

  
  
**THỰC TẬP CHUYÊN ĐỀ VÀ ĐỒ ÁN CHUYÊN MÔN**

***ĐỀ TÀI***

**<<HỆ THỐNG ĐẶT MÓN TRỰC TUYẾN “FATFOOD”>>**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Th.S Trịnh Văn Chung** |
| **Sinh viên thực hiện:**  **Các thành viên:** | **Nguyễn Hữu Tuấn**  **Nhữ Anh Tuấn**  **Vũ Đỗ Minh Thành** |
| **Mã sinh viên:** | **2310900112**  **2310900111**  **2310900099** |
| **Lớp:** | **K23CNT3** |
| **Khóa:** | **K23** |

**MỤC LỤC**

[**DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ**](#_3i1jkf6y884s)

[**DANH MỤC BẢNG BIỂU**](#_pkdr4jaxtcpf)

[**DANH MỤC HÌNH ẢNH**](#_ki8789rzk01v)

[**CHƯƠNG 1: NGHIÊN CỨU CƠ SỞ LÝ THUYẾT** 1](#_6fedhc83byso)

[1.1.](#_35vaevsszz2d) Tổng quan về lập trình thiết kế giao diện website 1

[1.1.1.](#_2t8h867uhcqh) Giới thiệu về HTML 1

[1.1.2.](#_2oxnxnl1ygv5) Giới thiệu về CSS3 1

[1.1.3.](#_6ystx7ykrjtb) Tổng quan về ngôn ngữ lập trình Javascript 2

[1.1.4.](#_mpo0a6lwjzh5) Thư viện Jquery 2

[1.1.5.](#_1nxklsrmd52w) Framework Bootstrap 2

[1.2.](#_oin3ulg8p2yx) Tổng quan về hệ quản trị cơ sở đữ liệu MySQL 3

[1.2.1.](#_y0u6pm6iifou) Giới thiệu về MySQL 31

[1.2.2.](#_prfzjoqtbm2k) Ưu điểm của MySQL 31

[1.2.3.](#_11gfw9s4bb8u) Nhược điểm của MySQL 31

[1.3.](#_1wmuegh7itv4) Tổng quan về ngôn ngữ lập trình phía máy chủ 41

[1.3.1.](#_kgw2llagmgo7) Giới thiệu về nền tảng công nghệ nguồn mở 41

[1.3.2.](#_g9xb16gsekg0) Giới thiệu về ngôn ngữ lập trình PHP 41

[1.3.3.](#_hh7076m2s2i9) Tổng quan về Framework PHP Laravel 52

[**CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI**](#_pwsvftrq2kgy) **HỆ THỐNG ĐẶT MÓN TRỰC TUYẾN**  [7](#_pwsvftrq2kgy)

[2.1.](#_yqurgccbc9se) Đặt vấn đề 73

[2.2.](#_253o2lxd3a3q) Hệ thống hiện tại 73

[2.3.](#_abl3hlvpc5oh) Hệ thống đề xuất 73

[2.4.](#_yxuyzcp4txt0) Giới hạn của hệ thống 83

[2.5.](#_31a4he2oi9me) Yêu cầu về phần cứng và phần mềm 83

[2.5.1.](#_x6ocex84zwa6) Yêu cầu tối thiểu 83

[2.5.2.](#_d044mt6383gr) Yêu cầu đề nghị 93

[**CHƯƠNG 3:**](#_vkynxkmn83jl) [**PHÂN TÍCH YÊU CẦU KHÁCH HÀNG VÀ THIẾT KẾ HỆ THỐNG** 10](#_s41iob8h5tdd)

[3.1.](#_d4blxuyu4uye) Người dùng của hệ thống 10

[3.2.](#_arzijzoqfmcp) Chức năng của hệ thống 10

3.2.1 Guest(Chức năng dành cho khách vãng lai) 10

3.2.2 Member functions (Chức năng dành cho thành viên) 11

[3.3.](#_4a3t9f4yvty9) Systeme Designs (Thiết kế hệ thống) 13

[3.3.1.](#_ea37abk92jum) Entity Relationship Diagram (Mô hình quan hệ thực thể) 13

[3.4.](#_g0fnub8qhnt5) Database Design (Thiết kế cơ sở dữ liệu) 14

[3.5.](#_5p7bwx61bv5g) Table RelationShip Diagram (Mô hình quan hệ bảng) 23

**3.5.1. Bảng Users**

[3.6.](#_e8gh92fcbg8i) SiteMap (Sơ đồ cấu trúc site) 8

[3.6.1.](#_d6wjc278i8n8) Site map (Cấu trúc trang): Dành cho khách vãng lai 9

[3.6.2.](#_n6xqdy9t6jyg) Sitemap (Cấu trúc trang): Dành cho thành viên 9

[3.6.3.](#_ne36ns2uoihn) Sitemap (Cấu trúc trang): Dành cho quản trị viên 9

[3.7.](#_esxx1o1zphbd) Algorithms (Giải thuật) 10

[3.7.1.](#_6fb16gm7zkex) Đăng ký 10

[3.7.2.](#_msiro2rhwzqs) Đăng nhập 11

[3.7.3.](#_uv7o3ods00ft) Đăng xuất 11

[3.7.4.](#_225w6up0wekh) Đổi mật khẩu 12

[3.7.5.](#_pbpbl0scqrkn) Thay đổi thông tin cá nhân 13

[3.7.6.](#_r3qf9nnw49b3) Thêm sản phẩm 14

[3.7.7.](#_whab65qwl1we) Xóa sản phẩm 15

[3.7.8.](#_6feltooj73qg) Đặt mua 16

[3.7.9.](#_9zbxwr1kd8m2) Tìm kiếm 17

**CHƯƠNG 4: Thiết kế kiến trúc hệ thống & phân công module**

[**TÀI LIỆU THAM KHẢO** 123](#_tism42hxbzku)

# **DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ**

# **DANH MỤC BẢNG BIỂU**

# **DANH MỤC HÌNH ẢNH**

# **CHƯƠNG 1: NGHIÊN CỨU CƠ SỞ LÝ THUYẾT**

## Tổng quan về lập trình thiết kế giao diện website

## Giới thiệu về HTML

HTML5 là phiên bản mới nhất của HyperText Markup Language (HTML) – ngôn ngữ đánh dấu siêu văn bản được sử dụng để xây dựng cấu trúc và nội dung của trang web. HTML5 được phát triển bởi W3C (World Wide Web Consortium) và WHATWG (Web Hypertext Application Technology Working Group) nhằm thay thế các chuẩn HTML/XHTML cũ, cung cấp nền tảng cho các ứng dụng web hiện đại với tính tương thích cao, đa nền tảng và đa thiết bị.

Khác với HTML truyền thống chỉ tập trung vào việc hiển thị nội dung, HTML5 được thiết kế với mục tiêu hỗ trợ các ứng dụng web phong phú (rich web applications), giảm sự phụ thuộc vào các plugin bên ngoài như Adobe Flash, đồng thời tối ưu trải nghiệm người dùng.

***Một số đặc điểm chính của HTML5***:

* Thẻ ngữ nghĩa (Semantic Elements): Bổ sung các thẻ như <header>, <footer>, <section>, <article>, <nav> giúp mô tả cấu trúc nội dung rõ ràng, hỗ trợ SEO và truy cập cho người khuyết tật.
* Hỗ trợ đa phương tiện: Tích hợp sẵn thẻ <audio> và <video> để phát nhạc, video mà không cần cài đặt plugin ngoài.
* Đồ họa và hiệu ứng: Cung cấp thẻ <canvas> và hỗ trợ SVG (Scalable Vector Graphics) để vẽ đồ họa, game, biểu đồ trực tiếp trên trình duyệt.
* Biểu mẫu nâng cao: Bổ sung nhiều loại input mới (email, date, number, range, color...), hỗ trợ kiểm tra dữ liệu (validation) phía client.
* Lưu trữ cục bộ: Hỗ trợ localStorage, sessionStorage, và IndexedDB, cho phép lưu dữ liệu offline trên trình duyệt.
* API mạnh mẽ: Bao gồm Geolocation API (xác định vị trí), Web Workers (xử lý song song), WebSocket (giao tiếp thời gian thực), Drag-and-Drop API…

## Giới thiệu về CSS3

CSS3 được dùng để định dạng và thiết kế giao diện trang web, tạo nên trải nghiệm trực quan và đẹp mắt. Các tính năng sử dụng trong dự án: • Flexbox: Hệ thống bố cục một chiều (hàng hoặc cột), giúp căn chỉnh và phân phối phần tử linh hoạt. • Grid Layout: Cung cấp lưới 2 chiều, cho phép xây dựng layout phức tạp, dễ dàng responsive. • Responsive Design: Sử dụng Media Queries để trang web tương thích với nhiều thiết bị (PC, tablet, mobile). • Animation & Transition: Tạo hiệu ứng chuyển cảnh mượt mà (hover, fade, slide) tăng trải nghiệm người dùng.

## Tổng quan về ngôn ngữ lập trình Javascript

JavaScript là ngôn ngữ lập trình chính trên trình duyệt, giúp trang web trở nên động (dynamic). Phiên bản ES6+ mang đến nhiều cải tiến hiện đại. Các đặc điểm được dùng trong dự án:

• Module: Cho phép chia code thành các file nhỏ, dễ quản lý và tái sử dụng.

• Async/Await: Xử lý bất đồng bộ (asynchronous) gọn gàng, thay thế cho callback phức tạp.

• Fetch API: Dùng để gọi dữ liệu từ server (API) theo chuẩn Promise, thay thế XMLHttpRequest cũ.

## Thư viện Jquery

jQuery là thư viện JavaScript phổ biến, giúp rút ngắn code và thao tác DOM nhanh chóng. Ứng dụng trong dự án:

• Thao tác DOM: Thay đổi nội dung, CSS, hiệu ứng chỉ với vài dòng lệnh.

• AJAX: Gửi và nhận dữ liệu không cần tải lại trang, hỗ trợ các tương tác nhỏ.

• Hiệu ứng (Effects): Ẩn/hiện, fade in/out, slide... giúp cải thiện trải nghiệm người dùng.

## Framework Bootstrap

Bootstrap là framework CSS/JS giúp thiết kế UI nhanh chóng và nhất quán. Các tính năng sử dụng trong dự án:

• Hệ lưới 12 cột (Grid System): Tạo layout responsive cho mọi kích thước màn hình.

• Component UI: Cung cấp sẵn các thành phần như Navbar, Button, Modal, Form, Carousel.

• Utility Classes: Các lớp CSS tiện lợi để căn chỉnh, margin, padding mà không cần viết CSS mới.

• Responsive & Mobile-first: Thiết kế ưu tiên cho thiết bị di động, đảm bảo hiển thị tốt trên mọi nền tảng.

## Tổng quan về hệ quản trị cơ sở đữ liệu MySQL

## Giới thiệu về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở, được phát triển bởi công ty MySQL AB (sau này thuộc Sun Microsystems và hiện nay do Oracle Corporation quản lý). MySQL sử dụng ngôn ngữ truy vấn có cấu trúc (SQL – Structured Query Language) để thực hiện các thao tác với dữ liệu như lưu trữ, chỉnh sửa, tìm kiếm và quản lý.  
MySQL được ứng dụng rộng rãi trong các hệ thống web và phần mềm nhờ tính ổn định, tốc độ xử lý nhanh và khả năng hỗ trợ đa nền tảng. Đây cũng là thành phần quan trọng trong mô hình **LAMP** (Linux – Apache – MySQL – PHP/Python/Perl), vốn được sử dụng phổ biến trong phát triển web.

## Ưu điểm của MySQL

**Miễn phí và mã nguồn mở**: Người dùng có thể tải về, sử dụng và tùy chỉnh theo nhu cầu.

**Tốc độ xử lý nhanh**: MySQL tối ưu cho các ứng dụng web và có khả năng truy vấn dữ liệu với hiệu năng cao.

**Đa nền tảng**: Hỗ trợ nhiều hệ điều hành như Windows, Linux, macOS,…

**Bảo mật tốt**: Cung cấp các cơ chế phân quyền người dùng, mã hóa và chứng thực để bảo vệ dữ liệu.

**Hỗ trợ mạnh mẽ**: Có cộng đồng phát triển lớn, nhiều tài liệu hướng dẫn và dễ dàng tích hợp với các ngôn ngữ lập trình như PHP, Java, Python,…

**Khả năng mở rộng**: Dữ liệu có thể quản lý từ vài MB đến hàng trăm GB, phù hợp cho cả dự án nhỏ và lớn.

## Nhược điểm của MySQL

 **Hạn chế về tính năng nâng cao**: So với các hệ quản trị cơ sở dữ liệu khác như Oracle hay SQL Server, MySQL thiếu một số tính năng như phân tích dữ liệu nâng cao, quản lý giao dịch phức tạp.

 **Khả năng xử lý dữ liệu cực lớn**: Dù có thể mở rộng, nhưng với hệ thống có quy mô dữ liệu hàng chục TB, MySQL thường không tối ưu bằng các giải pháp thương mại.

 **Hỗ trợ giao dịch hạn chế ở một số phiên bản cũ**: Các bản MySQL trước đây chưa hỗ trợ tốt ACID transactions; điều này đã cải thiện nhưng vẫn có giới hạn khi so với các hệ quản trị cao cấp.

 **Quản lý phân tán chưa mạnh**: Khi triển khai trong môi trường phân tán hoặc yêu cầu đồng bộ dữ liệu phức tạp, MySQL không mạnh bằng các hệ NoSQL hoặc DBMS thương mại khác.

## Tổng quan về ngôn ngữ lập trình phía máy chủ

## Giới thiệu về nền tảng công nghệ nguồn mở

- Nền tảng công nghệ nguồn mở (Open Source Platform) là tập hợp các công cụ, framework và thư viện được cộng đồng phát triển, chia sẻ miễn phí và cho phép tùy chỉnh theo nhu cầu của từng dự án. Ưu điểm nổi bật của mã nguồn mở là chi phí thấp, cộng đồng hỗ trợ đông đảo và khả năng mở rộng linh hoạt.

- Trong phát triển ứng dụng web, nhiều công nghệ nguồn mở phổ biến được sử dụng như: **Linux** (hệ điều hành), **Apache/Nginx** (máy chủ web), **MySQL/PostgreSQL** (cơ sở dữ liệu), và các ngôn ngữ lập trình như **PHP, Python, Node.js**.  
Việc lựa chọn nền tảng nguồn mở giúp các nhóm phát triển có thể tiết kiệm chi phí bản quyền, đồng thời dễ dàng tiếp cận các tài liệu và cộng đồng hỗ trợ khi triển khai hệ thống.

## Giới thiệu về ngôn ngữ lập trình PHP

PHP (Hypertext Preprocessor) là ngôn ngữ lập trình phía máy chủ (server-side scripting language) được sử dụng phổ biến trong phát triển web. PHP được tạo ra bởi **Rasmus Lerdorf** vào năm 1995 và hiện nay được duy trì bởi cộng đồng nguồn mở cùng tổ chức **The PHP Group**.

**Đặc điểm chính của PHP:**

**Thân thiện với web**: PHP được thiết kế đặc biệt để nhúng vào trong HTML và làm việc chặt chẽ với HTTP, form, cookie, session.

**Hỗ trợ nhiều hệ quản trị cơ sở dữ liệu**: PHP tích hợp tốt với MySQL, PostgreSQL, Oracle, SQL Server,…

**Tốc độ xử lý nhanh, cú pháp dễ học**: Phù hợp với cả người mới bắt đầu và dự án thực tế.

**Mã nguồn mở, cộng đồng lớn**: Có hàng ngàn thư viện, package hỗ trợ, dễ tìm kiếm tài liệu học tập.

**Khả năng mở rộng**: PHP hỗ trợ lập trình hướng đối tượng (OOP), tích hợp API RESTful, và kết nối với nhiều dịch vụ bên ngoài.

**Ứng dụng:** Nhiều hệ thống quản trị nội dung (CMS) nổi tiếng như **WordPress, Joomla, Drupal** đều được xây dựng bằng PHP. Ngoài ra, các website thương mại điện tử, diễn đàn, và mạng xã hội cũng thường dùng PHP nhờ tính đơn giản, linh hoạt và chi phí triển khai thấp.

## Tổng quan về Framework PHP Laravel

Laravel là một **framework PHP hiện đại** được phát triển bởi **Taylor Otwell** vào năm 2011, nhằm giúp việc phát triển ứng dụng web trở nên nhanh chóng, an toàn và dễ bảo trì. Laravel hoạt động dựa trên mô hình **MVC (Model – View – Controller)**, giúp tách biệt rõ ràng phần giao diện, xử lý logic và dữ liệu.

**Ưu điểm chính của Laravel:**

* **Cú pháp rõ ràng, dễ đọc**: Laravel cung cấp các hàm, class, và công cụ có tính gợi nhớ cao, giảm bớt sự phức tạp khi lập trình.
* **ORM Eloquent mạnh mẽ**: Cho phép làm việc với cơ sở dữ liệu bằng cú pháp hướng đối tượng thay vì SQL thuần.
* **Hỗ trợ Migration & Seeder**: Quản lý và khởi tạo cơ sở dữ liệu dễ dàng, thuận tiện cho phát triển nhóm.
* **Routing linh hoạt**: Laravel hỗ trợ định tuyến đơn giản, dễ dàng tạo API hoặc website có cấu trúc rõ ràng.
* **Tích hợp sẵn các chức năng bảo mật**: Bảo vệ CSRF, XSS, SQL Injection, mã hóa mật khẩu bằng Bcrypt/Argon2.
* **Cộng đồng lớn, tài liệu phong phú**: Laravel có hệ sinh thái mạnh với nhiều package hỗ trợ (Laravel Breeze, Jetstream, Livewire, Sanctum, Passport,…).

**Ứng dụng trong dự án:** Laravel được chọn để xây dựng hệ thống đặt món trực tuyến **FatFood** nhờ:

* Khả năng phát triển nhanh các module CRUD.
* Hệ thống bảo mật tốt cho việc đăng ký, đăng nhập và quản lý đơn hàng.
* Dễ dàng tích hợp với MySQL – hệ quản trị CSDL được sử dụng trong dự án.
* Khả năng mở rộng trong tương lai nếu hệ thống cần nâng cấp (thêm API cho mobile app, tích hợp thanh toán trực tuyến, v.v.).

# **CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI ĐẶT MÓN TRỰC TUYẾN**

## Đặt vấn đề

Ngày nay, cùng với sự phát triển mạnh mẽ của công nghệ thông tin và thương mại điện tử, nhu cầu đặt món ăn trực tuyến ngày càng phổ biến. Người dùng không cần trực tiếp đến cửa hàng mà vẫn có thể lựa chọn món ăn, thanh toán và nhận hàng tận nơi. Tuy nhiên, nhiều cửa hàng ăn uống vẫn quản lý thủ công hoặc sử dụng các hệ thống rời rạc, gây mất thời gian và dễ sai sót. Vì vậy, việc xây dựng một hệ thống đặt món trực tuyến tiện lợi, hiện đại và dễ sử dụng là cần thiết để hỗ trợ khách hàng cũng như nhà quản trị trong việc quản lý kinh doanh.

## Hệ thống hiện tại

Hiện nay, phần lớn các quán ăn, nhà hàng nhỏ lẻ chủ yếu nhận đơn hàng qua điện thoại hoặc tin nhắn trực tiếp. Việc quản lý đơn hàng, khách hàng và sản phẩm chưa được tự động hóa hoàn toàn, dẫn đến:

* Khó kiểm soát số lượng đơn hàng, tình trạng giao hàng.
* Dữ liệu khách hàng, lịch sử mua hàng không được lưu trữ tập trung.
* Quản trị viên phải quản lý sản phẩm, giá cả và khuyến mãi thủ công, gây tốn thời gian.  
  Điều này làm giảm hiệu quả kinh doanh và trải nghiệm của khách hàng.

## Hệ thống đề xuất

Hệ thống đặt món trực tuyến **FatFood** được xây dựng nhằm khắc phục những hạn chế trên, với các tính năng chính:

* Cho phép khách hàng xem thông tin chi tiết món ăn, tìm kiếm và đặt hàng trực tuyến.
* Hỗ trợ thanh toán nhanh chóng và theo dõi trạng thái đơn hàng.
* Lưu trữ dữ liệu khách hàng, lịch sử giao dịch, tích điểm thưởng.
* Quản trị viên dễ dàng quản lý sản phẩm, danh mục, tài khoản khách hàng, đơn hàng và thống kê doanh thu.



* Hệ thống hoạt động trực tuyến, thân thiện với nhiều thiết bị (PC, mobile, tablet).

## Giới hạn của hệ thống

- Hệ thống được xây dựng chủ yếu phục vụ cho cửa hàng/chuỗi cửa hàng vừa và nhỏ.

- Chỉ hỗ trợ các tính năng đặt món cơ bản (xem sản phẩm, giỏ hàng, thanh toán, quản lý đơn hàng) trong giai đoạn đầu.

- Chưa tích hợp đầy đủ các phương thức thanh toán trực tuyến đa dạng (như ví điện tử, QR Pay…), chủ yếu dùng thanh toán khi nhận hàng (COD) và một số cổng thanh toán phổ biến.

- Hệ thống không bao gồm chức năng vận chuyển chuyên nghiệp, chỉ hỗ trợ quản lý đơn hàng và thông tin giao hàng.

## Yêu cầu về phần cứng và phần mềm

## Yêu cầu tối thiểu

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | - RAM: ≥ 2GB  - HDD trống: ≥ 20GB  - CPU: Intel Core i3 hoặc tương đương  - Kết nối Internet ổn định | - RAM: ≥ 2GB  - CPU: Intel Core i3 hoặc tương đương  - Kết nối Internet. |
| Software | - Hệ điều hành: Windows Server 2012 hoặc Linux Ubuntu Server  - CSDL: MySQL ≥ 5.7  - PHP ≥ 7.4  - Framework Laravel ≥ 8.x | - Hệ điều hành: Windows 7/8/10 hoặc tương đương  - Trình duyệt: Chrome ≥ 80, Firefox ≥ 70, Edge ≥ 80 |

## Yêu cầu đề nghị

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | |  | | --- | |  |  |  | | --- | | - RAM: ≥ 4GB  - HDD trống: ≥ 50GB  - CPU: Intel Core i5 hoặc tương đương  - Kết nối Internet tốc độ cao | | - RAM: ≥ 4GB  - CPU: Intel Core i5 hoặc tương đương  - Kết nối Internet |
| Software | |  | | --- | |  |  |  | | --- | | - Hệ điều hành: Windows Server 2016/2019 hoặc Linux mới nhất  - MySQL ≥ 8.0  - PHP ≥ 8.0  - Laravel ≥ 9.x | | - Hệ điều hành: Windows 10/11 hoặc macOS  - Trình duyệt: Chrome/Firefox/Edge bản mới nhất  - Hỗ trợ thiết bị di động Android/iOS |

# **CHƯƠNG 3:**

# **PHÂN TÍCH YÊU CẦU KHÁCH HÀNG VÀ THIẾT KẾ HỆ THỐNG**

## Người dùng của hệ thống

***Đối tượng người dùng của hệ thống bao gồm***

* Guest (Khách vãng lai)
* Member (Thành viên)
* Admin (Quản trị hệ thống).

## Chức năng của hệ thống

* + 1. ***Guest*** *(Chức năng dành cho khách vãng lai)*
* **Xem sản phẩm**

|  |  |
| --- | --- |
| **Mô tả** | Khách hàng có thể xem thông tin chi tiết từng loại giày như tên giày, kích cỡ, màu sắc, giá cả… |
| **Nhập** | Lựa chọn loại giày cần xem. |
| **Xử lý** | Tìm kiếm trong CSDL. |
| **Xuất** | Hiển thị thông tin mà khách hàng muốn xem. |

* ………….
* …………….
* ………………..
* ……………..
* **Đăng ký thành viên**

|  |  |
| --- | --- |
| **Mô tả** | Khách vãng lai có thể đăng ký làm thành viên. |
| **Nhập** | Đưa vào các thông tin đã nhập trong form để kiểm tra. |
| **Xử lý** | * Kiểm tra username, email đã được đăng ký hay chưa. * Mật khẩu phải gồm ít nhất 6 ký tự và không có ký tự trắng. * Email nhập vào phải đúng định dạng. * ... * Yêu cầu nhập các thông tin bắt buộc. * Thêm mới các thông tin khách hàng, mã hóa mật khẩu vào CSDL nếu đúng. Nếu không yêu cầu nhập lại. |
| **Xuất** | Thông báo khách hàng đăng ký thành công sau khi đã điền đầy đủ thông tin hợp lệ. Quay về trang chủ. |

* + 1. ***Member functions*** *(Chức năng dành cho thành viên)*

Thành viên chính thức của website sẽ có các chức năng giống như khách vãng lai như:

* + Xem sản phẩm.
  + Tìm kiếm sản phẩm.
  + Lựa chọn sản phẩm vào giỏ hàng.

***Ngoài ra thành viên chính thức còn có các chức năng sau***:

* **Đăng nhập**

|  |  |
| --- | --- |
| **Mô tả** | Đăng nhập vào website bằng cách nhập username và password đã đăng ký. |
| **Nhập** | Nhập username và mật khẩu đã đăng ký. |
| **Xử lý** | Truy cập kiểm tra xem tài khoản có tồn tại hay không. |
| **Xuất** | Hiển thị thông báo đăng nhập thành công khi khách hàng nhập đúng thông tin đã đăng ký và quay về trang chủ. Thông báo sai thông tin đăng nhập nếu tài khoản chưa đăng ký hoặc bị khóa, yêu cầu nhập lại thông tin đăng nhập! |

* **Đăng xuất**

|  |  |
| --- | --- |
| **Mô tả** | Đăng xuất khỏi website. |
| **Nhập** | Nhấn nút thoát để đăng xuất tài khoản. |
| **Xử lý** | * Hủy toàn bộ session của người dùng hiện tại * Đưa trở về chức năng của khách vãng lai. |
| **Xuất** | Hiển thị giao diện như khách vãng lai sau khi đã đăng xuất khỏi hệ thống. |

* **Đặt hàng**
* **………………..**
* **………………**
* **………………….**
  + 1. ***Admin functions*** *(Chức năng dành cho quản trị viên)*

***Quản trị viên có các chức năng chính như sau:***

* **Đăng nhập quản trị**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên đăng nhập vào hệ thống bằng cách nhập username và password vào form đăng nhập. |
| **Nhập** | Nhập tài khoản và mật khẩu của quản trị viên. |
| **Xử lý** | Kiểm tra thông tin nhập vào có hợp lệ hay không. |
| **Xuất** | * Chuyển đến trang quản trị nếu đăng nhập thành công * Quay lại trang đăng nhập nếu nhập sai. |

* **Đăng xuất quản trị**

|  |  |
| --- | --- |
| **Mô tả** | Thoát khỏi tài khoản quản trị viên. |
| **Nhập** | Click vào nút thoát từ giao diện quản trị. |
| **Xử lý** | Hủy session quản trị viên đã đăng nhập. |
| **Xuất** | Quay lại trang chủ. |

* **Quản lý tài khoản khách hàng**

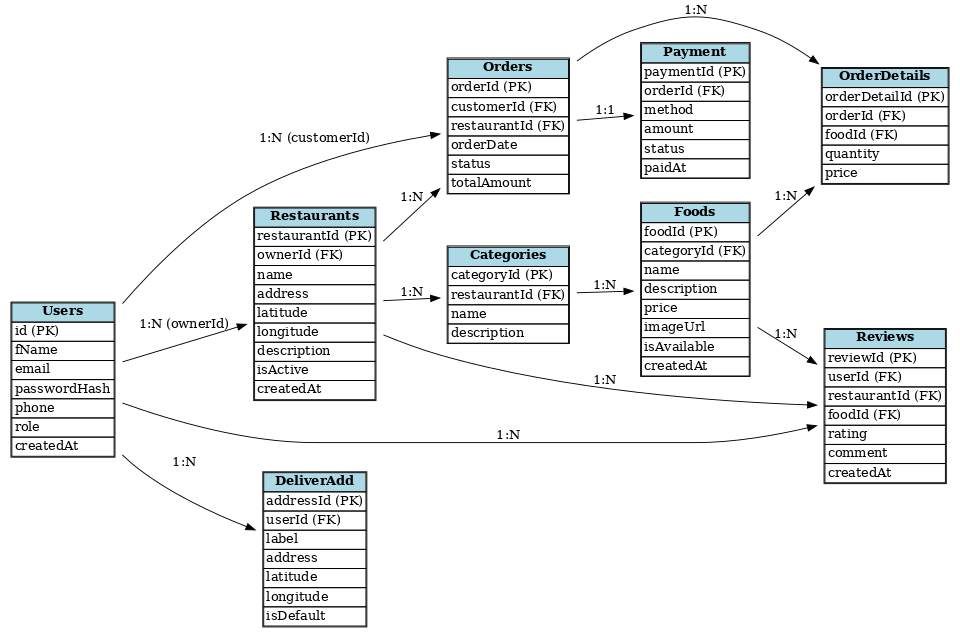
|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên có quyền xem thông tin hoặc khóa/kích hoạt, xóa các tài khoản của thành viên. |
| **Nhập** | Chọn lệnh tương ứng của tài khoản cần thao tác. |
| **Xử lý** | * Với việc xem thông tin tài khoản: * Hiển thị thông tin của khách hàng. * Đối với việc xóa tài khoản: * Thay đổi trạng thái quyền đăng nhập website. * Khóa/kích hoạt tài khoản: * Admin có quyền khóa/kích hoạt tài khoản của khách hàng. |
| **Xuất** | Hiện thông tin đã thay đổi. |

* **Quản lý sản phẩm**
* **………….**
* **………….**
* **……………**

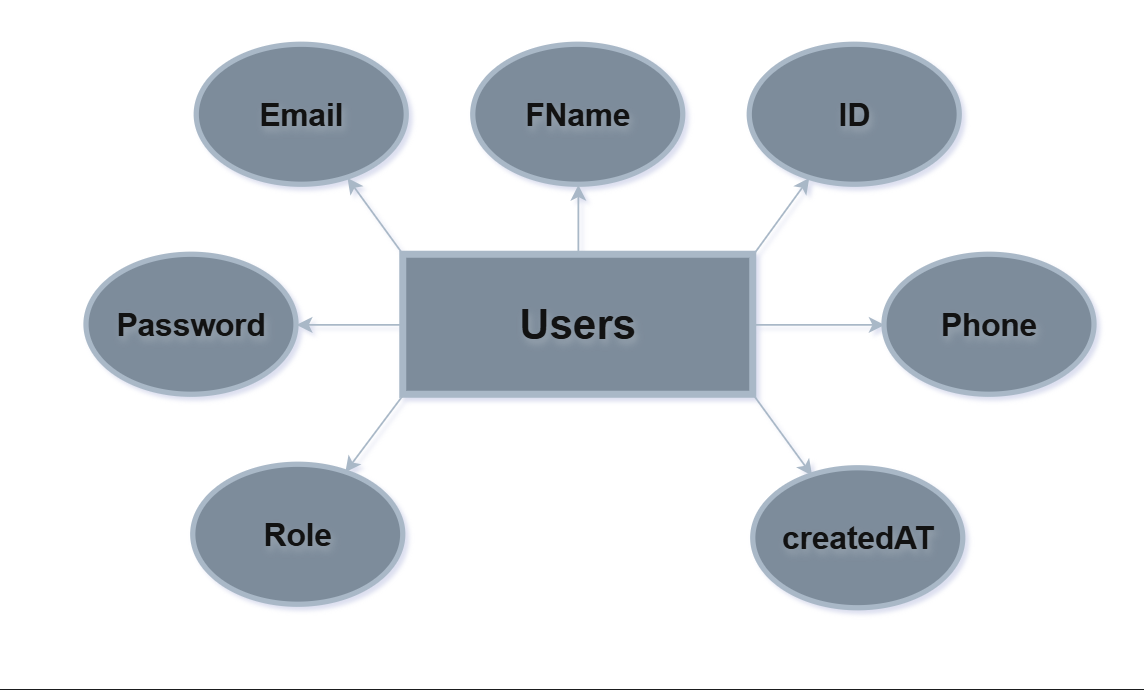
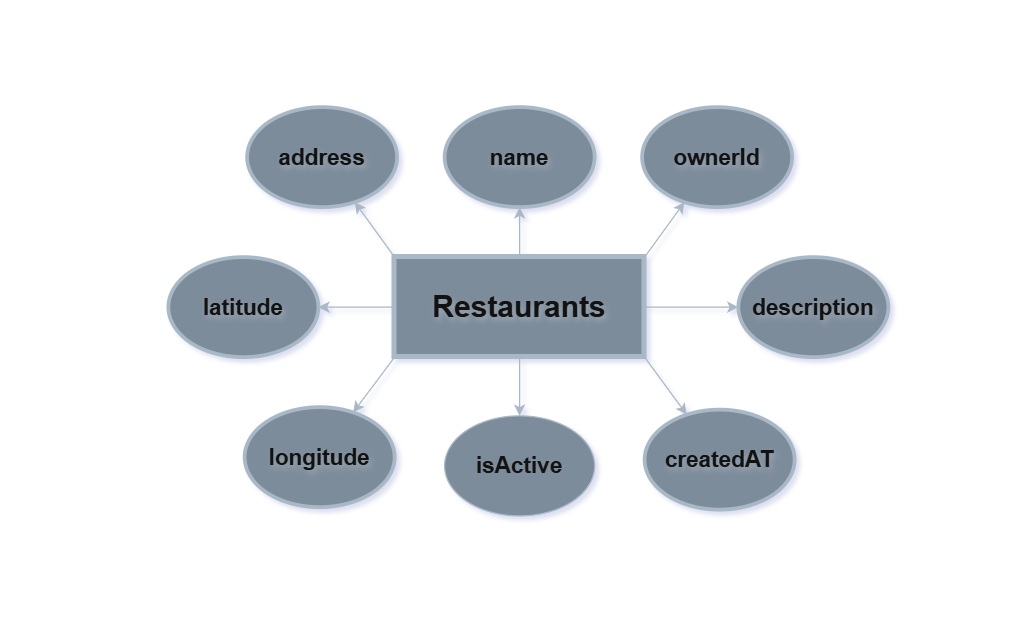
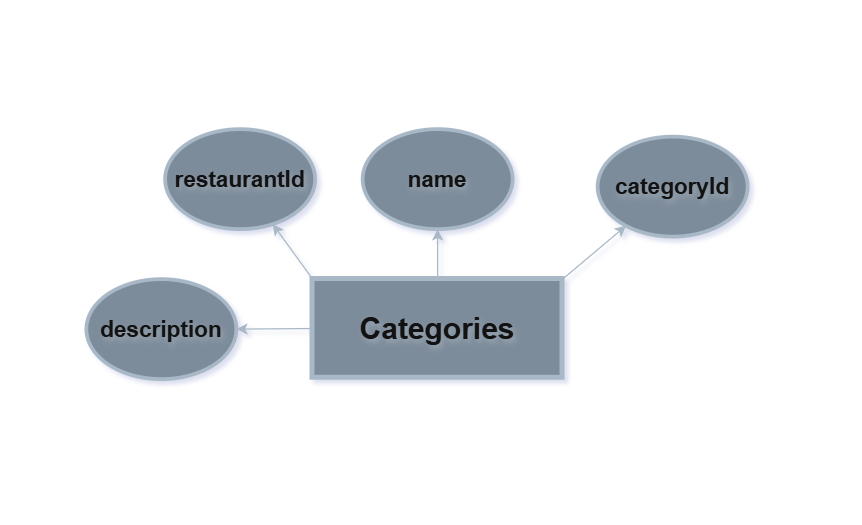
## Systeme Designs (Thiết kế hệ thống)

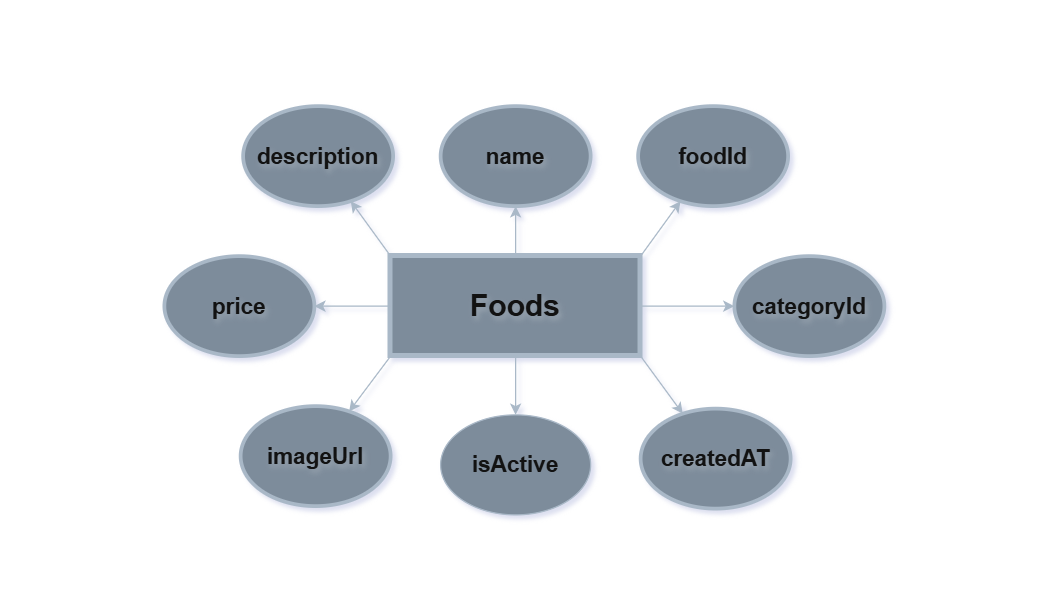
## Entity Relationship Diagram (Mô hình quan hệ thực thể)

Thực thể quản trị (**QUAN\_TRI**):   
Lưu trữ thông tin người quản trị viên hệ thống.

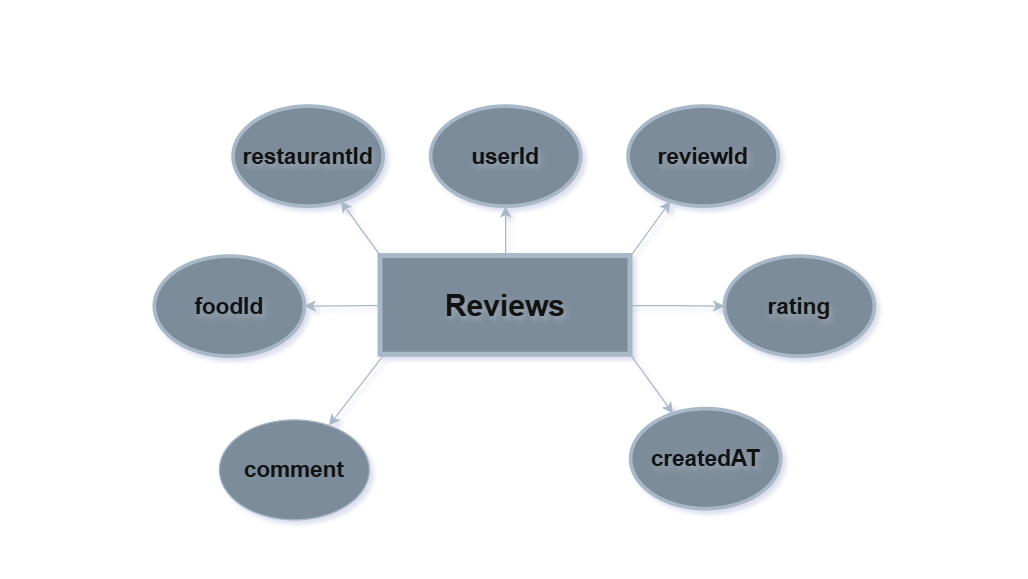


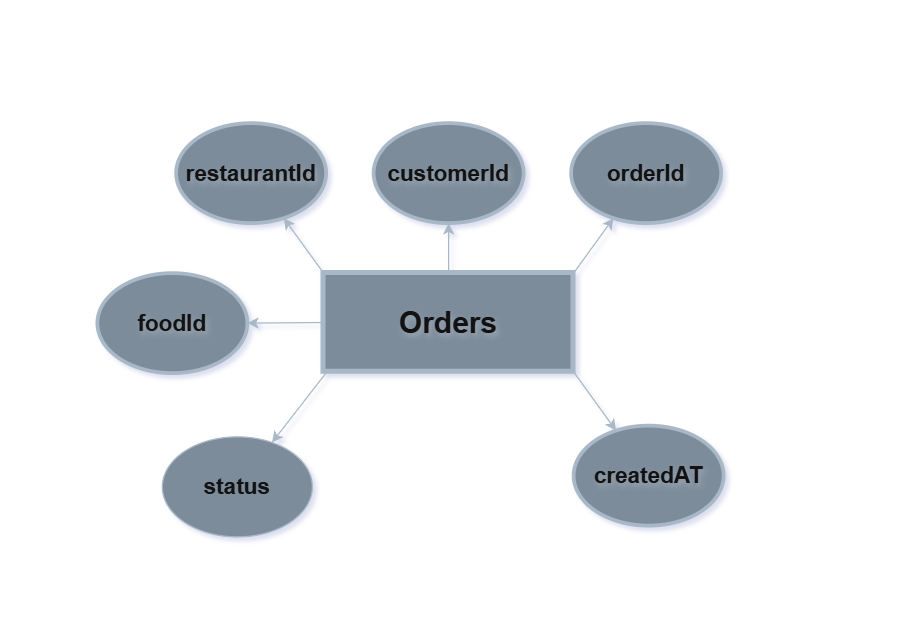
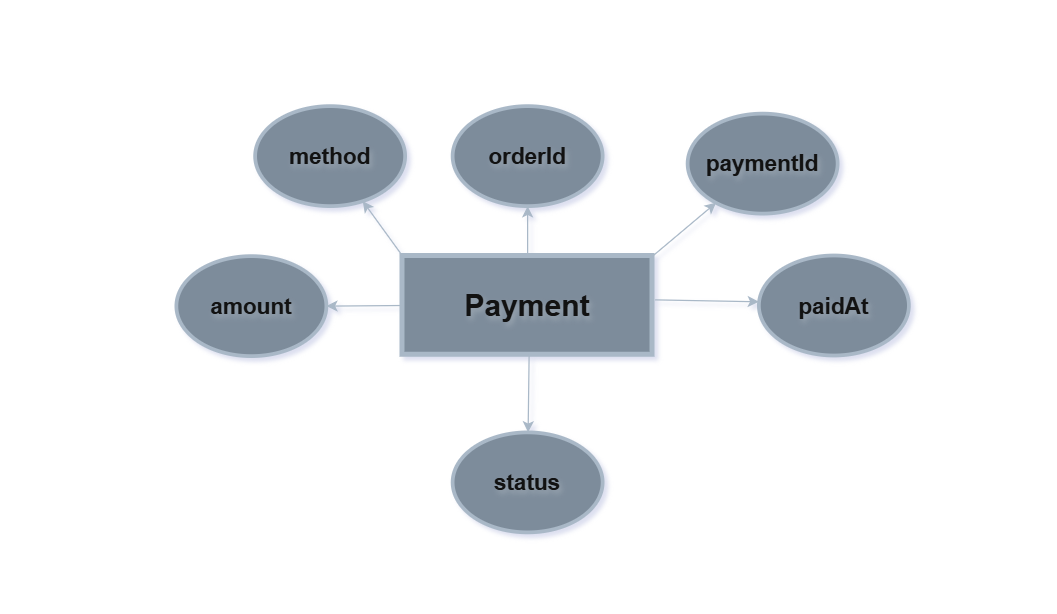
***Mô tả chi tiết các thuộc tính của các thực thể***:

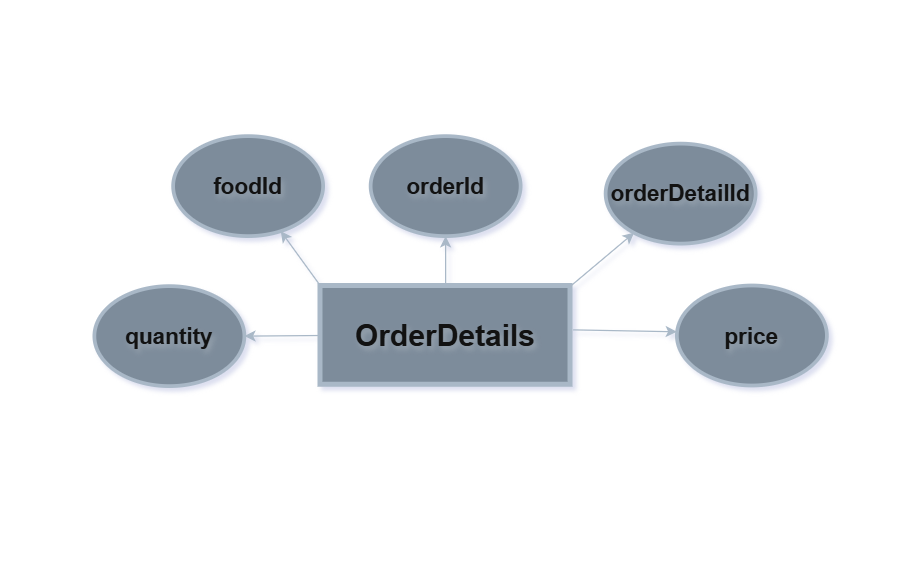
1. Thực thể người dùng (**Users**) : Lưu trữ thông tin người dùng hệ thống bao gồm admin , khách hàng , chủ nhà hàng.
2. Thực thể Nhà hàng (Restaurants): Lưu trữ thông tin nhà hàng khi đăng ký, sử dụng tại hệ thống. 
3. Thực thể Danh mục đồ ăn(Categories): Lưu trữ các danh mục thức ăn của nhà hàng như đồ ăn nóng , đồ ăn nhanh , đồ uống ,... 
4. Thực thể đồ ăn(Foods): Lưu trữ các đồ ăn của nhà hàng .



1. Thực thể Đánh giá(Reviews): Lưu trữ các đánh giá đồ ăn của nhà hàng .



1. Thực thể Đơn hàng(Orders):Lưu lại đơn hàng mà khách đã đặt
2. Thực thể Thanh toán(Payments):Lưu lại đơn hàng mà khách đã đặt
3. Thực thể Địa chỉ giao hàng():Lưu lại địa chỉ giao hàng , giúp thuận tiện vận chuyển. 
4. Thực thể chi tiết đơn hàng():Lưu lại chi tiết đơn hàng.



## Database Design (Thiết kế cơ sở dữ liệu)

Thiết kế có sở dữ liệu trên hệ quản trị dữ liệu MS SQL (SQL Server)

# 1. Bảng Users

|  |  |  |  |
| --- | --- | --- | --- |
| Field | DataType | Constraint | Content |
| ID | INT | PK, Auto Increment | Mã người dùng |
| FName | NVARCHAR(50) | NOT NULL, UNIQUE | Tên đăng nhập |
| PasswordHash | NVARCHAR(100) | NOT NULL | Mật khẩu (mã hoá) |
| Email | NVARCHAR(100) | UNIQUE | Email người dùng |
| Phone | NVARCHAR(20) | NULL | Số điện thoại |
| Role | Int | NOT NULL | Quyền (Admin/User) |
| createdAT | Datetime | NULL | Ngày tạo |

# 2. Bảng Restaurants

|  |  |  |  |
| --- | --- | --- | --- |
| Field | DataType | Constraint | Content |
| OwnerID | INT | PK, Auto Increment | Mã nhà hàng |
| Name | NVARCHAR(200) | NOT NULL | Tên nhà hàng |
| Address | NVARCHAR(255) | NOT NULL | Địa chỉ |
| Latitude | Decimal | NULL |  |
| Description | NVARCHAR(500) | NULL | Mô tả |
| Longitude | Decimal | NULL |  |
| createdAT | Datetime | NOT NULL | Ngày tạo |
| isActive | BIT | NOT NULL | Trạng thái |

# 3. Bảng Categories

|  |  |  |  |
| --- | --- | --- | --- |
| Field | DataType | Constraint | Content |
| CategoryID | INT | PK, Auto Increment | Mã danh mục |
| Name | NVARCHAR(100) | NOT NULL, UNIQUE | Tên danh mục món ăn |
| RestaurantID | INT | PK Restaurants | Mã nhà hàng |
| Description | NVARCHAR(500) | NULL | Mô tả |

# 4. Bảng Foods

|  |  |  |  |
| --- | --- | --- | --- |
| Field | DataType | Constraint | Content |
| FoodID | INT | PK, Auto Increment | Mã món ăn |
| Name | NVARCHAR(150) | NOT NULL | Tên món ăn |
| Price | DECIMAL(10,2) | NOT NULL | Giá món ăn |
| CategoryID | INT | FK → Categories(CateID) | Danh mục |
| isAvailble | BIT | NOT NULL | Trạng thái |
| imgURL | NVARCHAR(500) | NULL | ảnh |
| Description | NVARCHAR(500) | NULL | Mô tả |

# 5. Bảng Orders

|  |  |  |  |
| --- | --- | --- | --- |
| Field | DataType | Constraint | Content |
| OrderID | INT | PK, Auto Increment | Mã đặt hàng |
| CustomerID | INT | PK, User | Người đặt |
| RestaurantID | INT | PK Restaurants | Người bán |
| OrderDate | DATETIME | DEFAULT GETDATE() | Ngày đặt |
| Status | NVARCHAR(50) | NOT NULL | Trạng thái |
| totalAmount | Decimal | NOT NULL | Tổng giá |

# 6. Bảng OrderDetails

|  |  |  |  |
| --- | --- | --- | --- |
| Field | DataType | Constraint | Content |
| DetailID | INT | PK, Auto Increment | Mã chi tiết đơn hàng |
| OrderID | INT | FK → Orders(OrderID) | Đơn hàng |
| FoodID | INT | FK → Foods(FoodID) | Món ăn |
| Quantity | INT | NOT NULL | Số lượng |
| Price | DECIMAL(10,2) | NOT NULL | Giá tại thời điểm đặt |

# 7. Bảng Payment

|  |  |  |  |
| --- | --- | --- | --- |
| Field | DataType | Constraint | Content |
| PaymentID | INT | PK, Auto Increment | Mã thanh toán |
| OrderID | INT | FK → Orders(OrderID) | Đơn hàng |
| Method | NVARCHAR(50) | NOT NULL | Hình thức |
| Amount | DECIMAL(10,2) | NOT NULL | Số tiền thanh toán |
| Status | NVARCHAR(50) | NOT NULL | Trạng thái |
| PayDate | DATETIME | DEFAULT GETDATE() | Ngày thanh toán |

# 8. Bảng Reviews

|  |  |  |  |
| --- | --- | --- | --- |
| Field | DataType | Constraint | Content |
| ReviewID | INT | PK, Auto Increment | Mã đánh giá |
| UserID | INT | FK → Users(UserID) | Người đánh giá |
| RestaurantID | INT | PK Restaurants | Nhà hàng |
| FoodID | INT | FK → Foods(FoodID) | Món ăn |
| Rating | INT | CHECK(Rating BETWEEN 1 AND 5) | Điểm đánh giá |
| Comment | NVARCHAR(500) | NULL | Nội dung nhận xét |
| createdAt | DATETIME | NOT NULL | Ngày tạo |

# 9. Bảng DeliverAdd

|  |  |  |  |
| --- | --- | --- | --- |
| Field | DataType | Constraint | Content |
| AddressID | INT | PK, Auto Increment | Mã địa chỉ |
| UserID | INT | FK → Users(UserID) | Người dùng |
| Address | NVARCHAR(255) | NOT NULL | Địa chỉ giao hàng |
| Latitude | Decimal | NOT NULL |  |
| Longitude | Decimal | NOT NULL |  |
| isDefault | BIT | NOT NULL |  |

## Table RelationShip Diagram (Mô hình quan hệ bảng)

Mô hình quan hệ bảng (Table Relationship Diagram – TRD) thể hiện mối quan hệ giữa các bảng trong cơ sở dữ liệu. Hệ thống đặt món trực tuyến **FatFood** được thiết kế theo mô hình quan hệ, trong đó mỗi bảng đảm nhận một vai trò riêng và có liên kết thông qua **khóa chính (Primary Key – PK)** và **khóa ngoại (Foreign Key – FK)**.

**3.5.1. Bảng Users**

**- UserID (PK)**: Mã người dùng.

- Mối quan hệ:

**1–N** với **Orders** (một người dùng có thể đặt nhiều đơn hàng).

**1–N** với **Reviews** (một người dùng có thể viết nhiều đánh giá).

**1–N** với **DeliverAdd** (một người dùng có thể có nhiều địa chỉ giao hàng).

**1–N** với **Restaurants** (một người dùng có thể là chủ sở hữu của nhiều nhà hàng).

**3.5.2. Bảng Restaurants**

**- ResID (PK)**: Mã nhà hàng.

**- OwnerID (FK)** → Users(UserID).

- Mối quan hệ:

**1–N** với **Foods** (một nhà hàng có nhiều món ăn).

**1–N** với **Categories** (một nhà hàng có nhiều danh mục món).

**3.5.3. Bảng Categories**

**- CateID (PK)**: Mã danh mục.

- Mối quan hệ:

**1–N** với **Foods** (một danh mục có nhiều món ăn).

**3.5.4. Bảng Foods**

**- FoodID (PK)**: Mã món ăn.

**- CateID (FK)** → Categories(CateID).

**- ResID (FK)** → Restaurants(ResID).

- Mối quan hệ:

**1–N** với **OrderDetails** (một món ăn có thể xuất hiện trong nhiều chi tiết đơn hàng).

**1–N** với **Reviews** (một món ăn có nhiều đánh giá).

**3.5.5. Bảng Orders**

**- OrderID (PK)**: Mã đơn hàng.

**- UserID (FK)** → Users(UserID).

- Mối quan hệ:

**1–N** với **OrderDetails** (một đơn hàng có nhiều chi tiết đơn hàng).

**1–1** với **Payments** (mỗi đơn hàng có một giao dịch thanh toán).

**3.5.6. Bảng OrderDetails**

**- DetailID (PK)**: Mã chi tiết đơn hàng.

**- OrderID (FK)** → Orders(OrderID).

**- FoodID (FK)** → Foods(FoodID).

- Mối quan hệ:

Kết nối giữa bảng **Orders** và **Foods** theo quan hệ **N–N** (một đơn hàng có nhiều món ăn, một món ăn có thể nằm trong nhiều đơn hàng).

**3.5.7. Bảng Payments**

**- PayID (PK)**: Mã thanh toán.

**- OrderID (FK)** → Orders(OrderID).

- Mối quan hệ:

**1–1** với **Orders** (một đơn hàng có một giao dịch thanh toán duy nhất).

**3.5.8. Bảng Reviews**

**- ReviewID (PK)**: Mã đánh giá.

**- UserID (FK)** → Users(UserID).

**- FoodID (FK)** → Foods(FoodID).

- Mối quan hệ:

**N–1** với **Users** (mỗi đánh giá thuộc về một người dùng).

**- N–1** với **Foods** (mỗi đánh giá gắn liền với một món ăn).

**3.5.9. Bảng DeliverAdd (Địa chỉ giao hàng)**

**- AddressID (PK)**: Mã địa chỉ.

**- UserID (FK)** → Users(UserID).

- Mối quan hệ:

**1–N** với **Users** (một người dùng có thể có nhiều địa chỉ giao hàng khác nhau).

* + 1. **Tóm tắt mối quan hệ chính:**

**- Users – Orders**: 1–N.

**- Users – DeliverAdd**: 1–N.

**- Users – Reviews**: 1–N.

**- Restaurants – Foods**: 1–N.

**- Categories – Foods**: 1–N.

**-Foods – Reviews**: 1–N.

**- Orders – OrderDetails – Foods**: quan hệ N–N thông qua OrderDetails.

**- Orders – Payments**: 1–1.

## SiteMap (Sơ đồ cấu trúc site)

Sơ đồ cấu trúc site mô tả cách tổ chức và điều hướng các trang trong hệ thống đặt món trực tuyến FatFood. Hệ thống được thiết kế phục vụ ba nhóm đối tượng chính: Khách vãng lai (Guest), Thành viên đã đăng ký (Member) và Quản trị viên (Admin).  
Việc xây dựng sitemap giúp đảm bảo giao diện trực quan, dễ sử dụng, đồng thời hỗ trợ người dùng nhanh chóng tiếp cận các chức năng cần thiết.

## Site map (Cấu trúc trang): Dành cho khách vãng lai



## Sitemap (Cấu trúc trang): Dành cho thành viên

Trang chủ

Sản phẩm

Nhãn hiệu

Tìm kiếm

Giới thiệu

Liên hệ

Hướng dẫn

Đăng xuất

Thông tin cá nhân

Giỏ hàng và đơn hàng

Đặt món

Đổi mật khẩu

Sửa thông tin

Bình luận

## Sitemap (Cấu trúc trang): Dành cho quản trị viên

Trang chủ Admin

Quản lý nhà hàng

Doanh thu theo tháng/toàn hệ thống.

Quản lý đơn hàng & thanh toán

Bật/tắt hoạt động (isActive).

Quản lý Hóa đơn

Xóa/sửa món không hợp lệ (từ foods).

Quản lý hệ thống món ăn

Theo dõi toàn bộ orders.

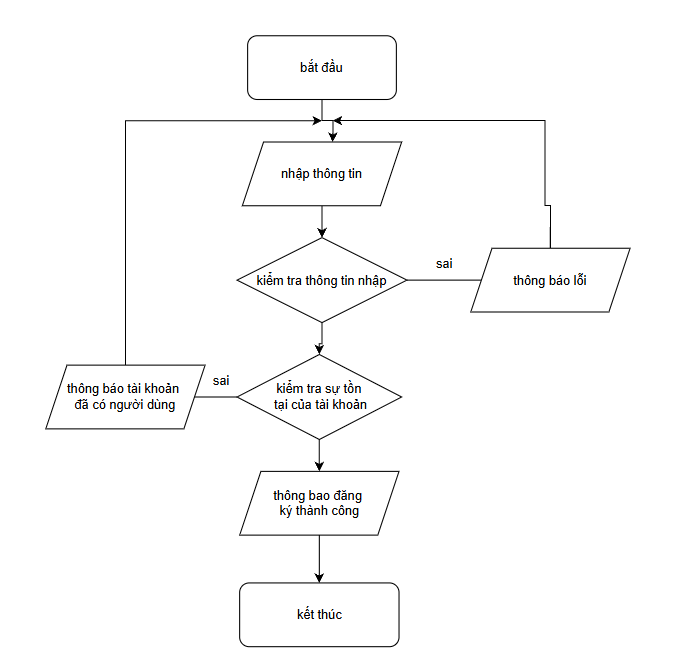
Số lượng đơn, tỉ lệ đơn thành công/hủy.

Giải quyết khiếu nại/refund.

## Algorithms (Giải thuật)

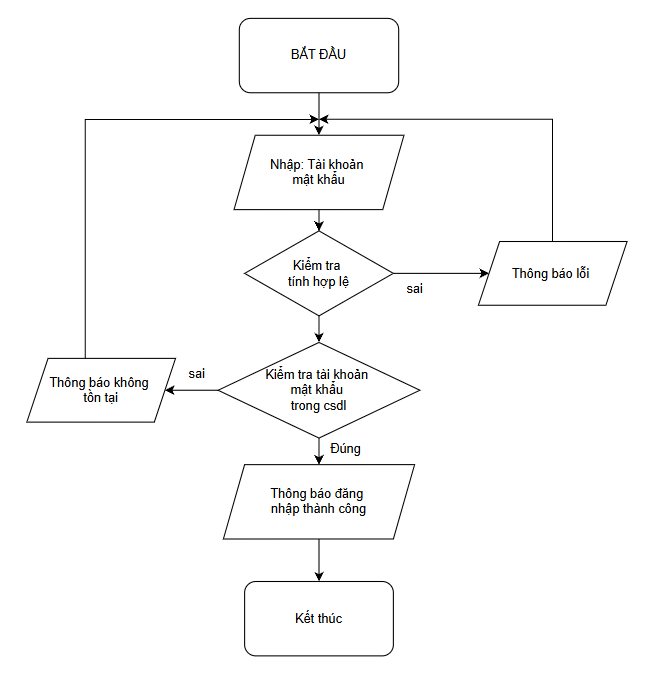
Phần này mô tả các giải thuật chính được áp dụng trong hệ thống đặt món trực tuyến FatFood, nhằm đảm bảo xử lý đúng yêu cầu của người dùng và quản trị viên. Các giải thuật được trình bày dưới dạng mô tả bước (step by step) để dễ hiểu và triển khai.

## Đăng ký



|  |  |
| --- | --- |
| * **Kiểm tra thông tin nhập**:   Kiểm tra các thông tin bắt buộc nhập theo ràng buộc khi thiết kế cơ sở dữ liệu:   * + Tài khoản, Mật khẩu, Email, Điện thoại,.. | * **Kiểm tra sự tồn tại của tài khoản:**   Tài khoản là duy nhất đối với mỗi khách hàng. Nếu đã có người dùng rồi thì không cho phép đăng ký. |

## Đăng nhập



## Đăng xuất



## Đổi mật khẩu



## Thay đổi thông tin cá nhân



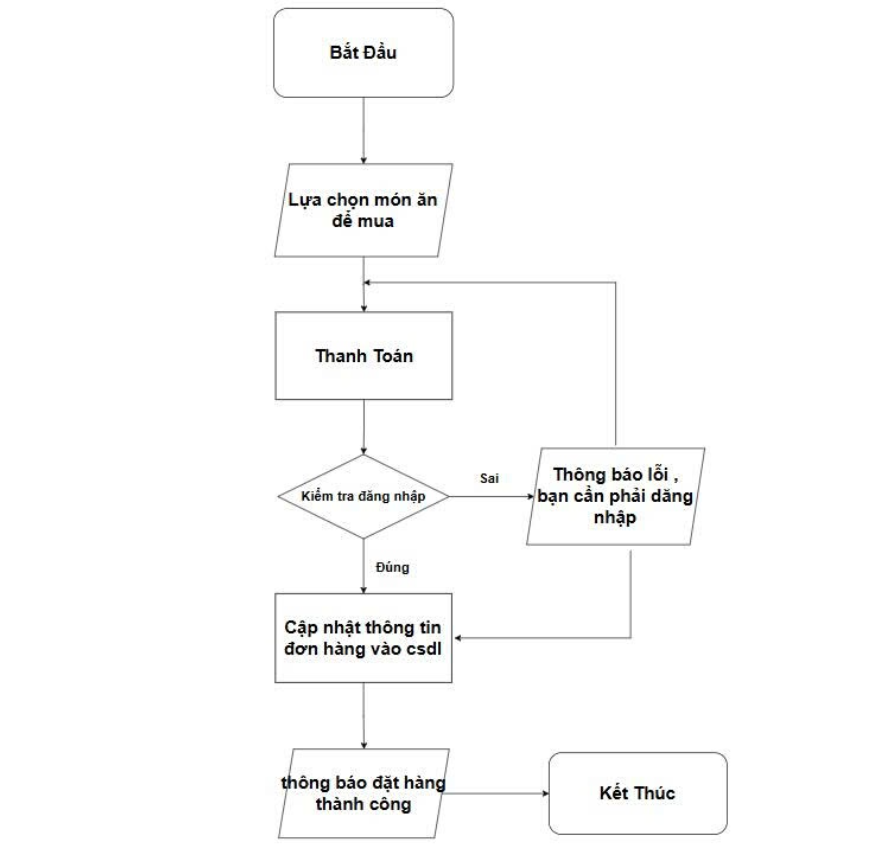
## Thêm sản phẩm



## Xóa sản phẩm



## Đặt mua



## Tìm kiếm



**CHƯƠNG 4:  
Thiết kế kiến trúc hệ thống   
& phân công module**

1. **Nội dung chính**

Xác định kiến trúc tổng thể: **Client (Browser) ↔ Web server (ASP.NET Core MVC) ↔ Database (MS SQL/MySQL)**. (Có thể dùng API layer nếu cần phục vụ mobile).

Chia thành các tầng: Presentation (Frontend), Business Logic (Controller/Services), Data Access (Repository/ORM), Database.

Chọn thư viện/stack:

- Frontend: HTML5, CSS3 (Flex/Grid), Bootstrap 5, jQuery, vanilla JS.

- Backend: ASP.NET Core MVC (C#) + Entity Framework Core.

- DB: SQL Server (theo file thiết kế) hoặc MySQL (nếu muốn). Dùng migrations.

- Tooling: Git + GitHub/GitLab, Postman, Visual Studio 2022.

-Xác định API / route chính (ví dụ):

\* /Products (GET, POST, PUT, DELETE)

\* /Orders (GET, POST)

\* /Auth/Login, /Auth/Register

\* /Users/{id}/Addresses

- Xác định cơ chế xác thực: Cookie-based auth (ASP.NET Identity) hoặc JWT (nếu có SPA/mobile).

1. **Sơ đồ luồng dữ liệu (tóm tắt)**

Khách truy cập → trang danh mục → gọi API lấy danh sách món → thêm vào giỏ hàng (session/localStorage) → đặt hàng → backend lưu Orders + OrderDetails → Payment xử lý → cập nhật trạng thái order.

1. **Phân công module**

**- Nguyễn Hữu Tuấn (Team lead / Backend lead)**

Thiết kế DB / thực hiện migrations, tạo models & repositories (Users, Restaurants, Categories, Foods, Orders, OrderDetails, Payments, Reviews, DeliverAdd).

Triển khai API cơ bản cho Products & Orders.

**- Nhữ Anh Tuấn (Frontend lead)**

Thiết kế UI/UX trang chủ, trang danh mục, trang chi tiết món, giỏ hàng, form đặt hàng (HTML/CSS/Bootstrap).

Tạo template Razor views + partials (nếu dùng MVC view engine).

**Vũ Đỗ Minh Thành (Integrations & QA)**

Tạo chức năng đăng ký/đăng nhập (Auth), trang quản lý cá nhân, tích hợp jQuery/AJAX cho tương tác (thêm giỏ, update số lượng).

Viết test thao tác chính (test thủ công + checklist QA).

1. **Deliverables**

- Tài liệu kiến trúc 1 trang (diagram + stack).

- Danh sách API endpoints và contract (request/response mẫu).

- Bảng phân công nhiệm vụ rõ ràng + timeline ngắn (ai làm gì trong Buổi 5–8).

# **DANH MỤC TÀI LIỆU THAM KHẢO**

#### **Tài liệu:**

#### **Website:**

1. W3C – *HTML5: A vocabulary and associated APIs for HTML and XHTML* (2021). Truy cập tại: <https://www.w3.org/TR/html5>;
2. Mozilla Developer Network (MDN) – *Introduction to HTML5*. Truy cập tại: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML>;
3. *Chat gpt. Truy cập tại:* [*https://chatgpt.com/?model=auto*](https://chatgpt.com/?model=auto)