

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM**

**ĐỒ ÁN LẬP TRÌNH DI ĐỘNG**

**ỨNG DỤNG BOOKING KHÁCH SẠN SỬ DỤNG FLUTTER**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **LẬP TRÌNH DI ĐỘNG**

Giảng viên hướng dẫn :Lê Nhật Tùng

Sinh viên thực hiện :

Phan Thanh Tú MSSV: 2180608173 Hoàng Lê Hữu Châu MSSV: 2180608776

Trần Mai Trung Nguyên MSSV: 2180607812

Lớp: 21DTHD5

TP. Hồ Chí Minh, 2024

**MỤC LỤC**

[Chương I. TỔNG QUAN 4](#_Toc186846446)

[1 Nhiệm vụ đồ án 4](#_Toc186846447)

[Chương II. CƠ SỞ LÝ THUYẾT: 5](#_Toc186846450)

[1 Tổng quan công nghệ 5](#_Toc186846451)

[1.1 Ngôn ngữ lập trình Dart 5](#_Toc186846452)

[1.2 Ngôn ngữ lập trình Flutter 7](#_Toc186846453)

[2 Xác định yêu cầu 7](#_Toc186846456)

[2.1 Yêu cầu chức năng 7](#_Toc186846457)

[2.2 Yêu cầu phi chức năng 8](#_Toc186846458)

[3 Thiết kế mô hình hóa 9](#_Toc186846459)

[3.1 Mô hình erd 9](#_Toc186846460)

[3.2 Mô hình use case 10](#_Toc186846461)

[3.3 Mô hình activity diagram 11](#_Toc186846462)

[Chương III. KẾT QUẢ THỰC NGHIỆM 16](#_Toc186846463)

[1 Giao diện trang chủ của user 16](#_Toc186846464)

[2 Giao diện đăng nhập 16](#_Toc186846465)

[3 Giao diện đăng ký 18](#_Toc186846466)

[4 Giao diện danh sách yêu thích 19](#_Toc186846467)

[5 Giao diện thông tin 20](#_Toc186846468)

[6 Giao diện chi tiết khách sạn 20](#_Toc186846469)

[7 Giao diện thanh toán 21](#_Toc186846470)

[8 Giao diện bình luận 21](#_Toc186846471)

[9 Giao diện nhắn tin hỗ trợ 22](#_Toc186846472)

[10 Chức năng tìm kiếm khách sạn 22](#_Toc186846473)

[Chương IV: KẾT LUẬN VÀ KIẾN NGHỊ 23](#_Toc186846474)

[1 Kết Luận Chung 23](#_Toc186846475)

[2 Phương hướng phát triển 24](#_Toc186846476)

[Tài liệu tham khảo 25](#_Toc186846477)

**Danh mục hình ảnh**

[Hình 1 Ngôn ngữ lập trình Dart 4](#_Toc186846423)

[Hình 2 Ngôn ngữ lập trình Flutter 6](#_Toc186846424)

[Hình 3: Mô hình erd 8](#_Toc186846425)

[Hình 4: Mô hình của người dùng 9](#_Toc186846426)

[Hình 5: Mô hình của Admin 10](#_Toc186846427)

[Hình 6: Tìm kiếm khách sạn 10](#_Toc186846428)

[Hình 7: Người dùng chọn một khách sạn và xem thông tin chi tiết về các phòng trong khách sạn 11](#_Toc186846429)

[Hình 8: Người dùng đặt phòng 11](#_Toc186846430)

[Hình 9: Người dùng thanh toán 12](#_Toc186846431)

[Hình 10: Người dùng xem lịch sử đặt phòng 12](#_Toc186846432)

[Hình 11: Người dùng đánh giá và nhận xét 13](#_Toc186846433)

[Hình 12: Quản trị viên quản lý khách sạn 13](#_Toc186846434)

[Hình 13: Quản trị viên quản lý phòng 14](#_Toc186846435)

[Hình 14: Giao diện trang chủ 15](#_Toc186846436)

[Hình 15: Giao diện đăng nhập 16](#_Toc186846437)

[Hình 16: Giao diện đăng ký 17](#_Toc186846438)

[Hình 17: Giao diện danh sách yêu thích 18](#_Toc186846439)

[Hình 18: Giao diện thông tin (Profile) 19](#_Toc186846440)

[Hình 19: Giao diện chi tiết khách sạn 19](#_Toc186846441)

[Hình 20: Giao diện thanh toán 20](#_Toc186846442)

[Hình 21:Giao diện bình luận và đánh giá 20](#_Toc186846443)

[Hình 22: Giao diện nhắn tin hỗ trợ 21](#_Toc186846444)

[Hình 23: Chức năng tìm kiếm khách sạn 21](#_Toc186846445)

# Chương I. TỔNG QUAN

## Nhiệm vụ đồ án

Ứng dụng đặt phòng khách sạn là một giải pháp tiềm năng trong lĩnh vực du lịch và dịch vụ, đặc biệt trong bối cảnh nhu cầu số hóa và cá nhân hóa trải nghiệm người dùng ngày càng tăng. Ứng dụng này được phát triển trên nền tảng Flutter, cho phép hỗ trợ đa nền tảng Android với giao diện người dùng mượt mà và hiệu năng cao.



Hệ thống sẽ cung cấp các tính năng chính như:

1. **Tìm kiếm khách sạn**: Cho phép người dùng tìm kiếm khách sạn dựa trên vị trí, giá cả, loại phòng và các tiện nghi khác.
2. **Đặt phòng trực tuyến**: Hỗ trợ người dùng đặt phòng theo thời gian thực, với thông tin chi tiết về phòng và dịch vụ.
3. **Thanh toán**: Tích hợp các phương thức thanh toán an toàn và tiện lợi như thẻ tín dụng, ví điện tử.
4. **Đánh giá và phản hồi**: Người dùng có thể đánh giá khách sạn và để lại nhận xét để cải thiện chất lượng dịch vụ.
5. **Quản lý đặt phòng**: Người dùng có thể xem, chỉnh sửa hoặc hủy đặt phòng một cách dễ dàng.

Ứng dụng không chỉ nâng cao trải nghiệm đặt phòng khách sạn mà còn tối ưu hóa quy trình quản lý dịch vụ cho các nhà cung cấp. Để đảm bảo hiệu quả và an toàn, cần chú trọng vào các yếu tố như:

* Thiết kế giao diện thân thiện, dễ sử dụng.
* Bảo mật dữ liệu khách hàng.
* Đảm bảo tính chính xác trong việc hiển thị thông tin phòng và trạng thái đặt phòng.

# Chương II. CƠ SỞ LÝ THUYẾT:

## Tổng quan công nghệ

### Ngôn ngữ lập trình Dart



Hình 1 Ngôn ngữ lập trình Dart

Dart là một ngôn ngữ lập trình đa mục đích mã nguồn mở, đã được đặt nền móng bởi Google. Đây là một khía cạnh của ngôn ngữ lập trình hướng đối tượng, đặc trưng bởi cú pháp kiểu C. Tinh thần lập trình hướng đối tượng thể hiện qua việc hỗ trợ giao diện và lớp, mở ra khả năng sáng tạo không giới hạn khi đặt lên bàn cân với những ngôn ngữ khác. Sự đa dạng của Dart thể hiện thông qua khả năng phát triển ứng dụng web, di động, máy chủ và máy tính để bàn.

Không chỉ dừng lại ở việc phát triển ứng dụng cho một nền tảng, Dart mở ra cánh cửa cho sự sáng tạo trên cả hai nền tảng quan trọng: Android và iOS. Và trong cuộc hành trình này, Flutter nổi lên như một công cụ vượt trội, được Google chắp cánh. Flutter, một framework ra đời từ Google, trở thành điểm đặc biệt giúp xây dựng những ứng dụng tuyệt đẹp trên cả hai nền tảng bằng một nền tảng mã nguồn duy nhất.

* Cách hoạt động cơ bản
* Kiểu ngôn ngữ:
  + Dart là ngôn ngữ hướng đối tượng (Object-Oriented): Mọi thứ trong Dart đều là đối tượng, kể cả các giá trị nguyên thủy như số và chuỗi.
  + Statically Typed: Dart sử dụng kiểu dữ liệu tĩnh. Các kiểu dữ liệu được kiểm tra tại thời gian biên dịch, nhưng cũng hỗ trợ type inference (suy luận kiểu) để giảm bớt việc phải khai báo kiểu rõ ràng.
* Chạy trên các nền tảng khác nhau:
  + Dart VM: Dart chạy trên Máy ảo Dart (Dart VM) khi thực hiện các chương trình máy tính để bàn hoặc trên server.
  + AOT (Ahead-of-Time) Compilation: Dart được biên dịch trước để tạo mã máy gốc, đảm bảo hiệu suất cao, đặc biệt trên các ứng dụng di động.
  + JIT (Just-in-Time) Compilation: Dart có thể biên dịch trong thời gian chạy, giúp tăng tốc quá trình phát triển bằng cách hot reload ứng dụng.

### 1.2 Ngôn ngữ lập trình Flutter



Hình 2 Ngôn ngữ lập trình Flutter



* Tổng quan về Flutter
* Flutter là một khung nguồn mở do Google phát triển và hỗ trợ. Các nhà phát triển frontend và fullstack sử dụng Flutter để xây dựng giao diện người dùng (UI) của ứng dụng cho nhiều nền tảng chỉ với một nền mã duy nhất.
* Tại thời điểm ra mắt vào năm 2018, Flutter chủ yếu hỗ trợ phát triển ứng dụng di động. Hiện nay, Flutter hỗ trợ phát triển ứng dụng trên sáu nền tảng: iOS, Android, web, Windows, MacOS và Linux.
* Flutter được hỗ trợ như thế nào?
* Flutter được hỗ trợ bởi Google và một cộng đồng nguồn mở hoạt động trên Reddit, Discord, Slack, Stack Overflow và Gitter. Google đã liên tục cập nhật Flutter kể từ khi phát hành vào năm 2018, bao gồm bản cập nhật Flutter 3 vào năm 2022, mở rộng hỗ trợ ổn định cho macOS và Linux.
* Để giúp Flutter dễ học hơn, Google đã viết các tài liệu và hướng dẫn chi tiết trên trang của Flutter. Để tương tác với người dùng Flutter, Google cũng tổ chức các sự kiện toàn cầu, quảng bá các dự án cộng đồng và tài trợ cho các thử thách của nhà phát triển. Các sự kiện sắp tới có thể được tìm thấy trên trang của Flutter.
* Cộng đồng của Flutter đã tạo ra hàng nghìn gói bên thứ ba và các công cụ tuyệt vời giúp hợp lý hóa trải nghiệm của nhà phát triển. Các thư viện này có sẵn tại pub.dev.

## Xác định yêu cầu

### 2.1 Yêu cầu chức năng

 **Tìm kiếm khách sạn:**

* Người dùng có thể tìm kiếm khách sạn dựa trên các tiêu chí:
  + Vị trí (thành phố, khu vực).
  + Khoảng giá.
  + Đánh giá từ người dùng.
  + Tiện ích (wifi, bể bơi, bữa sáng miễn phí, v.v.).

 **Xem thông tin chi tiết:**

* Hiển thị thông tin chi tiết về khách sạn:
  + Tên, địa chỉ, hình ảnh.
  + Loại phòng và giá cả.
  + Tiện ích.
  + Chính sách hủy phòng.

 **Đặt phòng:**

* Người dùng có thể chọn loại phòng và đặt phòng.
* Hệ thống xác nhận đơn đặt phòng sau khi thanh toán thành công.

 **Thanh toán:**

* Tích hợp các phương thức thanh toán:
  + Thẻ tín dụng/thẻ ghi nợ.
  + Ví điện tử (Momo, ZaloPay, v.v.).
  + Thanh toán khi nhận phòng (COD).

 **Quản lý đặt phòng:**

* Người dùng có thể:
  + Xem danh sách các đặt phòng của mình.
  + Hủy đặt phòng trước thời hạn quy định.

 **Đánh giá và nhận xét:**

* Người dùng có thể để lại đánh giá và nhận xét sau khi trải nghiệm.

 **Hỗ trợ người dùng:**

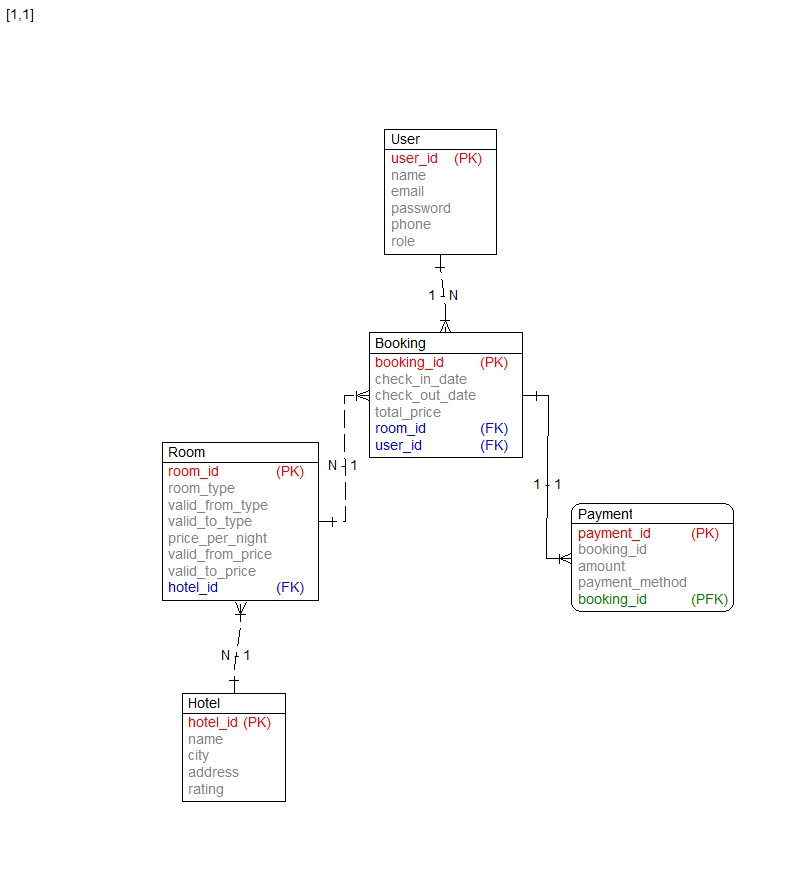
* Tích hợp tính năng trò chuyện (chat) hoặc form liên hệ để hỗ trợ.

### 2.2 Yêu cầu phi chức năng

1. **Hiệu suất:**
   * Ứng dụng phải tải nhanh và phản hồi trong vòng < 2 giây trên các thao tác chính.
2. **Bảo mật:**
   * Bảo mật dữ liệu người dùng và giao dịch thanh toán.
   * Mã hóa dữ liệu nhạy cảm (SSL/TLS).
3. **Khả năng mở rộng:**
   * Hệ thống dễ dàng mở rộng để tích hợp thêm các khách sạn và tính năng mới.
4. **Đa nền tảng:**
   * Ứng dụng phải hoạt động mượt mà trên cả Android và iOS.
5. **Khả dụng (Availability):**
   * Đảm bảo ứng dụng hoạt động 24/7.
6. **Thiết kế giao diện:**
   * Giao diện thân thiện, dễ sử dụng, tối ưu hóa trải nghiệm người dùng (UI/UX).
7. **Hỗ trợ nhiều ngôn ngữ (nếu có):**
   * Tùy chọn ngôn ngữ Anh, Việt, hoặc các ngôn ngữ khác.

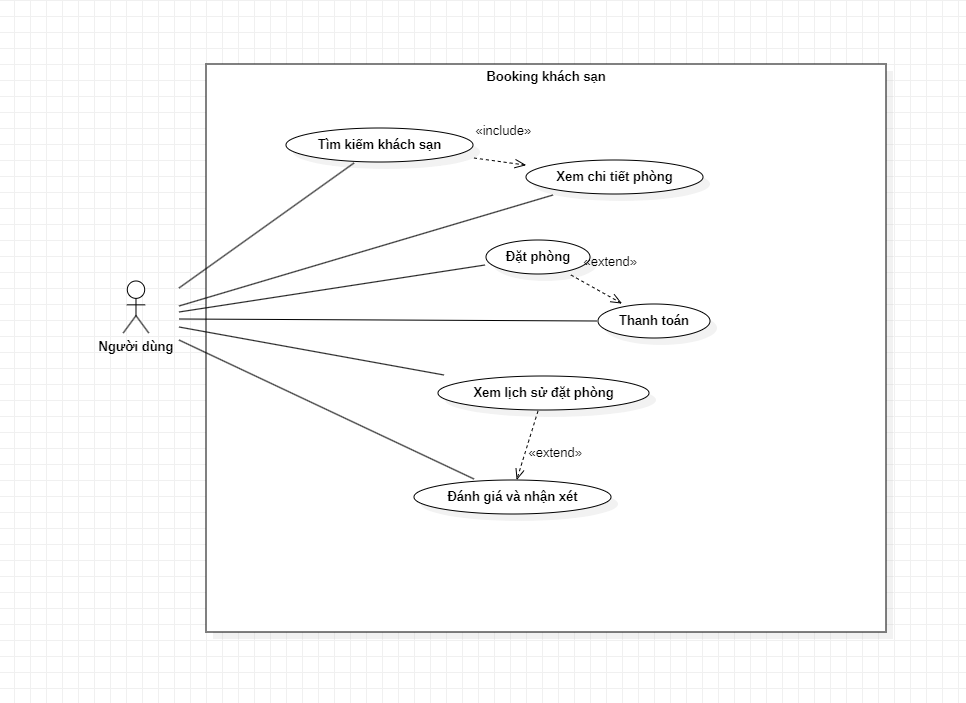
## Thiết kế mô hình hóa

### 3.1 Mô hình erd

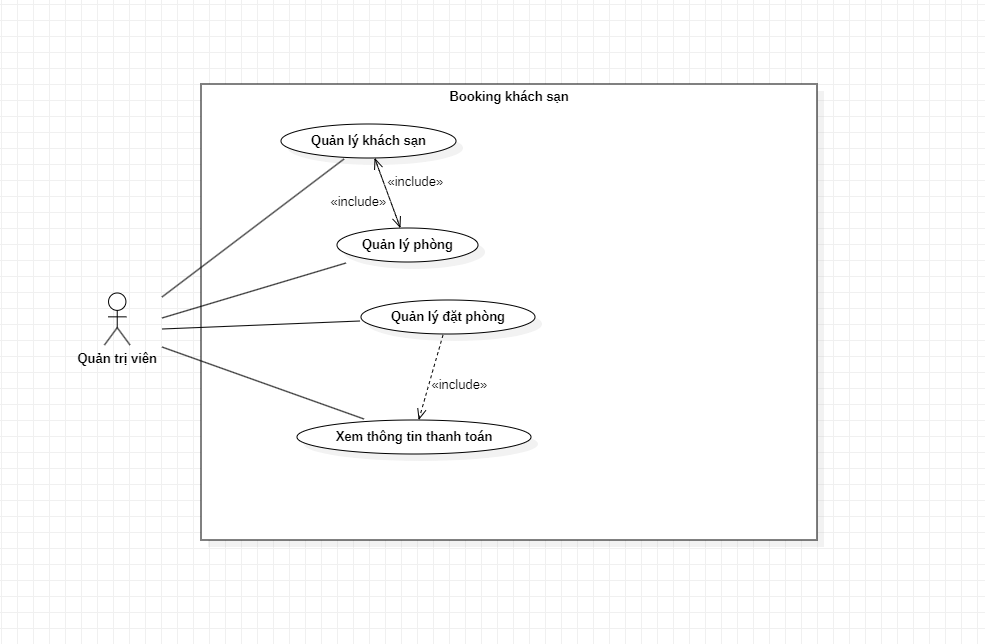


Hình 3: Mô hình erd

### 3.2 Mô hình use case



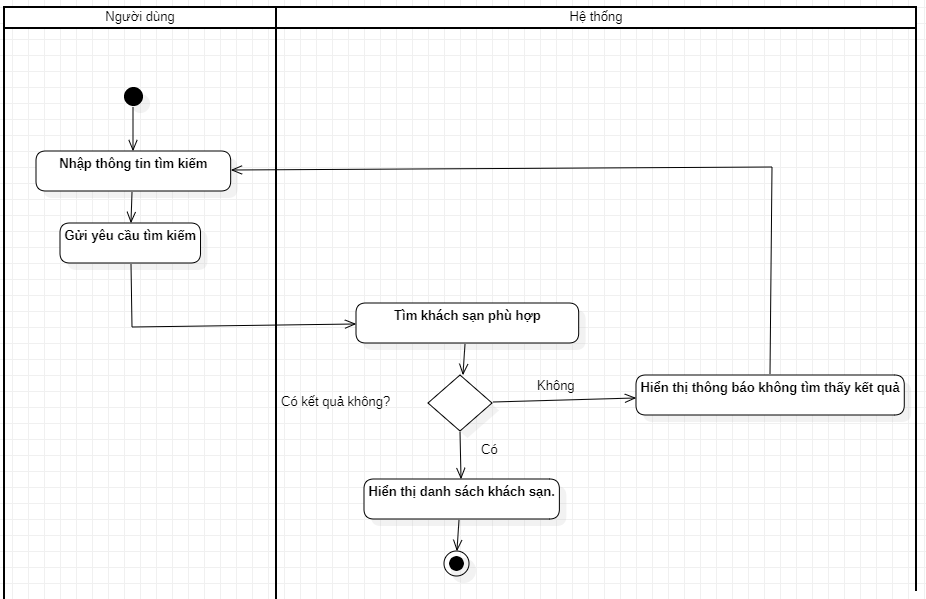
Hình 4: Mô hình của người dùng



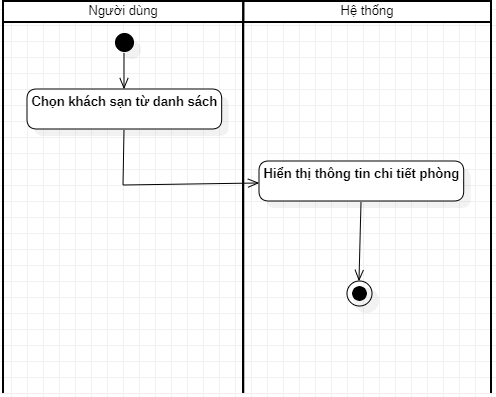
Hình 5: Mô hình của Admin

### 3.3 Mô hình activity diagram

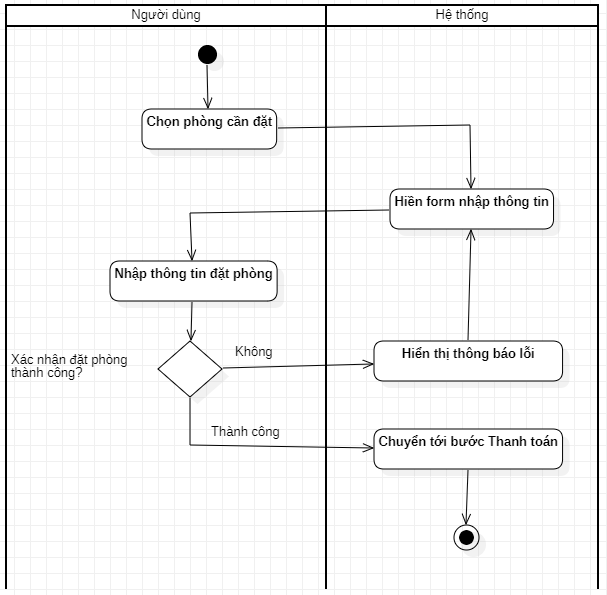
* Đối với người dùng



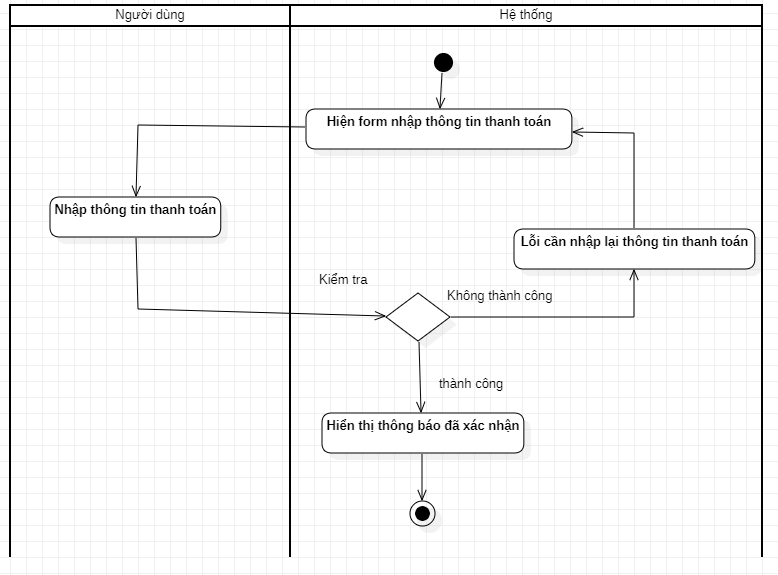
Hình 6: Tìm kiếm khách sạn



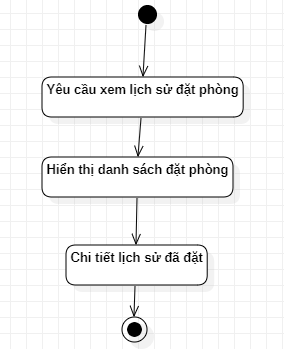
Hình 7: Người dùng chọn một khách sạn và xem thông tin chi tiết về các phòng trong khách sạn



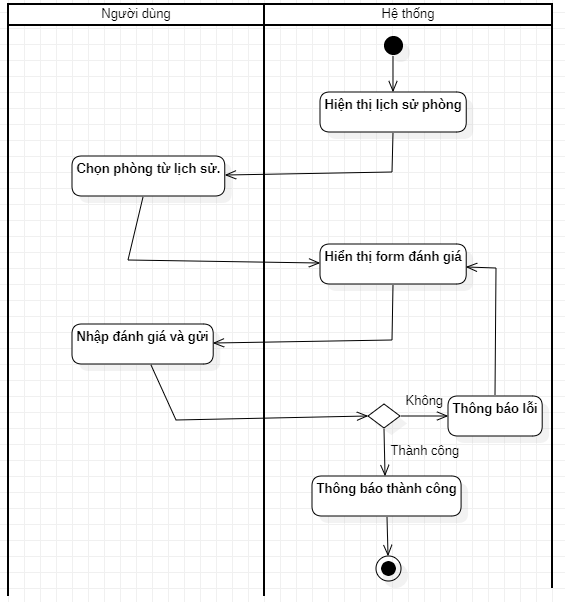
Hình 8: Người dùng đặt phòng



Hình 9: Người dùng thanh toán



Hình 10: Người dùng xem lịch sử đặt phòng



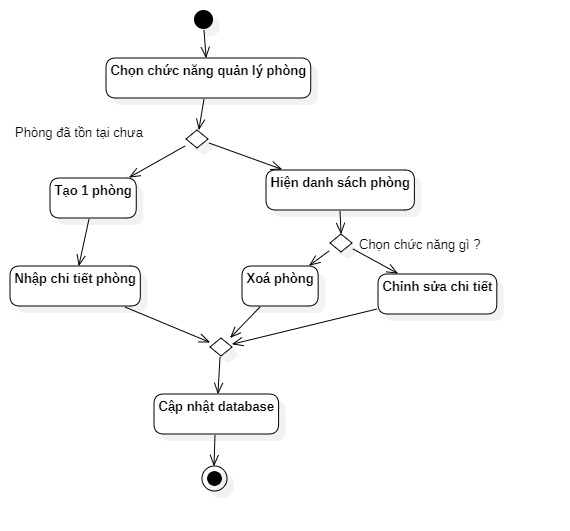
Hình 11: Người dùng đánh giá và nhận xét

* Đối với quản trị viên (Admin)

A diagram of a diagram

Description automatically generated

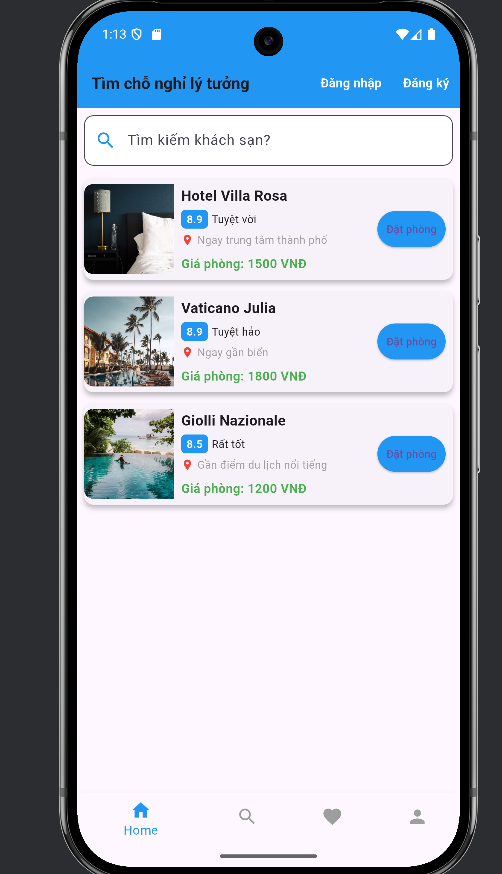
Hình 12: Quản trị viên quản lý khách sạn



Hình 13: Quản trị viên quản lý phòng

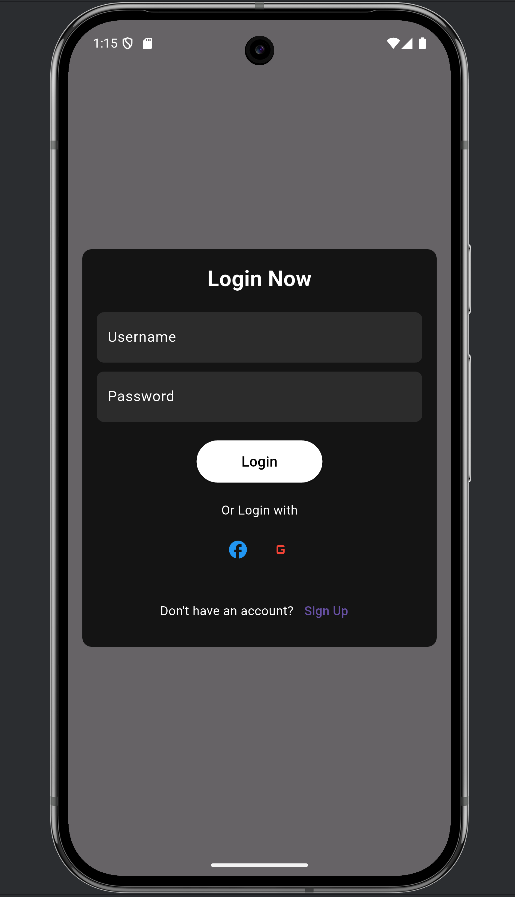
# Chương III. KẾT QUẢ THỰC NGHIỆM

## 1 Giao diện trang chủ của user



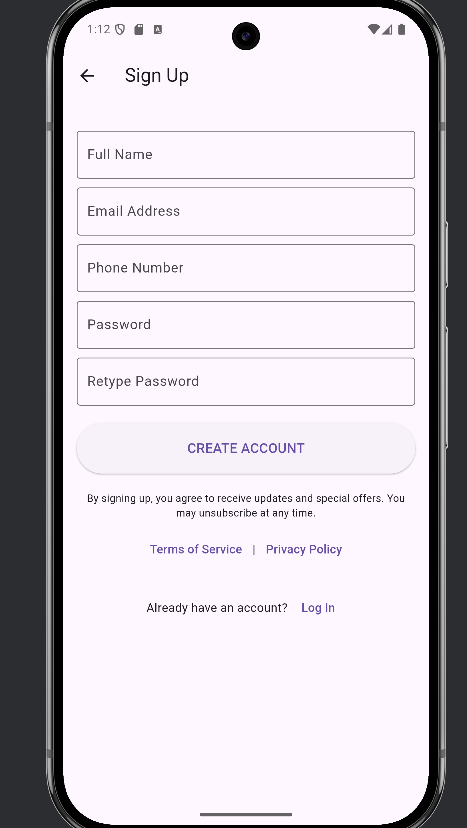
Hình 14: Giao diện trang chủ

## 2 Giao diện đăng nhập



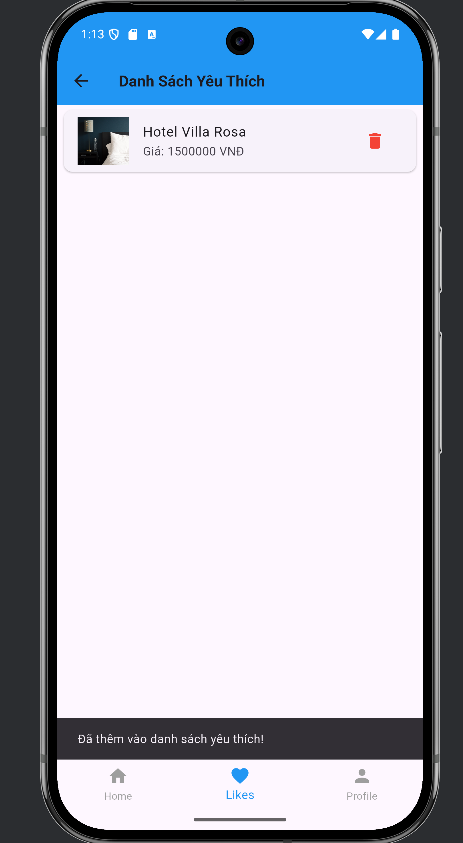
Hình 15: Giao diện đăng nhập

## 3 **Giao diện đăng ký**



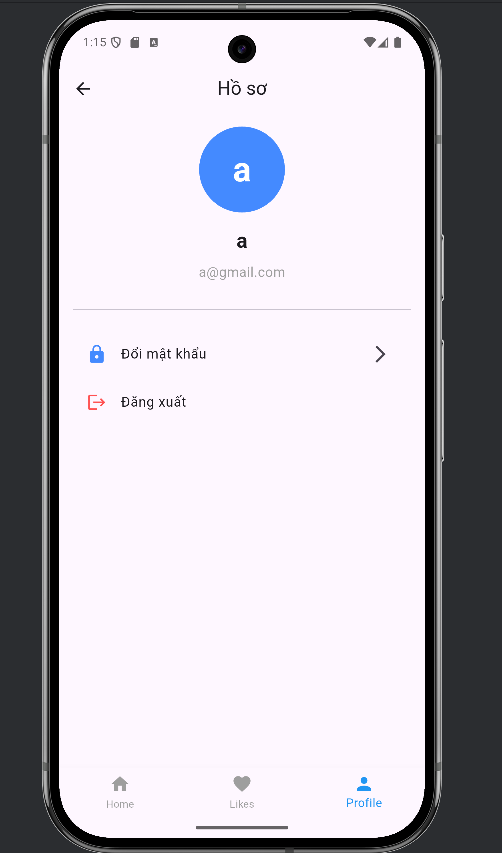
Hình 16: Giao diện đăng ký

## 4 **Giao diện danh sách yêu thích**



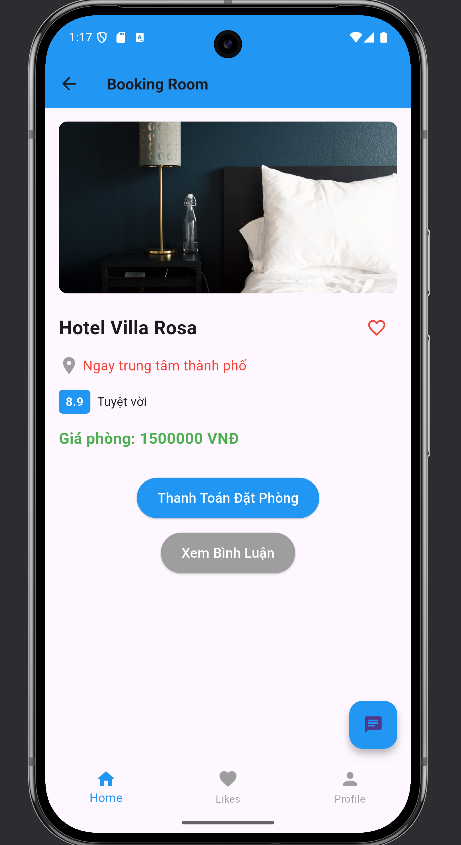
Hình 17: Giao diện danh sách yêu thích

## 5 **Giao diện thông tin**



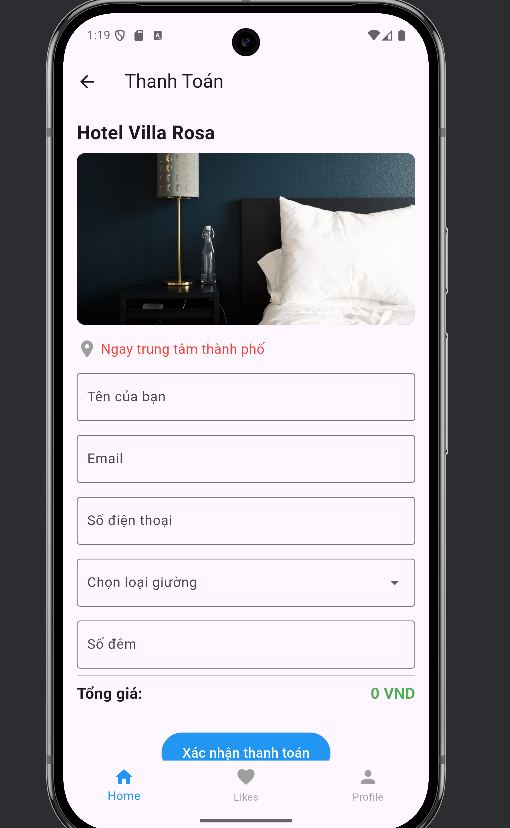
Hình 18: Giao diện thông tin (Profile)

## 6 **Giao diện chi tiết khách sạn**



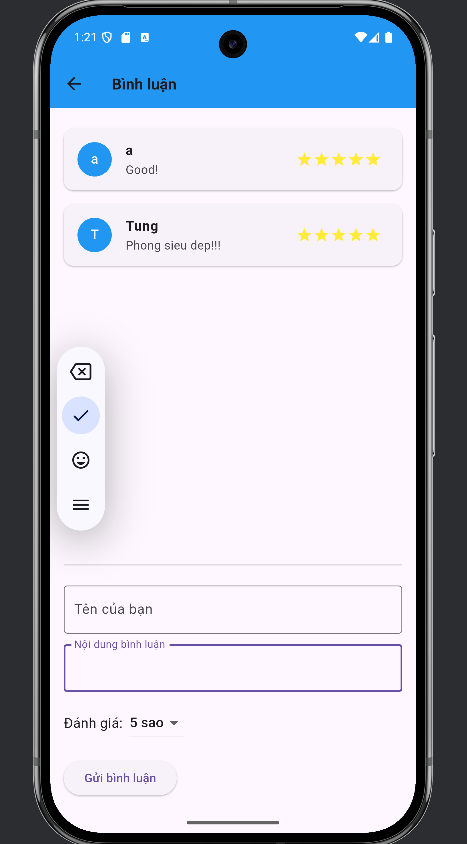
Hình 19: Giao diện chi tiết khách sạn

## 7 **Giao diện thanh toán**



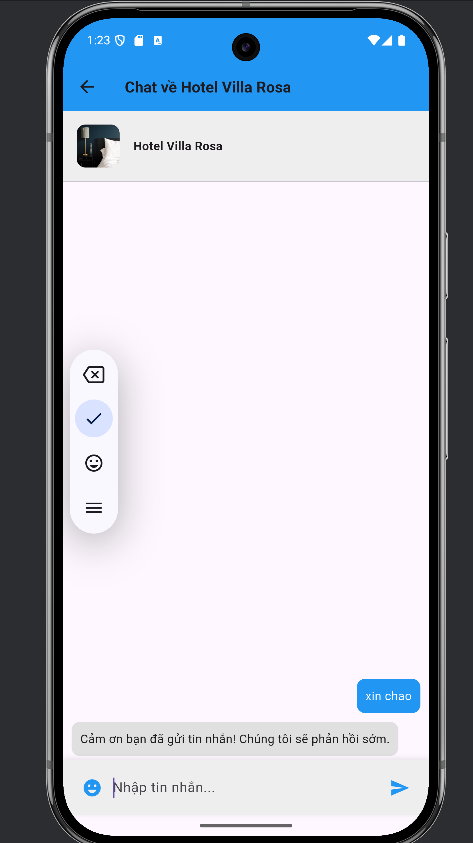
Hình 20: Giao diện thanh toán

## 8 **Giao diện bình luận**



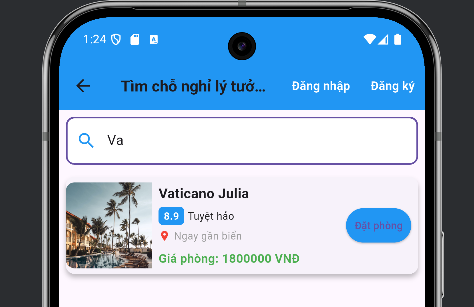
Hình 21:Giao diện bình luận và đánh giá

## 9 **Giao diện nhắn tin hỗ trợ**



Hình 22: Giao diện nhắn tin hỗ trợ

## 10 **Chức năng tìm kiếm khách sạn**



Hình 23: Chức năng tìm kiếm khách sạn

# Chương IV: KẾT LUẬN VÀ KIẾN NGHỊ

## 1 Kết Luận Chung

Dự án xây dựng ứng dụng Booking Khách Sạn sử dụng Flutter là một ví dụ điển hình về cách áp dụng công nghệ di động hiện đại vào lĩnh vực du lịch và khách sạn. Qua việc phát triển ứng dụng này, chúng ta có thể rút ra một số kết luận quan trọng:

**1. Tính khả thi và hiệu quả của ứng dụng:**

* **Flutter** là một framework mạnh mẽ và linh hoạt, giúp phát triển ứng dụng di động đa nền tảng với hiệu suất cao và chi phí thấp. Việc sử dụng Flutter cho ứng dụng này giúp tiết kiệm thời gian và nguồn lực khi phát triển cho cả hệ điều hành **iOS** và **Android** mà không cần phải viết mã riêng biệt cho mỗi nền tảng.

**2. Quản lý và trải nghiệm người dùng:**

* Ứng dụng cung cấp một quy trình đặt phòng khách sạn đơn giản và mượt mà, từ việc tìm kiếm khách sạn, xem thông tin chi tiết về phòng, đặt phòng cho đến thanh toán.
* Việc tích hợp các tính năng như **đặt phòng trực tuyến**, **xem lịch sử đặt phòng**, và **để lại đánh giá** mang lại sự tiện lợi và cải thiện trải nghiệm người dùng.

**3. Các mô hình và mối quan hệ dữ liệu rõ ràng:**

* Các mô hình như **User**, **Hotel**, **Room**, **Booking**, và **Payment** được xác định rõ ràng với các mối quan hệ **1:N** và **N:1** giúp việc quản lý dữ liệu trở nên dễ dàng và hiệu quả.
* Mối quan hệ giữa các thực thể này đảm bảo rằng hệ thống có thể theo dõi mọi giao dịch, từ việc người dùng đặt phòng đến thanh toán và quản lý phòng.

**4. Khả năng mở rộng và bảo trì hệ thống:**

* Thiết kế hệ thống theo hướng **modular** và dễ dàng mở rộng, cho phép thêm các tính năng mới như quản lý khuyến mãi, đánh giá và phản hồi của khách hàng, hay các báo cáo thống kê mà không làm ảnh hưởng đến các chức năng hiện tại.

**5. Thách thức và cải tiến trong tương lai:**

* Mặc dù ứng dụng đã đáp ứng được các yêu cầu cơ bản, nhưng vẫn còn một số thách thức cần giải quyết trong tương lai như:
  + **Bảo mật dữ liệu**: Cần đảm bảo rằng thông tin người dùng, bao gồm dữ liệu thanh toán, được bảo vệ an toàn.
  + **Khả năng hoạt động trong điều kiện thực tế**: Cần kiểm tra và tối ưu hóa ứng dụng để có thể hoạt động ổn định trên nhiều loại thiết bị và mạng internet có tốc độ thấp.
  + **Tối ưu hóa giao diện**: Mặc dù ứng dụng có thể chạy trên cả hai nền tảng iOS và Android, việc tối ưu hóa giao diện người dùng cho từng nền tảng riêng biệt sẽ giúp nâng cao trải nghiệm.

**6. Tầm quan trọng của công nghệ trong ngành du lịch:**

* Việc áp dụng công nghệ, đặc biệt là các ứng dụng di động trong ngành du lịch, đã mở ra cơ hội lớn cho các doanh nghiệp khách sạn trong việc nâng cao chất lượng dịch vụ, tăng cường tương tác với khách hàng, và gia tăng doanh thu.

## 2 Phương hướng phát triển

* Tiếp tục hoàn thiện sản phẩm và sửa các lỗi còn tồn đọng
* Tích hợp thêm nhiều công nghệ mới
* Đưa sản phẩm lên hoạt động

# Tài liệu tham khảo

[1] [Use case là gì ?](https://itnavi.com.vn/blog/use-case-la-gi)

[2] [Giới thiệu về flutter](https://viblo.asia/p/gioi-thieu-ve-flutter-bWrZnNxrZxw)

[3] [Ưu điểm vượt trội và cơ hội việc làm hấp dẫn](https://topdev.vn/blog/flutter-la-gi/)

[4] [Giới thiệu cơ bản về ngôn ngữ lập trình Dart](https://200lab.io/blog/tu-hoc-ngon-dart-nhung-dieu-can-biet-truoc-khi-bat-dau/)

[5] [Giới thiệu ngôn ngữ Dart](https://viblo.asia/p/gioi-thieu-ngon-ngu-dart-ORNZqdv3K0n)

[6] [Ứng dụng của ngôn ngữ lập trình Dart](https://topdev.vn/blog/dart-la-gi/)

[7] [Dart API reference documentation](https://api.dart.dev/)

[8] [Flutter Widgets Catalog](https://docs.flutter.dev/ui/widgets)

[9] [Chat GPT](https://chatgpt.com)