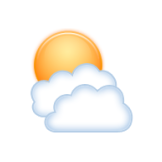
|  |
| --- |
| FPT POLYTECHNIC |
| **DỰ ÁN 1** |
| NGÀNH LẬP TRÌNH MÁY TÍNH (LẬP TRÌNH MOBILE) |



|  |
| --- |
| HÀ NỘI 2020 |

**Đề Tài: Thời Tiết Và Chỉ Số Ô Nhiễm**



**Tên nhóm: Nhóm 8**

**Thành viên: Tô Xuân Trường - Cung Hồng Hải**

**Lớp: PT15151-MOB**

**Giáo viên: Nguyễn Hữu Huy**

MỤC LỤC

[1 Phân tích 2](#_Toc520132211)

[1.1 Hiện trạng 2](#_Toc520132212)

[1.2 Yêu cầu hệ thống 2](#_Toc520132213)

[2 Thiết kế 2](#_Toc520132215)

[2.1 Mô hình triển khai 3](#_Toc520132216)

[2.2 Thiết kế giao diện 4](#_Toc520132220)

[2.2.1 Sơ đồ tổ chức giao diện 4](#_Toc520132221)

[2.2.2 Thiết kế giao diện cho các chức năng nghiệp vụ 4](#_Toc520132222)

[2.3 Công nghệ sử dụng 4](#_Toc520132220)

[3 Thực hiện viết mã 13](#_Toc520132223)

[3.1 Viết mã tạo CSDL 13](#_Toc520132224)

[3.2 Lập trình JDBC 15](#_Toc520132228)

[3.2.1 Lớp hỗ trợ 15](#_Toc520132229)

[3.2.2 Model class - Các lớp mô tả dữ liệu 21](#_Toc520132230)

[3.3 Viết mã cho ứng dụng 23](#_Toc520132232)

[4 Kiểm thử 38](#_Toc520132235)

[5 Đóng gói và triển khai 42](#_Toc520132238)

# 

# Phân tích

## Hiện trạng

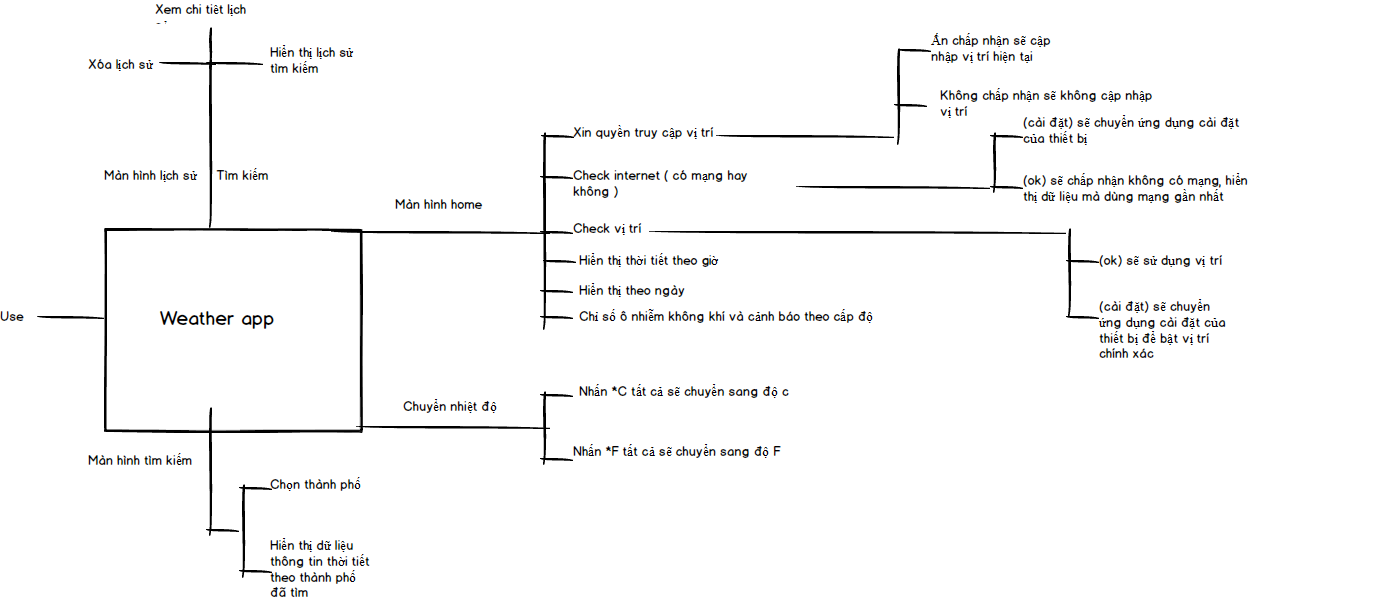
* Hiện nay ô nhiễm không khí hiện đang là mối quan tâm chung của xã hội toàn cầu. Bởi nó được xem là tác nhân hàng đầu gây nên những ảnh hưởng nghiêm trọng đến môi trường và sức khỏe cộng đồng. Cùng với đó là sự biến đổi khí hậu về thời tiết đang diễn ra ngày càng khắc nhiệt
* Chất lượng không khí và thời tiết rất được nhiều người quan tâm, mỗi ngày thức dậy để chuyển bị ngày mới đi làm, ra ngoài và đi chơi
* Do xã hội phát triển, công việc bận rộn, chúng ta không thể hôm nào cũng bỏ ra thời gian để xem dự báo thời tiết vì vậy có 1 ứng dụng về thời tiết là rất cần thiết đối với chúng ta.
* Rất nhiều chất thải khói bụi được thải ra môi trường, chỉ số không khí và mức độ ô nhiễm đang được báo động đỏ ở những thành phố lớn ở Việt Nam • Trên thị trường rất ít ứng dụng tích hợp cả hai chức năng quan trọng và dễ sử dụng với mọi người.

## Yêu cầu hệ thống

* Lấy vị trí hiện tại người dùng
* Hiển thị thời tiết theo ngày và giờ
* Thông báo chỉ số ô nhiễm và chat lượng không khí của giờ hiện tại và vị trí hiện tại của người dùng
* Tìm kiếm thời tiết theo thành phố
* Tắt mạng cũng có thể hiển thị thời tiết và chất lược không khí
* Chuyển từ độ C sang độ F
* Lịch sử tìm kiếm đc lưu trên ZoomDatabase
* Yêu cầu về môi trường công nghệ**:** Ứng dụng cài đặt trên điện thoại, máy tính bảng chạy hệ điều hành Android

# Thiết kế

## Mô hình triển khai

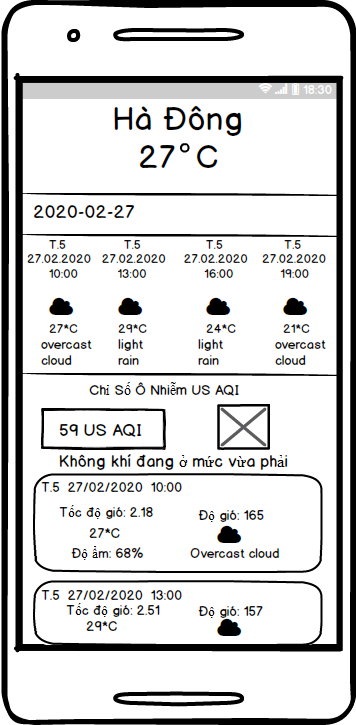
****

## Thiết kế giao diện

### Sơ đồ tổ chức giao diện

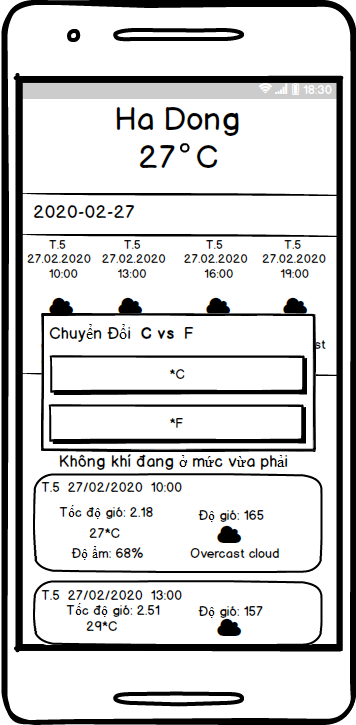
### Thiết kế giao diện cho các chức năng nghiệp vụ

* Màn Hình Chính



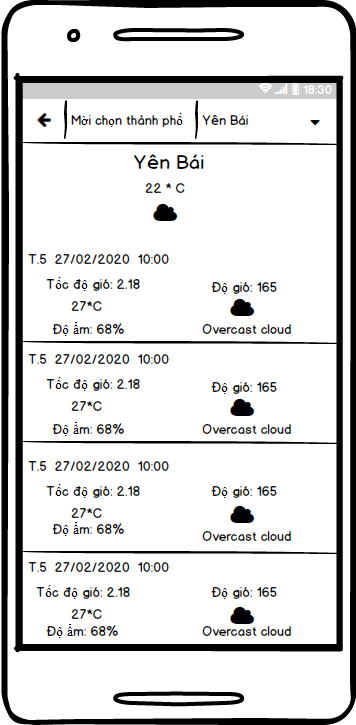
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".presentation.ui.screen.main.MainActivity">  
  
 <androidx.coordinatorlayout.widget.CoordinatorLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true">  
  
 <com.google.android.material.appbar.AppBarLayout  
 android:id="@+id/AppBarLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/height\_AppBarLayout\_"  
 android:background="@drawable/anhdau"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">  
  
 <com.google.android.material.appbar.CollapsingToolbarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/height\_AppBarLayout\_"  
 app:contentScrim="@drawable/anhdau"  
 app:layout\_scrollFlags="scroll|exitUntilCollapsed">  
  
 <TextView  
 android:id="@+id/tv\_temperature"  
 style="@style/tv\_temperature"  
 android:layout\_below="@+id/toolbar"  
 android:scaleType="centerCrop"  
 android:text="@string/nhietdotrong"  
 android:textColor="@color/white"  
 app:layout\_collapseMode="parallax" />  
  
  
 <androidx.appcompat.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/height\_Toolbar\_main"  
 app:contentScrim="@drawable/backgroundweather"  
 app:layout\_collapseMode="pin"  
 app:popupTheme="@style/ThemeOverlay.AppCompat.Light">  
  
 <TextView  
 android:id="@+id/tv\_city"  
 style="@style/textviewcity"  
 android:gravity="center"  
 android:text="@string/nhietdotrong"  
 android:textColor="@color/white" />  
 </androidx.appcompat.widget.Toolbar>  
 </com.google.android.material.appbar.CollapsingToolbarLayout>  
 </com.google.android.material.appbar.AppBarLayout>  
  
 <include layout="@layout/conten\_main" />  
  
 </androidx.coordinatorlayout.widget.CoordinatorLayout>  
  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/bottomnavigation"  
 android:layout\_height="@dimen/height\_BottomNavigationView\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_below="@+id/nes"  
 android:layout\_alignParentBottom="true"  
 app:itemIconTint="@color/white"  
 app:menu="@menu/menu\_main"  
 app:itemIconSize="@dimen/size\_icon\_BottomNavigationView"/>  
</RelativeLayout>

* Màn Hình Chuyển Nhiệt Độ



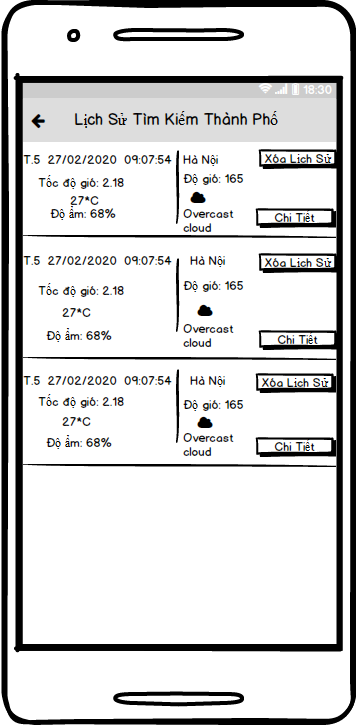
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <Button  
 android:id="@+id/c"  
 style="@style/CF\_dialog"  
 android:background="@color/ground"  
 android:text="@string/doC"  
 android:textColor="@color/white" />  
  
 <Button  
 android:id="@+id/f"  
 style="@style/CF\_dialog"  
 android:background="@color/ground"  
 android:text="@string/doF"  
 android:textColor="@color/white" />  
</LinearLayout>

* Màn Hình Tìm Kiếm Theo Thành Phố



<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/backgroundweather"  
 tools:context=".presentation.ui.screen.searchCity.AboutActivity">  
  
 <RelativeLayout  
 android:id="@+id/rela2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/height\_relativeLayout\_Search">  
  
 <ImageView  
 android:id="@+id/back\_timkiem"  
 style="@style/img\_back\_timkiem"  
 android:layout\_centerVertical="true"  
 android:src="@drawable/ic\_action\_back" />  
  
 <TextView  
 style="@style/gach\_doc2"  
 android:layout\_toRightOf="@+id/back\_timkiem"  
 android:background="@color/white" />  
  
 <TextView  
 android:id="@+id/tv\_choncity"  
 style="@style/tieude\_timkiem"  
 android:layout\_toRightOf="@id/back\_timkiem"  
 android:textColor="@color/white"  
 android:textSize="@dimen/city\_size\_Search" />  
  
 <TextView  
 android:id="@+id/gach\_doc"  
 android:layout\_width="@dimen/vertical\_relativeLayout"  
 android:layout\_height="match\_parent"  
 android:layout\_centerHorizontal="true"  
 android:background="@color/white" />  
  
 <Spinner  
 android:id="@+id/spin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_centerVertical="true"  
 android:layout\_toRightOf="@+id/gach\_doc" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/horizontal\_relativeLayout"  
 android:layout\_alignParentBottom="true"  
 android:background="@color/white" />  
 </RelativeLayout>  
  
 <TextView  
 android:id="@+id/tv\_city\_manCity"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/rela2"  
 android:layout\_centerHorizontal="true"  
 android:text="@string/nhietdotrong"  
 android:textColor="@color/bg\_color"  
 android:textSize="@dimen/text\_City" />  
  
 <TextView  
 android:id="@+id/tv\_temperature\_mancity"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/tv\_city\_manCity"  
 android:layout\_centerHorizontal="true"  
 android:text="@string/nhietdotrong"  
 android:textColor="@color/bg\_color"  
 android:textSize="@dimen/text\_temperature" />  
  
 <ImageView  
 android:id="@+id/imgIconchinh"  
 android:layout\_width="@dimen/width\_icon"  
 android:layout\_height="@dimen/height\_icon"  
 android:layout\_below="@+id/tv\_temperature\_mancity"  
 android:layout\_centerHorizontal="true" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerViewCity"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/imgIconchinh">  
  
 </androidx.recyclerview.widget.RecyclerView>  
</RelativeLayout>

* Màn Hình Lịch Sử Tìm Kiếm



<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/ground"  
 android:orientation="vertical"  
 tools:context=".presentation.ui.screen.history.HistoryActivity">  
  
 <androidx.appcompat.widget.Toolbar  
 android:layout\_height="@dimen/height\_Toolbar\_main"  
 android:layout\_width="match\_parent"  
 android:background="@color/ground\_toolbar\_city">  
  
 <ImageView  
 android:id="@+id/back\_history"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/ic\_action\_back" />  
  
 <TextView  
 style="@style/tieude\_lichsu"  
 android:textColor="@color/white"  
 />  
 </androidx.appcompat.widget.Toolbar>  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerViewHistory"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
</LinearLayout>

## Công nghệ sử dụng

* Githup: Lưu trữ và trao đổi code
* Balsamiq Mockups: Thiết kế phác thảo bố cục giao diện
* Adobe XD: Thiết kế và tạo nguyên mẫu bố cục giao diện
* Android studio: Môi trường code
* Room: lưu trữ dữ liệu
* Retrofit
* SharedPreferences
* Picasso
* Recyclerview
* MVP Models

# Thực hiện viết mã

## Viết mã tạo CSDL

public class History {  
 @PrimaryKey  
 @NonNull  
 public String giohientai**;** @ColumnInfo(name = "thanhpho")  
 public String thanhpho**;** @ColumnInfo(name = "ngayhientai")  
 public String ngayhientai**;** @ColumnInfo(name = "nhietDoTemp")  
 public double nhietDoTemp**;** @ColumnInfo(name = "dogioDeg")  
 public String dogioDeg**;** @ColumnInfo(name = "tocdogioSpeed")  
 public String tocdogioSpeed**;** @ColumnInfo(name = "doamHumidity")  
 public String doamHumidity**;** @ColumnInfo(name = "trangthaiDescription")  
 public String trangthaiDescription**;** @ColumnInfo(name = "iconSql")  
 public String iconSql**;** public String getGiohientai() {  
 return giohientai**;** }  
  
 public void setGiohientai(String giohientai) {  
 this.giohientai = giohientai**;** }  
  
 public String getThanhpho() {  
 return thanhpho**;** }  
  
 public void setThanhpho(String thanhpho) {  
 this.thanhpho = thanhpho**;** }  
  
 public String getNgayhientai() {  
 return ngayhientai**;** }  
  
 public void setNgayhientai(String ngayhientai) {  
 this.ngayhientai = ngayhientai**;** }  
  
 public double getNhietDoTemp() {  
 return nhietDoTemp**;** }  
  
 public void setNhietDoTemp(double nhietDoTemp) {  
 this.nhietDoTemp = nhietDoTemp**;** }  
  
 public String getDogioDeg() {  
 return dogioDeg**;** }  
  
 public void setDogioDeg(String dogioDeg) {  
 this.dogioDeg = dogioDeg**;** }  
  
 public String getTocdogioSpeed() {  
 return tocdogioSpeed**;** }  
  
 public void setTocdogioSpeed(String tocdogioSpeed) {  
 this.tocdogioSpeed = tocdogioSpeed**;** }  
  
 public String getDoamHumidity() {  
 return doamHumidity**;** }  
  
 public void setDoamHumidity(String doamHumidity) {  
 this.doamHumidity = doamHumidity**;** }  
  
 public String getTrangthaiDescription() {  
 return trangthaiDescription**;** }  
  
 public void setTrangthaiDescription(String trangthaiDescription) {  
 this.trangthaiDescription = trangthaiDescription**;** }  
  
 public String getIconSql() {  
 return iconSql**;** }  
  
 public void setIconSql(String iconSql) {  
 this.iconSql = iconSql**;** }  
  
 public double onConverF(double temp) {  
 return temp \* **1.8000** + **32.00;** }  
}

## Lập trình JDBC

### Lớp hỗ trợ

public interface WeatherService {  
 @GET("v2/nearest\_city?key=cd058c32-889e-467e-a840-e640670099d2")  
 Call<Weather> getPeople(@Query("lat") double lat**,** @Query("lon") double lon)**;** @GET("data/2.5/forecast?appid=ecdd51cc4c60e1fb08cf11263bbb546a&units=metric&lang=vi")  
 Call<WeatherList> getWeather(@Query("lat") double lat**,** @Query("lon") double lon)**;** @GET("data/2.5/forecast?appid=ecdd51cc4c60e1fb08cf11263bbb546a&units=metric&lang=vi")  
 Call<WeatherList> getWeatherCity(@Query("q") String q)**;**}

public class APICallManager {  
 public String endpoint = Constants.Path.*DEFAULT\_URL\_API\_PRODUCTION***;** public static APICallManager *instance***;** private static Retrofit *retrofit***;** public PeopleManager peopleManager**;** */\*\*  
 \* singleton class instance  
 \*  
 \** ***@return*** *APICallManager  
 \*/* public static APICallManager getInstance() {  
 if (*instance* == null) {  
 synchronized (APICallManager.class) {  
 if (*instance* == null) {  
 *instance* = new APICallManager()**;** }  
 }  
  
 }  
 return *instance***;** }  
  
  
 public APICallManager() {  
 // enable logging  
 HttpLoggingInterceptor interceptor = new HttpLoggingInterceptor()**;** interceptor.setLevel(HttpLoggingInterceptor.Level.*BODY*)**;** OkHttpClient client = new OkHttpClient.Builder()  
 .addInterceptor(interceptor)  
 .build()**;** *retrofit* = new Retrofit.Builder()  
 .baseUrl(endpoint)  
 .addConverterFactory(GsonConverterFactory.*create*())  
 .client(client)  
 .build()**;** // registering API endpoint manager  
 this.peopleManager = new PeopleManager()**;** }  
  
 public static <**T**> **T** getService(Class<**T**> serviceClass) {  
 return *retrofit*.create(serviceClass)**;** }  
  
 PeopleManager getPeopleManager**;** public class PeopleManager {  
 WeatherService service**;** public PeopleManager() {  
 this.service = *getService*(WeatherService.class)**;** }  
 public Call<Weather> getContacts(GPSTracker gpsTracker) {  
 return service.getPeople(gpsTracker.getLatitude()**,** gpsTracker.getLongtitude())**;** }  
  
  
 }

public class GPSTracker extends Service implements LocationListener {  
 protected LocationManager locationManager**;** Context mcontext**;** private Location location**;** // flag for GPS status  
 boolean isGPSEnabled = false**;** // flag for network status  
 boolean isNetworkEnabled = false**;** // flag for GPS status  
 boolean canGetLocation = false**;** double latitude**;** // latitude  
 double longitude**;** // longitude  
 private static final long *MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES* = **10;** // 10 meters  
 // The minimum time between updates in milliseconds  
 private static final long *MIN\_TIME\_BW\_UPDATES* = **1000** \* **60** \* **1;** // 1 minute  
  
  
 public GPSTracker(Context context) {  
 this.mcontext = context**;** getToaDo()**;** }  
 @SuppressLint("MissingPermission")  
 public Location getToaDo() {  
 try {  
 locationManager = (LocationManager) mcontext  
 .getSystemService(*LOCATION\_SERVICE*)**;** // getting GPS status  
 isGPSEnabled = locationManager  
 .isProviderEnabled(LocationManager.*GPS\_PROVIDER*)**;** // getting network status  
 isNetworkEnabled = locationManager  
 .isProviderEnabled(LocationManager.*NETWORK\_PROVIDER*)**;** if (!isGPSEnabled && !isNetworkEnabled) {  
 // no network provider is enabled  
 } else {  
 this.canGetLocation = true**;** if (isNetworkEnabled) {  
// if (checkSelfPermission(Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && checkSelfPermission(Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  
// // *TODO: Consider calling*// // Activity#requestPermissions  
// // here to request the missing permissions, and then overriding  
// // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
// // int[] grantResults)  
// // to handle the case where the user grants the permission. See the documentation  
// // for Activity#requestPermissions for more details.  
// return null;  
// }  
 locationManager.requestLocationUpdates(  
 LocationManager.*NETWORK\_PROVIDER***,** *MIN\_TIME\_BW\_UPDATES***,** *MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES***,** this)**;** Log.*d*("Network"**,** "Network")**;** if (locationManager != null) {  
 location = locationManager  
 .getLastKnownLocation(LocationManager.*NETWORK\_PROVIDER*)**;** if (location != null) {  
 latitude = location.getLatitude()**;** longitude = location.getLongitude()**;** }  
 }  
 }  
 // if GPS Enabled get lat/long using GPS Services  
 if (isGPSEnabled) {  
 if (location == null) {  
 locationManager.requestLocationUpdates(  
 LocationManager.*GPS\_PROVIDER***,** *MIN\_TIME\_BW\_UPDATES***,** *MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES***,** this)**;** Log.*d*("GPS Enabled"**,** "GPS Enabled")**;** if (locationManager != null) {  
 location = locationManager  
 .getLastKnownLocation(LocationManager.*GPS\_PROVIDER*)**;** if (location != null) {  
 latitude = location.getLatitude()**;** longitude = location.getLongitude()**;** }  
 }  
 }  
 }  
 }  
  
 } catch (Exception e) {  
 e.printStackTrace()**;** }  
 return location**;** }  
 public double getLatitude() {  
 if (location != null) {  
 latitude = location.getLatitude()**;** }  
 // return latitude  
 return latitude**;** }  
 public double getLongtitude() {  
 if (location != null) {  
 longitude = location.getLongitude()**;** }  
 // return Longitude  
 return longitude**;** }  
 @Override  
 public void onLocationChanged(Location location) {  
  
 }  
  
 @Override  
 public void onStatusChanged(String s**,** int i**,** Bundle bundle) {  
  
 }  
  
 @Override  
 public void onProviderEnabled(String s) {  
  
 }  
  
 @Override  
 public void onProviderDisabled(String s) {  
  
 }  
  
 @Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null**;** }  
}

public class WeatherInteractor {  
 APICallListener listener**;** public WeatherInteractor(APICallListener listener) {  
 this.listener = listener**;** }  
  
 public void callAPIGetContacts(GPSTracker gpsTracker) {  
 final Enums.APIRoute route = Enums.APIRoute.*GET\_WEATHER***;** Call<Weather> call = APICallManager.*getInstance*().peopleManager.getContacts(gpsTracker)**;** call.enqueue(new Callback<Weather>() {  
 @Override  
 public void onResponse(Call<Weather> call**,** Response<Weather> response) {  
 listener.onAPICallSucceed(route**,** response.body())**;** }  
 @Override  
 public void onFailure(Call<Weather> call**,** Throwable t) {  
 listener.onAPICallFailed(route**,** t)**;** Log.*e*("Ket Quaaaaaaaa"**,** String.*valueOf*(t))**;** }  
 })**;** }  
  
 public void callAPIlist(GPSTracker gpsTracker) {  
  
 Call<WeatherList> call = APICallManagerList.*getListDay*().peopleManagerList.getContactsListDay(gpsTracker)**;** call.enqueue(new Callback<WeatherList>() {  
 @Override  
 public void onResponse(Call<WeatherList> call**,** Response<WeatherList> response) {  
 listener.onAPICallSucceedList(response.body())**;** Log.*e*("Chay vao Dung"**,** String.*valueOf*(response.body()))**;** }  
 @Override  
 public void onFailure(Call<WeatherList> call**,** Throwable t) {  
 Log.*e*("Ket Qua"**,** String.*valueOf*(t))**;** }  
 })**;** }  
  
 public void callAPICity( String s) {  
 Call<WeatherList> call = APICallManagerCity.*getCity*().peopleManagerCity.getContactsCityDay(s)**;** call.enqueue(new Callback<WeatherList>() {  
 @Override  
 public void onResponse(Call<WeatherList> call**,** Response<WeatherList> response) {  
 listener.onAPICallSucceedCity(response.body())**;** Log.*e*("Chay vao "**,** String.*valueOf*(response.body()))**;** }  
 @Override  
 public void onFailure(Call<WeatherList> call**,** Throwable t) {  
 Log.*e*("Ket Qua"**,** String.*valueOf*(t))**;** }  
 })**;** }

### Model class - Các lớp mô tả dữ liệu

#### Model 1

public class WeatherList {  
  
 @SerializedName("cod")  
 @Expose  
 private String cod**;** @SerializedName("message")  
 @Expose  
 private Integer message**;** @SerializedName("cnt")  
 @Expose  
 private Integer cnt**;** @SerializedName("list")  
 @Expose  
 private List<ListAPI> list = null**;** @SerializedName("city")  
 @Expose  
 private City city**;** public String getCod() {  
 return cod**;** }  
  
 public void setCod(String cod) {  
 this.cod = cod**;** }  
  
 public Integer getMessage() {  
 return message**;** }  
  
 public void setMessage(Integer message) {  
 this.message = message**;** }  
  
 public Integer getCnt() {  
 return cnt**;** }  
  
 public void setCnt(Integer cnt) {  
 this.cnt = cnt**;** }  
  
 public List<ListAPI> getList() {  
 return list**;** }  
  
 public void setList(List<ListAPI> list) {  
 this.list = list**;** }  
  
 public City getCity() {  
 return city**;** }  
  
 public void setCity(City city) {  
 this.city = city**;** }  
  
}

#### Model 2

public class Weather {  
  
 @SerializedName("status")  
 @Expose  
 private String status**;** @SerializedName("data")  
 @Expose  
 private Data data**;** public String getStatus() {  
 return status**;** }  
  
 public void setStatus(String status) {  
 this.status = status**;** }  
  
 public Data getData() {  
 return data**;** }  
  
 public void setData(Data data) {  
 this.data = data**;** }  
  
}

### DAO Class - Các lớp truy xuất dữ liệu

@Dao  
public interface HistoryDao {  
 @Query("SELECT \* FROM history")  
 List<History> getAll()**;** // thêm 1 hoặc nhiều User  
 @Insert  
 long[] insertAll(History ... histories)**;** // xóa 1 User  
 @Delete  
 int delete(History histories)**;**}

## Viết mã cho ứng dụng

public class MainActivity extends BaseActivity implements MainPresenter**,** BottomNavigationView.OnNavigationItemSelectedListener {  
 private BottomNavigationView bottomNavigationView**;** private MainPresenterImpl presenter**;** private TextView tvThanhpho**,** tvNhietdo**,** tvNgay**,** tvUsAQI**,** tvonhiem**,** tvTieudeOnhiem**;** private RecyclerView recyNgay**,** recyList**;** private GPSTracker gpsTracker**;** private WeatherHorizontalAdapter weatherListDayAdapter**;** private ImageView imageView**;** private WeatherDayAdapter weatherListAdapter**;** ///SharedPreferences  
 private List<ListAPI> enums = new ArrayList<>()**;** private SharedPreferences preferences**;** private SharedPreferences.Editor editor**;** private Gson gson**;** private int type\_degree = **0;** private String oC**,** oF**;** private static final String *IS\_DEGREE* = "IS\_DEGREE"**;** private static final String *IS\_KELVIN* = "IS\_KELVIN"**;** @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState)**;** setContentView(R.layout.*activity\_main*)**;** showToastGPS()**;** CheckLocationPermission()**;** CheckInternetshowCaidat()**;** init()**;** Managaer()**;**// gpsTracker = new GPSTracker(getApplicationContext());  
// enums = getValueFromPreference();  
// initRecyclerView(enums);  
 gpsTracker = new GPSTracker(getApplicationContext())**;** presenter = new MainPresenterImpl(this**,** gpsTracker**,** this)**;** enums = getValueFromPreference()**;** initRecyclerView(enums)**;** }  
  
 private void init() {  
 preferences = getSharedPreferences("key"**,** *MODE\_PRIVATE*)**;** editor = preferences.edit()**;** gson = new Gson()**;** initLayout()**;** initData()**;** }  
  
 private void initData() {  
 boolean c = preferences.getBoolean(*IS\_DEGREE***,** true)**;** boolean k = preferences.getBoolean(*IS\_KELVIN***,** false)**;** if (c && !k) {  
 type\_degree = **0;** } else if (!c && k) {  
 type\_degree = **1;** }  
 }  
  
 private void initLayout() {  
 tvThanhpho = findViewById(R.id.*tv\_city*)**;** tvNhietdo = findViewById(R.id.*tv\_temperature*)**;** tvNgay = findViewById(R.id.*tv\_title*)**;** recyNgay = findViewById(R.id.*recyclerView*)**;** tvUsAQI = findViewById(R.id.*tv\_pollution\_AQI*)**;** recyList = findViewById(R.id.*recyclerviewDay*)**;** tvonhiem = findViewById(R.id.*tv\_pollution2*)**;** tvTieudeOnhiem = findViewById(R.id.*tieude*)**;** imageView = findViewById(R.id.*icon\_onhiem*)**;** bottomNavigationView = findViewById(R.id.*bottomnavigation*)**;** bottomNavigationView.setOnNavigationItemSelectedListener(this)**;** }  
  
 @Override  
 public void getRecyclerView(List<ListAPI> weatherListDays) {  
 oC = String.*valueOf*(weatherListDays.get(**0**).getMain().getTemp()).substring(**0, 2**)**;** oF = String.*valueOf*(weatherListDays.get(**0**).getMain().onConvertCelsiusToF(Double.*parseDouble*(oC))).substring(**0, 2**)**;** saveValueToPreference(weatherListDays)**;** weatherListDayAdapter = new WeatherHorizontalAdapter(this**,** weatherListDays**,** type\_degree)**;** recyNgay.setAdapter(weatherListDayAdapter)**;** weatherListAdapter = new WeatherDayAdapter(MainActivity.this**,** weatherListDays**,** type\_degree)**;** recyList.setAdapter(weatherListAdapter)**;** }  
  
 private void initRecyclerView(List<ListAPI> list) {  
 ///hien thi du lieu list khi mat mang  
 weatherListDayAdapter = new WeatherHorizontalAdapter(this**,** list**,** type\_degree)**;** recyNgay.setAdapter(weatherListDayAdapter)**;** weatherListAdapter = new WeatherDayAdapter(MainActivity.this**,** list**,** type\_degree)**;** recyList.setAdapter(weatherListAdapter)**;** String thanhpho = preferences.getString("keyThanhpho"**,** "")**;** tvThanhpho.setText(thanhpho)**;** String ngay = preferences.getString("keyngay"**,** "")**;** tvNgay.setText(ngay)**;** presenter.mainCvsF()**;** }  
  
 private void saveValueToPreference(List<ListAPI> list) {  
 String json = gson.toJson(list)**;** editor.putString("keyList"**,** json)**;** editor.commit()**;** }  
  
 private List<ListAPI> getValueFromPreference() {  
 Type collectionType = new TypeToken<List<ListAPI>>() {  
 }.getType()**;** return gson.fromJson(preferences.getString("keyList"**,** "")**,** collectionType)**;** }  
  
 private void Managaer() {  
 LinearLayoutManager horizontalLayoutManagaer = new LinearLayoutManager(getApplicationContext()**,** LinearLayoutManager.*HORIZONTAL***,** false)**;** recyNgay.setLayoutManager(horizontalLayoutManagaer)**;** LinearLayoutManager LayoutManagaer = new LinearLayoutManager(getApplicationContext())**;** recyList.setLayoutManager(LayoutManagaer)**;** Integer integer = preferences.getInt("keyOnhiem"**, 1**)**;** tvUsAQI.setText(String.*valueOf*(integer))**;** if (integer >= **301**) {  
 tvUsAQI.setBackgroundResource(R.color.*MauNguyHiem*)**;** tvonhiem.setText(R.string.*NguyHiem*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauNguyHiem*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauNguyHiem*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_301*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;**///hien thi do o nhiem khi tat mang  
 } else if (integer >= **201**) {  
 tvUsAQI.setBackgroundResource(R.color.*MauRatONhiem*)**;** tvonhiem.setText(R.string.*RatONhiem*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauRatONhiem*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauRatONhiem*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_201*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** } else if (integer >= **151**) {  
 tvUsAQI.setBackgroundResource(R.color.*MauOnhiem*)**;** tvonhiem.setText(R.string.*Onhiem*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauOnhiem*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauOnhiem*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_151*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** } else if (integer >= **101**) {  
 tvUsAQI.setBackgroundResource(R.color.*MauNhayCam*)**;** tvonhiem.setText(R.string.*NhayCam*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauNhayCam*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauNhayCam*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_101*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** } else if (integer >= **51**) {  
 tvUsAQI.setBackgroundResource(R.color.*MauVuaPhai*)**;** tvonhiem.setText(R.string.*VuaPhai*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauVuaPhai*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauVuaPhai*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_51*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** } else {  
 tvUsAQI.setBackgroundResource(R.color.*MauTot*)**;** tvonhiem.setText(R.string.*Tot*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauTot*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauTot*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_50*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** }  
  
 }  
  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {  
 switch (menuItem.getItemId()) {  
 case R.id.*menu\_bottomn\_Left*:  
 Intent intent = new Intent(this**,** AboutActivity.class)**;** startActivity(intent)**;** return true**;** case R.id.*menu\_bottomn\_Right*:  
 nhietDoF()**;** return true**;** case R.id.*menu\_history*:  
 Intent history = new Intent(this**,** HistoryActivity.class)**;** startActivity(history)**;** return true**;** }  
 return false**;** }  
  
 private void nhietDoF() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this)**;** View view1 = LayoutInflater.*from*(this).inflate(R.layout.*c\_f\_dialog***,** null)**;** builder.setView(view1)**;** builder.setTitle(R.string.*TitleFvsC*)**;** final AlertDialog dialog = builder.show()**;** Button buttonC**,** buttonF**;** buttonC = dialog.findViewById(R.id.*c*)**;** buttonF = dialog.findViewById(R.id.*f*)**;** buttonC.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 type\_degree = **0;** editor.putBoolean(*IS\_DEGREE***,** true)**;** editor.putBoolean(*IS\_KELVIN***,** false)**;** editor.commit()**;** initRecyclerView(enums)**;** dialog.dismiss()**;** }  
 })**;** buttonF.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 type\_degree = **1;** editor.putBoolean(*IS\_DEGREE***,** false)**;** editor.putBoolean(*IS\_KELVIN***,** true)**;** editor.commit()**;** initRecyclerView(enums)**;** dialog.dismiss()**;** }  
 })**;** }  
  
 @Override  
 public void nhietdoC(String C) {  
 Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvNhietdo.setTypeface(typeface)**;** tvNhietdo.setText(C + "ºC")**;** }  
  
 @Override  
 public void nhietdoF(String F) {  
 Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvNhietdo.setTypeface(typeface)**;** tvNhietdo.setText(F + "ºF")**;** }  
  
 @Override  
 public void thanhpho(String s) {  
 Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvThanhpho.setTypeface(typeface)**;** tvThanhpho.setText(s)**;** }  
  
 @Override  
 public void ngay(String ngay) {  
 tvNgay.setText(ngay)**;** }  
  
 @Override  
 public void usAQI(Integer usAQI) {  
// Integer integer = preferences.getInt("keyOnhiem", 0);  
// tvUsAQI.setText(integer + " US AQI");  
 tvUsAQI.setText(String.*valueOf*(usAQI) + " US AQI")**;** }  
  
 @Override  
 public void AQI301() {  
 tvUsAQI.setBackgroundResource(R.color.*MauNguyHiem*)**;** tvonhiem.setText(R.string.*NguyHiem*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauNguyHiem*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauNguyHiem*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_301*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** }  
  
 @Override  
 public void AQI201() {  
 tvUsAQI.setBackgroundResource(R.color.*MauRatONhiem*)**;** tvonhiem.setText(R.string.*RatONhiem*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauRatONhiem*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauRatONhiem*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_201*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** }  
  
 @Override  
 public void AQI151() {  
 tvUsAQI.setBackgroundResource(R.color.*MauOnhiem*)**;** tvonhiem.setText(R.string.*Onhiem*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauOnhiem*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauOnhiem*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_151*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** }  
  
 @Override  
 public void AQI101() {  
 tvUsAQI.setBackgroundResource(R.color.*MauNhayCam*)**;** tvonhiem.setText(R.string.*NhayCam*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauNhayCam*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauNhayCam*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_101*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** }  
  
 @Override  
 public void AQI51() {  
 tvUsAQI.setBackgroundResource(R.color.*MauVuaPhai*)**;** tvonhiem.setText(R.string.*VuaPhai*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauVuaPhai*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauVuaPhai*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_51*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** }  
  
 @Override  
 public void AQI00() {  
 tvUsAQI.setBackgroundResource(R.color.*MauTot*)**;** tvonhiem.setText(R.string.*Tot*)**;** tvonhiem.setTextColor(getResources().getColor(R.color.*MauTot*))**;** tvTieudeOnhiem.setTextColor(getResources().getColor(R.color.*MauTot*))**;** imageView.setImageResource(R.mipmap.*ic\_onhiem\_50*)**;** Typeface typeface = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** Typeface type = Typeface.*createFromAsset*(getAssets()**,** "SpaceMonoBold.ttf")**;** tvonhiem.setTypeface(typeface)**;** tvUsAQI.setTypeface(typeface)**;** tvTieudeOnhiem.setTypeface(type)**;** }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode**,** @NonNull String[] permissions**,** @NonNull int[] grantResults) {  
 switch (requestCode) {  
 case **1**: {  
 if (grantResults.length > **0** && grantResults[**0**] == PackageManager.*PERMISSION\_GRANTED*) {  
 if (ContextCompat.*checkSelfPermission*(this**,** Manifest.permission.*ACCESS\_FINE\_LOCATION*) == PackageManager.*PERMISSION\_GRANTED*) {  
 gpsTracker = new GPSTracker(getApplicationContext())**;** presenter = new MainPresenterImpl(this**,** gpsTracker**,** this)**;** enums = getValueFromPreference()**;** initRecyclerView(enums)**;** Toast.*makeText*(this**,** "Lấy Vị Trí Thành Công "**,** Toast.*LENGTH\_SHORT*).show()**;** }  
 } else {  
 Toast.*makeText*(this**,** "Lấy Vị Trí Thất Bại"**,** Toast.*LENGTH\_SHORT*).show()**;** }  
 return**;** }  
  
 }

### Xử lý màn chính

public class MainPresenterImpl implements APICallListener {  
 // private MainView view;  
 private WeatherInteractor peopleInteractor**;** private MainPresenter main**;** private Integer USaqi**;** private double nhietdo**;** private String ngaygio**;** private List<ListAPI> weatherListDays = new ArrayList<>()**;** private Context context**;** private SharedPreferences sharedPreferences**;** private SharedPreferences.Editor editor**;** private String nhietdof**;** private double doF**,** doC**;** private static final String *IS\_DEGREE* = "IS\_DEGREE"**;** private static final String *IS\_KELVIN* = "IS\_KELVIN"**;** public MainPresenterImpl(MainPresenter main**,** GPSTracker gpsTracker**,** Context context) {  
 this.context = context**;** this.main = main**;** this.peopleInteractor = new WeatherInteractor(this)**;** peopleInteractor.callAPIGetContacts(gpsTracker)**;** peopleInteractor.callAPIlist(gpsTracker)**;** }  
  
 @Override  
 public void onAPICallSucceed(Enums.APIRoute route**,** Weather weather) {  
 USaqi = weather.getData().getCurrent().getPollution().getAqius()**;** sharedPreferences = context.getSharedPreferences("key"**,** context.*MODE\_PRIVATE*)**;** editor = sharedPreferences.edit()**;** editor.putInt("keyOnhiem"**,** USaqi)**;** editor.commit()**;** Integer i = sharedPreferences.getInt("keyOnhiem"**, 1**)**;** if (i >= **301**) {  
 main.AQI301()**;** } else if (i >= **201**) {  
 main.AQI201()**;** } else if (i >= **151**) {  
 main.AQI151()**;** } else if (i >= **101**) {  
 main.AQI101()**;** } else if (i >= **51**) {  
 main.AQI51()**;** } else {  
 main.AQI00()**;** }  
 main.usAQI(i)**;** }  
  
  
 @Override  
 public void onAPICallSucceedList(WeatherList weatherList) {  
 weatherListDays = weatherList.getList()**;** doC = Double.*parseDouble*(String.*valueOf*(weatherListDays.get(**0**).getMain().getTemp()))**;** doF = Double.*parseDouble*(String.*valueOf*(weatherListDays.get(**0**).getMain().onConvertCelsiusToF(doC)))**;** main.getRecyclerView(weatherListDays)**;** String thanhpho = weatherList.getCity().getName()**;** nhietdo = weatherList.getList().get(**0**).getMain().getTemp()**;** ngaygio = weatherList.getList().get(**0**).getDtTxt()**;**///luu 1 string vao sharedPreferences  
 SharedPreferences sharedPreferences = context.getSharedPreferences("key"**,** context.*MODE\_PRIVATE*)**;** SharedPreferences.Editor editor = sharedPreferences.edit()**;** editor.putString("keyThanhpho"**,** thanhpho)**;** editor.putString("keynhietdo"**,** String.*valueOf*(nhietdo).substring(**0, 2**))**;** editor.putString("keyngay"**,** ngaygio)**;** editor.putString("keyC"**,** String.*valueOf*(doC).substring(**0, 2**))**;** editor.putString("keyF"**,** String.*valueOf*(doF).substring(**0, 2**))**;** editor.apply()**;** main.ngay(ngaygio)**;** main.thanhpho(thanhpho)**;** }  
  
 public void mainCvsF() {  
 SharedPreferences sharedPreferences = context.getSharedPreferences("key"**,** context.*MODE\_PRIVATE*)**;** boolean s = sharedPreferences.getBoolean(*IS\_DEGREE***,** true)**;** boolean k = sharedPreferences.getBoolean(*IS\_KELVIN***,** false)**;** if (s && !k) {  
 String keyC = sharedPreferences.getString("keyC"**,** "")**;** main.nhietdoC(keyC)**;** } else if (!s && k) {  
 String keyF = sharedPreferences.getString("keyF"**,** "")**;** main.nhietdoF(keyF)**;** }  
 }  
  
  
 @Override  
 public void onAPICallSucceedCity(WeatherList weatherCity) {  
  
 }  
  
 @Override  
 public void onAPICallFailed(Enums.APIRoute route**,** Throwable throwable) {  
// onError(throwable.getMessage());  
 }

### Xử lý Form màn tìm kiếm

public class AboutPresenterImpl implements APICallListener {  
 WeatherInteractor weatherInteractor**;** AboutPresenter mView**;** Context mcontext**;** List<ListAPI> listCityList = new ArrayList<>()**;** private String gio**,** ngay**,** dogio**,** tocdogio**,** doam**,** trangthai**,** icon**;** private double nhietdo**,** nhietdoF**;** private double s**;** private double s1**;** // RoomDatacbase  
 private AppDatabase appDatabase**;** private History historyModel = new History()**;** private HistoryQueryTask historyQueryTask**;** private static final String *IS\_DEGREE* = "IS\_DEGREE"**;** private static final String *IS\_KELVIN* = "IS\_KELVIN"**;** public AboutPresenterImpl(AboutPresenter mView**,** Context context**,** HistoryQueryTask QueryTask) {  
 this.mView = mView**;** this.mcontext = context**;** this.historyQueryTask=QueryTask**;** this.weatherInteractor = new WeatherInteractor(this)**;** }  
  
 @Override  
 public void onAPICallSucceed(Enums.APIRoute route**,** Weather weather) {  
 }  
  
 @Override  
 public void onAPICallSucceedList(WeatherList weatherList) {  
  
 }  
  
 @Override  
 public void onAPICallSucceedCity(WeatherList weatherCity) {  
 listCityList = weatherCity.getList()**;** mView.getRecyCity(listCityList)**;** String thanhpho = weatherCity.getCity().getName()**;** String nhietdo = String.*valueOf*(weatherCity.getList().get(**0**).getMain().getTemp()).substring(**0, 2**)**;** String iconchinh = weatherCity.getList().get(**0**).getWeather().get(**0**).getIcon()**;** s = listCityList.get(**0**).getMain().getTemp()**;** s1 = listCityList.get(**0**).getMain().onConvertCelsiusToF(s)**;** SharedPreferences sharedPreferences = mcontext.getSharedPreferences("key"**,** mcontext.*MODE\_PRIVATE*)**;** boolean c = sharedPreferences.getBoolean(*IS\_DEGREE***,** true)**;** boolean k = sharedPreferences.getBoolean(*IS\_KELVIN***,** false)**;** if (c && !k) {  
 mView.nhietdoC(s)**;** } else if (!c && k) {  
 mView.nhietdoF(s1)**;** }  
 mView.icon(iconchinh)**;** mView.thanhpho(thanhpho)**;** }  
  
 @Override  
 public void onAPICallFailed(Enums.APIRoute route**,** Throwable throwable) {  
  
 }  
  
 public void Tim(String snhap) {  
 weatherInteractor.callAPICity(snhap)**;** historyModel.setThanhpho(snhap)**;** }  
  
 public void insetLichSu(List<ListAPI> listCityList) {  
 appDatabase = Room.*databaseBuilder*(mcontext**,** AppDatabase.class**,** "HistoryRooom.db").allowMainThreadQueries().build()**;** nhietdo = listCityList.get(**0**).getMain().getTemp()**;** nhietdoF = listCityList.get(**0**).getMain().onConvertCelsiusToF(nhietdo)**;** dogio = String.*valueOf*(listCityList.get(**0**).getWind().getDeg())**;** tocdogio = String.*valueOf*(listCityList.get(**0**).getWind().getSpeed())**;** doam = String.*valueOf*(listCityList.get(**0**).getMain().getHumidity())**;** trangthai = listCityList.get(**0**).getWeather().get(**0**).getDescription()**;** icon = listCityList.get(**0**).getWeather().get(**0**).getIcon()**;** Calendar calendar = Calendar.*getInstance*()**;** gio = DateFormat.*getTimeInstance*(DateFormat.*MEDIUM*).format(calendar.getTime())**;** ngay = DateFormat.*getDateInstance*(DateFormat.*SHORT*).format(calendar.getTime())**;** historyModel.setGiohientai(gio)**;** historyModel.setNgayhientai(ngay)**;** historyModel.setNhietDoTemp(nhietdo)**;** historyModel.setDogioDeg(dogio)**;** historyModel.setTocdogioSpeed(tocdogio)**;** historyModel.setDoamHumidity(doam)**;** historyModel.setTrangthaiDescription(trangthai)**;** historyModel.setIconSql(icon)**;** long[] kiemtra = appDatabase.historyDao().insertAll(historyModel)**;** if (kiemtra[**0**] > **0**) {  
 Toast.*makeText*(mcontext**,** R.string.*ThemLichSu***,** Toast.*LENGTH\_LONG*).show()**;** } else {  
 Toast.*makeText*(mcontext**,** R.string.*ThemLichSuThatBai***,** Toast.*LENGTH\_LONG*).show()**;** }  
  
 historyQueryTask.insertHistory(new HistoryQueryTask.OnQuery<long[]>() {  
 @Override  
 public void onResult(long[] longs) {  
 }  
 }**,**historyModel)**;**

# Kiểm thử

| **Poject Name: Weather** | **Test Case** |
| --- | --- |
| Test Case ID: WE\_001 | Test Designed by: Tô Xuân Trường |
| Test Priority (Low/Medium/High): Med | Test Designed date: 23/04/2020 |
| Module Name: màn hinh chính | Test Executed by: Tô Xuân Trường |
| Test Title: màn home | Test Execution date: 23/04/2020 |
| Description: hiển thị dữ liệu thời tiết trên màn hình |  |

| **Pre-conditions: Hiển thị vị chí người dung và mọi dữ liệu thời tiết và mức độ ô nhiễm** |
| --- |
| Dependencies: |

| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Chuyển đến màn home |  | Hiển thị được dữ liệu theo tọa độ | Hiển thị được dữ liệu theo tọa độ | Pass |  |
| 2 | Kiểm thử tọa độ người dùng | Lat: 20.918805  Long:105.924309 | Hiển thị được dữ liệu theo đúng tọa độ | Hiển thị được dữ liệu theo đúng tọa độ | Pass |  |

| **Poject Name: Weather** | **Test Case** |
| --- | --- |
| Test Case ID: WE\_002 | Test Designed by: Tô Xuân Trường |
| Test Priority (Low/Medium/High): Med | Test Designed date: 23/04/2020 |
| Module Name: chức năng chuyển từ đọ C sang độ F | Test Executed by: Tô Xuân Trường |
| Test Title: chuyển đọ C sang F | Test Execution date: 23/04/2020 |
| Description: Chuyển nhiệt độ từ đọ C sang F Và Fsang C |  |

| **Pre-conditions: khi nhấn F Tất cả nhiệt độ của ứng dụng sẽ chuể về đọ C và ngược lại** |
| --- |
| Dependencies: |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Chuyển đến chức năng F vs C |  | Hiển thị được chức năng | Hiển thị được chức năng | Pass |  |
| 2 | Click button F |  | Tất cả đồng bộ nhiệt độ là F | Tất cả đồng bộ chuyển nhiệt độ là F | Pass |  |
| 3 | Click button C |  | Tất cả đồng bộ nhiệt độ là C | Tất cả đồng bộ chuyển nhiệt độ là F | Pass |  |

| **Poject Name: Weather** | **Test Case** |
| --- | --- |
| Test Case ID: WE\_003 | Test Designed by: Tô Xuân Trường |
| Test Priority (Low/Medium/High): Med | Test Designed date: 23/04/2020 |
| Module Name: Màn tìm tiếm theo thành phố | Test Executed by: Tô Xuân Trường |
| Test Title: Màn tìm Kiếm | Test Execution date: 23/04/2020 |
| Description: Chọn thành phố và hiển thị nhiệt độ và dữ liệu thời tiết |  |

| **Pre-conditions: Chọn thành phố và hiển thị nhiệt độ và dữ liệu thời tiết** |
| --- |
| Dependencies: |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Chuyển đến màn Tìm Kiếm |  | Hiển thị được dữ liệu theo thành phố | Hiển thị được dữ liệu theo thành phố | Pass |  |
| 2 | Click chọn thành phố |  | Hiển thị đsung dữ liệu của thành phố và thêm vào lịch sử tìm kiếm | Hiển thị đsung dữ liệu của thành phố và thêm vào lịch sử tìm kiếm | Pass |  |

| **Poject Name: Weather** | **Test Case** |
| --- | --- |
| Test Case ID: WE\_004 | Test Designed by: Tô Xuân Trường |
| Test Priority (Low/Medium/High): Med | Test Designed date: 23/04/2020 |
| Module Name: Màn lịch tìm kiếm | Test Executed by: Tô Xuân Trường |
| Test Title: Màn Lich Sử tìm Kiếm | Test Execution date: 23/04/2020 |
| Description: Hiển Thị lich sử tìm kiếm và xóa lịch sử, xem chi tiết |  |

| **Pre-conditions: Hiển Thị lijch sử tìm kiếm xóa lịch sử và xem chi tiết** |
| --- |
| Dependencies: |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Chuyển đến màn lịch sử Tìm Kiếm |  | Hiển thị được danh sách thành phố đã tìm kiếm | Hiển thị được danh sách thành phố đã tìm kiếm | Pass |  |
| 2 | Xóa Lịch sử tìm kiếm |  | Xóa dữ liệu lịch sử hiển thị trên màn hifnhvaf trong sql | Xóa dữ liệu lịch sử hiển thị trên màn hifnhvaf trong sql | Pass |  |
| 3 | Chi tiết tìm kiếm |  | Hiển thị được chi tiết dữ liệu đã tìm kiếm trong thành phố | Hiển thị được chi tiết dữ liệu đã tìm kiếm trong thành phố |  |  |

# Đóng gói và triển khai