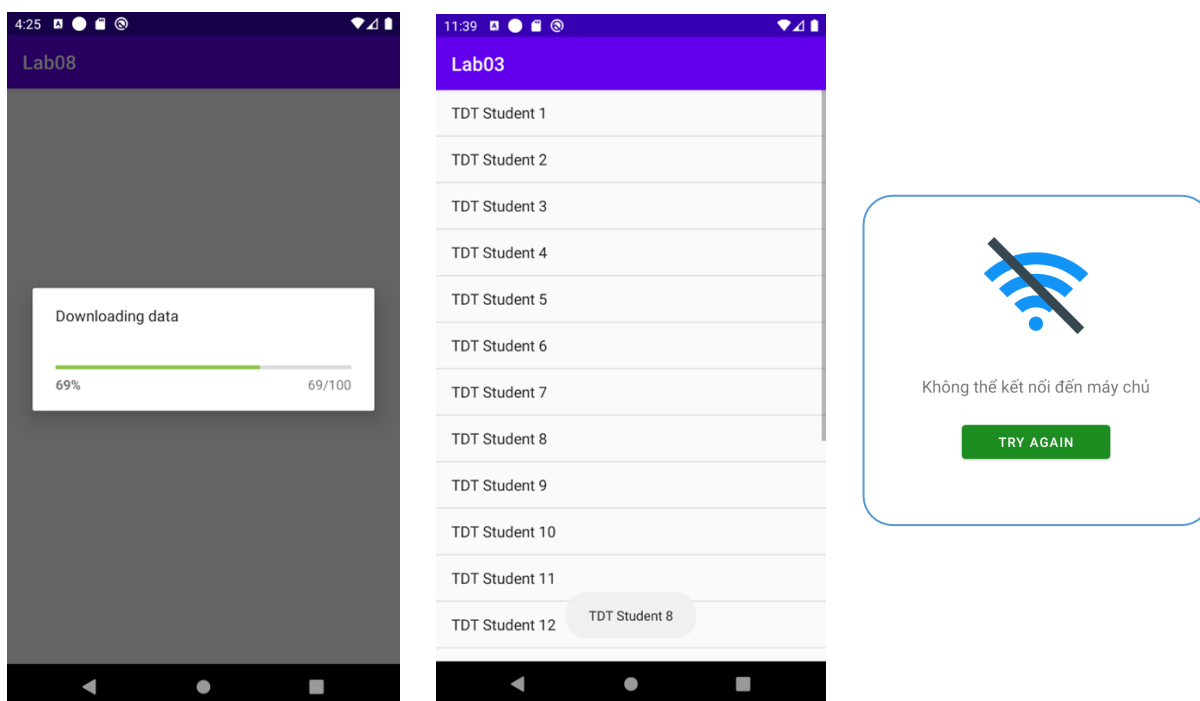


PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG - 503074

TUẦN 08

GIẢNG VIÊN BIÊN SOẠN: MAI VĂN MẠNH

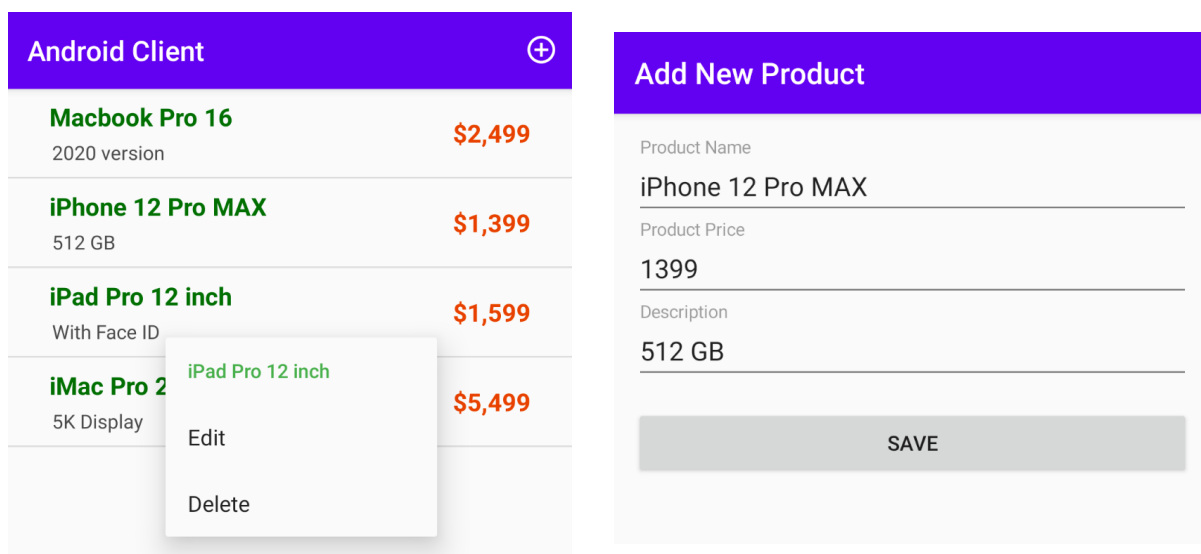
Bài 1. Viết ứng dụng hiển thị danh sách sinh viên như hình bên dưới.



Yêu cầu:

- 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 dùng **AsyncTask** để chạy một phương thức **loadData()** kết hợp với **Thread.sleep()** nhằm mô phỏng quá trình tải dữ liệu.
- Phương thức **loadData()** cũng sẽ mô phỏng việc xảy ra lỗi kết nối mạng bằng cách cho random lỗi với xác suất là 50%.
- 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.
- 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 (**ViewModel** và **LiveData**).

Bài 2. Cho trước mã nguồn của ứng dụng có giao diện như bên dưới.



Mô tả giao diện:

- Ứng dụng có hai màn hình: màn hình hiển thị danh sách và màn hình thêm/cập nhật.
- Màn hình chính cung cấp context menu với hai chức năng: edit và delete.
- Khi click vào một sản phẩm trong màn hình chính, màn hình chỉnh sửa thông tin sản phẩm sẽ xuất hiện.
- Khi nhấn dấu + trên Option menu, màn hình thêm sản phẩm mới sẽ hiển thị.
- Các chức năng cơ bản đều đã được viết sẵn.

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 nâng cao:

- Ngay khi mở ứng dụng, nếu không có kết nối internet, ứng dụng cần hiển thị thông báo cho người dùng biết (dialog hoặc Toast). 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.

- 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. Ví dụ khi thêm một sản phẩm mới mà chưa có kết nối mạng thì vẫn cho thêm, ngay khi có kết nối mạng thì sản phẩm sẽ được upload lên server một cách tự động.

Hướng dẫn thiết lập Server:

- Cài đặt công cụ XAMPP (đã cài sẵn trên phòng máy thực hành).
- Chép thư mục **api** được cung cấp sẵn vào thư mục **htdocs** của XAMPP.
- Mở tập tin **database_config.php**, điều chỉnh lại thông số cho phù hợp với máy.
- Vào PhpMyAdmin, import tập tin **android_lab08_products_db.sql** để tạo database.
- Truy cập http://localhost/all_products.php để kiểm tra kết quả (máy tính). **Trên máy ảo Android, đổi localhost thành 10.0.2.2.**

Bài 3. Ứng dụng trong bài tập 1 có nhược điểm dễ thấy khi xoay màn hình, đó là việc load dữ liệu sẽ phải thực hiện lại từ đầu. Vấn đề thậm chí còn nghiêm trọng hơn nếu chúng ta xoay màn hình khi quá trình load dữ liệu chưa toàn tất. Để khắc phục, có thể áp dụng một số cơ chế sau:

- ViewModel + Live Data
- Retained Fragment
- AsyncTaskLoader

Hãy tìm hiểu các cơ chế trên và đưa vào sử dụng thay thế cho AsyncTask.