dụng được hầu hết các function

```
//Tạo mới 1 table

queryInterface.createTable(
  'nameOfTheNewTable',
  {
    id: {
      type: Sequelize.INTEGER,
      primaryKey: true,
      autoIncrement: true
    },
    createdAt: {
      type: Sequelize.DATE
    },
    updatedAt: {
      type: Sequelize.DATE
    },
    attr1: Sequelize.STRING,
    attr2: Sequelize.INTEGER,
    attr3: {
      type: Sequelize.BOOLEAN,
      defaultValue: false,
      allowNull: false
    }
  }
);
```

```

    }

    },

    {
      engine: 'MYISAM', // default: 'InnoDB'
      charset: 'latin1' // default: null
    }
  )

  //Xóa 1 Table
  queryInterface.dropTable('nameOfTheExistingTable')

  //Xóa tất cả các Table
  queryInterface.dropAllTables()

  //Đổi tên table
  queryInterface.renameTable('Person', 'User')

  //Hiện thị tất cả các table
  showAllTables(options)

  //Thêm cột mới
  queryInterface.addColumn(
    'nameOfAnExistingTable',
    'nameOfTheNewAttribute',
    Sequelize.STRING
  )

  // or
  queryInterface.addColumn(
    'nameOfAnExistingTable',
    'nameOfTheNewAttribute',

```

```
{
  type: Sequelize.STRING,
  allowNull: false
}
)

//Xóa 1 cột
queryInterface.removeColumn('Person', 'signature')

//Đổi tên cột
queryInterface.renameColumn('Person', 'signature', 'sig')

//Thêm index
queryInterface.addIndex(
  'Person',
  ['firstname', 'lastname'],
  {
    indexName: 'SuperDuperIndex',
    indicesType: 'UNIQUE'
  }
)

//Xóa index
queryInterface.removeIndex('Person', 'SuperDuperIndex')
```

CHƯƠNG III. BÀI TOÁN QUẢN LÝ DU LỊCH

1. Mô tả bài toán

Quản lý tour du lịch là công việc rất quan trọng và phức tạp mà bất kỳ công ty du lịch nào cũng cần phải thực hiện. Công ty du lịch quản lý tour và lên lịch trình cho từng tour du lịch cụ thể sau đó cập nhật thông tin các tour với đầy đủ thông tin về giá tour, lịch trình, các địa điểm đi đến và hình ảnh tour nếu có. Theo dõi lịch trình các tour du lịch, quản lý nhân viên hướng dẫn tour du lịch. Một hệ thống như vậy có ở hầu hết các công ty du lịch trong nước. Không chỉ dừng lại như vậy, một hệ thống quản lý tour du lịch còn cần phải quản lý thêm nhiều chức năng khác nữa. Chẳng hạn như việc bảo mật thông tin khách hàng, hệ thống đánh giá nhận xét góp ý của khách hàng về chất lượng dịch vụ, tổ chức tour theo đăng ký lựa chọn tour của khách hàng, quản lý một số vấn đề liên quan đến như khách hàng, nhân viên, phương tiện,.. được tốt hơn, đảm bảo chính xác nhanh chóng và tiện lợi.

2. Đặc tả yêu cầu bài toán

Công ty du lịch thiết kế tour và lên lịch trình cho từng tour cụ thể sau đó cập nhật thông tin các tour này với đầy đủ thông tin về giá cả loại tour, lịch trình, các địa phương đi đến và hình ảnh minh họa nếu có).

Người dùng tìm kiếm thông tin về các tour du lịch hiện hành hoặc tìm kiếm tour theo các yêu cầu cụ thể về giá cả, địa phương muốn đến, ngày khởi hành của tour... sau đó khách hàng có thể tiến hành đặt chỗ cho tôi đang xem nếu muốn.

Khách hàng có thể đặt tour trực tuyến, hệ thống sẽ cập nhật thông tin đặt chỗ này và lưu trực tiếp lên cơ sở dữ liệu, cho phép người quản lý được phân quyền theo các công việc sau :

- Tìm kiếm các đơn đặt chỗ đã được xác nhận hay chưa xác nhận.
 - Xác nhận đơn đặt chỗ hợp lệ chỉnh sửa thông tin khách hàng và thông tin đặt chỗ trên từng đơn cụ thể thống kê tình hình đặt chỗ của tour.
 - Xem danh sách khách hàng tham gia các tour đó
 - Thay đổi, thêm mới hoặc xóa bỏ thông tin, hình ảnh về các địa điểm du lịch ở các địa phương khác nhau mà công ty muốn giới thiệu cho khách hàng
- Về tổ chức lưu trữ thực hiện các yêu cầu:

- Thêm xóa sửa thông tin hình ảnh về các tour du lịch do công ty tổ chức, phục vụ cho công tác quản lý, thống kê tình hình hoạt động của công ty.
- Thêm xóa sửa thông tin hình ảnh về các điểm du lịch ở từng địa phương khác nhau.

3. Thực trạng

Trong những năm gần đây, khi đất nước ta đang trên đà phát triển kéo theo sự phát triển cao về mọi mặt của nền kinh tế xã hội, ngành du lịch nước ta đang đứng trước cơ hội và thách thức rất lớn. Tuy nhiên, hiện nay việc ứng dụng công nghệ thông tin tại các công ty du lịch chưa được chú trọng, tất cả các nghiệp vụ quản lý đều trên sổ sách, giấy tờ. Điều này dẫn đến các công ty tốn kém nhiều nhân lực cũng như kinh phí, chưa đem lại hiệu quả kinh doanh cao. Khi có khách yêu cầu đặt Tour, hay đại lý báo khách, nhân viên phải tìm và tính toán xem ngày nào đó có những Tour nào khởi hành, hay Tour nào đó có khởi hành trong ngày nào đó không... Điều này làm mất nhiều thời gian của khách, của đại lý báo khách, vì khách đặt Tour hay đại lý báo khách chủ yếu qua điện thoại. Hiện tại, các thông tin đặt Tour hay điều Tour đều chỉ lưu trên sổ sách, giấy tờ, nhân viên xử lý dữ liệu chi sao lại thông tin vào máy và xử lý thống kê, báo cáo. Các thông tin lưu dưới dạng này rất khó quản lý, nhất là khi nhân viên muốn tìm kiếm thông tin thì mất nhiều thời gian. Đồng thời, nhân viên phải tự thống kê tạo báo cáo thủ công, điều này rất mất nhiều thời gian và có thể không chính xác. Khi lưu trữ thông tin như trên, công ty cần phải có một số lượng lớn nhân viên làm việc, điều này tốn kém về nhân lực cũng như kinh phí. Để nâng cao hiệu quả kinh doanh du lịch, cũng như chất lượng phục vụ du khách, công ty cần phải cải tiến quy trình quản lý của mình, cần tin học hóa quá trình quản lý nhằm giảm bớt nhân lực, đem lại hiệu quả quản lý cao.

4. Giải quyết

Để giải quyết các yêu cầu trên, tôi đã xây dựng và triển khai hệ thống quản lý tour du lịch. Mục đích giúp cho các công ty có thể quản lý các hoạt động nghiệp vụ của mình tốt hơn, nhân viên có thể quản lý được các thông tin lịch trình các tour, thông tin đặt tour của khách, thông tin khởi hành và quản lý khách hàng.

CHƯƠNG IV. PHÂN TÍCH THÉT KẾ HỆ THỐNG

1. Biểu đồ Use case

1.1 Xác định tác nhân :

- Tác nhân : nhân viên
- Tác nhân : khách hàng
- Tác nhân : người quản lý

2. Xác định các ca sử dụng

Dựa trên việc mô tả bài toán và việc phân tích để tìm ra các tác nhân ta xác định được các ca sử dụng như sau :

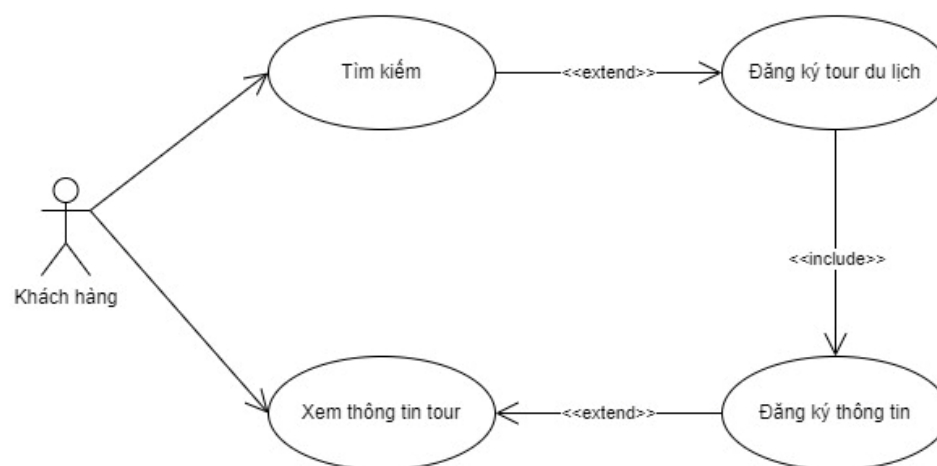
- Đăng ký tour du lịch
- Xem thông tin tour du lịch
- Đăng nhập
- Quản lý thông tin khách hàng
- Tư vấn khách hàng
- Tìm kiếm
- Cập nhật thông tin tour du lịch
- Quản lý thông tin nhân viên
- Tạo phiếu đăng ký tour

Tác nhân	Ca sử dụng
Khách hàng	Xem thông tin tour Tìm kiếm tour Đăng ký tour
Nhân viên	Đăng nhập Quản lý thông tin khách hàng Cập nhật thông tin tour Tư vấn khách hàng Tạo phiếu đăng ký tour

Đối tác sử dụng : Khách hàng

Diễn biến :

Hành động của tác nhân	Phản ứng hệ thống
1. Khách hàng xem tour, tìm kiếm tour rồi chọn tour để đăng ký	2. Yêu cầu khách hàng đăng ký thông tin cá nhân
3. Đăng ký thông tin cá nhân	4. Kiểm tra thông tin đăng ký
5. Đăng ký tour và thanh toán	6. Thông báo đăng ký thành công



Hình 2. Use case đăng ký tour du lịch

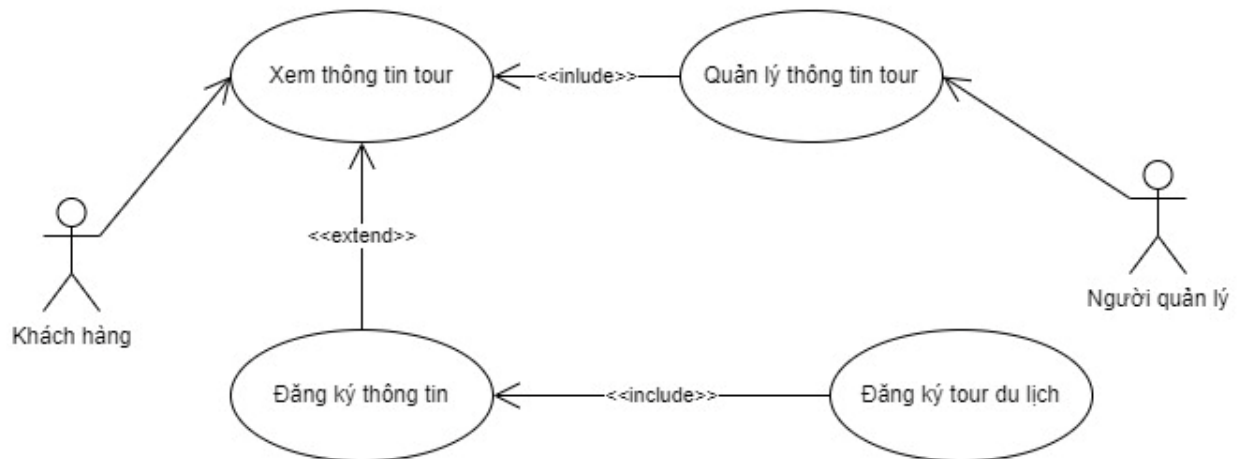
4.2 Đặt tả use case “Xem và quản lý thông tin du lịch”

Tên ca sử dụng : Xem và quản lý thông tin tour.

Mục đích : Giúp khách hàng xem thông tin tour, nhân viên có thể xem thông tin rồi từ đó quản lý tour đó.

Đối tác : Khách hàng, nhân viên

Hành động của tác nhân	Phản ứng của hệ thống
Khách hàng chọn xem tour	Hệ thống hiển thị thông tin tour
Người quản lý yêu cầu hệ thống cho phép sửa thêm xóa	Đáp ứng yêu cầu của người quản lý
Khách hàng xem tour và xem chi tiết lịch trình, thời gian...	Hiển thị thông tin cho khách hàng



Hình 3. Use case quản lý thông tin tour du lịch

4.3 Đặc tả use case “tìm kiếm”

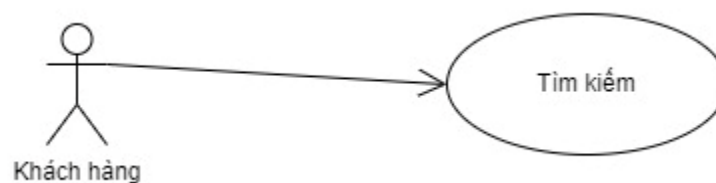
Tên ca sử dụng: Tìm kiếm.

Mục đích: Cho phép người dùng tìm kiếm các thông tin về tour du lịch như địa điểm...

Nội dung: Khách hàng tìm kiếm thông tin về các tour du lịch, xem thông tin và đăng ký.

Đối tác: Khách hàng

Hành động của tác nhân	Phản ứng của hệ thống
Chọn chức năng tìm kiếm	Hiện thanh tìm kiếm để khách hàng có thể nhập từ khóa để tìm kiếm.
Điền từ khóa tìm kiếm	Trả về thông tin tìm kiếm khách hàng đã nhập.



Hình 4. Use case tìm kiếm

4.4 Đặc tả use case quản lý thông tin nhân viên

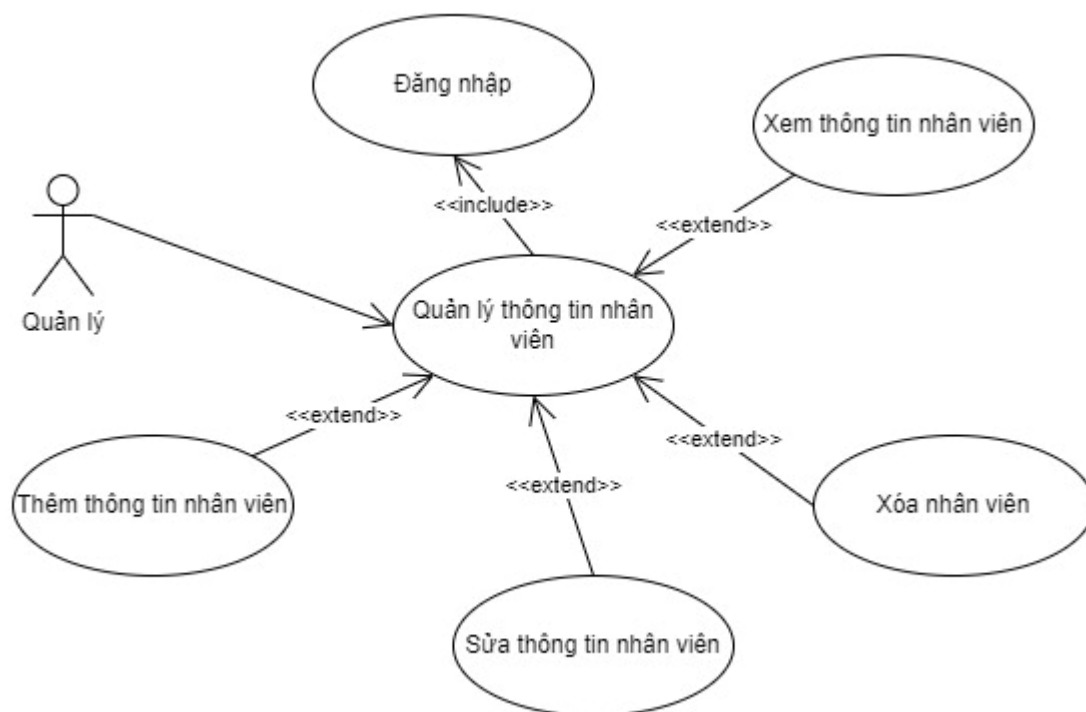
Tên ca sử dụng: Quản lý thông tin nhân viên

Mục đích: Nhà quản lý luôn cần nắm vững một số thông tin về nhân viên của mình.

Tóm lược: Ca sử dụng cho phép quản lý duy trì thông tin nhân viên trong hệ thống.

Đối tác: Quản lý

Điều kiện đầu vào: Ca sử dụng khi quản lý đăng nhập thành công vào hệ thống.



Hình 5. Use case quản lý thông tin nhân viên

4.5 Đặc tả use case đăng nhập

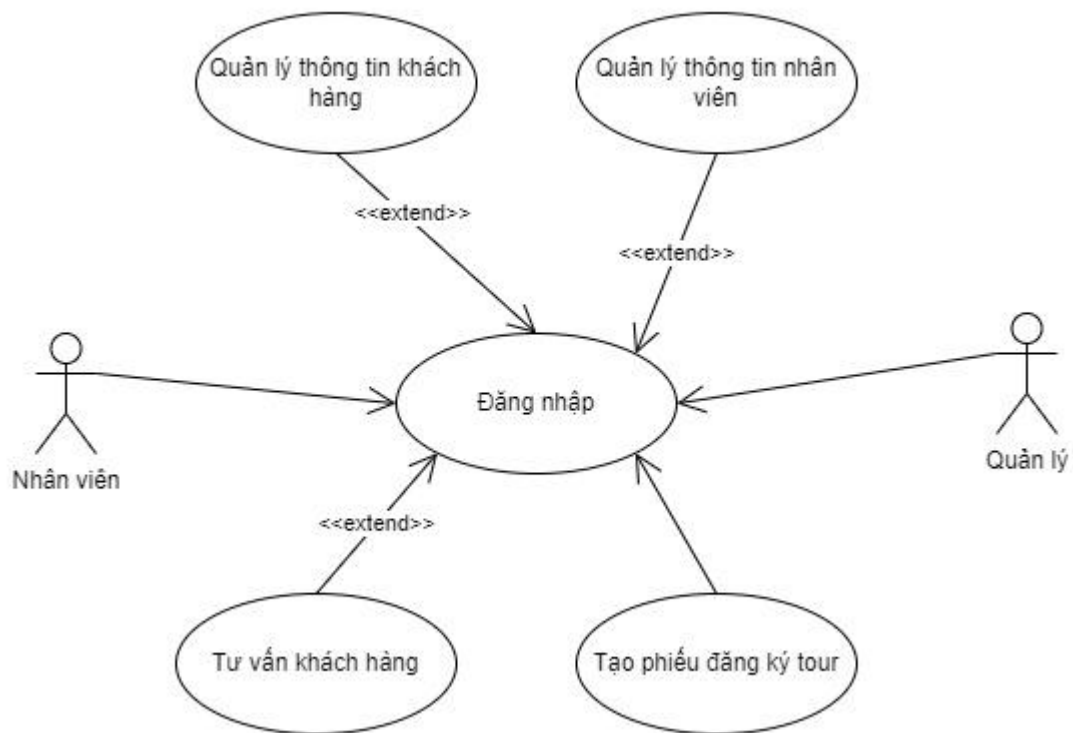
Tên use case: Đăng nhập

Mục đích: Cho phép nhân viên công ty đăng nhập qua tài khoản mật khẩu được cho trước đó.

Mỗi nhân viên có thể đăng nhập vào hệ thống để thay đổi thông tin các tour..

Đối tác: Nhân viên

Hành động của tác nhân	Phản ứng của hệ thống
Đăng nhập	Yêu cầu nhập tài khoản và mật khẩu
Nhập tài khoản và mật khẩu	Kiểm tra đúng hay sai. Nếu đúng cho phép đăng nhập vào hệ thống nếu sai yêu cầu đăng nhập lại.



Hình 6. Use case đăng nhập

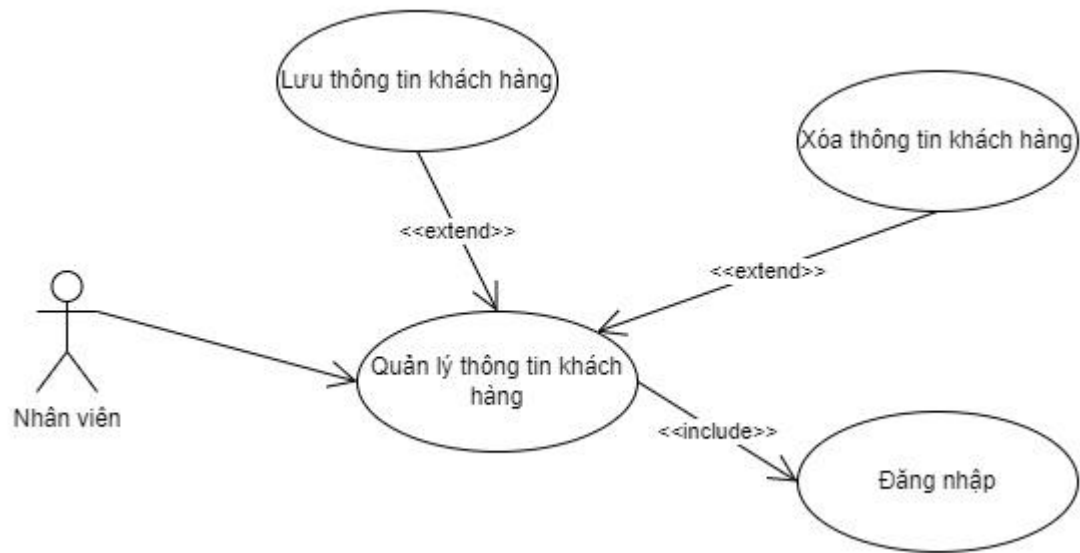
4.6 Đặc tả use case quản lý thông tin khách hàng

Tên ca sử dụng: Quản lý thông tin khách hàng

Mục đích: Kiểm soát thông tin khách hàng đã đăng ký tour du lịch.

Tóm lược: Sau khi khách hàng đăng ký tour và gửi thông tin cá nhân về cho hệ thống hệ thống sẽ lưu lại các thông tin đó.

Hành động tác nhân	Phản ứng của hệ thống
Đăng ký tour du lịch	Hiện bảng đăng ký gồm các thông tin cá nhân.
Nhập đầy đủ thông tin cá nhân	Lưu lại các thông tin đó. Nhân viên kiểm tra xem đúng hay không.
Yêu cầu xóa thông tin cá nhân nếu thấy không phù hợp, hoạt không còn yêu cầu hệ thống ghi nhận.	Ghi nhận yêu cầu của khách hàng.



Hình 7. Use case quản lý thông tin khách hàng

4.7 Đặc tả use case tư vấn khách hàng

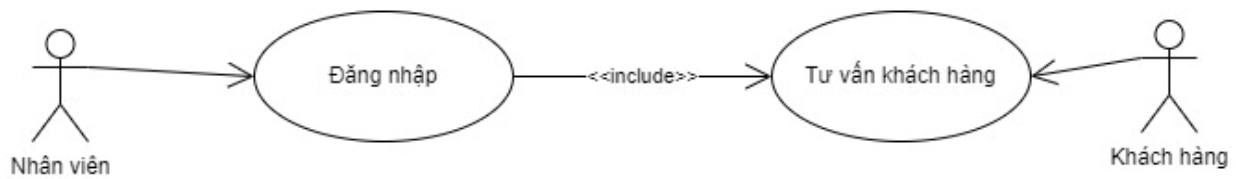
Tên ca sử dụng: tư vấn khách hàng.

Mục đích: Nhân viên sẽ tư vấn với khách hàng để giải đáp những thắc mắc của khách hàng.

Tóm lược: Ca sử dụng cho phép nhân viên dùng chức năng tư vấn để giải đáp những thắc mắc của khách hàng.

Đối tác: Nhân viên

Hành động của tác nhân	Phản ứng của hệ thống
Chọn tư vấn khách hàng	Hiển thị cửa sổ chat với khách hàng
Nếu chọn gửi tin nhắn	Nhân viên tiếp nhận thông tin thắc mắc của khách hàng và phản hồi lại cho khách hàng.
Nếu chọn gọi điện thoại	Nhân viên sẽ nghe máy và tư vấn cho khách hàng



Hình 8. Use case tư vấn khách hàng

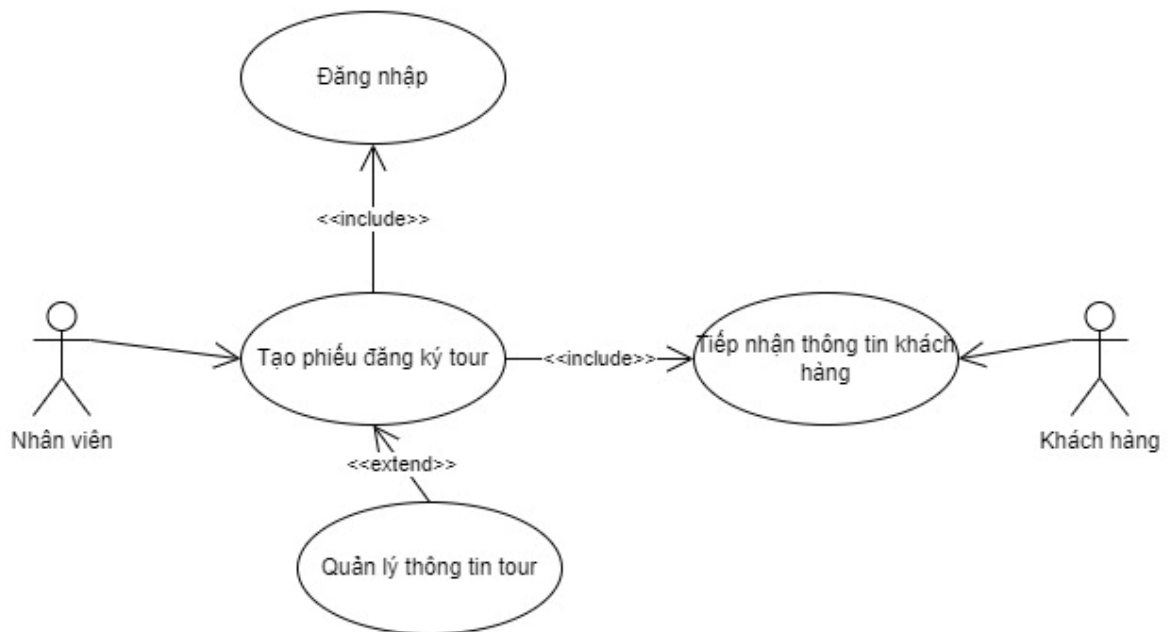
4.8 Đặc tả use case tạo phiếu đăng ký tour

Tên ca sử dụng: Tạo phiếu đăng ký tour

Mục đích: Cho phép nhân viên công ty lập phiếu đăng ký cho khách hàng bằng hình thức trực tiếp hoặc qua trang web.

Đối tác: Nhân viên

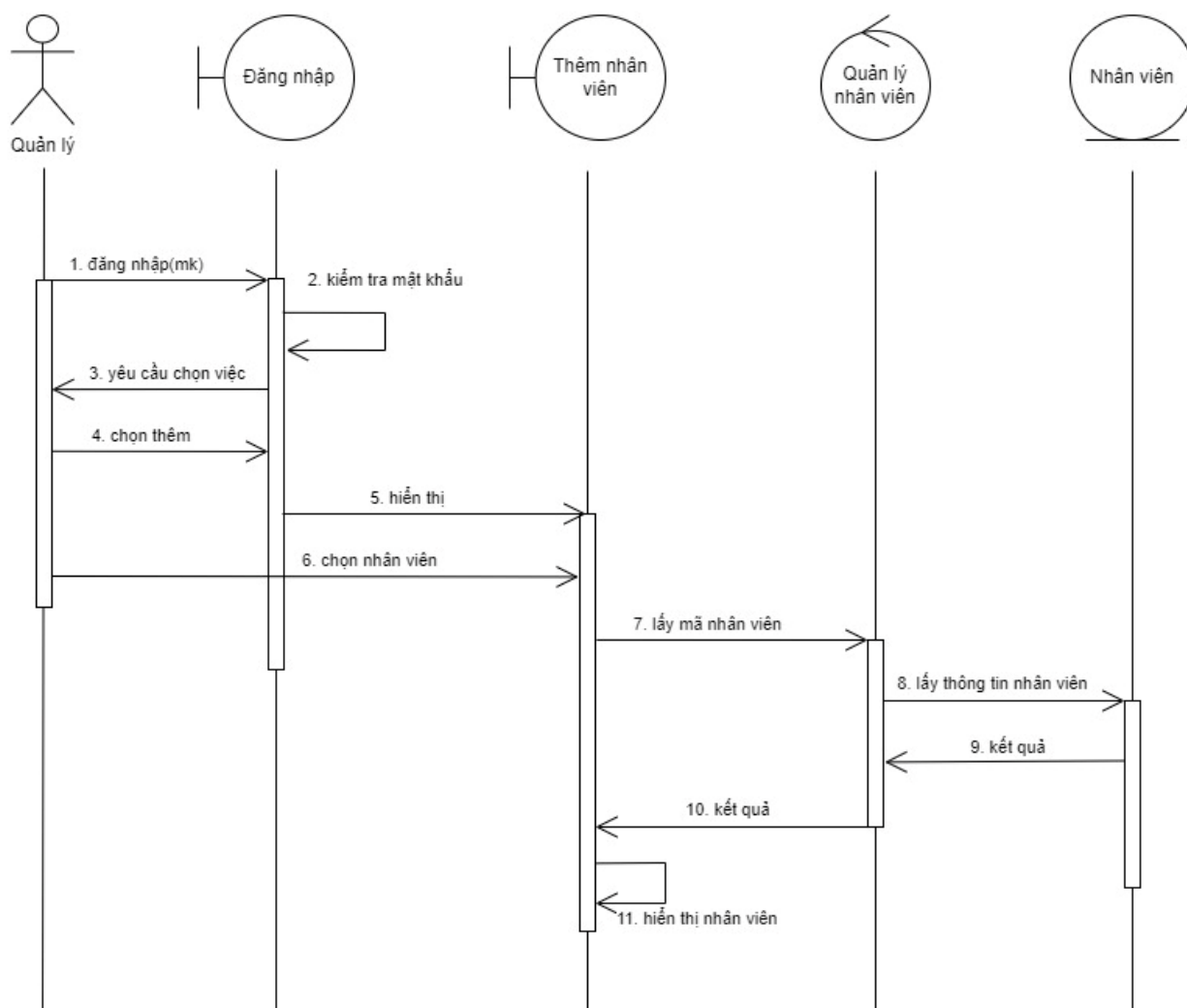
Hành động của tác nhân	Phản ứng của hệ thống
Nhân viên chọn tạo phiếu đăng ký tour	Hệ thống yêu cầu nhân viên chọn tour mà khách hàng đã chọn Hệ thống liệt kê danh sách các tour
Nhân viên nhập thông tin mà khách hàng đã đăng ký	Thông tin hợp lệ lưu thông tin. Còn không hợp lệ thì yêu cầu khách hàng nhập lại.



Hình 9. Use case tạo phiếu đăng ký tour

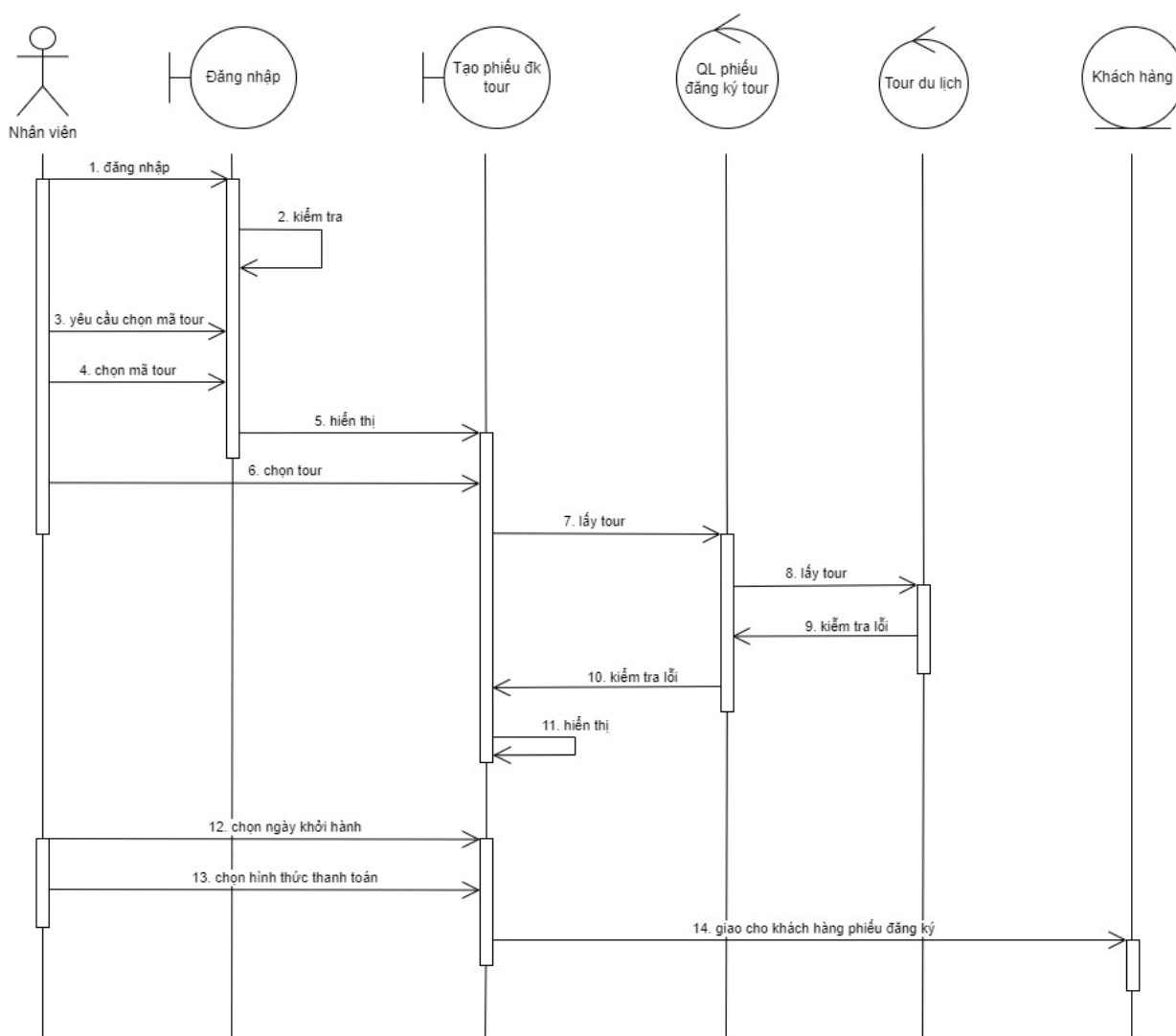
5. Biểu đồ tuần tự

5.1 Biểu đồ tuần tự quản lý nhân viên



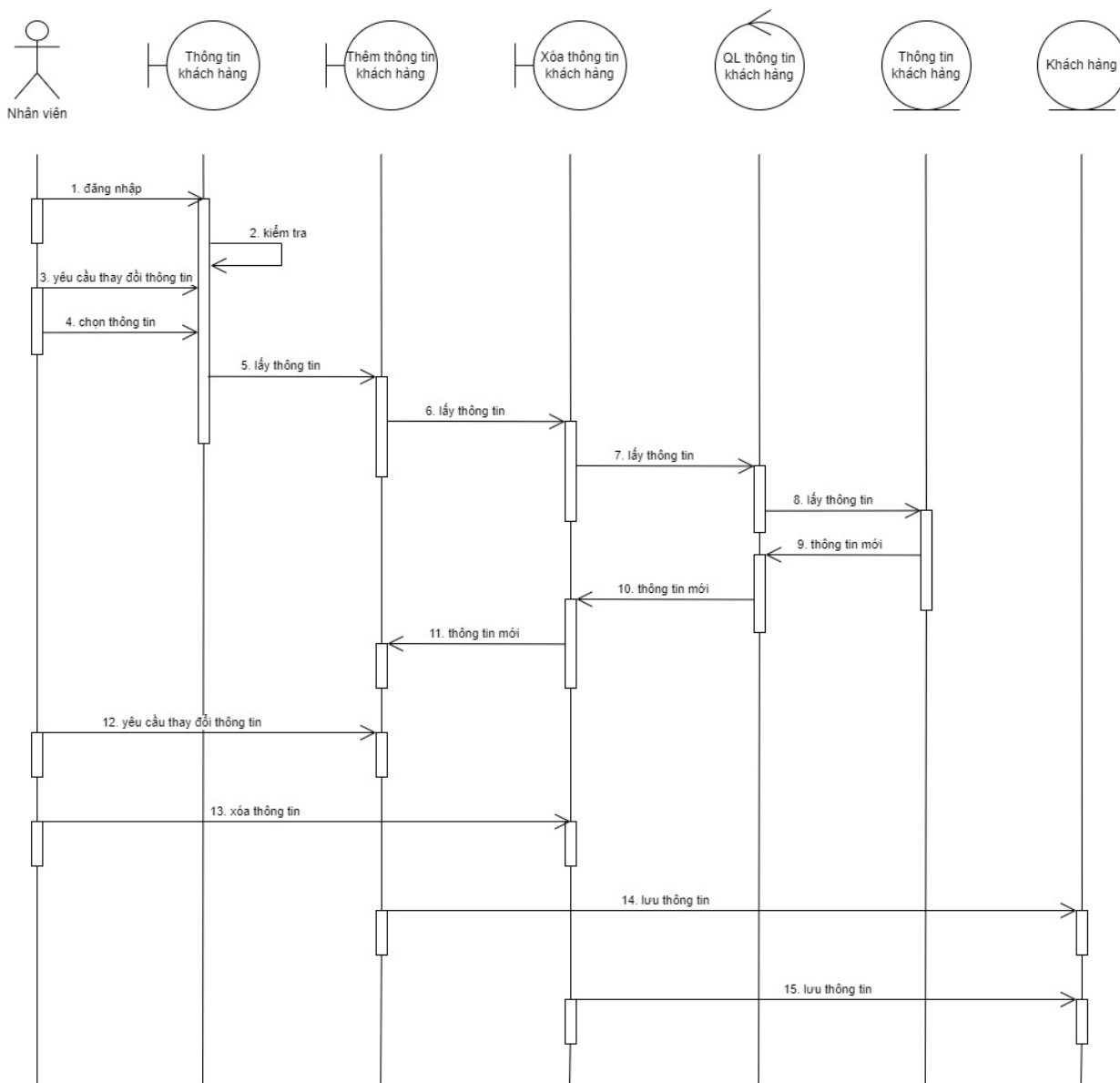
Hình 10. BDTT quản lý nhân viên

5.2 Biểu đồ tuần tự tạo phiếu đăng ký tour



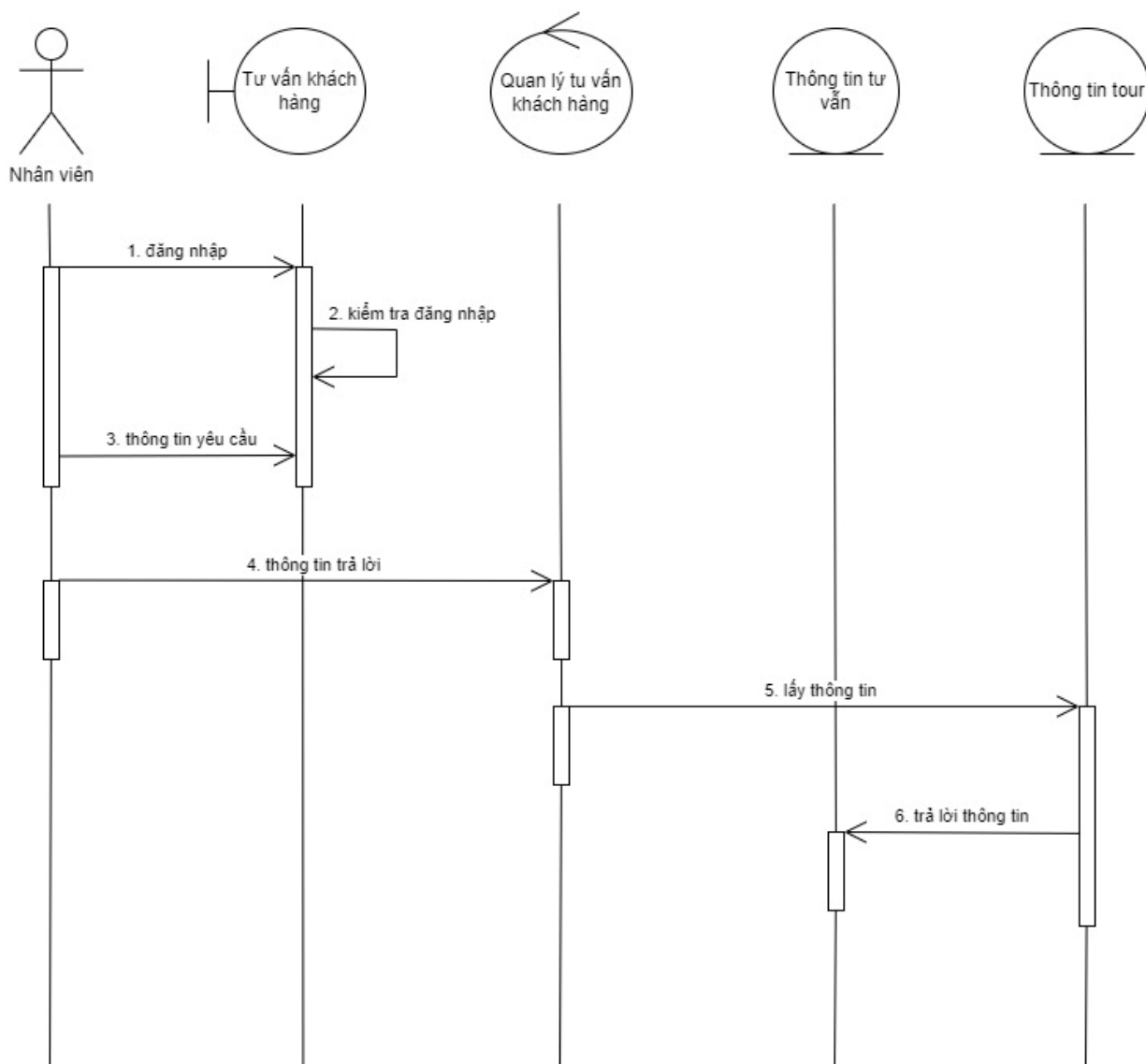
Hình 11. BDTT tạo phiếu đăng ký tour

5.3 Biểu đồ tuần tự quản lý thông tin khách hàng



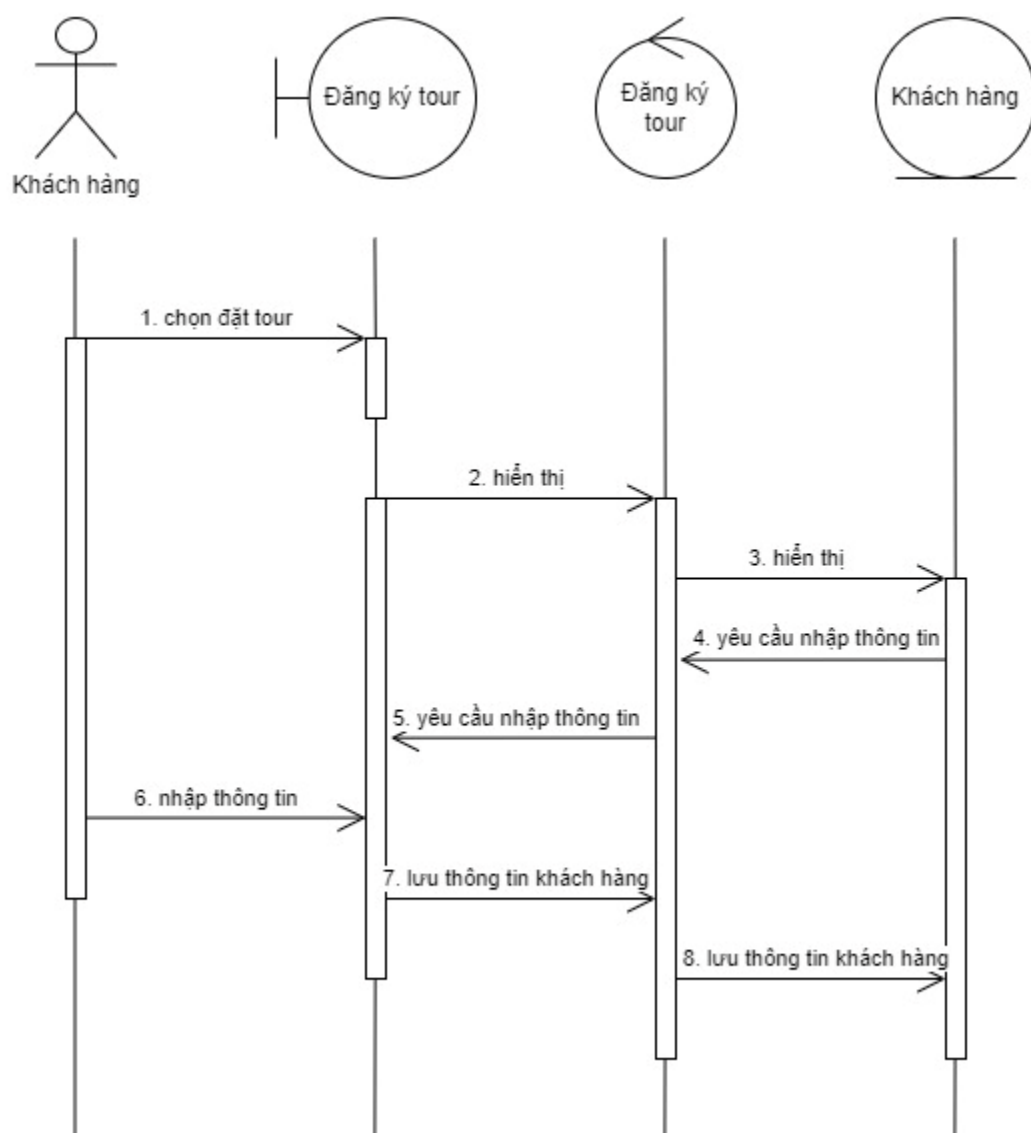
Hình 12. BDTT quản lý thông tin khách hàng

5.4 Biểu đồ tuần tự tư vấn khách hàng



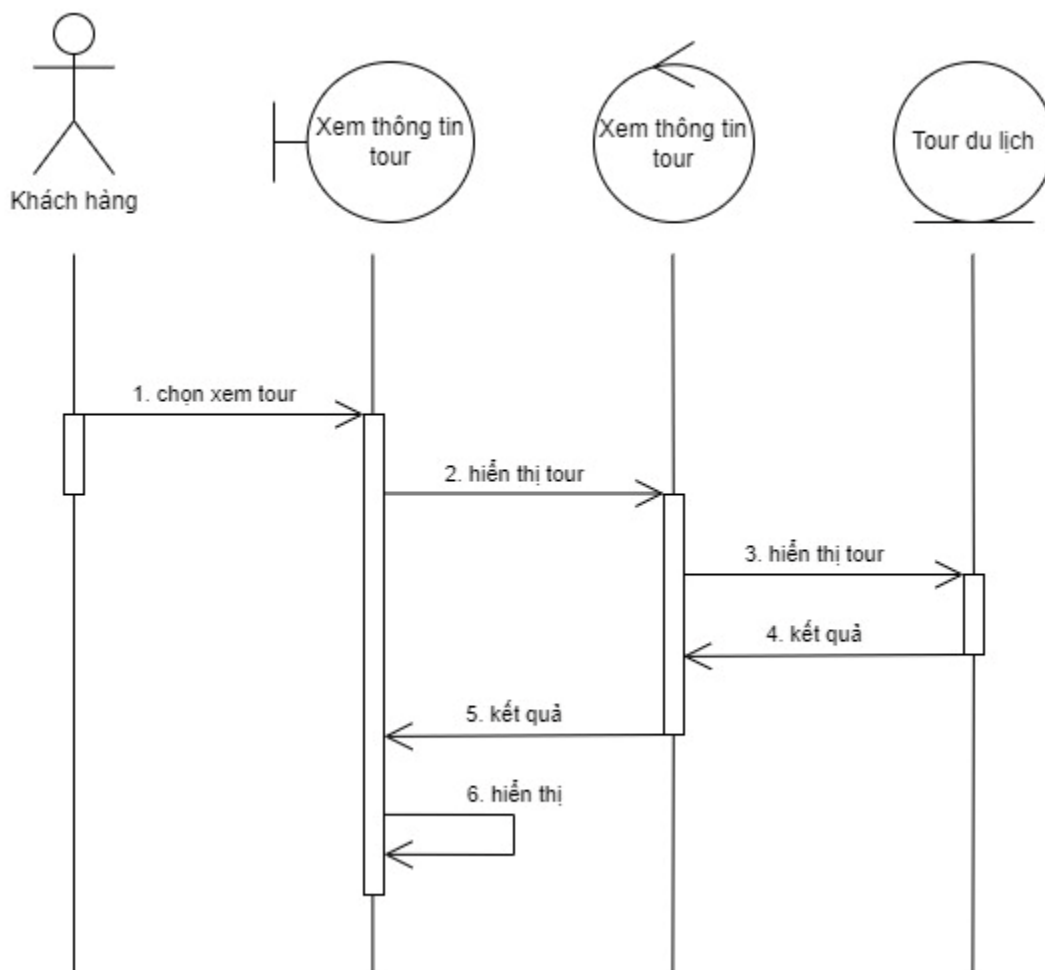
Hình 13. BDTT tư vấn khách hàng

5.5 Biểu đồ tuần tự đăng ký tour



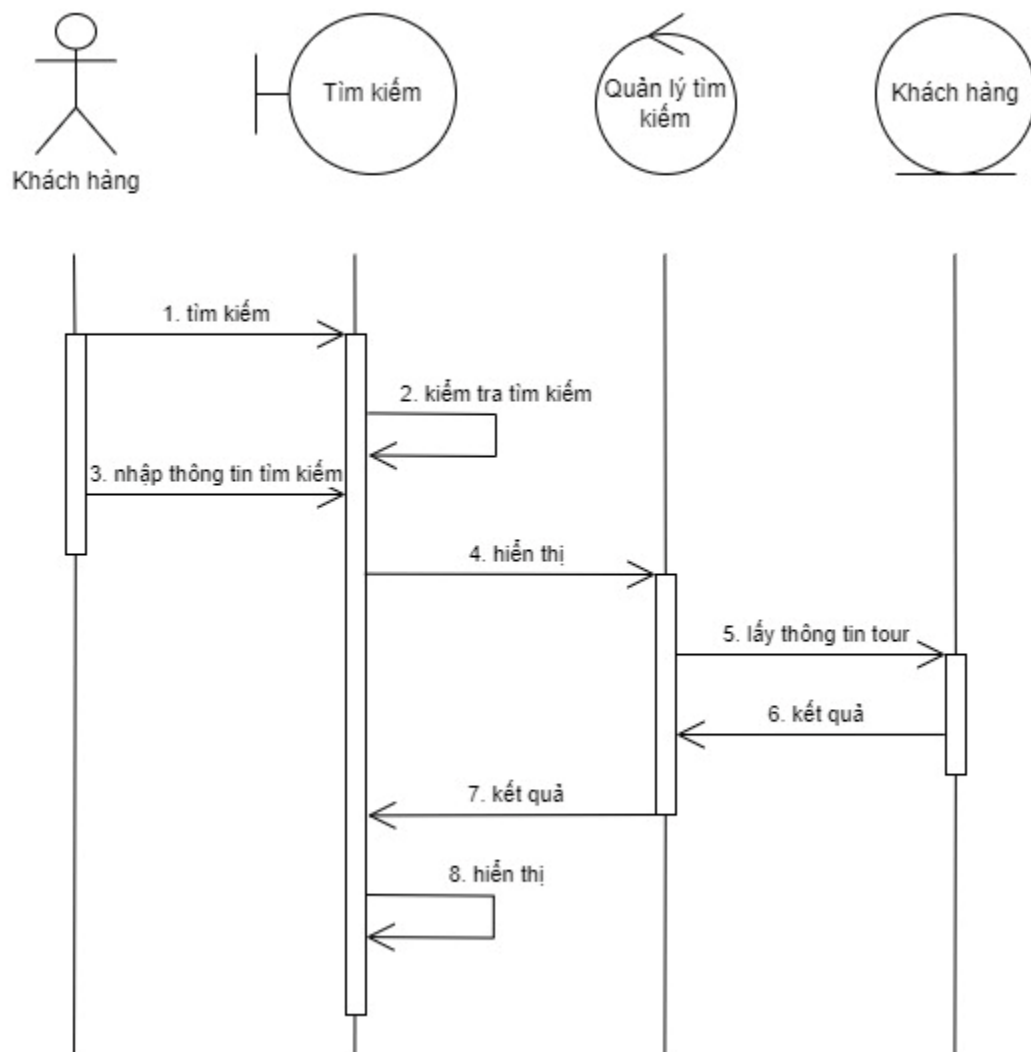
Hình 14. BDTT đăng ký tour

5.6 Biểu đồ tuần tự xem thông tin khách hàng



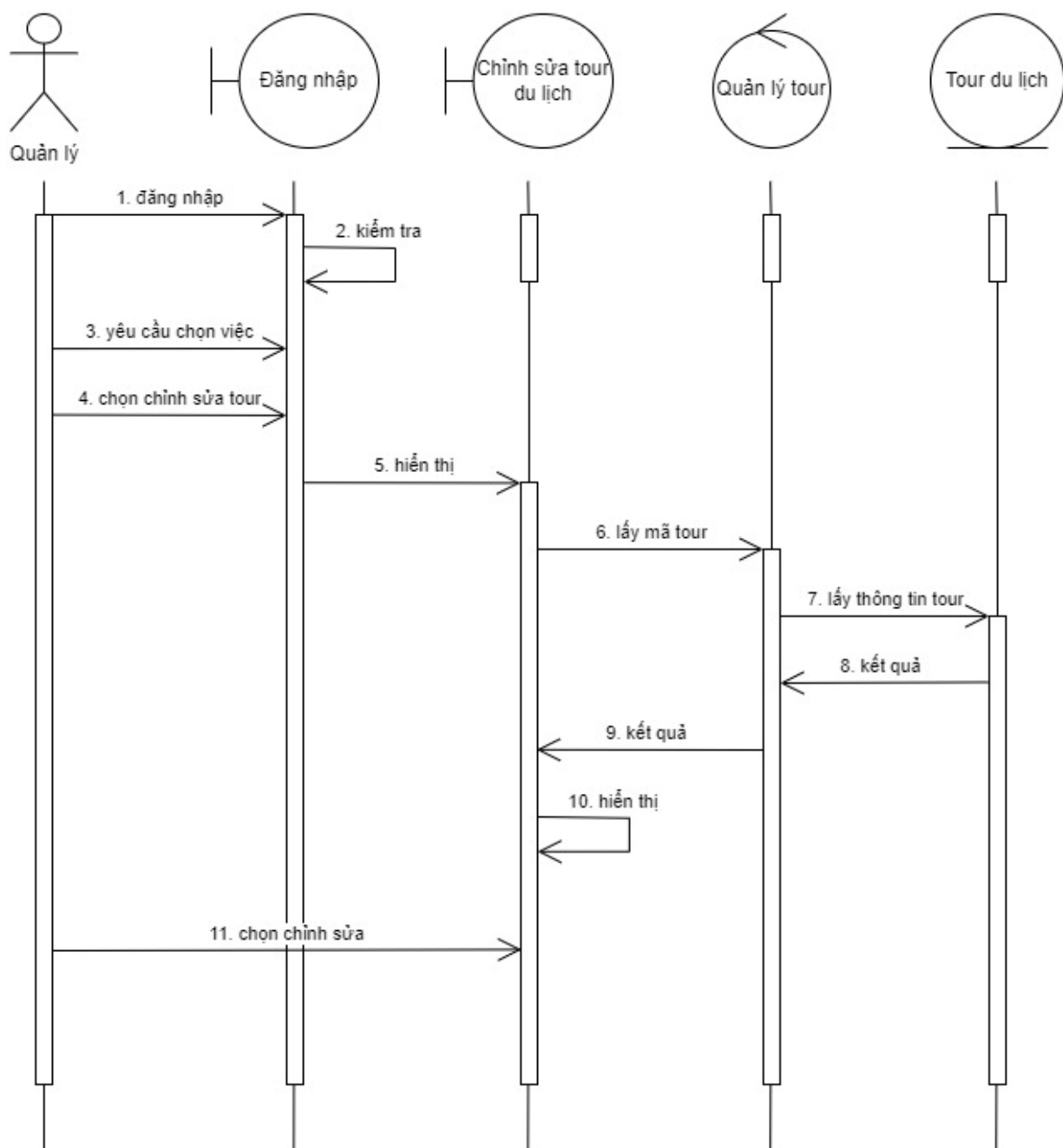
Hình 15. BDTT xem thông tin khách hàng

5.7 Biểu đồ tuần tự khách hàng tìm kiếm



Hình 16. BDTT khách hàng tìm kiếm

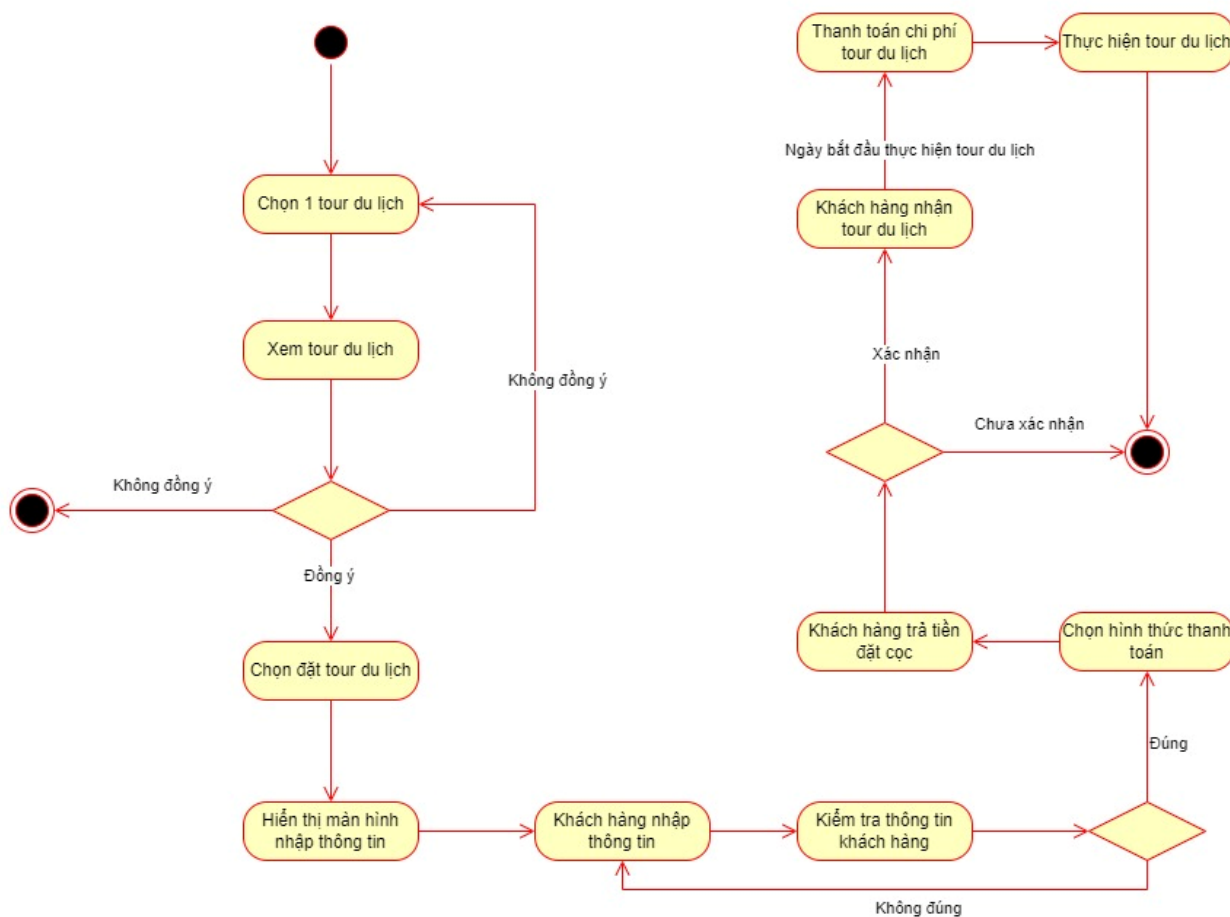
5.8 Biểu đồ tuần tự quản lý tour du lịch



Hình 17. BDTT quản lý tour du lịch

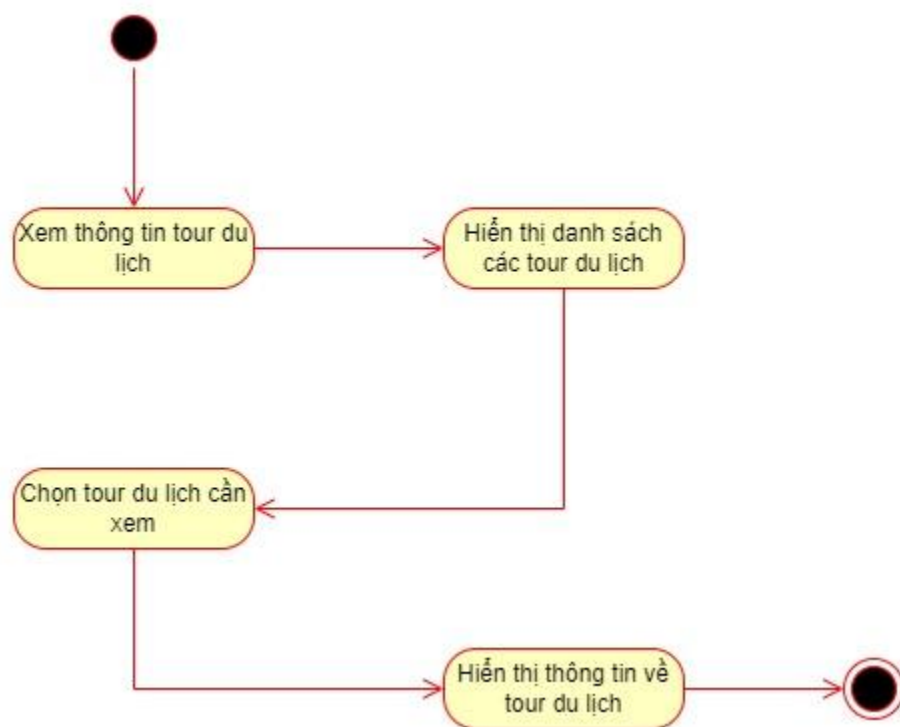
6. Biểu đồ hoạt động

6.1 Biểu đồ hoạt động khách hàng đăng ký tour du lịch



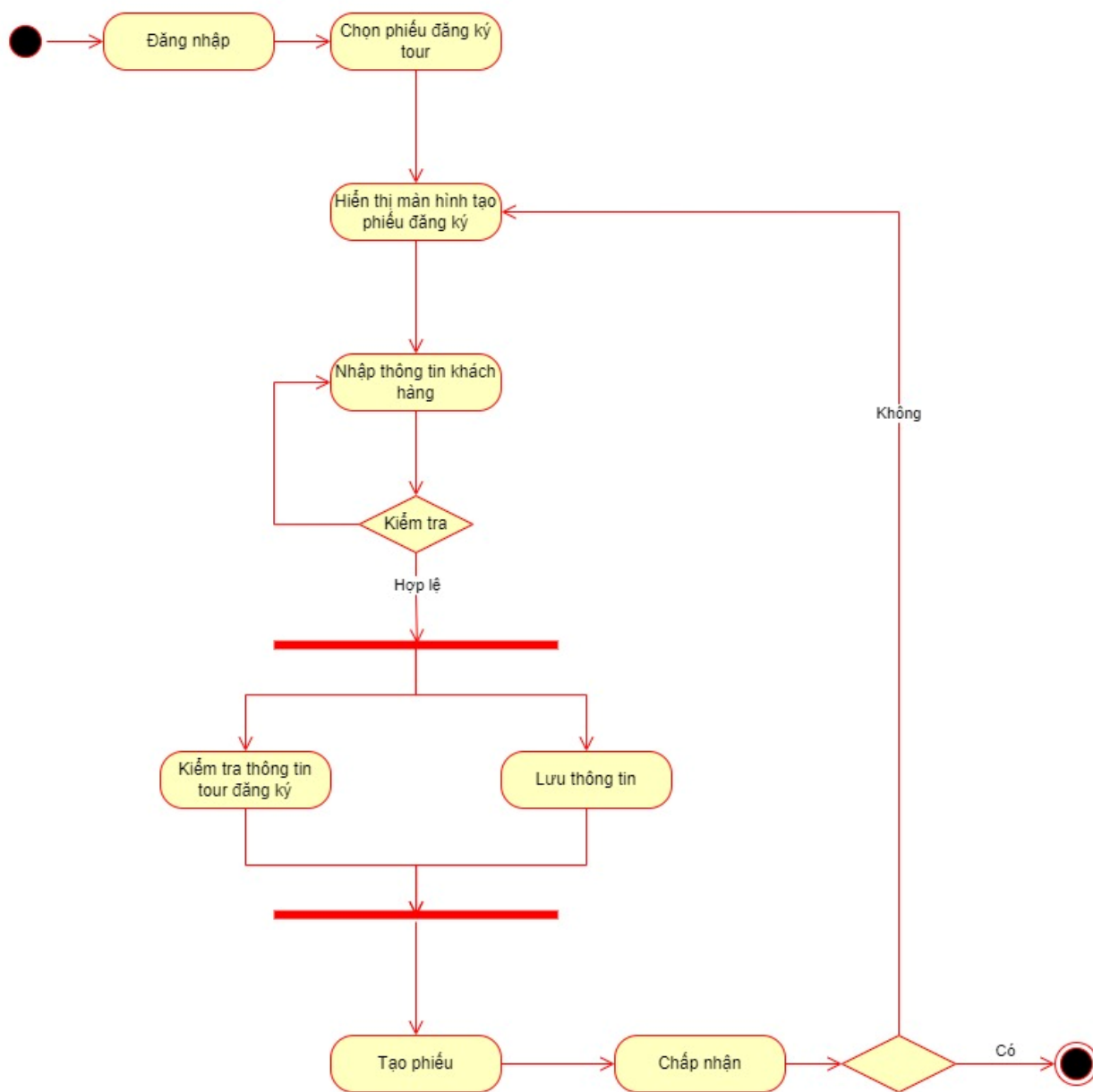
Hình 18. BDHD đăng ký tour du lịch

6.2 Biểu đồ hoạt động khách hàng xem tour du lịch



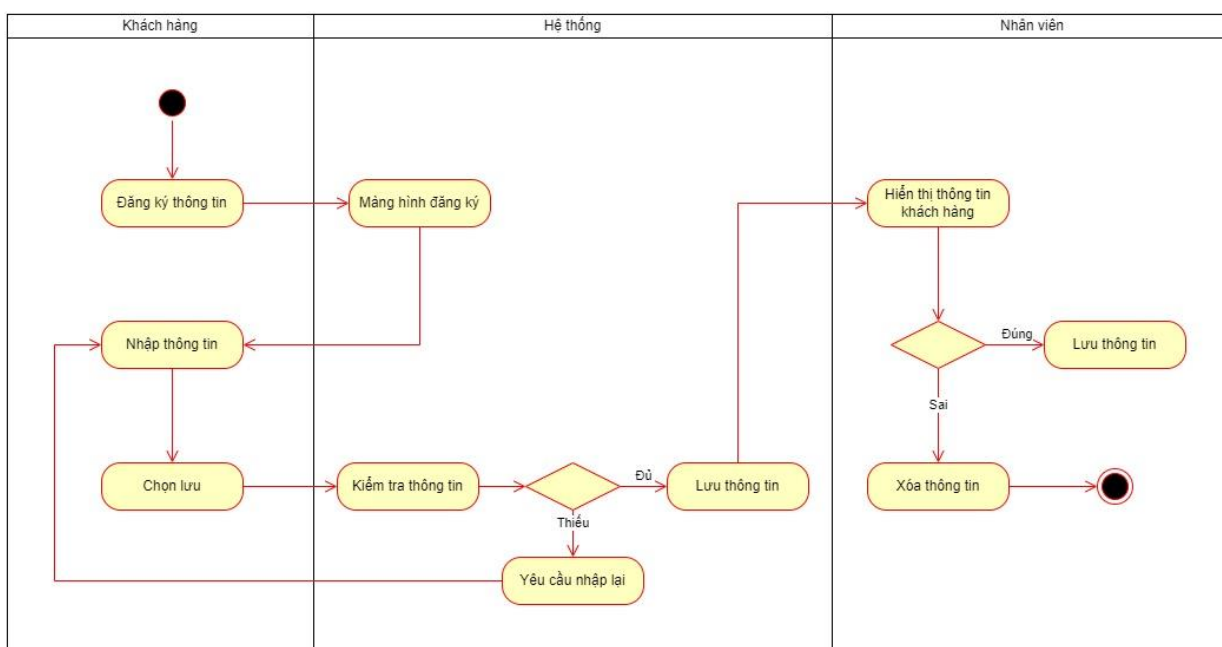
Hình 19. BDHD xem tour du lịch

6.3 Biểu đồ hoạt động tạo phiếu đăng ký tour



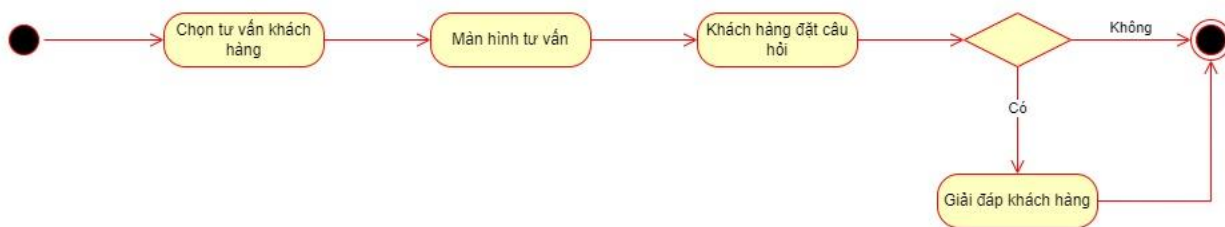
Hình 20. BDHD tạo phiếu đăng ký tour

6.4 Biểu đồ hoạt động quản lý thông tin khách hàng



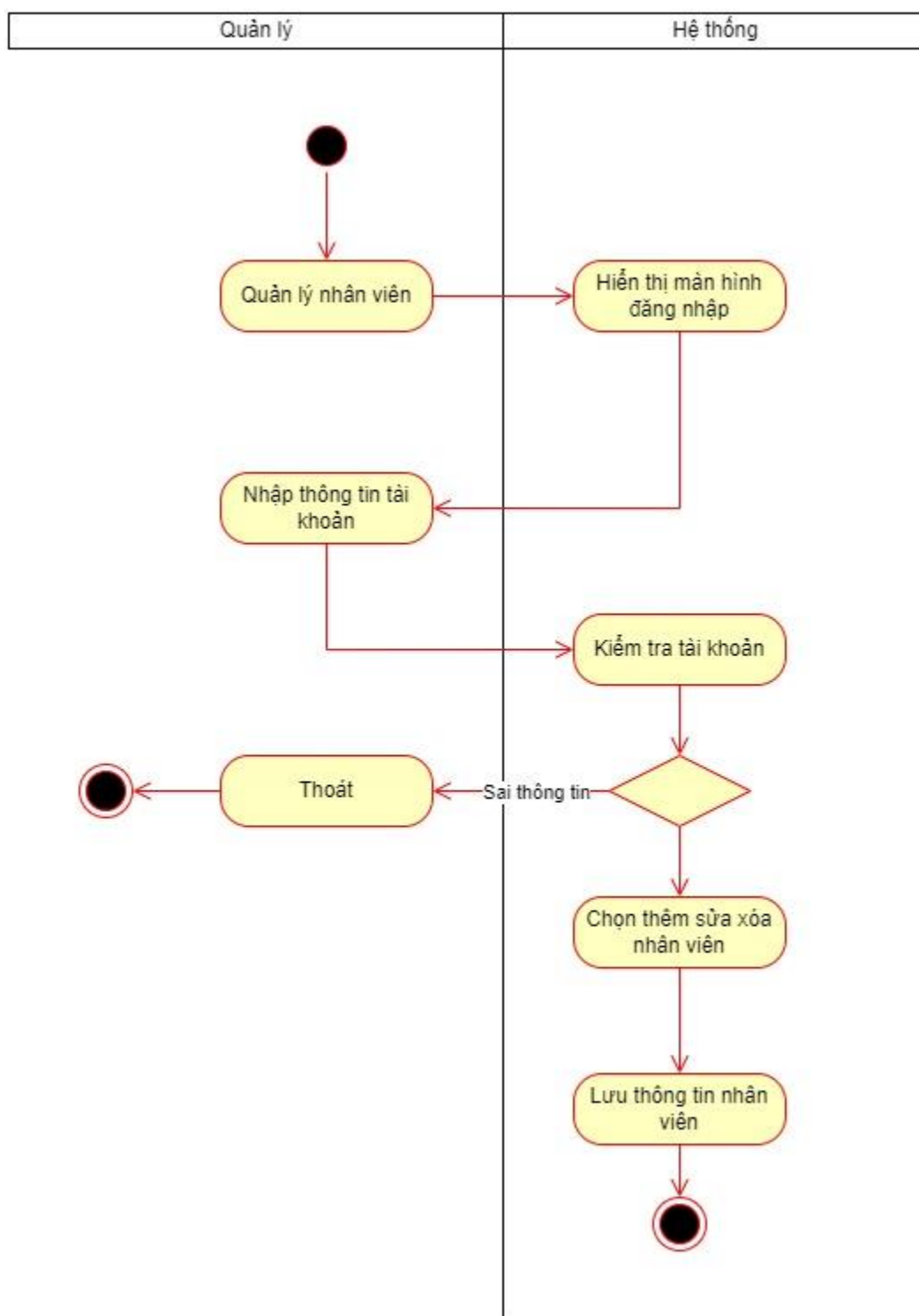
Hình 21. BDHD quản lý thông tin khách hàng

6.5 Biểu đồ hoạt động tư vấn khách hàng



Hình 22. BDHD tư vấn khách hàng

6.6 Biểu đồ hoạt động quản lý nhân viên



Hình 23. BDHD quản lý nhân viên

6.7 Biểu đồ hoạt động quản lý tour du lịch



Hình 24. BDHD quản lý tour du lịch

7. Biểu đồ lớp

7.1 Tìm lớp

Lớp Người dùng:

- Mã đăng nhập (taikhoan)
- Mật khẩu(matkhau)

Lớp Nhân viên:

- Mã nhân viên(manv)
- Tên nhân viên(tennv)
- Ngày sinh(ngaysinh)
- Giới tính (gioitinh)
- Số điện thoại(sdt)
- Email(email)
- Địa chỉ(diachi)

Lớp Khách:

- Tên khách hàng (tenkh)
- Mã khách hàng (makh)
- Giới tính(gioitinh)
- Số điện thoại(sdt)
- Email(email)
- Địa chỉ(diachi)
- Chứng minh thư / hộ chiếu (cmt/hochieu)

Lớp Phiếu đặt tour :

- Số phiếu(sophieu)
- Tên khách hàng(tenkh)
- Mã tour(matour)
- Loại tour(loaitour)
- Mã khách hàng(mach)
- Người lập phiếu(nguoilapphieu)

Lớp Vé du lịch:

- Số vé(sove)
- Tên tour(tentour)
- Giá vé(giave)
- Ngày khởi hành(ngaykhoi hanh)

Lớp Tour du lịch:

- Mã tour(matour)
- Tên touritentour)
- Giá (giá)
- Thời gian thoigian)
- Điểm khởi hành(diemkhoihanh)

Lớp Địa danh:

- Mã địa danh (madiadanh)
- Tên địa danh (tendiadanh)
- Mô tả(mota)
- Tỉnh thành tỉnh thành)

Gán trách nhiệm cho lớp vừa tìm được:

Khách hàng:

- Khách hàng là nguồn được đăng ký tour du lịch.
- Cung cấp các thông tin cần thiết cho nhân viên để đăng ký tour.

Người dùng:

Nhân Viên:

- Giúp khách hàng đăng ký tour.
- Tư vấn khách bằng việc chọn tour phù hợp giải đáp kịp thời thắc mắc của khách hàng

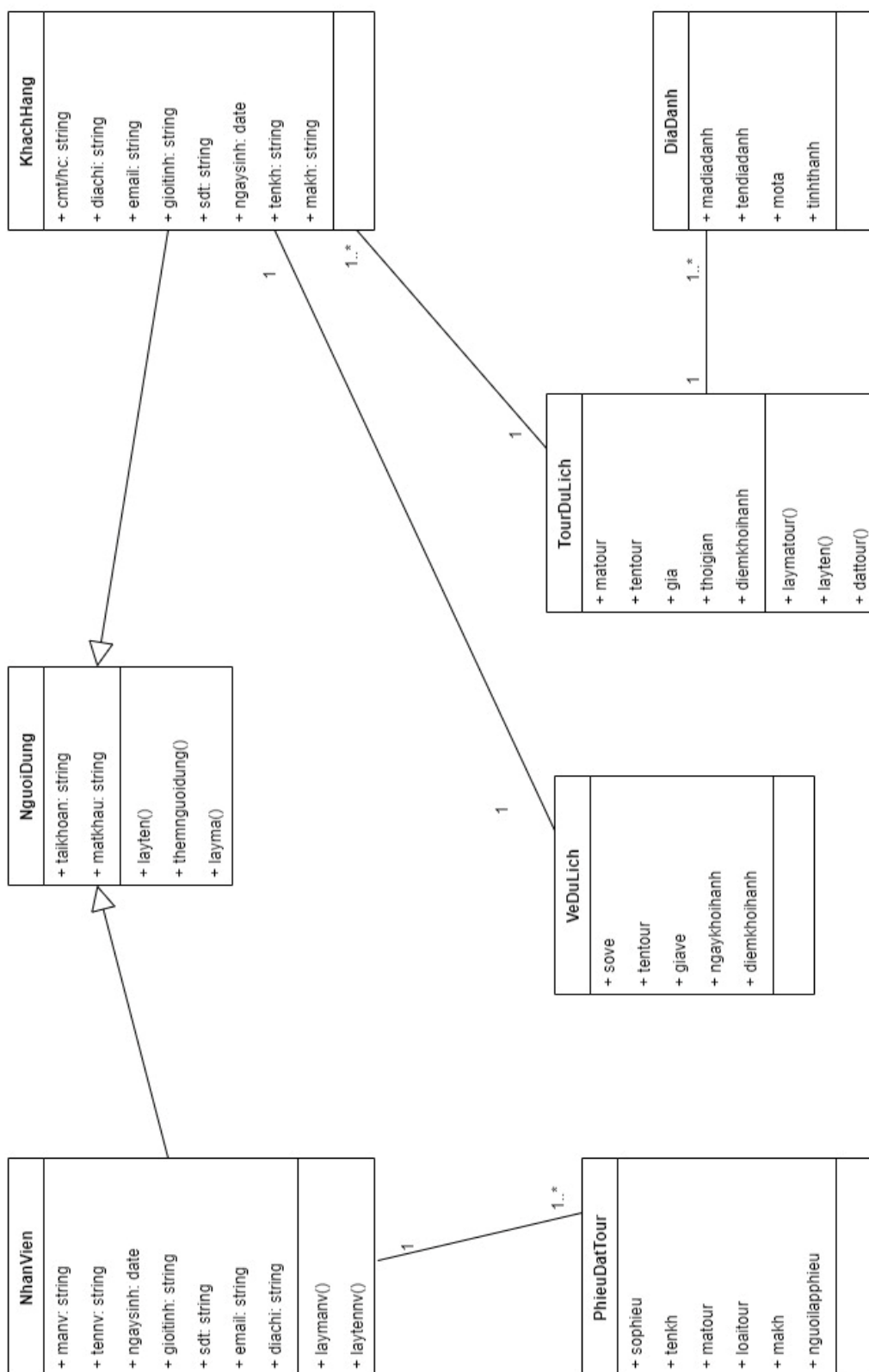
Tour Du Lịch:

- Là bảng giới thiệu về các thông tin cho khách hàng biết

Địa danh:

- Quản lý các địa điểm trong tour du lịch
- Giới thiệu về tour du lịch

7.2 Biểu đồ lớp chi tiết

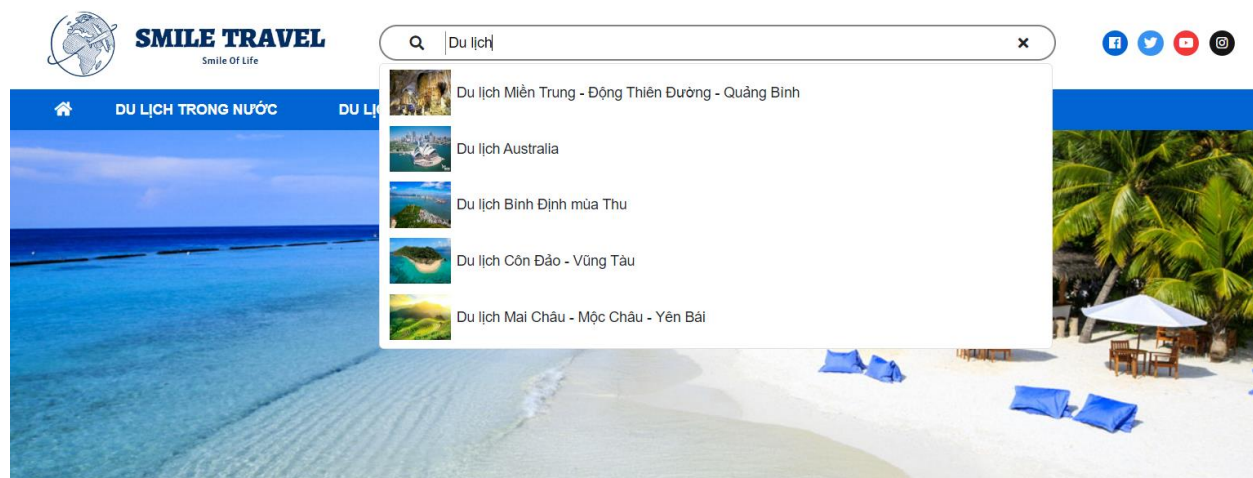


Hình 25. Biểu đồ lớp

CHƯƠNG V. THIẾT KẾ GIAO DIỆN

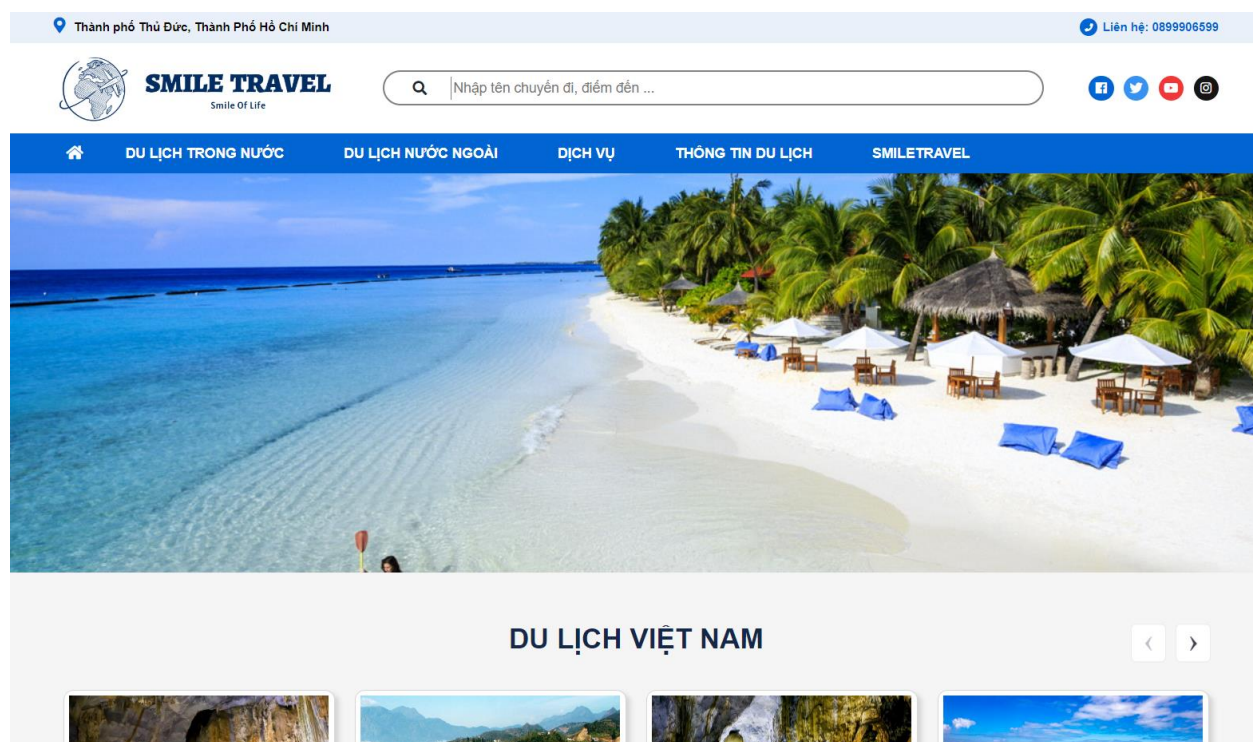
1. Một số giao diện hệ thống

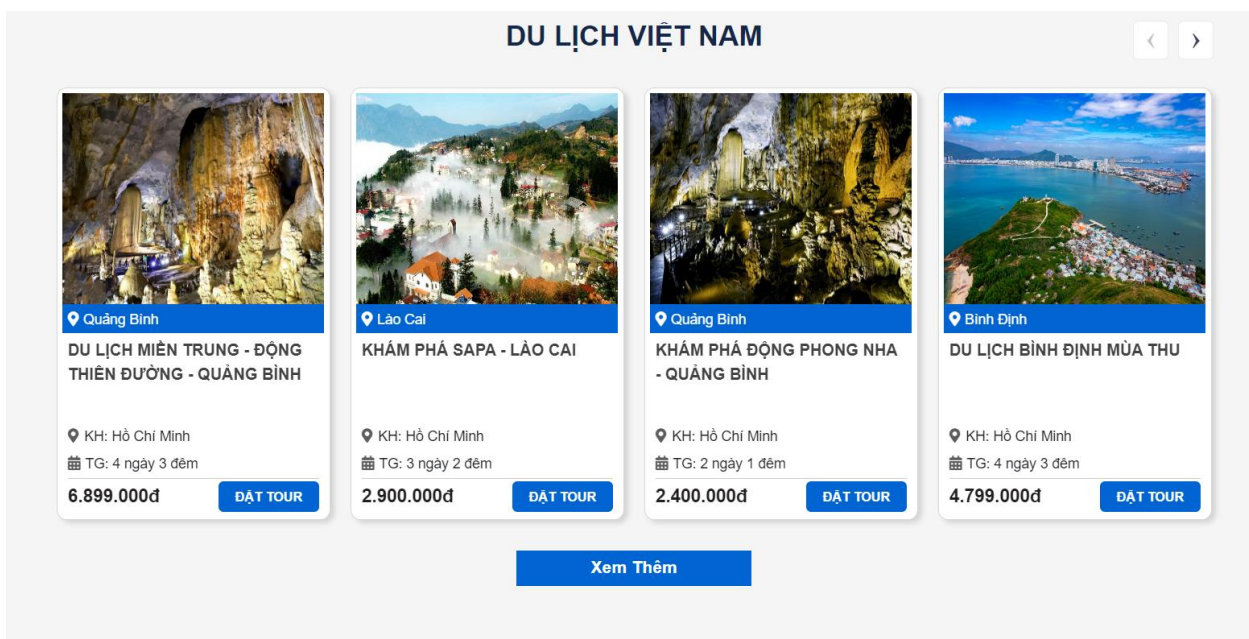
1.1 Giao diện tìm kiếm



Hình 26. Giao diện tìm kiếm

1.2 Giao diện trang Chủ





Hình 27. Giao diện trang chủ

1.3 Giao diện điền thông tin đặt tour

ĐẶT TOUR

THANH TOÁN

Chọn hình thức thanh toán

Tiền mặt

Chuyển Khoản

Quý khách vui lòng thanh toán tại 150 Lê Văn Việt, phường Tăng Nhơn Phú A, Tp. Thủ Đức, Tp. Hồ Chí Minh. Ngoài ra Quý khách có thể thanh toán bằng hình thức chuyển khoản.

THÔNG TIN LIÊN HỆ

Họ và tên:

Số điện thoại:

Giới tính:

Nam

Email:

Địa chỉ:

12

KHÁM PHÁ ĐỘNG PHONG NHA - QUẢNG BÌNH

KH: Hồ Chí Minh

01/07/2022

2 ngày 1 đêm

Số khách: 9

Tổng: 21.600.000 đ

Sau khi hoàn tất đơn hàng, nhân viên hỗ trợ sẽ liên hệ với quý khách để xác nhận tình trạng tour. Mọi thắc mắc, xin Quý khách vui lòng liên hệ 0899906599.

HOÀN TẤT ĐẶT TOUR

Hình 28. Giao diện đặt tour

Trang | 48

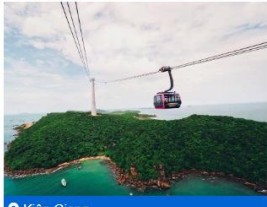
1.4 Giao diện tour du lịch

DU LỊCH TRONG NƯỚC

Việt Nam tự hào là đất nước được thiên nhiên ban tặng với nhiều điểm du lịch hấp dẫn. Smile Travel cung cấp các tour du lịch trong nước khởi hành từ TPHCM đến mọi miền đất nước – những nơi mà bạn chưa tới, đã tới và muốn tới.

Đến với tour trong nước của Smile Travel bạn sẽ có những giây phút thư giãn thoải mái, chiêm ngưỡng thiên nhiên thơ mộng và hùng vĩ, khám phá những nét văn hóa mới lạ, những di tích cổ kính và hiện đại của cộng đồng người Việt.

Chúng tôi hiện có các tour trong nước như sau:



Kiên Giang

PHÚ QUỐC - HÒN THƠM NATURE PARK - KIẾN GIANG

KH: Hồ Chí Minh
TG: 5 ngày 4 đêm

5.200.000đ **ĐẶT TOUR**



Đắk Lắk

DU LỊCH KHÁM PHÁ TÂY NGUYÊN

KH: Hồ Chí Minh
TG: 5 ngày 4 đêm

2.800.000đ **ĐẶT TOUR**



Khánh Hoà

DU LỊCH ĐẢO TÔM HÙM BÌNH BA - KHÁNH HOÀ

KH: Hồ Chí Minh
TG: 5 ngày 4 đêm

1.590.000đ **ĐẶT TOUR**

ĐIỂM ĐẾN DU LỊCH



- Du lịch Lào Cai
- Du lịch Yên Bái
- Du lịch Điện Biên
- Du lịch Hoà Bình
- Du lịch Lai Châu
- Du lịch Sơn La
- Du lịch Hà Giang
- Du lịch Cao Bằng
- Du lịch Bắc Kan
- Du lịch Lạng Sơn
- Du lịch Tuyên Quang
- Du lịch Thái Nguyên

Hình 29. Giao diện tour du lịch

CHƯƠNG VI. KẾT QUẢ VÀ KIẾN NGHỊ

Cùng với sự phát triển không ngừng của về kỹ thuật máy tính và mạng điện tử, công nghệ thông tin cũng được những công nghệ có đẳng cấp cao và lần lượt chinh phục hết đỉnh cao này đến đỉnh cao khác. Mạng internet là một trong những sản phẩm có giá trị hết sức lớn lao và ngày càng trở nên một công cụ không thể thiếu, là nền tảng chính cho sự truyền tải, trao đổi thông tin trên toàn cầu.

Bằng internet, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Vì vậy mà hàng loạt các phần mềm ra đời phục vụ nhu cầu của người dùng và các công ty. Chính vì thế các công ty đều quảng bá và giới thiệu thông tin qua các ứng dụng phần mềm.

Đề tài “ Quản lý tour du lịch” nhằm xây dựng một phần mềm với mong muốn cung cấp các thông tin về các tour du lịch của các công ty du lịch đến các khách hàng, giúp cho khách hàng dễ dàng tìm kiếm và đặt tour nhanh chóng và tiện lợi.

Về cơ bản, phần mềm đã thực hiện được các chức năng đề ra như sau:

- + Cho phép khách hàng đăng ký và đặt tour

- + Quản lý thông tin khách hàng.

- +Quản lý nhân viên, tour du lịch, Đề tài này, em thực hiện về cơ bản đã đạt được những mục tiêu đề ra. Ngoài ra, em cũng mong được sự giúp đỡ của các thầy cô trong khoa CNTT để có thể đưa ra một số phương hướng để phát triển được đề lớn hơn và hoàn thiện hơn.

Em xin chân thành cảm ơn!!!

TÀI LIỆU THAM KHẢO

- [1]. Đặng Văn Đức, Phân tích thiết kế hệ thống đối tượng bằng UML, nhà xuất bản Khoa học và kỹ thuật 2002.
- [2]. Tài liệu về Reactjs Getting Started – React (reactjs.org)
- [3]. Tài liệu về Nodejs Documentation | Node.js (nodejs.org)