

**BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



BÁO CÁO ĐỒ ÁN MÔN HỌC

ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG MUA HÀNG NÔNG SẢN DI ĐỘNG

Môn học: PHÁT TRIỂN ỨNG DỤNG CHO CÁC THIẾT BỊ DI ĐỘNG

Giảng viên hướng dẫn: THS. NGUYỄN TRUNG HIẾU

Thực hiện bởi nhóm sinh viên, bao gồm:

1. BÙI HỮU TRÍ	N21DCAT058	Trưởng nhóm
2. ĐINH QUỐC TOÀN	N21DCAT057	Thành viên
3. VÕ HỒNG NGUYỄN	N21DCAT036	Thành viên
4. NGUYỄN BÁ TRUNG	N21DCAT060	Thành viên

TP.HCM, tháng 05 /2025

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LỜI CẢM ƠN

Nhóm chúng em xin chân thành cảm ơn thầy Nguyễn Trung Hiếu trong thời gian qua đã dạy, hướng dẫn, giúp đỡ chúng em trong quá trình học tập. Giúp chúng em nắm vững kiến thức của môn học **PHÁT TRIỂN ỨNG DỤNG CHO CÁC THIẾT BỊ DI ĐỘNG**. Giúp chúng em có cái nhìn sâu hơn về lập trình phát triển ứng dụng di động Java. Chúc thầy luôn luôn mạnh khỏe, luôn vui tươi, dồi dào sức sống và có nhiều thành công trong công việc giảng dạy và động viên chúng em trong suốt quá trình học tập để thực hiện đề tài “**XÂY DỰNG ỨNG DỤNG MUA HÀNG NÔNG SẢN DI ĐỘNG**” này.

Nhóm sinh viên thực hiện.

NHÓM 06

MỤC LỤC

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	2
LỜI CẢM ƠN	3
MỤC LỤC	4
DANH MỤC TỪ VIẾT TẮT	5
CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI	8
1.1 Lý do chọn đề tài	8
1.2 Mục tiêu của đề tài	8
1.3 Công nghệ sử dụng	9
1.4 Cấu trúc báo cáo	9
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	9
2.1 Cơ sở lý thuyết	10
2.1.1 Android studio	10
2.1.2 Firebase Realtime Database	10
2.2 Phân tích và Thiết kế hệ thống	11
CHƯƠNG 3 THỰC NGHIỆM	21
3.1 Cài đặt môi trường và kết nối Firebase	21
3.2 Xây dựng các chức năng chính	21
CHƯƠNG 4 TỔNG KẾT	46
4.1 Kết quả đạt được	46
4.2 Hạn chế	47
4.3 Hướng phát triển	47
TÀI LIỆU THAM KHẢO	51

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Viết đầy đủ
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
CRUD	Create Read Update Delete
COD	Cash on Delivery
SDK	Software Development Kit

DANH MỤC HÌNH ẢNH

Hình 1: Cấu trúc JSON	13
Hình 2: Cấu trúc của Categories	14
Hình 3: Usecase tổng quát	14
Hình 4: Usecase đăng ký tài khoản	15
Hình 5: Usecase đăng nhập	17
Hình 6: Usecase quản lý sản phẩm	18
Hình 7: Usecase quản lý giỏ hàng	19
Hình 8: Usecase quản lý đơn hàng	20
Hình 9: Giao diện đăng ký	22
Hình 10: Mã minh họa cho chức năng đăng ký	23
Hình 11: Giao diện đăng nhập	24
Hình 12: Danh mục sản phẩm trên firebase	25
Hình 13: Danh mục sản phẩm trong ứng dụng	26
Hình 14: Danh sách sản phẩm	27
Hình 15: Mã minh họa hiển thị danh sách sản phẩm	28
Hình 16: RecyclerView hiển thị danh sách các sản phẩm	29
Hình 17: Mục giỏ hàng trên firebase	29
Hình 18: Giao diện thêm hàng và giỏ hàng	30
Hình 19: Mã minh họa cho chức năng tạo đơn hàng	31
Hình 20: Mã minh họa xóa giỏ hàng	32
Hình 21: Mục đơn hàng trên firebase	32
Hình 22: Mã minh họa cho chức năng xem lịch sử mua hàng	33
Hình 23: Giao diện xem lịch sử đơn hàng trên ứng dụng	34
Hình 24: Giao diện người dùng	35
Hình 25: Mã minh họa hiển thị sản phẩm	36
Hình 26: Giao diện tài khoản trong ứng dụng	37
Hình 27: Giao diện thay đổi mật khẩu	38
Hình 28: Mã minh họa cho chức năng đổi mật khẩu	39

Hình 29: Giao diện sửa đổi thông tin	40
Hình 30: Mã minh họa cho chức năng đổi thông tin tài khoản	41
Hình 31: Giao diện lọc sản phẩm theo giá	42
Hình 32: Mã minh họa chức năng tìm kiếm sản phẩm	43
Hình 33: Giao diện quản lý sản phẩm yêu thích	45
Hình 34: Sản phẩm yêu thích	45
Hình 35: Mã minh họa chức năng lấy sản phẩm yêu thích	46

CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Trong những năm gần đây, nhu cầu mua sắm trực tuyến tại Việt Nam không ngừng gia tăng, đặc biệt là trong lĩnh vực thực phẩm và nông sản. Người tiêu dùng ngày càng quan tâm hơn đến chất lượng, độ an toàn và nguồn gốc xuất xứ của sản phẩm, trong khi đó thị trường lại đang thiếu những kênh phân phối trực tuyến chuyên biệt, uy tín dành riêng cho mặt hàng nông sản sạch.

Song song với đó, sự phát triển mạnh mẽ của các thiết bị di động thông minh và nền tảng Android đã tạo ra cơ hội lớn để xây dựng những ứng dụng tiện lợi, thân thiện với người dùng, giúp kết nối trực tiếp giữa nhà cung cấp và người tiêu dùng mà không cần qua trung gian.

Với mong muốn góp phần giải quyết bài toán tiếp cận nông sản chất lượng cho người tiêu dùng, đồng thời hỗ trợ nhà nông, hợp tác xã hay các doanh nghiệp nhỏ có thêm kênh phân phối hiệu quả, nhóm chúng tôi lựa chọn xây dựng ứng dụng bán nông sản trên nền tảng Java. Ứng dụng không chỉ mang tính thực tiễn mà còn phù hợp với xu hướng chuyển đổi số trong lĩnh vực nông nghiệp hiện nay.

1.2 Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng một ứng dụng Android hoàn chỉnh cho phép người dùng thực hiện các thao tác mua bán nông sản một cách thuận tiện, nhanh chóng và hiệu quả. Ứng dụng hướng đến việc hỗ trợ người tiêu dùng tiếp cận dễ dàng với nguồn nông sản chất lượng, đồng thời giúp người bán – bao gồm các hộ nông dân, hợp tác xã và doanh nghiệp nhỏ – có thêm một kênh phân phối trực tuyến hiện đại và thiết thực.

Bên cạnh mục tiêu tổng thể, đề tài còn đặt ra các mục tiêu cụ thể như sau:

- + Xây dựng chức năng quản lý, hiển thị và tìm kiếm sản phẩm để người dùng có thể dễ dàng duyệt và lựa chọn các mặt hàng nông sản phù hợp với nhu cầu.
- + Phát triển các tính năng tiện ích như giỏ hàng, danh sách yêu thích và chức năng đặt hàng, nhằm nâng cao trải nghiệm mua sắm cho người dùng.
- + Tích hợp hệ thống thanh toán bao gồm hình thức thanh toán khi nhận hàng (COD) và mô phỏng quy trình thanh toán online, đảm bảo tính linh hoạt cho người mua.
- + Xây dựng hệ thống quản lý tài khoản người dùng và đơn hàng giúp cá nhân hóa trải nghiệm sử dụng, đồng thời giúp người bán theo dõi hoạt động kinh doanh một cách hiệu quả.
- + Thiết kế giao diện ứng dụng theo hướng trực quan, thân thiện, dễ sử dụng với mọi đối tượng người dùng, kể cả người lớn tuổi hay người ít tiếp xúc với công nghệ.

1.3 Công nghệ sử dụng

Trong quá trình xây dựng ứng dụng bán nông sản, nhóm đã lựa chọn các công nghệ và công cụ phù hợp nhằm đảm bảo tính ổn định, dễ bảo trì và thân thiện với người dùng. Cụ thể như sau:

- + **Ngôn ngữ lập trình:** Java được sử dụng làm ngôn ngữ chính để phát triển ứng dụng. Đây là một trong những ngôn ngữ phổ biến, mạnh mẽ và được hỗ trợ tốt trên nền tảng Android.
- + **Môi trường phát triển (IDE):** Android Studio – công cụ chính thức do Google phát triển cho lập trình Android – được sử dụng để xây dựng và triển khai ứng dụng. Android Studio cung cấp bộ công cụ hỗ trợ mạnh mẽ trong việc thiết kế giao diện, kiểm thử và quản lý dự án.
- + **Cơ sở dữ liệu (Backend):** Firebase Realtime Database. Đây là một hệ quản trị cơ sở dữ liệu thời gian thực do Google cung cấp, cho phép lưu trữ và đồng bộ dữ liệu giữa các client một cách nhanh chóng. Firebase rất phù hợp với các ứng dụng có yêu cầu cập nhật dữ liệu tức thời như theo dõi trạng thái đơn hàng, hệ thống chat, hay thông báo đẩy. Ngoài ra, Firebase còn hỗ trợ tích hợp đơn giản với Android, tiết kiệm thời gian triển khai và tối ưu hiệu suất ứng dụng.
- + **Các thư viện hỗ trợ:** Trong quá trình phát triển, nhóm cũng tích hợp một số thư viện ngoài nhằm nâng cao hiệu quả và trải nghiệm người dùng. Ví dụ như thư viện Glide hoặc Picasso được sử dụng để tải và hiển thị hình ảnh sản phẩm nhanh chóng và tối ưu bộ nhớ.

1.4 Cấu trúc báo cáo

Báo cáo được chia thành 4 chương chính như sau:

- + Chương 1 – Tổng quan: Trình bày lý do chọn đề tài, mục tiêu, công nghệ sử dụng và cấu trúc tổng thể của báo cáo.
- + Chương 2 – Cơ sở lý thuyết: Trình bày kiến thức nền tảng liên quan đến lập trình Android, cơ sở dữ liệu Firebase và các công nghệ hỗ trợ trong quá trình xây dựng ứng dụng.
- + Chương 3 – Thực nghiệm: Mô tả quá trình phân tích, thiết kế và hiện thực ứng dụng. Bao gồm các chức năng chính, giao diện và kết quả triển khai.
- + Chương 4 – Tổng kết: Đưa ra nhận xét, đánh giá kết quả đạt được, khó khăn trong quá trình thực hiện và hướng phát triển trong tương lai.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Cơ sở lý thuyết

Trong quá trình phát triển ứng dụng bán nông sản trên nền tảng Android, việc lựa chọn công nghệ phù hợp đóng vai trò rất quan trọng để đảm bảo hiệu quả triển khai và chất lượng sản phẩm. Phần này trình bày hai công nghệ chính được sử dụng: Android Studio và Firebase Realtime Database.

2.1.1 Android studio

Android Studio là môi trường phát triển tích hợp (IDE) chính thức do Google cung cấp dành riêng cho lập trình Android. Android Studio hỗ trợ lập trình viên trong toàn bộ quy trình phát triển ứng dụng, từ việc viết mã, thiết kế giao diện, gỡ lỗi, biên dịch cho đến triển khai ứng dụng. Môi trường này tích hợp sẵn các công cụ như trình giả lập thiết bị (Emulator), trình quản lý gói SDK, hệ thống build Gradle,... giúp quá trình phát triển ứng dụng trở nên hiệu quả và chuyên nghiệp hơn.

Để phát triển ứng dụng đầu tiên của mình, bạn cần tải xuống Android Studio cho nền tảng ưa thích (Windows, Mac OS X hoặc Linux) từ trang web dành cho nhà phát triển Android. Android Studio có thể hỗ trợ phát triển và thử nghiệm ứng dụng của bạn trên thiết bị thực hoặc trình giả lập [1].

2.1.2 Firebase Realtime Database

Firebase là một nền tảng giúp phát triển các ứng dụng di động trong web. Bên cạnh đó, Firebase còn được hiểu là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây cloud với hệ thống máy chủ mạnh mẽ của Google [2].

Firebase Realtime Database là một cơ sở dữ liệu NoSQL được lưu trữ trên nền tảng đám mây (cloud-hosted), cho phép đồng bộ hóa dữ liệu theo thời gian thực giữa các client. Dữ liệu trong Firebase được lưu dưới dạng cây JSON, rất phù hợp cho các ứng dụng di động cần phản hồi nhanh với sự thay đổi dữ liệu như ứng dụng thương mại điện tử, chat, theo dõi đơn hàng,... Một trong những ưu điểm nổi bật của Firebase là khả năng tích hợp dễ dàng với Android thông qua SDK chính thức.

Firebase hỗ trợ đầy đủ các thao tác CRUD cơ bản trong quản lý dữ liệu, bao gồm:

- + Thêm dữ liệu (Create): Cho phép tạo mới các nút (node) dữ liệu vào cơ sở dữ liệu bằng phương thức setValue() hoặc push().
- + Đọc dữ liệu (Retrieve): Sử dụng các listener như ValueEventListener để truy vấn và lắng nghe sự thay đổi dữ liệu theo thời gian thực.

- + Cập nhật dữ liệu (Update): Firebase cho phép cập nhật từng phần của dữ liệu mà không cần ghi đè toàn bộ, thông qua phương thức `updateChildren()`.
- + Xóa dữ liệu (Delete): Có thể xóa dữ liệu bằng cách đặt giá trị của nút đó thành `null`.

Những đặc điểm trên giúp Firebase trở thành một giải pháp phù hợp cho việc phát triển các ứng dụng có tính tương tác cao và yêu cầu cập nhật tức thời.

2.2 Phân tích và Thiết kế hệ thống

Trước khi tiến hành xây dựng ứng dụng, việc phân tích và thiết kế hệ thống là bước quan trọng nhằm đảm bảo việc triển khai được thực hiện có định hướng và hiệu quả. Trong ứng dụng bán nông sản này, các chức năng được thiết kế xoay quanh người dùng với tư cách là tác nhân chính. Để mô hình hóa mối quan hệ giữa người dùng và hệ thống, biểu đồ Use Case được sử dụng để xác định các hành vi chức năng quan trọng.

Các chức năng chính mà người dùng có thể thực hiện trong hệ thống bao gồm:

- + Quản lý tài khoản (đăng ký, đăng nhập, đăng xuất, chỉnh sửa thông tin).
- + Quản lý sản phẩm (xem danh sách sản phẩm, xem chi tiết từng sản phẩm, tìm kiếm theo tên, lọc theo danh mục).
- + Quản lý giỏ hàng (thêm sản phẩm vào giỏ, xóa sản phẩm khỏi giỏ, cập nhật số lượng)
- + Quản lý đơn hàng (đặt hàng, xem lịch sử đơn hàng, hủy đơn hàng).
- + Tương tác như đánh giá sản phẩm hoặc thêm sản phẩm vào danh sách yêu thích.

Các ca sử dụng này đều được thể hiện trong biểu đồ Use Case tổng quan, trong đó tác nhân chính là người dùng, tương tác với từng ca sử dụng thông qua giao diện ứng dụng Android.

Về mặt lưu trữ dữ liệu, hệ thống sử dụng Firebase Realtime Database – một cơ sở dữ liệu NoSQL được lưu trữ trên nền tảng đám mây. Firebase có khả năng đồng bộ hóa dữ liệu theo thời gian thực giữa các client, cho phép người dùng nhận các cập nhật ngay lập tức mà không cần làm mới lại giao diện. Điều này đặc biệt phù hợp với ứng dụng thương mại điện tử nơi trạng thái đơn hàng, giỏ hàng hoặc đánh giá sản phẩm cần được cập nhật ngay tức thì.

Firebase tổ chức dữ liệu dưới dạng một cây JSON, với các nhánh chính như `users`, `products`, `cart`, `orders`, `favorites` và `reviews`. Việc thiết kế cấu trúc dữ liệu cần đảm bảo rõ ràng, phân tách tốt các loại thông tin, đồng thời thuận tiện cho việc thao tác CRUD (Create, Read, Update, Delete). Dưới đây là mô tả rõ ràng về cấu trúc dữ liệu:

Các danh mục sản phẩm (Categories) chứa thông tin như mã danh mục, tên, hình ảnh, mùa vụ liên quan và danh sách sản phẩm thuộc danh mục đó (CategoriesItems). Mỗi sản phẩm có các thuộc tính: mô tả, mã sản phẩm, hình ảnh, tồn kho, tên, giá, số lượng trong giỏ hàng, loại, và đơn vị tính.











Mã giảm giá (Coupons) được lưu kèm thông tin loại giảm giá (phần trăm hoặc cố định), mô tả, giá trị giảm, thời gian hiệu lực và mã sản phẩm áp dụng. Dữ liệu yêu thích (LikedItems) được lưu theo từng người dùng với danh sách các sản phẩm yêu thích tương ứng.

Đơn hàng (OrderBills) ghi lại thông tin như danh sách sản phẩm đã đặt, mã đơn hàng, thời gian đặt, trạng thái (PENDING, PAID, Success), tổng giá trị và ID người dùng. Đánh giá sản phẩm (Reviews) được tổ chức theo từng danh mục và sản phẩm, mỗi đánh giá bao gồm bình luận, điểm số, thời gian, ID và thông tin người đánh giá.

Giỏ hàng (ShoppingCarts) được tổ chức theo userId, bao gồm danh sách sản phẩm và tổng giá trị giỏ. Sản phẩm đã bán (SoldItems) được lưu theo từng ngày, mỗi ngày chứa danh sách sản phẩm với mã sản phẩm và số lượng đã bán.

Thông tin người dùng (Users) bao gồm email, họ tên, mật khẩu, số điện thoại, địa chỉ, ảnh đại diện, cùng với dữ liệu liên quan như sản phẩm yêu thích, giỏ hàng và danh sách đơn hàng. Ngoài ra, dữ liệu đơn hàng và người dùng còn có thể được tổ chức lại ở các nút riêng biệt (OrderBills, Users) để phục vụ cho việc truy xuất nhanh hoặc xuất dữ liệu riêng theo nhu cầu của ứng dụng.

Cách tổ chức này đảm bảo rõ ràng, phân tách dữ liệu theo vai trò và quan hệ, giúp dễ dàng thao tác CRUD, truy vấn, hiển thị và mở rộng tính năng trong tương lai.

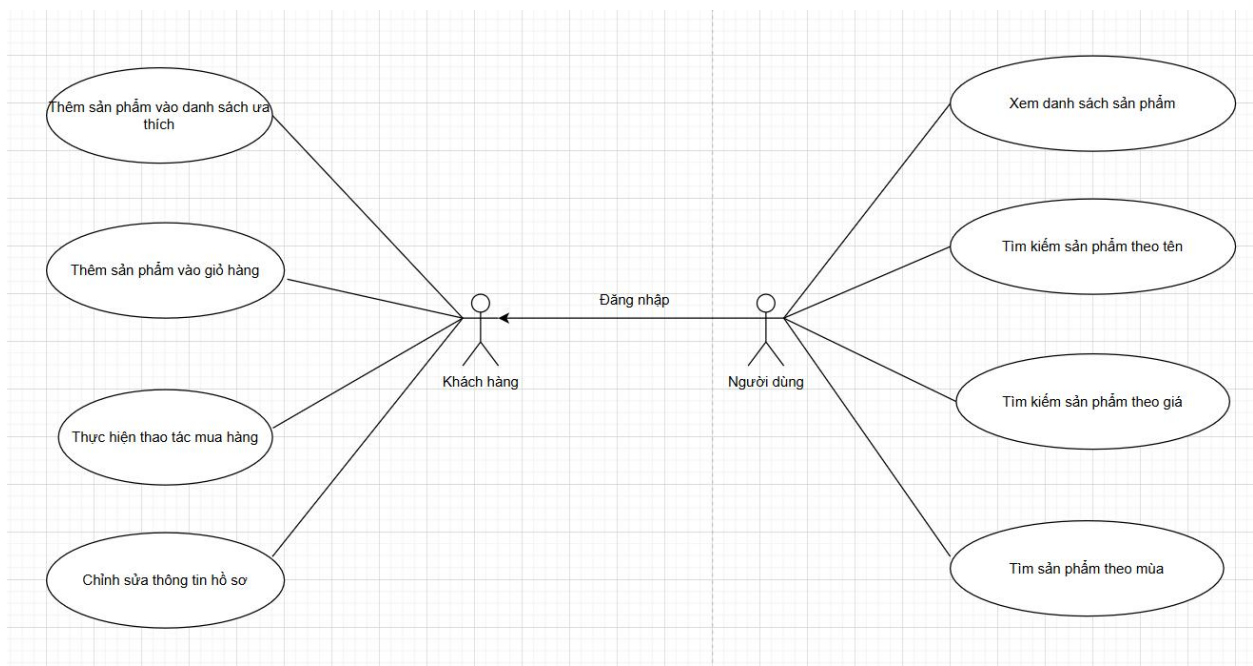
 Data	
 Categories	8 fields >
 CategoriesItems	7 fields >
 Coupons	17 fields >
 LikedItems	1 field >
 OrderBills	8 fields >
 Reviews	4 fields >
 ShoppingCarts	1 field >
 SoldItems	11 fields >
 Users	7 fields >

Hình 1: Cấu trúc JSON



Hình 2: Cấu trúc của Categories

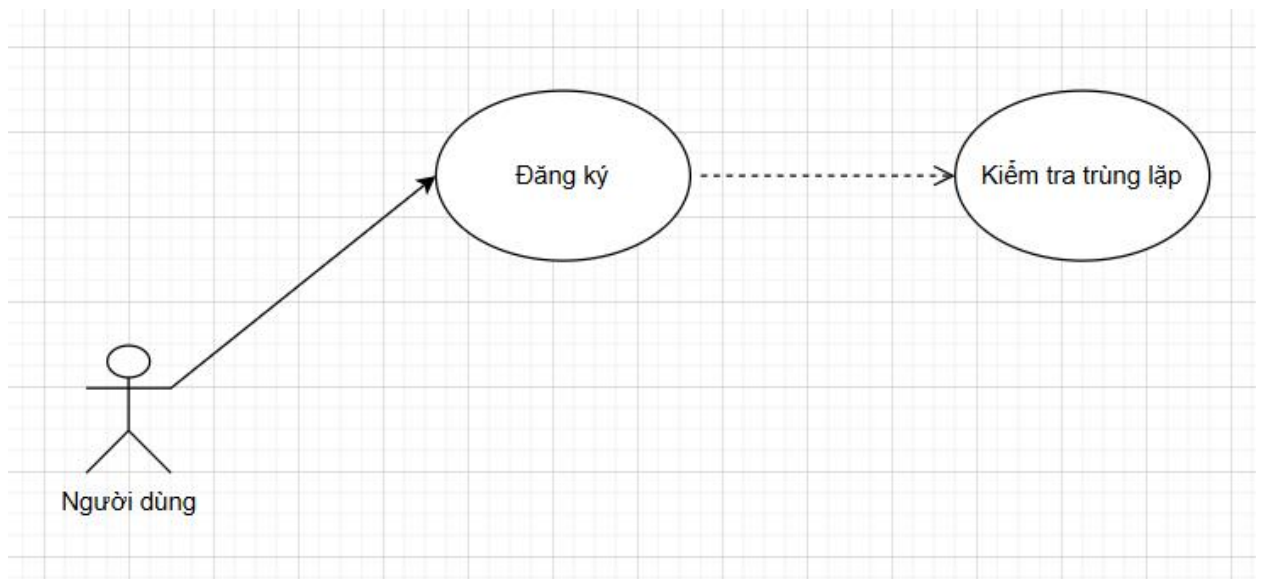
Sơ đồ usecase tổng quát



Hình 3: Usecase tổng quát

Bất kỳ người dùng nào cũng có quyền vào xem danh sách sản phẩm. Tuy nhiên để thực hiện chức năng mua hàng thì phải đăng nhập tài khoản có role khách hàng.

Sơ đồ usecase đăng ký tài khoản

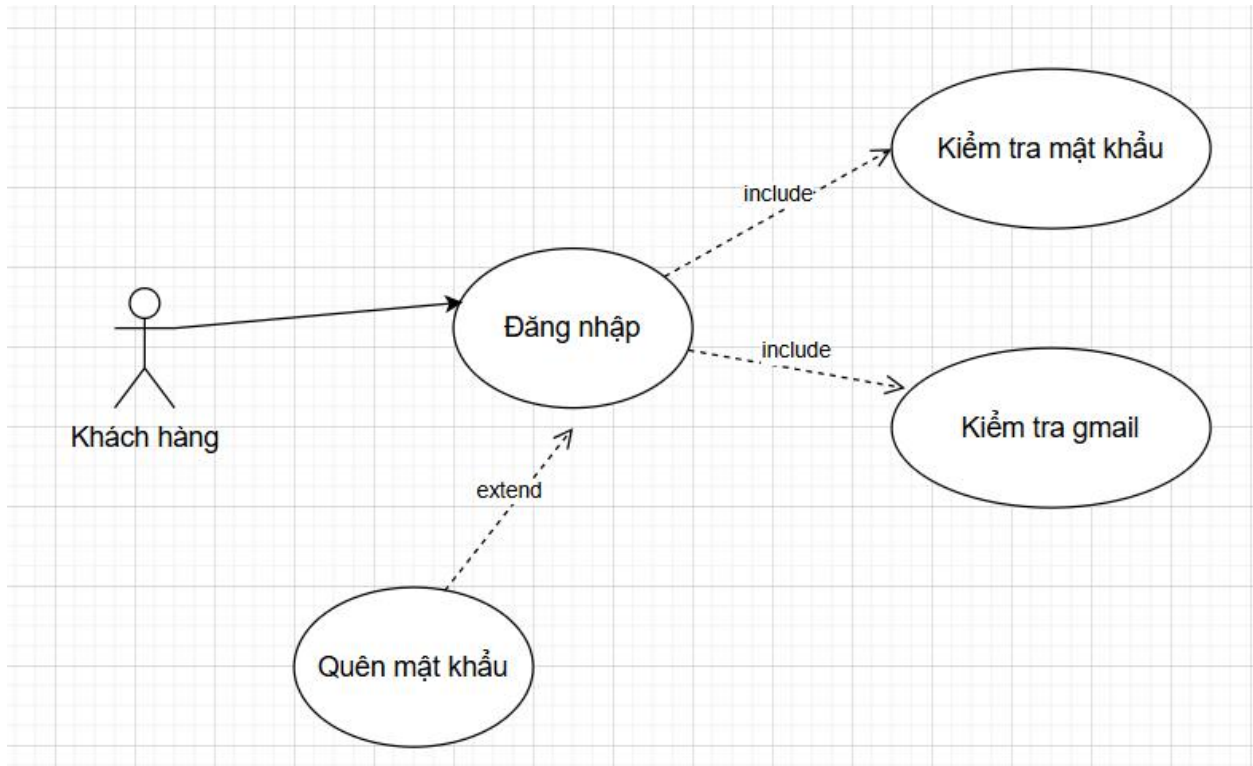


Hình 4: Usecase đăng ký tài khoản

Đặc tả usecase

Tên usecase	Đăng ký tài khoản
Phạm vi	Hệ thống đăng ký tài khoản
Tác nhân	Người dùng, khách hàng
Mô tả	Người dùng đăng ký một tài khoản để sử dụng nhiều chức năng nâng cao như: giỏ hàng, mua sản phẩm, nhận voucher khuyến mãi.
Sự kiện kích hoạt	Người dùng click vào nút đăng ký tài khoản
Kết quả	Điều hướng đến trang đăng ký tài khoản

Sơ đồ usecase đăng nhập tài khoản

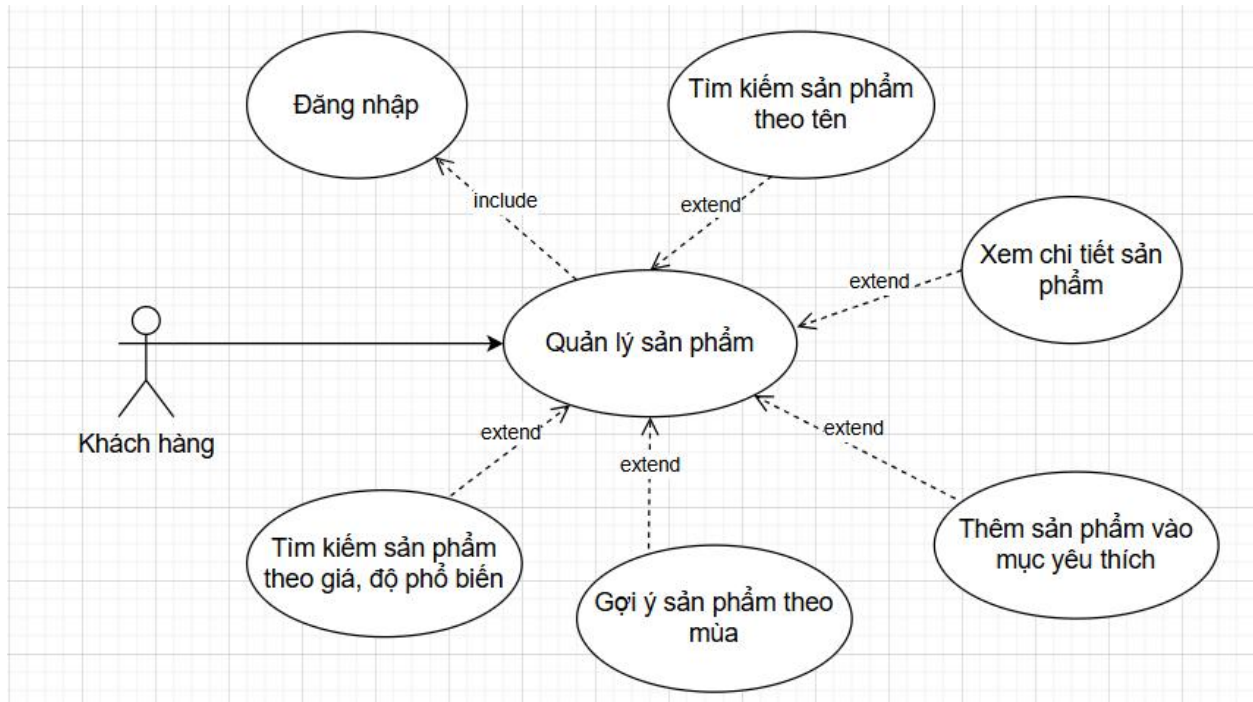


Hình 5: Usecase đăng nhập

Đặc tả usecase

Tên usecase	Đăng nhập tài khoản người dùng
Phạm vi	Hệ thống đăng nhập tài khoản khách hàng
Tác nhân	Người dùng (Khách hàng)
Mô tả	Người dùng sử dụng gmail và mật khẩu để đăng nhập vào hệ thống.
Sự kiện kích hoạt	Người dùng click vào nút “Đăng nhập”
Kết quả	<ul style="list-style-type: none"> Nếu người dùng đăng nhập thành công, hệ thống sẽ điều hướng đến trang chủ của app mua hàng Nếu thông tin đăng nhập sai, hệ thống sẽ thông báo lỗi, yêu cầu người dùng nhập lại

Sơ đồ usecase quản lý sản phẩm

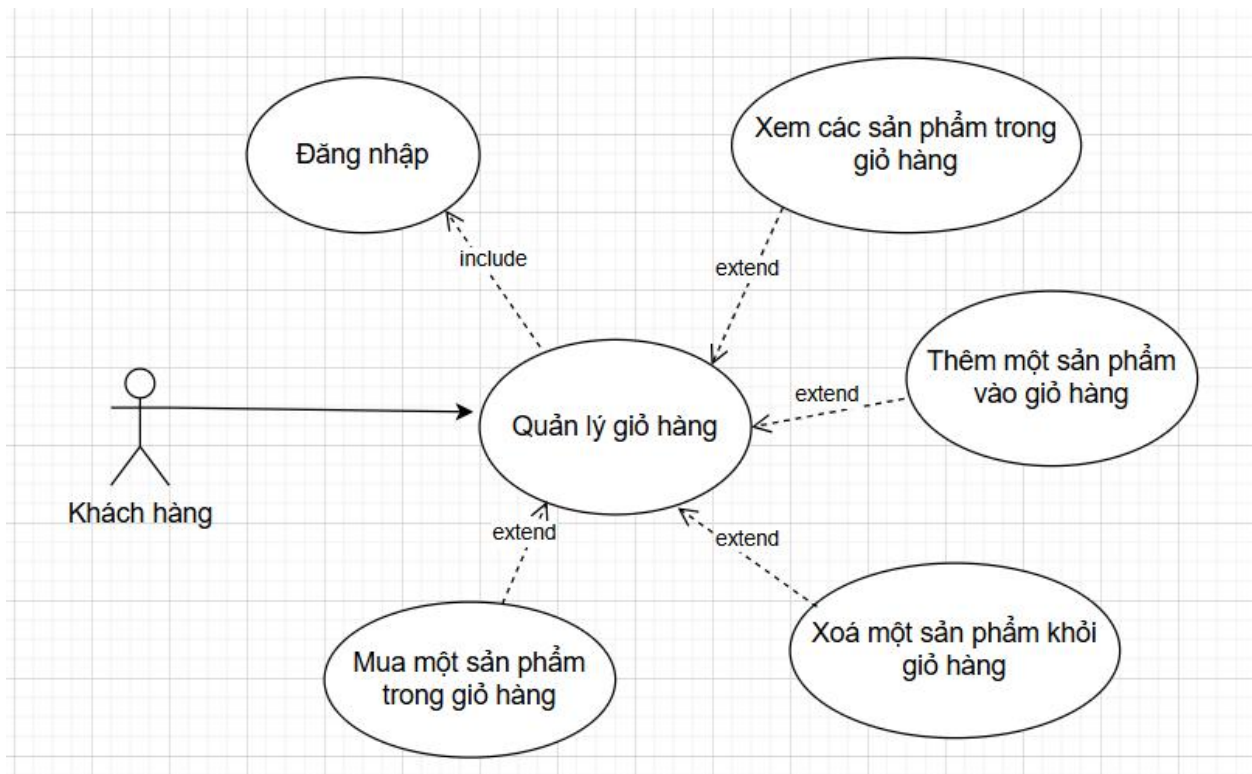


Hình 6: Usecase quản lý sản phẩm

Đặc tả usecase

Tên usecase	Quản lý, tìm kiếm sản phẩm
Phạm vi	Hệ thống quản lý sản phẩm phía người dùng
Tác nhân	Khách hàng
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công
Mô tả	Người dùng tìm kiếm các sản phẩm theo tên, thể loại, độ phổ biến, theo mùa.
Sự kiện kích hoạt	Người dùng click vào nút “Search”
Kết quả	<ul style="list-style-type: none">Nếu có sản phẩm người dùng cần tìm thì sản phẩm sẽ hiển thị ra dưới dạng list.Nếu không có sản phẩm hợp lệ, hệ thống sẽ báo “Không có sản phẩm cần tìm”

Sơ đồ usecase quản lý giỏ hàng

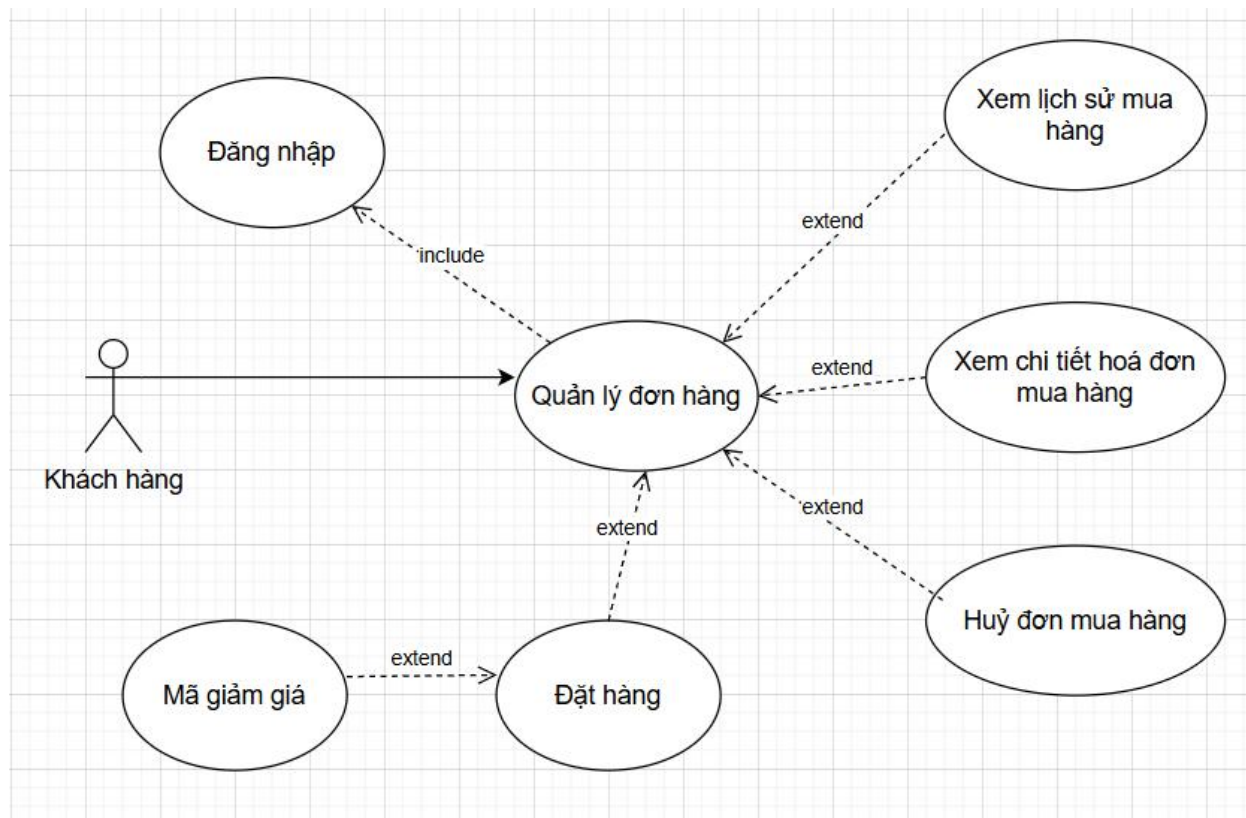


Hình 7: Usecase quản lý giỏ hàng

Đặc tả usecase

Tên usecase	Quản lý giỏ hàng
Phạm vi	Hệ thống quản lý giỏ hàng của khách hàng
Tác nhân	Khách hàng
Mô tả	Người dùng sẽ xem, thêm, xoá một sản phẩm bên trong giỏ hàng.
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công.
Sự kiện kích hoạt	Người dùng click vào biểu tượng giỏ hàng bên thanh điều hướng.
Kết quả	Chuyển người dùng đến trang giỏ hàng, hiển thị các sản phẩm bên trong giỏ hàng

Sơ đồ usecase quản lý đơn hàng



Hình 8: Usecase quản lý đơn hàng

Đặc tả usecase

Tên usecase	Quản lý đơn hàng
Phạm vi	Hệ thống quản lý đơn hàng của khách hàng
Tác nhân	Khách hàng
Mô tả	Người dùng có thể thực hiện thao tác mua hàng, thanh toán, xem trạng thái hoá đơn, huỷ đơn hàng.
Điều kiện tiên quyết	Khách hàng đã đăng nhập thành công và chọn ít nhất một sản phẩm để thanh toán.
Sự kiện kích hoạt	Người dùng click nút “Thanh toán”
Kết quả	<ul style="list-style-type: none"> Người sản phẩm còn hàng và người dùng cung cấp đủ thông tin, hệ thống sẽ xác nhận và thông báo trạng thái đơn ở trạng thái chờ người bán phê duyệt. Nếu phê duyệt thành công, hoá đơn sẽ bước sang trạng

CHƯƠNG 3 THỰC NGHIỆM

Chương này trình bày chi tiết quá trình xây dựng ứng dụng bán nông sản trên nền tảng Android, sử dụng ngôn ngữ Java kết hợp Firebase Realtime Database. Các bước từ cài đặt môi trường, kết nối cơ sở dữ liệu đến lập trình giao diện và xử lý logic đều được thực hiện tuần tự và có tổ chức.

3.1 Cài đặt môi trường và kết nối Firebase

Trước hết, môi trường phát triển Android Studio được cài đặt và thiết lập. Dự án Android mới được tạo với package name riêng để nhận dạng ứng dụng. Tiếp theo, tiến hành kết nối dự án với Firebase thông qua các bước cơ bản sau:

- + Tạo một dự án mới trên Firebase Console.
- + Thêm ứng dụng Android vào dự án, cung cấp package name khớp với dự án trong Android Studio.
- + Tải file google-services.json và đặt vào thư mục app của dự án. Thêm các dependency Firebase vào file build.gradle cấp project và cấp module (app), bao gồm các thư viện như Firebase Database, Firebase Auth, v.v.
- + Cấu hình google-services plugin để kích hoạt các dịch vụ của Firebase.
- + Thiết lập Rule cho Firebase Realtime Database ở chế độ thử nghiệm (test mode) để cho phép đọc và ghi dữ liệu trong quá trình phát triển.

3.2 Xây dựng các chức năng chính

Ứng dụng được xây dựng xoay quanh các chức năng cốt lõi như đăng ký, đăng nhập, quản lý sản phẩm, giỏ hàng, đơn hàng và tương tác người dùng. Trong phần này, các chức năng chính được mô tả cùng với ảnh minh họa màn hình thực tế và đoạn mã tiêu biểu để làm rõ quy trình xử lý.

Chức năng đăng ký và đăng nhập

Được triển khai thông qua Firebase Authentication. Đây là dịch vụ xác thực người dùng nhanh chóng và an toàn, hỗ trợ nhiều phương thức như email/password, Google, Facebook, v.v. Ứng dụng sử dụng phương thức phổ biến nhất là email và mật khẩu. Khi người dùng đăng ký, ứng dụng kiểm tra định dạng email và độ dài mật khẩu. Sau đó gọi phương thức `createUserWithEmailAndPassword()` của Firebase để tạo tài khoản.

AgrStore

Đăng ký

First name

Last name

Email

Password

Confirm password

+972

Phone number

SIGN UP

Already have an account?

Log in

Hình 9: Giao diện đăng ký

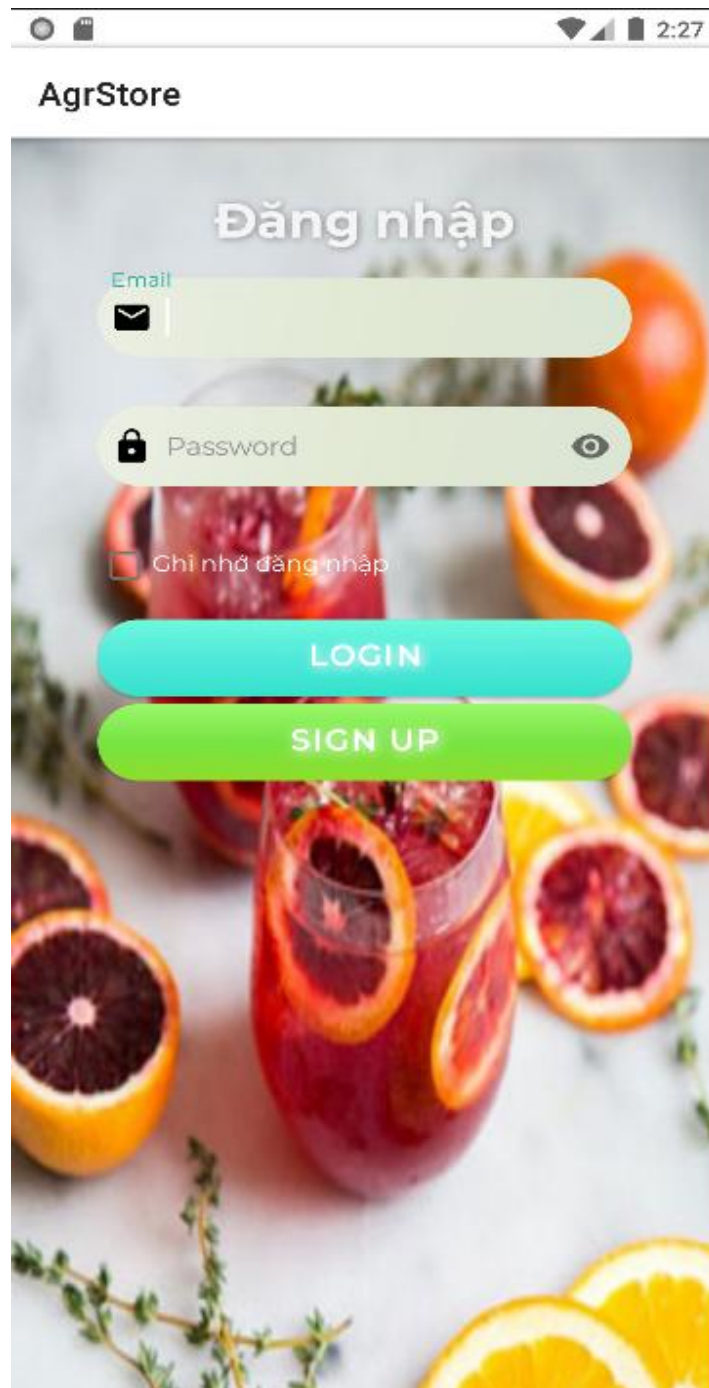
Nếu thành công, tài khoản được lưu trữ trên Firebase Auth và có thể đồng bộ hóa với các dữ liệu khác như giỏ hàng, đơn hàng.

```
1 usage
public void Register(User i_NewUser, AuthCallback callback) {
    m_Auth.createUserWithEmailAndPassword(i_NewUser.getEmail(), i_NewUser.getPassword())
        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    m_FirebaseUser = m_Auth.getCurrentUser();
                    String userId = m_FirebaseUser.getId();

                    m_Ref.child(userId).setValue(i_NewUser).addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if (task.isSuccessful()) {
                                callback.onSuccess();
                            } else {
                                callback.onFailure(task.getException());
                            }
                        }
                    });
                } else {
                    callback.onFailure(task.getException());
                }
            }
        });
}
```

Hình 10: Mã minh họa cho chức năng đăng ký

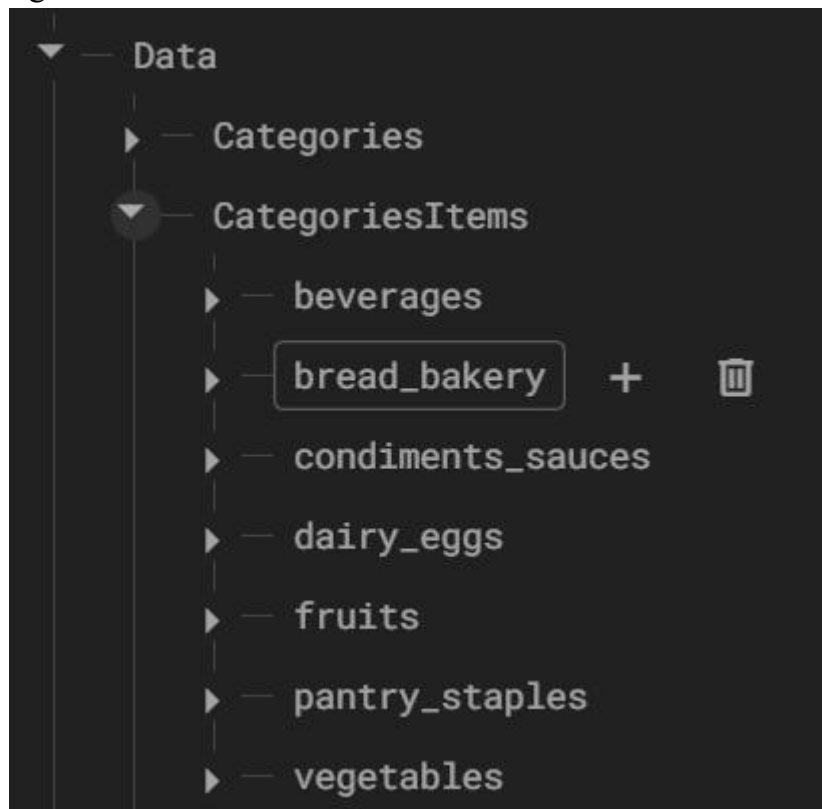
Ngược lại, khi đăng nhập, người dùng nhập email và mật khẩu đã đăng ký trước đó. Ứng dụng sử dụng phương thức `signInWithEmailAndPassword()` để xác thực. Nếu hợp lệ, hệ thống cho phép truy cập và chuyển sang giao diện chính của ứng dụng.



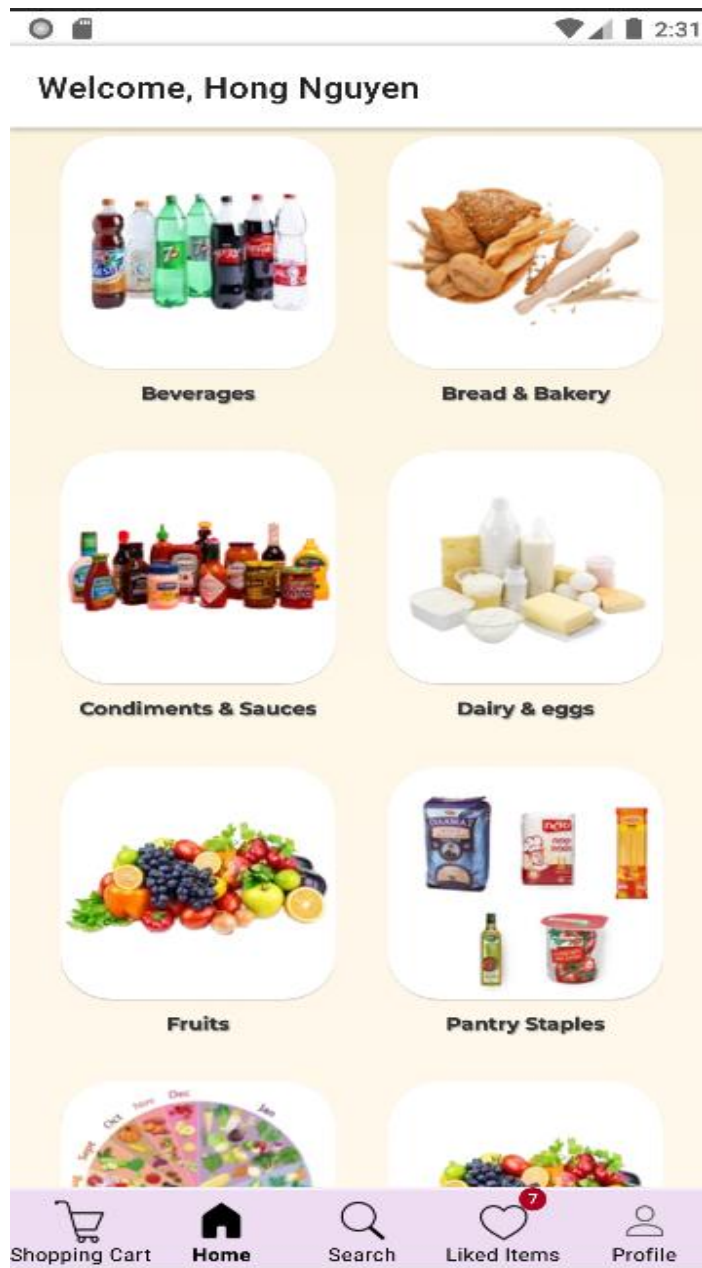
Hình 11: Giao diện đăng nhập

Chức năng hiển thị danh sách sản phẩm

Chức năng này cho phép người dùng xem danh sách các sản phẩm hiện có, được tổ chức theo từng danh mục. Dữ liệu sản phẩm được lấy từ Firebase Realtime Database, cụ thể là từ nút Data/CategoriesItems.

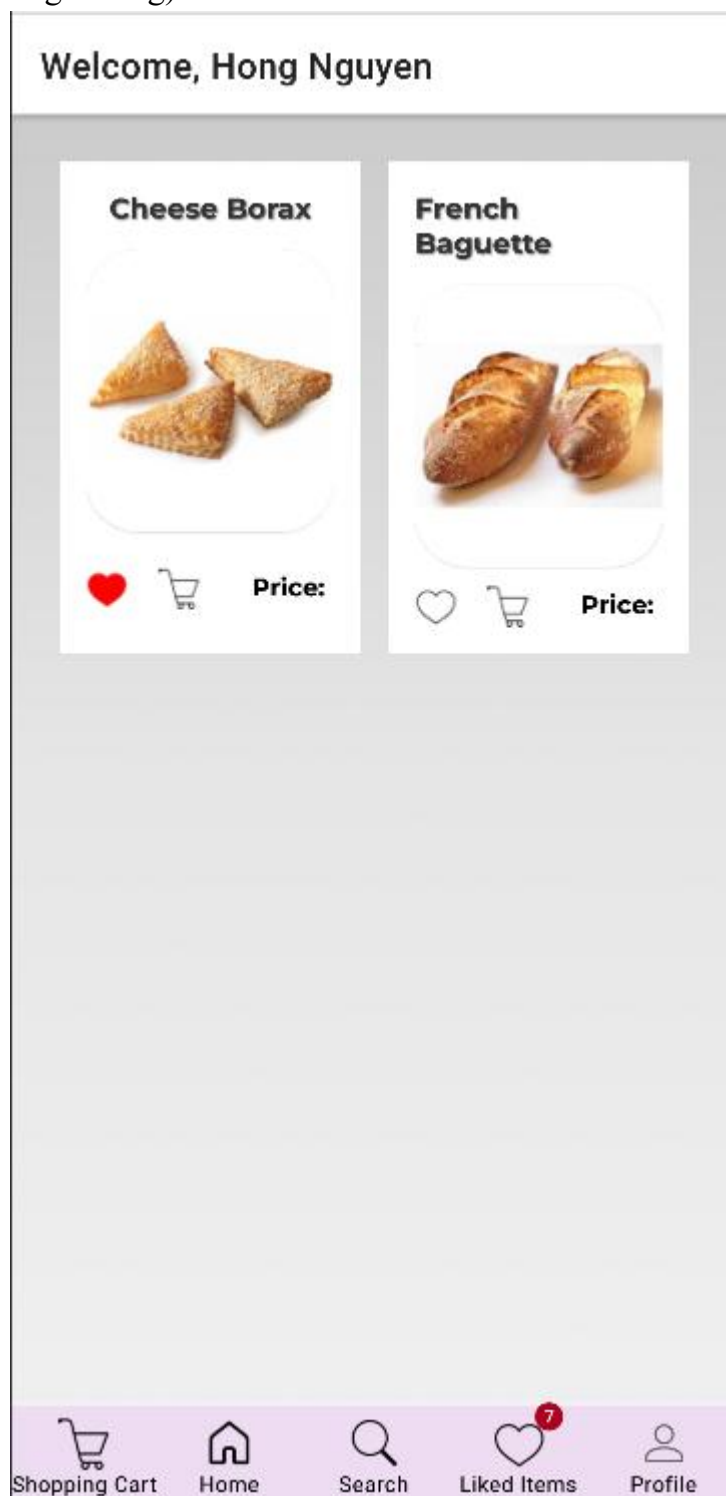


Hình 12: Danh mục sản phẩm trên firebase



Hình 13: Danh mục sản phẩm trong ứng dụng

Mỗi danh mục chứa một danh sách sản phẩm, mỗi sản phẩm có các thuộc tính như: id, name, image, price, inventory, description, unit, type, và quantity (mặc định là 0 nếu chưa được thêm vào giỏ hàng).



Hình 14: Danh sách sản phẩm

Để hiển thị danh sách sản phẩm, ứng dụng sử dụng ValueEventListener để lắng nghe dữ liệu từ Firebase. Khi có thay đổi trong cơ sở dữ liệu (ví dụ: thêm sản phẩm mới, cập nhật tồn kho), danh sách trên giao diện sẽ được cập nhật tự động. Cách tiếp cận này đảm bảo tính real-time, mang lại trải nghiệm mượt mà cho người dùng.

```
m_Ref.addListenerForSingleValueEvent(new ValueEventListener() {  
    @Override  
    public void onDataChange(@NonNull DataSnapshot snapshot) {  
        for (DataSnapshot itemSnapshot : snapshot.getChildren()) {  
            String itemId = itemSnapshot.child( path: "Id").getValue(String.class);  
            String itemType = itemSnapshot.child( path: "Type").getValue(String.class);  
            String itemName = itemSnapshot.child( path: "Name").getValue(String.class);  
            Double itemPrice = itemSnapshot.child( path: "Price").getValue(Double.class);  
            String itemImage = itemSnapshot.child( path: "Image").getValue(String.class);  
            String itemDescription = itemSnapshot.child( path: "Description").getValue(String.class);  
            String itemUnit = itemSnapshot.child( path: "Unit").getValue(String.class);  
  
            int itemQuantity = 0;  
  
            Item item = new Item(itemId, itemType, itemName, itemPrice, itemQuantity, itemImage, it  
  
            // Thiết lập các trường mới nếu có dữ liệu  
            if (itemDescription != null) {  
                item.setDescription(itemDescription);  
            }  
            if (itemUnit != null) {  
                item.setUnit(itemUnit);  
            }  
            m_CategoryItems.add(item);  
        }  
    }  
});
```

Hình 15: Mã minh họa hiển thị danh sách sản phẩm

Các sản phẩm được hiển thị thông qua RecyclerView, một thành phần giao diện mạnh mẽ và tối ưu hóa hiệu suất. Mỗi item hiển thị gồm hình ảnh, tên sản phẩm, giá và nút thao tác như “Thêm vào giỏ hàng”.

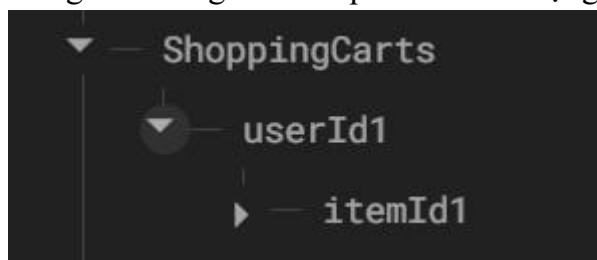
```
</LinearLayout>

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/categoryItemsRecyclerView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:clipToPadding="false"
    android:layout_margin="4dp"
    android:padding="15dp" />
</LinearLayout>
```

Hình 16: RecyclerView hiển thị danh sách các sản phẩm

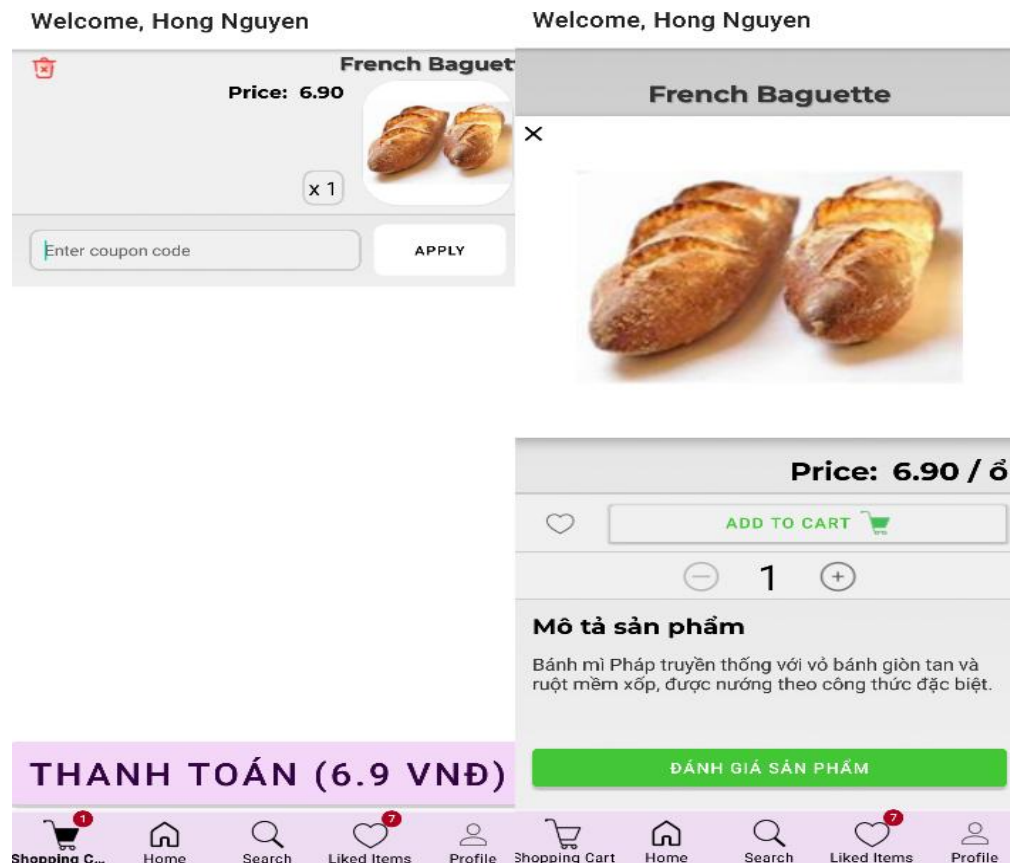
Chức năng giỏ hàng và đặt hàng

Giỏ hàng là thành phần trung gian quan trọng giữa thao tác chọn sản phẩm và tiến hành đặt hàng. Trong ứng dụng, dữ liệu giỏ hàng được lưu trong Firebase Realtime Database tại node ShoppingCarts/{userId}. Mỗi người dùng có một giỏ hàng riêng, lưu trữ danh sách sản phẩm đã chọn cùng các thông tin liên quan như số lượng và tổng giá trị.



Hình 17: Mục giỏ hàng trên firebase

Khi người dùng nhấn nút “Thêm vào giỏ hàng” trên mỗi sản phẩm, thông tin sản phẩm đó sẽ được ghi vào node `ShoppingCarts/{userId}`. Nếu sản phẩm đã tồn tại trong giỏ, hệ thống chỉ cập nhật số lượng thay vì thêm mới. Mỗi mục trong giỏ hàng bao gồm các trường: `id`, `name`, `image`, `price`, `quantity`, `unit`.



Hình 18: Giao diện thêm hàng và giỏ hàng

Quá trình đặt hàng diễn ra khi người dùng xác nhận thanh toán giỏ hàng. Hệ thống thực hiện hai bước chính:

- + Tạo đơn hàng mới: Một bản ghi đơn hàng được tạo trong node OrderBills (hoặc orders) với các thông tin như: danh sách sản phẩm, tổng giá trị đơn hàng, thời gian đặt hàng (timestamp), trạng thái đơn hàng (ví dụ: PENDING), và userId.

```
m_AddToCart.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // Lấy số lượng hiện tại từ TextView  
        int quantity = Integer.parseInt(m_Quantity.getText().toString());  
  
        // Thêm sản phẩm vào giỏ hàng với số lượng đã chọn  
        m_UserShoppingCart.AddToCart(m_CurrentItem, quantity);  
  
        // Nếu sản phẩm này cũng đang nằm trong danh sách yêu thích,  
        // cập nhật thông tin của nó trong danh sách yêu thích  
        if(m_UserLikedItemsList.containsKey(m_CurrentItem.getName()))  
        {  
            m_UserLikedItemsList.put(m_CurrentItem.getName(), m_CurrentItem);  
        }  
  
        // Cập nhật giỏ hàng và danh sách yêu thích trong MainActivity  
        m_HostedActivity.UpdateShoppingCart(m_UserShoppingCart);  
        m_HostedActivity.UpdateLikedItemsList(m_UserLikedItemsList);  
    }  
});
```

Hình 19: Mã minh họa cho chức năng tạo đơn hàng

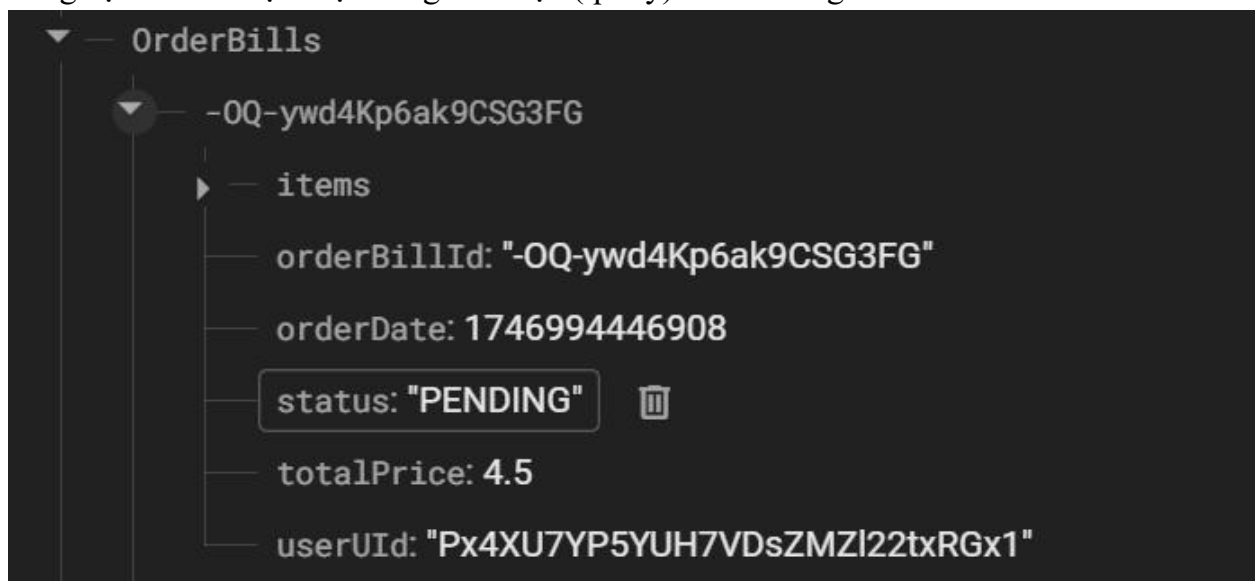
- + Xóa giỏ hàng: Sau khi đơn hàng được lưu thành công, toàn bộ node ShoppingCarts/{userId} sẽ bị xóa để đảm bảo giỏ hàng trống cho các lần mua tiếp theo.

```
1 usage
@Override
public void onRemoveItem(Item item) {
    m_UserShoppingCartAsList.remove(item);
    m_HostedActivity.RemoveItem(item);
    m_CurrentAction = CartAction.ITEM_REMOVED;
}
```

Hình 20: Mã minh họa xóa giỏ hàng

Chức năng quản lý đơn hàng

Chức năng quản lý đơn hàng cho phép người dùng theo dõi các đơn hàng mà họ đã đặt trước đó. Dữ liệu đơn hàng được lưu trữ tại node OrderBills trong Firebase Realtime Database. Mỗi đơn hàng là một bản ghi chứa các thông tin như: orderBillId, items, totalPrice, orderDate, status, và userId. Dữ liệu được tổ chức theo cấu trúc phẳng, không phân nhóm theo userId, vì vậy việc truy xuất danh sách đơn hàng của một người dùng cụ thể cần thực hiện bằng cách lọc (query) theo trường userId.



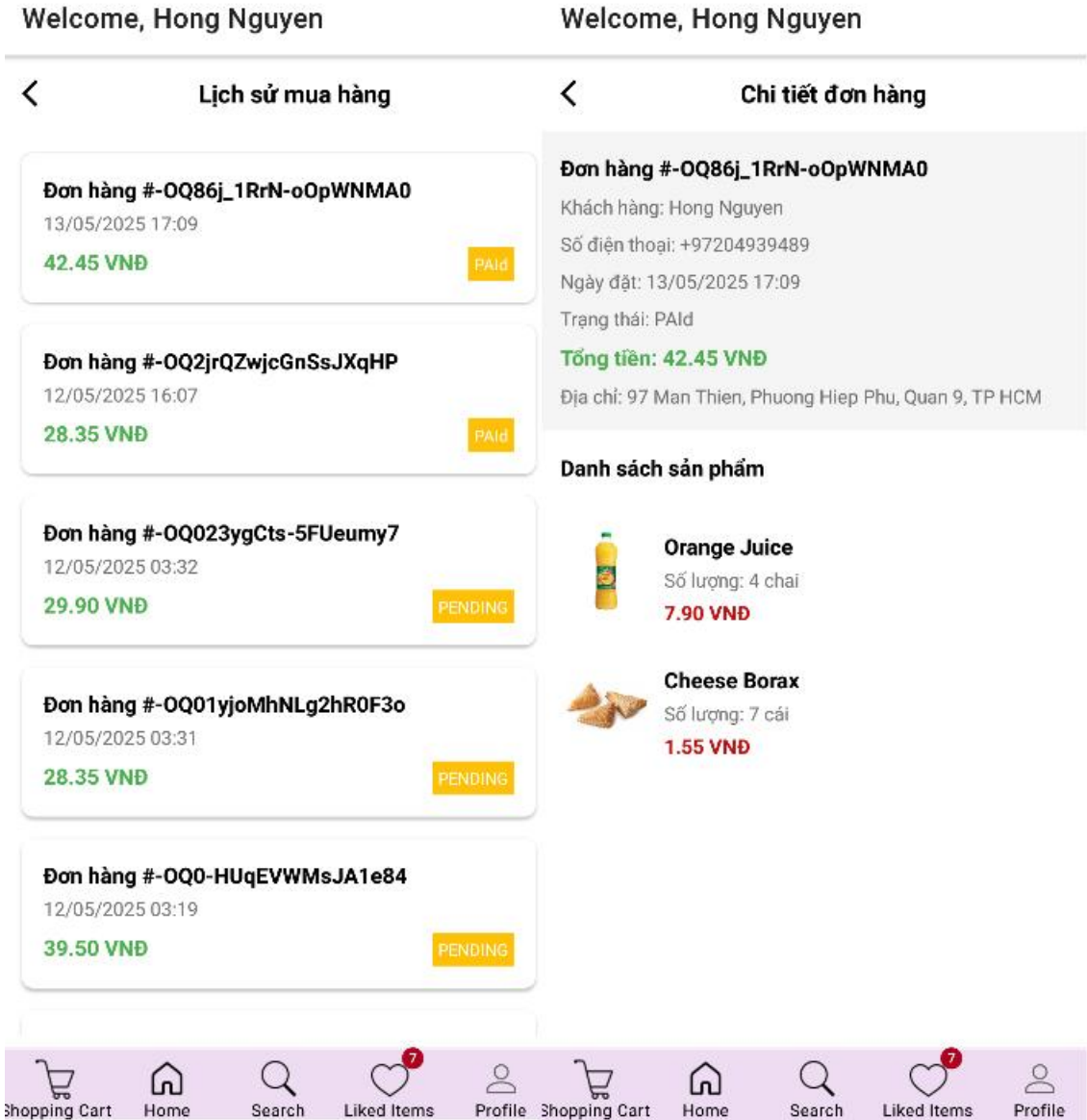
Hình 21: Mục đơn hàng trên firebase

Khi người dùng truy cập màn hình "Lịch sử mua hàng", ứng dụng sẽ thực hiện truy vấn với điều kiện `userId == currentUserId`. Firebase cung cấp phương thức `orderByChild()` kết hợp với `equalTo()` để thực hiện truy vấn có điều kiện:

```
private void loadOrderHistory() {  
    // Hiển thị ProgressBar  
    progressBar.setVisibility(View.VISIBLE);  
    orderBillRecyclerView.setVisibility(View.GONE);  
    emptyView.setVisibility(View.GONE);  
  
    // Lấy danh sách đơn hàng  
    userController.getUserOrderBills(new UserModel.OrderBillsCallback() {  
        @Override  
        public void onSuccess(List<OrderBill> orderBills) {  
            // Ẩn ProgressBar  
            progressBar.setVisibility(View.GONE);  
  
            // Cập nhật danh sách đơn hàng  
            orderBillList.clear();  
            if (orderBills != null) {  
                orderBillList.addAll(orderBills);  
            }  
  
            // Cập nhật adapter  
            orderBillAdapter.notifyDataSetChanged();  
  
            // Hiển thị thông báo nếu không có đơn hàng  
            if (orderBillList.isEmpty()) {  
                emptyView.setVisibility(View.VISIBLE);  
                orderBillRecyclerView.setVisibility(View.GONE);  
            } else {  
                emptyView.setVisibility(View.GONE);  
                orderBillRecyclerView.setVisibility(View.VISIBLE);  
            }  
        }  
    });  
}
```

Hình 22: Mã minh họa cho chức năng xem lịch sử mua hàng

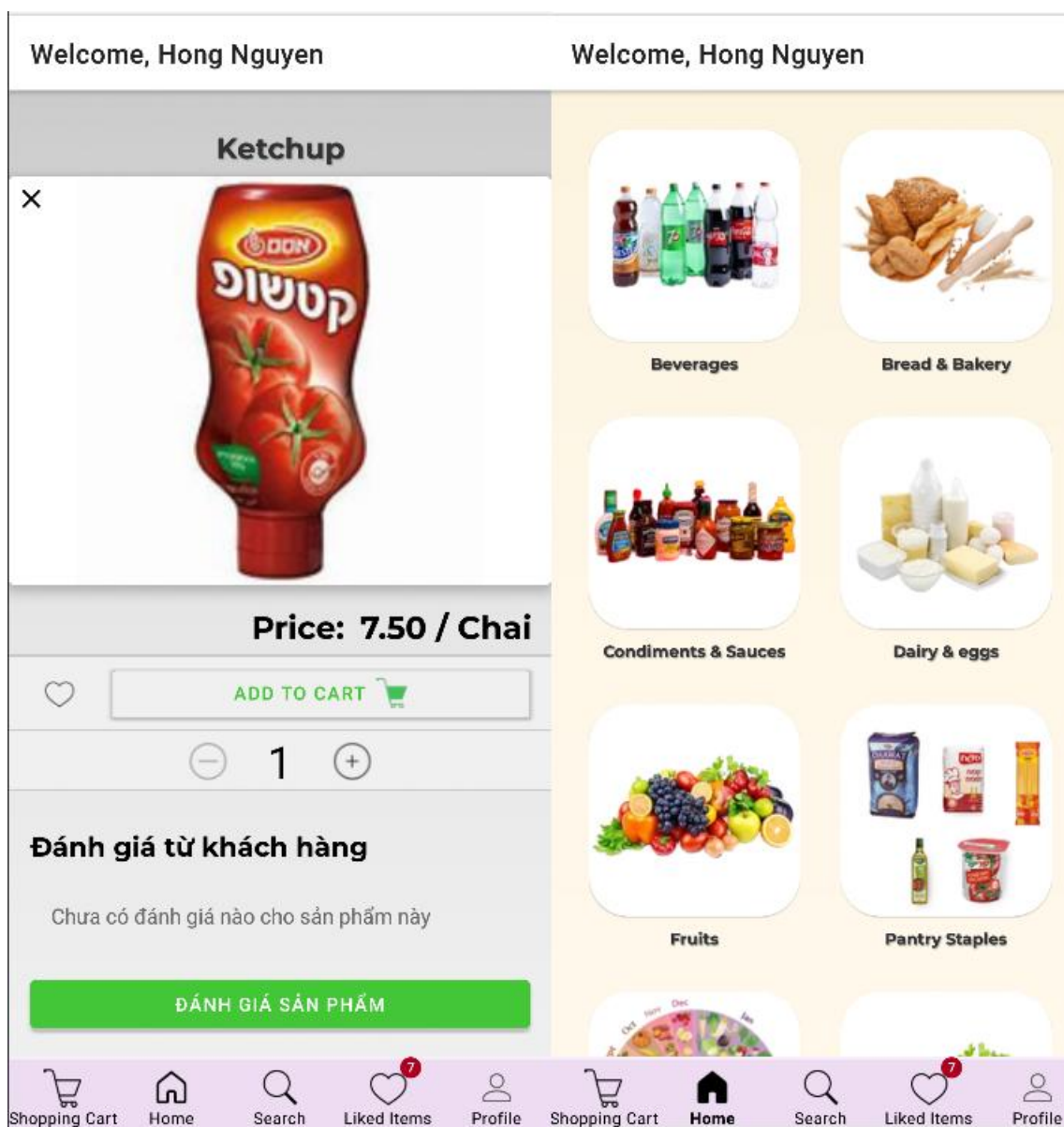
Mỗi đơn hàng hiển thị các thông tin cơ bản như mã đơn hàng (orderBillId), trạng thái (PENDING, PAID, Success), ngày đặt (orderDate) và tổng giá trị (totalPrice). Danh sách sản phẩm trong đơn hàng được hiển thị chi tiết bao gồm tên sản phẩm, số lượng, đơn giá và ảnh minh họa. Giao diện quản lý đơn hàng hỗ trợ người dùng theo dõi tiến trình của từng đơn hàng, bao gồm các trạng thái như chờ xác nhận, đang giao, đã giao thành công hoặc bị hủy. Người dùng có thể bấm vào từng đơn để xem chi tiết hoặc gửi yêu cầu hỗ trợ.



Hình 23: Giao diện xem lịch sử đơn hàng trên ứng dụng

Giao diện người dùng (UI)

Giao diện người dùng của ứng dụng được xây dựng theo hướng trực quan, dễ sử dụng, phù hợp với trải nghiệm mua sắm trên thiết bị di động. Các tệp giao diện được thiết kế bằng XML, với bố cục linh hoạt sử dụng `LinearLayout` và `ConstraintLayout`. Một số layout chính bao gồm: màn hình chính hiển thị danh mục sản phẩm (`Categories_page.xml`), màn hình hiển thị chi tiết sản phẩm (`edit_item_page.xml`), giỏ hàng (`shopping_cart_page.xml`), và lịch sử mua hàng (`fragment_order_history.xml`). Mỗi danh sách sản phẩm hoặc đơn hàng đều sử dụng `RecyclerView` để hiển thị dữ liệu động từ Firebase.



Hình 24: Giao diện người dùng

Giao diện sử dụng các thành phần chuẩn như ImageView để hiển thị hình ảnh sản phẩm (dữ liệu từ Firebase – đường dẫn ảnh), TextView để hiển thị tên và giá sản phẩm, và Button cho các thao tác như "Thêm vào giỏ" hoặc "Đặt hàng".

```
<ImageView
    android:id="@+id/itemImage"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/apple" />

<ImageButton
    android:id="@+id/closeButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:background="@drawable/baseline_close_24" />
/com.google.android.material.card.MaterialCardView>

<TextView
    android:id="@+id/itemName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
```

Hình 25: Mã minh họa hiển thị sản phẩm

Ứng dụng hỗ trợ chế độ sáng (Light Mode) và tối (Dark Mode) bằng cách triển khai hai bộ theme riêng biệt trong thư mục res/values/themes.xml và res/values-night/themes.xml. Các màu sắc được định nghĩa trong colors.xml và được sử dụng dưới dạng thuộc tính (?attr/colorPrimary, ?attr/backgroundColor) để tự động thay đổi theo chế độ hệ thống. Việc chuyển đổi giữa hai chế độ dựa trên cài đặt hệ thống của Android hoặc tùy chọn người dùng.





Giao diện người dùng đóng vai trò trung gian quan trọng giữa người dùng và dữ liệu từ Firebase, đảm bảo quá trình truy xuất, hiển thị và tương tác với thông tin như sản phẩm, giỏ hàng, đơn hàng hay đánh giá diễn ra mượt mà và rõ ràng.

Chức năng thay đổi mật khẩu

TÀI KHOẢN



Hong Nguyen
hongnguyen@gmail.com

-  Thông tin chi tiết >
-  Đổi mật khẩu >
-  Lịch sử mua hàng >
-  **Đăng xuất** >



Hình 26: Giao diện tài khoản trong ứng dụng

Click vào nút “Đổi mật khẩu”. Hệ thống sẽ điều hướng đến trang thay đổi mật khẩu như hình bên dưới.

< Thay đổi mật khẩu



Mật khẩu hiện tại

Mật khẩu mới

Xác thực mật khẩu

CẬP NHẬT

Shopping Cart Home Search Liked Items Profile

Hình 27: Giao diện thay đổi mật khẩu

Đây là trang thay đổi mật khẩu. Hệ thống yêu cầu người dùng nhập mật khẩu hiện tại, nhập mật khẩu mới, nhập lại xác thực mật khẩu mới. Sau khi nhập đầy đủ thông tin, click nút “Cập nhật” để tiến hành quá trình đổi mật khẩu. Nếu hệ thống xác nhận các thông tin hợp lệ, ta sẽ hoàn thành chức năng đổi mật khẩu.

```
1 usage
private void changePassword() {
    String currentPwd = currentPassword.getText().toString().trim();
    String newPwd = newPassword.getText().toString().trim();
    String confirmPwd = confirmPassword.getText().toString().trim();

    if (TextUtils.isEmpty(currentPwd) || TextUtils.isEmpty(newPwd) || TextUtils.isEmpty(confirmPwd)) {
        Toast.makeText(getContext(), text: "Vui lòng nhập đầy đủ thông tin", Toast.LENGTH_SHORT).show();
        return;
    }

    if (!newPwd.equals(confirmPwd)) {
        Toast.makeText(getContext(), text: "Mật khẩu mới và xác nhận không khớp", Toast.LENGTH_SHORT).show();
        return;
    }

    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    if (user == null || user.getEmail() == null) {
        Toast.makeText(getContext(), text: "Không tìm thấy người dùng", Toast.LENGTH_SHORT).show();
        return;
    }

    AuthCredential credential = EmailAuthProvider.getCredential(user.getEmail(), currentPwd);

    // Xác thực lại
    user.reauthenticate(credential).addOnCompleteListener( Task<Void> task -> {
        if (task.isSuccessful()) {
            // Cập nhật mật khẩu trong Firebase Authentication
            user.updatePassword(newPwd).addOnCompleteListener( Task<Void> updateTask -> {
```

Hình 28: Mã minh họa cho chức năng đổi mật khẩu

Đây là hàm thực hiện chức năng thay đổi mật khẩu. Hàm được kích hoạt khi người dùng bấm nút “btnSetPassword”. Hàm có xử lý các ngoại lệ khi người dùng nhập sai mật khẩu, nhập xác thực mật khẩu không khớp với mật khẩu mới.

Chỉnh sửa thông tin khách hàng

<

CHỈNH SỬA THÔNG TIN

LƯU

Họ

Nguyen

Tên


Hong

Số điện thoại


+97204939489

Địa chỉ


97 Man Thien, Phuong Hiep Phu, Quan 9, T




Shopping Cart



Home




Search



7

Liked Items



Profile

Hình 29: Giao diện sửa đổi thông tin

40

Đây là giao diện của trang đổi thông tin tài khoản. Ở đây ta có thể thay đổi họ tên, số điện thoại, địa chỉ. Hệ thống sẽ hiển thị ra các thông tin hiện tại. Nếu người dùng muốn thay đổi thông tin nào, chỉ cần click vào ô input đó và nhập thông tin mới; lưu ý không được để trống bất kỳ trường thông tin nào. Sau khi đã nhập các thông tin cần sửa đổi xong, click nút “Lưu” để gọi hàm cập nhật thông tin khách hàng.

```
// Lưu thông tin người dùng xuống Firebase
1 usage
private void saveUserData() {
    // Lấy giá trị từ các trường nhập liệu
    String lastName = lastNameInput.getText().toString().trim();
    String firstName = firstNameInput.getText().toString().trim();
    String phoneNumber = phoneNumberInput.getText().toString().trim();
    String address = addressInput.getText().toString().trim();

    // Kiểm tra dữ liệu
    if (lastName.isEmpty()) {
        lastNameInput.setError("Vui lòng nhập họ");
        return;
    }

    if (firstName.isEmpty()) {
        firstNameInput.setError("Vui lòng nhập tên");
        return;
    }

    if (phoneNumber.isEmpty()) {
        phoneNumberInput.setError("Vui lòng nhập số điện thoại");
        return;
    }

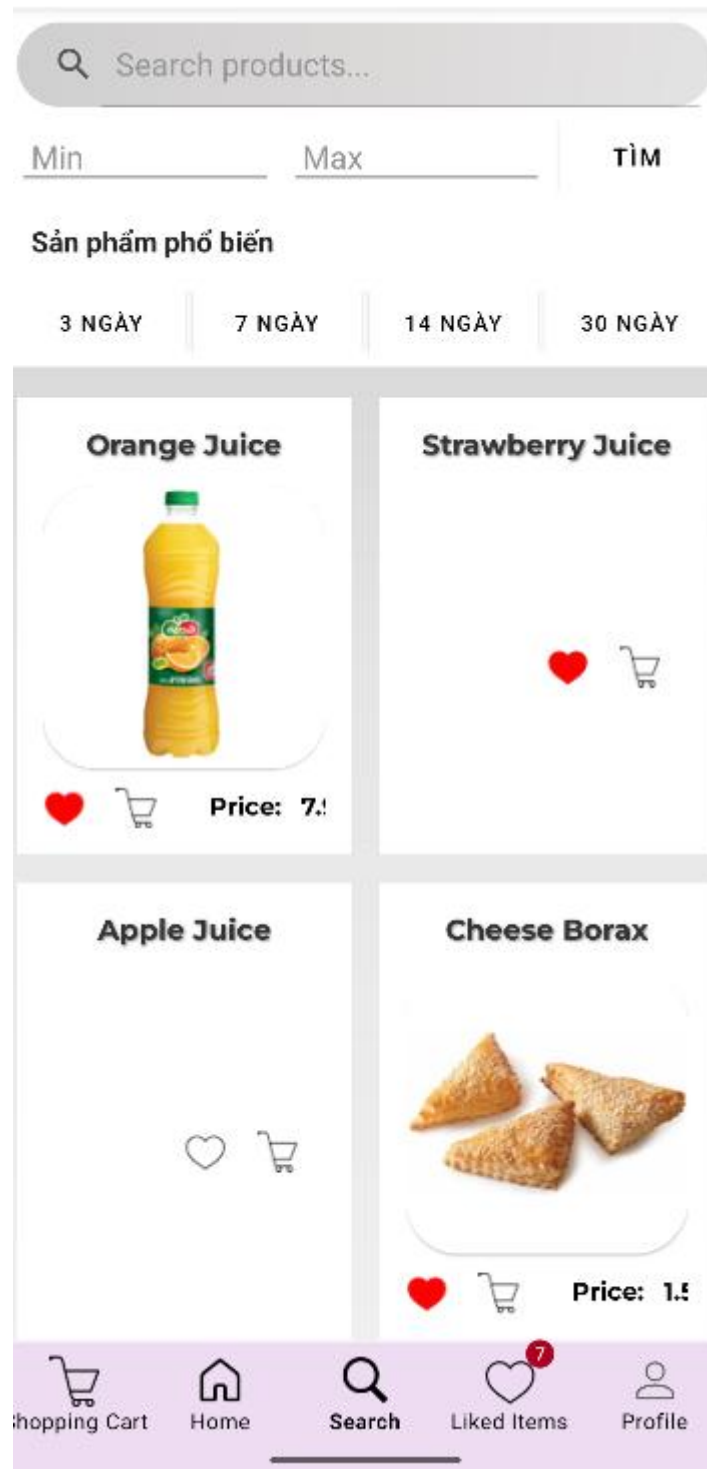
    // Hiển thị thông báo đang cập nhật
    Toast.makeText(getActivity(), text: "Đang cập nhật thông tin...", Toast.LENGTH_SHORT).show();

    // Gọi UserController để cập nhật thông tin
}
```

Hình 30: Mã minh họa cho chức năng đổi thông tin tài khoản

Đây là hàm lưu thông tin người dùng xuống Firebase khi người dùng chỉnh sửa thông tin. Hàm có xử lý ngoại lệ, yêu cầu người dùng nhập đầy đủ thông tin. Ngoài ra hàm trên cũng có phần xử lý nếu quá trình cập nhật có lỗi.

Chức năng lọc sản phẩm theo giá



Hình 31: Giao diện lọc sản phẩm theo giá

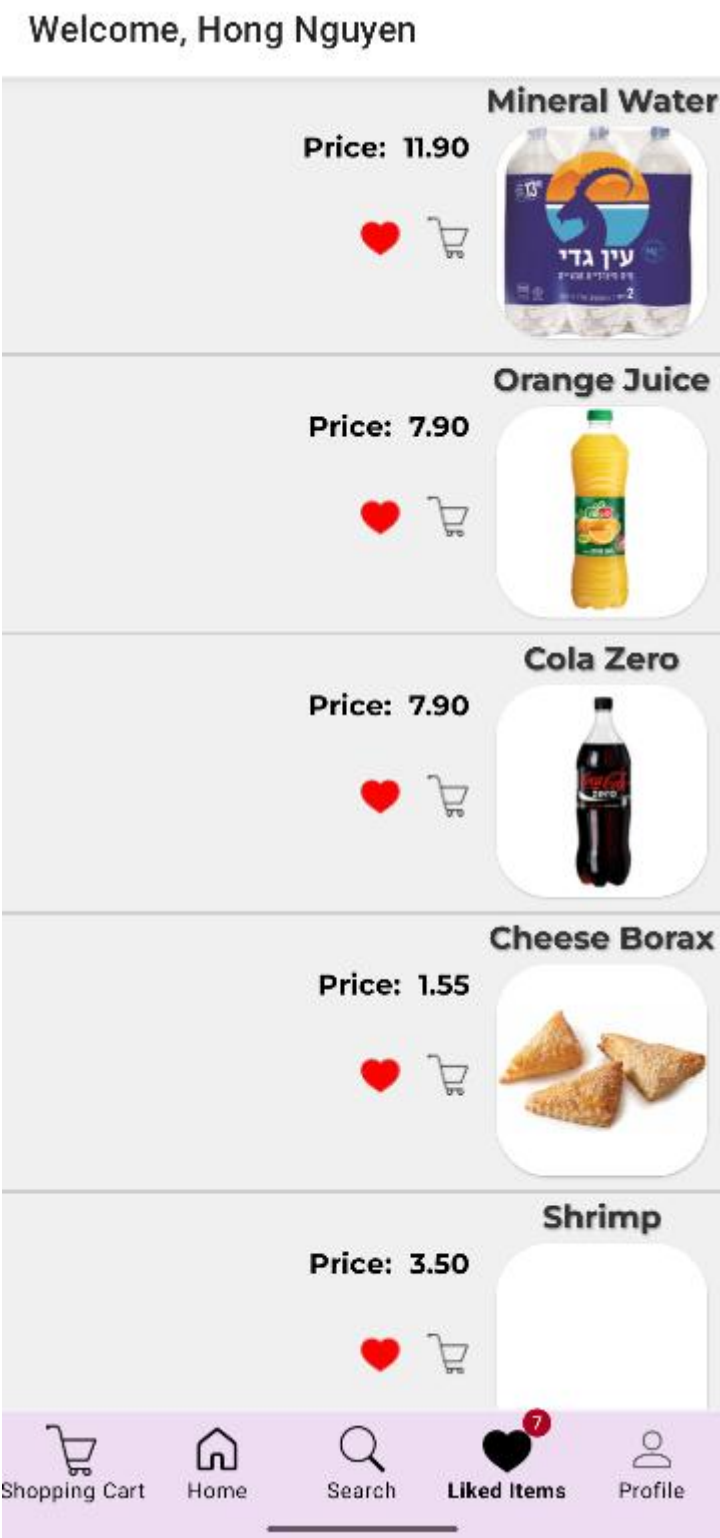
Đây là giao diện chức năng tìm kiếm sản phẩm. Chức năng tìm kiếm bao gồm tìm kiếm sản phẩm theo tên, tìm kiếm sản phẩm theo giá, tìm kiếm sản phẩm theo độ phổ biến. Người dùng nhập tên sản phẩm cần tìm vào ô “Search products”.

```
priceFilterButton.setOnClickListener( View v -> {  
    String minPriceStr = minPriceEditText.getText().toString().trim();  
    String maxPriceStr = maxPriceEditText.getText().toString().trim();  
  
    // Kiểm tra giá trị nhập vào  
    if (minPriceStr.isEmpty() && maxPriceStr.isEmpty()) {  
        Toast.makeText(getContext(), text: "Please enter at least one price value", Toast.LENGTH_SHORT).show();  
        return;  
    }  
    Double minPrice = null;  
    Double maxPrice = null;  
    try {  
        if (!minPriceStr.isEmpty()) {  
            minPrice = Double.parseDouble(minPriceStr);  
        }  
        if (!maxPriceStr.isEmpty()) {  
            maxPrice = Double.parseDouble(maxPriceStr);  
        }  
        // Kiểm tra trường hợp cả minPrice và maxPrice đều được nhập  
        if (minPrice != null && maxPrice != null && minPrice > maxPrice) {  
            Toast.makeText(getContext(), text: "Minimum price cannot be greater than maximum price", Toast.LENGTH_SHORT).show();  
            return;  
        }  
    } catch (NumberFormatException e) {  
        Toast.makeText(getContext(), text: "Please enter valid numbers for price", Toast.LENGTH_SHORT).show();  
        return;  
    }  
}
```

Hình 32: Mã minh họa chức năng tìm kiếm sản phẩm

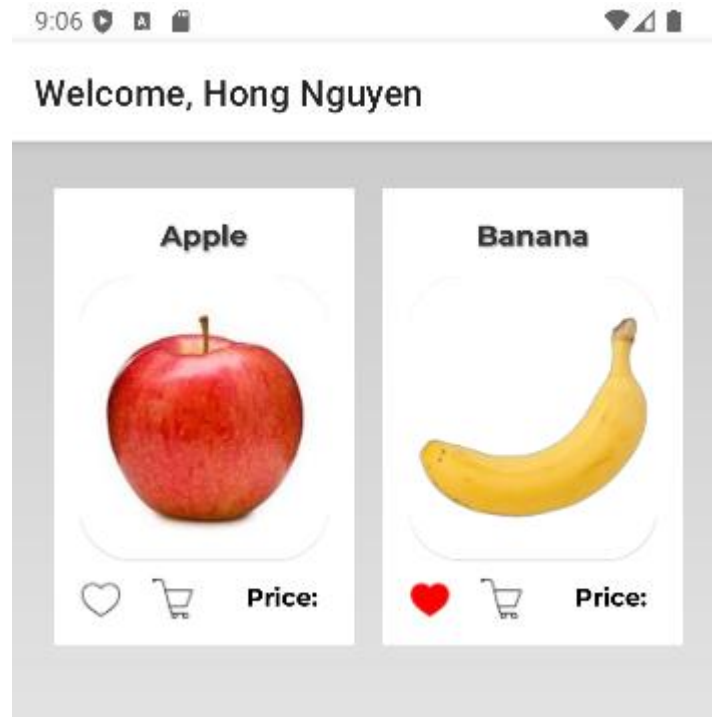
Đây là hàm dùng để tìm kiếm sản phẩm theo giá. Hàm sẽ bắt lỗi nếu người dùng nhập giá sản phẩm không hợp lệ (số âm hoặc chứa ký tự chữ cái), đồng thời giá Min phải < giá Max. Nếu kết quả trả về hợp lệ, hệ thống sẽ hiển thị kết quả dưới dạng list. Nếu không có sản phẩm cần tìm thì hệ thống sẽ thông báo “No items found in this price range”.

Chức năng quản lý sản phẩm yêu thích



Hình 33: Giao diện quản lý sản phẩm yêu thích

Đây là danh sách các sản phẩm đã được thêm vào Liked Items. Đây là chức năng phổ biến trên nhiều ứng dụng mua hàng, nó giúp người dùng đánh dấu được các sản phẩm mà họ thích và dựa trên cơ sở này, hệ thống có thể thu thập thông tin và đề xuất cho người dùng nhiều sản phẩm tương tự như vậy.



Hình 34: Sản phẩm yêu thích

Để thêm một sản phẩm vào Liked Items, chỉ cần bấm vào biểu tượng hình trái tim. Nếu sản phẩm đã được thêm vào Liked Items, trái tim sẽ có màu đỏ như trên.

```

1 usage
private void createRecyclerView()
{
    LinearLayoutManager layoutManager = new LinearLayoutManager(getContext());
    m_LikedItemsRecyclerView.setLayoutManager(layoutManager);
    m_Adapter = new LikedItemsAdapter(m_LikedItemsList, m_HostedActivity, i_Listener: this);
    m_LikedItemsRecyclerView.setAdapter(m_Adapter);
}

```

Hình 35: Mã minh họa chức năng lấy sản phẩm yêu thích

Đây là hàm lấy danh sách các sản phẩm trong Liked Items để hiển thị ra cho người dùng.

CHƯƠNG 4 TỔNG KẾT

4.1 Kết quả đạt được

Kết thúc quá trình phát triển ứng dụng, nhóm đã cơ bản hoàn thành đầy đủ các chức năng theo đúng mục tiêu đề ra trong Chương 1. Hệ thống đã được kết nối thành công với Firebase Realtime Database, hoạt động ổn định trên nền tảng Android và có khả năng lưu trữ, xử lý dữ liệu người dùng cũng như các thao tác mua hàng một cách hiệu quả.

Kết quả đạt được gồm có:

- + Hệ thống quản lý tài khoản người dùng bằng Firebase Authentication, hỗ trợ đăng ký, đăng nhập, chỉnh sửa thông tin.
- + Giao diện hiển thị danh sách sản phẩm được thiết kế bằng RecyclerView, dữ liệu lấy trực tiếp từ Firebase.
- + Chức năng tìm kiếm, lọc sản phẩm theo danh mục, mùa và từ khóa hoạt động tốt.
- + Tính năng thêm/xóa sản phẩm vào giỏ hàng, cập nhật số lượng, tính tổng giá trị.
- + Hệ thống đặt hàng và lưu đơn hàng về node OrderBills theo từng userId, người dùng có thể theo dõi lịch sử đơn hàng của mình.
- + Chức năng đánh giá sản phẩm, thêm vào danh sách yêu thích, đảm bảo tương tác giữa người dùng và sản phẩm.
- + Hệ thống mã giảm giá được hỗ trợ, có thể áp dụng theo sản phẩm và thời gian hiệu lực.
- + Giao diện được tối ưu với hỗ trợ chế độ sáng/tối linh hoạt, mang lại trải nghiệm thân thiện.

Dự án đã đạt được tính hoàn thiện cơ bản, đáp ứng được bài toán đặt ra ban đầu, và sẵn sàng để tiếp tục phát triển thêm các tính năng nâng cao như thanh toán trực tuyến, quản trị viên (admin), hoặc hệ thống đề xuất sản phẩm thông minh dựa trên hành vi người dùng.

4.2 Hạn chế

Trong quá trình xây dựng và triển khai dự án, nhóm đã cố gắng hoàn thiện các chức năng cơ bản, tuy nhiên vẫn còn tồn tại một số hạn chế nhất định như sau:

- + Giao diện người dùng mới chỉ tối ưu tốt trên một số kích thước màn hình phổ biến, chưa phản hồi tốt trên các thiết bị có độ phân giải quá thấp hoặc quá cao.
- + Ứng dụng chưa triển khai chức năng theo dõi đơn hàng theo thời gian thực (ví dụ: trạng thái giao hàng trên bản đồ).
- + Chưa tích hợp các phương thức thanh toán trực tuyến như ví điện tử hoặc thẻ ngân hàng, người dùng chỉ có thể đặt hàng mô phỏng.
- + Chức năng chat hỗ trợ người dùng mới chỉ dừng ở mức mô phỏng, chưa kết nối với dịch vụ chat thật sự.
- + Dữ liệu Firebase hiện đang sử dụng Rule test (cho phép đọc/ghi tự do), chưa đảm bảo an toàn bảo mật trong môi trường triển khai thực tế.
- + Việc quản lý dữ liệu giảm giá chưa tự động hóa việc kiểm tra hạn dùng hoặc xung đột giữa nhiều mã giảm giá.

Những hạn chế trên sẽ được nhóm tiếp tục nghiên cứu và cải thiện trong các giai đoạn phát triển tiếp theo để hoàn thiện hệ thống hơn nữa, tăng trải nghiệm và độ tin cậy cho người dùng.

4.3 Hướng phát triển

Để nâng cao trải nghiệm người dùng và phát triển hệ thống ngày càng hoàn thiện hơn, trong tương lai nhóm đề xuất một số hướng mở rộng sau:

- + Gợi ý sản phẩm theo mùa vụ: Dựa trên trường "Season" trong danh mục sản phẩm, hệ thống có thể tự động đề xuất các mặt hàng phù hợp với thời điểm hiện tại trong năm (xuân, hạ, thu, đông), giúp người dùng dễ lựa chọn và tăng khả năng mua hàng.
- + Tích hợp bản đồ theo dõi đơn hàng thời gian thực: Sử dụng dịch vụ bản đồ (Google Maps API), người dùng có thể theo dõi vị trí đơn hàng khi đang được giao, gia tăng tính minh bạch và độ tin cậy cho hệ thống.
- + Xây dựng hệ thống thông báo khuyến mãi: Phát triển tính năng gửi thông báo (Firebase Cloud Messaging) khi có các mã giảm giá mới, sản phẩm sắp hết hàng, hoặc cập nhật trạng thái đơn hàng. Điều này giúp tăng tương tác với người dùng và nâng cao hiệu quả bán hàng.
- + Phát triển thêm phiên bản cho nền tảng iOS: Hiện tại hệ thống mới hoạt động trên nền tảng Android. Việc xây dựng phiên bản iOS sẽ mở rộng đối tượng người dùng, nâng cao tính phổ biến và chuyên nghiệp cho ứng dụng.

PHÂN CHIA CÔNG VIỆC NHÓM

STT	Công việc	Tên thành viên thực hiện - MSSV
1	<p>Danh mục sản phẩm nông sản:</p> <ul style="list-style-type: none"> • Hiện thị danh sách sản phẩm: rau củ, trái cây, thực phẩm khô, sản phẩm hữu cơ,... • Tìm kiếm và lọc sản phẩm theo loại.. <p>Giỏ hàng, Yêu thích, và đặt mua:</p> <ul style="list-style-type: none"> • Cho phép thêm/xoá sản phẩm, cập nhật số lượng trong giỏ hàng. • Tính tổng chi phí đơn hàng. • Hiện thị danh sách sản phẩm trong giỏ hàng , danh sách sản phẩm yêu thích. • Thêm sản phẩm vào danh sách yêu thích <p>Thanh toán:</p> <p>Quản lý đơn hàng:</p> <ul style="list-style-type: none"> • Cho phép huỷ đơn hàng trong thời gian cho phép. <p>Tài khoản người dùng:</p> <ul style="list-style-type: none"> • Đăng xuất ,chỉnh sửa thông tin cá nhân . • Lưu trạng thái đăng nhập người dùng <p>Tương tác và hỗ trợ:</p> <ul style="list-style-type: none"> • Hệ thống đánh giá, phản hồi về sản phẩm. <p>Giao diện người dùng:</p> <ul style="list-style-type: none"> • Thiết kế trực quan, phù hợp người tiêu dùng phổ thông. <p>Mở rộng khác (tùy chọn):</p>	Bùi Hữu Trí - N21DCAT058

2	<p>Danh mục sản phẩm nông sản:</p> <ul style="list-style-type: none"> • Tìm kiếm và lọc sản phẩm theo độ phổ biến. <p>Giỏ hàng, Yêu thích, và đặt mua:</p> <ul style="list-style-type: none"> • Cho phép thêm/xoá sản phẩm, cập nhật số lượng trong giỏ hàng. • Tính tổng chi phí đơn hàng. <p>Thanh toán:</p> <ul style="list-style-type: none"> • Hỗ trợ thanh toán khi nhận hàng (COD) <p>Quản lý đơn hàng:</p> <ul style="list-style-type: none"> • Hiện thị trạng thái đơn hàng: đang xử lý, đang giao, hoàn thành. <p>Tài khoản người dùng:</p> <ul style="list-style-type: none"> • Đăng ký, đăng nhập . <p>Tương tác và hỗ trợ:</p> <p>Giao diện người dùng:</p> <ul style="list-style-type: none"> • Thiết kế trực quan, phù hợp người tiêu dùng phổ thông. <p>Mở rộng khác (tùy chọn):</p>	Đinh Quốc Toàn – N21DCAT057
3	<p>Danh mục sản phẩm nông sản:</p> <ul style="list-style-type: none"> • Thông tin chi tiết: tên, mô tả, hình ảnh, giá, nguồn gốc. <p>Giỏ hàng, Yêu thích, và đặt mua:</p> <p>Thanh toán:</p>	Nguyễn Bá Trung –N21DCAT060

	<p>Quản lý đơn hàng:</p> <p>Tài khoản người dùng:</p> <p>Tương tác và hỗ trợ:</p> <p>Giao diện người dùng:</p> <ul style="list-style-type: none"> Thiết kế trực quan, phù hợp người tiêu dùng phổ thông. <p>Mở rộng khác (tùy chọn):</p> <p>Gợi ý sản phẩm theo mùa vụ.</p> <p>Hỗ trợ Light/Dark Mode</p>	
4	<p>Danh mục sản phẩm nông sản:</p> <ul style="list-style-type: none"> Tìm kiếm và lọc sản phẩm theo giá. <p>Giỏ hàng, Yêu thích, và đặt mua:</p> <ul style="list-style-type: none"> Đặt hàng với thông tin địa chỉ và số điện thoại. Hiển thị danh sách sản phẩm trong giỏ hàng, danh sách sản phẩm yêu thích. Thêm sản phẩm vào danh sách yêu thích <p>Thanh toán:</p> <p>Quản lý đơn hàng:</p> <ul style="list-style-type: none"> Hiển thị trạng thái đơn hàng: đang xử lý, đang giao, hoàn thành. <p>Tài khoản người dùng:</p> <ul style="list-style-type: none"> Xem chi tiết thông tin cá nhân. Lưu lịch sử đơn hàng, địa chỉ giao hàng mặc định. <p>Tương tác và hỗ trợ:</p> <p>Giao diện người dùng:</p> <ul style="list-style-type: none"> Thiết kế trực quan, phù hợp người tiêu dùng 	Võ Hồng Nguyên - N21DCAT036

	phổ thông. Mở rộng khác (tùy chọn):	
5	Phân tích , thiết kế hệ thống , database	Bùi Hữu Trí – N21DCAT058 Võ Hồng Nguyên - N21DCAT036 Nguyễn Bá Trung –N21DCAT060 Đinh Quốc Toàn
6	Lý thuyết ,Tổng hợp kết quả , viết báo cáo	Võ Hồng Nguyên - N21DCAT036 Nguyễn Bá Trung –N21DCAT060

TÀI LIỆU THAM KHẢO

- [1] L. Chi, “Android Studio là gì? Hướng dẫn tải và cài đặt Android Studio,” FPT, 2023.
[Trực tuyến]. Available: <https://fptshop.com.vn/tin-tuc/thu-thuat/android-studio-la-gi->

150412. [Đã truy cập 10 06 2025].

[2] FPT, “Firebase là gì? Ưu nhược điểm & Các dịch vụ của Firebase,” FPT Cloud, 28 08 2024. [Trực tuyến]. Available: <https://fptcloud.com/firebase-la-gi/>. [Đã truy cập 09 06 2025].