

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

ĐỀ TÀI:
XÂY DỰNG ỨNG HỖ TRỢ
ĐẶT KHÁCH SẠN VÀ TOUR DU LỊCH

Người hướng dẫn : TS.Trương Ngọc Châu
Sinh viên thực hiện : Nguyễn Hoàng Vũ
Số thẻ sinh viên : 102200201
Lớp : 20TCLC_DT4

Đà Nẵng, 06/2024

TÓM TẮT

Tên đề tài: XÂY DỰNG ỨNG DỤNG HỖ TRỢ ĐẶT KHÁCH SẠN VÀ TOUR DU LỊCH

Sinh viên thực hiện: Nguyễn Hoàng Vũ

Số thẻ SV: 102200201

Lớp: 20TCLC_DT4

Tóm tắt đề tài:

Đề tài "Xây dựng ứng dụng hỗ trợ đặt khách sạn và tour du lịch" tập trung vào việc phát triển một ứng dụng di động đáp ứng nhu cầu đặt khách sạn và tour du lịch cho người dùng. Ứng dụng sẽ cung cấp các chức năng hỗ trợ đặt phòng khách sạn và tour du lịch, bao gồm thông tin chi tiết về các khách sạn và tour du lịch, lịch trình, giá cả, và khả năng đặt phòng và tour trực tuyến.

Mục tiêu chính của đề tài:

- **Tạo ra một nền tảng trực tuyến thuận tiện và linh hoạt:** Ứng dụng sẽ giúp người dùng dễ dàng tìm kiếm và lựa chọn các khách sạn và tour du lịch phù hợp với nhu cầu của họ.
- **Cung cấp thông tin chi tiết:** Người dùng có thể xem thông tin chi tiết về khách sạn (địa điểm, tiện nghi, giá phòng) và các tour du lịch (địa điểm, lịch trình, giá tour).
- **Đặt phòng và tour trực tuyến:** Người dùng có thể đặt phòng khách sạn và tour du lịch trực tuyến một cách dễ dàng và nhanh chóng.
- **Quản lý đặt phòng và tour:** Ứng dụng sẽ cho phép người dùng quản lý các đơn đặt phòng và tour, xác nhận đặt chỗ, và xử lý thanh toán.
- **Bảo mật và bảo vệ thông tin cá nhân:** Hệ thống cần đảm bảo tính bảo mật và bảo vệ thông tin cá nhân của người dùng.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Nguyễn Hoàng Vũ

Số thẻ sinh viên: 102200201

Lớp: 20TCLC_DT4

Khoa: Công nghệ thông tin

Ngành: Hệ thống thông tin

1. Tên đề tài đồ án: XÂY DỰNG ỨNG DỤNG HỖ TRỢ ĐẶT KHÁCH SẠN VÀ TOUR DU LỊCH

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu: Không có

4. Nội dung các phần thuyết minh và tính toán:

Đồ án tốt nghiệp bao gồm:

Mở đầu: Giới thiệu về tổng quan đề tài.

➤ **Chương 1:** Cơ sở lý thuyết

➤ **Chương 2:** Phân tích và thiết kế hệ thống

➤ **Chương 3:** Triển khai dự án và kết quả

Kết luận: Rút ra những bài học đạt được qua quá trình xây dựng hệ thống, những điểm hạn chế và phương hướng phát triển trong tương lai

5. Họ tên người hướng dẫn: TS. Trương Ngọc Châu.

6. Ngày giao nhiệm vụ đồ án: 07/04/2024

7. Ngày hoàn thành đồ án: 29/06/ 2024

Đà Nẵng, ngày tháng năm 2024

Trưởng Bộ môn

Người hướng dẫn

TS. Trương Ngọc Châu

LỜI NÓI ĐẦU

Em xin gửi lời cảm ơn sâu sắc đến các thầy cô trong khoa Công nghệ thông tin, trường Đại học Bách khoa Đà Nẵng. Nhờ những kiến thức và kinh nghiệm quý báu mà các thầy cô đã truyền đạt trong suốt những năm qua, em đã có nền tảng vững chắc để học tập và thực hiện đề tài tốt nghiệp này.

Đặc biệt, em xin bày tỏ lòng biết ơn chân thành đến thầy TS. Trương Ngọc Châu, người đã hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án. Sự tận tình, những lời khuyên hữu ích và những góp ý của thầy đã giúp em hoàn thiện và phát triển ứng dụng một cách tốt nhất.

Dù đã nỗ lực học tập, trau dồi kiến thức và tìm hiểu để hoàn thành đồ án, em hiểu rằng không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm, góp ý và chỉ bảo từ quý Thầy Cô.

Em xin chân thành cảm ơn!

Đà Nẵng, ngày 20 tháng 06 năm 2024

Sinh viên thực hiện

Nguyễn Hoàng Vũ

CAM ĐOAN

LỜI CAM ĐOAN

Em xin cam đoan:

1. Nội dung trong luận văn này là do em thực hiện dưới sự hướng dẫn trực tiếp của Thầy TS. Trương Ngọc Châu.
2. Các tham khảo dùng trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
3. Nếu có những sao chép không hợp lệ, vi phạm, em xin chịu hoàn toàn trách nhiệm

Đà Nẵng, ngày 20 tháng 06 năm 2024

Sinh viên thực hiện

Nguyễn Hoàng Vũ

MỤC LỤC

TÓM TẮT

LỜI NÓI ĐẦU

LỜI CAM ĐOAN

MỤC LỤC

DANH SÁCH CÁC BẢNG BIỂU, HÌNH VẼ VÀ SƠ ĐỒ

DANH SÁCH CÁC CỤM TỪ VIẾT TẮT

MỞ ĐẦU.....	1
Chương 1: CƠ SỞ LÝ THUYẾT	3
1. Mô hình MVC	3
1.1. Các thành phần trong mô hình MVC	3
1.2. Nguyên lý hoạt động trong mô hình MVC	4
2. Tổng quan về RESTful API.....	4
2.1. Các thành phần của RESTful API.....	4
2.2. Cách RESTful API hoạt động	5
2.3. Status code:	6
3. Khái niệm cơ bản về UML.....	7
4. Tổng quan về NodeJS.....	7
4.1. Giới thiệu	7
4.2. Ưu điểm	8
4.3. Nhược điểm	9
5. Tổng quan về React Native.....	9
5.1. Giới thiệu	9
5.2. Ưu điểm	10
5.3. Nhược điểm	10
6. Tổng quan về MongoDB	11
6.1. Giới thiệu	11
6.2. Lợi ích khi sử dụng MongoDB	11
6.3. Yếu tố nổi bật	12
7. Các công cụ khác	13
7.1. Cloundinary	13
8. Kết chương.....	13
Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	14
1. Phân tích chức năng và yêu cầu	14

1.1.	Chức năng cơ bản.....	14
1.2.	Tác nhân.....	14
1.3.	Yêu cầu	15
2.	Sơ đồ	15
2.1.	Sơ đồ usecase.....	15
2.2.	Sơ đồ tuần tự.....	18
2.3.	Sơ đồ hoạt động	24
2.4.	Sơ đồ lớp.....	34
2.5.	Sơ đồ cơ sở dữ liệu.....	34
3.	Kết chương.....	38
Chương 3: TRIỂN KHAI DỰ ÁN VÀ KẾT QUẢ		38
1.	Môi trường triển khai dự án.....	38
2.	Kết quả dự án.....	39
2.1.	Màn hình đăng nhập và đăng ký	39
2.2.	Màn hình quên mật khẩu	40
2.3.	Màn hình trang chủ.....	41
2.4.	Màn hình khách sạn	41
2.5.	Màn hình chi tiết khách sạn	42
2.6.	Màn hình chi tiết phòng khách sạn.....	43
2.7.	Màn hình tour	43
2.8.	Các màn hình khác.....	44
2.9.	Màn hình thống kê	44
2.10.	Màn hình quản lý người dùng.....	45
2.11.	Màn hình thêm người dùng.....	45
2.12.	Màn hình chi tiết người dùng	46
2.13.	Màn hình danh sách tour.....	46
2.14.	Màn hình thêm mới tour.....	47
2.15.	Màn hình danh sách khách sạn.....	47
2.16.	Màn hình thêm mới khách sạn.....	48
2.17.	Màn hình chi tiết khách sạn	48
2.18.	Màn hình danh sách các loại phòng khách sạn	48
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		49

1.	Kết luận	49
2.	Hạn chế	49
3.	Hướng phát triển	49
	TÀI LIỆU THAM KHẢO.....	50

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Danh sách hình vẽ

Hình 1. 1 Các thành phần trong mô hình MVC	3
Hình 1. 2 Nguyên lý hoạt động trong mô hình MVC.....	4
Hình 1. 3 Tổng quan về RESTful API	5
Hình 1. 4 Tổng quan về NodeJS.....	8
Hình 1. 5 Tổng quan về React Native	9
Hình 1. 6 Tổng quan về MongoDB	11
Hình 1. 7 Cloundinary	13
Hình 2. 1 Sơ đồ usecase tổng quát của hệ thống	15
Hình 2. 2 Sơ đồ usecase của người dùng.....	16
Hình 2. 3 Sơ đồ usecase của người quản lý.....	17
Hình 2. 4 Sơ đồ tuần tự đăng nhập hệ thống	18
Hình 2. 5 Sơ đồ tuần tự đăng ký tài khoản	19
Hình 2. 6 Sơ đồ tuần tự tìm kiếm chỉnh sửa thông tin cá nhân	20
Hình 2. 7 Sơ đồ tuần tự tìm kiếm tour du lịch, khách sạn.....	20
Hình 2. 8 Sơ đồ tuần tự chức năng đặt tour du lịch, khách sạn.....	21
Hình 2. 9 Sơ đồ tuần tự quản lý người dùng	22
Hình 2. 10 Sơ đồ tuần tự quản lý tour du lịch khách sạn	22
Hình 2. 11 Sơ đồ tuần tự quản lý phòng khách sạn.....	23
Hình 2. 12 Sơ đồ tuần tự reset mật khẩu	23
Hình 2. 13 Sơ đồ tuần tự chức năng đăng xuất .. Error! Bookmark not defined.	
Hình 2. 14 Sơ đồ hoạt động chức năng đăng ký.....	24
Hình 2. 15 Sơ đồ hoạt động chức năng đăng nhập.....	25
Hình 2. 16 Sơ đồ hoạt động chức năng tổng quát của người dùng	26
Hình 2. 17 Sơ đồ hoạt động chức năng đăng nhập.....	27
Hình 2. 18 Sơ đồ hoạt động chức năng đăng ký.....	28
Hình 2. 19 Sơ đồ hoạt động chức năng tìm kiếm và đặt của người dùng	29
Hình 2. 20 Sơ đồ hoạt động chức năng quản lý người dùng.....	30
Hình 2. 21 Sơ đồ hoạt động chức năng quản lý tour du lịch.....	31
Hình 2. 22 Sơ đồ hoạt động chức năng quản lý khách sạn	32

Hình 2. 23 Sơ đồ hoạt động chức năng quản lý phòng	33
Hình 2. 24 Sơ đồ lớp của hệ thống	34
Hình 2. 25 Cơ sở dữ liệu của hệ thống	34
Hình 3. 1 Màn hình đăng nhập	39
Hình 3. 2 Màn hình đăng ký	40
Hình 3. 3 Màn hình quên mật khẩu	40
Hình 3. 5 Màn hình trang chủ.....	41
Hình 3. 6 Màn hình danh sách khách sạn.....	41
Hình 3. 7 Màn hình chi tiết khách sạn.....	42
Hình 3. 8 Màn hình chi tiết chọn ngày khách sạn	42
Hình 3. 9 Màn hình chi tiết phòng khách sạn.....	43
Hình 3. 10 Màn hình thống kê.....	44
Hình 3. 11 Màn hình quản lý người dùng	45
Hình 3. 12 Màn hình thêm người dùng	45
Hình 3. 13 Màn hình chi tiết người dùng	46
Hình 3. 14 Màn hình danh sách tour	46
Hình 3. 15 Màn hình thêm mới tour.....	47
Hình 3. 16 Màn hình danh sách khách sạn	47
Hình 3. 17 Màn hình thêm mới khách sạn	48
Hình 3. 18 Màn hình chi tiết khách sạn.....	48
Hình 3. 19 Màn hình danh sách các loại phòng khách sạn	48

Danh sách bảng

Bảng 2. 1 Bảng người dùng.....	35
Bảng 2. 2 Bảng tài khoản	35
Bảng 2. 3 Bảng tour.....	36
Bảng 2. 4 Bảng khách sạn	36
Bảng 2. 5: Bảng phòng	Error! Bookmark not defined.
Bảng 2. 6: Bảng số phòng.....	37
Bảng 2. 7: Bảng bill	46
Bảng 2. 8: Bảng store	46

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

CHỮ VIẾT TẮT:

STT	Chữ viết tắt	Giải nghĩa
1	SDK	Software develop kit
2	API	Application Programing Interface
3	HTML	Hypertext Markup Language
4	HTTP	Hypertext Transfer Protocol
5	CSS	Cascading Style Sheets
6	MVC	Model-View-Controller

MỞ ĐẦU

1. Bối cảnh đề tài

Trong bối cảnh hiện đại, ngành du lịch đang nổi lên như một trong những lĩnh vực phát triển nhanh nhất trên thế giới. Du lịch và khám phá các điểm đến mới không chỉ mang lại niềm vui và sự thư giãn cho con người, mà còn tạo ra lợi ích kinh tế lớn cho các địa phương và doanh nghiệp. Trong tình hình này, việc phát triển một ứng dụng hỗ trợ đặt khách sạn và tour du lịch trở nên vô cùng quan trọng và cần thiết.

Ngày nay, du khách đang đối mặt với nhiều khó khăn trong việc quản lý và tổ chức các chuyến đi. Quá trình tìm kiếm, so sánh và chọn lựa khách sạn và tour du lịch phù hợp, thực hiện các bước đặt chỗ, quản lý thông tin cá nhân và giao tiếp với các đối tác đòi hỏi nhiều nỗ lực và sự phối hợp chặt chẽ. Để vượt qua những khó khăn này, việc áp dụng công nghệ thông tin và phát triển một ứng dụng hỗ trợ đặt khách sạn và tour du lịch là một giải pháp hiệu quả.

2. Phạm vi đề tài

Sản phẩm được phát triển để sử dụng trên ứng dụng di động

3. Triển khai đề tài

Để triển khai dự án "Xây dựng ứng dụng hỗ trợ đặt khách sạn và tour du lịch", tôi đã áp dụng một quy trình phát triển chặt chẽ và toàn diện. Quá trình triển khai này bao gồm các bước quan trọng như lên kế hoạch, thiết kế, phát triển, kiểm thử và triển khai chính thức. Dưới đây là một phần mô tả về quá trình triển khai dự án này:

- **Lên kế hoạch:** Trước khi bắt đầu phát triển ứng dụng, tôi đã tiến hành phân tích yêu cầu của người dùng và xác định các chức năng cần thiết cho ứng dụng. Điều này đảm bảo rằng ứng dụng có thể đáp ứng được các nhu cầu cụ thể của người sử dụng khi đặt khách sạn và tour du lịch.
- **Thiết kế:** Sau khi xác định yêu cầu, tôi đã tiến hành thiết kế giao diện và cấu trúc của ứng dụng. Đội ngũ đã phát triển một giao diện người dùng thân thiện, dễ sử dụng và tương thích với các thiết bị di động. Đồng thời, tôi đã thiết kế cơ sở dữ liệu để lưu trữ thông tin về khách sạn, tour du lịch, khách hàng và các giao dịch liên quan.
- **Phát triển:** Tiếp theo, tôi đã thực hiện việc phát triển các tính năng và chức năng của ứng dụng. Sử dụng các công nghệ và ngôn ngữ lập trình phù hợp, chúng tôi đã xây dựng hệ thống quản lý đặt phòng khách sạn và tour du lịch hiệu quả. Đảm bảo rằng ứng dụng hoạt động ổn định, có thời gian tải trang nhanh và bảo mật thông tin cá nhân của người dùng.
- **Triển khai chính thức:** Cuối cùng, sau khi hoàn thành phát triển, chúng tôi đã triển khai ứng dụng trên nền tảng điện thoại di động và/hoặc máy tính bảng. Chúng tôi đã cài đặt ứng dụng trên các nền tảng và kiểm tra tính tương thích và khả năng hoạt động trước khi đưa vào sử dụng thực tế.

Từ quá trình triển khai này, tôi hy vọng rằng ứng dụng sẽ giúp người dùng dễ dàng đặt khách sạn và tour du lịch một cách thuận tiện và hiệu quả hơn, mang lại trải nghiệm du lịch tuyệt vời và tiết kiệm thời gian cho họ.

4. Bố cục đồ án

Đồ án bao gồm các nội dung sau:

Mở đầu

Chương 1: Cơ sở lý thuyết và công nghệ áp dụng

Chương 2: Phân tích và thiết kế hệ thống

Chương 3: Triển khai dự án và kết quả

Kết luận và hướng phát triển

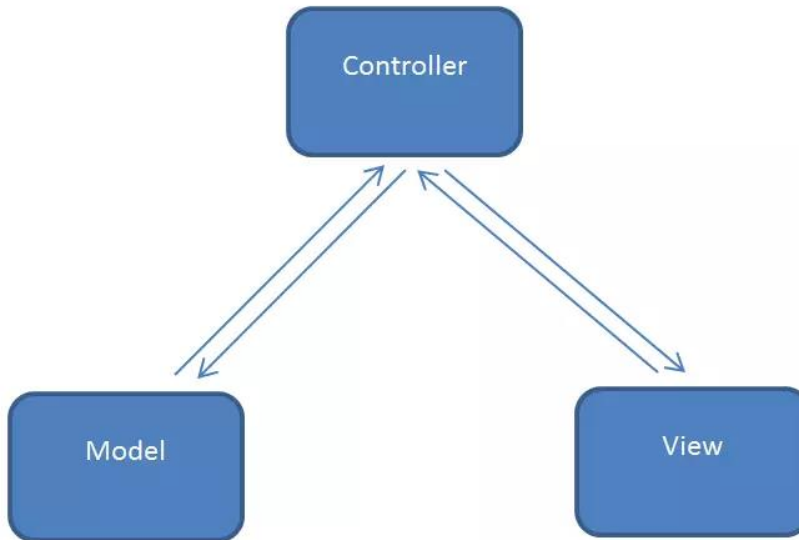
Tài liệu tham khảo

Chương 1: CƠ SỞ LÝ THUYẾT

1. Mô hình MVC

1.1. Các thành phần trong mô hình MVC

Mô hình MVC là một kiến trúc phổ biến được sử dụng để phân chia logic ứng dụng thành các thành phần riêng biệt, giúp tăng tính tổ chức, tái sử dụng và bảo trì của mã nguồn.



Hình 1. 1 Các thành phần trong mô hình MVC

- Model (Mô hình):

Trong mô hình MVC, Model đóng vai trò quản lý và xử lý dữ liệu của ứng dụng. Trong môi trường Node.js và JavaScript, Model có thể là một đối tượng đơn giản hoặc một lớp JavaScript phức tạp. Model chịu trách nhiệm cho các hoạt động như truy vấn, tạo, cập nhật và xóa dữ liệu từ cơ sở dữ liệu. Nó cũng có thể định nghĩa các quy tắc xác thực và hợp lệ cho dữ liệu để đảm bảo tính toàn vẹn và an toàn.

- View (Giao diện):

View trong mô hình MVC là phần giao diện người dùng của ứng dụng. Trong Node.js và JavaScript, View thường là các tệp HTML hoặc các framework giao diện như React, Angular. Nhiệm vụ chính của View là hiển thị dữ liệu từ Model và cung cấp giao diện cho phép người dùng tương tác một cách trực quan và thân thiện. View không xử lý logic ứng dụng hoặc thay đổi dữ liệu, mà chỉ chịu trách nhiệm hiển thị thông tin theo yêu cầu của Controller.

- Controller (Điều khiển):

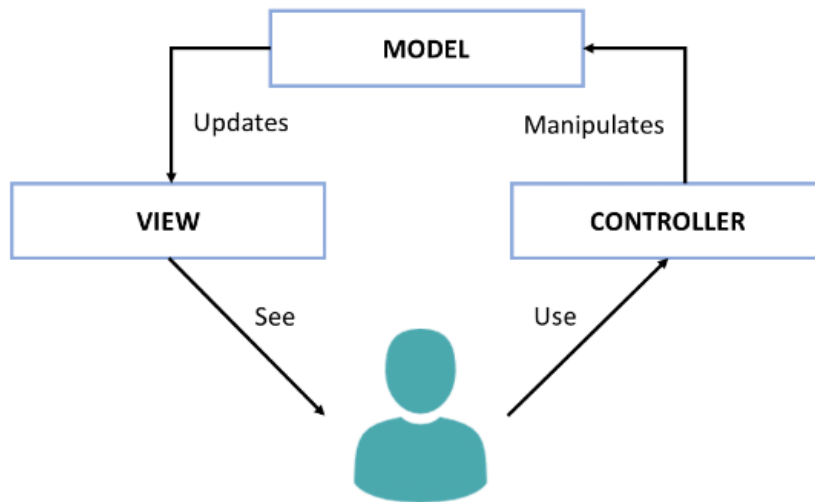
Controller là thành phần trung gian trong mô hình MVC, giữ vai trò điều khiển luồng dữ liệu và logic của ứng dụng. Trong Node.js và JavaScript, Controller là các hàm hoặc lớp JavaScript chịu trách nhiệm xử lý yêu cầu từ người dùng thông qua các route (đường dẫn). Controller có thể truy xuất và cập nhật dữ liệu từ Model và sau đó chuyển dữ liệu đã xử lý đến View để hiển thị kết quả cho người dùng. Điều này giúp phân chia

rõ ràng các công việc trong ứng dụng và tăng tính tái sử dụng mã nguồn, giảm thiểu việc lặp lại và nâng cao khả năng bảo trì của hệ thống.

Sự phân chia rõ ràng của mô hình MVC giúp tăng tính tái sử dụng mã nguồn. Bằng cách tách biệt logic ứng dụng và dữ liệu, chúng ta có thể tái sử dụng Model và Controller trong các phần khác nhau của ứng dụng hoặc trong các dự án khác nhau. Điều này giúp giảm thiểu việc lặp lại và tăng khả năng bảo trì của mã nguồn.

1.2. Nguyên lý hoạt động trong mô hình MVC

Người dùng chọn một Browser bất kỳ: Firefox, Chrome hay Internet Explorer,... để gửi yêu cầu. Bạn có thể gửi những dữ liệu nhập tới các Controller và để Controller xử lý dựa trên bộ Routing điều hướng.



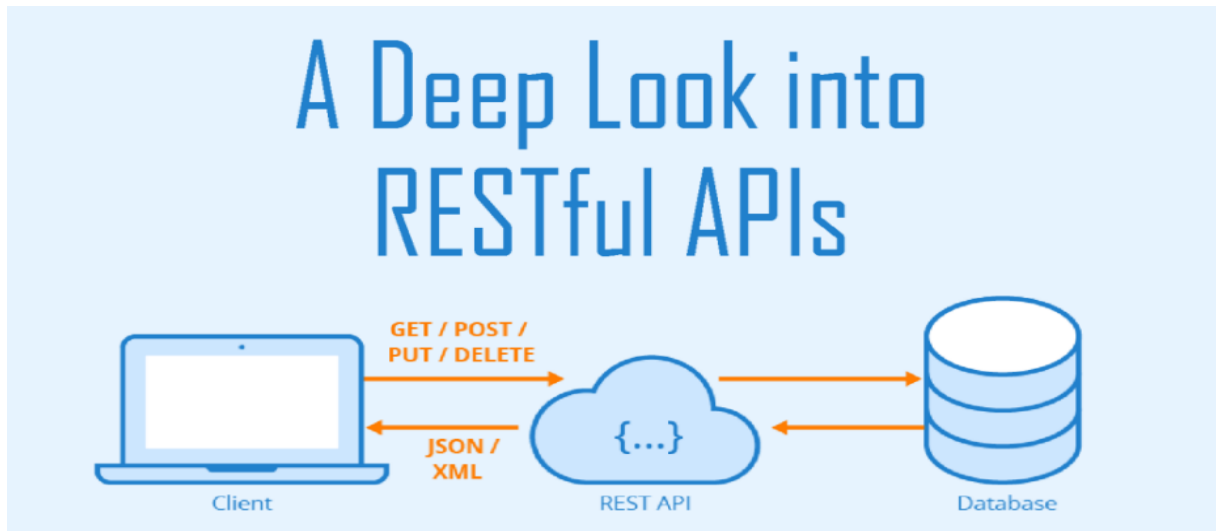
Hình 1. 2 Nguyên lý hoạt động trong mô hình MVC

Khi Controller nhận được yêu cầu từ người dùng, chúng sẽ kiểm tra yêu cầu có cần lấy dữ liệu từ Model hay không. Nếu cần, Controller sẽ dùng các class/ function có sẵn để trả kết quả. Sau khi xử lý các giá trị đó, nó sẽ trả qua View để hiển thị.

Tiếp đến, View chịu trách nhiệm cho phần hiển thị gồm: thông tin dữ liệu, hình ảnh, video,... Rồi trả về GUI Content để Controller đưa kết quả lên Browser. Trình duyệt sẽ tiếp nhận dữ liệu và trả lại kết quả tìm kiếm cho người dùng

2. Tổng quan về RESTful API

2.1. Các thành phần của RESTful API



Hình 1. 3 Tổng quan về RESTful API

RESTful API là một tiêu chuẩn được sử dụng rộng rãi trong thiết kế các API quản lý tài nguyên trong ứng dụng web. Nó sử dụng các phương thức HTTP như GET, POST, PUT và DELETE để quản lý tài nguyên và thực hiện các hoạt động như tạo mới, đọc, cập nhật và xóa dữ liệu. RESTful API không bị ràng buộc bởi ngôn ngữ lập trình hay framework cụ thể, cho phép sử dụng bất kỳ công nghệ nào để triển khai.

RESTful API tuân thủ các nguyên tắc quan trọng sau:

- **Quy chuẩn HTTP:** Sử dụng các phương thức HTTP như GET, POST, PUT, DELETE để đảm bảo tính nhất quán và tương thích khi giao tiếp giữa các ứng dụng.
- **Định dạng dữ liệu:** Thường sử dụng các định dạng dữ liệu như JSON (JavaScript Object Notation) hoặc XML (eXtensible Markup Language) để trao đổi dữ liệu giữa các ứng dụng.
- **Quản lý tài nguyên:** Mỗi tài nguyên được đại diện bằng một URL duy nhất và các phương thức HTTP được áp dụng để thực hiện các hoạt động liên quan đến tài nguyên đó.
- **Stateless (Không lưu trạng thái):** RESTful không lưu trạng thái giữa các yêu cầu từ phía máy khách. Mỗi yêu cầu phải chứa đầy đủ thông tin để xác định và xử lý yêu cầu độc lập.

Những đặc tính này giúp RESTful API đảm bảo tính linh hoạt, dễ bảo trì và mở rộng, phù hợp cho các hệ thống phân tán và cho phép các ứng dụng khác nhau (bao gồm cả ứng dụng web và di động) tương tác với nhau một cách hiệu quả.

2.2. Cách RESTful API hoạt động

REST hoạt động chủ yếu dựa trên giao thức HTTP và các hoạt động cơ bản sau sẽ sử dụng các phương thức HTTP tương ứng:

- **GET (SELECT):** Phương thức này được sử dụng để truy xuất một Resource hoặc một danh sách Resource.

- **POST (CREATE):** Phương thức này được sử dụng để tạo mới một Resource.
- **PUT (UPDATE):** Phương thức này được sử dụng để cập nhật một Resource đã tồn tại.
- **DELETE (DELETE):** Phương thức này được sử dụng để xóa một Resource.

Các hoạt động trên thường được gọi là **CRUD**, viết tắt của Create (Tạo), Read (Đọc), Update (Sửa), và Delete (Xóa). Qua đó, việc áp dụng phương thức CRUD trong REST API giúp chúng ta quản lý và thao tác với các tài nguyên một cách linh hoạt và hiệu quả trên nền tảng web.

2.3. Status code:

REST Khi chúng ta gửi một yêu cầu tới một API, thông thường sẽ nhận được một mã trạng thái (status code) để chỉ ra kết quả của yêu cầu đó. Dưới đây là một số mã trạng thái phổ biến và ý nghĩa tương ứng:

- **200 OK:** Yêu cầu thành công và kết quả đã được trả về. Đây là mã trạng thái chung cho các phương thức GET, PUT, PATCH, hoặc DELETE để báo hiệu rằng tất cả hoạt động đã diễn ra thành công.
- **201 Created:** Tài nguyên đã được tạo thành công. Mã này thông báo rằng yêu cầu đã tạo thành công một tài nguyên mới và trả về thông tin chi tiết về tài nguyên đó.
- **204 No Content:** Yêu cầu thành công nhưng không có dữ liệu để trả về. Thường được sử dụng cho các phương thức DELETE hoặc khi yêu cầu xử lý thành công mà không cần trả về bất kỳ dữ liệu nào.
- **304 Not Modified:** Dữ liệu không thay đổi kể từ lần truy cập gần nhất. Mã này cho biết rằng dữ liệu đã được cache và client có thể sử dụng phiên bản cache mà không cần tải lại từ server.
- **400 Bad Request:** Yêu cầu không hợp lệ do lỗi cú pháp hoặc tham số. Server không thể hiểu yêu cầu do định dạng không chính xác.
- **401 Unauthorized:** Yêu cầu yêu cầu xác thực để tiếp tục. Người dùng phải đăng nhập hoặc cung cấp thông tin xác thực hợp lệ.
- **403 Forbidden:** Server từ chối yêu cầu vì người dùng không có quyền truy cập vào tài nguyên.
- **404 Not Found:** Tài nguyên được yêu cầu không tồn tại trên server.
- **405 Method Not Allowed:** Phương thức yêu cầu không được hỗ trợ hoặc không được phép với tài nguyên hiện tại.
- **410 Gone:** Tài nguyên không còn tồn tại và đã bị xóa hoàn toàn.
- **415 Unsupported Media Type:** Kiểu dữ liệu yêu cầu không được hỗ trợ bởi server.
- **422 Unprocessable Entity:** Yêu cầu không thể xử lý do dữ liệu không hợp lệ.
- **500 Internal Server Error:** Lỗi nội bộ của server khi không thể xử lý yêu cầu.
- **503 Service Unavailable:** Dịch vụ không khả dụng tạm thời, thường do server đang bảo trì hoặc quá tải.

- **504 Gateway Timeout:** Server không nhận được phản hồi từ server cuối cùng trong thời gian chờ quy định.
- **505 HTTP Version Not Supported:** Server không hỗ trợ phiên bản HTTP được sử dụng.
- **507 Insufficient Storage:** Server không có đủ không gian lưu trữ để xử lý yêu cầu.
- **511 Network Authentication Required:** Client cần xác thực với mạng trước khi truy cập tài nguyên được yêu cầu.

3. Khái niệm cơ bản về UML

UML (Unified Modeling Language) là một ngôn ngữ mô hình hóa phổ biến được sử dụng để thiết kế, mô hình hóa và phân tích hệ thống phần mềm. Dưới đây là các khái niệm cơ bản trong UML:

- **Lược đồ UML:** Được sử dụng để biểu diễn các phần tử và mối quan hệ giữa chúng trong hệ thống phần mềm. Các loại lược đồ UML phổ biến bao gồm lược đồ lớp, lược đồ use case, lược đồ hoạt động, lược đồ tuần tự, và nhiều loại khác.
- **Lớp:** Đại diện cho một đối tượng trong hệ thống, bao gồm các thuộc tính và phương thức của đối tượng đó. Lược đồ lớp được sử dụng để biểu diễn các lớp và mối quan hệ giữa chúng.
- **Use case:** Biểu thị một tác vụ hoặc chức năng cụ thể mà hệ thống cung cấp cho người dùng. Lược đồ use case được sử dụng để mô tả các use case và mối quan hệ giữa các use case và các tác nhân (actors).
- **Biểu đồ tuần tự:** Mô tả sự tương tác giữa các đối tượng theo thứ tự thời gian. Biểu đồ tuần tự thể hiện các thông điệp được gửi giữa các đối tượng trong một tương tác cụ thể.
- **Biểu đồ hoạt động:** Biểu diễn quá trình hoạt động của hệ thống hoặc một phần của nó dưới dạng các hoạt động và mối quan hệ giữa chúng. Lược đồ hoạt động thể hiện luồng công việc và điều kiện chuyển đổi giữa các hoạt động.
- **Biểu đồ trạng thái:** Biểu diễn các trạng thái của một đối tượng hoặc hệ thống và các sự kiện và hành động mà đối tượng hoặc hệ thống đó có thể trải qua. Lược đồ trạng thái sử dụng để mô tả các trạng thái, sự kiện và hành động.
- **Package:** Đại diện cho một nhóm các phần tử liên quan trong UML. Package được sử dụng để tổ chức và quản lý các phần tử UML trong một hệ thống lớn.

Tóm lại, các khái niệm cơ bản trong UML như Biểu đồ Use Case, Biểu đồ Tuần tự, Biểu đồ Cộng tác, Biểu đồ Hoạt động, Biểu đồ Thành phần, Biểu đồ Triển khai và Biểu đồ Lớp cung cấp các phương tiện để mô hình hóa, mô tả và phân tích các khía cạnh khác nhau của một hệ thống phần mềm.

4. Tổng quan về NodeJS

4.1. Giới thiệu

Node.js là một môi trường chạy mã JavaScript ngoài trình duyệt, được xây dựng trên JavaScriptEngine V8 của Google Chrome. Node.js cung cấp một bộ thư viện mạnh mẽ và hệ sinh thái phong phú, cho phép người phát triển xây dựng các ứng dụng web

thời gian thực, ứng dụng mạng, API và hơn thế nữa. Các module của Node.js cho phép tái sử dụng mã, giảm thiểu công sức phát triển và tăng tốc độ triển khai. Bên cạnh đó, Node.js cũng được tích hợp với npm (Node Package Manager), một công cụ quản lý gói phổ biến và mạnh mẽ cho phép người phát triển tải xuống, cài đặt và quản lý các module và thư viện cần thiết cho dự án. Tổng kết lại, Node.js đã đóng vai trò quan trọng trong việc mở rộng khả năng sử dụng của JavaScript từ phía máy chủ, cho phép xây dựng ứng dụng web mạnh mẽ và hiệu quả



Hình 1. 4 Tổng quan về NodeJS

4.2. Ưu điểm

Node.js có nhiều ưu điểm quan trọng, làm cho nó trở thành một lựa chọn hấp dẫn cho việc phát triển ứng dụng web. Dưới đây là một số điểm nổi bật của Node.js:

- **Hiệu suất cao và tối ưu hóa tài nguyên:** Node.js được xây dựng với mô hình non-blocking và sử dụng event loop, giúp tối ưu hóa sử dụng tài nguyên hệ thống và đạt được hiệu suất cao trong xử lý các yêu cầu đồng thời.
- **Kiến trúc không đồng bộ (Asynchronous):** Node.js hỗ trợ các thư viện và APIs không đồng bộ, cho phép xử lý đa nhiệm và nhiều yêu cầu cùng một lúc mà không gây block thread, từ đó tăng tính thích ứng và đáp ứng của ứng dụng.
- **Hệ sinh thái phong phú với npm:** Node.js đi kèm với npm (Node Package Manager), một kho lưu trữ lớn các module và thư viện có thể tái sử dụng, giúp cho việc phát triển và quản lý dependencies trở nên dễ dàng và nhanh chóng.
- **Phát triển đơn giản với JavaScript:** Sử dụng JavaScript trên cả client và server giúp cho quá trình phát triển đơn giản hơn, đồng thời cũng giảm thiểu sự phân tách giữa các nhóm phát triển.
- **Khả năng mở rộng và xử lý quy mô lớn:** Node.js được thiết kế để xử lý các ứng dụng có thể mở rộng một cách linh hoạt, từ các ứng dụng nhỏ đến các hệ thống có quy mô lớn và có yêu cầu cao về độ phản hồi.
- **Hỗ trợ trên nhiều nền tảng:** Node.js hỗ trợ chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux, cung cấp tính linh hoạt cho việc triển khai và vận hành ứng dụng.

Tóm lại, Node.js là một công nghệ mạnh mẽ với các ưu điểm như hiệu suất cao, hỗ trợ kiến trúc không đồng bộ, sự phong phú của hệ sinh thái npm, dễ dàng phát triển với JavaScript, khả năng mở rộng và đa nền tảng. Những ưu điểm này cùng nhau làm cho Node.js trở thành một lựa chọn hàng đầu cho việc phát triển các ứng dụng web hiệu quả và linh hoạt.

4.3. Nhược điểm

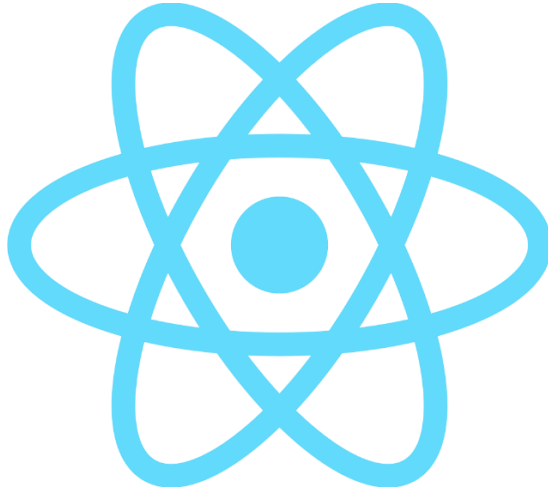
Mặc dù Node.js là một nền tảng phát triển mạnh mẽ và được sử dụng rộng rãi, nhưng nó cũng có một số nhược điểm cần được lưu ý

- Xử lý đơn luồng có thể gây trì hoãn và giảm hiệu suất.
- Không phù hợp cho các ứng dụng yêu cầu tính toán mạnh.
- Quản lý mã nguồn phức tạp, có thể dẫn đến callback hell hoặc Promise chaining dài.
- Phụ thuộc vào bên thứ ba, tăng rủi ro bảo mật và sự ổn định.
- Điều chỉnh tài nguyên và quản lý luồng là thách thức.

5. Tổng quan về React Native

5.1. Giới thiệu

React là một thư viện của Javascript được phát triển bởi Facebook theo chuẩn SPA (Single Page Application) được ra mắt vào năm 2013. React mang đến giải pháp hữu hiệu, đơn giản để xây dựng một giao diện người dùng với tốc độ phản hồi tuyệt vời cùng với phương pháp render nhanh chóng



Hình 1. 5 Tổng quan về React Native

React Native là một framework phát triển ứng dụng di động cross-platform, cho phép sử dụng JavaScript và React để xây dựng các ứng dụng cho cả iOS và Android từ một mã nguồn duy nhất. Nó kết hợp tính linh hoạt của React với khả năng sử dụng các thành phần native, mang lại hiệu suất gần như native và tiện ích trong việc tái sử dụng code. Điều này làm cho React Native trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng di động hiện đại và hiệu quả.

React cho phép lập trình viên phân tách các giao diện phức tạp thành nhiều thành phần nhỏ lẻ, đơn giản, giúp cho việc phát triển, bảo trì và tái sử dụng được hiệu quả hơn

5.2. Ưu điểm

React Native là một lựa chọn phù hợp để phát triển sản phẩm này với những ưu điểm sau:

- **Phát triển ứng dụng di động đa nền tảng:** React Native cho phép phát triển ứng dụng cho cả iOS và Android từ một mã nguồn duy nhất, giúp tiết kiệm thời gian và chi phí phát triển.
- **Sử dụng JavaScript và React:** Được xây dựng trên cơ sở của JavaScript và React, React Native giúp những lập trình viên quen thuộc với React có thể dễ dàng chuyển sang phát triển ứng dụng di động.
- **Hiệu suất gần như native:** React Native sử dụng các thành phần native để đạt hiệu suất gần như ứng dụng native trên iOS và Android.
- **Cộng đồng hỗ trợ mạnh mẽ:** React Native có một cộng đồng lớn và sôi động, cùng với các công cụ phát triển và thư viện hỗ trợ đa dạng, giúp cho việc phát triển và bảo trì ứng dụng trở nên dễ dàng hơn.
- **Hot Reloading và Live Reload:** Tính năng Hot Reloading và Live Reload của React Native giúp phát triển viên thực hiện thay đổi và xem kết quả ngay lập tức trên thiết bị hoặc m

5.3. Nhược điểm

React Native có một số nhược điểm cần được lưu ý:

- **Hiệu suất không đồng đều:** Mặc dù React Native hứa hẹn hiệu suất gần như native, nhưng trong một số trường hợp ứng dụng có thể gặp phải vấn đề hiệu suất không đồng đều, đặc biệt là trong các ứng dụng yêu cầu xử lý đồ họa phức tạp hoặc đòi hỏi tốc độ xử lý cao.
- **Phụ thuộc vào các thư viện và bản cập nhật:** Khi sử dụng React Native, phụ thuộc vào các thư viện của bên thứ ba và các bản cập nhật của React Native có thể dẫn đến sự không ổn định và khó khăn trong việc duy trì ứng dụng.
- **Độ phức tạp của lập trình native:** Mặc dù React Native giúp tái sử dụng mã nguồn, nhưng đôi khi vẫn cần phải viết mã native để xử lý các vấn đề đặc thù của từng nền tảng (iOS và Android), điều này có thể làm tăng độ phức tạp của dự án.
- **Kích thước ứng dụng:** Ứng dụng React Native có thể có kích thước lớn hơn so với ứng dụng native tương đương, do việc bao gồm các thư viện và mã JS mà không cần thiết.
- **Vấn đề tương thích và debug:** Do sự khác biệt giữa React Native và các nền tảng native, bạn có thể gặp phải vấn đề về tương thích và gỡ lỗi khó khăn, đặc biệt là khi cần phải thao tác trực tiếp với mã native.

Tóm lại, mặc dù React Native là một công nghệ mạnh mẽ cho phát triển ứng dụng di động đa nền tảng, nhưng cũng có những nhược điểm cần xem xét và giải quyết để đảm bảo hiệu quả và ổn định trong quá trình phát triển và triển khai ứng dụng.

6. Tổng quan về MongoDB

6.1. Giới thiệu

MongoDB là một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) phổ biến được phát triển bởi MongoDB Inc. Nó được thiết kế để xử lý và lưu trữ dữ liệu phi cấu trúc và có khả năng mở rộng tốt. MongoDB sử dụng mô hình lưu trữ dựa trên tài liệu (document), trong đó dữ liệu được lưu trữ dưới dạng các bản ghi JSON-like có tên là "document".

Một trong những ưu điểm quan trọng của MongoDB là khả năng linh hoạt trong cấu trúc dữ liệu. Điều này giúp giải quyết các vấn đề liên quan đến sự thay đổi yêu cầu dữ liệu trong quá trình phát triển phần mềm.

MongoDB cung cấp ngôn ngữ truy vấn mạnh mẽ và linh hoạt (MongoDB Query Language - MQL), cho phép truy vấn dữ liệu theo các tiêu chí phù hợp với nhu cầu của ứng dụng. Điều này giúp đơn giản hóa việc truy xuất và xử lý dữ liệu trong MongoDB.

Ngoài ra, MongoDB có một hệ sinh thái phong phú với các công cụ và thư viện hỗ trợ phát triển ứng dụng. Điều này giúp người phát triển tận dụng tối đa tiềm năng của MongoDB và tăng cường năng suất trong quá trình phát triển.



Hình 1. 6 Tổng quan về MongoDB

Tổng quan, MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mạnh mẽ, linh hoạt và có khả năng mở rộng, phù hợp cho việc lưu trữ và xử lý dữ liệu phi cấu trúc hoặc có yêu cầu độ linh hoạt cao

6.2. Lợi ích khi sử dụng MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) được sử dụng rộng rãi trong các dự án phát triển phần mềm. Có nhiều lợi ích khi sử dụng MongoDB, bao gồm:

- **Linh hoạt và mở rộng:** MongoDB cho phép lưu trữ dữ liệu phi cấu trúc và linh hoạt trong cấu trúc dữ liệu. Điều này giúp đáp ứng các yêu cầu dữ liệu đa dạng và thay đổi trong quá trình phát triển.
- **Hiệu suất cao:** MongoDB được thiết kế để xử lý lượng dữ liệu lớn và tải cao một cách hiệu quả. Với khả năng mở rộng ngang và khả năng phân tán dữ liệu, MongoDB đảm bảo hiệu suất cao và khả năng chịu tải tốt.
- **Truy vấn linh hoạt:** MongoDB cung cấp ngôn ngữ truy vấn mạnh mẽ và linh hoạt, giúp người dùng truy xuất dữ liệu theo các tiêu chí phù hợp với nhu cầu của ứng dụng. Điều này giúp đơn giản hóa việc truy xuất và xử lý dữ liệu.
- **Tính sẵn sàng và độ tin cậy:** MongoDB hỗ trợ replica set, cho phép sao chép dữ liệu và đảm bảo tính sẵn sàng và độ tin cậy của hệ thống. Điều này đảm bảo rằng dữ liệu luôn sẵn sàng và không bị mất mát.
- **Hỗ trợ cộng đồng và tài liệu phong phú:** MongoDB có một cộng đồng lớn và tích cực, cung cấp hỗ trợ và tài liệu phong phú. Người dùng có thể tìm thấy tài liệu, ví dụ mã nguồn mở và câu trả lời cho các câu hỏi của mình thông qua các diễn đàn và tài liệu trực tuyến.

MongoDB là một lựa chọn phù hợp cho việc lưu trữ và xử lý dữ liệu trong các dự án phát triển phần mềm. Với linh hoạt, hiệu suất cao, truy vấn linh hoạt, tính sẵn sàng và độ tin cậy, cùng với sự hỗ trợ từ cộng đồng và tài liệu phong phú, MongoDB mang lại nhiều lợi ích quan trọng cho việc.

6.3. Yếu tố nổi bật

MongoDB là một hệ quản trị cơ sở dữ liệu phi quan hệ với nhiều yếu tố nổi bật đáng chú ý. Đầu tiên, MongoDB sử dụng mô hình dữ liệu linh hoạt, cho phép lưu trữ dữ liệu có cấu trúc và phi cấu trúc một cách dễ dàng. Điều này giúp các nhà phát triển linh hoạt trong việc làm việc với dữ liệu và thích ứng với các thay đổi trong yêu cầu ứng dụng.

Tiếp theo, MongoDB có tính nhất quán và khả năng mở rộng cao. Với khả năng triển khai Replica Set, dữ liệu có thể được sao chép trên nhiều máy chủ, đảm bảo tính nhất quán và khả năng sẵn sàng cao. Đồng thời, MongoDB hỗ trợ Sharding để phân tán dữ liệu trên nhiều mảnh, cho phép mở rộng quy mô và xử lý tải cao một cách hiệu quả.

Một ưu điểm quan trọng khác của MongoDB là truy vấn linh hoạt. MongoDB Query Language (MQL) cung cấp một tập hợp các truy vấn mạnh mẽ và linh hoạt, từ truy vấn đơn giản đến truy vấn phức tạp. MQL hỗ trợ các tiêu chí truy vấn, tìm kiếm văn bản, phân trang và nhiều tính năng khác, giúp người dùng truy xuất và xử lý dữ liệu một cách hiệu quả.

Cuối cùng, MongoDB được tối ưu cho việc làm việc với dữ liệu lớn và tải cao. Với mô hình phân tán, các tính năng như indexing, replication và sharding, MongoDB đáp ứng tốt trong việc xử lý dữ liệu lớn và tải cao, đảm bảo hiệu suất và khả năng mở rộng.

Tóm lại, MongoDB có các yếu tố nổi bật gồm mô hình dữ liệu linh hoạt, tính nhất quán và khả năng mở rộng cao, truy vấn linh hoạt và hỗ trợ cho dữ liệu lớn và tải cao. Đây là những đặc điểm quan trọng đã làm cho MongoDB trở thành một lựa chọn phổ biến cho các ứng dụng có yêu cầu lưu trữ và xử lý dữ liệu phi quan hệ.

7. Các công cụ khác

7.1. Cloundinary

Cloudinary là một công cụ quản lý và tối ưu hóa hình ảnh, video và nội dung đa phương tiện trong các ứng dụng web và di động. Với Cloudinary, người dùng có thể dễ dàng tải lên, lưu trữ và xử lý hình ảnh và video một cách nhanh chóng và hiệu quả.

Một trong những ưu điểm của Cloudinary là khả năng tích hợp dễ dàng với các ngôn ngữ lập trình và framework phổ biến như JavaScript, PHP, Ruby, Python và nhiều hơn nữa. Người dùng có thể truy cập và quản lý các tài nguyên đa phương tiện của họ thông qua API đơn giản và dễ sử dụng của Cloudinary.

Cloudinary cũng cung cấp các tính năng quản lý tài nguyên mạnh mẽ, bao gồm quản lý phiên bản, quản lý đám mây và quản lý tải lên. Điều này giúp người dùng kiểm soát và theo dõi các tài nguyên đa phương tiện của họ một cách hiệu quả và tiết kiệm thời gian.



Hình 1. 7 Cloundinary

8. Kết chương

Trong chương 1, tôi đã giới thiệu tổng quan về các công nghệ được sử dụng trong đồ án lần này, bao gồm mô hình MVC kết hợp NodeJS để phát triển phần back-end, ReactJS để phát triển giao diện website, MongoDB để quản lý cơ sở dữ liệu, cùng với bộ công cụ phát triển Cloundary và nhiều công nghệ khác. Qua quá trình nghiên cứu, tìm hiểu và học tập cách sử dụng các công nghệ này, tôi đã thành công áp dụng chúng vào đồ án, tạo ra một ứng dụng hoàn chỉnh và chuyên nghiệp.

Chương này cung cấp những kiến thức cơ bản về các công nghệ để phục vụ cho quá trình tìm hiểu cách thức hoạt động của chúng và cách tôi áp dụng chúng trong quá trình phát triển sản phẩm. Những kiến thức này sẽ được đề cập chi tiết hơn trong chương tiếp theo, giúp tôi đi sâu vào tìm hiểu và ứng dụng chúng một cách hiệu quả trong việc phát triển sản phẩm.

Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

1. Phân tích chức năng và yêu cầu

1.1. Chức năng cơ bản

- Đăng ký tài khoản: Ứng dụng yêu cầu người dùng phải có một tài khoản để sử dụng. Thao tác đăng ký rất nhanh với một số ít thông tin cơ bản như: email, tên.
- Xác thực tài khoản: Sau khi đăng ký, người dùng cần xác thực tài. Đường dẫn xác thực được gửi về email được dùng khi đăng ký. Người dùng nhấn vào đường dẫn, khi xác thực thành công hệ thống sẽ hiển thị thông báo thành công.
- Đăng nhập – đăng xuất: Sử dụng tài khoản đã đăng ký để đăng nhập và trải nghiệm ứng dụng.
- Khôi phục mật khẩu đã quên: Đối với những tài khoản mà người dùng quên mật khẩu, người dùng có thể khôi phục bằng cách nhấn 'Quên mật khẩu', hệ thống sẽ gửi mật khẩu cũ thông qua email mà người dùng đã đăng ký từ trước.
- Thay đổi mật khẩu: Người dùng còn có thể thay đổi mật khẩu cho tài khoản của mình.
- Thay đổi thông tin tài khoản cá nhân: Người dùng có thể thay các thông tin cá nhân của tài khoản kể cả ảnh đại diện

1.2. Tác nhân

Đối với Access user:

- Đăng ký tài khoản của chính mình.
- Xem thông tin của khách sạn và các địa điểm tour du lịch, tin tức tour du lịch.
- Tìm kiếm khách sạn và tour du lịch
- Xem thông tin địa điểm khách sạn, tin tức của khách sạn đó, Ngoài ra, còn có thể xem giá cả, các tiện nghi của phòng trong khách sạn đó.

Đối với user

- Đăng ký tài khoản.
- Đăng nhập hệ thống.
- Quên mật khẩu, reset mật khẩu, thay đổi mật khẩu
- Thay đổi thông tin cá nhân.
- Xem thông tin khách sạn và địa điểm tour du lịch, xem tin tức tour du lịch.
- Xem các khách sạn và tour du lịch mình đã lưu trữ.
- Xem thông tin khách sạn, giá cả và loại phòng của khách sạn đó.
- Tìm kiếm địa điểm tour du lịch
- Đặt tour du lịch, có thể đặt thêm các dịch vụ riêng khác
- Xem thông tin địa điểm khách sạn, tin tức của khách sạn đó, Ngoài ra, còn có thể xem giá cả, các tiện nghi của phòng trong khách sạn đó.
- Thanh toán, đặt

Đối với admin

- Quản lý người dùng (thêm, sửa, xóa)
- Quản lý tour du lịch (thêm, sửa, xóa)
- Cập nhật hình ảnh cho tour du lịch

- Quản lý khách sạn (thêm, sửa, xóa)
- Cập nhật hình ảnh cho khách sạn
- Reset mật khẩu cho người dùng
- Quản lý phòng khách sạn (thêm, sửa, xóa)

1.3. Yêu cầu

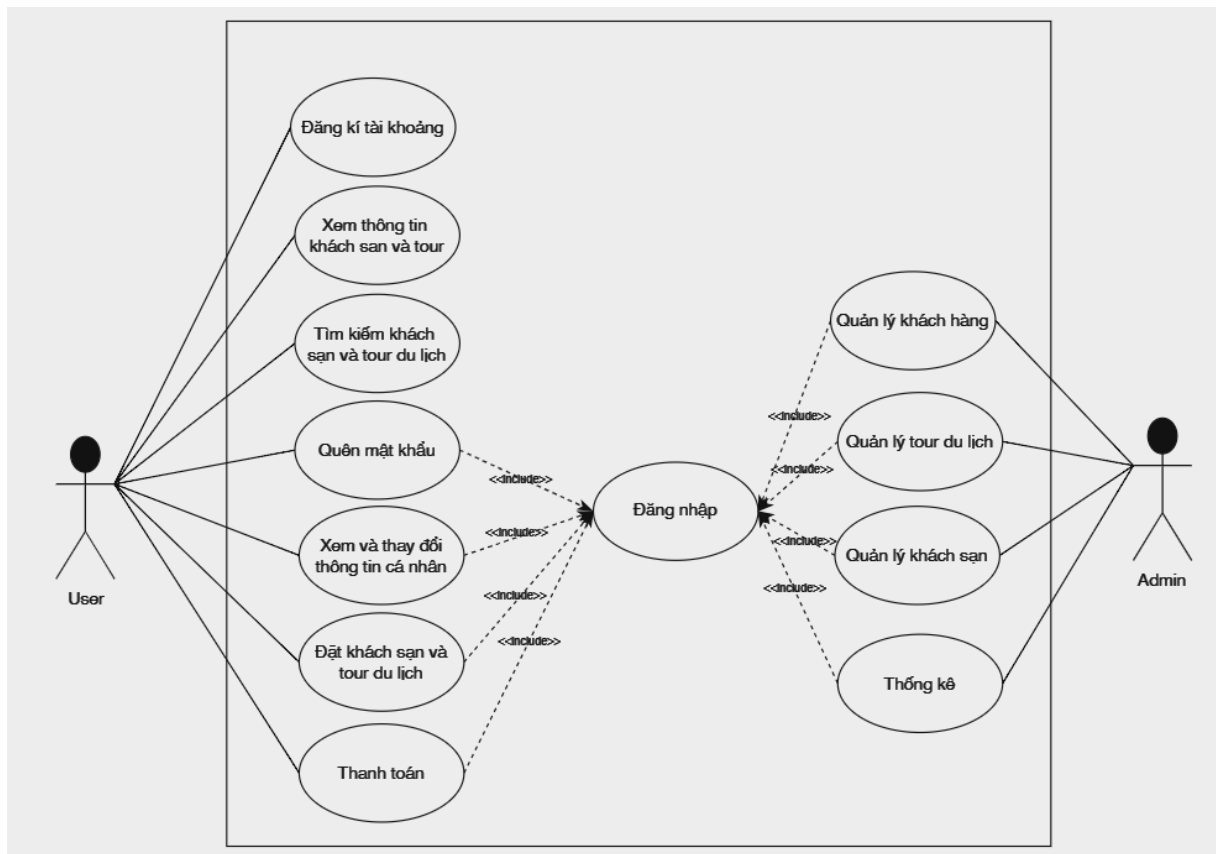
Ứng dụng cần đạt các yêu cầu sau:

- Trải nghiệm tốt trên ứng dụng.
- Khả dụng trên mọi nền tảng di động
- Vì liên quan đến hình ảnh, video cho nên yêu cầu giao diện người dùng phải đẹp, bắt mắt, theo một chủ đề rõ ràng, dễ dàng sử dụng.
- Ưu tiên trải nghiệm dành cho người dùng: tải ảnh, file hình ảnh, âm thanh nhanh, các tác vụ chuyển giao mượt mà, không giật lag.

2. Sơ đồ

2.1. Sơ đồ usecase

- Tổng quan

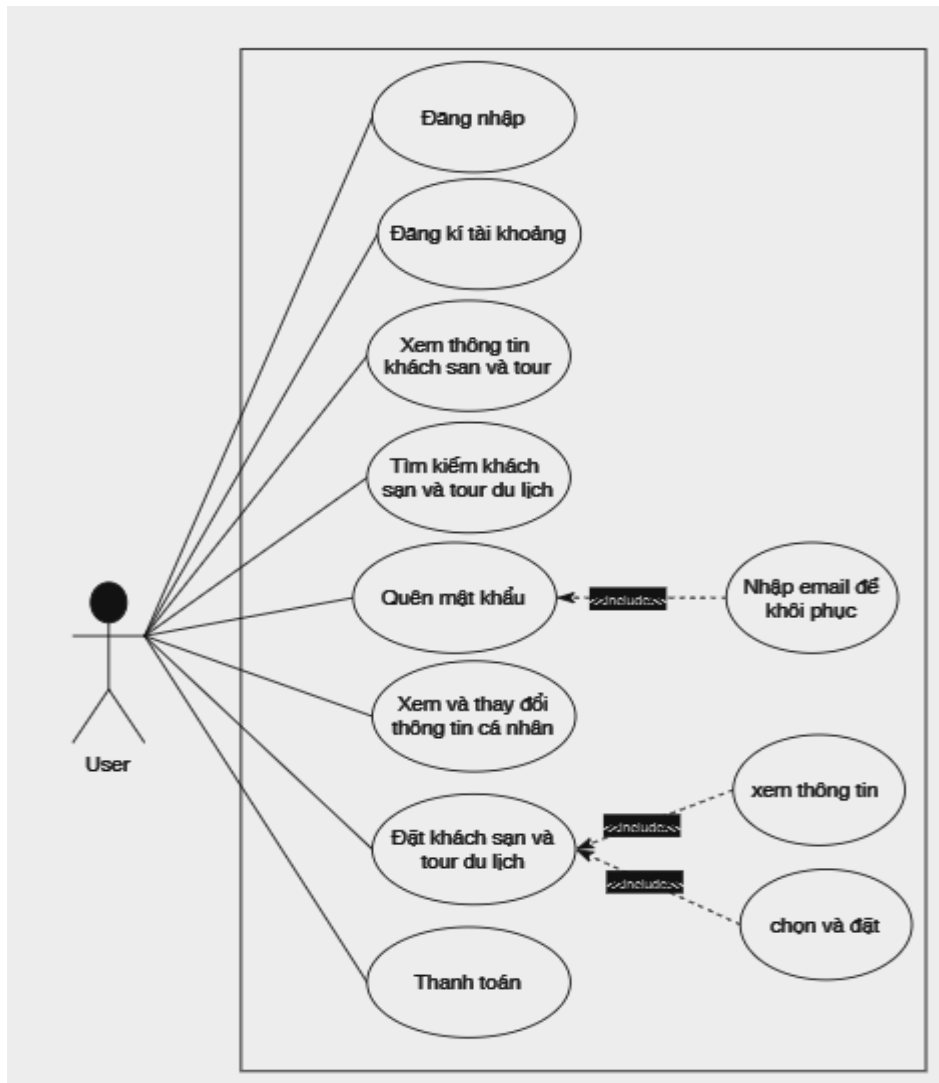


Hình 2. 1 Sơ đồ usecase tổng quát của hệ thống

• **Người dùng**

Tổng quát các chức năng của người dùng:

- Đăng nhập
- Đăng ký
- Quên mật khẩu
- Thay đổi mật khẩu
- Xem và chỉnh sửa thông tin cá nhân
- Xem các tour và khách sạn đã lưu
- Đặt tour
- Đặt phòng khách sạn

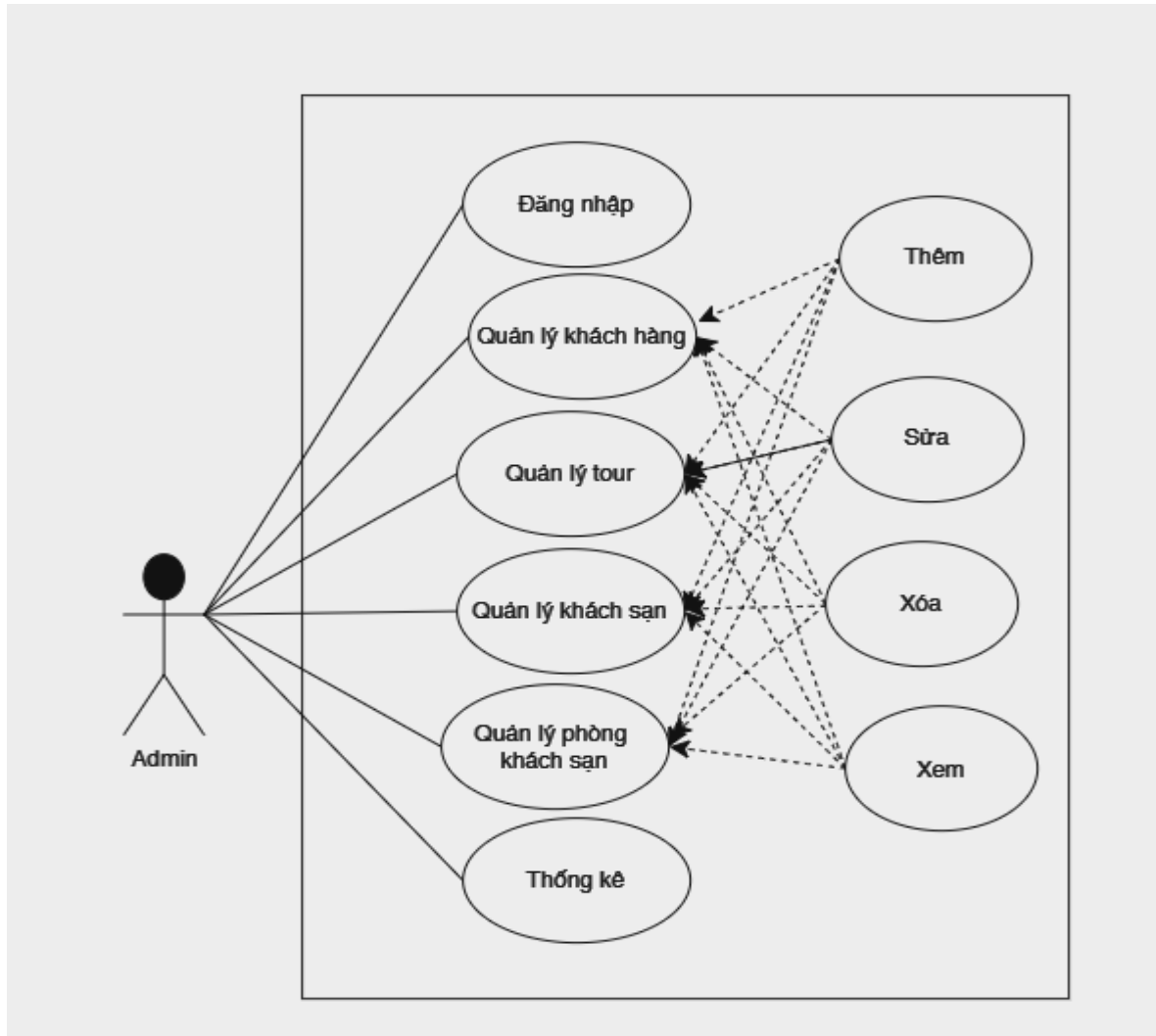


Hình 2. 2 Sơ đồ usecase của người dùng

- **Người quản lý**

Tổng quát các chức năng của người quản lý:

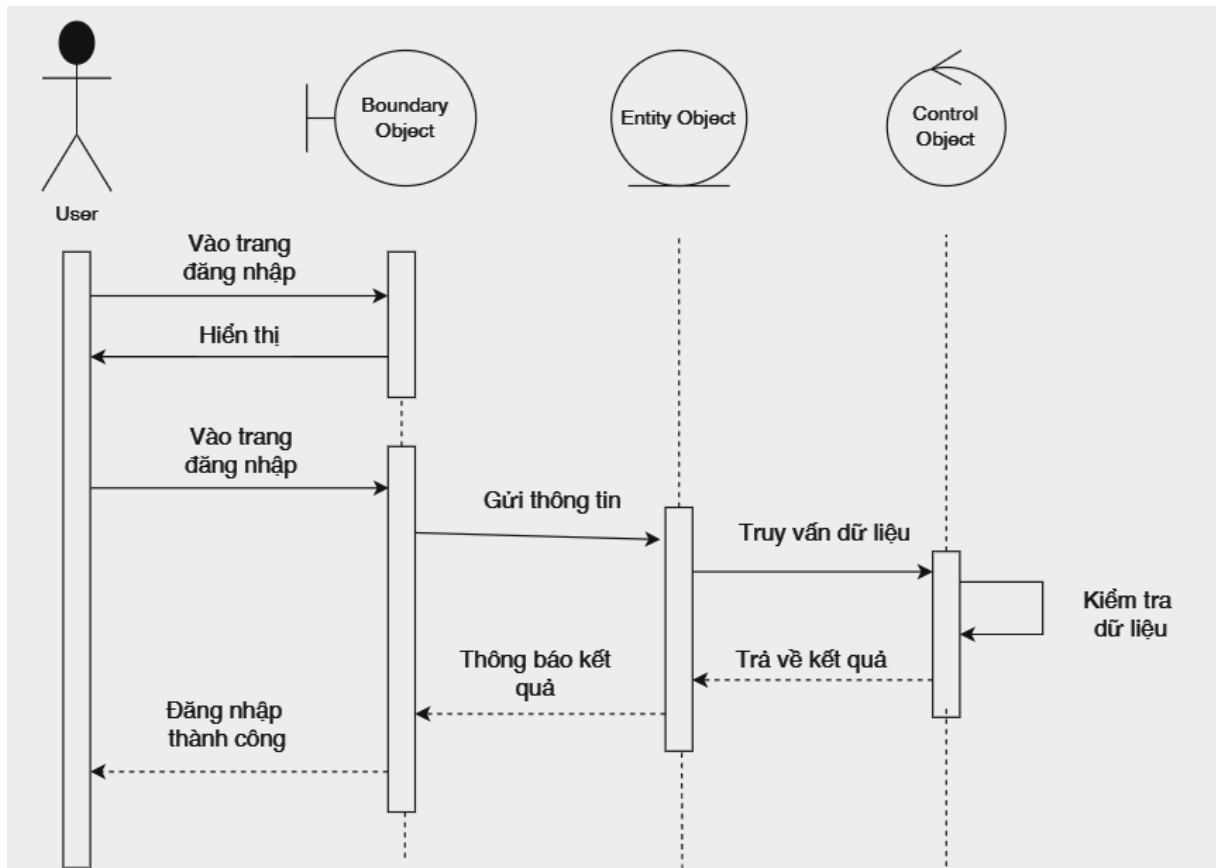
- Đăng nhập
- Thống kê
- Quản lý tài khoản người dùng
- Quản lý tour du lịch
- Quản lý khách sạn
- Quản lý phòng



Hình 2. 3 Sơ đồ usecase của người quản lý

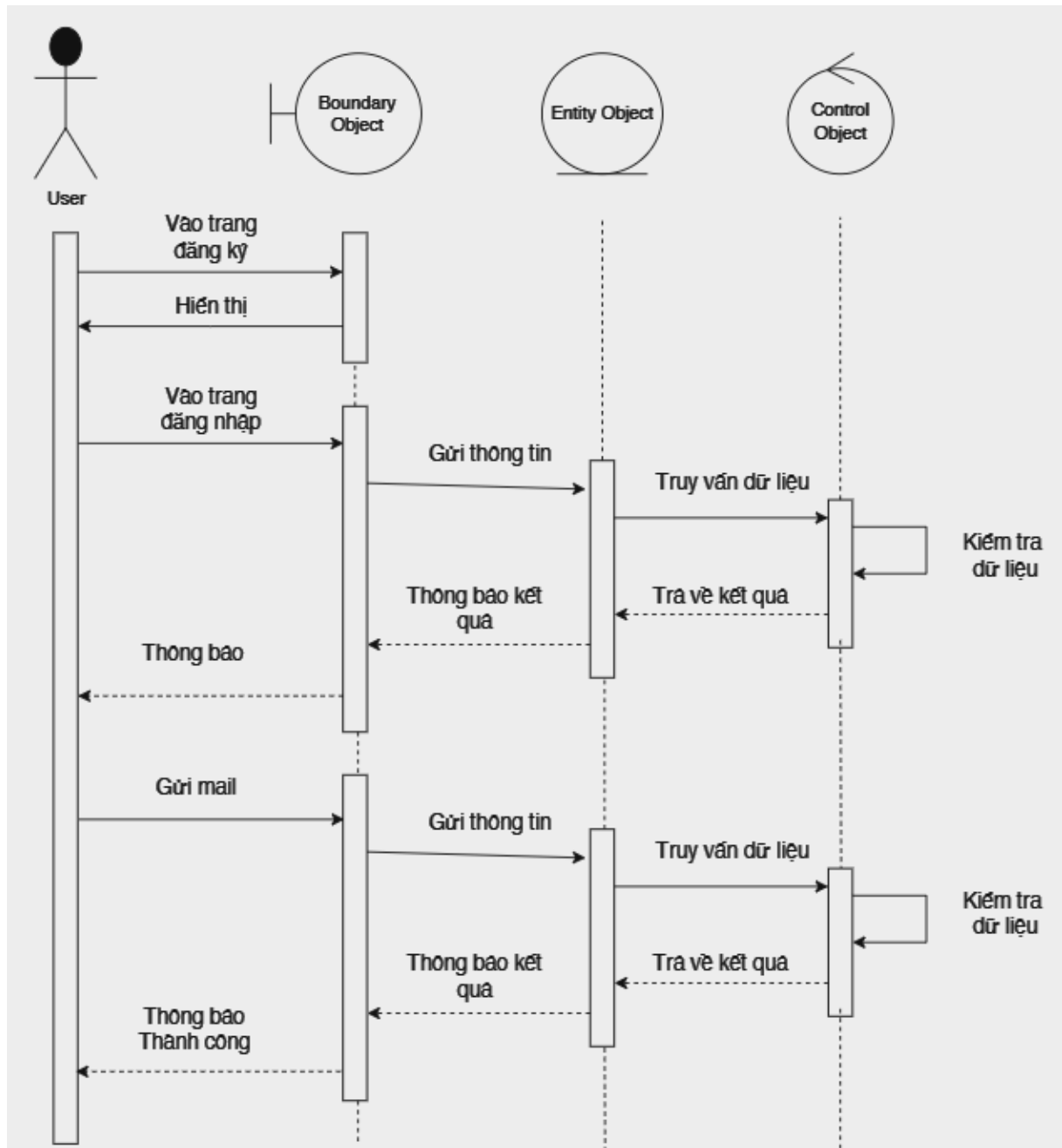
2.2. Sơ đồ tuần tự

Đăng nhập



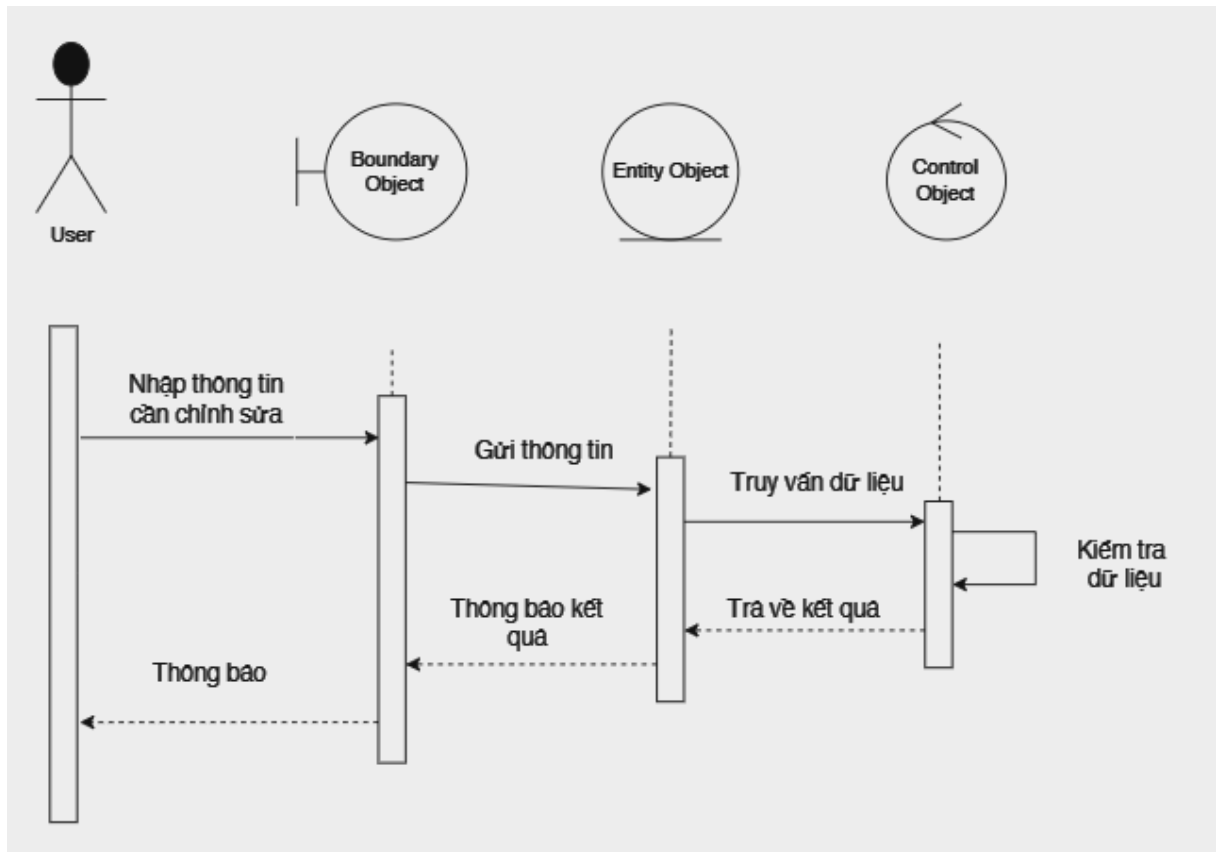
Hình 2. 4 Sơ đồ tuần tự đăng nhập hệ thống

Đăng ký



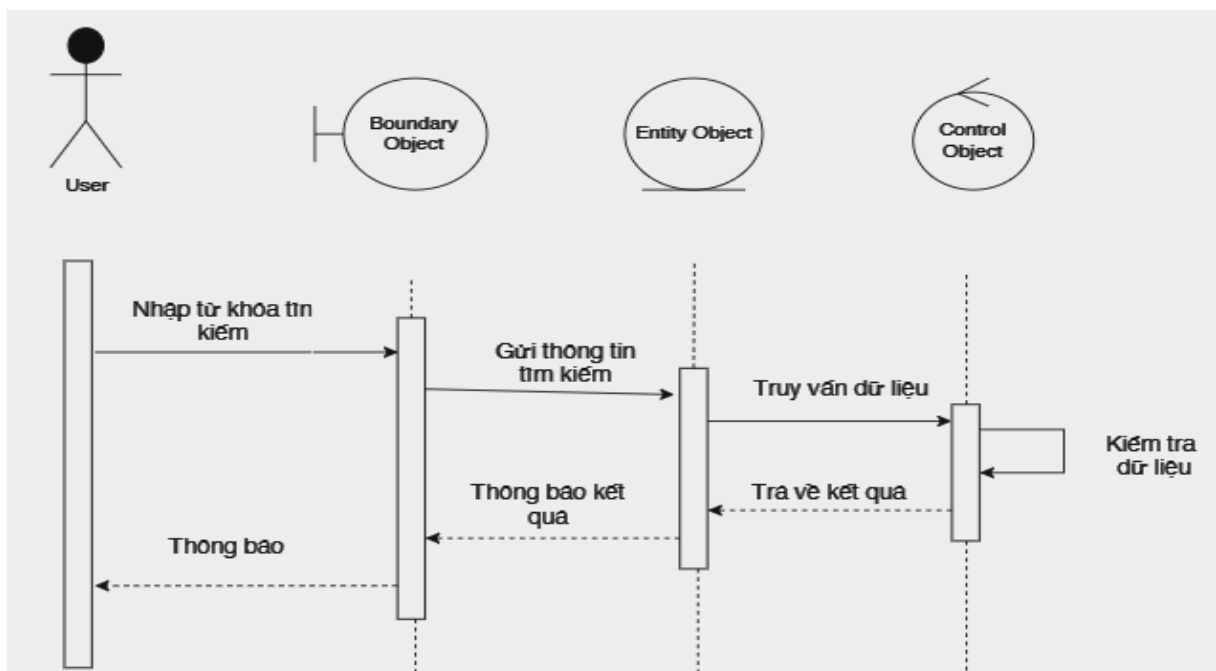
Hình 2. 5 Sơ đồ tuần tự đăng ký tài khoản

Chỉnh sửa thông tin cá nhân



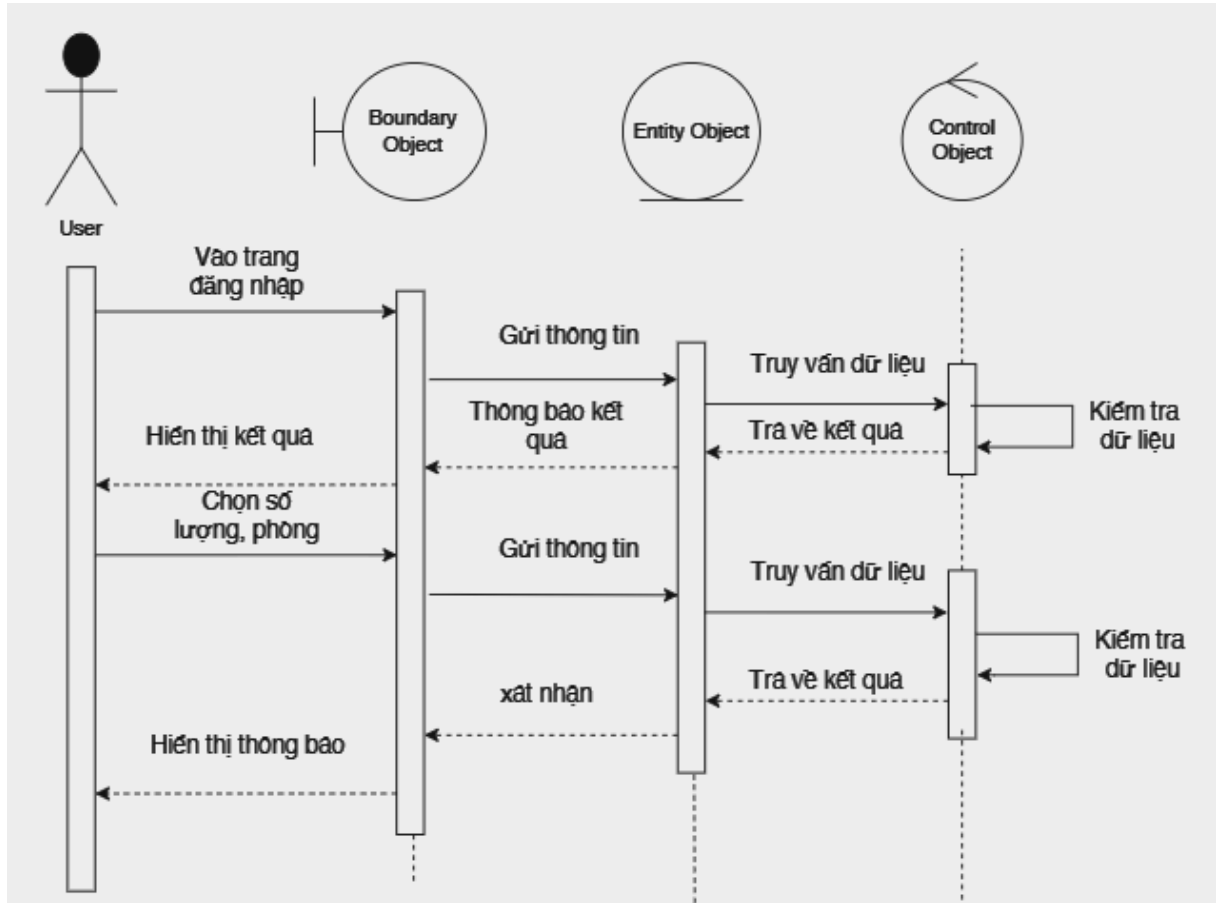
Hình 2. 6 Sơ đồ tuần tự tìm kiếm chỉnh sửa thông tin cá nhân

Tìm kiếm tour du lịch, khách sạn



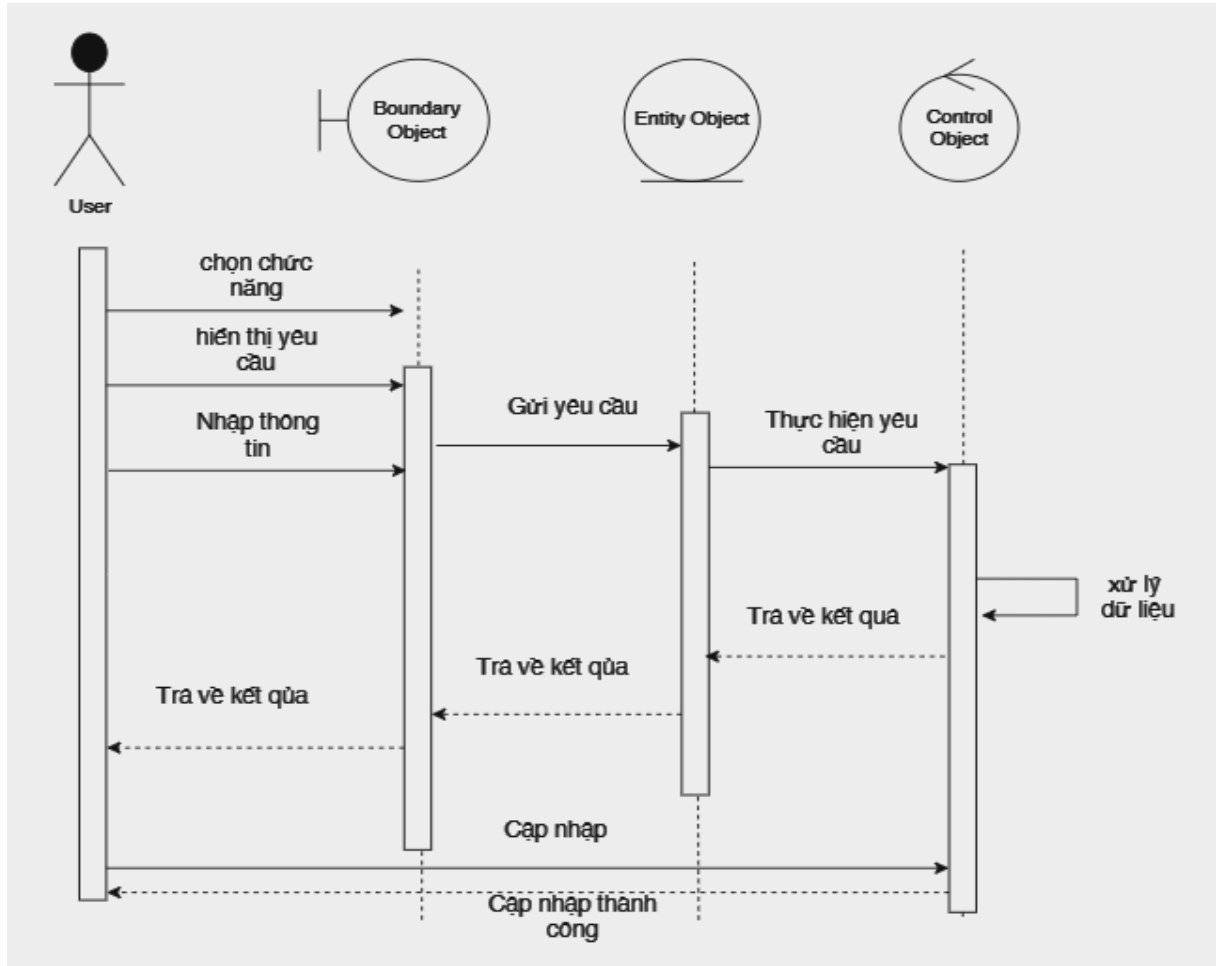
Hình 2. 7 Sơ đồ tuần tự tìm kiếm tour du lịch, khách sạn

Đặt tour du lịch, khách sạn



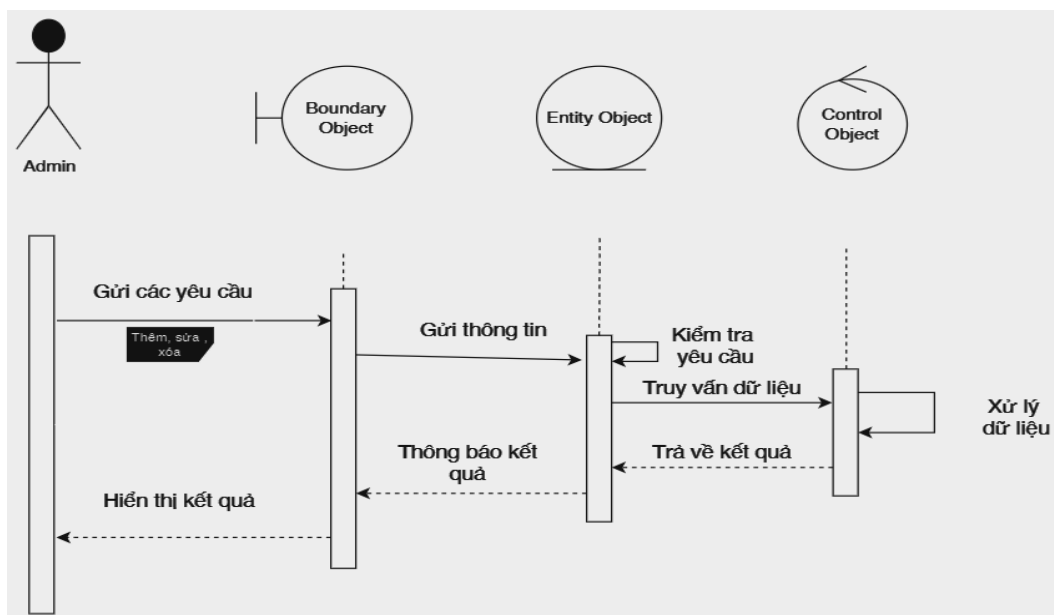
Hình 2. 8 Sơ đồ tuần tự chức năng đặt tour du lịch, khách sạn

Quản lý người dùng



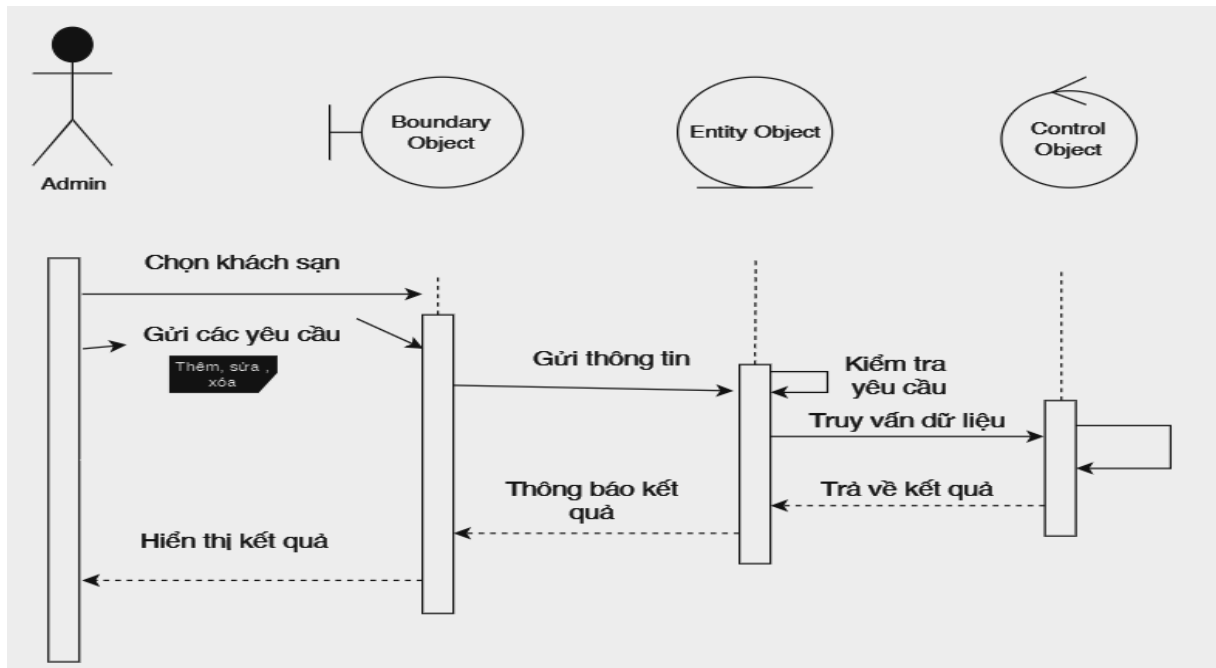
Hình 2. 9 Sơ đồ tuần tự quản lý người dùng

Quản lý tour du lịch, khách sạn



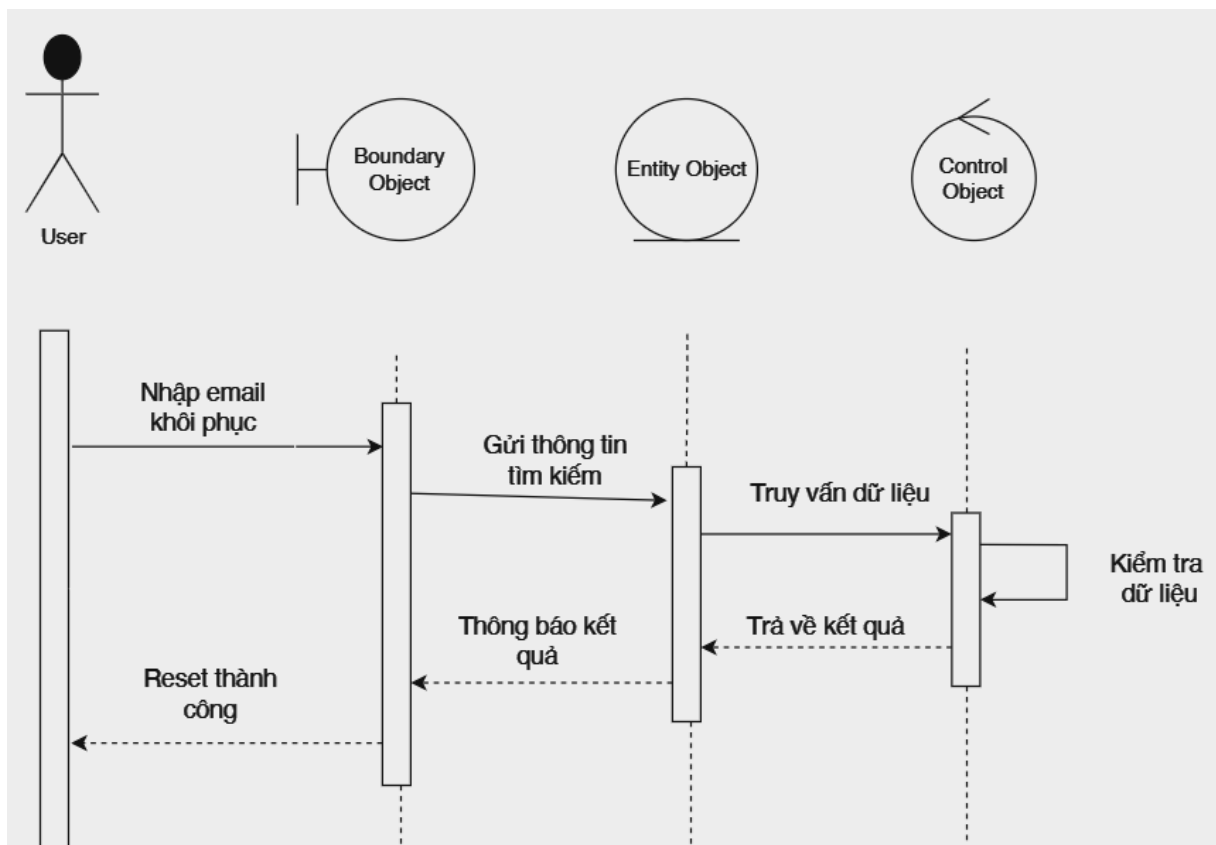
Hình 2. 10 Sơ đồ tuần tự quản lý tour du lịch khách sạn

Quản lý phòng khách sạn



Hình 2. 11 Sơ đồ tuần tự quản lý phòng khách sạn

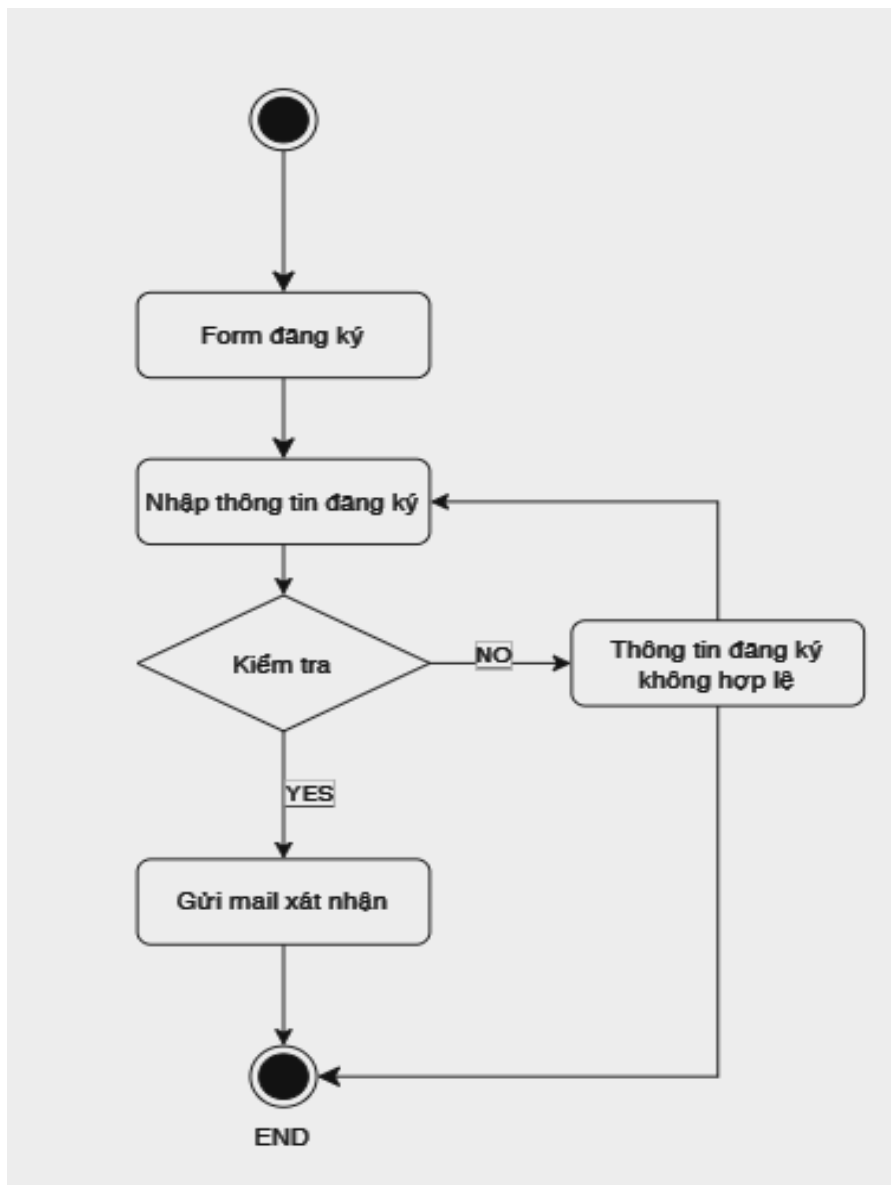
Quên mật khẩu



Hình 2. 12 Sơ đồ tuần tự reset mật khẩu

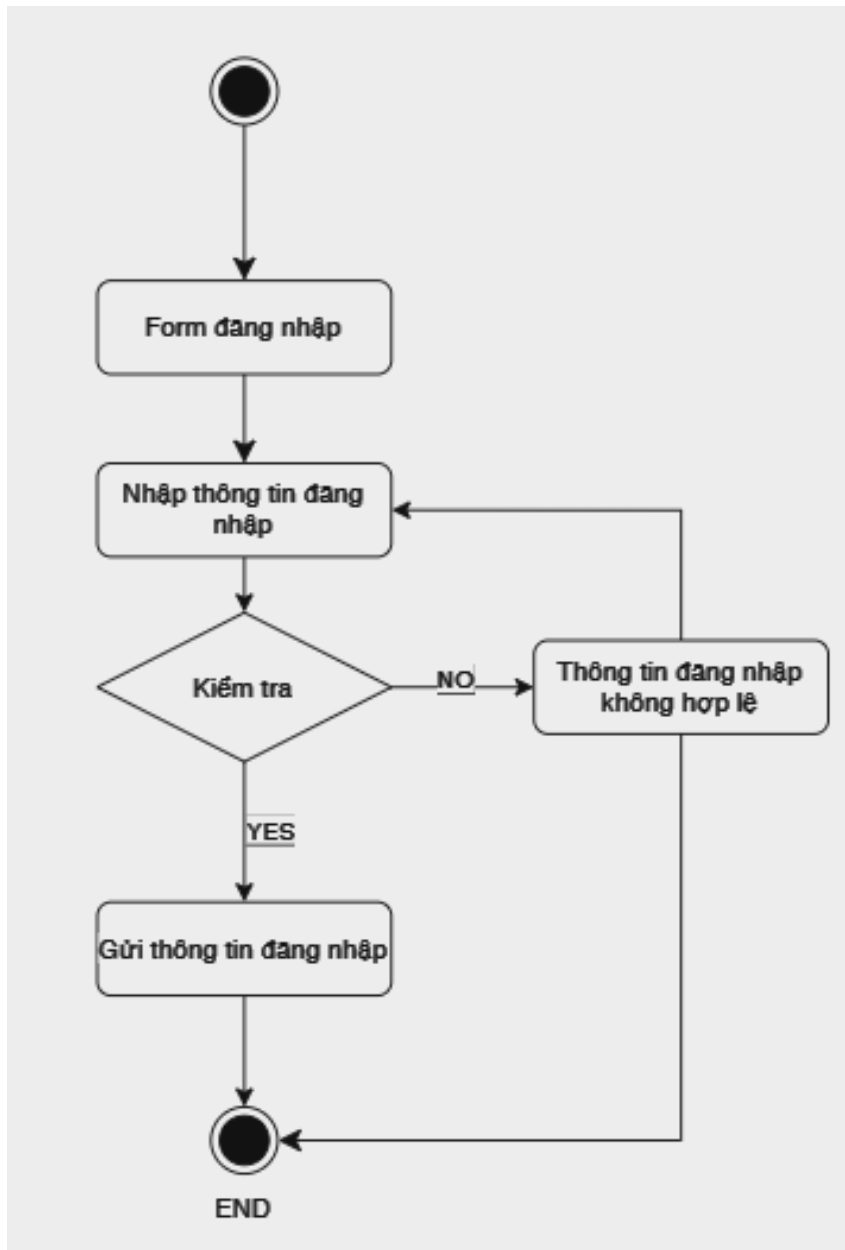
2.3. Sơ đồ hoạt động

Chức năng đăng ký



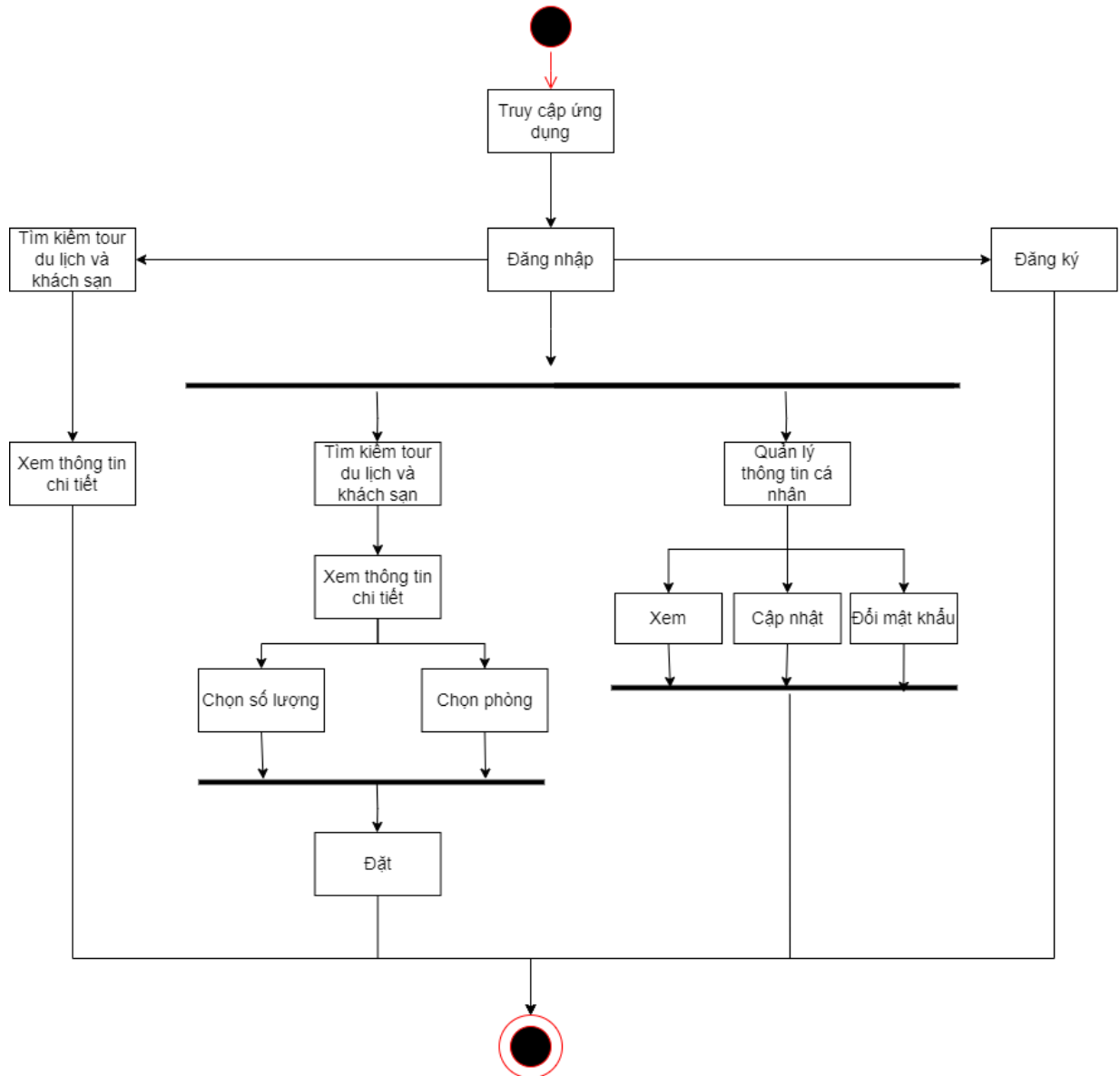
Hình 2. 13 Sơ đồ hoạt động chức năng đăng ký

Chức năng đăng nhập



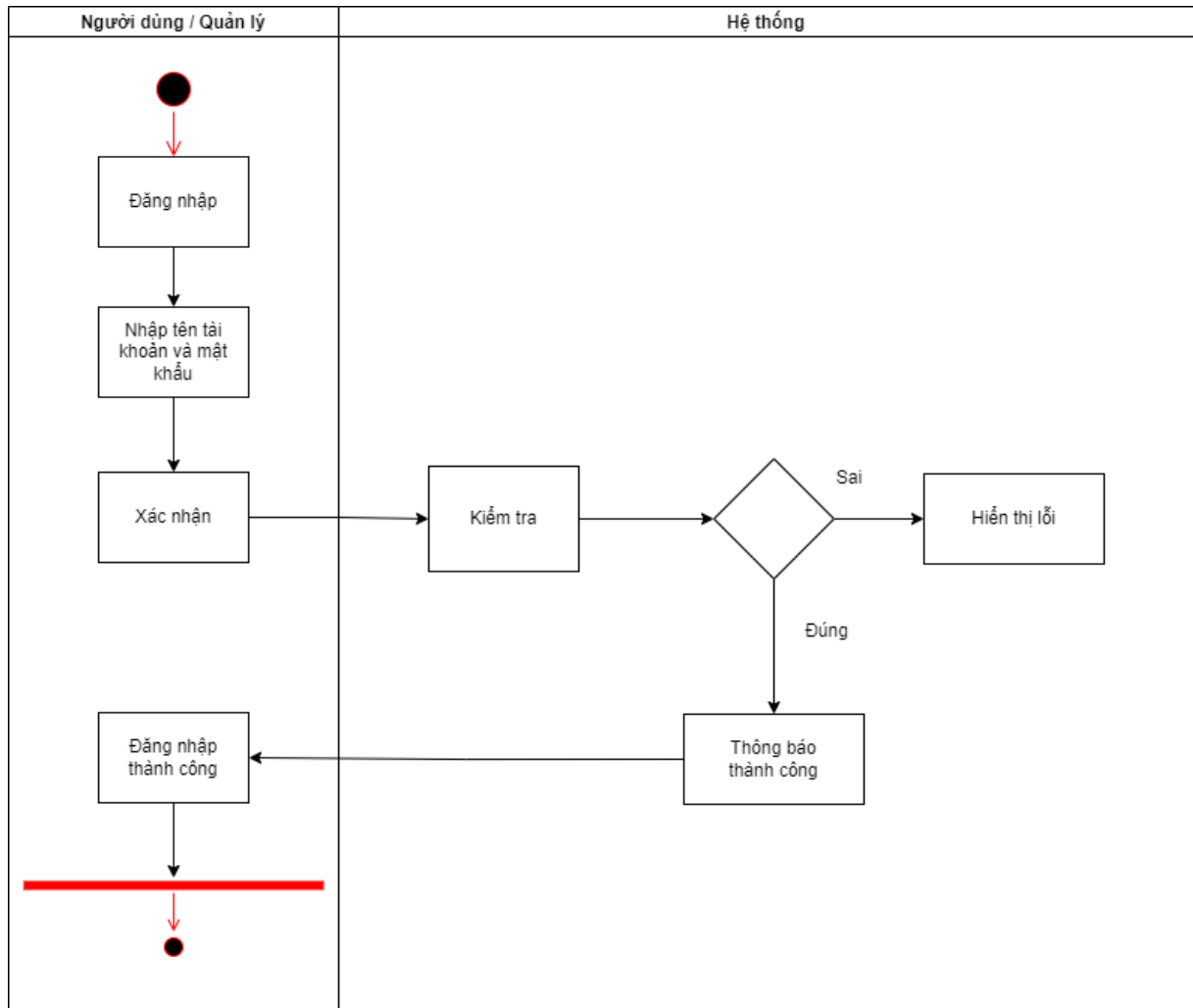
Hình 2. 14 Sơ đồ hoạt động chức năng đăng nhập

Chức năng tổng quát của người dùng



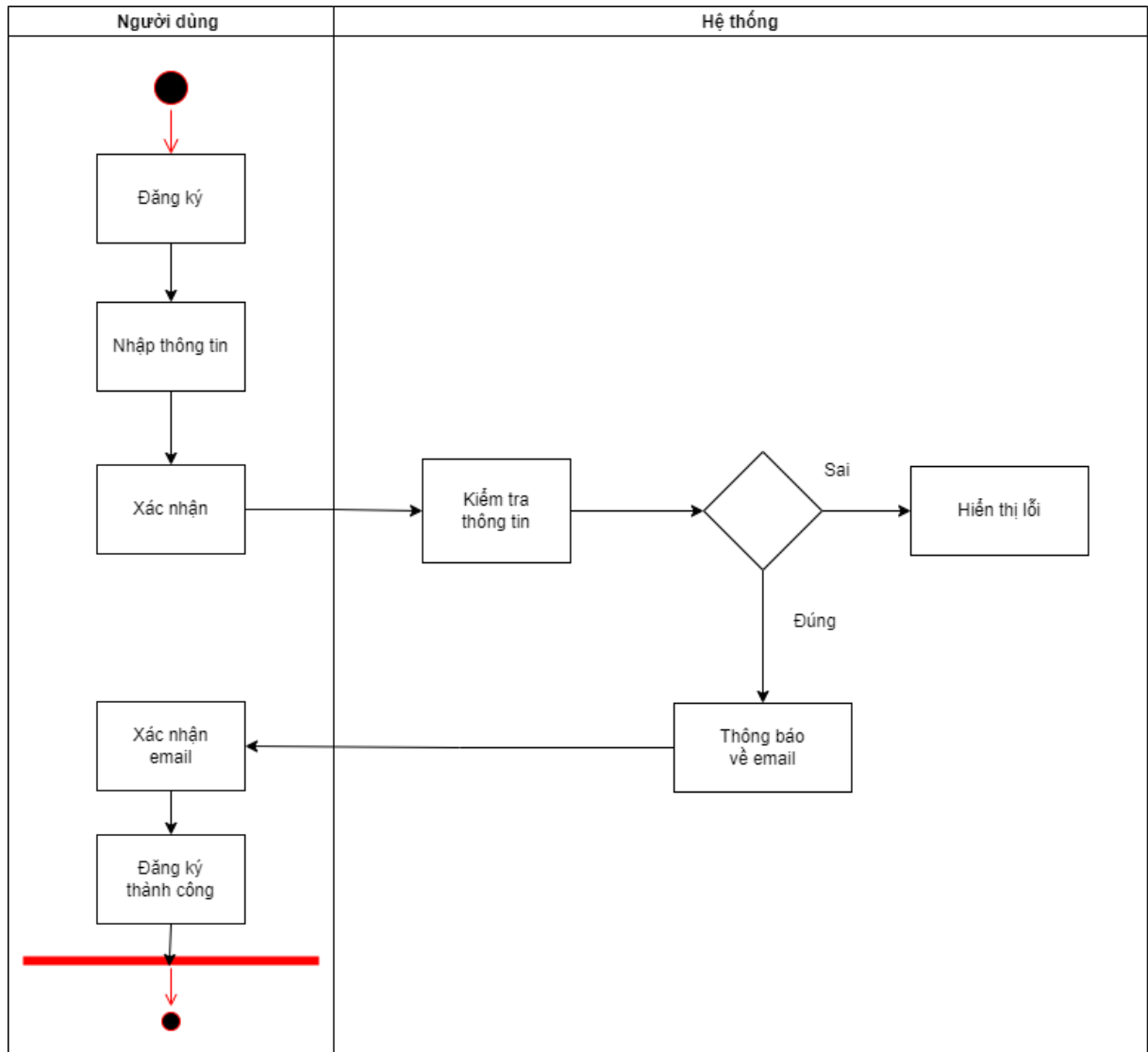
Hình 2. 15 Sơ đồ hoạt động chức năng tổng quát của người dùng

Chức năng đăng nhập



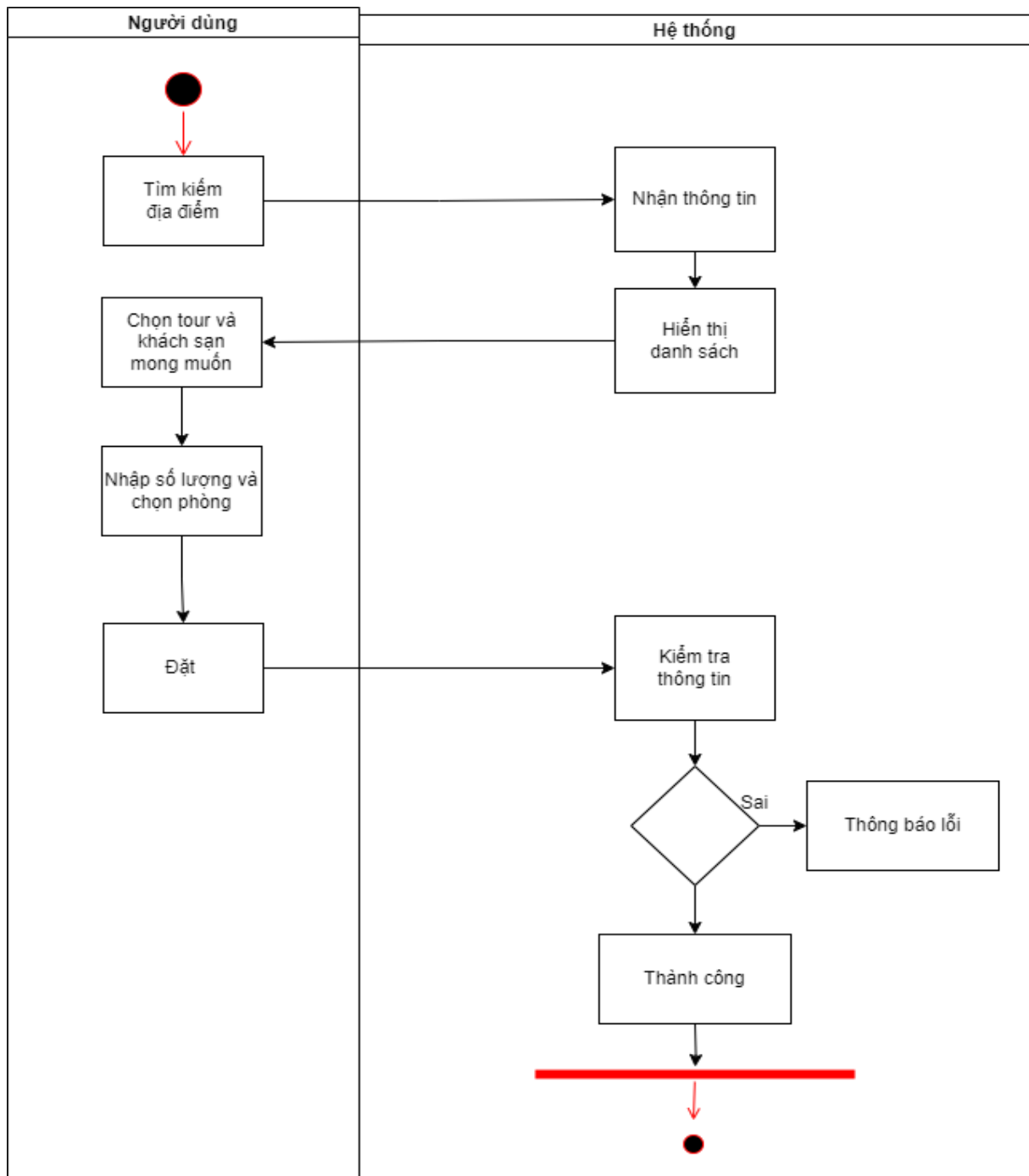
Hình 2. 16 Sơ đồ hoạt động chức năng đăng nhập

Chức năng đăng ký



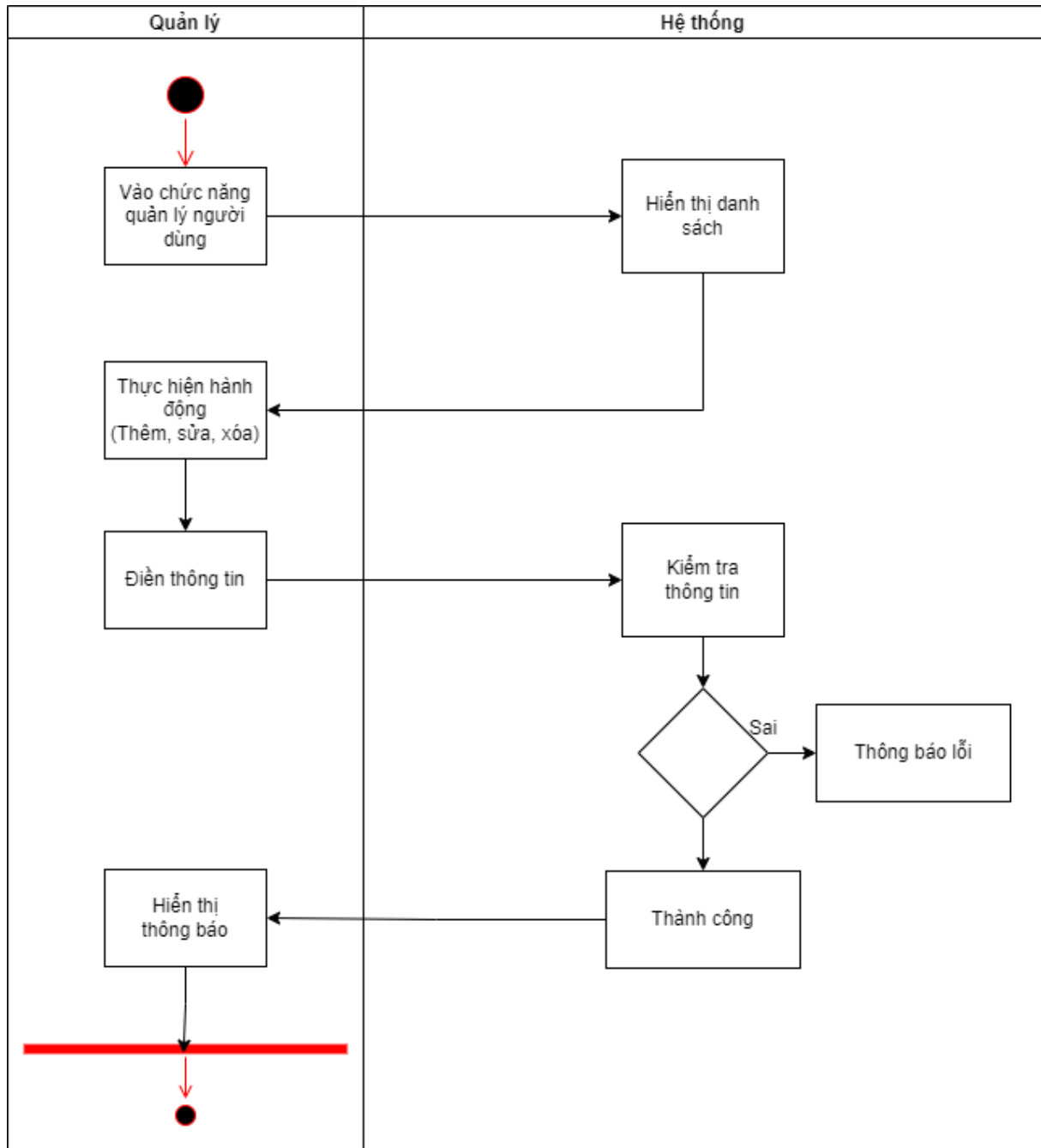
Hình 2. 17 Sơ đồ hoạt động chức năng đăng ký

Chức năng tìm kiếm và đặt của người dùng



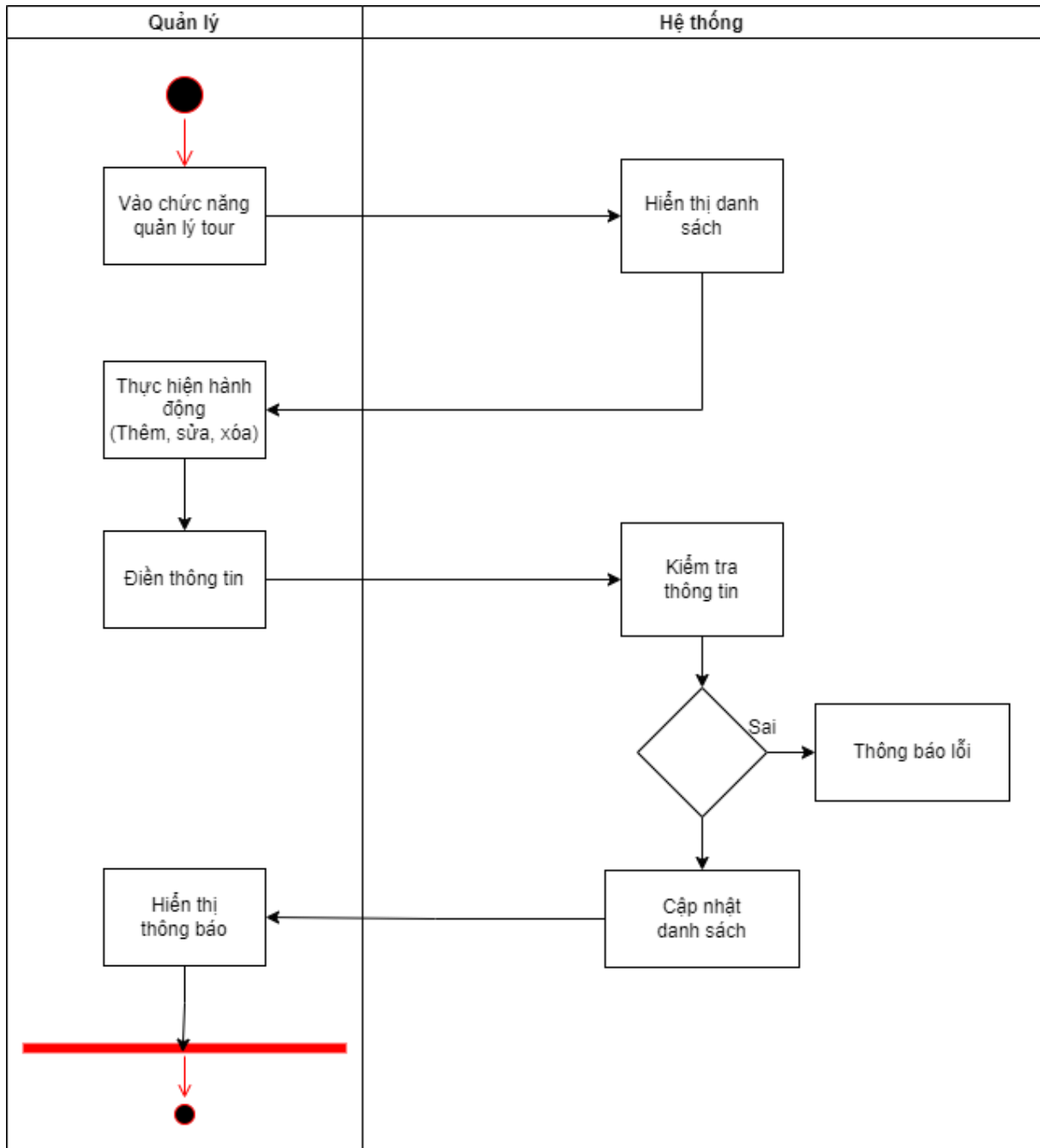
Hình 2. 18 Sơ đồ hoạt động chức năng tìm kiếm và đặt của người dùng

Chức năng quản lý người dùng



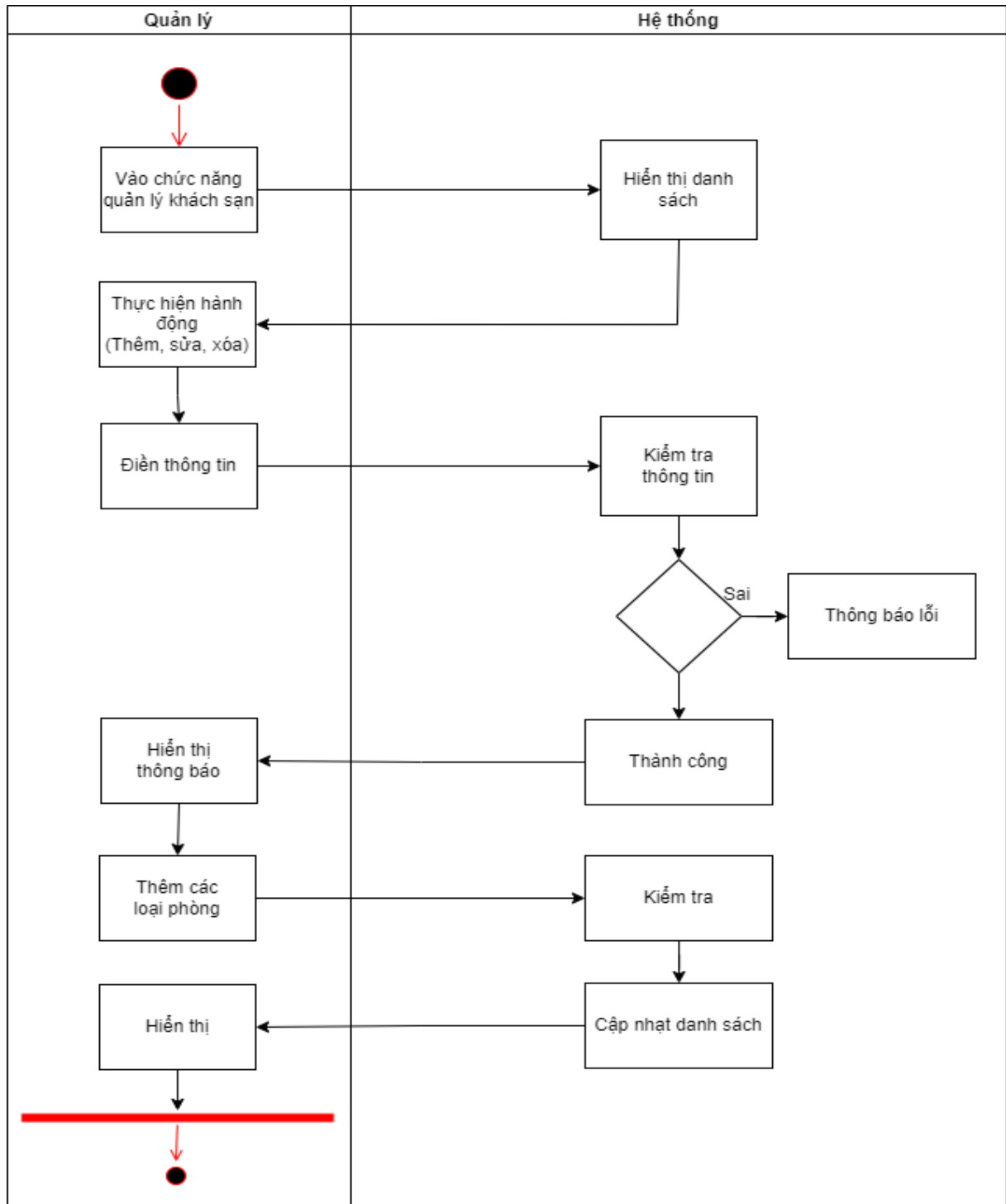
Hình 2. 19 Sơ đồ hoạt động chức năng quản lý người dùng

Chức năng quản lý tour du lịch



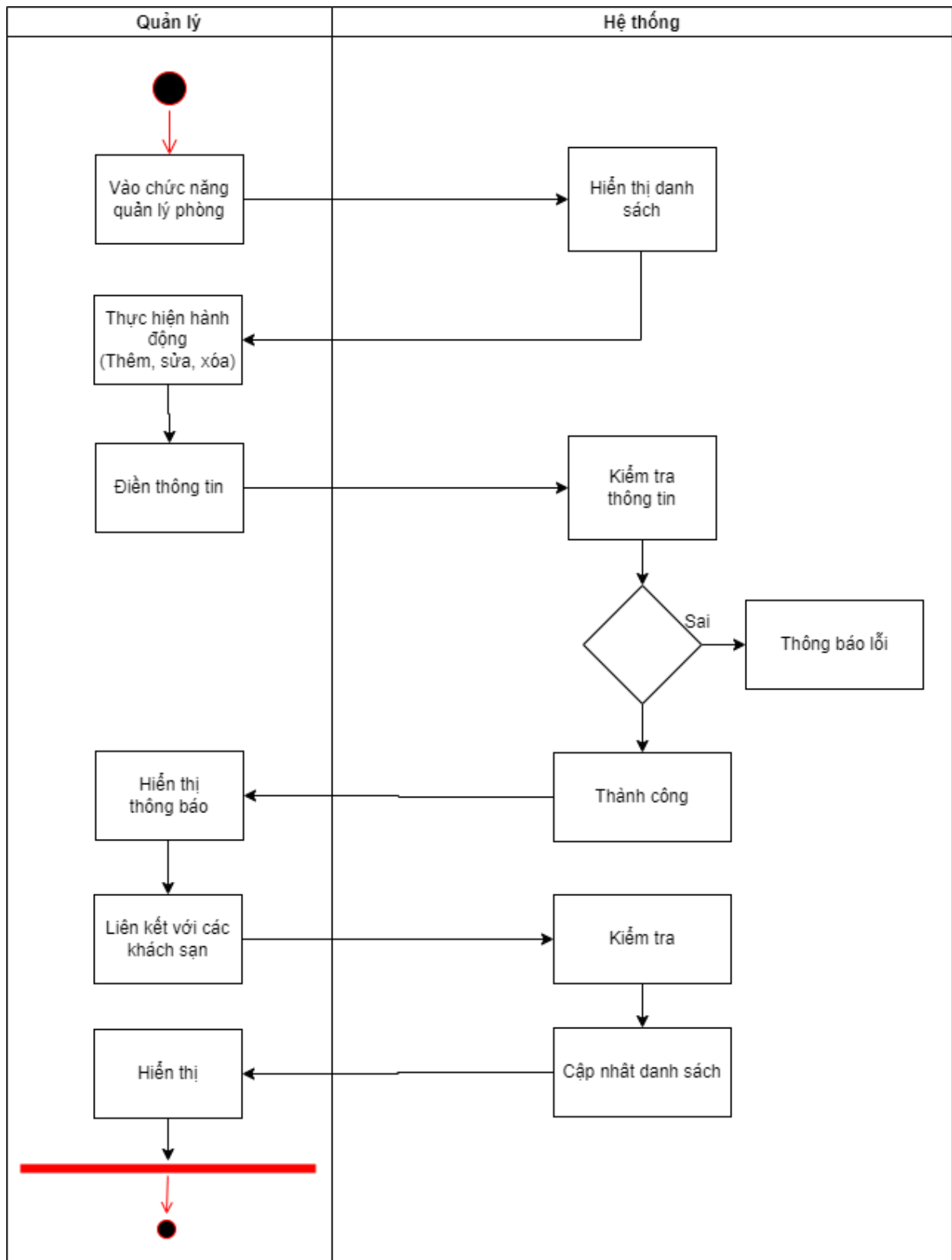
Hình 2. 20 Sơ đồ hoạt động chức năng quản lý tour du lịch

Chức năng quản lý khách sạn



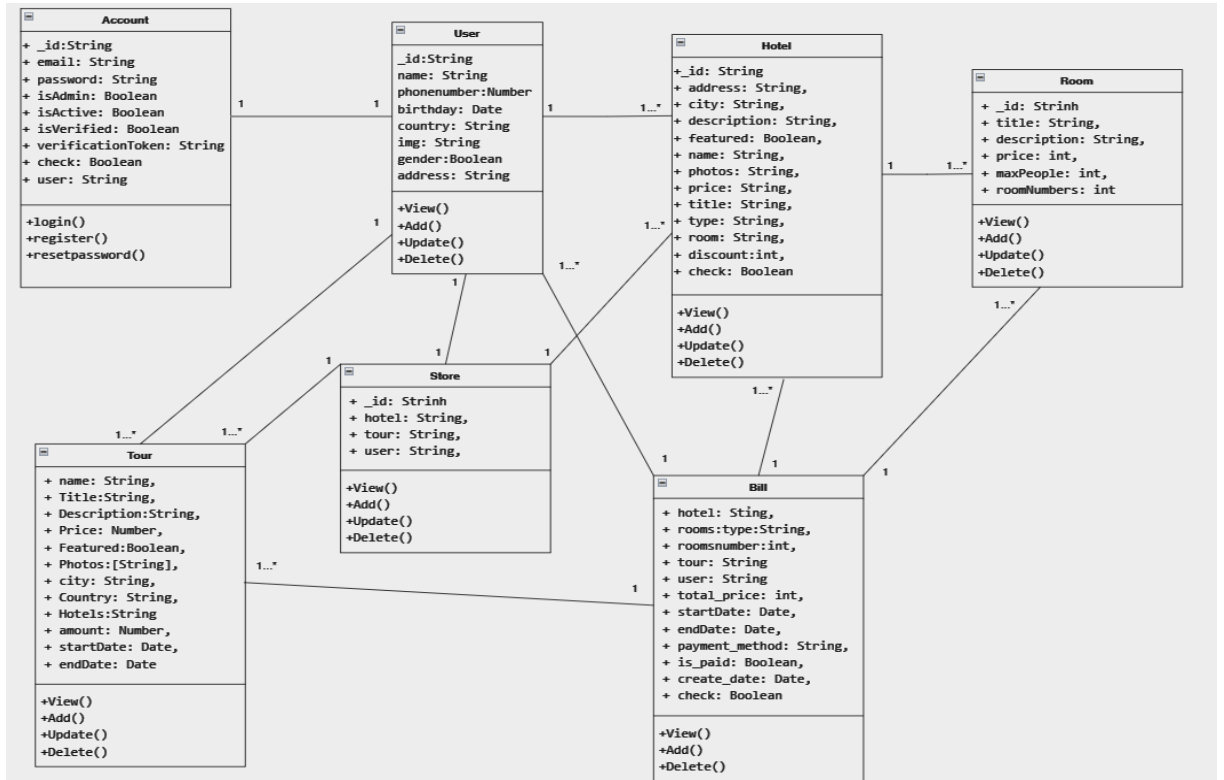
Hình 2. 21 Sơ đồ hoạt động chức năng quản lý khách sạn

Chức năng quản lý phòng



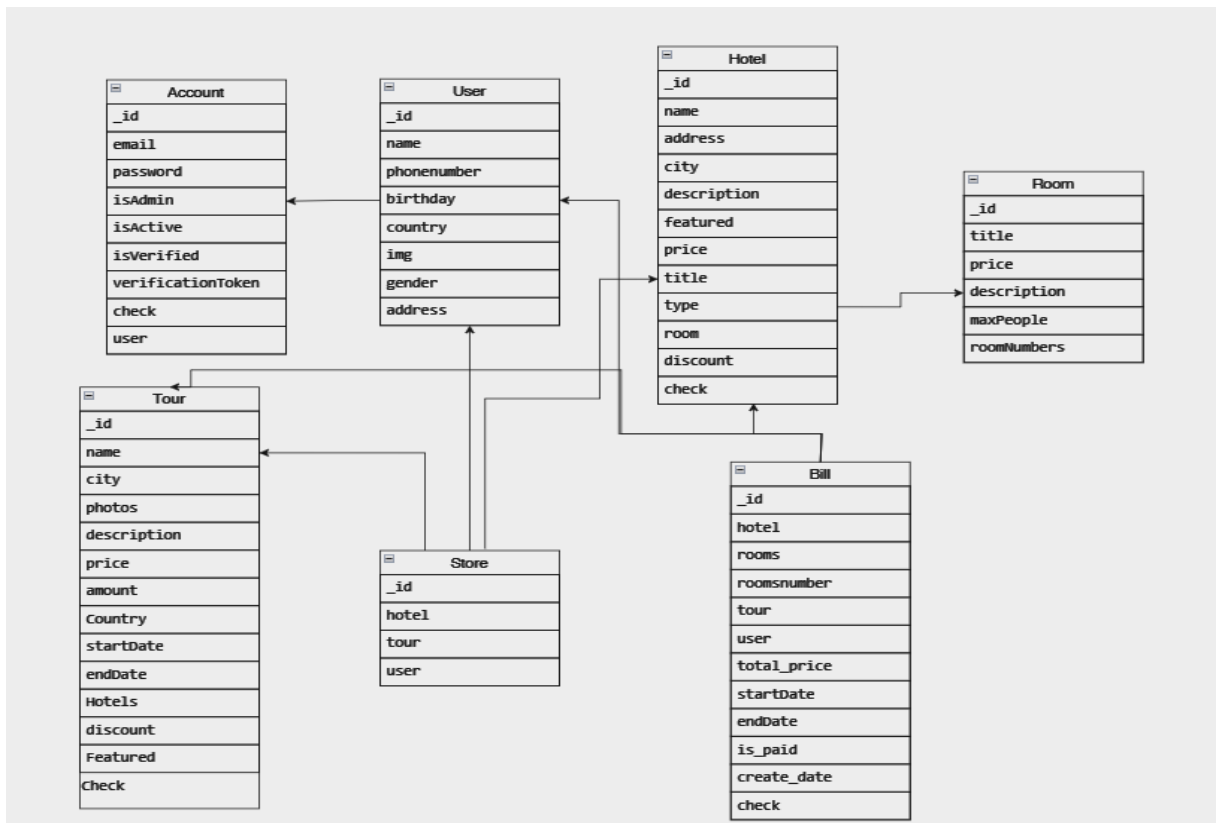
Hình 2. 22 Sơ đồ hoạt động chức năng quản lý phòng

2.4. Sơ đồ lớp



Hình 2. 23 Sơ đồ lớp của hệ thống

2.5. Sơ đồ cơ sở dữ liệu



Hình 2. 24 Cơ sở dữ liệu của hệ thống

Tên trường	Kiểu	Mô tả
_Id	UUID	ID người dùng
Fullname	String	Tên người dùng
PhoneNumber	Float	Số điện thoại
Gender	Boolean	Giới tính
Birthday	Date	Ngày sinh
Country	String	Đất nước
City	String	Thành phố
Img	String	Ảnh đại diện

Bảng 2. 1 Bảng người dùng

Tên trường	Kiểu	Mô tả
_Id	UUID	ID tài khoản
Fullname	String	Tên người dùng
Username	String	Tên tài khoản người dùng
Password	String	Mật khẩu
Email	String	Email
isActive	Boolean	Trạng thái tài khoản
isVerified	Boolean	Trạng thái xác nhận
User_id	Uuid	Id người dùng

Bảng 2. 2 Bảng tài khoản

Tên trường	Kiểu	Mô tả
_Id	UUID	ID tour
Name	String	Tên tour du lịch
Title	String	Tiêu đề
Description	String	Mô tả

Price	Float	Giá cả
Featured	Boolean	Trạng thái nổi bật
Photos	Array	Ảnh của tour
City	String	Nơi diễn ra tour
Country	String	Đất nước diễn ra tour
Amount	Int	Số lượng trên 1 tour
Startdate	Date	Ngày bắt đầu tour
Enddate	Date	Ngày kết thúc tour
Check	Boolean	Kiểm tra tour bị xóa

Bảng 2. 3 Bảng tour

Tên trường	Kiểu	Mô tả
_Id	UUID	ID khách sạn
Name	String	Tên tour khách sạn
Title	String	Tiêu đề
Description	String	Mô tả
Price	Float	Giá cả
Type	String	Loại khách sạn
Featured	Boolean	Trạng thái nổi bật
Photos	Array	Ảnh của khách sạn
City	String	Thành phố
Address	String	Địa chỉ khách sạn
Room_id	Uuid	Id phòng
Check	Boolean	Kiểm tra khách sạn bị xóa

Bảng 2. 4 Bảng khách sạn

Tên trường	Kiểu	Mô tả
------------	------	-------

Id	UUID	ID phòng
Title	String	Tiêu đề phòng
Description	String	Mô tả phòng
Price	Float	Giá cả
MaxPeople	Float	Số người tối đa
RoomNumber_id	Uuid	Id số phòng
Check	Boolean	Kiểm tra phòng bị xóa

Bảng 2. 5: Bảng số phòng

Tên trường	Kiểu	Mô tả
_Id	UUID	ID Bill
HotelId	UUID	Id khách sạn
RoomID	UUID	Id phòng
Roomnumber	Array	Số phòng đặt
Total_price	Float	Số người tối đa
Payment_method	String	Phương thức thanh toán
User_id	Uuid	Id người dùng
Check	Boolean	Kiểm tra bill bị xóa
Startdate	Date	Ngày bắt đầu
Enddate	Date	Ngày kết thúc

Bảng 2.6 : Bảng bill

Tên trường	Kiểu	Mô tả
_Id	UUID	ID Bill
Hotel	Array	Id khách sạn
User	Array	Id người dùng

Tour	Array	Id tour
------	-------	---------

Bảng 2.7 : Bảng Store

3. Kết chương

Thông qua chương 2, tôi đã thể hiện và trình bày rõ các đặc tả yêu cầu mà hệ thống có thể đáp ứng cho nhu cầu của người dùng. Những yêu cầu này chủ yếu là yêu cầu thiết kế. Các sơ đồ hoạt động, vẽ luồng hoạt động của một hệ thống, hiển thị các bước và chuỗi hành động cũng như các tương tác giữa người dùng và hệ thống. Qua đó, tổng quan và các luồng hoạt động của hệ thống được trình bày đầy đủ.

Chương 3: TRIỂN KHAI DỰ ÁN VÀ KẾT QUẢ

1. Môi trường triển khai dự án

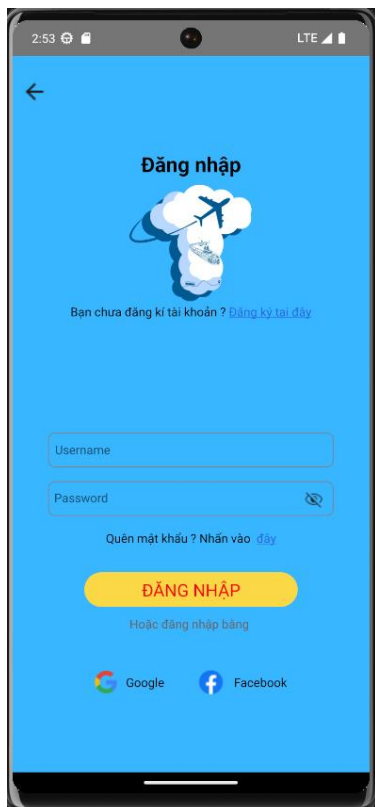
Dưới đây là những các công cụ và framework chính đã triển khai trong đồ án này

Loại	Tên
------	-----

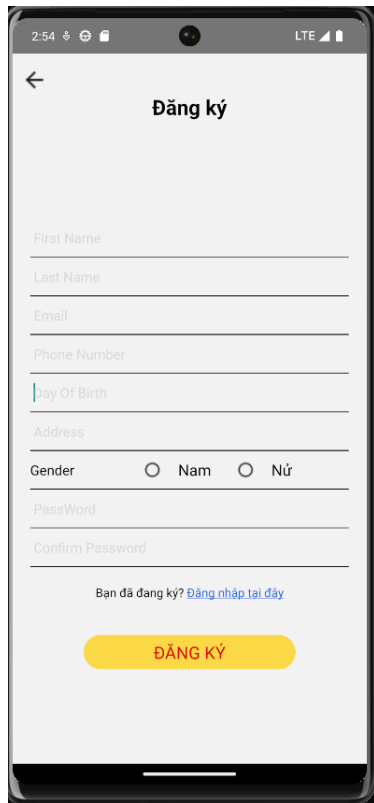
IDE	Visual Studio Code
Ngôn ngữ lập trình	JavaScript
Công cụ quản trị cơ sở dữ liệu	MongoDB
Front-end	React Native, Reactjs
Back-end	NodeJS
Hệ điều hành	Androi, Windows, MacOS, ...

2. Kết quả dự án

2.1. Màn hình đăng nhập và đăng ký

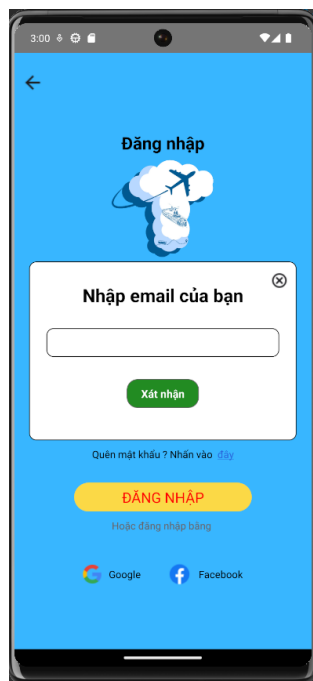


Hình 3. 1 Màn hình đăng nhập



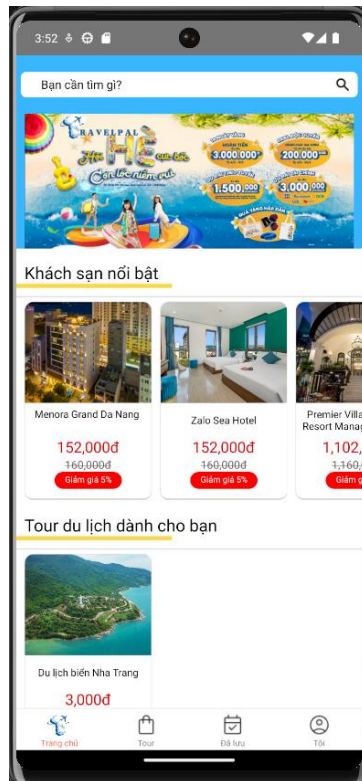
Hình 3. 2 Màn hình đăng ký

2.2. Màn hình quên mật khẩu



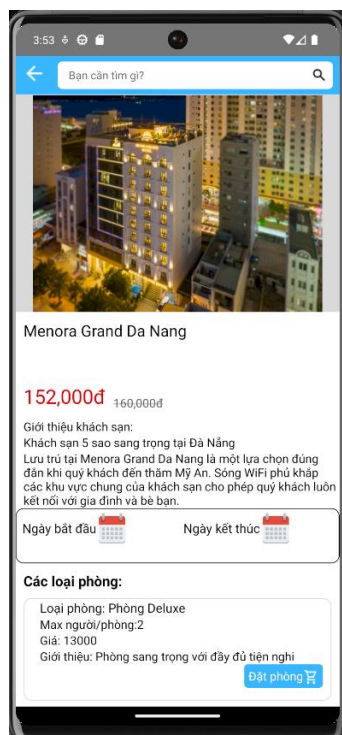
Hình 3. 3 Màn hình quên mật khẩu

2.3. Màn hình trang chủ



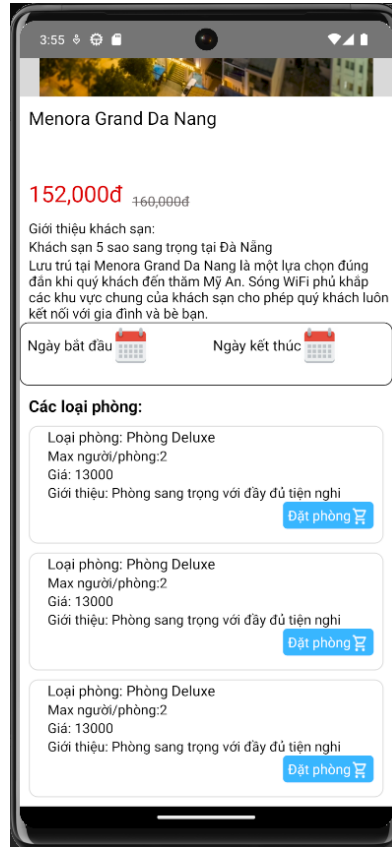
Hình 3. 4 Màn hình trang chủ

2.4. Màn hình khách sạn

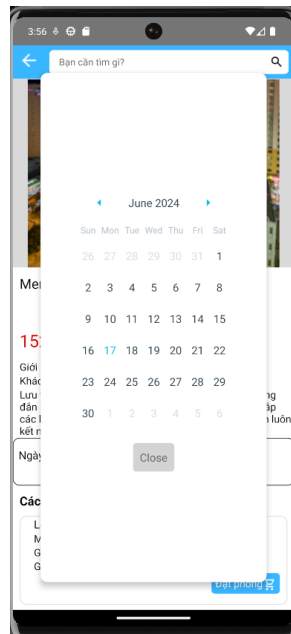


Hình 3. 5 Màn hình danh sách khách sạn

2.5. Màn hình chi tiết khách sạn

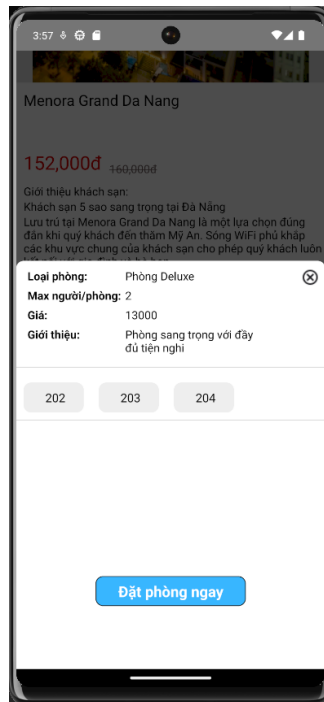


Hình 3. 6 Màn hình chi tiết khách sạn



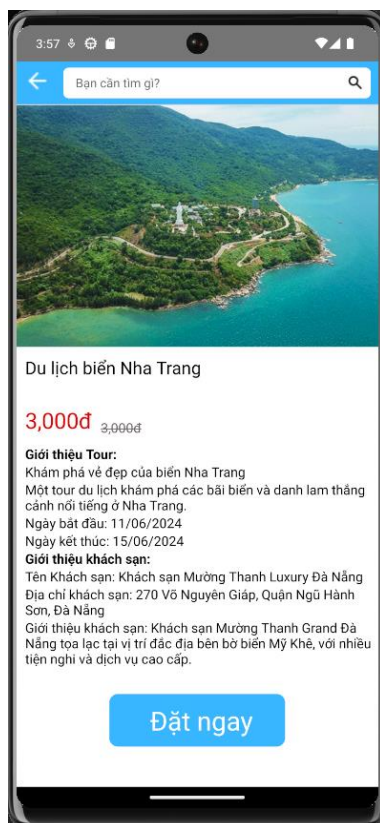
Hình 3. 7 Màn hình chi tiết chọn ngày khách sạn

2.6. Màn hình chi tiết phòng khách sạn



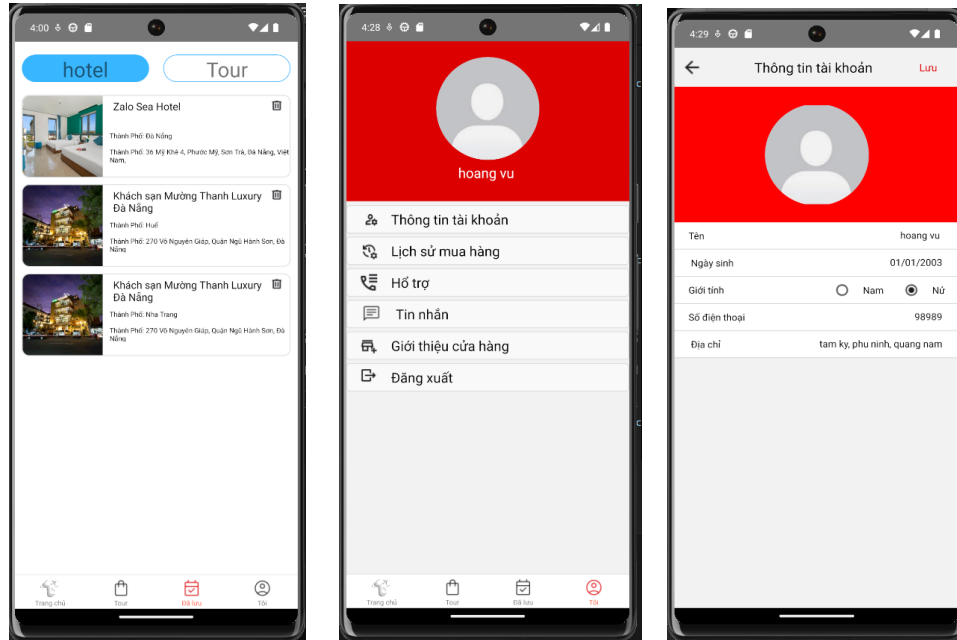
Hình 3. 8 Màn hình chi tiết phòng khách sạn

2.7. Màn hình tour



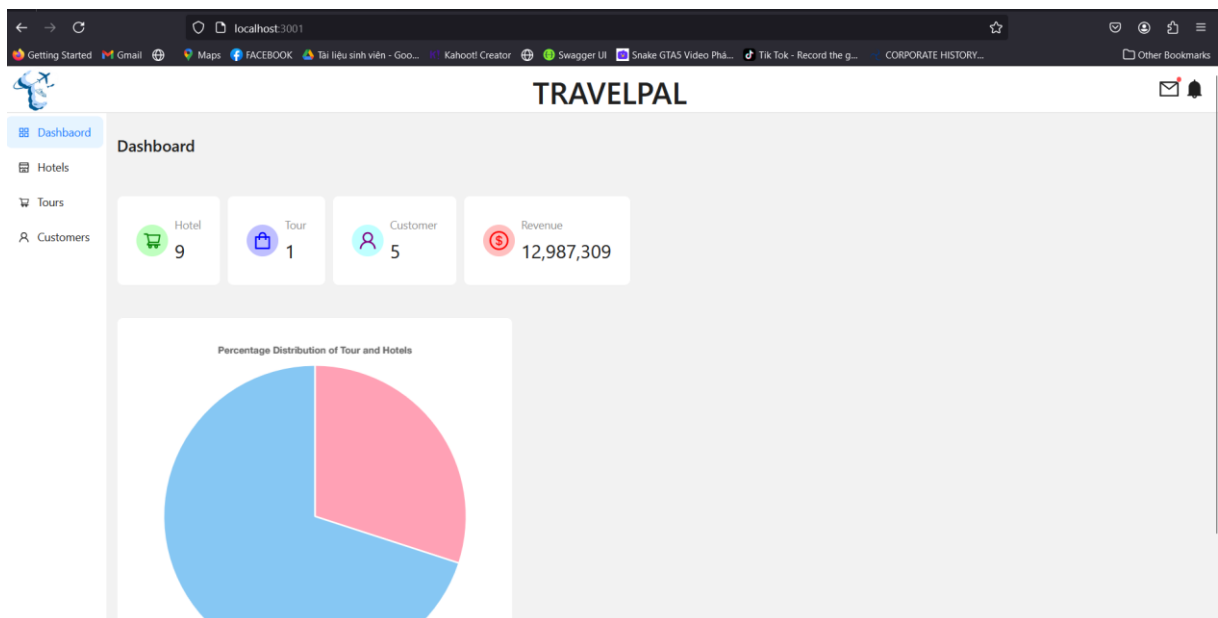
Hình 3. 10 Màn hình chi tiết tour

2.8. Các màn hình khác



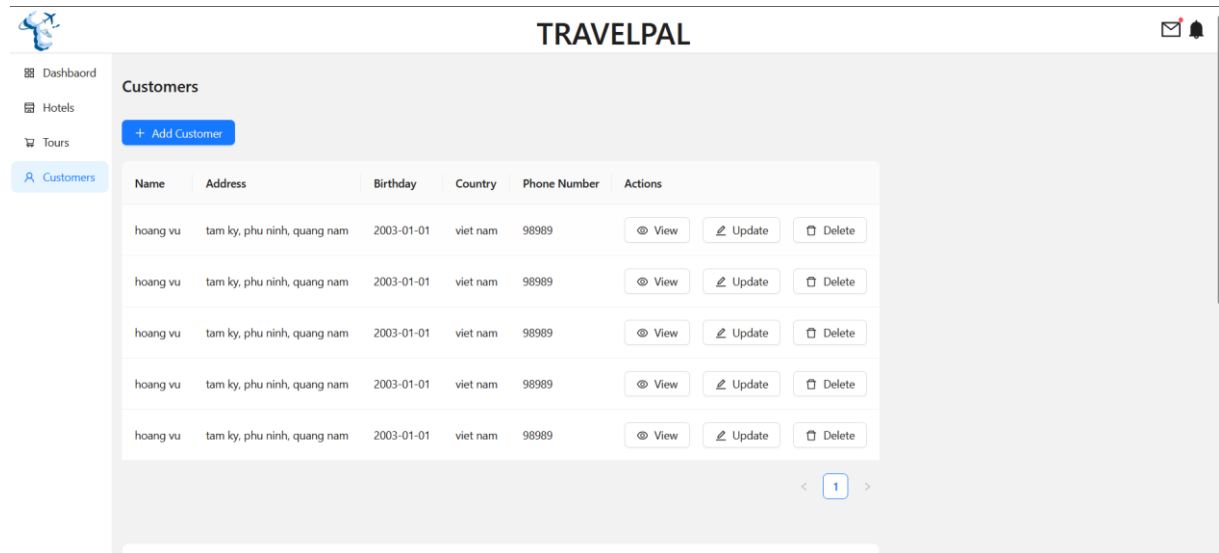
Hình 3. 11 Màn hình lưu trữ và thông tin cá nhân

2.9. Màn hình thống kê



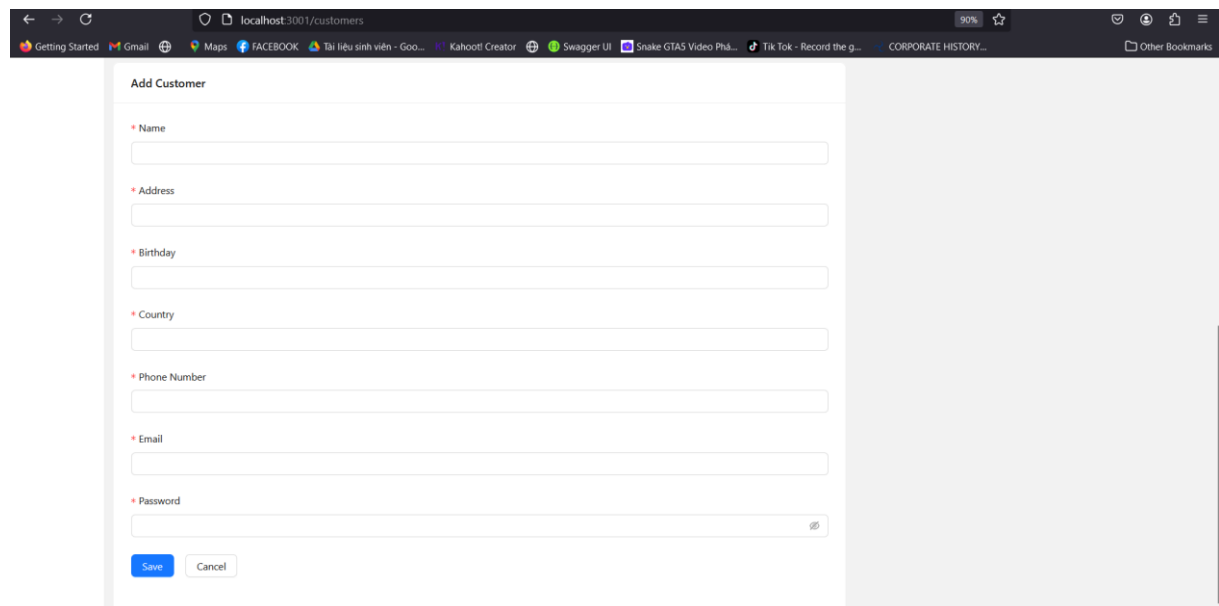
Hình 3. 9 Màn hình thống kê

2.10. Màn hình quản lý người dùng



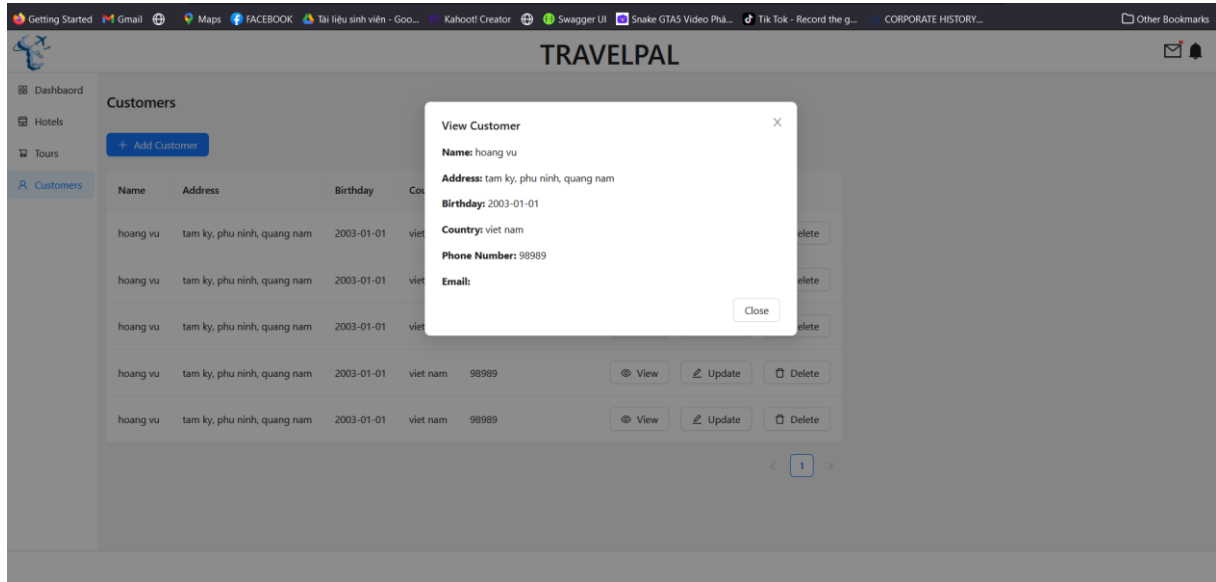
Hình 3. 10 Màn hình quản lý người dùng

2.11. Màn hình thêm người dùng



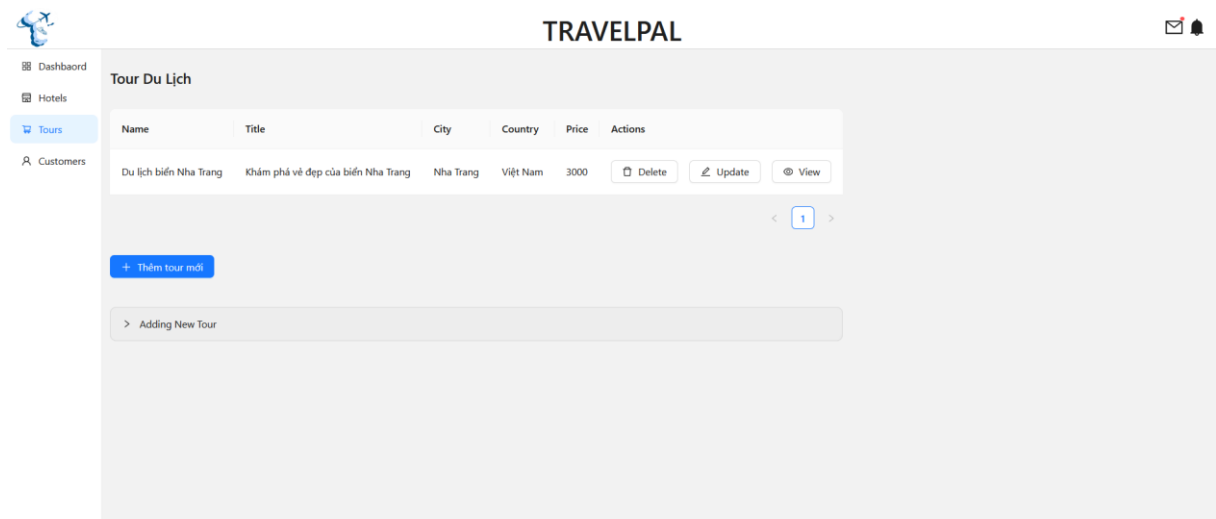
Hình 3. 11 Màn hình thêm người dùng

2.12. Màn hình chi tiết người dùng



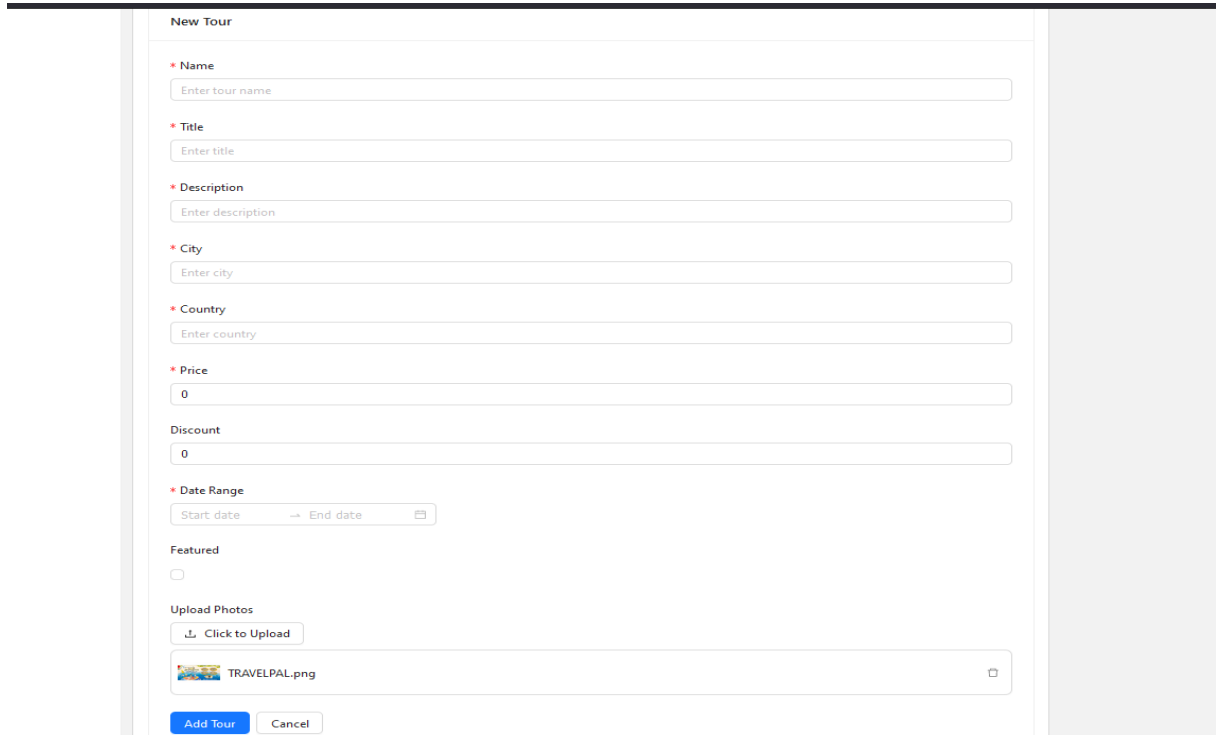
Hình 3. 12 Màn hình chi tiết người dùng

2.13. Màn hình danh sách tour



Hình 3. 13 Màn hình danh sách tour

2.14. Màn hình thêm mới tour



New Tour

* Name
Enter tour name

* Title
Enter title

* Description
Enter description

* City
Enter city

* Country
Enter country

* Price
0

Discount
0

* Date Range
Start date → End date

Featured
☐

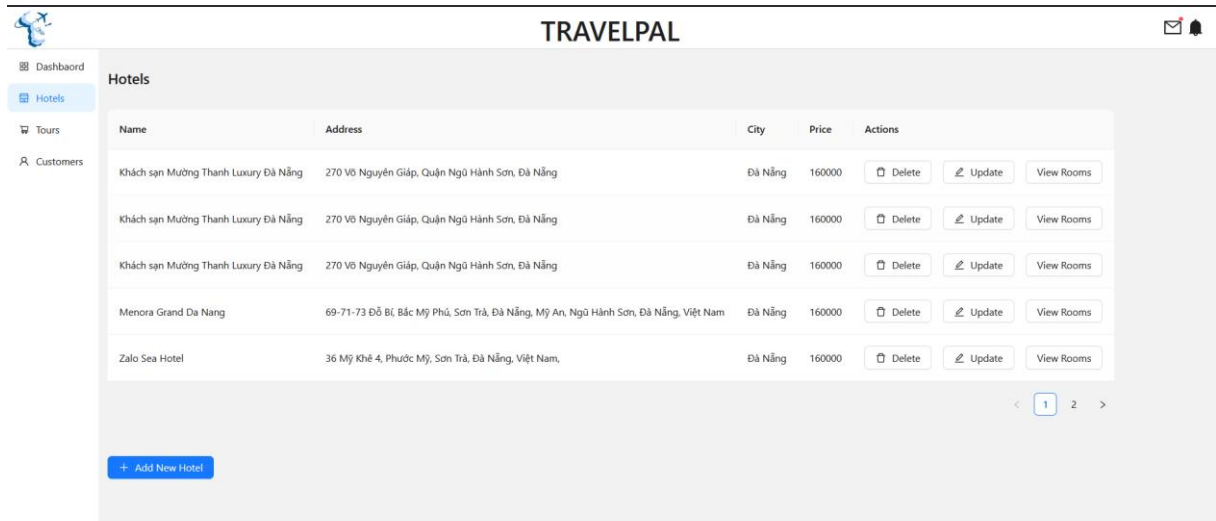
Upload Photos
Click to Upload

TRAVELPAL.png

Add Tour Cancel

Hình 3. 14 Màn hình thêm mới tour

2.15. Màn hình danh sách khách sạn



TRAVELPAL

Dashboard
Hotels
Tours
Customers

Hotels

Name	Address	City	Price	Actions
Khách sạn Mường Thanh Luxury Đà Nẵng	270 Võ Nguyên Giáp, Quận Ngũ Hành Sơn, Đà Nẵng	Đà Nẵng	160000	Delete Update View Rooms
Khách sạn Mường Thanh Luxury Đà Nẵng	270 Võ Nguyên Giáp, Quận Ngũ Hành Sơn, Đà Nẵng	Đà Nẵng	160000	Delete Update View Rooms
Khách sạn Mường Thanh Luxury Đà Nẵng	270 Võ Nguyên Giáp, Quận Ngũ Hành Sơn, Đà Nẵng	Đà Nẵng	160000	Delete Update View Rooms
Menora Grand Da Nang	69-71-73 Đỗ Bì, Bắc Mỹ Phú, Sơn Trà, Đà Nẵng, Mỹ An, Ngũ Hành Sơn, Đà Nẵng, Việt Nam	Đà Nẵng	160000	Delete Update View Rooms
Zalo Sea Hotel	36 Mỹ Khê 4, Phước Mỹ, Sơn Trà, Đà Nẵng, Việt Nam,	Đà Nẵng	160000	Delete Update View Rooms

+ Add New Hotel

< 1 2 >

Hình 3. 15 Màn hình danh sách khách sạn

2.16. Màn hình thêm mới khách sạn

New Hotel

Name
Enter hotel name

Address
Enter address

City
Enter city

Description
Enter description

Type
Enter type

Price
Enter price

Discount
Enter discount

Add Hotel **Cancel**

Hình 3. 16 Màn hình thêm mới khách sạn

2.17. Màn hình chi tiết khách sạn

▼ Menora Grand Da Nang

Khách sạn 5 sao sang trọng tại Đà Nẵng

Description: Lưu trú tại Menora Grand Da Nang là một lựa chọn đúng đắn khi quý khách đến thăm Mỹ An. Sóng WIFI phủ khắp các khu vực chung của khách sạn cho phép quý khách luôn kết nối với gia đình và bè bạn.

Address: 69-71-73 Đỗ Bì, Bắc Mỹ Phú, Sơn Trà, Đà Nẵng, Mỹ An, Ngũ Hành Sơn, Đà Nẵng, Việt Nam

City: Đà Nẵng

Price: 160000

Type: Resort

Discount: 5

Close Details

> Rooms

Hình 3. 17 Màn hình chi tiết khách sạn

2.18. Màn hình danh sách các loại phòng khách sạn

Phòng Deluxe

Description: Phòng sang trọng với đầy đủ tiện nghi

Price: 13000

Max People: 2

Room Numbers: 202, 203, 204

Edit Room

Phòng Deluxe

Description: Phòng sang trọng với đầy đủ tiện nghi

Price: 13000

Max People: 2

Room Numbers: 202, 201, 204, 205, 206, 207

Edit Room

Hình 3. 18 Màn hình danh sách các loại phòng khách sạn

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Qua quá trình xây dựng hệ thống này, em đã có cơ hội hiểu rõ hơn về cách thức tạo ra một ứng dụng web hoàn chỉnh, từ giai đoạn lên ý tưởng, thiết kế giao diện, vẽ sơ đồ, xử lý dữ liệu cho đến triển khai toàn hệ thống. Các kỹ năng của em đã được cải thiện đáng kể, bao gồm: nghiên cứu kỹ thuật, tìm kiếm và đọc tài liệu tiếng Anh, thuyết trình và tạo slide trình bày. Về dự án, em đã hoàn thành các chức năng cơ bản theo kế hoạch ban đầu được đề ra trong tuần đầu tiên của thời gian làm đồ án tốt nghiệp. Đồng thời, em cũng đã cố gắng tối ưu hóa, đơn giản hóa và nâng cao giao diện, làm cho ứng dụng trở nên linh hoạt và thân thiện hơn với người dùng.

Ứng dụng này cho phép người dùng đặt du lịch và khách sạn đăng ký tài khoản, quản lý thông tin dịch vụ, tạo và chỉnh sửa các gói dịch vụ, và tiếp nhận các đơn đặt phòng hoặc tour từ khách hàng. Người dùng có thể tìm kiếm khách sạn và tour theo địa điểm, ngày khởi hành, giá cả và các tiêu chí khác. Ngoài ra, ứng dụng cũng cung cấp thông tin chi tiết về các dịch vụ, bao gồm lịch trình, điểm đến, giá cả và hình ảnh minh họa.

2. Hạn chế

Dù đã đạt được nhiều kết quả tích cực, em nhận thấy hệ thống vẫn còn tồn tại một số hạn chế sau:

- Một số vấn đề quản trị thông tin dịch vụ vẫn chưa được giải quyết triệt để.
- Em vẫn chưa hoàn toàn nắm vững hết các công nghệ mới.
- Để đạt hiệu quả tối đa, cần thêm các giai đoạn phát triển, kiểm tra và tối ưu hóa hệ thống.
- Một số chức năng vẫn còn phức tạp và chưa được đơn giản hóa hoàn toàn.

3. Hướng phát triển

- Tiếp tục nâng cấp, cập nhật thông tin cũng như các tính năng người dùng yêu cầu.
- Cải thiện giao diện nhằm nâng cao trải nghiệm người dùng.
- Tích hợp hệ thống đánh giá và nhận xét
- Mở rộng hệ thống và kết nối với đối tác

TÀI LIỆU THAM KHẢO

- [1] Tìm hiểu Express:
[\[Phần 1\] Tìm hiểu Express js Framework \(viblo.asia\)](#)
- [2] Giới thiệu ReactJS:
<https://vi.reactjs.org/tutorial/tutorial.html>
- [3] Giới thiệu về React Native
<https://reactnative.dev/docs/environment-setup>
- [4] Giới thiệu NodeJS:
<https://nodejs.org/en/docs>
- [5] Giới thiệu MongoDB:
<https://www.mongodb.com/docs/>
- [6] Cloudinary:
<https://cloudinary.com/documentation>
- [7] Google Cloud:
<https://cloud.google.com/docs>
- [8] Mô hình MVC:
<https://monamedia.co/mvc-la-gi-ung-dung-cua-mo-hinh-mvc-trong-lap-trinh/>