Mục lục

[Giới thiệu 2](#_Toc50719029)

[Thuật ngữ 3](#_Toc50719030)

[Các yêu cầu 4](#_Toc50719031)

[Kiến trúc hệ thống 5](#_Toc50719032)

[1) Trang chủ: 5](#_Toc50719033)

[2) Trang quản trị: 5](#_Toc50719034)

[Đặc tả yêu cầu hệ thống 6](#_Toc50719035)

[1) Trang chủ: 6](#_Toc50719036)

[a) Tra cứu: 6](#_Toc50719037)

[b) Xem thông tin chuyến bay. 6](#_Toc50719038)

[2) Trang quản trị: 6](#_Toc50719039)

[a) Tương tác dữ liệu: Xem chi tiết cơ sở dữ liệu. 6](#_Toc50719040)

[b) Xem báo cáo thống kê: thống kê tổng doanh thu và loại vé bán được. 6](#_Toc50719041)

[c) Xem và thay đổi quy định: thay đổi các quy định trước đó. 6](#_Toc50719042)

[Các mô hình hệ thống 7](#_Toc50719043)

[Usecase 7](#_Toc50719044)

[SequenceDiagram 8](#_Toc50719045)

[Database Schema 9](#_Toc50719046)

# **Giới thiệu**

Flask là một trong các công nghệ phổ biến, ứng dụng web Flask tương đối rõ ràng.

Flask cũng dễ bắt đầu cho người mới bắt đầu vì có rất nhiều mã soạn sẵn để thiết lập và chạy một ứng dụng đơn giản.

Flask ban đầu được thiết kế và phát triển bởi Armin Ronacher như một trò đùa ngày Cá tháng Tư vào năm 2010. Mặc dù có nguồn gốc là một trò đùa, nhưng Flask framework đã trở nên phổ biến như một sự thay thế cho các dự án Django với cấu trúc nguyên khối và các phụ thuộc của chúng.

Thành công của Flask đã tạo ra rất nhiều công việc. Armin cuối cùng đã tạo ra bộ sưu tập thư viện mã nguồn mở The Pallets Projects sau khi anh quản lý Flask bằng tài khoản GitHub của riêng mình trong vài năm. Dự án Pallets hiện đóng vai trò là tổ chức do cộng đồng điều hành xử lý Flask và các thư viện Python liên quan khác như Lektor, Jinja và một số thư viện khác.

Đề tài quản lý bán vé chuyến bay được triển khai dễ dàng nhờ sử dụng Python Flask.

# **Thuật ngữ**

**Flask** là một khuôn khổ web vi mô được viết bằng Python.

Nó được phân loại là **microframework** vì nó không yêu cầu các công cụ hoặc thư viện cụ thể. Nó không có lớp trừu tượng cơ sở dữ liệu, xác thực biểu mẫu hoặc bất kỳ thành phần nào khác mà các thư viện bên thứ ba đã có từ trước cung cấp các chức năng chung.

Tuy nhiên, Flask hỗ trợ các tiện ích mở rộng có thể thêm các tính năng của ứng dụng như thể chúng được triển khai trong chính Flask. Các tiện ích mở rộng tồn tại cho người lập bản đồ quan hệ đối tượng, xác thực biểu mẫu, xử lý tải lên, các công nghệ xác thực mở khác nhau và một số công cụ liên quan đến khuôn khổ chung. Các tiện ích mở rộng được cập nhật thường xuyên hơn nhiều so với chương trình Flask cốt lõi.

Flask sử dụng bộ công cụ **Werkzeug WSGI** và bộ template **Jinja2**.

* ▪  WSGI (Web Server Gateway Interface) là chuẩn phát triển ứng dụng web trong python.
* ▪  Werkzeug thực thi các đối tượng request, reponse và các chức năng hữu ích khác.
* ▪  Jinga2 là một template engine phổ biến, kết hợp template với một data source nào đó để kết xuất hiển thị trên web động.

# **Các yêu cầu**

1. Nhận lịch vé chuyến bay: Trang web có thể hiển thị lịch chuyến bay và cho phép hiển thị thêm chi tiết của chuyến bay đó.
2. Bán vé: Khách hàng có thể điền thông tin theo biểu mẫu và đặt vé chuyến bay mình muốn.
3. Ghi nhận đặt vé: Hệ thống ghi nhận lưu trữ thông tin đặt vé của người dùng.
4. Tra cứu chuyến bay: Người dùng truy cập có thể tra cứu tìm kiếm chuyến bay.
5. Lập báo cáo: Có thể tra cứu lập báo cáo cho việc bán vé.
6. Thay đổi quy định: Được thay đổi quy định, giới hạn quy định,...

# **Kiến trúc hệ thống**

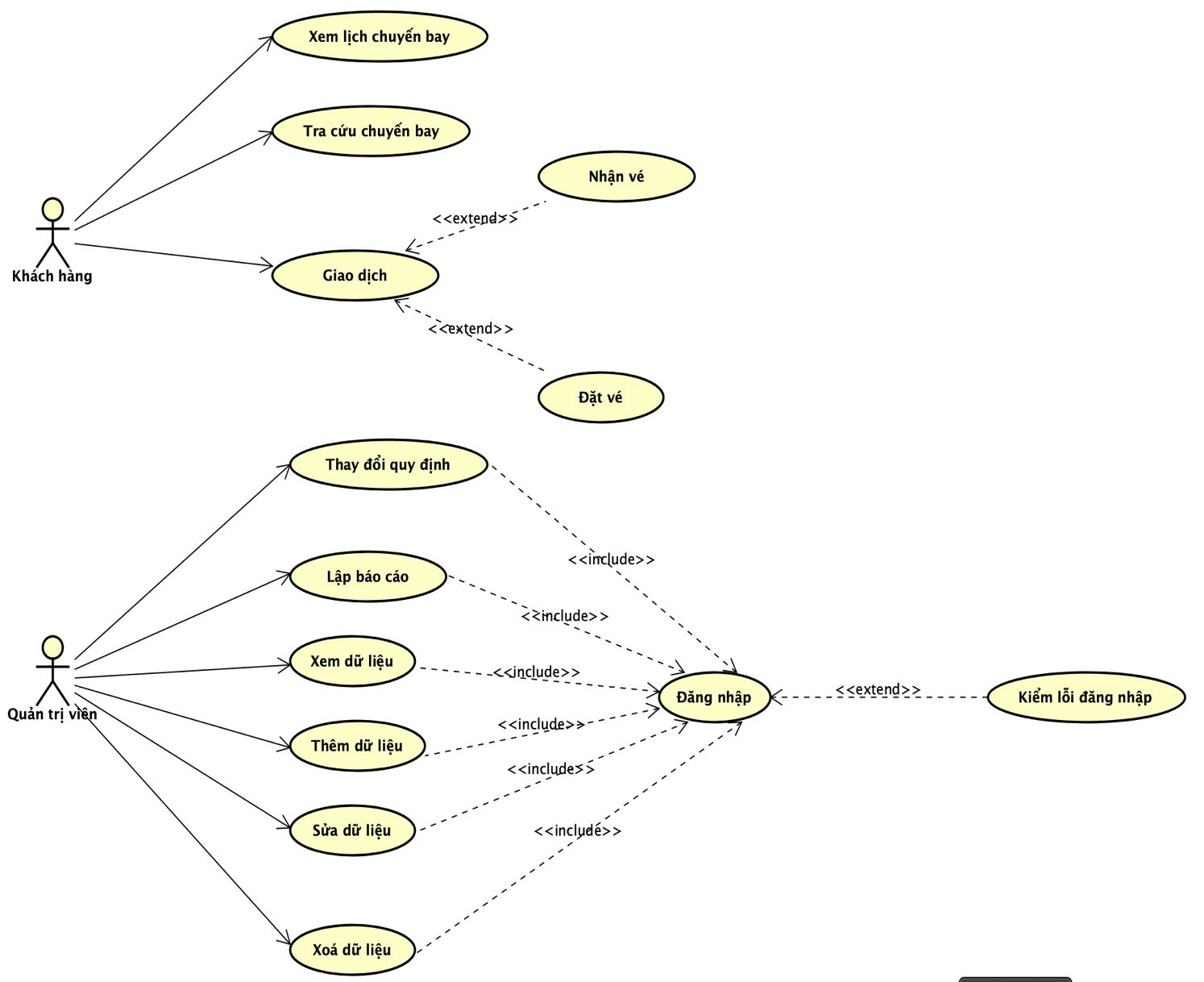
1. Trang chủ:
   1. Tra cứu:
      1. Tra cứu bằng từ khóa.
      2. Tra cứu dựa trên các trường có sẵn từ cơ sở dữ liệu.
   2. Xem thông tin chuyến bay.
      1. Xem thông tin chi tiết của chuyến bay.
      2. Đặt vé chuyến bay.
         1. Điền vào biểu mẫu đặt vé.
         2. Hủy đặt vé.
2. Trang quản trị:
   1. Tương tác dữ liệu.
      1. Thêm.
      2. Sửa.
      3. Xóa.
   2. Xem báo cáo thống kê.
   3. Xem và thay đổi quy định.

# **Đặc tả yêu cầu hệ thống**

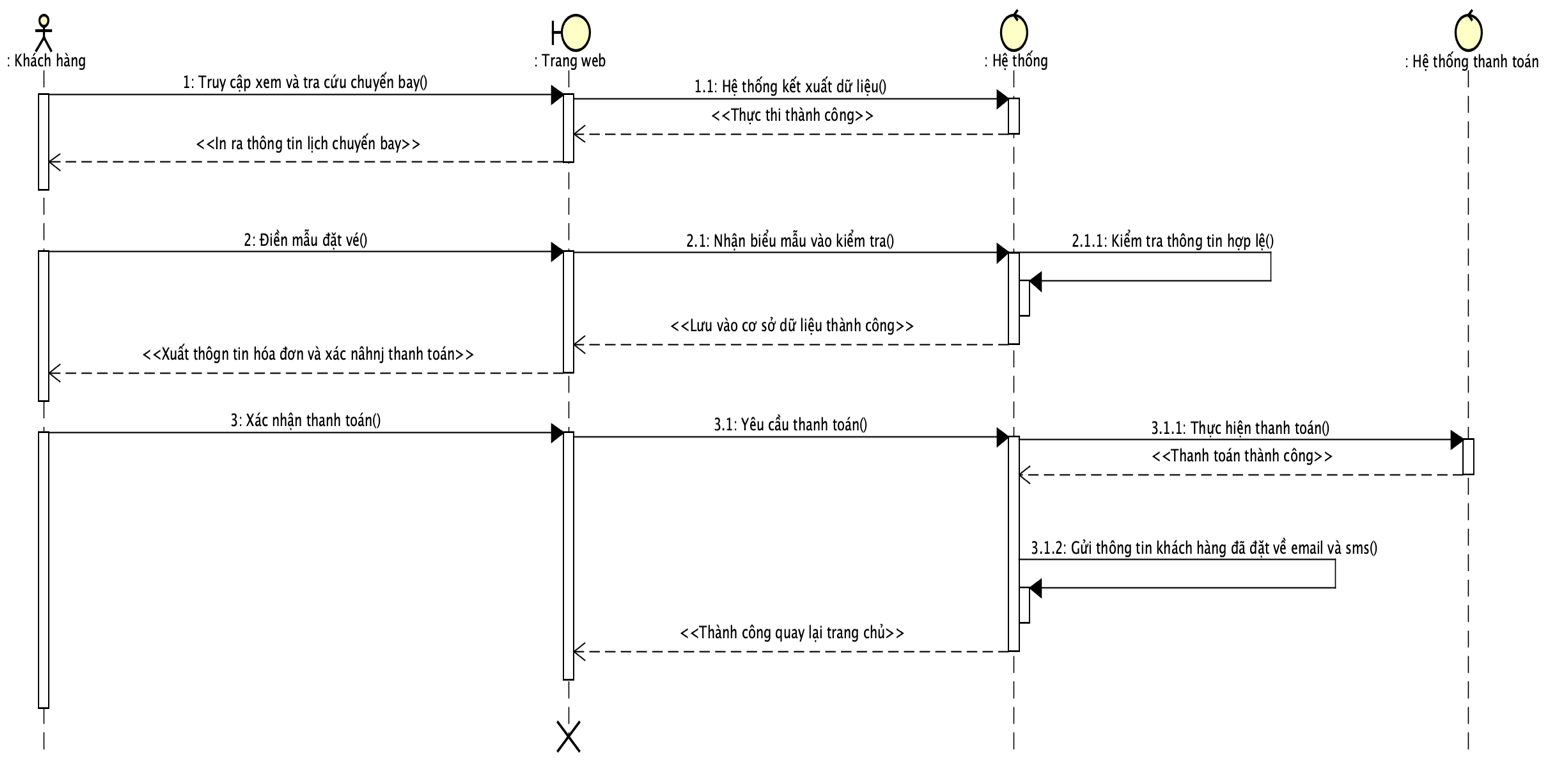
1. Trang chủ:
   1. Tra cứu:
      1. Tra cứu bằng từ khóa: Người dùng có thể nhập từ khóa bất kỳ và hệ thống sẽ tra cứu thông tin hai trường dữ liệu là *sân bay cất cánh* và *hạ cánh.*
      2. Tra cứu dựa trên các trường có sẵn từ cơ sở dữ liệu: Trang web sẽ hiển thị hai trường dữ liệu *sân bay cất cánh* và *hạ cánh* có sẵn từ cơ sở dữ liệu cho phép người dừng chọn giá trị và thực hiện tra cứu.
   2. Xem thông tin chuyến bay.
      1. Xem thông tin chi tiết của chuyến bay: Hiển thị chi tiết của chuyến bay.
      2. Đặt vé chuyến bay.
         1. Điền vào biểu mẫu đặt vé:
            1. Nếu thông tin điền vào phù hợp thì được lưu vào cơ sở dữ liệu và truy xuất lại thông tin cho người dùng.
            2. Nếu trường số điện thoại không hợp lệ thì báo lỗi không lưu dữ liệu vào cơ sở dữ liệu.
         2. Hủy đặt vé: chọn hủy đặt vé thông tin trên dữ liệu sẽ xóa theo trường mã hóa đơn.
2. Trang quản trị:
   1. Tương tác dữ liệu: Xem chi tiết cơ sở dữ liệu.
      1. Thêm.
      2. Sửa.
      3. Xóa.
   2. Xem báo cáo thống kê: thống kê tổng doanh thu và loại vé bán được.
   3. Xem và thay đổi quy định: Có tối đa 10 sân bay và thời gian bay tối thiểu là 30 phút.

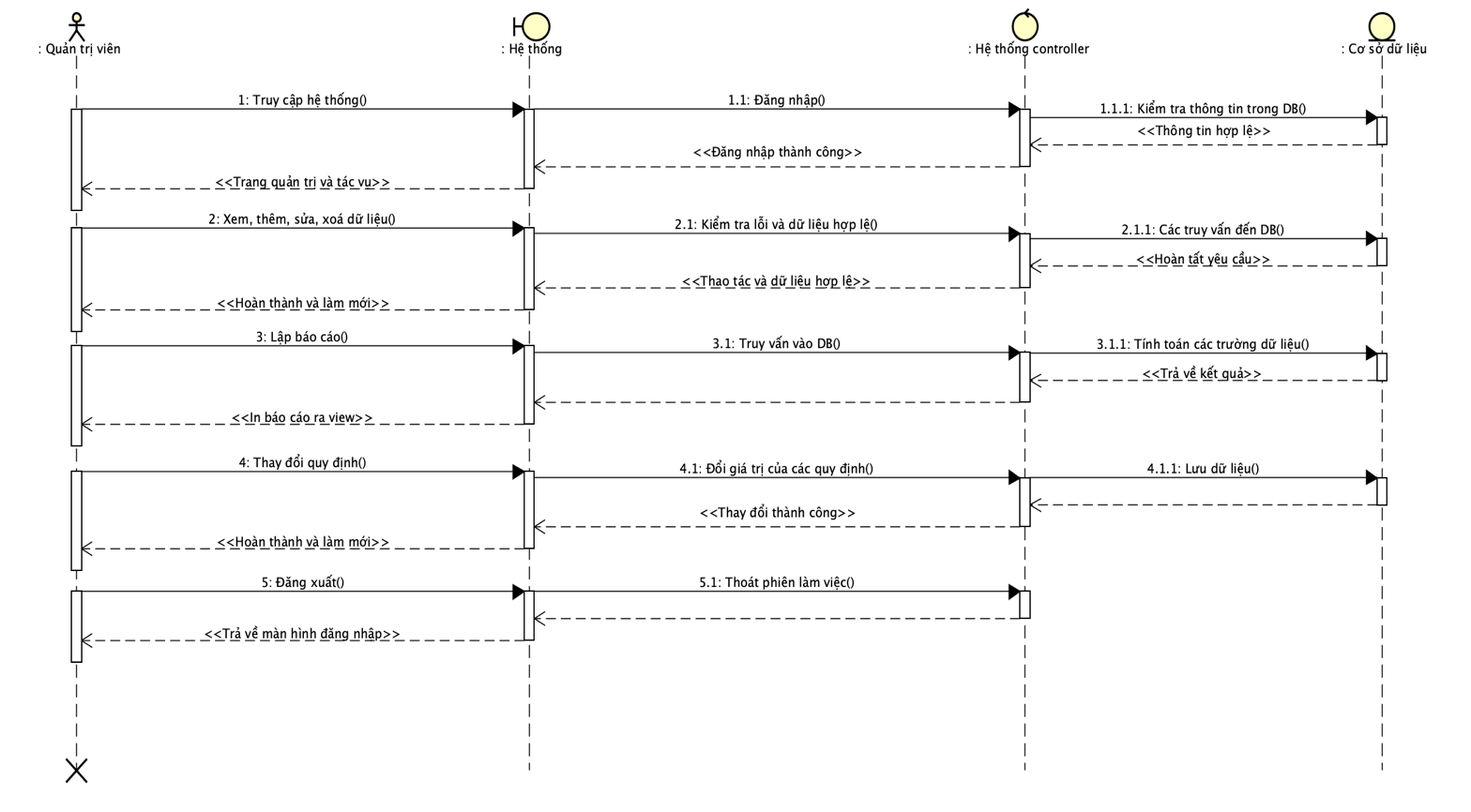
# **Các mô hình hệ thống**

## Usecase



## SequenceDiagram





## Database Schema

