|  |
| --- |
| **HỌC VIỆN HÀNG KHÔNG VIỆT NAM**  **KHOA CÔNG NGHỆ THÔNG TIN**      **BÁO CÁO**  **LẬP TRÌNH THIẾT BỊ DI ĐỘNG**  **QUẢN LÝ ĐẶT VÉ MÁY BAY**  **Giảng viên hướng dẫn: HUỲNH THANH SƠN**  **Sinh viên/ Nhóm sinh viên thực hiện: 09**  **Mã số sinh viên:**  **Lớp: 010100085501**  **TP. Hồ Chí Minh, tháng 07/2025** |
| **HỌC VIỆN HÀNG KHÔNG VIỆT NAM**  **KHOA CÔNG NGHỆ THÔNG TIN**      **BÁO CÁO**  **LẬP TRÌNH THIẾT BỊ DI ĐỘNG**  **QUẢN LÝ ĐẶT VÉ MÁY BAY**  **Giảng viên hướng dẫn: HUỲNH THANH SƠN**  **Sinh viên/ Nhóm sinh viên thực hiện: 09**  **Mã số sinh viên:**  **Lớp: 010100085501**  **Thành phố Hồ Chí Minh, tháng 07/2025** |

**Danh sách Nhóm:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Lớp** | **Ghi chú** |
| 1 | Dương Quang Minh | 2331540103 | 23ĐHTT02 | Nhóm Trưởng |
| 2 | Dương Minh Huy | 2331540043 | 23ĐHTT01 | Thành viên |
| 3 | Lê Tấn Đạt | 2331540097 | 23ĐHTT02 | Thành viên |
| 4 | Vương Trần Tuấn Anh | 2331540104 | 23ĐHTT02 | Thành viên |
| 5 | Nguyễn Đăng Khoa | 2331540030 | 23ĐHTT01 | Thành viên |

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1**  *(ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(ký và ghi rõ họ tên)* |
| **Cán bộ chấm thi phúc khảo 1**  *(ký và ghi rõ họ tên)* | **Cán bộ chấm thi phúc khảo 2**  *(ký và ghi rõ họ tên)* |

**DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| **Ký hiệu, chữ viết tắt** | **Chữ viết đầy đủ** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**DANH MỤC BẢNG**

# *Trang*

**Bảng 3.1.2 – Thiết kế bảng tác nhân 9**

**Bảng 3.1.3.a – Thiết kế bảng Usecase tổng quát 9**

**Bảng 3.1.3.b.1 – Thiết kế bảng Usecase đăng ký 12**

**Bảng 3.1.3.b.2 – Thiết kế bảng Usecase đăng nhập 15**

**Bảng 3.1.3.b.3 – Thiết kế bảng Usecase đăng xuất 17**

**Bảng 3.1.3.b.4 – Thiết kế bảng Usecase tra cứu chuyến bay 19**

**Bảng 3.1.3.b.5 – Thiết kế bảng Usecase đặt vé 22**

**Bảng 3.1.3.b.6 – Thiết kế bảng Usecase thanh toán 25**

**Bảng 3.1.3.b.7 – Thiết kế bảng Usecase lịch sử đặt vé 27**

**Bảng 3.1.3.b.8 – Thiết kế bảng Usecase quản lý hồ sơ cá nhân 29**

**Bảng 3.1.3.b.9 – Thiết kế bảng Usecase quản lý chuyến bay 32**

**Bảng 3.1.3.b.10 – Thiết kế bảng Usecase chuyến bay yêu thích 35**

**Bảng 3.2.1.1 – Thiết kế bảng NguoiDung 56**

**Bảng 3.2.1.2 – Thiết kế bảng ChuyenBay 57**

**Bảng 3.2.1.3 – Thiết kế bảng ChuyenBayYeuThich 57**

**Bảng 3.2.1.4 – Thiết kế bảng Booking 58**

**DANH MỤC HÌNH ẢNH**

# *Trang*

**Hình 2.1 – Logo Kotlin 3**

**Hình 2.2 – Logo Android Studio 4**

**Hình 2.3 – Logo Microsoft SQL Server 6**

**Hình 3.1.3.a – Usecase diagram 11**

**Hình 3.1.3.b.1 – Usecase đăng ký 14**

**Hình 3.1.3.b.2 – Usecase đăng nhập 17**

**Hình 3.1.3.b.3 – Usecase đăng xuất 19**

**Hình 3.1.3.b.4 – Usecase tra cứu chuyến bay 21**

**Hình 3.1.3.b.5 – Usecase đặt vé 24**

**Hình 3.1.3.b.6 – Usecase thanh toán 27**

**Hình 3.1.3.b.7 – Usecase lịch sử đặt vé 29**

**Hình 3.1.3.b.8 – Usecase quản lý hồ sơ cá nhân 31**

**Hình 3.1.3.b.9 – Usecase quản lý chuyến bay 34**

**Hình 3.1.3.b.10 – Usecase chuyến bay yêu thích 37**

**Hình 3.1.4.a – Sơ đồ đăng ký 38**

**Hình 3.1.4.b – Sơ đồ đăng nhập 39**

**Hình 3.1.4.c – Sơ đồ đăng xuất 40**

**Hình 3.1.4.d – Sơ đồ tra cứu chuyến bay 41**

**Hình 3.1.4.e – Sơ đồ đặt vé 42**

**Hình 3.1.4.f – Sơ đồ thanh toán 43**

**Hình 3.1.4.g – Sơ đồ lịch sử đặt vé 44**

**Hình 3.1.4.h – Sơ đồ quản lý hồ sơ cá nhân 45**

**Hình 3.1.4.i – Sơ đồ quản lý chuyến bay 46**

**Hình 3.1.4.j – Sơ đồ chuyến bay yêu thích 47**

**Hình 3.1.5.a – Sơ đồ tuần tự đăng ký 48**

**Hình 3.1.5.b – Sơ đồ tuần tự đăng nhập 48**

**Hình 3.1.5.c – Sơ đồ tuần tự đăng xuất 49**

**Hình 3.1.5.d – Sơ đồ tuần tự tra cứu chuyến bay 49**

**Hình 3.1.5.e – Sơ đồ tuần tự đặt vé 50**

**Hình 3.1.5.f – Sơ đồ tuần tự thanh toán 50**

**Hình 3.1.5.g – Sơ đồ tuần tự lịch sử đặt vé 51**

**Hình 3.1.5.h – Sơ đồ tuần tự quản lý hồ sơ cá nhân 51**

**Hình 3.1.5.i – Sơ đồ tuần tự chỉnh sửa hồ sơ cá nhân 52**

**Hình 3.1.5.j – Sơ đồ tuần tự quản lý chuyến bay 52**

**Hình 3.1.5.k – Sơ đồ tuần tự thêm chuyến bay 53**

**Hình 3.1.5.l – Sơ đồ tuần tự sửa chuyến bay 54**

**Hình 3.1.5.m – Sơ đồ tuần tự xoá chuyến bay 55**

**Hình 3.1.5.n – Sơ đồ tuần tự chuyến bay yêu thích 55**

**Hình 3.1.6 – Biểu đồ thực thể ERD 56**

**Hình 3.3.1 – Giao diện mở đầu 62**

**Hình 3.3.2 – Giao diện đăng nhập 63**

**Hình 3.3.3 – Giao diện đăng ký 64**

**Hình 3.3.4 – Giao diện trang chủ 65**

**Hình 3.3.5 – Giao diện tra cứu chuyến bay 66**

**Hình 3.3.6 – Giao diện đặt vé 67**

**Hình 3.3.7 – Giao diện thông tin người dùng 68**

**Hình 3.3.8 – Giao diện chỉnh sửa thông tin người dùng 69**

**Hình 3.3.9 – Giao diện quản lý chuyến bay 70**

**Hình 3.3.10 – Giao diện thêm chuyến bay 71**

**Hình 3.3.11 – Giao diện chỉnh sửa chuyến bay 72**

**Hình 3.3.12 – Giao diện chuyến bay yêu thích 73**

**MỤC LỤC**

# *Trang*

# **MỞ ĐẦU**

## **CHƯƠNG 1: GIỚI THIỆU 1**

# **1.1. Lý do chọn đề tài 1**

# **1.2. Mục tiêu đề tài 1**

# **1.3. Phạm vi đề tài 1**

# **1.4. Đối tượng nghiên cứu 1**

# **1.5. Phương pháp nghiên cứu 2**

# **1.6. Bố cục đề tài 2**

## **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3**

**2.1. Cơ sở lý thuyết A: Ngôn ngữ lập trình Kotlin 3**

# **2.2. Cơ sở lý thuyết B: Android Studio 4**

# **2.3. Cơ sở lý thuyết C: MySQL Server 6**

## **CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM 8**

# **3.1. Phân tích hệ thống 8**

3.1.1. Khảo sát 8

3.1.2. Tác nhân 9

3.1.3. Usecase và đặc tả 9

3.1.4. Hoạt động 38

3.1.5. Tuần tự 48

3.1.6. Biểu đồ thực thể ERD 56

# **3.2. Thiết kế cơ sở dữ liệu 56**

3.2.1. Thiết kế 56

3.2.2. Tạo cấu trúc và dữ liệu trong hệ quản trị CSDL 58

# **3.3. Xây dựng giao diện sản phẩm 62**

3.3.1. Giao diện mở đầu 62

3.3.2. Giao diện đăng nhập 63

3.3.3. Giao diện đăng ký 64

3.3.4. Giao diện trang chủ 65

3.3.5. Giao diện tra cứu chuyến bay 66

3.3.6. Giao diện đặt vé chuyến bay 67

3.3.7. Giao diện thông tin người dùng 68

3.3.8. Giao diện chỉnh sửa thông tin người dùng 69

3.3.9. Giao diện quản lý chuyến bay 70

3.3.10. Giao diện thêm chuyến bay 71

3.3.11. Giao diện chỉnh sửa chuyến bay 72

3.3.12. Giao diện chuyến bay yêu thích 73

# **KẾT LUẬN 74**

# **PHỤ LỤC 76**

# **DANH MỤC TÀI LIỆU THAM KHẢO 77**

**MỞ ĐẦU**

Trong thời đại công nghệ 4.0 hiện nay, công nghệ thông tin đã và đang đóng vai trò then chốt trong việc đổi mới, nâng cao chất lượng dịch vụ và tối ưu hóa quy trình vận hành trong hầu hết các lĩnh vực của đời sống xã hội. Trong đó, ngành vận tải hàng không – một ngành có tốc độ tăng trưởng mạnh mẽ và nhu cầu sử dụng dịch vụ ngày càng cao – cũng không nằm ngoài xu hướng chuyển đổi số mạnh mẽ này.

Tại Việt Nam, với sự gia tăng của tầng lớp trung lưu, nhu cầu đi lại bằng máy bay cho cả mục đích công việc lẫn du lịch ngày càng trở nên phổ biến. Tuy nhiên, việc đặt vé máy bay qua các kênh truyền thống đôi khi còn gặp bất tiện: từ việc người dùng phải thao tác trên những website không tối ưu, đến việc không có thông tin trực quan về chuyến bay, giá cả hoặc thiếu sự hỗ trợ trong quá trình đặt vé. Điều đó đặt ra nhu cầu cấp thiết về một ứng dụng di động hiện đại, đơn giản, thuận tiện và thân thiện với người dùng để hỗ trợ quá trình tìm kiếm và đặt vé một cách hiệu quả.

Xuất phát từ thực tiễn trên, nhóm chúng em đã thực hiện đề tài **“Quản lý đặt vé máy bay”** – một ứng dụng Android cho phép người dùng tra cứu, lựa chọn chuyến bay phù hợp và thực hiện đặt vé trực tiếp trên điện thoại. Ứng dụng không chỉ phục vụ nhu cầu tìm kiếm và đặt vé đơn giản mà còn hỗ trợ người dùng quản lý thông tin cá nhân, theo dõi lịch sử đặt vé, và xuất hóa đơn sau khi hoàn tất giao dịch.

Đề tài được xây dựng dựa trên nền tảng **Android Studio**, sử dụng ngôn ngữ lập trình **Kotlin** – một ngôn ngữ hiện đại, an toàn và đang được Google khuyến khích cho phát triển ứng dụng Android. Ngoài ra, nhóm còn tích hợp cơ sở dữ liệu nội bộ **SQLite** để lưu trữ và xử lý thông tin người dùng, chuyến bay và đơn đặt vé một cách hiệu quả.

Mục tiêu của đề tài là tạo ra một sản phẩm phần mềm có giao diện đẹp, dễ sử dụng, đáp ứng tốt các chức năng đặt ra và có khả năng mở rộng trong tương lai. Thông qua quá trình thực hiện đề tài, nhóm sinh viên không chỉ nâng cao kỹ năng lập trình Android mà còn tiếp cận được quy trình xây dựng một phần mềm hoàn chỉnh, từ khâu thiết kế cơ sở dữ liệu, xây dựng giao diện người dùng đến xử lý nghiệp vụ và kiểm thử ứng dụng.

Với tinh thần học hỏi và ứng dụng kiến thức vào thực tiễn, nhóm hy vọng rằng đề tài “Quản lý đặt vé máy bay” sẽ là một bước khởi đầu hữu ích trong hành trình tiếp cận công nghệ phát triển ứng dụng di động, đồng thời góp phần giải quyết một bài toán thực tế mang tính ứng dụng cao.

**CHƯƠNG 1. GIỚI THIỆU**

**1.1. Lý do chọn đề tài**

Xuất phát từ thực tế hiện nay, nhu cầu đi lại bằng đường hàng không ngày càng tăng cao, đặc biệt là ở các thành phố lớn. Tuy nhiên, không phải ai cũng có thể dễ dàng tiếp cận các nền tảng đặt vé phức tạp hoặc yêu cầu mạng mạnh, tài khoản ngân hàng. Vì vậy, việc xây dựng một ứng dụng di động đơn giản, thân thiện, tích hợp các tính năng tra cứu chuyến bay, đặt vé, quản lý thông tin người dùng... là rất cần thiết. Việc giải quyết được bài toán này sẽ mang lại lợi ích lớn cho người dùng phổ thông, các hãng du lịch nhỏ, cũng như giúp sinh viên nâng cao kỹ năng xây dựng phần mềm di động thực tiễn. Ngoài ra, ứng dụng này có thể mở rộng trong các lĩnh vực khác như du lịch, quản lý vận chuyển hay thương mại điện tử.

**1.2. Mục tiêu đề tài**

* Xây dựng một ứng dụng Android có khả năng tra cứu và đặt vé máy bay dễ dàng.
* Hệ thống cho phép quản lý thông tin chuyến bay, thông tin người dùng, chức năng đăng ký – đăng nhập, lọc chuyến bay theo ngày và nơi đi – đến.
* Ứng dụng sử dụng ngôn ngữ lập trình Kotlin và công cụ Android Studio, kết hợp SQLite làm cơ sở dữ liệu nội bộ.
* Cung cấp một nền tảng học tập thực tiễn để áp dụng kiến thức đã học về lập trình mobile, cơ sở dữ liệu và thiết kế giao diện người dùng.

**1.3. Phạm vi đề tài**

* Nền tảng: Android (phiên bản từ Android 8.0 trở lên).
* Đối tượng sử dụng: người dùng cá nhân có nhu cầu tra cứu và đặt vé máy bay.
* Dữ liệu: sử dụng SQLite để lưu trữ các thông tin về chuyến bay, người dùng, đơn đặt vé.
* Thời gian thực hiện: trong học kỳ hiện tại.
* Đề tài không xử lý thanh toán trực tuyến, không kết nối với API các hãng hàng không thực tế.

**1.4 Đối tượng nghiên cứu**

* Hành vi đặt vé máy bay của người dùng trên thiết bị di động.
* Cách tổ chức cơ sở dữ liệu cho hệ thống đặt vé.
* Quy trình thiết kế giao diện và xử lý sự kiện trong ứng dụng Android sử dụng Kotlin.
* Mô hình hoạt động CRUD (Create, Read, Update, Delete) trong hệ thống quản lý đặt vé.

**1.5. Phương pháp nghiên cứu**

* **Phương pháp thu thập thông tin:** tìm hiểu tài liệu kỹ thuật từ trang chủ Android, tài liệu Kotlin chính thức, giáo trình Android cơ bản, tham khảo các ứng dụng đặt vé thực tế.
* **Phương pháp xử lý thông tin:** phân tích yêu cầu, thiết kế chức năng dựa trên hành vi người dùng, xây dựng sơ đồ use-case và database.
* **Phương pháp thực nghiệm:** lập trình thử nghiệm các module nhỏ, kiểm tra từng phần trước khi tích hợp tổng thể.

**1.6. Bố cục đề tài**

Phần còn lại của báo cáo được tổ chức như sau:

* **Chương 2** nhóm em trình bày về cơ sở lý thuyết và công nghệ nền tảng được sử dụng trong quá trình phát triển ứng dụng, bao gồm ngôn ngữ Kotlin, công cụ Android Studio và Microsoft SQL Server.
* **Chương 3** nhóm em trình bày quá trình phân tích hệ thống, thiết kế cơ sở dữ liệu, giao diện người dùng và cách xây dựng sản phẩm. Đây là phần chính thể hiện quy trình hiện thực hóa ứng dụng quản lý đặt vé máy bay.

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

**2.1. Cơ sở lý thuyết A – Ngôn ngữ lập trình Kotlin**

***🔹 Khái quát về Kotlin***

**Hình 2.1 – Logo Kotlin**

**Kotlin** là một ngôn ngữ lập trình hướng đối tượng hiện đại, được phát triển bởi JetBrains và chính thức được Google công nhận là ngôn ngữ ưu tiên cho phát triển ứng dụng Android kể từ năm 2017. Kotlin nhanh chóng trở thành sự lựa chọn hàng đầu trong cộng đồng lập trình viên nhờ cú pháp ngắn gọn, rõ ràng, cùng nhiều tính năng ưu việt hỗ trợ lập trình an toàn và hiệu quả.

Một điểm nổi bật của Kotlin là cơ chế kiểm soát lỗi null (null safety), giúp giảm thiểu tối đa các lỗi phổ biến liên quan đến giá trị null – nguyên nhân chủ yếu gây ra lỗi trong các ứng dụng Android truyền thống. Bên cạnh đó, Kotlin cho phép kết hợp linh hoạt với Java, tận dụng tối đa các thư viện sẵn có trong hệ sinh thái Android mà không gặp trở ngại về khả năng tương thích.

Kotlin còn cung cấp nhiều tính năng lập trình hiện đại như **extension functions** (mở rộng chức năng cho lớp mà không cần kế thừa), **smart casting** (tự động nhận diện kiểu dữ liệu hợp lý), và **lambda expressions** (hàm ẩn danh), giúp tăng năng suất lập trình, rút ngắn mã nguồn và làm cho code dễ đọc, dễ bảo trì hơn so với các ngôn ngữ truyền thống.

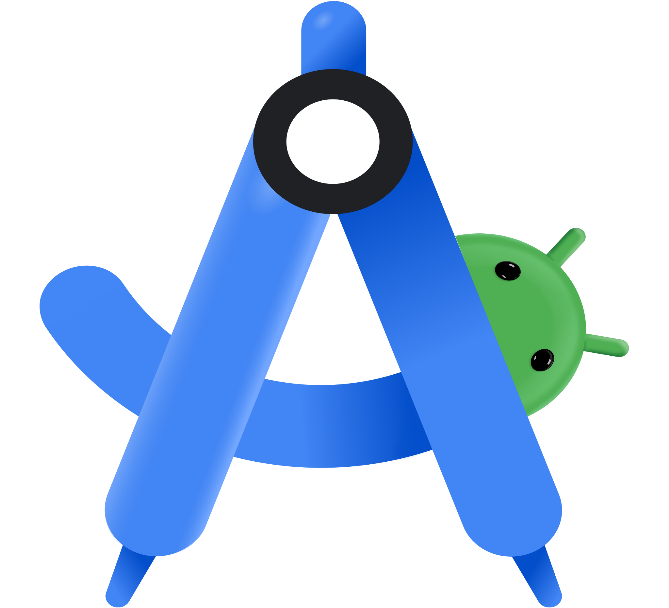
***🔹 Ứng dụng trong đề tài***  
Trong đề tài **“Quản lý đặt vé máy bay”**, **Kotlin** được sử dụng làm ngôn ngữ chính để phát triển toàn bộ logic và chức năng của ứng dụng, bao gồm:

* **Xử lý thao tác logic:** Cài đặt các chức năng nghiệp vụ như tìm kiếm, lọc chuyến bay phù hợp, đặt vé, kiểm tra lịch sử giao dịch, ...
* **Điều hướng ứng dụng:** Quản lý luồng chuyển đổi giữa các màn hình (activities, fragments) bằng các intent, đảm bảo trải nghiệm người dùng mượt mà và nhất quán.
* **Quản lý dữ liệu người dùng:** Xử lý các thao tác đăng ký, đăng nhập, cập nhật thông tin cá nhân, kiểm tra hợp lệ dữ liệu đầu vào, đảm bảo an toàn và chính xác.
* **Kết nối và thao tác với cơ sở dữ liệu:** Thực hiện các tác vụ lưu trữ, truy xuất thông tin người dùng, chuyến bay, lịch sử đặt vé thông qua SQLite, đảm bảo dữ liệu luôn được quản lý nhất quán và an toàn.

🔹 ***Lý do chọn***

* **Cú pháp ngắn gọn, dễ hiểu:** Kotlin giúp lập trình viên viết mã nguồn nhanh hơn, dễ đọc, giảm thiểu các lỗi do thao tác dư thừa như khi sử dụng Java.
* **An toàn và đáng tin cậy:** Tính năng kiểm soát null hiệu quả giúp hạn chế tối đa lỗi runtime, nâng cao chất lượng và độ ổn định của ứng dụng.
* **Khả năng tương thích hoàn toàn với Java:** Cho phép tận dụng mọi thư viện, API, và các công cụ hỗ trợ sẵn có của hệ sinh thái Android, giúp tiết kiệm thời gian phát triển.
* **Cộng đồng phát triển rộng lớn:** Kotlin ngày càng được sử dụng rộng rãi trong các dự án thực tế, có tài liệu học tập và nguồn hỗ trợ phong phú, rất phù hợp với sinh viên trong môi trường học tập và nghiên cứu.
* **Xu hướng phát triển hiện đại:** Google đã xác nhận Kotlin là ngôn ngữ được ưu tiên cho phát triển Android, do đó việc học và sử dụng Kotlin giúp sinh viên bắt kịp xu hướng, đồng thời mở rộng cơ hội nghề nghiệp trong lĩnh vực lập trình di động.

**2.2. Cơ sở lý thuyết B – Android Studio**

***🔹 Khái quát về Android Studio***

**Hình 2.2 – Logo Android Studio**

**Android Studio** là môi trường phát triển tích hợp (IDE – Integrated Development Environment) chính thức do Google phát triển và cung cấp dành riêng cho lập trình ứng dụng Android. Android Studio tích hợp đầy đủ các công cụ cần thiết hỗ trợ toàn bộ quy trình xây dựng ứng dụng từ thiết kế giao diện, lập trình logic, kiểm thử đến đóng gói và phát hành sản phẩm.

Một trong những điểm nổi bật của Android Studio là khả năng thiết kế giao diện người dùng linh hoạt, cho phép lập trình viên lựa chọn giữa thao tác trực tiếp với mã XML hoặc sử dụng công cụ thiết kế trực quan (Layout Editor) kéo – thả tiện lợi. Bên cạnh đó, môi trường này còn tích hợp sẵn trình biên dịch mạnh mẽ Gradle, hệ thống giả lập thiết bị đa dạng (Android Emulator), cũng như các công cụ debug hiện đại, giúp người dùng dễ dàng theo dõi luồng dữ liệu, kiểm soát quá trình xử lý và phát hiện lỗi trong quá trình phát triển ứng dụng.

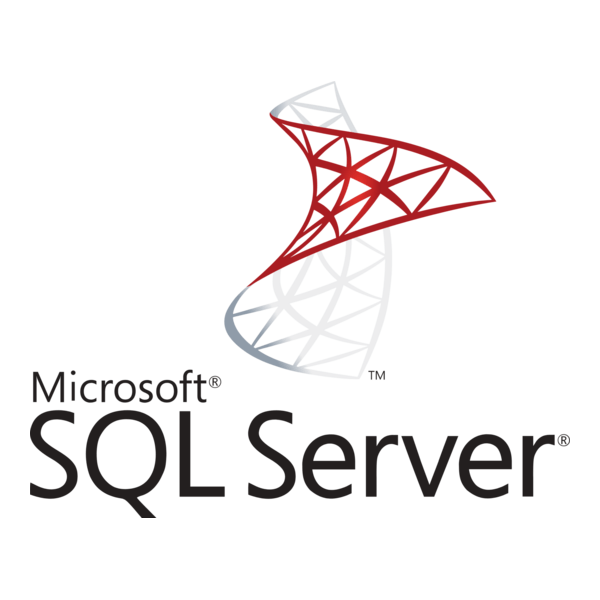
***🔹 Ứng dụng trong đề tài***  
Trong đề tài **“Quản lý đặt vé chuyến bay”**, Android Studio được sử dụng làm công cụ phát triển chủ đạo, đảm nhận các nhiệm vụ sau:

* **Xây dựng giao diện người dùng:** Thiết kế các màn hình chức năng như đăng nhập, đăng ký, tìm kiếm chuyến bay, đặt vé, lịch sử đặt vé, quản lý tài khoản, … giúp nâng cao trải nghiệm người dùng cuối.
* **Phát triển logic ứng dụng:** Lập trình các chức năng cốt lõi bằng ngôn ngữ Kotlin – ngôn ngữ được Google khuyến nghị cho phát triển Android hiện đại, đảm bảo hiệu năng, an toàn và dễ bảo trì.
* **Kiểm thử và chạy thử ứng dụng:** Thực hiện kiểm thử toàn diện trên giả lập thiết bị hoặc điện thoại thật để đảm bảo ứng dụng vận hành ổn định, đáp ứng các yêu cầu nghiệp vụ đã đề ra.
* **Quản lý tài nguyên và cấu trúc dự án:** Tổ chức thư mục dự án, quản lý thư viện, tài nguyên hình ảnh, dữ liệu chuyến bay, thông tin người dùng… một cách khoa học, thuận tiện cho việc mở rộng và bảo trì ứng dụng trong tương lai.

***🔹 Lý do chọn***

* **Chính thức và toàn diện:** Android Studio là IDE được Google phát triển và cập nhật thường xuyên, đảm bảo tương thích hoàn hảo với hệ điều hành Android và các công nghệ liên quan.
* **Hỗ trợ ngôn ngữ lập trình hiện đại:** Android Studio hỗ trợ tối ưu cho Kotlin – ngôn ngữ lập trình chính của dự án, cùng với khả năng tích hợp Java nếu cần.
* **Cộng đồng phát triển lớn mạnh:** Có hệ sinh thái tài liệu, diễn đàn, nhóm hỗ trợ phong phú, giúp sinh viên dễ dàng tìm kiếm hướng dẫn, giải đáp thắc mắc và chia sẻ kinh nghiệm trong quá trình học tập.
* **Nhiều tài nguyên học tập:** Android Studio sở hữu kho tài nguyên học tập đa dạng từ tài liệu chính thức, các khóa học miễn phí đến video hướng dẫn thực tế, phù hợp với môi trường đào tạo và nghiên cứu.
* **Dễ sử dụng, dễ triển khai:** Việc cài đặt, cấu hình và sử dụng Android Studio rất đơn giản, phù hợp cho cả người mới bắt đầu cũng như những sinh viên đã có kinh nghiệm lập trình.
* **Công cụ kiểm thử mạnh mẽ:** Hệ thống giả lập và các công cụ debug giúp kiểm thử ứng dụng hiệu quả, tiết kiệm thời gian và đảm bảo chất lượng sản phẩm đầu ra.

**2.3. Cơ sở lý thuyết C – Microsoft SQL Server**

*🔹 Khái quát về Microsoft SQL Server*

**Hình 2.3 – Logo Microsoft SQL Server**

**Microsoft SQL Server** là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) do Microsoft phát triển. Đây là một trong những giải pháp quản lý dữ liệu mạnh mẽ và phổ biến nhất hiện nay, được ứng dụng rộng rãi trong các hệ thống doanh nghiệp, ứng dụng thương mại, và các giải pháp dữ liệu lớn.

SQL Server hỗ trợ lưu trữ, truy xuất và xử lý dữ liệu một cách an toàn, đồng thời cung cấp các công cụ quản trị mạnh mẽ cho phép người dùng dễ dàng thiết kế, quản lý cơ sở dữ liệu, thực thi các truy vấn SQL cũng như thực hiện các thao tác sao lưu, khôi phục dữ liệu và bảo mật hệ thống.

Các tính năng nổi bật của Microsoft SQL Server bao gồm:

* **Hỗ trợ đa nền tảng:** SQL Server có khả năng chạy trên nhiều môi trường khác nhau như Windows, Linux, Docker, giúp đáp ứng linh hoạt các yêu cầu triển khai thực tế của doanh nghiệp.
* **Quản lý dữ liệu lớn:** SQL Server tích hợp các công cụ mạnh mẽ để quản lý dữ liệu lớn, hỗ trợ lưu trữ, phân tích và xử lý dữ liệu hiệu quả. Ngoài ra, còn tích hợp các tính năng về trí tuệ doanh nghiệp (Business Intelligence – BI) như phân tích, báo cáo và trực quan hóa dữ liệu.
* **Bảo mật toàn diện:** SQL Server được trang bị hệ thống bảo mật nhiều lớp, bao gồm mã hóa dữ liệu, kiểm soát truy cập, giám sát hoạt động, giúp đảm bảo an toàn tối đa cho dữ liệu cá nhân cũng như dữ liệu nhạy cảm của tổ chức.
* **Giao diện quản lý trực quan:** Thông qua công cụ SQL Server Management Studio (SSMS), người dùng có thể dễ dàng quản lý, thiết kế và kiểm thử cơ sở dữ liệu với giao diện thân thiện, trực quan, giảm thiểu sai sót và tăng hiệu quả vận hành.

*🔹 Ứng dụng trong đề tài*

* Trong đề tài **“Quản lý đặt vé chuyến bay”**, Microsoft SQL Server đóng vai trò là nền tảng lưu trữ và quản lý toàn bộ dữ liệu của hệ thống, bao gồm dữ liệu người dùng, chuyến bay, lịch trình, giao dịch đặt vé và các hoạt động quản trị liên quan. Việc sử dụng SQL Server giúp đảm bảo hệ thống có thể vận hành ổn định khi lượng dữ liệu phát sinh lớn, đồng thời đáp ứng được các yêu cầu kiểm tra, thống kê và báo cáo số liệu nhanh chóng, chính xác.
* Cụ thể, SQL Server thường được triển khai trong các hệ thống đòi hỏi hiệu năng cao và độ tin cậy lớn, điển hình như: hệ thống ngân hàng, thương mại điện tử, ứng dụng quản trị doanh nghiệp (ERP, CRM), cũng như các dịch vụ đặt vé trực tuyến với nhu cầu xử lý giao dịch lớn và liên tục.

*🔹 Lý do chọn*

* **Độ ổn định và bảo mật cao:** SQL Server cung cấp các giải pháp bảo mật toàn diện với nhiều lớp bảo vệ như kiểm soát truy cập, mã hóa dữ liệu và giám sát hoạt động, từ đó đảm bảo an toàn thông tin cá nhân và dữ liệu giao dịch của khách hàng.
* **Khả năng xử lý dữ liệu lớn:** Hệ thống có thể lưu trữ, xử lý và truy xuất đồng thời một lượng lớn bản ghi liên quan đến người dùng, chuyến bay, cũng như các giao dịch đặt vé, đáp ứng tốt yêu cầu của một hệ thống quản lý vé quy mô thực tế.
* **Công cụ quản lý trực quan:** SQL Server Management Studio (SSMS) giúp cho việc thiết kế, xây dựng, kiểm thử và vận hành cơ sở dữ liệu trở nên thuận tiện, hiệu quả, phù hợp với cả người mới và các quản trị viên hệ thống chuyên nghiệp.
* **Khả năng mở rộng:** Khi hệ thống phát triển hoặc số lượng người dùng tăng cao, SQL Server dễ dàng mở rộng về quy mô dữ liệu cũng như hiệu năng xử lý, đảm bảo đáp ứng nhu cầu mở rộng trong tương lai mà không ảnh hưởng đến hiệu suất hoặc tính ổn định.
* **Tính tương thích cao:** SQL Server tương thích tốt với các ngôn ngữ lập trình phổ biến như .NET, Java, Python… cũng như các công cụ báo cáo, phân tích và ứng dụng bên ngoài, hỗ trợ tích hợp linh hoạt với các hệ thống khác nếu cần thiết.

**CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM**

**3.1. Phân tích hệ thống**

**3.1.1 Khảo sát**

1. **Thực trạng hiện nay**

* Hiện nay, nhu cầu đi lại bằng đường hàng không ngày càng phổ biến, đặc biệt là giữa các thành phố lớn tại Việt Nam như Hà Nội, TP. Hồ Chí Minh, Đà Nẵng, Phú Quốc,...
* Các hãng hàng không triển khai nhiều hình thức khuyến mãi và lịch bay phong phú, kéo theo sự gia tăng của việc tra cứu và đặt vé máy bay trực tuyến.
* Tuy nhiên, không phải hệ thống nào cũng tối ưu trải nghiệm người dùng. Một số vấn đề phổ biến:
  + Giao diện phức tạp, khó sử dụng với người dùng không am hiểu công nghệ.
  + Quá trình đặt vé có nhiều bước thủ công, gây mất thời gian.
  + Thiếu hệ thống quản lý đơn giản dành cho admin (quản lý chuyến bay, chỉnh sửa, xóa,...).
  + Việc cập nhật và đồng bộ thông tin chuyến bay chưa hiệu quả, ảnh hưởng đến độ chính xác khi người dùng tra cứu.

1. **Nhu cầu thực tế**

Hệ thống cần đáp ứng các yêu cầu:

**Dành cho người dùng:**

* Giao diện đơn giản, dễ thao tác.
* Có thể đăng ký tài khoản, đăng nhập nhanh chóng.
* Cho phép **tra cứu chuyến bay theo ngày, điểm đi, điểm đến**.
* Thao tác đặt vé, chọn số lượng, thanh toán linh hoạt.
* Theo dõi được lịch sử đặt vé và thông tin cá nhân.

**Dành cho Admin:**

* Có thể đăng nhập bằng tài khoản quản trị.
* Quản lý đầy đủ danh sách chuyến bay: **thêm mới, chỉnh sửa, xóa**.
* Cập nhật trực tiếp ảnh, giá vé, thời gian bay trên hệ thống

1. **Mục tiêu khảo sát**

* Xác định các chức năng cần thiết cho một ứng dụng đặt vé máy bay đơn giản và hiệu quả.
* Đảm bảo hệ thống đáp ứng được **2 nhóm đối tượng** chính:
  + **Người dùng phổ thông** (đặt vé, tra cứu).
  + **Quản trị viên (Admin)** (quản lý chuyến bay).
* Tối ưu giao diện và quy trình xử lý dữ liệu bằng SQLite để vận hành tốt trên môi trường Android.

|  |  |
| --- | --- |
| **Tác nhân** | **Vai trò chính** |
| Người dùng (User) | Đặt vé, tra cứu chuyến bay, xem lịch sử đặt vé, thêm và quản lý chuyến bay yêu thích, chỉnh sửa hồ sơ cá nhân |
| Quản trị viên (Admin) | Quản lý dữ liệu chuyến bay |

**3.1.2 Tác nhân**

**Bảng 3.1.2 – Bảng tác nhân**

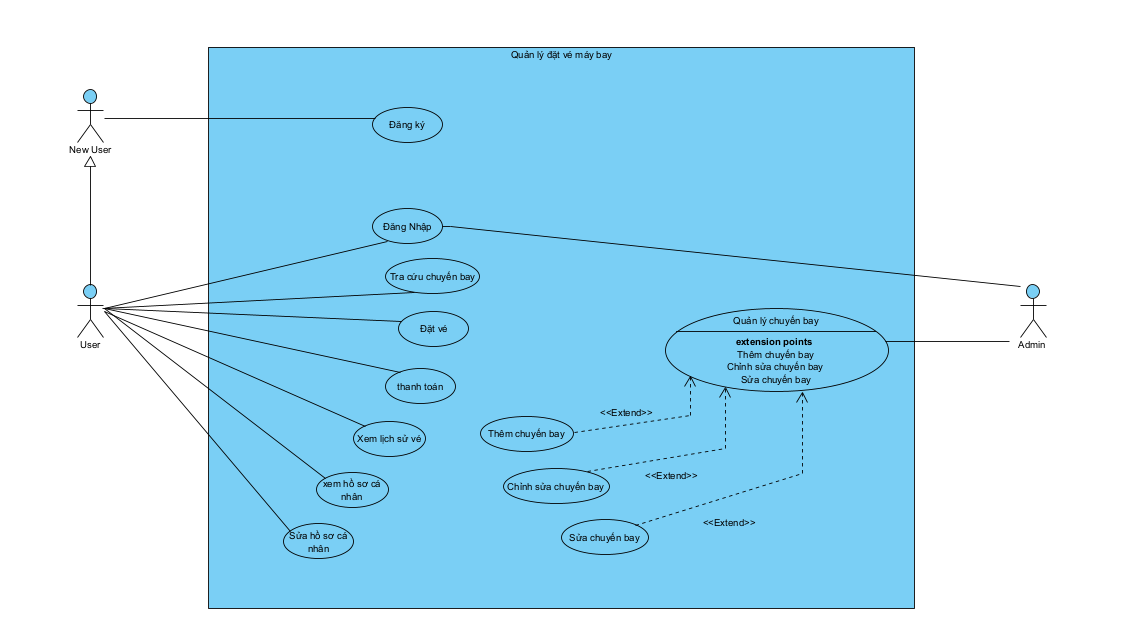
**3.1.3 Usecase và đặc tả**

1. **Xác định Usecase**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Usecase** | **Mô tả** | **Tác nhân** |
| 1 | Đăng nhập | Người dùng cung cấp tên đăng nhập và mật khẩu. Hệ thống xác thực thông tin và cho phép truy cập vào hệ thống nếu thông tin là hợp lệ. | Người dùng, Quản trị viên |
| 2 | Đăng ký | Người dùng cung cấp các thông tin đăng ký như họ tên, ngày sinh, email, số điện thoại, tên đăng nhập và mật khẩu. Hệ thống kiểm tra tính hợp lệ và tạo tài khoản mới. | Người dùng |
| 3 | Đăng xuất | Người dùng có thể đăng xuất khỏi hệ thống và quay lại màn hình đăng nhập. | Người dùng, Quản trị viên |
| 4 | Tra cứu chuyến bay | Người dùng nhập thông tin cần tìm kiếm như điểm đi, điểm đến, ngày đi và ngày về. Hệ thống lọc dữ liệu và hiển thị danh sách các chuyến bay phù hợp. | Người dùng |
| 5 | Đặt vé | Người dùng chọn chuyến bay, sau đó chọn số lượng vé và loại, hệ thống xác nhận đủ thông tin và tính tổng tiền cần thanh toán. | Người dùng |
| 6 | Thanh toán | Người dùng chọn hình thức thanh toán, nhập thông tin thanh toán (ngân hàng, số tài khoản, nội dung chuyển tiền) và xác nhận thanh toán. Hệ thống kiểm tra thông tin, lưu vào CSDL và hiện hoá đơn. | Người dùng |
| 7 | Xem lịch sử đặt vé | Người dùng được chuyển đến giao diện hiển thị danh sách vé đã đặt gần đây. Mỗi vé bao gồm thông tin điểm đi, điểm đến, ngày đi, ngày về, số lượng vé, hình thức thanh toán và tổng tiền. | Người dùng |
| 8 | Quản lý hồ sơ cá nhân | Người dùng, admin đã đăng nhập, muốn kiểm tra các thông tin cá nhân và thay đổi thông tin cá nhân của mình. Hệ thống truy vấn thông tin từ cơ sở dữ liệu và hiển thị đầy đủ các trường thông tin liên quan. | Người dùng,  Quản trị viên |
| 9 | Quản lý chuyến bay | **N**gười có toàn quyền thực hiện các thao tác quản trị trên hệ thống. | Quản trị viên |
| 10 | Chuyến bay yêu thích | **Người dùng có thể đánh dấu chuyến bay yêu thích, quản lý danh sách chuyến bay yêu thích** | Người dùng |

**Bảng 3.1.3.a – Bảng Usecase tổng quát**

* **Sơ đồ tổng quát – Usecase diagram**



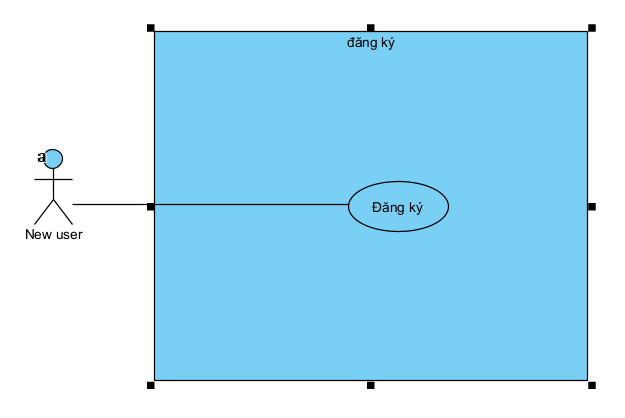
**Hình 3.1.3.a – Usecase diagram**

1. **Đặc tả Usecase**
2. **Usecase đăng ký**

* **Cấu trúc bảng**

|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 1 | |
| **Usecase** | Đăng ký | |
| **Ngữ cảnh** | Chức năng cho phép người dùng mới tạo tài khoản để sử dụng ứng dụng đặt vé máy bay. | |
| **Sự kiện kích hoạt** | Người dùng nhấn vào chức năng “Đăng ký” trên giao diện ứng dụng. | |
| **Mô tả** | Người dùng cung cấp các thông tin đăng ký như họ tên, ngày sinh, email, số điện thoại, tên đăng nhập và mật khẩu. Hệ thống kiểm tra tính hợp lệ và tạo tài khoản mới. | |
| **Tác nhân** | Người dùng (User) | |
| **Điều kiện tiên quyết** | * Người dùng chưa có tài khoản trên hệ thống. * Truy cập được vào giao diện đăng ký. | |
| **Điều kiện sau** | * Tài khoản mới được tạo thành công và chuyển sang màn hình đăng nhập. * Người dùng có thể dùng tài khoản vừa tạo để đăng nhập vào hệ thống. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| 1. Truy cập vào hệ thống.  2. Chọn chức năng “Đăng Ký trên giao diện đăng nhập”.  4. Nhập thông tin đăng ký (họ tên, ngày sinh, email, số điện thoại, tên đăng nhập, mật khẩu).  5. Nhấn nút “Đăng ký”.  10. Nhận thông báo và chuyển sang giao diện đăng nhập. | 3. Hiển thị giao diện đăng ký.  6. Kiểm tra tính hợp lệ của dữ liệu.  - Nếu thiếu hoặc sai định dạng, hiển thị thông báo lỗi và quay lại bước 4.  7. Kiểm tra tên đăng nhập đã tồn tại chưa.  - Nếu tên đăng nhập đã tồn tại, hiển thị cảnh báo và quay lại bước 4.  8. Nếu hợp lệ và chưa tồn tại, tiến hành tạo tài khoản mới.  9. Thông báo đăng ký thành công và chuyển sang màn hình đăng nhập. |
| **Ngoại lệ** | Không có. | |

**Bảng 3.1.3.b.1 – Bảng cấu trúc Usecase đăng ký**

* **Hình ảnh**

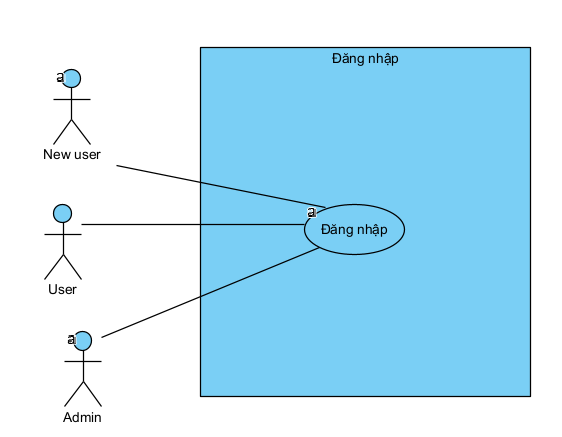
**Hình 3.1.3.b.1 - Usecase đăng ký**

1. **Usecase đăng nhập**

* **Cấu trúc bảng**

|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 2 | |
| **Usecase** | Đăng nhập | |
| **Ngữ cảnh** | |  | | --- | | Chức năng cho phép người dùng đã có tài khoản truy cập vào hệ thống ứng dụng đặt vé máy bay. |  |  | | --- | |  | | |
| **Sự kiện kích hoạt** | Người dùng nhấn vào chức năng “Đăng nhập” trên giao diện ứng dụng. | |
| **Mô tả** | |  | | --- | |  |  |  | | --- | | Người dùng cung cấp tên đăng nhập và mật khẩu. Hệ thống kiểm tra tính hợp lệ của thông tin, xác thực tài khoản và cho phép truy cập nếu thông tin hợp lệ. | | |
| **Tác nhân** | Người dùng, Quản trị viên | |
| **Điều kiện tiên quyết** | |  | | --- | |  |  |  | | --- | | * Người dùng đã có tài khoản trên hệ thống. * Truy cập được vào giao diện đăng nhập. | |  | | |
| **Điều kiện sau** | * Đăng nhập thành công và chuyển đến giao diện chính của ứng dụng. * Người dùng có thể sử dụng các chức năng của hệ thống sau khi đăng nhập. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| 1. Truy cập vào ứng dụng.  2. Chọn chức năng “Đăng nhập trên giao diện”.  4. Nhập thông tin đăng nhập (tên đăng nhập và mật khẩu).  5. Nhấn nút “Đăng nhập”.  10. Bắt đầu sử dụng các chức năng ứng dụng. | 3. Hiển thị giao diện nhập.  6. Kiểm tra tính hợp lệ của dữ liệu đầu vào (không để trống, đúng định dạng).  6.1 Nếu thiếu hoặc sai định dạng, hiển thị thông báo lỗi và quay lại bước 4.  7. Kiểm tra thông tin đăng nhập với dữ liệu trong hệ thống.  - Nếu tên đăng nhập hoặc mật khẩu sai, hiển thị thông báo lỗi và quay lại bước 4.  8. Nếu thông tin là hợp lệ, xác thực thành công và chuyển đến giao diện chính.  9. Thông báo đăng nhập thành công. |
| **Ngoại lệ** | Không có. | |

**Bảng 3.1.3.b.2 – Bảng cấu trúc Usecase đăng nhập**

* **Hình ảnh**

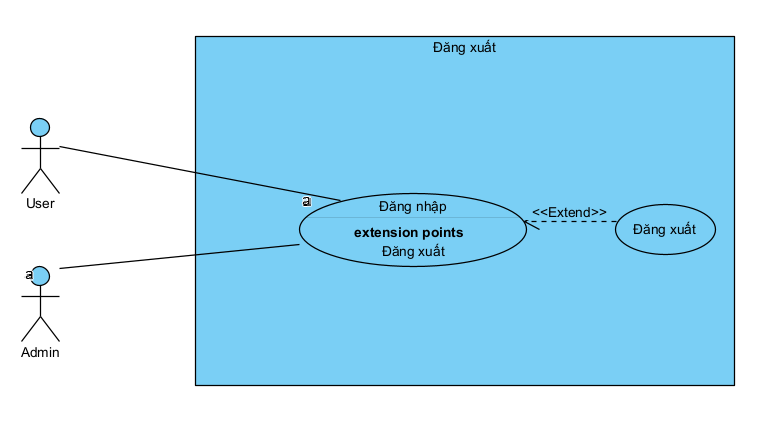
**Hình 3.1.3.b.2 - Usecase đăng nhập**

1. **Usecase đăng xuất**

* **Cấu trúc bảng**

|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 3 | |
| **Usecase** | Đăng xuất | |
| **Ngữ cảnh** | Người dùng đã đăng nhập vào hệ thống và muốn kết thúc phiên của họ. | |
| **Sự kiện kích hoạt** | Nhấn chức năng “Đăng xuất” trên giao diện thông tin cá nhân | |
| **Mô tả** | |  | | --- | | Hệ thống thực hiện đăng xuất tài khoản, kết thúc phiên làm việc hiện tại, đồng thời chuyển người dùng về giao diện đăng nhập để đảm bảo an toàn thông tin. |  |  | | --- | |  | | |
| **Tác nhân** | Người dùng (User), Quản trị viên | |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập vào hệ thống. * Đã chuyển đến giao diện thông tin cá nhân. | |
| **Điều kiện sau** | * Người dùng đã đăng xuất hoàn toàn khỏi hệ thống. * Để truy cập lại, người dung cần đăng nhập lại. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| 1. Đăng nhập vào hệ thống.  2. Chuyển sang giao diện thông tin cá nhân (profile).   |  | | --- | | 3. Nhấn nút “Đăng xuất” trên giao diện cá nhân. |  |  | | --- | |  |  |  | | --- | |  |  |  | | --- | |  | | 5. Xác nhận thao tác đăng xuất (nếu có). | |  |  | | --- | |  | | |  | | --- | | 4. Hiển thị hộp thoại xác nhận đăng xuất. |  |  | | --- | |  |  |  | | --- | |  |  |  |  |  | | --- | --- | --- | | 6. Thực hiện đăng xuất: xóa phiên đăng nhập hiện tại, làm sạch dữ liệu tạm thời.   |  | | --- | |  |  |  | | --- | | 7. Chuyển người dùng về giao diện đăng nhập. | | |
| **Ngoại lệ** | Không có. | |

**Bảng 3.1.3.b.3 – Bảng cấu trúc Usecase đăng xuất**

* **Hình ảnh**

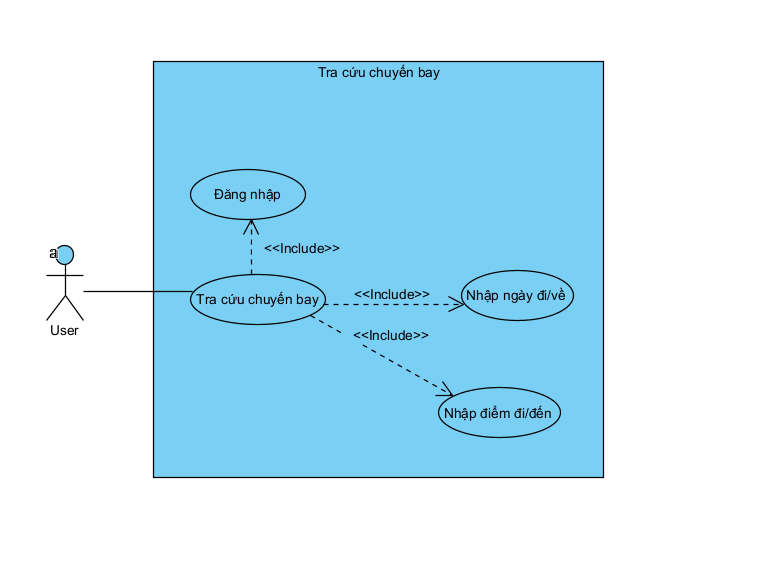
**Hình 3.1.3.b.3 - Usecase đăng xuất**

1. **Usecase Tra cứu chuyến bay**

* **Cấu trúc bảng**

|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 4 | |
| **Usecase** | Tra cứu chuyến bay | |
| **Ngữ cảnh** | Chức năng cho phép người dùng tìm kiếm các chuyến bay phù hợp theo điểm đi, điểm đến và ngày đi và ngày về. | |
| **Sự kiện kích hoạt** | Người dùng chọn chức năng “Tra cứu chuyến bay” từ màn hình chính. | |
| **Mô tả** | Người dùng nhập thông tin cần tìm kiếm như điểm đi, điểm đến, ngày đi, ngày về. Hệ thống lọc dữ liệu và hiển thị danh sách các chuyến bay phù hợp, cho ơheps người dung xem chi tiết và đặt vé. | |
| **Tác nhân** | Người dùng (User) | |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập vào hệ thống. * Cơ sở dữ liệu đã có chuyến bay được thêm từ Admin. | |
| **Điều kiện sau** | * Hiển thị danh sách chuyến bay phù hợp theo điều kiện lọc. * Cho phép chọn chuyến bay để xem chi tiết hoặc đặt vé. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| |  | | --- | | 1. Đăng nhập vào hệ thống. | |  |  |  | | --- | |  | | 2. Chọn chức năng “Tra cứu chuyến bay” trên màn hình chính. | |  |  | | --- | |  |   4. Nhập thông tin tra cứu và nhấn nút “Tìm kiếm”.  8. Chọn chuyến bay mong muốn từ danh sách đang hiển thị. | |  | | --- | | 3. Hiển thị giao diện nhập thông tin tìm kiếm (điểm đi, điểm đến, ngày đi, ngày về). |  |  | | --- | |  |   5. Kiểm tra tính hợp lệ của dữ liệu đầu vào (không rỗng, đúng định dạng).  - Nếu sai, thông báo lỗi và quay lại bước 4.  6. Truy vấn cơ sở dữ liệu với thông tin đã nhập.  7. Hiển thị danh sách các chuyến bay phù hợp với điều kiện lọc.  9. Điều hướng sang màn hình chi tiết chuyến bay để tiến hành đặt vé. |
| **Ngoại lệ** |  Nếu không tìm thấy chuyến bay phù hợp với tiêu chí tra cứu, hệ thống hiển thị thông báo: **“Không tìm thấy chuyến bay phù hợp với điều kiện tìm kiếm.”**   Nếu xảy ra lỗi kết nối hoặc truy vấn cơ sở dữ liệu, hệ thống thông báo lỗi kỹ thuật và hướng dẫn người dùng thử lại sau. | |

**Bảng 3.1.3.b.4 – Bảng cấu trúc Usecase tra cứu chuyến bay**

* **Hình ảnh**

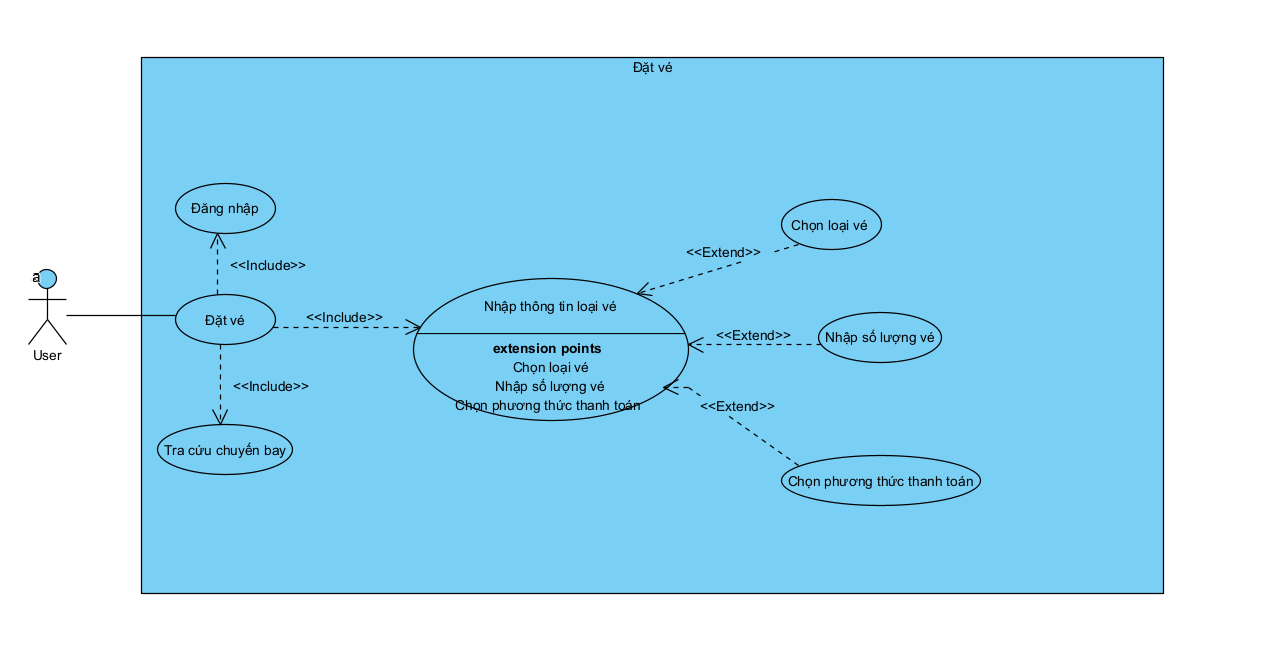
**Hình 3.1.3.b.4 - Usecase tra cứu chuyến bay**

1. **Usecase đặt vé**

* **Cấu trúc bảng**

|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 5 | |
| **Usecase** | Đặt vé | |
| **Ngữ cảnh** | Chức năng cho phép người dùng chọn chuyến bay phù hợp từ danh sách đã tra cứu và tiến hành đặt chỗ | |
| **Sự kiện kích hoạt** | Người dùng chọn chuyến bay và nhấn nút “Đặt vé”. | |
| **Mô tả** | Người dùng chọn số lượng vé và loại vé (ví dụ: phổ thông, phổ thông đặc biệt, thương gia), hệ thống xác nhận thông tin, tính tổng tiền và lưu thông tin đặt vé tạm thời, sau đó điều hướng sang bước thanh toán. | |
| **Tác nhân** | Người dùng (User) | |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập. * Đã tra cứu và chọn được chuyến bay phù hợp. | |
| **Điều kiện sau** | * Thông tin vé được lưu tạm (chưa thanh toán). * Người dùng được điều hướng sang bước thanh toán. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| |  |  | | --- | --- | | |  | | --- | |  | |   1. Đăng nhập vào hệ thống.  2. Chọn chức năng “Tra cứu chuyến bay”.  4. Chọn chuyến bay mong muốn.  6. Nhập thông tin đặt vé (số lượng vé, loại vé, phương thức thanh toán).   |  | | --- | | 7. Nhấn nút “Đặt vé”. |  |  | | --- | |  |  |  | | --- | | 12. Tiến hành thanh toán để hoàn tất đặt vé. |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | 3. Hiển thị danh sách chuyến bay theo điều kiện lọc. | |  |  | | --- | |  |   5. Hiển thị thông tin chi tiết chuyến bay và tuỳ chọn đặt vé.  8. Kiểm tra tính hợp lệ của dữ liệu (không rỗng, số hợp lệ, loại hợp lệ).  - Nếu sai: thông báo lỗi và yêu cầu nhập lại.  9. Tính tổng tiền (số lượng \* đơn giá vé \* thuế).   |  | | --- | |  |  |  | | --- | | 10. Lưu thông tin đặt vé tạm thời (chưa thanh toán) vào hệ thống. | | 11. Hiển thị tổng tiền và điều hướng sang giao diện thanh toán. |  |  | | --- | |  | |
| **Ngoại lệ** |  Nếu không có chuyến bay phù hợp với điều kiện tìm kiếm, hệ thống thông báo: **“Không tìm thấy chuyến bay phù hợp.”**   Nếu chọn thiếu thông tin, hệ thống sẽ hiển thị lỗi và yêu cầu chọn đầy đủ thông tin.   Nếu trong quá trình lưu tạm hoặc xử lý gặp lỗi hệ thống, thông báo lỗi kỹ thuật và hướng dẫn người dùng thử lại. | |

**Bảng 3.1.3.b.5 – Bảng cấu trúc Usecase đặt vé**

* **Hình ảnh**

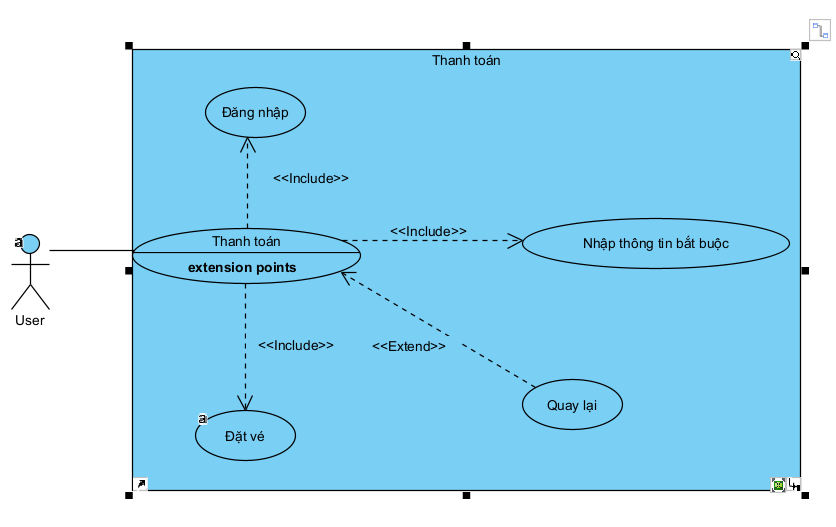
**Hình 3.1.3.b.5 - Usecase đặt vé**

1. **Usecase thanh toán**

* **Cấu trúc bảng**

|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 6 | |
| **Usecase** | Thanh toán | |
| **Ngữ cảnh** | Chức năng cho phép người dùng thanh toán cho vé đã đặt bằng cách chọn hình thức thanh toán (chuyển khoản hoặc quét thẻ) và cung cấp thông tin cần thiết. | |
| **Sự kiện kích hoạt** | Người dùng chọn chuyến bay và nhấn nút “Đặt vé”. | |
| **Mô tả** | Người dùng chọn hình thức thanh toán, nhập thông tin thanh toán (ngân hàng, số tài khoản, nội dung thanh toán) và xác nhận thanh toán. Hệ thống kiểm tra thông tin, lưu vào CSDL và trả kết quả. | |
| **Tác nhân** | Người dùng (User) | |
| **Điều kiện tiên quyết** | * Đã đăng nhập và hoàn tất bước đặt vé. * Chưa thực hiện thanh toán cho vé đó. | |
| **Điều kiện sau** | * Hệ thống lưu thông tin thanh toán vào CSDL. * Hóa đơn được tạo và hiển thị. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| |  |  | | --- | --- | | |  | | --- | |  | |  |  | | --- | | 1. Được điều hướng đến giao diện thanh toán.  3. Chọn hình thức thanh toán (chuyển khoản / quét thẻ). | | |  | | --- | |  | | |  |  | | --- | |  |   4. Nhập thông tin thanh toán: tên ngân hàng, số tài khoản, nội dung thanh toán.   |  | | --- | | 5. Nhấn nút “Xác nhận thanh toán”. |  |  | | --- | |  |   9. Hiển thị tổng tiền vừa thanh toán và hỏi có muốn xem hoá đơn hay không. | |  | | --- | | 2. Hiển thị thông tin cần thanh toán + giao diện chọn hình thức thanh toán. |  |  | | --- | |  |   6. Kiểm tra tính hợp lệ của thông tin thanh toán.  - Nếu sai → Hiển thị lỗi cụ thể và quay lại bước 4.  7. Nếu đúng → Lưu thông tin vào CSDL.  8. Tạo hóa đơn để hiển thị.  10. Nếu người dung muốn xem hoá đơn → hiển thị hoá đơn lên màn hình. |
| **Ngoại lệ** | Nếu thông tin thanh toán không hợp lệ hoặc giao dịch thất bại, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng thử lại. | |

**Bảng 3.1.3.b.6 – Bảng cấu trúc Usecase thanh toán**

* **Hình ảnh**

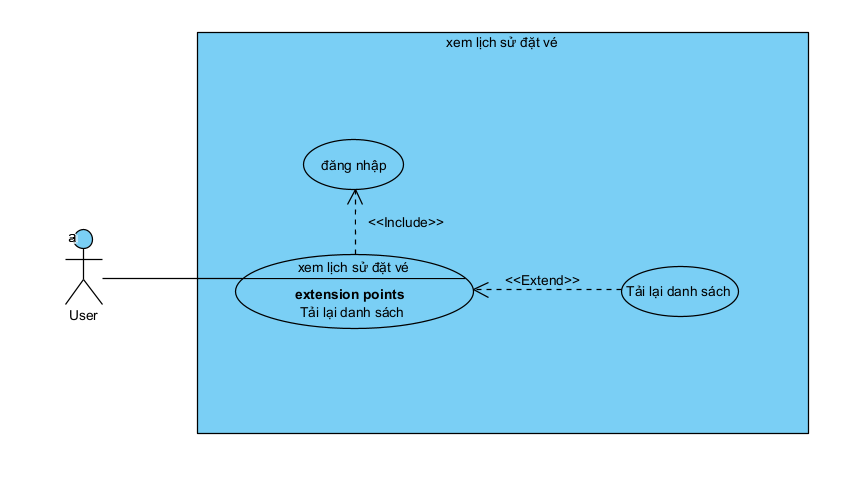
**Hình 3.1.3.b.6 - Usecase thanh toán**

1. **Usecase lịch sử đặt vé**

* **Cấu trúc bảng**

|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 7 | |
| **Usecase** | Xem lịch sử đặt vé | |
| **Ngữ cảnh** | Chức năng cho phép người dùng đã đăng nhập xem lại các vé máy bay mà họ đã từng đặt trước đó. | |
| **Sự kiện kích hoạt** | Người dùng nhấn vào biểu tượng “ticket” (vé) trên thanh điều hướng. | |
| **Mô tả** | Người dùng được chuyển đến giao diện hiển thị danh sách vé đã đặt gần đây. Mỗi vé bao gồm thông tin chi tiết như điểm đi, điểm đến, ngày đi, ngày về, số lượng vé, hình thức thanh toán và tổng tiền. | |
| **Tác nhân** | Người dùng (User) | |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập. * Hệ thống có ít nhất một bản ghi đặt vé của người dùng. | |
| **Điều kiện sau** | * Danh sách lịch sử đặt vé được hiển thị đầy đủ. * Mỗi vé đều hiển thị thông tin chi tiết ngay tại danh sách, không cần nhấn vào để xem chi tiết riêng. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| 1. Nhấn vào biểu tượng “Ticket” trong thanh điều hướng. | 2. Điều hướng đến giao diện lịch sử đặt vé.  3. Truy vấn dữ liệu đặt vé liên quan đến người dùng trong cơ sở dữ liệu.  4. Hiển thị danh sách các vé đã đặt (kèm trạng thái thanh toán, tổng tiền). |
| **Ngoại lệ** | **Nếu không tìm thấy lịch sử đặt vé, hệ thống hiển thị thông báo “Không có dữ liệu”.** | |

**Bảng 3.1.3.b.7 – Bảng cấu trúc Usecase lịch sử đặt vé**

* **Hình ảnh**

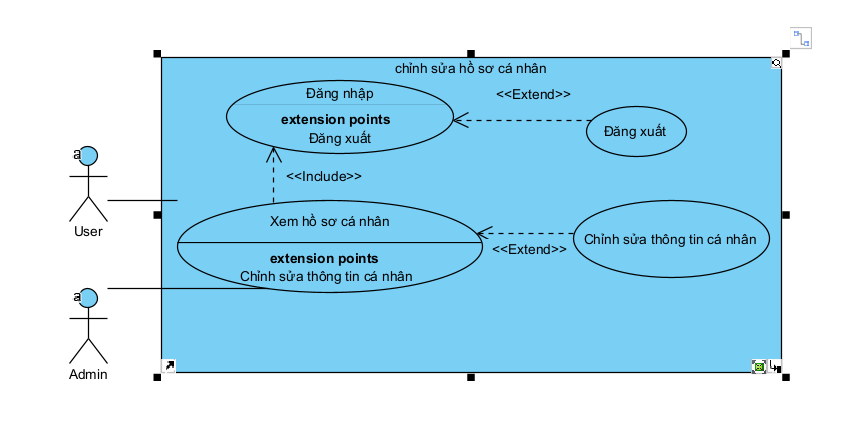
**Hình 3.1.3.b.7 - Usecase lịch sử đặt vé**

1. **Usecase quản lý hồ sơ cá nhân**

* **Cấu trúc bảng**

|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 8 | |
| **Usecase** | Quản lý hồ sơ cá nhân | |
| **Ngữ cảnh** | |  | | --- | |  |   Chức năng cho phép người dùng đã đăng nhập xem và cập nhật thông tin cá nhân như: họ tên, ngày sinh, số điện thoại, email, mật khẩu (ẩn) và số lượng vé đã đặt. | |
| **Sự kiện kích hoạt** | Người dùng nhấn vào nút **“Profile”** hoặc biểu tượng người dùng trong thanh điều hướng. | |
| **Mô tả** | Người dùng, admin đã đăng nhập, muốn kiểm tra các thông tin cá nhân và thay đổi thông tin cá nhân của mình. Hệ thống truy vấn thông tin từ cơ sở dữ liệu và hiển thị đầy đủ các trường thông tin liên quan. | |
| **Tác nhân** | Người dùng (User), Admin | |
| **Điều kiện tiên quyết** | • Người dùng đã đăng nhập thành công.  • Đang ở trong giao diện chính của ứng dụng. | |
| **Điều kiện sau** | • Hệ thống hiển thị đúng và đầy đủ thông tin cá nhân.  • Nếu có chỉnh sửa, thông tin được cập nhật thành công trong CSDL và hiển thị lại. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| 1. Truy cập vào giao diện chính sau khi đăng nhập    2. Nhấn nút “Profile” hoặc biểu tượng người dùng  5. Nhấn nút “Chỉnh sửa”  6. Nhập các thông tin mới  8. Nhấn nút “Lưu thay đổi”  11. Xem lại thông tin | 3. Kiểm tra trạng thái đăng nhập và truy vấn thông tin người dùng từ CSDL  4. Hiển thị thông tin hồ sơ cá nhân gồm: Họ tên, ngày sinh, số điện thoại, email, mật khẩu (ẩn).  7. Kiểm tra định dạng và tính hợp lệ của dữ liệu  9. Nếu hợp lệ → Cập nhật thông tin người dùng trong CSDL  10. Hiển thị thông báo “Cập nhật thành công” và quay lại giao diện hồ sơ |
| **Ngoại lệ** | * Nếu người dùng chưa đăng nhập → hệ thống chuyển hướng về màn hình đăng nhập. * Nếu dữ liệu chỉnh sửa không hợp lệ (trống, sai định dạng…) → hệ thống yêu cầu người dùng nhập lại. | |

**Bảng 3.1.3.b.8 – Bảng cấu trúc Usecase quản lý hồ sơ cá nhân**

* **Hình ảnh**

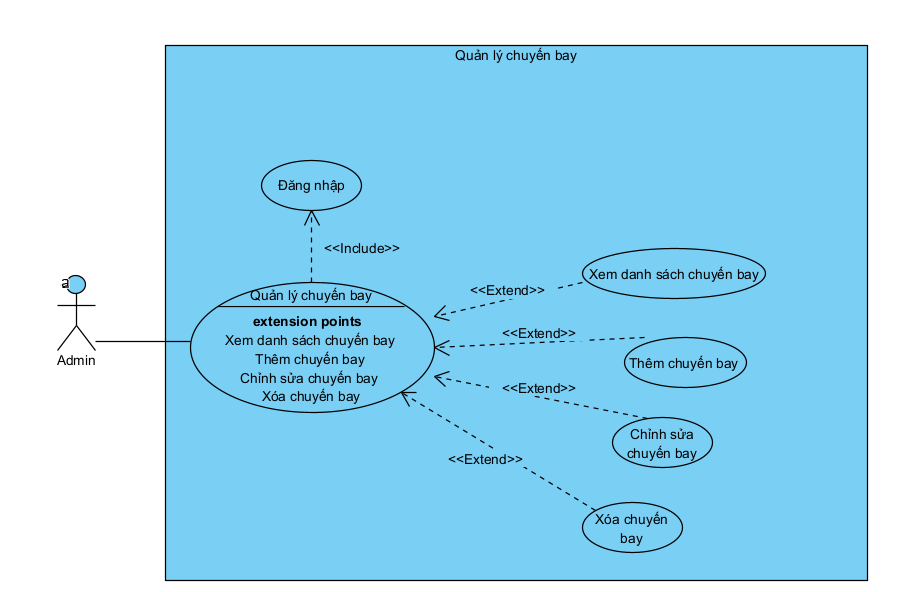
**Hình 3.1.3.b.8 - Usecase quản lý hồ sơ cá nhân**

1. **Usecase Quản lý chuyến bay**

* **Cấu trúc bảng**

|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 9 | |
| **Usecase** | Quản lý chuyến bay | |
| **Ngữ cảnh** | Chức năng cho phép Admin có quyền xem, thêm mới, chỉnh sửa hoặc xóa thông tin các chuyến bay trên hệ thống, bao gồm các thông tin như: điểm đi, điểm đến, ngày đi, ngày về, giá vé, hình ảnh minh họa. | |
| **Sự kiện kích hoạt** | Quản trị viên nhấn vào nút **“Chuyến bay”** từ màn hình thông tin cá nhân hoặc thanh điều hướng dưới. | |
| **Mô tả** | **Quản trị viên** – người có toàn quyền thực hiện các thao tác quản trị trên hệ thống. | |
| **Tác nhân** | |  | | --- | |  |  |  | | --- | | Quản trị viên | | |
| **Điều kiện tiên quyết** | * Quản trị viên đã đăng nhập vào hệ thống và có quyền truy cập giao diện quản lý. | |
| **Điều kiện sau** | * Danh sách chuyến bay được hiển thị đúng. * Các thao tác Thêm, Sửa, Xóa đều được cập nhật vào cơ sở dữ liệu thành công. * Giao diện người dùng được cập nhật lại để phản ánh thay đổi. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| 1. Truy cập vào giao diện thông tin cá nhân  2. Nhấn nút “Quản lý”  5. Nhấn “+” để thêm chuyến bay mới  6. Nhập thông tin (ảnh, điểm đi, điểm đến, ngày đi/về, giá vé)  7. Nhấn nút “Thêm chuyến bay” | 3. Kiểm tra quyền Admin  4. Hiển thị danh sách chuyến bay với thông tin & nút thao tác.  8. Kiểm tra tính hợp lệ của dữ liệu  9. Nếu hợp lệ → thêm mới vào CSDL, hiển thị thông báo thành công  10. Cập nhật danh sách chuyến bay |
| **Luồng mở rộng (tùy chọn)** | **Chỉnh sửa chuyến bay:**   * **Nhấn biểu tượng ✏ tại chuyến bay → Hiển thị form với dữ liệu cũ.** * **Nhập dữ liệu mới → Nhấn “Lưu thay đổi”.** * **Kiểm tra tính hợp lệ → Cập nhật trong CSDL → Hiển thị thông báo thành công.**   **Xóa chuyến bay:**   * **Nhấn biểu tượng 🗑 tại chuyến bay → Hiển thị hộp thoại xác nhận.** * **Nếu đồng ý → Xóa khỏi CSDL → Cập nhật lại danh sách.** | |
| **Ngoại lệ** | * **Nếu dữ liệu không hợp lệ → hiển thị thông báo lỗi và yêu cầu nhập lại.** * **Nếu không có quyền Admin → từ chối truy cập giao diện quản lý.** | |

**Bảng 3.1.3.b.9 – Bảng cấu trúc Usecase quản lý chuyến bay**

* **Hình ảnh**

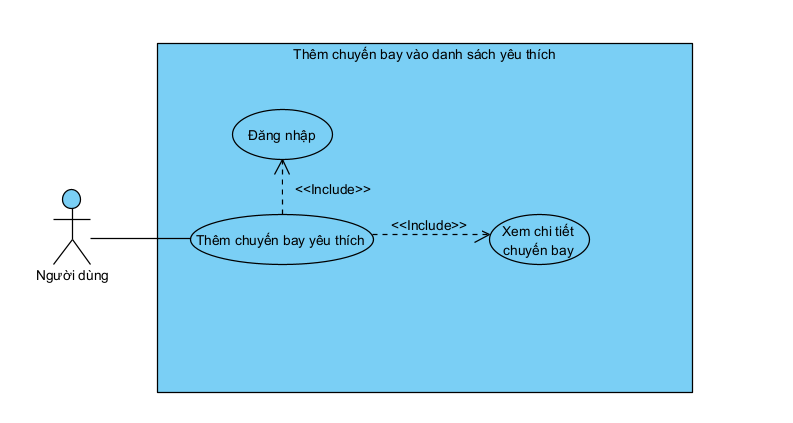
**Hình 3.1.3.b.9 - Usecase quản lý chuyến bay**

1. **Usecase chuyến bay yêu thích**

* **Cấu trúc bảng**

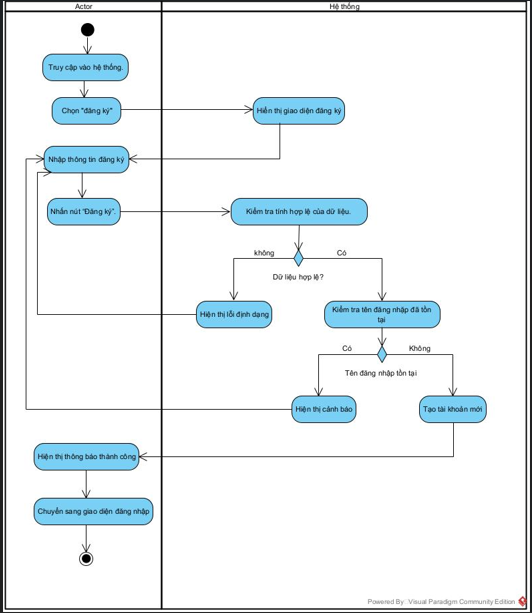
|  |  |  |
| --- | --- | --- |
| **ID** | Usecase 10 | |
| **Usecase** | Thêm chuyến bay vào danh sách yêu thích | |
| **Ngữ cảnh** | Chức năng cho phép người dùng thêm một chuyến bay cụ thể vào danh sách chuyến bay yêu thích để dễ dàng theo dõi hoặc đặt vé sau này. | |
| **Sự kiện kích hoạt** | Người dùng nhấn biểu tượng "trái tim" tại màn hình chi tiết chuyến bay. | |
| **Mô tả** | **Sau khi xem thông tin chi tiết một chuyến bay, người dùng có thể nhấn vào biểu tượng yêu thích. Hệ thống lưu thông tin chuyến bay vào danh sách yêu thích của người dùng và hiển thị thông báo xác nhận.** | |
| **Tác nhân** | |  | | --- | |  |  |  | | --- | | Người dùng (User) | | |
| **Điều kiện tiên quyết** | * Người dùng đã đăng nhập vào hệ thống. * Chuyến bay đã tồn tại trong hệ thống. | |
| **Điều kiện sau** | * Chuyến bay được lưu thành công vào danh sách yêu thích. * Hiển thị thông báo xác nhận và đề xuất chuyển đến màn hình danh sách yêu thích. | |
| **Luồng hoạt động** | **Tác nhân** | **Hệ thống** |
| 1. Đăng nhập hệ thống.    2. Chọn một chuyến bay cụ thể.  3. Nhấn biểu tượng yêu thích (trái tim).  7. Chọn “Có” hoặc “Không”. | 4. Kiểm tra xem chuyến bay đã có trong danh sách yêu thích chưa.  5. Nếu chưa có, lưu thông tin chuyến bay vào danh sách yêu thích trong cơ sở dữ liệu.  6. Hiển thị thông báo xác nhận “Đã thêm vào danh sách yêu thích. Bạn có muốn chuyển đến danh sách chuyến bay yêu thích không?”.  8. Nếu chọn “Có” → điều hướng đến màn hình danh sách yêu thích. Nếu “Không” → quay lại màn hình chi tiết. |
| **Ngoại lệ** | **Nếu chuyến bay đã tồn tại trong danh sách yêu thích → hiển thị thông báo “Chuyến bay này đã có trong danh sách yêu thích”.** | |

**Bảng 3.1.3.b.10 – Bảng cấu trúc Usecase chuyến bay yêu thích**

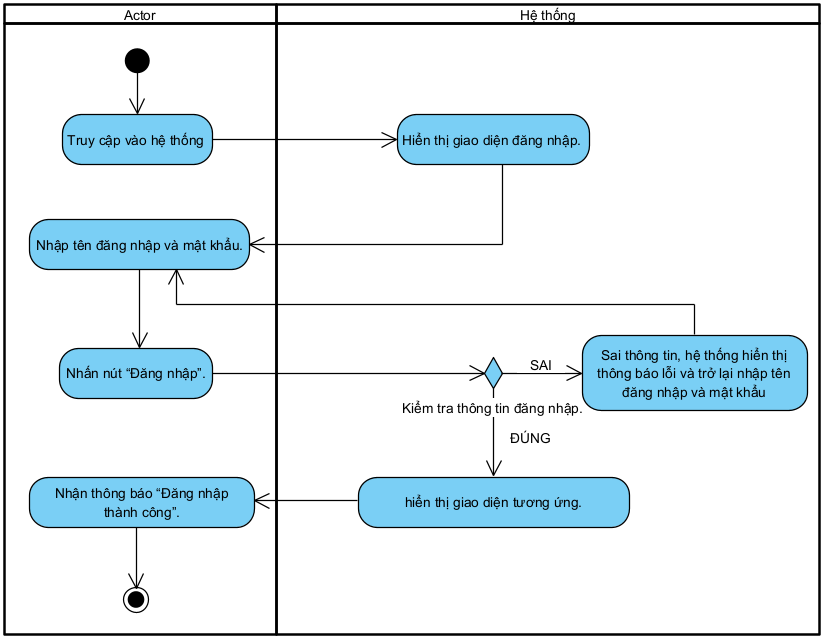
* **Hình ảnh**

**Hình 3.1.3.b.10 - Usecase chuyến bay yêu thích**

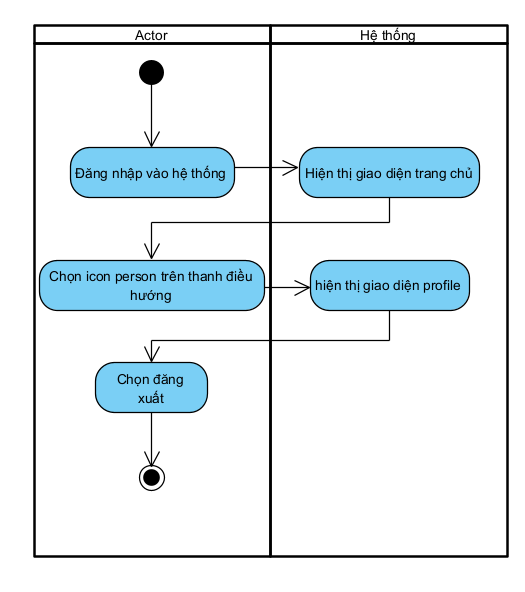
**3.1.4 Sơ đồ hoạt động**

1. **Sơ đồ đăng ký**

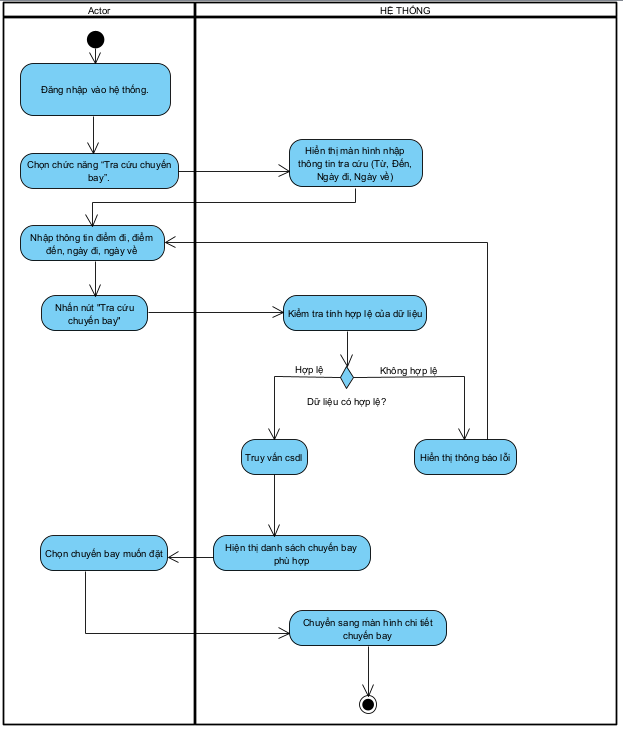
**Hình 3.1.4.a - Sơ đồ đăng ký**

1. **Sơ đồ đăng nhập**

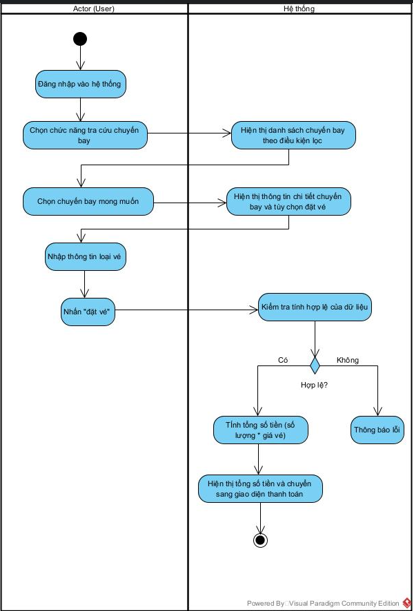
**Hình 3.1.4.b - Sơ đồ đăng nhập**

1. **Sơ đồ đăng xuất**

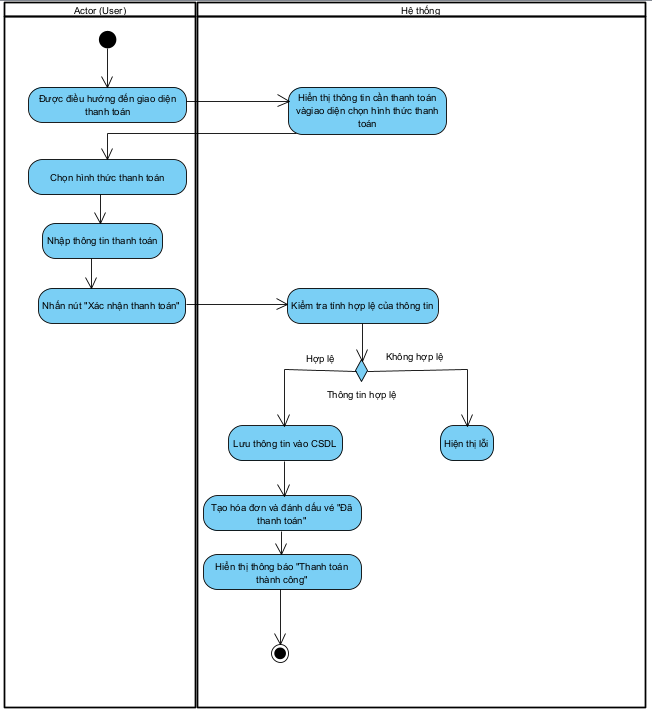
**Hình 3.1.4.c - Sơ đồ đăng xuất**

1. **Sơ đồ tra cứu chuyến bay**

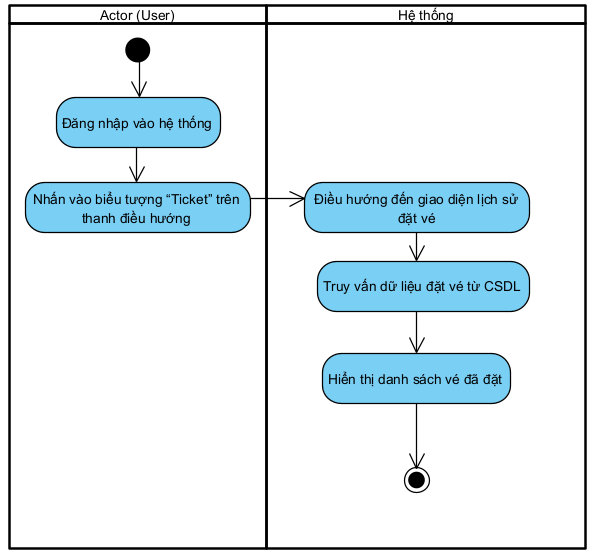
**Hình 3.1.4.d - Sơ đồ tra cứu chuyến bay**

1. **Sơ đồ đặt vé**

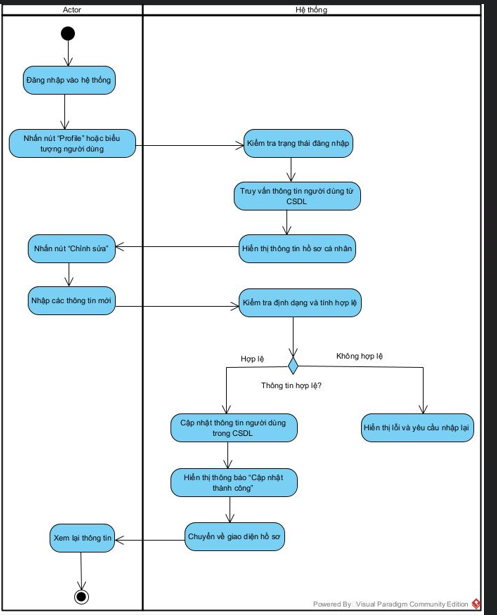
**Hình 3.1.5.e - Sơ đồ tra cứu chuyến bay**

1. **Sơ đồ thanh toán**

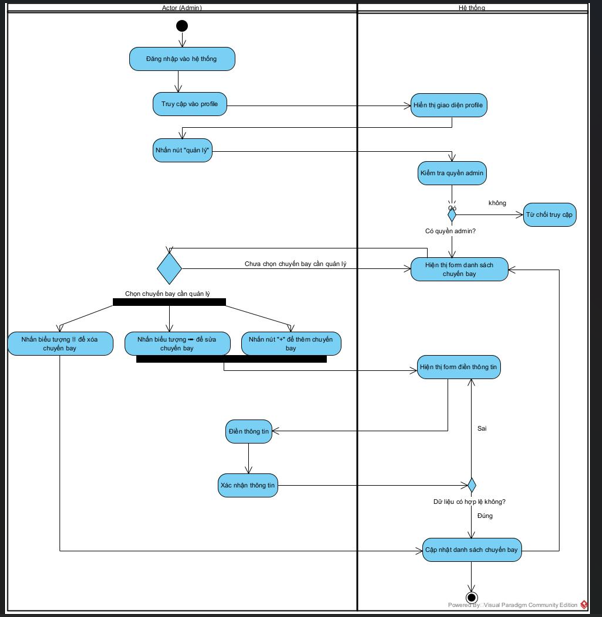
**Hình 3.1.4.f - Sơ đồ thanh toán**

1. **Sơ đồ lịch sử đặt vé**

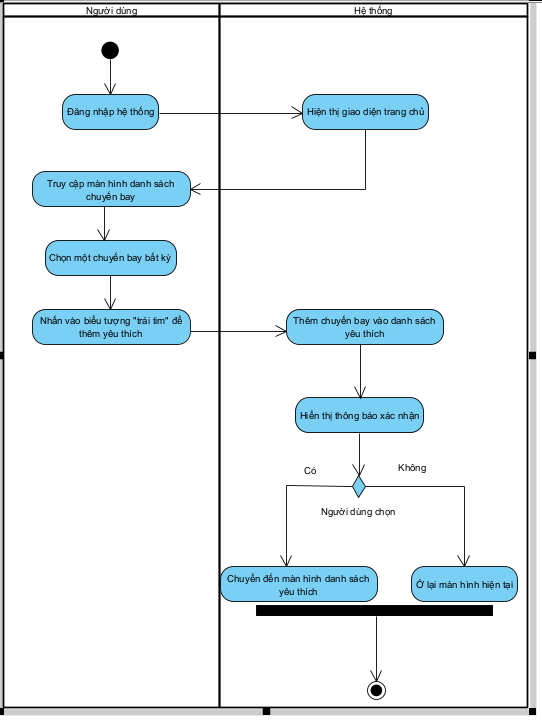
**Hình 3.1.4.g - Sơ đồ lịch sử đặt vé**

1. **Sơ đồ quản lý hồ sơ cá nhân**

**Hình 3.1.4.h - Sơ đồ quản lý hồ sơ cá nhân**

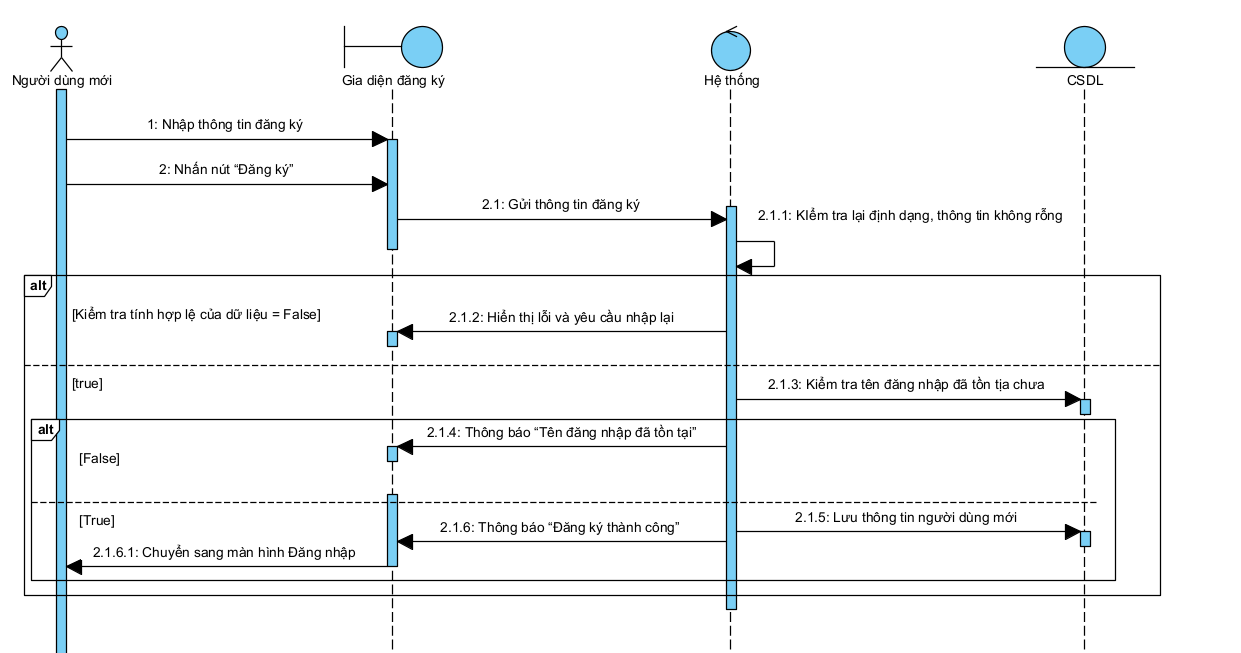
1. **Sơ đồ quản lý chuyến bay**

**Hình 3.1.4.i - Sơ đồ quản lý chuyến bay**

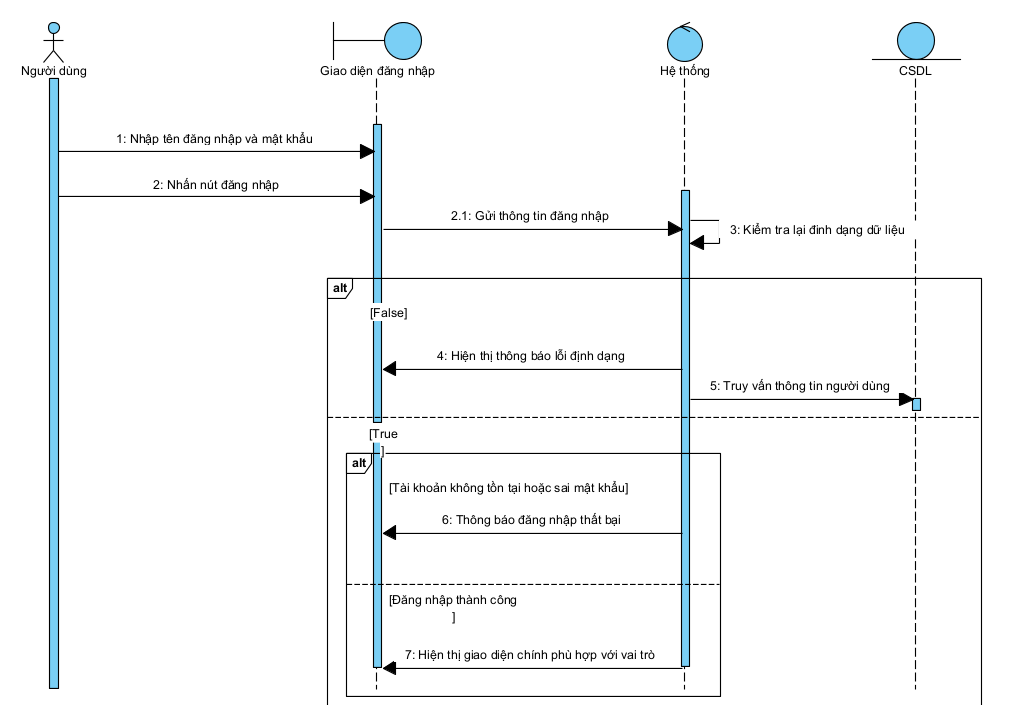
1. **Sơ đồ chuyến bay yêu thích**

**Hình 3.1.4.j - Sơ đồ chuyến bay yêu thích**

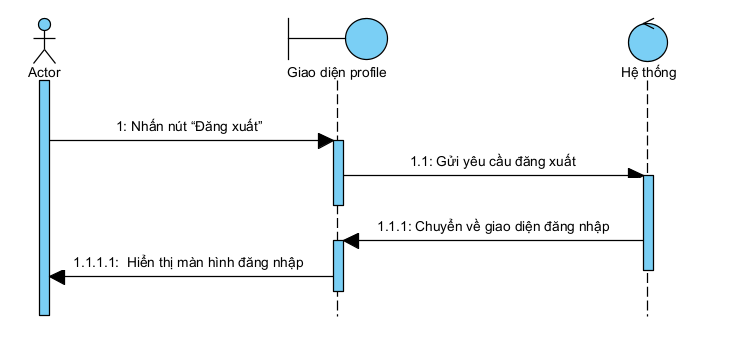
**3.1.5 Tuần tự hoạt động**

1. **Đăng ký**

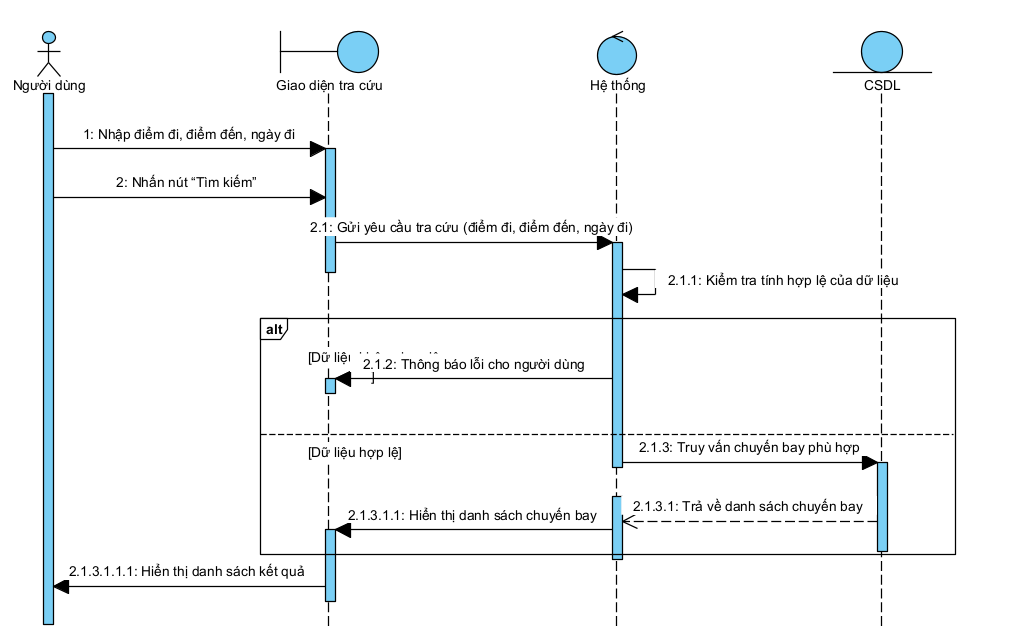
**Hình 3.1.5.a - Sơ đồ tuần tự đăng ký**

1. **Đăng nhập**

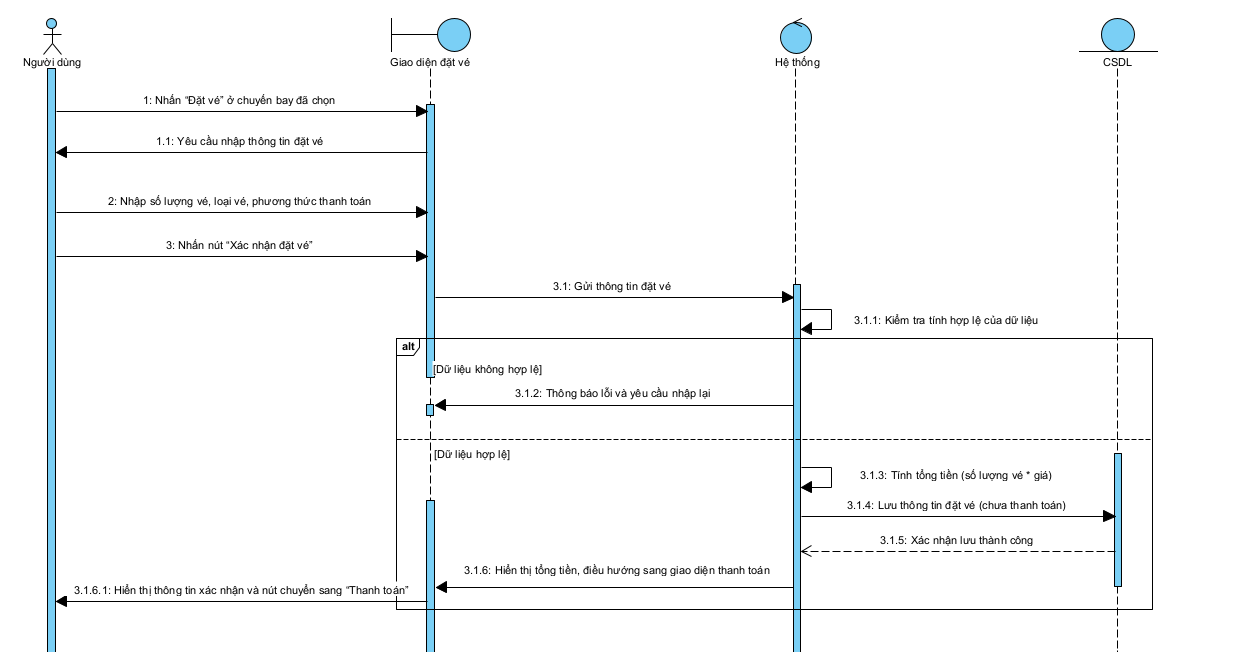
**Hình 3.1.5.b - Sơ đồ tuần tự đăng nhập**

1. **Đăng xuất**

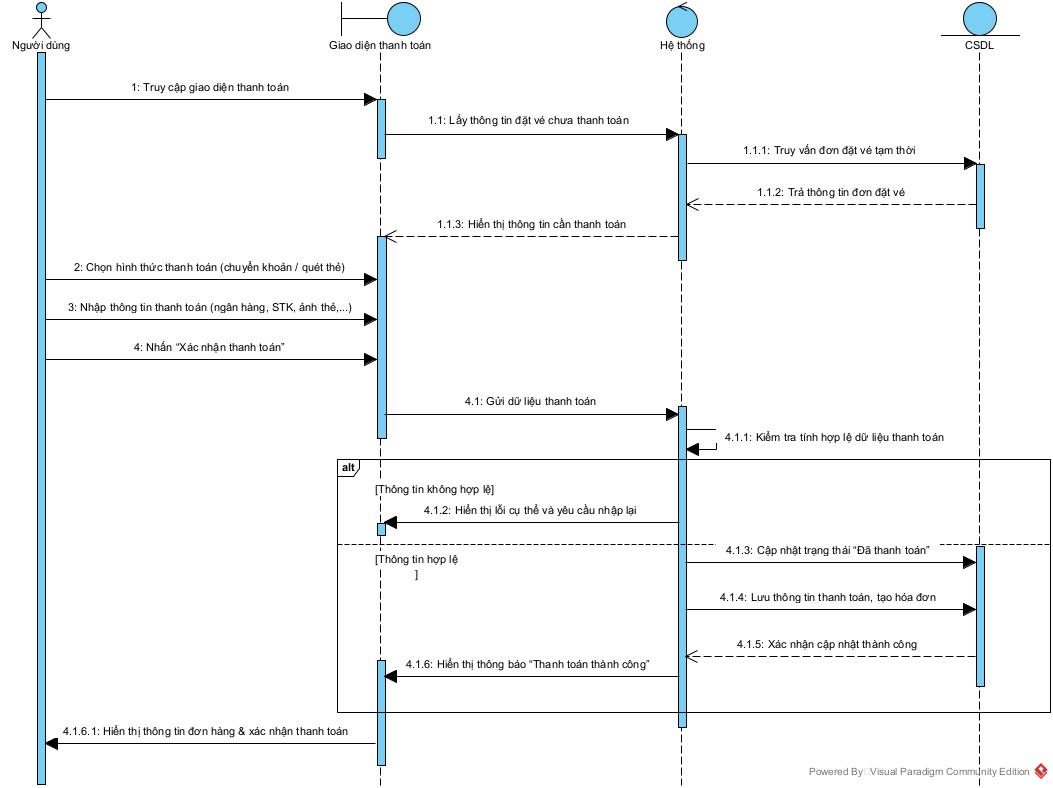
**Hình 3.1.5.c - Sơ đồ tuần tự đăng xuất**

1. **Tra cứu chuyến bay**

**Hình 3.1.5.d - Sơ đồ tuần tự đăng xuất**

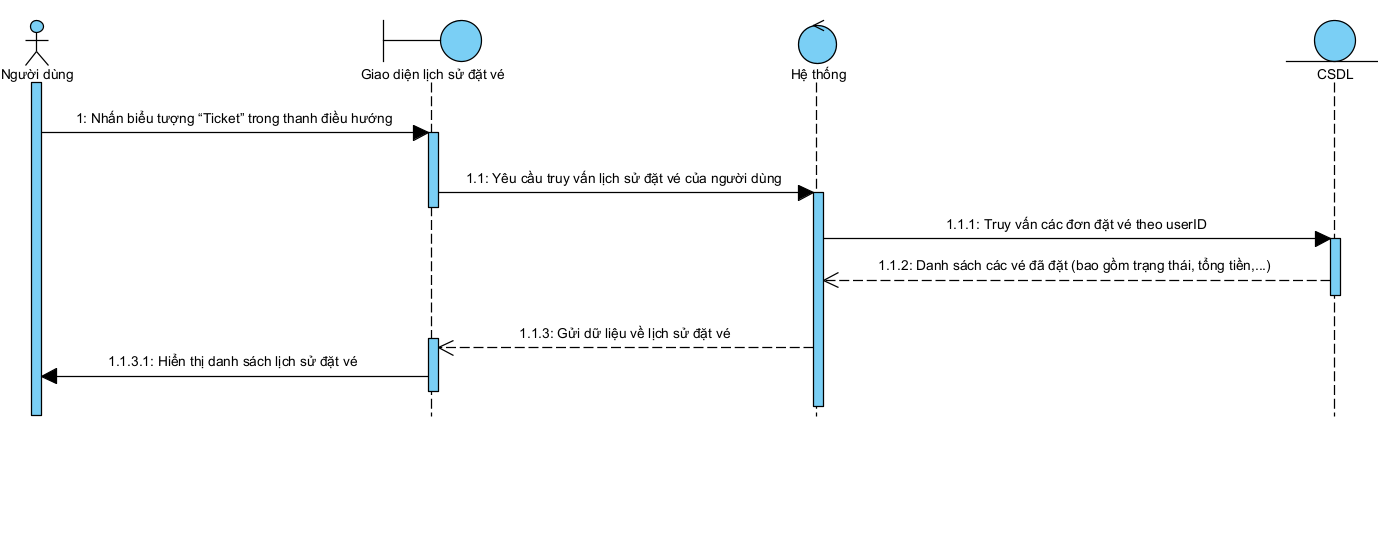
1. **Đặt vé**

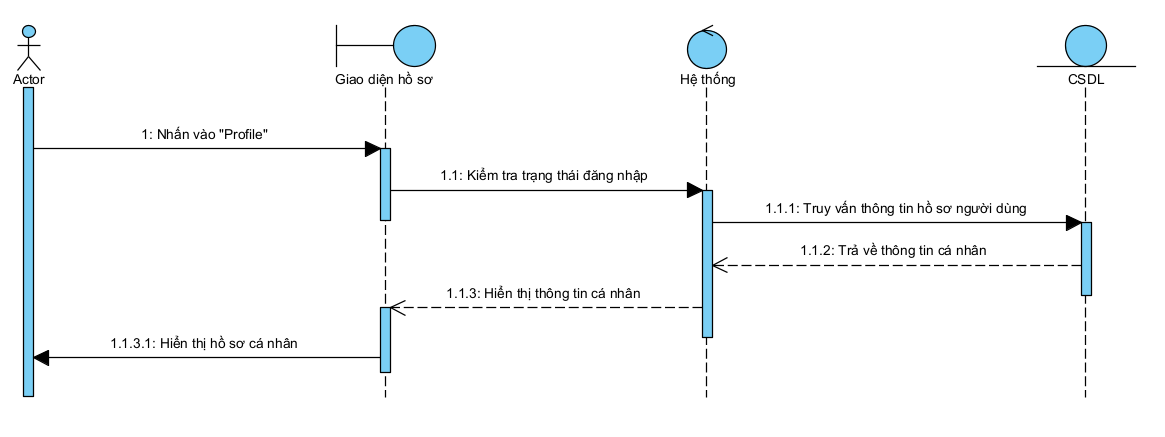
**Hình 3.1.5.e - Sơ đồ tuần tự đặt vé**

1. **Thanh toán**

**Hình 3.1.5.f - Sơ đồ tuần tự thanh toán**

1. **Lịch sử đặt vé**

**Hình 3.1.5.g - Sơ đồ tuần tự lịch sử đặt vé**

1. **Quản lý hồ sơ cá nhân**

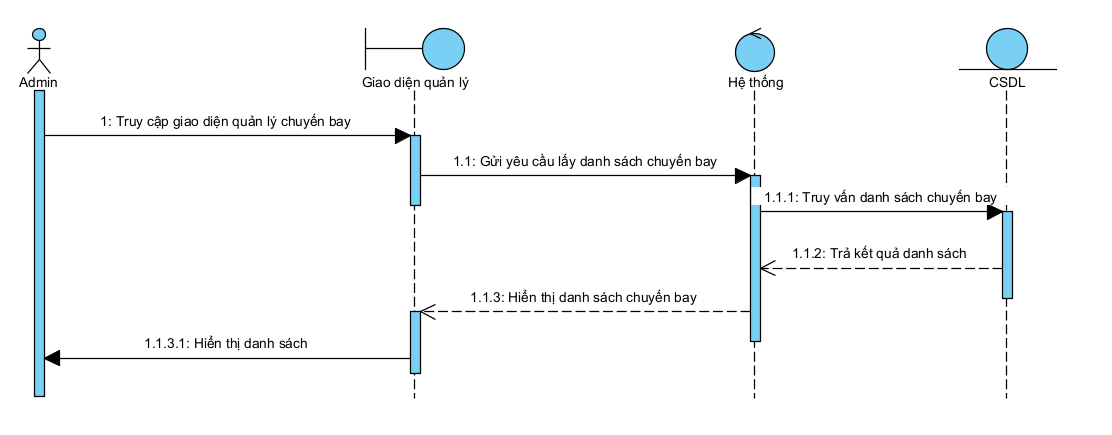
**Hình 3.1.5.h - Sơ đồ tuần tự quản lý hồ sơ cá nhân**

1. **Chỉnh sửa hồ sơ**

****

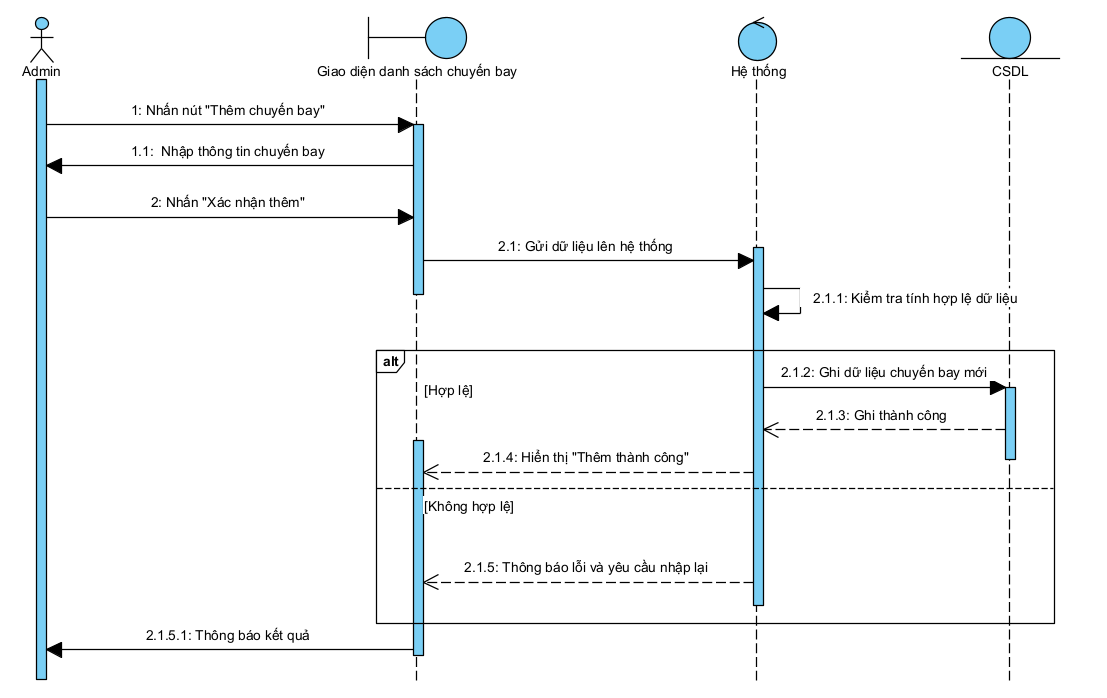
**Hình 3.1.5.i – Sơ đồ tuần tự chỉnh sửa thông tin cá nhân**

1. **Quản lý chuyến bay**

****

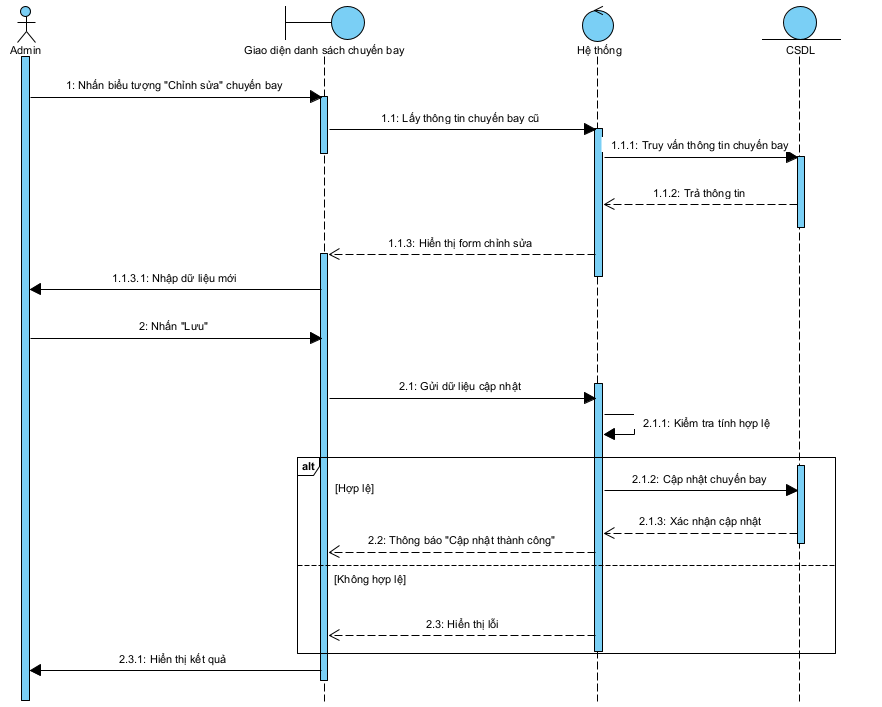
**Hình 3.1.5.j – Sơ đồ tuần tự quản lý chuyến bay**

1. **Thêm chuyến bay**

****

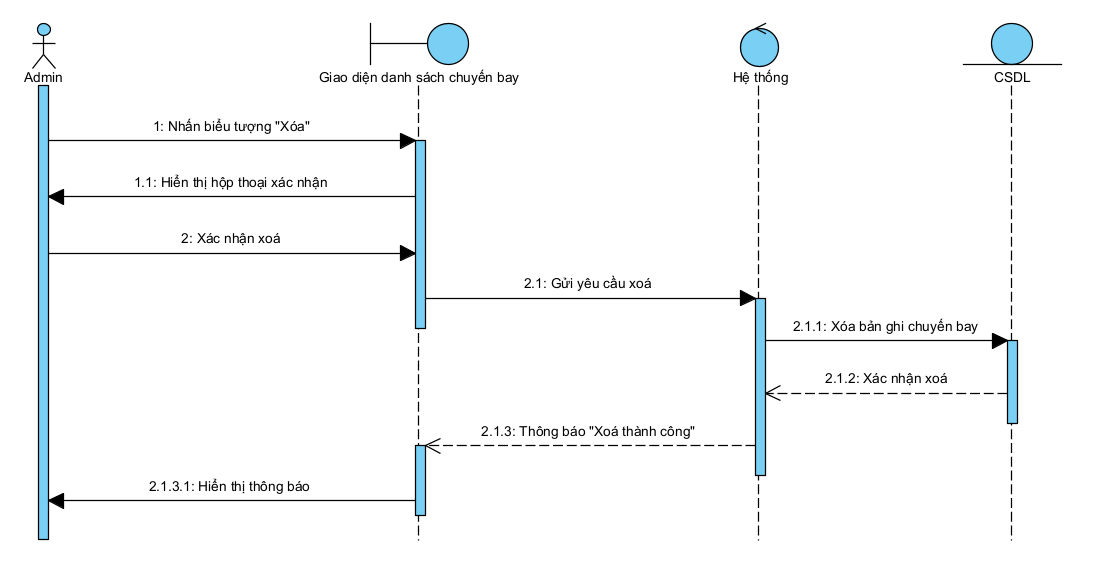
**Hình 3.1.5.k – Sơ đồ tuần tự thêm chuyến bay**

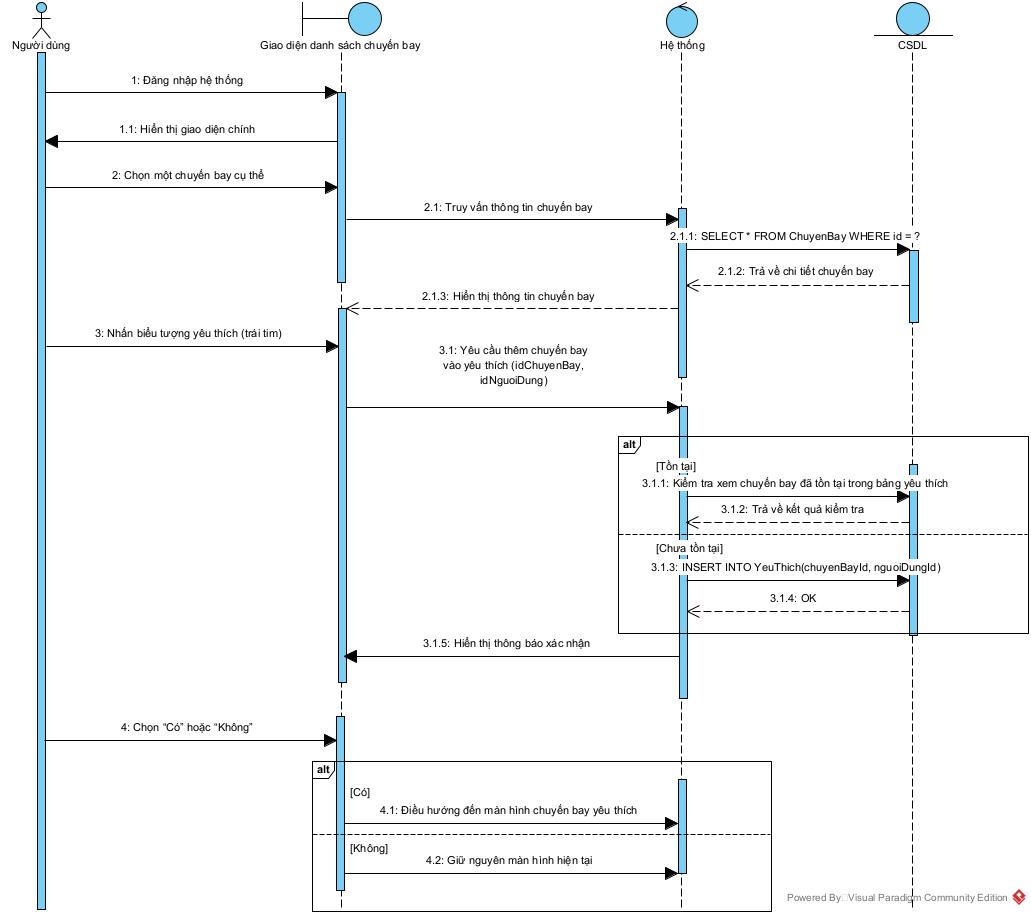
1. **Sửa chuyến bay**

****

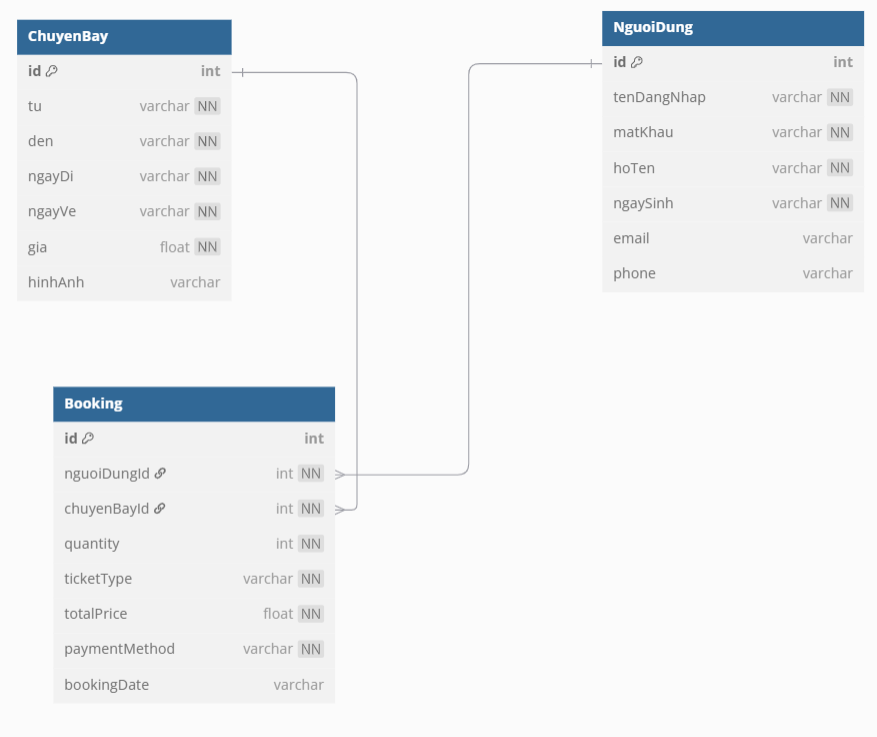
**Hình 3.1.5.l – Sơ đồ tuần tự sửa chuyến bay**

1. **Xóa chuyến bay**

**Hình 3.1.5.m – Sơ đồ tuần tự xóa chuyến bay**

1. **Chuyến bay yêu thích**

**Hình 3.1.5.n – Sơ đồ tuần tự chuyến bay yêu thích**

**3.1.6 Biểu đồ thực thể - liên kết (ERD)**

**Hình 3.1.6. – Biểu đồ thực thể ERD**

**3.2 Thiết kế cơ sở dữ liệu**

**3.2.1 Thiết kế bảng**

1. **Thiết kế bảng NguoiDung:**

**Bảng 3.2.1.1 – Bảng NguoiDung**

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Ý nghĩa** |
| id | INTEGER | ID người dùng   |  | | --- | |  | |
| tenDangNhap | TEXT | Tên đăng nhập |
| matKhau | TEXT | Mật khẩu |
| hoTen | TEXT | Họ tên người dùng |
| ngaySinh | TEXT | Ngày sinh |
| email | TEXT | Email |
| phone | TEXT | Số điện thoại |

Bảng này dùng để lưu thông tin của người dùng

1. **Thiết kế bảng ChuyenBay:**

**Bảng 3.2.1.2 – Bảng ChuyenBay**

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Ý nghĩa** |
| id | INTEGER | ID chuyến bay   |  | | --- | |  | |
| tu | TEXT | Nơi bắt đầu |
| den | TEXT | Nơi đến |
| ngayDi | TEXT | Ngày đi |
| ngayVe | TEXT | Ngày về |
| gia | REAL | Giá |
| hinhAnh | TEXT | Hình ảnh |

Bảng này dùng để lưu thông tin của chuyến bay

1. **Thiết kế bảng Booking:**

**Bảng 3.2.1.3 – Bảng Booking**

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Ý nghĩa** |
| id | INTEGER | ID   |  | | --- | |  | |
| tu | TEXT | Điểm đi |
| den | TEXT | Điểm đến |
| ngayDi | TEXT | Ngày khởi hành |
| ngayVe | TEXT | Ngày trở về |
| quantity | INTEGER | Số lượng vé |
| ticketType | TEXT | Loại vé |
| ticketPrice | REAL | Tổng tiền thanh toán |
| paymentMethod | TEXT | Hình thức thanh toán |

Bảng này dùng để lưu thông tin đặt vé

1. **Thiết kế bảng ChuyenBayYeuThich:**

**Bảng 3.2.1.4 – Bảng ChuyenBayYeuThich**

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Ý nghĩa** |
| id | INTEGER | ID   |  | | --- | |  | |
| userID | INTEGER | ID người dùng |
| chuyenbayID | INTEGER | ID chuyến bay |
| ngayDi | TEXT | Ngày đi |
| ngayVe | TEXT | Ngày về |
| tu | TEXT | Điểm bắt đầu |
| den | TEXT | Điểm đến |
| hinhAnh | TEXT | Hình ảnh chuyến bay |
| giaVe | DOUBLE | Giá vé |
| thoigianYeuThich | TEXT | Thời gian khi vào yêu thích |

Bảng này dùng để lưu thông tin chuyến bay yêu thích

**3.2.2 Tạo cấu trúc và dữ liệu trong hệ quản trị CSDL**

1. **Tạo bảng trong cơ sở dữ liệu**
   * **Bảng NguoiDung: Lưu thông tin tài khoản người dùng**

*CREATE TABLE IF NOT EXISTS NguoiDung (*

*id INTEGER PRIMARY KEY AUTOINCREMENT,*

*tenDangNhap TEXT NOT NULL UNIQUE,*

*matKhau TEXT NOT NULL,*

*hoTen TEXT NOT NULL,*

*ngaySinh TEXT NOT NULL,*

*email TEXT,*

*phone TEXT,*

*role TEXT NOT NULL*

*);*

* + **Bảng Booking: Lưu thông tin các giao dịch đặt vé**

*CREATE TABLE IF NOT EXISTS Booking (*

*id INTEGER PRIMARY KEY AUTOINCREMENT,*

*tu TEXT NOT NULL,*

*den TEXT NOT NULL,*

*ngayDi TEXT NOT NULL,*

*ngayVe TEXT NOT NULL,*

*quantity INTEGER NOT NULL,*

*ticketType TEXT NOT NULL,*

*totalPrice REAL NOT NULL,*

*paymentMethod TEXT NOT NULL,*

*tenNganHang TEXT,*

*soTaiKhoan TEXT,*

*noiDung TEXT,*

*userId INTEGER,*

*price REAL,*

*tax REAL*

*);*

* + **Bảng ChuyenBay: Lưu thông tin các chuyến bay**

*CREATE TABLE IF NOT EXISTS ChuyenBay (*

*id INTEGER PRIMARY KEY AUTOINCREMENT,*

*tu TEXT NOT NULL,*

*den TEXT NOT NULL,*

*ngayDi TEXT NOT NULL,*

*ngayVe TEXT NOT NULL,*

*gia REAL NOT NULL,*

*hinhAnh TEXT,*

*yeuThich INTEGER DEFAULT 0,*

*thoiGianYeuThich INTEGER*

*);*

* + **Bảng ChuyenBayYeuThich: Lưu danh sách chuyến bay được người dùng yêu thích**

*CREATE TABLE IF NOT EXISTS ChuyenBayYeuThich (*

*id INTEGER PRIMARY KEY AUTOINCREMENT,*

*userID INTEGER NOT NULL,*

*chuyenBayID INTEGER NOT NULL,*

*ngayDi TEXT NOT NULL,*

*ngayVe TEXT NOT NULL,*

*tu TEXT NOT NULL,*

*den TEXT NOT NULL,*

*hinhAnh TEXT,*

*giaVe REAL,*

*thoiGianYeuThich INTEGER*

*);*

1. **Chèn dữ liệu mẫu ban đầu**

Hệ thống kiểm tra nếu bảng ChuyenBay đang rỗng thì sẽ tự động chèn dữ liệu mẫu bao gồm danh sách các chuyến bay và thông tin người dùng mẫu.

* + Dữ liệu mẫu cho bảng ChuyenBay

*INSERT INTO ChuyenBay (tu, den, ngayDi, ngayVe, gia, hinhAnh) VALUES*

*('Hà Nội', 'Phú Quốc', '15/06/2025', '20/06/2025', 1500000, 'phuquoc'),*

*('Đà Nẵng', 'Hà Nội', '01/07/2025', '05/07/2025', 1200000, 'hanoi'),*

*('Phú Quốc', 'Đà Nẵng', '25/06/2025', '30/06/2025', 1400000, 'danang'),*

*('Hồ Chí Minh', 'Nha Trang', '10/07/2025', '15/07/2025', 1100000, 'nhatrang'),*

*('Đà Lạt', 'Hà Nội', '20/07/2025', '25/07/2025', 1300000, 'dalat'),*

*('Quảng Ngãi', 'Đà Nẵng', '05/08/2025', '10/08/2025', 900000, 'quangngai'),*

*('Vũng Tàu', 'Phú Quốc', '12/08/2025', '18/08/2025', 1250000, 'vungtau'),*

*('Hà Nội', 'Đà Lạt', '22/08/2025', '28/08/2025', 1350000, 'dalat'),*

*('Nha Trang', 'Vũng Tàu', '01/09/2025', '05/09/2025', 1150000, 'vungtau'),*

*('Đà Nẵng', 'Quảng Ngãi', '08/09/2025', '12/09/2025', 950000, 'quangngai'),*

*('Hà Nội', 'Phú Quốc', '05/07/2025', '10/07/2025', 1550000, 'phuquoc'),*

*('Đà Nẵng', 'Hà Nội', '10/07/2025', '15/07/2025', 1250000, 'hanoi'),*

*('Phú Quốc', 'Đà Nẵng', '01/08/2025', '06/08/2025', 1450000, 'danang'),*

*('Hà Nội', 'Đà Lạt', '01/09/2025', '06/09/2025', 1400000, 'dalat'),*

*('Nha Trang', 'Vũng Tàu', '10/09/2025', '15/09/2025', 1170000, 'vungtau');*

* + Dữ liệu mẫu cho bảng NguoiDung

*INSERT INTO NguoiDung (tenDangNhap, matKhau, hoTen, ngaySinh, email, phone, role) VALUES*

*('admin', '123456', 'Nguyễn Văn A', '30/06/2005', 'admin@example.com', '0123456789', 'admin'),*

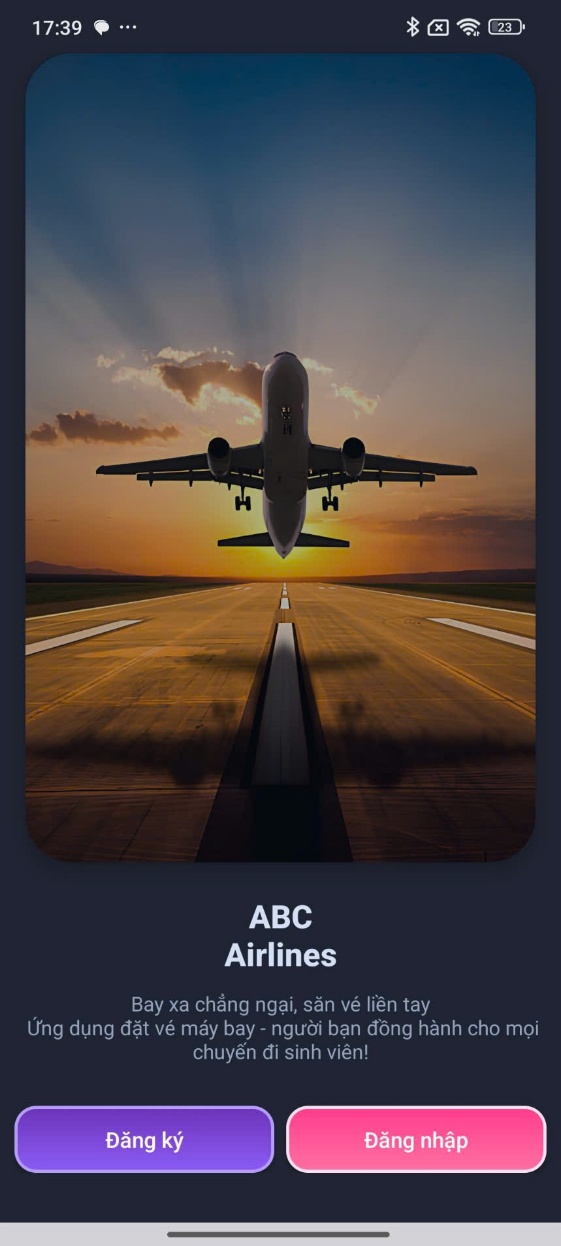
*('duongminhhuy', 'huy123', 'Dương Minh Huy', '30/06/2005', 'huy@example.com', '0981522360', 'user');*

**3.3. Xây dựng giao diện sản phẩm**

Ứng dụng được xây dựng với **Android Studio**, sử dụng **Kotlin**, tích hợp SQLite qua DatabaseHelper, và chia UI thành nhiều màn hình chức năng. Giao diện tuân theo nguyên tắc Material Design, dễ thao tác, rõ ràng.

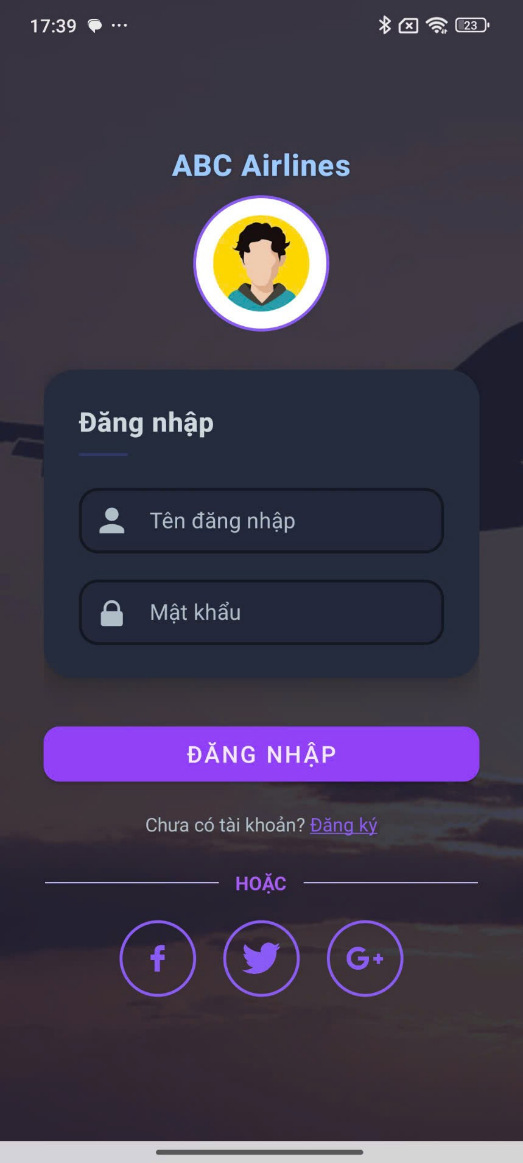
**3.3.1 Giao diện mở đầu**

* **Tên chức năng:** BeginActivity
* **Chức năng:**
  + Hiển thị giao diện bắt đầu ứng dụng.
* Hình ảnh:

****

**Hình 3.3.1 – Giao diện mở đầu**

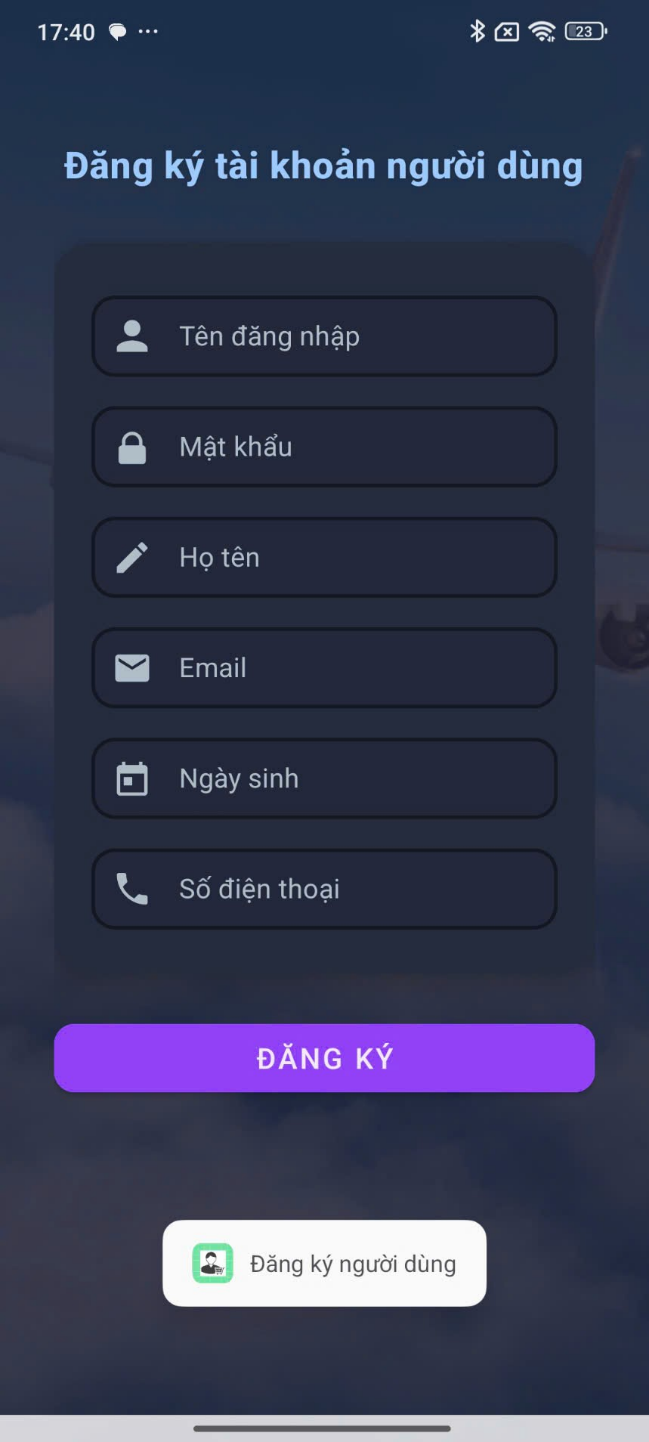
**3.3.2 Giao diện đăng nhập**

* **Tên chức năng:** DangNhapActivity
* **Chức năng:**
  + Cho phép người dùng đăng nhập bằng tên đăng nhập và mật khẩu.
* **Mô tả:**
  + Gọi hàm getNguoiDungByTenDangNhap() để kiểm tra trong bảng NguoiDung.
* ****Hình ảnh:

**Hình 3.3.2 – Giao diện đăng nhập**

**3.3.3 Giao diện đăng ký**

* **Tên chức năng:** DangKyActivity
* **Chức năng:**
  + Cho phép người dùng tạo tài khoản mới.
* **Giải thích:**
  + Sau khi người dùng nhập thông tin, hàm dangKyNguoiDung() sẽ lưu dữ liệu vào bảng NguoiDung.
* Hình ảnh:



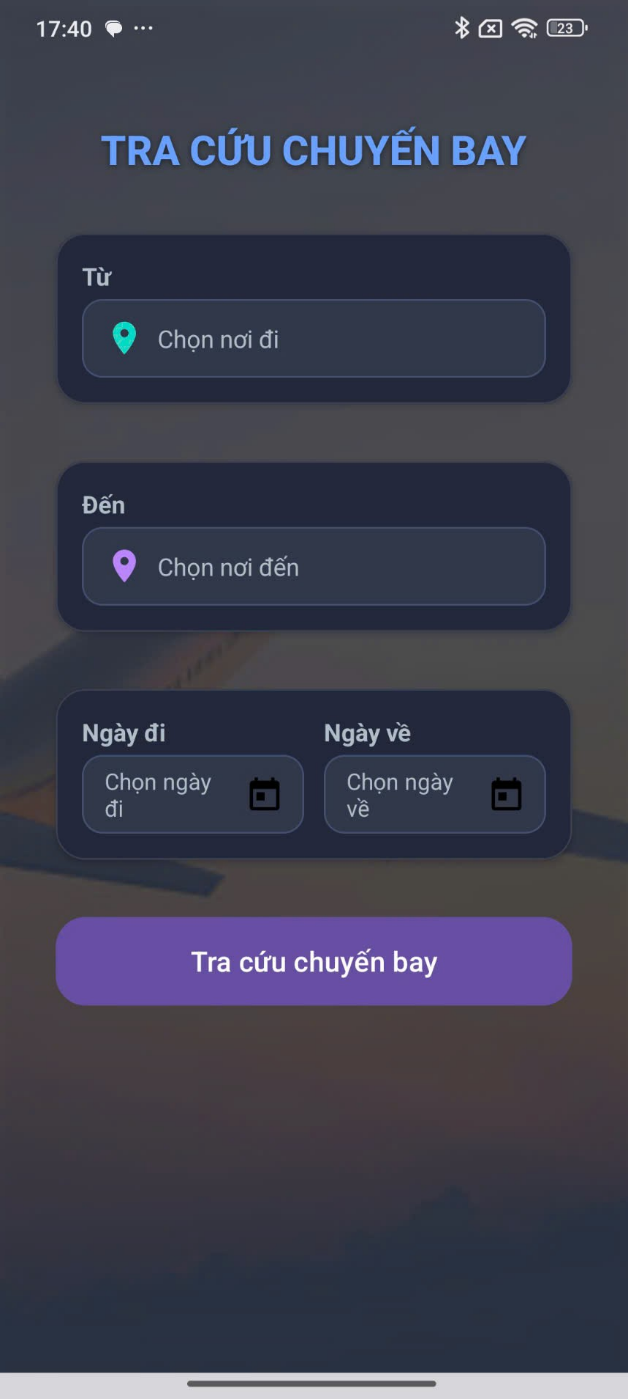
**Hình 3.3.3 – Giao diện đăng ký**

**3.3.4 Giao diện trang chủ - Chuyến bay nổi bật**

* **Tên chức năng:** MainActivity
* **Chức năng:**
  + Hiển thị 5 chuyến bay đầu tiên hấp dẫn.
* **Giải thích:**
  + Dữ liệu được lấy từ getFeaturedChuyenBays() và hiển thị bằng RecyclerView.
* ****Hình ảnh:

**Hình 3.3.4 – Giao diện trang chủ**

**3.3.5 Giao diện tra cứu chuyến bay**

* **Tên chức năng:** TraCuuChuyenBay
* **Chức năng:**
  + Người dùng chọn nơi đi, nơi đến, ngày đi và ngày về để tìm chuyến bay.
* **Giải thích:**
  + Gọi hàm getAllChuyenBay() để lấy dữ liệu lọc theo các tiêu chí.
* Hình ảnh:

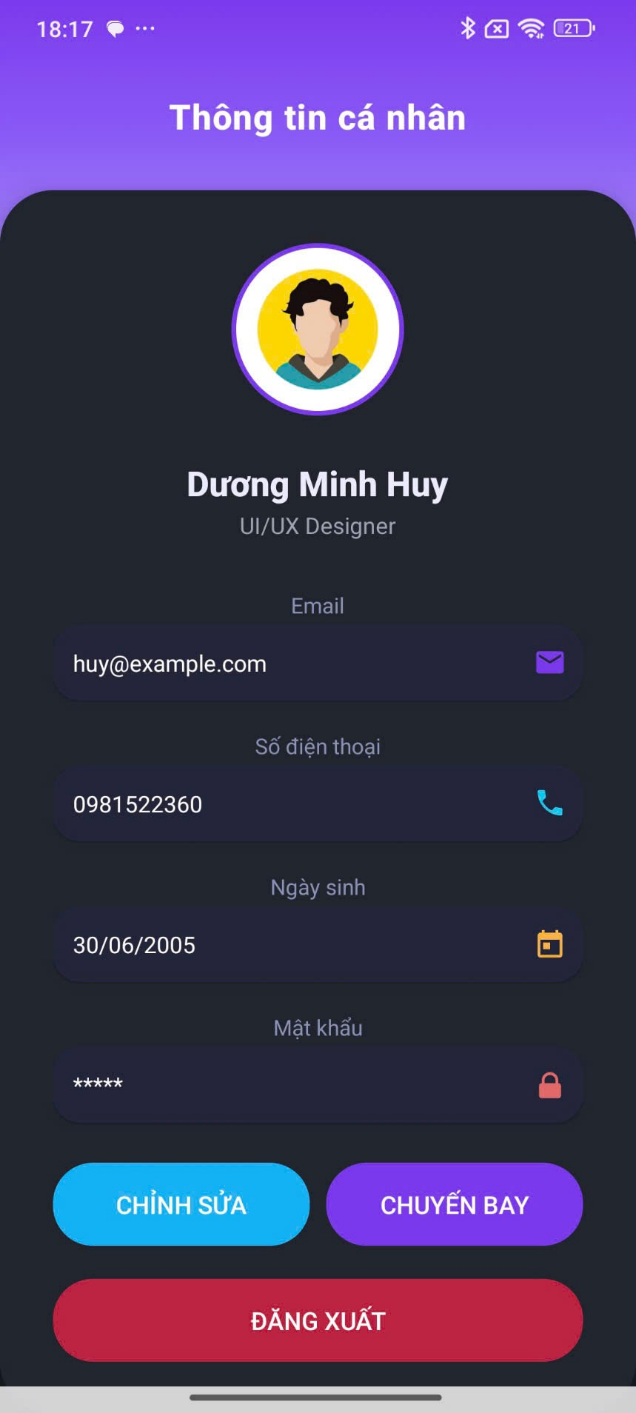
**Hình 3.3.5 – Giao diện tra cứu chuyến bay**

**3.3.6 Giao diện đặt vé**

* **Tên chức năng:** DatVeActivity
* **Chức năng:**
  + Cho phép người dùng chọn loại vé, số lượng, phương thức thanh toán.
* **Giải thích:**
  + Người dùng nhập đầy đủ thông tin đặt vé, nhấn “Đặt vé” sẽ chuyển sang giao diện thanh toán.
* Hình ảnh:

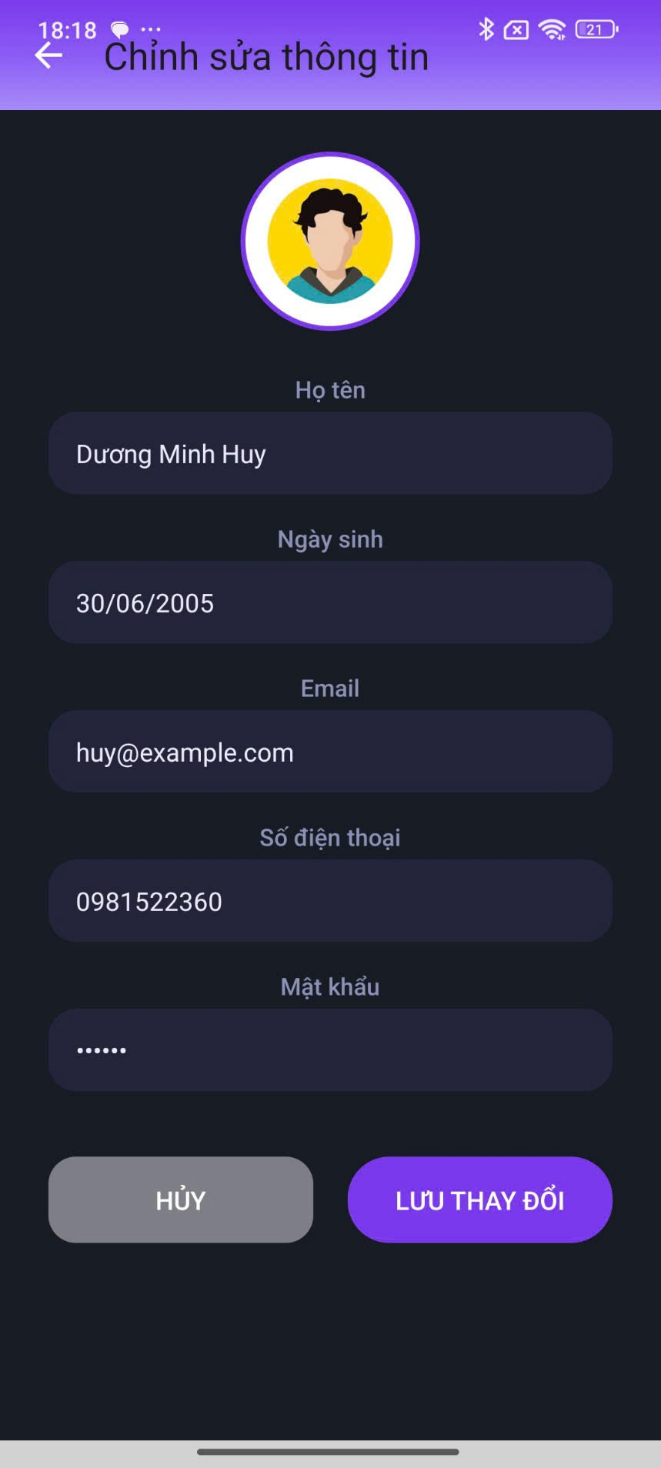
**Hình 3.3.6 – Giao diện đặt vé**

**3.3.7 Giao diện thông tin người dùng**

* **Tên chức năng:** ProfileActivity
* **Chức năng:**
  + Hiển thị và thông tin người dùng.
* **Giải thích:**
  + Lấy dữ liệu từ getUserByUsername().
* Hình ảnh:

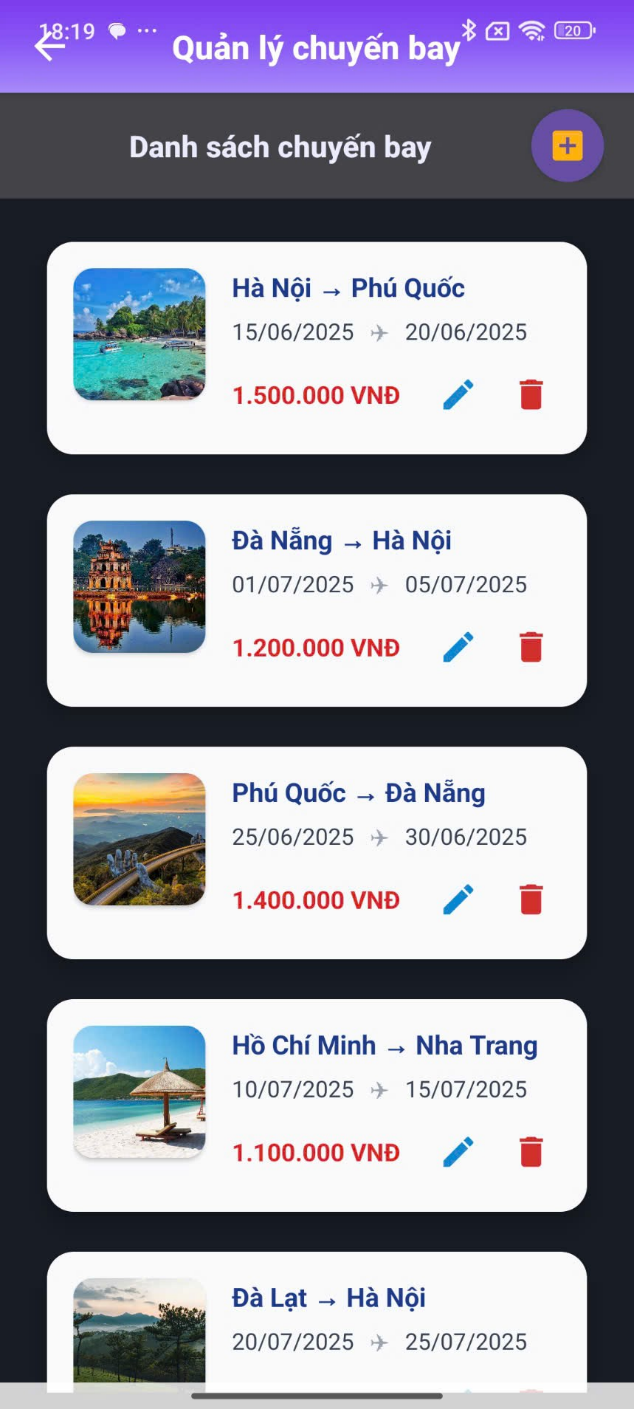
**Hình 3.3.7 – Giao diện thông tin người dùng**

**3.3.8 Giao diện chỉnh sửa thông tin người dùng**

* **Tên chức năng:** EditUserActivity
* **Chức năng:**
  + Chỉnh sửa và cập nhật thông tin người dùng.
* **Giải thích:**
  + Lấy dữ liệu từ getNguoiDung() và cập nhật qua updateNguoiDung().
* Hình ảnh:

**Hình 3.3.8 – Giao diện chỉnh sửa thông tin người dùng**

**3.3.9 Giao diện quản lý chuyến bay (Admin)**

* **Tên chức năng:** AdminQuanLyChuyenBay
* **Chức năng:**
  + Hiển thị tất cả chuyến bay có trong CSDL.
  + Thêm, xóa, sửa các chuyến bay có trong hệ thống.
* **Giải thích:**
  + Giao diện sử dụng insertChuyenBay(), updateChuyenBay() và deleteChuyenBay() để thao tác trực tiếp với bảng ChuyenBay.
* Hình ảnh:

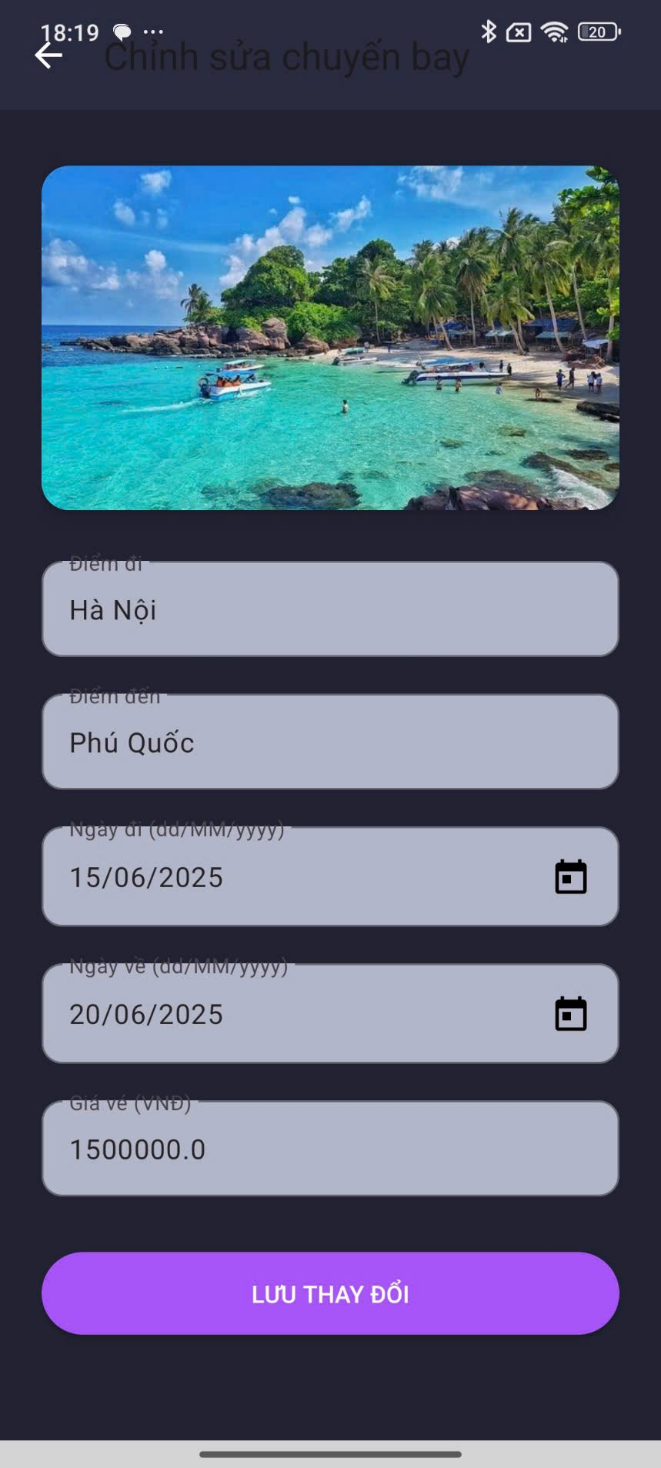
**Hình 3.3.9 – Giao diện quản lý chuyến bay**

**3.3.10 Giao diện thêm chuyến bay (Admin)**

* **Tên chức năng:** AddChuyenBayActivity
* **Chức năng:**
  + Thêm chuyến bay mới.
* **Giải thích:**
  + Quản trị viên nhập đầy đủ thông tin chuyến bay sau đó nhấn “Thêm chuyến bay”.
  + Gọi hàm insertChuyenbay() để kiểm tra thông tin và thêm chuyến bay mới.
* Hình ảnh:

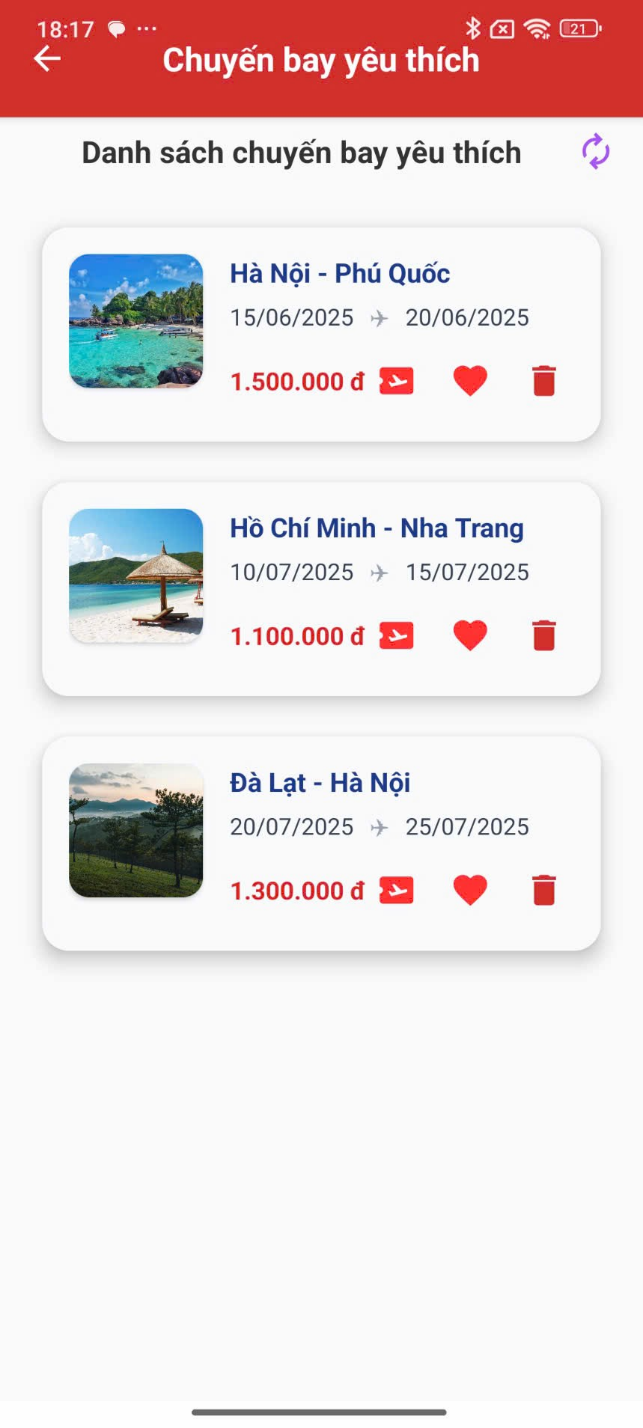
**Hình 3.3.10 – Giao diện thêm chuyến bay**

**3.3.11 Giao diện chỉnh sửa chuyến bay (Admin)**

* **Tên chức năng:** EditChuyenBayActivity
* **Chức năng:**
  + Chỉnh sửa và cập nhật thông tin chuyến bay.
* **Giải thích:**
  + Giao diện sử dụng updateChuyenBay() để kiểm tra thông tin và cập nhật dữ liệu chuyến bay.
* Hình ảnh:

**Hình 3.3.11 – Giao diện chỉnh sửa chuyến bay**

**3.3.12 Giao diện chuyến bay yêu thích**

* **Tên chức năng:** QuanLyChuyenBayYeuThich
* **Chức năng:**
  + Hiển thị danh sách chuyến bay yêu thích và cập nhật dữ liệu sau khi thay đổi.
* **Giải thích:**
  + Giao diện sử dụng xoaChuyenBayYeuThich() xoá chuyến bay yêu thích ra khỏi danh sách theo yêu cầu người dùng.
  + Giao diện sử dụng updateData() để cập nhật lại danh sách sau khi thay đổi.
* Hình ảnh:

**Hình 3.3.12 – Giao diện chuyến bay yêu thích**

**KẾT LUẬN**

**Kết quả đề tài**

Sau một quá trình tìm hiểu, thiết kế và xây dựng, nhóm chúng em đã hoàn thành đề tài **“Quản lý đặt vé máy bay”** – một ứng dụng Android có khả năng hỗ trợ người dùng trong việc tra cứu, lựa chọn và đặt vé máy bay một cách đơn giản, tiện lợi và hiệu quả. Đề tài đã thực hiện được những mục tiêu đề ra ban đầu, cụ thể:

* Thiết kế giao diện người dùng trực quan, thân thiện với người dùng theo chuẩn Material Design, giúp việc thao tác trở nên dễ dàng trên thiết bị di động.
* Xây dựng hệ thống chức năng hoàn chỉnh gồm: đăng ký, đăng nhập, tra cứu chuyến bay theo nơi đi – nơi đến – ngày khởi hành, đặt vé với số lượng – loại vé – phương thức thanh toán, quản lý thông tin cá nhân và chức năng quản lý chuyến bay dành cho Admin.
* Ứng dụng vận hành độc lập nhờ sử dụng cơ sở dữ liệu nội bộ SQLite, đảm bảo tính tiện dụng và bảo mật ở mức cơ bản mà không cần kết nối internet.
* Tổ chức mã nguồn khoa học, chia rõ các lớp giao diện, xử lý logic và cơ sở dữ liệu qua lớp DatabaseHelper.kt, thể hiện được kiến thức về mô hình 3 lớp đơn giản trong phát triển ứng dụng.

Qua đề tài, nhóm không chỉ nâng cao kỹ năng lập trình Kotlin và thiết kế ứng dụng Android mà còn hiểu rõ hơn về quy trình phát triển phần mềm từ giai đoạn phân tích – thiết kế – hiện thực – kiểm thử. Đây là trải nghiệm quý báu giúp nhóm hình dung rõ hơn công việc thực tế của một lập trình viên mobile, đồng thời tăng tính chủ động trong nghiên cứu và sáng tạo.

Tuy nhiên, do thời gian và giới hạn kiến thức, đề tài vẫn còn một số điểm chưa hoàn thiện, như: chưa hỗ trợ thanh toán trực tuyến, chưa có thông báo đặt vé thành công qua email hay SMS, chưa tích hợp bản đồ để xem vị trí sân bay, và chưa có chức năng lọc nâng cao theo hãng bay, giờ bay, mức giá...

**Hướng phát triển**

Trong tương lai, để hoàn thiện và mở rộng ứng dụng **“Quản lý đặt vé máy bay”**, nhóm xin đề xuất một số hướng phát triển như sau:

🔹 **1. Hoàn thiện và tối ưu chức năng hiện có**

* **Cải tiến giao diện** bằng cách thêm hiệu ứng chuyển cảnh (animation), hỗ trợ chế độ tối (dark mode) nhằm nâng cao trải nghiệm người dùng.
* **Tối ưu hiệu suất**, giảm độ trễ khi tải dữ liệu từ SQLite, và thêm xác thực mạnh hơn cho đăng nhập (ví dụ xác minh email hoặc OTP).
* **Bổ sung thông báo (Notification)** sau khi đặt vé thành công hoặc khi có thay đổi chuyến bay (áp dụng với cơ sở dữ liệu mở rộng sau này).

🔹 **2. Mở rộng tích hợp với hệ thống thực tế**

* **Tích hợp API của các hãng hàng không** (như Vietnam Airlines, VietJet Air...) để lấy dữ liệu chuyến bay thật, giúp ứng dụng vận hành như một nền tảng thương mại điện tử thực sự.
* **Tích hợp thanh toán trực tuyến** qua cổng VNPAY, Momo, hoặc thẻ ngân hàng để hoàn thiện quy trình đặt vé.
* **Gửi email xác nhận vé** qua SMTP hoặc tích hợp dịch vụ thứ ba (như Firebase hoặc SendGrid).

🔹 **3. Nâng cấp kiến trúc và dữ liệu**

* **Chuyển đổi từ SQLite sang Room Database** (thuộc Jetpack) để đảm bảo an toàn và dễ quản lý dữ liệu hơn.
* Xây dựng **RESTful API và Web Server** kết nối ứng dụng với cơ sở dữ liệu tập trung trên Cloud (ví dụ dùng Firebase hoặc ASP.NET Core Web API kết nối SQL Server).
* Áp dụng mô hình kiến trúc **MVVM (Model – View – ViewModel)** để dễ bảo trì và mở rộng dự án trong tương lai.

🔹 **4. Triển khai và đưa vào sử dụng thực tế**

* Đăng tải ứng dụng thử nghiệm trên **Google Play (beta test)** để thu nhận ý kiến phản hồi người dùng.
* Hợp tác thử nghiệm với các công ty lữ hành nhỏ hoặc sinh viên khoa du lịch để đánh giá khả năng ứng dụng thực tế của phần mềm.

**PHỤ LỤC**

**1. Link GitHub chứa toàn bộ mã nguồn của đề tài:**

<https://github.com/DuongQuangMinh2005/QuanLyDatVe_SQLite_Nhom9>

**2. Ghi chú:**

* Repository bao gồm toàn bộ mã nguồn Android Studio (Java/Kotlin), tài nguyên và tài liệu báo cáo.
* Mỗi chức năng trong ứng dụng đều được đặt trong thư mục tương ứng, có thể tham khảo theo README.md trong repository.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] JetBrains, Kotlin Language Documentation, Kotlinlang.org. [Online]. Available: https://kotlinlang.org/docs/home.html

[2] Google, Android Developer Guides, developer.android.com. [Online]. Available: <https://developer.android.com/guide>

[3] Google, Android Jetpack: Room Persistence Library, developer.android.com. [Online]. Available: <https://developer.android.com/jetpack/androidx/releases/room>

[4] D. Sauter, Android Studio IDE Quick Guide, 2nd ed., Apress, 2020. DOI: 10.1007/978-1-4842-5963-6

[5] M. Banerjee, Hands-On Android UI Development with Kotlin, Packt Publishing, 2020. ISBN: 9781838826398

[6] K. S. Anitha and P. M. Dinesh, "Mobile Application for Airline Reservation System using Android," International Journal of Engineering Research & Technology (IJERT), vol. 8, no. 12, pp. 386–390, Dec. 2019. [Online]. Available: https://www.ijert.org/mobile-application-for-airline-reservation-system-using-android

[7] P. Meier, *SQL and SQLite for Beginners: A Step by Step Guide to Learn SQL and SQLite Programming*, Independently published, 2020. ISBN: 9798652171752.

[8] J. N. Smith and K. Jones, *Mobile App Design: From Sketches to Prototype with Android Studio*, Wiley, 2021. ISBN: 9781119806820.

[9] J. H. Christensen, "User Interface Design Principles for Mobile Applications," *International Journal of Human-Computer Studies*, vol. 132, pp. 34–45, Jan. 2020. DOI: 10.1016/j.ijhcs.2019.06.003.

[10] M. Shafique, T. Theocharides, A. Sagahyroon, et al., "The 10 Emerging Technologies in Mobile Computing: A Research Roadmap," *IEEE Design & Test*, vol. 36, no. 5, pp. 7–20, Oct. 2019. DOI: 10.1109/MDAT.2019.2933044.