**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM**

**KHOA TIN HỌC**

****

**BÁO CÁO MÔN PHÂN TÍCH THIẾT KẾ HƯỚNG ĐỐI TƯỢNG**

**Đề tài:**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ QUÁN CÀ PHÊ 5AE**

Giảng viên hướng dẫn: Thầy Nguyễn Văn Vương

Sinh viên thực hiện: ...

Đà Nẵng, 12/2022

**Mục Lục**

[**I. Đặt vấn đề 4**](#_Toc123694072)

[**II. Phân tích thiết kế 5**](#_Toc123694073)

[**1. Mô hình C4 Model 5**](#_Toc123694074)

[**2. Sử dụng mô hình C4 model để giải quyết bài toán 6**](#_Toc123694075)

[Level 1: Context Diagram for a Coffee Shop System 6](#_Toc123694076)

[Level 2: Container Diagram for a Coffee Shop System 8](#_Toc123694077)

[Level 3: Component Diagram for a Coffee Shop System 14](#_Toc123694078)

[**III. Demo 21**](#_Toc123694079)

[**1. Giao diện người dùng 21**](#_Toc123694080)

[- Trang chủ 21](#_Toc123694081)

[- Trang thực đơn 22](#_Toc123694082)

[- Trang đăng ký 22](#_Toc123694083)

[- Trang quản lý thông tin cá nhân 23](#_Toc123694084)

[- Trang chi tiết sản phẩm 23](#_Toc123694085)

[- Trang giỏ hàng 24](#_Toc123694086)

[- Trang đặt hàng thành công 24](#_Toc123694087)

[- Trang quản lý đơn hàng 25](#_Toc123694088)

[- Trang chi tiết đơn hàng 25](#_Toc123694089)

[**2. Giao diện quản trị viên 26**](#_Toc123694090)

[- Trang chủ 26](#_Toc123694091)

[- Trang quản lý bình luận 26](#_Toc123694092)

[- Trang quản lý người dùng 27](#_Toc123694093)

[- Trang quản lý sản phẩm 27](#_Toc123694094)

[- Trang quản lý đơn hàng 28](#_Toc123694095)

[- Trang chỉnh sửa đơn hàng 28](#_Toc123694096)

[- Trang cập nhật đơn hàng 29](#_Toc123694097)

**Lời mở đầu**

# I. Đặt vấn đề

Theo khảo sát thì tỷ lệ người Việt Nam dùng cà phê khá nhiều, đặc biệt là ở các thành phố lớn. Theo số liệu thu thấp từ các cuộc khảo sát, tỷ lệ người Việt Nam dùng cà phê nóng và lạnh theo giới tính thì có đến 75% đàn ông thích uống nóng, 65% chọn cà phê đá. Ở phụ nữ, thì tỉ lệ lần lượt là 25% và 35%. Từ đó có thể thấy rằng đàn ông tiêu thụ nhiều hơn phụ nữ.

Cafe có thể dùng ở nhiều thời gian trong ngày. Tùy vào sở thích mà mỗi người có thể dùng nó vào những thời điểm khác nhau. Theo khảo sát cho thấy có 90% uống cafe cùng bữa ăn sáng, 50% uống trước khi ăn sáng, 10% uống sau khi ăn sáng, 10% uống sau khi chơi thể thao, 10% uống trước khi ngủ, 90% uống ngay khi thức dậy. Với nhiều thời điểm để dùng cafe như thể nên đã góp phần làm cho tỷ lệ người Việt Nam dùng cà phê ngày càng cao.

Nhiều người lựa chọn cafe để thư giãn sau những giờ làm việc đầy mệt mỏi. Thêm vào đó là lựa chọn ngồi nhăm nhi tách cafe và nói chuyện cùng gia đình hay bạn bè.

Cafe có một vị đắng đặc trưng và cả mùi thơm đầy quyến rũ. Từ những điều trên đã khiến nó trở thành loại thức uống yêu thích của nhiều người. Có đến 40% người lựa chọn dùng thức uống này là vì sở thích. Điều này chứng tỏ là cafe đã trở thành một phần trong cuộc số của nhiều người.

Nhiều người dùng cafe vào buổi sáng hoặc buổi tối để giúp tỉnh táo hơn. Để khỏi động ngày dày khi chưa tỉnh ngủ thì nhiều người đã dùng nó để tỉnh táo hơn. Khi thức khuya học bài hay chạy deadline thì loại thức uống này thường là sự lựa chọn của nhiều người để giúp qua cơn buồn ngủ.

Khi làm việc hay học tập luôn đồi hỏi khả năng tập trung cao độ. Vì thế nhiều người đã tin rằng dùng cafe để giúp tăng khả năng tập trung của họ.

Thế nhưng không phải ai cũng luôn có thời gian để đến quán cà phê vì nhiều vì học có nhiều việc cần xử lý. Nên để đáp ứng nhu cầu của người tiêu dùng vừa có thời gian xử lý công việc vừa có thời gian thưởng thức cà phê theo khẩu vị thì nhóm em quyết định quát triển website bán quán cà phê 5ae, mục đích để giải quyết bài toán nhu cầu mua cà phê qua online của khách hàng.

# II. Phân tích thiết kế

## 1. Mô hình C4 Model

Mô hình C4 là một phương pháp mô tả kiến trúc phần mềm một cách tinh gọn nhất. Nó bao gồm một tập hợp 4 sơ đồ mô tả **cấu trúc tĩnh** của một hệ thống phần mềm.

Chữ **C** được lấy từ **4** hình vẽ chính trong mô hình này đó là: Context, Container, Component, Code (hay còn được gọi là Class).

System **C**ontext: Sơ đồ này mô tả tổng quát về hệ thống theo hướng che đi thành phần bên trong mà chỉ làm nổi bật **thành phần bên ngoài** (blackbox); bao gồm các yếu tố phụ thuộc chính (dependencies) của hệ thống, các giao thức (interfaces) để giao tiếp giữa những hệ thống với nhau và con người (người dùng / theo vai trò / theo phòng ban / v.v). Biểu đồ ngữ cảnh này là tiêu chuẩn trong kỹ thuật phần mềm.

**C**ontainer: Sơ đồ này mô tả về hệ thống theo hướng hiển thị thành **phần bên ngoài** cùng với việc phóng to những thành **phần bên trong** của hệ thống (whitebox); do đó hiển thị được các container (building block), mục đích và nhiệm vụ của từng container này, cùng với giao thức giao tiếp giữa những container đó. Sơ đồ này thường được gọi là sơ đồ khối cấp 1 (first level building block). Vậy container cụ thể là cài vẹo gì? Tuỳ theo phạm vi hệ thống các bạn đang làm (có thể là hệ thống, có thể là 1 component, cũng có thể là 1 hệ thống con), container có thể là cái máy tính, vi xử lý, cũng có thể là 1 dịch vụ (service) nào đó gởi email, lưu trữ, APIs…

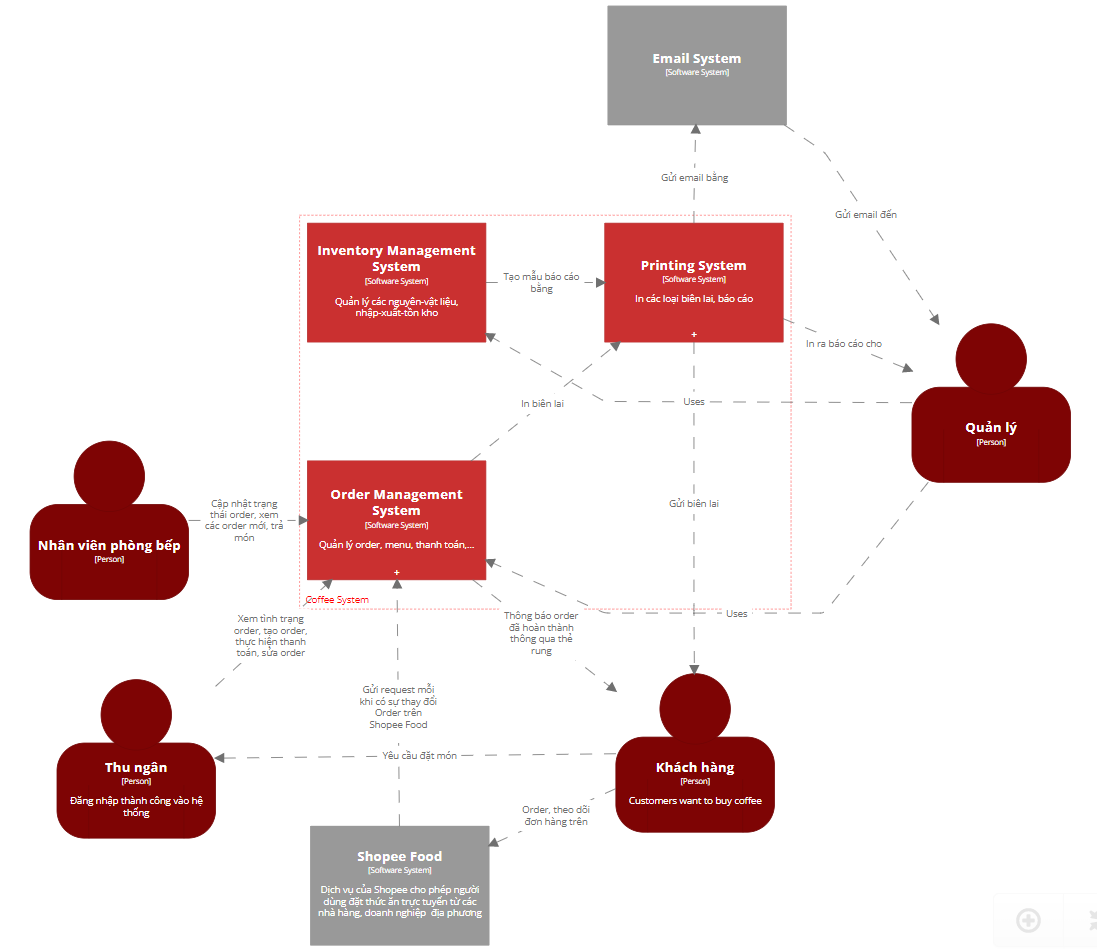
**C**omponent: Sở đồ này phóng to container để mô tả về thành phần bên trong; do đó hiển thị được các components (building block), mục đích và nhiệm vụ, mối quan hệ của từng component cũng như là những giao thức kết nối. Sơ đồ này thường được gọi là sơ đồ khối cấp 2.

**C**ode (hay **c**lass): Sơ đồ này mô tả bên trong một component; theo đó hiển thị được các phần hiện thực như là class, package… cùng với mối quan hệ giữa những thành phần này.

## 2. Sử dụng mô hình C4 model để giải quyết bài toán

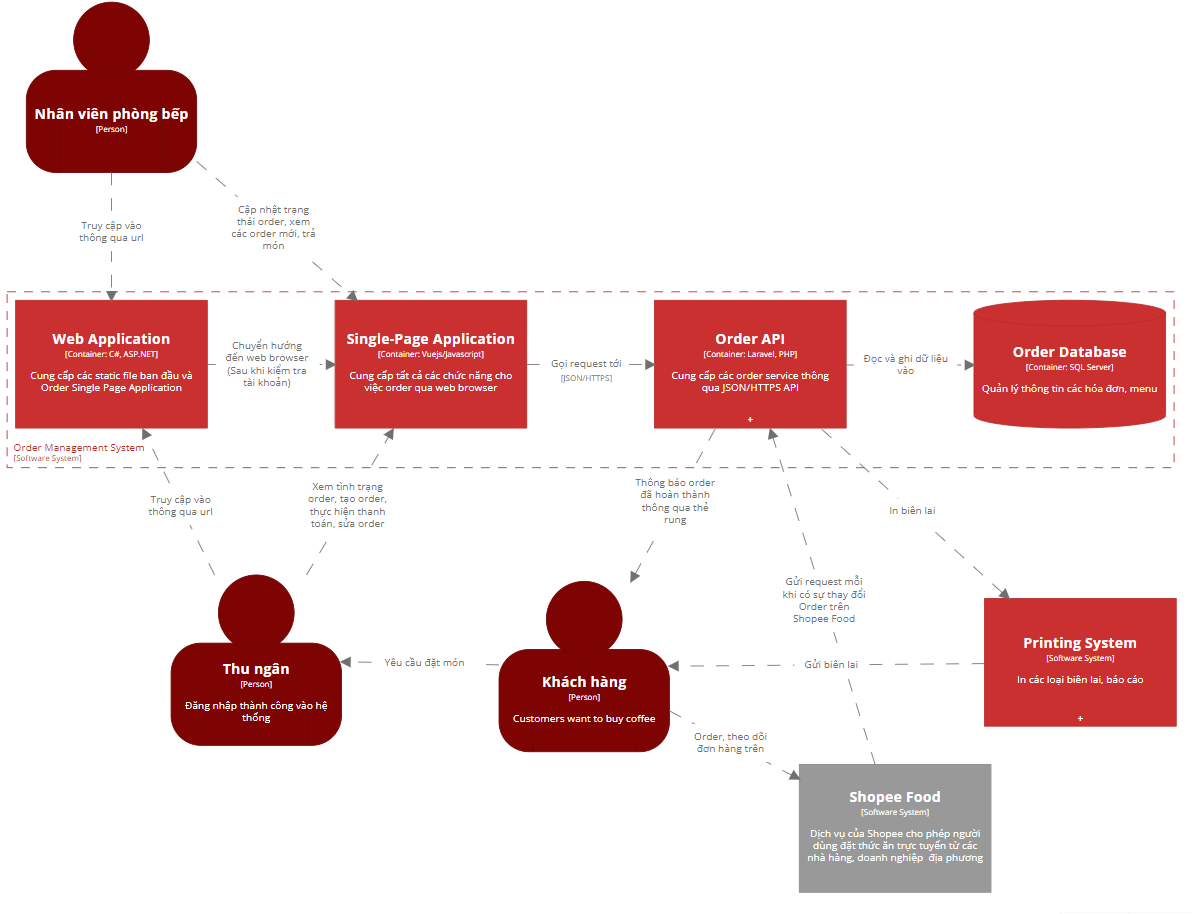
### Level 1: Context Diagram for a Coffee Shop System

* Khách hàng xem menu, yêu cầu đặt món , nhận món khi thẻ rung được rung lên
* Nhân viên thu ngân tạo các yêu cầu gọi món, thanh toán hóa đơn, in biên lai trên Order management system , thực hiện sửa/xóa các món trong order nếu như chưa được chế biến
* Nhân viên phòng bếp tiếp nhận, xem order và cập nhận trạng thái các món trong order, hoặc là yêu cầu trả món nếu bị thiếu nguyên vật liệu
* Printing System là hệ thống thực hiện việc tạo lập các mẫu biên lai, báo cáo, sau đó tùy thuộc vào nhu cầu mà có thể in ra mẫu hoặc là gửi mẫu qua email. Ngoài ra, hệ thống còn cho phép người dùng có thể quản lý các mẫu template được dùng cho nhiều loại mẫu biên lai, báo cáo khác nhau.
* Inventory Management System cho phép quản lý có thể quản lý các mặt hàng trong kho, tình trạng xuất nhập kho , tìm kiếm theo thông tin các mặt hàng, xem báo cáo. Ngoài ra, quản lý còn có thể quản lý các phiếu nhập kho, phiếu xuất kho, nhận cảnh báo khi số lượng mặt hàng nào đó giảm dưới hạn mức tồn hoặc mặt hàng đó sắp hết hạn



### Level 2: Container Diagram for a Coffee Shop System

**Order management system:**



* **Web application**

Web application yêu cầu một web server để quản lý các yêu cầu từ máy khách, một application server để thực hiện các tác vụ được yêu cầu và đôi khi, một database để lưu trữ thông tin. Công nghệ application server có các loại từ ASP.NET, ASP và ColdFusion, đến PHP và JSP.

Người dùng cụ thể là nhân viên hoặc khách hàng kích hoạt request tới web server qua Internet, thông qua trình duyệt web hoặc giao diện người dùng của ứng dụng. Web server chuyển tiếp request này đến web application server thích hợp.

Máy chủ ứng dụng Web (web application server) thực hiện nhiệm vụ được yêu cầu - chẳng hạn như truy vấn cơ sở dữ liệu hoặc xử lý dữ liệu - sau đó tạo ra các kết quả của dữ liệu được yêu cầu.

Máy chủ ứng dụng web gửi kết quả đến máy chủ web với thông tin được yêu cầu hoặc dữ liệu đã được xử lý.

Máy chủ web phản hồi response lại cho khách hàng các thông tin được yêu cầu sau đó xuất hiện trên màn hình của người dùng.

***Công nghệ sử dụng: C#, ASP.NET***

* **Single-page application (SPA)**

Ở single page, người dùng có 2 hướng lựa chọn:

+Chuyển trang trực tiếp để thực hiện yêu cầu chức năng mà không cần load lại trang nếu như đang đứng tại SPA.

+ Truy cập vào web application thông qua url rồi chuyển hướng đến SPA và thực hiện yêu cầu chức năng order sau khi đã đăng nhập vào hệ thống thành công.

Single page application và Client-side rendering (CSR) kết hợp với nhau sẽ giúp nâng cao trải nghiệm của người dùng một cách nhanh, hiệu quả. Trang web tương tác với người dùng bằng cách tự động viết lại trang hiện tại thay vì tải các trang hoàn toàn mới từ máy chủ. Cách tiếp cận này tránh làm gián đoạn trải nghiệm người dùng giữa các trang liên tiếp, giúp ứng dụng hoạt động mượt mà hơn đồng thời giúp giảm gánh nặng cho phía server. Trong một SPA, tất cả các file tĩnh như HTML, JavaScript và CSS được truy xuất với một lần tải trang đầu tiên hoặc các tài nguyên thích hợp được tải động và thêm vào trang khi cần, thường là để phản hồi lại hành động của người dùng. Trang không tải lại tại bất kỳ thời điểm nào trong quy trình, cũng như không chuyển quyền kiểm soát sang trang khác. Điểm khác biệt là nó chỉ lấy và hiển thị các dữ liệu cần thiết dựa theo yêu cầu của người dùng chứ không phải tải lại hoàn toàn dữ liệu trang web như cách làm truyền thống

Với Single page application, việc xử lý sẽ được thực hiện phần lớn ở phía client. Phía frontend sẽ tiếp nhận request từ người dùng, xử lý và lấy ra những thông tin cần thiết rồi mới gửi yêu cầu tới server để xử lý.

Nếu như theo mô hình truyền thống, việc chuyển hướng giữa các url được xử lý ở phía backend, thì với single page application, chúng sẽ được thực hiện ở phía frontend.

Theo mô hình Single page application, backend trả dữ liệu cho frontend dưới dạng chuỗi json, frontend sẽ dựa vào chuỗi json này để kết xuất (render) ra giao diện cho người dùng. Quá trình này được gọi là client side rendering.

***Công nghệ sử dụng: Javascript, Vue.js***

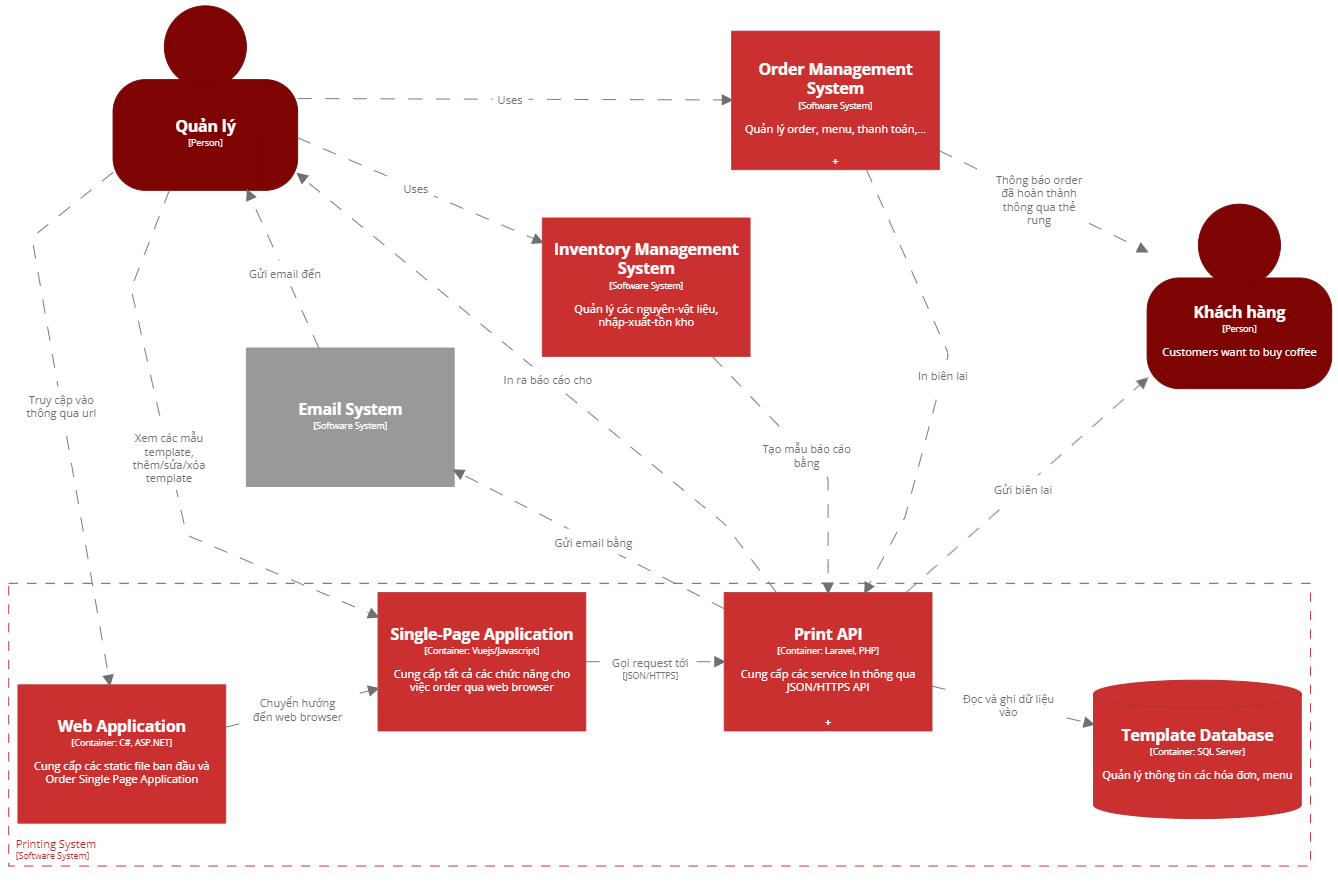
* **Order api**

Cho phép cung cấp thông tin các order service của quán thông qua JSON/HTTPS API, nhờ có API mà nhân viên sẽ nhận yêu cầu từ khách hàng và truyền đến (hệ thống) pha chế những thứ cần làm. Sau đó nhân viên sẽ phản hồi ngược lại cho khách hàng, trong trường hợp này, họ sẽ thông báo cho khách hàng thông qua thiết bị cảnh báo.

* **Order database**

Nơi lưu trữ lại dữ liệu các menu, hóa đơn mà khách hàng đã order và có thể truy xuất lại thông tin hóa đơn của khách hàng theo thời gian.

**Printing system:**



* **Print api**

Cung cấp các service in thông qua JSON/HTTPS API như order để in ra báo cáo hoặc hóa đơn để phản hồi lại với quản lý

* **Web Application**

Công nghệ sử dụng: C#, ASP.NET

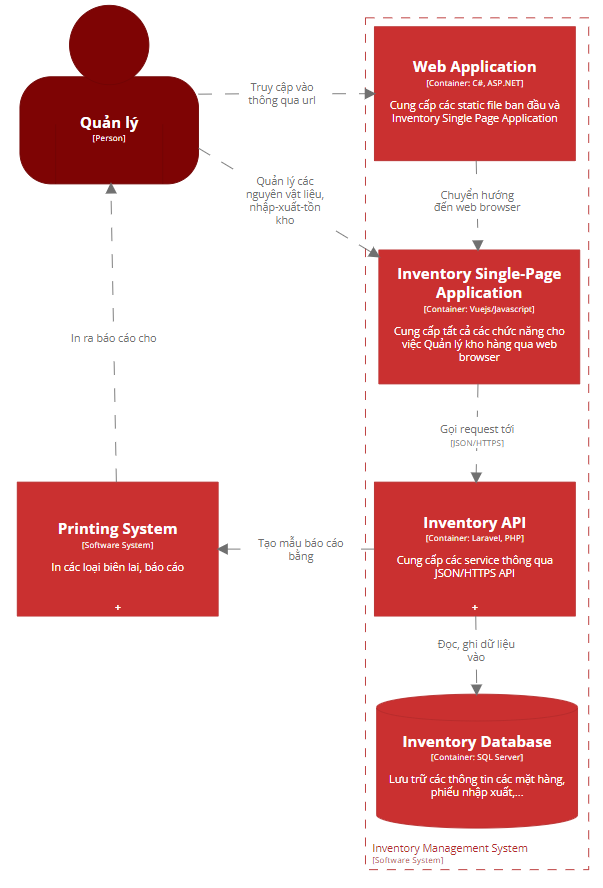
* **Single-page Application**

Công nghệ sử dụng: Javascript, Vue.js

* **Template database**

Sử dụng SQL Server để lưu trữ các thông tin các mẫu template của các biên lai, báo cáo, table, mẫu email,.. phục vụ mọi nhu cầu sử dụng

**Inventory management system:**

****

* **Inventory api**

Cung cấp các service quản lý kho hàng thông qua JSON/HTTPS API như quản lý nhập/xuất/tồn, quản lý các nguyên vật liệu và mặt hàng trong kho, tìm kiếm thông tin, xem báo cáo, cảnh báo khi số lượng dưới hạn mức tồn hoặc gần hết hạn sử dụng,..

* **Web Application**

Công nghệ sử dụng: C#, ASP.NET

* **Single-page Application**

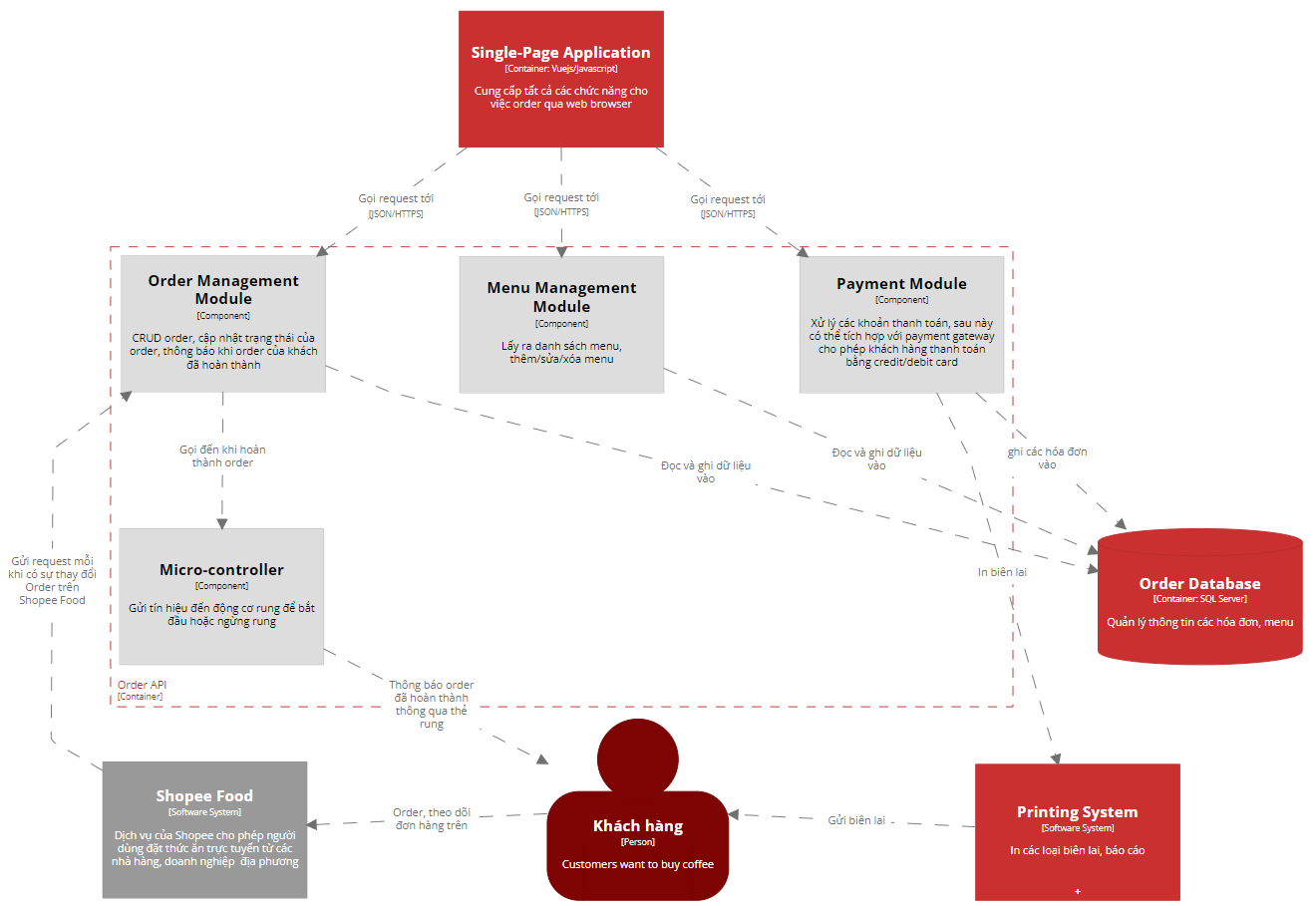
Công nghệ sử dụng: Javascript, Vue.js

* **Inventory database:**

Sử dụng SQL Server để lưu trữ các thông tin của phiếu nhập/ xuất kho như ngày nhập/xuất, tổng giá trị nhập/xuất, thông tin từng mặt hàng trong phiếu nhập/nhập như mã chứng từ, mã mặt hàng, số lượng, giá nhập/xuất, ngày hết hạn sử dụng , ngày thông báo hết hạn sử dụng, cuối cùng thông tin các mặt hàng như tên mặt hàng, số lượng, hạn sử dụng, vị trí lưu kho …

### Level 3: Component Diagram for a Coffee Shop System

- OrderAPI :



* + Menu Management Module :
  + CRUD menu : Hiển thị menu , tìm kiếm menu , chỉ cho phép quản lý thêm, sửa, xóa menu ví dụ như thông tin sản phẩm, công thức chế biển
  + Payment Module: Module này chịu trách nhiệm xử lý thanh toán từ khách hàng. Sau này , nó có thể được tích hợp với một cổng thanh toán (chẳng hạn như Stripe hoặc PayPal) để cho phép khách hàng thanh toán bằng credit/debit card của họ. Tất nhiên vẫn chấp nhận thanh toán bằng tiền mặt. Và cuối cùng là gửi request tới cho printing system thông qua GET method nhằm để in biên lai
  + Order Management Module :Nhờ module này, khách hàng có thể thông qua thu ngân để biết được tình trạng order của họ do hệ thống sẽ liên tục tracking và khách hàng sẽ được thông báo thông qua thẻ rung khi order của họ được sẵn sàng
  + CRUD order : Nhân viên có thể xem các order (đang chuẩn bị hoặc đã được phục vụ), thu ngân có thể tạo lập order và chỉ có thể sửa/xóa món khi món đó trong trạng thái “pending”.
  + Webhook endpoint : Bằng việc sử dụng Webhook Order (order\_status\_push) Tiếp nhận và xử lý các request của Shopee khi có sự thay đổi trạng thái đơn hàng như thêm mới, update trên ứng dụng Shopee sau đó chúng ta sẽ tiến hành đồng bộ với hệ thống.. Để set up để nhận POST request từ shopee tham khảo:

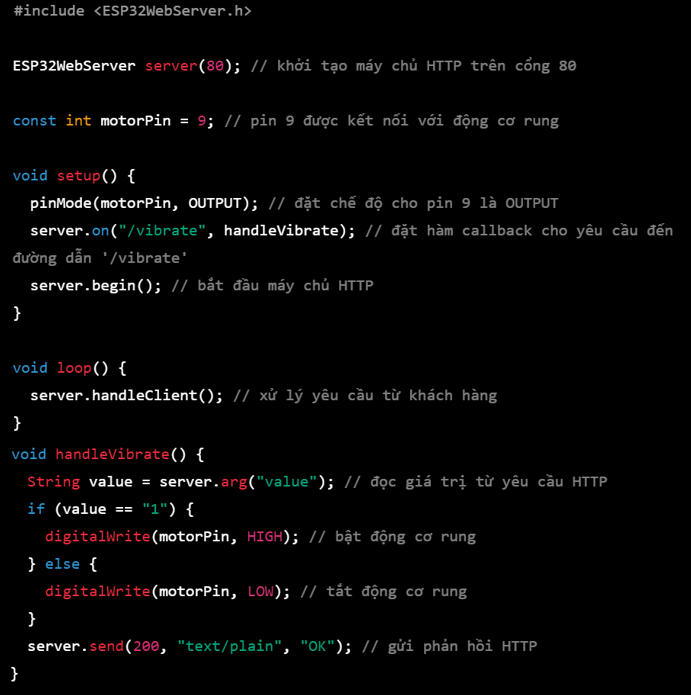
<https://open.shopee.com/developer-guide/18>

<https://open.shopee.com/documents/v2/v2.push.set_push_config?module=105&type=1>

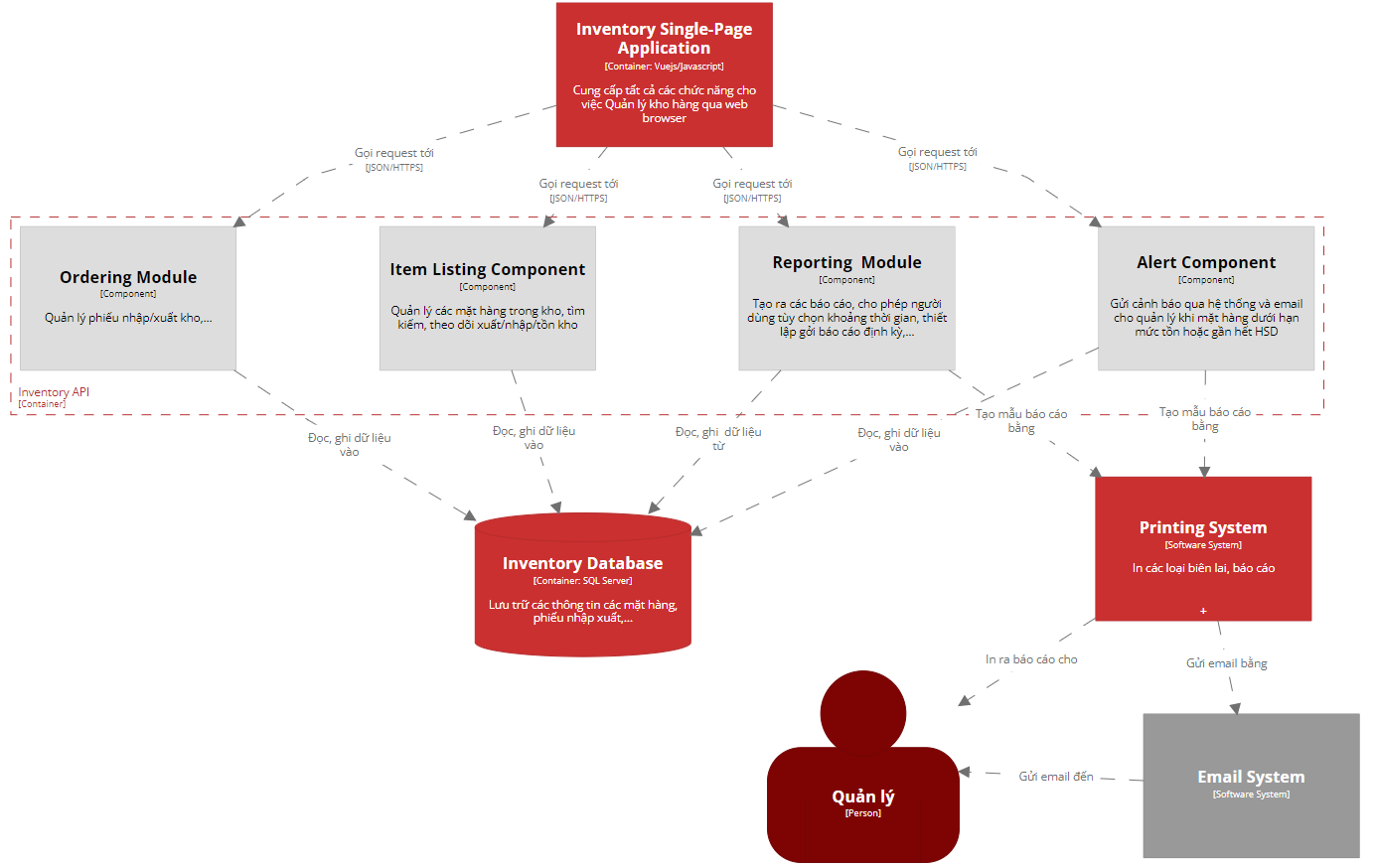
* + Cập nhật trạng thái các món/order : Nhân viên phòng bếp khi bắt đầu chuẩn bị món nào thì sẽ cập nhật trạng thái món đó từ “pending” sang “in progress”, và cập nhật từ “in progress” sang “completed” khi hoàn thành món đó
  + Thông báo order đã hoàn thành: Sau khi nhân viên phòng bếp cập nhật trạng thái order sang “served” thì hệ thống sẽ gọi tới Mico-controller
  + Micro-controller : Là một endpoint để kiểm soát việc gửi tín hiệu bật/tắt động cơ rung ở trong thẻ rung được chỉ định. Giúp thông báo cho khách hàng rằng order của họ đã sẵn sàng để phục vụ. Cụ thể thì :
  + Tạo một post request có data là 1 gửi tới hostname của esp32 với path tương ứng với id thẻ rung được chỉ định để bắt đầu rung
  + Sau 1 khoảng thời gian sẽ gửi 1 request tương tự với data là 0 để ngừng rung

Để làm được như vậy trước hết cần phải làm theo các bước sau:

1. Set up các đèn LED, mạch ESP32 và động cơ rung tương thích
2. Viết mã cho mạch để điều khiển động cơ rung và đèn LED trong Arduino IDE. Sau đó nạp chương trình cho các mạch. Khi có một request từ client gửi tới thì trong trường hợp data là 1 thì mạch sẽ bật động cơ rung và đèn LED, còn nếu data là 0 thì mạch sẽ tắt động cơ rung và đèn LED
3. Gia công thẻ rung bằng cách gắn mạch Arduino, động cơ rung, nút nhấn, và đèn LED vào một thẻ nhỏ (để lại cổng USB). Sau đó đánh dấu các thẻ rung cho từng bàn.

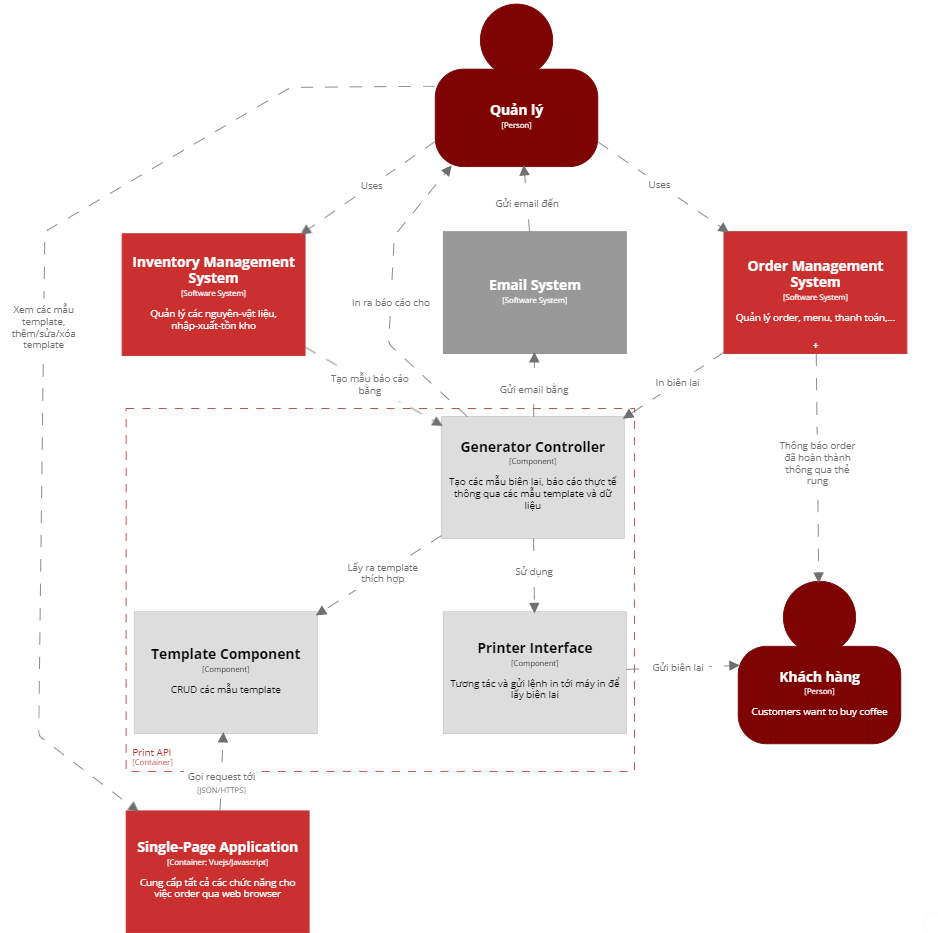


- InventoryAPI :



* + Item listing component : Đây là một component giúp quản lý các mặt hàng trong kho bao gồm cả thông tin cho các lần xuất, nhập kho của nó như ngày giờ, số lượng, giá trị, vị trí lưu kho,… Module này còn cung cấp các chức năng khác như tìm kiếm, tính giá trị tồn trong kho như số lượng, giá trị, hạn sử dụng, ngày thông báo gần hết hạn sử dụng,…
  + Ordering module : Đây là một component nhằm để tạo các đơn nhập hàng sau khi đã thương lượng với bên nhà cung cấp, ngoài ra còn có thể giúp theo dõi đơn nhập hàng đó cho đến khi đơn hoàn thành hoặc bị hủy. Component này cho phép tự động sinh ra các phiếu nhập hàng tạm cho các mặt hàng dưới hạn mức tồn để việc bổ sung hàng có thể được thực hiện nhanh nhất có thể. Ngoài ra, quản lý có thể tạo các phiếu xuất kho bao gồm các thông tin như tên các mặt hàng, số lượng, thời gian để tiện cho việc quản lý và phân phối các nguyên vật liệu tới các chi nhánh.
  + Alert component: Đây là một component giúp gửi các cảnh báo qua hệ thống và email cho quản lý trong trường hợp có mặt hàng dưới hạn mức tồn hoặc là khi đến thời gian thông báo gần hết hạn sử dụng thông qua email và trên hệ thống.
  + Reporting module: Module này giúp tạo ra các báo cáo Nhập, xuất, tồn trong khoảng thời gian bất kỳ, cung cấp các tùy chọn về cách trình bày báo cáo chẳng hạn như dạng bảng, đồ thị hoặc biểu đồ. Ngoài ra, module này có thể được thiết lập để gửi báo cáo hàng ngày về mức tồn kho hiện tại và bất kỳ mặt hàng nào sắp hết, để có thể lên kế hoạch trước cho việc bổ sung hàng. Module này có thể gọi đến Printing System khi người dùng muốn in ra báo cáo

PrintAPI :



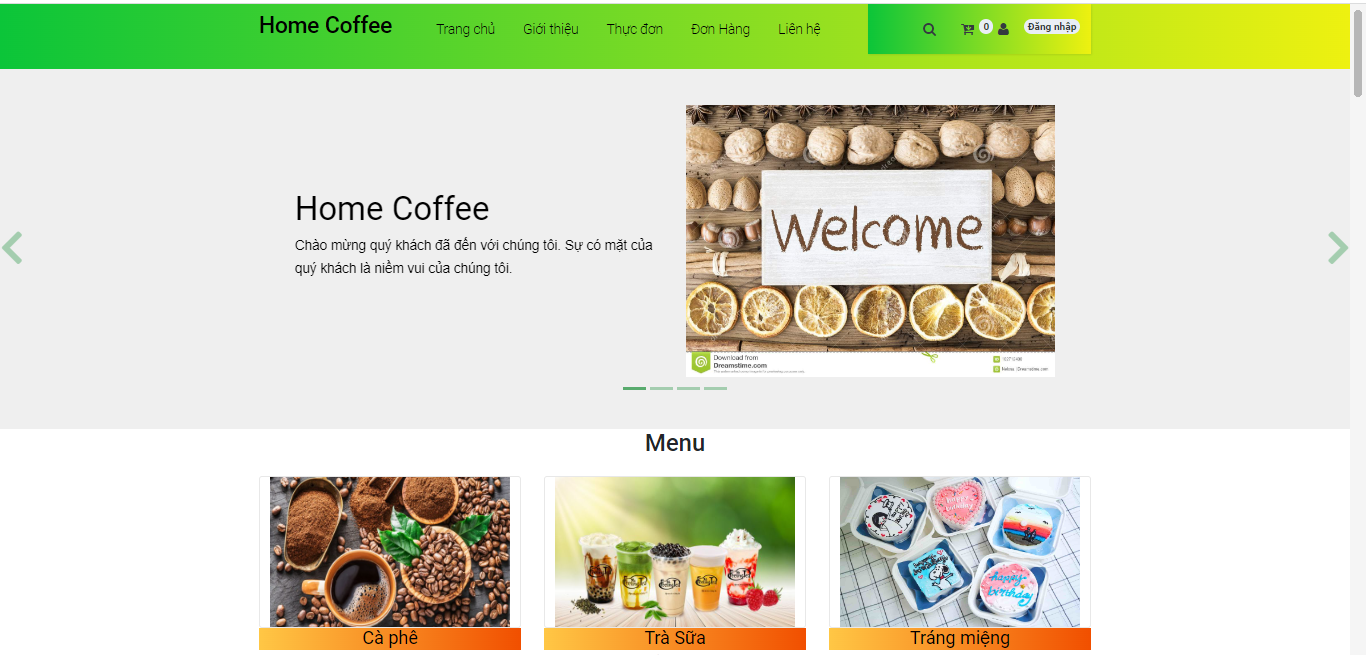
* + Template component : Đây là một component giúp hiển thị, thêm xóa sửa các template nhằm xác định bố cục và định dạng của từng mẫu báo cáo, biên lai, bao gồm phông chữ, màu sắc, và bất kỳ hình ảnh hoặc logo cần thiết. Các template ở đây có cú pháp phụ thuộc vào template engine compile và thực thi nó như ejs, Handlebars.js , JSP , Django templates, Jinja2.
  + Generator controller : Đây là một component giúp generate ra các mẫu báo cáo, biên lai thực tế dựa vào mẫu template sẵn có và data được truyền vào thông qua query param (GET / HEAD), hoặc thông qua phần body của các HTTP Request còn lại. Giả sử chúng ta cần in một biên lai,thì cần phải truyền các data cần thiết để in ra biên lai như Tên món, đơn giá, số lượng, tổng tiền, thời gian, các khoản phí hoặc thuế,… . Ngoài ra thì component này sẽ dùng template engine như ejs, Handlebars.js , JSP , Django templates, Jinja2 để compile và render mẫu template cùng với data. Component này cho phép in trực tiếp bằng máy in bằng cách gọi đến printer interface component hoặc tùy chọn gửi qua email bằng cách sử dụng Email System.
  + Printer interface: Module này sẽ tương tác trực tiếp với máy in để in ra biên lai hoặc báo cáo. Module này sẽ sử dụng các printer driver để thực hiện những công việc trên. Giả sử chúng ta sử dụng node-printer thì sẽ truyền vào data là template đã được render, tên máy in và type là “RAW”,”TEXT”, “PDF”,.. tùy thuộc vào nhu cầu. Sau đó module này sẽ gởi lệnh in tới máy in để in ra mẫu.



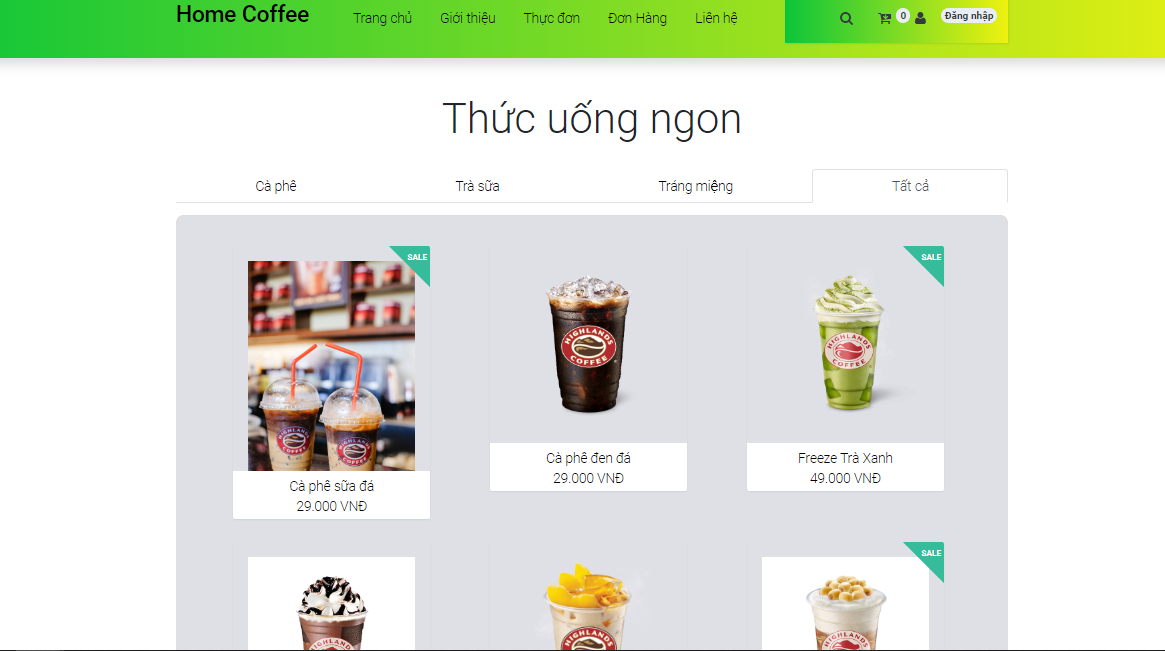
# III. Demo

## 1. Giao diện người dùng

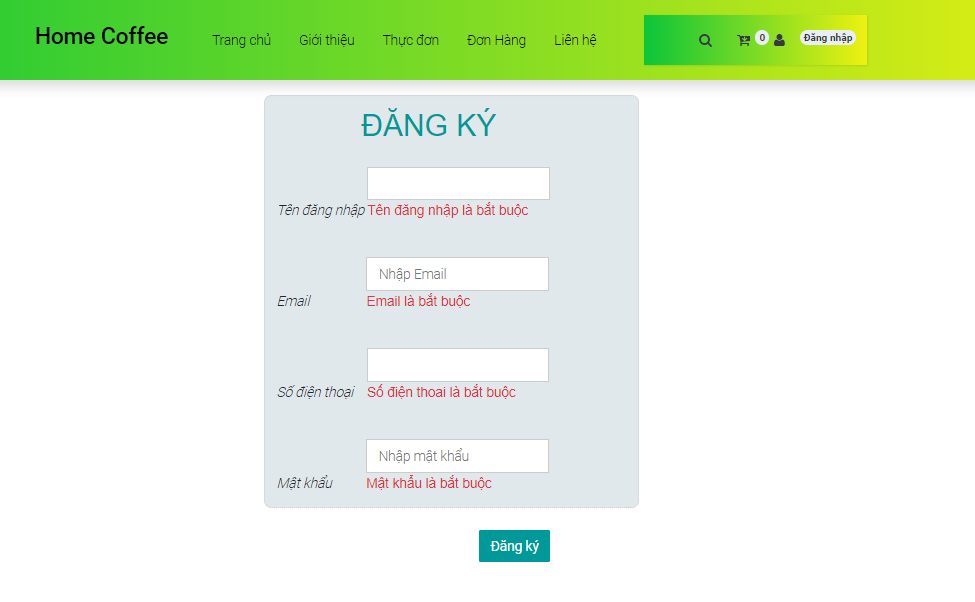
### - Trang chủ



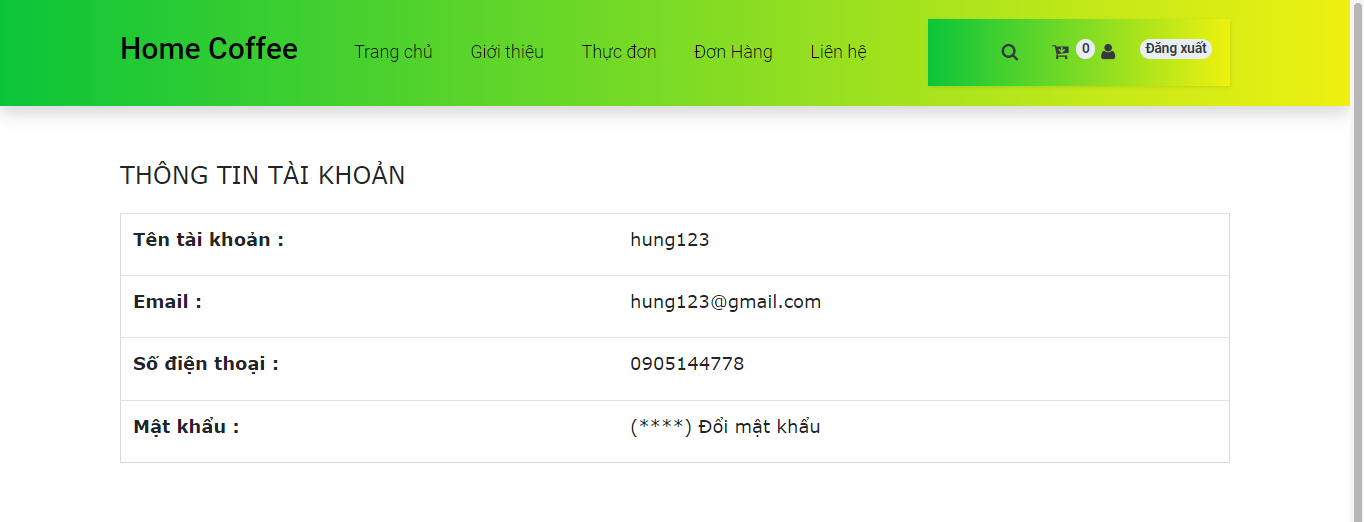
### - Trang thực đơn



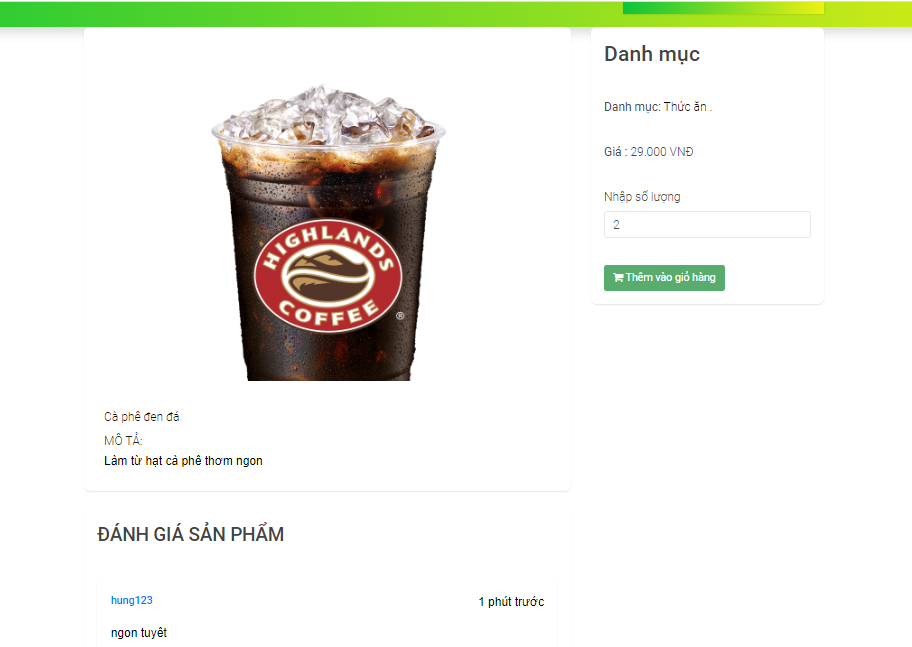
### - Trang đăng ký



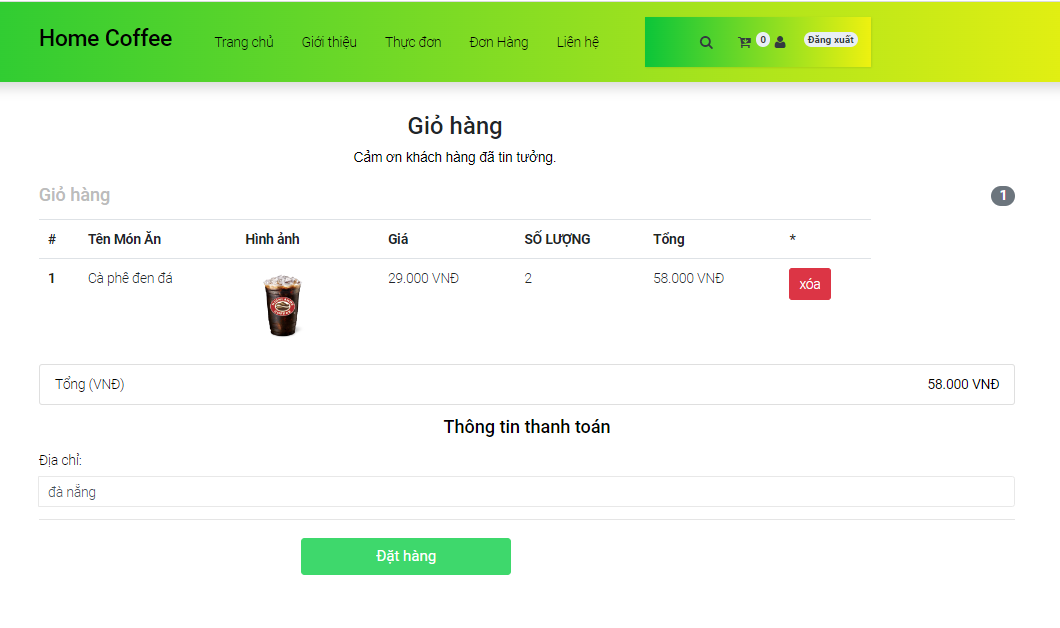
### - Trang quản lý thông tin cá nhân



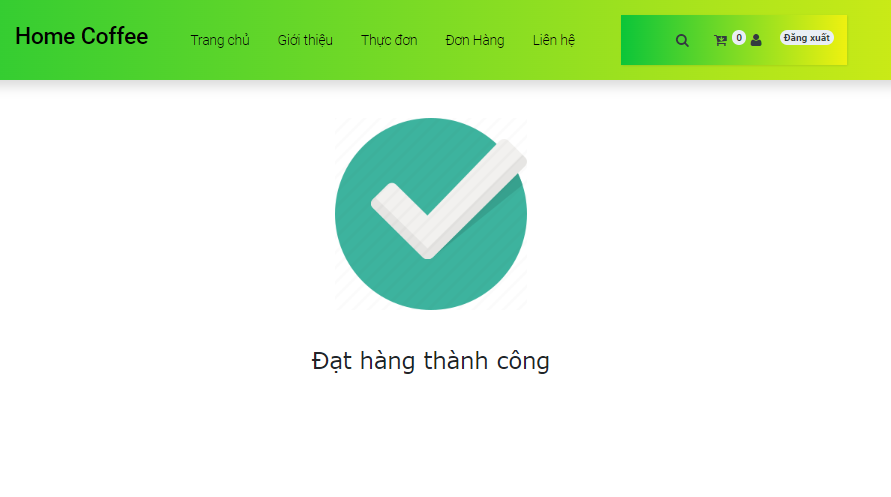
### - Trang chi tiết sản phẩm



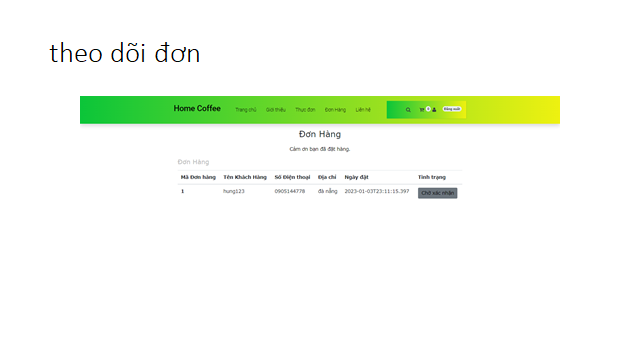
### - Trang giỏ hàng



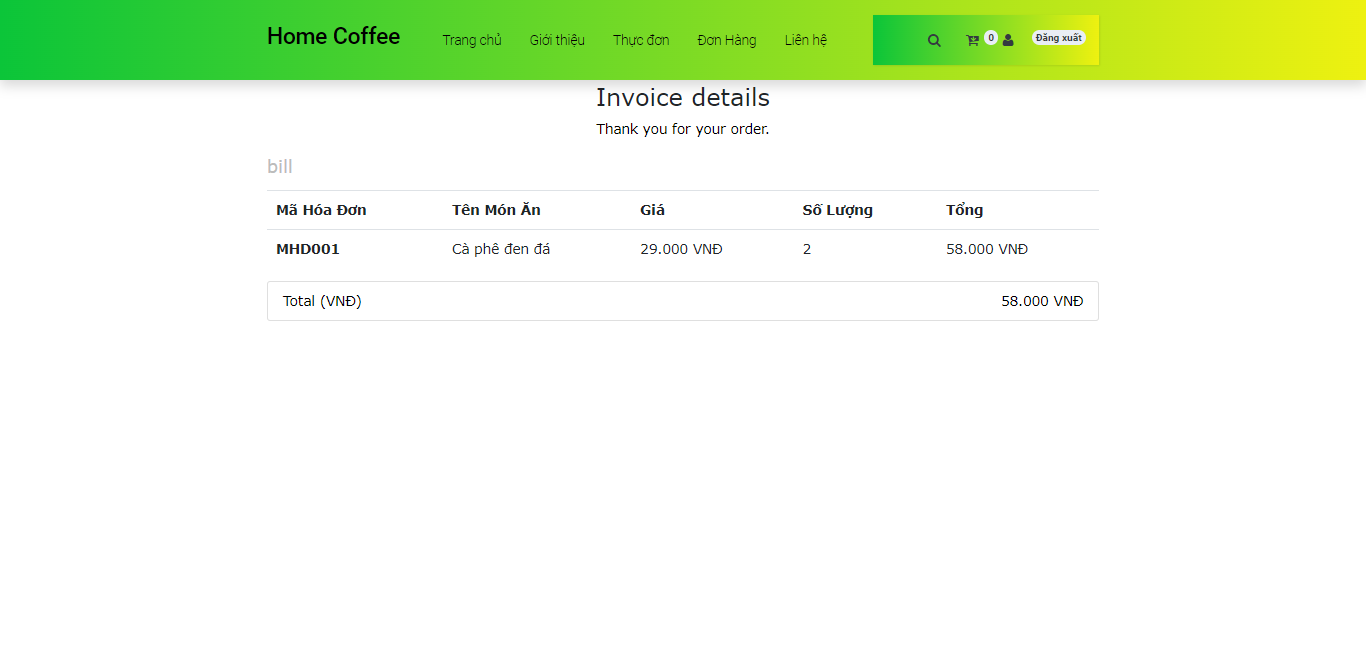
### - Trang đặt hàng thành công



### - Trang quản lý đơn hàng

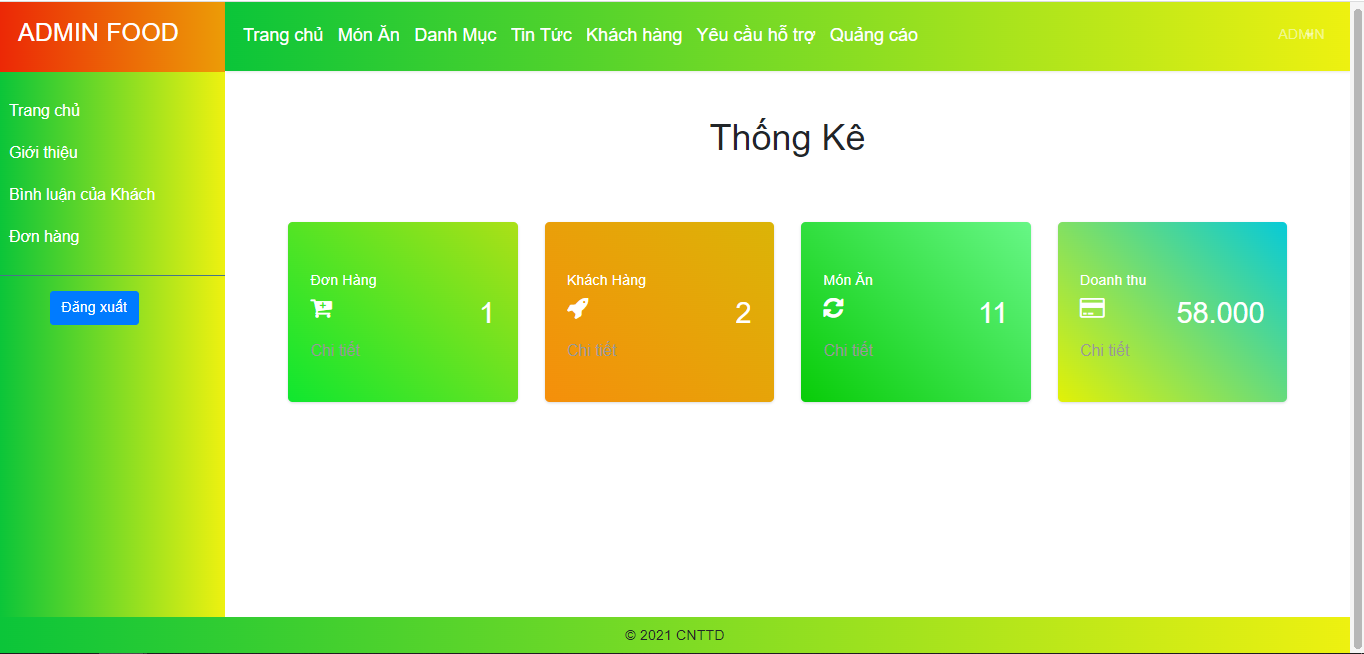


### - Trang chi tiết đơn hàng

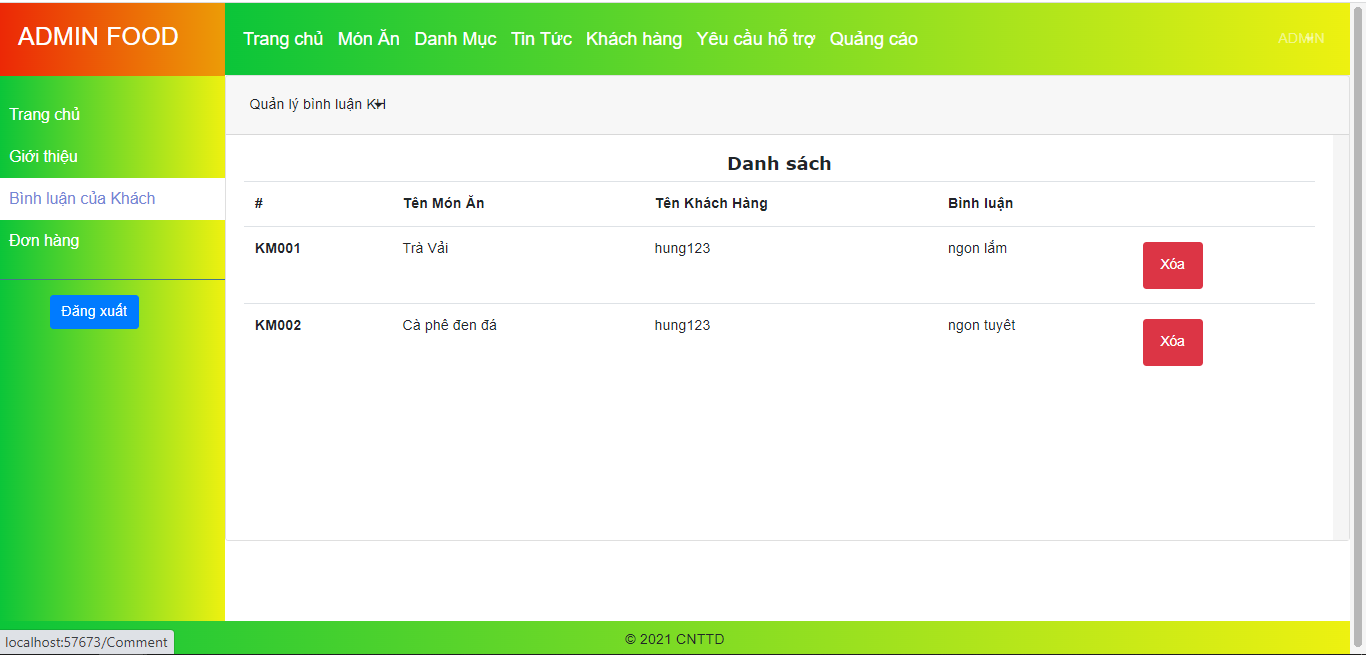


## 2. Giao diện quản trị viên

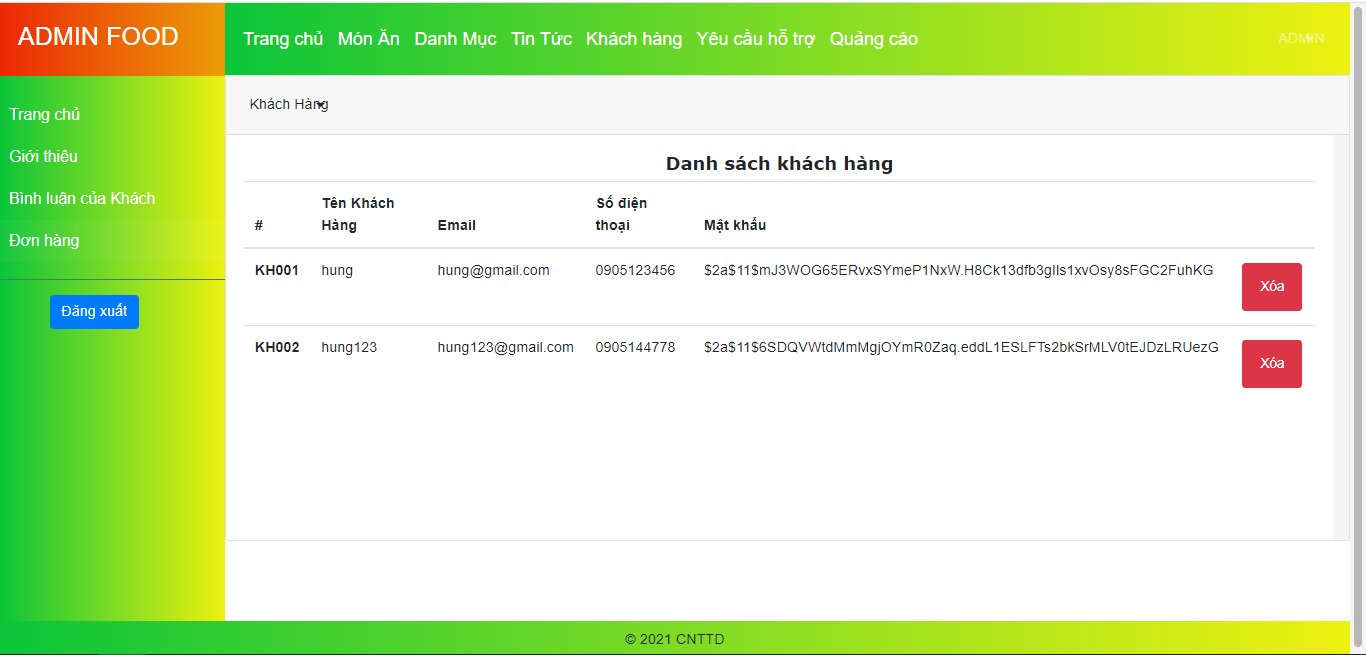
### - Trang chủ



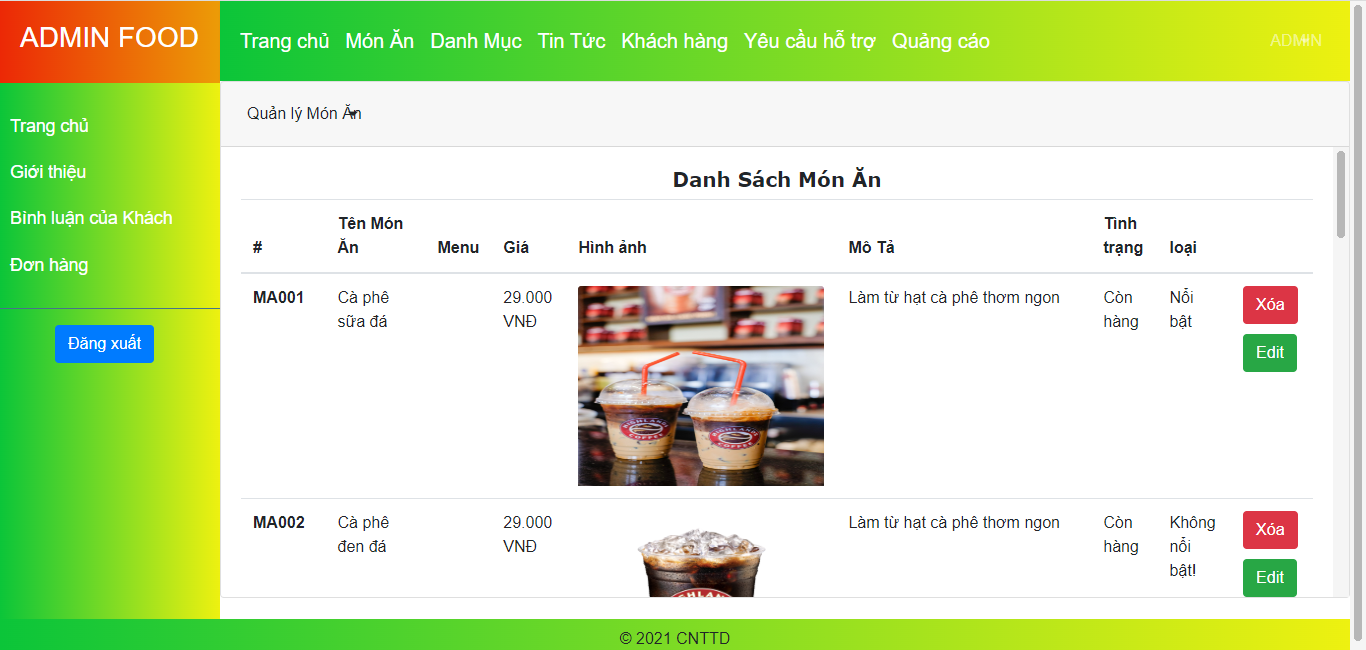
### - Trang quản lý bình luận



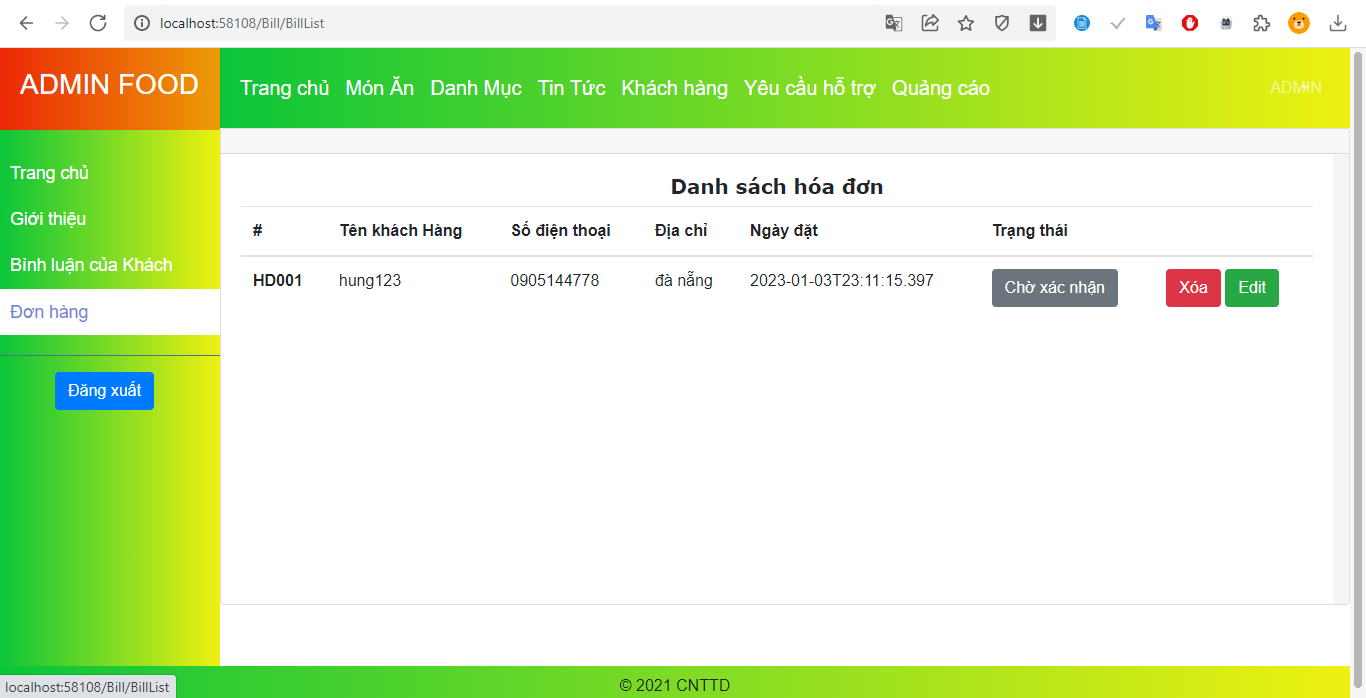
### - Trang quản lý người dùng



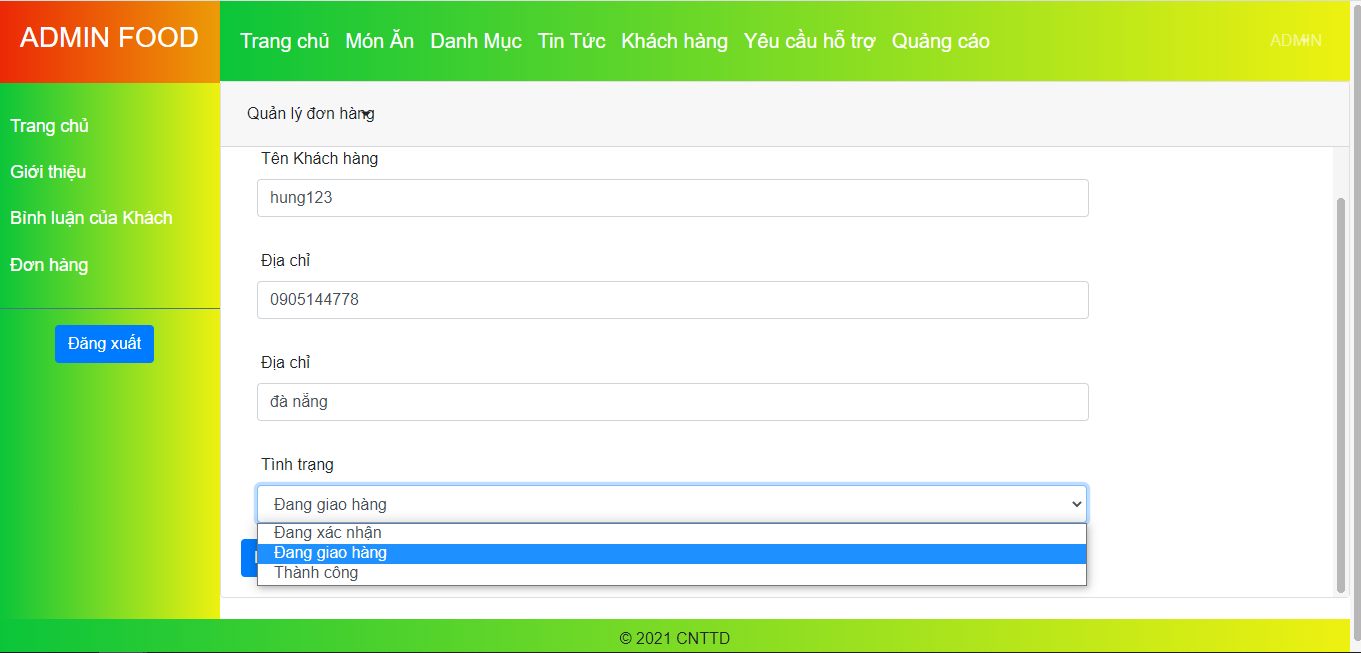
### - Trang quản lý sản phẩm



### - Trang quản lý đơn hàng



### - Trang chỉnh sửa đơn hàng



### - Trang cập nhật đơn hàng

