**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘi**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

⎯⎯⎯⎯⎯⎯✧⎯⎯⎯⎯⎯⎯

A yellow and red logo

Description automatically generated

**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI**

**XÂY DỰNG WEBSITE QUẢN LÝ NHÀ HÀNG CƠM NHÀ**

|  |  |
| --- | --- |
| GVHD | : TS. Phạm Văn Hiệp |
| Sinh viên  Mã sinh viên: | : Vũ Thị Duyên  : 2021605347 |
| Lớp-Khóa | : 2021DHCNTT05 – K16 |

**Hà Nội – 2025**

# MỤC LỤC

[MỤC LỤC 2](#_Toc194259856)

[DANH SÁCH HÌNH ẢNH 4](#_Toc194259857)

[DANH SÁCH BẢNG BIỂU 4](#_Toc194259858)

[MỞ ĐẦU 5](#_Toc194259859)

[1. Lý do chọn đề tài 5](#_Toc194259860)

[2. Mục đích 6](#_Toc194259861)

[3. Đối tượng và phạm vi nghiên cứu 6](#_Toc194259862)

[4. Ý nghĩa khoa học và thực tiễn của đề tài 6](#_Toc194259863)

[CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN 8](#_Toc194259864)

[1.1. Giới thiệu chung 8](#_Toc194259865)

[1.2. Nghiên cứu hiện trạng và giải quyết vấn đề 8](#_Toc194259866)

[1.2.1. Hiện trạng của nhà hàng 8](#_Toc194259870)

[1.2.2. Cơ cấu tổ chức 9](#_Toc194259871)

[1.2.3. Mô tả hoạt động của các bộ phận 9](#_Toc194259872)

[1.2.4. Xác định yêu cầu 10](#_Toc194259873)

[1.3. Giới thiệu về công nghệ, ngôn ngữ sử dụng 12](#_Toc194259874)

[1.3.1. Giới thiệu về ngôn ngữ TypeScript 12](#_Toc194259876)

[1.3.2. Giới thiệu về mô hình MVC 13](#_Toc194259877)

[1.3.3. Cơ sở dữ liệu PostgreSQL 15](#_Toc194259878)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 19](#_Toc194259879)

[2.1. Khảo sát hệ thống 19](#_Toc194259883)

[2.1.1. Mục tiêu 19](#_Toc194259886)

[2.1.2. Giải pháp 19](#_Toc194259887)

[2.1.3. Tổng quan về hệ thống 19](#_Toc194259888)

[2.1.4. Hoạt động của hệ thống 19](#_Toc194259889)

[2.2. Xác định các tác nhân của hệ thống, ca sử dụng 19](#_Toc194259890)

[2.2.1. Các tác nhân 19](#_Toc194259892)

[2.2.2. Các ca sử dụng (usecase) 19](#_Toc194259893)

[2.3. Các biểu đồ 19](#_Toc194259894)

[2.3.1. Biểu đồ use case tổng quát 19](#_Toc194259896)

[2.3.2. Biểu đồ use case phân rã 19](#_Toc194259897)

[2.4. Mô tả chi tiết các Usecase 19](#_Toc194259898)

[2.4.1. Mô tả use case Đăng nhập 19](#_Toc194259900)

[2.5. Xây dựng biểu đồ trình tự 19](#_Toc194259901)

[2.6. Thiết kế biểu đồ lớp chi tiết 19](#_Toc194259902)

[2.7. Cơ sở dữ liệu 19](#_Toc194259903)

[2.7.1. Sơ đồ dữ liệu quan hệ 19](#_Toc194259907)

[2.7.2. Chi tiết các bảng dữ liệu 19](#_Toc194259908)

[2.8. Thiết kế giao diện 19](#_Toc194259909)

[2.8.1. Giao diện use case Đăng nhập 19](#_Toc194259911)

[2.8.1.1. Hình dung màn hình 19](#_Toc194259923)

[2.8.1.2. Biểu đồ lớp màn hình 19](#_Toc194259924)

[2.8.1.3. Biểu đồ cộng tác của các màn hình 19](#_Toc194259925)

[CHƯƠNG 3. CÀI ĐẶT VÀ TRIỂN KHAI 20](#_Toc194259926)

[3.1. Giới thiệu về công cụ và môi trường cài đặt 20](#_Toc194259931)

[3.1.1. Công cụ lập trình Visual Studio Code 20](#_Toc194259934)

[3.1.2. Công cụ thiết kế giao diện Figma 20](#_Toc194259935)

[3.1.3. Môi trường Node.JS 20](#_Toc194259936)

[3.1.4. Thư viện React.JS 20](#_Toc194259937)

[3.1.5. Giới thiệu Nest.JS 20](#_Toc194259938)

[3.2. Trang màn hình giao diện của khách hàng (Front-end) 20](#_Toc194259939)

[3.3. Trang màn hình giao diện của người quản trị (Back-end): 20](#_Toc194259940)

[3.4. Một số giao diện khác 20](#_Toc194259941)

[3.5. Kiểm thử 20](#_Toc194259942)

[3.5.1. Lập kế hoạch kiểm thử 20](#_Toc194259947)

[3.5.2. Thiết kế testcase 20](#_Toc194259948)

[3.5.3. Thực thi testcase 20](#_Toc194259949)

[3.5.4. Báo cáo kiểm thử 20](#_Toc194259950)

[KẾT LUẬN 20](#_Toc194259951)

[TÀI LIỆU THAM KHẢO 20](#_Toc194259952)

# DANH SÁCH HÌNH ẢNH

# DANH SÁCH BẢNG BIỂU

# MỞ ĐẦU

## Lý do chọn đề tài

Trong những năm gần đây, ngành F&B (Thực phẩm và Đồ uống) tại Việt Nam đã phát triển nhanh chóng do sự thay đổi trong thói quen tiêu dùng và sự tiến bộ của công nghệ. Theo báo cáo của Hiệp hội Nhà hàng Việt Nam, thị trường F&B đã chuyển sang sử dụng tự động hóa để điều chỉnh quy trình, giúp giảm chi phí nhân sự và tăng trải nghiệm của khách hàng. Đặc biệt, sau đại dịch COVID-19, mọi người bắt đầu đánh giá cao lựa chọn đặt hàng tự phục vụ để hạn chế tương tác trực tiếp, làm tăng nhu cầu tự động hóa trong ngành.

Việc áp dụng công nghệ trong quản lý nhà hàng không chỉ nâng cao năng suất làm việc của nhân viên, đồng thời cũng nâng cao mức độ hài lòng của khách hàng. Theo truyền thống, việc nhận đơn hàng qua nhân viên phục vụ hoặc lấy đơn hàng theo cách thủ công có khả năng xảy ra sai sót, chậm trễ và kém hiệu quả hoạt động. Quản lý đơn hàng thông minh kết hợp đặt hàng qua mã QR giúp cho nhà hàng dễ dàng vận hành, giảm chi phí hoạt động và tăng lợi nhuận.

Mặc dù đã có nhiều nhà hàng ứng dụng các phần mềm quản lý, nhưng hầu hết vẫn yêu cầu khách hàng tải xuống một ứng dụng cụ thể, điều này có thể rất bất tiện. Hơn nữa, không phải tất cả các hệ thống đều cho phép tạo các combo món ăn với giá cố định hoặc cung cấp giảm giá tự động dựa trên giá trị của một combo. Thêm vào đó, quản lý nhân viên, theo dõi doanh thu và phân tích các món ăn phổ biến vẫn chưa được tối ưu hóa trong nhiều hệ thống hiện có.

Với sự cạnh tranh ngày càng cao trong ngành F&B, các nhà hàng cần một giải pháp vận hành hiệu quả, đơn giản hơn, nhanh chóng hơn và rẻ hơn để triển khai. Xu hướng sử dụng mã QR để đặt món và thanh toán trực tuyến đang gia tăng,  đặc biệt là ở các nhà hàng vừa và nhỏ. Do đó, cần thiết phải phát triển một hệ thống quản lý nhà hàng tích hợp việc đặt hàng qua mã QR để có thể giúp khách hàng chủ động trong việc chọn món, tối ưu quy trình phục vụ, hỗ trợ quản lý hiệu quả hơn, theo dõi đơn hàng thường xuyên và xử lý thanh toán.

Theo báo cáo của Statista (2024), tỷ lệ sử dụng công nghệ QR code trong ngành F&B đã tăng 65% trong vòng 3 năm qua, cho thấy nhu cầu chuyển đổi số trong lĩnh vực này ngày càng lớn. Một nghiên cứu của McKinsey cũng chỉ ra rằng các nhà hàng áp dụng hệ thống đặt món tự động có thể tăng hiệu suất phục vụ lên đến 30% và giảm sai sót khi đặt món hơn 40%. Dựa trên thực tế đó, em quyết định lựa chọn đề tài “Xây dựng website quản lý nhà hàng Cơm Nhà” nhằm giải quyết các vấn đề còn tồn đọng trong quy trình quản lý nhà hàng truyền thống.

## Mục đích

* Xây dựng hệ thống quản lý nhà hàng tích hợp quét mã QR để khách hàng có thể đặt món trực tiếp từ điện thoại mà không cần tải ứng dụng.
* Khách hàng có thể tự chọn món và thanh toán online mà không cần ra quầy thanh toán (chỉ cần xác nhận đã thanh toán với nhân viên).
* Tối ưu quy trình phục vụ, giảm sai sót và tăng tốc độ xử lý đơn hàng.
* Hỗ trợ nhân viên và quản lý dễ dàng theo dõi tình trạng đơn, doanh thu, thống kê món ăn bán chạy.
* Cung cấp giải pháp công nghệ hiện đại cho ngành dịch vụ ăn uống.

## Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: Các nhà hàng, quán ăn có nhu cầu ứng dụng công nghệ vào quy trình phục vụ.

- Phạm vi nghiên cứu:

Tập trung vào việc xây dựng hệ thống đặt món qua QR Code, xử lý đơn hàng, thanh toán và quản lý nhà hàng.

Nhu cầu ăn uống trong nhà hàng và cách thanh toán hiện nay.

## Ý nghĩa khoa học và thực tiễn của đề tài

* **Ý nghĩa khoa học**:
* Nâng cao kiến thức về thương mại điện tử: nghiên cứu và xây dựng Website quản lý nhà hàng Cơm Nhà đóng góp vào việc mở rộng kiến thức về phát triển và quản lý thương mại điện tử trong lĩnh vực kinh doanh đồng hồ.
* Phát triển kỹ năng công nghệ thông tin: dự án này cung cấp cơ hội để áp dụng và phát triển kỹ năng lập trình, thiết kế giao diện, và quản lý dự án web trong môi trường thực tế.
* Tối ưu hóa trải nghiệm người dùng: nghiên cứu về tối ưu hóa trải nghiệm người dùng trên nền tảng thương mại điện tử đóng vai trò quan trọng, mang lại những hiểu biết sâu về cách thiết kế giao diện để tăng tương tác và thu hút khách hàng.
* **Ý nghĩa thực tiễn**:
* Tăng khả năng cạnh tranh của nhà hàng Cơm Nhà: xây dựng Website quản lý nhà hàng giúp mở rộng thị trường và cạnh tranh hiệu quả trong ngành Thực phẩm và đồ uống F&B hiện nay.
* Tiết kiệm thời gian và nguồn lực: khách hàng có thể dễ dàng tìm kiếm, chọn món ăn mà không cần nhờ nhân viên tư vấn hoặc cầm bảng menu rất nhiều món ăn, giúp tiết kiệm thời gian và nguồn lực cho cả nhà hàng và khách hàng.
* Tối ưu hóa quy trình kinh doanh: quản lý quy trình đặt món, theo dõi đơn hàng, và quản lý nhân viên một cách hiệu quả hơn thông qua hệ thống Website, giúp cải thiện quy trình kinh doanh của nhà hàng Cơm Nhà.
* Giảm thiểu sai sót khi order món, tiết kiệm thời gian cho nhân viên.
* Hỗ trợ nhà hàng quản lý hiệu quả, tăng doanh thu nhờ phân tích dữ liệu bán hàng.

# GIỚI THIỆU TỔNG QUAN

## Giới thiệu chung

Nhà hàng Cơm Nhà là một mô hình kinh doanh ẩm thực với phong cách ấm cúng, gần gũi, mang đến cho thực khách những bữa ăn như tại nhà. Với phương châm "Cơm như mẹ nấu – Đậm đà hương vị Việt", nhà hàng chuyên phục vụ các món ăn truyền thống Việt Nam, tập trung vào sự đơn giản nhưng tinh tế, phù hợp với khẩu vị của đa số người Việt.

Nhà hàng Cơm Nhà hướng đến đối tượng khách hàng là gia đình, nhân viên văn phòng, sinh viên và du khách muốn thưởng thức những bữa ăn đậm chất quê hương. Nhà hàng hoạt động theo mô hình gọi món tại chỗ kết hợp quét mã QR để đặt món, giúp khách hàng có trải nghiệm tiện lợi và hiện đại hơn.

Không gian nhà hàng được thiết kế theo phong cách mộc mạc, ấm áp, sử dụng chủ yếu nội thất gỗ và ánh sáng vàng nhẹ tạo cảm giác thân thuộc. Các món ăn được bày trí đơn giản nhưng tinh tế, tạo cảm giác như đang thưởng thức bữa cơm gia đình thực thụ. Nhà hàng phục vụ đa dạng các món ăn truyền thống như: cơm tấm, canh chua, cá kho tộ, thịt kho trứng, rau luộc chấm kho quẹt, cùng nhiều món ăn đặc trưng theo vùng miền. Ngoài các món ăn riêng lẻ, nhà hàng còn cung cấp combo suất ăn dành cho cá nhân, nhóm hoặc gia đình với giá ưu đãi.

Với sự kết hợp giữa ẩm thực truyền thống và công nghệ hiện đại, nhà hàng Cơm Nhà không chỉ mang đến bữa ăn ngon mà còn tối ưu trải nghiệm khách hàng, tạo nên sự khác biệt so với các mô hình kinh doanh truyền thống.

## Nghiên cứu hiện trạng và giải quyết vấn đề



### Hiện trạng của nhà hàng

Nhà hàng Cơm Nhà là một cơ sở kinh doanh ẩm thực chuyên phục vụ các món ăn truyền thống Việt Nam. Tuy nhiên, trong quá trình hoạt động, nhà hàng gặp một số vấn đề trong khâu gọi món, phục vụ và quản lý đơn hàng, cụ thể như sau:

* Gọi món theo phương thức truyền thống: Khách hàng phải chờ nhân viên phục vụ ghi nhận món làm mất thời gian của khách và có thể dễ ghi sai món nếu không xác nhận lại với khách.
* Quy trình phục vụ chưa tối ưu: Khi quán đông, nhân viên phải xử lý nhiều đơn cùng lúc, có thể gây chậm trễ và sai sót.
* Quản lý đơn thủ công: Nhà hàng chưa có hệ thống theo dõi đơn hàng tự động, chủ yếu dựa vào nhân viên bếp và nhân viên phục vụ.
* Thống kê doanh thu thủ công: Việc tính toán doanh thu hàng ngày chưa được tự động hóa, dễ tính sai khi ghi thiếu đơn, gây khó khăn trong quản lý.
* Chưa có hệ thống đặt món online: Khách hàng chưa thể đặt món trực tiếp qua điện thoại mà phải gọi nhân viên, chưa tận dụng được công nghệ.

### Cơ cấu tổ chức

Nhà hàng Cơm Nhà có cơ cấu tổ chức gồm 3 bộ phận chính:

* Bộ phận quản lý: Giám sát hoạt động chung của nhà hàng, kiểm soát tài chính, quản lý nhân sự và chiến lược kinh doanh.
* Bộ phận phục vụ & thu ngân: Nhận order từ khách, thanh toán hóa đơn, đảm bảo khách hàng được phục vụ tốt.
* Bộ phận bếp: Chế biến món ăn theo order từ hệ thống, đảm bảo chất lượng món ăn.

Mỗi bộ phận đều có vai trò quan trọng trong hoạt động chung của nhà hàng, cần có sự phối hợp chặt chẽ để nâng cao hiệu suất làm việc của nhà hàng.

### Mô tả hoạt động của các bộ phận

|  |  |
| --- | --- |
| Bộ phận | Mô tả hoạt động |
| Quản lý | * Điều hành mọi hoạt động của cửa hàng, mở rộng quan hệ hợp tác kinh doanh. * Quản lý nhân sự (chấm công, phân công ca làm cho nhân viên) * Theo dõi tình hình kinh doanh, thống kê doanh thu, lợi nhuận. * Kiểm tra dịch vụ và sản phẩm của nhà hàng. |
| Nhân viên phục vụ | * Phục vụ, mời chào khách hàng tới nhà hàng. * hướng dẫn khách hàng gọi món (quét mã QR Code hoặc nhận order trực tiếp nếu khách hàng yêu cầu). * Nhận order của khách và truyền đơn order cho bộ phận bếp. * Phục vụ món ăn đến bàn của khách. * Hỗ trợ khách trong quá trình dùng bữa tại nhà hàng (món ăn có vấn đề, gọi thêm đồ…). |
| Nhân viên thu ngân | * Nhận thanh toán từ khách hàng (tiền mặt, thẻ, ví điện tử). * Xuất hóa đơn cho khách * Cập nhật doanh thu hàng ngày. * Hỗ trợ khách hàng khi có vấn đề khi thanh toán (giá cả, áp dụng khuyến mãi,…) |
| Nhân viên bếp | * Nhận đơn hàng từ hệ thống và chế biến món ăn. * Kiếm soát chất lượng và số lượng món ăn trước khi giao ra. * Đảo bảo vệ sinh an toàn thực phẩm trong khu vực làm. |

### Xác định yêu cầu

Dựa trên các vấn đề mà nhà hàng Cơm Nhà đang gặp phải, hệ thống quản lý nhà hàng cần phái đáp ứng các yêu cầu sau:

**Yêu cầu chức năng:**

* Đối với khách hàng:
* Quét mã QR để xem thực đơn món ăn, đặt món trực tiếp từ điện thoại của khách hàng.
* Xem thông tin chi tiết các món ăn (tên, giá cả, ghi chú, hình ảnh).
* Thay đổi món nếu chưa phục vụ hoặc chọn thêm món.
* Thanh toán trực tuyến hoặc tại quầy thu ngân.

- Đối với nhân viên phục vụ:

* Nhận đơn hàng trên hệ thống, không cần ghi chép giấy.
* Cập nhật trạng thái của đơn hàng (đã nhận, đang chuẩn bị, thanh toán).
* Hỗ trợ khi khách hàng yêu cầu.

- Đối với nhân viên thu ngân:

* Xác nhận với khách hàng đã thanh toán đơn hàng.
* Xuất hóa đơn điện tử dạng PDF, xuất hóa đơn đưa cho khách.
* Áp dụng mã giảm giá nếu có.
* Cập nhật doanh thu hàng ngày của nhà hàng, theo dõi đơn hàng trong thười gian làm việc.

- Đối với quản lý:

* Theo dõi doanh thu theo ngày/tuần/tháng/quý/năm.
* Quản lý danh sách món ăn, combo, nhân viên và hóa đơn.
* Thống kê món ăn bán chạy, bán chậm để tối ưu hóa thực đơn
* Phản ánh, ghi chép lại những điều cần thay đổi, cần sửa chữa để hệ thống quản lý hoạt động tốt và ổn định.

**Yêu cầu phi chức năng:**

* Giao diện đơn giản, dễ dùng và không yêu cầu khách hàng cài đặt ứng dụng bên ngoài.
* Hệ thống hoạt động ổn định, nhanh chóng, hỗ trợ nhiều người dùng cùng lúc.
* Bảo mật dữ liệu, không để lộ thông tin khách hàng và đơn hàng.
* Hỗ trợ mở rộng: Có thể nâng cấp thêm tính năng khi nhà hàng phát triển.

## Giới thiệu về công nghệ, ngôn ngữ sử dụng



### Giới thiệu về ngôn ngữ TypeScript

TypeScript là một ngôn ngữ lập trình mã nguồn mở được phát triển bởi Microsoft, mở rộng từ JavaScript bằng cách thêm kiểu tĩnh (static typing). Nó được thiết kế để giúp phát triển ứng dụng quy mô lớn dễ dàng hơn, đồng thời vẫn giữ được sự linh hoạt của JavaScript.

TypeScript biên dịch thành JavaScript thuần (plain JavaScript), nghĩa là có thể chạy trên bất kỳ trình duyệt nào hoặc môi trường Node.js.

**Ưu điểm:**

* Kiểu dữ liệu tĩnh: TypeScript cho phép bạn xác định kiểu dữ liệu của biến, tham số và giá trị trả về của hàm, giúp phát hiện lỗi và làm tăng tính ổn định của mã.
* Hỗ trợ ECMAScript mới: TypeScript thường hỗ trợ các tính năng của các phiên bản ECMAScript mới, giúp bạn sử dụng các tính năng mới nhất của JavaScript mà không cần chờ đợi sự hỗ trợ của các trình duyệt.
* Cải thiện tính bảo trì: Kiểu dữ liệu tĩnh và tính năng hướng đối tượng của TypeScript làm cho mã dễ bảo trì hơn, giúp ít lỗi hơn và dễ dàng hiểu hơn.
* Cộng đồng lớn: TypeScript có một cộng đồng lớn và sôi động, với nhiều tài liệu, thư viện và công cụ hỗ trợ.
* Tích hợp với các công cụ phổ biến: TypeScript có thể tích hợp dễ dàng với nhiều công cụ phát triển phổ biến như Visual Studio Code, Webpack và Angular.

**Nhược điểm:**

* Học phần mở rộng: Việc học TypeScript có thể đòi hỏi một thời gian để hiểu và làm quen với các khái niệm như kiểu dữ liệu tĩnh và hướng đối tượng nếu bạn không quen với chúng.
* Tăng thời gian phát triển ban đầu: Việc định nghĩa kiểu dữ liệu có thể làm tăng thời gian phát triển ban đầu so với việc viết mã JavaScript thuần túy.
* Khó khăn trong tích hợp với mã JavaScript cũ: Nếu bạn có một dự án sử dụng mã JavaScript đã tồn tại, việc tích hợp TypeScript có thể gặp khó khăn do cú pháp và cách thức làm việc của TypeScript khác với JavaScript thuần túy.

### Giới thiệu về mô hình MVC

Mô hình MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm được sử dụng rộng rãi trong việc phát triển ứng dụng để tạo ra các giao diện người dùng trực quan và có khả năng tương tác cao. MVC chia một ứng dụng thành 3 phần chính và mỗi phần có một vai trò riêng biệt:

* **Model** (Mô hình): Phụ trách lưu trữ và quản lý dữ liệu của ứng dụng. Model đại diện cho trạng thái hiện tại của dữ liệu và cung cấp các phương thức để truy cập và cập nhật dữ liệu đó.
* **View** (Giao diện): Hiển thị dữ liệu cho người dùng. View nhận dữ liệu từ Model và hiển thị nó dưới dạng phù hợp cho người dùng tương tác.
* **Controller** (Bộ điều khiển): Xử lý tương tác của người dùng với ứng dụng. Controller nhận đầu vào từ người dùng, cập nhật Model khi cần thiết và sau đó thông báo cho View để cập nhật giao diện.

**Luồng xử lý trong MVC**

Luồng xử lý trong của mô hình MVC, bạn có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:

* Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
* Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
* Model chuẩn bị data và gửi lại cho Controller.
* Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

A diagram of a control system

AI-generated content may be incorrect.

Ở đây, View không giao tiếp trực tiếp với Model. Sự tương tác giữa View và Model sẽ chỉ được xử lý bởi Controller.

**Ưu điểm**

* Tách biệt rõ ràng giữa giao diện người dùng và logic nghiệp vụ: Điều này giúp các nhà phát triển có thể làm việc độc lập trên các thành phần khác nhau mà không ảnh hưởng đến những phần khác, làm tăng tính bảo trì và khả năng mở rộng của ứng dụng.
* Phát triển đồng thời: Nhờ sự tách biệt, các nhóm phát triển có thể làm việc cùng lúc trên các thành phần khác nhau của ứng dụng, từ đó tăng tốc độ phát triển.
* Dễ dàng mở rộng: Có thể thêm tính năng mới hoặc sửa đổi hiện có mà không làm ảnh hưởng đến các thành phần khác của hệ thống.
* Thuận tiện cho kiểm thử: Việc tách biệt giúp kiểm thử từng phần của hệ thống trở nên dễ dàng hơn, có thể thực hiện kiểm thử đơn vị cho từng module một cách hiệu quả.
* Dễ dàng quản lý và triển khai code: Tạo điều kiện thuận lợi cho việc quản lý mã nguồn khi ứng dụng phát triển và trở nên phức tạp hơn.

**Nhược điểm**

* Độ phức tạp: Mô hình MVC có thể làm tăng độ phức tạp của ứng dụng, đặc biệt là khi ứng dụng lớn và các tương tác giữa các thành phần trở nên phức tạp.
* Sự phụ thuộc giữa các thành phần: Mặc dù MVC giúp tách biệt các thành phần, nhưng các thành phần này vẫn có sự phụ thuộc lẫn nhau.
* Khó khăn trong việc điều hướng và dữ liệu luân chuyển: Đôi khi việc duy trì dòng chảy dữ liệu rõ ràng giữa ba thành phần có thể trở nên khó khăn, đặc biệt là trong các ứng dụng lớn với nhiều trang và tương tác.
* Quá tải cho Controller: Trong một số trường hợp, Controller có thể trở nên phức tạp và quá tải do phải xử lý nhiều nhiệm vụ.
* Giới hạn linh hoạt: Mặc dù MVC cung cấp cấu trúc rõ ràng, nhưng điều này cũng có thể hạn chế sự sáng tạo trong một số trường hợp.

### Cơ sở dữ liệu PostgreSQL

PostgreSQL (viết tắt từ Postgres Structured Query Language) là một hệ thống quản lý cơ sở dữ liệu quan hệ - đối tượng (object - relational database management system) tiên tiến và mã nguồn mở. Nó đã được phát triển từ phiên bản POSTGRES 4.2 tại phòng khoa học máy tính của Đại học California, Berkeley. Hiện nay, PostgreSQL đã trở thành một lựa chọn quan trọng và đáng tin cậy để quản lý dữ liệu cho các ứng dụng từ cá nhân đến doanh nghiệp lớn.

Hoạt động trên các nền tảng UNIX, PostgreSQL cũng đã được điều chỉnh linh hoạt để tương thích với nhiều hệ điều hành khác như Mac OS X, Solaris và Windows. Là một phần mềm mã nguồn mở, PostgreSQL được cung cấp dưới dạng giấy phép tự do, cho phép người dùng sử dụng, sửa đổi và phân phối mã nguồn một cách tự do.

**Một số tính năng quan trọng của PostgreSQL:**

* Kiểu dữ liệu đa dạng: PostgreSQL hỗ trợ nhiều kiểu dữ liệu như số nguyên, số thực, chuỗi, Boolean, ngày/giờ, mảng, phạm vi, UUID, JSON/JSONB, XML, key-value (Hstore), hình học và các kiểu dữ liệu tùy chỉnh.
* Tính toàn vẹn dữ liệu: PostgreSQL cung cấp các cơ chế để đảm bảo tính toàn vẹn dữ liệu như ràng buộc UNIQUE, NOT NULL, Primary Keys, Foreign Keys và ràng buộc loại trừ. Ngoài ra, nó còn hỗ trợ khóa hàm số/explicit locks và khóa khuyến nghị/advisory locks.
* Đồng quy và hiệu suất: PostgreSQL cung cấp các phương pháp lập danh mục (indexing) như B-tree, multicolumn, expressions, partial và các phương pháp lập danh mục nâng cao như GiST, SP-Gist, KNN Gist, GIN, BRIN, Bloom filters. Nó cũng có trình lập kế hoạch/trình tối ưu hóa truy vấn phức tạp, hỗ trợ quét chỉ trên danh mục (index-only scan) và thống kê đa cột.
* Giao tác và đồng thời: PostgreSQL hỗ trợ giao tác ACID (Atomicity, Consistency, Isolation, Durability) và giao tác dạng nest thông qua lưu điểm (savepoints). Nó cũng hỗ trợ điều khiển đồng thời nhiều phiên bản (MVCC) và truy vấn đọc song song.
* Phân vùng bảng: PostgreSQL cho phép phân vùng bảng để tăng hiệu suất truy vấn và quản lý dữ liệu lớn.
* Độ tin cậy và phục hồi: PostgreSQL sử dụng ghi nhật ký ghi trước (Write-ahead Logging - WAL) để đảm bảo độ tin cậy và hỗ trợ khôi phục điểm-theo-thời gian (Point-in-time Recovery - PITR) và active standbys.
* Bảo mật: PostgreSQL cung cấp các phương pháp xác thực như GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Certificate và hệ thống kiểm soát truy cập mạnh mẽ (hỗ trợ bảo mật cấp độ cột và hàng).
* Khả năng mở rộng: PostgreSQL có khả năng mở rộng thông qua các phương pháp lưu trữ và hỗ trợ nhiều ngôn ngữ thủ tục (PL/PGSQL, Perl, Python và nhiều ngôn ngữ khác). Ngoài ra, nó hỗ trợ tích hợp với các cơ sở dữ liệu và ứng dụng khác thông qua giao diện SQL chuẩn và cung cấp nhiều tiện ích mở rộng bổ sung như PostGIS.
* Tìm kiếm văn bản: PostgreSQL hỗ trợ các bộ ký tự quốc tế thông qua ICU collations và cung cấp khả năng tìm kiếm văn bản toàn diện.

**Ưu điểm:**

* Dễ sử dụng: PostgreSQL cung cấp một cú pháp SQL trực quan và dễ hiểu, giúp người dùng dễ dàng tạo, truy vấn và quản lý cơ sở dữ liệu một cách hiệu quả.
* Hỗ trợ cho ứng dụng web động: PostgreSQL hoạt động tốt với môi trường LAMP (Linux, Apache, MySQL, PHP/Python/Perl), cho phép chạy các trang web và ứng dụng web động một cách linh hoạt và có hiệu suất cao.
* Khả năng lưu trữ nhật ký và khôi phục: PostgreSQL sử dụng ghi nhật ký ghi trước (WAL) để lưu trữ các thay đổi dữ liệu, giúp đảm bảo tính toàn vẹn và khả năng khôi phục dữ liệu khi có lỗi xảy ra.
* Mã nguồn mở miễn phí: PostgreSQL được phát triển dưới giấy phép nguồn mở, cho phép người dùng sửa đổi, tùy chỉnh và triển khai theo nhu cầu của họ mà không phải trả phí cho bản quyền phần mềm.
* Đối tượng hóa dữ liệu: PostgreSQL hỗ trợ đối tượng hóa dữ liệu, cho phép lưu trữ và truy vấn các đối tượng phức tạp như bảng, quan hệ, hình học, mảng và JSON.
* Cộng đồng hỗ trợ đông đảo: PostgreSQL có một cộng đồng người dùng và nhà phát triển rộng lớn, với nhiều kinh nghiệm và kiến thức. Điều này đảm bảo rằng người dùng có sự hỗ trợ và tài liệu phong phú khi gặp vấn đề hoặc cần tư vấn.
* Giảm thiểu công việc bảo trì hệ thống: Nó cung cấp các tính năng như sao lưu và khôi phục dễ dàng, cơ chế tự động tăng cường (autovacuum) để quản lý không gian lưu trữ, cùng nhiều công cụ hỗ trợ quản lý hiệu suất và tối ưu hóa.

**Nhược điểm:**

* Quản lý không thuộc về một tổ chức cụ thể: PostgreSQL không thuộc quyền quản lý của bất kỳ tổ chức nào, điều này có thể gây khó khăn cho người dùng khi muốn tiếp cận với đầy đủ tính năng và tài liệu hỗ trợ.
* Tập trung vào khả năng tương thích: PostgreSQL tập trung vào khả năng tương thích chuẩn SQL, điều này có thể dẫn đến việc thực hiện các thay đổi để cải thiện hiệu suất, yêu cầu nhiều công việc hơn so với MySQL.
* Hỗ trợ ứng dụng nguồn mở hạn chế: Một số ứng dụng nguồn mở không hỗ trợ PostgreSQL mặc dù hỗ trợ MySQL. Điều này có thể gây khó khăn cho việc tích hợp và triển khai các ứng dụng sử dụng PostgreSQL.
* Hiệu suất chậm hơn so với MySQL: Mặc dù PostgreSQL là một hệ quản trị cơ sở dữ liệu mạnh mẽ, tuy nhiên, trong một số trường hợp, hiệu suất của nó có thể chậm hơn so với MySQL, đặc biệt là khi xử lý các tác vụ có tính toán phức tạp.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG



## Khảo sát hệ thống



### Mục tiêu

Đối với người dùng:

Đối với quản lý:

### Giải pháp

### Tổng quan về hệ thống

### Hoạt động của hệ thống

## Xác định các tác nhân của hệ thống, ca sử dụng



### Các tác nhân

### Các ca sử dụng (usecase)

## Các biểu đồ



### Biểu đồ use case tổng quát

### Biểu đồ use case phân rã

## Mô tả chi tiết các Usecase



### Mô tả use case Đăng nhập

## Xây dựng biểu đồ trình tự

## Thiết kế biểu đồ lớp chi tiết

## Cơ sở dữ liệu



### Sơ đồ dữ liệu quan hệ

### Chi tiết các bảng dữ liệu

## Thiết kế giao diện



### Giao diện use case Đăng nhập



#### Hình dung màn hình

#### Biểu đồ lớp màn hình

#### Biểu đồ cộng tác của các màn hình

# CÀI ĐẶT VÀ TRIỂN KHAI



## Giới thiệu về công cụ và môi trường cài đặt



### Công cụ lập trình Visual Studio Code

### Công cụ thiết kế giao diện Figma

### Môi trường Node.JS

### Thư viện React.JS

### Giới thiệu Nest.JS

## Trang màn hình giao diện của khách hàng (Front-end)

trang chủ, đăng ký, thông tin khách hàng, danh sách sản phẩm, chi tiết sản phẩm, giỏ hàng, đặt hàng, lịch sử giao dịch, kiểm tra đơn hàng, tin tức …

## Trang màn hình giao diện của người quản trị (Back-end):

+ Quản lý: Danh mục, Sản phẩm, tài khoản, đơn hàng …

+ Thống kê doanh thu, báo cáo theo tháng, theo năm, SP bán chạy, SP bán chậm…

## Một số giao diện khác

## Kiểm thử



### Lập kế hoạch kiểm thử

### Thiết kế testcase

### Thực thi testcase

### Báo cáo kiểm thử

# KẾT LUẬN

# TÀI LIỆU THAM KHẢO

<https://fptshop.com.vn/tin-tuc/danh-gia/postgresql-la-gi-179964>

[**https://fptshop.com.vn/tin-tuc/danh-gia/mo-hinh-mvc-183671**](https://fptshop.com.vn/tin-tuc/danh-gia/mo-hinh-mvc-183671)