



## **THIẾT KẾ GIAO DIỆN ANDROID**

### **BÀI 5: PICKER – DIALOG - TOAST**

- ⊙ Kết thúc bài học này bạn có khả năng
  - ⊙ Hiểu được các loại Picker
  - ⊙ Biết được các Dialog
  - ⊙ Biết được cách tạo Toast



## Phần I: Picker, Dialog

 A DatePicker, TimePicker

 B DatePickerDialog, TimePickerDialog

 C ProgressDialog

## Phần II: AlertDialog, Toast

 D AlertDialog

 E AlertDialog dạng List

 F AlertDialog dạng SingleChoice

 G AlertDialog dạng MultiChoice

 H AlertDialog tùy chỉnh

 Toast và Toast tùy chỉnh





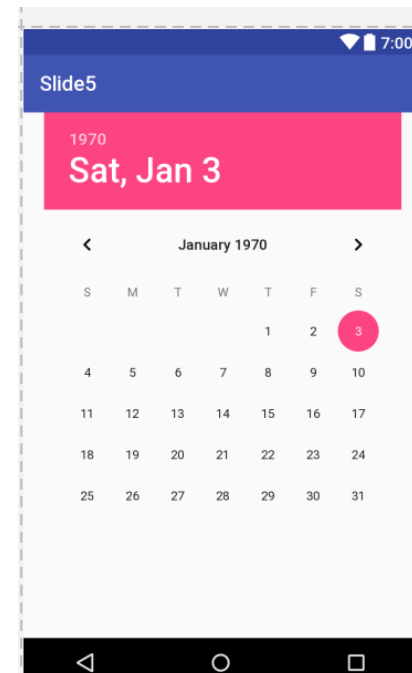
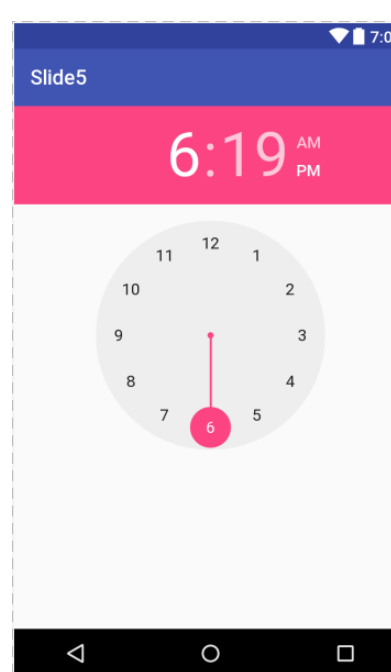
## **BÀI 5: PICKER – DIALOG - TOAST**

---

### **PHẦN 1: PICKER**

□ Về chọn ngày tháng, hay giờ thì có 2 dạng widget.

- ❖ Một dạng nằm trực tiếp trên giao diện là DatePicker và TimePicker.
- ❖ Dạng hai là kiểu dialog là DatePickerDialog và TimerPickerDialog.

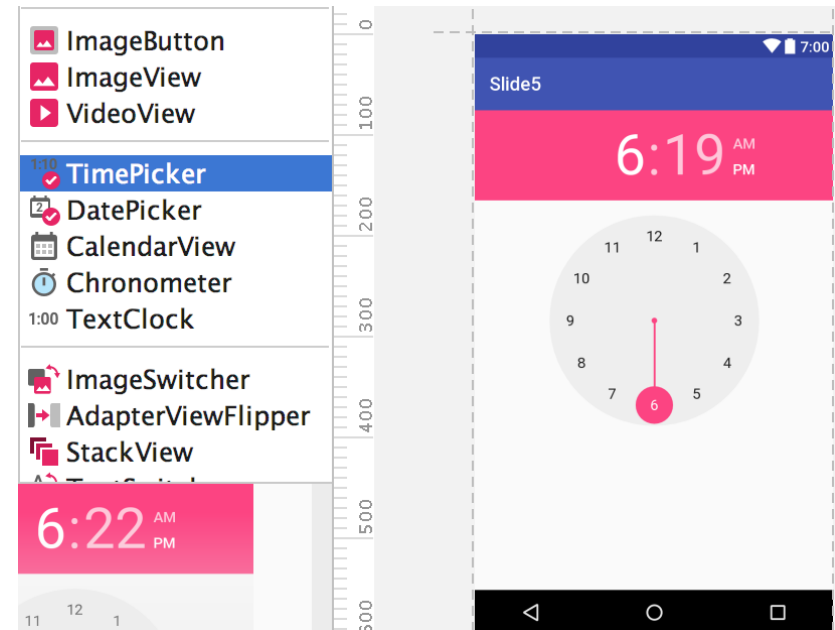


❑ TimerPicker kéo vào trên giao diện

❑ Bắt sự kiện thay đổi giờ bằng  
setOnTimeChangeListener

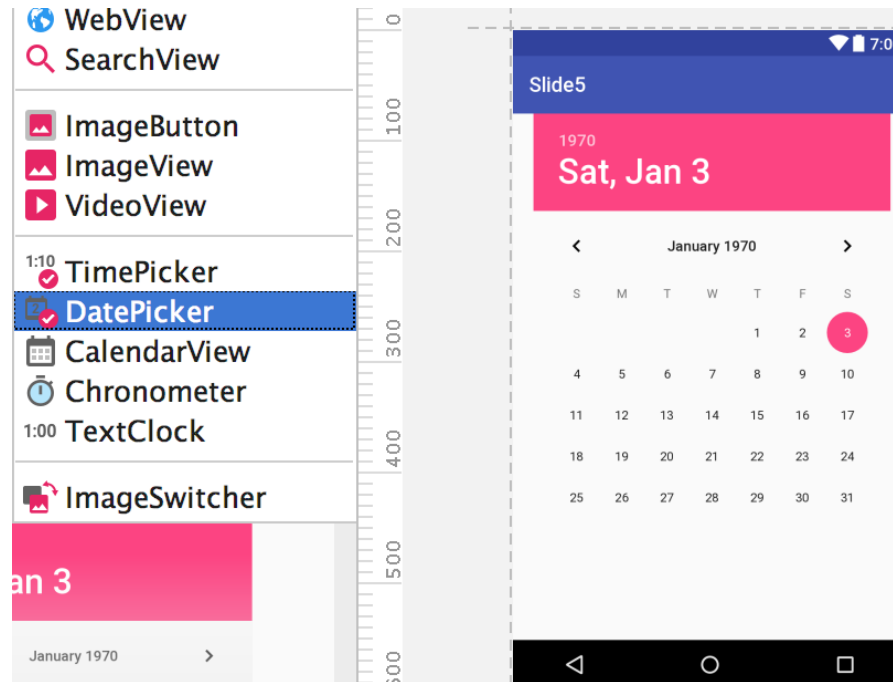
❑ Trong hàm onTimeChanged:

- ❖ Đối số thứ 2 là giờ (0-23)
- ❖ Đối số thứ 3 là phút (0-59)



```
timepicker.setOnTimeChangeListener(new OnTimeChangeListener() {
    @Override
    public void onTimeChanged(TimePicker timePicker, int i, int i1) {
        Toast.makeText(MainActivity.this, i + ":" + i1, Toast.LENGTH_SHORT)
    }
});
```

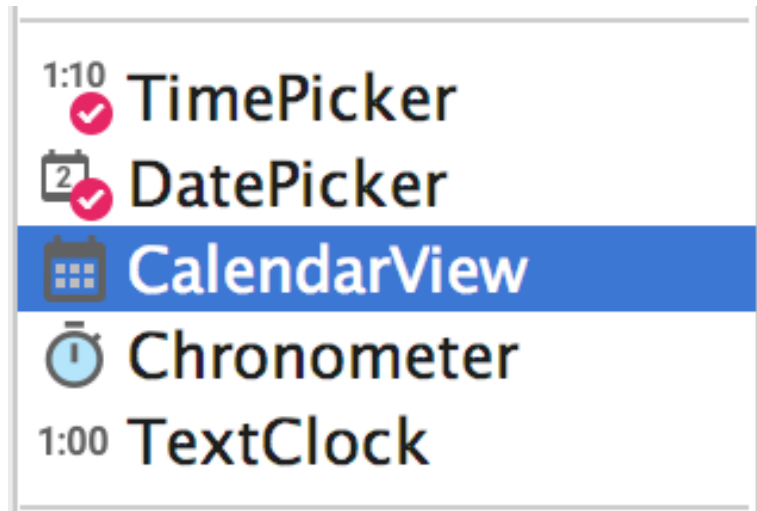
## ❑ DatePicker kéo trên giao diện



- ❑ DatePicker có các hàm getDateOfMonth(), getMonth(), getYear() để lấy ngày, tháng năm

- ❑ Sv nên tìm hiểu thêm các Class khác hỗ trợ về ngày giờ như CalendarView, Chronometer, TextClock, CountDownTimer, SimpleDateFormat

...







## **BÀI 5: PICKER – DIALOG - TOAST**

---

### **PHẦN 2: DIALOG**

- ❑ Hộp thoại là 1 cửa sổ nhỏ nhắc người dùng đưa ra quyết định hoặc nhập, chọn một thông tin nào đó.
- ❑ Có các dạng DatePickerDialog, TimerPicker, AlertDialog, ProgressDialog...

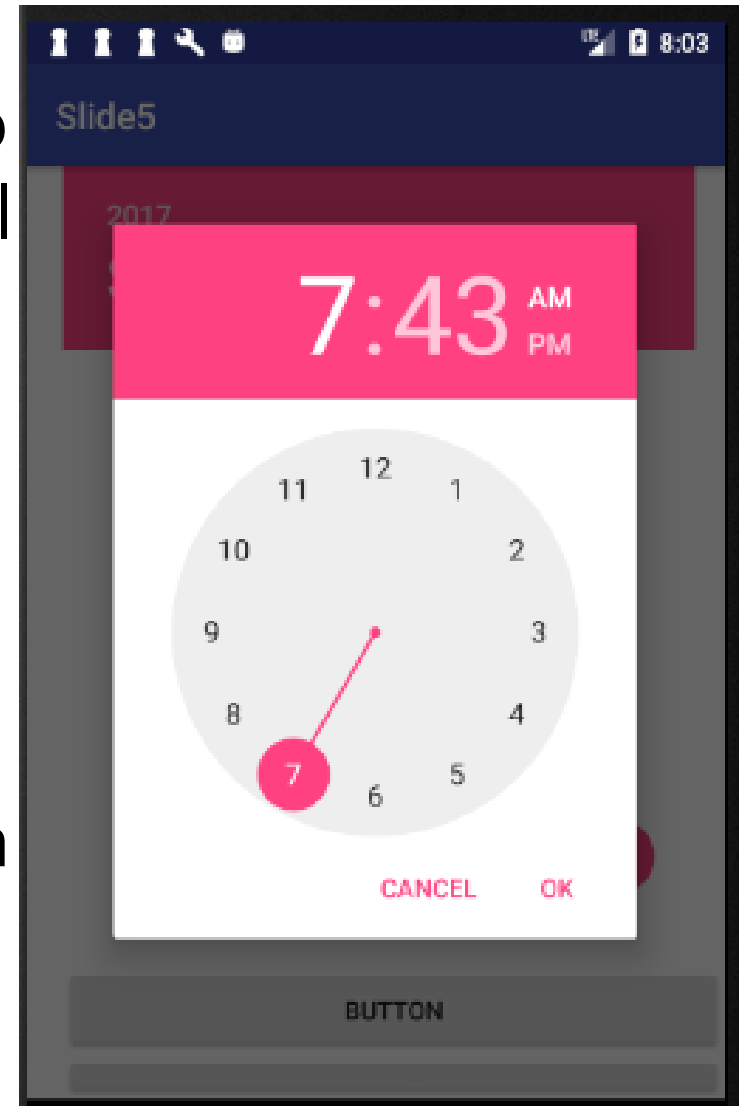
- ❑ Tạo một đối tượng calendar để lấy ngày giờ hiện tại

```
Calendar lịch=Calendar.getInstance();
```

- ❑ Tạo đối tượng TimePickerDialog và show nó

```
TimePickerDialog tpdialog=new TimePickerDialog(this,  
    new TimePickerDialog.OnTimeSetListener() {  
        @Override  
        public void onTimeSet(TimePicker view, int hourOfDay, int minute)  
            btngay.setText(hourOfDay+"/"+minute);  
    }  
),  
    lịch.get(Calendar.HOUR),  
    lịch.get(Calendar.MINUTE),  
    false);  
tpdialog.show();
```

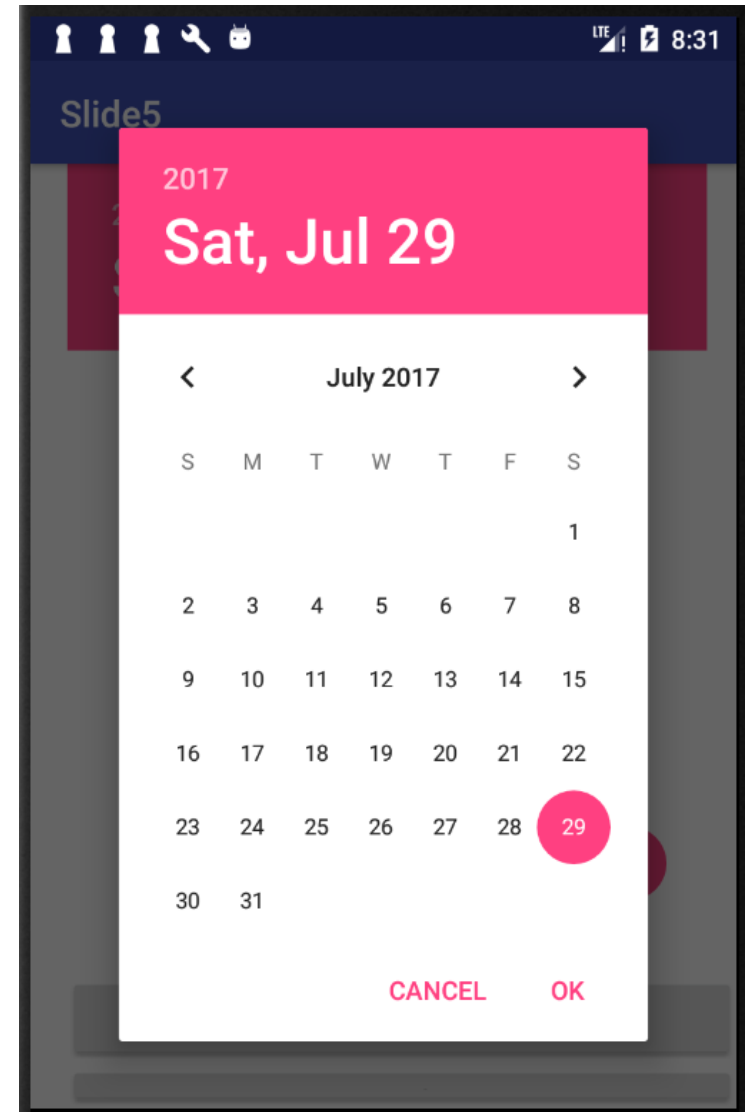
- ❑ TimePickerDialog nhận vào đối số thứ 2 là một Listener có hàm onTimeSetListener để call back khi chọn xong trên Dialog và nhấn nút OK. Đối số 3,4 là giờ phút mặc định của Dialog khi hiện lên, đối số 5 là is24H
- ❑ Hàm onTimeSet có đối số là hourOfDay và minute là giờ và phút khi người dùng chọn xong và nhấn OK



- ❑ Tạo một đối tượng calendar để lấy ngày giờ hiện tại  
`Calendar lịch=Calendar.getInstance();`
- ❑ Tạo đối tượng DatePickerDialog và show nó

```
DatePickerDialog dpdialog=new DatePickerDialog(this,  
    new DatePickerDialog.OnDateSetListener() {  
        @Override  
        public void onDateSet(DatePicker view, int year,  
            int monthOfYear,int dayOfMonth) {  
            btgio.setText(dayOfMonth+"/"+(monthOfYear+1)+"/"+year);  
        }  
    },  
    lịch.get(Calendar.YEAR),  
    lịch.get(Calendar.MONTH),  
    lịch.get(Calendar.DAY_OF_MONTH));  
dpdialog.show();
```

- ❑ DatePickerDialog có đối số 2 là  
OnDateSetListener là callback sau khi người dùng chọn xong. Đối số 3,4,5 là ngày tháng năm mặc định khi dialog hiện lên.
- ❑ Hàm OnDateSet có các đối số ngày tháng năm của người sử dụng sau khi đã chọn xong

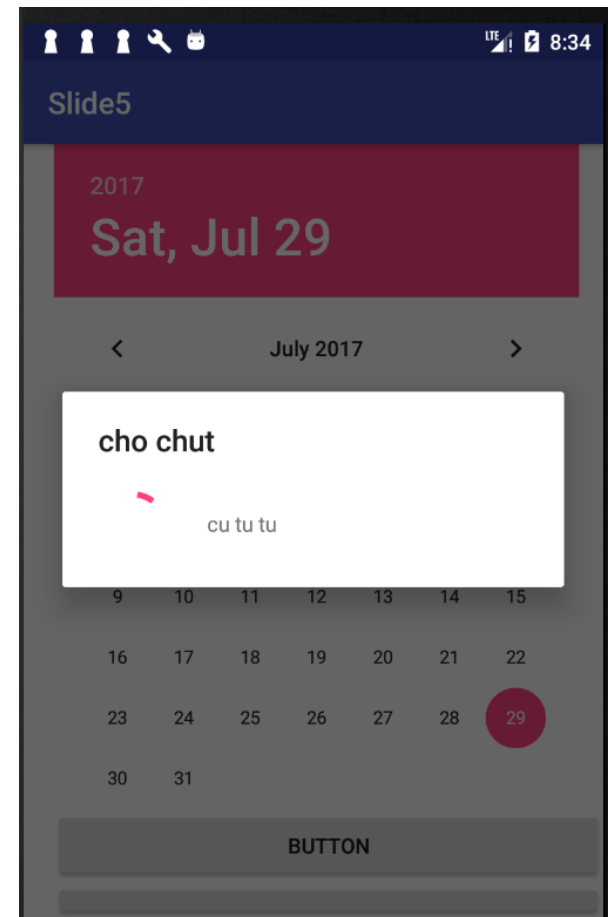


## ❑ Show progressDialog

```
ProgressDialog progressDialog= ProgressDialog.show(  
    Main2Activity.this, "cho chut", "cu tu tu");
```

## ❑ Ẩn progressDialog

```
progressDialog.dismiss();
```





# DEMO

DatePicker,TimePicker

DatePickerDialog

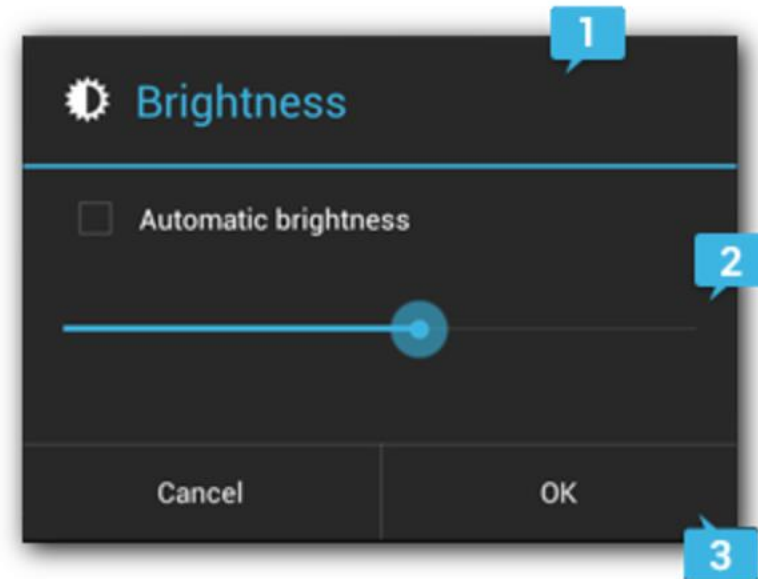
TimePickerDialog

ProgressDialog



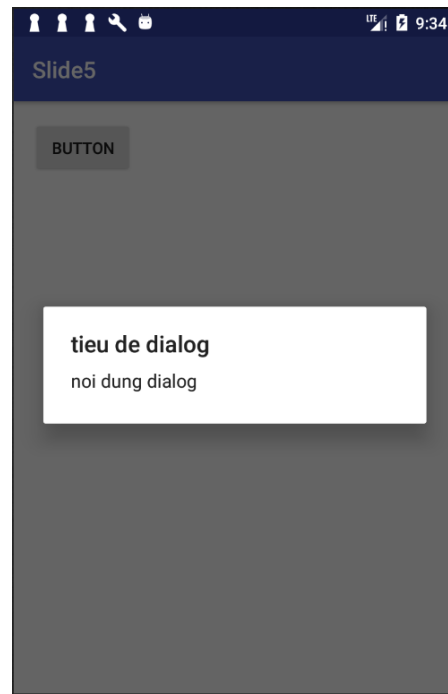


- ❑ Lớp AlertDialog cho phép ta tạo nhiều loại thiết kế hộp thoại.
- ❑ Có 3 vùng:
  1. Tiêu đề: không bắt buộc phải có
  2. Vùng nội dung: hiển thị 1 chuỗi thông báo, danh sách hoặc tự thiết kế nội dung
  3. Nút hành động: Chỉ có tối đa 3 nút trong 1 hộp thoại



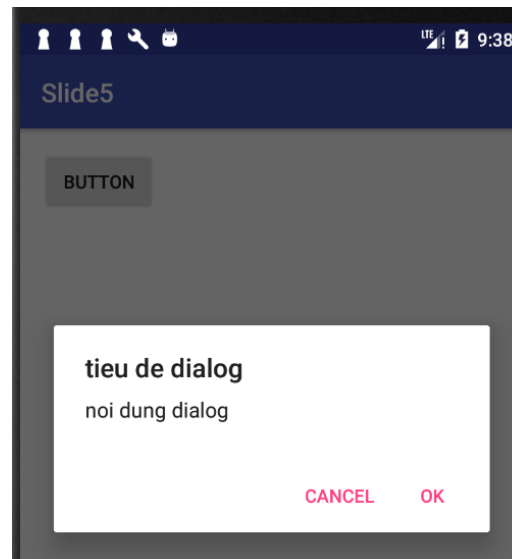
- ❑ Hộp thoại cảnh báo đưa ra 1 câu cảnh báo cho người

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(Main3Activity.this);  
builder.setMessage("noi dung dialog")  
    .setTitle("tieu de dialog");  
AlertDialog dialog = builder.create();  
dialog.show();
```



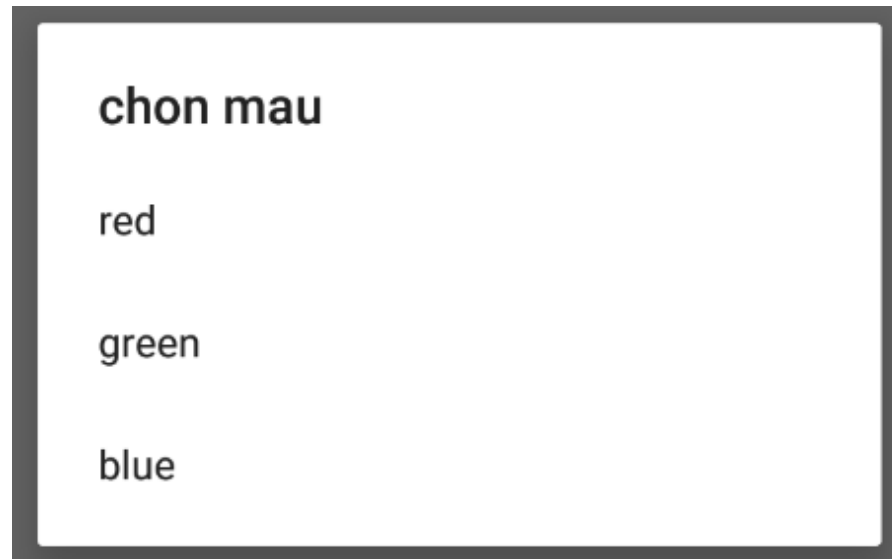
❑ Trong AlertDialog có thể tạo 1, 2 hoặc 3 button.

```
builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int id) {  
        // User clicked OK button  
    }  
});  
builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int id) {  
        // User cancelled the dialog  
    }  
});
```



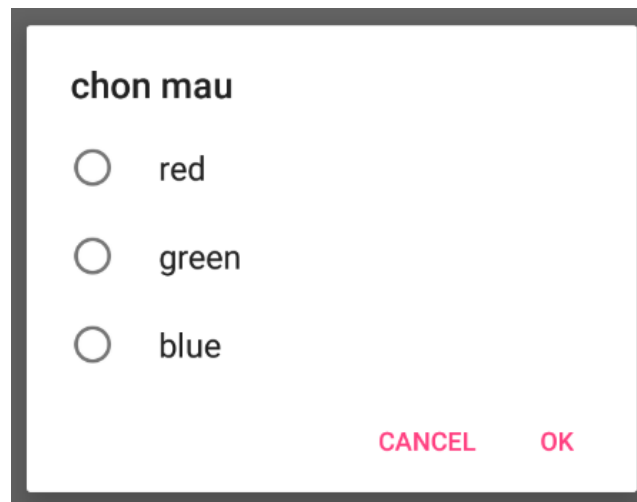
## ❑ AlertDialog dạng List

```
builder.setTitle("chon mau");  
builder.setItems(R.array.color, new DialogInterface.OnClickListener()  
    @Override  
    public void onClick(DialogInterface dialogInterface, int i) {  
        //i la index trong array  
    }  
});
```



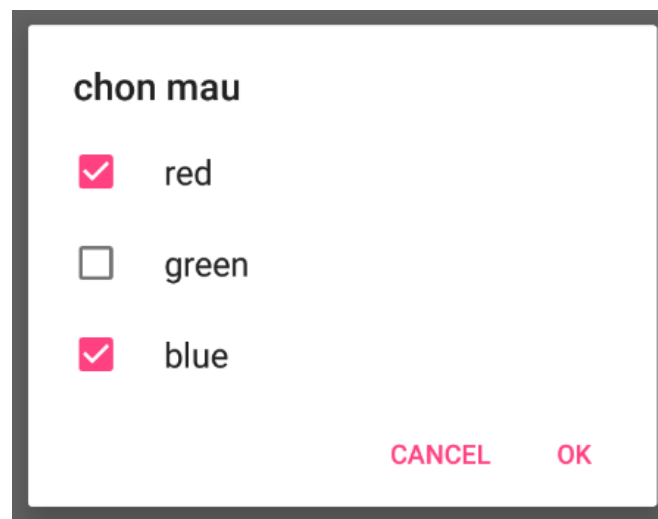
## ❑ AlertDialog dạng SingleChoice (radiobutton)

```
builder.setTitle("chon mau");  
builder.setSingleChoiceItems(R.array.color, R.array.colorcheck,  
    new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialogInterface, int i) {  
            }  
    }  
);
```

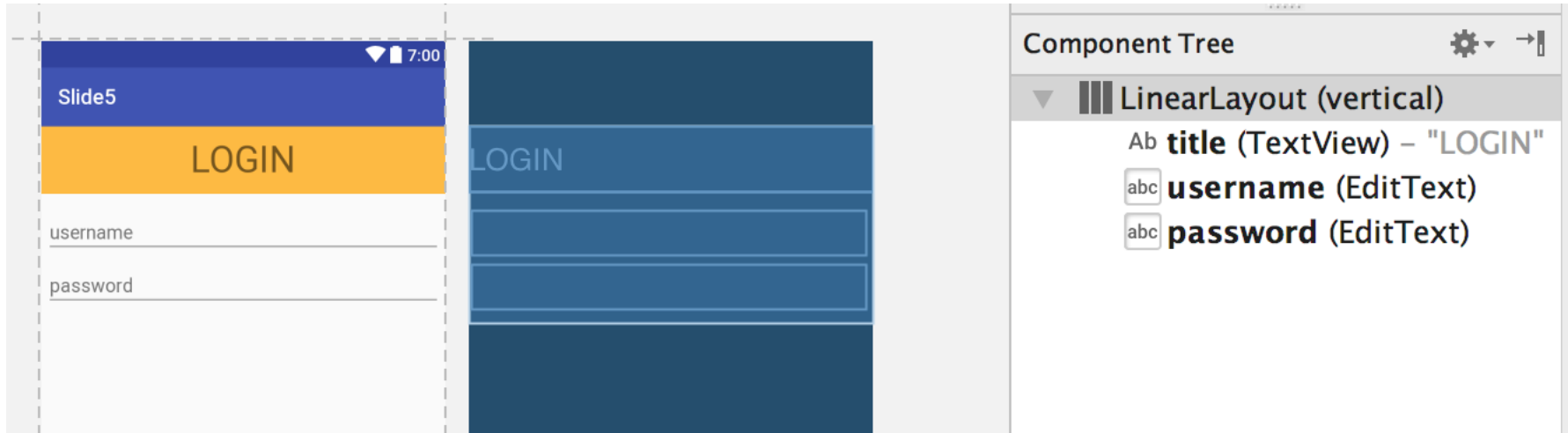


## ❑ AlertDialog dạng MultiChoice (checkbox)

```
builder.setTitle("chon mau");  
builder.setMultiChoiceItems(R.array.color,null,  
    new DialogInterface.OnMultiChoiceClickListener() {  
        @Override  
        public void onClick(DialogInterface dialogInterface, int i, boolean b) {  
        }  
    });
```



## □ Tạo 1 layout

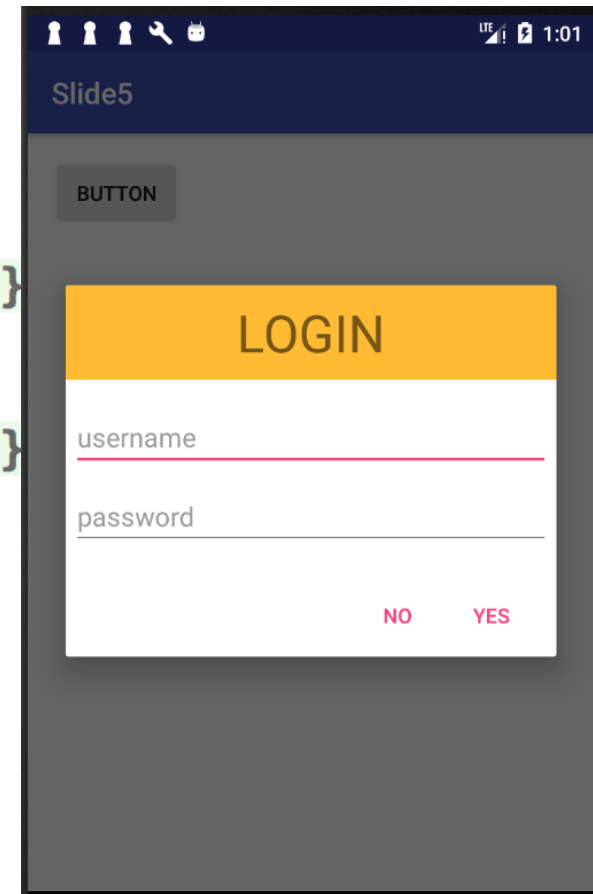


## □ Tạo dialog, gắn layout và gán view

```
AlertDialog.Builder alertDialog = new AlertDialog.Builder(  
    Main3Activity.this);  
LayoutInflater inflater = Main3Activity.this.getLayoutInflater();  
View v=inflater.inflate(R.layout.dialog_layout,null);  
alertDialog.setView(v);
```

## □ Gán thêm nút và hiển thị dialog

```
//nut yes duoc nhan  
AlertDialog.setPositiveButton("Yes",  
    (dialog, which) → { dialog.cancel(); }  
//nut no duoc nhan  
AlertDialog.setNegativeButton("No",  
    (dialog, which) → { dialog.cancel(); }  
  
AlertDialog.show();
```







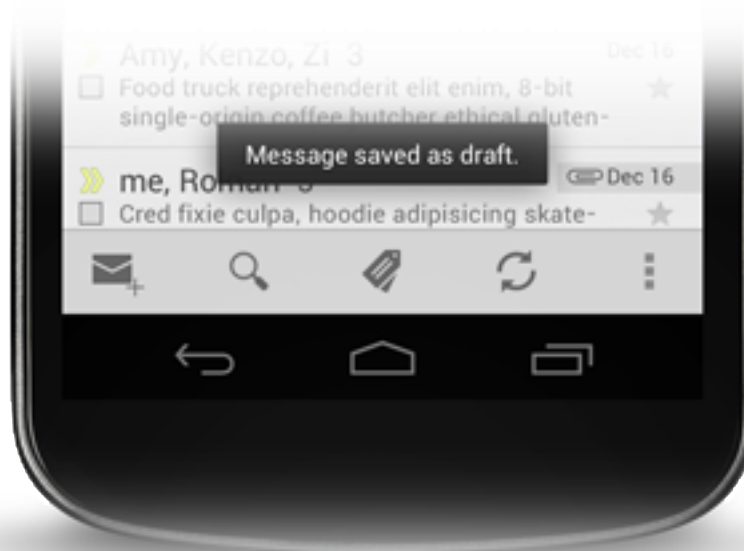
## **BÀI 5: PICKER – DIALOG - TOAST**

---

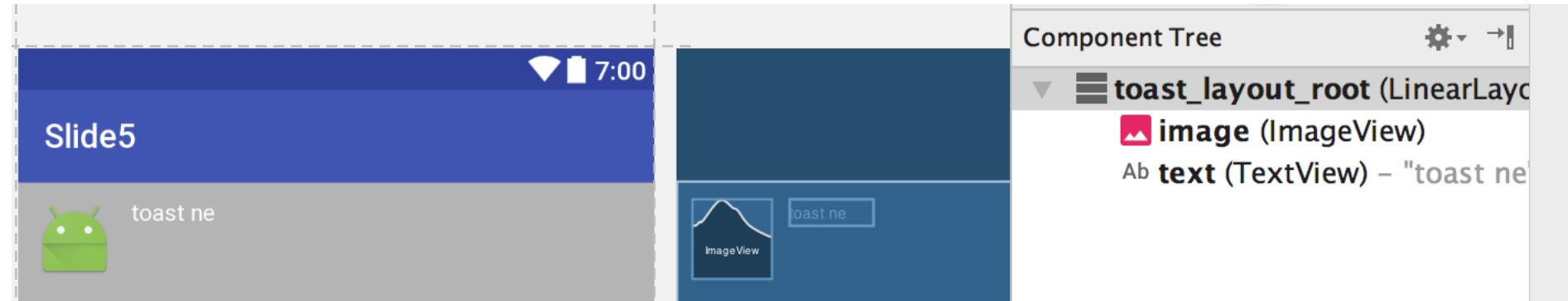
### **PHẦN 3: TOAST**

- ❑ Một thông báo nhỏ hiển thị trong 1 thời gian ngắn và tự động mất đi, chỉ có thể custom lại giao diện nhưng không thể tương tác với toast được.
- ❑ Toast đơn giản

```
Toast.makeText(Main3Activity.this, "tủi la toast", Toast.LENGTH_SHORT).show();
```



## ❑ Tạo 1 layout mới cho toast

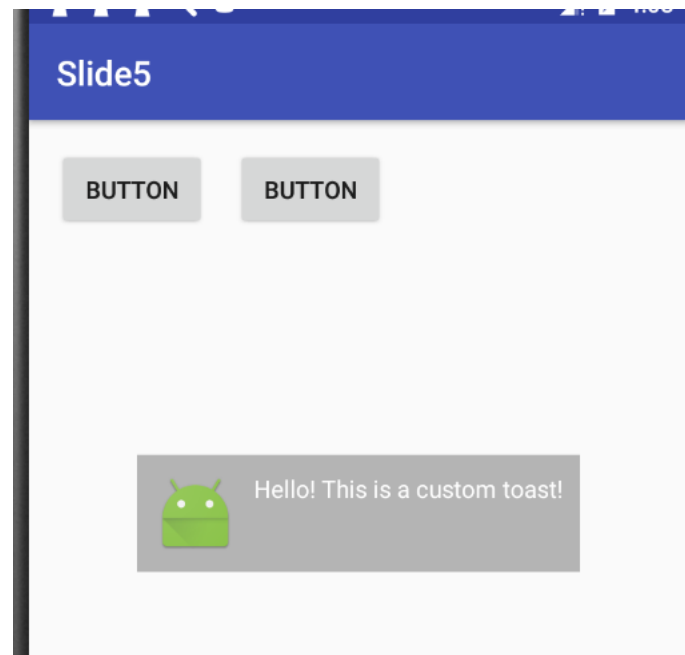


## ❑ Tạo view, gắn layout, ánh xạ các view và set giá trị

```
LayoutInflater inflater = getLayoutInflater();  
View layout = inflater.inflate(R.layout.toast_layout,  
    (ViewGroup) findViewById(R.id.toast_layout_root));  
ImageView image = (ImageView) layout.findViewById(R.id.image);  
image.setImageResource(R.mipmap.ic_launcher);  
TextView text = (TextView) layout.findViewById(R.id.text);  
text.setText("Hello! This is a custom toast!");
```

- ❑ Tạo toast, gắn view, set thuộc tính và hiển thị toast

```
Toast toast = new Toast(getApplicationContext());  
toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);  
toast.setDuration	Toast.LENGTH_LONG);  
toast.setView(layout);  
toast.show();
```





# DEMO

Các loại AlertDialog  
AlertDialog tùy chỉnh  
Toast, Toast tùy chỉnh



- ☐ Picker
- ☐ Dialog
- ☐ Toast





**Cảm ơn**