

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



**BÀI TẬP LỚN**

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

**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

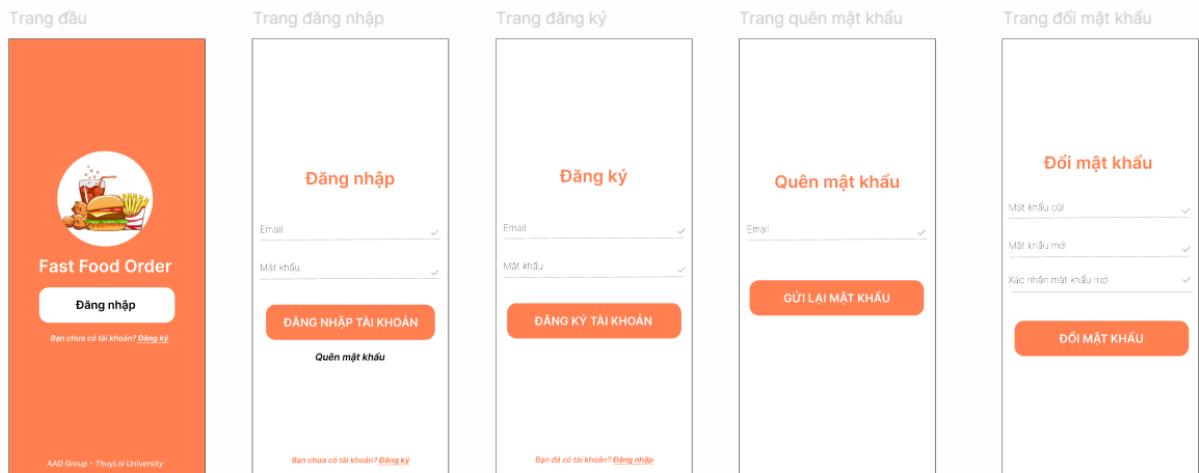
## Chương 3. XÂY DỰNG ỨNG DỤNG

### 3.1. Thiết kế Figma

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

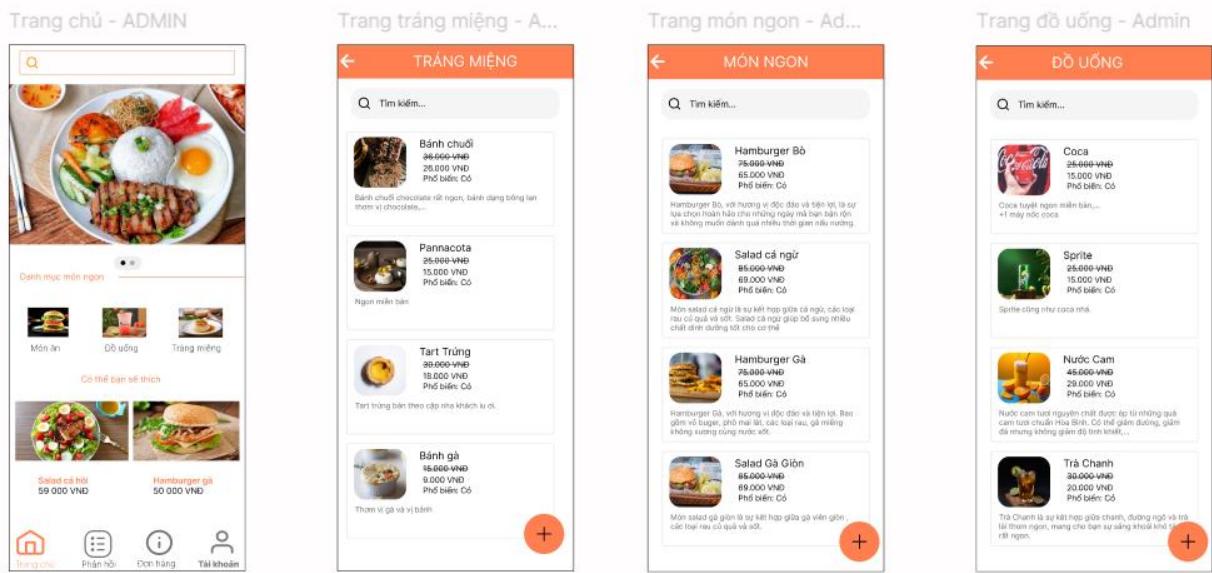
Ảnh kết xuất Figma:

- 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 screenshot displays the Admin Dashboard with several management pages:

- Trang đơn hàng...**: Order Details page showing an order from user ID 2251062321.
- Trang thêm món...**: Add Item page with fields for Name, Description, Price, and Discount.
- Trang thêm đồ uống...**: Add Drink page with similar fields.
- Trang thêm tráng miệng...**: Add Dessert page with similar fields.
- Trang Chi Tiết ...**: Item Detail page for a Hamburger.
- Trang sửa món...**: Edit Item page for the same Hamburger.
- Trang tài khoản...**: Account Management page for user admin123@gmail.com.
- Trang phản hồi...**: Feedback page listing user reviews.
- Trang doanh thu...**: Sales Report page showing sales details for different users.

Hình 3: Giao diện chức năng chi tiết ADMIN

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

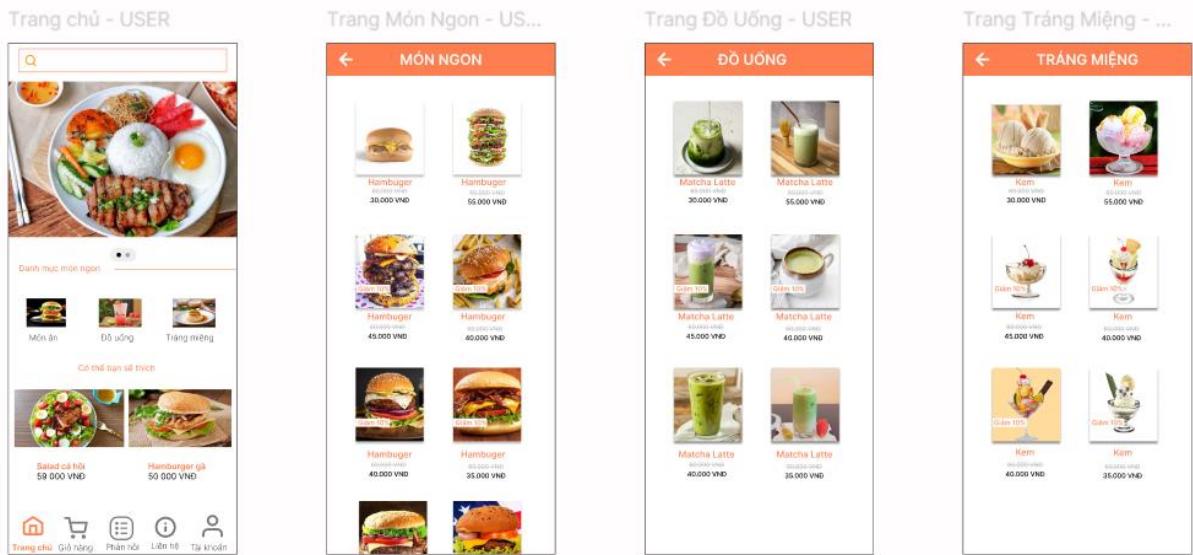
The screenshot displays the User Dashboard with four main sections:

- Trang tài khoản-USER**: Account Management page for user anhngoc123@gmail.com.
- Trang liên hệ-USER**: Contact page for Fast Food Order.
- Trang phản hồi-USER**: Feedback page where user can leave a review.
- Trang lịch sử đơn hàng...**: Order History page showing two recent orders.

Hình 4: Giao diện chức năng USER

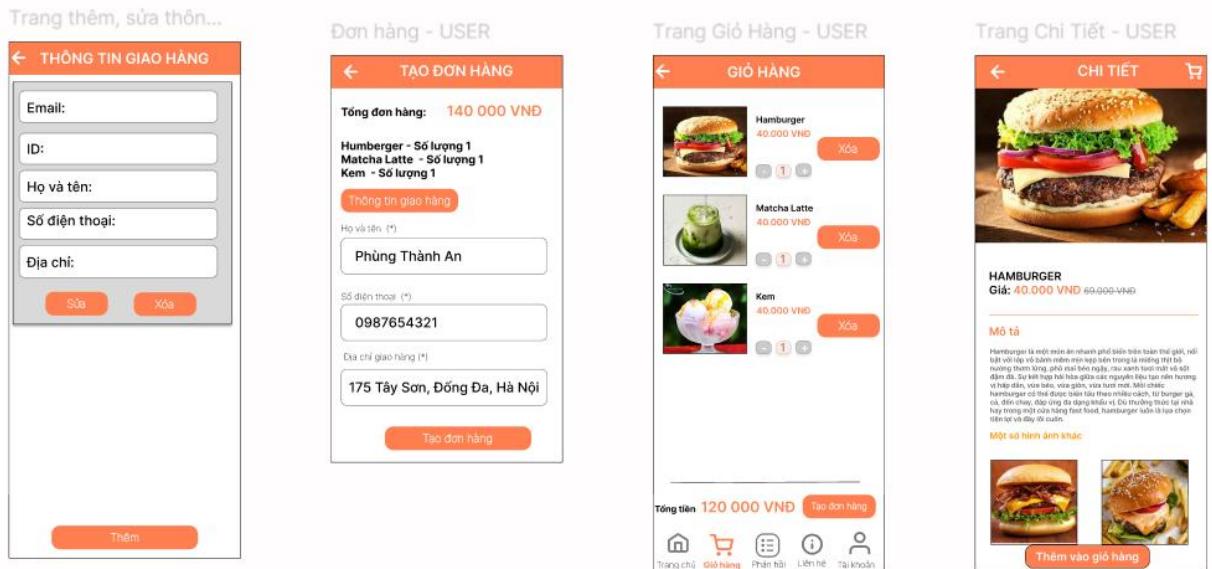
Phát triển ứng dụng cho thiết bị di động

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



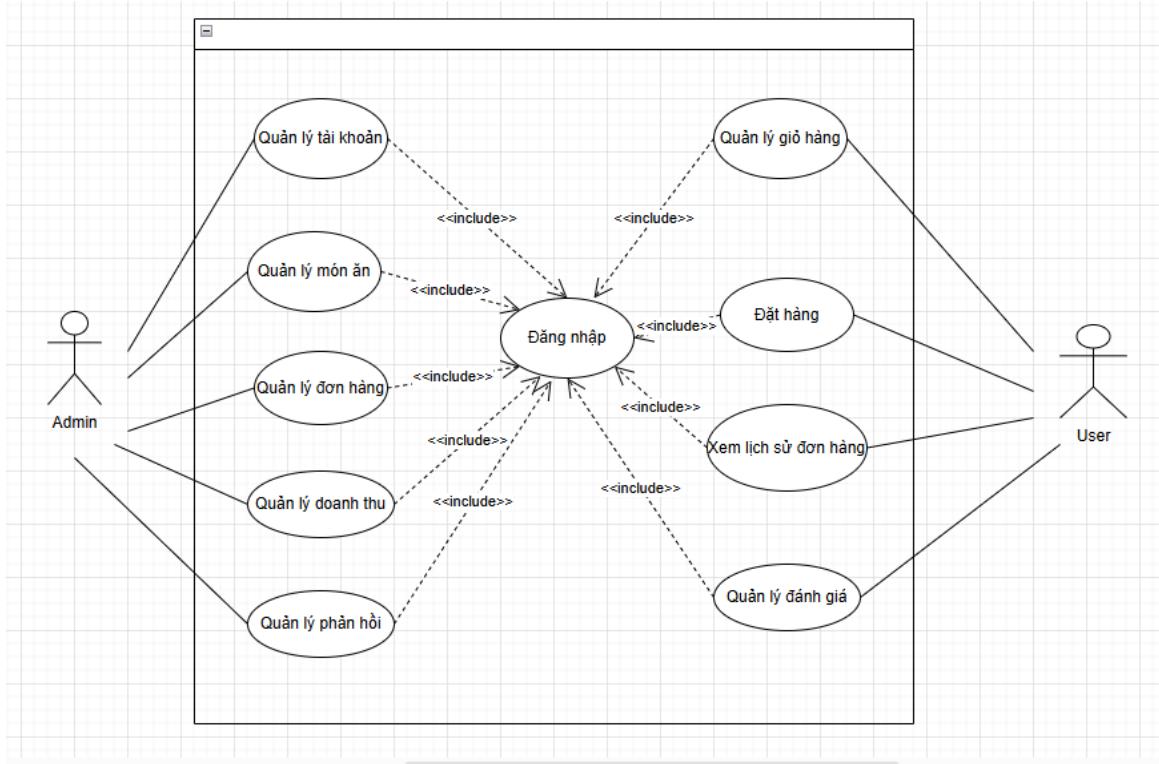
Hình 5 Giao diện thực đơn USER

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



Hình 6: Giao diện chi tiết USER

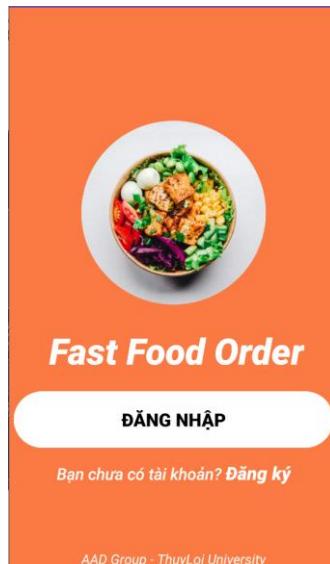
### 3.2. Thiết kế CSDL



Hình 7: Cơ sở dữ liệu

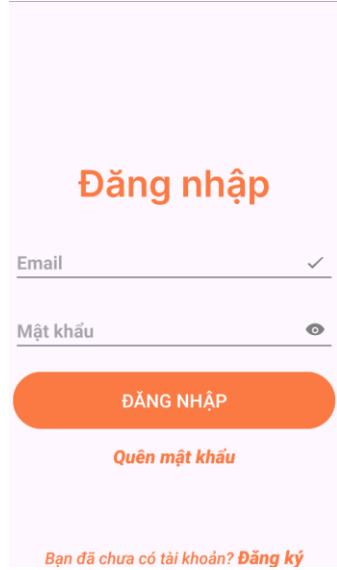
### 3.3. Giao diện ứng dụng

#### 3.3.1. Màn hình đầu tiên



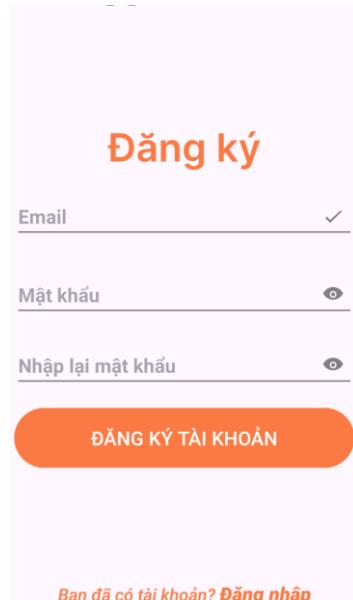
Hình 8: Giao diện màn hình đầu tiên

#### 3.3.2. Màn hình đăng nhập



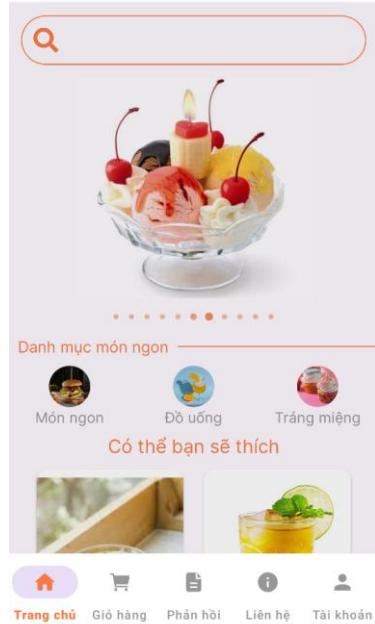
Hình 9: Giao diện đăng nhập

### 3.3.3. Màn hình đăng ký



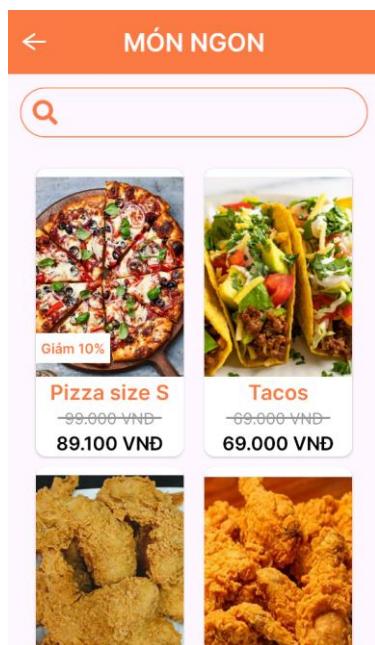
Hình 10: Giao diện đăng ký

### 3.3.4. Màn hình trang chủ user



Hình 11: Giao diện trang chủ USER

### 3.3.5. Màn hình trang món ngon user



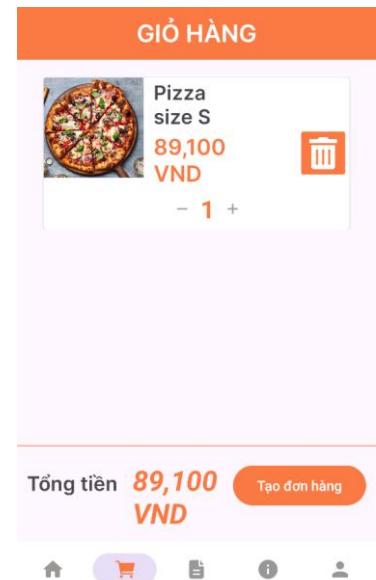
Hình 12: Giao diện trang món ngon USER

### 3.3.6. Màn hình trang chi tiết user



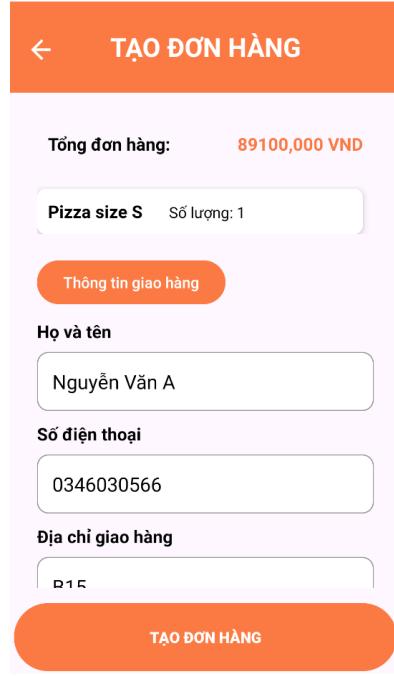
Hình 13: Giao diện chi tiết món ngon USER

### 3.3.7. Màn hình trang giỏ hàng user



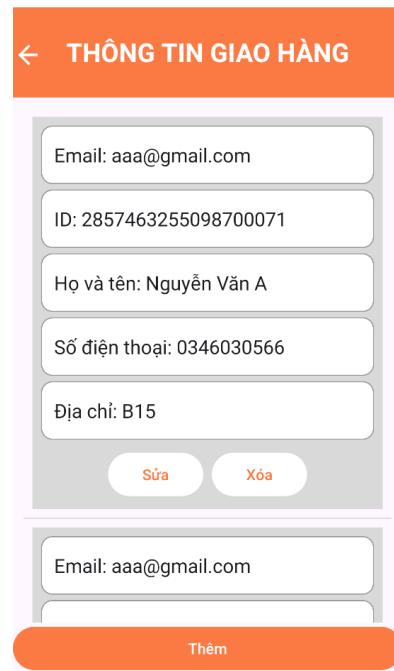
Hình 14: Giao diện giỏ hàng USER

### 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

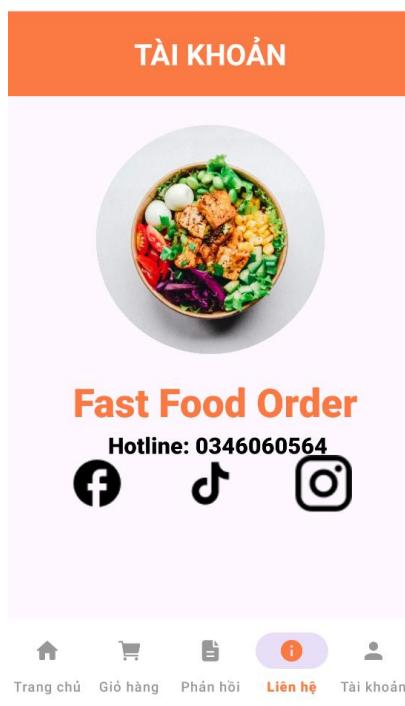
Bình luận  
Nhập bình luận

**Gửi phản hồi**

Trang chủ Giỏ hàng Phản hồi Liên hệ Tài khoản

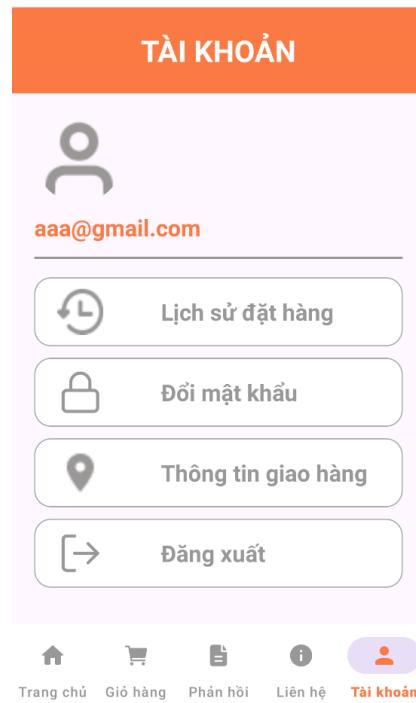
Hình 17: Giao diện phản hồi USER

### 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 screen titled 'ĐỔI MẬT KHẨU' (Change Password) in an orange header bar. Below the title, the text 'Đổi mật khẩu' is displayed in large orange letters. The form contains three input fields with eye icon placeholders: 'Mật khẩu cũ' (Old Password), 'Mật khẩu mới' (New Password), and 'Nhập lại mật khẩu mới' (Re-enter New Password). 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 screen titled 'THÊM MÓN NGON' (Add Dish) in an orange header bar. The form includes several input fields: 'Tên (\*)' (Name \*) with placeholder 'Nhập tên món', 'Mô tả' (Description) with placeholder 'Mô tả', 'Giá (\*)' (Price \*) with placeholder 'Nhập giá' and a 'VNĐ' suffix, 'Khuyến mại' (Promotion) with placeholder 'Nhập khuyến mại', and 'Ảnh chính (\*)' (Main Image \*) with a file upload icon. The background of the form area is light pink.

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

**← THÊM ĐỒ UỐNG**

Tên (\*)  
Nhập tên món

Mô tả  
Mô tả

Giá (\*)  
Nhập giá VNĐ

Khuyến mại  
Nhập khuyến mại

Ảnh chính (\*)  
Upload

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

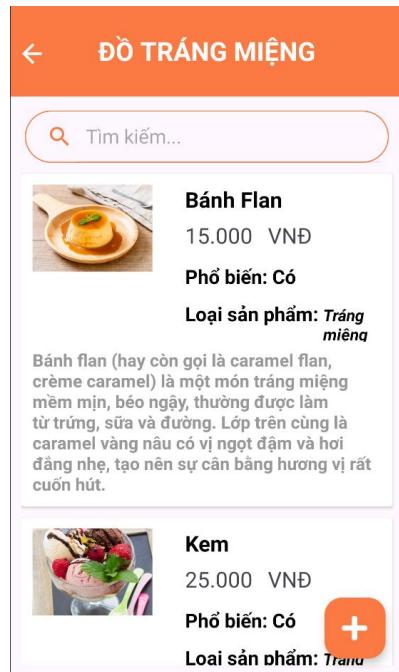
### 3.3.21. Màn hình trang sửa đồ uống admin

The screenshot shows a mobile application interface for editing a drink. At the top, a red header bar contains a back arrow icon and the text 'SỬA ĐỒ UỐNG'. The main form area has a white background with several input fields:

- A text input field labeled 'Tên (\*)' containing the value 'Matcha Latte'.
- A text input field labeled 'Mô tả' containing a description of Matcha Latte.
- A text input field labeled 'Giá (\*)' containing '55000' with a currency suffix 'VND'.
- A text input field labeled 'Khuyến mại' containing the placeholder 'Nhập khuyến mại'.
- An image input field labeled 'Ảnh chính (\*)' showing a thumbnail of the Matcha Latte drink.

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

Tên (\*)  
Nhập tên món

Mô tả  
Mô tả

Giá (\*)  
Nhập giá VNĐ

Khuyến mại  
Nhập khuyến mại

Ảnh chính (\*)  
[Upload icon]

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 five 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 tìm 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

```

```

public class FoodPageUserActivity extends AppCompatActivity {
    private void getListFood() {
        myRef.orderByChild("productType").equalTo("Món ngon").addChildEventListener(new ChildEventListener() {
            public void onChildAdded(@NotNull DataSnapshot snapshot, @Nullable String previousChildName) {
                adapter.setDataFood(list);
                adapter.notifyDataSetChanged();
            }

            @Override
            public void onChildRemoved(@NotNull DataSnapshot snapshot) {
                // Xoá lg khi cò sđy xoa dđi
                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) {
                // Xoá lg khi cò sđy thay đổi tên
            }

            @Override
            public void onCancelled(@NotNull DatabaseError error) {
                // Xoá lg khi cò 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.setDataFood(filteredList);
        adapter.notifyDataSetChanged();
    }
}

```

Hình 33 Chức năng xem trang món ngon USER

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

```

public class DrinkPageUserActivity extends AppCompatActivity {
    private RecyclerView rcv_drug;
    private DrinkUserRecycleViewAdapter adapter;
    private ImageView img_back;
    private List<Product> list = new ArrayList<>();
    private SearchView searchView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgesToEdge.enable((WindowInsetsManager) this);
        setContentView(R.layout.activity_drink_page_user);
        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;
        });
        rcv_drug = findViewById(R.id.rcv_drug);
        img_back = findViewById(R.id.img_back);
        searchView = findViewById(R.id.searchView);

        // Hỗ trợ SearchView để tìm kiếm
        searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String query) {
                filter(query);
            }
        });
    }

    private void getListDrink() {
        adapter.setDatabaseDrink(list);
        rcv_drug.setAdapter(adapter);
    }

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

```

```

public class DrinkPageUserActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        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("name").equalTo("Đồ 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).getName().equals(product.getName())) {
                            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.getName() == list.get(i).getName()) {
                        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
            }
        });
    }

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

```

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 implementation of a RecyclerView adapter for dessert items. The 'onCreate' method sets up the RecyclerView, initializes an adapter, and adds a click listener to the back button. It also includes a search bar with a query text listener that filters the data set.

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 files like CreateOrderActivity.java, ShipmentDetailPageActivity.java, AddDessertAdminActivity.java, UpdateDessertAdminActivity.java, and UpdateDrinkAdminActivity.java. 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();
        }
    });
}

```

The bottom status bar indicates the file is 2102 LF UTF-8 with 4 spaces.

Second screenshot of the same setup, showing the code for CreateOrderActivity.java again. The code is identical to the first screenshot.

```

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) {
                // 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();
            }
        });
    }
}

```

The bottom status bar indicates the file is 2102 LF UTF-8 with 4 spaces.

The screenshot shows the Android Studio interface with two main panes. The left pane displays the project structure for the 'Food\_Fast\_Order' project, which includes modules for 'app', 'test', and 'androidTest'. The 'app' module contains Java files like MainActivity.java, LoginActivity.java, and CreateOrderActivity.java, along with resources for colors, drawables, and layouts. The right pane shows the detailed code for CreateOrderActivity.java, which extends AppCompatActivity. The code handles saving order details to Firebase Database, creating new OrderDetail objects, and updating existing ones. It also manages onFailure listeners for database operations.

```
public class CreateOrderActivity extends AppCompatActivity {
    private void saveOrderDetails(String orderId) {
        DatabaseReference orderDetailRef = FirebaseDatabase.getInstance().getReference().child("orderDetails").child(orderId);
        List<Product> productList = orderAdapter.getProductList(); // Get products from adapter

        for (Product product : productList) {
            int quantity = orderAdapter.getProductQuantity(product); // Get quantity from adapter
            double price = Double.parseDouble(product.getPriceNew());
            double totalPrice = price * quantity;
        }
    }
}
```

Hình 36 Chức năng tạo đơn hàng - USER

### **3.4.5 Chức năng thêm địa chỉ - USER**

```
Android -> AddShipmentActivity.java <- CreateOrderActivity.java <- ShipmentDetailPageActivity.java <- AddDessertAdminActivity.java <- UpdateDessertAdminActivity.java
```

```
20 public class AddShipmentActivity extends AppCompatActivity {  
21     3 usages  
22     private EditText edit_email, edit_id, edit_name, edit_phone, edit_address;  
23     2 usages  
24     private Button btn_add;  
25  
26     @Override  
27     protected void onCreate(Bundle savedInstanceState) {  
28         super.onCreate(savedInstanceState);  
29         EdgeToEdge.enable();  
30         setContentView(R.layout.activity_add_shipment);  
31         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (View v, WindowInsets insets) -> {  
32             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
33             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
34             return insets;  
35         });  
36         edit_email = findViewById(R.id.edit_email);  
37         edit_id = findViewById(R.id.edit_id);  
38         edit_name = findViewById(R.id.edit_name);  
39         edit_phone = findViewById(R.id.edit_phone);  
40         edit_address = findViewById(R.id.edit_address);  
41         FirebaseAuth currentUser = FirebaseAuth.getInstance().getCurrentUser();  
42         if(currentUser != null){  
43             String email = currentUser.getEmail();  
44             edit_email.setText(email); // Hiện tài email của người dùng hiện tại trong trường edit_email  
45         } else {  
46             Toast.makeText(getApplicationContext(), "Không thể lấy thông tin người dùng hiện tại.", Toast.LENGTH_SHORT).show();  
47         }  
48  
49         btn_add = findViewById(R.id.btn_add);  
50  
51         btn_add.setOnClickListener(new View.OnClickListener() {  
52             @Override  
53             public void onClick(View v) {  
54                 // Lấy thông tin người dùng hiện tại từ Firebase Authentication  
55                 FirebaseAuth currentUser = FirebaseAuth.getInstance().getCurrentUser();  
56                 if(currentUser != null){  
57                     String email = currentUser.getEmail();  
58                     edit_email.setText(email);  
59                     Long id = Math.abs(UUID.randomUUID().getMostSignificant32Bits());  
60                     edit_id.setText(String.valueOf(id));  
61                     String name = edit_name.getText().toString();  
62                     String phone = edit_phone.getText().toString();  
63                     String address = edit_address.getText().toString();  
64                 }  
65             }  
66         });  
67     }  
68 }
```

```

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);
    }
}

```

```

public class OrderHistoryActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        imgBack = findViewById(R.id.imgBack); // Sửa lỗi ID ở đây
        imgBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { finish(); }
        });
        getListUserFromRealtimeDatabase();
    }

    private void getListUserFromRealtimeDatabase() {
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference("Orders");
        myRef.orderByChild("userId").equalTo(currentUser.getId()).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) {
                // Khi chỉ thêm 1 item
                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) {
                // Khi chỉ xóa 1 item
                Order order = snapshot.getValue(Order.class);
                if (order == null || orderList == null || orderList.isEmpty()) {
                    return;
                }
                // Xử lý khi xóa
                for (int i = 0; i < orderList.size(); i++) {
                    if (Objects.equals(order.getId(), orderList.get(i).getId())) {
                        orderList.remove(i);
                        break;
                    }
                }
                // Cập nhật RecyclerView
                orderHistoryAdapter.notifyDataSetChanged();
            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
            }

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

```

Hình 38 Chức năng xem lịch sử đơn hàng - USER

### 3.4.7 Chức năng phản hồi - USER

```

public class UserFeedbackActivity extends AppCompatActivity {
    ...
    private RecyclerView rcvFeedback;
    private FeedbackUserAdapter mAdapter;
    private List<Feedback> mListFeedback;
    private ProgressDialog progressDialog;
    ...
    private ImageButton imgBack;

    @Override
    protected 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();

        rcvFeedback = findViewById(R.id.rcv_feedback);
        mAdapter = new FeedbackUserAdapter(mListFeedback);
        rcvFeedback.setAdapter(mAdapter);
        rcvFeedback.setLayoutManager(new LinearLayoutManager(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("feedbacks");
            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 android.widget.Toast;
10
11 import androidx.annotation.NonNull;
12 import androidx.appcompat.app.AppCompatActivity;
13 import androidx.recyclerview.widget.LinearLayoutManager;
14 import androidx.recyclerview.widget.RecyclerView;
15
16 import com.google.firebase.database.DataSnapshot;
17 import com.google.firebase.database.DatabaseError;
18 import com.google.firebase.database.DatabaseReference;
19 import com.google.firebase.database.FirebaseDatabase;
20 import com.google.firebase.database.Query;
21
22 import java.util.ArrayList;
23 import java.util.List;
24
25 public class FoodPageAdminActivity extends AppCompatActivity {
26
27     private RecyclerView rcv_food;
28     private List<Product> mListProduct;
29     private DatabaseReference databaseReference;
30
31     private ValueEventListener eventListener;
32
33     private FloatingActionButton fab_add;
34
35     private ImageButton imgBack;
36
37     private SearchView search_view;
38
39     private FoodAdminAdapter adapter;
40
41     @Override
42     protected void onCreate(Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44
45         EdgeToEdge.enable( findViewById(R.id.activity_food_page_admin));
46         setContentView(R.layout.activity_food_page_admin);
47         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (View v, WindowInsetsCompat insets) -> {
48             insets.systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
49             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
50             return insets;
51         });
52
53         // init
54         rcv_food = findViewById(R.id.rcv_food);
55         fab_add = findViewById(R.id.id_fab_add);
56         imgBack = findViewById(R.id.id_imgBack);
57         search_view = findViewById(R.id.id_search_view);
58         search_view.clearFocus();
59
60         imgBack.setOnClickListener(new View.OnClickListener() {
61             @Override
62             public void onClick(View v) {
63                 Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
64                 startActivity(intent);
65             }
66         });
67
68         // An icon add để vào trang thêm
69         fab_add.setOnClickListener(new View.OnClickListener() {
70             @Override
71             public void onClick(View v) {
72                 Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
73                 startActivity(intent);
74             }
75         });
76
77         // load data
78         eventListener = query.addValueEventListener(new ValueEventListener() {
79             @Override
80             public void onDataChange(@NonNull DataSnapshot snapshot) {
81                 mListProduct.clear();
82                 for(DataSnapshot dataSnapshot: snapshot.getChildren()) {
83                     Product product = dataSnapshot.getValue(Product.class);
84                     product.setKey(dataSnapshot.getKey());
85                     mListProduct.add(product);
86                 }
87                 adapter.notifyDataSetChanged();
88                 dialog.dismiss();
89             }
90
91             @Override
92             public void onCancelled(@NonNull DatabaseError error) { dialog.dismiss(); }
93         });
94
95         search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
96             @Override
97             public void onQueryTextSubmit(String query) {
98             }
99
100            @Override
101            public void onQueryTextChange(String newText) {
102                Query query = productsRef.orderByChild("productType").equalTo("Món ăn");
103                dialog.show();
104            }
105        });
106    }
107
108    @Override
109    public void onBackPressed() {
110        Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
111        startActivity(intent);
112    }
113
114 }

```

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 android.widget.Toast;
10
11 import androidx.annotation.NonNull;
12 import androidx.appcompat.app.AppCompatActivity;
13 import androidx.recyclerview.widget.LinearLayoutManager;
14 import androidx.recyclerview.widget.RecyclerView;
15
16 import com.google.firebase.database.DataSnapshot;
17 import com.google.firebase.database.DatabaseError;
18 import com.google.firebase.database.DatabaseReference;
19 import com.google.firebase.database.FirebaseDatabase;
20 import com.google.firebase.database.Query;
21
22 import java.util.ArrayList;
23 import java.util.List;
24
25 public class FoodPageAdminActivity extends AppCompatActivity {
26
27     private RecyclerView rcv_food;
28     private List<Product> mListProduct;
29     private DatabaseReference databaseReference;
30
31     private ValueEventListener eventListener;
32
33     private FloatingActionButton fab_add;
34
35     private ImageButton imgBack;
36
37     private SearchView search_view;
38
39     private FoodAdminAdapter adapter;
40
41     @Override
42     protected void onCreate(Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44
45         EdgeToEdge.enable( findViewById(R.id.activity_food_page_admin));
46         setContentView(R.layout.activity_food_page_admin);
47         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (View v, WindowInsetsCompat insets) -> {
48             insets.systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
49             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
50             return insets;
51         });
52
53         // init
54         rcv_food = findViewById(R.id.rcv_food);
55         fab_add = findViewById(R.id.id_fab_add);
56         imgBack = findViewById(R.id.id_imgBack);
57         search_view = findViewById(R.id.id_search_view);
58         search_view.clearFocus();
59
60         imgBack.setOnClickListener(new View.OnClickListener() {
61             @Override
62             public void onClick(View v) {
63                 Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
64                 startActivity(intent);
65             }
66         });
67
68         // An icon add để vào trang thêm
69         fab_add.setOnClickListener(new View.OnClickListener() {
70             @Override
71             public void onClick(View v) {
72                 Intent intent = new Intent(getApplicationContext(), AddFoodAdminActivity.class);
73                 startActivity(intent);
74             }
75         });
76
77         // load data
78         eventListener = query.addValueEventListener(new ValueEventListener() {
79             @Override
80             public void onDataChange(@NonNull DataSnapshot snapshot) {
81                 mListProduct.clear();
82                 for(DataSnapshot dataSnapshot: snapshot.getChildren()) {
83                     Product product = dataSnapshot.getValue(Product.class);
84                     product.setKey(dataSnapshot.getKey());
85                     mListProduct.add(product);
86                 }
87                 adapter.notifyDataSetChanged();
88                 dialog.dismiss();
89             }
90
91             @Override
92             public void onCancelled(@NonNull DatabaseError error) { dialog.dismiss(); }
93         });
94
95         search_view.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
96             @Override
97             public void onQueryTextSubmit(String query) {
98             }
99
100            @Override
101            public void onQueryTextChange(String newText) {
102                Query query = productsRef.orderByChild("productType").equalTo("Món ăn");
103                dialog.show();
104            }
105        });
106    }
107
108    @Override
109    public void onBackPressed() {
110        Intent intent = new Intent(getApplicationContext(), HomeAdminActivity.class);
111        startActivity(intent);
112    }
113
114 }

```

```

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 Them
        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

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

    // code x
    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_imgSlider = findViewById(R.id.detail_imgSlider);
    detail_imgOther = findViewById(R.id.detail_imgOther);
    btn_back = findViewById(R.id.btn_back);
    delete_btn = findViewById(R.id.delete_btn);
    edt_btn = findViewById(R.id.edt_btn);

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

    // Nhận sản phẩm từ adapter
    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"));
        key = bundle.getString("key");
        imgUrl = bundle.getString("key_imgUrl");
        imgOtherUrl = bundle.getString("key_imgOther");
    }
}
...
}

// Khi nhấn nút delete
delete_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { showDeleteConfirmationDialog(); }
});

// Khi nhấn nút edit
edt_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("imgUrl", imgUrl);
        intent.putExtra("imgOther", imgOtherUrl);
        intent.putExtra("imgSlider", imgSliderUrl);
        intent.putExtra("key", key);
        startActivity(intent);
    }
});

// Phương thức hiển thị hộp thoại xác nhận xóa sản phẩm
private void showDeleteConfirmationDialog() {
    new AlertDialog.Builder(this)
        .setTitle("Xác nhận xóa")
        .setMessage("Bạn có chắc chắn muốn xóa sản phẩm này không?")
        .setPositiveButton("Đồng ý", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) { deleteProduct(); }
        })
        .setNegativeButton("Không", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) { dialog.cancel(); }
        })
        .setIcon(android.R.drawable.ic_dialog_alert)
        .show();
}

// Phương thức xóa sản phẩm và các ảnh liên quan từ Firebase Storage và 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) {
            // Xóa ảnh chính
            deleteImageFromStorageAndFinish(imgUrl);
            // Xóa ảnh phụ
            deleteImageFromStorageAndFinish(imgOtherUrl);
            // Xóa ảnh slider
            deleteImageFromStorageAndFinish(imgSliderUrl);
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(getApplicationContext(), "Xóa sản phẩm khỏi Realtime Database thất bại: " + e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
}

// Phương thức xóa ảnh từ Firebase Storage và kết thúc hoạt động nếu thành công hoặc thất bại
private void deleteImageFromStorageAndFinish(String imageUrl) {
    ...
}
```

#### 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 }

```

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

```

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; // Mdc 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("mainImage").putFile(uriMainImage)
  81                 .child(uriMainImage.getLastPathSegment());
  82
  83             AlertDialog.Builder builder = new AlertDialog.Builder(context, AddDrinkAdminActivity.this);
  84             builder.setCancelable(false);
  85             builder.setView(R.layout.progress_admin_layout);
  86             AlertDialog dialog = builder.create();
  87             dialog.show();
  88
  89             storageReference.putFile(uriMainImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
  90                 @Override
  91
  92             public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  93                 Task<Uri> urlTask = taskSnapshot.getStorage().getDownloadUrl();
  94                 while (!urlTask.isComplete());
  95                 Uri urlImage = urlTask.getResult();
  96                 imgMainImage = urlImage.toString();
  97                 saveSliderImage(dialog);
  98             }
  99         }).addOnFailureListener(new OnFailureListener() {
  100            @Override
  101            public void onFailure(@NonNull Exception e) {
  102                dialog.dismiss();
  103                Toast.makeText(context, AddDrinkAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
  104            }
  105        });
  106
  107        // Hàm upload ảnh slider lên Firebase
  108        // Lỗi
  109        private void saveSliderImage(AlertDialog dialog) {
  110            if (uriSliderImage == null) {
  111                Toast.makeText(context, AddDrinkAdminActivity.this, "Vui lòng chọn hình ảnh slider", Toast.LENGTH_SHORT).show();
  112                dialog.dismiss();
  113                return;
  114            }
  115
  116            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("mainImage")
  117                 .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("mainImage")
  147                 .child(uriOtherImage.getLastPathSegment());
  148
  149            storageReference.putFile(uriOtherImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
  150                 @Override
  151                 public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  152                     Task<Uri> urlTask = taskSnapshot.getStorage().getDownloadUrl();
  153                     while (!urlTask.isComplete());
  154                     Uri urlImage = urlTask.getResult();
  155                     imgOtherURI = urlImage.toString();
  156                     uploadData(dialog);
  157                 }
  158             }).addOnFailureListener(new OnFailureListener() {
  159                 @Override
  160                 public void onFailure(@NonNull Exception e) {
  161                     dialog.dismiss();
  162                     Toast.makeText(context, AddDrinkAdminActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
  163                 }
  164             });
  165
  166        // Hàm upload dữ liệu sản phẩm lên Firebase
  167
  168    }

```

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.setValValue(currentProductId);
 99                 // Tạo ID trên già trả mới currentProductId
100                 String id = String.valueOf(currentProductId);
101
102                 FirebaseDatabase database = FirebaseDatabase.getInstance();
103                 DatabaseReference reference = database.getReference("products").child(id);
104                 reference.setValue(product).addOnCompleteListener(new OnCompleteListener<Void>() {
105                     @Override
106                     public void onComplete(@NonNull Task<Void> task) {
107                         if (!task.isSuccessful()) {
108                             Toast.makeText(getApplicationContext(), "Thêm sản phẩm thành công", Toast.LENGTH_SHORT).show();
109                             dialog.dismiss();
110                             Intent intent = new Intent(getApplicationContext(), AddDrinkAdminActivity.this, DrinkPageAdminActivity.class);
111                             startActivity(intent);
112                         }
113                     }
114                 }).addOnFailureListener(new OnFailureListener() {
115                     @Override
116                     public void onFailure(@NonNull Exception e) {
117                         dialog.dismiss();
118                         Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
119                     }
120                 });
121             } catch (IllegalStateException e) {
122                 dialog.dismiss();
123                 Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
124             } catch (Exception e) {
125                 dialog.dismiss();
126                 Toast.makeText(getApplicationContext(), "Đã xảy ra lỗi: " + e.getMessage(), Toast.LENGTH_SHORT).show();
127             }
128         }
129     }

```

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          super.onCreate(savedInstanceState);
  48          setContentView(R.layout.activity_main);
  49          imgOther.setImageResource(R.drawable.ic_launcher_foreground);
  50          imgMain.setImageResource(R.drawable.ic_launcher_foreground);
  51          imgBack.setImageResource(R.drawable.ic_launcher_foreground);
  52          imgSlider.setImageResource(R.drawable.ic_launcher_foreground);
  53          imgOther.setOnClickListener(new View.OnClickListener() {
  54              @Override
  55              public void onClick(View v) {
  56                  Intent intent = new Intent(AddDessertAdminActivity.this, AddDessertAdminActivity.class);
  57                  intent.putExtra("image", "other");
  58                  startActivity(intent);
  59              }
  60          });
  61          imgMain.setOnClickListener(new View.OnClickListener() {
  62              @Override
  63              public void onClick(View v) {
  64                  Intent intent = new Intent(AddDessertAdminActivity.this, AddDessertAdminActivity.class);
  65                  intent.putExtra("image", "main");
  66                  startActivity(intent);
  67              }
  68          });
  69          imgBack.setOnClickListener(new View.OnClickListener() {
  70              @Override
  71              public void onClick(View v) {
  72                  Intent intent = new Intent(AddDessertAdminActivity.this, AddDessertAdminActivity.class);
  73                  intent.putExtra("image", "back");
  74                  startActivity(intent);
  75              }
  76          });
  77          imgSlider.setOnClickListener(new View.OnClickListener() {
  78              @Override
  79              public void onClick(View v) {
  80                  Intent intent = new Intent(AddDessertAdminActivity.this, AddDessertAdminActivity.class);
  81                  intent.putExtra("image", "slider");
  82                  startActivity(intent);
  83              }
  84          });
  85      }
  86      // Hàm upload ảnh chính lên Firebase
  87      private void saveMainImage() {
  88          if (uriMainImage == null) {
  89              Toast.makeText(getApplicationContext(), "Vui lòng chọn hình ảnh chính", Toast.LENGTH_SHORT).show();
  90              return;
  91          }
  92          StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("images").child("Save Images");
  93          storageReference.putFile(uriMainImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
  94              @Override
  95              public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  96                  TaskUri uriTask = taskSnapshot.getStorage().getDownloadUrl();
  97                  while (!uriTask.isComplete());
  98                  Uri urlImage = uriTask.getResult();
  99                  imgMainURL = urlImage.toString();
  100             }
  101         }).addOnFailureListener(new OnFailureListener() {
  102             @Override
  103             public void onFailure(@NonNull Exception e) {
  104                 dialog.dismiss();
  105                 Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
  106             }
  107         });
  108     }
  109     // Hàm upload ảnh slider lên Firebase
  110     private void saveSliderImage(AlertDialog dialog) {
  111         if (uriSliderImage == null) {
  112             Toast.makeText(getApplicationContext(), "Vui lòng chọn hình ảnh slider", Toast.LENGTH_SHORT).show();
  113             dialog.dismiss();
  114             return;
  115         }
  116         StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("images").child("Save Images");
  117         storageReference.putFile(uriSliderImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
  118             @Override
  119             public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  120                 TaskUri uriTask = taskSnapshot.getStorage().getDownloadUrl();
  121                 while (!uriTask.isComplete());
  122                 Uri urlImage = uriTask.getResult();
  123                 imgSliderURL = urlImage.toString();
  124                 saveOtherImage(dialog);
  125             }
  126         }).addOnFailureListener(new OnFailureListener() {
  127             @Override
  128             public void onFailure(@NonNull Exception e) {
  129                 dialog.dismiss();
  130                 Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
  131             }
  132         });
  133     }
  134     // Hàm upload ảnh khác lên Firebase
  135     private void saveOtherImage(AlertDialog dialog) {
  136         if (uriOtherImage == null) {
  137             Toast.makeText(getApplicationContext(), "Vui lòng chọn hình ảnh khác", Toast.LENGTH_SHORT).show();
  138             dialog.dismiss();
  139             return;
  140         }
  141         StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("images").child("Save Images");
  142         storageReference.putFile(uriOtherImage).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
  143             @Override
  144             public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
  145                 TaskUri uriTask = taskSnapshot.getStorage().getDownloadUrl();
  146                 while (!uriTask.isComplete());
  147                 Uri urlImage = uriTask.getResult();
  148                 imgOtherURL = urlImage.toString();
  149                 uploadData(dialog);
  150             }
  151         }).addOnFailureListener(new OnFailureListener() {
  152             @Override
  153             public void onFailure(@NonNull Exception e) {
  154                 dialog.dismiss();
  155                 Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
  156             }
  157         });
  158     }
  159     // Hàm upload dữ liệu sản phẩm lên Firebase
  160     private void uploadData(AlertDialog dialog) {
  161         try {
  162             String name = edit_name.getText().toString();
  163             String desc = edit_desc.getText().toString();
  164             String price = edit_price.getText().toString();
  165             String sale = edit_sale.getText().toString();
  166             Boolean popular = rad_popular.isChecked(); // Lưu giá trị từ RadioButton
  167         } catch (Exception e) {
  168             e.printStackTrace();
  169         }
  170     }
  171 }

```

```

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("imgurl")).into(update_imgUrl);
 49                 Glide.with(this)
 50                     .load(bundle.getString("imgSlider")).into(update_imgSlider);
 51                 Glide.with(this)
 52                     .load(bundle.getString("imgOther")).into(update_imgOther);
 53                 update_edit_name.setText(bundle.getString("name"));
 54                 update_edit_desc.setText(bundle.getString("desc"));
 55                 update_edit_price.setText(bundle.getString("price"));
 56                 update_edit_sale.setText(bundle.getString("sale"));
 57                 // Nhận chuỗi "popular" và thiết lập RadioButton
 58                 String popularStatus = bundle.getString("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("productType");
 67                 if (productType != null) {
 68                     switch (productType) {
 69                         case "Hàng rau":
 70                             update_rad_product_type1.setChecked(true);
 71                         break;
 72                         case "Rau sống":
 73                             update_rad_product_type_drink.setChecked(true);
 74                         break;
 75                         case "Trứng miếng":
 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");
 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 }
```

```


    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

```

1 README.md UpdateFoodAdminActivity.java UpdateDrinkAdminActivity.java
2 package com.example.android_food_app.ActivityAdmin;
3
4 import ...
5
6 public class UpdateDrinkAdminActivity extends AppCompatActivity {
7     3 usages
8     private EditText update_edt_name, update_edt_desc, update_edt_price, update_edt_sale;
9     4 usages
10    private ImageView update_imgUrl, update_imgSlider, update_imgOther;
11    3 usages
12    private RadioButton update_rad_popular1, update_rad_popular2, update_rad_product_type, update_rad_product_type_drink, update_rad_product_type_dessert;
13    2 usages
14    private RadioGroup update_radioGroup, product_type;
15    2 usages
16    private Button btn_update;
17    2 usages
18    private ImageButton imgBack;
19    2 usages
20    private String name, desc, price, sale;
21    3 usages
22    private String imgUrl, imgSlider, imgOther;
23    2 usages
24    private String key, oldImgUrl, oldImgSlider, oldImgOther;
25    6 usages
26    private Uri imgUrl, urlImgSlider, urlImgOther;
27    2 usages
28    DatabaseReference databaseReference;
29    no usages
30    StorageReference storageReference;
31
32    @Override
33    protected void onCreate(Bundle savedInstanceState) {
34        super.onCreate(savedInstanceState);
35        EdgeToEdge.enable(UpdateDrinkAdminActivity.this);
36        setContentView(R.layout.activity_update_drink_admin);
37        ViewCompat.setOnApplyWindowInsetsListener(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 README.md UpdateFoodAdminActivity.java UpdateDrinkAdminActivity.java
49
50    public class UpdateDrinkAdminActivity extends AppCompatActivity {
51        protected void onCreate(Bundle savedInstanceState) {
52            update_edt_name = findViewById(R.id.update_edt_name);
53            update_edt_desc = findViewById(R.id.update_edt_desc);
54            update_edt_price = findViewById(R.id.update_edt_price);
55            update_edt_sale = findViewById(R.id.update_edt_sale);
56            update_imgUrl = findViewById(R.id.update_imgUrl);
57            update_imgSlider = findViewById(R.id.update_imgSlider);
58            update_imgOther = findViewById(R.id.update_imgOther);
59            update_radPopular1 = findViewById(R.id.update_rad_popular1);
60            update_radPopular2 = findViewById(R.id.update_rad_popular2);
61            update_radProduct_type = findViewById(R.id.update_rad_product_type);
62            update_radProduct_type_drink = findViewById(R.id.update_rad_product_type_drink);
63            update_radProduct_type_dessert = findViewById(R.id.update_rad_product_type_dessert);
64            update_radioGroup = findViewById(R.id.update_radioGroup, product_type);
65            imgBack = findViewById(R.id.imgBack);
66            btn_update = findViewById(R.id.btn_update);
67
68            imgBack.setOnClickListener(new View.OnClickListener() {
69                @Override
70                public void onClick(View v) { finish(); }
71            });
72
73            // ActivityResultLauncher ảnh chính
74            ActivityResultLauncher<Intent> activityResultLauncher1 = registerForActivityResult(
75                new ActivityResultContracts.StartActivityForResult(),
76                new ActivityResultCallback<ActivityResult>() {
77                    @Override
78                    public void onActivityResult(ActivityResult result) {
79                        if (result.getResultCode() == Activity.RESULT_OK) {
80                            Intent data = result.getData();
81                            imgUrl = data.getData();
82                            update_imgUrl.setImageURI(imgUrl);
83                        } else {
84                            Toast.makeText(context, UpdateDrinkAdminActivity.this, text: "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
85                        }
86                    }
87                }
88            );
89
90            // ActivityResultLauncher ảnh Slider
91            ActivityResultLauncher<Intent> activityResultLauncher2 = registerForActivityResult(
92                new ActivityResultContracts.StartActivityForResult(),
93
94 README.md UpdateFoodAdminActivity.java UpdateDrinkAdminActivity.java
95
96    public class UpdateDrinkAdminActivity extends AppCompatActivity {
97        protected void onCreate(Bundle savedInstanceState) {
98
99            // ActivityResultLauncher ảnh Slider
100            ActivityResultLauncher<Intent> activityResultLauncher2 = registerForActivityResult(
101                new ActivityResultContracts.StartActivityForResult(),
102                new ActivityResultCallback<ActivityResult>() {
103                    @Override
104                    public void onActivityResult(ActivityResult result) {
105                        if (result.getResultCode() == Activity.RESULT_OK) {
106                            Intent data = result.getData();
107                            urlImgSlider = data.getData();
108                            update_imgSlider.setImageURI(urlImgSlider);
109                        } else {
110                            Toast.makeText(context, UpdateDrinkAdminActivity.this, text: "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
111                        }
112                    }
113                }
114            );
115
116            // ActivityResultLauncher ảnh Other
117            ActivityResultLauncher<Intent> activityResultLauncher3 = registerForActivityResult(
118                new ActivityResultContracts.StartActivityForResult(),
119                new ActivityResultCallback<ActivityResult>() {
120                    @Override
121                    public void onActivityResult(ActivityResult result) {
122                        if (result.getResultCode() == Activity.RESULT_OK) {
123                            Intent data = result.getData();
124                            urlImgOther = data.getData();
125                            update_imgOther.setImageURI(urlImgOther);
126                        } else {
127                            Toast.makeText(context, UpdateDrinkAdminActivity.this, text: "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
128                        }
129                    }
130                }
131            );
132
133            // Khiển sản phẩm từ Retrofit gửi về
134            Bundle bundle = getIntent().getExtras();
135            if (bundle != null) {
136                Glide.with(activity).load(UpdateDrinkAdminActivity.this).load(bundle.getString(key: "imgUrl")).into(update_imgUrl);
137                Glide.with(activity).load(UpdateDrinkAdminActivity.this).load(bundle.getString(key: "imgSlider")).into(update_imgSlider);
138                Glide.with(activity).load(UpdateDrinkAdminActivity.this).load(bundle.getString(key: "imgOther")).into(update_imgOther);
139
140        }
141
142    }
143
144
145

```

36 33

```
1 README.md  UpdateFoodAdminActivity.java  UpdateDrinkAdminActivity.java
2
3 public class UpdateFoodAdminActivity extends AppCompatActivity {
4     protected void onCreate(Bundle savedInstanceState) {
5
6         // Nhận sản phẩm từ Detial gửi về
7         Bundle bundle = getIntent().getExtras();
8         if (bundle != null) {
9             update_imgUrl.set("image", UpdateFoodAdminActivity.this).load(bundle.getString("key", "imgUrl")).into(update_imgUrl);
10            update_drinkImgUrl.set("image", UpdateDrinkAdminActivity.this).load(bundle.getString("key", "imgSlider")).into(update_imgSlider);
11            Glide.with(getApplicationContext()).load(bundle.getString("key", "imgOther")).into(update_imgOther);
12
13            update_edt_name.setText(bundle.getString("key", "name"));
14            update_edt_desc.setText(bundle.getString("key", "desc"));
15            update_edt_price.setText(bundle.getString("key", "price"));
16            update_edt_sale.setText(bundle.getString("key", "sale"));
17
18            // Khi chon "popular" va thiet lap RadioButton
19            String popularStatus = bundle.getString("key", "popular");
20            if (popularStatus != null) {
21                if (popularStatus.equals("true")) {
22                    update_rad_popular1.setChecked(true);
23                } else if (popularStatus.equals("false")) {
24                    update_rad_popular2.setChecked(true);
25                }
26            }
27
28            String productType = bundle.getString("key", "productType");
29            if (productType != null) {
30                switch (productType) {
31                    case "Món ngon":
32                        update_rad_product_type.setChecked(true);
33                        break;
34
35                    case "Bia uống":
36                        update_rad_product_type_drink.setChecked(true);
37                        break;
38
39                    case "Tráng miệng":
40                        update_rad_product_type_dessert.setChecked(true);
41                        break;
42
43                    default:
44                        // xử lý trường hợp không tìm thấy productType nào phù hợp
45                        break;
46                }
47            }
48
49            key = bundle.getString("key");
50            oldImgUrl = bundle.getString("key", "imgUrl");
51            oldImgSlider = bundle.getString("key", "imgSlider");
52            oldImgOther = bundle.getString("key", "imgOther");
53        }
54    }
55
56    @Override
57    protected void onCreate(Bundle savedInstanceState) {
58
59        databaseReference = FirebaseDatabase.getInstance().getReference("path:products");
60
61        update_imgUrl.setOnClickListener(new View.OnClickListener() {
62            @Override
63            public void onClick(View v) {
64                Intent photoPicker = new Intent(Intent.ACTION_PICK);
65                photoPicker.setType("image/*");
66                activityResultLauncher1.launch(photoPicker);
67            }
68        });
69
70        update_imgSlider.setOnClickListener(new View.OnClickListener() {
71            @Override
72            public void onClick(View v) {
73                Intent photoPicker = new Intent(Intent.ACTION_PICK);
74                photoPicker.setType("image/*");
75                activityResultLauncher2.launch(photoPicker);
76            }
77        });
78
79        update_imgOther.setOnClickListener(new View.OnClickListener() {
80            @Override
81            public void onClick(View v) {
82                Intent photoPicker = new Intent(Intent.ACTION_PICK);
83                photoPicker.setType("image/*");
84                activityResultLauncher3.launch(photoPicker);
85            }
86        });
87
88        btn_update.setOnClickListener(new View.OnClickListener() {
89            @Override
90            public void onClick(View v) {
91                updateMainImage();
92                Intent intent = new Intent(getApplicationContext(), UpdateDrinkAdminActivity.this, DrinkPageAdminActivity.class);
93                startActivity(intent);
94            }
95        });
96    }
97
98    private void updateMainImage() {
99        if (urlImgUrl == null) {
100            updateSliderImage();
101        } else {
102            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
103                .child(urlImgUrl).getLastPathSegment();
104
105            AlertDialog.Builder builder = new AlertDialog.Builder(getApplicationContext());
106            builder.setCancelable(false);
107            builder.setView(R.layout.progress_admin_layout);
108            AlertDialog dialog = builder.create();
109            dialog.show();
110
111            storageReference.putFile(Uri.parse(urlImgUrl)).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
112                @Override
113                public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
114                    taskSnapshot.getStorage().getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
115                        @Override
116                        public void onSuccess(Uri uri) {
117                            imgUrl = uri.toString();
118                            updateSliderImage();
119                        }
120                    });
121                }
122            }).addOnFailureListener(new OnFailureListener() {
123                @Override
124                public void onFailure(@NonNull Exception e) {
125                    dialog.dismiss();
126                    Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
127                }
128            });
129        }
130    }
131
132    private void updateSliderImage() {
133        if (urlImgSlider == null) {
134            updateOtherImage();
135        } else {
136            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
137                .child(urlImgSlider).getLastPathSegment();
138        }
139    }
140
141    private void updateOtherImage() {
142        if (urlImgOther == null) {
143            updateImgOtherImage();
144        } else {
145            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
146                .child(urlImgOther).getLastPathSegment();
147        }
148    }
149
150    private void updateImgOtherImage() {
151        if (urlImgOther == null) {
152            updateImgOtherImage();
153        } else {
154            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
155                .child(urlImgOther).getLastPathSegment();
156        }
157    }
158
159    private void updateImgOtherImage() {
160        if (urlImgOther == null) {
161            updateImgOtherImage();
162        } else {
163            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
164                .child(urlImgOther).getLastPathSegment();
165        }
166    }
167
168    private void updateImgOtherImage() {
169        if (urlImgOther == null) {
170            updateImgOtherImage();
171        } else {
172            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
173                .child(urlImgOther).getLastPathSegment();
174        }
175    }
176
177    private void updateImgOtherImage() {
178        if (urlImgOther == null) {
179            updateImgOtherImage();
180        } else {
181            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
182                .child(urlImgOther).getLastPathSegment();
183        }
184    }
185
186    private void updateImgOtherImage() {
187        if (urlImgOther == null) {
188            updateImgOtherImage();
189        } else {
190            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
191                .child(urlImgOther).getLastPathSegment();
192        }
193    }
194
195    private void updateImgOtherImage() {
196        if (urlImgOther == null) {
197            updateImgOtherImage();
198        } else {
199            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
200                .child(urlImgOther).getLastPathSegment();
201        }
202    }
203
204    private void updateImgOtherImage() {
205        if (urlImgOther == null) {
206            updateImgOtherImage();
207        } else {
208            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
209                .child(urlImgOther).getLastPathSegment();
210        }
211    }
212
213    private void updateImgOtherImage() {
214        if (urlImgOther == null) {
215            updateImgOtherImage();
216        } else {
217            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
218                .child(urlImgOther).getLastPathSegment();
219        }
220    }
221
222    private void updateImgOtherImage() {
223        if (urlImgOther == null) {
224            updateImgOtherImage();
225        } else {
226            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
227                .child(urlImgOther).getLastPathSegment();
228        }
229    }
230
231    private void updateImgOtherImage() {
232        if (urlImgOther == null) {
233            updateImgOtherImage();
234        } else {
235            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
236                .child(urlImgOther).getLastPathSegment();
237        }
238    }
239
240    private void updateImgOtherImage() {
241        if (urlImgOther == null) {
242            updateImgOtherImage();
243        } else {
244            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
245                .child(urlImgOther).getLastPathSegment();
246        }
247    }
248
249    private void updateImgOtherImage() {
250        if (urlImgOther == null) {
251            updateImgOtherImage();
252        } else {
253            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
254                .child(urlImgOther).getLastPathSegment();
255        }
256    }
257
258    private void updateImgOtherImage() {
259        if (urlImgOther == null) {
260            updateImgOtherImage();
261        } else {
262            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
263                .child(urlImgOther).getLastPathSegment();
264        }
265    }
266
267    private void updateImgOtherImage() {
268        if (urlImgOther == null) {
269            updateImgOtherImage();
270        } else {
271            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
272                .child(urlImgOther).getLastPathSegment();
273        }
274    }
275
276    private void updateImgOtherImage() {
277        if (urlImgOther == null) {
278            updateImgOtherImage();
279        } else {
280            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
281                .child(urlImgOther).getLastPathSegment();
282        }
283    }
284
285    private void updateImgOtherImage() {
286        if (urlImgOther == null) {
287            updateImgOtherImage();
288        } else {
289            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
290                .child(urlImgOther).getLastPathSegment();
291        }
292    }
293
294    private void updateImgOtherImage() {
295        if (urlImgOther == null) {
296            updateImgOtherImage();
297        } else {
298            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
299                .child(urlImgOther).getLastPathSegment();
300        }
301    }
302
303    private void updateImgOtherImage() {
304        if (urlImgOther == null) {
305            updateImgOtherImage();
306        } else {
307            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
308                .child(urlImgOther).getLastPathSegment();
309        }
310    }
311
312    private void updateImgOtherImage() {
313        if (urlImgOther == null) {
314            updateImgOtherImage();
315        } else {
316            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
317                .child(urlImgOther).getLastPathSegment();
318        }
319    }
320
321    private void updateImgOtherImage() {
322        if (urlImgOther == null) {
323            updateImgOtherImage();
324        } else {
325            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
326                .child(urlImgOther).getLastPathSegment();
327        }
328    }
329
330    private void updateImgOtherImage() {
331        if (urlImgOther == null) {
332            updateImgOtherImage();
333        } else {
334            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
335                .child(urlImgOther).getLastPathSegment();
336        }
337    }
338
339    private void updateImgOtherImage() {
340        if (urlImgOther == null) {
341            updateImgOtherImage();
342        } else {
343            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
344                .child(urlImgOther).getLastPathSegment();
345        }
346    }
347
348    private void updateImgOtherImage() {
349        if (urlImgOther == null) {
350            updateImgOtherImage();
351        } else {
352            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
353                .child(urlImgOther).getLastPathSegment();
354        }
355    }
356
357    private void updateImgOtherImage() {
358        if (urlImgOther == null) {
359            updateImgOtherImage();
360        } else {
361            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
362                .child(urlImgOther).getLastPathSegment();
363        }
364    }
365
366    private void updateImgOtherImage() {
367        if (urlImgOther == null) {
368            updateImgOtherImage();
369        } else {
370            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
371                .child(urlImgOther).getLastPathSegment();
372        }
373    }
374
375    private void updateImgOtherImage() {
376        if (urlImgOther == null) {
377            updateImgOtherImage();
378        } else {
379            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
380                .child(urlImgOther).getLastPathSegment();
381        }
382    }
383
384    private void updateImgOtherImage() {
385        if (urlImgOther == null) {
386            updateImgOtherImage();
387        } else {
388            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
389                .child(urlImgOther).getLastPathSegment();
390        }
391    }
392
393    private void updateImgOtherImage() {
394        if (urlImgOther == null) {
395            updateImgOtherImage();
396        } else {
397            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
398                .child(urlImgOther).getLastPathSegment();
399        }
399    }
400
401    private void updateImgOtherImage() {
402        if (urlImgOther == null) {
403            updateImgOtherImage();
404        } else {
405            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
406                .child(urlImgOther).getLastPathSegment();
407        }
407    }
408
409    private void updateImgOtherImage() {
410        if (urlImgOther == null) {
411            updateImgOtherImage();
412        } else {
413            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
414                .child(urlImgOther).getLastPathSegment();
415        }
415    }
416
417    private void updateImgOtherImage() {
418        if (urlImgOther == null) {
419            updateImgOtherImage();
420        } else {
421            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
422                .child(urlImgOther).getLastPathSegment();
423        }
423    }
424
425    private void updateImgOtherImage() {
426        if (urlImgOther == null) {
427            updateImgOtherImage();
428        } else {
429            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
430                .child(urlImgOther).getLastPathSegment();
431        }
431    }
432
433    private void updateImgOtherImage() {
434        if (urlImgOther == null) {
435            updateImgOtherImage();
436        } else {
437            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
438                .child(urlImgOther).getLastPathSegment();
439        }
439    }
440
441    private void updateImgOtherImage() {
442        if (urlImgOther == null) {
443            updateImgOtherImage();
444        } else {
445            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
446                .child(urlImgOther).getLastPathSegment();
447        }
447    }
448
449    private void updateImgOtherImage() {
450        if (urlImgOther == null) {
451            updateImgOtherImage();
452        } else {
453            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
454                .child(urlImgOther).getLastPathSegment();
455        }
455    }
456
457    private void updateImgOtherImage() {
458        if (urlImgOther == null) {
459            updateImgOtherImage();
460        } else {
461            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
462                .child(urlImgOther).getLastPathSegment();
463        }
463    }
464
465    private void updateImgOtherImage() {
466        if (urlImgOther == null) {
467            updateImgOtherImage();
468        } else {
469            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
470                .child(urlImgOther).getLastPathSegment();
471        }
471    }
472
473    private void updateImgOtherImage() {
474        if (urlImgOther == null) {
475            updateImgOtherImage();
476        } else {
477            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
478                .child(urlImgOther).getLastPathSegment();
479        }
479    }
480
481    private void updateImgOtherImage() {
482        if (urlImgOther == null) {
483            updateImgOtherImage();
484        } else {
485            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
486                .child(urlImgOther).getLastPathSegment();
487        }
487    }
488
489    private void updateImgOtherImage() {
490        if (urlImgOther == null) {
491            updateImgOtherImage();
492        } else {
493            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
494                .child(urlImgOther).getLastPathSegment();
495        }
495    }
496
497    private void updateImgOtherImage() {
498        if (urlImgOther == null) {
499            updateImgOtherImage();
500        } else {
501            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
502                .child(urlImgOther).getLastPathSegment();
503       }
503    }
504
505    private void updateImgOtherImage() {
506        if (urlImgOther == null) {
507            updateImgOtherImage();
508        } else {
509            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
510                .child(urlImgOther).getLastPathSegment();
511       }
511    }
512
513    private void updateImgOtherImage() {
514        if (urlImgOther == null) {
515            updateImgOtherImage();
516        } else {
517            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
518                .child(urlImgOther).getLastPathSegment();
519       }
519    }
520
521    private void updateImgOtherImage() {
522        if (urlImgOther == null) {
523            updateImgOtherImage();
524        } else {
525            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
526                .child(urlImgOther).getLastPathSegment();
527       }
527    }
528
529    private void updateImgOtherImage() {
530        if (urlImgOther == null) {
531            updateImgOtherImage();
532        } else {
533            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
534                .child(urlImgOther).getLastPathSegment();
535       }
535    }
536
537    private void updateImgOtherImage() {
538        if (urlImgOther == null) {
539            updateImgOtherImage();
540        } else {
541            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
542                .child(urlImgOther).getLastPathSegment();
543       }
543    }
544
545    private void updateImgOtherImage() {
546        if (urlImgOther == null) {
547            updateImgOtherImage();
548        } else {
549            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
550                .child(urlImgOther).getLastPathSegment();
551       }
551    }
552
553    private void updateImgOtherImage() {
554        if (urlImgOther == null) {
555            updateImgOtherImage();
556        } else {
557            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
558                .child(urlImgOther).getLastPathSegment();
559       }
559    }
560
561    private void updateImgOtherImage() {
562        if (urlImgOther == null) {
563            updateImgOtherImage();
564        } else {
565            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
566                .child(urlImgOther).getLastPathSegment();
567       }
567    }
568
569    private void updateImgOtherImage() {
570        if (urlImgOther == null) {
571            updateImgOtherImage();
572        } else {
573            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
574                .child(urlImgOther).getLastPathSegment();
575       }
575    }
576
577    private void updateImgOtherImage() {
578        if (urlImgOther == null) {
579            updateImgOtherImage();
580        } else {
581            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
582                .child(urlImgOther).getLastPathSegment();
583       }
583    }
584
585    private void updateImgOtherImage() {
586        if (urlImgOther == null) {
587            updateImgOtherImage();
588        } else {
589            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
590                .child(urlImgOther).getLastPathSegment();
591       }
591    }
592
593    private void updateImgOtherImage() {
594        if (urlImgOther == null) {
595            updateImgOtherImage();
596        } else {
597            StorageReference storageReference = FirebaseStorage.getInstance().getReference().child("pathString: Save Images")
598                .child(urlImgOther).getLastPathSegment();
599       }
599    }
599    
```

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` package and extends `AppCompatActivity`. It handles updating a product in a database. The code includes methods for calculating new price values, creating a new product object, setting its details, and updating it in the database. It also includes logic for displaying success or failure messages to the user.

```
public class UpdateDrinkAdminActivity extends AppCompatActivity {
    private void updateDrinksData() {
        double newPriceValue = priceValue - (priceValue * (saleValue / 100));
        String priceNewFormatted = decimalFormat.format(newPriceValue);
        Product product = new Product(name, desc, price, priceNewFormatted, sale, imgUrl, imgSlider, isPopular, productType, imgOther);
        // Tao doi tuong Product moi voi thong tin cap nhat
        Product product = new Product();
        product.setName(name);
        product.setDesc(desc);
        product.setPrice(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() {
            @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(@NotNull Exception e) {
                Toast.makeText(getApplicationContext(), UpdateDrinkAdminActivity.this, 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**

```
 41     public class UpdateDessertAdminActivity extends AppCompatActivity {
 42         protected void onCreate(Bundle savedInstanceState) {
 43
 44             // ActivityResultLauncher<Intent> activityResultLauncher3 = registerForActivityResult(
 45             //     new ActivityResultContracts.StartActivityForResult(),
 46             //     new ActivityResultCallback<ActivityResult>() {
 47             //         @Override
 48             //         public void onActivityResult(ActivityResult result) {
 49             //             if (result.getResultCode() == Activity.RESULT_OK) {
 50             //                 Intent data = result.getData();
 51             //                 urlImgOther = data.getData();
 52             //                 update_imgOther.setImageURI(urlImgOther);
 53             //             } else {
 54             //                 Toast.makeText(getApplicationContext(), "Không có hình ảnh nào được chọn", Toast.LENGTH_SHORT).show();
 55             //             }
 56             //         }
 57             //     });
 58
 59             // Nhận sản phẩm từ Detail gửi về
 60             Bundle bundle = getIntent().getExtras();
 61             if (bundle != null) {
 62                 Glide.with(activity).load(bundle.getString(key: "imgUrl")).into(update_imgUrl);
 63                 Glide.with(activity).load(bundle.getString(key: "imgSlider")).into(update_imgSlider);
 64                 Glide.with(activity).load(bundle.getString(key: "imgOther")).into(update_imgOther);
 65                 update_edit_name.setText(bundle.getString(key: "name"));
 66                 update_edit_desc.setText(bundle.getString(key: "desc"));
 67                 update_edit_price.setText(bundle.getString(key: "price"));
 68                 update_edit_sale.setText(bundle.getString(key: "sale"));
 69                 // Khoá nút "Popular" và "Not Popular"
 70                 String popularStatus = bundle.getString(key: "popular");
 71                 if (popularStatus != null) {
 72                     if (popularStatus.equals("0")) {
 73                         update_rad_popular1.setChecked(true);
 74                     } else if (popularStatus.equals("1")) {
 75                         update_rad_popular2.setChecked(true);
 76                     }
 77                 }
 78                 String productType = bundle.getString(key: "productType");
 79                 if (productType != null) {
 80                     switch (productType) {
 81                         case "Food":
```

```
 41     public class UpdateDessertAdminActivity extends AppCompatActivity {
 42         protected void onCreate(Bundle savedInstanceState) {
 43             super.onCreate(savedInstanceState);
 44
 45             switch (productType) {
 46                 case "Món ăn":
 47                     update_raw_product_type_food.setChecked(true);
 48                     break;
 49
 50                 case "Bđ uống":
 51                     update_raw_product_type_drink.setChecked(true);
 52                     break;
 53
 54                 case "Tráng miệng":
 55                     update_raw_product_type_dessert.setChecked(true);
 56                     break;
 57
 58                 default:
 59                     // Xử lý trường hợp không tìm thấy productType nào phù hợp
 60                     break;
 61             }
 62         }
 63
 64
 65         key = bundle.getString("key");
 66         oldImgUrl = bundle.getString("imgurl");
 67         oldImgSlider = bundle.getString("imgslider");
 68         oldImgOther = bundle.getString("imgother");
 69
 70     }
 71
 72     databaseReference = FirebaseDatabase.getInstance().getReference("products");
 73
 74     update_imgUrl.setOnClickListener(new View.OnClickListener() {
 75         @Override
 76         public void onClick(View v) {
 77             Intent photoPicker = new Intent(Intent.ACTION_PICK);
 78             photoPicker.setType("image/*");
 79             activityResultLauncher.launch(photoPicker);
 80         }
 81     });
 82
 83     update_imgSlider.setOnClickListener(new View.OnClickListener() {
 84         @Override
 85         public void onClick(View v) {
 86             Intent photoPicker = new Intent(Intent.ACTION_PICK);
 87             photoPicker.setType("image/*");
 88             activityResultLauncher.launch(photoPicker);
 89         }
 90     });
 91
 92     update_imgOther.setOnClickListener(new View.OnClickListener() {
 93         @Override
 94         public void onClick(View v) {
 95             Intent photoPicker = new Intent(Intent.ACTION_PICK);
 96             photoPicker.setType("image/*");
 97             activityResultLauncher.launch(photoPicker);
 98         }
 99     });
 100 }
```

```

 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(",", ".").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.setProductId(productId);
        product.setPriceNew(priceNewFormatted);
        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)