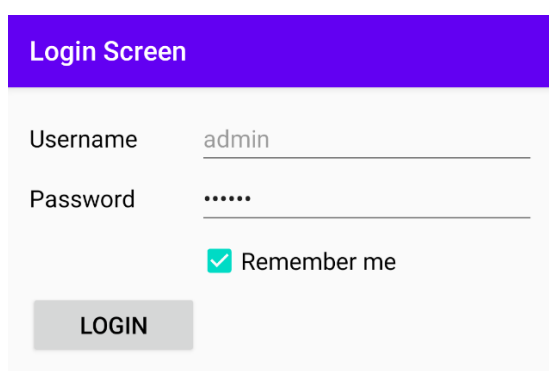


## PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG - 503074

### TUẦN 01

GIẢNG VIÊN BIÊN SOẠN: MAI VĂN MẠNH

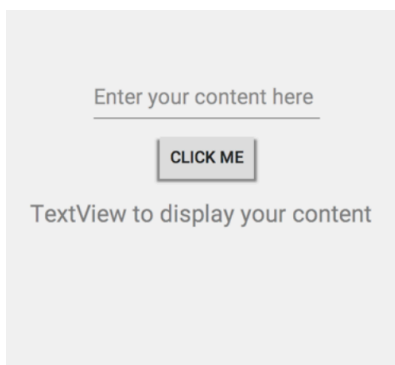
**Bài 1.** Thiết kế ứng dụng như hình bên dưới, sử dụng LinearLayout.



The image shows a login screen with a purple header labeled "Login Screen". Below the header, there are two input fields: "Username" with the text "admin" and "Password" with masked characters ".....". Below the password field, there is a checkbox labeled "Remember me" which is checked. At the bottom, there is a button labeled "LOGIN".

- Xuất ra thông báo (dạng **toast**) cụ thể cho từng trường hợp khi người dùng để trống username hoặc password. Nếu cả hai ô đều đang để trống thì chỉ hiển thị thông báo lỗi cho phần username. Focus con trỏ vào textbox tương ứng.
- **Ngay khi** người dùng tick chọn vào “**Remember me**” thì hiển thị thông báo “**Tài khoản của bạn sẽ được ghi nhớ**”. Ngay khi người dùng hủy chọn thì hiển thị thông báo “**Bạn sẽ cần phải đăng nhập trong các lần tiếp theo**”.
- Nếu người dùng nhập đúng username (**admin**) và password (**123456**) thì kết thúc activity hiện tại và mở một activity mới (trong bài tập 2); ngược lại, hiển thị thông báo “**Tên đăng nhập hoặc mật khẩu không chính xác**” và focus vào ô password.

**Bài 2.** Tạo activity có giao diện như hình bên dưới.



The image shows a simple activity layout with a light gray background. At the top, there is a text input field with the placeholder text "Enter your content here". Below the input field, there is a button labeled "CLICK ME". At the bottom, there is a text label that says "TextView to display your content".

Khi click vào nút **CLICK ME**:

- Nếu EditText đang rỗng: hiển thị thông báo dạng Toast với nội dung “**vui lòng nhập thông tin**”.

- Ngược lại, nội dung trong EditText sẽ được hiển thị ở TextView bên dưới. Nội dung bên trong EditText sau đó được xóa đi.

#### Nâng cao:

- Nếu người dùng gõ chữ “off” thì button sẽ **tự động** bị disable (không click được).
- Nếu người dùng gõ chữ “on” thì button sẽ **tự động** được enabled (cho phép click).
- Lưu ý, trạng thái (enabled/disabled) của button sẽ được thay đổi **ngay lập tức** sau khi người dùng gõ chữ “on” hoặc “off”.

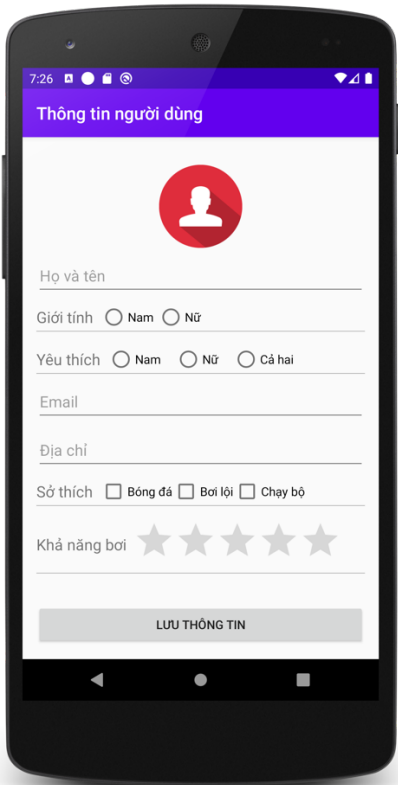
#### Gợi ý làm bài:

- Để viết sự kiện click cho button, gọi phương thức `setOnClickListener` trên đối tượng button đó, truyền vào một đối tượng thuộc kiểu `View.OnClickListener`.
- Để bắt sự kiện thay đổi text cho EditText, gọi phương thức `addTextChangedListener()` cho đối tượng EditText đó rồi thực hiện tương tự như đối với sự kiện click.
- Để lấy giá trị và gán giá trị cho TextView, EditText... chúng ta gọi hai phương thức `getText()`, `setText()`.
- Để hiện thông báo dạng Toast, sử dụng phương thức `Toast.Show()`.
- Để disable/enable một View (Button, EditText, TextView,...), gọi phương thức `setEnabled(true/false)`;

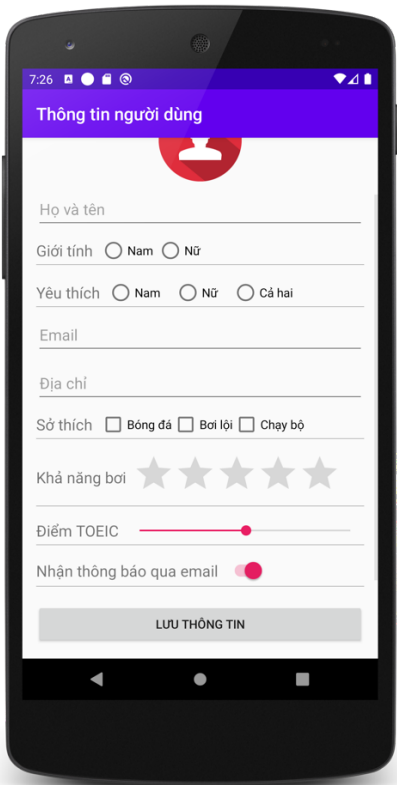
#### Bài 3. Tạo ứng dụng có giao diện như các hình bên dưới.

- Ứng dụng cung cấp hai giao diện: giao diện nhập thông tin (hình 1, hình 2) và giao diện hiển thị thông tin sau khi đã nhập (hình 3, hình 4).
- Cả hai activity đều sử dụng **scroll view** cho phép scroll theo chiều dọc trong trường hợp màn hình không đủ không gian để hiển thị các thành phần giao diện.
- Hai nút “**Lưu thông tin**” và “**Đóng ứng dụng**” luôn nằm ở cuối màn hình, kể cả khi ứng dụng chạy trên thiết bị có màn hình đủ rộng để hiển thị toàn bộ giao diện.
- Ô “**Họ và tên**” tự động viết hoa các ký tự đầu tiên của mỗi từ trong tên.
- Khi nhập email, ký tự `@` tự động xuất hiện trong bàn phím.
- Rating bar (khả năng bơi) chỉ hiển thị đúng chính xác 5 ngôi sao, giá trị của rating bar luôn là một số nguyên (không thể có trường hợp chọn 4.5 sao).
- SeekBar chọn điểm TOEIC cho ra kết quả từ 5 – 990 điểm (bội số của 5).
- Khi nhấn “**Lưu thông tin**” nếu có thông tin nào không hợp lệ thì hiển thị thông báo lỗi tương ứng bằng Toast. Khi tất cả các thông tin đều hợp lệ thì đóng activity số 1 và hiển thị activity số 2 với các thông tin vừa nhập.

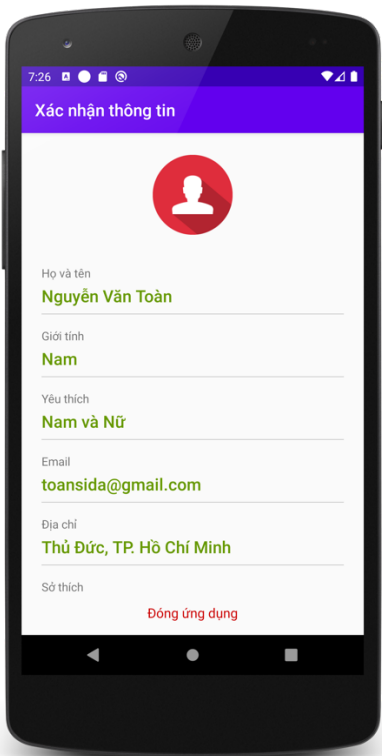
- Khi nhấn “Đóng ứng dụng” thì chương trình kết thúc.



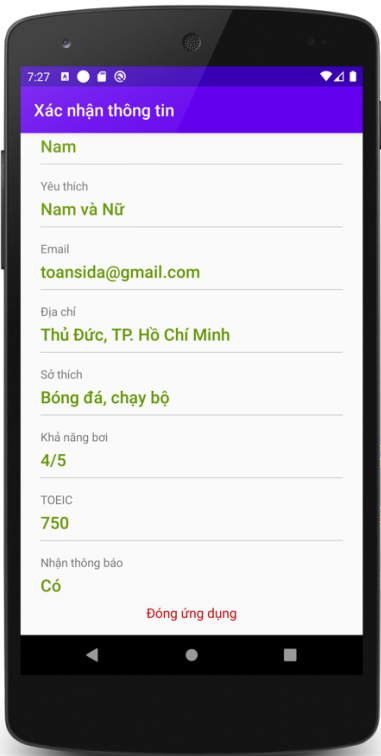
Hình 1



Hình 2



Hình 3





Hình 4

## Bài tập về nhà

**Bài 1.** Thiết kế lại giao diện bài tập 1 sử dụng **Relative Layout** và **Table Layout**.

**Bài 2.** Tạo ứng dụng như hình bên dưới.

<div> <b>Android Checkboxes</b></div> <div>List of Checkboxes for selection</div> <div><input checked="" type="checkbox"/> Android</div> <div><input checked="" type="checkbox"/> iOS</div> <div><input type="checkbox"/> Windows</div> <div><input type="checkbox"/> RIM</div> <div>Click here to see Results</div> <div>The following were selected...</div> <div>Android : true</div> <div>iOS : true</div> <div>Windows : false</div> <div>RIM : false</div>	<div> <b>Android RadioButtons</b></div> <div>List of Radio Buttons for selection</div> <div><input type="radio"/> Android</div> <div><input checked="" type="radio"/> iOS</div> <div><input type="radio"/> Windows</div> <div><input type="radio"/> RIM</div> <div>Click here to see Results</div>
---	---

- Khi nhấn nút **Click here to see Results**, TextView ở bên dưới sẽ hiển thị nội dung phản ánh các thông tin mà người dùng đã chọn.

### Gợi ý làm bài:

- Để biết một Checkbox/RadioButton có được chọn hay không, sử dụng phương thức **isChecked()** của đối tượng đó.
- Về nguyên tắc, chỉ một RadioButton được chọn tại một thời điểm. Để làm được điều đó, các RadioButton phải được đặt trong cùng một nhóm. Sử dụng thẻ **<RadioGroup>**. Xem ví dụ bên dưới.

```
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <RadioButton
        android:text="RadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="42dp"
        android:id="@+id/radioButton" />

    <RadioButton
        android:text="RadioButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/radioButton2" />

</RadioGroup>
```