



**FPT POLYTECHNIC**

LẬP TRÌNH ANDROID NÂNG CAO

**Bài 7: Truy cập dữ liệu  
Contact**

---

[www.poly.edu.vn](http://www.poly.edu.vn)

---

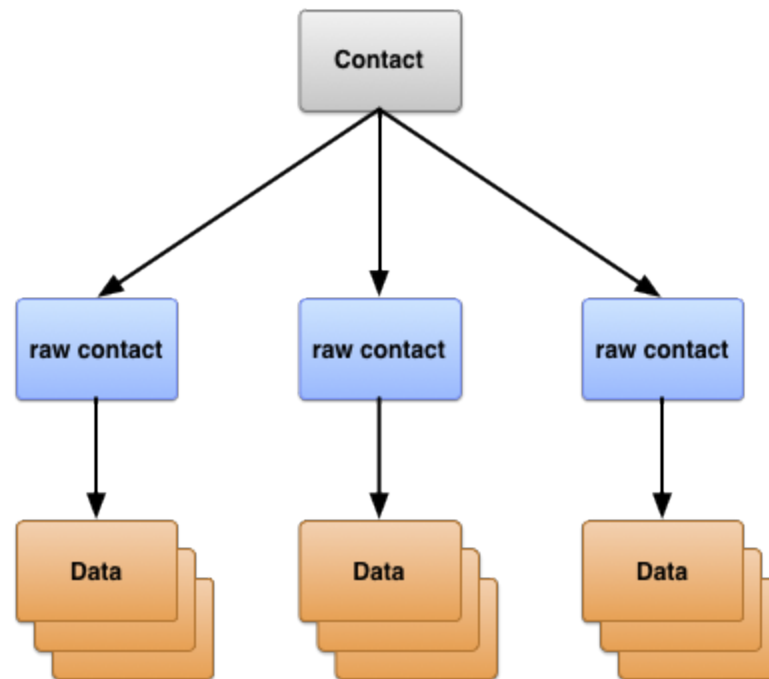
## Nội dung bài học

- Tổng quan về Contacts Provider
- Truy cập danh sách Contact
- Hiển thị thông tin chi tiết của Contact
- Thay đổi Contact sử dụng Intent
- Hiển thị Quick Contact Badge



# Contact Provider

- Contacts Provider là thành phần cung cấp nội dung của Android
- Cung cấp ba loại dữ liệu về person, mỗi loại tương ứng với một bảng được cung cấp bởi provider



# Contact Provider

- Ba bảng này được tham chiếu tới tên của các lớp contract
- Các lớp định nghĩa các hằng số như content URI, tên cột, giá trị của cột được sử dụng bởi ba bảng sau:
  - Bảng **ContactsContract.Contacts**: thể hiện các contact khác nhau, dựa trên các cột contact nguyên thủy (raw contact)
  - Bảng **ContactsContract.RawContacts**: cung cấp thông tin dữ liệu về person, như user account và type
  - Bảng **ContactsContract.Data**: cung cấp thông tin contact nguyên thủy

# Contact Provider

- Contacts Provider cung cấp thông tin liên hệ (contact) của người dùng bao gồm dữ liệu từ các ứng dụng contacts và các ứng dụng mạng xã hội
- Trong ứng dụng, bạn có thể truy cập thông tin contact sử dụng phương thức **ContentResolver** hoặc gửi intent tới ứng dụng **Contacts**
- Cung cấp các phương thức để hiển thị danh sách contact, xem thông tin chi tiết của contact, thay đổi thông tin contact sử dụng intent

## Contact nguyên thủy

- Contact nguyên thủy bao gồm dữ liệu người dùng đến từ một kiểu account cụ thể và tên account
- Contact Provider cho phép nhiều hơn một dịch vụ online cung cấp dữ liệu về contact, do đó Contacts Provider cho phép nhiều dữ liệu contact nguyên thủy cho cùng một người dùng
- Nhiều dữ liệu contact nguyên thủy cho phép một người dùng liên kết nhiều dữ liệu người dùng từ nhiều account khác nhau
- Hầu hết dữ liệu của contact nguyên thủy không được lưu trữ trong bảng `ContactsContract.RawContacts` mà được lưu trữ trong một hoặc nhiều dòng của bảng `ContactsContract.Data`

## Các cột dữ liệu nguyên thủy quan trọng

- **ACCOUNT\_NAME**: tên tài khoản của contact nguyên thủy, ví dụ tên của một tài khoản google account là một trong các địa chỉ Gmail của người dùng thiết bị Android, ví dụ kiểu tài khoản của Google account là com.google
- **ACCOUNT\_TYPE**: kiểu tài khoản của contact nguyên thủy
- **DELETED**: cờ khóa cho contact nguyên thủy

# Các nguồn dữ liệu nguyên thủy

- Giả sử người dùng “Emily Dickinson” có 3 tài khoản sau trên thiết bị của anh ấy
  - [emily.dickinson@gmail.com](mailto:emily.dickinson@gmail.com)
  - [emilyd@gmail.com](mailto:emilyd@gmail.com)
  - Twitter account “belle\_of\_amherst”
- Người dùng này kích hoạt chức năng Sync Contacts cho cả 3 tài khoản
- Khi Emily Dickinson mở tài khoản trên trình duyệt, đăng nhập vào Gmail với tài khoản [emily.dickinson@gmail.com](mailto:emily.dickinson@gmail.com) và thêm “Thomas Higginson”. Sau đó, đăng nhập vào [emilyd@gmail.com](mailto:emilyd@gmail.com) và gửi cho “Thomas Higginson” một email, “Thomas Higginson” sẽ tự động được thêm vào contact
- Anh ấy cũng follow “Thomas Higginson” trên twitter



# Các nguồn dữ liệu nguyên thủy

- Khi đó, Contacts Provider sẽ tạo ba contact nguyên thủy như sau:
  - Một contact nguyên thủy đối với "Thomas Higginson" ứng với email emily.dickinson@gmail.com
  - Một contact nguyên thủy thứ hai đối với "Thomas Higginson" ứng với email emilyd@gmail.com. Kiểu tài khoản vẫn là Google. Khi đó, sẽ có dòng contact nguyên thủy thứ hai mặc dù trùng tên với contact trước đó
  - Một contact nguyên thủy thứ ba đối với "Thomas Higginson" ứng với twitter "belle\_of\_amherst". Kiểu tài khoản người dùng là Twitter

## Dữ liệu

- Dữ liệu của contact nguyên thủy được lưu trữ trong dòng ContactsContract.Data được link tới giá trị \_ID của contact. Do đó cho phép một contact nguyên thủy có nhiều thể hiện cho cùng kiểu dữ liệu như địa chỉ email hoặc số điện thoại
- Ví dụ “Thomas Higginson” đối với account emily.dickinson@gmail.com (dòng dữ liệu của contact nguyên thủy của “Thomas Higginson” liên kết với tài khoản Google emily.dickinson@gmail.com) có địa chỉ email công việc là thao@gmail.com và email cá nhân là thao\_2000@gmail.com, Contacts Provider sẽ lưu trữ hai địa chỉ email và liên kết cả hai tới contact nguyên thủy

## Tên cột miêu tả

- Nhiều loại dữ liệu được lưu trữ trong một bảng. Display name, phone number, email, photo, website đều được lưu trữ trong bảng `ContactsContract.Data`
- Để quản lý các dữ liệu này, `ContactsContract.Data` có vài cột với tên miêu tả và một số cột khác với tên chung chung
- Nội dung của cột có tên miêu tả có chung ý nghĩa đối với mọi kiểu dữ liệu trong cột, trong khi đó nội dung của các cột với tên chung chung khác nhau đối với các kiểu dữ liệu khác nhau

## Tên cột miêu tả

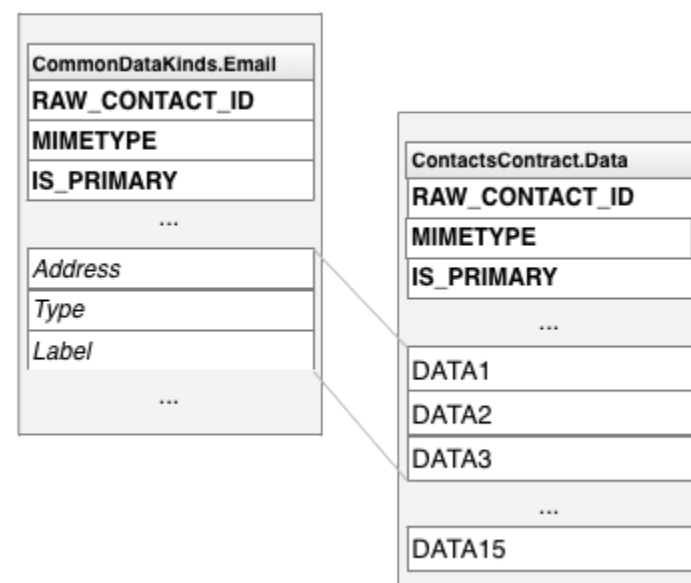
- Một số ví dụ về cột miêu tả:
  - RAW\_CONTACT\_ID: giá trị của cột \_ID trong contact nguyên thủy
  - MIMETYPE: kiểu dữ liệu được lưu trữ trong dòng, là kiểu MIMETYPE tùy biến. Contacts Provider sử dụng kiểu MINE định nghĩa trong các lớp con của ContactsContract.CommonDataKinds
  - IS\_PRIMARY: cột IS\_PRIMARY đánh dấu dòng dữ liệu chứa dữ liệu chính của kiểu. Ví dụ khi người dùng long-press trên số điện thoại của một contact và chọn **Set Default**, khi đó dòng ContactsContract.Data sẽ có giá trị khác 0 ở cột IS\_PRIMARY

## Các cột chung

- Có 15 cột chung tên từ DATA1 đến DATA15 và các cột SYNC1 đến SYNC4 (được sử dụng bởi sync adapter)
- Cột Data1 được đánh chỉ số (index). Contact Provider thường sử dụng cột này để lưu dữ liệu mà provider sử dụng nhiều nhất, ví dụ chứa email thực của contact
- Cột Data15 thường dùng để lưu dữ liệu BLOB như ảnh thumbnails

## Cột dữ liệu cụ thể

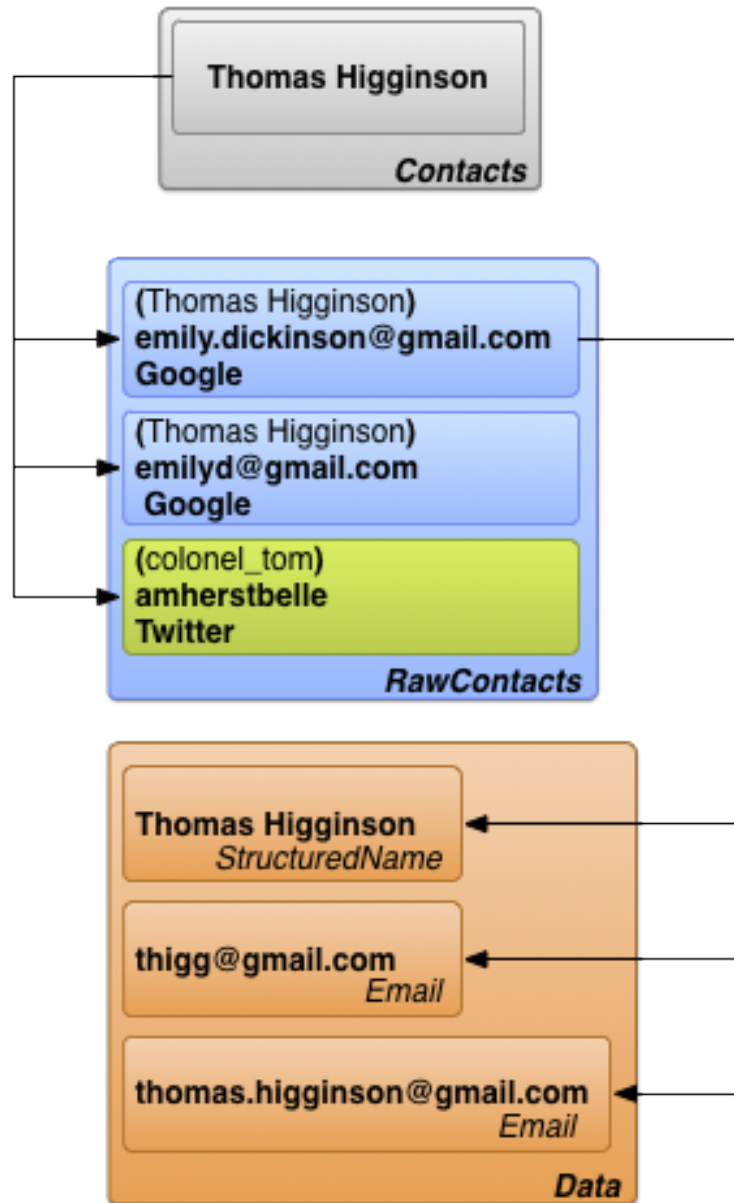
- Contacts Provider cung cấp hằng số tên cột cụ thể cho kiểu xác định, định nghĩa trong các lớp con của `ContactsContract.CommonDataKinds`
- Ví dụ lớp `ContactsContract.CommonDataKinds.Email` định nghĩa hằng số tên cột kiểu xác định cho dòng có MIME type là `Email.CONTENT_ITEM_TYPE`. Lớp này chứa hằng số `ADDRESS` cho cột địa chỉ Email. Giá trị thực tế của `ADDRESS1` là "data1"



## **Các lớp tên cột xác định**

- `ContactsContract.CommonDataKinds.StructuredName`
- `ContactsContract.CommonDataKinds.Photo`
- `ContactsContract.CommonDataKinds.Email`
- `ContactsContract.CommonDataKinds.StructuredPostal`
- `ContactsContract.CommonDataKinds.GroupMembership`

# Contact





## Yêu cầu Permission đối với Provider

- READ\_CONTACTS, thêm vào AndroidManifest.xml với phần tử <uses-permission>, cụ thể là **<uses-permission android:name="android.permission.READ\_CONTACTS">**
- WRITE\_CONTACTS, thêm vào AndroidManifest.xml với phần tử <uses-permission>, cụ thể là **<uses-permission android:name="android.permission.WRITE\_CONTACTS">**

# Hiển thị thông tin chi tiết của Contact

```
private void GetContactData()  
{  
    String phoneNumber = null;  
    String email = null;  
  
    Uri CONTENT_URI = ContactsContract.Contacts.CONTENT_URI;  
    String _ID = ContactsContract.Contacts._ID;  
    String DISPLAY_NAME = ContactsContract.Contacts.DISPLAY_NAME;  
    String HAS_PHONE_NUMBER = ContactsContract.Contacts.HAS_PHONE_NUMBER;  
  
    Uri PhoneCONTENT_URI = ContactsContract.CommonDataKinds.Phone.CONTENT_URI;  
    String Phone_CONTACT_ID = ContactsContract.CommonDataKinds.Phone.CONTACT_ID;  
    String NUMBER = ContactsContract.CommonDataKinds.Phone.NUMBER;  
  
    Uri EmailCONTENT_URI = ContactsContract.CommonDataKinds.Email.CONTENT_URI;  
    String EmailCONTACT_ID = ContactsContract.CommonDataKinds.Email.CONTACT_ID;  
    String DATA = ContactsContract.CommonDataKinds.Email.DATA;  
  
    StringBuffer output = new StringBuffer();  
  
    ContentResolver contentResolver = getContentResolver();  
  
    Cursor cursor = contentResolver.query(CONTENT_URI, null, null, null, null);
```

# Hiển thị thông tin chi tiết của Contact

```
if (cursor.getCount() > 0) {
    while (cursor.moveToNext()) {
        String contact_id = cursor.getString(cursor.getColumnIndex( _ID ));
        String name = cursor.getString(cursor.getColumnIndex( DISPLAY_NAME ));
        int hasPhoneNumber = Integer.parseInt(cursor.getString(cursor.getColumnIndex( HAS_PHONE_NUMBER )));
        if (hasPhoneNumber > 0) {
            output.append("\n Tên:" + name);
            // Query and loop for every phone number of the contact
            Cursor phoneCursor = contentResolver.query(PhoneCONTENT_URI,
                null,
                Phone_CONTACT_ID + " = ?",
                new String[] { contact_id }, null);
            while (phoneCursor.moveToNext()) {
                phoneNumber = phoneCursor.getString(phoneCursor.getColumnIndex(NUMBER));
                String str = phoneCursor.getString(phoneCursor.getColumnIndex(
                    ContactsContract.CommonDataKinds.Phone.TYPE));
                output.append("\n Số điện thoại(" + getPhoneType(str) + "):" + phoneNumber);
            }
            phoneCursor.close();
            // Query and loop for every email of the contact
            Cursor emailCursor = contentResolver.query(EmailCONTENT_URI,
                null, EmailCONTACT_ID+ " = ?", new String[] { contact_id }, null);
            while (emailCursor.moveToNext()) {
                email = emailCursor.getString(emailCursor.getColumnIndex(DATA));
                output.append("\nĐịa chỉ Email:" + email);
            }
            emailCursor.close();
        }
        output.append("\n");
    }
}
outputText.setText(output);
}
```



**DEMO**

Hiển thị danh sách Contact



# Thay đổi thông tin Contacts

- Bạn có thể thêm mới, sửa đổi, xóa dữ liệu trong Contacts Provider ở chế độ batch mode bằng cách tạo một ArrayList của các đối tượng ContentProviderOperation và gọi `applyBatch()`
- Để thay đổi thông tin một contact cụ thể, bạn nên gửi một intent tới ứng dụng contact của thiết bị

# Xóa Contact

```
private static long getContactID(ContentResolver contactHelper,
    String number) {
    Uri contactUri = Uri.withAppendedPath(PhoneLookup.CONTENT_FILTER_URI,
        Uri.encode(number));
    String[] projection = { PhoneLookup._ID };
    Cursor cursor = null;
    try {
        cursor = contactHelper.query(contactUri, projection, null, null,
            null);

        if (cursor.moveToFirst()) {
            int personID = cursor.getColumnIndex(PhoneLookup._ID);
            return cursor.getLong(personID);
        }
        return -1;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (cursor != null) {
            cursor.close();
            cursor = null;
        }
    }
    return -1;
}
```

# Xóa Contact

```
public static void deleteContact(ContentResolver contactHelper,
    String number) {

    ArrayList<ContentProviderOperation> ops = new ArrayList<ContentProviderOperation>();
    String[] args = new String[] { String.valueOf(getContactID(
        contactHelper, number)) };

    ops.add(ContentProviderOperation.newDelete(RawContacts.CONTENT_URI)
        .withSelection(RawContacts.CONTACT_ID + "=?", args).build());
    try {
        contactHelper.applyBatch(ContactsContract.AUTHORITY, ops);
    } catch (RemoteException e) {
        e.printStackTrace();
    } catch (OperationApplicationException e) {
        e.printStackTrace();
    }
}
```

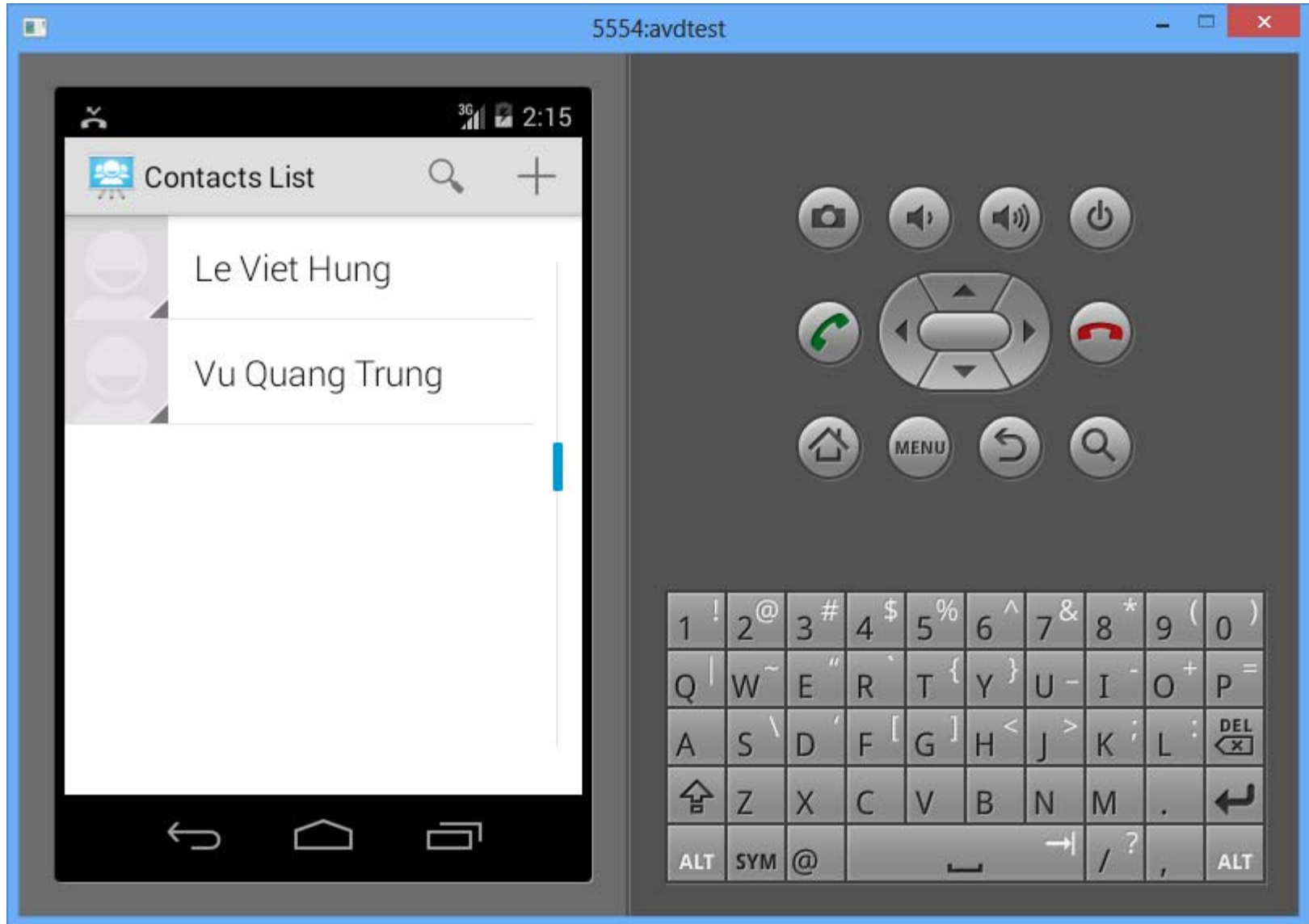




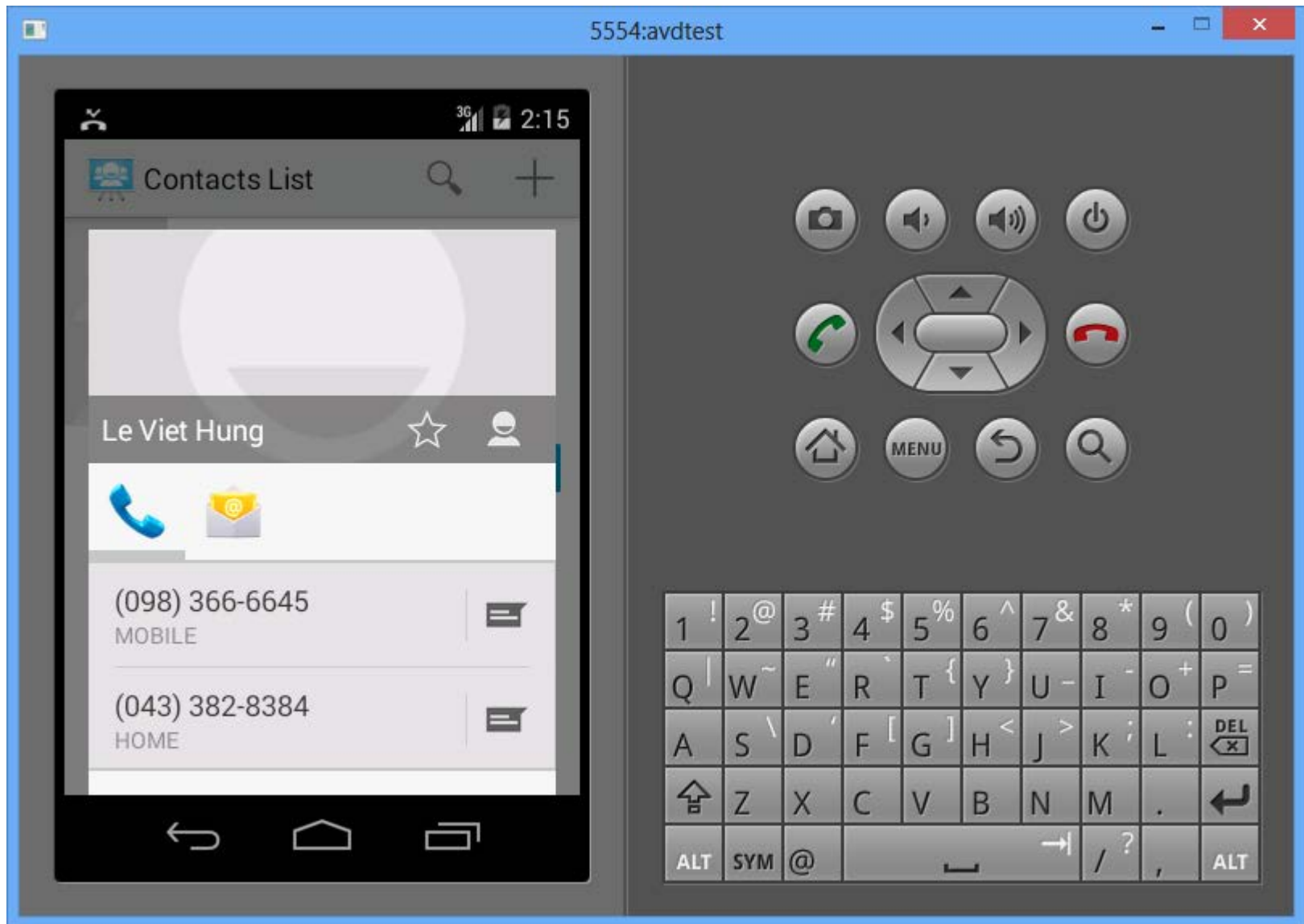
# Hiển thị Quick Contact Badge

- Quick Contact Badge là một widget xuất hiện như một ảnh thumbnail
- Bạn có thể sử dụng Bitmap cho ảnh thumbnail
- Ảnh nhỏ hoạt động như một control, khi người dùng click vào ảnh, QuickContactBadge mở rộng một dialog chứa các thông tin sau:
  - Một ảnh lớn: ảnh lớn liên quan đến contact
  - App icons:

# Hiển thị Quick Contact Badge



# Hiển thị Quick Contact Badge



# Hiển thị Quick Contact Badge

- Để thêm QuickContactBadge, bổ sung `<QuickContactBadge>` vào layout

```
<QuickContactBadge android:id="@android:id/icon"  
    android:layout_height="?android:attr/listPreferredItemHeight"  
    android:layout_width="?android:attr/listPreferredItemHeight"  
    android:scaleType="centerCrop"  
    style="@style/quickContactBadgeStyle"  
    android:src="@drawable/ic_contact_picture_holo_light"/>
```



DEMO

Sử dụng Quick Contact Badge



## Tổng kết nội dung bài học

- Tổng quan về Contacts Provider
- Truy cập danh sách Contact
- Hiển thị thông tin chi tiết của Contact
- Thay đổi Contact sử dụng Intent
- Hiển thị Quick Contact Badge

