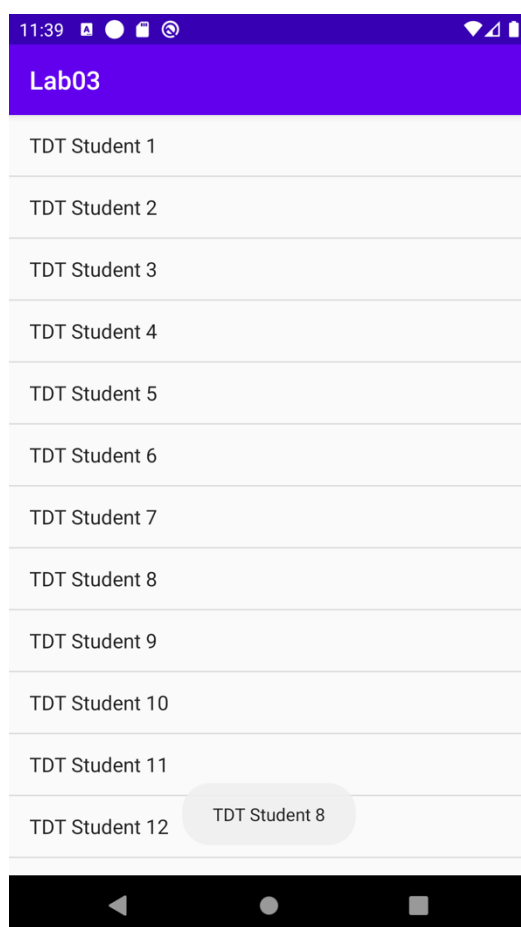


## PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG - 503074

### TUẦN 03

GIẢNG VIÊN BIÊN SOẠN: MAI VĂN MẠNH

**Bài 1.** Tạo ứng dụng sử dụng [Listview](#) để hiển thị danh sách sinh viên như hình bên dưới.



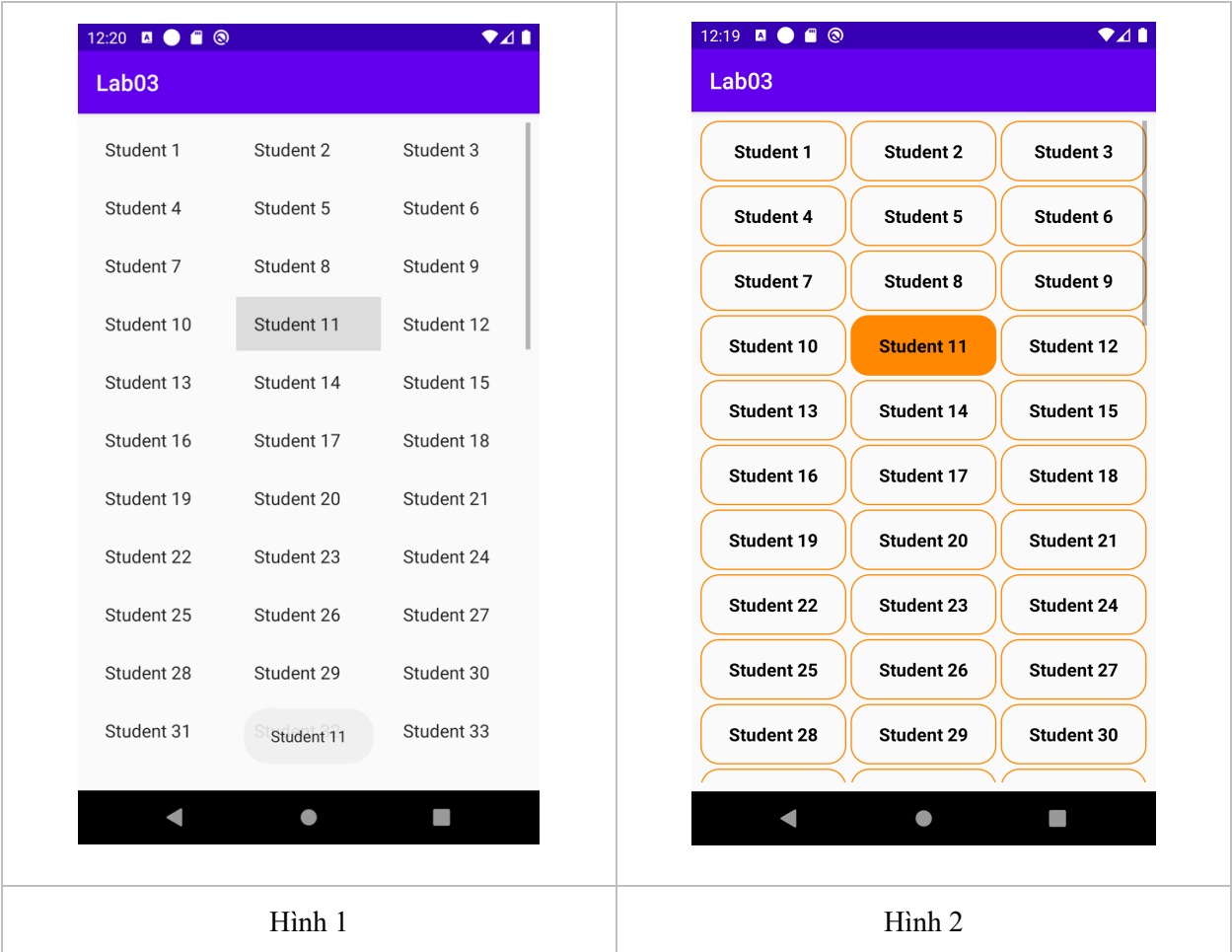
- Danh sách có tối thiểu 20 sinh viên.
- Khi click vào một dòng bất kỳ, hiển thị thông báo toast với nội dung tương ứng.
- **Yêu cầu nâng cao:** đổi màu chữ thành **orange** và **in đậm**.

#### Hướng dẫn:

- Sử dụng [ArrayAdapter](#) với tập tin layout được cung cấp sẵn trong hệ thống có tên là `android.R.layout.simple_list_item_1` để tạo giao diện cơ bản cho 1 dòng.
- Để đổi màu, font chữ ta chỉ cần sao chép nội dung của tập tin `simple_list_item_1` và lưu thành tập tin mới vào thư mục `res/layout` sau đó thay đổi các thuộc tính cần thiết theo yêu cầu của đề bài. Sau đó điều chỉnh lại source code để [ArrayAdapter](#) sử dụng tập tin layout mới vừa tạo làm tham số.

**Bài 2.** Tạo ứng dụng sử dụng **GridView** để hiển thị danh sách sinh viên như **hình 1** bên dưới.

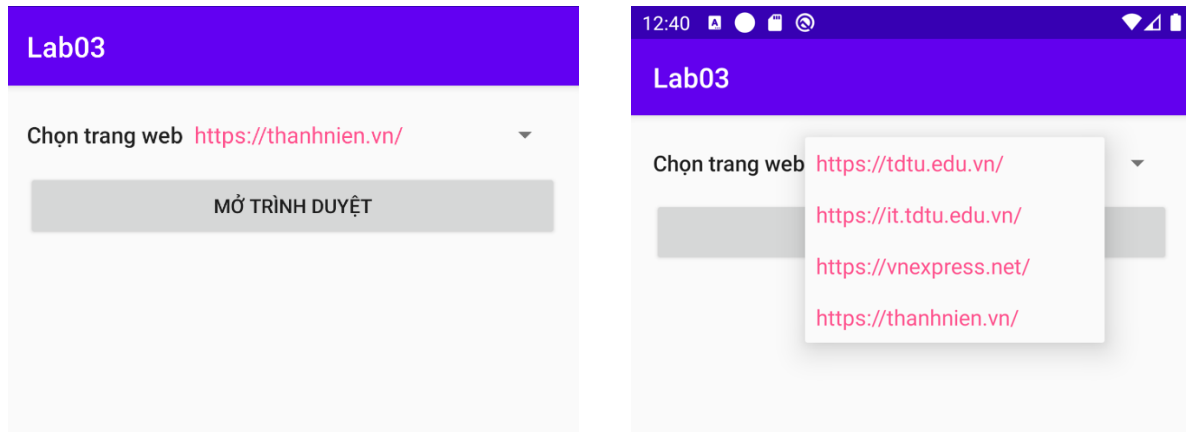
- Danh sách có tối thiểu 100 sinh viên.
- Gridview hiển thị dữ liệu trên 3 cột.
- Khi click vào một dòng bất kỳ, hiển thị thông báo toast với nội dung tương ứng.
- **Yêu cầu nâng cao (hình 2):**
  - o Tùy biến giao diện cho mỗi item bên trong Gridview: mặc định mỗi item chỉ có viền m, khi click vào thì màu nền sẽ thay đổi (StateListDrawable).
  - o Thiết lập khoảng trống giữa các dòng và các cột của GridView.



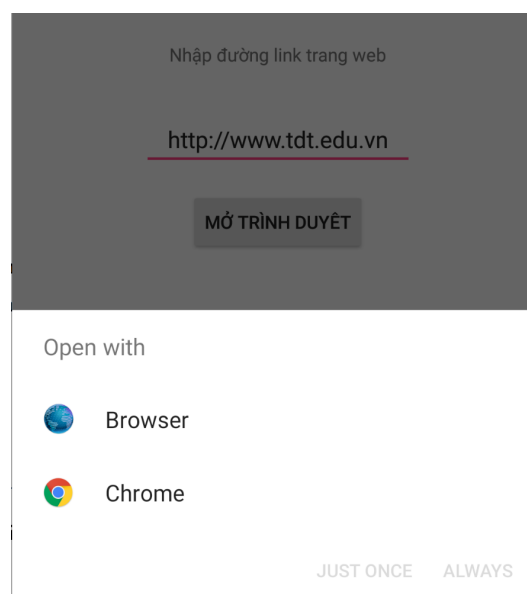
**Hướng dẫn làm bài:**

- Cách tạo Gridview trong hình 1 hoàn toàn tương tự với bài tập 1, chỉ cần thay thế ListView bằng GridView.
- Tùy biến giao diện cho từng item trong Gridview: tạo một tập tin có nội dung giống với `simple_list_item_1` và lưu nó vào `res/layout`. Sau đó điều chỉnh các thuộc tính của TextView cho phù hợp với yêu cầu của đề bài.
- Để tạo background có hình chữ nhật với các góc tròn chúng ta sử dụng `ShapeDrawable`. Khi kết hợp với `StateListDrawable` chúng ta sẽ thiết lập được các background khác nhau cho từng trạng thái clicked/unclicked.

**Bài 3.** Thiết kế ứng dụng như hình bên dưới. Ứng dụng sử dụng **Spinner** để cung cấp các lựa chọn url khác nhau.

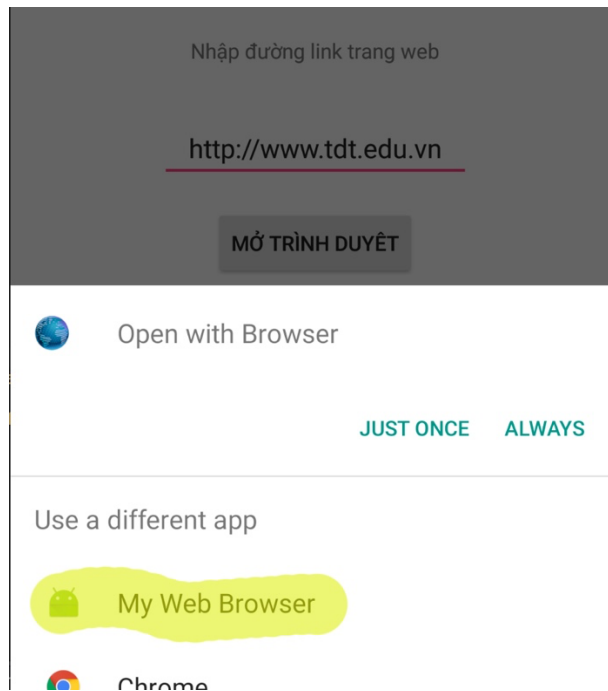


- Khi nhấn nút **MỞ TRÌNH DUYỆT**, app của chúng ta sẽ gọi một trình duyệt web (app khác) để hiển thị trang web được chọn.
- Nếu trong máy có cài nhiều hơn 1 trình duyệt web thì giao diện dưới đây sẽ được hiển thị (phần này hệ điều hành thực hiện tự động).



**Bài 4.** Tiếp tục thực hiện bài tập 3, bổ sung thêm một activity có chức năng tương tự một web browser để hiển thị các trang web.

- Activity mới này (tạm gọi là MyWebBrowser) sử dụng WebView để hiển thị nội dung của các url.
- MyWebBrowser sẽ hoạt động như các trình duyệt web khác, nó nhận vào tham số là một url (thông qua Intent) và hiển thị nội dung của url nhận được.
- Nếu thực hiện đúng, khi mở một trang web bất kỳ từ giao diện của bài 3, ứng dụng của chúng ta sẽ được liệt kê trong danh sách các Web Browser để người dùng lựa chọn.



### Hướng dẫn làm bài:

- Để gọi một web browser và hiển thị nội dung url:
  - Tạo một đối tượng **Intent** với action là **ACTION\_VIEW**.
  - Gọi phương thức **setData()** trên Intent vừa tạo với tham số là **url** cần load.
- Để tự tạo một web browser:
  - Tạo Activity có chứa một **WebView** để hiển thị nội dung trang web
  - Đăng ký **IntentFilter** cho activity này để hệ thống có thể gọi nó ra nhằm xử lý các trang url.
    - *Action: View*
    - *Category: Default và Browsables*
    - *Data schema: http và https*
      - `android:host="*"`
      - `android:pathPattern=".*"`
  - Khai báo quyền sử dụng **Internet**, vì lúc này chính app của chúng ta thực hiện việc load nội dung của một trang web.