**ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**⸎⸎⸎⸎⸎**

A logo of a university

AI-generated content may be incorrect.

**BÀI TẬP LỚN KẾT THÚC HỌC PHẦN PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM**

Đề tài: Xây dựng web bán đồ uống

**Giảng viên hướng dẫn: ThS.Vũ Quang Dũng**

**ThS.Nguyễn Xuân Quế**

**Sinh viên thực hiện:**

**Nguyễn Đình Đạt (23010282)**

**Nguyễn Đăng Hanh ( 23010243 )**

**Nguyễn Mạnh Hòa ( 23010254 )**

**Nhóm : 11**

**Khóa: K17**

**Ngành : Công nghệ thông tin**

**Năm học 2025-2026**

Mục lục

[**LỜI MỞ ĐẦU** 5](#_Toc212142036)

[**I.** **Giới thiệu đề tài.** 5](#_Toc212142037)

[**1.** **Lý do lựa chọn đề tài.** 5](#_Toc212142038)

[**2.** **Mục tiêu hệ thống.** 5](#_Toc212142039)

[**3.** **Phạm vi và đối tượng sử dụng.** 5](#_Toc212142040)

[**4.** **Công nghệ sử dụng.** 5](#_Toc212142041)

[**II.** **Quá trình tiếp xúc khách hàng lấy yêu cầu.** 6](#_Toc212142042)

[**1.** **Giai đoạn 1 : Tìm hiểu và xác định đối tượng khách hàng.** 6](#_Toc212142043)

[**2.** **Giai đoạn 2 : Khảo sát và thu thập yêu cầu.** 6](#_Toc212142044)

[**3.** **Giai đoạn 3 : Phân tích và tổng hợp kết quả khảo sát.** 6](#_Toc212142045)

[**4.** **Giai đoạn 4 : Kiểm thử và phản hồi sau khi trải nghiệm.** 6](#_Toc212142046)

[**5.** **Kết quả sau quá trình tiếp xúc với khách hàng.** 6](#_Toc212142047)

[**III.** **Phương pháp, quy trình quản lý dự án và lập kế hoạch.** 7](#_Toc212142048)

[**1.** **Mục tiêu quản lý dự án.** 7](#_Toc212142049)

[**2.** **Phương pháp quản lý dự án.** 7](#_Toc212142050)

[**3.** **Kế hoạch triển khai theo Sprint.** 8](#_Toc212142051)

[**3.1.** **Sprint 1 (tuần 1-2).** 8](#_Toc212142052)

[**3.2.** **Sprint 2 (tuần 3-4-5).** 8](#_Toc212142053)

[**3.3.** **Sprint 3 (tuần 6-7).** 8](#_Toc212142054)

[**3.4.** **Sprint 4 (tuần 8-9).** 8](#_Toc212142055)

[**4.** **Đánh giá kết quả dự án.** 8](#_Toc212142056)

[**IV.** **Phân tích yêu cầu chi tiết cho hệ thống quản lý bán đồ uống online.** 9](#_Toc212142057)

[**1.** **Yêu cầu chức năng.** 9](#_Toc212142058)

[**2.** **Yêu cầu phi chức năng.** 9](#_Toc212142059)

[**V.** **Phân tích nghiệp vụ và chức năng.** 9](#_Toc212142060)

[**1.** **Phân tích yêu cầu người dùng.** 9](#_Toc212142061)

[**2.** **Yêu cầu chức năng (Functional Requirements – FR).** 10](#_Toc212142062)

[**VI.** **Mô hình hóa yêu cầu.** 12](#_Toc212142063)

[**1.** **Mô hình hóa yêu cầu.** 12](#_Toc212142064)

[**2.** **Cấu trúc use-case (UC).** 12](#_Toc212142065)

[**3.** **Sơ đồ các use-cases.** 17](#_Toc212142066)

[**4.** **Luồng dữ liệu cơ bản.** 18](#_Toc212142067)

[**VII.** **Chi tiết luồng nghiệp vụ của khách hàng và admin.** 19](#_Toc212142068)

[**VIII.** **Cấu trúc sequence, activity, state diagrams cho mỗi luồng nghiệp vụ.** 22](#_Toc212142069)

[**1.** **Luồng nghiệp vụ : Đăng nhập/đăng ký.** 22](#_Toc212142070)

[**2.** **Luồng nghiệp vụ : Xem danh sách sản phẩm và chi tiết sản phẩm.** 25](#_Toc212142071)

[**3.** **Luồng nghiệp vụ : Tìm kiếm sản phẩm.** 26](#_Toc212142072)

[**4.** **Luồng nghiệp vụ : Quản lý giỏ hàng.** 28](#_Toc212142073)

[**5.** **Luồng nghiệp vụ : Đặt hàng và thanh toán.** 30](#_Toc212142074)

[**6.** **Luồng nghiệp vụ : Theo dõi đơn hàng.** 32](#_Toc212142075)

[**7.** **Luồng nghiệp vụ : Cập nhật thông tin cá nhân và địa chỉ giao hàng.** 34](#_Toc212142076)

[**8.** **Luồng nghiệp vụ : Quản lý sản phẩm.** 36](#_Toc212142077)

[**9.** **Luồng nghiệp vụ : Quản lý đơn hàng.** 38](#_Toc212142078)

[**10.** **Luồng nghiệp vụ : Quản lý tài khoản khách hàng.** 41](#_Toc212142079)

[**11.** **Luồng nghiệp vụ : Thống kê doanh thu.** 44](#_Toc212142080)

[**IX.** **Phân tích hướng đối tượng.** 46](#_Toc212142081)

[**1.** **Xác định các đối tượng ( Objects ) và lớp ( Classes ) chính.** 46](#_Toc212142082)

[**2.** **Thuộc tính (Attributes) và Phương thức (Methods).** 47](#_Toc212142083)

[**3.** **Mối quan hệ giữa các lớp.** 47](#_Toc212142084)

[**4.** **Áp dụng các nguyên lý Hướng đối tượng.** 48](#_Toc212142085)

[**X.** **Data flow.** 48](#_Toc212142086)

[**1.** **Sơ đồ quan hệ thực thể (ERD - Entity-Relationship Diagram).** 48](#_Toc212142087)

[**2.** **Sơ đồ luồng dữ liệu (DFD - Data Flow Diagram).** 49](#_Toc212142088)

[**XI.** **Sơ đồ lớp và cấu trúc cơ sở dữ liệu.** 50](#_Toc212142089)

[**1.** **Sơ đồ lớp (Class diagram ).** 50](#_Toc212142090)

[**2.** **Cấu trúc cơ sở dữ liệu.** 52](#_Toc212142091)

[**2.1.** **Bảng chính.** 53](#_Toc212142092)

[**2.2.** **Bảng phụ.** 54](#_Toc212142093)

[**XII.** **Cấu trúc quá trình thiết kế.** 54](#_Toc212142094)

[**1.** **Thiết kế tổng quan.** 54](#_Toc212142095)

[**2.** **Thiết kế kiến trúc.** 55](#_Toc212142096)

[**3.** **Thiết kế chức năng.** 56](#_Toc212142097)

[**3.1.** **Thiết kế cho Module Sản phẩm.** 56](#_Toc212142098)

[**3.2.** **Thiết kế cho Module Khách hàng (User).** 56](#_Toc212142099)

[**3.3.** **Thiết kế cho Module Địa chỉ khách hàng (UserAddress).** 57](#_Toc212142100)

[**3.4.** **Thiết kế cho Module Đơn hàng (Orders).** 57](#_Toc212142101)

[**3.5.** **Thiết kế cho Module Chi tiết đơn hàng (OrderDetail).** 57](#_Toc212142102)

[**3.6.** **Thiết kế cho Module Thanh toán (Payments).** 58](#_Toc212142103)

[**4.** **Thiết kế chương trình.** 58](#_Toc212142104)

[**XIII.** **Thiết kế tổng quan.** 59](#_Toc212142105)

[**1.** **Phân tách kiến trúc tổng thể.** 59](#_Toc212142106)

[**2.** **Sơ đồ tương tác cấp cao.** 60](#_Toc212142107)

[**2.1.** **Luồng đặt hàng của khách hàng.** 61](#_Toc212142108)

[**2.2.** **Luồng xử lý đơn hàng của Admin.** 61](#_Toc212142109)

[**3.** **Công nghệ dự kiến.** 61](#_Toc212142110)

[**XIV.** **Thiết kế kiến trúc.** 61](#_Toc212142111)

[**1.** **Mô hình kiến trúc 3 tầng.** 61](#_Toc212142112)

[**2.** **Chi tiết các tầng và thành phần.** 61](#_Toc212142113)

[**2.1.** **Tầng trình bày.** 62](#_Toc212142114)

[**2.2.** **Tầng nghiệp vụ.** 62](#_Toc212142115)

[**2.3.** **Tầng dữ liệu.** 62](#_Toc212142116)

[**3.** **Thiết kế tích hợp.** 63](#_Toc212142117)

[**XV.** **Thiết kế chức năng.** 63](#_Toc212142118)

[**1.** **Module khách hàng ( User module ).** 63](#_Toc212142119)

[**2.** **Module quản trị viên ( Admin module ).** 64](#_Toc212142120)

[**3.** **Tích hợp bên thứ ba.** 64](#_Toc212142121)

[**XVI.** **Thiết kế chương trình.** 64](#_Toc212142122)

[**1.** **Cấu trúc dữ liệu và mô hình hóa.** 64](#_Toc212142123)

[**2.** **Thiết kế mã nguồn.** 65](#_Toc212142124)

[**XVII.** **Thiết kế giao diện.** 66](#_Toc212142125)

[**1.** **Mục tiêu thiết kế giao diện.** 66](#_Toc212142126)

[**2.** **Nguyên tắc thiết kế chung.** 66](#_Toc212142127)

[**3.** **Thiết kế giao diện cho khách hàng.** 67](#_Toc212142128)

[**4.** **Thiết kế giao diện cho quản trị viên (Admin interface).** 67](#_Toc212142129)

[**XVIII.** **Kết quả thực hiện.** 68](#_Toc212142130)

[**1.** **Tổng quan kết quả đạt được.** 68](#_Toc212142131)

[**2.** **Các chức năng đã hoàn thiện.** 68](#_Toc212142132)

[**XIX.** **Kết luận và đánh giá.** 68](#_Toc212142133)

[**1.** **Kết luận chung.** 68](#_Toc212142134)

[**2.** **Những khó khăn gặp phải.** 69](#_Toc212142135)

[**3.** **Đánh giá của nhóm thực hiện.** 69](#_Toc212142136)

[**4.** **Bài học kinh nghiệm.** 69](#_Toc212142137)

[**5.** **Hướng phát triển trong tương lai.** 69](#_Toc212142138)

[**Lời kết** 71](#_Toc212142139)

**Phân chia công việc :**

|  |  |
| --- | --- |
| Nguyễn Đình Đạt  ( 23010282 ) | * Thiết kế và lập trình trang lịch sử đơn hàng. * Thiết kế và lập trình trang đặt hàng. * Thiết kế và lập trình chức năng sửa/xóa sản phẩm. * Thiết kế và lập trình chức năng tìm kiếm, xem trạng thái đơn hàng, trang thống kê. |
| Nguyễn Đăng Hanh  (23010243) | * Thiết kế giao diện trang chủ, xử lí hiển thị sản phẩm và xem chi tiết sản phẩm. * Thiết kế và lập trình chức năng đăng nhập, đăng ký, xử lý đăng nhập, xử lí đăng ký. |
| Nguyễn Mạnh Hòa  (23010254) | * Thiết kế và lập trình chức năng thêm sản phẩm. * Thiết kế chức năng xem thông tin khách hàng(admin). * Thiết kế trang giỏ hàng và xử lý đặt hàng thành công. |

# **LỜI MỞ ĐẦU**

Trước khi bắt đầu dự án, chúng em xin gửi lời cảm ơn chân thành đến thầy Vũ Quang Dũng và thầy Nguyễn Xuân Quế đã tận tình hướng dẫn và hỗ trợ chúng em trong suốt quá trình học tập.

1. **Giới thiệu đề tài.**
2. **Lý do lựa chọn đề tài.**

* Trong bối cảnh xã hội hiện nay, nhu cầu tiêu thụ đồ uống ngày càng tăng cao và đa dạng, đặc biệt là khi thói quen mua sắm trực tuyến ngày càng trở nên phổ biến. Chính vì vậy, nhóm lựa chọn đề tài “ Xây dựng website bán đồ uống ” nhằm đáp ứng nhu cầu thực tế của người tiêu dùng cũng như hỗ trợ hoạt động kinh doanh trong lĩnh vực này.

1. **Mục tiêu hệ thống.**

* Mục tiêu của hệ thống là xây dựng một website bán đồ uống trực tuyến hiện đại, thân thiện với người dùng và đáp ứng tốt nhu cầu của cả khách hàng lẫn người quản lý.
* Hệ thống hướng đến việc cung cấp một nền tảng giúp khách hàng có thể dễ dàng tìm kiếm, xem thông tin chi tiết về các loại đồ uống, thêm sản phẩm vào giỏ hàng, đặt mua. Bên cạnh đó, hệ thống còn hỗ trợ người quản trị trong việc quản lý danh mục sản phẩm, giá bán,trạng thái (còn hàng/hết hàng) , thông tin khách hàng và xử lý đơn hàng hiệu quả.
* Ngoài ra, website được xây dựng với mục tiêu đảm bảo tính ổn định, bảo mật dữ liệu, giao diện trực quan, dễ sử dụng và có khả năng mở rộng trong tương lai, đáp ứng xu hướng phát triển của thương mại điện tử hiện nay.

1. **Phạm vi và đối tượng sử dụng.**

* Khách hàng : Đăng ký tài khoản, đăng nhập, tìm kiếm sản phẩm, thêm sản phẩm vào giỏ hàng, đặt hàng, xem thông tin khách hàng, chỉnh sửa thông tin giao hàng, xem chi tiết thông tin sản phẩm, xem/cập nhật giỏ hàng, xem đơn hàng đã đặt và tổng số tiền cần thanh toán.
* Quản trị viên : Quản lý thông tin khách hàng(xóa tài khoản nếu cần), thêm/sửa/xóa sản phẩm, quản lý đơn hàng(xác nhận và cập nhật trạng thái giao hàng), thống kê doanh thu.

1. **Công nghệ sử dụng.**

* Ngôn ngữ lập trình : Java
* Framework : Spring Boot, Bootstrap
* Giao diện người dùng ( Front-end ) : HTML, CSS, JS
* Cơ sở dữ liệu ( Database ) : MySQL
* Kết nối cơ sở dữ liệu : XAMPP
* Môi trường phát triển : Visual Studio Code
* Hệ thống quản lý mã nguồn : github
* Máy chủ chạy ứng dụng : Máy chủ nhúng (Embedded Server)

1. **Quá trình tiếp xúc khách hàng lấy yêu cầu.**
2. **Giai đoạn 1 : Tìm hiểu và xác định đối tượng khách hàng.**

* Ở giai đoạn đầu, nhóm tiến hành tìm hiểu về thị trường đồ uống và xác định nhóm khách hàng mục tiêu, bao gồm người tiêu dùng cá nhân (đặt mua đồ uống trực tuyến) và các cửa hàng có nhu cầu bán hàng online.
* Việc xác định đúng đối tượng giúp nhóm định hướng phát triển chức năng phù hợp, tập trung vào trải nghiệm người dùng và công cụ quản lý cho chủ cửa hàng.

1. **Giai đoạn 2 : Khảo sát và thu thập yêu cầu.**

* Nhóm đã thực hiện khảo sát trực tiếp tại một số cửa hàng đồ uống, đồng thời phát phiếu hỏi và trao đổi với khách hàng để tìm hiểu về thói quen mua hàng, những khó khăn thường gặp khi đặt đồ uống online và các tính năng họ mong muốn trên website.
* Qua đó, nhóm ghi nhận các yêu cầu quan trọng như: giao diện dễ sử dụng, hiển thị rõ thông tin sản phẩm, có giỏ hàng tiện lợi, hỗ trợ đặt hàng nhanh và theo dõi trạng thái đơn hàng.

1. **Giai đoạn 3 : Phân tích và tổng hợp kết quả khảo sát.**

* Sau khi thu thập ý kiến, nhóm tiến hành tổng hợp, phân loại và phân tích dữ liệu để xác định các yêu cầu chính cần có trong hệ thống. Các yêu cầu này được sử dụng làm cơ sở xây dựng tài liệu đặc tả và thiết kế chức năng của website.

1. **Giai đoạn 4 : Kiểm thử và phản hồi sau khi trải nghiệm.**

* Sau khi hoàn thiện phiên bản thử nghiệm, nhóm tiếp tục gửi cho một số khách hàng và chủ cửa hàng dùng thử để lấy phản hồi thực tế. Những góp ý về giao diện, tốc độ tải trang, quy trình đặt hàng và tính năng quản lý được ghi nhận và điều chỉnh để hoàn thiện sản phẩm.

1. **Kết quả sau quá trình tiếp xúc với khách hàng.**

* Sau khi hoàn thành các giai đoạn tiếp xúc và khảo sát, nhóm đã thu được nhiều thông tin hữu ích giúp hoàn thiện hệ thống website bán đồ uống.
* Khách hàng đặc biệt quan tâm đến tốc độ đặt hàng nhanh, giao diện thân thiện, hình ảnh sản phẩm sinh động, và thông tin chi tiết rõ ràng như giá, dung tích, thành phần.
* Người quản lý cửa hàng quan tâm đến các tính năng như quản lý đơn hàng, cập nhật số lượng tồn kho, thống kê doanh thu và quản lý khách hàng được xem là cần thiết và ưu tiên hàng đầu.
* Sau khi nhận được ý kiến đóng góp của khách hàng, nhóm đã điều chỉnh thiết kế giao diện trở nên trực quan hơn, bổ sung chức năng tìm kiếm nhanh, cải thiện bố cục hiển thị sản phẩm, và tối ưu quy trình đặt hàng chỉ với vài thao tác. Ngoài ra, nhóm cũng thêm các chức năng hỗ trợ quản trị như quản lý sản phẩm, thống kê, giúp hệ thống không chỉ phục vụ khách hàng mua hàng mà còn hỗ trợ người bán quản lý kinh doanh hiệu quả.
* Qua quá trình tiếp xúc với khách hàng, nhóm hiểu rõ hơn nhu cầu thực tế, định hướng phát triển đúng mục tiêu và hoàn thiện hệ thống theo hướng thân thiện, tiện ích và mang tính ứng dụng cao.

1. **Phương pháp, quy trình quản lý dự án và lập kế hoạch.**
2. **Mục tiêu quản lý dự án.**

* Mục tiêu của công tác quản lý dự án là đảm bảo quá trình phát triển hệ thống Website bán đồ uống online diễn ra đúng tiến độ, đúng phạm vi, và đạt chất lượng mong muốn.
* Quản lý dự án giúp :
* Phân bổ công việc hợp lý giữa các thành viên.
* Theo dõi tiến độ, phát hiện sớm rủi ro.
* Tăng khả năng phối hợp và trao đổi thông tin giữa các thành viên.
* Đảm bảo sản phẩm cuối cùng đáp ứng yêu cầu người dùng và khách hàng.

1. **Phương pháp quản lý dự án.**

* Dự án được thực hiện theo phương pháp Agile – Scrum, phù hợp với nhóm sinh viên và các dự án phần mềm quy mô vừa.
* Đặc điểm :
* Chia dự án thành các giai đoạn nhỏ (Sprint), mỗi Sprint kéo dài 1–2 tuần.
* Sau mỗi Sprint, nhóm có bản demo sản phẩm hoạt động được để giảng viên/hướng dẫn viên góp ý.
* Linh hoạt thích ứng khi có thay đổi về yêu cầu hoặc chức năng.

1. **Kế hoạch triển khai theo Sprint.**

* Tổng thời gian dự án : 9 tuần.
  1. **Sprint 1 (tuần 1-2).**
* Mục tiêu chính : Phân tích yêu cầu và thiết kế UML cơ bản.
* Công việc chi tiết :
* Khảo sát quy trình bán hàng đồ uống.
* Xác định Actor và Use-case.
* Vẽ Use Case Diagram và kịch bản chi tiết.
* Vẽ biểu đồ hoạt động.
* Kết quả đầu ra : Tài liệu yêu cầu, biểu đồ Use Case, biểu đồ hoạt động.
  1. **Sprint 2 (tuần 3-4-5).**
* Mục tiêu chính : Thiết kế hệ thống.
* Công việc chi tiết :
* Vẽ Biểu đồ lớp, Biểu đồ trình tự, Biểu đồ trạng thái.
* Thiết kế cơ sở dữ liệu.
* Xây dựng khung giao diện mẫu.
* Kết quả đầu ra : Tập hợp các biểu đồ UML, CSDL.
  1. **Sprint 3 (tuần 6-7).**
* Mục tiêu chính : Lập trình và kiểm thử cơ bản.
* Công việc chi tiết :
* Xây dựng các chức năng: đăng nhập, xem sản phẩm, giỏ hàng, thanh toán.
* Cài đặt cơ sở dữ liệu và kết nối.
* Kiểm thử đơn vị
* Kết quả đầu ra : Chức năng cốt lõi hoạt động được.
  1. **Sprint 4 (tuần 8-9).**
* Mục tiêu chính : Hoàn thiện và báo cáo.
* Công việc chi tiết :
* Hoàn thiện giao diện và các chức năng quản trị.
* Kiểm thử hệ thống, sửa lỗi.
* Viết báo cáo và chuẩn bị thuyết trình.
* Kết quả đầu ra : Hệ thống hoàn chỉnh và báo cáo tổng kết.

1. **Đánh giá kết quả dự án.**

* Tất cả các Sprint đều được thực hiện đúng tiến độ theo kế hoạch ban đầu.
* Việc sử dụng mô hình Agile giúp nhóm phản hồi nhanh với yêu cầu thay đổi.
* Giao tiếp nhóm hiệu quả thông qua Zalo và họp định kỳ mỗi tuần.
* Kết quả: hệ thống đạt 100% chức năng cốt lõi và có thể mở rộng trong tương lai.

1. **Phân tích yêu cầu chi tiết cho hệ thống quản lý bán đồ uống online.**
2. **Yêu cầu chức năng.**

* Đối với khách hàng :
* Xem danh sách sản phẩm,chi tiết thông tin sản phẩm.
* Tìm kiếm,lọc sản phẩm.
* Thêm/xóa/sửa sản phẩm trong giỏ hàng.
* Tính tổng tiền tự động,hỗ trợ nhiều phương thức thanh toán.
* Đặt hàng trực tuyến và lưu lịch sử đơn hàng.
* Đăng nhập/Đăng ký.
* Sửa thông tin cá nhân,địa chỉ giao hàng.
* Đối với quản trị viên ( admin ) :
* Quản lý sản phẩm(thêm/sửa/xóa) sản phẩm.
* Quản lý đơn hàng(xác nhận và cập nhật trạng thái giao hàng).
* Quản lý tài khoản khách hàng(xem thông tin khách hàng )
* Thống kê doanh thu.

1. **Yêu cầu phi chức năng.**

* Giao diện thân thiện, dễ sử dụng.
* Hỗ trợ mobile.
* Website tải nhanh, đáp ứng nhiều người dùng đồng thời.
* Bảo vệ dữ liệu khách hàng.
* Dễ nâng cấp.

1. **Phân tích nghiệp vụ và chức năng.**
2. **Phân tích yêu cầu người dùng.**

* Phân tích yêu cầu người dùng giúp xác định ai là người sử dụng hệ thống và họ cần làm gì.
* Hệ thống có hai nhóm người dùng chính: Khách hàng và Quản trị viên (Admin).

**Nhóm 1 : Khách hàng ( User ) :**

* Nhu cầu chính :
* Tìm kiếm và mua sắm: Khách hàng muốn tìm kiếm và mua sản phẩm một cách nhanh chóng, dễ dàng.
* Xem thông tin sản phẩm: Khách hàng cần xem chi tiết sản phẩm (ảnh, mô tả, giá tiền, định lượng).
* Thêm sản phẩm vào giỏ hàng : Khách hàng muốn có tính năng thêm vào giỏ hàng để có thể mua trong tương lai.
* Hỗ trợ nhiều phương thức thanh toán : Để khách hàng có thể thanh toán bằng nhiều phương thức ( Momo, thẻ tín dụng, zalo pay, …).
* Chỉnh sửa thông tin cá nhân : Khách hàng muốn chỉnh sửa thông tin cá nhân để có thể thay đổi địa chỉ nhận hàng khi cần.
* Mong muốn :
* Giao diện thân thiện, dễ sử dụng trên cả máy tính và điện thoại di động.
* Tốc độ tải trang nhanh, không bị gián đoạn.
* Thông tin sản phẩm chính xác, hình ảnh đẹp, rõ nét.

**Nhóm 2 : Quản trị viên ( Admin ) :**

* Nhu cầu chính :
* Quản lý sản phẩm : Admin cần quyền thêm/sửa/xóa sản phẩm để có thể thay đổi thông tin sản phẩm hoặc cập nhật các sản phẩm mới ra mắt.
* Quản lý đơn hàng : Admin có thể theo dõi thông tin đơn hàng và trạng thái đơn hàng.
* Quản lý khách hàng : Admin cần xem được thông tin khách hàng (tên, số điện thoại, email, tài khoản, mật khẩu, địa chỉ).
* Thống kê và báo cáo: Admin cần các báo cáo về doanh thu.
* Mong muốn :
* Giao diện quản trị trực quan, dễ thao tác.
* Thông tin báo cáo chính xác, cập nhật theo thời gian thực.

1. **Yêu cầu chức năng (Functional Requirements – FR).**

* Dựa trên phân tích người dùng và các luồng nghiệp vụ đã có, các yêu cầu chức năng được xác định như sau

**FR-01 : Quản lý giao diện người dùng**

* FR-01.1 : Hệ thống cho phép khách hàng xem danh sách sản phẩm.
* FR-01.2 : Hệ thống cho phép khách hàng xem chi tiết sản phẩm.
* FR-01.3 : Hệ thống phải có tính năng tìm kiếm sản phẩm.
* FR-01.4 : Hệ thống phải có giỏ hàng, cho phép khách hàng thêm/xóa sản phẩm và cập nhật số lượng.

**FR-02 : Xử lý đặt hàng và thanh toán**

* FR-02.1 : Hệ thống phải có tính năng đặt hàng.
* FR-02.2 : Hệ thống phải hỗ trợ nhiều phương thức thanh toán (ví điện tử, thẻ tín dụng/ghi nợ).

**FR-03 : Xem chi tiết đơn hàng**

* FR-03.1 : Hệ thống phải có tính năng xem chi tiết đơn hàng.
* FR-03.2 : Hệ thống phải có tính năng xem lịch sử mua hàng.

**FR-04 : Quản lý sản phẩm**

* FR-04.1 : Hệ thống phải cho phép Admin thêm, sửa, xóa sản phẩm, bao gồm các thông tin cơ bản.
* FR-04.2 : Hệ thống phải cho phép Admin nhập và tồn kho(còn hàng/hết hàng).

**FR-05 : Quản lý đơn hàng**

* FR-05.1 : Hệ thống phải cho phép Admin xem danh sách các đơn hàng, lọc theo trạng thái (chờ xử lý, đang giao hàng, đã hoàn thành).
* FR-05.2 : Hệ thống phải cho phép Admin xác nhận, cập nhật trạng thái đơn hàng và hủy đơn hàng.

**FR-06 : Quản lý khách hàng**

* FR-06.1 : Hệ thống phải cho phép khách hàng đăng ký, đăng nhập và quản lý hồ sơ cá nhân (thông tin, địa chỉ giao hàng).
* FR-06.2 : Hệ thống phải cho phép Admin xem thông tin khách hàng và trạng thái đơn hàng.
* FR-06.3 : Hệ thống phải cho phép admin xem thông tin khách hàng hoặc tài khoản khách hàng.

**FR-07 : Thống kê doanh thu**

* FR-07.1 : Hệ thống phải cung cấp các báo cáo về doanh thu.
* FR-07.2 : Hệ thống phải cung cấp báo cáo về tình trạng tồn kho(còn hàng/hết hàng).

1. **Mô hình hóa yêu cầu.**
2. **Mô hình hóa yêu cầu.**

* Các tác nhân là những người hoặc hệ thống bên ngoài tương tác với hệ thống phần mềm.
* Khách hàng (Customer): Người dùng cuối, thực hiện các nghiệp vụ mua sắm.
* Quản trị viên (Admin): Người quản lý hệ thống, xử lý các nghiệp vụ nội bộ.

1. **Cấu trúc use-case (UC).**

**UC-01 : Đăng nhập/Đăng ký.**

* Actor: Khách hàng
* Mục tiêu: Cho phép khách hàng truy cập hệ thống bằng tài khoản hợp lệ.
* Mô tả: Khách hàng có thể đăng nhập bằng tài khoản đã có hoặc đăng ký tài khoản mới.
* Tiền điều kiện: Khách hàng đã truy cập trang web.
* Hậu điều kiện: Khách hàng được xác thực và sử dụng hệ thống.
* Luồng chính:
* Khách hàng chọn Đăng nhập/Đăng ký.
* Hệ thống hiển thị form đăng nhập.
* Khách hàng nhập email/số điện thoại và mật khẩu.
* Hệ thống xác thực thông tin.
* Nếu hợp lệ → đăng nhập thành công.
* Nếu chưa có tài khoản → khách hàng chuyển sang đăng ký.
* Khách hàng điền thông tin đăng ký.
* Hệ thống kiểm tra hợp lệ và lưu thông tin.
* Khách hàng đăng nhập thành công.
* Luồng ngoại lệ:
* Sai mật khẩu → hệ thống báo lỗi.
* Tài khoản không tồn tại → yêu cầu đăng ký.
* Thông tin đăng ký không hợp lệ → báo lỗi và yêu cầu nhập lại.
* Business Rules:
* Mật khẩu tối thiểu 6 ký tự.
* Email/SĐT duy nhất trong hệ thống.

**UC-02 : Xem danh sách sản phẩm & Chi tiết sản phẩm.**

* Actor: Khách hàng
* Mục tiêu: Khách hàng có thể duyệt sản phẩm và xem chi tiết từng sản phẩm.
* Mô tả: Hiển thị danh sách sản phẩm và thông tin chi tiết.
* Tiền điều kiện: Khách hàng đã vào trang “Sản phẩm”.
* Hậu điều kiện: Hệ thống hiển thị chính xác thông tin sản phẩm.
* Luồng chính:
* Khách hàng truy cập trang “Sản phẩm”.
* Hệ thống hiển thị danh sách sản phẩm (hình ảnh, giá, tồn kho).
* Khách hàng chọn sản phẩm bất kỳ.
* Hệ thống hiển thị chi tiết sản phẩm (mô tả, giá, khuyến mãi, đánh giá).
* Luồng ngoại lệ:
* Không tải được dữ liệu → báo “Lỗi tải dữ liệu, vui lòng thử lại”.
* Business Rules:
* Danh sách sản phẩm có phân trang.
* Thông tin chi tiết phải đầy đủ: tên, mô tả, giá, số lượng tồn.

**UC-03 : Tìm kiếm sản phẩm.**

* **Actor:** Khách hàng
* **Mục tiêu:** Giúp khách hàng tìm sản phẩm nhanh chóng.
* **Mô tả:** Cho phép tìm kiếm bằng từ khóa.
* **Tiền điều kiện:** Khách hàng đã truy cập trang sản phẩm.
* **Hậu điều kiện:** Hệ thống hiển thị sản phẩm phù hợp.
* **Luồng chính:**
* Khách hàng nhập tên sản phẩm vào ô tìm kiếm.
* Hệ thống xử lý truy vấn.
* Hệ thống hiển thị danh sách kết quả.
* Luồng ngoại lệ:
* Không có kết quả → hiển thị thông báo “Không có sản phẩm phù hợp”.
* Business Rules:
* Hỗ trợ tìm theo tên.
* Thời gian phản hồi < 3 giây.

**UC-04 : Quản lý giỏ hàng.**

* Actor: Khách hàng
* Mục tiêu: Quản lý các sản phẩm đã chọn trước khi đặt hàng.
* Mô tả: Cho phép thêm, sửa, xóa sản phẩm trong giỏ.
* Tiền điều kiện: Khách hàng đã chọn ít nhất 1 sản phẩm.
* Hậu điều kiện: Giỏ hàng được cập nhật chính xác.
* Luồng chính:
* Khách hàng thêm sản phẩm vào giỏ.
* Hệ thống cập nhật giỏ hàng.
* Khách hàng vào trang giỏ hàng.
* Khách hàng thay đổi số lượng hoặc xóa sản phẩm.
* Hệ thống cập nhật tổng tiền.
* Luồng ngoại lệ:
* Sản phẩm hết hàng → hệ thống không cho thêm.
* Business Rules:
* Tổng tiền tính tự động.
* Số lượng không vượt quá tồn kho.

**UC-05 : Đặt hàng và thanh toán.**

* Actor: Khách hàng
* Mục tiêu: Hoàn tất đặt hàng và thanh toán.
* Mô tả: Khách hàng xác nhận giỏ hàng và chọn phương thức thanh toán.
* Tiền điều kiện: Giỏ hàng có sản phẩm.
* Hậu điều kiện: Đơn hàng được ghi nhận.
* Luồng chính:
* Khách hàng xác nhận giỏ hàng.
* Hệ thống hiển thị thông tin đơn hàng.
* Khách hàng chọn phương thức thanh toán.
* Hệ thống xử lý giao dịch.
* Nếu thành công → đơn hàng được lưu.
* Luồng ngoại lệ:
* Thanh toán thất bại (thẻ hết tiền, lỗi mạng).
* Quá thời gian thanh toán → đơn hàng hủy.
* Business Rules:
* Hỗ trợ nhiều phương thức: tiền mặt, thẻ, ví điện tử.

**UC-06 : Theo dõi đơn hàng.**

* Actor: Admin
* Mục tiêu: Xem trạng thái đơn hàng đã đặt.
* Mô tả: Hiển thị danh sách đơn hàng và trạng thái.
* Tiền điều kiện: Khách hàng đã đặt hàng.
* Hậu điều kiện: Trạng thái đơn hàng được hiển thị.
* Luồng chính:
* Admin vào mục “Trạng thái đơn hàng”.
* Hệ thống hiển thị danh sách đơn hàng.
* Admin chọn 1 đơn hàng.
* Hệ thống hiển thị trạng thái (đang xử lý, đang giao, đã giao, đã hủy).
* Luồng ngoại lệ:
* Không có đơn hàng → hiển thị “Bạn chưa có đơn hàng nào”.
* Lỗi kết nối server → không tải được dữ liệu.
* Business Rules:
* Trạng thái đơn hàng phải được cập nhật theo tiến trình thực tế.

**UC-07 : Cập nhật thông tin cá nhân và địa chỉ giao hàng.**

* Actor: Khách hàng
* Mục tiêu: Cho phép khách hàng thay đổi thông tin tài khoản.
* Mô tả: Sửa tên, số điện thoại, email, địa chỉ giao hàng.
* Tiền điều kiện: Khách hàng đã đăng nhập.
* Hậu điều kiện: Thông tin được lưu.
* Luồng chính:
* Khách hàng vào mục “Tài khoản của tôi”.
* Chọn “Chỉnh sửa thông tin”.
* Nhập thông tin mới.
* Hệ thống kiểm tra hợp lệ.
* Lưu và hiển thị thông báo thành công.
* Luồng ngoại lệ:
* Dữ liệu không hợp lệ (ví dụ SĐT sai định dạng).
* Business Rules:
* Email duy nhất.
* Số điện thoại phải đúng định dạng.

**UC-08 : Quản lý sản phẩm.**

* Actor: Quản trị viên
* Mục tiêu: Quản lý danh mục sản phẩm trên hệ thống.
* Mô tả: Admin thêm, sửa, xóa sản phẩm.
* Tiền điều kiện: Quản trị viên đã đăng nhập.
* Hậu điều kiện: Danh sách sản phẩm cập nhật.
* Luồng chính:
* Quản trị viên vào trang quản lý sản phẩm.
* Chọn thêm mới, sửa hoặc xóa sản phẩm.
* Nhập thông tin hoặc cập nhật sản phẩm.
* Hệ thống lưu và hiển thị danh sách mới.
* Luồng ngoại lệ:
* Dữ liệu không hợp lệ → báo lỗi.
* Business Rules:
* Giá sản phẩm phải > 0.
* Tên sản phẩm không trùng trong cùng shop.

**UC-09 : Quản lý đơn hàng.**

* Actor: Quản trị viên.
* Mục tiêu: Theo dõi và xử lý đơn hàng.
* Mô tả: Admin xác nhận và cập nhật trạng thái đơn.
* Tiền điều kiện: Hệ thống có đơn hàng.
* Hậu điều kiện: Trạng thái đơn hàng được thay đổi.
* Luồng chính:
* Quản trị viên mở danh sách đơn hàng.
* Chọn đơn cụ thể.
* Cập nhật trạng thái: “Đang xử lý”, “Đang giao”, “Đã giao”, “Đã hủy”.
* Hệ thống lưu và thông báo thành công.
* Luồng ngoại lệ:
* Lỗi hệ thống → không cập nhật được.
* Business Rules:
* Trạng thái cập nhật theo đúng quy trình.

**UC-10 : Quản lý tài khoản khách hàng.**

* Actor: Quản trị viên
* Mục tiêu: Quản lý thông tin khách hàng.
* Mô tả: Xem thông tin khách hàng.
* Tiền điều kiện: Admin đã đăng nhập.
* Hậu điều kiện: Danh sách tài khoản cập nhật.
* Luồng chính:
* Admin vào mục “Khách hàng”.
* Hệ thống hiển thị danh sách khách hàng.
* Admin chọn 1 khách hàng.
* Có thể xem chi tiết thông tin khách hàng.
* Hệ thống lưu thay đổi.
* Business Rules:
* Dữ liệu khách hàng phải được bảo mật.

**UC-11 : Thống kê doanh thu.**

* Actor: Quản trị viên
* Mục tiêu: Xem báo cáo doanh thu.
* Mô tả: Hệ thống hiển thị báo cáo theo ngày/tháng/năm.
* Tiền điều kiện: Admin đã đăng nhập.
* Hậu điều kiện: Báo cáo hiển thị chính xác.
* Luồng chính:
* Admin vào mục “Doanh số”.
* Chọn khoảng thời gian cần xem.
* Hệ thống xử lý dữ liệu.
* Hiển thị báo cáo dạng bảng/biểu đồ.
* Luồng ngoại lệ:
* Không có dữ liệu trong thời gian chọn → hiển thị “Không có doanh thu”.
* Business Rules:
* Báo cáo phải phản ánh đúng dữ liệu đơn hàng đã hoàn tất.

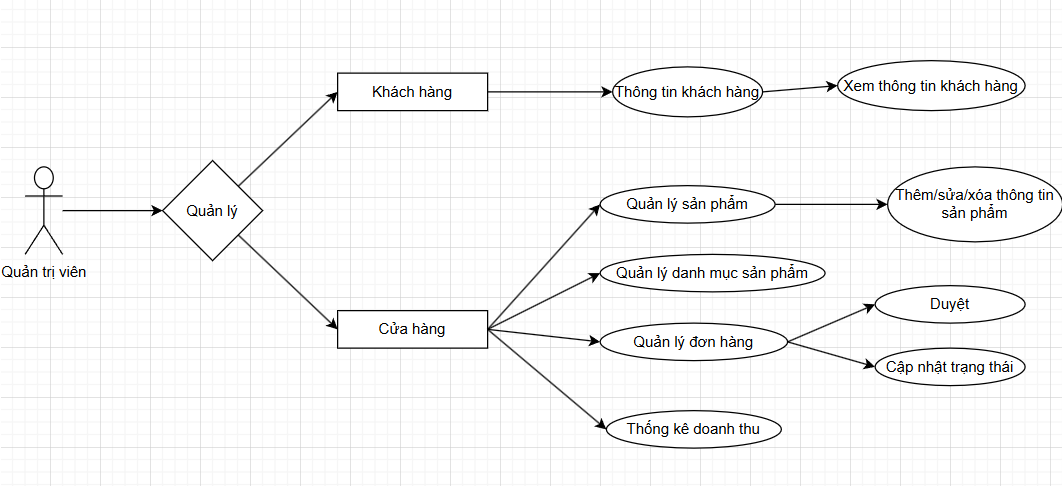
1. **Sơ đồ các use-cases.**

* Tác nhân khách hàng (User) :

A diagram of a diagram

AI-generated content may be incorrect.

* Tác nhân quản trị viên ( Admin ) :



1. **Luồng dữ liệu cơ bản.**

**DFD cấp 0 ( Tổng quan )**

* Từ khách hàng : Đăng ký, đăng nhập, yêu cầu tìm kiếm, yêu cầu đặt hàng.
* Đến khách hàng : Danh sách đồ uống, thông tin đơn hàng, xác nhận thanh toán.
* Từ quản trị viên : Thêm/sửa xóa sản phẩm, xem thông tin khách hàng, quản lý đơn hàng.
* Đến quản trị viên : Thông tin khách hàng, thông tin sản phẩm, thông tin đơn hàng, doanh thu.
* Với hệ thống thanh toán : Yêu cầu thanh toán, kết quả giao dịch.
* Với hệ thống vận chuyển : Thông tin giao hàng.

**DFD cấp 1 ( Chi tiết luồng đặt hàng )**

* Sơ đồ DFD cấp 1 (chi tiết) tập trung mô tả riêng tiến trình "Xử lý đơn hàng" trong hệ thống web bán đồ uống.
* Mục tiêu là làm rõ các bước xử lý nhỏ hơn, các luồng dữ liệu giữa tiến trình con và kho dữ liệu, cũng như các tương tác với khách hàng và hệ thống thanh toán.
* Tiến trình 1 : Chọn sản phẩm
* Dữ liệu vào :
  + query\_data: yêu cầu tìm kiếm, lọc sản phẩm (từ Khách hàng).
  + product\_data: dữ liệu sản phẩm và tồn kho (từ D1: Sản phẩm & Tồn kho).
* Xử lý:
* Nhận truy vấn tìm kiếm từ khách hàng.
* Lọc danh sách sản phẩm theo yêu cầu (tên, loại, giá...).
* Hiển thị danh sách phù hợp để khách hàng chọn sản phẩm.
* Dữ liệu ra:
* product\_list\_data: danh sách sản phẩm hiển thị (tới Khách hàng).
* cart\_data: thông tin sản phẩm được chọn (tới Tiến trình 2: Quản lý Giỏ hàng).
* Tiến trình 2 : Quản lý giỏ hàng
* Dữ liệu vào:
* cart\_data: danh sách sản phẩm được chọn (từ Tiến trình 1).
* item\_update\_data: thông tin thêm/xóa/cập nhật sản phẩm trong giỏ (từ Khách hàng).
* Xử lý:
* Nhận yêu cầu thêm, xóa hoặc cập nhật số lượng.
* Tính toán tổng tiền giỏ hàng.
* Cập nhật giỏ hàng theo thay đổi của khách hàng.
* Dữ liệu ra:
* cart\_data: giỏ hàng cập nhật (tới Khách hàng).
* order\_data: dữ liệu đơn hàng khi khách hàng xác nhận mua (tới Tiến trình 3: Thanh toán & Xử lý đơn hàng).
* Tiến trình 3 : Thanh toán và xử lý đơn hàng.
* Dữ liệu vào:
* order\_data: thông tin đơn hàng từ Tiến trình 2.
* customer\_info: thông tin khách hàng (từ Khách hàng).
* Xử lý:
* Kiểm tra tồn kho trong D1: Sản phẩm & Tồn kho để đảm bảo đủ hàng.
* Gọi API thanh toán đến Hệ thống thanh toán (Momo, zaloPay...).
* Cập nhật đơn hàng vào D2: Đơn hàng (thông tin sản phẩm, khách hàng, trạng thái).
* Cập nhật tồn kho trong D1: Sản phẩm & Tồn kho (giảm số lượng sau khi bán).
* Dữ liệu ra:
* order\_confirmation\_data: xác nhận đơn hàng, kết quả thanh toán (trả về Khách hàng).
* inventory\_update\_data: dữ liệu cập nhật tồn kho (lưu vào D1).

1. **Chi tiết luồng nghiệp vụ của khách hàng và admin.**

**Luồng 1 : Quản lý tài khoản và thông tin cá nhân.**

* Đăng ký và đăng nhập tài khoản:
* Khách hàng truy cập trang đăng nhập/đăng ký.
* Nếu đã có tài khoản → nhập email/số điện thoại và mật khẩu để đăng nhập.
* Nếu chưa có tài khoản → chọn “Đăng ký mới”.
* Điền thông tin cá nhân: họ tên, email, số điện thoại, mật khẩu.
* Hệ thống kiểm tra hợp lệ (email/sđt duy nhất, mật khẩu ≥ 6 ký tự).
* Nếu hợp lệ, hệ thống lưu thông tin và thông báo đăng ký thành công.
* Khách hàng đăng nhập và được truy cập các tính năng mua sắm.
* Quản lý hồ sơ cá nhân:
* Khách hàng vào trang “Tài khoản của tôi”.
* Có thể cập nhật họ tên, số điện thoại, địa chỉ giao hàng.
* Hệ thống kiểm tra dữ liệu, lưu và thông báo thành công.
* Thông tin mới được áp dụng cho các đơn hàng tiếp theo.

**Luồng 2 : Tìm kiếm và xem sản phẩm.**

* Xem chi tiết sản phẩm:
* Khách hàng truy cập trang “Sản phẩm”.
* Hệ thống hiển thị danh sách đồ uống (ảnh, tên, giá, tồn kho).
* Khách hàng chọn sản phẩm để xem chi tiết (mô tả, dung tích, thành phần, giá).
* Hệ thống hiển thị thông tin đầy đủ và cho phép thêm vào giỏ hàng.
* Tìm kiếm sản phẩm:
* Khách hàng nhập từ khóa .
* Hệ thống xử lý truy vấn và trả về danh sách kết quả phù hợp.
* Nếu không có sản phẩm phù hợp, hệ thống hiển thị thông báo tương ứng.

**Luồng 3 : Quản lý giỏ hàng và quy trình thanh toán.**

* Quản lý giỏ hàng:
* Khi khách hàng chọn “Thêm vào giỏ”, hệ thống cập nhật danh sách sản phẩm trong giỏ.
* Khách hàng truy cập giỏ hàng để xem các sản phẩm đã chọn.
* Có thể thay đổi số lượng, xóa sản phẩm hoặc tiếp tục mua thêm.
* Hệ thống tự động tính lại tổng tiền theo số lượng và giá từng sản phẩm.
* Đặt hàng và thanh toán:
* Khách hàng xác nhận giỏ hàng và chọn phương thức thanh toán (tiền mặt, thẻ, ví điện tử).
* Hệ thống hiển thị thông tin đơn hàng: sản phẩm, số lượng, tổng tiền, địa chỉ nhận hàng.
* Khách hàng xác nhận đặt hàng.
* Hệ thống gửi yêu cầu thanh toán (với thanh toán online).
* Nếu giao dịch thành công → lưu đơn hàng, trừ tồn kho, gửi thông báo xác nhận.
* Nếu thất bại → hiển thị lỗi và cho phép thử lại.

**Luồng 4 : Theo dõi đơn hàng.**

* Theo dõi đơn hàng:
* Quản trị viên truy cập mục “Trạng thái đơn hàng”
* Hệ thống hiển thị danh sách các đơn hàng và trạng thái đơn hàng (NEW, PAID).
* Quản trị viên chọn đơn để xem chi tiết.

**Luồng 5 : Quản lý sản phẩm và tồn kho.**

* Thêm sản phẩm mới:
* Quản trị viên đăng nhập vào hệ thống quản trị.
* Chọn “Thêm sản phẩm mới”.
* Nhập thông tin: tên đồ uống, giá, mô tả, dung tích, loại, hình ảnh.
* Nhập trạng thái tồn kho ban đầu.
* Hệ thống lưu và hiển thị danh sách sản phẩm mới.
* Cập nhật/xóa sản phẩm:
* Admin chọn sản phẩm cần chỉnh sửa hoặc xóa.
* Thực hiện thao tác cập nhật hoặc xác nhận xóa.
* Hệ thống lưu thay đổi và cập nhật dữ liệu hiển thị cho khách hàng.
* Quản lý tồn kho:
* Khi có đơn hàng mới thành công → hệ thống tự động cập nhật trạng thái tồn kho(còn hàng/ hết hàng).
* Admin có thể cập nhật thủ công khi nhập thêm hàng.

**Luồng 6 : Quản lý đơn hàng ( Admin ).**

* Xử lý đơn hàng:
* Admin truy cập trang “Quản lý đơn hàng”.
* Hệ thống hiển thị danh sách đơn hàng và trạng thái.
* Admin chọn đơn cần xử lý → xem chi tiết.
* Kiểm tra tồn kho và xác nhận đơn hàng.
* Xem trạng thái đơn hàng (NEW/PAID).
* Hệ thống lưu thay đổi và gửi thông báo cho khách hàng.

**Luồng 7 : Quản lý tài khoản khách hàng ( Admin ).**

* Admin vào mục “Khách hàng”.
* Hệ thống hiển thị danh sách khách hàng (họ tên, email, số điện thoại, địa chỉ).
* Admin có thể xem chi tiết thông tin khách hàng.
* Dữ liệu được lưu và bảo mật.

**Luồng 8 : Thống kê và báo cáo doanh thu ( Admin ).**

* Admin chọn mục “Thống kê”.
* Hệ thống xử lý và hiển thị báo cáo doanh thu, tồn kho.
* Có thể xuất báo cáo dạng bảng hoặc biểu đồ.
* Nếu không có dữ liệu → hiển thị thông báo “Không có doanh thu trong khoảng thời gian này”.

1. **Cấu trúc sequence, activity, state diagrams cho mỗi luồng nghiệp vụ.**

* Sơ đồ hoạt động ( Activity diagram ) :
* Mô tả luồng các hoạt động và các quyết định logic trong một quy trình.
* Sơ đồ tuần tự ( Sequence diagram ) :
* Thể hiện trình tự các thông điệp được gửi giữa các đối tượng để hoàn thành một tác vụ.
* Sơ đồ trạng thái ( State diagram ) :
* Mô tả các trạng thái khác nhau của một đối tượng (trong trường hợp này là DonHang) và các sự kiện gây ra sự chuyển đổi giữa các trạng thái.

1. **Luồng nghiệp vụ : Đăng nhập/đăng ký.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7.1.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7.1.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 7.1.3

1. **Luồng nghiệp vụ : Xem danh sách sản phẩm và chi tiết sản phẩm.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram with text and arrows

AI-generated content may be incorrect.

Hình 7.2.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram of a product

AI-generated content may be incorrect.

Hình 7.2.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a product

AI-generated content may be incorrect.

Hình 7.2.3

1. **Luồng nghiệp vụ : Tìm kiếm sản phẩm.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 7.3.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram of a product controller

AI-generated content may be incorrect.

Hình 7.3.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a search engine

AI-generated content may be incorrect.

Hình 7.3.3

1. **Luồng nghiệp vụ : Quản lý giỏ hàng.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 7.4.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7.4.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a process

AI-generated content may be incorrect.

Hình 7.4.3

1. **Luồng nghiệp vụ : Đặt hàng và thanh toán.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7.5.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7.5.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 7.5.3

1. **Luồng nghiệp vụ : Theo dõi đơn hàng.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram of a workflow

AI-generated content may be incorrect.

Hình 7.6.1

* Sơ đồ tuần tự ( Sequence diagram ) :

Hình 7.6.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a process

AI-generated content may be incorrect.

Hình 7.6.3

1. **Luồng nghiệp vụ : Cập nhật thông tin cá nhân và địa chỉ giao hàng.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 7.7.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram with text and arrows

AI-generated content may be incorrect.

Hình 7.7.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a program

AI-generated content may be incorrect.

Hình 7.7.3

1. **Luồng nghiệp vụ : Quản lý sản phẩm.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 7.9.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram with arrows and black text

AI-generated content may be incorrect.

Hình 7.9.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a process

AI-generated content may be incorrect.

Hình 7.9.3

1. **Luồng nghiệp vụ : Quản lý đơn hàng.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 7.10.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7.10.2

A diagram of a computer

AI-generated content may be incorrect.

Hình 7.10.3

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a delivery process

AI-generated content may be incorrect.

Hình 7.10.4

1. **Luồng nghiệp vụ : Quản lý tài khoản khách hàng.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram with text and symbols

AI-generated content may be incorrect.

Hình 7.11.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram of a computer

AI-generated content may be incorrect.

Hình 7.11.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram with black dots and black circles

AI-generated content may be incorrect.

Hình 7.11.3

1. **Luồng nghiệp vụ : Thống kê doanh thu.**

* Sơ đồ hoạt động ( Activity diagram ) :

A diagram of a flowchart

AI-generated content may be incorrect.

Hình 7.12.1

* Sơ đồ tuần tự ( Sequence diagram ) :

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7.12.2

* Sơ đồ trạng thái ( State diagram ) :

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7.12.3

1. **Phân tích hướng đối tượng.**
2. **Xác định các đối tượng ( Objects ) và lớp ( Classes ) chính.**

* Trong phân tích hướng đối tượng, chúng ta xem xét hệ thống như một tập hợp các đối tượng tương tác với nhau. Mỗi đối tượng là một thể hiện của một lớp. Dưới đây là các lớp chính được xác định từ các nghiệp vụ đã phân tích:
* User (Customer / Người dùng): Đại diện cho khách hàng sử dụng hệ thống để đăng ký, đăng nhập, đặt mua sản phẩm và quản lý đơn hàng.
* UserAddress (Địa chỉ người dùng): Đại diện cho địa chỉ cụ thể của khách hàng.
* Products (Sản phẩm): Đại diện cho các mặt hàng được bán trên hệ thống.
* Orders (Đơn hàng): Đại diện cho giao dịch mua hàng của người dùng, ghi nhận tổng tiền, trạng thái và thông tin người mua.
* OrderDetail (Chi tiết đơn hàng): Đại diện cho thông tin chi tiết từng sản phẩm trong đơn hàng, bao gồm số lượng, giá và tên sản phẩm tại thời điểm đặt hàng.
* Payments (Thanh toán): Đại diện cho thông tin thanh toán của một đơn hàng, bao gồm phương thức và số tiền thanh toán.

1. **Thuộc tính (Attributes) và Phương thức (Methods).**

* User (Người dùng):
* Thuộc tính: userId, name, email, phone, password, userAddress, orders.
* Phương thức: register(), login(), updateProfile(), deleteAccount(), viewOrderHistory().
* UserAddress (Địa chỉ người dùng):
* Thuộc tính: userAddressId, user, homeAddress, province, district, ward.
* Phương thức: createAddress(), getAddressByUser(), updateAddress(), deleteAddress().
* Products (Sản phẩm):
* Thuộc tính: productId, category, productName, price, imageUrl, status.
* Phương thức: addProduct(), getProductById(), getAllProducts(), updateProduct(), deleteProduct(), isAvailable().
* Orders (Đơn hàng):
* Thuộc tính: orderId, user, totalAmount, status, orderDetails, payment.
* Phương thức: createOrder(), getOrderById(), updateStatus(), cancelOrder(), calculateTotalAmount().
* OrderDetail (Chi tiết đơn hàng):
* Thuộc tính: orderDetailId, order, product, productName, productPrice, quantity.
* Phương thức: addOrderDetail(), getSubtotal(), updateQuantity(), removeDetail().
* Payments (Thanh toán):
* Thuộc tính: paymentId, order, method, paymentAmount.
* Phương thức: makePayment(), getPaymentByOrder(), confirmPayment().

1. **Mối quan hệ giữa các lớp.**

* Quan hệ kết hợp : Mối quan hệ “có” hoặc “liên kết” giữa các lớp.
* **Province** và **District** có mối quan hệ kết hợp (một Province có thể có nhiều District).
* **District** và **Wards** có mối quan hệ kết hợp (một District có thể có nhiều Wards).
* **User** và **UserAddress** có mối quan hệ kết hợp (một User có thể có một hoặc nhiều UserAddress).
* **UserAddress** và **Province/District/Wards** có mối quan hệ kết hợp (một UserAddress thuộc về một Province, một District và một Wards).
* **User** và **Orders** có mối quan hệ kết hợp (một User có thể có nhiều Orders).
* **Orders** và **OrderDetail** có mối quan hệ kết hợp (một Orders có thể có nhiều OrderDetail).
* **OrderDetail** và **Products** có mối quan hệ kết hợp (một OrderDetail liên kết với một Products).
* **Orders** và **Payments** có mối quan hệ kết hợp (một Orders có thể có một Payments).
* Quan hệ thành phần : Mối quan hệ “thành phần” thể hiện phụ thuộc tồn tại – khi đối tượng cha bị xóa, đối tượng con cũng bị xóa.
* **Orders** và **OrderDetail** có quan hệ thành phần (một OrderDetail chỉ tồn tại khi Orders còn tồn tại).
* **Orders** và **Payments** có quan hệ thành phần (Payments chỉ tồn tại khi đơn hàng tồn tại).

1. **Áp dụng các nguyên lý Hướng đối tượng.**

* Đóng gói (Encapsulation): Mỗi lớp sẽ gói gọn dữ liệu (thuộc tính) và các phương thức xử lý dữ liệu đó. Dữ liệu sẽ được bảo vệ, chỉ có thể truy cập thông qua các phương thức công khai (public methods).
* Kế thừa (Inheritance): Cho phép một lớp (lớp con) thừa hưởng thuộc tính và phương thức từ một lớp khác (lớp cha).
* Đa hình (Polymorphism): Cho phép các đối tượng của các lớp khác nhau có thể phản ứng với cùng một thông điệp theo các cách riêng.
* Trừu tượng hóa (Abstraction): Tập trung vào những đặc điểm cốt lõi của một đối tượng và bỏ qua các chi tiết không cần thiết.

1. **Data flow.**
2. **Sơ đồ quan hệ thực thể (ERD - Entity-Relationship Diagram).**

* Sơ đồ ERD thể hiện cấu trúc logic của cơ sở dữ liệu và mối quan hệ giữa các thực thể (bảng). Đây là một phần rất quan trọng trong thiết kế dữ liệu vì nó trực tiếp phản ánh nghiệp vụ của doanh nghiệp.
* Nguoidung (user) : Lưu trữ thông tin của khách hàng.
* Diachi (useraddress) : Lưu trữ địa chỉ của khách hàng.
* Sanpham (products) : Lưu trữ thông tin chung về sản phẩm (tên, mô tả, giá).
* Donhang (orders) : Đại diện cho một đơn hàng, có quan hệ 1-nhiều với KhachHang (một khách hàng có thể có nhiều đơn hàng)
* Thongtindonhang (order\_detail) : Lưu trữ thông tin đơn hàng ( mã sản phẩm, tên sản phẩm, giá).
* Thanhtoan (payments) : Thanh toán.

A diagram of a computer

AI-generated content may be incorrect.

Hình 9.1 : Sơ đồ quan hệ thực thể

1. **Sơ đồ luồng dữ liệu (DFD - Data Flow Diagram).**

* Sơ đồ DFD mô tả cách dữ liệu di chuyển trong hệ thống, từ khi bắt đầu (tác nhân) đến khi kết thúc. DFD giúp hình dung cách các quy trình xử lý dữ liệu với nhau. Dưới đây là DFD cấp 1 cho nghiệp vụ "Đặt hàng".
* Tác nhân :
* Khách hàng : Người dùng cuối.
* Tiến trình :
* 1.0 - Duyệt & Tìm kiếm: Khách hàng nhập thông tin tìm kiếm. Dữ liệu này được xử lý để truy xuất thông tin sản phẩm từ Kho dữ liệu Sản phẩm & Tồn kho.
* 2.0 - Quản lý Giỏ hàng: Dữ liệu sản phẩm được thêm vào Giỏ hàng. Giỏ hàng có thể được lưu tạm thời.
* 3.0 - Xử lý Thanh toán: Dữ liệu đơn hàng được xử lý, bao gồm xác thực thông tin và giao tiếp với Tiến trình Thanh toán Bên ngoài (cổng thanh toán).
* 4.0 - Tạo & Cập nhật Đơn hàng: Dữ liệu đơn hàng được lưu vào Kho dữ liệu Đơn hàng. Tiến trình này cũng gửi dữ liệu đến Kho dữ liệu Sản phẩm & Tồn kho để cập nhật trạng thái(còn hàng/hết hàng).
* Kho dữ liệu :
* D1 - Sản phẩm & Tồn kho: Lưu trữ thông tin sản phẩm và trạng thái tồn kho.
* D2 - Đơn hàng: Lưu trữ thông tin đơn hàng và chi tiết.

1. **Sơ đồ lớp và cấu trúc cơ sở dữ liệu.**
2. **Sơ đồ lớp (Class diagram ).**

* Sơ đồ lớp thể hiện các lớp chính của hệ thống, các thuộc tính (attribute), phương thức (method) và mối quan hệ giữa chúng. Nó là một phần quan trọng trong thiết kế hướng đối tượng, giúp lập trình viên hình dung cấu trúc mã nguồn.

A diagram of a computer

AI-generated content may be incorrect.

Hình 10.1.1

* User (Người dùng):
* Thuộc tính: userId, name, email, phone, password, userAddress, orders.
* Phương thức: register(), login(), updateProfile(), deleteAccount(), viewOrderHistory().
* UserAddress (Địa chỉ người dùng):
* Thuộc tính: userAddressId, user, homeAddress, province, district, ward.
* Phương thức: createAddress(), getAddressByUser(), updateAddress(), deleteAddress().
* Products (Sản phẩm):
* Thuộc tính: productId, category, productName, price, imageUrl, status.
* Phương thức: addProduct(), getProductById(), getAllProducts(), updateProduct(), deleteProduct(), isAvailable().
* Orders (Đơn hàng):
* Thuộc tính: orderId, user, totalAmount, status, orderDetails, payment.
* Phương thức: createOrder(), getOrderById(), updateStatus(), cancelOrder(), calculateTotalAmount().
* OrderDetail (Chi tiết đơn hàng):
* Thuộc tính: orderDetailId, order, product, productName, productPrice, quantity.
* Phương thức: addOrderDetail(), getSubtotal(), updateQuantity(), removeDetail().
* Payments (Thanh toán):
* Thuộc tính: paymentId, order, method, paymentAmount.
* Phương thức: makePayment(), getPaymentByOrder(), confirmPayment().
* Province (Tỉnh):
* Thuộc tính: provinceId, name.
* Phương thức: createProvince(), getProvinceById(), getAllProvinces(), updateProvince(), deleteProvince().
* District (Quận/Huyện):
* Thuộc tính: districtId, province, name.
* Phương thức: createDistrict(), getDistrictById(), getDistrictsByProvince(), updateDistrict(), deleteDistrict().
* Wards (Phường/Xã):
* Thuộc tính: wardsId, district, name.
* Phương thức: createWard(), getWardById(), getWardsByDistrict(), updateWard(), deleteWard().
* Mối quan hệ :
* Province – District: Một tỉnh có nhiều quận/huyện (1–n).
* District – Wards: Một quận/huyện có nhiều phường/xã (1–n).
* User – UserAddress: Mỗi người dùng có một địa chỉ (1–1).
* User – Orders: Một người dùng có nhiều đơn hàng (1–n).
* Orders – OrderDetail: Một đơn hàng có nhiều chi tiết sản phẩm (1–n).
* Orders – Payments: Mỗi đơn hàng có một thanh toán (1–1).
* OrderDetail – Products: Mỗi chi tiết đơn hàng tương ứng một sản phẩm (1–1).
* UserAddress – Province/District/Wards: Mỗi địa chỉ người dùng thuộc về một tỉnh, một quận và một phường (n–1).

1. **Cấu trúc cơ sở dữ liệu.**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 10.2.1

* 1. **Bảng chính.**
* User (Người dùng):
* user\_id (**PK**, int, auto\_increment) : Mã định danh người dùng.
* name (varchar) : Họ tên người dùng.
* email (varchar) : Địa chỉ email của người dùng.
* phone (varchar) : Số điện thoại của người dùng.
* password (varchar) : Mật khẩu đăng nhập (được mã hóa).
* userAddress\_id (**FK**, int) : Liên kết đến địa chỉ của người dùng (bảng useraddress).
* UserAddress (Địa chỉ người dùng):
* userAddress\_id (**PK**, int, auto\_increment): Mã định danh của địa chỉ.
* user\_id (**FK**, int): Mã người dùng sở hữu địa chỉ.
* homeAddress (varchar): Địa chỉ cụ thể (số nhà, tên đường).
* province\_id (**FK**, int): Mã tỉnh của địa chỉ.
* district\_id (**FK**, int): Mã quận/huyện của địa chỉ.
* ward\_id (**FK**, int): Mã phường/xã của địa chỉ.
* Products (Sản phẩm):
* product\_id (**PK**, int, auto\_increment): Mã định danh sản phẩm.
* category (varchar): Danh mục hoặc loại sản phẩm.
* product\_name (varchar): Tên sản phẩm.
* price (decimal(12,3)): Giá sản phẩm.
* image\_url (varchar): Đường dẫn ảnh của sản phẩm.
* status (enum): Trạng thái sản phẩm (AVAILABLE hoặc NOT\_AVAILABLE).
* Orders (Đơn hàng):
* order\_id (**PK**, int, auto\_increment): Mã định danh đơn hàng.
* user\_id (**FK**, int): Mã người dùng đặt đơn hàng.
* total\_amount (decimal(12,2)): Tổng tiền của đơn hàng.
* status (enum): Trạng thái đơn hàng (NEW, PAID, CANCELLED).
* Order\_Detail (Chi tiết đơn hàng):
* order\_detail\_id (**PK**, int, auto\_increment): Mã định danh chi tiết đơn hàng.
* order\_id (**FK**, int): Mã đơn hàng chứa chi tiết này.
* product\_id (**FK**, int): Mã sản phẩm trong đơn hàng.
* product\_name (varchar): Tên sản phẩm tại thời điểm đặt hàng.
* product\_price (decimal(12,3)): Giá sản phẩm tại thời điểm đặt hàng.
* quantity (int): Số lượng sản phẩm trong đơn hàng.
* Payments (Thanh toán):
* payment\_id (**PK**, int, auto\_increment): Mã định danh giao dịch thanh toán.
* order\_id (**FK**, int): Mã đơn hàng được thanh toán.
* method (enum): Phương thức thanh toán (CASH, MOMO, CARD).
* payment\_amount (decimal(12,3)): Số tiền thanh toán.
  1. **Bảng phụ.**
* Province (Tỉnh):
* province\_id (**PK**, int, auto\_increment): Mã định danh của tỉnh.
* name (varchar): Tên của tỉnh/thành phố trực thuộc trung ương.
* District (Quận/Huyện):
* district\_id (**PK**, int, auto\_increment): Mã định danh của quận/huyện.
* province\_id (**FK**, int): Mã tỉnh mà quận/huyện thuộc về.
* name (varchar): Tên quận/huyện.
* Wards (Phường/Xã):
* wards\_id (**PK**, int, auto\_increment): Mã định danh của phường/xã.
* district\_id (**FK**, int): Mã quận/huyện mà phường/xã thuộc về.
* name (varchar): Tên phường/xã.

1. **Cấu trúc quá trình thiết kế.**
2. **Thiết kế tổng quan.**

* Thiết kế tổng quan mô tả kiến trúc tổng thể của hệ thống, các module chính và cách chúng tương tác với nhau. Hệ thống sẽ được thiết kế theo kiến trúc microservices hoặc Service Oriented Architecture (SOA) để đảm bảo tính linh hoạt, khả năng mở rộng và dễ bảo trì.
* Các module chính :
* Module người dùng : Đăng ký, đăng nhập, đăng xuất, Quản lý hồ sơ cá nhân (tên, email, SĐT, địa chỉ), Cập nhật địa chỉ giao hàng, Xem lịch sử mua hàng.
* Module sản phẩm : Quản lý danh sách sản phẩm, Hiển thị chi tiết sản phẩm, Tìm kiếm, quản lý tồn kho.
* Module giỏ hàng và đặt hàng : Thêm/xóa/cập nhật sản phẩm trong giỏ hàng, tính tổng tiền tự động, tạo đơn hàng, quản lý trạng thái đơn hàng.
* Module thanh toán : Xử lý giao dịch thanh toán, tích hợp cổng thanh toán, lưu thông tin vào cơ sở dữ liệu.
* Module quản lý : Quản lý sản phẩm, quản lý đơn hàng, quản lý tài khoản, quản lý tồn kho, xem doanh thu.
* Module báo cáo : Thống kê doanh thu, báo cáo số lượng đơn hàng bán ra.
* Tương tác giữa các module: Các module sẽ giao tiếp với nhau thông qua API (Application Programming Interface), đảm bảo sự độc lập và linh hoạt trong quá trình phát triển.

1. **Thiết kế kiến trúc.**

* Hệ thống sẽ được xây dựng trên mô hình kiến trúc 3 lớp (3-tier architecture), giúp tách biệt các chức năng thành các tầng độc lập để dễ phát triển và bảo trì.
* Tầng trình bày (Presentation Tier):
* Mục đích: Giao tiếp với người dùng cuối.
* Công nghệ: Giao diện web được xây dựng bằng các framework Spring boot.
* Chức năng: Hiển thị trang chủ, danh sách sản phẩm, trang chi tiết, giỏ hàng, trang thanh toán và trang quản trị.
* Tầng nghiệp vụ (Business Logic Tier):
* Mục đích: Xử lý tất cả các nghiệp vụ của hệ thống.
* Công nghệ: Sử dụng một framework Spring boot.
* Chức năng: Chứa logic nghiệp vụ phức tạp, như xử lý các yêu cầu từ tầng trình bày, tương tác với tầng dữ liệu, và gọi các dịch vụ bên ngoài (như cổng thanh toán, API vận chuyển).
* Tầng dữ liệu (Data Tier):
* Mục đích: Lưu trữ tất cả dữ liệu của hệ thống.
* Công nghệ: Sử dụng hệ quản trị cơ sở dữ liệu quan hệ (Relational Database) như MySQL.
* Cấu trúc dữ liệu: Cơ sở dữ liệu sẽ được thiết kế với các bảng chính để lưu trữ thông tin về Sản phẩm, Tồn kho (Inventory), Đơn hàng (Orders), Khách hàng (Customers).

1. **Thiết kế chức năng.**

* Thiết kế chức năng chi tiết từng module, xác định các chức năng chính, các màn hình và các luồng dữ liệu.
  1. **Thiết kế cho Module Sản phẩm.**
* Giao diện quản trị :
* Màn hình cho phép **Admin** thêm, sửa, xóa thông tin sản phẩm; nhập tên, danh mục, giá và tải lên hình ảnh minh họa cho sản phẩm.
* Cơ sở dữ liệu: Sử dụng bảng **products** gồm các trường:
* product\_id: Mã sản phẩm (khóa chính).
* category: Danh mục sản phẩm.
* product\_name: Tên sản phẩm.
* price: Giá sản phẩm.
* image\_url: Đường dẫn hình ảnh sản phẩm.
* status: Trạng thái sản phẩm (còn hàng/hết hàng).
* Luồng dữ liệu :
* Khi **Admin** thêm hoặc cập nhật sản phẩm, dữ liệu sẽ được ghi hoặc cập nhật vào bảng products.
* Khi **khách hàng** đặt hàng, thông tin sản phẩm được lấy từ bảng products và lưu vào bảng order\_detail để phục vụ quản lý chi tiết đơn hàng.
  1. **Thiết kế cho Module Khách hàng (User).**
* Giao diện người dùng: Cho phép khách hàng đăng ký, đăng nhập, cập nhật thông tin cá nhân, và quản lý tài khoản (họ tên, email, số điện thoại, mật khẩu).
* Cơ sở dữ liệu : Sử dụng bảng **user** gồm các trường:
* user\_id: Mã khách hàng (khóa chính).
* name: Họ tên khách hàng.
* email: Email (dùng để đăng nhập).
* phone: Số điện thoại liên hệ.
* password: Mật khẩu (được mã hóa).
* userAddress\_id: Mã địa chỉ (liên kết với bảng useraddress).
* Luồng dữ liệu:
* Khi khách hàng đăng ký tài khoản, thông tin được lưu vào bảng user.
* Khi đăng nhập, hệ thống kiểm tra thông tin email và mật khẩu.
* Khi cập nhật thông tin, dữ liệu được ghi đè vào dòng tương ứng trong bảng user.
  1. **Thiết kế cho Module Địa chỉ khách hàng (UserAddress).**
* Giao diện người dùng: Cho phép khách hàng quản lý địa chỉ giao hàng, bao gồm nhập địa chỉ cụ thể, chọn tỉnh, quận, phường.
* Cơ sở dữ liệu : Sử dụng bảng useraddress gồm các trường:
* userAddress\_id: Mã địa chỉ (khóa chính).
* user\_id: Mã người dùng (liên kết với bảng user).
* homeAddress: Địa chỉ cụ thể (số nhà, tên đường).
* province\_id: Mã tỉnh (liên kết với bảng province).
* district\_id: Mã quận (liên kết với bảng district).
* ward\_id: Mã phường (liên kết với bảng wards).
* Luồng dữ liệu:
* Khi khách hàng thêm hoặc chỉnh sửa địa chỉ, dữ liệu được ghi vào bảng useraddress.
* Khi đặt hàng, hệ thống sẽ lấy thông tin địa chỉ giao hàng từ bảng này để hiển thị và lưu vào đơn hàng.
  1. **Thiết kế cho Module Đơn hàng (Orders).**
* Giao diện người dùng:
* Hiển thị danh sách đơn hàng của khách hàng, trạng thái (NEW, PAID) và tổng tiền.
* Cho phép người dùng xem chi tiết đơn hàng và hủy đơn nếu chưa thanh toán.
* Cơ sở dữ liệu: Sử dụng bảng **orders** gồm các trường:
* order\_id: Mã đơn hàng (khóa chính).
* user\_id: Mã khách hàng (liên kết bảng user).
* total\_amount: Tổng tiền đơn hàng.
* status: Trạng thái đơn hàng (NEW, PAID).
* Luồng dữ liệu:
* Khi khách hàng đặt hàng, hệ thống sẽ tạo bản ghi mới trong bảng orders.
* Khi đơn hàng được thanh toán, trạng thái được cập nhật sang PAID.
  1. **Thiết kế cho Module Chi tiết đơn hàng (OrderDetail).**
* Giao diện người dùng : Hiển thị chi tiết từng sản phẩm trong đơn hàng bao gồm tên, giá, số lượng, và tổng tiền.
* Cơ sở dữ liệu : Sử dụng bảng **order\_detail** gồm các trường:
* order\_detail\_id: Mã chi tiết đơn hàng (khóa chính).
* order\_id: Mã đơn hàng (liên kết bảng orders).
* product\_id: Mã sản phẩm (liên kết bảng products).
* product\_name: Tên sản phẩm tại thời điểm đặt hàng.
* product\_price: Giá sản phẩm tại thời điểm đặt hàng.
* quantity: Số lượng mua.
* Luồng dữ liệu:
* Khi đơn hàng được tạo, hệ thống tự động ghi chi tiết từng sản phẩm vào bảng order\_detail.
* Khi đơn hàng bị hủy, các chi tiết trong order\_detail tương ứng cũng bị xóa.
  1. **Thiết kế cho Module Thanh toán (Payments).**
* Giao diện người dùng:
* Cho phép khách hàng chọn phương thức thanh toán: tiền mặt, thẻ, hoặc ví điện tử (MOMO).
* Hiển thị tổng tiền và xác nhận khi giao dịch thành công.
* Cơ sở dữ liệu : Sử dụng bảng **payments** gồm các trường:
* payment\_id: Mã thanh toán (khóa chính).
* order\_id: Mã đơn hàng (liên kết bảng orders).
* method: Phương thức thanh toán (CASH, MOMO, CARD).
* payment\_amount: Số tiền thanh toán.
* Luồng dữ liệu :
* Khi khách hàng thực hiện thanh toán, hệ thống ghi thông tin vào bảng payments.
* Khi thanh toán thành công, trạng thái đơn hàng trong bảng orders được cập nhật thành PAID.

1. **Thiết kế chương trình.**

* Thiết kế chương trình tập trung vào chi tiết các thuật toán, cấu trúc dữ liệu và logic của từng chức năng nhỏ bên trong hệ thống.

**Thiết kế cho chức năng tìm kiếm sản phẩm :**

* Thuật toán:
* Hệ thống sử dụng truy vấn SQL kết hợp với các bộ lọc để tối ưu tốc độ tìm kiếm.
* Người dùng có thể tìm sản phẩm theo tên.
* Khi người dùng nhập từ khóa, hệ thống thực hiện tìm kiếm gần đúng, cho phép gợi ý kết quả tương tự.
* Các lớp (Classes):
* **Product**: đại diện cho sản phẩm với các thuộc tính như productId, category, productName, price, imageUrl, status.
* **ProductRepository**: chịu trách nhiệm truy vấn và tương tác với cơ sở dữ liệu (tìm kiếm theo tên, lọc theo danh mục, giá, v.v.).
* **SearchService**: lớp trung gian xử lý logic tìm kiếm. Nhận tham số đầu vào từ người dùng (từ khóa, giá, loại sản phẩm), gọi ProductRepository để lấy danh sách kết quả, và trả về cho tầng giao diện.

**Thiết kế cho chức năng xử lý đơn hàng :**

* Luồng xử lý: Khi khách hàng xác nhận đặt hàng, hệ thống thực hiện một giao dịch (transaction) để đảm bảo tính toàn vẹn dữ liệu.
* Các bước trong giao dịch gồm :
* **Kiểm tra tồn kho**: hệ thống gọi đến InventoryService để đảm bảo các sản phẩm trong giỏ còn đủ hàng.
* **Trừ tồn kho**: nếu hợp lệ, hệ thống giảm số lượng trong bảng products.
* **Tạo bản ghi đơn hàng**: dữ liệu được lưu vào bảng orders và chi tiết từng sản phẩm được ghi vào order\_detail.
* **Cập nhật trạng thái đơn hàng**: mặc định là NEW, sau thanh toán chuyển PAID, và tiếp tục cập nhật theo tiến trình (SHIPPING, DELIVERED, CANCELLED).
* Trạng thái đơn hàng: Các trạng thái được định nghĩa rõ ràng trong hệ thống : NEW, PAID, SHIPPING, CANCELLED.
* Mỗi thay đổi trạng thái đều được ghi log để theo dõi lịch sử đơn hàng.

1. **Thiết kế tổng quan.**
2. **Phân tách kiến trúc tổng thể.**

* Hệ thống sẽ được xây dựng dựa trên kiến trúc kiến trúc microservices, hoặc một biến thể là kiến trúc hướng dịch vụ (Service-Oriented Architecture - SOA).

A diagram of a computer service

AI-generated content may be incorrect.

* Module Người dùng & Tài khoản (User & Account Module): Xử lý mọi thứ liên quan đến người dùng, bao gồm đăng ký, đăng nhập, quản lý thông tin cá nhân và địa chỉ. Module này sẽ độc lập với các nghiệp vụ khác, giúp tăng cường bảo mật và dễ dàng mở rộng nếu cần thêm các phương thức xác thực khác trong tương lai.
* Module Sản phẩm (Product): Đây là module phức tạp nhất, chịu trách nhiệm quản lý thông tin sản phẩm (tên, mô tả, hình ảnh), các biến thể (size). Mọi yêu cầu liên quan đến sản phẩm từ các module khác đều phải thông qua API của module này.
* Module Giỏ hàng & Đặt hàng : Quản lý giỏ hàng tạm thời của khách hàng, tạo và lưu trữ các đơn hàng mới. Module này sẽ giao tiếp với Module Sản phẩm để kiểm tra tồn kho và lấy thông tin chi tiết trước khi hoàn tất đơn hàng.
* Module Thanh toán : Xử lý quy trình thanh toán và tích hợp với các dịch vụ bên ngoài.
* Module Quản lý (Admin Module): Cung cấp giao diện và chức năng cho quản trị viên, bao gồm quản lý tất cả các module khác. Nó sẽ là một hệ thống nội bộ, không công khai với người dùng cuối.

1. **Sơ đồ tương tác cấp cao.**

* Mô hình dưới đây thể hiện cách các module chính tương tác với nhau để hoàn thành một nghiệp vụ:
  1. **Luồng đặt hàng của khách hàng.**
* Khách hàng truy cập trang web (giao diện người dùng).
* Giao diện gọi API từ Module Sản phẩm để lấy danh sách sản phẩm.
* Khách hàng thêm sản phẩm vào giỏ hàng, giao diện gọi API của Module Giỏ hàng.
* Khi khách hàng thanh toán, Module Giỏ hàng tạo một đơn hàng mới và gọi API của Module Thanh toán.
* Module Thanh toán xử lý giao dịch qua cổng thanh toán bên ngoài và trả kết quả về.
* Nếu giao dịch thành công, Module Giỏ hàng gọi API của Module Sản phẩm để cập nhật trạng thái tồn kho.
  1. **Luồng xử lý đơn hàng của Admin.**
* Admin vào trang quản trị (giao diện Admin).
* Giao diện gọi API từ Module Giỏ hàng để lấy danh sách đơn hàng.
* Admin xác nhận đơn hàng, giao diện gọi API của Module Quản lý.
* Module Quản lý gọi API của Module Thanh toán (hoàn tiền) để hoàn tất quy trình.

1. **Công nghệ dự kiến.**

* Mặc dù không đi sâu vào chi tiết, thiết kế tổng quan cũng nên định hình công nghệ cốt lõi.
* Tầng giao diện (Front-end) : framework Spring Boot , HTML, CSS, JS.
* Tầng back-end : Ngôn ngữ lập trình Java.
* Tầng dữ liệu : Cơ sở dữ liệu MySql.

1. **Thiết kế kiến trúc.**
2. **Mô hình kiến trúc 3 tầng.**

* Hệ thống sẽ được xây dựng trên mô hình kiến trúc 3-tầng để tách biệt các chức năng thành các tầng độc lập:
* Tầng Trình bày (Presentation Tier): Giao diện người dùng cuối.
* Tầng Nghiệp vụ (Business Logic Tier): Xử lý các nghiệp vụ cốt lõi.
* Tầng Dữ liệu (Data Tier): Lưu trữ và quản lý dữ liệu.
* Mỗi tầng có thể được triển khai độc lập, cho phép dễ dàng nâng cấp hoặc thay thế công nghệ mà không ảnh hưởng đến các tầng khác.

1. **Chi tiết các tầng và thành phần.**
   1. **Tầng trình bày.**

* Tầng này chịu trách nhiệm cho mọi thứ người dùng thấy và tương tác. Nó sẽ được chia thành hai phần riêng biệt:
* Front-end khách hàng :
* Mục đích: Cung cấp trải nghiệm mua sắm mượt mà cho người dùng cuối.
* Công nghệ : html, css, js.
* Tính năng chính: Hiển thị sản phẩm, bộ lọc tìm kiếm, giỏ hàng, trang thanh toán, lịch sử đơn hàng. Giao diện phải được thiết kế responsive để hiển thị tốt trên cả máy tính và thiết bị di động.
* Front-end quản trị :
* Mục đích: Cung cấp giao diện trực quan cho quản trị viên.
* Công nghệ : html, css, js.
* Tính năng chính: Dashboard quản lý sản phẩm, tồn kho, đơn hàng, khách hàng, doanh thu.
  1. **Tầng nghiệp vụ.**
* Đây là "bộ não" của hệ thống, nơi xử lý tất cả các nghiệp vụ phức tạp. Tầng này sẽ được thiết kế theo kiến trúc microservices để xử lý các nghiệp vụ chuyên biệt.
* Microservices :
* Product Service: Quản lý tất cả các nghiệp vụ liên quan đến sản phẩm và biến thể. Dịch vụ này sẽ có API riêng để thêm, sửa, xóa sản phẩm, quản lý hình ảnh và các thuộc tính.
* Order Service: Xử lý việc tạo, cập nhật và quản lý trạng thái của các đơn hàng.
* Payment Service: Dịch vụ này sẽ là trung gian giữa Order Service và các cổng thanh toán bên ngoài. Nó sẽ xử lý logic tạo giao dịch, xác thực thanh toán và hoàn tiền.
  1. **Tầng dữ liệu.**
* Tầng này chịu trách nhiệm lưu trữ và quản lý dữ liệu.
* Cơ sở dữ liệu chính :
* Loại: Sử dụng cơ sở dữ liệu quan hệ (Relational Database) MySQL.
* Lý do: Đảm bảo tính toàn vẹn và nhất quán của dữ liệu, đặc biệt quan trọng cho các giao dịch liên quan đến đơn hàng .
* Cấu trúc bảng chính :
* **User:** Lưu trữ thông tin tài khoản khách hàng (họ tên, email, số điện thoại, mật khẩu, địa chỉ).
* **UserAddress:** Quản lý địa chỉ giao hàng, liên kết với các bảng địa lý (Province, District, Ward).
* **Products:** Thông tin chung về sản phẩm (tên, loại, giá, hình ảnh, trạng thái).
* **OrderDetail:** Liên kết các sản phẩm trong một đơn hàng (tên, giá, số lượng).
* **Orders:** Lưu thông tin đơn hàng của khách hàng (người mua, tổng tiền, trạng thái).
* **Payments:** Quản lý giao dịch thanh toán, phương thức và số tiền.

1. **Thiết kế tích hợp.**

* Cổng thanh toán : Hệ thống sẽ tích hợp với các cổng thanh toán phổ biến tại Việt Nam như VNPAY, Momo.

1. **Thiết kế chức năng.**
2. **Module khách hàng ( User module ).**

* Đăng ký & Đăng nhập:
* Đăng ký: Cho phép người dùng đăng ký tài khoản bằng email/số điện thoại và mật khẩu. Hệ thống kiểm tra trùng lặp thông tin và lưu tài khoản mới vào cơ sở dữ liệu.
* Đăng nhập: Cho phép người dùng đăng nhập bằng email và mật khẩu đã đăng ký. Hệ thống xác thực thông tin đăng nhập và cấp quyền truy cập. Có cơ chế báo lỗi khi nhập sai mật khẩu hoặc tài khoản không tồn tại.
* Quản lý hồ sơ:
* Cập nhật thông tin cá nhân: Cho phép người dùng cập nhật họ tên, số điện thoại, email, mật khẩu. Dữ liệu mới được lưu lại và sử dụng cho các đơn hàng tiếp theo.
* Quản lý địa chỉ: Người dùng có thể thêm, sửa hoặc xóa địa chỉ giao hàng. Mỗi địa chỉ bao gồm: tỉnh, quận, phường, địa chỉ cụ thể.
* Tìm kiếm sản phẩm: Cho phép người dùng tìm sản phẩm theo tên. Hệ thống hỗ trợ tìm kiếm nhanh bằng từ khóa, hiển thị kết quả phù hợp trong thời gian ngắn.
* Giỏ hàng:
* Thêm sản phẩm: Người dùng có thể thêm sản phẩm vào giỏ hàng từ trang chi tiết sản phẩm. Hệ thống tự động tính tổng tiền theo số lượng và giá.
* Cập nhật giỏ hàng: Cho phép thay đổi số lượng, xóa sản phẩm hoặc tiếp tục mua thêm. Giỏ hàng được cập nhật theo thời gian thực.
* Thanh toán:
* Phương thức thanh toán: Thanh toán bằng ví điện tử (Momo, ZaloPay)
* Xác nhận đơn hàng: Sau khi chọn phương thức thanh toán, hệ thống tạo bản ghi đơn hàng và gửi thông báo xác nhận.
* Theo dõi đơn hàng:
* Xem lịch sử: Cho phép khách hàng xem lại danh sách đơn hàng đã đặt trước đó.
* Xem chi tiết: Hiển thị chi tiết từng đơn hàng, bao gồm tên sản phẩm, số lượng, tổng tiền, trạng thái.

1. **Module quản trị viên ( Admin module ).**

* Quản lý sản phẩm (Admin):
* Thêm/Sửa/Xóa sản phẩm: Admin có thể thêm sản phẩm mới, chỉnh sửa thông tin (giá, mô tả, hình ảnh) hoặc xóa sản phẩm.
* Quản lý tồn kho: Hệ thống tự động trừ tồn kho khi có đơn hàng thành công. Admin có thể cập nhật thủ công khi nhập thêm hàng.
* Quản lý đơn hàng (Admin):
* Xem danh sách đơn hàng: Admin xem được toàn bộ đơn hàng.
* Thêm/sửa/xóa đơn hàng.
* Quản lý tài khoản khách hàng (Admin):
* Xem danh sách tài khoản: Hiển thị thông tin người dùng (tên, số điện thoại, email, địa chỉ).
* Theo dõi đơn hàng:
* Xem chi tiết: Hiển thị chi tiết từng đơn hàng, bao gồm tên sản phẩm, số lượng, tổng tiền, trạng thái.
* Theo dõi trạng thái: Trạng thái được cập nhật theo tiến trình giao hàng: NEW/PAID.
* Thống kê doanh thu:
* Báo cáo doanh thu: Hệ thống tổng hợp doanh thu theo ngày, tháng, năm, hiển thị dạng bảng hoặc biểu đồ.

1. **Tích hợp bên thứ ba.**

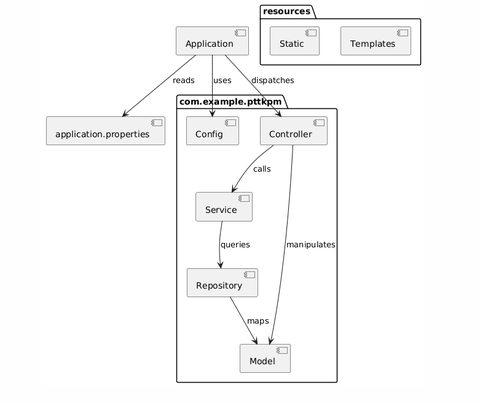
* Cổng thanh toán: Tích hợp với API của các cổng thanh toán để xử lý các giao dịch online.

1. **Thiết kế chương trình.**
2. **Cấu trúc dữ liệu và mô hình hóa.**

* Thiết kế chương trình bắt đầu với việc định nghĩa các thực thể (entities), các lớp (classes) và mối quan hệ giữa chúng, sử dụng các nguyên lý lập trình hướng đối tượng (Object-Oriented Programming).
* Lớp user : Đại diện cho 1 khách hàng.
* Thuộc tính : user\_id, name, email, phone, password, userAddress\_id.
* Phương thức : getPassword(), getEmail()
* Lớp useraddress : Đại diện cho thông tin, địa chỉ của khách hàng.
* Thuộc tính : userAddress\_id, user\_id, homeAddress, province\_id, district\_id, ward\_id.
* Lớp products : Đại diện cho 1 sản phẩm chung.
* Thuộc tính : product\_id, category, product\_name, price, image\_url, status ENUM.
* Phương thức : getProductName(), getCategory(), getPrice().
* Lớp orders : Đại diện cho 1 đơn hàng.
* Thuộc tính : order\_id, user\_id, total\_amount, status ENUM.
* Phương thức :
* Lớp order\_detail : Thông tin đơn hàng.
* Thuộc tính : order\_detail\_id, order\_id, product\_id, product\_name, product\_price, quanlity.
* Lớp payments : Thanh toán.
* Thuộc tính : payment\_id, order\_id, method ENUM, payment\_amount.

1. **Thiết kế mã nguồn.**

* Mã nguồn sẽ được tổ chức theo cấu trúc module và tuân thủ các nguyên lý thiết kế phần mềm như SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) để dễ dàng bảo trì và mở rộng.
* Tổ chức thư mục (Folder Structure):
* src/models: Định nghĩa các lớp và thực thể dữ liệu.
* src/services: Chứa logic nghiệp vụ.
* src/repositories: Chứa logic truy cập cơ sở dữ liệu.
* src/controllers: Xử lý các yêu cầu HTTP từ front-end và gọi các service tương ứng.
* src/utils: Các hàm tiện ích dùng chung.



Sơ đồ cấu trúc thư mục ( Package diagram )

* Thiết kế chương trình chi tiết này đảm bảo rằng mỗi chức năng đều được suy nghĩ kỹ lưỡng từ cấp độ logic đến cấu trúc mã nguồn, từ đó tạo nền tảng vững chắc để phát triển một phần mềm chất lượng cao.

1. **Thiết kế giao diện.**

**Link figma :** [**https://www.figma.com/design/f6dQiujnYUxwEfnTLhUSb9/Diang-Drink?node-id=0-1&p=f&t=GAkm0LKkE0IR5S3A-0**](https://www.figma.com/design/f6dQiujnYUxwEfnTLhUSb9/Diang-Drink?node-id=0-1&p=f&t=GAkm0LKkE0IR5S3A-0&fbclid=IwZXh0bgNhZW0CMTAAYnJpZBExTEdyZzZUTTB2elNueTNSTwEecJBpd0-JTNlJ7j61cNYooiCh3wjW8_6H5kWwldYbyRlIZXPZJst8514l9z8_aem_H3BU0RxzI_7vggHlkO9kcA)

1. **Mục tiêu thiết kế giao diện.**

* Giao diện người dùng (UI) của hệ thống bán đồ uống online được thiết kế nhằm:
* Mang lại trải nghiệm thân thiện, dễ sử dụng cho người dùng ở mọi lứa tuổi.
* Tối giản số bước thao tác, giúp khách hàng có thể đặt hàng nhanh nhất có thể.
* Thống nhất bố cục và màu sắc, thể hiện phong cách thương hiệu của cửa hàng đồ uống.
* Hỗ trợ hiển thị tốt trên mọi thiết bị (máy tính, điện thoại, máy tính bảng).

1. **Nguyên tắc thiết kế chung.**

* Đơn giản (Simplicity): Chỉ giữ lại các thành phần cần thiết cho từng màn hình, tránh gây rối mắt.
* Nhất quán (Consistency): Sử dụng cùng phong cách màu sắc, font chữ và biểu tượng trên toàn bộ trang.
* Trực quan (Intuitive): Các nút, biểu tượng, và thao tác đều quen thuộc và dễ hiểu.
* Phản hồi (Feedback): Mỗi hành động người dùng (như thêm sản phẩm, thanh toán) đều có thông báo rõ ràng.
* Khả năng phản hồi (Responsive): Giao diện tự động điều chỉnh phù hợp với kích thước màn hình.

1. **Thiết kế giao diện cho khách hàng.**

* Trang chủ (Homepage) :
* Thanh điều hướng (Navigation Bar) gồm: Trang chủ, Menu, Giỏ hàng, Liên hệ.
* Thanh tìm kiếm nhanh để tra cứu đồ uống theo tên.
* Trang chi tiết sản phẩm (Product detail page) :
* Ảnh lớn của sản phẩm.
* Thông tin chi tiết: Tình trạng, dung tích, giá.
* Nút “Thêm vào giỏ hàng” .
* Có cho chọn số lượng .
* Trang giỏ hàng (Shopping Cart) :
* Danh sách các sản phẩm đã chọn, tên, số lượng, giá từng món.
* Có thể tăng giảm số lượng.
* Hiển thị tổng tiền và nút “Thanh toán”.
* Trang thanh toán (Check out page) :
* Thông tin khách hàng: địa chỉ giao hàng.
* Lựa chọn phương thức thanh toán: ví điện tử (Momo, ZaloPay).
* Thông tin sản phẩm và giỏ hàng.
* Thông báo khi thanh toán thành công.

1. **Thiết kế giao diện cho quản trị viên (Admin interface).**

* Trang thêm sản phẩm :
* Thêm tên sản phẩm, size, giá, trạng thái và hình ảnh sản phẩm.
* Trang quản lý sản phẩm (Product management) :
* Hiển thị thông tin chi tiết của sản phẩm .
* Các nút chức năng : Sửa/xóa.
* Trang quản lý khách hàng (Customer management) : Hiển thị thông tin khách hàng.
* Trạng thái đơn hàng : Hiển thị mã đơn hàng, tên khách hàng, tổng tiền đơn hàng và trạng thái đơn hàng.
* Trang doanh số : Hiển thị tổng doanh thu, số đơn hàng và có biểu đồ doanh thu theo phương thức thanh toán.

1. **Kết quả thực hiện.**
2. **Tổng quan kết quả đạt được.**

* Sau quá trình 8 tuần thực hiện theo mô hình Agile – Scrum, nhóm đã hoàn thành toàn bộ các chức năng chính của hệ thống Website bán đồ uống online, bao gồm:
* Phân tích yêu cầu và thiết kế UML (Use Case, Class, Sequence, Activity, State, Package).
* Thiết kế cơ sở dữ liệu.
* Thiết kế giao diện người dùng (UI/UX) cho cả khách hàng và quản trị viên.
* Cài đặt và lập trình các chức năng lõi của hệ thống.
* Thực hiện kiểm thử và chạy demo hệ thống.
* Hệ thống đã hoạt động ổn định.

1. **Các chức năng đã hoàn thiện.**

* Đối với khách hàng :
* Đăng ký / Đăng nhập tài khoản: Hệ thống xác thực thông tin người dùng, cho phép đăng nhập bằng email.
* Xem danh sách đồ uống: Hiển thị toàn bộ sản phẩm với hình ảnh, giá bán.
* Tìm kiếm sản phẩm theo tên.
* Xem chi tiết sản phẩm: Hiển thị thông tin chi tiết, ảnh minh họa.
* Thêm vào giỏ hàng: Quản lý số lượng, cập nhật tổng tiền theo thời gian thực.
* Thanh toán: Hỗ trợ thanh toán bằng tiền mặt, zalopay hoặc momo.
* Đối với quản trị viên :
* Quản lý sản phẩm: Thêm, sửa, xóa, cập nhật thông tin đồ uống, giá và tồn kho.
* Quản lý đơn hàng: Theo dõi danh sách đơn, thay đổi trạng thái, xử lý hủy đơn.
* Quản lý khách hàng : Xem thông tin khách hàng và xóa dữ liệu khách hàng trong database.
* Thống kê doanh thu: Hiển thị biểu đồ và số liệu doanh thu.

1. **Kết luận và đánh giá.**
2. **Kết luận chung.**

* Đề tài “Website bán đồ uống online” đã hoàn thành đúng yêu cầu của học phần Phân tích và Thiết kế Phần mềm.
* Nhóm đã áp dụng đầy đủ các bước trong quy trình phát triển phần mềm :
* Phân tích yêu cầu – Khảo sát thực tế và xác định nghiệp vụ.
* Thiết kế hệ thống UML – Use Case, Class, Sequence, Activity, State, Package.
* Thiết kế cơ sở dữ liệu – Tối ưu hóa mối quan hệ giữa các bảng.
* Thiết kế giao diện và lập trình – Hoàn thiện cả giao diện người dùng và giao diện quản trị.
* Kiểm thử và nghiệm thu – Đảm bảo hệ thống hoạt động đúng mục tiêu.
* Kết quả cho thấy nhóm có khả năng phối hợp tốt, nắm vững kỹ năng phân tích nghiệp vụ – thiết kế mô hình – lập trình ứng dụng.

1. **Những khó khăn gặp phải.**

* Khó khăn trong việc vẽ các loại sơ đồ.
* Khó chuyển yêu cầu thực tế thành phần mềm cụ thể.
* Chưa có kỹ năng phỏng vấn khách hàng.
* Không quen với các ký hiệu UML (Use Case, Class Diagram, Sequence Diagram, v.v.).
* Thiếu tư duy trừu tượng, phân tích hệ thống ở mức khái niệm.
* Khoảng cách giữa bản thiết kế và code thực tế.

1. **Đánh giá của nhóm thực hiện.**

* Hoàn thành chức năng : 100% các chức năng chính hoạt động ổn định.
* Giao diện người dùng : Dễ sử dụng, thân thiện, thẩm mỹ.
* Cấu trúc dữ liệu : Tối ưu, dễ mở rộng.
* Hiệu năng hệ thống : Ổn định, phản hồi nhanh.
* Làm việc nhóm : Hiệu quả.
* Đáp ứng yêu cầu học phần : Đầy đủ và đúng cấu trúc báo cáo.

1. **Bài học kinh nghiệm.**

* Tầm quan trọng của phân tích yêu cầu chính xác ngay từ đầu.
* Kỹ năng giao tiếp và phối hợp nhóm ảnh hưởng lớn đến tiến độ.
* Việc sử dụng mô hình UML giúp nhóm hình dung rõ hệ thống trước khi lập trình.
* Phương pháp Agile – Scrum giúp nhóm linh hoạt khi thay đổi yêu cầu.

1. **Hướng phát triển trong tương lai.**

* Xây dựng ứng dụng di động (mobile app) cho Android/iOS.
* Tích hợp hệ thống điểm thưởng – thẻ thành viên cho khách hàng thân thiết.
* Áp dụng AI gợi ý đồ uống dựa trên lịch sử đặt hàng của khách.
* Khách hàng có thể theo dõi đơn hàng.
* Hỗ trợ đánh giá và bình luận sản phẩm bằng hình ảnh.
* Đặt đồ uống qua bản đồ (Map API) để tự động định vị địa chỉ giao hàng.

# **Lời kết**

Dự án Website bán đồ uống online không chỉ giúp nhóm củng cố kiến thức về phân tích và thiết kế phần mềm, mà còn rèn luyện kỹ năng làm việc nhóm, tư duy hệ thống, và lập trình thực tế. Kết quả đạt được là minh chứng cho tinh thần học hỏi, sáng tạo và nỗ lực của cả nhóm trong suốt quá trình học phần. Chúng em xin chân thành cảm ơn !

Tài liệu tham khảo :

* [**https://fujimart.vn/**](https://fujimart.vn/?fbclid=IwZXh0bgNhZW0CMTAAYnJpZBExTEdyZzZUTTB2elNueTNSTwEeaH5bN4LLV9u6TvH4a_IVz4asIhWjsaEYlOR9OY1f6__k8IvvuX6DGgXzmog_aem_Bcg3PhGb-gjHnNZ-P31bGA)
* Sommerville, I. (2016). Software Engineering (10th Edition). Pearson Education Limited : Tài liệu nền tảng cho quy trình phát triển phần mềm, mô hình hóa yêu cầu, UML và kiến trúc hệ thống.
* Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner’s Approach (9th Edition). McGraw-Hill Education : Trình bày chi tiết về phân tích hướng đối tượng, thiết kế hệ thống, sơ đồ lớp, và quy trình phát triển phần mềm.
* Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition). Addison-Wesley : Hướng dẫn cụ thể về cách xây dựng sequence diagram, activity diagram, state diagram, class diagram, và các biểu đồ UML khác.
* Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition). Prentice Hall : Tài liệu chuyên sâu về phân tích hướng đối tượng (OOA/OOD) và cách áp dụng UML trong phát triển phần mềm thực tế.
* Kendall, K. E., & Kendall, J. E. (2019). Systems Analysis and Design (10th Edition). Pearson : Cung cấp kiến thức về Data Flow Diagram (DFD), use case modeling, và thiết kế cơ sở dữ liệu.
* Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Database System Concepts (7th Edition). McGraw-Hill Education : Tham khảo cho phần thiết kế cơ sở dữ liệu, mô hình quan hệ, và ràng buộc dữ liệu.
* Shalloway, A., & Trott, J. R. (2004). Design Patterns Explained: A New Perspective on Object-Oriented Design (2nd Edition). Addison-Wesley : Tài liệu hỗ trợ thiết kế kiến trúc hệ thống và sử dụng mẫu thiết kế (Design Patterns) trong phát triển phần mềm.
* Figma Inc. (2024). Figma: The Collaborative Interface Design Tool [Online]. Available at: <https://www.figma.com/> : Tài liệu tham khảo cho phần thiết kế giao diện người dùng (UI/UX) và prototyping.
* Object Management Group (OMG). (2017). Unified Modeling Language (UML) Version 2.5.1 Specification : Tài liệu chính thức mô tả chuẩn UML cho các loại biểu đồ trong mô hình hóa yêu cầu và thiết kế hướng đối tượng.
* IEEE Std 1471-2000. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems : Tham khảo cho phần thiết kế kiến trúc phần mềm (Software Architecture Design).
* Chatgpt