TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHÊ THÔNG TIN



Công nghệ phần mềm

**Ứng dụng Đặt hàng & Nhà hàng**

GVHD: Từ Lãng Phiêu

SV: Trần Thái Thanh Phương – 3122410332

Chu Thị Bảo Yến – 3122410496

Nguyễn Bạch Phú Lộc – 3122410220

Hoàng Quốc Cường – 3123410037

Võ Duy Thanh – 3123560080

Dương Tấn Tài – 3123410313

TP.HỒ CHÍ MINH, THÁNG 2/2025

**Task 1: Khai thác yêu cầu**

**1.1** **Xác định bối cảnh của dự án này. Ai là các bên liên quan chính? Những gì được kỳ vọng sẽ được thực hiện? Phạm vi của dự án là gì?**

Bối cảnh dự án: Hệ thống quản lí bán hàng cho 1 nhà hàng bán đồ ăn nhanh

Ai liên quan: Nhân viên, Khách hàng,Nhà bếp.

Kỳ vọng:

* Thanh toán thành công,linh hoạt
* Vật phẩm đầy đủ như thông tin đã đặt
* Đảm bảo bảo mật và an toàn thông tin cho khách hàng
* Tiết kiệm thời gian và nâng cao hiệu suất

Phạm vi dự án:

* Phạm vi dự án là những gì cần phải đạt được, những công việc cần thiết để hoàn thành dự án. Các key feature: giao diện POS, QR code, menu, Pay Bill
* Giao diện POS cho thu ngân trong trường hợp khách hàng muốn gọi môn trực tiếp QR code khởi tạo riêng cho mỗi bàn thuộc 1 nhà hàng trong chuỗi nhà hàng (nên có) có giá trị trong mạng nội bộ
* Cùng một lúc, mỗi khách hàng quét cũng một mã QR có 1 mà đơn hàng riêng biệt
* Các món ăn trong thực đơn được hiển thị trên màn hình theo nhóm, cùng mục lúc trên màn hình hiển thị tối đa 9 món ăn.
* Cho phép thanh toán tiền mặt (đối với những thực khách gọi món trực tiếp) Hình thức thanh toán không tiền mặt chấp nhận các ví điện tử Momo, Zalopay Shopeepay, thẻ tín dung, VISA, MasterCard
* Xây dựng hệ thống Web giao diện khách hàng
* Triển khai backend với **NestJS (TypeScript)** và frontend bằng **ReactJS**
* Kết nối cơ sở dữ liệu để lưu trữ đơn hàng, món ăn, người dùng,…
* Hỗ trợ thanh toán online (giả lập hoặc thật nếu tích hợp được)
* Đảm bảo hệ thống có thể triển khai trên nhiều thiết bị
* Tài liệu hướng dẫn sử dụng cơ bản

**1.2 Mô tả tất cả các yêu cầu chức năng và phi chức năng của hệ thống mong muốn. Vẽ sơ**

**đồ use-case cho toàn bộ hệ thống.**

Chức năng:

* Xem thực đơn
* Chọn món và đặt món
* Xem giỏ hàng
* Thanh toán đơn hàng
* Nhận món ăn
* Đánh giá

(Dự kiến)

* Nhận và ghi nhận đơn hàng
* Kiểm tra tính khả dụng món ăn
* Chuyển đơn hàng xuống bếp
* Ghi nhận thanh toán
* Lưu lịch sử giao dịch
* Cập nhật trạng thái đơn hàng
* Thống kê số lượng bán ra
* Quản lý thực đơn (thêm, sửa, xóa món)
* Xem đơn hàng
* Xử lý đơn hàng
* Đóng gói món ăn
* Quản lý phản hồi khách hàng

Phi chức năng:

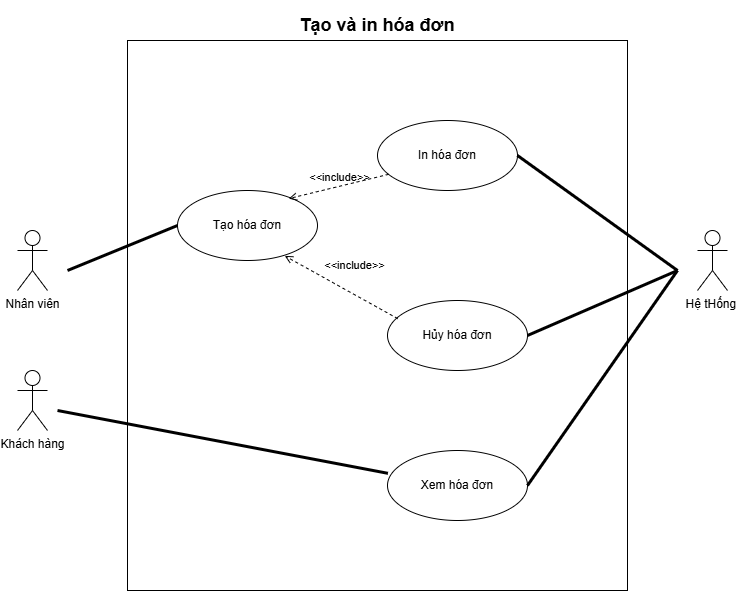
* Tương thích
* Dễ sử dụng
* Bảo mật
* Tin cậy
* Hiệu suất
* Yêu cầu về tổ chức phần mềm

Ảnh có chứa văn bản, biểu đồ, hàng, hình vẽ

Nội dung do AI tạo ra có thể không chính xác.

*Sơ đồ use case tổng quát*

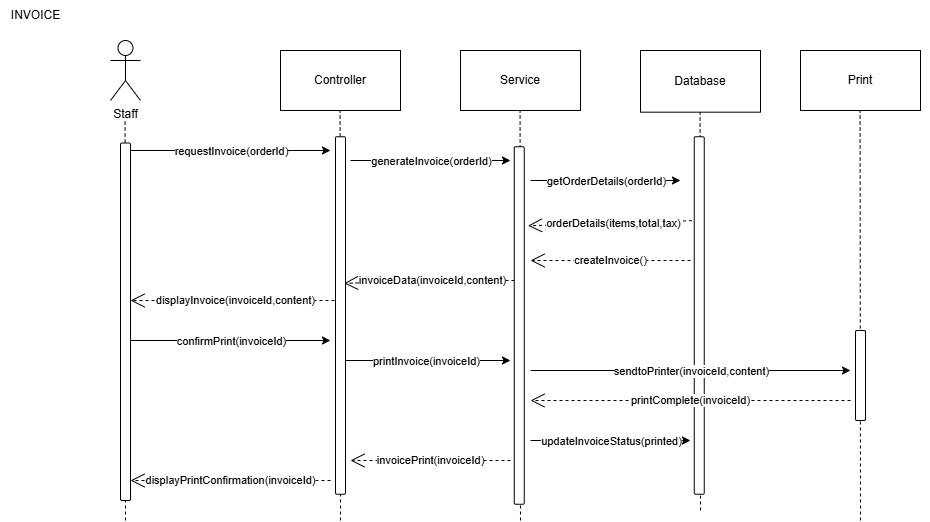
**1.3 Chọn một tính năng cụ thể, ví dụ: đặt món, đặt bàn, quản lý khách hàng. Vẽ sơ đồ use-case của nó và mô tả use-case bằng bảng.**



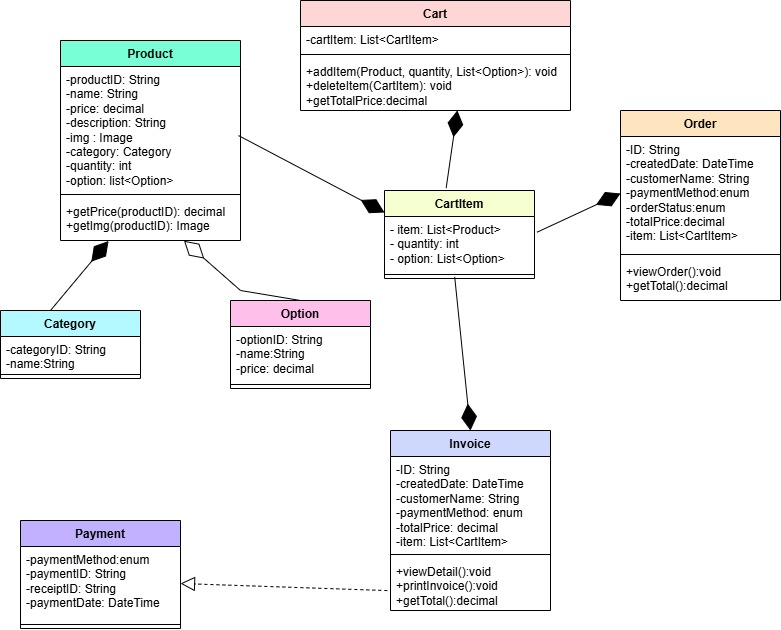
Chức năng tạo và in hóa đơn

|  |  |
| --- | --- |
| Tên use-case | Tạo và in hóa đơn |
| Tác nhân | Nhân viên,Khách hàng |
| Mô tả | Chức năng cho phép nhân viên tạo và in hóa đơn cho khách hàng sau khi hoàn tất thanh toán. |
| Dòng sự kiện chính | 1. Nhân viên nhận yêu cầu từ khách hàng. 2. Nhân viên nhập thông tin cần thiết để tạo hóa đơn. 3. Hệ thống xác nhận thông tin hóa đơn. 4. Hệ thống in hóa đơn. 5. Hệ thống gửi hóa đơn đến khách hàng |
| Tiền điều kiện | Khách hàng đã hoàn tất thanh toán,yêu cầu in hóa đơn. |
| Dòng sự kiện  phụ | - Nếu thông tin không chính xác, hệ thống thông báo lỗi và hủy hóa đơn để tạo lại. - Nếu máy in không hoạt động, hệ thống thông báo lỗi và yêu cầu kiểm tra máy in. |
| Hậu điều kiện | Hóa đơn được tạo và in thành công, và khách hàng nhận được hóa đơn. |

**Task 2: Mô hình hóa hệ thống**

**2.2 Vẽ sơ đồ trình tự cho use-case trong Task 1.3.**

*Sequence diagram*

**2.3 Vẽ sơ đồ lớp**

*Class diagram*

**Task 3: Thiết kế kiến trúc**

**3.1** **Mô tả phương pháp kiến trúc các nhóm sẽ sử dụng để triển khai hệ thống mong muốn.**

Do website bán hàng lần này được yêu cầu phải dùng Nestjs ở backend và có thể dùng Reactjs ở frontend nên chúng em sử dụng kiến trúc Decoupled Architecture. Kiến trúc tách rời là một phương pháp thiết kế phần mềm nhằm mục đích tăng cường tính linh hoạt, khả năng mở rộng và khả năng bảo trì bằng cách giảm sự phụ thuộc lẫn nhau giữa các thành phần hoặc mô-đun khác nhau của một hệ thống phần mềm.

**1. Frontend (ReactJS)**

* Giao diện người dùng chạy hoàn toàn trên trình duyệt
* Gọi API để lấy dữ liệu từ backend
* Có thể host ở một nơi riêng

**2. Backend (NestJS)**

* Xử lý nghiệp vụ, định tuyến API.
* Xử lý kết nối database,...
* Dễ mở rộng sang kiến trúc microservices nếu cần

**3. Database Layer**

* Dùng MySQL
* NestJS dùng TypeORM để kết nối đến database

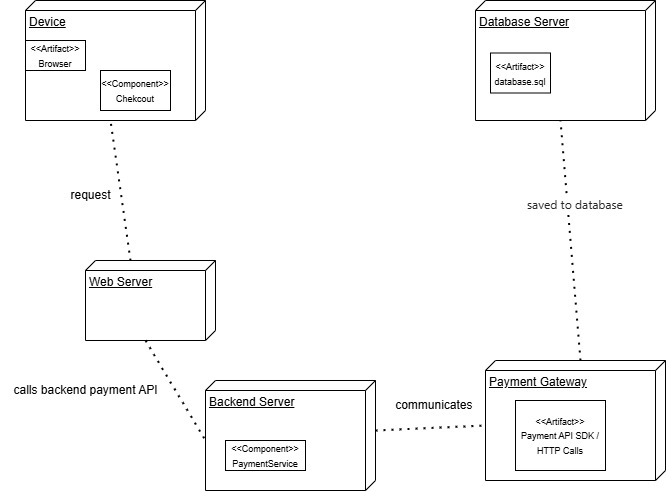
**3.2 Vẽ sơ đồ triển khai cho các yêu cầu chức năng chính (không phải tất cả)**

1. Đặt món

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Kế hoạch

Nội dung do AI tạo ra có thể không chính xác.

1. Thanh toán

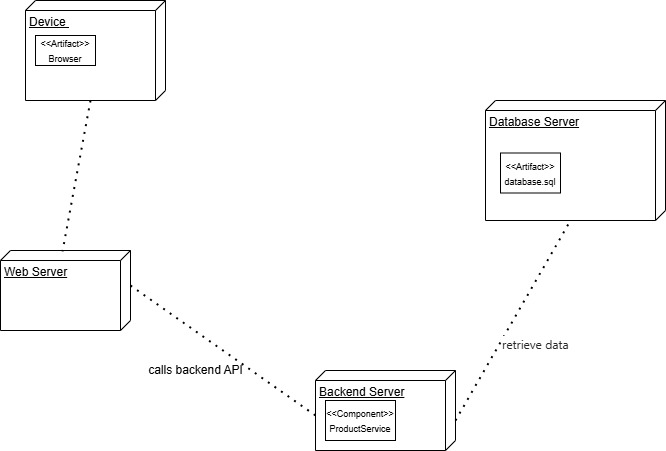


1. Hỗ trợ quét QR để vào trang web

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Kế hoạch

Nội dung do AI tạo ra có thể không chính xác.

1. Xem thông tin chi tiết món ăn



1. Xem và chọn thêm món ăn kèm khi nhấn vào hình ảnh món ăn

Ảnh có chứa văn bản, biểu đồ, bản phác thảo, Kế hoạch

Nội dung do AI tạo ra có thể không chính xác.

1. Xem hóa đơn và in hóa đơn

**Ảnh có chứa văn bản, biểu đồ, bản phác thảo, hàng

Nội dung do AI tạo ra có thể không chính xác.**