

**MỤC LỤC**

[NHẬN XÉT CỦA GVHD 3](#_Toc44414216)

[LỜI CẢM ƠN 4](#_Toc44414217)

[PHẦN MỞ ĐẦU 5](#_Toc44414218)

[1. **MỤC TIÊU CỦA ĐỀ TÀI** 6](#_Toc44414219)

[**2.** **CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU** 7](#_Toc44414220)

[**2.1 Đối tượng nghiên cứu** 7](#_Toc44414221)

[**2.2 Phạm vi nghiên cứu của đề này hướng đến** 7](#_Toc44414222)

[**3.** **KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC** 8](#_Toc44414223)

[PHẦN NỘI DUNG 9](#_Toc44414224)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 10](#_Toc44414225)

[**1.**  **TÌM HIỂU ANDROID STUDIO** 10](#_Toc44414226)

[**1.1. Khái niệm** 10](#_Toc44414227)

[**1.2. Các phiên bản của Android** 10](#_Toc44414228)

[**1.3. Ưu điểm và nhược điểm** 11](#_Toc44414229)

[**1.4. Bảo mật** 12](#_Toc44414230)

[**2.** **ỨNG DỤNG TRÊN ANDROID STUDIO TRONG PHÁT TRIỂN APP** 12](#_Toc44414231)

[**3.** **CẤU TRÚC CỦA ỨNG DỤNG ANDROID** 12](#_Toc44414232)

[**3.1. Cấu trúc thư mục** 12](#_Toc44414233)

[**3.2. Cấu trúc giao diện chung** 12](#_Toc44414234)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU 14](#_Toc44414235)

[**1.** **XÁC ĐỊNH YÊU CẦU** 14](#_Toc44414236)

[**1.1. Yêu cầu chức năng** 14](#_Toc44414237)

[**1.2. Yêu cầu phi chức năng** 14](#_Toc44414238)

[**2.** **MÔ HÌNH HÓA YÊU CẦU** 15](#_Toc44414239)

[**2.1. Đặc tả chức năng của foody** 15](#_Toc44414240)

[**2.2. Lược đồ usecase** 22](#_Toc44414241)

[CHƯƠNG 3: THIẾT KẾ PHẦN MỀM 23](#_Toc44414242)

[**1.** **THIẾT KẾ HỆ THỐNG** 23](#_Toc44414243)

[**Lược đồ tuần tự** 23](#_Toc44414244)

[**Thiết kế Database** 25](#_Toc44414245)

[**Đặc tả Database** 25](#_Toc44414246)

[**2.** **THIẾT KẾ GIAO DIỆN** 26](#_Toc44414247)

[**2.1. Màn hình chính** 27](#_Toc44414248)

[**2.2. Đặc tả màn hình chính** 28](#_Toc44414249)

[**2.3. Màn hình chọn tỉnh, thành phố** 29](#_Toc44414250)

[**2.4. Đặc tả màn hình chọn tỉnh, thành phố** 30](#_Toc44414251)

[**2.5. Màn hình tìm kiếm** 31](#_Toc44414252)

[**2.6. Đặc tả màn hình tìm kiếm** 32](#_Toc44414253)

[**2.7. Màn hình chi tiết quán ăn.** 33](#_Toc44414254)

[**2.8. Màn hình đặc tả chi tiết quán ăn.** 34](#_Toc44414255)

[**2.9. Màn hình menu quán ăn** 35](#_Toc44414256)

[**2.10. Đặc tả màn hình menu quán ăn.** 36](#_Toc44414257)

[**2.11. Dialog thêm hoặc cập nhật wifi** 37](#_Toc44414258)

[**2.12. Đặc tả dialog thêm hoặc cập nhật wifi** 39](#_Toc44414259)

[CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ 40](#_Toc44414260)

[**1.** **CÀI ĐẶT** 40](#_Toc44414261)

[**1.1 Các hằng số value** 40](#_Toc44414262)

[**1.2. Các biến trong class MainActivity** 40](#_Toc44414263)

[**1.3. Các hàm trong class MainActivity** 41](#_Toc44414264)

[**1.4. Các biến trong class SearchResultActivity** 44](#_Toc44414265)

[**1.5. Các hàm trong class SearchResultActivity** 46](#_Toc44414266)

[**1.6. Các biến trong class ChooseProvinceActivity** 48](#_Toc44414267)

[**1.7. Các hàm trong class ChooseProvinceActivity** 49](#_Toc44414268)

[**1.8. Các biến trong class DetailsActivity** 50](#_Toc44414269)

[**1.9. Các hàm trong class DetailsActivity** 52](#_Toc44414270)

[**1.10. Các biến trong class MenuActivity** 55](#_Toc44414271)

[**1.11. Các hàm trong class MenuActivity** 56](#_Toc44414272)

[**1.12. Các view model** 57](#_Toc44414273)

[**1.13. WifiAdapter** 59](#_Toc44414274)

[**1.14. ProvinceViewAdapter** 60](#_Toc44414275)

[**1.15 FoodViewAdapter** 61](#_Toc44414276)

[**1.16. FoodPlaceFullViewAdapter** 62](#_Toc44414277)

[**1.17. FoodPlaceCardViewAdapter** 63](#_Toc44414278)

[**1.18. ExpandMenuAdapter** 64](#_Toc44414279)

[**2.** **KIỂM THỬ** 66](#_Toc44414280)

[**2.1. Bảng mô tả quá trình test** 66](#_Toc44414281)

[**2.2. Bảng mô tả Data quá trình test** 68](#_Toc44414282)

[PHẦN KẾT LUẬN 71](#_Toc44414283)

[**1.** **KẾT QUẢ ĐẠT ĐƯỢC** 72](#_Toc44414284)

[**2.** **ƯU ĐIỂM** 72](#_Toc44414285)

[**3.** **NHƯỢC ĐIỂM** 72](#_Toc44414286)

[BẢNG PHÂN CÔNG CÔNG VIỆC 73](#_Toc44414287)

# NHẬN XÉT CỦA GVHD

…….., Ngày ….. Tháng ….. Năm 2019

Cô Trương Thị Ngọc Phượng

# LỜI CẢM ƠN

Để hoàn thành project này, ngoài sự cố gắng và làm việc của cả nhóm, nhóm chúng em còn nhận được rất nhiều sự giúp đỡ từ các cá nhân và nhóm khác.

Qua đó, nhóm chúng em xin gửi lời cảm ơn chân thành nhất đến các bạn và đặc biệt là Cô Trương Thị Ngọc Phượng đã tận tình hướng dẫn và dành thời gian quý báu của mình để giúp đỡ tụi em để hoàn thành project này.

Vì thời gian có hạn và kiến thức còn nhiều hạn chế nên báo cáo này của chúng em chắc chắn không tránh khỏi thiếu sót, chúng em rất mong nhận được sự chỉ bảo của cô để bài báo cáo này hoàn thiện hơn. Chúng em xin chân thành cảm ơn!

# PHẦN MỞ ĐẦU

## 1. **MỤC TIÊU CỦA ĐỀ TÀI**

Tìm hiểu và xây dựng UI Design và Event Handing cho app Foody trên nền tảng lập trình di động android.

Tìm hiểu về môi trường và công nghệ

Cài đặt môi trường để phát triển ứng dụng Android Studio. Cụ thể :

- JDK

- Android SDK

- Về dùng emulator (thiết bị giả lặp), lựa chọn: Emulator của chính Android Studio, Genymotion. Và sử dụng thiết bị thực tế trên các điện thoại di động.

Xây dựng app Foody có các chức năng cơ bản:

* Chọn tỉnh/ thành phố muốn xem quán ăn
* Xem quán ăn trong tỉnh/thành phố
* Tìm kiếm quán ăn theo ba tiêu chí: Đúng nhất, phổ biến, gần tôi
* Xem chi tiết quán ăn
* Xem menu quán ăn
* Xem món có trong menu quán ăn
* Tìm vị trí quán ăn
* Xem khoảng cách từ quán ăn đến người dùng, chỉ đường tới quán ăn
* Tra cứu vị trí quán ăn trên map
* Xem, thêm, chia sẻ wifi quán ăn

## **2. CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU**

### **2.1 Đối tượng nghiên cứu**

Với đề tài này nhóm chúng em tập trung vào các nội dung:

* Cách cài đặt các chương trình để có thể chạy Android.
* Tìm hiểu về Activity trong Android.
* Cách tạo giao diện trong Android.
* Xây dựng ứng dụng Foody đơn giản trên Android.

### **2.2 Phạm vi nghiên cứu của đề này hướng đến**

Công cụ hỗ trợ lập trình tạo ứng dựng trên android Studio Các tính năng của Foody trên các thiết bị sử dụng hệ điều hành Android hiện nay.

## **3. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC**

Nghiên cứu được các thành phần chính của công nghệ lập trình Android Studio như : môi trường cài đặt, cách sử dụng, các biến, các hàm chứa giao diện, chức năng. Ngoài ra, còn nghiên cứu các xử lý chức năng cơ bản của của app Foody trên cái thiết bị sử dụng hệ điều hành android hiện nay.

# PHẦN NỘI DUNG

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## **1. TÌM HIỂU ANDROID STUDIO**

### **1.1. Khái niệm**

Là một phầm mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet...

Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập AVD để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

### **1.2. Các phiên bản của Android**

Kỉ nguyên Android chính thức bắt đầu ngày 22/10/2008 khi T – Mobile ra mắt tại thị trường Mỹ. Phiên bản này thiếu rất nhiều tính năng chẳng hạn như bàn phím màn hình (on – screen keyboard), khả năng cảm ứng đa điểm, các ứng dụng thương mại App Market.[1]

|  |  |  |
| --- | --- | --- |
| **Phiên bản** | **Tên** | **Ngày phát hành** |
| Android 1.5 | Cupcake | 27/4/2009 |
| Android 1.6 | Donut | 15/9/2009 |
| Android 2.0 - 2.1 | Eclair | 26/9/2009 (phát hành lần đầu) |
| Android 2.2 - 2.2.3 | Froyo | 20/5/2010 (phát hành lần đầu) |
| Android 2.3 - 2.3.7 | Gingerbread | 6/12/2010 (phát hành lần đầu) |
| Android 3.0 - 3.2.6 | Honeycomb | 22/2/2011 (phát hành lần đầu) |
| Android 4.0 - 4.0.4 | Ice Cream Sandwich | 18/10/2011 (phát hành lần đầu) |
| Android 4.1 - 4.3.1 | Jelly Bean | 9/7/2012 (phát hành lần đầu) |
| Android 4.4 - 4.4.4 | KitKat | 31/10/2013 (phát hành lần đầu) |
| Android 5.0 - 5.1.1 | Lollipop | 12/11/2014 (phát hành lần đầu) |
| Android 6.0 - 6.0.1 | Marshmallow | 5/10/2015 (phát hành lần đầu) |
| Android 7.0 - 7.1.2 | Nougat | 22/8/2016 (phát hành lần đầu) |
| Android 8.0 - 8.1 | Oreo | 21/8/2017 (phát hành lần đầu) |

### **1.3. Ưu điểm và nhược điểm**

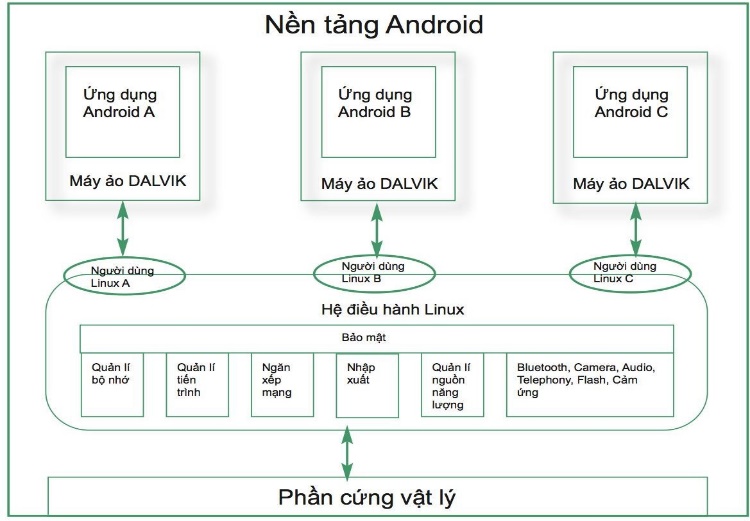
**Ưu điểm**

* An ninh
* Chất lượng
* Tính tuỳ biến
* Chi phí thấp
* Công nghệ mở
* Miễn phí cho sử dụng thương mại
* Phát triển và quảng cáo ứng dụng dễ dàng.
* Kho ứng dụng khổng lồ
* Hỗ trợ đa nhiệm.

**Nhược điểm**

* Bảo mật
* Đột phá ý tưởng

### **1.4. Bảo mật**



## **2. ỨNG DỤNG TRÊN ANDROID STUDIO TRONG PHÁT TRIỂN APP**

Một điện thoại Android sẽ đi kèm với một vài ứng dụng được cài đặt sẵn, bao gồm: Gmail, SMS, Google Map, Chat, Trình đa phương tiện, Android Market

**=> Tất cả các ứng dụng đều viết bằng Java, sử dụng Android SDK.**

## **3. CẤU TRÚC CỦA ỨNG DỤNG ANDROID**

### **3.1. Cấu trúc thư mục**

Một dự án khi được tạo xong sẽ gồm các thư mục sau: Src (java)

Res

Android Manifest Android Version

Những vấn đề liên quan:

Min và Target SDK

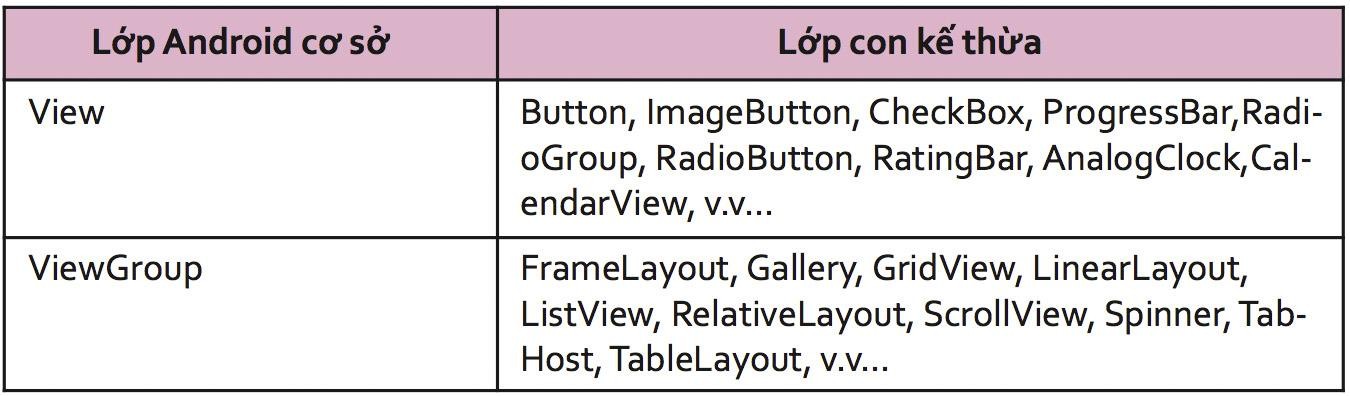
Các phiên bản thay thế resource

Định dạnh của từng loại file trong ứng dụng.

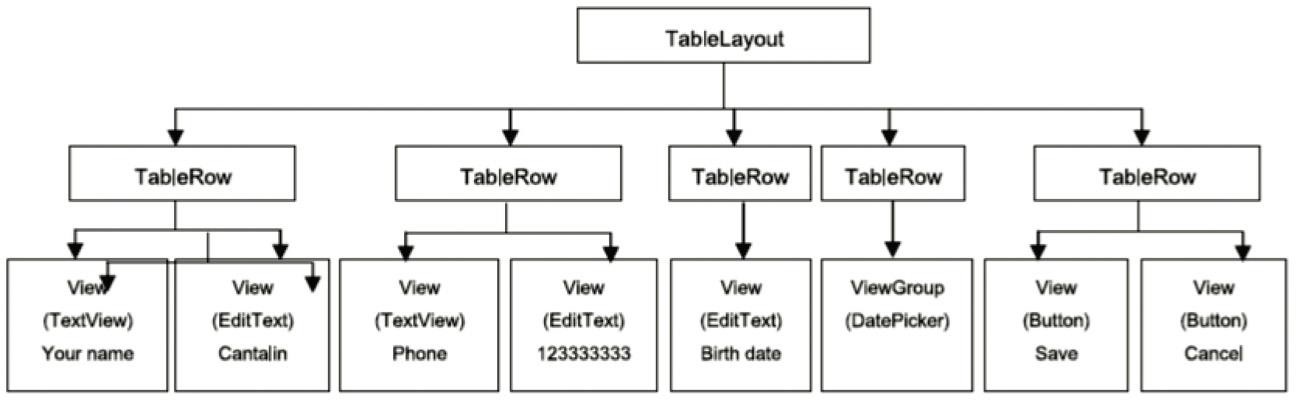
### **3.2. Cấu trúc giao diện chung**

- Giao diện được lưu trữ dưới dạng các tập tin XML trong thư mục res/layout

- Sử dụng hai lớp Java: View và ViewGroup. Các lớp con kế thừa từ View gọi là Widget, ví dụ: Button, TextView, v.v… Còn các lớp kế thừa từ ViewGroup gọi là Layout, ví dụ: LinearLayout, RelativeLayout, v.v…



Cấu trúc giao diện như sau:



***Hình 1.*** *Cấu trúc giao diện*

# CHƯƠNG 2: PHÂN TÍCH YÊU CẦU

## **1. XÁC ĐỊNH YÊU CẦU**

### **1.1. Yêu cầu chức năng**

* Tìm kiếm quán ăn đa dạng phù hợp với yêu cầu của người dùng
* Keyword dùng để tìm quán ăn không tự động xuống hàng khi độ dài keyword vượt quá độ dài Edit Text.
* Hỗ trợ Load More và Delayed Search để tối ưu hóa việc tải dữ liệu.
* Yêu cầu quyền truy cập từ người dùng khi cần truy cập vào vị trí hay mở ứng dúng khác từ máy của người dùng.
* Xoay màn hình tốt ở mọi định dạng.
* Hiển thị chi tiết quán ăn một cách tổng quát nhất để người dùng có cái nhìn toàn diện nhất về quán ăn.
* Cho phép người dùng xem vị trí của quán ăn trên bản đồ.
* Cho phép người dùng xem, thêm và chia sẻ wifi
* Hiển thị menu theo dạng key – value giúp người dùng có cái nhìn tổng quan nhất về menu của quán ăn
* Sử dụng Database online để tối ưu dung lượng ứng dụng

### **1.2. Yêu cầu phi chức năng**

- Bất kỳ sự tìm kiếm nào cũng không vượt quá 5 giây để tải xong.

- Giao diện thân thiện, màu sắc tươi tắn, hài hòa, phong chữ vừa vặn dễ sử dụng với người dùng. Thích hợp với mọi máy di động trên nhiều hãng khác nhau không bị lỗi.

- Dễ dàng mở rộng các chức năng của phần mềm đối với nhà phát triển.

- Đảm bảo duy trì hoạt động lâu dài, hiệu quả.

## **2. MÔ HÌNH HÓA YÊU CẦU**

### **2.1. Đặc tả chức năng của foody**

#### 2.1.1. Đặc tả chức năng Tìm kiếm theo độ phổ biến

|  |  |
| --- | --- |
| **Use Case Tìm kiếm theo độ phổ biến** | |
| **Mô tả** | Mỗi khi user muốn tìm địa điểm của quán ăn |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Không có |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Nhập key word vào ô tìm kiếm 3. Ấn tìm kiếm 4. Sau đó chọn “Phổ biến” 5. Các quán ăn phù hợp sẽ xuất hiện |

*Bảng 1. Bảng mô tả dữ liệu của Use Case Tìm kiếm theo độ phổ biến*

#### 2.1.2. Đặc tả chức năng Tìm kiếm theo khoảng cách

|  |  |
| --- | --- |
| **Use Case Tìm kiếm theo khoảng cách** | |
| **Mô tả** | Mỗi khi user muốn tìm địa điểm của quán ăn |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Không có |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Nhập key word vào ô tìm kiếm 3. Ấn tìm kiếm 4. Sau đó chọn “Gần tôi” 5. Các quán ăn phù hợp sẽ xuất hiện |

*Bảng 2. Bảng mô tả dữ liệu của Use Case Tìm kiếm theo khoảng cách*

#### 2.1.3. Đặc tả chức năng Tìm kiếm theo độ chính xác

|  |  |
| --- | --- |
| **Use Case Tìm kiếm theo độ chính xác** | |
| **Mô tả** | Mỗi khi user muốn tìm địa điểm của quán ăn |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Không có |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Nhập key word vào ô tìm kiếm 3. Ấn tìm kiếm 4. Sau đó chọn “Đúng nhất” 5. Các quán ăn phù hợp sẽ xuất hiện |

*Bảng 3. Bảng mô tả dữ liệu của Use Case Tìm kiếm theo độ chính xác*

#### 2.1.4. Đặc tả chức năng Xem thông tin quán

|  |  |
| --- | --- |
| **Use Case Xem thông tin quán** | |
| **Mô tả** | Mỗi khi user muốn xem các thông tin của quán |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Quán ăn phải có trong CSDL |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Nhấn vào hình ảnh các quán ăn được hiển thị hoặc các kết quả tìm kiếm được 3. Các thông tin của quán ăn sẽ xuất hiện |

*Bảng 4. Bảng mô tả dữ liệu của Use Case Xem thông tin quán*

#### 2.1.5. Đặc tả chức năng Thêm wifi

|  |  |
| --- | --- |
| **Use Case Thêm wifi** | |
| **Mô tả** | Mỗi khi user muốn kết nối wifi |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Không có |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Nhấn vào hình ảnh các quán ăn bất kỳ được hiển thị 3. Nhấn thêm wifi 4. Nếu đã kết nối wifi thì tên wifi sẽ được hiển thị 5. Nếu chưa kết nối user tiến hành nhập tên wifi và password 6. Nếu thành công hiển thị thông báo: “Kết nối thành công”, nếu không hiển thị thông báo “Kết nối thất bại” |

*Bảng 5. Bảng mô tả dữ liệu của Use Case Thêm wifi*

#### 2.1.6. Đặc tả chức năng Xem menu quán ăn

|  |  |
| --- | --- |
| **Use Case Xem menu quán ăn** | |
| **Mô tả** | Mỗi khi user muốn xem menu quán ăn |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Quán ăn phải có trong cơ sở dữ liệu |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Nhấn hình ảnh quán ăn muốn xem menu 3. Nhấn vào thực đơn 4. Xem các loại thức ăn và thức ăn thuộc loại đó có trong menu |

*Bảng 6. Bảng mô tả dữ liệu của Use Case Xem menu quán ăn*

#### 2.1.7. Đặc tả chức năng Xem vị trí quán ăn trên bản đồ

|  |  |
| --- | --- |
| **Use Case Xem vị trí quán ăn** | |
| **Mô tả** | Mỗi khi user muốn xem vị trí của quán ăn |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Quán ăn phải có trong cơ sở dữ liệu |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Cấp quyền truy cập vị trí cho ứng dụng 3. Nhấn vào hỉnh ảnh bản đồ trên màn hình 4. Hỉnh ảnh vị trí quán ăn sẽ được hiển thị trên google map |

*Bảng 7. Bảng mô tả dữ liệu của Use Case Xem vị trí quán ăn*

#### 2.1.8 Đặc tả chức năng Thêm hoặc Cập nhật Wifi

|  |  |
| --- | --- |
| **Use Case Thêm hoặc Cập nhật Wifi** | |
| **Mô tả** | Mỗi khi user muốn thêm hoặc cập nhật wifi vào một quán ăn nào đó |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Không có |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Chọn quán ăn muốn thêm hoặc cập nhật wifi 3. Nhập WifiName, Password 4. Sau đó nhấn “Cập nhật Wifi” 5. Nếu chưa có wifi sẽ thêm vào CSDL ngược lại sẽ cập nhật lại wifi |

*Bảng 8. Bảng mô tả dữ liệu của Use Case Thêm hoặc Cập nhật wifi*

#### 2.1.9 Đặc tả chức năng Xem Wifi

|  |  |
| --- | --- |
| **Use Case Xem Wifi** | |
| **Mô tả** | Mỗi khi user muốn xem Wifi của quán |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Không có |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Chọn quán ăn muốn xem Wifi 3. Thông tin wifi sẽ được hiển thị cùng với một số thông tin liên quan |

*Bảng 9. Bảng mô tả dữ liệu của Use Case Xem Wifi*

**2.1.10 Đặc tả chức năng Xem khoảng cách**

|  |  |
| --- | --- |
| **Use Case Xem khoảng cách từ quán ăn đến người dùng** | |
| **Mô tả** | Mỗi khi user muốn xem khoảng cách từ quán ăn đến vị trí hiện tại của bản thân |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | User phải cấp quyền truy cập vị trí |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Chọn quán ăn muốn xem khoảng cách 3. Khoảng cách sẽ hiển thị cùng với một số thông tin liên quan |

*Bảng 10. Bảng mô tả dữ liệu của Use Case Xem khoảng cách*

**2.1.11 Đặc tả chức năng Chỉ đường tới quán ăn**

|  |  |
| --- | --- |
| **Use Case Chỉ đường tới quán ăn** | |
| **Mô tả** | Mỗi khi user muốn xem hướng dẫn đường đi tới quán ăn |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | User phải cấp quyền truy cập vị trí |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Chọn quán ăn muốn xem hướng dẫn 3. Nhấn vào hỉnh ảnh google map 4. Nhấn “Direction” để xem đường tới quán ăn |

*Bảng 11. Bảng mô tả dữ liệu của Use Case Chỉ đường tới quán ăn*

**2.1.12 Đặc tả chức năng Xem quán ăn theo tỉnh thành phố**

|  |  |
| --- | --- |
| **Use Case Xem quán ăn theo tỉnh/thành phố** | |
| **Mô tả** | Mỗi khi user muốn xem quán ăn ở 1 tỉnh/thành phố nào đó |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Không có |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Chọn tên thành phố bất kỳ ở góc phải màn hình 3. Chọn thành phố muốn xem quán ăn 4. Nhấn Xong 5. Ứng dụng sẽ hiển thị các quán ăn phù hợp |

*Bảng 12. Bảng mô tả dữ liệu của Use Case Xem quán ăn theo tỉnh/thành phố*

**2.1.13 Đặc tả chức năng Tìm kiếm tỉnh thành phố**

|  |  |
| --- | --- |
| **Use Case Tìm kiếm tỉnh\thành phố** | |
| **Mô tả** | Mỗi khi user muốn tìm kiếm tỉnh/thành phố |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Không có |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Chọn tên thành phố bất kỳ ở góc phải màn hình 3. Nhập từ khóa để tìm kiếm tỉnh thành phố 4. Ứng dụng sẽ hiển thị các thành phố phù hợp |

*Bảng 13. Bảng mô tả dữ liệu của Use Case Tìm kiếm tỉnh/thành phố*

**2.1.14 Đặc tả chức năng Liên hệ với chủ quán**

|  |  |
| --- | --- |
| **Use Case Liên hệ với chủ quán** | |
| **Mô tả** | Mỗi khi user muốn liên hệ với chủ quán ăn |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | User phải cấp quyền truy cập vào Điện Thoại |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Chọn quán ăn muốn liên hệ 3. Cấp quyền truy cập vào điện thoại 4. Nhấn vào “Liên hệ” 5. Ứng dụng sẽ gọi đến số điện thoại của chủ quán ăn |

*Bảng 14. Bảng mô tả dữ liệu của Use Case Liên hệ với chủ quán*

**2.1.15 Đặc tả chức năng Xem món có trong menu quán ăn**

|  |  |
| --- | --- |
| **Use Case Xem món có trong menu quán ăn** | |
| **Mô tả** | Mỗi khi user muốn xem món ăn của quán |
| **Tác nhân kích hoạt** | User |
| **Tiền điều kiện** | Không có |
| **Các bước thực hiện** | 1. Mở ứng dụng 2. Chọn quán ăn muốn xem món ăn 3. Lướt xuống dưới để xem các món ăn có trong menu |

*Bảng 15. Bảng mô tả dữ liệu của Use Case Xem món có trong menu*

### **2.2. Lược đồ usecase**



# CHƯƠNG 3: THIẾT KẾ PHẦN MỀM

## **1. THIẾT KẾ HỆ THỐNG**

### **Lược đồ tuần tự**

****

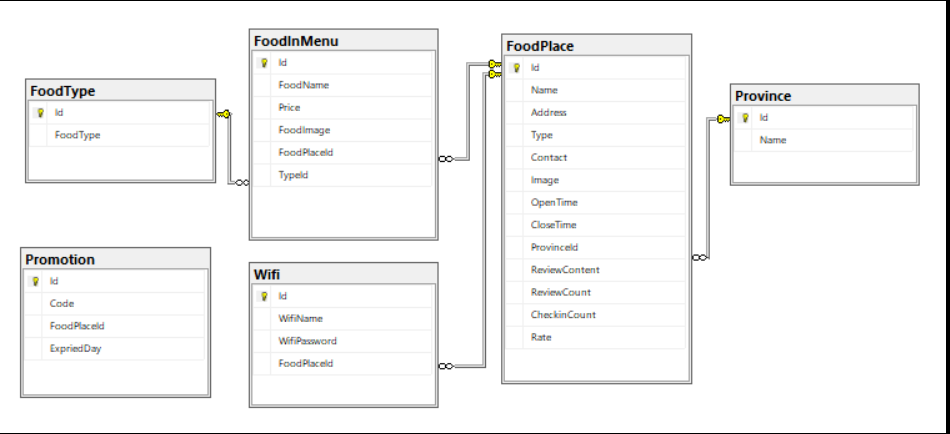
#### *Hình 2. Sơ đồ tuần tự Tìm kiếm theo keyword*



#### *Hình 3. Sơ đồ tuần tự Thêm wifi*

#### *Hình 4. Sơ đồ tuần tự Xem thông tin quán ăn*

### **Thiết kế Database**



### **Đặc tả Database**

Quán ăn (FoodPlace): Mỗi quán ăn có tên, địa chỉ cụ thể, có loại quán ăn (cơm, bún, phở), hình ảnh đại diện, mô tả quán ăn. Ngoài ra còn có giờ đóng/ mở cửa và Id để phân biệt.

Địa chỉ của quán ăn bao gồm số nhà, tên đường, tên phường, tên quận/ thành phố, tỉnh/ thành phố và có Id để phân biệt.

Mỗi tỉnh/ thành phố có Id để phân biệt và có thể thuộc nhiều địa chỉ quán ăn khác nhau.

Mỗi Quán ăn có nhiều món trong menu (FoodInMenu). Mỗi món có tên, giá, hình ảnh mô tả và thuộc một loại món nào đó (cơm chiên, cơm trắng, bún khô, bún nước) và Id để phân biệt.

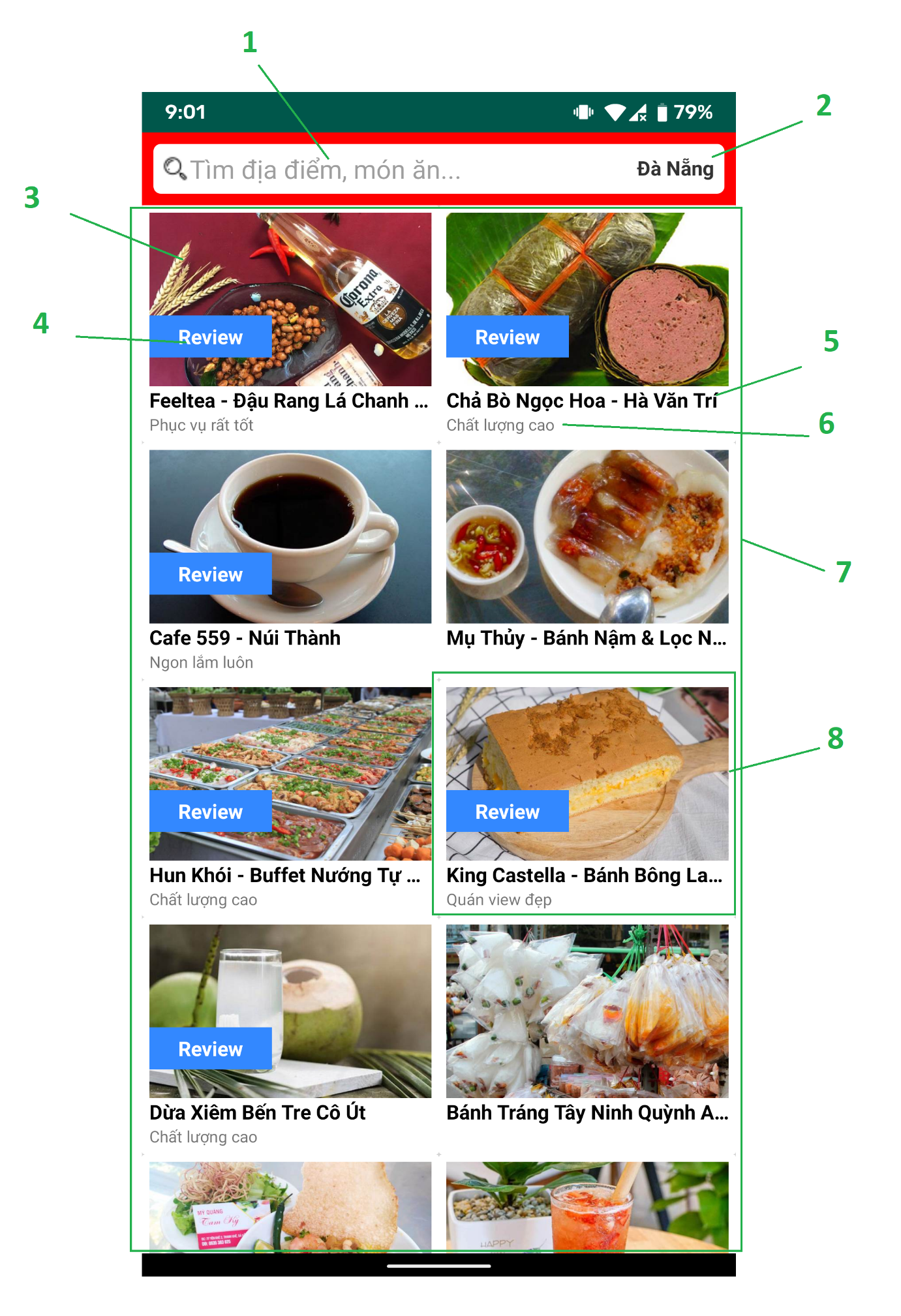
Các loại món ăn là xác định và có thể thuộc nhiều món ăn khác nhau, có Id để phân biệt với nhau.

Mỗi quán ăn có thể có các điểm truy cập wifi gần đó. Mỗi điểm truy cập wifi sẽ có tên và password để đăng nhập. Ngoài ra còn có Id để phân biệt.

Mỗi quán ăn có thể có các khuyến mãi. Mỗi khuyến mãi sẽ có mã khuyến mãi riêng và ngày hết hạn. Có Id để phân biệt.

## **2. THIẾT KẾ GIAO DIỆN**

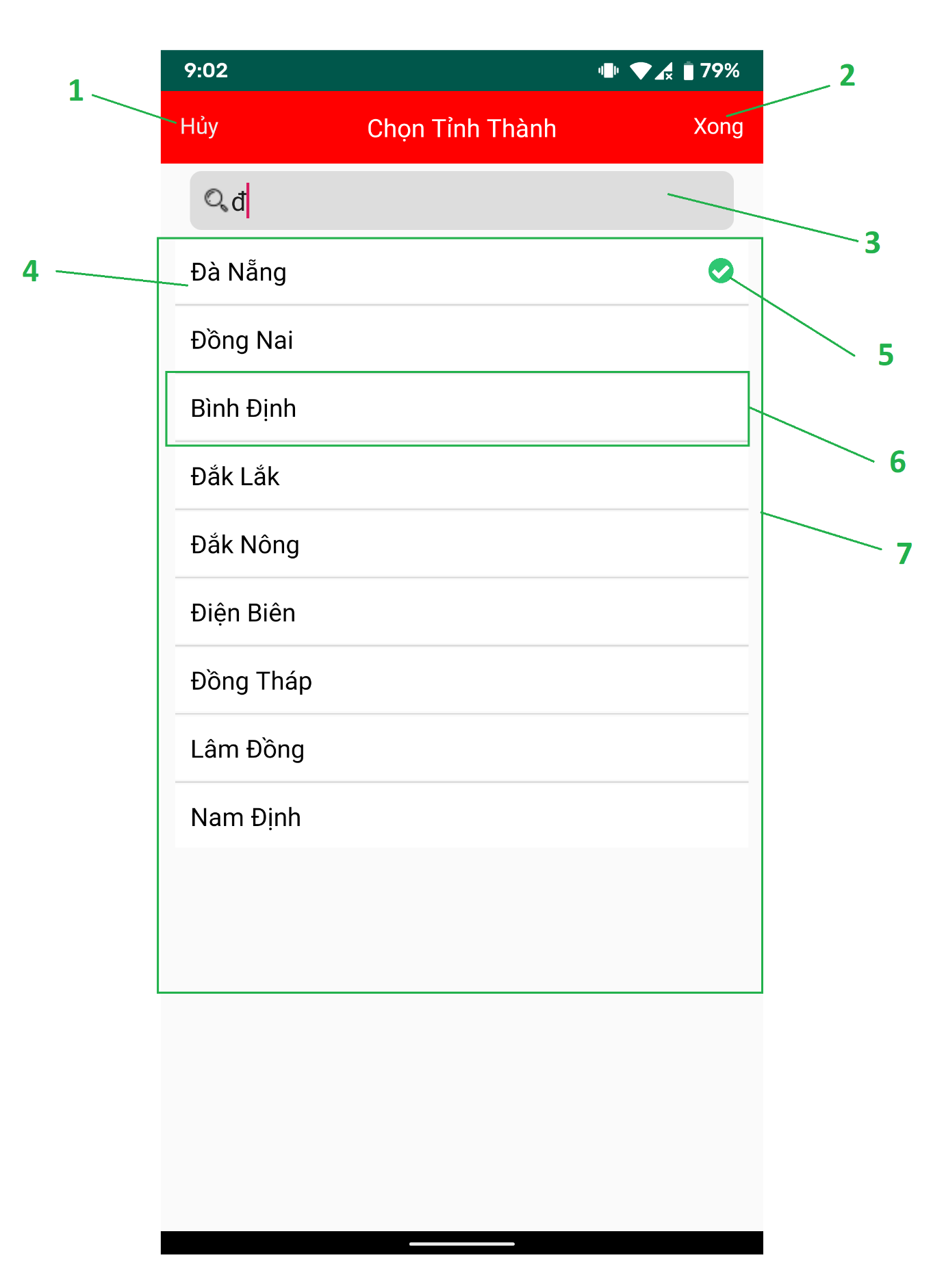
### **2.1. Màn hình chính**



### **2.2. Đặc tả màn hình chính**

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Loại** | **Mô tả** |
| 1 | EditText | Click vào để chuyển sang màn hình tìm kiếm để tìm kiếm địa điểm. |
| 2 | TextView | Click vào để chuyển sang màn hình chọn tỉnh/thành phố muốn đặt món. |
| 3 | ImageView | Dùng để hiển thị hình ảnh đại diện của quán ăn. |
| 4 | TextView | Hiện thị như tag để gây chú ý người dùng. |
| 5 | TextView | Hiển thị tên quán ăn. |
| 6 | TextView | Hiển thị các phản hồi review của quán. |
| 7 | RecyclerView | Dùng để hiển thị danh sách các quán ăn trong thành phố đã chọn. |
| 8 | CardView | Hiển thị chi tiết 1 món ăn trong danh sách. |

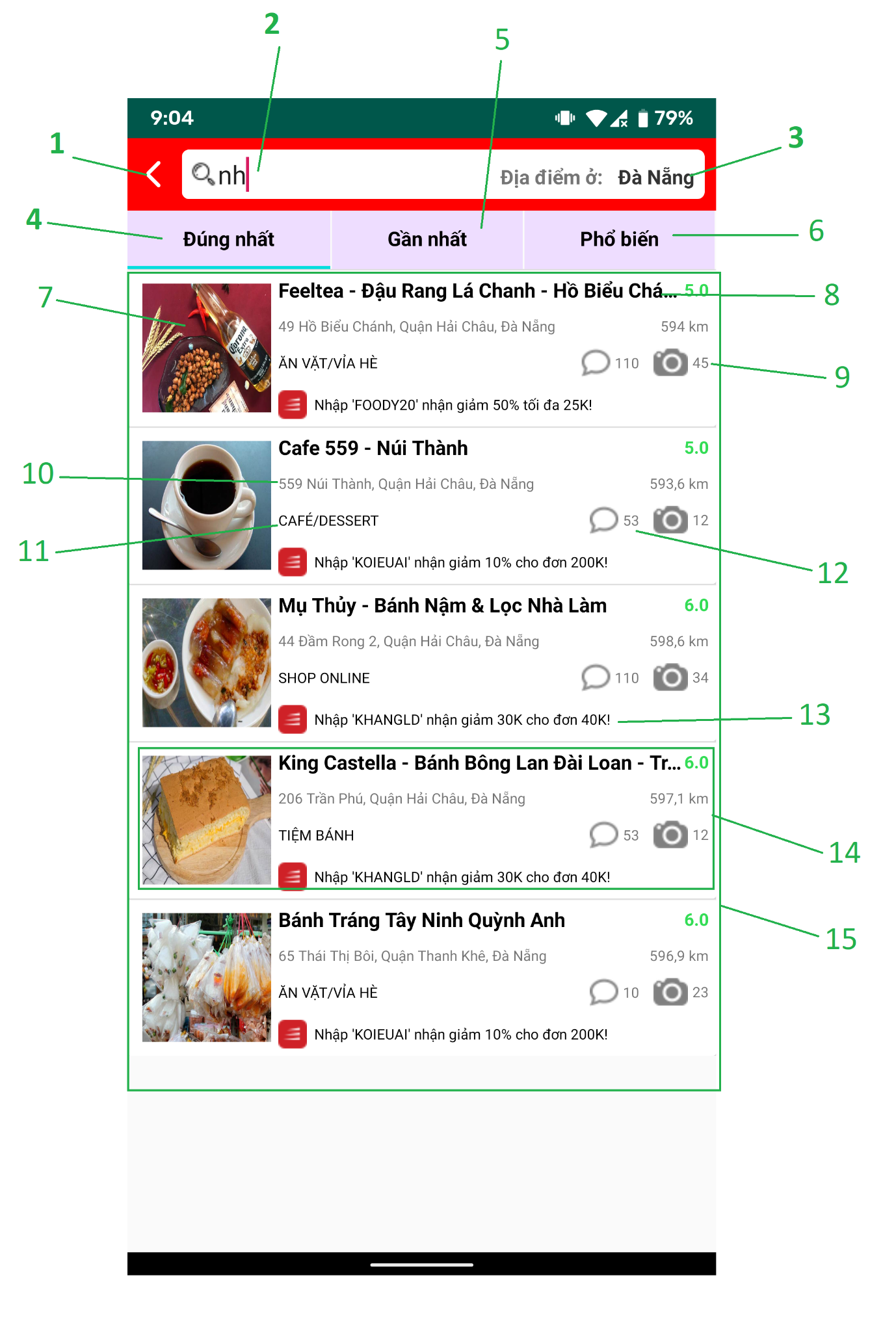
### **2.3. Màn hình chọn tỉnh, thành phố**



### **2.4. Đặc tả màn hình chọn tỉnh, thành phố**

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Loại** | **Mô tả** |
| 1 | TextView | Click vào để quay trở lại màn hình trước đó. |
| 2 | TextView | Click để lưu lại sự thay đổi thành phố. |
| 3 | EditText | Nhập từ khóa để tìm kiếm thành phố mong muốn. |
| 4 | TextView | Hiện thị tên của thành phố. |
| 5 | Drawable | Hiển thị nếu như là thành phố đang được chọn lúc đó. |
| 6 | Layout | Hiển thị thông tin mỗi thành phố trong danh sách, click vào để xem chi tiết của quán ăn đó. |
| 7 | ListView | Hiển thị danh sách các thành phố. |

### **2.5. Màn hình tìm kiếm**

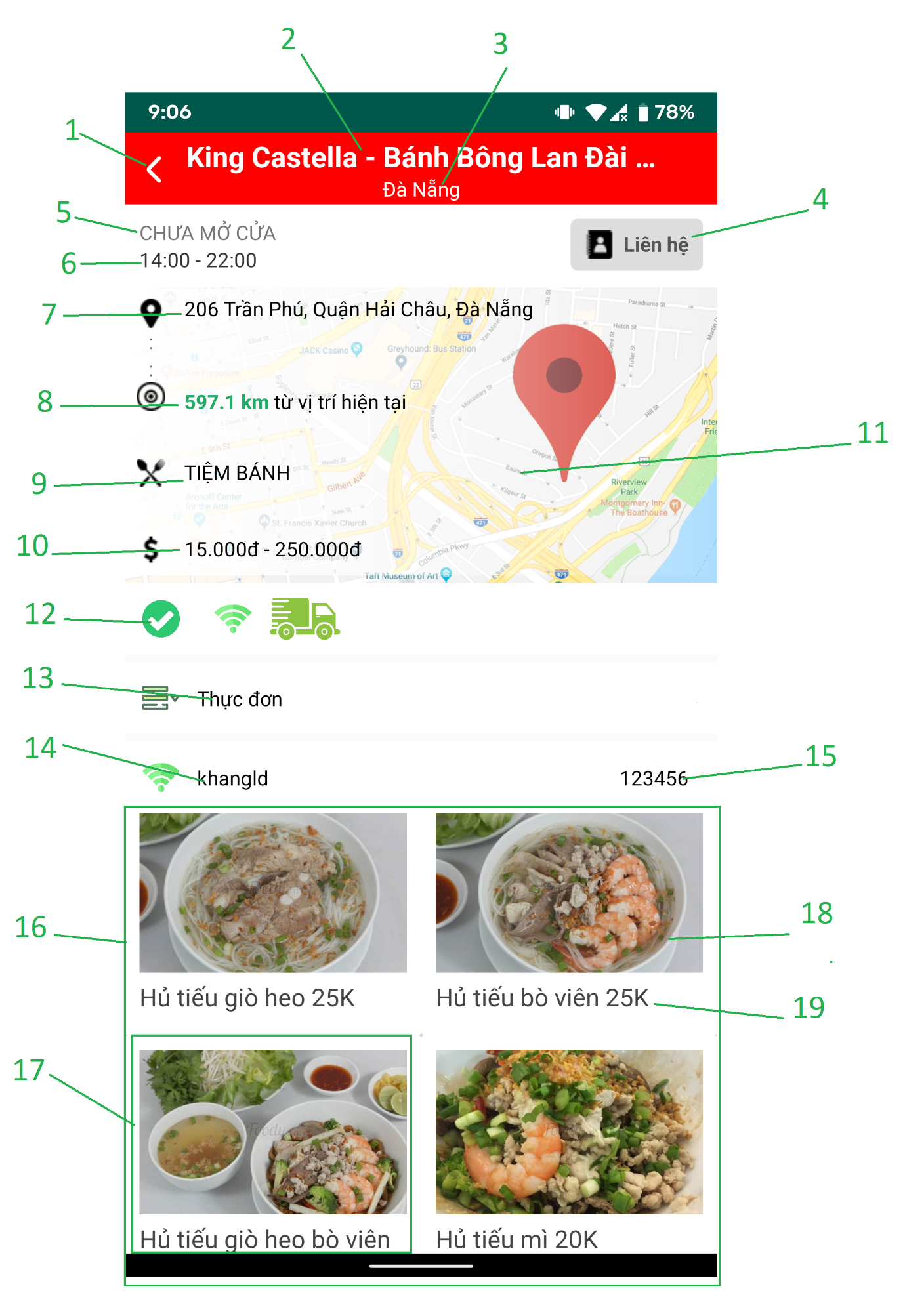


16

### **2.6. Đặc tả màn hình tìm kiếm**

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Loại** | **Mô tả** |
| 1 | ImageView | Click để quay trở lại màn hình trước đó. |
| 2 | EditText | Nhập vào kí tự đẻ tìm kiếm quán ăn , sau 0.6s sau khi ngừng nhập tự động load tìm kiếm hoặc có thể bấm tìm trong bàn phím. |
| 3 | TextView | Hiển thị thành phố hiện tại, bấm vào để thay đổi. |
| 4 | TextView | Click vào để tìm kiếm theo đúng nhất với kí tự tìm kiếm. |
| 5 | TextView | Click vào để tìm kiếm theo khoản cách gần nhất so với vị trí hiện tại. |
| 6 | TextView | Click vào để tìm kiếm theo độ phổ biến dựa trên lượt check in của khách. |
| 7 | ImageView | Hiển thị hình ảnh của quán ăn tìm kiếm được. |
| 8 | TextView | Hiển thị tên của quán ăn. |
| 9 | TextView | Hiển thị số lượt check in tại quán. |
| 10 | TextView | Hiển thị địa chỉ của quán. |
| 11 | TextView | Hiển thị loại hình quán. |
| 12 | TextView | Hiển thị số lượt nhận xét. |
| 13 | TextView | Hiển thị các mã giảm giá, khuyến mãi. |
| 14 | CardView | Dùng để hiển thị chi tiết 1 món ăn trong danh sách tìm kiếm, click vào để xem chi tiết quán ăn đó. |
| 15 | RecyclerView | Dùng để hiển thị danh sách các quán ăn theo tim kiếm. |
| 16 | TextView | Hiển thị khoảng cách của quán so với vị trí hiện tại. |

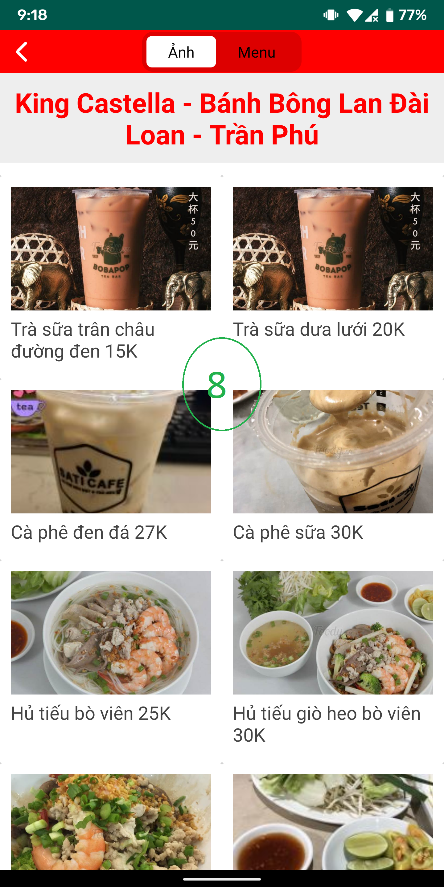
### **2.7. Màn hình chi tiết quán ăn.**



### **2.8. Màn hình đặc tả chi tiết quán ăn.**

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Loại** | **Mô tả** |
| 1 | ImageView | Click để quay lại màn hình trước |
| 2 | TextView | Hiển thị tên của quán ăn. |
| 3 | TextView | Hiển thị thành phố của quán ăn. |
| 4 | Button | Click để gọi đến sđt của quán ăn. |
| 5 | TextView | Hiển thị tình trạng đóng mở của của quán |
| 6 | TextView | Hiển thị giờ đóng mở cửa của quán |
| 7 | TextView | Hiển thị địa chỉ của quán |
| 8 | TextView | Hiển thị khoảng cách của quán |
| 9 | TextView | Hiển thị loai hình quán ăn. |
| 10 | TextView | Hiển thị các khoảng giá tiền trong quán. |
| 11 | ImageView | Click vào để xem địa chỉ quán trên Google Maps |
| 12 | ImageView | Hiển thị các trạng thái của quán |
| 13 | TextView | Click để chuyển sang màn hình menu của quán |
| 14 | TextView | Hiển thị tên wifi mới nhất đã có, click để hiển thị dialog thêm hoặc cập nhật wifi. |
| 15 | TextView | Hiển thị mật khẩu của wifi mới nhất đã có. |
| 16 | RecyclerView | Dùng để hiển thị danh sách món ăn. |
| 17 | CardView | Dùng để hiển thị món ăn trong dánh sách |
| 18 | ImageView | Hiển thị hình ảnh của món ăn |
| 19 | TextView | Hiển thị tên món ăn |

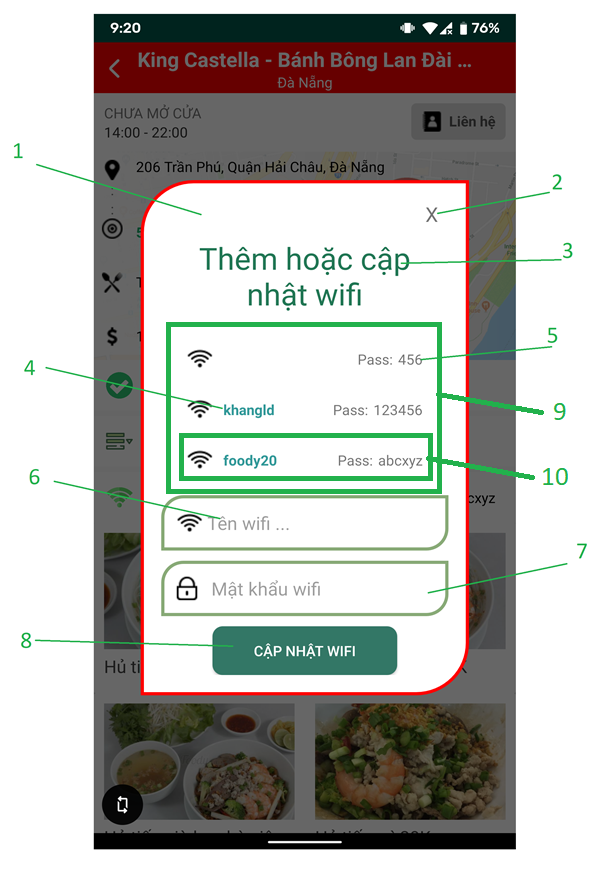
### **2.9. Màn hình menu quán ăn**



### **2.10. Đặc tả màn hình menu quán ăn.**

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Loại** | **Mô tả** |
| 1 | ImageView | Click để quay lại màn hình trước |
| 2 | TextView | Click để xem menu dạng danh sách |
| 3 | TextView | Hiện thị tên của quán |
| 4 | TextView | Hiển thị tên loại món ăn |
| 5 | TextView | Hiển thị tên món ăn |
| 6 | TextView | Hiện thị giá của món ăn |
| 7 | ListView | Dùng để hiển thị danh sách các món ăn trong quán. |
| 8 | Textview | Click để xem menu bằng hình ảnh |

### **2.11. Dialog thêm hoặc cập nhật wifi**



### **2.12. Đặc tả dialog thêm hoặc cập nhật wifi**

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Loại** | **Mô tả** |
| 1 | Dialog | Dùng để hiện thị dialog 1 tab để xem wifi đã có, thêm wifi mới hoặc update mật khẩu wifi đã có. |
| 2 | TextView | Click để tắt dialog. |
| 3 | TextView | Tiêu đề |
| 4 | TextView | Hiển thị tên wifi |
| 5 | TextView | Hiển thị mật khẩu wifi |
| 6 | EditText | Nhập tên wifi cần thêm hoặc cập nhập. |
| 7 | EditText | Nhập mật khẩu wifi |
| 8 | Button | Click để lưu lại. |
| 9 | ListView | Dùng để hiển thị danh sách wifi đã có ở quán. |
| 10 | Layout | Hiển thị chi tiết của 1 wifi trong danh sách. |

# CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ

## **1. CÀI ĐẶT**

### **1.1 Các hằng số value**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Giá trị | Ý nghĩa | Ghi chú |
| 1 | share\_key | String | app\_share\_key | Key của SharedPreferences của ứng dụng |  |
| 2 | key\_province\_name | String | province\_name | Key lưu tên tỉnh/ thành phố trong  SharedPreferences |  |
| 3 | key\_province\_id | String | province\_id | Key lưu mã tỉnh/ thành phố trong  SharedPreferences |  |
| 4 | default\_province\_name | String | Hồ Chí Minh | Tên tỉnh/ thành phố mặc định |  |
| 5 | default\_province\_id | Integer | 1 | Mã tỉnh/ thành phố đặc biệt |  |
| 6 | connection | String | jdbc:jtds:sqlserver:  //Nhom20FoodyDB  .mssql.somee.com/  Nhom20FoodyDB;  user=phucni;password=12345678 | Chuỗi thông tin kết nối tới database online | Chuỗi là liên tục, không xuống dòng |

### **1.2. Các biến trong class MainActivity**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | txtProvinces | TextView | TextView hiển thị tên tỉnh/ thành phố |  |
| 2 | lstFoodPlace | List<FoodPlaceCardViewModel> | Danh sách model chứa dữ liệu của các quán ăn trong một card item | Kiểu dữ liệu sẽ được mô tả rõ hơn ở phần sau |
| 3 | myFoodPlaceAdapter | FoodPlaceCardViewAdapter | Adapter liên kết dữ liệu và view của card item quán ăn |  |
| 4 | edtSearch | EditText | EditText nhập nội dung tìm kiếm |  |
| 5 | progressBar | ProgressBar | Dùng để thông tin cho người dùng biết là dữ liệu đang được tải |  |
| 6 | rcvFoodPlace | RecyclerView | RecyclerView hiển thị danh sách quán ăn |  |
| 7 | isLoading | boolean | Cho biết là có đang tải dữ liệu hay không |  |
| 8 | pageIndex | int | Số thứ tự trang tải | Dùng để tải thêm dữ liệu |
| 9 | pageSize | int | Số lượng bản ghi mỗi trang | Dùng để tải thêm dữ liệu |

### **1.3. Các hàm trong class MainActivity**

* OnCreate
  + Input: Bundle savedInstanceState
  + Output: Không có
  + Mô tả: Hàm này được gọi khi activity create.
  + Chức năng: Gán sự kiện khi refresh activity, liên kết view và biến của activiy, gán các sự kiện của view, gán sự kiện khi cuộn recycler view, tải và gán dữ liệu bất đồng bộ.
* SetEventRefresh
  + Input: Không có
  + Output: Không có
  + Chức năng: Gán sự kiện: khi refresh activity, làm sạch danh sách dữ liệu sau đó tải và gán dữ liệu bất đồng bộ
* BindView
  + Input: Không có
  + Output: Không có
  + Chức năng: Liên kết view và biến của activity
* SetViewEvent
  + Input: Không có
  + Output: Không có
  + Chức năng: Gán sự kiện cho view
* initScrollListener
  + Input: Không có
  + Output: Không có
  + Chức năng: Gán sự kiện: khi recycler view bị cuộn xuống dưới cùng thì tải thêm dữ liệu bất đồng bộ
* loadMore
  + input: Không có
  + Output: Không có
  + Chức năng: tải trang dữ liệu tiếp theo bất đồng bộ
* GetSelectedProvinceId
  + Input: Không có
  + Output: Mã của tỉnh/ thành phố đang được chọn
  + Mô tả: Lấy giá trị của mã tỉnh/ thành phố từ SharedPreferences theo các key trong hằng số
* GetSelectedProvinceName
  + Input: Không có
  + Output: Tên của tỉnh/ thành phố đang được chọn
  + Mô tả: Lấy giá trị của tên tỉnh/ thành phố từ SharedPreferences theo các key trong hằng số
* SetFoodPlaceCards
  + Input: ArrayList<FoodPlaceCardViewModel> foodPlaceCardViewModels
  + Output: Không có
  + Chức năng: Gộp input vào danh sách dữ liệu hiện tại
* Class AsyncTask: GetFoodPlaceCardAsync
  + Mô tả: Liên kết tới database online và tải dữ liệu FoodPlaceCardViewModel bất đồng bộ từ chỉ số trang và kích thước trang.
  + Phương thức khởi tạo: Nhận vào chỉ số trang và kích thước trang
  + Hàm doInBackground: Xây dựng câu query, liên kết tới database online, lấy dữ liệu, gán vào view model sau đó đóng kết nối tới database. Hàm này sẽ trả ra danh sách dữ liệu lấy được và thực hiện điều này một cách bất đồng bộ.
  + Hàm onPreExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đang tải dữ liệu
  + Hàm onPostExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đã tải dữ liệu xong và đổ dữ liệu này lên UI
  + Sử dụng: khởi tạo một đối tượng GetFoodPlaceCardAsync và gọi hàm execute()

### **1.4. Các biến trong class SearchResultActivity**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | tvChooseProvince | TextView | TextView hiển thị tên tỉnh/ thành phố |  |
| 2 | txtBackContainer | LinearLayout | Nút quay lại |  |
| 3 | edtSearch | EditText | EditText nhập thông tin tìm kiếm |  |
| 4 | foodPlaceArrayList | ArrayList<FoodPlaceFullViewModel> | Danh sách dữ liệu hiển thị quán ăn |  |
| 5 | foodPlaceFullViewAdapter | FoodPlaceFullViewAdapter | Adapter liên kết dữ liệu và view quán ăn |  |
| 6 | progressBar | ProgressBar | Thông tin cho người dùng biết là đang tải dữ liệu |  |
| 7 | btnBestMatch | TextView | Nút chọn loại tìm kiếm là giống nhất |  |
| 8 | btnNearby | TextView | Nút chọn loại tìm kiếm là gần đây |  |
| 9 | btnPopular | TextView | Nút chọn loại tìm kiếm theo độ phổ biến |  |
| 10 | pageIndex | Int | Chỉ số trang |  |
| 11 | searchstring | String | Chuỗi thông tin tìm kiếm |  |
| 12 | timer | Timer | Timer để deley |  |
| 13 | SearchType {BestMatch, Popular, Nearby} | enum | Quy định kiểu tìm kiếm |  |
| 14 | lnlPopular | LinearLayout | Chứa kết quả tìm kiếm nổi tiếng nhất |  |
| 15 | lnlBestmatch | LinearLayout | Chứa kết quả tìm kiếm giống nhất |  |
| 16 | lnlNearBy | LinearLayout | Chứa kết quả tìm kiếm gần đây |  |
| 17 | currentSearchType | SearchType | Loại tìm kiếm hiện tại |  |
| 18 | myLocation | Location | Vị trí của thiết bị |  |
| 19 | foodplaceLocation | Location | Vị trí quán ăn |  |
| 20 | client | FusedLocationProviderClient | Provider lấy vị trí |  |
| 21 | geocoder | Geocoder | Lấy vị trí |  |
| 22 | Foodplaceaddresses | List<Address> | Danh sách địa chỉ |  |
| 23 | rcvFoodPlace | RecyclerView | RecyclerView hiển thị quán ăn |  |

### **1.5. Các hàm trong class SearchResultActivity**

* onResume
  + Input: Không có
  + Output: Không có
  + Mô tả: được gọi khi activity resume
  + Chức năng: Hiển thị tên tỉnh/ thành phố và tải dữ liệu
* finish
  + Input: Không có
  + Output: Không có
  + Mô tả: được gọi khi activity finish
  + Chức năng: Trở về main activity
* onCreate
  + Input: Bundle savedInstanceState
  + Output: Không có
  + Mô tả: Hàm này được gọi khi activity create.
  + Chức năng: Liên kết view và biến của activiy, gán các sự kiện của view, lấy vị trí hiện tại, hiển thị tỉnh/ thành phố đang chọn, tải và gán dữ liệu bất đồng bộ.
* BindView
  + Input: Không có
  + Output: Không có
  + Chức năng: Liên kết view và biến của activity
* SetViewEvent
  + Input: Không có
  + Output: Không có
  + Chức năng: Gán sự kiện cho view
* resetBgr
  + Input: Không có
  + Output: Không có
  + Chức năng: gán lại màu nền
* Distance
  + Input: String foodplaceAddress
  + Output: Khoảng cách từ vị trị hiện tại đến địa chỉ truyền vô
  + Mô tả: Tính toán và chuẩn hóa khoảng cách
* GetMyLocation:
  + Input: Không có
  + Output: Không có
  + Chức năng: Lấy vị trí hiện tại
* SortByDistance
  + Input: ArrayList<FoodPlaceFullViewModel> tempfoodPlaceArrayList
  + Output: Danh sách đã được sắp xếp tăng dần theo khoảng cách
  + Mô tả: Tính toán khoảng cách và sắp xếp tăng dần
* ExecuteQuery
  + Input: Không có
  + Output: Không có
  + Chức năng: Tải dữ liệu
* GetSelectedProvinceId
  + Input: Không có
  + Output: Mã của tỉnh/ thành phố đang được chọn
  + Mô tả: Lấy giá trị của mã tỉnh/ thành phố từ SharedPreferences theo các key trong hằng số
* GetSelectedProvinceName
  + Input: Không có
  + Output: Tên của tỉnh/ thành phố đang được chọn
  + Mô tả: Lấy giá trị của tên tỉnh/ thành phố từ SharedPreferences theo các key trong hằng số
* SetFoodPlaceFull
  + Input: ArrayList<FoodPlaceFullViewModel> foodPlaceFullViewModels
  + Output: Không có
  + Chức năng: Làm sạch danh sách dữ liệu hiện tại rồi gán danh sách dữ liệu truyền vô vào danh sách hiên tại
* Class AsyncTask: GetFoodPlaceFullAsync
  + Mô tả: Liên kết tới database online và tải dữ liệu FoodPlaceCardViewModel bất đồng bộ từ mã tỉnh/ thành phố, chuỗi tìm kiếm và loại tìm kiếm
  + Phương thức khởi tạo: Nhận vào từ mã tỉnh/ thành phố, chuỗi tìm kiếm và loại tìm kiếm
  + Hàm doInBackground: Xây dựng câu query, liên kết tới database online, lấy dữ liệu, gán vào view model sau đó đóng kết nối tới database. Hàm này sẽ trả ra danh sách dữ liệu lấy được và thực hiện điều này một cách bất đồng bộ.
  + Hàm onPreExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đang tải dữ liệu
  + Hàm onPostExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đã tải dữ liệu xong và đổ dữ liệu này lên UI
  + Sử dụng: khởi tạo một đối tượng GetFoodPlaceFullAsync và gọi hàm execute()

### **1.6. Các biến trong class ChooseProvinceActivity**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | btnCancel | TextView | Nút hủy |  |
| 2 | btnDone | TextView | Nút xong |  |
| 3 | lstProvinces | ListView | ListView hiển thị tên tỉnh/ thành phố |  |
| 4 | edtSearch | EditText | EditText tìm kiếm |  |
| 5 | provinceArrayList | ArrayList<Province> | Danh sách tỉnh |  |
| 6 | provinceTempArrayList | ArrayList<Province> | Danh sách tỉnh tạm | Để tìm kiếm |
| 7 | provinceAdapter | ProvinceViewAdapter | Adapter liên kết dữ liệu và view tỉnh |  |
| 8 | selectedProvinceName | String | Tỉnh được chọn |  |
| 9 | selectedProvinceId | int | Mã tỉnh được chọn |  |
| 10 | intentProvince | Intent | Intent trở về main activity |  |
| 11 | progressBar | ProgressBar | Thông tin cho người dùng biết là đang tải dữ liệu |  |

### **1.7. Các hàm trong class ChooseProvinceActivity**

* onCreate
  + Input: Bundle savedInstanceState
  + Output: Không có
  + Mô tả: Được gọi khi activity create
  + Chức năng: Liên kết view và biến, gán sự kiện view, tải dữ liệu bất đồng bộ
* SaveProvince
  + Input: Không có
  + Output: Không có
  + Chức năng: Lưu tên và mã tỉnh vào SharePreferences theo key liệt kê ở hằng số
* GetSelectedProvinceName
  + Input: Không có
  + Output: Tên của tỉnh/ thành phố đang được chọn
  + Mô tả: Lấy giá trị của tên tỉnh/ thành phố từ SharedPreferences theo các key trong hằng số
* SendTempProvinceName
  + Input: String name, Integer id
  + Output: Không có
  + Chức năng: Gán mã và tên của tỉnh được chọn
* SetProvinces
  + Input: ArrayList<Province> provinces
  + Output: Không có
  + Chức năng: Làm mới và gán tỉnh truyền vô
* Class TaskAsync GetProvincesAsync:
  + Mô tả: Liên kết tới database online và tải dữ liệu Province bất đồng bộ
  + Hàm doInBackground: Xây dựng câu query, liên kết tới database online, lấy dữ liệu, gán vào view model sau đó đóng kết nối tới database. Hàm này sẽ trả ra danh sách dữ liệu lấy được và thực hiện điều này một cách bất đồng bộ.
  + Hàm onPreExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đang tải dữ liệu
  + Hàm onPostExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đã tải dữ liệu xong và đổ dữ liệu này lên UI
  + Sử dụng: khởi tạo một đối tượng GetProvincesAsync và gọi hàm execute()

### **1.8. Các biến trong class DetailsActivity**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | Client | FusedLocationProviderClient | Provider lấy vị trí |  |
| 2 | myLocation | Location | Vị trí của thiết bị |  |
| 3 | foodplaceLocation | Location | Vị trí quán ăn |  |
| 4 | geocoder | Geocoder | Lấy vị trí |  |
| 5 | addresses | List<Address> | Danh sách địa chỉ |  |
| 6 | tvAddress | TextView | TextView hiển thị địa chỉ |  |
| 7 | tvDistance | TextView | TextView hiển thị khoảng cách |  |
| 8 | tvType | TextView | TextView hiển thị loại quán ăn |  |
| 9 | tvPrice | TextView | TextView hiển thị giá |  |
| 10 | tvFoodPlaceName | TextView | TextView hiển thị tên quán ăn |  |
| 11 | tvTime | TextView | TextView hiển thị giờ đóng/ mở cửa |  |
| 12 | tvProvinceName | TextView | TextView hiển thị tên tỉnh |  |
| 13 | tvStatus | TextView | TextView hiển thị quán đang đóng hay mở cửa |  |
| 14 | tvContact | TextView | TextView hiển thị liên lạc |  |
| 15 | txtWifiPass | TextView | TextView hiển thị mật khẩu wifi |  |
| 16 | txtWifiName | TextView | TextView hiển thị tên wifi |  |
| 17 | contact | String | Liên lạc |  |
| 18 | progressBar | ProgressBar | Thông tin cho người dùng biết là đang tải dữ liệu |  |
| 19 | lstWifi | List<WifiViewModel> | Danh sách wifi |  |
| 20 | lstFood | List<FoodViewModel> | Danh sách món ăn |  |
| 21 | wifiAdapter | WifiAdapter | Adapter wifi |  |
| 22 | myFoodAdapter | FoodViewAdapter | Adapter món ăn |  |
| 23 | addressFoodPlace | String | Địa chỉ quán ăn |  |
| 24 | btnBackDetails | LinearLayout | Nút quay lại |  |
| 25 | lineMenu | ConstraintLayout | Dòng thực đơn |  |
| 26 | lineWifi | ConstraintLayout | Dòng wifi |  |
| 27 | maps | LinearLayout | Hiển thị hình ảnh bản đồ |  |
| 28 | wifi | WifiViewModel | Wifi |  |
| 29 | Id | int | Mã tỉnh |  |

### **1.9. Các hàm trong class DetailsActivity**

* onCreate
  + Input: Bundle savedInstanceState
  + Output: Không có
  + Mô tả: Được gọi khi activity create
  + Chức năng: Liên kết view và biến, gán sự kiện view, lấy vị trí hiện tại, tải dữ liệu bất đồng bộ
* DialogAddWifi
  + Input: Không có
  + Output: Không có
  + Chức năng: mở dialog wifi
* SetWifis
  + Input: ArrayList<WifiViewModel> wifiViewModels
  + Output: Không có
  + Chức năng: Gán danh sách wifi truyền vô vào danh sách dữ liệu wifi
* GetTimeSystem
  + Input: Không có
  + Output: Chuỗi giờ đã định dạng
  + Mô tả: Lấy và định dạng giờ hiện tại
* Distance
  + Input: String foodplaceAddress
  + Output: Khoảng cách từ vị trị hiện tại đến địa chỉ truyền vô
  + Mô tả: Tính toán và chuẩn hóa khoảng cách
* GetMyLocation:
  + Input: Không có
  + Output: Không có
  + Chức năng: Lấy vị trí hiện tại
* SetFoodWithImage
  + Input: ArrayList<FoodViewModel> foodViewModels
  + Output: Không có
  + Chức năng: Gán danh sách thức ăn truyền vô vào danh sách thức ăn hiện tại
* SetFoodPlaceDetail
  + Input: FoodPlaceDetailViewModel foodPlaceDetailViewModel
  + Output: Không có
  + Chức năng: Gán thông tin quán ăn từ input
* Class AsyncTask: GetFoodPlaceDetailAsync
  + Mô tả: Liên kết tới database online và tải dữ liệu FoodPlaceDetailViewModel bất đồng bộ từ mã quán ăn
  + Phương thức khởi tạo: Nhận vào mã quán ăn
  + Hàm doInBackground: Xây dựng câu query, liên kết tới database online, lấy dữ liệu, gán vào view model sau đó đóng kết nối tới database. Hàm này sẽ trả ra danh sách dữ liệu lấy được và thực hiện điều này một cách bất đồng bộ.
  + Hàm onPreExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đang tải dữ liệu
  + Hàm onPostExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đã tải dữ liệu xong và đổ dữ liệu này lên UI
  + Sử dụng: khởi tạo một đối tượng GetFoodPlaceDetailAsync và gọi hàm execute()
* Class AsyncTask: GetFoodWithImageAsync
  + Mô tả: Liên kết tới database online và tải dữ liệu ArrayList<FoodViewModel> bất đồng bộ từ mã quán ăn
  + Phương thức khởi tạo: Nhận vào mã quán ăn
  + Hàm doInBackground: Xây dựng câu query, liên kết tới database online, lấy dữ liệu, gán vào view model sau đó đóng kết nối tới database. Hàm này sẽ trả ra danh sách dữ liệu lấy được và thực hiện điều này một cách bất đồng bộ.
  + Hàm onPreExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đang tải dữ liệu
  + Hàm onPostExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đã tải dữ liệu xong và đổ dữ liệu này lên UI
  + Sử dụng: khởi tạo một đối tượng GetFoodWithImageAsync và gọi hàm execute()
* Class AsyncTask: GetWifisAsync
  + Mô tả: Liên kết tới database online và tải dữ liệu ArrayList<WifiViewModel> bất đồng bộ từ mã quán ăn
  + Phương thức khởi tạo: Nhận vào mã quán ăn
  + Hàm doInBackground: Xây dựng câu query, liên kết tới database online, lấy dữ liệu, gán vào view model sau đó đóng kết nối tới database. Hàm này sẽ trả ra danh sách dữ liệu lấy được và thực hiện điều này một cách bất đồng bộ.
  + Hàm onPreExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đang tải dữ liệu
  + Hàm onPostExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đã tải dữ liệu xong và đổ dữ liệu này lên UI
  + Sử dụng: khởi tạo một đối tượng GetWifisAsync và gọi hàm execute()
* Class AsyncTask: AddOrUpdateWifiAsync
  + Mô tả: Liên kết tới database online và thực hiện cập nhật hoặc thêm một WifiViewModel bất đồng bộ
  + Phương thức khởi tạo: Nhận vào WifiViewModel
  + Hàm doInBackground: Xây dựng câu query, liên kết tới database online thực thi lệnh sau đó đóng kết nối tới database.
  + Hàm onPreExecute: Hàm này được gọi ở main thread(UI thread)
  + Hàm onPostExecute: Hàm này được gọi ở main thread(UI thread)
  + dụng: khởi tạo một đối tượng AddOrUpdateWifiAsync và gọi hàm execute()

### **1.10. Các biến trong class MenuActivity**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | btnChangeMenu | LinearLayout | Nút đổi menu |  |
| 2 | btnBackMenu | LinearLayout | Nút quay lại |  |
| 3 | rcvImageMenu | RecyclerView | Hiển thị món ăn |  |
| 4 | eplMenu | ExpandableListView | HIển thị món ăn theo nhóm |  |
| 5 | tvNameMenu | TextView | TextView hiển thị tên |  |
| 6 | tvShowPicture | TextView | TextView chọn hiển thị hình |  |
| 7 | tvShowMenu | TextView | TextView chọn hiển thị không hình |  |
| 8 | lstFood | List<FoodViewModel> | Danh sách thức ăn |  |
| 9 | ImgFoodAdapter | FoodViewAdapter | Adapter thức ăn |  |
| 10 | adapter | ExpandMenuAdapter | adapter nhóm thức ăn |  |
| 11 | lstFoodKind | List<String> | Danh sách nhóm thức ăn |  |
| 12 | lstFoodInKind | HashMap<String,List<FoodViewModel>> | Key-value thức ăn và nhóm thức ăn |  |

### **1.11. Các hàm trong class MenuActivity**

* onCreate
  + Input: Bundle savedInstanceState
  + Output: Không có
  + Mô tả: Được gọi khi activity create
  + Chức năng: Liên kết view và biến, gán sự kiện view, lấy mã quán ăn hiện tại, tải dữ liệu bất đồng bộ
* SetFood
  + Input: ArrayList<FoodViewModel> foodViewModels
  + Output: Không có
  + Chức năng: Gán thông tin nhận được vào danh sách hiện tại
* Class AsyncTask: GetFoodAsync
  + Mô tả: Liên kết tới database online và tải dữ liệu ArrayList<FoodViewModel> bất đồng bộ từ mã quán ăn
  + Phương thức khởi tạo: Nhận vào mã quán ăn
  + Hàm doInBackground: Xây dựng câu query, liên kết tới database online, lấy dữ liệu, gán vào view model sau đó đóng kết nối tới database. Hàm này sẽ trả ra danh sách dữ liệu lấy được và thực hiện điều này một cách bất đồng bộ.
  + Hàm onPreExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đang tải dữ liệu
  + Hàm onPostExecute: Hàm này được gọi ở main thread(UI thread), dùng để thông tin cho người dùng biết là đã tải dữ liệu xong và đổ dữ liệu này lên UI
  + Sử dụng: khởi tạo một đối tượng GetFoodAsync và gọi hàm execute()

### **1.12. Các view model**

#### 1.14.1 WifiViewModel

ViewModel bao gồm member variable, getter, setter và constructor của tất cả member variable

Danh sách member variable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | name | String | Tên wifi |  |
| 2 | password | String | Mật khẩu wifi |  |
| 3 | foodPlaceId | Int | Mã của quán ăn |  |

#### 1.14.2. Province

ViewModel bao gồm member variable, getter, setter và constructor của tất cả member variable

Danh sách member variable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | ProvinceId | Int | Mã tỉnh |  |
| 2 | ProvinceName | String | Tên tỉnh |  |

#### 1.14.3 FoodViewModel

ViewModel bao gồm member variable, getter, setter và constructor của tất cả member variable

Danh sách member variable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | id | Int | Mã thức ăn |  |
| 2 | foodName | String | Tên thức ăn |  |
| 3 | price | BigDecimal | Giá |  |
| 4 | foodImage | String | Đường dẫn hình ảnh |  |
| 5 | foodPlaceId | int | Mã quán ăn |  |
| 6 | typeId | int | Mã loại thưc ăn |  |
| 7 | typeName | String | Tên loại thức ăn |  |

#### 1.14.4 FoodPlaceFullViewModel

ViewModel bao gồm member variable, getter, setter và constructor của tất cả member variable

Danh sách member variable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | id | int | Mã quán ăn |  |
| 2 | name | String | Tên quán ăn |  |
| 3 | address | String | Địa chỉ |  |
| 4 | type | String | Loại quán ăn |  |
| 5 | image | String | Đường dẫn hình ảnh |  |
| 6 | openTime | Time | GIờ mở cửa |  |
| 7 | closeTIme | Time | GIờ đóng cửa |  |
| 8 | reviewContent | String | Nội dung đánh giá |  |
| 9 | reviewCount | int | Số lượng đánh giá |  |
| 10 | checkinCount | int | Số lượng checkin |  |
| 11 | rate | float | Điểm đánh giá |  |
| 12 | distance | double | Khoảng cách của quán ăn tới vị trí hiện tại |  |

#### 1.14.5 FoodPlaceDetailsViewModel

ViewModel bao gồm member variable, getter, setter và constructor của tất cả member variable

Danh sách member variable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | id | int | Mã quán ăn |  |
| 2 | name | String | Tên quán ăn |  |
| 3 | address | String | Địa chỉ |  |
| 4 | type | String | Loại quán ăn |  |
| 5 | image | String | Đường dẫn hình ảnh |  |
| 6 | openTime | Time | GIờ mở cửa |  |
| 7 | closeTIme | Time | GIờ đóng cửa |  |
| 8 | minPrice | BigDecimal | Giá nhỏ nhất |  |
| 9 | maxPrice | BigDecimal | Giá cao nhất |  |
| 10 | provinceName | String | Tên tỉnh |  |

#### 1.14.6 FoodPlaceCardViewModel

ViewModel bao gồm member variable, getter, setter và constructor của tất cả member variable

Danh sách member variable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | id | int | Mã quán ăn |  |
| 2 | name | String | Tên quán ăn |  |
| 3 | image | String | Đường dẫn hình ảnh |  |
| 4 | review | String | Đánh giá chung |  |

### **1.13. WifiAdapter**

Các biến

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | context | Context | Context gắn với adapter |  |
| 2 | lstWifi | List<WifiViewModel> | Danh sách wifi |  |

Các hàm

* Constructor:
  + Input: Context context, List<WifiViewModel> lstWifi
  + Output: Không có
  + Chức năng: Gán thông tin cho adapter
* getCount
  + Input: Không có
  + Output: kích thước của danh sách dữ liệu
* Class ViewHolder: Bao gồm các thuộc tính cần thiết để bind với view layout
* GetView
  + Input: int position, View convertView, ViewGroup parent
  + Output: view đã được thêm thông tin
  + Mô tả: Lấy thông tin từ input và danh sách dữ liệu gán vào view trả ra

### **1.14. ProvinceViewAdapter**

Các biến

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | context | Context | Context gắn với adapter |  |
| 2 | layout | Int | Mã layout view |  |
| 3 | provinceList | List<Province> | Danh sách tỉnh |  |
| 4 | provinceName | String | Tên tỉnh |  |
| 5 | mSelectedItem | int | Mã tỉnh được chọn |  |

Các hàm

* Constructor:
  + Input: ChooseProvincesActivity context, int layout, List<Province> provinceList, String provinceName
  + Output: Không có
  + Chức năng: Gán thông tin cho adapter
* getCount
  + Input: Không có
  + Output: kích thước của danh sách dữ liệu
* Class ViewHolder: Bao gồm các thuộc tính cần thiết để bind với view layout
* GetView
  + Input: int position, View convertView, ViewGroup parent
  + Output: view đã được thêm thông tin
  + Mô tả: Lấy thông tin từ input và danh sách dữ liệu gán vào view trả ra

### **1.15 FoodViewAdapter**

Các biến

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | mContext | Context | Context gắn với adapter |  |
| 2 | mData | List<FoodViewModel> | Danh sách dữ liệu |  |

Các hàm

* Constructor:
  + Input: Context mContext, List<FoodViewModel> mData
  + Output: Không có
  + Chức năng: Gán thông tin cho adapter
* onCreateViewHolder
  + Input: @NonNull ViewGroup parent, int viewType
  + Output: ViewHolder
  + Mô tả: Tạo view holder từ input
* onBindViewHolder
  + Input: @NonNull MyViewHolder holder, int position
  + Output: Không có
  + Chức năng: Gán thông tin vào view holder
* getItemCount
  + Input: Không có
  + Output: kích thước của danh sách dữ liệu
* Class ViewHolder: Bao gồm các thuộc tính cần thiết để bind với view layout

### **1.16. FoodPlaceFullViewAdapter**

Các biến

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | mActivity | Activity | Activity |  |
| 2 | Layout | int | View layout |  |
| 3 | foodplaceList | List<FoodPlaceFullViewModel> | Danh sách dữ liệu |  |
| 4 | client | FusedLocationProviderClient | Provider lấy vị trí |  |
| 5 | myLocation | Location | Vị trí hiện tại |  |
| 6 | foodplaceLocation | Location | Vị trí của quán ăn |  |
| 7 | geocoder | Geocoder | Lấy vị trí |  |
| 8 | foodplaceaddresses | List<Address> | Danh sách địa chỉ |  |
| 9 | mContext | Context | Context gắn với adapter |  |

Các hàm

* Constructor:
  + Input: Activity mActivity,Context mContext, List<FoodPlaceFullViewModel> foodplaceList
  + Output: Không có
  + Chức năng: Gán thông tin cho adapter
* onCreateViewHolder
  + Input: @NonNull ViewGroup parent, int viewType
  + Output: ViewHolder
  + Mô tả: Tạo view holder từ input
* onBindViewHolder
  + Input: @NonNull MyViewHolder holder, int position
  + Output: Không có
  + Chức năng: Gán thông tin vào view holder
* getItemCount
  + Input: Không có
  + Output: kích thước của danh sách dữ liệu
* Class ViewHolder: Bao gồm các thuộc tính cần thiết để bind với view layout
* Distance
  + Input: String foodplaceAddress
  + Output: Khoảng cách từ vị trị hiện tại đến địa chỉ truyền vô
  + Mô tả: Tính toán và chuẩn hóa khoảng cách
* GetMyLocation:
  + Input: Không có
  + Output: Không có
  + Chức năng: Lấy vị trí hiện tại

### **1.17. FoodPlaceCardViewAdapter**

Các biến

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | mContext | Context | Context gắn với adapter |  |
| 2 | mData | List<FoodPlaceCardViewModel> | Danh sách dữ liệu |  |

Các hàm

* Constructor:
  + Input: Context mContext, List<FoodPlaceCardViewModel> mDatafoodplaceList
  + Output: Không có
  + Chức năng: Gán thông tin cho adapter
* onCreateViewHolder
  + Input: @NonNull ViewGroup parent, int viewType
  + Output: ViewHolder
  + Mô tả: Tạo view holder từ input
* onBindViewHolder
  + Input: @NonNull MyViewHolder holder, int position
  + Output: Không có
  + Chức năng: Gán thông tin vào view holder
* getItemCount
  + Input: Không có
  + Output: kích thước của danh sách dữ liệu
* getItemViewType
  + Input: int position
  + Output: item type
* Class ItemViewHolder: Bao gồm các thuộc tính cần thiết để bind với view layout
* Class LoadingViewHolder: View Loading
* populateItemRows
  + input: ItemViewHolder viewHolder, final int position
  + output: Không có
  + Chức năng: tạo viewholder dựa trên position

### **1.18. ExpandMenuAdapter**

Các biến

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa | Ghi chú |
| 1 | context | Context | Context gắn với adapter |  |
| 2 | lstFoodKind | List<String> | Danh sách nhóm |  |
| 3 | lstFoodInKind | HashMap<String,List<FoodViewModel>> | Key-Value nhóm và thức ăn trong nhóm |  |

* Constructor:
  + Input: Context context, List<String> lstFoodKind, HashMap<String, List<FoodViewModel>> lstFoodInKind
  + Output: Không có
  + Chức năng: Gán thông tin cho adapter
* getGroupCount
  + Input: Không có
  + Output: kích thước của danh sách dữ liệu
* getChildrenCount
  + input: int position
  + Output: kích thước danh sách dữ liệu con
* getGroup
  + input: int position
  + Output: dữ liệu trong danh sách
* getChild
  + Input: int groupPosition, int childPosition
  + Output: dữ liệu trong danh sách con tương ứng
* getGroupView
  + Input: int groupPosition, boolean isExpanded, View convertView, ViewGroup parent
  + Output: Group view
* getChildView
  + Input: int groupPosition, int childPosition, boolean isLastChild, View convertView, ViewGroup parent
  + Output: Group child view

## **2. KIỂM THỬ**

### **2.1. Bảng mô tả quá trình test**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| **Functional Test Case** | | | | | | |
| Foody\_01 | Load dữ liệu khi mở ứng dụng | 1. Mở ứng dụng | ref DATA: Bảng dữ liệu FoodPlace | 1. Màn hình chính hiển thị kết quả quán ăn | Hiển thị đúng kết quả | Pass |
| Foody\_02 | Hiển thị chi tiết quán ăn | 1. Mở ứng dụng 2. Nhấn vào hình ảnh quán ăn bất kỳ | ref DATA: Bảng dữ liệu FoodPlace, FoodInMenu, Wifi | 1. Hiển thị đúng được thông tin quán ăn vừa được nhấn | Hiển thị đúng kết quả | Pass |
| Foody\_03 | Chọn thành phố, tỉnh | 1. Mở ứng dụng 2. Nhấn tên tỉnh/thành phố 3. Chọn thành phố hoặc tỉnh 4. Nhấn xong | ref DATA: Bảng dữ liệu Province | 1.Hiển thị đúng tỉnh, thành phố trong Màn hình chính  2.Nếu bấm hủy thì thành phố không được lưu lại  3.Màn hình chính tự động load lại quán ăn của tỉnh đó | Hiển thị đúng kết quả | Pass |
| Foody \_04 | Tìm kiếm tỉnh thành phố | 1. Mở ứng dụng 2. Nhấn tên tỉnh /thành phố 3. Nhập keyword vào EditText tìm kiếm | ref DATA: INPUT\_01 | 1.Hiển thị đúng tên tỉnh/thảnh phố có chưa keyword | Hiển thị đúng kết quả | Pass |
| Foody \_05 | Tìm kiếm quán ăn | 1. Mở ứng dụng 2. Bấm chọn thanh tìm kiếm 3. Nhập keyword vào EditText tìm kiếm | ref DATA: INPUT\_02 | 1.Hiển thị đúng tên quán ăn có chưa keyword | Hiển thị đúng kết quả | Pass |
| Foody \_06 | Liên hệ với chủ quán ăn | 1.Mở ứng dụng  2.Nhấn vào hình ảnh quán ăn bất kỳ  3.Ấn vào “Liên Hệ” | ref DATA: Bảng Data FoodPlace | 1. Có yêu cầu quyền sử dụng ứng dụng điện thoại của người dùng  2.Gọi tới được số điện thoại của chủ quán ăn | Kết quả mong đợi | Pass |
| Foody \_07 | Hiển thị vị trí của quán ăn trên bản đồ | 1.Mở ứng dụng  2.Nhấn vào hình ảnh quán ăn bất kỳ  3.Ấn vào hình ảnh bản đồ | ref DATA: Bảng Data FoodPlace | 1. Có yêu cầu quyền mở bản đồ từ người dùng 2. Quán ăn được hiển thị trên bản đồ | Kết quả mong đợi | Pass |
| Foody \_08 | Hiển thị wifi đã có, thêm hoặc cập nhật wifi | 1.Mở ứng dụng  2.Nhấn vào hình ảnh quán ăn bất kỳ  3.Ấn vào dòng wifi quán ăn | ref DATA: Bảng Data FoodPlace, Wifi  INPUT\_03 | Hiển thị ra 1 dialog thể hiện đúng các wifi đã có. | Kết quả mong đợi | Pass |

### **2.2. Bảng mô tả Data quá trình test**

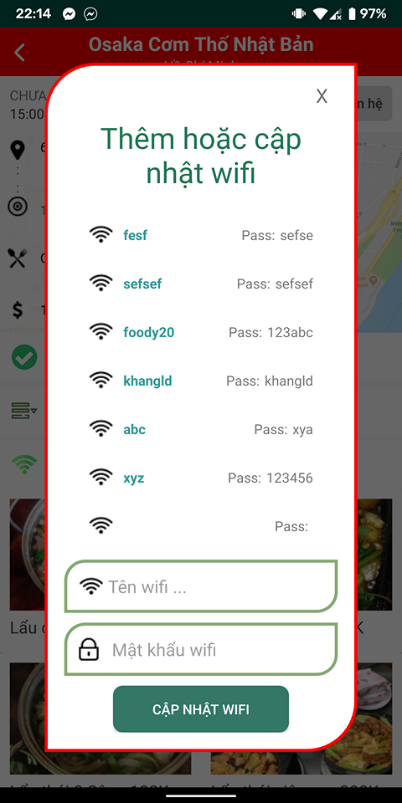
INPUT\_01

|  |  |
| --- | --- |
| **a** | **b** |
| L | * Bạc Liêu * Đắk Lắk * Gia Lai * Lai Châu * Lạng Sơn * Lào Cai * Lâm Đồng * Long An * Sơn La * Vĩnh Long |
| Th | * Cần Thơ * Bình Thuận * Đồng Tháp * Ninh Thuận * Phú Thọ * Thái Bình * Thái Nguyên * Thanh Hóa |

INPUT\_02 (thanh phố hiện tại là TP HCM)

|  |  |
| --- | --- |
| **a** | **b** |
| th | * Osaka cơm thố Nhật Bản * The coffee X * Light Cueicha – Huỳnh Thúc Kháng * Trà sữa Chamichi – Cao Thắng * Smoothie Factory Việt Nam – Thức uống * The coffee house – Xô viết Nghệ tinh. |
| Sữa | * Ăn vặt đóm – Sữa chua uống * Trà sữa Chamichi – Cao thắng * Trà sữa Sharetea – Nguyễn Huệ. |

INPUT\_03



# PHẦN KẾT LUẬN

## **1. KẾT QUẢ ĐẠT ĐƯỢC**

Nghiên cứu các thành phần cơ bản của công nghệ lập trình di động Android Studio như:các thành phần giao diện, thành phần chức năng.

Sử dụng công nghệ Android Studio kết hợp để xây dựng một ứng dụng app tính toán Foody với các chức năng:

* Chọn tỉnh/ thành phố muốn xem quán ăn
* Xem quán ăn trong tỉnh/thành phố
* Tìm kiếm quán ăn theo ba tiêu chí: Đúng nhất, phổ biến, gần tôi
* Xem chi tiết quán ăn
* Xem menu quán ăn
* Xem món có trong menu quán ăn
* Tìm vị trí quán ăn
* Xem khoảng cách từ quán ăn đến người dùng, chỉ đường tới quán ăn
* Tra cứu vị trí quán ăn trên map
* Xem, thêm, chia sẻ wifi quán ăn

## **2. ƯU ĐIỂM**

* Thực hiện các chức năng cơ bản của một app Foody.
* Xử lý tốt các chức năng cơ bản .
* Giao diện app đẹp, dễ sử dụng. .

## **3. NHƯỢC ĐIỂM**

* Chưa xử lý được phần Delivery
* Vì sử dụng Database online nên đổi khi dữ liệu load lâu

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **NHIỆM VỤ** | **KẾT QUẢ** |
| **1** | Huỳnh Ngọc Hưng Thịnh | * Xử lý sự kiện * Kiểm thử ứng dụng * Viết báo cáo | Hoàn thành tốt |
| **2** | Lê Đức Khang | * Thiết kế giao diện * Kiểm thử ứng dụng * Tổng hợp báo cáo | Hoàn thành tốt |
| **3** | Nguyễn Huỳnh Phúc | * Tạo dataabase online, tạo dữ liệu * Nghiên cứu luồng xử lý thuật toán * Viết báo cáo | Hoàn thành tốt |