### **khung doi**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

-------\*\*\*-------

****

**BÁO CÁO MÔN LẬP TRÌN DI ĐỘNG**

**Đề tài:** **Xây dựng ứng dụng android My Lovely Place**

**Giảng viên hướng dẫn: Nguyễn Minh Đạo**

**Nhóm thực hiện: Tào Lao Team**

**Lê Hoàng Mạnh 15110076**

**Nguyễn Công Hậu 15110043**

**TP. Hồ Chí Minh, Ngày 04 tháng 06 Năm 2018**

Link github : <https://github.com/15110076/ExamFinal_MyLovelyPlace.git>

# Chi tiết source code

## Hình cấu trúc source code

## Activity

### Màn hình Splash: Hiệu ứng khi mới mở ứng dụng

package com.example.lemanh\_dev.mylovelyplace.activity;

import android.content.Intent;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.WindowManager;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.Toast;

import com.example.lemanh\_dev.mylovelyplace.R;

import butterknife.BindView;

import butterknife.ButterKnife;

public class SplashActivity extends AppCompatActivity implements Animation.AnimationListener {

@BindView(R.id.layoutSplash)

RelativeLayout layout;

@BindView(R.id.imgIcon\_logo)

ImageView img\_icon;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_splash);

ButterKnife.bind(this);

Animation transitionAmin = AnimationUtils.loadAnimation(this,R.anim.transition\_icon);

img\_icon.setAnimation(transitionAmin);

Animation alphaAmin = AnimationUtils.loadAnimation(this,R.anim.alphe\_bacground);

layout.setAnimation(alphaAmin);

alphaAmin.setAnimationListener(this);

}

@Override

public void onAnimationStart(Animation animation) {

}

@Override

public void onAnimationEnd(Animation animation) {

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

//chuyen activity

startActivity(new Intent(SplashActivity.this,CategoriesActivity.class));

finish();//khi man hinh splash chay xong thi tat lun

}

},3000);

}

@Override

public void onAnimationRepeat(Animation animation) {

}

}

### Màn hình Categories: Các Category có trong database

package com.example.lemanh\_dev.mylovelyplace.activity;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.TextView;

import com.example.lemanh\_dev.mylovelyplace.ActivityUtils;

import com.example.lemanh\_dev.mylovelyplace.R;

import com.example.lemanh\_dev.mylovelyplace.data.PlaceRepo;

import com.example.lemanh\_dev.mylovelyplace.data.models.Category;

import java.util.ArrayList;

import java.util.List;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

public class CategoriesActivity extends AppCompatActivity {

@BindView(R.id.txtCatergories\_Restaurant)

TextView txtRestaurant;

@BindView(R.id.txtCategories\_Cinema)

TextView txtCinema;

@BindView(R.id.txtCategories\_Fashion)

TextView txtFashion;

@BindView(R.id.txtCategories\_ATM)

TextView txtATM;

private PlaceRepo placeRepo;

private List<Category> categoryList = new ArrayList<>();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_categories);

//hkoiwr tao ButterKnife

ButterKnife.bind(this);

//Hàm khởi tạo ban đầu

init();

}

private void init() {

placeRepo = PlaceRepo.getInstance(this);

categoryList=placeRepo.getAllCategory();

txtRestaurant.setText(categoryList.get(0).getCategoryName());

txtCinema.setText(categoryList.get(1).getCategoryName());

txtFashion.setText(categoryList.get(2).getCategoryName());

txtATM.setText(categoryList.get(3).getCategoryName());

}

//Click chọn Restaurant

@OnClick(R.id.layoutRestaurant)

void clickOnRestaurant(){

String categoryID = categoryList.get(0).getCategoryID();

startActivity(categoryID);

}

//Click chọn Cinema

@OnClick(R.id.layoutCinema)

void clickOnCinema(){

String categoryID = categoryList.get(1).getCategoryID();

startActivity(categoryID);

}

//Click chọn Fashion

@OnClick(R.id.layoutFashion)

void clickOnFashion(){

String categoryID = categoryList.get(2).getCategoryID();

startActivity(categoryID);

}

//Click chọn ATM

@OnClick(R.id.layoutATM)

void clickOnATM(){

String categoryID = categoryList.get(3).getCategoryID();

startActivity(categoryID);

}

//hàm xử lý start activity theo từng lại category lên trang PlaceAcitivity

private void startActivity(String categoryID){

Intent placesIntent = new Intent(CategoriesActivity.this,PlaceActivity.class);

placesIntent.putExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA,categoryID);

startActivity(placesIntent);

}

}

### Màn hình Places: danh sách Place dựa trên 1 Category nhất định

package com.example.lemanh\_dev.mylovelyplace.activity;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Handler;

import android.support.design.widget.FloatingActionButton;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import com.example.lemanh\_dev.mylovelyplace.ActivityUtils;

import com.example.lemanh\_dev.mylovelyplace.MapActivity;

import com.example.lemanh\_dev.mylovelyplace.R;

import com.example.lemanh\_dev.mylovelyplace.adapter.PlacesAdapter;

import com.example.lemanh\_dev.mylovelyplace.data.PlaceRepo;

import com.example.lemanh\_dev.mylovelyplace.data.models.Category;

import com.example.lemanh\_dev.mylovelyplace.data.models.Place;

import java.util.ArrayList;

import java.util.List;

import java.util.UUID;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

public class PlaceActivity extends AppCompatActivity {

@BindView(R.id.lvPlaceAct)

ListView lvPlaces;

@BindView(R.id.txtNoData)

TextView txtnodata;

private String categoryID;

private List<Place> places = new ArrayList<>();

private PlacesAdapter placesAdapter;

private PlaceRepo placeRepo;

private ProgressDialog progressDialog;

List<Category> categoryList;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_places);

//khởi tạo để có thể dung thư viện butterKnife

ButterKnife.bind(this);

//khởi tạo dữ liệu

init();

}

//method khởi tạo data

private void init() {

//lấy thông tin category ID từ activity CategoryActivity truyền qua

categoryID=getIntent().getStringExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA);

//getinstant class PlaceRepo

placeRepo=PlaceRepo.getInstance(this);

//đỗ danh sách place vào Adapter vì danh sách place ko thể bỏ trực tiếp

//vào ListView được phải qua trung gian là Adapter

placesAdapter=new PlacesAdapter(this,places);

//khởi tạo 1 ProgressDialog

initProgressDialog();

//hàm lấy danh sách place và bỏ vào ListView

getPlaceByCategoryID();

//

onPlaceClick();

}

//method lấy danh sách place từ category ID

private void getPlaceByCategoryID(){

//lấy dánh sách

places = placeRepo.getAllPlace(categoryID);

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

//kiểm tra danh sách có rỗng hay không

if(!places.isEmpty()){

txtnodata.setVisibility(View.GONE);

Toast.makeText(getApplicationContext(),"Load Data Successed",Toast.LENGTH\_SHORT).show();

}

//đỗ PlacesAdapter vào ListView

lvPlaces.setAdapter(placesAdapter);

//cập nhật lại dánh sách PlacesAdapter nếu có thay đổi xảy ra

placesAdapter.updatePlace(places);

//Hủy progressDialog

progressDialog.dismiss();

}

},3000);

}

//method khởi tạo ProgressDialog

private void initProgressDialog(){

progressDialog=new ProgressDialog(PlaceActivity.this);

progressDialog.setIndeterminate(true);

progressDialog.setMessage(getResources().getString(R.string.text\_retrieving\_data));

progressDialog.setCanceledOnTouchOutside(false);

progressDialog.show();

}

//Xử lý sự kiện click của từng item trong listview

//khi click sẽ put thông tin category ID vs Place ID qua activity datailPlace

private void onPlaceClick(){

lvPlaces.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

Place place = places.get(position);

Intent detailPlace = new Intent(PlaceActivity.this,DetailPlaceActivity.class);

detailPlace.putExtra(ActivityUtils.PLACE\_KEY\_PUT\_EXTRA,place.getPlaceID());

detailPlace.putExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA,place.getCategoryID());

startActivity(detailPlace);

}

});

}

//Xử lý sự kiện click thêm 1 place mới với 1 category thuộc danh sách hiện tại

//Put CategoryID qua để xác định thêm Place theo Category nào

//và 1 cái cờ xác định là dung chức năng insert hay update

@OnClick(R.id.btnAddNewPlace)

public void addNewPlace(){

Intent addPlace = new Intent(PlaceActivity.this,AddPlaceActivity.class);

addPlace.putExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA,categoryID);

addPlace.putExtra(ActivityUtils.FLAG\_IS\_INSERT\_OR\_UPDATE,true);

startActivity(addPlace);

}

//Xử lý sự kiện hiện thị tất cả địa điểm có trong danh sách lên Map

//Put CategoryID qua Activity MapActivity để lấy danh sách place theo categoryID

@OnClick(R.id.btnShowOnMap)

public void showAllPlaceOnMap(){

Intent intent = new Intent(this, MapActivity.class);

intent.putExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA,categoryID);

startActivity(intent);

}

}

### Màn hình Detail Place: xem chi tiết 1 place

package com.example.lemanh\_dev.mylovelyplace.activity;

import android.app.AlertDialog;

import android.app.ProgressDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.example.lemanh\_dev.mylovelyplace.ActivityUtils;

import com.example.lemanh\_dev.mylovelyplace.R;

import com.example.lemanh\_dev.mylovelyplace.data.PlaceRepo;

import com.example.lemanh\_dev.mylovelyplace.data.models.Place;

import java.util.UUID;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

public class DetailPlaceActivity extends AppCompatActivity {

@BindView(R.id.detailPlace\_imgPlace)

ImageView imgPlace;

@BindView(R.id.detalPlace\_txtPlaceName)

EditText txtplaceName;

@BindView(R.id.detailPlace\_txtPlaceAddress)

EditText txtplaceAddress;

@BindView(R.id.detailPlace\_txtPlaceDescription)

EditText txtplaceDescription;

private String placeID;

private String categoryID;

private PlaceRepo placeRepo;

private ProgressDialog progressDialog;

private ProgressDialog progressDialogDelete;

private ProgressDialog progressDialogUpdate;

private Place place;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_detail\_place);

//khởi tao ButterKnife

ButterKnife.bind(this);

//hàm khởi tạo

init();

}

private void init() {

//Lấy placeID

placeID =getIntent().getStringExtra(ActivityUtils.PLACE\_KEY\_PUT\_EXTRA);

//lấy CategoryID

categoryID=getIntent().getStringExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA);

placeRepo=PlaceRepo.getInstance(this);

//log xem thông tin lấy được hay ko

Log.d("Test",placeID +" "+ categoryID);

initProgressDialog();

setPlace();

}

//method set Place

private void setPlace(){

//lấy danh sách Place dựa trên placeiD và categoryID

place = placeRepo.getPlaceByID(categoryID,placeID);

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

if(place.getPlaceImages() != null) {

Bitmap placeBitmap = BitmapFactory.decodeByteArray(place.getPlaceImages(), 0, place.getPlaceImages().length);

imgPlace.setImageBitmap(placeBitmap);

}

txtplaceName.setText(place.getPlaceName());

txtplaceAddress.setText(place.getPlaceAddress());

txtplaceDescription.setText(place.getPlaceDescription());

progressDialog.dismiss();

}

},4000);

}

//method khởi tạo ProgressDialog cho setup place

private void initProgressDialog(){

progressDialog=new ProgressDialog(DetailPlaceActivity.this);

progressDialog.setIndeterminate(true);

progressDialog.setMessage(getResources().getString(R.string.text\_retrieving\_data));

progressDialog.setCanceledOnTouchOutside(false);

progressDialog.show();

}

//method khởi tạo ProgressDialog cho delete

private void initProgressDialogdelete(){

progressDialogDelete=new ProgressDialog(DetailPlaceActivity.this);

progressDialogDelete.setIndeterminate(true);

progressDialogDelete.setMessage(getResources().getString(R.string.text\_deleting\_data));

progressDialogDelete.setCanceledOnTouchOutside(false);

progressDialogDelete.show();

}

//method khởi tạo ProgressDialog cho update

private void initProgressDialogupdate(){

progressDialogUpdate=new ProgressDialog(DetailPlaceActivity.this);

progressDialogUpdate.setIndeterminate(true);

progressDialogUpdate.setMessage(getResources().getString(R.string.text\_deleting\_data));

progressDialogUpdate.setCanceledOnTouchOutside(false);

progressDialogUpdate.show();

}

//Xử lý sự kiện back về Activity trước đó

@OnClick(R.id.btnClose)

public void closeActivitiDetail(){

//chuển đên Activity PlaceActivity

redirecttoListPlace();

}

//xử lý chức năng Delete 1 place

@OnClick(R.id.btnDelete)

public void deletePlace(View view){

//khởi tạo ProgressDialog

initProgressDialogdelete();

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

//Delete place theo PlaceID

placeRepo.delete(placeID);

//hủy progressDialog

progressDialogDelete.dismiss();

redirecttoListPlace();

}

},3000);

}

//Xử lý sự kiện click edit 1 place với 1 category thuôc danh sách hiện tại

//Put CategoryID qua để xác định thêm Place theo Category nào

//và 1 cái cờ xác định là dung chức năng insert hay update

@OnClick(R.id.btnEdit)

public void editPlace(View view){

Intent placeEditIntent = new Intent(DetailPlaceActivity.this,AddPlaceActivity.class);

placeEditIntent.putExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA,categoryID);

placeEditIntent.putExtra(ActivityUtils.PLACE\_KEY\_PUT\_EXTRA,placeID);

placeEditIntent.putExtra(ActivityUtils.FLAG\_IS\_INSERT\_OR\_UPDATE,false);

startActivity(placeEditIntent);

finish();

}

//chuyen huong den trang danh sanh place

private void redirecttoListPlace(){

Intent placeIntent = new Intent(DetailPlaceActivity.this,PlaceActivity.class);

placeIntent.putExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA,categoryID);

startActivity(placeIntent);

finish();

}

}

### Màn hình Thêm mới hoặc Cập nhật 1 Place

#### Class AddPlaceActivity

package com.example.lemanh\_dev.mylovelyplace.activity;

import android.app.ProgressDialog;

import android.content.Intent;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.graphics.drawable.BitmapDrawable;

import android.os.AsyncTask;

import android.os.Handler;

import android.provider.MediaStore;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.text.TextUtils;

import android.util.Log;

import android.view.View;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import com.example.lemanh\_dev.mylovelyplace.ActivityUtils;

import com.example.lemanh\_dev.mylovelyplace.HttpDataHandler;

import com.example.lemanh\_dev.mylovelyplace.R;

import com.example.lemanh\_dev.mylovelyplace.data.PlaceRepo;

import com.example.lemanh\_dev.mylovelyplace.data.models.Place;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.io.ByteArrayOutputStream;

import java.util.UUID;

import butterknife.BindView;

import butterknife.ButterKnife;

import butterknife.OnClick;

import static com.example.lemanh\_dev.mylovelyplace.ActivityUtils.FLAG\_IS\_INSERT\_OR\_UPDATE;

public class AddPlaceActivity extends AppCompatActivity {

@BindView(R.id.titleAct)

TextView txtTitleAct;

@BindView(R.id.txtLat)

TextView txtlat;

@BindView(R.id.txtLng)

TextView txtlng;

@BindView(R.id.addPlace\_txtPlaceAddress)

EditText txtPlaceAddress;

@BindView(R.id.addPlace\_txtPlaceName)

EditText txtPlacename;

@BindView(R.id.addPlace\_txtPlaceDescription)

EditText txtPlacedescription;

@BindView(R.id.addPlace\_imgPlace)

ImageView imagePlace;

private static final int REQUEST\_CODE =1;

private String categoryID;

private String placeDetailID;

//cợ xác định có chụp ảnh hay chưa

private boolean isTaken = false;

private ProgressDialog progressDialog;

private PlaceRepo placeRepo;

private Place placeDetail;

//cờ xác định insert hay update

private boolean flag = true ;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_add\_place);

ButterKnife.bind(this);

//lấy thtông in cờ là Insert hay Update

flag = getIntent().getExtras().getBoolean(FLAG\_IS\_INSERT\_OR\_UPDATE);

//true là Insert và ngược lại

if(flag){

initInsert();

}

else {

initUpdate();

}

}

//khởi tạo khi chức năng là insert

private void initInsert() {

categoryID = getIntent().getStringExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA);

txtTitleAct.setText(getResources().getText(R.string.text\_title\_act\_insert));

placeRepo= PlaceRepo.getInstance(this);

initProgressDialog();

}

//khởi tạo khi chức năng là update

private void initUpdate() {

categoryID = getIntent().getStringExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA);

placeDetailID = getIntent().getStringExtra(ActivityUtils.PLACE\_KEY\_PUT\_EXTRA);

txtTitleAct.setText(getResources().getText(R.string.text\_title\_act\_update));

placeRepo= PlaceRepo.getInstance(this);

initProgressDialog();

progressDialog.show();

setPlace();

}

//set place khi chức năng là update

private void setPlace(){

placeDetail = placeRepo.getPlaceByID(categoryID,placeDetailID);

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

if(placeDetail.getPlaceImages() != null) {

Bitmap placeBitmap = BitmapFactory.decodeByteArray(placeDetail.getPlaceImages(), 0, placeDetail.getPlaceImages().length);

imagePlace.setImageBitmap(placeBitmap);

}

txtPlacename.setText(placeDetail.getPlaceName());

txtPlaceAddress.setText(placeDetail.getPlaceAddress());

txtPlacedescription.setText(placeDetail.getPlaceDescription());

txtlat.setText(String.valueOf(placeDetail.getPlaceLat()));

txtlng.setText(String.valueOf(placeDetail.getPlaceLng()));

progressDialog.dismiss();

}

},4000);

}

//khởi tạo Progressdialog

private void initProgressDialog() {

progressDialog=new ProgressDialog(AddPlaceActivity.this);

progressDialog.setIndeterminate(true);

progressDialog.setMessage(getResources().getString(R.string.text\_saving\_data));

progressDialog.setCanceledOnTouchOutside(false);

}

//lay hinh anh chup duoc

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

//nếu chụp ảnh rồi

if(requestCode == REQUEST\_CODE && resultCode == RESULT\_OK){

//kiểm tra dữ liệu trả về có null hay ko

if(data ==null){

isTaken =false;

}

else {

//cờ đã chụp ảnh rồi thành true

isTaken=true;

//lấy ảnh vữa chuoj bỏ lên Image Place

Bitmap placeImg = (Bitmap) data.getExtras().get("data");

imagePlace.setImageBitmap(placeImg);

}

}

}

//lay thong tin lat va lng

@OnClick(R.id.addPlace\_getLocation)

public void getLatLng(View v){

new GetCoordinates().execute(txtPlaceAddress.getText().toString().replace(" ","+"));

}

//chup anh tu camera

@OnClick(R.id.addPlace\_imgPlace)

public void openCamera(View v){

Intent cameraIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

startActivityForResult(cameraIntent,REQUEST\_CODE);

}

//tro ve activity truoc

@OnClick(R.id.btnClose)

public void goBack(){

if(flag){

//về activity PlaceActivity

redirecttoListPlace();

}

else {

//về activity DetailActivity

redirecttoDetailPlace();

}

}

//convert imageview to byte[]

private byte[] convertImageToByte(ImageView imageView){

Bitmap bitmap = ((BitmapDrawable) imageView.getDrawable()).getBitmap();

ByteArrayOutputStream baos = new ByteArrayOutputStream();

bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);

byte[] imageInByte = baos.toByteArray();

return imageInByte;

}

//xu ly su kien them moi 1 dia diem hoặc update 1 địa điểm

@OnClick(R.id.btnAddSave)

public void clickSave(View v){

if(flag) {

if (isTaken) {

final String placeName = txtPlacename.getText().toString();

final String placeAddress = txtPlaceAddress.getText().toString();

final String placeDescription = txtPlacedescription.getText().toString();

final String placeID = UUID.randomUUID().toString();

final byte[] placeImage = convertImageToByte(imagePlace);

if (Place.valiadteInput(categoryID, placeName, placeAddress, placeDescription)) {

if (TextUtils.isEmpty(txtlat.getText()) && TextUtils.isEmpty(txtlng.getText())) {

Toast.makeText(getApplicationContext(), "Lat and Lng current null. Please get location!!!", Toast.LENGTH\_SHORT).show();

} else {

final double placeLat = Double.parseDouble(txtlat.getText().toString());

final double placeLng = Double.parseDouble(txtlng.getText().toString());

progressDialog.show();

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

Place place = new Place.Builder()

.setPlaceID(placeID)

.setCategoryID(categoryID)

.setPlaceName(placeName)

.setPlaceAddress(placeAddress)

.setPlaceDescription(placeDescription)

.setPlaceImages(placeImage)

.setPlaceLat(placeLat)

.setPlaceLng(placeLng)

.buil();

placeRepo.insert(place);

progressDialog.dismiss();

redirecttoListPlace();

}

}, 3000);

}

} else {

Toast.makeText(getApplicationContext(), "Please fill in place's inform", Toast.LENGTH\_SHORT).show();

}

} else {

Toast.makeText(getApplicationContext(), "Please take a picture!!!", Toast.LENGTH\_SHORT).show();

}

}

else {

final String placeName = txtPlacename.getText().toString();

final String placeAddress = txtPlaceAddress.getText().toString();

final String placeDescription = txtPlacedescription.getText().toString();

final byte[] placeImage = convertImageToByte(imagePlace);

if (Place.valiadteInput(categoryID, placeName, placeAddress, placeDescription)) {

if (TextUtils.isEmpty(txtlat.getText()) && TextUtils.isEmpty(txtlng.getText())) {

Toast.makeText(getApplicationContext(), "Lat and Lng current null. Please get location!!!", Toast.LENGTH\_SHORT).show();

} else {

final double placeLat = Double.parseDouble(txtlat.getText().toString());

final double placeLng = Double.parseDouble(txtlng.getText().toString());

progressDialog.show();

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

Place place = new Place.Builder()

.setPlaceID(placeDetailID)

.setCategoryID(categoryID)

.setPlaceName(placeName)

.setPlaceAddress(placeAddress)

.setPlaceDescription(placeDescription)

.setPlaceImages(placeImage)

.setPlaceLat(placeLat)

.setPlaceLng(placeLng)

.buil();

placeRepo.update(place);

progressDialog.dismiss();

redirecttoDetailPlace();

}

}, 3000);

}

} else {

Toast.makeText(getApplicationContext(), "Please fill in place's inform", Toast.LENGTH\_SHORT).show();

}

}

}

//chuyen huong den trang chi tiet san place vua chon chinh sua

private void redirecttoDetailPlace(){

Intent detailPlaceIntent = new Intent(AddPlaceActivity.this,DetailPlaceActivity.class);

detailPlaceIntent.putExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA,categoryID);

detailPlaceIntent.putExtra(ActivityUtils.PLACE\_KEY\_PUT\_EXTRA,placeDetailID);

startActivity(detailPlaceIntent);

finish();

}

//chuyen huong den trang danh sanh place

private void redirecttoListPlace(){

Intent placeIntent = new Intent(AddPlaceActivity.this,PlaceActivity.class);

placeIntent.putExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA,categoryID);

startActivity(placeIntent);

finish();

}

//xu ly lay kinh do va vi do cua dia chi

private class GetCoordinates extends AsyncTask<String,Void,String> {

ProgressDialog dialog = new ProgressDialog(AddPlaceActivity.this);

//khi vựa nhấn Get loaction

//mở progessDialog

@Override

protected void onPreExecute() {

super.onPreExecute();

dialog.setMessage("Please wait....");

dialog.setCanceledOnTouchOutside(false);

dialog.show();

}

//bắt đấu lấy dữ liệu

@Override

protected String doInBackground(String... strings) {

String response;

try{

String address = strings[0];

HttpDataHandler http = new HttpDataHandler();

String url = String.format("https://maps.googleapis.com/maps/api/geocode/json?address=%s",address);

response = http.getHTTPData(url);

return response;

}

catch (Exception ex)

{

}

return null;

}

//post lên giao diện

//và tắt progressDialog

@Override

protected void onPostExecute(String s) {

try{

JSONObject jsonObject = new JSONObject(s);

String lat = ((JSONArray)jsonObject.get("results")).getJSONObject(0).getJSONObject("geometry")

.getJSONObject("location").get("lat").toString();

String lng = ((JSONArray)jsonObject.get("results")).getJSONObject(0).getJSONObject("geometry")

.getJSONObject("location").get("lng").toString();

txtlat.setText(String.format("%s",lat));

txtlng.setText(String.format("%s",lng));

if(dialog.isShowing())

dialog.dismiss();

} catch (JSONException e) {

e.printStackTrace();

}

}

}

}

#### Class HttpDataHandler

package com.example.lemanh\_dev.mylovelyplace;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.ProtocolException;

import java.net.URL;

import javax.net.ssl.HttpsURLConnection;

public class HttpDataHandler {

public HttpDataHandler() {

}

public String getHTTPData(String requestURL)

{

URL url;

String response = "";

try{

url = new URL(requestURL);

HttpURLConnection conn = (HttpURLConnection)url.openConnection();

conn.setRequestMethod("GET");

conn.setReadTimeout(15000);

conn.setConnectTimeout(15000);

conn.setDoInput(true);

conn.setDoOutput(true);

conn.setRequestProperty("Content-Type","application/x-www-form-urlencoded");

int responseCode = conn.getResponseCode();

if(responseCode == HttpsURLConnection.HTTP\_OK)

{

String line;

BufferedReader br = new BufferedReader(new InputStreamReader(conn.getInputStream()));

while((line = br.readLine()) != null)

response+=line;

}

else

response = "";

} catch (ProtocolException e) {

e.printStackTrace();

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return response;

}

}

### Màn hình Map:Xem các location Place trong 1 Category nhất định

package com.example.lemanh\_dev.mylovelyplace;

import android.app.ProgressDialog;

import android.os.Handler;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import com.example.lemanh\_dev.mylovelyplace.activity.PlaceActivity;

import com.example.lemanh\_dev.mylovelyplace.data.PlaceRepo;

import com.example.lemanh\_dev.mylovelyplace.data.models.Place;

import com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;

import com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.MarkerOptions;

import java.util.List;

public class MapActivity extends AppCompatActivity implements OnMapReadyCallback{

private GoogleMap googleMap;

private PlaceRepo placeRepo;

private String categoryID;

private List<Place> places;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_map);

//method khởi tạo

init();

// Nhận thông báo SupportMapFragment và yêu cầu

// khi bản đồ đã sẵn sàng để sử dụng.

SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()

.findFragmentById(R.id.map);

mapFragment.getMapAsync(this);

}

private void init() {

//lấy thông tin category ID bên activity Place gửi qua

categoryID = getIntent().getStringExtra(ActivityUtils.CATEGORY\_KEY\_PUT\_EXTRA);

//get instant class PlceRepo

placeRepo=PlaceRepo.getInstance(this);

//lấy danh sách place tương úng với category vừa nhận được

getPlaces();

}

//method lấy danh sách place

private void getPlaces() {

places=placeRepo.getAllPlace(categoryID);

}

@Override

public void onMapReady(final GoogleMap googleMap) {

this.googleMap=googleMap;

//duyệt từng place trong danh sách places

//lấy thông tin lat va lng tạo thành 1 location trên map

for (Place place:places){

//setup latlng

LatLng locationPlace = new LatLng(place.getPlaceLat(), place.getPlaceLng());

//thêm 1 marker trên map và sat title cho nó là PlaceName

googleMap.addMarker(new MarkerOptions().position(locationPlace)

.title(place.getPlaceName()));

//di chuyển camera đến location mới thêm gần nhất

googleMap.moveCamera(CameraUpdateFactory.newLatLng(locationPlace));

}

}

}

## Adapter

### Class PlaceAdaper.java:Tạo liên kết giữa 1 ập hợp dữ liệu(ListCategory, ListPlace) với 1 AdapterView(ListView)

package com.example.lemanh\_dev.mylovelyplace.adapter;

import android.content.Context;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.ImageView;

import android.widget.ListView;

import android.widget.TextView;

import com.example.lemanh\_dev.mylovelyplace.R;

import com.example.lemanh\_dev.mylovelyplace.data.models.Category;

import com.example.lemanh\_dev.mylovelyplace.data.models.Place;

import java.util.List;

import butterknife.BindView;

import butterknife.ButterKnife;

public class PlacesAdapter extends BaseAdapter {

private Context context;

private List<Place> places;

public PlacesAdapter(Context context, List<Place> places) {

this.context = context;

setPlaces(places);

}

private void setPlaces(List<Place> places){

this.places=places;

}

public void updatePlace(List<Place> places){

this.places=places;

notifyDataSetChanged();

}

@Override

public int getCount() {

return places.size();

}

@Override

public Object getItem(int position) {

return places.get(position);

}

@Override

public long getItemId(int position) {

return position;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

PlaceViewHolder viewHolder = new PlaceViewHolder();

LayoutInflater layoutInflater = (LayoutInflater) context.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);

if(convertView==null){

convertView = layoutInflater.inflate(R.layout.itemplace,parent,false);

ButterKnife.bind(viewHolder,convertView);

convertView.setTag(viewHolder);

}

viewHolder= (PlaceViewHolder) convertView.getTag();

Place place = (Place) getItem(position);

if(place.getPlaceImages() != null) {

Bitmap placeBitmap = BitmapFactory.decodeByteArray(place.getPlaceImages(), 0, place.getPlaceImages().length);

viewHolder.imgPlace.setImageBitmap(placeBitmap);

}

viewHolder.txtPlaceName.setText(place.getPlaceName());

viewHolder.txtPlaceAddress.setText(place.getPlaceAddress());

viewHolder.txtPlaceDescriptio.setText(place.getPlaceDescription());

return convertView;

}

class PlaceViewHolder {

@BindView(R.id.imgItemPlace)

ImageView imgPlace;

@BindView(R.id.txtPlaceName)

TextView txtPlaceName;

@BindView(R.id.txtPlaceAddress)

TextView txtPlaceAddress;

@BindView(R.id.txtPlaceDescription)

TextView txtPlaceDescriptio;

}

}

## Model

### Class Category.java: 1 bảng Category

package com.example.lemanh\_dev.mylovelyplace.data.models;

public class Category {

private String categoryID;

private String categoryName;

public Category(String categoryID, String categoryName) {

this.categoryID = categoryID;

this.categoryName = categoryName;

}

public String getCategoryID() {

return categoryID;

}

public void setCategoryID(String categoryID) {

this.categoryID = categoryID;

}

public String getCategoryName() {

return categoryName;

}

public void setCategoryName(String categoryName) {

this.categoryName = categoryName;

}

}

### Class Place.java: 1 bảng Place có fung design pattern Builder

package com.example.lemanh\_dev.mylovelyplace.data.models;

import android.text.TextUtils;

public class Place {

private String placeID;

private String CategoryID;

private String placeName;

private String placeAddress;

private String placeDescription;

private byte[] placeImages;

private double placeLat;

private double placeLng;

public Place(Builder builder) {

this.placeID = builder.placeID;

this.CategoryID = builder.CategoryID;

this.placeName = builder.placeName;

this.placeAddress = builder.placeAddress;

this.placeDescription = builder.placeDescription;

this.placeImages = builder.placeImages;

this.placeLat = builder.placeLat;

this.placeLng = builder.placeLng;

}

public String getPlaceID() {

return placeID;

}

public String getCategoryID() {

return CategoryID;

}

public String getPlaceName() {

return placeName;

}

public String getPlaceAddress() {

return placeAddress;

}

public String getPlaceDescription() {

return placeDescription;

}

public byte[] getPlaceImages() {

return placeImages;

}

public double getPlaceLat() {

return placeLat;

}

public double getPlaceLng() {

return placeLng;

}

public static class Builder {

private String placeID;

private String CategoryID;

private String placeName;

private String placeAddress;

private String placeDescription;

private byte[] placeImages;

private double placeLat;

private double placeLng;

public Builder setPlaceID(String placeID) {

this.placeID = placeID;

return this;

}

public Builder setCategoryID(String categoryID) {

CategoryID = categoryID;

return this;

}

public Builder setPlaceName(String placeName) {

this.placeName = placeName;

return this;

}

public Builder setPlaceAddress(String placeAddress) {

this.placeAddress = placeAddress;

return this;

}

public Builder setPlaceDescription(String placeDescription) {

this.placeDescription = placeDescription;

return this;

}

public Builder setPlaceImages(byte[] placeImages) {

this.placeImages = placeImages;

return this;

}

public Builder setPlaceLat(double placeLat) {

this.placeLat = placeLat;

return this;

}

public Builder setPlaceLng(double placeLng) {

this.placeLng = placeLng;

return this;

}

public Place buil(){

return new Place(this);

}

}

public static boolean valiadteInput(String categoryID, String placeName, String placeAddress, String placeDescription){

return (TextUtils.isEmpty(categoryID) || TextUtils.isEmpty(placeName) ||TextUtils.isEmpty(placeAddress)

|| TextUtils.isEmpty(placeDescription)) ? false : true;

}

}

## Utils

### Utils Database : các biến tĩnh trong database

package com.example.lemanh\_dev.mylovelyplace.data.models;

public final class DB\_Utitls {

public static final String PLACE\_TBL\_NAME = "PLACE";

public static final String COLUMN\_PLACE\_ID = "PalceID";

public static final String COLUMN\_PLACE\_CATEGORY\_ID = "PalceCategoryID";

public static final String COLUMN\_PLACE\_NAME = "PlaceName";

public static final String COLUMN\_PLACE\_ADDRESS = "PalceAddress";

public static final String COLUMN\_PLACE\_DESCRIPTION = "PlaceDescription";

public static final String COLUMN\_PLACE\_IMAGES = "PalceImages";

public static final String COLUMN\_PLACE\_LAT = "PlaceLat";

public static final String COLUMN\_PLACE\_LNG = "PalceLng";

public static final String CATEGORY\_TBL\_NAME = "CATEGORY";

public static final String COLUMN\_CATEGORY\_ID = "CategoryID";

public static final String COLUMN\_CATEGORY\_NAME = "CategoryName";

public static final String PRIMARY\_KEY = "PRIMARY KEY";

public static final String NOT\_NULL = "NOT NULL";

public static final String NULL = "NULL";

public static final String TEXT\_DATA\_TYPE = "TEXT";

public static final String REAL\_DATA\_TYPE = "REAL";

public static final String BLOB\_DATA\_TYPE = "BLOB";

}

### Utils Activity: các biến tĩnh dùng trong các Activity

package com.example.lemanh\_dev.mylovelyplace;

public class ActivityUtils {

public static final String CATEGORY\_KEY\_PUT\_EXTRA = "CATEGORY\_ID";

public static final String PLACE\_KEY\_PUT\_EXTRA = "PLACE\_ID";

public static final String FLAG\_IS\_INSERT\_OR\_UPDATE = "INSERT\_OR\_UPDATE";

}

## SQLITE

### Class SQLiteHelper: Tạo 1 csdl trên SQLite

package com.example.lemanh\_dev.mylovelyplace.data;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import com.example.lemanh\_dev.mylovelyplace.data.models.DB\_Utitls;

public class PlaceSqliteHelper extends SQLiteOpenHelper {

private static final String DB\_NAME = "PLACE";

private static final int DB\_VERSION = 1;

private static final String CREATE\_PLACE\_TABLE\_SQL =

"CREATE TABLE " + DB\_Utitls.PLACE\_TBL\_NAME + "("

+DB\_Utitls.COLUMN\_PLACE\_ID + " " +DB\_Utitls.TEXT\_DATA\_TYPE + " " +DB\_Utitls.PRIMARY\_KEY +", "

+DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID + " " +DB\_Utitls.TEXT\_DATA\_TYPE + " " +DB\_Utitls.NOT\_NULL +", "

+DB\_Utitls.COLUMN\_PLACE\_NAME + " " +DB\_Utitls.TEXT\_DATA\_TYPE + " " +DB\_Utitls.NOT\_NULL +", "

+DB\_Utitls.COLUMN\_PLACE\_ADDRESS + " " +DB\_Utitls.TEXT\_DATA\_TYPE + " " +DB\_Utitls.NOT\_NULL +", "

+DB\_Utitls.COLUMN\_PLACE\_DESCRIPTION + " " +DB\_Utitls.TEXT\_DATA\_TYPE + " " +DB\_Utitls.NOT\_NULL +", "

+DB\_Utitls.COLUMN\_PLACE\_IMAGES + " " +DB\_Utitls.BLOB\_DATA\_TYPE + " " +DB\_Utitls.NOT\_NULL +", "

+DB\_Utitls.COLUMN\_PLACE\_LAT + " " +DB\_Utitls.REAL\_DATA\_TYPE + " " +DB\_Utitls.NOT\_NULL +", "

+DB\_Utitls.COLUMN\_PLACE\_LNG + " " +DB\_Utitls.REAL\_DATA\_TYPE + " " +DB\_Utitls.NOT\_NULL

+")";

private static final String CREATE\_CATEGORY\_TABLE\_SQL =

"CREATE TABLE " + DB\_Utitls.CATEGORY\_TBL\_NAME + "("

+DB\_Utitls.COLUMN\_CATEGORY\_ID + " " +DB\_Utitls.TEXT\_DATA\_TYPE + " " +DB\_Utitls.PRIMARY\_KEY +", "

+DB\_Utitls.COLUMN\_CATEGORY\_NAME + " " +DB\_Utitls.TEXT\_DATA\_TYPE + " " +DB\_Utitls.NOT\_NULL

+")";

private static final String INSERT\_CATEGORY\_SQL =

"INSERT INTO " + DB\_Utitls.CATEGORY\_TBL\_NAME + " (" +DB\_Utitls.COLUMN\_CATEGORY\_ID+", "+ DB\_Utitls.COLUMN\_CATEGORY\_NAME +")"

+ " VALUES "

+ "('1', 'Restaurant'), "+"('2', 'Cinema'), "+"('3', 'Fashion'), "+"('4', 'ATM')";

public PlaceSqliteHelper(Context context) {

super(context, DB\_NAME, null, DB\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL(CREATE\_PLACE\_TABLE\_SQL);

db.execSQL(CREATE\_CATEGORY\_TABLE\_SQL);

db.execSQL(INSERT\_CATEGORY\_SQL);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

}

}

### Class PlaceRepo : các hàm xử lý dữ liệu tới database

package com.example.lemanh\_dev.mylovelyplace.data;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import com.example.lemanh\_dev.mylovelyplace.data.models.Category;

import com.example.lemanh\_dev.mylovelyplace.data.models.DB\_Utitls;

import com.example.lemanh\_dev.mylovelyplace.data.models.Place;

import java.util.ArrayList;

import java.util.List;

public class PlaceRepo {

private static PlaceRepo instance = null;

private PlaceSqliteHelper placeSqliteHelper;

//design pattern SingleTon

protected PlaceRepo(Context context) {

// Exists only to defeat instantiation.

placeSqliteHelper = new PlaceSqliteHelper(context);

}

public static PlaceRepo getInstance(Context context) {

return(instance == null)? new PlaceRepo(context) :instance;

}

//lay danh sach Category

public List<Category> getAllCategory(){

List<Category> lstCategory = new ArrayList<>();

//quyền read

SQLiteDatabase db = placeSqliteHelper.getReadableDatabase();

//các cột trong bảng Category

String[] columns = {

DB\_Utitls.COLUMN\_CATEGORY\_ID,

DB\_Utitls.COLUMN\_CATEGORY\_NAME

};

//lấy dữ liệu bỏ vào 1 cursor

Cursor cursor = db.query(DB\_Utitls.CATEGORY\_TBL\_NAME,columns,null,null,null,null,null);

//kiểm tra cursor vừa lấy được có dữ liệu hay không

if(cursor != null && cursor.getCount() >0){

while (cursor.moveToNext())

{

String categoryID = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_CATEGORY\_ID));

String categoryName =cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_CATEGORY\_NAME));

lstCategory.add(new Category(categoryID,categoryName));

}

}

//đóng cursor lại

if(cursor!= null){

cursor.close();

}

//đóng kết nối

db.close();

return lstCategory ;

}

//lay danh sach Place theo categoryID

public List<Place> getAllPlace1(){

List<Place> lstPlace = new ArrayList<>();

//mở kết nối database quyền read

SQLiteDatabase db = placeSqliteHelper.getReadableDatabase();

//các cột trong bản Place

String[] columns = {

DB\_Utitls.COLUMN\_PLACE\_ID,

DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID,

DB\_Utitls.COLUMN\_PLACE\_NAME,

DB\_Utitls.COLUMN\_PLACE\_ADDRESS,

DB\_Utitls.COLUMN\_PLACE\_DESCRIPTION,

DB\_Utitls.COLUMN\_PLACE\_IMAGES,

DB\_Utitls.COLUMN\_PLACE\_LAT,

DB\_Utitls.COLUMN\_PLACE\_LNG

};

//tạo một cursor lưu result cuả kết quả trả về sau khi thực hiện câu truy vấn

Cursor cursor = db.query(DB\_Utitls.PLACE\_TBL\_NAME,columns,null,null,null,null,null);

//kiểm tra cursor có dữ liệu hay ko

if(cursor != null && cursor.getCount() >0){

while (cursor.moveToNext())

{

String placeID = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_ID));

String categoryID = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID));

String placeName =cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_NAME));

String placeAddress = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_ADDRESS));

String placeDescription = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_DESCRIPTION));

byte[] placeImages = cursor.getBlob(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_IMAGES));

double placeLat = cursor.getDouble(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_LAT));

double placeLng = cursor.getDouble(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_LNG));

//khởi tạo 1 Place

Place place = new Place.Builder()

.setPlaceID(placeID)

.setCategoryID(categoryID)

.setPlaceName(placeName)

.setPlaceAddress(placeAddress)

.setPlaceDescription(placeDescription)

.setPlaceImages(placeImages)

.setPlaceLat(placeLat)

.setPlaceLng(placeLng)

.buil();

lstPlace.add(place);

}

}

//đóng cursor

if(cursor!= null){

cursor.close();

}

//đóng kết nôid

db.close();

return lstPlace ;

}

//lay danh sach Place theo categoryID

public List<Place> getAllPlace(String cateID){

List<Place> lstPlace = new ArrayList<>();

SQLiteDatabase db = placeSqliteHelper.getReadableDatabase();

String[] columns = {

DB\_Utitls.COLUMN\_PLACE\_ID,

DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID,

DB\_Utitls.COLUMN\_PLACE\_NAME,

DB\_Utitls.COLUMN\_PLACE\_ADDRESS,

DB\_Utitls.COLUMN\_PLACE\_DESCRIPTION,

DB\_Utitls.COLUMN\_PLACE\_IMAGES,

DB\_Utitls.COLUMN\_PLACE\_LAT,

DB\_Utitls.COLUMN\_PLACE\_LNG

};

String selection = DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID + " = ?";

String[] selectionArgs = {cateID};

Cursor cursor = db.query(DB\_Utitls.PLACE\_TBL\_NAME,columns,selection,selectionArgs,null,null,null);

if(cursor != null && cursor.getCount() >0){

while (cursor.moveToNext())

{

String placeID = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_ID));

String categoryID = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID));

String placeName =cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_NAME));

String placeAddress = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_ADDRESS));

String placeDescription = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_DESCRIPTION));

byte[] placeImages = cursor.getBlob(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_IMAGES));

double placeLat = cursor.getDouble(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_LAT));

double placeLng = cursor.getDouble(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_LNG));

Place place = new Place.Builder()

.setPlaceID(placeID)

.setCategoryID(categoryID)

.setPlaceName(placeName)

.setPlaceAddress(placeAddress)

.setPlaceDescription(placeDescription)

.setPlaceImages(placeImages)

.setPlaceLat(placeLat)

.setPlaceLng(placeLng)

.buil();

lstPlace.add(place);

}

}

if(cursor!= null){

cursor.close();

}

db.close();

return lstPlace ;

}

//lay thong tin Place theo placeID

public Place getPlaceByID(String cateID, String plID){

Place place =null;

SQLiteDatabase db = placeSqliteHelper.getReadableDatabase();

String[] columns = {

DB\_Utitls.COLUMN\_PLACE\_ID,

DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID,

DB\_Utitls.COLUMN\_PLACE\_NAME,

DB\_Utitls.COLUMN\_PLACE\_ADDRESS,

DB\_Utitls.COLUMN\_PLACE\_DESCRIPTION,

DB\_Utitls.COLUMN\_PLACE\_IMAGES,

DB\_Utitls.COLUMN\_PLACE\_LAT,

DB\_Utitls.COLUMN\_PLACE\_LNG

};

String selection = DB\_Utitls.COLUMN\_PLACE\_ID + " = ? AND " + DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID + " = ?";

String[] selectionArgs = {plID, cateID};

Cursor cursor = db.query(DB\_Utitls.PLACE\_TBL\_NAME,columns,selection,selectionArgs,null,null,null);

if(cursor != null && cursor.getCount() >0){

cursor.moveToFirst();

String placeID = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_ID));

String categoryID = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID));

String placeName =cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_NAME));

String placeAddress = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_ADDRESS));

String placeDescription = cursor.getString(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_DESCRIPTION));

byte[] placeImages = cursor.getBlob(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_IMAGES));

double placeLat = cursor.getDouble(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_LAT));

double placeLng = cursor.getDouble(cursor.getColumnIndexOrThrow(DB\_Utitls.COLUMN\_PLACE\_LNG));

place = new Place.Builder()

.setPlaceID(placeID)

.setCategoryID(categoryID)

.setPlaceName(placeName)

.setPlaceAddress(placeAddress)

.setPlaceDescription(placeDescription)

.setPlaceImages(placeImages)

.setPlaceLat(placeLat)

.setPlaceLng(placeLng)

.buil();

}

if(cursor!= null){

cursor.close();

}

db.close();

return place;

}

//insert palce

public void insert(Place place){

SQLiteDatabase db = placeSqliteHelper.getWritableDatabase();

ContentValues contentValues = new ContentValues();

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_ID,place.getPlaceID());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID,place.getCategoryID());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_NAME,place.getPlaceName());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_ADDRESS,place.getPlaceAddress());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_DESCRIPTION,place.getPlaceDescription());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_IMAGES,place.getPlaceImages());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_LAT,place.getPlaceLat());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_LNG,place.getPlaceLng());

db.insert(DB\_Utitls.PLACE\_TBL\_NAME,null,contentValues);

db.close();

}

//update place

public void update(Place place){

SQLiteDatabase db = placeSqliteHelper.getWritableDatabase();

ContentValues contentValues = new ContentValues();

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_ID,place.getPlaceID());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_CATEGORY\_ID,place.getCategoryID());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_NAME,place.getPlaceName());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_ADDRESS,place.getPlaceAddress());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_DESCRIPTION,place.getPlaceDescription());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_IMAGES,place.getPlaceImages());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_LAT,place.getPlaceLat());

contentValues.put(DB\_Utitls.COLUMN\_PLACE\_LNG,place.getPlaceLng());

String selection = DB\_Utitls.COLUMN\_PLACE\_ID + " = ?";

String[] selectionArgs = {place.getPlaceID()};

db.update(DB\_Utitls.PLACE\_TBL\_NAME,contentValues,selection,selectionArgs);

db.close();

}

//delete Place

public void delete(String placeID){

SQLiteDatabase db = placeSqliteHelper.getWritableDatabase();

String selection = DB\_Utitls.COLUMN\_PLACE\_ID + " = ?";

String[] selectionArgs = {placeID};

db.delete(DB\_Utitls.PLACE\_TBL\_NAME,selection,selectionArgs);

db.close();

}

}

## Set up AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.lemanh\_dev.mylovelyplace">

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@drawable/logo"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<meta-data

android:name="com.google.android.gms.version"

android:value="@integer/google\_play\_services\_version" />

<meta-data

android:name="com.google.android.geo.API\_KEY"

android:value="@string/key\_map\_api" />

<activity android:name=".activity.SplashActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name=".activity.CategoriesActivity" />

<activity android:name=".activity.PlaceActivity" />

<activity android:name=".activity.DetailPlaceActivity" />

<activity android:name=".activity.AddPlaceActivity"></activity>

<activity android:name=".MapActivity"></activity>

</application>

</manifest>

## Buil.gradle

apply plugin: 'com.android.application'

android {

compileSdkVersion 27

defaultConfig {

applicationId "com.example.lemanh\_dev.mylovelyplace"

minSdkVersion 15

targetSdkVersion 27

versionCode 1

versionName "1.0"

testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

dependencies {

implementation fileTree(dir: 'libs', include: ['\*.jar'])

implementation 'com.android.support:appcompat-v7:27.1.1'

implementation 'com.android.support:cardview-v7:27.1.1'

implementation 'com.android.support.constraint:constraint-layout:1.1.0'

implementation 'com.jakewharton:butterknife:8.8.1'

annotationProcessor 'com.jakewharton:butterknife-compiler:8.8.1'

testImplementation 'junit:junit:4.12'

androidTestImplementation 'com.android.support.test:runner:1.0.2'

androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'

implementation 'com.android.support:design:27.1.1'

implementation 'com.google.android.gms:play-services-maps:15.0.1'

}

# Phân chia công việc

|  |  |
| --- | --- |
| Lê Hoàng Mạnh | Dữ liệu, màn hình Splash,Insert,Update,Show Places on Map |
| Nguyễn Công Hậu | Màn hình Category,Detail,Delete Place |