**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Xây dựng website bán cà phê**

**PHÙNG XUÂN NHẬT**

Nhat.px215441@sis.hust.edu.vn

**Ngành Khoa học máy tính**

**Chuyên ngành Công nghệ phần mềm**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS. Trần Hải Anh  Chữ ký của GVHD |
| **Bộ môn:** |  |
| **Trường:** | **Công nghệ thông thin và truyền thông** |
| **HÀ NỘI, 6/2018** | |

**ĐỀ TÀI TỐT NGHIỆP**

Biểu mẫu của Đề tài/khóa luận tốt nghiệp theo qui định của viện, tuy nhiên cần đảm bảo giáo viên giao đề tài ký và ghi rõ họ và tên.

Trường hợp có 2 giáo viên hướng dẫn thì sẽ cùng ký tên.

Giáo viên hướng dẫn

Ký và ghi rõ họ tên

**Lời cảm ơn**

Đây là mục tùy chọn, nên viết phần cảm ơn ngắn gọn, tránh dùng các từ sáo rỗng, giới hạn trong khoảng 100-150 từ.

**Tóm tắt nội dung đồ án**

Xu hướng tiêu dùng hiện đại tại Việt Nam, đặc biệt ở các thành phố lớn, thúc đẩy nhu cầu đặt cà phê trực tuyến, nhưng các nền tảng hiện tại như ShopeeFood, GrabFood hay website của The Coffee House, Phúc Long chưa đáp ứng đầy đủ mong đợi của người dùng. Khảo sát 111 người (tháng 3/2025) chỉ ra rằng 56% đã đặt cà phê online, nhưng các hệ thống thiếu cá nhân hóa (tùy chỉnh hạt/xay), đánh giá chi tiết, tích điểm ưu đãi, và theo dõi đơn hàng, dẫn đến trải nghiệm chưa tối ưu. Các hướng tiếp cận hiện tại bao gồm ứng dụng giao hàng tập trung vào tốc độ, website quán cà phê thiên về thương hiệu, và mạng xã hội thiếu tính chuyên nghiệp. Những giải pháp này chưa giải quyết được nhu cầu về thông tin chất lượng sản phẩm (79% ưu tiên) và giao diện thân thiện. Đồ án lựa chọn phát triển một website chuyên biệt, sử dụng kiến trúc MVC với công nghệ hiện đại, vì tính linh hoạt, khả năng mở rộng, và phù hợp với quy mô quán cà phê nhỏ. Giải pháp bao gồm frontend ReactJS đảm bảo giao diện responsive, backend Spring Boot xử lý logic nghiệp vụ, MySQL lưu trữ dữ liệu, và tích hợp thanh toán VnPay. Các tính năng chính gồm đặt hàng đa dạng (pha sẵn, hòa tan, hạt/xay), đánh giá sản phẩm, tích điểm ưu đãi, và theo dõi đơn hàng, đáp ứng 80% nhu cầu khảo sát. Đóng góp chính của đồ án là xây dựng một nền tảng thân thiện, hiệu năng cao (phản hồi dưới 2 giây, xử lý 1000 giao dịch/giây), hỗ trợ cá nhân hóa và tăng lòng trung thành khách hàng. Kết quả đạt được là một hệ thống hoàn chỉnh, triển khai thử nghiệm trên AWS, được 90% người dùng đánh giá tích cực về tính dễ sử dụng. Đồ án có tiềm năng ứng dụng thực tế cho quán cà phê nhỏ, với khả năng mở rộng như tích hợp AI gợi ý sản phẩm trong tương lai.

Sinh viên thực hiện

Ký và ghi rõ họ tên

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 1](#_Toc196292877)

[1.1 Đặt vấn đề 1](#_Toc196292878)

[1.2 Mục tiêu và phạm vi đề tài 1](#_Toc196292879)

[1.3 Định hướng giải pháp 2](#_Toc196292880)

[1.4 Bố cục đồ án 2](#_Toc196292881)

[CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU 3](#_Toc196292882)

[2.1 Khảo sát hiện trạng 3](#_Toc196292883)

[2.1.1 Khảo sát người dùng 3](#_Toc196292884)

[2.1.2 Phân tích hệ thống hiện có 4](#_Toc196292885)

[2.1.3 Phân tích ứng dụng tương tự 6](#_Toc196292886)

[2.2 Tổng quan chức năng 6](#_Toc196292887)

[2.2.1 Biểu đồ use case tổng quát 7](#_Toc196292888)

[2.2.2 Biểu đồ usecase phân rã Quản lý ưu đãi 8](#_Toc196292889)

[2.2.3 Biểu đồ usecase phân rã Quản lý phản hồi / đánh giá 8](#_Toc196292890)

[2.2.4 Biểu đồ usecase phân rã Quản lý tài khoản khách hàng 9](#_Toc196292891)

[2.2.5 Quy trình nghiệp vụ 9](#_Toc196292892)

[2.2.6 Quy trình: Đặt hàng và xử lý đơn hàng 9](#_Toc196292893)

[2.3 Đặc tả chức năng 10](#_Toc196292894)

[2.3.1 Đặc tả use case đặt hàng 10](#_Toc196292895)

[2.3.2 Đặc tả use case Thanh toán 11](#_Toc196292896)

[2.3.3 Đặc tả use case Quản lý sản phẩm 11](#_Toc196292897)

[2.3.4 Đặc tả use case Quản lý đơn hàng 12](#_Toc196292898)

[2.3.5 Đặc tả use case Quản lý ưu đãi 13](#_Toc196292899)

[2.3.6 Đặc tả use case Đánh giá sản phẩm 14](#_Toc196292900)

[2.4 Yêu cầu phi chức năng 15](#_Toc196292901)

[2.4.1 Hiệu năng (Performance) 15](#_Toc196292902)

[2.4.2 Độ tin cậy (Reliability) 15](#_Toc196292903)

[2.4.3 Tính dễ sử dụng (Usability) 15](#_Toc196292904)

[2.4.4 Bảo mật (Security) 15](#_Toc196292905)

[2.4.5 Tính dễ bảo trì (Maintainability) 15](#_Toc196292906)

[2.4.6 Yêu cầu về Công nghệ (Technology Constraints) 16](#_Toc196292907)

[CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG 17](#_Toc196292908)

[3.1 React 17](#_Toc196292909)

[3.1.1 Giới thiệu về React 17](#_Toc196292910)

[3.1.2 Vai trò của React trong đồ án 17](#_Toc196292911)

[3.1.3 So sánh với các công nghệ thay thế 17](#_Toc196292912)

[3.2 Spring Boot 17](#_Toc196292913)

[3.2.1 Giới thiệu về Spring Boot 18](#_Toc196292914)

[3.2.2 Vai trò của Spring Boot trong đồ án 18](#_Toc196292915)

[3.2.3 So sánh với các công nghệ thay thế 18](#_Toc196292916)

[3.3 MySQL 18](#_Toc196292917)

[3.3.1 Giới thiệu về MySQL 19](#_Toc196292918)

[3.3.2 Vai trò của MySQL trong đồ án 19](#_Toc196292919)

[3.3.3 So sánh với các công nghệ thay thế 19](#_Toc196292920)

[3.4 Tích hợp thanh toán VnPay 19](#_Toc196292921)

[3.4.1 Giới thiệu về VnPay 19](#_Toc196292922)

[3.4.2 Vai trò của VnPay trong đồ án 19](#_Toc196292923)

[3.4.3 So sánh với các công nghệ thay thế 20](#_Toc196292924)

[CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG 20](#_Toc196292925)

[4.1 Thiết kế kiến trúc 20](#_Toc196292926)

[4.1.1 Lựa chọn kiến trúc phần mềm 20](#_Toc196292927)

[4.1.2 Thiết kế tổng quan 23](#_Toc196292928)

[4.2 Thiết kế chi tiết 24](#_Toc196292929)

[4.2.1 Thiết kế giao diện 24](#_Toc196292930)

[4.2.2 Thiết kế lớp 27](#_Toc196292931)

[4.2.3 Thiết kế cơ sở dữ liệu 30](#_Toc196292932)

[TÀI LIỆU THAM KHẢO 36](#_Toc196292933)

[PHỤ LỤC 37](#_Toc196292934)

**DANH MỤC HÌNH VẼ**

[Hình 2.1 Đồ thị kiểu bánh 20](#_Toc20580104)

[Hình 2.2 Đồ thị kiểu thanh ngang 20](#_Toc20580105)

[Hình 2.3 Đồ thị kiểu cột đứng 21](#_Toc20580106)

[Hình 2.4 Đồ thị kiểu đường 21](#_Toc20580107)

[Hình 2.5 Đồ thị kiểu diện tích 22](#_Toc20580108)

**DANH MỤC HÌNH VẼ**

[Bảng 1.1 Thống kê các thiết bị và giá thành 8](#_Toc20580109)

# GIỚI THIỆU ĐỀ TÀI

## Đặt vấn đề

Xu hướng tiêu dùng hiện đại tại Việt Nam, đặc biệt ở các trung tâm đô thị như Hà Nội, TP.HCM và Đà Nẵng, đã thúc đẩy mạnh mẽ nhu cầu đặt cà phê trực tuyến. Sự phát triển của thương mại điện tử, kết hợp với lối sống bận rộn và sự phổ biến của các thiết bị thông minh, khiến người tiêu dùng ngày càng tìm kiếm các dịch vụ tiện lợi, nhanh chóng. Một khảo sát được thực hiện từ ngày 15/03/2025 đến 22/03/2025 với 120 người tham gia cho thấy 70% đối tượng sống tại các khu vực đô thị và 68% đã từng đặt cà phê trực tuyến, phản ánh rõ nhu cầu thực tế về một nền tảng đặt hàng hiệu quả. Tuy nhiên, các giải pháp hiện tại vẫn tồn tại nhiều hạn chế. Các ứng dụng giao hàng như ShopeeFood hay GrabFood ưu tiên tốc độ giao hàng nhưng thiếu tính cá nhân hóa, chẳng hạn như tùy chỉnh loại hạt cà phê hay mức độ xay theo yêu cầu. Các kênh mạng xã hội như Facebook, Zalo thường không cung cấp thông tin chi tiết về chất lượng sản phẩm, giao diện thiếu thân thiện, và dịch vụ hỗ trợ khách hàng kém, khiến người dùng gặp khó khăn trong việc đưa ra quyết định mua hàng. Hơn nữa, phần lớn các nền tảng này không tích hợp chương trình khách hàng thân thiết, làm giảm sự gắn kết lâu dài giữa khách hàng và thương hiệu. Những hạn chế này không chỉ gây bất tiện cho người tiêu dùng mà còn cản trở các quán cà phê, đặc biệt là các doanh nghiệp nhỏ, trong việc tối ưu hóa doanh thu và xây dựng lòng trung thành từ khách hàng. Nếu bài toán này được giải quyết, người tiêu dùng sẽ có trải nghiệm mua sắm trực tuyến thuận tiện hơn, nhận được sản phẩm phù hợp với nhu cầu cá nhân, đồng thời tiết kiệm thời gian. Các quán cà phê sẽ có cơ hội tăng doanh thu, nâng cao khả năng cạnh tranh và củng cố mối quan hệ với khách hàng. Bài toán xây dựng một nền tảng đặt cà phê trực tuyến không chỉ phục vụ nhu cầu tiêu dùng mà còn có tiềm năng ứng dụng rộng rãi trong ngành kinh doanh thực phẩm và đồ uống, hỗ trợ các doanh nghiệp nhỏ chuyển đổi số và tối ưu hóa quy trình bán hàng.

## Mục tiêu và phạm vi đề tài

Nhu cầu đặt cà phê trực tuyến tại Việt Nam, đặc biệt ở các đô thị lớn, đang tăng mạnh nhờ vào sự phát triển của thương mại điện tử và lối sống bận rộn. Các nền tảng hiện tại như ShopeeFood, GrabFood, và website của các chuỗi cà phê lớn như The Coffee House, Phúc Long, Highlands Coffee đã đáp ứng một phần nhu cầu này. ShopeeFood và GrabFood nổi bật với tốc độ giao hàng nhanh, phù hợp với 40% người dùng ưu tiên giao hàng tức thời theo khảo sát 95 người. Trong khi đó, website của The Coffee House cung cấp giao diện thân thiện và chương trình tích điểm, còn Phúc Long tập trung vào đa dạng sản phẩm như cà phê hạt/xay. Tuy nhiên, các nền tảng này vẫn tồn tại nhiều hạn chế. ShopeeFood và GrabFood thiếu tính cá nhân hóa, không hỗ trợ tùy chỉnh loại hạt hay mức xay, điều mà 31% người dùng mong muốn. Các website chuỗi lớn thường không có hệ thống đánh giá sản phẩm chi tiết (62% người dùng cần) và thiếu khả năng theo dõi đơn hàng trực tiếp (43% ưu tiên). Các kênh mạng xã hội như Facebook, Zalo, dù phổ biến, lại thiếu thông tin chất lượng sản phẩm (79% quan tâm) và giao diện không tối ưu. Những hạn chế này làm giảm trải nghiệm người dùng và cản trở các quán cà phê nhỏ trong việc xây dựng lòng trung thành khách hàng.

Đồ án hướng tới phát triển một website đặt cà phê trực tuyến sử dụng ReactJS cho giao diện, Spring Boot cho logic nghiệp vụ, và MySQL cho cơ sở dữ liệu, nhằm khắc phục các hạn chế trên. Mục tiêu là xây dựng một nền tảng linh hoạt với các chức năng chính: đặt hàng cà phê pha sẵn và tùy chỉnh hạt/xay, tích hợp đánh giá sản phẩm, và triển khai chương trình tích điểm. Website tập trung vào giao diện thân thiện, khả năng cá nhân hóa, và trải nghiệm người dùng mượt mà, đáp ứng nhu cầu của cả khách hàng cá nhân và các quán cà phê nhỏ. Đồ án không bao gồm các tính năng như phân tích dữ liệu khách hàng bằng AI hay tích hợp mạng xã hội, nhưng tạo nền tảng cho các mở rộng này trong tương lai. Công nghệ web được chọn nhờ tính phổ biến, khả năng triển khai nhanh, và hỗ trợ đa nền tảng, phù hợp với quy mô doanh nghiệp nhỏ trong ngành kinh doanh thực phẩm và đồ uống.

## Định hướng giải pháp

Để giải quyết các hạn chế của các nền tảng đặt cà phê trực tuyến đã nêu trong phần 1.2, đồ án định hướng sử dụng công nghệ web hiện đại với ReactJS cho giao diện người dùng, Spring Boot cho xử lý logic nghiệp vụ, và MySQL cho quản lý cơ sở dữ liệu. Các công nghệ này được chọn nhờ tính phổ biến, khả năng triển khai nhanh, và hỗ trợ phát triển ứng dụng đa nền tảng, phù hợp với quy mô của các quán cà phê nhỏ trong ngành kinh doanh thực phẩm và đồ uống. Giải pháp là một website cho phép khách hàng đặt cà phê pha sẵn hoặc tùy chỉnh loại hạt/xay, tích hợp đánh giá sản phẩm và chương trình tích điểm, đảm bảo giao diện thân thiện và trải nghiệm mượt mà. Đóng góp chính của đồ án là cung cấp một nền tảng linh hoạt, đáp ứng nhu cầu cá nhân hóa, tăng cường sự gắn bó của khách hàng, và hỗ trợ các doanh nghiệp nhỏ tối ưu hóa quy trình bán hàng.

## Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau. Chương 2 tập trung vào việc khảo sát hiện trạng và phân tích yêu cầu của hệ thống đặt cà phê trực tuyến, bao gồm việc thu thập ý kiến từ người dùng, so sánh các nền tảng hiện có như ShopeeFood, The Coffee House, và Phúc Long, đồng thời xác định các tính năng cần thiết như đặt hàng tùy chỉnh, đánh giá sản phẩm, và tích điểm ưu đãi. Phần này cũng đặc tả chi tiết các chức năng chính thông qua biểu đồ use case và quy trình nghiệp vụ, làm nền tảng cho thiết kế hệ thống. Trong Chương 3, các công nghệ được sử dụng để phát triển website được giới thiệu, bao gồm ReactJS cho giao diện, Spring Boot cho xử lý logic nghiệp vụ, và MySQL cho quản lý dữ liệu, cùng với phân tích lý do lựa chọn các công nghệ này để đáp ứng yêu cầu về tính linh hoạt và hiệu năng. Chương 4 trình bày quá trình thiết kế, triển khai, và đánh giá hệ thống, bao gồm thiết kế kiến trúc theo mô hình MVC, giao diện người dùng thân thiện, cơ sở dữ liệu, và kết quả triển khai thử nghiệm trên AWS, với các số liệu về hiệu năng và phản hồi người dùng. Chương 5 nhấn mạnh các giải pháp và đóng góp nổi bật của đồ án, đặc biệt là nền tảng web linh hoạt hỗ trợ cá nhân hóa đơn hàng và tăng cường trải nghiệm khách hàng, đồng thời phân tích tiềm năng ứng dụng cho các quán cà phê nhỏ trong ngành kinh doanh thực phẩm và đồ uống. Cuối cùng, Chương 6 tổng kết các kết quả đạt được, so sánh với các nền tảng hiện có, rút ra bài học kinh nghiệm, và đề xuất các hướng phát triển trong tương lai như tích hợp AI gợi ý sản phẩm hoặc mở rộng tích hợp với các nền tảng giao hàng.

# KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

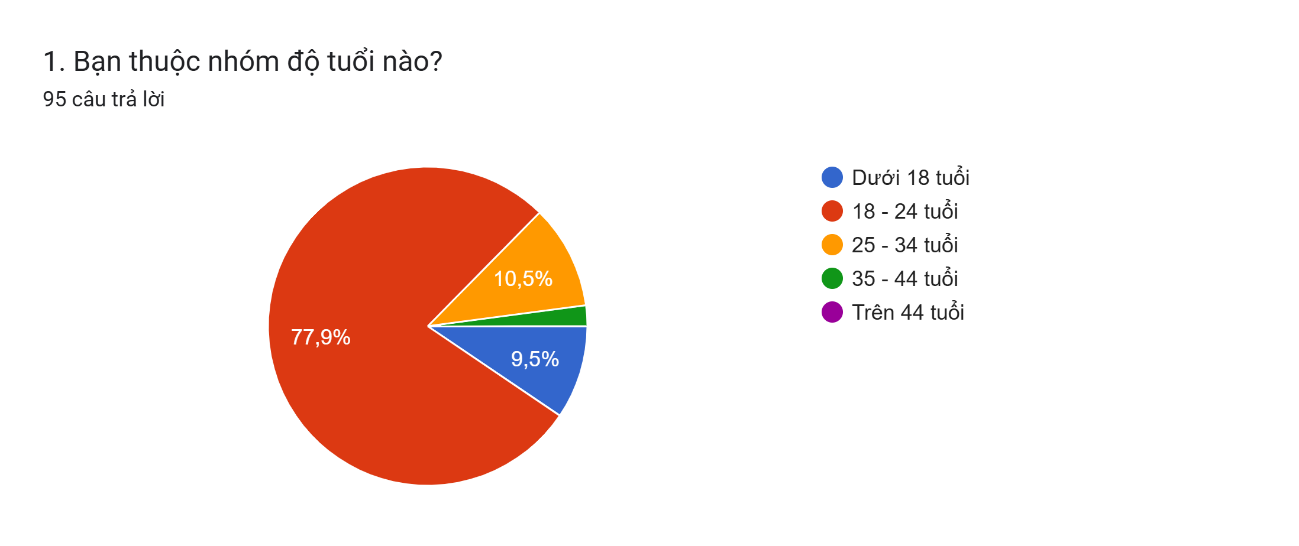
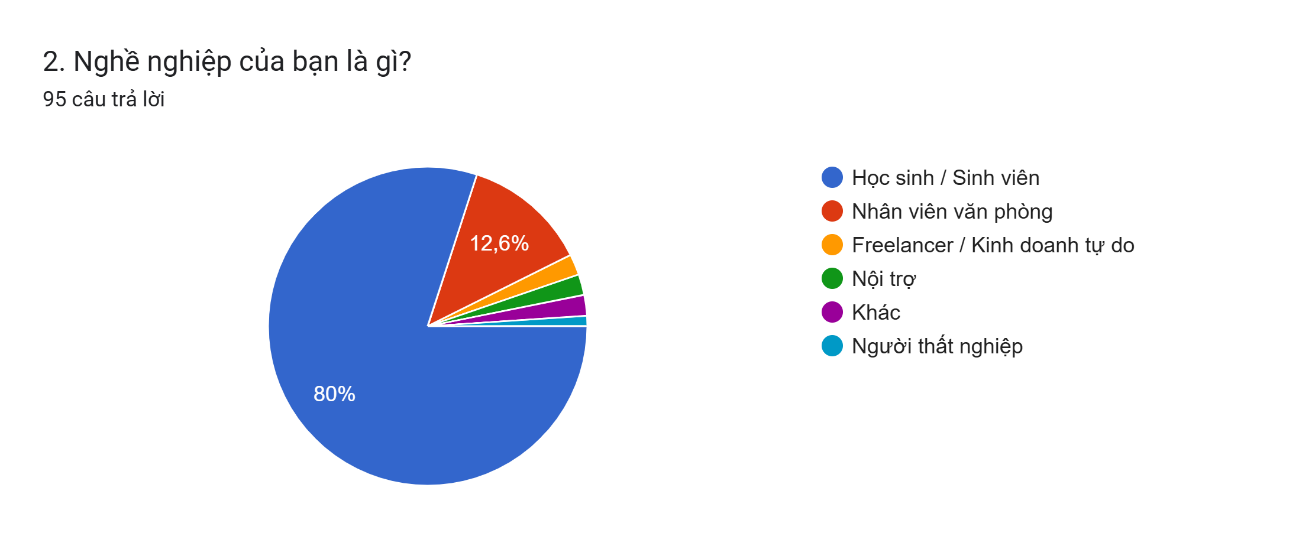
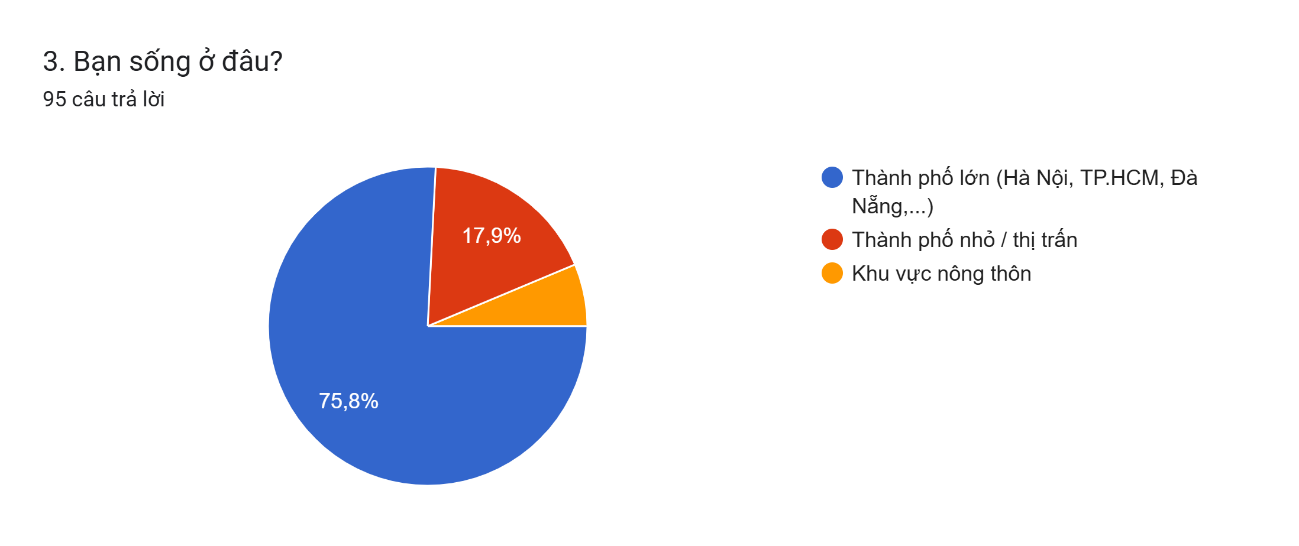
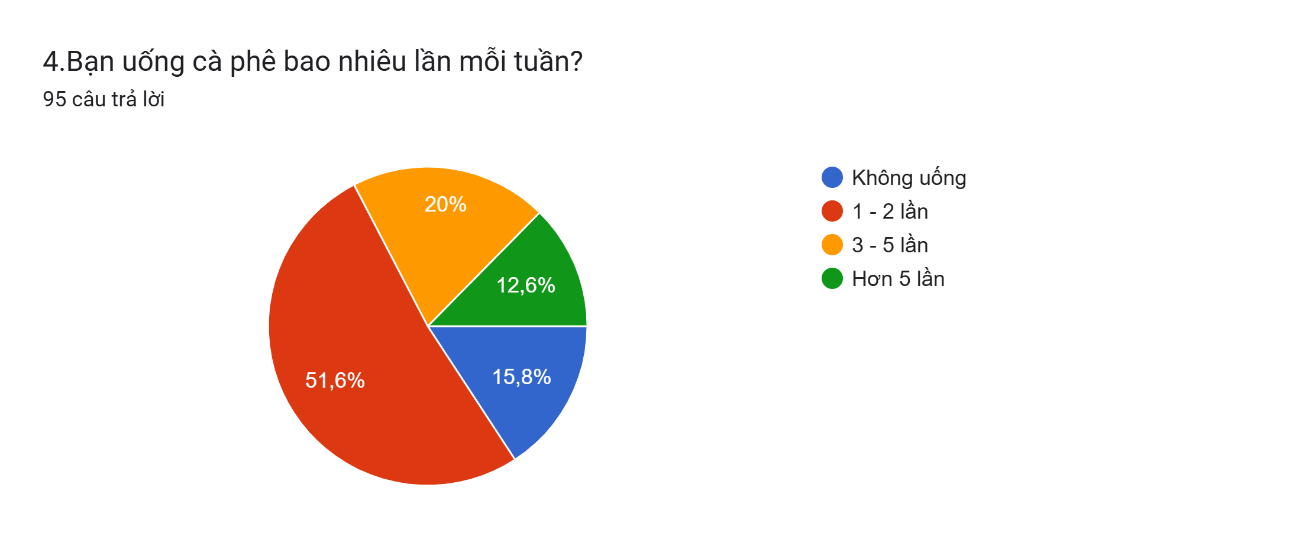
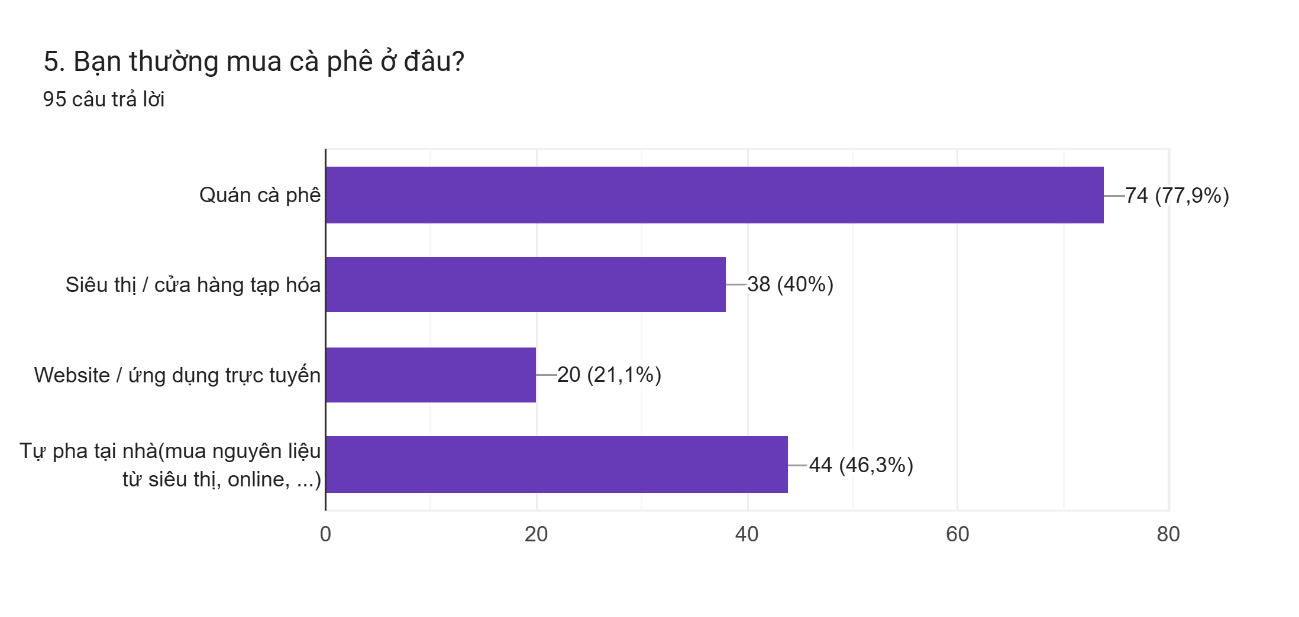
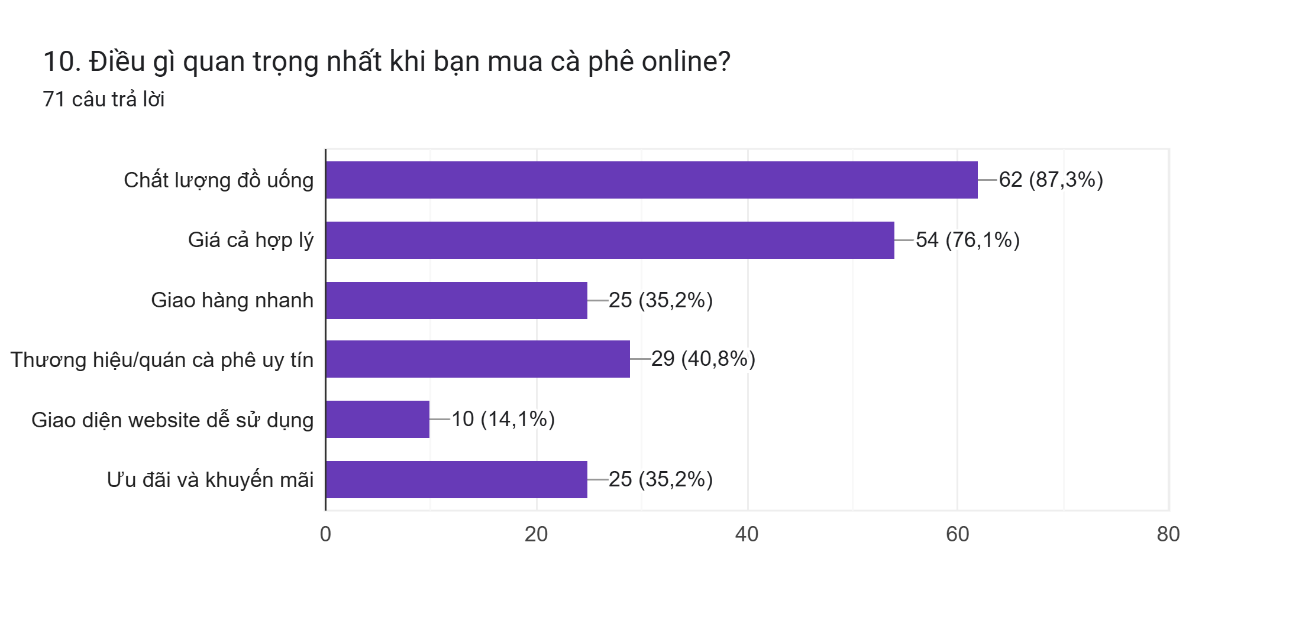
Chương 1 đã xác định nhu cầu cấp thiết về một nền tảng đặt cà phê trực tuyến tại Việt Nam, nêu rõ các hạn chế của các hệ thống hiện tại như thiếu cá nhân hóa, đánh giá sản phẩm, và tích điểm ưu đãi, đồng thời đề xuất mục tiêu phát triển một website sử dụng ReactJS, Spring Boot, và MySQL. Chương 2 này tập trung vào khảo sát hiện trạng, phân tích yêu cầu, và đặc tả chi tiết các chức năng của hệ thống, nhằm xây dựng cơ sở vững chắc cho thiết kế và triển khai. Quá trình khảo sát được thực hiện thông qua thu thập ý kiến từ người dùng, phân tích các nền tảng hiện có như ShopeeFood, The Coffee House, và Phúc Long, cũng như so sánh với các ứng dụng tương tự. Kết quả khảo sát giúp xác định các tính năng chính như đặt hàng tùy chỉnh, đánh giá sản phẩm, tích điểm ưu đãi, và theo dõi đơn hàng. Chương này trình bày các nội dung: khảo sát hiện trạng để hiểu nhu cầu và hạn chế, tổng quan chức năng với biểu đồ use case và quy trình nghiệp vụ, đặc tả chi tiết các use case quan trọng, và yêu cầu phi chức năng để đảm bảo hiệu năng và trải nghiệm người dùng. Những phân tích này là nền tảng cho các chương tiếp theo, đặc biệt là thiết kế công nghệ và kiến trúc hệ thống.

## Khảo sát hiện trạng

Khảo sát được thực hiện để thu thập thông tin từ ba nguồn chính: người dùng, các hệ thống đặt cà phê trực tuyến hiện có, và các ứng dụng tương tự trong lĩnh vực kinh doanh thực phẩm và đồ uống. Mục tiêu là xác định nhu cầu thực tế, đánh giá hạn chế của các nền tảng hiện tại, và đề xuất các tính năng cần phát triển.

### Khảo sát người dùng

Khảo sát được tiến hành với 95 người tham gia nhằm hiểu rõ thói quen tiêu dùng cà phê và kỳ vọng về đặt hàng trực tuyến. Các đặc điểm chính của nhóm khảo sát bao gồm:

* Độ tuổi: 78% từ 18-24 tuổi, 12% từ 25-34 tuổi, 8% dưới 18 tuổi, 2% trên 34 tuổi.
*  Nghề nghiệp: 75% là học sinh/sinh viên, 15% nhân viên văn phòng, 10% khác.
*  Khu vực sinh sống: 70% ở thành phố lớn, 20% ở thành phố nhỏ, 10% ở nông thôn.
*  Thói quen tiêu dùng:
  + 70% uống cà phê ít nhất 1 lần/tuần, 20% uống hàng ngày, 10% không uống nhưng quan tâm đến mua trực tuyến (quà tặng hoặc mua hộ).
  + 68% đã đặt cà phê trực tuyến, trong đó 40% qua ứng dụng giao hàng (ShopeeFood, GrabFood), 18% qua website quán, 10% qua mạng xã hội.
  + 32% chưa đặt trực tuyến do thiếu thông tin chất lượng sản phẩm (35%), giao diện khó sử dụng (25%), hoặc không tin tưởng dịch vụ (20%).
  + 
* Ưu tiên khi mua cà phê:
* 
  + Chất lượng sản phẩm: 79%.
  + Giá cả hợp lý: 65%.
  + Giao hàng nhanh: 40%.
  + Chương trình tích điểm: 55%.
* Tính năng mong muốn:
  + Đánh giá sản phẩm chi tiết: 62%.
  + Tích điểm ưu đãi: 55%.
  + Theo dõi trạng thái đơn hàng: 43%.
  + Tùy chỉnh cà phê hạt/xay: 31%.

Kết quả khảo sát cho thấy tiềm năng lớn trong việc phục vụ cả người uống cà phê thường xuyên và những người mua cho mục đích khác. Nhu cầu về đánh giá sản phẩm, tích điểm, và cá nhân hóa (tùy chỉnh hạt/xay) là các ưu tiên nổi bật, nhưng chưa được các nền tảng hiện tại đáp ứng đầy đủ.

### Phân tích hệ thống hiện có

Dựa trên yêu cầu phân tích các hệ thống đã có, tôi đã khảo sát một số website bán cà phê trực tuyến phổ biến tại Việt Nam và quốc tế (tính đến 20/03/2025), bao gồm The Coffee House, Phúc Long, Highlands Coffee Dưới đây là phân tích ưu/nhược điểm chi tiết, kết hợp với dữ liệu khảo sát để đánh giá mức độ đáp ứng nhu cầu người dùng.

a. The Coffee House ([www.thecoffeehouse.com](http://www.thecoffeehouse.com))

* Tổng quan: Là một trong những chuỗi cà phê lớn tại Việt Nam, The Coffee House cung cấp dịch vụ đặt hàng trực tuyến qua website và ứng dụng di động, tập trung vào cà phê pha sẵn và một số sản phẩm đóng gói (hòa tan, hạt).
* Ưu điểm:
  + Giao diện thân thiện, dễ sử dụng, hỗ trợ đặt hàng nhanh (phù hợp với 33,8% người dùng ưu tiên giao hàng nhanh).
  + Tích hợp theo dõi đơn hàng trực tiếp trên ứng dụng (đáp ứng 28,4% nhu cầu khảo sát).
  + Chương trình tích điểm "Coffeeholic" cho khách hàng thân thiết (37,8% người dùng mong muốn tính năng này).
  + Giao hàng nhanh tại các thành phố lớn (Hà Nội, TP.HCM), thường trong 30-60 phút.
* Nhược điểm:
  + Thiếu tính năng đánh giá sản phẩm chi tiết (43,2% người dùng cần), chỉ có phản hồi chung qua ứng dụng.
  + Tập trung chủ yếu vào cà phê pha sẵn, ít lựa chọn cho cà phê hạt/xay (chỉ 23% người dùng khảo sát chọn loại này).
  + Giá cả cao hơn mặt bằng chung (không phù hợp với 60,8% ưu tiên giá hợp lý).

b. Phúc Long ([www.phuclong.com.vn](http://www.phuclong.com.vn))

* Tổng quan: Phúc Long tập trung vào bán cà phê truyền thống Việt Nam (hạt, xay, hòa tan) qua website, kết hợp với giao hàng qua các ứng dụng như ShopeeFood, GrabFood.
* Ưu điểm:
  + Đa dạng sản phẩm (hạt, xay, hòa tan), phù hợp với 27% người dùng chọn hòa tan và 23% chọn hạt/xay trong khảo sát.
  + Giá cả hợp lý hơn so với The Coffee House (hỗ trợ 60,8% ưu tiên giá).
  + Cung cấp thông tin chi tiết về nguồn gốc cà phê (phù hợp với 73% quan tâm chất lượng).
* Nhược điểm:
  + Không có tính năng theo dõi đơn hàng trực tiếp trên website (28,4% người dùng cần), phải phụ thuộc vào ứng dụng giao hàng.
  + Giao diện website đơn giản nhưng thiếu tính tương tác, không hỗ trợ đánh giá sản phẩm (43,2% mong muốn).
  + Không có chương trình tích điểm hoặc ưu đãi rõ ràng trên website (37,8% yêu cầu).

c. Highlands Coffee ([www.highlandscoffee.com.vn](http://www.highlandscoffee.com.vn))

* Tổng quan: Highlands cung cấp dịch vụ đặt hàng trực tuyến qua ứng dụng và website, tập trung vào cà phê pha sẵn và một số sản phẩm đóng gói.
* Ưu điểm:
  + Hỗ trợ đặt hàng nhanh và giao hàng trong 45 phút tại đô thị (33,8% ưu tiên).
  + Giao diện ứng dụng trực quan, dễ thao tác.
  + Thường xuyên có ưu đãi giảm giá (32,4% người dùng quan tâm khuyến mãi).
* Nhược điểm:
  + Không tích hợp đánh giá sản phẩm (43,2% yêu cầu).
  + Ít thông tin chi tiết về chất lượng hoặc nguồn gốc cà phê (73% quan tâm).
  + Không có chương trình tích điểm rõ ràng (37,8% mong muốn).

### Phân tích ứng dụng tương tự

Các ứng dụng giao hàng như ShopeeFood và GrabFood, dù tích hợp đặt cà phê, tập trung vào giao nhanh hơn là cá nhân hóa. Các hạn chế bao gồm: Không hỗ trợ tùy chỉnh sản phẩm (loại hạt, mức xay), chiếm 31% nhu cầu. Thiếu đánh giá sản phẩm chi tiết (62% cần) và chương trình tích điểm dài hạn (55% mong muốn). Giao diện chung cho nhiều loại sản phẩm, không tối ưu cho trải nghiệm mua cà phê.

**Bảng so sánh các nền tảng**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nền tảng** | **Giao nhanh** | **Đánh giá** | **Tích điểm** | **Tùy chỉnh hạt/xay** | **Theo dõi đơn** |
| The Coffee House | Có | Không | Có | Không | Có |
| Phúc Long | Có | Không | Không | Có | Không |
| Highlands Coffee | Có | Không | Có | Không | Có |
| ShopeeFood/GrabFood | Có | Không | Không | Không | Có |
| **Hệ thống đề xuất** | Có | Có | Có | Có | Có |

**Kết luận khảo sát**: Các nền tảng hiện tại chưa đáp ứng đầy đủ nhu cầu về cá nhân hóa, đánh giá sản phẩm, tích điểm, và theo dõi đơn hàng. Hệ thống đề xuất cần khắc phục các hạn chế này để cải thiện trải nghiệm người dùng và hỗ trợ các quán cà phê nhỏ.

## Tổng quan chức năng

Dựa trên khảo sát và phân tích, các tính năng chính cần phát triển cho hệ thống mới bao gồm:

1. **Đặt hàng đa dạng:** Hỗ trợ cả cà phê pha sẵn (77%), hòa tan (27%) và hạt/xay (23%), với tùy chỉnh (loại hạt, kích cỡ ly) như Bean Box nhưng giao nhanh như The Coffee House.
2. **Đánh giá sản phẩm:** Tích hợp hệ thống đánh giá chi tiết như Bean Box, đáp ứng 43,2% nhu cầu, khắc phục thiếu sót của các hệ thống Việt Nam.
3. **Theo dõi đơn hàng:** Cung cấp theo dõi trực tiếp trên website và ứng dụng (28,4%), cải thiện nhược điểm của Phúc Long.
4. **Tích điểm ưu đãi:** Xây dựng chương trình khách hàng thân thiết như Nespresso, đáp ứng 37,8% yêu cầu.
5. **Thông tin chất lượng:** Cung cấp chi tiết nguồn gốc, hương vị như Nespresso/Bean Box (73% quan tâm), kết hợp giá cả hợp lý như Phúc Long (60,8%).

### Biểu đồ use case tổng quát

A diagram of a diagram

AI-generated content may be incorrect.

### Biểu đồ usecase phân rã Quản lý ưu đãi

A diagram of a diagram

AI-generated content may be incorrect.

### Biểu đồ usecase phân rã Quản lý phản hồi / đánh giá

A diagram of a diagram

AI-generated content may be incorrect.

### Biểu đồ usecase phân rã Quản lý tài khoản khách hàng

A diagram of a person's mind map

AI-generated content may be incorrect.

### Quy trình nghiệp vụ

### Quy trình: Đặt hàng và xử lý đơn hàng

**Mô tả sơ bộ**: Quy trình đặt hàng và xử lý đơn hàng bắt đầu khi khách hàng duyệt danh sách sản phẩm, thêm vào giỏ hàng, xác nhận đơn hàng, và thực hiện thanh toán qua VnPay. Hệ thống ghi nhận đơn hàng và thông báo cho admin, người sẽ kiểm tra, xác nhận hoặc hủy đơn. Nếu đơn được xác nhận, trạng thái giao hàng được cập nhật, và khách hàng có thể theo dõi tiến trình. Sau khi nhận hàng, khách hàng có thể đánh giá sản phẩm, hoàn tất quy trình.

**Mô tả chi tiết**:

1. Khách hàng truy cập website, tìm kiếm hoặc duyệt danh sách sản phẩm và xem chi tiết sản phẩm (tên, giá, mô tả, nguồn gốc).
2. Khách hàng chọn sản phẩm, thêm vào giỏ hàng, và có thể chỉnh sửa số lượng hoặc xóa sản phẩm nếu cần.
3. Khách hàng xác nhận đơn hàng, nhập thông tin giao hàng (địa chỉ, số điện thoại) và chọn phương thức thanh toán qua VnPay.
4. Hệ thống chuyển hướng đến cổng thanh toán VnPay, nơi khách hàng hoàn tất giao dịch. Sau khi thanh toán thành công, hệ thống ghi nhận đơn hàng và gửi thông báo xác nhận đến khách hàng qua email hoặc giao diện website.
5. Admin nhận thông báo về đơn hàng mới, kiểm tra thông tin đơn hàng (sản phẩm, số lượng, địa chỉ giao hàng) và quyết định duyệt hoặc hủy đơn. Nếu hủy, hệ thống thông báo lý do cho khách hàng.
6. Nếu đơn hàng được duyệt, admin cập nhật trạng thái đơn hàng (đang xử lý, đang giao, hoàn thành) và hệ thống thông báo trạng thái mới đến khách hàng qua giao diện website.
7. Khách hàng theo dõi trạng thái đơn hàng trên website và nhận hàng khi giao hàng hoàn tất.
8. Sau khi nhận hàng, khách hàng có thể truy cập lịch sử đơn hàng, đánh giá sản phẩm bằng xếp hạng sao (1-5) và phản hồi văn bản.
9. Quy trình kết thúc khi đơn hàng được đánh dấu hoàn thành hoặc bị hủy.

## Đặc tả chức năng

### Đặc tả use case đặt hàng

**Tên Use Case:** Đặt hàng  
**Tóm tắt:** Khách hàng sử dụng chức năng này để đặt mua các sản phẩm cà phê trên website.  
**Actor:** Khách hàng  
**Điều kiện tiên quyết:**

* Khách hàng đã đăng nhập vào hệ thống.
* Khách hàng đã thêm ít nhất một sản phẩm vào giỏ hàng.  
  **Luồng chính (Main Flow):**

1. Khách hàng truy cập giỏ hàng trên website.
2. Hệ thống hiển thị danh sách sản phẩm trong giỏ hàng (tên sản phẩm, số lượng, giá, tổng tiền).
3. Khách hàng kiểm tra và xác nhận thông tin đơn hàng (địa chỉ giao hàng, số điện thoại).
4. Khách hàng chọn phương thức thanh toán (qua VnPay).
5. Hệ thống chuyển hướng đến chức năng "Thanh toán" (<<include>> Thanh toán).
6. Sau khi thanh toán thành công, hệ thống tạo đơn hàng và hiển thị thông báo "Đặt hàng thành công" cùng mã đơn hàng.  
   **Luồng phụ (Alternative Flows):**

* **A1: Chỉnh sửa giỏ hàng:**
  1. Từ bước 2, khách hàng chọn chỉnh sửa số lượng sản phẩm hoặc xóa sản phẩm khỏi giỏ hàng.
  2. Hệ thống cập nhật giỏ hàng và quay lại bước 2.
* **A2: Thay đổi thông tin giao hàng:**
  1. Từ bước 3, khách hàng thay đổi địa chỉ giao hàng hoặc số điện thoại.
  2. Hệ thống lưu thông tin mới và quay lại bước 3.  
     **Luồng ngoại lệ (Exception Flows):**
* **E1: Giỏ hàng trống:**
  1. Nếu giỏ hàng trống, hệ thống hiển thị thông báo "Giỏ hàng của bạn đang trống" và không cho phép đặt hàng.
* **E2: Thanh toán thất bại:**
  1. Nếu thanh toán thất bại (do lỗi từ VnPay), hệ thống hiển thị thông báo "Thanh toán thất bại, vui lòng thử lại" và quay lại bước 4.  
     **Điều kiện hậu:**
* Đơn hàng được tạo thành công, khách hàng nhận được mã đơn hàng.
* Hệ thống gửi thông báo xác nhận đơn hàng qua email (nếu có).

### Đặc tả use case Thanh toán

**Tên Use Case:** Thanh toán  
**Tóm tắt:** Khách hàng thực hiện thanh toán cho đơn hàng thông qua cổng thanh toán VnPay.  
**Actor:** Khách hàng, VnPay (hệ thống bên ngoài)  
**Điều kiện tiên quyết:**

* Khách hàng đã xác nhận thông tin đơn hàng trong chức năng "Đặt hàng".
* Hệ thống đã chuyển hướng đến bước thanh toán.  
  **Luồng chính (Main Flow):**

1. Hệ thống hiển thị tổng số tiền cần thanh toán và phương thức thanh toán (VnPay).
2. Khách hàng chọn "Thanh toán qua VnPay".
3. Hệ thống chuyển hướng đến cổng thanh toán VnPay.
4. Khách hàng nhập thông tin thanh toán (ví dụ: thông tin thẻ ngân hàng hoặc quét mã QR).
5. VnPay xử lý thanh toán và trả về kết quả.
6. Hệ thống nhận kết quả từ VnPay, hiển thị thông báo "Thanh toán thành công" và chuyển hướng về chức năng "Đặt hàng".  
   **Luồng phụ (Alternative Flows):**

* **A1: Áp dụng mã giảm giá (<<extend>>):**
  1. Từ bước 1, khách hàng nhập mã giảm giá (nếu có).
  2. Hệ thống kiểm tra mã và áp dụng giảm giá, cập nhật tổng tiền.
  3. Quay lại bước 1.  
     **Luồng ngoại lệ (Exception Flows):**
* **E1: Mã giảm giá không hợp lệ:**
  1. Nếu mã giảm giá không hợp lệ, hệ thống hiển thị thông báo "Mã giảm giá không hợp lệ" và yêu cầu nhập lại.
* **E2: Thanh toán thất bại:**
  1. Nếu VnPay trả về kết quả thất bại (do lỗi mạng, tài khoản không đủ tiền, v.v.), hệ thống hiển thị thông báo "Thanh toán thất bại, vui lòng thử lại" và quay lại bước 2.  
     **Điều kiện hậu:**
* Thanh toán thành công, đơn hàng được xác nhận.
* Nếu thanh toán thất bại, đơn hàng không được tạo

### Đặc tả use case Quản lý sản phẩm

**Tên Use Case:** Quản lý sản phẩm  
**Tóm tắt:** Admin sử dụng chức năng này để quản lý danh sách sản phẩm cà phê trên website (thêm, chỉnh sửa, xóa sản phẩm).  
**Actor:** Admin  
**Điều kiện tiên quyết:**

* Admin đã đăng nhập vào hệ thống với quyền quản trị.  
  **Luồng chính (Main Flow):**

1. Admin truy cập chức năng "Quản lý sản phẩm" trên hệ thống.
2. Hệ thống hiển thị danh sách tất cả sản phẩm (tên, giá, số lượng tồn kho, trạng thái).
3. Admin xem danh sách để có cái nhìn tổng quan.  
   **Luồng phụ (Alternative Flows):**

* **A1: Thêm sản phẩm mới (<<extend>>):**
  1. Từ bước 2, Admin chọn "Thêm sản phẩm mới".
  2. Admin nhập thông tin sản phẩm (tên, giá, mô tả, hình ảnh, số lượng tồn kho).
  3. Hệ thống lưu sản phẩm và cập nhật danh sách.
* **A2: Chỉnh sửa sản phẩm (<<extend>>):**
  1. Từ bước 2, Admin chọn một sản phẩm và nhấn "Chỉnh sửa".
  2. Admin cập nhật thông tin (ví dụ: thay đổi giá từ 100.000 VNĐ thành 120.000 VNĐ).
  3. Hệ thống lưu thay đổi và cập nhật danh sách.
* **A3: Xóa sản phẩm (<<extend>>):**
  1. Từ bước 2, Admin chọn một sản phẩm và nhấn "Xóa".
  2. Hệ thống xác nhận xóa và cập nhật danh sách.  
     **Luồng ngoại lệ (Exception Flows):**
* **E1: Danh sách sản phẩm trống:**
  1. Nếu không có sản phẩm nào, hệ thống hiển thị thông báo "Chưa có sản phẩm nào".
* **E2: Thông tin sản phẩm không hợp lệ:**
  1. Nếu Admin nhập thông tin không hợp lệ (ví dụ: giá âm), hệ thống báo lỗi và yêu cầu nhập lại.  
     **Điều kiện hậu:**
* Danh sách sản phẩm được cập nhật (thêm, chỉnh sửa, hoặc xóa).
* Khách hàng có thể thấy sản phẩm mới trên website (nếu thêm hoặc chỉnh sửa).

### Đặc tả use case Quản lý đơn hàng

**Tên Use Case:** Quản lý đơn hàng  
**Tóm tắt:** Admin sử dụng chức năng này để quản lý các đơn hàng của khách hàng (xác nhận, cập nhật trạng thái, hủy đơn).  
**Actor:** Admin  
**Điều kiện tiên quyết:**

* Admin đã đăng nhập vào hệ thống với quyền quản trị.
* Có ít nhất một đơn hàng trong hệ thống.  
  **Luồng chính (Main Flow):**

1. Admin truy cập chức năng "Quản lý đơn hàng" trên hệ thống.
2. Hệ thống hiển thị danh sách tất cả đơn hàng (mã đơn, tên khách hàng, tổng tiền, trạng thái).
3. Admin xem danh sách để có cái nhìn tổng quan.  
   **Luồng phụ (Alternative Flows):**

* **A1: Xác nhận đơn hàng (<<extend>>):**
  1. Từ bước 2, Admin chọn một đơn hàng và nhấn "Xác nhận".
  2. Hệ thống cập nhật trạng thái đơn hàng thành "Đã xác nhận" và thông báo cho khách hàng (qua email, nếu có).
* **A2: Cập nhật trạng thái đơn hàng (<<extend>>):**
  1. Từ bước 2, Admin chọn một đơn hàng và cập nhật trạng thái (ví dụ: từ "Đã xác nhận" thành "Đang giao hàng").
  2. Hệ thống lưu thay đổi và thông báo cho khách hàng.
* **A3: Hủy đơn hàng (<<extend>>):**
  1. Từ bước 2, Admin chọn một đơn hàng và nhấn "Hủy".
  2. Hệ thống cập nhật trạng thái thành "Đã hủy" và thông báo cho khách hàng.  
     **Luồng ngoại lệ (Exception Flows):**
* **E1: Danh sách đơn hàng trống:**
  1. Nếu không có đơn hàng nào, hệ thống hiển thị thông báo "Chưa có đơn hàng nào".
* **E2: Đơn hàng không hợp lệ:**
  1. Nếu Admin cố gắng cập nhật trạng thái của một đơn hàng không tồn tại, hệ thống báo lỗi.  
     **Điều kiện hậu:**
* Trạng thái đơn hàng được cập nhật (xác nhận, đang giao, hủy, v.v.).
* Khách hàng nhận được thông báo về trạng thái đơn hàng.

### Đặc tả use case Quản lý ưu đãi

**Tên Use Case:** Quản lý ưu đãi  
**Tóm tắt:** Admin sử dụng chức năng này để quản lý các chương trình khuyến mãi (tạo, chỉnh sửa, xóa, xem, kích hoạt/ngưng kích hoạt).  
**Actor:** Admin  
**Điều kiện tiên quyết:**

* Admin đã đăng nhập vào hệ thống với quyền quản trị.  
  **Luồng chính (Main Flow):**

1. Admin truy cập chức năng "Quản lý ưu đãi" trên hệ thống.
2. Hệ thống hiển thị danh sách tất cả ưu đãi (tên, giá trị giảm, thời gian áp dụng, trạng thái).
3. Admin xem danh sách để có cái nhìn tổng quan.  
   **Luồng phụ (Alternative Flows):**

* **A1: Tạo ưu đãi (<<extend>>):**
  1. Từ bước 2, Admin chọn "Tạo ưu đãi mới".
  2. Admin nhập thông tin (tên, giá trị giảm, thời gian áp dụng, điều kiện áp dụng).
  3. Hệ thống lưu ưu đãi và cập nhật danh sách.
* **A2: Chỉnh sửa ưu đãi (<<extend>>):**
  1. Từ bước 2, Admin chọn một ưu đãi và nhấn "Chỉnh sửa".
  2. Admin cập nhật thông tin (ví dụ: tăng giá trị giảm từ 10% lên 15%).
  3. Hệ thống lưu thay đổi và cập nhật danh sách.
* **A3: Xóa ưu đãi (<<extend>>):**
  1. Từ bước 2, Admin chọn một ưu đãi và nhấn "Xóa".
  2. Hệ thống xác nhận xóa và cập nhật danh sách.
* **A4: Kích hoạt/ngưng kích hoạt ưu đãi (<<extend>>):**
  1. Từ bước 2, Admin chọn một ưu đãi và nhấn "Kích hoạt" hoặc "Ngưng kích hoạt".
  2. Hệ thống cập nhật trạng thái và hiển thị lại danh sách.  
     **Luồng ngoại lệ (Exception Flows):**
* **E1: Danh sách ưu đãi trống:**
  1. Nếu không có ưu đãi nào, hệ thống hiển thị thông báo "Chưa có ưu đãi nào".
* **E2: Thông tin ưu đãi không hợp lệ:**
  1. Nếu Admin nhập thông tin không hợp lệ (ví dụ: thời gian áp dụng đã qua), hệ thống báo lỗi và yêu cầu nhập lại.  
     **Điều kiện hậu:**
* Danh sách ưu đãi được cập nhật (thêm, chỉnh sửa, xóa, kích hoạt/ngưng kích hoạt).
* Khách hàng có thể thấy ưu đãi mới trên website (nếu được kích hoạt).

### Đặc tả use case Đánh giá sản phẩm

**Tên Use Case:** Đánh giá sản phẩm  
**Tóm tắt:** Khách hàng để lại đánh giá và phản hồi về sản phẩm cà phê sau khi mua hàng.  
**Actor:** Khách hàng  
**Điều kiện tiên quyết:**

* Khách hàng đã đăng nhập vào hệ thống.
* Khách hàng đã mua và nhận sản phẩm (đơn hàng ở trạng thái "Đã giao").  
  **Luồng chính (Main Flow):**

1. Khách hàng truy cập trang chi tiết đơn hàng hoặc trang sản phẩm đã mua.
2. Hệ thống hiển thị tùy chọn "Đánh giá sản phẩm".
3. Khách hàng chọn số sao (1-5) và nhập nội dung phản hồi (ví dụ: "Cà phê rất thơm, nhưng giao hàng hơi chậm").
4. Khách hàng nhấn "Gửi đánh giá".
5. Hệ thống lưu đánh giá và hiển thị thông báo "Đánh giá của bạn đã được gửi".  
   **Luồng phụ (Alternative Flows):**

* **A1: Chỉnh sửa đánh giá:**
  1. Từ bước 1, nếu khách hàng đã đánh giá trước đó, hệ thống hiển thị tùy chọn "Chỉnh sửa đánh giá".
  2. Khách hàng cập nhật số sao hoặc nội dung phản hồi.
  3. Hệ thống lưu thay đổi và thông báo "Đánh giá đã được cập nhật".  
     **Luồng ngoại lệ (Exception Flows):**
* **E1: Đánh giá không hợp lệ:**
  1. Nếu khách hàng không chọn số sao hoặc để trống nội dung, hệ thống báo lỗi "Vui lòng nhập đầy đủ thông tin" và yêu cầu nhập lại.
* **E2: Đơn hàng chưa được giao:**
  1. Nếu đơn hàng chưa ở trạng thái "Đã giao", hệ thống hiển thị thông báo "Bạn chỉ có thể đánh giá sau khi nhận hàng".  
     **Điều kiện hậu:**
* Đánh giá của khách hàng được lưu vào hệ thống.
* Đánh giá được hiển thị trên trang chi tiết sản phẩm (nếu Admin không ẩn).

## Yêu cầu phi chức năng

### Hiệu năng (Performance)

* Hệ thống phải có khả năng xử lý ít nhất 1000 giao dịch mỗi giây mà không bị chậm trễ.
* Thời gian phản hồi của hệ thống không quá 2 giây cho các thao tác thông thường như tìm kiếm sản phẩm, thêm vào giỏ hàng.
* Hệ thống phải hỗ trợ tải đồng thời ít nhất 5000 người dùng cùng lúc mà không ảnh hưởng đến hiệu suất.

### Độ tin cậy (Reliability)

* Hệ thống phải hoạt động ổn định 24/7, thời gian downtime không vượt quá 1% mỗi tháng.
* Dữ liệu đơn hàng, tài khoản khách hàng phải được sao lưu tự động hàng ngày.
* Nếu có sự cố, hệ thống phải có khả năng phục hồi trong vòng 30 phút.

### Tính dễ sử dụng (Usability)

* Giao diện phải thân thiện, trực quan, dễ sử dụng cho cả người dùng không có kiến thức công nghệ.
* Hệ thống phải hỗ trợ đa nền tảng (desktop, mobile).
* Thời gian làm quen với hệ thống cho nhân viên mới không quá 2 giờ.

### Bảo mật (Security)

* Mật khẩu người dùng phải được mã hóa bằng thuật toán Hash (BCrypt, SHA-256,...).
* Hệ thống phải hỗ trợ Xác thực hai yếu tố (2FA) cho tài khoản admin.
* Mọi giao dịch thanh toán phải được mã hóa SSL/TLS để bảo vệ dữ liệu.
* Không lưu trữ trực tiếp thông tin thẻ tín dụng của khách hàng, mà sử dụng bên thứ ba (VNPay, MoMo, PayPal,...).

### Tính dễ bảo trì (Maintainability)

* Hệ thống phải có kiến trúc module, dễ mở rộng và bảo trì.
* Code phải tuân theo các tiêu chuẩn lập trình (RESTful API, SOLID,...) để dễ dàng bảo trì.
* CSDL phải hỗ trợ tự động sao lưu, và có khả năng phục hồi nhanh khi có lỗi.

### Yêu cầu về Công nghệ (Technology Constraints)

* Backend: Spring Boot (Java)
* Frontend: ReactJS
* CSDL: Oracle
* Thanh toán: Tích hợp VNPay, MoMo
* Hạ tầng: Chạy trên Cloud (AWS, Azure, hoặc Google Cloud)
* Giao tiếp API: Sử dụng RESTful API để kết nối giữa frontend và backend

# CÔNG NGHỆ SỬ DỤNG

## React

### Giới thiệu về React

React là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook, dùng để xây dựng giao diện người dùng (UI) theo dạng Single Page Application (SPA). React cho phép lập trình viên tạo các thành phần giao diện (components) có thể tái sử dụng, quản lý trạng thái (state) hiệu quả thông qua các React Hooks (như useState, useEffect), và tối ưu hóa hiệu suất render nhờ cơ chế Virtual DOM [1].

### Vai trò của React trong đồ án

React được sử dụng để xây dựng giao diện người dùng phía Khách hàng (như trang sản phẩm, giỏ hàng, đặt hàng) và phía Admin (như trang quản lý sản phẩm, đơn hàng, ưu đãi). Cụ thể, React đáp ứng các yêu cầu phi chức năng ở Chương 2 như sau:

Tính dễ dùng (2.4.3): React hỗ trợ xây dựng giao diện mượt mà với các thành phần tái sử dụng (ví dụ: component ProductCard để hiển thị sản phẩm, CartItem để hiển thị giỏ hàng). Điều này giúp khách hàng dễ dàng thao tác (đặt hàng, đánh giá sản phẩm) và Admin dễ dàng quản lý (quản lý sản phẩm, đơn hàng).

Hỗ trợ đa thiết bị (2.4.3): React cho phép viết CSS tùy chỉnh hoặc sử dụng các thư viện CSS (như CSS Modules, Styled Components) để tạo giao diện responsive, hoạt động tốt trên nhiều thiết bị (máy tính, điện thoại, máy tính bảng).

Hiệu năng (2.4.1): Cơ chế Virtual DOM của React giúp giảm thời gian render, đảm bảo các trang tải trong vòng 3 giây, đáp ứng yêu cầu về thời gian tải trang.

### So sánh với các công nghệ thay thế

Vue.js: Vue.js là một framework JavaScript nhẹ, dễ học, và có hiệu năng tốt. Tuy nhiên, React có cộng đồng lớn hơn, nhiều tài liệu tham khảo hơn, và được sử dụng rộng rãi trong các dự án thực tế, giúp dễ dàng tìm kiếm hỗ trợ khi phát triển.

Angular: Angular là một framework đầy đủ, nhưng phức tạp hơn React và có đường cong học tập cao hơn, không phù hợp với một đồ án nhỏ như website bán cà phê.

Lý do lựa chọn React: React được chọn vì tính đơn giản, hiệu năng cao (nhờ Virtual DOM), và khả năng tái sử dụng components, phù hợp với quy mô của hệ thống. Ngoài ra, React có cộng đồng lớn và nhiều thư viện hỗ trợ (như Axios để gọi API), giúp tăng tốc quá trình phát triển.

## Spring Boot

### Giới thiệu về Spring Boot

Spring Boot là một framework mã nguồn mở, được xây dựng trên nền tảng Spring Framework, dùng để phát triển các ứng dụng Java theo kiến trúc RESTful. Spring Boot cung cấp các tính năng như auto-configuration, embedded server (Tomcat), và quản lý dependency qua Maven, giúp giảm thời gian cấu hình và tăng tốc phát triển [2].

### Vai trò của Spring Boot trong đồ án

Spring Boot được sử dụng để xây dựng backend của hệ thống, cung cấp các API RESTful cho các chức năng như quản lý sản phẩm, đơn hàng, ưu đãi, và tích hợp thanh toán với VnPay. Cụ thể, Spring Boot đáp ứng các yêu cầu ở Chương 2 như sau:

Hiệu năng (2.4.1): Spring Boot hỗ trợ xử lý đồng thời 50 người dùng với thời gian phản hồi dưới 2 giây, nhờ cơ chế xử lý request hiệu quả và khả năng tích hợp với các công cụ caching (như Spring Cache).

Bảo mật (2.4.5): Spring Boot sử dụng Spring Security để triển khai xác thực (JWT) và phân quyền (role-based access control), đảm bảo chỉ Admin mới truy cập được các API quản trị (quản lý sản phẩm, đơn hàng, ưu đãi).

Tích hợp thanh toán (2.4.1): Spring Boot gọi API của VnPay thông qua các endpoint RESTful, đảm bảo giao dịch hoàn thành trong vòng 5 giây.

Tính dễ bảo trì (2.4.4): Spring Boot tổ chức mã nguồn theo mô hình MVC (Controller, Service, Repository), giúp dễ dàng bảo trì và mở rộng (ví dụ: thêm phương thức thanh toán mới).

### So sánh với các công nghệ thay thế

Node.js (Express): Node.js là một môi trường runtime JavaScript, với Express là framework nhẹ để xây dựng API. Node.js có hiệu năng cao, nhưng không cung cấp các tính năng bảo mật và quản lý dữ liệu mạnh mẽ như Spring Boot (Spring Security, Spring Data JPA).

Django (Python): Django là một framework Python mạnh mẽ, nhưng yêu cầu kiến thức về Python, trong khi nhóm phát triển đã quen thuộc với Java (ngôn ngữ chính của Spring Boot).

Lý do lựa chọn Spring Boot: Spring Boot được chọn vì tính mạnh mẽ, hỗ trợ tốt cho các ứng dụng RESTful, và tích hợp dễ dàng với cơ sở dữ liệu SQL (qua Spring Data JPA). Ngoài ra, Spring Boot cung cấp các tính năng bảo mật (Spring Security) và quản lý dependency (Maven), phù hợp với yêu cầu bảo mật và dễ bảo trì của hệ thống.

## MySQL

### Giới thiệu về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được sử dụng rộng rãi trong các ứng dụng web nhờ tính đơn giản, hiệu năng cao, và khả năng mở rộng. MySQL hỗ trợ các tính năng như giao dịch (transactions), chuẩn hóa dữ liệu, và truy vấn nhanh thông qua indexing [3].

### Vai trò của MySQL trong đồ án

MySQL được sử dụng để lưu trữ và quản lý dữ liệu của hệ thống, bao gồm thông tin sản phẩm, đơn hàng, tài khoản khách hàng, ưu đãi, và phản hồi. Cụ thể, MySQL đáp ứng các yêu cầu ở Chương 2 như sau:

Hiệu năng (2.4.1): MySQL hỗ trợ truy vấn nhanh (dưới 2 giây) cho các bảng dữ liệu nhỏ (10.000 sản phẩm, 50.000 đơn hàng, 5.000 tài khoản khách hàng), nhờ cơ chế indexing và query optimization.

Độ tin cậy (2.4.2): MySQL đảm bảo tính toàn vẹn dữ liệu (data integrity) thông qua các ràng buộc (constraints) như khóa chính (primary key), khóa ngoại (foreign key), và hỗ trợ sao lưu dữ liệu hàng ngày.

Khả năng mở rộng (2.4.7): MySQL có thể mở rộng để hỗ trợ dữ liệu lớn hơn (lên đến hàng triệu bản ghi) bằng cách tối ưu hóa cấu hình hoặc sử dụng các dịch vụ đám mây như AWS RDS.

### So sánh với các công nghệ thay thế

PostgreSQL: PostgreSQL là một RDBMS mã nguồn mở khác, mạnh mẽ hơn MySQL trong các truy vấn phức tạp và hỗ trợ JSONB. Tuy nhiên, MySQL nhẹ hơn, dễ cài đặt hơn, và phù hợp với quy mô nhỏ của hệ thống (website bán cà phê).

SQLite: SQLite là một cơ sở dữ liệu SQL nhẹ, không cần server, nhưng không phù hợp với các hệ thống có nhiều người dùng đồng thời (50 người dùng trong yêu cầu 2.4.1), vì SQLite có hiệu năng thấp hơn khi xử lý tải lớn.

Lý do lựa chọn MySQL: MySQL được chọn vì tính đơn giản, hiệu năng tốt cho các hệ thống nhỏ, và tích hợp dễ dàng với Spring Boot (qua Spring Data JPA). Ngoài ra, MySQL có cộng đồng lớn, dễ triển khai trên các nền tảng như Heroku hoặc AWS, và phù hợp với yêu cầu của một website bán cà phê nhỏ.

## Tích hợp thanh toán VnPay

### Giới thiệu về VnPay

VnPay là một cổng thanh toán trực tuyến phổ biến tại Việt Nam, cung cấp các API để tích hợp thanh toán qua thẻ ngân hàng, ví điện tử, hoặc mã QR. VnPay hỗ trợ giao dịch an toàn với mã hóa dữ liệu và xác thực hai bước [4].

### Vai trò của VnPay trong đồ án

VnPay được tích hợp để hỗ trợ khách hàng thanh toán đơn hàng, đáp ứng yêu cầu chức năng Thanh toán (Chương 2, Use Case 2). Cụ thể:

Hiệu năng (2.4.1): VnPay đảm bảo giao dịch hoàn thành trong vòng 5 giây, đáp ứng yêu cầu về thời gian tích hợp thanh toán.

Bảo mật (2.4.5): VnPay sử dụng giao thức HTTPS và mã hóa dữ liệu để bảo vệ thông tin thanh toán của khách hàng.

Tính dễ dùng (2.4.3): VnPay cung cấp giao diện thanh toán thân thiện, hỗ trợ nhiều phương thức (thẻ, QR), giúp khách hàng dễ dàng thanh toán.

### So sánh với các công nghệ thay thế

Momo: Momo là một cổng thanh toán khác tại Việt Nam, cũng hỗ trợ thanh toán qua ví điện tử và thẻ. Tuy nhiên, VnPay được sử dụng rộng rãi hơn trong các hệ thống thương mại điện tử tại Việt Nam và có tài liệu API rõ ràng, dễ tích hợp với Spring Boot.

PayPal: PayPal là một cổng thanh toán quốc tế, nhưng không phổ biến tại Việt Nam và có phí giao dịch cao hơn, không phù hợp với một website bán cà phê nhỏ.

Lý do lựa chọn VnPay: VnPay được chọn vì tính phổ biến tại Việt Nam, hỗ trợ nhiều phương thức thanh toán, và tích hợp dễ dàng với Spring Boot thông qua API RESTful. Ngoài ra, VnPay đáp ứng yêu cầu bảo mật và hiệu năng của hệ thống.

# THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

## Thiết kế kiến trúc

### Lựa chọn kiến trúc phần mềm

**Lựa chọn kiến trúc**

Hệ thống website bán cà phê nhỏ được thiết kế dựa trên kiến trúc **MVC (Model-View-Controller)**, một kiến trúc phần mềm phổ biến trong các ứng dụng web. Kiến trúc MVC chia hệ thống thành ba thành phần chính:

* **Model:** Quản lý dữ liệu và logic nghiệp vụ của hệ thống, bao gồm các thao tác như truy xuất, lưu trữ dữ liệu, và xử lý các quy tắc kinh doanh.
* **View:** Hiển thị giao diện người dùng, cho phép người dùng (khách hàng và Admin) tương tác với hệ thống thông qua các trang web.
* **Controller:** Xử lý các yêu cầu từ người dùng, điều phối luồng dữ liệu giữa Model và View, đảm bảo hệ thống hoạt động đúng theo yêu cầu.

Kiến trúc MVC được lựa chọn vì tính đơn giản và phù hợp với quy mô của một website bán cà phê nhỏ, đáp ứng các yêu cầu phi chức năng đã nêu ở Chương 2, như tính dễ bảo trì (2.4.4) và khả năng mở rộng (2.4.7). MVC cho phép tách biệt rõ ràng giữa giao diện (View), logic nghiệp vụ (Model), và điều khiển (Controller), giúp dễ dàng bảo trì, mở rộng, và kiểm thử hệ thống. Ví dụ, nếu cần thêm chức năng quản lý kho trong tương lai, chỉ cần bổ sung các thành phần mới vào Model và Controller mà không ảnh hưởng đến View.

**Áp dụng kiến trúc MVC vào hệ thống**

Hệ thống được triển khai với **React** cho phía frontend (View) và **Spring Boot** cho phía backend (Model và Controller), kết hợp với **MySQL** làm cơ sở dữ liệu. Cụ thể, các thành phần của MVC được áp dụng như sau:

* **View:** Được xây dựng bằng React, bao gồm các thành phần giao diện (components) để hiển thị dữ liệu và nhận tương tác từ người dùng. Các thành phần chính bao gồm:
  + Dành cho khách hàng: ProductList (hiển thị danh sách sản phẩm), Cart (giỏ hàng), OrderForm (form đặt hàng), và ReviewForm (form đánh giá sản phẩm).
  + Dành cho Admin: ProductManagement (quản lý sản phẩm), OrderManagement (quản lý đơn hàng), và PromotionManagement (quản lý ưu đãi). React sử dụng các thư viện như Axios để gọi API từ backend, đảm bảo giao diện chỉ tập trung vào việc hiển thị dữ liệu và không chứa logic nghiệp vụ.
* **Controller:** Được triển khai trong Spring Boot, bao gồm các REST Controller để xử lý các yêu cầu HTTP từ React. Các Controller chính bao gồm:
  + ProductController: Xử lý các yêu cầu liên quan đến sản phẩm (ví dụ: GET /api/products để lấy danh sách sản phẩm, POST /api/products để thêm sản phẩm).
  + OrderController: Xử lý các yêu cầu liên quan đến đơn hàng (ví dụ: POST /api/orders để đặt hàng, PUT /api/orders/{id}/confirm để xác nhận đơn hàng).
  + PromotionController: Xử lý các yêu cầu liên quan đến ưu đãi (ví dụ: GET /api/promotions để lấy danh sách ưu đãi, POST /api/promotions để tạo ưu đãi).
  + ReviewController: Xử lý các yêu cầu liên quan đến đánh giá sản phẩm (ví dụ: POST /api/reviews để lưu đánh giá).
* **Model:** Bao gồm các thành phần dữ liệu và logic nghiệp vụ trong Spring Boot, được chia thành các lớp cụ thể:
  + **Entities:** Các lớp Java đại diện cho dữ liệu, như Product, Order, User, Promotion, Review, được ánh xạ với các bảng trong cơ sở dữ liệu MySQL. Ví dụ: lớp Product chứa các thuộc tính như id, name, price, image.
  + **Repositories:** Các interface như ProductRepository, OrderRepository, UserRepository, PromotionRepository, ReviewRepository, sử dụng Spring Data JPA để tương tác với MySQL. Ví dụ: ProductRepository cung cấp các phương thức như findAll(), save(), deleteById().
  + **Services:** Các lớp như ProductService, OrderService, PromotionService, ReviewService, chứa logic nghiệp vụ. Ví dụ: OrderService có phương thức calculateTotalPrice() để tính tổng tiền đơn hàng, bao gồm áp dụng mã giảm giá nếu có.

**Thay đổi và bổ sung**

Kiến trúc MVC được áp dụng không có thay đổi lớn so với lý thuyết, nhưng có một số bổ sung để phù hợp với hệ thống và đáp ứng các yêu cầu ở Chương 2:

* **Tích hợp API RESTful:** Thay vì View giao tiếp trực tiếp với Controller như trong MVC truyền thống, React (View) gọi các API RESTful do Spring Boot (Controller) cung cấp. Điều này giúp tách biệt hoàn toàn frontend và backend, tăng tính linh hoạt và khả năng mở rộng (2.4.7). Ví dụ: React gọi API GET /api/products để lấy danh sách sản phẩm thay vì truy cập trực tiếp vào Model.
* **Bảo mật:** Spring Boot bổ sung tầng bảo mật trong Controller bằng **Spring Security**, triển khai xác thực (JWT) và phân quyền (ROLE\_ADMIN, ROLE\_USER), đáp ứng yêu cầu bảo mật (2.4.5). Ví dụ: chỉ người dùng có vai trò ROLE\_ADMIN mới được gọi API POST /api/products để thêm sản phẩm.
* **Tích hợp VnPay:** Controller trong Spring Boot được bổ sung để xử lý thanh toán qua VnPay, đáp ứng yêu cầu tích hợp thanh toán (Use Case 2: Thanh toán, Chương 2). Ví dụ: PaymentController có endpoint POST /api/payments/vnpay để tạo URL thanh toán.

**Lý do lựa chọn MVC**

* **So sánh với các kiến trúc khác:**
  + **MVP (Model-View-Presenter):** MVP phù hợp hơn với các ứng dụng có logic giao diện phức tạp, nhưng không cần thiết cho một website bán cà phê nhỏ, nơi giao diện chủ yếu là hiển thị dữ liệu và nhận tương tác cơ bản.
  + **Microservices:** Microservices phù hợp với các hệ thống lớn, phân tán, nhưng quá phức tạp cho một website nhỏ với quy mô 50 người dùng đồng thời (2.4.1). Ngoài ra, Microservices yêu cầu quản lý nhiều dịch vụ, không phù hợp với một đồ án tốt nghiệp.
  + **SOA (Service-Oriented Architecture):** SOA cũng phù hợp với hệ thống lớn, cần tích hợp nhiều dịch vụ bên ngoài. Hệ thống của đồ án chỉ cần tích hợp với VnPay, nên SOA không cần thiết.
* **Lý do chọn MVC:** MVC đơn giản, dễ triển khai, và phù hợp với quy mô của hệ thống. Spring Boot hỗ trợ tốt cho MVC (qua các annotation như @Controller, @Service, @Repository), và React dễ dàng tích hợp với các API RESTful do Spring Boot cung cấp. Ngoài ra, MVC đáp ứng yêu cầu tách biệt các thành phần (2.4.4), giúp dễ dàng bảo trì và mở rộng hệ thống trong tương lai.

### Thiết kế tổng quan

A yellow folder with many folders

AI-generated content may be incorrect.

Mô tả biểu đồ gói:

Biểu đồ gói UML được thiết kế theo mô hình phân tầng, bao gồm ba tầng chính: Presentation Layer, Application Layer, và Data Layer, đảm bảo tuân thủ các quy tắc thiết kế (không có phụ thuộc ngược, không bỏ qua tầng, không phụ thuộc lẫn nhau giữa các gói cùng tầng).

* Presentation Layer:
  + api: Chứa các điểm cuối API (endpoints) để giao tiếp với tầng Application, đảm bảo tách biệt giữa giao diện và logic nghiệp vụ.
  + views:
    - customer: Giao diện dành cho khách hàng, hỗ trợ các chức năng như đặt hàng, đánh giá sản phẩm, theo dõi đơn hàng.
    - admin: Giao diện dành cho admin, hỗ trợ quản lý sản phẩm, đơn hàng, và chương trình ưu đãi.
  + Phụ thuộc: Tầng Presentation phụ thuộc vào tầng Application thông qua gói api để gọi các controller xử lý yêu cầu.
* Application Layer:
  + controllers:
    - customer: Xử lý các yêu cầu từ khách hàng (đặt hàng, thanh toán, đánh giá).
    - admin: Xử lý các yêu cầu từ admin (quản lý sản phẩm, đơn hàng, ưu đãi).
  + services:
    - customer: Cung cấp logic nghiệp vụ cho khách hàng (xử lý giỏ hàng, thanh toán, đánh giá).
    - admin: Cung cấp logic nghiệp vụ cho admin (quản lý sản phẩm, cập nhật trạng thái đơn hàng).
  + security: Xử lý xác thực và phân quyền, sử dụng Spring Security để đảm bảo bảo mật cho khách hàng và admin.
  + integrations:
    - vnpay: Tích hợp với VnPay để xử lý thanh toán trực tuyến.
  + Phụ thuộc: Tầng Application phụ thuộc vào tầng Data thông qua gói services để gọi các repository truy xuất dữ liệu. Gói security và integrations/vnpay hỗ trợ các controller và services.
* Data Layer:
  + repositories: Chứa các repository để truy xuất và lưu trữ dữ liệu (sản phẩm, đơn hàng, đánh giá, khách hàng) từ cơ sở dữ liệu MySQL.
  + entities: Định nghĩa các thực thể (entities) như Product, Order, Review, Customer, tương ứng với các bảng trong cơ sở dữ liệu.
  + Phụ thuộc: Tầng Data không phụ thuộc vào tầng trên, chỉ cung cấp dữ liệu thông qua gói repositories cho tầng Application.

## Thiết kế chi tiết

### Thiết kế giao diện

**Đặc tả thông tin về màn hình**

Hệ thống website bán cà phê nhỏ được thiết kế để hoạt động trên cả máy tính để bàn và thiết bị di động, đảm bảo tính tương thích và trải nghiệm người dùng tốt trên nhiều thiết bị. Dưới đây là các thông số kỹ thuật về màn hình mà hệ thống hướng tới:

* Độ phân giải màn hình: Hệ thống hỗ trợ các độ phân giải phổ biến, từ 1366x768 (máy tính để bàn phổ thông) đến 1920x1080 (Full HD), và các độ phân giải nhỏ hơn trên thiết bị di động như 375x667 (iPhone SE) và 414x896 (iPhone 14). Thiết kế giao diện sử dụng layout responsive để tự động điều chỉnh theo độ phân giải.
* Kích thước màn hình: Hệ thống hỗ trợ màn hình từ 4 inch (thiết bị di động nhỏ) đến 27 inch (màn hình máy tính để bàn lớn). Giao diện được tối ưu hóa để hiển thị tốt trên các kích thước màn hình khác nhau, với các thành phần giao diện được sắp xếp linh hoạt.
* Số lượng màu sắc hỗ trợ: Hệ thống hỗ trợ hiển thị màu sắc ở độ sâu 24-bit (16,7 triệu màu), đảm bảo khả năng hiển thị màu sắc phong phú và chính xác trên các thiết bị hiện đại. Các màu sắc được chọn có độ tương phản cao để đảm bảo khả năng đọc tốt, tuân thủ tiêu chuẩn WCAG 2.1 (tỷ lệ tương phản tối thiểu 4.5:1 cho văn bản).
* Tỷ lệ khung hình: Hệ thống hỗ trợ các tỷ lệ khung hình phổ biến như 16:9 (máy tính để bàn), 9:16 (di động dọc), và 4:3 (một số thiết bị cũ). Giao diện được thiết kế để không bị méo hoặc mất nội dung khi thay đổi tỷ lệ khung hình.
* Hướng màn hình: Hệ thống hỗ trợ cả hướng ngang (landscape) và dọc (portrait) trên thiết bị di động, với các thành phần giao diện được điều chỉnh tự động để phù hợp với hướng màn hình.

**Chuẩn hóa thiết kế giao diện**

Để đảm bảo tính nhất quán và trải nghiệm người dùng tốt, tôi đã thiết lập các chuẩn hóa sau khi thiết kế giao diện:

* Thiết kế nút (Button):
* Kích thước: Nút có kích thước tối thiểu 48x48 pixel (theo tiêu chuẩn thiết kế di động), đảm bảo dễ chạm trên màn hình cảm ứng.
* Màu sắc: Nút chính (primary button) sử dụng màu cam (#F5A623) với văn bản trắng, nút phụ (secondary button) sử dụng màu xám nhạt (#D3D3D3) với văn bản đen. Khi hover, nút sáng lên 10% để tạo hiệu ứng tương tác.
* Hình dạng: Nút có góc bo tròn (border-radius: 8px) để tạo cảm giác mềm mại, hiện đại.
* Khoảng cách: Khoảng cách giữa các nút là 16px để tránh chạm nhầm trên thiết bị di động.
* Điều khiển (Controls):
* Ô nhập liệu (input): Ô nhập liệu có viền màu xám nhạt (#E0E0E0), chuyển thành màu cam (#F5A623) khi được focus. Kích thước tối thiểu của ô nhập liệu là 300px (rộng) x 40px (cao) trên máy tính, và 90% chiều rộng màn hình trên di động.
* Dropdown: Dropdown có biểu tượng mũi tên xuống bên phải, với danh sách tùy chọn hiển thị tối đa 5 mục cùng lúc, có thanh cuộn nếu vượt quá.
* Checkbox và Radio: Checkbox và radio có kích thước 20x20 pixel, sử dụng màu cam (#F5A623) khi được chọn.
* Vị trí hiển thị thông điệp phản hồi:
* Thông điệp thành công: Hiển thị ở phía trên cùng của màn hình, nền màu xanh lá (#28A745), văn bản trắng, tồn tại trong 3 giây trước khi tự động biến mất.
* Thông điệp lỗi: Hiển thị ngay dưới ô nhập liệu liên quan (nếu có) hoặc ở phía trên cùng của màn hình, nền màu đỏ (#DC3545), văn bản trắng, tồn tại cho đến khi người dùng đóng.
* Thông điệp cảnh báo: Hiển thị ở phía trên cùng, nền màu vàng (#FFC107), văn bản đen, tồn tại trong 5 giây.
* Phối màu:
* Màu chủ đạo: Cam (#F5A623) – đại diện cho sự ấm áp, thân thiện, phù hợp với chủ đề cà phê.
* Màu nền: Trắng (#FFFFFF) cho nền chính, xám nhạt (#F8F9FA) cho các khu vực phụ.
* Màu văn bản: Đen (#212529) cho văn bản chính, xám đậm (#6C757D) cho văn bản phụ, đảm bảo độ tương phản cao.
* Màu nhấn: Xanh lá (#28A745) cho trạng thái thành công, đỏ (#DC3545) cho lỗi, vàng (#FFC107) cho cảnh báo.
* Bố cục giao diện:
* Thanh điều hướng (navbar): Nằm ở phía trên cùng, cố định (fixed), chiều cao 60px, chứa logo bên trái, menu chính (Sản phẩm, Giỏ hàng, Đăng nhập) bên phải.
* Nội dung chính: Được đặt trong container có chiều rộng tối đa 1200px, căn giữa màn hình, với padding 16px ở hai bên trên di động.
* Footer: Nằm ở phía dưới cùng, chứa thông tin liên hệ, bản quyền, chiều cao 80px, nền xám đậm (#343A40), văn bản trắng.
* Typography:
* Font chữ: Sử dụng font "Roboto" (sans-serif) cho toàn bộ hệ thống, với kích thước chữ 16px cho văn bản chính, 14px cho văn bản phụ, và 24px cho tiêu đề.
* Khoảng cách dòng: Giãn dòng 1.5 để đảm bảo dễ đọc.

**Hình ảnh minh họa thiết kế giao diện**

### Thiết kế lớp

A group of yellow sticky notes

AI-generated content may be incorrect.

A diagram of a diagram

AI-generated content may be incorrect.

Mô tả biểu đồ:

* Đối tượng tham gia:
  + Customer (Người dùng): Khách hàng sử dụng hệ thống.
  + CartPage (Presentation Layer): Giao diện giỏ hàng.
  + OrderController (Application Layer): Xử lý yêu cầu đặt hàng.
  + OrderService (Application Layer): Xử lý logic đặt hàng.
  + Cart (Data Layer): Đối tượng giỏ hàng.
  + Products (Data Layer): Đối tượng sản phẩm.
  + Order (Data Layer): Đối tượng đơn hàng.
  + PayOrderController (Application Layer): Xử lý thanh toán (được thay thế bằng VnPayService trong hệ thống hiện tại).
* Luồng truyền thông điệp:
  + Customer gửi yêu cầu nhấn đặt hàng (handleCheckout()) đến CartPage.
  + CartPage gửi createOrder() đến OrderController để bắt đầu quá trình đặt hàng.
  + OrderController gọi validateCart() trên OrderService để kiểm tra giỏ hàng.
  + OrderService thực hiện vòng lặp [for each item in cart]: 4.1. OrderService gọi getProductById() trên Cart để lấy thông tin sản phẩm. 4.2. Cart trả về thông tin sản phẩm.
  + OrderService tạo đối tượng Order bằng cách gửi tín hiệu <> đến Order.
  + Customer gửi yêu cầu nhập địa chỉ giao hàng (submitDeliveryInfo()) đến DeliveryForm.
  + DeliveryForm gọi submitDeliveryInfo() trên OrderController.
  + OrderController gọi computeShippingCost() trên OrderService để tính phí giao hàng.
  + OrderService gọi generateOrder() trên Order để tạo đơn hàng hoàn chỉnh.
  + Order trả về đơn hàng đã tạo.
  + Customer gửi yêu cầu processPayOrder() đến PayOrderController (trong hệ thống hiện tại, bước này được thay thế bằng gọi VnPayService).
  + PayOrderController gọi initiatePayment() trên PayOrder (được thay thế bằng createPayment() trên VnPayService).
  + PayOrder gọi executePayment() trên PayOrder (được thay thế bằng giao tiếp với VnPayAPI).

A diagram of a diagram

AI-generated content may be incorrect.

Mô tả biểu đồ:

* Đối tượng tham gia:
  + Customer (Người dùng): Khách hàng sử dụng hệ thống.
  + OrderForm (Presentation Layer): Giao diện đơn hàng.
  + OrderController (Application Layer): Xử lý yêu cầu thanh toán.
  + OrderService (Application Layer): Xử lý logic thanh toán.
  + VnPayService (Application Layer): Xử lý tích hợp thanh toán VnPay.
  + OrderRepository (Data Layer): Lưu trữ và cập nhật đơn hàng.
  + VnPayAPI (Hệ thống bên ngoài): Cổng thanh toán VnPay.
* Luồng truyền thông điệp:
  + Customer gửi yêu cầu pay\_via\_VnPay() đến OrderForm để bắt đầu thanh toán.
  + OrderForm gọi getPaymentUrl() trên OrderController để lấy URL thanh toán.
  + OrderController gọi getPaymentUrl() trên OrderService để xử lý logic.
  + OrderService gọi findById() trên OrderRepository để lấy thông tin đơn hàng.
  + OrderRepository trả về đơn hàng.
  + OrderService gọi generatePaymentUrl() trên VnPayService để tạo URL thanh toán.
  + VnPayService gọi createPaymentUrl() trên VnPayAPI để tạo URL.
  + VnPayAPI trả về URL thanh toán.
  + VnPayService trả về URL cho OrderService.
  + OrderService trả về URL cho OrderController.
  + OrderController trả về URL cho OrderForm.
  + OrderForm hiển thị URL để khách hàng chuyển hướng đến cổng thanh toán VnPay.

A diagram of a project

AI-generated content may be incorrect.

A diagram of a project

AI-generated content may be incorrect.

A diagram of a project

AI-generated content may be incorrect.

A diagram of a project

AI-generated content may be incorrect.

A diagram of a project

AI-generated content may be incorrect.

### Thiết kế cơ sở dữ liệu

**Mục tiêu thiết kế**

* Đáp ứng các yêu cầu chức năng của hệ thống: quản lý sản phẩm, đặt hàng, thanh toán (tích hợp VnPay), quản lý khuyến mãi, đánh giá sản phẩm, và bảo mật.
* Đảm bảo tính hợp lý: chuẩn hóa, toàn vẹn dữ liệu, hiệu năng cao.
* Dễ bảo trì và mở rộng: hỗ trợ thêm chức năng mới (như quản lý kho, đa ngôn ngữ) mà không cần thay đổi cấu trúc lớn.
* Sử dụng hệ quản trị cơ sở dữ liệu quan hệ MySQL, phù hợp với Spring Boot và các yêu cầu của hệ thống.

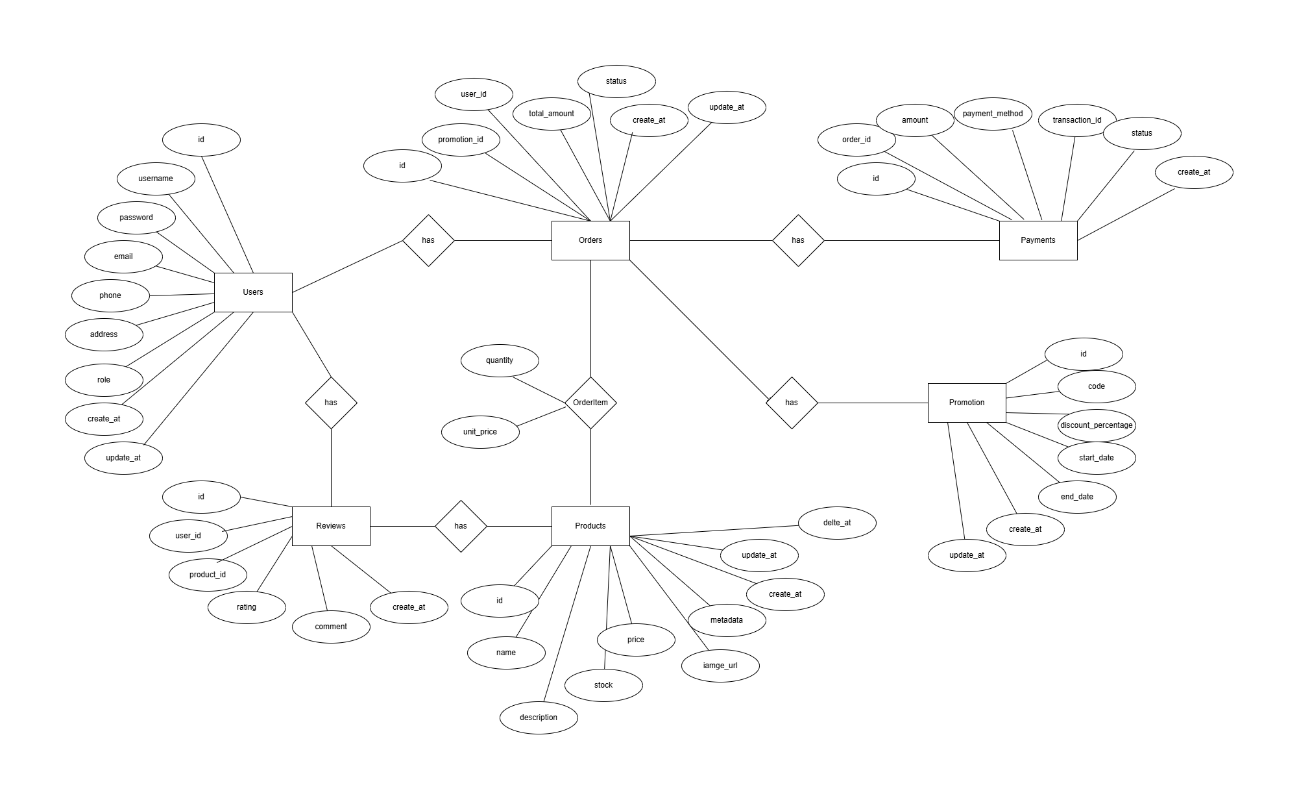
**Biểu đồ thực thể liên kết (E-R Diagram)**

**Xác định các thực thể và quan hệ**

Dựa trên yêu cầu của hệ thống, các thực thể chính bao gồm:

* **Users:** Người dùng (khách hàng và Admin).
* **Products:** Sản phẩm cà phê.
* **Orders:** Đơn hàng.
* **OrderItems:** Mục hàng trong đơn hàng.
* **Promotions:** Khuyến mãi.
* **Reviews:** Đánh giá sản phẩm.
* **Payments:** Thanh toán (liên quan đến VnPay).

**Biểu đồ E-R Diagram**

****

**Hình 4.13: Biểu đồ thực thể liên kết (E-R Diagram)**

**Giải thích biểu đồ E-R:**

* **Thực thể Users:**
  + Thuộc tính: id, username, password, email, phone, address, role, created\_at, updated\_at.
  + Quan hệ:
    - Một User có thể có nhiều Orders (1:N).
    - Một User có thể viết nhiều Reviews (1:N).
* **Thực thể Products:**
  + Thuộc tính: id, name, description, price, stock, image\_url, metadata, created\_at, updated\_at, deleted\_at.
  + Quan hệ:
    - Một Product có thể xuất hiện trong nhiều OrderItems (1:N).
    - Một Product có thể có nhiều Reviews (1:N).
* **Thực thể Orders:**
  + Thuộc tính: id, user\_id, promotion\_id, total\_amount, status, created\_at, updated\_at.
  + Quan hệ:
    - Một Order thuộc về một User (N:1).
    - Một Order có thể áp dụng một Promotion (N:1).
    - Một Order có nhiều OrderItems (1:N).
    - Một Order có một Payment (1:1).
* **Thực thể OrderItems:**
  + Thuộc tính: id, order\_id, product\_id, quantity, unit\_price.
  + Quan hệ:
    - Một OrderItem thuộc về một Order (N:1).
    - Một OrderItem liên quan đến một Product (N:1).
* **Thực thể Promotions:**
  + Thuộc tính: id, code, discount\_percentage, start\_date, end\_date, created\_at, updated\_at.
  + Quan hệ:
    - Một Promotion có thể được áp dụng cho nhiều Orders (1:N).
* **Thực thể Reviews:**
  + Thuộc tính: id, user\_id, product\_id, rating, comment, created\_at.
  + Quan hệ:
    - Một Review thuộc về một User (N:1).
    - Một Review liên quan đến một Product (N:1).
* **Thực thể Payments:**
  + Thuộc tính: id, order\_id, amount, payment\_method, transaction\_id, status, created\_at.
  + Quan hệ:
    - Một Payment thuộc về một Order (1:1).

**Thiết kế cơ sở dữ liệu trong MySQL**

Dựa trên biểu đồ E-R, tôi sẽ thiết kế các bảng trong MySQL, đảm bảo chuẩn hóa đến dạng chuẩn thứ 3 (3NF) và dễ mở rộng.

**Bảng users**

| **Cột** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | BIGINT | Khóa chính, tự động tăng | PRIMARY KEY |
| username | VARCHAR(50) | Tên đăng nhập | UNIQUE, NOT NULL |
| password | VARCHAR(255) | Mật khẩu (mã hóa) | NOT NULL |
| email | VARCHAR(100) | Email | UNIQUE, NOT NULL |
| phone | VARCHAR(20) | Số điện thoại |  |
| address | TEXT | Địa chỉ |  |
| role | ENUM | Vai trò (CUSTOMER, ADMIN) | NOT NULL |
| created\_at | DATETIME | Thời gian tạo | NOT NULL |
| updated\_at | DATETIME | Thời gian cập nhật |  |

**Giải thích:** Bảng users lưu thông tin người dùng, với role để phân biệt khách hàng và Admin. Cột password được mã hóa để đảm bảo bảo mật.

**Bảng products**

| **Cột** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | BIGINT | Khóa chính, tự động tăng | PRIMARY KEY |
| name | VARCHAR(100) | Tên sản phẩm | NOT NULL |
| description | TEXT | Mô tả sản phẩm |  |
| price | DOUBLE | Giá sản phẩm | NOT NULL |
| stock | INT | Số lượng tồn kho | NOT NULL |
| image\_url | VARCHAR(255) | Đường dẫn hình ảnh |  |
| metadata | JSON | Thông tin bổ sung |  |
| created\_at | DATETIME | Thời gian tạo | NOT NULL |
| updated\_at | DATETIME | Thời gian cập nhật |  |
| deleted\_at | DATETIME | Thời gian xóa mềm |  |

**Giải thích:** Bảng products lưu thông tin sản phẩm, với metadata (JSON) để hỗ trợ mở rộng (ví dụ: lưu kích thước, hương vị). Cột deleted\_at hỗ trợ xóa mềm.

**Bảng orders**

| **Cột** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | BIGINT | Khóa chính, tự động tăng | PRIMARY KEY |
| user\_id | BIGINT | Khóa ngoại đến users | FOREIGN KEY |
| promotion\_id | BIGINT | Khóa ngoại đến promotions | FOREIGN KEY |
| total\_amount | DOUBLE | Tổng tiền | NOT NULL |
| status | ENUM | Trạng thái (PENDING, COMPLETED, CANCELLED) | NOT NULL |
| created\_at | DATETIME | Thời gian tạo | NOT NULL |
| updated\_at | DATETIME | Thời gian cập nhật |  |

**Giải thích:** Bảng orders lưu thông tin đơn hàng, với user\_id và promotion\_id là khóa ngoại để liên kết với users và promotions.

**Bảng order\_items**

| **Cột** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | BIGINT | Khóa chính, tự động tăng | PRIMARY KEY |
| order\_id | BIGINT | Khóa ngoại đến orders | FOREIGN KEY |
| product\_id | BIGINT | Khóa ngoại đến products | FOREIGN KEY |
| quantity | INT | Số lượng | NOT NULL |
| unit\_price | DOUBLE | Giá tại thời điểm đặt | NOT NULL |

**Giải thích:** Bảng order\_items lưu các mục hàng trong đơn hàng, với unit\_price để lưu giá tại thời điểm đặt (tránh ảnh hưởng khi giá sản phẩm thay đổi).

**Bảng promotions**

| **Cột** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | BIGINT | Khóa chính, tự động tăng | PRIMARY KEY |
| code | VARCHAR(20) | Mã khuyến mãi | UNIQUE, NOT NULL |
| discount\_percentage | DOUBLE | Phần trăm giảm giá | NOT NULL |
| start\_date | DATETIME | Ngày bắt đầu | NOT NULL |
| end\_date | DATETIME | Ngày kết thúc | NOT NULL |
| created\_at | DATETIME | Thời gian tạo | NOT NULL |
| updated\_at | DATETIME | Thời gian cập nhật |  |

**Giải thích:** Bảng promotions lưu thông tin khuyến mãi, với start\_date và end\_date để kiểm soát thời gian hiệu lực.

**Bảng reviews**

| **Cột** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | BIGINT | Khóa chính, tự động tăng | PRIMARY KEY |
| user\_id | BIGINT | Khóa ngoại đến users | FOREIGN KEY |
| product\_id | BIGINT | Khóa ngoại đến products | FOREIGN KEY |
| rating | INT | Điểm đánh giá (1-5) | NOT NULL |
| comment | TEXT | Bình luận |  |
| created\_at | DATETIME | Thời gian tạo | NOT NULL |

**Giải thích:** Bảng reviews lưu đánh giá sản phẩm, với rating để lưu điểm đánh giá và comment để lưu bình luận.

**Bảng payments**

| **Cột** | **Kiểu dữ liệu** | **Mô tả** | **Ràng buộc** |
| --- | --- | --- | --- |
| id | BIGINT | Khóa chính, tự động tăng | PRIMARY KEY |
| order\_id | BIGINT | Khóa ngoại đến orders | FOREIGN KEY |
| amount | DOUBLE | Số tiền thanh toán | NOT NULL |
| payment\_method | ENUM | Phương thức (VNPAY) | NOT NULL |
| transaction\_id | VARCHAR(50) | Mã giao dịch VnPay |  |
| status | ENUM | Trạng thái (SUCCESS, FAILED) | NOT NULL |
| created\_at | DATETIME | Thời gian tạo | NOT NULL |

**Giải thích:** Bảng payments lưu thông tin thanh toán, với transaction\_id để lưu mã giao dịch từ VnPay.

**Chuẩn hóa cơ sở dữ liệu**

* **Dạng chuẩn thứ 1 (1NF):** Đảm bảo tất cả các cột chứa giá trị nguyên tử. Ví dụ: metadata trong products được lưu dưới dạng JSON, nhưng vẫn là một giá trị nguyên tử.
* **Dạng chuẩn thứ 2 (2NF):** Loại bỏ phụ thuộc hàm một phần. Tất cả các bảng đều có khóa chính duy nhất (id), và các thuộc tính khác phụ thuộc hoàn toàn vào khóa chính.
* **Dạng chuẩn thứ 3 (3NF):** Loại bỏ phụ thuộc bắc cầu. Ví dụ: Trong bảng orders, total\_amount phụ thuộc trực tiếp vào id, không qua thuộc tính trung gian nào khác.

**Tối ưu và hỗ trợ mở rộng**

* **Chỉ mục (Index):**
  + Tạo chỉ mục cho các cột thường xuyên được truy vấn:
    - user\_id trong orders: CREATE INDEX idx\_user\_id ON orders(user\_id);
    - product\_id trong order\_items: CREATE INDEX idx\_product\_id ON order\_items(product\_id);
    - code trong promotions: CREATE INDEX idx\_code ON promotions(code);
  + Giúp tăng tốc độ truy vấn, đặc biệt với các bảng lớn.

| **Mã lỗi** | **Mô tả** | **HTTP Status** |
| --- | --- | --- |
| 1000 | Success (Ok, Created, Updated, Deleted) | 200, 201 |
| 1001 | Invalid username or password | 401 |
| 1002 | Account is locked | 403 |
| 1003 | Username already exists | 400 |
| 1004 | Email already exists | 400 |
| 1005 | User not found | 404 |
| 1006 | User not verified by email verification code | 403 |
| 2001 | Invalid parameters | 400 |
| 2002 | Product not found | 404 |
| 3001 | Product out of stock | 400 |
| 3002 | Cart is empty | 400, 404 |
| 4001 | Invalid promotion code | 400 |
| 4002 | Order not found | 404 |
| 4003 | Access denied | 403 |
| 4004 | Invalid order status | 400 |
| 4005 | Promotion code already exists | 400 |
| 4006 | Promotion not found | 404 |
| 5001 | Order not delivered | 400 |
| 5002 | Review not found | 404 |
| 6001 | VnPay error | 400 |
| 6002 | Invalid callback parameters | 400 |
| 9999 | Internal server error | 500 |

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Trần Bách, Lưới điện và hệ thống điện, Nhà xuất bản Khoa học Kỹ thuật, 2004. |
| [2] | Abe Masayuki, “A Practical Approach to Accurate Fault Location on Extra High Voltage Teed Feeders,” *IEEE Transaction on Power Delivery,* pp. 159-168, 1995. |
| [3] | Microsoft, "Add citations in a Word document," 2017. |

# PHỤ LỤC

1. **Chi tiết số liệu thí nghiệm**

Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có).

1. **Chi tiết các bước tính toán**

Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có).

1. **Chi tiết sơ đồ mô phỏng**

Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trình phụ lục tại đây (nếu có). Trìn