

BÁO CÁO THỰC HÀNH

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

Buổi báo cáo: Lab 02

Tên chủ đề: LISTVIEW và RECYCLERVIEW

GVHD: Phan Xuân Thiện

Ngày thực hiện: 4/10/2024

THÔNG TIN CHUNG:

Lớp: NT118.P13

STT	Họ và tên	MSSV	Email
1	Lê Hoàng Vũ	22521691	22521691@gm.uit.edu.vn

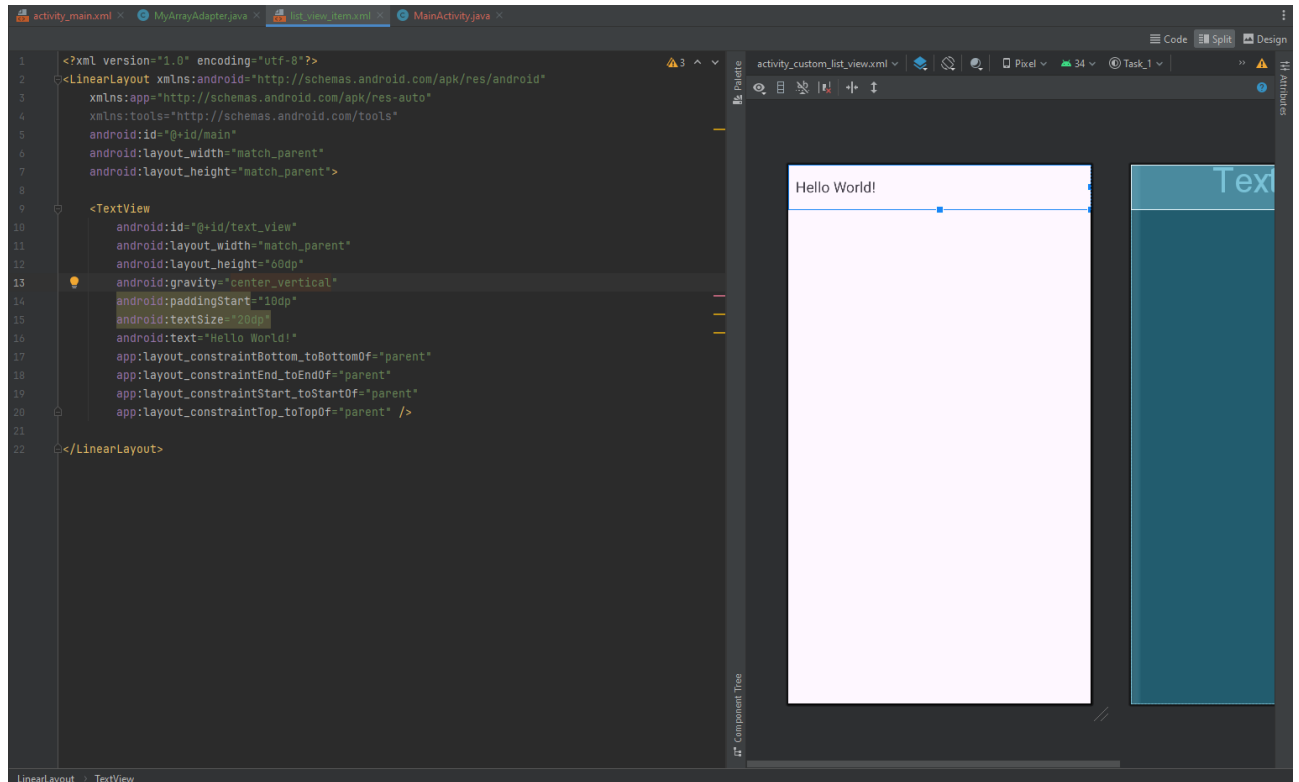
1. ĐÁNH GIÁ KHÁC:

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	7 ngày
Link Video thực hiện (nếu có)	
Mã nguồn	https://github.com/kOpHAlvU/LAB_NT118
Điểm tự đánh giá	10/10

BÁO CÁO CHI TIẾT

1. Câu hỏi 1

- Tạo layout list_view_item.xml để thiết kế một item sẽ được xuất hiện như thế nào:



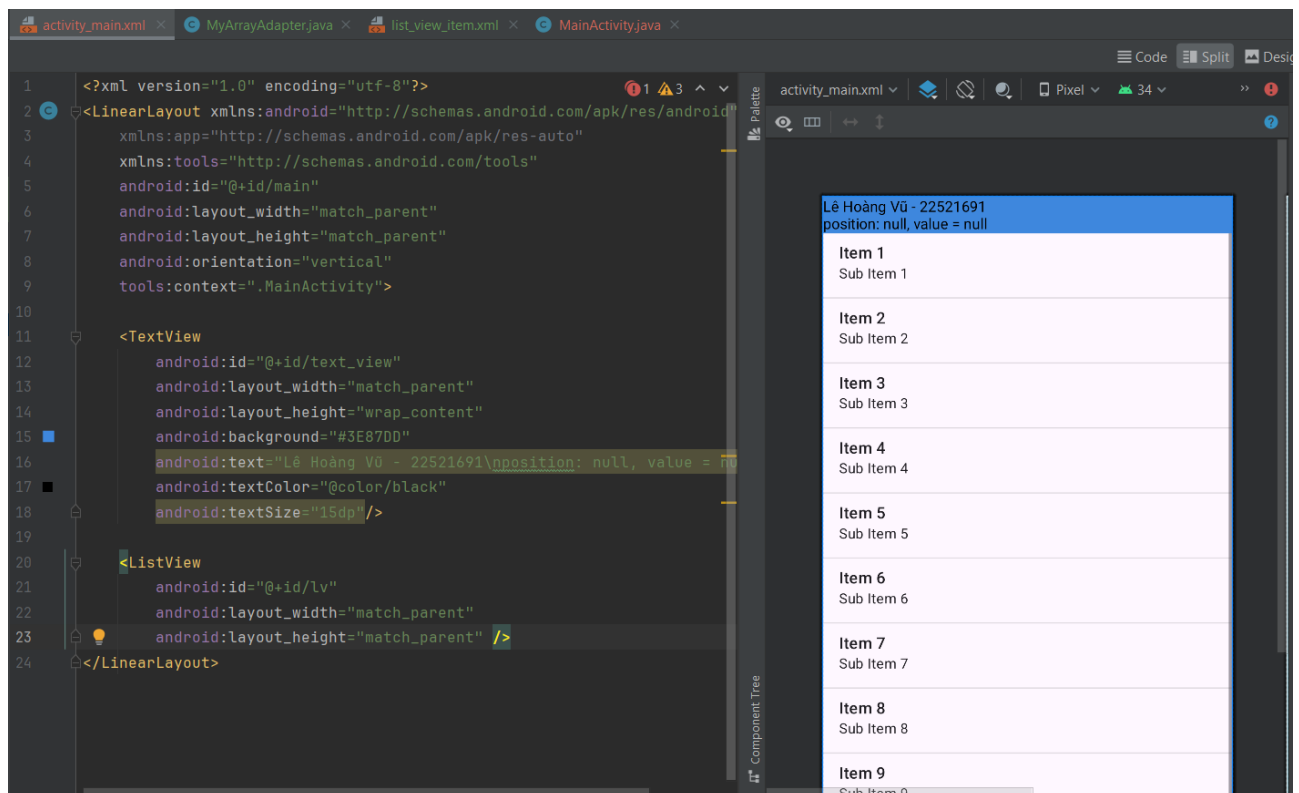
- Sử dụng hàm `MyArrayAdapter` để tạo một Adapter:

```

16 public class MyArrayAdapter extends ArrayAdapter<String> { 2 usages
17     Activity context; 2 usages
18     int IdLayout; 2 usages
19     ArrayList<String> myList; 2 usages
20
21     public MyArrayAdapter(Activity context, int idLayout, ArrayList<String> myList) { 1 usage
22         super(context, idLayout, myList);
23         this.context = context;
24         IdLayout = idLayout;
25         this.myList = myList;
26     }
27
28     @NonNull
29     @Override
30     public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
31         LayoutInflater myInflater = context.getLayoutInflater();
32         convertView = myInflater.inflate(IdLayout, root: null);
33         String name = myList.get(position);
34         TextView tv = convertView.findViewById(R.id.text_view);
35         tv.setText(name);
36         return convertView;
37     }
38 }

```

- Trong giao diện ở MainActivity tạo một TextView chứa Họ & Tên, MSSV và vị trí, giá trị của item được click vào:



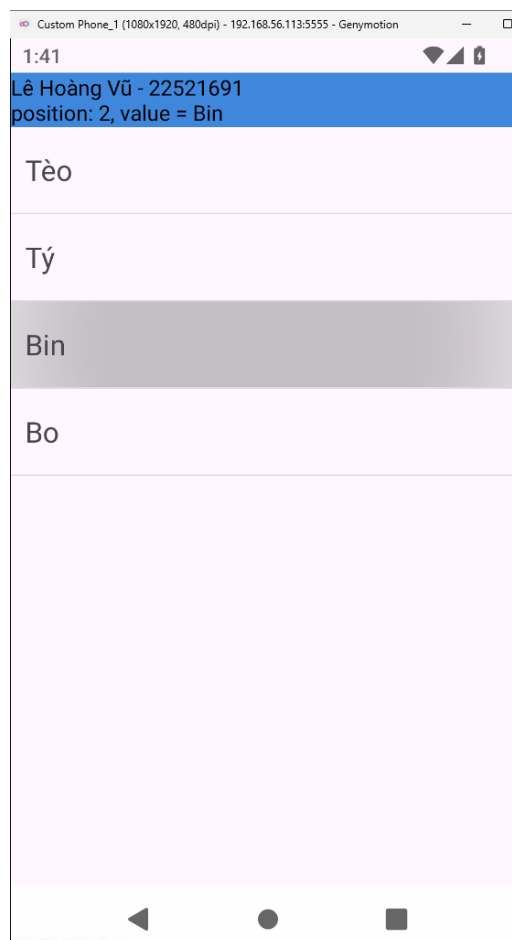
- Trong chương trình, ta đưa 4 giá trị tên Tèo, Tý, Bìn, Bo vào ArrayAdapter với layout là list_view_item để đem giao diện đó xuất hiện ở giao diện



MainActivity. Bên cạnh đó thiết lập hàm `lv.setOnItemClickListener(...)` để nó chỉnh sửa giá trị của Text View thành vị trí vừa được click vào:

```
24 protected void onCreate(Bundle savedInstanceState) {  
25     super.onCreate(savedInstanceState);  
26     EdgeToEdge.enable(this);  
27     setContentView(R.layout.activity_main);  
28     ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {  
29         Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
30         v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
31         return insets;  
32     });  
33  
34     lv = findViewById(R.id.lv);  
35     text_view = findViewById(R.id.text_view);  
36     myList = new ArrayList<>();  
37     for (int i = 0; i < name.length; i++)  
38         myList.add(name[i]);  
39     myArrayAdapter = new MyArrayAdapter(context MainActivity.this, R.layout.list_view_item, myList);  
40  
41     lv.setAdapter(myArrayAdapter);  
42  
43     lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
44         @Override  
45         public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
46             text_view.setText("Lê Hoàng Vũ - 22521691\nposition: " + position + ", value = " + name[position]);  
47         }  
48     });  
49 }
```

- Từ đó ra được kết quả như hình dưới:



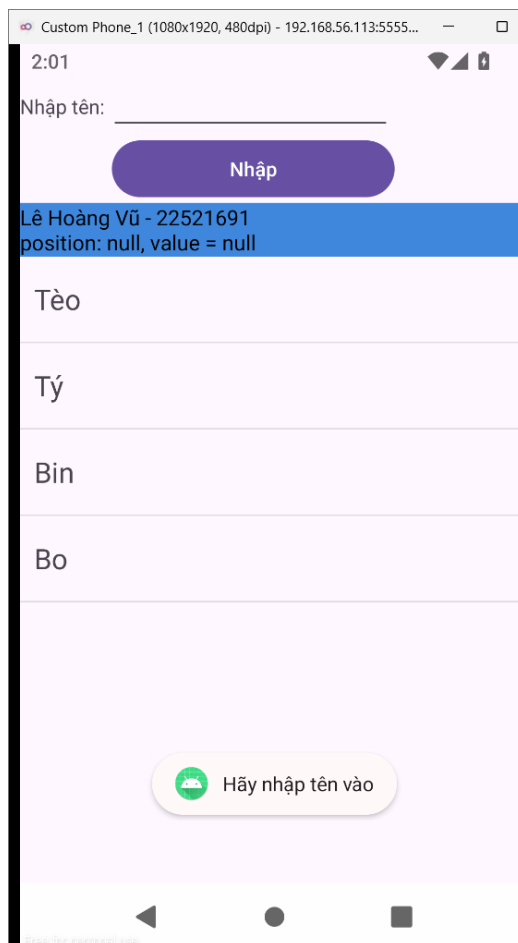
2. Câu hỏi 2

- Sử dụng lại mã nguồn từ Câu 1, bổ sung thêm ở giao diện Edit Text và Button để thêm dữ liệu.
- Ở phần code thêm sự kiện cho nút như dưới:

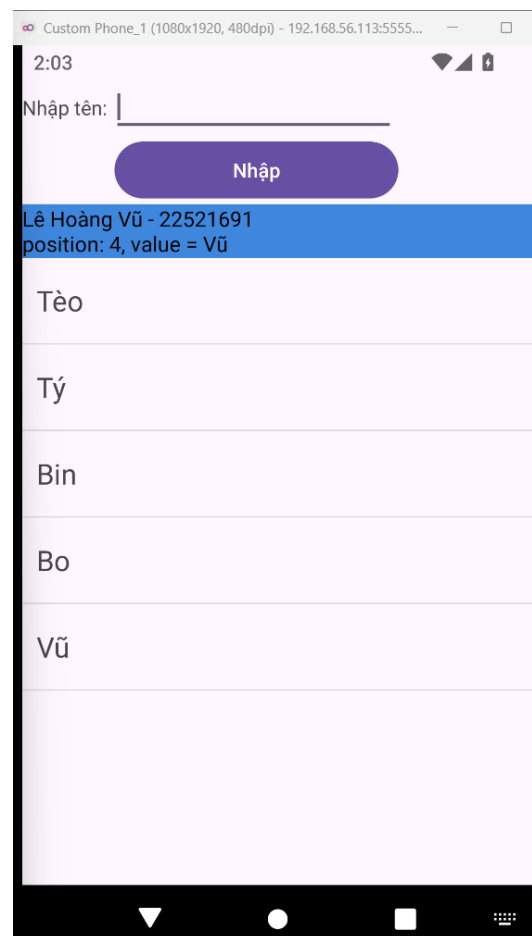
```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(edit_text.getText().toString().isEmpty()) {  
            Toast.makeText(context: MainActivity.this, text: "Hãy nhập tên vào", Toast.LENGTH_SHORT).show();  
            return;  
        }  
        myList.add(edit_text.getText().toString());  
        edit_text.setText("");  
        myArrayAdapter.notifyDataSetChanged();  
    }  
});
```

- Hàm trên sẽ kiểm tra Edit Text có trống hay không, nếu có thì sẽ không cho nhập thêm dữ liệu. Nếu không trống thì sẽ tiến hành thêm dữ liệu và cập nhập lại Adapter. Từ đó ra được kết quả như dưới:

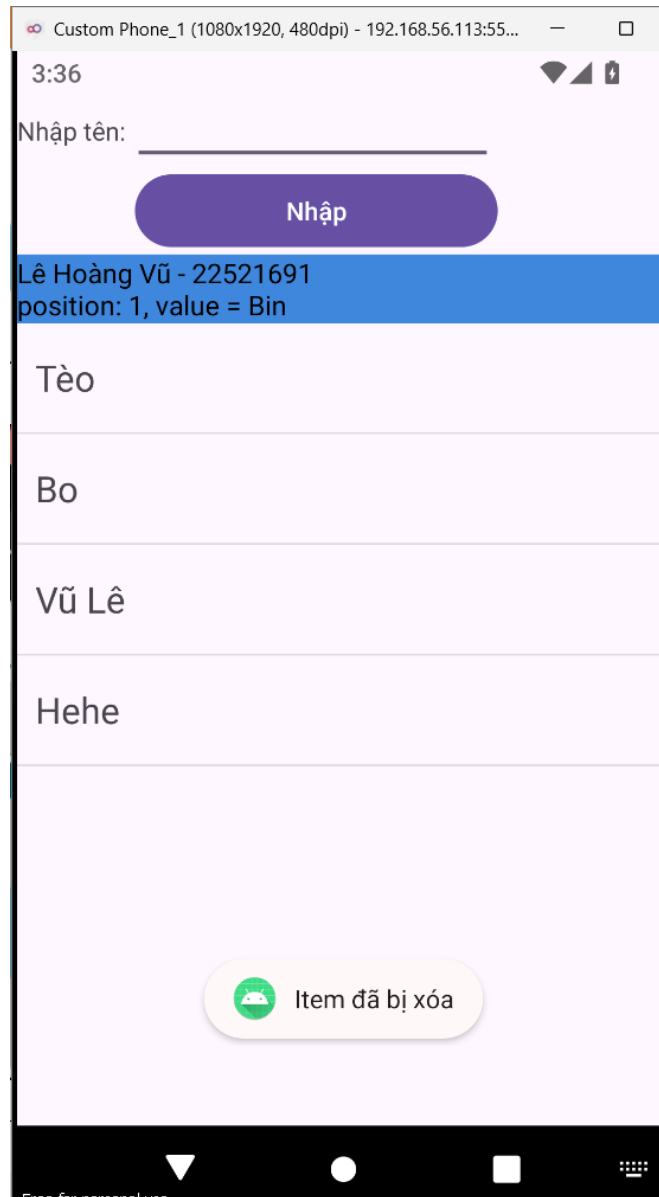
Trường dữ liệu để trống



Trường dữ liệu được nhập



- Khi ấn giữ lâu (long click) sẽ kích hoạt hàm `setOnItemLongClickListener(...)` của List View và ngay lập tức xóa Item đó và hiện ra thông báo như hình dưới:



3. Câu hỏi 3

- Thiết kế giao diện như hình dưới, tạo các id cho các trường cần thiết và tham chiếu id nó sang class:

Quản lý nhân viên

Mã NV: _____

Tên NV: _____

Loại NV: ☐ Chính thức ☐ Thời vụ

Nhập NV

2521691 - Lê Hoàng Vũ

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

- Khi chạy chương trình, giao diện như thế này sẽ hiện ra:

Custom Phone_1 (1080x1920, 480dpi) - 192.168.56.113:5555...

3:01

Quản lý nhân viên

Mã NV: _____

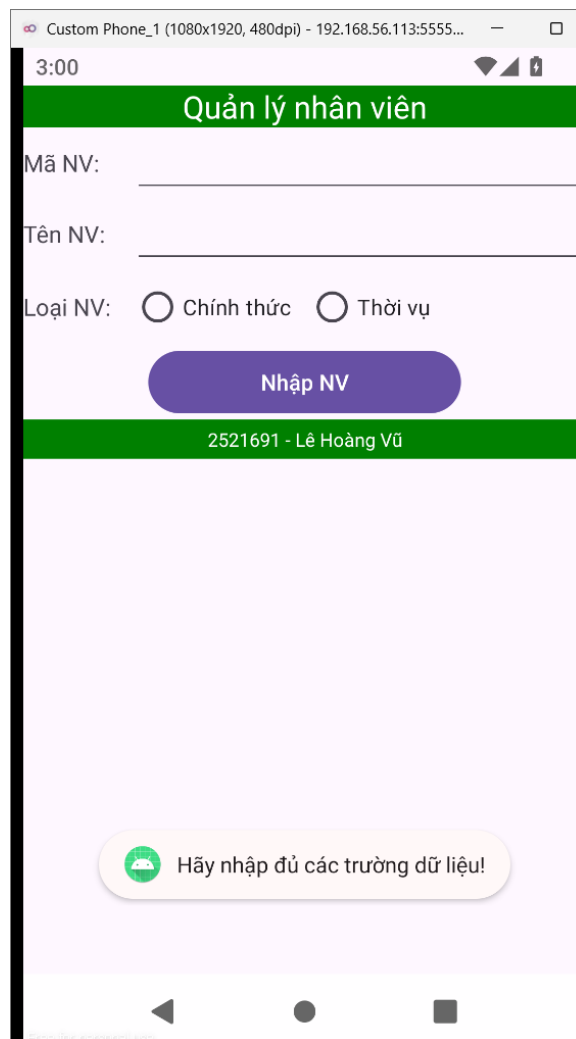
Tên NV: _____

Loại NV: ☐ Chính thức ☐ Thời vụ

Nhập NV

2521691 - Lê Hoàng Vũ

- Tạo sự kiện `onClickListener` khi ấn vào nút “Nhập NV”. Khi thiếu trường dữ liệu sẽ có thông báo bắt nhập đủ trường dữ liệu mới có thể thêm nhân viên:



```
public void addNewEmployee() { 1 usage
    Employee employee;
    if (employeeId.getText().toString().isEmpty()
        || employeeName.getText().toString().isEmpty()
        || (!fulltime.isChecked() && !parttime.isChecked())) {
        Toast.makeText(context: MainActivity.this, text: "Hãy nhập đủ các trường dữ liệu!", Toast.LENGTH_SHORT).show();
        return;
    }

    if (parttime.isChecked())
        employee = new EmployeeParttime();
    else
        employee = new EmployeeFullTime();

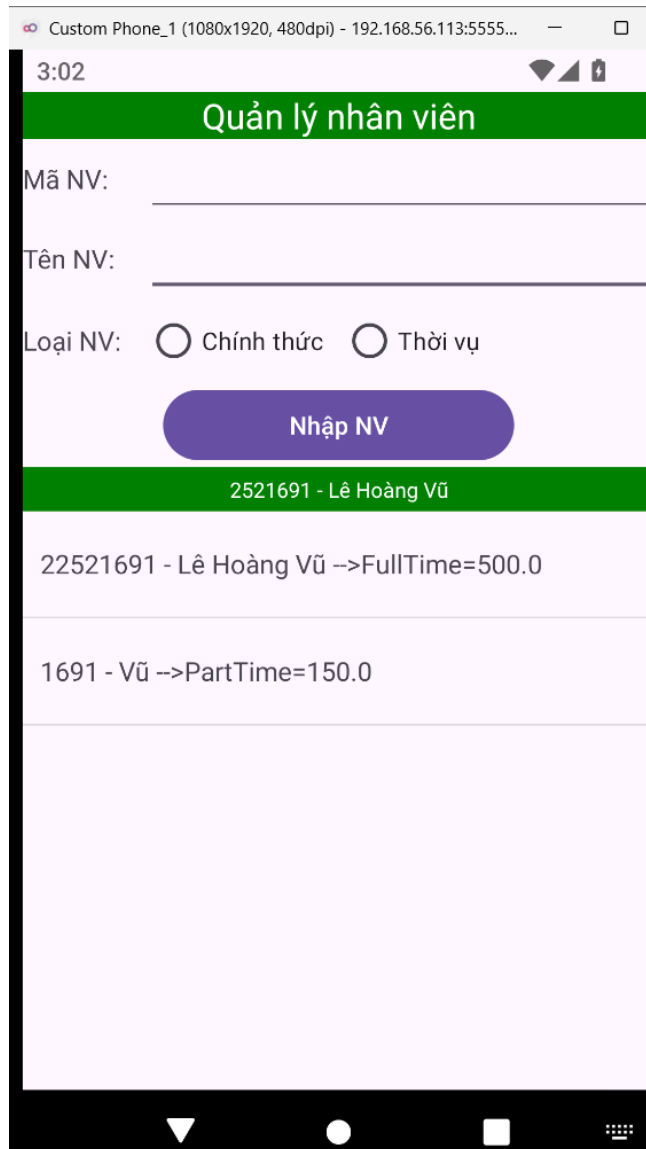
    employee.setEmployId(employeeId.getText().toString());
    employee.setEmployName(employeeName.getText().toString());

    myList.add(employee);

    myArrayAdapter.notifyDataSetChanged();

    // Clear data
    employeeId.setText("");
    employeeName.setText("");
    employType.clearCheck();
}
```

- Sau khi kiểm tra trường dữ liệu đầy đủ, dữ liệu nhân viên sẽ được thêm và hiển thị ở phía dưới như hình. Các Edit Text và Radio Button sẽ được clear để có thể tiếp tục nhập dữ liệu:



4. Câu hỏi 4

- Thiết kế giao diện như hình dưới:

The screenshot shows an Android application interface. At the top, there is a form with three input fields: 'ID:', 'Full Name:', and 'Is A Manager:'. The 'Is A Manager:' field has a checkbox. Below the form is a purple button labeled 'ADD'. Under the button is a header bar with the text '22521691 - Lê Hoàng VŨ'. Below the header bar is a list of seven items, each with a title and a subtitle:

- 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
- Item 7
Sub Item 7

- Cũng tương tự như bài trước nhưng kiểm tra giá trị position trong Adapter để có thể chia màu nền cho các Item, nếu là số chẵn thì nền trắng và số lẻ thì nền da trời.
- Từ đó ta được kết quả:

Custom Phone_1 (1080x1920, 480dpi) - 192.168.56.113:5...

8:35


ID: _____

Full Name: _____

Is A Manager: ☐

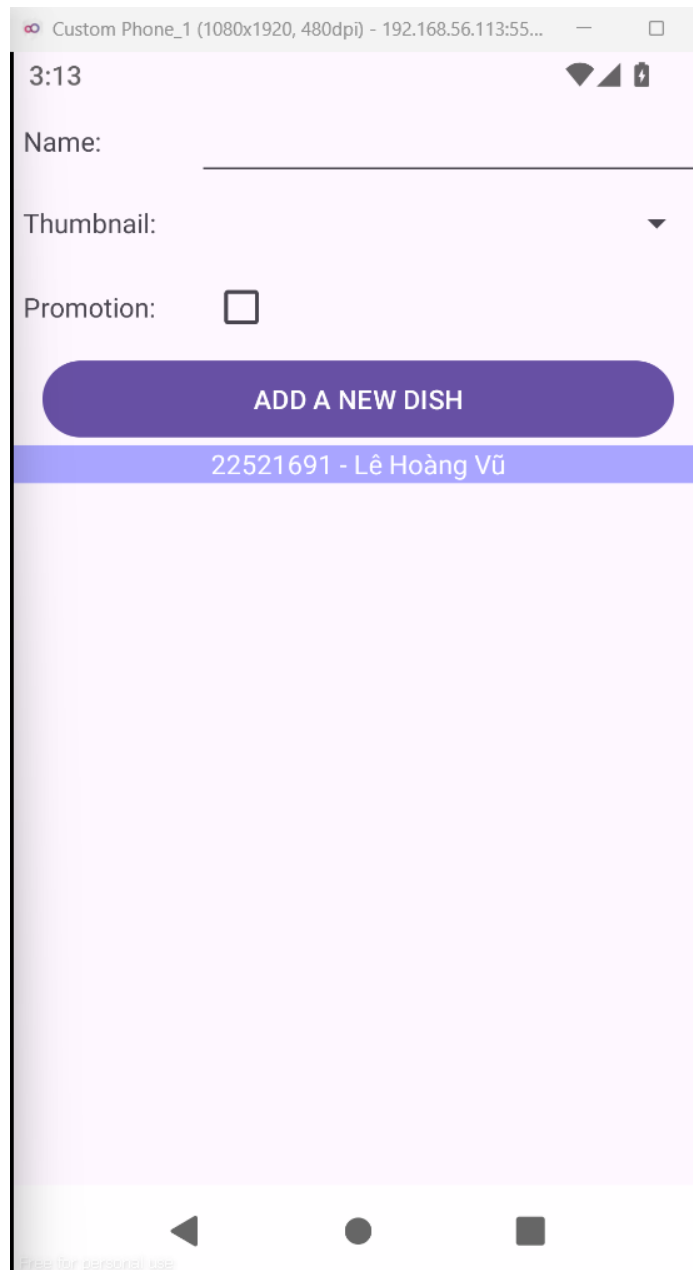
ADD

22521691 - Lê Hoàng Vũ

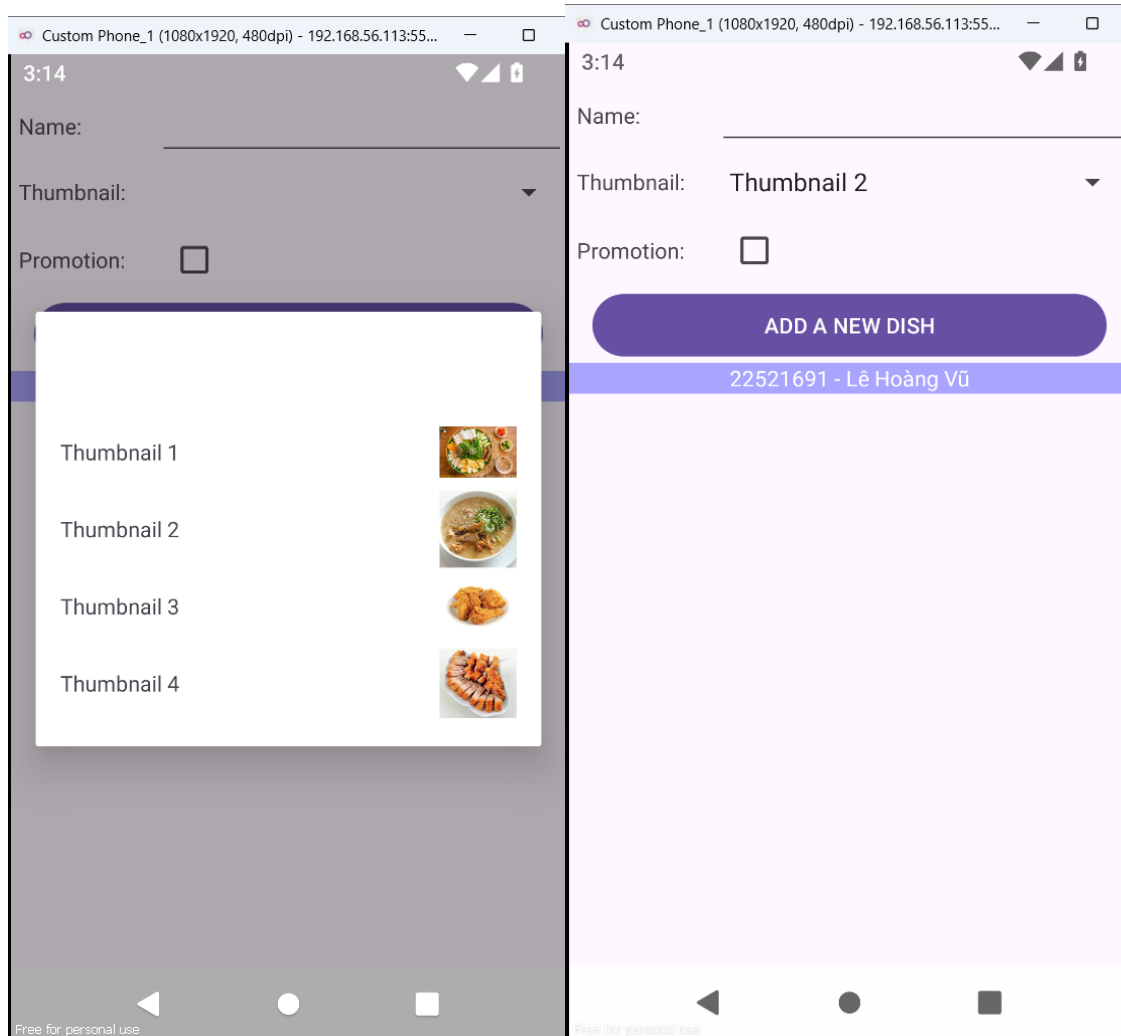
Vũ	Staff
Hoàng Vũ	Staff
Lê Vũ	

5. Câu hỏi 5

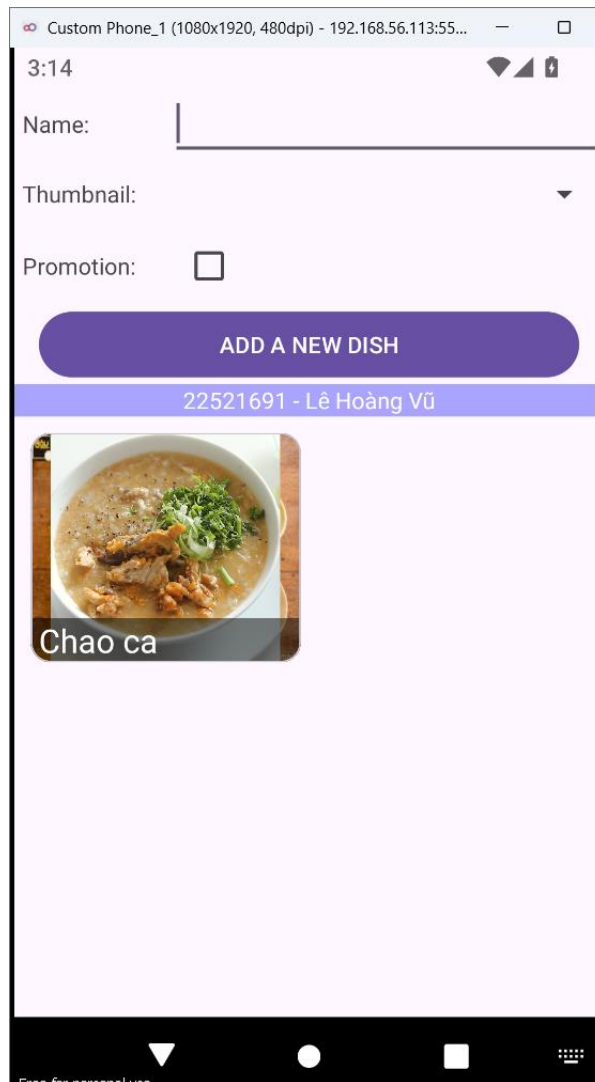
- Thiết kế giao diện như hình dưới:



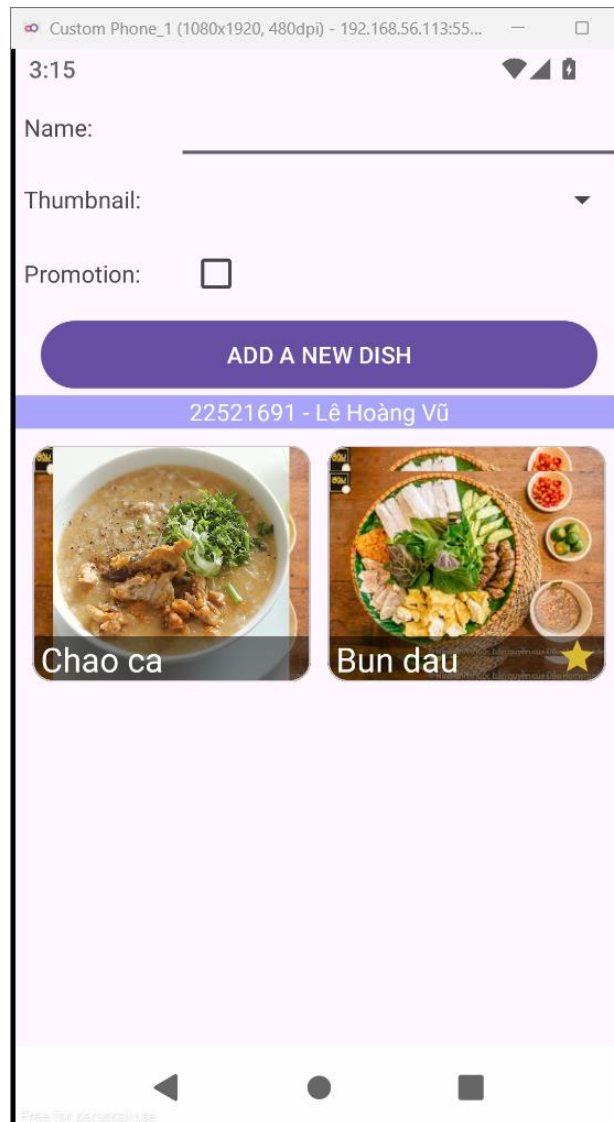
- Tạo Dialog khi ấn vào Thumbnail, sau khi chọn thì giá trị ở Thumbnail sẽ thay đổi:



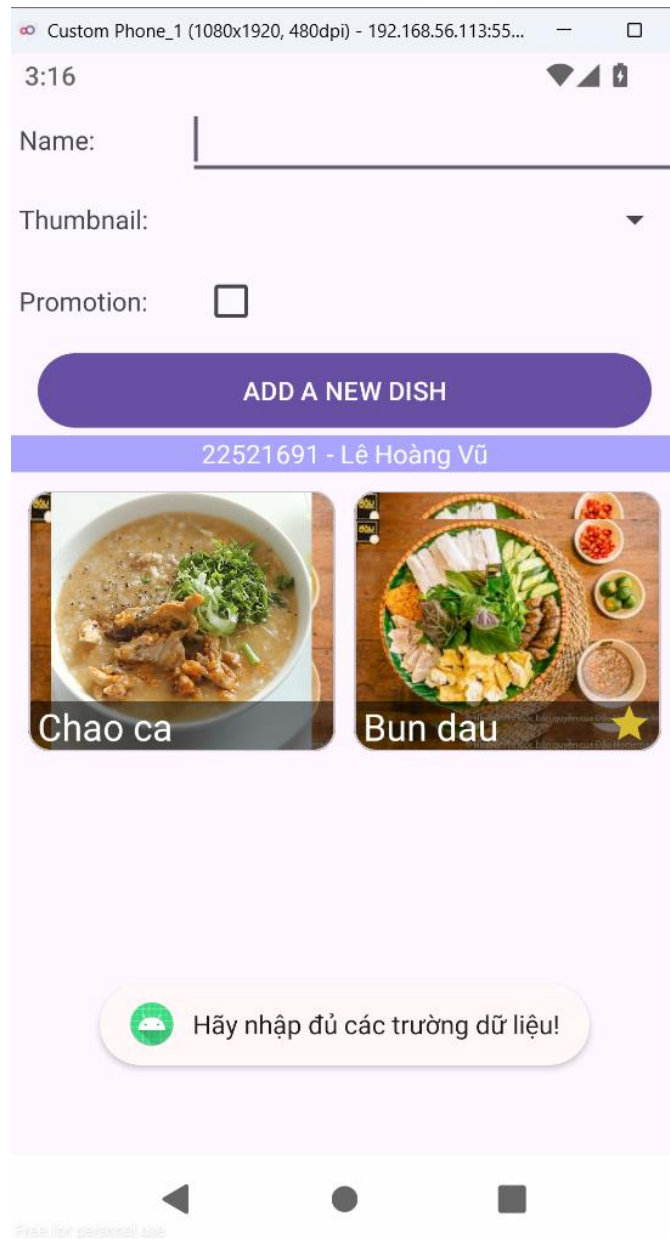
- Nhập tên món ăn, nếu như không chọn Promotion thì sẽ không có ngôi sao ở món ăn:



- Khi chọn Promotion thì sẽ có thêm dấu sao:



- Khi không nhập gì thì sẽ có Toast thời gian ngắn như hình dưới và không có dữ liệu nào được nhập thêm:



6. Câu hỏi 6

- Giao diện không khác gì ở câu 4, điểm khác là ở cách tạo Adapter và xử lý ở bên phía Adapter:
 - Tạo Adapter:

```
myArrayList = new ArrayList<>();  
myArrayAdapter = new MyArrayAdapter(myArrayList);  
rv.setLayoutManager(new LinearLayoutManager(context, this));  
rv.setAdapter(myArrayAdapter);
```

- Xử lý Adapter:

```

public class MyArrayAdapter extends RecyclerView.Adapter<MyArrayAdapter.MyViewHolder> { 3 usages
    private ArrayList<Employee> myList; 3 usages

    public MyArrayAdapter(ArrayList<Employee> myList) { 1 usage
        this.myList = myList;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.staff_item, parent, attachToRoot: false);
        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
        Employee employee = myList.get(position);
        holder.fullName.setText(employee.getName());

        if (employee.getPosition() != null) {
            holder.img.setVisibility(View.VISIBLE);
            holder.positionText.setVisibility(View.GONE);
        } else {
            holder.img.setVisibility(View.GONE);
            holder.positionText.setVisibility(View.VISIBLE);
        }

        // Đổi màu nền cho các item
        if (position % 2 == 1) {
            holder.linearLayout.setBackgroundColor(holder.linearLayout.getContext().getResources().getColor(R.color.light_blue));
        } else {
            holder.linearLayout.setBackgroundColor(holder.linearLayout.getContext().getResources().getColor(android.R.color.transparent));
        }
    }
}

```

```

@Override
public int getItemCount() {
    return myList.size();
}

public static class MyViewHolder extends RecyclerView.ViewHolder { 4 usages
    TextView fullName; 2 usages
    ImageView img; 3 usages
    TextView positionText; 3 usages
    LinearLayout linearLayout; 5 usages

    public MyViewHolder(@NonNull View itemView) { 1 usage
        super(itemView);
        fullName = itemView.findViewById(R.id.item_employee_tv_fullname);
        img = itemView.findViewById(R.id.item_employee_iv_manager);
        positionText = itemView.findViewById(R.id.item_employee_iv_position);
        linearLayout = itemView.findViewById(R.id.linear_layout);
    }
}

```

YÊU CẦU CHUNG

1) Đánh giá

- Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
- Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
- Nộp báo cáo kết quả chi tiết những đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

2) Báo cáo

- File .PDF hoặc .docx. Tập trung vào nội dung, giải thích.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Avo)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: LabX_MSSV1. (trong đó X là Thứ tự buổi Thực hành).

Ví dụ: Lab01_21520001

- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT