# TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

## KHOA CÔNG NGHỆ THÔNG TIN

-----------🙞🙜🕮🙞🙜-----------

****

**ĐỒ ÁN KIẾN TRÚC THIẾT KẾ PHẦN MỀM**

**ĐỀ TÀI**

**Trang web hỗ trợ lên kế hoạch du lịch**

Giảng viên hướng dẫn: Phạm Hữu Tùng

**NHÓM THỰC HIỆN: 6**

Lớp quản lý: 66PM5

Sinh viên thực hiện:

1. Nguyễn Anh Tú - 0207566

2. An Đăng Vinh - 0209266

***Hà Nội, 05/2025***

Mục lục

**Chương 1: Giới thiệu đề tài**

**1.1. Lý do chọn đề tài**

Trong thời đại số hóa hiện nay, nhu cầu du lịch ngày càng tăng cao, đặc biệt là đối với giới trẻ và những người yêu thích khám phá. Tuy nhiên, việc lên kế hoạch cho một chuyến đi thường tiêu tốn nhiều thời gian, đòi hỏi phải tìm kiếm thông tin từ nhiều nguồn khác nhau như nơi ở, phương tiện, hoạt động vui chơi, địa điểm ăn uống, và chi phí. Điều này có thể gây khó khăn cho người dùng, đặc biệt là những người chưa có kinh nghiệm.

Từ thực tiễn đó, việc xây dựng một **trang web hỗ trợ lập kế hoạch du lịch** giống như **stippl.io** – một nền tảng cho phép người dùng dễ dàng tạo, và quản lý kế hoạch chuyến đi – là điều cần thiết. Đề tài này vừa mang tính thực tiễn cao, vừa giúp người thực hiện củng cố và phát triển kỹ năng lập trình web, phân tích yêu cầu và thiết kế hệ thống.

**1.2. Mục tiêu của đồ án**

Mục tiêu chính của đồ án là xây dựng một trang web hỗ trợ người dùng trong việc lập kế hoạch du lịch một cách tiện lợi, trực quan và cộng tác hiệu quả. Cụ thể, trang web cần đáp ứng các chức năng sau:

* Cho phép người dùng đăng ký, đăng nhập, đăng xuất tài khoản.
* Cho phép tạo, chỉnh sửa và xem các kế hoạch chuyến đi, bao gồm thông tin về nơi ở, phương tiện, hoạt động, chi phí ước tính và danh sách vật dụng.
* Hỗ trợ người dùng xem gợi ý hoạt động, đính kèm đường dẫn tham khảo, hiển thị bản đồ trực quan hành trình.
* Quản lý quyền riêng tư, chia sẻ kế hoạch với người khác, xem các kế hoạch công khai của người dùng khác.
* Hỗ trợ tìm kiếm, theo dõi trang cá nhân người dùng khác và nhận thông báo khi có cập nhật mới.

Thông qua việc xây dựng hệ thống này, sinh viên có thể áp dụng kiến thức về thiết kế giao diện người dùng (UI/UX), phát triển frontend với React, backend với Node.js/Express, quản lý cơ sở dữ liệu với PostgreSQL, và vận dụng các kỹ thuật lập trình hướng sự kiện, bảo mật, phân quyền và tối ưu hiệu năng.

**1.3. Phạm vi và giới hạn**

**Phạm vi:**

* Phát triển ứng dụng web với các chức năng chính như đã nêu ở mục tiêu.
* Người dùng có thể tạo và quản lý nhiều chuyến đi.
* Hệ thống cho phép thao tác CRUD (Create, Read, Update, Delete) cho các thành phần trong chuyến đi: nơi ở, phương tiện, hoạt động, danh sách vật dụng,...
* Tích hợp bản đồ để hiển thị địa điểm du lịch.
* Cho phép chia sẻ kế hoạch công khai và theo dõi người dùng khác.

**Giới hạn:**

* Trang web chỉ hỗ trợ người dùng thông thường, chưa có hệ thống quản trị viên (admin).
* Không tích hợp các dịch vụ thanh toán thực tế (chỉ tính chi phí ước lượng).
* Gợi ý địa điểm và hoạt động dựa trên dữ liệu mẫu, chưa áp dụng AI.
* Ứng dụng ưu tiên hoạt động trên nền web, chưa phát triển phiên bản mobile app.

**1.4. Phương pháp thực hiện**

Đề tài được thực hiện dựa trên quy trình phát triển phần mềm Agile với các bước cụ thể sau:

1. **Phân tích yêu cầu người dùng**: Dựa trên các User Story đã xây dựng để xác định đầy đủ các chức năng cần có.
2. **Thiết kế hệ thống**:
   * Thiết kế kiến trúc tổng thể (frontend-backend-database).
   * Thiết kế giao diện người dùng (UI/UX).
   * Thiết kế cơ sở dữ liệu (ERD, các bảng như User, Trip, Activity,...).
3. **Phát triển ứng dụng**:
   * Sử dụng React và Tailwind CSS cho giao diện người dùng.
   * Sử dụng Node.js, Express cho backend.
   * Dùng PostgreSQL kết hợp Prisma ORM để quản lý cơ sở dữ liệu.
   * Tích hợp JWT cho xác thực và phân quyền người dùng.

# **Chương 2: Phân tích hệ thống**

## **2.1. Mô tả bài toán**

Hiện nay, người dùng thường gặp khó khăn trong việc lập kế hoạch du lịch do phải tìm kiếm thông tin từ nhiều nguồn khác nhau, gây mất thời gian và thiếu tính hệ thống. Ngoài ra, việc ghi chú, ước lượng chi phí, ghi nhớ hành trình, danh sách vật dụng cần mang theo cũng là thách thức nếu không có công cụ hỗ trợ rõ ràng.

Trang web được xây dựng nhằm giải quyết các vấn đề sau:

* Không có nền tảng trung tâm để tổ chức toàn bộ thông tin của một chuyến đi.
* Người dùng thiếu công cụ trực quan hóa lịch trình và hành trình.
* Việc chia sẻ kế hoạch với bạn bè hoặc tham khảo kế hoạch công khai của người khác còn hạn chế.
* Không có cách thức quản lý riêng tư, cập nhật hoạt động và thông báo hiệu quả.

Mục tiêu là xây dựng một nền tảng trực tuyến hỗ trợ người dùng tạo, chỉnh sửa, xem và chia sẻ kế hoạch du lịch, đồng thời cung cấp các công cụ gợi ý, tính toán chi phí, hiển thị bản đồ, quản lý quyền truy cập và cá nhân hóa trải nghiệm người dùng.

**2.2. Yêu cầu chức năng**

Dựa trên các User Stories, hệ thống cần đảm bảo các chức năng chính sau:

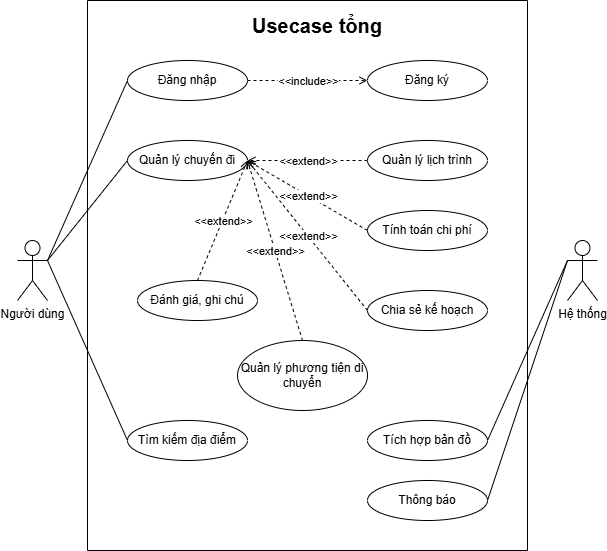
| **STT** | **Chức năng** | **Mô tả** |
| --- | --- | --- |
| 1 | **Đăng ký tài khoản (US001)** | Người dùng có thể tạo tài khoản mới để sử dụng hệ thống. |
| 2 | **Đăng nhập/đăng xuất (US002 - US003)** | Cho phép người dùng truy cập hoặc thoát khỏi hệ thống. |
| 3 | **Thanh điều hướng (US004)** | Hiển thị menu điều hướng giúp người dùng dễ dàng truy cập các tính năng. |
| 4 | **Tạo và quản lý chuyến đi (US005)** | Tạo kế hoạch du lịch với nhiều mục thành phần. |
| 5 | **Thêm/cập nhật nơi ở, phương tiện, hoạt động (US006 - US008)** | Quản lý các yếu tố quan trọng của chuyến đi. |
| 6 | **Tính toán chi phí ước tính (US009)** | Tự động tổng hợp chi phí từ các mục đã nhập. |
| 7 | **Gợi ý địa điểm/hoạt động (US010)** | Đưa ra gợi ý giúp người dùng lên kế hoạch nhanh hơn. |
| 8 | **Danh sách vật dụng mang theo (US011)** | Quản lý các vật dụng cần chuẩn bị trước chuyến đi. |
| 9 | **Thiết lập quyền riêng tư chuyến đi (US012)** | Cho phép giới hạn người xem kế hoạch. |
| 10 | **Xem kế hoạch công khai người khác (US013)** | Truy cập các kế hoạch public để tham khảo. |
| 11 | **Đính kèm đường link tham khảo (US014)** | Cho phép người dùng lưu các link hữu ích liên quan đến chuyến đi. |
| 12 | **Bản đồ trực quan (US015)** | Hiển thị các địa điểm trong kế hoạch qua bản đồ tương tác. |
| 13 | **Trang cá nhân (US016)** | Người dùng có thể xem và cập nhật thông tin cá nhân. |
| 14 | **Tìm kiếm và theo dõi người dùng khác (US017)** | Tìm kiếm người dùng khác, theo dõi và nhận thông báo khi họ cập nhật kế hoạch mới. |

**2.3. Yêu cầu phi chức năng**

Bên cạnh các yêu cầu chức năng, hệ thống cũng cần đáp ứng một số yêu cầu phi chức năng để đảm bảo hiệu quả sử dụng và độ tin cậy:

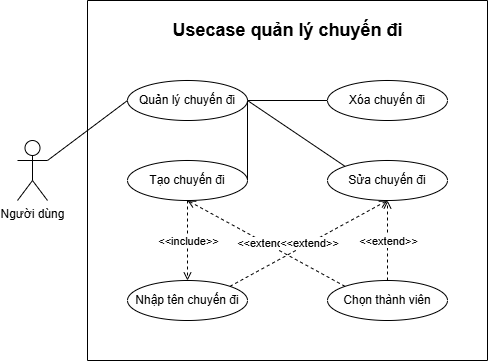
| **Loại yêu cầu** | **Mô tả** |
| --- | --- |
| **Hiệu năng** | Hệ thống phải đảm bảo phản hồi nhanh với thời gian tải trang dưới 3 giây cho các thao tác chính. |
| **Khả năng mở rộng** | Thiết kế kiến trúc có khả năng mở rộng thêm nhiều tính năng mới trong tương lai (ví dụ: đánh giá địa điểm, lên kế hoạch nhóm,...). |
| **Bảo mật** | Thông tin tài khoản và kế hoạch du lịch phải được bảo vệ bằng xác thực JWT, mã hóa mật khẩu, phân quyền truy cập. |
| **Tính khả dụng** | Hệ thống cần hoạt động ổn định, tối ưu trải nghiệm người dùng (UX/UI) và hỗ trợ trên nhiều thiết bị (desktop, tablet, mobile). |
| **Tính thân thiện** | Giao diện trực quan, dễ sử dụng, hướng tới trải nghiệm người dùng tốt. |
| **Khả năng bảo trì** | Mã nguồn được tổ chức rõ ràng, có chú thích đầy đủ để dễ dàng nâng cấp và sửa lỗi. |
| **Khả năng tương tác** | Giao diện cần tương tác mượt mà, tích hợp drag & drop khi sắp xếp lịch trình hoặc địa điểm nếu có thể. |

**2.4. Use Case Diagram tổng quát**

****

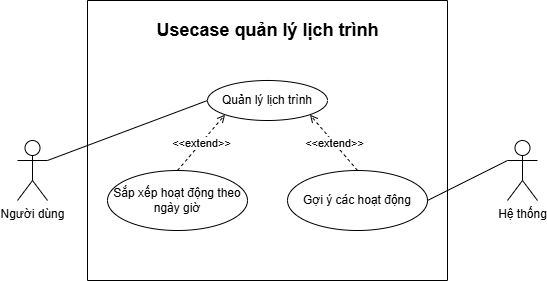
**2.5. Mô tả các Use Case chính:**

Usecase quản lý chuyến đi



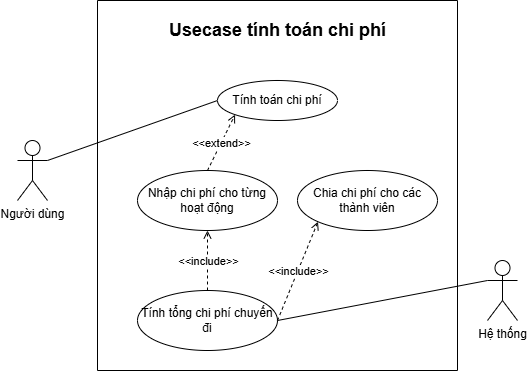
|  |  |
| --- | --- |
| **Name** | Quản lý chuyến đi |
| **Brief description** | Tạo, chỉnh sửa, xoá kế hoạch chuyến đi.  Thêm tên chuyến đi, định dạng ngày tháng, địa điểm.  Chọn danh sách thành viên tham gia chuyến đi. |
| **Actors** | Người dùng |
| **Pre-conditions** | * Người dùng phải đăng nhập để có thể sử dụng chức năng này |
| **Basic flow** | 1. Người dùng bấm tạo chuyến đi 2. Người dùng nhập tên chuyến đi 3. Người dùng chọn thành viên tham gia chuyến đi 4. Người dùng có thể sửa tên hoặc thành viên của chuyến đi 5. Người dùng có thể xóa chuyến đi |
| **Alternative flow** | * Nếu người dùng chưa đăng nhập thì sẽ không thể tạo chuyến đi * Nếu người dùng không nhập tên chuyến đi sẽ báo lỗi |
| **Result** | * Người dùng tạo được chuyến đi |
| **Exceptions** | Nếu hệ thống gặp lỗi: Hệ thống hiển thị thông báo lỗi. |

1. Usecase quản lý lịch trình



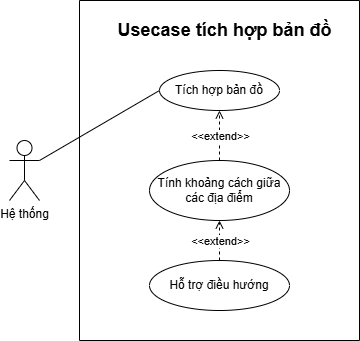
|  |  |
| --- | --- |
| **Name** | Quản lý lịch trình |
| **Brief description** | Quản lý lịch trình với các hoạt động ứng với thời gian cụ thể |
| **Actors** | Người dùng |
| **Pre-conditions** | * Người dùng cần đăng nhập để sử dụng các tính |
| **Basic flow** | 1. Người dùng chọn khung giờ 2. Người dùng chọn hoạt động ứng với khung giời chọn trước đó 3. Hệ thống sẽ gợi ý các hoạt động mà nhiều người hoạt động vào lúc đó |
| **Alternative flow** | * Nếu người dùng nhập khung giờ không hợp lệ thì hệ thống báo lỗi. |
| **Result** | * Người dùng có được thời gian biểu cụ thể cho từng hoạt động * Cập nhật lịch trình nhanh chóng. |
| **Exceptions** | Nếu hệ thống gặp lỗi: Hệ thống hiển thị thông báo lỗi. |

Usecase tính toán chi phí



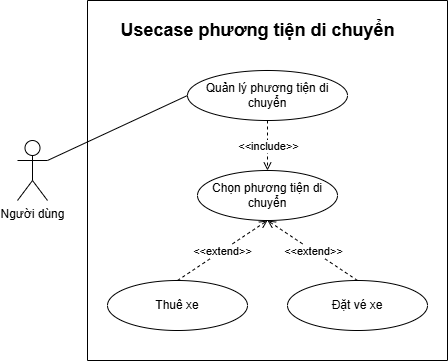
|  |  |
| --- | --- |
| **Name** | Tính toán chi phí |
| **Brief description** | Cung cấp các chức năng tính chi phí chuyến đi |
| **Actors** | Người dùng |
| **Pre-conditions** | Người dùng cần đăng nhập để sử dụng các tính năng |
| **Basic flow** | 1. Người dùng nhập chi phí cho từng hoạt động tương ứng 2. Hệ thống sẽ  tự động tính tổng chi phí 3. Hệ thống sẽ tự động chia tiền cho các thành viên trong chuyến đi |
| **Alternative flow** |  |
| **Result** | 1. Người dùng có thể biết mọi chi phí với hoạt động tương ứng. 2. Người dùng biết số tiền của từng thành viên cần đóng góp |
| **Exceptions** | Nếu hệ thống gặp lỗi: Hệ thống hiển thị thông báo lỗi. |

Usecase tích hợp bản đồ



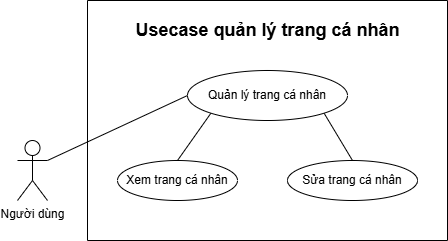
|  |  |
| --- | --- |
| **Name** | Tích hợp bản đồ |
| **Brief description** | Giúp người dùng xem được quãng đường, khoảng cách giữa các điểm đến |
| **Actors** | Hệ thống |
| **Pre-conditions** | * Người dùng cần đăng nhập để sử dụng các tính năng này. |
| **Basic flow** | 1. Người dùng chọn điểm bắt đầu và điểm đến 2. Hệ thống hiển thị đường đi 3. Hệ thống tính toán khoảng cách 4. Hệ thống điều hướng dựa trên vị trí thực tế |
| **Alternative flow** |  |
| **Result** | Người dùng biết quãng đường và khoảng cách của chuyến đi |
| **Exceptions** | Nếu hệ thống gặp lỗi: Hệ thống hiển thị thông báo lỗi. |

Usecase quản lý phương tiện di chuyển



|  |  |
| --- | --- |
| **Name** | Quản lý phương tiện di chuyển |
| **Brief description** | Quản lý phương tiện di chuyển ứng với mỗi chặng đường |
| **Actors** | Người dùng |
| **Pre-conditions** | * Người dùng cần đăng nhập để sử dụng các tính năng khác. |
| **Basic flow** | 1. Người dùng chọn phương tiện di chuyển 2. Người dùng chọn địa điểm di chuyển với phương tiện tương ứng vừa chọn 3. Người dùng có thể thuê xe nếu chưa có xe 4. Người dùng có thể đặt vé xe nếu cần |
| **Alternative flow** |  |
| **Result** | Người dùng có thể biết được phương tiện cần dùng để di chuyển trong mỗi chặng đường |
| **Exceptions** | Nếu hệ thống gặp lỗi: Hệ thống hiển thị thông báo lỗi. |

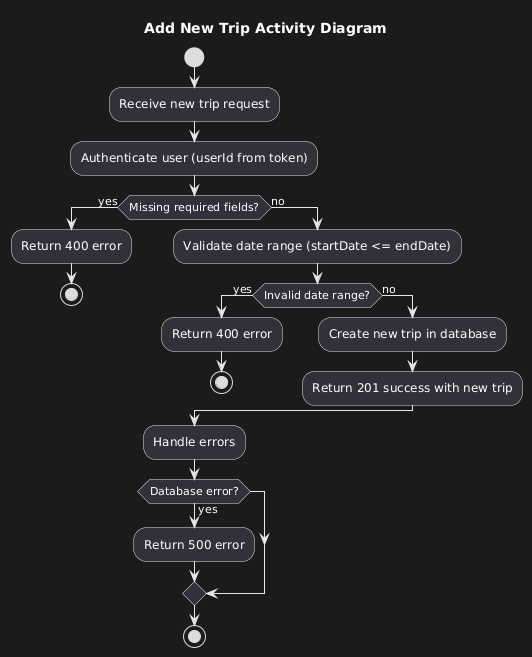
Usecase quản lý trang cá nhân



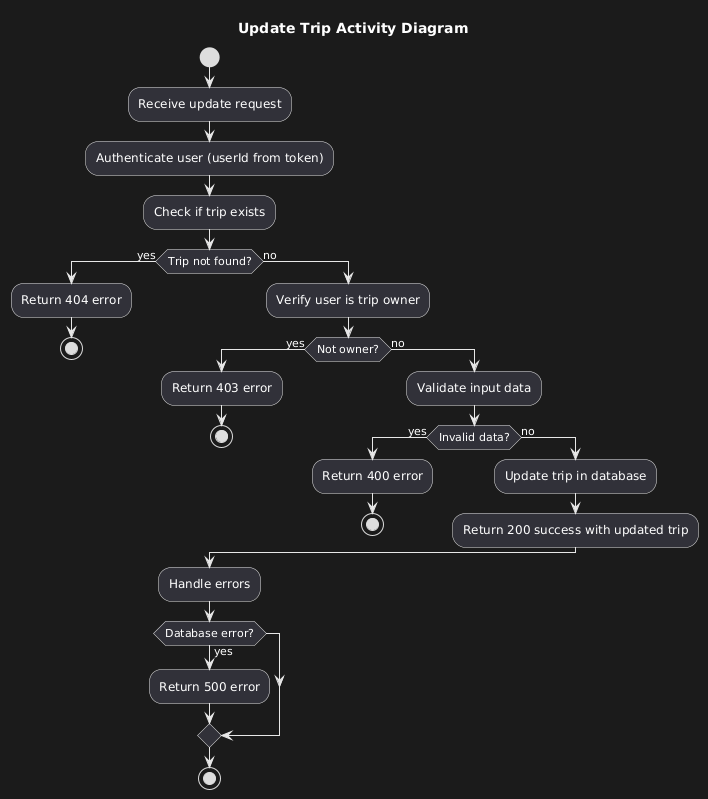
|  |  |
| --- | --- |
| **Name** | Quản lý trang cá nhân |
| **Brief description** | Người dùng có thể chọn xem trang cá nhân của mình hoặc người dùng khác.  Người dùng có thể chỉnh sửa trang cá nhân của mình |
| **Actors** | Người dùng |
| **Pre-conditions** | * Người dùng có thể đăng ký để tạo tài khoản. * Người dùng cần đăng nhập để sử dụng các tính năng khác. |
| **Basic flow** | 1. Người dùng truy cập hệ thống. 2. Người dùng có thể đăng ký tài khoản 3. Nếu đã có tài khoản, người dùng có thể đăng nhập 4. Khi đăng nhập thành công, người dùng có thể:  * Xem thông tin cá nhân * Xem thông tin của người dùng  1. Người dùng có thể đăng xuất |
| **Alternative flow** | * Nếu người dùng nhập sai thông tin khi đăng nhập, hệ thống yêu cầu nhập lại. * Nếu người dùng quên mật khẩu, hệ thống cung cấp chức năng đặt lại mật khẩu. |
| **Result** | Người dùng có thể quản lý tài khoản của mình và truy cập thông tin cần thiết. |
| **Exceptions** | Nếu hệ thống gặp lỗi: Hệ thống hiển thị thông báo lỗi.  Lỗi kết nối khiến người dùng không thể đăng nhập hoặc đăng xuất.   Lỗi cơ sở dữ liệu khiến thông tin cá nhân không hiển thị đúng. |

**2.6 Biểu đồ hoạt động (Các chức năng chính)**

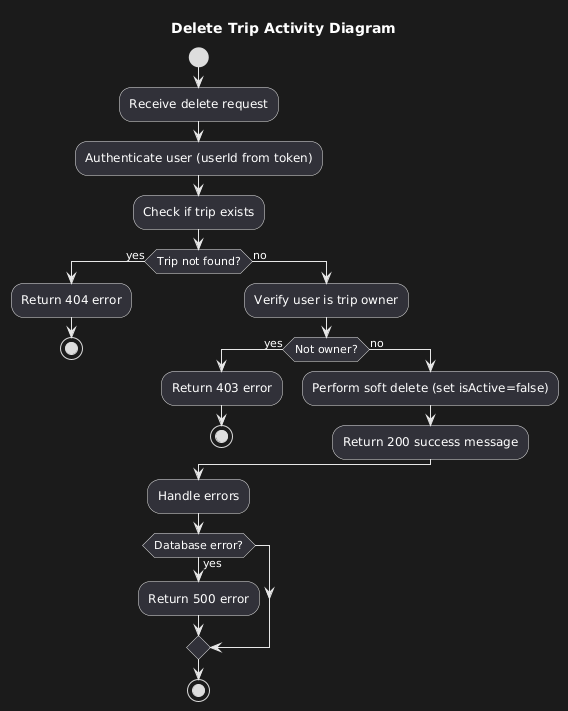
Add trip



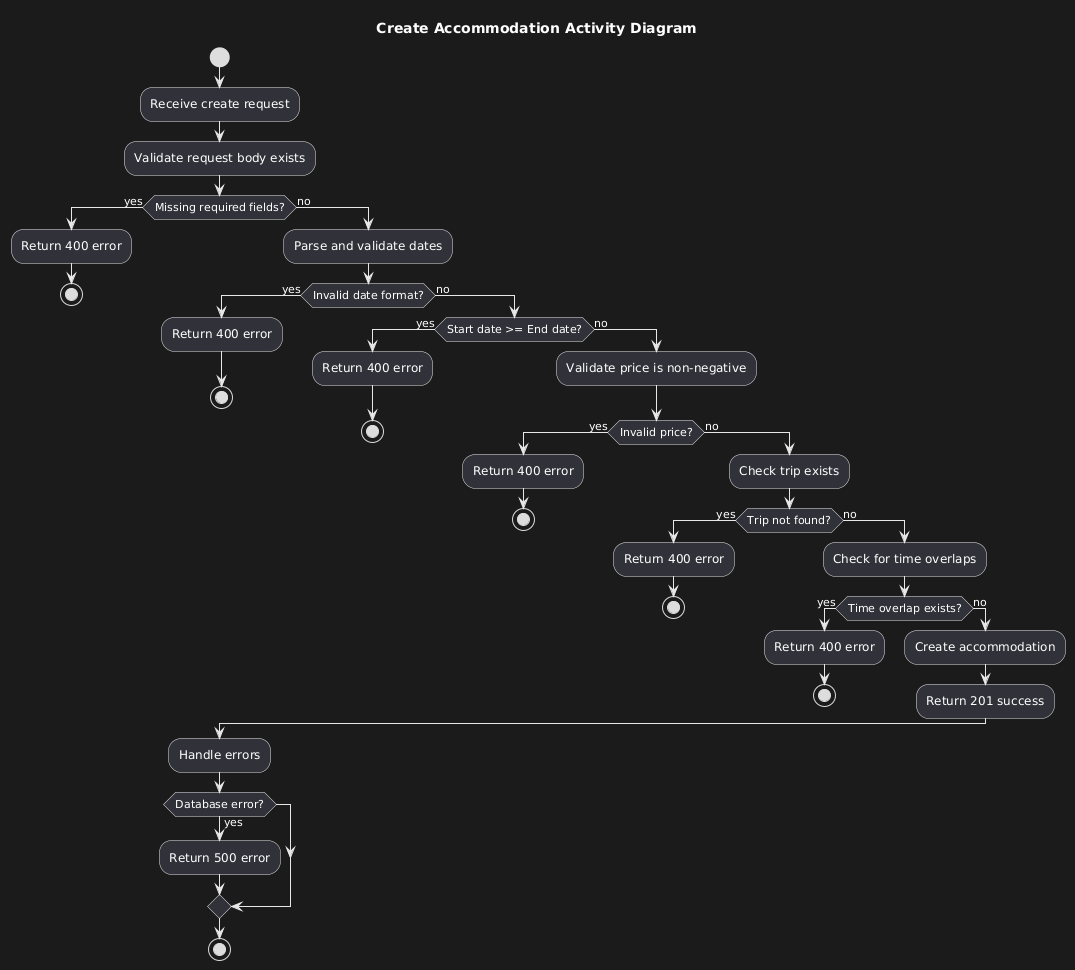
Update trip



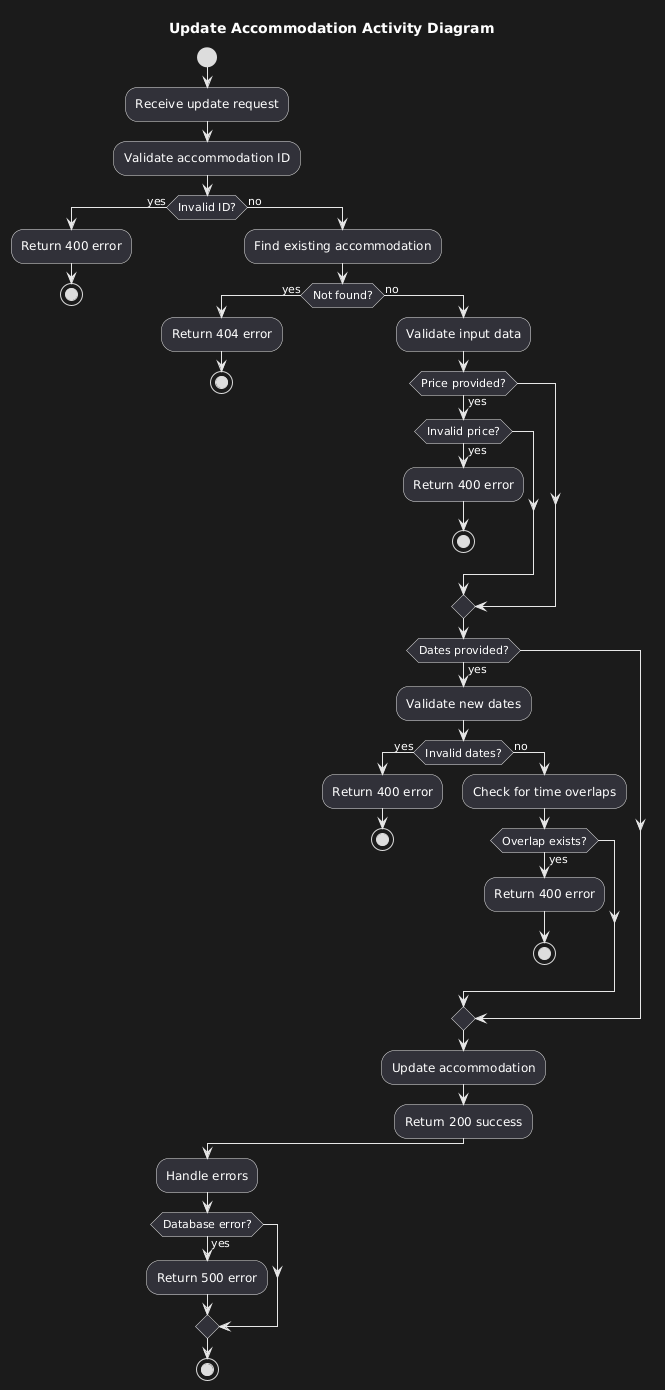
Delete trip



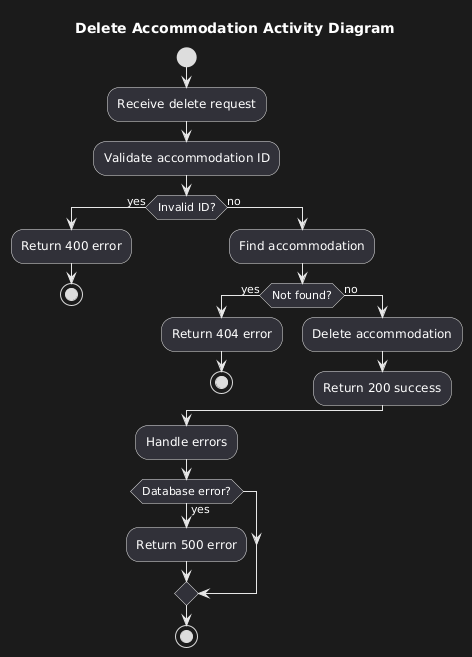
Add accommodation



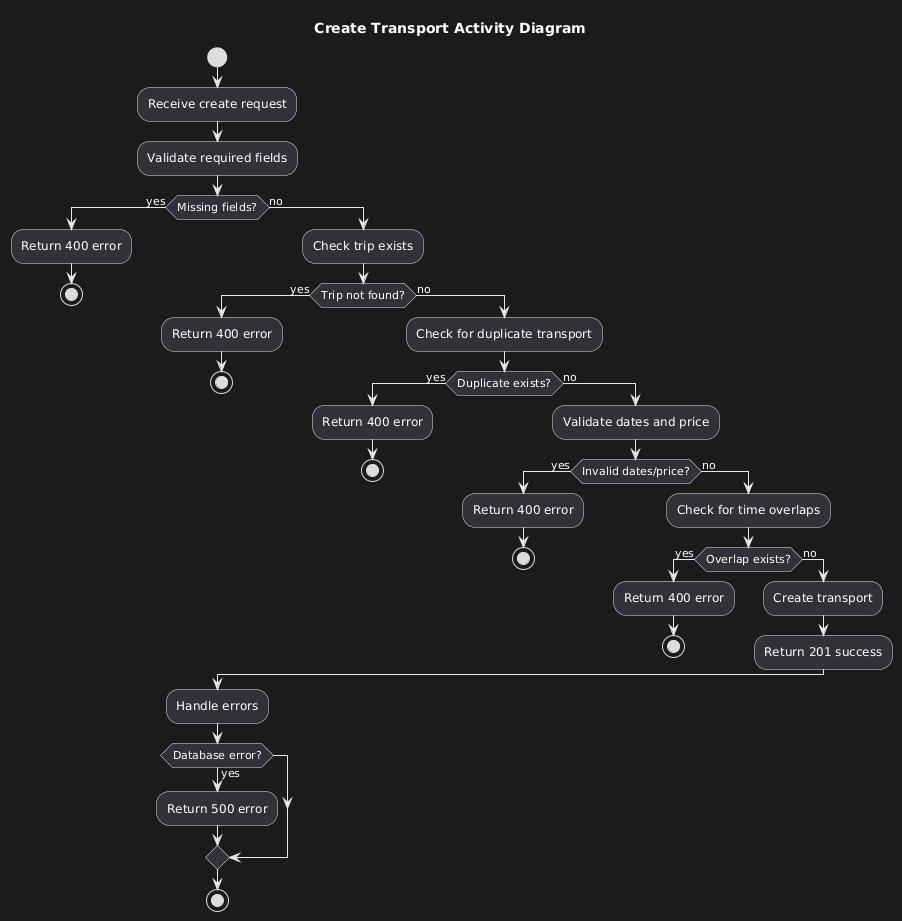
Update accommodation



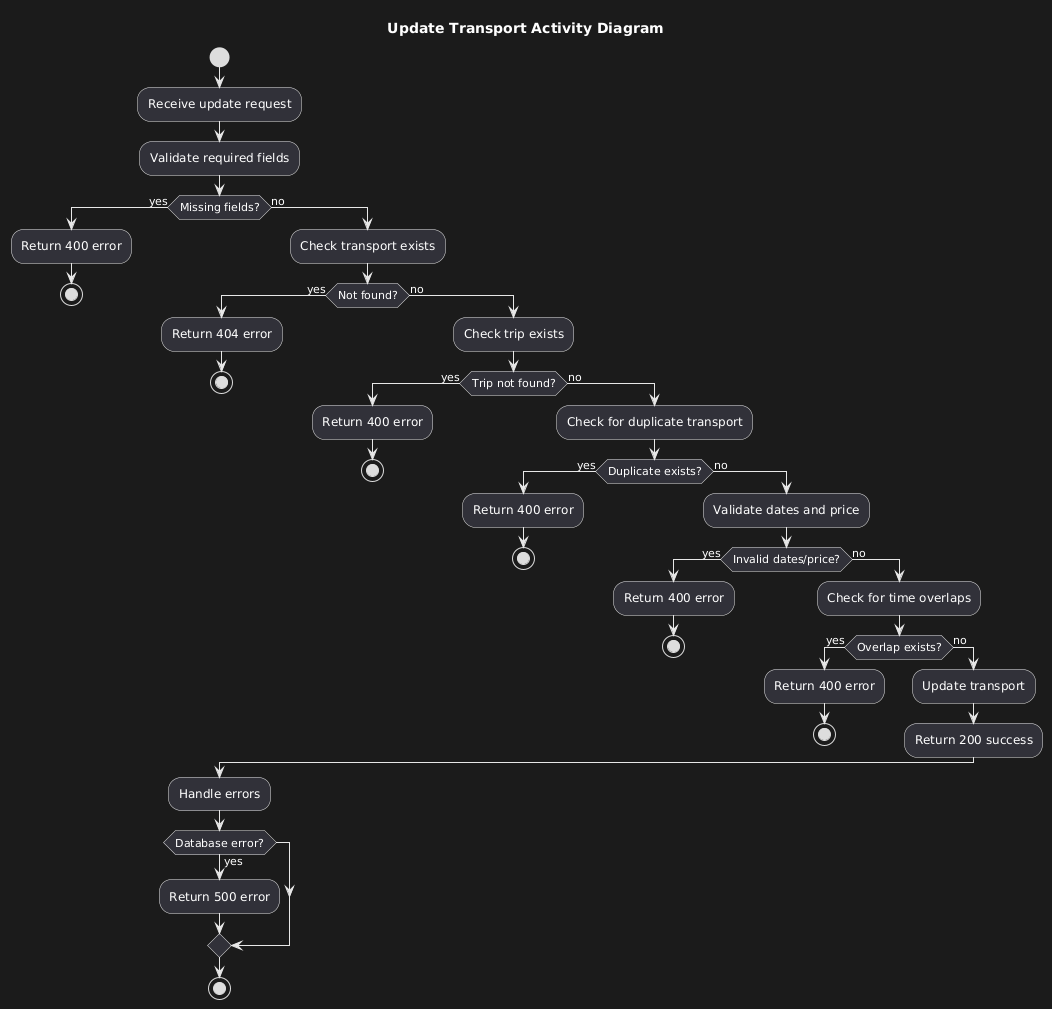
Delete accommodation



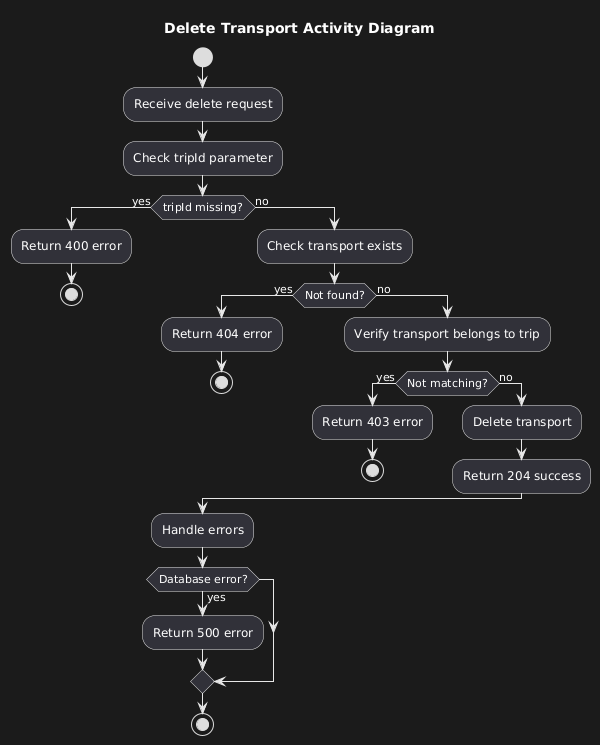
Add transport



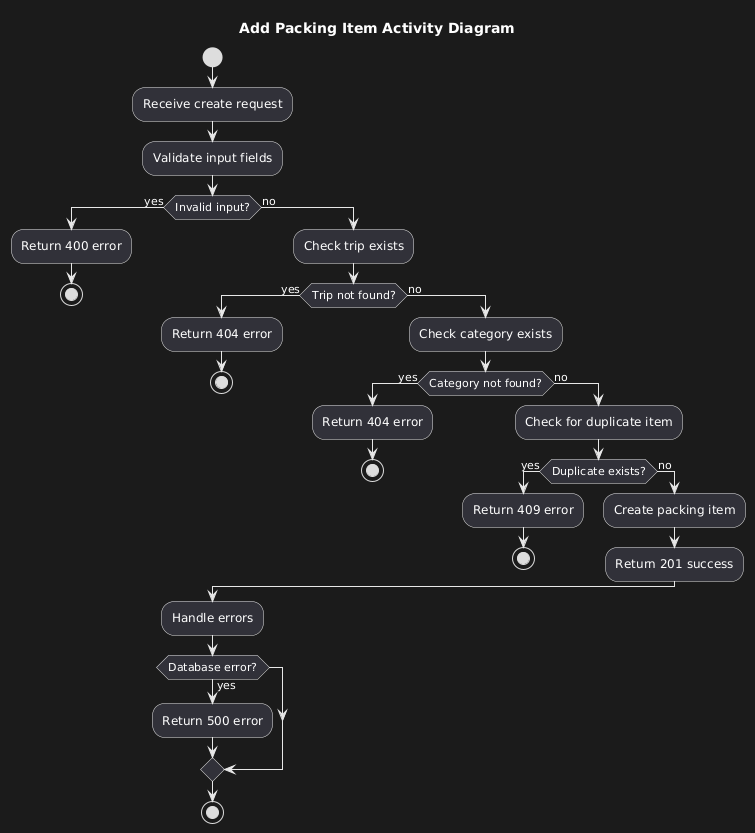
Update transport



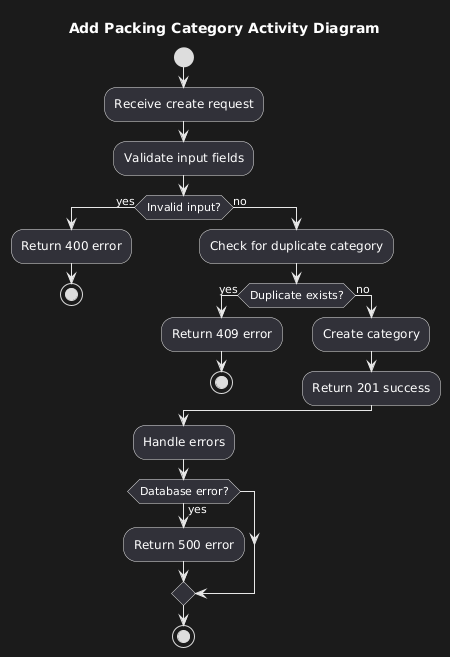
Delete Transport



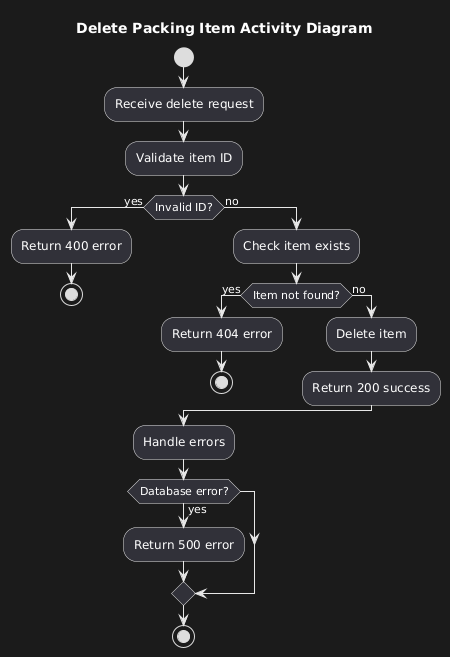
Add packing item



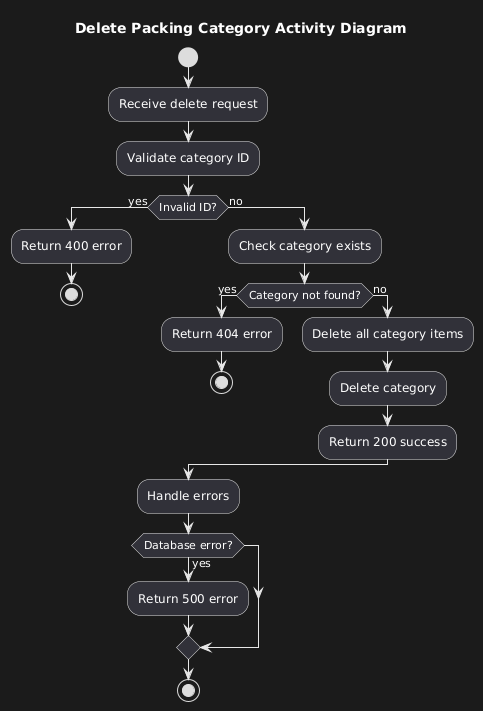
Add packing category



Delete packing item



Delete packing category

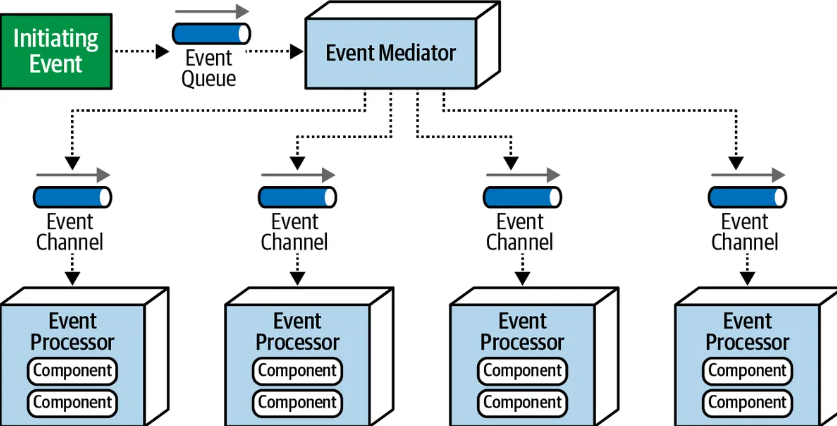


**Chương 3. Thiết kế hệ thống**

**3.1. Lựa chọn mô hình kiến trúc (Monolithic, Microservices, MVC, Layered, Clean Architecture, ...)**

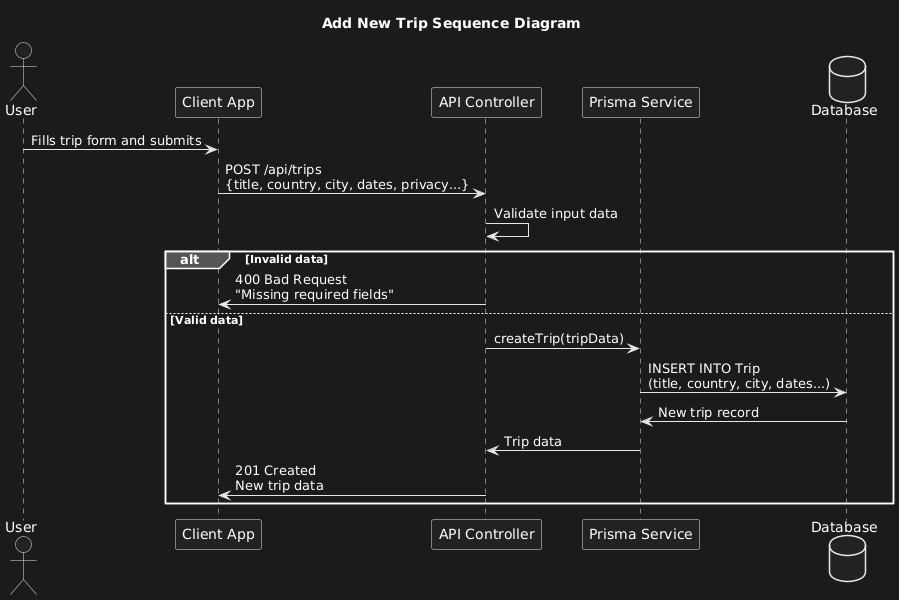
- Chọn Event Driven Architecture – kiến trúc hướng sự kiện

**3.2. Sơ đồ kiến trúc tổng thể**

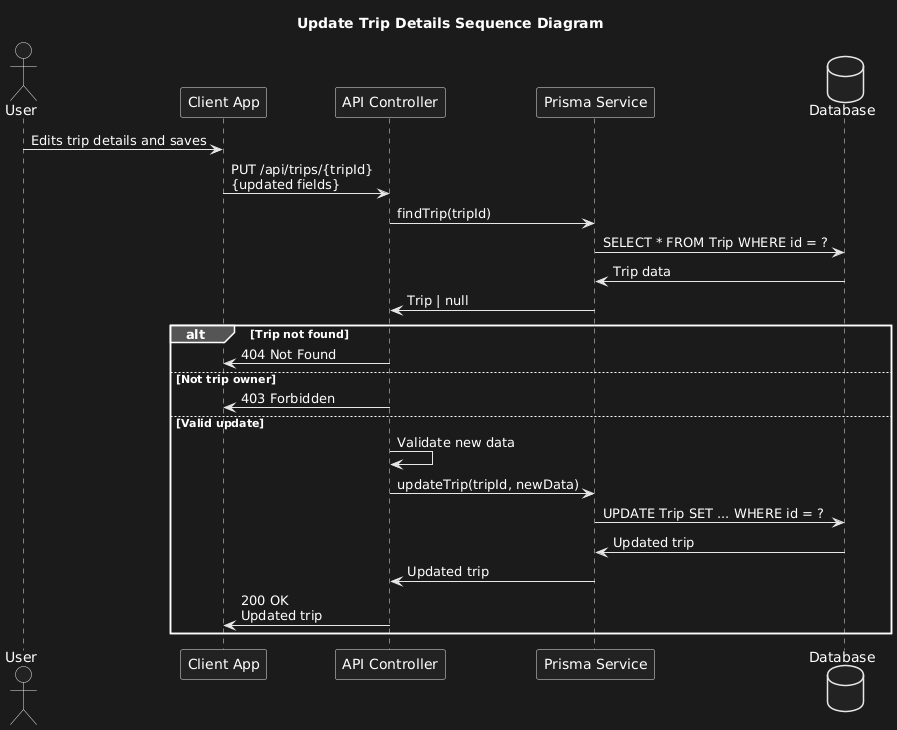


**3.3. Biểu đồ tuần tự (các chức năng chính)**

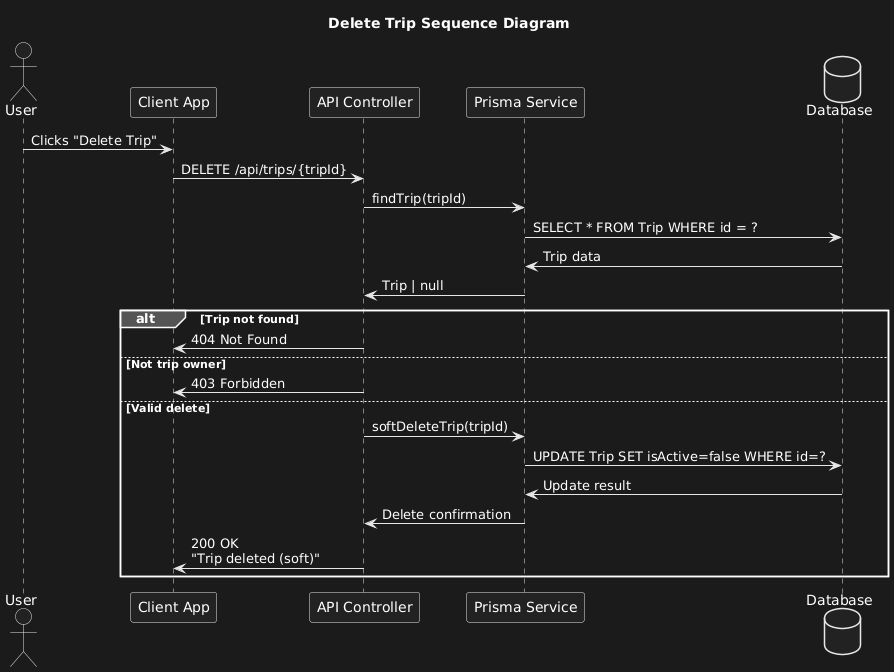
Add trip



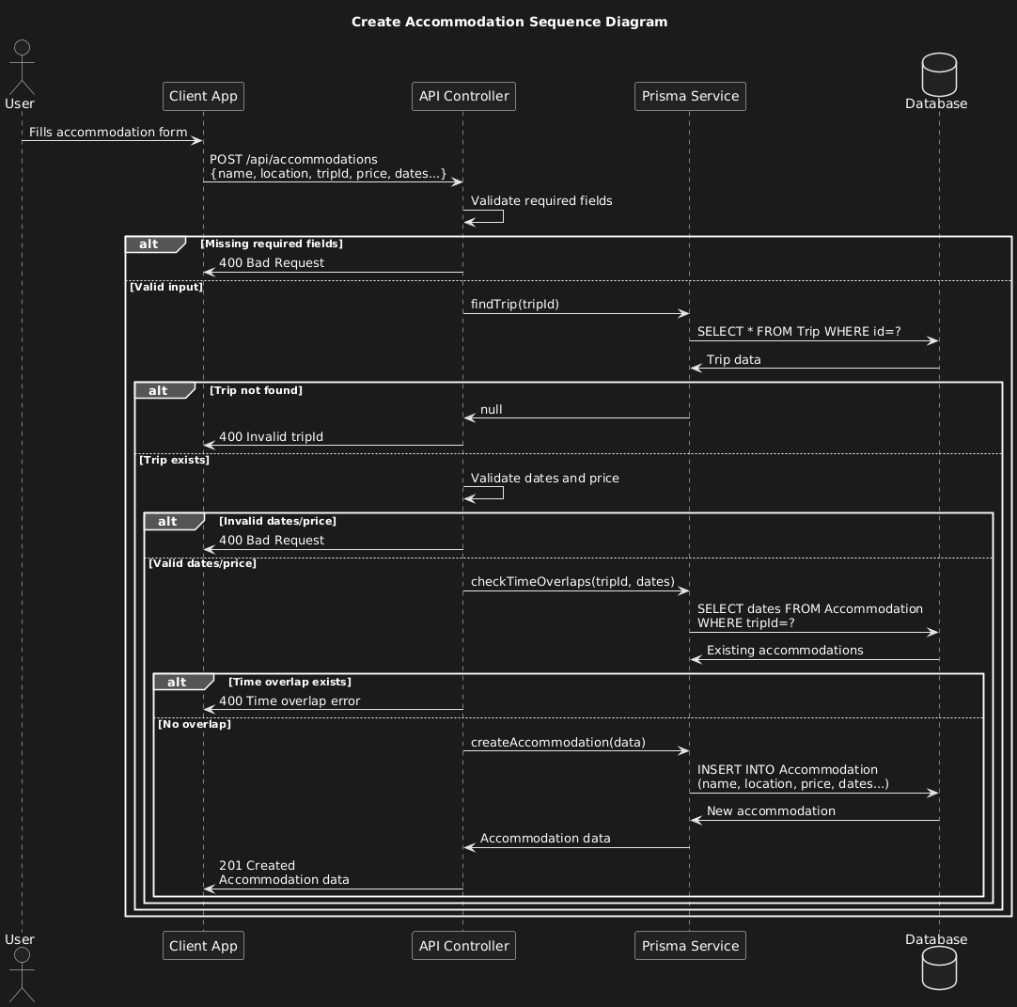
Update trip



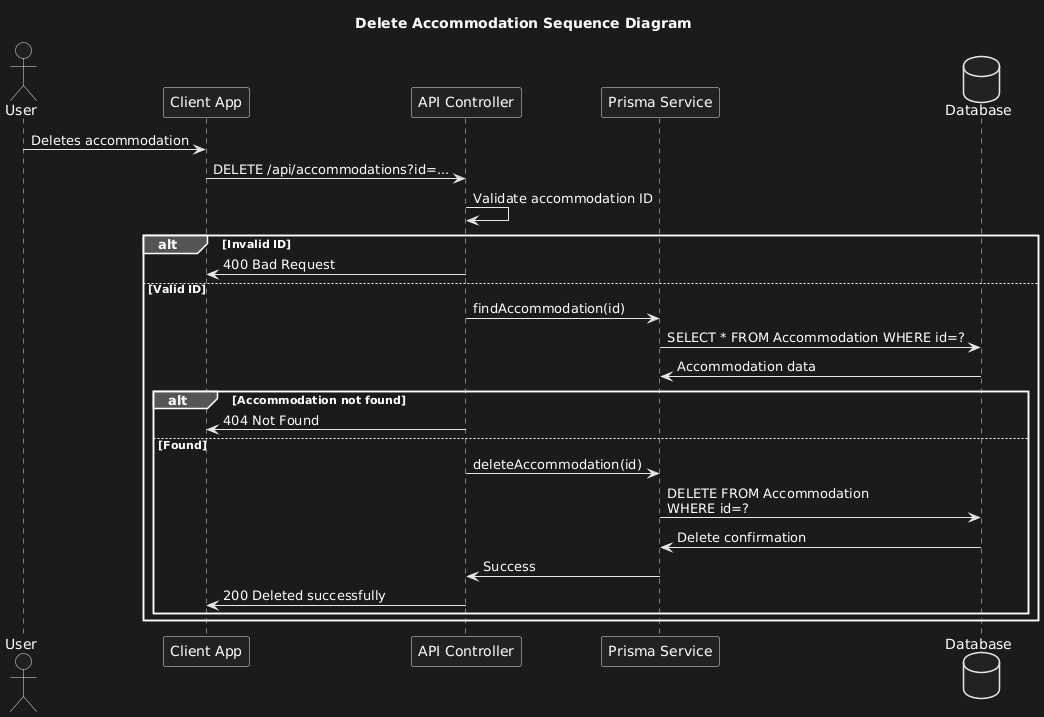
Delete trip



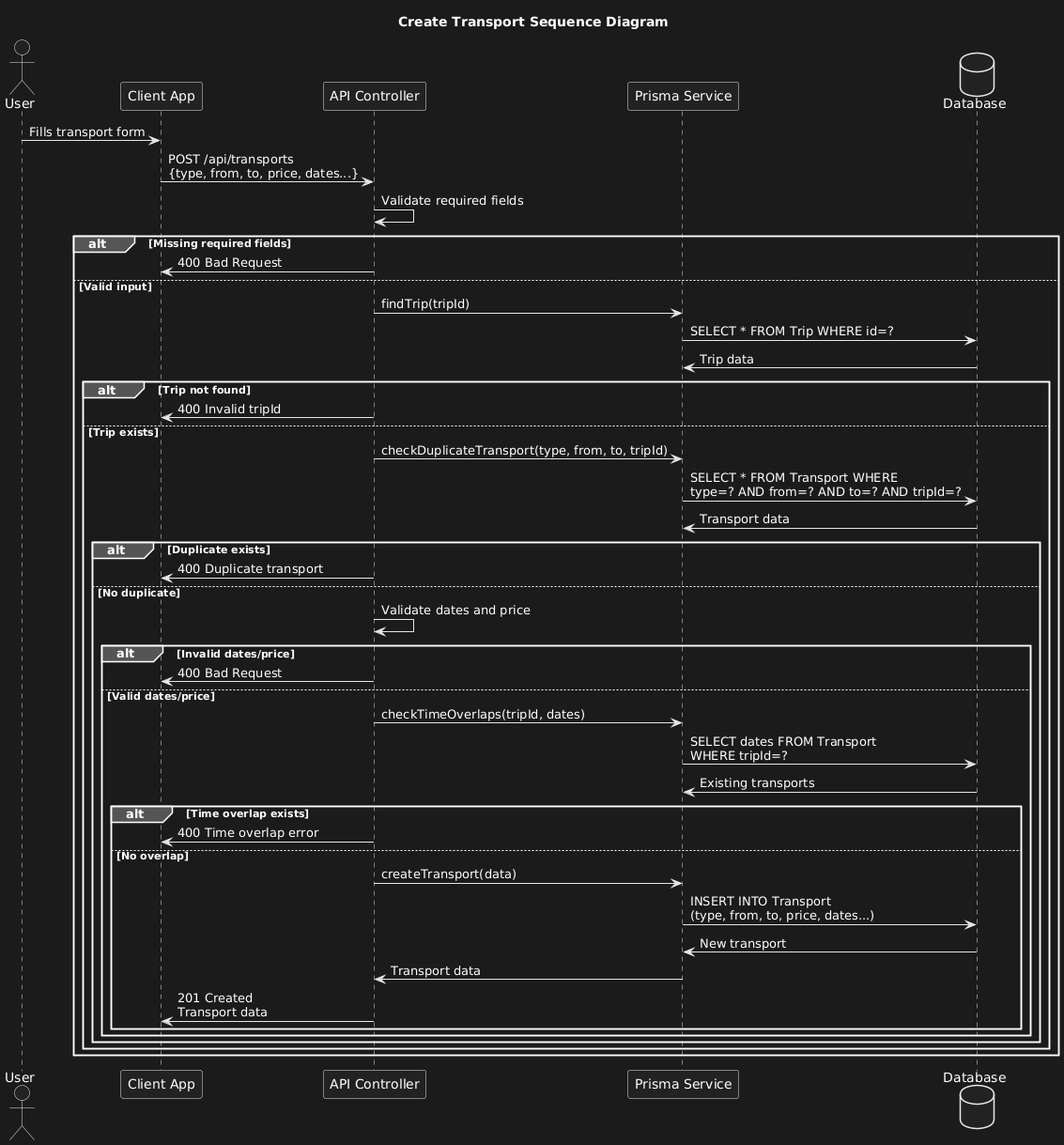
Add accommodation

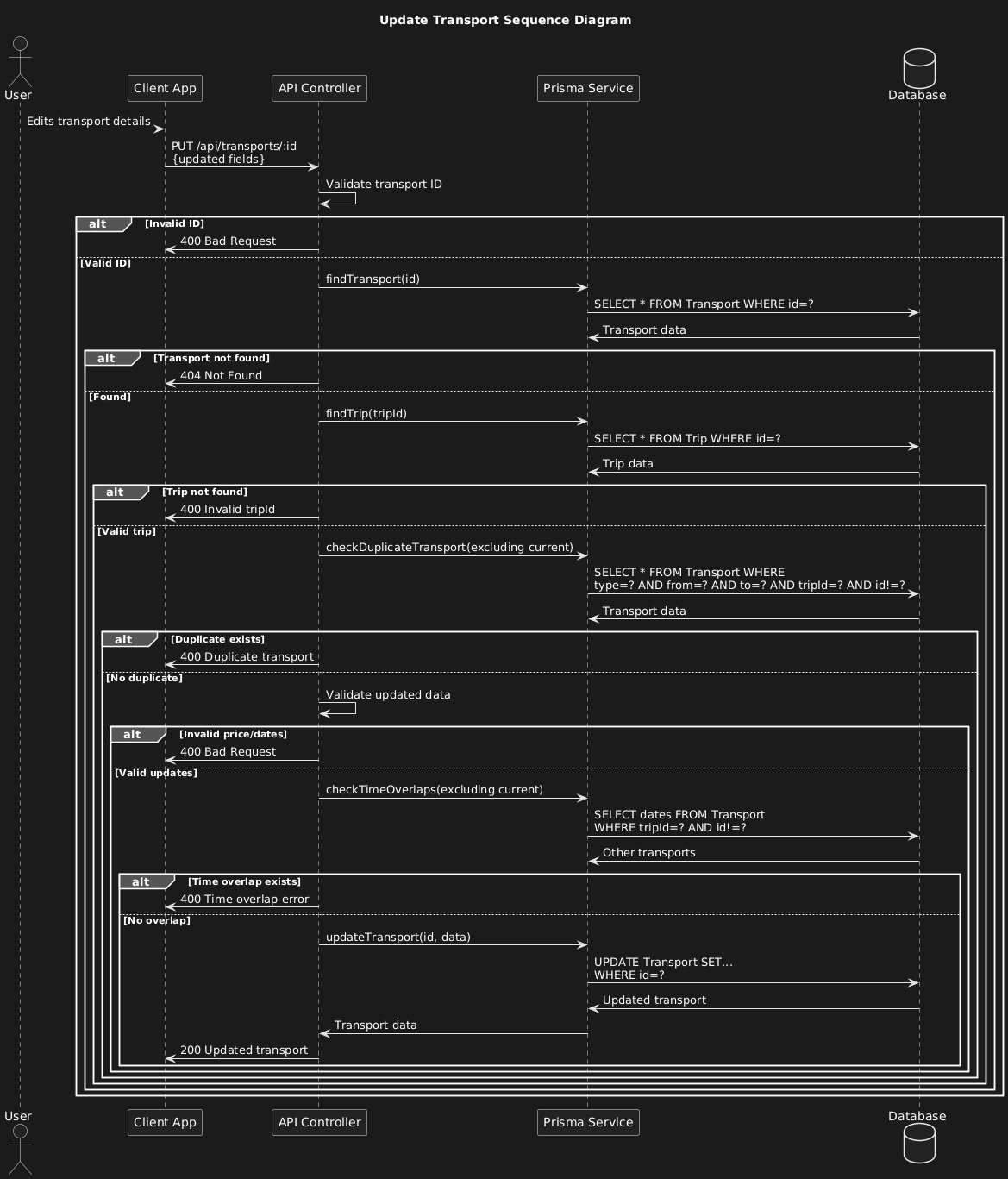


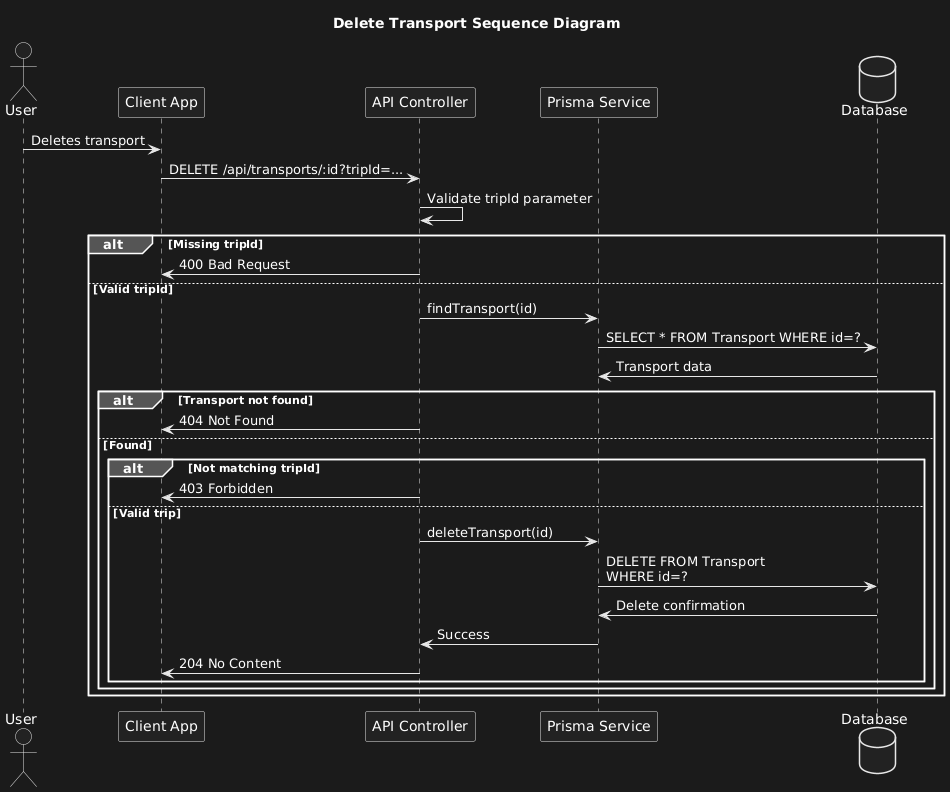
Delete accommodation



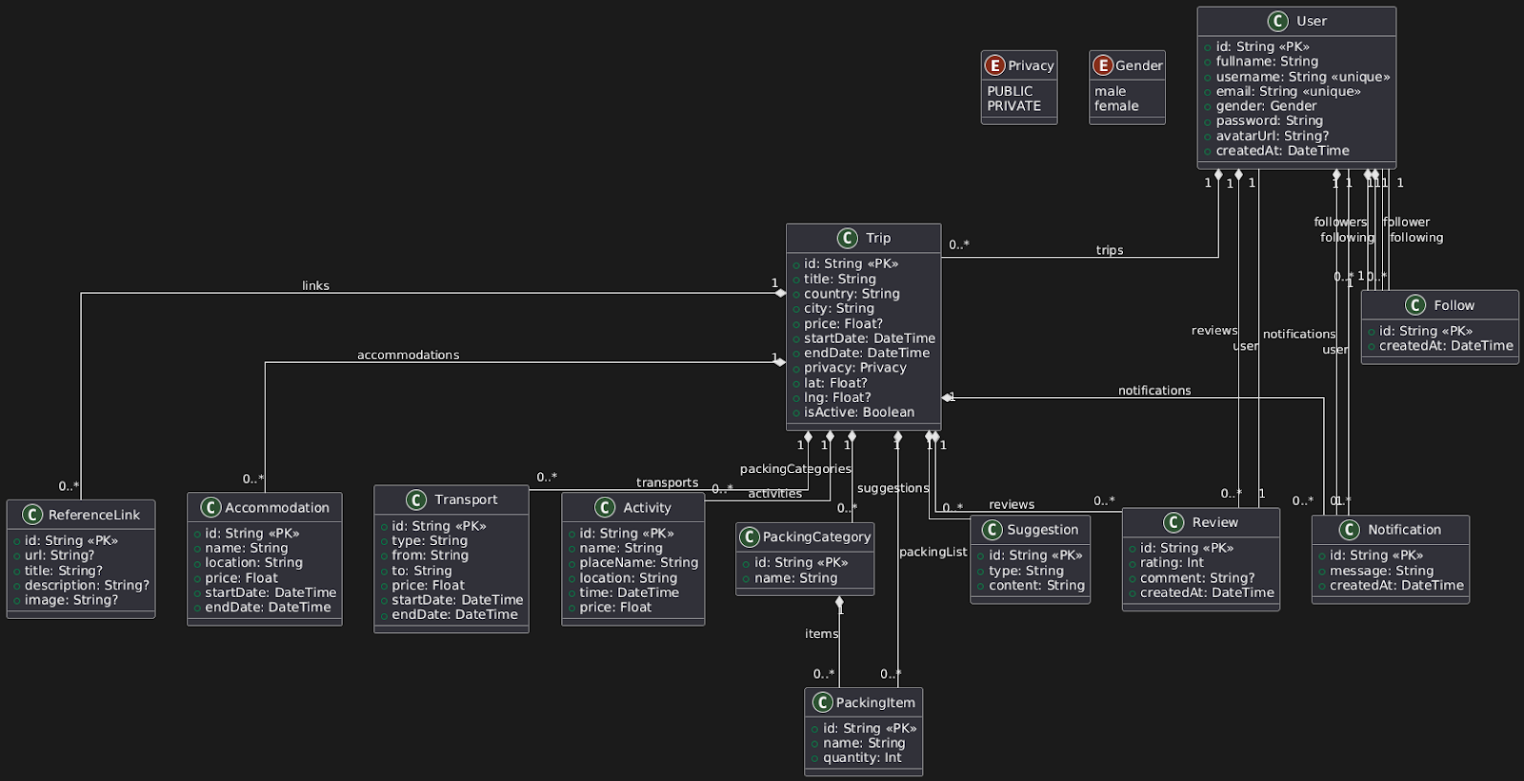
Add transport



Update transport  


Delete transport  


**3.4. Thiết kế lớp (Class Diagram)**



**3.5. Thiết kế cơ sở dữ liệu (ERD, bảng, mối quan hệ)**

**Chương 4. Cài đặt và triển khai hệ thống**

**4.1. Môi trường triển khai**

- Hệ thống được phát triển và triển khai trên nền tảng web sử dụng mô hình Client-Server với các công nghệ hiện đại, giúp đảm bảo khả năng mở rộng, bảo trì và hiệu suất cao.

| **Thành phần** | **Công nghệ sử dụng** |
| --- | --- |

|  |  |
| --- | --- |
| **Frontend** | ReactJS, TypeScript, TailwindCSS |

|  |  |
| --- | --- |
| **Backend** | Node.js, Express.js |

|  |  |
| --- | --- |
| **Cơ sở dữ liệu** | PostgreSQL, Prisma ORM |

|  |  |
| --- | --- |
| **Xác thực** | JWT (JSON Web Token) |

|  |  |
| --- | --- |
| **Triển khai** | Vercel (Frontend), Render / Railway / Supabase (Backend + Database) |

**4.2. Cài đặt hệ thống**

**4.2.1 Yêu cầu hệ thống**

- Cài đặt nodejs, postgresql.

**4.2.2 Các bước cài đặt chương trình**

**Frontend:**

### 1. Clone project về máy

git clone https://github.com/nguyenatu188/tpfrontend.git

### 2. Cài Dependencies:

npm i

### 3. Chạy development

npm run dev

**Backend:**

### 1. Clone project về máy

git clone https://github.com/nguyenatu188/tpbackend.git

### 2. Cài Dependencies

npm i

### Tạo file ".env". Tạo tài khoản và tạo project mới trên neon console. Ở phần connect to your database, nhấn connect, chuyển lựa chọn connection string sang prisma. Nhấn vào ".env" => copy paste vào file ".env" của mình

### 3. Set up các biến môi trường

.env file:

DATABASE\_URL="set up neon console để lấy url"

PORT=5001

JWT\_SECRET=mysecretkey

NODE\_ENV=development

CLIENT\_ID= người dùng set up Oath2 authentication và tạo credentials để lấy clientID

CLIENT\_SECRET=làm như trên để lấy client secret

EMAIL\_USER= “email muốn dùng để làm nguồn gửi email reset mật khẩu”

EMAIL\_PASS=kích hoạt bảo mật 2 lớp rồi copy app password vô đây.

JWT\_RESET\_SECRET=matmasieubimat

FE\_URL=http://localhost:{port frontend}

//đăng ký tài khoản cloudinary rồi copy 3 giá trị sau vào đây.

CLOUDINARY\_CLOUD\_NAME=

CLOUDINARY\_API\_KEY=

CLOUDINARY\_API\_SECRET=

### 4. Migrate database

Chạy lệnh: npx prisma generate && npx prisma migrate

### 5. Chạy development

npm run dev

**Chương 5. Kết quả và đánh giá**

5.1. Kết quả thử nghiệm hệ thống

5.1.1. Hệ thống đã được thử nghiệm trên với các chức năng chính được mô tả trong mục tiêu đề tài. Các bước thử nghiệm được thực hiện để đảm bảo hệ thống đáp ứng đầy đủ yêu cầu chức năng và phi chức năng. Kết quả thử nghiệm được tổng hợp như sau:

1. Thử nghiệm chức năng
   1. Đăng ký, đăng nhập, đăng xuất
      1. Người dùng có thể tạo tài khoản mới, đăng nhập và đăng xuất thành công.
      2. Người dùng có thể đăng nhập bằng google
   2. Quản lý chuyến đi
      1. Người dùng có thể tạo, chỉnh sửa, xóa kế hoạch chuyến đi với các thành phần như nơi ở, phương tiện, hoạt động và danh sách vật dụng.
      2. Hệ thống hỗ trợ thêm, cập nhật và xóa các mục như nơi ở, phương tiện di chuyển, hoạt động một cách mượt mà.
   3. Tính toán chi phí ước tính
      1. Hệ thống tự động tổng hợp chi phí từ các mục đã nhập và chia đều chi phí cho các thành viên tham gia chuyến đi.
      2. Kết quả hiển thị chính xác, phù hợp với dữ liệu đầu vào.
   4. Tích hợp bản đồ
      1. Bản đồ tương tác hiển thị chính xác các địa điểm quốc gia trong kế hoạch
   5. Quản lý quyền riêng tư và chia sẻ
      1. Người dùng có thể thiết lập quyền riêng tư cho kế hoạch (công khai hoặc riêng tư) và xem các kế hoạch công khai của người dùng khác.
   6. Trang cá nhân và theo dõi người dùng
      1. Người dùng có thể quản lý thông tin cá nhân, tìm kiếm và theo dõi người dùng khác, nhận thông báo khi có cập nhật từ người dùng được theo dõi.
2. Thử nghiệm hiệu năng
   * Thời gian tải trang trung bình dưới 3 giây đối với các thao tác chính (đăng nhập, tạo chuyến đi, xem bản đồ).
   * Hệ thống xử lý tốt với dữ liệu mẫu lên đến 1000 chuyến đi và 500 người dùng đồng thời trong môi trường thử nghiệm.
   * Tích hợp Prisma ORM giúp tối ưu truy vấn cơ sở dữ liệu, giảm thời gian phản hồi.
3. Thử nghiệm bảo mật
   * Xác thực JWT hoạt động hiệu quả, ngăn chặn truy cập trái phép vào các tài nguyên riêng tư.
   * Mật khẩu được mã hóa bằng bcrypt trước khi lưu vào cơ sở dữ liệu.
   * Các kiểm tra phân quyền đảm bảo người dùng chỉ truy cập được các kế hoạch mà họ có quyền xem.
4. Thử nghiệm tính khả dụng
   * Giao diện được thiết kế với Tailwind CSS, đảm bảo tính responsive trên các thiết bị (desktop, tablet, mobile).
   * Các thao tác drag-and-drop (nếu có) và menu điều hướng hoạt động mượt mà, dễ sử dụng.
   * Người dùng mới có thể làm quen với hệ thống trong vòng 5-10 phút nhờ giao diện trực quan.
5. Thử nghiệm lỗi
   * Các trường hợp ngoại lệ (nhập sai định dạng, hệ thống mất kết nối) được xử lý với thông báo lỗi rõ ràng, không gây crash hệ thống.
   * Một số lỗi nhỏ liên quan đến hiển thị giao diện trên trình duyệt cũ (như IE11) đã được ghi nhận và khắc phục.

**5.2. Đánh giá hiệu quả hệ thống**

Hệ thống đã đạt được các mục tiêu đề ra trong đồ án, mang lại hiệu quả cao trong việc hỗ trợ người dùng lập kế hoạch du lịch. Đánh giá cụ thể như sau:

1. **Hiệu quả về chức năng**:
   * Hệ thống đáp ứng đầy đủ các yêu cầu chức năng được liệt kê trong User Stories, từ quản lý chuyến đi, tính toán chi phí, đến tích hợp bản đồ và chia sẻ kế hoạch.
   * Các tính năng như gợi ý hoạt động và quản lý danh sách vật dụng giúp tiết kiệm thời gian và tăng tính tiện lợi cho người dùng.
   * Tính năng theo dõi người dùng và thông báo cập nhật giúp tăng tính cộng đồng và tương tác.
2. **Hiệu quả về hiệu năng**:
   * Hệ thống hoạt động ổn định với thời gian phản hồi nhanh, đáp ứng yêu cầu phi chức năng về hiệu năng.
   * Kiến trúc hướng sự kiện (Event-Driven Architecture) cho phép xử lý các tác vụ bất đồng bộ hiệu quả, giảm tải cho server.
3. **Hiệu quả về bảo mật và tính khả dụng**:
   * Các biện pháp bảo mật như JWT và mã hóa mật khẩu đảm bảo an toàn thông tin người dùng.
   * Giao diện thân thiện, dễ sử dụng, phù hợp với cả người dùng không có nhiều kinh nghiệm công nghệ.
4. **Hạn chế và hướng cải thiện**:
   * **Hạn chế**:
     + Chưa tích hợp AI để gợi ý địa điểm/hoạt động thông minh hơn.
     + Chưa có hệ thống quản trị viên để quản lý nội dung hoặc kiểm duyệt kế hoạch công khai.
     + Chưa hỗ trợ phiên bản mobile app, giới hạn trải nghiệm trên thiết bị di động.
   * **Hướng cải thiện**:
     + Tích hợp thuật toán AI hoặc API bên thứ ba (như Google Places) để cung cấp gợi ý địa điểm/hoạt động chính xác hơn.
     + Phát triển thêm vai trò quản trị viên để kiểm soát nội dung công khai.
     + Mở rộng hệ thống sang nền tảng mobile app để tăng tính tiện lợi.
     + Thêm tính năng đánh giá địa điểm hoặc lên kế hoạch nhóm để tăng tính tương tác.
5. **Ý nghĩa thực tiễn**:
   * Hệ thống cung cấp một giải pháp toàn diện cho việc lập kế hoạch du lịch, giúp người dùng tiết kiệm thời gian và tổ chức chuyến đi một cách hệ thống.
   * Đề tài giúp nhóm thực hiện củng cố kiến thức về phát triển web full-stack, từ thiết kế giao diện, lập trình backend, đến quản lý cơ sở dữ liệu và triển khai ứng dụng.
6. **Đánh giá tổng quan**:
   * Hệ thống đáp ứng tốt các yêu cầu đề ra, mang lại trải nghiệm người dùng tích cực và có tiềm năng mở rộng trong tương lai.
   * Việc áp dụng các công nghệ hiện đại (React, Node.js, PostgreSQL, JWT) giúp hệ thống đạt được độ tin cậy và hiệu năng cao, phù hợp với xu hướng phát triển phần mềm hiện nay.

Kết luận, đồ án đã hoàn thành các mục tiêu đề ra, tạo ra một sản phẩm có giá trị thực tiễn và là nền tảng để phát triển thêm các tính năng nâng cao trong tương lai.