



BÁO CÁO THỰC HÀNH

Bài thực hành số 02: ListView và RecyclerView

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

Lớp: NT118.P13.1

SINH VIÊN THỰC HIỆN :

STT	Họ và tên	MSSV
1	Nguyễn Hữu Thắng	22521334

ĐÁNH GIÁ KHÁC:

Tổng thời gian thực hiện	7 ngày
Phân chia công việc	
Ý kiến (nếu có) + Khó khăn + Đề xuất, kiến nghị	

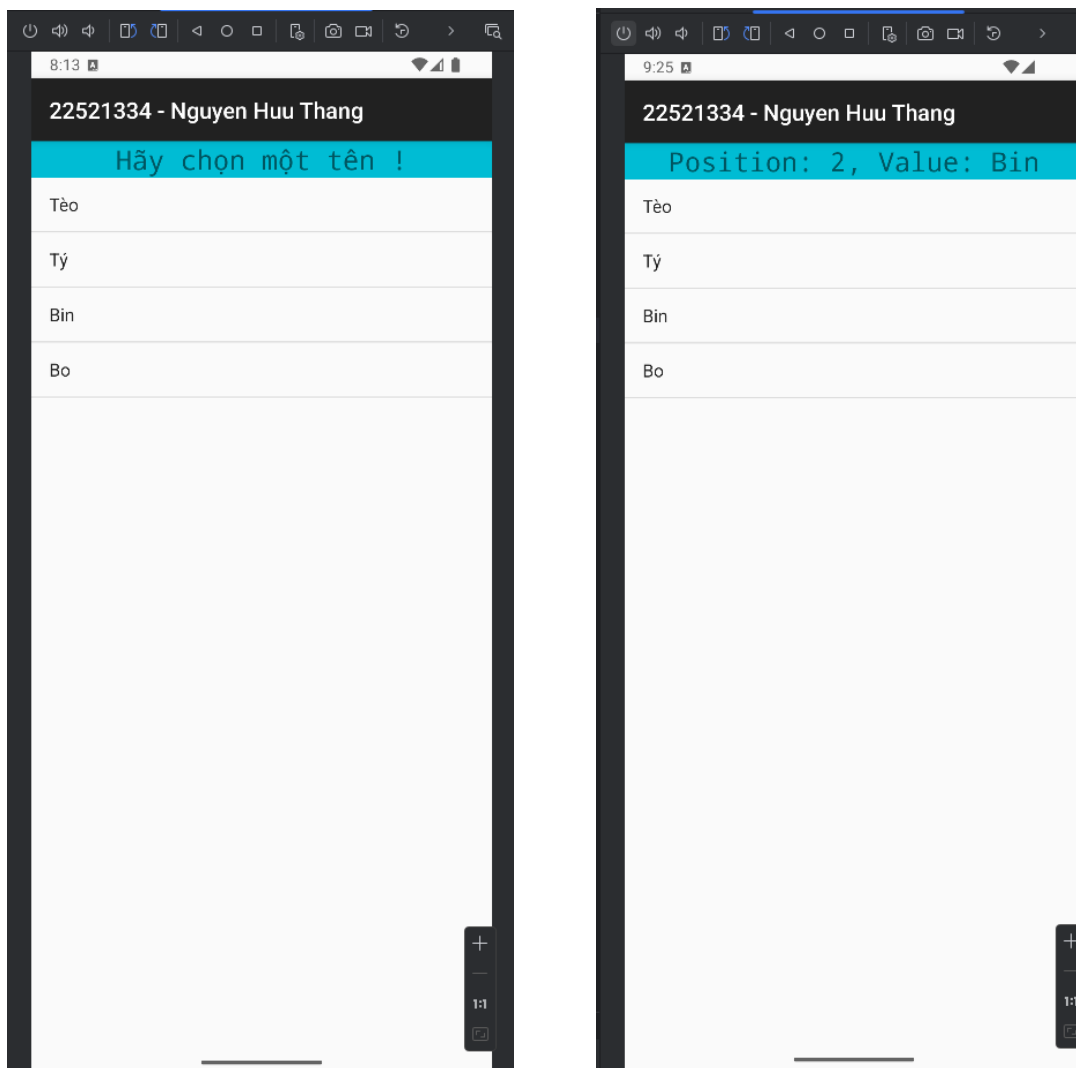
MỤC LỤC

A.	BÁO CÁO CHI TIẾT	3
1.	Sử dụng ListView control với mảng dữ liệu có sẵn.	3
a.	Hình ảnh minh chứng	3
b.	Mô tả	5
2.	Sử dụng ArrayList và ListView control.	6
a.	Hình ảnh minh chứng	6
b.	Mô tả	7
3.	Sử dụng ArrayList và ListView có phần tử trong ArrayList là các Object	8
a.	Hình ảnh minh chứng	8
b.	Mô tả	11
4.	Sử dụng CustomAdapter cho ListView	13
a.	Hình ảnh minh chứng	13
b.	Mô tả	16
5.	Sử dụng GridView, Spinner	17
a.	Hình ảnh minh chứng	17
b.	Mô tả	22
6.	RecyclerView	23
a.	Hình ảnh minh chứng	23
b.	Mô tả	27

A. BÁO CÁO CHI TIẾT

1. Sử dụng ListView control với mảng dữ liệu có sẵn.

a. Hình ảnh minh chứng



```

</> activity_main.xml ×
1  <?xml version="1.0" encoding="utf-8"?>
2  ©  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/main"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:orientation="vertical"
9      tools:context=".MainActivity">
10
11      <TextView
12          android:id="@+id/tv_selection"
13          android:layout_width="match_parent"
14          android:layout_height="wrap_content"
15          ■  android:background="#00BCD4"
16          android:fontFamily="monospace"
17          android:text="@string/h_y_ch_n_m_t_t_n"
18          android:textAlignment="center"
19          android:textSize="24sp" />
20
21      <ListView
22          android:id="@+id/lv_person"
23          android:layout_width="match_parent"
24          android:layout_height="match_parent" />
25  </LinearLayout>
    
```

```

18 <> public class MainActivity extends AppCompatActivity {
19     String[] lvPerson = {"Tèo", "Tý", "Bin", "Bo"}; 2 usages
20     ArrayAdapter<String> myAdapter; 2 usages
21     ListView lv_person; 3 usages
22     TextView tv_selection; 2 usages
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
27         setContentView(R.layout.activity_main);
28         //
29         tv_selection = findViewById(R.id.tv_selection);
30         lv_person = findViewById(R.id.lv_person);
31         myAdapter = new ArrayAdapter<>( context: MainActivity.this, android.R.layout.simple_list_item_1,lvPerson);
32         lv_person.setAdapter(myAdapter);
33         // Sự kiện khi click vào list view
34         lv_person.setOnItemClickListener(new AdapterView.OnItemClickListener() {
35             @Override
36             public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
37                 tv_selection.setText("Position: " +position+ ", Value: "+ lvPerson[position]);
38             }
39         });
40         // Title bar
41
42         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
43             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
44             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
45             return insets;
46         });
47     }
48 }

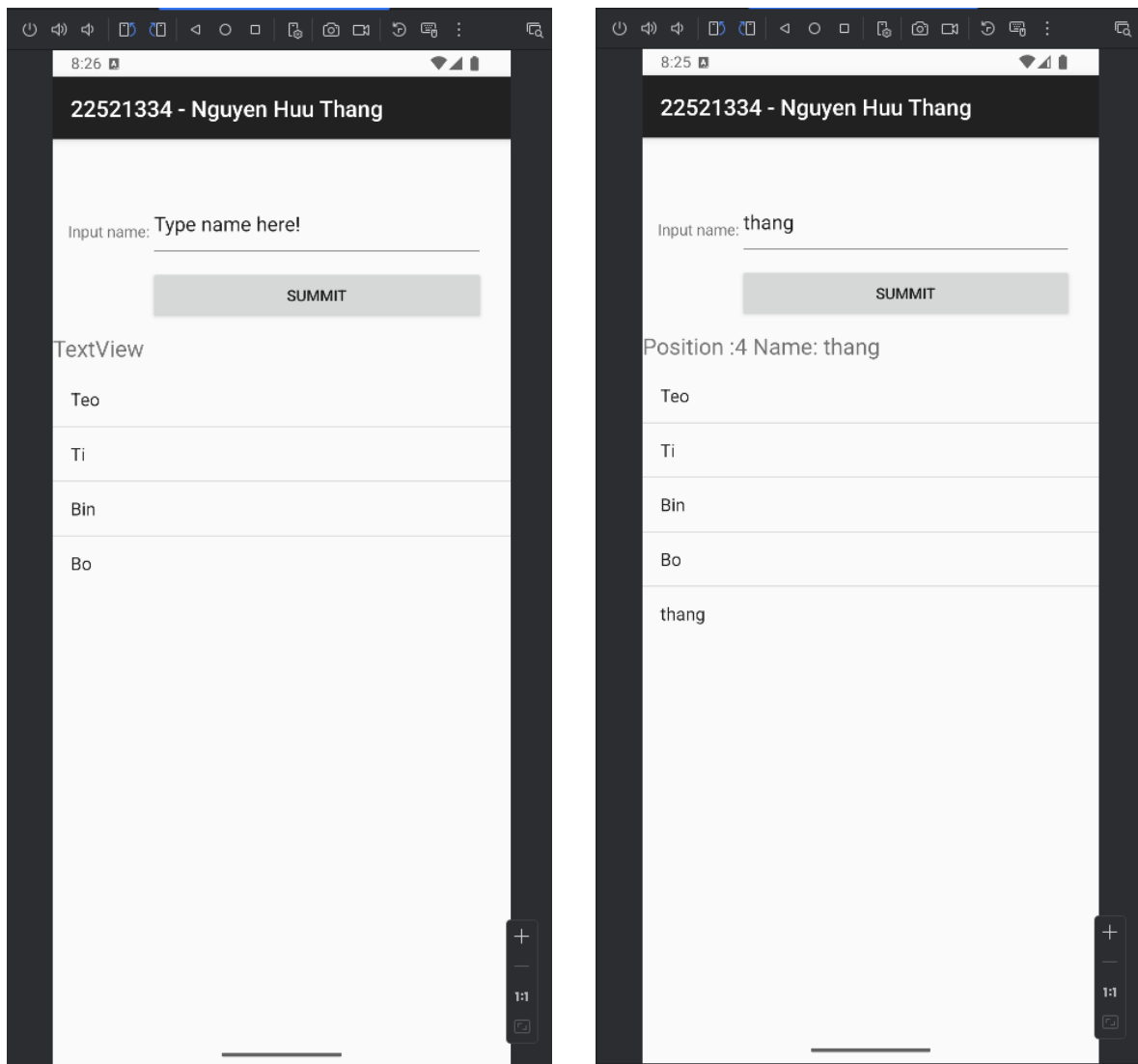
```

b. Mô tả

- ❖ Khai báo một mảng lvPerson chứa các tên ("Tèo", "Tý", "Bin", "Bo").
- ❖ Tạo ArrayAdapter kết nối mảng lvPerson với ListView (lv_person), hiển thị danh sách các tên trong giao diện.
- ❖ Thiết lập ArrayAdapter cho ListView và xử lý sự kiện khi người dùng click vào một mục trong danh sách, cập nhật TextView để hiển thị vị trí và giá trị được chọn.

2. Sử dụng ArrayList và ListView control.

a. Hình ảnh minh chứng



```

20 <> public class MainActivity extends AppCompatActivity {
21     // declare variable
22     ArrayAdapter<String> adapter; 4 usages
23     ListView lvPerson; 5 usages
24     TextView tvPerson; 2 usages
25     ArrayList<String> arrayName; 8 usages
26     Button btnSummit; 2 usages
27     EditText etName; 2 usages
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         EdgeToEdge.enable($this$enableEdgeToEdge: this);
32         setContentView(R.layout.activity_main);
33         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
34             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
35             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
36             return insets;
37         });
38         // bind view by id
39         tvPerson = findViewById(R.id.tvPerson);
40         lvPerson = findViewById(R.id.lvPerson);
41         btnSummit = findViewById(R.id.btnSummit);
42         etName = findViewById(R.id.etName);
43
44         // create arraylist
45         arrayName = new ArrayList<String>();
46         arrayName.add("Teo");
47         arrayName.add("Ti");
48         arrayName.add("Bin");
49         arrayName.add("Bo");
50
51         // create adapter and set adapter for listview
52         adapter = new ArrayAdapter<String>(context: this, android.R.layout.simple_list_item_1, arrayName);
53         lvPerson.setAdapter(adapter);
54
55         btnSummit.setOnClickListener(new View.OnClickListener() {
56             public void onClick(View view) {
57                 String ten = etName.getText().toString();
58                 arrayName.add(ten); // add item in listview
59                 adapter.notifyDataSetChanged(); // update listview
60             }
61         });
62         // bind event click
63         lvPerson.setOnItemClickListener(new AdapterView.OnItemClickListener() {
64             @Override
65             public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
66                 String name = lvPerson.getItemAtPosition(position).toString();
67                 tvPerson.setText("Position : " + position + " Name: " + name);
68             }
69         });
70         // bind event long click
71         lvPerson.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
72             @Override no usages
73             public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {
74                 arrayName.remove(position); // remove item in listview
75                 adapter.notifyDataSetChanged(); // update listview
76                 return false;
77             }
78         });
79     }

```

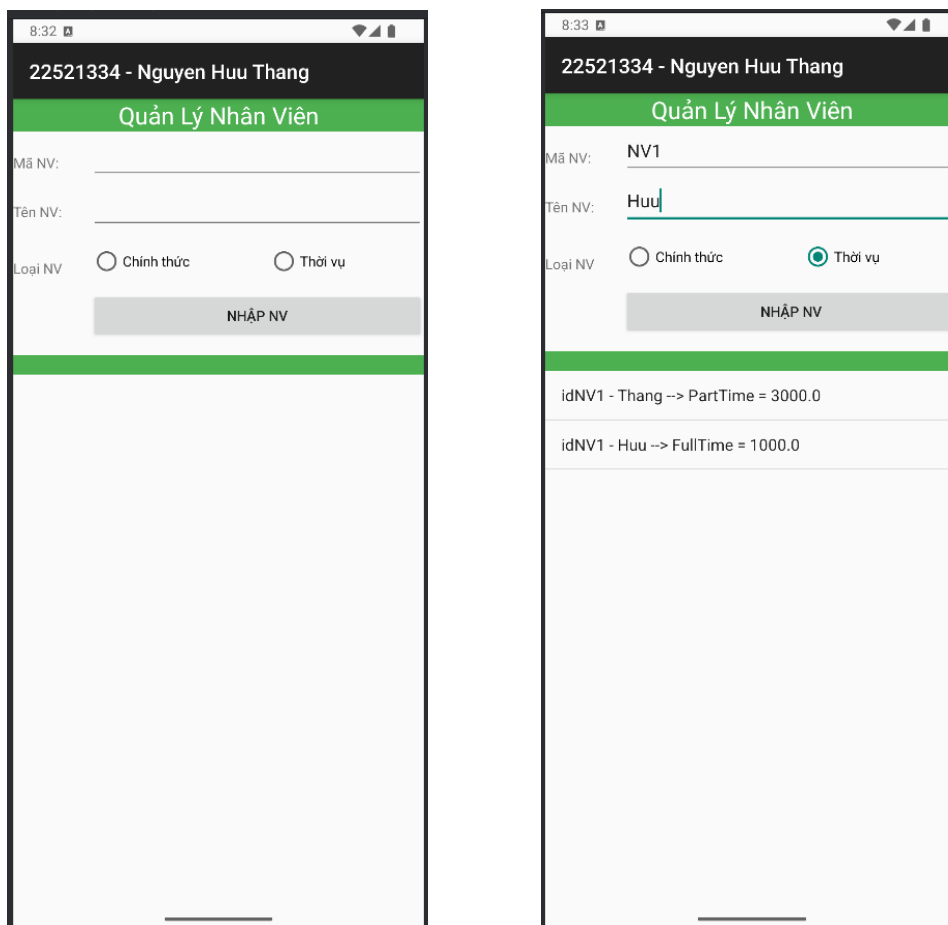
b. Mô tả

❖ Khai báo biến:

- ArrayAdapter<String> adapter: Kết nối dữ liệu với ListView.
 - ListView lvPerson: Hiển thị danh sách các tên.
 - TextView tvPerson: Hiển thị thông tin chi tiết khi người dùng chọn một tên.
 - ArrayList<String> arrayName: Danh sách chứa các tên.
 - Button btnSummit: Nút để thêm tên mới.
 - EditText etName: Ô nhập liệu cho người dùng để nhập tên.
- ❖ arrayName chứa các tên mẫu ban đầu ("Teo", "Ti", "Bin", "Bo").
 - ❖ adapter kết nối arrayName với ListView, giúp hiển thị danh sách này trong ListView (lvPerson).
 - ❖ Khi người dùng nhấn nút "Submit", tên nhập trong EditText sẽ được thêm vào arrayName.
 - ❖ Khi người dùng nhấn vào một mục trong ListView, TextView (tvPerson) sẽ hiển thị vị trí và tên của mục đó.
 - ❖ Khi người dùng nhấn giữ lâu vào một mục trong danh sách, mục đó sẽ bị xóa khỏi arrayName.

3. Sử dụng ArrayList và ListView có phần tử trong ArrayList là các Object

a. Hình ảnh minh chứng




```

1 public class Employee { 5 usages 2 inheritors
2     private String id; 4 usages
3     private String name; 4 usages
4
5     public Employee(String id, String name) { 2 usages
6         this.id = id;
7         this.name = name;
8     }
9
10    // Getter và Setter
11    public String getId() { 2 usages
12        return id;
13    }
14
15    public void setId(String id) { no usages
16        this.id = id;
17    }
18
19    public String getName() { 2 usages
20        return name;
21    }
22
23    public void setName(String name) { no usages
24        this.name = name;
25    }
26
27    public double tinhLuong() { 2 usages 2 overrides
28        return 0;
29    }
30
31    @NonNull 2 overrides
32    @Override
33    public String toString() {
34        return "Employee [id=" + id + ", name=" + name + "];"
35    }
36
37 }
38
39 > class EmployeePartTime extends Employee {...}
40
41 > class EmployeeFullTime extends Employee {...}

```

```

41  class EmployeePartTime extends Employee { 1 usage
42
43      // Constructor
44      public EmployeePartTime(String id, String name) { 1 usage
45          super(id, name);
46      }
47
48      @Override 2 usages
49      public double tinhLuong() {
50          return 1000;
51      }
52
53      @NonNull
54      @Override
55      public String toString() {
56          return "id" + getId() + " - " + getName() + " --> " + "FullTime = " + tinhLuong();
57      }
58  }
59
60  class EmployeeFullTime extends Employee { 1 usage
61
62      // Constructor
63      public EmployeeFullTime(String id, String name) { 1 usage
64          super(id, name);
65      }
66
67      @Override 2 usages
68      public double tinhLuong() {
69          return 3000;
70      }
71
72      @NonNull
73      @Override
74      public String toString() {
75          return "id" + getId() + " - " + getName() + " --> " + "PartTime = " + tinhLuong();
76      }
77  }

```

```

19 </> public class MainActivity extends AppCompatActivity {
20     EditText etMaNV; 2 usages
21     EditText etTenNV; 2 usages
22     RadioGroup rgNV; 2 usages
23     ListView lvNV; 2 usages
24     Button bNV; 2 usages
25     ArrayList<Employee> employees; 3 usages
26     ArrayAdapter<Employee> adapter; 3 usages
27     Employee employee; 3 usages
28
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         EdgeToEdge.enable(this);
32         setContentView(R.layout.activity_main);
33
34         // Khởi tạo các view
35         etMaNV = findViewById(R.id.etMaNV);
36         etTenNV = findViewById(R.id.etTenNV);
37         lvNV = findViewById(R.id.lvNV); // Khởi tạo ListView
38         rgNV = findViewById(R.id.rgNV);
39         bNV = findViewById(R.id.bNV);
40
41         employees = new ArrayList<>();
42         adapter = new ArrayAdapter<>(context: this, android.R.layout.simple_list_item_1, employees);
43         lvNV.setAdapter(adapter);
44
45         bNV.setOnClickListener(new View.OnClickListener() {
46             @Override
47             public void onClick(View v) {
48                 int type = rgNV.getCheckedRadioButtonId();
49                 String id = etMaNV.getText().toString();
50                 String name = etTenNV.getText().toString();
51
52                 if (type == R.id.rbCT) {
53                     employee = new EmployeeFullTime(id, name);
54                 } else {
55                     employee = new EmployeePartTime(id, name);
56                 }
57
58                 employees.add(employee);
59                 adapter.notifyDataSetChanged();
60             }
61         });
62
63         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
64             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
65             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
66             return insets;
67         });
68     }

```

b. Mô tả

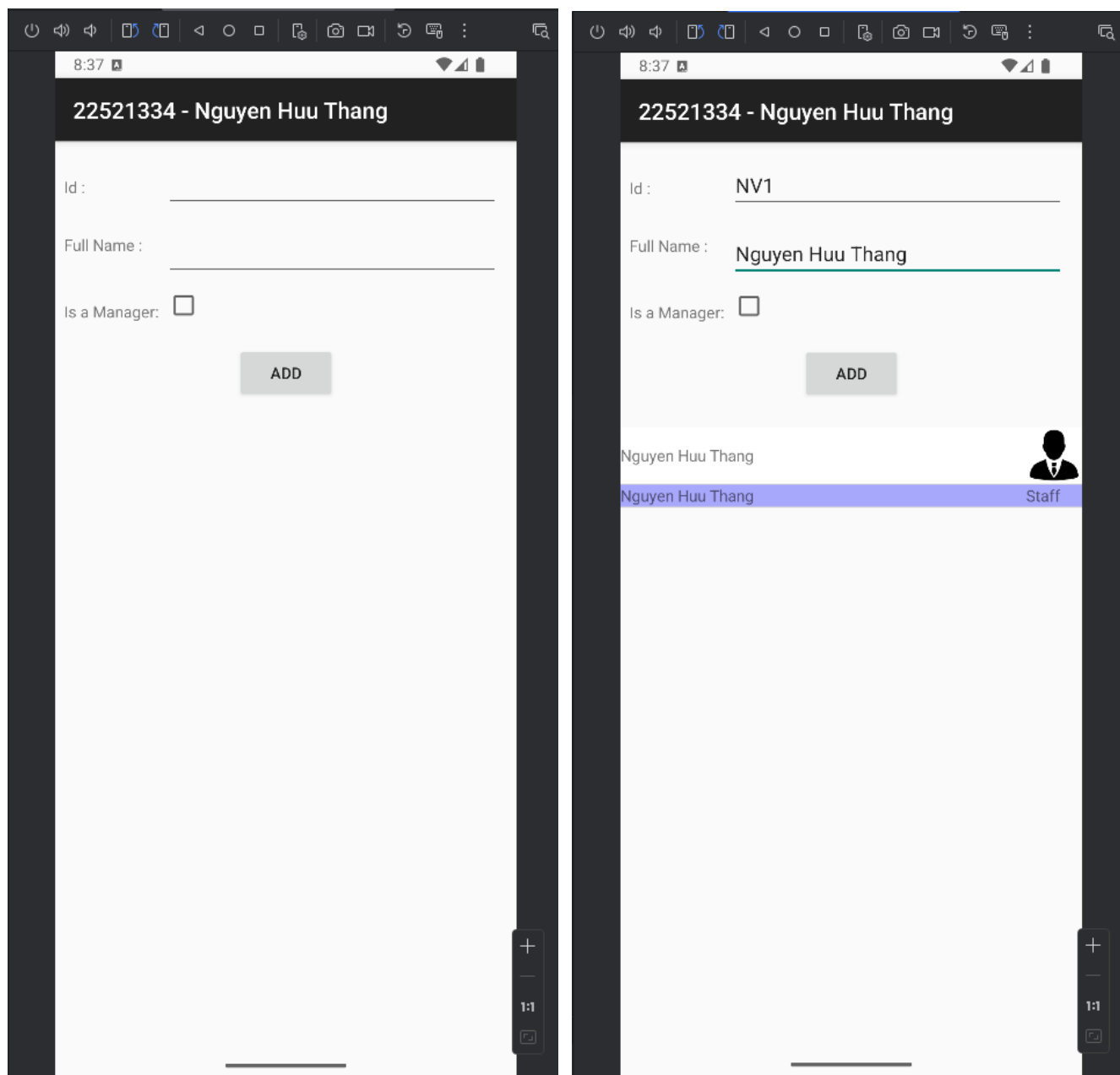
❖ Khai báo biến:

- EditText etMaNV, etTenNV: Ô nhập liệu cho mã nhân viên và tên nhân viên.
- RadioGroup rgNV: Nhóm radio button để lựa chọn loại nhân viên (Full-Time hoặc Part-Time).
- ListView lvNV: Hiển thị danh sách nhân viên.

- Button bNV: Nút để thêm nhân viên mới vào danh sách.
 - ArrayList<Employee> employees: Danh sách chứa các đối tượng nhân viên.
 - ArrayAdapter<Employee> adapter: Kết nối dữ liệu danh sách nhân viên với ListView.
 - Employee employee: Biến tạm để lưu thông tin nhân viên khi thêm vào danh sách.
- ❖ Khởi tạo ArrayList và ArrayAdapter:
 - ❖ employees chứa danh sách các nhân viên.
 - ❖ adapter kết nối employees với ListView (lvNV), giúp hiển thị danh sách nhân viên trong ListView.
 - ❖ Khi người dùng nhấn nút "Thêm nhân viên", dữ liệu nhập từ các ô etMaNV (mã nhân viên) và etTenNV (tên nhân viên) sẽ được lấy ra.
 - ❖ Kiểm tra lựa chọn loại nhân viên từ RadioGroup (rgNV):
 - Nếu chọn loại Full-Time (R.id.rbCT), tạo đối tượng EmployeeFullTime.
 - Nếu chọn loại Part-Time, tạo đối tượng EmployeePartTime.

4. Sử dụng CustomAdapter cho ListView

a. Hình ảnh minh chứng



```

1 package com.example.bth2_4;
2
3 public class Employee { 8 usages
4     private String id; 3 usages
5     private String fullName; 3 usages
6     private boolean isManager; 3 usages
7
8     public Employee (String id, String fullName, boolean isManager) { 1
9         this.id = id;
10        this.fullName = fullName;
11        this.isManager = isManager;
12    }
13
14    public String getId(){ no usages
15        return id;
16    }
17
18    public String getFullName(){ 2 usages
19        return fullName;
20    }
21
22    public void setId(String id){ no usages
23        this.id = id;
24    }
25
26    public void setFullName(String fullName){ no usages
27        this.fullName = fullName;
28    }
29
30    public boolean isManager(){ no usages
31        return isManager;
32    }
33    public void setManager(boolean isManager){ 2 usages
34        this.isManager = isManager;
35    }
36 }

```

```

17 </> public class EmployeeAdapter extends ArrayAdapter < Employee > { 2 usages
18     private Activity context; 3 usages
19     private int layoutID; no usages
20     private List< Employee > objects; no usages
21     public EmployeeAdapter(Activity context, int layoutID, List < Employee > 1 usage
22 >         objects) {...}
26     @NonNull
27     @Override
28     public View getView(final int position, View convertView, ViewGroup parent) {
29         if (convertView == null) {
30             convertView =
31                 LayoutInflater.from(context).inflate(R.layout.item_employee, root: null,
32                 attachToRoot: false);
33         }
34         // Get item
35         Employee employee = getItem(position);
36         // Get view
37         TextView tvFullName = (TextView)
38             convertView.findViewById(R.id.item_employee_tv_fullname);
39         TextView tvPosition = (TextView)
40             convertView.findViewById(R.id.item_employee_tv_position);
41         ImageView ivManager = (ImageView)
42             convertView.findViewById(R.id.item_employee_iv_manager);
43         LinearLayout llParent = (LinearLayout)
44             convertView.findViewById(R.id.item_employee_ll_parent);
45         // Set fullname
46         assert employee != null;
47 >         if (employee.getFullName() != null) {...} else tvFullName.setText("");
50         // If this is a manager -> show icon manager. Otherwise, show Staff in tvPosition
51 >         if (employee.isManager()) {...} else {
55             ivManager.setVisibility(View.GONE);
56             tvPosition.setVisibility(View.VISIBLE);
57             tvPosition.setText(context.getString(R.string.staff));
58         }
59         // Show different color backgrounds for 2 continuous employees
60 >         if (position % 2 == 0) {...} else {
63 ■             llParent.setBackgroundResource(R.color.light_blue);
64         }
65         return convertView;
66     }
67 }
    
```

```

18 public class MainActivity extends AppCompatActivity {
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         EdgeToEdge.enable(this);
24         setContentView(R.layout.activity_main);
25         // Declare
26         EditText edId = findViewById(R.id.edId);
27         EditText edName = findViewById(R.id.edName);
28         CheckBox cbManager = findViewById(R.id.cbManager);
29         Button bAdd = findViewById(R.id.bAdd);
30         ListView lvEmployee = findViewById(R.id.lvEmployee);
31         ArrayList<Employee> employees = new ArrayList<Employee>();
32         EmployeeAdapter adapter = new EmployeeAdapter(context, this, R.layout.item_employee, employees);
33         lvEmployee.setAdapter(adapter);
34         bAdd.setOnClickListener(new View.OnClickListener() {
35             @Override
36             public void onClick(View v) {
37                 String name = edName.getText().toString();
38                 String id = edId.getText().toString();
39                 Employee employee = new Employee(id, name, cbManager.isChecked());
40                 if (cbManager.isChecked()) {
41                     employee.setManager(true);
42                 } else {
43                     employee.setManager(false);
44                 }
45                 employees.add(employee);
46                 adapter.notifyDataSetChanged();
47             }
48         });
49
50         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
51             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
52             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
53             return insets;
54         });
55     }
56 }

```

b. Mô tả

❖ Khai báo và liên kết các phần tử giao diện:

- EditText edId: Ô nhập mã nhân viên.
- EditText edName: Ô nhập tên nhân viên.
- CheckBox cbManager: Hộp kiểm để xác định liệu nhân viên có phải là quản lý (Manager) hay không.
- Button bAdd: Nút để thêm nhân viên vào danh sách.
- ListView lvEmployee: Hiển thị danh sách nhân viên.
- ArrayList<Employee> employees: Danh sách chứa các đối tượng nhân viên.
- EmployeeAdapter adapter: Adapter tùy chỉnh kết nối dữ liệu với ListView.

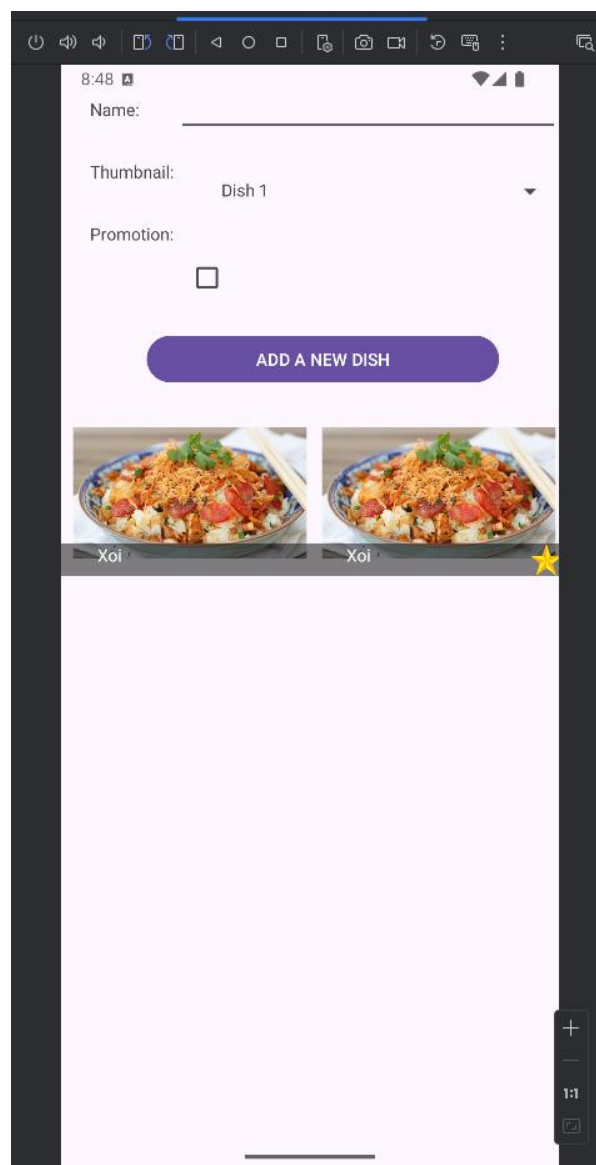
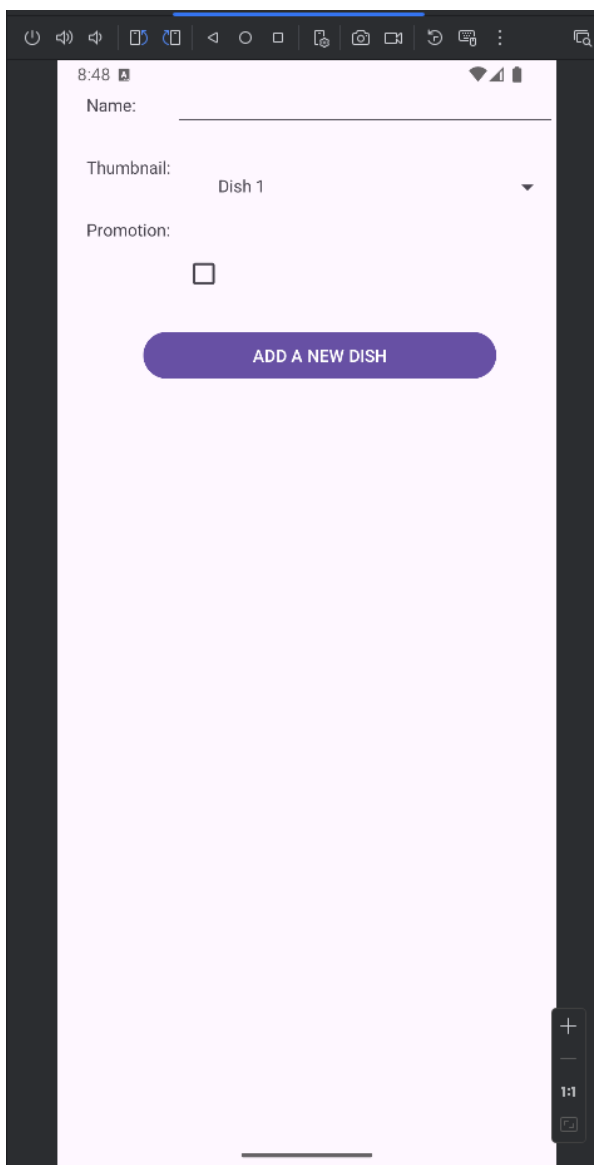
❖ Khởi tạo ArrayList và EmployeeAdapter:

❖ employees: Mảng chứa danh sách các nhân viên.

- ❖ EmployeeAdapter: Adapter tùy chỉnh kết nối danh sách nhân viên với ListView để hiển thị từng nhân viên theo định dạng tùy chỉnh từ tệp XML (item_employee.xml).
- ❖ Sự kiện khi nhấn nút bAdd (Thêm nhân viên):
 - Lấy dữ liệu từ các ô nhập liệu edId và edName.
 - Tạo một đối tượng Employee với thông tin mã nhân viên và tên nhân viên.
 - Kiểm tra hộp kiểm cbManager để xác định xem nhân viên có phải là quản lý hay không. Nếu được chọn, thuộc tính manager của nhân viên sẽ được đặt thành true, nếu không thì là false.
 - Thêm nhân viên vào danh sách employees.
 - Gọi adapter.notifyDataSetChanged() để cập nhật danh sách hiển thị trên ListView.

5. Sử dụng GridView, Spinner

a. Hình ảnh minh chứng



```

19 </> public class DishAdapter extends BaseAdapter { 2 usages
20     private Context context=null; 2 usages
21     private int layout; 1 usage
22     private ArrayList<Dish> dishList=null; 3 usages
23     public DishAdapter(Context context, int layout, ArrayList<Dish> dishList) { 1 usage
24         this.context = context;
25         this.layout = layout;
26         this.dishList = dishList;
27     }
28     @Override
29     public int getCount(){
30         return dishList.size();
31     }
32     @Override
33     public Object getItem(int i) {
34         return null;
35     }
36     @Override
37     public long getItemId(int i) {
38         return 0;
39     }
40     public View getView(final int position, View convertView, ViewGroup parent)
41     {
42         if (convertView == null) {...}
43         Dish dish = dishList.get(position);
44         ImageView imgDish = (ImageView) convertView.findViewById(R.id.imgMonan);
45         TextView txtDish = (TextView) convertView.findViewById(R.id.txtMonan);
46         txtDish.setSelected(true);
47         ImageView icnStar = (ImageView) convertView.findViewById(R.id.icnStar);
48         if (dish.isPromotion())
49             {...}
50         else
51             {...}
52         imgDish.setImageResource(dish.getThumbnail());
53         txtDish.setText(dish.getName());
54         return convertView;
55     }
56 }

```

```

22 public class SpinnerAdapter extends BaseAdapter { 2 usages
23     Context context; 4 usages
24     int layout; 1 usage
25     ArrayList<Dish> arrDish; 5 usages
26 public SpinnerAdapter(Context context, int layout, ArrayList<Dish> arrDish) { 1 usage
27     this.context = context;
28     this.layout = layout;
29     this.arrDish = arrDish;
30 }
31 @Override
32 public int getCount() {
33     return arrDish.size();
34 }
35 @Override
36 public Object getItem(int i) {
37     return null;
38 }
39 @Override
40 public long getItemId(int i) {
41     return 0;
42 }
43 @Override
44 public View getDropDownView(int position, View convertView, ViewGroup parent) {
45     LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
46     View row = inflater.inflate(R.layout.dropdown, parent, attachToRoot: false);
47     TextView thbName = row.findViewById(R.id.thbnailText);
48     ImageView flag = row.findViewById(R.id.thbnailImage);
49     Drawable drawable = context.getResources().getDrawable(arrDish.get(position).getThumbnail());
50     thbName.setText(arrDish.get(position).getName());
51     flag.setImageDrawable(drawable);
52     return row;
53 }
54 @Override
55 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
56     LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
57     View row = inflater.inflate(R.layout.dropdown, parent, attachToRoot: false);
58     TextView thbName = row.findViewById(R.id.thbnailText);
59     thbName.setText(arrDish.get(position).getName());
60     return row;
61 }

```

```

3 public class Dish { 17 usages
4     private String name; 4 usages
5     private int thumbnail; 4 usages
6     private boolean isPromotion; 4 usages
7     public Dish(String name, int thumbnail, boolean isPromotion) { 1
8         this.name = name;
9         this.thumbnail = thumbnail;
10        this.isPromotion = isPromotion;
11    }
12    public Dish(String name, int thumbnail) { 5 usages
13        this.name = name;
14        this.thumbnail = thumbnail;
15        this.isPromotion = true;
16    }
17    public String getName() { 3 usages
18        return name;
19    }
20    public int getThumbnail() { 3 usages
21        return thumbnail;
22    }
23    public boolean isPromotion() { no usages
24        return isPromotion;
25    }
26    public void setPromotion(boolean isPromotion) { no usages
27        this.isPromotion = isPromotion;
28    }
29    public void setName(String name) { no usages
30        this.name = name;
31    }
32    public void setThumbnail(int thumbnail) { no usages
33        this.thumbnail = thumbnail;
34    }
35 }
36

```

```

18 </> public class MainActivity extends AppCompatActivity {
19     Spinner spinnerDish; 4 usages
20     SpinnerAdapter spinnerAdapter; 2 usages
21     Dish res_dish; 2 usages
22     Button bAdd; 2 usages
23     GridView gvDish; 2 usages
24     EditText etName; 3 usages
25     DishAdapter adapter; 3 usages
26     CheckBox cbPromotion; 3 usages
27     ArrayList<Dish> arrDish; 8 usages
28     ArrayList<Dish> arrGvDish; 3 usages
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         EdgeToEdge.enable(this);
33         setContentView(R.layout.dish);
34
35         arrDish = new ArrayList<Dish>();
36         arrDish.add(new Dish(name: "Dish 1", R.drawable.dish1));
37         arrDish.add(new Dish(name: "Dish 2", R.drawable.dish2));
38         arrDish.add(new Dish(name: "Dish 3", R.drawable.dish3));
39         arrDish.add(new Dish(name: "Dish 4", R.drawable.dish4));
40         arrDish.add(new Dish(name: "Dish 5", R.drawable.dish5));
41
42         spinnerDish = (Spinner) findViewById(R.id.spinnerDish);
43         spinnerAdapter = new SpinnerAdapter(context: this, R.layout.dropdown, arrDish);
44         spinnerDish.setAdapter(spinnerAdapter);
45
46         spinnerDish.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
47             @Override 1 usage
48             public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
49                 res_dish = arrDish.get(position);
50             }
51             @Override 1 usage
52             public void onNothingSelected(AdapterView<?> parent) {
53             }
54         });
55
56         cbPromotion = (CheckBox) findViewById(R.id.checkBox);
57         bAdd = (Button) findViewById(R.id.bAdd);
58         gvDish = (GridView) findViewById(R.id.gvDish);
59         etName = (EditText) findViewById(R.id.etName);
60
61         arrGvDish = new ArrayList<Dish>();
62         adapter = new DishAdapter(context: this, R.layout.items, arrGvDish);
63         gvDish.setAdapter(adapter);
64
65         bAdd.setOnClickListener(new View.OnClickListener() {
66             @Override
67             public void onClick(View v) {
68                 String name = etName.getText().toString();
69                 Dish dish = new Dish(name, res_dish.getThumbnail(), cbPromotion.isChecked());
70                 arrGvDish.add(dish);
71                 Toast.makeText(context: MainActivity.this, text: "Added successfully", Toast.LENGTH_SHORT).show();
72                 etName.setText("");
73                 spinnerDish.setSelection(0);
74                 cbPromotion.setChecked(false);
75                 adapter.notifyDataSetChanged();
76             }
77         });
78     }
79 }

```

b. Mô tả

❖ Khai báo biến:

- Spinner spinnerDish: Dropdown menu chứa danh sách các món ăn.
- SpinnerAdapter spinnerAdapter: Adapter kết nối dữ liệu món ăn với Spinner.
- Dish res_dish: Biến lưu trữ món ăn được chọn từ Spinner.
- Button bAdd: Nút để thêm món ăn vào danh sách.
- GridView gvDish: Hiển thị danh sách món ăn đã thêm dưới dạng lưới.
- EditText etName: Ô nhập tên món ăn.
- DishAdapter adapter: Adapter kết nối dữ liệu món ăn với GridView.
- CheckBox cbPromotion: Tùy chọn để đánh dấu món ăn có khuyến mãi hay không.
- ArrayList<Dish> arrDish: Danh sách chứa các món ăn hiển thị trong Spinner.
- ArrayList<Dish> arrGvDish: Danh sách chứa các món ăn hiển thị trong GridView.

❖ Khởi tạo ArrayList chứa danh sách món ăn (arrDish):

- Các món ăn như "Dish 1", "Dish 2",... với hình ảnh tương ứng (được định nghĩa trong R.drawable) được thêm vào danh sách arrDish.

❖ Khởi tạo và gán Spinner:

- spinnerAdapter: Adapter kết nối arrDish với Spinner, hiển thị danh sách món ăn trong dropdown.
- spinnerDish.setAdapter(spinnerAdapter): Gán adapter cho Spinner để hiển thị dữ liệu.

❖ Xử lý sự kiện chọn món trong Spinner (setOnItemSelectedListener):

- Khi người dùng chọn một món ăn trong Spinner, món ăn đó được lưu trong biến res_dish.

❖ Thiết lập CheckBox, Button, và GridView:

- CheckBox cbPromotion: Người dùng có thể chọn để đánh dấu món ăn có khuyến mãi hay không.
- arrGvDish: Danh sách các món ăn được thêm vào GridView.
- adapter: Adapter kết nối dữ liệu món ăn với GridView để hiển thị các món ăn đã thêm.
- gvDish.setAdapter(adapter): Gán adapter cho GridView.

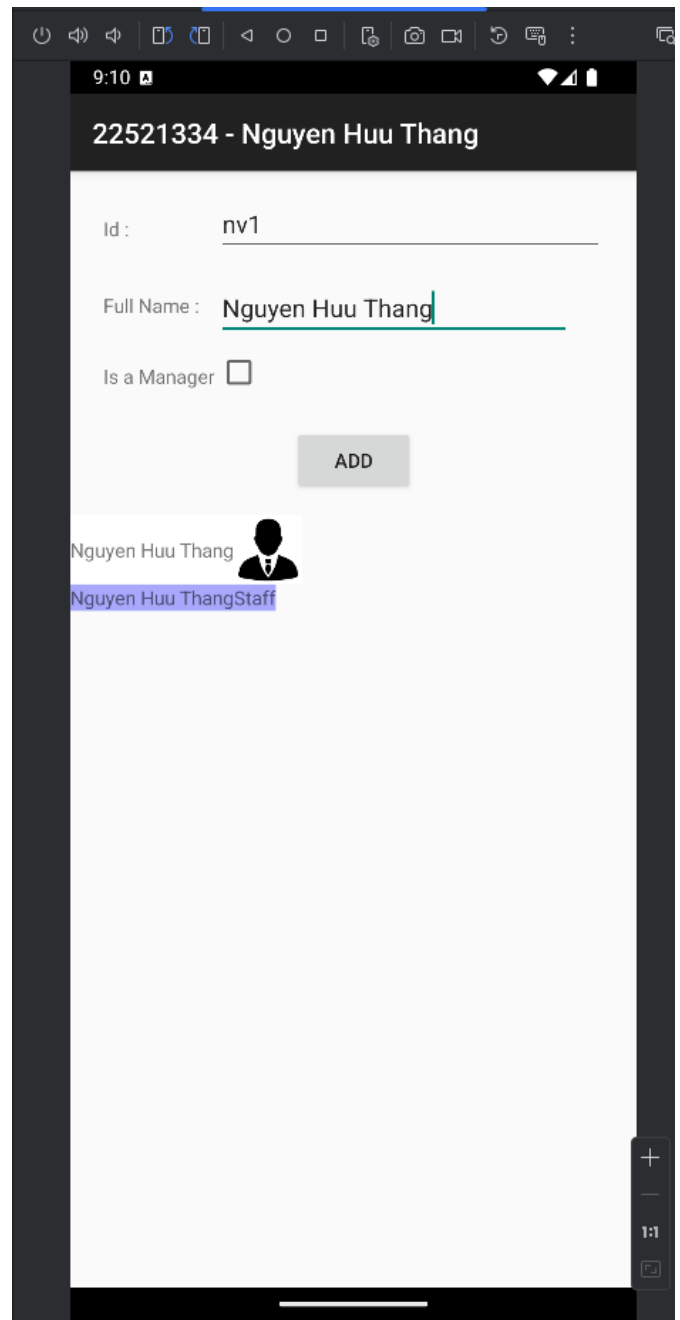
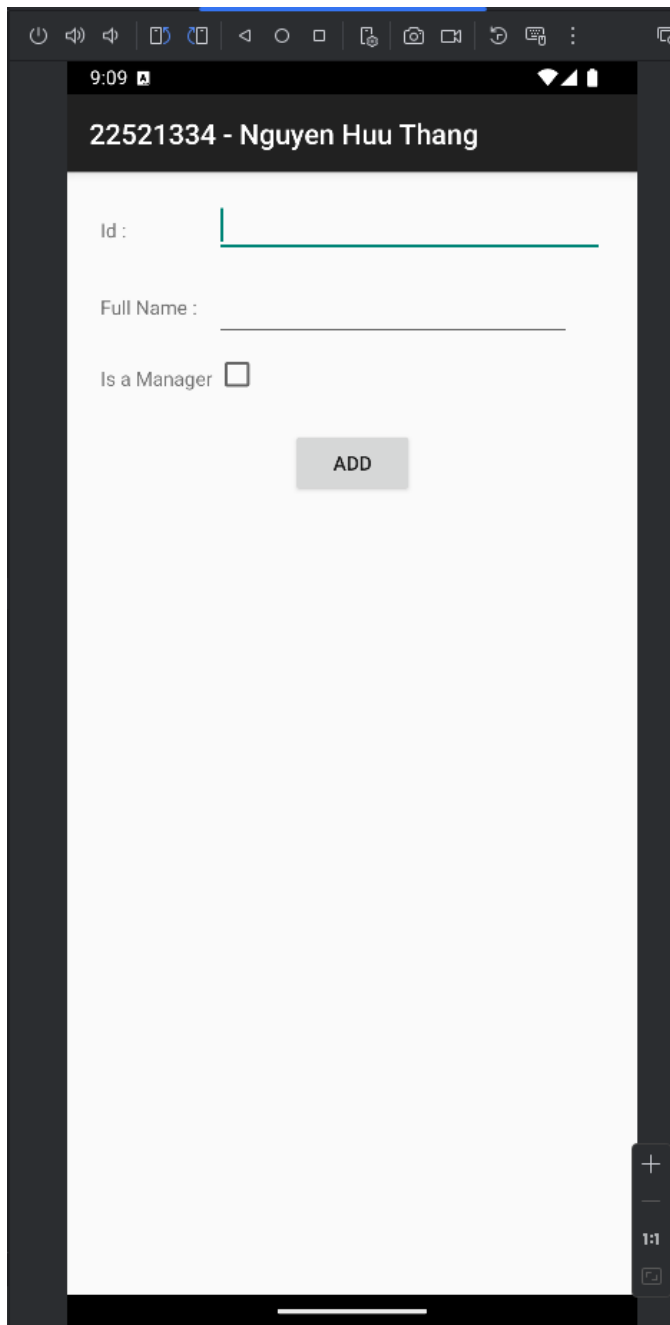
❖ Xử lý sự kiện nhấn nút bAdd (Thêm món ăn):

- Lấy tên món ăn từ ô EditText (etName) và món đã chọn từ Spinner (res_dish).
- Tạo đối tượng Dish mới với tên, hình ảnh, và trạng thái khuyến mãi (dựa trên trạng thái CheckBox).
- Thêm món ăn mới vào danh sách arrGvDish.
- Hiển thị thông báo thành công qua Toast.

- ❖ Sau khi thêm món, reset lại các trường: xóa nội dung trong EditText, đặt Spinner về vị trí mặc định, và bỏ chọn CheckBox.

6. RecyclerView

a. Hình ảnh minh chứng



```

1  public class Employee { 9 usages
2      private String Id; 3 usages
3      private String FullName; 3 usages
4      private boolean isManager; 3 usages
5
6      public Employee(String id, String fullName, boolean isManager) { no usages
7          Id = id;
8          FullName = fullName;
9          this.isManager = isManager;
10     }
11     public Employee() {}
12     public String getId() { no usages
13         return Id;
14     }
15     public void setId(String id) { no usages
16         Id = id;
17     }
18     public String getFullName() { 2 usages
19         return FullName;
20     }
21     public void setFullName(String fullName) { 1 usage
22         FullName = fullName;
23     }
24     public boolean isManager() { no usages
25         return isManager;
26     }
27     public void setManager(boolean manager) { 1 usage
28         isManager = manager;
29     }
30 }
31
32
33
34

```



```

18 public class EmployeeAdapter extends RecyclerView.Adapter<EmployeeAdapter.EmployeeHolder>{ 3 usag
19     private Activity context=null; 3 usages
20     private int layoutID; 1 usage
21     private List<Employee> mListEmployee=null; 5 usages
22
23     public EmployeeAdapter(Activity context, int layoutID, List<Employee> mListEmployee) { 1 usage
24         this.context = context;
25         this.layoutID = layoutID;
26         this.mListEmployee = mListEmployee;
27     }
28
29     public List<Employee> getListEmployee() { no usages
30         return mListEmployee;
31     }
32
33     public void setmListEmployee(List<Employee> mListEmployee) { no usages
34         this.mListEmployee = mListEmployee;
35     }
36     class EmployeeHolder extends RecyclerView.ViewHolder { 4 usages
37         TextView tvFullName, tvPosition; 3 usages
38         ImageView ivManager; 3 usages
39         LinearLayout llParent; 3 usages
40         public EmployeeHolder(@NonNull View itemView) { 1 usage
41             super(itemView);
42             tvFullName = (TextView)
43                 itemView.findViewById(R.id.item_employee_tv_fullname);
44             tvPosition = (TextView)
45                 itemView.findViewById(R.id.item_employee_tv_position);
46             ivManager = (ImageView)
47                 itemView.findViewById(R.id.item_employee_iv_manager);
48             llParent = (LinearLayout)
49                 itemView.findViewById(R.id.item_employee_ll_parent);
50         }
51     }
52 }

```

```

55  public EmployeeHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
56      View view = LayoutInflater.from(context).inflate(R.layout.item_employee, root: null,
57          attachToRoot: false);
58      return new EmployeeHolder(view);
59  }
60  @Override
61  public void onBindViewHolder(@NonNull EmployeeHolder holder, int position) {
62      Employee employee = mListEmployee.get(position);
63      if (employee == null) {...}
64      if (employee.getFullName() != null) {
65          holder.tvFullName.setText(employee.getFullName());
66      }
67      else holder.tvFullName.setText("");
68      if (employee.isManager())
69      {
70          holder.ivManager.setVisibility(View.VISIBLE);
71          holder.tvPosition.setVisibility(View.GONE);
72      }
73      else
74      {
75          holder.ivManager.setVisibility(View.GONE);
76          holder.tvPosition.setVisibility(View.VISIBLE);
77          holder.tvPosition.setText(context.getString(R.string.staff));
78      }
79      if (position%2==0)
80      {
81          holder.llParent.setBackgroundResource(R.color.white);
82      }
83      else
84      {
85          holder.llParent.setBackgroundResource(R.color.light_blue);
86      }
87  }
88  @Override
89  public int getItemCount() {
90      return mListEmployee.size();
91  }
92  }
93  }
94  }
95  }

```

```

15 </> public class MainActivity extends AppCompatActivity {
16     EditText etName; 2 usages
17     EditText etID; no usages
18     CheckBox cbManager; 2 usages
19     Button bAdd; 2 usages
20     RecyclerView rcvEmployee;; 3 usages
21     ArrayList<Employee> employees; 3 usages
22     EmployeeAdapter adapter; 3 usages
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_main);
28         etName = (EditText) findViewById(R.id.etName);
29         cbManager = (CheckBox) findViewById(R.id.chbxManager);
30         bAdd = (Button) findViewById(R.id.bAdd);
31         rcvEmployee = (RecyclerView) findViewById(R.id.rcvEmployee);
32         employees = new ArrayList<Employee>();
33
34         adapter = new EmployeeAdapter( context: this, R.layout.item_employee,employees);
35         LinearLayoutManager layoutManager = new LinearLayoutManager( context: this);
36         rcvEmployee.setLayoutManager(layoutManager);
37         rcvEmployee.setAdapter(adapter);
38
39         bAdd.setOnClickListener(new View.OnClickListener() {
40             @Override
41             public void onClick(View view) {
42                 String name = etName.getText().toString();
43                 Employee employee = new Employee();
44                 employee.setManager(cbManager.isChecked());
45                 employee.setFullName(name);
46                 employees.add(employee);
47                 adapter.notifyDataSetChanged();
48             }
49         });
50     }
51 }

```

b. Mô tả

❖ Khai báo biến:

- EditText etName: Ô nhập tên nhân viên.
- EditText etID: Ô nhập mã nhân viên (mặc dù trong đoạn mã này etID chưa được sử dụng).
- CheckBox cbManager: Hộp kiểm để xác định liệu nhân viên có vai trò quản lý hay không.
- Button bAdd: Nút để thêm nhân viên vào danh sách.
- RecyclerView rcvEmployee: Danh sách hiển thị các nhân viên dưới dạng cuộn dọc.
- ArrayList<Employee> employees: Danh sách chứa các đối tượng Employee.

- EmployeeAdapter adapter: Adapter tùy chỉnh để kết nối dữ liệu nhân viên với RecyclerView.
- ❖ Gán các thành phần giao diện:
 - etName, cbManager, bAdd, và rcvEmployee được gán các giá trị từ layout (activity_main.xml).
- ❖ Thiết lập RecyclerView:
 - EmployeeAdapter adapter: Adapter tùy chỉnh để kết nối danh sách employees với RecyclerView.
 - LinearLayoutManager layoutManager: Định nghĩa cách sắp xếp các item trong RecyclerView, ở đây là sắp xếp theo chiều dọc (list dọc).
 - rcvEmployee.setLayoutManager(linearLayoutManager): Gán layout cho RecyclerView để nó hiển thị danh sách theo chiều dọc.
 - rcvEmployee.setAdapter(adapter): Gán adapter cho RecyclerView để hiển thị danh sách nhân viên.
- ❖ Sự kiện nhấn nút bAdd (Thêm nhân viên):
 - Lấy thông tin từ người dùng:
 - Lấy tên nhân viên từ ô nhập liệu etName.
 - Tạo một đối tượng Employee mới, gán giá trị name và trạng thái manager (quản lý hay không) từ CheckBox cbManager.
 - Thêm nhân viên vào danh sách:
 - Thêm đối tượng Employee vào danh sách employees.
 - Gọi adapter.notifyDataSetChanged() để cập nhật danh sách hiển thị trong RecyclerView.