TRƯỜNG ĐẠI HỌC CẦN THƠ

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH KỸ THUẬT PHẦN MỀM**

**Đề tài**

**PHÁT TRIỂN HỆ THỐNG ĐẶT PHÒNG HOMESTAY SỬ DỤNG THIẾT KẾ DOMAIN DRIVEN**

**Sinh viên:Phạm Huỳnh Thống**

**Mã số: DC2096N525**

**Khóa: 46**

**Cần Thơ, 04/2025**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH KỸ THUẬT PHẦN MỀM**

**Đề tài**

**PHÁT TRIỂN HỆ THỐNG ĐẶT PHÒNG HOMESTAY SỬ DỤNG THIẾT KẾ DOMAIN DRIVEN**

**(DEVELOPING A HOMESTAY BOOKING SYSTEM USING DOMAIN DRIVEN DESIGN)**

**Người hướng dẫn Sinh viên thực hiện**

**TS. Phan Phương Lan Phạm Huỳnh Thống**

**Mã số: DC2096N525**

**Khóa: 47**

**Cần Thơ, 04/2025**

# LỜI CẢM ƠN

Cần Thơ, ngày … … tháng … … năm 2025

**Sinh viên thực hiện**

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.………...………...………...………...………...………...………...…………………

.………...………...………...………...………...………...………...…………………

.………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...………………………….………...………...………...………...………...………...………...……………………………………………

Cần Thơ, ngày … … tháng … … năm 2025

**Giáo viên hướng dẫn**

# CAM KẾT KẾT QUẢ

Tôi xin cam kết luận văn tốt nghiệp với tên đề tài “Phát triển hệ thống đặt phòng HomeStay sử dụng thiết kế Domain Driven” được hoàn thành dựa trên kết quả nghiên cứu của tôi dưới sự hướng dẫn của TS. Phan Phương Lan. Các nguồn tài liệu tham khảo đã được ghi rõ trong phần tài liệu tham khảo.

Cần Thơ, ngày … … tháng … … năm 2025

**Sinh viên thực hiện**

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc198325027)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN ii](#_Toc198325028)

[CAM KẾT KẾT QUẢ iii](#_Toc198325029)

[MỤC LỤC iv](#_Toc198325030)

[DANH MỤC BẢNG vii](#_Toc198325031)

[DANH MỤC HÌNH ẢNH viii](#_Toc198325032)

[DANH MỤC TỪ VIẾT TẮT ix](#_Toc198325033)

[TÓM TẮT x](#_Toc198325034)

[ABSTRACT xi](#_Toc198325035)

[PHẦN GIỚI THIỆU 1](#_Toc198325036)

[1. Đặt vấn đề 1](#_Toc198325037)

[2. Lịch sử giải quyết vấn đề 1](#_Toc198325038)

[3. Mục tiêu đề tài 1](#_Toc198325039)

[4. Đối tượng và phạm vi nghiên cứu 1](#_Toc198325040)

[4.1. Đối tượng nghiên cứu 1](#_Toc198325041)

[4.2. Phạm vi nghiên cứu 1](#_Toc198325042)

[5. Phương pháp và nội dung nghiên cứu 2](#_Toc198325043)

[5.1. Phương pháp nghiên cứu 2](#_Toc198325044)

[5.2. Nội dung nghiên cứu 2](#_Toc198325045)

[5.2.1.Quy trình thực hiện 2](#_Toc198325046)

[5.2.2.Công nghệ cho phát triển ứng dụng 2](#_Toc198325047)

[6. Những đóng góp chính của đề tài 2](#_Toc198325048)

[7. Bố cục luận văn 2](#_Toc198325049)

[PHẦN NỘI DUNG 3](#_Toc198325050)

[CHƯƠNG 1. MÔ TẢ BÀI TOÁN PHÁT TRIỂN HỆ THỐNG ĐẶT PHÒNG HOMESTAY 3](#_Toc198325051)

[1. Giới thiệu 3](#_Toc198325052)

[2. Các yêu cầu chức năng 3](#_Toc198325053)

[2.1. Các trường hợp sử dụng của khách hàng 3](#_Toc198325054)

[2.1.1.Sơ đồ trường hợp sử dụng 3](#_Toc198325055)

[2.1.2.Lên kế hoạch du lịch bằng AI 4](#_Toc198325056)

[2.1.3.Đặt phòng 5](#_Toc198325057)

[2.2. Các trường hợp sử dụng của chủ homestay 6](#_Toc198325058)

[2.2.1.Sơ đồ trường hợp sử dụng 6](#_Toc198325059)

[2.2.2.Thêm phòng 7](#_Toc198325060)

[3. Các yêu cầu phi chức năng 8](#_Toc198325061)

[3.1. Yêu cầu thực thi 8](#_Toc198325062)

[3.2. Yêu cầu an toàn 8](#_Toc198325063)

[3.3. Yêu cầu bảo mật 8](#_Toc198325064)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 9](#_Toc198325065)

[1. Giới thiệu nguyên tắc thiết kế Domain-Driven Design (DDD) 9](#_Toc198325066)

[2. So sánh giữa Domain-Driven Design và Clean Architecture 9](#_Toc198325067)

[3. Kiến trúc ứng dụng theo Domain-Driven Design 10](#_Toc198325068)

[CHƯƠNG 3. THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP ĐỂ PHÁT TRIỂN HỆ THỐNG ĐẶT PHÒNG HOMESTAY 11](#_Toc198325069)

[1. Kiến trúc hệ thống 11](#_Toc198325070)

[1.1. Web Client 11](#_Toc198325071)

[1.2. Web Server 11](#_Toc198325072)

[1.2.1.Frontend 11](#_Toc198325073)

[1.2.2.Backend 11](#_Toc198325074)

[1.3. Database Server 11](#_Toc198325075)

[1.4. Dịch vụ bên thứ ba 12](#_Toc198325076)

[2. Thiết kế dữ liệu 12](#_Toc198325077)

[2.1. Mô hình dữ liệu 12](#_Toc198325078)

[2.2. Từ điển dữ liệu 15](#_Toc198325079)

[2.2.1.Bảng permission 15](#_Toc198325080)

[2.2.2.Bảng users 15](#_Toc198325081)

[3. Thiết kế theo chức năng 16](#_Toc198325082)

[3.1. Chức năng lên kế hoạch du lịch với AI 16](#_Toc198325083)

[3.2. Chức năng đặt phòng homestay 18](#_Toc198325084)

[CHƯƠNG 4. KIỂM THỬ VÀ ĐÁNH GIÁ CHO HỆ THỐNG ĐẶT PHÒNG HOMESTAY 19](#_Toc198325085)

[1. Giới thiệu 19](#_Toc198325086)

[1.1. Mục tiêu kiểm thử 19](#_Toc198325087)

[1.2. Phạm vi kiểm thử 19](#_Toc198325088)

[2. Chi tiết kế hoạch kiểm thử 19](#_Toc198325089)

[2.1. Các yêu cầu được kiểm thử 19](#_Toc198325090)

[2.2. Cách tiếp cận 19](#_Toc198325091)

[2.3. Tiêu chí kiểm thử thành công / thất bại 19](#_Toc198325092)

[2.4. Tiêu chí đình chỉ và yêu cầu bắt đầu lại 19](#_Toc198325093)

[2.5. Sản phẩm bàn giao của kiểm thử 19](#_Toc198325094)

[2.6. Môi trường kiểm thử 19](#_Toc198325095)

[2.7. Tài nguyên 19](#_Toc198325096)

[2.8. Các rủi ro 19](#_Toc198325097)

[3. Các trường hợp kiểm thử 20](#_Toc198325098)

[3.1. Kiểm thử chức năng chủ homestay thêm phòng mới 20](#_Toc198325099)

[3.2. Kiểm thử chức năng khách hàng đặt phòng 21](#_Toc198325100)

[4. Đánh giá kết quả kiểm thử 23](#_Toc198325101)

[PHẦN KẾT LUẬN 24](#_Toc198325102)

[1. Kết quả đạt được 24](#_Toc198325103)

[1.1. Về kiến thức và kỹ năng 24](#_Toc198325104)

[1.2. Về chương trình và khả năng ứng dụng trong thực tiễn 24](#_Toc198325105)

[2. Hạn chế và khó khăn 24](#_Toc198325106)

[3. Hướng phát triển 24](#_Toc198325107)

[TÀI LIỆU THAM KHẢO 25](#_Toc198325108)

# DANH MỤC BẢNG

[Bảng 1. Yêu cầu chức năng lên kế hoạch du lịch bằng AI 4](#_Toc198325330)

[Bảng 2. Yêu cầu chức năng đặt phòng 5](#_Toc198325331)

[Bảng 3. Yêu cầu chức năng thêm phòng 7](#_Toc198325332)

[Bảng 4. So sánh DDD và Clean Architecture 9](#_Toc198325333)

[Bảng 5 Bảng permission 15](#_Toc198325334)

[Bảng 6 Bảng users 15](#_Toc198325335)

[Bảng 7 Các thành phần trong giao diện Lên kế hoạch du lịch với AI 17](#_Toc198325336)

[Bảng 8 Dữ liệu sử dụng trong Lên kế hoạch du lịch với AI 17](#_Toc198325337)

[Bảng 9 Bảng kiểm tra chức năng chủ homestay thêm phòng mới 20](#_Toc198325338)

[Bảng 10 Bảng kiểm tra chức năng khách hàng đặt phòng 21](#_Toc198325339)

[Bảng 11 Bảng tổng hợp kết quả kiểm thử 23](#_Toc198325340)

# DANH MỤC HÌNH ẢNH

[Hình 2. Mô hình tổng quan kiến trúc hệ thống đặt phòng homestay 11](#_Toc198325560)

[Hình 3. Mô hình dữ liệu mức quan niệm (CDM) 13](#_Toc198325561)

[Hình 4: Mô hình dữ liệu mức vật lý (PDM) 14](#_Toc198325562)

[Hình 5 Giao diện lên kế hoạch du lịch với AI 16](#_Toc198325563)

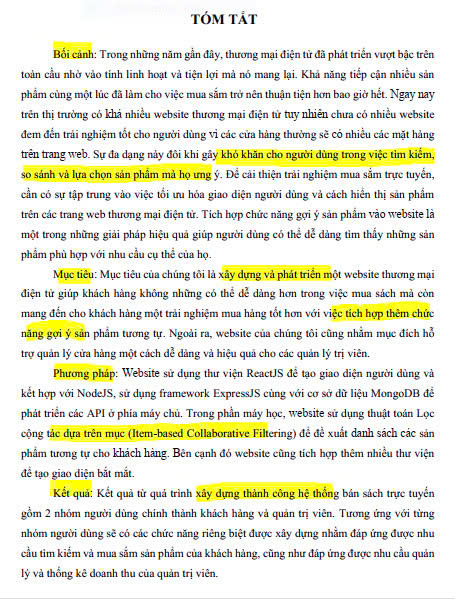
[Hình 6. Sơ đồ cách xử lý Lên kế hoạch du lịch với AI 18](#_Toc198325564)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuật ngữ/ Viết tắt** | **Định nghĩa/ Giải thích** |
| 1 | DDD | Domain-Driven Design – Phương pháp tiếp cận thiết kế phần mềm tập trung vào mô hình nghiệp vụ và domain của hệ thống. |
| 2 | Homestay | Chỗ ở dạng homestay – Loại hình lưu trú ngắn hạn do chủ nhà cung cấp, cho phép khách thuê trải nghiệm như người bản địa. |
| 3 | Framework | Bộ khung phát triển phần mềm – Tập hợp các đoạn mã, thư viện và công cụ hỗ trợ lập trình, giúp xây dựng ứng dụng nhanh chóng và hiệu quả. |
| 4 | API | Application Programming Interface – Giao diện lập trình ứng dụng, giúp các hệ thống phần mềm giao tiếp và trao đổi dữ liệu với nhau. |
| 5 | Admin | Quản trị viên – Người có quyền quản lý hệ thống, kiểm soát người dùng, quản lý đặt phòng, nội dung và cấu hình hệ thống. |
| 6 | Landlord | Chủ homestay – Người sở hữu và quản lý các homestay trên nền tảng, có quyền tạo, chỉnh sửa thông tin chỗ ở, đặt giá và quản lý đặt phòng. |
| 7 | Domain | Miền nghiệp vụ – Phạm vi nghiệp vụ mà hệ thống đặt phòng homestay cần giải quyết, bao gồm các quy tắc và logic nghiệp vụ. |
| 8 | Bounded Context | Ngữ cảnh giới hạn – Một phần riêng biệt của domain trong DDD, xác định ranh giới cho một tập hợp logic nghiệp vụ liên quan. |
| 9 | Testcase | Trường hợp kiểm thử – Một tập hợp điều kiện và dữ liệu đầu vào được thiết kế để kiểm tra tính đúng đắn của hệ thống. |

# TÓM TẮT

Viết tóm tắt bằng tiếng Việt theo gợi ý bên dưới.



# ABSTRACT

Viết tóm tắt bằng Tiếng Anh

# PHẦN GIỚI THIỆU

## Đặt vấn đề

Phần này cần trình bày rõ bối cảnh thực tiễn hoặc xu hướng công nghệ đang phát triển có liên quan đến đề tài nghiên cứu. Trên cơ sở đó, cần xác định và làm rõ vấn đề cụ thể, thách thức hoặc khoảng trống chưa được giải quyết trong lĩnh vực kỹ thuật phần mềm. Nội dung phải thể hiện được tính cấp thiết và ý nghĩa thực tiễn hoặc khoa học của vấn đề, qua đó dẫn dắt hợp lý đến lý do lựa chọn đề tài.

Lưu ý: đây là phần mở đầu định hướng cho toàn bộ tiểu luận, vì vậy cần được trình bày ngắn gọn, chặt chẽ và tập trung làm nổi bật động cơ nghiên cứu..

## Lịch sử giải quyết vấn đề

Phần này trình bày tổng quan các hệ thống đã từng được xây dựng để giải quyết vấn đề tương tự trước đây. Người viết cần tóm lược các nghiên cứu liên quan, chỉ ra ưu điểm và hạn chế của chúng, từ đó xác định được sự khác biệt hoặc đóng góp mới (nếu có) của đề tài hiện tại.

## Mục tiêu đề tài

Phần này cần trình bày cụ thể các mục tiêu mà đề tài hướng đến trong suốt quá trình nghiên cứu và phát triển. Nội dung mục tiêu nên được phân chia thành hai cấp độ: mục tiêu tổng quát và các mục tiêu cụ thể.

***Mục tiêu tổng quát*** thể hiện định hướng chung và kết quả cuối cùng mà đề tài mong muốn đạt được.

***Mục tiêu cụ thể*** mô tả từng mục tiêu nhỏ, đo lường được.

## Đối tượng và phạm vi nghiên cứu

### Đối tượng nghiên cứu

Phần này cần xác định và mô tả rõ đối tượng chính mà đề tài tập trung nghiên cứu, khảo sát, phân tích hoặc phát triển. Đối tượng có thể là một hệ thống phần mềm cụ thể, một công nghệ, một quy trình nghiệp vụ, một nhóm người dùng, hoặc một vấn đề kỹ thuật đặc thù cần giải quyết. Việc làm rõ đối tượng nghiên cứu sẽ giúp định hướng chính xác cho các nội dung và phương pháp nghiên cứu được trình bày trong các phần tiếp theo.

### Phạm vi nghiên cứu

Phạm này xác định rõ giới hạn của đề tài nhằm đảm bảo nội dung nghiên cứu tập trung, phù hợp với nguồn lực và thời gian thực hiện. Trong lĩnh vực kỹ thuật phần mềm, phạm vi có thể được giới hạn theo nhiều khía cạnh như không gian địa lý, thời gian thực hiện, chức năng hệ thống, đối tượng người dùng, loại dữ liệu xử lý, hoặc công nghệ áp dụng. Việc xác định phạm vi một cách cụ thể sẽ giúp tránh lan man, đồng thời đảm bảo tính khả thi và hiệu quả trong quá trình triển khai đề tài.

## Phương pháp và nội dung nghiên cứu

### Phương pháp nghiên cứu

Phần này trình bày các phương pháp sẽ được sử dụng trong quá trình thực hiện đề tài. Có thể bao gồm phương pháp thu thập yêu cầu, phân tích hệ thống, v.v.

***Ví dụ:***

*Trong quá trình thực hiện tiểu luận, đề tài đã được triển khai dựa trên ba phương pháp nghiên cứu chính: nghiên cứu tài liệu, quan sát và thực nghiệm.*

*Việc nghiên cứu tài liệu tập trung vào các công nghệ ….*

*Việc quan sát và phân tích các ….*

*Phương pháp thực nghiệm được áp dụng xuyên suốt trong quá trình thiết kế, lập trình ….*

### Nội dung nghiên cứu

* + 1. **Quy trình thực hiện**

Phần này trình bày các bước cụ thể trong quá trình thực hiện đề tài, từ giai đoạn khảo sát, phân tích, thiết kế, lập trình cho đến kiểm thử và đánh giá kết quả.

Lưu ý:

* Nên mô tả rõ ràng trình tự thực hiện, có thể chia thành các giai đoạn với mục tiêu và kết quả mong đợi ở từng giai đoạn.
* Nếu sử dụng mô hình phát triển phần mềm cụ thể (ví dụ: Waterfall, Agile, Scrum...), nên nêu rõ tại đây.
  + 1. **Công nghệ cho phát triển ứng dụng**

Phần này này liệt kê các công nghệ, ngôn ngữ lập trình, nền tảng và công cụ phần mềm sẽ được sử dụng trong đề tài. Bao gồm cả frontend, backend, cơ sở dữ liệu, framework, công cụ hỗ trợ kiểm thử hoặc triển khai.

## Những đóng góp chính của đề tài

Phần này chỉ rõ những giá trị mà đề tài mang lại về mặt lý thuyết hoặc thực tiễn. Các đóng góp có thể là một giải pháp kỹ thuật mới, một hệ thống phần mềm có tính ứng dụng cao, một quy trình được cải tiến, hay một cách tiếp cận mới cho vấn đề cũ. Cần nhấn mạnh đâu là điểm mới và giá trị nổi bật của đề tài.

## Bố cục luận văn

Phần này cung cấp cho người đọc cái nhìn tổng quan về cấu trúc của toàn bộ luận văn. Người viết nên trình bày ngắn gọn nội dung chính của từng chương và mối liên hệ giữa các chương. Cách viết nên logic, rõ ràng để người đọc dễ hình dung tiến trình thực hiện đề tài.

# PHẦN NỘI DUNG

# MÔ TẢ BÀI TOÁN PHÁT TRIỂN WEBSTIE QUẢN LÝ VÀ BÁN BÁNH TRỰC TIẾP “LUNA BAKERY”

## Giới thiệu

Phần này nhằm cung cấp cái nhìn toàn diện và trực quan về bài toán mà phần mềm cần giải quyết. Người viết cần trình bày một cách rõ ràng bối cảnh, mục tiêu, người dùng và môi trường vận hành của hệ thống. Cấu trúc trình bày nên gồm những nội dung chính như sau:

(1) Xác định sản phẩm (sẽ phát triển) là: một phần mở rộng của một hệ thống/họ sản phẩm đã có; hay một bản thay thế cho hệ thống cũ; hay một sản phẩm mới được phát triển độc lập. Nếu phần mềm là một module trong hệ thống lớn hơn, cần mô tả mối liên hệ giữa hệ thống chính và hệ thống con, cũng như cách các thành phần giao tiếp với nhau. Có thể sử dụng sơ đồ khối để minh họa các thành phần chính, các luồng tương tác và kết nối với hệ thống ngoài.

(2) Nhận diện các nhóm người dùng dự kiến sẽ tương tác với hệ thống. Mỗi nhóm có thể khác nhau về: Tần suất sử dụng, Vai trò, quyền truy cập và bảo mật, Mức độ hiểu biết về công nghệ.

(3) Mô tả môi trường kỹ thuật mà phần mềm sẽ hoạt động. Bao gồm cả phần cứng, hệ điều hành, phần mềm liên quan và các giao tiếp cần thiết.

## Các yêu cầu chức năng

Trong phần này, các trường hợp sử dụng tiêu biểu của ứng dụng sẽ được trình bày một cách chi tiết, tập trung vào những chức năng cốt lõi và đặc trưng thể hiện rõ mục tiêu và giá trị của hệ thống.

### Các trường hợp sử dụng của khách hàng

* + 1. **Sơ đồ trường hợp sử dụng**
* Sơ đồ trường hợp sử dụng của khách hàng …

UseCaseKhachHang.diagram



*Hình 1. Sơ đồ trường hợp sử dụng của khách hàng*

#### Thanh toán ZaloPay

Bảng 1. Thanh toán zalo pay

|  |  |
| --- | --- |
| **Tên trường hợp sử dụng**: Thanh toán ZaloPay | ID: UC\_05 |
| Tác nhân chính: Khách hàng | Mức độ cần thiết: Bắt buộc |
| Phân loại: Phức tạp |
| * Các thành phần tham gia và mối quan tâm: thanh toán đơn hàng thông qua ví điện tử ZaloPay với các phương thức thanh toán đa dạng (ví ZaloPay, thẻ ATM, thẻ tín dụng). | |
| Mô tả tóm tắt:Chức năng "Thanh toán ZaloPay" cho phép khách hàng thanh toán đơn hàng thông qua cổng thanh toán ZaloPay, bao gồm việc tạo đơn hàng thanh toán, chuyển hướng đến ZaloPay, xử lý callback và hoàn tất đơn hàng sau khi thanh toán thành công. | |
| Các mối quan hệ:  +Association (kết hợp): Khách hàng  +Include(bao gồm): Tạo đơn hàng thanh toán, Xử lý callback ZaloPay,giúp chức năng hoàn tất đơn hàng  +Extend (mở rộng): Kiểm tra trạng thái thanh toán, Xử lý lỗi thanh toán  +Generalization (tổng quát hóa):Không có | |
| Luồng xử lý bình thường của sự kiện:   * Khách hàng đã tạo checkout và chọn phương thức thanh toán "ZaloPay" * Hệ thống hiển thị nút "Thanh toán với ZaloPay" với thông tin đơn hàng | |
| Các luồng sự kiện con (Subflows):   1. **Khách hàng chọn thanh toán ZaloPay**  * Khách hàng đã tạo checkout và chọn phương thức thanh toán "ZaloPay" * Hệ thống hiển thị nút "Thanh toán với ZaloPay" với thông tin đơn hàng  1. **Khách hàng nhấn nút thanh toán ZaloPay**  * Khách hàng click nút "Thanh toán với ZaloPay" * Hệ thống lưu thông tin thanh toán vào localStorage (amount, payment\_time)  1. **Hệ thống tạo đơn hàng ZaloPay**  * Hệ thống gọi API /api/payment/zalopay/create * Tạo app\_trans\_id với format: YYMMDD\_XXXXXX * Tạo order object với thông tin: app\_id, app\_trans\_id, app\_user, app\_time, amount, description * Tạo MAC signature theo chuẩn ZaloPay * Gửi request đến ZaloPay API  1. **ZaloPay trả về payment URL**  * Nếu thành công, ZaloPay trả về order\_url * Hệ thống lưu app\_trans\_id vào localStorage * Chuyển hướng khách hàng đến trang thanh toán ZaloPay  1. **Khách hàng hoàn tất thanh toán trên ZaloPay**  * Khách hàng thực hiện thanh toán trên ZaloPay (ví điện tử, thẻ ATM, thẻ tín dụng) * ZaloPay xử lý thanh toán và gửi callback về hệ thống  1. **ZaloPay gửi callback về hệ thống**  * ZaloPay gọi callback URL: /api/payment/zalopay/callback * Hệ thống xác thực MAC signature * Parse dữ liệu callback và cập nhật trạng thái thanh toán  1. **Hệ thống tự động hoàn tất đơn hàng**  * Tìm checkout chưa finalize với amount matching * Cập nhật checkout: isPaid = true, paymentStatus = "paid" * Tạo order từ checkout * Xóa giỏ hàng của khách hàng  1. **Chuyển hướng về trang thành công**  * ZaloPay redirect về /zalopay-return với thông tin thanh toán * Hệ thống hiển thị trang xác nhận thanh toán thành công * Khách hàng có thể xem chi tiết đơn hàng | |
| Luồng luân phiên/đặc biệt (Alternate/Exceptional flows): 1. Tính toán giá thanh toán:  * Giá được tính dựa trên tổng giá trị đơn hàng (cart.totalPrice) * Bao gồm phí vận chuyển và các khoản giảm giá (nếu có)  2. Xử lý callback ZaloPay:  * Xác thực MAC signature với key2 * Parse dữ liệu JSON từ callback * Cập nhật trạng thái thanh toán trong database  3. Xử lý lỗi thanh toán:  * Nếu ZaloPay trả về lỗi, hiển thị thông báo lỗi * Cho phép khách hàng thử lại hoặc chọn phương thức thanh toán khác * Log lỗi để debug và hỗ trợ khách hàng  4. Xử lý redirect thủ công:  * Nếu ZaloPay không tự động redirect, khách hàng có thể click nút "Quay về Luna Bakery" * Hệ thống kiểm tra trạng thái thanh toán và hoàn tất đơn hàng * Hiển thị hướng dẫn chi tiết cho khách hàng | |

#### Đặt phòng

Bảng 2. Yêu cầu chức năng đặt phòng

|  |  |
| --- | --- |
| **Tên trường hợp sử dụng**: Đặt phòng | ID: UC\_04 |
| Tác nhân chính: Khách hàng | Mức độ cần thiết: Bắt buộc |
| Phân loại: Phức tạp |
| Các thành phần tham gia và mối quan tâm:  **-** Khách hàng: Muốn đặt phòng homestay theo ngày check-in, check-out, số lượng khách và thông tin cá nhân. | |
| Mô tả tóm tắt: Chức năng "Đặt phòng" cho phép khách hàng lựa chọn homestay, nhập thông tin đặt phòng (ngày nhận/trả phòng, số lượng khách), xác nhận giá và tiến hành đặt phòng thành công. | |
| Các mối quan hệ:  +Association (kết hợp): Khách hàng  +Include (bao gồm): Thanh toán trực tuyến  +Extend (mở rộng): Xem lịch sử đặt phòng  +Generalization (tổng quát hóa): Không có | |
| Luồng xử lý bình thường của sự kiện:  1. Khách hàng đăng nhập vào hệ thống.  2. Hệ thống hiển thị danh sách homestay.  3. Khách hàng chọn homestay muốn đặt.  4. Hệ thống hiển thị trang chi tiết homestay với các thông tin: Mô tả homestay, giá phòng (ngày thường, cuối tuần), các tiện nghi, chính sách đặt phòng.  5. Khách hàng nhập thông tin đặt phòng: Ngày check-in, check-out, số lượng khách, số lượng phòng.  6. Hệ thống tính toán tổng chi phí, bao gồm: Giá cơ bản (ngày thường, cuối tuần), giảm giá (nếu có).  7. Khách hàng xác nhận thông tin đặt phòng và nhập thông tin cá nhân (tên, số điện thoại, email).  8. Khách hàng bấm nút “Xác nhận đặt phòng”.  9. Hệ thống lưu thông tin đơn đặt phòng và gửi email xác nhận. | |
| Các luồng sự kiện con (Subflows):  1. Tính giá đặt phòng:  - Tính giá dựa trên số đêm lưu trú, số lượng khách, giá ngày thường, cuối tuần.  - Áp dụng các mức giảm giá (nếu có).  2. Gửi email xác nhận: Nội dung email bao gồm mã đơn đặt phòng, thông tin homestay, ngày check-in, check-out, tổng chi phí. | |
| Luồng luân phiên/đặc biệt (Alternate/Exceptional flows): Không có | |

### Các trường hợp sử dụng của chủ homestay

* + 1. **Sơ đồ trường hợp sử dụng**
* Sơ đồ trường hợp sử dụng của củ homestay …

#### Thêm phòng

Bảng 3. Yêu cầu chức năng thêm phòng

|  |  |
| --- | --- |
| **Tên trường hợp sử dụng**: Thêm phòng | ID: UC\_09 |
| Tác nhân chính: Chủ cửa hàng | Mức độ cần thiết: Bắt buộc |
| Phân loại: Phức tạp |
| Các thành phần tham gia và mối quan tâm:  **-** Chủ homestay: Muốn thêm phòng mới vào homestay để khách hàng có thể đặt phòng. | |
| Mô tả tóm tắt: Chức năng “Thêm phòng” cho phép chủ homestay tạo và quản lý thông tin chi tiết của từng phòng trong homestay, bao gồm tên phòng, kích thước, giá phòng, giảm giá, và các tiện nghi đi kèm. | |
| Các mối quan hệ:  +Association (kết hợp): Chủ homestay  +Include (bao gồm): Không có  +Extend (mở rộng): Không có  +Generalization (tổng quát hóa): Quản lý phòng | |
| Luồng xử lý bình thường của sự kiện:  1. Chủ homestay đăng nhập vào trang quản lý.  2. Hệ thống hiển thị danh sách homestay của chủ homestay.  3. Chủ homestay chọn homestay muốn thêm phòng.  4. Chủ cửa hàng bấm vào nút “Thêm phòng mới”.  5. Hệ thống hiển thị biểu mẫu nhập thông tin phòng bao gồm: Tên phòng, kích thước phòng, giá phòng, giảm giá, tiện nghi của phòng  6. Chủ cửa hàng điền đầy đủ thông tin và bấm nút “Lưu”.  7. Hệ thống kiểm tra tính hợp lệ của dữ liệu:  Nếu hợp lệ: Lưu thông tin phòng mới vào cơ sở dữ liệu và cập nhật danh sách phòng.  Nếu không hợp lệ: Hiển thị thông báo lỗi yêu cầu chỉnh sửa thông tin.  8. Hệ thống điều hướng trở lại trang danh sách phòng, hiển thị phòng mới vừa được thêm. | |
| Các luồng sự kiện con (Subflows): Không có. | |
| Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):  1. Nếu thiếu thông tin bắt buộc:  - Hệ thống hiển thị thông báo lỗi, yêu cầu nhập lại.  2. Nếu giá phòng không hợp lệ (ví dụ: giá âm hoặc bằng 0):  - Hệ thống hiển thị cảnh báo và yêu cầu nhập lại giá hợp lệ. | |

## Các yêu cầu phi chức năng

Tham khảo thêm Công văn hướng dẫn yêu cầu phi chức năng   
https://mst.gov.vn/van-ban-phap-luat/13659.htm

### Yêu cầu thực thi

- NFRQ1: Hệ thống phải đảm bảo tốc độ xử lý nhanh, thời gian phản hồi thấp (dưới 200ms khi tải thấp và dưới 3s khi tải cao).

- NFRQ2: Tương thích với các trình duyệt web phổ biến (Chrome, Firefox, Safari, Edge).

- ….

### Yêu cầu an toàn

- NFRQ4: Các dữ liệu trên hệ thống phải đảm bảo tính chính xác về mặt thông tin.

- NFRQ5: Hệ thống không chưa virus, không chứa các đường link độc hại.

- ….

### Yêu cầu bảo mật

- NFRQ7: Áp dụng cơ chế xác thực mạnh mẽ với mật khẩu và mã xác thực để đảm bảo chỉ có người dùng hợp lệ mới có thể truy cập hệ thống

- NFRQ8: Hệ thống phải đảm bảo rằng chỉ những người dùng được phân quyền mới có thể truy cập và xử lý dữ liệu nhạy cảm.

- …

# CƠ SỞ LÝ THUYẾT

### 1. Backend Technology Stack

#### Node.js và Express.js

* **Node.js**: Runtime environment cho JavaScript server-side
* **Express.js**: Web framework cho Node.js, cung cấp routing, middleware, và API endpoints
* **Lý do lựa chọn**: Hiệu suất cao, cộng đồng lớn, phù hợp cho real-time applications

#### Database: MongoDB

* **MongoDB**: NoSQL database với schema linh hoạt
* **Mongoose ODM**: Object Document Mapper cho MongoDB
* **Lý do lựa chọn**: Phù hợp với dữ liệu phi cấu trúc, scalability tốt

#### Authentication: JWT

* **JSON Web Token**: Stateless authentication mechanism
* **bcrypt**: Password hashing và security
* **Lý do lựa chọn**: Bảo mật cao, không cần session storage

### 2. Frontend Technology Stack

#### React 19

* **React**: JavaScript library cho building user interfaces
* **React Router v7**: Client-side routing
* **Lý do lựa chọn**: Component-based architecture, virtual DOM, ecosystem phong phú

#### State Management: Redux Toolkit

* **Redux Toolkit**: Modern Redux với simplified API
* **RTK Query**: Data fetching và caching
* **Lý do lựa chọn**: Predictable state management, developer tools

#### UI Framework: Tailwind CSS v4

* **Tailwind CSS**: Utility-first CSS framework
* **Responsive Design**: Mobile-first approach
* **Lý do lựa chọn**: Rapid development, consistent design system

#### Build Tool: Vite

* **Vite**: Modern build tool với fast hot module replacement
* **ESLint**: Code quality và formatting
* **Lý do lựa chọn**: Development speed, optimized production builds

### 3. Real-time Communication

#### Socket.IO

* **WebSocket**: Real-time bidirectional communication
* **Room-based messaging**: Order status updates
* **Lý do lựa chọn**: Real-time notifications, live order tracking

### 2.2.4. Payment Integration

#### PayPal và ZaloPay

* **PayPal**: International payment gateway
* **ZaloPay**: Vietnamese payment solution
* **Lý do lựa chọn**: Market coverage, security standards

Bảng 4. So sánh DDD và Clean Architecture

|  |  |  |
| --- | --- | --- |
| **Đặc điểm** | **Domain-Driven Design** | **Clean Architecture** |
| **Bản chất** | Một nguyên lý, cách tiếp cận để mô hình hóa nghiệp vụ. | Một kiến trúc phần mềm giúp tổ chức mã nguồn. |
| **Mục tiêu** | Tập trung vào mô hình hóa nghiệp vụ và làm sao để phần mềm phản ánh đúng logic thực tế. | Tập trung vào cấu trúc mã nguồn, đảm bảo hệ thống có cấu trúc rõ ràng, dễ bảo trì và mở rộng. |
| **Cấu trúc** | Chia hệ thống theo Bounded Context, với các Entity, Value Object, Domain Service. | Chia hệ thống thành các lớp (Layers/Vòng tròn): Core, Application, Infrastructure, UI. |
| **…** |  |  |

## Kiến trúc ứng dụng theo Domain-Driven Design

Mô hình kiến trúc ứng dụng sử dụng kiến trúc nhiều lớp (Layered Architecture) theo DDD, bao gồm Domain Layer, Application Layer, Infrastructure Layer và User Interface Layer.

…

# THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP ĐỂ PHÁT TRIỂN HỆ THỐNG ĐẶT PHÒNG HOMESTAY

## Kiến trúc hệ thống

A diagram of a software application

AI-generated content may be incorrect.

Hình 2. Mô hình tổng quan kiến trúc hệ thống đặt phòng homestay

Hệ thống đặt phòng homestay được xây dựng theo mô hình Client-Server, phân chia rõ ràng giữa các thành phần để đảm bảo tính độc lập và dễ dàng trong việc quản lý, bảo trì và mở rộng. Hệ thống bao gồm các thành phần chính như sau:

### Web Client

Client của hệ thống đặt phòng homestay bao gồm ba loại thiết bị truy cập chính:

* PC, laptop: Người dùng có thể truy cập vào hệ thống thông qua trình duyệt web, nơi họ có thể thực hiện các thao tác như tìm kiếm và đặt phòng homestay, quản lý tài khoản và xem thông tin đặt phòng.
* Postman: …
* Mobile: ….

### Web Server

Trong hệ thống, Web Server đóng vai trò trung tâm xử lý các yêu cầu từ phía Client. …. Web Server được xây dựng với kiến trúc gồm hai phần chính: *Frontend* và *Backend*, ….

* + 1. **Frontend**

Phần giao diện người dùng được phát triển bằng React kết hợp với TypeScript, ….

* + 1. **Backend**

Phần xử lý nghiệp vụ phía sau được phát triển bằng Spring Boot, cho phép xây dựng các API theo kiểu RESTful, đồng thời áp dụng nguyên lý thiết kế Domain-Driven Design (DDD) để phân chia rõ ràng giữa các tầng nghiệp vụ, cơ sở hạ tầng và giao diện.

….

### Database Server

Hệ thống sử dụng PostgreSQL và Redis để quản lý và tối ưu hóa lưu trữ dữ liệu:

* PostgreSQL: Là cơ sở dữ liệu quan hệ chính của hệ thống, được sử dụng để lưu trữ các thông tin cốt lõi. ….
* Redis: Được sử dụng làm ....

### Dịch vụ bên thứ ba

Hệ thống đặt phòng homestay tích hợp nhiều dịch vụ bên thứ ba nhằm nâng cao tính năng và tối ưu trải nghiệm người dùng:

* VNPay: Tích hợp với cổng thanh toán trực tuyến VNPay giúp người dùng thực hiện các giao dịch thanh toán nhanh chóng và an toàn.
* Cloudinary: ….
* OpenStreetMap: ….
* ….

## Thiết kế dữ liệu

### Mô hình dữ liệu



Hình 3. Mô hình dữ liệu mức quan niệm (CDM)

….

Hình 4: Mô hình dữ liệu mức vật lý (PDM)

### Từ điển dữ liệu

#### Bảng permission

Lưu thông tin về các quyền (permission) trong hệ thống, phục vụ cho cơ chế phân quyền và kiểm soát truy cập đối với các API.

Bảng 5 Bảng permission

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| 1 | permission\_id | Int | Khóa chính, Không rỗng | ID duy nhất của mỗi quyền. |
| 2 | permission\_name | Varchar (50) | Không rỗng | Tên của quyền, mô tả ngắn gọn về quyền hạn (ví dụ: "Manage Users", "View Reports"). |
| 3 | api\_path | Varchar (30) | Không rỗng | Đường dẫn (endpoint) của API mà quyền này áp dụng (ví dụ: /api/v1/users). |
| … |  |  |  |  |

#### Bảng users

Bảng user lưu trữ thông tin của người dùng trong hệ thống, bao gồm thông tin đăng nhập, thông tin cá nhân cơ bản và liên kết với các vai trò, tài khoản ngân hàng và hồ sơ chi tiết.

Bảng 6 Bảng users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ghi chú** | **Mô tả** |
| 1 | user\_id | Int | Khóa chính, Không rỗng | ID duy nhất xác định mỗi người dùng. |
| 2 | bank\_id | Int | Khóa ngoại | Mã tài khoản ngân hàng liên kết với người dùng. |
| 3 | profile\_id | Int | Khóa ngoại | Mã hồ sơ chi tiết người dùng. |
| 4 | role\_id | Int | Khóa ngoại | Vai trò người dùng trong hệ thống. |
| 5 | email | Varchar (100) | Không rỗng, Duy nhất | Địa chỉ email dùng để đăng nhập. |
| 6 | password | Text | Không rỗng | Mật khẩu của người dùng (được mã hóa). |
| … |  |  |  |  |

…

## Thiết kế theo chức năng

Trong phần này, các chức năng cốt lõi và đặt trưng của ứng dụng sẽ được trình bày một cách chi tiết.

### Chức năng lên kế hoạch du lịch với AI

* **Mục đích:** Cho phép người dùng lên kế hoạch chi tiết cho chuyến du lịch bao gồm nơi lưu trú (homestay) và các địa điểm tham quan phù hợp với ngân sách và thời gian.
* **Giao diện:**

A screenshot of a chat

AI-generated content may be incorrect.

Hình 5 Giao diện lên kế hoạch du lịch với AI

* **Các thành phần trong giao diện:**

Bảng 7 Các thành phần trong giao diện Lên kế hoạch du lịch với AI

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Loại điều khiển** | **Nội dung** | **Lưu ý** |
| 1 | Select | Chọn tỉnh, thành phố |  |
| 2 | Select | Chọn quận, huyện | Chọn tỉnh / thành phố trước khi chọn quận / huyện |
| 3 | Select | Chọn phường, xã | Chọn quận / huyện trước khi chọn phường / xã |
| 4 | Input | Nhập số ngày cho chuyến đi | Số ngày lớn hơn 0 |
| 5 | Slider | Chọn khoảng ngân sách |  |
| 6 | Button | Số người đi cùng: 1 mình |  |
| 7 | Button | Số người đi cùng: cặp đôi |  |
| 8 | Button | Số người đi cùng: gia đình |  |
| 9 | Button | Số người đi cùng: nhóm bạn |  |
| 10 | Button | Quay lại | Bấm để quay lại trang chủ |
| 11 | Button | Bắt đầu tạo kế hoạch du lịch | Bấm để bắt đầu tạo kế hoạch du lịch bằng AI |

* **Dữ liệu sử dụng:**

Bảng 8 Dữ liệu sử dụng trong Lên kế hoạch du lịch với AI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên bảng / Cấu trúc dữ liệu** | **Phương thức** | | | |
| **Thêm** | **Sửa** | **Xóa** | **Truy vấn** |
| 1 | Homestay |  |  |  | X |
| 2 | City |  |  |  | X |
| 3 | District |  |  |  | X |
| 4 | Room |  |  |  | X |
| 5 | trip\_plans (ElasticSearch) | X |  |  |  |

* **Cách xử lý:** Tạo lịch trình du lịch tự động dựa trên thông tin người dùng cung cấp bằng cách sử dụng Gemini AI để đề xuất homestay và các địa điểm du lịch theo từng ngày. Cụ thể như sau:



Hình 6. Sơ đồ cách xử lý Lên kế hoạch du lịch với AI

### Chức năng đặt phòng homestay

* **Mục đích:** Cho phép khách hàng lựa chọn homestay, nhập thông tin đặt phòng (ngày nhận/trả phòng, số lượng khách), xác nhận giá và tiến hành đặt phòng thành công.
* **Giao diện: …**
* **Các thành phần trong giao diện:**
* **Dữ liệu sử dụng:**
* **Cách xử lý:**

# KIỂM THỬ VÀ ĐÁNH GIÁ CHO HỆ THỐNG ĐẶT PHÒNG HOMESTAY

## Giới thiệu

### Mục tiêu kiểm thử

Kiểm thử “Hệ thống đặt phòng homestay” nhằm thực hiện các mục tiêu sau:

* …

### Phạm vi kiểm thử

Quy trình kiểm thử được thực hiện qua những công đoạn như sau:

* Thực hiện kiểm thử các trường hợp tương ứng với yêu cầu chức năng được mô tả trong tài liệu này.
* Kiểm thử thiết kế: kiểm tra giao diện thiết kế có đúng với đặc tả.
* …

## Chi tiết kế hoạch kiểm thử

### Các yêu cầu được kiểm thử

*Về kiểm thử các yêu cầu chức năng:*

* Quản trị viên thêm tài khoản mới.
* …

*Về kiểm thử các yêu cầu phi chức năng:*

* Sử dụng công cụ kiểm thử tự động Apache Bench để kiểm thử tốc độ phản hồi của API trên hệ thống.
* …

### Cách tiếp cận

…

### Tiêu chí kiểm thử thành công / thất bại

…

### Tiêu chí đình chỉ và yêu cầu bắt đầu lại

…

### Sản phẩm bàn giao của kiểm thử

…

### Môi trường kiểm thử

* Phần cứng: …
* Phần mềm: ….
* Công cụ kiểm thử tự động: Apache Bench. Công cụ này được sử dụng để kiểm thử các yêu cầu phi chức năng về khả năng chịu tải và tốc độ phản hồi.

### Tài nguyên

* Công cụ hỗ trợ: …
* Thiết bị: …

### Các rủi ro

* …

## Các trường hợp kiểm thử

### Kiểm thử chức năng chủ homestay thêm phòng mới

Bảng 9 Bảng kiểm tra chức năng chủ homestay thêm phòng mới

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã số** | **Mô tả** | **Dữ liệu kiểm thử** | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Đánh giá** |
| TC\_TPM\_01 | Kiểm thử chức năng khi chủ homestay thêm phòng mới với các trường thông tin hợp lệ. | Tên phòng: Phòng đôi VIP  Sức chứa: 2  Số giường: 1  Giá ngày thường: 500.000  Giá cuối tuần: 700.000  Trạng thái: Còn trống | 1. Đăng nhập vào trang quản lý.  2. Chọn homestay cần thêm phòng.  3. Chọn chức năng "Thêm phòng".  4. Nhập thông tin phòng mới.  5. Nhấn nút "Lưu". | Hiển thị thông báo "Thêm phòng thành công", phòng mới được thêm vào danh sách phòng của homestay. | Hiển thị thông báo "Thêm phòng thành công", phòng mới được thêm vào danh sách phòng của homestay. | Thành công |
| TC\_TPM\_02 | Kiểm thử chức năng khi chủ homestay thêm phòng mới với tên phòng bị trùng. | Tên phòng: Phòng đôi VIP  Sức chứa: 3  Số giường: 2  Giá ngày thường: 600.000  Giá cuối tuần: 800.000  Trạng thái: Còn trống | 1. Đăng nhập vào trang quản lý.  2. Chọn homestay cần thêm phòng.  3. Chọn chức năng "Thêm phòng".  4. Nhập thông tin phòng mới.  5. Nhấn nút "Lưu". | Hiển thị thông báo "Tên phòng đã tồn tại", không cho phép thêm phòng. | Hiển thị thông báo "Tên phòng đã tồn tại", không cho phép thêm phòng. | Thành công |
| TC\_TPM\_03 | Kiểm thử chức năng khi chủ homestay thêm phòng mới với sức chứa không hợp lệ (nhỏ hơn 0). | Tên phòng: Phòng đơn  Sức chứa: -1  Số giường: 1  Giá ngày thường: 300.000  Giá cuối tuần: 400.000  Trạng thái: Còn trống | 1. Đăng nhập vào trang quản lý.  2. Chọn homestay cần thêm phòng.  3. Chọn chức năng "Thêm phòng".  4. Nhập thông tin phòng mới.  5. Nhấn nút "Lưu". | Hiển thị thông báo "Sức chứa không hợp lệ", không cho phép thêm phòng. | Hiển thị thông báo "Sức chứa không hợp lệ", không cho phép thêm phòng. | Thành công |
| … |  |  |  |  |  |  |

### Kiểm thử chức năng khách hàng đặt phòng

Bảng 10 Bảng kiểm tra chức năng khách hàng đặt phòng

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mã số** | **Mô tả** | **Dữ liệu kiểm thử** | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Đánh giá** |
| TC\_DP\_01 | Kiểm thử chức năng đặt phòng thành công với thông tin hợp lệ. | Homestay: Homestay A  Phòng: Phòng đôi VIP  Ngày đến: 20/10/2023  Ngày đi: 22/10/2023  Số khách: 2  Thông tin khách hàng: Họ tên, email, số điện thoại | 1. Chọn homestay và phòng.  2. Chọn ngày đến và ngày đi.  3. Nhập số khách và thông tin khách hàng.  4. Nhấn nút "Đặt phòng". | Hiển thị thông báo "Đặt phòng thành công", gửi email xác nhận đặt phòng. | Hiển thị thông báo "Đặt phòng thành công", gửi email xác nhận đặt phòng. | Thành công |
| TC\_DP\_02 | Kiểm thử chức năng đặt phòng khi phòng đã được đặt trong khoảng thời gian đó. | Homestay: Homestay A  Phòng: Phòng đôi VIP  Ngày đến: 20/10/2023  Ngày đi: 22/10/2023  Số khách: 2  Thông tin khách hàng: Họ tên, email, số điện thoại | 1. Chọn homestay và phòng.  2. Chọn ngày đến và ngày đi (trùng với khoảng thời gian đã đặt).  3. Nhập số khách và thông tin khách hàng.  4. Nhấn nút "Đặt phòng". | Hiển thị thông báo "Phòng đã được đặt trong khoảng thời gian này", không cho phép đặt phòng. | Hiển thị thông báo "Phòng đã được đặt trong khoảng thời gian này", không cho phép đặt phòng. | Thành công |
| TC\_DP\_03 | Kiểm thử chức năng đặt phòng khi số khách vượt quá sức chứa của phòng. | Homestay: Homestay A  Phòng: Phòng đơn  Ngày đến: 20/10/2023  Ngày đi: 22/10/2023  Số khách: 2  Thông tin khách hàng: Họ tên, email, số điện thoại | 1. Chọn homestay và phòng.  2. Chọn ngày đến và ngày đi.  3. Nhập số khách (lớn hơn sức chứa của phòng).  4. Nhấn nút "Đặt phòng". | Hiển thị thông báo "Số khách vượt quá sức chứa của phòng", không cho phép đặt phòng. | Hiển thị thông báo "Số khách vượt quá sức chứa của phòng", không cho phép đặt phòng. | Thành công |
| … |  |  |  |  |  |  |

## Đánh giá kết quả kiểm thử

Bảng 11 Bảng tổng hợp kết quả kiểm thử

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Phân loại** | **Chức năng** | **Số testcase** | **Kết quả** |
| 1 | Kiểm thử chức năng | Chủ homestay thêm phòng | 7 | 7/7 |
| 2 | Khách hàng lên kế hoạch du lịch với AI | 7 | 7/7 |
| 3 | Khách hàng đặt phòng | 6 | 6/6 |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 | Kiểm thử phi chức năng | Kiểm thử tốc độ phản hồi của API trên hệ thống | 3 | 3/3 |
| 10 |  |  |  |
| 11 |  |  |  |

Kiểm thử đã được tiến hành trên ?? kịch bản kiểm thử với tổng cộng ?? test case ….

# PHẦN KẾT LUẬN

## Kết quả đạt được

### Về kiến thức và kỹ năng

Trình bày những kiến thức chuyên môn và kỹ năng thực hành mà sinh viên đã tích lũy được trong quá trình thực hiện đề tài. Có thể bao gồm:

* Hiểu biết về quy trình phát triển phần mềm, công nghệ sử dụng.
* Kỹ năng phân tích, thiết kế hệ thống, lập trình, kiểm thử, triển khai.
* Kỹ năng mềm liên quan: quản lý thời gian, làm việc nhóm, tự học công nghệ mới.
* …

*Lưu ý: Viết ngắn gọn, thể hiện giá trị học tập và trưởng thành chuyên môn của người thực hiện.*

### Về chương trình và khả năng ứng dụng trong thực tiễn

Trình bày kết quả cụ thể của sản phẩm phần mềm hoặc mô hình nghiên cứu:

* Mức độ hoàn thiện của hệ thống (giao diện, tính năng...).
* Khả năng áp dụng vào thực tế (trong doanh nghiệp, nhà trường, người dùng cụ thể...).
* Giá trị sử dụng hoặc tiềm năng cải tiến trong môi trường thực tế.
* …

*Lưu ý: Tập trung vào tính ứng dụng và hiệu quả giải quyết vấn đề mà đề tài đặt ra.*

## Hạn chế và khó khăn

Nêu trung thực những mặt còn hạn chế trong quá trình nghiên cứu và phát triển, như:

* Giới hạn về thời gian, kinh nghiệm, công nghệ.
* Tính năng chưa hoàn thiện, còn lỗi, chưa tối ưu.
* Khó khăn khi khảo sát người dùng, triển khai thực tế...
* …

## Hướng phát triển

Đề xuất các hướng mở rộng hoặc nâng cấp hệ thống trong tương lai:

* Bổ sung chức năng mới, cải thiện hiệu năng, tăng tính bảo mật.
* Nghiên cứu kết hợp công nghệ mới (AI, dữ liệu lớn, tích hợp đa nền tảng...).
* …

*Lưu ý: Nội dung này thể hiện tầm nhìn và sự quan tâm đến giá trị lâu dài của đề tài.*

# TÀI LIỆU THAM KHẢO

1. **Huỳnh Xuân Hiệp, Phan Phương Lan**, *Giáo trình Nhập môn công nghệ phần mềm*, NXB Đại học Cần Thơ, 2011.
2. **…**
3. Tài liệu ReactJS, tại: https://react.dev/, 04/03/2025.
4. …
5. Website đặt phòng Booking, tại: https://www.booking.com/, 04/03/2025.
6. …

…