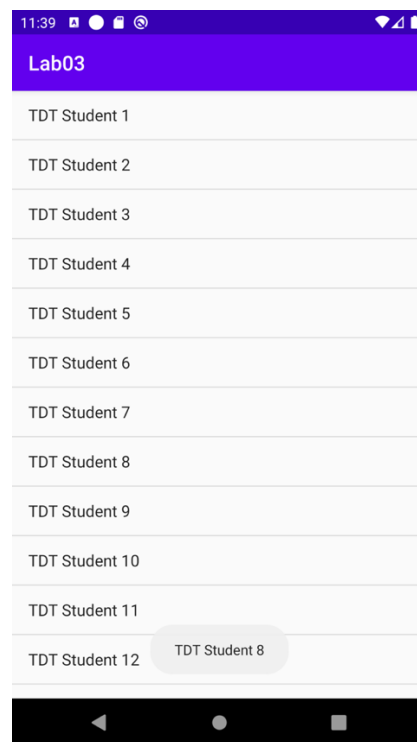
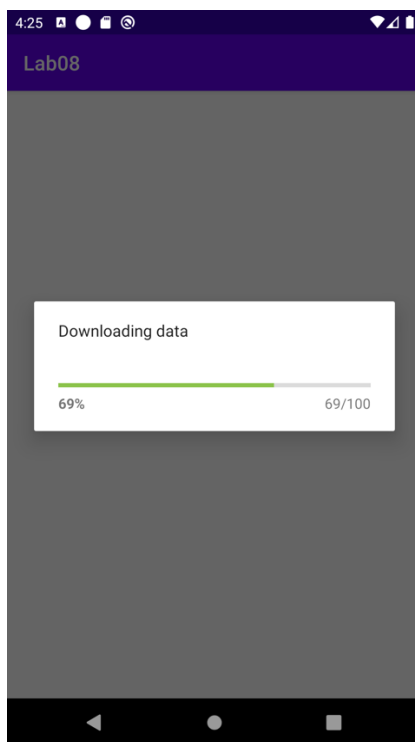


PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG - 503074

TUẦN 08

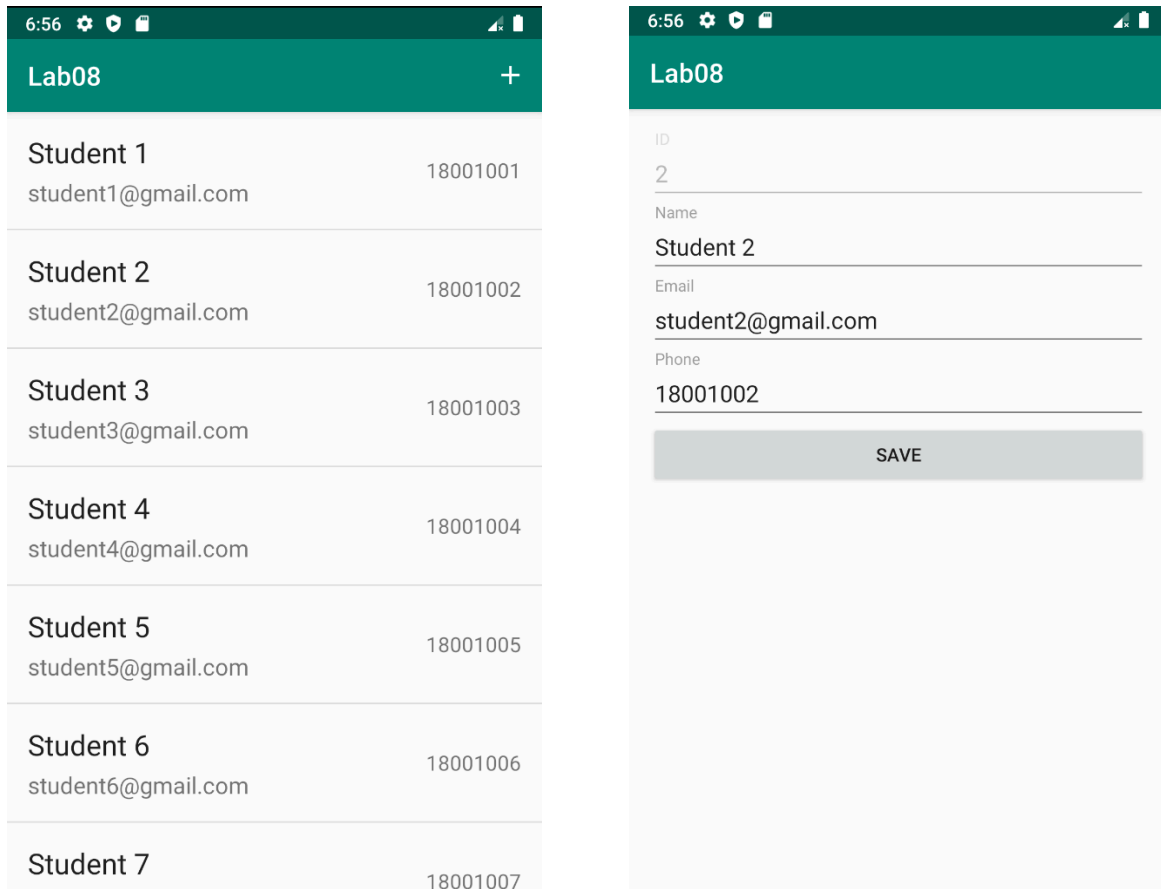
GIẢNG VIÊN BIÊN SOẠN: MAI VĂN MẠNH

Bài 1. Viết ứng dụng hiển thị danh sách sinh viên như hình bên dưới.



- Bài tập này yêu cầu sử dụng `AsyncTask` để giải quyết vấn đề.
- Khi vừa được mở lên, ứng dụng cần phải nạp danh sách sinh viên từ một nguồn cụ thể. Trong ví dụ này chúng ta sẽ không thực hiện download hay đọc file mà sẽ viết một phương thức `loadData()` để mô phỏng quá trình tải dữ liệu cần nhiều thời gian bằng cách sử dụng phương thức `Thread.sleep()` trong vòng lặp.
- Khi quá trình đọc dữ liệu đang diễn ra, app cần hiển thị Progress Dialog để người dùng có thể theo dõi được tiến độ.
- Vì `Thread.sleep()` thường sẽ làm UI bị block nên quá trình nạp dữ liệu cần phải được thực hiện ở background, nói cách khác phương thức `loadData()` cần được gọi trong một thread khác main thread (`AsyncTask`).
- Sau khi hoàn thành bài tập: thử mở app lên, chờ cho đến khi danh sách sinh viên được nạp & hiển thị thành công rồi thực hiện xoay màn hình thiết bị → mỗi khi thiết bị xoay màn hình, quá trình nạp dữ liệu phải thực hiện lại từ đầu → Nghiên cứu giải pháp cho vấn đề này.

Bài 2. Thiết kế ứng dụng có giao diện như hình bên dưới.



The left screenshot shows a list of students with their IDs and emails. The right screenshot shows a form to edit a student's information, including ID, Name, Email, and Phone, with a 'SAVE' button.

| ID | Name | Email | Phone |
|----|-----------|--------------------|----------|
| 1 | Student 1 | student1@gmail.com | 18001001 |
| 2 | Student 2 | student2@gmail.com | 18001002 |
| 3 | Student 3 | student3@gmail.com | 18001003 |
| 4 | Student 4 | student4@gmail.com | 18001004 |
| 5 | Student 5 | student5@gmail.com | 18001005 |
| 6 | Student 6 | student6@gmail.com | 18001006 |
| 7 | Student 7 | student7@gmail.com | 18001007 |

The right screenshot shows a form to edit a student's information, including ID, Name, Email, and Phone, with a 'SAVE' button.

ID: 2

Name: Student 2

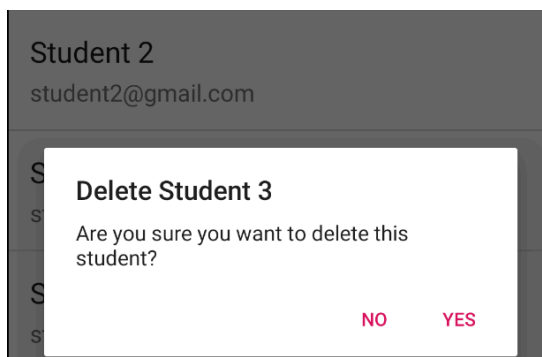
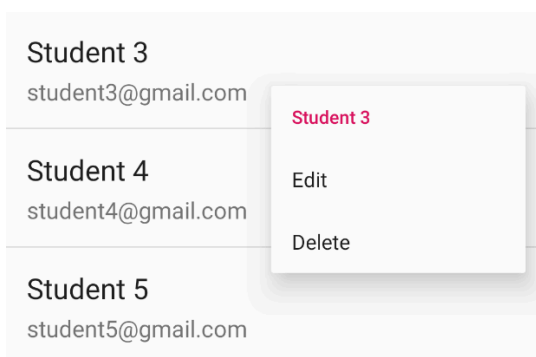
Email: student2@gmail.com

Phone: 18001002

SAVE

Yêu cầu về giao diện:

- Ứng dụng có hai màn hình: màn hình chính hiển thị danh sách sinh viên sử dụng [RecyclerView](#), và màn hình phụ dùng để thêm một sinh viên mới hoặc chỉnh sửa thông tin của một sinh viên đang có (sử dụng [TextInputLayout](#)).
- Khi nhấn click vào một sinh viên đang có, màn hình thứ 2 sẽ xuất hiện và cho phép điều chỉnh thông tin của sinh viên này.
- Khi nhấn và giữ vào một sinh viên thì context menu sẽ xuất hiện, cung cấp hai lựa chọn là [Edit](#) hoặc [Delete](#); nếu người dùng nhấn '[Edit](#)' thì app xử lý tương tự như trường hợp click trực tiếp vào sinh viên; nếu người dùng nhấn '[Delete](#)' thì app hiển thị dialog để xác nhận trước khi xóa sinh viên.
- Ngoài ra sinh viên cần viết thêm các chức năng kiểm tra tính hợp lệ của input và hiển thị thông báo phù hợp đến người dùng nếu có các thông tin không hợp lệ.



Yêu cầu về dữ liệu:

- Sinh viên sử dụng thư mục **api** được cung cấp sẵn để thiết lập phần server. Xem hướng dẫn cách thiết lập phía server bên dưới.
- Sinh viên cần sử dụng các thư viện như **Retrofit**, **Volley** hoặc **URLConnection** để thực hiện lời gọi API từ phía Android đến phía PHP Server nhằm thực hiện các thao tác CRUD (đọc, thêm, xóa, cập nhật).

Yêu cầu khác:

- Ngay khi mở ứng dụng, nếu không có kết nối internet, ứng dụng cần hiển thị thông báo cho người dùng biết. Lúc này nếu người dùng nhấn **+** để thêm một student thì app không cho phép thêm và hiển thị dialog để người dùng biết thiết bị chưa có kết nối internet.
- Ngay khi có kết nối internet (app lúc này vẫn đang được mở), ứng dụng cần tự động tải danh sách sinh viên về và hiển thị trên RecyclerView. Sử dụng broadcast receiver với action là **CONNECTIVITY_ACTION** để thực hiện tính năng này.
- **Nâng cao:** bổ sung thêm Room database để làm bộ nhớ cache, cho phép sử dụng ứng dụng ngay cả khi thiết bị không có kết nối internet.

Hướng dẫn thiết lập Server:

- Cài đặt công cụ XAMPP.
- Chép thư mục **api** được cung cấp sẵn vào thư mục **htdocs** của XAMPP.
- Mở tập tin **api/connection.php**, điều chỉnh lại thông số cho phù hợp.
- Vào PhpMyAdmin, tạo database tên '**students**' rồi import tập tin **api/students.sql** vào database vừa tạo.
- Truy cập **http://localhost/api/get-students.php** để kiểm tra kết quả (máy tính). **Trên máy ảo Android, đổi localhost thành 10.0.2.2.**

Bài 3. Thực hiện lại bài tập 1 với các giải pháp khác như:

- ViewModel + Live Data
- Retained Fragment
- AsyncTaskLoader