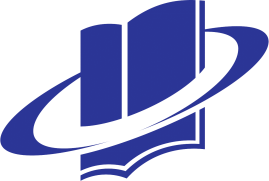
TRƯỜNG ĐẠI HỌC MỞ TP.HCM

**KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO CHI TIẾT VỀ SẢN PHẨM

**XÂY DỰNG ỨNG DỤNG GIỚI THIỆU ẨM THỰC CHO SMARTPHONE**

CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

Sinh viên thực hiện: Nguyễn Quốc Việt

MSSV: 1651012221

Người hướng dẫn: Thạc sĩ Lưu Quang Phương

Tp.HCM, năm 2019

# Mục lục

[Mục lục 2](#_Toc22486280)

[CHƯƠNG 1: ỨNG DỤNG GIỚI THIỆU ẨM THỰC CHO SMARTPHONE 5](#_Toc22486281)

[1. Mô tả hoạt động của phần mềm 5](#_Toc22486282)

[2. Cơ sở dữ liệu 5](#_Toc22486283)

[3. Node dacsans 6](#_Toc22486284)

[4. Sơ đồ phân rã chức năng phần mềm 8](#_Toc22486285)

[5. Ứng dụng giới thiệu ẩm thực cho smartphone 11](#_Toc22486286)

[CHƯƠNG 2.Models 17](#_Toc22486287)

[2.1 Firebase Models 17](#_Toc22486288)

[**1.1.1.** 17](#_Toc22486289)

[**1.1.2.** Objdanhgia 18](#_Toc22486290)

[**1.1.3.** Objluubaiviet 18](#_Toc22486291)

[**1.1.4.** Objnguoidungs 19](#_Toc22486292)

[**1.1.5.** Objthanhphan 20](#_Toc22486293)

[**1.1.6.** Objtinhthanh 20](#_Toc22486294)

[1.2. 2.2 App Models 21](#_Toc22486295)

[**1.2.1.** Objbaiviet\_app 21](#_Toc22486296)

[**1.2.2.** Objbinhluan\_app 21](#_Toc22486297)

[**1.2.3.** Objdanhgia\_app 21](#_Toc22486298)

[**1.2.4.** Objgoiy\_app 22](#_Toc22486299)

[**1.2.5.** Objkhuvuc\_app 22](#_Toc22486300)

[**1.2.6.** Objnguoidung\_app 22](#_Toc22486301)

[**1.2.7.** Objthongkedanhgia\_app 23](#_Toc22486302)

[**1.2.8.** Objthucpham\_app 23](#_Toc22486303)

[1.3. Model Admin 24](#_Toc22486304)

[CHƯƠNG 3. Utils 25](#_Toc22486305)

[1.4. CountryData Class 25](#_Toc22486306)

[1.5. getDataApp 26](#_Toc22486307)

[1.6. DowloadImage 27](#_Toc22486308)

[1.7. CustomDiaLogLoading 28](#_Toc22486309)

[CHƯƠNG 4. View 28](#_Toc22486310)

[1.8. 4.1FplashScreen 28](#_Toc22486311)

[**1.8.1.** SplashScreenActivity 28](#_Toc22486312)

[**1.8.2.** IndexActivity 29](#_Toc22486313)

[**1.8.3.** SelectKhuVucActivity 31](#_Toc22486314)

[1.9. Main 33](#_Toc22486315)

[**1.9.1.** MainActivity 33](#_Toc22486316)

[**1.9.2.** AngiFragment 35](#_Toc22486317)

[**1.9.3.** UongGiFragment 38](#_Toc22486318)

[**1.9.4.** SettingActivity 40](#_Toc22486319)

[1.10. Tài khoản 41](#_Toc22486320)

[**1.10.1.** Admin Activity 41](#_Toc22486321)

[**1.10.2.** ChinhsuabaivietActivity 47](#_Toc22486322)

# CHƯƠNG 1: ỨNG DỤNG GIỚI THIỆU ẨM THỰC CHO SMARTPHONE

## Mô tả hoạt động của phần mềm

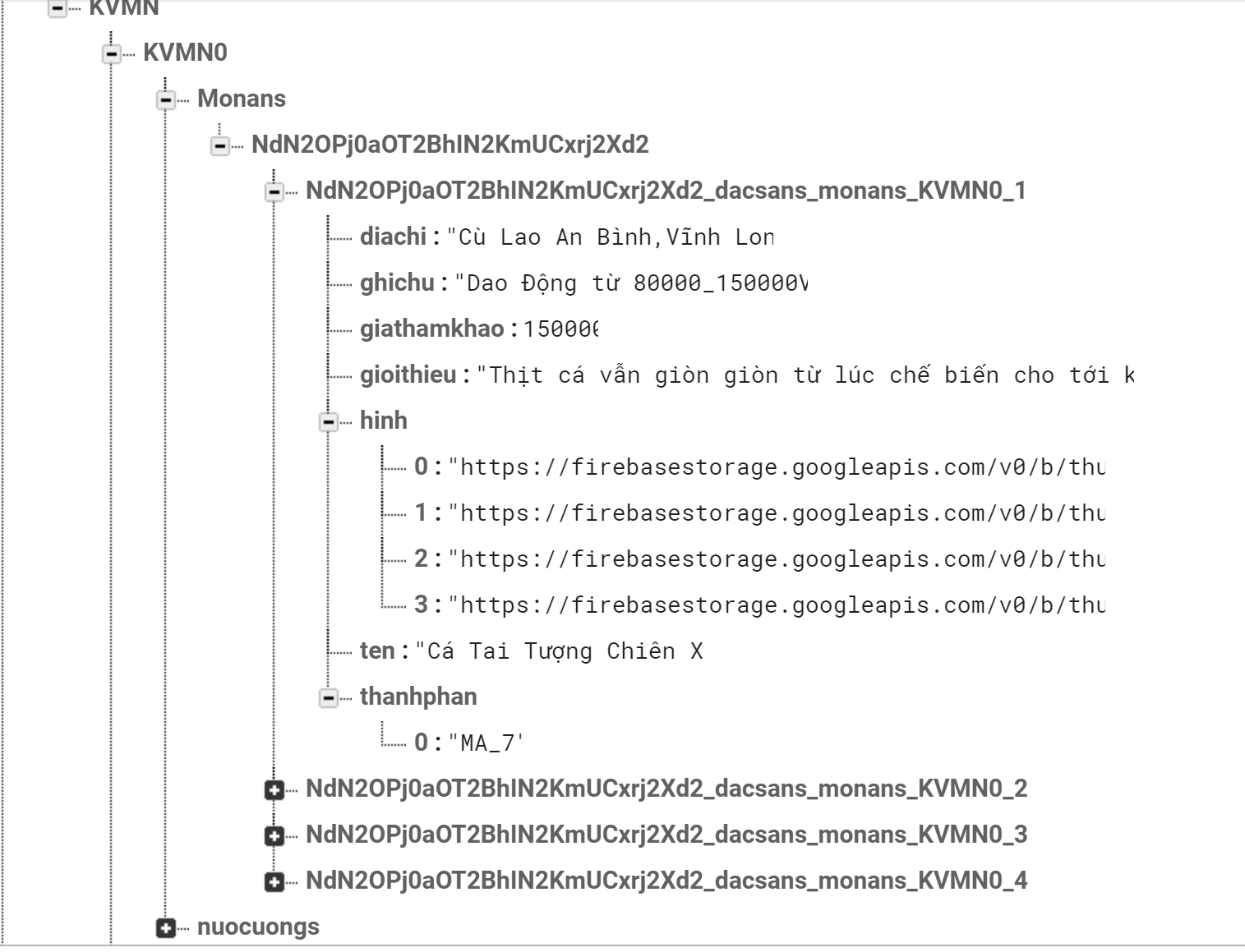
Khi khởi động ứng dụng, sẽ xuất hiện màn hình splash screen sau đó 4s, màng hình sẽ tự động chuyển qua giao diện chọn khu vực, khi người dùng click vào button chọn khu vực một list các khu vực sẽ sổ ra, người dùng chỉ cần chọn khu vực muốn xem. Sau đó click vào Button bắt đầu màn hình sẽ hiện ra giao diện ăn gì sẽ bào gồm các danh sách món ăn có trong database của khu vực đó, khu người dùng click vào button uống gì sẽ xuất hiện giao diện gồm chi tiết các thức uống.Khi người dùng click vào một món ăn bắt kỳ sẽ xuất giao diện chi tiết bài viết. Ở giao diện này sẽ có hai hộp thoại đó là giới thiệu và đánh giá. Hộp thoại giới thiệu sẽ giới thiệu chi tiết về món ăn như thành phần, giá, địa điểm, và gợi ý các món ăn tương tự và có thể lưu món ăn đó. Hộp thoại đánh giá người dùng có thể đánh giá món ăn đó và chấm điểm thông qa số sao. Nhưng nếu muốn đánh giá người dùng sẽ phải đằng nhập bằng việc click vào biểu tưởng hình cửa sổ, ở giao diện này người dùng có thể đăng nhập bằng sđt và có thể đăng ký mới. Nếu người dùng muốn tìm món ăn mình yêu thích có thể click vào icon hình cái kính lúp sau đó nhập tên món ăn mình muốn tìm, hoặc có thể tìm qa khu vực, giá bán. Ngoài ra ứng dụng còn có giao diện các món ăn mà người dùng có đã lưu trước đó. Chỉ có admin mới có quyền tạo một bài viết sau khi đã đăng nhập

## Cơ sở dữ liệu



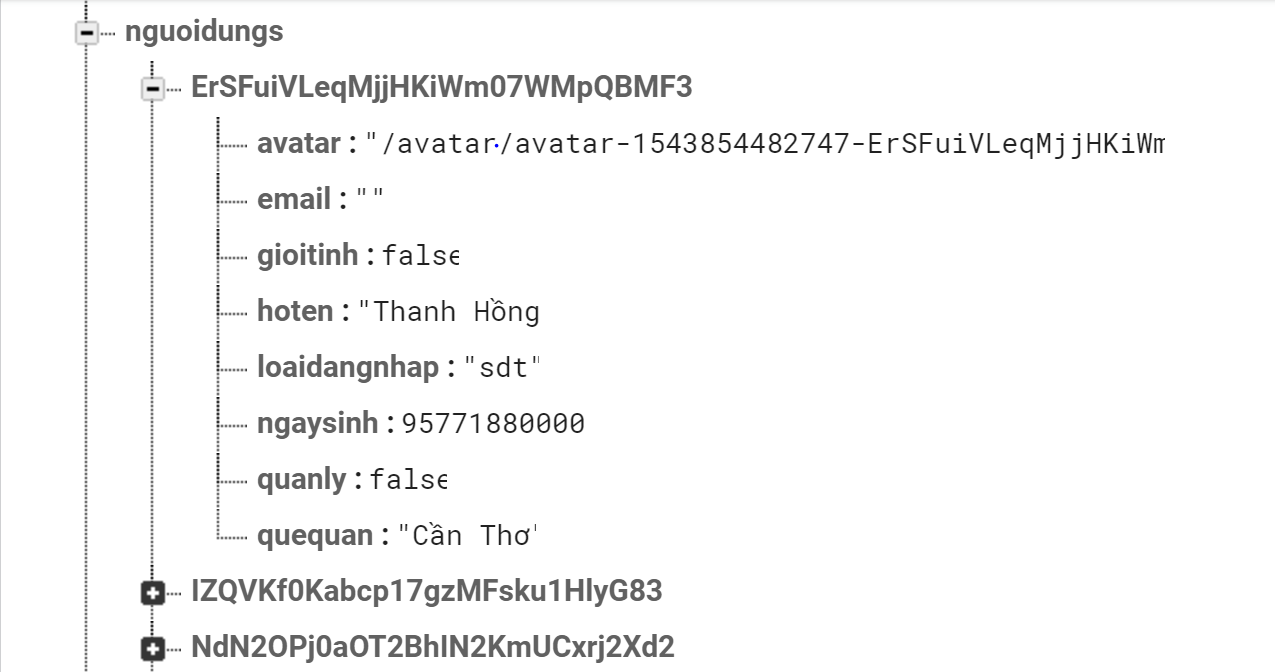
Cơ sở dữ liệu được lưu trưc dưới dạng cây, gồm 7 node để lưu trữ dữ liệu, node đặc sản bao gồm các trường

## Node dacsans

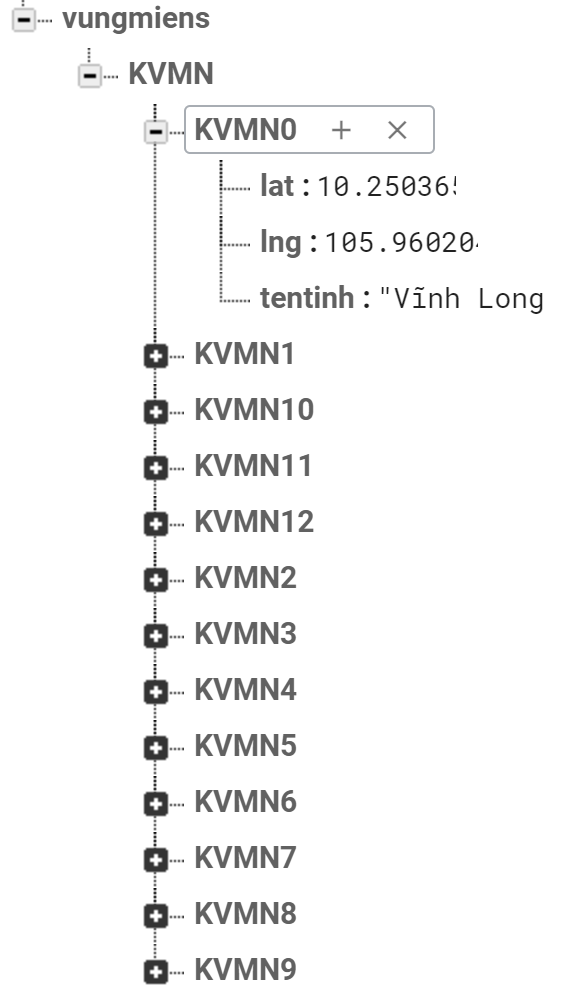


Node dacsans ->KVMN ->các nhánh con (KVMN0-KVMN9) tương ứng với các khu vưc. Trong mỗi nhánh con từ 0 tới 9 đó sẽ có hai node monans và nuocuongs để lưu các giá tri như: địa chỉ, hình ảnh giá tham khảo… như trên hình

1. Node nguoidungs



Node nguoidungs sẽ dduocj quản ký bởi những key động, trong key động này sẽ có các trương để lưu thông tin người đung

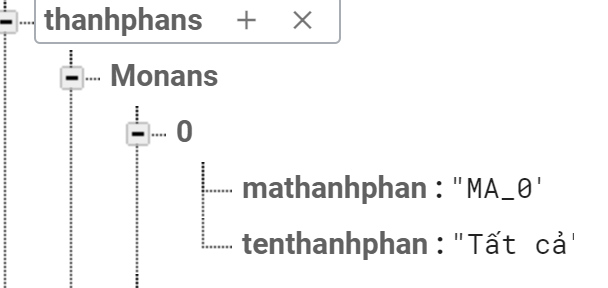


Node vungmien lien hệ vs node dacsans quan các mã khu vực, node này sẽ lưu tên tỉnh thành và vị trí qa thuộc tính kinh độ và vĩ độ

1. Node luulais



Node lưu lại sẽ được quản lý bs cái key động, trong mỗi key động này có node Monans và cũng được quản lý bằng các key động để quản lý mã bài viết và mã khu vực

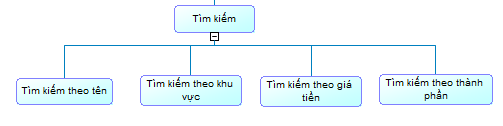


Node thanhphans quản lý các thành phần mathanhphan và tên thành phần, lien hệ vs node dacsan

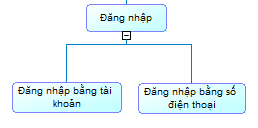
## Sơ đồ phân rã chức năng phần mềm



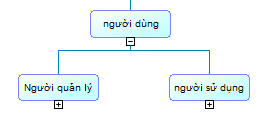
Hình 1. 1 Sơ đồ phân rã chức năng phần mềm



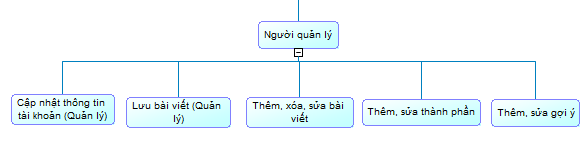
Hình 1. 2 Sơ đồ phân rã chức năng tìm kiếm



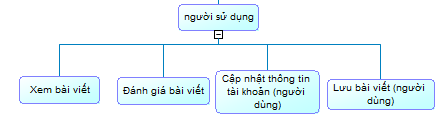
Hình 1. 3 Sơ đồ phân rã chức năng đăng nhập



Hình 1. 4 Sơ đồ phân rã chức năng người dùng



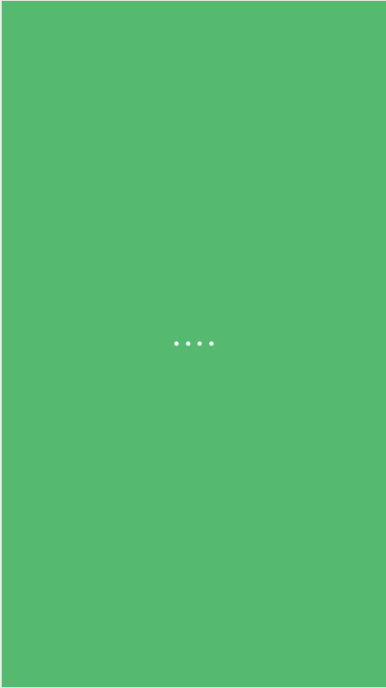
Hình 1. 5 Sơ đồ phân rã chức năng người quản lý



Hình 1. 6 Sơ đồ phân rã chức năng người sử dụng

## 5. Ứng dụng giới thiệu ẩm thực cho smartphone

* ***Màn hình splash screen:***



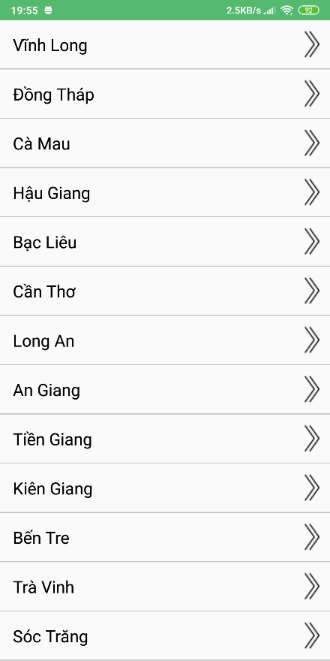
Hình 2. 1 Giao diện splash screen

* ***Màn hình chọn khu vực:***



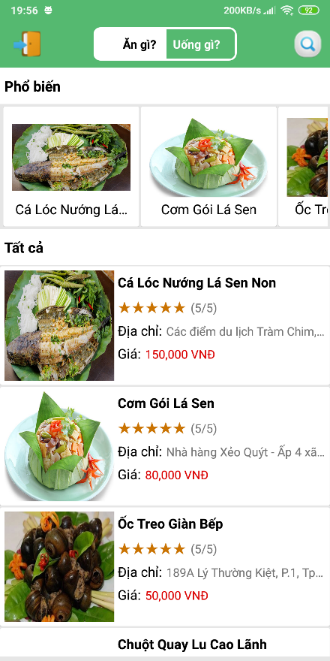
Hình 2.2 Giao diện chọn khu vực

* ***Danh sách các khu vực:***



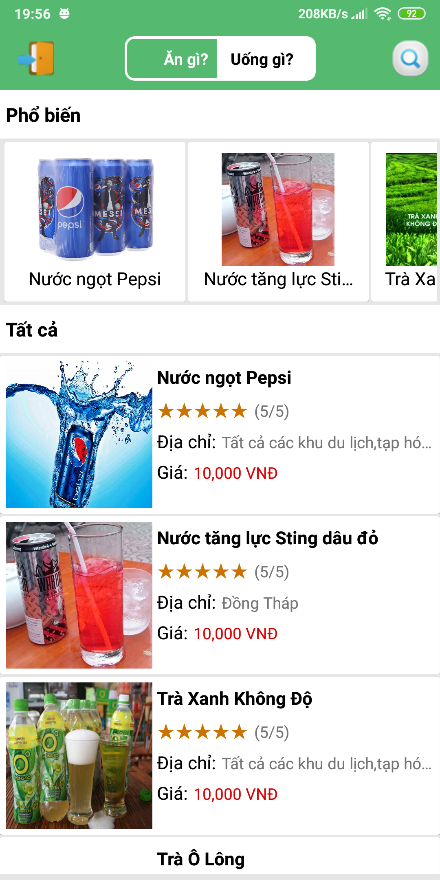
Hình 2.3 Danh sách các khu vực

* ***Giao diện ăn gì:***



Hình 2.4 Giao diện ăn gì

* ***Giao diện uống gì:***



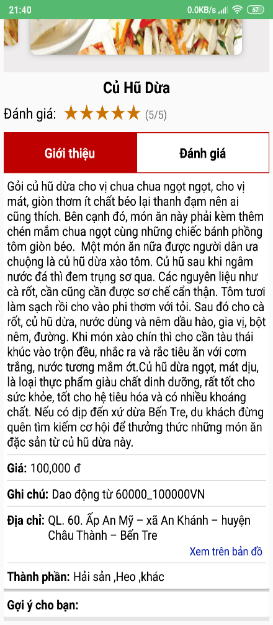
Hình 2.5 Giao diện uông gì

Giao diện “uống gì” danh sách các bài viết về ước uống cho người dùng lựa chọn

* ***Giao diện chi tiết bài viết:***

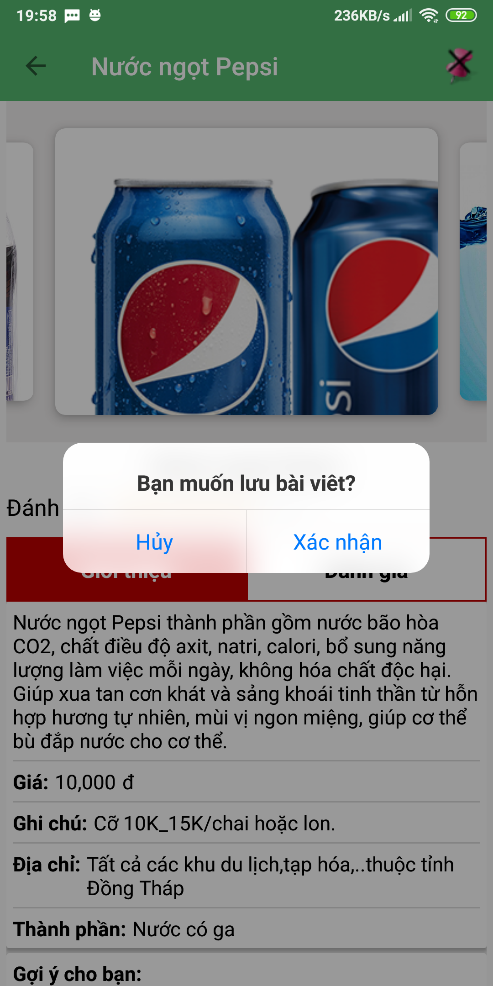


Hình 2.6 Đánh giá bài viết



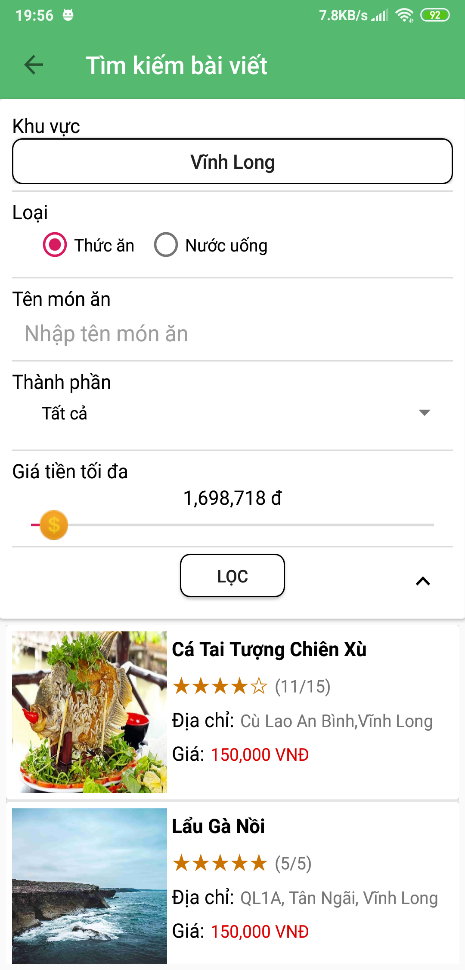
Hình 2.7 Chi tiết bài viết

* Lưu bài viết:



Hình 2.9 Lưu bài viết

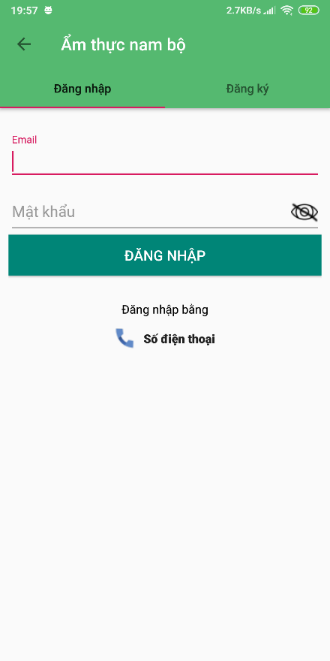
* Tìm bài viết:



Hình 2.10 Tìm bài viết

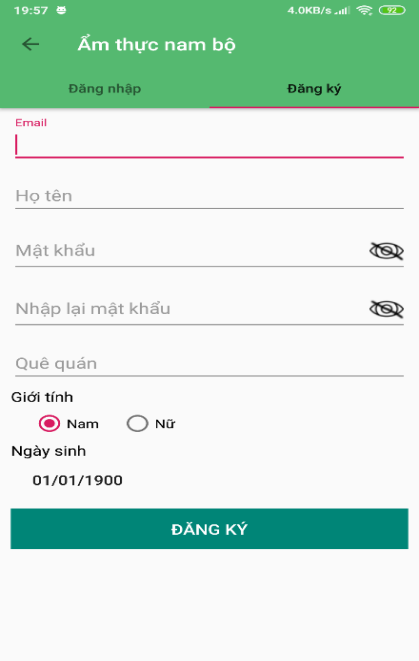
Người dùng có thể tìm bài viết theo tên, khu vực, loại. thành phần và giá tiền tối đa

* ***Giao diện đăng nhập, đăng ký:***



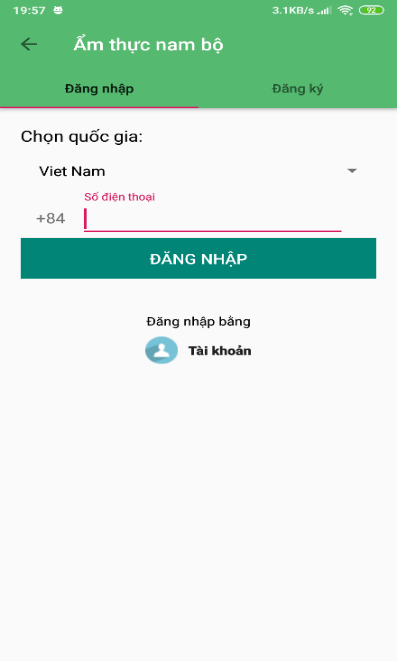
Hình 2.11 Đăng nhập bằng tài khoản

Người dùng đăng nhập hệ thống bằng tài khoản đã đăng ký ở phần mềm để sử dụng một số chức năng như lưu bài viết, thêm bài viết….



Hình 2.12 Đăng ký tài khoản

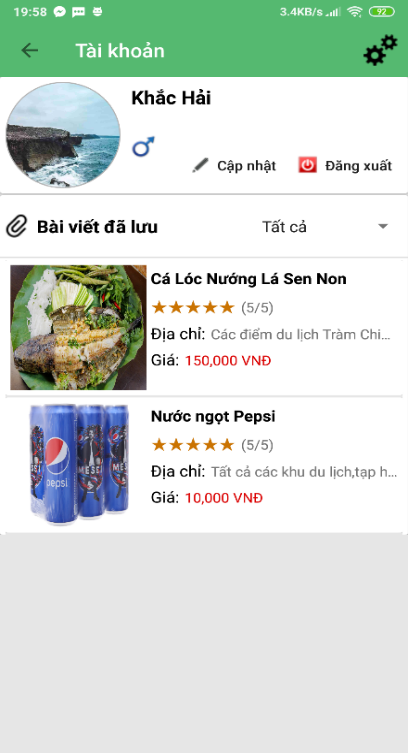
Nếu người dùng chưa có tài khoản thì có thể đăng ký tài khoản để sử dụng các chức năng của phần mềm



Hình 2.13 Đăng nhập bằng số điện thoại

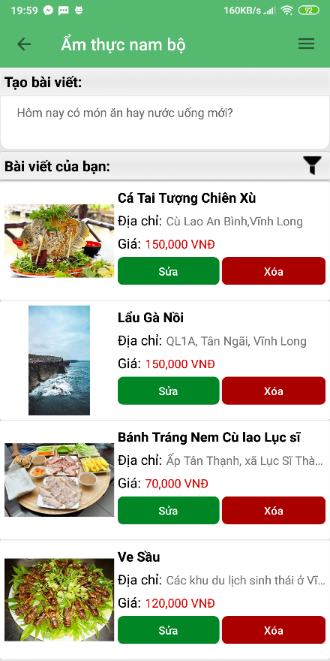
Người dùng có thể đăng nhập bằng số điện thoại để sử dụng phần mềm mà không cần đăng ký

* ***Giao diện người dùng:***

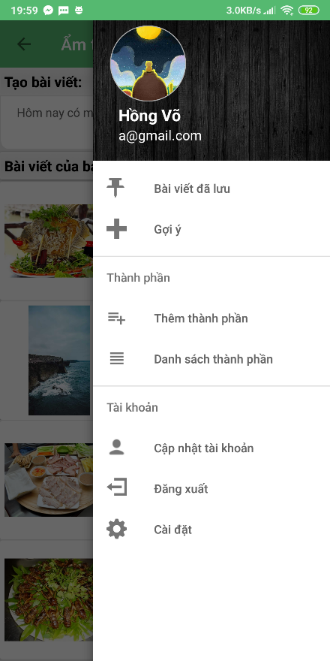


Hình 2.14 Trang cá nhân người dùng

* ***Giao diện người quản lý:***



Hình 2.15 Trang cá nhân người quản lý



Hình 2.16 Menu chức năng người quản lý

* ***Giao diện thêm bài viết mới:***



Hình 2.17 Chức năng thêm bài viết mới



Hình 2.18 Chức năng cập nhật bài viết

# CHƯƠNG 2.Models

## Firebase Models

### 

**public class** objchitietbaiviet **implements** Serializable {  
 String **ten**;  
 String **gioithieu**;  
 String **ghichu**;  
 String **diachi**;  
 **long giathamkhao**;  
 List<String> **hinh**;  
 List<String> **thanhphan**;  
  
 **public** objchitietbaiviet(String ten, String gioithieu, String ghichu, String diachi, **long** giathamkhao, List<String> hinh, List<String> thanhphan) {  
 **this**.**ten** = ten;  
 **this**.**gioithieu** = gioithieu;  
 **this**.**ghichu** = ghichu;  
 **this**.**diachi** = diachi;  
 **this**.**giathamkhao** = giathamkhao;  
 **this**.**hinh** = hinh;  
 **this**.**thanhphan** = thanhphan;  
 }  
  
 **public** objchitietbaiviet() {  
 }  
  
 **public** String getDiachi() {  
 **return diachi**;  
 }  
  
 **public void** setDiachi(String diachi) {  
 **this**.**diachi** = diachi;  
 }  
  
 **public** String getTen() {  
 **return ten**;  
 }  
  
 **public void** setTen(String ten) {  
 **this**.**ten** = ten;  
 }  
  
 **public** String getGioithieu() {  
 **return gioithieu**;  
 }  
  
 **public void** setGioithieu(String gioithieu) {  
 **this**.**gioithieu** = gioithieu;  
 }  
  
 **public** String getGhichu() {  
 **return ghichu**;  
 }  
  
 **public void** setGhichu(String ghichu) {  
 **this**.**ghichu** = ghichu;  
 }  
  
 **public long** getGiathamkhao() {  
 **return giathamkhao**;  
 }  
  
 **public void** setGiathamkhao(**long** giathamkhao) {  
 **this**.**giathamkhao** = giathamkhao;  
 }  
  
  
 **public** List<String> getHinh() {  
 **return hinh**;  
 }  
  
 **public void** setHinh(List<String> hinh) {  
 **this**.**hinh** = hinh;  
 }  
  
 **public** List<String> getThanhphan() {  
 **return thanhphan**;  
 }  
  
 **public void** setThanhphan(List<String> thanhphan) {  
 **this**.**thanhphan** = thanhphan;  
 }  
}

### Objdanhgia

**public class** objdanhgia {  
  
 **long sosao**;  
 String **binhluan**;  
 String **tennguoidanhgia**;  
 **long ngaydanhgia**;  
  
 **public** objdanhgia(**long** sosao, String binhluan, String tennguoidanhgia, **long** ngaydanhgia) {  
 **this**.**sosao** = sosao;  
 **this**.**binhluan** = binhluan;  
 **this**.**tennguoidanhgia** = tennguoidanhgia;  
 **this**.**ngaydanhgia** = ngaydanhgia;  
 }  
  
 **public** String getTennguoidanhgia() {  
 **return tennguoidanhgia**;  
 }  
  
 **public void** setTennguoidanhgia(String tennguoidanhgia) {  
 **this**.**tennguoidanhgia** = tennguoidanhgia;  
 }  
  
 **public long** getNgaydanhgia() {  
 **return ngaydanhgia**;  
 }  
  
 **public void** setNgaydanhgia(**long** ngaydanhgia) {  
 **this**.**ngaydanhgia** = ngaydanhgia;  
 }  
  
 **public** objdanhgia() {  
 }  
  
 **public long** getSosao() {  
 **return sosao**;  
 }  
  
 **public void** setSosao(**long** sosao) {  
 **this**.**sosao** = sosao;  
 }  
  
 **public** String getBinhluan() {  
 **return binhluan**;  
 }  
  
 **public void** setBinhluan(String binhluan) {  
 **this**.**binhluan** = binhluan;  
 }  
}

### Objluubaiviet

**public class** objluubaiviet {  
 String **mabaiviet**;  
 String **makhuvuc**;  
  
 **public** objluubaiviet( String mabaiviet, String makhuvuc) {  
 **this**.**mabaiviet** = mabaiviet;  
 **this**.**makhuvuc** = makhuvuc;  
 }  
  
 **public** objluubaiviet() {  
 }  
  
  
 **public** String getMabaiviet() {  
 **return mabaiviet**;  
 }  
  
 **public void** setMabaiviet(String mabaiviet) {  
 **this**.**mabaiviet** = mabaiviet;  
 }  
  
 **public** String getMakhuvuc() {  
 **return makhuvuc**;  
 }  
  
 **public void** setMakhuvuc(String makhuvuc) {  
 **this**.**makhuvuc** = makhuvuc;  
 }  
}

### Objnguoidungs

**public class** objnguoidungs {  
 **boolean quanly**;  
 String **avatar**;  
 String **email**;  
 Boolean **gioitinh**;  
 String **hoten**;  
 **long ngaysinh**;  
 String **quequan**;  
 String **loaidangnhap**;  
  
 **public** objnguoidungs(**boolean** quanly, String avatar, String email, Boolean gioitinh, String hoten, **long** ngaysinh, String quequan, String loaidangnhap) {  
 **this**.**quanly** = quanly;  
 **this**.**avatar** = avatar;  
 **this**.**email** = email;  
 **this**.**gioitinh** = gioitinh;  
 **this**.**hoten** = hoten;  
 **this**.**ngaysinh** = ngaysinh;  
 **this**.**quequan** = quequan;  
 **this**.**loaidangnhap** = loaidangnhap;  
 }  
  
 **public** objnguoidungs() {  
 }  
  
 **public** String getLoaidangnhap() {  
 **return loaidangnhap**;  
 }  
  
 **public void** setLoaidangnhap(String loaidangnhap) {  
 **this**.**loaidangnhap** = loaidangnhap;  
 }  
  
 **public boolean** isQuanly() {  
 **return quanly**;  
 }  
  
 **public void** setQuanly(**boolean** quanly) {  
 **this**.**quanly** = quanly;  
 }  
  
 **public** String getAvatar() {  
 **return avatar**;  
 }  
  
 **public void** setAvatar(String avatar) {  
 **this**.**avatar** = avatar;  
 }  
  
 **public** String getEmail() {  
 **return email**;  
 }  
  
 **public void** setEmail(String email) {  
 **this**.**email** = email;  
 }  
  
 **public** Boolean getGioitinh() {  
 **return gioitinh**;  
 }  
  
 **public void** setGioitinh(Boolean gioitinh) {  
 **this**.**gioitinh** = gioitinh;  
 }  
  
 **public** String getHoten() {  
 **return hoten**;  
 }  
  
 **public void** setHoten(String hoten) {  
 **this**.**hoten** = hoten;  
 }  
  
 **public long** getNgaysinh() {  
 **return ngaysinh**;  
 }  
  
 **public void** setNgaysinh(**long** ngaysinh) {  
 **this**.**ngaysinh** = ngaysinh;  
 }  
  
 **public** String getQuequan() {  
 **return quequan**;  
 }  
  
 **public void** setQuequan(String quequan) {  
 **this**.**quequan** = quequan;  
 }  
}

### Objthanhphan

**public class** objthanhphan {  
 String **mathanhphan**;  
 String **tenthanhphan**;  
  
 **public** objthanhphan(String mathanhphan, String tenthanhphan) {  
 **this**.**mathanhphan** = mathanhphan;  
 **this**.**tenthanhphan** = tenthanhphan;  
 }  
  
 **public** objthanhphan() {  
 }  
  
 **public** String getMathanhphan() {  
 **return mathanhphan**;  
 }  
  
 **public void** setMathanhphan(String mathanhphan) {  
 **this**.**mathanhphan** = mathanhphan;  
 }  
  
 **public** String getTenthanhphan() {  
 **return tenthanhphan**;  
 }  
  
 **public void** setTenthanhphan(String tenthanhphan) {  
 **this**.**tenthanhphan** = tenthanhphan;  
 }  
}

### Objtinhthanh

**public class** objtinhthanh {  
 String **tentinh**;  
 **double lat**,**lng**;  
  
 **public** objtinhthanh(String tentinh, **double** lat, **double** lng) {  
 **this**.**tentinh** = tentinh;  
 **this**.**lat** = lat;  
 **this**.**lng** = lng;  
 }  
  
 **public** objtinhthanh() {  
 }  
  
 **public** String getTentinh() {  
 **return tentinh**;  
 }  
  
 **public void** setTentinh(String tentinh) {  
 **this**.**tentinh** = tentinh;  
 }  
  
 **public double** getLat() {  
 **return lat**;  
 }  
  
 **public void** setLat(**double** lat) {  
 **this**.**lat** = lat;  
 }  
  
 **public double** getLng() {  
 **return lng**;  
 }  
  
 **public void** setLng(**double** lng) {  
 **this**.**lng** = lng;  
 }  
}

## 2.2 App Models

### Objbaiviet\_app

**public class** objbaiviet\_app **implements** Serializable {  
  
 String **idbaiviet**;  
 objchitietbaiviet **chitiet**;  
  
 **public** objbaiviet\_app(String idbaiviet, objchitietbaiviet chitiet) {  
 **this**.**idbaiviet** = idbaiviet;  
 **this**.**chitiet** = chitiet;  
 }  
  
 **public** objbaiviet\_app() {  
 }  
  
  
 **public** String getIdbaiviet() {  
 **return idbaiviet**;  
 }  
  
 **public void** setIdbaiviet(String idbaiviet) {  
 **this**.**idbaiviet** = idbaiviet;  
 }  
  
 **public** objchitietbaiviet getChitiet() {  
 **return chitiet**;  
 }  
  
 **public void** setChitiet(objchitietbaiviet chitiet) {  
 **this**.**chitiet** = chitiet;  
 }  
}

### Objbinhluan\_app

**public class** objbinhluan\_app {  
 objnguoidungs **nguoidanhgia**;  
 objdanhgia **danhgia**;  
}

### Objdanhgia\_app

**public class** objdanhgia\_app {  
 String **iduser**;  
 objdanhgia **chitietdanhgia**;  
  
 **public** objdanhgia\_app(String iduser, objdanhgia chitietdanhgia) {  
 **this**.**iduser** = iduser;  
 **this**.**chitietdanhgia** = chitietdanhgia;  
 }  
  
 **public** objdanhgia\_app() {  
 }  
  
 **public** String getIduser() {  
 **return iduser**;  
 }  
  
 **public void** setIduser(String iduser) {  
 **this**.**iduser** = iduser;  
 }  
  
 **public** objdanhgia getChitietdanhgia() {  
 **return chitietdanhgia**;  
 }  
  
 **public void** setChitietdanhgia(objdanhgia chitietdanhgia) {  
 **this**.**chitietdanhgia** = chitietdanhgia;  
 }  
}

### Objgoiy\_app

**public class** objgoiy\_app {  
 String **mathanhphan**;  
 List<String> **listgoiy**;  
  
 **public** objgoiy\_app(String mathanhphan, List<String> listgoiy) {  
 **this**.**mathanhphan** = mathanhphan;  
 **this**.**listgoiy** = listgoiy;  
 }  
  
 **public** objgoiy\_app() {  
 }  
  
 **public** String getMathanhphan() {  
 **return mathanhphan**;  
 }  
  
 **public void** setMathanhphan(String mathanhphan) {  
 **this**.**mathanhphan** = mathanhphan;  
 }  
  
 **public** List<String> getListgoiy() {  
 **return listgoiy**;  
 }  
  
 **public void** setListgoiy(List<String> listgoiy) {  
 **this**.**listgoiy** = listgoiy;  
 }  
}

### Objkhuvuc\_app

**public class** objkhuvuc\_app {  
  
 String **makhuvuc**;  
 objtinhthanh **chitietkhuvuc**;  
  
 **public** objkhuvuc\_app(String makhuvuc, objtinhthanh chitietkhuvuc) {  
 **this**.**makhuvuc** = makhuvuc;  
 **this**.**chitietkhuvuc** = chitietkhuvuc;  
 }  
  
 **public** objkhuvuc\_app() {  
 }  
  
 **public** String getMakhuvuc() {  
 **return makhuvuc**;  
 }  
  
 **public void** setMakhuvuc(String makhuvuc) {  
 **this**.**makhuvuc** = makhuvuc;  
 }  
  
 **public** objtinhthanh getChitietkhuvuc() {  
 **return chitietkhuvuc**;  
 }  
  
 **public void** setChitietkhuvuc(objtinhthanh chitietkhuvuc) {  
 **this**.**chitietkhuvuc** = chitietkhuvuc;  
 }  
}

### Objnguoidung\_app

**public class** objnguoidung\_app {  
 objnguoidungs **nguoidung**;  
 String **anhdaidien**;  
 String **password**;  
  
  
 **public** objnguoidung\_app(objnguoidungs nguoidung, String anhdaidien, String password) {  
 **this**.**nguoidung** = nguoidung;  
 **this**.**anhdaidien** = anhdaidien;  
 **this**.**password** = password;  
 }  
  
 **public** objnguoidung\_app() {  
 }  
  
 **public** String getAnhdaidien() {  
 **return anhdaidien**;  
 }  
  
 **public void** setAnhdaidien(String anhdaidien) {  
 **this**.**anhdaidien** = anhdaidien;  
 }  
  
 **public** objnguoidungs getNguoidung() {  
 **return nguoidung**;  
 }  
  
 **public void** setNguoidung(objnguoidungs nguoidung) {  
 **this**.**nguoidung** = nguoidung;  
 }  
  
 **public** String getPassword() {  
 **return password**;  
 }  
  
 **public void** setPassword(String password) {  
 **this**.**password** = password;  
 }  
}

### Objthongkedanhgia\_app

**public class** objthongkedanhgia\_app {  
 String **idbaiviet**;  
 List<objdanhgia\_app> **danhgia**;  
 **float diemdanhgia**;  
  
 **public** objthongkedanhgia\_app(String idbaiviet, List<objdanhgia\_app> danhgia) {  
 **this**.**idbaiviet** = idbaiviet;  
 **this**.**danhgia** = danhgia;  
 **diemdanhgia** = 0;  
 }  
  
 **public float** getDiemdanhgia() {  
 **return diemdanhgia**;  
 }  
  
 **public void** setDiemdanhgia(**float** diemdanhgia) {  
 **this**.**diemdanhgia** = diemdanhgia;  
 }  
  
 **public** objthongkedanhgia\_app() {  
 }  
  
 **public** String getIdbaiviet() {  
 **return idbaiviet**;  
 }  
  
 **public void** setIdbaiviet(String idbaiviet) {  
 **this**.**idbaiviet** = idbaiviet;  
 }  
  
 **public** List<objdanhgia\_app> getDanhgia() {  
 **return danhgia**;  
 }  
  
 **public void** setDanhgia(List<objdanhgia\_app> danhgia) {  
 **this**.**danhgia** = danhgia;  
 }  
}

### Objthucpham\_app

**public class** objthucpham\_app {  
 String **idUser**;  
 List<objbaiviet\_app> **baiviet**;  
  
 **public** objthucpham\_app(String idUser, List<objbaiviet\_app> baiviet) {  
 **this**.**idUser** = idUser;  
 **this**.**baiviet** = baiviet;  
 }  
  
 **public** objthucpham\_app() {  
 }  
  
 **public** String getIdUser() {  
 **return idUser**;  
 }  
  
 **public void** setIdUser(String idUser) {  
 **this**.**idUser** = idUser;  
 }  
  
 **public** List<objbaiviet\_app> getBaiviet() {  
 **return baiviet**;  
 }  
  
 **public void** setBaiviet(List<objbaiviet\_app> baiviet) {  
 **this**.**baiviet** = baiviet;  
 }  
}

## Model Admin

**public class** modelAdmin {  
 **public** String **TAG** = **"modelAdmin"**;  
 FirebaseDatabase **mDatabase**;  
  
 iViewAdmin **mInterface**;  
 FirebaseUser **mUser**;  
 Context **context**;  
Phương thức khởi tạo  
 **public** modelAdmin(iViewAdmin mInterface, Context context) {  
 **this**.**mInterface** = mInterface;  
 **this**.**mDatabase** = FirebaseDatabase.*getInstance*();  
 **this**.**mUser** = FirebaseAuth.*getInstance*().getCurrentUser();  
 **this**.**context** = context;  
 }  
**Hàm getBàiVietCuaToi() lấy về được danh sách các bài viết thông qua UID.**

**Biến mRef sẽ trỏ dến CSDL firebase và trỏ đến KVMN(khu vực miền nam), nó sẽ duyệt qua các thành phố để tìm bài viết của UID.khi đã vào node thành phố thì nó dẽ duyệt Node Monan lấy Key, chi tiết bài viết gán vào vào một đối tượng mới khởi tạo. Sau đó add dối tượng đó vào danh sách bài viết của tôi”Listbaivietcuatoi”**

**public void** getBaiVietCuaToi(){  
 **final** String uid = **mUser**.getUid();  
 **final** DatabaseReference mRef = **mDatabase**.getReference().child(**"dacsans"**).child(**"KVMN"**);  
 mRef.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **if**(dataSnapshot!=**null**){  
 ArrayList<objbaiviet\_app> listbaivietcuatoi = **new** ArrayList<>();  
  
 *//DUYỆT QUA CÁC THÀNH PHỐ TÌM BÀI VIẾT CỦA UID* **for** (DataSnapshot i :dataSnapshot.getChildren()) {  
  
 *//LẶP NODE MONANS* **for**(DataSnapshot iduserbaiviet :i.child(**"monans"**).getChildren()){  
 **if**(iduserbaiviet.getKey().matches(uid)){  
 **for** (DataSnapshot val :iduserbaiviet.getChildren()) {  
 objbaiviet\_app bv = **new** objbaiviet\_app();  
 bv.setIdbaiviet(val.getKey());  
 bv.setChitiet(val.getValue(objchitietbaiviet.**class**));  
 listbaivietcuatoi.add(bv);  
  
 }  
  
 **break**;  
 }  
 }  
  
 *//LẶP NODE NUOCUONGS* **for**(DataSnapshot iduserbaiviet :i.child(**"nuocuongs"**).getChildren()){  
 **if**(iduserbaiviet.getKey().matches(uid)){  
 **for** (DataSnapshot val :iduserbaiviet.getChildren()) {  
 objbaiviet\_app bv = **new** objbaiviet\_app();  
 bv.setIdbaiviet(val.getKey());  
 bv.setChitiet(val.getValue(objchitietbaiviet.**class**));  
 listbaivietcuatoi.add(bv);  
 }  
 **break**;  
 }  
 }  
  
 }*//END FOR THÀNH PHỐ* **mInterface**.result\_baivietcuatoi(listbaivietcuatoi);  
  
 }*//END IF* }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 Log.*e*(**TAG**,databaseError.toString());  
 }  
 });  
 }

**Hàm delede sẽ lấy vị trí bài viết muốn xóa, thông quan biến mRef trỏ tới ID bài viết, rối sử dụng hàm delete() để xóa bài viết**

**public void** deleteBaiViet(**final** String idbaiviet){  
 **final** String data[] = idbaiviet.split(**"\_"**);  
 **final** String type = data[2];  
 String vitri = data[3];  
  
 **final** DatabaseReference mRef = **mDatabase**.getReference()  
 .child(**"dacsans"**)  
 .child(**"KVMN"**)  
 .child(vitri)  
 .child(type)  
 .child(**mUser**.getUid())  
 .child(idbaiviet);  
 mRef.addListenerForSingleValueEvent(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 GenericTypeIndicator<ArrayList<String>> gen = **new** GenericTypeIndicator<ArrayList<String>>(){};  
 ArrayList<String> listHinh = dataSnapshot.child(**"hinh"**).getValue(gen);  
 **for**(String idimage : listHinh){  
 StorageReference str = FirebaseStorage.*getInstance*().getReferenceFromUrl(idimage);  
 str.delete();  
 }  
 mRef.removeValue().addOnSuccessListener(**new** OnSuccessListener<Void>() {  
 @Override  
 **public void** onSuccess(Void aVoid) {  
 **mInterface**.result\_delete(**true**);  
 }  
 }).addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 **mInterface**.result\_delete(**false**);  
 }  
 });  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
  
 }  
 });  
  
  
 }

# CHƯƠNG 3. Utils

## CountryData Class

Class này gồm 2 biến kiểu mảng String countryName để lưu 1 mảng tên các quốc gia và countryAreaCodes để lưu mảng đầu số tương ứng với từng quốc gia

**public class** CountryData {  
 **public static final** String[] ***countryNames*** = {**"Afghanistan"**, **"Albania"**,  
 **"Algeria"**, **"Andorra"**, **"Angola"**, **"Antarctica"**, **"Argentina"**,  
 **"Armenia"**, **"Aruba"**, **"Australia"**, **"Austria"**, **"Azerbaijan"**,  
 **"Bahrain"**, **"Bangladesh"**, **"Belarus"**, **"Belgium"**, **"Belize"**, **"Benin"**,  
 **"Bhutan"**, **"Bolivia"**, **"Bosnia And Herzegovina"**, **"Botswana"**,  
 **"Brazil"**, **"Brunei Darussalam"**, **"Bulgaria"**, **"Burkina Faso"**,  
 **"Myanmar"**, **"Burundi"**, **"Cambodia"**, **"Cameroon"**, **"Canada"**,  
 **"Cape Verde"**, **"Central African Republic"**, **"Chad"**, **"Chile"**, **"China"**,  
 **"Christmas Island"**, **"Cocos (keeling) Islands"**, **"Colombia"**,  
 **"Comoros"**, **"Congo"**, **"Cook Islands"**, **"Costa Rica"**, **"Croatia"**,  
 **"Cuba"**, **"Cyprus"**, **"Czech Republic"**, **"Denmark"**, **"Djibouti"**,  
 **"Timor-leste"**, **"Ecuador"**, **"Egypt"**, **"El Salvador"**,  
 **"Equatorial Guinea"**, **"Eritrea"**, **"Estonia"**, **"Ethiopia"**,  
 **"Falkland Islands (malvinas)"**, **"Faroe Islands"**, **"Fiji"**, **"Finland"**,  
 **"France"**, **"French Polynesia"**, **"Gabon"**, **"Gambia"**, **"Georgia"**,  
 **"Germany"**, **"Ghana"**, **"Gibraltar"**, **"Greece"**, **"Greenland"**,  
 **"Guatemala"**, **"Guinea"**, **"Guinea-bissau"**, **"Guyana"**, **"Haiti"**,  
 **"Honduras"**, **"Hong Kong"**, **"Hungary"**, **"India"**, **"Indonesia"**, **"Iran"**,  
 **"Iraq"**, **"Ireland"**, **"Isle Of Man"**, **"Israel"**, **"Italy"**, **"Ivory Coast"**,  
 **"Jamaica"**, **"Japan"**, **"Jordan"**, **"Kazakhstan"**, **"Kenya"**, **"Kiribati"**,  
 **"Kuwait"**, **"Kyrgyzstan"**, **"Laos"**, **"Latvia"**, **"Lebanon"**, **"Lesotho"**,  
 **"Liberia"**, **"Libya"**, **"Liechtenstein"**, **"Lithuania"**, **"Luxembourg"**,  
 **"Macao"**, **"Macedonia"**, **"Madagascar"**, **"Malawi"**, **"Malaysia"**,  
 **"Maldives"**, **"Mali"**, **"Malta"**, **"Marshall Islands"**, **"Mauritania"**,  
 **"Mauritius"**, **"Mayotte"**, **"Mexico"**, **"Micronesia"**, **"Moldova"**,  
 **"Monaco"**, **"Mongolia"**, **"Montenegro"**, **"Morocco"**, **"Mozambique"**,  
 **"Namibia"**, **"Nauru"**, **"Nepal"**, **"Netherlands"**, **"New Caledonia"**,  
 **"New Zealand"**, **"Nicaragua"**, **"Niger"**, **"Nigeria"**, **"Niue"**, **"Korea"**,  
 **"Norway"**, **"Oman"**, **"Pakistan"**, **"Palau"**, **"Panama"**,  
 **"Papua New Guinea"**, **"Paraguay"**, **"Peru"**, **"Philippines"**, **"Pitcairn"**,  
 **"Poland"**, **"Portugal"**, **"Puerto Rico"**, **"Qatar"**, **"Romania"**,  
 **"Russian Federation"**, **"Rwanda"**, **"Saint Barthélemy"**, **"Samoa"**,  
 **"San Marino"**, **"Sao Tome And Principe"**, **"Saudi Arabia"**, **"Senegal"**,  
 **"Serbia"**, **"Seychelles"**, **"Sierra Leone"**, **"Singapore"**, **"Slovakia"**,  
 **"Slovenia"**, **"Solomon Islands"**, **"Somalia"**, **"South Africa"**,  
 **"Korea, Republic Of"**, **"Spain"**, **"Sri Lanka"**, **"Saint Helena"**,  
 **"Saint Pierre And Miquelon"**, **"Sudan"**, **"Suriname"**, **"Swaziland"**,  
 **"Sweden"**, **"Switzerland"**, **"Syrian Arab Republic"**, **"Taiwan"**,  
 **"Tajikistan"**, **"Tanzania"**, **"Thailand"**, **"Togo"**, **"Tokelau"**, **"Tonga"**,  
 **"Tunisia"**, **"Turkey"**, **"Turkmenistan"**, **"Tuvalu"**,  
 **"United Arab Emirates"**, **"Uganda"**, **"United Kingdom"**, **"Ukraine"**,  
 **"Uruguay"**, **"United States"**, **"Uzbekistan"**, **"Vanuatu"**,  
 **"Holy See (vatican City State)"**, **"Venezuela"**, **"Viet Nam"**,  
 **"Wallis And Futuna"**, **"Yemen"**, **"Zambia"**, **"Zimbabwe"**};  
  
 **public static final** String[] ***countryAreaCodes*** = {**"93"**, **"355"**, **"213"**,  
 **"376"**, **"244"**, **"672"**, **"54"**, **"374"**, **"297"**, **"61"**, **"43"**, **"994"**, **"973"**,  
 **"880"**, **"375"**, **"32"**, **"501"**, **"229"**, **"975"**, **"591"**, **"387"**, **"267"**, **"55"**,  
 **"673"**, **"359"**, **"226"**, **"95"**, **"257"**, **"855"**, **"237"**, **"1"**, **"238"**, **"236"**,  
 **"235"**, **"56"**, **"86"**, **"61"**, **"61"**, **"57"**, **"269"**, **"242"**, **"682"**, **"506"**,  
 **"385"**, **"53"**, **"357"**, **"420"**, **"45"**, **"253"**, **"670"**, **"593"**, **"20"**, **"503"**,  
 **"240"**, **"291"**, **"372"**, **"251"**, **"500"**, **"298"**, **"679"**, **"358"**, **"33"**,  
 **"689"**, **"241"**, **"220"**, **"995"**, **"49"**, **"233"**, **"350"**, **"30"**, **"299"**, **"502"**,  
 **"224"**, **"245"**, **"592"**, **"509"**, **"504"**, **"852"**, **"36"**, **"91"**, **"62"**, **"98"**,  
 **"964"**, **"353"**, **"44"**, **"972"**, **"39"**, **"225"**, **"1876"**, **"81"**, **"962"**, **"7"**,  
 **"254"**, **"686"**, **"965"**, **"996"**, **"856"**, **"371"**, **"961"**, **"266"**, **"231"**,  
 **"218"**, **"423"**, **"370"**, **"352"**, **"853"**, **"389"**, **"261"**, **"265"**, **"60"**,  
 **"960"**, **"223"**, **"356"**, **"692"**, **"222"**, **"230"**, **"262"**, **"52"**, **"691"**,  
 **"373"**, **"377"**, **"976"**, **"382"**, **"212"**, **"258"**, **"264"**, **"674"**, **"977"**,  
 **"31"**, **"687"**, **"64"**, **"505"**, **"227"**, **"234"**, **"683"**, **"850"**, **"47"**, **"968"**,  
 **"92"**, **"680"**, **"507"**, **"675"**, **"595"**, **"51"**, **"63"**, **"870"**, **"48"**, **"351"**,  
 **"1"**, **"974"**, **"40"**, **"7"**, **"250"**, **"590"**, **"685"**, **"378"**, **"239"**, **"966"**,  
 **"221"**, **"381"**, **"248"**, **"232"**, **"65"**, **"421"**, **"386"**, **"677"**, **"252"**, **"27"**,  
 **"82"**, **"34"**, **"94"**, **"290"**, **"508"**, **"249"**, **"597"**, **"268"**, **"46"**, **"41"**,  
 **"963"**, **"886"**, **"992"**, **"255"**, **"66"**, **"228"**, **"690"**, **"676"**, **"216"**, **"90"**,  
 **"993"**, **"688"**, **"971"**, **"256"**, **"44"**, **"380"**, **"598"**, **"1"**, **"998"**, **"678"**,  
 **"39"**, **"58"**, **"84"**, **"681"**, **"967"**, **"260"**, **"263"**};  
}

## getDataApp

Class này có nhiệm vụ lấy dữ liệu cho App

**public** getDataApp(Context context) {  
 **this**.**context** = context;  
}

Hàm getListThanhPhan sẽ truyền vào một key kiểu String và trả vể một danh sách các thành phần (objthanhphan :Firebase). Hàm dùng đối tượng SharePreferences(spf) để lưu trữ và đọc dữ liệu và chuyển dữ liệu về kiểu json  
  
**public** ArrayList<objthanhphan> getListThanhPhan(String key){  
 *// lưu trữ và đọc dữ liệu* SharedPreferences spf = **context**.getSharedPreferences(**"thanhphan"**,Context.***MODE\_PRIVATE***);  
 String data = spf.getString(key,**null**);  
 *//tạo đối tượng gson để chuyển data thanh json* Gson gson = **new** Gson();  
 **if**(data!=**null**) {  
 *//tạo kiểu dữ diêu Token* Type type = **new** TypeToken<ArrayList<objthanhphan>>(){}.getType();  
 **return** gson.fromJson(data,type);  
 }**else**{  
 **return null**;  
 }  
}

Hàm getListKhuVuc sẽ truyền vào một key kiểu String và trả vể một danh sách các khuvuc (objkhuvuc :Firebase). Hàm dùng đối tượng SharePreferences(spf) để lưu trữ và đọc dữ liệu và chuyển dữ liệu về kiểu json  
  
  
**public** ArrayList<objkhuvuc\_app> getListKhuVuc(){  
 *//spf lưu và đọc dữ liệu* SharedPreferences spf = **context**.getSharedPreferences(**"data\_app"**,Context.***MODE\_PRIVATE***);  
 *//Lưu dưới dạng string vs tên khuvuc* String data = spf.getString(**"khuvuc"**,**null**);  
 *//tạo đối tượng gson để chuyển data thanh json* Gson gson = **new** Gson();  
 **if**(data!=**null**) {  
 *//tạo kiểu dữ diêu Token* Type type = **new** TypeToken<ArrayList<objkhuvuc\_app>>(){}.getType();  
 **return** gson.fromJson(data,type);  
 }**else**{  
 **return null**;  
 }  
}

## DowloadImage

**public class** downloadImage {  
 **public** String **TAG** = **"downloadImage"**;  
 FirebaseStorage **storage**;  
  
 **private static** downloadImage *instance*;  
  
  
 **public static** downloadImage getInstance() {  
 **if**(*instance*==**null**)  
 *instance* = **new** downloadImage();  
 **return** *instance*;  
 }  
  
 **private** downloadImage() {  
  
 **this**.**storage** = FirebaseStorage.*getInstance*(**"gs://thucannambo.appspot.com"**);  
 }

Hàm dowloadImage() truyền vào một đường dẫn(url) và một đối tượng ImageView trong layout. Hàm dung đối tượng StorageRefence(sRef) để lưu trữ hình ảnh trong firebase. Nếu kết nối thành công sẽ tạo một đối tượng bitmap lưu trữ dưới dạng byte, tạo một đối tượng drawable để đổ dữ liệu từ đối tượng bitmap vào sau đó set đối tượng ImageView đã truyền vào từ đầu để xuất ra ảnh. Nếu thất bại sẽ set Image màu xám  
  
 **public void** dowloadImage(String url, **final** ImageView imageView){  
 **try** {  
 *// Đối tượng lưu trữ hình ảnh trong firebase* StorageReference sRef = **storage**.getReference().child(url);  
 **final long** ONE\_MEGABYTE = 1024 \* 1024;  
 *// lắng nghe sự kiện* sRef.getBytes(ONE\_MEGABYTE).addOnSuccessListener(**new** OnSuccessListener<**byte**[]>() {  
 @Override  
 *// thành công* **public void** onSuccess(**byte**[] bytes) {  
 *// tạo đối tượng bitmap để lưu mảng dưới dạng byte* Bitmap bm = BitmapFactory.*decodeByteArray*(bytes, 0, bytes.**length**);  
 *// tạo đối tượng ảnh và truyền đối tượng bitmap để tạo ảnh* Drawable drawable = **new** BitmapDrawable(Resources.*getSystem*(), bm);  
 *// set drawable vào đối tượng imageview đã truyền vào* imageView.setImageDrawable(drawable);  
 }  
 }) *// thất bại* .addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Log.*e*(**TAG**, e.toString());  
 *//set Image màu xám cho image truyền vào và đoạn text"dowload Image"* imageView.setImageResource(R.color.***colorErrorImage***);  
 }  
 });  
 }**catch** (Exception e){  
 Log.*e*(**TAG**,e.toString());  
 imageView.setImageResource(R.color.***colorErrorImage***);  
 }  
 }  
  
  
}

## CustomDiaLogLoading

Class này chỉ có chức năng tùy chỉnh lại giao diện của DiaLog và có hàm chức năng showDialog() truyền vào một dòng text muốn hiển thị dùng để hiện thị dialog và dismissDialog() dùng để bỏ qua dialog

**public void** showDialog(String mess){  
 **if**(**progressDialog**!=**null**) {  
 *//set View cho Dialog* **progressDialog**.setContentView(R.layout.***custom\_dialog\_layout***);  
 *// set thuộc tính dialog bị cancel* **progressDialog**.setCancelable(**false**);  
 *// set nền* **progressDialog**.getWindow().setBackgroundDrawableResource(android.R.color.***transparent***);  
 *// ánh xạ textview tới textview của layout\_custom\_dialog* TextView msg = (TextView) **progressDialog**.findViewById(R.id.***id\_tv\_loadingmsg***);  
 *// set thông điệp cho dialog* msg.setText(mess);  
 *// hiện dialog* **progressDialog**.show();  
 }**else** {  
 *// khoi tao 1 dialog* **progressDialog** = **new** Dialog(**context**, R.style.***progress\_dialog***);  
 *//set View cho dialog* **progressDialog**.setContentView(R.layout.***custom\_dialog\_layout***);  
 *// set thuộc tính dialog bị cancel* **progressDialog**.setCancelable(**false**);  
 *// set thuộc tính nền* **progressDialog**.getWindow().setBackgroundDrawableResource(android.R.color.***transparent***);  
 *// ánh xa textView* TextView msg = (TextView) **progressDialog**.findViewById(R.id.***id\_tv\_loadingmsg***);  
 *// set đoạn thông điệp cho textView* msg.setText(mess);  
 **progressDialog**.show();  
 }  
}  
*// bỏ qua dialog***public void** dismissDialog(){  
 **if**(**progressDialog**!=**null**)  
 **progressDialog**.dismiss();  
}

# CHƯƠNG 4. View

## 4.2 FplashScreen

### SplashScreenActivity

Activity này kế thừa AppcompatActivity và interface igetdataapp

**public class** SplashScreenActivity **extends** AppCompatActivity **implements** igetdataapp {  
  
 lgetdataapp **mGetData**;  
 LVCircularJump **mLVCircularJump**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_splat\_screen***);  
 *//khởi tạo đối tượng mGetData* **mGetData** = **new** lgetdataapp(**this**,**this**);  
 *// lấy hàm thành phần* **mGetData**.getthanhphan();  
 *//ánh xạ đối tượng mLVCcircularJump sang giao diện activity\_splat\_screen* **mLVCircularJump** = findViewById(R.id.***lv\_CircularJump***);  
 *// set màu cho đối tượng* **mLVCircularJump**.setViewColor(Color.***WHITE***);  
 *// chạy hiệu ứng* **mLVCircularJump**.startAnim();  
  
 }  
Hàm showAlert() truyền vào biến kiểu string để set tiêu đề cho dialog, nhiêm vụ là hiển thị dialog  
  
 **private void** showAlert(String title){  
 *//khởi tạo dialog, tạo giao diện cho dialog* AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 *//set tiêu đề hiện thị trên dialog* builder.setMessage(title);  
 *// khi người dùng click vào bên ngoài dialog thì dialog ko biến mất* builder.setCancelable(**false**);  
 *// thêm vào dialog button Ok, khi click thì sẽ đóng dialog lại* builder.setNegativeButton(**"OK"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 System.*exit*(0);  
 }  
 });  
 *// gắn dao diện dialog bằng thằng builder* Dialog dialog = builder.create();  
 *// show dialog* dialog.show();  
 }  
  
 @Override  
 **public void** loadkhuvuc(**boolean** isSuccess) {  
  
 }  
  
 @Override  
 **public void** saveMyData(**boolean** isSuccess, objkhuvuc\_app mdata) {  
  
 }  
  
 @Override  
 **public void** loitaidulieu\_getDataApp() {  
 showAlert(getText(R.string.***loitaidulieukhoidonglai***).toString());  
 }  
Hàm taidulieuthanhcong là methor của interface igetdataapp. Phương thức này truyền vào một biến kiểu boolean. Hàm có chức năng chuyển từ dao diện splashcreen thì dao diện của IndexActivity sau 5s

@Override  
 **public void** taidulieuthanhcong(**boolean** isRealTime) {  
 **if**(!isRealTime) {  
 *//khởi tạo đối tượng handler* Handler handler = **new** Handler();  
 *// Sau 5 giây thì sẽ chuyển sang Activity Index* handler.postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Intent intent = **new** Intent(SplashScreenActivity.**this**, IndexActivity.**class**);  
 startActivity(intent);  
 finish();  
 }  
 }, 500);  
 }  
 }  
  
  
  
}

### IndexActivity

Activity này kế thừa AppcompatActivity và interface igetdataapp

**public class** IndexActivity **extends** AppCompatActivity **implements** igetdataapp {  
  
 Button **btnbatdau**,**btnchonkhuvuc**;  
 lgetdataapp **mKhuVuc**;  
 CustomDialogLoading **mDialogLoading**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_index***);  
  
 initView();  
  
 actionChonKhuVuc();  
 actionStart();  
 checkDataOld();  
 }  
  
 *Hàm checkDataOld sẽ kiểm tra xem đã có dữ liệu trước đó hay chưa nếu đã có rồi thì sẽ dùng dữ liệu cũ* **private void** checkDataOld() {  
 *// khởi tạo đối tượng objkv v* objkhuvuc\_app objkv = **mKhuVuc**.getmykhuvuc();  
 *// nếu đối tượng có giá trị* **if**(objkv!=**null**){  
 *// sẽ set đoạn text cho button là tên tỉnh đã chọn* **btnchonkhuvuc**.setText(objkv.getChitietkhuvuc().getTentinh());  
 *// có thể click vào button* **btnbatdau**.setEnabled(**true**);  
 *// hiển thị button* **btnbatdau**.setVisibility(View.***VISIBLE***);  
 }  
 }  
 Hàm actionStart() dùng để chuyển dao diện từ IndexActivity sang MainActivity

**private void** actionStart() {  
 **btnbatdau**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent(IndexActivity.**this**,MainActivity.**class**);  
 startActivity(intent);  
 }  
 });  
 }

Hàm actionChonKhuVuc() dùng để lắng nghe sự kiện click vào button chọn khu vực, khi click thì sẽ xuất hiện một dialog cho cho người dùng biết  
  
 **private void** actionChonKhuVuc() {  
 **btnchonkhuvuc**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 *//show dialoag cho người dùng* **mDialogLoading**.showDialog(getText(R.string.***dangtaidulieu***).toString());  
 *// lấy khu vực với key"KVMN"* **mKhuVuc**.getKhuVuc(**"KVMN"**);  
 }  
 });  
 }  
  
Hàm initView() dùng để khởi tạo dao diện cho Dialog (mDialogLoading), button(btnKhuVuc, btnchonkhuvuc)  
  
 **private void** initView() {  
 *// khởi tạo đội tượng mDialogLoading* **mDialogLoading** = **new** CustomDialogLoading(**this**);  
 *// khởi tạo biến mKhuVuc* **mKhuVuc** = **new** lgetdataapp(**this**,**this**);  
 *// ánh xạ Button sang button Bat dau giao diện* **btnbatdau** = findViewById(R.id.***btnbatdau***);  
 *// ánh xạ button sang button chontinhthanh* **btnchonkhuvuc** = findViewById(R.id.***btnchontinhthanh***);  
 *// khởi tạo đối tượng Bundle và lấy dữ liệu* Bundle bundle = getIntent().getBundleExtra(**"data"**);  
 *// nếu mà buble khác rỗng thì sẽ set btnkhuvuc = tenkhuvuc* **if**(bundle!=**null**){  
 **btnchonkhuvuc**.setText(bundle.getString(**"tenkhuvuc"**));  
 }  
 *//nếu phần text của button trùng với khu vực* **if**(**btnchonkhuvuc**.getText().toString().matches(getText(R.string.***chonvitri***).toString())){  
 *// biến mất button* **btnbatdau**.setVisibility(View.***INVISIBLE***);  
 *// không cho phép click vào button* **btnbatdau**.setEnabled(**false**);  
 }**else**{  
 *// có thể click vào button* **btnbatdau**.setEnabled(**true**);  
 *// hiển thị button* **btnbatdau**.setVisibility(View.***VISIBLE***);  
 }  
 }  
Hàm loadkhuvuc() là methor của interface igetdataapp kiểm tra xem dữ liệu có load thành công hay ko, nếu thành công thì sẽ chuyển sang layout chọn khuvuc, nếu không sẽ xuât đoạn text cho người dùng biết

@Override  
 **public void** loadkhuvuc(**boolean** isSuccess) {  
 *// bỏ qua dialog* **mDialogLoading**.dismissDialog();  
 *// nếu biến isSuccess = true thì sẽ chuyển dai diện sang SelectKhuVucActivity* **if**(isSuccess){  
 Intent intent = **new** Intent(**this**,SelectKhuVucActivity.**class**);  
 startActivity(intent);  
  
 }**else**{  
 *// xuất xòng text cho người dùng xem để biết load khu vực failse* Toast.*makeText*(**this**, getText(R.string.***taidulieukhongthanhcong***), Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
 @Override  
 **public void** saveMyData(**boolean** isSuccess, objkhuvuc\_app mdata) {  
  
 }  
  
 @Override  
 **public void** loitaidulieu\_getDataApp() {  
  
 }  
  
  
  
 @Override  
 **public void** taidulieuthanhcong( **boolean** i) {  
  
 }  
  
  
}

### SelectKhuVucActivity

Activity này kế thừa AppcompatActivity và interface igetdataapp

**public class** SelectKhuVucActivity **extends** AppCompatActivity **implements** igetdataapp {  
  
 RecyclerView **rclvKhuVuc**;  
 aRclvChonKhuVuc **adapter**;  
 ArrayList<objkhuvuc\_app> **mList**;  
  
 lgetdataapp **mKV**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_select\_khu\_vuc***);  
 initView();  
 }  
  
Hàm initView() khởi tạo View dao diện cho RecyclerView và sẽ tạo và gắn mảng dánh sách khu vực về dạng Json. Sau đó gọi hàm setupRecyclerView() để tạo giao diện  
  
 **private void** initView(){  
 *//Ánh xạ biến RecylerView sang giao diện* **rclvKhuVuc** = findViewById(R.id.***rcrkhuvuc***);  
 *// khởi tạo đối tượng* **mKV** = **new** lgetdataapp(**this**,**this**);  
 *// khởi tạo đối tượng lưu trữ dữ liệu* SharedPreferences preferences = getSharedPreferences(**"data\_app"**,***MODE\_PRIVATE***);  
 *// khởi tạo biến string và gán các giá trị với key khuvuc* String jsonkhuvuc = preferences.getString(**"khuvuc"**,**null**);  
 *// nếu biến jsonkhuvuc không rỗng* **if**(jsonkhuvuc!=**null**){  
 *//tạo đối tượng gson* Gson gson = **new** Gson();  
 *// khởi tạo một type kiểu token cho mản danh sách các đối tượng khu vực* Type listtype = **new** TypeToken<ArrayList<objkhuvuc\_app>>(){}.getType();  
 *// gán cho mlist thành json( truyền vào text và kiểu danh sách)* **mList** = gson.fromJson(jsonkhuvuc,listtype);  
 *//goi hàm* setupRecyclerView();  
 }**else**{  
 Toast.*makeText*(**this**, getText(R.string.***loikhuvuc***), Toast.***LENGTH\_SHORT***).show();  
 finish();  
 }  
 }  
Hàm setupRecyclerView() cài đặt một số thành phần dao diện và truyền dữ liệu, lưu lại dữ liêu vào danh sách  
 **private void** setupRecyclerView() {  
 *// khởi tạo adapter* **adapter** = **new** aRclvChonKhuVuc(**this**,**mList**);  
 *// hởi tạo đối tượng linearlayoutmanager* LinearLayoutManager manager = **new** LinearLayoutManager((Context) **this**,LinearLayoutManager.***VERTICAL***,**false**);  
 *// tạo size tương đối* **rclvKhuVuc**.hasFixedSize();  
 *// set recylerView kiểu Linear layout* **rclvKhuVuc**.setLayoutManager(manager);  
 *// truyền adapter vào recyclerView* **rclvKhuVuc**.setAdapter(**adapter**);  
 *// lắng nghe sự kiện click cho từng item* **adapter**.setOnClickListener(**new** aRclvChonKhuVuc.OnClickListener() {  
 @Override  
 **public void** OnClick(View itemView, **int** position) {  
 *// lưu dữ liệu vào danh sách có vị trí* **mKV**.savemykhuvuc(**mList**.get(position));  
 }  
 });  
 }  
  
 @Override  
 **public void** loadkhuvuc(**boolean** isSuccess) {  
  
 }  
Hàm saveMyData() dùng để lưu dữ liệu và chuyển về giao diện index và một methor của Interface igetdata  
 @Override  
 **public void** saveMyData(**boolean** isSuccess, objkhuvuc\_app mdata) {  
 **if**(isSuccess) {  
 *// tạo đối tượng intent để chuyển sang giao diện indexactivity* Intent intent = **new** Intent(**this**, IndexActivity.**class**);  
 *// gán cờ cho intent và xóa các activity trong stack, để nó không quay về activity cũ* intent.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK*** | Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 *// khởi tạo bundle* Bundle bundle = **new** Bundle();  
 *// thêm tên khu vực vào bundle với key là tenkhuvuc* bundle.putString(**"tenkhuvuc"**,mdata.getChitietkhuvuc().getTentinh());  
 *// thêm dữ liệu vào itent vớ key là data* intent.putExtra(**"data"**,bundle);  
 *// chạy itent để chuyển trang* startActivity(intent);  
 finish();  
 }**else** Toast.*makeText*(**this**, getText(R.string.***loikhuvuc***), Toast.***LENGTH\_SHORT***).show();  
 }  
  
 @Override  
 **public void** loitaidulieu\_getDataApp() {  
  
 }  
  
  
 @Override  
 **public void** taidulieuthanhcong(**boolean** i) {  
  
 }

## Main

### MainActivity

Class MainActivity có dao diện là layout activity\_main và kế thừa interface ilogin. Class này có chức năng phân quyền account là admin hoặc là account bình thường thông qua hai hàm là initView() và phanquyen();

**public class** MainActivity **extends** AppCompatActivity **implements** ilogin {  
  
 aViewPagerMenu **adapter**;  
 ViewPager **vpMain**;  
 RadioButton **rdoangi**,**rdouonggi**;  
 ImageView **imvSearch**,**imvLogin**,**imvUser**;  
  
 llogin **mLogin**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 initView();  
  
 }  
  
 @Override  
 **protected void** onResume() {  
 **super**.onResume();  
 phanquyen();  
 }  
  
Hàm phanquyen() có chức năng kết nối tới firebasedatabase để lấy ra uesr, nếu tồn tại user thì ẩn hình của sổ đi và hiện lên ảnhđại diện của account đã đăng ký, nếu acccount ko có ảnh thid dùng ảnh mặc định.kkhi clck vào ảnh user nếu là admin thì sẽ chuyển sang dao diện admin, nêu người dùng bình thường thì sẽ chuyển tới dao diên người dùng. Còn nếu ko tồn tại user sẽ chuyên sang dao diện đăng kí  
  
 **private void** phanquyen() {  
 *// khởi tạo và kết nối tới firebase uer* FirebaseUser user = FirebaseAuth.*getInstance*().getCurrentUser();  
 *// nếu tồn tại uer* **if**(user!=**null**){  
 *// ẩn image đi* **imvLogin**.setVisibility(View.***GONE***);  
 *// hiện image* **imvUser**.setVisibility(View.***VISIBLE***);  
 *//nếu tồn tại ảnh người dùng* **if**(**mLogin**.getImage(**mLogin**.getDataNguoiDung().getAnhdaidien())!=**null**)  
 *// set ảnh đó làm ảnh của user đó lên app* **imvUser**.setImageDrawable(**mLogin**.getImage(**mLogin**.getDataNguoiDung().getAnhdaidien()));  
 **else** *// nếu ko có ảnh đại diện thì set ảnh mặt định* **imvUser**.setImageResource(R.drawable.***defaultuser***);  
 *// lắng nghe sự kiện click vào ảnh user* **imvUser**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 *// nếu là không pai là admin* **if**(!**mLogin**.getDataNguoiDung().getNguoidung().isQuanly())  
 *// chuyển sang dao diện của taikhoanmainActivity* startActivity(**new** Intent(MainActivity.**this**,TaiKhoanMainActivity.**class**));  
 **else** *// chuyển sang dao diện của AdminActivity* startActivity(**new** Intent(MainActivity.**this**,AdminActivity.**class**));  
 }  
 });  
 }**else**{  
 *// hiện imageview* **imvLogin**.setVisibility(View.***VISIBLE***);  
 *// ẩn imageview user* **imvUser**.setVisibility(View.***GONE***);  
 *// lắng nghe sự kiện click cho imageview* **imvLogin**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 *// chuyển sang giao diện LoginActivity* startActivity(**new** Intent(MainActivity.**this**,MainLoginActivity.**class**));  
 }  
 });  
 }  
 }  
Hàm initView() dùng để khởi tạo giao diện cho người chỉnh của trang chủ, và sẽ gán cho page ăn gì vị trí 0, page uống gì vị trí 1. Khi check vào radiobutton ăn gì( kq= 0) sẽ chuyển qua giao diện ăn gi ngược lại trả về giao diện uống gì  
  
 **private void** initView() {  
 *// khởi tạo dội tượng login* **mLogin** = **new** llogin(**this**,**this**);  
 *// ánh xạ sang dao diện* **imvLogin** = findViewById(R.id.***imvLogin***);  
 **imvUser** = findViewById(R.id.***imvUser***);  
  
  
  
  
 *//ánh xạ radiobutton sang radio giao diện* **rdoangi** = findViewById(R.id.***rdoangi***);  
 **rdouonggi** = findViewById(R.id.***rdouonggi***);  
 *// lắng nghe sự kiện click cho radiobutton* **rdoangi**.setOnCheckedChangeListener(**new** CompoundButton.OnCheckedChangeListener() {  
 @Override  
 **public void** onCheckedChanged(CompoundButton compoundButton, **boolean** b) {  
 **if**(b){  
 *//set dao diên ăn gì là item ví trí 0* **vpMain**.setCurrentItem(0);  
 }**else** *// set dao diện uống gì item vị trí 1* **vpMain**.setCurrentItem(1);  
 }  
 });  
 *// ánh xạ ViewPage sang giao diện* **this**.**vpMain** = findViewById(R.id.***vpMain***);  
 *// khởi tạo đối tượng adapter để truyền dữ liệu* **adapter** = **new** aViewPagerMenu(getSupportFragmentManager());  
 *// truyền dữ liệu vào ViewPage bên ma giao diện* **vpMain**.setAdapter(**adapter**);  
 *// lắng nghe sự kiện thay đổi cho ViewPage* **vpMain**.setOnPageChangeListener(**new** ViewPager.OnPageChangeListener() {  
 @Override  
 **public void** onPageScrolled(**int** i, **float** v, **int** i1) {  
  
 }  
  
 @Override  
 *//hàm xem page nào đã được chọn* **public void** onPageSelected(**int** i) {  
 **switch** (i){  
 **case** 0:  
 *// page dồ ăn gì được chọn, set checked* **rdoangi**.setChecked(**true**);  
 **break**;  
 **case** 1:  
 *// page dồ uống gì được chọn, set checked* **rdouonggi**.setChecked(**true**);  
 **break**;  
  
 }  
 }  
  
 @Override  
 **public void** onPageScrollStateChanged(**int** i) {  
  
 }  
 });  
 *// ánh xã imageView sang giao diện* **imvSearch** = findViewById(R.id.***imvSearch***);  
 *// lắng nghe sự kiện click vào biểu tượng tìm kiếm* **imvSearch**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 *//chuyển sang dao diện tìm kiếm searchActivity.class* startActivity(**new** Intent(MainActivity.**this**,SearchActivity.**class**));  
 }  
 });  
 }  
  
  
 @Override  
 **public void** ResultLogin(**boolean** isSuccess) {  
  
 }  
  
 @Override  
 **public void** result\_listenner\_values\_user(objnguoidung\_app muser) {  
  
 }  
  
 @Override  
 **public void** result\_dangnhap\_sdt(**boolean** isSuccess) {  
  
 }  
  
 @Override  
 **public void** code(String code) {  
  
 }  
}

### AngiFragment

Class AngiFragment kế thừa interface igetdataapp và iangi. Claas này có chức năng trả về một giao diện các món ăn chi tiết. Có các hàm con initView() và getData()

**public class** AnGiFragment **extends** Fragment **implements** igetdataapp, iangi {  
  
 lgetdataapp **mDataApp**;  
  
 objkhuvuc\_app **mObjKhuVuc**;  
  
 RecyclerView **rclvChungAn**,**rclvDatBietAn**;  
 aRclvChung **adapterChungAn**;  
 aRclvDacSan **adapterDacSanAn**;  
  
 langi **mAnGi**;  
  
 CustomDialogLoading **mDialogLoading**;  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 *// tạo view và gán giao diện cho fragment\_an\_gi* View v = inflater.inflate(R.layout.***fragment\_an\_gi***, container, **false**);  
  
 initView(v);  
 getData();  
 **mDialogLoading**.showDialog(**"Loading.."**);  
  
 *// trả về view* **return** v;  
 }  
Hàm initView() dùng để khởi tạo một view để truyền dữ liệu vào, sử dụng đối tượng recyclerview,

tạo một dialog thông báo cho người dùng  
  
 **private void** initView(View v) {  
 *// khởi tạo đối tuong igetdataapp để lấy dưc liệu* **mDataApp** = **new** lgetdataapp(getContext(),**this**);  
 *// ánh xạ recyclerView sang giao diện* **rclvChungAn** = v.findViewById(R.id.***rclvAnChung***);  
 **rclvDatBietAn** = v.findViewById(R.id.***rclvAnDatBiet***);  
 *// khởi tạo đối tượng ăn gì* **mAnGi** = **new** langi(**this**);  
 *// khởi tạo một dialog và gán cho nó ở dao điện hiện tại* **mDialogLoading** = **new** CustomDialogLoading(getContext());  
  
  
  
  
 }

Hàm getData() dùng để lấy các đặc sản thông qua mã khu vực, nếu ko có đối tượng khu vưc thì nó sẽ chuyển về giao diện Indexactivity  
  
 **private void** getData(){  
 *// gán đối tượng khu vực bằng khu vực đã lấy hiện tại* **mObjKhuVuc** = **mDataApp**.getmykhuvuc();  
 *// nếu mà chưa có khu vực thì nó sẽ chuyển về giao diện indexactivity* **if**(**mObjKhuVuc** == **null**){  
 Intent intent = **new** Intent(getActivity(),IndexActivity.**class**);  
 startActivity(intent);  
 getActivity().finish();  
 }  
 *// nếu tồn tại thì nó đặc sản dựa trên mã khu vực* **mAnGi**.getDacSan(**mObjKhuVuc**.getMakhuvuc());  
  
 }  
  
 @Override  
 **public void** loadkhuvuc(**boolean** isSuccess) {  
  
 }  
  
 @Override  
 **public void** saveMyData(**boolean** isSuccess, objkhuvuc\_app mdata) {  
  
 }  
  
 @Override  
 **public void** loitaidulieu\_getDataApp() {  
  
 }  
Hàm loitaidulieu\_Angi() là một methor của iangi. Hàm này sẽ tạo ra mộ dialog thông báo người dùng khi không tải được dữ liệu  
 @Override  
 **public void** loitaidulieu\_AnGi() {  
 *// khởi tạo một dialoag* AlertDialog.Builder builder = **new** AlertDialog.Builder(getContext());  
 *//set nội dung của dialog* builder.setMessage(getText(R.string.***loitaidulieukhoidonglai***));  
 *// tạo button bên trái ok và bỏ qua dialog* builder.setNegativeButton(**"OK"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 System.*exit*(0);  
 }  
 });  
 *// tạo button cancel bên pải cancel và bỏ qua dialog* builder.setPositiveButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 }  
 });  
 *// khi click vào vùng bên ngoài thì dialog ko biến mất* builder.setCancelable(**false**);  
 *// khởi tạo dialog* Dialog dialog = builder.create();  
 *// hiển thị dia log* dialog.show();  
 }  
  
 @Override  
 **public void** danhsachbaivietchung\_AnGi(**final** ArrayList<objbaiviet\_app> list) {  
  
  
 }

Hàm dánhachbaivietdacsan\_AnGi() là một methor của iangi. Hàm này cho phét xuất dữ liệu lên RecyclerView(RecyclerView DaBietAn bên giao diện) thông qua adapter. Và khi click vào từng item thì nó sẽ chuyển sang giao diện baivietchitietActivity  
  
  
 @Override  
 **public void** danhsachbaivietdacsan\_AnGi(**final** ArrayList<objbaiviet\_app> list) {  
 *// tải xong dữ liệu thì bỏ qua dialog* **mDialogLoading**.dismissDialog();  
 *// neus danh sách mảng objectbaiviet\_app ko rổng* **if**(list!=**null**) {  
 *// thì sẽ tạo một linearlayout* LinearLayoutManager manager = **new** LinearLayoutManager(getContext(), LinearLayoutManager.***VERTICAL***, **false**);  
 *// set kich thước vừa phải* **rclvDatBietAn**.setHasFixedSize(**true**);  
 *//set recyclerviewDaBietAn cái layout* **rclvDatBietAn**.setLayoutManager(manager);  
 *// khởi tạo adapter để truyền dữ liệu* **adapterDacSanAn** = **new** aRclvDacSan(getActivity(), list);  
 *// truyền dữ liệu vào recyclerview qa dapter* **rclvDatBietAn**.setAdapter(**adapterDacSanAn**);  
 *// lắng nghe sự kiên click từng item cho adapter* **adapterDacSanAn**.setOnItemClickListener(**new** aRclvDacSan.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(View itemView, **int** position) {  
 *// khởi tạo đối tượng bundle để lưu dữ liệu* Bundle bundle = **new** Bundle();  
 *// put bài viết lên theo vị trí* bundle.putSerializable(**"baiviet"**, list.get(position));  
 bundle.putString(**"loai"**, **"monan"**);  
 *// chuyển sang giao diện ChiTietBaiVietActivity* Intent intent = **new** Intent(getContext(), ChiTietBaiVietActivity.**class**);  
 *// truyền dữ liêu vào intent với key "data"* intent.putExtra(**"data"**, bundle);  
 *// chạy activity* getActivity().startActivity(intent);  
 }  
 });  
 }  
  
 }  
Hàm dánhachbaivietphobien\_AnGi() là một methor của iangi. Hàm nỳ cho phét xuất dữ liệu lên RecyclerView(RecyclerView ChungAn bên giao diện) thông qua adapter. Và khi click vào từng item thì nó sẽ chuyển sang giao diện baivietchitietActivity  
 @Override  
 **public void** danhsachbaivietphobien\_AnGi(**final** ArrayList<objbaiviet\_app> list) {  
 *//TẢI XONG DỮ LIỆU DISMISS DIALOG* **mDialogLoading**.dismissDialog();  
 *//nếu danh sach không rỗng* **if**(list!=**null**) {  
 *//khởi tạo một linerlayout* LinearLayoutManager manager = **new** LinearLayoutManager(getContext(), LinearLayoutManager.***HORIZONTAL***, **false**);  
 *//set kich thuoc phù hợp* **rclvChungAn**.setHasFixedSize(**true**);  
 *// truyền layout vào recyclerview* **rclvChungAn**.setLayoutManager(manager);  
 *// khởi tạo adapter và truyền vào danh sách bài viết* **adapterChungAn** = **new** aRclvChung(getActivity(), list);  
 *//truyền dữ liệu vào recyclerview bằng adapter* **rclvChungAn**.setAdapter(**adapterChungAn**);  
 *// lắng nghe sự kiện click vào các item của recyclerview* **adapterChungAn**.setOnClickListener(**new** aRclvChung.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(View itemView, **int** position) {  
 *// khởi tạo đối tượng bundle để lưu dữ liệu* Bundle bundle = **new** Bundle();  
 *// put dữ liệu vào danh sách* bundle.putSerializable(**"baiviet"**, list.get(position));  
 bundle.putString(**"loai"**, **"monan"**);  
 *// chuyên sang giao diện chitietactivity.class* Intent intent = **new** Intent(getContext(), ChiTietBaiVietActivity.**class**);  
 *// truyền dữ liệu vào intent thông qua key dataapp* intent.putExtra(**"data"**, bundle);  
 *// chay activity* getActivity().startActivity(intent);  
 }  
 });  
 }  
  
 }  
  
 @Override  
 **public void** taidulieuthanhcong(**boolean** i) {  
  
 }  
}

### UongGiFragment

Class này tương tự class AngiFragment

**public class** UongGiFragment **extends** Fragment **implements** igetdataapp,iuonggi {  
  
 lgetdataapp **mDataApp**;  
  
 objkhuvuc\_app **mObjKhuVuc**;  
  
 RecyclerView **rclvChung**,**rclvDatBiet**;  
 aRclvChung **adapterChungUong**;  
 aRclvDacSan **adapterDacSanUong**;  
  
 luonggi **mUongGi**;  
 CustomDialogLoading **mDialogLoading**;  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 *//khởi tạo view* View v = inflater.inflate(R.layout.***fragment\_uong\_gi***, container, **false**);  
  
 initView(v);  
 getData();  
 **mDialogLoading**.showDialog(**"Loading.."**);  
  
  
 **return** v;  
 }  
  
  
  
 **private void** initView(View v) {  
 **mDataApp** = **new** lgetdataapp(getContext(),**this**);  
  
 **rclvChung** = v.findViewById(R.id.***rclvUongChung***);  
 **rclvDatBiet** = v.findViewById(R.id.***rclvUongDatBiet***);  
  
 **mUongGi** = **new** luonggi(**this**);  
 **mDialogLoading** = **new** CustomDialogLoading(getContext());  
  
  
 }  
  
 **private void** getData(){  
 **mObjKhuVuc** = **mDataApp**.getmykhuvuc();  
 **if**(**mObjKhuVuc** == **null**){  
 Intent intent = **new** Intent(getActivity(),IndexActivity.**class**);  
 startActivity(intent);  
 getActivity().finish();  
 }  
 **mUongGi**.getBaiVietDatSan(**mObjKhuVuc**.getMakhuvuc());  
 }  
  
  
 @Override  
 **public void** loadkhuvuc(**boolean** isSuccess) {  
  
 }  
  
 @Override  
 **public void** saveMyData(**boolean** isSuccess, objkhuvuc\_app mdata) {  
  
 }  
  
 @Override  
 **public void** loitaidulieu\_getDataApp() {  
  
 }  
  
 @Override  
 **public void** taidulieuthanhcong(**boolean** isRealtime) {  
  
 }  
  
 @Override  
 **public void** loitaidulieu\_UongGi() {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(getContext());  
 builder.setMessage(getText(R.string.***loitaidulieukhoidonglai***));  
 builder.setNegativeButton(**"OK"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 System.*exit*(0);  
 }  
 });  
 builder.setPositiveButton(**"Cancel"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 }  
 });  
 builder.setCancelable(**false**);  
 Dialog dialog = builder.create();  
 dialog.show();  
 }  
  
 @Override  
 **public void** danhsachbaivietchung\_uonggi(**final** ArrayList<objbaiviet\_app> list) {  
  
 }  
  
 @Override  
 **public void** danhsachbaivietdacsan\_uonggi(**final** ArrayList<objbaiviet\_app> list) {  
 **mDialogLoading**.dismissDialog();  
 **if**(list!=**null**) {  
 **adapterDacSanUong** = **new** aRclvDacSan(getActivity(), list);  
  
 LinearLayoutManager manager = **new** LinearLayoutManager(getContext(), LinearLayoutManager.***VERTICAL***, **false**);  
 **rclvDatBiet**.setHasFixedSize(**true**);  
 **rclvDatBiet**.setLayoutManager(manager);  
  
 **rclvDatBiet**.setAdapter(**adapterDacSanUong**);  
  
 **adapterDacSanUong**.setOnItemClickListener(**new** aRclvDacSan.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(View itemView, **int** position) {  
 Bundle bundle = **new** Bundle();  
 bundle.putSerializable(**"baiviet"**, list.get(position));  
 bundle.putString(**"loai"**, **"nuocuong"**);  
 Intent intent = **new** Intent(getContext(), ChiTietBaiVietActivity.**class**);  
 intent.putExtra(**"data"**, bundle);  
 getActivity().startActivity(intent);  
 }  
 });  
 }  
  
 }  
  
 @Override  
 **public void** danhsachbaivietphobien\_uonggi(**final** ArrayList<objbaiviet\_app> list) {  
 **mDialogLoading**.dismissDialog();  
 **if**(list!=**null**) {  
 LinearLayoutManager manager = **new** LinearLayoutManager(getContext(), LinearLayoutManager.***HORIZONTAL***, **false**);  
 **rclvChung**.setHasFixedSize(**true**);  
 **rclvChung**.setLayoutManager(manager);  
  
 **adapterChungUong** = **new** aRclvChung(getContext(), list);  
 **rclvChung**.setAdapter(**adapterChungUong**);  
  
 **adapterChungUong**.setOnClickListener(**new** aRclvChung.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(View itemView, **int** position) {  
 Bundle bundle = **new** Bundle();  
 bundle.putSerializable(**"baiviet"**, list.get(position));  
 bundle.putString(**"loai"**, **"nuocuong"**);  
 Intent intent = **new** Intent(getContext(), ChiTietBaiVietActivity.**class**);  
 intent.putExtra(**"data"**, bundle);  
 getActivity().startActivity(intent);  
 }  
 });  
 }  
 }

### SettingActivity

**public class** SettingActivity **extends** AppCompatActivity {  
  
 Toolbar **toolbar**;  
 Button **btnEn**,**btnVi**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_setting***);  
 initView();  
 actionSetingLanguage();  
 }  
Hàm actionStingLanguage() cho phép người dùng click vào biểu tượng lá cờ việt nam thì app sẽ load lại và chuyễn sang ngôn ngữ việt nam, click vào lá cờ mỹ sẽ chuyển sang ngôn ngữ tiếng anh thông qua phương thức gán ngôn ngữ  
 **private void** actionSetingLanguage() {  
 **btnVi**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 ganngonngu(**"vi"**);  
 }  
 });  
 **btnEn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 ganngonngu(**"en"**);  
 }  
 });  
 }  
Hàm khởi tạo View cho phép ánh xạ với các button bên giao diện. Biến Toolbar thành ActionBar  
 **private void** initView() {  
 *//Ánh xã button sang giao diện* **btnEn** = findViewById(R.id.***btnEN***);  
 **btnVi** = findViewById(R.id.***btnVI***);  
 **toolbar** = findViewById(R.id.***toolbarsetting***);  
 *//biến toolbar thành actionbar* setSupportActionBar(**toolbar**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 *//lắng nghe sự kiện click vào actionbar* **toolbar**.setNavigationOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//chuyển về giao diện trước* onBackPressed();  
 finish();  
 }  
 });  
 }  
  
 Hàm ganngonngu() truyền vào một ngôn ngũ để chuyển ngôn ngữ cho app

**public void** ganngonngu(String language){  
 *// khởi tạo đối tượng đa ngôn ngữ* Locale myLocale = **new** Locale(language);  
 *// cấu tạo lại hệ thống* Configuration config = **new** Configuration();  
 *// cấu hình lại ngôn ngữ* config.**locale** = myLocale;  
 *// cập nhật lại resources* getBaseContext().getResources().updateConfiguration(config,getBaseContext().getResources().getDisplayMetrics());  
 *//load lại màn hình* Intent intent = **new** Intent(SettingActivity.**this**,MainActivity.**class**);  
 intent.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK*** | Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 startActivity(intent);  
 Toast.*makeText*(**this**,getText(R.string.***thanhcong***),Toast.***LENGTH\_SHORT***).show();  
 finish();  
  
  
 }  
}

## Tài khoản

### Admin Activity

**public class** AdminActivity **extends** AppCompatActivity **implements** iViewAdmin, NavigationView.OnNavigationItemSelectedListener {  
  
 **private** DrawerLayout **drawerLayout**;  
 **private** EndDrawerToggle **toggle**;  
 **private** Toolbar **toolbar**;  
 **private** NavigationView **navigation**;  
  
 AlertView **mAlertView**;  
  
 llogin **mlogin**;  
 logicAdmin **mLogicAdmin**;  
  
 objnguoidung\_app **mCTTK**;  
  
 TextView **tvTen**, **tvEmail**, **tvThemBV**;  
 ImageView **imvanhdaidien**, **imvFilter**;  
  
 XRecyclerView **rclvBaiVietCuaToi**;  
 aRclvBaiVietCuaToi **adapterRclv**;  
 ArrayList<objbaiviet\_app> **listBaiVietFilter**, **listBaiVietDefault**, **listLoadmore**;  
  
 getDataApp **mGetData**;  
  
 String[] **listNameKhuVuc**;  
 ArrayList<objkhuvuc\_app> **listKhuVuc**;  
 objkhuvuc\_app **mKhuVucFilter**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_admin***);  
 initView();  
 getData();  
 actionGetImage();  
 **mLogicAdmin**.getlistbaivietcuatoi();  
  
 actionLoc();  
 }  
  
 **private void** actionLoc() {  
 **imvFilter**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **final** DialogPlus dialog = DialogPlus.*newDialog*(AdminActivity.**this**)  
 .setGravity(Gravity.***TOP***)  
 .setContentHolder(**new** ViewHolder(R.layout.***dialog\_filter\_baivietcuatoi***))  
 .setExpanded(**false**)  
 .setPadding(10,40,10,10)  
 .setCancelable(**true**)  
 .create();  
 dialog.show();  
  
 *//INIT VIEW* View v = dialog.getHolderView();  
 **final** SwitchButton sbLoaiLoc = v.findViewById(R.id.***sbkieuloc\_filter***);  
 **final** Button btnchonkhuvuc = v.findViewById(R.id.***btnChonKhuVuc\_dialogfilter***);  
 **final** TextView tvKieuLoc = v.findViewById(R.id.***tvkieuloc\_filter***);  
 **final** LinearLayout lnlCustomLoc = v.findViewById(R.id.***lnlCustomLoc\_filter***);  
 Button btnLoc = v.findViewById(R.id.***btnLoc\_dialogfilter***);  
 **final** RadioButton rdoThucAn = v.findViewById(R.id.***rodthucan\_dialogfilter***);  
  
 btnLoc.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 dialog.dismiss();  
 **mLogicAdmin**.locbaiviet(!sbLoaiLoc.isChecked(),rdoThucAn.isChecked(),**mKhuVucFilter**,**listBaiVietDefault**);  
 }  
 });  
  
  
 btnchonkhuvuc.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 dialogChonKhuVuc(btnchonkhuvuc);  
 }  
 });  
  
 **if**(!sbLoaiLoc.isChecked()){  
 tvKieuLoc.setText(getText(R.string.***loctatcafilter***));  
 lnlCustomLoc.setVisibility(View.***GONE***);  
 }**else** {  
 tvKieuLoc.setText(getText(R.string.***loctuychonfilter***));  
 lnlCustomLoc.setVisibility(View.***VISIBLE***);  
 }  
  
 sbLoaiLoc.setOnCheckedChangeListener(**new** CompoundButton.OnCheckedChangeListener() {  
 @Override  
 **public void** onCheckedChanged(CompoundButton compoundButton, **boolean** b) {  
 **if**(!b){  
 tvKieuLoc.setText(getText(R.string.***loctatcafilter***));  
 lnlCustomLoc.setVisibility(View.***GONE***);  
 }**else** {  
 tvKieuLoc.setText(getText(R.string.***loctuychonfilter***));  
 lnlCustomLoc.setVisibility(View.***VISIBLE***);  
 **mKhuVucFilter** = **new** objkhuvuc\_app(**"null"**,**new** objtinhthanh(**"Tất cả"**,0,0));  
 btnchonkhuvuc.setText(**"Tất cả"**);  
 }  
 }  
 });  
  
  
  
 }  
 });  
 }  
Hàm dialogChonKhuVuc() sẽ tạo list danh sách khu vực, sau đó sẽ load danh sách các tỉnh thành lên dialog sau đó sẽ chuyển sang đi trang đăng bào viết  
 **private void** dialogChonKhuVuc(**final** Button btnChonKhuVuc){  
 **listKhuVuc** = **new** ArrayList<>();  
 **listKhuVuc**.add(**new** objkhuvuc\_app(**"null"**,**new** objtinhthanh(**"Tất cả"**,0,0)));  
 **listKhuVuc**.addAll(**mGetData**.getListKhuVuc());  
 **if**(**listKhuVuc**!=**null**){  
 *//SETUP LIST NAME KHU VUC* **int** size = **listKhuVuc**.size();  
 **listNameKhuVuc** = **new** String[size];  
 **for** (**int** i=0; i<size; i++){  
 **listNameKhuVuc**[i]=**listKhuVuc**.get(i).getChitietkhuvuc().getTentinh();  
 }  
  
 android.app.AlertDialog.Builder mBuilder = **new** android.app.AlertDialog.Builder(**this**);  
 mBuilder.setCancelable(**false**);  
 mBuilder.setTitle(getText(R.string.***chonkhuvuc***));  
 mBuilder.setItems(**listNameKhuVuc**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 **mKhuVucFilter** = **listKhuVuc**.get(i);  
 btnChonKhuVuc.setText(**mKhuVucFilter**.getChitietkhuvuc().getTentinh());  
 }  
 });  
 mBuilder.setPositiveButton(getText(R.string.***huy***), **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 }  
 });  
 Dialog dialog = mBuilder.create();  
 dialog.show();  
  
 }**else** Toast.*makeText*(**this**, getText(R.string.***loikhuvuc***), Toast.***LENGTH\_SHORT***).show();  
 }

**private void** actionGetImage() {  
 **tvThemBV**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 startActivity(**new** Intent(AdminActivity.**this**,DangBaiVietActivity.**class**));  
 }  
 });  
 }  
  
  
Hàm initView() setup khởi tạo những thành phần ban đâu cho view như ánh xạ sang giao diện toolbar,navigation…  
 **private void** initView() {  
 **toolbar** = findViewById(R.id.***toolbaradmin***);  
 setSupportActionBar(**toolbar**);  
 **toolbar**.setTitle(getText(R.string.***trangcanhan***));  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 **toolbar**.setNavigationOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 finish();  
 }  
 });  
  
 **imvFilter** = findViewById(R.id.***imvFilter***);  
  
 **drawerLayout** = findViewById(R.id.***drawarlayoutadmin***);  
 **toggle** = **new** EndDrawerToggle(**this**,**drawerLayout**,**toolbar**,R.string.***open***,R.string.***close***);  
 **drawerLayout**.addDrawerListener(**toggle**);  
  
 **navigation** = findViewById(R.id.***navigationviewadmin***);  
 **if** (**navigation** != **null**) {  
 **navigation**.setNavigationItemSelectedListener(**this**);  
 }  
  
  
 **mLogicAdmin** = **new** logicAdmin(**this**,**this**);  
  
 View header = **navigation**.getHeaderView(0);  
  
 **tvTen** = header.findViewById(R.id.***tvten\_admin***);  
 **tvEmail** = header.findViewById(R.id.***tvemai\_admin***);  
 **imvanhdaidien** = header.findViewById(R.id.***imvanhdaidien\_admin***);  
  
 **rclvBaiVietCuaToi** = findViewById(R.id.***rclvBaiVietCuaToi***);  
  
 **tvThemBV** = findViewById(R.id.***tvThembaivietmoi***);  
  
 **mGetData** = **new** getDataApp(**this**);  
  
 }

Hàm getData sẽ lấy dữ liệu của người vừa đăng nhập như: lấy tên,email,nếu có ảnh sẽ lấy ảnh gắn cho imageview còn không sẽ lấy ảnh mặt định  
  
 **private void** getData() {  
 **mlogin** = **new** llogin( **new** ilogin() {  
 @Override  
 **public void** ResultLogin(**boolean** isSuccess) {  
  
 }  
  
 @Override  
 **public void** result\_listenner\_values\_user(objnguoidung\_app muser) {  
 **mCTTK** = muser;  
 **tvTen**.setText(**mCTTK**.getNguoidung().getHoten());  
 **tvEmail**.setText(**mCTTK**.getNguoidung().getEmail());  
 String urlImg = **mCTTK**.getNguoidung().getAvatar();  
 **if** (urlImg.matches(**""**) || **mlogin**.getImage(**mCTTK**.getAnhdaidien())==**null**) {  
 **imvanhdaidien**.setImageResource(R.drawable.***defaultuser***);  
 } **else** {  
 **imvanhdaidien**.setImageDrawable(**mlogin**.getImage(**mCTTK**.getAnhdaidien()));  
 }  
 }  
  
 @Override  
 **public void** result\_dangnhap\_sdt(**boolean** isSuccess) {  
  
 }  
  
 @Override  
 **public void** code(String code) {  
  
 }  
 },**this**);  
 **mCTTK** = **mlogin**.getDataNguoiDung();  
 **if**(**mCTTK**!=**null**) {  
 **mlogin**.listenerValuesUser();  
 **tvTen**.setText(**mCTTK**.getNguoidung().getHoten());  
 **tvEmail**.setText(**mCTTK**.getNguoidung().getEmail());  
 String urlImg = **mCTTK**.getNguoidung().getAvatar();  
 **if** (urlImg.matches(**""**) || **mlogin**.getImage(**mCTTK**.getAnhdaidien())==**null**) {  
 **imvanhdaidien**.setImageResource(R.drawable.***defaultuser***);  
 } **else** {  
 **imvanhdaidien**.setImageDrawable(**mlogin**.getImage(**mCTTK**.getAnhdaidien()));  
 }  
 }  
 }  
Hàm dialogDangXuat cho phép khi dialog hiện lên sẽ có hai thông báo bên trái là hủy, bên phải là đồng ý. Nếu người dùng click vào đăng xuất thì sẽ gọi tới firebase Auth để đăng xuất, còn nếu click vào hủy thì bỏ qua dialog  
  
 **private void** dialogDangXuat(){  
 **mAlertView** = **new** AlertView(getText(R.string.***banmuondangxuat***).toString(),  
 **null**, getText(R.string.***ok***).toString()  
 , **new** String[]{getText(R.string.***huy***).toString()},  
 **null**,  
 **this**,  
 AlertView.Style.***Alert***,  
 **new** OnItemClickListener() {  
 @Override  
 **public void** onItemClick(Object o, **int** position) {  
 **if**(position==-1){  
 FirebaseAuth.*getInstance*().signOut();  
 finish();  
 }**else  
 mAlertView**.dismiss();  
 *//Toast.makeText(AdminActivity.this, position+"", Toast.LENGTH\_SHORT).show();* }  
 }).setCancelable(**false**)  
 .setOnDismissListener(**null**);  
 **mAlertView**.show();  
  
 }  
  
  
 @Override  
 **protected void** onPostCreate(Bundle savedInstanceState) {  
 **super**.onPostCreate(savedInstanceState);  
 **toggle**.syncState();  
 }  
  
  
  
 @Override  
 **public void** result\_baivietcuatoi(**final** ArrayList<objbaiviet\_app> list) {  
 **if**(list!=**null**) {  
 **int** size = list.size();  
 **listLoadmore** = **new** ArrayList<>();  
 **if**(size>10){  
 **for**(**int** i=0; i<10; i++){  
 **listLoadmore**.add(list.get(i));  
 }  
 }**else**{  
 **listLoadmore**.addAll(list);  
 }  
 **rclvBaiVietCuaToi**.refreshComplete();  
 **listBaiVietFilter** = list;  
 **listBaiVietDefault** = list;  
 LinearLayoutManager manager = **new** LinearLayoutManager((Context) **this**, LinearLayoutManager.***VERTICAL***, **false**);  
 **rclvBaiVietCuaToi**.setHasFixedSize(**true**);  
 **rclvBaiVietCuaToi**.setLayoutManager(manager);  
 **adapterRclv** = **new** aRclvBaiVietCuaToi(**this**, **listLoadmore**);  
 **rclvBaiVietCuaToi**.setAdapter(**adapterRclv**);  
  
 **rclvBaiVietCuaToi**.setArrowImageView(R.drawable.***iconfont\_downgrey***);  
 **rclvBaiVietCuaToi**.getDefaultFootView().setLoadingHint(getText(R.string.***dangtai***).toString());  
 **rclvBaiVietCuaToi**.setLoadingMoreProgressStyle(ProgressStyle.***BallSpinFadeLoader***);  
 **rclvBaiVietCuaToi**.setRefreshProgressStyle(ProgressStyle.***BallSpinFadeLoader***);  
  
 *//LOAD MORE* **final int** itemLimit = 10;  
 **rclvBaiVietCuaToi**.setNoMore(**false**);  
 **rclvBaiVietCuaToi**.setLoadingListener(**new** XRecyclerView.LoadingListener() {  
 @Override  
 **public void** onRefresh() {  
 **mLogicAdmin**.getlistbaivietcuatoi();  
 }  
  
 @Override  
 **public void** onLoadMore() {  
 **new** Handler().postDelayed(**new** Runnable() {  
 **public void** run() {  
 **int** count = **listLoadmore**.size();  
 **for**(**int** i = 0; i < itemLimit ;i++){  
 **if**((i+count)<list.size()){  
 **listLoadmore**.add(list.get(i+count));  
 }  
 **else  
 break**;  
 }  
  
 **if**(**listLoadmore**.size()==list.size()) {  
 **rclvBaiVietCuaToi**.setNoMore(**true**);  
 Toast.*makeText*(AdminActivity.**this**, getText(R.string.***dataixong***), Toast.***LENGTH\_SHORT***).show();  
 }  
 **adapterRclv**.notifyDataSetChanged();  
 **rclvBaiVietCuaToi**.loadMoreComplete();  
 }  
 }, 1000);  
 }  
 });  
  
  
 **adapterRclv**.setOnItemClickListener(**new** aRclvBaiVietCuaToi.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(View itemView, **int** position) {  
 Log.*e*(**"kiemtra"**,position + **" "**);  
 Bundle bundle = **new** Bundle();  
 bundle.putSerializable(**"baiviet"**, **listLoadmore**.get(position));  
 bundle.putString(**"loai"**, **"monan"**);  
 Intent intent = **new** Intent(AdminActivity.**this**, ChiTietBaiVietActivity.**class**);  
 intent.putExtra(**"data"**, bundle);  
 startActivity(intent);  
 }  
 });  
 }  
 }  
  
 @Override  
 **public void** result\_delete(**boolean** isSuccess) {  
  
 }  
Hàm list baivietdafilter dùng để add một danh sách bài viết đã lọc( lọc theo “nước uống” và lọc theo “thức ăn”. Sau khi đã lọc người dùng click vào item của list món ăn, thì sẽ chuyển đến trang chi tiết của món ăn đó và setup một số giao diện cho list  
 @Override  
 **public void** listbaivietdafilter(**final** ArrayList<objbaiviet\_app> list) {  
 **if**(list!=**null**) {  
 **this**.**listBaiVietFilter** = list;  
 Log.*e*(**"kiemtra"**,**listBaiVietFilter**.size()+ **" listBaiVietFilter"**);  
 **int** size = **listBaiVietFilter**.size();  
 **listLoadmore** = **new** ArrayList<>();  
 **if**(size>10){  
 **for**(**int** i=0; i<10; i++){  
 **listLoadmore**.add(**listBaiVietFilter**.get(i));  
 }  
 }**else**{  
 **listLoadmore**.addAll(**listBaiVietFilter**);  
 }  
  
 LinearLayoutManager manager = **new** LinearLayoutManager((Context) **this**, LinearLayoutManager.***VERTICAL***, **false**);  
 **rclvBaiVietCuaToi**.setHasFixedSize(**true**);  
 **rclvBaiVietCuaToi**.setLayoutManager(manager);  
 **adapterRclv** = **new** aRclvBaiVietCuaToi(**this**, **listLoadmore**);  
 **rclvBaiVietCuaToi**.setAdapter(**adapterRclv**);  
  
 **rclvBaiVietCuaToi**.setArrowImageView(R.drawable.***iconfont\_downgrey***);  
 **rclvBaiVietCuaToi**.getDefaultFootView().setLoadingHint(getText(R.string.***dangtai***).toString());  
 **rclvBaiVietCuaToi**.setLoadingMoreProgressStyle(ProgressStyle.***BallSpinFadeLoader***);  
 **rclvBaiVietCuaToi**.setRefreshProgressStyle(ProgressStyle.***BallSpinFadeLoader***);  
  
 *//LOAD MORE* **final int** itemLimit = 10;  
 **rclvBaiVietCuaToi**.setNoMore(**false**);  
 **rclvBaiVietCuaToi**.setLoadingListener(**new** XRecyclerView.LoadingListener() {  
 @Override  
 **public void** onRefresh() {  
 **mLogicAdmin**.getlistbaivietcuatoi();  
 }  
  
 @Override  
 **public void** onLoadMore() {  
 **new** Handler().postDelayed(**new** Runnable() {  
 **public void** run() {  
 **int** count = **listLoadmore**.size();  
 **for**(**int** i = 0; i < itemLimit ;i++){  
 **if**((i+count)<**listBaiVietFilter**.size()){  
 **listLoadmore**.add(**listBaiVietFilter**.get(i+count));  
 }  
 **else  
 break**;  
 }  
 **if**(**listLoadmore**.size()==**listBaiVietFilter**.size()) {  
 **rclvBaiVietCuaToi**.setNoMore(**true**);  
 Toast.*makeText*(AdminActivity.**this**, getText(R.string.***dataixong***), Toast.***LENGTH\_SHORT***).show();  
 }  
  
 **adapterRclv**.notifyDataSetChanged();  
 **rclvBaiVietCuaToi**.loadMoreComplete();  
 }  
 }, 1000);  
 }  
 });  
  
  
 **adapterRclv**.setOnItemClickListener(**new** aRclvBaiVietCuaToi.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(View itemView, **int** position) {  
 Log.*e*(**"kiemtra"**,**new** Gson().toJson(**listLoadmore**.get(position)));  
 Bundle bundle = **new** Bundle();  
 bundle.putSerializable(**"baiviet"**, **listLoadmore**.get(position));  
 bundle.putString(**"loai"**, **"monan"**);  
 Intent intent = **new** Intent(AdminActivity.**this**, ChiTietBaiVietActivity.**class**);  
 intent.putExtra(**"data"**, bundle);  
 startActivity(intent);  
 }  
 });  
 }  
 }  
Hàm onNavigationItemSelected có nhiệm vụ chọn những item trong Navigation, nếu chọn item có id là đăng xuất thì sẽ xuất hiện 1 dialog thông báo xem co đăng xuất hay không, nếu chọn idcaapj nhật thông tin sẽ chuyển sang giao diện activity cập nhật tài khoản, nếu chọn vào idcaidat sẽ chuyển sang giao diện cài đặt ngôn ngữ cho app, còn nếu chọn item có id là bài viết đã lưu thì chuyển sang giao diện bài viết đã lứu  
 @Override  
 **public boolean** onNavigationItemSelected(@NonNull MenuItem menuItem) {  
 **int** id = menuItem.getItemId();  
 **switch** (id){  
 **case** R.id.***itdangxuat***:  
 dialogDangXuat();  
 **break**;  
 **case** R.id.***itcapnhatthongtin***:  
 startActivity(**new** Intent(**this**,CapNhatTaiKhoanActivity.**class**));  
 **break**;  
 **case** R.id.***itcaidat***:  
 startActivity(**new** Intent(**this**,SettingActivity.**class**));  
 **break**;  
 **case** R.id.***itbaivietdaluu***:  
 startActivity(**new** Intent(**this**,BaiVietDaLuuActivity.**class**));  
 **break**;  
 **case** R.id.***itthemgoiy***:  
 **break**;  
 }  
 **drawerLayout**.closeDrawers();  
 **return true**;  
 }  
}

### ChinhsuabaivietActivity

**public class** ChinhSuaBaiVietActivity **extends** AppCompatActivity **implements** iViewChinhSuaBaiViet {  
  
  
 objbaiviet\_app **mBaiVietEdit**;  
  
 **int PICK\_IMAGE\_MULTIPLE** = 1;  
 ArrayList<String> **paths**;  
  
 Button **btnChonHinhAnh**, **btnLuu**;  
 ImageView **imv1**,**imv2**,**imv3**,**imv4**,**imv5**, **imvDel1**,**imvDel2**,**imvDel3**,**imvDel4**,**imvDel5**;  
 FrameLayout **frm1**,**frm2**,**frm3**,**frm4**,**frm5**;  
  
 Button **btnChonThanhPhan**;  
 EditText **edtTen**, **edtGioiThieu**, **edtGhiChu**, **edtDiaChi**, **edtGiaTien**;  
  
 Toolbar **toolbar**;  
  
 **boolean**[] **checkedItems**;  
 ArrayList<objthanhphan> **listThanhPhan**;  
 ArrayList<objthanhphan> **listThanhPhanBaiViet**;  
 String[] **listNameThanhPhan**;  
 ArrayList<Integer> **mIndexThanhPhan**;  
  
 ArrayList<objkhuvuc\_app> **listKhuVuc**;  
 objkhuvuc\_app **mKhuVucBaiViet**;  
  
  
 ArrayList<ImageView> **listImv**;  
 ArrayList<FrameLayout> **listFrm**;  
 ArrayList<ImageView> **listRemove**;  
  
 getDataApp **mGetData**;  
 logicChinhSuaBaiViet **mLogicChinhSua**;  
  
  
 CustomDialogLoading **mDialogLoading**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_chinh\_sua\_bai\_viet***);  
  
 initView();  
 getdata();  
 actionChonHinhAnh();  
 actionChinhSua();  
 actionChonThanhPhan();  
 actionLoai();  
 }  
Hàm lấy danh dữ liệu từ “data”, lấy bài viết sau đó gán cho biến data  
 **private void** getdata() {  
 Bundle bundle = getIntent().getBundleExtra(**"data"**);  
 **if**(bundle!=**null**){  
 **mBaiVietEdit** = (objbaiviet\_app) bundle.getSerializable(**"baiviet"**);  
 **listKhuVuc** = **mGetData**.getListKhuVuc();  
 setData(**mBaiVietEdit**);  
 }  
 }  
Hàm setData dùng để set up lại bài viết setup button, setup ảnh, món ăn theo khu vực, thành phần chi tiết của từng món ăn  
 **private void** setData(objbaiviet\_app baiviet) {  
  
 String[] decode = baiviet.getIdbaiviet().split(**"\_"**);  
 String type = decode[2];  
 String makhuvuc = decode[3];  
  
 **for**(objkhuvuc\_app i : **listKhuVuc**){  
 **if**(i.getMakhuvuc().matches(makhuvuc)){  
 **mKhuVucBaiViet** = i;  
 **break**;  
 }  
 }  
  
 String loai = **""**;  
 **if**(type.matches(**"monans"**))  
 loai = **"thanhphanmonan"**;  
 **else** loai = **"thanhphannuocuong"**;  
 **listThanhPhan** = **new** ArrayList<>();  
 **listThanhPhan**.clear();  
 **listThanhPhan** = **mGetData**.getListThanhPhan(loai);  
 **listThanhPhan**.remove(0);  
  
  
 *//=================================* **mIndexThanhPhan** = **new** ArrayList<>();  
 **int** size\_all = **listThanhPhan**.size();  
 **checkedItems** = **new boolean**[size\_all];  
 **listNameThanhPhan** = **new** String[size\_all];  
 *//KHỞI TẠO LIST NAME* **for**(**int** i = 0 ; i<size\_all; i++){  
 **listNameThanhPhan**[i] = **listThanhPhan**.get(i).getTenthanhphan();  
 }  
  
  
 *//GET LIST THÀNH PHẦN CỦA BÀI VIẾT* **listThanhPhanBaiViet** = **new** ArrayList<>();  
 **for**(objthanhphan i : **listThanhPhan**){  
 **for**(String id: **mBaiVietEdit**.getChitiet().getThanhphan()){  
 **if**(i.getMathanhphan().matches(id)){  
 **listThanhPhanBaiViet**.add(i);  
 }  
 }  
 }  
  
 **for**(**int** i=0; i<size\_all; i++){  
 **for**(objthanhphan tp : **listThanhPhanBaiViet**){  
 **if**(tp.getMathanhphan().matches(**listThanhPhan**.get(i).getMathanhphan())){  
 **checkedItems**[i] = **true**;  
 }  
 }  
 }  
  
 *//SET PATH HÌNH* **for**(String i: **mBaiVietEdit**.getChitiet().getHinh()){  
 **paths**.add(i);  
 }  
 setupImage();  
  
 **edtTen**.setText(baiviet.getChitiet().getTen());  
 **edtDiaChi**.setText(baiviet.getChitiet().getDiachi());  
 **edtGhiChu**.setText(baiviet.getChitiet().getGhichu());  
 **edtGiaTien**.setText(String.*valueOf*(baiviet.getChitiet().getGiathamkhao()));  
 **edtGioiThieu**.setText(baiviet.getChitiet().getGioithieu());  
  
 }  
  
Hàm action Loại lấy tên thành phần từ listThanhPhanBaiViet theo vị trí từ đó gán cho buttoon ThanhPhan  
 **private void** actionLoai() {  
  
 *//SETUP TÊN BUTTON* String tp = **""**;  
 **int** size\_bv = **listThanhPhanBaiViet**.size();  
 **for**(**int** i=0; i<size\_bv; i++){  
 **if** (i != size\_bv - 1)  
 tp = **listThanhPhanBaiViet**.get(i).getTenthanhphan() + **", "** + tp;  
 **else** tp += **" "** + **listThanhPhanBaiViet**.get(i).getTenthanhphan();  
 }  
 **btnChonThanhPhan**.setText(tp);  
  
 }  
  
 **private void** actionChonThanhPhan() {  
 **btnChonThanhPhan**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 showMultiSelectDialog();  
 }  
 });  
 }  
  
 **private void** showMultiSelectDialog(){  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 builder.setTitle(getText(R.string.***chonthanhphan***));  
  
 builder.setMultiChoiceItems(**listNameThanhPhan**, **checkedItems**, **new** DialogInterface.OnMultiChoiceClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** position, **boolean** checked) {  
 **if**(checked) {  
 **if** (!**mIndexThanhPhan**.contains(position)) {  
 **mIndexThanhPhan**.add(position);  
 }  
 }**else**{  
 **if** (**mIndexThanhPhan**.contains(position)) {  
 **int** count = **mIndexThanhPhan**.size();  
 **for**(**int** i=0; i<count; i++){  
 **if**(**mIndexThanhPhan**.get(i)==position) {  
 **mIndexThanhPhan**.remove(i);  
 **break**;  
 }  
 }  
 }  
 }  
  
 }  
 });  
  
 builder.setCancelable(**false**);  
 builder.setPositiveButton(getText(R.string.***ok***), **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 **int** count = **mIndexThanhPhan**.size();  
 **if**(count>0) {  
 String tp = **""**;  
 **for** (**int** position = 0; position < count; position++) {  
 **if** (position != count - 1)  
 tp = **listThanhPhan**.get(**mIndexThanhPhan**.get(position)).getTenthanhphan() + **", "** + tp;  
 **else** tp += **" "** + **listThanhPhan**.get(**mIndexThanhPhan**.get(position)).getTenthanhphan();  
 }  
 **btnChonThanhPhan**.setText(tp);  
 }**else** {  
 **btnChonThanhPhan**.setText(getText(R.string.***chonthanhphan***));  
 }  
 }  
 });  
 builder.setNegativeButton(getText(R.string.***huy***), **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 dialogInterface.dismiss();  
 }  
 });  
  
 Dialog dialog = builder.create();  
 dialog.show();  
 }  
  
 **private void** actionChinhSua() {  
 **btnLuu**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **edtDiaChi**.setError(**null**);  
 **edtGhiChu**.setError(**null**);  
 **edtGiaTien**.setError(**null**);  
 **edtGioiThieu**.setError(**null**);  
 **edtTen**.setError(**null**);  
 **mLogicChinhSua**.kiemtralogiceditbaiviet(**edtTen**,**edtGioiThieu**,**edtGiaTien**,**edtDiaChi**,**btnChonThanhPhan**, **listFrm**);  
 }  
 });  
 }  
  
 **private void** actionChonHinhAnh() {  
 **btnChonHinhAnh**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 actionSelectMultiImage();  
 }  
 });  
 }

Hàm initView để khởi tạo view, ánh xạ sang dao diện  
  
 **private void** initView() {  
 **toolbar** = findViewById(R.id.***toolbarEditBaiViet***);  
 setSupportActionBar(**toolbar**);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 **toolbar**.setNavigationOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 finish();  
 }  
 });  
  
 **mGetData** = **new** getDataApp(**this**);  
 **mLogicChinhSua** = **new** logicChinhSuaBaiViet(**this**,**this**);  
  
 **btnChonHinhAnh** = findViewById(R.id.***btnchonhinhanh\_editbv***);  
 **btnLuu** = findViewById(R.id.***btnluu\_editbv***);  
  
 **imv1** = findViewById(R.id.***imv1\_editbv***);  
 **imv2** = findViewById(R.id.***imv2\_editbv***);  
 **imv3** = findViewById(R.id.***imv3\_editbv***);  
 **imv4** = findViewById(R.id.***imv4\_editbv***);  
 **imv5** = findViewById(R.id.***imv5\_editbv***);  
  
 **listImv** = **new** ArrayList<>();  
 **listFrm** = **new** ArrayList<>();  
  
 **imvDel1** = findViewById(R.id.***imvDelete1\_editbv***);  
 **imvDel2** = findViewById(R.id.***imvDelete2\_editbv***);  
 **imvDel3** = findViewById(R.id.***imvDelete3\_editbv***);  
 **imvDel4** = findViewById(R.id.***imvDelete4\_editbv***);  
 **imvDel5** = findViewById(R.id.***imvDelete5\_editbv***);  
  
 **frm1** = findViewById(R.id.***frmImv1\_editbv***);  
 **frm2** = findViewById(R.id.***frmImv2\_editbv***);  
 **frm3** = findViewById(R.id.***frmImv3\_editbv***);  
 **frm4** = findViewById(R.id.***frmImv4\_editbv***);  
 **frm5** = findViewById(R.id.***frmImv5\_editbv***);  
  
 **paths** = **new** ArrayList<>();  
 **paths**.clear();  
  
 **btnChonThanhPhan** = findViewById(R.id.***btnchonthanhphan\_editbv***);  
  
 **edtDiaChi** = findViewById(R.id.***edtdiachi\_editbv***);  
 **edtGhiChu** = findViewById(R.id.***edtghichu\_editbv***);  
 **edtGiaTien** = findViewById(R.id.***edtgia\_editbv***);  
 **edtGioiThieu** = findViewById(R.id.***edtgioithieu\_editbv***);  
 **edtTen** = findViewById(R.id.***edttenbaiviet\_editbv***);  
  
 **mDialogLoading** = **new** CustomDialogLoading(**this**);  
  
 }  
  
 **private void** actionSelectMultiImage() {  
 Intent intent = **new** Intent();  
 intent.setType(**"image/\*"**);  
 intent.putExtra(Intent.***EXTRA\_ALLOW\_MULTIPLE***, **true**);  
 intent.setAction(Intent.***ACTION\_GET\_CONTENT***);  
 startActivityForResult(Intent.*createChooser*(intent,**"Select Picture"**), **PICK\_IMAGE\_MULTIPLE**);  
 }  
  
  
 @Override  
 **public void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **if**(requestCode == **PICK\_IMAGE\_MULTIPLE**) {  
 **if**(resultCode == Activity.***RESULT\_OK*** && data != **null**) {  
 **if**(data.getClipData() != **null**) {  
 **int** count = data.getClipData().getItemCount(); *//evaluate the count before the for loop --- otherwise, the count is evaluated every loop.* **if**(count>5)  
 count = 5;  
 **for** (**int** i = 0; i < count; i++) {  
 **paths**.add(data.getClipData().getItemAt(i).getUri().toString());  
 }  
  
 }**else** {  
 **if**(**paths**.size()<5)  
 **paths**.add(data.getData().toString());  
 }  
  
 setupImage();  
 }  
 }  
  
 }  
  
  
 **private void** setupImage() {  
 **int** count\_path = **paths**.size();  
 **if**(count\_path==5){  
 **btnChonHinhAnh**.setVisibility(View.***GONE***);  
 }**else  
 btnChonHinhAnh**.setVisibility(View.***VISIBLE***);  
  
 **listImv** = **new** ArrayList<>();  
 **listImv**.add(**imv1**);  
 **listImv**.add(**imv2**);  
 **listImv**.add(**imv3**);  
 **listImv**.add(**imv4**);  
 **listImv**.add(**imv5**);  
  
  
 **listFrm** = **new** ArrayList<>();  
 **listFrm**.add(**frm1**);  
 **listFrm**.add(**frm2**);  
 **listFrm**.add(**frm3**);  
 **listFrm**.add(**frm4**);  
 **listFrm**.add(**frm5**);  
  
 **listRemove** = **new** ArrayList<>();  
 **listRemove**.add(**imvDel1**);  
 **listRemove**.add(**imvDel2**);  
 **listRemove**.add(**imvDel3**);  
 **listRemove**.add(**imvDel4**);  
 **listRemove**.add(**imvDel5**);  
  
 *//ẨN HẾT HÌNH* **for**(**int** i=0; i<5; i++){  
 **listFrm**.get(i).setVisibility(View.***GONE***);  
 }  
  
 *//LOAD HÌNH* **for**(**int** i=0; i<count\_path; i++){  
 Picasso.*get*().load(**paths**.get(i)).into(**listImv**.get(i));  
 **listFrm**.get(i).setVisibility(View.***VISIBLE***);  
 }  
  
  
 **for** (**int** i=0; i<5; i++){  
 **final int** positon = i;  
 **listRemove**.get(i).setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **paths**.remove(positon);  
 setupImage();  
 }  
 });  
 }  
  
 }  
  
 @Override  
 **public void** loiten() {  
 **edtTen**.setError(getText(R.string.***loitenbaiviet***));  
 }  
  
 @Override  
 **public void** loidiachi() {  
 **edtDiaChi**.setError(getText(R.string.***loidiachibaiviet***));  
 }  
  
 @Override  
 **public void** loigioithieu() {  
 **edtGioiThieu**.setError(getText(R.string.***loigioithieubaiviet***));;  
 }  
  
 @Override  
 **public void** loithanhphan() {  
 Toast.*makeText*(**this**, getText(R.string.***loithanhphanbaiviet***), Toast.***LENGTH\_SHORT***).show();  
 }  
  
 @Override  
 **public void** loigiatien() {  
 **edtGiaTien**.setError(getText(R.string.***loigiabaiviet***));  
 }  
  
 @Override  
 **public void** loichonhinhanh() {  
 Toast.*makeText*(**this**, getText(R.string.***loichonhinhanhbaiviet***), Toast.***LENGTH\_SHORT***).show();  
 }  
  
 @Override  
 **public void** kiemtralogicthanhcong() {  
  
 **int** count = **mIndexThanhPhan**.size();  
 List<String> thanhphan = **new** ArrayList<>();  
 **if**(count!=0) {  
 **for** (**int** i = 0; i < count; i++) {  
 thanhphan.add(**listThanhPhan**.get(**mIndexThanhPhan**.get(i)).getMathanhphan());  
 }  
 }**else** {  
 thanhphan = **mBaiVietEdit**.getChitiet().getThanhphan();  
 }  
  
 String ten, gioithieu, gia, ghichu, diachi;  
 ten = **edtTen**.getText().toString();  
 gioithieu = **edtGioiThieu**.getText().toString();  
 gia = **edtGiaTien**.getText().toString();  
 ghichu = **edtGhiChu**.getText().toString();  
 diachi = **edtDiaChi**.getText().toString();  
  
 objchitietbaiviet chitiet = **new** objchitietbaiviet();  
 chitiet.setDiachi(diachi);  
 chitiet.setTen(ten);  
 chitiet.setGhichu(ghichu);  
 chitiet.setGiathamkhao(Long.*parseLong*(gia));  
 chitiet.setGioithieu(gioithieu);  
 chitiet.setThanhphan(thanhphan);  
  
 **mLogicChinhSua**.capnhatbaiviet(**paths**,**mBaiVietEdit**,chitiet);  
 **btnLuu**.setEnabled(**false**);  
 **mDialogLoading**.showDialog(**"Loading.."**);  
 }  
  
 @Override  
 **public void** resultDangBaiViet(**boolean** isSuccess) {  
 **mDialogLoading**.dismissDialog();  
 **if**(isSuccess){  
 Toast.*makeText*(**this**, getText(R.string.***chinhsuabaivietthanhcong***), Toast.***LENGTH\_SHORT***).show();  
 finish();  
 }  
 **else**{  
 **btnLuu**.setEnabled(**true**);  
 Toast.*makeText*(**this**, getText(R.string.***chinhsuabaivietthatbai***), Toast.***LENGTH\_SHORT***).show();  
 }  
  
 }  
}