|  |
| --- |
| BỘ CÔNG THƯƠNG  **TRƯỜNG ĐẠI HỌC SAO ĐỎ**  **[21-08-2014_10** |
| **ĐỒ ÁN TỐT NGHIỆP** |
| **Ngành đào tạo: Công nghệ thông tin** |
|  |
| **TÊN ĐỀ TÀI: NGHIÊN CỨU, XÂY DỰNG ỨNG DỤNG ĐẶT ĐỒ ĂN**  **TRỰC TUYẾN TRÊN NỀN TẢNG DI ĐỘNG** |
|  |
| **Họ và tên sinh viên: Nguyễn Văn Hiến**  **Nguyễn Thị Mai** |
| **Lớp, khoá: DK12-CNTT2, Đại học khóa 12**  **Giảng viên hướng dẫn: ThS. Nguyễn Thị Bích Ngọc** |
|  |
|  |
|  |
| HẢI PHÒNG – NĂM 2025 |

# LỜI CAM ĐOAN

Chúng tôi xin cam đoan các kết quả đưa ra trong đồ án tốt nghiệp này là các kết quả thu được trong quá trình nghiên cứu, thực nghiệm của chúng tôi dưới sự hướng dẫn của Ths Nguyễn Thị Bích Ngọc, không sao chép bất kỳ kết quả nghiên cứu nào của các tác giả khác.

Nội dung nghiên cứu có tham khảo và sử dụng một số thông tin, tài liệu từ các nguồn tài liệu đã được liệt kê trong danh mục các tài liệu tham khảo.

Nếu sai chúng tôi xin chịu mọi hình thức kỷ luật theo quy định.

*Hải Phòng, ngày 30 tháng 11 năm 2025*

**Sinh viên thực hiện**

*(Ký, ghi rõ họ và tên)*

Nguyễn Văn Hiến

Nguyễn Thị Mai

# LỜI CẢM ƠN

Để hoàn thành quá trình nghiên cứu và hoàn thiện đề tài “Nghiên cứu, xây dựng ứng dụng đặt đồ ăn trực tuyến trên nền tảng di động”, lời đầu tiên chúng em xin chân thành cảm ơn sâu sắc đến cô Nguyễn Thị Bích Ngọc thuộc Khoa Công nghệ thông tin. Cô đã trực tiếp chỉ bảo và hướng dẫn chúng em trong suốt quá trình nghiên cứu để chúng em hoàn thiện đề tài này. Xin cảm ơn cô vì những kiến thức, những kĩ năng và hơn hết là tinh thần trách nhiệm với học trò của cô. Ngoài ra, chúng em xin chân thành cảm ơn các Thầy, Cô trong Khoa Công nghệ thông tin đã đóng góp những ý kiến quý báu cho đề tài của chúng em.

Trong quá trình thực hiện bài báo cáo, nếu như có thiếu sót, chúng em xin các Thầy Cô trong khoa có thể tiếp tục hỗ trợ, đóng góp ý kiến, phản hồi cũng như cung cấp thêm những hướng dẫn chi tiết cụ thể về mặt chuyển môn để chúng em có thể thực hiện được nội dung được hoàn chỉnh nhất.

Cuối cùng, chúng em xin cảm ơn những người thân, bạn bè đã luôn bên chúng em, động viên chúng em hoàn thành đề tài này.

# MỤC LỤC

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 7](#_Toc215741801)

[1.1. Tổng quan về Java 7](#_Toc215741802)

[*1.1.1.* *Khái niệm về Java* 7](#_Toc215741803)

[*1.1.2.* *Các tính năng đặc trưng của Java* 7](#_Toc215741804)

[*1.1.3.* *Ưu và nhược điểm của Java* 9](#_Toc215741805)

[1.2. Tổng quan về Android Studio 10](#_Toc215741806)

[*1.2.1.* *Khái niệm về Android Studio* 10](#_Toc215741807)

[*1.2.2.* *Thành phần trong Android Studio* 11](#_Toc215741808)

[*1.2.3.* *Ưu và nhược điểm của Android Studio* 13](#_Toc215741809)

[1.3. Tổng quan về Firebase 14](#_Toc215741810)

[*1.3.1.* *Khái niệm về Firebase* 14](#_Toc215741811)

[*1.3.2.* *Cơ chế hoạt động của Firebase* 15](#_Toc215741812)

[*1.3.3.* *Ưu và nhược điểm của Firebase* 15](#_Toc215741813)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 17](#_Toc215741814)

[2.1. Khảo sát 17](#_Toc215741815)

[2.2. Phân tích hệ thống 19](#_Toc215741816)

[*2.2.1.* *Yêu cầu chức năng* 19](#_Toc215741817)

[*2.2.2.* *Yêu cầu phi chức năng* 20](#_Toc215741818)

[2.3. Mô tả chi tiết Use case 21](#_Toc215741819)

[2.4. Biểu đồ Use case 24](#_Toc215741820)

[2.5. Thiết kế cơ sở dữ liệu 28](#_Toc215741821)

[CHƯƠNG 3. XÂY DỰNG CHƯƠNG TRÌNH 36](#_Toc215741822)

[3.1. Xây dựng giao diện người dùng 36](#_Toc215741823)

[*3.1.1.* *Phân loại người dùng hệ thống* 36](#_Toc215741824)

[*3.1.2.* *Giao diện khách hàng* 36](#_Toc215741825)

[*3.1.3.* *Giao diện nhà hàng* 36](#_Toc215741826)

[*3.1.4.* *Giao diện quản trị viên* 37](#_Toc215741827)

[3.2. Kết quả thực hiện 38](#_Toc215741828)

[*3.2.1.* *Giao diện khởi chạy ứng dụng* 38](#_Toc215741829)

[*3.2.2.* *Giao diện đăng nhập và đăng ký* 39](#_Toc215741830)

[*3.2.3.* *Giao diện người dùng* 41](#_Toc215741831)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 56](#_Toc215741832)

[TÀI LIỆU THAM KHẢO 58](#_Toc215741833)

# DANH MỤC HÌNH ẢNH

[Hình 1.1. Khái niệm về Java 7](#_Toc215741736)

[Hình 1.2. Các tính năng đặng trưng của Java 8](#_Toc215741737)

[Hình 1.3. Ưu và nhược điểm của Java 9](#_Toc215741738)

[Hình 1.4. Khái niệm Android Studio 10](#_Toc215741739)

[Hình 1.5. Lịch sử phát triển Android Studio 11](#_Toc215741740)

[Hình 1.6. Khái niệm về Firebase 14](#_Toc215741741)

[Hình 2.1. Biểu đồ use case tổng quát 24](#_Toc215741742)

[Hình 2.2. Biểu đồ use case admin 25](#_Toc215741743)

[Hình 2.3. Biểu đồ use case khách hàng 26](#_Toc215741744)

[Hình 2.4. Biểu đồ use case nhà hàng 27](#_Toc215741745)

[Hình 2.5. Mô tả dữ liệu yêu thích 28](#_Toc215741746)

[Hình 2.6. Mô tả dữ liệu món ăn 29](#_Toc215741747)

[Hình 2.7. Mô tả dữ liệu thông báo 30](#_Toc215741748)

[Hình 2.8. Mô tả dữ liệu đơn hàng 31](#_Toc215741749)

[Hình 2.9. Mô tả dữ liệu nhà hàng 32](#_Toc215741750)

[Hình 2.10. Mô tả dữ liệu đánh giá 33](#_Toc215741751)

[Hình 2.11. Mô tả dữ liệu tài khoản 34](#_Toc215741752)

[Hình 2.12. Mô tả dữ liệu vouchers 35](#_Toc215741753)

[Hình 3.1. Giao diện khởi chạy ứng dụng 38](#_Toc215741754)

[Hình 3.2. Giao diện đăng nhập tài khoản 39](#_Toc215741755)

[Hình 3.3. Giao diện đăng ký tài khoản 40](#_Toc215741756)

[Hình 3.4. Giao diện trang chủ 41](#_Toc215741757)

[Hình 3.5. Giao diện chi tiết sản phẩm 42](#_Toc215741758)

[Hình 3.6. Giao diện chọn nhà hàng và chọn món 43](#_Toc215741759)

[Hình 3.7. Giao diện giỏ hàng 44](#_Toc215741760)

[Hình 3.8. Giao diện đặt hàng và thanh toán 45](#_Toc215741761)

[Hình 3.9. Giao diện tra cứu tình trạng đơn hàng 46](#_Toc215741762)

[Hình 3.10. Giao diện yêu thích sản phẩm 47](#_Toc215741763)

[Hình 3.11. Giao diện thống kê mua hàng 48](#_Toc215741764)

[Hình 3.12. Giao diện thiết lập nhà hàng 49](#_Toc215741765)

[Hình 3.13. Giao diện quản lý món ăn 50](#_Toc215741766)

[Hình 3.14. Giao diện quản lý hóa đơn và thống kê 51](#_Toc215741767)

[Hình 3.15. Giao diện quản trị thống kê 52](#_Toc215741768)

[Hình 3.16. Giao diện quản lý tài khoản 53](#_Toc215741769)

[Hình 3.17. Giao diện quản lý duyệt nhà hàng 54](#_Toc215741770)

[Hình 3.18. Giao diện quản lý duyệt món ăn 55](#_Toc215741771)

# DANH MỤC BẢNG

[Bảng 2.1. Bảng mô tả tác nhân 21](#_Toc215741712)

[Bảng 2.2. Bảng mô tả Use case 21](#_Toc215741713)

[Bảng 2.3. Bảng dữ liệu yêu thích 28](#_Toc215741714)

[Bảng 2.4. Bảng dữ liệu món ăn 28](#_Toc215741715)

[Bảng 2.5. Bảng dữ liệu thông báo 29](#_Toc215741716)

[Bảng 2.6. Bảng dữ liệu đơn hàng 30](#_Toc215741717)

[Bảng 2.7. Bảng dữ liệu nhà hàng 32](#_Toc215741718)

[Bảng 2.8. Bảng dữ liệu đánh giá 32](#_Toc215741719)

[Bảng 2.9. Bảng dữ liệu tài khoản 33](#_Toc215741720)

[Bảng 2.10. Bảng dữ liệu vouchers 34](#_Toc215741721)

# MỞ ĐẦU

1. **Tính cấp thiết của đề tài**

Sự phát triển mạnh mẽ của công nghệ di động và xu hướng số hóa các dịch vụ đời sống đã tạo điều kiện cho các ứng dụng đặt đồ ăn trực tuyến trở thành nhu cầu thiết yếu của người dùng hiện đại. Trong bối cảnh đó, người dùng mong muốn tiếp cận món ăn nhanh chóng, tiện lợi, hạn chế thời gian chờ đợi và giảm thao tác thủ công. Các mô hình đặt đồ ăn truyền thống qua điện thoại hoặc trực tiếp tại quán đang bộc lộ nhiều hạn chế như khó kiểm soát đơn hàng, thiếu minh bạch, mất thời gian trao đổi và dễ xảy ra sai sót. Vì vậy, việc lựa chọn đề tài “Xây dựng ứng dụng đặt đồ ăn trực tuyến trên nền tảng di động” mang tính thiết thực cao và phù hợp với xu thế phát triển của thị trường dịch vụ số hiện nay.

Bên cạnh nhu cầu từ phía người dùng, các cửa hàng và quán ăn cũng đang đối mặt với yêu cầu cấp thiết trong việc tối ưu hóa quy trình quản lý đơn hàng. Các phương pháp ghi nhận thủ công khiến quá trình vận hành thiếu đồng bộ, khó theo dõi doanh thu và mất nhiều thời gian xử lý. Việc xây dựng ứng dụng đặt đồ ăn trực tuyến giúp nhà hàng quản lý món ăn, kiểm soát đơn hàng, cập nhật trạng thái cũng như theo dõi hoạt động kinh doanh một cách khoa học và trực quan.

Về mặt kỹ thuật, đề tài tận dụng sức mạnh của Android Studio, ngôn ngữ Java và hệ sinh thái Firebase để xây dựng ứng dụng có hiệu năng tốt và khả năng mở rộng. Firebase Authentication hỗ trợ xác thực người dùng an toàn và đơn giản, trong khi Firebase Storage được sử dụng để lưu trữ và quản lý hình ảnh món ăn một cách hiệu quả. Sự kết hợp của các công nghệ này giúp hệ thống vận hành ổn định, bảo mật và phù hợp triển khai thực tế.

1. **Mục tiêu nghiên cứu**

***Mục tiêu tổng quát:***

Nghiên cứu hướng tới việc xây dựng một ứng dụng đặt đồ ăn trực tuyến trên nền tảng Android, cho phép người dùng thực hiện đầy đủ các thao tác từ xem món, đặt hàng, thanh toán đến theo dõi trạng thái đơn. Đồng thời, hệ thống cũng cung cấp các chức năng hỗ trợ nhà hàng quản lý thực đơn, xử lý đơn hàng và theo dõi doanh thu. Bên cạnh đó, Admin có thể giám sát toàn bộ hoạt động của hệ thống, đảm bảo tính minh bạch, an toàn và ổn định khi vận hành.

***Mục tiêu cụ thể:***

Về mặt công nghệ: Phát triển ứng dụng trên nền tảng Android sử dụng ngôn ngữ Java trong môi trường Android Studio, bảo đảm tương thích với đa số thiết bị phổ biến. Ứng dụng Firebase Authentication nhằm hỗ trợ chức năng đăng ký, đăng nhập và quản lý các nhóm vai trò (Khách hàng – Nhà hàng – Admin), sử dụng Cloud Firestore để lưu trữ dữ liệu động như thông tin người dùng, thực đơn, đơn hàng, đánh giá và thống kê, đồng thời sử dụng Firebase Storage để quản lý tệp hình ảnh món ăn, đảm bảo tốc độ tải nhanh và tối ưu dung lượng.

Về mặt chức năng: Hệ thống được xây dựng nhằm đáp ứng nhu cầu sử dụng của ba nhóm đối tượng chính: Khách hàng, Nhà hàng và Admin. Đối với Khách hàng, ứng dụng cho phép xem danh sách món ăn theo từng danh mục, tìm kiếm theo tên hoặc bộ lọc, thêm món vào giỏ hàng và đặt đơn, đồng thời theo dõi trạng thái xử lý đơn hàng. Ngoài ra, người dùng còn có thể đánh giá món ăn nhằm góp phần nâng cao chất lượng dịch vụ. Đối với Nhà hàng, hệ thống hỗ trợ quản lý thực đơn thông qua việc thêm, sửa, xoá món ăn và cập nhật giá hoặc hình ảnh; tiếp nhận và xử lý đơn hàng theo từng giai đoạn; cũng như xem báo cáo thống kê doanh thu theo ngày, tháng hoặc năm để phục vụ công tác quản lý kinh doanh. Đối với Admin, ứng dụng cho phép quản lý và phê duyệt tài khoản nhà hàng trước khi hoạt động trên hệ thống, theo dõi và xử lý các vấn đề liên quan đến người dùng hoặc món ăn, đồng thời truy cập vào hệ thống báo cáo tổng hợp để giám sát và đảm bảo tính ổn định của toàn bộ nền tảng.

Về mặt bảo mật: Hệ thống áp dụng cơ chế phân quyền chặt chẽ giữa ba vai trò chính gồm Admin, Nhà hàng và Khách hàng nhằm đảm bảo từng đối tượng chỉ có quyền truy cập vào những chức năng phù hợp với nhiệm vụ của mình. Việc xác thực tài khoản được thực hiện thông qua Firebase Authentication, giúp tăng cường mức độ an toàn và hạn chế tối đa tình trạng truy cập trái phép. Đồng thời, dữ liệu trong hệ thống được bảo vệ thông qua Firebase Security Rules, kiểm soát quyền đọc và ghi dựa trên vai trò và quyền hạn của từng loại người dùng. Bên cạnh đó, toàn bộ quá trình truyền tải dữ liệu đều được thực hiện qua kết nối bảo mật, giúp giảm thiểu nguy cơ rò rỉ thông tin hoặc bị can thiệp trái phép trong quá trình giao tiếp giữa người dùng và hệ thống.

1. **Đối tượng nghiên cứu**

Đối tượng nghiên cứu của hệ thống bao gồm ba nhóm người sử dụng chính, mỗi nhóm đảm nhận một vai trò và chức năng nhất định nhằm đảm bảo hoạt động liên tục và hiệu quả của ứng dụng.

Admin là thành phần quan trọng trong bộ máy vận hành hệ thống, chịu trách nhiệm quản lý toàn bộ tài khoản người dùng, bao gồm khách hàng và nhà hàng. Admin thực hiện việc phê duyệt các cửa hàng đăng ký mới để đảm bảo tính xác thực và chất lượng dịch vụ. Đồng thời, Admin còn giám sát các món ăn được đăng tải, xử lý các trường hợp vi phạm quy định hoặc nội dung không phù hợp. Bên cạnh đó, Admin có thể theo dõi báo cáo và thống kê tổng quan nhằm đánh giá mức độ hoạt động, phát hiện sai sót và hỗ trợ cải thiện trải nghiệm người dùng.

Khách hàng là đối tượng chính sử dụng ứng dụng để đặt món ăn trực tuyến. Người dùng có thể tạo tài khoản, đăng nhập vào hệ thống, xem danh sách các món ăn theo danh mục hoặc tìm kiếm nhanh theo từ khóa. Sau khi chọn món, khách hàng có thể thêm vào giỏ hàng, tiến hành đặt đơn và theo dõi trạng thái xử lý đơn hàng. Bên cạnh đó, khách hàng cũng có thể đánh giá món ăn sau khi thưởng thức nhằm giúp nhà hàng cải thiện chất lượng dịch vụ và mang lại trải nghiệm tốt hơn cho cộng đồng người dùng.

Nhà hàng là đơn vị cung cấp sản phẩm trong hệ thống. Mỗi nhà hàng có thể quản lý danh sách món ăn của mình bao gồm việc thêm mới, cập nhật thông tin, thay đổi giá hoặc hình ảnh minh họa. Khi khách hàng đặt đơn, nhà hàng sẽ tiếp nhận và thực hiện các thao tác xử lý như xác nhận, chuẩn bị và giao hàng. Đồng thời, nhà hàng có thể cập nhật trạng thái của đơn theo từng giai đoạn để khách hàng nắm bắt tiến độ. Ngoài ra, nhà hàng còn có quyền xem các thống kê doanh thu theo từng khoảng thời gian để phục vụ công tác đánh giá và phát triển kinh doanh.

Công nghệ nghiên cứu: Hệ thống được xây dựng dựa trên một số công nghệ hiện đại, phù hợp với yêu cầu phát triển ứng dụng di động và đảm bảo khả năng mở rộng trong tương lai.

Android Studio kết hợp với ngôn ngữ Java được lựa chọn làm nền tảng phát triển chính cho ứng dụng di động. Java mang lại sự ổn định, dễ bảo trì và phù hợp với các dự án cần cấu trúc rõ ràng. Android Studio cung cấp môi trường lập trình trực quan, hỗ trợ mạnh mẽ trong việc xây dựng giao diện, quản lý tài nguyên và kiểm thử ứng dụng.

Firebase Authentication được sử dụng để quản lý quá trình đăng ký và đăng nhập của người dùng, đảm bảo tính bảo mật và hỗ trợ nhiều phương thức xác thực như email và mật khẩu. Công nghệ này giúp giảm bớt khối lượng xử lý logic cho lập trình viên và đảm bảo mức độ an toàn cao cho tài khoản người dùng.

Firebase Storage đóng vai trò lưu trữ hình ảnh món ăn và các tệp đa phương tiện khác. Công nghệ này hỗ trợ khả năng lưu trữ dung lượng lớn, phân quyền truy cập linh hoạt và đảm bảo tốc độ tải xuống nhanh, giao diện món ăn hiển thị rõ ràng và hấp dẫn.

1. **Phạm vi nghiên cứu**

Phạm vi nghiên cứu của đề tài tập trung vào việc xây dựng và triển khai các chức năng cốt lõi của một ứng dụng đặt đồ ăn trực tuyến, đảm bảo hệ thống có thể hoạt động ổn định và đáp ứng đúng nhu cầu của các nhóm người dùng chính. Các nội dung được triển khai bao gồm:

- Quản lý tài khoản theo ba vai trò chính: Admin – Khách hàng – Nhà hàng, áp dụng cơ chế phân quyền rõ ràng nhằm đảm bảo mỗi nhóm chỉ có quyền truy cập vào các chức năng phù hợp. Mỗi loại tài khoản được thiết kế với giao diện và chức năng khác nhau, phù hợp với mục tiêu sử dụng thực tế.

- Chức năng đặt món và giỏ hàng, cho phép khách hàng duyệt danh sách món ăn, xem thông tin chi tiết, thêm món vào giỏ và tiến hành đặt hàng. Người dùng có thể theo dõi trạng thái đơn hàng thông qua các cập nhật được nhà hàng cung cấp.

- Quản lý món ăn và phê duyệt món ăn từ nhà hàng, trong đó nhà hàng có thể đăng món mới, chỉnh sửa thông tin, còn admin có quyền kiểm duyệt nội dung nhằm đảm bảo chất lượng và tính minh bạch.

- Thống kê và báo cáo doanh thu, cung cấp số liệu cơ bản giúp nhà hàng dễ dàng theo dõi tình hình kinh doanh và hỗ trợ admin trong việc giám sát toàn hệ thống.

Bên cạnh những nội dung triển khai, đề tài không thực hiện các chức năng ở phạm vi nâng cao như:

- Thanh toán online, bao gồm thanh toán qua ví điện tử hoặc thẻ ngân hàng.

- Tích hợp dịch vụ giao hàng qua bên thứ ba.

- Phát triển phiên bản ứng dụng dành cho nền tảng iOS, do hạn chế về thời gian và công cụ triển khai.

Phạm vi ứng dụng: Ứng dụng hướng đến việc triển khai thực tế trong quy mô nhỏ và vừa, phù hợp với:

- Các cửa hàng, quán ăn, nhà hàng nhỏ và vừa, có nhu cầu quản lý thực đơn và đơn hàng hằng ngày.

- Người dùng thông thường có nhu cầu đặt món ăn tiện lợi và nhanh chóng.

- Khu vực nội bộ như trường học, ký túc xá, khu dân cư, nơi nhu cầu đặt món tăng nhưng chưa có hệ thống chuyên nghiệp.

Hệ thống không nhằm mục tiêu phục vụ các đơn vị lớn, không áp dụng cho mô hình đặt món liên tỉnh hoặc hệ thống vận hành quy mô doanh nghiệp lớn với hàng nghìn đơn hàng mỗi ngày.

1. **Phương pháp nghiên cứu**

Phương pháp nghiên cứu tài liệu: Tìm hiểu các mô hình ứng dụng đặt đồ ăn phổ biến trên thị trường, phân tích kiến trúc hoạt động, giao diện người dùng và quy trình quản lý đơn hàng. Đồng thời, nghiên cứu các tài liệu kỹ thuật liên quan đến Firebase, Android Studio và mô hình phân quyền nhằm ứng dụng vào việc xây dựng hệ thống.

Phương pháp thực nghiệm: Tiến hành xây dựng ứng dụng theo từng module chức năng nhỏ, sau đó thực hiện kiểm thử độc lập nhằm đánh giá tính đúng đắn, ổn định và hiệu năng. Sau khi các module ổn định, tiến hành kiểm thử tích hợp theo quy trình phát triển phần mềm hướng Agile, đảm bảo hệ thống hoạt động mượt mà khi kết hợp toàn bộ chức năng. Cuối cùng là thử nghiệm chạy thực tế để đánh giá trải nghiệm người dùng.

1. **Ý nghĩa khoa học và thực tiễn của đồ án**

Ý nghĩa khoa học: Đề tài mang lại giá trị khoa học thông qua việc ứng dụng Firebase Authentication và Firebase Storage trong quá trình xây dựng ứng dụng di động, từ đó giúp sinh viên và người nghiên cứu hiểu rõ hơn cách triển khai cơ chế xác thực, phân quyền người dùng và quản lý tệp phương tiện trên nền tảng Android. Việc sử dụng Java và Firebase làm nền tảng phát triển cũng góp phần bổ sung nguồn tài liệu tham khảo cho các nghiên cứu liên quan đến bảo mật tài khoản, kiểm soát quyền truy cập và xây dựng giao diện mobile theo chuẩn Android.

Ý nghĩa thực tiễn: Về khía cạnh ứng dụng, hệ thống mang lại giải pháp đặt đồ ăn tiện lợi giúp người dùng tiết kiệm thời gian và thao tác. Nhà hàng có thêm công cụ hỗ trợ quản lý đơn hàng khoa học, hạn chế sai sót và nâng cao trải nghiệm cho khách hàng. Đồng thời, admin có thể dễ dàng giám sát hoạt động toàn hệ thống thông qua các báo cáo trực quan. Với cấu trúc gọn nhẹ và khả năng triển khai thực tế, ứng dụng phù hợp với các quán ăn nhỏ, cửa hàng trong khu dân cư hoặc các khu vực trường học – nơi nhu cầu đặt món online ngày càng tăng nhưng chưa có giải pháp tối ưu.

1. **Kết cấu của đồ án**

Đồ án trình bày gồm 3 chương:

**Chương 1. Cơ sở lý thuyết** trình bày tổng quan về công nghệ sử dụng.

**Chương 2. Phân tích và thiết kế hệ thống** đưa ra yêu cầu chức năng, mô hình Use case cho các vai trò, cơ sở dữ liệu.

**Chương 3. Xây dựng chương trình** mô tả triển khai giao diện.

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1. **Tổng quan về Java**
2. ***Khái niệm về Java***

Java là ngôn ngữ lập trình đa nền tảng (cross-platform), được phát triển bởi James Gosling tại Sun Microsystems (nay là Oracle Corporation). Ngôn ngữ lập trình này ra đời vào năm 1995 và được thiết kế để có thể chạy trên các nền tảng khác nhau, từ máy tính cá nhân đến thiết bị di động, các máy chủ và thiết bị nhúng.

Java sử dụng cấu trúc lập trình hướng đối tượng (object-oriented programming - OOP) và được xây dựng trên cơ sở của ngôn ngữ lập trình C++. Nó cung cấp một môi trường chạy ảo (virtual machine) gọi là Java Virtual Machine (JVM), giúp các chương trình Java có thể chạy trên nhiều nền tảng khác nhau mà không cần phải biên dịch lại.



1. Khái niệm về Java

Hiện tại, Java là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay, được sử dụng rộng rãi trong các ứng dụng di động, phát triển web, lập trình trên các thiết bị nhúng, máy tính cá nhân, máy chủ, game và nhiều lĩnh vực khác.

1. ***Các tính năng đặc trưng của Java***

**-** Đa nền tảng: Java được thiết kế để có thể chạy trên nhiều nền tảng khác nhau, vì vậy nó rất phù hợp cho việc phát triển các ứng dụng đa nền tảng. Java sử dụng một máy ảo (JVM - Java Virtual Machine) để chạy mã nguồn, vì vậy mã nguồn được viết một lần và có thể chạy trên nhiều hệ điều hành khác nhau mà không cần thay đổi.

**-** Quản lý bộ nhớ tự động: Java có tính năng tự động quản lý bộ nhớ, tức là nó tự động thu dọn các vùng nhớ không sử dụng nữa để giảm thiểu các lỗi bộ nhớ. Điều này giúp cho các ứng dụng được viết bằng Java có thể chạy ổn định và tránh các lỗi liên quan đến bộ nhớ.

**-** Hỗ trợ đa luồng: Java có thể xử lý đa luồng, cho phép chương trình thực hiện nhiều tác vụ cùng một lúc. Điều này giúp cho các ứng dụng có thể chạy nhanh và hiệu quả hơn, đặc biệt là khi phải xử lý nhiều tác vụ cùng một lúc.

1. Các tính năng đặng trưng của Java

**-** Tính bảo mật cao: Java có các tính năng bảo mật như kiểm tra kiểu tĩnh và kiểm tra lỗi trên đường dẫn. Java được thiết kế để giảm thiểu các lỗ hổng bảo mật và các vấn đề liên quan đến an ninh.

**-** Hỗ trợ các thư viện và công cụ phát triển mạnh mẽ: Java có các thư viện tiêu chuẩn và các công cụ phát triển như IDE (Integrated Development Environment) để giúp các nhà phát triển tạo ra các ứng dụng dễ dàng hơn. Các thư viện và công cụ này cung cấp các tính năng như: Tạo giao diện người dùng, xử lý dữ liệu, kết nối cơ sở dữ liệu và nhiều hơn nữa,...

**-** Tính di động: Java được sử dụng rộng rãi trong lĩnh vực di động bao gồm: các thư viện hỗ trợ việc phát triển ứng dụng di động, đóng gói ứng dụng thành các file .jar hoặc .apk, cung cấp các tính năng như xử lý đa nhiệm, kết nối mạng và tích hợp với các thiết bị phần cứng như máy ảnh.

**-** Tính độc lập với nền tảng :Java có thể chạy trên nhiều nền tảng khác nhau và không phụ thuộc vào bất kỳ nền tảng cụ thể nào. Điều này giúp cho các ứng dụng Java có thể được triển khai trên nhiều hệ thống khác nhau mà không cần sửa đổi mã nguồn.

**-** Tính kế thừa và đa hình: Java là ngôn ngữ lập trình hướng đối tượng (OOP), vì vậy nó có các tính năng như kế thừa, đa hình và đóng gói. Các tính năng này giúp cho mã nguồn được tái sử dụng và giảm thiểu sự trùng lặp trong mã nguồn.

**-** Tính mở rộng: Java có tính năng mở rộng, cho phép các nhà phát triển thêm các tính năng mới vào ngôn ngữ bằng cách tạo các thư viện và API riêng. Điều này giúp cho Java có thể được sử dụng trong nhiều lĩnh vực khác nhau và được phát triển theo các hướng khác nhau.

1. ***Ưu và nhược điểm của Java***

- Ưu điểm của Java:

+ Độ tin cậy cao

+ Tính đa nền tảng

+ Quản lý bộ nhớ tự động

+ Công cụ phát triển phong phú

+ Hỗ trợ đa luồng

1. Ưu và nhược điểm của Java

**-** Nhược điểm của Java:

+ Tốc độ chậm hơn so với các ngôn ngữ lập trình gần sát với phần cứng, chẳng hạn như C hoặc C++.

+ Java có thể chạy trên nhiều nền tảng khác nhau, nhưng ứng dụng này có thể cần đến một trình biên dịch hoặc máy ảo Java riêng biệt để có thể chạy trên các thiết bị di động.

+ Sử dụng bộ nhớ lớn hơn so với một số ngôn ngữ lập trình khác.

+ Cú pháp phức tạp hơn so với một số ngôn ngữ lập trình khác.

1. **Tổng quan về Android Studio**
2. ***Khái niệm về Android Studio***

Android Studio là một môi trường tích hợp phát triển (Integrated Development Environment – IDE) được phát triển bởi Google dành cho việc phát triển ứng dụng trên nền tảng Android. Nó cung cấp một loạt các công cụ và tính năng để giúp nhà phát triển xây dựng và triển khai ứng dụng Android một cách dễ dàng.



1. Khái niệm Android Studio

Công cụ này cung cấp một giao diện trực quan và các trình biên dịch, trình gỡ lỗi, trình tạo giao diện người dùng, trình quản lý phiên bản và nhiều công cụ khác. Nó hỗ trợ viết code trong ngôn ngữ Java hoặc Kotlin và tích hợp sẵn các thư viện và công cụ phát triển Android.

Một trong những tính năng quan trọng của công cụ này là khả năng sử dụng Gradle, một hệ thống quản lý dự án mạnh mẽ. Gradle cho phép bạn quản lý phụ thuộc, xây dựng, kiểm thử và đóng gói ứng dụng Android một cách linh hoạt và hiệu quả.

Lịch sử phát triển Android Studio: Android Studio đã được công bố lần đầu tiên vào năm 2013 tại hội nghị Google I/O và chính thức phát hành rộng rãi vào năm 2014. Trước đó, các nhà phát triển ứng dụng Android thường sử dụng các công cụ như Eclipse IDE hoặc IDE của Java để phát triển ứng dụng.



1. Lịch sử phát triển Android Studio

Sau khi được công bố công cụ này đã trở thành một công cụ phát triển ứng dụng Android chính thức và được Google khuyến nghị sử dụng. Nó đã thay thế Eclipse IDE và trở thành môi trường phát triển chính cho các dự án Android.

Đối với những người có kinh nghiệm trong phát triển phần mềm, việc tạo ứng dụng trở nên dễ dàng hơn. Công cụ này cung cấp nhiều tính năng và công cụ hỗ trợ như trình biên dịch thông minh, gỡ lỗi tiện lợi, thiết kế giao diện trực quan và nhiều tính năng khác giúp tăng năng suất và hiệu quả trong quá trình phát triển ứng dụng Android.

1. ***Thành phần trong Android Studio***

Activity là thành phần chính của một ứng dụng Android. Nó đại diện cho một màn hình giao diện người dùng, mà người dùng có thể tương tác để thực hiện các hành động nhất định. Activity được viết bằng Java và là một lớp Java.

Service là một thành phần quan trọng trong Android, được sử dụng để xử lý các tác vụ chạy trong thời gian dài mà không cần sự tương tác trực tiếp từ người dùng. Nhờ có Service, ứng dụng có thể tiếp tục thực hiện công việc ở chế độ nền ngay cả khi người dùng mở ứng dụng khác hoặc thoát khỏi giao diện chính. Các Service thường được dùng cho những tác vụ như phát nhạc, đồng bộ dữ liệu, cập nhật thông báo hoặc theo dõi vị trí. Thành phần này được viết bằng Java và được triển khai dưới dạng một lớp Java kế thừa từ lớp Service do Android cung cấp, giúp đảm bảo quá trình xử lý diễn ra ổn định và độc lập với giao diện người dùng.

Broadcast Receiver là một thành phần được sử dụng để nhận các thông báo từ hệ thống hoặc ứng dụng khác. Nó được sử dụng để xử lý các sự kiện hệ thống như kết nối mạng thay đổi, pin yếu và các sự kiện khác. Broadcast Receiver được viết bằng Java và là một lớp Java.

Content Provider là một thành phần được sử dụng để quản lý và chia sẻ dữ liệu giữa các ứng dụng khác nhau. Nó cung cấp cơ chế để truy cập và thao tác với dữ liệu được lưu trữ trong cơ sở dữ liệu của ứng dụng. Content Provider được viết bằng Java và là một lớp Java.

Fragment là một thành phần được sử dụng để tạo ra một phần của giao diện người dùng trong một Activity. Nó cho phép bạn tạo ra giao diện người dùng động và linh hoạt hơn. Fragment được viết bằng Java và là một lớp Java.

Layout là một thành phần được sử dụng để định dạng giao diện người dùng. Nó đại diện cho cách các thành phần khác nhau như TextView, EditText, Button... được sắp xếp trên màn hình. Layout được viết bằng XML, tuy nhiên, Java cũng được sử dụng để thao tác với các thành phần của Layout.

View là một thành phần cơ bản của giao diện người dùng. Nó đại diện cho các thành phần như TextView, EditText, Button, CheckBox... được hiển thị trên màn hình. ViewGroup là một lớp cha của các thành phần View và được sử dụng để sắp xếp các thành phần View trên màn hình. Cả View và ViewGroup được viết bằng Java và là các lớp Java.

Style và Theme là hai thành phần được sử dụng để định dạng giao diện người dùng. Style được sử dụng để định dạng một thành phần cụ thể, trong khi Theme được sử dụng để định dạng toàn bộ ứng dụng. Cả Style và Theme được viết bằng XML.

Intent là một thành phần được sử dụng để kết nối các thành phần khác nhau của ứng dụng Android. Nó được sử dụng để chuyển dữ liệu và thực hiện các hành động nhất định. Intent được viết bằng Java và là một lớp Java.

AsyncTask và Thread là hai thành phần được sử dụng để thực hiện các tác vụ nền trong ứng dụng Android. AsyncTask được sử dụng để thực hiện các tác vụ nền nhỏ nhẹ, trong khi Thread được sử dụng để thực hiện các tác vụ nền nặng hơn và cần nhiều thời gian hơn. Cả AsyncTask và Thread đều được viết bằng Java và là các lớp Java.

1. ***Ưu và nhược điểm của Android Studio***

- Ưu điểm của Android Studio:

+ Đây là môi trường phát triển phần mềm chính thức của Google, đây cũng chính là chủ sở hữu của hệ điều hành Android.

+ Công cụ này cung cấp một loạt các công cụ phát triển và tính năng hỗ trợ cho quá trình phát triển ứng dụng Android. Điều này bao gồm trình biên dịch, trình gỡ lỗi, trình quản lý dự án, thiết kế giao diện và nhiều tính năng khác để tăng năng suất và hiệu quả của nhà phát triển.

+ Giao diện người dùng thân thiện và dễ sử dụng, giúp tăng hiệu suất và giảm thời gian cần thiết để phát triển ứng dụng.

+ Hỗ trợ bởi một số tài liệu tham khảo và hướng dẫn phong phú. Có sẵn các tài liệu chính thức từ Google, cùng với các diễn đàn lập trình viên Android và các nguồn tài liệu trực tuyến khác để giúp nhà phát triển tìm hiểu và giải quyết các vấn đề phát triển ứng dụng.

+ Công cụ này được hỗ trợ thông qua các khóa học đào tạo về lập trình Android cơ bản và nâng cao. Các khóa học này giúp nhà phát triển nắm vững các khái niệm và kỹ năng cần thiết để phát triển ứng dụng Android chất lượng.

- Nhược điểm của Android Studio:

+ Yêu cầu tài nguyên hệ thống cao: Android Studio là một phần mềm khá nặng và yêu cầu một hệ thống mạnh để chạy mượt mà.

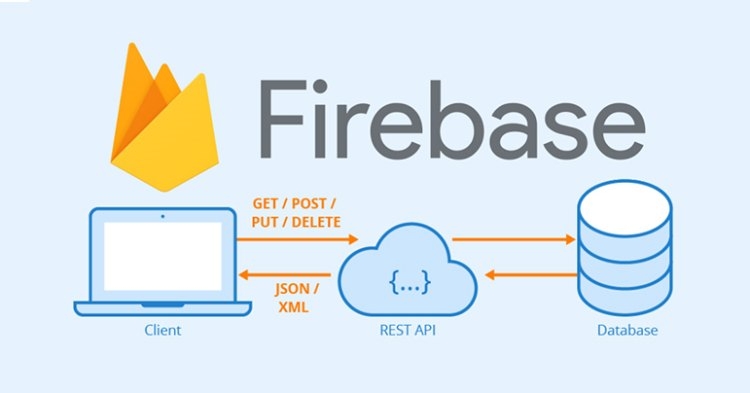
+ Học tập khá phức tạp: Android Studio đòi hỏi người sử dụng có kiến thức về lập trình, điều này có thể gây khó khăn cho những người mới bắt đầu phát triển ứng dụng.

+ Đôi khi chậm và không ổn định: Mặc dù Android Studio đã được cải tiến rất nhiều, nhưng vẫn có thể gặp phải các vấn đề về tốc độ và ổn định.

+ Không hỗ trợ tốt cho các phiên bản Android cũ: Android Studio hỗ trợ tốt nhất cho các phiên bản Android mới nhất, điều này có nghĩa là nếu bạn muốn phát triển ứng dụng cho các phiên bản cũ hơn của Android, bạn sẽ phải tìm kiếm giải pháp khác.

1. **Tổng quan về Firebase**
2. ***Khái niệm về Firebase***

Firebase là một nền tảng phát triển ứng dụng (Backend-as-a-Service – BaaS) do Google cung cấp, hỗ trợ mạnh mẽ cho việc xây dựng các ứng dụng web và ứng dụng di động hiện đại. Firebase cung cấp nhiều dịch vụ tích hợp như cơ sở dữ liệu thời gian thực, xác thực người dùng, lưu trữ tệp, phân tích và thông báo đẩy, giúp quá trình phát triển trở nên nhanh chóng, linh hoạt và dễ dàng hơn.



1. Khái niệm về Firebase

Firebase cho phép lưu trữ và quản lý dữ liệu thông qua hai hệ quản trị cơ sở dữ liệu chính: Realtime Database và Cloud Firestore. Cả hai đều hỗ trợ mô hình phi quan hệ (NoSQL), trong đó dữ liệu được lưu dưới dạng JSON hoặc dạng tài liệu – bộ sưu tập (document – collection). Nhờ mô hình này, Firebase có khả năng đồng bộ hóa dữ liệu theo thời gian thực giữa nhiều thiết bị, giúp trải nghiệm người dùng luôn được cập nhật nhanh chóng.

Firebase đã trở nên phổ biến nhờ khả năng tích hợp dễ dàng với các ứng dụng Android, iOS và web, đồng thời cung cấp nhiều API mạnh mẽ hỗ trợ truy xuất, cập nhật và đồng bộ dữ liệu. Bên cạnh đó, cộng đồng lập trình viên Firebase rất lớn, đóng góp vào việc chia sẻ kiến thức và mở rộng hệ sinh thái cho nền tảng này.

Hiện nay, Firebase liên tục được Google cải tiến với nhiều tính năng mới nhằm hỗ trợ lập trình viên xây dựng ứng dụng quy mô lớn. Firebase có phiên bản miễn phí (Spark Plan) và có phí (Blaze Plan). Phiên bản miễn phí phù hợp cho sinh viên, dự án cá nhân hoặc ứng dụng nhỏ, trong khi phiên bản trả phí cung cấp khả năng mở rộng mạnh mẽ hơn dành cho doanh nghiệp.

1. ***Cơ chế hoạt động của Firebase***

Cơ chế hoạt động của Firebase được xây dựng theo kiến trúc không máy chủ (serverless), nghĩa là lập trình viên không cần quản lý máy chủ mà chỉ tập trung vào việc phát triển ứng dụng. Firebase vận hành như sau:

Quản lý và lưu trữ dữ liệu: Firebase cho phép tạo cơ sở dữ liệu NoSQL theo dạng cây JSON (Realtime Database) hoặc dạng tài liệu – bộ sưu tập (Cloud Firestore). Mỗi tài liệu có thể chứa nhiều trường và dữ liệu có thể được tổ chức linh hoạt tùy theo nhu cầu ứng dụng. Hệ thống hỗ trợ quy tắc bảo mật (Security Rules) để kiểm soát quyền truy cập của người dùng.

Gửi yêu cầu đọc/ghi dữ liệu: Người dùng hoặc ứng dụng (Client) có thể gửi các yêu cầu đọc, ghi, cập nhật hoặc xóa dữ liệu thông qua API do Firebase cung cấp. Các thao tác này không sử dụng SQL mà thao tác trực tiếp trên tài liệu hoặc nút dữ liệu.

Xử lý yêu cầu trên máy chủ: Firebase Server tiếp nhận yêu cầu từ Client, xác thực người dùng, kiểm tra quyền truy cập và thực hiện hành động tương ứng trên cơ sở dữ liệu. Đối với Cloud Firestore và Realtime Database, Firebase tự động tối ưu tốc độ truy vấn và đảm bảo dữ liệu được đồng bộ theo thời gian thực.

Đồng bộ và trả kết quả: Sau khi hoàn tất xử lý, Firebase sẽ trả về kết quả cho Client. Với Realtime Database và Firestore chế độ realtime, bất kỳ thay đổi nào sẽ được tự động cập nhật đến tất cả thiết bị đang truy cập dữ liệu, không cần làm mới thủ công.

1. ***Ưu và nhược điểm của Firebase***

- Ưu điểm của Firebase:

+ Đồng bộ thời gian thực: Firebase hỗ trợ cập nhật dữ liệu tức thì trên nhiều thiết bị, rất phù hợp với các ứng dụng chat, đặt đồ ăn online, theo dõi đơn hàng,...

+ Không cần quản lý máy chủ: Firebase hoạt động theo mô hình serverless, giúp giảm chi phí và công sức vận hành hệ thống.

+ Khả năng mở rộng linh hoạt: Firebase tự động mở rộng theo nhu cầu, phù hợp với cả ứng dụng nhỏ và hệ thống lớn.

+ Tích hợp nhiều dịch vụ: Firebase đi kèm Authentication, Storage, Cloud Messaging, Analytics,… giúp ứng dụng phát triển nhanh hơn.

+ Bảo mật mạnh mẽ: Cung cấp cơ chế Firebase Security Rules giúp kiểm soát quyền truy cập theo người dùng hoặc theo dữ liệu.

+ Hỗ trợ đa nền tảng: Dễ dàng sử dụng cho Android, iOS, web và game.

- Nhược điểm của Firebase:

+ Không hỗ trợ SQL: Vì là NoSQL nên không có truy vấn SQL phức tạp. Dữ liệu phải thiết kế theo dạng JSON hoặc document, đôi khi gây khó khăn với người mới.

+ Chi phí tăng theo dung lượng sử dụng: Mặc dù miễn phí ở mức cơ bản, nhưng khi ứng dụng lớn, chi phí Blaze có thể tăng cao.

+ Giới hạn trong xử lý dữ liệu phức tạp: Các truy vấn phức tạp hoặc phân tích dữ liệu nâng cao không mạnh như các hệ quản trị CSDL quan hệ.

+ Khó di chuyển sang nền tảng khác: Vì Firebase sử dụng cấu trúc NoSQL đặc thù, việc chuyển đổi sang hệ thống RDBMS có thể tốn nhiều thời gian.

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1. **Khảo sát**

Ứng dụng đặt đồ ăn trực tuyến là một nền tảng hỗ trợ sinh viên và người dùng đặt món ăn nhanh chóng, tiện lợi từ các cửa hàng, quán ăn trong khu vực. Hệ thống được xây dựng nhằm số hóa toàn bộ quy trình đặt, chuẩn bị, giao, thanh toán món ăn, giúp giảm thời gian chờ đợi, hạn chế sai sót và tăng trải nghiệm của người dùng. Bên cạnh đó, ứng dụng còn hỗ trợ các cửa hàng tối ưu quy trình kinh doanh hằng ngày, quản lý thực đơn, theo dõi đơn hàng và thống kê doanh thu. Với việc sử dụng Java trong phát triển ứng dụng Android, cùng Firebase Authentication để xác thực tài khoản và Firestore Database để lưu trữ dữ liệu theo thời gian thực, hệ thống đảm bảo tốc độ xử lý nhanh, độ tin cậy cao và khả năng mở rộng linh hoạt.

Hệ thống bao gồm ba phân quyền chính: Admin, Khách hàng và Nhà hàng, mỗi phân quyền có mục đích và chức năng riêng.

- Các thành phần chính của hệ thống bao gồm:

+ Quản lý cửa hàng, quán ăn và món ăn:

* Mỗi cửa hàng được lưu trữ đầy đủ các thông tin như: T ên cửa hàng, chủ cửa hàng, số điện thoại, email, địa chỉ, ảnh đại diện,… phục vụ cho việc hiển thị và quản lý từ phía Admin và khách hàng.
* Các cửa hàng có thể cung cấp đa dạng loại món ăn như: Món chính, đồ uống, món ăn vặt,…giúp người dùng dễ dàng lựa chọn theo nhu cầu.
* Mỗi loại món ăn chứa nhiều món khác nhau, mỗi món đều được lưu trữ với các thuộc tính như mô tả chi tiết, hình ảnh món, giá tiền, loại món ăn và đánh giá của người mua khác.
* Giá món ăn và thông tin mô tả có thể được cửa hàng cập nhật thường xuyên, hỗ trợ các chương trình khuyến mãi, điều chỉnh giá theo thời điểm.
* Các dữ liệu này được lưu trữ và đồng bộ thời gian thực trên Firestore Database, giúp người dùng luôn xem được thông tin mới nhất.

+ Quản lý tài khoản người dùng (người mua và cửa hàng)

* Hệ thống sử dụng Firebase Authentication để quản lý quá trình đăng ký, đăng nhập bảo mật, hỗ trợ đăng nhập bằng email và mật khẩu.
* Mỗi tài khoản người dùng được lưu trữ các thông tin riêng như: Họ tên, số điện thoại, email, mật khẩu, địa chỉ và phân quyền.
* Nhà hàng quán ăn đăng nhập với phân quyền riêng để quản lý thông tin cửa hàng, cập nhật menu, xem đơn đặt hàng, quản lý voucher, xem thống kê doanh thu và đơn hàng.
* Khách hàng đăng nhập để xem danh sách cửa hàng, xem món ăn, tìm kiếm món ăn, đặt món, theo dõi đơn hàng, thanh toán đơn hàng, thêm món ăn vào món ăn yêu thích, đánh giá nhà hàng và món ăn đã mua, xem thống kê chi tiêu.

+ Quản lý đơn đặt hàng:

* Khi khách hàng đặt món, hệ thống tự động tạo đơn hàng bao gồm các thông tin: Cửa hàng nhận đơn, danh sách món ăn, giá tiền từng món, số lượng, ghi chú của người dùng, tổng tiền, địa chỉ giao hàng và phương thức thanh toán.
* Người dùng được phép thay đổi món nếu đơn chưa được cửa hàng xác nhận chế biến.
* Sau khi cửa hàng xác nhận, món đang chế biến sẽ không được hủy hoặc chỉnh sửa.
* Toàn bộ thông tin đơn hàng được lưu trữ trên Firestore giúp khách hàng và cửa hàng theo dõi theo thời gian thực.

+ Quy trình xử lý từ phía cửa hàng:

* Khi người dùng đặt món, đơn hàng sẽ xuất hiện trên giao diện quản lý của cửa hàng.
* Cửa hàng kiểm tra đơn và xác nhận đơn.
* Trạng thái đơn hàng được cập nhật liên tục: Chờ xác nhận – Đang giao – Đã hoàn thành.
* Khi hoàn tất đơn, hệ thống lưu lại lịch sử giao dịch để hỗ trợ thống kê cho cửa hàng.

+ Quản lý hệ thống từ phía Admin:

* Admin có khả năng quản lý toàn bộ hệ thống: Quản lý tài khoản, quản lý cửa hàng và quản lý món.
* Admin có thể duyệt hoặc khóa tài khoản cửa hàng nếu vi phạm quy định.
* Admin xem báo cáo tổng quan: Tổng số lượng đơn hàng, doanh thu tổng hợp của hệ thống, tổng số lượng cửa hàng, tổng số lượng tài khoản.

+ Thanh toán và hóa đơn:

* Người dùng có thể thanh toán online hoặc thanh toán khi nhận hàng.
* Hệ thống tự động tính tổng tiền và tạo hóa đơn chi tiết lưu vào Firestore để khách hàng xem lại khi cần.

Tóm lại: Ứng dụng đặt đồ ăn trực tuyến không chỉ hỗ trợ sinh viên và người dùng đặt món ăn thuận tiện, nhanh chóng mà còn cung cấp công cụ quản lý hiệu quả cho các cửa hàng trong việc điều phối đơn hàng, quản lý thực đơn và tối ưu hoạt động kinh doanh. Với sự hỗ trợ của Java, Firebase Authentication và Firestore Database, hệ thống đảm bảo tốc độ xử lý nhanh, tính ổn định cao, dễ mở rộng và mang lại trải nghiệm tốt hơn cho người dùng. Việc số hóa quy trình đặt món giúp giảm tải cho cửa hàng, hạn chế sai sót và nâng cao chất lượng dịch vụ trong toàn bộ quá trình đặt, thanh toán và giao nhận món ăn.

1. **Phân tích hệ thống**
2. ***Yêu cầu chức năng***

- Chức năng của người dùng:

+ Đăng ký và đăng nhập vào ứng dụng.

+ Tìm kiếm và xem danh sách các cửa hàng, món ăn.

+ Xem chi tiết món ăn, giá, đánh giá từ người mua khác.

+ Đặt món trực tuyến, thêm món ăn vào mục yêu thích.

+ Điều chỉnh hoặc hủy đơn hàng nếu cửa hàng chưa xác nhận đơn hàng.

+ Theo dõi trạng thái đơn hàng theo từng giai đoạn: Chờ xác nhận, đang giao, đã hoàn thành.

+ Lưu địa chỉ giao hàng.

+ Thanh toán online hoặc thanh toán khi nhận hàng.

+ Xem lịch sử đơn hàng, đánh giá cửa hàng và món ăn.

- Chức năng của cửa hàng/quán ăn:

+ Đăng nhập vào hệ thống quản lý cửa hàng.

+ Quản lý cửa hàng.

+ Quản lý danh mục món ăn, bao gồm thêm, sửa, xóa món.

+ Cập nhật giá món ăn.

+ Quản lý đơn hàng từ người dùng: Xác nhận đơn, từ chối đơn.

+ Xem thống kê doanh thu và lượt đặt món.

- Chức năng của quản trị viên:

+ Quản lý tài khoản cửa hàng và tài khoản người dùng.

+ Duyệt nhà hàng và món ăn.

+ Quản lý thống kê tổng hợp.

1. ***Yêu cầu phi chức năng***

Ngoài các yêu cầu chức năng, hệ thống đặt đồ ăn trực tuyến cũng cần đáp ứng các yêu cầu phi chức năng nhằm đảm bảo hiệu quả, ổn định và an toàn trong quá trình vận hành. Đây là các yếu tố quyết định giúp hệ thống hoạt động lâu dài, xử lý lượng người dùng lớn và mang lại trải nghiệm mượt mà cho khách hàng, nhà hàng và admin.

Về cơ chế bảo mật của hệ thống: Ứng dụng sử dụng Firebase Authentication để xác thực tài khoản, đảm bảo chỉ những người dùng hợp lệ mới có thể truy cập và sử dụng các chức năng. Khách hàng và nhà hàng đều phải đăng ký và xác thực email trước khi thao tác như đặt món, quản lý thực đơn hay xử lý đơn hàng. Đối với tài khoản nhà hàng, hệ thống yêu cầu admin kiểm duyệt trước khi được phép hoạt động, giúp nâng cao tính minh bạch và độ tin cậy. Song song đó, dữ liệu người dùng được bảo vệ thông qua cơ chế mã hóa của Firebase, trong đó mật khẩu được xử lý và lưu trữ an toàn. Firestore Database áp dụng Security Rules để phân quyền chi tiết theo từng vai trò (khách hàng, nhà hàng, admin), đảm bảo mỗi đối tượng chỉ có thể truy cập đúng phạm vi dữ liệu của mình. Các thông tin nhạy cảm như lịch sử đặt hàng, thông tin cửa hàng, thực đơn hay dữ liệu cá nhân đều được lưu trữ trong môi trường bảo mật và kiểm soát chặt chẽ, ngăn ngừa truy cập hoặc chỉnh sửa trái phép.

Về hiệu suất và khả năng mở rộng: Hệ thống đảm bảo tốc độ xử lý nhanh và ổn định trong các thao tác như tải danh sách món ăn, tìm kiếm, đặt hàng và cập nhật trạng thái đơn hàng. Firestore Database hỗ trợ đồng bộ dữ liệu tức thời, giúp trạng thái đơn hàng giữa khách hàng, nhà hàng và admin luôn được cập nhật kịp thời. Khi số lượng người dùng, cửa hàng hoặc đơn hàng tăng lên, Firestore có khả năng tự động mở rộng quy mô xử lý và lưu trữ mà không cần can thiệp từ người phát triển. Trong tương lai, ứng dụng có thể bổ sung các tính năng nâng cao như gợi ý món ăn, chatbot hỗ trợ hoặc phân tích dữ liệu nhằm cải thiện trải nghiệm người dùng.

Về khả năng sử dụng và giao diện người dùng: Ứng dụng được thiết kế với giao diện trực quan và dễ sử dụng, phù hợp cho cả khách hàng, chủ cửa hàng và admin. Các chức năng như xem danh sách món, thêm vào giỏ hàng, quản lý thực đơn, xử lý đơn hàng hay xem báo cáo đều được bố trí hợp lý, giúp người dùng thao tác thuận tiện và nhanh chóng. Hệ thống hỗ trợ thông báo tức thời như đơn hàng mới, trạng thái đơn hàng thay đổi, đơn hàng đã hoàn tất hoặc cảnh báo lỗi truy cập, giúp người dùng luôn nắm được thông tin quan trọng. Ngoài ra, các bước thao tác đều có hướng dẫn rõ ràng, hỗ trợ người dùng tiếp cận ứng dụng dễ dàng dù không có nhiều kinh nghiệm công nghệ.

1. **Mô tả chi tiết Use case**

- Mô tả tác nhân:

1. Bảng mô tả tác nhân

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên tác nhân** | **Mô tả** |
| 1 | Admin | Là quản trị viên của hệ thống, có mọi quyền và thực hiện mọi chức năng |
| 2 | Khách hàng | Là khách hàng sử dụng |
| 3 | Nhà hàng | Nhà hàng quản lý món ăn của mình |

1. Bảng mô tả Use case

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use case** | **Mô tả** |
| 1 | Quản lý tài khoản | Đăng ký, đăng nhập/xuất, quản lý hồ sơ & mật khẩu. |
| 2 | Quản lý món ăn | Nhà hàng quản lý món ăn và chờ admin duyệt |
| 3 | Giỏ hàng | Khách hàng lựa chọn món ăn và thêm vào giỏ hàng |
| 4 | Tra cứu đơn hàng | Khách hàng tra cứu đơn hàng khi mua |
| 5 | Thống kê & báo cáo | Thống kê doanh thu (tổng hợp), thống kê chuyên sâu bằng biểu đồ |
| 6 | Duyệt sản phẩm | Admin duyệt sản phẩm khi nhà hàng đăng món |
| 7 | Yêu thích | Yêu thích món ăn |

- Mô tả Use case đăng ký tài khoản:

+ Sự kiện chính:

* Người dùng (Khách hàng hoặc Nhà hàng) truy cập vào màn hình đăng ký tài khoản trên ứng dụng.
* Hệ thống hiển thị biểu mẫu đăng ký yêu cầu nhập thông tin như tên, số điện thoại, email, mật khẩu và chọn loại tài khoản (Khách hàng hoặc Nhà hàng).
* Người dùng điền đầy đủ thông tin và nhấn nút “Đăng ký”.
* Hệ thống gửi mã xác nhận tới email người dùng thông qua Firebase Authentication.
* Người dùng nhập mã xác nhận email trên ứng dụng.
* Hệ thống kiểm tra mã xác nhận hợp lệ. Nếu hợp lệ, tài khoản được tạo và lưu vào Firestore Database, phân quyền theo loại tài khoản.
* Đối với tài khoản Nhà hàng, trạng thái tài khoản được đánh dấu “chờ duyệt” và yêu cầu admin kiểm duyệt trước khi có thể hoạt động.
* Người dùng được điều hướng đến màn hình đăng nhập hoặc trang chủ sau khi hoàn tất đăng ký và email đã được xác thực.

+ Kết thúc:

* Nếu có lỗi trong quá trình đăng ký (mã xác nhận không hợp lệ, thông tin không hợp lệ hoặc lỗi kết nối Firebase), hệ thống hiển thị thông báo lỗi và người dùng có thể thử lại.

- Mô tả Use case đăng nhập tài khoản:

+ Sự kiện chính:

* Người dùng (Khách hàng hoặc Nhà hàng) truy cập màn hình đăng nhập trên ứng dụng.
* Hệ thống hiển thị biểu mẫu đăng nhập yêu cầu nhập email và mật khẩu.
* Người dùng nhập đầy đủ thông tin và nhấn nút “Đăng nhập”.
* Hệ thống kiểm tra thông tin đăng nhập thông qua Firebase Authentication, nếu thông tin hợp lệ, người dùng được xác thực, phân quyền theo loại tài khoản (Khách hàng, Nhà hàng, Admin) và chuyển hướng đến trang chủ hoặc trang quản lý tương ứng.

+ Kết thúc:

* Nếu thông tin đăng nhập không hợp lệ hoặc xảy ra lỗi kết nối với Firebase, hệ thống sẽ hiển thị thông báo lỗi tương ứng, đồng thời yêu cầu người dùng kiểm tra lại thông tin và thực hiện đăng nhập lại.

- Mô tả Use case đặt món ăn:

+ Sự kiện chính:

* Khách hàng truy cập vào danh sách các món ăn được hiển thị trên ứng dụng, xem thông tin chi tiết như giá, hình ảnh, mô tả món và sau đó lựa chọn món muốn đặt theo nhu cầu của mình.
* Hệ thống hiển thị thông tin chi tiết của món, bao gồm giá, mô tả, hình ảnh và các đánh giá của khách hàng khác.
* Khách hàng chọn số lượng và các tùy chọn đi kèm, hệ thống tự động tính toán tổng tiền và cập nhật thông tin giỏ hàng.
* Khi khách hàng xác nhận đặt món, hệ thống lưu thông tin đơn hàng vào Firestore Database, đồng thời cập nhật trạng thái đơn hàng cho cả khách hàng và nhà hàng liên quan.

+ Kết thúc:

* Nếu quá trình đặt món gặp lỗi kết nối hoặc dữ liệu không hợp lệ, hệ thống hiển thị thông báo lỗi và khách hàng có thể thử lại thao tác đặt món.

- Mô tả Use case đơn hàng:

+ Sự kiện chính:

* Khách hàng sau khi đặt món thành công có thể theo dõi toàn bộ tiến trình xử lý đơn hàng ngay trên ứng dụng. Hệ thống sẽ cập nhật liên tục các trạng thái như đang xử lý, đang giao và hoàn thành.
* Nhà hàng nhận được thông báo về đơn hàng mới, kiểm tra và cập nhật trạng thái đơn hàng trong Firestore Database.

+ Kết thúc:

* Nếu quá trình theo dõi hoặc cập nhật trạng thái đơn hàng gặp lỗi kết nối hoặc dữ liệu không hợp lệ, hệ thống hiển thị thông báo lỗi và người dùng có thể thử lại thao tác.

- Mô tả Use case hủy đơn hàng:

+ Sự kiện chính:

* Khách hàng đăng nhập vào ứng dụng và truy cập chi tiết đơn hàng muốn hủy.
* Khách hàng kích nút “Hủy món” trên màn hình chi tiết đơn hàng.
* Hệ thống cập nhật trạng thái đơn hàng trong Firestore Database và hiển thị lại chi tiết đơn hàng với trạng thái đã hủy.

+ Kết thúc:

* Nếu có lỗi trong quá trình cập nhật dữ liệu, hệ thống hiển thị thông báo lỗi và khách hàng có thể thử lại thao tác hủy.

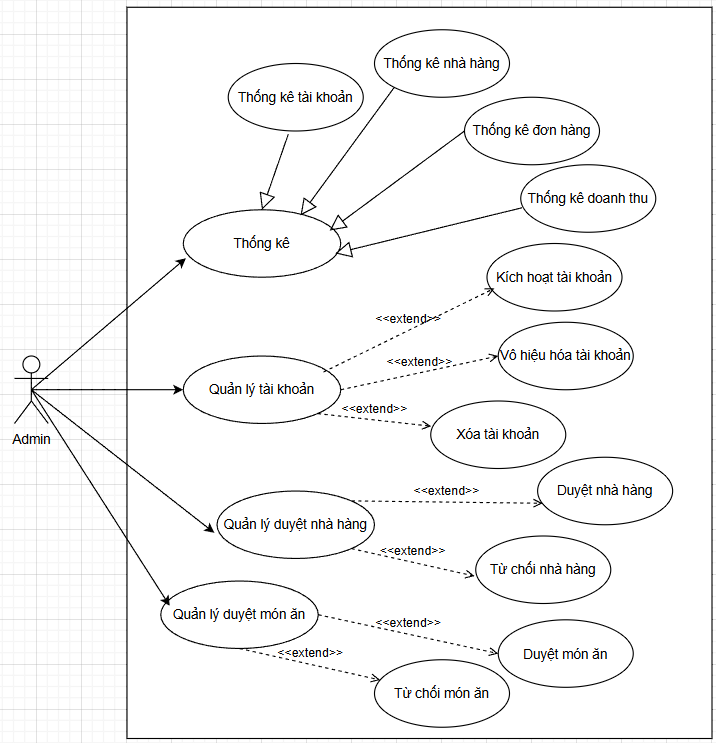
1. **Biểu đồ Use case**

- Biểu đồ Use case tổng quát



1. Biểu đồ use case tổng quát

- Biểu đồ use case admin



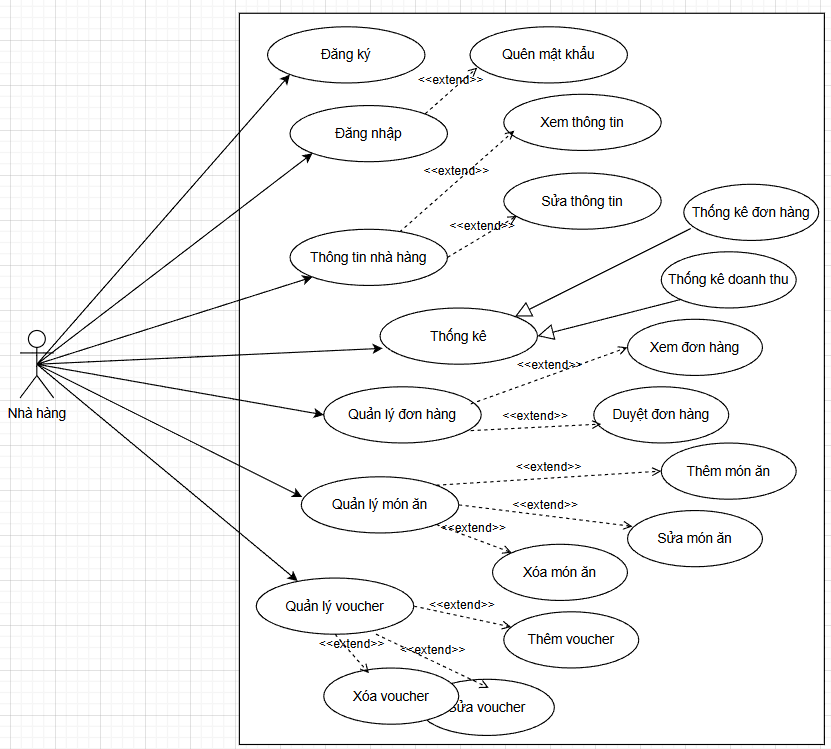
1. Biểu đồ use case admin

- Biểu đồ use case khách hàng



1. Biểu đồ use case khách hàng

- Biểu đồ use case nhà hàng



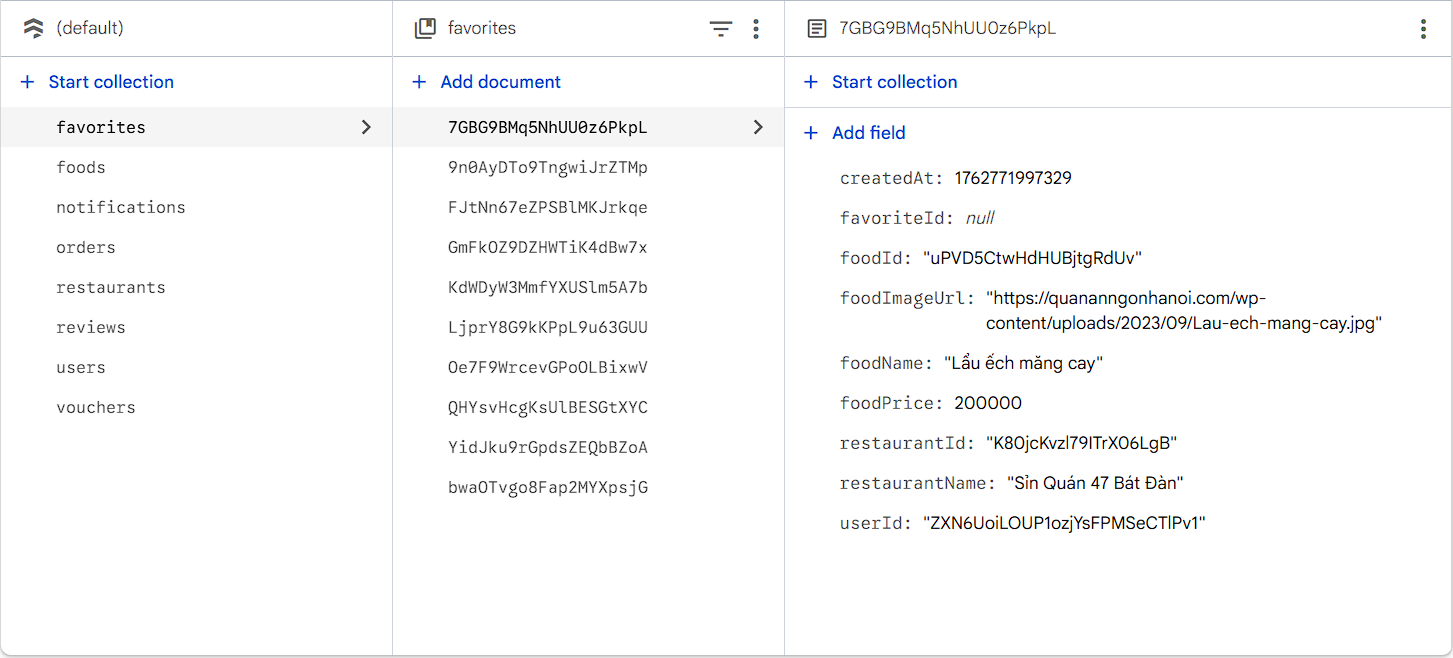
1. Biểu đồ use case nhà hàng

1. **Thiết kế cơ sở dữ liệu**

Ứng dụng gồm các bảng dữ liệu sau:

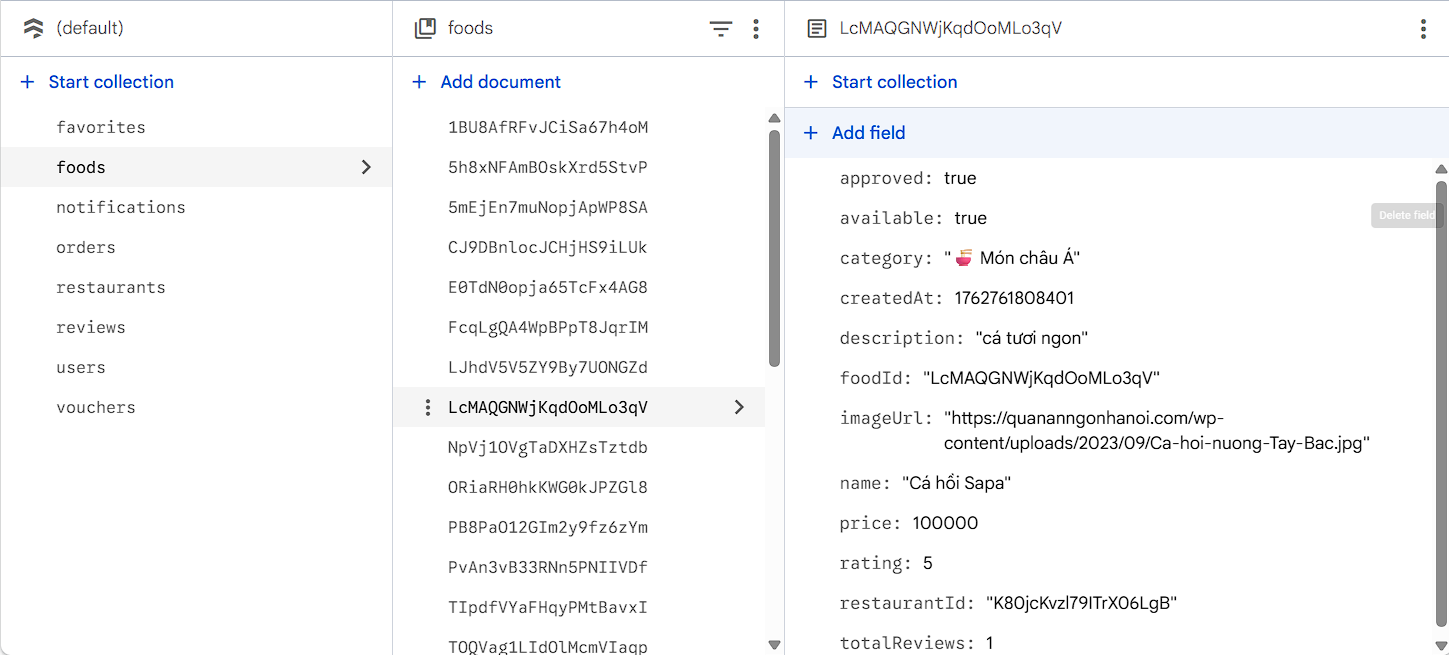
1. Bảng dữ liệu yêu thích

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| createdAt | Timestamp | Thời gian tạo tài liệu |
| favoriteId | String | ID của mục yêu thích |
| foodId | String | ID của món ăn trong danh sách yêu thích |
| foodName | String | Tên của món ăn |
| foodImageUrl | String | Đường dẫn URL đến hình ảnh của món ăn |
| foodPrice | Number | Giá của món ăn |
| restaurantId | String | ID của nhà hàng |
| restaurantName | String | Tên của nhà hàng |
| userId | String | ID của người dùng |



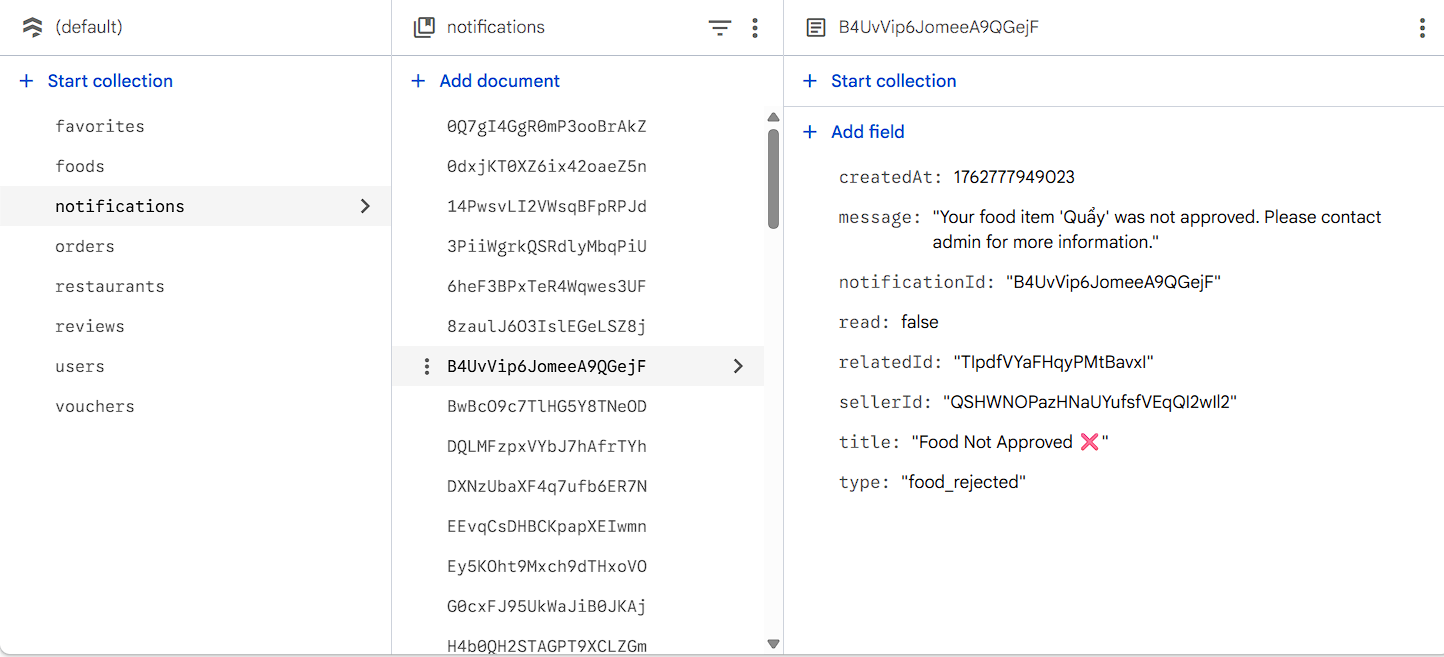
1. Mô tả dữ liệu yêu thích
2. Bảng dữ liệu món ăn

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| approved | Boolean | Trạng thái phê duyệt món ăn (true/false) |
| available | Boolean | Trạng thái có sẵn của món ăn (true/false) |
| category | String | Thể loại của món ăn |
| createdAt | Timestamp | Thời gian thêm món ăn |
| description | String | Mô tả ngắn về món ăn |
| foodId | String | ID của món ăn |
| imageUrl | String | Đường dẫn URL đến hình ảnh của món ăn |
| name | String | Tên của món ăn |
| price | Number | Giá của món ăn |
| rating | Number | Điểm đánh giá của món ăn (từ 1 đến 5) |
| restaurantId | String | ID của nhà hàng liên kết với món ăn |
| totalReviews | Number | Tổng số đánh giá về món ăn |



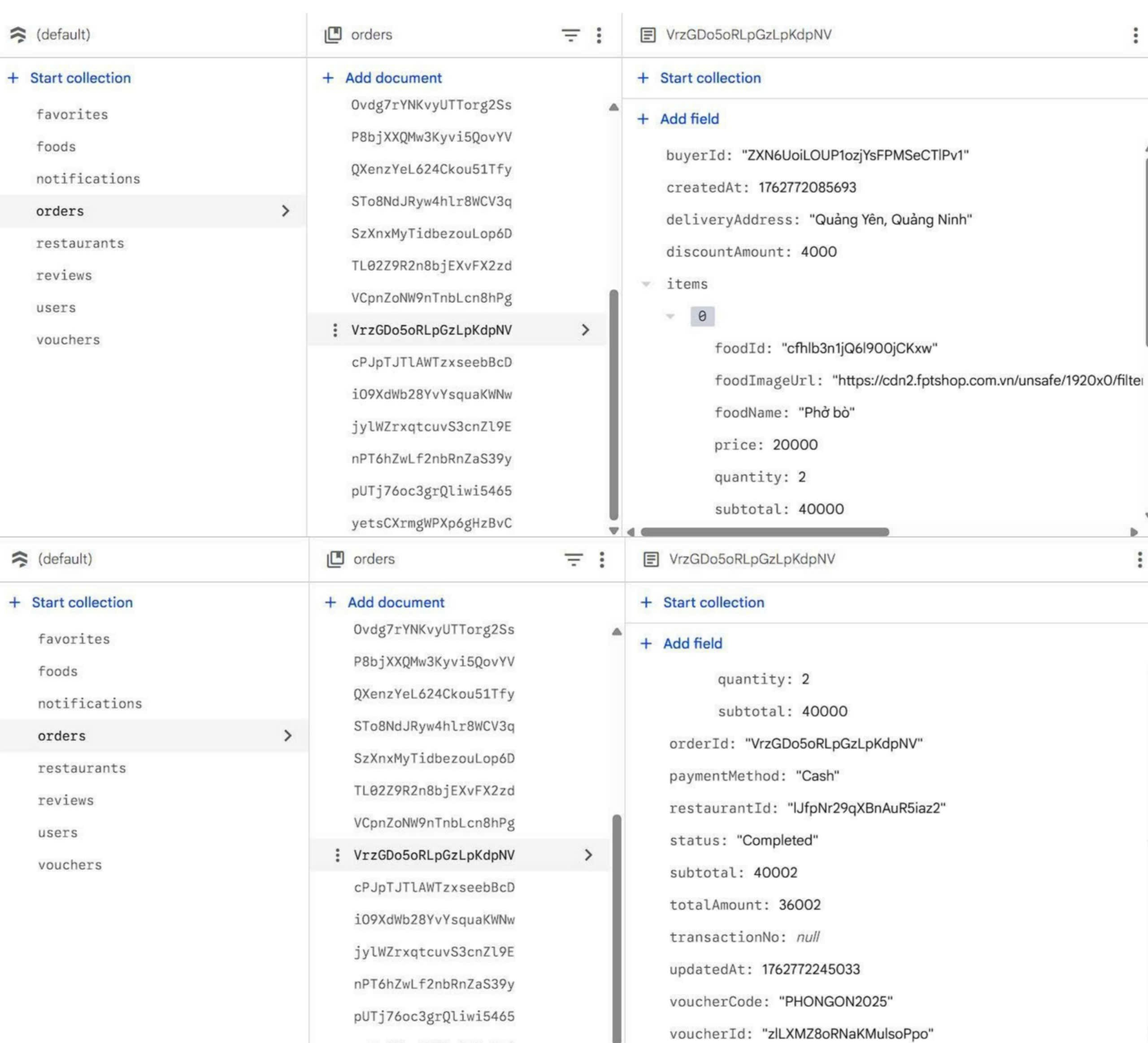
1. Mô tả dữ liệu món ăn
2. Bảng dữ liệu thông báo

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| createdAt | Timestamp | Thời gian tạo thông báo |
| message | String | Nội dung thông báo |
| notificationId | String | ID của thông báo |
| read | Boolean | Trạng thái đã đọc của thông báo (true/false) |
| relatedId | String | ID liên quan đến thông báo |
| sellerId | String | ID của người bán liên quan đến thông báo |
| title | String | Tiêu đề của thông báo |
| type | String | Loại thông báo |



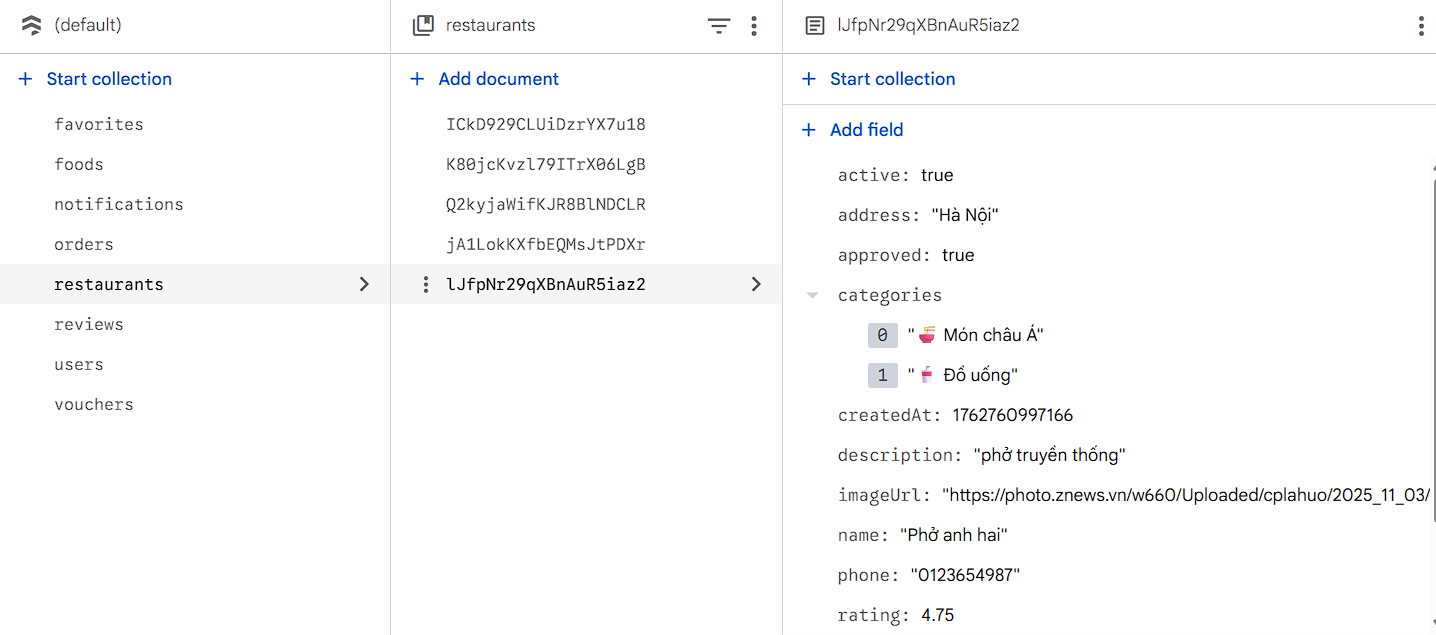
1. Mô tả dữ liệu thông báo
2. Bảng dữ liệu đơn hàng

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| buyerId | String | ID của người mua liên quan đến đơn hàng |
| createdAt | Timestamp | Thời gian tạo đơn hàng |
| deliveryAddress | String | Địa chỉ giao hàng |
| discountAmount | Number | Số tiền giảm giá áp dụng cho đơn hàng |
| items | Array | Danh sách các món ăn trong đơn hàng |
| foodId | String | ID của món ăn |
| foodImageUrl | String | Đường dẫn đến ảnh của món ăn |
| foodName | String | Tên món ăn |
| price | Number | Giá mỗi món ăn |
| quantity | Number | Số lượng của món ăn trong đơn hàng |
| subtotal | Number | Thành tiền của từng món ăn (giá \* số lượng) |
| orderId | String | ID của đơn hàng |
| paymentMethod | String | Phương thức thanh toán |
| restaurantId | String | ID của nhà hàng mà đơn hàng được đặt |
| status | String | Trạng thái đơn hàng |
| subtotal | Number | Tổng tiền trước khi tính các khoản giảm giá |
| totalAmount | Number | Tổng số tiền cần thanh toán sau khi trừ giảm giá |
| transactionNo | String | Số giao dịch |
| updatedAt | Timestamp | Thời gian cập nhật đơn hàng gần nhất |
| voucherCode | String | Mã giảm giá áp dụng cho đơn hàng nếu có |
| voucherId | String | ID của voucher giảm giá nếu có |



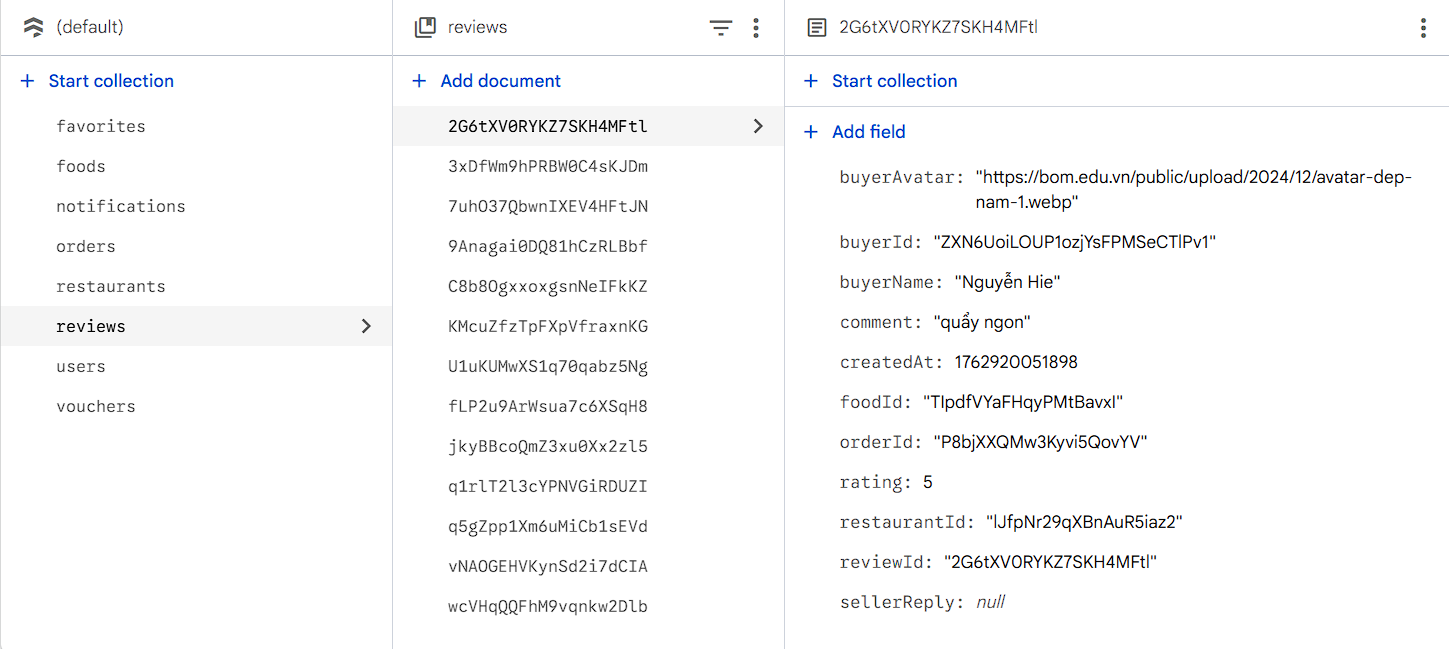
1. Mô tả dữ liệu đơn hàng
2. Bảng dữ liệu nhà hàng

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| active | Boolean | Trạng thái hoạt động của nhà hàng (true/false) |
| address | String | Địa chỉ của nhà hàng |
| approved | Boolean | Trạng thái phê duyệt (true nếu đã được phê duyệt) |
| categories | Array | Danh sách các loại món ăn mà nhà hàng phục vụ |
| createdAt | Timestamp | Thời gian tạo nhà hàng |
| description | String | Mô tả về nhà hàng |
| imageUrl | String | Đường dẫn đến ảnh đại diện của nhà hàng |
| name | String | Tên của nhà hàng |
| phone | String | Số điện thoại liên hệ của nhà hàng |
| rating | Number | Đánh giá trung bình của nhà hàng |
| restaurantId | String | ID của nhà hàng |
| sellerId | String | ID của người bán |
| totalReviews | Number | Tổng số đánh giá mà nhà hàng nhận được |



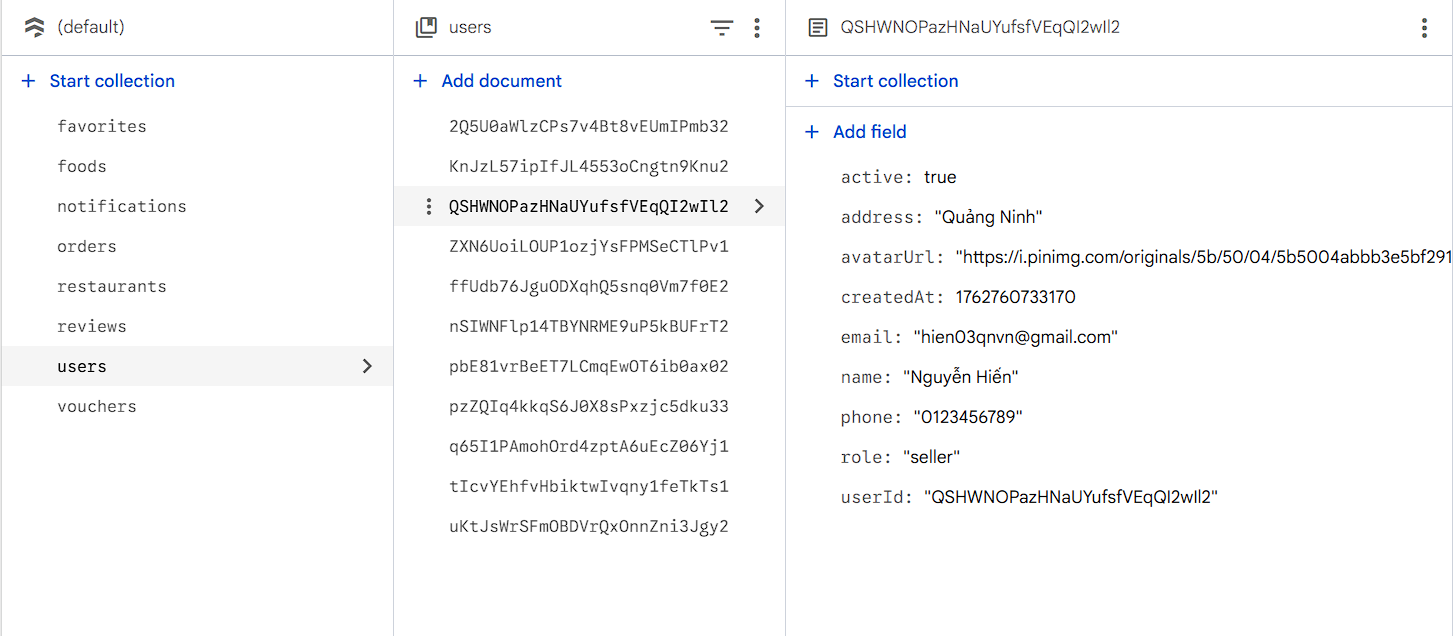
1. Mô tả dữ liệu nhà hàng
2. Bảng dữ liệu đánh giá

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| buyerAvatar | String | Đường dẫn đến ảnh đại diện của người mua |
| buyerId | String | ID của người mua |
| buyerName | String | Tên của người mua |
| comment | String | Nội dung đánh giá của người mua |
| createdAt | Timestamp | Thời gian đánh giá được tạo |
| foodId | String | ID của món ăn được đánh giá |
| orderId | String | ID của đơn hàng mà đánh giá liên quan đến |
| rating | Number | Điểm đánh giá |
| restaurantId | String | ID của nhà hàng mà món ăn đã được đánh giá |
| reviewId | String | ID của bài đánh giá |



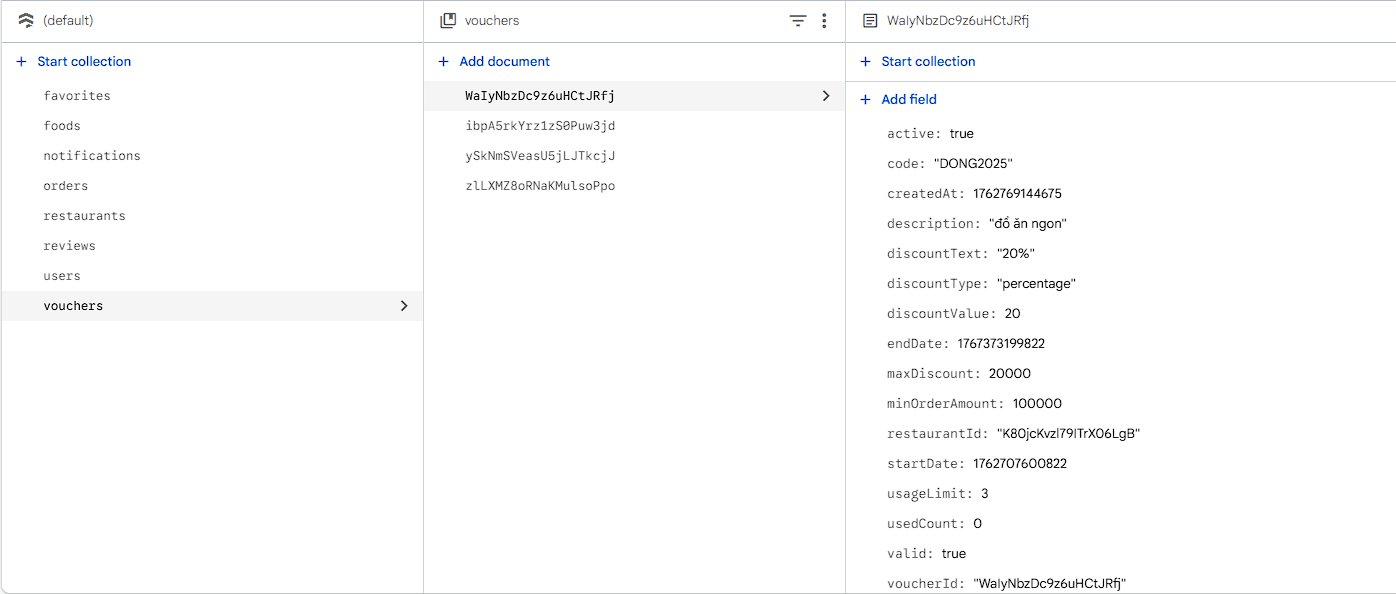
1. Mô tả dữ liệu đánh giá
2. Bảng dữ liệu tài khoản

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| active | Boolean | Trạng thái hoạt động của người dùng (true/false) |
| address | String | Địa chỉ của người dùng |
| avatarUrl | String | Đường dẫn đến ảnh đại diện của người dùng |
| createdAt | Timestamp | Thời gian người dùng được tạo tài khoản |
| email | String | Địa chỉ email của người dùng |
| name | String | Tên của người dùng |
| phone | String | Số điện thoại của người dùng |
| role | String | Vai trò của người dùng |
| userId | String | ID của người dùng |



1. Mô tả dữ liệu tài khoản
2. Bảng dữ liệu vouchers

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| active | Boolean | Trạng thái hoạt động của voucher (true/false) |
| code | String | Mã voucher |
| createdAt | Timestamp | Thời gian voucher được tạo |
| description | String | Mô tả ngắn về voucher |
| discountText | String | Nội dung mô tả loại giảm giá |
| discountType | String | Loại giảm giá |
| discountValue | Number | Giá trị giảm giá (số tiền hoặc tỷ lệ phần trăm) |
| endDate | Timestamp | Thời điểm hết hạn của voucher |
| maxDiscount | Number | Giá trị tối đa của số tiền giảm giá |
| minOrderAmount | Number | Số tiền tối thiểu để áp dụng voucher |
| restaurantId | String | ID của nhà hàng liên kết với voucher |
| startDate | Timestamp | Thời gian bắt đầu áp dụng voucher |
| usageLimit | Number | Giới hạn số lần sử dụng voucher |
| usedCount | Number | Số lần đã sử dụng voucher |
| valid | Boolean | Trạng thái hợp lệ của voucher (true/false) |
| voucherId | String | ID của voucher |



1. Mô tả dữ liệu vouchers

# CHƯƠNG 3. XÂY DỰNG CHƯƠNG TRÌNH

## Xây dựng giao diện người dùng

### *Phân loại người dùng hệ thống*

Hệ thống được thiết kế phục vụ ba nhóm người dùng chính với các vai trò và quyền hạn khác nhau:

- Khách hàng: Người sử dụng ứng dụng để xem món ăn, đặt món, thanh toán và theo dõi đơn hàng.

- Nhà hàng: Người quản lý thực đơn, nhận và xử lý đơn đặt hàng từ khách.

- Admin: Người quản trị toàn hệ thống, quản lý tài khoản, cửa hàng, món ăn và dữ liệu trên Firestore.

### *Giao diện khách hàng*

Yêu cầu thiết kế giao diện:

- Giao diện thân thiện, bố cục rõ ràng, giúp người dùng dễ dàng tìm kiếm món ăn và thao tác đặt hàng.

- Hiển thị trực quan trạng thái đơn hàng với các bước: Chờ xác nhận – Đang giao – Đã hoàn thành, giúp khách hàng dễ dàng theo dõi tiến trình.

- Danh sách món ăn được trình bày rõ ràng, hiển thị hình ảnh, mô tả, giá và số lượng để khách hàng lựa chọn.

- Giao diện chi tiết đơn hàng thể hiện đầy đủ: cửa hàng nhận đơn, danh sách món, giá từng món, tổng tiền, ghi chú và phương thức thanh toán.

- Khách hàng được phép chỉnh sửa đơn trong giao diện nếu đơn vẫn đang ở trạng thái Chờ xác nhận. Khi cửa hàng đã xác nhận, giao diện sẽ khóa chức năng chỉnh sửa/hủy món.

- Dữ liệu đơn hàng được đồng bộ thời gian thực từ Firestore, đảm bảo thông tin trên giao diện luôn chính xác và cập nhật liên tục.

### *Giao diện nhà hàng*

Yêu cầu thiết kế giao diện:

- Menu quản lý được bố trí rõ ràng, dễ thao tác cho chủ cửa hàng.

- Giao diện hiển thị đầy đủ thông tin cửa hàng như tên, ảnh đại diện, địa chỉ, thông tin liên hệ và trạng thái hoạt động.

- Màn hình quản lý món ăn được thiết kế trực quan, hỗ trợ xem nhanh hình ảnh, mô tả, loại món và giá tiền của từng món.

- Chức năng chỉnh sửa giá và thông tin món được bố trí thuận tiện để nhà hàng có thể cập nhật menu bất cứ lúc nào.

- Phần thống kê giúp cửa hàng theo dõi số lượng đơn hàng và doanh thu.

- Giao diện xử lý đơn hàng được thiết kế hiển thị rõ các trạng thái: Đang xử lý – Đang giao – Hoàn thành.

- Mỗi đơn hàng hiển thị đầy đủ thông tin: danh sách món, số lượng, ghi chú, tổng tiền, địa chỉ giao và phương thức thanh toán.

- Tự động cập nhật thông tin đơn hàng từ Firestore, giúp cửa hàng theo dõi và xử lý đơn kịp thời.

### *Giao diện quản trị viên*

Yêu cầu thiết kế giao diện:

- Giao diện tổng quan trực quan, hiển thị đầy đủ các thống kê quan trọng của hệ thống như tổng số đơn hàng, tổng doanh thu, số lượng nhà hàng đang hoạt động và số lượng tài khoản người dùng.

- Hỗ trợ chức năng quản lý tài khoản, cho phép Admin xem danh sách người dùng, duyệt tài khoản nhà hàng mới, khóa hoặc mở khóa tài khoản vi phạm.

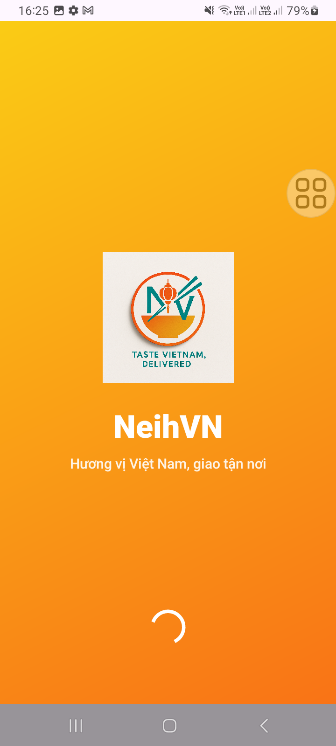
- Màn hình quản lý nhà hàng và món ăn được trình bày rõ ràng, cho phép Admin duyệt nhà hàng và món ăn.

- Cơ chế phân quyền chặt chẽ, đảm bảo chỉ Admin mới có quyền truy cập và chỉnh sửa các thông tin hệ thống quan trọng.

- Tất cả dữ liệu được đồng bộ thời gian thực từ Firestore, giúp Admin luôn nắm được tình trạng mới nhất của toàn bộ hệ thống đặt đồ ăn online.

## Kết quả thực hiện

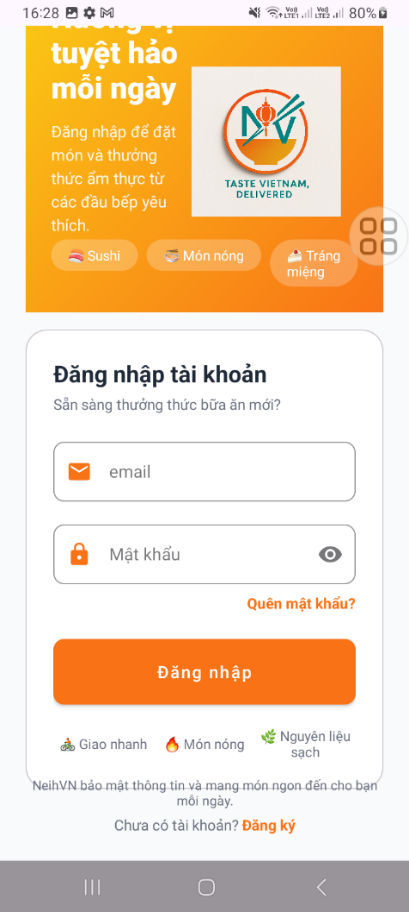
### *Giao diện khởi chạy ứng dụng*



1. Giao diện khởi chạy ứng dụng

Màn hình giao diện khởi chạy ứng dụng.

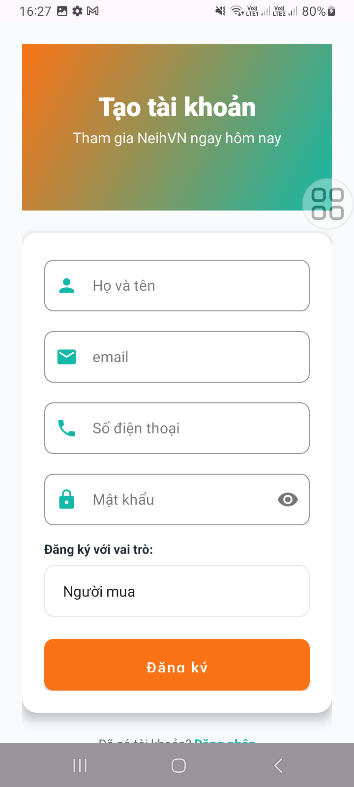
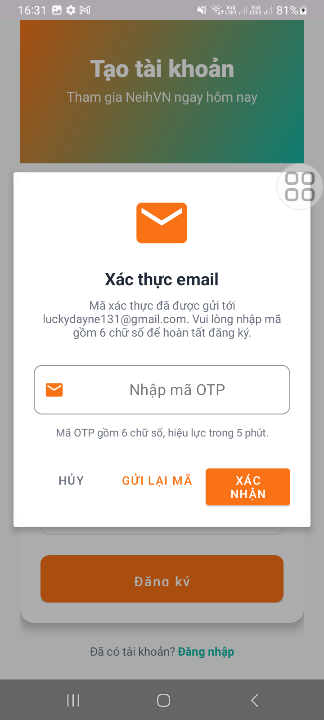
### *Giao diện đăng nhập và đăng ký*

**Giao diện màn hình đăng nhập****:**

1. Giao diện đăng nhập tài khoản

Người mua và người bán cần đăng nhập tài khoản để truy cập vào app đặt đồ ăn, nếu chưa có tài khoản người dùng sẽ cần đăng ký tài khoản, nếu đã có tài khoản người dùng nhập email, mật khẩu và bấm đăng nhập để sử dụng ứng dụng, người dùng quên mật khẩu có thể bấm vào quên mật khẩu hệ thồng sẽ gửi email đặt mật khẩu mới.

**Giao diện màn hình đăng ký:**

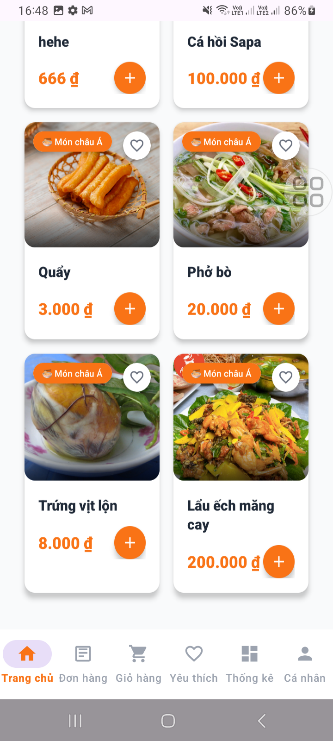
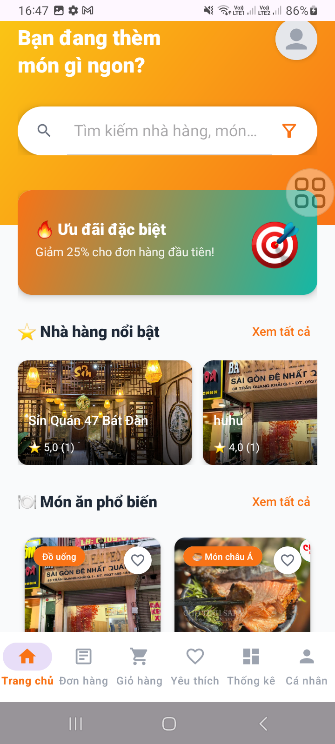
 

1. Giao diện đăng ký tài khoản

Người dùng cần nhập họ tên, email, số điện thoại, mật khẩu, chọn loại tài khoản và bấm nút đăng ký để đăng ký tài khoản, người dùng kiểm tra mã OTP được gửi về email và nhập mã OTP để xác nhận email.

### *Giao diện người dùng*

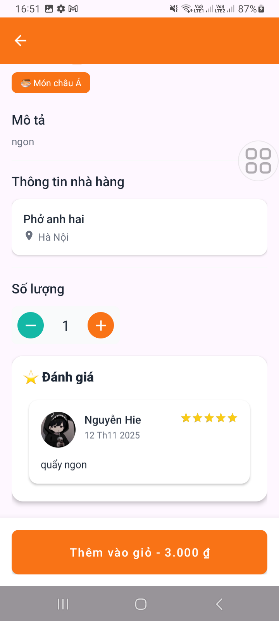
**Giao diện trang chủ:**



1. Giao diện trang chủ

Phần giao diện trang chủ được đăng nhập bằng tài khoản người mua chúng ta được xem danh sách các nhà hàng nổi bật cùng với các món ăn được ưa chuộng nhất, bên cạnh đó còn có nút tìm kiếm nhà hàng, món ăn.

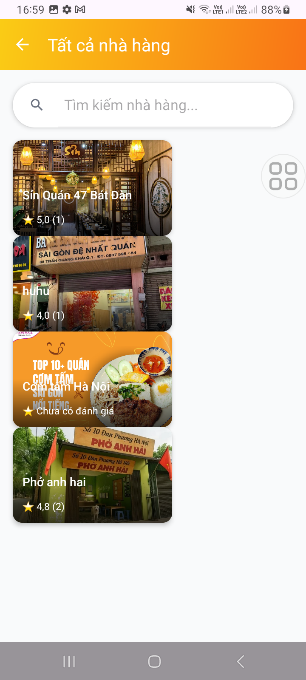
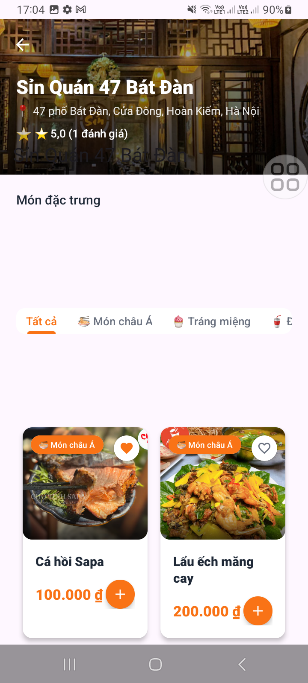
**Giao diện chi tiết sản phẩm:**



1. Giao diện chi tiết sản phẩm

Phần giao diện trang chi tiết sản phẩm cho phép người mua xem các thông tin về món ăn địa chỉ của nhà hàng, các đánh giá của người mua trước đó, bấm cộng trừ để tăng giảm số lượng món và bấm thêm vào giỏ để thêm món ăn vào giỏ hàng.

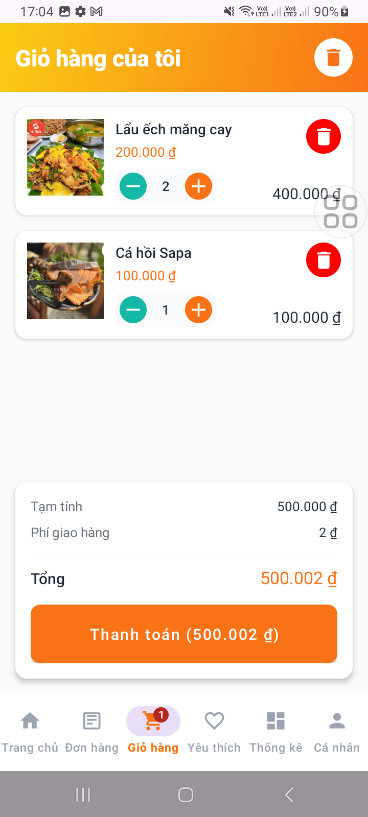
**Giao diện chọn nhà hàng và chọn món:**

1. Giao diện chọn nhà hàng và chọn món

Ở giao diện chọn nhà hàng chúng ta sẽ có tính năng bao gồm việc: tìm kiếm nhà hàng yêu thích và xem các món ăn của nhà hàng đó.

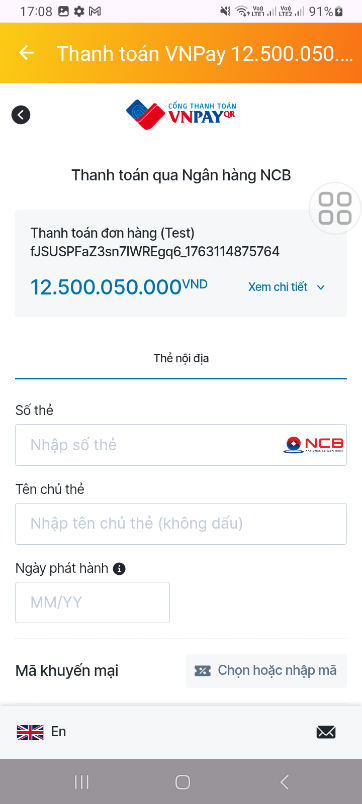
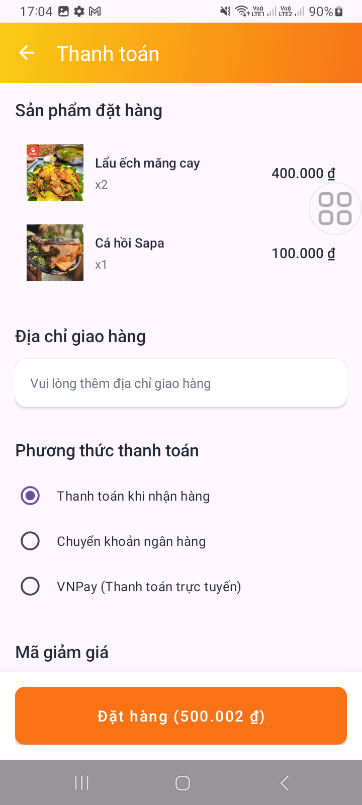
**Giao diện giỏ hàng:**



1. Giao diện giỏ hàng

Người mua vào phần giỏ hàng điều chỉnh số lượng món ăn hoặc xóa đồ ăn khỏi giỏ hàng, bấm thanh toán để mua đồ ăn.

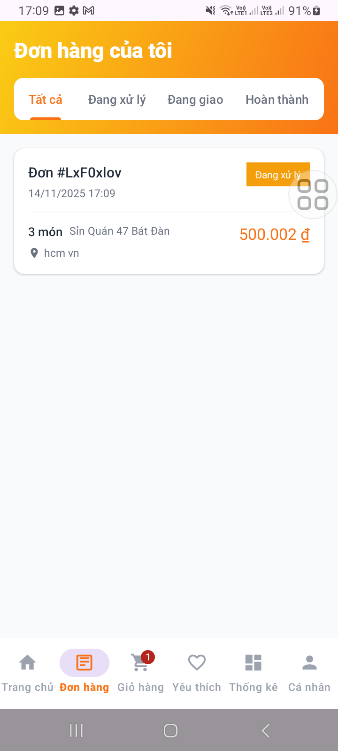
**Giao diện đặt hàng và thanh toán:**



1. Giao diện đặt hàng và thanh toán

Người mua chọn phương thức thanh toán, nhập mã giảm giá nếu có và bấm đặt hàng rồi chờ nhà hàng xác nhận đơn hàng.

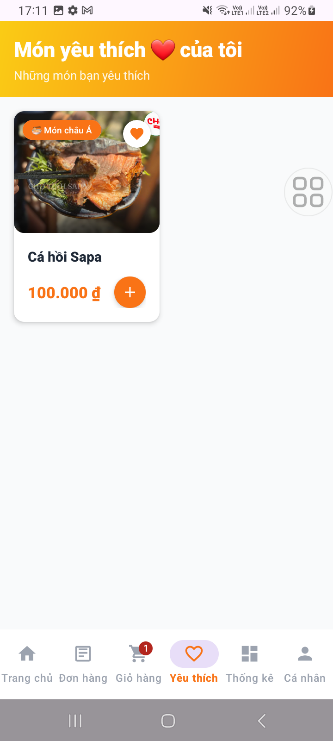
**Giao diện tra cứu tình trạng đơn hàng:**



1. Giao diện tra cứu tình trạng đơn hàng

Người mua vào phần đơn hàng để xem tình trạng đơn hàng.

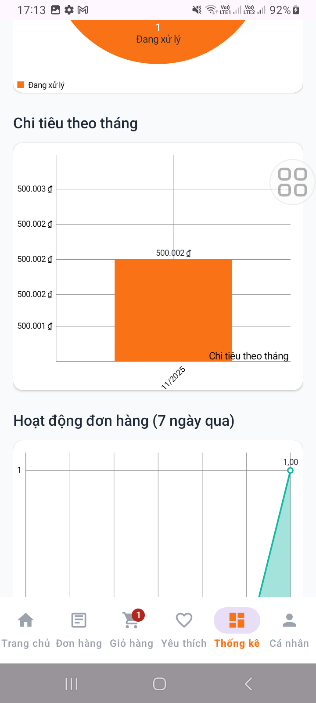
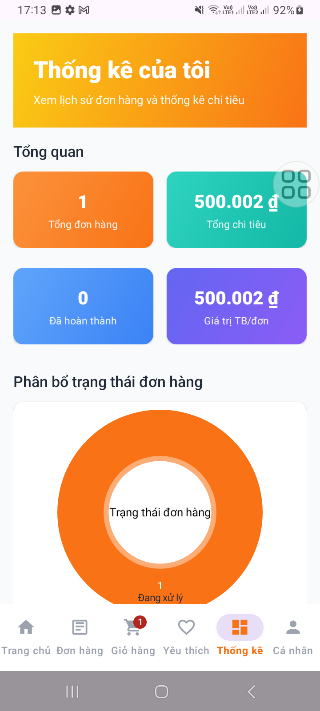
**Giao diện yêu thích sản phẩm:**



1. Giao diện yêu thích sản phẩm

Người mua bấm vào hình trái tim trên hình món ăn để thêm món ăn vào mục món ăn yêu thích.

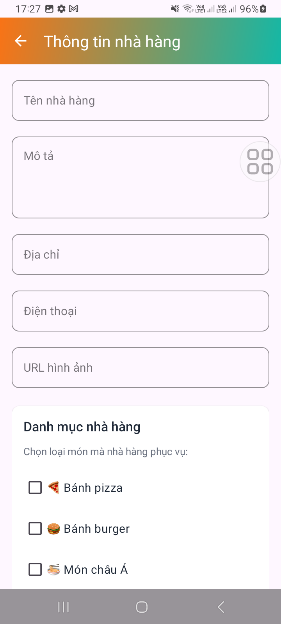
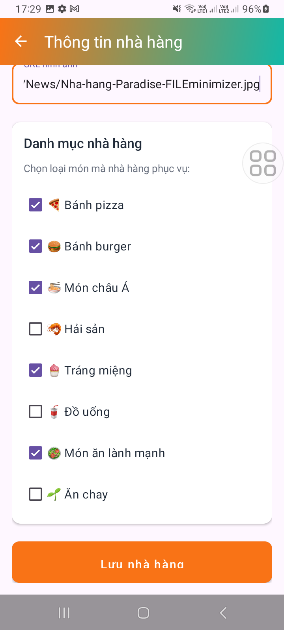
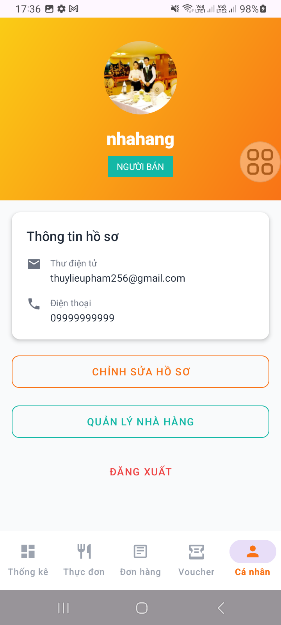
**Giao diện quản lý thống kê mua hàng:**



1. Giao diện thống kê mua hàng

Người mua vào phần thống kê để xem thống kê mua hàng.

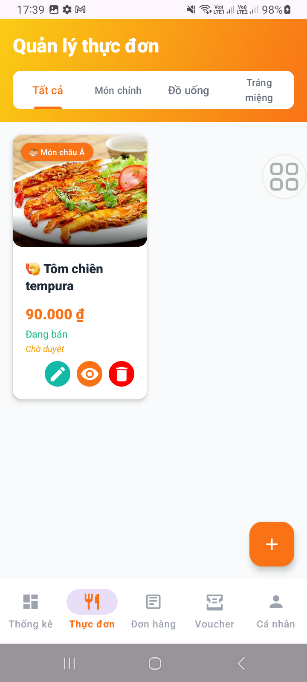
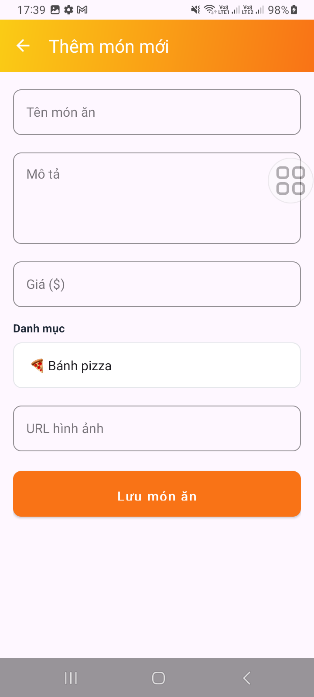
**Giao diện thiết lập nhà hàng:**

1. Giao diện thiết lập nhà hàng

Người bán cần nhập thông tin nhà hàng, thiết lập danh mục nhà hàng rồi chờ quản trị viên duyệt.

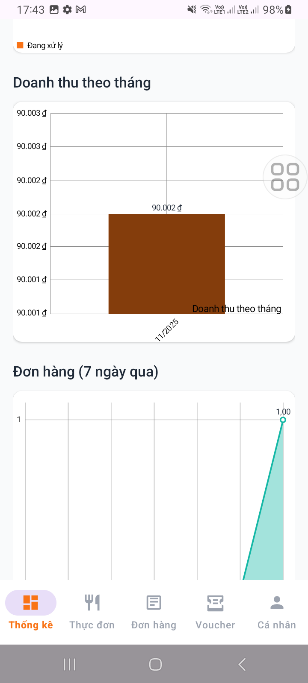
**Giao diện quản lý món ăn:**



1. Giao diện quản lý món ăn

Người bán vào phần thực đơn để thêm, sửa và xóa món ăn.

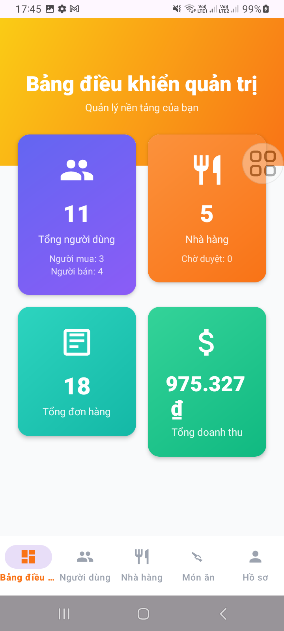
**Giao diện quản lý đơn hàng và thống kê:**



1. Giao diện quản lý hóa đơn và thống kê

Người bán vào phần đơn hàng để xác nhận đơn hàng, vào phần thống kê để xem thống kê tổng quát.

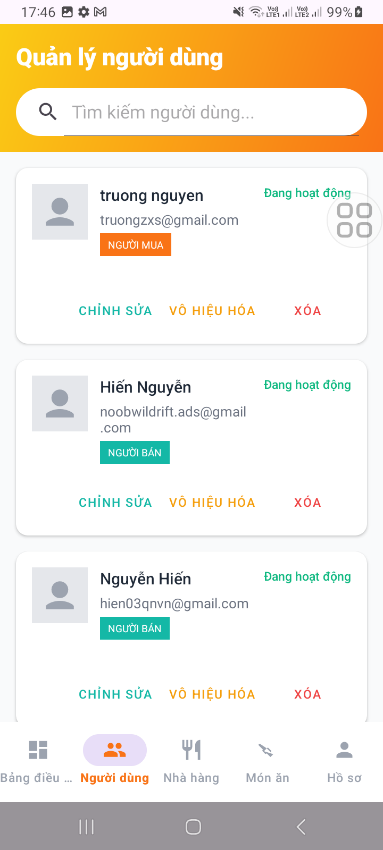
**Giao diện quản trị thống kê:**



1. Giao diện quản trị thống kê

Quản trị viên vào phần bảng điều khiển để xem thống kê tổng quát.

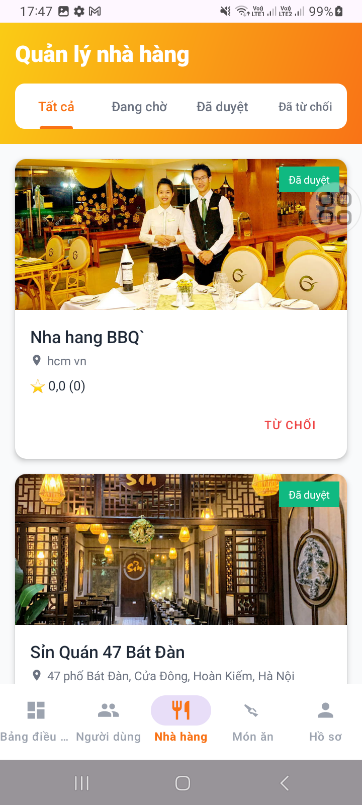
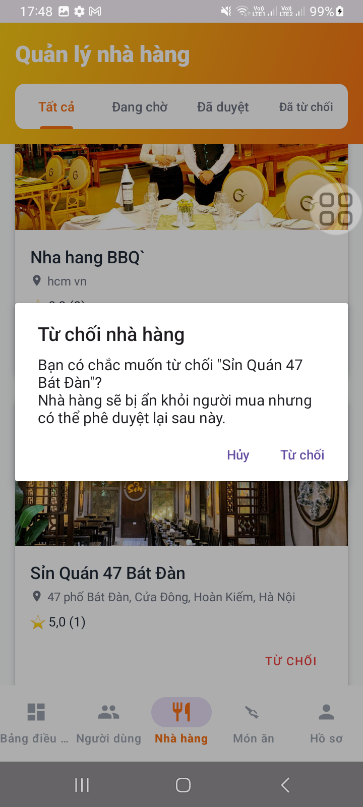
**Giao diện quản lý tài khoản:**



1. Giao diện quản lý tài khoản

Quản trị viên vào phần người dùng để quản lý tài khoản.

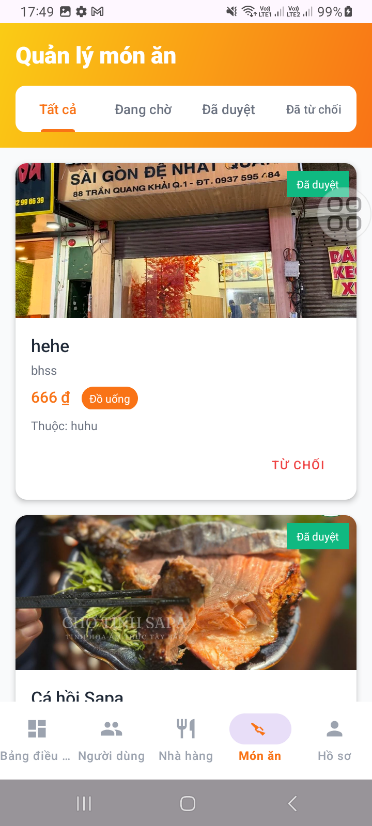
**Giao diện quản lý duyệt nhà hàng:**



1. Giao diện quản lý duyệt nhà hàng

Quản trị viên vào phần nhà hàng để duyệt nhà hàng.

**Giao diện quản lý duyệt món ăn:**



1. Giao diện quản lý duyệt món ăn

Quản trị viên vào phần món ăn để duyệt món ăn của các nhà hàng.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* **Đánh giá**
* **Ưu điểm:**

Ứng dụng đã hoàn thành tốt các mục tiêu đặt ra trong đồ án, đặc biệt là xây dựng được một hệ thống hỗ trợ hiệu quả cho công tác quản lý đặt đồ ăn online và hoạt động của nhà hàng. Hệ thống vận hành ổn định và đáp ứng được các nghiệp vụ cơ bản của một ứng dụng quản lý chuyên nghiệp.

Giao diện người dùng được thiết kế theo hướng trực quan, dễ hiểu, phù hợp với cả người dùng phổ thông lẫn nhân viên nhà hàng; giúp thao tác trở nên nhanh chóng, thuận lợi và giảm thiểu sai sót trong quá trình sử dụng.

Khả năng truy cập và xử lý dữ khá tốt, đảm bảo kết nối nhanh, ổn định, không gây ảnh hưởng đến các cơ sở dữ liệu khác và hỗ trợ việc quản lý đồng bộ trên nhiều thiết bị.

Hình ảnh các món ăn được trình bày sinh động, rõ nét, giao diện tổng thể mang tính thẩm mỹ cao, giúp người dùng có trải nghiệm trực quan và thúc đẩy quyết định đặt món.

Ứng dụng hỗ trợ quản lý chi tiết và hiệu quả các danh mục như: Tài khoản người dùng, thông tin nhà hàng, loại món ăn, món ăn, giá món, theo dõi lịch sử đơn đặt hàng, lập hóa đơn và thống kê doanh thu. Điều này góp phần nâng cao tính chuyên nghiệp và khả năng vận hành của hệ thống.

* **Nhược điểm:**

Một số chức năng nâng cao vẫn chưa được phát triển đầy đủ như mong muốn, khiến ứng dụng chưa đạt mức hoàn thiện tối ưu. Giao diện tuy dễ sử dụng nhưng chưa được tối ưu hóa về trải nghiệm người dùng và tính hiện đại.

Trong quá trình vận hành thử nghiệm, ứng dụng có thể vẫn còn tồn tại một số lỗi kỹ thuật tiềm ẩn liên quan đến thao tác nghiệp vụ, xử lý dữ liệu hoặc kết nối hệ thống do hạn chế về kỹ năng lập trình và kinh nghiệm triển khai của nhóm.

Tiến độ thực hiện đồ án còn chậm, khiến việc hoàn thiện các chức năng chưa trọn vẹn. Một vài tính năng nâng cao như gợi ý món ăn theo người dùng, theo dõi đơn hàng theo thời gian thực, tích hợp thanh toán trực tuyến,… chưa được phát triển.

* **Kết luận**

Ứng dụng đã cơ bản đáp ứng các yêu cầu nền tảng của một hệ thống đặt đồ ăn online hiện đại, bao gồm quản lý món ăn, quản lý đơn hàng, thống kê doanh thu và hỗ trợ tương tác giữa khách hàng với nhà hàng. Điều này chứng tỏ sản phẩm có tiềm năng lớn để tiếp tục phát triển, hoàn thiện và nâng cấp thành một ứng dụng chính thức có khả năng đưa vào sử dụng thực tế.

Tuy nhiên, bên cạnh những ưu điểm đạt được, ứng dụng vẫn còn một số tồn tại liên quan đến hiệu năng, giao diện và tính năng. Các vấn đề này cần được nghiên cứu, khắc phục và cải thiện để đảm bảo ứng dụng hoạt động ổn định, mượt mà và phù hợp với nhu cầu thị trường.

* **Hướng phát triển**

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ và xu hướng số hóa ngày càng phổ biến, việc tiếp tục mở rộng và nâng cấp ứng dụng đặt đồ ăn trực tuyến là rất cần thiết. Để đáp ứng tốt nhu cầu thực tế, nhóm cần tiếp tục nghiên cứu và tích hợp thêm nhiều chức năng cải tiến như: Theo dõi trạng thái đơn hàng theo thời gian thực, gợi ý món ăn dựa trên hành vi người dùng, tích hợp hệ thống thanh toán đa nền tảng, tối ưu hóa giao diện theo xu hướng mới và tăng cường bảo mật thông tin người dùng.  
Bên cạnh đó, ứng dụng có thể mở rộng sang nhiều mảng khác như quản lý giao hàng, quản lý chuỗi nhà hàng, đặt bàn trực tuyến và tự động hóa quy trình phục vụ. Dựa trên nền tảng hiện tại, đây sẽ là cơ sở quan trọng để phát triển thêm nhiều hệ thống quản lý trong các lĩnh vực khác nhau. Trên cơ sở kế thừa những thành tựu đã đạt được, nhóm cần tiếp tục khắc phục các hạn chế, tối ưu toàn bộ chương trình và đảm bảo ứng dụng sẵn sàng đưa vào thực tế với hiệu quả cao nhất, đem lại trải nghiệm tiện lợi, nhanh chóng và đáng tin cậy cho người sử dụng.

# TÀI LIỆU THAM KHẢO

1. **Sách tham khảo**
2. Nguyễn Thị Lan. (2018). *Hướng dẫn lập trình Java cho ứng dụng di động*. Nhà xuất bản Giáo dục.
3. Lê Thị Thanh Hương. (2020). *Xây dựng ứng dụng di động với Android và Java*. Nhà xuất bản Bài Giảng.
4. Phạm Văn Thành. (2021). Lập trình Android nâng cao với Firebase. Nhà xuất bản Khoa học & Kỹ thuật.
5. Trần Minh Quang. (2019). Cơ sở dữ liệu NoSQL và ứng dụng Firebase Firestore. Nhà xuất bản Thông tin & Truyền thông.
6. **Trang web tham khảo**
7. <https://github.com/> (Ngày truy cập 10/8/2025)
8. <https://firebase.google.com/docs/auth> (Ngày truy cập 12/8/2025)
9. <https://firebase.google.com/docs/firestore> (Ngày truy cập 15/8/2025)
10. <https://developer.android.com/topic/architecture> (Ngày truy cập 18/8/2025)
11. <https://developer.android.com/studio/intro> (Ngày truy cập 20/8/2025)