

I. Giới thiệu Broadcast Receiver

Sử dụng Broadcast Intent để gửi thông điệp giữa các ứng dụng. Broadcast Receiver được thiết kế để nhận các broadcast intent. Ví dụ làm sao để biết có tin nhắn đến, hoặc có cuộc gọi đến, đang sạc pin hay đã rút sạc...

➤ Cách thức gửi Broadcast Intent

- Tạo một intent

Intent intent = new Intent("com.hieu.copyToApp");

- Gọi hàm `sendBroadcast()` của Activity

sendBroadcast(intent);

- Có hai dạng broadcast được nhận

- Normal broadcast được gửi bởi `sendBroadcast()`
- Ordered broadcast được gửi bởi `sendOrderedBroadcast()`

➤ Có 2 cách đăng ký Broadcast Receiver:

- **Đăng ký trong coding:** Lắng nghe mọi thứ trong Intent- filter, nếu tắt ứng dụng sẽ không lắng nghe nên ít được dùng.
- **Đăng ký trong Manifest:** Nó trở thành dịch vụ, tự động lắng nghe mọi thứ trong Intent – filter kể cả khi đã đóng ứng dụng. Thường được sử dụng.

Broadcast Receiver mới được kế thừa từ class *BroadcastReceiver* và override hàm `onReceive()`. Mẫu hiện thực Broadcast Receiver

```
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class MyBroadcastReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        //TODO: React to the Intent received.
    }
}
```

II. Thực hành

1. Bài 1: Đăng ký BroadcastReceiver bằng code. Tạo ứng dụng nhận tin nhắn

a) Mô tả ứng dụng

- Ứng dụng sử dụng BroadcastReceiver để lắng nghe sự kiện có tin nhắn đến trong điện thoại. Khi có tin nhắn mới, ứng dụng sẽ hiển thị message thông báo có tin nhắn mới dưới dạng Toast. SĐT gửi và nội dung tin nhắn được hiển thị lên textview.
- Vì sử dụng BroadcastReceiver được đăng ký bằng code khi bị tắt (Stop/Destroy), ứng dụng sẽ không thể lắng nghe tin nhắn đến.

b) Các bước thực hiện

- Xây dựng các file dimens.xml

Name	Value
margin_base	5dp
margin_basex2	10dp
text_medium	16sp

- Xây dựng file strings.xml

Name	Value
you_have_a_new_message	Hey! You have a new message

- Cấp quyền trong Manifest

```
<uses-permission  
    android:name="android.permission.RECEIVE_SMS"  
>
```

- Xây dựng giao diện

```
<LinearLayout 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:orientation="vertical"
        android:padding="@dimen/margin_base"
        tools:context=".MainActivity" >

        <TextView
            android:id="@+id/tv_content"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="@dimen/text_medium"
            />
    </LinearLayout>

```

- Nội dung code

Mô tả	Nội dung
Khai báo biến	<pre> private BroadcastReceiver broadcastReceiver; private IntentFilter filter; </pre>
Xây dựng hàm xử lý khi tin nhắn đến	<pre> public void processReceive(Context context, Intent intent) { Toast.makeText(context, getString(R.string.you_have_a_new_message), Toast.LENGTH_LONG).show(); TextView tvContent = (TextView) findViewById(R.id.tv_content); // Use "pdus" as key to get message final String SMS_EXTRA = "pdus"; Bundle bundle = intent.getExtras(); // Get array of messages which were received at the same time Object[] messages = (Object[]) bundle.get(SMS_EXTRA); String sms = ""; SmsMessage smsMsg; for (int i = 0; i < messages.length; i++) { if (android.os.Build.VERSION.SDK_INT >= 23) { smsMsg = SmsMessage.createFromPdu((byte[]) messages[i], ""); } else { smsMsg = SmsMessage.createFromPdu((byte[]) messages[i]); } // Get message body String msgBody = smsMsg.getMessageBody(); // Get source address of message String address = smsMsg.getDisplayOriginatingAddress(); sms += address + ":\n" + msgBody + "\n"; } // Show messages in textview tvContent.setText(sms); } </pre>

Xây dựng hàm khởi tạo broadcastReceiver	<pre>private void initBroadcastReceiver() { // Create filter to listen to incoming messages filter = new IntentFilter ("android.provider.Telephony.SMS_RECEIVED"); // Create broadcastReceiver broadcastReceiver = new BroadcastReceiver() { // Process when a message comes public void onReceive(Context context, Intent intent) { processReceive(context, intent); } }; }</pre>
Tự động đăng ký và hủy đăng ký BroadcastReceiver khi Resume, Stop Activity	<pre>@Override protected void onResume() { super.onResume(); // Make sure broadcastReceiver was created if (broadcastReceiver == null) initBroadcastReceiver(); // Register Receiver registerReceiver(broadcastReceiver, filter); } @Override protected void onStop() { super.onStop(); // UnregisterReceiver when app is destroyed unregisterReceiver(broadcastReceiver); }</pre>
Khởi tạo BroadcastReceiver trong onCreate() của Activity	<pre>protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); initBroadcastReceiver(); }</pre>

Lưu ý: Khi cài đặt xong lần đầu tiên, nhớ vào setting trên máy ảo, chọn quản lý app, mở app vừa cài đặt vào, bật permission cho việc nhận tin nhắn (nó sẽ yêu cầu đối với những API sau này)

2. Bài 2: Sử dụng BroadcastReceiver đăng ký trong AndroidManifest để tạo ứng dụng nhận biết trạng thái sạc pin/ không sạc pin.

a) Mô tả ứng dụng

- Ứng dụng hiển thị thông báo dưới dạng Toast khi điện thoại được cắm sạc hoặc rút sạc.
- Ứng dụng sử dụng BroadcastReceiver được đăng ký trong AndroidManifest nên nó vẫn có thể lắng nghe và hiển thị được thông báo ngay cả khi ứng dụng bị tắt (Stop/Destroy).

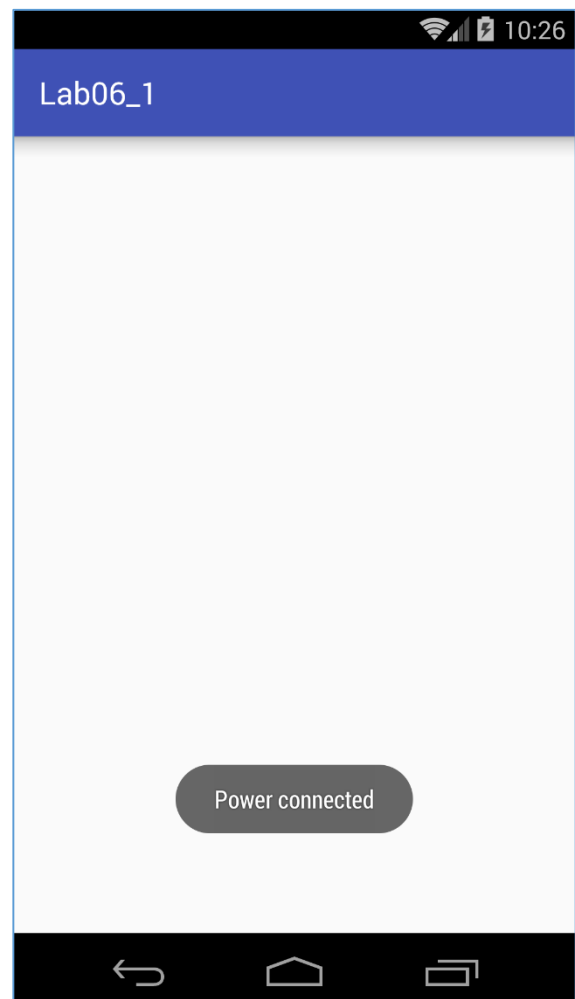
b) Các bước thực hiện

- Xây dựng file strings.xml

Name	Value
<code>power_connected</code>	Power connected
<code>power_disconnected</code>	Power disconnected

- Xây dựng class

PowerStateChangeReceiver extend từ BroadcastReceiver xử lý các sự kiện cắm/ngắt sạc:



```
public class PowerStateChangeReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        if (context == null) return;  
        if (intent.getAction().equals(Intent.ACTION_POWER_CONNECTED)) {  
            Toast.makeText(context, "Power connected", Toast.LENGTH_LONG).show();  
        }  
  
        if (intent.getAction().endsWith(Intent.ACTION_POWER_DISCONNECTED)) {  
            Toast.makeText(context, "Power disconnected", Toast.LENGTH_LONG).show();  
        }  
    }  
}
```

- Đăng ký PowerStateChangeReceiver trong AndroidManifest

```
<receiver android:name=".PowerStateChangeReceiver">  
    <intent-filter>  
        <action android:name="android.intent.action.ACTION_POWER_CONNECTED" />  
        <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED" />  
    </intent-filter>  
</receiver>
```

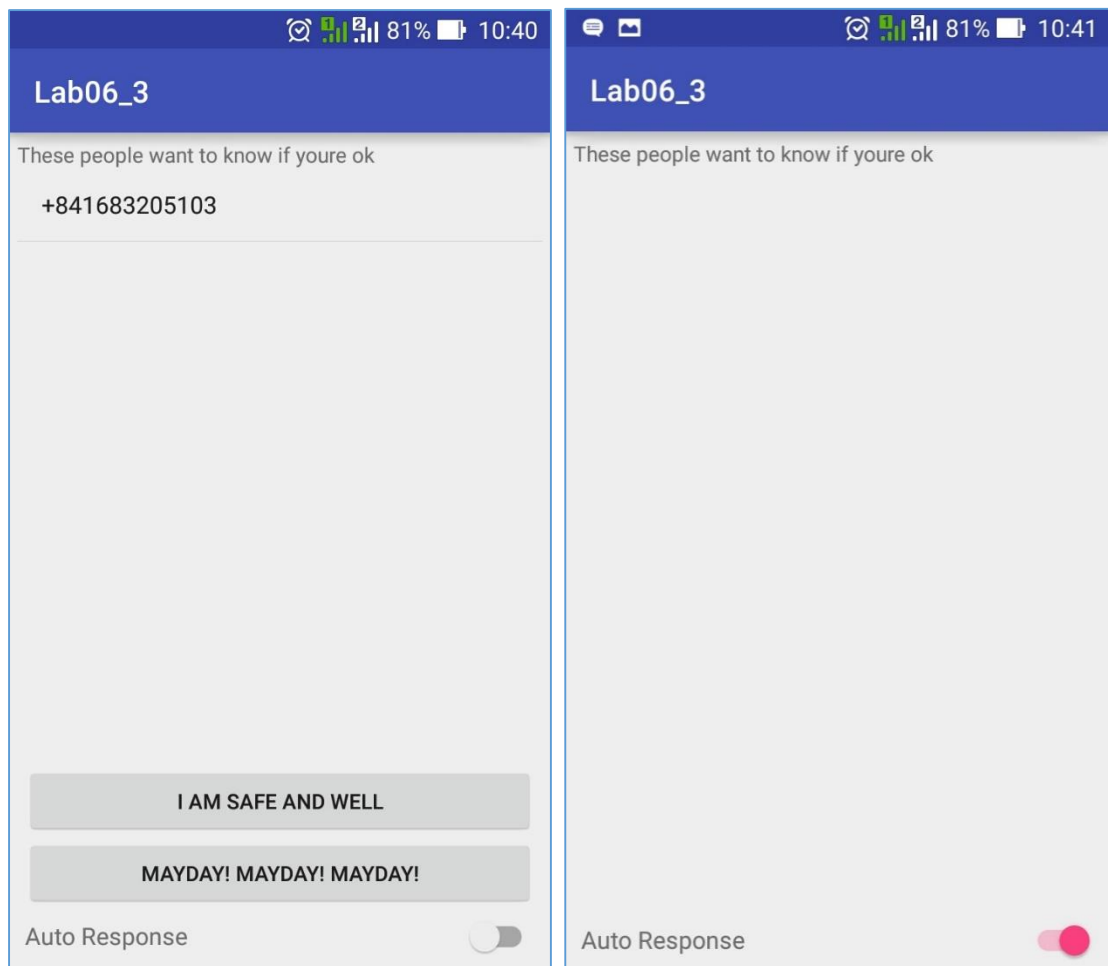
3. Bài 4: Dùng BroadcastReceiver để tạo app trả lời tin nhắn trong trường hợp khẩn cấp

a) Mô tả ứng dụng

- Ứng dụng lắng nghe tin nhắn đến và xử lý khi tin nhắn đến chứa keyword “are you ok?”. Ứng dụng cho phép người dùng tự phản hồi bằng 1 trong 2 thông điệp (bằng cách nhấn button tương ứng): i) “I am fine and safe. Worry not!”; ii) “Tell my mother I love her”. Ngoài ra, người dùng còn có thể sử dụng tùy chọn “Auto response” để tự động phản hồi “I am fine and safe. Worry not!” khi có tin nhắn đến.
- Ứng dụng sử dụng SmsReceiver (kế thừa từ BroadcastReceiver) đăng ký trong AndroidManifest nên cho dù ứng dụng bị tắt, việc lắng nghe tin nhắn đến vẫn được thực hiện. Khi tin nhắn đến sẽ xảy ra 2 trường hợp:

+ TH1: ứng dụng đang mở (chưa bị Stop/Destroy) thì SmsReceiver sẽ broadcast nội dung nhận được để MainActivity dùng một BroadcastReceiver khác được đăng ký bằng code để nhận và xử lý tin nhắn đến lên giao diện cho người sử dụng.

+ TH2: ứng dụng đang tắt, SmsReceiver sẽ tự động start MainActivity lên và gửi danh sách SĐT vừa gửi tin nhắn “are you ok?” đến MainActivity thông qua việc truyền dữ liệu qua Intent. MainActivity nhận dữ liệu và hiển thị lên giao diện cho người sử dụng. Trong trường hợp này, ứng dụng không dùng đến BroadcastReceiver thứ 2 được đăng ký bằng code như trường hợp 1.



b) Các bước thực hiện

- Xây dựng các file `dimens.xml`

Name	Value
<code>margin_base</code>	5dp
<code>margin_basex2</code>	10dp
<code>text_medium</code>	16sp

- Xây dựng file `strings.xml`

Name	Value
<code>i_am_safe_and_well</code>	I am Safe and Well
<code>mayday</code>	MAYDAY! MAYDAY! MAYDAY!
<code>auto_response</code>	Auto Response
<code>i_am_safe_and_well_worry_not</code>	I am safe and well. Worry not!
<code>tell_my_mother_i_love_her</code>	Tell my mother I love her.
<code>are_you_ok</code>	Are you OK?
<code>these_people_want_to_know_if_u_are_ok</code>	These people want to know if you\'re ok

- Cấp quyền trong Manifest

```
<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"/>
```

- Xây dựng giao diện

```
<?xml version="1.0" encoding="utf-8"?>
<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:padding="@dimen/margin_base"
    tools:context="uit.edu.vn.lab06_3.MainActivity">

    <TextView
        android:id="@+id/tv_request_list"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
```



```
        android:text="@string/these_people_want_to_know_if_u_are_ok"
    />

    <LinearLayout
        android:id="@+id/ll_control"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:orientation="vertical"
        android:padding="@dimen/margin_base">

        <LinearLayout
            android:id="@+id/ll_buttons"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <Button
                android:id="@+id/btn_safe"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/i_am_safe_and_well" />

            <Button
                android:id="@+id/btn_mayday"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="@string/mayday" />

        </LinearLayout>
    <LinearLayout
        android:layout_marginTop="@dimen/margin_base"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/auto_response"
            android:textSize="@dimen/text_medium" />

        <Switch
            android:id="@+id/sw_auto_response"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="false" />
    </LinearLayout>
</LinearLayout>

<ListView
    android:id="@+id/lv_messages"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_above="@id/ll_control"
    android:layout_below="@id/tv_request_list" />

</RelativeLayout>
```

- Nội dung code

Mô tả	Nội dung code
Khai báo biến	<pre> private ReentrantLock reentrantLock; private Switch swAutoResponse; private LinearLayout llButtons; private Button btnSafe, btnMayday; private ArrayList<String> requesters; private ArrayAdapter<String> adapter; private ListView lvMessages; private BroadcastReceiver broadcastReceiver; public static boolean isRunning; private SharedPreferences.Editor editor; private SharedPreferences sharedPreferences; private final String AUTO_RESPONSE = "auto_response"; </pre>
Xây dựng hàm findViewsByIds()	<pre> private void findViewsByIds() { swAutoResponse = (Switch) findViewById(R.id.sw_auto_response); llButtons = (LinearLayout) findViewById(R.id.ll_buttons); lvMessages = (ListView) findViewById(R.id.lv_messages); btnSafe = (Button) findViewById(R.id.btn_safe); btnMayday = (Button) findViewById(R.id.btn_mayday); } </pre>

Broadcast Receiver

LAB
06

PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG

Xây dựng hàm reply 1 tin nhắn:
- Xóa SĐT đã trả lời khỏi listview
- Trả lời tin nhắn

```
private void respond(String to, String response) {  
    reentrantLock.lock();  
    requesters.remove(to);  
    adapter.notifyDataSetChanged();  
    reentrantLock.unlock();  
  
    // Send the message  
    SmsManager sms = SmsManager.getDefault();  
    sms.sendTextMessage(to, null, response, null, null);  
}
```

Xây dựng hàm reply tất cả tin nhắn

```
public void respond(boolean ok) {  
    String okString = getString(  
        R.string.i_am_safe_and_well_worry_not);  
    String notOkString = getString(  
        R.string.tell_my_mother_i_love_her);  
    String outputString = ok ? okString : notOkString;  
    ArrayList<String> requestersCopy =  
        (ArrayList<String>) requesters.clone();  
    for (String to : requestersCopy)  
        respond(to, outputString);  
}
```

Xây dựng hàm xử lý khi nhận danh sách các SĐT gửi tin nhắn “are you ok?” tới máy mình:

- Cập nhật listview hiển thị danh sách SĐT.
- Nếu chế độ autoResponse được bật thì gọi hàm trả lời.

```
public void processReceiveAddresses(ArrayList<String> addresses) {
    for (int i = 0; i < addresses.size(); i++) {
        if (!requesters.contains(addresses.get(i))) {
            reentrantLock.lock();
            requesters.add(addresses.get(i));
            adapter.notifyDataSetChanged();
            reentrantLock.unlock();
        }

        // Check to response automatically
        if (swAutoResponse.isChecked()) respond(true);
    }
}
```

Xử lý các sự kiện click và change Switch:

- onClick của btnSafe và btnMayday: reply tin nhắn.
- onCheckedChangeListener của switch autoResponse thì cập nhật giao diện (ẩn 2 btn trả lời) và lưu lại trạng thái của switch xuống

```
private void handleClickListener() {
    // Handle onClickListener
    btnSafe.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            respond(true);
        }
    });

    btnMayday.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            respond(false);
        }
    });

    swAutoResponse.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
```

Broadcast Receiver

LAB
06

PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG

Sharepreference để load lại trong lần chạy app tiếp theo.	<pre>if (isChecked) llButtons.setVisibility(View.GONE); else llButtons.setVisibility(View.VISIBLE); // Save auto response setting editor.putBoolean(AUTO_RESPONSE, isChecked); editor.commit(); }); }</pre>
Xây dựng hàm khởi tạo BroadcastReceiver chờ nhận dữ liệu từ SmsReceiver	<pre>private void initBroadcastReceiver() { broadcastReceiver = new BroadcastReceiver() { @Override public void onReceive(Context context, Intent intent) { // Get ArrayList addresses ArrayList<String> addresses = intent. getStringArrayListExtra(SmsReceiver.SMS_MESSAGE_ADDRESS_KEY); // Process these addresses processReceiveAddresses(addresses); } }; }</pre>
Đăng ký và hủy đăng ký BroadcastReceiver	<pre>@Override protected void onResume() { super.onResume(); }</pre>

Broadcast Receiver

LAB
06

PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG

bằng code khi
MainActivity
Resume/Stop; biến
isRunning cho biết
activity đang chạy hay
đã bị tắt

```
isRunning = true;

// Make sure broadcastReceiver was initied
if (broadcastReceiver == null) initBroadcastReceiver();

// RegisterReceiver
IntentFilter intentFilter = new IntentFilter(SmsReceiver.SMS_FORWARD_BROADCAST_RECEIVER);
registerReceiver(broadcastReceiver, intentFilter);
}

@Override
protected void onStop() {
    super.onStop();
    isRunning = false;

    // UnregisterReceiver
    unregisterReceiver(broadcastReceiver);
}
```

Xây dựng hàm khởi tạo các biến:

- Khởi tạo giá trị ban đầu cho các biến.
- Load lại trạng thái của switch autoResponse
- Gọi hàm khởi tạo broadcastReceiver

```
private void initVariables() {  
    sharedPreferences = getPreferences(MODE_PRIVATE);  
    editor = sharedPreferences.edit();  
    reentrantLock = new ReentrantLock();  
    requesters = new ArrayList<String>();  
    adapter = new ArrayAdapter<String>(this,  
        android.R.layout.simple_list_item_1, requesters);  
    lvMessages.setAdapter(adapter);  
  
    // Load auto response setting  
    boolean autoResponse = sharedPreferences.  
        getBoolean(AUTO_RESPONSE, false);  
    swAutoResponse.setChecked(autoResponse);  
    if (autoResponse) llButtons.setVisibility(View.GONE);  
  
    initBroadcastReceiver();  
}
```

Gọi các hàm trong onCreate()

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    findViewsByIds();  
    initVariables();  
    handleClickListener();  
}
```

Xây dựng SmsReceiver kế thừa từ BroadcastReceiver:

- Lấy danh sách các tin nhắn đến
- Lập danh sách SĐT đã gửi tin nhắn có chứa keyword "are you OK?"
- Kiểm tra trạng thái của MainActivity:

+ Nếu MainActivity đã bị Stop/Destroy thì start lại MainActivity và truyền danh sách các SĐT qua cho MainActivity -> **phần này vẫn đang bỏ trống**

+ Nếu MainActivity đang chạy thì Forward (Broadcast) danh sách này để broadcastReceiver thứ 2 (được đăng ký ở

```
public class SmsReceiver extends BroadcastReceiver {

    public static final String SMS_FORWARD_BROADCAST_RECEIVER = "sms_forward_broadcast_receiver";
    public static final String SMS_MESSAGE_ADDRESS_KEY = "sms_messages_key";

    @Override
    public void onReceive(Context context, Intent intent) {
        String queryString = "Are you OK?".toLowerCase();

        Bundle bundle = intent.getExtras();
        if (bundle != null) {
            Object[] pdus = (Object[]) bundle.get("pdus");
            SmsMessage[] messages = new SmsMessage[pdus.length];
            for (int i = 0; i < pdus.length; i++) {
                if (android.os.Build.VERSION.SDK_INT >= 23) {
                    messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i], "");
                } else {
                    messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
                }
            }
        }
    }
}
```


Broadcast Receiver

LAB
06

PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG

MainActivity) nhận và xử lý tiếp.

```
// Create ArrayList of OriginatingAddress of messages which contain queryString
ArrayList<String> addresses = new ArrayList<>();

for (SmsMessage message : messages) {
    if (message.getMessageBody().toLowerCase().contains(queryString)) {
        addresses.add(message.getOriginatingAddress());
    }
}

if (addresses.size() > 0) {
    if (!MainActivity.isRunning) {
        // Start MainActivity if it stopped
    } else {
        // Forward these addresses to MainActivity to process
        Intent iForwardBroadcastReceiver = new Intent(SMS_FORWARD_BROADCAST_RECEIVER);
        iForwardBroadcastReceiver.putStringArrayListExtra(SMS_MESSAGE_ADDRESS_KEY, addresses);
        context.sendBroadcast(iForwardBroadcastReceiver);
    }
}
}
```

Đăng ký SmsReceiver trong AndroidManifest

```
<receiver android:name=".SmsReceiver">
    <intent-filter>
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
```

Chạy thử chương trình. Hiện tại, ứng dụng đã có thể lắng nghe tin nhắn đến và hỗ trợ người dùng trả lời hoặc trả lời tự động. Tuy nhiên, ứng dụng chưa xử lý trường hợp khi activity bị tắt. Theo yêu cầu ban đầu: khi activity bị tắt, SmsReceiver phải tự start lại activity và gửi danh sách SĐT đến activity đó. Tiếp theo, activity nhận danh sách và xử lý như trường hợp activity không tắt. **Yêu cầu: sinh viên hoàn thành ứng dụng xử lý trường hợp trên.**

Hướng dẫn: sinh viên thực hiện theo các bước:

- Thêm code xử lý start lại MainActivity và gửi danh sách SĐT (biến addresses) thông qua phương thức

`iMain.putStringArrayListExtra(SMS_MESSAGE_ADDRESS_KEY, addresses);` trong SmsReceiver tại đoạn code sau:

```
if (addresses.size() > 0) {  
    if (!MainActivity.isRunning) {  
        // Start MainActivity if it stopped  
    } else {  
        // Forward these addresses to MainActivity to process  
        Intent iForwardBroadcastReceiver = new Intent(SMS_FORWARD_BROADCAST_RECEIVER);  
        iForwardBroadcastReceiver.putStringArrayListExtra(SMS_MESSAGE_ADDRESS_KEY, addresses);  
        context.sendBroadcast(iForwardBroadcastReceiver);  
    }  
}
```

Trong đó, intent mới (*iMain*) cần được set 2 Flags là *FLAG_ACTIVITY_CLEAR_TOP* và *FLAG_ACTIVITY_NEW_TASK*. Sinh viên tự bổ sung thêm những đoạn code cần thiết.

- Nhận danh sách SĐT (biến *addresses*) tại *MainActivity* và gọi hàm ***processReceiveAddresses*** để xử lý danh sách SĐT này. Nội dung xử lý trên, sinh viên thêm vào hàm *initVariable()*:

```
private void initVariables() {  
    sharedPreferences = getPreferences(MODE_PRIVATE);  
    editor = sharedPreferences.edit();  
    reentrantLock = new ReentrantLock();  
    requesters = new ArrayList<String>();  
    adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, requesters);  
    lvMessages.setAdapter(adapter);  
  
    // Load auto response setting  
    boolean autoResponse = sharedPreferences.getBoolean(AUTO_RESPONSE, false);  
    swAutoResponse.setChecked(autoResponse);  
    if (autoResponse) llButtons.setVisibility(View.GONE);  
  
    initBroadcastReceiver();  
  
    // Send addresses to broadcastReceiver  
}
```

III. THAM KHẢO THÊM (Tự xem thêm)

1. Yêu cầu Permission trong Android

2. Crash Report Tools

- <http://try.crashlytics.com/>
- <https://firebase.google.com/docs/crash/>

3. Sử dụng Facebook SDK, LinkedIn SDK, Google Play Services, Twitter Kit để login with Facebook, LinkedIn, Google, Twitter

- **Facebook:** <https://developers.facebook.com/docs/facebook-login/android>; <http://giasutinhoc.vn/android/tich-hop-login-cua-facebook-trong-ung-dung-android/>
- **LinkedIn:** <https://developer.linkedin.com/docs/android-sdk-auth>; <https://www.studytutorial.in/linkedin-integration-and-login-in-android-tutorial>
- **Google:** <https://developers.google.com/identity/sign-in/android/>; <http://www.androidhive.info/2014/02/android-login-with-google-plus-account-1/>
- **Twitter:** <https://dev.twitter.com/twitter-kit/android/twittercore>; <https://docs.fabric.io/android/twitter/log-in-with-twitter.html>