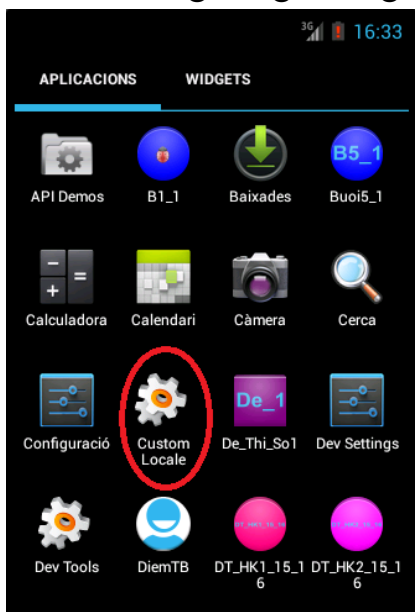


BÀI THỰC HÀNH BUỔI 5

I. LÝ THUYẾT CẦN ÔN TẬP

- Truy vấn dữ liệu với Content provider dựng sẵn (xem giáo trình/Slide bài giảng).
- Lưu trữ dữ liệu bằng **Hệ thống tập tin** truyền thống (xem giáo trình/Slide bài giảng).
- Telephony API.
- Đa ngôn ngữ trong Android



Hình 5.1

Đa ngôn ngữ trong android giúp cho ứng dụng android có thể hiển thị giao diện với nhiều ngôn ngữ khác nhau tùy thuộc vào việc người dùng lựa chọn ngôn ngữ hiển thị trên thiết bị của họ.

Android mặc định tiếng Anh là ngôn ngữ chính, tạo sẵn và nạp chuỗi từ tập tin strings trong thư mục **values** (đường dẫn: **res/values/strings.xml**). Khi ta muốn bổ sung thêm những ngôn ngữ khác, ta cần **phải tạo thư mục mới** với tên được đặt theo quy tắc là **value-mã ngôn ngữ quốc gia**. Ví dụ nếu bạn muốn thêm tiếng Việt, ta phải tạo một thư mục với tên **values-vi**, sao chép tập tin **strings.xml** vào trong thư mục này và dịch tất cả các chuỗi từ tiếng Anh sang ngôn ngữ tiếng Việt.

Sau đây là vài mã ngôn ngữ quốc gia thông dụng: Pháp - fr, Đức - de, TQ - zh, Nhật - ja, Hàn quốc - ko,

Việc thay đổi chuỗi sang ngôn ngữ khác nhau được gọi là bản địa hóa (localization). Với ứng dụng được hỗ trợ nhiều ngôn ngữ, người dùng có thể chọn ngôn ngữ như

sau:

1. Trên điện thoại (kể cả máy ảo, hình 5.1), chọn **Settings** hoặc **Custom Locale** ⇒ check chọn ngôn ngữ mà ta muốn.
2. Nếu ứng dụng hỗ trợ lựa chọn ngôn ngữ, android tìm kiếm các nguồn tài nguyên phù hợp với ngôn ngữ được chọn từ **values-(mã nguồn ngữ)**
3. Nếu các chuỗi nội địa hoá bị thiếu, android sẽ nạp các chuỗi đó từ tập tin **strings.xml** (res ⇒ values ⇒ strings.xml)

Một số lưu ý:

Nên làm

1. Luôn luôn khai báo các chuỗi trong tập tin strings.xml và sử dụng nó trong tập tin Layout.xml, thay vì khai báo trực tiếp trong tập tin Layout.

Ví dụ: Ta muốn hiển thị 1 chuỗi như “Hãy nhập vào một message”
Ta khai báo chuỗi này trong tập tin strings.xml như sau”

```
<string name="txt1">Hãy nhập vào một message</string>
```

Và sử dụng trong tập tin layout:

```
android:text="@string/txt1"
```

Không nên khai báo trực tiếp chuỗi trong Layout.xml hay trong mã java

```
android:text="Hãy nhập vào một message"
```

2. Dùng hàm setText() trong mã java khi cần

```
txtName.setText("Enter your address");
```

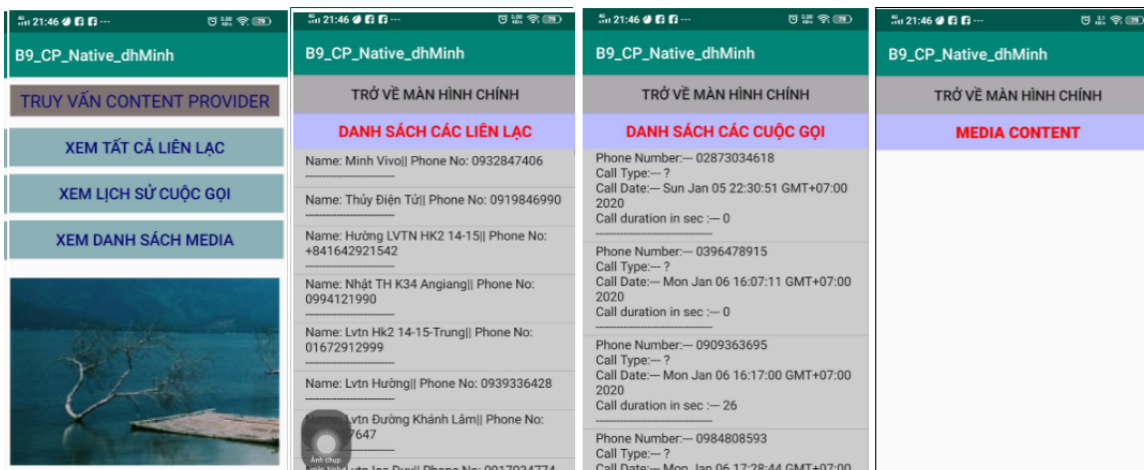
Trong đó chuỗi “Enter your address” là nội dung mới cần thay thế cho nội dung cũ.

Áp dụng vào các bài sau đây để xây dựng giao diện tiếng Anh và Việt.

II. THỰC HÀNH

BÀI 9: TRUY XUẤT DỮ LIỆU VỚI CONTENT PROVIDERS DỰNG SẴN

Hãy xây dựng một ứng dụng truy vấn và xem danh sách contact, call log và media trên một điện thoại di động. Giao diện chính như hình 5.1.a, khi người dùng muốn



Hình 5.1.a

Hình 5.1.b

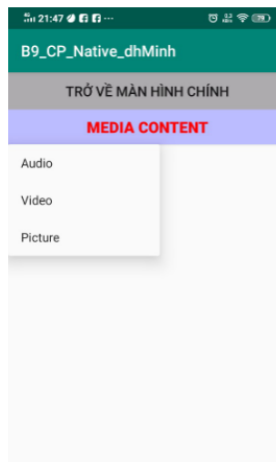
Hình 5.1.c

Hình 5.1.d

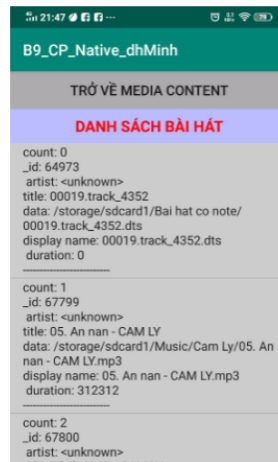
xem nội dung nào thì chọn button đó.

Khi chọn button “XEM TẤT CẢ LIÊN LẠC” thì sẽ hiển thị danh sách danh bạ như hình 5.1.b, khi chọn button “XEM TẤT LỊCH SỬ CUỘC GỌI” thì sẽ hiển thị danh sách calllog như hình 5.1.c, khi chọn button “XEM DANH SÁCH MEDIA” thì sẽ hiển thị giao diện như hình 5.1.d. Trên giao diện hình 5.1.d, khi ấn nút “MEDIA CONTENT” thì một popup menu bung xuống như hình 5.2.a, khi chọn mục Audio thì sẽ hiển thị danh sách bài hát như Hình 5.2.b, khi chọn mục Video thì sẽ hiển thị danh sách video như Hình 5.2.c, tương tự, khi chọn mục Picture thì sẽ hiển thị danh sách hình ảnh như Hình 5.2.d. Khi nhấn chọn nút “TRỞ VỀ MÀN HÌNH CHÍNH” thì sẽ trở về giao diện chính như hình 5.1.a. Khi nhấn chọn nút

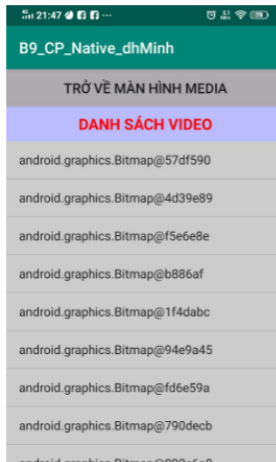
“TRỞ VỀ MÀN HÌNH MEDIA” thì sẽ trở về giao diện chọn loại media như hình 5.1.d.



Hình 5.2.a



Hình 5.2.b



Hình 5.2.c



Hình 5.2.d

CÁC BƯỚC THỰC HIỆN

1. Hãy mở project mới tên là **B9_Tên SV_MSSV**, với Activity chính là **MainActivity.java** và layout là **activity_main.xml**. Lưu ý: Chọn Minimum Required SDK là API 16.
2. Thêm các **permission** sau đây vào tập tin AndroidManifest.xml:

```
<uses-permission android:name="android.permission.READ_STUDENTS" />
<uses-permission android:name="android.permission.READ_CALL_LOG" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

3. Lập trình giao diện chính như Hình 5.1.a (không cung cấp chương trình mẫu), chạy thử để chỉnh sửa giao diện đúng theo yêu cầu: 1 TextView hiển thị chủ đề của ứng dụng “TRUY VẤN CONTENT PROVIDER”, 3 Button để người dùng click chọn nội dung muốn truy vấn “XEM TẤT CẢ LIÊN LẠC”, “XEM LỊCH SỬ CUỘC GỌI” và “XEM DANH SÁCH MEDIA”, 1 ImageView để trang trí (bước này chưa cần lập trình MainActivity).
4. Bằng cách nhấp chuột phải vào thư mục tên miền của project để lần lượt tạo ra các Activity:
 - **DisplayAllContact** có chức năng truy vấn và hiển thị danh sách tất cả liên lạc trong danh bạ.
 - **DisplayAllCallLog** có chức năng truy vấn và hiển thị lịch sử cuộc gọi trong 30 ngày gần nhất.
 - **MediaContent** thực hiện giao diện popup menu chọn lựa truy vấn và hiển thị danh sách media các loại.
 - **DisplayAudioList** truy vấn và hiển thị danh sách các tập tin âm thanh.

- **DisplayVideoList** truy vấn và hiển thị danh sách các tập tin video.
 - **DisplayPictureList** truy vấn và hiển thị danh sách các tập tin hình ảnh.
5. Lập trình MainActivity xử lý tương tác của người dùng lên các nút XEM TẤT CẢ LIÊN LẠC, XEM LỊCH SỬ CUỘC GỌI và XEM DANH SÁCH MEDIA như sau: Khi người dùng chọn nút XEM TẤT CẢ LIÊN LẠC thì sẽ chuyển đến Activity **DisplayAllContact.java** để truy vấn và hiển thị danh sách tất cả liên lạc trong danh bạ. Khi người dùng chọn nút XEM LỊCH SỬ CUỘC GỌI thì sẽ chuyển đến Activity **DisplayAllCallLog.java** để truy vấn và hiển thị danh sách các cuộc gọi trong 30 ngày gần nhất. Khi người dùng chọn nút XEM DANH SÁCH MEDIA thì sẽ chuyển đến Activity **MediaContent.java** để tạo ra một popup menu gồm 3 mục chọn lựa: Audio, Video và Picture. (Không cung cấp chương trình mẫu).
 6. Lập trình Activity **DisplayAllContact.java** (Xem Chương trình mẫu 1) với giao diện người dùng gồm button “TRỞ VỀ MÀN HÌNH CHÍNH” (trở về giao diện chính khi nút này được nhấn chọn), TextView “DANH SÁCH CÁC LIÊN LẠC” và một ListView để hiển thị danh sách các liên lạc trong danh bạ (Như Hình 5.1.b, không cung cấp chương trình mẫu). Chạy thử và chỉnh sửa (Trước khi chạy thử, chúng ta phải mở thiết bị giả lập và nhập một số liên lạc trong danh bạ).

Chương trình mẫu 1

```
package com.example.b9_cp_native_dhminh;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.content.ContentResolver;
import android.database.Cursor;
import android.provider.ContactsContract;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import java.util.ArrayList;

public class DisplayAllContact extends AppCompatActivity {
    Button back;
    ListView lv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_all_contact);
        back = (Button) findViewById(R.id.btntrove);
        lv = (ListView) findViewById(R.id.lvcontact);
        ArrayList<String> list=new ArrayList<String>();
        ContentResolver cr = getContentResolver();
        ArrayAdapter<String> adapter=new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, list);
        lv.setAdapter(adapter);
        // lay toan bo danh ba
        Cursor cur = cr.query(ContactsContract.Contacts.CONTENT_URI, null,
```

```

        null, null, null);
    if (cur.getCount() > 0) {
        while (cur.moveToNext()) {
            // get id va name của từng người trong danh bạ
            String id = cur.getString(cur
                .getColumnIndex(ContactsContract.Contacts._ID));
            String name = cur
                .getString(cur
                    .getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));
            if (Integer
                .parseInt(cur.getString(cur
                    .getColumnIndex(ContactsContract.Contacts.HAS_PHONE_NUMBER))) > 0) {
                // lấy tất cả các số của người có id = id
                Cursor pCur = cr.query(
                    ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
                    null,
                    ContactsContract.CommonDataKinds.Phone.CONTACT_ID
                        + " = ?", new String[] { id }, null);
                while (pCur.moveToNext()) {
                    String phoneNo = pCur
                        .getString(pCur
                            .getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER));
                    // log toàn bộ số điện thoại ra logcat
                    String contact = "Name: " + name
                        + "|| Phone No: " + phoneNo
                        + "\n-----";
                    list.add(contact);
                    adapter.notifyDataSetChanged();
                }
                pCur.close();
            }
        }
    }
    cur.close();

    back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });
}
}

```

7. Lập trình Activity **DisplayAllCallLog.java** (Xem Chương trình mẫu 2) với giao diện người dùng gồm button “TRỞ VỀ MÀN HÌNH CHÍNH” (trở về giao diện chính khi nút này được nhấn chọn), TextView “DANH SÁCH CÁC CUỘC GỌI” và một ListView để hiển thị danh sách các cuộc gọi trong 30 ngày gần nhất (Như Hình 5.1.c, không cung cấp chương trình mẫu). Chạy thử và chỉnh sửa. (Trước khi chạy thử, chúng ta phải mở thiết bị giả lập và thực hiện vài cuộc gọi để có lịch sử cuộc gọi).

Chương trình mẫu 2

```

package com.example.b9_cp_native_dhminh;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.os.Bundle;
import android.annotation.SuppressLint;
import android.content.ContentResolver;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.provider.CallLog;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import java.util.ArrayList;
import java.util.Date;

public class DisplayAllCallLog extends AppCompatActivity {
    Button back2;
    ListView lvcallLog;

    @SuppressLint("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_all_call_log);
        back2 = (Button) findViewById(R.id.btntrove2);
        lvcallLog = (ListView) findViewById(R.id.lvcallLog);

        ArrayList<String> list2 = new ArrayList<String>();
        ContentResolver cr2 = getContentResolver();
        ArrayAdapter<String> adapter2 = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, list2);
        lvcallLog.setAdapter(adapter2);

        // Lấy lịch sử cuộc gọi
        String[] projection = new String[]{CallLog.Calls.DATE,
        CallLog.Calls.NUMBER,
        CallLog.Calls.DURATION};

        if (checkSelfPermission(Manifest.permission.READ_CALL_LOG) !=
        PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            // Activity#requestPermissions

            return;
        }
        Cursor managedCursor =
            getContentResolver().query(CallLog.Calls.CONTENT_URI, projection,
            CallLog.Calls.DURATION + "<?", new String[]{"30"},
            CallLog.Calls.DATE + " Asc"); // Asc: sắp theo thứ tự tăng dần

        int number = managedCursor.getColumnIndex(CallLog.Calls.NUMBER);
        int type = managedCursor.getColumnIndex(CallLog.Calls.TYPE);
        int date = managedCursor.getColumnIndex(CallLog.Calls.DATE);
        int duration = managedCursor.getColumnIndex(CallLog.Calls.DURATION);
        //Log.d("Call_Logs", "Call Details :");
        while (managedCursor.moveToNext()) {
            // lay cac cot trong cursor
            String phoneNumber = managedCursor.getString(number);
            //String callType = managedCursor.getString(type);
            String callDate = managedCursor.getString(date);

```

```

        Date callDayTime = new Date(Long.valueOf(callDate));
        String callDuration = managedCursor.getString(duration);

        // hiển thị toàn bộ lịch sử cuộc gọi
        String cuocgoi = "Phone Number:--- " + phNumber
            + " \nCall Type:--- " + "?" + " \nCall Date:--- "
            + callDayTime + " \nCall duration in sec :--- "
            + callDuration + "\n-----";

        list2.add(cuocgoi);
        adapter2.notifyDataSetChanged();
    }
    managedCursor.close();

    back2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });
}
}

```

8. Lập trình Activity **MediaContent.java** (Xem Chương trình mẫu 3) với giao diện người dùng gồm button “TRỞ VỀ MÀN HÌNH CHÍNH” (trở về giao diện chính khi nút này được nhấn chọn), Button “MEDIA CONTENT” để sinh ra một popup menu (Như Hình 5.1.d, không cung cấp chương trình mẫu, xem lại Bài 2_Buổi 1 để tạo popup menu 3 mục chọn lựa: Audio, Video và Picture như hình 5.2.a). Chạy thử và chỉnh sửa.

Chương trình mẫu 3

```

package com.example.b9_cp_native_dhminh;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.PopupMenu;

public class MediaContent extends AppCompatActivity {
    Button popupmenu, back5;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_media_content);
        popupmenu = (Button) findViewById(R.id.mediaMenu);
        back5 = (Button) findViewById(R.id.btntrove5);
        back5.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
    }
}

```



```

        popupmenu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showPopupMenu();
            }
        });
    }
    // Tạo Popup menu
    private void showPopupMenu() {
        // Tạo 1 popupmenu.
        PopupMenu popupMenu=new PopupMenu(this,popupmenu);
        //Lấy giao diện menu trong resouce.

        popupMenu.getMenuInflater().inflate(R.menu.menu_popup,popupMenu.getMenu());
        //Lập trình hành động khi chọn item.

        popupMenu.setOnMenuItemClickListener(new
        PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.audio:
                        Intent intent1 = new Intent(MediaContent.this,
                                DisplayMediaList.class);
                        startActivity(intent1);
                        break;
                }
                switch (item.getItemId()) {
                    case R.id.video:
                        Intent intent2 = new Intent(MediaContent.this,
                                DisplayVideoList.class);
                        startActivity(intent2);
                        break;
                }
                switch (item.getItemId()) {
                    case R.id.picture:
                        Intent intent3 = new Intent(MediaContent.this,
                                DisplayPictureList.class);
                        startActivity(intent3);
                        break;
                }
                return false;
            }
        });

        popupMenu.show();
    }
}

```

9. Lập trình Activity **DisPLayAudioList.java** (Xem Chương trình mẫu 4) với giao diện người dùng gồm button “TRỞ VỀ MEDIA CONTENT” (trở về giao diện Hình 5.1.d khi nút này được nhấn chọn), một TextView “DANH SÁCH BÀI HÁT” và một ListView để hiển thị danh sách các bài hát, như Hình 5.2.b (xem chương trình mẫu 5). Chạy thử và chỉnh sửa (Trước khi chạy thử, chúng ta phải mở thiết bị giả lập và tải vào thư mục sdcard/Music hoặc sdcard/DCIM một số file âm thanh/bài hát mp3, chú ý rằng, tên của các file âm thanh/bài hát không có khoảng cách).

Chương trình mẫu 4


```

package com.example.b9_cp_native_dhminh;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.database.Cursor;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import java.util.ArrayList;

public class DisplayAudioList extends AppCompatActivity {
    Button backaudio;
    ListView lvaudio;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_audio_list);
        backaudio = (Button) findViewById(R.id.btntrove3);
        lvaudio = (ListView) findViewById(R.id.lvmedia);

        ArrayList<String> listaudio = new ArrayList<String>();
        ArrayAdapter<String> adapteraudio = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, listaudio);
        lvaudio.setAdapter(adapteraudio);

        // lay cac cot nhu trong bang projection

        String[] projection = new String[]{ MediaStore.Audio.Media._ID,
            MediaStore.Audio.Media.ARTIST, MediaStore.Audio.Media.TITLE,
            MediaStore.Audio.Media.DATA,
            MediaStore.Audio.Media.DISPLAY_NAME,
            MediaStore.Audio.Media.DURATION };

        Cursor cursor =
            getApplicationContext().getContentResolver().query(MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
                projection, null, null,
                "LOWER(" + MediaStore.Audio.Media.TITLE + ") ASC");
        int count = 0;
        cursor.moveToFirst();
        while (cursor.isAfterLast() == false) {
            String ds =
                "count: " + count
                + "\n_id: " + cursor.getString(0)
                + "\n_artist: " + cursor.getString(1)
                + "\n_title: " + cursor.getString(2)
                + "\ndata: " + cursor.getString(3)
                + "\ndisplay name: " + cursor.getString(4)
                + "\nduration: " + cursor.getString(5)
                + "\n-----";
            listaudio.add(ds);
            adapteraudio.notifyDataSetChanged();

            // log toan bo danh sach bai hat ra logcat
            Log.d("List Music",
                "\ncount: " + count + "\n_id: " + cursor.getString(0)
                + "\n_artist: " + cursor.getString(1) + "\n_title: "
                + cursor.getString(2) + "\ndata: "
                + cursor.getString(3) + "\ndisplay name: "

```

```

        + cursor.getString(4) + "\n duration: "
        + cursor.getString(5)

        + "\n-----");
    cursor.moveToNext();
    count++;
}
cursor.close();

backaudio.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
}
}

```

Chương trình mẫu 5

```

<?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=".DisplayAudioList">
    <Button
        android:id="@+id/btntrove3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:background="#ADABAD"
        android:textSize="18sp"
        android:text="TRỞ VỀ MEDIA CONTENT" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:background="#bbbbbf"
        android:gravity="center_vertical|center_horizontal"
        android:text="DANH SÁCH AUDIO"
        android:textAllCaps="true"
        android:textColor="#ff0000"
        android:textSize="20sp"
        android:textStyle="bold" />

    <ListView
        android:id="@+id/lvaudio"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#cccccc">
    </ListView>
</LinearLayout>

```

10. Lập trình Activity **DisplayVideoList.java** (Xem Chương trình mẫu 6) với giao diện người dùng gồm button “TRỞ VỀ MEDIA CONTENT” (trở về giao

diện Hình 5.1.d khi nút này được nhấn chọn), một TextView “DANH SÁCH VIDEO” và một ListView để hiển thị danh sách Thumbnail các video, như Hình 5.2.c (tham khảo chương trình mẫu 5 và chỉnh sửa). Chạy thử và chỉnh sửa (Trước khi chạy thử, chúng ta phải mở thiết bị giả lập và tải vào thư mục sdcard/Movies hoặc sdcard/DCIM một số file video/bài hát mp4, chú ý rằng, tên của các video/bài hát không có khoảng cách).

Chương trình mẫu 6

```
package com.example.b9_cp_native_dhminh;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentUri;
import android.graphics.Bitmap;
import android.media.MediaMetadataRetriever;
import android.net.Uri;
import android.os.Bundle;
import android.content.ContentResolver;
import android.database.Cursor;
import android.provider.BaseColumns;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.concurrent.TimeUnit;

public class DisplayVideoList extends AppCompatActivity {
    Button back4;
    ListView video;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_video_list);
        back4 = (Button) findViewById(R.id.btntrove4);
        video = (ListView) findViewById(R.id.lvvideo);

        ArrayList<Bitmap> list4 = new ArrayList<Bitmap>();
        ArrayAdapter<Bitmap> adapter4 = new ArrayAdapter<Bitmap>(this,
            android.R.layout.simple_list_item_1, list4);
        video.setAdapter(adapter4);

        ContentResolver cr = getContentResolver();
        String[] proj = {BaseColumns._ID};
        Cursor c = cr.query(MediaStore.Video.Media.EXTERNAL_CONTENT_URI, proj,
null, null, null);
        if (c.moveToFirst()) {
            do
            {
                int id = c.getInt(0);
                Bitmap b = MediaStore.Video.Thumbnails.getThumbnail(cr, id,
MediaStore.Video.Thumbnails.MINI_KIND, null);
                Log.d("*****My Thumbnail*****", "onCreate bitmap " + b);
```

```

        list4.add(b);
        adapter4.notifyDataSetChanged();
    }
    while( c.moveToNext() );
}
c.close();
back4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
}

public static Bitmap retrieveVideoFrameFromVideo(String videoPath) throws
Throwable {
    Bitmap bitmap = null;
    MediaMetadataRetriever mediaMetadataRetriever = null;
    try {
        mediaMetadataRetriever = new MediaMetadataRetriever();
        mediaMetadataRetriever.setDataSource(videoPath, new HashMap<String,
String>());
        // mediaMetadataRetriever.setDataSource(videoPath);
        bitmap = mediaMetadataRetriever.getFrameAtTime();
    } catch (Exception e) {
        e.printStackTrace();
        throw new Throwable("Exception in retrieveVideoFrameFromVideo(String
videoPath)" + e.getMessage());
    } finally {
        if (mediaMetadataRetriever != null) {
            mediaMetadataRetriever.release();
        }
    }
    return bitmap;
}
}

```

- 11. Lập trình Activity `DisplayPictureList.java`** (Tham khảo Chương trình mẫu 7) với giao diện người dùng gồm button “TRỞ VỀ MEDIA CONTENT” (trở về giao diện Hình 5.1.d khi nút này được nhấn chọn), một TextView “DANH SÁCH HÌNH” và một ListView để hiển thị danh sách các hình (tham khảo chương trình mẫu 6 và chỉnh sửa). Chạy thử và chỉnh sửa (Trước khi chạy thử, chúng ta phải mở thiết bị giả lập và tải vào thư mục sdcard/Pictures hoặc sdcard/DCIM một số file ảnh, chú ý rằng, tên của các file ảnh không có khoảng cách).

Chương trình mẫu 7

```

package com.example.b9_cp_native_dhminh;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.database.Cursor;
import android.provider.MediaStore;
import android.util.Log;

```

```
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import java.util.ArrayList;

public class DisplayPictureList extends AppCompatActivity {
    Button backfrompicture;
    ListView lvpicture;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_picture_list);
        backfrompicture = (Button) findViewById(R.id.btntrove3);
        lvpicture = (ListView) findViewById(R.id.lvmedia);
        backfrompicture.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
        ArrayList<String> listImage = new ArrayList<String>();
        ArrayAdapter<String> adapterImage = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, listImage);
        lvpicture.setAdapter(adapterImage);
        // get toan bo danh sach hinh anh
        String[] projection = new String[]{
            MediaStore.Images.Media._ID,
            MediaStore.Images.Media.TITLE,
            MediaStore.Images.Media.DATE_ADDED
        };
        Cursor cursor =
        getApplicationContext().getContentResolver().query(MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
            projection, null, null, null);
        int count = 0;
        if (cursor.moveToFirst()) {
            do {
                String ds =
                    "count: " + count
                    + "\n_id: " + cursor.getString(0)
                    + "\ntitle: " + cursor.getString(1)
                    + "\ndata_added: " + cursor.getString(2)
                    + "\n-----";
                listImage.add(ds);
                adapterImage.notifyDataSetChanged();
                count++;
            } while (cursor.moveToNext());
        }
        cursor.close();
    }
}
```

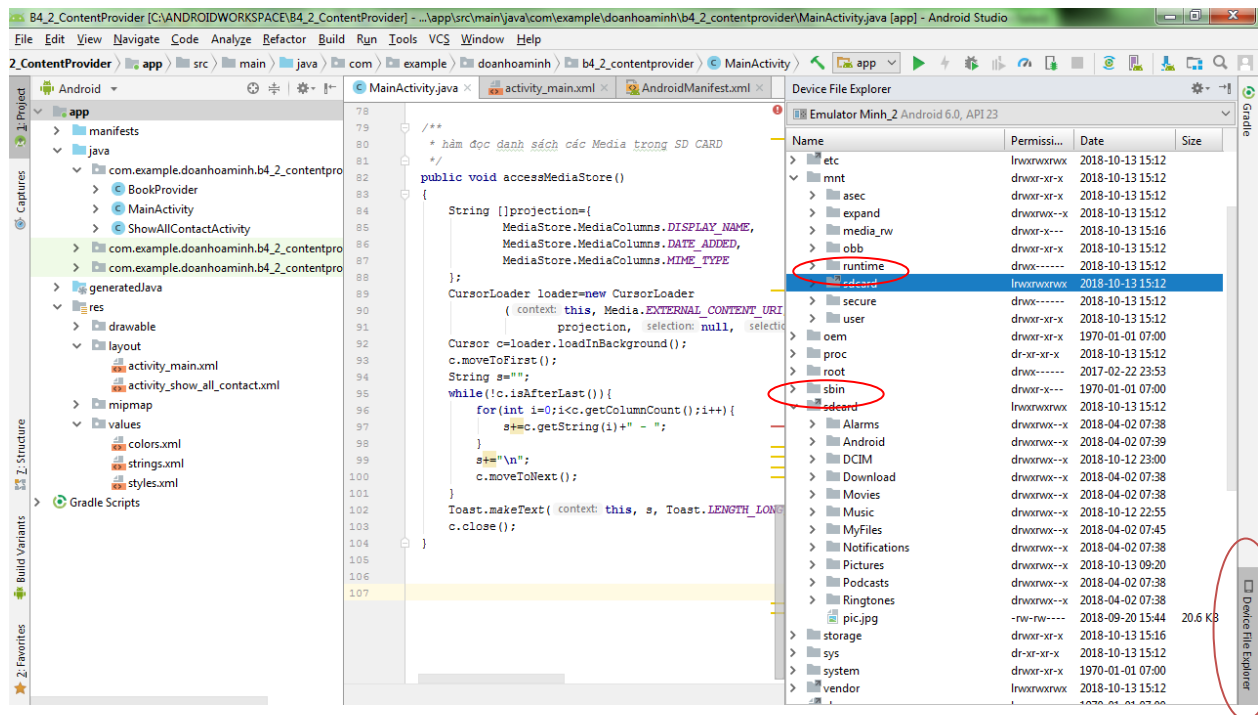
Ghi chú:

1. Với phiên bản cao hơn API 22, Android 5.0, Android không còn hỗ trợ Browser Bookmark, lớp **BookmarkColumns** bị từ chối (deprecate). Muốn truy xuất dữ liệu Bookmark ta phải lập trình tạo ra dữ liệu này. Chúng ta không thực hành Browser Bookmark.
2. **Cách thêm thư mục, tập tin vào SDCard:**

Để chạy thử với Andoid emulator ta phải tạo danh bạ trên emulator, thêm các tập tin âm thanh, hình ảnh, video vào bộ nhớ của emulator. Dùng emulator thực hiện một số cuộc gọi (tạo Call Logs) . Tham khảo hướng dẫn sau đây:

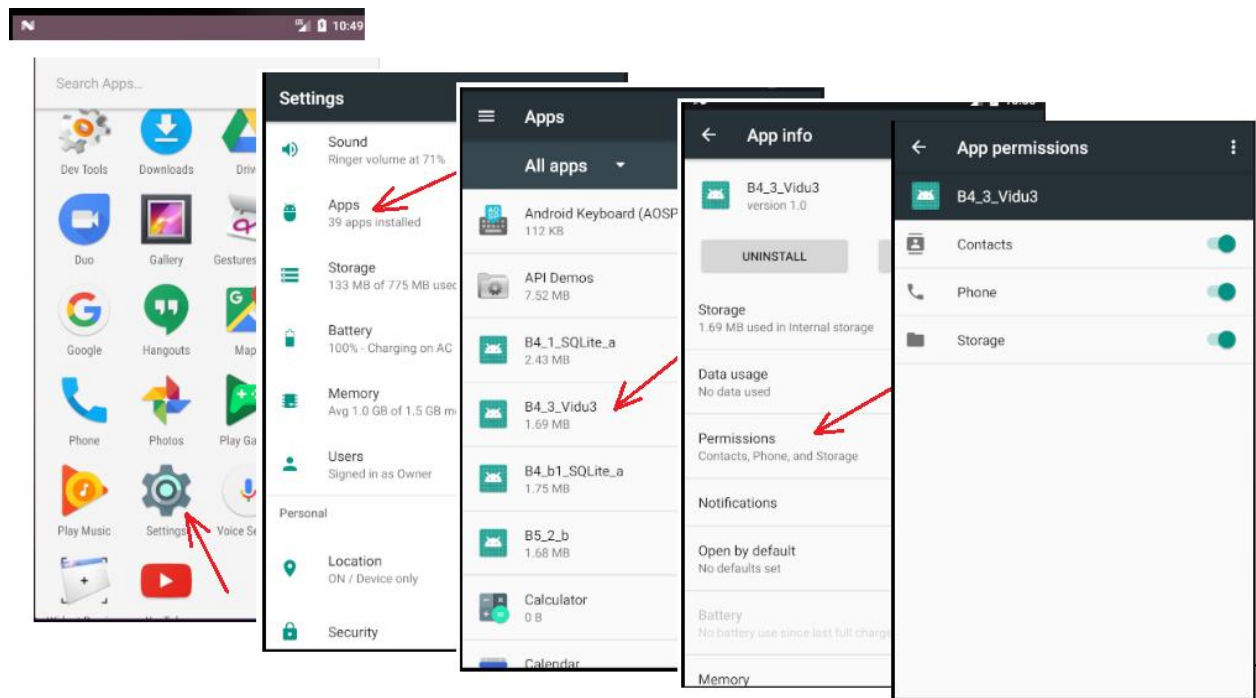
Trên Android studio, nhấp chọn “Device File Explorer” ở góc phải phía dưới của màn hình, cửa sổ “Device File Explorer” hiện ra (Hình 5.3). Ta thấy thư mục SDCard nằm ở 2 vị trí, trong thư mục **mnt** và ở ngang hàng với mnt. Trong đó có nhiều thư mục như Alarm, Android, DCIM, Download, Movies, Notifications, Pictures, Poscasts Ringtones,...

- ➔ Chúng ta muốn thêm thư mục con, tập tin vào thư mục nào thì click chuột phải vào thư mục đó và chọn mục thích hợp.
- ➔ Muốn chép âm nhạc, hình ảnh, video từ máy tính PC thì nhấp chuột phải lên thư mục tương ứng trên sdcard → chọn Upload và chọn đường dẫn đến tập tin trên máy tính mà ta muốn chép vào *hoặc có thể tải từ trên mạng về bằng trình duyệt trên máy ảo*.
- ➔ Để thêm liên lạc và danh bạ của máy ảo thì thao tác trên máy ảo (giống như làm trên máy thật).



Hình 5.3

3. **Xin phép (Permission) trên thiết bị:** truy xuất vào Danh bạ, Phone và Storage. Các bước thực hiện như Hình 5.4: Chọn Settings → Chọn Apps → Chọn ứng dụng muốn chạy thử → Chọn Permissions → Check chọn tất cả (Students, Phone, Storage).



Hình 5.4

Tham Khảo: <https://developer.android.com/guide/topics/providers/content-provider-basics>

4. Giải thích thêm

Để truy xuất nội dung tóm tắt của các tập tin media được lưu dưới dạng Content provider trên bộ nhớ ngoài, ta sử dụng đối tượng **ContentResolver** với biến context của ứng dụng, Java code như sau:

```
String[] projection = new String[] {  
    media-database-columns-to-retrieve  
};  
String selection = sql-where-clause-with-placeholder-variables;  
String[] selectionArgs = new String[] {  
    values-of-placeholder-variables  
};  
String sortOrder = sql-order-by-clause;  
  
Cursor cursor = getApplicationContext().getContentResolver().query(  
    MediaStore.media-type.Media.EXTERNAL_CONTENT_URI,  
    projection,  
    selection,  
    selectionArgs,  
    sortOrder  
);  
  
while (cursor.moveToNext()) {  
    // Use an ID column from the projection to get  
    // a URI representing the media item itself.}
```



```
}
```

Hệ thống tự động quét thư mục tương ứng và thêm các tập tin media vào các bộ sưu tập như sau:

- **Images**, bao gồm các ảnh được lưu trong các thư mục **DCIM/** và **Pictures/**. Hệ thống sẽ thêm các tập này vào bảng **MediaStore.Images**.
- **Videos**, bao gồm các video được lưu trong các thư mục **DCIM/**, **Movies/**, và **Pictures/**. Hệ thống sẽ thêm các tập này vào bảng **MediaStore.Video**.
- **Audio files**, bao gồm các tập Âm thanh được lưu trong các thư mục **Alarms/**, **Audiobooks/**, **Music/**, **Notifications/**, **Podcasts/**, và **Ringtones/**, cũng như các bài hát được lưu trong các thư mục **Music/** hay **Movies/**. Hệ thống sẽ thêm các tập này vào bảng **MediaStore.Audio**.
- **Downloaded files**, đây nó được lưu trong thư mục **Download/**. Trên các thiết bị chạy Android 10 (API level 29) về sau, các tập này được lưu trong bảng **MediaStore.Downloads**. Bảng này không có trong phiên bản *Android 9* (API level 28) về trước.

BÀI 10: LƯU TRỮ TẬP TIN TRÊN BỘ NHỚ VÀ SỬ DỤNG TÀI NGUYÊN TĨNH

Các bước thực hiện như sau:

- (1) Mở Android project, trong ví dụ này project được đặt tên là **B10_Tên SV_MSSV**.
- (2) Sửa lại tập tin **Layout.xml** để có giao diện tiếng Anh và tiếng Việt như hình 5.5.a và 5.5.b. (Tham khảo chương trình mẫu 8).

Chương trình mẫu 8

```
<?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">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#ff0000"
        android:textSize="24sp"
        android:textAllCaps="true"
        android:layout_marginTop="10dp"
        android:padding="5dp"
        android:layout_gravity="center"
        android:gravity="center"
        android:background="#ffffaa"
        android:text="@string/txt0">
```

```
        android:textStyle="bold"/>

<EditText
    android:id="@+id/txtText1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:background="#fbf"
    android:layout_gravity="center"
    android:layout_marginTop="10dp"
    android:padding="5dp"
    android:hint="@string/edittext1"/>

<Button
    android:id="@+id/btnSave1"
    android:text="@string/btn1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:layout_marginTop="10dp"
    android:padding="5dp"
    android:layout_gravity="center"
    android:background="#bff"
/>

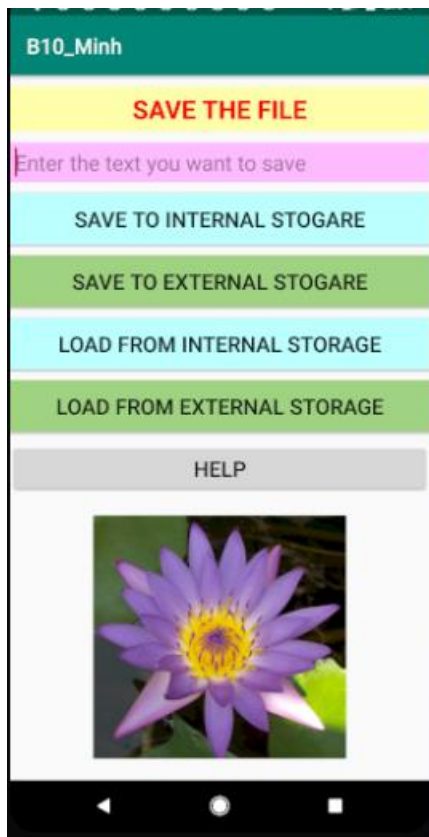
<Button
    android:id="@+id/btnSave2"
    android:text="@string/btn3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:layout_marginTop="10dp"
    android:padding="5dp"
    android:background="#A0D381"
    android:layout_gravity="center"
/>

<Button
    android:id="@+id/btnLoad1"
    android:text="@string/btn2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:layout_marginTop="10dp"
    android:padding="5dp"
    android:background="#bff"
    android:layout_gravity="center"
/>

<Button
    android:id="@+id/btnLoad2"
    android:text="@string/btn4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:layout_marginTop="10dp"
    android:padding="5dp"
    android:background="#A0D381"
    android:layout_gravity="center"
/>

<Button
    android:id="@+id/help"
    android:text="@string/btn6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:layout_marginTop="10dp"
```

```
        android:padding="5dp"
        android:layout_gravity="center"
    />
    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent" >
        <ImageView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:paddingBottom="20dp"
            android:paddingTop="20dp"
            android:src="@drawable/hoasung" />
    </FrameLayout>
</LinearLayout>
```



Hình 5.5.a



Hình 5.5.b

Tập tin strings.xml

```
<resources>
    <string name="app_name">B10_Minh</string>
    <string name="txt0">Save the file</string>
    <string name="edittext1">Enter the text you want to save</string>
    <string name="btn1">SAVE TO INTERNAL STOGARE</string>
    <string name="btn3">SAVE TO EXTERNAL STOGARE</string>
    <string name="btn2">LOAD FROM INTERNAL STORAGE</string>
    <string name="btn4">LOAD FROM EXTERNAL STORAGE</string>
</resources>
```

```
<string name="btn6">HELP</string>
<string name="txtstatic1">STATIC RESOURCES</string>
<string name="txtstatic2">HELP</string>
</resources>
```

- Chuyển sang kiểu hiển thị **project**. Nhấp chuột phải vào thư mục **src** → Chọn **New** → Chọn **Android Resource Directory** → Cửa sổ **New Resource Directory** hiện ra → Đặt tên Directory là **values-vi** → Nhấp OK.
- Chuyển cây thư mục sang chế độ project (để thấy thư mục values-vi). Copy tập tin **strings.xml** trong thư mục **values** dán vào thư mục **values-vi** → Đổi các text tiếng Anh thành tiếng Việt.

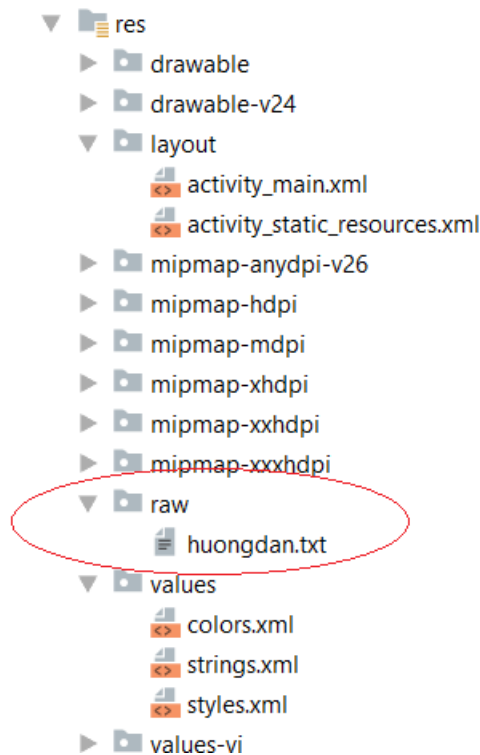
Tập tin string.xml trong thư mục values-vi như sau:

```
<resources>
  <string name="app_name">B10_Minh</string>
  <string name="txt0">Lưu tập tin</string>
  <string name="edittext1">Nhập vào văn bản muốn lưu</string>
  <string name="btn1">LƯU VÀO BỘ NHỚ TRONG</string>
  <string name="btn3">LƯU VÀO BỘ NHỚ NGOÀI</string>
  <string name="btn2">TẢI TỪ BỘ NHỚ TRONG</string>
  <string name="btn4">TẢI TỪ BỘ NHỚ NGOÀI</string>
  <string name="btn6">HƯỚNG DẪN</string>
  <string name="txtstatic1">TÀI NGUYÊN TĨNH</string>
  <string name="txtstatic2">HƯỚNG DẪN</string>
</resources>
```

- Trên máy ảo Android, kích hoạt ứng dụng Custom Locale và chọn vi-Tiếng Việt.

(3) Tạo ra Activity xử lý tài nguyên tĩnh:

Bên cạnh việc sinh ra tập tin trong thời gian chạy, ta cũng cần tạo ra các tập tin trong thời gian phát triển ứng dụng, sau đó sẽ được dùng trong thời gian chạy. Ví dụ, ta muốn tạo ra tập tin trợ giúp (Help) cho ứng dụng. Trong trường hợp này, ta thêm tập tin help vào thư mục res/raw. Hình 5.6 trình bày thư mục res/raw chứa tập tin có tên là huongdan.txt đã được tạo ra khi lập trình.



Hình 5.6: Thư mục res/raw chứa tập tin textfile.

Để dễ xây dựng thêm một tài nguyên tĩnh. Các bước thực hiện như sau:

- Nhấp chuột phải vào thư mục res, tạo 1 thư mục mới tên là **raw** trong thư mục **res**.
- Mở Notepad để tạo ra 1 tập tin văn bản và lưu vào thư mục res/raw với tên là **huongdan** (không có đuôi).
- Mở một Activity mới có tên là **StaticResouces** (Tham khảo chương trình mẫu 9) và lập trình giao diện của nó. Trong activity này, ta sử dụng phương thức getResources() để trả về một đối tượng resources InputStream và dùng phương thức openRawResource() để mở tập tin chứa trong thư mục res/raw, dùng lớp InputStreamReader để tạo ra một luồng dữ liệu đọc, dùng lớp BufferedReader để tạo ra bộ đệm chứa dữ liệu đọc, dùng thành phần TextView để hiển thị văn bản trong giao diện của nó như hình 5.7 (Không cung cấp chương trình mẫu).

Chương trình mẫu 9

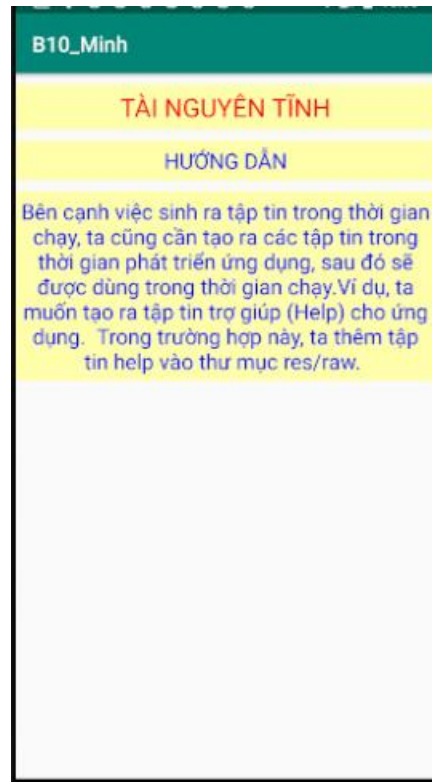
```
package com.example.b10_minh;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class StaticResources extends AppCompatActivity {
    TextView textBoxdh;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_static_resources);
        textBoxdh = (TextView) findViewById(R.id.edithd);
        // Load tập tin văn bản trong thư mục raw
        InputStream is =
this.getResources().openRawResource(R.raw.huongdan);
        InputStreamReader ir=new InputStreamReader(is);
        BufferedReader br = new BufferedReader(ir);
        String str = "";
        String buffer;
        try {
            if ((buffer=br.readLine()) != null) {
                str+= buffer;
            }
            is.close();
            br.close();
            textBoxdh.setText(str);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



Hình 5.7

- (4) **Lập trình MainActivity.java:** xử lý tương tác của người dùng như sau: Lấy văn bản đã nhập vào EditText và biến đổi thành String, tạo ra 4 phương thức xử lý tương tác của người dùng lên 4 button LƯU VÀO BỘ NHỚ TRONG, LƯU VÀO BỘ NHỚ NGOÀI, TẢI TỪ BỘ NHỚ TRONG, TẢI TỪ BỘ NHỚ NGOÀI (các phương thức này nằm ngoài phương thức onCreate(), trong project này chúng có tên lần lượt là onClickSaveInternal, onClickLoadInternal, onClickSaveExternal và onClickLoadExternal), xử lý tương tác của người dùng lên nút HƯỚNG DẪN (Tham khảo chương trình mẫu 10).

Chương trình mẫu 10

```
package com.example.b10_minh;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```



```

public class MainActivity extends AppCompatActivity {
    EditText textBox; Button SaveExt, LoadExt, help;
    static final int READ_BLOCK_SIZE = 100;
    /** Called when the activity is first created. */

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textBox = (EditText) findViewById(R.id.txtText1);
        SaveExt = (Button) findViewById(R.id.btnSave2);
        LoadExt = (Button) findViewById(R.id.btnLoad2);
        help = (Button) findViewById(R.id.help);
        help.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent tnt= new Intent(MainActivity.this,StaticResources.class);
                startActivity(tnt);
            }
        });
    }

    public void onClickSaveInternal(View view) {
        String str = textBox.getText().toString();
        try {

            FileOutputStream fOut = openFileOutput("textfile.txt", MODE_APPEND);
            OutputStreamWriter osw = new OutputStreamWriter(fOut);
            ///---write the string to the file---
            osw.write(str);
            osw.flush();
            osw.close();

            ///---display file saved message---
            Toast.makeText(getBaseContext(), "File saved successfully!",
                Toast.LENGTH_SHORT).show();

            ///---clears the EditText---
            textBox.setText(" ");
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
    }

    public void onClickLoadInternal(View view) {
        try {
            ///-----
            FileInputStream fIn = openFileInput("textfile.txt");
            InputStreamReader isr = new InputStreamReader(fIn);
            char[] inputBuffer = new char[READ_BLOCK_SIZE];
            String s = "";
            int charRead;
            while ((charRead = isr.read(inputBuffer)) > 0) {
                ///---convert the chars to a String---
                String readString =
                    String.valueOf(inputBuffer, 0, charRead);
                s += readString;
                inputBuffer = new char[READ_BLOCK_SIZE];
            }
            ///---set the EditText to the text that has been
            /// read---
            textBox.setText(s);
            Toast.makeText(getBaseContext(),

```

```

        "File loaded successfully!", Toast.LENGTH_SHORT).show();
    } catch (IOException ioe) {
        ioe.printStackTrace();
    }
}

public void onClickSaveExternal (View view) {
    String str = textBox.getText().toString();
    try
    {
        ///---Luu tren SD Card---
        File sdCard = Environment.getExternalStorageDirectory();
        File directory =
            new File (sdCard.getAbsolutePath() + "/MyFiles");
        directory.mkdirs();
        File file = new File(directory, "textfile.txt");
        FileOutputStream fOut = new FileOutputStream(file);
        OutputStreamWriter osw = new OutputStreamWriter(fOut);
        ///---write the string to the file---
        osw.write(str);
        osw.flush();
        osw.close();
        ///---display file saved message---
        Toast.makeText(getApplicationContext(), "File saved successfully!",
            Toast.LENGTH_SHORT).show();
        ///---clears the EditText---
        textBox.setText(" ");
    }
    catch (IOException ioe)
    {
        ioe.printStackTrace();
    }
}

public void onClickLoadExternal(View view){
    try {
        ///--- Đọc file lưu trên SD---
        File sdCard = Environment.getExternalStorageDirectory();
        File directory =
            new File (sdCard.getAbsolutePath()+"/MyFiles");
        File file = new File(directory, "textfile.txt");
        FileInputStream fIn = new FileInputStream(file);
        InputStreamReader isr = new InputStreamReader(fIn);

        char[] inputBuffer = new char[READ_BLOCK_SIZE];
        String s = "";
        int charRead;
        while ((charRead = isr.read(inputBuffer)) > 0) {
            ///---convert the chars to a String---
            String readString =
                String.valueOf(inputBuffer, 0, charRead);
            s += readString;
            inputBuffer = new char[READ_BLOCK_SIZE];
        }
        ///---set the EditText to the text that has been
        // read---
        textBox.setText(s);
        Toast.makeText(getApplicationContext(),
            "File loaded successfully!", Toast.LENGTH_SHORT).show();
    } catch (IOException ioe) {
        ioe.printStackTrace();
    }
}

```

```
}
```

Lưu ý: Sinh viên cần xem lại lý thuyết để hiểu các câu lệnh trong mã nguồn.

(5) **Bổ sung các thuộc tính onClick vào các button LƯU VÀO BỘ NHỚ TRONG, LƯU VÀO BỘ NHỚ NGOÀI, TẢI TỪ BỘ NHỚ TRONG, TẢI TỪ BỘ NHỚ NGOÀI** trong tập tin Layout chính. Các thuộc tính sau:

```
android:onClick="onClickSaveInternal"  
android:onClick="onClickSaveExternal"  
android:onClick="onClickLoadInternal"  
android:onClick="onClickLoadExternal"
```

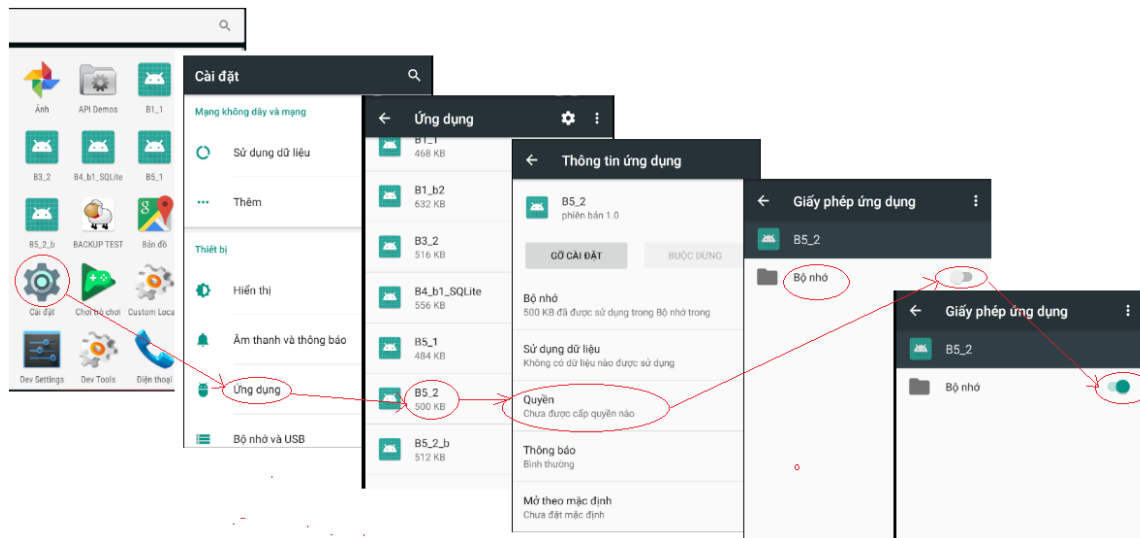
(6) **Chạy thử ứng dụng:**

- Nhập văn bản vào EditText và nhấn nút LƯU VÀO BỘ NHỚ TRONG. Nếu tập tin được lưu thành công thì sẽ xuất hiện một Toast là “File save successfully” và văn bản trong EditText bị xóa.
- Nhấn nút TẢI TỪ BỘ NHỚ TRONG ta sẽ thấy văn bản xuất hiện lại trên EditText. Việc này cho thấy văn bản đã được lưu.
- Kiểm tra tập tin đã lưu trong bộ nhớ trong: Trên Android Studio, click chọn thẻ Device File Explorer (ở cạnh phải của màn hình), cửa sổ Device File Explorer được mở ra, theo đường dẫn data/data/Tên miền của ứng dụng/files/T\textfile.txt, để kiểm tra, ta click đôi vào tên tập tin để mở tập tin ra xem.
- Nhập văn bản mới vào EditText (nên khác với nội dung lưu vào bộ nhớ trong) và nhấn nút LƯU VÀO BỘ NHỚ NGOÀI. Nếu tập tin được lưu thành công thì sẽ xuất hiện một Toast là “File save successfully” và văn bản trong EditText bị xóa.
- Nhấn nút TẢI TỪ BỘ NHỚ NGOÀI ta sẽ thấy văn bản xuất hiện lại trên EditText. Việc này cho thấy văn bản đã được lưu.
- Kiểm tra tập tin đã lưu trong bộ nhớ ngoài: Trên Android Studio, click chọn thẻ Device File Explorer (ở cạnh phải của màn hình), cửa sổ Device File Explorer được mở ra, theo đường dẫn sdcard/Myfiles/textfile.txt, để kiểm tra, ta click đôi vào tên tập tin để mở tập tin ra xem.
- Ghi chú: Nếu không thấy thẻ Device File Explorer (ở cạnh phải của màn hình), ta chọn View (trên Navigation bar) → Chọn Tools Windows → Chọn Device File Explorer.

Lưu ý:

- Có thể emulator không có phần mềm gõ tiếng Việt, nên trong hình 5.3, các chữ nhập vào EditText không bỏ dấu được.
- Nếu không lưu được tập tin vào thẻ nhớ (sdCard) trên Android emulator, ta kiểm tra lại lệnh permission trong AndroidManifest.xml, nếu đã có lệnh này rồi thì ta mở ứng dụng “Cài đặt” của thiết bị → Chọn “Ứng dụng” →

Chọn ứng dụng muốn thử → Chọn “Quyền” → Nhấn bật thẻ nhớ (Hình 5.8).



Hình 5.8

- **Chạy thử chức năng xem tài nguyên tĩnh:** nhấp chọn nút HƯỚNG DẪN ta thấy tập tin hướng dẫn hiện trên giao diện của lớp StaticResources.java như Hình 5.7.

(7) Sinh viên hoàn thành các nội dung bổ sung sau đây

- Tạo ra một phương thức để hiển thị thông báo thay cho đối tượng Toast,** với yêu cầu là thông điệp được hiển thị với phông chữ lớn hơn, vị trí hiển thị khoảng giữa màn hình. Gợi ý mã nguồn như sau:

```
public void displayToast (String st){
    Toast toast1=new Toast(this);
    toast1.setGravity(Gravity.CENTER,100,100);
    TextView tv1=new TextView(this);
    tv1.setBackgroundColor(Color.BLACK);
    tv1.setTextColor(Color.YELLOW);
    tv1.setTextSize(25);
    Typeface
    t1=Typeface.create("serif",Typeface.BOLD);
    tv1.setTypeface(t1);
    tv1.setPadding(10,10,10,10);
    tv1.setText(st);
    toast1.setView(tv1);
    toast1.setDuration	Toast.LENGTH_LONG);
    toast1.show();
}
```

- b. Thêm 2 nút tương tác “XÓA FILE TRÊN BỘ NHỚ TRONG” và “XÓA FILE TRÊN BỘ NHỚ NGOÀI” để xóa các file đã lưu trong bộ nhớ trong hoặc bộ nhớ ngoài để lưu nội dung mới (Nếu không còn chỗ có thể bỏ bớt ImageView. Giao diện như hình 5.9.



Hình 5.9

Hướng dẫn: tạo thêm 2 phương thức xóa file trên bộ nhớ trong và xóa file trên bộ nhớ ngoài, sau đó thêm thuộc tính onClick vào các nút tương ứng.

Để xóa file trong bộ nhớ trong ta dùng phương thức `deleteFile(“tên file”)`, đối số trả về là boolean, chúng ta có thể dùng để báo cho người dùng biết là có xóa được hay không.

Ví dụ câu lệnh: `boolean thongtinxoax= deleteFile("textfile.txt");`

Để xóa file trong bộ nhớ ngoài ta phải xác định đường dẫn đến tập tin, khai báo tập tin, sau đó dùng phương thức delete, đối số trả về là boolean, chúng ta có thể dùng để báo cho người dùng biết là có xóa được hay không.

Hãy xem đoạn mã nguồn sau:

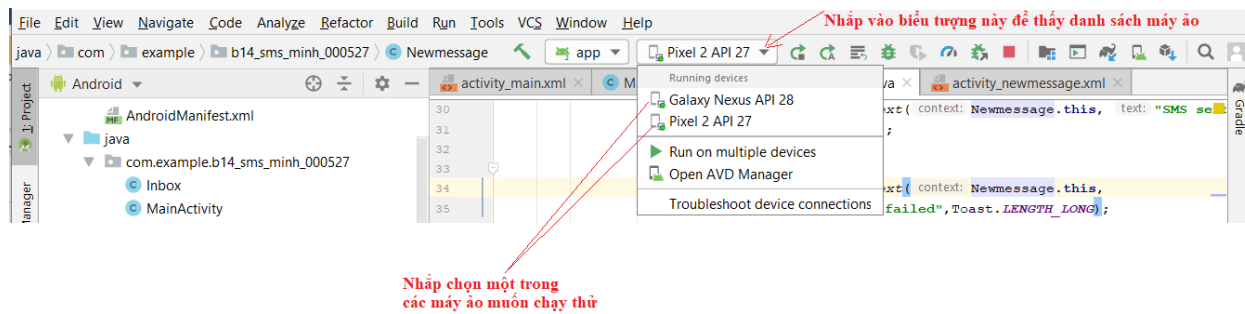
```
String kqxoa="";  
// Tìm đường dẫn đến thư mục chứa tập tin  
String root = Environment.getExternalStorageState().toString();  
// Khai báo tập tin  
File filecard = new File(root+"/Myfile.txt");  
//Xóa tập tin  
boolean xoatc= filecard.delete();  
if (xoatc==true){  
    kqxoa = " Xoa tập tin thành công";  
}  
else {  
    kqxoa = " Xoa tập tin không thành công";  
}  
//Toast.makeText(getBaseContext(),kqxoa,Toast.LENGTH_SHORT).show();  
displayToast(kqxoa);  
nhapVB.setText(""); // Xóa EditText hiển thị nội dung tập tin  
//("nhapVB" là tên EditText đã được dùng để hiển thị nội dung tập tin.
```

BÀI 11: TIN NHẮN SMS

Phát triển ứng dụng gồm các chức năng: tạo tin nhắn mới, gửi tin nhắn và xem tin nhắn trong hộp thư đến (Inbox).

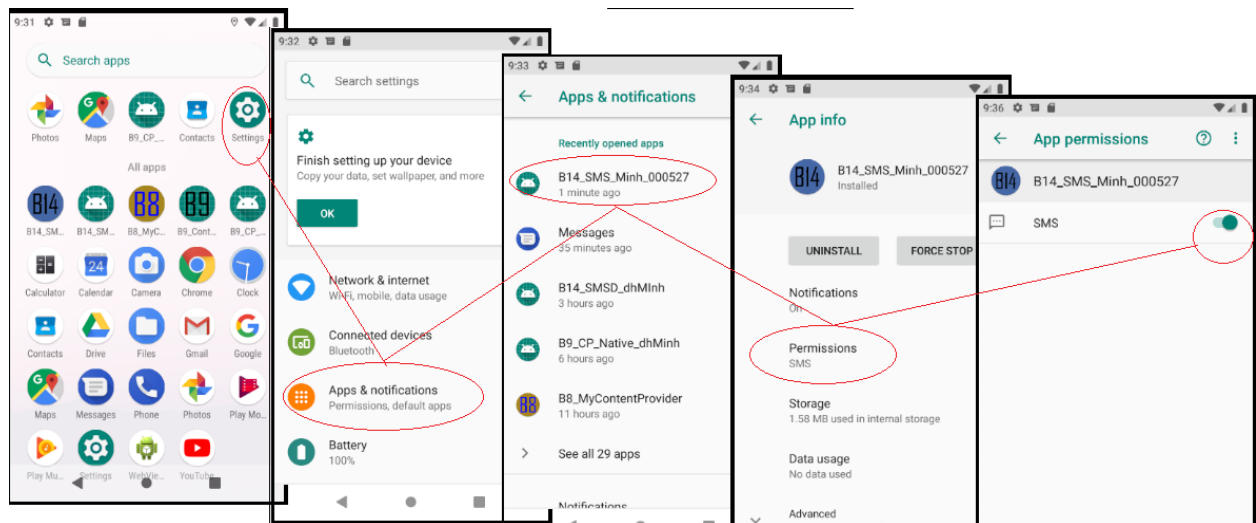
Ứng dụng được thử bằng cách mở đồng thời 2 máy ảo (emulator). Để chạy hai máy ảo cùng lúc, trước tiên ta cần tạo hai emulator và chạy project 2 lần với 2 điện thoại (emulator) khác nhau (số điện thoại tương ứng với máy ảo được mở trước là 5554 và máy ảo được mở sau là 5556), sau đó thực hiện việc gửi 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 11.1).



Hình 11.1

- 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 11.2).



Hình 11.2

Hướng dẫn thực hành:

Bước 1: Tạo project mới có tên là **B11_MSSV_Tên SV**, Activity chính là **MainActivity.java** và layout là **activity_main.xml**.

Bước 2: Lập trình layout chính activity_main.xml để có giao diện chính như Hình 11.3



Hình 11.3. 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">

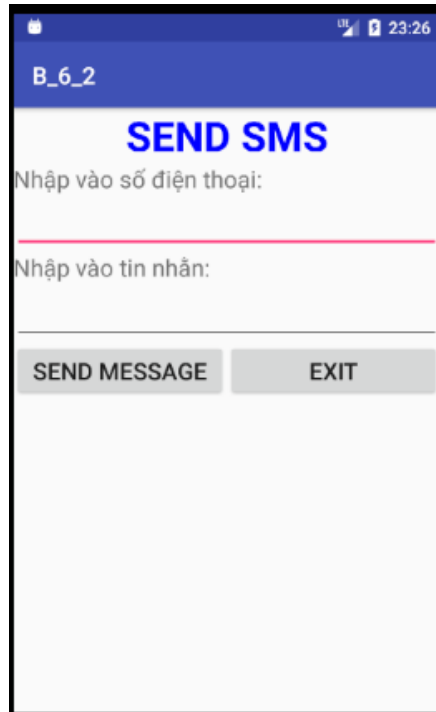
    <TextView
        android:layout_width="match_parent"
```

```
        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 11.4):

- 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 11.4: 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 = getContentResolver().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"/>

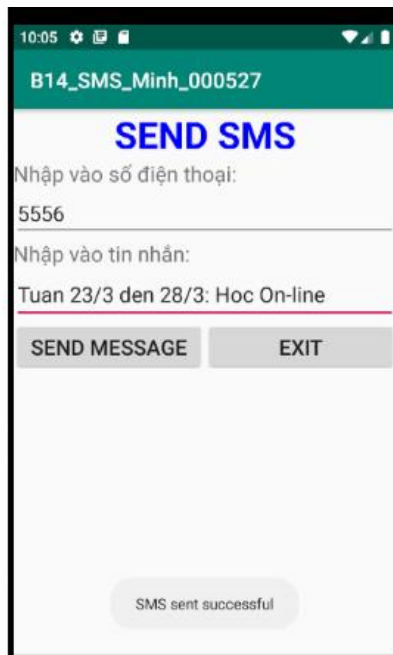
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <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 11.5.b.



Hình 11.5.a Nhập vào tin nhắn và nhấn nút SEND MESSAGE



Hình 11.5.b Nhấn nút INBOX ở máy nhận



Hình 11.5.c Hiển thị tin nhắn đã nhận được

Hình 11.5. 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.