

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KÌ**  
**Hệ thống Du lịch Thông minh**

CSC10014 - Tư duy Tính toán

**Thành viên:**

24127457 - Nguyễn Hồ Nam  
24127053 - Trần Hoàng Minh Khang  
24127575 - Nguyễn Thị Hồng Trúc  
24127367 - Đặng Trung Hiện  
24127561 - Đặng Thế Tony  
24127402 - Phùng Quốc Huy  
24127324 - Vũ Thị Ánh

**Giáo viên hướng dẫn:**

**GV Lí thuyết:** TS. Châu Thành Đức  
**GV Thực hành:** Đỗ Đức Hào  
Trần Hoàng Quân

Thành phố Hồ Chí Minh, ngày 10, tháng 12, năm 2025

# Mục lục

<b>1 Giới thiệu</b>	<b>2</b>
1.1 Đặt vấn đề . . . . .	2
1.2 Mục tiêu đề tài . . . . .	3
1.3 Đối tượng hướng đến . . . . .	3
1.4 Phạm vi thực hiện . . . . .	4
1.5 Tổng quan giải pháp và công nghệ liên quan . . . . .	4
<b>2 Phân tích yêu cầu hệ thống</b>	<b>5</b>
2.1 Yêu cầu chức năng (Functional Requirements) . . . . .	5
2.2 Yêu cầu phi chức năng (Non-functional Requirements) . . . . .	6
2.3 Mô hình hóa Use Case . . . . .	6
2.3.1 Sơ đồ Use Case tổng quan . . . . .	6
2.3.2 Đặc tả Use Case . . . . .	7
<b>3 Thiết kế hệ thống</b>	<b>10</b>
3.1 Kiến trúc hệ thống tổng quan . . . . .	10
3.2 Thiết kế chi tiết . . . . .	11
3.2.1 Sơ đồ lớp (Class Diagram) . . . . .	11
3.2.2 Sơ đồ tuần tự (Sequence Diagram) . . . . .	12
3.2.3 3.3 Thiết kế giao diện . . . . .	13
<b>4 Quá trình phát triển</b>	<b>16</b>
4.1 Phân công thành viên . . . . .	16
4.2 Kết quả đã đạt được . . . . .	17
4.3 Kế hoạch giai đoạn tiếp theo . . . . .	17
4.4 Khó khăn và rủi ro . . . . .	18
<b>References</b>	<b>19</b>

# Chương 1

## Giới thiệu

Chương này trình bày bối cảnh hình thành hệ thống Du lịch Thông minh (Smart Tourism System), nêu rõ lý do hệ thống cần được xây dựng, xác định mục tiêu kỹ thuật của đề tài, định nghĩa đối tượng sử dụng, phạm vi triển khai, đồng thời phân tích các giải pháp và công nghệ hiện có để làm nổi bật tính thiết thực của hệ thống được đề xuất.

### 1.1 Đặt vấn đề

Trong thời đại công nghệ số, thông tin về du lịch ngày càng phong phú, đa dạng và thay đổi nhanh chóng. Tuy nhiên, chính sự dư thừa thông tin lại khiến người dùng gặp khó khăn trong việc lựa chọn điểm đến và lên kế hoạch du lịch phù hợp. Một số vấn đề cụ thể mà khách du lịch thường đối mặt bao gồm:

- **Quá tải thông tin** (Information Overload): Người dùng phải tự tìm kiếm hàng trăm bài viết, video, hoặc đánh giá trước khi ra quyết định.
- **Khó khăn trong cá nhân hóa kế hoạch du lịch**: Mỗi người có sở thích, thời gian và ngân sách khác nhau, khiến việc tạo lịch trình tối ưu trở nên phức tạp.
- **Thiếu sự kết nối giữa các dịch vụ**: Thông tin về địa điểm, khách sạn, ẩm thực thường nằm rời rạc, không có hệ thống tổng hợp thông minh.
- **Thiếu công cụ hỗ trợ khoảng cách**: Việc tìm khách sạn gần điểm tham quan phù hợp (<10km) vẫn phải làm thủ công.

Những hạn chế trên cho thấy nhu cầu rõ ràng về một hệ thống thông minh có khả năng xử lý dữ liệu, phân tích sở thích và tự động đề xuất các lựa chọn phù hợp cho người dùng.

## 1.2 Mục tiêu đề tài

Dựa trên các vấn đề đã nêu, đề tài “Hệ thống Du lịch Thông minh” được xây dựng với các mục tiêu sau:

- **Xây dựng hệ thống gợi ý địa điểm du lịch** dựa trên sở thích cá nhân và hành vi người dùng.
- **Phát triển chức năng gợi ý món ăn** theo khẩu vị và loại ẩm thực yêu thích.
- **Thiết kế mô-đun gợi ý khách sạn gần địa điểm du lịch** với bán kính dưới 10km.
- **Xây dựng tính năng tạo lịch trình du lịch thông minh** (Smart Itinerary) dựa trên thời gian, sở thích và vị trí địa lý.
- **Mô hình hóa và triển khai thuật toán mô phỏng hệ thống gợi ý** sử dụng tư duy tính toán.
- **Thiết kế giao diện trực quan và dễ sử dụng** nhằm nâng cao trải nghiệm người dùng.

Các mục tiêu trên được xây dựng để giải quyết trực tiếp các pain points của khách du lịch và thể hiện rõ yêu cầu kỹ thuật mà đồ án hướng tới.

## 1.3 Đối tượng hướng đến

Hệ thống được thiết kế dành cho nhiều nhóm người dùng có nhu cầu lập kế hoạch du lịch một cách nhanh chóng và cá nhân hoá, bao gồm:

- **Khách du lịch tự túc:** muốn tự lên kế hoạch nhưng không có nhiều thời gian.
- **Nhóm bạn trẻ yêu thích khám phá,** thích trải nghiệm địa điểm và ẩm thực mới.
- **Gia đình có trẻ nhỏ:** cần lịch trình tối ưu và tiết kiệm thời gian di chuyển.
- **Người đi công tác:** ưu tiên sự tiện lợi, khách sạn gần địa điểm họp, điểm ăn uống phù hợp.

Nhóm đối tượng này có nhu cầu lớn về một công cụ hỗ trợ thông minh, đơn giản và hiệu quả.

## 1.4 Phạm vi thực hiện

Phạm vi thực hiện của đồ án bao gồm:

### Chức năng thực hiện (In-scope):

- Hệ thống gợi ý địa điểm du lịch dựa trên sở thích.
- Gợi ý món ăn theo loại ẩm thực yêu thích.
- Gợi ý khách sạn trong bán kính dưới 10km so với địa điểm du lịch.
- Tự động tạo lịch trình du lịch cơ bản.
- Thiết kế giao diện demo mô phỏng hoạt động hệ thống.

### Ngoài phạm vi (Out-of-scope):

- Không tích hợp API đặt phòng/đặt vé thực tế.
- Không xử lý dữ liệu thời gian thực từ các nền tảng du lịch lớn.
- Không phát triển ứng dụng mobile hoàn chỉnh.
- Không hỗ trợ tính năng chỉ đường GPS theo thời gian thực.

Việc giới hạn phạm vi giúp đảm bảo đồ án tập trung vào các chức năng cốt lõi và phù hợp với thời gian triển khai.

## 1.5 Tổng quan giải pháp và công nghệ liên quan

Hiện nay, trên thị trường đã có nhiều ứng dụng du lịch thông minh như Google Travel, TripAdvisor, Klook hay Traveloka. Tuy nhiên, mỗi ứng dụng đều có những hạn chế riêng:

- **Google Travel:** mạnh về bản đồ và đánh giá, nhưng thiếu khả năng cá nhân hóa sâu.
- **TripAdvisor:** nhiều đánh giá từ cộng đồng nhưng không hỗ trợ gợi ý theo sở thích.
- **Klook:** mạnh về đặt tour nhưng không xây dựng itinerary thông minh.
- **Traveloka:** hỗ trợ đặt phòng/đặt vé nhưng không cung cấp gợi ý địa điểm/ẩm thực.

Hệ thống được đề xuất trong đồ án có sự khác biệt ở chỗ:

- Tập trung mạnh vào **cá nhân hóa trải nghiệm du lịch**.
- Kết hợp đa dạng các mô-đun: địa điểm, ẩm thực, khách sạn và itinerary.
- Dễ mở rộng và tích hợp AI trong tương lai.
- Tối ưu cho người dùng phổ thông và khách du lịch tự túc.

Do đó, Smart Tourism System có tiềm năng trở thành một nền tảng hỗ trợ du lịch thông minh linh hoạt, hiện đại và phù hợp với xu hướng du lịch cá nhân hóa.

# Chương 2

## Phân tích yêu cầu hệ thống

Chương này trình bày chi tiết các yêu cầu chức năng, yêu cầu phi chức năng và mô hình hóa tương tác giữa người dùng và hệ thống thông qua Use Case Diagram và đặc tả Use Case.

### 2.1 Yêu cầu chức năng (Functional Requirements)

Hệ thống Du lịch Thông minh cần đáp ứng các yêu cầu chức năng sau:

#### 1. Gợi ý địa điểm du lịch

- Cho phép tìm kiếm địa điểm theo từ khóa.
- Gợi ý địa điểm theo sở thích cá nhân (thiên nhiên, lịch sử, văn hoá...).
- Hiển thị thông tin chi tiết về địa điểm (mô tả, giờ mở cửa, ảnh...).

#### 2. Gợi ý món ăn theo sở thích

- Cho phép người dùng chọn khẩu vị hoặc loại món ăn yêu thích.
- Hệ thống gợi ý món ăn/nhà hàng dựa trên sở thích đã chọn.

#### 3. Gợi ý khách sạn gần địa điểm du lịch

- Cho phép người dùng xem khách sạn gần địa điểm đang xem.
- Chỉ hiển thị khách sạn trong bán kính dưới 10km.

#### 4. Tạo và quản lý lịch trình du lịch thông minh

- Hệ thống đề xuất lịch trình tự động dựa trên sở thích và thời gian.
- Cho phép người dùng chỉnh sửa lịch trình.

#### 5. Quản lý dữ liệu cho Admin

- Admin có thể thêm/chỉnh sửa/xóa địa điểm du lịch.
- Admin có thể quản lý dữ liệu khách sạn và món ăn.

## 2.2 Yêu cầu phi chức năng (Non-functional Requirements)

### 1. Hiệu năng

- Thời gian phản hồi tìm kiếm dưới 2 giây.
- Hệ thống xử lý tối thiểu 100 yêu cầu đồng thời.

### 2. Khả năng mở rộng

- Dễ tích hợp thêm API bên thứ ba (Google Maps, Booking,...).
- Dễ mở rộng dữ liệu địa điểm và khách sạn.

### 3. Tính ổn định

- Hệ thống hoạt động liên tục 24/7.
- Tỷ lệ lỗi phải dưới 1%.

## 2.3 Mô hình hóa Use Case

### 2.3.1 Sơ đồ Use Case tổng quan

Hệ thống gồm hai tác nhân chính:

- **User (khách du lịch):** sử dụng các tính năng gợi ý và lịch trình.
- **Admin:** quản lý dữ liệu địa điểm, món ăn, khách sạn.

Các nhóm Use Case chính:

- Tìm kiếm và gợi ý địa điểm.
- Gợi ý món ăn theo sở thích.
- Gợi ý khách sạn gần địa điểm.
- Tạo và quản lý lịch trình.

### **2.3.2 Đặc tả Use Case**

#### **Use Case 1: Gợi ý địa điểm theo sở thích**

**Mục đích:** Cá nhân hóa danh sách địa điểm dựa trên sở thích người dùng.

**Tác nhân:** Người dùng (User)

**Mô tả ngắn gọn:** Người dùng chọn các sở thích cá nhân (thiên nhiên, văn hoá, lịch sử, vui chơi...). Hệ thống phân tích dữ liệu và hiển thị danh sách địa điểm phù hợp nhất.

**Tiền điều kiện:**

- Người dùng đã chọn tỉnh (province) trên giao diện.
- Người dùng đã thiết lập hồ sơ sở thích hoặc chọn sở thích ở bước hiện tại.

**Hậu điều kiện:**

- Danh sách địa điểm phù hợp được hiển thị.

**Luồng sự kiện chính:**

1. User chọn bất kỳ một tỉnh trên giao diện bản đồ Việt Nam.
2. Hệ thống hiển thị danh sách các sở thích.
3. User chọn sở thích mong muốn.
4. Hệ thống phân tích dữ liệu địa điểm.
5. Hệ thống trả về danh sách địa điểm phù hợp.

**Luồng ngoại lệ:**

- User chưa chọn sở thích → Hệ thống mặc định in ra tất cả các địa điểm
- Không có địa điểm nào phù hợp → Hệ thống báo không tìm thấy

#### **Use Case 2: Gợi ý khách sạn gần địa điểm (<10km)**

**Mục đích:** Hỗ trợ tìm khách sạn phù hợp và gần địa điểm du lịch.

**Tác nhân:** Người dùng (User)

**Mô tả ngắn gọn:** Người dùng chọn một địa điểm du lịch. Hệ thống tìm khách sạn nằm trong bán kính nhỏ hơn 10km và hiển thị các khách sạn kèm thông tin mô tả.

**Tiền điều kiện:**

- Người dùng đang xem chi tiết 1 địa điểm.

**Hậu điều kiện:**

- Danh sách khách sạn gần được hiển thị.

### **Luồng chính:**

1. User chọn chức năng “Khách sạn gần đây”.
2. Hệ thống lấy vị trí địa điểm.
3. Hệ thống lọc khách sạn trong bán kính < 10km.
4. Hệ thống hiển thị danh sách khách sạn.

### **Luồng ngoại lệ:**

- Không có khách sạn trong phạm vi: hệ thống hiển thị thông báo “Không tìm thấy khách sạn gần đây”.

## **Use Case 3: Gợi ý món ăn đặc sản từng địa điểm**

**Mục đích:** Hỗ trợ tìm hiểu về món ăn đặc sản ở địa phương.

**Tác nhân:** Người dùng (User)

**Mô tả ngắn gọn:** Người dùng chọn một địa điểm du lịch. Hệ thống sẽ gợi ý sở thích của người dùng cho từng món ăn để có thể gợi ý ra các món đặc sản địa phương phù hợp với người dùng.

### **Tiền điều kiện:**

- Người dùng đang xem chi tiết 1 địa điểm.
- Người dùng chọn các sở thích

### **Hậu điều kiện:**

- Danh sách món ăn phù hợp được hiển thị.

### **Luồng chính:**

1. User chọn chức năng “Gợi ý món ăn”.
2. Hệ thống lấy vị trí địa điểm.
3. Hệ thống lọc sở thích ở các món ăn trong cơ sở dữ liệu
4. Hệ thống hiển thị danh sách các món ăn địa phương phù hợp với người dùng.

### **Luồng ngoại lệ:**

- Không có khẩu vị phù hợp, hệ thống sẽ báo về là không tìm thấy.

## **Use Case 4: Tạo lịch trình du lịch thông minh**

**Mục đích:** Tự động tạo lộ trình tối ưu cho người dùng.

**Tác nhân:** Người dùng (User)

**Mô tả ngắn gọn:** User nhập thời gian, sở thích và số địa điểm mong muốn. Hệ thống tạo một lịch trình tối ưu dựa trên gợi ý địa điểm.

**Tiền điều kiện:**

- Người dùng đã có được các địa điểm đã được gợi ý.
- Chọn các sở thích

**Hậu điều kiện:**

- Lịch trình được tạo và hiển thị cho người dùng.

**Luồng chính:**

1. User chọn “Tạo lịch trình thông minh”.
2. User chọn thời gian (số ngày), sở thích và số lượng địa điểm.
3. Hệ thống phân tích sở thích và dữ liệu vị trí.
4. Hệ thống xây dựng lịch trình.
5. Hệ thống hiển thị lịch trình.

**Luồng ngoại lệ:**

- Không đủ địa điểm để xây dựng lịch trình → hệ thống hiển thị cảnh báo.
- Phải chọn ít nhất 1 địa điểm.

# Chương 3

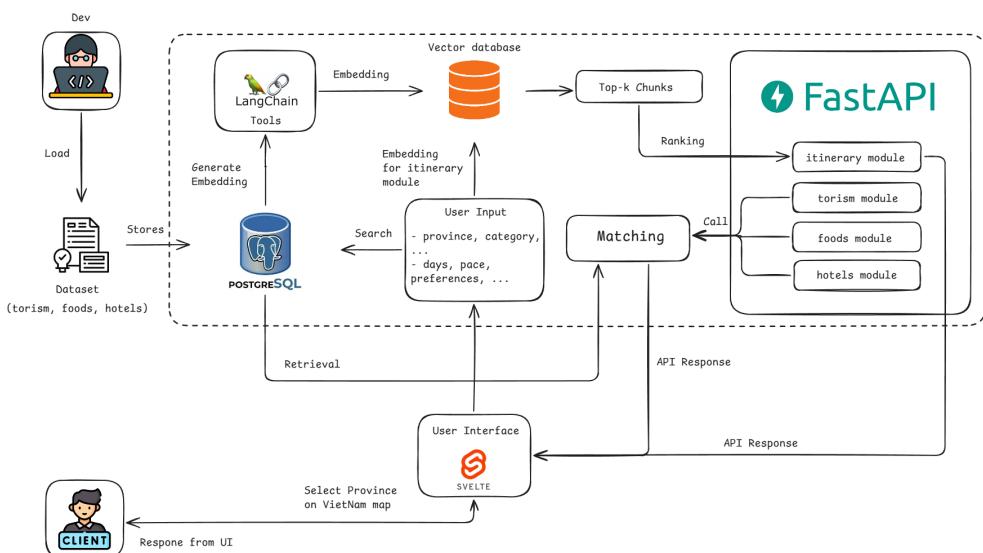
## Thiết kế hệ thống

### 3.1 Kiến trúc hệ thống tổng quan

Hệ thống được thiết kế theo mô hình **Client-Server** với kiến trúc **modular** để dễ dàng mở rộng (có thể chuyển một số module thành microservices khi cần).

Các thành phần chính:

- **Frontend (Web / Mobile)**: Giao diện người dùng, xử lý hiển thị, tương tác, gọi API tới Backend.
- **Backend / API Server**: Cung cấp REST API, xử lý xác thực, quản lý business logic, orchestration giữa các thành phần.
- **Recommendation Engine**: module chuyên trách gợi ý (kết hợp vector search / RAG và rule-based logic).
- **Database (PostgreSQL)**: Lưu trữ dữ liệu cấu trúc (Users, Destinations, Hotels, Itineraries), hỗ trợ truy vấn địa lý.



## 3.2 Thiết kế chi tiết

### 3.2.1 Sơ đồ lớp (Class Diagram)

Hệ thống bao gồm các nhóm lớp chính sau:

#### (1) Nhóm User & UI

- **User**: Lưu trữ thông tin đầu vào từ người dùng gồm: province, category, days, pace, preferences.
- **UserInterface**: Thu thập input và hiển thị kết quả.

#### (2) Nhóm API Backend (FastAPI)

- Chịu trách nhiệm cung cấp các API:
  - `getTourismRecommendations()`
  - `getFoodRecommendations()`
  - `getHotelRecommendations()`
  - `getItinerary()`
- Đóng vai trò điều phối xử lý giữa các module.

#### (3) Nhóm Recommendation Modules

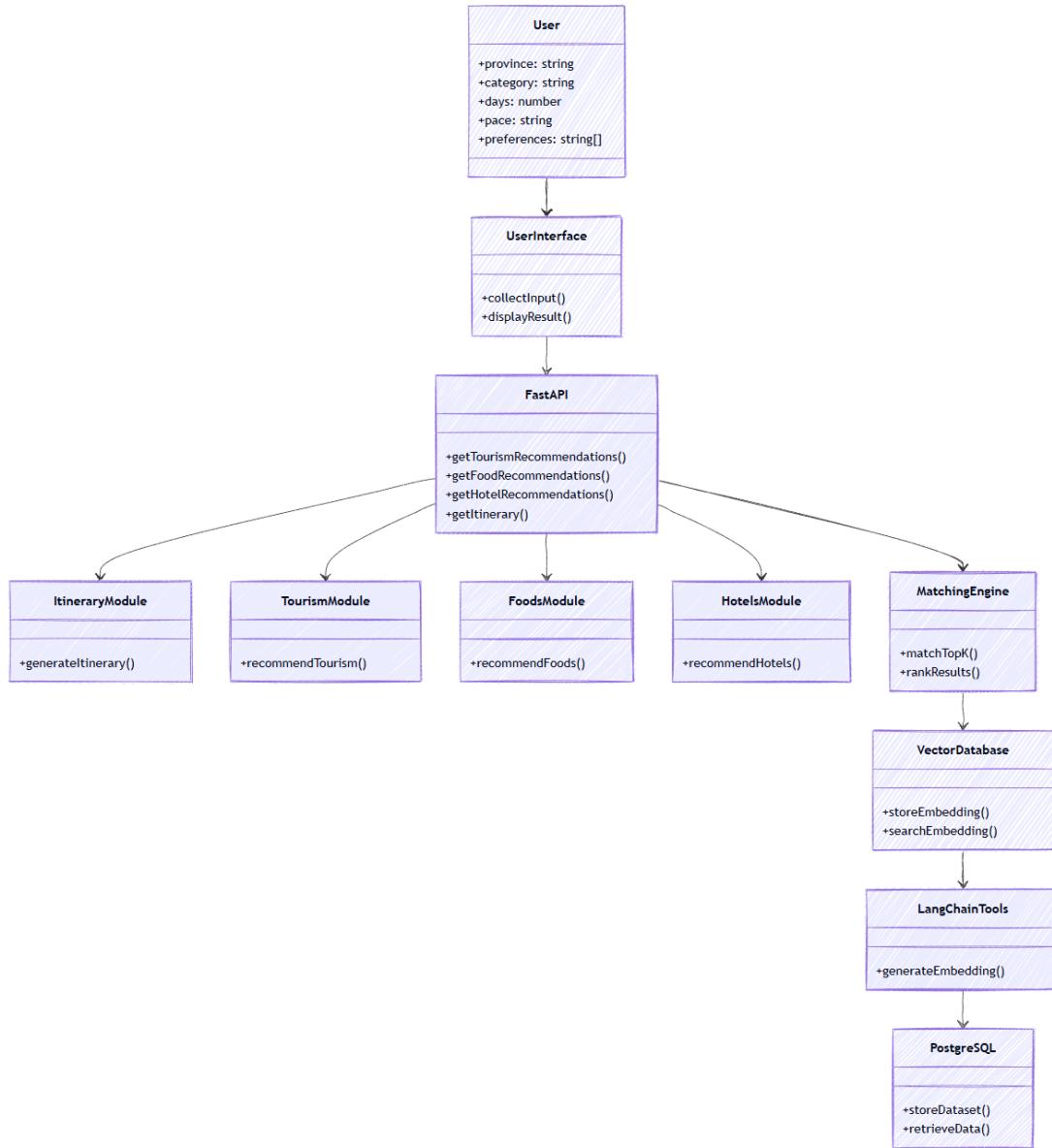
- **TourismModule**: gợi ý địa điểm du lịch.
- **FoodsModule**: gợi ý món ăn/quán ăn.
- **HotelsModule**: gợi ý khách sạn gần địa điểm (<10km).
- **ItineraryModule**: xây dựng lịch trình.

#### (4) Matching Engine

- Thực hiện `matchTopK()` và `rankResults()`.

#### (5) VectorDB & LangChain Tools

- VectorDB: lưu và tìm kiếm embedding.
- LangChainTools: tạo embedding.
- PostgreSQL: lưu dataset tourism / foods / hotels.

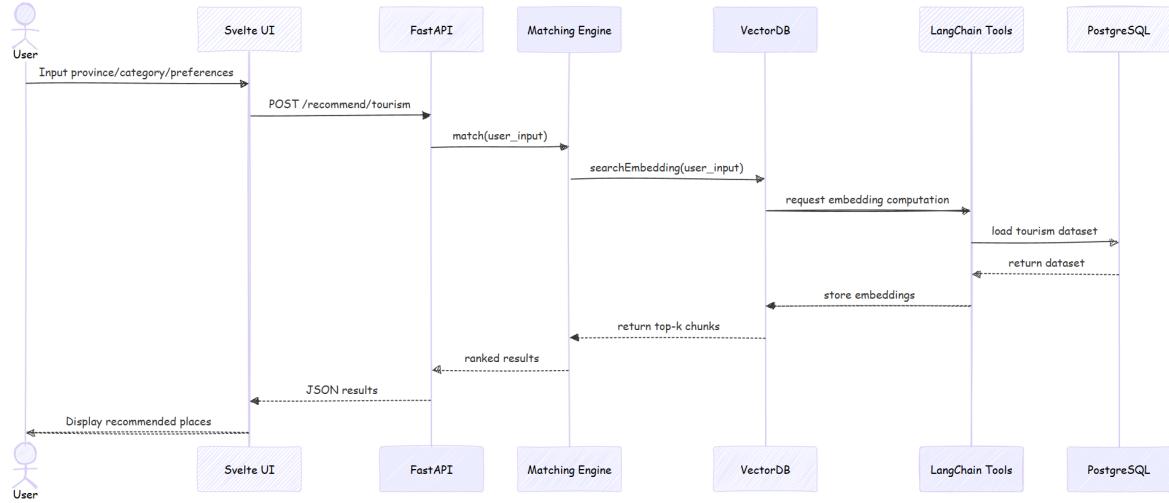


### 3.2.2 Sơ đồ tuần tự (Sequence Diagram)

Sơ đồ tuần tự dưới đây mô tả luồng xử lý của use case “**Người dùng yêu cầu gợi ý địa điểm du lịch**”:

1. Người dùng nhập dữ liệu: tỉnh, sở thích, thời gian.
2. Giao diện gửi yêu cầu đến API Server.
3. Backend gọi TourismModule để xử lý gợi ý.
4. TourismModule truy vấn MatchingEngine.
5. MatchingEngine truy vấn VectorDB/Database.
6. Kết quả được trả về Backend.

7. Backend gửi danh sách gợi ý lại cho người dùng.



### 3.2.3 3.3 Thiết kế giao diện

Giao diện người dùng (UI) của hệ thống được xây dựng theo hướng đơn giản, trực quan và dễ thao tác đối với người dùng phổ thông. Các mockup được thiết kế bằng Figma với mục tiêu giúp người dùng:

- Nhập thông tin cá nhân và nhu cầu du lịch một cách nhanh chóng.
- Quan sát danh sách gợi ý về địa điểm, khách sạn và món ăn một cách trực quan.
- Có thể xem chi tiết từng địa điểm và quản lý lịch trình của bản thân.

Dưới đây là mô tả chi tiết các giao diện chính.

#### 3.3.1 Các màn hình chính

Mục này trình bày các bản thiết kế giao diện (Figma mockups) của các màn hình then chốt của hệ thống. Hiện tại các hình ảnh chưa được bổ sung, vì vậy báo cáo sẽ mô tả bối cảnh chức năng của từng giao diện và chưa vị trí để chèn hình sau này.

**(1) Màn hình tìm kiếm (Search Screen)** Màn hình này cho phép người dùng nhập các thông tin cần thiết để hệ thống tạo gợi ý:

- Tỉnh / Thành phố du lịch.
- Thể loại (Du lịch, Ăm thực, Khách sạn, Lịch trình).
- Số ngày dự định.
- Tốc độ di chuyển (Nhanh, Trung bình, Chậm).

- Các sở thích cá nhân (biển, núi, văn hoá, ẩm thực,...).

(Hình mockup Search Screen sẽ được thêm tại đây)

**(2) Màn hình kết quả gợi ý (Recommendation Screen)** Sau khi nhập thông tin, người dùng được chuyển đến màn hình gợi ý:

- Danh sách địa điểm/khách sạn/món ăn được đề xuất.
- Mỗi mục bao gồm:
  - Tên.
  - Hình ảnh minh họa.
  - Khoảng cách (đối với khách sạn).
  - Mức phù hợp (matching score).
- Thanh lọc (filter) theo mức độ phù hợp, khoảng cách hoặc loại hình.

(Hình mockup Recommendation Screen sẽ được thêm tại đây)

**(3) Màn hình chi tiết địa điểm (Detail Screen)** Khi chọn một địa điểm trong danh sách gợi ý, người dùng được chuyển sang trang chi tiết:

- Hình ảnh địa điểm (ảnh lớn).
- Mô tả đầy đủ (từ dataset).
- Bản đồ mini (Nếu có).
- Danh sách điểm gần đó (ưu tiên khách sạn < 10km).
- Nút “Thêm vào lịch trình”.

(Hình mockup Detail Screen sẽ được thêm tại đây)

**(4) Màn hình quản lý lịch trình (Itinerary Management Screen)** Màn hình này hiển thị lịch trình được tạo tự động hoặc lịch trình người dùng đã tùy chỉnh:

- Danh sách các ngày (Day 1, Day 2, Day 3,...).
- Mỗi ngày có danh sách các hoạt động (địa điểm → thời gian → hoạt động).

(Hình mockup Itinerary Screen sẽ được thêm tại đây)

**(5) Màn hình khách sạn gần (Nearby Hotels Screen)** Màn hình này chỉ tập trung hiển thị các khách sạn gần địa điểm mà người dùng đang xem:

- Danh sách khách sạn được sắp xếp theo khoảng cách (tăng dần).
- Hiển thị tên, hình ảnh, giá tham khảo, và khoảng cách (km).
- Nút “Chọn khách sạn” hoặc “Xem chi tiết”.

*(Hình mockup Nearby Hotels Screen sẽ được thêm tại đây)*

**(6) Màn hình gợi ý món ăn (Foods Recommendation Screen)** Màn hình này hiển thị danh sách các món ăn đặc sản, phù hợp với tỉnh/thành phố mà người dùng đã lựa chọn. Đây là một trong những màn hình quan trọng giúp người dùng khám phá văn hoá ẩm thực địa phương và bổ sung các điểm ăn uống vào lịch trình.

Các thành phần chính của màn hình:

- **Danh sách món ăn gợi ý:** Hệ thống hiển thị danh sách các món ăn phù hợp nhất dựa trên:
  - Đặc sản theo tỉnh/thành phố.
  - Sở thích cá nhân của người dùng.
- **Thông tin cho mỗi món ăn:**
  - Tên món ăn.
  - Hình ảnh minh họa.
  - Mô tả ngắn về món ăn.
- **Thanh lọc (filters):**
  - Theo loại món (bún/phở/cơm/hải sản...).

*(Hình mockup Foods Recommendation Screen sẽ được thêm tại đây)*

# Chương 4

## Quá trình phát triển

### 4.1 Phân công thành viên

Trong suốt quá trình xây dựng hệ thống Smart Tourism, nhóm đã phân chia nhiệm vụ dựa trên kỹ năng và thế mạnh của từng thành viên nhằm tối ưu hiệu suất làm việc. Bảng dưới đây trình bày vai trò và các module chính mà mỗi thành viên phụ trách:

- **Nam – Backend Developer, System Architect:** Chịu trách nhiệm thiết kế kiến trúc hệ thống, xây dựng API bằng FastAPI, triển khai module Recommendation Engine và tích hợp cơ sở dữ liệu PostgreSQL.
- **Trúc, Khang, Ánh – Frontend Developer (bao gồm cả UI/UX Design):** Xây dựng giao diện người dùng bằng Svelte (framework nhóm thống nhất), thiết kế luồng UI, đảm nhiệm kết nối API, hiển thị dữ liệu cho các màn hình Tourism, Foods, Hotels và Itinerary, thiết kế mockups giao diện (Figma), chuẩn hoá bố cục các màn hình, đảm bảo tính trực quan và trải nghiệm người dùng tốt nhất.
- **Tony, Hiện – Data Engineer, Backend Developer:** Thu thập, tiền xử lý và chuẩn hóa dữ liệu: danh sách địa điểm du lịch, món ăn, khách sạn, kinh độ/vĩ độ, khoảng cách. Tối ưu câu truy vấn địa lý và hỗ trợ backend về dữ liệu, xây dựng FastAPI cho các module gợi ý món ăn, khách sạn gần nhất (<10km).
- **Huy – Tester, Backend Developer:** Chịu trách nhiệm kiểm thử từng module (unit test, system test), ghi nhận bug, kiểm thử API, xây dựng cơ chế RAG (Retrieval Augmented Generation) phục vụ cho module gợi ý lịch trình.

## 4.2 Kết quả đã đạt được

Tính đến thời điểm hiện tại, nhóm đã hoàn thiện các module cốt lõi theo đúng mục tiêu đề ra ở Chương 1. Cụ thể như sau:

- **Hoàn thành Backend API**

- API gợi ý địa điểm du lịch theo tỉnh/thành phố, sở thích, độ phổ biến.
- API gợi ý món ăn đặc sản theo tỉnh/thành và preferences.
- API liệt kê khách sạn gần địa điểm tham quan (bán kính < 10km).
- API sinh lịch trình theo số ngày và tốc độ du lịch của người dùng.

- **Hoàn thành Recommendation Engine:** Logic gợi ý rule-based hoạt động ổn định, độ chính xác cao, xử lý nhanh hơn so với hệ sử dụng vector database.

- **Hoàn thiện cấu trúc cơ sở dữ liệu PostgreSQL:** Bao gồm bảng tourism\_places, vietnam\_foods, vietnam\_hotels và các bảng quan hệ.

- **Hoàn thành Frontend cơ bản**

- Màn hình gợi ý du lịch.
- Màn hình gợi ý món ăn.
- Màn hình hiển thị khách sạn gần địa điểm.
- Màn hình lịch trình.

- **Kiểm thử mức API và mức quy trình:** Các endpoint quan trọng đã được kiểm thử bằng Postman và hoạt động đúng theo yêu cầu.

Nhìn chung, nhóm đã hoàn thành hơn 70% tổng khối lượng công việc theo kế hoạch ban đầu.

## 4.3 Kế hoạch giai đoạn tiếp theo

Vì dự án đã bước vào tuần cuối của kỳ học, giai đoạn tiếp theo sẽ tập trung chủ yếu vào việc hoàn thiện dữ liệu, tinh chỉnh mô hình gợi ý và triển khai hệ thống để chuẩn bị cho buổi báo cáo cuối kỳ. Các đầu việc chính bao gồm:

- **Mở rộng và làm sạch dữ liệu** Bổ sung thêm dữ liệu địa điểm du lịch, món ăn đặc sản, và khách sạn tại nhiều tỉnh/thành phố nhằm giúp hệ thống gợi ý đa dạng và chính xác hơn. Đồng thời chuẩn hóa dữ liệu (loại bỏ trùng lặp, thiếu trường, sai định dạng).
- **Cải thiện thuật toán gợi ý** Điều chỉnh logic gợi ý dựa trên dữ liệu bổ sung, tối ưu bộ lọc tìm kiếm theo khoảng cách, sở thích, mức độ phổ biến.

- **Hoàn thiện API và kiểm thử tính năng** Kiểm thử end-to-end tất cả các API chính: gọi ý địa điểm, gọi ý lịch trình, gọi ý món ăn và gọi ý khách sạn gần địa điểm. Sửa lỗi phát sinh trong quá trình chạy thử.
- **Triển khai hệ thống (Deploy)**
  - Dưa backend lên dịch vụ đám mây (Render, Railway hoặc AWS).
  - Triển khai frontend lên Vercel/Netlify.
  - Cấu hình tên miền, HTTPS và kiểm tra khả năng truy cập.
- **Chuẩn bị báo cáo cuối kỳ** Hoàn thiện tài liệu báo cáo, slide trình bày, và video demo để phục vụ buổi bảo vệ.

## 4.4 Khó khăn và rủi ro

Trong quá trình phát triển, nhóm đã gặp phải một số vấn đề và rủi ro tiềm ẩn, bao gồm:

- **Thiếu dữ liệu du lịch thực tế** Dữ liệu nhiều tỉnh/thành không đồng bộ, phải tiến hành thu thập thủ công và chuẩn hóa.
- **Tối ưu logic gợi ý** Thuật toán gợi ý ban đầu chưa chính xác, nhóm phải cải thiện rule-based và làm sạch dữ liệu mô tả.
- **Vấn đề tích hợp frontend và backend** Một số API trả về dữ liệu phức tạp (như lịch trình nhiều ngày) khiến UI hiển thị khó khăn, cần thời gian điều chỉnh.
- **Rủi ro mở rộng hệ thống** Nếu về sau bổ sung vector database hoặc hệ thống microservices, chi phí triển khai và bảo trì sẽ cao hơn.

Nhóm đã và đang tiếp tục khắc phục từng vấn đề, đảm bảo hệ thống đạt chất lượng tốt nhất khi bàn giao.

# Tài liệu tham khảo

- [1] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, “*Retrieval-Augmented Generation for Large Language Models: A Survey*,” Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University; School of Computer Science, Fudan University; College of Design and Innovation, Tongji University.
- [2] Keivalya Pandya, Prof. Dr. Mehfuzা Holia, “*Automating Customer Service using LangChain, Building custom open-source GPT Chatbot for organizations*”, Birla Vishvakarma Mahavidyalaya, Gujarat, India.
- [3] N. Toyaad Kumar Reddy, Manas Ranjan Patra, Brojo Kishore Mishra, “*Design and Implementation of an AI-Based Chatbot Framework with Retrieval Augmented Generation and Integrated Recommender System for Interactive User Support*”, Dept. of Computer Science and Engineering NIST University, Institute Park, Berhampur.
- [4] Verena Traubinger, Martin Gaedke, “*Interaction Design Patterns of Web Chatbots*”, Faculty of Computer Science, Chemnitz University of Technology, Germany.