ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue circle with text

Description automatically generated

ĐỒ ÁN MÔN HỌC: CÔNG NGHỆ PHẦN MỀM

**ĐỀ TÀI: Tạo 4 Services (User, Product, Order, Payment) giao tiếp với nhau**

GVHD: Nguyễn Quốc Huy.

Lớp: DCT122C5.

Nhóm: Trần Gia Phúc\_3122411158.

Lê Tấn Đạt\_3122411038.

Thành phố Hồ Chí Minh, tháng 9 năm 2025

**BẢNG PHÂN CHIA CÔNG VIỆC**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thành viên** | **Phân công công việc** | **Mức độ hoàn thành** |
| 1 | Trần Gia Phúc – 3122411158 | 50% công việc | 100% |
| 2 | Lê Tấn Đạt – 3122411038 | 50% công việc | 100% |

**LỜI CAM ĐOAN**

Tôi xin cam đoan nội dung được trình bày trong đồ án “**3. Tạo 4 services (User, Product, Order, Payment) giao tiếp với nhau**” là do chúng tôi nghiên cứu, tìm hiểu và phát triển dưới sự dẫn dắt của GV: Nguyễn Quốc Huy. Đồ án không sao chép từ các tài liệu, công trình nghiên cứu của người khác mà không ghi rõ trong tài liệu tham khảo. Tôi xin chịu trách nhiệm về lời cam đoan này.

TP. Hồ Chí Minh, ngày tháng 9 năm 2025.

**LỜI CẢM ƠN**

Tôi xin gửi lời cảm ơn tới các thầy cô trường đại học Sài Gòn, đã tận tình giảng dạy và truyền đạt kiến thức trong suốt thời gian vừa qua. Tôi cũng xin được gửi lời cảm ơn đến các thầy cô trong bộ môn "Công Nghệ Phần Mềm" cũng như khoa Công Nghệ Thông Tin đã mang lại cho tôi những kiến thức vô cùng quý giá và bổ ích trong quá trình học tập tại trường.

Đặc biệt xin chân thành cảm ơn thầy giáo, giảng viên: Nguyễn Quốc Huy, người đã định hướng, giúp đỡ, trực tiếp hướng dẫn và tận tình chỉ bảo tôi trong suốt quá trình nghiên cứu, hoàn thiện đồ án này.

TP.Hồ Chí Minh, ngày tháng 9 năm 2025.

**Mục lục**

[LỜI MỞ ĐẦU 6](#_Toc210352357)

[1. TỔNG QUAN DỰ ÁN 6](#_Toc210352358)

[1.1. Mô tả dự án 6](#_Toc210352359)

[1.2. Hiện trạng khách hàng 7](#_Toc210352360)

[1.2.1. Giới thiệu về công ty FuDa Food 7](#_Toc210352361)

[1.2.2. Quy trình nghiệp vụ hiện tại 8](#_Toc210352362)

[1.3. Yêu cầu nghiệp vụ 8](#_Toc210352363)

[1.3.1. Actor & Yêu cầu chức năng 8](#_Toc210352364)

[1.3.2. Yêu cầu phi chức năng 9](#_Toc210352365)

[1.4. Giải pháp lựa chọn 10](#_Toc210352366)

[1.5. Kế hoạch dự án 11](#_Toc210352367)

[1.5.1. Phạm vi và mục tiêu 11](#_Toc210352368)

[1.5.1.1. Phạm vi 11](#_Toc210352369)

[1.5.1.2. Mục tiêu 12](#_Toc210352370)

[1.5.2. Cơ sở hạ tầng 12](#_Toc210352371)

[1.5.2.1. Môi trường phát triển 12](#_Toc210352372)

[1.5.2.2. Phần cứng & phần mềm 13](#_Toc210352373)

[1.5.3. Lịch sử thực hiện 13](#_Toc210352374)

[1.5.4. Quản lý cấu hình 14](#_Toc210352375)

[1.5.4.1. Định nghĩa CI và baseline 14](#_Toc210352376)

[1.5.4.2. Thư mục dự án 14](#_Toc210352377)

[2. Tái thiết kế - Chuẩn hoá quy trình 15](#_Toc210352378)

[2.1. Chuẩn hoá quy trình 15](#_Toc210352379)

[2.1.1. Sd 15](#_Toc210352380)

[2.1.2. Sdfsdfs 15](#_Toc210352381)

[2.1.3. Sdf 15](#_Toc210352382)

[2.2. Thiết kế hệ thống 15](#_Toc210352383)

[2.2.1. Sdasd 15](#_Toc210352384)

[2.2.2. Đá 15](#_Toc210352385)

[2.2.3. Sdf 15](#_Toc210352386)

[2.2.4. s 15](#_Toc210352387)

[2.3. Đặc tả chức năng 15](#_Toc210352388)

[UC #001: Đăng ký 15](#_Toc210352389)

[UC #002: Đăng nhập 18](#_Toc210352390)

[UC #003: Danh sách nhà hàng 21](#_Toc210352391)

[UC #004: Danh mục sản phẩm 23](#_Toc210352392)

[UC #005 : Giỏ hàng 25](#_Toc210352393)

[UC #006: Thanh toán 29](#_Toc210352394)

[UC #007: Trạng thái đơn 35](#_Toc210352395)

[UC #008: Đánh giá 38](#_Toc210352396)

[3. Phát triển các chức năng mới 42](#_Toc210352397)

[4. Cài đặt 42](#_Toc210352398)

[5. Kết luận 42](#_Toc210352399)

[CHƯƠNG 1: INTRODUCTION 43](#_Toc210352400)

[CHƯƠNG 2: BUSSINESS CONTEXT 44](#_Toc210352401)

[CHƯƠNG 3: CONCEPTUAL MODEL 46](#_Toc210352402)

[CHƯƠNG 4: USE CASES 47](#_Toc210352403)

[1. Usecase: ProductCatalog 47](#_Toc210352404)

[2. Usecase: StoreCatalog 47](#_Toc210352405)

[3. Usecase: OrderCart 48](#_Toc210352406)

[4. Usecase: PaymentProcess 48](#_Toc210352407)

[5. Usecase: Ratinng 49](#_Toc210352408)

[6. Usecase: AccessControl 49](#_Toc210352409)

[CHƯƠNG 5: EVENT STORMING: EVENTS 50](#_Toc210352410)

[1. Product Catalog (Danh mục sản phẩm)​ 50](#_Toc210352411)

[2. Store Catalog (Danh mục cửa hàng)​ 50](#_Toc210352412)

[3. Order Cart (Giỏ hàng)​ 50](#_Toc210352413)

[4. Payment Process (Thanh toán)​ 51](#_Toc210352414)

[5. Rating (Đánh giá)​ 51](#_Toc210352415)

[6. Access Control (Kiểm soát truy cập)​ 51](#_Toc210352416)

[CHƯƠNG 6: USER STORIES 52](#_Toc210352417)

[1. Với Danh mục sản phẩm (Product Catalog): 52](#_Toc210352418)

[2. Với Danh mục cửa hàng(Store Catalog): 53](#_Toc210352419)

[3. Đặt hàng (Order Cart) 53](#_Toc210352420)

[4. Với Quy trình thanh toán (Payment Process): 54](#_Toc210352421)

[5. Với Đánh giá (Rating): 55](#_Toc210352422)

[6. Với Kiểm soát truy cập (Access Control): 56](#_Toc210352423)

LỜI MỞ ĐẦU

Trong bối cảnh thương mại điện tử và dịch vụ đặt hàng trực tuyến phát triển mạnh mẽ, việc xây dựng một nền tảng đặt hàng hiện đại, thân thiện với người dùng trở thành yếu tố then chốt giúp doanh nghiệp nâng cao năng lực cạnh tranh. Việc sở hữu kênh bán hàng trực tuyến không chỉ mang lại sự tiện lợi cho khách hàng, mà còn mở ra cơ hội mở rộng thị trường và tối ưu quy trình vận hành cho doanh nghiệp.

Dự án “**Phát triển giao diện đặt hàng trực tuyến trên nền tảng Web bằng React.js**” được khởi tạo với mục tiêu xây dựng một hệ thống đặt hàng trực quan, dễ sử dụng, và vận hành mượt mà trên cả máy tính và thiết bị di động. Giao diện này sẽ trở thành cầu nối quan trọng giữa khách hàng và doanh nghiệp, giúp nâng cao trải nghiệm mua sắm, gia tăng tính cạnh tranh, đồng thời thúc đẩy hoạt động kinh doanh trực tuyến của công ty **FuDa Food** phát triển bền vững.

**Tầm nhìn dài hạn**, hệ thống không chỉ dừng lại ở kênh web responsive mà còn có thể được mở rộng thành ứng dụng di động native để gia tăng độ tiện lợi, tích hợp thêm nhiều phương thức thanh toán điện tử, kết nối dịch vụ giao hàng tự động, và triển khai công cụ phân tích dữ liệu hành vi khách hàng. Điều này sẽ tạo tiền đề cho FuDa Food hình thành một hệ sinh thái kinh doanh trực tuyến toàn diện, phục vụ đa dạng nhu cầu của khách hàng trong tương lai.

1. TỔNG QUAN DỰ ÁN
2. Mô tả dự án

Dự án tập trung phát triển **giao diện đặt hàng trực tuyến trên nền tảng Web**, sử dụng **React.js** làm công nghệ chính. Giao diện được thiết kế theo hướng **responsive**, đảm bảo hiển thị tối ưu trên nhiều loại thiết bị, từ máy tính để bàn (desktop, laptop) đến thiết bị di động (smartphone, tablet).

Thay vì mỗi nhà hàng phải tự xây dựng website riêng, họ có thể **đăng ký tài khoản trên nền tảng**, **đăng tải menu món ăn**, cập nhật giá cả, hình ảnh và thông tin chi tiết.

Tập trung xử lý **UI/UX đặt hàng** gồm:

* Danh sách nhà hàng.
* Menu từng nhà hàng.
* Giỏ hàng (Cart).
* Trang thanh toán.
* Xem đơn hàng (Order tracking).

Khách hàng (buyer) khi truy cập sẽ:

* Xem **danh sách nhà hàng** theo khu vực, loại món ăn, hoặc gợi ý nổi bật.
* Chọn một nhà hàng → **xem menu chi tiết** (danh sách món, giá, hình ảnh).
* Đặt hàng trực tiếp trên giao diện nền tảng mà không cần chuyển sang web riêng của nhà hàng.

Điểm nhấn của dự án là **tập trung vào xử lý giao diện đặt hàng** (Frontend):

* Tối ưu trải nghiệm người dùng.
* Tạo các màn hình trực quan (danh sách nhà hàng, menu, giỏ hàng, đơn hàng).
* Tích hợp cơ bản với API (mock API hoặc backend tích hợp sau).

Mục tiêu là mang lại trải nghiệm mua sắm trực tuyến **nhanh chóng, tiện lợi và nhất quán**, đồng thời giảm tải cho quy trình đặt hàng thủ công truyền thống.

1. Hiện trạng khách hàng
2. Giới thiệu về công ty FuTa Food

Công ty FuDa Food là một doanh nghiệp hoạt động trong lĩnh vực phân phối thực phẩm và dịch vụ ăn uống. Với định hướng mở rộng kênh bán hàng trực tuyến, công ty mong muốn phát triển một nền tảng web trung gian cho phép nhiều nhà hàng cùng tham gia, qua đó:

* Tạo kênh phân phối mới cho nhà hàng.
* Thu hút khách hàng thông qua trải nghiệm đặt hàng tiện lợi.
* Gia tăng doanh thu từ phí dịch vụ hoặc hoa hồng giao dịch.

1. Quy trình nghiệp vụ hiện tại

Hiện nay, khách hàng muốn đặt món từ các nhà hàng phải:

* Gọi điện thoại hoặc đặt qua Zalo/Facebook → thủ công, tốn thời gian.
* Một số nhà hàng có website/app riêng → nhưng khách hàng phải tải/đăng nhập từng app → bất tiện.
* Thiếu một **nền tảng chung** để khách hàng chỉ cần vào **một nơi duy nhất** và có thể chọn từ nhiều nhà hàng.

**Vấn đề gặp phải:**

* Khó tiếp cận khách hàng mới.
* Nhà hàng nhỏ không đủ nguồn lực xây web/app riêng.
* Trải nghiệm khách hàng không đồng nhất.

1. Yêu cầu nghiệp vụ
2. Actor & Yêu cầu chức năng

Customer (Buyer):

* Đăng ký
* Đăng nhập
* Danh sách nhà hàng (lọc theo khu vực, món ăn).
* Danh mục sản phẩm (tên món, giá, ảnh, mô tả).
* Giỏ hàng (Thêm sản phẩm vào giỏ hàng, cập nhật/xóa sản phẩm).
* Thanh toán (xem tổng tiền, nhập thông tin giao hàng, phương thức thanh toán, xác nhận đơn hàng).
* Trạng thái đơn hàng: pending, delivering, completed (mock API).
* Đánh giá món ăn / nhà hàng.
* Đăng xuất (mock API).

Merchant (Nhà hàng):

* Đăng nhập (mock API).
* Quản lý đơn hàng (đơn mới (xác nhận tay, auto), đã nhận, lịch sử).
* Quản lý menu (bao gồm đăng tải, chỉnh sửa, xóa, cập nhật(còn hàng, hết hàng)).

Admin

* Quản lý tài khoản người dùng (Merchant, Buyer).
* Quản lý danh sách nhà hàng (phê duyệt, bật/tắt).
* Quản lý review (ẩn/xóa nếu vi phạm).(kiểm duyệt nội dung)
* Theo dõi hoạt động đơn hàng toàn hệ thống.

1. Yêu cầu phi chức năng

* **Hiệu năng (Performance)**
* Load sản phẩm nhanh (< 2s).
* Giỏ hàng xử lý tức thì khi thêm/xóa sản phẩm.
* **Tương thích (Compatibility)**
* Web: chạy tốt trên các trình duyệt phổ biến (Chrome, Safari, Edge, Firefox).
* Mobile: chạy tốt trên iOS và Android.
* **Trải nghiệm người dùng (UX/UI)**
* Giao diện thân thiện, dễ thao tác, responsive (Web + Mobile).
* Validation form rõ ràng (ví dụ nhập sai số điện thoại phải báo lỗi).
* **Bảo mật (Security)**
* Mã hóa thông tin nhạy cảm khi gửi qua API (HTTPS).
* Bảo vệ token đăng nhập (JWT/Session).
* **Khả năng mở rộng (Scalability)**
* Code phải dễ maintain, dễ thêm tính năng mới (sử dụng kiến trúc component-based, state management chuẩn).
* Tách component rõ ràng.
* Dùng state management (Redux hoặc Context API)
* **Tích hợp API giả lập (mock API)**
* Dữ liệu lưu trong file **Data.js** hoặc **db.json.**
* API mock bằng **json-server** hoặc **hard-code JSON**.
* **Khả năng chịu tải (Reliability & Availability)**
* Hệ thống FE không bị crash khi có nhiều sản phẩm/đơn hàng.

1. Giải pháp lựa chọn

* Công nghệ sử dụng
* HTML5: xây dựng cấu trúc trang web.
* CSS3: định dạng giao diện, hỗ trợ responsive cơ bản.
* Bootstrap: framework CSS để tạo bố cục responsive nhanh chóng, sẵn có nhiều component UI (navbar, button, form...).
* React.js: framework chính để phát triển giao diện đặt hàng, hỗ trợ component-based, state management.
* Mock API (Data.js hoặc json-server): giả lập dữ liệu nhà hàng, menu, giỏ hàng, đơn hàng.
* Axios / Fetch API: để gọi dữ liệu từ mock API.
* Điểm nhấn
* Giao diện Web responsive, chạy mượt trên desktop, laptop, tablet, smartphone.
* Chỉ tập trung vào Frontend (FE), không làm Backend thật.
* Ưu tiên trải nghiệm đặt hàng trực quan (xem danh sách nhà hàng → menu → thêm giỏ hàng → thanh toán).

1. Kế hoạch dự án
2. Phạm vi và mục tiêu
3. Phạm vi

Dự án xây dựng giao diện đặt hàng trực tuyến (Frontend), gồm 6 phân hệ chính:

* **Phân hệ danh sách nhà hàng:** hiển thị danh sách nhà hàng theo khu vực, loại món ăn, gợi ý nổi bật.
* **Phân hệ menu & đặt món:** xem menu từng nhà hàng, chi tiết sản phẩm (tên, ảnh, giá, mô tả), thêm/xóa sản phẩm vào giỏ hàng.
* **Phân hệ giỏ hàng & thanh toán:** xem giỏ hàng, tính tổng tiền (giả lập qua mock API/Data.js), xác nhận đơn hàng.
* **Phân hệ quản lý đơn hàng (buyer):** cho phép khách hàng xem lại đơn hàng đã đặt.
* **Phân hệ quản lý menu (Seller):** quản lý sản phẩm, giá cả, trạng thái.
* **Phân hệ quản lý nền tảng (Admin):** quản lý user, nhà hàng, review, thống kê.

**Lưu ý:** Dự án chỉ tập trung vào Frontend (giao diện & xử lý mock API), chưa triển khai Backend thật và chưa có Seller/Admin.

1. Mục tiêu

|  |  |  |
| --- | --- | --- |
| **Mục tiêu** | **Đơn vị** | **Cam kết** |
| Ngày bắt đầu | 21/09/2025 | Xác định yêu cầu & khởi tạo dự án |
| Ngày kết thúc | 30/12/2025 | Hoàn thiện FE đặt hàng chạy demo trên web responsive |
| Thời gian thực hiện |  |  |

1. Cơ sở hạ tầng
2. Môi trường phát triển

|  |  |  |
| --- | --- | --- |
| **Công việc/Sản phẩm** | **Mục đích** | **Chú thích** |
| Hệ điều hành | Windows 10 / macOS | Cho client và developer |
| Visual Studio Code | IDE chính để code React.js |  |
| Node.js + npm | Môi trường chạy React.js |  |
| React.js | Framework frontend chính |  |
| Bootstrap 5 | Responsive UI framework |  |
| CSS3/HTML5 | Thiết kế giao diện |  |
| Mock API (Data.js / json-server) | Giả lập dữ liệu menu, giỏ hàng, đơn hàng |  |
| GitHub/GitLab | Quản lý source code |  |

1. Phần cứng & phần mềm

|  |  |
| --- | --- |
| **Công việc/Sản phẩm** | **Mục đích** |
| Laptop/PC cấu hình tối thiểu i5, MacOS, RAM 8GB | Máy phát triển FE |
| Trình duyệt Chrome/Firefox/Safari/Edge | Test giao diện |
| Figma | Thiết kế prototype UI/UX |
| Postman | Test mock API |

1. Lịch sử thực hiện

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên giai đoạn** | **Công việc chính** | **Bắt đầu** | **Kết thúc** | **Độ dài (ngày)** | **Người thực hiện** |
| Khởi tạo dự án | Xác định phạm vi, mục tiêu, yêu cầu FE | 16/10 | 20/10 | 4 |  |
| Phân tích yêu cầu | Xác định use case (Buyer), mock API cần thiết | 20/10 | 05/10 | 15 |  |
| Thiết kế UI/UX | Thiết kế prototype (Figma), luồng đặt hàng | 06/10 | 20/10 | 14 |  |
| Thiết kế kiến trúc FE | Tạo cấu trúc React.js (component, route, state) | 21/10 | 21/11 | 7 |  |
| Phát triển FE | Code từng màn hình: Home, Restaurant List, Menu, Cart, Checkout, Orders | ... | ... | ... |  |
| Tích hợp mock API | Kết nối Data.js/json-server với FE |  |  |  |  |
| Kiểm thử nội bộ | Test chức năng FE (giỏ hàng, thanh toán giả lập) |  |  |  |  |
| Hoàn thiện & báo cáo | Chạy demo, viết tài liệu hướng dẫn |  |  |  |  |

1. Quản lý cấu hình
2. Định nghĩa CI và baseline
3. Thư mục dự án

/node-modules

/public

/src/assets

1. Tái thiết kế - Chuẩn hoá quy trình
2. Chuẩn hoá quy trình
3. Sd
4. Sdfsdfs
5. Sdf
6. Thiết kế hệ thống
7. Sdasd
8. Đá
9. Sdf
10. s
11. Đặc tả chức năng

UC #001: Đăng ký

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Use Case Number:** | UC01 | | |
| **Use Case Name:** | Đăng ký. | | |
| **Actor (s):** | Customer. | | |
| **Maturity:** | Focused. | | |
| **Summary:** | Customer tạo tài khoản mới để sử dụng dịch vụ đặt món. | | |
| **Basic Course of Events:** | **Actor Action** | | **System Response** |
| 1. Customer mở web và chọn “Đăng ký”. | |  |
|  | | 2. Hệ thống hiển thị form đăng ký với các trường bắt buộc: (Họ tên, Số điện thoại (bắt đầu là số 0 và 10 ký tự số), Email, Giới tính, Ngày sinh, Mật khẩu, Xác nhận mật khẩu). |
| 3. Customer nhập thông tin đăng ký. **A1, E1, E2, E3.** | |  |
| 4. Customer nhấn “Tiếp tục”. | |  |
|  | | 5. Hệ thống kiểm tra (Email/SĐT chưa tồn tại). **E4** |
|  | | 6. Hệ thống gửi mã OTP (One Time Password) đến SĐT/Email của actor. |
| 7. Customer nhập OTP. **A2** | |  |
|  | | 8. Hệ thống xác thực OTP. **E5** |
|  | | 9. Hệ thống hiển thị thông báo “Đăng ký thành công” và chuyển actor đến màn hình đăng nhập. |
|  | | 10. Usecase kết thúc tại đây |
| **Alternative Path:** | **A1.** Customer hủy đăng ký. | | |
| **Actor Action** | **System Response** | |
| **A1.1.** Customer hủy bỏ đăng ký |  | |
|  | **A1.2.** Hệ thống quay về trang chủ | |
| **A2.** Customer yêu cầu gửi lại OTP. | | |
| **Actor Action** | **System Response** | |
| **A2.1.** Customer nhấn “Gửi lại OTP”. |  | |
|  | **A2.2.** Hệ thống gửi mã OTP mới | |
|  | **A2.3.** Hệ thống quay lại bước 6 của Basic Course Of Event | |
| **Exception Path:** | **E1. Thông tin bị bỏ trống → thông báo “Vui lòng điền đầy đủ thông tin” và yêu cầu nhập.**  **E2.** Số điện thoại/Email không đúng định dạng → thông báo “Số điện thoại/Email không hợp lệ” và yêu cầu nhập lại.  **E3.** Mật khẩu không hợp lệ (quá ngắn, thiếu ký tự đặc biệt…) → thông báo lỗi “Mật khẩu không hợp lệ”.  **E4.** Số điện thoại/Email đã tồn tại → thông báo “Tài khoản đã được đăng ký” và yêu cầu nhập lại thông tin khác.  **E5.** OTP không đúng hoặc hết hạn → thông báo “OTP không hợp lệ, vui lòng thử lại”. | | |
| **Extension Points:** | UC02 – Đăng nhập (sau khi đăng ký xong, actor có thể đăng nhập) | | |
| **Triggers:** | Actor muốn sử dụng Web FuDaFood và chưa có tài khoản. | | |
| **Assumptions:** | Customer có thiết bị nhận OTP (SMS/Email).  Hệ thống Web FuDaFood hoạt động bình thường và kết nối server OTP. | | |
| **Preconditions:** | Customer chưa có tài khoản FuDaFood. | | |
| **Post Conditions:** | Customer có tài khoản mới trên FuDaFood.  Customer có thể đăng nhập và sử dụng các tính năng đặt món. | | |
| **Reference: Business Rules** | Một số điện thoại/Email chỉ được đăng ký một tài khoản.  Mật khẩu tối thiểu 8 ký tự, bao gồm chữ và số.  OTP chỉ hợp lệ trong 2 phút kể từ khi gửi. | | |
| **Author (s):** | Bảo | | |
| **Date:** | Điền sau | | |
| **Activity Diagram:** |  | | |
| **Sequence Diagram:** |  | | |
| **Boudary:** | | | |

UC #002: Đăng nhập

|  |  |  |
| --- | --- | --- |
|  | | |
| **Use Case Number:** | UC#002 | |
| **Use Case Name:** | Đăng nhập. | |
| **Actor (s):** | Customer, Merchant. | |
| **Maturity:** | Focused. | |
| **Summary:** | Actor đăng nhập vào hệ thống để thực hiện các chức năng phù hợp với vai trò của mình. | |
| **Basic Course of Events:** | **Actor Action** | **System Response** |
| 1. Actor nhập Email và mật khẩu. |  |
|  | 2. Hệ thống kiểm tra thông tin đăng nhập. **E1, E2.** |
|  | 3. Hệ thống chuyển hướng actor đến trang chủ theo vai trò. |
| **Alternative Path:** | None | |
| **Exception Path:** | **E1.** Nếu Email không có trong cơ sở dữ liệu:   * Hệ thống thông báo lỗi “Email không tồn tại” và đưa actor trở lại màn hình đăng nhập.   **E2.** Nếu mật khẩu không khớp:   * Hệ thống thông báo lỗi “Mật khẩu không chính xác” và đưa actor trở lại màn hình đăng nhập. * Actor có thể thử lại đăng nhập mà không bị giới hạn số lần. | |
| **Extension Points:** | None | |
| **Triggers:** | Actor truy cập vào hệ thống mà không có phiên làm việc (session).  Phiên làm việc của actor đã hết hạn hoặc actor đã đăng xuất | |
| **Assumptions:** | Actor đã có Email và mật khẩu hợp lệ để đăng nhập vào hệ thống.  Hệ thống đang hoạt động bình thường và có thể truy cập cơ sở dữ liệu để xác thực thông tin.  Actor có quyền truy cập vào hệ thống dựa trên vai trò của mình.  Các thông tin đăng nhập đã được bảo mật và không bị tiết lộ cho người khác.  Không có sự cố kỹ thuật hoặc lỗi mạng trong quá trình đăng nhập. | |
| **Preconditions:** | Actor chưa có phiên làm việc và đang ở trang Đăng nhập. | |
| **Post Conditions:** | Actor có một phiên làm việc đang hoạt động và được chuyển đến trang chủ với các tính năng phù hợp. | |
| **Reference: Business Rules** | None | |
| **Author (s):** | XXXXXX | |
| **Date:** | Điền sau | |
| **Giao diện:** | | |

UC #003: Danh sách nhà hàng

|  |  |  |
| --- | --- | --- |
|  | | |
| **Use Case Number:** | UC#003 | |
| **Use Case Name:** | Danh sách nhà hàng. | |
| **Actor (s):** | Customer. | |
| **Maturity:** | Focused. | |
| **Summary:** | Use case này mô tả quy trình Customer xem danh sách nhà hàng, bao gồm hiển thị gợi ý, lọc, tìm theo vị trí, và chọn nhà hàng để xem chi tiết | |
| **Basic Course of Events:** | **Actor Action** | **System Response** |
| 1. Customer truy cập web. |  |
|  | 2. Hệ thống hiển thị danh sách nhà hàng gợi ý ban đầu (một số nổi bật, có ưu đãi). |
| 3. Customer chọn một nhà hàng trong danh sách nhà hàng.**A1** |  |
|  | 4. Hệ thống hiển thị chi tiết nhà hàng (menu, ưu đãi, thông tin giao hàng).Từ bước này chuyển sang **UC#004** |
|  | 5. Use case kết thúc. |
| **Alternative Path:** | **A1. Customer chọn bộ lọc.** | |
| **Actor Action** | **System Response** |
| A1.1 Customer chọn bộ lọc theo vị trí Gần tôi (Quận/Huyện) hoặc tùy chọn Sort option (Tất cả, mới nhất, cũ nhất). |  |
|  | A1.2 Hệ thống hiển thị danh sách nhà hàng theo bộ lọc đã chọn. **E1** |
| A1.3 Customer chọn nhà hàng trong danh sách đã lọc. |  |
|  | A1.4 Hệ thống quay lại bước 4 của Basic |
| **Exception Path:** | E1. Nếu không có nhà hàng gần tôi → Hệ thống báo “Không tìm thấy nhà hàng gần vị trí của bạn” và gợi ý nhập địa chỉ khác. Quay lại A1.1 của Alternative | |
| **Extension Points:** | None | |
| **Triggers:** | Customer truy cập ứng dụng/web FuDaFood và yêu cầu xem danh sách nhà hàng. | |
| **Assumptions:** | Customer có thiết bị kết nối Internet.  Hệ thống đã có dữ liệu nhà hàng trong CSDL. | |
| **Preconditions:** | Customer đã truy cập được vào hệ thống (không nhất thiết phải đăng nhập).  Hệ thống sẵn sàng hiển thị dữ liệu nhà hàng. | |
| **Post Conditions:** | Hệ thống hiển thị thành công chi tiết nhà hàng mà Customer chọn.  Nếu Customer chọn lọc, danh sách được cập nhật theo bộ lọc. | |
| **Reference: Business Rules** | None | |
| **Author (s):** | XXXXXX | |
| **Date:** | Điền sau | |
| **Activity Diagram:** | | |
| **Sequence Diagram:** | | |
| **Giao diện:** | | |

UC #004: Danh mục sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Use Case Number:** | UC#004 | | |
| **Use Case Name:** | Danh mục sản phẩm | | |
| **Actor (s):** | Customer. | | |
| **Maturity:** | Focused. | | |
| **Summary:** | Use case này mô tả quy trình Customer xem danh mục sản phẩm của một nhà hàng, bao gồm: xem thông tin chi tiết nhà hàng, tìm kiếm/lọc món ăn, và thêm sản phẩm vào giỏ hàng. | | |
| **Basic Course of Events:** | **Actor Action** | | **System Response** |
| 1. Customer chọn nhà hàng từ danh sách từ **(UC#003).** | |  |
|  | | 2. Hệ thống hiển thị thông tin chi tiết nhà hàng (tên, địa chỉ, ưu đãi, phí giao hàng) và danh sách toàn bộ món ăn. **E1** |
| 3. Customer có thể xem tất cả thức ăn từ danh sách tất cả món và có thể thêm vào giỏ hàng từ catalog hoặc chi tiết sản phẩm . **A1** | |  |
|  | | 4. Hệ thống chuyển sang UC giỏ hàng **(UC#005).** |
|  | | 5. Use case kết thúc. |
| **Alternative Path:** | A1. Chọn xem chi tiết sản phẩm | | |
| **Actor Action** | **System Response** | |
| A1.1 Customer nhấn vào sản phẩm. |  | |
|  | A1.2 Hệ thống hiển thị chi tiết sản phẩm. | |
| A1.3 Customer có thể xem và thêm món “Thêm vào giỏ hàng”. |  | |
|  | A1.4 Hệ thống quay lại bước **4 của Basic** | |
| **Exception Path:** | E1. Nếu dữ liệu món ăn bị lỗi (không tồn tại/không còn bán) → Hệ thống thông báo: “Món ăn hiện không khả dụng”. | | |
| **Extension Points:** | Liên kết với **UC#005 – Giỏ hàng** khi Customer thêm sản phẩm. | | |
| **Triggers:** | Customer chọn một nhà hàng từ danh sách để xem danh mục sản phẩm. | | |
| **Assumptions:** | Nhà hàng đã cập nhật menu đầy đủ và chính xác.  Customer có kết nối Internet ổn định. | | |
| **Preconditions:** | Customer đã chọn một nhà hàng (từ UC#003). | | |
| **Post Conditions:** | Customer đã xem danh mục món ăn của nhà hàng.  Nếu thêm sản phẩm → giỏ hàng của Customer được cập nhật. | | |
| **Reference: Business Rules** | Có thể vừa tìm kiếm vừa lọc cùng lúc. | | |
| **Author (s):** | XXXXXX | | |
| **Date:** | Điền sau | | |
| **Activity Diagram:** | | | |
| **Sequence Diagram:** | | | |
| **Giao diện:** | | | |

UC #005 : Giỏ hàng

|  |  |  |
| --- | --- | --- |
|  | | |
| **Use Case Number:** | UC #005 | |
| **Use Case Name:** | Giỏ hàng | |
| **Actor (s):** | Customer | |
| **Maturity:** | Focused | |
| **Summary:** | Use case này mô tả toàn bộ quy trình quản lý giỏ hàng của Customer, bao gồm: tạo giỏ, thêm sản phẩm (từ Catalog hoặc trang chi tiết), lựa chọn topping/tùy chọn thêm, xem giỏ, cập nhật/xóa sản phẩm, và tiến hành thanh toán. | |
| **Basic Course of Events:** | **Actor Action** | **System Response** |
| 1. Buyer nhấp vào sản phẩm trên trang danh mục sản phẩm. **A1** |  |
|  | 2. Hệ thống hiển thị thông tin chi tiết sản phẩm. |
| 3. Buyer lựa chọn số lượng và “Thêm vào giỏ hàng”, từ trang chi tiết sản phẩm. |  |
|  | 4. Hệ thống kiểm tra đủ tồn kho. **E1** |
|  | 5. Hệ thống hiển thị màn hình chọn topping/tùy chọn thêm (size, sốt, topping...). |
| 6. Buyer có thể lựa chọn topping và xác nhận. **E2** |  |
|  | 7. Hệ thống tạo giỏ hàng số lượng = số lượng mà Buyer đã chọn ở bước 3 hoặc mặc định = 1 nếu không thay đổi hoặc mặc định = 1 khi thêm từ catalog. |
| 8. Buyer truy cập giỏ hàng. |  |
|  | 9. Hệ thống hiển thị danh sách sản phẩm trong giỏ và bảng thông tin tóm tắt (tổng chi phí, tiết kiệm khuyến mãi, phí vận chuyển, tiết kiệm phí vận chuyển, tổng số tiền). |
| 10. Buyer cập nhật số lượng sản phẩm trong giỏ(tăng, giảm sản phẩm). **A2** |  |
|  | 11. Hệ thống cập nhật giỏ và làm mới bảng thông tin tóm tắt. |
| 12. Buyer tiến hành checkout. |  |
|  | 13. Hệ thống kiểm tra giỏ hàng không rỗng và bắt đầu quy trình thanh toán. **E3** |
|  | 14. UseCase kết thúc tại đây. |
| **Alternative Path:** | **A1. Buyer thêm sản phẩm từ Catalog** | |
| **Actor Action** | **System Response** |
| **A1.1.** Buyer chọn “Thêm vào giỏ hàng” từ trang Catalog. |  |
|  | **A1.2.** Hệ thống quay lại **Bước 4** của **Basic Course of Events** |
| **A2. Buyer xóa một sản phẩm khỏi giỏ.** | |
| **Actor Action** | **System Response** |
| **A2.1.** Buyer chọn “Xóa” một sản phẩm trong giỏ. |  |
|  | **A2.2.** Hệ thống quay lại bước 9 của **Basic Course of Events.** |
| **Exception Path:** | **E1.** Buyer thêm sản phẩm “vượt quá số lượng tồn kho” → Hệ thống hiển thị thông báo: “Lỗi 404.”  **E2.** Buyer không chọn topping bắt buộc → Hệ thống thông báo “Vui lòng chọn topping/tùy chọn đầy đủ”.  **E3.** Checkout khi giỏ rỗng → Hệ thống báo “Giỏ hàng rỗng, hãy thêm sản phẩm”. | |
| **Extension Points:** | Liên kết với UC #004 – Danh mục sản phẩm (để đồng bộ tình trạng còn hàng/hết hàng).  Liên kết với UC #006 – Thanh toán (chi tiết quá trình checkout sau khi xác nhận giỏ hàng). | |
| **Triggers:** | Buyer muốn mua sản phẩm và tiến hành đặt hàng. | |
| **Assumptions:** | Buyer đã đăng nhập vào hệ thống.  Buyer có kết nối Internet ổn định trong suốt quá trình thao tác.  Hệ thống hoạt động bình thường và kết nối được với server backend.  Sản phẩm, topping và khuyến mãi còn hiệu lực trong catalog. | |
| **Preconditions:** | Buyer có tài khoản và đăng nhập thành công.  Danh mục sản phẩm hiển thị đúng dữ liệu. | |
| **Post Conditions:** | Nếu thành công: Giỏ hàng của Buyer được tạo/cập nhật với các sản phẩm và topping đã chọn. Buyer có thể tiến hành thanh toán.  Nếu thất bại (E1, E2): Giỏ hàng không được cập nhật hoặc checkout không thực hiện, hệ thống yêu cầu thao tác lại. | |
| **Reference: Business Rules** | Một sản phẩm có thể được thêm nhiều lần vào giỏ hàng (tuỳ số lượng).  Mỗi sản phẩm khi thêm vào giỏ phải kèm topping bắt buộc (nếu có).  Không cho phép thêm vào giỏ hàng sản phẩm đã hết hàng hoặc ngừng kinh doanh.  Giỏ hàng phải có ít nhất một sản phẩm thì mới được thanh toán.  Hệ thống tính toán tổng tiền = (giá sản phẩm + topping) – khuyến mãi + phí vận chuyển. | |
| **Author (s):** | XXXXXX | |
| **Date:** | Điền sau | |
| **Activity Diagram:** | | |
| **Sequence Diagram:** | | |
| **Giao diện:** | | |

UC #006: Thanh toán

|  |  |  |
| --- | --- | --- |
|  | | |
| **Use Case Number:** |  | |
| **Use Case Name:** | Thanh toán | |
| **Actor (s):** | Customer | |
| **Maturity:** | focused | |
| **Summary:** | Use case này mô tả quy trình thanh toán đơn hàng sau khi Customer đã xác nhận giỏ hàng, bao gồm nhập thông tin giao hàng, lựa chọn phương thức thanh toán, xác nhận đơn hàng và xử lý kết quả thanh toán. | |
| **Basic Course of Events:** | **Actor Action** | **System Response** |
| 1. Customer nhấn nút “Giao hàng” từ giỏ hàng. |  |
|  | 2. Hệ thống kiểm tra giỏ hàng không rỗng (liên kết từ UC #005). |
|  | 3. Hệ thống hiển thị form thông tin thanh toán: (địa chỉ, tên, số điện thoại, giờ dự kiến giao, thông tin sản phẩm. Đồng thời tính lại giá cộng phí giao hàng, áp dụng mã giảm giá (nếu có) và hiển thị tổng số tiền phải trả, phương thức thanh toán). |
| 4. Customer nhập thông tin giao hàng (họ tên, số điện thoại, địa chỉ, ghi chú). **A1** |  |
| 5. Customer chọn phương thức thanh toán COD (tiền mặt). **A2** |  |
|  | 6. Hệ thống hiển thị thông tin giao hàng đã nhập. |
| 7. Customer nhấn nút “Đặt đơn” từ form xác nhận đơn hàng. |  |
|  | 8. Hệ thống kiểm tra tính hợp lệ của thông tin giao hàng. **E1** |
|  | 9. Hệ thống hiển thị pop-up “Xác nhận đặt đơn”. |
| 10. Customer chọn “Đặt ngay”. **A3** |  |
|  | 11. Hệ thống hiển thị form “Đã đặt đơn” |
|  | 12. Hệ thống hiển thị trang hàng đang được chuẩn bị và chờ shiper nhận hàng. |
|  | 13. Usecase kết thúc tại đây. |
| **Alternative Paths:** | **A1. Dùng lại thông tin đã có** | |
| **Actor Action** | **System Response** |
| **A1.1** Customer sử dụng lại thông tin giao hàng đã có sẵn hoặc hệ thống lấy local. |  |
|  | A1.1. Hệ thống mặc định địa chỉ cũ để giao hàng. |
|  | A1.2. Quay lại bước 5 của basic |
| **A2. Thanh toán online** | |
| **Actor Action** | **System Response** |
| A2.1. Customer chọn phương thức thanh toán Online. |  |
|  | A2.2. Hệ thống hiển thị các tùy chọn Online (Momo, Ngân hàng, Ví điện tử). |
| A2.3. Customer chọn một phương thức Online. |  |
|  | A2.4. Hệ thống hiển thị đầy đủ thông tin giao hàng + phương thức đã chọn = Online). |
| A2.5. Customer nhấn “Đặt đơn”. |  |
|  | A2.6. Hệ thống kiểm tra tính hợp lệ của thông tin giao hàng (**E1**). |
|  | A2.7. Hệ thống hiển thị màn hình nhập thông tin thẻ. (số thẻ, tên chủ thẻ, ngày phát hành, mã xác nhận). Nếu Customer đã có thẻ lưu sẵn thì bỏ qua bước này. **A4** |
| A2.8. Customer nhập thông tin thẻ và nhấn “Xác nhận”. **A5** |  |
|  | A2.9. Hệ thống kiểm tra tính hợp lệ của thông tin thẻ. **E2** |
|  | A2.10. Hệ thống hiển thị pop-up “Xác nhận đặt đơn”. |
| A2.11. Customer chọn “Đặt ngay”. **A3** |  |
|  | A2.12. Hệ thống gửi request sang cổng thanh toán. |
| A2.13. Customer xác nhận trên giao diện cổng thanh toán (OTP/Mã PIN/Ứng dụng liên kết). |  |
|  | A2.14. Hệ thống xác nhận thành công → Hệ thống quay lại bước 11 của Basic. **E3** |
| **A3. Huỷ xác nhận** | |
| **Actor Action** | **System Response** |
| A3.1. Customer huỷ xác nhận đặt đơn. |  |
|  | A3.2. Hệ thống quay lại bước 6 của Basic |
| **A4. Đã có thông tin ngân hàng** | |
|  | A4.1. Hệ thống hiển thị thông tin thẻ đã lưu. |
| A4.2. Customer xác nhận sử dụng thẻ đã lưu. |  |
|  | A4.3. Hệ thống chuyển đến bước A2.12. |
| **A5. Thoát nhập thông tin thẻ** | |
| **Actor Action** | **System Response** |
| A5.1. Customer nhấn thoát |  |
|  | A5.2. Hệ thống hiện Pop-up xác nhận thoát hay không |
| A5.3. Customer nhấn nút “Đồng ý” **A6** |  |
|  | A5.4. Hệ thống huỷ đơn hàng và chuyển tới trang đã huỷ. |
| **A6. Nhấn “Huỷ” đồng ý thoát** | |
| **Actor Action** | **System Response** |
| A6.1. Customer nhấn “Huỷ” |  |
|  | A6.2. Hệ thống trở lại bước A2.9 của alternative. |
| **Exception Paths:** | E1: Thông tin giao hàng không hợp lệ → báo lỗi. Quay lại bước 3 của basic  E2: Thông tin ngân hàng không hợp lệ / thất bại → báo lỗi. Quay lại bước A2.7 của Alternative  E3: Thanh toán online thất bại (timeout, hủy giao dịch). Quay lại bước 6 của Basic | |
| **Extension Points:** |  | |
| **Triggers:** | “Khách hàng nhấn ‘Đặt hàng/Thanh toán’ từ giỏ hàng để bắt đầu quy trình.” | |
| **Assumptions:** | Hệ thống tính phí ship và mã giảm sẵn sàng.  Kết nối mạng ổn định.  Nếu Online Payment: cổng thanh toán hoạt động, callback thành công. | |
| **Preconditions:** | Giỏ hàng không rỗng và chỉ thuộc một nhà hàng.  Nhà hàng đang mở.  Khách hàng đã nhập địa chỉ giao hàng. | |
| **Post Conditions:** | Order được tạo (status = PLACED).   * Payment: * COD → PENDING, sẽ cập nhật CAPTURED khi giao hàng. * Online → AUTHORIZED nếu thanh toán thành công. * Nếu thanh toán Online thất bại (FAILED) → Order không được tạo, trạng thái = CANCELLED. * Khách hàng được chuyển sang trang theo dõi đơn hàng. | |
| **Reference: Business Rules** | * Mặc định PTTT = COD, khách có thể đổi sang Online trước -khi xác nhận. * Giỏ hàng chỉ thuộc một nhà hàng. * Không dùng giá từ client, hệ thống luôn tính lại từ DB. * Phí ship = base\_fee + per\_km \* ceil(distance\_km). * Voucher chỉ hợp lệ khi trong thời gian và điều kiện áp dụng. * Online Payment cần callback từ cổng thanh toán để xác nhận kết quả. * Đặt hàng” chỉ khả dụng khi hệ thống đã xác nhận rằng Customer có thông tin giao hàng hợp lệ (từ nhập mới hoặc lấy từ local/profile). * Nếu chưa có thông tin hợp lệ → hệ thống hiển thị cảnh báo: “Vui lòng nhập hoặc chọn thông tin giao hàng trước khi đặt hàng. | |
| **Author(s):** |  | |
| **Date:** |  | |
| **Activity Diagram:** | | |
| **Sequence Diagram:** | | |
| **Giao diện:** | | |

UC #007: Trạng thái đơn

|  |  |  |
| --- | --- | --- |
|  | | |
| **Use Case Number:** | UC#007 | |
| **Use Case Name:** | Trạng thái đơn hàng | |
| **Actor (s):** | Customer. | |
| **Maturity:** | Focused. | |
| **Summary:** | Use case này mô tả quá trình khách hàng theo dõi trạng thái các đơn hàng của mình, bao gồm các đơn Đang đến, Đã mua, và Đơn nháp, Rating, xem chi tiết đơn hàng, tiến trình giao hàng, và thao tác đặt lại hoặc huỷ đơn, đánh giá. | |
| **Basic Course of Events:** | **Actor Action** | **System Response** |
| 1. Customer truy cập trang “Đơn hàng của tôi”. |  |
|  | 2. Hệ thống hiển thị các trạng thái: Đang giao, Đã giao, Đã huỷ, Rating. |
| 3. Customer chọn tab “Đang giao”. **A1, A2.** |  |
|  | 4. Hệ thống hiển thị danh sách các đơn đang được giao đến. **E1** |
| 5. Customer nhấn vào nút “Theo dõi đơn hàng”. |  |
|  | 6. Hệ thống hiển thị chi tiết đơn hàng, bao gồm: sản phẩm, số lượng, địa chỉ giao hàng, thời gian đặt hàng, trạng thái hiện tại và tiến trình giao hàng còn lại. |
|  | 7. Hệ thống hiển thị timeline trạng thái: Đang xử lý → Shipper nhận → Đang giao → Đã giao. |
|  | 8. Usecase kết thúc tại đây. |
| **Alternative Path:** | **A1. Chuyển sang tab “Đã giao”** | |
| **Actor Action** | **System Response** |
| A1.1 Customer nhấn tab “Đã giao”. **A2, basic 2.** |  |
|  | A1.2 Hệ thống hiển thị danh sách các đơn đã mua và nút “Đặt lại” và “Rating”. **E2** |
| A1.3 Customer xem các đơn đã đặt. **A3** |  |
|  | A1.4 Usecase kết thúc tại đây |
| **A2. Chuyển sang tab “Đã huỷ”** | |
| **Actor Action** | **System Response** |
| A2.1 Customer nhấn tab “Đã huỷ”. **A1, basic 2** |  |
|  | A2.2 Hệ thống hiển thị danh sách các đã huỷ. **E3** |
| A2.4 Customer xem các đơn đã huỷ. **A3** |  |
|  | A2.5 Usecase kết thúc tại đây |
| **A3. Đặt lại đơn hàng** | |
| A3.1 Customer chọn “Đặt lại đơn hàng” |  |
|  | A3.2 Hệ thống đặt lại đơn hàng đã chọn và mặt định là 1 sản phẩm |
|  | A3.3 Hệ thống quay lại giỏ hàng để tiếp tục thanh toán |
|  | A3.4 Usecase kết thúc tại đây |
| **Exception Path:** | E1. Nếu không có đơn hàng nào → thông báo “Chưa có đơn hàng đang giao” và chuyển về catalog để mua hàng  E2. Nếu không có đơn hàng nào → thông báo “Chưa có đơn hàng đã mua” và chuyển về catalog để mua hàng  E3. Nếu không có đơn hàng nào → thông báo “Chưa có đơn hàng đã huỷ” và chuyển về catalog để mua hàng | |
| **Extension Points:** | None | |
| **Triggers:** | Customer truy cập trang ‘Đơn hàng của tôi | |
| **Assumptions:** | Hệ thống có khả năng lưu trạng thái đơn hàng”, “Kết nối mạng ổn định | |
| **Preconditions:** | Customer đã đăng nhập”, “Có ít nhất 1 đơn hàng trong hệ thống | |
| **Post Conditions:** | Customer đã xem trạng thái đơn hàng”, “Các thao tác đặt lại hoặc xóa đơn nháp được thực hiện thành công | |
| **Reference: Business Rules** | None | |
| **Author (s):** | XXXXXX | |
| **Date:** | Điền sau | |
| **Activity Diagram:** | | |
| **Sequence Diagram:** | | |
| **Giao diện:** | | |

UC #008: Đánh giá

|  |  |  |
| --- | --- | --- |
|  | | |
| **Use Case Number:** | UC#008 | |
| **Use Case Name:** | Đánh giá sản phẩm (Rating) | |
| **Actor (s):** | Customer. | |
| **Maturity:** | Focused. | |
| **Summary:** | Use case này mô tả quá trình khách hàng đánh giá sản phẩm sau khi nhận hàng, bao gồm nhập số sao, bình luận, xem lại đánh giá và chỉnh sửa/xoá đánh giá. | |
| **Basic Course of Events:** | **Actor Action** | **System Response** |
| 1. Customer truy cập trang “Đơn hàng của tôi” |  |
|  | 2. Hệ thống hiển thị “Trạng thái đơn hàng” trong đó có Tab “Đánh giá” |
| 3. Customer chọn Tab “Đánh giá” |  |
|  | 4. Hệ thống hiển thị danh sách các đơn đã đánh giá và chưa đánh giá.(có bộ lọc: all, đã rate (fix, delete) , chưa rate (rating)). **E1** |
| 5. Customer chọn đơn muốn đánh giá. Nếu đơn chưa đánh giá. **A1, A2** |  |
|  | 6. Hệ thống hiển thị form đánh giá (số sao, bình luận). |
| 7. Customer nhập số sao (1–5) và nội dung đánh giá |  |
|  | 8. Hệ thống hiển thị bản xem trước đánh giá và nút “Gửi đánh giá”. |
| 9. Customer nhấn “**Gửi đánh giá**”. |  |
|  | 10. Hệ thống lưu đánh giá vào cơ sở dữ liệu, cập nhật điểm trung bình sản phẩm, thông báo “Đánh giá thành công”.**E2** |
|  | 11. Hệ thống hiển thị đánh giá đã gửi cùng với các đánh giá khác của người dùng khác. |
|  | 12. UseCase kết thúc tại đây. |
| **Alternative Path:** | **A1. Chỉnh sửa đánh giá**(nếu đơn đã đánh giá trước đó) | |
| **Actor Action** | **System Response** |
| A1.1 Customer chọn đơn cần chỉnh sửa mà đã gửi và nhấn “Chỉnh sửa” |  |
|  | A1.2 Hệ thống hiển thị form chỉnh sửa (số sao, bình luận). |
| A1.3 Customer chỉnh sửa và nhấn “Cập nhật” |  |
|  | A1.4 Hệ thống cập nhật đánh giá và thông báo thành công. |
|  | A1.5. Hệ thống quay lại bước 11 của Basic |
| **A2. Xoá đánh giá** (nếu đơn đã đánh giá trước đó) | |
| **Actor Action** | **System Response** |
| A2.1 Customer chọn đánh giá cần xoá mà đã gửi và nhấn “Xoá” |  |
|  | A2.2 Hệ thống hiển thị pop-up xác nhận xoá. |
| A2.3 Customer xác nhận xoá |  |
|  | A2.4 Hệ thống xoá đánh giá và cập nhật điểm trung bình sản phẩm. |
|  | A2.5 Hệ thống quay lại bước 11 của basic |
| **Exception Path:** | E1: Không có đơn nào để đánh giá → hiển thị “Chưa có đơn để đánh giá”.  E2: Lỗi kết nối / lưu đánh giá thất bại → thông báo “Đánh giá không thành công, vui lòng thử lại”. | |
| **Extension Points:** | None | |
| **Triggers:** |  | |
| **Assumptions:** |  | |
| **Preconditions:** | Customer đã nhận hàng (đơn trạng thái “Đã giao”).  Customer đã đăng nhập. | |
| **Post Conditions:** | Đánh giá được lưu vào DB.  Điểm trung bình sản phẩm được cập nhật.  Giao diện hiển thị đánh giá mới nhất. | |
| **Reference: Business Rules** | Số sao: 1–5.  Bình luận: tối đa 500 ký tự.  Mỗi đơn chỉ đánh giá 1 lần / sản phẩm. | |
| **Author (s):** | XXXXXX | |
| **Date:** | Điền sau | |
| **Activity Diagram:** | | |
| **Sequence Diagram** | | |
| **Giao diện:** | | |

UC #009: Quản lý đơn hàng

Bxzcvx

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Use Case Number:** | | UC #009 | | |
| **Use Case Name:** | | Quản lý đơn hàng | | |
| **Actor (s):** | | Merchant | | |
| **Maturity:** | |  | | |
| **Summary:** | | Use case này mô tả quy trình quản lý đơn hàng, bao gồm theo dõi trạng thái đơn hàng, chỉnh sửa đơn hàng (do hết món hoặc sai giá), và hủy đơn hàng trong các trường hợp khác nhau. | | |
| **Basic Course of Events:** | | **Actor Action** | | **System Response** |
| 1. Merchant đăng nhập vào web Partner. | |  |
|  | | 2. Hệ thống hiển thị các chức năng |
| 3. Merchant chọn chức năng “Quản lý đơn hàng” | |  |
|  | | 4. Hệ thống hiển thị Tab (Mới, Đã xác nhận, Lịch sử, Đã hủy). |
| 5. Merchant chọn Tab “Mới” (đơn chưa xác nhận). | |  |
|  | | 6. Hệ thống hiển thị nút **“Bật/tắt”** Auto xác nhận đơn hàng là ở trạng thái mặt định là **Tắt** và danh sách các đơn hàng ở trạng thái chưa xác nhận và các nút “Xác nhận đơn”, “Huỷ đơn”, “Chỉnh sửa”. **A1** |
| 7. Merchant chọn “Xác nhận đơn”.  **(A2). “Huỷ đơn” hoặc chọn (A3). “Chỉnh sửa đơn”** | |  |
|  | | 8. Hệ thống tìm tài xế và chuyển đơn hàng vào trạng thái đang chuẩn bị. |
| 9. Merchant chuyển sang Tab “Đã xác nhận” (đơn Merchant đã xác nhận nhưng chưa bàn giao cho tài xế). | |  |
|  | | 10. Hệ thống hiển thị danh sách đơn đã xác nhận. |
| 11. Merchant nhấn vào 1 đơn. | |  |
|  | | 12. Hệ thống hiển thị thông tin đơn hàng đã xác nhận và có 3 nút “Thông báo cho tài xế” và “Huỷ đơn” và  “Chỉnh sửa đơn”. |
| 13. Merchant nhấn “Thông báo cho tài xế” để tài xế đến lấy đơn. **A2, A3.** | |  |
|  | | 14. Hệ thống xác nhận hoàn thành đơn và chuyển đơn sang Tab “Lịch sử”. |
| 15. Merchant chuyển sang Tab “Lịch sử” (đơn đã hoàn tất). | |  |
|  | | 16. Hệ thống hiển thị danh sách các đơn đã đặt theo ngày tháng năm. |
| 17. Merchant chuyển sang Tab “Đã huỷ” (đơn bị hủy bất kỳ lý do nào). | |  |
|  | | 18. Hệ thống hiển thị danh sách tất cả món đã bị huỷ |
|  | | 19. Usecase kết thúc tại đây. |
| **A1. Bật Auto xác nhận đơn** | | |
| A1.1 Merchant nhấn nút “Bật” auto xác nhận đơn | |  |
|  | | A1.2 Hệ thống tự động xác nhận tất cả các đơn. |
|  | | A1.3 Hệ thống chuyển sang 8 của Basic |
| **A2. Nhấn “Huỷ đơn”** | | |
| **Actor Action** | | **System Response** |
| **A2.1.** Merchant nhấn nút “Huỷ đơn” | |  |
|  | | **A2.2.** Hệ thống hiển thị pop-up thông báo lý do huỷ đơn Radio (Hết hàng, Quán quá tải, Quán đóng cửa, Khác). |
| **A2.3**. Merchant chọn lý do huỷ. **E1.** | |  |
| **A2.4** Merchant nhấn “Xác nhận huỷ”. **A4** | |  |
|  | | **A2.5** Hệ thống huỷ đơn hàng và thông báo “Đơn đã huỷ” và **Quay lại bước 6 của Basic khi chưa xác nhận đơn và Quay lại bước 10 của Basic khi xác nhận đơn rồi.** |
| **A3. Nhấn “Chỉnh sửa đơn” (Hết món/ Đổi món/ Hết topping)** | | |
| **A3.1** Merchant “Nhấn chỉnh đơn”. | |  |
|  | | **A3.2** Hệ thống hiển thị thông tin cần chỉnh ( Đổi món, thêm topping, thêm đá, ...) |
| **A3.3** Merchant chọn thông tin cần chỉnh. **E2** | |  |
| **A3.4** Merchant nhấnxác nhận. | |  |
|  | | **A3.5** Hệ thống xác nhận thông tin chỉnh sửa và quay lại bước **6 của Basic khi chưa xác nhận đơn và bước 10 của Basic khi xác nhận đơn rồi.** |
| **A4. Nhấn “Không huỷ”** | | |
| **A4.1** Merchant nhấn “Không huỷ” | |  |
|  | | **A4.2** Hệ thống **quay lại bước 6 của Basic khi chưa xác nhận đơn và quay lại bước 12 của Basic khi xác nhận đơn rồi.** |
| **Exception Path:** | | E1: Merchant chọn “Huỷ đơn” nhưng không nhập lý do → hệ thống báo lỗi.  E2: Merchant chọn “Chỉnh sửa đơn” nhưng không lưu thay đổi → hệ thống giữ nguyên đơn. | |
| **Extension Points:** | | Có thể mở rộng tích hợp thông báo SMS/Push cho tài xế | | |
| **Triggers:** | | Merchant đăng nhập và chọn tính năng quản lý đơn hàng | | |
| **Assumptions:** | | Merchant có quyền truy cập vào hệ thống web Partner, kết nối mạng ổn định | | |
| **Preconditions:** | Merchant đã đăng nhập, hệ thống có dữ liệu đơn hàng. | | |
| **Post Conditions:** | | Đơn hàng được cập nhật chính xác theo thao tác (xác nhận / chỉnh sửa / hủy / hoàn tất). | | |
| **Reference: Business Rules** | | Đơn hàng chỉ có thể chỉnh sửa khi chưa bàn giao cho tài xế.  Nếu Auto Xác nhận bật, tất cả đơn hàng mới sẽ được chuyển sang trạng thái “Đang chuẩn bị” mà không cần Merchant thao tác thủ công.  Lý do hủy đơn hàng phải bắt buộc chọn (không để trống).  Một đơn hàng chỉ có thể “Thông báo cho tài xế” sau khi đã được Merchant xác nhận.  BR5Đơn ở trạng thái “Lịch sử” hoặc “Đã huỷ” sẽ không thể chỉnh sửa thêm. | | |
| **Author (s):** | | XXXXXX | | |
| **Date:** | | Điền sau | | |
| **Activity Diagram:** | | | | |
| **Sequence Diagram:** | | | | |
| **Giao diện:** | | | | |

UC #010: Quản lý menu

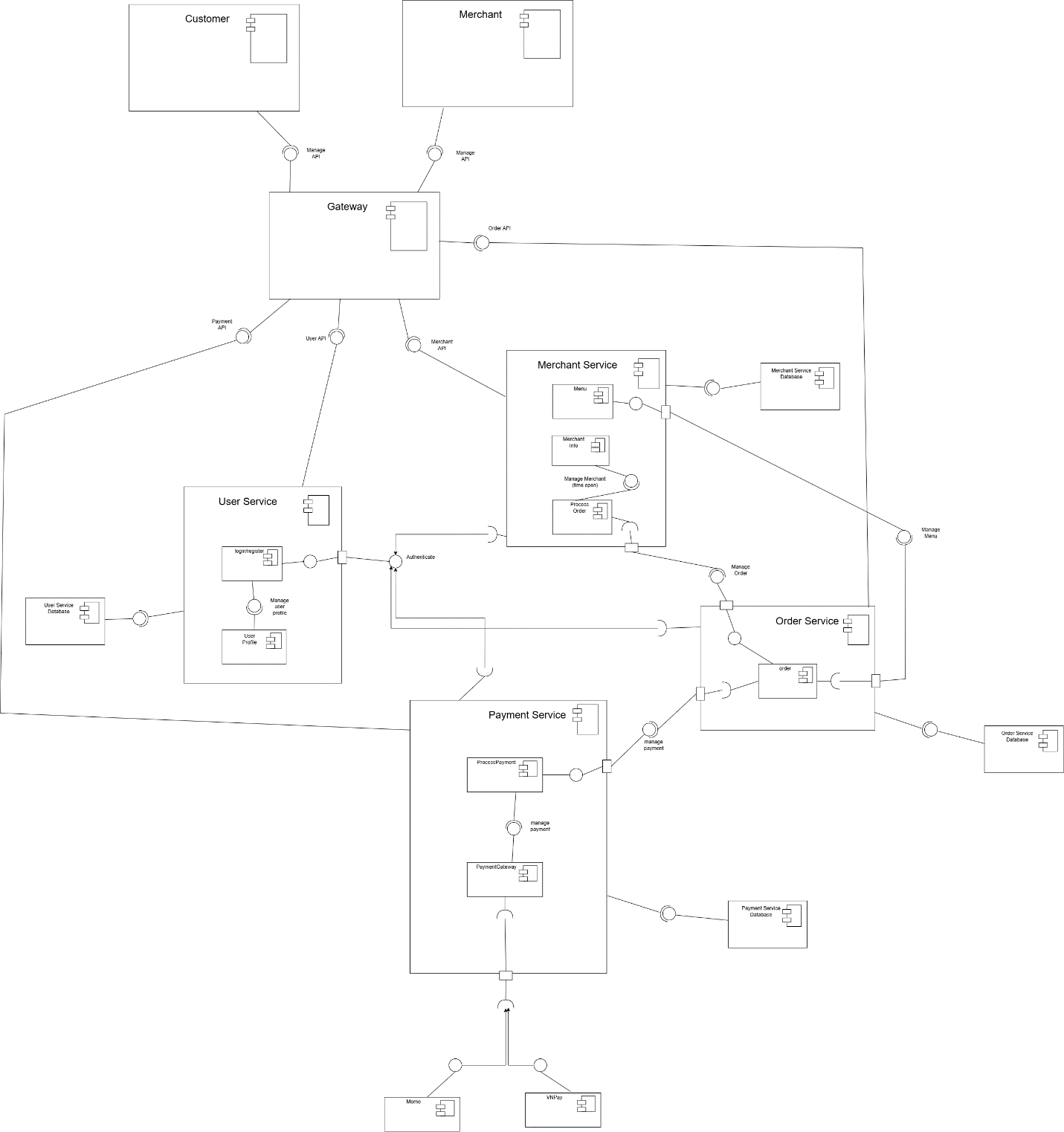
|  |  |  |
| --- | --- | --- |
| [**https://youtu.be/g2V1c2IkZi0?si=Sd1gJcxuFlDhB8zm**](https://youtu.be/g2V1c2IkZi0?si=Sd1gJcxuFlDhB8zm) | | |
| **Use Case Number:** | UC#010 | |
| **Use Case Name:** | Quản lý menu. | |
| **Actor (s):** | Merchant. | |
| **Maturity:** | Focused. | |
| **Summary:** | Use case này mô tả quy trình Merchant quản lý menu của nhà hàng trên web Partner, bao gồm xem danh sách món ăn, thêm sản phẩm, chỉnh sửa, bật/tắt trạng thái món ăn, và xóa món. Ngoài ra Merchant có thể quản lý nhóm topping.  Use case bao gồm 2 nhánh chính: (1) Quản lý món ăn, (2) Quản lý topping. | |
| **Basic Course of Events:** | **Actor Action** | **System Response** |
| 1. Merchant đăng nhập vào web Partner. |  |
|  | 2. Hệ thống hiển thị giao diện chào mừng và các chức năng. |
| 3. Merchant chọn Tab “Thực đơn” |  |
|  | 4. Hệ thống hiển thị 2 Tabs “Món” và “Nhóm topping”. |
| 5. Merchant chọn Tab “Món”. **A1** (nếu chọn Tab “Nhóm topping” → xem Alternative A1). |  |
|  | 6. Hệ thống hiển thị danh sách các món đã có và có thao tác “Thêm sản phẩm” và có thể “Bật/tắt” và “Xoá” sản phẩm. |
| Merchant nhấn “Thêm sản phẩm” |  |
|  | Hệ thống hiển thị form “Thêm món ăn mới” (Nhập tên món, Giá món, Nhập danh mục món, mô tả Hình ảnh, nguyên liệu, chất gây dị ứng, calo, Protein, carb, fat... ) |
|  |  |
| 7. Merchant nhấn vào sản phẩm để có thể chỉnh sửa. **A2 bật tắt, A3 xoá.** |  |
|  | 8. Hệ thống hiển thị Form để chỉnh sửa bao gồm: Hình ảnh, tên, giá, danh mục, mô tả. |
| 9. Merchant nhấn xác nhận. **A4 huỷ** |  |
|  | 10. Hệ thống update lên csdl. |
|  | 11. Use case kết thúc. |
| **Alternative Path:** | **A1. Merchant chọn Tab “Nhóm topping”** | |
| **Actor Action** | **System Response** |
| A1.1 Merchant chọn tab “Nhóm topping”. |  |
|  | A1.2 Hệ thống hiển thị danh sách nhóm topping hiện tại và thao tác “Thêm topping”. |
| A1.3 Merchant nhấn “Thêm topping”. |  |
|  | A1.4 Hệ thống hiển thị form nhập gồm có “Tên nhóm” “Món thêm” và “Quyền tuỳ chọn”, “Món liên kết”. **(VD: tên nhóm: tương 🡪 Món thêm: tương cà , tương ớt...)** |
| A1.5 Merchant nhập “Tên nhóm” |  |
| A1.6 Merchant chọn “Món thêm” có (Tên và giá) bắt buộc nhập |  |
|  | A1.7 Hệ thống hiển thị form “Món thêm” gồm có tên và giá. |
| A1.8 Merchant nhập thông tin. |  |
| A1.9 Merchant nhấn “hoàn tất”. **A5** |  |
|  | A1.10 Hệ thống kiểm tra dữ liệu, quay lại giao diện chính và bắt buộc sang bước Quyền tùy chọn. **E1** |
|  | A1.11 Hệ thống chuyển sang thao tác tiếp theo và bắt buộc chọn “quyền tuỳ chọn” |
| A1.12 Merchant nhấn “Quyền tuỳ chọn” |  |
|  | A1.13 Hệ thống hiển thị form chứa “Radio” lựa chọn : “Bắt buộc chọn” hoặc “Không bắt buộc”. |
| A1.14 Merchant chọn 1 trong 2 |  |
| A1.15 Merchant nhấn “Hoàn tất” **A5** |  |
|  | A1.16 Hệ thống trở về giao diện ban đầu. |
|  | A1.17 Hệ thống chuyển sang thao tác tiếp theo và bắt buộc chọn “liên kết món” |
| A1.18 Merchant nhấn “liên kết món” |  |
|  | A.1.19 Hệ thống hiển thị trang liên kết món và danh sách các món cần liên kết với topping. |
| A1.20 Merchant chọn các món ăn đã có bên menu cần liên kết với topping. |  |
| A1.21 Merchant nhấn “Xác nhận” A5 |  |
|  | A1.22 Hệ thống hiển thị lại thông tin topping đã thêm |
|  | A.1.23 Hệ thống cho phép lưu. |
| A1.24 Merchant nhấn “Lưu” **A6** |  |
|  | A1.25 Hệ thống quay lại bước 10 của Basic |
| **A2. Bật/Tắt món ăn** | |
| A2.1 Merchant nhấn “Bật/Tắt”. |  |
|  | A2.2 Hệ thống cập nhật trạng thái hiển thị món ăn cho Customer. |
|  |  |
| A3. Xóa món ăn | |
|  |  |
| A3.1 Merchant nhấn “Xóa”. |  |
|  | A3.2 Hệ thống hiển thị cảnh báo xác nhận. |
| A3.3 Nếu Merchant đồng ý → hệ thống xóa khỏi menu. |  |
|  | A.3.4 Hệ thống quay lại bước 10 của Basic |
| **A4. Hủy chỉnh sửa** | |
| A4.1 Merchant nhấn “Hủy chỉnh sửa”. |  |
|  | A4.2 Hệ thống không thay đổi dữ liệu, quay lại bước 6 của Basic. |
| A5. Chọn thoát | |
| A5.1 Merchant nhấn nút “Thoát” |  |
|  | A5.2 Hệ thống quay về bước A1.4 |
| **A6. Chọn Không lưu** |  |
| A6.1 Merchant nhấn nút “Không lưu” |  |
|  | A6.2 Hệ thống chuyển về bước A1.22 của Alternative |
|  |  |
| **Exception Path:** | E1: Thêm topping nhưng thiếu tên hoặc giá → báo lỗi và quay lại bước A1.7 và yêu cầu nhập lại.  E2: Liên kết topping với món nhưng không chọn món nào → hệ thống báo “Bạn phải chọn ít nhất 1 món”. Và quay lại bước A1.19  E3: Lỗi kết nối / server → hiển thị “Không thể lưu. Vui lòng thử lại sau”. | |
| **Extension Points:** | None | |
| **Triggers:** | Merchant muốn quản lý menu (thêm, sửa, xóa món hoặc topping).  Merchant đăng nhập và truy cập tab ‘Thực đơn’ để quản lý món hoặc topping | |
| **Assumptions:** | Merchant đã có nhà hàng được phê duyệt và có quyền quản lý menu. | |
| **Preconditions:** | Merchant đã đăng nhập vào web Partner. | |
| **Post Conditions:** | Nếu thao tác thành công → menu/topping được cập nhật trong CSDL, hiển thị ngay cho Customer.  Nếu thao tác bị hủy/thoát → dữ liệu không thay đổi.  Nếu lỗi hệ thống (E3) → dữ liệu không được cập nhật và Merchant nhận thông báo lỗi để thử lại sau. | |
| **Reference: Business Rules** | Món ăn bắt buộc có Tên, Giá (>0), Danh mục; tên không trùng trong cùng nhà hàng.  Món ăn có trạng thái Bật/Tắt; xóa thì ẩn khỏi menu nhưng lưu log.  Nhóm topping bắt buộc có Tên nhóm, ít nhất 1 topping (Tên + Giá).  Merchant phải chọn “Quyền tùy chọn” và liên kết topping với ≥1 món.  Chỉ Merchant có quyền mới được thêm/sửa/xóa; mọi thao tác đều ghi log.  Lỗi hệ thống/kết nối → không cập nhật, phải hiển thị thông báo lỗi. | |
| **Author (s):** | XXXXXX | |
| **Date:** | Điền sau | |
| **Activity Diagram:** | | |
| **Sequence Diagram:** |  | |
| **Giao diện:** | | |

UC #011: Quản lý tài khoản người dùng (Merchant, Customer).

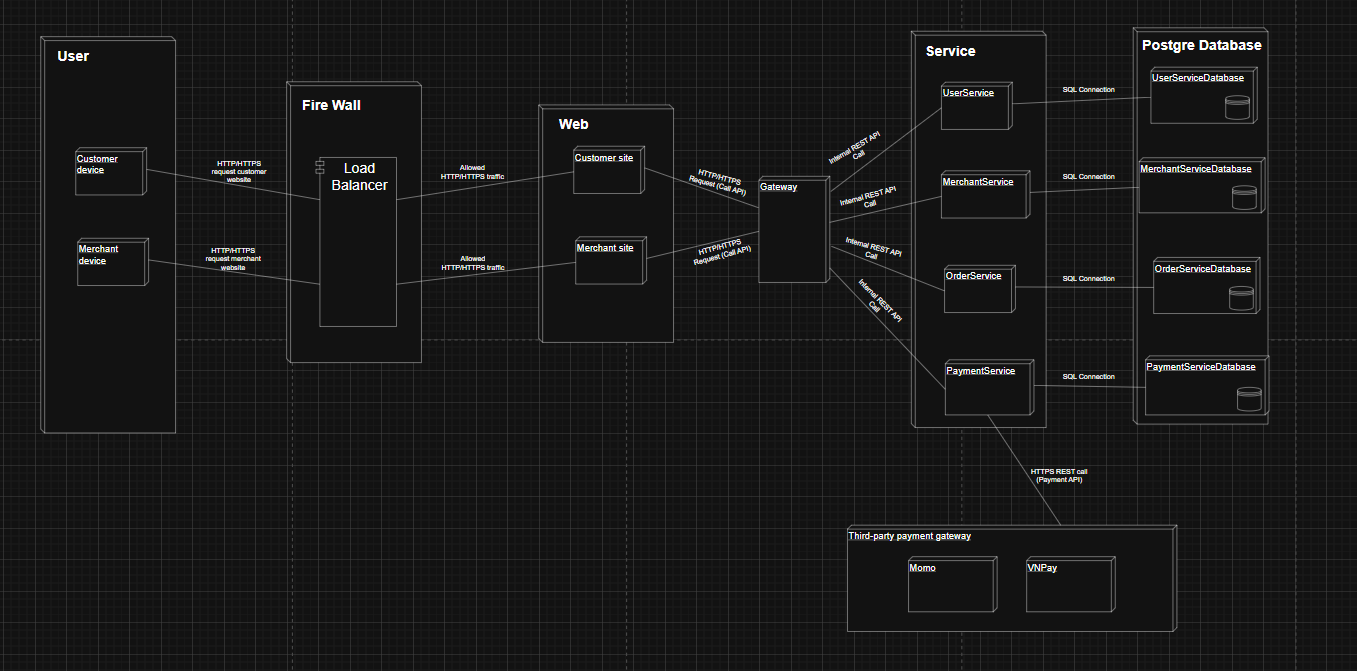
UC #012: Quản lý danh sách nhà hàng (phê duyệt, bật/tắt).

UC #013: Quản lý review (ẩn/xóa nếu vi phạm).(kiểm duyệt nội dung)

Sơ đồ component



Sơ đồ Deployment



1. Phát triển các chức năng mới
2. Cài đặt
3. Kết luận

CHƯƠNG 1: INTRODUCTION

**Trang web FuDa Food** là một ứng dụng dịch vụ vi mô đa ngôn ngữ, bao gồm nhiều **dịch vụ nhỏ (microservices)** được xây dựng bằng **React** và **Node.js**. Nó trình bày cách kết nối các dịch vụ vi mô riêng lẻ để xây dựng một ứng dụng lớn hơn theo **nguyên tắc kiến trúc microservices**.

Trang web này chủ yếu được thiết kế cho **hệ sinh thái JavaScript/TypeScript**, tận dụng các thư viện và công cụ phổ biến của cộng đồng **React – Node.js**. Đồng thời, nó cũng thử nghiệm các thành phần và thư viện mới để xây dựng **ứng dụng hiện đại, gốc đám mây (cloud-native)**.

CHƯƠNG 2: BUSSINESS CONTEXT

**PDBT** có kịch bản kinh doanh cơ bản cho **Danh mục thức ăn**(Product Catalog), **Đặt hàng** (Order Cart), **Quy trình thanh toán** (Payment Process), **Quản lý store** (Store manage), **Đánh giá** (Rating), **Kiểm soát truy cập** (Access Control).

**Với Danh mục sản phẩm (Product Catalog):**

* Người mua (Buyer) có thể duyệt danh sách thức ăn store với chức năng lọc và sắp xếp theo tên sản phẩm hoặc giá. Người mua có thể xem chi tiết món ăn bằng cách nhấp vào món ăn trong danh sách. Trong trang chi tiết món ăn, người mua có thể xem tên, mô tả, trạng thái (còn hàng hay không).
* Quản trị hệ thống (SysAdmin) có thể quản lý món ăn và có khả năng điều chỉnh trạng thái của món ăn (còn hàng hay không).

**Với Danh mục cửa hàng(Store Catalog):**

* Người mua (Buyer) có thể duyệt danh sách store với chức năng lọc và sắp xếp theo tên store hoặc vị trí gần nhất. Người mua có thể xem danh mục sản phẩm khi nhấn vào store và người mua có thể thấy được store có lượt rating cao được hiển thị trên store đó.
* Quản trị hệ thống (SysAdmin) có thể …???

**Đặt hàng (Order cart):**

* Người mua có thể mua bất kỳ món ăn nào trong danh sách bằng nút thêm vào giỏ hàng (Add food), hoặc cũng có thể mua ngay từ trang chi tiết món ăn. Sau khi mua, những món ăn đó sẽ hiển thị trong giỏ hàng cùng bảng tóm tắt thông tin, bao gồm: tổng chi phí giỏ hàng, số tiền tiết kiệm từ khuyến mãi sản phẩm, chi phí tạm tính, phí vận chuyển, số tiền tiết kiệm từ khuyến mãi vận chuyển, và tổng số tiền đơn hàng.
* Khi người mua thêm hoặc xóa thức ăn trong giỏ, thì giỏ hàng và bảng tóm tắt cũng sẽ được cập nhật. Sau cùng, người mua có thể tiến hành thanh toán bằng cách nhấn nút Thanh toán trên trang giỏ hàng.
* Quản trị cửa hàng có thể xem giỏ hàng của mình trên hệ thống.

để có thể bật/tắt những giỏ hàng không hợp lệ trong hệ thống CoolStore.

**Với Quy trình thanh toán (Payment Process):**

* Sau khi người mua nhấn nút Thanh toán, hệ thống sẽ bắt đầu xác thực thông tin món ăn, xử lý thanh toán theo nhiều phương thức có thể chọn (visa, momo, tiền mặt, …), và gửi email thông báo cho người mua để họ biết trạng thái giao dịch.

**Với Đánh giá (Rating):**

* Người mua có thể đánh giá bất kỳ store nào mà họ cho là tốt sau mỗi đơn hàng của mình (từ 1 đến 5 sao).

**Với Kiểm soát truy cập (Access Control):**

* Người mua hoặc Quản trị hệ thống có thể đăng nhập/đăng xuất khỏi hệ thống. Nếu là Buyer, họ sẽ được đưa đến trang danh mục sản phẩm; nếu là SysAdmin, họ sẽ được đưa đến trang quản trị.

**Một số tác vụ khởi tạo** cần được thực hiện khi Website **PDBT** bắt đầu hoạt động, bao gồm: tạo tài khoản SysAdmin, hai tài khoản Buyer và dữ liệu mẫu cho sản phẩm, kho hàng, cùng đánh giá cho một vài sản phẩm.

CHƯƠNG 3: CONCEPTUAL MODEL

A diagram of a company

Description automatically generated with medium confidence

CHƯƠNG 4: USE CASES

1. Usecase: ProductCatalog

A screenshot of a cell phone

Description automatically generated

1. Usecase: StoreCatalog

A screenshot of a chat

Description automatically generated

1. Usecase: OrderCart

A screenshot of a chat

Description automatically generated

1. Usecase: PaymentProcess

A group of white ovals with black text

Description automatically generated

1. Usecase: Ratinng

A screenshot of a black background with white ovals

Description automatically generated

1. Usecase: AccessControl

A black background with white ovals

Description automatically generated

CHƯƠNG 5: EVENT STORMING: EVENTS

1. Product Catalog (Danh mục sản phẩm)​

* ProductListDisplayed (Danh sách món ăn đã hiển thị với lọc/sắp xếp).​
* ProductDetailDisplayed (Chi tiết món ăn đã hiển thị).​
* ProductCreated (Món ăn đã được tạo).​
* ProductUpdated (Món ăn đã được cập nhật).​
* ProductDeleted (Món ăn đã bị xóa).​
* ProductAvailabilityUpdated (Trạng thái còn hàng/không đã được cập nhật).​

1. Store Catalog (Danh mục cửa hàng)​

* StoreListDisplayed (Danh sách cửa hàng đã hiển thị với lọc/sắp xếp).​
* StoreDetailDisplayed (Chi tiết cửa hàng đã hiển thị).​
* StoreRatedDisplayed (Cửa hàng có rating cao đã hiển thị nổi bật).​
* StoreCreated (Cửa hàng đã được tạo).​
* StoreUpdated (Cửa hàng đã được cập nhật).​
* StoreDeleted (Cửa hàng đã bị xóa).​

1. Order Cart (Giỏ hàng)​

* ProductAddedToCart (Món ăn đã được thêm vào giỏ).​
* ProductRemovedFromCart (Món ăn đã bị xóa khỏi giỏ).​
* ShoppingCartUpdated (Giỏ hàng đã được cập nhật – số lượng, giá, khuyến mãi).​
* ShoppingCartCheckedOut (Giỏ hàng đã được thanh toán).​
* InvalidShoppingCartDisabled (Giỏ hàng không hợp lệ đã bị vô hiệu hóa bởi SysAdmin).​

1. Payment Process (Thanh toán)​

* OrderValidated (Đơn hàng đã được xác thực).​
* PaymentProcessed (Thanh toán đã được xử lý).​
* EmailNotificationSent (Email thông báo đã được gửi).​

1. Rating (Đánh giá)​

* StoreRated (Cửa hàng đã được đánh giá 1–5 sao).​

1. Access Control (Kiểm soát truy cập)​

* UserLoggedIn (Người dùng đã đăng nhập).​
* UserLoggedOut (Người dùng đã đăng xuất).​
* SysAdminAccountCreated (Tài khoản SysAdmin đã được khởi tạo).​
* BuyerAccountCreated (Tài khoản Buyer đã được khởi tạo).​

CHƯƠNG 6: USER STORIES

1. Với Danh mục sản phẩm (Product Catalog):

**Buyer (Người mua)**

* **Xem danh sách món ăn**
* Là một Người mua, tôi muốn duyệt toàn bộ danh sách món ăn có trên hệ thống, để có cái nhìn tổng quan và lựa chọn món ăn mình quan tâm.
* Lọc danh sách món ăn
* Là một Người mua, tôi muốn lọc danh sách món ăn theo tên hoặc theo khoảng giá, để nhanh chóng tìm được món ăn phù hợp với nhu cầu.
* Sắp xếp danh sách món ăn
* Là một Người mua, tôi muốn sắp xếp món ăn theo tên (A–Z/Z–A) hoặc giá (tăng/giảm), để dễ dàng so sánh và đưa ra quyết định chọn món.
* Xem chi tiết món ăn
* Là một Người mua, tôi muốn xem thông tin chi tiết của một món ăn, bao gồm tên, mô tả, giá và trạng thái (còn hàng/hết hàng), để đảm bảo có đầy đủ thông tin trước khi đặt món.

**SysAdmin (Quản trị hệ thống)**

* Quản lý món ăn (CRUD)
* Là một Quản trị viên hệ thống, tôi muốn có khả năng thêm mới, chỉnh sửa hoặc xóa món ăn trong danh mục, để quản lý và cập nhật thực đơn của hệ thống một cách hiệu quả.
* Điều chỉnh trạng thái món ăn
* Là một Quản trị viên hệ thống, tôi muốn thay đổi trạng thái của một món ăn (còn hàng hay đã hết hàng), để đảm bảo người mua chỉ nhìn thấy và đặt được những món ăn khả dụng.

1. Với Danh mục cửa hàng(Store Catalog):

* Là Người mua (Buyer), tôi muốn xem danh sách cửa hàng, có thể lọc và sắp xếp theo:
* Tên cửa hàng, khu vực, trạng thái hoạt động (enable/disable), đánh giá trung bình, và cờ “Hot Store”.
* Khi lọc theo khu vực hoặc trạng thái, danh sách cửa hàng phải thu hẹp đúng với điều kiện đã chọn.
* Khi sắp xếp theo tên hoặc đánh giá (tăng dần/giảm dần), danh sách phải hiển thị đúng thứ tự.
* Khi kết hợp cả lọc và sắp xếp, danh sách phải áp dụng đồng thời cả hai điều kiện.
* Là Người mua (Buyer), tôi muốn có thể truy cập vào trang chi tiết cửa hàng với các thông tin:
* Tên, địa chỉ, giờ mở cửa, danh sách sản phẩm, trạng thái hoạt động, cờ “Hot Store” (nếu có).

1. Đặt hàng (Order Cart)

* Là một Người mua, tôi muốn mua bất kỳ món ăn nào trong danh sách trên trang danh mục, bằng cách nhấn Thêm vào giỏ hàng (Add food), để món ăn đó được thêm vào giỏ hàng theo mặc định.
* Là một Người mua, tôi muốn xem chi tiết món ăn và mua món ăn này nếu tôi thích, để món ăn đó được thêm vào giỏ hàng theo mặc định.
* Là một Người mua, tôi muốn xem danh sách các món ăn tôi vừa thêm vào giỏ hàng và bảng thông tin tóm tắt cho giỏ hàng hiện tại, bao gồm: tổng chi phí giỏ hàng, số tiền tiết kiệm từ khuyến mãi sản phẩm, chi phí tạm tính, phí vận chuyển, số tiền tiết kiệm từ khuyến mãi vận chuyển, và tổng số tiền đơn hàng.
* Là một Người mua, tôi muốn cập nhật số lượng món ăn trong giỏ hàng.
* Bất cứ khi nào số lượng món ăn thay đổi, bảng thông tin tóm tắt cũng cần được cập nhật tương ứng.
* Là một Người mua, tôi muốn xóa bất kỳ món ăn nào trong giỏ hàng mà tôi không muốn mua nữa.
* Bất cứ khi nào món ăn bị xóa, bảng thông tin tóm tắt cũng cần được cập nhật tương ứng.
* Là một Người mua, tôi muốn kiểm tra giỏ hàng của mình.
* Bất cứ khi nào giỏ hàng trống (số lượng món ăn bằng 0), quá trình kiểm tra này sẽ không xảy ra.
* Khi giỏ hàng được kiểm tra, quá trình thanh toán sẽ bắt đầu.
* Là một SysAdmin, tôi muốn xem giỏ hàng của tất cả Người mua với đầy đủ thông tin: tổng chi phí giỏ hàng, số tiền tiết kiệm từ khuyến mãi sản phẩm, chi phí tạm tính, phí vận chuyển, số tiền tiết kiệm từ khuyến mãi vận chuyển, và tổng số tiền đơn hàng.
* Là một SysAdmin, tôi muốn bật/tắt bất kỳ giỏ hàng nào của bất kỳ Người mua nào.

1. Với Quy trình thanh toán (Payment Process):

**Buyer (Người mua)**

* Tiến hành thanh toán giỏ hàng
* Là một Người mua, tôi muốn nhấn nút Thanh toán sau khi đã chọn món ăn trong giỏ hàng, để hoàn tất đơn hàng của mình. Bên cạnh đó, tôi muốn lựa chọn giữa nhiều phương thức thanh toán (Visa, Momo, tiền mặt, …), để sử dụng cách thanh toán thuận tiện nhất cho tôi.
* Nhận được thông báo kết quả thanh toán
* Là một Người mua, tôi muốn nhận email hoặc thông báo về kết quả giao dịch (thành công/thất bại), để biết chắc chắn đơn hàng của tôi đã được xử lý.

**System (Hệ thống)**

* Xác thực giỏ hàng trước khi thanh toán
* Là Hệ thống, tôi muốn kiểm tra thông tin món ăn trong giỏ hàng (còn hàng, giá chính xác), để đảm bảo dữ liệu đơn hàng là hợp lệ trước khi xử lý thanh toán.
* Xử lý thanh toán
* Là Hệ thống, tôi muốn kết nối đến cổng thanh toán tương ứng (Visa, Momo, …) và xử lý giao dịch, để hoàn tất quá trình mua hàng của Người mua.
* Gửi thông báo giao dịch
* Là Hệ thống, tôi muốn gửi email hoặc thông báo đến Người mua ngay sau khi giao dịch được xử lý, để xác nhận trạng thái thanh toán và chi tiết đơn hàng.

1. Với Đánh giá (Rating):

* Là Người mua (Buyer), tôi muốn tạo đánh giá cho sản phẩm mà tôi đã mua, với số sao (1–5) và nhận xét tùy chọn, để chia sẻ trải nghiệm của mình.
* Nếu tôi chỉ chọn số sao mà không nhập nhận xét, hệ thống vẫn phải lưu đánh giá.
* Khi tôi gửi đánh giá, hệ thống phải cập nhật điểm trung bình của sản phẩm ngay lập tức.
* Là Người mua (Buyer), tôi muốn có thể chỉnh sửa hoặc xóa đánh giá của chính mình, để điều chỉnh khi có trải nghiệm mới hoặc khi tôi nhập sai thông tin.
* Là Người mua (Buyer), tôi muốn thấy tất cả các đánh giá của những Buyer khác trong trang chi tiết sản phẩm, để tham khảo trước khi quyết định mua.
  + Ràng buộc:
* Buyer chỉ có thể đánh giá sản phẩm đã mua.
  + Là Người mua (Buyer), tôi muốn có thể tạo đánh giá cho cửa hàng mà tôi đã mua sản. phẩm/dịch vụ, để phản ánh trải nghiệm tổng thể về cửa hàng.
  + Đánh giá gồm số sao (1–5) và nhận xét tùy chọn.
  + Nếu chỉ nhập số sao mà không nhập nhận xét, hệ thống vẫn lưu đánh giá.
  + Khi tôi gửi đánh giá, điểm trung bình của cửa hàng phải được cập nhật ngay.
* Là Người mua (Buyer), tôi muốn có thể chỉnh sửa hoặc xóa đánh giá cửa hàng của mình, để thay đổi nhận xét nếu có trải nghiệm mới.
* Là Người mua (Buyer), tôi muốn xem các đánh giá cửa hàng từ những Buyer khác trong Store Detail, để tham khảo chất lượng dịch vụ trước khi quyết định mua.
* Ràng buộc:
* Buyer chỉ có thể đánh giá cửa hàng mà mình đã từng mua sản phẩm/dịch vụ.

1. Với Kiểm soát truy cập (Access Control):

* Là một Người dùng (Người mua hoặc SysAdmin), tôi muốn đăng nhập vào hệ thống, để truy cập các chức năng phù hợp với vai trò của mình.
* Nếu vai trò là Người mua, tôi sẽ được đưa đến trang danh mục sản phẩm.
* Nếu vai trò là SysAdmin, tôi sẽ được đưa đến trang quản trị.
* Là một Người dùng (Người mua hoặc SysAdmin), tôi muốn đăng xuất khỏi hệ thống, để đảm bảo tài khoản của mình được bảo mật khi rời khỏi hệ thống.

**---HẾT---**

**TRÂN TRỌNG CẢM ƠN**