

# Lab Sheet: Read & Write Contacts in Android

## Objective

Enhance the existing contacts demo project to:

- Read contacts and display them in a RecyclerView.
- Add new contacts using WRITE\_CONTACTS permission.
- Refresh the list after insertion.

## Step 1: Update Manifest

- Open AndroidManifest.xml.

- Add both permissions:

```
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
```

Checkpoint: Manifest includes both permissions.

## Step 2: Add UI for Writing Contacts

- In activity\_main.xml, add a Button above the RecyclerView:

```
<Button
    android:id="@+id/btnAddContact"
    android:text="Add Contact"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

Checkpoint: Button appears in the app.

## Step 3: Handle Button Click

- In MainActivity.java, find the button with findViewById.
- Set an OnClickListener to trigger adding a new contact.

Checkpoint: Clicking the button shows a Toast (temporary test).

## Step 4: Request WRITE\_CONTACTS Permission

- Use ContextCompat.checkSelfPermission and ActivityCompat.requestPermissions.
- Request code: 200.

- Handle result in onRequestPermissionsResult.

Checkpoint: App asks for write permission when button is clicked.

### Step 5: Insert a Contact

- Use ContentProviderOperation with applyBatch.

- Insert:

- RawContact

- Display Name

- Phone Number

- Example: "Demo User", "9876543210".

Checkpoint: Contact appears in emulator's Contacts app.

### Step 6: Refresh RecyclerView

- After adding a contact, call loadContacts() again.
- RecyclerView updates with the new entry.

Checkpoint: Newly added contact shows up in the list.

### Step 7: Test on Emulator

- Add a few contacts manually in the emulator.

- Run the app:

- Verify read contacts list.

- Click Add Contact    new contact appears.

- Check Contacts app to confirm.

### Step 8: Extensions (Optional)

- Replace hardcoded contact with an input form (EditText for name & phone).
- Add validation (non-empty fields).
- Allow multiple contact types (mobile, home, work).

## Deliverables

- Working app with both read and write functionality.
- Screenshot of RecyclerView before and after adding a contact.
- Short reflection: “What did you learn about runtime permissions and ContentProviderOperation?”

This lab sheet builds directly on their existing project, so trainees can incrementally add write functionality without starting from scratch.