Tutorial 5: RecyclerView

Objectives

- Practicing with `RecyclerView`
- Using communication internal services (telephony, SMS, email).

  Don’t forget to review the others with provided code examples:
  - Audio (record & playback), Video (record & playback), Image (capture & display)
  - Sensors
- Practice using `startActivityForResult()` & `setResult()`

Resources

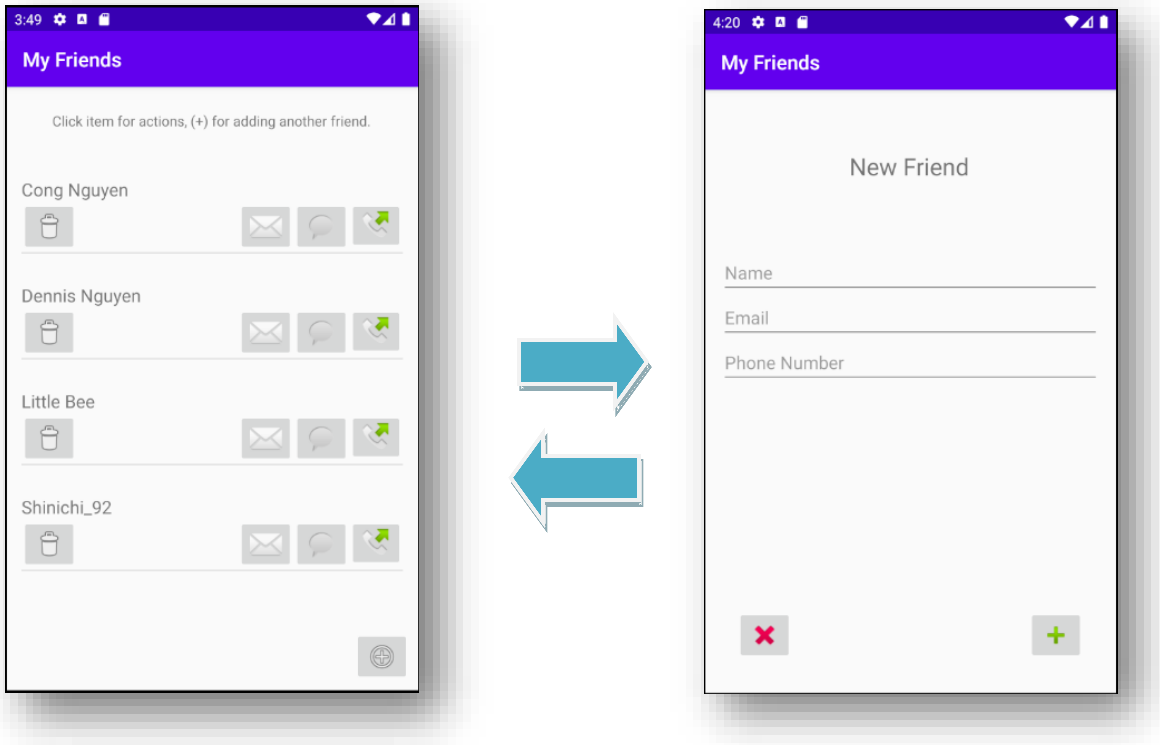
- Tutorial instruction (this file)
- *[Optional]* Lecture slides & Lecture code example

Tutorial Exercises

In this tutorial, we will create an application named “**My Friends**”, containing 02 activities. This application also uses activities from communication services for completed functioning.

Activities:

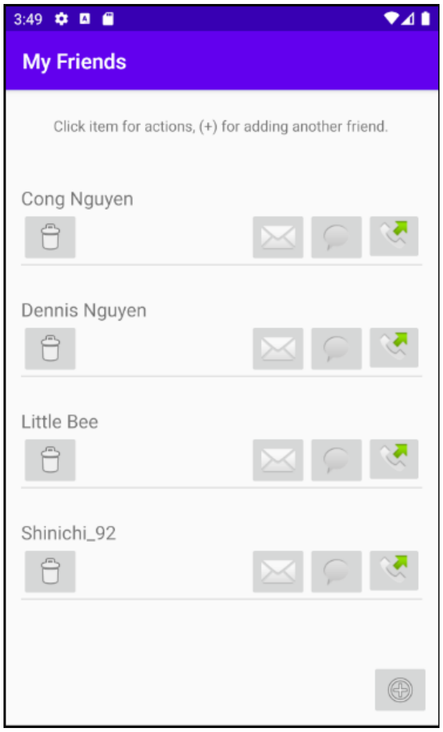
- (1) Activity: Friend List: to display a list of friends, click on item for communicating actions including
  - Send email
  - Send SMS
  - Make a phone call
- (2) Activity: Add new Friend: to get information for new friend



Click item for actions, (+) for adding another friend.

Exercise 1 – `MainActivity`: Friend list

- UI as below:



Click item for actions, (+) for adding another friend.

- Functionality
  - User clicks on item for communicating actions (send mail, SMS, make a phone call)
  - User clicks button “**Delete**” to remove item from friend list
  - User clicks button “**Add**” to add a new friend, system navigates user to `AddFriendActivity` for getting data

Task 1: Layout with `RecyclerView`

- Use `RecyclerView` to display a list of friends.
- Use some demo data for testing

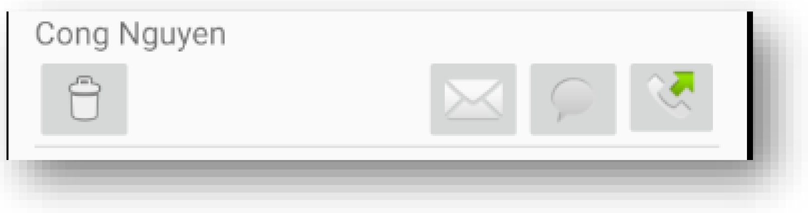
Note: don’t forget to set the `LayoutManager.`

Task 2: Creating the model: `Friend`

- Create a new package for model classes named **`/models`**
- Create a new model class **`models/Friend`** containing these attributes, namely `name, email, phoneNo`. You may extend as you want, for example, `title, organization`…

Task 3: Creating view for item: `item_friend.xml`

- Create a layout file **`/res/layout/item_friend.xml`**
- Layout components as below



Task 4: Create the Adapter & ViewHolder

- Create a new package for adapter classes named **`/adapters`**
- Create a new adapter class **`/adapters/FriendAdapter`** extends `RecyclerView.Adapter`
- Create an inner view holder class **`/adapters/FriendAdapter.FriendHolder`** extends `RecyclerView.ViewHolder`

Task 5: Handle events on item

- `ViewHolder` contains inflated layout from `itemView`, so this is the best position for handling events related to item’s components.

Hint: just remind

- Telephony intent

```
Intent intent = new Intent(Intent.ACTION_DIAL);
intent.setData(Uri.parse("tel:035.8877.210"));
```

- SMS intent

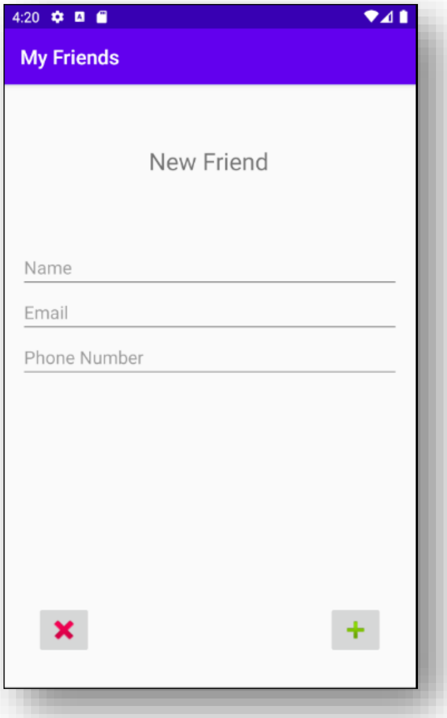
```
Intent intent = new Intent(Intent.ACTION_SENDTO);
intent.setData(Uri.parse("smsto:035.8877.210"));
```

- Email intent

```
Intent intent = new Intent(Intent.ACTION_SENDTO);
intent.setData(Uri.parse("mailto:"+Uri.encode("congnv@hanu.edu.vn")));
```

Exercise 2 – `AddFriendActivity`: add new Friend

- Create a new Activity named `AddFriendActivity` for getting friend’s information.
- UI as below



- Functionality
  - User inputs information & click “OK” to complete adding a new friend
  - User clicks “Cancel” to go back

Hint: just remind

- Set result & finish the current activity

```
setResult(RESULT_OK, intent);
finish();
```

Task 6: Handle adding event

- Start `AddFriendActivity` for a `Friend` object. In this case, we use

  `Intent.getSerializableExtra()`

  → Make sure that class `Friend` implement **`java.io.Serializable`**

Hint: just remind

- Start activity for result

```
Intent intent = new Intent(MainActivity.this, AnotherActivity.class);
startActivityForResult(intent, REQUEST_CODE);
```

- Handle activity result

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK && requestCode == REQUEST_CODE) {
        // TODO:
    }
}
```

Exercise 3 – Delete friend

- User clicks “Delete” to remove selected item.

Task 7: Handle deleting event

Hint: just remind

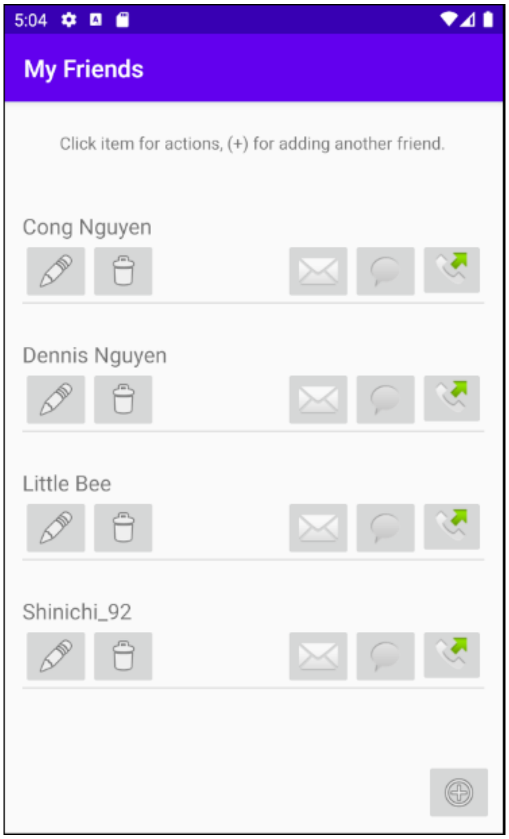
- Notify adapter about data changes

```
adapter.notifyDataSetChanged();
```

- You may use other suitable methods for optimization

[Optional] Exercise 4 – Edit an existing friend

- Create another activity for editing a selected friend



If you finish them all...

If you finish all the exercises, you can improve the apps above or your apps (with better input styles, images, functionality, handing exceptions…)

Posting them on our Facebook group with *#I'm_Android_developer* for discussion mark. **Note**: feel free if it somehow not so special – you know, we are all beginners :D

Good bye!