

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 RECYCLERVIER

GVHD: Phan Xuân Thiện

Ngày thực hiện: 05/10/2024

THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT118.P13

STT	Họ và tên	MSSV	Email
1	Phạm Thị Thanh Vinh	22521680	22521680@gm.uit.edu.vn

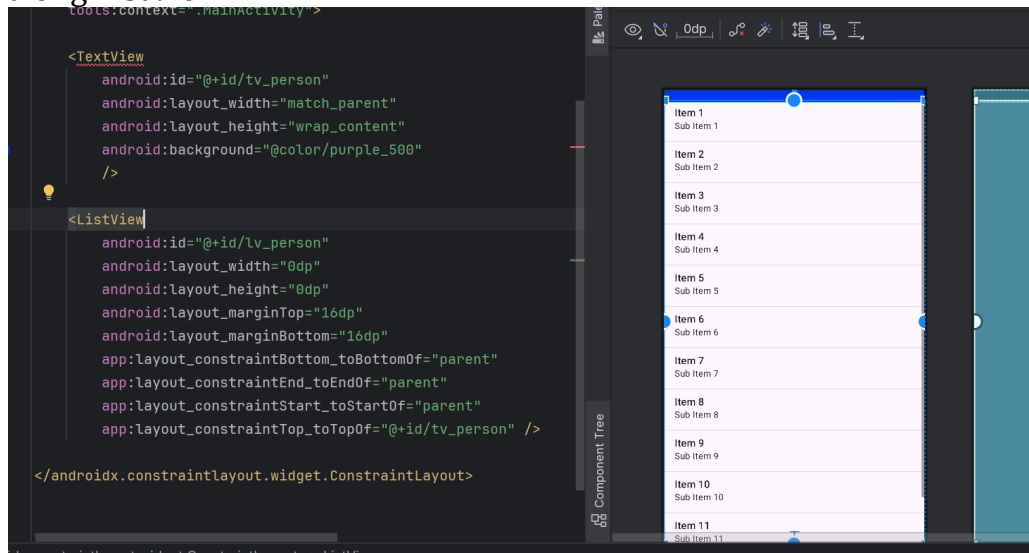
1. ĐÁNH GIÁ KHÁC:

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	8 ngày
Mã nguồn (Github):	https://github.com/Pttvinh253/Lab_NT118.git
Ý kiến (nếu có) + Khó khăn + Đề xuất ...	
Điểm tự đánh giá	10/10

BÁO CÁO CHI TIẾT

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

- Giao diện có 2 control:
 - + ListView: dùng để hiển thị mảng dữ liệu
 - + TextView: có màu xanh lục: Dùng để hiển thị vị trí và giá trị của phần tử được chọn trong ListView...



- Khai báo một mảng gồm 4 tên: {"Tèo", "Tý", "Bin", "Bo"}, sau đó dùng `ArrayAdapter` để liên kết danh sách (mảng tên) với `ListView`. Sử dụng sự kiện chọn tên trong `ListView`: Khi người dùng bấm vào một tên trong danh sách, sự kiện `OnItemClickListener` sẽ được kích hoạt, sau đó nó lấy vị trí của mục được chọn (biến `arg2`), hiển thị thông tin vị trí và tên trong `TextView` (`tvPerson`).

```
public class MainActivity extends AppCompatActivity {
    ListView lvPerson; 3 usages
    TextView tvPerson; 2 usages
    final String arr[] = {"Tèo", "Tý", "Bin", "Bo"}; 2 usages

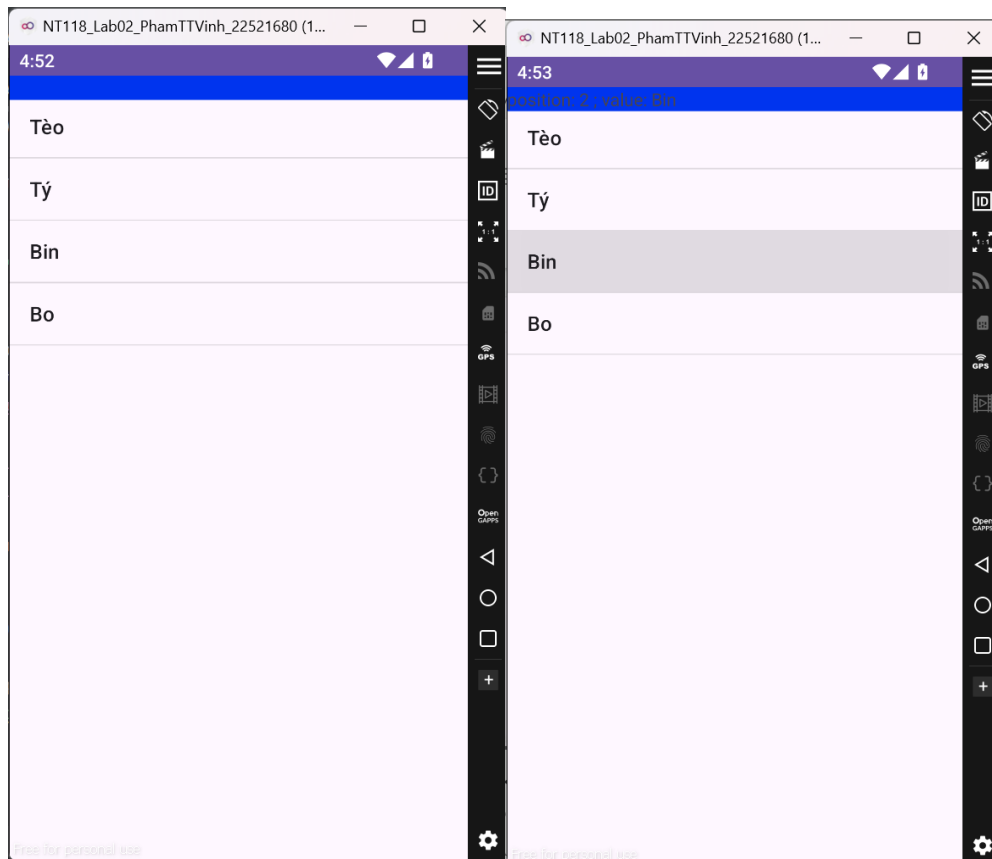
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tvPerson = findViewById(R.id.tv_person);
        lvPerson = findViewById(R.id.lv_person);

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

        lvPerson.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
                tvPerson.setText("position: " + arg2 + " ; value: " + arr[arg2]);
            }
        });
    }
}
```

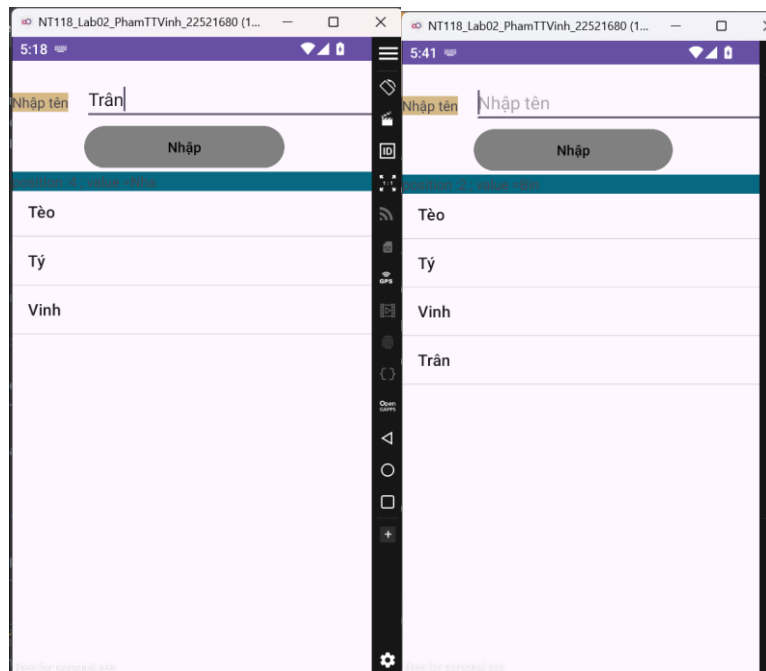
- Kết quả khi nhấn mục Bin thì Textview sẽ thay đổi thành position:2;value:Bin.



2. Sử dụng ArrayList và Listview control.

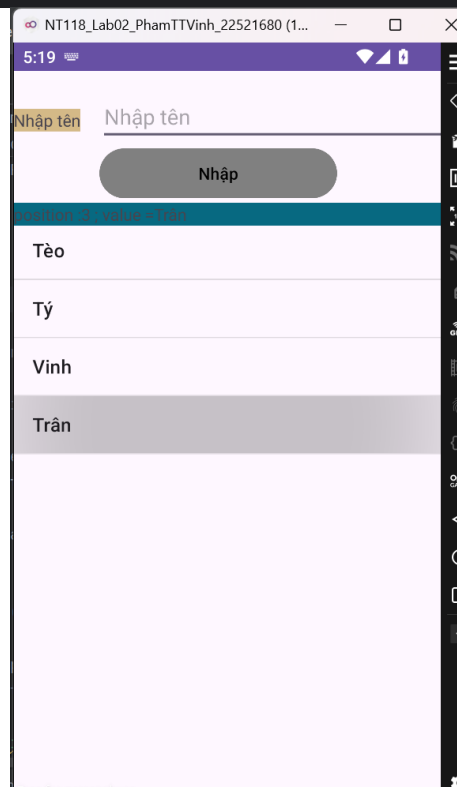
- Ứng dụng là một trình quản lý cho phép nhập dữ liệu vào danh sách. Hiển thị thông tin khi người dùng chọn tên. Xóa tên khỏi danh sách khi nhấn giữ.
- Sử dụng ArrayList để lưu trữ danh sách tên, gồm các tên ban đầu: "Tèo", "Tý", "Bin", "Bo". Sau đó dùng ArrayAdapter để liên kết dữ liệu từ ArrayList với ListView.
- Chức năng chính:
 - + Thêm tên mới:

```
protected void onCreate(Bundle savedInstanceState) {  
    btnAdd = (Button) findViewById(R.id.btnAdd);  
    arrayName = new ArrayList<>();  
    arrayName.add("Tèo");  
    arrayName.add("Tý");  
    arrayName.add("Bin");  
    arrayName.add("Bo");  
  
    adapter = new ArrayAdapter<String>(context, this, android.R.layout.simple_list_item_1, arrayName);  
    lvPerson.setAdapter(adapter);  
    btnAdd.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            String ten = edtName.getText().toString();  
            arrayName.add(ten);  
            adapter.notifyDataSetChanged();  
        }  
    });  
}
```



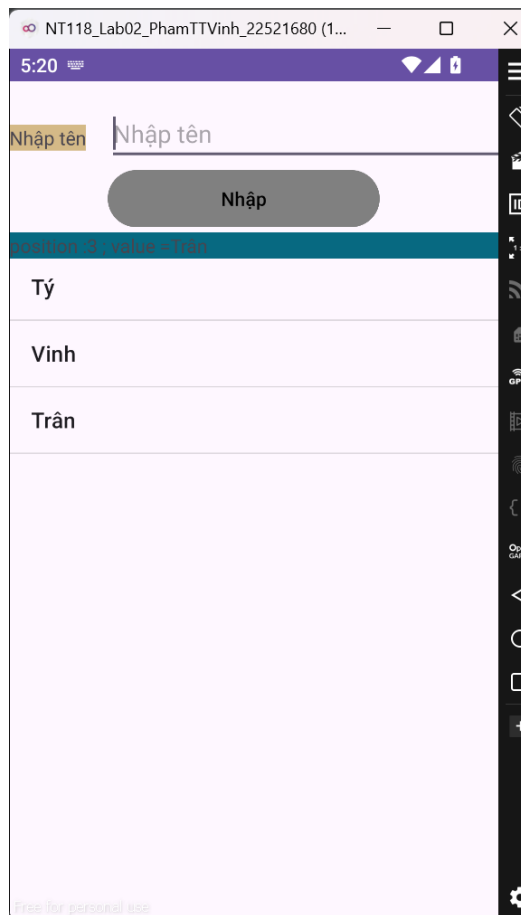
+ Hiển thị thông tin khi nhấn vào tên:

```
lvPerson.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?>parent, View view, int position, long id) {  
        Toast.makeText(MainActivity.this, "" + position, Toast.LENGTH_SHORT).show();  
        String value = lvPerson.getItemAtPosition(position).toString();  
        tvPerson.setText("position : " + position + " ; value = " + value);  
    }  
});
```



+ Xóa tên khi nhấn giữ:

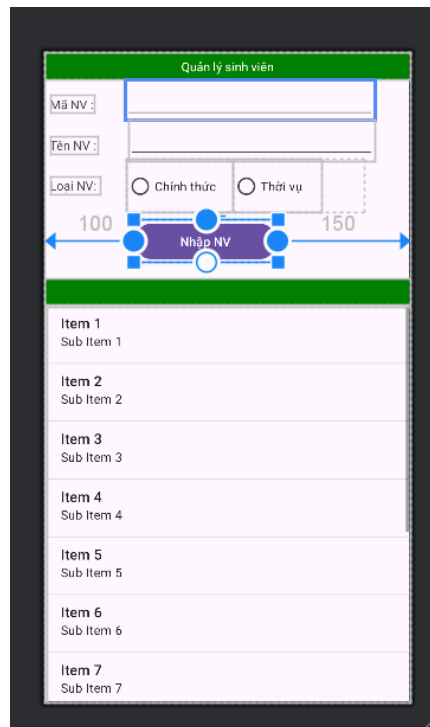
```
lvPerson.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  
    @Override no usages  
    public boolean onItemLongClick(AdapterView<?> adapterView, View view, int position, long id) {  
        arrayName.remove(position);  
        adapter.notifyDataSetChanged();  
        return false;  
    }  
});
```



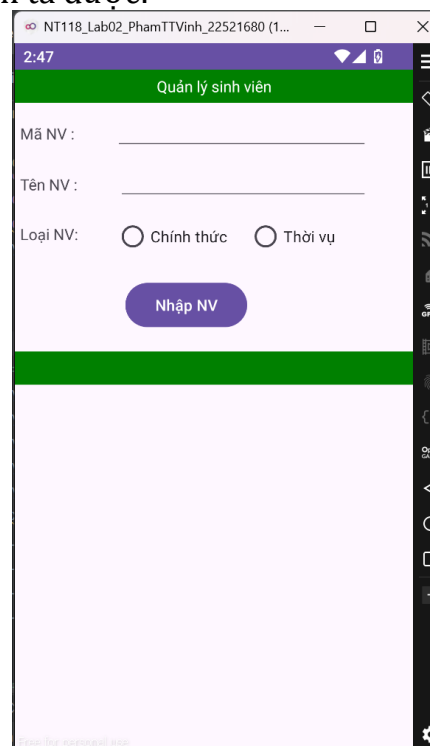
3. Sử dụng ArrayList và ListView mà từng phần tử trong ArrayList là các Object bất kỳ.

a) Nội dung 1

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



- Sau khi chạy chương trình ta được:



- Tạo sự kiện onClick() khi ấn vào nút “Nhập NV” để thêm dữ liệu và cập nhật dưới màn hình:

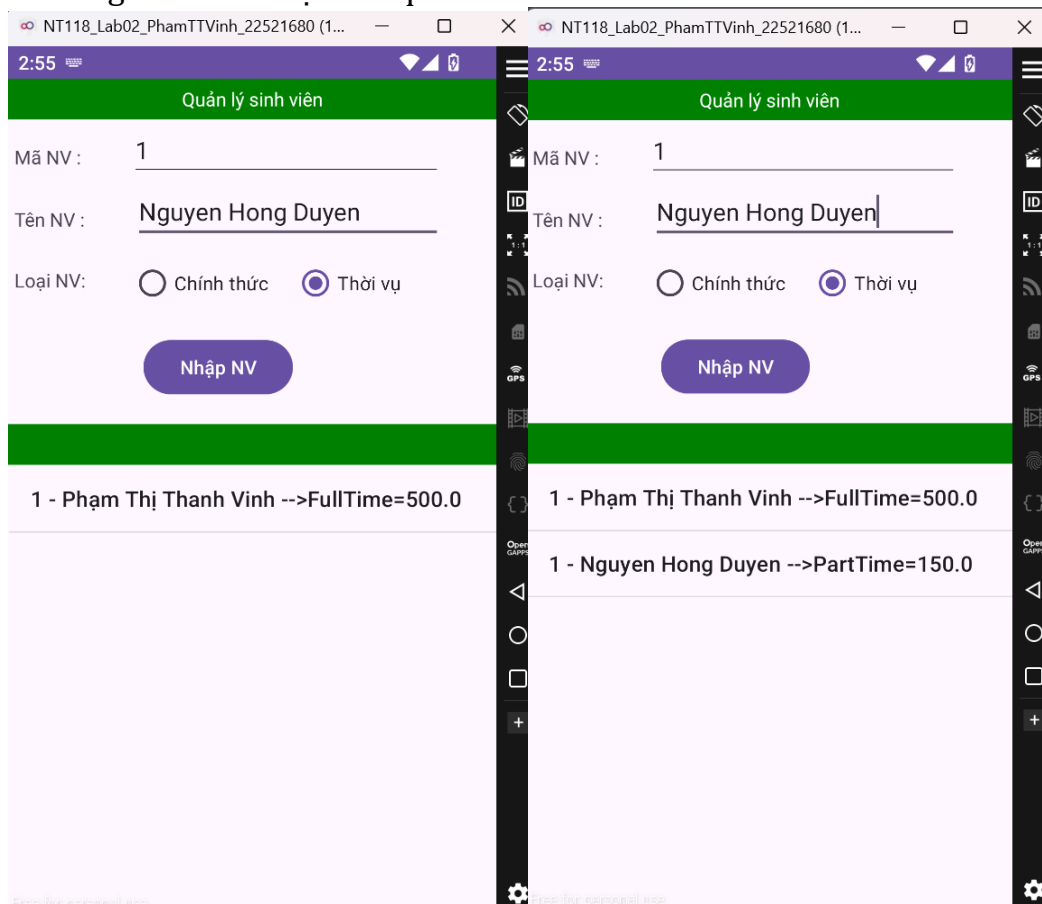


```

btnNhap.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        int radId = rgType.getCheckedRadioButtonId();
        String id = etId.getText().toString();
        String name = etName.getText().toString();
        if (radId == R.id.rd_chinhthuc) {
            //tạo instance là FullTime
            employee = new EmployeeFulltime();
        } else {
            //Tạo instance là Parttime
            employee = new EmployeeParttime();
        }
        employee.setId(id);
        employee.setName(name);
        //Đưa employee vào ArrayList
        employees.add(employee);
        //Cập nhật giao diện
        adapter.notifyDataSetChanged();
    }
}

```

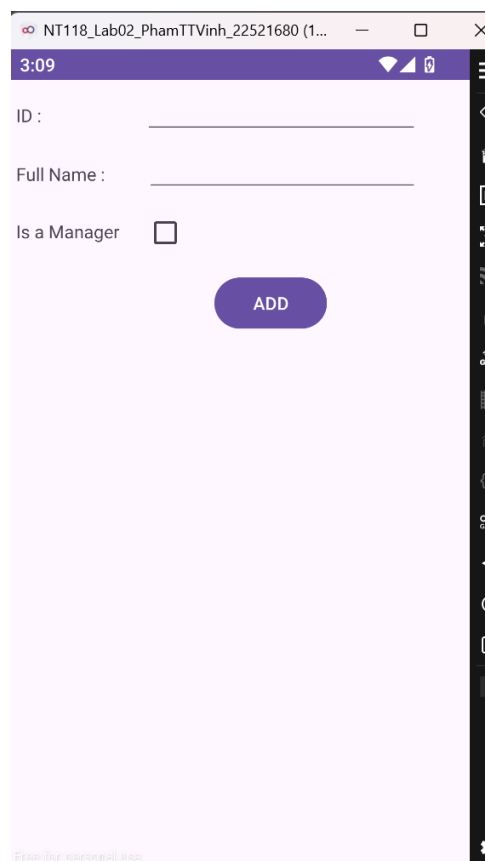
- Chạy chương trình và được kết quả:



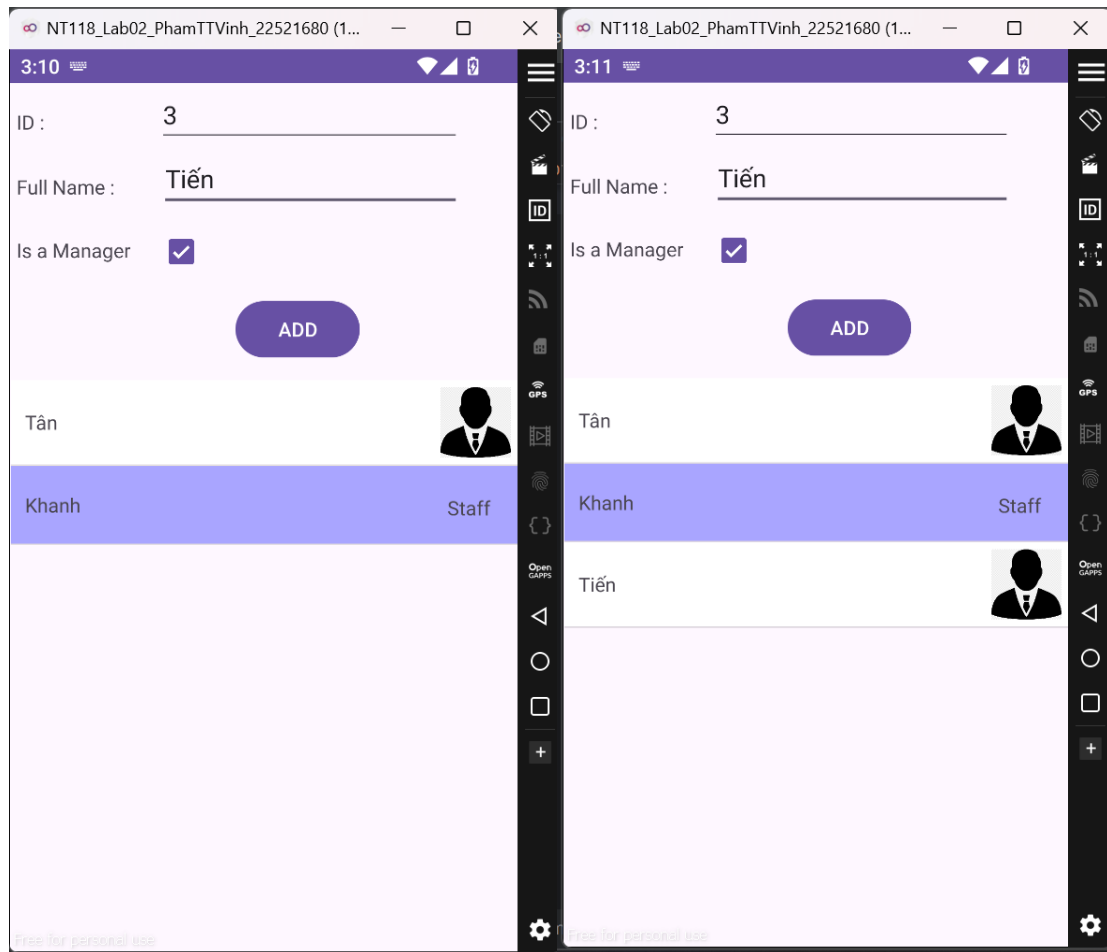
4. Sử dụng CustomAdapter cho Listview.

- Tạo sự kiện onClick() khi ấn vào nút “Nhập NV” để thêm dữ liệu và cập nhật dưới màn hình:

```
@Override
public void onClick(View view) {
    String name = edtName.getText().toString();
    Employee employee = new Employee();
    if (chbxManager.isChecked())
    {
        employee.setManager(true);
    }
    else
    {
        employee.setManager(false);
    }
    employee.setFullName(name);
    //Đưa employee vào ArrayList
    employees.add(employee);
    //Cập nhật giao diện
    adapter.notifyDataSetChanged();
}
```



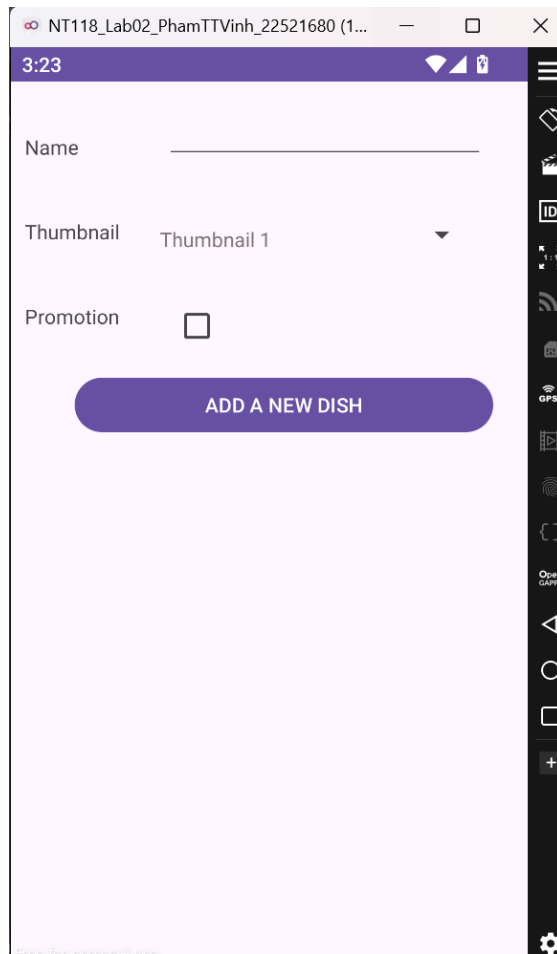
- Chạy chương trình và được kết quả:



5. Sử dụng GridView, Spinner.

b) Nội dung 1

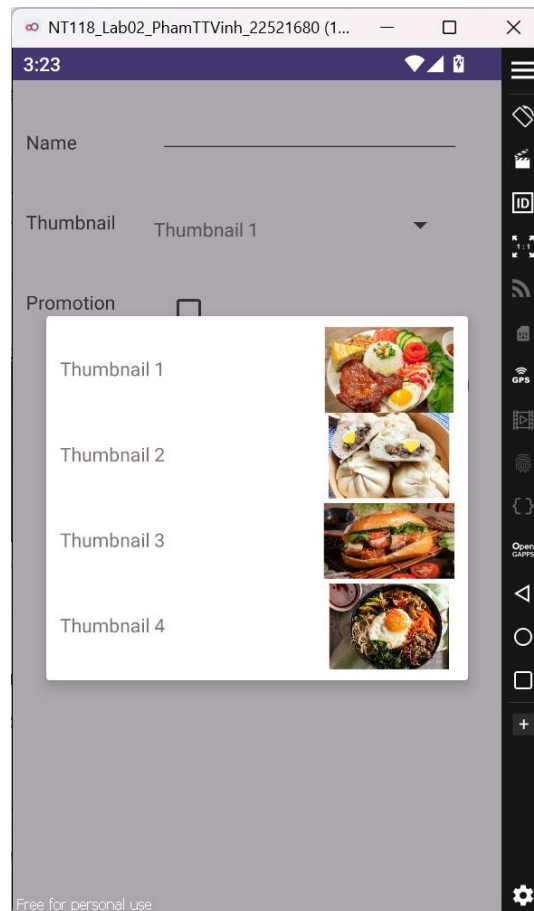
- Giao diện:



- Sử dụng sự kiện OnItemSelectedListener

```
spinnerDish.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
    @Override no usages  
  
    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {  
        Toast.makeText(context: MainActivity.this, text: "Added successfully", Toast.LENGTH_SHORT).show(  
            res_dish = arrayDish.get(i);  
    }  
})
```

- Sau khi nhấn vào Thumbnail ta có kết quả:



- Tạo sự kiện khi nhấn vào nút ADD A NEW DISH

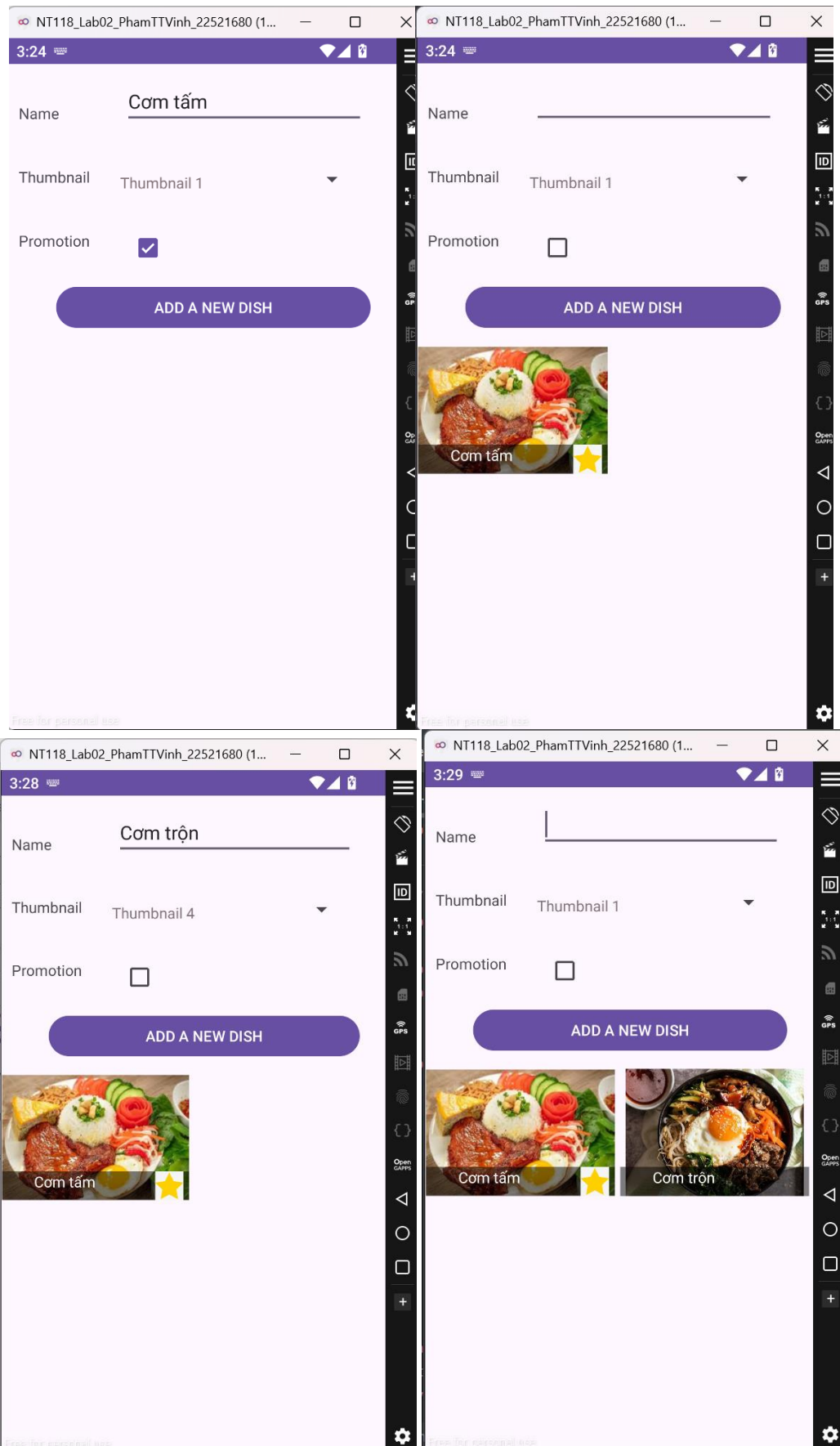
```
@Override
public void onClick(View view) {
    String name = edtName.getText().toString();
    Dish dish = new Dish();
    dish.setName(name);

    dish.setThumbnail(res_dish.getThumbnail());

    if (chbxPromotion.isChecked())
    {
        dish.setPromotion(true);
    }
    else
    {
        dish.setPromotion(false);
    }

    dishGv.add(dish);
    edtName.setText("");
    spinnerDish.setSelection(0);
    chbxPromotion.setChecked(false);
    adapter.notifyDataSetChanged();
}
```

- Điền tất cả trường thông tin và nhấn vào nút ADD A NEW DISH ta có được kết quả:



6. Thay việc sử dụng CustomAdapter cho Listview ở câu 4 bằng RecyclerView.

- Giao diện giống câu 4, khác ở cách tạo Adapter và xử lý ở bên phía Adapter:
- + Tạo Adapter:

```
employees = new ArrayList<>();
adapter = new EmployeeAdapter(context, this, employees);

rvEmployee.setLayoutManager(new LinearLayoutManager(context, this));
rvEmployee.setAdapter(adapter);
```

+ Xử lý:

```
public class EmployeeAdapter extends RecyclerView.Adapter<EmployeeAdapter.EmployeeViewHolder> { 3 usages
    private List<Employee> employees; 3 usages
    private Activity context; 3 usages
    public EmployeeAdapter(Activity context, List<Employee> employees) { 1 usage
        this.context = context;
        this.employees = employees;
    }
    @NonNull
    @Override
    public EmployeeViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.item_employee, parent, attachToRoot: false);
        return new EmployeeViewHolder(view);
    }
    @Override
    public void onBindViewHolder(@NonNull EmployeeViewHolder holder, int position) {
        Employee employee = employees.get(position);
        holder.tvFullName.setText(employee.getFullName() != null ? employee.getFullName() : "");
        if (employee.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(context.getString(R.string.staff));
        }
        if (position % 2 == 0) {
            holder.llParent.setBackgroundResource(R.color.white);
        } else {
            holder.llParent.setBackgroundResource(R.color.light_blue);
        }
    }
    @Override
    public int getItemCount() {
        return employees.size();
    }
    public static class EmployeeViewHolder extends RecyclerView.ViewHolder { 4 usages
        TextView tvFullName, tvPosition; 2 usages
        ImageView ivManager; 3 usages
        LinearLayout llParent; 3 usages
        public EmployeeViewHolder(@NonNull View itemView) { 1 usage
            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);
        }
    }
}
```