**LAB 4 BUỔI 5**

Họ tên: Nguyễn Hưng Văn Khánh

Lớp: 11\_THMT

MSSV: 1150080021

**AndroidManifest.xml**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.broadcast">  
  
 <!-- Quyền -->  
 <uses-permission android:name="android.permission.RECEIVE\_SMS" />  
 <uses-permission android:name="android.permission.SEND\_SMS" />  
 <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
 <uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
  
 <!-- Khai báo phần cứng (để tránh warning khi không có SIM) -->  
 <uses-feature android:name="android.hardware.telephony" android:required="false" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.App">  
  
 <!-- Activity chính -->  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN"/>  
 <category android:name="android.intent.category.LAUNCHER"/>  
 </intent-filter>  
 </activity>  
  
 <!-- Receiver nhận SMS -->  
 <receiver  
 android:name=".SmsReceiver"  
 android:exported="true">  
 <intent-filter android:priority="1000">  
 <action android:name="android.provider.Telephony.SMS\_RECEIVED"/>  
 </intent-filter>  
 </receiver>  
  
 <!-- Receiver cắm/rút sạc -->  
 <receiver  
 android:name=".PowerStateChangeReceiver"  
 android:exported="false">  
 <intent-filter>  
 <action android:name="android.intent.action.ACTION\_POWER\_CONNECTED"/>  
 <action android:name="android.intent.action.ACTION\_POWER\_DISCONNECTED"/>  
 </intent-filter>  
 </receiver>  
  
 </application>  
</manifest>

**MainActivity.java**

package com.example.broadcast;  
  
import android.Manifest;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.content.SharedPreferences;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.telephony.SmsManager;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.CompoundButton;  
import android.widget.LinearLayout;  
import android.widget.ListView;  
import android.widget.Switch;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.localbroadcastmanager.content.LocalBroadcastManager;  
  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity {  
 public static boolean *isRunning* = false;  
 public static final String *AUTO\_RESPONSE* = "auto\_response";  
 private static final int *REQ\_PERMS* = 101;  
  
 private Switch swAutoResponse;  
 private Button btnSafe, btnMayday;  
 private ListView lvMessages;  
 private LinearLayout llButtons;  
 private ArrayList<String> requesters;  
 private ArrayAdapter<String> adapter;  
 private BroadcastReceiver broadcastReceiver;  
 private SharedPreferences prefs;  
 private LocalBroadcastManager localBroadcastManager;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 swAutoResponse = findViewById(R.id.*sw\_auto\_response*);  
 btnSafe = findViewById(R.id.*btn\_safe*);  
 btnMayday = findViewById(R.id.*btn\_mayday*);  
 lvMessages = findViewById(R.id.*lv\_messages*);  
 llButtons = findViewById(R.id.*ll\_buttons*);  
  
 prefs = getSharedPreferences("broadcast\_prefs", *MODE\_PRIVATE*);  
 boolean auto = prefs.getBoolean(*AUTO\_RESPONSE*, false);  
 swAutoResponse.setChecked(auto);  
 if (auto) llButtons.setVisibility(android.view.View.*GONE*);  
  
 requesters = new ArrayList<>();  
 adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*, requesters);  
 lvMessages.setAdapter(adapter);  
  
 // Request permissions  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*RECEIVE\_SMS*) != PackageManager.*PERMISSION\_GRANTED* || ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*SEND\_SMS*) != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(  
 this,  
 new String[]{Manifest.permission.*RECEIVE\_SMS*, Manifest.permission.*SEND\_SMS*},  
 *REQ\_PERMS* );  
 }  
  
 handleClicks();  
 initBroadcastReceiver();  
  
 // init LocalBroadcastManager  
 localBroadcastManager = LocalBroadcastManager.*getInstance*(this);  
  
 // Nếu Activity được start bởi SmsReceiver (khi app k chạy), kiểm tra intent extras  
 Intent i = getIntent();  
 if (i != null && i.hasExtra(SmsReceiver.*SMS\_MESSAGE\_ADDRESS\_KEY*)) {  
 ArrayList<String> addrs = i.getStringArrayListExtra(SmsReceiver.*SMS\_MESSAGE\_ADDRESS\_KEY*);  
 if (addrs != null) processReceiveAddresses(addrs);  
 }  
 }  
  
 private void initBroadcastReceiver() {  
 if (broadcastReceiver != null) return;  
 broadcastReceiver = new BroadcastReceiver() {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 ArrayList<String> addrs = intent.getStringArrayListExtra(SmsReceiver.*SMS\_MESSAGE\_ADDRESS\_KEY*);  
 if (addrs != null) processReceiveAddresses(addrs);  
 }  
 };  
 }  
  
 private void processReceiveAddresses(ArrayList<String> addresses) {  
 for (String a : addresses) if (!requesters.contains(a)) requesters.add(a);  
 adapter.notifyDataSetChanged();  
  
 if (swAutoResponse.isChecked()) {  
 replyAll(true); // mặc định auto reply "I am fine..."  
 }  
 }  
  
 private void handleClicks() {  
 btnSafe.setOnClickListener(v -> respond(true));  
 btnMayday.setOnClickListener(v -> respond(false));  
 swAutoResponse.setOnCheckedChangeListener((CompoundButton b, boolean checked) -> {  
 if (checked) llButtons.setVisibility(android.view.View.*GONE*);  
 else llButtons.setVisibility(android.view.View.*VISIBLE*);  
 prefs.edit().putBoolean(*AUTO\_RESPONSE*, checked).apply();  
 });  
  
 lvMessages.setOnItemClickListener((parent, view, position, id) -> {  
 // click 1 item -> trả lời 1 số  
 String number = requesters.get(position);  
 replyOne(number, true);  
 });  
 }  
  
 private void respond(boolean safe) {  
 replyAll(safe);  
 }  
  
 private void replyOne(String number, boolean safe) {  
 String msg = safe ? "I am safe and well. Worry not!" : "Tell my mother I love her";  
 sendSms(number, msg);  
 requesters.remove(number);  
 adapter.notifyDataSetChanged();  
 }  
  
 private void replyAll(boolean safe) {  
 String msg = safe ? "I am safe and well. Worry not!" : "Tell my mother I love her";  
 ArrayList<String> copied = new ArrayList<>(requesters);  
 for (String n : copied) {  
 sendSms(n, msg);  
 requesters.remove(n);  
 }  
 adapter.notifyDataSetChanged();  
 }  
  
 private void sendSms(String number, String message) {  
 try {  
 SmsManager sms = SmsManager.*getDefault*();  
 sms.sendTextMessage(number, null, message, null, null);  
 Toast.*makeText*(this, "Đã gửi tới " + number, Toast.*LENGTH\_SHORT*).show();  
 } catch (Exception e) {  
 Toast.*makeText*(this, "Gửi thất bại: " + e.getMessage(), Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 *isRunning* = true;  
 if (broadcastReceiver == null) initBroadcastReceiver();  
  
 // Register using LocalBroadcastManager (no exported flag needed)  
 localBroadcastManager.registerReceiver(broadcastReceiver,  
 new IntentFilter(SmsReceiver.*SMS\_FORWARD\_BROADCAST\_RECEIVER*));  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 *isRunning* = false;  
 try {  
 localBroadcastManager.unregisterReceiver(broadcastReceiver);  
 } catch (Exception ignored) {}  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,  
 @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 // xử lý khi người dùng cấp/không cấp quyền  
 }  
}

**PowerStateChangeReceiver.java**

package com.example.broadcast;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.widget.Toast;  
  
public class PowerStateChangeReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
 if (Intent.*ACTION\_POWER\_CONNECTED*.equals(action)) {  
 Toast.*makeText*(context, "Power connected", Toast.*LENGTH\_SHORT*).show();  
 } else if (Intent.*ACTION\_POWER\_DISCONNECTED*.equals(action)) {  
 Toast.*makeText*(context, "Power disconnected", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

**SmsReceiver.java**

package com.example.broadcast;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.provider.Telephony;  
import android.telephony.SmsMessage;  
  
import androidx.localbroadcastmanager.content.LocalBroadcastManager;  
  
import java.util.ArrayList;  
  
public class SmsReceiver extends BroadcastReceiver {  
 public static final String *SMS\_FORWARD\_BROADCAST\_RECEIVER* = "com.example.broadcast.SMS\_FORWARD";  
 public static final String *SMS\_MESSAGE\_ADDRESS\_KEY* = "addresses";  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 SmsMessage[] msgs = Telephony.Sms.Intents.*getMessagesFromIntent*(intent);  
 ArrayList<String> addresses = new ArrayList<>();  
 for (SmsMessage m : msgs) {  
 String body = m.getMessageBody();  
 String from = m.getOriginatingAddress();  
 if (body != null && body.toLowerCase().contains("are you ok")) {  
 if (!addresses.contains(from)) addresses.add(from);  
 }  
 }  
  
 if (addresses.size() == 0) return;  
  
 if (MainActivity.*isRunning*) {  
 Intent iForward = new Intent(*SMS\_FORWARD\_BROADCAST\_RECEIVER*);  
 iForward.putStringArrayListExtra(*SMS\_MESSAGE\_ADDRESS\_KEY*, addresses);  
 // send internal broadcast  
 LocalBroadcastManager.*getInstance*(context).sendBroadcast(iForward);  
 } else {  
 Intent iMain = new Intent(context, MainActivity.class);  
 iMain.setFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 iMain.putStringArrayListExtra(*SMS\_MESSAGE\_ADDRESS\_KEY*, addresses);  
 context.startActivity(iMain);  
 }  
 }  
}

**activity\_main**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@android:color/white">  
  
 <!-- App bar (Material) -->  
 <com.google.android.material.appbar.MaterialToolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 app:title="@string/app\_name"  
 app:titleTextColor="@android:color/white"  
 android:background="@color/purple\_500" />  
  
 <!-- Header text -->  
 <TextView  
 android:id="@+id/tv\_request\_list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Bạn cảm thấy ok chứ"  
 android:paddingStart="16dp"  
 android:paddingEnd="16dp"  
 android:paddingTop="12dp"  
 android:textSize="14sp"  
 android:textColor="#444" />  
  
 <!-- Main list -->  
 <ListView  
 android:id="@+id/lv\_messages"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:divider="@android:color/transparent"  
 android:paddingStart="8dp"  
 android:paddingEnd="8dp" />  
  
 <!-- Buttons block (2 nút stacked) -->  
 <LinearLayout  
 android:id="@+id/ll\_buttons"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <Button  
 android:id="@+id/btn\_safe"  
 style="@style/Widget.App.RoundedButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="48dp"  
 android:text="I AM SAFE AND WELL"  
 android:textAllCaps="true"/>  
  
 <View android:layout\_width="match\_parent" android:layout\_height="8dp"/>  
  
 <Button  
 android:id="@+id/btn\_mayday"  
 style="@style/Widget.App.RoundedButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="48dp"  
 android:text="MAYDAY!"  
 android:textAllCaps="true"/>  
 </LinearLayout>  
  
 <!-- Auto Response row -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="56dp"  
 android:orientation="horizontal"  
 android:gravity="center\_vertical"  
 android:paddingStart="12dp"  
 android:paddingEnd="12dp">  
  
 <TextView  
 android:layout\_width="0dp"  
 android:layout\_weight="1"  
 android:layout\_height="wrap\_content"  
 android:text="Auto Response"  
 android:textSize="14sp"  
 android:textColor="#777"/>  
  
 <Switch  
 android:id="@+id/sw\_auto\_response"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"/>  
 </LinearLayout>  
  
</LinearLayout>

**colors.xml**

<resources>  
 <color name="purple\_200">#BB86FC</color>  
 <color name="purple\_500">#6200EE</color>  
 <color name="purple\_700">#3700B3</color>  
 <color name="teal\_200">#03DAC5</color>  
 <color name="white">#FFFFFF</color>  
</resources>

themes.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources xmlns:tools="http://schemas.android.com/tools">  
 <style name="Theme.App" parent="Theme.MaterialComponents.DayNight.NoActionBar">  
  
 <item name="colorPrimary">@color/purple\_500</item>  
 <item name="colorPrimaryVariant">@color/purple\_700</item>  
 <item name="colorOnPrimary">@android:color/white</item>  
 <item name="android:statusBarColor">@color/purple\_700</item>  
 </style>  
</resources>

A black rectangular device with a white screen

AI-generated content may be incorrect.



