**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 khách hàng: nghiên cứu về tối ưu hóa trải nghiệm khách hàng 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 2 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.

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 và thu ngân | * 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…). * 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,…) |

### 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:

* 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.
* 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 khách hàng 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 khách hàng 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 khách hàng. View nhận dữ liệu từ Model và hiển thị nó dưới dạng phù hợp cho khách hàng tương tác.
* **Controller** (Bộ điều khiển): Xử lý tương tác của khách hàng với ứng dụng. Controller nhận đầu vào từ khách hàng, 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 khách hàng 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 khách hàng 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 khách hàng 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 khách hàng 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 khách hàng 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 khách hàng và nhà phát triển rộng lớn, với nhiều kinh nghiệm và kiến thức. Điều này đảm bảo rằng khách hàng 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 khách hàng 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

Mục tiêu của đề tài dựa trên những kiến thức đã học và tìm hiểu như TypeScript, PostgerSQL, CSS, HTML và phương pháp thiết kế giao diện để hỗ trợ cho việc thiết kế Website quản lý nhà hàng, các đối tượng hỗ trợ trong website như: khách hàng, nhân viên và quản lý.

* **Đối với khách hàng:**
* Cho phép khách hàng có thể tìm kiếm sản phẩm một cách nhanh chóng, hiệu quả về những tiêu chí khác nhau như: tên sản phẩm, danh mục.
* Khách hàng có thể thực hiện một số chức năng của hệ thống Website như: tìm kiếm sản phẩm, xem thông tin sản phẩm, xem danh mục sản phẩm, gọi món bằng quét mã QR, theo dõi đơn đặt món trực tiếp
* Cũng giúp khách hàng giảm thời gian chờ nhân viên, giúp trải nghiệm ăn uống mượt mà hơn.
* **Đối với nhân viên:**
* Nhân viên có thể thực hiện chức năng chỉnh sửa đơn đặt, theo dõi đơn đặt món.
* Quản lý tài khoản cá nhân như: đổi thông tin cá nhân, đổi mật khẩu.
* Một số chức năng khác như: đăng nhập
* **Đối với quản lý:** Người quản lý thực hiện một số chức năng quản lý như:
* Quản lý tập trung món ăn, nhân viên và đơn hàng qua hệ thống.
* Giảm rủi ro ghi sai, nhầm đơn, tiết kiệm thời gian vận hành.
* Theo dõi doanh thu, món ăn bán chạy/chậm để tối ưu thực đơn.
* Tăng hiệu suất làm việc của nhân viên thông qua hệ thống hóa quy trình.
* Quản lý danh mục và quản lý thực đơn: người quản lý có thể thêm, sửa, xóa các thông tin về món ăn và danh mục món ăn.
* Quản lý nhân viên: cho phép người quản lý có thể biết được những thông tin cơ bản của nhân viên để phục vụ cho việc xác nhận đơn đặt món, tạo tài khoản đăng nhập cho nhân viên.
* Quản lý đơn đặt món: cho phép người quản lý xem đơn đặt món, có thể xem chi tiết đơn, xem thông tin bàn đã đặt và trạng thái của đơn đặt món đó. Người quản lý có quyền kiểm duyệt đơn đặt món và hủy đơn.
* Quản lý hóa đơn: cho phép người quản lý xem chi tiết hóa đơn về sản phẩm, số lượng, tổng tiền, ngày giờ thanh toán. Cho phép nhà quản lý lưu hóa đơn để thống kê.
* Thống kê doanh thu: cho phép nhà quản lý thống kê doanh thu bán sản phẩm trong từng khoảng thời gian cụ thể (theo từng ngày, tuần, tháng, năm).

### Giải pháp

Nắm bắt được nhu cầu của cửa hàng và khách hàng, em đã vận dụng các kiến thức đã học để phát triển “Website quản lý nhà hàng Cơm Nhà” . Website sẽ giúp cho nhà hàng đem lại nhiều lợi nhuận, trình bày sản phẩm dễ dàng, còn khách hàng sẽ dễ dàng chọn sản phẩm, giảm thời gian chờ nhân viên, giúp trải nghiệm ăn uống mượt mà hơn. Website quản lý nhà hàng sẽ được phát triển với những giải pháp như:

* Phát triển một website quản lý nhà hàng tích hợp mã QR cho gọi món.
* Mỗi bàn ăn được gán một mã QR riêng, liên kết với website gọi món.
* Giao diện tối ưu cho thiết bị di động, khách quét mã QR để:
* Xem thực đơn
* Chọn món, số lượng
* Gửi đơn hàng
* Theo dõi đơn và gọi phục vụ khi cần
* Nhân viên sử dụng hệ thống để xác nhận đơn, gửi cho bếp, cập nhật trạng thái.
* Quản lý có giao diện riêng để:
* Cập nhật thực đơn, quản lý nhân viên
* Xem báo cáo, thống kê
* Giám sát hoạt động của toàn hệ thống

### Tổng quan về hệ thống

Đây là nhà hàng chuyên cung cấp các dịch vụ phục vụ khách hàng với những trải nghiệm tốt nhất với bữa cơm đậm chất người Việt, cơm mẹ nấu. Nhà hàng luôn luôn cố gắng liên tục đem đến cho khách hàng những trải nghiệm văn hóa và những món ăn truyền thống lâu đời của người Việt.

Hình thức kinh doanh của nhà hàng vẫn theo phương pháp truyền thống là khách đến và gọi món trực tiếp với nhân viên nhà hàng, nhà hàng quản lý sản phẩm (món ăn) vẫn bằng sách menu, trong menu có tất cả các sản phẩm của nhà hàng (món ăn, đồ uống, combo,…). Tuy nhiên, với quy mô nhà hàng lơn như nhà hàng Cơm Nhà và số lượng khách thường xuyên tới dùng món là khá đông, cho nên hình thức hoạt động truyền thống như vậy rất gây mất thời gian, gặp khó khăn trong việc quản lý nhà hàng và cũng ảnh hưởng tới các dịch vụ khác.

Từ những lý do thực tế trên, cùng với sự phát triển của thương mại điện tử trong thời đại công nghệ thông tin ngày nay nên nhà hàng đã nghĩ đến việc xây dựng một Website để giới thiệu và quản lý sản phẩm (món ăn), cũng như giúp khách hàng có sự lựa chọn dễ dàng và thuận tiện hơn trong việc chọn thực đơn tại nhà hàng.

**Hệ thống sẽ bao gồm 3 phân vùng chức năng chính tương ứng với 3 vai trò:**

* Khách hàng:
* Giao diện truy cập bằng mã QR
* Xem thực đơn
* Tìm sản phẩm theo tên
* Gọi món
* Theo dõi đơn
* Nhân viên:
* Đăng nhập hệ thống
* Chỉnh sửa thông tin cá nhân
* Đổi mật khẩu tài khoản
* Xác nhận đơn khách gửi
* Cập nhật tình trạng món
* Ghi nhận thanh toán
* Quản lý:
* Đăng nhập hệ thống quản trị
* Quản lý nhân viên, tài khoản
* Quản lý thực đơn
* Xem thống kê doanh thu, món ăn
* Tạo mã QR cho các bàn

### Hoạt động của hệ thống

* ***Hoạt động chọn thực đơn***: Khách hàng đến nhà hàng và ngồi vào bàn
* Quét mã QR: Khách hàng được cấp một mã QR để quét. Hệ thống sẽ nhận diện bàn và chuyển hướng tới giao diện thực đơn để khách hàng chọn món.
* Chọn món: Khách hàng sẽ xem danh sách thực đơn (có thể theo danh mục), sau đó thêm món ăn vào đơn đặt món.
* Tìm kiếm sản phẩm: Khách hàng có thể tìm kiếm các sản phẩm tìm kiếm theo tên để tiện hơn trong việc lựa chọn thực đơn.
* Gửi đơn: Sau khi chọn xong thì khách hàng sẽ xem lại đơn đặt món và gửi đơn lên hệ thống. hệ thống sẽ gửi đơn đặt món cho hệ thống và nhân viên sẽ nhận và xác nhận đơn.
* Theo dõi trạng thái đơn: Khách hàng có thể theo dõi trạng thái đơn dặt món của bàn mình ngay trên điện thoại vừa gửi đơn đi.
* Thanh toán: Sau khi dùng bữa xong thì khách hàng sẽ thanh toán, khách hàng có thể lựa chọn nhiều hình thức thanh toán khác nhau như thanh toán trực tiếp, thanh toán qua ví điện tử, thanh toán qua tài khoản ngân hàng, ...
* Sử dụng các mã giảm giá: Khách hàng có thể lựa chọn các mã giảm giá khác nhau trong quá trình thanh toán.
* ***Hoạt động khuyến mại*:** Khách hàng có thể xem được tên, thời gian diễn ra chương trình, thông tin về sự kiện và các sản phẩm được giảm giá phân theo mức giá.
* ***Hoạt động nhận thông báo*:** Khách hàng có thể nhận được những thông báo mới nhất về các chương trình khuyến mãi, các voucher giảm giá, các sản phẩm đang được giảm giá ngay trên điện thoại khi quét mã QR xem thực đơn.
* ***Hoạt động thống kê:*** Quản lý có thể thực hiện thống kê các thông tin:
* Sản phẩm
* Hóa đơn
* Doanh thu
* ***Hoạt động xác nhận và thanh toán:***
* Nhân viên thực hiện nhận đơn đặt món được gửi trên hệ thống, sau đó thông báo cho bộ phận bếp.
* Khách yêu cầu thanh toán, nhân viên chọn hình thức thanh toán: thanh toán trực tiếp, thanh toán qua ví điện tử, thanh toán qua tài khoản ngân hàng, ... Sau đó xác nhận thanh toán, đơn được hoàn tất.
* ***Cập nhật thông tin hệ thống:*** Quản lý tiến hành xem, thêm, sửa, xóa các thông tin.
* Tài khoản nhân viên
* Thực đơn
* Đơn đặt hàng

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



### Các tác nhân

|  |  |
| --- | --- |
| Actor | Mô tả |
| Khách hàng | Người đến ăn tại nhà hàng, quét mã QR để xem và gọi món |
| Nhân viên | Theo dõi đơn, giao món, cập nhật trạng thái đơn, thanh toán |
| Quản lý | Quản lý thực đơn, nhân viên, doanh thu, mã giảm giá, theo dõi hoạt động của nhà hàng. |

### Các ca sử dụng (usecase)

|  |  |  |
| --- | --- | --- |
| Tác nhân | Các ca sử dụng | Mô tả |
| Khách hàng | Xem thực đơn theo danh mục | Xem danh sách các món ăn theo danh mục |
| Xem thực đơn theo giá | Xem danh sách món ăn theo sắp xếp của giá món ăn |
| Xem chi tiết món ăn | Xem chi tiết một món ăn nào đó |
| Thêm món ăn vào đơn đặt món | Thêm món ăn nào đó vào đơn đặt món |
| Tìm thực đơn theo tên | Tìm món ăn trên hệ thông theo tên |
| Gửi đơn đặt món | Gửi đơn đặt món đã chọn đến hệ thống |
| Nhân viên | Đăng nhập | Đăng nhập bằng tài khoản do quản lý cấp |
| Xác nhận đơn hàng từ khách | Kiểm tra và xác nhận đơn sau khi khách gửi |
| Xác nhận thanh toán | Ghi nhận hình thức thanh toán: tiền mặt, thẻ, chuyển khoản |
| Xuất hóa đơn | Xuất phiếu đặt món sau khi khách đã thanh toán đơn đặt món |
| Hủy đơn đặt món | Hủy đơn chưa thực hiện (nếu có yêu cầu hợp lý) |
| Đăng xuất | Đăng xuất khỏi tài khoản khi hết ca làm. |
| Quản lý | Đăng nhập hệ thống | Đăng nhập vào quản trị hệ thống qua tài khoản quản lý |
| Quản lý thực đơn | Cập nhật menu, combo, giá. Thêm, sửa, xóa món ăn |
| Quản lý nhân viên | Tạo tài khoản, phân quyền, sửa thông tin nhân viên (thêm, sửa, xóa) |
| Quản lý hóa đơn | Xem, sửa, xóa hóa đơn nếu cần thiết (đơn đặt món đã thanh toán) |
| Quản lý mã giảm giá | Xem, thêm, sửa, xóa mã giảm giá |
| Xem thống kê | Xem báo cáo doanh thu theo thời gian (ngày, tuần, tháng, quý, năm) |

## Các biểu đồ



### Biểu đồ use case tổng quát



### Biểu đồ use case phân rã

Phía Frontend





Phía Backend



## Mô tả chi tiết các Usecase



### Mô tả use case Xem thực đơn theo danh mục

|  |  |  |
| --- | --- | --- |
| Tên use case | | Xem thực đơn theo danh mục |
| Mô tả vắn tắt | | Use case này cho phép khách hàng xem danh sách các món ăn theo danh mục. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi khách hàng đã truy cập vào ứng dụng và hiện lên danh sách danh mục lên trên màn hình gồm tên danh mục từ bảng Categories. Khách hàng chọn một danh mục muốn xem. 2. Hệ thống đọc bảng MenuItems và lấy ra các thông tin bao gồm: tên món ăn, mô tả, giá bán và hình ảnh món ăn và hiển thị ra màn hình. Usecase kết thúc. |
| Luồng rẽ nhánh | 1. Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case. 2. Tại bước 1 trong luồng cơ bản, nếu bảng Categories không có dữ liệu thì hệ thống sẽ hiển thị màn hình sẽ không hiển thị các danh mục. 3. Tại bước 2 trong luồng cơ bản, nếu bảng MenuItems không có dữ liệu thì hệ thống sẽ hiển thị màn hình sẽ không hiển thị thực đơn và use case kết thúc. |
| Các yêu cầu đặc biệt | | Không |
| Tiền điều kiện | | Không |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Xem thực đơn theo giá

|  |  |  |
| --- | --- | --- |
| Tên use case | | Xem thực đơn theo giá |
| Mô tả vắn tắt | | Use case này cho phép khách hàng xem thực đơn được sắp xếp thứ tự theo giá của nó. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi khách hàng đã truy cập với ứng dụng và click vào nút “Lọc” và lọc thực đơn theo giá sắp xếp từ thấp đến cao và ngược lại. Hệ thống đọc bảng MenuItems và lấy ra các thông tin bao gồm: tên món ăn, mô tả, giá bán, thời gian chờ, hình ảnh món ăn và hiển thị ra màn hình. Usecase kết thúc. |
| Luồng rẽ nhánh | Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case. |
| Các yêu cầu đặc biệt | | Không |
| Tiền điều kiện | | Không |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Xem chi tiết món ăn

|  |  |  |
| --- | --- | --- |
| Tên use case | | Xem chi tiết món ăn |
| Mô tả vắn tắt | | Use case này cho phép khách hàng xem chi tiết món ăn ngay trên điện thoại. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi khách hàng đã truy cập với ứng dụng và click vào một món ăn muốn xem bất kì trên màn hình. 2. Hệ thống đọc bảng MenuItems và lấy ra các thông tin bao gồm: tên món ăn, mô tả, giá bán, thời gian chờ, hình ảnh món ăn và hiển thị ra màn hình. Usecase kết thúc. |
| Luồng rẽ nhánh | Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case. |
| Các yêu cầu đặc biệt | | Không |
| Tiền điều kiện | | Không |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Thêm món ăn

|  |  |  |
| --- | --- | --- |
| Tên use case | | Thêm món ăn |
| Mô tả vắn tắt | | Use case này cho phép khách hàng thêm món ăn vào đơn đặt món. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi khách hàng đã truy cập với ứng dụng và click vào một món ăn muốn gọi trên màn hình. 2. Sau đó khách hàng click vào nút “Thêm” để thêm món đó vào đơn đặt món. Usecase kết thúc. |
| Luồng rẽ nhánh | Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi.  Tại bước 1, nếu món ăn đó không phục vụ được thì hệ thống sẽ thông báo “Rấc tiếc món ăn không thể phục vụ! Mời bạn chọn món khác.” Usecase kết thúc. |
| Các yêu cầu đặc biệt | | Không |
| Tiền điều kiện | | Không |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Tìm thực đơn

|  |  |  |
| --- | --- | --- |
| Tên use case | | Tìm thực đơn |
| Mô tả vắn tắt | | Use case này cho phép khách hàng tìm món ăn theo tên. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi khách hàng đã truy cập với ứng dụng và nhập tên của món ăn trên ”Tim kiếm”. 2. Khách hàng click vào nút “Tìm” bên cạnh và hệ thống sẽ trả danh sách các món ăn có tên được tìm bao gồm các thông tin: tên món ăn, mô tả, giá bán, thời gian chờ và hình ảnh món ăn được lấy từ bảng MenuItems . Usecase kết thúc. |
| Luồng rẽ nhánh | Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi.  Tại bước 2, nếu nhà hàng không có món ăn nào trùng tên với tên mà khách hàng đang tìm thì hệ thống sẽ hiển thị thông báo “Xin lỗi, nhà hàng không có món này”. Usecase kết thúc use case. |
| Các yêu cầu đặc biệt | | Không |
| Tiền điều kiện | | Không |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Gửi đơn đặt món

|  |  |  |
| --- | --- | --- |
| Tên use case | | Gửi đơn đặt món |
| Mô tả vắn tắt | | Use case này cho phép khách hàng gửi đơn đặt món cho hệ thống. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi khách hàng đã truy cập với ứng dụng chọn xong danh sách các món ăn muốn gọi. sau đó khách hàng click vào nút “Gửi” để gửi đơn cho hệ thống nhà hàng. Usecase kết thúc. |
| Luồng rẽ nhánh | Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case. |
| Các yêu cầu đặc biệt | | Không |
| Tiền điều kiện | | Không |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Đăng nhập tài khoản nhân viên

|  |  |  |
| --- | --- | --- |
| Tên use case | | Đăng nhập tài khoản nhân viên |
| Mô tả vắn tắt | | Use case này cho phép nhân viên đăng nhập vào tài khoản đã có sẵn trên hệ thống. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi nhân viên truy cập hệ thống và click vào nút “Đăng nhập tài khoản” và chuyển hướng đến màn hình mới và chọn “Nhân viên”. 2. Sau đó nhân viên nhập thông tin tài khoản gồm: email đăng nhập và mật khẩu vào form đăng nhập. 3. Nhân viên click vào nút “Đăng nhập”. 4. Hệ thống đọc bảng Users và kiểm tra thông tin đăng nhập của nhân viên, nếu hợp lệ thì cho phép truy cập vào hệ thống và chuyển đến trang chính, ngược lại thì hiển thị thông báo lỗi và yêu cầu nhân viên nhập lại thông tin đăng nhập. Use case kết thúc. |
| Luồng rẽ nhánh | 1. Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case. 2. Tại bước 2 trong luồng cơ bản, nếu nhân viên không nhập đầy đủ thông tin hoặc vi phạm các ràng buộc (số lượng ký tự, ký tự đặc biệt) hệ thống thông báo lỗi và yêu cầu nhân viên nhập đầy đủ thông tin và theo đúng quy chuẩn. 3. Tại bước 2 trong luồng cơ bản, tên tài khoản không tồn tại trong cơ sở dữ liệu, hệ thống sẽ đưa ra thông báo “Tài khoản không tồn tại” và yêu cầu nhân viên nhập lại. |
| Các yêu cầu đặc biệt | | Mật khẩu nhập vào phải đúng với thông tin đã đăng ký trên hệ thống. |
| Tiền điều kiện | | Tài khoản nhân viên đã được đăng kí trước đó |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Xác nhận thanh toán

|  |  |  |
| --- | --- | --- |
| Tên use case | | Xác nhận thanh toán |
| Mô tả vắn tắt | | Use case này cho phép nhân viên xác nhận thanh toán trên hệ thống. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi nhân viên truy cập hệ thống và click vào một đơn đặt món cần xác nhận thanh toán. 2. Sau đó sẽ chuyển đến màn hình chi tiết đơn đặt món. Để thanh toán, nhân viên chọn nút “Thanh toán” và chọn hình thức thanh toán: tiền mặt, thẻ, chuyển khoản. Use case kết thúc. |
| Luồng rẽ nhánh | 1. Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case. |
| Các yêu cầu đặc biệt | | Nhân viên đãng đăng nhập tài khoản vào hệ thống |
| Tiền điều kiện | | Tài khoản nhân viên đã được đăng kí trước đó |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Xuất hóa đơn

|  |  |  |
| --- | --- | --- |
| Tên use case | | Xuất hóa đơn |
| Mô tả vắn tắt | | Use case này cho phép nhân viên xuất hóa đơn trên hệ thống. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi nhân viên truy cập hệ thống và click vào một đơn hàng đã thanh toán. 2. Nhân viên chọn nút “Xuất hóa đơn” và hệ thống sẽ xuất 1 file PDF là thông tin của hóa đơn đó:mã đơn, danh sách các món, số lượng, giá, ghi chú, mã giảm giá, combo và thời gian xuất hóa đơn. Use case kết thúc. |
| Luồng rẽ nhánh | 1. Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case. 2. Tại bước 2 trong luồng cơ bản, nếu đơn đặt món chưa thanh toán thì không thể xuất hóa đơn của đơn đó. Usecase kết thúc. |
| Các yêu cầu đặc biệt | | Nhân viên đãng đăng nhập tài khoản vào hệ thống |
| Tiền điều kiện | | Đơn đặt món đó phải được thanh toán trước đó |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Hủy đơn đặt món

|  |  |  |
| --- | --- | --- |
| Tên use case | | Hủy đơn đặt món |
| Mô tả vắn tắt | | Use case này cho nhân viên hủy đơn đặt món trên hệ thống. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi nhân viên truy cập hệ thống và click vào một đơn đặt món muốn hủy. 2. Sau đó nhân viên xác nhận đơn và click và nút “Hủy đơn”. Use case kết thúc. |
| Luồng rẽ nhánh | 1. Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi. 2. Tại bước 2 trong luồng cơ bản, nếu đơn đặt món đó đã thanh toán rồi thì hệ thống sẽ báo lỗi “Không thể hủy đơn này”. Usecase kết thúc. |
| Các yêu cầu đặc biệt | | Nhân viên đãng đăng nhập tài khoản vào hệ thống |
| Tiền điều kiện | | Đơn đặt món đó chưa được thanh toán. |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Đăng nhập hệ thống

|  |  |  |
| --- | --- | --- |
| Tên use case | | Đăng nhập hệ thống |
| Mô tả vắn tắt | | Use case này cho phép quản lý đăng nhập vào tài khoản đã có sẵn trên hệ thống. |
| Luồng các sự kiện | Luồng cơ bản | 1. Use case bắt đầu khi quản lý truy cập hệ thống và click vào nút “Đăng nhập” và chuyển hướng đến màn hình mới và chọn “Quản lý”. 2. Sau đó quản lý nhập thông tin tài khoản gồm: email đăng nhập và mật khẩu vào form đăng nhập. 3. Quản lý click vào nút “Đăng nhập”. 4. Hệ thống đọc bảng Users và kiểm tra thông tin đăng nhập của quản lý, nếu hợp lệ thì cho phép truy cập vào hệ thống và chuyển đến trang quản trị viên, ngược lại thì hiển thị thông báo lỗi và yêu cầu quản lý nhập lại thông tin đăng nhập. Use case kết thúc. |
| Luồng rẽ nhánh | 1. Tại bất kỳ bước nào của quá trình thực hiện, nếu không kết nối được với cơ sở dữ liệu thì hiển thị thông báo lỗi và kết thúc use case. 2. Tại bước 2 trong luồng cơ bản, nếu quản lý không nhập đầy đủ thông tin hoặc vi phạm các ràng buộc (số lượng ký tự, ký tự đặc biệt) hệ thống thông báo lỗi và yêu cầu quản lý nhập đầy đủ thông tin và theo đúng quy chuẩn. 3. Tại bước 2 trong luồng cơ bản, tên tài khoản không tồn tại trong cơ sở dữ liệu, hệ thống sẽ đưa ra thông báo “Tài khoản không tồn tại” và yêu cầu quản lý nhập lại. |
| Các yêu cầu đặc biệt | | Mật khẩu nhập vào phải đúng với thông tin đã đăng ký trên hệ thống. |
| Tiền điều kiện | | Tài khoản quản lý đã được lưu trong hệ thống trước đó |
| Hậu điều kiện | | Không |
| Điểm mở rộng | | Không |

### Mô tả use case Quản lý thực đơn

|  |  |  |
| --- | --- | --- |
| Tên use case | | Quản lý thực đơn |
| Mô tả vắn tắt | | Use case này cho phép người quản lý xem và quản lý các thông tin của thực đơn nhà hàng. |
| Các luồng sự kiện | Luồng cơ bản | 1. Use case này bắt đầu khi người quản lý kích chuột vào mục “Quản lý thực đơn” trên thanh menu. Hệ thống hiển thị lên màn hình danh sách các tài khoản nhân viên có trong bảng MenuItems của hệ thống. 2. Người quản lý kích chuột vào 1 món ăn. Hệ thống lấy thông tin chi tiết bao gồm: mã món ăn, tên món ăn, mô tả, giá bán, thời gian chờ và hình ảnh món ăn từ bảng MenuItems và hiển thị lên màn hình. 3. Thêm món ăn: 4. Người quản lý kích vào nút “+” trên màn hình. Hệ thống hiển thị màn hình yêu cầu nhập thông tin mới cho món ăn gồm: tên món ăn, mô tả, giá bán, thời gian chờ và hình ảnh món ăn. 5. Người quản lý kích nút “Save”. Hệ thống sẽ sinh ra một món ăn trong bảng MenuItems và hiển thị món ăn đã được thêm. 6. Sửa thông tin món ăn: 7. Người quản lý kích vào nút “Edit” của một món ăn trên màn hình. Hệ thống sẽ lấy thông tin cũ của món ăn gồm: mã món ăn, tên món ăn, mô tả, giá bán, thời gian chờ và hình ảnh món ăn từ bảng MenuItems và hiện lên màn hình. 8. Người quản lý nhập thông tin mới cho phần tên món ăn, mô tả, giá bán, thời gian chờ và hình ảnh món ăn và kích vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của món ăn trong bảng MenuItems và hiển thị món ăn đã cập nhật. 9. Xóa công việc: 10. Người quản lý sẽ kích vào nút “Xóa” của một món ăn trên màn hình. Hệ thống sẽ hiển thị 1 màn hình yêu cầu xoá. 11. Người quản lý kích vào nút “Đồng ý”. Hệ thống sẽ xoá món ăn khỏi bảng MenuItems và hiển thị danh sách đã cập nhật. Use case kết thúc. |
| Luồng rẽ nhánh | 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo “Lỗi kết nối!” và use case kết thúc. |
| Các yêu cầu đặc biệt | | Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện. |
| Tiền điều kiện | | Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case. |
| Hậu điều kiện | | Không có |
| Điểm mở rộng | | Không có |

### Mô tả use case Quản lý nhân viên

|  |  |  |
| --- | --- | --- |
| Tên use case | | Quản lý nhân viên |
| Mô tả vắn tắt | | Use case này cho phép người quản lý xem và quản lý các thông tin của nhân viên. |
| Các luồng sự kiện | Luồng cơ bản | 1. Use case này bắt đầu khi người quản lý kích chuột vào mục “Quản lý nhân viên” trên thanh menu. Hệ thống hiển thị lên màn hình danh sách các tài khoản nhân viên có trong bảng Users của hệ thống. 2. Người quản lý kích chuột vào 1 tài khoản. Hệ thống lấy thông tin chi tiết bao gồm: mã nhân viên, email, mật khẩu, tên nhân viên, ca làm việc, chức vụ từ bảng Users và hiển thị lên màn hình. 3. Thêm tài khoản: 4. Người quản lý kích vào nút “+” trên màn hình. Hệ thống hiển thị màn hình yêu cầu nhập thông tin mới cho nhân viên gồm: Tên nhân viên, email, mật khẩu, ca làm việc, chức vụ. 5. Người quản lý kích nút “Save”. Hệ thống sẽ sinh ra một tài khoản nhân viên trong bảng Users và hiển thị nhân viên đã được thêm. 6. Sửa thông tin nhân viên: 7. Người quản lý kích vào nút “Edit” của một tài khoản nhân viên trên màn hình. Hệ thống sẽ lấy thông tin cũ của nhân viên gồm: mã nhân viên, email, mật khẩu, tên nhân viên, ca làm việc, chức vụ từ bảng Users và hiện lên màn hình. 8. Người quản lý nhập thông tin mới cho phần mã nhân viên, email, mật khẩu, tên nhân viên, ca làm việc, chức vụ và kích vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của nhân viên trong bảng Users và hiển thị tài khoản nhân viên đã cập nhật. 9. Xóa nhân viên: 10. Người quản lý sẽ kích vào nút “Xóa” của một tài khoản nhân viên trên màn hình. Hệ thống sẽ hiển thị 1 màn hình yêu cầu xoá. 11. Người quản lý kích vào nút “Đồng ý”. Hệ thống sẽ xoá nhân viên khỏi bảng Users và hiển thị danh sách đã cập nhật. Use case kết thúc. |
| Luồng rẽ nhánh | 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo “Lỗi kết nối!” và use case kết thúc. |
| Các yêu cầu đặc biệt | | Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện. |
| Tiền điều kiện | | Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case. |
| Hậu điều kiện | | Không có |
| Điểm mở rộng | | Không có |

### Mô tả use case Quản lý hóa đơn

|  |  |  |
| --- | --- | --- |
| Tên use case | | Quản lý hóa đơn |
| Mô tả vắn tắt | | Use case này cho phép người quản lý xem và quản lý các thông tin của hóa đơn. |
| Các luồng sự kiện | Luồng cơ bản | 1. Use case này bắt đầu khi người quản lý kích chuột vào mục “Quản lý đơn đặt món” trên thanh menu. Hệ thống hiển thị lên màn hình danh sách các hóa đơn có trong hệ thống. 2. Người quản lý kích chuột vào 1 hóa đơn. Hệ thống lấy thông tin chi tiết bao gồm: mã hóa đơn, số lượng, giá, ghi chú, mã đơn đặt món, món ăn, combo, mã giảm giá từ bảng OrderItems và hiển thị lên màn hình. 3. Sửa thông tin hóa đơn: 4. Người quản lý kích vào nút “Edit” của một hóa đơn trên màn hình. Hệ thống sẽ lấy thông tin cũ của hóa đơn gồm: mã hóa đơn, số lượng, giá, ghi chú, mã đơn đặt món, món ăn, combo, mã giảm giá từ bảng OrderItems và hiện lên màn hình. 5. Người quản lý nhập thông tin chỉnh sửa cho phần số lượng, giá, ghi chú, món ăn, combo, mã giảm giá và kích vào nút “Cập nhật”. Hệ thống sẽ sửa thông tin của hóa đơn trong bảng OrderItems và hiển thị hóa đơn đã cập nhật. 6. Xóa hóa đơn: 7. Người quản lý sẽ kích vào nút “Delete” của một hóa đơn trên màn hình. Hệ thống sẽ hiển thị 1 màn hình yêu cầu xoá. 8. Người quản lý kích vào nút “Đồng ý”. Hệ thống sẽ xoá hóa đơn khỏi bảng OrderItems và hiển thị danh sách đã cập nhật. Use case kết thúc. |
| Luồng rẽ nhánh | 1. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo “Lỗi kết nối!” và use case kết thúc. 2. Tại bước 4b, nếu quản lý chọn nút “x” thì thông tin hóa đơn vừa sửa sẽ không được lưu lại vào hệ thống. |
| Các yêu cầu đặc biệt | | Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện. |
| Tiền điều kiện | | Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case. |
| Hậu điều kiện | | Không có |
| Điểm mở rộng | | Không có |

## Xây dựng biểu đồ trình tự

## Thiết kế biểu đồ lớp chi tiết

### Xem thực đơn theo danh mục





### Xem thực đơn theo giá





### Xem chi tiết món ăn





### Thêm món ăn





### Tìm thực đơn





### Gửi đơn đặt món





### Đăng nhập tài khoản nhân viên





### Xác nhận thanh toán





### Xuất hóa đơn





### Hủy đơn đặt món





### Đăng nhập hệ thống





### Quản lý thực đơn





### Quản lý nhân viên





### Quản lý hóa đơn

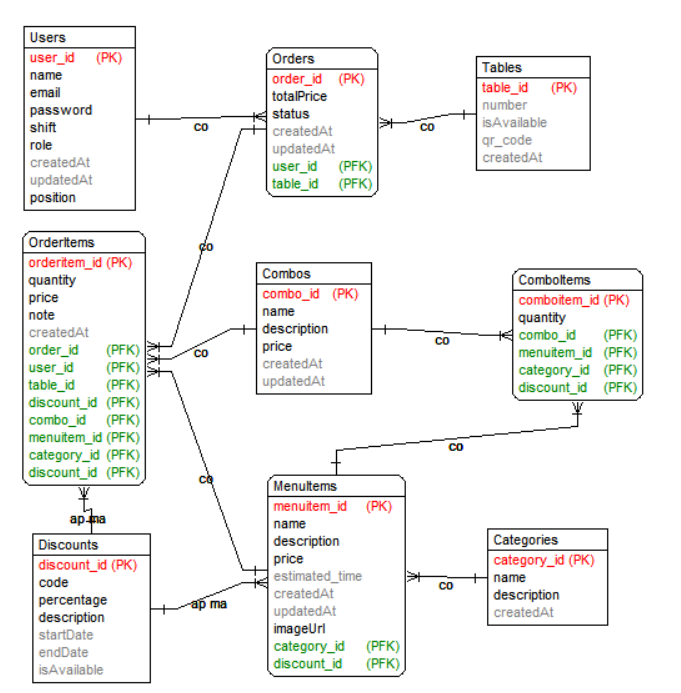
****

****

## Cơ sở dữ liệu



### Sơ đồ dữ liệu quan hệ



### Chi tiết các bảng dữ liệu

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Thiết kế giao diện



### Giao diện use case Xem thực đơn theo danh mục



#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Xem thực đơn theo giá

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Xem chi tiết món ăn

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Thêm món ăn

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Tìm thực đơn

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Gửi đơn đặt món

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Đăng nhập tài khoản nhân viên

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Xác nhận thanh toán

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Xuất hóa đơn

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Hủy đơn đặt món

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Đăng nhập hệ thống

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Quản lý thực đơn

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Quản lý nhân viên

#### Hình dung màn hình

#### Biểu đồ lớp màn hình



### Giao diện use case Quản lý hóa đơn

#### Hình dung màn hình

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