

Thực hành Buổi 6: LẬP TRÌNH MẠNG PHẦN 1 (Gồm 4 bài: 13, 14, 15, 16)

I. LÝ THUYẾT CẦN ÔN TẬP

- Hoạt động mạng (networking) là thuộc tính cơ bản của thiết bị di động
- Lướt web với Android
- Telephony API (SMS, Email)
- Khai thác Webservice dùng giao thức HTTP

II. THỰC HÀNH

Bài 13: Lập trình lướt web và kiểm tra loại kết nối mạng.

Mở một project mới có tên là **B6_1** có giao diện chính như hình 6.1 và thực hiện các chức năng như sau:

1. Nhúng trình duyệt web có sẵn dưới dạng một widget vào trong các Activity để duyệt web, có thể phóng to/thu nhỏ nội dung trang web được hiển thị.

Người dùng nhập URL của trang web vào EditText vào chọn Button “DUYỆT WEB” thì một màn hình mới được mở vào tải về trang web.

2. Kiểm tra điện thoại có kết nối mạng hay không và cho loại kết nối mạng đang sử dụng trên chương trình giả lập, hỗ trợ và hiển thị tình trạng kết nối: "No default network is currently active" nếu thông tin về kết nối là null, "Network is not connected" nếu thông tin trả về là không có kết nối; "Network not available" nếu thông tin trả về là mạng không sẵn sàng; "Network OK" nếu kết nối thành công, và hiển thị tên của kết nối ra màn hình khi người dùng ấn chọn Button “LOẠI KẾT NỐI”, hình 6.1.

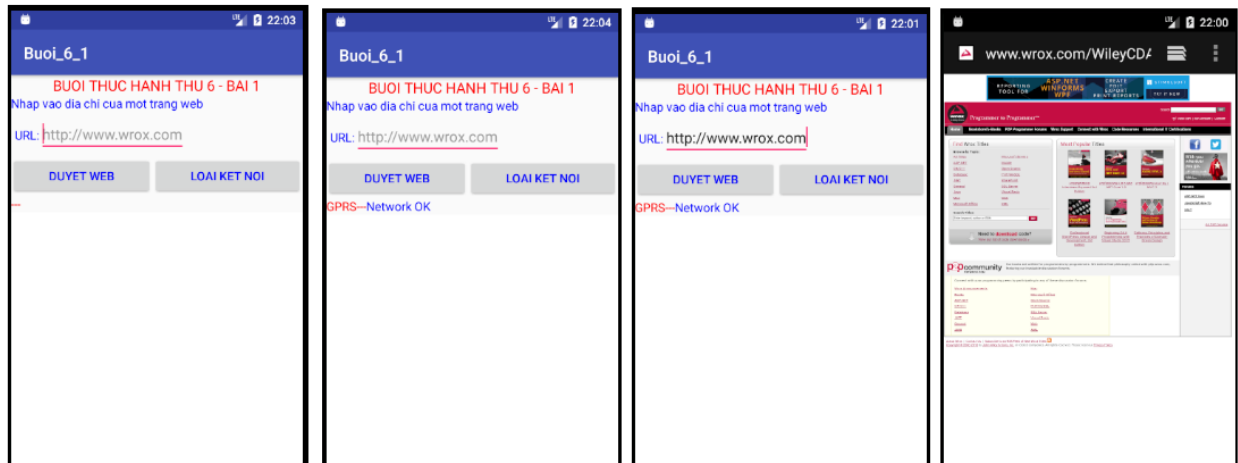
Nhớ thêm permission:

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

vào tập tin AndroidManifest.xml

Do Android emulator không hỗ trợ Bluetooth nên chỉ kiểm tra kết nối Mobile (3G) và Wifi.

Sinh viên tham khảo giáo trình, slide bài giảng và thực hiện, bài này không được cung cấp mã nguồn.



Hình 6.1.a: Giao diện khi khởi chạy

Hình 6.1.b: Khi chọn nút "LOAI KET NOI"

Hình 6.1.c: Nhập vào URL và ấn nút "DUYET WEB"

Hình 6.1.d: Tải về trang web

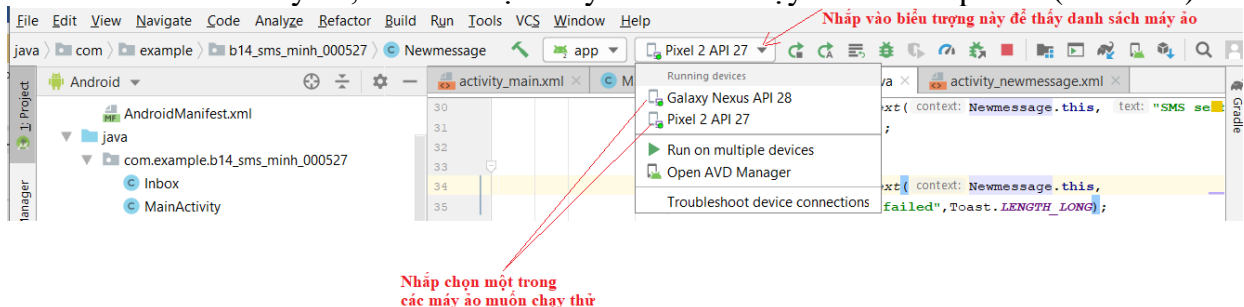
Hình 6.1

Bài 14: Thực hành lập trình gửi, nhận và xem tin nhắn SMS với Telephony API

Viết chương trình gửi tin nhắn SMS. Chương trình gồm các chức năng: tạo tin nhắn mới, gửi tin nhắn và xem tin nhắn trong hộp thư đến (Inbox).

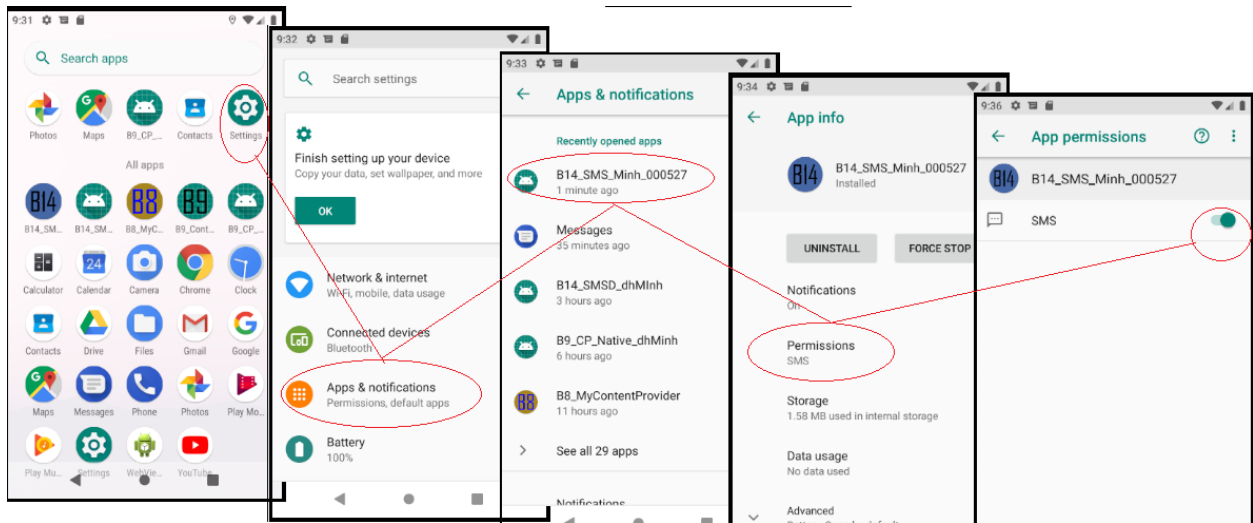
Ứng dụng được thử bằng cách mở đồng thời 2 máy ảo (emulator). Để chạy hai máy ảo cùng lúc, trước tiên ta cần tạo hai emulator và chạy project 2 lần với 2 điện thoại (emulator) khác nhau (số điện thoại tương ứng với máy ảo được mở trước là 5554 và máy ảo được mở sau là 5556), sau đó thực hiện việc gửi tin nhắn qua lại giữa 2 điện thoại này.

- Cách chạy thử ứng dụng với lần lượt 2 máy ảo khác nhau trên Android Studio. Tại cửa sổ hiển thị máy ảo trên thanh công cụ, ta nhấp vào biểu tượng ▼ để hiển thị danh sách máy ảo, sau đó chọn máy ảo muốn chạy thử và nhấp Run (Hình 6.2).



Hình 6.2

- Để gửi tin nhắn, ta cần phải check SMS permission on trên máy ảo. Cách thực hiện như sau: Chọn Setting → Chọn Apps¬ifications → Chọn ứng dụng gửi tin nhắn → Chọn Permissions → Check on cho SMS (Hình 6.3).



Hình 6.3

Hướng dẫn thực hành:

Bước 1: Tạo project mới có tên là **B14_MSSV_Tên SV**, Activity chính là **MainActivity.java** và layouty là **activity_main.xml**.

Bước 2: Lập trình layout chính activity_main.xml để có giao diện chính như Hình 6.4



Hình 6.4. Giao diện chính

Trong ví dụ này, giao diện chính gồm có 4 TextView và 1 ImageView (SV chọn hình tùy ý, chỉ để trang trí cho đẹp). Trong đó:

- TextView 1 dùng để hiển thị chủ đề của ứng dụng là “SMS MANAGER”,

3 TextView còn lại *có chức năng tương tác chạm tay* (TextView cũng có thể tương tác được như Button):

- TextView “New message” có tác dụng mở một Intent gọi đến lớp “NewMessage” để nhập vào tin nhắn và gửi tin nhắn;
- TextView “Inbox” có tác dụng mở một Intent gọi đến lớp “Inbox” để xem tin nhắn;
- TextView “Exit” có tác dụng gọi phương thức exit để thoát khỏi ứng dụng.

Mã nguồn tham khảo cho Layout chính:

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SMS MANAGER"
        android:textSize="20sp"
        android:textColor="#f00"
        android:background="#AEAEF7"
        android:layout_marginTop="20dp"
        android:gravity="center"
    />

    <TextView
        android:id="@+id/newmessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="NEW MESSAGE"
        android:textSize="20sp"
        android:textColor="#04259C"
        android:background="#AEF5F7"
        android:layout_marginTop="20dp"
        android:gravity="center"
    />

    <TextView
        android:id="@+id/inbox"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="INBOX"
        android:textSize="20sp"
        android:textColor="#062B9B"
        android:background="#AEF7B5"
        android:layout_marginTop="20dp"
        android:gravity="center"
    />
```

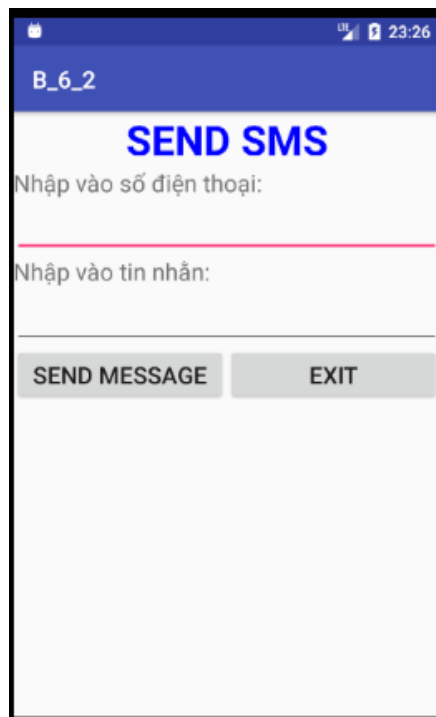
```

<TextView
    android:id="@+id/exit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="EXIT"
    android:textSize="20sp"
    android:textColor="#000000"
    android:background="#E4AEF7"
    android:layout_marginTop="20dp"
    android:gravity="center"
/>
<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/tren_song"
/>
</LinearLayout>

```

Bước 3: Tạo một Activity để soạn và gửi tin nhắn SMS, trong ví dụ này nó đặt đặt tên là **Newmessage**. Giao diện của nó gồm có 3 textView, 2 EditText và 2 Button (Hình 6.5):

- TextView 1: Hiển thị chủ đề là SEND SMS;
- TextView 2: Hiển thị text yêu cầu nhập số phone, “Enter number phone”;
- TextView 3: Hiển thị text yêu cầu nhập message, “Enter message”.
- EditText 1: dùng để nhập số phone;
- EditText 2: dùng để nhập message.
- Button 1: ra lệnh gửi message, “SEND MESSAGE”;
- Button 2: EXIT.



Hình 6.4: Giao diện của lớp Newmessage

Mã nguồn layout của lớp Newmessage (*SV nên tìm hiểu và nhập từng lệnh, không nên copy và paste*).

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Newmessage">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="SEND SMS"
        android:textSize="35sp"
        android:textStyle="bold"
        android:textColor="#0000ff"
        android:gravity="center" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="Nhập vào số điện thoại:"/>
    <EditText
        android:id="@+id/phone"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"/>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="Nhập vào tin nhắn:"/>
    <EditText
        android:id="@+id/message"
        android:layout_width="fill_parent"
        android:layout_height="100px"
        android:textSize="20sp"/>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button
            android:id="@+id/send"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Send message"
            android:textSize="20sp"
            android:layout_weight="1"/>
        <Button
```

```

        android:id="@+id/exitinbox"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Exit"
        android:textSize="20sp"
        android:layout_weight="1"/>
    </LinearLayout>
</LinearLayout>

```

Mã nguồn của lớp Newmessage như sau (SV nên tìm hiểu và nhập từng lệnh, không nên copy và paste):

```

package com.example.b14_sms_minh_000527;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Newmessage extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_newmessage);
        Button btn = (Button) this.findViewById(R.id.send);
        Button exit = (Button) this.findViewById(R.id.exitinbox);

        btn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub
                EditText phone = (EditText).findViewById(R.id.phone);
                EditText message = (EditText).findViewById(R.id.message);
                try{
                    SmsManager sms = SmsManager.getDefault();
                    sms.sendTextMessage(phone.getText().toString(), null,
                        message.getText().toString(), null, null);
                    Toast toast = Toast.makeText(Newmessage.this,
                        "SMS sent successful",
                        Toast.LENGTH_LONG);
                    toast.show();
                } catch (Exception e) {
                    Toast toast = Toast.makeText(Newmessage.this,
                        "Sending SMS failed", Toast.LENGTH_LONG);
                    toast.show();
                    e.printStackTrace();
                }
            }
        });
        exit.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub
                finish();
            }
        });
    }
}

```

```

    }
    });
}
}

```

Bước 4: Tạo lớp **Inbox**, đây là một **ListActivity** có chức năng hiển thị một danh sách các mục (item) bằng cách gắn vào một nguồn dữ liệu như một mảng hoặc con trỏ, và đưa ra xử lý sự kiện khi người dùng chọn một mục. ListActivity làm chủ một đối tượng ListView có thể được liên kết với các nguồn dữ liệu khác nhau, thường là một mảng hoặc một tổ chức con trỏ điều khiển kết quả truy vấn). Lớp này dùng để duyệt và xem tin nhắn SMS. Cách làm: Nhấp chuột phải vào thư mục tên miền → Chọn New → Chọn Activity → Chọn Empty Activity → Đặt tên lớp → Chọn Finish → Trong Activity sửa lại là **extends** ListActivity.

Mã nguồn của Layout của lớp Inbox:

```

<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Inbox">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:id="@+id/row"/>
    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/list"/>
</LinearLayout>

```

Mã nguồn Java của lớp Inbox như sau:

```

package com.example.b14_sms_minh_000527;

import androidx.appcompat.app.AppCompatActivity;

import android.app.ListActivity;
import android.os.Bundle;
import android.database.Cursor;
import android.net.Uri;
import android.view.KeyEvent;
import android.view.View;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.Toast;

public class Inbox extends ListActivity {
    private ListAdapter adapter;
}

```



```

@SuppressWarnings("deprecation")
@Override
public void onCreate(Bundle bundle) {
    super.onCreate(bundle);

    Cursor c = getResolver().query( Uri.parse("content://sms/inbox"), null,
        null, null, null);
    startManagingCursor(c);
    String[] columns = new String[] { "body" };
    int[] names = new int[] { R.id.row };
    adapter=new SimpleCursorAdapter(this, R.layout.activity_inbox,
        c, columns, names);
    setListAdapter(adapter);
}
@Override
protected void onItemClick(ListView l, View v, int position, long ida) {
    super.onItemClick(l, v, position, ida);
    Cursor mycursor = (Cursor) getListView().getItemAtPosition(position);
    Toast toast = Toast.makeText(Inbox.this, "From " +
        mycursor.getString(mycursor.getColumnIndex("address")) + ":\n" +
        mycursor.getString(mycursor.getColumnIndex("body")),
        Toast.LENGTH_LONG);
    toast.show();
}
}

```

Bước 5: Tạo lớp Java được đặt tên là SMS, đây là một **BroadcastReceiver** có chức năng nhận tin nhắn SMS (mở 1 java class mới tên là SMS sau đó extends BroadcastReceiver). Mã nguồn Java như sau:

```

package com.example.b14_sms_minh_000527;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.telephony.SmsMessage;
import android.util.Log;

public class SMS extends BroadcastReceiver {
    private static final String ACTION = "android.provider.Telephony.SMS_RECEIVED";
    @Override
    public void onReceive(Context context, Intent intent)
    {
        if(intent!=null && intent.getAction()!=null &&
            ACTION.compareToIgnoreCase(intent.getAction())==0)
        {
            Object[]pduArray = (Object[]) intent.getExtras().get("pdus");
            SmsMessage[] messages = new SmsMessage[pduArray.length];
            for (int i = 0; i<pduArray.length; i++) {
                messages[i] = SmsMessage.createFromPdu((byte[])pduArray [i]);
                Log.d("MySMSMonitor", "From: " + messages[i].getOriginatingAddress());
                Log.d("MySMSMonitor", "Msg: " + messages[i].getMessageBody());
            }
            Log.d("MySMSMonitor","SMS Message Received.");
        }
    }
}

```

Bước 6: Lập trình MainActivity.java, xử lý tương tác của người dùng.

Mã nguồn tham khảo như sau:

```
package com.example.b14_sms_minh_000527;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView newmessage=(TextView) findViewById(R.id.newmessage);
        TextView inbox=(TextView) findViewById(R.id.inbox);
        final TextView exit=(TextView) findViewById(R.id.exit);
        newmessage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent newmessage=new Intent(MainActivity.this,Newmessage.class);
                startActivity(newmessage);
            }
        });
        inbox.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent inbox=new Intent(MainActivity.this,Inbox.class);
                startActivity(inbox);
            }
        });
        exit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                System.exit(0);
            }
        });
    }
}
```

Bước 7: Bổ sung các permission sau đây vào tập tin Androidmanifest.xml

```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.READ_SMS"/>
```

Và **đăng ký BroadcastReceiver trong Manifest**, giúp chương trình hoạt động như service chạy nền. Những dòng được in đậm và highlight trong mã nguồn của tập tin Androidmanifest.xml như sau:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.doanhoaminh.b_6_2">

    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.WRITE_SMS"/>
    <uses-permission android:name="android.permission.READ_SMS"/>

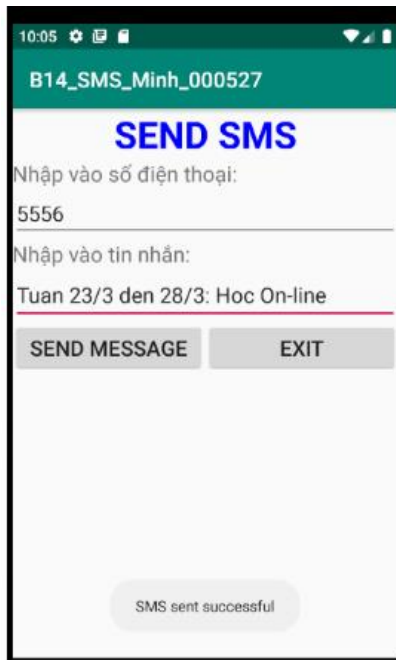
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Newmessage" />
        <activity android:name=".Inbox"/>
        <receiver android:name="SMS">
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>

    </application>
</manifest>

```

Bước 8: Chạy thử với 2 Emulator khác nhau và gỡ lỗi nếu có. Thực hiện soạn và gửi tin nhắn vài lần, kết quả hiển thị khi ấn chọn nút “Inbox” như hình 6.5.b.



Hình 6.5.a. Nhập vào tin nhắn và nhấn nút SEND MESSAGE



Hình 6.5.b. Nhấn nút INBOX ở máy nhận



Hình 6.5.c. Hiển thị tin nhắn đã nhận được

Hình 6.5. nhập số điện thoại, nhập message, ấn chọn text SEND MESSAGE để gửi tin nhắn, báo tin nhắn gửi thành công (“Sucees”). Ấn chọn text Inbox để duyệt và xem tin nhắn.

Bài 15: Thực hành lập trình thực hiện kết nối HTTP bằng phương thức GET

Trong trường hợp HTTP-GET, dữ liệu được gửi tường minh, chúng ta có thể nhìn thấy trên URL, đây là lý do khiến nó không bảo mật như loại HTTP-POST. GET còn bị giới hạn số ký tự trên URL của trình duyệt.

GET thực thi nhanh hơn POST vì những dữ liệu gửi đi luôn được lưu lại trong cache của trình duyệt. Khi gửi dữ liệu đi trình duyệt sẽ xem trong cache có kết quả tương ứng với yêu cầu đó không và trả về ngay, không cần phải thực thi các yêu cầu đó ở phía Server. Nên dùng phương thức này trong trường hợp dữ liệu ít thay đổi vì sẽ truy xuất và xử lý nhanh hơn.

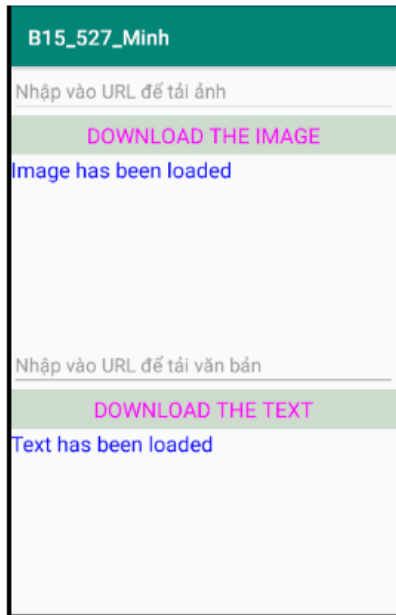
Các bước sử dụng phương thức GET

- (1) Khởi tạo một đối tượng InputStream.
- (2) Mở một kết nối HTTP với một URL từ xa.
- (3) Thiết lập các thuộc tính của kết nối.
- (4) Lấy đáp ứng HTTP_OK để biết kết nối đã được thiết lập hay chưa.

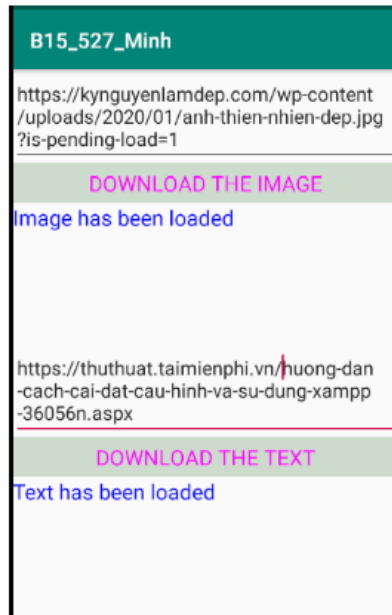
- (5) Nếu kết nối được thiết lập thì tiến hành lấy đối tượng InputStream từ kết nối để bắt đầu tải dữ liệu từ Server

Hướng dẫn thực hành

Mở project mới có tên là **B15_MSSV_Tên SV**, có Activity chính là **MainActivity** và giao diện chính như **Hình 6.6.b**. Khi người dùng nhập URL vào EditText 1 (có dòng gợi ý là “Nhập vào URL để tải ảnh”) và EditText 2 (có dòng gợi ý là “Nhập vào URL để tải văn bản”) (Hình 6.6.b), sau đó nhấn nút “DOWNLOAD THE IMAGE” và nhấn nút “DOWNLOAD THE TEXT” thì hình ảnh và văn bản được tải về và hiển thị (Hình 6.6.c). Văn bản tải về có thể trực tiếp đọc (khi bị khuất do kích thước lớn).



Hình 6.6.a. Giao diện chính



Hình 6.6.b. Nhập vào URL



Hình 6.6.c. Sau khi nhấn nút Download the image và nút Download the text

➤ Tập tin Layout (Tham khảo)

```
<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <EditText
        android:id="@+id/nhapurl1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```

        android:hint="Nhập vào URL để tải ảnh"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:text="DOWNLOAD THE IMAGE"
    android:textColor="#f0f"
    android:background="#CCDBCC"
    android:padding="5dp"
    android:gravity="center"
    android:id="@+id/tvLoadimg" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:textColor="#00f"
    android:text="Image has been loaded"
    android:id="@+id/tvURLimg" />
<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/img"
    android:layout_gravity="center_vertical"
    android:contentDescription="img1"
    android:layout_weight="1" />
<EditText
    android:id="@+id/nhapurl2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Nhập vào URL để tải văn bản"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:text="DOWNLOAD THE TEXT"
    android:background="#CCDBCC"
    android:textColor="#f0f"
    android:padding="5dp"
    android:gravity="center"
    android:id="@+id/tvLoadtext" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:text="Text has been loaded"
    android:textColor="#00f"
    android:id="@+id/tvURLtext" />
<TextView
    android:id="@+id/tv"
    android:layout_weight="1"
    android:layout_width="match_parent"

```

```
        android:layout_height="match_parent"
        android:textSize="20sp"
        android:scrollbarStyle="insideInset"/>
</LinearLayout>
```

➤ Mã nguồn tham khảo cho MainActivity.java

Lưu ý: đây là mã nguồn mà URL được khai báo trực tiếp trong khi lập trình, SV tham khảo và chỉnh sửa lại để lấy URL do người dùng nhập vào các EditText. Sinh viên có thể lưu ảnh và văn bản trong localhost và nhập vào URL như trong hướng dẫn để download ảnh và văn bản trong localhost thay vì tải về từ một trang web thật.

```
package example.com.b6_3;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.widget.ImageView;
import android.widget.TextView;

import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;

public class GetHttpMainActivity extends AppCompatActivity {
    public InputStream OpenHttpConnection(String urlString) throws IOException
    {

        /***** (1) Tạo đối tượng InputStream *****/

        InputStream in = null;
        int response = -1;
        URL url = new URL(urlString);

        // Mở kết nối tới urlString
        URLConnection conn = url.openConnection();

        // Nếu không là một HttpURLConnection
        // thì thông báo không có kết nối HTTP
        if (!(conn instanceof HttpURLConnection))
            throw new IOException("Not an HTTP connection");
        try {

            /***** (2) Mở kết nối HTTP với URL từ xa *****/

            HttpURLConnection httpConn = (HttpURLConnection) conn;
```

```

/***** (3) Thiết lập các thuộc tính kết nối *****/

//Hàm đặt cờ để xác định giao thức sẽ
// tự động theo địa chỉ mới (true) hoặc không
httpConn.setInstanceFollowRedirects(true);

//Xác định phương thức kết nối là GET
httpConn.setRequestMethod("GET");

//Kết nối với máy chủ
httpConn.connect();

/* (4) Lấy đáp ứng HTTP_OK để biết kết nối đã được thiết lập hay chưa */
response = httpConn.getResponseCode();
Log.w("Response Code", ""+response);

/*****
* (5) Nếu kết nối được thiết lập thì tiến hành
lấy đối tượng InputStream từ kết nối để lấy dữ liệu từ Server
*****/

if (response == HttpURLConnection.HTTP_OK) {
    in = httpConn.getInputStream();
}
} catch (Exception e) {
    Log.e("Networking", e.getMessage());
}

return in;
/***** LƯU Ý *****/
*Để kết nối Internet, cần thêm lệnh sau vào AndroidManifest.xml: *
*<uses-permission android:name="android.permission.INTERNET" /> *
*****/
}
/*****
TẢI DỮ LIỆU HÌNH ẢNH THÔNG QUA GET
*****/
/**
* Phương thức DownloadImage() lấy URL để thực hiện việc tải ảnh về.
*
* @param URL
* @return bitmap
*/
private Bitmap DownloadImage(String URL) {

    Bitmap bitmap = null;
    InputStream in = null;
    try {
        // Mở kết nối đến Server, phương thức đã được định nghĩa ở trên
        in = OpenHttpConnection(URL);

        if (in == null) {
            Log.e("Image URL", "Check connection or URL again!");
            return bitmap;
        }
        // Tải dữ liệu thông qua InputStream in

```



```

        // và giải mã vào đối tượng bitmap
        bitmap = BitmapFactory.decodeStream(in);
        in.close();
    } catch (IOException e) {
        Log.e("NetworkingActivity", e.getLocalizedMessage());
    }
    return bitmap;
}

/*****
 * Do việc tải dữ liệu từ máy chủ thường mất thời gian,
 * để giao diện chính không bị treo trong thời gian tải dữ liệu,
 * ta cần tạo luồng riêng để tải tài nguyên trên mạng,
 * AsyncTask cho phép thực hiện tác vụ chạy nền trong thời gian riêng biệt
 * và trả kết quả trong một luồng UI
 * Bằng cách này, ta có thể thực hiện hoạt động nền mà
 * không cần xử lý vấn đề luồng phức tạp
 *****/

public class DownloadImageTask extends AsyncTask<String, Void, Bitmap> {

    //Thực hiện tải dữ liệu
    // Khi hoàn tất, kết quả được truyền vào phương thức onPostExecute
    @Override
    protected Bitmap doInBackground(String... urls) {
        return DownloadImage(urls[0]);
    }

    @Override
    protected void onPostExecute(Bitmap result) {

        ImageView img = (ImageView) findViewById(R.id.img);
        //Hiển thị ảnh trên màn hình
        img.setImageBitmap(result);
        TextView tv = (TextView) findViewById(R.id.tvURLimg);
        if (result != null)
            tv.setText("Got image.");
        else
            tv.setText("Can't get image.");
    }
}

/*****
 TẢI DỮ LIỆU TEXT THÔNG QUA GET
 *****/
/**
 * Phương thức tải dữ liệu từ URL
 *
 * @param URL
 * @return chuỗi kí tự tải về từ URL
 */
private String DownloadText(String URL) {

    int BUFFER_SIZE = 2000;
    InputStream in = null;
    String str = "";

```

```

        try {
            in = OpenHttpConnection(URL);
        } catch (IOException e) {
            Log.e("Networking", e.getLocalizedMessage());
            return str;
        }

        if (in == null){
            Log.e("Text URL", "Check connection or URL again!");
            return str;
        }

        try {
            InputStreamReader isr = new InputStreamReader(in);
            int charRead;
            char[] inputBuffer = new char[BUFFER_SIZE];

            while ((charRead = isr.read(inputBuffer)) > 0) {
                //Chuyển chars thành String
                String readString = String.valueOf(inputBuffer, 0, charRead);
                str += readString;
                inputBuffer = new char[BUFFER_SIZE];
            }
            in.close();
        } catch (IOException e) {
            Log.e("Networking", e.getLocalizedMessage());
            return str;
        }
        return str;
    }

    public class DownloadTextTask extends AsyncTask<String, Void, String> {

        protected String doInBackground(String... urls) {
            return DownloadText(urls[0]);
        }

        @Override
        protected void onPostExecute(String result) {

            TextView tv1 = (TextView) findViewById(R.id.tv);
            tv1.setText(result);
            TextView tv = (TextView) findViewById(R.id.tvURLtext);
            if (!result.trim().equals(""))
                tv.setText("Got text.");
            else
                tv.setText("Can't get text.");
        }
    }

    /*****
     * THỰC THI CÁC THAO TÁC TẢI DỮ LIỆU
     *****/
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_get_http_main);
    }

```

```

// Nếu dùng địa chỉ localhost(127.0.0.1) để chạy thử và kiểm lỗi,
// máy ảo sẽ tự lấy địa chỉ loopback của chính nó để thực thi
// như thế ứng dụng không truy cập được server bên ngoài.
// Để truy cập vào localhost từ máy ảo,
// cần dùng 1 trong các địa chỉ sau:
//Địa chỉ của máy tính: lấy địa chỉ thật của máy đang dùng
//Hoặc nếu dùng máy ảo AVD: 10.0.2.2
//Hoặc nếu dùng máy ảo genymotion: 10.0.3.2
String URL = "http://10.0.2.2:8080/Buoi_6/";

//Thực thi tải dữ liệu hình ảnh
new DownloadImageTask().execute(URL+"cantho_3.png");

//Thực thi tải dữ liệu văn bản
new DownloadTextTask().execute(URL + "Thuhttpget.txt");
}
}

```

Ghi chú :

Vì emulator có thể không kết nối được đến các URL ở xa, để bảo đảm có thể chạy thử được ta dùng Localhost. Các bước chuẩn bị (trước khi chạy ứng dụng) như sau:

- 1) Thêm thư mục, đặt tên là **"Buoi_6"**, vào thư mục htdocs của xampp, đường dẫn như sau: **C:\xampp\htdocs\Buoi_6**.
- 2) Chép một ảnh png vào thư mục **Buoi_6** (nên chọn ảnh **cantho_3.png** nếu muốn giữ nguyên địa chỉ URL như trong mã nguồn hướng dẫn).
- 3) Mở Notepad tạo tập tin văn bản (nội dung tùy ý, phần mở rộng là .txt), lưu vào thư mục Buoi_6 với tên là **Thuhttpget.txt** (nếu muốn giữ nguyên địa chỉ URL như trong mã nguồn hướng dẫn).
- 4) Mở XAMPP control Panel, kích hoạt (start) Apache và MySQL.

Bài 16: Thực hành lập trình thực hiện kết nối HTTP bằng phương thức POST

POST bảo mật hơn GET vì dữ liệu được gửi ngầm, không xuất hiện trên URL. Khi dùng phương thức POST thì Server luôn thực thi và trả về kết quả cho Client. Đối với những dữ liệu luôn được thay đổi và bảo mật thì nên sử dụng phương thức POST.

Các bước kết nối HTTP bằng phương thức POST

- (1) Khởi tạo một đối tượng InputStream từ URL và đối số truyền vào.

- (2) Mở một kết nối HTTP với một URL từ xa.
- (3) Thiết lập các thuộc tính của kết nối.
- (4) Gửi đi POST request.
- (5) Lấy đáp ứng HTTP_OK để biết kết nối đã được thiết lập hay chưa.
- (6) Nếu kết nối được thiết lập thì tiến hành lấy đối tượng InputStream từ kết nối để bắt đầu tải dữ liệu từ Server

Mở project mới có tên là B16_MSSV_Tên SV, thực hiện kết nối HTTP POST yêu cầu được gửi theo một luồng riêng, có tập tin Layout và Activity chính như sau (phần giải thích được ghi trực tiếp trong mã nguồn):

➤ Layout

```
<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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.b6_4.B6_4_MainActivity" >

    <TextView
        android:id="@+id/tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="Try HTTP POST" />

</RelativeLayout>
```

➤ Java code

```
package com.example.b6_4;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.widget.TextView;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
```

```

import java.net.URLConnection;
import java.net.URLEncoder;
import java.util.Iterator;

public class B6_4_Activity extends AppCompatActivity {

    public InputStream OpenHttpConnection(PostComponent component) throws
IOException {

    /** (1) Khởi tạo một đối tượng InputStream từ URL và đối số truyền vào */
        InputStream in = null;
        int response = -1;
        URL url = new URL(component.getURL());

        // Mở kết nối tới url
        URLConnection conn = url.openConnection();

        // Nếu không là một HttpURLConnection
        // thì thông báo không có kết nối HTTP
        if (!(conn instanceof HttpURLConnection))
            throw new IOException("Not an HTTP connection");
        try {

            /******* (2) Mở kết nối HTTP với URL từ xa *****/

            HttpURLConnection httpConn = (HttpURLConnection) conn;

            /******* (3) Thiết lập các thuộc tính kết nối *****/

            //Thời gian đọc InputStream tối đa 15000ms
            httpConn.setReadTimeout(15000);

            //Thời gian tối đa giữ kết nối với Server
            httpConn.setConnectTimeout(15000);

            //Phương thức kết nối là POST
            httpConn.setRequestMethod("POST");

            //Đặt cờ chỉ ra kết nối này cho phép có Output hoặc Input
            httpConn.setDoInput(true);
            httpConn.setDoOutput(true);

            /******* (4) Gửi đi POST request *****/

            OutputStream os = conn.getOutputStream();
            BufferedWriter writer = new BufferedWriter(
                new OutputStreamWriter(os, "UTF-8"));

            //Hàm getPostDataString dùng để chuyển dữ liệu thành chuỗi
            writer.write(getPostDataString(component.getParam()));

            writer.flush();
            writer.close();
            os.close();

            /** (5) Lấy đáp ứng HTTP_OK để biết kết nối đã được thiết lập hay chưa */

```

```

        response = httpConn.getResponseCode();
        Log.w("Response Code", "" + response);

/*****
    * (6) Nếu kết nối được thiết lập thì tiến hành
    * lấy đối tượng InputStream từ kết nối để lấy dữ liệu từ Server
*****/

        if (response == HttpURLConnection.HTTP_OK) {
            in = httpConn.getInputStream();
        }
    } catch (Exception e) {
        Log.e("Networking", e.getMessage());
    }
    return in;
}

/***** LƯU Ý *****/
* Để kết nối Internet, cần thêm lệnh sau vào AndroidManifest.xml:
* <uses-permission android:name="android.permission.INTERNET" />
*****/

/**
 * Lấy dữ liệu text từ Server
 *
 * @param component
 * @return String
 */
public String getTextInfo(PostComponent component) {

    int BUFFER_SIZE = 2000;
    InputStream in = null;
    String str = "";

    try {
        in = OpenHttpConnection(component);
    } catch (IOException e) {
        Log.e("Networking", e.getMessage());
        return str;
    }

    if (in == null){
        Log.e("URL", "Check connection or URL again!");
        return str;
    }

    try {
        InputStreamReader isr = new InputStreamReader(in);
        int charRead;
        char[] inputBuffer = new char[BUFFER_SIZE];
        while ((charRead = isr.read(inputBuffer)) > 0) {

            //Chuyển chars thành String
            String readString = String.valueOf(inputBuffer, 0, charRead);

```

```

        str += readString;
        inputBuffer = new char[BUFFER_SIZE];
    }
    in.close();
} catch (IOException e) {
    Log.e("Networking", e.getLocalizedMessage());
    return "";
}
return str;
}

private class GetServerResponse extends AsyncTask<PostComponent, Void,
String> {

    protected String doInBackground(PostComponent... postComponents) {
        return getTextInfo(postComponents[0]);
    }

    @Override
    protected void onPostExecute(String result) {
        TextView tv = (TextView) findViewById(R.id.tv);
        if (tv != null) {
            tv.setText(result);
        }
    }
}

/**
 * Đối tượng POST gồm
 * URL: giữ địa chỉ dẫn tới nơi lấy dữ liệu
 * JSONObject: đối số để lấy được dữ liệu
 */
private class PostComponent {

    String URL;
    JSONObject param;

    PostComponent(String URL, JSONObject param) {
        this.URL = URL;
        this.param = param;
    }

    public String getURL() {
        return URL;
    }

    public JSONObject getParam() {
        return param;
    }
}

/**
 * Chuyển đối tượng JSON (giữ 2 giá trị) về kiểu String
 *
 * @param params
 * @return String
 */

```

```

public String getPostDataString(JSONObject params) throws Exception {

    StringBuilder result = new StringBuilder();
    boolean first = true;

    Iterator<String> itr = params.keys();

    while (itr.hasNext()) {

        String key = itr.next();
        Object value = params.get(key);

        if (first)
            first = false;
        else
            result.append("&");

        result.append(URLEncoder.encode(key, "UTF-8"));
        result.append("=");
        result.append(URLEncoder.encode(value.toString(), "UTF-8"));
    }
    return result.toString();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_post_http_main);

    String URL = "http://10.0.2.2/dhminh/thongtin.php";
    JSONObject postParams = new JSONObject();
    try {
        postParams.put("mscb", "000555");
    } catch (JSONException e) {
        e.printStackTrace();
    }
    PostComponent postRequest = new PostComponent(URL, postParams);
    new GetServerResponse().execute(postRequest);
}
}

```

Ứng dụng này được chạy thử với Localhost có URL là <http://10.0.2.2/dhminh/thongtin.php> với cặp khoá giá trị là **mscb:000555**. Kết quả đáp ứng từ server như hình 6.7.

Ghi chú: Vì emulator có thể không kết nối được đến các URL ở xa, để bảo đảm có thể chạy thử được ta dùng Localhost. Các bước chuẩn bị (trước khi chạy ứng dụng) như sau:

- 1) Mở Notepad tạo ra tập tin bằng ngôn ngữ PHP và lưu vào thư mục **Buoi_6**, nếu đặt tên là “thongtin.php” thì URL như sau:

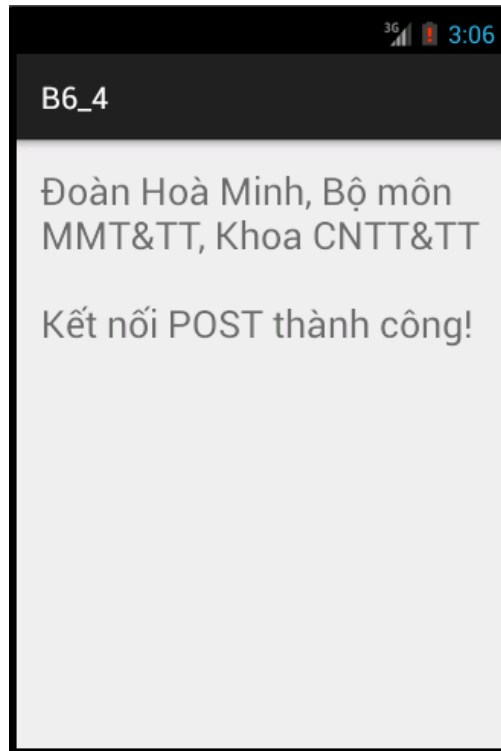
["http://10.0.2.2/Buoi_6/thongtin.php"](http://10.0.2.2/Buoi_6/thongtin.php)

(ta có thể đặt tên khác, tuy nhiên phải đổi URL trong mã nguồn cho thích hợp), mẫu như sau tập tin này như sau:


```
<?php
$mscb = $_POST['mscb'];
if($mscb == '000555'){
    echo "Đoàn Hoà Minh, Bộ môn MMT&TT, Khoa CNTT&TT ";
}
else{
    echo "Sai mã số cán bộ";
}
echo "\nKết nối POST thành công!"
?>
```

Trong đó cặp khoá/giá trị là **mscb** / **000555**, SV viên có thể đổi lại là mssv của mình và đổi nội dung hiển thị là tên SV. Ví dụ:

```
$mssv = $_POST['mssv'];
if($mssv == 'b130277377'){
    echo "Nguyễn Thu Hương, Lớp CT27401, ngành TT&MMT, Khoa CNTT&TT
";
}
```



Hình 6.7

- 2) Trên giao diện của ứng dụng này, sinh viên hãy **thêm 1 EditText** (Nhập URL) và **1 button** (Connect), để người dùng nhập URL vào EditText và ấn Button để thực hiện kết nối, gửi yêu cầu và lấy đáp ứng.