

## **BÀI 5: INTENT VÀ INTENT FILTER**

### **I. Tóm tắt nội dung thực hành**

#### **1. Yêu cầu lý thuyết**

Trong lập trình Android, Intent và Intent Filter là hai khái niệm quan trọng để hiểu về cách các thành phần của ứng dụng tương tác với nhau và làm thế nào chúng giao tiếp thông qua hệ thống của hệ điều hành Android.

##### **a. Intent:**

Intent là một đối tượng được sử dụng để yêu cầu một hành động từ một thành phần của ứng dụng hoặc từ một ứng dụng khác trong hệ điều hành Android.

Intent có thể được sử dụng để khởi động các hoạt động (Activity), dịch vụ (Service) hoặc gửi và nhận dữ liệu giữa các thành phần của ứng dụng.

Có hai loại Intent: Implicit Intent và Explicit Intent.

##### **❖ Implicit Intent:**

Implicit Intent được sử dụng khi bạn muốn khởi chạy một hoạt động hoặc một dịch vụ mà không cần chỉ định trực tiếp tên của thành phần cụ thể.

Trong Implicit Intent, bạn chỉ xác định hành động mà bạn muốn thực hiện và hệ thống Android sẽ tìm kiếm các thành phần phù hợp để xử lý hành động đó.

Ví dụ:

```
Intent sendEmailIntent = new Intent(Intent.ACTION_SEND);
sendEmailIntent.setType("text/plain");
sendEmailIntent.putExtra(Intent.EXTRA_EMAIL, new String[]
{"example@example.com"});
sendEmailIntent.putExtra(Intent.EXTRA_SUBJECT, "Subject");
sendEmailIntent.putExtra(Intent.EXTRA_TEXT, "Message Body");
startActivity(Intent.createChooser(sendEmailIntent, "Send Email"));
```

Trong đoạn mã trên, Implicit Intent được sử dụng để khởi chạy một Intent để gửi email. Trong trường hợp này, hệ thống Android sẽ tìm kiếm tất cả các ứng dụng hỗ trợ việc gửi email và hiển thị cho người dùng lựa chọn.

#### ❖ **Explicit Intent:**

Explicit Intent được sử dụng khi mà muốn chính xác xác định thành phần (Activity hoặc Service) muốn khởi chạy.

Trong Explicit Intent, cần cung cấp tên của lớp thành phần muốn khởi chạy, thông qua Constructor của Intent hoặc phương thức `setClass()`.

Ví dụ:

```
Intent detailIntent = new Intent(MainActivity.this, NoteDetailActivity.class);
detailIntent.putExtra("note_id", noteId);
startActivity(detailIntent);
```

Trong đoạn mã trên, Explicit Intent được sử dụng để khởi chạy một Intent để hiển thị chi tiết ghi chú. Trong trường hợp này, Intent sẽ chỉ rõ rằng nó muốn mở NoteDetailActivity.

#### **b. Intent Filter:**

- Intent Filter là một cấu hình được định nghĩa trong tệp AndroidManifest.xml của ứng dụng, dùng để xác định loại hành động hoặc sự kiện mà một thành phần của ứng dụng có thể xử lý.

- Intent Filter cho phép các thành phần của ứng dụng đăng ký để nhận các Intent với các hành động hoặc sự kiện cụ thể.

- Một thành phần có thể có nhiều Intent Filter, mỗi Intent Filter có thể định nghĩa một hành động cụ thể mà thành phần đó có thể xử lý.

#### **Một số loại Intent Filter thông dụng trong Android:**

1. ACTION\_VIEW: Intent này được sử dụng để hiển thị dữ liệu được chỉ định bởi một Uri. Ví dụ, nếu muốn mở một trang web, ta có thể sử dụng ACTION\_VIEW với Uri của trang web đó.

2. ACTION\_SEND: Intent này được sử dụng để chia sẻ dữ liệu với các ứng dụng khác. Ví dụ, có thể sử dụng ACTION\_SEND để chia sẻ một hình ảnh từ ứng dụng của mình đến ứng dụng khác.

3. ACTION\_CALL: Intent này được sử dụng để khởi động cuộc gọi điện. Cần cung cấp Uri của số điện thoại mình muốn gọi.

4. ACTION\_SENDTO: Intent này cũng được sử dụng để chia sẻ dữ liệu nhưng nó yêu cầu ứng dụng nhận xử lý dữ liệu phải là một ứng dụng email hoặc tin nhắn. Ví dụ, ta có thể sử dụng ACTION\_SENDTO để mở ứng dụng email với địa chỉ email đã được điền sẵn.

5. ACTION\_PICK: Intent này được sử dụng để chọn một loại dữ liệu từ một danh sách. Ví dụ, có thể sử dụng ACTION\_PICK để chọn một hình ảnh từ thư viện ảnh của thiết bị.

6. ACTION\_EDIT: Intent này được sử dụng để chỉnh sửa dữ liệu. Ví dụ, có thể sử dụng ACTION\_EDIT để chỉnh sửa thông tin liên hệ trong ứng dụng Contacts.

7. ACTION\_MAIN: Intent này được sử dụng để chạy một Activity là Activity chính của ứng dụng.

8. ACTION\_SEARCH: Intent này được sử dụng để thực hiện tìm kiếm trong ứng dụng.

#### **Ví dụ:**

Intent Filter có thể hiện bằng Dialpad để bấm số gọi điện thoại:

```
Intent intent = new Intent(Intent.ACTION_DIAL,Uri.parse("tel:(+84)123456789"));
```

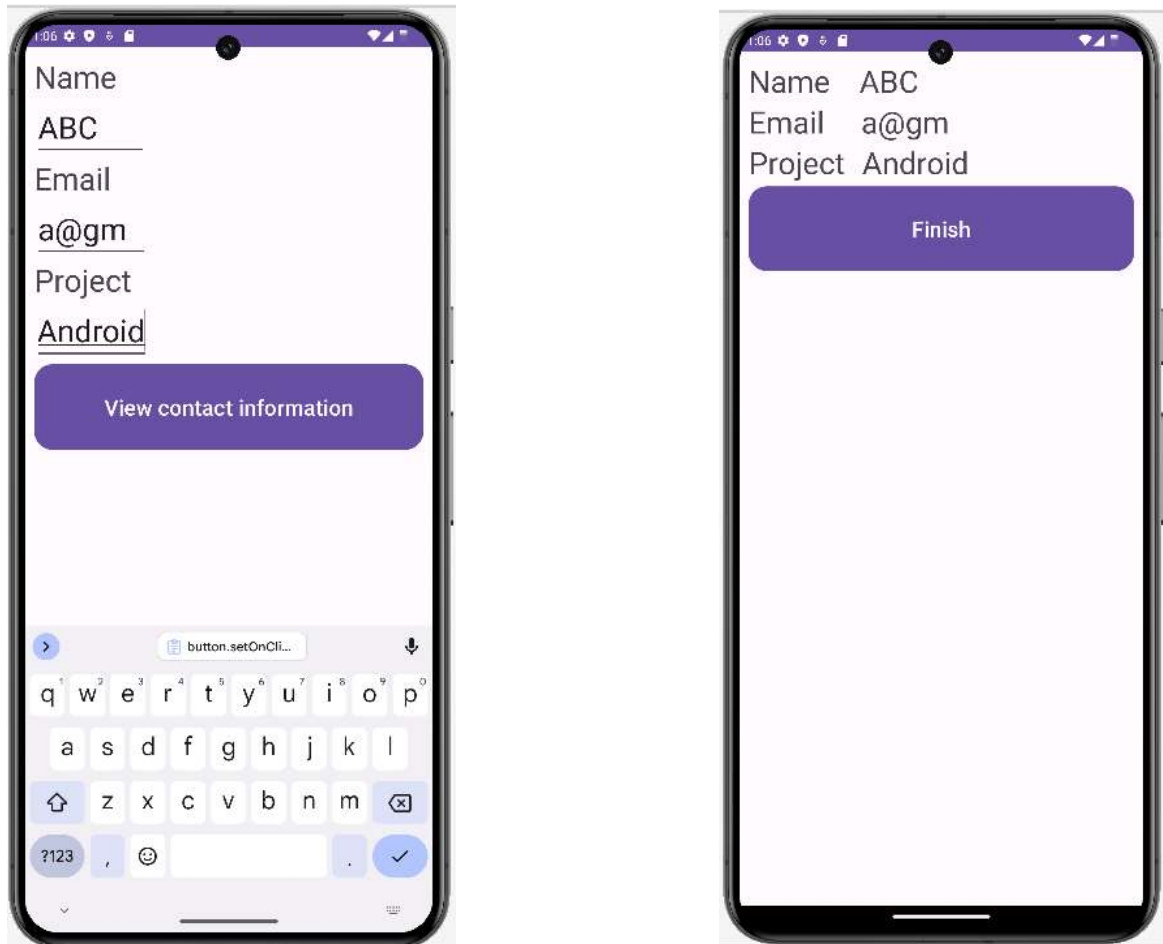
## **2. Nội dung**

Thực hiện các bài tập ứng dụng intent và intent filter theo hướng dẫn.

## **II. Bài tập yêu cầu**

**Bài 1:** Xây dựng ứng dụng có giao diện như hình, cho phép nhập các thông tin về tên, email và tên dự án. Khi nhấn button “View Contact Information thì màn hình sẽ

chuyển sang màn hình thứ hai, hiển thị các thông tin người dùng vừa nhập.



Hình 43 Giao diện bài tập 5.1

### **Hướng dẫn:**

-Tạo giao diện trên 2 file activity\_main.xml (nhập thông tin) và contactinfo.xml (hiển thị thông tin)

Activity\_main.xml với edit text nhập tên:

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView...>

    <EditText
        android:id="@+id/txtName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="text"
        android:textSize="34sp"
        tools:layout_editor_absoluteX="146dp"
        tools:layout_editor_absoluteY="20dp"
        android:contentDescription="Nhập tên"/>
    </EditText>
</LinearLayout>

```

-Tương tự cho EditText nhập email và nhập dự án.

-Xây dựng Activity chính với giao diện activity\_main.xml

```

public class MainActivity extends AppCompatActivity {

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

        EditText txtName = findViewById(R.id.txtName);
        EditText txtEmail = findViewById(R.id.txtEmail);
        EditText txtProject = findViewById(R.id.txtProject);

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {...});
    }
}

```

-Viết xử lý cho button

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent getContactInfo = new Intent(getApplicationContext(), contactinfo.class);
        Bundle bundle = new Bundle();
        bundle.putString("nameKey", txtName.getText().toString());
        bundle.putString("emailKey", txtEmail.getText().toString());
        bundle.putString("projectKey", txtProject.getText().toString());
        getContactInfo.putExtras(bundle);

        startActivity(getContactInfo);
    }
});
```

Device

-Giải thích code:

Tạo một intent liên lạc giữa context (activity hiện thời) và contactinfo

```
Intent getContactInfo = new Intent(getApplicationContext(), contactinfo.class);
```

Tạo 1 bundle chứa các thông tin bổ sung

```
Bundle bundle = new Bundle();
bundle.putString("nameKey", txtName.getText().toString());
bundle.putString("emailKey", txtEmail.getText().toString());
bundle.putString("projectKey", txtProject.getText().toString());
```

Thêm các thông tin bổ sung vào intent

```
getContactInfo.putExtras(bundle);
```

Hoặc có thể thêm extras trực tiếp vào intent bằng lệnh

```
intent.putExtra("nameKey", txtName.getText().toString());
```

Start activity mới

```
startActivity(getContactInfo);
```

-Xây dựng contactinfo (file contactinfo.java), lấy và hiển thị các giá trị do activity chính gửi qua

```
public class contactinfo extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        TextView txtName;  
        TextView txtEmail;  
        TextView txtProject;  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_contactinfo);  
        txtName = (TextView) findViewById(R.id.name);  
        txtEmail = (TextView) findViewById(R.id.email);  
        txtProject = (TextView) findViewById(R.id.project);  
        Button finishBtn = (Button) findViewById(R.id.button);  
        finishBtn.setOnClickListener(finishClick);  
        Bundle bundle = getIntent().getExtras();  
        String name = bundle.getString(key: "nameKey");  
        String email = bundle.getString(key: "emailKey");  
        String project = bundle.getString(key: "projectKey");  
        txtName.setText(name);  
        txtEmail.setText(email);  
        txtProject.setText(project);  
    }  
}
```

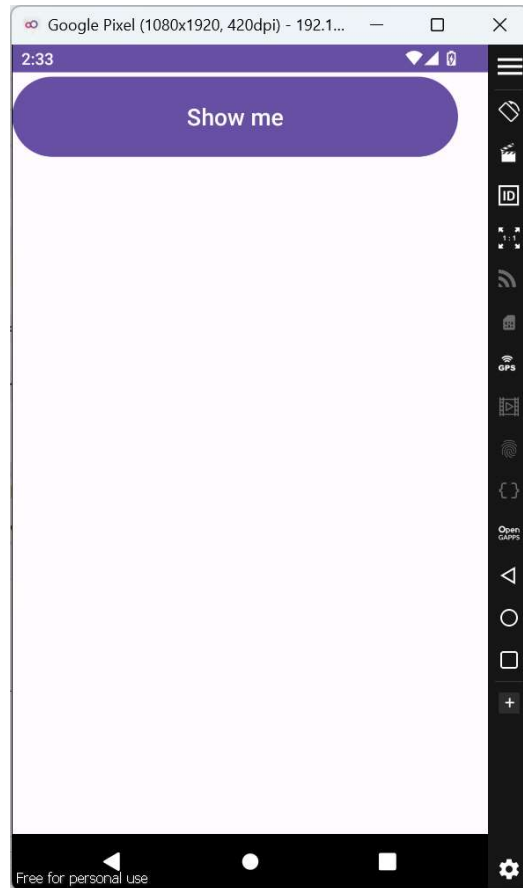
-Viết xử lý cho button finish: kết thúc activity

1 usage

```
private View.OnClickListener finishClick = new View.OnClickListener() {  
    @Override  
    public void onClick(View v) { finish(); }  
};
```

## Bài 2:

Tạo ứng dụng minh họa các intent-filter



Hình 44 Giao diện bài tập 5.2

### Hướng dẫn:

-Tạo giao diện ứng dụng có 1 button như hình



```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:layout_editor_absoluteY="10dp">

    <Button
        android:id="@+id/button2"
        android:layout_width="385dp"
        android:layout_height="77dp"
        android:text="Show me"
        android:textSize="20sp"
        tools:layout_editor_absoluteX="10dp"
        tools:layout_editor_absoluteY="16dp" />

</LinearLayout>

```

-Viết xử lý cho button

```

Button button = (Button) findViewById(R.id.button2);

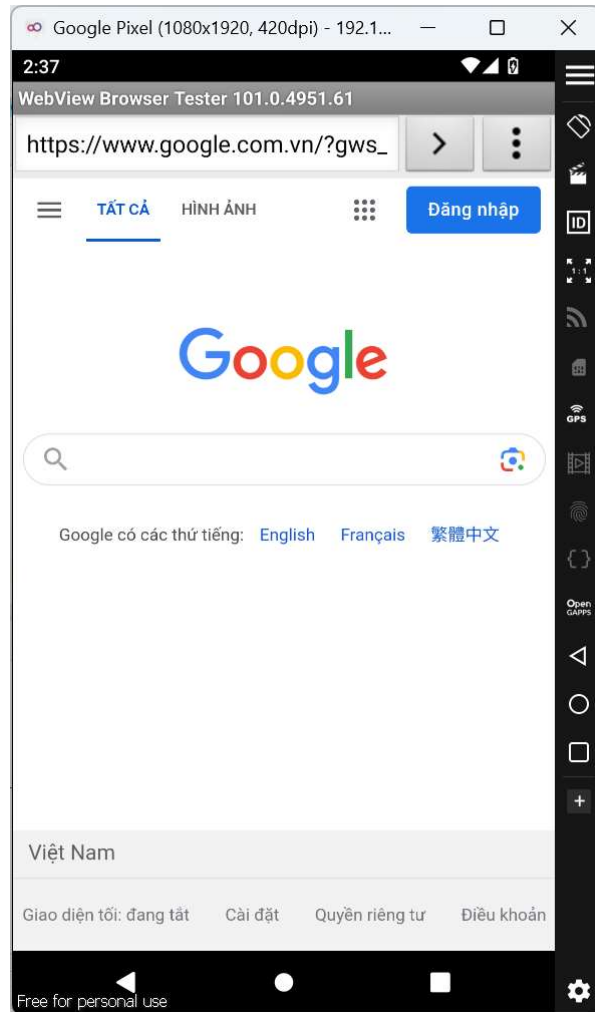
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse( uriString: "http://www.google.com.vn"));
        startActivity(intent);
    }
});

```

-Giải thích code: ở đây ta đang tạo ra một intent có khả năng View 1 trang web.

Constructor của Intent có 2 tham số tương ứng.

-Chạy ứng dụng và xem kết quả



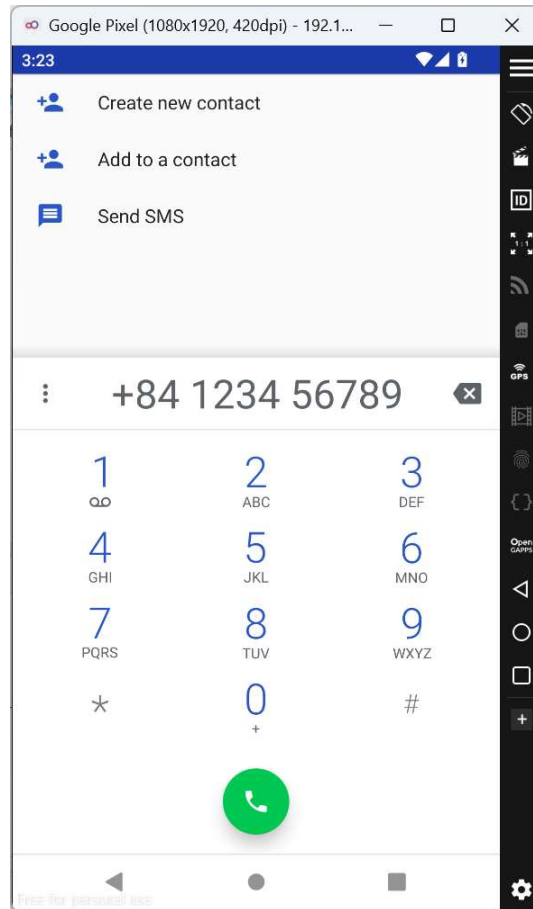
Hình 45 Giao diện bài tập 5.2

### Bài 3:

Tiếp tục ứng dụng của bài tập 2, thay thế bằng các intent filter khác và xem kết quả

Intent.ACTION\_DIAL

```
Intent intent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:(+84)123456789"));
```

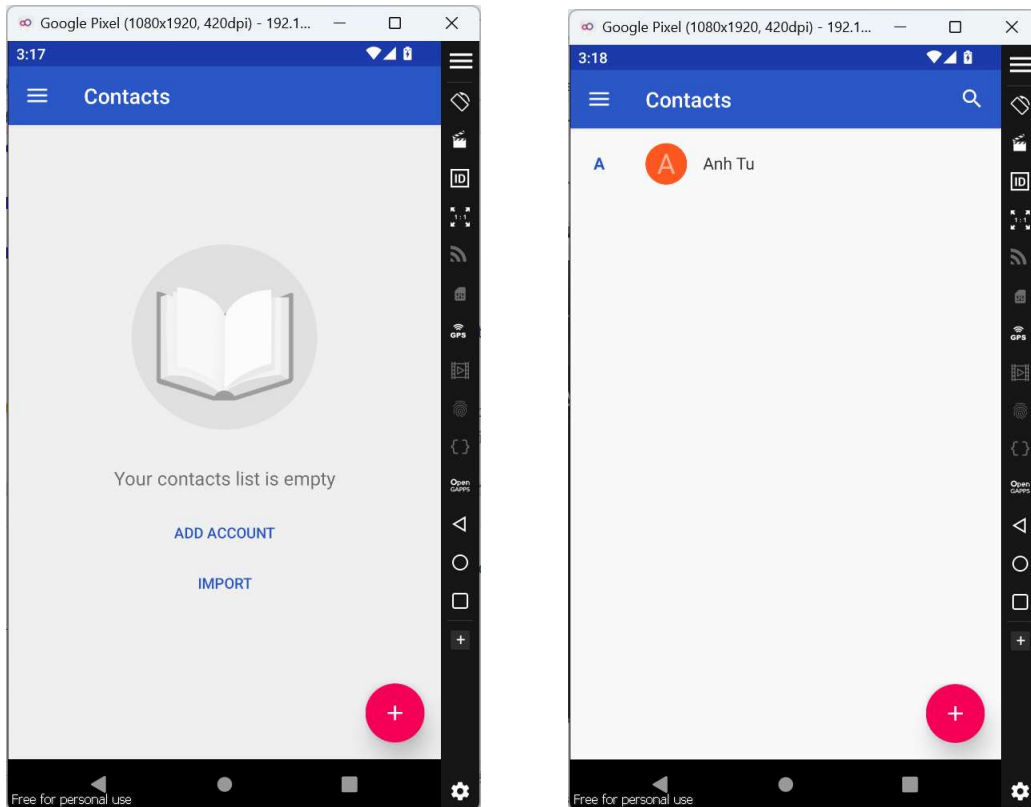


Hình 46 Giao diện bài tập 5.3

Intent.ACTION\_VIEW

```
Intent intent = new
```

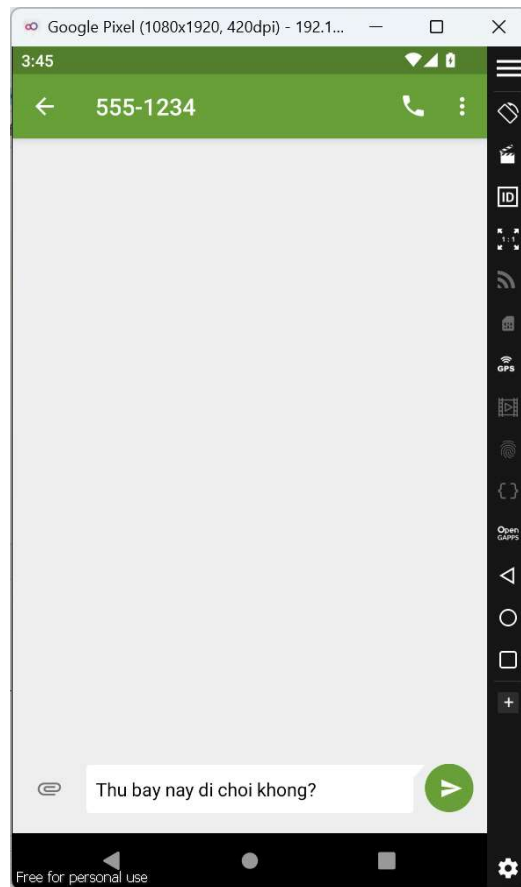
```
Intent(Intent.ACTION_VIEW, Uri.parse("content://contacts/people/"));
```



Hình 47 Giao diện bài tập 5.3

Intent.ACTION\_SENDTO

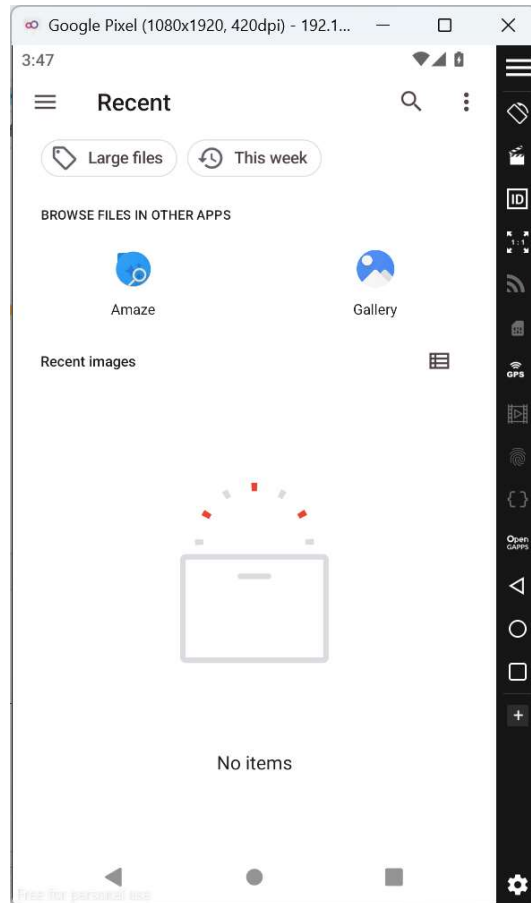
```
Intent intent = new Intent( Intent.ACTION_SENDTO, Uri.parse("sms:5551234"));
intent.putExtra("sms_body", "Thu bay nay di choi khong?");
```



Hình 48 Giao diện bài tập 5.3

Xem ảnh

```
Intent myIntent = new Intent();  
  
myIntent.setType("image/pictures/*");  
  
myIntent.setAction(Intent.ACTION_GET_CONTENT);  
  
startActivity(myIntent);
```



Hình 49 Giao diện bài tập 5.3

Nghe nhạc

```
Intent myActivity2 = new Intent("android.intent.action.MUSIC_PLAYER");
startActivity(myActivity2);
```

Tìm đường đi từ điểm A đến điểm B

```
String url = "http://maps.google.com/maps?" +
    "saddr=9.938083,-84.054430&daddr=9.926392,-84.055964";

Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
startActivity(intent);
```

-Tìm hiểu và tham khảo thêm các intent, intent-filter khác.

### III. Câu hỏi ôn tập:

1. Intent được sử dụng để làm gì?

A. Chuyển dữ liệu giữa các Activity `putExtra()` để đóng gói dữ liệu và `getExtra()` để lấy ra.

B. Khởi chạy các Component khác trong ứng dụng Dùng cho Activity và khởi chạy các thành phần khác như Service, BroadcastReceiver, và hiển thị Content Provider.

C. Cả A và B đều đúng

2. Có bao nhiêu loại Intent chính trong Android?

A. 1 loại

B. 2 loại (Explicit Intent và Implicit Intent)

C. 3 loại

3. Khi nào chúng ta sử dụng Explicit Intent?

A. Khi muốn khởi chạy một Component cụ thể trong ứng dụng

B. Khi muốn khởi chạy một Component bên ngoài ứng dụng

C. Cả A và B đều đúng

4. Khi nào chúng ta sử dụng Implicit Intent?

A. Khi muốn khởi chạy một Component cụ thể trong ứng dụng

B. Khi muốn khởi chạy một Component bên ngoài ứng dụng

C. Cả A và B đều đúng

5. Để truyền dữ liệu giữa các Activity, chúng ta sử dụng phương thức nào?

A. `putExtra()` truyền dữ liệu dạng key-value

B. `setData()` truyền một URI đến component đích

C. Cả A và B đều đúng

6. Khi sử dụng Implicit Intent, chúng ta cần khai báo Intent Filter trong file AndroidManifest.xml. Những thông tin nào cần khai báo trong Intent Filter?

- A. Action
- B. Category
- C. Cả A và B đều đúng

7. Phương thức nào được sử dụng để lấy dữ liệu từ Intent?

- A. `getStringExtra()`
- B. `getData()`
- C. Cả A và B đều đúng

8. Với Intent Filter có action "android.intent.action.MAIN", Component sẽ được hiển thị như thế nào?

- A. Như một entry point cho ứng dụng
- B. Như một component bị ẩn
- C. Không được hiển thị

9. Intent Filter với action "android.intent.action.VIEW" được sử dụng để:

- A. Hiển thị một Component như entry point của ứng dụng
- B. Xử lý các yêu cầu xem nội dung (view content)
- C. Xử lý các yêu cầu chỉnh sửa nội dung

10. Intent Filter với action "android.intent.action.CALL" được sử dụng để:

- A. Thực hiện một cuộc gọi
- B. Tiếp nhận một cuộc gọi
- C. Cả A và B đều đúng