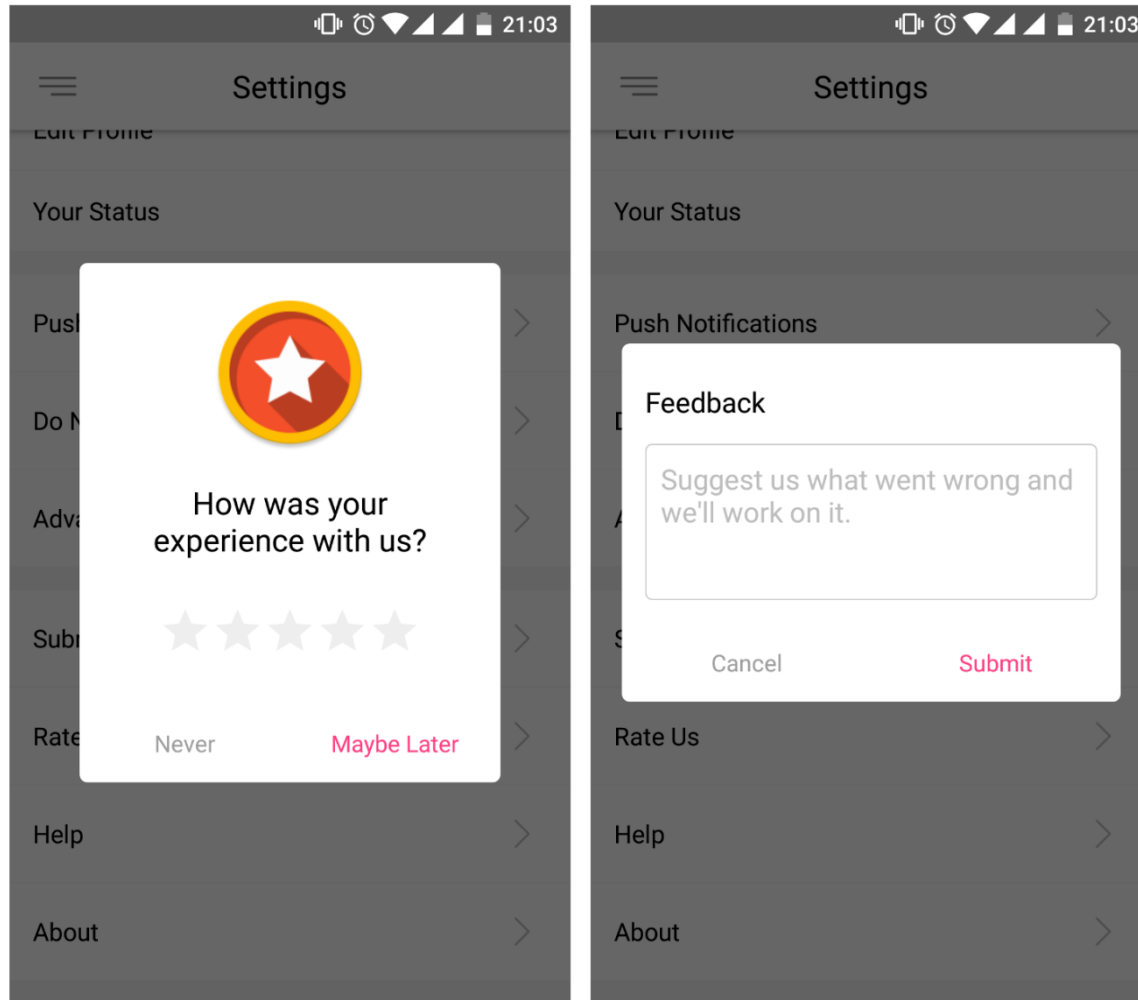


LẬP TRÌNH CHO CÁC THIẾT BỊ DI ĐỘNG

Bài 8: DIALOG & MENU

Giới thiệu Dialog trong Android



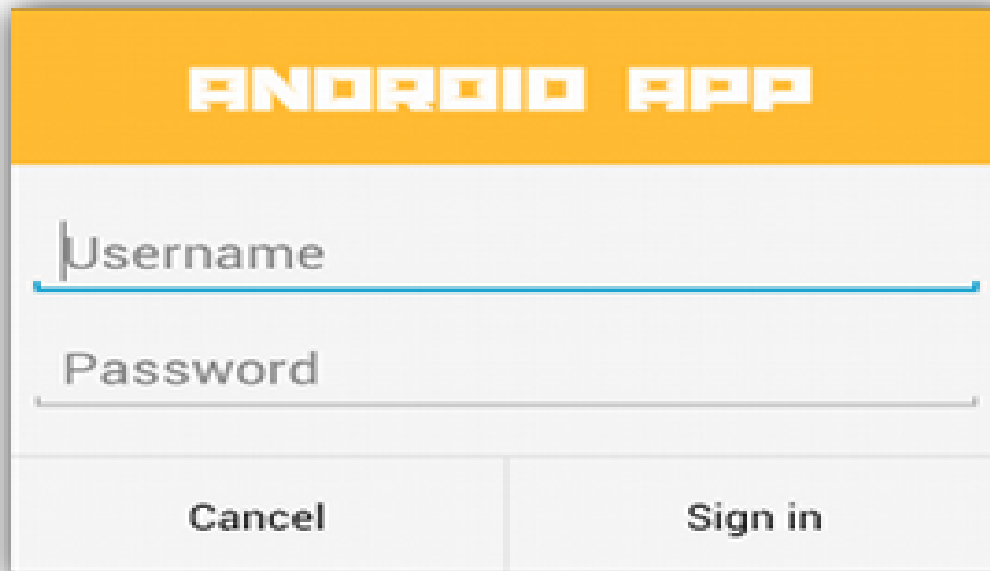
Giới thiệu Dialog trong Android

- ▶ Dialog trong Android là một cửa sổ nhỏ dùng để nhắc người dùng đưa ra quyết định hoặc nhập thông tin.
- ▶ Trong ứng dụng, nếu muốn hỏi người dùng về quyết định giữa có hoặc không để phản hồi bất kỳ hành động cụ thể nào do người dùng thực hiện, bằng cách giữ nguyên hoạt động và không thay đổi màn hình, bạn có thể sử dụng hộp thoại cảnh báo (Alert Dialog).



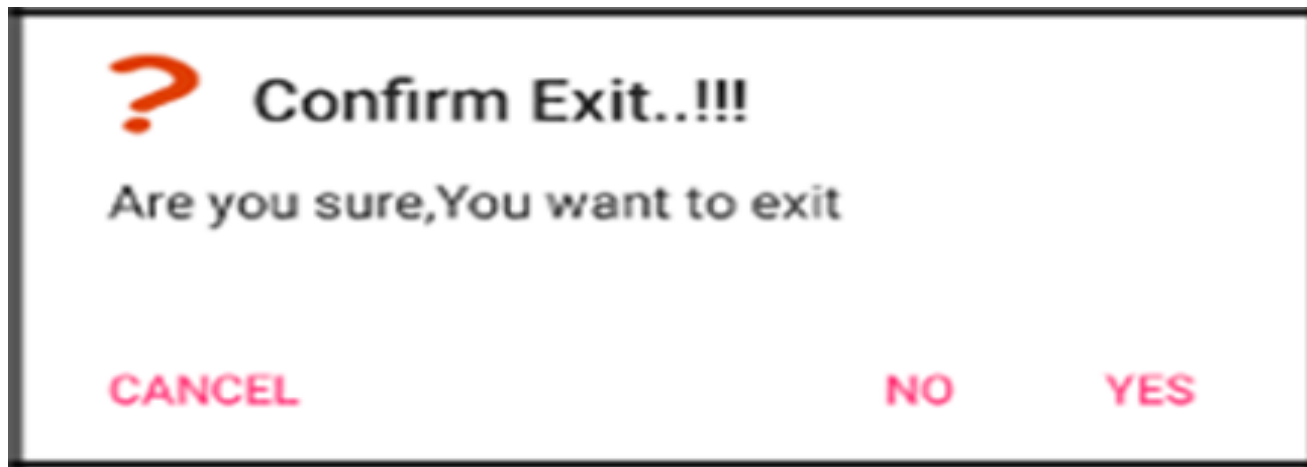
Giới thiệu Dialog trong Android

- ▶ Có nhiều loại Dialog như AlertDialog hay AlertDialog
- ▶ Có thể tùy chỉnh giao diện của Dialog sao cho phù hợp với ứng dụng của mình.



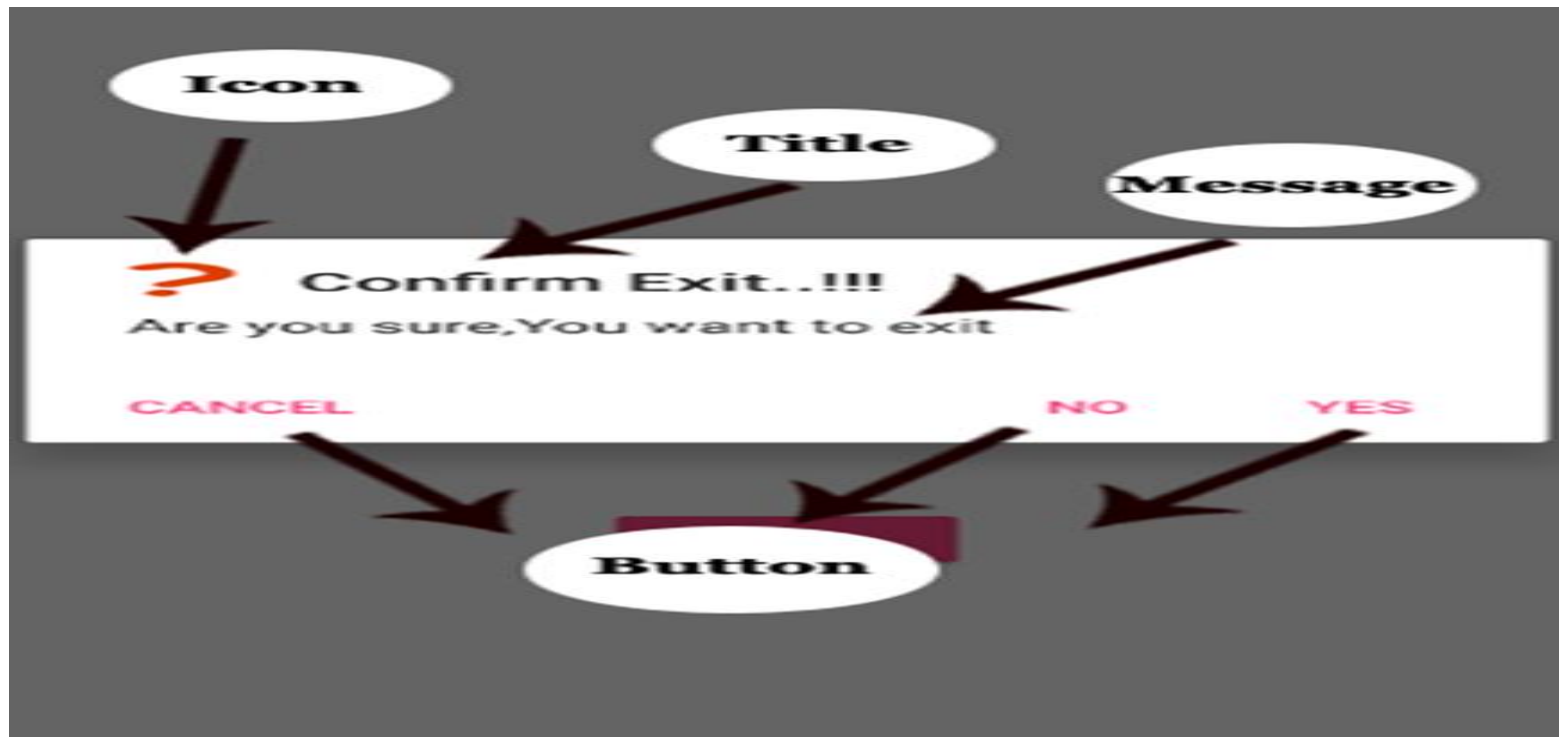
AlertDialog

- ▶ Dialog có thể coi là một thông báo mà người dùng có thể tương tác trực tiếp được.
- ▶ Ví dụ khi các bạn muốn xóa một tập tin quan trọng hay muốn thoát một chương trình nào đấy thì việc hiển thị một thông báo để người dùng chắc chắn về hành vi của mình là rất quan trọng.



AlertDialog

- ▶ Android AlertDialog gồm có 3 vùng: Tiêu đề (title), nội dung(content) và các button hành động.
- ▶ Android AlertDialog là lớp con của dialog



Tạo Alert Dialog

- ▶ AlertDialog.Builder được sử dụng để tạo ra một giao diện là một hộp thoại cảnh báo trong Android. Thường thì chúng ta phải thiết lập các vùng: Tiêu đề, nội dung, image, button và xử lý sự kiện cho button.

```
AlertDialog.Builder alertDialogBuilder = new  
    AlertDialog.Builder(this);
```



Các phương thức thường dùng trong Alert Dialog

- ▶ 1. setTitle(CharSequence title) – Phương thức này thường dùng để thiết lập tiêu đề cho Alert Dialog. Phương thức này tùy chọn.

//Thiết lập tiêu đề cho alert dialog

```
alertDialogBuilder.setTitle("Xác nhận để thoát..!!!");
```

- ▶ 2. setIcon(Drawable icon) – Phương thức này dùng để thêm icon trước tiêu đề. Icon phải được lưu trong thư mục drawable.

// Icon Of Alert Dialog

```
alertDialogBuilder.setIcon(R.drawable.question);
```



Các phương thức thường dùng trong Alert Dialog

- ▶ 3. `setMessage(CharSequence message)` – Phương thức này dùng hiển thị nội dung trong Alert Dialog. Nội dung này bắt buộc phải có.

`// Setting Alert Dialog Message`

`AlertDialogBuilder.setMessage("Bạn có chắc thoát?");`

- ▶ 4. `setCancelable(boolean cancelable)` – Phương thức này có 2 giá trị true/false. Nếu thiết lập false thì khi show dialog lên người dùng click ra bên ngoài dialog thì nó vẫn không bị mất, nếu set true thì sẽ mất khi click vào bất kì đâu ngoài dialog.

`AlertDialogBuilder.setCancelable(false);`



Các phương thức thường dùng trong Alert Dialog

- ▶ 5. `setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)` – Phương thức này thiết lập một Button Positive (dương), mà dương thì nó nằm bên phải.

```
AlertDialogBuilder.setPositiveButton("yes", new  
    DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface arg0, int arg1) {  
            // Đóng activity hiện tại  
            finish();  
        }  
    });
```



Các phương thức thường dùng trong Alert Dialog

- ▶ 6. `setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener)` – Phương thức này thiết lập một Button Negative (âm), mà âm thì nó nằm bên trái.

```
AlertDialogBuilder.setNegativeButton("No", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int
which) {

    Toast.makeText(MainActivity.this, "You clicked over
No", Toast.LENGTH_SHORT).show();

    }

});
```



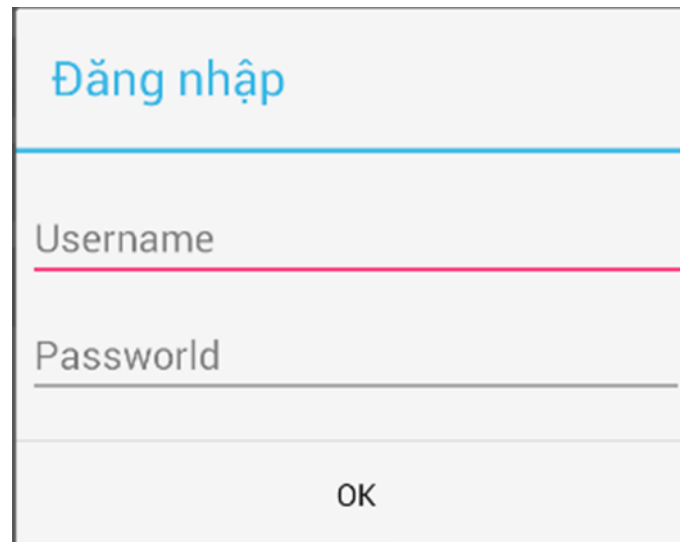
Các phương thức thường dùng trong Alert Dialog

- ▶ 7. `setNeutralButton(CharSequence text, DialogInterface.OnClickListener listener)` – Phương thức này đơn giản là thêm một Button mới, với button này chúng ta có thể xử lý sự kiện cho nó. Ví dụ 2 phương thức trên bạn đã thiết lập 2 Button "Yes" và "No" trong phương thức này chúng ta thiết lập Button "Cancel".



Custom (Tùy chỉnh) Alert Dialog

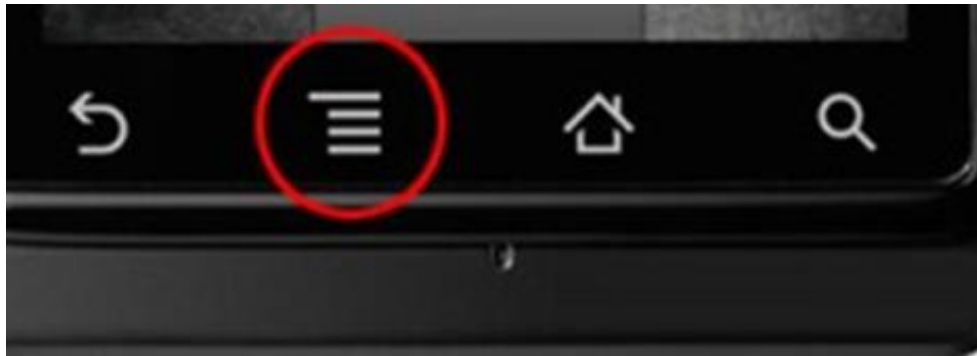
- ▶ Đôi khi, chúng ta muốn hiển thị hộp thoại cho người sử dụng cho EditText, Button. Thì chúng ta phải đi định nghĩa lại lớp AlertDialog



The image shows a custom AlertDialog dialog box with a light gray background and a thin black border. At the top, the text "Đăng nhập" (Login) is displayed in a blue font. Below this, there are two input fields: "Username" and "Passworld" (sic). The "Username" field has a pink underline, and the "Passworld" field has a gray underline. At the bottom of the dialog, there is a single button labeled "OK".

OptionsMenu

- ▶ Một ứng dụng có thể mở ra 1 menu để hiển thị các hành động thông qua 1 popup. Các menu chọn lựa chỉ tồn tại nếu thiết bị có hỗ trợ phím “option”. Nhưng mặc dù có sẵn phím này, Android 4.0 trở lên khuyến khích người lập trình sử dụng ActionBar thay cho menu.



Các phương thức thường dùng trong OptionMenu

- ▶ 1. onCreateOptionsMenu(): Phương thức onCreateOptionsMenu() sẽ thực hiện các công việc khởi tạo menu cho đối tượng Activity, ở đây chúng ta dùng phương thức inflate() của lớp android.view.MenuInflater để lấy dữ liệu của menu từ file options_menu.xml về sử dụng. Phương thức này nhận vào một đối tượng android.view.Menu.
- ▶ 2. onOptionsItemSelected(): Phương thức onOptionsItemSelected() sẽ xử lý sự kiện click menu. Phương thức này nhận vào một đối tượng android.view.MenuItem.



ContextMenu

- ▶ ContextMenu (menu cho ngữ cảnh) ngữ cảnh hiển thị nổi lên trên màn hình nhưng được gắn với một đối tượng View nào đó chứ không liên quan gì tới Activity, menu ngữ cảnh sẽ hiện ra khi chúng ta click và giữ tay trên View đó trong một khoảng thời gian.



ContextMenu



Các phương thức thường dùng trong ContextMenu

- ▶ 1. `registerForContextMenu(view)`: Phương thức này được sử dụng để gắn một `ContextMenu` cho một `View`
- ▶ 2. `onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo)`: Phương thức này sẽ thực hiện các công việc khởi tạo menu cho đối tượng `Activity`, ở đây chúng ta dùng phương thức `inflate()` của lớp `android.view.MenuInflater` để lấy dữ liệu của menu từ file `options_menu.xml` về sử dụng. Phương thức này nhận vào một đối tượng `android.view.Menu`.
- ▶ 3. `onContextItemSelected(MenuItem item)`: Phương thức `onContextItemSelected()` sẽ làm nhiệm vụ xử lý sự kiện click trên từng item.

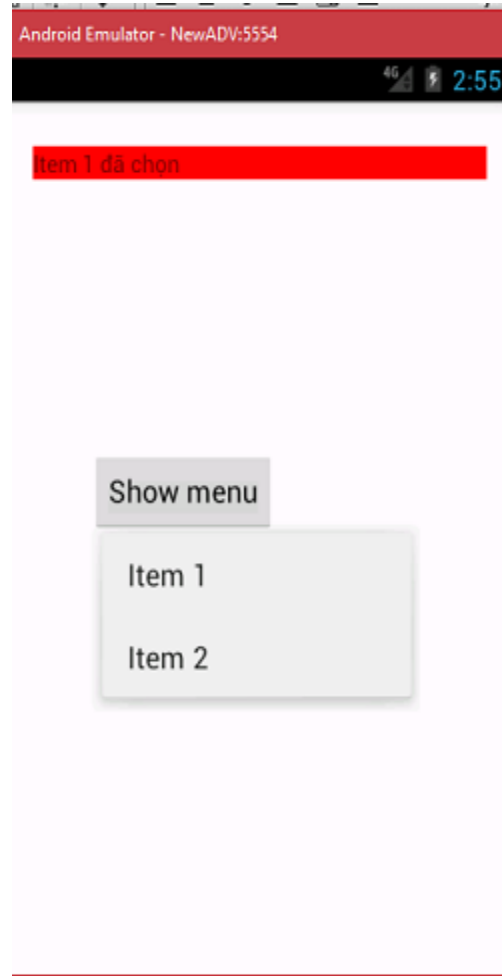


Popup menu trong Android

- ▶ Android Popup Menu hiển thị menu bên dưới một view mà nó được neo nếu không gian có sẵn nếu không ở trên các view mà nó được neo. Nó biến mất nếu bạn nhấp vào bên ngoài popup menu. Các `android.widget.PopupMenu` là lớp con trực tiếp của lớp `java.lang.Object`.



Popup menu



Phương thức thường sử dụng trong PopupMenu

- ▶ 1. `setOnMenuItemClickListener`: Phương thức này được sử dụng để gắn listener cho đối tượng `PopupMenu`, khác với menu ngữ cảnh và menu tùy chọn là 2 loại menu này đã có sẵn trong `Activity` và `View` nên chúng ta không cần gọi trực tiếp ra như đối tượng menu popup.
- ▶ 2. `show()`: Sau khi tạo `PopupMenu`, chúng ta phải gọi phương thức `show()` nếu muốn hiện menu này ra.
- ▶ 3. `onMenuItemClick()`: Phương thức `onOptionsItemSelected()` sẽ xử lý sự kiện click menu. Phương thức này nhận vào một đối tượng `android.view.MenuItem`.

