

BÁO CÁO DỰ ÁN CUỐI KỲ

HỌC KỲ 2, NĂM HỌC 2023-2024

CT484: PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG

- **Tên dự án/ứng dụng:** Ứng Dụng Quản Lý Cửa Hàng Trà Sữa
- **Link GitHub:** <https://github.com/23-24Sem2-Courses/ct48402-project-phuocleaf.git>
- **MSSV 1:** B2014630
- **Họ tên SV 1:** Trương Ngọc Viên
- **MSSV 2:** B2014601
- **Họ tên SV 2:** Nguyễn Tấn Phước
- **Lớp học phần:** 02

I. Tổng quan

- **Miêu tả dự án/ứng dụng:** Ứng dụng quản lý cửa hàng Trà Sữa là một ứng dụng di động được thiết kế để cung cấp dịch vụ đặt hàng, quản lý sản phẩm và tương tác với khách hàng một cách trực quan, dễ sử dụng. Sử dụng Flutter để phát triển ứng dụng và Firebase để lưu trữ dữ liệu người dùng, quản lý đăng nhập và xác thực, cũng như quản lý dữ liệu sản phẩm và đơn hàng. Ứng dụng sẽ tạo một trải nghiệm mua sắm dễ dàng, thuận tiện bởi giao diện ưa nhìn, dễ chịu và dễ sử dụng.
- Bảng phân công công việc:

| Thành viên | Công việc |
|------------------|---|
| Nguyễn Tấn Phước | <ul style="list-style-type: none">- Thực hiện chức năng Đăng nhập Quản Trị Viên- Tạo trang Thông tin cá nhân người dùng- Tạo AppDrawer- Thực hiện chức năng thêm sản phẩm yêu thích, thêm vào giỏ hàng tại trang Home- Thực hiện chức năng Quản lý Sản Phẩm- Tạo trang và thực hiện chức năng trang Đơn hàng- Điều chỉnh các phần giao diện |
| Trương Ngọc Viên | <ul style="list-style-type: none">- Tạo trang và thực hiện Đăng ký, Đăng nhập, xác thực người dùng- Tạo BottomNavigationBar- Tạo trang và thực hiện chức năng thêm sản phẩm yêu thích, tăng giảm số lượng và thêm vào giỏ hàng tại trang Chi tiết Sản phẩm- Tạo trang Quản lý Sản Phẩm- Tạo trang Home: Carousel Slider, Sản Phẩm |

| | |
|--|--|
| | <ul style="list-style-type: none"> - Tạo trang và thực hiện chức năng trang Giỏ hàng - Điều chỉnh các phần giao diện - Viết báo cáo |
|--|--|

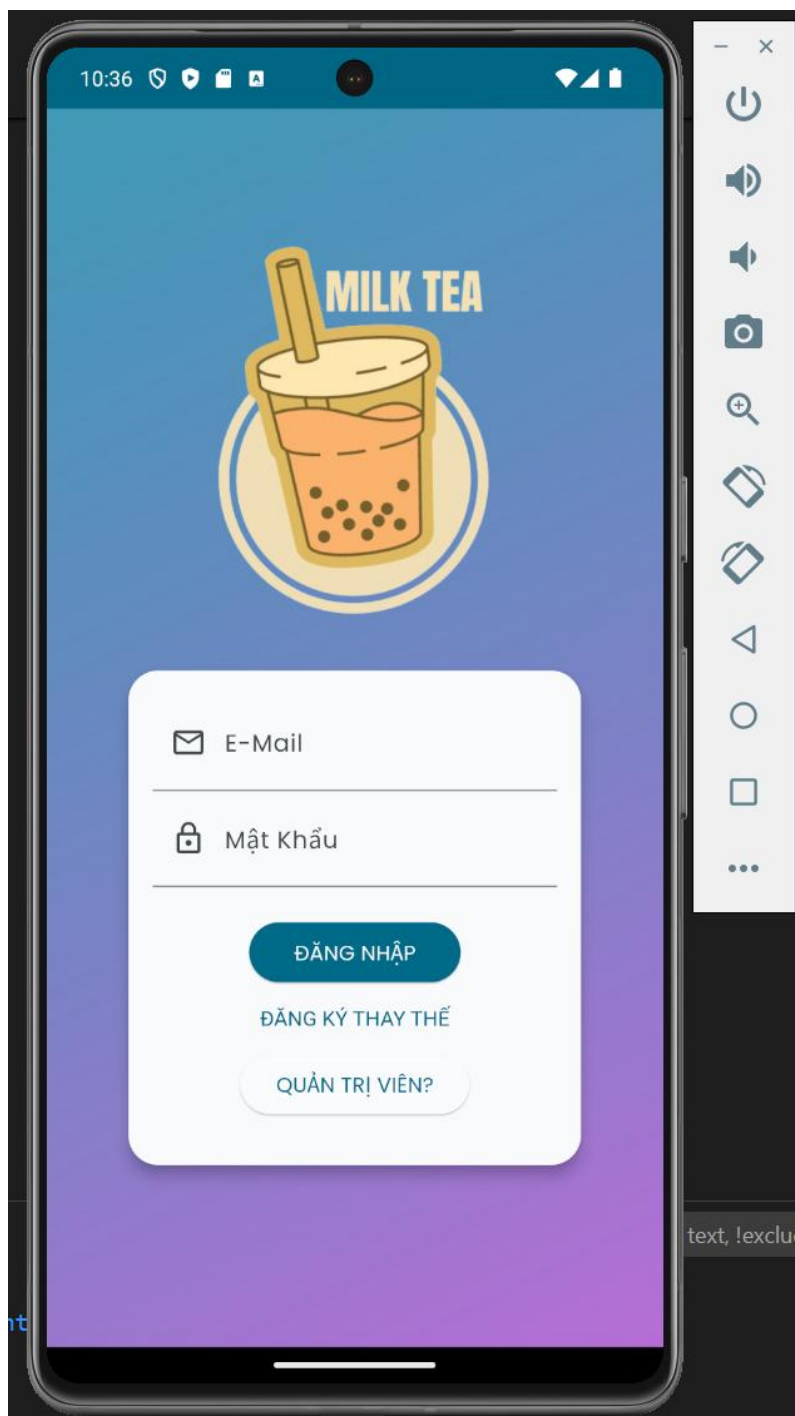
II. Chi tiết các chức năng

1. Chức năng/giao diện 1: Trang Đăng nhập/Đăng ký

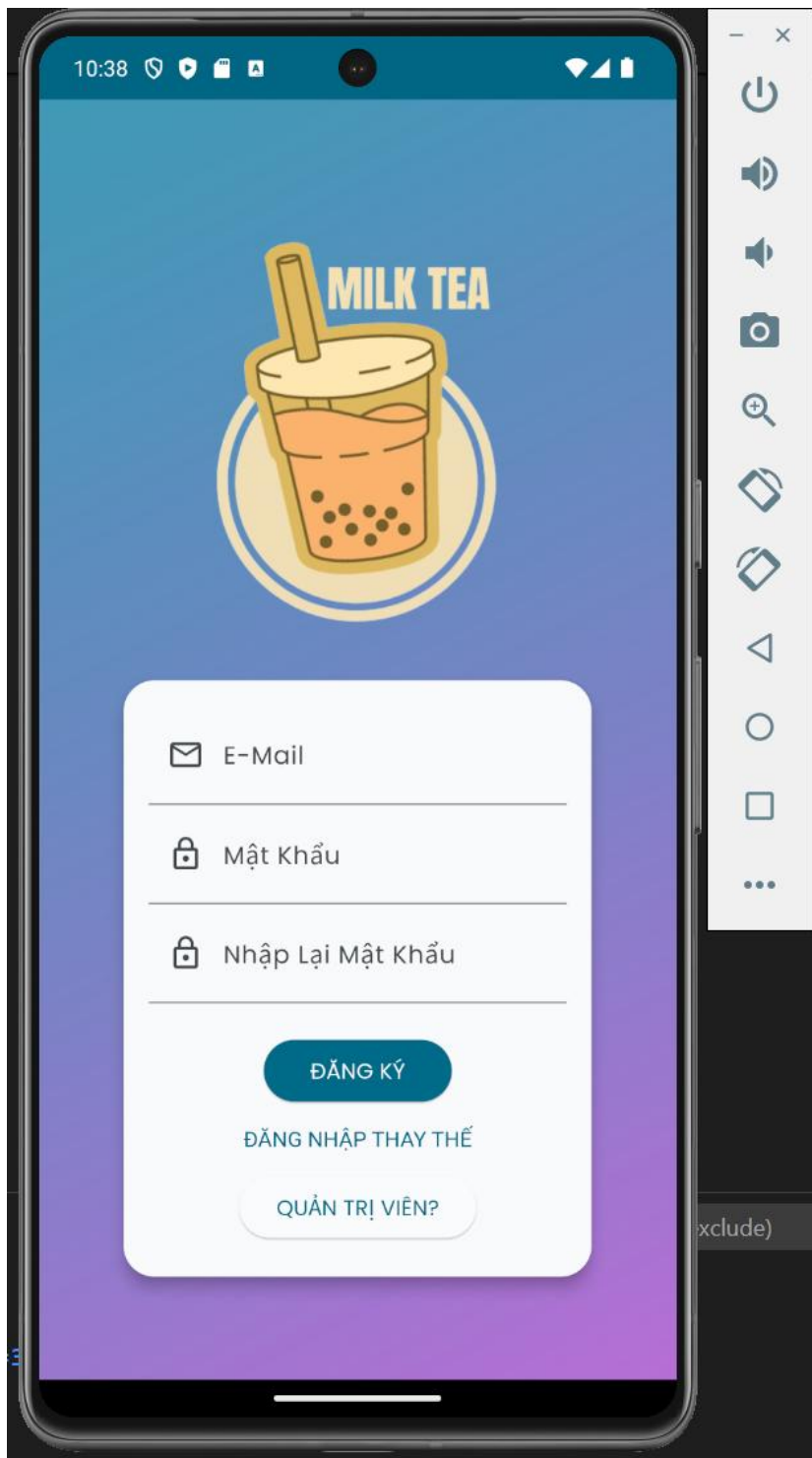
- Miêu tả chức năng/giao diện:

- + Đăng nhập: Người dùng nhập email và mật khẩu đã đăng ký để đăng nhập vào ứng dụng
- + Đăng ký: Người dùng cần nhập email, mật khẩu và xác nhận lại mật khẩu để đăng ký tài khoản
- + Đăng nhập “Quản Trị Viên”: Người dùng “Quản Trị Viên” dùng email và mật khẩu được cấp để đăng nhập vào ứng dụng

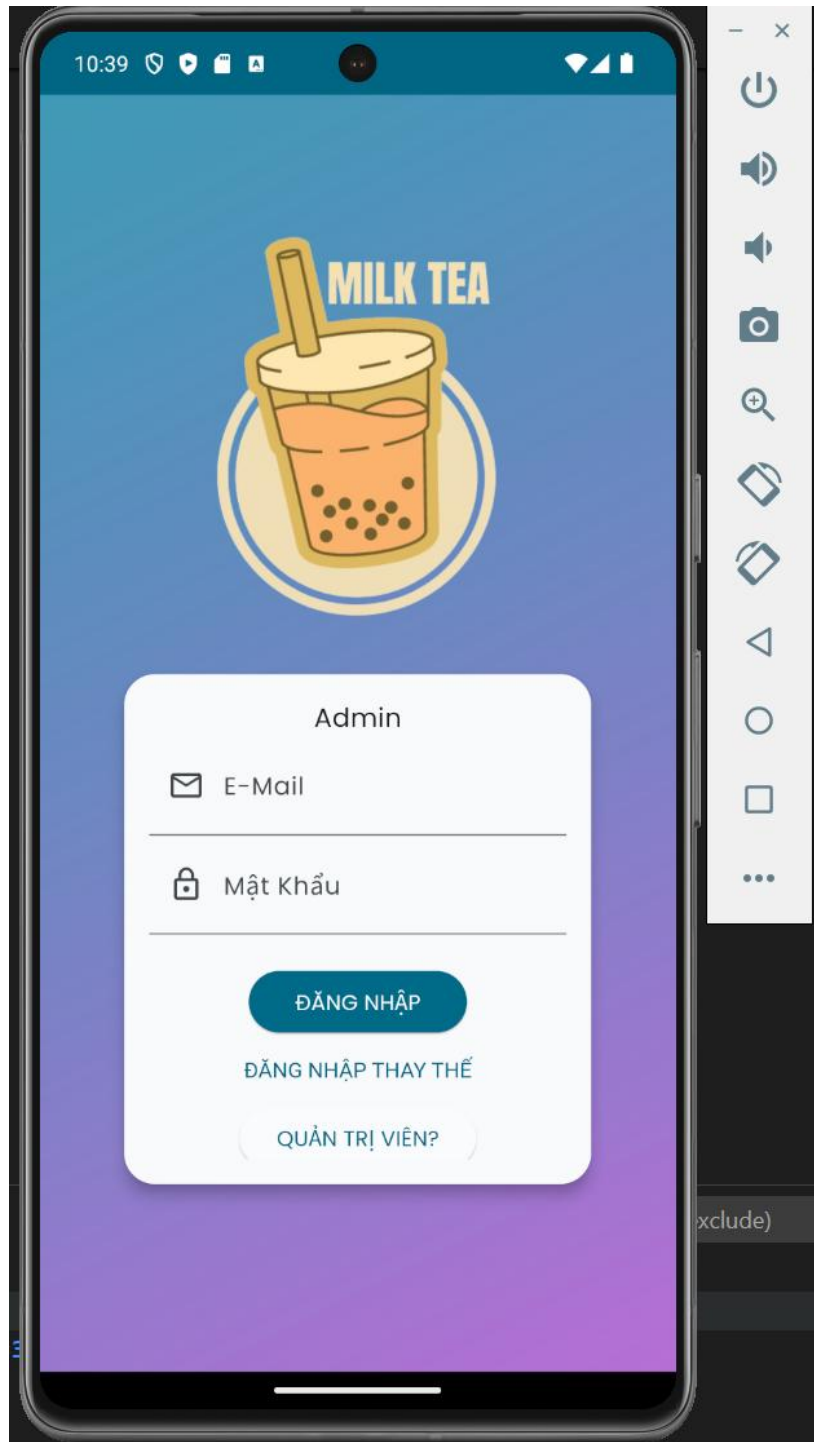
- Ảnh chức năng/giao diện:



Hình 1: Giao diện “Đăng Nhập” người dùng



Hình 2: Giao diện “Đăng ký” người dùng



Hình 3: Giao diện “Đăng Nhập” người Quản Trị Viên

- **Chi tiết cài đặt:**
 - + Các Widget được sử dụng cho chức năng:

- Container: Widget này được sử dụng để tạo một hộp chứa có thể tùy chỉnh với các thuộc tính như margin, padding, decoration,... Được sử dụng để chứa các thành phần trong thẻ.
- Row: Widget này được sử dụng để xếp các widget con theo hàng ngang. Được sử dụng để chứa hình ảnh banner.
- Image.asset: Được sử dụng để hiển thị hình ảnh banner.
- Card: Widget này được sử dụng để tạo một thẻ có góc bo tròn và đổ bóng. Sử dụng để bao bọc nội dung của phần đăng nhập hoặc đăng ký.
- Form: Widget này được sử dụng để tạo một biểu mẫu để nhập dữ liệu. Sử dụng để quản lý, xác thực dữ liệu đầu vào từ người dùng, và để bao bọc các trường nhập liệu và nút gửi.
- Stack: Được sử dụng để xếp chồng widget con để xếp background gradient và các widget khác lên nhau.
- SingleChildScrollView: Widget này được sử dụng để tạo một khung cuộn cho nội dung bên trong khi không gian hiển thị không đủ.
- Column: Widget này được sử dụng để xếp các widget con theo chiều dọc. Sử dụng để chứa các trường nhập liệu và nút gửi trong biểu mẫu.
- Text: Được sử dụng để hiển thị văn bản "Admin" khi ở chế độ đăng nhập của "Quản Trị Viên".
- TextFormField: Được sử dụng để nhập dữ liệu về email và mật khẩu từ người dùng.
- ElevatedButton: Được sử dụng để tạo nút gửi dữ liệu và nút chuyển đổi chế độ đăng nhập/đăng ký.
- ValueListenableBuilder: Được sử dụng để tạo một widget phản ứng dựa trên thay đổi của một giá trị nghe (listenable value).
- + Sử dụng thư viện:
 - Provider: Được sử dụng để quản lý trạng thái và chia sẻ dữ liệu giữa các widget.
- + Quản lý trạng thái chia sẻ:
 - AuthScreen: Sử dụng Consumer để lắng nghe trạng thái xác thực từ AuthManager và hiển thị màn hình tương ứng (màn hình đăng nhập hoặc màn hình chính) dựa trên trạng thái này. Nó cũng có trách nhiệm cho việc điều hướng người dùng đến màn hình phù hợp sau khi xác thực thành công hoặc khi ứng dụng được khởi động.
 - AuthManager: Sử dụng Provider để chia sẻ thông tin về trạng thái xác thực (đã đăng nhập hay chưa) và dữ liệu người dùng (ví dụ: thông tin token) trong toàn bộ ứng dụng. AuthManager cũng cung cấp các phương thức để thực hiện đăng nhập, đăng ký và đăng xuất.
- + Đọc hoặc lưu trữ dữ liệu:

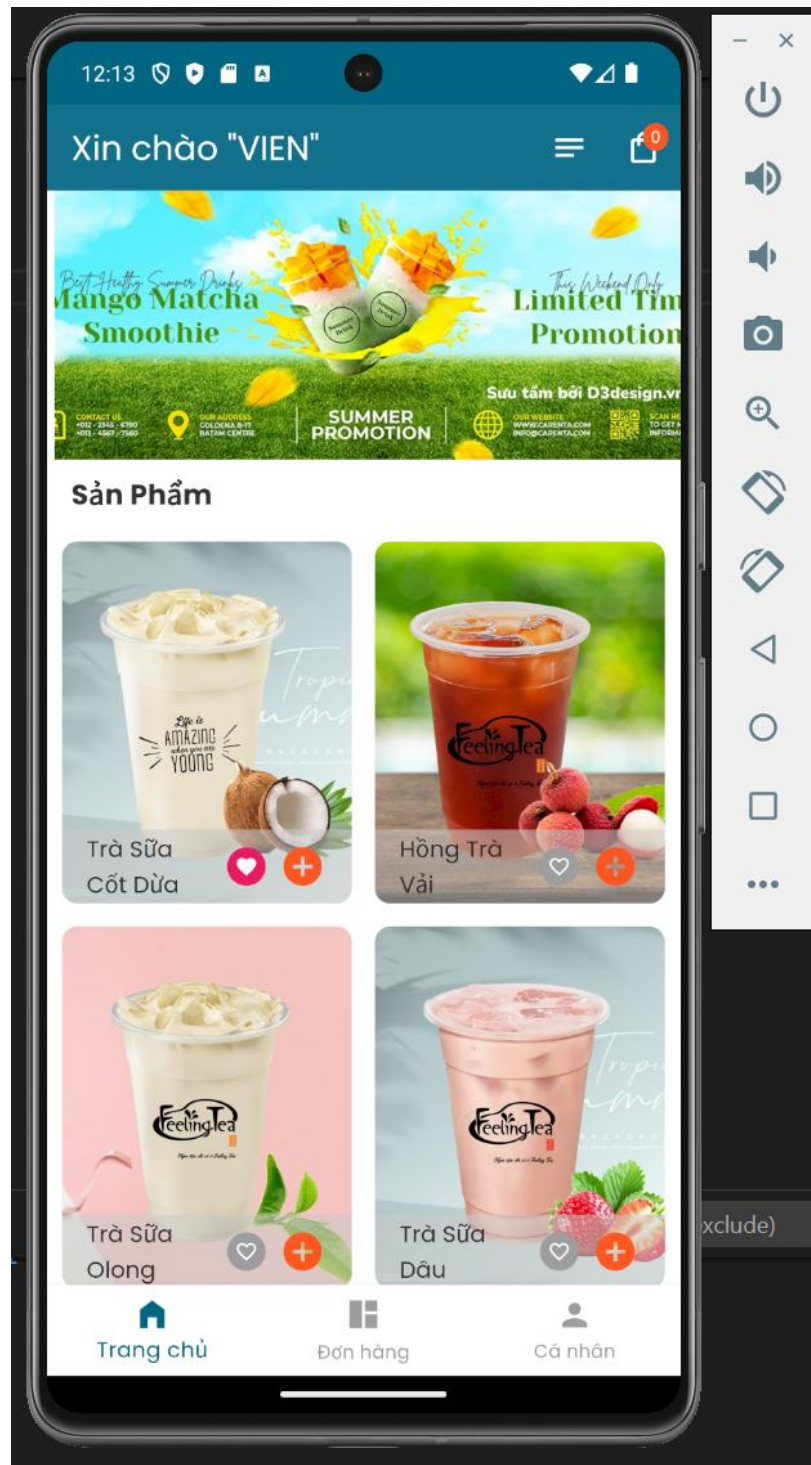
- Dữ liệu được lưu trữ và truy xuất thông qua các API Firebase của Google, chủ yếu là Firebase Authentication để xác thực người dùng.

2. Chức năng/giao diện 2: Trang Home

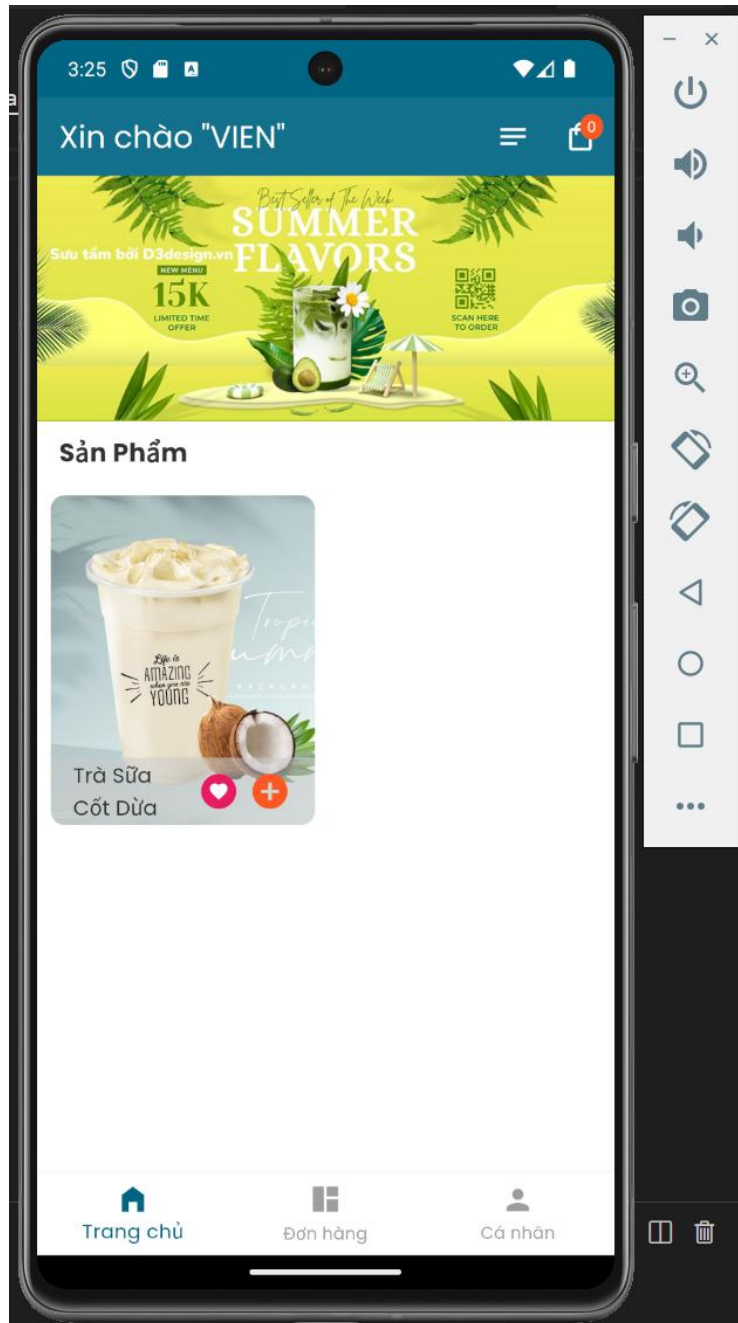
- Miêu tả chức năng/giao diện:

- + Giao diện này chủ yếu hiển thị hình ảnh quảng cáo và tất cả sản phẩm để người dùng chọn sản phẩm.
- + Người dùng có thể thêm sản phẩm vào giỏ hàng ngay ở trang Home, trên từng sản phẩm có Icon “Thêm sản phẩm” và có thể thêm sản phẩm vào danh sách sản phẩm yêu thích nhờ vào Icon “Thêm sản phẩm yêu thích”.
- + Trên AppBar, người dùng có thể chọn cách hiển thị sản phẩm như: “Tất cả Sản Phẩm” hoặc “Chỉ những sản phẩm yêu thích”, không chỉ vậy, thanh AppBar còn có “Giỏ hàng” để người dùng có thể nhanh chóng vào xem những sản phẩm mà mình đã thêm vào giỏ.
- + Trang Home còn có BottomNavigationBar, giúp người dùng dễ dàng di chuyển qua lại để xem thông tin đơn hàng, trang cá nhân và trang chủ.

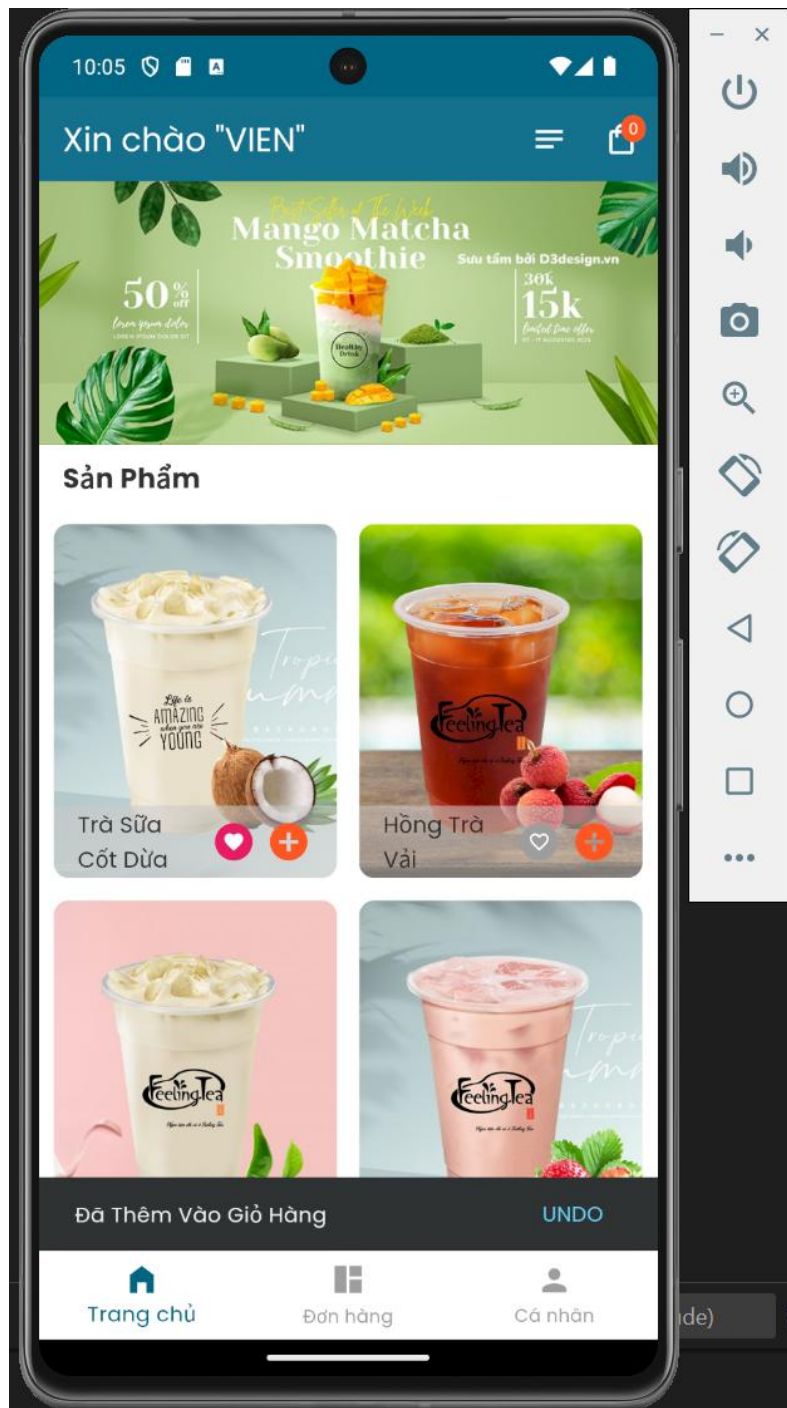
- Ảnh chức năng/giao diện:



Hình 4: Giao diện Trang Home



Hình 5: Giao diện trang Home hiển thị “Chỉ Sản phẩm yêu thích”



Hình 6: “Đã thêm sản phẩm vào giỏ hàng” tại trang Home

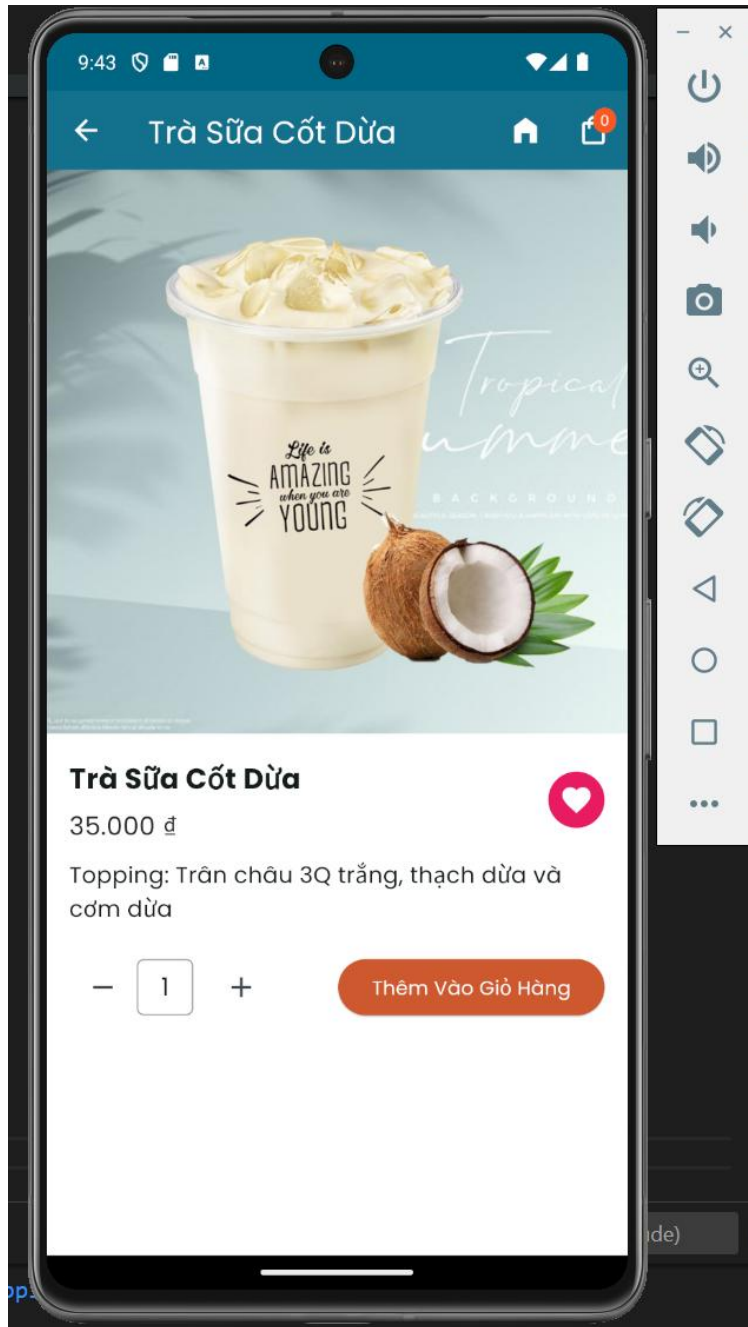
- Chi tiết cài đặt:
 - + Các Widget được sử dụng cho chức năng

- Scaffold: Được sử dụng để bao bọc toàn bộ nội dung của màn hình ProductsOverviewScreen.
- AppBar: Là thanh tiêu đề hiển thị ở đỉnh màn hình, sử dụng để hiển thị tên người dùng, nút lọc sản phẩm và biểu tượng giỏ hàng.
- FutureBuilder: Được sử dụng để hiển thị danh sách sản phẩm sau khi dữ liệu đã được tải về.
- ProductFilterMenu: Đây là menu lọc sản phẩm.
- PopUpMenuButton: Là một nút mở một menu pop-up khi được nhấn. Được sử dụng để cho phép người dùng chọn giữa hai tùy chọn lọc sản phẩm.
- ShoppingCartButton: Là nút giỏ hàng hiển thị số lượng sản phẩm trong giỏ hàng.
- Consumer: Được sử dụng để cập nhật số lượng sản phẩm trong giỏ hàng khi có thay đổi.
- TopRightBadge: Để hiển thị số lượng sản phẩm trong giỏ hàng trên biểu tượng giỏ hàng.
- ProductGridTile: Là một widget dùng để hiển thị một ô lưới chứa hình ảnh sản phẩm và các hành động tương tác như thêm vào giỏ hàng hoặc chuyển đến trang chi tiết sản phẩm.
- GestureDetector: Sử dụng để bắt sự kiện khi người dùng nhấn vào ô lưới để chuyển đến trang chi tiết sản phẩm.
- ProductGridFooter: Dùng để hiển thị tiêu đề và các hành động phụ của sản phẩm trong ô lưới.
- ValueListenableBuilder: Sử dụng để theo dõi trạng thái yêu thích của sản phẩm và cập nhật biểu tượng tương ứng.
- IconButton: Nút thêm vào giỏ hàng, khi nhấn vào sẽ gọi hàm xử lý được cung cấp.
- Text: Dùng để hiển thị tiêu đề "Sản Phẩm".
- GridView.builder: Widget này được sử dụng để hiển thị danh sách sản phẩm dưới dạng lưới.
- Provider: Được sử dụng để cung cấp dữ liệu sản phẩm từ ProductsManager cho các widget con.
- Padding: Sử dụng để tạo khoảng trống giữa tiêu đề và danh sách sản phẩm.
- ClipRRect: Được sử dụng để làm tròn góc của hình ảnh sản phẩm.
- GridTile: Được sử dụng để hiển thị nội dung của sản phẩm trong lưới.
- Stack: Được sử dụng để xếp chồng các widget lên nhau, cho phép hiển thị nhiều widget cùng một lúc trên cùng một vị trí trên giao diện người dùng.

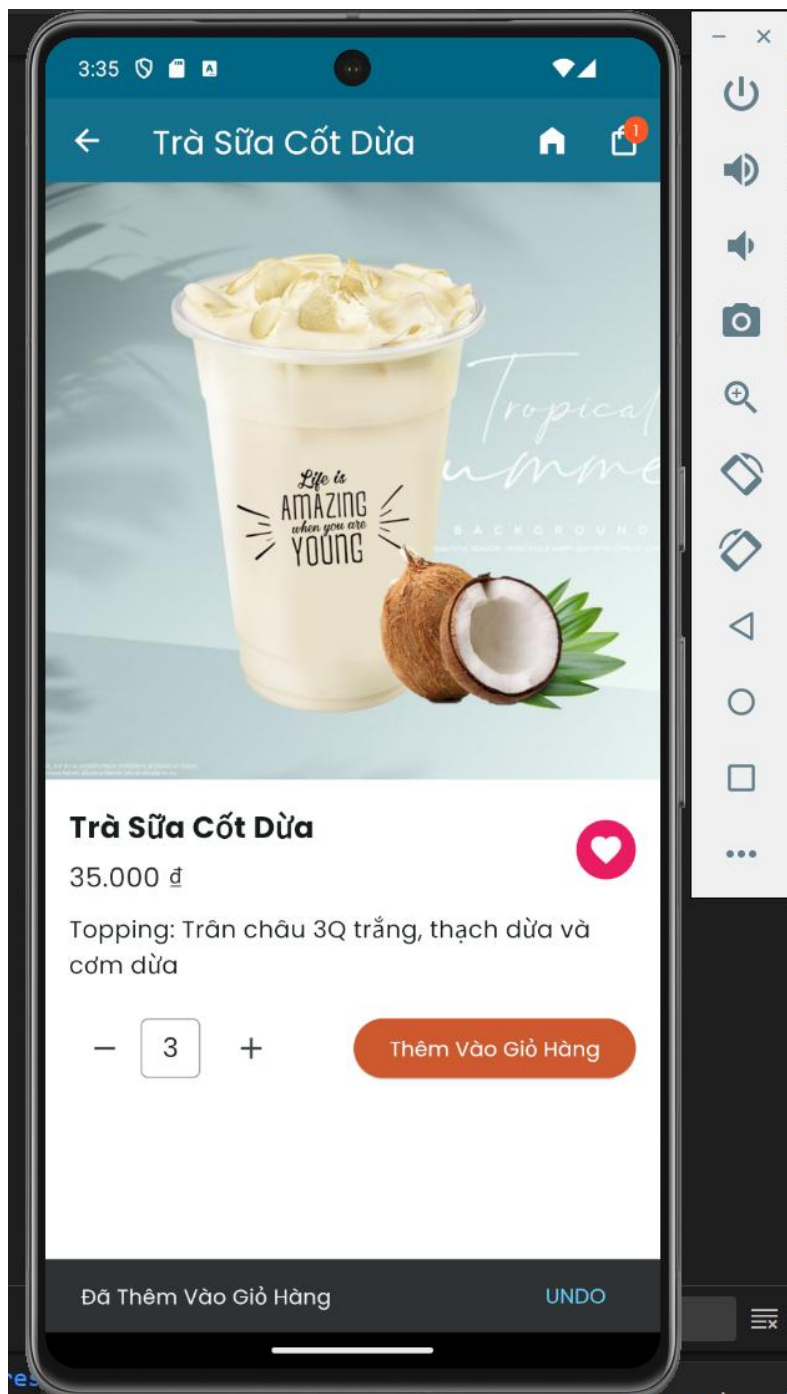
- Positioned: Được sử dụng để xác định vị trí của một widget con trong Stack.
- Container: Được sử dụng để tạo một hộp chứa cho số lượng hoặc biểu tượng, có thể có màu nền và các hình dạng khác nhau.
- BottomNavigationBar: Thanh điều hướng dưới cùng, cho phép người dùng chuyển đổi giữa các màn hình khác nhau.
- BottomNavigationBarItem: Các mục trong thanh điều hướng dưới cùng, mỗi mục có một biểu tượng và một nhãn.
- + Widget đặc biệt:
 - CarouselSlider: Được sử dụng để hiển thị slideshow của các hình ảnh banner quảng cáo.
- + Sử dụng thư viện
 - Provider: Để quản lý trạng thái chia sẻ giữa các widget và giữa các màn hình trong ứng dụng.
 - Carousel_Slider: Được sử dụng để hiển thị banner ảnh dưới dạng Slider
- + Quản lý trạng thái chia sẻ
 - ProductsOverviewScreen sử dụng Provider để truy cập và cập nhật trạng thái của ProductsManager và CartManager.
 - ProductGridTile sử dụng Provider để quản lý trạng thái chia sẻ giữa các widget và giữa các màn hình trong ứng dụng. Ví dụ, khi người dùng nhấn vào biểu tượng yêu thích, ProductGridFooter gọi phương thức toggleFavoriteStatus từ ProductsManager để cập nhật trạng thái yêu thích của sản phẩm.
 - ProductGrid sử dụng Provider để quản lý trạng thái chia sẻ của danh sách sản phẩm, cho phép chức năng hiển thị danh sách sản phẩm yêu thích hoặc tất cả sản phẩm tùy thuộc vào giá trị showFavorites.
- + Đọc hoặc lưu trữ dữ liệu
 - ProductGridTile, ProductGrid đọc dữ liệu sản phẩm từ ProductsManager thông qua Provider và hiển thị chúng trong lưới sản phẩm.

3. Chức năng/giao diện 3: Trang Chi tiết Sản Phẩm

- **Miêu tả chức năng/giao diện:** Giao diện hiển thị chi tiết thông tin sản phẩm, giúp người dùng dễ dàng xem thông tin sản phẩm, cập nhật số lượng và thêm sản phẩm vào giỏ hàng để thực hiện đặt hàng.
- **Ảnh chức năng/giao diện:**



Hình 7: Giao diện “Chi tiết Sản phẩm”



Hình 8: “Đã thêm sản phẩm vào giỏ hàng” tại trang Chi tiết Sản Phẩm

- **Chi tiết cài đặt:**

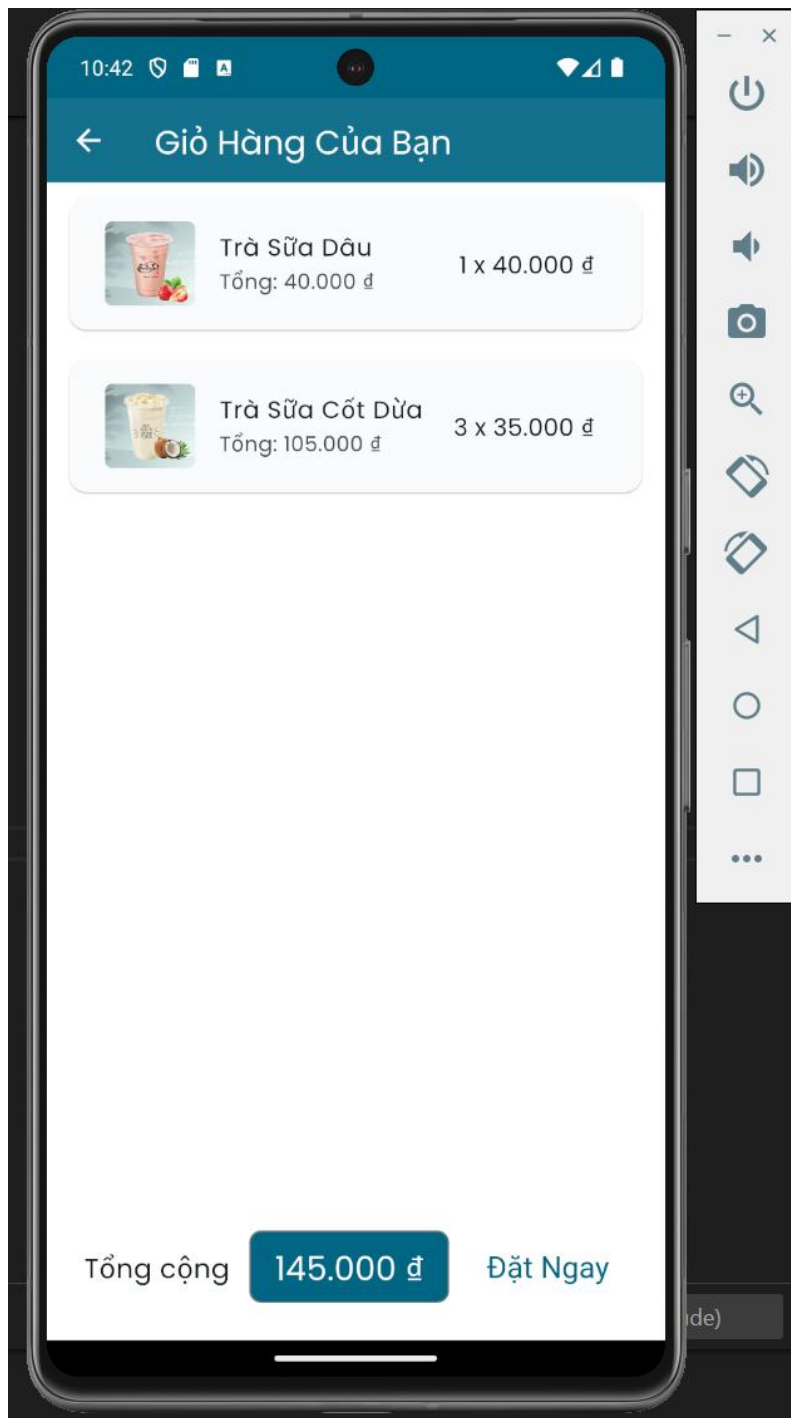
- + Các Widget được sử dụng cho chức năng
 - Scaffold: Đây là widget cơ bản để tạo giao diện chứa các thành phần như AppBar, body...

- AppBar: Hiển thị tiêu đề của sản phẩm và một số actions như nút quay về trang chính và nút giỏ hàng.
 - SingleChildScrollView: Widget này cho phép cuộn nội dung của trang, đảm bảo khả năng cuộn khi nội dung vượt quá màn hình.
 - Column: Sắp xếp các widget theo hướng dọc, chứa tất cả các thành phần trong trang.
 - SizedBox: Widget này tạo ra một hộp rỗng với kích thước cố định, được sử dụng để tạo khoảng cách giữa các widget.
 - Image: Hiển thị hình ảnh của sản phẩm.
 - Padding: Widget này được sử dụng để tạo ra các khu vực đệm xung quanh các widget con.
 - Text: Hiển thị tiêu đề, giá và mô tả của sản phẩm.
 - IconButton: Nút ấn được sử dụng để thêm hoặc xóa sản phẩm khỏi danh sách yêu thích.
 - ElevatedButton: Nút được sử dụng để thêm sản phẩm vào giỏ hàng.
 - ValueListenableBuilder: Widget này được sử dụng để lắng nghe sự thay đổi của một ValueNotifier và cập nhật UI tương ứng.
- + Sử dụng thư viện hay plugin
- Provider: Để quản lý trạng thái chia sẻ giữa các widget và giữa các màn hình trong ứng dụng.
 - Intl: Để định dạng số tiền sang định dạng tiền tệ VND.
- + Quản lý trạng thái chia sẻ
- Provider: Khi trạng thái yêu thích của sản phẩm thay đổi, giá trị của isFavoriteNotifier được cập nhật, từ đó cập nhật giao diện người dùng.
- + Đọc hoặc lưu trữ dữ liệu
- Dữ liệu sản phẩm được truyền vào ProductDetailScreen thông qua constructor khi màn hình được khởi tạo. Thông qua thuộc tính widget.product, màn hình có thể truy cập vào thông tin chi tiết của sản phẩm.
 - Người dùng thực hiện các thao tác như thêm sản phẩm vào giỏ hàng, thông tin về sản phẩm và số lượng được truyền vào CartManager thông qua phương thức addItem(Product product, int quantity).
 - Trong addItem, sản phẩm và số lượng được gửi đến CartManager để thêm vào giỏ hàng. CartManager sẽ xử lý việc thêm sản phẩm này vào danh sách các mặt hàng trong giỏ hàng và thông báo cho người nghe biết về sự thay đổi bằng cách gọi notifyListeners().
 - Khi người dùng thực hiện thao tác xóa sản phẩm khỏi giỏ hàng, thông tin về sản phẩm cần xóa được truyền vào CartManager thông qua phương thức removeItem(String cartItemId). CartManager xóa

sản phẩm tương ứng khỏi danh sách giỏ hàng và thông báo về sự thay đổi cho người nghe bằng cách gọi `notifyListeners()`.

4. Chức năng/giao diện 4: Trang Giỏ Hàng

- **Miêu tả chức năng/giao diện:** Đây là giao diện hiển thị các sản phẩm mà người dùng đã thêm vào giỏ hàng, và người dùng có thể “Đặt ngay” các sản phẩm đó.
- **Ảnh chức năng/giao diện:**



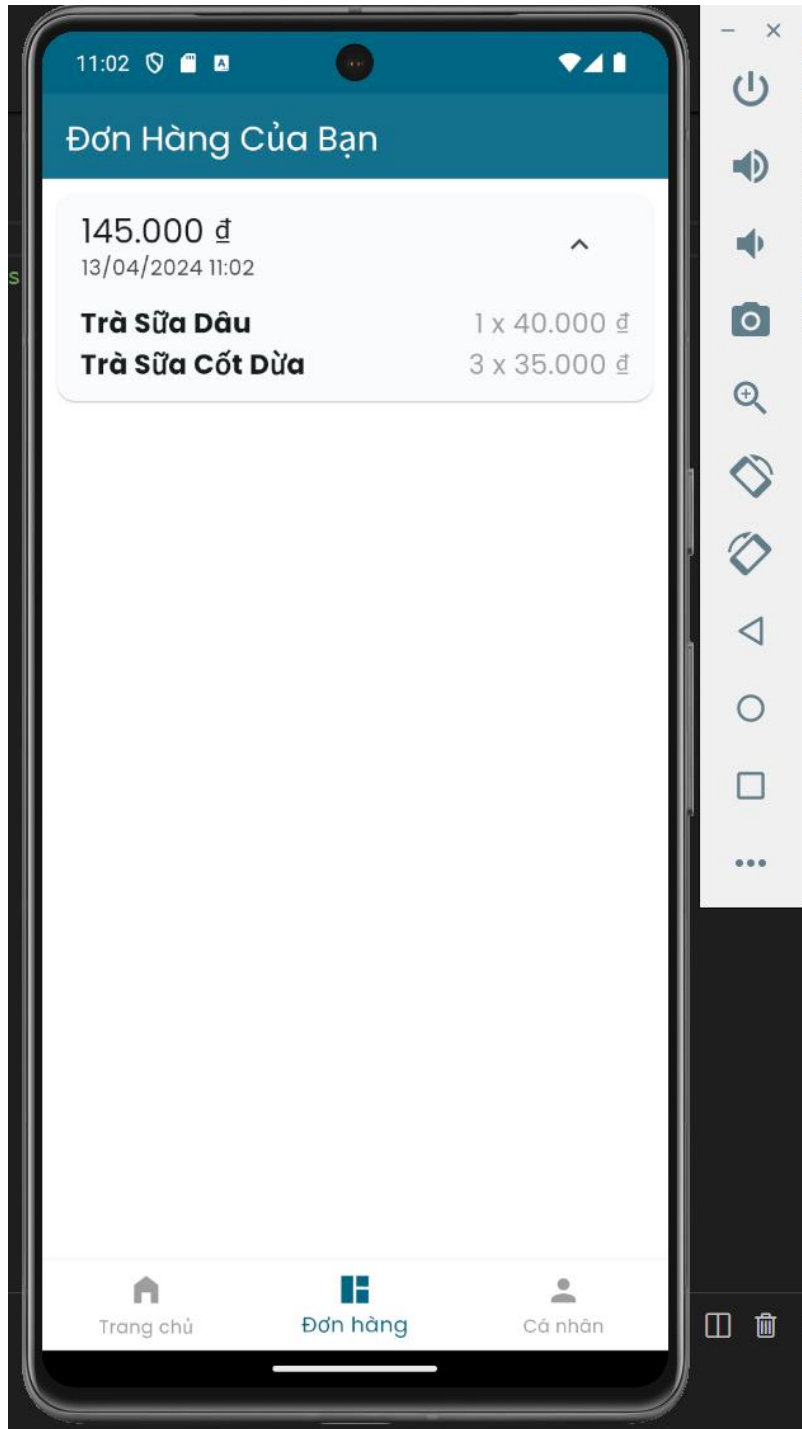
Hình 9: Giao diện “Giỏ hàng”

- **Chi tiết cài đặt:**

- + Các Widget được sử dụng cho chức năng
 - Scaffold: Để tạo layout cơ bản của màn hình.
 - AppBar: Hiển thị tiêu đề của màn hình, hiển thị “Giỏ hàng của bạn”.
 - Column: Để sắp xếp các thành phần theo chiều dọc.
 - Expanded: Để mở rộng không gian của một widget con để lấp đầy không gian còn trống.
 - ListView: Để hiển thị danh sách các mục giỏ hàng.
 - Dismissible: Để tạo khả năng vuốt để xóa các mục giỏ hàng.
 - Card: Để hiển thị thông tin chi tiết của một mục giỏ hàng.
 - ListTile: Để hiển thị thông tin của một mục trong danh sách.
- + Sử dụng thư viện hay plugin
 - Provider: Để quản lý trạng thái chia sẻ giữa các widget và giữa các màn hình trong ứng dụng.
 - Intl: Để định dạng số tiền sang định dạng tiền tệ VND.
- + Quản lý trạng thái chia sẻ
 - Các dữ liệu liên quan đến giỏ hàng được quản lý bởi CartManager, sử dụng provider để chia sẻ dữ liệu giữa các widget trong ứng dụng.
 - Khi người dùng thực hiện thao tác thêm hoặc xóa sản phẩm trong giỏ hàng, CartManager sẽ cập nhật dữ liệu.
- + Đọc hoặc lưu trữ dữ liệu
 - Chức năng này thực hiện đọc và lưu trữ dữ liệu thông qua dịch vụ lưu trữ trực tuyến Firebase Realtime Database, được triển khai trong CartsService.
 - API này cung cấp các phương thức như fetchCarts để lấy danh sách các mục trong giỏ hàng, addCart để thêm một mục mới vào giỏ hàng, updateCart để cập nhật thông tin của một mục trong giỏ hàng, và deleteCart để xóa một mục khỏi giỏ hàng. Dữ liệu được truyền qua và trả về dưới dạng JSON.

5. Chức năng/giao diện 5: Trang Đơn Hàng

- **Miêu tả chức năng/giao diện:** Đây là giao diện hiển thị các đơn hàng mà người dùng đã đặt mua.
- **Ảnh chức năng/giao diện:**



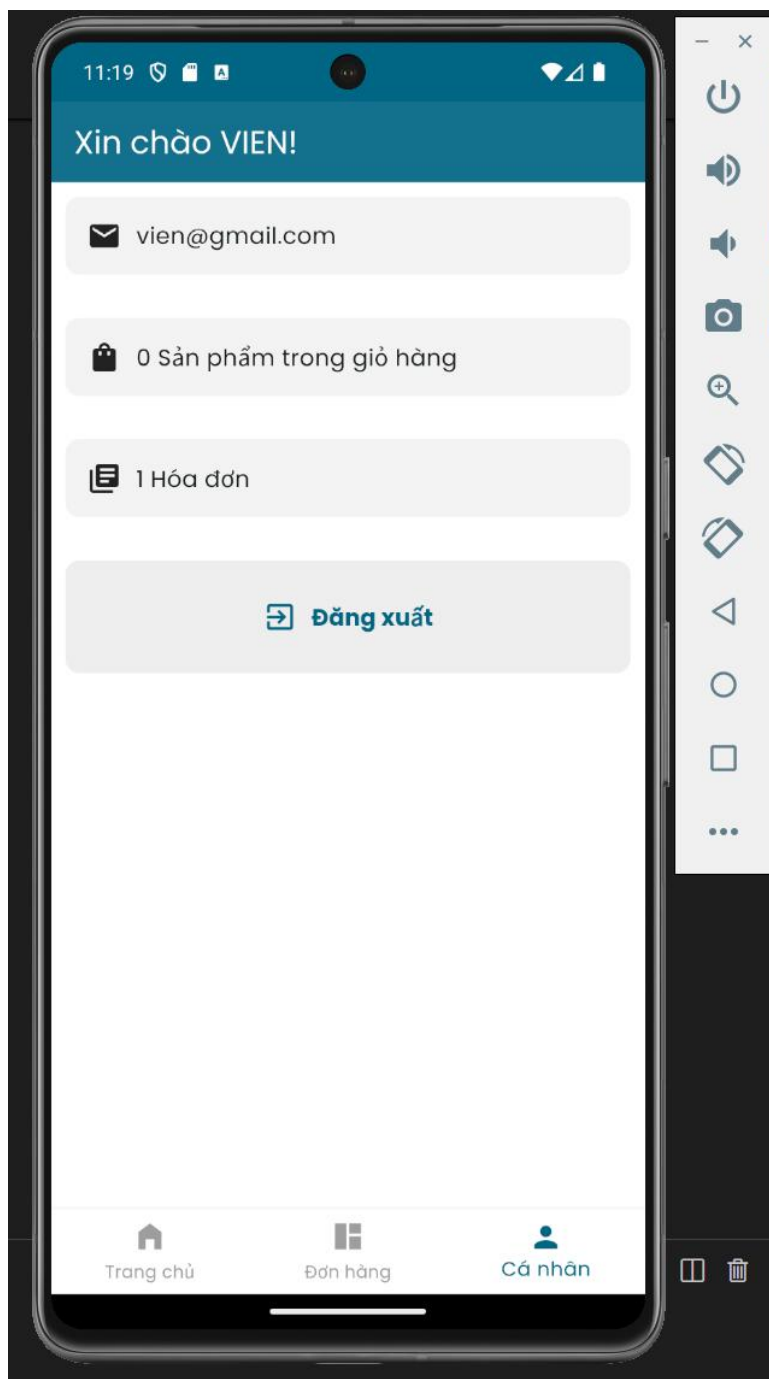
Hình 10: Giao diện “Đơn hàng”

- **Chi tiết cài đặt:**
 - + Các Widget được sử dụng cho chức năng
 - Scaffold: Được sử dụng để tạo layout cơ bản của màn hình, bao gồm app bar và body.

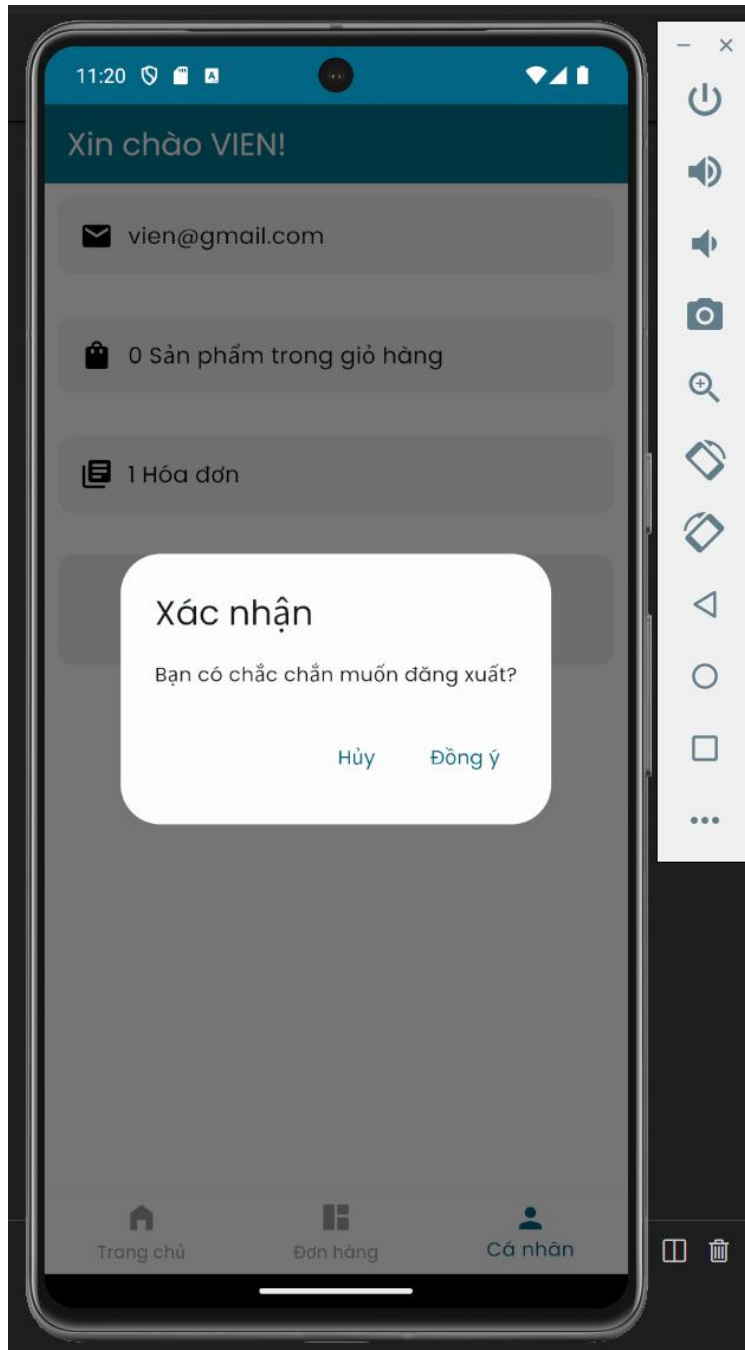
- AppBar: Thanh tiêu đề của màn hình, hiển thị tên "Đơn Hàng Của Bạn".
- Card: Được sử dụng để tạo một card chứa thông tin của mỗi đơn hàng.
- Column: Sử dụng để sắp xếp các widget con dọc theo trục chính.
- OrderSummary: Widget này hiển thị tổng số tiền của mỗi đơn hàng cùng với ngày và giờ đặt hàng.
- OrderItemList: Widget này hiển thị danh sách các sản phẩm trong mỗi đơn hàng, chỉ hiển thị khi đơn hàng được mở rộng.
- Consumer<OrdersManager>: Được sử dụng để lắng nghe thay đổi trong OrdersManager và cập nhật giao diện khi có sự thay đổi.
- ListView.builder: Widget này được sử dụng để hiển thị danh sách các đơn hàng trong OrdersManager, mỗi đơn hàng được hiển thị bằng một OrderItemCard.
- + Sử dụng thư viện hay plugin
 - Provider: Để quản lý trạng thái chia sẻ giữa các widget và giữa các màn hình trong ứng dụng.
 - Intl: Để định dạng số tiền sang định dạng tiền tệ VND.
- + Quản lý trạng thái chia sẻ
 - Provider được sử dụng để quản lý trạng thái của đơn hàng thông qua OrdersManager, cung cấp dữ liệu đơn hàng cho widget.
- + Đọc hoặc lưu trữ dữ liệu:
 - Chức năng này thực hiện đọc và lưu trữ dữ liệu thông qua dịch vụ lưu trữ trực tuyến Firebase Realtime Database, được triển khai trong OrdersService. Dữ liệu được lưu trữ dưới dạng JSON, trong đó mỗi đơn hàng được lưu trữ như một đối tượng JSON, bao gồm thông tin về số tiền, danh sách sản phẩm và ngày giờ đặt hàng.
 - fetchOrders(): Được sử dụng để lấy danh sách các đơn hàng từ cơ sở dữ liệu Firebase Realtime Database thông qua một HTTP request.
 - addOrder(OrderItem order): Được sử dụng để thêm một đơn hàng mới vào cơ sở dữ liệu Firebase Realtime Database thông qua một HTTP request.

6. Chức năng/giao diện 6: Trang Cá Nhân

- **Miêu tả chức năng/giao diện:** Giao diện hiển thị thông tin người dùng như email, số sản phẩm trong giỏ hàng, số hóa đơn và đăng xuất khỏi ứng dụng.
- **Ảnh chức năng/giao diện:**



Hình 11: Giao diện trang “Cá Nhân”



Hình 12: Người dùng “Đăng xuất”

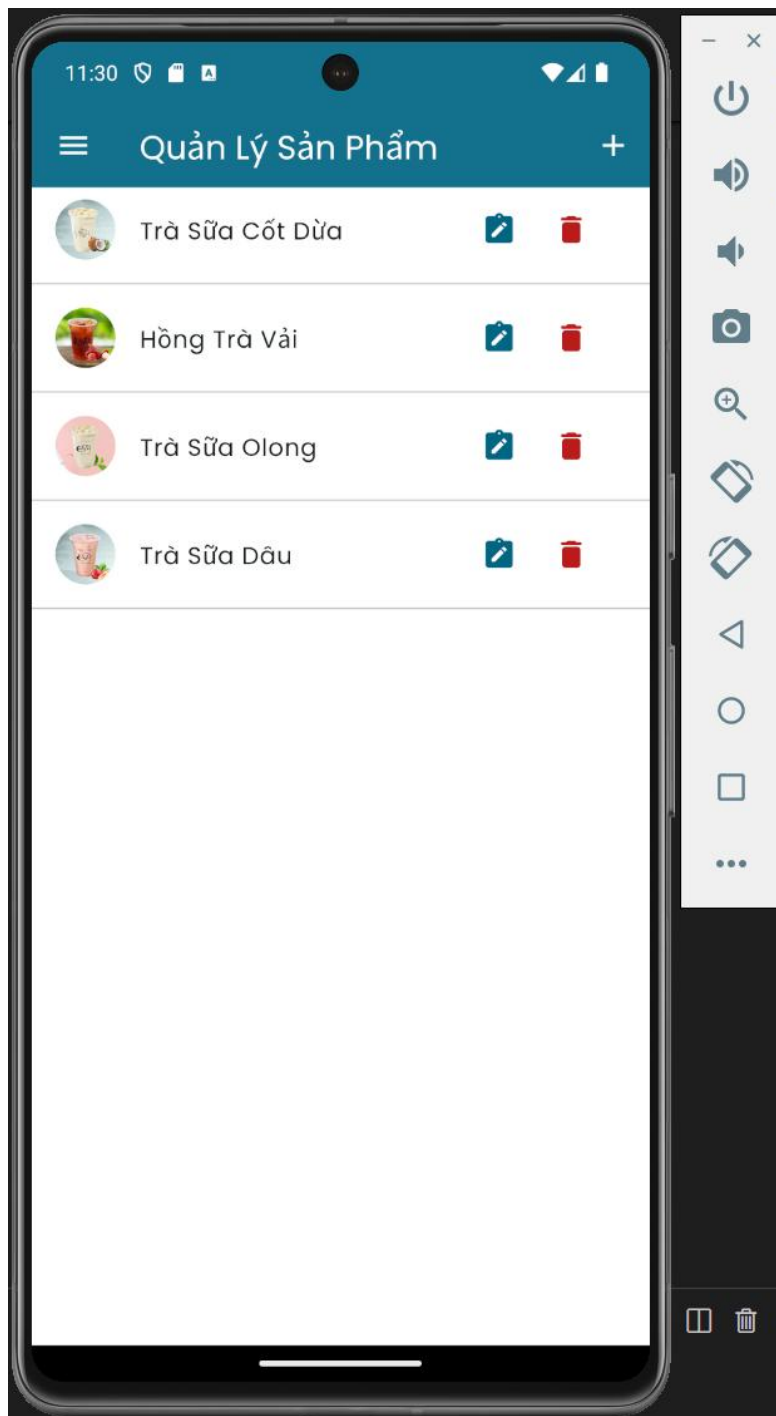
- **Chi tiết cài đặt:**

- + Các Widget được sử dụng cho chức năng
 - Scaffold: Định nghĩa layout cơ bản của trang.
 - AppBar: Thanh tiêu đề của trang, chứa tên người dùng.
 - ListView: Widget để hiển thị danh sách các thông tin cá nhân và nút đăng xuất.

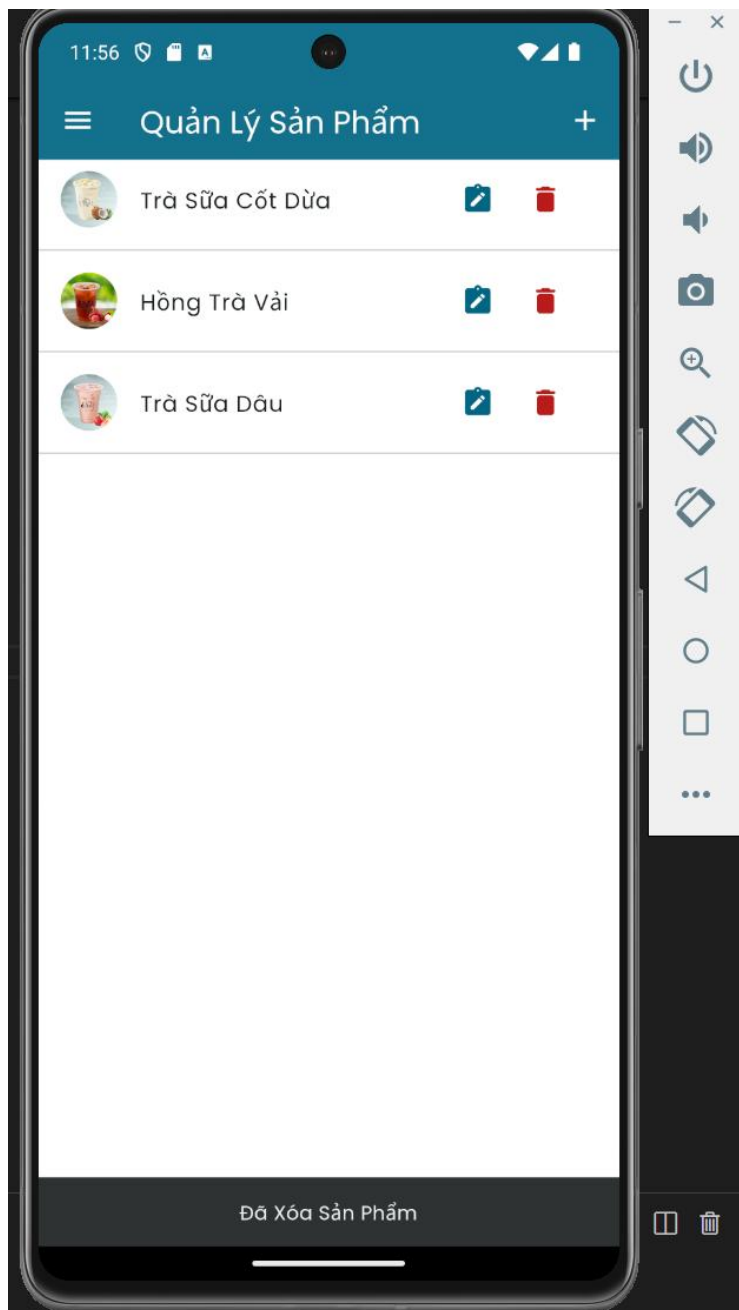
- Container: Để tạo khung cho mỗi phần thông tin cá nhân và nút đăng xuất.
- TextButton: Để tạo nút "Đăng xuất".
- AlertDialog: Hiện thị hộp thoại xác nhận khi người dùng muốn đăng xuất.
- + Sử dụng thư viện hay plugin
 - Provider: Để quản lý trạng thái chia sẻ giữa các widget và giữa các màn hình trong ứng dụng.
- + Quản lý trạng thái chia sẻ
 - Sử dụng Provider để quản lý trạng thái chia sẻ. Trong ProfileScreen, các dữ liệu như thông tin người dùng, sản phẩm trong giỏ hàng và đơn hàng được lấy thông qua context.read<...>()
- + Đọc hoặc lưu trữ dữ liệu:
 - Không có thao tác đọc hoặc lưu trữ dữ liệu trực tiếp trong ProfileScreen. Tuy nhiên, thông qua các provider (AuthManager, ProductsManager, CartManager, OrdersManager), dữ liệu được lấy từ các dịch vụ lưu trữ trực tuyến như Firebase Realtime Database.

7. Chức năng/giao diện 7: Trang Quản Lý Sản Phẩm Của Quản Trị Viên

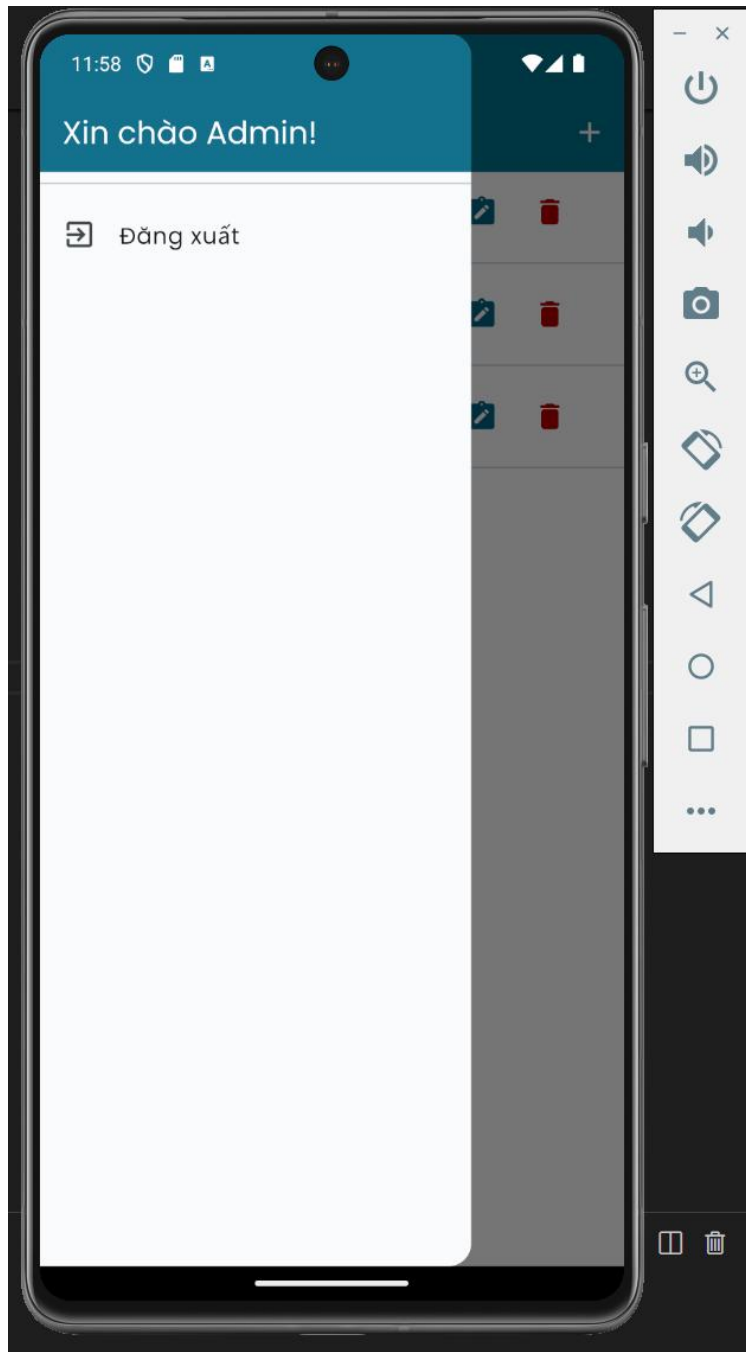
- **Miêu tả chức năng/giao diện:** Giao diện cho phép người dùng Quản Trị Viên thực hiện thao tác Thêm, Sửa, Xóa Sản Phẩm
- **Ảnh chức năng/giao diện:**



Hình 13: Giao diện trang “Quản lý sản phẩm” của Quản Trị Viên



Hình 14: Xóa Sản Phẩm



Hình 15: AppDrawer cho phép Admin “Đăng xuất” khỏi ứng dụng

- **Chi tiết cài đặt:**

- + Các Widget được sử dụng cho chức năng
 - Scaffold: Định nghĩa layout cơ bản của trang và mỗi item sản phẩm.
 - AppBar: Thanh tiêu đề của trang, chứa tiêu đề trang và nút thêm sản phẩm.
 - FloatingActionButton: Widget để xây dựng UI dựa trên một Future.

- RefreshIndicator: Widget để tạo hiệu ứng làm mới khi kéo xuống.
- ListView và ListView.builder: Widget để hiển thị danh sách sản phẩm và danh sách có thể tạo được từ một danh sách sản phẩm.
- Column: Để xếp các widget theo chiều dọc.
- ListTile: Widget để hiển thị một hàng trong danh sách, chứa thông tin sản phẩm và các nút chức năng sửa và xóa sản phẩm.
- CircleAvatar: Widget để hiển thị hình ảnh tròn, sử dụng để hiển thị hình ảnh sản phẩm.
- IconButton: Nút hình ảnh, được sử dụng cho các nút chức năng như xóa và chỉnh sửa sản phẩm.
- AppDrawer: Tạo menu trượt, trong đó có chức năng cho người Admin đăng xuất khỏi ứng dụng
- + Sử dụng thư viện hay plugin
 - Provider: Để quản lý trạng thái chia sẻ giữa các widget và giữa các màn hình trong ứng dụng.
- + Quản lý trạng thái chia sẻ:
 - Sử dụng Provider để quản lý trạng thái chia sẻ. AdminProductsScreen và AdminProductListTile đều sử dụng Consumer để lấy dữ liệu từ ProductsManager. Khi có thay đổi trong dữ liệu, ProductsManager sẽ thông báo cho các widget con để cập nhật giao diện.
- + Đọc hoặc lưu trữ dữ liệu:
 - Sử dụng dịch vụ lưu trữ trực tuyến thông qua ProductsManager. Dữ liệu sản phẩm được lưu trữ trong Firebase Realtime Database dưới dạng JSON. Khi cần, ProductsManager sẽ gọi các phương thức để lấy danh sách sản phẩm hoặc thực hiện các thao tác như xóa sản phẩm.

8. Chức năng/giao diện 8: Trang Thêm/Chỉnh Sửa Sản Phẩm

- **Miêu tả chức năng/giao diện:** Đây là giao diện để người dùng Quản Trị Viên Thêm và Chỉnh Sửa Sản Phẩm.
- **Ảnh chức năng/giao diện:**

The image shows a mobile application interface for managing product information. The screen is titled "Thông Tin Sản Phẩm" (Product Information) with a back arrow on the left and a save icon on the right. The form contains several input fields: a large image upload area labeled "Nhập URL" (Enter URL), a "URL Hình Ảnh" (Image URL) field, a "Tên Sản Phẩm" (Product Name) field, a "Giá" (Price) field with a default value of "0 đ", and a "Mô tả" (Description) field. To the right of the phone screen is a vertical toolbar with icons for power, volume, camera, zoom, and other functions. At the bottom right, there are icons for a list and a trash can.

12:00

← Thông Tin Sản Phẩm

Nhập URL

URL Hình Ảnh

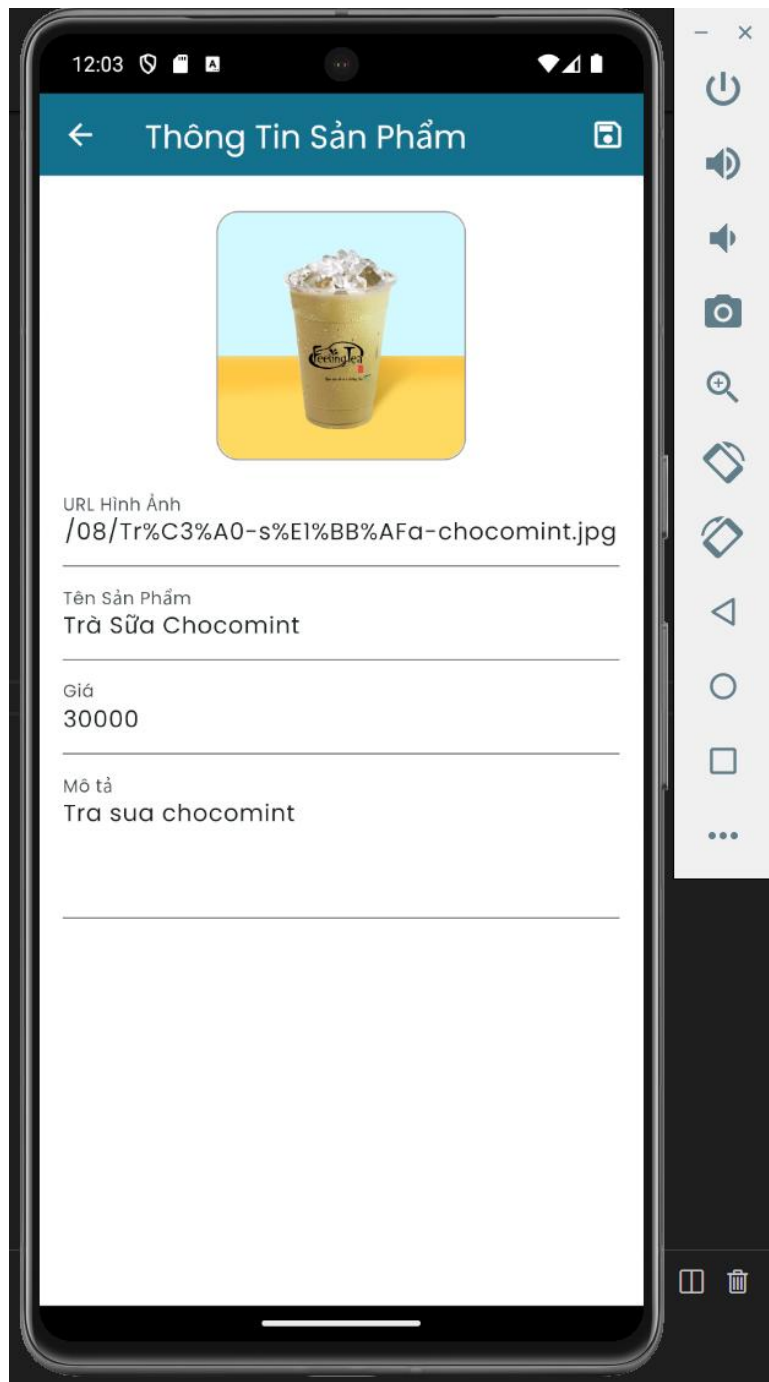
Tên Sản Phẩm

Giá
0 đ

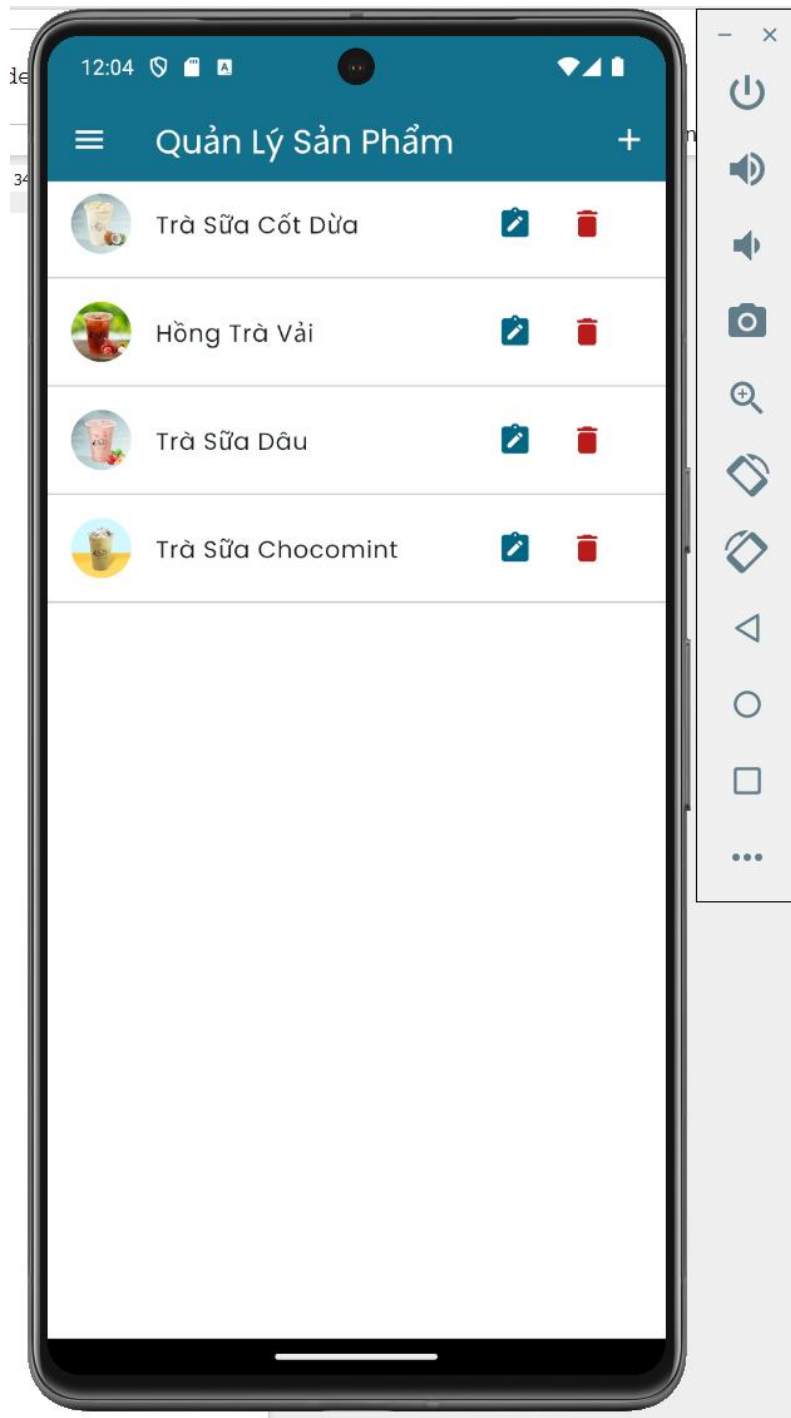
Mô tả

☰ ☒

Hình 16: Giao diện Thêm/Chỉnh sửa sản phẩm



Hình 17: Thêm một Sản Phẩm



Hình 18: Đã thêm Sản phẩm

- **Chi tiết cài đặt:**
 - + Các Widget được sử dụng cho chức năng
 - Scaffold: Định nghĩa layout cơ bản của trang.

- AppBar: Thanh tiêu đề của trang, chứa tiêu đề trang và nút lưu.
 - CircularProgressIndicator: Widget để hiển thị vòng tròn chứa hình ảnh
 - Padding: Để tạo các khoảng cách xung quanh nội dung.
 - Form: Widget để quản lý và xác thực dữ liệu nhập vào.
 - ListView: Widget để hiển thị danh sách các trường nhập liệu.
 - Column: Để xếp các widget theo chiều dọc.
 - TextFormField: Widget để nhập liệu văn bản.
 - IconButton: Nút hình ảnh, được sử dụng cho nút lưu.
- + Sử dụng thư viện hay plugin
- Provider: Để quản lý trạng thái chia sẻ giữa các widget và giữa các màn hình trong ứng dụng.
- + Quản lý trạng thái chia sẻ:
- Sử dụng Provider để quản lý trạng thái chia sẻ. Dữ liệu sản phẩm được quản lý trong ProductsManager, và thông qua `context.read<ProductsManager>()`, trạng thái của ProductsManager được sử dụng và cập nhật trong EditProductScreen. Khi người dùng chỉnh sửa hoặc thêm mới sản phẩm, dữ liệu được cập nhật trong ProductsManager, và UI sẽ được cập nhật tương ứng thông qua Consumer.
- + Đọc hoặc lưu trữ dữ liệu:
- Thực hiện cả đọc và lưu trữ dữ liệu thông qua ProductsManager. Dữ liệu sản phẩm có thể được lưu trữ thông qua dịch vụ lưu trữ trực tuyến Firebase Realtime Database. Dữ liệu sản phẩm được biểu diễn dưới dạng các đối tượng Product, và thông qua các phương thức trong ProductsManager, dữ liệu có thể được lấy, cập nhật hoặc thêm mới.