

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG  
TRƯỜNG ĐẠI HỌC THỦY LỢI**



**BÀI TẬP LỚN**

**PHÁT TRIỂN ÚNG DỤNG CHO THIẾT BỊ DI ĐỘNG  
ĐỀ TÀI: ÚNG DỤNG ...**

**Giáo viên hướng dẫn: ThS. Kiều Tuấn Dũng**

**Sinh viên thực hiện:**

STT	Mã sinh viên	Họ và tên	Lớp
1	2251061692	Phùng Thành An	64CNTT2
2	2251061717	Đào Ngọc Ánh	64CNTT2
3	2251061747	Nguyễn Xuân Đức	64CNTT2

**Hà Nội, năm 2025**

BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG  
TRƯỜNG ĐẠI HỌC THỦY LỢI



**BÀI TẬP LỚN**

**PHÁT TRIỂN ÚNG DỤNG CHO THIẾT BỊ DI ĐỘNG  
ĐỀ TÀI: ÚNG DỤNG ĐẶT ĐỒ ĂN (FAST FOOD ORDER)**

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Băng Số	Băng Chữ
1	2251061692	Phùng Thành An	25/07/2004		
2	2251061717	Đào Ngọc Ánh	06/02/2004		
3	2251061747	Nguyễn Xuân Đức	05/07/2004		

**CÁN BỘ CHẤM THI**

**Hà Nội, năm 2025**

## LỜI NÓI ĐẦU

Ngày nay, việc gọi món ăn qua mạng đã trở nên quá quen thuộc với nhiều người, đặc biệt là trong cuộc sống bận rộn. Thay vì phải ra ngoài, chỉ cần mở điện thoại, vài thao tác đơn giản là đã có thể đặt món từ quán ăn và chờ được giao tận nơi. Điều này không chỉ tiện lợi mà còn giúp tiết kiệm khá thời gian và công sức.

Nhờ vào sự phát triển mạnh mẽ của công nghệ và mạng internet, việc thưởng thức ẩm thực cũng trở nên đa dạng hơn bao giờ hết. Dù bạn cần một bữa sáng nhanh gọn, một bữa trưa gọn nhẹ hay một bữa tối đầy đủ, tất cả đều có thể được đặt trong tích tắc. Hơn nữa, ứng dụng còn thường xuyên có chương trình khuyến mãi, giảm giá, giúp việc ăn uống vừa tiện lợi lại vừa kinh tế.

Không chỉ phục vụ nhu cầu của người tiêu dùng, các ứng dụng đặt đồ ăn còn hỗ trợ rất lớn cho các quán ăn trong việc tiếp cận khách hàng mới và mở rộng thị trường. Nhờ đó, việc kinh doanh trở nên linh hoạt và hiệu quả hơn rất nhiều.

Trong nhịp sống ngày càng hối hả, những tiện ích từ các ứng dụng đặt đồ ăn thực sự góp phần làm cho cuộc sống trở nên nhẹ nhàng, thoái mái và phong phú hơn mỗi ngày.

## MỤC LỤC

<b>LỜI NÓI ĐẦU .....</b>	3
<b>DANH MỤC HÌNH ẢNH .....</b>	6
<b>DANH MỤC BẢNG BIỂU .....</b>	8
<b>DANH MỤC CÁC TỪ VIẾT TẮT .....</b>	9
<b>Chương 1. TỔNG QUAN VỀ ĐỀ TÀI .....</b>	10
<b>1.1. Giới thiệu về đề tài.....</b>	10
<b>1.2. Mục tiêu của đề tài .....</b>	10
<b>1.3. Phạm vi của đề tài.....</b>	10
<b>1.4 Phân chia nhiệm vụ .....</b>	11
<b>Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ .....</b>	14
<b>2.1. Kiến trúc hệ thống .....</b>	14
2.1.1. Thành phần hệ thống .....	14
2.1.2. Kiến trúc MVVM của ứng dụng .....	14
<b>2.2. Giới thiệu về Công nghệ phát triển.....</b>	14
2.1.1. Nền tảng phát triển .....	14
2.1.2. Các công nghệ chính .....	14
<b>Chương 3. XÂY DỰNG ỨNG DỤNG .....</b>	16
<b>3.1. Thiết kế Figma .....</b>	16
<b>3.2. Thiết kế CSDL .....</b>	18
<b>3.3. Giao diện ứng dụng .....</b>	19
<b>3.4. Code minh họa các chức năng cốt lõi.....</b>	31
<b>KẾT LUẬN .....</b>	67
<b>1. Kết quả đạt được .....</b>	67

<b>2. Nhược điểm .....</b>	67
<b>3. Hướng phát triển .....</b>	67
<b>TÀI LIỆU THAM KHẢO.....</b>	68

## **DANH MỤC HÌNH ẢNH**

Hình 1: Giao diện chức năng cơ bản .....	16
Hình 2: Giao diện các trang ADMIN .....	16
Hình 3: Giao diện chức năng chi tiết ADMIN .....	17
Hình 4: Giao diện chức năng USER.....	17
Hình 5 Giao diện thực đơn USER .....	18
Hình 6: Giao diện chi tiết USER .....	18
Hình 7: Cơ sở dữ liệu .....	19
Hình 8: Giao diện màn hình đầu tiên.....	19
Hình 9: Giao diện đăng nhập.....	20
Hình 10: Giao diện đăng ký.....	20
Hình 11: Giao diện trang chủ USER .....	21
Hình 12: Giao diện trang món ngon USER.....	21
Hình 13: Giao diện chi tiết món ngon USER .....	22
Hình 14: Giao diện giỏ hàng USER .....	22
Hình 15: Giao diện tạo đơn hàng USER .....	23
Hình 16: Giao diện thông tin giao hàng USER .....	23
Hình 17: Giao diện phản hồi USER .....	24
Hình 18: Giao diện trang liên hệ USER .....	24
Hình 19: Giao diện trang tài khoản USER .....	25
Hình 20: Giao diện trang lịch sử đặt hàng.....	25
Hình 21: Giao diện trang đổi mật khẩu USER .....	26
Hình 22: Giao diện thêm món ngon ADMIN.....	26
Hình 23 Giao diện chi tiết món ngon ADMIN.....	27
Hình 24 Giao diện trang sửa món ngon ADMIN .....	27
Hình 25 Giao diện trang đồ uống ADMIN.....	28
Hình 26 Giao diện trang thêm đồ uống ADMIN.....	28
Hình 27 Giao diện trang chi tiết đồ uống ADMIN.....	29
Hình 28 Giao diện trang sửa đồ uống ADMIN .....	29
Hình 29 Giao diện trang đồ tráng miệng ADMIN.....	30
Hình 30 Giao diện trang thêm tráng miệng ADMIN.....	30
Hình 31 Giao diện trang sửa đồ tráng miệng ADMIN .....	31
Hình 32 Giao diện trang doanh thu ADMIN .....	31
Hình 33 Chức năng xem trang món ngon USER .....	33
Hình 34 Chức năng xem trang đồ uống USER .....	34

Hình 35 Chức năng xem trang đồ tráng miệng – USER .....	36
Hình 36 Chức năng tạo đơn hàng - USER .....	38
Hình 37 Chức năng thêm địa chỉ - USER .....	39
Hình 38 Chức năng xem lịch sử đơn hàng - USER.....	40
Hình 39 Chức năng phản hồi - USER .....	41
Hình 40 Chức năng xem trang món ngon – ADMIN .....	43
Hình 41 Chức năng xem trang đồ uống - ADMIN.....	44
Hình 42 Chức năng xem trang tráng miệng – ADMIN .....	46
Hình 43 Chức năng thêm món ngon - ADMIN .....	49
Hình 44 Chức năng thêm đồ uống - ADMIN .....	52
Hình 45 Chức năng thêm món tráng miệng - ADMIN.....	55
Hình 46 Chức năng cập nhật món ngon - ADMIN .....	59
Hình 47 Chức năng cập nhật đồ uống – ADMIN.....	63
Hình 48 Chức năng cập nhật tráng miệng - ADMIN .....	66

## **DANH MỤC BẢNG BIỂU**

Bảng 1.1: Bảng phân chia công việc 1 .....13

## DANH MỤC CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	MVVM	Model-View-ViewModel
2	FE	FrontEnd
3	BE	BackEnd
4	AI	Artificial Intelligence

# Chương 1. TỔNG QUAN VỀ ĐÈ TÀI

## 1.1. Giới thiệu về đè tài

Trong thời đại hiện nay, nhu cầu đặt đồ ăn trực tuyến ngày càng trở nên phổ biến nhờ vào sự tiện lợi mà dịch vụ này mang lại. Người dùng không cần phải đến tận quán, mà chỉ cần vài thao tác đơn giản trên điện thoại là đã có thể lựa chọn món ăn yêu thích và đặt hàng nhanh chóng. Những ứng dụng đặt đồ ăn giúp tiết kiệm thời gian, mang đến thực đơn phong phú từ quán ăn khác nhau.

Ứng dụng được xây dựng trong khuôn khổ đè tài này không chỉ tập trung vào việc phục vụ người dùng trong quá trình đặt món, mà còn hỗ trợ hiệu quả cho các nhà hàng, quán ăn trong việc quản lý thực đơn, xử lý đơn hàng và tiếp nhận phản hồi từ khách hàng. Toàn bộ dữ liệu được lưu trữ và đồng bộ thông qua Firebase Realtime Database, giúp hệ thống hoạt động ổn định, liên tục cập nhật và mang lại trải nghiệm mượt mà, chính xác cho cả người dùng và người quản trị.

## 1.2. Mục tiêu của đè tài

Mục tiêu chính của đè tài là xây dựng một ứng dụng đặt đồ ăn trên nền tảng Android, giúp người dùng có thể:

- Dễ dàng đặt hàng món ăn từ danh sách có sẵn.
- Quản lý giờ hàng, đặt hàng và theo dõi đơn hàng theo thời gian thực.
- Gửi phản hồi, đánh giá về chất lượng món ăn và dịch vụ.

Đối với quản trị viên (admin), ứng dụng hỗ trợ:

- Quản lý danh sách món ăn (thêm, sửa, xóa, tìm kiếm).
- Quản lý đơn hàng và cập nhật trạng thái giao hàng.
- Quản lý phản hồi từ người dùng, giúp cải thiện dịch vụ.
- Theo dõi doanh thu bán hàng theo từng ngày cụ thể.

Về mặt kỹ thuật, ứng dụng sử dụng kiến trúc MVVM kết hợp với Firebase Authentication để quản lý tài khoản người dùng và Firebase Realtime Database để xử lý dữ liệu theo thời gian thực.

## 1.3. Phạm vi của đè tài

- Nền tảng: Ứng dụng được phát triển trên hệ điều hành Android.
- Đối tượng sử dụng:
  - + Người dùng (khách hàng) đặt món ăn.
  - + Quản trị viên (admin) quản lý hệ thống.
- Tính năng chính:
  - + Quản lý tài khoản người dùng (đăng ký, đăng nhập, đổi mật khẩu, đăng xuất).
  - + Hiển thị danh sách menu món ăn.
  - + Tìm kiếm, xem chi tiết và đặt món ăn vào giờ hàng.
  - + Thực hiện đặt hàng và theo dõi lịch sử đơn hàng.
  - + Gửi phản hồi, đánh giá về món ăn.

- + Quản lý món ăn, đơn hàng, phản hồi và doanh thu (dành cho admin).
- Công nghệ sử dụng:
  - + Firebase Realtime Database để lưu trữ dữ liệu.
  - + Firebase Authentication để xác thực người dùng.
  - + Firebase Storage để lưu trữ hình ảnh.
  - + View Binding, RecyclerView, BottomSheetDialog, ViewPager2 để xây dựng giao diện.

#### 1.4 Phân chia nhiệm vụ

Họ tên	Tên công việc
Nguyễn Xuân Đức	<p>FE:</p> <ul style="list-style-type: none"> <li>- Frontend trang tráng miệng – Admin</li> <li>- Frontend trang thêm tráng miệng – Admin</li> <li>- Frontend trang đồ uống – Admin</li> <li>- Frontend trang thêm đồ uống – Admin</li> <li>- Frontend trang món ngon – Admin</li> <li>- Frontend trang thêm món ngon – Admin</li> <li>- Frontend trang phản hồi – Admin</li> <li>- Frontend trang đơn hàng – Admin</li> <li>- Frontend trang tài khoản – Admin</li> <li>- Frontend trang doanh thu - Admin</li> </ul> <p>BE:</p> <ul style="list-style-type: none"> <li>- Backend Admin có thể thêm, sửa, xóa, hiển thị danh sách các món ăn và tìm kiếm món ăn theo tên</li> <li>- Backend Admin có thể thêm, sửa, xóa, hiển thị danh sách các đồ uống và tìm kiếm đồ uống theo tên</li> <li>- Backend Admin có thể thêm, sửa, xóa, hiển thị danh sách các đồ tráng miệng và tìm kiếm đồ tráng miệng theo tên</li> <li>- Backend Admin quản lý danh sách order (trang đơn hàng) của tất cả người dùng, cập nhật trạng thái order khi đơn hàng hoàn thành</li> </ul>
Phùng Thành An	<p>FE:</p> <ul style="list-style-type: none"> <li>- Frontend trang chủ - User</li> <li>- Frontend trang món ngon – User</li> <li>- Frontend đồ uống – User</li> <li>- Frontend tráng miệng – User</li> <li>- Frontend trang chi tiết – User</li> <li>- Frontend trang giỏ hàng – User</li> <li>- Frontend trang thông tin khách hàng</li> </ul>

	<ul style="list-style-type: none"> <li>- Frontend trang thêm, sửa thông tin khách hàng</li> <li>- Frontend trang tài khoản – User</li> <li>- Frontend trang phản hồi -User</li> </ul> <p>BE:</p> <ul style="list-style-type: none"> <li>- Backend Hiển thị danh sách lịch sử đã Order theo từng User (màu xám: đã hoàn thành; màu trắng: chưa hoàn thành)</li> <li>- Backend Hiển thị danh sách Foods phổ biến ở màn hình Home dưới dạng Slide Images + Auto Run (Realtime)</li> <li>- Backend Tính năng search Product ở Home, search trang Món ngon, Đồ uống, Tráng miệng</li> <li>- Backend Hiển thị danh sách Product ở các trang Món ngon, Đồ uống, Tráng miệng</li> <li>- Backend Hiển thị thông tin chi tiết của Food: thông tin cơ bản, miêu tả, một số hình ảnh khác...</li> <li>- Backend User có thể quản lý thêm, sửa, xóa, hiển thị danh sách các thông tin địa chỉ muốn giao hàng</li> </ul>
Đào Ngọc Ánh	<p>FE:</p> <ul style="list-style-type: none"> <li>- Frontend trang đầu</li> <li>- Frontend trang đăng nhập</li> <li>- Frontend trang đăng ký</li> <li>- Frontend trang quên mật khẩu</li> <li>- Frontend trang đổi mật khẩu</li> <li>- Frontend trang chủ ADMIN</li> <li>- Frontend trang liên hệ - User</li> <li>- Frontend trang lịch sử đơn hàng-User</li> </ul> <p>BE:</p> <ul style="list-style-type: none"> <li>- Backend đăng nhập, đăng ký, quên mật khẩu, đổi mật khẩu, hiển thị user profile, đăng xuất.</li> <li>- Backend module quản lý tài khoản theo 2 role riêng biệt là Admin (quản trị viên) và người dùng (user)</li> <li>- Backend Feedback gửi phản hồi, đóng góp ý kiến, đánh giá cho quán của User</li> <li>- Backend Admin có thể quản lý danh sách feedback (phản hồi, đóng góp ý kiến) của user</li> <li>- Backend User xem được danh sách các phản hồi mình đã gửi</li> </ul>

	<ul style="list-style-type: none"><li>- Backend Trang liên hệ</li><li>- Backend Admin có thể theo dõi doanh thu bán hàng theo từng ngày cụ thể</li><li>- Backend Thực hiện Order theo thực đơn đã chọn</li><li>- Backend Hiển thị tất cả danh sách Foods người dùng đã thêm vào giỏ hàng: có thể update số lượng hoặc xóa món ăn nếu không thích.</li><li>- Backend thêm Food vào giỏ hàng</li></ul>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Bảng 1.1: Bảng phân chia công việc 1

## Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ

### 2.1. Kiến trúc hệ thống

Hệ thống ứng dụng đặt đồ ăn được xây dựng theo mô hình MVVM (Model-View-ViewModel) kết hợp với Firebase Realtime Database để đảm bảo dữ liệu được cập nhật theo thời gian thực. Hệ thống gồm hai thành phần chính:

#### 2.1.1. Thành phần hệ thống

- Người dùng (User): Khách hàng sử dụng ứng dụng để đặt món ăn, quản lý giỏ hàng, theo dõi đơn hàng và gửi phản hồi.
- Quản trị viên (Admin): Quản lý danh sách món ăn, đơn hàng, phản hồi của khách hàng và theo dõi doanh thu.
- Hệ thống Firebase: Lưu trữ dữ liệu người dùng, đơn hàng, phản hồi và thực hiện xác thực đăng nhập, lưu trữ hình ảnh.

#### 2.1.2. Kiến trúc MVVM của ứng dụng

- Mô hình MVVM giúp tách biệt logic xử lý dữ liệu với giao diện người dùng, giúp dễ bảo trì và mở rộng.
- Model (M): Chứa các lớp dữ liệu như Food, User, Order, Feedback, đồng thời tương tác với Firebase để lưu trữ và truy xuất dữ liệu.
- ViewModel (VM): Xử lý logic nghiệp vụ, gọi dữ liệu từ Firebase và cập nhật giao diện thông qua LiveData.
- View (V): Các Activity và Fragment chịu trách nhiệm hiển thị dữ liệu và nhận tương tác từ người dùng.

### 2.2. Giới thiệu về Công nghệ phát triển

Ứng dụng sử dụng các công nghệ hiện đại để đảm bảo hiệu suất cao, giao diện mượt mà và cập nhật dữ liệu theo thời gian thực.

#### 2.2.1. Nền tảng phát triển

- Ngôn ngữ lập trình: Java
- IDE sử dụng: Android Studio
- Kiến trúc: MVVM

#### 2.2.2. Các công nghệ chính

- View Binding trong Android
- Realtime Database from Firebase
- Thêm, Sửa, Xóa, Hiển thị danh sách data, Search data tất cả đều Realtime từ Database
- Sử dụng Authentication Firebase, Firebase Storage
- Sử dụng control BottomSheetDialog để thiết kế layout Add to Cart and Order
- BottomNavigationView + ViewPager2 + Fragments trong Android

- ViewPage2 + CircleIndicator3 + Auto Run Slide Images
- Xử lý logic hiển thị danh sách dữ liệu với RecyclerView trong NestedScrollView
- Load Image from Url với thư viện: Glide

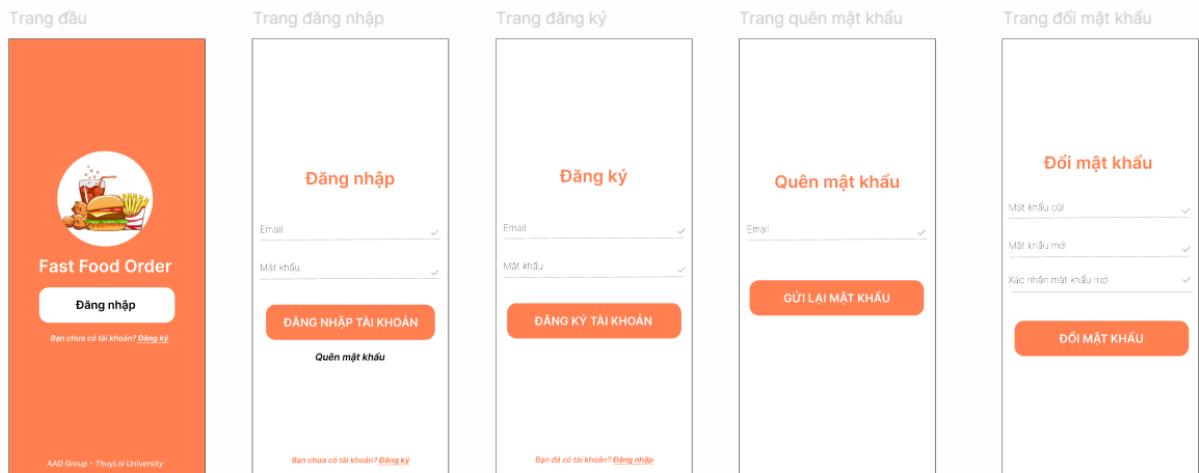
## Chương 3. XÂY DỰNG ỨNG DỤNG

### 3.1. Thiết kế Figma

Link Github: [https://github.com/Mashall-Anie/AAD\\_Project](https://github.com/Mashall-Anie/AAD_Project)

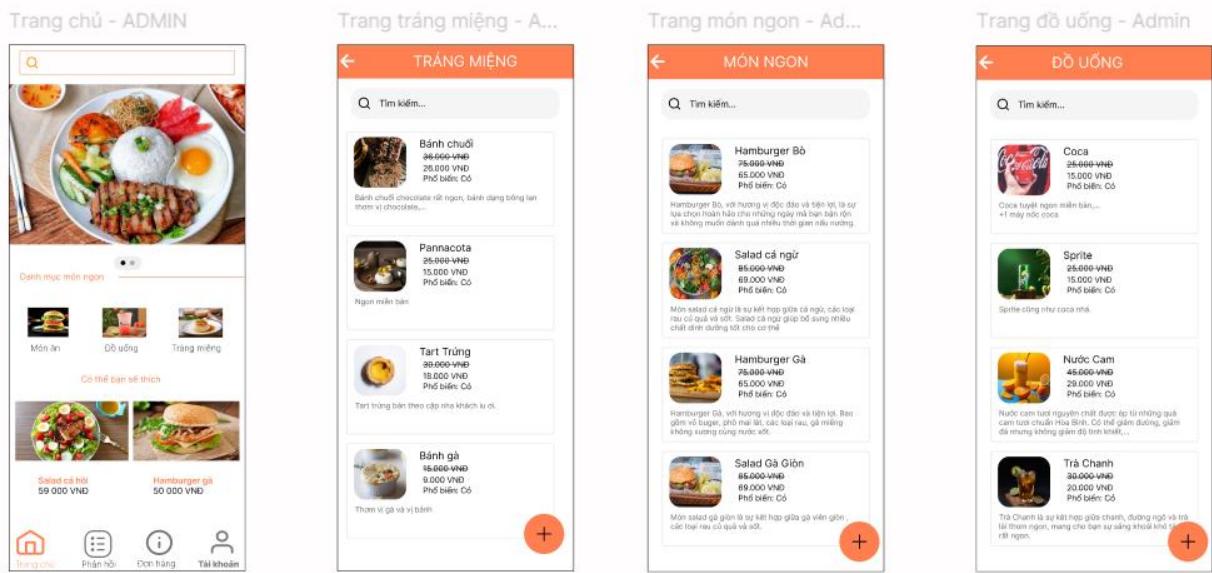
Ảnh kết xuất Figma:

- Trang đầu, đăng nhập, đăng ký, quên mật khẩu, đổi mật khẩu



Hình 1: Giao diện chức năng cơ bản

- Trang chủ, trang món ngon, trang tráng miệng, trang đồ uống admin



Hình 2: Giao diện các trang ADMIN

- Trang đơn hàng, thêm món ngon, thêm tráng miệng, thêm đồ uống, chi tiết món ngon, sửa món ngon, trang tài khoản, phản hồi, doanh thu của admin

The image displays a series of screenshots from a mobile application's admin interface. The screens include:

- Trang đơn hàng...**: Order Details screen showing an order for a Hamburger (SL: 01) worth 130,000 VND.
- Trang thêm món...**: Add Item screen for adding a new dish.
- Trang thêm đồ uống...**: Add Drink screen for adding a new drink.
- Trang thêm tráng miệng...**: Add Dessert screen for adding a new dessert.
- Trang Chi Tiết ...**: Item Detail screen for a Hamburger.
- Trang sửa món...**: Edit Item screen for the same Hamburger.
- Trang tài khoản...**: Account Management screen showing a user account (admin123@gmail.com).
- Trang phản hồi...**: Feedback screen listing user reviews.
- Trang doanh thu...**: Sales Report screen showing total sales and breakdown by user.

Hình 3: Giao diện chức năng chi tiết ADMIN

- Trang tài khoản, liên hệ, phản hồi, lịch sử đơn hàng của user

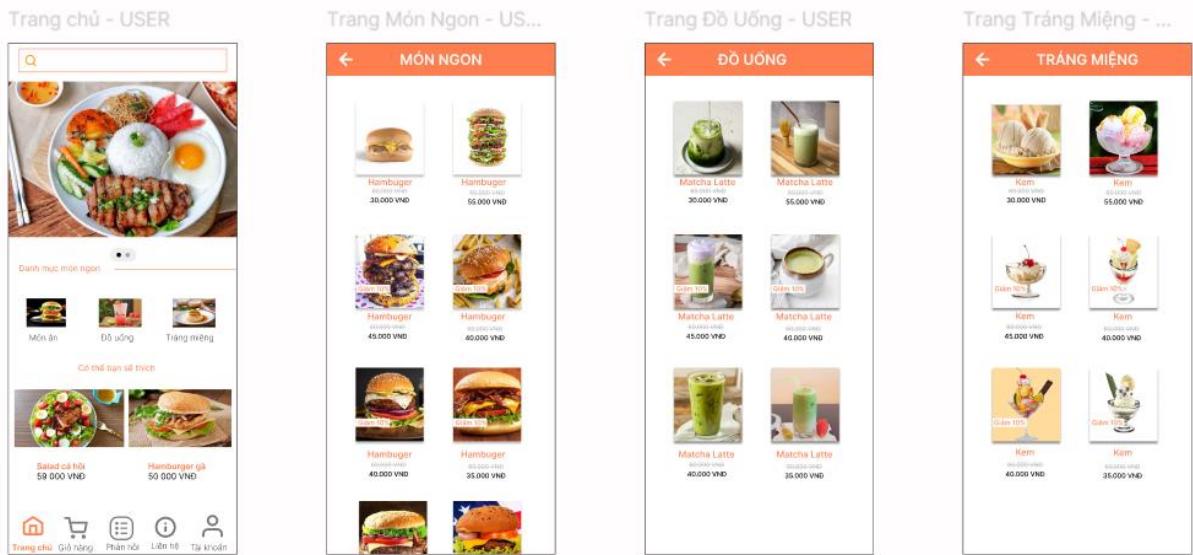
The image displays a series of screenshots from a mobile application's user interface. The screens include:

- Trang tài khoản-USER**: Account Management screen for the user 'anhngoc123@gmail.com'.
- Trang liên hệ-USER**: Contact screen for 'Fast Food Order' with a hotline of 0335169928.
- Trang phản hồi-USER**: Feedback screen where the user can enter their email and leave a comment.
- Trang lịch sử đơn hàng...**: Order History screen showing two recent orders (Order ID 123456 and 123457) with details like item names, quantities, and total price.

Hình 4: Giao diện chức năng USER

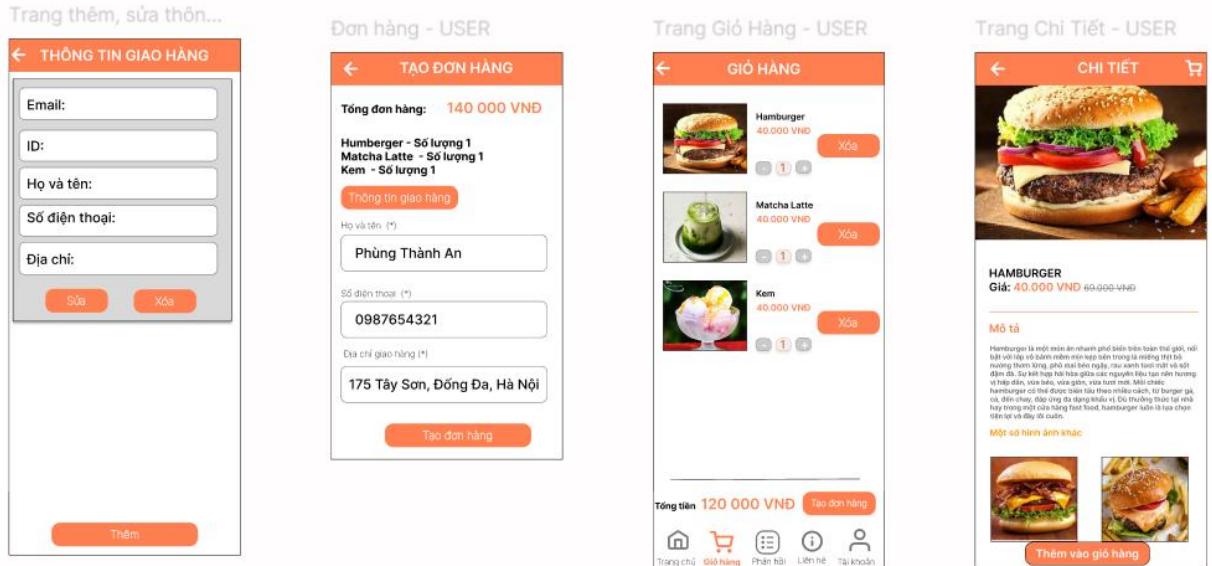
Phát triển ứng dụng cho thiết bị di động

- Trang chủ, món ngon, đồ uống, tráng miệng user



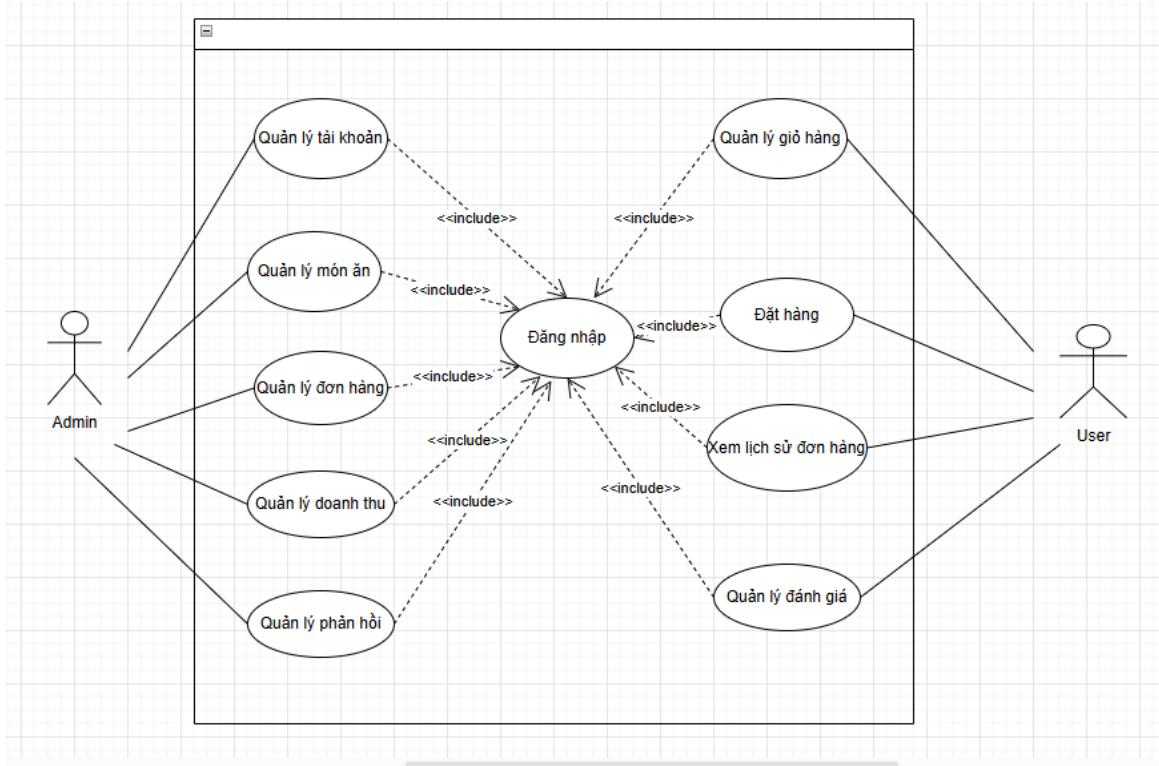
Hình 5 Giao diện thực đơn USER

- Trang thông tin giao hàng, đơn hàng, giỏ hàng, chi tiết user



Hình 6: Giao diện chi tiết USER

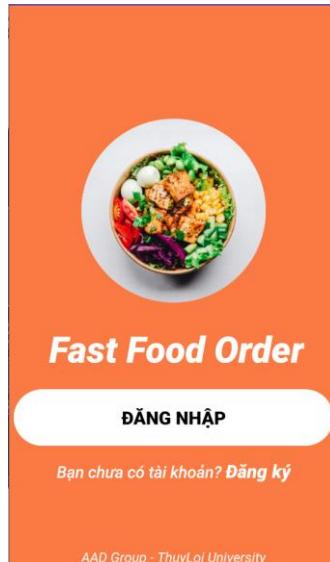
### 3.2. Thiết kế CSDL



Hình 7: Cơ sở dữ liệu

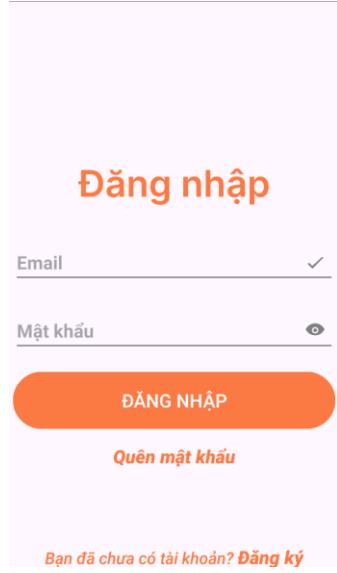
### 3.3. Giao diện ứng dụng

#### 3.3.1. Màn hình đầu tiên



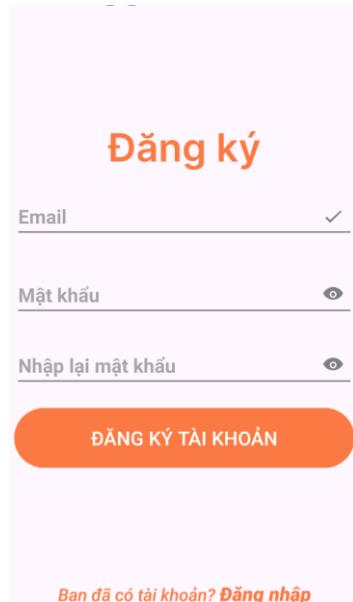
Hình 8: Giao diện màn hình đầu tiên

#### 3.3.2. Màn hình đăng nhập



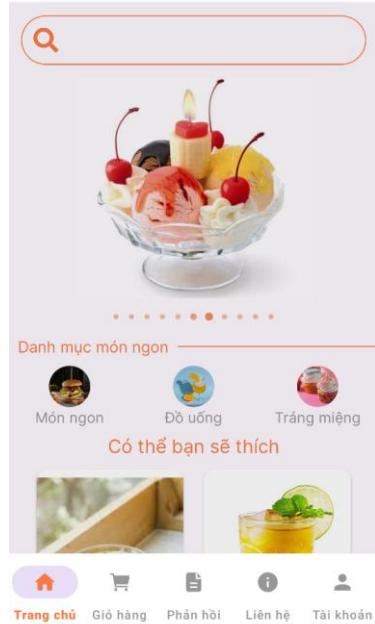
Hình 9: Giao diện đăng nhập

### 3.3.3. Màn hình đăng ký



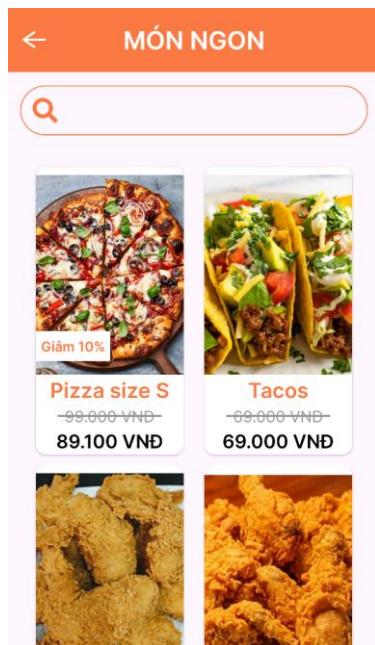
Hình 10: Giao diện đăng ký

### 3.3.4. Màn hình trang chủ user



Hình 11: Giao diện trang chủ USER

### 3.3.5. Màn hình trang món ngon user



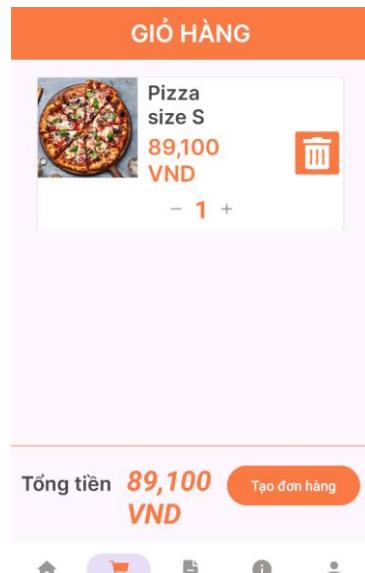
Hình 12: Giao diện trang món ngon USER

### 3.3.6. Màn hình trang chi tiết user



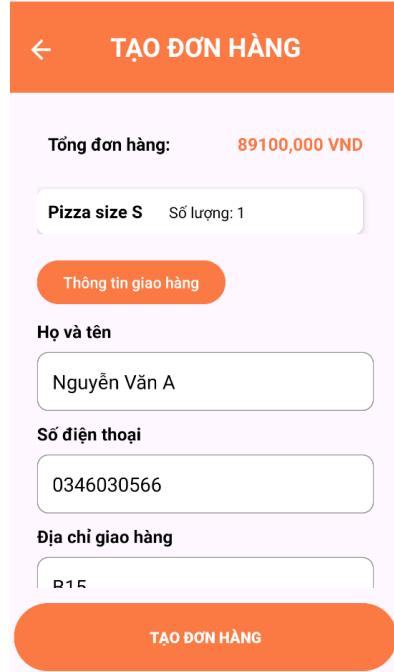
Hình 13: Giao diện chi tiết món ngon USER

### 3.3.7. Màn hình trang giỏ hàng user



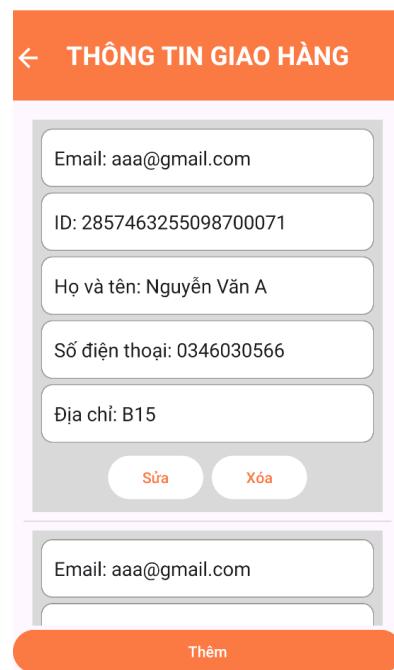
Hình 14: Giao diện giỏ hàng USER

### 3.3.8. Màn hình tạo đơn hàng user



Hình 15: Giao diện tạo đơn hàng USER

### 3.3.9. Màn hình thông tin giao hàng user



Hình 16: Giao diện thông tin giao hàng USER

### 3.3.10. Màn hình trang phản hồi user

**PHẢN HỒI**

Email(\*)  
aaa@gmail.com

Bình luận  
Nhập bình luận

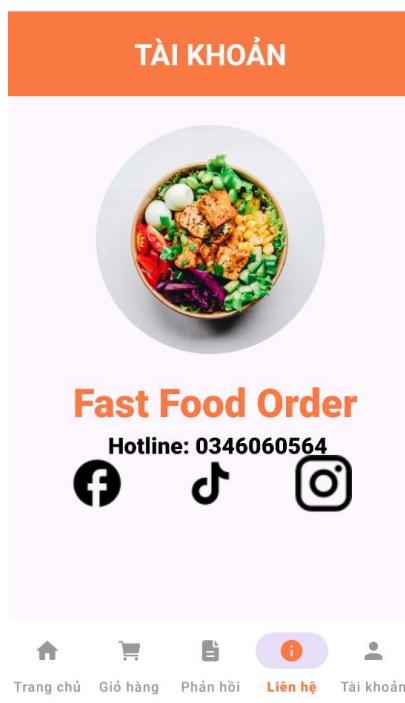
Gửi phản hồi



Trang chủ Giỏ hàng Phản hồi Liên hệ Tài khoản

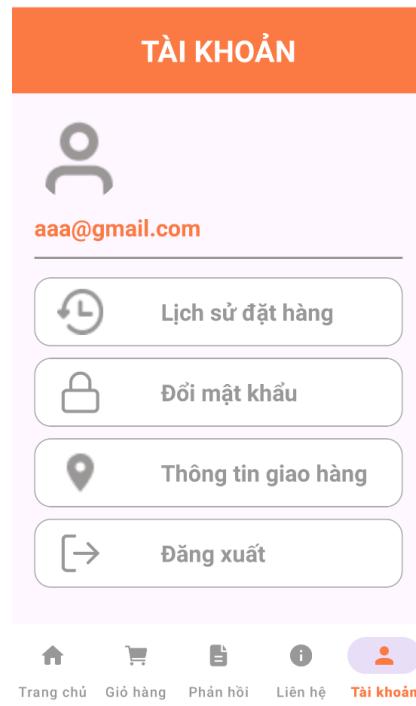
Hình 17: Giao diện phản hồi USER

### 3.3.11. Màn hình trang liên hệ user



Hình 18: Giao diện trang liên hệ USER

### 3.3.12. Màn hình trang tài khoản user



Hình 19: Giao diện trang tài khoản USER

### 3.3.13. Màn hình trang lịch sử đặt hàng



Hình 20: Giao diện trang lịch sử đặt hàng

### 3.3.14. Màn hình trang đổi mật khẩu

The screenshot shows a mobile application interface for changing a password. At the top, there is a navigation bar with a back arrow and the text 'ĐỔI MẬT KHẨU' (Change Password). Below this, the main title 'Đổi mật khẩu' is displayed in orange. The form contains three input fields: 'Mật khẩu cũ' (Old Password), 'Mật khẩu mới' (New Password), and 'Nhập lại mật khẩu mới' (Re-enter New Password). Each input field has an 'eye' icon to the right for password visibility. A large orange button at the bottom labeled 'ĐỔI MẬT KHẨU' (Change Password) is centered.

Hình 21: Giao diện trang đổi mật khẩu USER

### 3.3.15. Màn hình trang thêm món ngon admin

The screenshot shows a mobile application interface for adding a dish. At the top, there is a navigation bar with a back arrow and the text 'THÊM MÓN NGON' (Add Dish). The form consists of several input fields: 'Tên (\*)' (Name \*) with a placeholder 'Nhập tên món', 'Mô tả' (Description) with a placeholder 'Mô tả', 'Giá (\*)' (Price \*) with a placeholder 'Nhập giá' and a unit 'VNĐ', 'Khuyến mại' (Promotion) with a placeholder 'Nhập khuyến mại', and 'Ảnh chính (\*)' (Main Image \*) with a file upload icon. The entire form is set against a light pink background.

Hình 22: Giao diện thêm món ngon ADMIN

### 3.3.16. Màn hình trang chi tiết món ngon admin



Hình 23 Giao diện chi tiết món ngon ADMIN

### 3.3.17. Màn hình trang sửa món ngon admin

Tên (\*)  
Pizza size S

Mô tả  
Pizza là món ăn nổi tiếng của Ý với lớp

Giá (\*)  
99.000 VND

Khuyến mại  
10

Ảnh chính (\*)

Hình 24 Giao diện trang sửa món ngon ADMIN

### 3.3.18. Màn hình trang đồ uống admin



Hình 25 Giao diện trang đồ uống ADMIN

### 3.3.19. Màn hình trang thêm đồ uống admin

Tên (\*)  
Nhập tên món

Mô tả  
Mô tả

Giá (\*)  
Nhập giá VND

Khuyến mại  
Nhập khuyến mại

Ảnh chính (\*)  
[Upload icon]

Hình 26 Giao diện trang thêm đồ uống ADMIN

### 3.3.20. Màn hình trang chi tiết đồ uống admin



Hình 27 Giao diện trang chi tiết đồ uống ADMIN

### 3.3.21. Màn hình trang sửa đồ uống admin

Tên (\*)  
Matcha Latte

Mô tả  
Matcha latte là một loại đồ uống pha từ bột trà xanh matcha và sữa (có thể là sữa tươi, sữa hạt...), thường được khuấy đều tạo lớp bọt mịn. Đồ uống này có hương vị nhẹ nhàng, hơi đắng đặc trưng của matcha hòa quyện với ngọt của sữa, vừa thơm vừa giòn. Vừa giàu chất chống oxy hóa

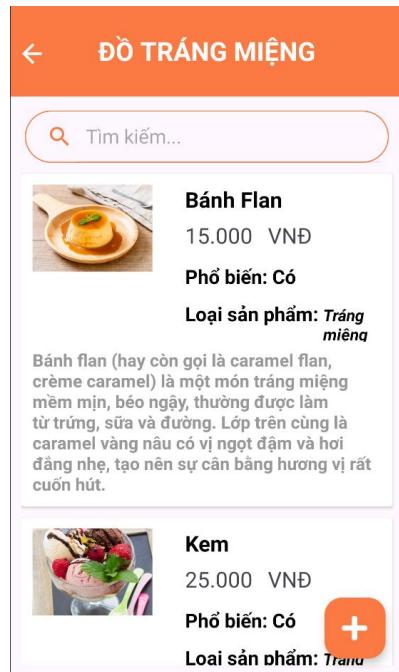
Giá (\*)  
55000 VND

Khuyến mãi  
Nhập khuyến mại

Ảnh chính (\*)

Hình 28 Giao diện trang sửa đồ uống ADMIN

### 3.3.22. Màn hình trang đồ tráng miệng admin



Hình 29 Giao diện trang đồ tráng miệng ADMIN

### 3.3.23. Màn hình trang thêm tráng miệng

This is a form for adding a new dessert. It includes fields for name, description, price, discount, and main image.

Thông tin	Nhập	Đơn vị
Tên (*)	Nhập tên món	VND
Mô tả	Mô tả	
Giá (*)	Nhập giá	VND
Khuyến mại	Nhập khuyến mại	
Ảnh chính (*)	Chọn ảnh	

Hình 30 Giao diện trang thêm tráng miệng ADMIN

### 3.3.24. Màn hình trang sửa đồ tráng miệng admin

The screenshot shows a mobile application interface for editing a dish. At the top, a red header bar contains the title 'SỬA TRÁNG MIỆNG' (Edit Dish) and a back arrow icon. Below the header, there are several input fields:

- Tên (\*)**: A text input field containing 'Bánh Flan'.
- Mô tả**: A text input field containing 'Bánh flan (hay còn gọi là caramel flan,'.
- Giá (\*)**: A text input field containing '15.000' with a unit indicator 'VND' to its right.
- Khuyến mại**: A text input field containing 'Nhập khuyến mại'.
- Ảnh chính (\*)**: A placeholder image showing a bowl of flan with a spoon.

Hình 31 Giao diện trang sửa đồ tráng miệng ADMIN

### 3.3.25. Màn hình trang doanh thu admin

The screenshot shows a mobile application interface for viewing revenue. At the top, a red header bar contains the title 'DOANH THU' and a back arrow icon. Below the header, there are two sections of order details, each with a timestamp and total amount:

Mã đơn hàng:	Ngày đặt hàng:	Tổng tiền
TEnRLoj0lc	2025-04-10 21:16:40	89,100 VND
gQIRk3xaJx	2025-04-10 12:19:31	336,300 VND

A large orange footer bar at the bottom displays the total revenue: 'Tổng doanh thu: 761,700 VND'.

Hình 32 Giao diện trang doanh thu ADMIN

## 3.4. Code minh họa các chức năng cốt lõi

### 3.4.1. Chức năng trang món ngon - USER

Android Studio screenshot showing two code editors side-by-side. Both editors are displaying Java code for the 'FoodPageUserActivity' class.

**Left Editor Content:**

```

29
30 public class FoodPageUserActivity extends AppCompatActivity {
31     3 usages
32     private RecyclerView rcv_food;
33     10 usages
34     private FoodUserRecyclerViewAdapter adapter;
35     2 usages
36     private ImageView img_back;
37     14 usages
38     private List<Product> list = new ArrayList<>();
39     2 usages
40     private SearchView searchView;
41
42     @Override
43     protected void onCreate(Bundle savedInstanceState) {
44         super.onCreate(savedInstanceState);
45         EdgeEffect edgeEffect = new EdgeEffect(this);
46         setOnFlingListener(new OnFlingListener() {
47             @Override
48             public void onFling(MotionEvent e1, MotionEvent e2, float velocity) {
49                 insets = insets.getInsets(WindowInsetsCompat.Type.systemBars());
50                 insets.systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
51                 rcv_food.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
52             }
53         });
54         rcv_food = findViewById(R.id.rcv_food);
55         img_back = findViewById(R.id.img_back);
56         searchView = findViewById(R.id.searchView);
57
58         // Thêm id SearchView để tim kiếm
59         searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
60             @Override
61             public boolean onQueryTextSubmit(String query) {
62                 filter(query);
63                 return false;
64             }
65
66             @Override
67             public boolean onQueryTextChange(String newText) {
68                 filter(newText);
69                 return false;
70             }
71         });
72
73         // Khởi tạo adapter
74         adapter = new FoodUserRecyclerViewAdapter(list, mContext, this);
75     }
76
77     // Khởi tạo adapter
78     adapter.setListFood(list);
79     rcv_food.setAdapter(adapter);
80     img_back.setOnClickListener(new View.OnClickListener() {
81         @Override
82         public void onClick(View v) {
83             finish();
84         }
85     });
86
87     list = new ArrayList<>();
88     getListFood();
89
90 }

```

**Right Editor Content:**

```

50
51 public class FoodPageUserActivity extends AppCompatActivity {
52     3 usages
53     protected void onCreate(Bundle savedInstanceState) {
54         super.onCreate(savedInstanceState);
55         adapter = new FoodUserRecyclerViewAdapter(list, mContext, this);
56         GridLayoutManager gridLayoutManager = new GridLayoutManager(mContext, 2);
57         rcv_food.setLayoutManager(gridLayoutManager);
58
59         adapter.setData(list);
60         rcv_food.setAdapter(adapter);
61         img_back.setOnClickListener(new View.OnClickListener() {
62             @Override
63             public void onClick(View v) {
64                 finish();
65             }
66         });
67         list = new ArrayList<>();
68         getListFood();
69
70 }
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111

```

```

30     public class FoodPageUserActivity extends AppCompatActivity {
31         private void getListFood() {
32             myRef.orderByChild("productType").equalTo("Món ngon").addChildEventListener(new ChildEventListener() {
33                 public void onChildAdded(@NotNull DataSnapshot snapshot, @Nullable String previousChildName) {
34                     adapter.setDataFood(list);
35                     adapter.notifyDataSetChanged();
36                 }
37             });
38         }
39
40         @Override
41         public void onChildRemoved(@NotNull DataSnapshot snapshot, @Nullable String previousChildName) {
42             // Xoá lg khi cò sđt kia da lìu
43             Product product = snapshot.getValue(Product.class);
44             if (product == null || list == null || list.isEmpty()) {
45                 return;
46             }
47             for(int i = 0; i < list.size(); i++) {
48                 if(product.getName() == list.get(i).getName()) {
49                     list.remove(list.get(i));
50                     break;
51                 }
52             }
53             adapter.notifyDataSetChanged();
54         }
55
56         @Override
57         public void onChildMoved(@NotNull DataSnapshot snapshot, @Nullable String previousChildName) {
58             // Xoá lg khi cò sđt da chayen do liu
59         }
60
61         @Override
62         public void onCancelled(@NotNull DatabaseError error) {
63             // Xoá lg khi cò lỗi
64         }
65     });
66
67     2 usages
68     private void filter(String text) {
69         List<Product> filteredList = new ArrayList<>();
70         for (Product item : list) {
71             if (item.getName().toLowerCase().contains(text.toLowerCase())) {
72                 filteredList.add(item);
73             }
74         }
75         adapter.setDataFood(filteredList);
76         adapter.notifyDataSetChanged();
77     }

```

Hình 33 Chức năng xem trang món ngon USER

### 3.4.2 Chức năng xem trang đồ uống - USER

```

30     public class DrinkPageUserActivity extends AppCompatActivity {
31
32         3 usages
33         private RecyclerView rcv_drug;
34         private DrinkUserRecycleViewAdapter adapter;
35         private ImageView img_back;
36
37         protected void onCreate(Bundle savedInstanceState) {
38             super.onCreate(savedInstanceState);
39             EdgesToEdge.enable((WindowManager) getSystemService.WindowManager);
40             setContentView(R.layout.activity_drink_page_user);
41             ViewCompat.setLayoutWindowInsetListener(findViewById(R.id.main), (View v, WindowInsetsCompat insets) -> {
42                 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
43                 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
44             });
45
46             rcv_drug = findViewById(R.id.rcv_drug);
47             img_back = findViewById(R.id.img_back);
48             searchView = findViewById(R.id.searchView);
49
50             // Hđc top adapter
51             adapter = new DrinkUserRecycleViewAdapter(list, mContext: this);
52             GridLayoutManager gridLayoutManager = new GridLayoutManager(mContext, spanCount: 2);
53             rcv_drug.setLayoutManager(gridLayoutManager);
54
55             getListDrink();
56             adapter.setDataAdapter(list);
57             rcv_drug.setAdapter(adapter);
58
59             img_back.setOnClickListener(new View.OnClickListener() {
60                 @Override
61                 public void onClick(View v) { finish(); }
62             });
63
64             // Thiết lập SearchView để tìm kiếm
65             searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
66                 @Override
67                 public boolean onQueryTextSubmit(String query) {
68                     filter(query);
69                 }
70             });
71

```

The screenshot shows the Android Studio interface with two main panes. The left pane displays the project structure for the 'FastFoodOrder' application, which includes packages for 'app', 'com.example.android.food\_app', and 'com.example.android.food\_app (test)'. The right pane shows the source code for the 'DrinkPageUserActivity.java' file.

```

public class DrinkPageUserActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_drink_page_user);
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public void onQueryTextChange(String newText) {
                filter(newText);
                return false;
            }
        });
    }

    private void getListDrink() {
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference("products");

        // Sắp xếp danh sách sản phẩm có trữ lượng popular = true
        myRef.orderByChild("productType").equalTo("Bé uống").addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
                Product product = snapshot.getValue(Product.class);
                if (product != null) {
                    list.add(product);
                    adapter.setDataDrink(list); // Cập nhật dữ liệu vào adapter
                    adapter.notifyDataSetChanged();
                }
            }

            @Override
            public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
                // Xử lý khi có sự thay đổi dữ liệu
                Product product = snapshot.getValue(Product.class);
                if (product != null) {
                    for (int i = 0; i < list.size(); i++) {
                        if (list.get(i).getProductName().equals(product.getProductName())) {
                            list.set(i, product);
                            break;
                        }
                    }
                }
                adapter.setDataDrink(list);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onChildRemoved(@NonNull DataSnapshot snapshot) {
                // Xử lý khi có sự xóa dữ liệu
                Product product = snapshot.getValue(Product.class);
                if (product == null || list == null || list.isEmpty()) {
                    return;
                }
                for (int i = 0; i < list.size(); i++) {
                    if (product.getProductName() == list.get(i).getProductName()) {
                        list.remove(list.get(i));
                        break;
                    }
                }
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
                // Xử lý khi có sự di chuyển dữ liệu
            }
        });

        myRef.orderByChild("productType").equalTo("Bé ăn").addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
                Product product = snapshot.getValue(Product.class);
                if (product != null) {
                    list.add(product);
                    adapter.setDataDrink(list); // Cập nhật dữ liệu vào adapter
                    adapter.notifyDataSetChanged();
                }
            }

            @Override
            public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
                // Xử lý khi có sự thay đổi dữ liệu
                Product product = snapshot.getValue(Product.class);
                if (product != null) {
                    for (int i = 0; i < list.size(); i++) {
                        if (list.get(i).getProductName().equals(product.getProductName())) {
                            list.set(i, product);
                            break;
                        }
                    }
                }
                adapter.setDataDrink(list);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onChildRemoved(@NonNull DataSnapshot snapshot) {
                // Xử lý khi có sự xóa dữ liệu
                Product product = snapshot.getValue(Product.class);
                if (product == null || list == null || list.isEmpty()) {
                    return;
                }
                for (int i = 0; i < list.size(); i++) {
                    if (product.getProductName() == list.get(i).getProductName()) {
                        list.remove(list.get(i));
                        break;
                    }
                }
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
                // Xử lý khi có sự di chuyển dữ liệu
            }
        });
    }
}

```

Hình 34 Chức năng xem trang đồ uống USER

### 3.4.3 Chức năng xem trang đồ uống miếng - USER

Android Studio screenshot showing two Java files side-by-side:

```

DessertPageUserActivity.java
public class DessertPageUserActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dessert_page_user);
        ViewCompat.setWindowInsetsInsetEdges(findViewById(R.id.main), ViewInsetsCompatInsets insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        };
        rcv_dessert = findViewById(R.id.rcv_dessert);
        img_back = findViewById(R.id.img_back);
        searchView = findViewById(R.id.searchView);
        rcv_dessert.setAdapter(adapter);
        adapter = new DessertUserRecyclerViewAdapter(list, mContext, this);
        GridLayoutManager gridLayoutManager = new GridLayoutManager(context, spanCount: 2);
        rcv_dessert.setLayoutManager(gridLayoutManager);
        getlistDesert();
        adapter.setDataSet(list);
        rcv_dessert.setAdapter(adapter);
        img_back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { finish(); }
        });
        // This id SearchView of tin hien
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String query) {
                filter(query);
                return false;
            }
        });
    }
    @Override
    public void onQueryTextChange(String newText) {
        filter(newText);
        return false;
    }
}

```

Detailed description: The screenshot shows the Android Studio interface with two Java files open. The left file is 'DessertPageUserActivity.java' and the right file is 'DrinkPageUserActivity.java'. Both files are part of the 'com.example.android\_food\_app' package. The code in both files is identical, showing the setup of a RecyclerView with a custom adapter and a search functionality. The code uses Java 8-style lambdas and annotations like @Override and @Override. The Android Studio interface includes toolbars, a navigation bar, and a status bar at the bottom.

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right.

**Project Structure:**

- app
  - manifests
  - java
    - ActivityAdmin
    - ActivityUser
     - DessertPageUserActivity
    - DrinkPageUserActivity
    - FoodPageUserActivity
    - HomeUserActivity
    - OrderHistoryActivity
    - ShipmentDetailPageActivity
    - UserFeedbackActivity
  - AdapterAdmin
  - AdapterUser
  - AnimationViewpager2
  - Fragment
  - FragmentAdmin
  - Model
  - MainActivity
  - com.example.android\_food\_app (androidTest)
  - ExampleInstrumentedTest
  - com.example.android\_food\_app (test)
  - java (generated)
- res
- color
- drawable
- font
- layout
  - activity\_add\_dessert\_admin.xml
  - activity\_add\_drink\_admin.xml
  - activity\_add\_food\_admin.xml
  - activity\_add\_shipment.xml
  - activity\_change\_password.xml
  - activity\_create\_order.xml
  - activity\_dessert\_page\_admin.xml
  - activity\_dessert\_page\_user.xml
  - activity\_detail\_page\_user.xml
  - activity\_detail\_product\_dessert\_admin.xml

**Code Editor (DesserPageUserActivity.java):**

```

public class DessertPageUserActivity extends AppCompatActivity {
    private void getListDesert() {
        myRef.orderByChild("path").equalTo("Trang miệng").addChildEventListener(new ChildEventListener() {
            ...
        });
    }

    @Override
    public void onChildRenoved(@NotNull DataSnapshot snapshot) {
        // Xo lý khi có sự xóa dữ liệu
        Product product = snapshot.getValue(Product.class);
        if (product == null || list == null || list.isEmpty()) {
            return;
        }
        for(int i = 0; i < list.size(); i++) {
            if(product.getName() == list.get(i).getName()){
                list.remove(list.get(i));
                break;
            }
        }
        adapter.notifyDataSetChanged();
    }

    @Override
    public void onChildMoved(@NotNull DataSnapshot snapshot, @Nullable String previousChildName) {
        // Xo lý khi có sự di chuyển dữ liệu
    }

    @Override
    public void onCancelled(@NotNull DatabaseError error) {
        // Xo lý khi có lỗi
    }
};

private void filter(String text) {
    List<Product> filteredList = new ArrayList<>();
    for (Product item : list) {
        if (item.getName().toLowerCase().contains(text.toLowerCase())) {
            filteredList.add(item);
        }
    }
    adapter.setDataSet(filteredList);
    adapter.notifyDataSetChanged();
}

```

Hình 35 Chức năng xem trang đồ tráng miệng – USER

### 3.4.4 Chức năng tạo đơn hàng - USER

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right.

**Project Structure:**

- app
  - manifests
  - java
    - ActivityAdmin
    - ActivityUser
     - CreateOrderActivity
    - DessertPageUserActivity
    - DrinkPageUserActivity
    - FoodPageUserActivity
    - HomeUserActivity
    - OrderHistoryActivity
    - ShipmentDetailPageActivity
    - UserFeedbackActivity
  - AdapterAdmin
  - AdapterUser
  - AnimationViewpager2
  - Fragment
  - FragmentAdmin
  - Model
  - MainActivity
  - com.example.android\_food\_app (androidTest)
  - ExampleInstrumentedTest
  - com.example.android\_food\_app (test)
  - java (generated)
- res
- color
- drawable
- font
- layout
  - activity\_add\_desert\_admin.xml
  - activity\_add\_drink\_admin.xml
  - activity\_add\_food\_admin.xml
  - activity\_add\_shipment.xml
  - activity\_change\_password.xml
  - activity\_create\_order.xml
  - activity\_dessert\_page\_admin.xml
  - activity\_dessert\_page\_user.xml
  - activity\_detail\_page\_user.xml
  - activity\_detail\_product\_dessert\_admin.xml

**Code Editor (CreateOrderActivity.java):**

```

public class CreateOrderActivity extends AppCompatActivity {
    private static final int REQUEST_CODE_SELECT_CUSTOMER = 1;

    TextView txtTotalAmount;
    private RecyclerView rvCreateOrder;
    private OrderUserAdapter orderUserAdapter;
    private Button btnCreateOrder, btn_customer;
    private EditText edt_name_order, edt_phone_order, edt_address_order;
    private ImageView img_back;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(findViewById(R.id.create_order));
        setContentView(LayoutInflater.from(this).inflate(R.layout.activity_create_order, null));
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (View v, WindowInsetsCompat insets) -> {
            Insets systemBars = Insets.getInsetsFromWindowInsetsCompat(insets);
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        txtTotalAmount = findViewById(R.id.txtTotalAmount);
        rvCreateOrder = findViewById(R.id.rvCreateOrder);
        btnCreateOrder = findViewById(R.id.btnCreateOrder);
        btn_customer = findViewById(R.id.btnCustomer);
        edt_name_order = findViewById(R.id.edt_name_order);
        edt_phone_order = findViewById(R.id.edt_phone_order);
        edt_address_order = findViewById(R.id.edt_address_order);
        // Get the total amount from the Intent
        double totalAmount = getIntent().getDoubleExtra("TOTAL_AMOUNT", defaultValue: 0.0) * 1000; // Nhân cho 1000 để có 3 số 0 ở cuối
        String formattedTotalAmount = String.format(Locale.getDefault(), "%,.3f VNĐ", totalAmount);
        txtTotalAmount.setText(formattedTotalAmount);

        rvCreateOrder.setLayoutManager(new LinearLayoutManager(this));
        orderUserAdapter = new OrderUserAdapter(this);
        rvCreateOrder.setAdapter(orderUserAdapter);

        btn_customer.setOnClickListener(v -> {
            Intent intent = new Intent(packageContext, CreateOrderActivity.this, ShipmentDetailPageActivity.class);
            startActivityForResult(intent, REQUEST_CODE_SELECT_CUSTOMER);
        });
    }
}

```

Android Studio screenshot showing two code editors side-by-side. The left editor shows the project structure for 'com.example.android\_food\_app' with various Java files like CreateOrderActivity.java, ShipmentDetailPageActivity.java, etc. The right editor shows the content of CreateOrderActivity.java.

```

public class CreateOrderActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_create_order);
    }
    btnCreateOrder = findViewById(R.id.btnCreateOrder);
    btnCreateOrder.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Lấy thông tin từ các input field
            String name = edt_name_order.getText().toString().trim();
            String phone = edt_phone_order.getText().toString().trim();
            String address = edt_address_order.getText().toString().trim();
            double total = getIntent().getDoubleExtra("TOTAL_AMOUNT", 0.0);
            double deliveryCost = 20000; // Chi phí giao hàng mặc định
            String currentDate = getCurrentDateTime(); // Lấy ngày giờ hiện tại
            // Lấy userID của tài khoản đang đăng nhập
            String userID = getCurrentUserID();
            // Tạo ID ngắn cho đơn hàng
            String orderId = generateRandomOrderId();
            // Trạng thái mới định sẵn đơn hàng
            String status = "Chưa xác nhận";
            // Tạo đối tượng Order mới
            Order newOrder = new Order(orderId, userID, name, phone, address, total, deliveryCost, currentDate, status);
            // Lưu đơn hàng vào Realtime Database (vd Firebase)
            DatabaseReference ordersRef = FirebaseDatabase.getInstance().getReference().child("Orders");
            ordersRef.child(userID).setValue(newOrder)
                .addOnSuccessListener(new OnSuccessListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        // Xử lý lỗi nếu có
                        Toast.makeText(getApplicationContext(), "Lỗi: " + e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        // Xử lý lỗi nếu có
                        Toast.makeText(getApplicationContext(), "Lỗi: " + e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                });
            // Thông báo thành công
            Toast.makeText(getApplicationContext(), "Bạn đã tạo đơn hàng thành công!", Toast.LENGTH_SHORT).show();
            // Cập nhật trạng thái đơn hàng
            saveOrderDetails(orderId);
            finish();
        }
    });
}

```

Below the code editor, the status bar shows: 21:02 LF UTF-8 4 spaces cf.

Second code editor (bottom):

```

public class CreateOrderActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        btnCreateOrder = findViewById(R.id.btnCreateOrder);
        btnCreateOrder.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Method to get current date and time
                usage
                private String getCurrentDateTime() {
                    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault());
                    return sdf.format(new Date());
                }
                // Method to get current user ID
                usage
                private String getCurrentUserID() {
                    FirebaseAuth user = FirebaseAuth.getInstance().getCurrentUser();
                    if (user != null) {
                        return user.getUid();
                    } else {
                        // Handle the case where user is not authenticated or null
                        return ""; // Or throw an exception, based on your app's logic
                    }
                }
                // Method to generate a random order ID
                usage
                private String generateRandomOrderId() {
                    String allowedChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
                    StringBuilder randomOrderId = new StringBuilder();
                    Random random = new Random();
                    for (int i = 0; i < 10; i++) {
                        randomOrderId.append(allowedChars.charAt(random.nextInt(allowedChars.length())));
                    }
                    return randomOrderId.toString();
                }
                // Usage
                private void saveOrderDetails(String orderId) {
                    DatabaseReference orderDetailsRef = FirebaseDatabase.getInstance().getReference().child("OrderDetails");
                    List<Product> productList = orderUserAdapter.getProductList(); // Get products from adapter

```

Below the second code editor, the status bar shows: 21:02 LF UTF-8 4 spaces cf.

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `CreateOrderActivity.java` file, which extends `AppCompatActivity`. The code handles saving order details to Firebase Database. It includes logic to get products from an adapter, create a new `OrderDetail` object, and save it to Firebase. It also includes `onSuccessListener` and  `onFailure` methods for handling database operations.

```

public class CreateOrderActivity extends AppCompatActivity {
    ...
    private void saveOrderDetails(String orderId) {
        DatabaseReference orderDetailsRef = FirebaseDatabase.getInstance().getReference().child("OrderDetails");
        List<Product> productList = orderUserAdapter.getProductList();
        ...
        OrderDetail orderDetail = new OrderDetail(orderId, product.getName(), quantity, totalPrice);
        ...
        String orderDetailId = orderDetailsRef.push().getKey(); // Generate unique key for OrderDetail
        orderDetailsRef.child(orderDetailId).setValue(orderDetail)
            .addOnSuccessListener(new OnSuccessListener() {
                ...
            })
            .addOnFailureListener(new OnFailureListener() {
                ...
            });
    }
}

```

Hình 36 Chức năng tạo đơn hàng - USER

### 3.4.5 Chức năng thêm địa chỉ - USER

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `AddShipmentActivity.java` file, which extends `AppCompatActivity`. The code handles adding a new shipment address. It retrieves the current user from Firebase Authentication and updates their address information. It also includes logic for handling system bars and displaying a toast message if the current user is null.

```

public class AddShipmentActivity extends AppCompatActivity {
    ...
    protected void onCreate(Bundle savedInstanceState) {
        ...
        FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();
        if (currentUser != null) {
            ...
        } else {
            Toast.makeText(getApplicationContext(), "Không thể lấy thông tin người dùng hiện tại.", Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

public class AddShipmentActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        btn_add.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // Lấy thông tin người dùng hiện tại từ Firebase Authentication
                FirebaseAuth mAuth = FirebaseAuth.getInstance();
                if (mAuth.getCurrentUser() != null) {
                    String email = mAuth.getCurrentUser().getEmail();
                    edit_email.setText(email);
                    Long id = Math.abs(UUID.randomUUID().getMostSignificantBits());
                    edit_id.setText(String.valueOf(id));
                    String name = edit_name.getText().toString();
                    String phone = edit_phone.getText().toString();
                    String address = edit_address.getText().toString();
                    Customer customer = new Customer(email, id, name, phone, address);
                    onClickAddCustomer(customer);
                } else {
                    Toast.makeText(getApplicationContext(), "Không thể lấy thông tin người dùng hiện tại.", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    private void onClickAddCustomer(Customer customer) {
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference("customers");

        // Đặt dữ liệu của khách hàng lên Firebase Realtime Database
        String pathObject = String.valueOf(customer.getId());
        myRef.child(pathObject).setValue(customer, new DatabaseReference.CompletionListener() {
            @Override
            public void onComplete(@Nullable DatabaseError error, @NonNull DatabaseReference ref) {
                if (error == null) {
                    Toast.makeText(getApplicationContext(), "Thêm Khách Hàng Thành Công!", Toast.LENGTH_SHORT).show();
                    finish();
                }
            }
        });
    }
}

```

Hình 37 Chức năng thêm địa chỉ - USER

### 3.4.6 Chức năng xem lịch sử đơn hàng - USER

```

public class OrderHistoryActivity extends AppCompatActivity {
    private RecyclerView recyclerViewOrderHistory;
    private OrderHistoryAdapter orderHistoryAdapter;
    private List<Order> orderList;
    private ImageButton imgBack;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_order_history);

        recyclerViewOrderHistory = findViewById(R.id.rv_order_history);
        orderHistoryAdapter = new OrderHistoryAdapter(orderList, new OrderHistoryAdapter.IClickListener() {
            no usage
            @Override
            public void onClickUpdateItem(Order order) {
            }
            no usage
            @Override
            public void onClickDeleteItem(Order order) {
            }
            @Override
            public void onClick(Order order) {
            }
        });
        recyclerViewOrderHistory.setLayoutManager(new LinearLayoutManager(this));
        recyclerViewOrderHistory.setAdapter(orderHistoryAdapter);
    }
}

```

```

OrderHistoryActivity.java
public class OrderHistoryActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        ...
        imgBack = findViewById(R.id.imgBack); // Ở đây là ID của ảnh
        imgBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { finish(); }
        });
        ...
    }
}

private void getListUserFromRealtimeDatabase() {
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef = database.getReference("Orders");
    ...
    myRef.orderByChild("userId").equalTo(currentUserId).addChildEventListener(new ChildEventListener() {
        @Override
        public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
            try {
                Order order = snapshot.getValue(Order.class);
                if (order != null) {
                    orderList.add(order);
                    orderHistoryAdapter.notifyDataSetChanged();
                }
            } catch (DatabaseException e) {
                e.printStackTrace();
            }
        }

        @Override
        public void onChildChanged(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
            ...
            Order order = snapshot.getValue(Order.class);
            if (order == null || orderList == null || orderList.isEmpty()) {
                return;
            }
            for (int i = 0; i < orderList.size(); i++) {
                if (Objects.equals(order.getId(), orderList.get(i).getId())) {
                    orderList.set(i, order);
                    break;
                }
            }
        }

        @Override
        public void onChildRemoved(@NonNull DataSnapshot snapshot) {
            ...
            Order order = snapshot.getValue(Order.class);
            if (order == null || orderList == null || orderList.isEmpty()) {
                return;
            }
            for (int i = 0; i < orderList.size(); i++) {
                if (Objects.equals(order.getId(), orderList.get(i).getId())) {
                    orderList.remove(i);
                    break;
                }
            }
            orderHistoryAdapter.notifyDataSetChanged();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            ...
        }
    });
}
}

OrderHistoryAdapter.java
public class OrderHistoryAdapter extends RecyclerView.Adapter {
    ...
    public void onBindViewHolder(@NonNull OrderViewHolder holder, int position) {
        ...
        Order order = orderList.get(position);
        if (order != null) {
            holder.txtOrderID.setText("Mã đơn hàng: " + order.getId());
            holder.txtOrderTime.setText("Thời gian đặt hàng: " + order.getTime());
            holder.txtOrderStatus.setText("Trạng thái: " + order.getStatus());
            holder.txtOrderAddress.setText("Địa chỉ: " + order.getAddress());
        }
    }

    ...
}

OrderHistoryAdapter.OrderViewHolder
class OrderViewHolder extends RecyclerView.ViewHolder {
    ...
}

```

Hình 38 Chức năng xem lịch sử đơn hàng - USER

### 3.4.7 Chức năng phản hồi - USER

```

public class UserFeedbackActivity extends AppCompatActivity {
    ...
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user_feedback);

        // Set up EdgeToEdge
        ViewCompat.setWindowInsetsListener(findViewById(R.id.main), (View v, WindowInsetsCompat insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("Loading feedback...");
        progressDialog.show();

        rvFeedback = findViewById(R.id.rv_feedback);
        mAdapter = new FeedbackUserAdapter(mcontext: this);
        rvFeedback.setAdapter(mAdapter);
        rvFeedback.setLayoutManager(new LinearLayoutManager(mcontext: this));

        fetchDataFromFirebase();
        imgBack = findViewById(R.id.imgBack); // Sử dụng ID đã định
        imgBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { finish(); }
        });
    }

    private void fetchDataFromFirebase() {
        FirebaseUser currentUser = FirebaseAuth.getInstance().getCurrentUser();
        if (currentUser != null) {
            DatabaseReference feedbackRef = FirebaseDatabase.getInstance().getReference("feedback");
            feedbackRef.addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    mListFeedback = new ArrayList();
                    progressDialog.dismiss();
                    for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
                        Feedback feedback = dataSnapshot.getValue(Feedback.class);
                        if (feedback != null && feedback.getUserId().equals(currentUser.getId())) {
                            mListFeedback.add(feedback);
                        }
                    }
                    mAdapter.notifyDataSetChanged();
                }

                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                    // Handle potential errors here
                    progressDialog.dismiss();
                }
            });
        } else {
            progressDialog.dismiss();
            // Handle case when user is not logged in
        }
    }
}

```

Hình 39 Chức năng phản hồi - USER

### 3.4.8 Chức năng xem trang món ngon - ADMIN

Android -> FoodPageAdminActivity.java

```

1 package com.example.android_food_app;
2
3 import android.os.Bundle;
4 import android.view.View;
5 import android.widget.Button;
6 import android.widget.EditText;
7 import android.widget.ListView;
8 import android.widget.SearchView;
9 import androidx.appcompat.app.AppCompatActivity;
10 import androidx.recyclerview.widget.LinearLayoutManager;
11 import androidx.recyclerview.widget.RecyclerView;
12 import com.google.firebase.database.DatabaseReference;
13 import com.google.firebase.database.FirebaseDatabase;
14 import com.google.firebase.database.Query;
15 import com.google.firebase.database.ValueEventListener;
16 import java.util.ArrayList;
17 import java.util.List;
18
19 public class FoodPageAdminActivity extends AppCompatActivity {
20     private RecyclerView rcv_food;
21     private List<Product> mListProduct;
22     private DatabaseReference databaseReference;
23     private ValueEventListener eventListener;
24     private FloatingActionButton fab_add;
25     private ImageButton imgBack;
26     private SearchView search_view;
27     private FoodAdminAdapter adapter;
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         EdgeToEdge.enable(this);
33         setContentView(R.layout.activity_food_page_admin);
34         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (View v, WindowInsetsCompat insets) -> {
35             insets.systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
36             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
37             return insets;
38         });
39
40         // init
41         rcv_food = findViewById(R.id.rcv_food);
42         fab_add = findViewById(R.id.fb_add);
43         imgBack = findViewById(R.id.fb_back);
44         search_view = findViewById(R.id.search_view);
45         search_view.clearFocus();
46
47         imgBack.setOnClickListener(new View.OnClickListener() {
48             @Override
49             public void onClick(View v) {
50                 Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
51                 startActivity(intent);
52             }
53         });
54
55         // An icon add để vào trangThêm
56         fab_add.setOnClickListener(new View.OnClickListener() {
57             @Override
58             public void onClick(View v) {
59                 Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
60                 startActivity(intent);
61             }
62         });
63
64         // load data
65         eventListener = query.addValueEventListener(new ValueEventListener() {
66             @Override
67             public void onDataChange(@NonNull DataSnapshot snapshot) {
68                 mListProduct.clear();
69                 for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
70                     Product product = dataSnapshot.getValue(Product.class);
71                     product.setKey(dataSnapshot.getKey());
72                     mListProduct.add(product);
73                 }
74                 adapter.notifyDataSetChanged();
75                 dialog.dismiss();
76             }
77
78             @Override
79             public void onCancelled(@NonNull DatabaseError error) {
80                 dialog.dismiss();
81             }
82         });
83
84         search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
85             @Override
86             public boolean onQueryTextSubmit(String query) {
87                 return false;
88             }
89
90             @Override
91             public boolean onQueryTextChange(String query) {
92                 return false;
93             }
94         });
95     }
96
97     private void setAdapter() {
98         LinearLayoutManager layoutManager = new LinearLayoutManager(this);
99         rcv_food.setLayoutManager(layoutManager);
100        rcv_food.setAdapter(adapter);
101    }
102
103    private void setListener() {
104        adapter.setOnItemClickListener(new FoodAdminAdapter.OnItemClickListener() {
105            @Override
106            public void onItemClick(int position) {
107                Intent intent = new Intent(getApplicationContext(), DetailFoodAdminActivity.class);
108                intent.putExtra("product", mListProduct.get(position));
109                startActivity(intent);
110            }
111        });
112    }
113
114    private void setFab() {
115        fab_add.setOnClickListener(new View.OnClickListener() {
116            @Override
117            public void onClick(View v) {
118                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
119                startActivity(intent);
120            }
121        });
122    }
123
124    private void setImgBack() {
125        imgBack.setOnClickListener(new View.OnClickListener() {
126            @Override
127            public void onClick(View v) {
128                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
129                startActivity(intent);
130            }
131        });
132    }
133
134    private void setSearch() {
135        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
136            @Override
137            public boolean onQueryTextSubmit(String query) {
138                return false;
139            }
140
141            @Override
142            public boolean onQueryTextChange(String query) {
143                return false;
144            }
145        });
146    }
147
148    private void setDialog() {
149        AlertDialog.Builder builder = new AlertDialog.Builder(this);
150        builder.setCancelable(false);
151        builder.setView(progress_admin_layout);
152        AlertDialog dialog = builder.create();
153        dialog.show();
154
155        mListProduct = new ArrayList<>();
156        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
157        rcv_food.setAdapter(adapter);
158
159        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
160        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
161        dialog.show();
162
163        // load data
164        eventListener = query.addValueEventListener(new ValueEventListener() {
165            @Override
166            public void onDataChange(@NonNull DataSnapshot snapshot) {
167                mListProduct.clear();
168                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
169                    Product product = dataSnapshot.getValue(Product.class);
170                    product.setKey(dataSnapshot.getKey());
171                    mListProduct.add(product);
172                }
173                adapter.notifyDataSetChanged();
174                dialog.dismiss();
175            }
176
177            @Override
178            public void onCancelled(@NonNull DatabaseError error) {
179                dialog.dismiss();
180            }
181        });
182
183        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
184            @Override
185            public boolean onQueryTextSubmit(String query) {
186                return false;
187            }
188
189            @Override
190            public boolean onQueryTextChange(String query) {
191                return false;
192            }
193        });
194    }
195
196    private void setImgBack() {
197        imgBack.setOnClickListener(new View.OnClickListener() {
198            @Override
199            public void onClick(View v) {
200                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
201                startActivity(intent);
202            }
203        });
204    }
205
206    private void setFab() {
207        fab_add.setOnClickListener(new View.OnClickListener() {
208            @Override
209            public void onClick(View v) {
210                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
211                startActivity(intent);
212            }
213        });
214    }
215
216    private void setSearch() {
217        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
218            @Override
219            public boolean onQueryTextSubmit(String query) {
220                return false;
221            }
222
223            @Override
224            public boolean onQueryTextChange(String query) {
225                return false;
226            }
227        });
228    }
229
230    private void setDialog() {
231        AlertDialog.Builder builder = new AlertDialog.Builder(this);
232        builder.setCancelable(false);
233        builder.setView(progress_admin_layout);
234        AlertDialog dialog = builder.create();
235        dialog.show();
236
237        mListProduct = new ArrayList<>();
238        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
239        rcv_food.setAdapter(adapter);
240
241        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
242        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
243        dialog.show();
244
245        // load data
246        eventListener = query.addValueEventListener(new ValueEventListener() {
247            @Override
248            public void onDataChange(@NonNull DataSnapshot snapshot) {
249                mListProduct.clear();
250                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
251                    Product product = dataSnapshot.getValue(Product.class);
252                    product.setKey(dataSnapshot.getKey());
253                    mListProduct.add(product);
254                }
255                adapter.notifyDataSetChanged();
256                dialog.dismiss();
257            }
258
259            @Override
260            public void onCancelled(@NonNull DatabaseError error) {
261                dialog.dismiss();
262            }
263        });
264
265        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
266            @Override
267            public boolean onQueryTextSubmit(String query) {
268                return false;
269            }
270
271            @Override
272            public boolean onQueryTextChange(String query) {
273                return false;
274            }
275        });
276    }
277
278    private void setImgBack() {
279        imgBack.setOnClickListener(new View.OnClickListener() {
280            @Override
281            public void onClick(View v) {
282                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
283                startActivity(intent);
284            }
285        });
286    }
287
288    private void setFab() {
289        fab_add.setOnClickListener(new View.OnClickListener() {
290            @Override
291            public void onClick(View v) {
292                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
293                startActivity(intent);
294            }
295        });
296    }
297
298    private void setSearch() {
299        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
300            @Override
301            public boolean onQueryTextSubmit(String query) {
302                return false;
303            }
304
305            @Override
306            public boolean onQueryTextChange(String query) {
307                return false;
308            }
309        });
310    }
311
312    private void setDialog() {
313        AlertDialog.Builder builder = new AlertDialog.Builder(this);
314        builder.setCancelable(false);
315        builder.setView(progress_admin_layout);
316        AlertDialog dialog = builder.create();
317        dialog.show();
318
319        mListProduct = new ArrayList<>();
320        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
321        rcv_food.setAdapter(adapter);
322
323        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
324        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
325        dialog.show();
326
327        // load data
328        eventListener = query.addValueEventListener(new ValueEventListener() {
329            @Override
330            public void onDataChange(@NonNull DataSnapshot snapshot) {
331                mListProduct.clear();
332                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
333                    Product product = dataSnapshot.getValue(Product.class);
334                    product.setKey(dataSnapshot.getKey());
335                    mListProduct.add(product);
336                }
337                adapter.notifyDataSetChanged();
338                dialog.dismiss();
339            }
340
341            @Override
342            public void onCancelled(@NonNull DatabaseError error) {
343                dialog.dismiss();
344            }
345        });
346
347        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
348            @Override
349            public boolean onQueryTextSubmit(String query) {
350                return false;
351            }
352
353            @Override
354            public boolean onQueryTextChange(String query) {
355                return false;
356            }
357        });
358    }
359
360    private void setImgBack() {
361        imgBack.setOnClickListener(new View.OnClickListener() {
362            @Override
363            public void onClick(View v) {
364                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
365                startActivity(intent);
366            }
367        });
368    }
369
370    private void setFab() {
371        fab_add.setOnClickListener(new View.OnClickListener() {
372            @Override
373            public void onClick(View v) {
374                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
375                startActivity(intent);
376            }
377        });
378    }
379
380    private void setSearch() {
381        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
382            @Override
383            public boolean onQueryTextSubmit(String query) {
384                return false;
385            }
386
387            @Override
388            public boolean onQueryTextChange(String query) {
389                return false;
390            }
391        });
392    }
393
394    private void setDialog() {
395        AlertDialog.Builder builder = new AlertDialog.Builder(this);
396        builder.setCancelable(false);
397        builder.setView(progress_admin_layout);
398        AlertDialog dialog = builder.create();
399        dialog.show();
400
401        mListProduct = new ArrayList<>();
402        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
403        rcv_food.setAdapter(adapter);
404
405        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
406        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
407        dialog.show();
408
409        // load data
410        eventListener = query.addValueEventListener(new ValueEventListener() {
411            @Override
412            public void onDataChange(@NonNull DataSnapshot snapshot) {
413                mListProduct.clear();
414                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
415                    Product product = dataSnapshot.getValue(Product.class);
416                    product.setKey(dataSnapshot.getKey());
417                    mListProduct.add(product);
418                }
419                adapter.notifyDataSetChanged();
420                dialog.dismiss();
421            }
422
423            @Override
424            public void onCancelled(@NonNull DatabaseError error) {
425                dialog.dismiss();
426            }
427        });
428
429        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
430            @Override
431            public boolean onQueryTextSubmit(String query) {
432                return false;
433            }
434
435            @Override
436            public boolean onQueryTextChange(String query) {
437                return false;
438            }
439        });
440    }
441
442    private void setImgBack() {
443        imgBack.setOnClickListener(new View.OnClickListener() {
444            @Override
445            public void onClick(View v) {
446                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
447                startActivity(intent);
448            }
449        });
450    }
451
452    private void setFab() {
453        fab_add.setOnClickListener(new View.OnClickListener() {
454            @Override
455            public void onClick(View v) {
456                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
457                startActivity(intent);
458            }
459        });
460    }
461
462    private void setSearch() {
463        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
464            @Override
465            public boolean onQueryTextSubmit(String query) {
466                return false;
467            }
468
469            @Override
470            public boolean onQueryTextChange(String query) {
471                return false;
472            }
473        });
474    }
475
476    private void setDialog() {
477        AlertDialog.Builder builder = new AlertDialog.Builder(this);
478        builder.setCancelable(false);
479        builder.setView(progress_admin_layout);
480        AlertDialog dialog = builder.create();
481        dialog.show();
482
483        mListProduct = new ArrayList<>();
484        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
485        rcv_food.setAdapter(adapter);
486
487        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
488        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
489        dialog.show();
490
491        // load data
492        eventListener = query.addValueEventListener(new ValueEventListener() {
493            @Override
494            public void onDataChange(@NonNull DataSnapshot snapshot) {
495                mListProduct.clear();
496                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
497                    Product product = dataSnapshot.getValue(Product.class);
498                    product.setKey(dataSnapshot.getKey());
499                    mListProduct.add(product);
500                }
501                adapter.notifyDataSetChanged();
502                dialog.dismiss();
503            }
504
505            @Override
506            public void onCancelled(@NonNull DatabaseError error) {
507                dialog.dismiss();
508            }
509        });
510
511        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
512            @Override
513            public boolean onQueryTextSubmit(String query) {
514                return false;
515            }
516
517            @Override
518            public boolean onQueryTextChange(String query) {
519                return false;
520            }
521        });
522    }
523
524    private void setImgBack() {
525        imgBack.setOnClickListener(new View.OnClickListener() {
526            @Override
527            public void onClick(View v) {
528                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
529                startActivity(intent);
530            }
531        });
532    }
533
534    private void setFab() {
535        fab_add.setOnClickListener(new View.OnClickListener() {
536            @Override
537            public void onClick(View v) {
538                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
539                startActivity(intent);
540            }
541        });
542    }
543
544    private void setSearch() {
545        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
546            @Override
547            public boolean onQueryTextSubmit(String query) {
548                return false;
549            }
550
551            @Override
552            public boolean onQueryTextChange(String query) {
553                return false;
554            }
555        });
556    }
557
558    private void setDialog() {
559        AlertDialog.Builder builder = new AlertDialog.Builder(this);
560        builder.setCancelable(false);
561        builder.setView(progress_admin_layout);
562        AlertDialog dialog = builder.create();
563        dialog.show();
564
565        mListProduct = new ArrayList<>();
566        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
567        rcv_food.setAdapter(adapter);
568
569        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
570        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
571        dialog.show();
572
573        // load data
574        eventListener = query.addValueEventListener(new ValueEventListener() {
575            @Override
576            public void onDataChange(@NonNull DataSnapshot snapshot) {
577                mListProduct.clear();
578                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
579                    Product product = dataSnapshot.getValue(Product.class);
580                    product.setKey(dataSnapshot.getKey());
581                    mListProduct.add(product);
582                }
583                adapter.notifyDataSetChanged();
584                dialog.dismiss();
585            }
586
587            @Override
588            public void onCancelled(@NonNull DatabaseError error) {
589                dialog.dismiss();
590            }
591        });
592
593        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
594            @Override
595            public boolean onQueryTextSubmit(String query) {
596                return false;
597            }
598
599            @Override
600            public boolean onQueryTextChange(String query) {
601                return false;
602            }
603        });
604    }
605
606    private void setImgBack() {
607        imgBack.setOnClickListener(new View.OnClickListener() {
608            @Override
609            public void onClick(View v) {
610                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
611                startActivity(intent);
612            }
613        });
614    }
615
616    private void setFab() {
617        fab_add.setOnClickListener(new View.OnClickListener() {
618            @Override
619            public void onClick(View v) {
620                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
621                startActivity(intent);
622            }
623        });
624    }
625
626    private void setSearch() {
627        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
628            @Override
629            public boolean onQueryTextSubmit(String query) {
630                return false;
631            }
632
633            @Override
634            public boolean onQueryTextChange(String query) {
635                return false;
636            }
637        });
638    }
639
640    private void setDialog() {
641        AlertDialog.Builder builder = new AlertDialog.Builder(this);
642        builder.setCancelable(false);
643        builder.setView(progress_admin_layout);
644        AlertDialog dialog = builder.create();
645        dialog.show();
646
647        mListProduct = new ArrayList<>();
648        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
649        rcv_food.setAdapter(adapter);
650
651        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
652        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
653        dialog.show();
654
655        // load data
656        eventListener = query.addValueEventListener(new ValueEventListener() {
657            @Override
658            public void onDataChange(@NonNull DataSnapshot snapshot) {
659                mListProduct.clear();
660                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
661                    Product product = dataSnapshot.getValue(Product.class);
662                    product.setKey(dataSnapshot.getKey());
663                    mListProduct.add(product);
664                }
665                adapter.notifyDataSetChanged();
666                dialog.dismiss();
667            }
668
669            @Override
670            public void onCancelled(@NonNull DatabaseError error) {
671                dialog.dismiss();
672            }
673        });
674
675        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
676            @Override
677            public boolean onQueryTextSubmit(String query) {
678                return false;
679            }
680
681            @Override
682            public boolean onQueryTextChange(String query) {
683                return false;
684            }
685        });
686    }
687
688    private void setImgBack() {
689        imgBack.setOnClickListener(new View.OnClickListener() {
690            @Override
691            public void onClick(View v) {
692                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
693                startActivity(intent);
694            }
695        });
696    }
697
698    private void setFab() {
699        fab_add.setOnClickListener(new View.OnClickListener() {
700            @Override
701            public void onClick(View v) {
702                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
703                startActivity(intent);
704            }
705        });
706    }
707
708    private void setSearch() {
709        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
710            @Override
711            public boolean onQueryTextSubmit(String query) {
712                return false;
713           }
714
715            @Override
716            public boolean onQueryTextChange(String query) {
717                return false;
718           }
719        });
720    }
721
722    private void setDialog() {
723        AlertDialog.Builder builder = new AlertDialog.Builder(this);
724        builder.setCancelable(false);
725        builder.setView(progress_admin_layout);
726        AlertDialog dialog = builder.create();
727        dialog.show();
728
729        mListProduct = new ArrayList<>();
730        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
731        rcv_food.setAdapter(adapter);
732
733        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
734        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
735        dialog.show();
736
737        // load data
738        eventListener = query.addValueEventListener(new ValueEventListener() {
739            @Override
740            public void onDataChange(@NonNull DataSnapshot snapshot) {
741                mListProduct.clear();
742                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
743                    Product product = dataSnapshot.getValue(Product.class);
744                    product.setKey(dataSnapshot.getKey());
745                    mListProduct.add(product);
746                }
747                adapter.notifyDataSetChanged();
748                dialog.dismiss();
749           }
750
751            @Override
752            public void onCancelled(@NonNull DatabaseError error) {
753                dialog.dismiss();
754           }
755        });
756
757        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
758            @Override
759            public boolean onQueryTextSubmit(String query) {
760                return false;
761           }
762
763            @Override
764            public boolean onQueryTextChange(String query) {
765                return false;
766           }
767        });
768    }
769
770    private void setImgBack() {
771        imgBack.setOnClickListener(new View.OnClickListener() {
772            @Override
773            public void onClick(View v) {
774                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
775                startActivity(intent);
776           }
777        });
778    }
779
780    private void setFab() {
781        fab_add.setOnClickListener(new View.OnClickListener() {
782            @Override
783            public void onClick(View v) {
784                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
785                startActivity(intent);
786           }
787        });
788    }
789
790    private void setSearch() {
791        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
792            @Override
793            public boolean onQueryTextSubmit(String query) {
794                return false;
795           }
796
797            @Override
798            public boolean onQueryTextChange(String query) {
799                return false;
800           }
801        });
802    }
803
804    private void setDialog() {
805        AlertDialog.Builder builder = new AlertDialog.Builder(this);
806        builder.setCancelable(false);
807        builder.setView(progress_admin_layout);
808        AlertDialog dialog = builder.create();
809        dialog.show();
810
811        mListProduct = new ArrayList<>();
812        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
813        rcv_food.setAdapter(adapter);
814
815        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
816        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
817        dialog.show();
818
819        // load data
820        eventListener = query.addValueEventListener(new ValueEventListener() {
821            @Override
822            public void onDataChange(@NonNull DataSnapshot snapshot) {
823                mListProduct.clear();
824                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
825                    Product product = dataSnapshot.getValue(Product.class);
826                    product.setKey(dataSnapshot.getKey());
827                    mListProduct.add(product);
828                }
829                adapter.notifyDataSetChanged();
830                dialog.dismiss();
831           }
832
833            @Override
834            public void onCancelled(@NonNull DatabaseError error) {
835                dialog.dismiss();
836           }
837        });
838
839        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
840            @Override
841            public boolean onQueryTextSubmit(String query) {
842                return false;
843           }
844
845            @Override
846            public boolean onQueryTextChange(String query) {
847                return false;
848           }
849        });
850    }
851
852    private void setImgBack() {
853        imgBack.setOnClickListener(new View.OnClickListener() {
854            @Override
855            public void onClick(View v) {
856                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
857                startActivity(intent);
858           }
859        });
860    }
861
862    private void setFab() {
863        fab_add.setOnClickListener(new View.OnClickListener() {
864            @Override
865            public void onClick(View v) {
866                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
867                startActivity(intent);
868           }
869        });
870    }
871
872    private void setSearch() {
873        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
874            @Override
875            public boolean onQueryTextSubmit(String query) {
876                return false;
877           }
878
879            @Override
880            public boolean onQueryTextChange(String query) {
881                return false;
882           }
883        });
884    }
885
886    private void setDialog() {
887        AlertDialog.Builder builder = new AlertDialog.Builder(this);
888        builder.setCancelable(false);
889        builder.setView(progress_admin_layout);
890        AlertDialog dialog = builder.create();
891        dialog.show();
892
893        mListProduct = new ArrayList<>();
894        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
895        rcv_food.setAdapter(adapter);
896
897        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
898        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
899        dialog.show();
900
901        // load data
902        eventListener = query.addValueEventListener(new ValueEventListener() {
903            @Override
904            public void onDataChange(@NonNull DataSnapshot snapshot) {
905                mListProduct.clear();
906                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
907                    Product product = dataSnapshot.getValue(Product.class);
908                    product.setKey(dataSnapshot.getKey());
909                    mListProduct.add(product);
910                }
911                adapter.notifyDataSetChanged();
912                dialog.dismiss();
913           }
914
915            @Override
916            public void onCancelled(@NonNull DatabaseError error) {
917                dialog.dismiss();
918           }
919        });
920
921        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
922            @Override
923            public boolean onQueryTextSubmit(String query) {
924                return false;
925           }
926
927            @Override
928            public boolean onQueryTextChange(String query) {
929                return false;
930           }
931        });
932    }
933
934    private void setImgBack() {
935        imgBack.setOnClickListener(new View.OnClickListener() {
936            @Override
937            public void onClick(View v) {
938                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
939                startActivity(intent);
940           }
941        });
942    }
943
944    private void setFab() {
945        fab_add.setOnClickListener(new View.OnClickListener() {
946            @Override
947            public void onClick(View v) {
948                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
949                startActivity(intent);
950           }
951        });
952    }
953
954    private void setSearch() {
955        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
956            @Override
957            public boolean onQueryTextSubmit(String query) {
958                return false;
959           }
960
961            @Override
962            public boolean onQueryTextChange(String query) {
963                return false;
964           }
965        });
966    }
967
968    private void setDialog() {
969        AlertDialog.Builder builder = new AlertDialog.Builder(this);
970        builder.setCancelable(false);
971        builder.setView(progress_admin_layout);
972        AlertDialog dialog = builder.create();
973        dialog.show();
974
975        mListProduct = new ArrayList<>();
976        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
977        rcv_food.setAdapter(adapter);
978
979        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
980        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
981        dialog.show();
982
983        // load data
984        eventListener = query.addValueEventListener(new ValueEventListener() {
985            @Override
986            public void onDataChange(@NonNull DataSnapshot snapshot) {
987                mListProduct.clear();
988                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
989                    Product product = dataSnapshot.getValue(Product.class);
990                    product.setKey(dataSnapshot.getKey());
991                    mListProduct.add(product);
992                }
993                adapter.notifyDataSetChanged();
994                dialog.dismiss();
995           }
996
997            @Override
998            public void onCancelled(@NonNull DatabaseError error) {
999                dialog.dismiss();
1000           }
1001        });
1002
1003        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
1004            @Override
1005            public boolean onQueryTextSubmit(String query) {
1006                return false;
1007           }
1008
1009            @Override
1010            public boolean onQueryTextChange(String query) {
1011                return false;
1012           }
1013        });
1014    }
1015
1016    private void setImgBack() {
1017        imgBack.setOnClickListener(new View.OnClickListener() {
1018            @Override
1019            public void onClick(View v) {
1020                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
1021                startActivity(intent);
1022           }
1023        });
1024    }
1025
1026    private void setFab() {
1027        fab_add.setOnClickListener(new View.OnClickListener() {
1028            @Override
1029            public void onClick(View v) {
1030                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
1031                startActivity(intent);
1032           }
1033        });
1034    }
1035
1036    private void setSearch() {
1037        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
1038            @Override
1039            public boolean onQueryTextSubmit(String query) {
1040                return false;
1041           }
1042
1043            @Override
1044            public boolean onQueryTextChange(String query) {
1045                return false;
1046           }
1047        });
1048    }
1049
1050    private void setDialog() {
1051        AlertDialog.Builder builder = new AlertDialog.Builder(this);
1052        builder.setCancelable(false);
1053        builder.setView(progress_admin_layout);
1054        AlertDialog dialog = builder.create();
1055        dialog.show();
1056
1057        mListProduct = new ArrayList<>();
1058        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
1059        rcv_food.setAdapter(adapter);
1060
1061        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
1062        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
1063        dialog.show();
1064
1065        // load data
1066        eventListener = query.addValueEventListener(new ValueEventListener() {
1067            @Override
1068            public void onDataChange(@NonNull DataSnapshot snapshot) {
1069                mListProduct.clear();
1070                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
1071                    Product product = dataSnapshot.getValue(Product.class);
1072                    product.setKey(dataSnapshot.getKey());
1073                    mListProduct.add(product);
1074                }
1075                adapter.notifyDataSetChanged();
1076                dialog.dismiss();
1077           }
1078
1079            @Override
1080            public void onCancelled(@NonNull DatabaseError error) {
1081                dialog.dismiss();
1082           }
1083        });
1084
1085        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
1086            @Override
1087            public boolean onQueryTextSubmit(String query) {
1088                return false;
1089           }
1090
1091            @Override
1092            public boolean onQueryTextChange(String query) {
1093                return false;
1094           }
1095        });
1096    }
1097
1098    private void setImgBack() {
1099        imgBack.setOnClickListener(new View.OnClickListener() {
1100            @Override
1101            public void onClick(View v) {
1102                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
1103                startActivity(intent);
1104           }
1105        });
1106    }
1107
1108    private void setFab() {
1109        fab_add.setOnClickListener(new View.OnClickListener() {
1110            @Override
1111            public void onClick(View v) {
1112                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
1113                startActivity(intent);
1114           }
1115        });
1116    }
1117
1118    private void setSearch() {
1119        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
1120            @Override
1121            public boolean onQueryTextSubmit(String query) {
1122                return false;
1123           }
1124
1125            @Override
1126            public boolean onQueryTextChange(String query) {
1127                return false;
1128           }
1129        });
1130    }
1131
1132    private void setDialog() {
1133        AlertDialog.Builder builder = new AlertDialog.Builder(this);
1134        builder.setCancelable(false);
1135        builder.setView(progress_admin_layout);
1136        AlertDialog dialog = builder.create();
1137        dialog.show();
1138
1139        mListProduct = new ArrayList<>();
1140        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
1141        rcv_food.setAdapter(adapter);
1142
1143        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
1144        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
1145        dialog.show();
1146
1147        // load data
1148        eventListener = query.addValueEventListener(new ValueEventListener() {
1149            @Override
1150            public void onDataChange(@NonNull DataSnapshot snapshot) {
1151                mListProduct.clear();
1152                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
1153                    Product product = dataSnapshot.getValue(Product.class);
1154                    product.setKey(dataSnapshot.getKey());
1155                    mListProduct.add(product);
1156                }
1157                adapter.notifyDataSetChanged();
1158                dialog.dismiss();
1159           }
1160
1161            @Override
1162            public void onCancelled(@NonNull DatabaseError error) {
1163                dialog.dismiss();
1164           }
1165        });
1166
1167        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
1168            @Override
1169            public boolean onQueryTextSubmit(String query) {
1170                return false;
1171           }
1172
1173            @Override
1174            public boolean onQueryTextChange(String query) {
1175                return false;
1176           }
1177        });
1178    }
1179
1180    private void setImgBack() {
1181        imgBack.setOnClickListener(new View.OnClickListener() {
1182            @Override
1183            public void onClick(View v) {
1184                Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
1185                startActivity(intent);
1186           }
1187        });
1188    }
1189
1190    private void setFab() {
1191        fab_add.setOnClickListener(new View.OnClickListener() {
1192            @Override
1193            public void onClick(View v) {
1194                Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
1195                startActivity(intent);
1196           }
1197        });
1198    }
1199
1200    private void setSearch() {
1201        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
1202            @Override
1203            public boolean onQueryTextSubmit(String query) {
1204                return false;
1205           }
1206
1207            @Override
1208            public boolean onQueryTextChange(String query) {
1209                return false;
1210           }
1211        });
1212    }
1213
1214    private void setDialog() {
1215        AlertDialog.Builder builder = new AlertDialog.Builder(this);
1216        builder.setCancelable(false);
1217        builder.setView(progress_admin_layout);
1218        AlertDialog dialog = builder.create();
1219        dialog.show();
1220
1221        mListProduct = new ArrayList<>();
1222        adapter = new FoodAdminAdapter(getApplicationContext(), FoodPageAdminActivity.this, mListProduct);
1223        rcv_food.setAdapter(adapter);
1224
1225        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("products");
1226        Query query = productsRef.orderByChild("productType").equalTo("Món ngon");
1227        dialog.show();
1228
1229        // load data
1230        eventListener = query.addValueEventListener(new ValueEventListener() {
1231            @Override
1232            public void onDataChange(@NonNull DataSnapshot snapshot) {
1233                mListProduct.clear();
1234                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
1235                    Product product = dataSnapshot.getValue(Product.class);
1236                    product.setKey(dataSnapshot.getKey());
1237                    mListProduct.add(product);
1238                }
1239                adapter.notifyDataSetChanged();
1240                dialog.dismiss();
1241           }
1242
1243            @Override
1244            public void onCancelled(@NonNull DatabaseError error) {
1245                dialog.dismiss();
1246           }
1247        });
1248
1249        search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
1250            @Override
1251            public boolean onQueryTextSubmit(String query) {
1252                return false;
1253           }
1254
1255            @Override
1256            public boolean onQueryTextChange(String query) {
1257                return false;
1258           }
1259        });
1260    }
1261
1262    private void setImgBack() {
1263        imgBack.setOnClickListener(new View.OnClickListener() {
1264            @Override
1265            public void onClick(View v
```

```

public class FoodPageAdminActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        eventListener = query.addSnapshotListener(eventValueListener);
        public void onEvent(@Nullable DocumentSnapshot documentSnapshot) {
            if (documentSnapshot != null & & documentSnapshot.exists()) {
                product.setKeyItemSnapshot(documentSnapshot);
                mListProduct.add(product);
                mAdapter.notifyDataSetChanged();
                dialog.dismiss();
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            dialog.dismiss();
        }
    }

    search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            return false;
        }

        @Override
        public boolean onQueryTextChange(String newText) {
            searchList(newText);
            return true;
        }
    });
}

public void searchList(String text) {
    ArrayList<Product> searchList = new ArrayList<>();
    for (Product product : mListProduct) {
        if (product.getProductName().toLowerCase().contains(text.toLowerCase())) {
            searchList.add(product);
        }
    }
    mAdapter.searchDataList(searchList);
}

```

Hình 40 Chức năng xem trang món ngon – ADMIN

### 3.4.9 Chức năng xem trang đồ uống - ADMIN

```

public class DrinkPageAdminActivity extends AppCompatActivity {
    private RecyclerView rv_drink;
    private List<Product> mListProduct;
    private DatabaseReference databaseReference;
    private ValueEventListener eventListener;
    private FloatingActionButton fab_add;
    private ImageButton imgBack;
    private SearchView search_view;
    private DrinkAdminAdapter adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable((HomeAdminActivity) this);
        setContentView(R.layout.activity_drink_admin);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (View v, WindowInsetsCompat insets) -> {
            insets.systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
        // anh xa
        rv_drink = findViewById(R.id.rv_drink);
        fab_add = findViewById(R.id.fab_add);
        imgBack = findViewById(R.id.imgBack);
        search_view = findViewById(R.id.search_view);
        search_view.clearFocus();
        imgBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(packageContext, DrinkPageAdminActivity.this, HomeAdminActivity.class);
                startActivity(intent);
            }
        });
        // ẩn icon add vào trang Thêm
        fab_add.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(packageContext, DrinkPageAdminActivity.this, AddDrinkAdminActivity.class);
                startActivity(intent);
            }
        });
    }
}

```

```

public class DrinkPageAdminActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        // Lấy icon add vào trong Theme
        fab_add.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), AddDrinkAdminActivity.class);
                startActivity(intent);
            }
        });

        GridLayoutManager gridLayoutManager = new GridLayoutManager(getApplicationContext(), spanCount);
        rv_drink.setLayoutManager(gridLayoutManager);

        AlertDialog.Builder builder = new AlertDialog.Builder(getApplicationContext());
        builder.setTitle("Đang tải");
        builder.setView(progress_admin_layout);
        AlertDialog dialog = builder.create();
        dialog.show();

        mListProduct = new ArrayList();
        adapter = new DrinkAdminAdapter(getApplicationContext(), DrinkPageAdminActivity.this, mListProduct);
        rv_drink.setAdapter(adapter);

        DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference("nhom_products");
        Query query = productsRef.orderByChild("nhom").equalTo("Bé uống");
        dialog.show();

        // Load data
        eventListener = query.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                mListProduct.clear();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
                    Product product = dataSnapshot.getValue(Product.class);
                    product.setKey(dataSnapshot.getKey());
                    mListProduct.add(product);
                }
                adapter.notifyDataSetChanged();
                dialog.dismiss();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                dialog.dismiss();
            }
        });
    }
}

```

```

public class DrinkPageAdminActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        eventListener = query.addValueEventListener(new ValueEventListener() {
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                mListProduct.clear();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()) {
                    Product product = dataSnapshot.getValue(Product.class);
                    product.setKey(dataSnapshot.getKey());
                    mListProduct.add(product);
                }
                adapter.notifyDataSetChanged();
                dialog.dismiss();
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                dialog.dismiss();
            }
        });
    }

    search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            return false;
        }

        @Override
        public boolean onQueryTextChange(String newText) {
            searchList(newText);
            return true;
        }
    });

    /**
     * Tạo list
     */
    public void searchList(String text) {
        ArrayList<Product> searchList = new ArrayList<>();
        for (Product product : mListProduct) {
            if (product.getNome().toLowerCase().contains(text.toLowerCase())) {
                searchList.add(product);
            }
        }
        adapter.searchDataList(searchList);
    }
}

```

Hình 41 Chức năng xem trang đồ uống - ADMIN

### 3.4.10 Chức năng xem trang trắng miệng - ADMIN

```

dAdminActivity.java UpdateDrinkAdminActivity.java UpdateDessertAdminActivity.java AddDessertAdminActivity.java AddFoodAdminActivity.java AddDrinkAdminActivity.java DetailProductDessertAdminActivity.java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeEffect.enable();
    setContentView(R.layout.activity_detail_product);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.detail_main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
}

// Anh xa
detail_name = findViewById(R.id.detail_name);
detail_desc = findViewById(R.id.detail_desc);
detail_price = findViewById(R.id.detail_price);
detail_sale = findViewById(R.id.detail_sale);
detail_popular = findViewById(R.id.detail_popular);
detail_product_type = findViewById(R.id.detail_product_type);
detail_laptop = findViewById(R.id.detail_laptop);
detail_laptopSlider = findViewById(R.id.detail_laptopSlider);
detail_laptopOther = findViewById(R.id.detail_laptopOther);
delete_btn = findViewById(R.id.delete_btn);
edit_btn = findViewById(R.id.edit_btn);

btn_back.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { finish(); }
});

// Khi click phan tu update
Bundle bundle= getIntent().getExtras();
if (bundle != null) {
    detail_name.setText(bundle.getString(key: "name"));
    detail_desc.setText(bundle.getString(key: "desc"));
    detail_price.setText(bundle.getString(key: "price"));
    detail_sale.setText(bundle.getString(key: "sale"));
    detail_popular.setText(bundle.getBoolean(key: "popular") ? "Có" : "Không");
    detail_product_type.setText(bundle.getString(key: "producttype"));
    detail_laptop.setText(bundle.getString(key: "laptop"));
    imgurl = bundle.getString(key: "imgurl");
    imgOtherUrl = bundle.getString(key: "imgother");
}

AAD_Project-main > app > src > main > java > com > example > android_food_app > ActivityAdmin > DetailProductDessertAdminActivity > imgOtherUrl
37:37 LF UTF-8 4 spaces

dAdminActivity.java UpdateDrinkAdminActivity.java UpdateDessertAdminActivity.java AddDessertAdminActivity.java AddFoodAdminActivity.java AddDrinkAdminActivity.java DetailProductDessertAdminActivity.java
public class DetailProductDessertAdminActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_product);
        imgurl = bundle.getString(key: "imgurl");
        imgOtherUrl = bundle.getString(key: "imgother");
        imgSliderUrl = bundle.getString(key: "imgSlider");
        Glide.with(this).load(bundle.getString(key: "imgurl")).into(detail_imgurl);
        Glide.with(this).load(bundle.getString(key: "imgSlider")).into(detail_lapSlider);
        Glide.with(this).load(bundle.getString(key: "imgOther")).into(detail_lapOther);
    }

    delete_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { showDeleteConfirmationDialog(); }
    });

    // Khi moi phan tu detail sang trang update
    edit_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { Intent intent = new Intent(getApplicationContext(), DetailProductDessertAdminActivity.this, UpdateDessertAdminActivity.class);
            intent.putExtra("name", detail_name.getText().toString());
            intent.putExtra("desc", detail_desc.getText().toString());
            intent.putExtra("price", detail_price.getText().toString());
            intent.putExtra("sale", detail_sale.getText().toString());
            intent.putExtra("popular", detail_popular.getText().toString());
            intent.putExtra("productType", detail_product_type.getText().toString());
            intent.putExtra("laptop", detail_laptop.getText().toString());
            intent.putExtra("imgurl", imgurl);
            intent.putExtra("laptop", imgSliderUrl);
            intent.putExtra("name", key);
            startActivity(intent);
        }
    });
}

// Phuong thuc hien thi hinh thoai xac nhan xoa san pham
private void showDeleteConfirmationDialog() {
    new AlertDialog.Builder(context: this)
        .setTitle("Xác nhận xóa")
        .setMessage("Bạn có chắc chắn muốn xóa sản phẩm này không?")
        .setPositiveButton(text: "Có", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) { deleteProduct(); }
        })
        .setNegativeButton(text: "Không", listener: null)
        .setOnCancelListener(R.drawable.ic_dialog_error)
        .show();
}

// Phuong thuc xoa san pham va cac anh lienn quan tu Firebase Storage va Realtime Database
private void deleteProduct() {
    DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReference(path: "products").child(key);
    databaseReference.removeValue().addOnSuccessListener(new OnSuccessListener() {
        @Override
        public void onSuccess(Void unused) {
            // Xoa anh chinh
            deleteImageFromStorageAndFinish(imgurl);
            // Xoa anh phu
            deleteImageFromStorageAndFinish(imgOtherUrl);
            // Xoa anh slider
            deleteImageFromStorageAndFinish(imgSliderUrl);
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(context: DetailProductDessertAdminActivity.this, text: "Xóa sản phẩm khỏi Realtime Database thất bại: " + e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
}

// Phuong thuc xoa anh tu Firebase Storage va ket thuc hoat dong neu thanh cong huoc that bai
private void deleteImageFromStorageAndFinish(String imgurl) {
}

```

Phát triển ứng dụng cho thiết bị di động

```

20  public class DetailProductDessertAdminActivity extends AppCompatActivity {
21
22     // Phương thức xóa sản phẩm và các ảnh liên quan từ Firebase Storage và Realtime Database
23
24     private void deleteProduct() {
25         DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReference("products").child(key);
26         databaseReference.removeValue().addOnSuccessListener(new OnSuccessListener() {
27             @Override
28             public void onSuccess(Void unused) {
29                 // Xóa ảnh chính
30                 deleteImageFromStorageAndFinish(imgUrl);
31                 // Xóa ảnh phụ
32                 deleteImageFromStorageAndFinish(imgOtherUrl);
33                 // Xóa ảnh slider
34                 deleteImageFromStorageAndFinish(imgSliderUrl);
35             }
36         }).addOnFailureListener(new OnFailureListener() {
37             @Override
38             public void onFailure(@NonNull Exception e) {
39                 Toast.makeText(context, DetailProductDessertAdminActivity.this, "Xóa sản phẩm khỏi Realtime Database thất bại: " + e.getMessage(), Toast.LENGTH_SHORT).show();
40             }
41         });
42     }
43
44     // Phương thức xóa ảnh từ Firebase Storage và kết thúc hàng đợi nếu thành công node thất bại
45     private void deleteImageFromStorageAndFinish(String imageUrl) {
46         FirebaseStorage storage = FirebaseStorage.getInstance();
47         StorageReference storageReference = storage.getReferenceFromUrl(imageUrl);
48         storageReference.delete().addOnSuccessListener(new OnSuccessListener() {
49             @Override
50             public void onSuccess(Void unused) {
51                 Toast.makeText(context, DetailProductDessertAdminActivity.this, "Xóa sản phẩm thành công", Toast.LENGTH_SHORT).show();
52                 startActivity(new Intent(getApplicationContext(), DessertPageAdminActivity.class));
53                 finish();
54             }
55         });
56     }
57 }
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

AAD\_Project-main > app > src > main > java > com > example > android\_food\_app > ActivityAdmin > DetailProductDessertAdminActivity > ImgOtherUrl

37:37 LF UTF-8 4 spaces

Hình 42 Chức năng xem trang trống miếng – ADMIN

### 3.4.11 Chức năng thêm món ngon - ADMIN

```

45  public class AddFoodAdminActivity extends AppCompatActivity {
46
47     @Override
48     protected void onCreate(Bundle savedInstanceState) {
49         super.onCreate(savedInstanceState);
50         EdgeEffect边缘效果
51         EdgeEffect边缘效果
52         setContentView(R.layout.activity_add_food_admin);
53         setActionBarTitle("Thêm món ăn");
54         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
55             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
56             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
57             return insets;
58         });
59
60         // Ann xu
61         edt_name = findViewById(R.id.edt_name);
62         edt_desc = findViewById(R.id.edt_desc);
63         edt_price = findViewById(R.id.edt_price);
64         edt_sale = findViewById(R.id.edt_sale);
65         imgUrl = findViewById(R.id.imgUrl);
66         imgSlider = findViewById(R.id.imgSlider);
67         imgOther = findViewById(R.id.imgOther);
68         rad_popular1 = findViewById(R.id.rad_popular1);
69         rad_popular2 = findViewById(R.id.rad_popular2);
70         btn_add = findViewById(R.id.btnAdd);
71         imgBack = findViewById(R.id.imgBack);
72         radioGroup_product_type = findViewById(R.id.radioGroup_product_type);
73
74         // tạo 1 node để lưu trữ giá trị id
75         currentReference = FirebaseDatabase.getInstance().getReference("currentProductId");
76         // Lấy giá trị currentProductId hiện tại
77         currentReference.addValueEventListener(new ValueEventListener() {
78             @Override
79             public void onDataChange(@NotNull DataSnapshot snapshot) {
80                 if (snapshot.exists()) {
81                     currentProductId = snapshot.getValue(Long.class);
82                 } else {
83                     currentProductId = 0; // Mặc định là 0 nếu chưa có sản phẩm nào
84                 }
85             }
86
87             @Override
88             public void onCancelled(@NotNull DatabaseError error) {
89                 Toast.makeText(context, AddFoodAdminActivity.this, error.getMessage(), Toast.LENGTH_SHORT).show();
90             }
91         });
92
93
94
95
96
97
98
99

```

AAD\_Project-main > app > src > main > java > com > example > android\_food\_app > ActivityAdmin > AddFoodAdminActivity

45:14 LF UTF-8 4 spaces

```

  45  public class AddFoodAdminActivity extends AppCompatActivity {
  46      protected void onCreate(Bundle savedInstanceState) {
  47
  48          // ActivityResultLauncher<Intent> mainImageResultLauncher = registerForActivityResult(
  49          //     new ActivityResultContracts.StartActivityForResult(),
  50          //     new ActivityResultContracts.StartActivityForResult() {
  51          //         @Override
  52          //         public void onActivityResult(ActivityResult result) {
  53          //             if (result.getResultCode() == Activity.RESULT_OK) {
  54          //                 Intent data = result.getData();
  55          //                 Uri mainImage = data.getData();
  56          //                 imgMain.setImageURI(Uri.parse(mainImage));
  57          //             } else {
  58          //                 Toast.makeText(context, AddFoodAdminActivity.this, "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
  59          //             }
  60          //         }
  61          //     );
  62
  63          // ActivityResultLauncher<Intent> sliderImageResultLauncher = registerForActivityResult(
  64          //     new ActivityResultContracts.StartActivityForResult(),
  65          //     new ActivityResultContracts.StartActivityForResult() {
  66          //         @Override
  67          //         public void onActivityResult(ActivityResult result) {
  68          //             if (result.getResultCode() == Activity.RESULT_OK) {
  69          //                 Intent data = result.getData();
  70          //                 Uri sliderImage = data.getData();
  71          //                 imgSlider.setImageURI(Uri.parse(sliderImage));
  72          //             } else {
  73          //                 Toast.makeText(context, AddFoodAdminActivity.this, "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
  74          //             }
  75          //         }
  76          //     );
  77
  78          // ActivityResultLauncher<Intent> otherImageResultLauncher = registerForActivityResult(
  79          //     new ActivityResultContracts.StartActivityForResult(),
  80          //     new ActivityResultContracts.StartActivityForResult() {
  81          //         @Override
  82          //         public void onActivityResult(ActivityResult result) {
  83          //             if (result.getResultCode() == Activity.RESULT_OK) {
  84          //                 Intent data = result.getData();
  85          //                 Uri otherImage = data.getData();
  86          //                 imgOther.setImageURI(Uri.parse(otherImage));
  87          //             } else {
  88          //                 Toast.makeText(context, AddFoodAdminActivity.this, "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
  89          //             }
  90          //         }
  91          //     );
  92
  93          imgMain.setOnClickListener(new View.OnClickListener() {
  94              @Override
  95              public void onClick(View v) {
  96                  Intent photoPicker = new Intent(Intent.ACTION_PICK);
  97                  photoPicker.setType("image/*");
  98                  mainImageResultLauncher.launch(photoPicker);
  99              }
  100         });
  101
  102         imgSlider.setOnClickListener(new View.OnClickListener() {
  103             @Override
  104             public void onClick(View v) {
  105                 Intent photoPicker = new Intent(Intent.ACTION_PICK);
  106                 photoPicker.setType("image/*");
  107                 sliderImageResultLauncher.launch(photoPicker);
  108             }
  109         });
  110
  111         imgOther.setOnClickListener(new View.OnClickListener() {
  112             @Override
  113             public void onClick(View v) {
  114                 Intent photoPicker = new Intent(Intent.ACTION_PICK);
  115                 photoPicker.setType("image/*");
  116                 otherImageResultLauncher.launch(photoPicker);
  117             }
  118         });
  119
  120         imgBack.setOnClickListener(new View.OnClickListener() {
  121             @Override
  122             public void onClick(View v) {
  123                 finish();
  124             }
  125         });
  126
  127         // Hộm upload ảnh chính lên Firebase
  128         private void saveMainImage() {
  129             if (uriMainImage == null) {
  130                 Toast.makeText(context, AddFoodAdminActivity.this, "Vui lòng chọn hình ảnh chính", Toast.LENGTH_SHORT).show();
  131                 return;
  132             }
  133
  134             StorageReference storageReference = FirebaseStorage.getInstance().getReference().child(pathString + "Save Images")
  135             .child(uriMainImage.getLastPathSegment());
  136
  137             AlertDialog.Builder builder = new AlertDialog.Builder(context, AddFoodAdminActivity.this);
  138             builder.setCancelable(false);
  139             builder.setView(R.layout.progress_admin_layout);
  140             AlertDialog dialog = builder.create();
  141             dialog.show();
  142
  143             storageReference.putFile(uriMainImage).addOnSuccessListener(new OnSuccessListener() {
  144                 @Override
  145                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  146                     Task<Uri> urlTask = taskSnapshot.getStorage().getDownloadUrl();
  147                     while (!urlTask.isComplete()) {
  148                         Uri downloadUrl = urlTask.getResult();
  149                     }
  150                 }
  151             });
  152         }
  153
  154     }
  155 }

```

```

  45     public class AddFoodAdminActivity extends AppCompatActivity {
  46         private void saveMainImage() {
  47             storageReference.putFile(UriMainImage).addOnSuccessListener(new OnSuccessListener<TaskSnapshot>() {
  48                 @Override
  49                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  50                     Uri urlTask = taskSnapshot.getStorage().getDownloadUrl();
  51                     while (!urlTask.isComplete());
  52                     Uri urlImage = urlTask.getResult();
  53                     imgMainURL = urlImage.toString();
  54                     saveSliderImage(dialog);
  55                 }
  56             }).addOnFailureListener(new OnFailureListener() {
  57                 @Override
  58                 public void onFailure(@NonNull Exception e) {
  59                     dialog.dismiss();
  60                     Toast.makeText(context, AddFoodAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
  61                 }
  62             });
  63         }
  64         // Hm upload ảnh slider lên Firebase
  65         private void saveSliderImage(AlertDialog dialog) {
  66             if (UriSliderImage == null) {
  67                 Toast.makeText(context, AddFoodAdminActivity.this, "Vui lòng chọn hình ảnh slider", Toast.LENGTH_SHORT).show();
  68                 dialog.dismiss();
  69                 return;
  70             }
  71             StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("anhHinh").child(UriSliderImage.getLastPathSegment());
  72             storageReference.putFile(UriSliderImage).addOnSuccessListener(new OnSuccessListener<TaskSnapshot>() {
  73                 @Override
  74                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  75                     Uri urlTask = taskSnapshot.getStorage().getDownloadUrl();
  76                     while (!urlTask.isComplete());
  77                     Uri urlImage = urlTask.getResult();
  78                     imgSliderURL = urlImage.toString();
  79                     saveOtherImage(dialog);
  80                 }
  81             }).addOnFailureListener(new OnFailureListener() {
  82                 @Override
  83                 public void onFailure(@NonNull Exception e) {
  84                     dialog.dismiss();
  85                     Toast.makeText(context, AddFoodAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
  86                 }
  87             });
  88         }
  89         // Hm upload ảnh khác lên Firebase
  90         private void saveOtherImage(AlertDialog dialog) {
  91             if (UriOtherImage == null) {
  92                 Toast.makeText(context, AddFoodAdminActivity.this, "Vui lòng chọn hình ảnh khác", Toast.LENGTH_SHORT).show();
  93                 dialog.dismiss();
  94                 return;
  95             }
  96             StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("anhHinh").child(UriOtherImage.getLastPathSegment());
  97             storageReference.putFile(UriOtherImage).addOnSuccessListener(new OnSuccessListener<TaskSnapshot>() {
  98                 @Override
  99                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  100                     Uri urlTask = taskSnapshot.getStorage().getDownloadUrl();
  101                     while (!urlTask.isComplete());
  102                     Uri urlImage = urlTask.getResult();
  103                     imgOtherURL = urlImage.toString();
  104                     uploadData(dialog);
  105                 }
  106             }).addOnFailureListener(new OnFailureListener() {
  107                 @Override
  108                 public void onFailure(@NonNull Exception e) {
  109                     dialog.dismiss();
  110                     Toast.makeText(context, AddFoodAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
  111                 }
  112             });
  113         }
  114     }
  115 }

  AAD_Project-main > app > src > main > java > com > example > android_food_app > ActivityAdmin > AddFoodAdminActivity

```

```

45  public class AddFoodAdminActivity extends AppCompatActivity {
46      // Màn upload đồ ăn phryn tên Firebase
47      private void uploadData(AlertDialog dialog) {
48          try {
49              String name = edt_name.getText().toString().trim();
50              String desc = edt_desc.getText().toString().trim();
51              String price = edt_price.getText().toString().trim();
52              String sale = edt_sale.getText().toString().trim();
53              Boolean popular = rad_popular.isChecked(); // Lấy giá trị từ RadioButton
54
55              // Kiểm tra các trường dữ liệu
56              if (name.isEmpty()) {
57                  throw new IllegalArgumentException("Vui lòng nhập tên sản phẩm");
58              } else if (desc.isEmpty()) {
59                  throw new IllegalArgumentException("Vui lòng nhập mô tả sản phẩm");
60              } else if (price.isEmpty()) {
61                  throw new IllegalArgumentException("Vui lòng nhập giá sản phẩm");
62              }
63
64              // Tính priceNew nếu có giảm giá
65              double priceOld = Double.parseDouble(price.replaceAll("\\s+", "")); // Xóa khoảng trắng
66              double priceNew = priceOld; // Giá gốc và giá mới
67              if (!sale.isEmpty()) { // Nếu có giảm giá
68                  int salePercent = Integer.parseInt(sale.replaceAll("\\s+", "")); // Xóa khoảng trắng
69                  priceNew = priceOld * (100 - salePercent) / 100;
70              }
71
72              // Format priceNew để lưu vào Firebase
73              DecimalFormat formatter = new DecimalFormat("###,###");
74              String priceNewFormatted = formatter.format(priceNew);
75
76              // Lấy ID của nhóm đã chọn từ RadioButton
77              int selectedProductType = radioGroup_product_type.getCheckedRadioButtonId();
78              RadioButton selectedProductTypeRadioButton = findViewById(selectedProductType);
79              String productType = selectedProductTypeRadioButton.getText().toString();
80
81              // Tạo một sản phẩm mới với ID tự động
82              Product product = new Product();
83              product.setName(name);
84              product.setDesc(desc);
85              product.setPrice(price);
86
87              // Set các giá trị
88              product.setSale(sale);
89              product.setImgMainUrl(imgMainURL);
90              product.setImgSliderUrl(imgSliderURL);
91              product.setPopular(popular);
92              product.setProductType(productType);
93              product.setImgOtherUrl(imgOtherURL);
94
95              // Set ID thay thế I++ cho node trên firebase để lưu trữ
96              currentProductId++;
97              currentReference.setValue(currentProductId);
98
99              // Tạo ID trên giá trị node currentProductId
100             String id = String.valueOf(currentProductId);
101
102             FirebaseDatabase.getInstance().getReference("products").child(id)
103                 .setValue(product).addOnCompleteListener(new OnCompleteListener() {
104                     @Override
105                     public void onComplete(@NonNull Task<Void> task) {
106                         if (task.isSuccessful()) {
107                             Toast.makeText(getApplicationContext(), "Thêm sản phẩm thành công", Toast.LENGTH_SHORT).show();
108                             dialog.dismiss();
109                             Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.this, FoodPageAdminActivity.class);
110                             startActivity(intent);
111                         }
112                     }
113                 }).addOnFailureListener(new OnFailureListener() {
114                     @Override
115                     public void onFailure(@NonNull Exception e) {
116                         dialog.dismiss();
117                         Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
118                     }
119                 });
120             }
121         }
122     }
123
124     catch (IllegalArgumentException e) {
125         dialog.dismiss();
126         Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
127     }
128     catch (Exception e) {
129         dialog.dismiss();
130         Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
131     }
132 }

```

Hình 43 Chức năng thêm món ngon - ADMIN

### 3.4.12 Chức năng thêm đồ uống - ADMIN

```

m: README.md ⑥ UpdateFoodAdminActivity.java ⑦ UpdateDrinkAdminActivity.java ⑧ UpdateDessertAdminActivity.java ⑨ AddDessertAdminActivity.java ⑩ AddFoodAdminActivity.java ⑪ AddDrinkAdminActivity.java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add_drink_admin);
    ViewCompat.setNestedScrollingEnabled(false, findViewById(R.id.main));
    insets = ViewCompat.getRootWindowInsets(findViewById(R.id.main));
    insets.getInsets(ViewCompat.Type.systemBars).left, insets.getInsets(ViewCompat.Type.systemBars).top, insets.getInsets(ViewCompat.Type.systemBars).right, insets.getInsets(ViewCompat.Type.systemBars).bottom);
    return insets;
}

// ảnh xa
ext_main = findViewById(R.id.ext_main);
ext_price = findViewById(R.id.ext_price);
ext_sale = findViewById(R.id.ext_sale);
imgUrl = findViewById(R.id.imgUrl);
imgSlider = findViewById(R.id.imgSlider);
imgOther = findViewById(R.id.imgOther);
radio_group_product_type = findViewById(R.id.radioGroup_product_type);

// tao I need de luu tru giua tri id
currentProductReference = FirebaseDatabase.getInstance().getReference("path").child("currentProductId");
// Ldg giua tri currentProductReference tren tai
currentProductReference.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        if (snapshot.exists()) {
            currentProductId = snapshot.getValue(Long.class);
        } else {
            currentProductId = 0; // Mac dinh la 0 neu chua co san pham nao
        }
    }
});

@Override
public void onCancelled(@NonNull DatabaseError error) {
    Toast.makeText(context, AddDrinkAdminActivity.this, error.getMessage(), Toast.LENGTH_SHORT).show();
}

```

AAD\_Project-main > app > src > main > java > com > example > android\_food\_app > ActivityAdmin > AddDrinkAdminActivity

---

```

m: README.md ⑥ UpdateFoodAdminActivity.java ⑦ UpdateDrinkAdminActivity.java ⑧ UpdateDessertAdminActivity.java ⑨ AddDessertAdminActivity.java ⑩ AddFoodAdminActivity.java ⑪ AddDrinkAdminActivity.java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // ActivityResultLauncher che ảnh chính
    ActivityResultLauncher<Intent> mainImageResultLauncher = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        new ActivityResultCallback<ActivityResult>() {
            @Override
            public void onActivityResult(ActivityResult result) {
                if (result.getResultCode() == Activity.RESULT_OK) {
                    Intent data = result.getData();
                    Uri mainImage = data.getData();
                    imgUrl.setImageURI(Uri.parse(mainImage));
                } else {
                    Toast.makeText(context, AddDrinkAdminActivity.this, "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
                }
            }
        });
    // ActivityResultLauncher che ảnh slider
    ActivityResultLauncher<Intent> sliderImageResultLauncher = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        new ActivityResultCallback<ActivityResult>() {
            @Override
            public void onActivityResult(ActivityResult result) {
                if (result.getResultCode() == Activity.RESULT_OK) {
                    Intent data = result.getData();
                    Uri sliderImage = data.getData();
                    imgSlider.setImageURI(Uri.parse(sliderImage));
                } else {
                    Toast.makeText(context, AddDrinkAdminActivity.this, "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
                }
            }
        });
    // ActivityResultLauncher che ảnh khác
    ActivityResultLauncher<Intent> otherImageResultLauncher = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        new ActivityResultCallback<ActivityResult>() {
            @Override
            public void onActivityResult(ActivityResult result) {
                if (result.getResultCode() == Activity.RESULT_OK) {
                    Intent data = result.getData();
                    Uri otherImage = data.getData();
                    imgOther.setImageURI(Uri.parse(otherImage));
                } else {
                    Toast.makeText(context, AddDrinkAdminActivity.this, "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
                }
            }
        });
    imgUrl.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent photoPicker = new Intent(Intent.ACTION_PICK);
            photoPicker.setType("image/*");
            mainImageResultLauncher.launch(photoPicker);
        }
    });

    imgSlider.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent photoPicker = new Intent(Intent.ACTION_PICK);
            photoPicker.setType("image/*");
            sliderImageResultLauncher.launch(photoPicker);
        }
    });

    imgOther.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent photoPicker = new Intent(Intent.ACTION_PICK);
        }
    });
}

```

AAD\_Project-main > app > src > main > java > com > example > android\_food\_app > ActivityAdmin > AddDrinkAdminActivity

```

  47     public class AddDrinkAdminActivity extends AppCompatActivity {
  48         protected void onCreate(Bundle savedInstanceState) {
  49
  50             imgOther.setOnClickListener(new View.OnClickListener() {
  51                 @Override
  52                 public void onClick(View v) {
  53                     Intent photoPicker = new Intent(Intent.ACTION_PICK);
  54                     photoPicker.setType("image/*");
  55                     otherImageResultLauncher.launch(photoPicker);
  56                 }
  57             });
  58
  59             // add data
  60             btn_add.setOnClickListener(new View.OnClickListener() {
  61                 @Override
  62                 public void onClick(View v) { saveMainImage(); }
  63             });
  64
  65             imgBack.setOnClickListener(new View.OnClickListener() {
  66                 @Override
  67                 public void onClick(View v) { finish(); }
  68             });
  69
  70         }
  71
  72         // Hàm upload ảnh chính lên Firebase
  73         // Lỗi
  74         private void saveMainImage() {
  75             if (uriMainImage == null) {
  76                 Toast.makeText(context, AddDrinkAdminActivity.this, "Vui lòng chọn hình ảnh chính", Toast.LENGTH_SHORT).show();
  77                 return;
  78             }
  79
  80             StorageReference storageReference = FirebaseStorage.getInstance().getReference().child(uriMainImage.getLastPathSegment());
  81
  82             AlertDialog.Builder builder = new AlertDialog.Builder(context, AddDrinkAdminActivity.this);
  83             builder.setCancelable(false);
  84             builder.setView(R.layout.progress_admin_layout);
  85             AlertDialog dialog = builder.create();
  86             dialog.show();
  87
  88             storageReference.putFile(uriMainImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
  89                 @Override
  90
  91             public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  92                 Task<Uri> urlTask = taskSnapshot.getStorage().getDownloadUrl();
  93                 while (!urlTask.isComplete());
  94                 Uri urlImage = urlTask.getResult();
  95                 imgMainImage = urlImage.toString();
  96                 saveSliderImage(dialog);
  97             }
  98             // addOnFailureListener(new OnFailureListener() {
  99                 @Override
 100                 public void onFailure(@NonNull Exception e) {
 101                     dialog.dismiss();
 102                     Toast.makeText(context, AddDrinkAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
 103                 }
 104             });
 105
 106         }
 107
 108         // Hàm upload ảnh slider lên Firebase
 109         // Lỗi
 110         private void saveSliderImage(AlertDialog dialog) {
 111             if (uriSliderImage == null) {
 112                 Toast.makeText(context, AddDrinkAdminActivity.this, "Vui lòng chọn hình ảnh slider", Toast.LENGTH_SHORT).show();
 113                 dialog.dismiss();
 114                 return;
 115             }
 116
 117             StorageReference storageReference = FirebaseStorage.getInstance().getReference().child(uriSliderImage.getLastPathSegment());
 118
 119             storageReference.putFile(uriSliderImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
 120                 @Override
 121                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
 122                     Task<Uri> urlTask = taskSnapshot.getStorage().getDownloadUrl();
 123                     while (!urlTask.isComplete());
 124                     Uri urlImage = urlTask.getResult();
 125                     imgSliderURI = urlImage.toString();
 126                     saveOtherImage(dialog);
 127                 }
 128             }.addOnFailureListener(new OnFailureListener() {
 129                 @Override
 130
 131             public void onFailure(@NonNull Exception e) {
 132                 dialog.dismiss();
 133                 Toast.makeText(context, AddDrinkAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
 134             }
 135         });
 136
 137         // Hàm upload ảnh khác lên Firebase
 138         // Lỗi
 139         private void saveOtherImage(AlertDialog dialog) {
 140             if (uriOtherImage == null) {
 141                 Toast.makeText(context, AddDrinkAdminActivity.this, "Vui lòng chọn hình ảnh khác", Toast.LENGTH_SHORT).show();
 142                 dialog.dismiss();
 143                 return;
 144             }
 145
 146             StorageReference storageReference = FirebaseStorage.getInstance().getReference().child(uriOtherImage.getLastPathSegment());
 147
 148             storageReference.putFile(uriOtherImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
 149                 @Override
 150                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
 151                     Task<Uri> urlTask = taskSnapshot.getStorage().getDownloadUrl();
 152                     while (!urlTask.isComplete());
 153                     Uri urlImage = urlTask.getResult();
 154                     imgOtherImage = urlImage.toString();
 155                     uploadData(dialog);
 156                 }
 157             }.addOnFailureListener(new OnFailureListener() {
 158                 @Override
 159                 public void onFailure(@NonNull Exception e) {
 160                     dialog.dismiss();
 161                     Toast.makeText(context, AddDrinkAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
 162                 }
 163             });
 164
 165         }
 166
 167         // Hàm upload dữ liệu sản phẩm lên Firebase
 168
 169     }

```

Phát triển ứng dụng cho thiết bị di động

```

 47     public class AddDrinkAdminActivity extends AppCompatActivity {
 48         // Hàm upload dữ liệu sản phẩm lên Firebase
 49         private void uploadData(AlertDialog dialog) {
 50             try {
 51                 String name = edit_name.getText().toString().trim();
 52                 String desc = edit_desc.getText().toString().trim();
 53                 String price = edit_price.getText().toString().trim();
 54                 String sale = edit_sale.getText().toString().trim();
 55                 Boolean popular = edit_popular.isChecked(); // Lấy giá trị từ RadioButton
 56
 57                 // Kiểm tra các trường dữ liệu
 58                 if (name.isEmpty()) {
 59                     throw new IllegalStateException("Vui lòng nhập tên sản phẩm");
 60                 } else if (desc.isEmpty()) {
 61                     throw new IllegalStateException("Vui lòng nhập mô tả sản phẩm");
 62                 } else if (price.isEmpty()) {
 63                     throw new IllegalStateException("Vui lòng nhập giá sản phẩm");
 64                 }
 65
 66                 // Tính priceNew nếu có giảm giá
 67                 double priceOld = Double.parseDouble(price.replaceAll("\\s+", " ")); // Xóa khoảng trắng
 68                 double priceNew = priceOld; // Gia gốc
 69                 if (sale.length() > 0) { // Nếu có giảm giá
 70                     int salePercent = Integer.parseInt(sale.replaceAll("\\s+", " ")); // Xóa khoảng trắng
 71                     priceNew = priceOld * (100 - salePercent) / 100;
 72
 73                 // Format priceNew để lưu vào Firebase
 74                 DecimalFormat formatter = new DecimalFormat("###,###");
 75                 String priceNewFormatted = formatter.format(priceNew);
 76
 77                 // Lấy type sản phẩm để chọn vào RadioButton
 78                 int selectedProductId = radioButton_product_type.getCheckedRadioButtonId();
 79                 RadioButton selectedProductTypeRadioButton = findViewById(selectedProductId);
 80                 String productType = selectedProductTypeRadioButton.getText().toString();
 81
 82                 // Thêm dữ liệu, upload dữ liệu
 83                 // Tạo một sản phẩm mới với ID tự sinh
 84                 Product product = new Product();
 85                 product.setProductName(name);
 86                 product.setDesc(desc);
 87                 product.setPriceOld(price);
 88                 product.setPriceNew(priceNewFormatted);
 89
 90                 // Thêm ảnh sản phẩm
 91                 product.setImgURL(imgMainURL);
 92                 product.setImgURLSlider(imgSliderURL);
 93                 product.setPopular(popular);
 94                 product.setProductType(productType);
 95                 product.setImgURLOther(imgOtherURL);
 96
 97                 // Gán id thay thế id++ cũ now trao đổi lưu trữ
 98                 currentProductId = currentProductId + 1;
 99                 currentProductId.setValValue(currentProductId);
100                 // Tạo ID trên già trả mới currentProductId
101                 String id = String.valueOf(currentProductId);
102
103                 FirebaseDatabase database = FirebaseDatabase.getInstance();
104                 DatabaseReference reference = database.getReference("products").child(id);
105                 reference.setValue(product).addOnCompleteListener(new OnCompleteListener<Void>() {
106                     @Override
107                     public void onComplete(@NonNull Task<Void> task) {
108                         if (!task.isSuccessful()) {
109                             Toast.makeText(getApplicationContext(), "Thêm sản phẩm thành công", Toast.LENGTH_SHORT).show();
110                             dialog.dismiss();
111                             Intent intent = new Intent(getApplicationContext(), AddDrinkAdminActivity.this, DrinkPageAdminActivity.class);
112                             startActivity(intent);
113                         }
114                     }
115                 }).addOnFailureListener(new OnFailureListener() {
116                     @Override
117                     public void onFailure(@NonNull Exception e) {
118                         dialog.dismiss();
119                         Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
120                     }
121                 });
122             }
123         } catch (IllegalStateException e) {
124             dialog.dismiss();
125             Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
126         } catch (Exception e) {
127             dialog.dismiss();
128             Toast.makeText(getApplicationContext(), "Đã xảy ra lỗi: " + e.getMessage(), Toast.LENGTH_SHORT).show();
129         }
130     }
131 }

```

Hình 44 Chức năng thêm đồ uống - ADMIN

### 3.4.13 Chức năng thêm món tráng miệng - ADMIN



```

  M: README.md  UpdateFoodAdminActivity.java  UpdateDrinkAdminActivity.java  UpdateDessertAdminActivity.java  AddDessertAdminActivity.java
  -> 45 public class AddDessertAdminActivity extends AppCompatActivity {
  46     protected void onCreate(Bundle savedInstanceState) {
  47         imgOther.setOnTouchListener(new View.OnTouchListener() {
  48             public void onTouch(View v) {
  49                 ...
  50             }
  51         });
  52         // add dots
  53         btn_dot.setOnTouchListener(new View.OnTouchListener() {
  54             @Override
  55             public void onClick(View v) { saveMainImage(); }
  56         });
  57         imgBack.setOnTouchListener(new View.OnTouchListener() {
  58             @Override
  59             public void onClick(View v) { finish(); }
  60         });
  61     }
  62
  63     // Hàm upload ảnh chính lên Firebase
  64     private void saveMainImage() {
  65         if (uriMainImage == null) {
  66             Toast.makeText(context, AddDessertAdminActivity.this, "Vui lòng chọn hình ảnh chính", Toast.LENGTH_SHORT).show();
  67             return;
  68         }
  69
  70         StorageReference storageReference = FirebaseStorage.getInstance().getReference().child( pathString + "Save Images")
  71             .child(uriMainImage.getLastPathSegment());
  72
  73         AlertDialog.Builder builder = new AlertDialog.Builder(context, AddDessertAdminActivity.this);
  74         builder.setCancelable(false);
  75         builder.setView(R.layout.progress_admin_layout);
  76         AlertDialog dialog = builder.create();
  77         dialog.show();
  78
  79         storageReference.putFile(Uri.fromFile(imgMainImage)).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
  80             @Override
  81             public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  82                 TaskUri uriTask = taskSnapshot.getStorage().getDownloadUrl();
  83                 while (!uriTask.isComplete());
  84                 Uri urlImage = uriTask.getResult();
  85                 imgMainURL = urlImage.toString();
  86             }
  87         }).addOnFailureListener(new OnFailureListener() {
  88             @Override
  89             public void onFailure(@NonNull Exception e) {
  90                 dialog.dismiss();
  91                 Toast.makeText(context, AddDessertAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
  92             }
  93         });
  94
  95     }
  96
  97     // Hàm upload ảnh slider lên Firebase
  98     private void saveSliderImage(AlertDialog dialog) {
  99         if (uriSliderImage == null) {
 100             Toast.makeText(context, AddDessertAdminActivity.this, "Vui lòng chọn hình ảnh slider", Toast.LENGTH_SHORT).show();
 101             dialog.dismiss();
 102             return;
 103         }
 104
 105         StorageReference storageReference = FirebaseStorage.getInstance().getReference().child( pathString + "Save Images")
 106             .child(uriSliderImage.getLastPathSegment());
 107
 108         storageReference.putFile(Uri.fromFile(imgSliderImage)).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
 109             @Override
 110             public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
 111                 TaskUri uriTask = taskSnapshot.getStorage().getDownloadUrl();
 112                 while (!uriTask.isComplete());
 113                 Uri urlImage = uriTask.getResult();
 114                 imgSliderURL = urlImage.toString();
 115                 saveOtherImage(dialog);
 116             }
 117         }).addOnFailureListener(new OnFailureListener() {
 118             @Override
 119             public void onFailure(@NonNull Exception e) {
 120                 dialog.dismiss();
 121                 Toast.makeText(context, AddDessertAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
 122             }
 123         });
 124
 125     }
 126
 127     // Hàm upload ảnh khác lên Firebase
 128     private void saveOtherImage(AlertDialog dialog) {
 129         if (uriOtherImage == null) {
 130             Toast.makeText(context, AddDessertAdminActivity.this, "Vui lòng chọn hình ảnh khác", Toast.LENGTH_SHORT).show();
 131             dialog.dismiss();
 132             return;
 133         }
 134
 135         StorageReference storageReference = FirebaseStorage.getInstance().getReference().child( pathString + "Save Images")
 136             .child(uriOtherImage.getLastPathSegment());
 137
 138         storageReference.putFile(Uri.fromFile(imgOtherImage)).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
 139             @Override
 140             public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
 141                 TaskUri uriTask = taskSnapshot.getStorage().getDownloadUrl();
 142                 while (!uriTask.isComplete());
 143                 Uri urlImage = uriTask.getResult();
 144                 imgOtherURL = urlImage.toString();
 145                 uploadData(dialog);
 146             }
 147         }).addOnFailureListener(new OnFailureListener() {
 148             @Override
 149             public void onFailure(@NonNull Exception e) {
 150                 dialog.dismiss();
 151                 Toast.makeText(context, AddDessertAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
 152             }
 153         });
 154
 155     }
 156
 157     // Hàm upload dữ liệu sản phẩm lên Firebase
 158     private void uploadData(AlertDialog dialog) {
 159         try {
 160             String name = edit_name.getText().toString();
 161             String desc = edit_desc.getText().toString();
 162             String price = edit_price.getText().toString();
 163             String sale = edit_sale.getText().toString();
 164             Boolean popular = rad_popular1.isChecked(); // Lưu giá trị từ RadioButton
 165         } catch (Exception e) {
 166             e.printStackTrace();
 167         }
 168     }
 169
 170 }

```

```

45 public class AddDessertAdminActivity extends AppCompatActivity {
46     // Hàm upload dữ liệu sản phẩm lên Firebase
47     ...
48     private void uploadData(AlertDialog dialog) {
49         try {
50             String name = edt_name.getText().toString().trim();
51             String desc = edt_desc.getText().toString().trim();
52             String price = edt_price.getText().toString().trim();
53             String sale = edt_sale.getText().toString().trim();
54             Boolean popular = rd_popular.isChecked(); // Lấy giá trị từ RadioButton
55
56             // Kiểm tra các trường dữ liệu
57             if (name.isEmpty()) {
58                 throw new IllegalArgumentException("Vui lòng nhập tên sản phẩm");
59             } else if (desc.isEmpty()) {
60                 throw new IllegalArgumentException("Vui lòng nhập mô tả sản phẩm");
61             } else if (price.isEmpty()) {
62                 throw new IllegalArgumentException("Vui lòng nhập giá sản phẩm");
63             }
64
65             // Tính priceNew nếu có giảm giá
66             double priceOld = Double.parseDouble(price.replaceAll("\\s+", ""));
67             double priceNew = priceOld; // Giá gốc để sau sẽ là giá cũ
68             if (sale.isEmpty()) {
69                 int salePercent = Integer.parseInt(sale.replaceAll("\\s+", ""));
70                 priceNew = priceOld * (100 - salePercent) / 100;
71             }
72
73             // Format priceNew để lưu vào Firebase
74             DecimalFormat formatter = new DecimalFormat("##,##");
75             String priceNewFormatted = formatter.format(priceNew);
76
77             // Lấy loại sản phẩm đã chọn từ RadioButton
78             int selectedProductType = radioGroup_product_type.getCheckedRadioButtonId();
79             RadioButton selectedProductTypeRadioButton = findViewById(selectedProductType);
80             String productType = selectedProductTypeRadioButton.getText().toString();
81
82             // Tạo một sản phẩm mới với ID tự sinh
83             Product product = new Product();
84             product.setName(name);
85             product.setDesc(desc);
86             product.setPriceOld(price);
87             product.setPriceNew(priceNewFormatted);
88             product.setSale(sale);
89
90             // Set giá trị sản phẩm đã chọn
91             product.setProductType(productType);
92             product.setImgURL(imgThumbnailURL);
93
94             // Sửa id theo stt i++ có node trên firebase để lưu trữ
95             currentProductId++;
96             currentProductIdReference.setValue(currentProductId);
97             // Tạo ID trên giá trị node currentProductId
98             String id = String.valueOf(currentProductId);
99
100            FirebaseDatabase.getInstance().getReference().child("products")
101                .child(id).setValue(product).addOnCompleteListener(new OnCompleteListener<Void>() {
102                    @Override
103                    public void onComplete(@NonNull Task<Void> task) {
104                        if (!task.isSuccessful()) {
105                            Toast.makeText(getApplicationContext(), "Thêm sản phẩm thành công", Toast.LENGTH_SHORT).show();
106                            dialog.dismiss();
107                            Intent intent = new Intent(getApplicationContext(), DessertPageAdminActivity.class);
108                            startActivity(intent);
109                        }
110                    }
111                }).addOnFailureListener(new OnFailureListener() {
112                    @Override
113                    public void onFailure(@NonNull Exception e) {
114                        dialog.dismiss();
115                        Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
116                    }
117                });
118
119            } catch (IllegalArgumentException e) {
120                dialog.dismiss();
121                Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
122            } catch (Exception e) {
123                dialog.dismiss();
124                Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
125            }
126        }
127    }
128
129 }

```

Hình 45 Chức năng thêm món tráng miệng - ADMIN

### 3.4.14 Chức năng cập nhật món ngon - ADMIN



```
 41     public class UpdateFoodAdminActivity extends AppCompatActivity {
 42         protected void onCreate(Bundle savedInstanceState) {
 43             ...
 44             // Nhận sản phẩm từ Intent gửi về
 45             Bundle bundle = getIntent().getExtras();
 46             if (bundle != null) {
 47                 Glide.with(this)
 48                     .load(bundle.getString(key: "imgurl")).into(update_imgUrl);
 49                 Glide.with(this)
 50                     .load(bundle.getString(key: "imgSlider")).into(update_imgSlider);
 51                 Glide.with(this)
 52                     .load(bundle.getString(key: "imgOther")).into(update_imgOther);
 53                 update_edit_name.setText(bundle.getString(key: "name"));
 54                 update_edit_desc.setText(bundle.getString(key: "desc"));
 55                 update_edit_price.setText(bundle.getString(key: "price"));
 56                 update_edit_sale.setText(bundle.getString(key: "sale"));
 57                 // Nhận chuỗi "popular" và thiết lập RadioButton
 58                 String popularStatus = bundle.getString(key: "popular");
 59                 if (popularStatus != null) {
 60                     if (popularStatus.equals("Có")) {
 61                         update_rad_popular1.setChecked(true);
 62                     } else if (popularStatus.equals("Không")) {
 63                         update_rad_popular2.setChecked(true);
 64                     }
 65                 }
 66                 String productType = bundle.getString(key: "productType");
 67                 if (productType != null) {
 68                     switch (productType) {
 69                         case "Hàng rau":
 70                             update_rad_product_type1.setChecked(true);
 71                         break;
 72                         case "Rôt ướp":
 73                             update_rad_product_type_drink.setChecked(true);
 74                         break;
 75                         case "Tráng miệng":
 76                             update_rad_product_type_dessert.setChecked(true);
 77                         break;
 78                         default:
 79                             // Xử lý trường hợp không tìm thấy productType nào phù hợp
 80                             break;
 81                     }
 82                 }
 83             }
 84             key = bundle.getString(key: "Key");
 85         }
 86     }
 87 }
```

```
 41     public class UpdateFoodAdminActivity extends AppCompatActivity {
 42         private void updateImage(Uri imgUri) {
 43             dialog.show();
 44             Toast.makeText(context, UpdateFoodAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
 45         }
 46     }
 47 }
 48
 49     2 usages
 50     private void updateSliderImage() {
 51         if (urlImgSlider == null) {
 52             updatedOtherImage();
 53         } else {
 54             StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("uploads/* Save Images")
 55                 .child(urlImgSlider.getLastPathSegment());
 56
 57             storageReference.putFile(urlImgSlider).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
 58                 @Override
 59                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
 60                     taskSnapshot.getStorage().getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
 61                         @Override
 62                         public void onSuccess(Uri uri) {
 63                             imgSlider = uri.toString();
 64                             updateOtherImage();
 65                         }
 66                     });
 67                 }
 68             }).addOnFailureListener(new OnFailureListener() {
 69                 @Override
 70                 public void onFailure(@NonNull Exception e) {
 71                     Toast.makeText(context, UpdateFoodAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
 72                 }
 73             });
 74         }
 75     }
 76
 77     2 usages
 78     private void updateOtherImage() {
 79         if (urlImgOther == null) {
 80             updateData();
 81         } else {
 82             StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("uploads/* Save Images")
 83                 .child(urlImgOther.getLastPathSegment());
 84
 85             storageReference.putFile(urlImgOther).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
 86                 @Override
 87                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
 88                     taskSnapshot.getStorage().getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
 89                         @Override
 90                         public void onSuccess(Uri uri) {
 91                             imgOther = uri.toString();
 92                             updateData();
 93                         }
 94                     });
 95                 }
 96             }).addOnFailureListener(new OnFailureListener() {
 97                 @Override
 98                 public void onFailure(@NonNull Exception e) {
 99                     Toast.makeText(context, UpdateFoodAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
100                 }
101             });
102         }
103     }
104
105     2 usages
106     private void updateImageData() {
107         name = update_edt_name.getText().toString().trim();
108         desc = update_edt_desc.getText().toString().trim();
109         price = update_edt_price.getText().toString().trim();
110         sale = update_edt_sale.getText().toString().trim();
111         boolean isPopular = update_rad_popular1.isChecked();
112
113         if (urlImgUrl == null) {
114             urlImg = oldImgUrl;
115         } else {
116             if (urlImgSlider == null) {
117                 imgSlider = oldImgSlider;
118             }
119         }
120     }
121
122     2 usages
123     private void updateData() {
124         if (urlImgUrl == null) {
125             urlImg = oldImgUrl;
126         } else {
127             if (urlImgSlider == null) {
128                 imgSlider = oldImgSlider;
129             }
130         }
131     }
132 }
```

AAD\_Project-main > app > src > main > java > com.example.android\_food\_app > ActivityAdmin > UpdateFoodAdminActivity > onCreate

```


    public class UpdateFoodAdminActivity extends AppCompatActivity {
        private void updateData() {
            imgSlider = oldImgSlider;
            if (imgUrlOther != null) {
                imgOther = oldImgOther;
            }
            String productType = "";
            int checkedRadioButtonId = update_radioGroup_product_type.getCheckedRadioButtonId();
            if (checkedRadioButtonId == R.id.update_rad_product_type) {
                productType = "Nhóm sản phẩm";
            } else if (checkedRadioButtonId == R.id.update_rad_product_type_drink) {
                productType = "Đồ uống";
            } else if (checkedRadioButtonId == R.id.update_rad_product_type_dessert) {
                productType = "Tráng miệng";
            }
            // Logi bô tát cả khoảng trắng từ chuỗi giá
            price = price.replaceAll("\\s+", "");
            sale = sale.replaceAll("\\s+", "");
            // Format giá
            DecimalFormat decimalFormat = new DecimalFormat("#,###");
            double priceValue = Double.parseDouble(price.replace(".", ",").replace(",", "."));
            price = decimalFormat.format(priceValue);
            // Tính giá mới sau khuyến mãi
            double saleValue;
            if (!sale.isEmpty()) {
                saleValue = Double.parseDouble(sale.replace(".", ",").replace(",", "."));
            } else {
                saleValue = 0;
            }
            double newValue = priceValue - (priceValue * (saleValue / 100));
            String priceNewFormatted = decimalFormat.format(newValue);
            // Tạo đối tượng Product với thông tin mới
            Product product = new Product(name, desc, price, priceNewFormatted, sale, imgUrl, imgSlider, isPopular, productType, imgOther);
            // Tạo đối tượng Product mới với thông tin cập nhật
            Product product = new Product();
            product.setName(name);
            product.setDesc(desc);
            product.setPriceOld(price);
            product.setPriceNew(priceNewFormatted);
            product.setSale(sale);
            product.setImgSlider(imgSlider);
            product.setPopular(isPopular);
            product.setProductType(productType);
            product.setImgOther(imgOther);
            databaseReference.child(key).setValue(product).addOnSuccessListener(new OnSuccessListener() {
                @Override
                public void onSuccess(Void unused) {
                    startActivity(new Intent(packageContext, UpdateFoodAdminActivity.this, FoodPageAdminActivity.class));
                    Toast.makeText(context, "Cập nhật thành công", Toast.LENGTH_SHORT).show();
                    finish();
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    Toast.makeText(context, UpdateFoodAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
                }
            });
        }
    }


```

Hình 46 Chức năng cập nhật món ngon - ADMIN

### 3.4.15 Chức năng cập nhật đồ uống - ADMIN





File: README.md

```
public class UpdateDrinkAdminActivity extends AppCompatActivity {
    private void updateSliderImage() {
        if (urlImgSlider == null) {
            updateOtherImage();
        } else {
            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("anhHuong").child(urlImgSlider).getlastPathSegment());
            storageReference.putFile(urlImgSlider).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                    taskSnapshot.getStorage().getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                        @Override
                        public void onSuccess(Uri uri) {
                            imgSlider = uri.toString();
                            updateOtherImage();
                        }
                    });
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    Toast.makeText(context, UpdateDrinkAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
                }
            });
        }
    }

    private void updateOtherImage() {
        if (urlImgOther == null) {
            updateData();
        } else {
            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("anhHuong").child(urlImgOther.getlastPathSegment());
            storageReference.putFile(urlImgOther).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                    taskSnapshot.getStorage().getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                        @Override
                        public void onSuccess(Uri uri) {
                            imgOther = uri.toString();
                            updateData();
                        }
                    });
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    Toast.makeText(context, UpdateDrinkAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
                }
            });
        }
    }

    private void updateData() {
        name = update_edt_name.getText().toString().trim();
        desc = update_edt_desc.getText().toString().trim();
        price = update_rad_price.getText().toString().trim();
        sale = update_rad_sale.getText().toString().trim();
        boolean isPopular = update_rad_popular.isChecked();

        if (urlImgInr == null) {
            imgInr = oldImgInr;
        }
        if (urlImgSlider == null) {
            imgSlider = oldImgSlider;
        }
        if (urlImgOther == null) {
            imgOther = oldImgOther;
        }

        String productType = "";
        int checkedRadioButtonId = update_radioGroup_product_type.getCheckedRadioButtonId();

        if (checkedRadioButtonId == R.id.update_rad_product_type1) {
            productType = "Món ngon";
        } else if (checkedRadioButtonId == R.id.update_rad_product_type_drink) {
            productType = "Bđ uống";
        } else if (checkedRadioButtonId == R.id.update_rad_product_type_dessert) {
            productType = "Tráng miệng";
        }

        // Lỗi bỏ tất cả khoảng trắng từ chuỗi giá
        price = price.replaceAll("\\s+", replacement: "");
        sale = sale.replaceAll("\\s+", replacement: "");

        // Format giá
        DecimalFormat decimalFormat = new DecimalFormat(pattern: "#,###");
        double priceValue = Double.parseDouble(price.replace(target: ".", replacement: "").replace(target: ",", replacement: "")));
        price = decimalFormat.format(priceValue);

        // Tìm giá mới sau khi giảm
        double saleValue;
        if (!sale.isEmpty()) {
            saleValue = Double.parseDouble(sale.replace(target: ".", replacement: "").replace(target: ",", replacement: "")));
        } else {
            saleValue = 0;
        }
        double newPriceValue = priceValue - (priceValue * (saleValue / 100));
        String priceNewFormatted = decimalFormat.format(newPriceValue);

        // Tạo đối tượng Product với thông tin mới
        Product product = new Product(name, desc, price, priceNewFormatted, sale, imgUrl, imgSlider, isPopular, productType, imgOther);

        // Tạo đối tượng Product mới với thông tin cũ
        Product product = new Product();
        product.setName(name);
        product.setDesc(desc);
        product.setPriceOld(price);
        product.setPriceNew(priceNewFormatted);
        product.setSale(sale);
        product.setImgSlider(imgSlider);
        product.setImgOther(imgOther);
        product.setProductType(productType);
    }
}
```

The screenshot shows the Android Studio code editor with the file `UpdateDrinkAdminActivity.java` open. The code is part of the `ActivityAdmin` class and handles updating a drink's price. It includes logic to calculate a new price value based on a percentage change, create a new product object with the updated price, and then update the database reference with the new product information. The code uses `OnSuccessListener` and `OnFailureListener` to handle database operations.

```
public class UpdateDrinkAdminActivity extends AppCompatActivity {
    private void updateDrinksData() {
        double newPriceValue = priceValue - (priceValue * (saleValue / 100));
        String priceNewFormatted = decimalFormat.format(newPriceValue);
        // Tao doi tuong Product voi thong tin moi
        Product product = new Product(name, desc, price, priceNewFormatted, sale, imgUrl, imgSlider, isPopular, productType, imgOther);
        // Tao doi tuong Product moi voi thong tin cap nhap
        Product product = new Product();
        product.setName(name);
        product.setDesc(desc);
        product.setPriceOld(price);
        product.setPriceNew(priceNewFormatted);
        product.setSale(sale);
        product.setImgUrl(imgUrl);
        product.setImgURLSlider(imgSlider);
        product.setPopular(isPopular);
        product.setProductType(productType);
        product.setImgURLOther(imgOther);
        databaseReference.child(key).setValue(product).addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void unused) {
                startActivity(new Intent(getApplicationContext(), UpdateDrinkAdminActivity.this, DrinkPageAdminActivity.class));
                Toast.makeText(getApplicationContext(), "Cập nhật thành công", Toast.LENGTH_SHORT).show();
                finish();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Hình 47 Chức năng cập nhật đồ uống – ADMIN

### **3.4.16 Chức năng cập nhật trang miệng - ADMIN**

The screenshot shows the Android Studio code editor with the file `UpdateDessertAdminActivity.java` selected. The code implements an `AppCompatActivity` for updating dessert details. It includes fields for EditTexts, RadioButtons, RadioGroups, and ImageViews for updating name, description, price, and sale. It also handles image URLs and database references. The code uses `EdgeToEdge` for notch support and `ViewCompat` for applying insets.

```
1 package com.example.android.food_app.ActivityAdmin;
2
3 import ...
4
5 public class UpdateDessertAdminActivity extends AppCompatActivity {
6
7     ...
8     private EditText update_edt_name, update_edt_desc, update_edt_price, update_edt_sale;
9     ...
10    private ImageView update_imgUrl, update_imgSlider, update_imgOther;
11    ...
12    private RadioButton update_rad_popular1, update_rad_popular2, update_rad_product_type1, update_rad_product_type_drink, update_rad_product_type_dessert;
13    ...
14    private RadioGroup update_radioGroup_product_type;
15    ...
16    private Button btn_update;
17    ...
18    private ImageButton imgBack;
19    ...
20    private String name, desc, price, sale;
21    ...
22    private String imgUrl, imgSlider, imgOther;
23    ...
24    private String key, imgDessertUrl, imgSlider, imgOther;
25    ...
26    private Uri urlImgUrl, urlImgSlider, urlImgOther;
27    ...
28    DatabaseReference databaseReference;
29    ...
30    StorageReference storageReference;
31
32    @Override
33    protected void onCreate(Bundle savedInstanceState) {
34        super.onCreate(savedInstanceState);
35        EdgeToEdge.enable((ShimmerLinearLayoutManager) this);
36        setContentView(R.layout.activity_update_dessert_admin);
37        ViewCompat.setApplyInsetCallback(findViewById(R.id.main), (v, insets) -> {
38            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
39            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
40            return insets;
41        });
42
43        update_edt_name = findViewById(R.id.update_edt_name);
44        update_edt_desc = findViewById(R.id.update_edt_desc);
45
46    }
47
48    ...
49
50    ...
51
52    ...
53
54    ...
55
56    ...
57
58    ...
59
60    ...
61
62    ...
63
64    ...
65
66    ...
67}
```

```

  41  public class UpdateDessertAdminActivity extends AppCompatActivity {
  42      protected void onCreate(Bundle savedInstanceState) {
  43
  44         // ActivityResultLauncher ảnh chính
  45         ActivityResultLauncher<Intent> activityResultLauncher1 = registerForActivityResult(
  46             new ActivityResultContracts.StartActivityForResult(),
  47             new ActivityResultCallback<ActivityResult>() {
  48                 @Override
  49                 public void onActivityResult(ActivityResult result) {
  50                     if (result.getResultCode() == Activity.RESULT_OK) {
  51                         Intent data = result.getData();
  52                         urlImgUrl = data.getData();
  53                         update_imgUrl.setImageURI(urlImgUrl);
  54                     } else {
  55                         Toast.makeText(context, UpdateDessertAdminActivity.this, "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
  56                     }
  57                 }
  58             });
  59
  60         // ActivityResultLauncher ảnh slider
  61         ActivityResultLauncher<Intent> activityResultLauncher2 = registerForActivityResult(
  62             new ActivityResultContracts.StartActivityForResult(),
  63             new ActivityResultCallback<ActivityResult>() {
  64                 @Override
  65                 public void onActivityResult(ActivityResult result) {
  66                     if (result.getResultCode() == Activity.RESULT_OK) {
  67                         Intent data = result.getData();
  68                         urlImgSlider = data.getData();
  69                         update_imgSlider.setImageURI(urlImgSlider);
  70                     } else {
  71                         Toast.makeText(context, UpdateDessertAdminActivity.this, "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
  72                     }
  73                 }
  74             });
  75
  76         // ActivityResultLauncher ảnh khác
  77         ActivityResultLauncher<Intent> activityResultLauncher3 = registerForActivityResult(
  78             new ActivityResultContracts.StartActivityForResult(),
  79             new ActivityResultCallback<ActivityResult>() {
  80                 @Override
  81                 public void onActivityResult(ActivityResult result) {
  82                     if (result.getResultCode() == Activity.RESULT_OK) {
  83                         Intent data = result.getData();
  84                         urlImgOther = data.getData();
  85                         update_imgOther.setImageURI(urlImgOther);
  86                     } else {
  87                         Toast.makeText(context, UpdateDessertAdminActivity.this, "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
  88                     }
  89                 }
  90             });
  91
  92         // Nhận bản phím từ Detail gửi về
  93         Bundle bundle = getIntent().getExtras();
  94         if (bundle != null) {
  95             Glide.with(activity).load(DessertAdminActivity.this).load(bundle.getString(key: "imgUrl")).into(update_imgUrl);
  96             Glide.with(activity).load(DessertAdminActivity.this).load(bundle.getString(key: "imgSlider")).into(update_imgSlider);
  97             Glide.with(activity).load(DessertAdminActivity.this).load(bundle.getString(key: "imgOther")).into(update_imgOther);
  98             update_edt_name.setText(bundle.getString(key: "name"));
  99             update_edt_price.setText(bundle.getString(key: "price"));
 100             update_edt_sale.setText(bundle.getString(key: "sale"));
 101             // Hiện nút "popular" và thiết lập RadioGroup
 102             String popularStatus = bundle.getString(key: "popular");
 103             if (popularStatus != null) {
 104                 if (popularStatus.equals("0")) {
 105                     update_rad_popular1.setChecked(true);
 106                 } else if (popularStatus.equals("1")) {
 107                     update_rad_popular2.setChecked(true);
 108                 }
 109             }
 110             String productType = bundle.getString(key: "productType");
 111             if (productType != null) {
 112                 switch (productType) {
 113                     case "Mon ngon":
 114                         update_rad_product_type1.setChecked(true);
 115                         break;
 116                     case "Trang nhung":
 117                         update_rad_product_type_drink.setChecked(true);
 118                         break;
 119                     case "Trang mieng":
 120                         update_rad_product_type_dessert.setChecked(true);
 121                         break;
 122                     default:
 123                         // Xử lý trường hợp không tìm thấy productType nào phù hợp
 124                         break;
 125                 }
 126             }
 127
 128             key = bundle.getString(key: "Key");
 129             oldImgUrl = bundle.getString(key: "imgUrl");
 130             oldImgSlider = bundle.getString(key: "imgSlider");
 131             oldImgOther = bundle.getString(key: "imgOther");
 132         }
 133
 134         databaseReference = FirebaseDatabase.getInstance().getReference(path: "products");
 135
 136         update_imgUrl.setOnClickListener(new View.OnClickListener() {
 137             @Override
 138             public void onClick(View v) {
 139                 Intent photoPicker = new Intent(Intent.ACTION_PICK);
 140                 photoPicker.setType("image/*");
 141                 activityResultLauncher1.launch(photoPicker);
 142             }
 143         });
 144
 145         update_imgSlider.setOnClickListener(new View.OnClickListener() {
 146             @Override
 147             public void onClick(View v) {
 148                 Intent photoPicker = new Intent(Intent.ACTION_PICK);
 149                 photoPicker.setType("image/*");
 150                 activityResultLauncher2.launch(photoPicker);
 151             }
 152         });
 153
 154         update_imgOther.setOnClickListener(new View.OnClickListener() {
 155             @Override
 156             public void onClick(View v) {
 157                 Intent photoPicker = new Intent(Intent.ACTION_PICK);
 158                 photoPicker.setType("image/*");
 159                 activityResultLauncher3.launch(photoPicker);
 160             }
 161         });
 162     }
 163
 164     protected void onStart() {
 165         super.onStart();
 166         update_edt_name.requestFocus();
 167     }
 168
 169     protected void onStop() {
 170         super.onStop();
 171         update_edt_name.clearFocus();
 172     }
 173
 174     protected void onDestroy() {
 175         super.onDestroy();
 176         update_edt_name.clearFocus();
 177     }
 178
 179     protected void onPause() {
 180         super.onPause();
 181         update_edt_name.clearFocus();
 182     }
 183
 184     protected void onResume() {
 185         super.onResume();
 186         update_edt_name.requestFocus();
 187     }
 188
 189     protected void onSaveInstanceState(Bundle savedInstanceState) {
 190         super.onSaveInstanceState(savedInstanceState);
 191         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 192         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 193         savedInstanceState.putString(key: "imgOther", urlImgOther);
 194         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 195         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 196         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 197         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 198         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 199     }
 200
 201     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 202         super.onActivityResult(requestCode, resultCode, data);
 203         if (requestCode == 1) {
 204             if (resultCode == RESULT_OK) {
 205                 Uri imgUri = data.getData();
 206                 update_imgUrl.setImageURI(imgUri);
 207             }
 208         }
 209         if (requestCode == 2) {
 210             if (resultCode == RESULT_OK) {
 211                 Uri imgSliderUri = data.getData();
 212                 update_imgSlider.setImageURI(imgSliderUri);
 213             }
 214         }
 215         if (requestCode == 3) {
 216             if (resultCode == RESULT_OK) {
 217                 Uri imgOtherUri = data.getData();
 218                 update_imgOther.setImageURI(imgOtherUri);
 219             }
 220         }
 221     }
 222
 223     protected void onRestart() {
 224         super.onRestart();
 225         update_edt_name.requestFocus();
 226     }
 227
 228     protected void onStart() {
 229         super.onStart();
 230         update_edt_name.requestFocus();
 231     }
 232
 233     protected void onStop() {
 234         super.onStop();
 235         update_edt_name.clearFocus();
 236     }
 237
 238     protected void onDestroy() {
 239         super.onDestroy();
 240         update_edt_name.clearFocus();
 241     }
 242
 243     protected void onPause() {
 244         super.onPause();
 245         update_edt_name.clearFocus();
 246     }
 247
 248     protected void onResume() {
 249         super.onResume();
 250         update_edt_name.requestFocus();
 251     }
 252
 253     protected void onSaveInstanceState(Bundle savedInstanceState) {
 254         super.onSaveInstanceState(savedInstanceState);
 255         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 256         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 257         savedInstanceState.putString(key: "imgOther", urlImgOther);
 258         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 259         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 260         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 261         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 262         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 263     }
 264
 265     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 266         super.onActivityResult(requestCode, resultCode, data);
 267         if (requestCode == 1) {
 268             if (resultCode == RESULT_OK) {
 269                 Uri imgUri = data.getData();
 270                 update_imgUrl.setImageURI(imgUri);
 271             }
 272         }
 273         if (requestCode == 2) {
 274             if (resultCode == RESULT_OK) {
 275                 Uri imgSliderUri = data.getData();
 276                 update_imgSlider.setImageURI(imgSliderUri);
 277             }
 278         }
 279         if (requestCode == 3) {
 280             if (resultCode == RESULT_OK) {
 281                 Uri imgOtherUri = data.getData();
 282                 update_imgOther.setImageURI(imgOtherUri);
 283             }
 284         }
 285     }
 286
 287     protected void onRestart() {
 288         super.onRestart();
 289         update_edt_name.requestFocus();
 290     }
 291
 292     protected void onStart() {
 293         super.onStart();
 294         update_edt_name.requestFocus();
 295     }
 296
 297     protected void onStop() {
 298         super.onStop();
 299         update_edt_name.clearFocus();
 300     }
 301
 302     protected void onDestroy() {
 303         super.onDestroy();
 304         update_edt_name.clearFocus();
 305     }
 306
 307     protected void onPause() {
 308         super.onPause();
 309         update_edt_name.clearFocus();
 310     }
 311
 312     protected void onResume() {
 313         super.onResume();
 314         update_edt_name.requestFocus();
 315     }
 316
 317     protected void onSaveInstanceState(Bundle savedInstanceState) {
 318         super.onSaveInstanceState(savedInstanceState);
 319         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 320         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 321         savedInstanceState.putString(key: "imgOther", urlImgOther);
 322         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 323         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 324         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 325         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 326         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 327     }
 328
 329     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 330         super.onActivityResult(requestCode, resultCode, data);
 331         if (requestCode == 1) {
 332             if (resultCode == RESULT_OK) {
 333                 Uri imgUri = data.getData();
 334                 update_imgUrl.setImageURI(imgUri);
 335             }
 336         }
 337         if (requestCode == 2) {
 338             if (resultCode == RESULT_OK) {
 339                 Uri imgSliderUri = data.getData();
 340                 update_imgSlider.setImageURI(imgSliderUri);
 341             }
 342         }
 343         if (requestCode == 3) {
 344             if (resultCode == RESULT_OK) {
 345                 Uri imgOtherUri = data.getData();
 346                 update_imgOther.setImageURI(imgOtherUri);
 347             }
 348         }
 349     }
 350
 351     protected void onRestart() {
 352         super.onRestart();
 353         update_edt_name.requestFocus();
 354     }
 355
 356     protected void onStart() {
 357         super.onStart();
 358         update_edt_name.requestFocus();
 359     }
 360
 361     protected void onStop() {
 362         super.onStop();
 363         update_edt_name.clearFocus();
 364     }
 365
 366     protected void onDestroy() {
 367         super.onDestroy();
 368         update_edt_name.clearFocus();
 369     }
 370
 371     protected void onPause() {
 372         super.onPause();
 373         update_edt_name.clearFocus();
 374     }
 375
 376     protected void onResume() {
 377         super.onResume();
 378         update_edt_name.requestFocus();
 379     }
 380
 381     protected void onSaveInstanceState(Bundle savedInstanceState) {
 382         super.onSaveInstanceState(savedInstanceState);
 383         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 384         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 385         savedInstanceState.putString(key: "imgOther", urlImgOther);
 386         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 387         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 388         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 389         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 390         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 391     }
 392
 393     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 394         super.onActivityResult(requestCode, resultCode, data);
 395         if (requestCode == 1) {
 396             if (resultCode == RESULT_OK) {
 397                 Uri imgUri = data.getData();
 398                 update_imgUrl.setImageURI(imgUri);
 399             }
 400         }
 401         if (requestCode == 2) {
 402             if (resultCode == RESULT_OK) {
 403                 Uri imgSliderUri = data.getData();
 404                 update_imgSlider.setImageURI(imgSliderUri);
 405             }
 406         }
 407         if (requestCode == 3) {
 408             if (resultCode == RESULT_OK) {
 409                 Uri imgOtherUri = data.getData();
 410                 update_imgOther.setImageURI(imgOtherUri);
 411             }
 412         }
 413     }
 414
 415     protected void onRestart() {
 416         super.onRestart();
 417         update_edt_name.requestFocus();
 418     }
 419
 420     protected void onStart() {
 421         super.onStart();
 422         update_edt_name.requestFocus();
 423     }
 424
 425     protected void onStop() {
 426         super.onStop();
 427         update_edt_name.clearFocus();
 428     }
 429
 430     protected void onDestroy() {
 431         super.onDestroy();
 432         update_edt_name.clearFocus();
 433     }
 434
 435     protected void onPause() {
 436         super.onPause();
 437         update_edt_name.clearFocus();
 438     }
 439
 440     protected void onResume() {
 441         super.onResume();
 442         update_edt_name.requestFocus();
 443     }
 444
 445     protected void onSaveInstanceState(Bundle savedInstanceState) {
 446         super.onSaveInstanceState(savedInstanceState);
 447         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 448         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 449         savedInstanceState.putString(key: "imgOther", urlImgOther);
 450         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 451         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 452         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 453         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 454         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 455     }
 456
 457     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 458         super.onActivityResult(requestCode, resultCode, data);
 459         if (requestCode == 1) {
 460             if (resultCode == RESULT_OK) {
 461                 Uri imgUri = data.getData();
 462                 update_imgUrl.setImageURI(imgUri);
 463             }
 464         }
 465         if (requestCode == 2) {
 466             if (resultCode == RESULT_OK) {
 467                 Uri imgSliderUri = data.getData();
 468                 update_imgSlider.setImageURI(imgSliderUri);
 469             }
 470         }
 471         if (requestCode == 3) {
 472             if (resultCode == RESULT_OK) {
 473                 Uri imgOtherUri = data.getData();
 474                 update_imgOther.setImageURI(imgOtherUri);
 475             }
 476         }
 477     }
 478
 479     protected void onRestart() {
 480         super.onRestart();
 481         update_edt_name.requestFocus();
 482     }
 483
 484     protected void onStart() {
 485         super.onStart();
 486         update_edt_name.requestFocus();
 487     }
 488
 489     protected void onStop() {
 490         super.onStop();
 491         update_edt_name.clearFocus();
 492     }
 493
 494     protected void onDestroy() {
 495         super.onDestroy();
 496         update_edt_name.clearFocus();
 497     }
 498
 499     protected void onPause() {
 500         super.onPause();
 501         update_edt_name.clearFocus();
 502     }
 503
 504     protected void onResume() {
 505         super.onResume();
 506         update_edt_name.requestFocus();
 507     }
 508
 509     protected void onSaveInstanceState(Bundle savedInstanceState) {
 510         super.onSaveInstanceState(savedInstanceState);
 511         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 512         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 513         savedInstanceState.putString(key: "imgOther", urlImgOther);
 514         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 515         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 516         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 517         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 518         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 519     }
 520
 521     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 522         super.onActivityResult(requestCode, resultCode, data);
 523         if (requestCode == 1) {
 524             if (resultCode == RESULT_OK) {
 525                 Uri imgUri = data.getData();
 526                 update_imgUrl.setImageURI(imgUri);
 527             }
 528         }
 529         if (requestCode == 2) {
 530             if (resultCode == RESULT_OK) {
 531                 Uri imgSliderUri = data.getData();
 532                 update_imgSlider.setImageURI(imgSliderUri);
 533             }
 534         }
 535         if (requestCode == 3) {
 536             if (resultCode == RESULT_OK) {
 537                 Uri imgOtherUri = data.getData();
 538                 update_imgOther.setImageURI(imgOtherUri);
 539             }
 540         }
 541     }
 542
 543     protected void onRestart() {
 544         super.onRestart();
 545         update_edt_name.requestFocus();
 546     }
 547
 548     protected void onStart() {
 549         super.onStart();
 550         update_edt_name.requestFocus();
 551     }
 552
 553     protected void onStop() {
 554         super.onStop();
 555         update_edt_name.clearFocus();
 556     }
 557
 558     protected void onDestroy() {
 559         super.onDestroy();
 560         update_edt_name.clearFocus();
 561     }
 562
 563     protected void onPause() {
 564         super.onPause();
 565         update_edt_name.clearFocus();
 566     }
 567
 568     protected void onResume() {
 569         super.onResume();
 570         update_edt_name.requestFocus();
 571     }
 572
 573     protected void onSaveInstanceState(Bundle savedInstanceState) {
 574         super.onSaveInstanceState(savedInstanceState);
 575         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 576         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 577         savedInstanceState.putString(key: "imgOther", urlImgOther);
 578         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 579         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 580         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 581         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 582         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 583     }
 584
 585     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 586         super.onActivityResult(requestCode, resultCode, data);
 587         if (requestCode == 1) {
 588             if (resultCode == RESULT_OK) {
 589                 Uri imgUri = data.getData();
 590                 update_imgUrl.setImageURI(imgUri);
 591             }
 592         }
 593         if (requestCode == 2) {
 594             if (resultCode == RESULT_OK) {
 595                 Uri imgSliderUri = data.getData();
 596                 update_imgSlider.setImageURI(imgSliderUri);
 597             }
 598         }
 599         if (requestCode == 3) {
 600             if (resultCode == RESULT_OK) {
 601                 Uri imgOtherUri = data.getData();
 602                 update_imgOther.setImageURI(imgOtherUri);
 603             }
 604         }
 605     }
 606
 607     protected void onRestart() {
 608         super.onRestart();
 609         update_edt_name.requestFocus();
 610     }
 611
 612     protected void onStart() {
 613         super.onStart();
 614         update_edt_name.requestFocus();
 615     }
 616
 617     protected void onStop() {
 618         super.onStop();
 619         update_edt_name.clearFocus();
 620     }
 621
 622     protected void onDestroy() {
 623         super.onDestroy();
 624         update_edt_name.clearFocus();
 625     }
 626
 627     protected void onPause() {
 628         super.onPause();
 629         update_edt_name.clearFocus();
 630     }
 631
 632     protected void onResume() {
 633         super.onResume();
 634         update_edt_name.requestFocus();
 635     }
 636
 637     protected void onSaveInstanceState(Bundle savedInstanceState) {
 638         super.onSaveInstanceState(savedInstanceState);
 639         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 640         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 641         savedInstanceState.putString(key: "imgOther", urlImgOther);
 642         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 643         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 644         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 645         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 646         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 647     }
 648
 649     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 650         super.onActivityResult(requestCode, resultCode, data);
 651         if (requestCode == 1) {
 652             if (resultCode == RESULT_OK) {
 653                 Uri imgUri = data.getData();
 654                 update_imgUrl.setImageURI(imgUri);
 655             }
 656         }
 657         if (requestCode == 2) {
 658             if (resultCode == RESULT_OK) {
 659                 Uri imgSliderUri = data.getData();
 660                 update_imgSlider.setImageURI(imgSliderUri);
 661             }
 662         }
 663         if (requestCode == 3) {
 664             if (resultCode == RESULT_OK) {
 665                 Uri imgOtherUri = data.getData();
 666                 update_imgOther.setImageURI(imgOtherUri);
 667             }
 668         }
 669     }
 670
 671     protected void onRestart() {
 672         super.onRestart();
 673         update_edt_name.requestFocus();
 674     }
 675
 676     protected void onStart() {
 677         super.onStart();
 678         update_edt_name.requestFocus();
 679     }
 680
 681     protected void onStop() {
 682         super.onStop();
 683         update_edt_name.clearFocus();
 684     }
 685
 686     protected void onDestroy() {
 687         super.onDestroy();
 688         update_edt_name.clearFocus();
 689     }
 690
 691     protected void onPause() {
 692         super.onPause();
 693         update_edt_name.clearFocus();
 694     }
 695
 696     protected void onResume() {
 697         super.onResume();
 698         update_edt_name.requestFocus();
 699     }
 700
 701     protected void onSaveInstanceState(Bundle savedInstanceState) {
 702         super.onSaveInstanceState(savedInstanceState);
 703         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 704         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 705         savedInstanceState.putString(key: "imgOther", urlImgOther);
 706         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 707         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 708         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 709         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 710         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 711     }
 712
 713     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 714         super.onActivityResult(requestCode, resultCode, data);
 715         if (requestCode == 1) {
 716             if (resultCode == RESULT_OK) {
 717                 Uri imgUri = data.getData();
 718                 update_imgUrl.setImageURI(imgUri);
 719             }
 720         }
 721         if (requestCode == 2) {
 722             if (resultCode == RESULT_OK) {
 723                 Uri imgSliderUri = data.getData();
 724                 update_imgSlider.setImageURI(imgSliderUri);
 725             }
 726         }
 727         if (requestCode == 3) {
 728             if (resultCode == RESULT_OK) {
 729                 Uri imgOtherUri = data.getData();
 730                 update_imgOther.setImageURI(imgOtherUri);
 731             }
 732         }
 733     }
 734
 735     protected void onRestart() {
 736         super.onRestart();
 737         update_edt_name.requestFocus();
 738     }
 739
 740     protected void onStart() {
 741         super.onStart();
 742         update_edt_name.requestFocus();
 743     }
 744
 745     protected void onStop() {
 746         super.onStop();
 747         update_edt_name.clearFocus();
 748     }
 749
 750     protected void onDestroy() {
 751         super.onDestroy();
 752         update_edt_name.clearFocus();
 753     }
 754
 755     protected void onPause() {
 756         super.onPause();
 757         update_edt_name.clearFocus();
 758     }
 759
 760     protected void onResume() {
 761         super.onResume();
 762         update_edt_name.requestFocus();
 763     }
 764
 765     protected void onSaveInstanceState(Bundle savedInstanceState) {
 766         super.onSaveInstanceState(savedInstanceState);
 767         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 768         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 769         savedInstanceState.putString(key: "imgOther", urlImgOther);
 770         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 771         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 772         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 773         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 774         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 775     }
 776
 777     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 778         super.onActivityResult(requestCode, resultCode, data);
 779         if (requestCode == 1) {
 780             if (resultCode == RESULT_OK) {
 781                 Uri imgUri = data.getData();
 782                 update_imgUrl.setImageURI(imgUri);
 783             }
 784         }
 785         if (requestCode == 2) {
 786             if (resultCode == RESULT_OK) {
 787                 Uri imgSliderUri = data.getData();
 788                 update_imgSlider.setImageURI(imgSliderUri);
 789             }
 790         }
 791         if (requestCode == 3) {
 792             if (resultCode == RESULT_OK) {
 793                 Uri imgOtherUri = data.getData();
 794                 update_imgOther.setImageURI(imgOtherUri);
 795             }
 796         }
 797     }
 798
 799     protected void onRestart() {
 800         super.onRestart();
 801         update_edt_name.requestFocus();
 802     }
 803
 804     protected void onStart() {
 805         super.onStart();
 806         update_edt_name.requestFocus();
 807     }
 808
 809     protected void onStop() {
 810         super.onStop();
 811         update_edt_name.clearFocus();
 812     }
 813
 814     protected void onDestroy() {
 815         super.onDestroy();
 816         update_edt_name.clearFocus();
 817     }
 818
 819     protected void onPause() {
 820         super.onPause();
 821         update_edt_name.clearFocus();
 822     }
 823
 824     protected void onResume() {
 825         super.onResume();
 826         update_edt_name.requestFocus();
 827     }
 828
 829     protected void onSaveInstanceState(Bundle savedInstanceState) {
 830         super.onSaveInstanceState(savedInstanceState);
 831         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 832         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 833         savedInstanceState.putString(key: "imgOther", urlImgOther);
 834         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 835         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 836         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 837         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 838         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 839     }
 840
 841     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 842         super.onActivityResult(requestCode, resultCode, data);
 843         if (requestCode == 1) {
 844             if (resultCode == RESULT_OK) {
 845                 Uri imgUri = data.getData();
 846                 update_imgUrl.setImageURI(imgUri);
 847             }
 848         }
 849         if (requestCode == 2) {
 850             if (resultCode == RESULT_OK) {
 851                 Uri imgSliderUri = data.getData();
 852                 update_imgSlider.setImageURI(imgSliderUri);
 853             }
 854         }
 855         if (requestCode == 3) {
 856             if (resultCode == RESULT_OK) {
 857                 Uri imgOtherUri = data.getData();
 858                 update_imgOther.setImageURI(imgOtherUri);
 859             }
 860         }
 861     }
 862
 863     protected void onRestart() {
 864         super.onRestart();
 865         update_edt_name.requestFocus();
 866     }
 867
 868     protected void onStart() {
 869         super.onStart();
 870         update_edt_name.requestFocus();
 871     }
 872
 873     protected void onStop() {
 874         super.onStop();
 875         update_edt_name.clearFocus();
 876     }
 877
 878     protected void onDestroy() {
 879         super.onDestroy();
 880         update_edt_name.clearFocus();
 881     }
 882
 883     protected void onPause() {
 884         super.onPause();
 885         update_edt_name.clearFocus();
 886     }
 887
 888     protected void onResume() {
 889         super.onResume();
 890         update_edt_name.requestFocus();
 891     }
 892
 893     protected void onSaveInstanceState(Bundle savedInstanceState) {
 894         super.onSaveInstanceState(savedInstanceState);
 895         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 896         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 897         savedInstanceState.putString(key: "imgOther", urlImgOther);
 898         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 899         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 900         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 901         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 902         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 903     }
 904
 905     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 906         super.onActivityResult(requestCode, resultCode, data);
 907         if (requestCode == 1) {
 908             if (resultCode == RESULT_OK) {
 909                 Uri imgUri = data.getData();
 910                 update_imgUrl.setImageURI(imgUri);
 911             }
 912         }
 913         if (requestCode == 2) {
 914             if (resultCode == RESULT_OK) {
 915                 Uri imgSliderUri = data.getData();
 916                 update_imgSlider.setImageURI(imgSliderUri);
 917             }
 918         }
 919         if (requestCode == 3) {
 920             if (resultCode == RESULT_OK) {
 921                 Uri imgOtherUri = data.getData();
 922                 update_imgOther.setImageURI(imgOtherUri);
 923             }
 924         }
 925     }
 926
 927     protected void onRestart() {
 928         super.onRestart();
 929         update_edt_name.requestFocus();
 930     }
 931
 932     protected void onStart() {
 933         super.onStart();
 934         update_edt_name.requestFocus();
 935     }
 936
 937     protected void onStop() {
 938         super.onStop();
 939         update_edt_name.clearFocus();
 940     }
 941
 942     protected void onDestroy() {
 943         super.onDestroy();
 944         update_edt_name.clearFocus();
 945     }
 946
 947     protected void onPause() {
 948         super.onPause();
 949         update_edt_name.clearFocus();
 950     }
 951
 952     protected void onResume() {
 953         super.onResume();
 954         update_edt_name.requestFocus();
 955     }
 956
 957     protected void onSaveInstanceState(Bundle savedInstanceState) {
 958         super.onSaveInstanceState(savedInstanceState);
 959         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 960         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 961         savedInstanceState.putString(key: "imgOther", urlImgOther);
 962         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 963         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 964         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 965         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 966         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 967     }
 968
 969     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 970         super.onActivityResult(requestCode, resultCode, data);
 971         if (requestCode == 1) {
 972             if (resultCode == RESULT_OK) {
 973                 Uri imgUri = data.getData();
 974                 update_imgUrl.setImageURI(imgUri);
 975             }
 976         }
 977         if (requestCode == 2) {
 978             if (resultCode == RESULT_OK) {
 979                 Uri imgSliderUri = data.getData();
 980                 update_imgSlider.setImageURI(imgSliderUri);
 981             }
 982         }
 983         if (requestCode == 3) {
 984             if (resultCode == RESULT_OK) {
 985                 Uri imgOtherUri = data.getData();
 986                 update_imgOther.setImageURI(imgOtherUri);
 987             }
 988         }
 989     }
 990
 991     protected void onRestart() {
 992         super.onRestart();
 993         update_edt_name.requestFocus();
 994     }
 995
 996     protected void onStart() {
 997         super.onStart();
 998         update_edt_name.requestFocus();
 999     }
 1000
 1001     protected void onStop() {
 1002         super.onStop();
 1003         update_edt_name.clearFocus();
 1004     }
 1005
 1006     protected void onDestroy() {
 1007         super.onDestroy();
 1008         update_edt_name.clearFocus();
 1009     }
 1010
 1011     protected void onPause() {
 1012         super.onPause();
 1013         update_edt_name.clearFocus();
 1014     }
 1015
 1016     protected void onResume() {
 1017         super.onResume();
 1018         update_edt_name.requestFocus();
 1019     }
 1020
 1021     protected void onSaveInstanceState(Bundle savedInstanceState) {
 1022         super.onSaveInstanceState(savedInstanceState);
 1023         savedInstanceState.putString(key: "imgUrl", urlImgUrl);
 1024         savedInstanceState.putString(key: "imgSlider", urlImgSlider);
 1025         savedInstanceState.putString(key: "imgOther", urlImgOther);
 1026         savedInstanceState.putString(key: "name", update_edt_name.getText().toString());
 1027         savedInstanceState.putString(key: "price", update_edt_price.getText().toString());
 1028         savedInstanceState.putString(key: "sale", update_edt_sale.getText().toString());
 1029         savedInstanceState.putString(key: "popular", update_rad_popular1.isChecked() ? "0" : "1");
 1030         savedInstanceState.putString(key: "productType", update_rad_product_type1.isChecked() ? "Mon ngon" : "Trang nhung");
 1031     }
 1032
 1033     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
 1034         super.onActivityResult(requestCode, resultCode, data);
 1035         if (requestCode == 1) {
 1036             if (resultCode == RESULT_OK) {
 1037                 Uri imgUri = data.getData();
 1038                 update_imgUrl.setImageURI(imgUri);
 1039             }
 1040         }
 1041         if (requestCode == 2) {
 1042             if (resultCode == RESULT_OK) {
 1043                 Uri imgSliderUri = data.getData();
 1044                 update_imgSlider.setImageURI(imgSliderUri);
 1045             }
 1046         }
 1047         if (requestCode == 3) {
 1048             if (resultCode == RESULT_OK) {
 1049                 Uri imgOtherUri = data.getData();
 1050                 update_imgOther.setImageURI(imgOtherUri);
 1051             }
 1052         }
 1053     }
 1054
 1055     protected void onRestart() {
 1056         super.on
```

```

 41  public class UpdateDessertAdminActivity extends AppCompatActivity {
 42      protected void onCreate(Bundle savedInstanceState) {
 43          updateImgOther.setOnClickListener(new View.OnClickListener() {
 44              @Override
 45              public void onClick(View v) {
 46                  Intent photoPicker = new Intent(Intent.ACTION_PICK);
 47                  photoPicker.setType("image/*");
 48                  activityResultLauncher.launch(photoPicker);
 49              }
 50          });
 51
 52          btn_update.setOnClickListener(new View.OnClickListener() {
 53              @Override
 54              public void onClick(View v) {
 55                  updateMainImage();
 56                  Intent intent = new Intent(getApplicationContext(), DessertPageAdminActivity.class);
 57                  startActivity(intent);
 58              }
 59          });
 60
 61      }
 62
 63      private void updateMainImage() {
 64          if (uriImgUrl == null) {
 65              updateSliderImage();
 66          } else {
 67              StorageReference storageReference = FirebaseStorage.getInstance().getReference().child(uriImgUrl.getLastPathSegment());
 68
 69              AlertDialog.Builder builder = new AlertDialog.Builder(UpdateDessertAdminActivity.this);
 70              builder.setCancelable(true);
 71              builder.setView(R.layout.progress_admin_layout);
 72              AlertDialog dialog = builder.create();
 73              dialog.show();
 74
 75              storageReference.putFile(uriImgUrl).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
 76                  @Override
 77                  public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
 78                      taskSnapshot.getStorage().getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
 79                          @Override
 80                          public void onSuccess(Uri uri) {
 81                              imgUrl = uri.toString();
 82                          }
 83                      });
 84                  }
 85              }).addOnFailureListener(new OnFailureListener() {
 86                  @Override
 87                  public void onFailure(@NonNull Exception e) {
 88                      dialog.dismiss();
 89                      Toast.makeText(context, UpdateDessertAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
 90                  }
 91              });
 92
 93      }
 94
 95      private void updateSliderImage() {
 96          if (uriImgSlider == null) {
 97              updateOtherImage();
 98          } else {
 99              StorageReference storageReference = FirebaseStorage.getInstance().getReference().child(uriImgSlider.getLastPathSegment());
100
101              storageReference.putFile(uriImgSlider).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
102                  @Override
103                  public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
104                      taskSnapshot.getStorage().getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
105                          @Override
106                          public void onSuccess(Uri uri) {
107                              imgSlider = uri.toString();
108                          }
109                      });
110                  }
111              }).addOnFailureListener(new OnFailureListener() {
112                  @Override
113                  public void onFailure(@NonNull Exception e) {
114                      Toast.makeText(context, UpdateDessertAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
115                  }
116              });
117
118      }
119
120      private void updateOtherImage() {
121          if (uriImgOther == null) {
122              updateData();
123          } else {
124              StorageReference storageReference = FirebaseStorage.getInstance().getReference().child(uriImgOther.getLastPathSegment());
125
126              storageReference.putFile(uriImgOther).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
127                  @Override
128                  public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
129                      taskSnapshot.getStorage().getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
130                          @Override
131                          public void onSuccess(Uri uri) {
132                              imgOther = uri.toString();
133                          }
134                      });
135                  }
136              }).addOnFailureListener(new OnFailureListener() {
137                  @Override
138                  public void onFailure(@NonNull Exception e) {
139                      Toast.makeText(context, UpdateDessertAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
140                  }
141              });
142
143      }
144
145      private void updateData() {
146          ...
147      }
148
149  }

```

Phát triển ứng dụng cho thiết bị di động

```

public class UpdateDessertAdminActivity extends AppCompatActivity {
    ...
    private void updateData() {
        name = update_edt_name.getText().toString().trim();
        desc = update_edt_desc.getText().toString().trim();
        price = update_edt_price.getText().toString().trim();
        sale = update_edt_sale.getText().toString().trim();
        boolean isPopular = update_rad_popular.isChecked();
        ...
        if (urlImgUrl == null) {
            imgURL = oldImgURL;
        }
        if (urlImgSlider == null) {
            imgSlider = oldImgSlider;
        }
        if (urlImgOther == null) {
            imgOther = oldImgOther;
        }
        ...
        String productType = "";
        int checkedRadioButtonId = update_radioGroup_product_type.getCheckedRadioButtonId();
        ...
        if (checkedRadioButtonId == R.id.update_rad_product_type1) {
            productType = "Món ăn";
        } else if (checkedRadioButtonId == R.id.update_rad_product_type_drink) {
            productType = "Đồ uống";
        } else if (checkedRadioButtonId == R.id.update_rad_product_type_dessert) {
            productType = "Tráng miệng";
        }
        ...
        // Logi bđ tắt cd không trắng tđ chđi già
        price = price.replaceAll("\\s+", replacement);
        sale = sale.replaceAll("\\s+", replacement);
        ...
        // Format giá
        DecimalFormat decimalFormat = new DecimalFormat("###,###");
        double priceValue = Double.parseDouble(price.replace(".", ",").replace(",", "."));
        price = decimalFormat.format(priceValue);
        ...
        // Tính giá mới sau khi giảm
        double saleValue;
        if (sale.isEmpty()) {
            ...
        }
        ...
        // Tạo đối tượng Product với thông tin mới
        Product product = new Product(name, desc, price, priceNewFormatted, sale, imgURL, imgSlider, isPopular, productType, imgOther);
        ...
        // Tạo đối tượng Product mới với thông tin cập nhật
        Product product = new Product();
        product.setName(name);
        ...
        product.setPriceId(prince);
        product.setPriceNew(princeNewFormatted);
        product.setSale(sale);
        product.setImgURL(imgURL);
        product.setImgURLSlider(imgSlider);
        product.setImgURLOther(imgOther);
        ...
        databaseReference.child(key).setValue(product).addOnSuccessListener(new OnSuccessListener<Void>() {
            ...
            @Override
            public void onSuccess(Void unused) {
                startActivity(new Intent(getApplicationContext(), UpdateDessertAdminActivity.this, DessertPageAdminActivity.class));
                Toast.makeText(getApplicationContext(), "Cập nhật thành công", Toast.LENGTH_SHORT).show();
                finish();
            }
        }).addOnFailureListener(new OnFailureListener() {
            ...
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

Hình 48 Chức năng cập nhật trang miêng - ADMIN

## KẾT LUẬN

### 1. Kết quả đạt được

Nhóm đã cơ bản hoàn thiện dự án, với các chức năng chính cho phép khách hàng dễ dàng xem danh sách món ăn và tiến hành đặt món. Bên cạnh đó, chủ nhà hàng có thể thực hiện các thao tác thêm, chỉnh sửa hoặc xóa món ăn, đồ uống và món tráng miệng một cách linh hoạt. Đồng thời, hệ thống cũng hỗ trợ người quản lý theo dõi và quản lý doanh thu hiệu quả.

### 2. Nhược điểm

Mặc dù ứng dụng đã đáp ứng được các chức năng cơ bản như đặt món, quản lý thực đơn và theo dõi doanh thu, nhưng hiện tại vẫn còn một số hạn chế cần được khắc phục trong các phiên bản tiếp theo:

- Chưa tích hợp hệ thống giao hàng: Ứng dụng chưa có chức năng kết nối với đội ngũ giao hàng hoặc đơn vị vận chuyển, do đó việc giao nhận vẫn phải thực hiện thủ công bên ngoài ứng dụng, gây bất tiện cho cả người bán lẫn khách hàng.
- Thiếu đa dạng hình thức thanh toán: Hiện tại, ứng dụng chưa hỗ trợ các phương thức thanh toán trực tuyến như ví điện tử, chuyển khoản hay thanh toán bằng mã QR. Điều này có thể ảnh hưởng đến trải nghiệm người dùng và hạn chế khả năng mở rộng quy mô dịch vụ.

### 3. Hướng phát triển

- Tích hợp hệ thống giao hàng: Kết nối đối tác vận chuyển (Giao hàng nhanh, Ahamove, GrabExpress).
- Bổ sung các phương thức thanh toán hiện đại:
  - Hỗ trợ thanh toán bằng ví điện tử (Momo, ZaloPay), thẻ ngân hàng và QR Code.
  - Cho phép lưu phương thức thanh toán để thanh toán nhanh ở lần sau.
- Hệ thống khách hàng thân thiết: Tích điểm sau mỗi lần đặt hàng để đổi ưu đãi.
- Ứng dụng AI và Machine Learning: Gợi ý món ăn dựa trên lịch sử đặt món và thói quen ăn uống. Tự động sắp xếp thực đơn theo xu hướng tiêu dùng. Chatbot hỗ trợ khách hàng 24/7.
- Thêm trạng thái món ăn đã hết hàng hay còn hàng.

## TÀI LIỆU THAM KHẢO

- [1]<https://www.youtube.com/playlist?list=PL3Ob3F0T-08Y72KEKPlmdsKhLMcWhJLcf>
- [2][https://www.youtube.com/playlist?list=PL3Ob3F0T-08Zb-3wiXNwetWTICtMU2\\_ns](https://www.youtube.com/playlist?list=PL3Ob3F0T-08Zb-3wiXNwetWTICtMU2_ns)
- [3]<https://www.youtube.com/playlist?list=PL3Ob3F0T-08awPo1n263zt76wB5MfLsUq>
- [4][https://www.youtube.com/playlist?list=PLQ\\_Ai1O7sMV0vXXEfFUe05mHCMqDYWDrz](https://www.youtube.com/playlist?list=PLQ_Ai1O7sMV0vXXEfFUe05mHCMqDYWDrz)
- [5] [https://www.youtube.com/watch?v=p\\_LjlsxcKxw&t=193s](https://www.youtube.com/watch?v=p_LjlsxcKxw&t=193s)