



BÁO CÁO THỰC HÀNH

Bài thực hành số 02: List View and Recycler View

Môn học: NT118 – Phát triển ứng dụng trên thiết bị di động

Lớp: NT118.Q11.2

Điểm tự đánh giá
10

THÀNH VIÊN THỰC HIỆN (Nhóm xx):

STT	Họ và tên	MSSV
1	Nguyễn Lê Nhật Đăng	23520231

ĐÁNH GIÁ KHÁC:

Tổng thời gian thực hiện	6 ngày (13/10/2025 – 18/10/2025)
Phân chia công việc	
Ý kiến (nếu có) + Khó khăn + Đề xuất, kiến nghị	Link chi tiết mã nguồn bài Lab: MobileApplicationDevelopment/LAB02 at main · tofumapu/MobileApplicationDevelopment

Phần bên dưới của báo cáo này là báo cáo chi tiết của n thực hiện



Mục lục

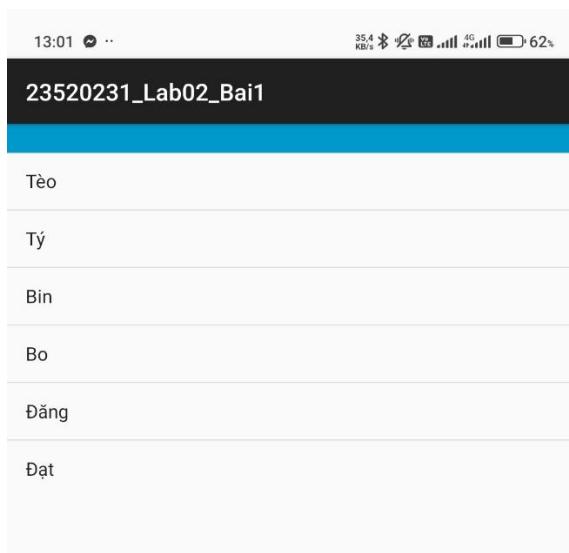
A. BÁO CÁO CHI TIẾT.....	2
1. Câu 1. Sử dụng ListView control với mảng dữ liệu định sẵn	2
a. Tổng quan.....	2
b. activity_main.xml	2
c. MainActivity.java.....	3
2. Câu 2. RelativeLayout.....	4
a. Tổng quan.....	4
b. activity_main.xml	5
c. MainActivity.java.....	6
3. Câu 3. Sử dụng ArrayList và ListView mà từng phần tử trong ArrayList là các Object bất kỳ	7
a. Tổng quan.....	7
b. activity_main.xml	8
c. MainActivity.java.....	10
4. Sử dụng CustomAdapter cho Listview	13
a. Tổng quan.....	13
b. activity_main.xml	14
c. item_employee.xml	14
d. Employee.java	15
e. EmployeeAdapter.java	15
f. MainActivity.java	16
5. Sử dụng GridView, Spinner	17
a. Tổng quan.....	17
b. Activity_main.xml.....	18
c. dish.xml, item_dish.xml, my_dropdown_item.xml và my_selected_item.xml	18
d. Dishes.java.....	20
e. DishesAdapter.java.....	21
f. SpinnerAdapter.java.....	21
g. MainActivity.java	22
6. RecyclerView	23
a. So sánh RecyclerView và ListView.....	23
b. Tổng quan.....	23
c. activity_main.xml	24
d. Item_employee.xml	25
e. Employee.java	25
f. EmployeeAdapter.java	25
g. MainActivity.java	27



A. BÁO CÁO CHI TIẾT

1. Câu 1. Sử dụng ListView control với mảng dữ liệu định sẵn

a. Tổng quan



Hình 1: Kết quả sau khi khởi động đoạn code



Hình 2: Kết quả sau khi chọn vào thành phần bất kì trong ListView

b. activity_main.xml

- Khai báo biến: tạo 2 thành phần bao gồm:

- TextView: Hiển thị tiêu đề bao gồm vị trí và giá trị của item trong ListView được chọn
- ListView: là danh sách hiển thị các item bao gồm nhiều dòng, mỗi dòng chứa giá trị tên duy nhất. Với cài đặt:



- android:choiceMode="singleChoice": chỉ cho phép chọn 1 item tại 1 thời điểm.
- android:listSelector="@color/blue": là thành phần chọn tới sẽ hiển thị sang màu xanh như kết quả trình bày.

c. MainActivity.java

- Tham chiếu lvPerson (ListView) và tvSelection (TextView) đến ListView và TextView tương ứng trong file activity_main.xml.

```
lvPerson = findViewById(R.id.lv_person);
tvSelection = findViewById(R.id.tv_selection);
```

Hình 3: Ánh xạ ListView và TextView

- Tạo một chuỗi chứa danh sách dữ liệu mẫu dạng tĩnh.

```
final String[] arr = {"Tèo", "Tý", "Bin", "Bo", "Đặng", "Đạt"};
```

Hình 4: Tạo mảng dữ liệu tĩnh

- Tạo Adapter có vai trò là cầu nối giữa dữ liệu (arr) và giao diện (ListView). Sử dụng layout mặc định (simple_list_item_1 – mỗi dòng là một dòng chữ được hiển thị).

```
ArrayAdapter<String> adapter = new ArrayAdapter<>(
    context: this,
    android.R.layout.simple_list_item_1,
    arr
);
```

Hình 5: Tạo Adapter

- Đưa dữ liệu vào giao diện, mỗi phần tử trong arr sẽ hiển thị thành 1 dòng trong ListView.

```
lvPerson.setAdapter(adapter);
```

Hình 6: Gắn Adapter vào ListView

- Khi người dùng chạm vào một phần tử trong ListView thì sẽ đưa lên TextView hiển thị giá trị vị trí (position) và giá trị (tên) trong ListView.

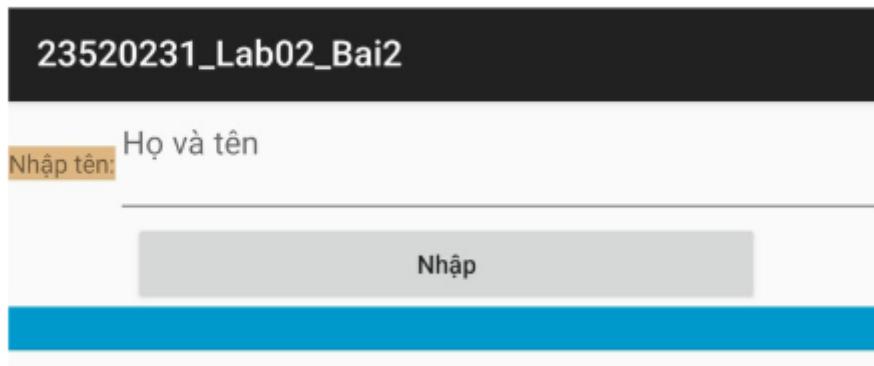
```
lvPerson.setOnItemClickListener(( AdapterView<?> parent, View view, int position, long id) ->
    tvSelection.setText("position: " + position + "; value: " + arr[position]));
```

Hình 7: Xử lý dữ liệu khi click vào phần tử

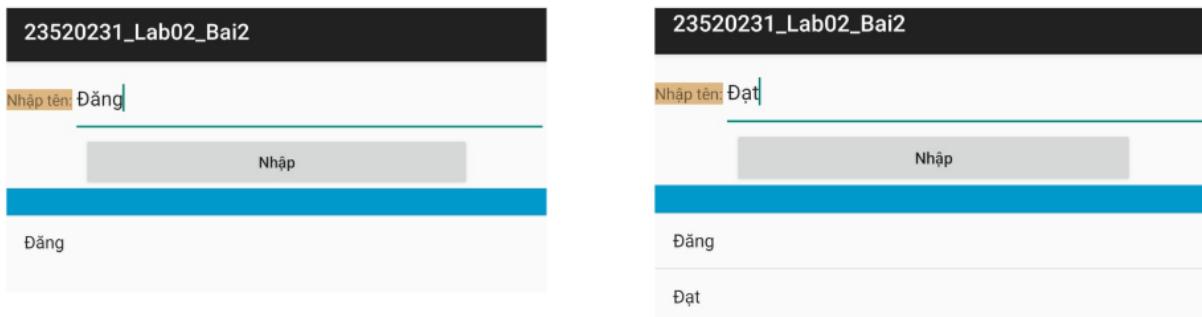


2. Câu 2. RelativeLayout

a. Tổng quan



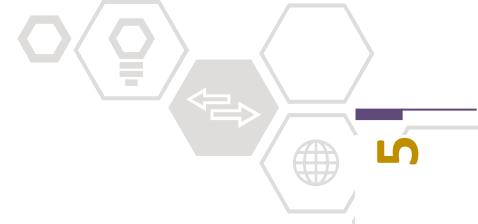
Hình 8: Kết quả sau khi thực hiện đoạn code



Hình 9: Kết quả sau khi thêm một phần tử



Hình 10: Kết quả sau khi chọn vào một phần tử (hiển thị vị trí và giá trị trên TextView)



Hình 11: Kết quả sau khi nhấn giữ vào một phần tử (Trong ví dụ sau khi nhấn giữ item "Đạt" -> xoá item đó)

b. activity_main.xml

```

<activity_main.xml>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    tools:context=".MainActivity">
    <LinearLayout
        android:id="@+id/child_linear_layout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/tv_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Nhập tên"
            android:background="#color/earth_brown"/>
        <EditText
            android:id="@+id/et_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Họ và tên ">
        </EditText>
    </LinearLayout>
    <androidx.appcompat.widget.AppCompatButton
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btn_input"
        android:layout_marginStart="70dp"
        android:layout_marginEnd="70dp"
        android:text="Nhập"
        android:textAllCaps="false"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </TextView>
</activity_main.xml>

```

Hình 12: Nội dung đoạn code đã thực hiện

- Phát triển giao diện từ bài 1, xây dựng thêm một LinearLayout (chứa TextView nhập tên và EditText nhập tên), một Button nhập tên người dùng đưa vào ListView.



- LinearLayout con
 - Có chiều rộng là match_parent và chiều dài là wrap_content.
 - Orientation là horizontal: Hiển thị tên và trường nhập tên
- TextView title: Hiển thị nhập tên (nằm trong LinearLayout con)
 - Có chiều rộng là wrap_parent và chiều dài là wrap_content.
 - Text là “Nhập tên”, khai báo trong string.xml.
 - Có background là màu hiển thị trong bài LAB.
- EditText name: Nhập tên (nằm trong LinearLayout con)
 - Có chiều rộng là match_parent và chiều dài là wrap_content.
 - Hint là “Họ và tên”, khai báo trong string.xml.
- Button: Đưa tên của người dùng vào trong ListView
 - Có chiều rộng là match_parent và chiều dài là wrap_content.
 - Text là “Nhập”, khai báo trong string.xml.
 - Có margin ngăn cách giữa các thành phần.

c. MainActivity.java

- Khai báo và ánh xạ các thành phần trong MainActivity.java

```
ArrayList<String> names;
4 usages
ArrayAdapter<String> adapter;
3 usages
private EditText et_name;
2 usages
private Button btn_input;
4 usages
private ListView lv_person;
2 usages
private TextView tv_selection;
```

Hình 13: Khai báo các thành phần

```
et_name = findViewById(R.id.et_name);
btn_input = findViewById(R.id.btn_input);
lv_person = findViewById(R.id.lv_person);
tv_selection = findViewById(R.id.tv_selection);
```

Hình 14: Ánh xạ các thành phần

- Tạo mảng dữ liệu tĩnh, tạo Adapter, gắn Adapter vào ListView, hiển thị vị trí của item được chọn vào TextView. (phát triển từ LAB1)

Bài thực hành số 02: List View and Recycler View



```
lv_person.setAdapter(adapter);
btn_input.setOnClickListener( View v -> {
    names.add(et_name.getText().toString());
    adapter.notifyDataSetChanged();
    et_name.setText("");
});
lv_person.setOnItemTouchListener(( AdapterView<?> parent, View view, int pos, long id) ->
    tv_selection.setText("position: " + pos + "; value: " + names.get(pos)));
});
```

Hình 15: Tạo mảng dữ liệu tĩnh, tạo Adapter, gắn Adapter vào ListView, hiển thị vị trí của item vào TextView khi click vào item.

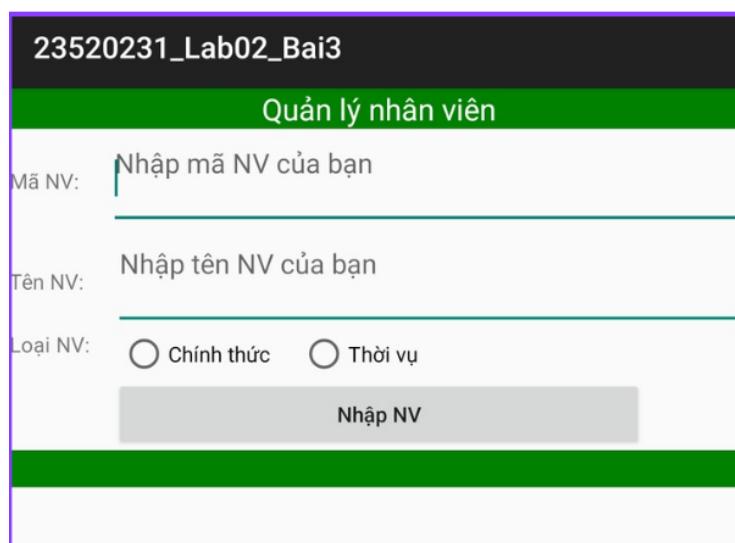
- Đặt một sự kiện khi nhấn giữ vào item trong ListView sẽ xoá phần tử đó khỏi danh sách và cập nhật lại danh sách và hiển thị lên trên màn hình.

```
lv_person.setOnItemLongClickListener(( AdapterView<?> parent, View view, int pos, long id) -> {
    names.remove(pos);
    adapter.notifyDataSetChanged();
    return true;
});
```

Hình 16: Xoá một item khi nhấn giữ vào phần tử đó trong ListView

3. Câu 3. Sử dụng ArrayList và ListView mà từng phần tử trong ArrayList là các Object bất kỳ

a. Tổng quan



Hình 17: Kết quả sau khi thực thi đoạn code

Bài thực hành số 02: List View and Recycler View



23520231_Lab02_Bai3	
Quản lý nhân viên	
Mã NV:	<input type="text" value="m1"/>
Tên NV:	Nguyen Thi Teo
Loại NV:	<input checked="" type="radio"/> Chính thức <input type="radio"/> Thời vụ
<input type="button" value="Nhập NV"/>	
23520231_Lab02_Bai3	
Quản lý nhân viên	
Mã NV:	Nhập mã NV của bạn
Tên NV:	Nhập tên NV của bạn
Loại NV:	<input type="radio"/> Chính thức <input checked="" type="radio"/> Thời vụ
<input type="button" value="Nhập NV"/>	
m1 - Nguyen Thi Teo --> FullTime: 500.0	

Hình 18: Kết quả tiến hành nhập 1 NV

23520231_Lab02_Bai3	
Quản lý nhân viên	
Mã NV:	<input type="text" value="m4"/>
Tên NV:	Nguyen thi ngoc uyen
Loại NV:	<input type="radio"/> Chính thức <input checked="" type="radio"/> Thời vụ
<input type="button" value="Nhập NV"/>	
m1 - Nguyen Thi Teo --> FullTime: 500.0	
m2 - Tran Van Ty --> PartTime: 150.0	
m3 - Ho DO --> FullTime: 500.0	
23520231_Lab02_Bai3	
Quản lý nhân viên	
Mã NV:	Nhập mã NV của bạn
Tên NV:	Nhập tên NV của bạn
Loại NV:	<input type="radio"/> Chính thức <input checked="" type="radio"/> Thời vụ
<input type="button" value="Nhập NV"/>	
m1 - Nguyen Thi Teo --> FullTime: 500.0	
m2 - Tran Van Ty --> PartTime: 150.0	
m3 - Ho DO --> FullTime: 500.0	
m4 - Nguyen thi ngoc uyen --> PartTime: 150.0	

Hình 19: Kết quả nhập một danh sách nhân viên

b. activity_main.xml

- Giao diện được phát triển từ bài 2, các View được thêm là các TextView hiển thị các trường (Mã nhân viên, Tên nhân viên, Quản lý nhân viên), các EditText (nhập mã nhân viên, tên nhân viên), các RadioButton (cho biết loại nhân viên đó làm chính thức hay thời vụ).

Bài thực hành số 02: List View and Recycler View



XML code for activity_main.xml:

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Quản lý nhân viên"  
    android:textAlignment="center"  
    android:textColor="@color/white"  
    android:background="@color/green"  
    android:textSize="20sp">  
</TextView>  
<LinearLayout  
    android:id="@+id/layoutMaNV"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal">  
    <TextView  
        android:id="@+id/tv_MaNV"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Mã NV: />  
    <EditText  
        android:id="@+id/et_MaNV"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginStart="20dp"  
        android:hint="Nhập mã NV của bạn ">  
    </EditText>  
</LinearLayout>
```

Annotations in the screenshot:

- TextView title (points to the title text in the XML)
- LinearLayout nhập MãNV (points to the LinearLayout containing the Mã NV TextView and EditText)
- TextView nhập mã NV (points to the Mã NV TextView)
- EditText nhập mã NV (points to the Mã NV EditText)

Hình 20: Thiết kế giao diện activity_main.xml (P1)

XML code for activity_main.xml:

```
28 23520231_Lab02_Bai3 Version control小米 220117TG app Component Tree  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" ① 2 ⑨ ^>  
35     </LinearLayout>  
36     <LinearLayout  
37         android:id="@+id/layoutTenNV"  
38         android:layout_width="match_parent"  
39         android:layout_height="wrap_content"  
40         android:orientation="horizontal">  
41             <TextView  
42                 android:id="@+id/tv_TenNV"  
43                 android:layout_width="wrap_content"  
44                 android:layout_height="wrap_content"  
45                 android:text="Tên NV: />  
46             <EditText  
47                 android:id="@+id/et_TenNV"  
48                 android:layout_width="match_parent"  
49                 android:layout_height="wrap_content"  
50                 android:layout_marginStart="20dp"  
51                 android:hint="Nhập tên NV của bạn ">  
52             </EditText>  
53         </LinearLayout>
```

Annotations in the screenshot:

- LinearLayout nhập TênNV (points to the LinearLayout containing the Tên NV TextView and EditText)
- TextView TênNV (points to the Tên NV TextView)
- EditText nhập TênNV (points to the Tên NV EditText)

Hình 21: Thiết kế giao diện activity_main.xml (P2)

Bài thực hành số 02: List View and Recycler View



10

The screenshot shows the Android Studio interface with the activity_main.xml file open. The code includes:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" ...>
    <LinearLayout
        android:id="@+id/LoaiNV"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/tv_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Loại NV:"/>
        <RadioGroup
            android:id="@+id/rg_type"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <RadioButton
                android:id="@+id/rd_chinhthuc"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:marginStart="20dp"
                android:text="Chính thức" />
            <RadioButton
                android:id="@+id/rd_thoivu"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="20dp"
                android:text="Thời vụ" />
        </RadioGroup>
    </LinearLayout>
```

The preview window shows a green header "Quản lý nhân viên". Below it are input fields for "Mã NV:" and "Tên NV:". A radio button group labeled "Loại NV:" contains two options: "Chính thức" and "Thời vụ". Below this is a list view with items: Item 1, Sub item 1; Item 2, Sub item 2; Item 3, Sub item 3; Item 4, Sub item 4; Item 5, Sub item 5; Item 6, Sub item 6; and Item 7, Sub item 7.

Hình 22: Thiết kế giao diện activity_main.xml (P3)

- Các TextView:

- Có chiều rộng là wrap_content, chiều cao là wrap_content.
- Nội dung là “Mã NV:”, “Tên NV:”, “Loại NV” khai báo trong string.xml.
- Nằm trong LinearLayout theo chiều ngang

- Các EditText:

- Có chiều rộng là match_parent, chiều cao là wrap_content.
- Nội dung hint là “Nhập mã NV của bạn”, “Nhập tên NV của bạn”, được khai báo trong string.xml.
- Nằm trong LinearLayout theo chiều ngang.
- Sử dụng margin để cách đều.

- Các RadioButton

- Có chiều rộng là wrap_content, chiều cao là wrap_content.
- Nội dung hint là “Chính thức”, “Thời vụ”, được khai báo trong string.xml.
- Nằm trong RadioGroup.
- Sử dụng margin để cách đều.

c. MainActivity.java

- Dựa vào sơ đồ thiết kế ở bài LAB, ta xây dựng Class cha là Employee và 2 Class con là EmployeeFullTime và EmployeePartTime

Bài thực hành số 02: List View and Recycler View



11

```
activity_main.xml colors.xml MainActivity.java Employee.java EmployeeFullTime.java EmployeePartTime.java values\themes.xml night\the...  
1 package com.example.a23520231_lab02_bai3;  
2  
3 public class Employee {  
4     private String id;  
5     private String name;  
6     public double tinhLuong() { return 0; }  
7  
8     // getters & setters  
9     public String getId() { return id; }  
10    public void setId(String id) { this.id = id; }  
11    public String getName() { return name; }  
12    public void setName(String name) { this.name = name; }  
13  
14    @Override  
15    public String toString() {  
16        return id + " - " + name;  
17    }  
18}
```

Hình 23: Class Employee (Khai báo biến, các phương thức getter, setter, hàm tính lương)

```
activity_main.xml colors.xml MainActivity.java Employee.java EmployeeFullTime.java EmployeePartTime.java values\themes.xml night\the...  
1 package com.example.a23520231_lab02_bai3;  
2  
3 public class EmployeeFullTime extends Employee {  
4     @Override  
5     public double tinhLuong() { return 500; }  
6     @Override  
7     public String toString() {  
8         return super.toString() + " --> FullTime: " + tinhLuong();  
9     }  
10 }  
11
```

Hình 24: Class con EmployeeFullTime hàm tinhLuong trả về giá trị 500 và hàm trả về chuỗi --> FullTime

```
activity_main.xml colors.xml MainActivity.java Employee.java EmployeeFullTime.java EmployeePartTime.java values\themes.xml night\the...  
1 package com.example.a23520231_lab02_bai3;  
2  
3 public class EmployeePartTime extends Employee {  
4     @Override  
5     public double tinhLuong() { return 150; }  
6     @Override  
7     public String toString() {  
8         return super.toString() + " --> PartTime: " + tinhLuong();  
9     }  
10 }  
11
```

Hình 25: Class con EmployeePartTime hàm tinhLuong trả về giá trị 150 và hàm trả về chuỗi --> PartTime



```
// Ánh xạ view
etId = findViewById(R.id.et_MaNV);
etName = findViewById(R.id.et_TenNV);
rgType = findViewById(R.id.rg_type);
btnAdd = findViewById(R.id.btn_input);
lvEmployee = findViewById(R.id.lv_person);

// Khởi tạo danh sách + adapter
employees = new ArrayList<>();
adapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1, employees);
lvEmployee.setAdapter(adapter);

// Bắt sự kiện nút "Nhập" → gọi hàm addNewEmployee()
btnAdd.setOnClickListener( View v -> addNewEmployee());
```

Hình 26: Khởi tạo biến, ánh xạ view, khởi tạo danh sách và adapter và bắt sự kiện nhấn nút từ bài 1

- Thêm hàm public void addNewEmployee từ bài hướng dẫn LAB, cụ thể, hàm sẽ lấy Id của RadioButton được check, lấy thông tin của Mã NV và Tên NV, sau đó gọi hàm EmployeeFullTime hoặc EmployeePartTime để tạo instance giá trị, gọi hàm setter để set giá trị mã NV và tên NV, thêm employee vào ArrayList (ArrayList chứa các Employee, đã khai báo class ở trên). Và cập nhật lại giao diện, reset ô nhập dữ liệu.

```
public void addNewEmployee() {
    // Lấy ra id của Radio Button được checked
    int radId = rgType.getCheckedRadioButtonId();
    String id = etId.getText().toString();
    String name = etName.getText().toString();

    if (radId == R.id.rd_chinhhuc) {
        // tạo instance là FullTime
        employee = new EmployeeFullTime();
    } else {
        // Tạo instance là PartTime
        employee = new EmployeePartTime();
    }

    // FullTime hay PartTime thi cũng là Employee nên có các hàm này là hiển nhiên
    employee.setId(id);
    employee.setName(name);

    // Đưa employee vào ArrayList
    employees.add(employee);

    // Cập nhật giao diện
    adapter.notifyDataSetChanged();

    // Reset lại ô nhập
    etId.setText("");
    etName.setText("");
    rgType.clearCheck();
}
```

Hình 27: Hàm public void addNewEmployee()

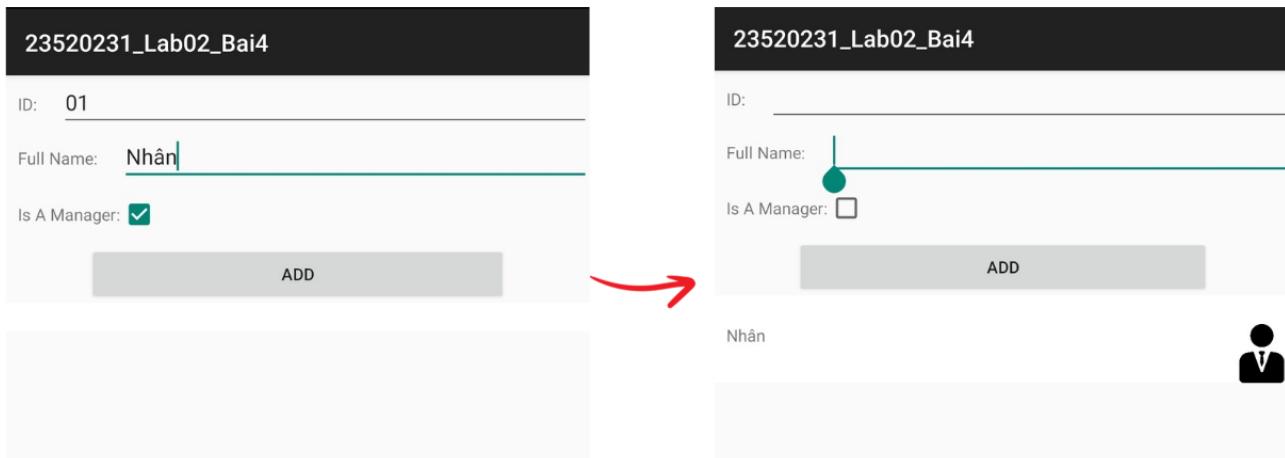


4. Sử dụng CustomAdapter cho Listview

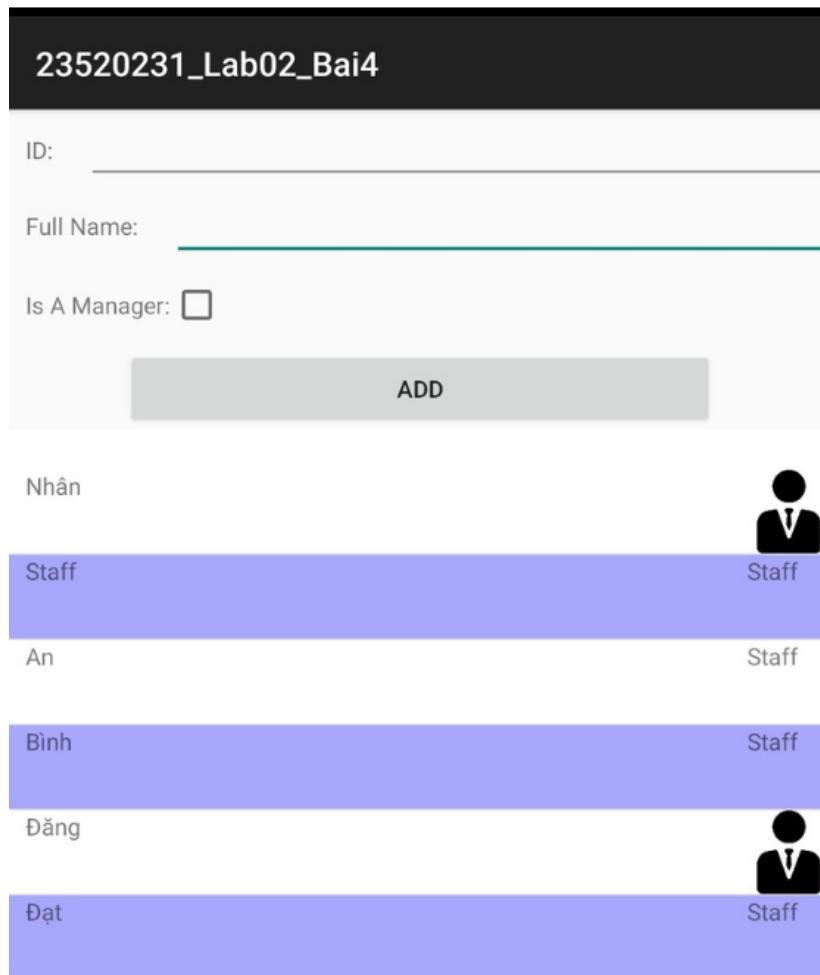
a. Tổng quan



Hình 28: Giao diện tổng quan khi chạy code



Hình 29: Kết quả khi thêm nhân viên thứ nhất



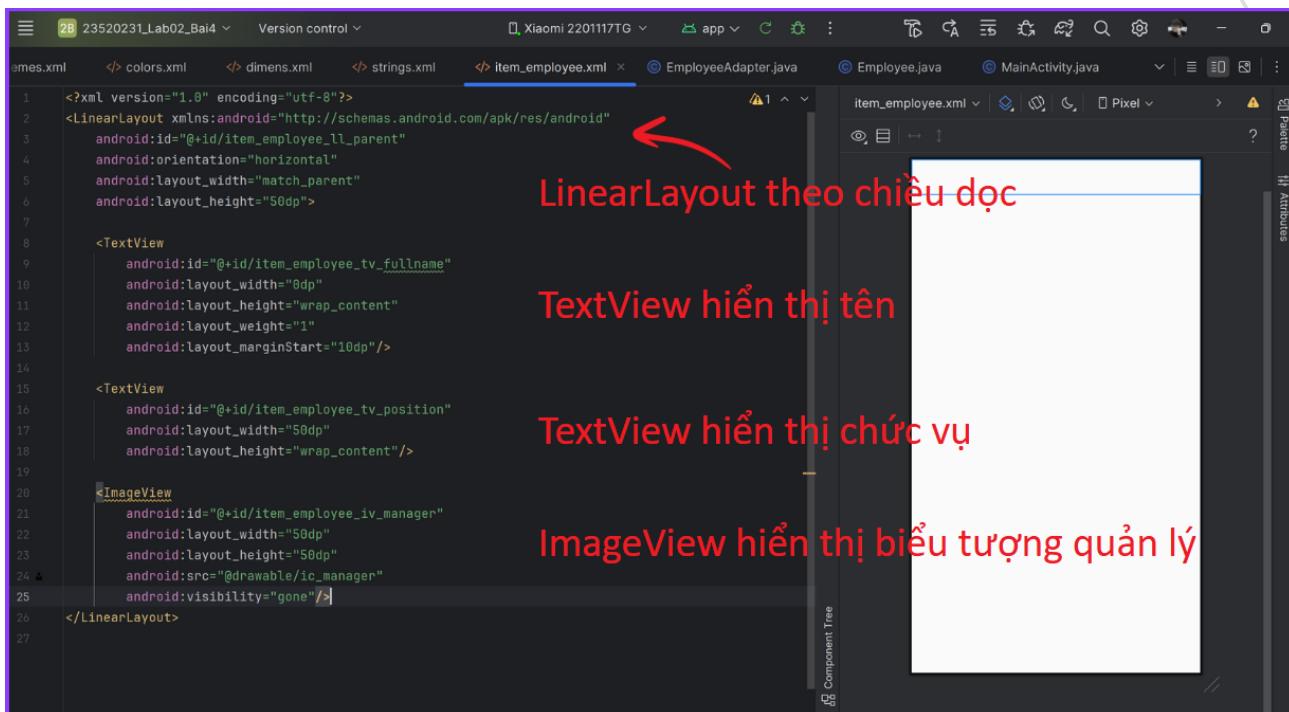
Hình 30: Kết quả khi thêm một list các nhân viên

b. activity_main.xml

- Được phát triển từ bài 3, sử dụng thêm thành phần View checkBox để đánh dấu nhân viên đó có phải là quản lý không.

c. item_employee.xml

- Thiết kế giao diện một item là một LinearLayout hiển thị theo chiều ngang. Nội dung hiển thị bao gồm TextView hiển thị tên nhân viên, TextView hiển thị chức vụ, ImageView hiển thị biểu thị biểu tượng Manager.



Hình 31: Phân tích thành phần trong item_employee.xml

d. Employee.java

- Khai báo class Employee và các hàm getter, setter cần thiết, hàm trả về kiểu dữ liệu boolean cho biết nhân viên đó có phải quản lý hay không.

```

public class Employee {
    public Employee(String id, String fullName, boolean isManager) {
        Id = id;
        FullName = fullName;
        this.isManager = isManager;
    }
    no usages
    public Employee() { }

    no usages
    public String getId() {
        return Id;
    }
    no usages
    public void setId(String id) {
        Id = id;
    }
    2 usages
    public String getFullName() {
        return FullName;
    }
    no usages
    public void setFullName(String fullName) {
        FullName = fullName;
    }
    1 usage
    public boolean isManager() {
        return isManager;
    }
    no usages
}

```

Hình 32: Class Employee

e. EmployeeAdapter.java

- Đoạn code LAB cung cấp dùng khai báo lớp và các constructor cần thiết, phương thức getView(), Inflate Layout tạo ra các đối tượng trong ListView với mỗi đối tượng là một đối tượng giống như đã được khai báo Item_employee.xml. Hàm lấy dữ liệu và ánh



xạ View. Hiển thị tên và chức vụ, hiển thị icon manager nếu nhân viên đó là quản lý và set xen kẽ màu nền.

```

14  public class EmployeeAdapter extends ArrayAdapter<Employee> {
15      ...
16
17      public View getView(final int position, View convertView, ViewGroup parent) {
18          ...
19
20          Employee employee = getItem(position);
21
22          TextView tvFullName = convertView.findViewById(R.id.item_employee_tv_fullname);
23          TextView tvPosition = convertView.findViewById(R.id.item_employee_tv_position);
24          ImageView ivManager = convertView.findViewById(R.id.item_employee_iv_manager);
25          LinearLayout llParent = convertView.findViewById(R.id.item_employee_ll_parent);
26
27          // Set full name
28          if (employee != null && employee.getFullName() != null) {
29              tvFullName.setText(employee.getFullName());
30          } else {
31              tvFullName.setText("");
32          }
33
34          // Manager icon / Staff text
35          if (employee != null && employee.isManager()) {
36              ivManager.setVisibility(View.VISIBLE);
37              tvPosition.setVisibility(View.GONE);
38          } else {
39              ivManager.setVisibility(View.GONE);
40              tvPosition.setVisibility(View.VISIBLE);
41              tvPosition.setText(context.getString(R.string.staff));
42          }
43
44          // Alternate background color
45          llParent.setBackgroundResource(position % 2 == 0 ? R.color.white : R.color.light_blue);
46
47      }
48
49  }

```

Hình 33: EmployeeAdapter.java

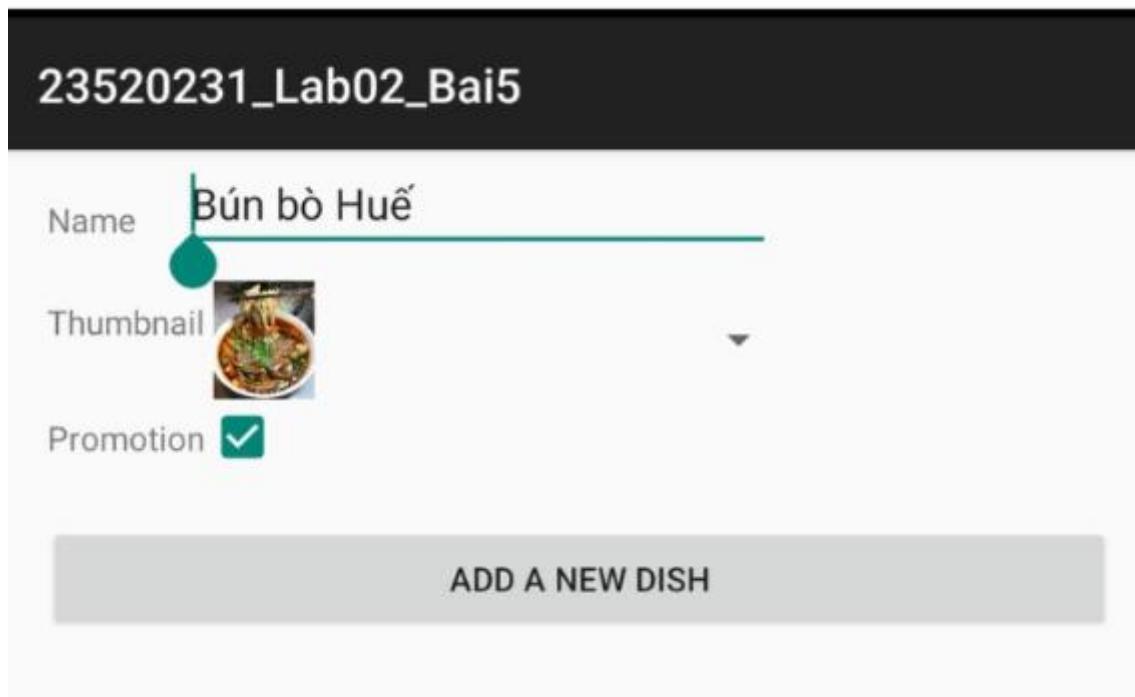
f. MainActivity.java

- Phát triển dựa trên các hàm đã được báo cáo chi tiết ở bài 3.

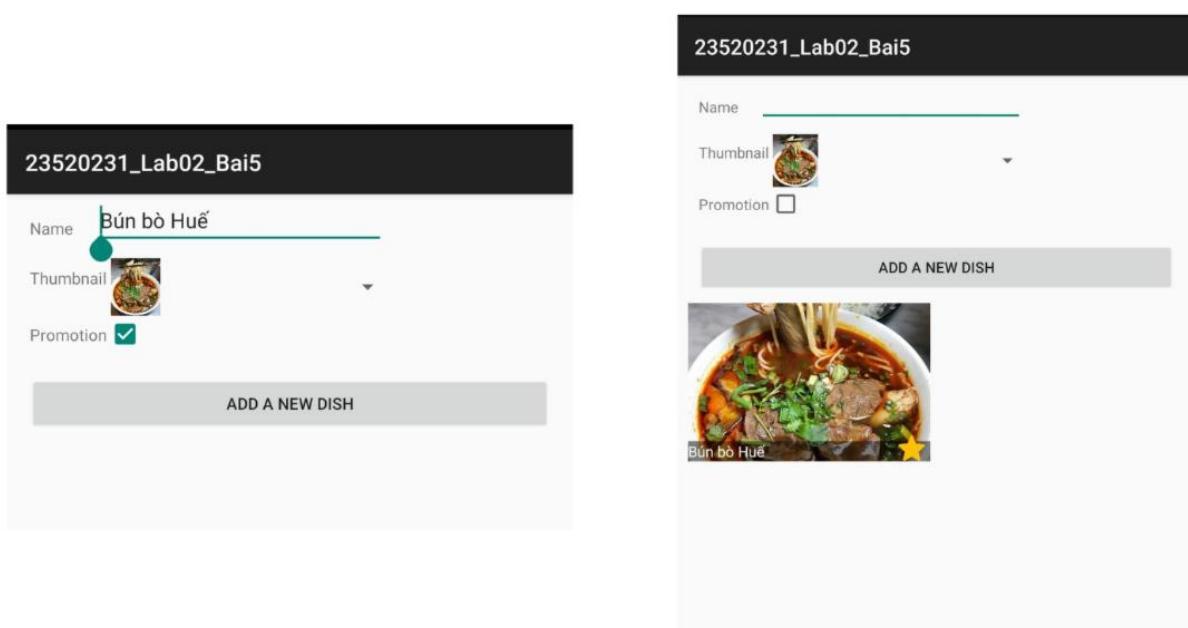


5. Sử dụng GridView, Spinner

a. Tổng quan



Hình 34: Kết quả khi thực hiện chạy đoạn code



Hình 35: Kết quả khi thêm một món ăn



b. Activity_main.xml

- Được phát triển từ bài 4, xây dựng giao diện để người dùng nhập vào món ăn (EditText), chọn ảnh đại diện món ăn (List Thumbnail) và chọn khuyến mãi (CheckBox) tiến hành nhập vào và hiển thị dưới dạng GridView (hiển thị dưới dạng xem 2 cột).

```

<activity_main.xml>
<ConstraintLayout>
    <CheckBox>
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toEndOf="@+id/tvPromotion"
        app:layout_constraintTop_toBottomOf="@+id/spinnerThumbnail"/>
    <Button
        android:id="@+id/btnAdd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ADD A NEW DISH"
        android:layout_margin="20dp"
        app:layout_constraintTop_toBottomOf="@+id/chkPromotion"/>
    <GridView
        android:id="@+id/gvDish"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:numColumns="2"
        android:layout_margin="10dp"
        android:horizontalSpacing="12dp"
        android:verticalSpacing="12dp"
        app:layout_constraintTop_toBottomOf="@+id/btnAdd"
        app:layout_constraintBottom_toBottomOf="parent"/>
</ConstraintLayout>

```

Sử dụng hiển thị dạng GridView với 2 cột và khoảng cách giữa các Item

Hình 36: activity_main.xml

c. dish.xml, item_dish.xml, my_dropdown_item.xml và my_selected_item.xml

- Đóng vài trò thiết kế layout từng phần tử trong GridView.

Bài thực hành số 02: List View and Recycler View



19

The screenshot shows the Android Studio interface with the dish.xml layout file open. The layout consists of three main components: an ImageView at the top, followed by a TextView, and another ImageView below it. The TextView contains placeholder text. The preview window on the right shows a dark blue background with white text and icons. The code editor highlights certain parts of the XML code.

```
<?xml version="1.0" encoding="utf-8"?>
<ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <ImageView
        android:id="@+id/imgDish"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:scaleType="centerCrop"/>
    <TextView
        android:id="@+id/tvDishName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#F80000"
        android:textColor="Android:color/white"
        android:singleLine="true"
        android:ellipsize="marquee"
        android:focused="true"
        android:focusableInTouchMode="true"
        android:scrollHorizontally="true"
        android:marqueeRepeatLimit="marquee_forever"
        app:layout_constraintBottom_toTopOf="@+id/imgStar"/>
    <ImageView
        android:id="@+id/imgStar"
        android:layout_width="25dp"
        android:layout_height="25dp"
        android:layout_marginEnd="5dp"
        android:visibility="gone"
        app:srcCompat="@drawable/star"
        app:layout_constraintEnd_toEndOf="@+id/imgDish"
        app:layout_constraintBottom_toBottomOf="@+id/imgDish"/>

```

Hình 37: dish.xml

The screenshot shows the Android Studio interface with two layout files open: item_dish.xml and my_selected_item.xml. item_dish.xml displays a simple layout with an ImageView and a TextView. my_selected_item.xml shows a list of four items, each with a thumbnail icon and a name: Thumbnail 1, Thumbnail 2, Thumbnail 3, and Thumbnail 4. The preview window on the right shows a list of these items.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp">
    <ImageView
        android:id="@+id/spinnerImage"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:scaleType="fitCenter"/>
    <TextView
        android:id="@+id/spinnerText"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_vertical"
        android:textSize="16sp"
        android:paddingStart="10dp"
        android:text="Thumbnail name"/>

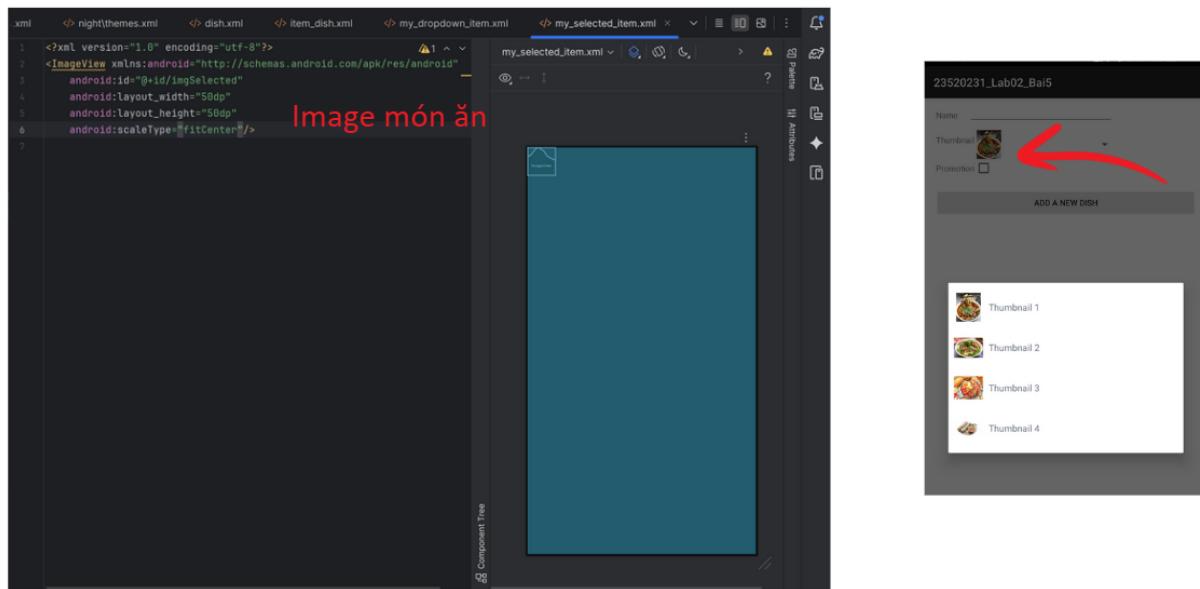
```

Hình 38: item_dish.xml và my_selected_item.xml là bản thiết kế cho giao diện chọn món ăn ở bên phải

Bài thực hành số 02: List View and Recycler View



20



Hình 39: `my_selected_item.xml` là bản thiết kế cho việc hiển thị thumbnail item đã chọn ở bên phải

d. Dishes.java

- Class khai báo các biến, các getter/setter và các hàm liên quan.

The screenshot shows the Android Studio Java editor with the file `Dishes.java` open. The code defines a class `Dishes` with private fields `name`, `thumbnail`, and `promotion`. It includes a constructor, getters, and setters for these fields.

```
public class Dishes {
    private String name;
    private int thumbnail;
    private boolean promotion;
    public Dishes() {}
    public Dishes(String name, int thumbnail, boolean promotion) {
        this.name = name;
        this.thumbnail = thumbnail;
        this.promotion = promotion;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getThumbnail() {
        return thumbnail;
    }
}
```

Hình 40: `Dishes.java`



e. DishesAdapter.java

- Xây dựng Adapter tùy chỉnh (Custom Adapter) dùng để hiển thị danh sách món ăn trong GridView. Mỗi item hiển thị ảnh (thumbnail), tên món (chạy chữ nếu dài), và biểu tượng ngôi sao khi có khuyến mãi.

```

13 public class DishesAdapter extends BaseAdapter {
14     ...
15     ...
16     ...
17     ...
18     ...
19     ...
20     ...
21     ...
22     ...
23     ...
24     ...
25     ...
26     ...
27     ...
28     ...
29     ...
30     ...
31     ...
32     ...
33     ...
34     ...
35     ...
36     ...
37     ...
38     ...
39     ...
40     ...
41     ...
42     ...
43     ...
44     ...
45     ...
46     ...
47     ...
48     ...
49     ...
50     ...
51     ...
52     ...
53     ...
54     ...
55     ...
56     ...
57     ...
58     ...
}

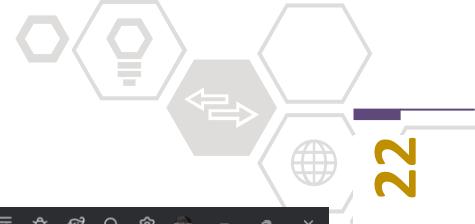
```

Hình 41: DishesAdapter.java

f. SpinnerAdapter.java

- Hiển thị danh sách thumbnail hình ảnh món ăn trong Spinner (ô chọn ảnh trong phần “Thumbnail”).
- Tùy chỉnh cách Spinner hiển thị ở hai trạng thái:
 - Khi Spinner đóng: chỉ hiển thị ảnh được chọn (my_selected_item.xml)
 - Khi Spinner mở: hiển thị danh sách các lựa chọn (ảnh + tên món) (my_dropdown_item.xml)

Bài thực hành số 02: List View and Recycler View



22

```
13 public class SpinnerAdapter extends BaseAdapter {
14 ...
15 ...
16 ...
17 ...
18 ...
19 ...
20 ...
21 ...
22 ...
23 ...
24 ...
25 ...
26 ...
27 ...
28 ...
29 ...
30 ...
31 ...
32 ...
33 ...
34 ...
35 ...
36 ...
37 ...
38 ...
39 ...
40 ...
41 ...
42 ...
43 ...
44 ...
45 ...
46 ...
47 ...
48 ...
49 ...
50 ...
51 ...
52 ...
53 ...
54 ...
55 ...
56 ...
57 ...
58 ...
59 ...
60 }
```

Hình 42: SpinnerAdapter.java

g. MainActivity.java

- Tái sử dụng từ bài 4. Đây là lớp điều khiển chính của ứng dụng, xử lý toàn bộ logic: nhập dữ liệu món ăn, chọn hình thumbnail, xác định khuyến mãi, và hiển thị danh sách ra GridView.

```
9 public class MainActivity extends AppCompatActivity {
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
18 ...
19 ...
20 ...
21 ...
22 ...
23 ...
24 ...
25 ...
26 ...
27 ...
28 ...
29 ...
30 ...
31 ...
32 ...
33 ...
34 ...
35 ...
36 ...
37 ...
38 ...
39 ...
40 ...
41 ...
42 ...
43 ...
44 ...
45 ...
46 ...
47 ...
48 ...
49 ...
50 ...
51 ...
52 ...
53 ...
54 ...
55 ...
56 ...
57 ...
58 ...
59 ...
60 ...
61 ...
62 ...
63 ...
64 ...
65 ...
66 ...
67 ...
68 ...
69 ...
70 ...
71 ...
72 ...
73 ...
74 ...
75 }
```

Hình 43: MainActivity.java



6. RecyclerView

a. So sánh RecyclerView và ListView

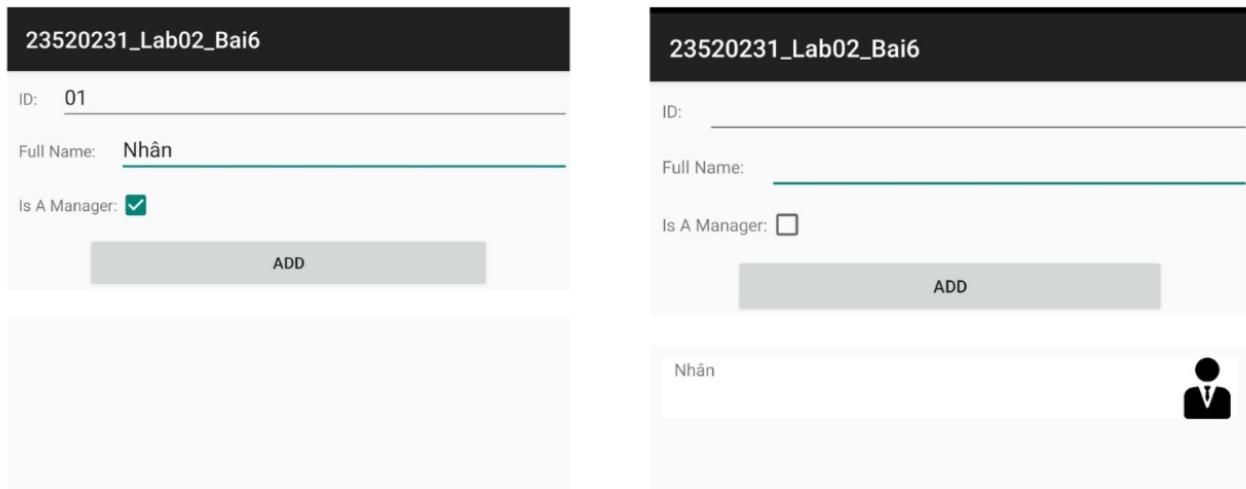
Thành phần	ListView	RecyclerView
Linh hoạt	Cố định dạng danh sách	ViewHolder tích hợp, linh hoạt hơn
Animation	Hạn chế	Hỗ trợ thêm/xoá/sửa 1 item dễ dàng
Adapter	ArrayAdapter hoặc BaseAdapter	RecyclerView.Adapter với ViewHolder bắt buộc

b. Tổng quan

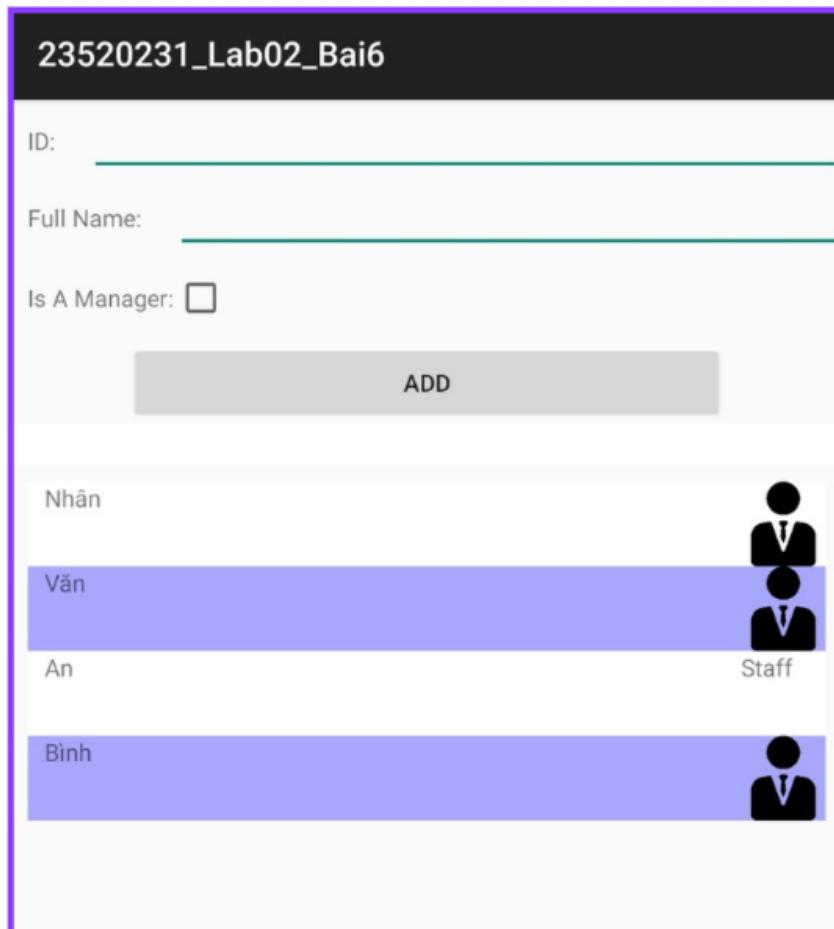
- Bài LAB phát triển từ bài 4, tuy nhiên có một số điểm khác nhau nhất định (sẽ trình bày chi tiết ở dưới) khi sử dụng RecyclerView.

The screenshot displays a mobile application's user interface. At the top, there is a dark header bar with the text "23520231_Lab02_Bai6". Below the header, the main content area has a white background. It contains three text input fields: "ID:" followed by a horizontal line for input, "Full Name:" followed by a horizontal line for input, and "Is A Manager:" followed by a checkbox. Below these inputs is a large, light-grey rectangular button with the word "ADD" centered in capital letters. The entire interface is framed by a thick purple border.

Hình 44: Tổng quan giao diện sau khi chạy đoạn code



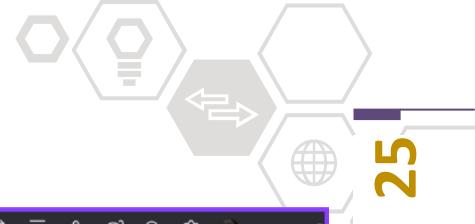
Hình 45: Kết quả sau khi thêm một nhân viên



Hình 46: Kết quả sau khi thêm toàn bộ nhân viên

c. activity_main.xml

- Được phát triển từ bài 4. Sử dụng RecyclerView thay thế ListView với các thuộc tính:
 - o Có chiều dài là match_parent, chiều rộng là 0dp (scale theo item trong RecyclerView), có weight là 1 (nghĩa là RecyclerView sẽ chiếm toàn bộ màn hình sau khi thêm các View ở trên).



```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    <LinearLayout
        <TextView
            android:layout_height="wrap_content"
            android:text="@string/is_manager"/>
        <CheckBox
            android:id="@+id/cb_IsManager"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </LinearLayout>
    <androidx.appcompat.widget.AppCompatButton
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/btn_input"
        android:layout_marginStart="70dp"
        android:layout_marginEnd="70dp"
        android:text="@string/input"
        android:textAllCaps="false"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/tv_selection"
        android:textColor="@color/black"
        android:background="@color/white"
        android:textSize="18dp"/>
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvEmployee"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:layout_margin="10dp"/>
</LinearLayout>

```

Thay bằng RecyclerView

Hình 47: activity_main.xml

d. Item_employee.xml

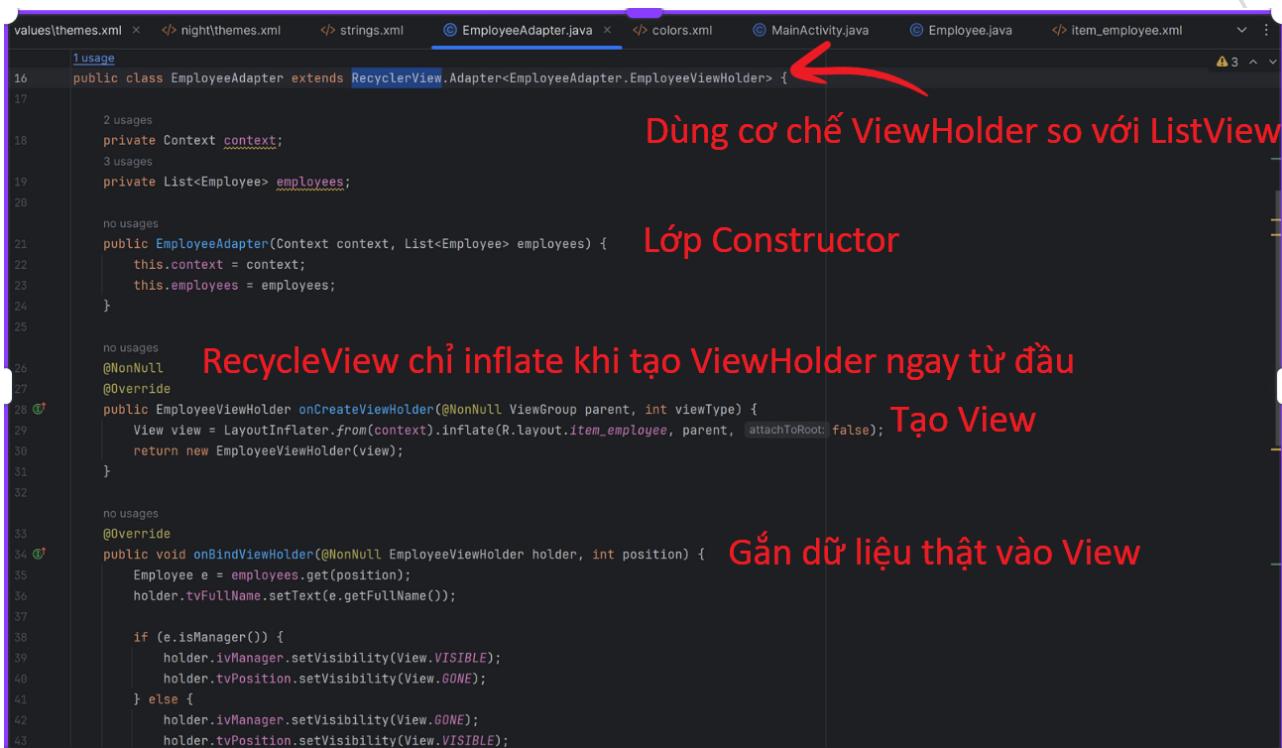
- Giữ nguyên, không thay đổi gì so với bài 4.

e. Employee.java

- Giữ nguyên, không thay đổi gì so với bài 4.

f. EmployeeAdapter.java

- Giải thích chi tiết sẽ được đính kèm trong hình.

```

values/themes.xml      nightthemes.xml      strings.xml      EmployeeAdapter.java      colors.xml      MainActivity.java      Employee.java      item_employee.xml
16  public class EmployeeAdapter extends RecyclerView.Adapter<EmployeeAdapter.EmployeeViewHolder> { ←
17
18     2 usages
19     private Context context;
20     3 usages
21     private List<Employee> employees;
22
23     no usages
24     public EmployeeAdapter(Context context, List<Employee> employees) {
25         this.context = context;
26         this.employees = employees;
27     }
28
29     no usages
30     @NonNull
31     public EmployeeViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
32         View view = LayoutInflater.from(context).inflate(R.layout.item_employee, parent, attachToRoot: false);
33         return new EmployeeViewHolder(view);
34     }
35
36     no usages
37     @Override
38     public void onBindViewHolder(@NonNull EmployeeViewHolder holder, int position) {
39         Employee e = employees.get(position);
40         holder.tvFullName.setText(e.getFullName());
41
42         if (e.isManager()) {
43             holder.ivManager.setVisibility(View.VISIBLE);
44             holder.tvPosition.setVisibility(View.GONE);
45         } else {
46             holder.ivManager.setVisibility(View.GONE);
47             holder.tvPosition.setVisibility(View.VISIBLE);
48         }
49     }
50
51     no usages
52     @Override
53     public int getItemCount() {
54         return employees.size();
55     }

```

Dùng cơ chế ViewHolder so với ListView

Lớp Constructor

RecyclerView chỉ inflate khi tạo ViewHolder ngay từ đầu

Tạo View

Gắn dữ liệu thật vào View

Hình 48: EmployeeAdapter.java (P1)

```

public void onBindViewHolder(@NonNull EmployeeViewHolder holder, int position) {
    Employee e = employees.get(position);
    holder.tvFullName.setText(e.getFullName()); ← Lấy vị trí và set tên người dùng nhập từ EditText

    if (e.isManager()) {
        holder.ivManager.setVisibility(View.VISIBLE);
        holder.tvPosition.setVisibility(View.GONE);
    } else {
        holder.ivManager.setVisibility(View.GONE);
        holder.tvPosition.setVisibility(View.VISIBLE);
        holder.tvPosition.setText("Staff");
    }

    holder.llParent.setBackgroundResource(
        position % 2 == 0 ? R.color.white : R.color.light_blue
    );
}

```

Nếu là nhân viên sẽ hiển thị Text là Staff
hoặc là icon nếu là Manager

Đếm số lượng nhân viên

Hiển thị 2 màu xen kẽ
giữa 2 item

Hình 49: EmployeeAdapter.java (P2)



```

public static class EmployeeViewHolder extends RecyclerView.ViewHolder {
    2 usages
    TextView tvFullName, tvPosition;
    3 usages
    ImageView ivManager;
    2 usages
    LinearLayout llParent;

    1 usage
    public EmployeeViewHolder(@NonNull View itemView) {
        super(itemView);
        tvFullName = itemView.findViewById(R.id.item_employee_tv_fullname);
        tvPosition = itemView.findViewById(R.id.item_employee_tv_position);
        ivManager = itemView.findViewById(R.id.item_employee_iv_manager);
        llParent = itemView.findViewById(R.id.item_employee_ll_parent);
    }
}

```

Khởi tạo biến

Ánh xạ

Hình 50: EmployeeAdapter.java (P3)

g. MainActivity.java

```

Project Alt+1 ml  <> night\themes.xml  <> strings.xml  EmployeeAdapter.java  <> colors.xml  MainActivity.java  Employee.java  <> item_employee.xml  :
15     public class MainActivity extends AppCompatActivity {
25         protected void onCreate(Bundle savedInstanceState) {
31             chkManager = findViewById(R.id.cb_ismanager);
32             btnAdd = findViewById(R.id.btn_input);
33             rvEmployee = findViewById(R.id.rvEmployee);

35             employees = new ArrayList<>();
36             adapter = new EmployeeAdapter( context: this, employees);
37             rvEmployee.setAdapter(adapter);
38             rvEmployee.setLayoutManager(new LinearLayoutManager( context: this));

40             btnAdd.setOnClickListener(new View.OnClickListener() {
41                 @Override
42                 public void onClick(View view) {
43                     String id = edtId.getText().toString();
44                     String name = edtName.getText().toString();
45                     boolean isManager = chkManager.isChecked();

46                     employees.add(new Employee(id, name, isManager));
47                     adapter.notifyItemInserted( position: employees.size() - 1);

48                     edtId.setText("");
49                     edtName.setText("");
50                     chkManager.setChecked(false);

51                 }
52             });
53         }
54     }
55 }

```

Hình 51: MainActivity.java

- employees = new ArrayList<>(); : Khởi tạo arraylist, mỗi item là một Employee (class được khai báo từ trước).
- adapter = new EmployeeAdapter(this, employees);: Truyền Employees vào làm context.
- rvEmployee.setAdapter(adapter); : Gắn Adapter vào RecyclerView để hiển thị dữ liệu.
- rvEmployee.setLayoutManager(new LinearLayoutManager(this));: Hiển thị danh sách theo chiều dọc.



- Sự kiện nút Add: Khi người dùng nhấn nút thêm nhân viên mới, chương trình lấy dữ liệu từ các trường EditText và CheckBox để kiểm tra nhân viên không. Tiến hành tạo Employee mới và thêm vào ArrayList. Sau đó tiến hành báo cho adapter biết sẽ có item mới. Cuối cùng, tiến hành xoá dữ liệu để người dùng nhập thông tin nhân viên mới.