

**Trường công nghệ thông tin  
Phenikaa**



**ĐỒ ÁN MÔN HỌC  
LẬP TRÌNH CHO THIẾT BỊ DI ĐỘNG**

**THIẾT KẾ ỨNG DỤNG ĐẶT VÉ MÁY BAY FLYBOX**

**Giảng viên hướng dẫn: THS. NGUYỄN VĂN CƯỜNG**

**Lớp: Lập trình cho thiết bị di động - N04**

**Khoá: K16**

**Sinh viên thực hiện: Phạm Thị Vân Anh - 22010048**

**Trần Thị Thùy Trang - 22010294**

**Nguyễn Thị Cẩm Tú - 22010230**

**Hà Nội, tháng 3 năm 2025**

## MỤC LỤC

Lời cảm ơn.....	1
Lời cam đoan.....	2
Phiếu chấm điểm.....	3
Phân công nhiệm vụ.....	4
1.Giới thiệu.....	5
1.1. Mô tả tổng quan về ứng dụng Flybox.....	5
1.2. Mục tiêu phát triển ứng dụng.....	5
1.3. Bảng thuật ngữ.....	6
1.4. Công nghệ sử dụng.....	7
2.Phân tích và Thiết kế.....	8
2.1. Kiến trúc ứng dụng (MVVM).....	8
2.2. Luồng hoạt động chính của ứng dụng.....	10
2.3. Thiết kế giao diện người dùng (UI/UX).....	16
3.Chức năng chính của ứng dụng.....	18
3.1. Đăng ký, đăng nhập, đăng xuất.....	18
3.2. Tìm kiếm và đặt vé chuyến bay.....	20
3.3. Quản lý thông tin cá nhân.....	21
3.4. Thanh toán vé máy bay.....	22
3.5. Quản lý danh sách chuyến bay đã đặt.....	23
4.Triển khai và Kiểm thử.....	23
4.1. Môi trường phát triển và triển khai.....	23
4.2. Các công cụ kiểm thử.....	24
4.3. Debugging và tối ưu hóa hiệu suất sử dụng.....	25
5. Báo cáo thực nghiệm.....	27
6.Kết luận và Đánh giá.....	35
6.1. Kết quả đạt được.....	35
6.2. Hạn chế và hướng phát triển trong tương lai.....	35
6.3. Bài học kinh nghiệm.....	35
7. Tài liệu tham khảo.....	35
8. Link slide và github.....	36

## LỜI CẢM ƠN

Kính gửi ThS. Nguyễn Văn Cường – Giảng viên môn Lập trình cho thiết bị di động.

Chúng em là thành viên nhóm 7, sinh viên lớp Lập trình cho thiết bị di động (N04), xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy vì sự tận tâm trong giảng dạy và hướng dẫn chúng em trong suốt học kỳ vừa qua. Đặc biệt, chúng em rất cảm kích vì sự hỗ trợ và định hướng của thầy trong quá trình thực hiện bài tập lớn môn học này.

Trước hết, chúng em xin bày tỏ lòng biết ơn vì thầy đã truyền đạt cho chúng em những kiến thức quý báu về lập trình mobile, giúp chúng em hiểu rõ về cách xây dựng ứng dụng di động một cách hiệu quả. Thông qua các bài giảng chi tiết và thực tiễn, chúng em không chỉ nắm vững các khái niệm lập trình bất đồng bộ, quản lý trạng thái, và kiến trúc ứng dụng mà còn có cơ hội ứng dụng vào dự án thực tế.

Không chỉ dừng lại ở kiến thức chuyên môn, thầy còn khuyến khích chúng em rèn luyện tư duy logic, kỹ năng làm việc nhóm và khả năng giải quyết vấn đề trong phát triển ứng dụng di động. Nhờ sự hướng dẫn tận tình của thầy, chúng em đã vượt qua nhiều thử thách, hoàn thiện bài tập lớn và tích lũy thêm nhiều kinh nghiệm quý giá để áp dụng trong tương lai.

Chúng em cũng xin chân thành cảm ơn thầy vì luôn sẵn sàng lắng nghe, giải đáp thắc mắc và hỗ trợ chúng em trong suốt quá trình học tập. Sự nhiệt huyết và tận tâm của thầy chính là nguồn động lực lớn giúp chúng em không ngừng nỗ lực và phát triển bản thân.

Một lần nữa, chúng em xin chân thành cảm ơn thầy. Kính chúc thầy luôn mạnh khỏe, hạnh phúc và thành công trong sự nghiệp giảng dạy của mình.

Trân trọng!

Sinh viên nhóm 7

Lớp Lập trình cho thiết bị di động (N04)

## **LỜI CAM ĐOAN**

### **1. Tính độc lập:**

- Đồ án môn học này là sản phẩm trí tuệ của tập thể chúng em, được thực hiện dưới sự hướng dẫn của ThS. Nguyễn Văn Cường.
- Mọi ý tưởng, nội dung, kết quả nghiên cứu và phân tích trong bài tập lớn đều do chúng em tự nghiên cứu, phát triển và hoàn thành, không sao chép hay sử dụng trái phép tài liệu của bất kỳ cá nhân, tổ chức nào khác.

### **2. Tính chính xác:**

- Mọi thông tin, dữ liệu, hình ảnh, tài liệu tham khảo sử dụng trong bài tập lớn đều được kiểm chứng kỹ lưỡng về độ chính xác, tin cậy và được trích dẫn đầy đủ theo đúng quy định.
- Các nguồn tài liệu tham khảo được liệt kê đầy đủ trong danh mục tài liệu tham khảo và có thể kiểm chứng một cách minh bạch.

### **3. Trách nhiệm:**

- Chúng em xin hoàn toàn chịu trách nhiệm về nội dung và hình thức trình bày của bài tập lớn, đảm bảo không vi phạm các quy định về đạo đức học tập và quy chế thi cử của nhà trường.
- Chúng em cam kết mọi nội dung trong bài báo cáo là kết quả làm việc nghiêm túc và có sự đóng góp thực tế của nhóm.

Hà Nội, ngày 17 tháng 3 năm 2025

Nguyễn Thị Cẩm Tú, Phạm Thị Vân Anh, Trần Thị Thùy Trang (Ký tên)

**Trường Đại học Phenikaa  
Khoa Công nghệ thông tin**

**Kỳ thi kết thúc học phần  
Học kì 2, năm học 2024 - 2025**

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

Môn học: Lập trình cho thiết bị di động

Lớp học phần: N04

Sinh viên thực hiện:.....

1..... Điểm:

2..... Điểm:

3..... Điểm:

4..... Điểm:

5..... Điểm:

Ngày thi: 19/03/2025.....Phòng thi:A6 - 405

Giảng viên chấm thi 1

Giảng viên chấm thi 2

(Ký và ghi rõ họ tên)

(Ký và ghi rõ họ tên)

### Phân công nhiệm vụ

Công việc	Nhiệm vụ
Khảo sát và phân tích yêu cầu	<ol style="list-style-type: none"> <li>1. Xác định nhu cầu người dùng.</li> <li>2. Xác định yêu cầu chức năng.</li> <li>3. Xác định yêu cầu phi chức năng.</li> <li>4. Lập tài liệu yêu cầu (SRS)</li> </ol>
Thiết kế hệ thống	<ol style="list-style-type: none"> <li>1. Thiết kế kiến trúc tổng thể.</li> <li>2. Thiết kế mô hình dữ liệu.</li> <li>3. Thiết kế giao diện người dùng.</li> <li>4. Thiết kế luồng xử lý chính.</li> </ol>
Triển khai và phát triển	<ol style="list-style-type: none"> <li>1. Chọn môi trường phát triển (flutter).</li> <li>2. Phát triển các module chính.</li> <li>3. Tích hợp các hệ thống.</li> <li>4. Kiểm thử phần mềm.</li> <li>5. Triển khai lên máy chủ.</li> </ol>
Báo cáo và slide	<ol style="list-style-type: none"> <li>1. Làm báo cáo</li> <li>2. Làm slide</li> </ol>

### Tỷ lệ tham gia công việc

Công việc	Phạm Thị Vân Anh	Nguyễn Thị Cẩm Tú	Trần Thị Thùy Trang
Khảo sát và phân tích yêu cầu	35%	30%	35%
Thiết kế hệ thống	35%	35%	30%
Triển khai và phát triển	30%	35%	35%
Báo cáo và slide	30%	30%	40%

# **1. Giới thiệu**

## **1.1. Mô tả tổng quan về ứng dụng Flybox.**

Flybox là một ứng dụng được phát triển nhằm hỗ trợ người dùng tìm kiếm, so sánh và cài đặt vé máy bay một cách nhanh chóng, thuận tiện trên điện thoại thông minh. Ứng dụng được thiết kế với giao diện thân thiện, dễ sử dụng, giúp người dùng dễ dàng tìm kiếm thông tin chuyến bay, đơn vị hãng hàng không phù hợp, đặt vé trực tuyến và quản lý lịch trình bay một cá

Trong bối cảnh nhu cầu chuyển bằng đường hàng không ngày càng tăng, Flybox ra đời như một giải pháp công nghệ tối ưu, giúp khách hàng tiết kiệm thời gian, công sức và chi phí khi đặt vé máy bay. Vì phải truy cập nhiều trang web khác nhau để tìm kiếm vé hoặc đến trực tiếp các đại lý bán vé, người dùng chỉ cần mở Flybox ứng dụng để thực hiện mọi thao tác một cách nhanh chóng

Bên cạnh đó, Flybox không chỉ cung cấp tính năng thiết lập vé đơn thuần mà còn hỗ trợ các tiện ích đi kèm như lưu trữ thông tin cá nhân, quản lý lịch sử đặt vé, thông báo nhắc nhở chuyến bay sắp diễn ra và cập nhật tình trạng chuyến bay theo thời gian thực. Ứng dụng cũng tích hợp nhiều phương thức thanh toán an toàn, giúp người dùng dễ dàng giao dịch mà không cần lo lắng về vấn đề bảo trì.

## **1.2. Mục tiêu phát triển ứng dụng.**

Ứng dụng Flybox được phát triển với mục tiêu chính là cung cấp nền tảng đặt vé máy bay nhanh chóng, tiện lợi và an toàn, giúp người dùng dễ dàng tìm kiếm, so sánh giá vé và quản lý chuyến bay ngay trên thiết bị di động. Cụ thể, như sau:

- Cải thiện trải nghiệm của người dùng khi cài đặt máy bay
  - Tạo ra một giao diện đơn giản, dễ sử dụng cho người dùng ở mọi độ tuổi đều có thể thao tác dễ dàng.
  - Hỗ trợ tìm kiếm nhanh và lọc vé theo nhu cầu (giá cả, hãng bay, thời gian khởi động, vv).
  - Tích hợp hệ thống thông báo nhắc nhở về chuyến đi, giúp người dùng không bỏ lỡ hành trình
- Hỗ trợ cài đặt máy bay một cách linh hoạt và an toàn

- Cung cấp thông tin đầy đủ, chính xác về chuyến đi , giúp người dùng dễ dàng đưa ra quyết định.
  - Hỗ trợ nhiều phương thức thanh toán trực tuyến (ví điện tử, thẻ tín dụng, chuyển tài khoản, vv) để tăng cường sự thuận tiện.
  - Tích hợp nâng cao bảo mật , đảm bảo thông tin cá nhân và giao dịch của người dùng được bảo vệ tốt nhất.
- Quản lý vé và kết quả hiệu quả của lịch trình
- Cho phép người dùng lưu trữ và xem lại vé đã đặt , giúp dễ dàng quản lý lịch trình.

### 1.3. Bảng thuật ngữ

Thuật Ngữ	Định Nghĩa
Flutter	Bộ công cụ UI của Google giúp xây dựng ứng dụng trên nhiều nền tảng từ một codebase duy nhất.
Dart	Ngôn ngữ lập trình do Google phát triển, được sử dụng chính trong Flutter.
Hot Reload	Tính năng của Flutter giúp lập trình viên thấy ngay thay đổi mà không cần build lại toàn bộ ứng dụng.
Material Design	Nguyên tắc thiết kế giao diện do Google phát triển.
Cupertino	Bộ giao diện trong Flutter mô phỏng phong cách của iOS.
API (Application Programming Interface)	Giao diện lập trình ứng dụng, giúp các hệ thống tương tác với nhau.
Firebase	Nền tảng cung cấp các dịch vụ backend như authentication, database, cloud functions.
Performance Optimization	Kỹ thuật tối ưu hóa hiệu suất ứng dụng, bao gồm giảm tải UI, quản lý bộ nhớ,...
Debugging	Quá trình tìm và sửa lỗi trong code.
Widget Test	Kiểm thử các thành phần giao diện trong ứng dụng Flutter.



Unit Test	Kiểm thử từng đơn vị nhỏ trong code để đảm bảo chúng hoạt động đúng.
Integration Testing	Kiểm thử tích hợp nhằm đảm bảo các thành phần của ứng dụng hoạt động đúng khi kết hợp với nhau.
JSON (JavaScript Object Notation)	Định dạng dữ liệu phổ biến dùng để trao đổi giữa client và server.
Pubspec.yaml	Tập tin cấu hình dự án Flutter, chứa thông tin về thư viện, asset, và dependencies.
GestureDetector	Widget giúp xử lý các tương tác cảm ứng của người dùng.
Async/Await	Cách xử lý bất đồng bộ trong Dart giúp viết code dễ đọc hơn.
Firebase	Nền tảng cung cấp các dịch vụ backend như authentication, database, cloud functions.

#### 1.4. Công nghệ sử dụng

Ứng dụng Flybox được phát triển bằng các công nghệ hiện đại nhằm đảm bảo hiệu suất, khả năng mở rộng và trải nghiệm người dùng tốt nhất.

- Ngôn ngữ lập trình
  - Dart: Ngôn ngữ lập trình hiện đại, được sử dụng trong Flutter để phát triển ứng dụng đa nền tảng.
- Framework & SDK
  - Flutter: Bộ công cụ phát triển giao diện (UI toolkit) mã nguồn mở của Google, hỗ trợ xây dựng ứng dụng di động trên nhiều nền tảng (Android, iOS) với giao diện mượt mà và hiệu năng cao.
- Quản lý trạng thái
  - Provider / Riverpod: Hỗ trợ quản lý trạng thái hiệu quả, giúp tối ưu hiệu suất và tái sử dụng mã nguồn.
- Cơ sở dữ liệu
  - CSDL/API/ Hive: Lưu trữ dữ liệu cục bộ trên thiết bị để tăng tốc độ truy xuất dữ liệu.

- Firebase Firestore: Cung cấp cơ sở dữ liệu thời gian thực và đồng bộ trên nhiều thiết bị.
- API & Networking
- Dio / HTTP: Thư viện hỗ trợ gọi API RESTful để lấy dữ liệu chuyến bay và đặt vé trực tuyến.
- Google Maps API: Tích hợp bản đồ để tìm kiếm sân bay và hiển thị đường bay.

## **2. Phân tích và Thiết kế**

### **2.1. Kiến trúc ứng dụng (MVVM, Clean Architecture, v.v.)**

MVVM (Model-View-ViewModel) là một mô hình kiến trúc phần mềm giúp tách biệt giao diện người dùng (UI) khỏi logic nghiệp vụ và dữ liệu. Kiến trúc này giúp cải thiện khả năng bảo trì, mở rộng, kiểm thử và tái sử dụng mã nguồn, đồng thời nâng cao trải nghiệm phát triển ứng dụng.

Ứng dụng Flybox áp dụng mô hình MVVM để:

- Đảm bảo tính tách biệt giữa giao diện và logic xử lý, giúp dễ dàng thay đổi UI mà không ảnh hưởng đến logic nghiệp vụ.
- Hỗ trợ kiểm thử đơn vị (Unit Test) dễ dàng hơn, đặc biệt là với ViewModel.
- Tối ưu hóa hiệu suất và khả năng mở rộng của ứng dụng

#### **2.1.1. Các thành phần chính trong MVVM**

Mô hình MVVM trong Flybox bao gồm ba thành phần chính:

##### **1) View (Giao diện người dùng)**

Chịu trách nhiệm hiển thị dữ liệu và nhận tương tác từ người dùng.

- Không chứa logic nghiệp vụ, chỉ phản hồi sự kiện từ ViewModel.
- Cập nhật UI mỗi khi có thay đổi từ ViewModel.

Trong Flybox:

- Màn hình tìm kiếm chuyến bay hiển thị danh sách kết quả lấy từ ViewModel.
- Khi người dùng nhập thông tin tìm kiếm, View gửi sự kiện đến ViewModel để xử lý.

## 2) ViewModel (Xử lý logic UI)

Là cầu nối giữa View và Model.

- Nhận sự kiện từ View, xử lý logic UI và cập nhật dữ liệu để hiển thị.
- Không chứa tham chiếu trực tiếp đến View để giảm sự phụ thuộc.
- Lấy dữ liệu từ UseCase và xử lý trước khi cập nhật lên View.

### **Trong Flybox:**

- Khi người dùng nhập điểm đi, điểm đến và ngày bay, ViewModel sẽ gửi yêu cầu tìm kiếm chuyến bay đến UseCase.
- Khi có kết quả, ViewModel cập nhật LiveData hoặc StateNotifier để View hiển thị.

## 3) Model (Dữ liệu ứng dụng)

Quản lý dữ liệu của ứng dụng, bao gồm Entity, UseCase và Repository.

- Chịu trách nhiệm truy xuất, xử lý và lưu trữ dữ liệu.
- Dữ liệu có thể được lấy từ API, cơ sở dữ liệu hoặc bộ nhớ cache.

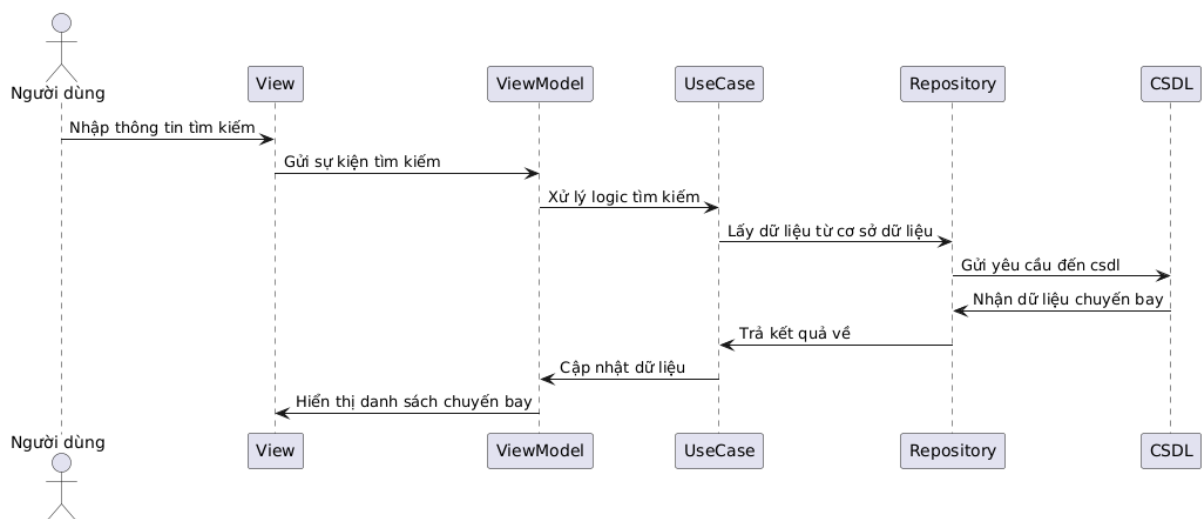
### **Trong Flybox:**

- Khi UseCase nhận yêu cầu tìm kiếm chuyến bay, nó sẽ lấy dữ liệu từ Repository.
- Sau đó, danh sách chuyến bay được gửi về ViewModel để cập nhật giao diện.

## **2.1.2. Luồng hoạt động của MVVM trong Flybox**

Quy trình hoạt động của mô hình MVVM trong ứng dụng Flybox diễn ra theo các bước sau:

1. Người dùng nhập thông tin tìm kiếm chuyến bay trong View.
2. View gửi sự kiện tới ViewModel.
3. ViewModel gọi UseCase để xử lý logic nghiệp vụ.
4. UseCase lấy dữ liệu từ Repository.
5. Repository gọi DataSource (API hoặc Database) để lấy dữ liệu.
6. Dữ liệu được trả về Repository, rồi đến UseCase, sau đó đến ViewModel.
7. ViewModel cập nhật dữ liệu mới lên View để hiển thị.



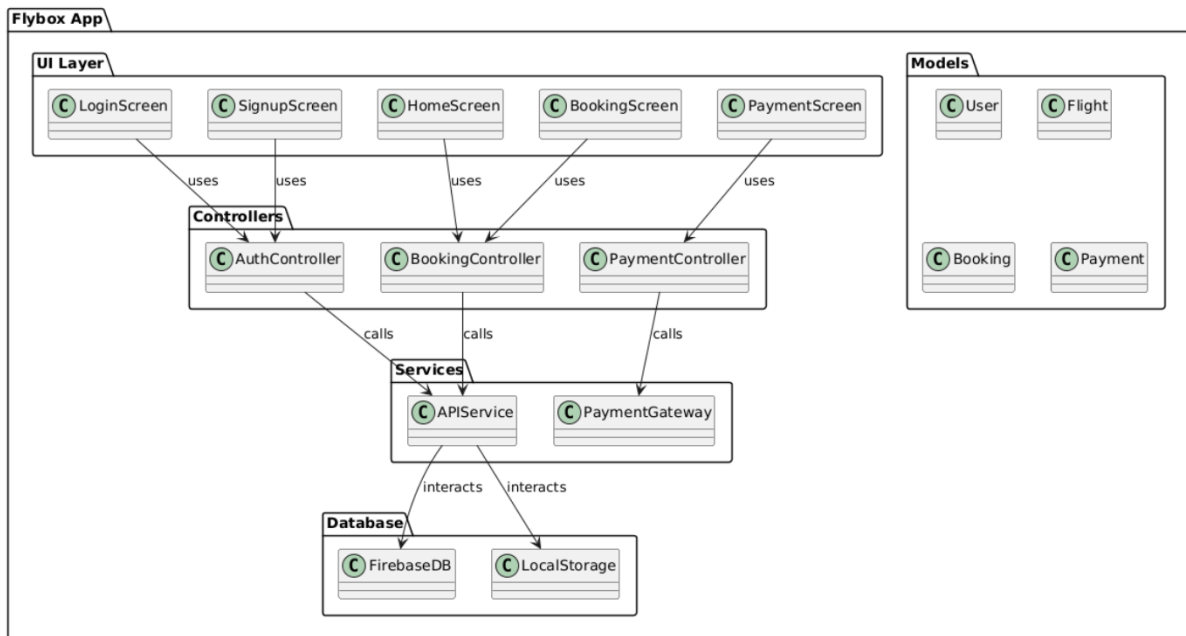
## 2.2. Luồng hoạt động chính của ứng dụng.

Ứng dụng Flybox được thiết kế nhằm hỗ trợ người dùng tìm kiếm, đặt vé máy bay và quản lý lịch trình chuyến bay một cách tiện lợi. Luồng hoạt động của ứng dụng được xây dựng dựa trên mô hình MVVM, giúp đảm bảo tính rõ ràng, dễ bảo trì và mở rộng.

Dưới đây là các luồng hoạt động chính của ứng dụng Flybox:

1. Tìm kiếm chuyến bay
2. Đặt vé máy bay
3. Quản lý vé đã đặt
4. Đăng nhập & Đăng ký
5. Thanh toán vé máy bay

Sơ đồ tổng quan hệ thống:

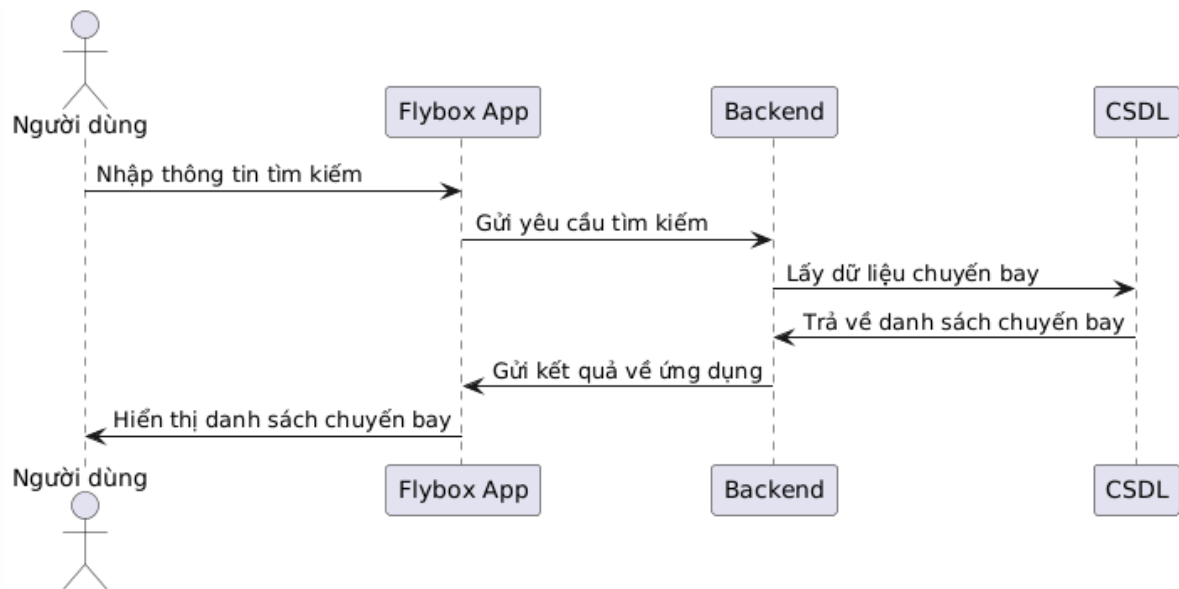


### 2.2.1. Luồng tìm kiếm chuyến bay

Mô tả: Người dùng nhập thông tin tìm kiếm (điểm đi, điểm đến, ngày bay) → Hệ thống hiển thị danh sách chuyến bay phù hợp.

Các bước thực hiện:

1. Người dùng nhập thông tin tìm kiếm chuyến bay.
  2. Ứng dụng gửi yêu cầu đến hệ thống backend.
  3. Backend xử lý yêu cầu, truy vấn dữ liệu từ API hoặc database.
  4. Kết quả danh sách chuyến bay được gửi về ứng dụng.
  5. Ứng dụng hiển thị danh sách chuyến bay phù hợp.
- Sơ đồ hoạt động (Sequence Diagram) của luồng tìm kiếm chuyến bay:

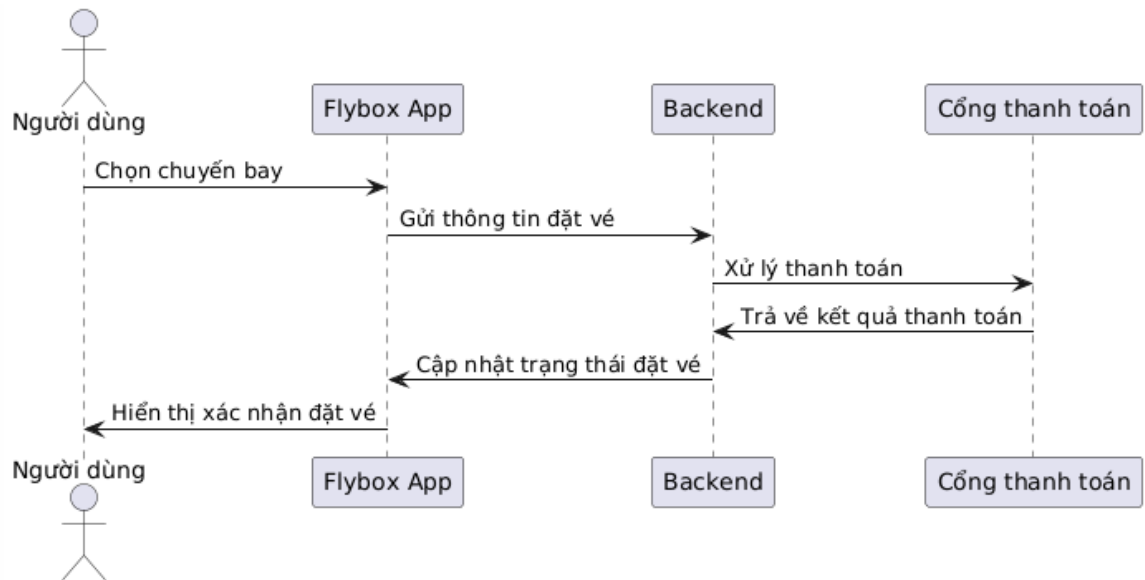


### 2.2.2. Luồng đặt vé máy bay.

Mô tả: Người dùng chọn chuyến bay → Nhập thông tin hành khách → Xác nhận đặt vé → Thanh toán.

Các bước thực hiện:

1. Người dùng chọn một chuyến bay từ danh sách.
  2. Nhập thông tin hành khách (họ tên, số CMND, email, số điện thoại).
  3. Hệ thống kiểm tra thông tin hợp lệ và hiển thị tổng giá vé.
  4. Người dùng xác nhận đặt vé.
  5. Hệ thống chuyển đến màn hình thanh toán.
  6. Sau khi thanh toán thành công, hệ thống gửi email xác nhận.
- Sơ đồ hoạt động của luồng đặt vé:

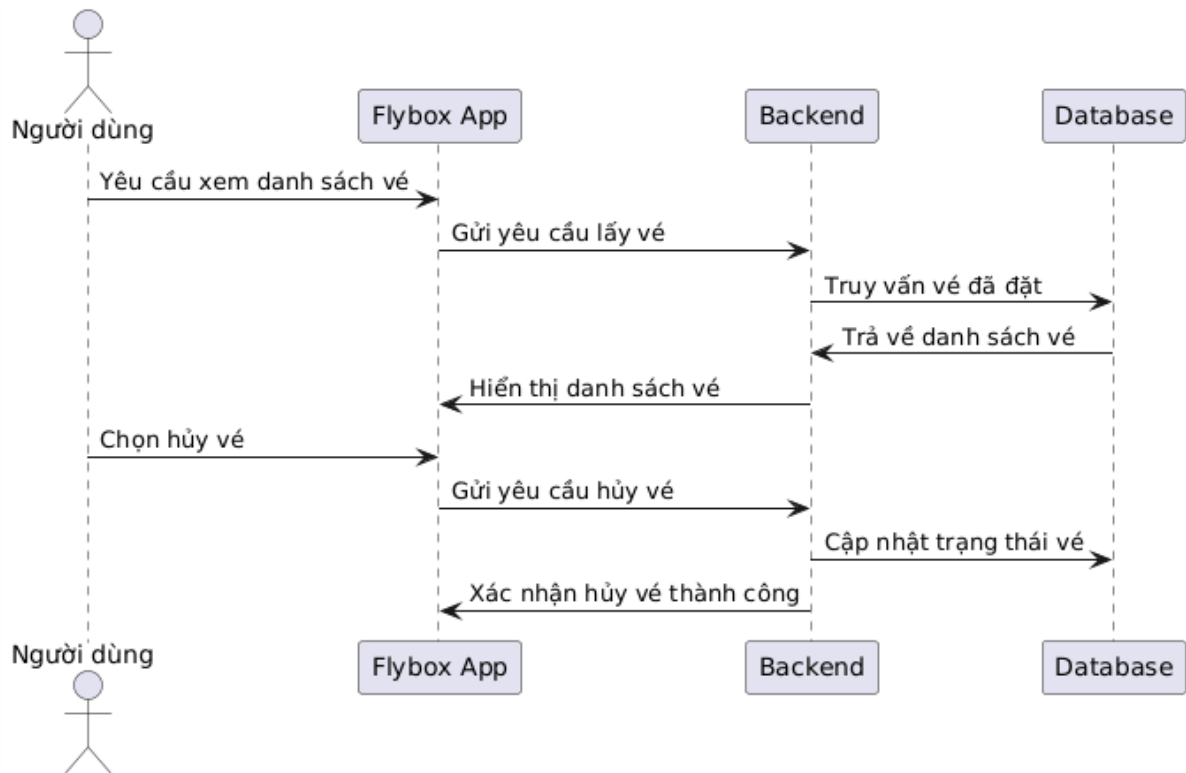


### 2.2.3. Luồng quản lý vé đã đặt.

Mô tả: Người dùng có thể xem, hủy vé hoặc kiểm tra trạng thái vé.

Các bước thực hiện:

1. Người dùng vào mục "Quản lý vé".
  2. Ứng dụng gửi yêu cầu lấy danh sách vé từ server.
  3. Backend truy vấn database và gửi kết quả về.
  4. Người dùng có thể chọn hủy vé (nếu được phép).
  5. Nếu hủy vé, hệ thống xử lý hoàn tiền theo chính sách.
- Sơ đồ hoạt động của luồng quản lý vé:



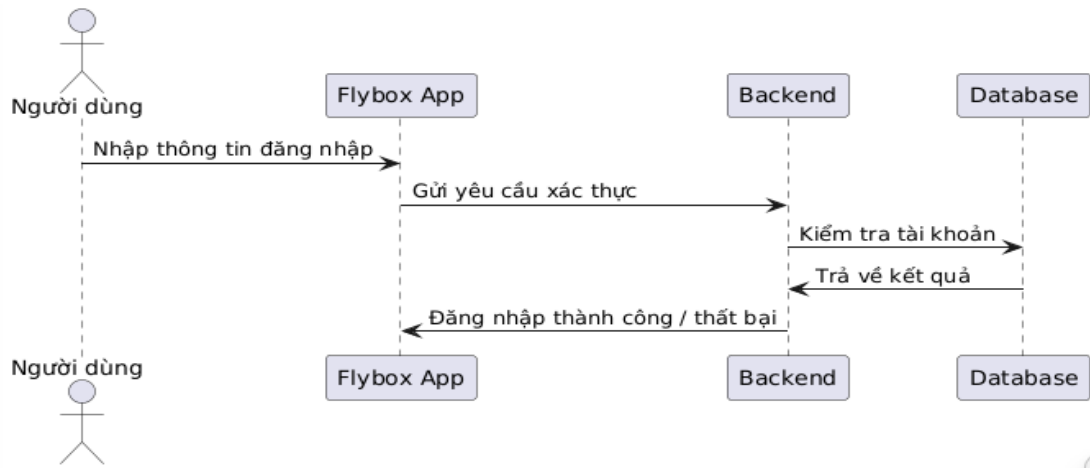
#### 2.2.4. Luồng đăng nhập và đăng ký

Mô tả: Người dùng đăng nhập hoặc đăng ký tài khoản để sử dụng đầy đủ tính năng của ứng dụng.

Các bước thực hiện:

1. Người dùng nhập thông tin đăng nhập (email/số điện thoại, mật khẩu).
  2. Ứng dụng gửi yêu cầu xác thực lên server.
  3. Backend kiểm tra thông tin và phản hồi kết quả.
  4. Nếu đăng ký, người dùng nhập thông tin cá nhân và hệ thống lưu vào database.
- Sơ đồ hoạt động của luồng đăng nhập:



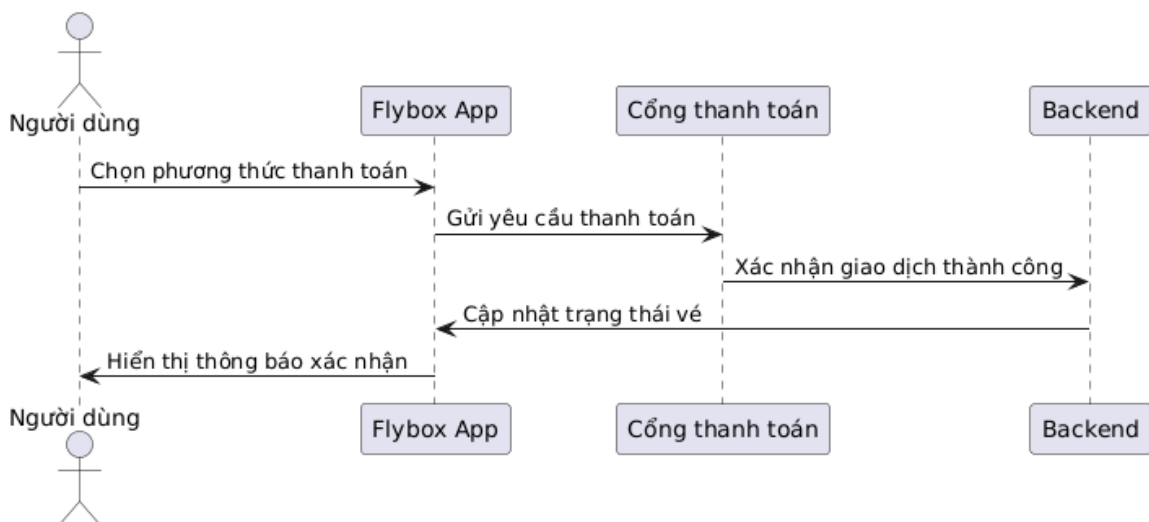


### 2.2.5. Luồng thanh toán vé máy bay

Mô tả: Người dùng thực hiện thanh toán để xác nhận đặt vé.

Các bước thực hiện:

1. Người dùng chọn phương thức thanh toán (thẻ tín dụng, ví điện tử, ngân hàng).
  2. Ứng dụng gửi yêu cầu thanh toán đến cổng thanh toán.
  3. Cổng thanh toán xử lý giao dịch và phản hồi kết quả.
  4. Nếu thanh toán thành công, hệ thống cập nhật trạng thái vé.
  5. Người dùng nhận thông báo và email xác nhận đặt vé.
- Sơ đồ hoạt động của luồng thanh toán:



## **2.3. Thiết kế giao diện người dùng (UI/UX)**

### **2.3.1. Màn hình chính**

Mô tả:

- Hiển thị ô tìm kiếm chuyến bay, giúp người dùng nhanh chóng nhập thông tin cần thiết.
- Danh sách các chuyến bay phổ biến, gợi ý dựa trên xu hướng đặt vé.
- Hiển thị địa điểm nổi bật và các ưu đãi đặc biệt từ các hãng hàng không.

### **2.3.2. Màn hình tìm kiếm chuyến bay**

Mô tả:

- Người dùng nhập:
  - Điểm đi
  - Điểm đến
  - Ngày bay
  - Số lượng hành khách
- Hệ thống tìm kiếm và hiển thị danh sách chuyến bay phù hợp.
- Bộ lọc chuyến bay theo giá, thời gian, hãng hàng không.

### **2.3.3. Màn hình đặt vé**

Mô tả:

- Hiển thị thông tin chi tiết về chuyến bay đã chọn.
- Người dùng nhập thông tin hành khách, bao gồm:
  - Họ tên
  - Số hộ chiếu / CMND
  - Thông tin liên hệ
- Người dùng xác nhận và chuyển đến bước thanh toán.

### **2.3.4. Màn hình thanh toán**

Mô tả:

- Người dùng chọn phương thức thanh toán:
  - Thẻ tín dụng / thẻ ghi nợ
  - Ví điện tử (Momo, ZaloPay, PayPal)
  - Chuyển khoản ngân hàng
- Nhập thông tin thanh toán và xác nhận.
- Hiện thị trạng thái thanh toán: Thành công hoặc thất bại.
- Khi thanh toán thành công, hệ thống gửi vé điện tử qua email.

### **2.3.5. Màn hình quản lý vé**

Mô tả:

- Hiện thị danh sách vé đã đặt, bao gồm:
  - Chuyến bay
  - Ngày giờ khởi hành
  - Trạng thái vé
- Người dùng có thể xem chi tiết vé, bao gồm mã đặt chỗ.
- Cung cấp tùy chọn hủy vé nếu cần.
- Hiện thị trạng thái vé:
  - Đã thanh toán
  - Chờ thanh toán
  - Đã hủy

## **3.Chức năng chính của ứng dụng**

### **3.1. Đăng ký, đăng nhập, đăng xuất**

Mô tả: Ứng dụng Flybox cung cấp các chức năng đăng ký, đăng nhập và đăng xuất để quản lý tài khoản người dùng. Người dùng có thể tạo tài khoản mới, đăng nhập để sử dụng các tính năng nâng cao như đặt vé, quản lý lịch trình, và nhận thông báo chuyến bay.

#### **3.1.1. Chức năng Đăng ký (Sign Up)**

Người dùng nhập thông tin cá nhân:

- Họ và tên
- Địa chỉ email
- Mật khẩu
- Số điện thoại (tùy chọn)

Hệ thống kiểm tra tính hợp lệ của thông tin.

Nếu hợp lệ, tài khoản được tạo và lưu vào cơ sở dữ liệu.

Hệ thống gửi email xác nhận tài khoản.

Sau khi xác nhận, người dùng có thể đăng nhập vào ứng dụng.



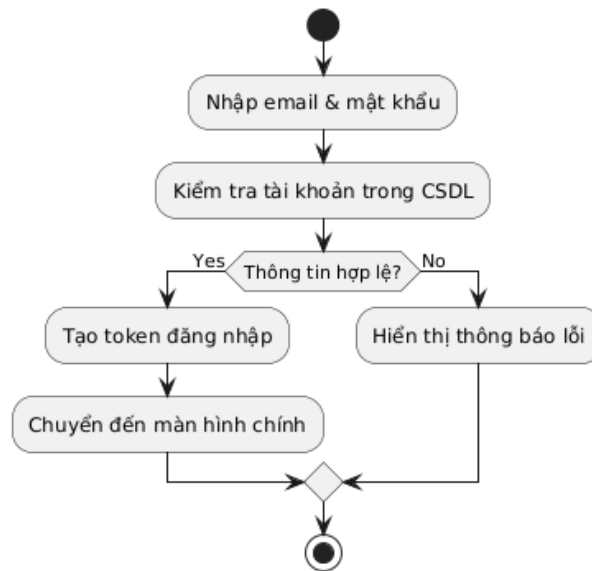
### 3.1.2. Chức năng đăng nhập (Login)

Người dùng nhập email và mật khẩu.

Hệ thống kiểm tra thông tin đăng nhập:

- Nếu hợp lệ → Cho phép truy cập vào ứng dụng.
- Nếu sai mật khẩu → Hiển thị thông báo lỗi.
- Nếu tài khoản chưa kích hoạt → Yêu cầu xác nhận email.

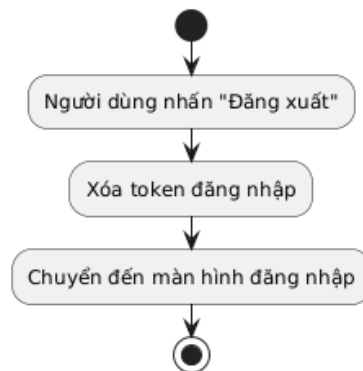
Khi đăng nhập thành công, hệ thống tạo token xác thực (JWT hoặc Firebase Auth) để duy trì phiên đăng nhập.



### 3.1.3. Chức năng Đăng xuất (Logout)

Khi người dùng nhấn "Đăng xuất", hệ thống sẽ:

- Xóa token xác thực khỏi bộ nhớ thiết bị.
- Chuyển người dùng về màn hình đăng nhập.

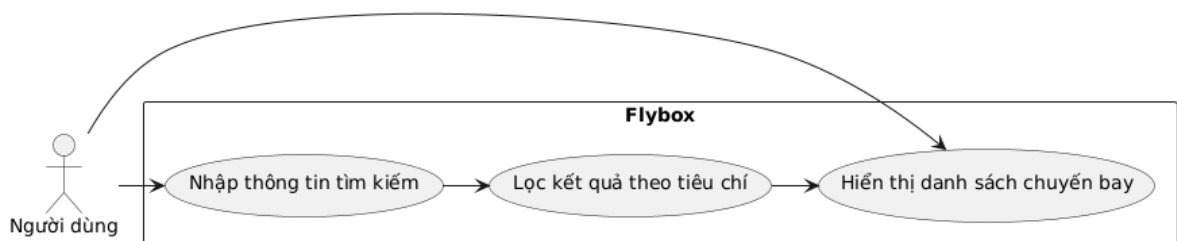


## 3.2. Tìm kiếm và đặt vé chuyến bay

Chức năng Tìm kiếm và Đặt vé chuyến bay giúp người dùng dễ dàng tìm thấy các chuyến bay phù hợp dựa trên điểm đi, điểm đến, ngày bay, hãng hàng không, giá vé, v.v. Sau khi chọn chuyến bay ưng ý, người dùng có thể tiến hành đặt vé bằng cách nhập thông tin hành khách và xác nhận thanh toán.

### 3.2.1. Chức năng tìm kiếm chuyến bay

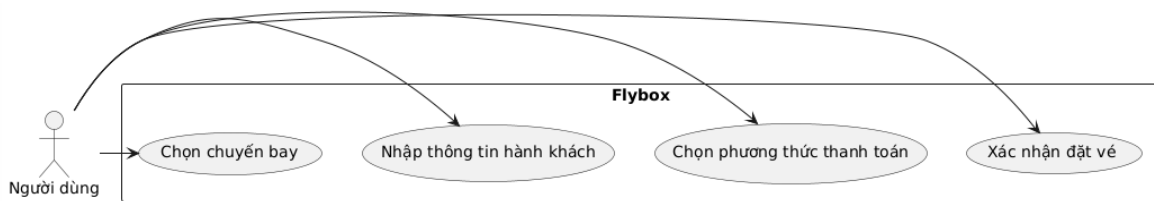
- Người dùng nhập các thông tin cần thiết để tìm kiếm chuyến bay:
  - Điểm đi (Sân bay khởi hành)
  - Điểm đến (Sân bay đến)
  - Ngày bay (Ngày khởi hành)
  - Số lượng hành khách
  - Loại vé (Phổ thông, Thương gia, Hạng nhất)
- Hệ thống kiểm tra thông tin nhập vào.
- Gửi yêu cầu tìm kiếm đến CSDL chuyến bay.
- Nhận kết quả và hiển thị danh sách các chuyến bay phù hợp.
- Người dùng có thể lọc và sắp xếp kết quả theo giá, thời gian bay, hãng hàng không, v.v.



### 3.2.2. Chức năng đặt vé chuyến bay

- Người dùng chọn một chuyến bay từ danh sách kết quả tìm kiếm.
- Hệ thống hiển thị thông tin chi tiết của chuyến bay, bao gồm:
  - Hãng hàng không
  - Giờ khởi hành, giờ đến
  - Thời gian bay
  - Giá vé và các loại thuế phí
- Người dùng nhập thông tin hành khách:
  - Họ và tên
  - Số CMND/Passport
  - Ngày sinh
  - Số điện thoại, email

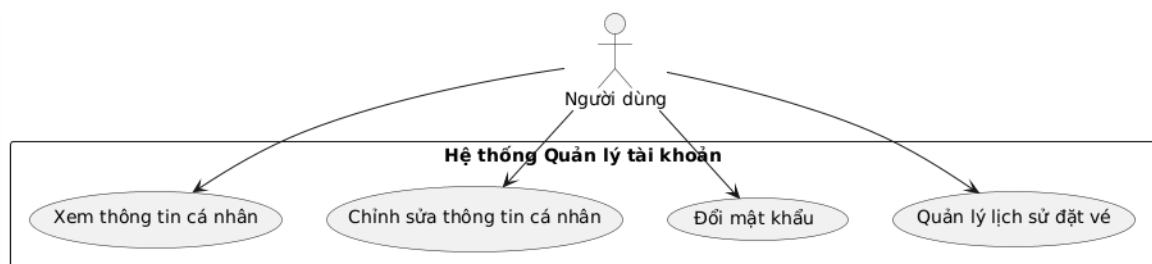
- Kiểm tra thông tin và hiển thị tổng chi phí.
- Người dùng xác nhận và chuyển đến bước thanh toán.



### 3.3. Quản lý thông tin cá nhân

Chức năng quản lý thông tin cá nhân cho phép người dùng cập nhật và quản lý các thông tin cá nhân trên hệ thống. Người dùng có thể chỉnh sửa hồ sơ, thay đổi mật khẩu, cập nhật phương thức thanh toán, và kiểm tra lịch sử giao dịch.

- Xem thông tin cá nhân
  - Hiển thị thông tin cá nhân của người dùng (Tên, Email, Số điện thoại, Ngày sinh, Giới tính).
- Chỉnh sửa thông tin cá nhân
  - Cập nhật họ tên, số điện thoại, địa chỉ.
  - Cập nhật phương thức thanh toán mặc định.
- Đổi mật khẩu
  - Xác nhận mật khẩu cũ.
  - Nhập mật khẩu mới và xác nhận lại.
- Quản lý lịch sử đặt vé
  - Xem danh sách vé đã đặt.
  - Kiểm tra chi tiết vé (trạng thái thanh toán, thông tin chuyến bay).



### 3.4. Thanh toán vé máy bay

Người dùng chọn một chuyến bay từ danh sách kết quả tìm kiếm.

Hệ thống hiển thị thông tin chi tiết của chuyến bay, bao gồm:

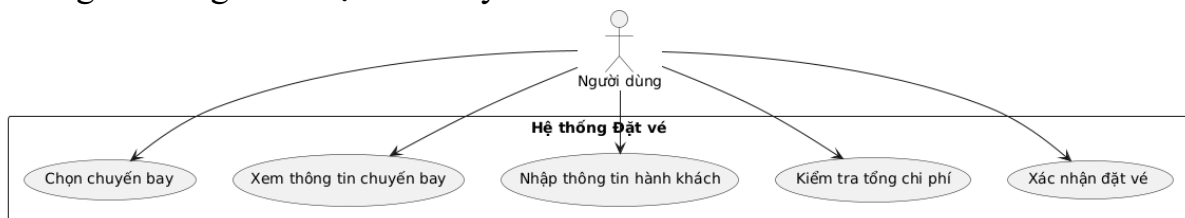
- Hãng hàng không
- Giờ khởi hành, giờ đến
- Thời gian bay
- Giá vé và các loại thuế phí

Người dùng nhập thông tin hành khách:

- Họ và tên
- Số CMND/Passport
- Ngày sinh
- Số điện thoại, email

Hệ thống kiểm tra thông tin và hiển thị tổng chi phí.

Người dùng xác nhận và chuyển đến bước thanh toán.



### 3.5. Quản lý danh sách chuyến bay đã đặt

Người dùng truy cập vào danh sách các chuyến bay đã đặt.

Hệ thống hiển thị danh sách các vé máy bay, bao gồm:

- Mã đặt chỗ
- Hãng hàng không
- Thời gian khởi hành và đến
- Trạng thái vé (đã thanh toán, chờ thanh toán, đã hủy)

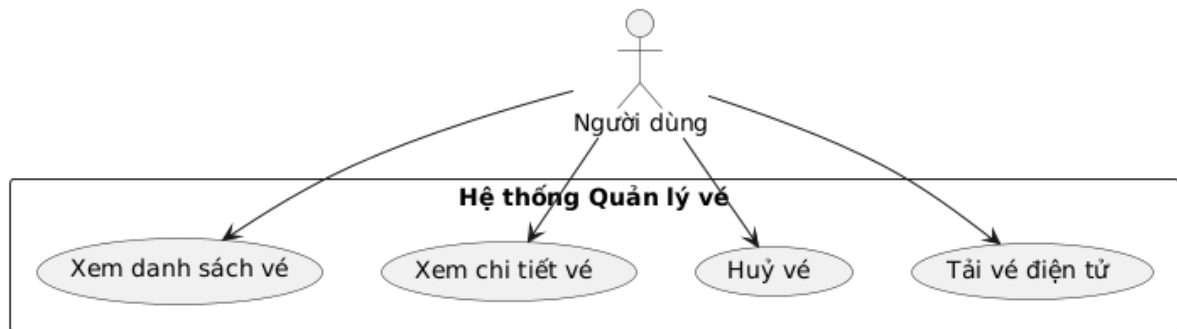
Người dùng chọn một vé để xem chi tiết.

Người dùng có thể thực hiện các thao tác sau:

- Xem thông tin vé
- Hủy vé (nếu được phép)
- Tải vé điện tử

Hệ thống cập nhật thông tin sau mỗi thao tác.





## 4. Triển khai và Kiểm thử

### 4.1. Môi trường phát triển và triển khai

#### 4.1.1 Môi trường phát triển

-Công cụ cần thiết:

- Flutter SDK: Bộ công cụ phát triển ứng dụng Flutter.
- Dart SDK: Ngôn ngữ lập trình Dart đi kèm với Flutter.
- IDE: Android Studio.
- Trình giả lập / thiết bị thực:

Android Emulator (cho Android).

iOS Simulator (cho iOS, yêu cầu macOS).

Thiết bị thật (để kiểm thử thực tế).

-Thiết lập môi trường: gyy7y

- Cài đặt Flutter từ trang chính thức: <https://docs.flutter.dev/get-started/install>
- Kiểm tra môi trường bằng lệnh:

```
flutter doctor
```

- Tạo project flutter:

```
flutter create flight_booking_app
```

- Chạy ứng dụng thử nghiệm:

```
flutter run
```

#### 4.1.2 Môi trường triển khai

- Backend API: Firebase, Node.js hoặc một hệ thống backend khác.

- Cơ sở dữ liệu: Firebase Firestore hoặc SQLite.
- Hosting:
  - Google Play Store (Android).
  - Apple App Store (iOS).
  - Web App (Flutter Web).
- CI/CD (Tích hợp & triển khai liên tục): Codemagic hoặc GitHub Actions để tự động build & deploy.

## 4.2. Các công cụ kiểm thử (Unit test, Widget test, Integration test)

### 4.2.1 Kiểm thử đơn vị (Unit test)

- Flutter DevTools:
  - Công cụ hỗ trợ kiểm tra hiệu suất, logs và lỗi mà không cần viết test case.
  - Chạy lệnh `flutter run --debug` và mở **DevTools** để xem lỗi.
- Firebase Test Lab:
  - Chạy test trên nhiều thiết bị thực mà không cần code.
  - Upload file APK/AAB và Firebase tự động kiểm thử.

### 4.2.2 Kiểm thử giao diện ( widget Test)

- Test trên thiết bị/thử nghiệm thủ công
  - Dùng máy ảo để kiểm tra giao diện
  - Kiểm tra các nút, hình ảnh văn bản hiển thị đúng không
- Appium (UI Test tự động):
  - Công cụ kiểm thử UI mà không cần viết mã Flutter.
  - Dùng script hoặc thao tác trực tiếp để kiểm tra giao diện

### 4.2.3 Kiểm thử tích hợp (Integration Test)

- TestCaseLab hoặc TestRail:
  - Viết test case bằng giao diện web
  - Theo dõi và ghi lại lỗi phát sinh khi kiểm thử.
- Firebase Test Lab hoặc BrowserStack App Automate:
  - Tải ứng dụng lên cloud, hệ thống tự động kiểm thử trên nhiều thiết bị.

## 4.3. Debugging và tối ưu hiệu suất ứng dụng

### 4.3.1 Debugging - Tìm và sửa lỗi hiệu quả

- a. Sử dụng DevTools để kiểm tra ứng dụng

- Chạy lệnh sau để mở Flutter DevTools:

```
flutter pub global activate devtools  
flutter run --debug
```

- Các công cụ chính:
  - **Widget Inspector**: Kiểm tra cấu trúc UI.
  - **Memory & CPU Profiler**: Giám sát bộ nhớ và CPU.
  - **Network Monitor**: Theo dõi API gọi đến backend.

b. Xử lý lỗi (Debugging)

- Bắt lỗi khi gọi API bằng **try-catch**:

```
try {  
    final response = await fetchFlights();  
} catch (e) {  
    print('Lỗi khi lấy dữ liệu chuyến bay: $e');  
}
```

- Kiểm tra lỗi bằng lệnh:

```
flutter logs
```

#### 4.3.2 Tối ưu hiệu suất

a. Giảm thời gian render UI

- Dùng **const** để tối ưu widget tĩnh:

```
const Text('Chào mừng bạn!')
```

=> Tránh re-build lại widget không cần thiết.

- Dùng **ListView.builder()** thay vì **ListView** thông thường:

```
ListView.builder(  
    itemCount: flights.length,  
    itemBuilder: (context, index) {  
        return FlightCard(flight: flights[index]);  
    },  
);
```

=> Giúp hiển thị danh sách mượt mà hơn

b. Tối ưu ảnh và tài nguyên

Dùng thư viện `cached_network_image` để load ảnh nhanh hơn:

```
CachedNetworkImage(imageUrl: flight.imageUrl)
```

=> Tránh tải lại ảnh mỗi lần mở ứng dụng.

c. Giảm kích thước file cài đặt

Build app với chế độ Release thay vì Debug

```
flutter build apk --release  
flutter build appbundle --release
```

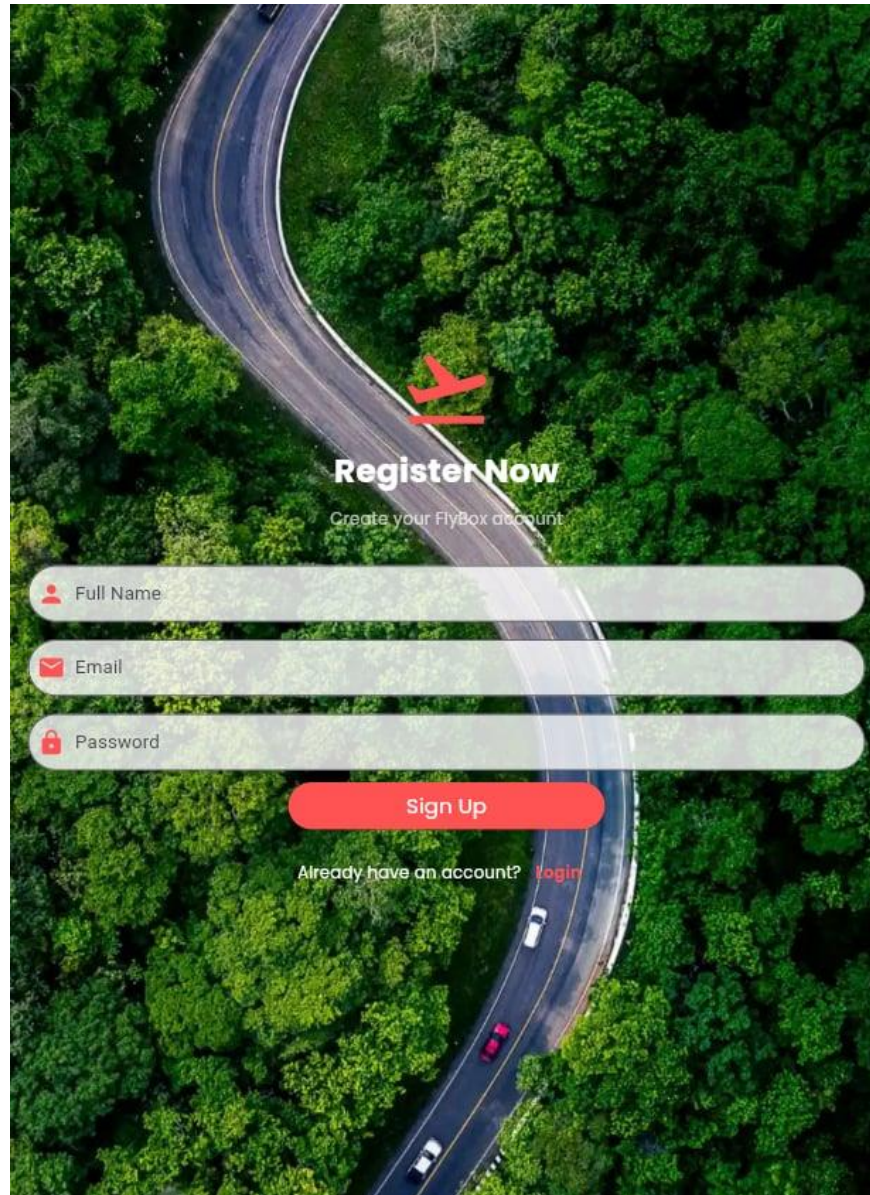
=> Giúp giảm dung lượng file APK và tăng tốc độ chạy ứng dụng.

## 5. Báo cáo thực nghiệm

### 5.1 Đăng kí tài khoản

Luồng chính:

1. User truy cập trang đăng ký.
2. Nhập thông tin tài khoản bao gồm email, mật khẩu và thông tin cá nhân.
3. Hệ thống kiểm tra tính hợp lệ của thông tin.
4. Nếu hợp lệ, hệ thống lưu thông tin tài khoản vào cơ sở dữ liệu.
5. Hệ thống gửi email xác thực tài khoản.
6. User xác nhận email để kích hoạt tài khoản.
7. Hệ thống cập nhật trạng thái tài khoản thành "Đã kích hoạt".
8. User có thể đăng nhập vào hệ thống.

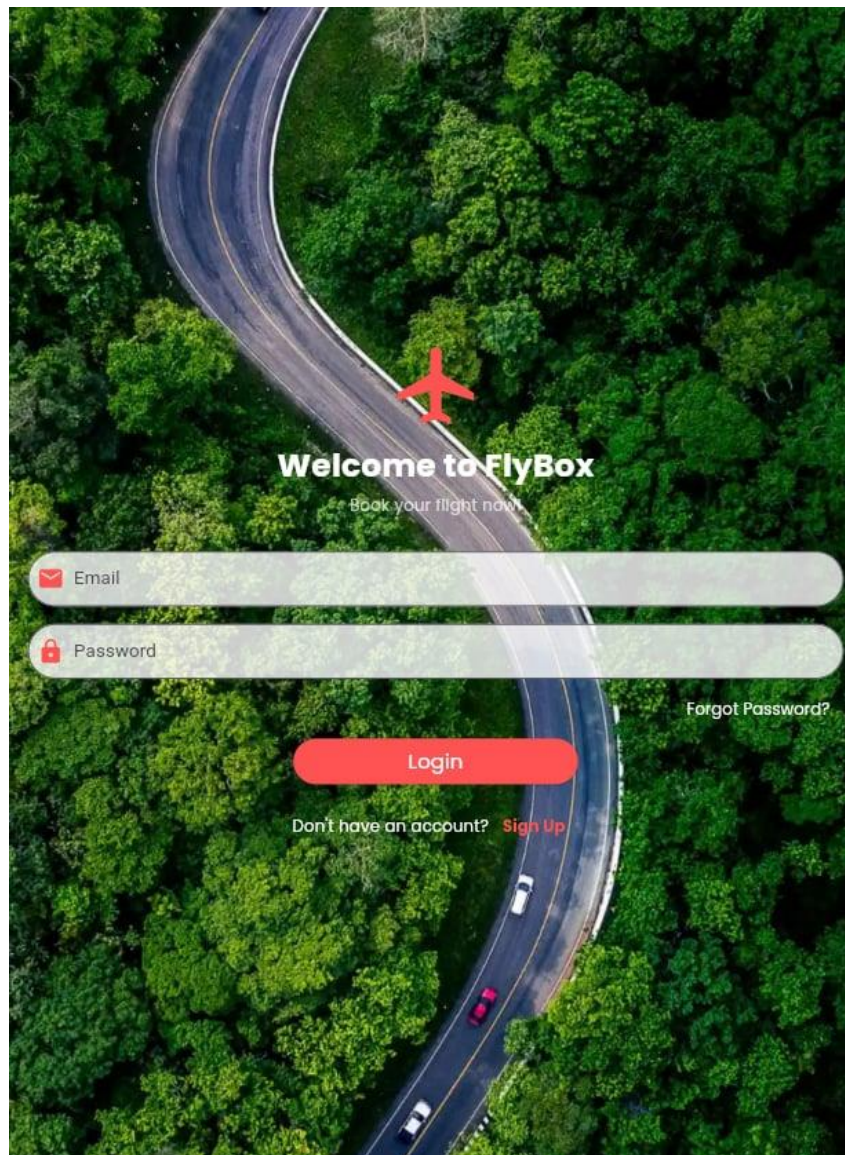


## 5.2 Đăng nhập tài khoản

Luồng chính:

1. Người dùng truy cập trang **Đăng nhập**.
2. Nhập **Email/Số điện thoại** và **Mật khẩu**.
3. Nhấn nút **Đăng nhập**.
4. Hệ thống kiểm tra thông tin:
  - Nếu hợp lệ → Điều hướng đến trang chủ hoặc trang cá nhân.
  - Nếu sai → Hiển thị thông báo lỗi.

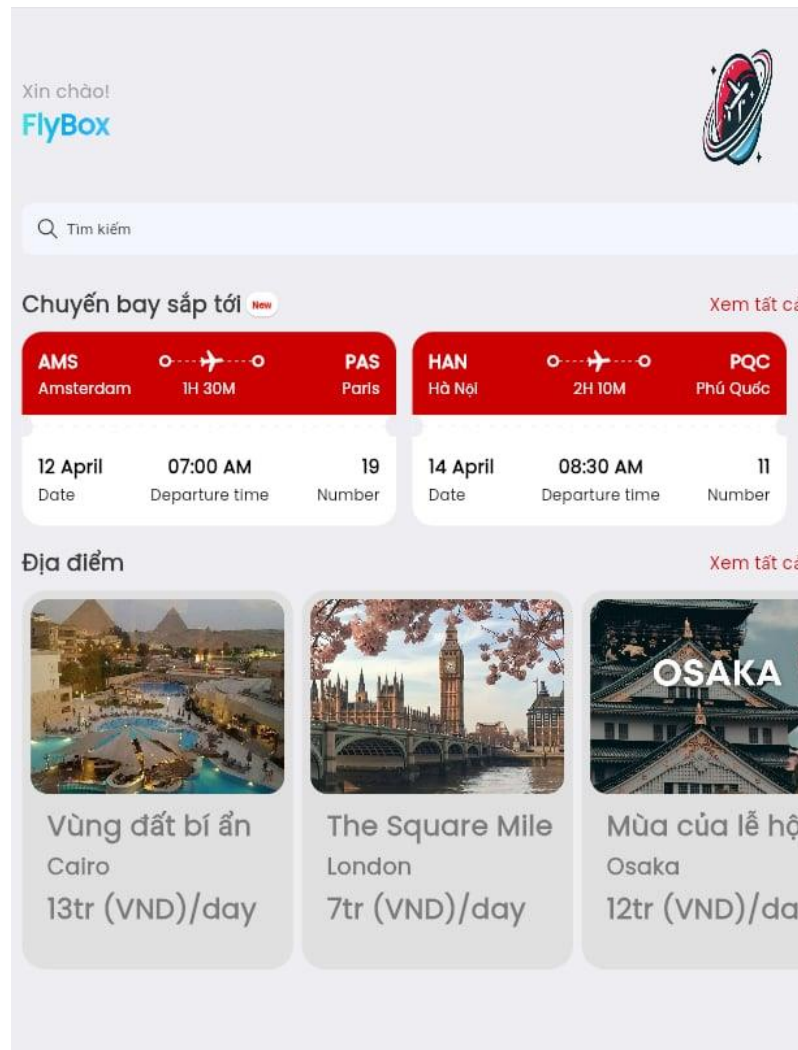




### 5.3 Trang chủ

**FlyBox** là nền tảng đặt vé máy bay trực tuyến. Kết nối người dùng với các chuyến bay nội địa và quốc tế.

Cung cấp các tính năng: tìm kiếm, đặt vé, thanh toán online, quản lý đặt chỗ. Hỗ trợ người dùng theo dõi chuyến bay và nhận thông báo kịp thời.



## 5.4 Tìm kiếm chuyến bay

Chức năng này giúp người dùng tìm kiếm các chuyến bay phù hợp theo nhu cầu của họ dựa trên thông tin đầu vào như điểm khởi hành, điểm đến.

1. Người dùng nhập thông tin tìm kiếm
  - Nhập điểm khởi hành.
  - Nhập điểm đến.
2. Người dùng nhấn nút "Tìm kiếm"
  - Nút này có màu đỏ, nằm bên dưới các ô nhập dữ liệu.
3. Hệ thống xử lý yêu cầu
  - Kiểm tra xem thông tin có đầy đủ không.
  - Nếu hợp lệ, gửi truy vấn đến máy chủ để lấy danh sách chuyến bay.
4. Hiển thị kết quả chuyến bay
  - Nếu tìm thấy chuyến bay phù hợp → Hiển thị danh sách các chuyến bay phù hợp.

- Nếu không có chuyến bay → Hiển thị thông báo "Không có chuyến bay nào phù hợp".

Chuyến bay bạn tìm kiếm

Đặt vé

Địa điểm


✈️ Khởi hành

✈️ Điểm đến

Tìm kiếm

Chuyến bay sắp tới


xem tất cả



Mừng sinh nhật 1 năm, giảm ngay 20%. Đừng bỏ lỡ!

Ưu đãi thành viên mới, đăng kí ngay thôi!

Sách báo và Flybox ngay thôi!



Đà Lạt gì chưa người đẹp!

## 5.5 Thanh toán và xác nhận đặt vé

Luồng chính:

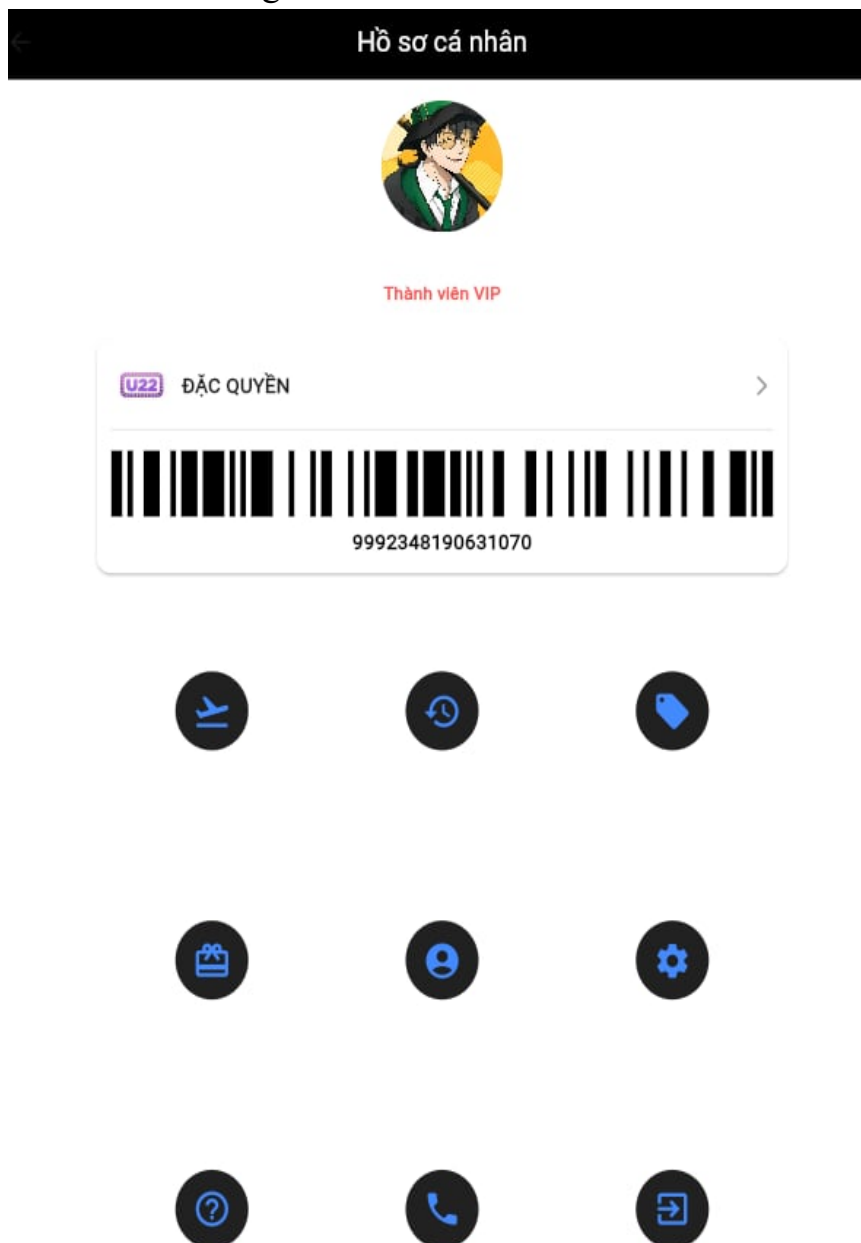
1. Chọn phương thức thanh toán (thẻ tín dụng, ví điện tử, chuyển khoản).
2. Nhập thông tin thanh toán.
3. Xác nhận thanh toán.
4. Hệ thống xử lý giao dịch.
5. Nếu thành công → Hiển thị trang xác nhận đặt vé và gửi email vé điện tử.

## 5.6 Hồ sơ cá nhân (Profile)

Trang **Hồ sơ cá nhân** trong ứng dụng đặt vé máy bay cung cấp thông tin cá nhân của người dùng và các tùy chọn quản lý tài khoản.

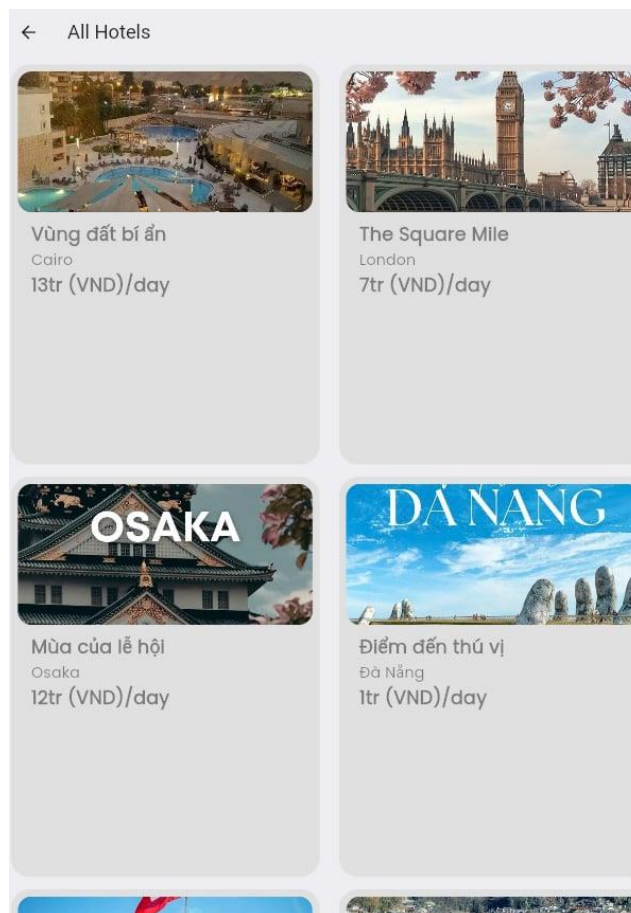


- Hiển thị **ảnh đại diện**, trạng thái thành viên (**VIP**).
- Có mã **barcode** hoặc **QR code** để xác thực quyền lợi thành viên.
- Danh sách các tính năng, gồm:
  - Thông tin chuyến bay (chuyến bay sắp tới, lịch sử đặt vé).
  - Lịch sử giao dịch.
  - Ưu đãi hoặc mã giảm giá dành riêng cho tài khoản.
  - Cài đặt tài khoản.
  - Hỗ trợ & Liên hệ.
  - Đăng xuất.



## 5.7 Thông tin vé

- Trang "All Hotels" trong ứng dụng đặt vé máy bay hiển thị danh sách các khách sạn tại nhiều địa điểm khác nhau trên thế giới.
- Mỗi khách sạn được hiển thị dưới dạng một thẻ thông tin, bao gồm:
  - Hình ảnh khách sạn
  - Tên khách sạn
  - Địa điểm (thành phố/quốc gia)
  - Giá thuê theo ngày
- Luồng chính:
  1. Người dùng truy cập trang “All Hotels”
  2. Người dùng chọn một khách sạn trong danh sách đã được hiển thị
  3. Xem chi tiết (hình ảnh, đánh giá, tiện nghi, giá phòng,...) và đặt phòng



## 5.8 Giỏ hàng

- **Hiển thị danh sách vé đã đặt:** Danh sách các vé máy bay mà người dùng đã chọn trước khi thanh toán.

- **Xóa vé khỏi giỏ hàng:** Cho phép người dùng loại bỏ vé không mong muốn trước khi thanh toán.
- **Luồng chính:**
  1. Người dùng kiểm tra giỏ hàng
    - Người dùng truy cập vào trang Giỏ hàng.
    - Ứng dụng hiển thị danh sách vé đã chọn.
    - Người dùng có thể xem chi tiết thông tin từng vé.
  2. Người dùng xóa vé khỏi giỏ
    - Người dùng nhấn vào nút "Xóa vé" của một vé cụ thể.
    - Hệ thống hiển thị xác nhận xóa vé (tùy thuộc vào thiết kế).
    - Nếu xác nhận, hệ thống xóa vé khỏi giỏ hàng.
  3. Người dùng tiếp tục thanh toán
    - Sau khi kiểm tra giỏ hàng, người dùng nhấn vào nút "Thanh toán" (nếu có).
    - Hệ thống chuyển hướng đến trang thanh toán.

#### Giỏ hàng

The screenshot shows a shopping cart titled "Giỏ hàng" containing two items. Each item is represented by a card with the following details:

- Item 1:**
  - Origin: AMS (Amsterdam)
  - Destination: PAS (Paris)
  - Flight Time: 1H 30M
  - Date: 12 April
  - Departure time: 07:00 AM
  - Number: 19
  - Seat code: Mã đặt chỗ: 19
  - QR code: A square QR code for verification.
  - Action: Xóa vé (Cancel ticket)
- Item 2:**
  - Origin: HAN (Hà Nội)
  - Destination: PQC (Phú Quốc)
  - Flight Time: 2H 10M
  - Date: 14 April
  - Departure time: 08:30 AM
  - Number: 11
  - Seat code: Mã đặt chỗ: 11
  - QR code: A square QR code for verification.
  - Action: Xóa vé (Cancel ticket)

## 6. Kết luận và đánh giá

### 6.1. Kết quả đạt được

- Hoàn thành ứng dụng đặt vé máy bay với các tính năng chính như:
  - Đặt vé, hủy vé, xem lịch sử giao dịch.
  - Tìm kiếm và lọc chuyến bay theo nhu cầu.
  - Đăng nhập và quản lý tài khoản cá nhân.
- Ứng dụng hoạt động ổn định trên nhiều thiết bị và nền tảng khác nhau.
- Các bước kiểm thử đã đảm bảo chất lượng và hiệu suất ứng dụng tốt

### 6.2. Hạn chế và hướng phát triển trong tương lai

- Hạn chế:
  - Chưa hỗ trợ thanh toán trực tuyến.
  - Hiệu suất trên một số thiết bị cũ chưa tối ưu hoàn toàn.
  - Chưa tích hợp AI để gợi ý chuyến bay phù hợp với người dùng.
- Hướng phát triển:
  - Bổ sung tính năng thanh toán qua ví điện tử và thẻ ngân hàng.
  - Cải thiện hiệu suất trên các thiết bị cũ bằng cách tối ưu code và hình ảnh.
  - Nâng cấp giao diện người dùng để tăng trải nghiệm.
  - Tích hợp AI để phân tích dữ liệu đặt vé và đưa ra gợi ý phù hợp.

### 6.3. Bài học kinh nghiệm

- Hiểu rõ quy trình phát triển ứng dụng Flutter từ thiết kế, lập trình đến kiểm thử và triển khai.
- Tầm quan trọng của kiểm thử: Giúp phát hiện lỗi sớm, cải thiện chất lượng sản phẩm.
- Cách tối ưu hiệu suất ứng dụng, giúp giảm thời gian load và tăng trải nghiệm người dùng.
- Làm việc nhóm hiệu quả: Sử dụng Git để quản lý mã nguồn, tổ chức công việc tốt hơn.
- Nâng cao kỹ năng phân tích yêu cầu: Xác định rõ nhu cầu của người dùng và chuyển hóa thành chức năng cụ thể trong ứng dụng.

## 7. Tài liệu tham khảo

<https://chatgpt.com>

<https://gemini.google.com/app>

Tài liệu môn lập trình cho thiết bị di động (Đại học Phenikaa).

## **8. Link slide và github**

**Link slide:**

<https://www.canva.com/design/DAGgQc1mWH4/aKprwt9gBeEY7R7W4N84nQ/edit?ui=eyJEljp7IlAiOnsiQiI6ZmFsc2V9fX0>

**Link github:**

<https://github.com/nguyentu-926/Flybox>