BÁO CÁO DỰ ÁN CUỐI KỲ

HỌC KỲ , NĂM HỌC 2022-2023

Tên dự án: Ứng Dụng Đặt Đồ Ăn

MSSV: B2014850

Họ tên SV: Lê Phước Lợi

Lớp học phần: CT48402

Link GitHub mã nguồn: https://github.com/23-24Sem1-Courses/ct48402-project-LePhuocLoi4850

I. Tổng quan

Ứng dụng đặt đồ ăn là một ứng dụng di động giúp người dùng đặt đồ ăn một cách đa dạng và nhiều sự lựa chọn. Ứng dụng này cho phép người dùng xem thực đơn, đặt hàng và thanh toán trực tuyến. Sau đó, đồ ăn sẽ được giao tận nơi bởi các shipper. Ứng dụng này cung cấp các chức năng cơ bản cho người dùng cũng như người bán. Với giao diện đơn giản và thẩm mĩ giúp trải nghiệm người dùng tốt hơn.

II. Chi tiết các chức năng

1. Chức năng/giao diện 1 : Giao diện đăng ký và đăng nhập

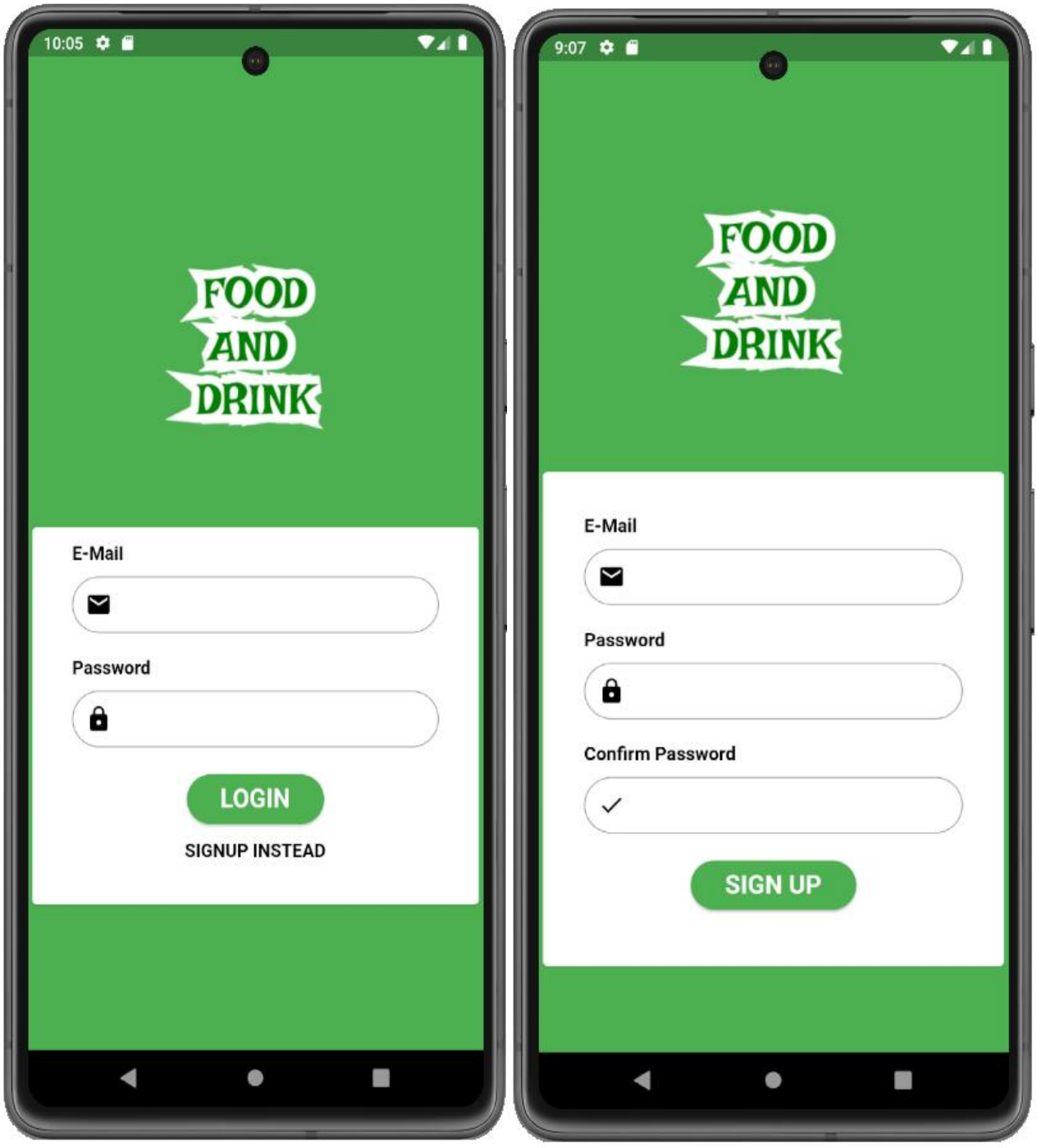
Miêu tả chức năng/giao diện:

Giao diện đăng ký giúp người dùng đăng ký tài khoản khi chưa có tài khoản

Giao diện đăng nhập giúp người dùng có tài khoản đăng nhập vào hệ thống

Giao diện có sự chuyển đổi linh hoạt giữa đăng ký và đăng nhập, tạo sự thuận tiện trong thao tác

Ảnh chức năng/giao diện:



Chi tiết cài đặt:

Các widget sử dụng cho chức năng/giao diện này.

Scaffold,
Stack,
SingleChildScrollView,
SizedBox,
Flexible,
AppBanner,
AuthCard,
Card,
Container,
DecorationImage,
image,
EdgeInsets.symmetric,
Image.asset,
Form,
SingleChildScrollView,
Column,
TextFormField,
ValueListenableBuilder<bool>,
TextButton,
ElevatedButton,
Icon,
Text.

Các thư viện và plugin có trong giao diện này

Flutter Material Package

- Thư viện/plugin: package:flutter/material.dart
- Vai trò: Cung cấp các widget và công cụ để xây dựng giao diện người dùng theo phong cách Material Design của Google. Đây là một phần quan trọng của Flutter và được sử dụng để tạo ra các thành phần giao diện như Scaffold, Container, Stack, SingleChildScrollView, Column, SizedBox, Flexible, ElevatedButton, TextButton, TextFormField, và nhiều widget khác.

Provider Package

- Thư viện/plugin: package:provider/provider.dart
- Vai trò: Cung cấp mô hình quản lý trạng thái và dependency injection trong Flutter. Trong các ví dụ trước đó, nó được sử dụng để quản lý trạng thái của đối tượng AuthManager, giúp truy cập và quản lý trạng thái đăng nhập và đăng ký từ nhiều nơi trong ứng dụng.

Assets Image

- Thư viện/plugin: Không cần import trực tiếp, nhưng sử dụng hình ảnh từ thư mục assets/image/
- Vai trò: Cung cấp hình ảnh để làm nền cho Container và DecorationImage trong widget AppBanner. Hình ảnh được sử dụng để hiển thị logo của ứng dụng.

Chức năng có sử dụng giải pháp quản lý trạng thái chia sẻ thông qua thư viện Provider. Dưới đây là miêu tả chi tiết

AuthManager Class

- File: auth_manager.dart
- Vai trò: Là một lớp quản lý trạng thái đăng nhập và đăng ký.
- Miêu tả: AuthManager có thể được sử dụng để thực hiện các hành động đăng nhập và đăng ký . Nó lưu trữ trạng thái đăng nhập thông qua một biến boolean, có thể được đọc và cập nhật từ các thành phần khác trong ứng dụng bằng cách sử dụng Provider.

Provider

- File: auth_card.dart và auth_screen.dart
- Vai trò: Cung cấp truy cập đến AuthManager từ các widget con mà không cần truyền qua nhiều cấp widget cha.
- Miêu tả: Trong auth_card.dart, sử dụng context.read<AuthManager>() để đọc trạng thái đăng nhập và đăng ký từ AuthManager. Điều này giúp widget có thể tương tác với trạng thái đăng nhập mà không cần truyền giá trị qua nhiều cấp widget cha.

ValueNotifier

- File: auth_card.dart
- Vai trò: Quản lý trạng thái cục bộ của widget.
- Miêu tả: Sử dụng ValueNotifier<bool> để theo dõi trạng thái của quá trình đăng nhập và đăng ký. ValueListenableBuilder được sử dụng để xây dựng UI dựa trên giá trị

của ValueNotifier, đồng thời hiển thị CircularProgressIndicator khi đang trong quá trình submitting.

Consumer

- File: auth_screen.dart
- Vai trò: Cập nhật giao diện người dùng dựa trên thay đổi trong AuthManager.
- Miêu tả: Trong auth_screen.dart, sử dụng Consumer để bao bọc AppBanner và AuthCard. Khi trạng thái của AuthManager thay đổi, chỉ những phần được bao bọc bởi Consumer sẽ được rebuild, giúp tối ưu hóa hiệu suất.

Chức năng này có thực hiện lưu trữ dữ liệu. dùng dịch vụ lưu trữ FireBase.

Đăng ký:

Phương thức yêu cầu: POST

Endpoint: https://identitytoolkit.googleapis.com/v1/accounts:signUp?key=[API_KEY] , với API_KEY là "Web API Key" lấy từ dự án firebase

Đăng nhập:

Phương thức yêu cầu: POST

Endpoint: https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?key=[API_KEY] , với API_KEY là "Web API Key" lấy từ dự án firebase

2. **Chức năng/giao diện 2:** Giao diện trang chủ

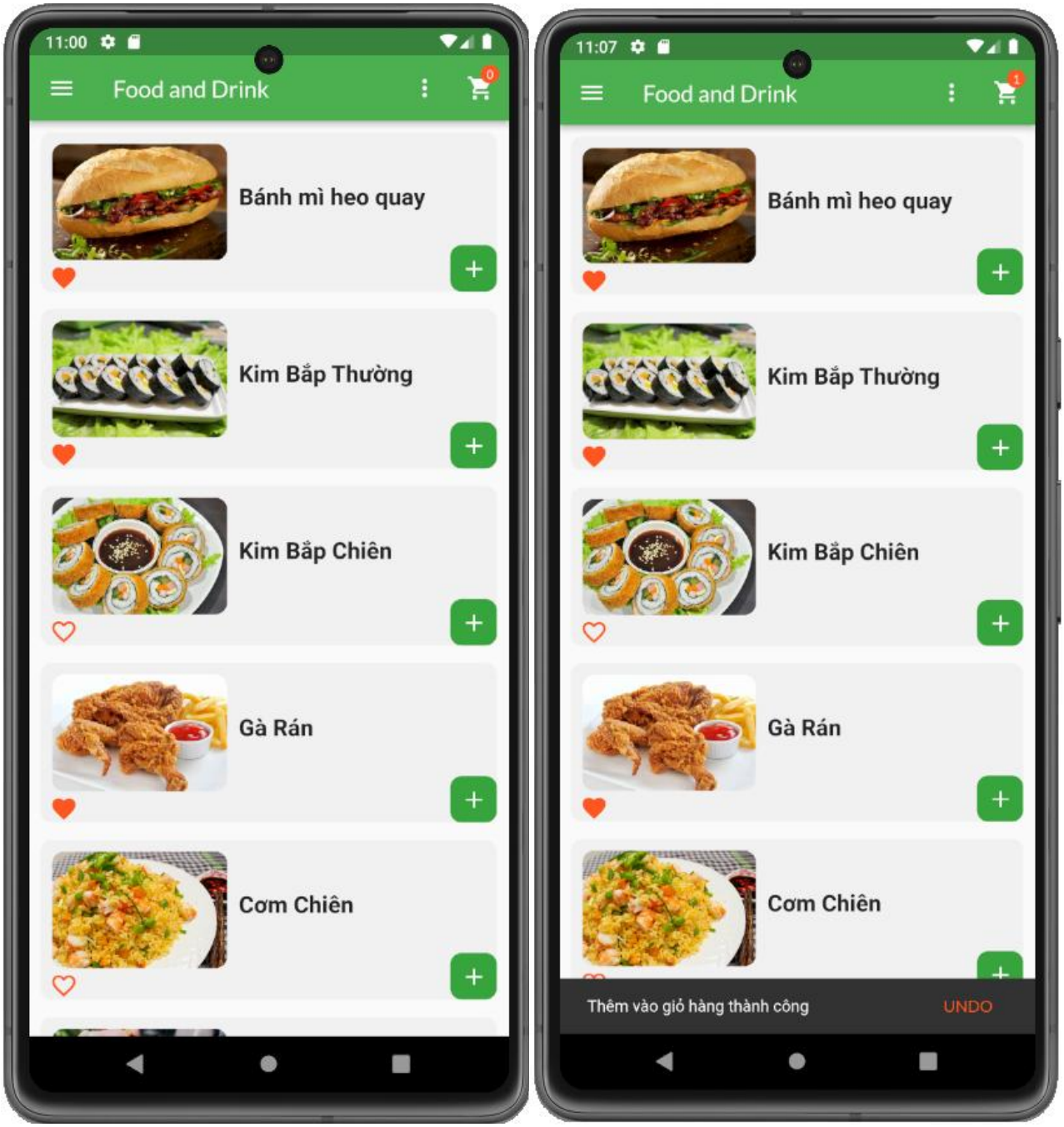
Miêu tả chức năng/giao diện:

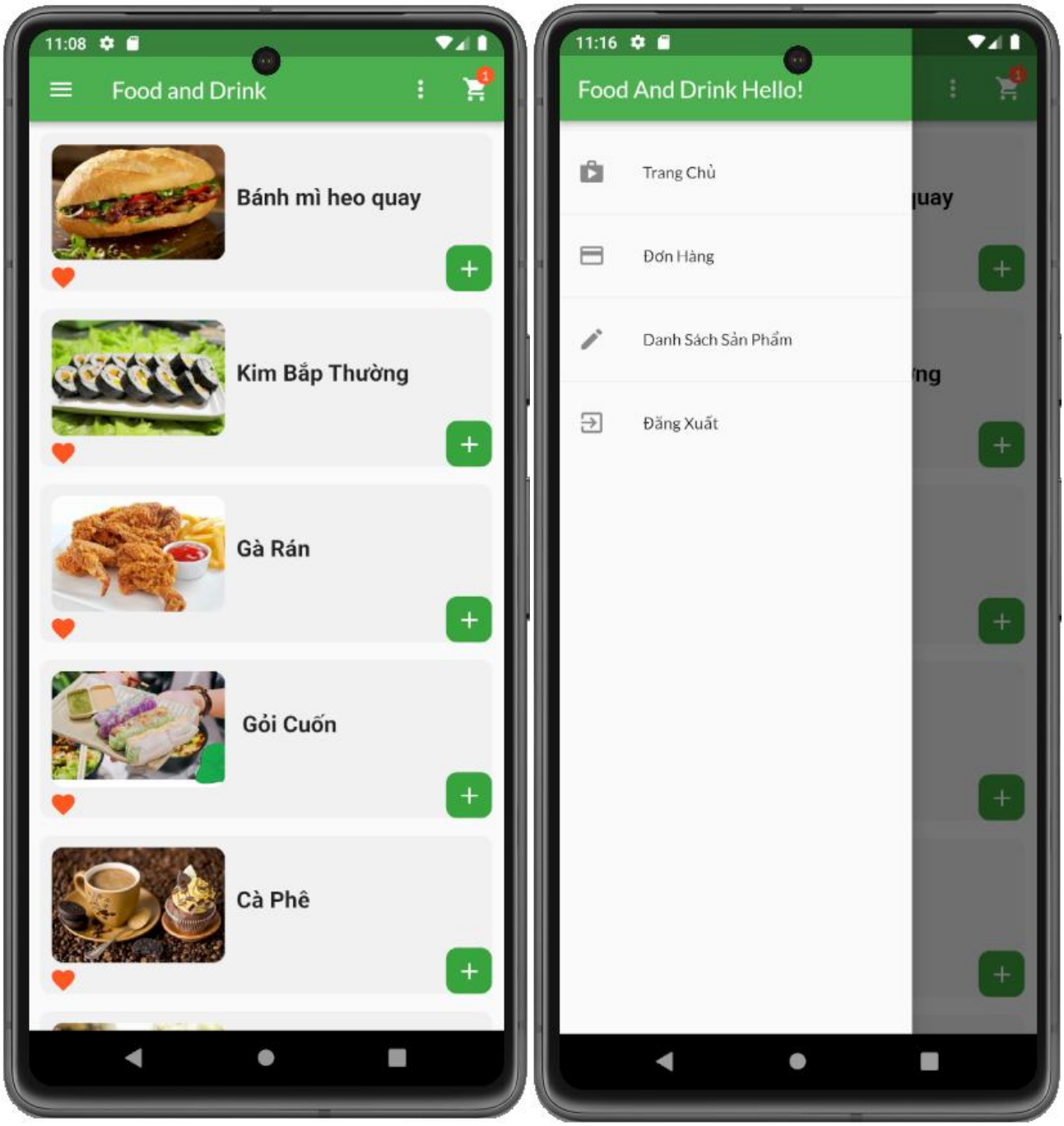
Là giao diện chính khi người dùng đăng nhập vào và hiển thị tất cả sản phẩm

Giao diện có các chức năng như hiển thị tất cả sản phẩm, sản phẩm yêu thích, thích sản phẩm, thêm sản phẩm vào giỏ hàng, menu để di chuyển đến các trang khác.

Giao diện còn cho phép người dùng xem được chi tiết sản phẩm khi chạm vào chúng và hiển thị được số sản phẩm được thêm vào giỏ hàng.

Ảnh chức năng/giao diện:




11:08
Food and Drink
Bánh mì heo quay
Kim Bắp Thường
Gà Rán
Gỏi Cuốn
Cà Phê
11:16
Food And Drink Hello!
Trang Chủ
Đơn Hàng
Danh Sách Sản Phẩm
Đăng Xuất

Chi tiết cài đặt:

Các widget sử dụng cho chức năng/giao diện này.

Scaffold,
AppBar,
ValueListenableBuilder<bool>,
FutureBuilder,
ProductsGrid,
AppDrawer,
Consumer<CartManager>,
TopRightBadge,
IconButton,
PopupMenuButton,
PopupMenuItem,
List<Product>,
Icon,
CircularProgressIndicator,
SingleChildScrollView,
Column,
Tex,
Image,
SliverGridDelegateWithFixedCr
ossAxisCount,
ProductGridTile,
Provider,
ProductsManager,

Các thư viện và plugin có trong giao diện này

Flutter Material Package

- Thư viện/plugin: package:flutter/material.dart
- Vai trò: Cung cấp các widget và công cụ để xây dựng giao diện người dùng theo phong cách Material Design của Google. Đây là một phần quan trọng của Flutter và được sử dụng để tạo ra các thành phần giao diện như Scaffold, Container, Stack, SingleChildScrollView, Column, SizedBox, Flexible, ElevatedButton, TextButton, TextFormField, và nhiều widget khác.

Provider Package

- Thư viện/plugin: package:provider/provider.dart
- Vai trò: Cung cấp mô hình quản lý trạng thái và dependency injection trong Flutter.

Chức năng có sử dụng giải pháp quản lý trạng thái chia sẻ thông qua thư viện Provider. Dưới đây là miêu tả chi tiết

ProductsManager (Quản lý Sản phẩm)

- Vai trò: Là một lớp mô hình quản lý trạng thái của danh sách sản phẩm.
- Sử dụng ChangeNotifier : Mở rộng từ ChangeNotifier để có khả năng thông báo sự thay đổi trong trạng thái, giúp cập nhật giao diện người dùng khi có sự thay đổi.

ProductsGrid (Lưới Sản phẩm):

- Vai trò: Widget hiển thị lưới sản phẩm dựa trên trạng thái được quản lý bởi ProductsManager .
- Sử dụng Provider : Sử dụng context.select để lấy danh sách sản phẩm từ ProductsManager . Cập nhật giao diện khi danh sách sản phẩm thay đổi.

ProductGridTile (Ô lưới Sản phẩm):

- Vai trò: Widget hiển thị thông tin chi tiết của một sản phẩm trong lưới sản phẩm.
- Sử dụng Consumer : Sử dụng Consumer để lắng nghe thay đổi trong CartManager và cập nhật badge khi có sự thay đổi.

CartManager (Quản lý Giỏ hàng):

- Vai trò: Là một lớp mô hình quản lý trạng thái của giỏ hàng.
- Sử dụng ChangeNotifier : Mở rộng từ ChangeNotifier để có khả năng thông báo sự thay đổi trong trạng thái và cập nhật giao diện người dùng.

TopRightBadge (Badge ở góc trên bên phải):

- Vai trò: Widget hiển thị badge ở góc trên bên phải của một widget khác.
- Sử dụng Stack , Positioned : Sử dụng Stack và Positioned để overlay badge lên trên widget khác. Badge hiển thị số lượng sản phẩm trong giỏ hàng.

ProductDetailScreen (Màn hình Chi tiết Sản phẩm):

- Vai trò: Màn hình hiển thị chi tiết của một sản phẩm.
- Sử dụng Provider : Sử dụng Provider để lấy thông tin chi tiết sản phẩm từ ProductsManager .

AppDrawer (Thanh điều hướng bên):

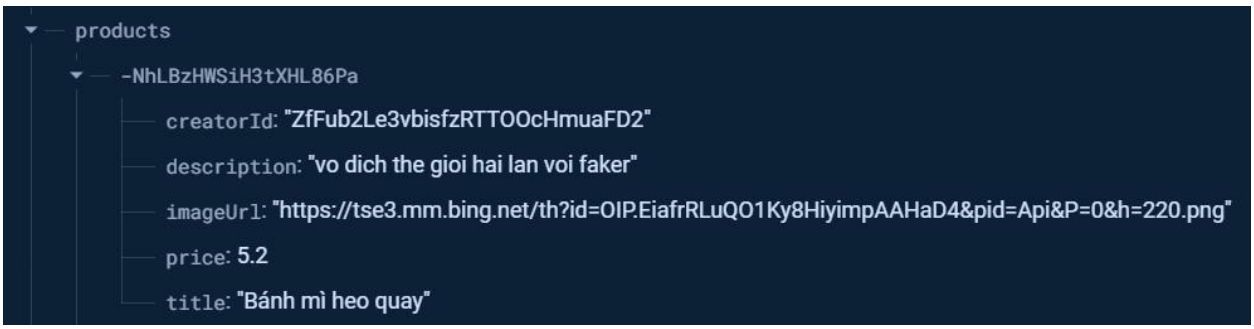
- Vai trò: Widget hiển thị thanh điều hướng bên.
- Sử dụng Provider : Sử dụng Provider để lấy thông tin người dùng hoặc các thông tin khác cần hiển thị trong thanh điều hướng.

ProductService (Dịch vụ Sản phẩm):

- Vai trò: Tương tác với backend để thực hiện các thao tác như lấy dữ liệu sản phẩm, thêm, cập nhật và xóa sản phẩm.
- Sử dụng AuthToken : Sử dụng AuthToken để đảm bảo xác thực khi gửi các yêu cầu đến backend.

Chức năng này có thực hiện đọc dữ liệu. dùng dịch vụ lưu trữ FireBase.

GET /products.json: đọc tất cả các đối tượng từ nhánh products

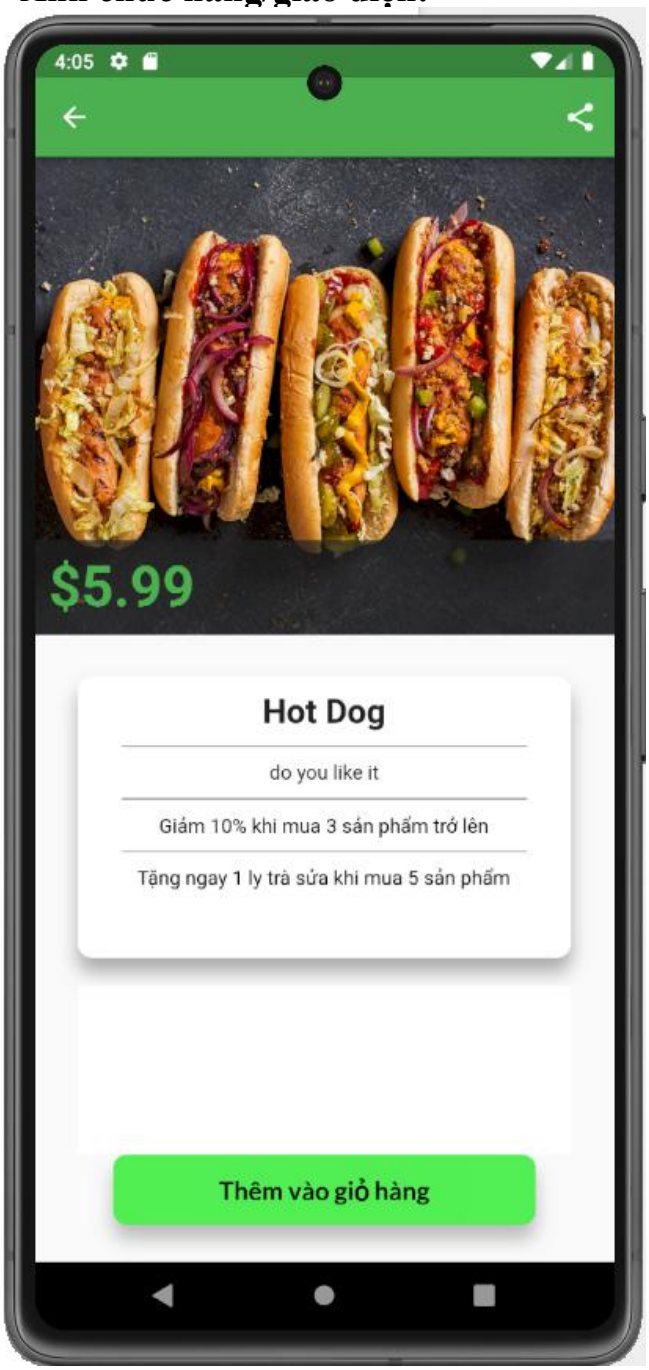


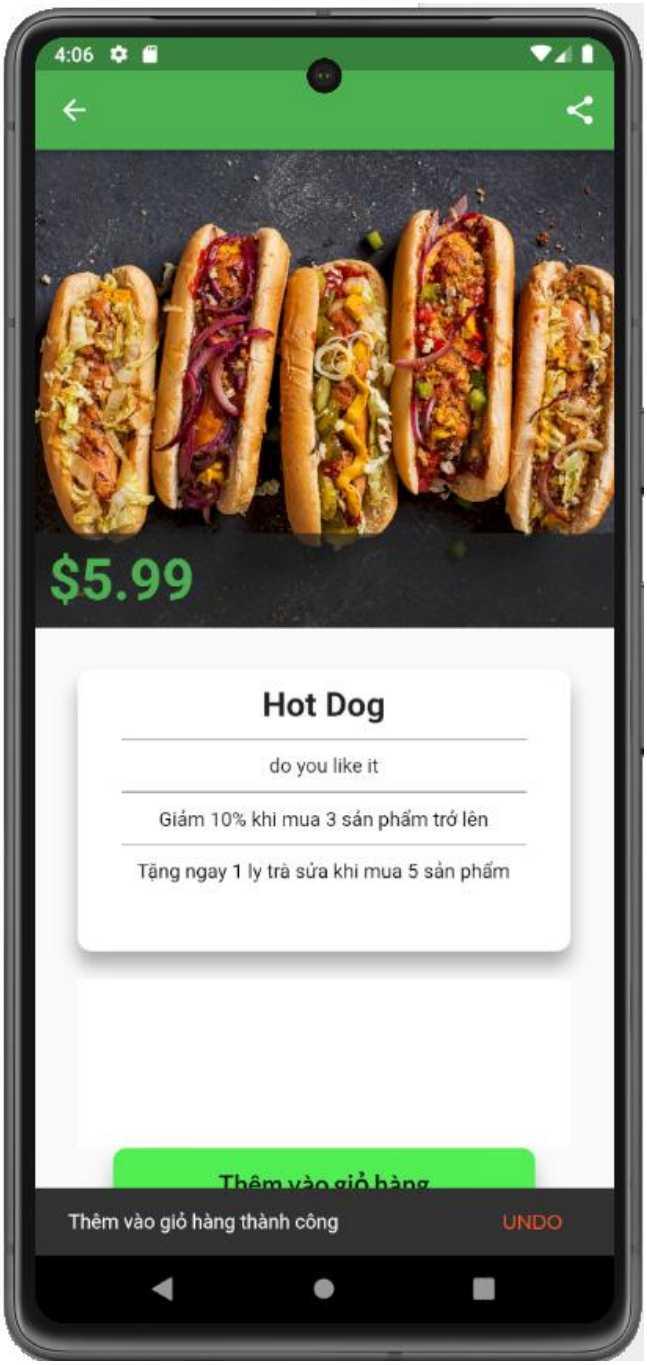
3. **Chức năng/giao diện 3:** Giao diện chi tiết sản phẩm

Miêu tả chức năng/giao diện:

Giao diện chi tiết sản phẩm cho người xem thấy rỏ và chi tiết một số thông tin cửa sản phẩm như là giá, hình ảnh, mô tả. Bên cạnh đó chúng ta cũng có thể thêm sản phẩm vào giỏ hàng trong giao diện này.

Ảnh chức năng/giao diện:





Chi tiết cài đặt:

Các widget sử dụng cho chức năng/giao diện này.

Scaffold,
AppBar,
IconButton,
SingleChildScrollView,
Column,
OverlaidProductInfo,
Stack,
Image.network,
Provider
Container,
Row,
Text,
Container,
Transform,
Divider,
TextButton,
SnackBar,
Positioned,

Các thư viện và plugin có trong giao diện này

Flutter/material.dart:

Thư viện cơ bản của Flutter, cung cấp các widget và công cụ để xây dựng giao diện người dùng.

Provider/provider.dart:

Thư viện quản lý trạng thái trong Flutter, đặc biệt là để sử dụng Provider pattern. Trong đoạn mã, nó được sử dụng để quản lý trạng thái của giỏ hàng thông qua context.read<CartManager>() .

Flutter/widgets.dart:

Chứa các widget cơ bản của Flutter như IconButton , TextButton , và các widget khác.

Sử dụng giải pháp quản lý trạng thái chia sẻ thông qua thư viện provider . Dưới đây là mô tả chi tiết.

Context.read<CartManager>():

Trong ProductDetailScreen, context.read<CartManager>() được sử dụng để đọc và sử dụng đối tượng CartManager từ Provider. Sau đó, các phương thức của CartManager được gọi khi người dùng thêm sản phẩm vào giỏ hàng.

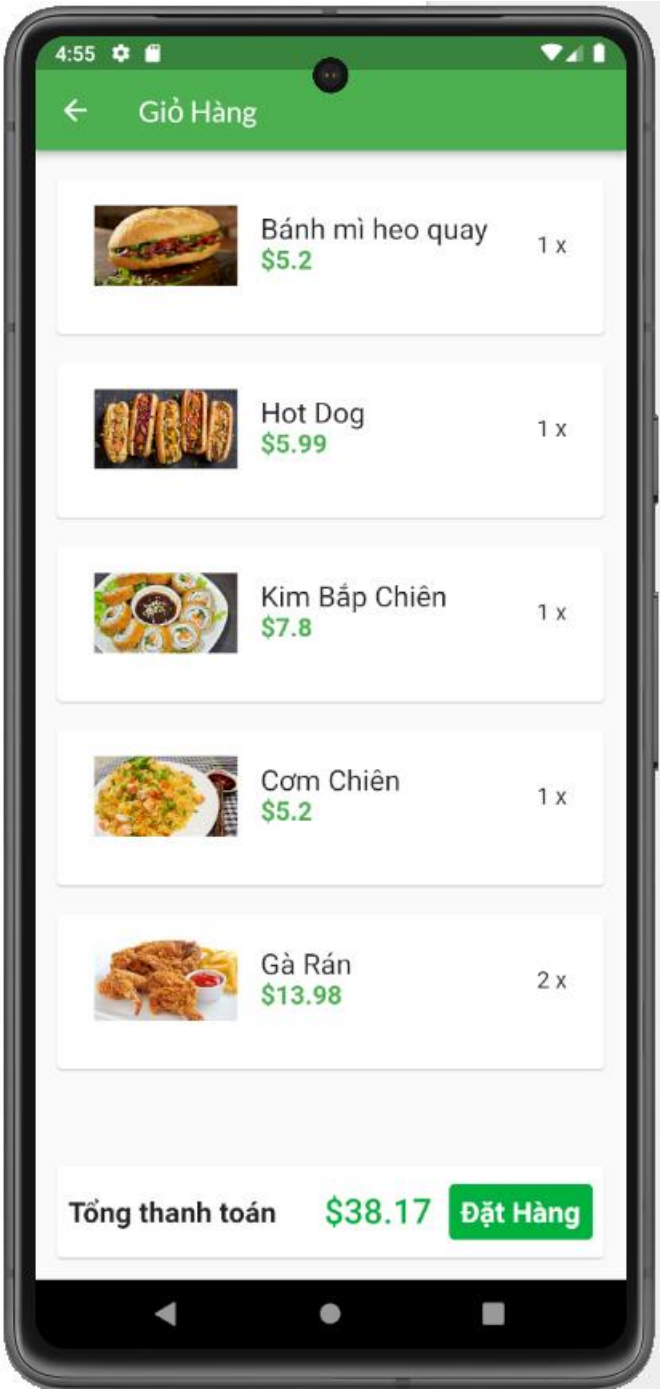
Chức năng này có thực hiện đọc dữ liệu. dùng dịch vụ lưu trữ FireBase.
GET /products.json: đọc tất cả các đối tượng từ nhánh products

4. **Chức năng/giao diện 4:** Giao diện giỏ hàng

Miêu tả chức năng/giao diện:

Giỏ hàng là nơi chứa các sản phẩm mà khách hàng đã thêm vào trong quá trình tham khảo các sản phẩm. Trong trang giỏ hàng hiện các thông tin về hình ảnh sản phẩm, giá cả của sản phẩm khi mua hoặc nhiều. Khách hàng có thể xóa sản phẩm đã thêm ra khỏi cửa hàng và cuối cùng là đặt hàng.

Ảnh chức năng/giao diện:



Chi tiết cài đặt:

Các widget sử dụng cho chức năng/giao diện này.

Dismissible	Text
Container	ChangeNotifier
Icon	Map
ValueKey	List
BuildContext	Iterable
showConfirmDialog	Product
Card	CartItem
SizedBox	Material
Padding	Theme
ListTile	Align
ClipRRect	EdgeInsets
Image.network	IconData
Image	TextStyle

Các thư viện và plugin có trong giao diện này

Flutter Material Package

- Thư viện/plugin: package:flutter/material.dart
- Vai trò: Cung cấp các widget và công cụ để xây dựng giao diện người dùng theo phong cách Material Design của Google. Đây là một phần quan trọng của Flutter và được sử dụng để tạo ra các thành phần giao diện như Scaffold, Container, Stack, SingleChildScrollView, Column, SizedBox, Flexible, ElevatedButton, TextButton, TextFormField, và nhiều widget khác.

Provider Package

- Thư viện/plugin: package:provider/provider.dart
- Vai trò: Cung cấp mô hình quản lý trạng thái và dependency injection trong Flutter.

Chức năng có sử dụng giải pháp quản lý trạng thái chia sẻ thông qua thư viện Provider. Dưới đây là miêu tả chi tiết

CartManager (Quản lý giỏ hàng) :

Đây là một lớp được mở rộng từ ChangeNotifier , là một phần của gói provider . Lớp này chứa trạng thái của giỏ hàng và cung cấp các phương thức để thay đổi trạng thái đó. Nó được sử dụng để theo dõi số lượng sản phẩm trong giỏ hàng, danh sách sản phẩm, tổng số tiền, và các phương thức thêm/xóa sản phẩm.

Provider :

Đoạn mã sử dụng Provider để cung cấp CartManager cho cây widget và widget con có thể truy cập và theo dõi trạng thái giỏ hàng. Điều này được thực hiện thông qua hàm context.watch<CartManager>() và context.read<OrdersManager>() .

Consumer Widget :

Trong màn hình giỏ hàng (CartScreen), bạn sử dụng các widget Consumer để bao bọc các phần cần lắng nghe sự thay đổi của CartManager . Các phần này bao gồm buildCartDetails và buildCartScreen . Khi trạng thái của CartManager thay đổi, chỉ những phần bên trong các Consumer này được rebuild.

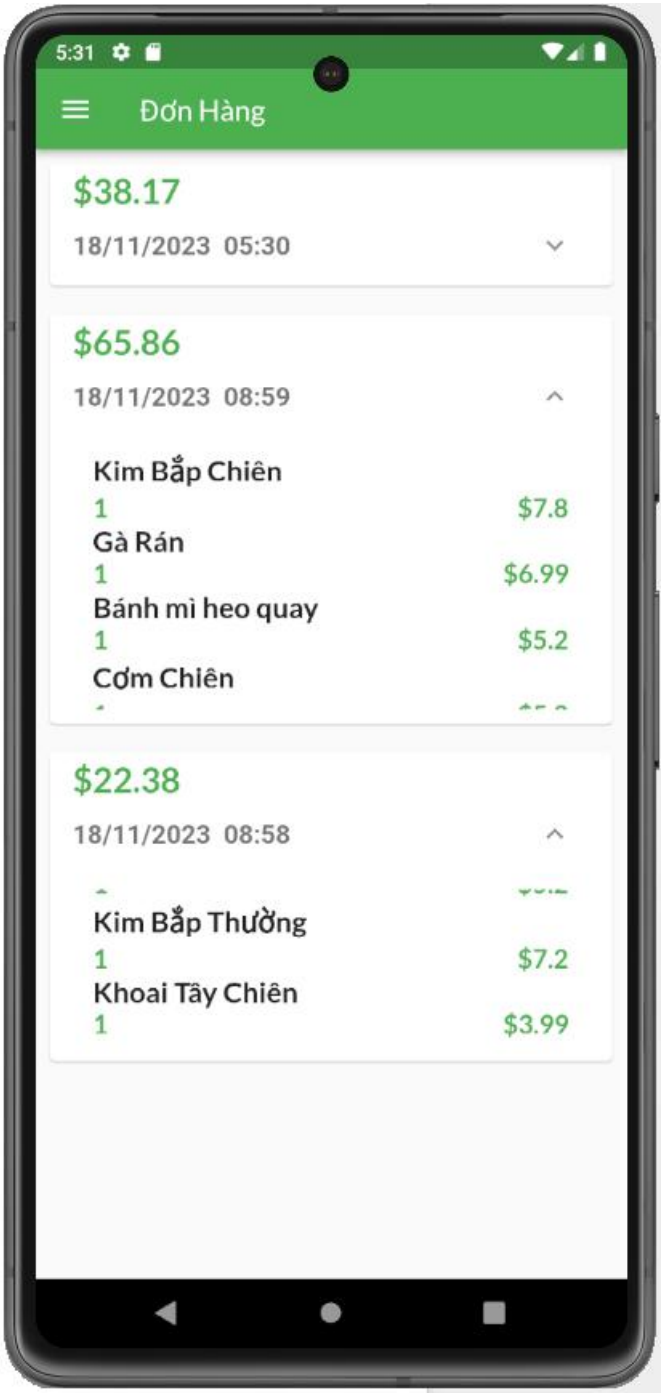
Chức năng này có thực hiện đọc dữ liệu. dùng dịch vụ lưu trữ FireBase.
GET /products.json: đọc tất cả các đối tượng từ nhánh products

5. **Chức năng/giao diện 5:** Giao diện đơn hàng

Miêu tả chức năng/giao diện:

Giao diện đơn hàng dùng để lưu trữ thông tin đơn hàng gồm: tổng giá tiền, ngày đặt hàng, thời gian đặt hàng, tên từng sản phẩm và giá tiền cửa từng sản phẩm

Ảnh chức năng/giao diện:



Chi tiết cài đặt:

Các widget sử dụng cho chức năng/giao diện này.

Scaffold

AppBar

Drawer (custom widget: AppDrawer)

Consumer<OrdersManager>

Center

Column

Text

ListView.builder

OrderItemCard

Card

buildOrderSummary() (ListTile, Text, Row, Flexible)

buildOrderDetails() (Container, ListView, Column, Text, Row, SizedBox)

AppDrawer

Các thư viện và plugin có trong giao diện này

Flutter Material Package

- Thư viện/plugin: package:flutter/material.dart
- Vai trò: Cung cấp các widget và công cụ để xây dựng giao diện người dùng theo phong cách Material Design của Google. Đây là một phần quan trọng của Flutter và được sử dụng để tạo ra các thành phần giao diện như Scaffold, Container, Stack, SingleChildScrollView, Column, SizedBox, Flexible, ElevatedButton, TextButton, TextFormField, và nhiều widget khác.

Provider Package

- Thư viện/plugin: package:provider/provider.dart
- Vai trò: Cung cấp mô hình quản lý trạng thái và dependency injection trong Flutter.

Dart:math: Được sử dụng để sử dụng hàm min trong _OrderItemCardState để giới hạn chiều cao của container.

Chức năng có sử dụng giải pháp quản lý trạng thái chia sẻ thông qua thư viện Provider. Dưới đây là miêu tả chi tiết

Provider Architecture:
- OrdersManager (ChangeNotifier):
  - Là một lớp mô hình (ChangeNotifier) chứa trạng thái liên quan đến các đơn hàng.
  - Cung cấp một danh sách các đơn hàng và các phương thức để thêm đơn hàng mới.
  - Khi có sự thay đổi trong danh sách đơn hàng, nó sẽ gọi notifyListeners() để thông báo cho các widget lắng nghe (Consumer) cần cập nhật.

OrdersScreen (StatelessWidget):
- Widget chính hiển thị danh sách đơn hàng.
- Sử dụng Consumer để lắng nghe thay đổi trong OrdersManager . Khi có thay đổi, nó sẽ rebuild các widget con bên trong nó.
- Hiển thị danh sách đơn hàng thông qua ListView.builder , và mỗi OrderItemCard được tạo bởi builder có thể truy cập đối tượng OrdersManager để lấy thông tin đơn hàng.

OrderItemCard (StatefulWidget):
- Widget này là một phần của danh sách đơn hàng.
- Sử dụng một state (_OrderItemCardState) để theo dõi trạng thái cụ thể của mỗi thẻ đơn hàng (mở rộng hoặc thu gọn).
- Khi người dùng nhấp vào nút mở rộng/thu gọn, nó sử dụng setState để cập nhật trạng thái và rebuild lại widget.

AppDrawer (StatelessWidget):
- Một widget dùng để hiển thị thanh điều hướng hoặc menu bên của ứng dụng.
- Không chứa trạng thái đặc biệt, nhưng có thể được mở rộng để chứa các mục điều hướng cho các tính năng khác của ứng dụng.

Chức năng này có thực hiện đọc dữ liệu. dùng dịch vụ lưu trữ FireBase.

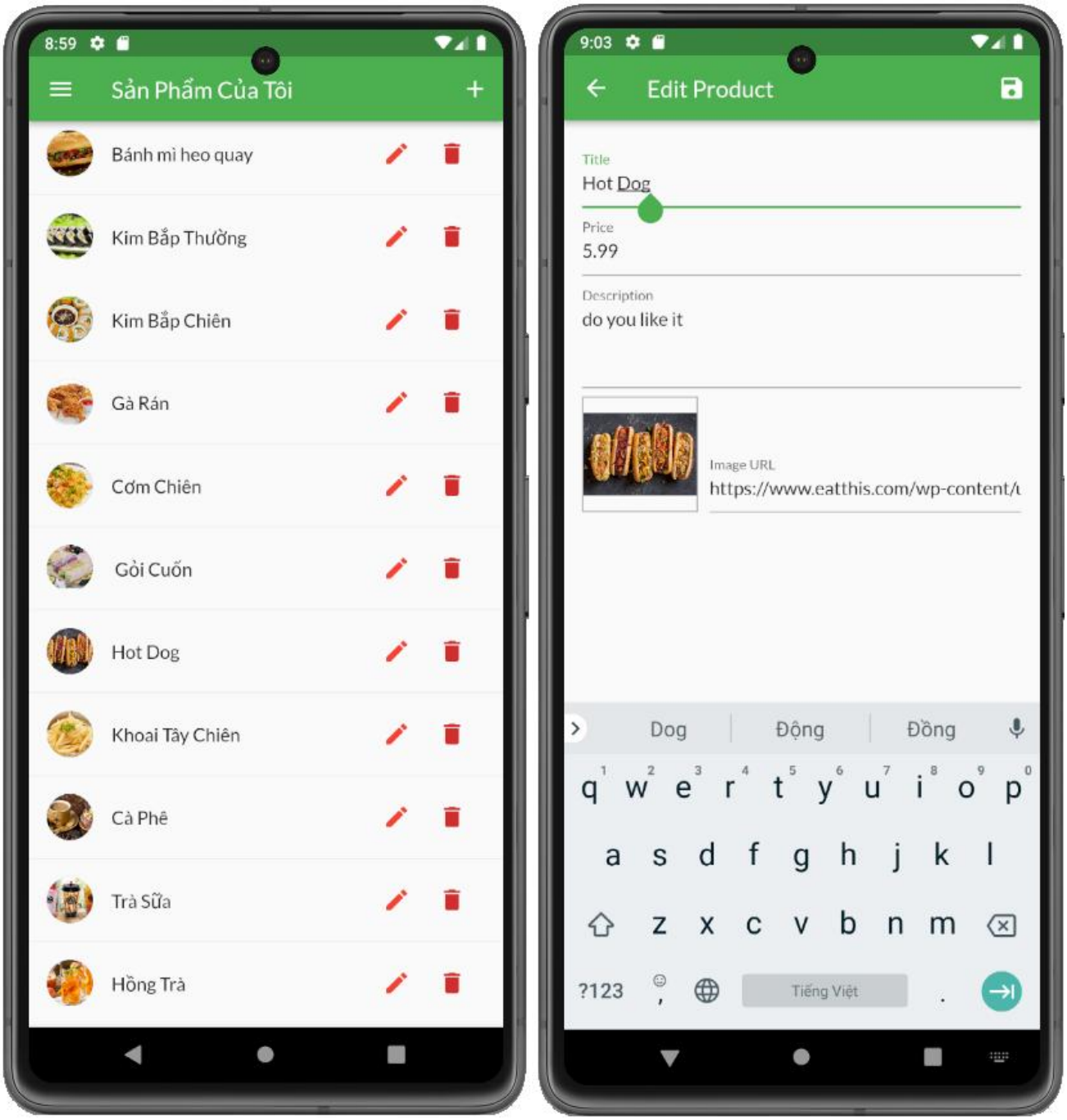
GET /products.json: đọc tất cả các đối tượng từ nhánh products

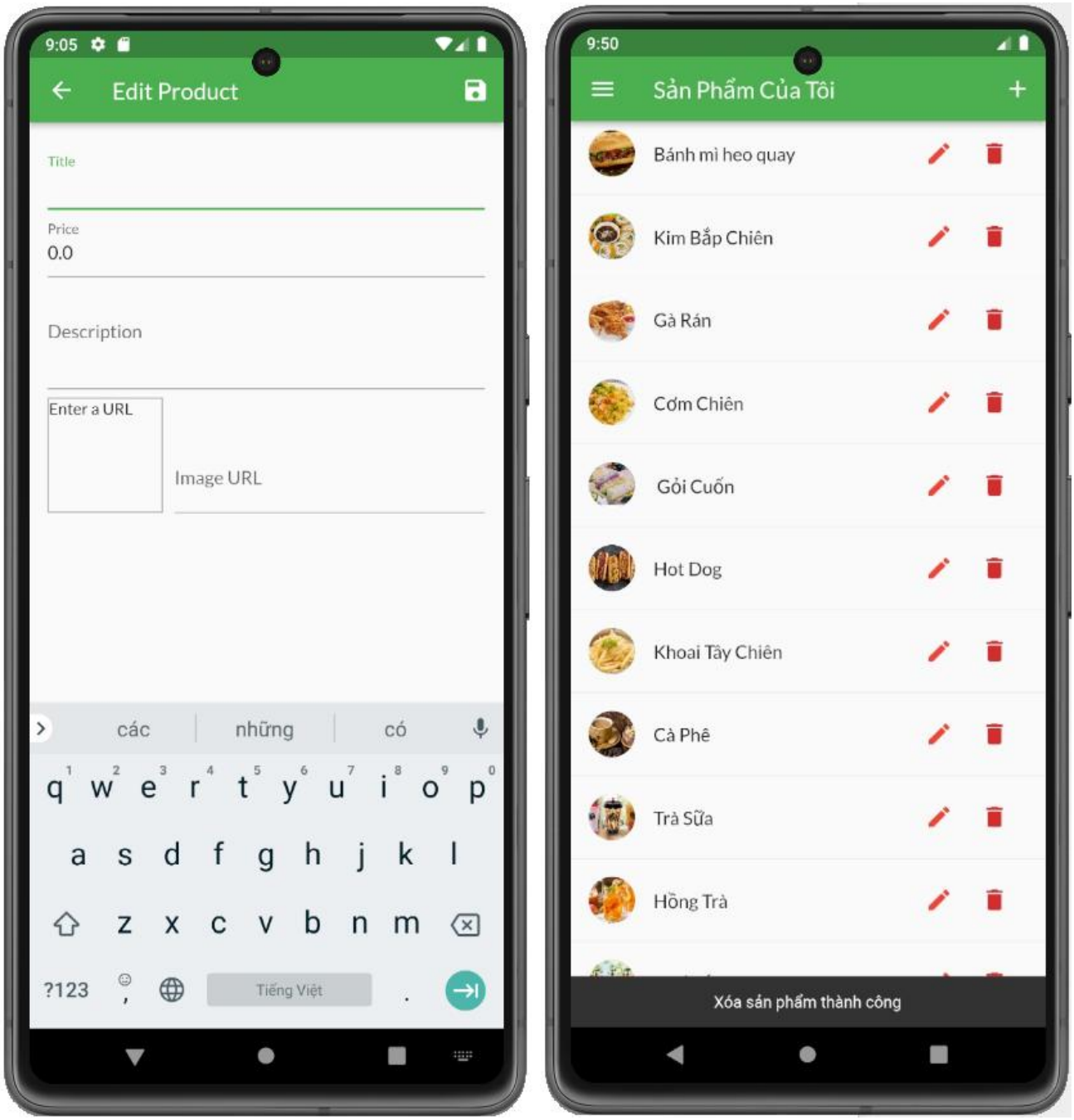
6. **Chức năng/giao diện 6:** Giao diện quản lí sản phẩm

Miêu tả chức năng/giao diện:

Giao diện này giúp người bán hàng có thể quản lí tất cả sản phẩm mà mình đang bán. Ngoài ra còn giúp người bán thêm sản phẩm mới, sửa sản phẩm đang kinh doanh và xóa sản phẩm, giao diện đơn giản dễ dàng sử dụng, giúp tiết kiệm thời gian trong khâu quản lí

Ảnh chức năng/giao diện:




9:05
Edit Product
Title
Price
0.0
Description
Enter a URL
Image URL
các
những
có
Tiếng Việt
9:50
Sản Phẩm Của Tôi
Bánh mì heo quay
Kim Bắp Chiên
Gà Rán
Cơm Chiên
Gỏi Cuốn
Hot Dog
Khoai Tây Chiên
Cà Phê
Trà Sữa
Hồng Trà
Xóa sản phẩm thành công

Các widget sử dụng cho chức năng/giao diện này.

- Scaffold
- AppBar
- IconButton
- Center
- CircularProgressIndicator
- RefreshIndicator
- ListView
- Column
- UserProductListTile
- Divider
- ListTile
- Text
- CircleAvatar
- SizedBox
- Row
- SnackBar
- Theme
- Padding
- Form
- TextFormField
- Row
- Container
- FittedBox
- Image.network
- FocusNode
- GlobalKey

Các thư viện và plugin có trong giao diện này

Provider Package:
- import 'package:provider/provider.dart' : Sử dụng để quản lý trạng thái ứng dụng và cung cấp các dịch vụ cho các widget.

Flutter Material Package:
- import 'package:flutter/material.dart' : Cung cấp các widget và công cụ thiết kế theo chủ đề của Material Design trong Flutter.

Navigator:
- Navigator : Dùng để điều hướng giữa các màn hình trong ứng dụng.

Dart Core Libraries:
- dart:async : Sử dụng cho việc xử lý các thao tác bất đồng bộ với Future và Stream .

Ignore Directive:
- ignore_for_file: unused_import : Sử dụng để bỏ qua cảnh báo về việc import các thư viện mà không sử dụng trong file code.

Chức năng có sử dụng giải pháp quản lý trạng thái chia sẻ thông qua thư viện Provider. Dưới đây là miêu tả chi tiết

UserProductsScreen:

- UserProductsScreen sử dụng Provider để quản lý trạng thái của ProductsManager, một lớp được sử dụng để quản lý danh sách sản phẩm.
- Thông qua context.read<ProductsManager>(), UserProductsScreen có thể đọc trạng thái hiện tại và làm cho widget này cập nhật khi có sự thay đổi trong danh sách sản phẩm.
- Dùng Consumer để lắng nghe thay đổi và xây dựng lại widget chỉ khi cần thiết.

UserProductListTile:

- UserProductListTile nhận Product thông qua tham số và không trực tiếp quản lý trạng thái.
- Sử dụng context.read<ProductsManager>() để gọi các phương thức xóa sản phẩm khi người dùng nhấn nút xóa.

EditProductScreen:

- EditProductScreen sử dụng Provider để quản lý trạng thái của ProductsManager, giống như UserProductsScreen.
- EditProductScreen cũng sử dụng context.read<ProductsManager>() để thêm hoặc cập nhật sản phẩm.

_EditProductScreenState:

- Form trong _EditProductScreenState được sử dụng để nhập và xác thực dữ liệu sản phẩm.
- Khi người dùng nhấn nút lưu, _saveForm sử dụng context.read<ProductsManager>() để thêm hoặc cập nhật sản phẩm trong danh sách quản lý bởi ProductsManager.

Chức năng này có thực hiện đọc và lưu trữ dữ liệu. Dùng dịch vụ lưu trữ FireBase.