

I. GIỚI THIỆU NỘI DUNG

- Sử dụng ListView control với mảng dữ liệu định sẵn.
- Sử dụng ListView với mảng dữ liệu được lưu trong Xml:
- Sử dụng ArrayList và ListView control
- Sử dụng ArrayList và ListView mà từng phần tử trong ArrayList là các Object bất kỳ.
- Sử dụng CustomAdapter cho ListView
- Sử dụng GridView và Spinner

II. GIỚI THIỆU CÁC BƯỚC XÂY DỰNG LISTVIEW

Bước	Nội dung	Ví dụ
1	Khởi tạo đối tượng listview: findViewById từ file XML hoặc tạo bằng code	<pre>ListView lvPerson = (ListView) findViewById(R.id.lv_person);</pre>
2	Load/Khởi tạo mảng chứa dữ liệu sẽ được hiển thị trong listview	<pre>final String arr[] = {"Teo", "Ty", "Bin", "Bo"};</pre>
3	Xây dựng adapter	<pre>ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, arr);</pre>
4	SetAdapter cho listview	<pre>lvPerson.setAdapter(adapter);</pre>
5	Xử lý các thao tác trên listview (click, longClick,...)	<pre>lvPerson.setOnItemClickListener(new AdapterView.OnItemClickListener() { public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) { //đổi số arg2 là vị trí phần tử trong Data Source (arr) tvSelection.setText("position : " + arg2 + " ; value = " + arr[arg2]); } });</pre>

Sinh viên thực hành các bước xây dựng listview theo hướng dẫn trực tiếp của giảng viên tại lớp.

III. YÊU CẦU THỰC HÀNH

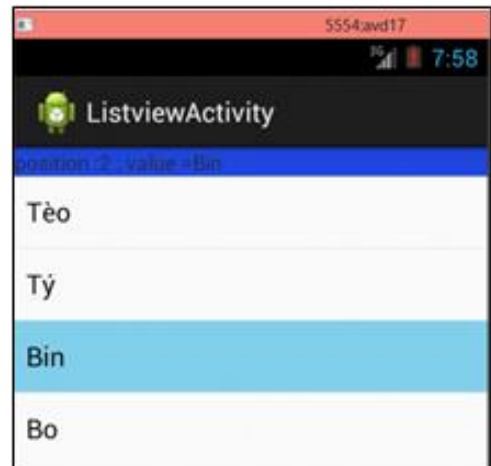
1. Trường hợp 1: Sử dụng ListView control với mảng dữ liệu định sẵn

Sử dụng ListView control với mảng dữ liệu định sẵn. Xây dựng Listview như hình minh họa:

- Giao diện trên 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.



2. Trường hợp 3: Sử dụng ArrayList và Listview control

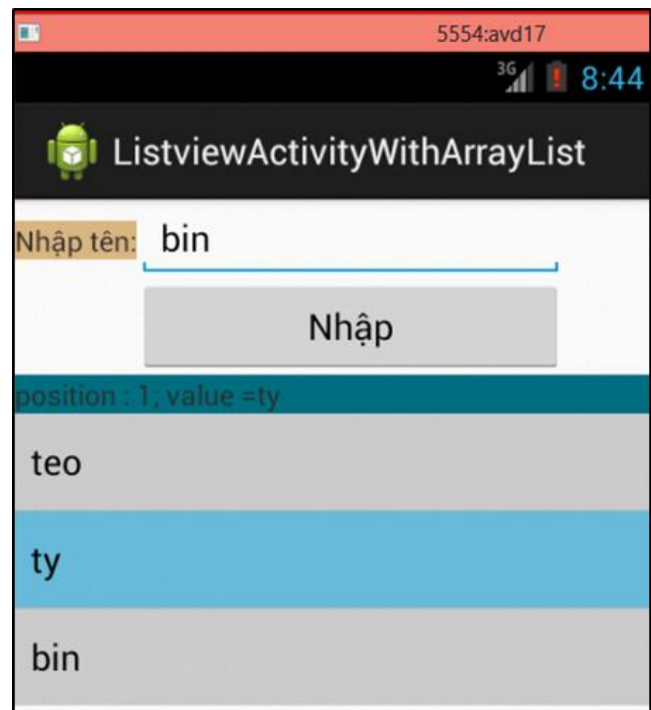
Xây dựng ứng dụng như sau:

- **Mô tả:**

+ Nhập dữ liệu và nhấn nút “Nhập” thì sẽ đưa vào ArrayList và hiển thị lên ListView.

+ Nhấn vào phần tử nào thì hiển thị vị trí và giá trị của phần tử đó lên TextView

+ Nhấn thật lâu (long click) vào phần tử nào đó trên ListView thì sẽ xóa phần tử đó.



- **Hướng dẫn:**

```
//1. Tạo ArrayList object
names = new ArrayList<String>();

...

//4. Xử lý sự kiện nhấn nút Nhập
btnSubmit.setOnClickListener(new View.OnClickListener() {
```

```
public void onClick(View arg0) {  
    //Thêm dữ liệu mới vào arraylist  
  
    ...  
  
    //Cập nhật dữ liệu mới lên giao diện  
    adapter.notifyDataSetChanged();  
}  
});  
//5. Xử lý sự kiện chọn một phần tử trong ListView  
lvPerson.setOnItemClickListener(...);  
//6. xử lý sự kiện Long click  
lvPerson.setOnItemLongClickListener(...);  
}
```

3. Trường hợp 4: Sử dụng ArrayList và ListView mà từng phần tử trong ArrayList là các Object bất kỳ:

Xây dựng ứng dụng theo mô tả sau: Có 2 loại nhân viên: Nhân viên chính thức (EmployeeFullTime) và nhân viên thời vụ (EmployeePartime). Mỗi loại nhân viên sẽ có cách tính lương khác nhau. Mỗi nhân viên có phương thức toString để xuất thông tin, Nội dung xuất khác nhau. Xem giao diện chương trình:

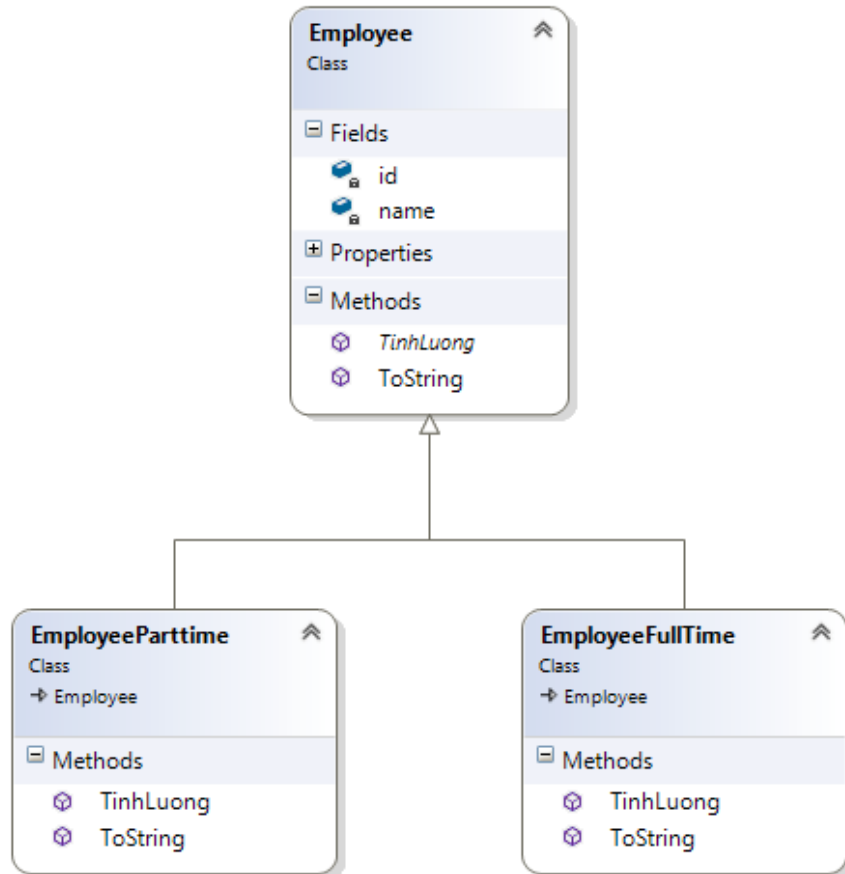
5554:avd17tes
3G 8:38
Vidu_ListView_ArrayList_Object
Quản lý nhân viên

Mã NV: m4
Tên NV: Nguyen thi ngoc uyen
Loại NV: ☐ Chính thức ☒ Thời vụ
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

Hướng dẫn:

- Sơ đồ các class lưu thông tin nhân viên:



- Thông tin mã màu:

```
<resources>
<color name="green">#008000</color>
<color name="white">#FFFFFF</color>
</resources>
```

- Override hàm tinhLuong và toString trên các class EmployeeFullTime và EmployeePartTime để hiển thị thông tin tương ứng:

```
public class EmployeeFulltime extends Employee {
    @Override
    public double tinhLuong() {...}

    @Override
    public String toString() {return super.toString() + ...;}
}
```

```
public class EmployeeParttime extends Employee {
    @Override
    public double tinhLuong() {...}

    @Override
    public String toString() {return super.toString() + ...}
}
```

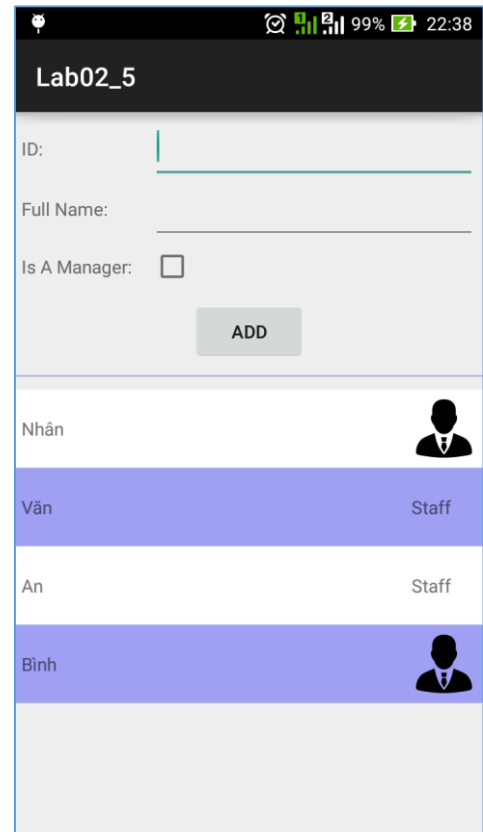
- Hàm xử lý thêm một nhân viên mới

```
public void addNewEmployee() {
    //Lấy ra đúng 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_chinhthuc) {
        //tạo instance là FullTime
        employee = new EmployeeFulltime();
    } else {
        //Tạo instance là Parttime
        employee = new EmployeeParttime();
    }
    //FullTime hay Parttime thì 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 nhập giao diện
    adapter.notifyDataSetChanged();
}
```

4. Trường hợp 5: Sử dụng CustomAdapter cho Listview

Xây dựng ứng dụng với giao diện như sau:

Người dùng nhập thông tin ở trên, sau đó nhấn Add. Thông tin employee sẽ được hiện xuống listview. Nếu là Manager thì hiện thêm icon manager ở bên phải ngược lại hiện chữ Staff. Ngoài ra, giữa 2 employee liên tiếp trong listview sẽ được hiện background màu khác nhau cho dễ nhìn.



Hướng dẫn:

- Thông tin mã màu và các resource files:

```
<resources>
  <color name="white">#FFFFFF</color>
  <color name="light_blue">#550000FF</color>
</resources>
<resources>
  <!-- Default screen margins, per the Android Design guidelines. -->
  <dimen name="activity_horizontal_margin">16dp</dimen>
  <dimen name="activity_vertical_margin">16dp</dimen>
  <dimen name="margin_base">5dp</dimen>
  <dimen name="margin_basex2">10dp</dimen>
</resources>
<resources>
  <string name="app_name">Lab02_5</string>
  <string name="id">ID:</string>
  <string name="full_name">Full Name:</string>
  <string name="is_manager">Is A Manager:</string>
  <string name="staff">Staff</string>
  <string name="add">Add</string>
</resources>
```

- Save ảnh sau thành ic_manager.png rồi kéo thả vào thư mục drawable
 - Để sử dụng CustomAdapter còn xây dựng 2 file: item_employee.xml và EmployeeAdapter.java. Trong đó, file item_employee.xml là file định nghĩa nội dung view hiển thị cho 1 dòng (1 nhân viên) trong listview. File EmployeeAdapter.java là file Custom Adapter để hiển thị nội dung lên listview.
- + File item_employee.xml cần chứa:
- 1 textview để hiển thị tên nhân viên
 - 1 textview hiển thị Staff dành cho nhân viên
 - 1 imageview để hiển thị icon dành cho manager



```
...  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal">  
  
    <TextView  
        android:id="@+id/item_employee_tv_fullname"  
        android:layout_width="0dp"  
        android:layout_height="wrap_content"  
        android:layout_gravity="center_vertical"  
        android:layout_weight="1"  
        android:ellipsize="end"  
        android:singleLine="true" />  
  
    <TextView  
        android:id="@+id/item_employee_tv_position"  
        android:layout_width="50dp"  
        android:layout_height="wrap_content"  
        android:layout_gravity="center_vertical"  
        android:singleLine="true" />  
  
    <ImageView  
        android:id="@+id/item_employee_iv_manager"  
        android:layout_width="50dp"  
        android:layout_height="50dp"  
        android:scaleType="centerCrop"  
        android:src="@drawable/ic_manager"  
        android:visibility="gone"/>  
</LinearLayout>  
  
...
```

+ File EmployeeAdapter.java. Sinh viên tham khảo nội dung:

```
public class EmployeeAdapter extends ArrayAdapter<Employee> {
    private Activity context;

    public EmployeeAdapter(Activity context, int layoutID, List<Employee>
objects) {
        super(context, layoutID, objects);
        this.context = context;
    }

    @Override
    public View getView(final int position, View convertView, ViewGroup parent)
{
        if (convertView == null) {
            convertView =
LayoutInflater.from(context).inflate(R.layout.item_employee, null,
false);
        }

        // Get item
        Employee employee = getItem(position);

        // Get view
        TextView tvFullName = (TextView)
convertView.findViewById(R.id.item_employee_tv_fullname);
        TextView tvPosition = (TextView)
convertView.findViewById(R.id.item_employee_tv_position);
        ImageView ivManager = (ImageView)
convertView.findViewById(R.id.item_employee_iv_manager);
        LinearLayout llParent = (LinearLayout)
convertView.findViewById(R.id.item_employee_ll_parent);

        // Set fullname
        if (employee.getFullName() != null) {
            tvFullName.setText(employee.getFullName());
        }
        else tvFullName.setText("");

        // If this is a manager -> show icon manager. Otherwise, show Staff in
tvPosition
        if (employee.isManager())
        {
            ivManager.setVisibility(View.VISIBLE);
            tvPosition.setVisibility(View.GONE);
        }
        else
        {
            ivManager.setVisibility(View.GONE);
            tvPosition.setVisibility(View.VISIBLE);
            tvPosition.setText(context.getString(R.string.staff));
        }

        // Show different color backgrounds for 2 continuous employees
        if (position%2==0)
        {
            llParent.setBackgroundResource(R.color.white);
        }
        else
        {

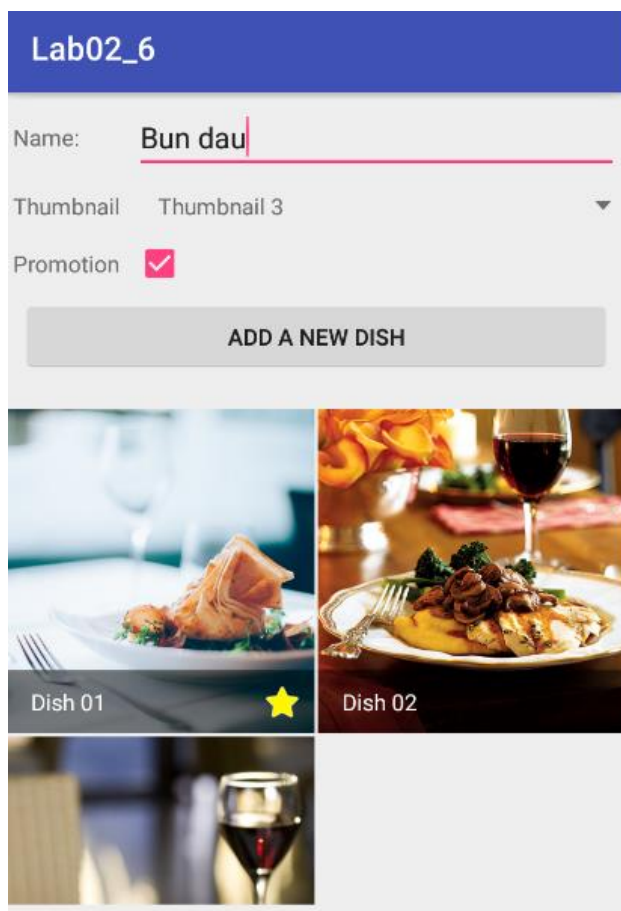
```



```
llParent.setBackgroundResource(R.color.light_blue);  
}  
  
return convertView;  
}  
}
```

5. Sử dụng GridView, Spinner

Các bước thiết đặt adapter để hiển thị nội dung lên gridview và spinner trong Android cũng tương tự trên listview. Sinh viên viết ứng dụng theo mô tả sau:



- Ứng dụng hỗ trợ chức năng thêm món ăn mới gồm các thông tin: tên món ăn, hình đại diện, và thông tin có khuyến mãi hay không. Sau khi thêm một món ăn, các trường name, thumbnail, promotion được reset về trạng thái ban đầu (name trống, thumbnail, promotion hiển thị giá trị mặc định). Thông báo “Added successfully” được hiển thị dưới dạng Toast.

- Danh sách các món ăn được hiển thị bằng gridview theo thiết kế như hình gồm 2 cột. Mỗi món ăn hiển thị hình đại diện, tên món ăn, nếu có khuyến mãi thì thêm icon star. Nếu tên món ăn quá dài, nội dung tên sẽ hiển thị trên gridview dạng chữ chạy.

- Hình đại diện được chọn từ spinner chứa 4 hình có sẵn. Danh sách này được

hiển thị ở dạng dialog (không phải dạng dropdown sổ xuống thông thường) như hình minh họa gồm tên hình và hình. Khi một hình được chọn chỉ có tên được hiển thị lên spinner.

Hướng dẫn:

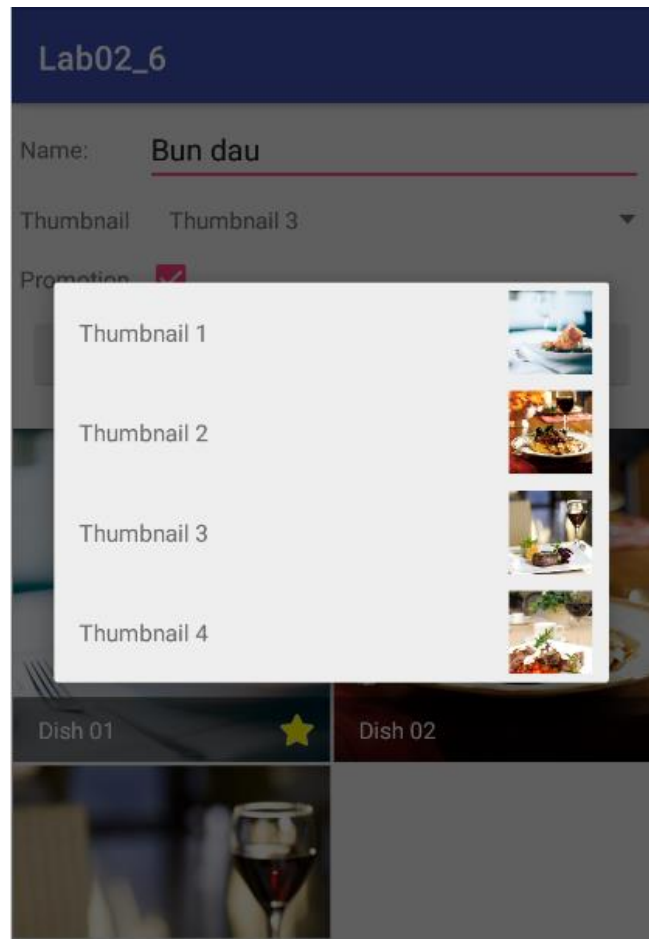
- Sinh viên download các resources cần thiết [TAI ĐÂY](#):
- Định nghĩa enum chứa các loại thumbnails

```
public enum Thumbnail{  
    Thumbnail1("Thumbnail 1", R.drawable.first_thumbnail),  
    Thumbnail2("Thumbnail 2", R.drawable.second_thumbnail),  
    Thumbnail3("Thumbnail 3", R.drawable.third_thumbnail),  
    Thumbnail4("Thumbnail 4", R.drawable.fourth_thumbnail);  
  
    private String name;  
    private int img;  
  
    Thumbnail(String name, int img) {  
        this.name = name;  
        this.img = img;  
    }  
  
    public String getName() { return name; }  
  
    public int getImg() { return img; }  
}
```

- Tạo class Dish có các thuộc tính và phương thức cần thiết.
- Tạo custom adapter cho gridview hiển thị các món ăn (item_dish.xml, DishAdapter.java).
- Tạo custom adapter cho spinner hiển thị danh sách thumbnails (item_thumbnail.xml, item_selected_thumbnail.xml, ThumbnailAdapter.java). Trong ThumbnailAdapter cần override các hàm:
 - + getDropDownView() sử dụng view item_thumbnail.xml để hiển thị nội dung thumbnail trong dropdown (dạng dialog) chứa danh sách thumbnail cho người dùng chọn.
 - + getView() sử dụng view item_selected_thumbnail.xml để hiển thị nội dung thumbnail được chọn lên spinner.

* GV lưu ý SV thêm một số nội dung:

- Hướng dẫn sử dụng chức năng generate code tự động khi xây dựng class Dish.



- Các cách xử lý khi hiển thị tên dài trên textview: chữ chạy (marquee), ngắt chữ (end, start, middle), xuống dòng.
- Phân biệt chế độ VISIBLE, GONE, INVISIBLE của một view.
- Các chế độ co giãn hình ảnh trên imageview: centerCrop, fitXY, fitCenter...

6. RecyclerView

Về nhà tìm hiểu và so sánh Listview và RecyclerView. Làm lại bài ở mục 4. (4. Trường hợp 5: Sử dụng CustomAdapter cho Listview) bằng RecyclerView

IV. THAM KHẢO THÊM (Tự xem thêm)

1. QuickAdapter

Tham khảo tại: <https://github.com/JoanZapata/base-adapter-helper>

2. Glide

Tham khảo tại: <https://github.com/bumptech/glide>

3. Gson

Tham khảo tại: <https://github.com/google/gson>