

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

**Smart Trip - Phần mềm hỗ trợ thiết kế và thực hiện
lộ trình du lịch cá nhân thông minh**

NGUYỄN THỊ THÙY DUNG

dung.ntt215009@sis.hust.edu.vn

Ngành Công nghệ thông tin và truyền thông

Giảng viên hướng dẫn: PGS. TS. Cao Tuấn Dũng

Chữ ký GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 06/2025

LỜI CẢM ƠN

Lời đầu tiên, em xin được bày tỏ lòng biết ơn sâu sắc tới PGS.TS Cao Tuấn Dũng – người thầy đã luôn tận tình hướng dẫn, chỉ bảo em trong suốt quá trình thực hiện đồ án tốt nghiệp. Em rất trân trọng những góp ý quý báu của thầy về các tính năng, về logic và tính thực tiễn của sản phẩm. Những định hướng rõ ràng, khoa học và sát với thực tế của thầy đã giúp em hoàn thiện đồ án không chỉ về mặt kỹ thuật mà còn về tư duy giải quyết vấn đề. Em biết rằng nếu không có sự đồng hành của thầy, em sẽ không thể hoàn thiện đồ án một cách trọn vẹn như ngày hôm nay.

Tiếp theo, em xin được gửi lời cảm ơn chân thành đến các thầy cô trong Trường Công nghệ Thông tin và Truyền thông – Đại học Bách Khoa Hà Nội. Trong suốt 4 năm học, thầy cô không chỉ truyền đạt kiến thức mà còn dạy em cách tư duy, làm việc chuyên nghiệp và không ngừng nỗ lực vươn lên. Những gì em học được từ mái trường này sẽ mãi là hành trang quý báu cho con đường phía trước.

Em cũng xin gửi lời tri ân sâu sắc đến bố mẹ và em gái – những người thân yêu luôn ở bên, đồng hành cùng em trong mọi giai đoạn của hành trình này. Đã có những lúc em cảm thấy mệt mỏi, hoài nghi, nhưng chính tình yêu thương và sự động viên âm thầm từ gia đình là điều giúp em mạnh mẽ vượt qua tất cả. Con biết rằng thành quả ngày hôm nay cũng là thành quả của niềm tin và sự hi sinh từ bố mẹ và em gái. Con cảm ơn cả nhà rất nhiều.

Em muốn đặc biệt gửi lời cảm ơn đến những người bạn trong nhóm học tập của em: Thảo, Trang, Châu, Linh, Thư, Nhung, Phúc, Trung Anh, Duy, Hà, Huy – những người bạn đã cùng em đi qua biết bao tiết học, cùng nhau ôn thi, cùng nhau chia sẻ những buổi đi chơi, cùng cười cùng với những áp lực. Các bạn không chỉ là bạn học mà còn là một phần thanh xuân đáng nhớ của em. Đặc biệt, em cảm ơn các bạn vì đã luôn sẵn lòng góp ý để giúp em cải tiến đồ án, hoàn thiện từng chi tiết nhỏ.

Cuối cùng, em xin cảm ơn Đại học Bách Khoa Hà Nội – nơi đã cho em 4 năm thanh xuân với thật nhiều trải nghiệm, thử thách và cơ hội. Bách Khoa không chỉ là một ngôi trường mà còn là một phần ký ức đáng trân trọng trong cuộc đời em. Dù mai này có đi đâu, em cũng sẽ luôn tự hào là sinh viên Bách Khoa.

Em xin chân thành cảm ơn.

LỜI CAM KẾT

Họ và tên sinh viên: Nguyễn Thị Thùy Dung

Điện thoại liên lạc: 0382175198

Email: Dung.ntt215009@sis.hust.edu.vn

Lớp: Việt Nhật 04 K66

Hệ đào tạo: Công nghệ thông tin Việt Nhật

Tôi – *Nguyễn Thị Thùy Dung* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *PGS. TS. Cao Tuấn Dũng*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 15 tháng 6 năm 2025

Tác giả ĐATN

Nguyễn Thị Thùy Dung

TÓM TẮT NỘI DUNG ĐỒ ÁN

Nhu cầu tự tổ chức du lịch ngày càng tăng do xu hướng cá nhân hóa trải nghiệm, tuy nhiên các dịch vụ hiện nay như đặt vé máy bay, khách sạn hay nhà hàng thường hoạt động riêng lẻ, buộc người dùng phải truy cập nhiều nền tảng, tổng hợp thông tin từ nhiều nguồn, dẫn đến mất thời gian và thiếu tiện lợi. Các giải pháp hiện tại bao gồm các ứng dụng đặt dịch vụ riêng biệt hoặc một số nền tảng tổng hợp nhưng chưa tích hợp đầy đủ các dịch vụ như thăm quan, vui chơi, ăn uống, lưu trú, và đặc biệt thiếu khả năng cá nhân hóa lịch trình. Những hạn chế này khiến người dùng gặp khó khăn trong việc lên kế hoạchtron vẹn và tối ưu.

Để giải quyết, em lựa chọn hướng tiếp cận xây dựng một hệ thống tích hợp tất cả các dịch vụ du lịch vào một nền tảng duy nhất, cho phép người dùng tự thiết kế lịch trình cá nhân hóa. Hướng tiếp cận này được chọn vì khả năng cung cấp trải nghiệm liền mạch, giảm thiểu thời gian tìm kiếm và tăng tính tiện lợi thông qua việc so sánh, sắp xếp dịch vụ tự động. Giải pháp của sinh viên là một hệ thống cho phép người dùng chọn các dịch vụ du lịch như thăm quan, vui chơi, nhà hàng, khách sạn; tự động đề xuất lịch trình dựa trên sở thích cá nhân; cung cấp mô tả chi tiết lộ trình, cẩm nang du lịch, hiển thị lộ trình trên bản đồ; đồng thời hỗ trợ đặt trước các dịch vụ và theo dõi các dịch vụ đã đặt. Đóng góp chính của đề tài là xây dựng một nền tảng tích hợp toàn diện, giúp người dùng dễ dàng lập kế hoạch du lịch cá nhân hóa mà không cần chuyển đổi giữa nhiều ứng dụng. Kết quả đạt được là một hệ thống hoạt động hiệu quả, cung cấp trải nghiệm người dùng mượt mà, tiết kiệm thời gian, và tăng cường tính linh hoạt trong việc tổ chức chuyến đi.

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

Nguyễn Thị Thùy Dung

ABSTRACT

The demand for self-organized travel is increasing due to the trend of personalized experiences. However, current services such as booking flights, hotels, or restaurants often operate independently, requiring users to access multiple platforms and aggregate information from various sources, which is time-consuming and inconvenient. Existing solutions include standalone booking apps or some aggregated platforms, but they lack comprehensive integration of services like sightseeing, entertainment, dining, and accommodation, and particularly fail to offer personalized itinerary planning. These limitations make it challenging for users to create complete and optimized travel plans.

To address this, I chose to develop an integrated system that consolidates all travel services into a single platform, enabling users to design personalized itineraries. This approach was selected for its ability to provide a seamless experience, minimize search time, and enhance convenience through automated service comparison and sorting. The proposed solution is a system that allows users to select travel services such as sightseeing, entertainment, dining, and hotels; automatically suggests itineraries based on personal preferences; provides detailed itinerary descriptions, travel guides, and map-based route visualization; and supports pre-booking of services and tracking of booked services. The main contribution of this project is the development of a comprehensive platform that enables users to effortlessly plan personalized trips without switching between multiple applications. The result is an efficient system that delivers a smooth user experience, saves time, and enhances flexibility in organizing travel.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	3
1.4 Bố cục đồ án	4
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	6
2.1 Khảo sát hiện trạng	6
2.1.1 Khảo sát dùng/khách hàng	6
2.1.2 Khảo sát hệ thống hiện có	7
2.2 Tổng quan chức năng	8
2.2.1 Biểu đồ use case tổng quát	8
2.2.2 Biểu đồ use case phân rã.....	10
2.2.3 Quy trình nghiệp vụ	11
2.3 Đặc tả chức năng	13
2.3.1 Đặc tả use case Xem danh sách dịch vụ.....	13
2.3.2 Đặc tả use case Thiết kế lộ trình du lịch	14
2.3.3 Đặc tả use case Quản lý lộ trình.....	15
2.3.4 Đặc tả use case Quản lý các dịch vụ đã đặt	17
2.4 Yêu cầu phi chức năng	18
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	19
3.1 ReactJs	19
3.2 ExpressJs + NodeJs.....	19
3.3 MySQL	20
3.4 Flask	21

3.5 Sử dụng RAG kết hợp LLM để tự động sinh câu trả lời	22
3.6 Các công nghệ hỗ trợ phát triển và kiểm thử hệ thống	22
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG	25
4.1 Thiết kế kiến trúc.....	25
4.1.1 Lựa chọn kiến trúc phần mềm	25
4.1.2 Thiết kế tổng quan.....	27
4.1.3 Thiết kế chi tiết gói	32
4.2 Thiết kế chi tiết.....	35
4.2.1 Thiết kế giao diện	35
4.2.2 Thiết kế lớp	38
4.2.3 Thiết kế cơ sở dữ liệu	40
4.3 Xây dựng ứng dụng.....	52
4.3.1 Thư viện và công cụ sử dụng	52
4.3.2 Kết quả đạt được	53
4.3.3 Minh họa các chức năng chính	55
4.4 Kiểm thử.....	62
4.4.1 Môi trường kiểm thử	62
4.4.2 Kiểm thử blackbox	63
4.5 Triển khai	64
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	65
5.1 Thu thập dữ liệu hiệu quả nhờ Crawl4AI.....	65
5.1.1 Giới thiệu vấn đề.....	65
5.1.2 Giải pháp	65
5.1.3 Kết quả đạt được	66
5.2 Tích hợp RAG để cung cấp thông tin chính xác hơn từ cẩm nang du lịch....	66
5.2.1 Giới thiệu vấn đề.....	66

5.2.2 Giải pháp	66
5.2.3 Kết quả đạt được	67
5.3 Tích hợp Text-to-Speech với giọng đọc tiếng Việt.....	68
5.3.1 Giới thiệu vấn đề.....	68
5.3.2 Giải pháp	68
5.3.3 Kết quả đạt được	69
5.4 Đề xuất địa điểm gần để tối ưu lộ trình	69
5.4.1 Giới thiệu vấn đề.....	69
5.4.2 Giải pháp	69
5.4.3 Kết quả đạt được	70
5.5 Trực quan hóa lộ trình trên bản đồ.....	71
5.5.1 Giới thiệu vấn đề.....	71
5.5.2 Giải pháp	71
5.5.3 Kết quả đạt được	72
5.6 Thiết kế giao diện responsive cho cả máy tính và điện thoại.....	72
5.6.1 Giới thiệu vấn đề.....	72
5.6.2 Giải pháp	73
5.6.3 Kết quả đạt được	73
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	74
6.1 Kết luận	74
6.2 Hướng phát triển.....	75
TÀI LIỆU THAM KHẢO.....	76

DANH MỤC HÌNH VẼ

Hình 2.1	Nhu cầu du lịch của người dùng	7
Hình 2.2	Usecase tổng quan	9
Hình 2.3	Biểu đồ use case thiết kế lộ trình	10
Hình 2.4	Biểu đồ use case quản lý lộ trình	10
Hình 2.5	Biểu đồ hoạt động chỉnh sửa lộ trình	11
Hình 2.6	Biểu đồ hoạt động quản lý dịch vụ	12
Hình 4.1	Kiến trúc phần mềm	25
Hình 4.2	Biểu đồ gói máy chủ Nodejs	27
Hình 4.3	Biểu đồ gói máy chủ Flask	29
Hình 4.4	Biểu đồ lớp phần giao diện	30
Hình 4.5	Biểu đồ lớp chức năng hiển thị danh sách dịch vụ	32
Hình 4.6	Biểu đồ lớp chức năng thiết kế lộ trình	33
Hình 4.7	Biểu đồ lớp chức năng quản lý lộ trình	34
Hình 4.8	Mockup bố cục trang cơ bản	35
Hình 4.9	Mockup hiển thị danh sách dịch vụ	36
Hình 4.10	Mockup thiết kế lộ trình du lịch	37
Hình 4.11	Biểu đồ luồng cho chức năng thiết kế lộ trình du lịch	38
Hình 4.12	Biểu đồ luồng cho chức năng quản lý lộ trình du lịch	39
Hình 4.13	Biểu đồ luồng cho chức năng thanh toán các dịch vụ trong lộ trình du lịch	40
Hình 4.14	Sơ đồ cơ sở dữ liệu tổng quan	41
Hình 4.15	Giao diện hiển thị danh sách địa điểm tham quan	55
Hình 4.16	Giao diện hiển thị danh sách địa điểm vui chơi	55
Hình 4.17	Giao diện hiển thị danh sách khách sạn	55
Hình 4.18	Giao diện hiển thị danh sách nhà hàng	56
Hình 4.19	Giao diện form nhập thông tin cơ bản của lộ trình	56
Hình 4.20	Giao diện hiển thị dịch vụ hệ thống đề xuất cho lộ trình	57
Hình 4.21	Giao diện hiển thị dịch vụ mà hệ thống đề xuất cho lộ trình	57
Hình 4.22	Giao diện hiển thị dịch vụ và lịch đến người dùng có thể sắp xếp lộ trình	57
Hình 4.23	Giao diện hiển thị thông tin chi tiết của lộ trình	58
Hình 4.24	Giao diện hiển thị mô tả lộ trình	58
Hình 4.25	Giao diện hiển thị cẩm nang của lộ trình	59
Hình 4.26	Giao diện hiển thị lộ trình trên bản đồ	59

Hình 4.27 Giao diện hiển thị đặt dịch vụ với thông tin dịch vụ	60
Hình 4.28 Giao diện hiển thị form nhập thông tin liên lạc khi đặt dịch vụ	60
Hình 4.29 Giao diện hiển thị thông tin cần thiết khi đặt dịch vụ để người dùng xác nhận	61
Hình 4.30 Giao diện hiển thị kết quả đặt dịch vụ thành công	61
Hình 4.31 Giao diện email đặt dịch vụ thành công	62

DANH MỤC BẢNG BIỂU

Bảng 1.1	Những giải pháp hiện tại và hạn chế	2
Bảng 2.2	So sánh các hệ thống đã có	8
Bảng 2.4	Use Case UC001 – Xem danh sách dịch vụ du lịch	13
Bảng 2.6	Use Case UC002 – Thiết kế lộ trình du lịch	14
Bảng 2.8	Use Case UC003 – Quản lý lộ trình	15
Bảng 2.10	Use Case UC004 – Quản lý các dịch vụ đã đặt	17
Bảng 3.1	Bảng so sánh MySQL, PostgreSQL, MongoDB	21
Bảng 4.1	Thông số giao diện	35
Bảng 4.2	Thiết kế chi tiết bảng User	42
Bảng 4.3	Thiết kế chi tiết bảng Sight	43
Bảng 4.4	Thiết kế chi tiết bảng Entertainment	44
Bảng 4.5	Thiết kế chi tiết bảng Hotel	45
Bảng 4.6	Thiết kế chi tiết bảng Room	46
Bảng 4.7	Thiết kế chi tiết bảng Restaurant	47
Bảng 4.8	Thiết kế chi tiết bảng Menus	48
Bảng 4.9	Thiết kế chi tiết bảng Repository	49
Bảng 4.10	Thiết kế chi tiết bảng DemoRepoDetail	50
Bảng 4.11	Thiết kế chi tiết bảng ServicePayment	51
Bảng 4.12	Thiết kế chi tiết bảng Liked	52
Bảng 4.13	Thư viện và công cụ sử dụng	53
Bảng 4.14	Thông tin ứng dụng	54
Bảng 4.15	Môi trường kiểm thử	63
Bảng 4.16	Kiểm thử cho chức năng thiết kế lộ trình du lịch	63
Bảng 4.17	Kiểm thử chức năng quản lý lộ trình du lịch	64
Bảng 4.18	Kiểm thử chức năng thanh toán dịch vụ	64
Bảng 4.19	Triển khai hệ thống	64

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
AI/ML	Trí tuệ nhân tạo / Học máy (Artificial Intelligence / Machine Learning) – phát triển hệ thống có khả năng học hỏi và suy luận như con người
API	Giao diện lập trình ứng dụng (Application Programming Interface) – cho phép các phần mềm giao tiếp với nhau thông qua các quy ước
BE	Phần xử lý logic (Back-End) – truy vấn cơ sở dữ liệu, điều phối dữ liệu cho giao diện
DATN	Đồ án tốt nghiệp
FE	Giao diện người dùng (Front-End) – phần mà người dùng nhìn thấy và tương tác trực tiếp như trình duyệt hoặc ứng dụng
LLM	Mô hình ngôn ngữ lớn (Large Language Model) – được huấn luyện từ lượng dữ liệu văn bản khổng lồ để hiểu và sinh ngôn ngữ tự nhiên
MVC	Mô hình kiến trúc phần mềm (Model - View - Controller) – chia chương trình thành 3 phần: Model, View và Controller
RAG	Phương pháp kết hợp truy xuất dữ liệu và mô hình ngôn ngữ lớn để tạo câu trả lời chính xác và có ngữ cảnh (Retrieval-Augmented Generation)
TPP	Nhà cung cấp dịch vụ bên thứ ba (Third-Party Provider) – thường tích hợp vào hệ thống như thanh toán, xác thực

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong bối cảnh ngành du lịch ngày càng phát triển, đặc biệt tại Việt Nam với lượng khách du lịch nội địa và quốc tế tăng trưởng mạnh mẽ, nhu cầu tự tổ chức các chuyến đi cá nhân hóa đang trở nên phổ biến. Tuy nhiên, nhiều du khách gặp khó khăn trong việc lập kế hoạch du lịch chi tiết, bao gồm việc tìm kiếm thông tin về các dịch vụ như địa điểm tham quan, nhà hàng, khách sạn, hay vui chơi, cũng như sắp xếp các hoạt động này thành một lộ trình hợp lý về thời gian và địa điểm.

Hiện tại, các nền tảng du lịch phổ biến thường chỉ cung cấp thông tin riêng lẻ hoặc các gói tour cố định, thiếu tính linh hoạt để người dùng tự xây dựng lộ trình phù hợp với sở thích cá nhân. Ngoài ra, việc đặt trước dịch vụ, theo dõi lộ trình trên bản đồ, hay chia sẻ kế hoạch với người khác vẫn chưa được tích hợp hiệu quả trên một nền tảng duy nhất.

Từ thực tế trên, bài toán đặt ra là làm sao để phát triển một hệ thống giúp người dùng dễ dàng lập lộ trình du lịch cá nhân, dựa trên việc lựa chọn các dịch vụ du lịch yêu thích, sắp xếp chúng vào một lịch trình cụ thể (ví dụ: tham quan chợ Đà Lạt từ 10h-12h), xem và điều chỉnh được lộ trình. Ngoài ra, nền tảng cần hỗ trợ đặt trước các dịch vụ trong lộ trình (như mua vé tham quan hoặc đặt phòng khách sạn) giúp người dùng có thể dễ dàng quản lý được lộ trình cũng như dễ dàng thực hiện hóa lộ trình.

Nếu vấn đề đó được giải quyết, người dùng có thể tiết kiệm thời gian và công sức trong việc lập kế hoạch du lịch, giúp họ dễ dàng tạo ra các lộ trình phù hợp với sở thích cá nhân, tối ưu hóa thời gian và chi phí. Đồng thời các nhà cung cấp dịch vụ tăng cơ hội tiếp cận khách hàng thông qua việc quảng bá các địa điểm tham quan, nhà hàng, khách sạn, hay dịch vụ vui chơi, đồng thời thúc đẩy doanh thu từ các giao dịch đặt trước trực tuyến.

1.2 Mục tiêu và phạm vi đề tài

Hiện nay, nhu cầu cá nhân hóa trong du lịch ngày càng tăng, đặc biệt với sự phổ biến của các thiết bị di động và internet. Một số sản phẩm nổi bật đã được phát triển để hỗ trợ người dùng trong việc lập kế hoạch du lịch, bao gồm:

- **TripAdvisor:** là nền tảng đánh giá và cung cấp thông tin du lịch toàn cầu với các chức năng nổi bật như cung cấp danh sách chi tiết các dịch vụ du lịch như khách sạn, nhà hàng, địa điểm tham quan kèm theo đánh giá và hình ảnh từ cộng đồng người dùng, hỗ trợ đặt trước một số dịch vụ như phòng khách sạn

hoặc nhà hàng thông qua liên kết với các đối tác nhưng chưa mang tính cá nhân hóa kế hoạch du lịch.

- **Wanderlog:** là nền tảng chuyên về lập kế hoạch du lịch cá nhân với các chức năng nổi bật như cho phép người dùng tạo lộ trình du lịch bằng cách thêm các địa điểm, dịch vụ, và sắp xếp theo thời gian cụ thể, hiển thị lộ trình trên cả lịch và bản đồ, hỗ trợ chia sẻ lộ trình với người khác nhưng chưa hỗ trợ quản lý lộ trình dễ dàng
- **Klook:** là nền tảng tập trung vào dịch vụ du lịch với các chức năng nổi bật như cung cấp danh sách các hoạt động du lịch như vé tham quan, tour, hoặc trải nghiệm vui chơi, với khả năng đặt trước trực tuyến, đề xuất các địa điểm hoặc hoạt động gần kề dựa trên vị trí hoặc lịch sử tìm kiếm nhưng không hỗ trợ lập lộ trình cá nhân

Để có cái nhìn tổng quan hơn về các giải pháp hiện tại cho vấn đề thiết kế lộ trình du lịch cá nhân, dưới đây là bảng tổng quan các nền tảng phổ biến đang có hiện nay:

Tiêu chí	TripAdvisor	Wanderlog	Klook
Tìm kiếm danh sách dịch vụ du lịch	✓	✓	✓
Lập lộ trình du lịch	✗	✓	✗
Hiển thị lộ trình trên lịch và bản đồ	✗	✓	✗
Đặt trước dịch vụ online	✓	✗	✓
Đề xuất địa điểm gần kề	✓	✓	✓
Chỉnh sửa lộ trình linh hoạt	✗	✓	✗

Bảng 1.1: Những giải pháp hiện tại và hạn chế

Từ vấn đề đã được nêu tại mục bảng 1.1 và những vấn đề còn tồn đọng của các giải pháp hiện tại, DATN cần phải giải quyết các vấn đề sau:

- **Thiết kế lộ trình:** Từ nhu cầu muôn cá nhân hóa kế hoạch du lịch theo sở thích, DATN cần đưa ra giải pháp để người dùng tự thiết kế kế hoạch du lịch dựa theo những tiêu chí xác định.
- **Chỉnh sửa lộ trình linh hoạt:** Từ việc thiết kế lộ trình, DATN cần cung cấp giải pháp để người dùng có thể linh hoạt cập nhật lộ trình theo mong muốn một cách nhanh chóng, đơn giản.
- **Đặt trước dịch vụ:** Với mỗi lộ trình, ngoài việc chỉnh sửa, DATN còn cần cho phép người dùng đặt trước các dịch vụ du lịch cần thiết giúp quá trình thực

hiện lộ trình nhanh chóng, dễ dàng hơn.

1.3 Định hướng giải pháp

Trong bối cảnh nhu cầu du lịch ngày càng gia tăng và xu hướng cá nhân hóa trải nghiệm du lịch đang trở nên phổ biến, đồ án hướng tới việc tích hợp đa dạng các dịch vụ du lịch (tham quan, khách sạn, nhà hàng, vui chơi) vào một nền tảng thống nhất. Mục tiêu là xây dựng một hệ thống hỗ trợ người dùng tự thiết kế, chỉnh sửa lộ trình du lịch theo sở thích cá nhân, cũng như có thể chủ động quản lý và theo dõi toàn bộ kế hoạch của mình. Nền tảng này sẽ giúp người dùng tiết kiệm thời gian lập kế hoạch, đồng thời nâng cao trải nghiệm sử dụng thông qua việc cá nhân hóa nội dung một cách linh hoạt và dễ tiếp cận.

Để hiện thực hóa định hướng nêu trên, phần mềm được xây dựng với giao diện người dùng thân thiện sử dụng ReactJS, kết hợp backend linh hoạt với Node.js để quản lý dịch vụ và kết nối dữ liệu, cùng Flask để xử lý các tác vụ liên quan đến trí tuệ nhân tạo. Nền tảng hỗ trợ người dùng thiết kế lộ trình du lịch tùy chỉnh thông qua giao diện trực quan, cho phép lựa chọn địa điểm, thời gian, loại hình dịch vụ, và có thể hiển thị trực tiếp trên bản đồ. Hệ thống còn tích hợp tính năng đề xuất cẩm nang và lời khuyên hành trình bằng cách sử dụng mô hình ngôn ngữ lớn (LLM) kết hợp với phương pháp RAG (Retrieval-Augmented Generation), từ đó cung cấp thông tin tham khảo sát với nhu cầu thực tế. Ngoài ra, người dùng có thể dễ dàng theo dõi và quản lý các dịch vụ đã đặt như khách sạn, vé tham quan, nhà hàng trong cùng một giao diện thống nhất.

Đồ án đã xây dựng và hoàn thiện một phần mềm hỗ trợ lập kế hoạch du lịch cá nhân hóa, cho phép người dùng tự thiết kế và theo dõi hành trình của mình một cách linh hoạt và chủ động. Dưới đây là các điểm nổi bật thể hiện đóng góp chính cũng như kết quả đạt được của đồ án:

- **Tích hợp đa dịch vụ vào một nền tảng thống nhất:** Hệ thống cho phép người dùng lựa chọn và quản lý nhiều loại dịch vụ du lịch như địa điểm tham quan, khách sạn, nhà hàng, vui chơi trong cùng một giao diện. Tính năng này giúp tiết kiệm thời gian và công sức khi không cần truy cập nhiều nền tảng khác nhau.
- **Hướng đến cá nhân hóa lộ trình và trải nghiệm người dùng:** Nền tảng hỗ trợ thiết kế lộ trình theo sở thích cá nhân với khả năng tùy chỉnh chi tiết từng ngày đi, giờ đến, dịch vụ sử dụng. Người dùng có thể chỉnh sửa, bổ sung hoặc loại bỏ các điểm đến, phù hợp với kế hoạch thực tế của mình.
- **Ứng dụng công nghệ hiện đại để nâng cao giá trị sử dụng:** Đồ án kết hợp

nhiều công nghệ tiên tiến như mô hình ngôn ngữ lớn (LLM) và phương pháp truy hồi kết hợp sinh văn bản (RAG) để đề xuất nội dung, gợi ý hành trình hoặc cung cấp cảm nang du lịch phù hợp với ngữ cảnh và nhu cầu thực tế. Điều này giúp cải thiện chất lượng thông tin gợi ý, giảm thời gian tìm kiếm cho người dùng.

- **Hiển thị trực quan lộ trình trên bản đồ:** Giao diện bản đồ tương tác được tích hợp trong hệ thống giúp người dùng dễ dàng hình dung tuyến đường, điểm đến và toàn bộ lịch trình của mình. Người dùng cũng có thể kiểm tra các điểm dừng chân, theo dõi các dịch vụ đã đặt, và cập nhật thông tin theo thời gian thực.
- **Hệ thống ổn định, có tính mở rộng và ứng dụng cao:** Sản phẩm cuối cùng hoạt động ổn định với khả năng xử lý linh hoạt các thay đổi từ người dùng. Cấu trúc hệ thống cũng được thiết kế hướng mở, cho phép dễ dàng phát triển thêm các tính năng trong tương lai như đề xuất tự động, tích hợp đánh giá người dùng, hoặc mở rộng sang phiên bản di động.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày quá trình khảo sát và phân tích yêu cầu hệ thống, tập trung đánh giá các giải pháp hiện tại thông qua việc phân tích ưu điểm, hạn chế và mức độ phù hợp với bài toán đặt ra. Tiếp theo, chương này mô tả hệ thống mục tiêu thông qua các sơ đồ use case, làm rõ chức năng chính và các luồng hoạt động điển hình của người dùng và hệ thống. Ngoài ra, các yêu cầu phi chức năng như hiệu năng, bảo mật và khả năng mở rộng cũng được trình bày nhằm hoàn thiện bức tranh tổng thể về nhu cầu thiết kế hệ thống.

Chương 3 giới thiệu các công nghệ và công cụ được sử dụng trong quá trình xây dựng hệ thống. Chương này mô tả lý do lựa chọn và vai trò của ReactJS trong việc phát triển giao diện người dùng, NodeJS và Flask trong phát triển backend, cùng với cơ sở dữ liệu quan hệ MySQL để lưu trữ dữ liệu. Bên cạnh đó, chương cũng đề cập đến việc ứng dụng mô hình RAG tích hợp LLM nhằm tăng cường khả năng sinh gợi ý và nội dung cho người dùng.

Chương 4 đi sâu vào phần thiết kế và kiến trúc chi tiết của hệ thống. Cấu trúc tổng thể và các thành phần chức năng được trình bày thông qua sơ đồ kiến trúc, kết hợp với thiết kế cơ sở dữ liệu, thiết kế giao diện người dùng và mô hình tổ chức các gói chức năng. Từng phần được mô tả rõ ràng để thể hiện cách các thành phần tương tác và đóng vai trò trong toàn bộ hệ thống, từ đó đảm bảo khả năng triển khai

hiệu quả.

Chương 5 trình bày các giải pháp đã được xây dựng và triển khai trong hệ thống, tập trung vào những đóng góp quan trọng của đồ án. Chương làm rõ cách hệ thống tích hợp được nguồn dữ liệu rộng lớn phục vụ cho việc xây dựng cẩm nang du lịch, cơ chế sinh gợi ý hành trình dựa trên nhu cầu cá nhân hóa, cũng như khả năng hiển thị lộ trình trực quan trên bản đồ. Những tính năng này không chỉ đáp ứng yêu cầu chức năng mà còn nâng cao đáng kể trải nghiệm người dùng trong việc lập kế hoạch và theo dõi chuyến đi.

Chương 6 tổng kết các kết quả đã đạt được trong suốt quá trình thực hiện đồ án. Những điểm mạnh trong thiết kế và triển khai hệ thống được làm rõ, đồng thời chương cũng đề cập đến một số hạn chế chưa thể giải quyết trong phạm vi hiện tại. Từ đó, đề xuất các hướng phát triển tiếp theo nhằm mở rộng tính năng, cải thiện hiệu năng hoặc tăng khả năng cá nhân hóa và thích ứng của hệ thống với nhu cầu thực tế ngày càng đa dạng.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

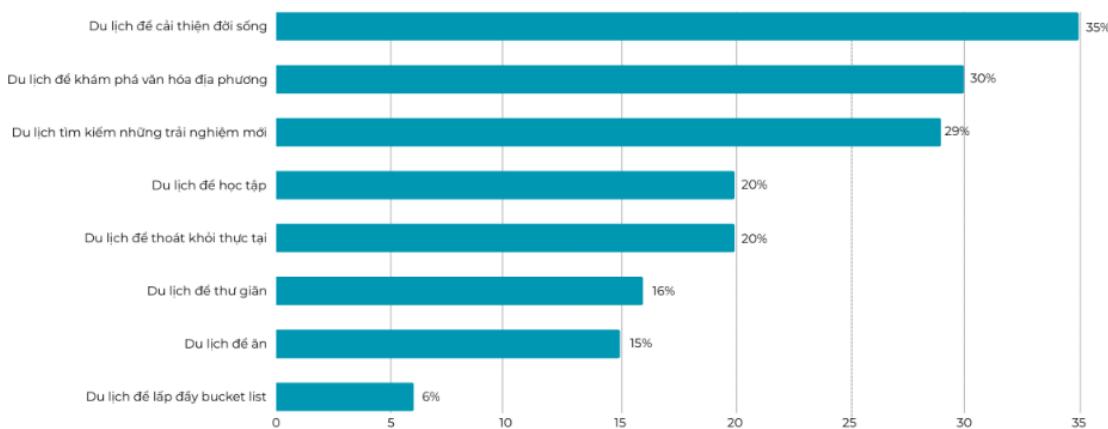
Chương này tập trung khảo sát và phân tích yêu cầu hệ thống, đánh giá các giải pháp hiện tại thông qua việc phân tích ưu, nhược điểm và mức độ phù hợp với bài toán đặt ra. Sau đó, chương trình bày hệ thống mục tiêu bằng các sơ đồ use case, thể hiện rõ chức năng chính và luồng hoạt động giữa người dùng và hệ thống. Cuối cùng, các yêu cầu phi chức năng như hiệu năng, bảo mật và khả năng mở rộng cũng được nêu để làm rõ các tiêu chí thiết kế toàn diện.

2.1 Khảo sát hiện trạng

Để đáp ứng nhu cầu phát triển một hệ thống hỗ trợ du lịch tự túc, việc khảo sát và phân tích các nguồn thông tin từ người dùng, hệ thống hiện có và ứng dụng tương tự là cần thiết để xác định các tính năng quan trọng cần phát triển. Dưới đây là phân tích chi tiết dựa trên các nguồn thông tin được cung cấp.

2.1.1 Khảo sát dùng/khách hàng

Xu hướng du lịch tự túc đang gia tăng mạnh mẽ, phản ánh sự thay đổi trong hành vi du lịch của người tiêu dùng. Theo báo cáo từ Brands Vietnam, từ khóa “du lịch tự túc” trên Google tăng 131% từ năm 2016 đến 2019, cho thấy nhu cầu ngày càng cao về các chuyến đi tự tổ chức. Trên nền tảng Klook, thuật ngữ “du lịch tự túc” chiếm 11% thị trường vào năm 2019, minh chứng cho sự phổ biến của loại hình này. Hashtag *#backpackingtravel* trên Instagram đạt hơn 15 triệu bài đăng, với hơn 222.000 bài từ châu Á, thể hiện sức hút toàn cầu của du lịch tự túc. Khảo sát của Outbox tháng 8/2024 chỉ ra rằng nhóm tuổi 45-54 dẫn đầu với 45,2% tỷ lệ đặt các gói dịch vụ combo và trải nghiệm tham quan, trong khi nhóm 55-65 tuổi cũng tham gia tích cực (22,2%), cho thấy du lịch tự túc không chỉ dành cho giới trẻ mà còn hấp dẫn mọi lứa tuổi. Nhu cầu này nhấn mạnh sự cần thiết của một hệ thống tích hợp, linh hoạt và cá nhân hóa để hỗ trợ người dùng tự lập kế hoạch du lịch.



Hình 2.1: Nhu cầu du lịch của người dùng

2.1.2 Khảo sát hệ thống hiện có

- **Traveloka:** Traveloka là nền tảng du lịch trực tuyến hàng đầu Đông Nam Á, cung cấp dịch vụ đặt vé máy bay, khách sạn, vé vui chơi và các hoạt động trải nghiệm với giao diện thân thiện và tích hợp thanh toán tiện lợi. Ưu điểm bao gồm khả năng tìm kiếm thông tin nhanh chóng, đề xuất điểm đến dựa trên xu hướng và nhiều ưu đãi khuyến mãi. Tuy nhiên, Traveloka chủ yếu tập trung vào đặt dịch vụ riêng lẻ, thiếu tính năng tích hợp để tạo lịch trình cá nhân hóa hoàn chỉnh hoặc hiển thị lộ trình trực quan trên bản đồ.
- **BestPrice:** BestPrice nổi bật với các gói combo du lịch (vé máy bay + khách sạn) và dịch vụ hỗ trợ như tư vấn lịch trình. Ưu điểm là giá cả cạnh tranh và dịch vụ chăm sóc khách hàng tốt, nhưng nhược điểm là thiếu tích hợp các dịch vụ như ăn uống, vui chơi hay đề xuất cẩm nang du lịch thông minh. Hệ thống cũng chưa cung cấp khả năng quản lý lộ trình trực quan hoặc cá nhân hóa sâu.
- **Vietravel:** Vietravel cung cấp các tour du lịch trọn gói và dịch vụ hỗ trợ như đặt vé, khách sạn, vận chuyển, với mạng lưới chi nhánh rộng khắp. Ưu điểm là uy tín thương hiệu và dịch vụ chuyên nghiệp, nhưng nhược điểm là tập trung vào tour truyền thống, ít hỗ trợ du lịch tự túc hoặc cá nhân hóa hành trình. Hệ thống cũng chưa tích hợp công nghệ đề xuất thông minh hay hiển thị lộ trình trực quan.

So sánh chi tiết ưu, nhược điểm của ba hệ thống:

Hệ thống	Ưu điểm	Nhược điểm
Traveloka	Giao diện thân thiện, nhiều ưu đãi, tìm kiếm nhanh, hỗ trợ đa dạng dịch vụ.	Thiếu tích hợp lịch trình cá nhân hóa, không hiển thị lộ trình trên bản đồ.
BestPrice	Giá cạnh tranh, hỗ trợ combo vé + khách sạn, chăm sóc khách hàng tốt.	Thiếu tích hợp dịch vụ vui chơi/ăn uống, không có đề xuất thông minh.
Vietravel	Uy tín, mạng lưới rộng, dịch vụ chuyên nghiệp.	Tập trung tour trọn gói, ít hỗ trợ du lịch tự túc, thiếu công nghệ thông minh.

Bảng 2.2: So sánh các hệ thống đã có

Dựa trên khảo sát người dùng và phân tích các hệ thống hiện có, các tính năng quan trọng của phần mềm cần phát triển bao gồm:

- **Tích hợp dữ liệu du lịch:** Hệ thống cần tích hợp các dịch vụ như vé máy bay, khách sạn, nhà hàng, vui chơi và thăm quan từ nhiều nguồn, cho phép người dùng truy cập và so sánh thông tin trên một nền tảng duy nhất.
- **Thiết kế và cá nhân hóa lộ trình:** Cho phép người dùng tự tạo, chỉnh sửa lịch trình du lịch dựa trên sở thích cá nhân, với gợi ý thông minh sử dụng RAG và LLM để đề xuất cẩm nang du lịch phù hợp.
- **Hiển thị lộ trình trực quan:** Tích hợp bản đồ để hiển thị lộ trình du lịch, giúp người dùng dễ dàng theo dõi và quản lý hành trình.
- **Quản lý dịch vụ đã đặt:** Cung cấp giao diện để người dùng theo dõi và quản lý các dịch vụ đã đặt (khách sạn, nhà hàng, vé tham quan) một cách tập trung và tiện lợi.

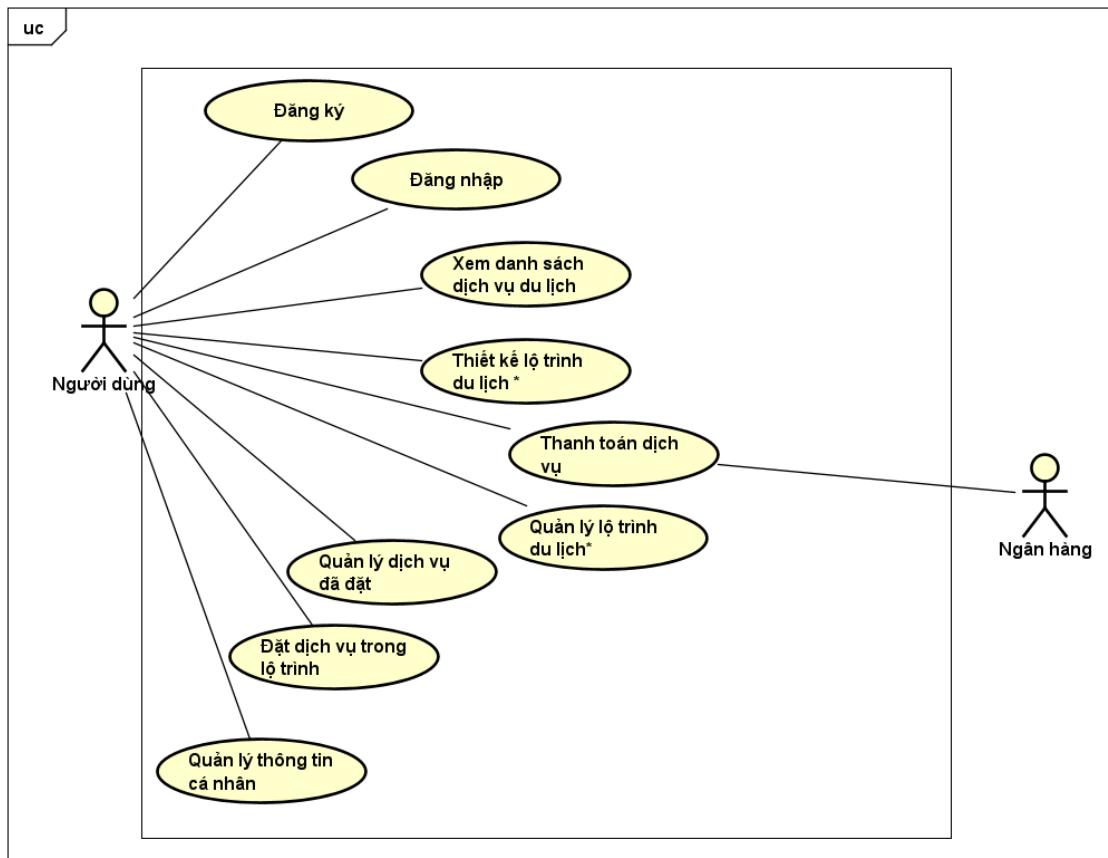
Những tính năng này đáp ứng nhu cầu cá nhân hóa, tiện lợi và trực quan, khắc phục các hạn chế của các hệ thống hiện có và phù hợp với xu hướng du lịch tự túc ngày càng tăng.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát

Để sử dụng hệ thống, người dùng cần đăng ký tài khoản cá nhân hoặc sử dụng tài khoản được cấp bởi quản trị viên. Các giao dịch thanh toán liên quan đến dịch vụ du lịch được tích hợp với hệ thống ngân hàng để đảm bảo an toàn và tiện lợi. Biểu đồ use case tổng quát được trình bày ở Hình 2.2.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU



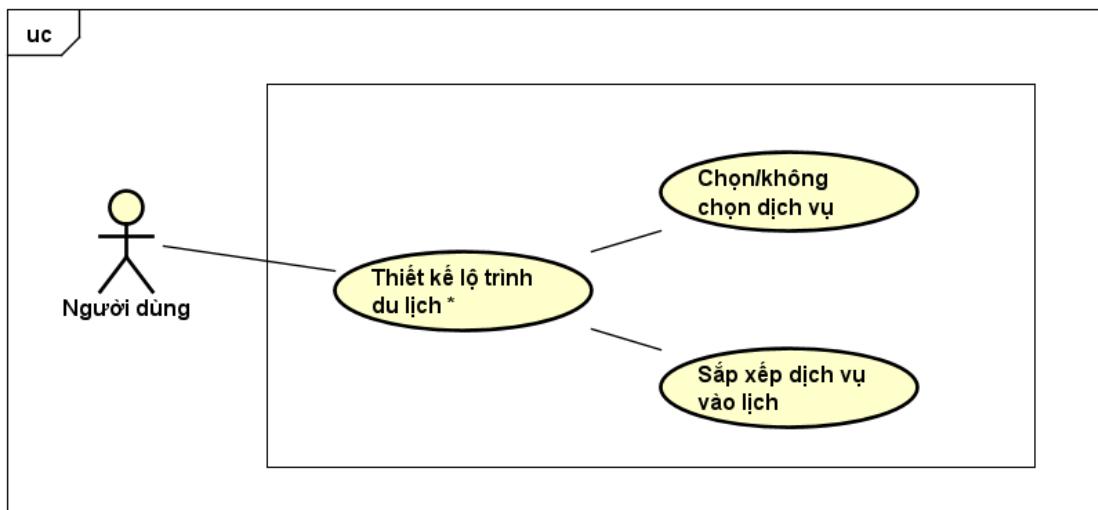
Hình 2.2: Usecase tổng quan

Các tác nhân tham gia vào phần mềm Smart Trip bao gồm:

- **Người dùng:** Là khách du lịch sử dụng hệ thống để thiết kế, chỉnh sửa và quản lý lộ trình du lịch cá nhân hóa, lựa chọn các dịch vụ như vé máy bay, khách sạn, nhà hàng, hoạt động vui chơi, đồng thời theo dõi và quản lý các dịch vụ đã đặt.
- **Ngân hàng:** Là hệ thống bên thứ ba tích hợp với Smart Trip, đảm nhiệm chức năng xử lý và xác nhận các giao dịch thanh toán cho các dịch vụ du lịch mà người dùng đặt qua nền tảng.

2.2.2 Biểu đồ use case phân rã

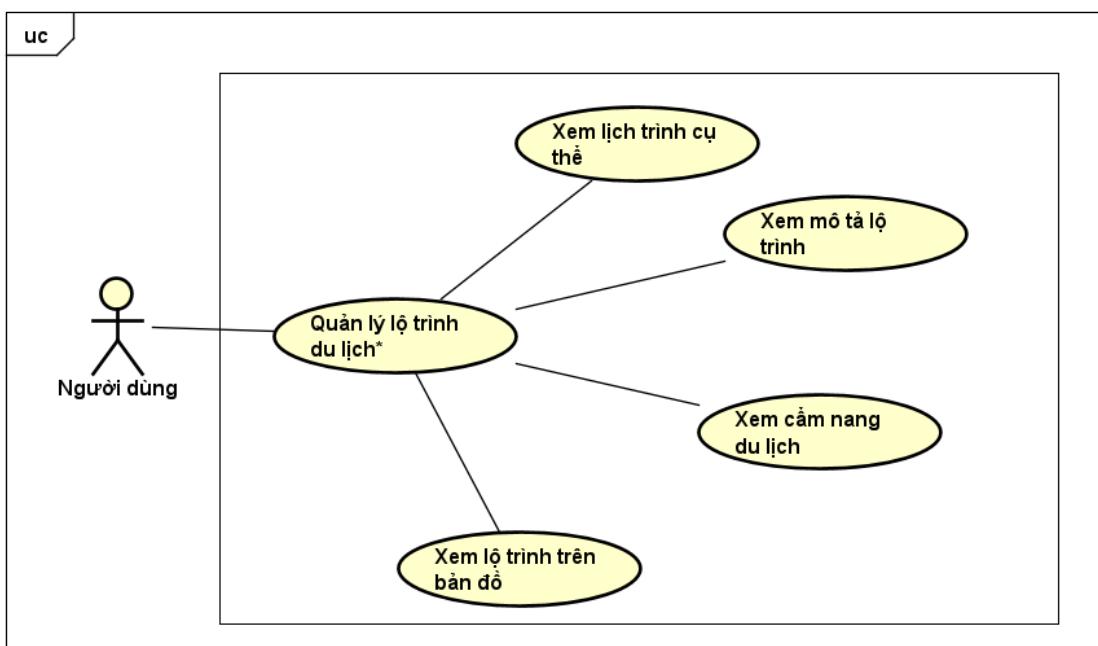
a, Use case thiết kế lộ trình



Hình 2.3: Biểu đồ use case thiết kế lộ trình

Tại biểu đồ 2.3 có thể thấy, người dùng khi thiết kế lộ trình du lịch cá nhân có thể chọn/bỏ chọn dịch vụ (thăm quan, vui chơi, nhà hàng, khách sạn) vào lộ trình và sắp xếp các dịch vụ đã chọn vào lịch trình cá nhân theo sở thích.

b, Use case quản lý lộ trình



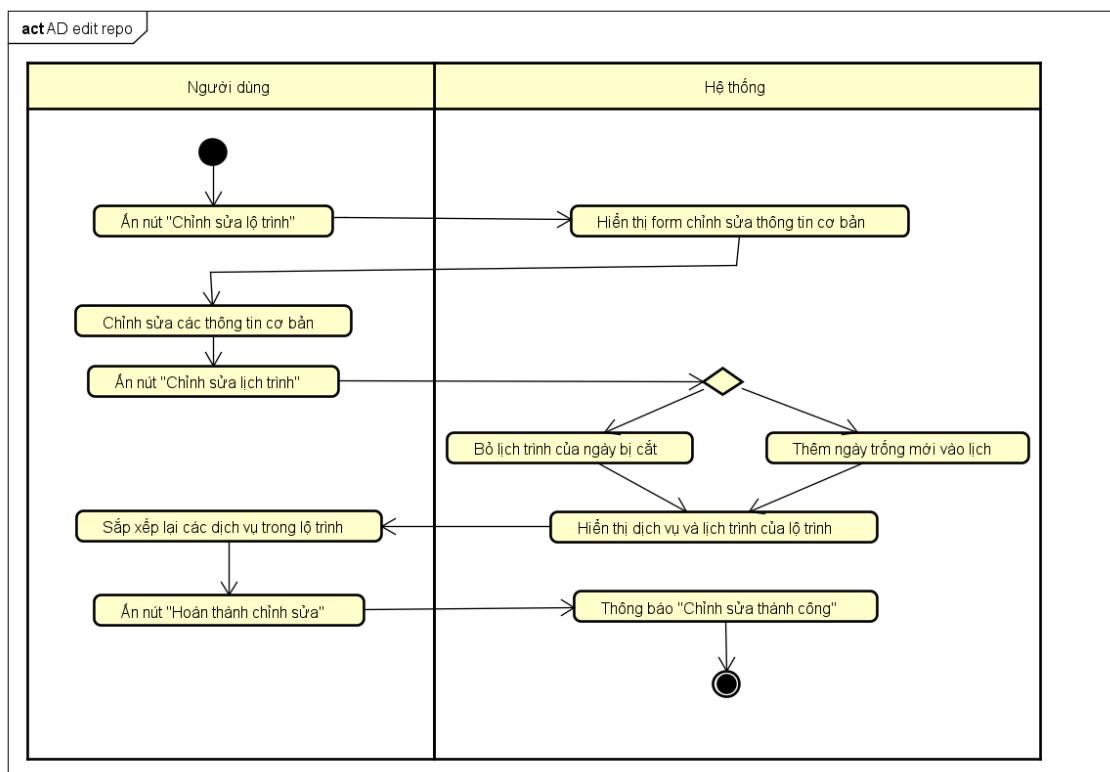
Hình 2.4: Biểu đồ use case quản lý lộ trình

Tại biểu đồ 2.4 có thể thấy, với lộ trình đã tạo, người dùng có thể xem lịch trình cụ thể bao gồm thời gian địa điểm như thế nào, bản mô tả lộ trình chuyên nghiệp như các tour du lịch, xem thông tin cẩm nang du lịch mà hệ thống đề xuất, hay có thể theo dõi trực quan nhất lộ trình trên bản đồ.

2.2.3 Quy trình nghiệp vụ

Để minh họa rõ hơn cách hệ thống hỗ trợ người dùng trong các tình huống thực tế, mục này trình bày một quy trình nghiệp vụ tiêu biểu, kết hợp nhiều chức năng (use case) liên quan nhằm hoàn thành một tác vụ quan trọng trong hệ thống.

a, Biểu đồ hoạt động chỉnh sửa lộ trình du lịch

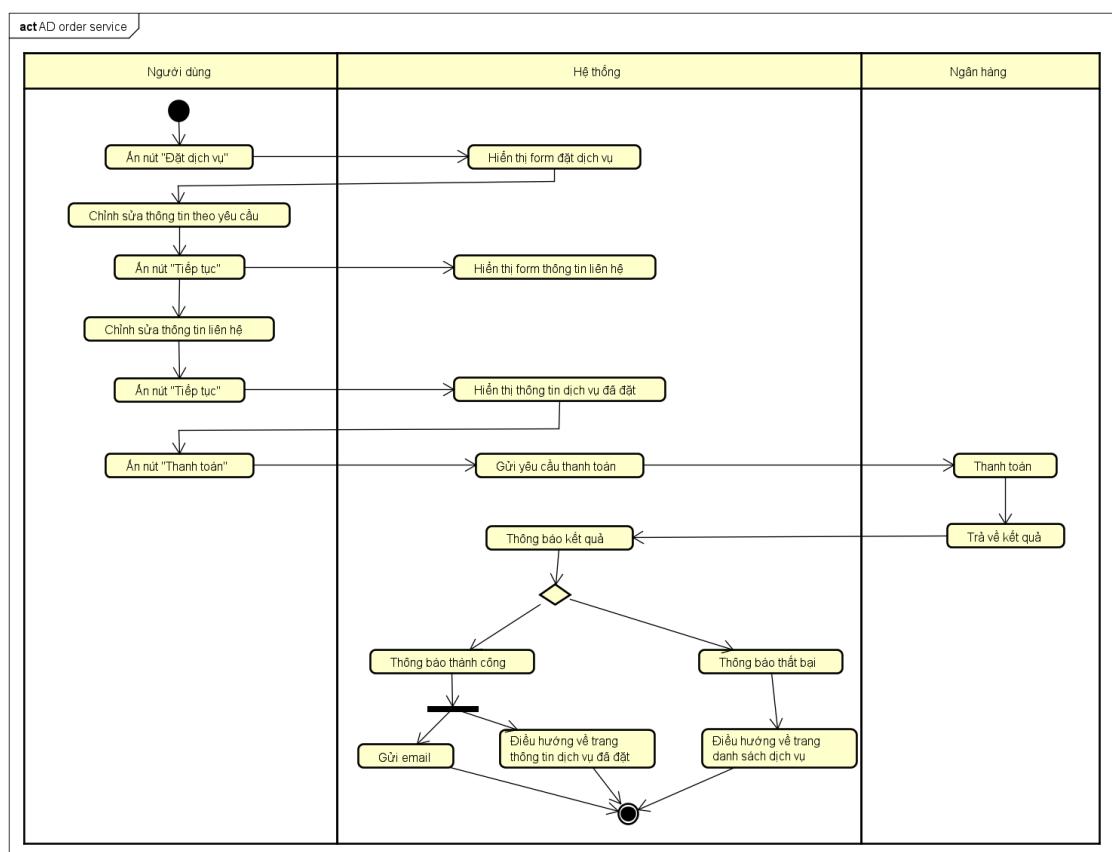


Hình 2.5: Biểu đồ hoạt động chỉnh sửa lộ trình

Biểu đồ 2.5 mô tả quy trình chỉnh sửa lộ trình, cho phép người dùng tinh chỉnh một lộ trình đã được tạo trước đó một cách linh hoạt và chi tiết. Quy trình này bao gồm hai thành phần cốt lõi: chỉnh sửa thông tin cơ bản của lộ trình và tái cấu trúc các dịch vụ đã được sắp xếp. Ở phần chỉnh sửa thông tin cơ bản, người dùng có thể điều chỉnh tên lộ trình, mô tả, số lượng người tham gia, cũng như thời gian bắt đầu và kết thúc, trong khi danh sách các điểm đến được giữ nguyên để đảm bảo tính nhất quán. Nếu thời gian lộ trình được mở rộng, hệ thống sẽ tự động tạo thêm các ngày mới trong lịch trình, cho phép người dùng bổ sung và sắp xếp các dịch vụ phù hợp. Ngược lại, nếu thời gian bị thu hẹp, các ngày và dịch vụ liên quan sẽ được tự

động loại bỏ để phù hợp với khung thời gian mới. Trong trường hợp người dùng không thay đổi thông tin cơ bản, hệ thống sẽ chuyển trực tiếp sang giao diện tái sắp xếp dịch vụ. Tại đây, người dùng có thể thêm hoặc xóa các dịch vụ, đồng thời điều chỉnh khung giờ của từng dịch vụ để tối ưu hóa lịch trình. Khi hoàn tất các chỉnh sửa, người dùng nhấn "Hoàn thành" để lưu lại lộ trình đã được tinh chỉnh, đảm bảo đáp ứng chính xác nhu cầu và mong muốn của họ.

b, Biểu đồ hoạt động quản lý dịch vụ trong lộ trình



Hình 2.6: Biểu đồ hoạt động quản lý dịch vụ

Biểu đồ 2.6 mô tả quy trình đặt dịch vụ trong lộ trình, bao gồm các dịch vụ như vé vui chơi, phòng khách sạn, và đặt trước bàn ăn, được thiết kế để đảm bảo trải nghiệm liền mạch và hiệu quả. Từ một lộ trình đã xác định, hệ thống hiển thị danh sách các dịch vụ khả dụng để người dùng lựa chọn đặt trước. Đối với các dịch vụ như vé vui chơi hoặc phòng khách sạn, người dùng có thể xem chi tiết thông tin bao gồm tên dịch vụ, địa điểm, thời gian (mặc định khớp với khung thời gian của lộ trình), và số lượng người tham gia (cho phép chỉnh sửa linh hoạt). Tiếp theo, người dùng nhập thông tin liên hệ để hoàn tất hồ sơ đặt dịch vụ. Hệ thống sau đó hiển thị tóm tắt thông tin quan trọng, bao gồm chi tiết dịch vụ và thông tin liên hệ, để người dùng xác nhận. Sau khi xác nhận, người dùng được chuyển hướng đến cổng thanh

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

toán VNPay, nơi họ nhập thông tin thanh toán. Kết quả thanh toán từ ngân hàng sẽ được trả về hệ thống và hiển thị trên màn hình. Nếu thanh toán thành công, người dùng nhận được thông tin chi tiết về dịch vụ đã đặt kèm mã QR để nhân viên xác nhận. Trong trường hợp thanh toán thất bại, hệ thống cho phép người dùng thực hiện lại quy trình đặt dịch vụ, đảm bảo tính thuận tiện và liên tục.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Xem danh sách dịch vụ

Mã	UC001	Tên	Xem danh sách dịch vụ du lịch
Tác nhân	Người dùng		
Tiền điều kiện	Không		
Luồng sự kiện	STT	Thực hành	Hành động
	1.1	Tác nhân	Truy cập hệ thống
	1.2	Hệ thống	Hiển thị danh sách dịch vụ (tham quan, vui chơi, khách sạn, nhà hàng)
	1.3	Tác nhân	Tìm kiếm theo tên địa điểm
	1.4	Hệ thống	Hiển thị danh sách dịch vụ có tên trùng hoặc chứa ký tự mà người dùng nhập
	1.5	Tác nhân	Lọc theo các tiêu chí: tỉnh/thành phố, yêu thích hay không, sắp xếp tăng dần/giảm dần đánh giá
	1.6	Hệ thống	Hiển thị danh sách theo tiêu chí đã lọc
	1.7	Tác nhân	Ấn xem chi tiết dịch vụ
	1.8	Hệ thống	Hiển thị thông tin chi tiết về dịch vụ
Luồng sự kiện thay thế	STT	Thực hành	Hành động
	1.4a	Hệ thống	Hiển thị không có dịch vụ nào có tên như tìm kiếm
	1.6a	Hệ thống	Hiển thị không có dịch vụ nào thỏa mãn tiêu chí
Hậu điều kiện	Không		

Bảng 2.4: Use Case UC001 – Xem danh sách dịch vụ du lịch

2.3.2 Đặc tả use case Thiết kế lộ trình du lịch

Mã	UC002	Tên	Thiết kế lộ trình du lịch
Tác nhân	Người dùng		
Tiền kiện	Đăng nhập vào hệ thống		
Luồng sự kiện	STT	Thực hành	Hành động
	1.1	Tác nhân	Ấn nút "Tạo lộ trình"
	1.2	Hệ thống	Hiển thị form thông tin cơ bản của lộ trình
	1.3	Tác nhân	Điền các thông tin cơ bản
	1.4	Tác nhân	Ấn nút "Tiếp tục"
	1.5	Hệ thống	Hiển thị danh sách các địa điểm dựa vào thông tin cơ bản của lộ trình cho người dùng chọn
	1.6	Tác nhân	Chọn/Loại các địa điểm
	1.7	Tác nhân	Ấn nút "Sắp xếp lộ trình"
	1.8	Hệ thống	Hiển thị các dịch vụ đã chọn và lịch để sắp xếp dịch vụ vào lịch
	1.9	Tác nhân	Sắp xếp các dịch vụ vào các khung giờ trong lịch
	1.10	Tác nhân	Lưu tạm thời lịch
	1.11	Hệ thống	Lưu dữ liệu lộ trình tạm thời
	1.12	Tác nhân	Ấn nút "Thêm dịch vụ"
	1.13	Hệ thống	Hiển thị lại danh sách dịch vụ để người dùng lựa chọn thêm
	1.14	Tác nhân	Chọn thêm dịch vụ
Luồng sự kiện thay thế	1.15	Tác nhân	Sắp xếp lịch và ấn nút "Tạo lộ trình"
	1.16	Hệ thống	Hiển thị thông báo Tạo lộ trình thành công
Hậu điều kiện	STT	Thực hành	Hành động
	1.5a	Hệ thống	Hiển thị lỗi tạo thông tin cơ bản như trùng tên lộ trình, số người không hợp lệ
	1.16a	Hệ thống	Hiển thị "Tạo lộ trình không thành công"
Hậu điều kiện	Không		

Bảng 2.6: Use Case UC002 – Thiết kế lộ trình du lịch

2.3.3 Đặc tả use case Quản lý lộ trình

Mã	UC003	Tên	Quản lý lộ trình
Tác nhân		Người dùng	
Tiền điều kiện		Sau khi tạo lộ trình thành công	
Luồng sự kiện	STT	Thực hành	Hành động
	1.1	Tác nhân	Nhấn vào lộ trình vừa tạo
	1.2	Hệ thống	Hiển thị thông tin chi tiết lộ trình: tên lộ trình, timeline có trong lộ trình
	1.3	Tác nhân	Ấn nút xem mô tả lộ trình
	1.4	Hệ thống	Hiển thị mô tả lộ trình như mô tả một tour du lịch
	1.5	Tác nhân	Ấn nút "In mô tả"
	1.6	Hệ thống	Lưu vào hệ thống tệp trên máy của tác nhân với tên tệp là <i>schedule.docx</i>
	1.7	Tác nhân	Ấn xem cẩm nang của lộ trình
	1.8	Hệ thống	Hiển thị cẩm nang lấy từ LLM kết hợp với dữ liệu cẩm nang có được từ hệ thống
	1.9	Tác nhân	Ấn nút "Đọc cẩm nang"
	1.10	Hệ thống	Phát âm thanh đọc cẩm nang với giọng đọc nữ tốc độ 1.0
	1.11	Tác nhân	Ấn nút "Xem lộ trình trên bản đồ"
	1.12	Hệ thống	Hiển thị bản đồ kèm các marker đánh dấu các địa điểm của lộ trình kèm theo tính toán thời gian khoảng cách
	1.13	Tác nhân	Ấn nút "Dịch vụ của lộ trình" (use-case này sẽ được mô tả sau)
Luồng sự kiện thay thế	STT	Thực hành	Hành động
	1.8a	Hệ thống	Thông báo hiển thị lỗi hiển thị trả lời từ LLM
	1.10a	Hệ thống	Thông báo hiển thị lỗi phát audio
	1.12a	Hệ thống	Thông báo lỗi không hiển thị được danh sách địa điểm trên bản đồ
Hậu điều kiện	Không		

Bảng 2.8: Use Case UC003 – Quản lý lộ trình

2.3.4 Đặc tả use case Quản lý các dịch vụ đã đặt

Mã	UC001	Tên	Quản lý các dịch vụ đã đặt
Tác nhân		Người dùng	
Tiền điều kiện		Sau khi tạo lộ trình thành công	
Luồng sự kiện	STT	Thực hành	Hành động
	1.1	Tác nhân	Ấn nút "Dịch vụ" của lộ trình
	1.2	Hệ thống	Hiển thị các dịch vụ của lộ trình: vé vui chơi, phòng khách sạn, đặt bàn trước
	1.3	Tác nhân	Ấn nút "Đặt vé"
	1.4	Hệ thống	Hiển thị thông tin vé vui chơi kèm theo ngày đặt vé chính là ngày tương ứng của dịch vụ đó trong lộ trình
	1.5	Tác nhân	Điều chỉnh các thông tin như số vé nếu cần
	1.6	Tác nhân	Ấn nút "Tiếp tục"
	1.7	Hệ thống	Hiển thị form thông tin liên hệ đặt vé
	1.8	Tác nhân	Điền form với các thông tin: tên, số điện thoại, email
	1.9	Tác nhân	Ấn nút "Tiếp tục"
	1.10	Hệ thống	Hiển thị tất cả các thông tin quan trọng như thông tin vé, thông tin liên hệ
	1.11	Tác nhân	Ấn nút "Thanh toán"
	1.12	Hệ thống	Chuyển người dùng sang hệ thống thanh toán
Luồng sự kiện thay thế	1.13	Hệ thống	Thông báo "Thanh toán thành công"
	1.14	Hệ thống	Chuyển người dùng về xem chi tiết vé đã đặt kèm kèm mã QR của vé
	1.9a	Hệ thống	Thông báo người dùng nhập thông tin không đúng định dạng như email, số điện thoại
	1.13a	Hệ thống	Thông báo "Thanh toán thất bại"
	1.14a	Hệ thống	Chuyển người dùng về danh sách dịch vụ và cho phép người dùng đặt lại vé
Hậu điều kiện	Không		

Bảng 2.10: Use Case UC004 – Quản lý các dịch vụ đã đặt

2.4 Yêu cầu phi chức năng

Hệ thống không chỉ đáp ứng các yêu cầu về mặt chức năng mà còn mang lại những yêu cầu phi chức năng quan trọng, bao gồm tính bảo mật, tính linh hoạt, phản hồi nhanh, hiệu suất cao, độ tin cậy, cũng như khả năng tăng cường tương tác người dùng. Mỗi khía cạnh này được thiết kế cẩn thận để đảm bảo trải nghiệm tối ưu và sự phát triển bền vững của hệ thống.

- **Bảo mật:** Hệ thống áp dụng các biện pháp bảo mật nghiêm ngặt, trong đó chỉ cho phép người dùng đã đăng nhập chính thức mới có quyền sửa đổi hoặc xóa các lô trình. Cơ chế này yêu cầu xác thực danh tính thông qua quy trình đăng nhập an toàn, sử dụng mã hóa dữ liệu và các giao thức bảo mật hiện đại. Điều này không chỉ bảo vệ thông tin cá nhân của người dùng mà còn ngăn chặn các hành vi không mong muốn, đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể thực hiện các thao tác quan trọng, từ đó nâng cao độ an toàn và uy tín của hệ thống.
- **Tính linh hoạt:** Hệ thống được thiết kế với khả năng mở rộng vượt trội, cho phép tùy chỉnh giao diện linh hoạt trên nhiều loại thiết bị khác nhau như máy tính bảng, điện thoại thông minh. Giao diện được tối ưu hóa để tự động điều chỉnh bối cảnh, đảm bảo trải nghiệm người dùng mượt mà bất kể kích thước màn hình. Ngoài ra, mã nguồn được tổ chức thành các mô-đun riêng biệt, rõ ràng, giúp việc nâng cấp, mở rộng chức năng hoặc bảo trì trở nên dễ dàng hơn. Sự linh hoạt này không chỉ hỗ trợ sự phát triển lâu dài của hệ thống mà còn tạo điều kiện thuận lợi cho việc tích hợp các tính năng mới trong tương lai.
- **Phản hồi nhanh, hiệu suất cao:** Hệ thống tận dụng công nghệ tiên tiến như ReactJs và ExpressJs – hai framework nổi tiếng với hiệu suất vượt trội – để đảm bảo hoạt động mượt mà và phản hồi nhanh chóng. Giao diện người dùng được tối ưu hóa để tải dữ liệu gần như tức thời, giảm thiểu thời gian chờ đợi ngay cả khi xử lý lượng lớn thông tin. Hiệu suất cao này không chỉ cải thiện trải nghiệm người dùng mà còn giúp hệ thống duy trì hoạt động ổn định dưới áp lực truy cập lớn, mang lại sự hài lòng tối đa cho người sử dụng.

Với sự kết hợp hài hòa giữa bảo mật, linh hoạt, hiệu suất cao, độ tin cậy, và tương tác người dùng, hệ thống không chỉ đáp ứng nhu cầu hiện tại mà còn đặt nền móng cho sự phát triển bền vững và mở rộng trong tương lai, mang lại trải nghiệm vượt trội so với các giải pháp tương tự.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Chương này giới thiệu các công nghệ và công cụ sử dụng trong xây dựng hệ thống, bao gồm ReactJS để phát triển giao diện người dùng thân thiện, NodeJS và Flask cho backend linh hoạt, MySQL quản lý dữ liệu hiệu quả, cùng RAG và LLM để tạo gợi ý thông minh. Ngoài ra, Git và Github được sử dụng để hỗ trợ quản lý mã nguồn và triển khai, đảm bảo quá trình phát triển mượt mà.

3.1 ReactJs

ReactJS là thư viện JavaScript mã nguồn mở được phát triển bởi Facebook được sử dụng để xây dựng giao diện người dùng (UI) cho các ứng dụng web một cách linh hoạt, tương tác cao và dễ mở rộng. Trong đồ án này, ReactJS được sử dụng để thiết kế giao diện chính cho người dùng thiết kế và quản lý lộ trình, hiển thị thông tin các dịch vụ du lịch, đảm bảo giao diện responsive, hiển thị tốt trên cả desktop và thiết bị di động.

Những điểm nổi bật của ReactJS:

- **Hiệu suất cao:** sử dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện, giảm thiểu thao tác trực tiếp với DOM thật, từ đó tăng tốc độ render UI.
- **Tính linh hoạt và dễ mở rộng:** React chỉ là thư viện View, giúp dễ dàng tích hợp với các công nghệ khác như Redux, GraphQL,... Không ép buộc cấu trúc như Angular, nên phù hợp với nhiều mô hình dự án.
- **Tính tái sử dụng cao:** React cho phép xây dựng ứng dụng dưới dạng component, dễ bảo trì và phát triển mở rộng.
- **Props và Data Binding:** Props cho phép truyền dữ liệu từ component cha sang component con, giúp dễ dàng cấu hình giao diện theo ngữ cảnh sử dụng. Data Binding cho phép đồng bộ hóa dữ liệu trong giao diện với dữ liệu logic bên trong component.
- **Quản lý trạng thái:** React quản lý trạng thái từng component một cách độc lập, không ảnh hưởng đến các phần còn lại giúp ứng dụng phản hồi nhanh theo thay đổi của người dùng như click, nhập dữ liệu, chuyển trang.

3.2 ExpressJs + NodeJs

Node.js là môi trường chạy JavaScript phía server, giúp phát triển backend hiệu quả bằng cùng ngôn ngữ với frontend (JavaScript). Express.js là framework nhẹ của Node.js, hỗ trợ xây dựng RESTful API nhanh chóng và rõ ràng. Trong các ứng dụng web hiện đại, backend đóng vai trò trung gian quan trọng giữa giao diện

người dùng (frontend) và cơ sở dữ liệu hoặc các dịch vụ xử lý logic phức tạp (như AI, lưu trữ, thanh toán, v.v). Backend chịu trách nhiệm xử lý các yêu cầu của người dùng, truy vấn dữ liệu, đảm bảo tính bảo mật và cung cấp các API để frontend giao tiếp.

Những điểm nổi bật của ExpressJs:

- **Hiệu suất cao:** Node.js hoạt động theo mô hình event-driven và non-blocking I/O, phù hợp với ứng dụng cần xử lý đồng thời nhiều request.
- **Express.js đơn giản, rõ ràng:** Cú pháp dễ hiểu, tổ chức code theo từng route giúp cấu trúc backend rõ ràng, dễ mở rộng. Dễ tích hợp với các thư viện middleware như body-parser, cors, helmet,...
- **Hệ sinh thái mạnh:** Có hàng nghìn package hỗ trợ từ npm như: jsonwebtoken, nodemailer, mysql2, multer,...
- **Xử lý bất đồng bộ:** Node.js hỗ trợ cơ chế xử lý bất đồng bộ cho tất cả các API, giúp hệ thống không bị “chặn” khi thực hiện các tác vụ như truy xuất cơ sở dữ liệu hoặc gọi đến dịch vụ bên ngoài. Điều này cho phép ứng dụng có thể tiếp tục xử lý các yêu cầu khác mà không cần chờ đợi tác vụ hiện tại hoàn tất, từ đó nâng cao hiệu suất tổng thể và mang lại trải nghiệm phản hồi nhanh, mượt mà cho người dùng.

3.3 MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở, được phát triển ban đầu bởi công ty MySQL AB và hiện nay được duy trì bởi Oracle Corporation. MySQL sử dụng ngôn ngữ truy vấn có cấu trúc SQL (Structured Query Language) để thao tác và quản lý dữ liệu. Nhờ vào hiệu suất cao, khả năng mở rộng linh hoạt và tính ổn định, MySQL được sử dụng rộng rãi trong nhiều ứng dụng từ nhỏ đến lớn, bao gồm các hệ thống web, hệ thống thương mại điện tử, nền tảng học trực tuyến, và các hệ thống quản lý doanh nghiệp. Với cộng đồng phát triển mạnh mẽ, tài liệu phong phú và khả năng tích hợp tốt với nhiều ngôn ngữ lập trình như PHP, Java, Python hay Node.js, MySQL trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới.

Những điểm nổi bật của MySQL:

- **Tính phổ biến và dễ sử dụng:** MySQL có cú pháp SQL quen thuộc, phù hợp với hầu hết các nhà phát triển, đặc biệt là trong các dự án vừa và nhỏ.
- **Hiệu năng cao:** Khả năng xử lý truy vấn nhanh, ổn định, đáp ứng tốt cho ứng dụng có lượng người dùng trung bình đến lớn.

- Tương thích cao:** Dễ dàng tích hợp với nhiều ngôn ngữ lập trình như Node.js, Python và công cụ ORM như Sequelize.

Tiêu chí	MySQL	PostgreSQL	MongoDB
Mô hình dữ liệu	Quan hệ	Quan hệ	No SQL
Hiệu năng với dữ liệu vừa	Tốt	Khá tốt	Phụ thuộc vào trường hợp cụ thể
Hệ sinh thái và cộng đồng	Rộng, ổn định	Lớn, thiên về học thuật	Đang phát triển rất nhanh
Tạo bảng và quản lý cấu trúc	Đơn giản, dễ tiếp cận	Linh hoạt nhưng phức tạp hơn	Không cần tạo bảng

Bảng 3.1: Bảng so sánh MySQL, PostgreSQL, MongoDB

3.4 Flask

Flask là một micro web framework mã nguồn mở được viết bằng ngôn ngữ lập trình Python, ra đời năm 2010 bởi Armin Ronacher. Mặc dù là “micro”, Flask không thiếu chức năng mà chỉ tập trung vào việc cung cấp những thành phần cốt lõi cần thiết để xây dựng ứng dụng web. Nó cho phép lập trình viên mở rộng và tùy chỉnh hệ thống dễ dàng bằng cách thêm các thành phần khi cần thiết. Với thiết kế nhẹ, linh hoạt và dễ tích hợp, Flask rất phù hợp cho các ứng dụng nhỏ đến trung bình hoặc những hệ thống yêu cầu tính tùy chỉnh cao như viết API cho hệ thống AI/LLM, xử lý dữ liệu, tích hợp RAG (Retrieval-Augmented Generation) như trong đồ án.

Những điểm nổi bật của Flask:

- Thiết kế tối giản – micro framework:** Flask không áp đặt cấu trúc cứng nhắc, cho phép người lập trình tự do tổ chức dự án theo cách phù hợp nhất với yêu cầu của mình.
- Khả năng mở rộng cao:** Flask có thể dễ dàng mở rộng bằng cách sử dụng các extension như Flask-RESTful (xây dựng REST API), Flask-SQLAlchemy (ORM), Flask-Cors (CORS middleware), ...
- Hỗ trợ tốt cho API và microservices:** Flask được ưa chuộng trong việc xây dựng RESTful API hoặc hệ thống microservices nhờ tốc độ nhẹ, cấu trúc rõ ràng và dễ tích hợp với các công nghệ AI (như OpenAI, Hugging Face, LangChain...).
- Tích hợp tốt với AI/ML:** Do được viết bằng Python – ngôn ngữ chính của cộng đồng học máy – Flask dễ dàng tích hợp với các thư viện như scikit-

learn, TensorFlow, PyTorch, hay các framework triển khai AI như LangChain, Transformers, ...

3.5 Sử dụng RAG kết hợp LLM để tự động sinh câu trả lời

Retrieval-Augmented Generation (RAG) là một kỹ thuật kết hợp việc truy vấn và sinh dữ liệu, cho phép mô hình ngôn ngữ như LLM (Large Language Model) tự động sinh câu trả lời dựa trên thông tin có sẵn trong cơ sở dữ liệu hoặc từ các nguồn truy vấn ngoài (như web hoặc tài liệu). Khi kết hợp với các mô hình ngôn ngữ như GPT, RAG giúp cải thiện chất lượng và độ chính xác của câu trả lời bằng cách cung cấp một phương pháp truy xuất thông tin linh hoạt trước khi thực hiện việc sinh ngữ nghĩa.

Những điểm nổi bật của việc sử dụng RAG kết hợp LLM:

- **Kết hợp sức mạnh của truy vấn và sinh dữ liệu:** RAG giúp mô hình ngôn ngữ không chỉ dựa vào thông tin có sẵn trong bản thân mô hình mà còn sử dụng các truy vấn từ cơ sở dữ liệu hoặc các nguồn ngoài để cải thiện độ chính xác của câu trả lời.
- **Tính linh hoạt cao:** Việc tích hợp RAG cho phép người phát triển ứng dụng linh hoạt trong việc sử dụng nhiều loại nguồn thông tin khác nhau, từ cơ sở dữ liệu văn bản, API đến các nguồn web, giúp hệ thống luôn cập nhật và mở rộng dữ liệu một cách hiệu quả.
- **Cải thiện độ chính xác của câu trả lời:** Với khả năng truy xuất và kết hợp thông tin từ các nguồn bên ngoài, RAG giúp giảm thiểu sự phụ thuộc vào kiến thức cố định của mô hình ngôn ngữ, từ đó mang lại câu trả lời chính xác hơn và phù hợp hơn với bối cảnh hiện tại.
- **Tích hợp dễ dàng với các hệ thống AI/LLM:** Việc sử dụng RAG cho phép dễ dàng kết hợp với các hệ thống học máy như scikit-learn, TensorFlow, PyTorch và các mô hình ngôn ngữ như GPT hay BERT. Việc này đặc biệt hữu ích trong các ứng dụng yêu cầu thông tin luôn được cập nhật và có khả năng mở rộng như các chatbot hỗ trợ khách hàng, hệ thống đề xuất thông tin, và trợ lý ảo.
- **Quá trình tối ưu hóa rõ ràng:** Quy trình RAG giúp tối ưu hóa việc kết hợp các mô hình truy vấn và sinh dữ liệu, tạo ra những câu trả lời từ sự kết hợp giữa prompt được cung cấp và thông tin truy vấn, từ đó tăng khả năng tự động hóa và tính chính xác trong các ứng dụng yêu cầu trả lời thông tin phức tạp.

3.6 Các công nghệ hỗ trợ phát triển và kiểm thử hệ thống

Trong quá trình phát triển hệ thống, việc kết hợp nhiều công nghệ hỗ trợ là yếu tố quan trọng giúp tăng hiệu suất làm việc, đảm bảo chất lượng sản phẩm và tối ưu

hóa trải nghiệm người dùng. Các công nghệ được sử dụng trong đồ án không chỉ hỗ trợ quá trình lập trình mà còn đóng vai trò quan trọng trong việc thiết kế, kiểm thử và triển khai hệ thống một cách chuyên nghiệp và hiệu quả. Dưới đây là một số công nghệ tiêu biểu:

- **Chuyển văn bản thành giọng nói tiếng Việt (Text-to-Speech - TTS):** Công nghệ TTS giúp hệ thống có thể chuyển đổi nội dung văn bản sang giọng nói tiếng Việt một cách tự nhiên. Điều này đặc biệt hữu ích trong các ứng dụng hỗ trợ người dùng yếu thị giác, hệ thống trợ lý ảo, hoặc các chức năng phản hồi bằng giọng nói. Các API như Google Text-to-Speech hoặc Viettel AI TTS được tích hợp dễ dàng với hệ thống backend để sinh âm thanh theo thời gian thực từ nội dung đầu vào.
- **Thiết kế giao diện bằng Figma:** Figma là công cụ thiết kế giao diện người dùng (UI) phổ biến hiện nay, cho phép làm việc nhóm thời gian thực, thiết kế prototype và chuyển đổi thiết kế sang mã HTML/CSS dễ dàng. Trong đồ án, Figma được sử dụng để xây dựng giao diện mẫu, lên bối cảnh hệ thống, thống nhất phong cách thiết kế và hỗ trợ giao tiếp giữa lập trình viên và người thiết kế, từ đó tăng tính nhất quán của sản phẩm.
- **Kiểm thử API bằng Postman:** Postman là công cụ mạnh mẽ để kiểm tra và xác minh các API trong quá trình phát triển backend. Việc sử dụng Postman giúp đảm bảo các API hoạt động chính xác, phản hồi dữ liệu đúng định dạng, và có khả năng xử lý lỗi hợp lý. Ngoài ra, Postman còn hỗ trợ viết các tập test tự động và chia sẻ bộ API giữa các thành viên trong nhóm phát triển.
- **Quản lý mã nguồn bằng GitHub:** GitHub là nền tảng quản lý mã nguồn phân tán dựa trên Git, giúp các thành viên trong nhóm phát triển có thể làm việc đồng thời, quản lý phiên bản, tạo nhánh (branch), và theo dõi thay đổi dễ dàng. Việc sử dụng GitHub trong đồ án giúp đảm bảo sự phối hợp hiệu quả giữa các thành viên, dễ dàng rollback khi cần và minh bạch quá trình phát triển hệ thống.
- **Sử dụng thư viện giao diện Ant Design:** Ant Design là một thư viện UI chuyên nghiệp dành cho React, cung cấp sẵn các thành phần giao diện đẹp, hiện đại và dễ sử dụng. Nhờ có Ant Design, hệ thống có thể nhanh chóng xây dựng các phần tử như bảng dữ liệu, biểu mẫu, nút bấm, hệ thống layout... mà không cần viết tay từ đầu. Việc này không chỉ tiết kiệm thời gian mà còn giúp giao diện nhất quán, dễ tùy chỉnh và thân thiện với người dùng.
- **Tích hợp Google Maps:** Việc sử dụng Google Maps API cho phép hệ thống hiển thị bản đồ, định vị và hiển thị tọa độ địa lý, cung cấp chức năng tìm đường

và đánh dấu các địa điểm du lịch. Đây là công nghệ rất phù hợp trong các ứng dụng liên quan đến du lịch, giúp người dùng tra cứu vị trí, xem khoảng cách giữa các điểm và tương tác trực tiếp với bản đồ trong thời gian thực.

- **Thiết kế responsive – tương thích đa thiết bị:** Giao diện hệ thống được xây dựng theo hướng responsive, tức là có thể tự động điều chỉnh để hiển thị tốt trên nhiều loại thiết bị khác nhau như điện thoại di động, máy tính bảng và máy tính để bàn. Điều này giúp nâng cao trải nghiệm người dùng, đặc biệt là trong bối cảnh người dùng thường xuyên sử dụng smartphone để tra cứu thông tin du lịch khi đang di chuyển.
- **Gửi email bằng Nodemailer trong Node.js – Gửi thông báo hoặc chia sẻ kết quả qua email:** Nodemailer là một thư viện phổ biến trong Node.js dùng để gửi email từ server. Trong đồ án, Nodemailer có thể được sử dụng để gửi email xác nhận đăng ký, gửi lịch trình du lịch do AI gợi ý cho người dùng, hoặc thông báo các cập nhật liên quan đến hành trình. Với cấu hình đơn giản, hỗ trợ SMTP (Gmail, Outlook...) và khả năng đính kèm nội dung HTML, PDF hoặc hình ảnh, Nodemailer phù hợp cho các ứng dụng cần tương tác qua email mà không phụ thuộc dịch vụ ngoài như Firebase hay SendGrid.

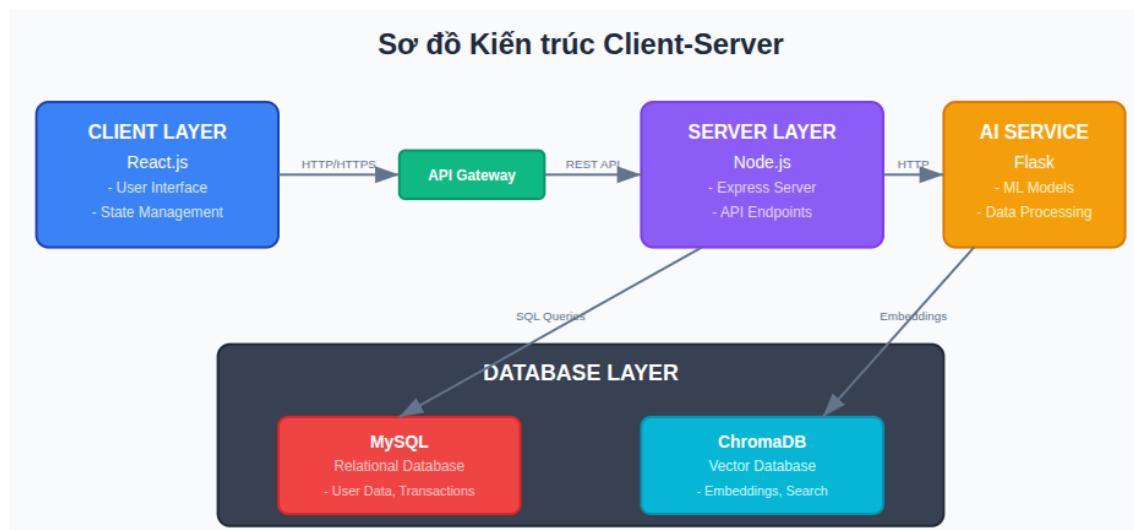
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Chương này trình bày thiết kế và kiến trúc hệ thống, với sơ đồ kiến trúc thể hiện rõ cấu trúc tổng thể, thiết kế cơ sở dữ liệu, giao diện người dùng và tổ chức các gói chức năng. Các thành phần được mô tả ngắn gọn, minh họa cách chúng tương tác để đảm bảo triển khai hiệu quả.

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Hệ thống được thiết kế theo mô hình kiến trúc client-server phân tầng, bao gồm bốn thành phần chính: Client Layer (React.js), Server Layer (Node.js), AI Service (Flask), và Database Layer (MySQL + ChromaDB).



Hình 4.1: Kiến trúc phần mềm

Client Layer được xây dựng trên nền tảng React.js, đóng vai trò là giao diện người dùng chính của hệ thống. Tầng này chịu trách nhiệm cung cấp các components tương tác trực quan, cho phép người dùng nhập thông tin về chuyến du lịch, xem cẩm nang được tạo tự động, và quản lý tài khoản cá nhân. React.js được lựa chọn nhờ khả năng quản lý trạng thái hiệu quả thông qua các hooks và thư viện state management, đồng thời hỗ trợ việc tương tác với API thông qua các HTTP requests đến Server Layer.

Server Layer được phát triển bằng Node.js với framework Express.js, đóng vai trò là trung tâm xử lý logic nghiệp vụ liên quan đến người dùng client. Thành phần này chịu trách nhiệm quản lý authentication, authorization cho toàn bộ hệ thống. Node.js server cung cấp các RESTful endpoints để Client Layer có thể tương tác. Ngoài ra, server còn xử lý các logic nghiệp vụ như validation dữ liệu đầu vào, xử

lý thông tin du lịch, và thiết kế lộ trình du lịch. Server Layer tương tác trực tiếp với MySQL database để lưu trữ và truy xuất các thông tin cần thiết cho hệ thống.

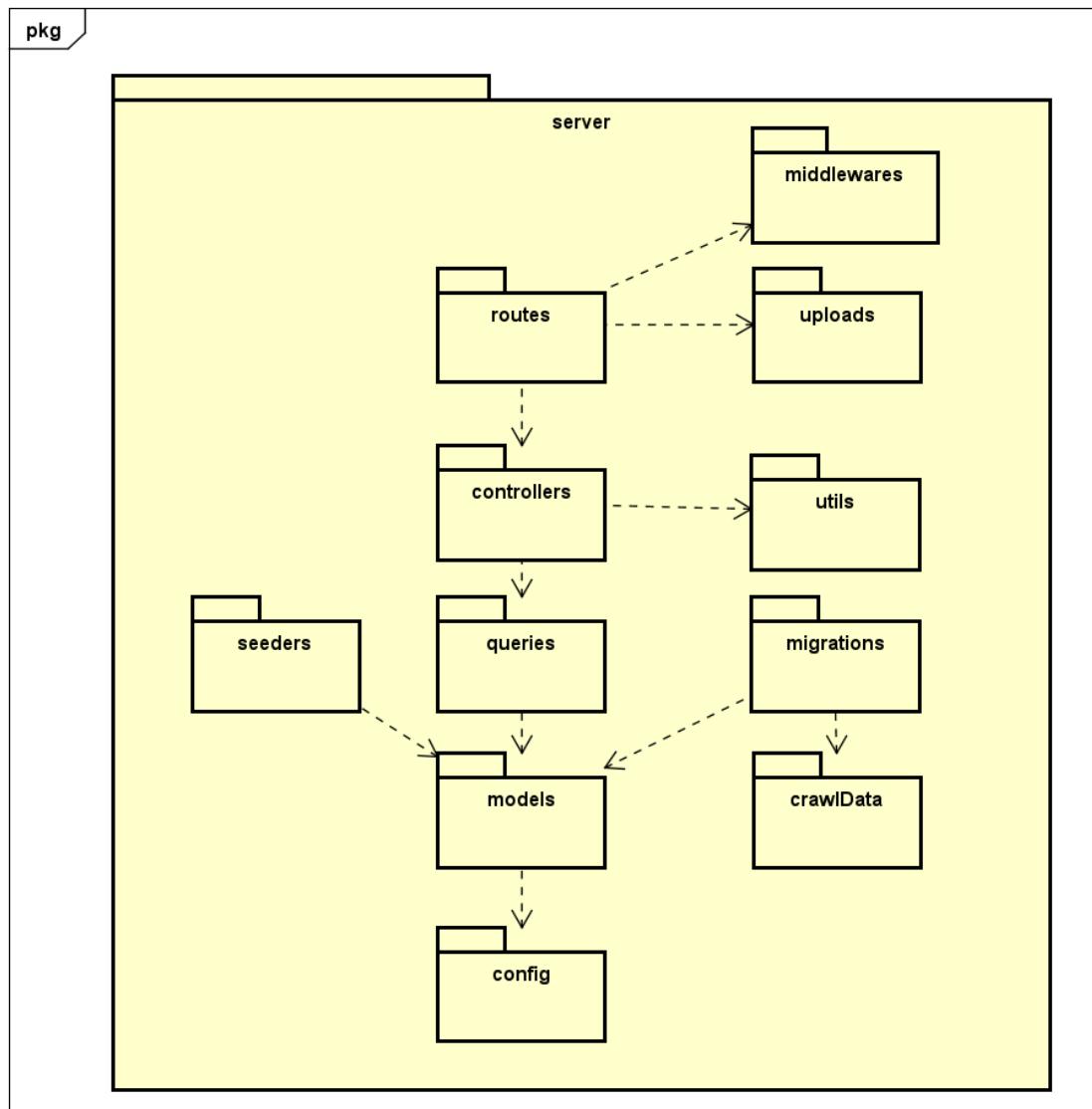
AI Service được xây dựng trên nền tảng Flask (Python), chuyên biệt xử lý các tác vụ liên quan đến việc tích hợp AI RAG LLM để tự động sinh cẩm nang du lịch. Thành phần này triển khai quy trình RAG (Retrieval-Augmented Generation) để kết hợp việc tìm kiếm thông tin có sẵn với khả năng sinh văn bản của LLM, đảm bảo cẩm nang du lịch được tạo ra có tính chính xác và cập nhật. Flask service tích hợp với các mô hình ngôn ngữ lớn để sinh tự động nội dung cẩm nang du lịch dựa trên thông tin đầu vào từ người dùng và dữ liệu được truy xuất từ vector database. AI Service quản lý và tương tác trực tiếp với ChromaDB để lưu trữ, tìm kiếm embeddings của các thông tin du lịch, địa điểm, và kinh nghiệm du lịch, thực hiện các tác vụ semantic search và similarity matching để hỗ trợ RAG pipeline.

Database Layer bao gồm hai hệ thống cơ sở dữ liệu với vai trò bổ trợ cho nhau, tạo nên một giải pháp hybrid storage tối ưu. MySQL Database được sử dụng để lưu trữ dữ liệu có cấu trúc như thông tin người dùng, thông tin lộ trình và thông tin các dịch vụ. ChromaDB (Vector Database) được triển khai để lưu trữ embeddings của các thông tin du lịch, mô tả địa điểm, kinh nghiệm du lịch dưới dạng vectors, hỗ trợ tìm kiếm ngữ nghĩa để tìm ra các thông tin liên quan nhất cho việc tạo cẩm nang và thực hiện việc so sánh độ tương đồng giữa các vectors để hỗ trợ RAG pipeline.

Giao tiếp giữa các thành phần trong hệ thống được thiết kế để đảm bảo hiệu suất và độ tin cậy. Client-Server communication sử dụng HTTP/HTTPS protocols với RESTful API, data exchange thông qua JSON format. Server-AI Service communication thực hiện qua HTTP requests, cho phép Node.js server gọi các endpoints của Flask service một cách linh hoạt. Đối với database access, Node.js sử dụng SQL queries để tương tác với MySQL, trong khi Flask service sử dụng vector queries để tương tác với ChromaDB. Kiến trúc này mang lại nhiều ưu điểm đáng kể cho hệ thống.

4.1.2 Thiết kế tổng quan

a, Biểu đồ phụ thuộc gói máy chủ Nodejs



Hình 4.2: Biểu đồ gói máy chủ Nodejs

Hệ thống phía máy chủ trong dự án được thiết kế theo mô hình module hóa với các thành phần chính bao gồm: route, controllers, middlewares, services, utils, models, migrate và seeder. Việc tổ chức này nhằm đảm bảo tính rõ ràng, dễ bảo trì và mở rộng trong quá trình phát triển hệ thống.

Thành phần **Route** chịu trách nhiệm định nghĩa các endpoint liên quan đến người dùng, đồng thời chuyển tiếp các yêu cầu đến các **controller** tương ứng. Các controller tiếp nhận yêu cầu, xử lý logic ban đầu và điều hướng tác vụ đến các **service** để thực hiện các chức năng nghiệp vụ chính. Trước khi đến được controller, các yêu cầu sẽ được đi qua **middlewares** – nơi thực hiện các tác vụ như xác thực người dùng, kiểm tra quyền truy cập và làm sạch dữ liệu đầu vào, đảm bảo an toàn

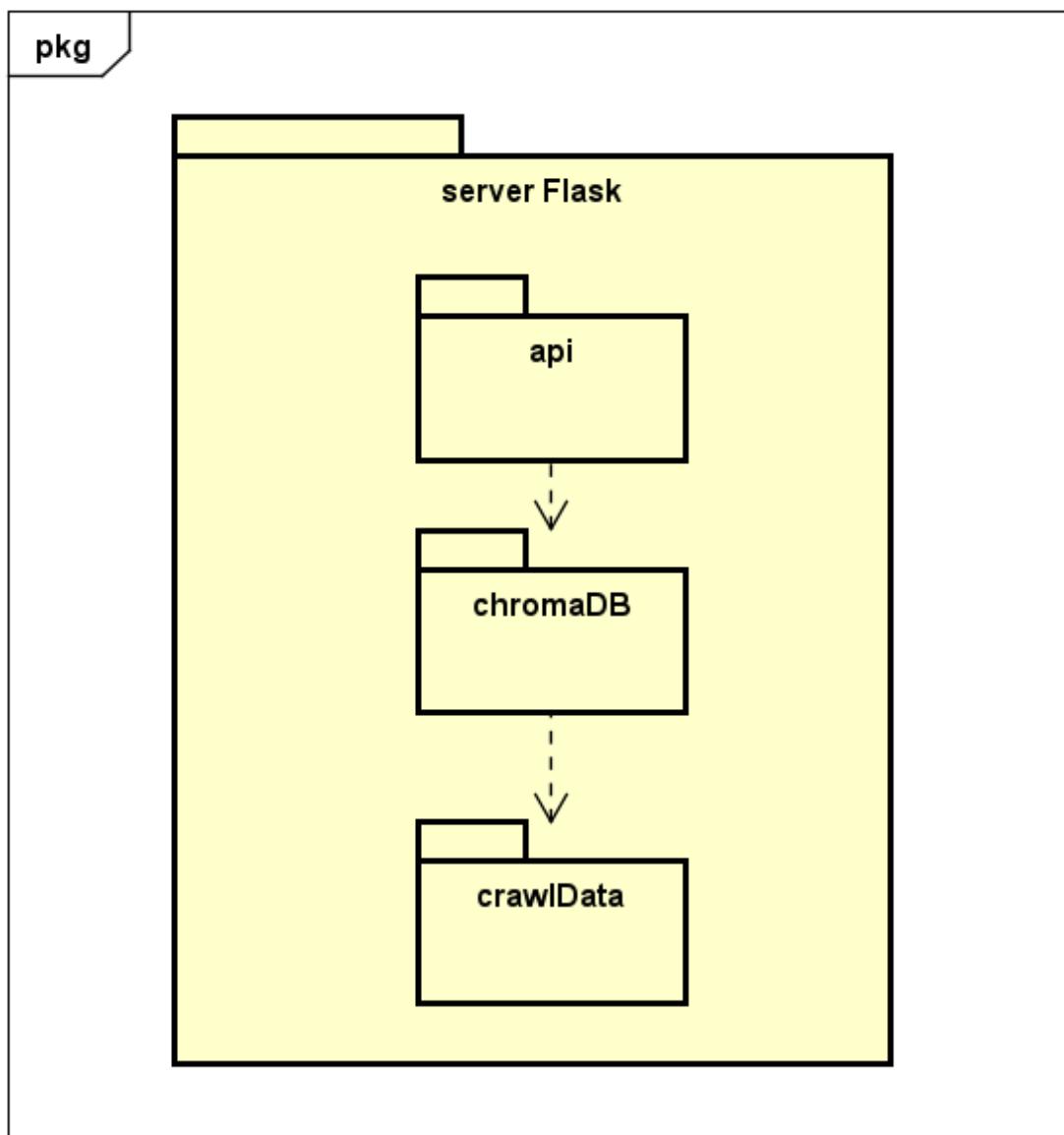
và hợp lệ cho toàn bộ hệ thống.

Services là tầng xử lý logic chính, nơi triển khai các thao tác nghiệp vụ và kết nối với models để thực hiện các truy vấn cơ sở dữ liệu. Models định nghĩa cấu trúc dữ liệu và cung cấp các phương thức thao tác với dữ liệu như tạo, cập nhật, truy vấn và xóa (CRUD).

Hệ thống còn được hỗ trợ bởi **utils**, nơi tập hợp các hàm tiện ích dùng chung như xử lý ảnh, gửi email, mã hóa dữ liệu, xử lý lỗi hoặc thời gian. Ngoài ra, để đảm bảo quản lý cơ sở dữ liệu có hệ thống, các thư mục **migrate** và **seeder** được sử dụng lần lượt cho việc cập nhật cấu trúc cơ sở dữ liệu và đổ dữ liệu mẫu hoặc dữ liệu mặc định vào hệ thống trong quá trình phát triển và kiểm thử.

Tổng thể, cấu trúc này giúp tách biệt rõ ràng giữa các tầng trong ứng dụng, đảm bảo khả năng mở rộng, tái sử dụng và bảo trì lâu dài cho hệ thống backend Node.js.

b, Biểu đồ phụ thuộc gói phần mềm chủ Flask



Hình 4.3: Biểu đồ gói máy chủ Flask

Hệ thống máy chủ Flask được thiết kế theo kiến trúc module hóa, bao gồm hai gói phần chính: Data, api và chromaDB. Mỗi gói đảm nhận một vai trò chuyên biệt trong quá trình xử lý dữ liệu và phản hồi truy vấn người dùng. Cấu trúc này giúp tăng tính tổ chức, dễ mở rộng và tối ưu hiệu suất khi xây dựng các ứng dụng dựa trên mô hình truy vấn và trí tuệ nhân tạo (LLM).

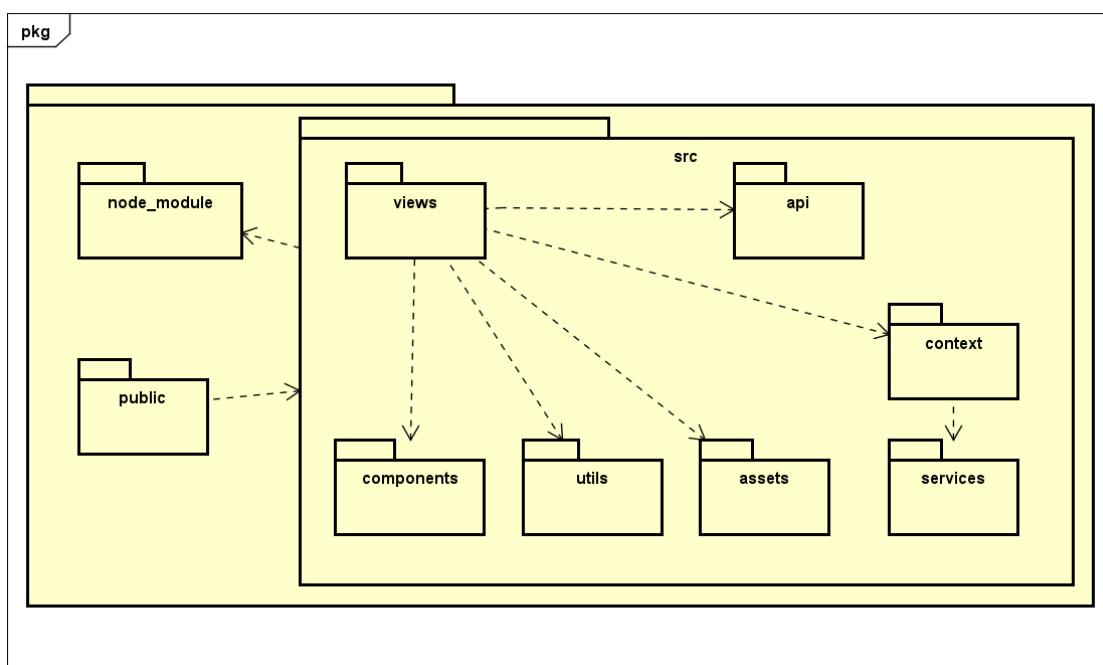
Gói **Data** là thành phần cốt lõi trong quá trình chuẩn bị dữ liệu để hệ thống có thể hiểu và phản hồi hiệu quả. Nó bao gồm nhiều bước xử lý nối tiếp nhau: bắt đầu bằng việc thu thập dữ liệu (crawl data) từ các nguồn web, sau đó lọc bỏ các phần không cần thiết và chia nhỏ dữ liệu thành các đoạn (chunk) phù hợp với dung lượng mà mô hình xử lý ngôn ngữ có thể tiếp nhận. Tiếp theo, từng đoạn dữ liệu

được vector hóa (embedding) để biểu diễn dưới dạng không gian số, và các vector này được lưu trữ vào **ChromaDB**, một cơ sở dữ liệu chuyên dụng cho tìm kiếm theo ngữ nghĩa. Nhờ đó, hệ thống có thể thực hiện các truy vấn hiệu quả dựa trên ý nghĩa của câu hỏi thay vì chỉ khớp từ khóa đơn thuần.

Gói **api** đảm nhận vai trò tiếp nhận truy vấn từ phía người dùng và phản hồi kết quả cuối cùng. Khi một yêu cầu được gửi lên, hệ thống sẽ sử dụng câu hỏi của người dùng để truy vấn vào ChromaDB, nhằm tìm ra các đoạn văn bản có liên quan nhất về mặt ngữ nghĩa. Sau đó, những kết quả truy vấn này sẽ được kết hợp với mô hình ngôn ngữ lớn (LLM) thông qua OpenRouter, nhằm tạo ra câu trả lời tự nhiên, chính xác và phù hợp với ngữ cảnh. Toàn bộ quá trình này được đóng gói trong các endpoint API, nơi xử lý yêu cầu và trả về câu trả lời dưới dạng JSON để frontend hoặc các hệ thống khác có thể dễ dàng tích hợp.

Tóm lại, mô hình phụ thuộc trong hệ thống Flask gồm hai phần chính: data đóng vai trò chuẩn bị và lưu trữ dữ liệu ngữ nghĩa, trong khi api đảm nhiệm việc truy vấn và tương tác với LLM để tạo ra câu trả lời cho người dùng. Việc tách biệt rõ ràng giữa hai gói phần này giúp nâng cao hiệu suất, dễ bảo trì và mở rộng khi cần nâng cấp chất lượng dữ liệu hoặc tích hợp các mô hình AI mới.

c, Biểu đồ phụ thuộc gói phần giao diện ReactJs



Hình 4.4: Biểu đồ lớp phần giao diện

Giao diện người dùng của hệ thống được xây dựng bằng ReactJS, được tổ chức theo mô hình module hóa với các gói phần rõ ràng, đảm bảo tính dễ bảo trì, mở

rộng và tái sử dụng. Cấu trúc này giúp phân tách trách nhiệm giữa các phần trong frontend và đồng bộ hiệu quả với backend cũng như các thành phần AI liên quan.

Gói **api** đóng vai trò là cầu nối giữa giao diện và backend, chứa toàn bộ các hàm gọi API sử dụng để lấy dữ liệu từ phía máy chủ. Việc gom nhóm các API vào một nơi giúp quản lý dễ dàng hơn, đồng thời tạo điều kiện để tái sử dụng và kiểm soát các thay đổi khi backend cập nhật.

Thư mục **assets** lưu trữ các tài nguyên tĩnh như hình ảnh, font chữ và biểu tượng. Đây là nơi tập trung toàn bộ tài nguyên giao diện, góp phần giữ cho mã nguồn React sạch sẽ và dễ quản lý, đồng thời hỗ trợ quá trình tùy biến giao diện nhanh chóng.

Gói **components** chứa các thành phần giao diện được dùng chung giữa nhiều trang như header, footer, button, modal,... Việc tách riêng các component giúp tái sử dụng hiệu quả, giảm thiểu lặp mã và hỗ trợ tổ chức UI theo hướng component-based rõ ràng.

Phần **context** chịu trách nhiệm quản lý trạng thái toàn cục của ứng dụng, đặc biệt là trạng thái xác thực người dùng. Kết hợp với useAuth – một custom hook được xây dựng để thực hiện xác thực người dùng và kiểm tra quyền truy cập – cả hai gói này giúp bảo vệ các route quan trọng và đảm bảo trải nghiệm người dùng cá nhân hóa.

Gói **pages** tổ chức các trang giao diện cụ thể như Trang chủ, Trang chi tiết, Trang quản trị,... Mỗi trang thường bao gồm phần bố cục riêng và sử dụng các component chung từ thư mục components.

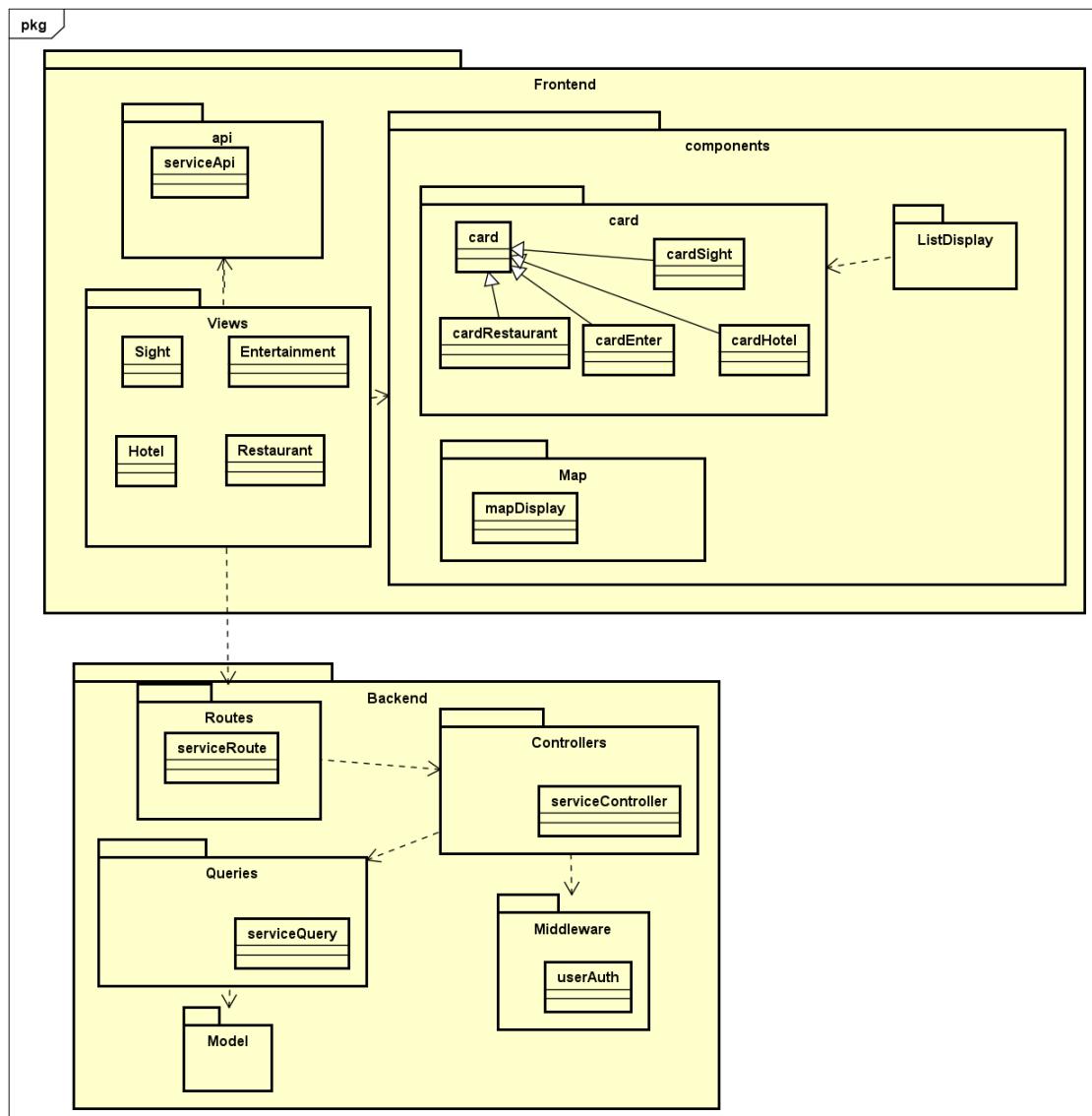
Gói **route** định nghĩa các đường dẫn điều hướng trong ứng dụng, sử dụng react-router-dom để ánh xạ giữa các URL và các trang trong pages. Cấu trúc này giúp duy trì điều hướng rõ ràng, dễ mở rộng khi thêm mới các chức năng.

Cuối cùng, thư mục **utils** tập hợp các hàm xử lý logic đơn lẻ như xử lý thời gian, định dạng văn bản, xử lý token,... Các hàm này hỗ trợ toàn bộ ứng dụng mà không bị phụ thuộc vào giao diện cụ thể.

Tổng thể, cấu trúc giao diện ReactJS này đảm bảo tính rõ ràng, khả năng mở rộng và dễ bảo trì, hỗ trợ việc xây dựng một frontend mạnh mẽ và chuyên nghiệp cho hệ thống web tích hợp AI.

4.1.3 Thiết kế chi tiết gói

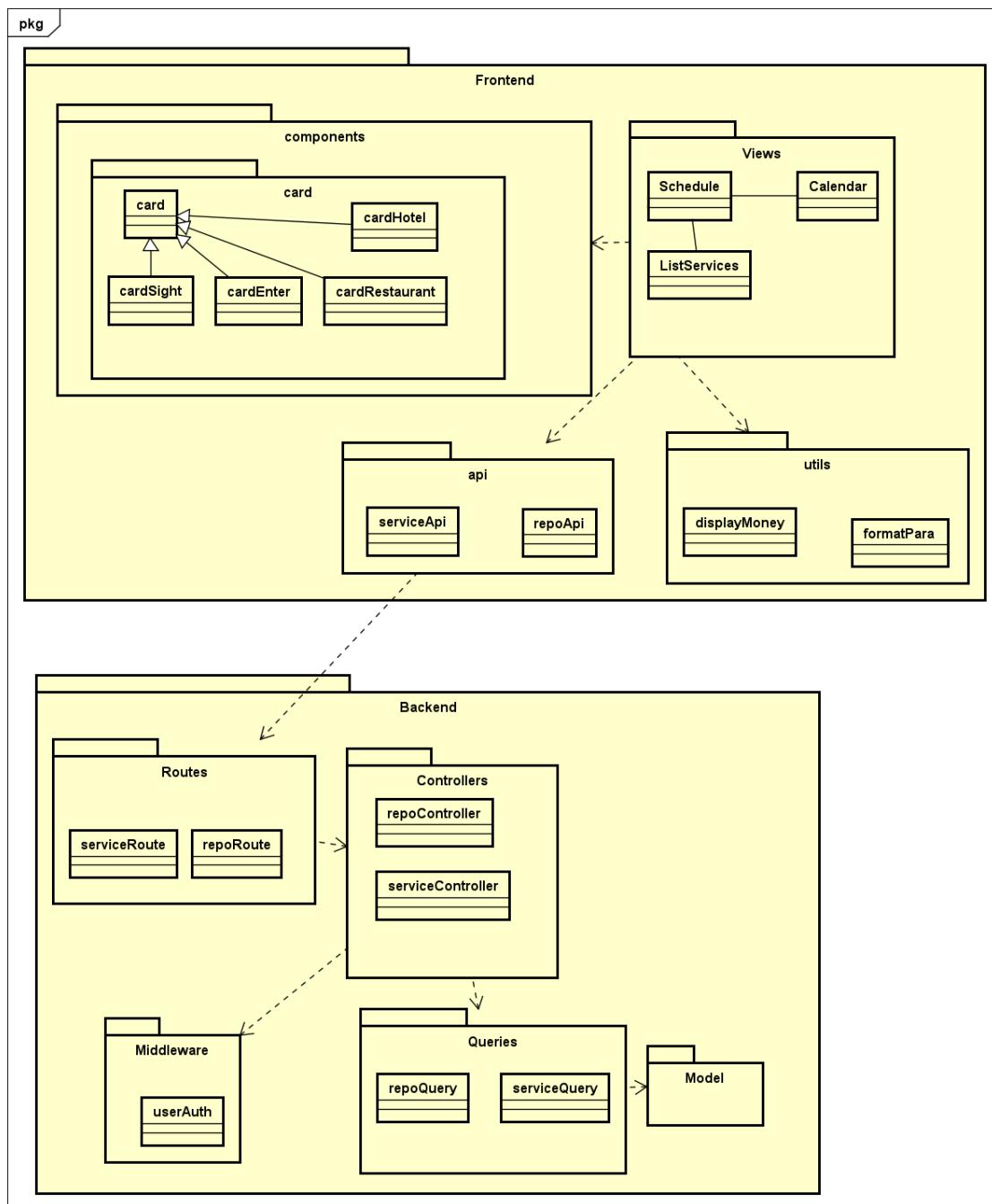
a, Sơ đồ chi tiết gói chức năng hiển thị danh sách dịch vụ



Hình 4.5: Biểu đồ lớp chức năng hiển thị danh sách dịch vụ

Biểu đồ 4.7 biểu thị thiết kế chi tiết của chức năng hiển thị danh sách dịch vụ. Giao diện nằm trong lớp Views sẽ gọi đến serviceApi từ lớp api và sử dụng các card từ component, lớp card sẽ được tạo ra coi như mẫu chung để hiển thị thông tin của dịch vụ như tên, địa điểm, đánh giá (sao), hình ảnh. Các dịch vụ (tham quan, vui chơi, khách sạn, nhà hàng) đều được thiết kế giao diện thừa kế từ lớp card, sau đó gọi đến gói ListDisplay để hiển thị các card dịch vụ theo kiểu danh sách và phân trang cùng một mẫu trong hệ thống. Cùng đó, lớp serviceApi bên frontend sẽ đảm nhận vai trò giao tiếp với máy chủ backend nodejs thông qua các routes được định nghĩa trong gói Routes, sử dụng các hàm trong lớp Controller.

b, Sơ đồ chi tiết gói chức năng thiết kế lô trình

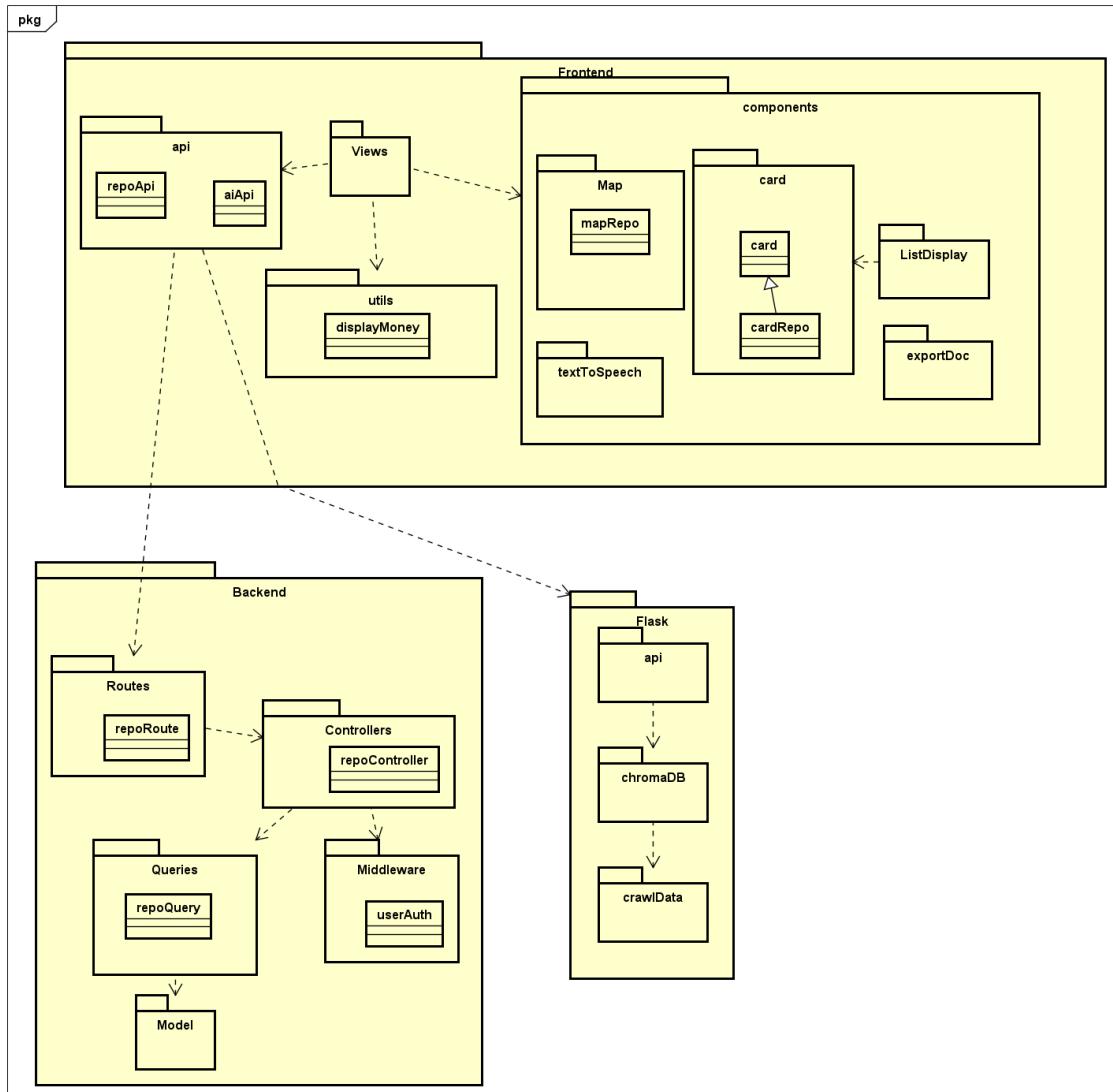


Hình 4.6: Biểu đồ lớp chức năng thiết kế lô trình

Biểu đồ 4.6 thiết kế gói chi tiết cho chức năng thiết kế lô trình du lịch. Chức năng này sử dụng các lớp giao diện Schedule, lớp này sử dụng lớp Calendar để hiển thị lịch giúp người dùng sắp xếp các dịch vụ vào lịch và lớp ListServices bao gồm các lớp card dịch vụ chứa thông tin về dịch vụ đó như tên, địa chỉ, đánh giá. Không những thế lớp giao diện còn sử dụng đến các lớp trong gói utils giúp hiển thị định dạng hiển thị tiền trong dịch vụ ẩm thực, khách sạn. Gói giao diện sẽ giao tiếp với máy chủ backend thông qua gói api và gói routes. Gói routes sẽ gọi đến các

hàm từ controller để xử lý dữ liệu từ người dùng và trả lại response cho bên phía giao diện.

c, Sơ đồ chi tiết gói chức năng quản lý lộ trình



Hình 4.7: Biểu đồ lớp chức năng quản lý lộ trình

Biểu đồ 4.7 thiết kế chi tiết gói cho chức năng quản lý lộ trình. Lớp giao diện trong gói Views sẽ gọi đến các components như mapRepo để hiển thị lịch trình chi tiết trên bản đồ, cardRepo để hiển thị thông tin một lộ trình, exportDoc phục vụ người dùng in mô tả, cẩm nang lộ trình, textToSpeech cho phép hệ thống đọc cẩm nang, phục vụ cho nhu cầu du lịch của người dùng. Ngoài ra lớp giao diện cũng sẽ sử dụng các api được viết trong lớp api và giao tiếp với máy chủ thông qua gói routes. Gói routes sẽ gọi đến controller để xử lý các yêu cầu đến từ phía giao diện.

4.2 Thiết kế chi tiết

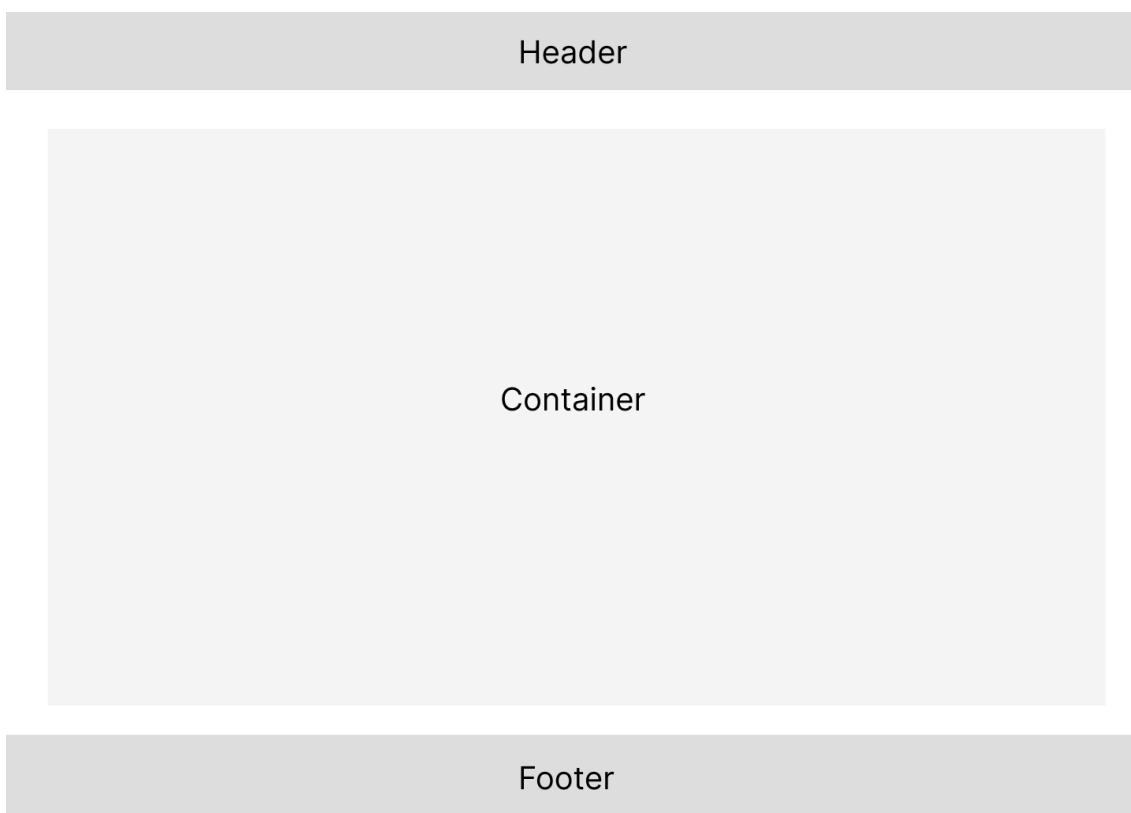
4.2.1 Thiết kế giao diện

a, Thông số giao diện

Thông số	Mô tả
Màn hình	Cung cấp giao diện thích ứng cho cả máy tính và điện thoại
Màu sắc	Màu chủ đạo: #336c72 Màu phụ: #006d75 Màu nền: #ddf0ff Màu chữ: #2c2c2c
Kiểu, cỡ chữ	16px Roboto
Ngôn ngữ	Tiếng Việt
Thông báo	Thông báo thành công màu xanh lá cây, thông báo lỗi màu đỏ, thông báo thông tin màu xanh dương
Nút	Nền: màu phụ #006d75, padding: 14px, màu chữ trắng

Bảng 4.1: Thông số giao diện

b, Thiết kế minh họa bố cục trang cơ bản



Hình 4.8: Mockup bố cục trang cơ bản

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Hình 4.8 minh họa bố cục trang chung gồm 3 phần cơ bản dưới đây:

- **Header:** Chứa tiêu đề trang và các thông tin chung như thông tin người dùng .
- **Container:** Chứa nội dung chính của trang đó .
- **Footer:** Chứa thông tin bổ sung và liên kết đến các phần của ứng dụng.

c, Thiết kế minh họa hiển thị danh sách các dịch vụ



Hình 4.9: Mockup hiển thị danh sách dịch vụ

Hình 4.9 minh họa giao diện hiển thị danh sách các dịch vụ. Bốn dịch vụ tham quan, vui chơi, nhà hàng, khách sạn đều sử dụng mẫu giao diện này. Minh họa gồm ô tìm kiếm theo tên dịch vụ, các bộ lọc và các dịch vụ được hiển thị bằng các card như hình minh họa.

d, Thiết kế minh họa thiết kế lộ trình du lịch

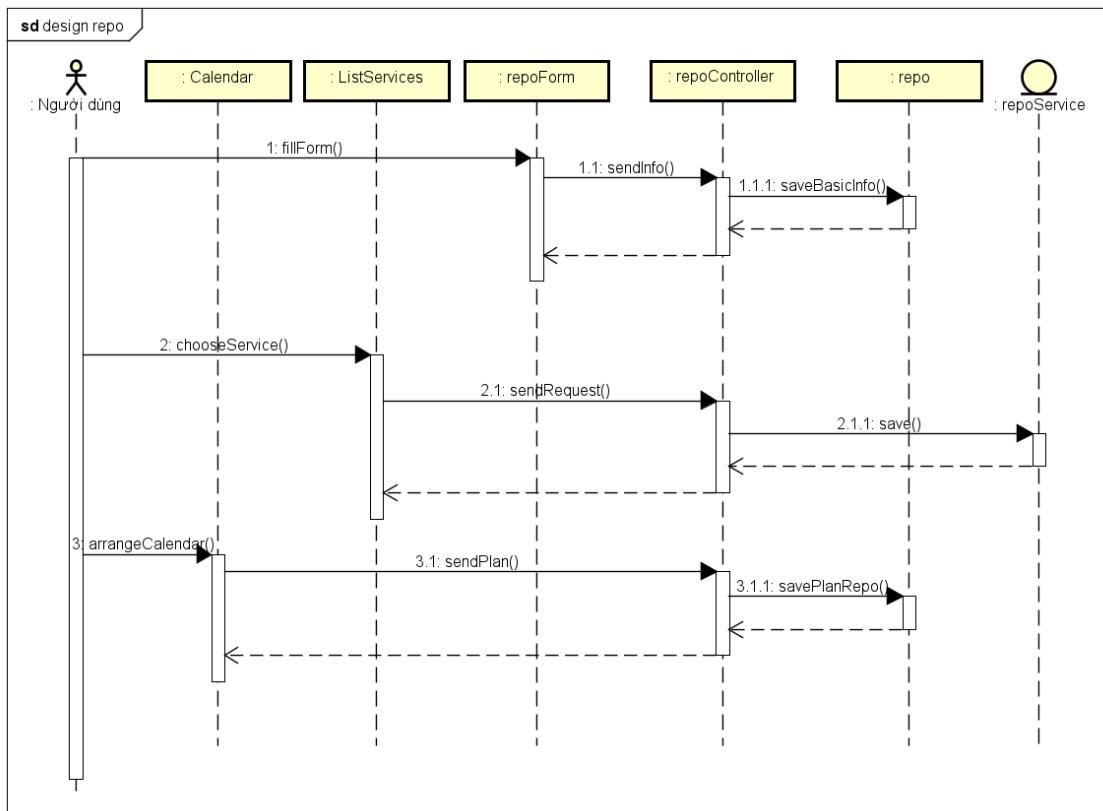


Hình 4.10: Mockup thiết kế lộ trình du lịch

Hình 4.10 minh họa giao diện thiết kế lộ trình du lịch với phần bên trái hiển thị các dịch vụ mà người dùng đã chọn cho lộ trình, có thể là một trong 4 dịch vụ tham quan, vui chơi, khách sạn, nhà hàng. Bên phải là layout lịch để người dùng có thể sắp xếp các dịch vụ vào đó theo các khung giờ mà người dùng muốn.

4.2.2 Thiết kế lớp

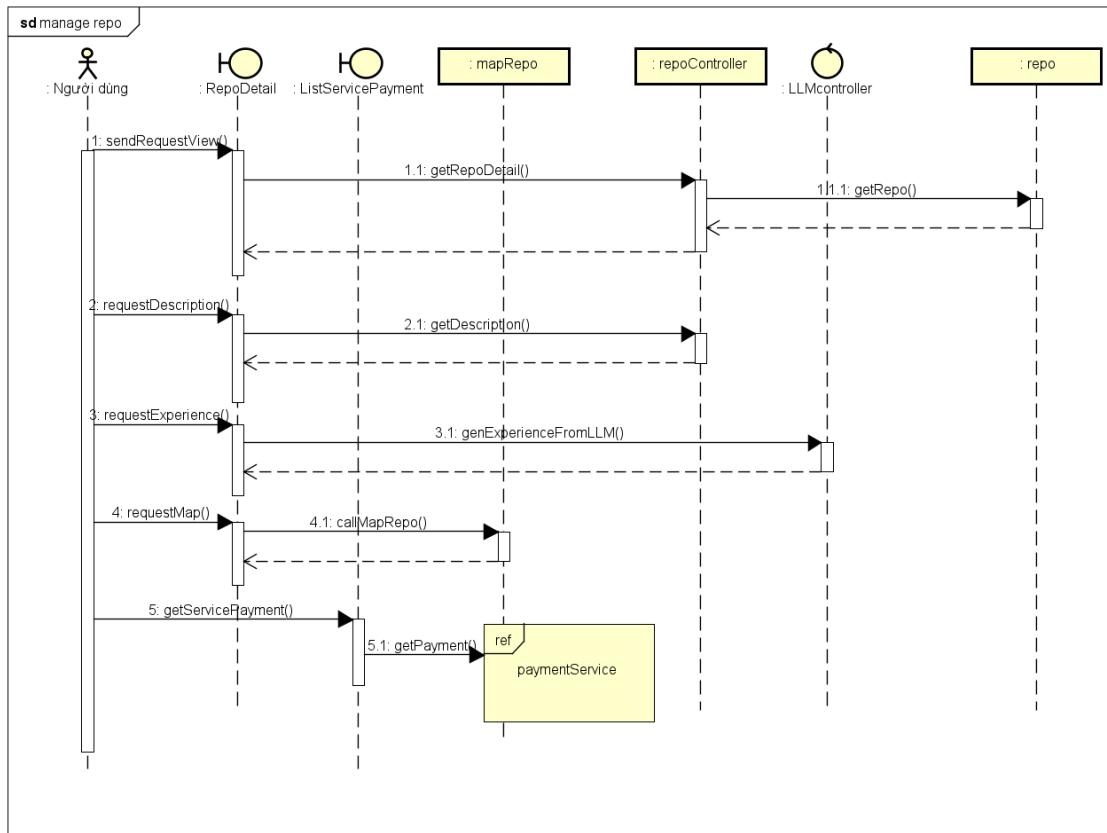
a, Sơ đồ luồng cho chức năng thiết kế lộ trình du lịch



Hình 4.11: Biểu đồ luồng cho chức năng thiết kế lộ trình du lịch

Để thiết kế lộ trình du lịch, trước tiên người dùng cần điền form thông tin cơ bản của lộ trình tại lớp repoForm, từ đó repoForm sẽ gọi đến repoController và lưu thông tin cơ bản vào model repo. Sau khi đã tạo được thông tin cơ bản của lộ trình, người dùng sẽ tương tác với các dịch vụ du lịch, chọn/bỏ dịch vụ và các dịch vụ được chọn sẽ được lưu trong model repoService. Tiếp đến người dùng tương tác với lớp Calendar để sắp xếp các dịch vụ vào lịch cũng như hoàn thành việc thiết kế lộ trình.

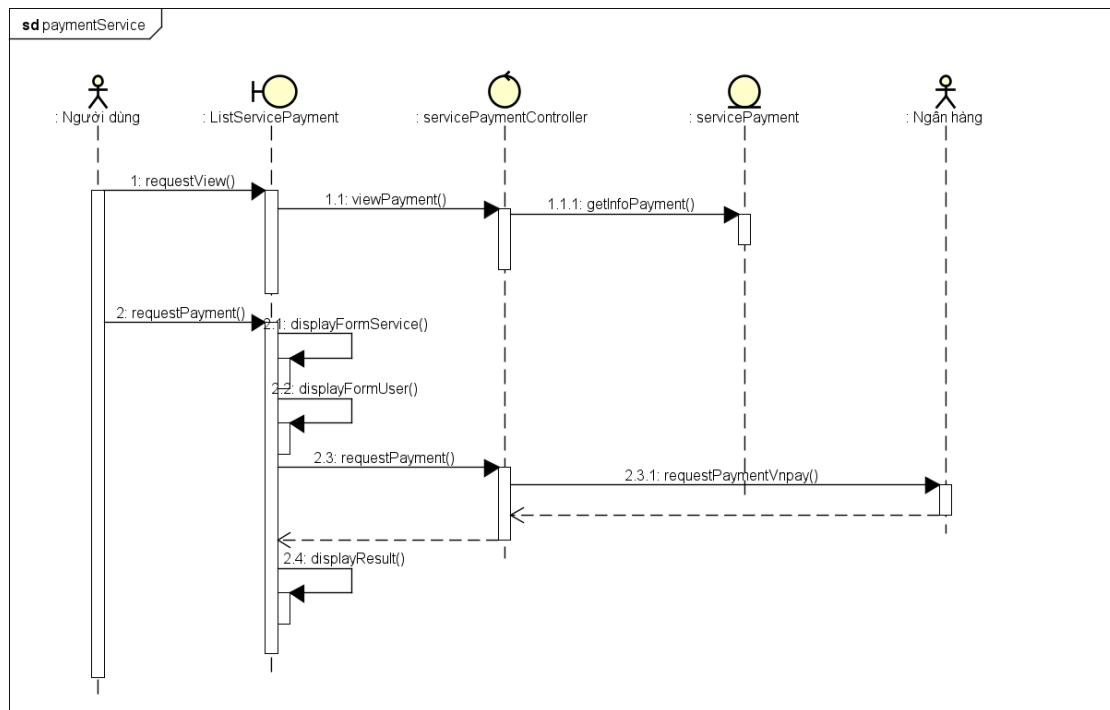
b, Sơ đồ luồng cho chức năng thiết kế quản lý lộ trình du lịch



Hình 4.12: Biểu đồ luồng cho chức năng quản lý lộ trình du lịch

Với chức năng quản lý lộ trình du lịch, người dùng sẽ tương tác chủ yếu với lớp RepoDetail. Người dùng đưa ra các yêu cầu đến lớp này, lớp này sẽ gọi đến repoController để xử lý các yêu cầu, sau đó sẽ sử dụng model repo để lấy được các thông tin mà người dùng yêu cầu như là xem chi tiết lộ trình, xem mô tả lộ trình. Ngoài ra khi người dùng yêu cầu xem lộ trình trên bản đồ thì người dùng sẽ tương tác với lớp mapRepo. Với cẩm nang du lịch, lớp giao diện repoDetail sẽ gọi đến xử lý LLM với LLMController.

c, Sơ đồ luồng cho chức năng thanh toán các dịch vụ trong lộ trình du lịch



Hình 4.13: Biểu đồ luồng cho chức năng thanh toán các dịch vụ trong lộ trình du lịch

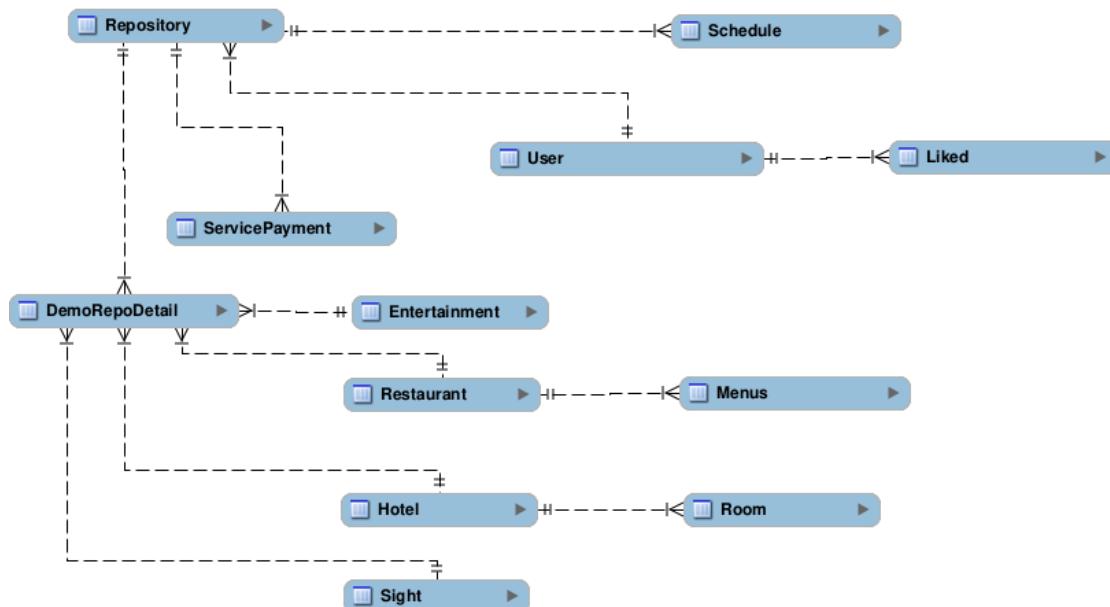
Người dùng yêu cầu xem danh sách các dịch vụ của lộ trình thông qua lớp giao diện ListServicePayment, lớp này sẽ cung cấp danh sách các dịch vụ vé vui chơi, đặt phòng khách sạn, đặt trước bàn ăn của lộ trình (nếu có).

Khi người dùng yêu cầu đặt dịch vụ, lớp giao diện sẽ hiển thị các form để người dùng nhập các thông tin dịch vụ, thông tin liên hệ, sau đó lớp servicePaymentController sẽ gọi đến hệ thống thanh toán và trả lại thông báo kết quả thanh toán cho người dùng.

4.2.3 Thiết kế cơ sở dữ liệu

Trong đồ án này, hệ thống sử dụng cơ sở dữ liệu MySQL để lưu trữ và quản lý dữ liệu. MySQL là một hệ quản trị cơ sở dữ liệu quan hệ phổ biến, cho phép tổ chức dữ liệu theo mô hình bảng với cấu trúc rõ ràng và nhất quán. Mỗi bảng bao gồm các cột với kiểu dữ liệu xác định, cùng với các ràng buộc như khóa chính, khóa ngoại nhằm đảm bảo tính toàn vẹn và đồng nhất của dữ liệu. Việc áp dụng mô hình quan hệ giúp hệ thống dễ dàng thực hiện các phép truy vấn, cập nhật và phân tích dữ liệu thông qua ngôn ngữ SQL tiêu chuẩn. Ngoài ra, MySQL còn hỗ trợ tối ưu hóa hiệu suất truy vấn thông qua các chỉ mục và cơ chế kiểm soát giao dịch, từ đó nâng cao độ tin cậy và hiệu quả trong quá trình xử lý dữ liệu. Với đặc

điểm đó, MySQL đặc biệt phù hợp cho các hệ thống yêu cầu quản lý thông tin có cấu trúc và quan hệ logic chặt chẽ.



Hình 4.14: Sơ đồ cơ sở dữ liệu tổng quan

Sơ đồ 4.14 mô tả mối quan hệ giữa các thực thể trong cơ sở dữ liệu của hệ thống. Một thực thể User (người dùng) có thể tạo ra nhiều Repository (lộ trình du lịch). Mỗi Repository bao gồm nhiều DemoRepoDetail, là danh sách các dịch vụ cụ thể được người dùng lựa chọn cho lộ trình đó, và nhiều ServicePayment, đại diện cho các dịch vụ có tính phí liên quan đến lộ trình.

Thực thể DemoRepoDetail có quan hệ trực tiếp với bốn bảng dịch vụ chính: Sight (địa điểm tham quan), Entertainment (khu vui chơi), Hotel (khách sạn) và Restaurant (nhà hàng), phản ánh các loại hình dịch vụ du lịch có thể được thêm vào lộ trình.

Đối với bảng Hotel, hệ thống còn lưu trữ thông tin chi tiết về các loại phòng thông qua bảng Room. Tương tự, bảng Restaurant liên kết với bảng Menus, cho phép lưu trữ các món ăn hoặc thực đơn cụ thể được cung cấp tại nhà hàng đó.

Sau đây em xin trình bày chi tiết thông tin từng bảng và các trường thông tin của bảng đó:

a, Thiết kế chi tiết bảng User

Bảng user mô tả thông tin người dùng cho hệ thống

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	integer	PK	id duy nhất của người dùng
name	varchar(255)	not null	tên người dùng
phone	varchar(10)	not null	số điện thoại của người dùng
address	varchar(255)	not null	địa chỉ của người dùng
email	varchar(255)	isEmail, unique	email của người dùng
password	varchar(255)	ít nhất 6 ký tự	mật khẩu tài khoản đăng nhập của người dùng
role	varchar(255)		vai trò của người dùng trong hệ thống
ava	varchar(255)		tên ảnh đại diện của người dùng

Bảng 4.2: Thiết kế chi tiết bảng User

b, Thiết kế chi tiết bảng Sight

Bảng Sight mô tả địa điểm tham quan

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	varchar(255)	PK, uuid	id duy nhất của địa điểm tham quan
name	varchar(255)	not null	tên địa điểm
description	text	not null	mô tả địa điểm
address	varchar(255)	not null	địa chỉ
images	text		danh sách các tên ảnh của địa điểm
startTime	integer		thời gian mở cửa của địa điểm
endTime	integer		thời gian đóng cửa của địa điểm
rate	integer		đánh giá (số sao) địa điểm
lat	float		kinh độ của địa điểm (được chuyển tự động từ địa chỉ)
lng	float		vĩ độ của địa điểm (được chuyển tự động từ địa chỉ)

Bảng 4.3: Thiết kế chi tiết bảng Sight

c, Thiết kế chi tiết bảng Entertainment

Bảng Entertainment mô tả địa điểm vui chơi

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	varchar(255)	PK, uuid	id duy nhất của địa điểm
name	varchar(255)	not null	tên địa điểm
description	text	not null	mô tả địa điểm
address	varchar(255)	not null	địa chỉ
price	float	not null	giá vé vui chơi
images	text		danh sách các tên ảnh của địa điểm
startTime	integer		thời gian mở cửa của địa điểm
endTime	integer		thời gian đóng cửa của địa điểm
rate	integer		đánh giá (số sao) địa điểm
lat	float		kinh độ của địa điểm (được chuyển tự động từ địa chỉ)
lng	float		vĩ độ của địa điểm (được chuyển tự động từ địa chỉ)

Bảng 4.4: Thiết kế chi tiết bảng Entertainment

d, Thiết kế chi tiết bảng Hotel

Bảng Hotel mô tả thông tin khách sạn

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	varchar(255)	PK, uuid	id duy nhất của địa điểm
name	varchar(255)	not null	tên địa điểm
description	text	not null	mô tả địa điểm
address	varchar(255)	not null	địa chỉ
images	text		danh sách các tên ảnh của địa điểm
checkin	integer		thời gian nhận phòng
checkout	integer		thời gian trả phòng
near_location	varchar(255)		địa điểm tham quan gần đó
rate	integer		đánh giá (số sao) địa điểm
lat	float		kinh độ của địa điểm (được chuyển tự động từ địa chỉ)
lng	float		vĩ độ của địa điểm (được chuyển tự động từ địa chỉ)

Bảng 4.5: Thiết kế chi tiết bảng Hotel

e, Thiết kế chi tiết bảng Room

Bảng room mô tả thông tin phòng của khách sạn

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	integer	PK	id duy nhất của phòng
name	varchar(255)	not null	tên phòng
description	text	not null	mô tả phòng
hotel_id	varchar(255)	FK	khóa ngoài trỏ tới Hotel (id)
image	text		tên ảnh của phòng đó
price	float		giá phòng đó một đêm

Bảng 4.6: Thiết kế chi tiết bảng Room

f, Thiết kế chi tiết bảng Restaurant

Bảng Restaurant mô tả thông tin của nhà hàng

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	varchar(255)	PK, uuid	id duy nhất của địa điểm
name	varchar(255)	not null	tên địa điểm
description	text	not null	mô tả địa điểm
address	varchar(255)	not null	địa chỉ
images	text		danh sách các tên ảnh của địa điểm
price	text		khoảng giá của món ăn
video	varchar(255)		link youtube giới thiệu nhà hàng
phone	varchar(10)		số điện thoại liên hệ của nhà hàng
open	integer		thời gian mở cửa quán
close	integer		thời gian đóng cửa quán
rate	integer		đánh giá (số sao) địa điểm
lat	float		kinh độ của địa điểm (được chuyển tự động từ địa chỉ)
lng	float		vĩ độ của địa điểm (được chuyển tự động từ địa chỉ)

Bảng 4.7: Thiết kế chi tiết bảng Restaurant

g, Thiết kế chi tiết bảng Menus

Bảng Menus mô tả thông tin món ăn của nhà hàng

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	integer	PK	id duy nhất của món ăn
name	varchar(255)	not null	tên món ăn
description	text	not null	mô tả món ăn
restaurant_id	varchar(255)	FK	khóa ngoài trỏ tới Restaurant (id)
image	text		tên ảnh của món đó
price	float		giá món đó với số lượng một

Bảng 4.8: Thiết kế chi tiết bảng Menus

h, Thiết kế chi tiết bảng Repository

Bảng Repository mô tả thông tin của một lô trình

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	varchar(255)	PK, uuid	id duy nhất của lộ trình
name	varchar(255)	not null	tên lộ trình
description	text	not null	mô tả địa điểm
address	varchar(255)	not null	địa chỉ
user_id	integer	FK, not null	khóa ngoài trỏ tới User (id)
numberPeople	integer		số người tham gia lộ trình
startDate	datetime	not null	thời gian bắt đầu lộ trình
endDate	datetime	not null	thời gian kết thúc lộ trình
destination	object	not null	danh sách các điểm đến của lộ trình
plan	object		mảng các dịch vụ - thời gian trong lộ trình
experience	text		cẩm nang của lộ trình
isHidden	boolean		lộ trình có bị ẩn đi hay không

Bảng 4.9: Thiết kế chi tiết bảng Repository

i, Thiết kế chi tiết bảng DemoRepoDetail

Bảng DemoRepoDetail mô tả thông tin các dịch vụ được chọn trong lộ trình

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	varchar(255)	PK, uuid	id duy nhất của bản ghi
service_id	varchar(255)	not null	id của dịch vụ
service_type	enum	not null	loại dịch vụ (tham quan, vui chơi, nhà hàng, khách sạn)
repository_id	varchar(255)	FK	khóa ngoài tới Repository(id)

Bảng 4.10: Thiết kế chi tiết bảng DemoRepoDetail

j, Thiết kế chi tiết bảng ServicePayment

Bảng ServicePayment mô tả thông tin kết quả thanh toán dịch vụ của lô trình

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	varchar(255)	PK, uuid	id duy nhất của bản ghi
repository	varchar(255)	FK	khóa ngoài trỏ tới Repository(id)
service_id	varchar(255)	not null	id của dịch vụ
service_type	enum	not null	loại dịch vụ (tham quan, vui chơi, nhà hàng, khách sạn)
amount	float		số tiền thanh toán
email	varchar(255)	isEmail	email liên hệ của người đặt dịch vụ
name	varchar(255)	not null	tên người đặt dịch vụ
phone	varchar(10)	not null	số điện thoại người đặt dịch vụ
note	varchar(255)	not null	ghi chú khi đặt dịch vụ
result	enum		kết quả thanh toán
countAdult	integer		số lượng người lớn
countChild	integer		số lượng trẻ con
orderDate	datetime		ngày đặt dịch vụ
subId	integer		trỏ tới id của dịch vụ con (phòng khách sạn)

Bảng 4.11: Thiết kế chi tiết bảng ServicePayment

k, Thiết kế chi tiết bảng Liked

Bảng Liked mô tả thông tin các dịch vụ được yêu thích bởi người dùng

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Tên	Kiểu dữ liệu	Ràng buộc	Mô tả
id	varchar(255)	PK, uuid	id duy nhất của bản ghi
service_id	varchar(255)	not null	id của dịch vụ
service_type	enum	not null	loại dịch vụ (tham quan, vui chơi, nhà hàng, khách sạn)
user_id	integer	FK	khóa ngoài tới User(id)

Bảng 4.12: Thiết kế chi tiết bảng Liked

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Thư viện và công cụ được sử dụng trong hệ thống

Mục đích	Công cụ	Đường link
IDE lập trình FE	Visual Studio code	https://code.visualstudio.com/
IDE lập trình BE	Visual Studio code	https://code.visualstudio.com/
Framework FE	ReactJs	https://react.dev/
Framework BE	ExpressJs	https://expressjs.com/
Framework BE	Flask	https://flask.palletsprojects.com/en/stable/
Hệ quản trị cơ sở dữ liệu	MySQL	https://www.mysql.com/
Thao tác với cơ sở dữ liệu	MySQL workbench	https://www.mysql.com/products/workbench/
Thư viện UI	Ant Design	https://ant.design/
Thiết kế giao diện	Figma	https://www.figma.com/
Quản lý code	Github	https://github.com/
Kiểm thử API	Postman	https://www.postman.com/
Tích hợp LLM	OpenRoute	https://openrouter.ai/
Bản đồ	Mapbox	https://www.mapbox.com/

Bảng 4.13: Thư viện và công cụ sử dụng

4.3.2 Kết quả đạt được

Về cấu trúc mã nguồn, hệ thống được tổ chức thành hai thành phần chính: backend và frontend, mỗi thành phần đảm nhận các vai trò riêng biệt. Backend được chia thành hai module: module Node.js phụ trách xử lý các yêu cầu người dùng và logic nghiệp vụ cốt lõi của hệ thống, trong khi module Flask tập trung vào chức năng sinh cảm nang du lịch với RAG tích hợp LLM. Tương tự, frontend quản lý giao diện người dùng. Frontend đều chịu trách nhiệm xây dựng các giao diện trực quan, hỗ trợ người dùng tương tác và gửi yêu cầu đến máy chủ. Cấu trúc phân tách này giúp tăng tính mô-đun, dễ dàng bảo trì và phát triển hệ thống. Dưới đây là các chức năng chính đã triển khai được:

- Tích hợp các địa điểm tham quan, vui chơi, khách sạn, nhà hàng trong một hệ thống:** Với **1000** địa điểm tham quan, **106** địa điểm vui chơi, **540** nhà hàng, **330** khách sạn trên khắp 63 tỉnh thành của đất nước, hệ thống cung cấp đầy đủ các dịch vụ có thể lựa chọn cho lộ trình du lịch của mình.

- **Thiết kế lộ trình du lịch:** Người dùng có thể tự sắp xếp các dịch vụ du lịch yêu thích vào các khung giờ khác nhau trong thời gian của lộ trình với những đề xuất dịch vụ từ hệ thống. Hệ thống không chỉ đề xuất các dịch vụ dựa vào điểm đến mà còn đề xuất các dịch vụ gần nhau, giúp người dùng có thể tối ưu được lộ trình.
- **Quản lý lộ trình linh hoạt:** Với lộ trình đã được tạo, hệ thống cho phép người dùng có thể xem danh sách các dịch vụ trong lộ trình dưới dạng mô tả tour và cả trên bản đồ giúp người dùng dễ dàng hình dung được lộ trình diễn ra như thế nào. Ngoài ra dựa vào những tri thức cẩm nang du lịch mà hệ thống có, hệ thống sẽ đề xuất các kinh nghiệm cho người dùng dựa vào mô tả của lộ trình.
- **Quản lý thanh toán trong lộ trình:** Trong một lộ trình, người dùng có thể xem các dịch vụ mà có thể đặt trước giúp tiết kiệm thời gian cũng như dễ kiểm soát được lộ trình. Với các dịch vụ có thể đặt như vé vui chơi, đặt phòng khách sạn, đặt bàn ăn, người dùng có thể đặt và thanh toán với vnpay, hệ thống sẽ thông báo kết quả và hiển thị dịch vụ thanh toán cho người dùng.

Thông kê về thông tin ứng dụng được mô tả trong bảng sau:

Thông tin	Thông kê
Số lớp mã nguồn	32 lớp
Số bảng trong cơ sở dữ liệu	12 bảng
Số dòng code Frontend	36713 dòng
Số dòng code Backend nodejs	108869 dòng
Số dòng code Backend flask	177724 dòng

Bảng 4.14: Thông tin ứng dụng

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

4.3.3 Minh họa các chức năng chính

a, Giao diện minh họa chức năng hiển thị danh sách dịch vụ

The screenshot shows a web-based travel service interface. At the top, there are five tabs: Lộ trình (Route), Địa điểm tham quan (Travel Destinations), Địa điểm vui chơi (Entertainment Venues), Địa điểm ẩm thực (Food & Beverage), and Địa điểm nghỉ dưỡng (Accommodation). Below the tabs, there is a search bar labeled 'Nhập tên địa điểm' (Enter location name) and several filters: 'Tỉnh Hải Dương' (Hai Duong Province), 'Giá cả' (Price), and 'Yêu thích' (Favorites). The main content area displays four travel destination cards:

- Ling noi nua Thanh Hai**: Located in Hai Duong. Rating: ★★★★★.
- Den Cao An Phu**: Located in Hai Duong. Rating: ★★★★★.
- Rung phong la do - Chua Thanh Mai**: Located in Hai Duong. Rating: ★★★★★.
- Chua - Dong Kinh Chu**: Located in Hai Duong. Rating: ★★★★★.

Below the cards, there are small thumbnail images for other travel options.

Hình 4.15: Giao diện hiển thị danh sách địa điểm tham quan

The screenshot shows a web-based travel service interface. At the top, there are five tabs: Lộ trình (Route), Địa điểm tham quan (Travel Destinations), Địa điểm vui chơi (Entertainment Venues), Địa điểm ẩm thực (Food & Beverage), and Địa điểm nghỉ dưỡng (Accommodation). Below the tabs, there is a search bar labeled 'Nhập tên địa điểm' (Enter location name) and several filters: 'Tỉnh Kiên Giang' (Kien Giang Province), 'Tất cả' (All), and 'Yêu thích' (Favorites). The main content area displays four entertainment venue cards:

- Vé Show Kiss of The Sea – SunWorld**: Located in Bãi Đất Đỏ, Phú Quốc. Price: 787.939đ. Rating: ★★★★★.
- Vé SunWorld Hon Thom Nature Park**: Located in Bãi Đất Đỏ, Phú Quốc. Price: 10.000.000.000đ. Rating: ★★★★★.
- Vé đi thuyền trên sông Venice**: Located in Khu Bãi Dài, Phú Quốc. Price: 200.000đ. Rating: ★★★★★.
- Vé Show Kiss of The Sea – SunWorld**: Located in Bãi Đất Đỏ, Phú Quốc. Price: 257.955đ. Rating: ★★★★★.

Below the cards, there are small thumbnail images for other entertainment options.

Hình 4.16: Giao diện hiển thị danh sách địa điểm vui chơi

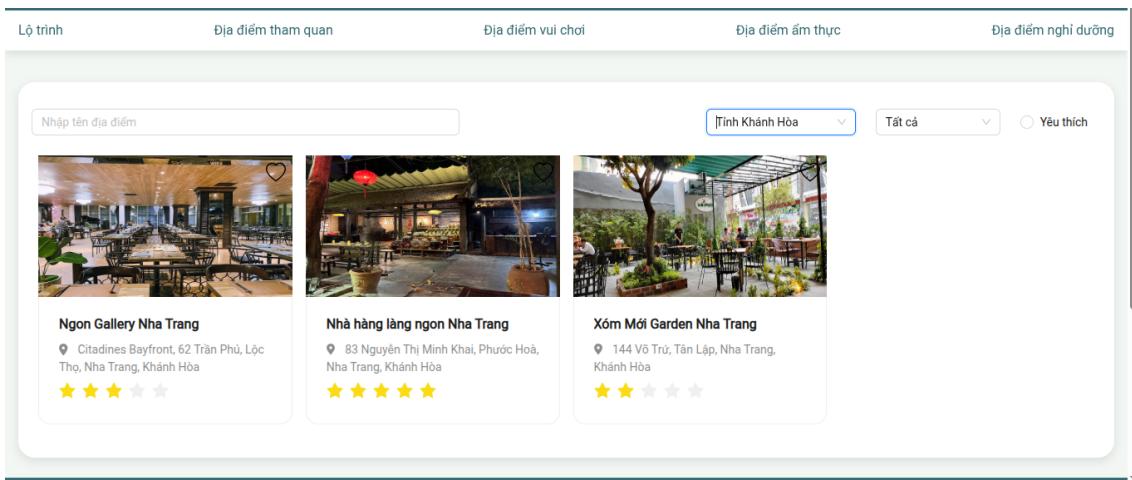
The screenshot shows a web-based travel service interface. At the top, there are five tabs: Lộ trình (Route), Địa điểm tham quan (Travel Destinations), Địa điểm vui chơi (Entertainment Venues), Địa điểm ẩm thực (Food & Beverage), and Địa điểm nghỉ dưỡng (Accommodation). Below the tabs, there is a search bar labeled 'Nhập tên địa điểm' (Enter location name) and several filters: 'Tỉnh Sơn La' (Sơn La Province), 'Tất cả' (All), and 'Yêu thích' (Favorites). The main content area displays four accommodation cards:

- Glenda Tower Mộc Châu Hotel**: Located in Mộc Châu. Price: 565 đường Trần Huy Liệu, thị trấn Mộc Châu, huyện Mộc Châu, tỉnh Sơn La. Rating: ★★★★★.
- Khách sạn Mường Thanh Holiday**: Located in Đường Hoàng Quốc Việt, Thị trấn Nông trường Mộc Châu, Huyện Mộc...
- Wooden House Moc Chau Homes...**: Located in Bản Chiềng Dì, thị trấn Mộc Châu, huyện Mộc Châu, tỉnh Sơn La. Rating: ★★★★★.
- Phoenix Moc Chau Resort**: Located in Rừng Thông Bản Áng, xã Đông Sang, huyện Mộc Châu, tỉnh Sơn La. Rating: ★★★★★.

Below the cards, there are small thumbnail images for other accommodation options.

Hình 4.17: Giao diện hiển thị danh sách khách sạn

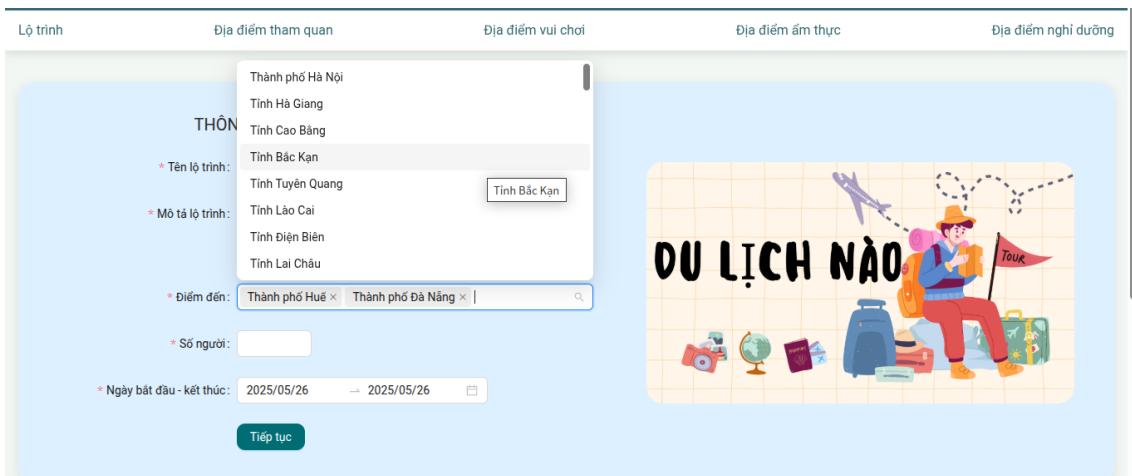
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.18: Giao diện hiển thị danh sách nhà hàng

Các dịch vụ đều được hiển thị theo dạng danh sách, mỗi trang sẽ có tám dịch vụ, phân trang cho người dùng dễ theo dõi các dịch vụ. Với mỗi dịch vụ đều được hiển thị các thông tin cơ bản như tên địa điểm, địa chỉ và đánh giá. Đồng thời hệ thống cũng cung cấp ô tìm kiếm theo tên và bộ lọc (lọc theo tỉnh/thành phố, sắp xếp theo số sao đánh giá tăng dần/giảm dần) giúp người dùng nhanh chóng tìm được địa điểm theo ý muốn.

b, Giao diện minh họa chức năng thiết kế lộ trình du lịch



Hình 4.19: Giao diện form nhập thông tin cơ bản của lộ trình

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

The screenshot shows a navigation bar with five tabs: Lộ trình (Itinerary), Địa điểm tham quan (Tourist Attraction), Địa điểm vui chơi (Entertainment Spot), Địa điểm ẩm thực (Food Spot), and Địa điểm nghỉ dưỡng (Accommodation). Below the tabs is a search bar labeled 'Nhập tên địa điểm bạn muốn tìm' (Enter the name of the destination you want to find) and a 'HƯỚNG DẪN' (Guide) button. A section titled 'DANH SÁCH ĐỊA ĐIỂM Ở HUẾ' (List of destinations in Huế) displays four cards with images, names, addresses, star ratings, and 'THÊM VÀO LỘ TRÌNH' (Add to itinerary) buttons. The cards are: Làng hoa Phú Mậu (Flower Village), Suối Voi (Waterfall), Phá Tam Giang (Breakthrough River), and Thác Anor (Anor Waterfall).

Hình 4.20: Giao diện hiển thị dịch vụ hệ thống để xuất cho lộ trình

This screenshot shows the same application interface as Figure 4.20, but for Da Nang. It features a 'DANH SÁCH ĐỊA ĐIỂM Ở ĐÀ NẴNG' (List of destinations in Da Nang) section with four cards: Cầu Rồng (Dragon Bridge), Bà Nà Hill, Suối khoáng nóng Thần Tài (Thermal Spring), and Rạn Nam Ô (Nam Ô Reef). Each card includes an image, name, address, star rating, and a 'THÊM VÀO LỘ TRÌNH' button.

Hình 4.21: Giao diện hiển thị dịch vụ mà hệ thống để xuất cho lộ trình

This screenshot shows a detailed itinerary view for June 5, 2025. The left side lists various services: Tham quan Làng hoa Phú Mậu, Tham quan Phá Tam Giang, Tham quan Suối khoáng nóng Thần Tài, Tham quan Cầu Rồng, Tham quan Rạn Nam Ô, Vui chơi tại Vé Suối khoáng nóng Alba Thanh Tân - Huế, and Nghỉ dưỡng tại Imperial Hotel Huế. The right side shows a calendar view for June 5, 2025, with specific events listed: 'Tham quan Rạn Nam Ô' at 08 giờ and 'Nghỉ dưỡng tại Imperial Hotel Huế' at 12 giờ. The interface includes tabs for different service types and a date selector.

Hình 4.22: Giao diện hiển thị dịch vụ và lịch đến người dùng có thể sắp xếp lộ trình

Người dùng nhập thông tin cơ bản của lộ trình vào form với giao diện hình 4.19, sau khi nhập các thông tin cơ bản đó, dựa vào điểm đến mà người dùng, hệ thống

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

để xuất các dịch vụ như hình 4.20, 4.21. Người dùng thực hiện ấn chọn các dịch vụ mà mình muốn cho vào lộ trình. Với các dịch vụ đã được chọn đó, người dùng có thể sắp xếp các dịch vụ đó vào các khung giờ trong lịch được minh họa ở hình 4.22. Trong khi sắp xếp, người dùng có thể lưu tạm thời lộ trình và thêm các dịch vụ khác, sau khi sắp xếp xong lộ trình, người dùng ấn hoàn thành lộ trình.

c, Giao diện minh họa chức năng quản lý lộ trình du lịch

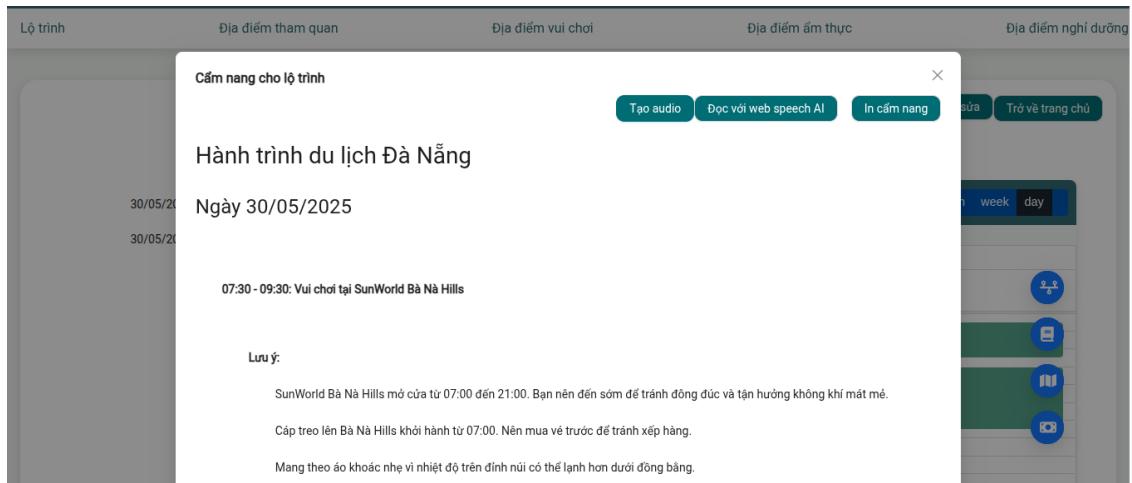
The screenshot shows a web-based application for managing travel itineraries. At the top, there are five tabs: 'Lộ trình' (Itinerary), 'Địa điểm tham quan' (Attraction), 'Địa điểm vui chơi' (Entertainment), 'Địa điểm ẩm thực' (Food), and 'Địa điểm nghỉ dưỡng' (Accommodation). Below the tabs, a search bar and three buttons ('Chỉnh sửa', 'Trở về trang chủ') are visible. The main content area is titled 'CHI TIẾT LỘ TRÌNH DU LỊCH MỪNG TỐT NGHIỆP'. It displays a list of activities with their times and locations, followed by a calendar view for June 5, 2025. The calendar shows a green-highlighted block from 8:00 to 11:00 labeled 'Tham quan Rạn Nam Ô' and another block from 12:00 to 15:00 labeled 'Nghỉ dưỡng tại Imperial Hotel Huế'.

Hình 4.23: Giao diện hiển thị thông tin chi tiết của lộ trình

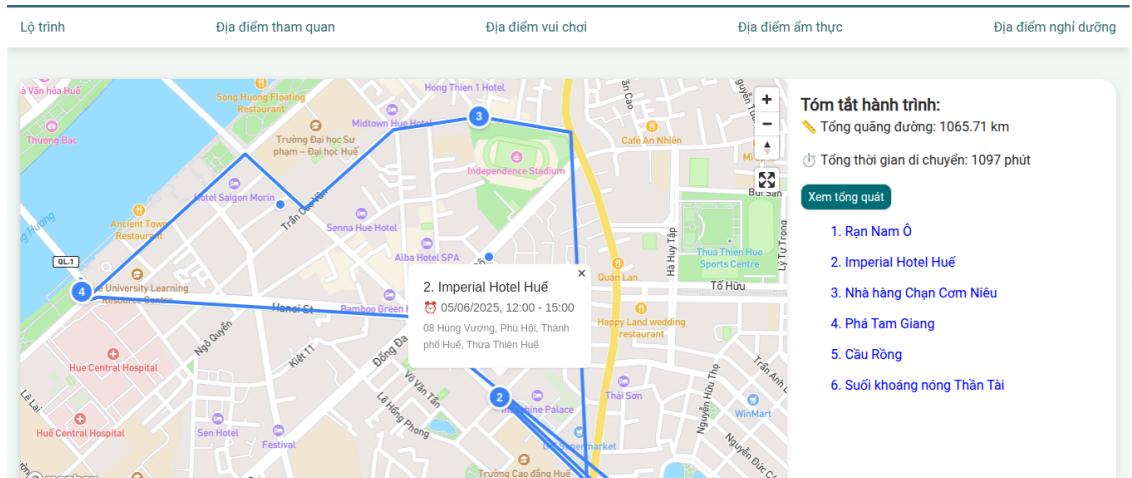
This screenshot shows a modal window titled 'Mô tả lộ trình' (Itinerary Description) overlaid on the main application. The modal contains a title 'LỊCH TRÌNH CHI TIẾT' (Detailed Schedule) and a section for 'Ngày 1 (05/06/2025)'. It lists activities: '08:00 - 11:00: Đoàn sẽ di chuyển đến tham quan Tham quan Rạn Nam Ô, một địa danh nổi bật với nét đặc trưng và giá trị văn hóa độc đáo.', '12:00 - 15:00: Quý khách sẽ có thời gian nghỉ ngơi, thư giãn tại Nghỉ dưỡng tại Imperial Hotel Huế – một nơi lý tưởng để tái tạo năng lượng sau hành trình.', and '17:00 - 20:00: Đoàn sẽ dùng bữa tại Ăn tại Nhà hàng Chợ Cồn Niêu, nơi nổi tiếng với các món ăn đặc sản mang đậm hương vị địa phương.'. Below this, there are sections for 'Ngày 2 (06/06/2025)' and 'Ngày 3 (07/06/2025)', each with its own list of activities. The background of the main application shows the same itinerary details as in Figure 4.23.

Hình 4.24: Giao diện hiển thị mô tả lộ trình

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.25: Giao diện hiển thị cẩm nang của lộ trình

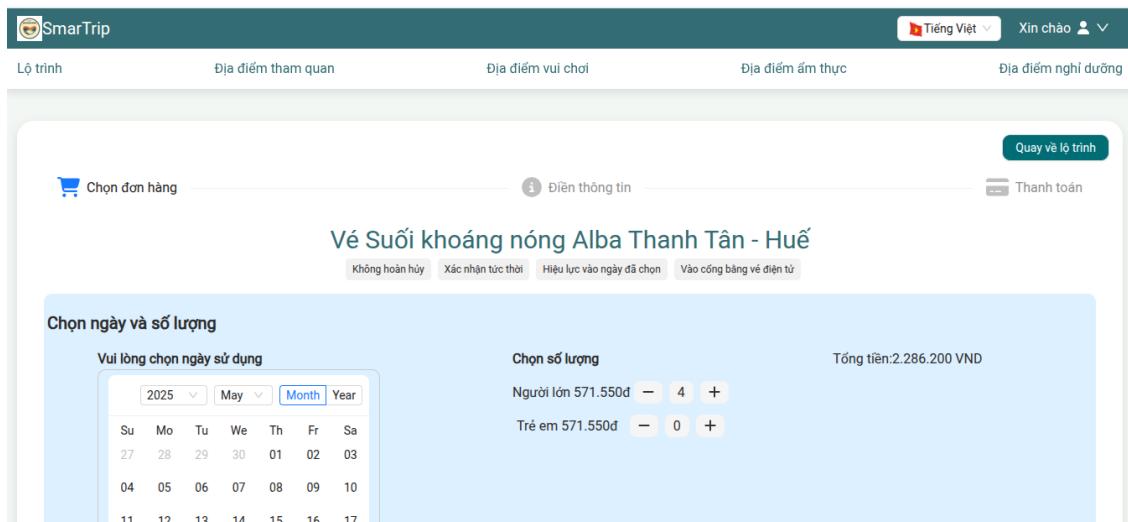


Hình 4.26: Giao diện hiển thị lộ trình trên bản đồ

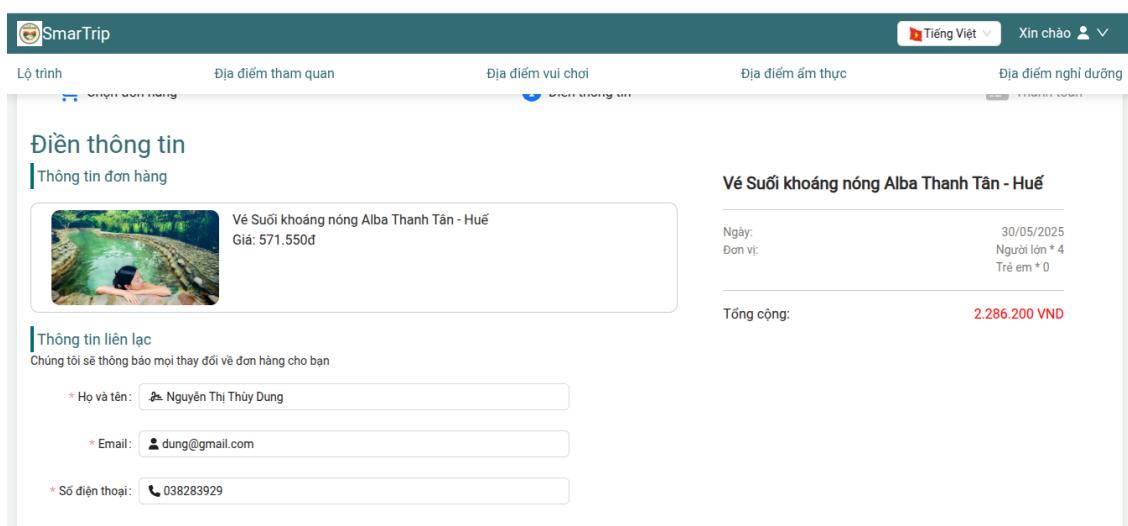
Với lộ trình đã được tạo thành công, người dùng sẽ được chuyển đến giao diện hiển thị chi tiết lộ trình với dịch vụ - khung giờ tương ứng như hình 4.23. Với lộ trình, người dùng còn có thể xem những thông tin quan trọng như sau: xem mô tả lộ trình dưới dạng tour du lịch do hệ thống tự động sinh như hình 4.24, xem cẩm nang du lịch mà hệ thống đề xuất như hình 4.25, và đặc biệt người dùng có thể theo dõi được lộ trình một cách trực quan nhất trên bản đồ như hình 4.26. Mỗi dịch vụ sẽ được hiển thị trên bản đồ, cách di chuyển, thời gian di chuyển giữa các dịch vụ cũng sẽ được hiển thị một cách trực quan.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

d, Giao diện minh họa chức năng thanh toán dịch vụ

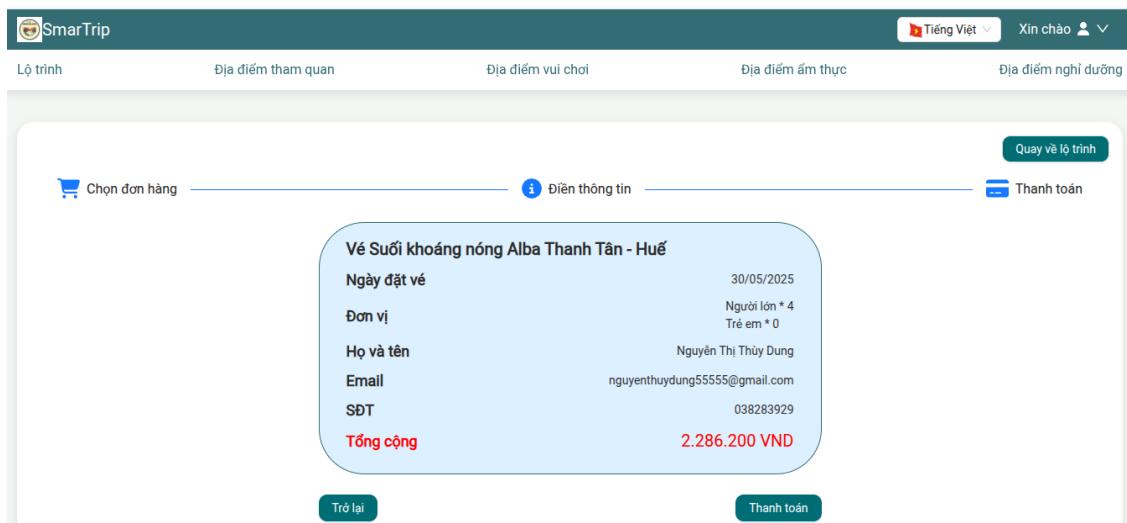


Hình 4.27: Giao diện hiển thị đặt dịch vụ với thông tin dịch vụ

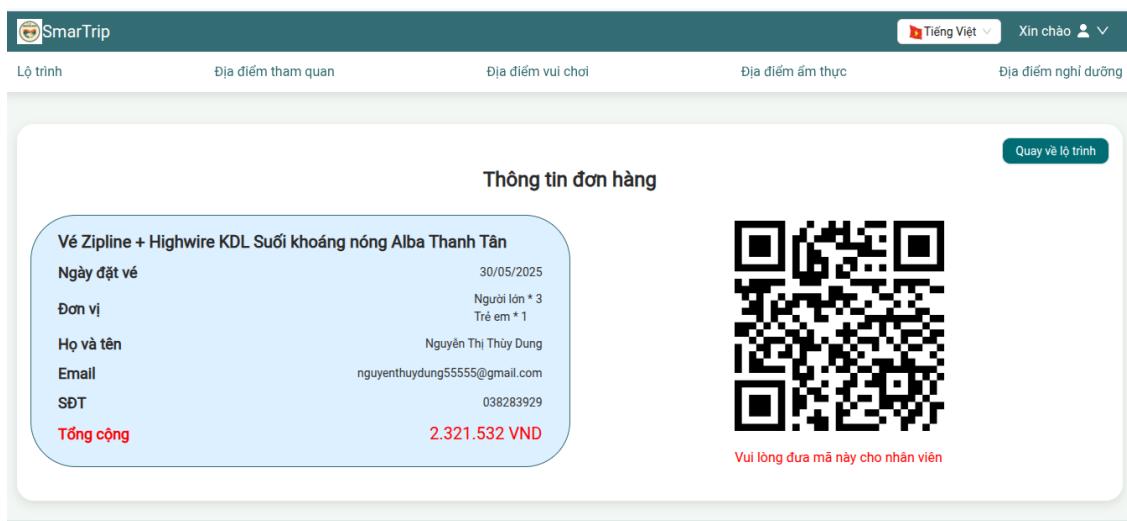


Hình 4.28: Giao diện hiển thị form nhập thông tin liên lạc khi đặt dịch vụ

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

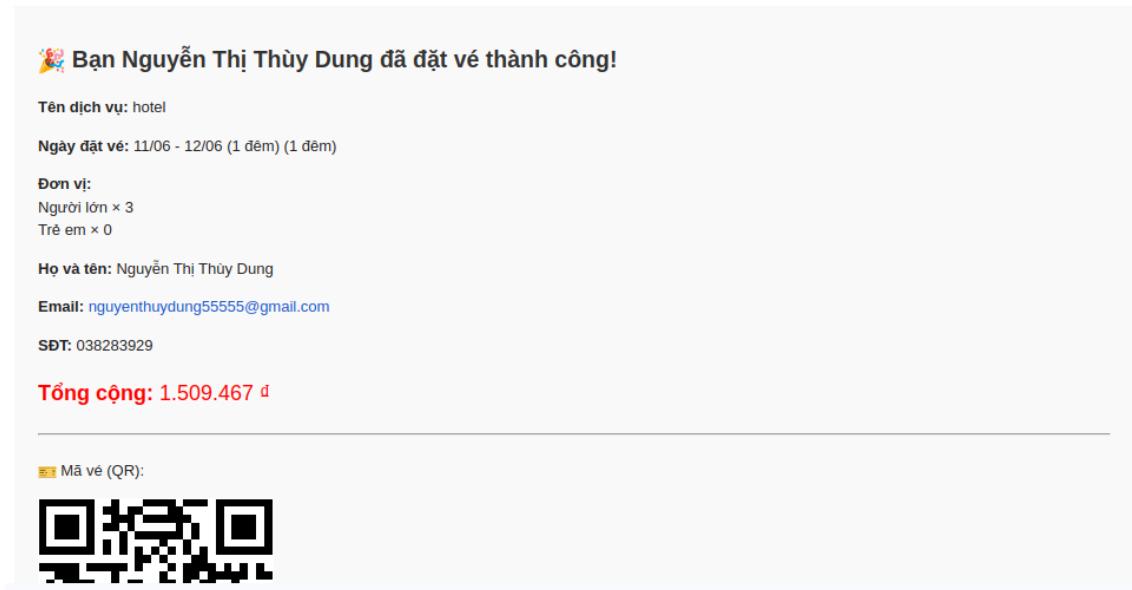


Hình 4.29: Giao diện hiển thị thông tin cần thiết khi đặt dịch vụ để người dùng xác nhận



Hình 4.30: Giao diện hiển thị kết quả đặt dịch vụ thành công

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG



Hình 4.31: Giao diện email đặt dịch vụ thành công

Người dùng có thể đặt trước vé vui chơi, đặt phòng khách sạn, nhà hàng trong lộ trình (nếu có). Giao diện 4.27 minh họa đặt vé vui chơi trong lộ trình với ngày và thời gian lây từ lộ trình. Người dùng có thể chỉnh sửa lại số lượng người khi đặt dịch vụ. Tiếp theo người dùng sẽ được yêu cầu cung cấp thông tin liên hệ như hình 4.28 bao gồm email, họ tên, số điện thoại và thông tin vé. Hệ thống sẽ hiển thị toàn bộ thông tin cho phép người dùng xác nhận một lần nữa trước khi thanh toán với vnpay. Người dùng thêm thẻ và thực hiện thanh toán bằng vnpay, khi thành công người dùng sẽ được hệ thống hiển thị thông tin vé đặt thành công như hình 4.30. Đồng thời, một email thông báo đặt dịch vụ thành công cũng được gửi email mà người dùng dùng để đặt dịch vụ.

4.4 Kiểm thử

4.4.1 Môi trường kiểm thử

Môi trường kiểm thử được thiết lập trên laptop dell, hệ điều hành ubuntu. Các thông số chi tiết như bảng dưới đây:

Thành phần	Thông tin chi tiết
Màn hình	14.0'
Chip	core i5
Hệ điều hành	Ubuntu
Ram	8Gb
Ổ cứng	SSD 256Gb

Bảng 4.15: Môi trường kiểm thử

4.4.2 Kiểm thử blackbox

Dưới đây là chi tiết các trường hợp kiểm thử:

Trường kiểm thử	Đầu vào	Đầu ra kỳ vọng	Kết quả
Tạo thông tin cơ bản của lô trình	Thông tin cơ bản của lô trình	Hiển thị thông báo chuẩn bị dữ liệu cho lô trình	Đạt
Chọn dịch vụ cho vào lô trình	Id và loại của dịch vụ	Hiển thị thông báo thêm vào lô trình thành công	Đạt
Loại dịch vụ khỏi lô trình	Id và loại của dịch vụ	Hiển thị thông báo loại khỏi lô trình thành công	Đạt
Xóa dịch vụ khỏi lô trình	Tên của dịch vụ	Dịch vụ biến mất trong danh sách dịch vụ của lô trình	Đạt
Kéo thả dịch vụ vào lịch	Tiêu đề của dịch vụ	Dịch vụ xuất hiện trong lịch	Đạt
Xóa dịch vụ khỏi lịch	Id của dịch vụ	Dịch vụ biến mất khỏi lịch	Đạt

Bảng 4.16: Kiểm thử cho chức năng thiết kế lô trình du lịch

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Trường kiểm thử	Đầu vào	Đầu ra kỳ vọng	Kết quả
Tạo mô tả cho lộ trình	Thông tin chi tiết của lộ trình	Hiển thị mô tả lộ trình như một tour du lịch	Đạt
Tạo cảm nang cho lộ trình	Thông tin chi tiết của lộ trình	Hiển thị cảm nang cho lộ trình	Đạt
Bật audio đọc cảm nang du lịch	Cảm nang du lịch	Đọc cảm nang với giọng đọc tiếng việt	Đạt
In mô tả lộ trình	Mô tả lộ trình	Lưu mô tả với tên file itinerary.docx vào máy của người dùng	Đạt
Hiển thị lộ trình trên bản đồ	Lịch trình trong lộ trình	Các dịch vụ hiển thị lần lượt trên bản đồ	Đạt

Bảng 4.17: Kiểm thử chức năng quản lý lộ trình du lịch

Trường kiểm thử	Đầu vào	Đầu ra kỳ vọng	Kết quả
Hiển thị thông tin vé	Dịch vụ vui chơi có trong lộ trình	Hiển thị thông tin vé và ngày muộn đặt vé	Đạt
Tạo thông tin liên lạc của người đặt dịch vụ	Không có	Hiển thị thông tin liên lạc của người dùng	Đạt
Thanh toán dịch vụ	Dịch vụ kèm số tiền đặt	Hiển thị thông báo thanh toán thành công và gửi email thông báo thanh toán thành công	Đạt
Hủy thanh toán dịch vụ	Dịch vụ kèm số tiền đặt	Hiển thị thông báo thanh toán thất bại	Đạt

Bảng 4.18: Kiểm thử chức năng thanh toán dịch vụ

4.5 Triển khai

Mục	Thông tin
URL Frontend Reactjs	
URL Backend ExpressJs	
URL Backend Flask	

Bảng 4.19: Triển khai hệ thống

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Chương này tập trung vào các giải pháp đã triển khai, nhấn mạnh việc tích hợp dữ liệu đa dạng để xây dựng cảm nang du lịch, cơ chế gợi ý hành trình cá nhân hóa và hiển thị lộ trình trực quan trên bản đồ. Những tính năng này nâng cao trải nghiệm người dùng trong lập kế hoạch và quản lý chuyến đi.

5.1 Thu thập dữ liệu hiệu quả nhờ Crawl4AI

5.1.1 Giới thiệu vấn đề

Việc thu thập dữ liệu từ các trang web thường dựa vào các phương pháp truyền thống như sử dụng thư viện requests kết hợp BeautifulSoup hoặc Puppeteer (Python) hay cheerio trong JS. Những phương pháp này đòi hỏi người dùng phải hiểu rõ cấu trúc HTML của trang web, xác định các CSS selector cụ thể để trích xuất dữ liệu mong muốn. Quá trình này không chỉ tốn thời gian mà còn gặp khó khăn khi xử lý các trang web có cấu trúc phức tạp, đặc biệt là những trang sử dụng các framework như React, nơi nội dung được sinh động (dynamic rendering). Hơn nữa, khi cần thu thập dữ liệu từ nhiều trang cùng lúc, các phương pháp này thường gây nghẽn tài nguyên, dẫn đến hiệu suất thấp và thời gian xử lý kéo dài.

5.1.2 Giải pháp

Để giải quyết các hạn chế trên, em đã ứng dụng công cụ Crawl4AI, một giải pháp tiên tiến giúp tối ưu hóa quá trình thu thập dữ liệu. Crawl4AI tận dụng các kỹ thuật hiện đại, kết hợp với cơ chế xử lý song song (parallel processing) thông qua semaphore, cho phép rút ngắn đáng kể thời gian thu thập dữ liệu. Công cụ này mang lại các lợi ích vượt trội như sau:

- Thu thập toàn diện và thông minh:** Crawl4AI tự động trích xuất toàn bộ nội dung hiển thị trên trang web, bao gồm văn bản, markdown, đường link, hình ảnh liên quan, và các thành phần khác, bất kể trang web được xây dựng bằng HTML tĩnh hay các framework động như React.
- Xử lý song song hiệu quả:** Nhờ tích hợp cơ chế semaphore, Crawl4AI cho phép chạy đồng thời nhiều tiến trình thu thập dữ liệu mà không gây quá tải tài nguyên hệ thống.
- Thay thế phương pháp truyền thống:** Với khả năng tự động hóa và tốc độ xử lý cao, Crawl4AI đang dần thay thế các công cụ crawl truyền thống, mang lại hiệu quả vượt trội.

Bên cạnh đó, Crawl4AI còn hỗ trợ tích hợp với các mô hình ngôn ngữ lớn (LLM) để trích xuất thông tin có chọn lọc, như tên gọi, địa điểm, mô tả sản phẩm, hoặc

các trường dữ liệu cụ thể khác, thông qua việc sử dụng các prompt tùy chỉnh. Tuy nhiên, phương pháp này đòi hỏi tài nguyên tính toán lớn và thời gian xử lý lâu, khiến nó chưa thực sự phù hợp cho các tác vụ thu thập dữ liệu quy mô lớn hoặc cần tốc độ cao. Với việc sử dụng máy tính cá nhân với các thông số em đã nêu trên thì dù cách này là một phương pháp mạnh mẽ nhưng chưa thực sự phù hợp với em.

5.1.3 Kết quả đạt được

So với các phương pháp truyền thống, việc sử dụng Crawl4AI đã mang lại những cải thiện đáng kể. Độ chính xác trong việc tách nội dung chính cũng được cải thiện rõ rệt, loại bỏ các dữ liệu không cần thiết và tạo điều kiện thuận lợi cho các bước xử lý tiếp theo như chuẩn hóa dữ liệu, phân tích, hoặc tạo embedding. Nhờ khả năng thu thập toàn diện và tốc độ vượt trội, Crawl4AI không chỉ nâng cao hiệu quả mà còn mở ra tiềm năng ứng dụng trong các dự án phân tích dữ liệu quy mô lớn.

5.2 Tích hợp RAG để cung cấp thông tin chính xác hơn từ cẩm nang du lịch

5.2.1 Giới thiệu vấn đề

Để giải quyết vấn đề này, tôi đã thiết kế và triển khai một quy trình **Retrieval-Augmented Generation (RAG)**, kết hợp việc truy xuất thông tin từ cơ sở dữ liệu với khả năng sinh văn bản của LLM. Quy trình này đảm bảo thông tin cung cấp cho người dùng không chỉ chính xác mà còn mang tính bản địa, dựa trên dữ liệu từ các nguồn đáng tin cậy. Dưới đây là các bước cụ thể:

5.2.2 Giải pháp

- **Thu thập dữ liệu từ các cẩm nang du lịch:** Dữ liệu được thu thập từ hai nguồn chính là khamphadisan.vn và checkintravel.vn, vốn là các trang web cung cấp thông tin du lịch Việt Nam chi tiết và đáng tin cậy. Em sử dụng Crawl4AI để thu thập toàn bộ nội dung hiển thị trên các trang, bao gồm văn bản, tiêu đề, mô tả địa danh, hình ảnh, và các liên kết liên quan. Crawl4AI giúp tự động trích xuất nội dung mà không cần phân tích cấu trúc HTML hay CSS selector.
- **Làm sạch dữ liệu** Dữ liệu thu thập được thường chứa các thành phần không cần thiết như thẻ HTML, markdown, hoặc quảng cáo. Em áp dụng các kỹ thuật xử lý văn bản để: Loại bỏ thẻ HTML, markdown, và các ký tự không mong muốn. Chuẩn hóa văn bản (chuyển về dạng chữ thường, loại bỏ dấu cách thừa, sửa lỗi chính tả cơ bản). Trích xuất các từ khóa mô tả đặc trưng (ví dụ: địa danh, đặc sản, hoạt động nổi bật) bằng cách sử dụng các phương pháp như phân tích ngữ nghĩa đơn giản.
- **Chia văn bản thành các đoạn nhỏ** Để tối ưu hóa việc lưu trữ và truy xuất,

văn bản được chia thành các đoạn có độ dài tối đa 324 token (tương đương khoảng 200-300 từ, tùy thuộc vào ngôn ngữ). Điều này giúp đảm bảo rằng các đoạn văn bản đủ ngắn để xử lý nhanh, nhưng vẫn chứa đủ ngữ cảnh để có ý nghĩa khi truy xuất.

- **Chuyển văn bản thành vector** Em sử dụng mô hình keepitreal/vietnamesesbert, một mô hình Sentence-BERT được tối ưu hóa cho tiếng Việt, để chuyển các đoạn văn bản thành các vector ngữ nghĩa. Mô hình này giúp biểu diễn văn bản dưới dạng vector số, cho phép so sánh ngữ nghĩa giữa các đoạn văn bản và câu hỏi của người dùng một cách hiệu quả.
- **Lưu trữ vector vào ChromaDB** Các vector được lưu trữ trong ChromaDB, một cơ sở dữ liệu vector mã nguồn mở, được thiết kế để hỗ trợ tìm kiếm ngữ nghĩa nhanh chóng. ChromaDB cho phép truy xuất các đoạn văn bản liên quan nhất dựa trên độ tương đồng ngữ nghĩa với câu hỏi của người dùng.
- **Tích hợp RAG để sinh lời khuyên** Khi người dùng nhập một lộ trình du lịch hoặc câu hỏi (ví dụ: "Gợi ý địa điểm tham quan ở Đà Lạt"), hệ thống thực hiện các bước sau: Chuyển câu hỏi của người dùng thành vector bằng mô hình vietnamese-sbert. Tìm kiếm trong ChromaDB để lấy ra các đoạn văn bản có độ tương đồng cao nhất với câu hỏi. Cung cấp các đoạn văn bản này làm ngữ cảnh cho LLM (qua OpenRouter) để sinh ra câu trả lời chính xác, có dẫn nguồn rõ ràng.
- **So sánh hiệu suất các mô hình LLM** Tôi đã thử nghiệm và so sánh hiệu suất của các mô hình LLM trên nền tảng OpenRouter, bao gồm GPT-3.5, Claude, và DeepSeek. Các tiêu chí đánh giá bao gồm: Độ tự nhiên: Mức độ trôi chảy và dễ hiểu của câu trả lời. Thời gian phản hồi: Tốc độ xử lý câu hỏi và sinh câu trả lời. Mức độ chính xác: Khả năng cung cấp thông tin phù hợp với dữ liệu đã thu thập và ngữ cảnh địa phương.

5.2.3 Kết quả đạt được

Hệ thống RAG được triển khai đã mang lại những cải tiến vượt trội so với các phương pháp sử dụng LLM thông thường:

- **Tính xác thực cao:** Các câu trả lời được sinh ra dựa trên dữ liệu từ khamphadisan.vn và checkintravel.vn, đảm bảo thông tin chính xác, phù hợp với địa danh và mang tính bản địa.
- **Khác biệt so với hệ thống sinh văn bản tự do:** Không giống các LLM thông thường có thể tạo ra thông tin không kiểm chứng, hệ thống RAG chỉ sử dụng dữ liệu đã được xác thực, loại bỏ nguy cơ sai lệch.

5.3 Tích hợp Text-to-Speech với giọng đọc tiếng Việt

5.3.1 Giới thiệu vấn đề

Trong các ứng dụng du lịch, việc cung cấp thông tin dưới dạng văn bản thường không phù hợp với một số nhóm người dùng, đặc biệt là người lớn tuổi hoặc những người đang di chuyển (ví dụ: khi lái xe hoặc tham quan). Những người dùng này khó có thể đọc các cẩm nang du lịch hoặc lịch trình chi tiết trên màn hình. Do đó, tích hợp tính năng chuyển văn bản thành giọng nói (Text-to-Speech - TTS) trở thành một giải pháp cần thiết để nâng cao khả năng tiếp cận và cải thiện trải nghiệm người dùng. Tuy nhiên, các giải pháp TTS phổ biến như Web Speech API thường không hỗ trợ tốt tiếng Việt, dẫn đến giọng đọc thiếu tự nhiên, méo mó hoặc mang âm điệu tiếng Anh, gây khó chịu cho người nghe.

5.3.2 Giải pháp

Để giải quyết vấn đề này, em đã khảo sát và thử nghiệm nhiều dịch vụ TTS, bao gồm Voice Bee, Google Cloud Text-to-Speech, và FPT.AI. Sau khi đánh giá, em quyết định sử dụng FPT.AI Text-to-Speech API vì các ưu điểm nổi bật:

- Hỗ trợ giọng đọc tiếng Việt tự nhiên, với nhiều tùy chọn giọng (nam, nữ, vùng miền như Bắc, Trung, Nam).
- Cho phép tùy chỉnh tốc độ, nhịp điệu và âm lượng.
- Cung cấp gói miễn phí phù hợp cho các ứng dụng thử nghiệm hoặc quy mô nhỏ.
- Dễ dàng tích hợp thông qua API RESTful, trả về file âm thanh (.mp3 hoặc .wav) hoặc URL phát trực tiếp.

Quy trình tích hợp TTS với FPT.AI được thực hiện qua các bước cụ thể sau:

- **Chuẩn bị văn bản cần chuyển đổi** Văn bản đầu vào (gợi ý được lấy từ hệ thống RAG (như mô tả trong mục 5.2). Văn bản được chuẩn hóa để đảm bảo không chứa ký tự đặc biệt hoặc định dạng gây lỗi khi gửi đến API.
- **Gửi yêu cầu đến FPT.AI TTS Endpoint** Sử dụng API key của FPT.AI, em gửi một yêu cầu POST đến endpoint TTS (<https://api.fpt.ai/hmi/tts/v5>). Yêu cầu bao gồm: Văn bản cần chuyển thành giọng nói. Các tham số tùy chỉnh: giọng đọc (ví dụ: "leminh" cho giọng nam miền Bắc), tốc độ nói (speed, từ 0.8 đến 1.2), và định dạng đầu ra (mp3 hoặc wav).
- **API key để xác thực yêu cầu** Nhận và xử lý file âm thanh FPT.AI xử lý yêu cầu và trả về một URL dẫn đến file âm thanh (.mp3 hoặc .wav) được lưu trữ tạm thời trên máy chủ của họ. Hệ thống tải file này về hoặc sử dụng URL để

phát trực tiếp.

- **Tích hợp vào giao diện người dùng** Trong giao diện ứng dụng (web hoặc mobile), em tích hợp một trình phát âm thanh (audio player) sử dụng audio. Khi người dùng nhấn nút "Phát âm thanh" hoặc yêu cầu tự động phát, hệ thống kiểm tra trạng thái sẵn sàng của file âm thanh (bằng cách gửi yêu cầu kiểm tra URL) và phát ngay khi file có sẵn.

5.3.3 Kết quả đạt được

Việc tích hợp FPT.AI TTS mang lại những cải tiến đáng kể:

- **Giọng đọc tự nhiên và rõ ràng:** Giọng đọc tiếng Việt từ FPT.AI (đặc biệt là các giọng miền Bắc và miền Nam) có chất lượng cao, gần giống giọng người thật, không bị méo tiếng hay mang âm điệu tiếng Anh như Web Speech API.
- **Trải nghiệm người dùng cải thiện:** Tính năng TTS cho phép du khách nghe thông tin về địa điểm, lịch trình, hoặc mèo du lịch trong khi di chuyển, giúp tăng tính tiện lợi và khả năng tiếp cận.
- **Hiệu suất ổn định:** Thời gian phản hồi của API nhanh (thường dưới 1 phút để tạo file âm thanh), với file đầu ra có chất lượng cao và kích thước nhỏ, phù hợp cho cả thiết bị di động.

5.4 Đề xuất địa điểm gần để tối ưu lộ trình

Đề xuất địa điểm gần để tối ưu lộ trình

5.4.1 Giới thiệu vấn đề

Khi lập kế hoạch du lịch, người dùng thường gặp khó khăn trong việc xác định các địa điểm nên ghé thăm tiếp theo hoặc cách sắp xếp lộ trình để tiết kiệm thời gian di chuyển. Ví dụ, sau khi tham quan một địa điểm, họ có thể không biết nhà hàng nào gần đó đang mở cửa hoặc khách sạn nào phù hợp để nghỉ ngơi. Việc thiếu thông tin về các địa điểm lân cận và không tối ưu hóa lộ trình dẫn đến lãng phí thời gian, tăng chi phí di chuyển, và làm giảm trải nghiệm du lịch. Đặc biệt, với các du khách không quen thuộc với khu vực, việc tự tìm kiếm và kết nối các địa điểm thuộc các danh mục khác nhau (tham quan, ăn uống, lưu trú) là một thách thức lớn.

5.4.2 Giải pháp

Để giải quyết vấn đề này, em đã phát triển một hệ thống gợi ý địa điểm thông minh dựa trên tọa độ địa lý và các yếu tố ngữ cảnh như loại hình dịch vụ. Hệ thống này không chỉ đề xuất các địa điểm gần vị trí hiện tại của người dùng mà còn đảm bảo chúng thuộc các danh mục dịch vụ khác nhau (ví dụ: từ điểm tham quan đến nhà hàng hoặc khách sạn) để tạo ra một lộ trình hợp lý và liền mạch. Quy trình

được thực hiện qua các bước cụ thể sau:

- **Thu thập và chuẩn hóa dữ liệu địa điểm** Dữ liệu về các địa điểm du lịch (điểm tham quan, nhà hàng, khách sạn,...) được thu thập từ các nguồn đáng tin cậy như bestprice và travelviet.net. Mỗi địa điểm được chuẩn hóa để bao gồm: Tên địa điểm. Tọa độ địa lý (kinh độ và vĩ độ). Loại hình dịch vụ (tham quan, ăn uống, lưu trú, v.v.). Thời gian hoạt động (giờ mở cửa, đóng cửa). Các thông tin bổ sung như đánh giá, giá cả, hoặc đặc điểm nổi bật.
- **Tính toán khoảng cách địa lý** Khi người dùng chọn một địa điểm, hệ thống sử dụng công thức

Gọi ϕ_1, ϕ_2 là vĩ độ, λ_1, λ_2 là kinh độ của hai điểm (đơn vị radian), khi đó khoảng cách d giữa hai điểm được tính theo công thức Haversine:

$$\Delta\phi = \phi_2 - \phi_1, \quad \Delta\lambda = \lambda_2 - \lambda_1$$

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \arctan 2\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$d = R \cdot c$$

Trong đó R là bán kính Trái Đất (thường lấy $R = 6371$ km).

để tính khoảng cách địa lý giữa địa điểm đó và các địa điểm khác trong cơ sở dữ liệu. Chỉ những địa điểm nằm trong bán kính nhất định (ví dụ: 5km) được xem xét.

- **Lọc địa điểm theo ngữ cảnh** Các địa điểm được lọc dựa trên: Loại hình dịch vụ để đảm bảo gợi ý địa điểm thuộc danh mục khác với địa điểm hiện tại (ví dụ: từ điểm tham quan đến nhà hàng hoặc khách sạn). Các địa điểm được sắp xếp theo khoảng cách (gần nhất đến xa nhất). Hệ thống trả về danh sách gợi ý, kèm theo thông tin chi tiết như tên, địa chỉ, đánh giá.
- **Tích hợp vào giao diện người dùng** Gợi ý được hiển thị trên giao diện web hoặc ứng dụng di động, cho phép người dùng chọn địa điểm tiếp theo.

5.4.3 Kết quả đạt được

Hệ thống gợi ý địa điểm mang lại những lợi ích nổi bật:

- **Tính thực tiễn cao:** Người dùng nhận được các gợi ý địa điểm gần vị trí hiện tại, phù hợp với nhu cầu (ăn uống, nghỉ ngơi, tham quan) và thời gian (đang mở cửa). Ví dụ, sau khi tham quan Hồ Xuân Hương ở Đà Lạt, hệ thống gợi ý một nhà hàng địa phương đang mở cửa phục vụ bữa trưa trong vòng 2km.
- **Tối ưu lộ trình:** Lộ trình được sắp xếp hợp lý, giảm thời gian và chi phí di chuyển, giúp du khách tận dụng tối đa thời gian.
- **Khả năng mở rộng:** Hệ thống có thể tích hợp thêm các yếu tố như dự báo thời tiết, tình hình giao thông, hoặc đánh giá từ cộng đồng để nâng cao chất lượng gợi ý.

5.5 Trực quan hóa lộ trình trên bản đồ

5.5.1 Giới thiệu vấn đề

Khi lập kế hoạch du lịch, người dùng thường gặp khó khăn trong việc hình dung toàn cảnh lộ trình, bao gồm thứ tự các địa điểm, khoảng cách giữa chúng, và cách di chuyển hiệu quả. Các thông tin dạng văn bản hoặc bảng biểu không đủ trực quan, đặc biệt với những người không quen thuộc với khu vực. Ngoài ra, việc thiếu hỗ trợ dẫn đường cụ thể (navigation) khiến người dùng phải chuyển đổi giữa các ứng dụng, gây mất thời gian và làm gián đoạn trải nghiệm. Do đó, cần một giải pháp trực quan hóa lộ trình trên bản đồ, hiển thị rõ ràng các địa điểm, thời gian hoạt động, và cung cấp khả năng dẫn đường liền mạch.

5.5.2 Giải pháp

Để giải quyết vấn đề này, em đã tích hợp Mapbox, một nền tảng bản đồ mạnh mẽ, để trực quan hóa lộ trình du lịch và hỗ trợ người dùng lập kế hoạch hiệu quả. Mapbox được chọn vì khả năng tùy chỉnh cao, giao diện tương tác thân thiện, và hỗ trợ tốt các tính năng không gian. Hệ thống cũng tích hợp với Google Maps để cung cấp dẫn đường chi tiết khi người dùng cần di chuyển. Quy trình thực hiện bao gồm các bước cụ thể sau:

- **Tính toán khoảng cách và thời gian di chuyển** Sử dụng Mapbox Directions API để tính toán khoảng cách và thời gian di chuyển giữa các địa điểm liên tiếp trong lộ trình. API này cung cấp thông tin về quãng đường (km) và thời gian dự kiến (phút), dựa trên phương tiện di chuyển (đi bộ, xe máy, ô tô). Hiển thị lộ trình trên bản đồ Mapbox
- **Giao diện bản đồ được xây dựng bằng Mapbox GL JS, hiển thị:** Các điểm đánh dấu (markers): Mỗi địa điểm được biểu thị bằng một điểm đánh dấu trên bản đồ, kèm thông tin chi tiết (tên, thời gian hoạt động, mô tả) khi nhấp vào. Đường nối (route): Lộ trình được vẽ dưới dạng đường polyline nối các địa

điểm theo thứ tự thời gian. Mốc thời gian: Mỗi địa điểm hiển thị thời gian hoạt động trong cửa sổ thông tin (popup). Khoảng cách và thời gian di chuyển: Thông tin này được hiển thị bên cạnh đường nối hoặc trong popup của địa điểm tiếp theo.

Tích hợp dẫn đường với Google Maps

Khi người dùng chọn một địa điểm để di chuyển, hệ thống tạo một liên kết đến Google Maps với tọa độ nguồn và đích, cho phép mở ứng dụng Google Maps (trên web hoặc di động) để dẫn đường chi tiết. Liên kết này sử dụng định dạng URL như <https://www.google.com/maps/dir/?api=1&origin=lat,lng&destination=lat,lng>

- **Xây dựng giao diện tương tác** Giao diện web sử dụng HTML, CSS, và JavaScript với Mapbox GL JS để hiển thị bản đồ tương tác. Người dùng có thể: Phóng to/thu nhỏ, kéo bản đồ để xem chi tiết. Nhấp vào các điểm đánh dấu để xem thông tin. Nhấn nút dẫn đường để mở Google Maps.

5.5.3 Kết quả đạt được

Hệ thống trực quan hóa lộ trình mang lại những lợi ích nổi bật:

- **Trực quan và dễ hiểu**: Người dùng dễ dàng hình dung toàn bộ lộ trình qua giao diện bản đồ, với các địa điểm được đánh dấu rõ ràng và đường nối trực quan.
- **Dẫn đường liền mạch**: Tích hợp với Google Maps cho phép người dùng chuyển sang dẫn đường chi tiết chỉ bằng một cú nhấp, tăng tính tiện lợi.
- **Cải thiện trải nghiệm**: So với các bảng văn bản hoặc danh sách tĩnh, bản đồ tương tác mang lại trải nghiệm sinh động, giúp người dùng cảm thấy tự tin hơn khi khám phá các địa điểm mới.

Khả năng mở rộng: Hệ thống có thể tích hợp thêm các tính năng như hiển thị thời tiết, tình trạng giao thông, hoặc gợi ý thay đổi lộ trình dựa trên thời gian thực.

5.6 Thiết kế giao diện responsive cho cả máy tính và điện thoại

5.6.1 Giới thiệu vấn đề

Hệ thống hỗ trợ du lịch cần đáp ứng nhu cầu sử dụng linh hoạt trên nhiều thiết bị, từ máy tính để lập kế hoạch trước chuyến đi đến điện thoại khi người dùng di chuyển. Tuy nhiên, sự khác biệt về kích thước màn hình, phương thức tương tác (chuột so với chạm), và thói quen sử dụng giữa các thiết bị đòi hỏi giao diện phải được tối ưu hóa để đảm bảo trải nghiệm mượt mà và nhất quán. Một giao diện không responsive có thể gây khó khăn cho người dùng, chẳng hạn như văn bản quá nhỏ trên điện thoại, nút bấm khó chạm, hoặc bố cục lộn xộn trên màn hình lớn. Do

đó, việc thiết kế một giao diện responsive là yếu tố then chốt để tăng khả năng tiếp cận và nâng cao trải nghiệm người dùng.

5.6.2 Giải pháp

Để đáp ứng nhu cầu này, em đã áp dụng các kỹ thuật responsive design sử dụng HTML, CSS, và JavaScript, với trọng tâm là đảm bảo giao diện hoạt động hiệu quả trên cả máy tính và điện thoại. Quy trình thiết kế bao gồm các bước cụ thể sau:

Sử dụng CSS Media Queries để điều chỉnh bố cục Media queries được sử dụng để áp dụng các quy tắc CSS khác nhau dựa trên kích thước màn hình, độ phân giải, hoặc loại thiết bị.

- Trên máy tính (màn hình lớn hơn 1024px), hiển thị bố cục dạng lưới (grid) với nhiều cột để tận dụng không gian. Trên điện thoại (màn hình nhỏ hơn 768px), chuyển sang bố cục cột đơn, ưu tiên hiển thị dọc để dễ đọc và cuộn.
- Sắp xếp lại các thành phần giao diện Các thành phần giao diện (như bản đồ, danh sách địa điểm, hoặc nút điều khiển) được tái cấu trúc để phù hợp với màn hình nhỏ. Với bản đồ thu nhỏ hoặc chuyển sang chế độ toàn màn hình trên điện thoại, với các nút điều khiển lớn hơn để dễ chạm. Với danh sách địa điểm chuyển từ hiển thị dạng lưới sang danh sách dọc, với các mục được xếp chồng để dễ cuộn.

5.6.3 Kết quả đạt được

Hệ thống giao diện responsive mang lại những lợi ích vượt trội:

- **Hoạt động mượt mà trên mọi thiết bị:** Giao diện hiển thị tốt trên máy tính, tablet, và điện thoại, với bố cục tự động điều chỉnh theo kích thước màn hình.
- **Khả năng mở rộng:** Giao diện có thể tích hợp thêm các tính năng như thông báo đẩy (push notifications) trên di động hoặc chế độ offline để hỗ trợ người dùng ở khu vực có kết nối yếu.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chương này tổng kết kết quả đạt được, làm nổi bật điểm mạnh của hệ thống, đồng thời chỉ ra các hạn chế còn tồn tại. Từ đó, đề xuất các hướng phát triển tương lai để mở rộng tính năng, cải thiện hiệu năng và đáp ứng tốt hơn nhu cầu đa dạng của người dùng.

6.1 Kết luận

Trong quá trình thực hiện đồ án, em đã phát triển một hệ thống hỗ trợ du lịch toàn diện, tích hợp các công nghệ hiện đại để đáp ứng nhu cầu của người dùng trong việc lập kế hoạch và trải nghiệm du lịch. Hệ thống bao gồm các chức năng chính: thiết kế lộ trình du lịch với nhiều điểm đến, quản lý lộ trình đã thiết kế, các tiện ích đi kèm như đặt dịch vụ trong lộ trình, sinh cẩm nang cho lộ trình. Các chức năng này được xây dựng để tạo ra một trải nghiệm du lịch thông minh, trực quan, và dễ tiếp cận.

So sánh với các nghiên cứu và sản phẩm tương tự So với các hệ thống du lịch hiện có như Google Maps, TripAdvisor, hoặc Traveloka, sản phẩm của em có một số điểm nổi bật:

- **Tích hợp RAG với dữ liệu bản địa:** Không giống các hệ thống phổ biến chủ yếu dựa trên dữ liệu tổng quát, hệ thống của em sử dụng RAG để cung cấp thông tin chính xác từ các nguồn cẩm nang du lịch Việt Nam như khamphadisan.vn và checkintravel.vn. Điều này đảm bảo tính bản địa và độ chính xác cao, đặc biệt phù hợp với người dùng Việt Nam.
- **Tối ưu lộ trình thông minh:** Hệ thống gợi ý địa điểm gần dựa trên tọa độ và thời gian hoạt động, một tính năng mà các ứng dụng như TripAdvisor chưa tối ưu hóa tốt, đặc biệt khi cần kết hợp nhiều danh mục dịch vụ (tham quan, ăn uống, lưu trú).

Tuy nhiên, so với các sản phẩm thương mại, hệ thống của em còn hạn chế về quy mô dữ liệu (chỉ sử dụng hai nguồn chính) và chưa tích hợp các tính năng thời gian thực như dự báo thời tiết hoặc tình trạng giao thông, hoặc tự đề xuất lộ trình cho người dùng.

- **Quy mô dữ liệu:** Hiện tại, dữ liệu các dịch vụ chỉ được thu thập từ hai nguồn bestprice và travelviet và dữ liệu cẩm nang từ hai nguồn checkintravel và khamphadisan, chưa đủ phong phú để bao quát tất cả các điểm đến tại Việt Nam.
- **Tính cá nhân hóa:** Hệ thống chưa hỗ trợ tốt việc cá nhân hóa dựa trên sở thích

chi tiết của người dùng (ví dụ: sở thích ẩm thực, ngân sách, hoặc loại hình du lịch).

6.2 Hướng phát triển

Để hoàn thiện và nâng cấp hệ thống, em định hướng các công việc sau trong tương lai:

- **Mở rộng nguồn dữ liệu** Tích hợp thêm các nguồn cẩm nang du lịch khác (ví dụ: vietnamtourism.gov.vn, dulichvietnam.com.vn) để tăng độ bao phủ và đa dạng thông tin.
- **Cải thiện gợi ý lộ trình** Tích hợp thêm tiêu chí lọc như ngân sách, đánh giá người dùng, hoặc loại hình du lịch (phiêu lưu, nghỉ dưỡng, văn hóa). Sử dụng thuật toán tối ưu hóa lộ trình (như Traveling Salesman Problem) để đề xuất thứ tự địa điểm tốt nhất dựa trên khoảng cách và thời gian.
- **Tích hợp nhận diện giọng nói** Thêm tính năng nhận diện giọng nói (Speech-to-Text) để người dùng có thể nhập câu hỏi hoặc yêu cầu lộ trình bằng giọng nói, phù hợp khi đang di chuyển. Kết hợp với FPT.AI Speech-to-Text để đảm bảo hỗ trợ tiếng Việt chính xác.

PHỤ LỤC