BÁO CÁO THỰC HÀNH

**Môn học: Bảo mật web và ứng dụng**

**Kỳ báo cáo: Buổi 5 (Session 5)**

**Tên chủ đề:** Basic Android Secure Programming

*GVHD: Nghi Hoàng Khoa*

*Ngày báo cáo: 1/4/2020*

**Nhóm: 1**

1. **THÔNG TIN CHUNG:**

Lớp: NT213.L21.ANTN

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Đoàn Thanh Phương | 18521267 | 18521267@gm.uit.edu.vn |
| 2 | Phạm Ngọc Tâm | 18521371 | 18521371@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-2)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Yêu cầu 2 | 100% |
| 2 | Yêu cầu 3 | 100% |
| 3 | Yêu cầu 4 | 100% |
| 4 | Yêu cầu 5 | 100% |
| 5 | Yêu cầu 6 | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

1. Yêu cầu 1,2,3,4

Sinh viên xây dựng ứng dụng Android gồm 3 giao diện chức năng chính:

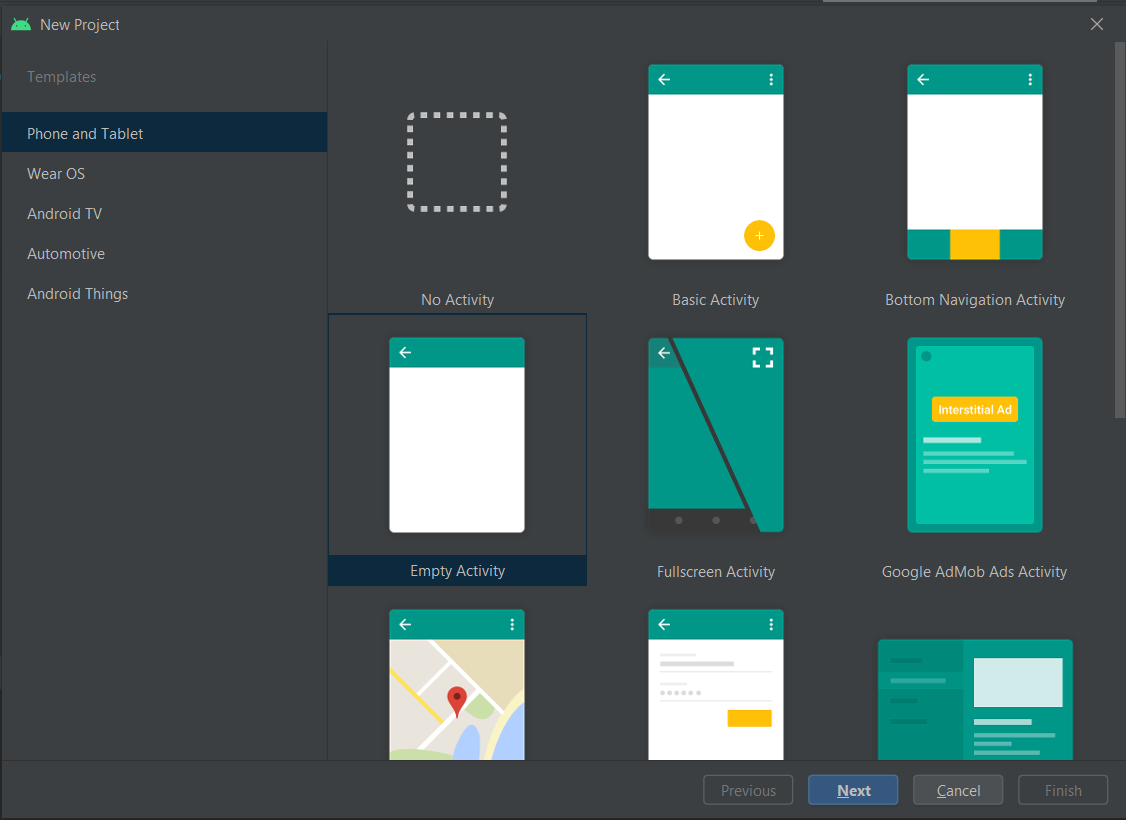
1) Register - Đăng ký thông tin với ứng dụng (email, username, password).

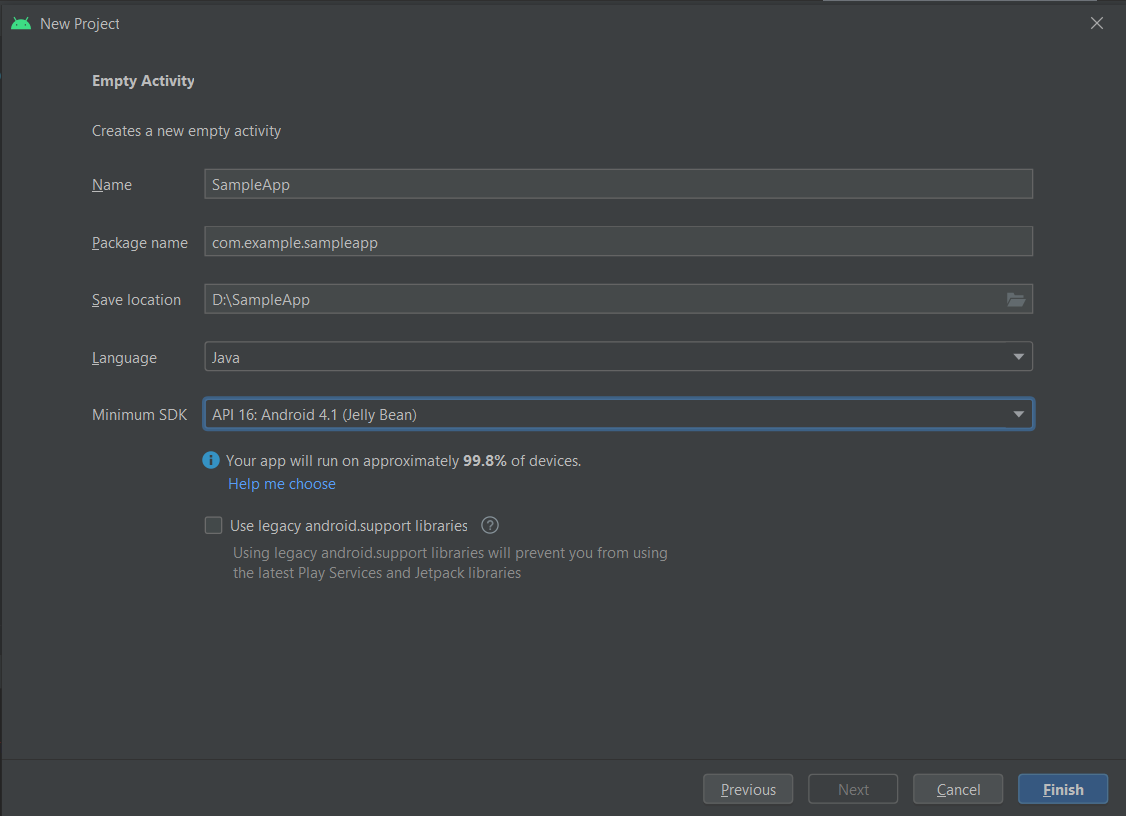
2) Login - Đăng nhập username, password).

3) Hiển thị thông tin người dùng (một lời chào có tên người dùng).

**Bước 1: Tạo project**

Thực hiện theo hướng dẫn bài lab

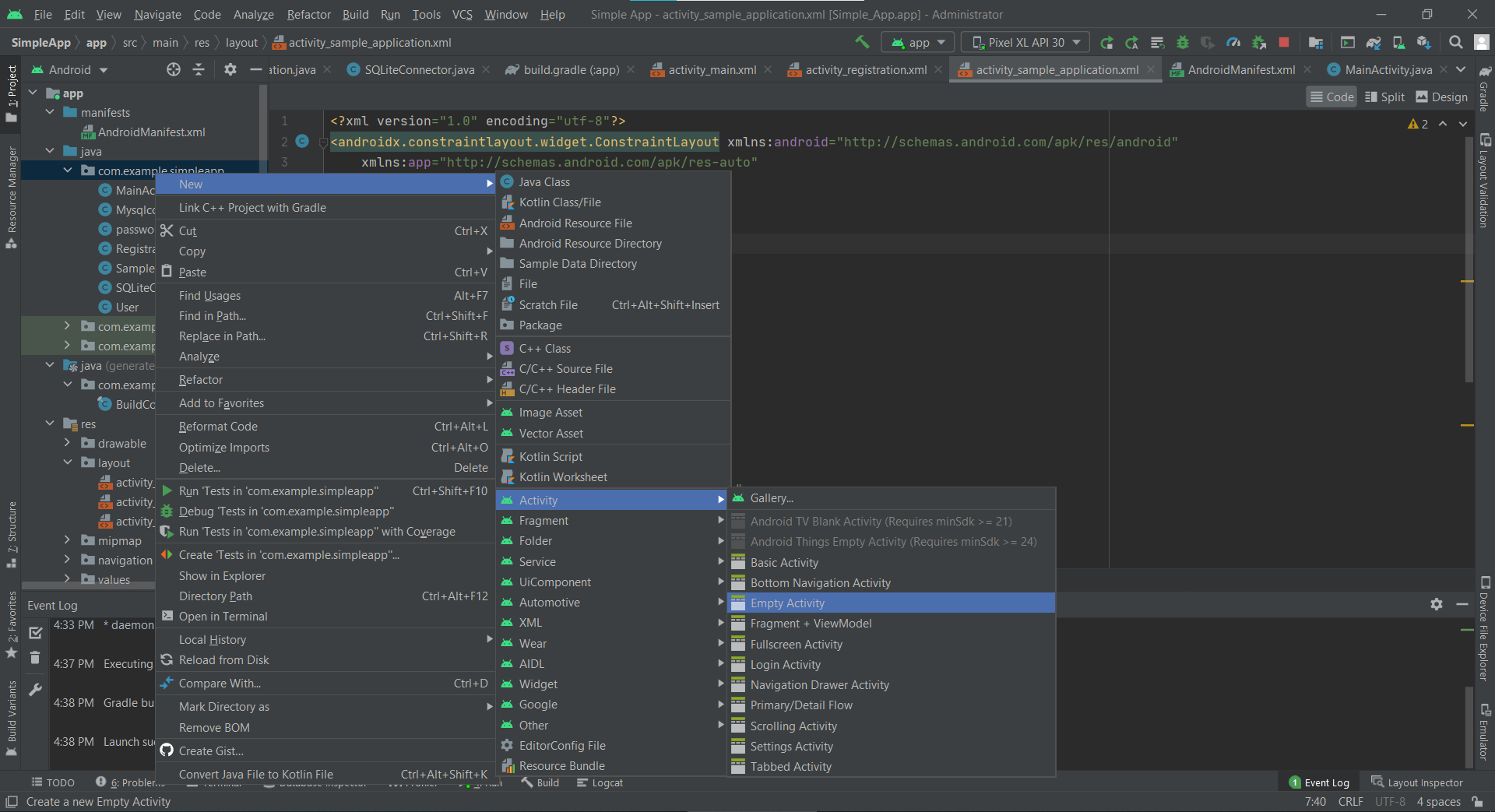


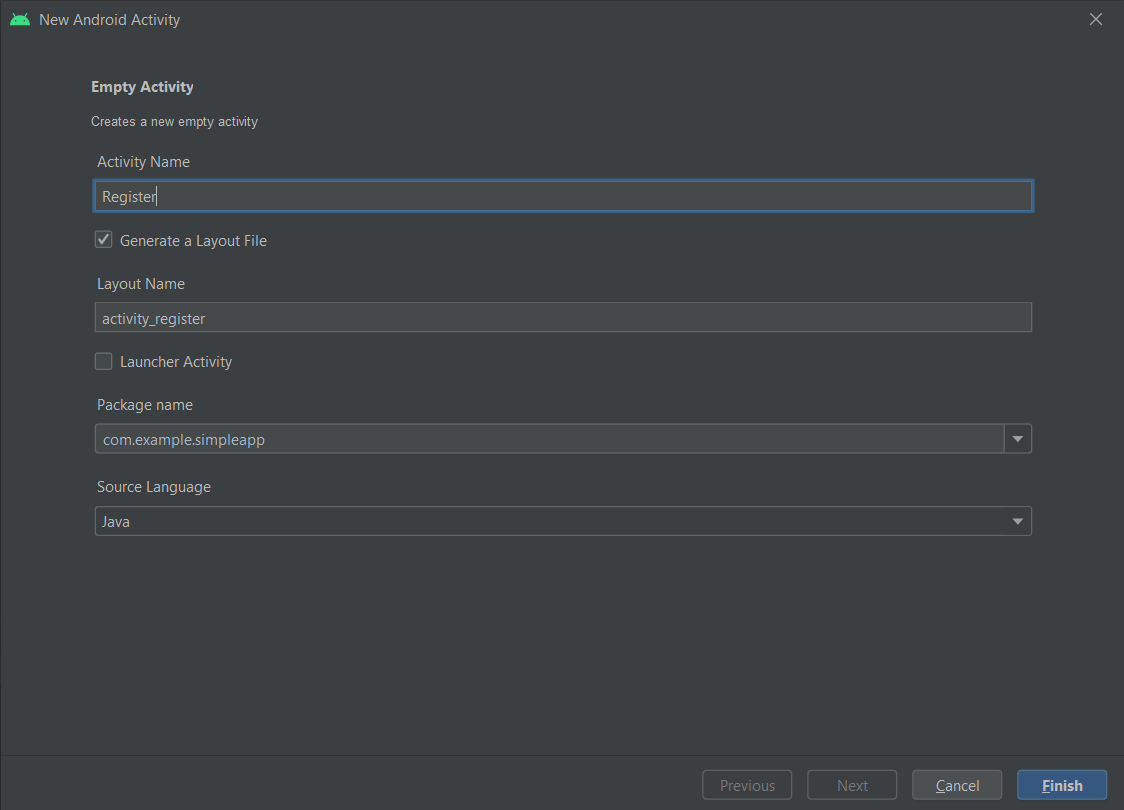


**Bước 2: Tạo các activity**

* Sử dụng các main\_activity để làm trang login
* Tạo thêm 2 activity tương ứng với 2 chức năng Register và Welcome

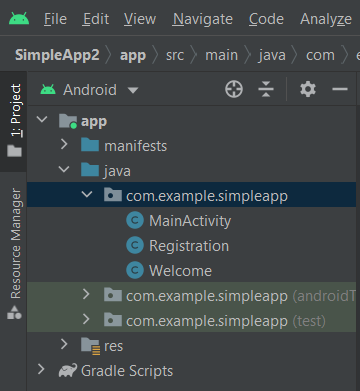
Vào thư mục java 🡺 com.example.simpleapp 🡺 New 🡺 Activity 🡺 Empty Activity





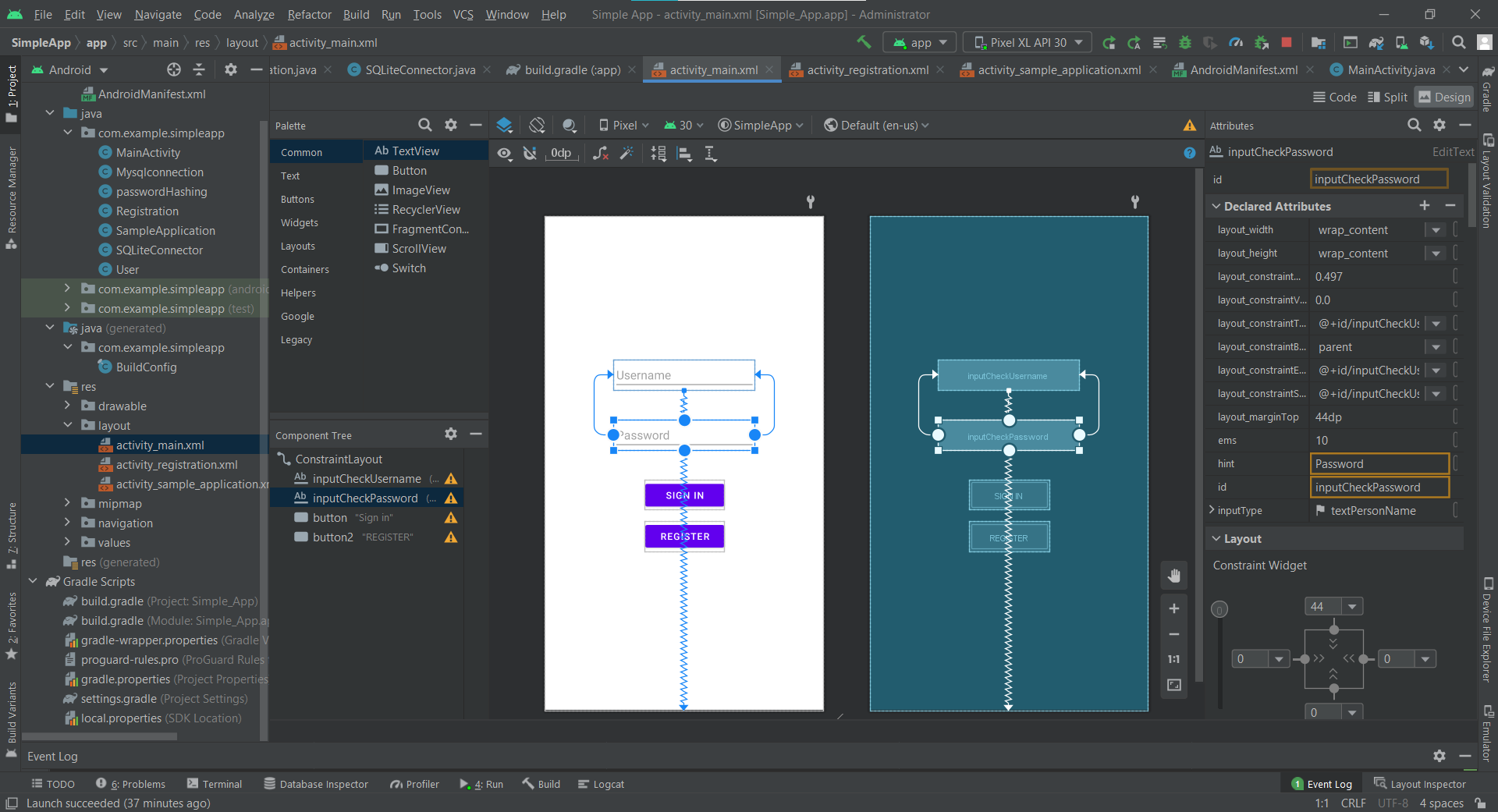
* Đặt tên cho Activity sau đó Finish
* Tương tự sẽ tạo thêm cho Welcome

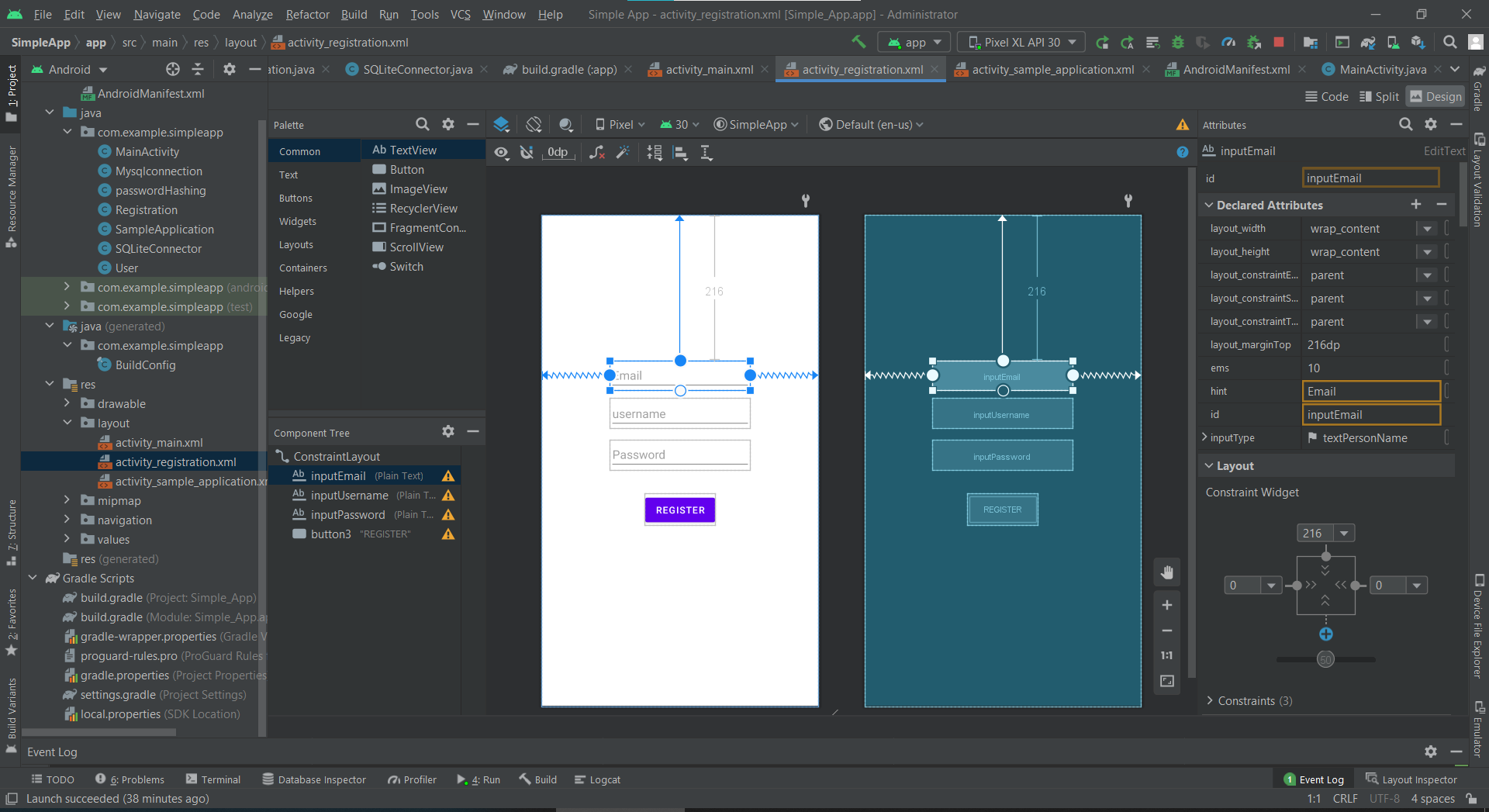
Kết quả sau khi tạo các Activity cần cho ứng dụng

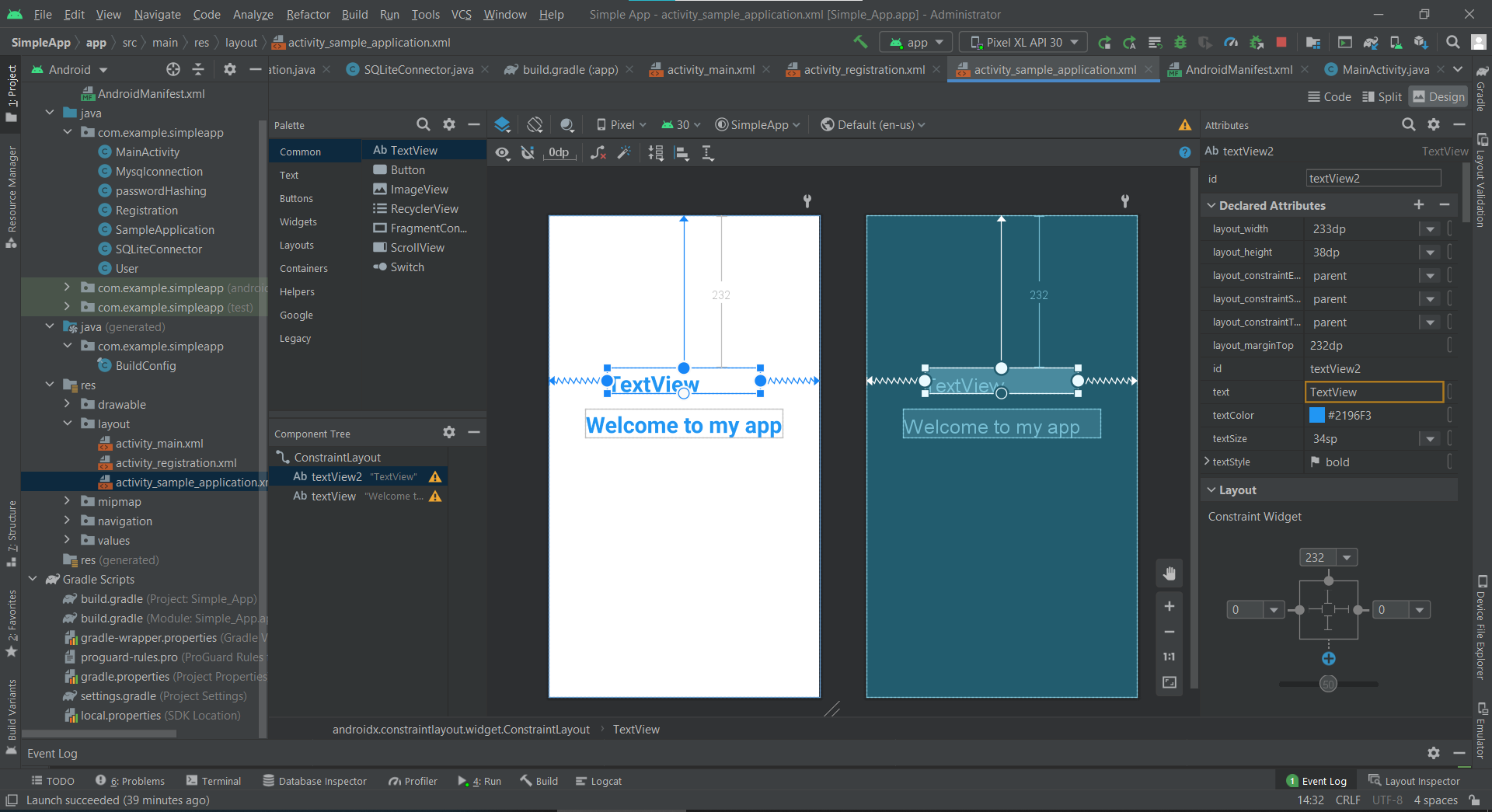


**Bước 3: Tạo layout và giao diện cho Activity**

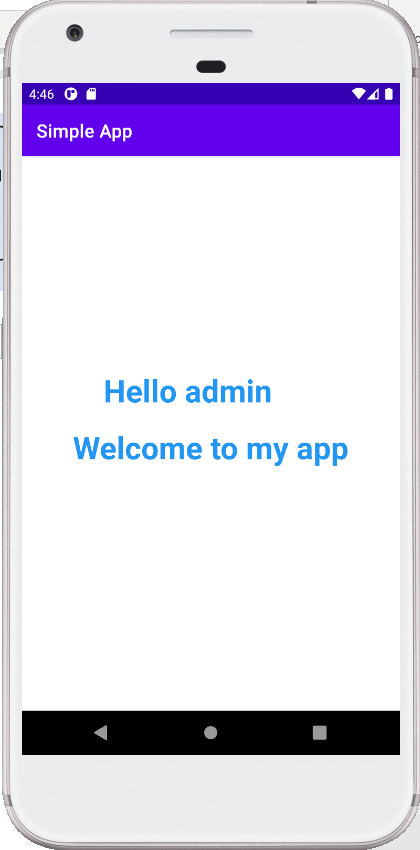
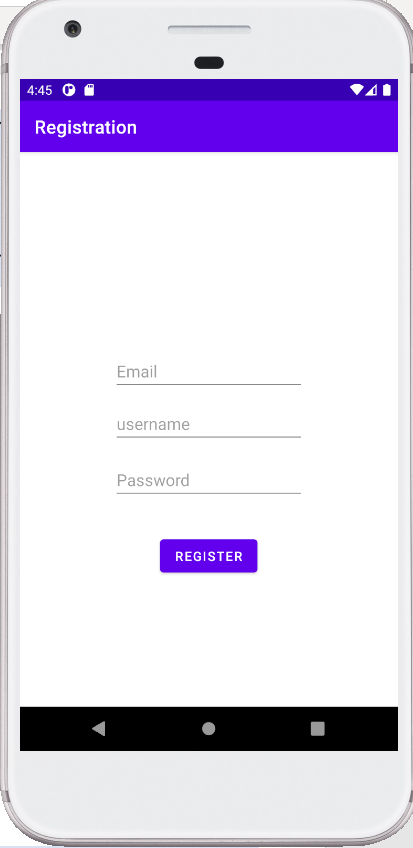
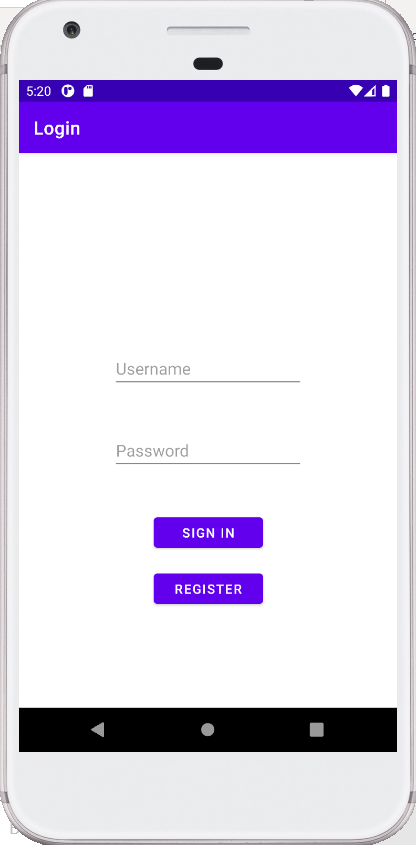
Chuyển sang chế độ Design để sử dụng kéo thả, thuận tiện hơn cho việc thiết kế UI và tập trung vào phần backend.

****

****

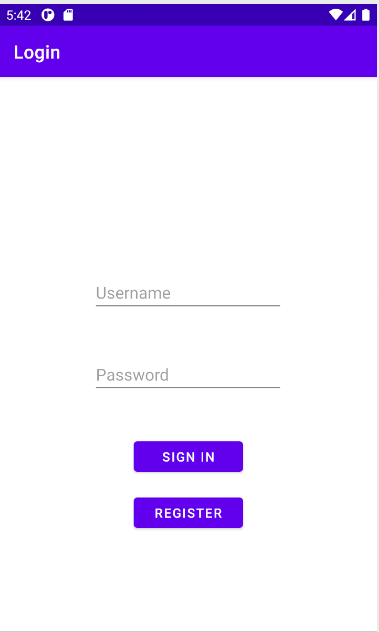
****

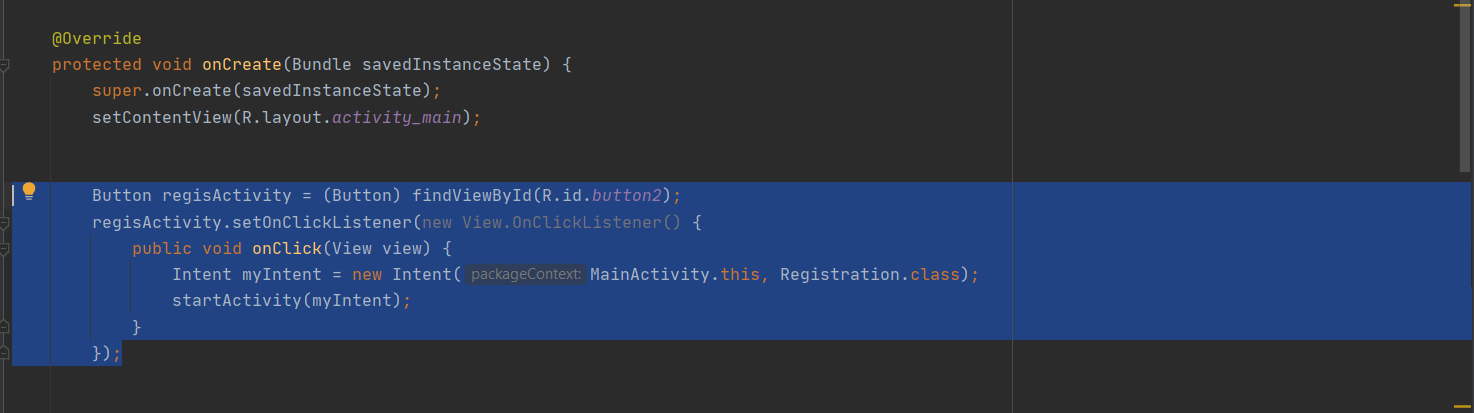
Sau khi hoàn thành kết quả giao diện các activity như bên dưới:



**Bước 4: Tạo sự kiện cho button và navigation giữa các activity**

* **Main Activity**

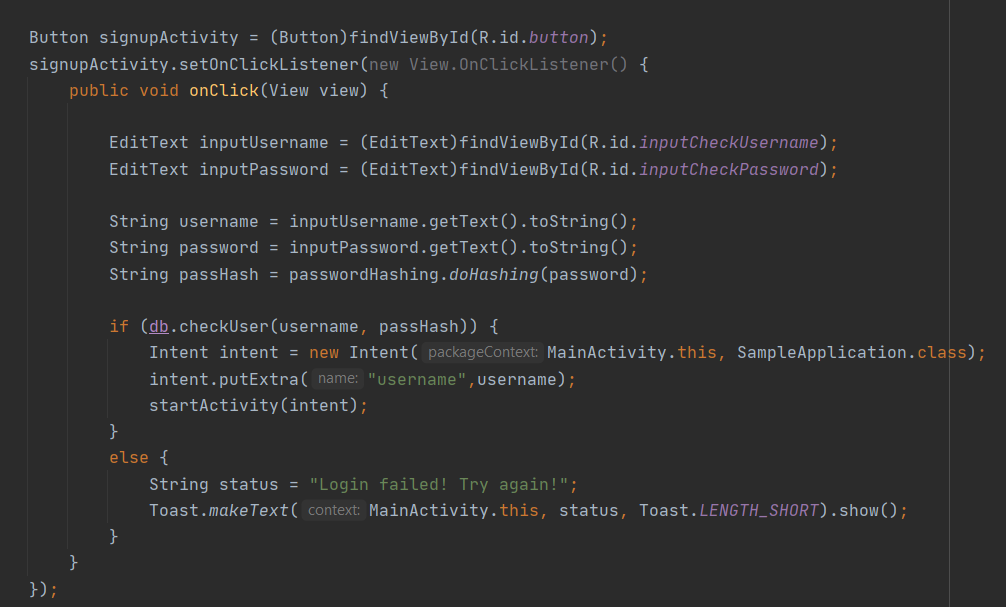
****

****

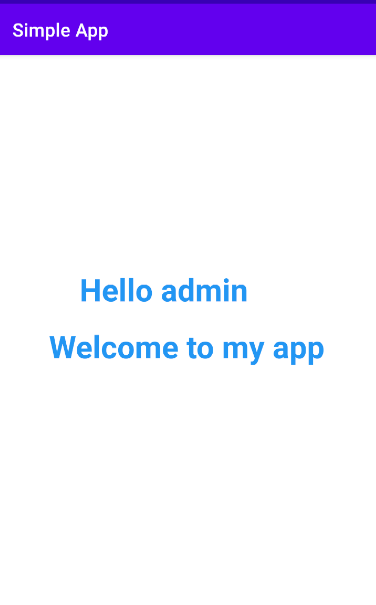
Sử dụng một intent để gọi Registration activity sau khi nhấn button Register trong Main activity.

Tiếp theo tạo sự kiện cho button Login trong main activity, khi nhấn nút nó sẽ lấy thông tin input username và password kiểm tra trong database nếu hợp lệ sẽ chuyển sang welcome, đồng thời gửi một intent kèm message là username sang Welcome activity. Ngược lại hiện một Toast với message là “Login failed! Try again”.

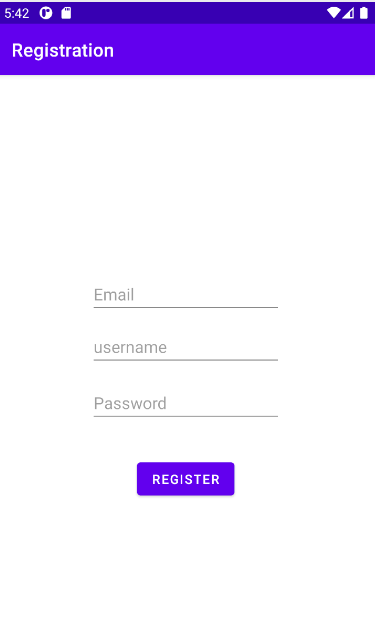
Tại điểm này password của người dùng khi nhập và sẽ được hash bằng MD5 sau đó đưa kết quả để kiểm tra trong database.



Kết quả sau khi login thành công với admin/admin



* **Registration Activity**

****

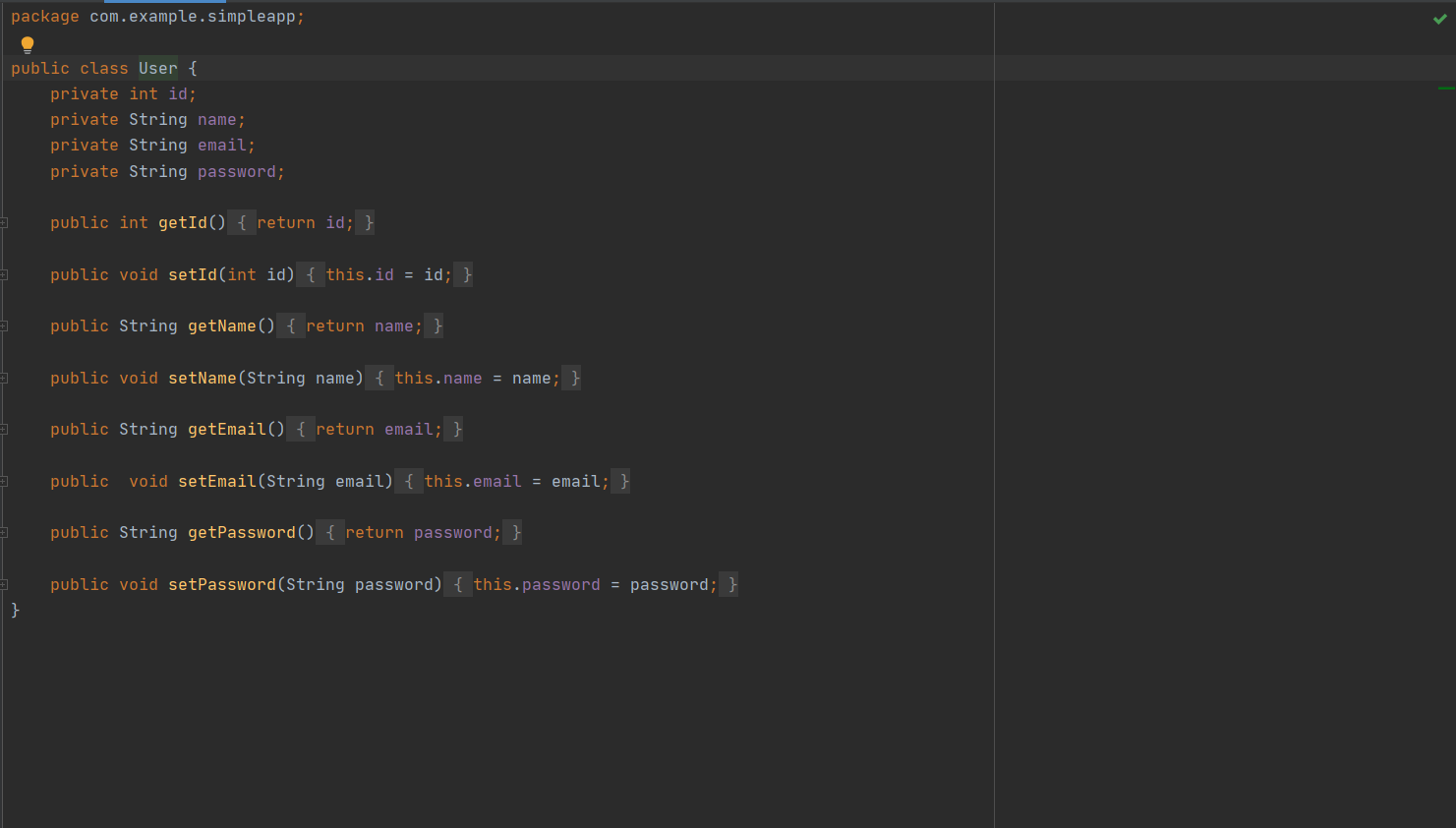
Khi nhấn button Register ở Registration activity, lấy thông tin người dùng nhập vào sau đó kiểm tra điều kiện nhập đủ hay chưa, nếu chưa sẽ thông báo là “Please fill all information”. Ngược lại sẽ thực hiện xử lí công việc chính.

****

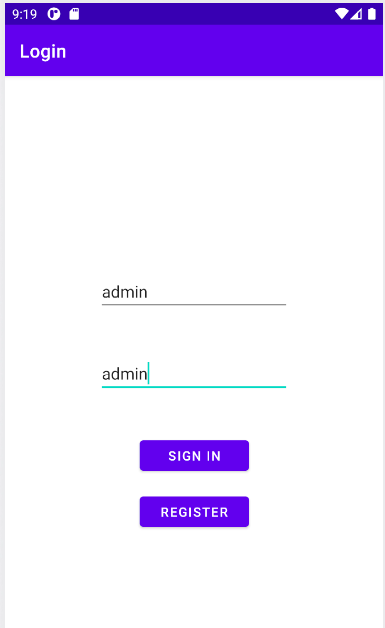
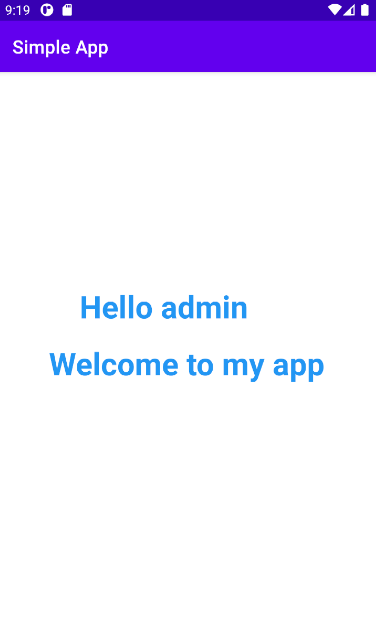
Ngược lại, nếu user nhập đủ thông tin email, username, password thì kiểm tra xem trong database có tồn tại user hay chưa sử dụng hàm checkUser trong SQLiteConnector bài lab đã cung cấp. Nếu tồn tại sẽ thông báo “User exist!” và cần nhập lại.

Ngược lại nếu chưa tồn tại sẽ tạo đối tượng User từ class đã tạo, thiết lập các thông tin cho user sau đó sử dụng hàm addUser được cung cấp để thêm user mới. Sau khi add thành công sẽ báo “create successfully!” chuyển sang main activity ngược lại sẽ báo lỗi “"Something went wrong! Try again!"





* **Demo và minh chứng**

****

**Sign up**

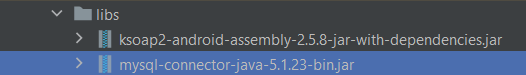
*\*Chi tiết xem ở* *phần link video đính kèm*

1. Yêu cầu 4

Tạo một cơ sở dữ liệu tương tự bên ngoài thiết bị, viết mã nguồn thực hiện kết nối đến CSDL này để truy vấn thay vì sử dụng SQLite.

Để thực hiện yêu cầu này nhóm sử dụng:

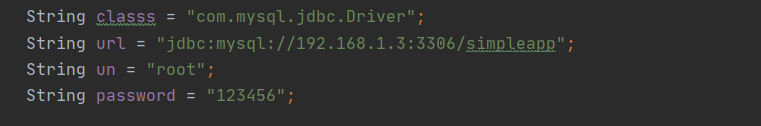
* MySql server 5.7 (UI phpmyadmin)
* Sử dụng mysql-connector-java



Trong code thêm 1 class Mysqlconnection với 3 function:

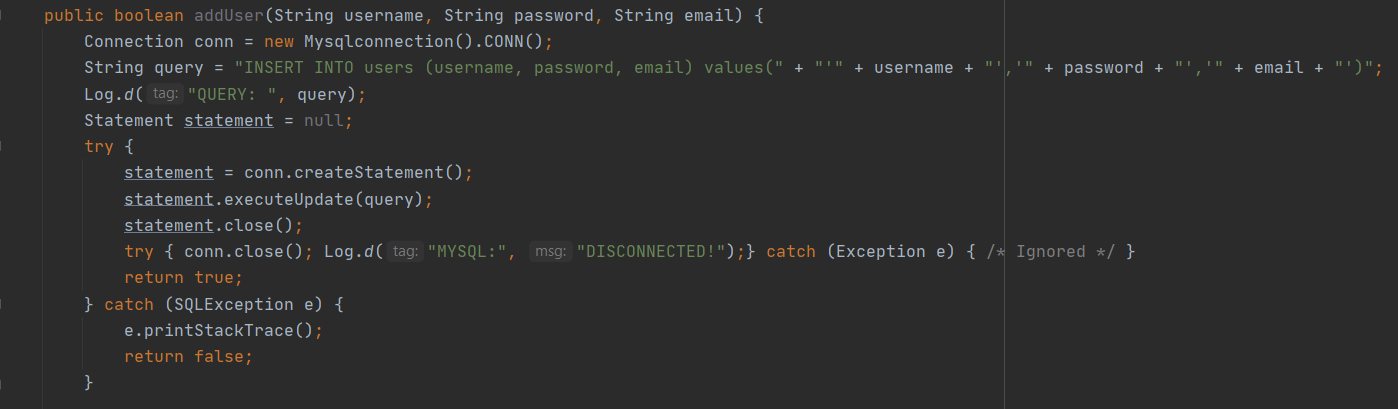
* Hàm tạo connection
* Hàm addUser
* Hàm checkUser

Khai báo các thông số để thực hiện kết nối đến MySQL server



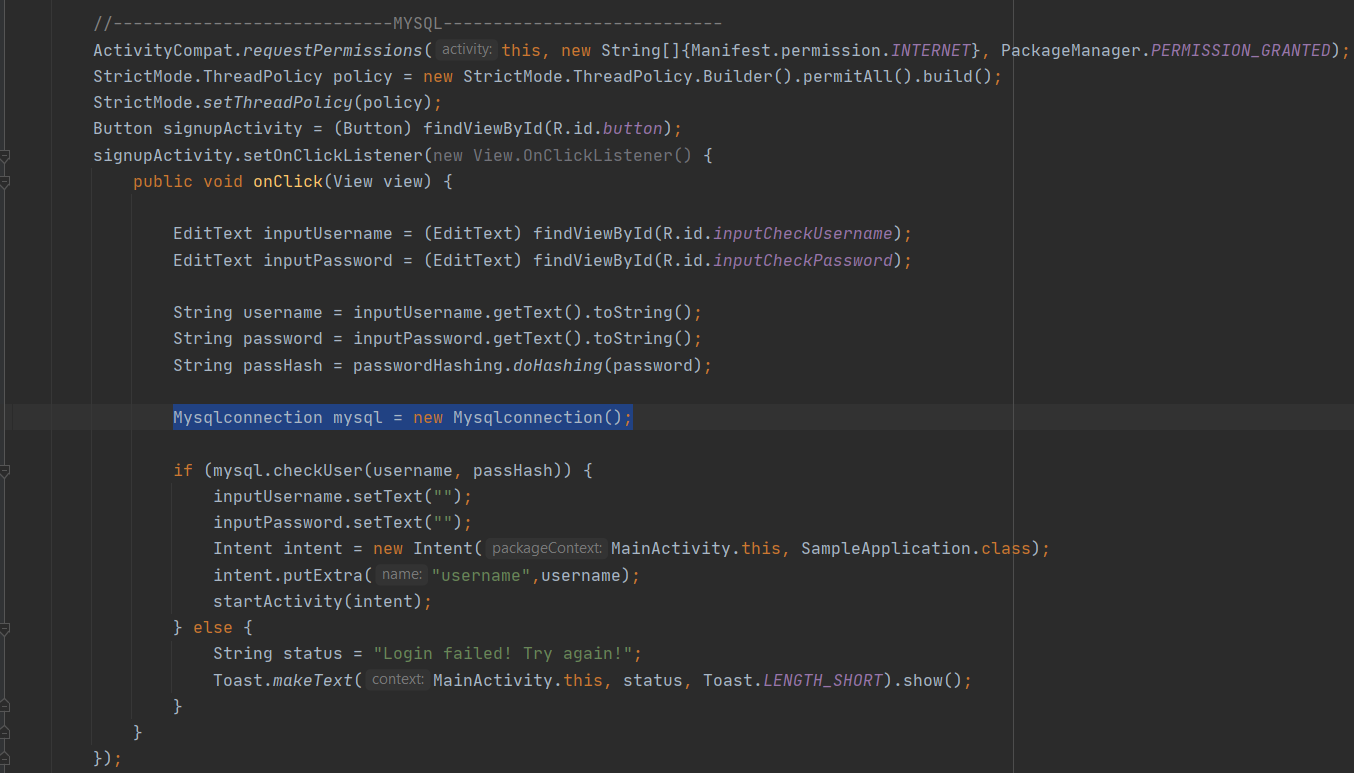


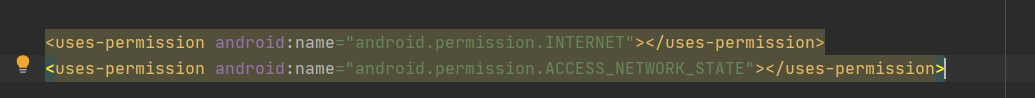




Một chút thay đổi so với SQLite: sau khi tạo class, tạo đối tượng và gọi các hàm checkUser và addUser tương tự như SQLite.

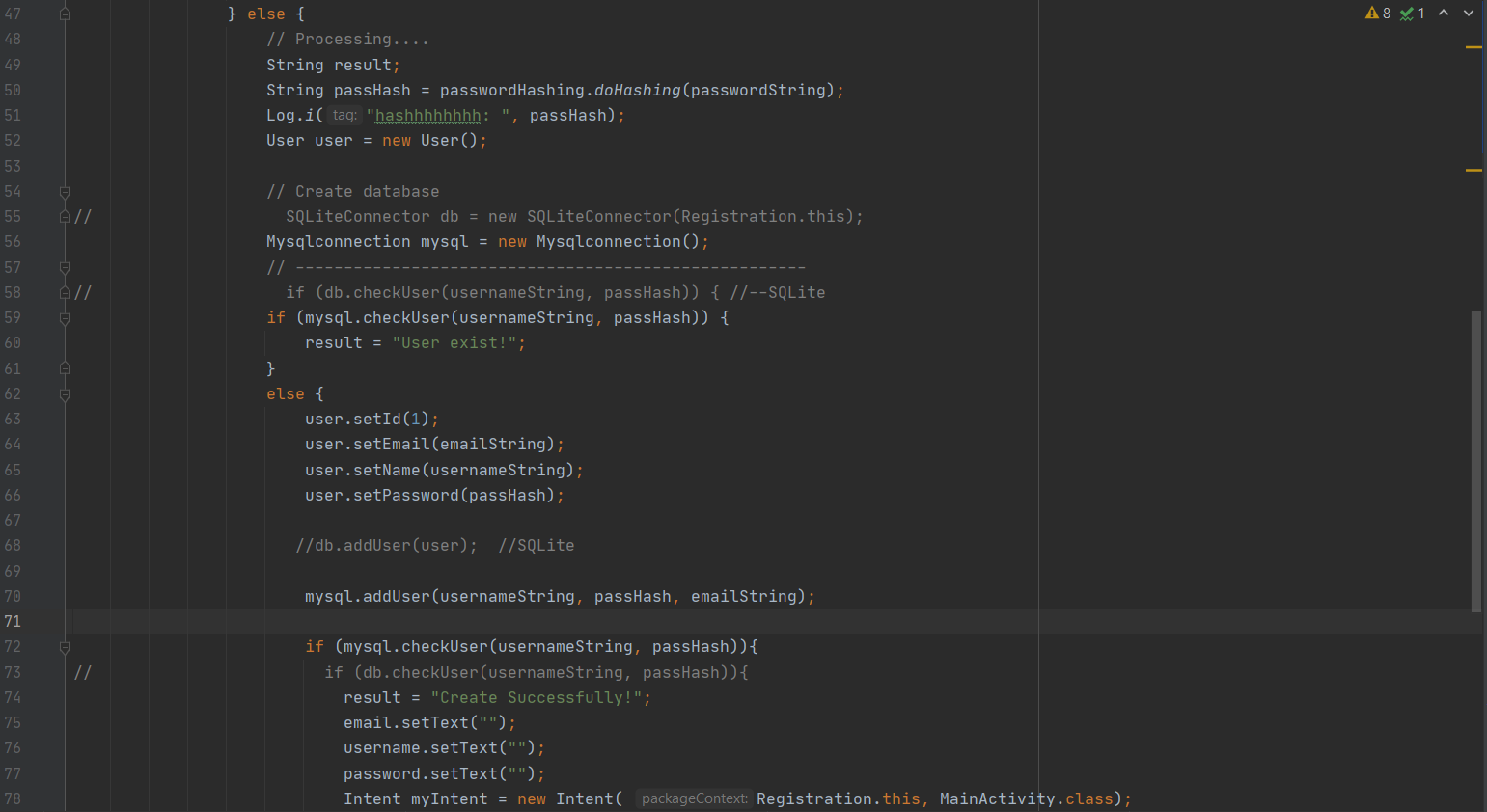
* Main acitivity





Lưu ý: để sử dụng MySQL server cần cấp thêm quyền sử dụng internet và thiết lập policy ở mức thấp cho phép tất nếu không sẽ không thể kết nối đến MySQL.

* Registration acitivity



* **Demo và minh chứng**

\*Chi tiết trong video đính kèm

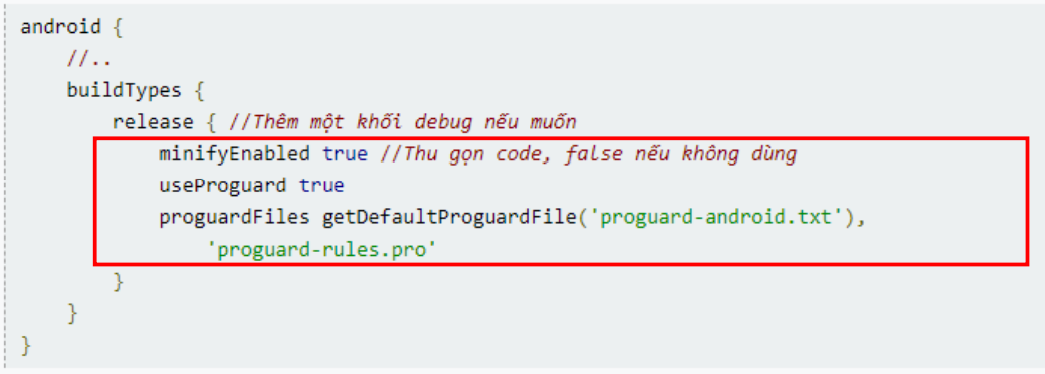
1. Yêu cầu 6

Với ứng dụng đã xây dựng, tìm hiểu và sử dụng công cụ ProGuard để tối ưu hóa mã nguồn. Trình bày khác biệt trước và sau khi sử dụng?

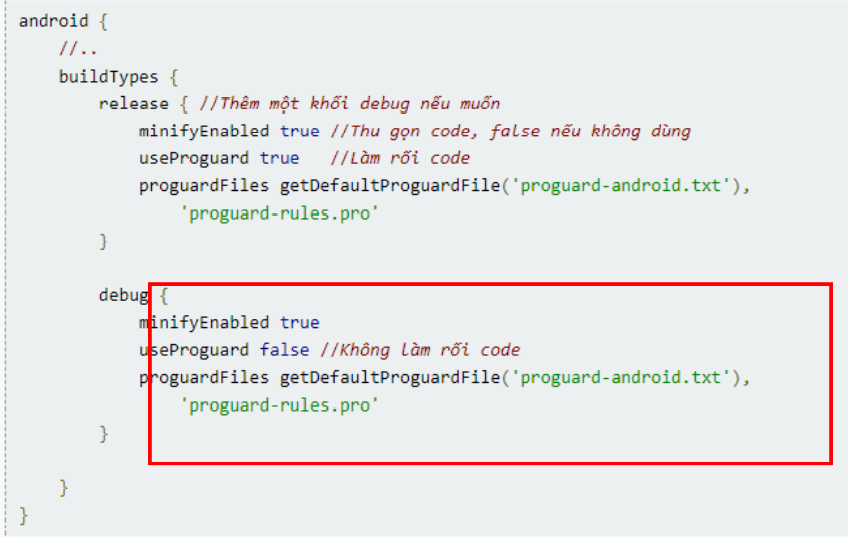
**ProGuard** là công cụ tích hợp sẵn trong Android Studio

* + Tính năng:
  + Thu gọn mã nguồn ứng → để dễ phân phối
  + Làm rối → để chống dịch ngược, vì tên các hàm, biến ... bị đổi tên khó đọc
  + Tối ưu để ứng dụng chạy nhanh hơn.
  + ProGuard là một công cụ rút gọn (shrink), tối ưu hoá (optimize) và làm mờ (obfuscate) code.
  + Sử dụng ProGuard để bảo vệ và tối ưu ứng dụng Android.

Cách sử dụng ProGuard trong android studio



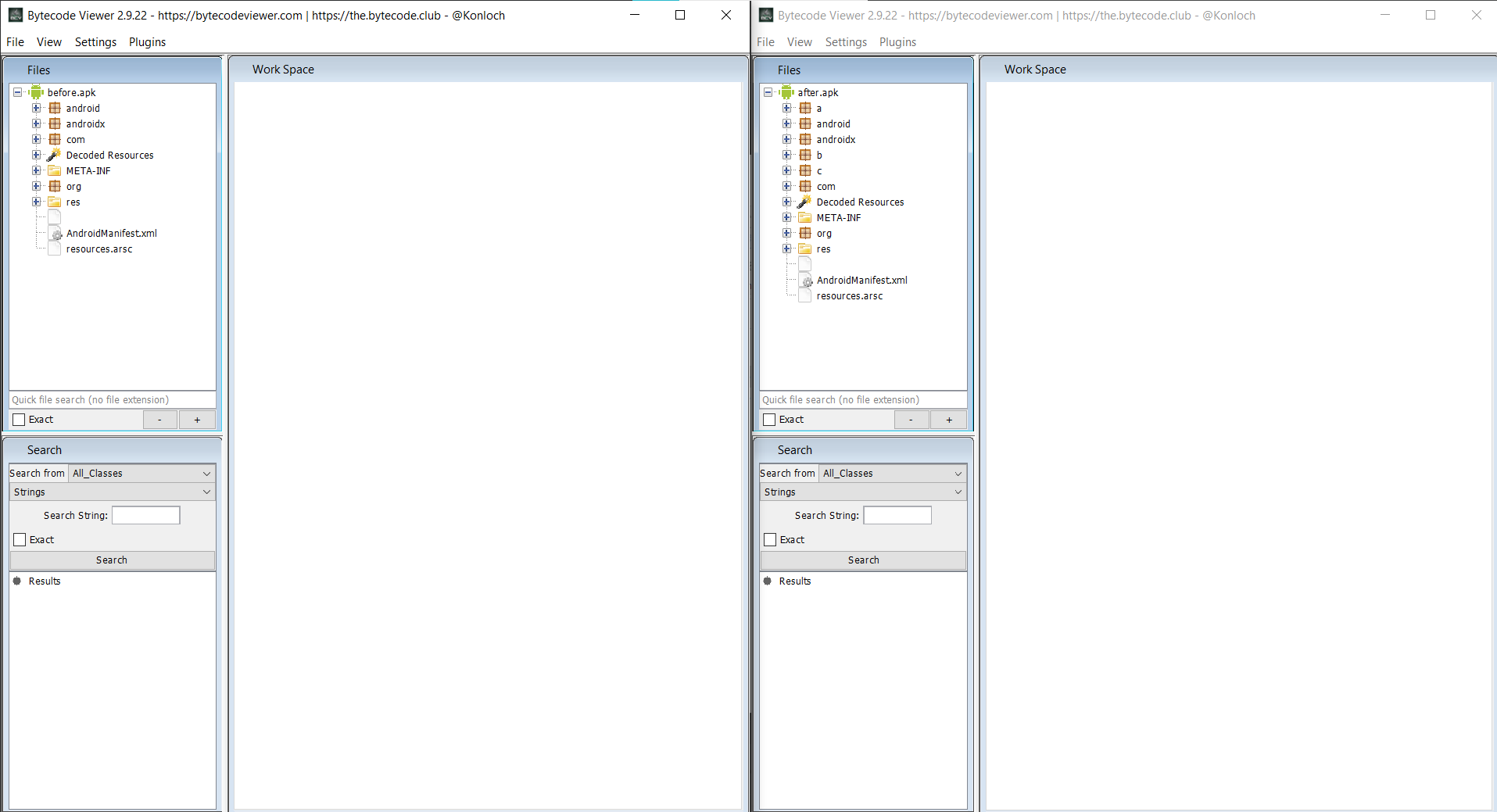
* MinifyEnabled: true bật tính năng ProGuard
* UseProguard: làm rối code



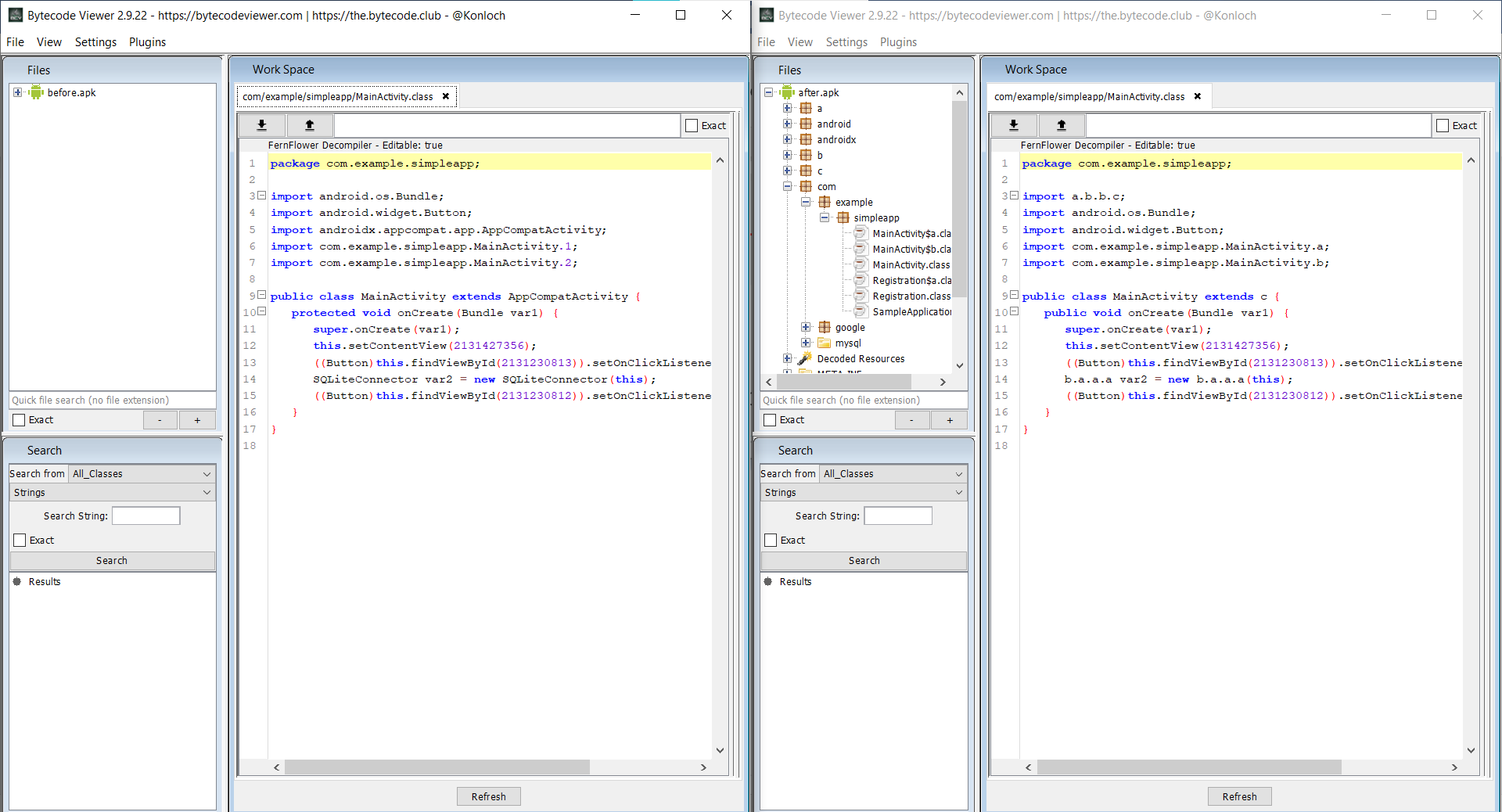
Để sử dụng ProGuard thêm một block debug tương tự như release bởi vì khi build ra file .apk sẽ nằm ở Debug.

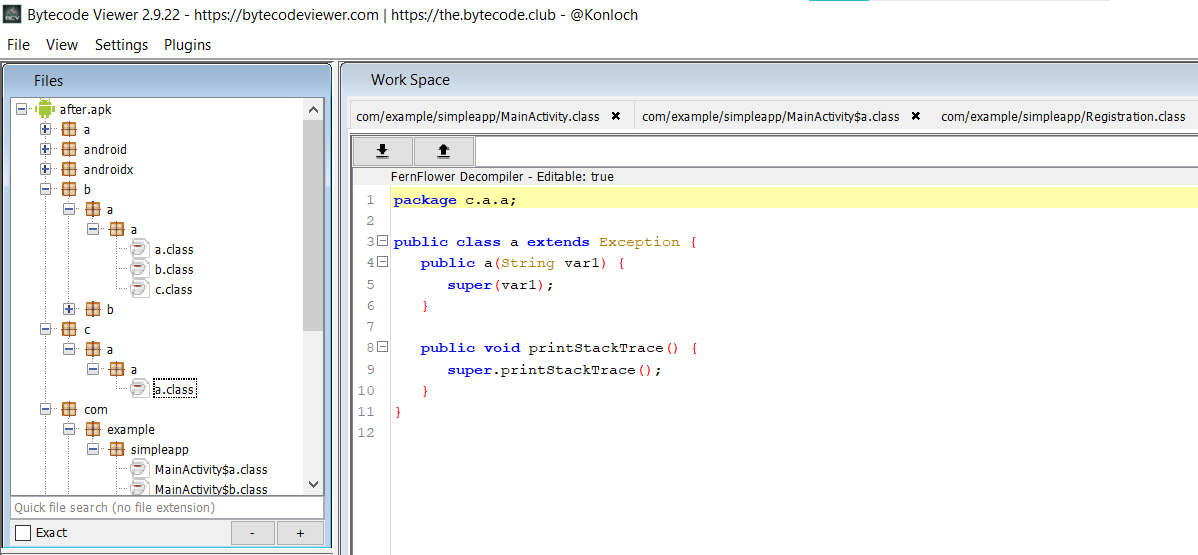
So sánh trước và sau khi sử dụng ProGuard

* Sau khi sử dụng proguard cấu trước sau khi extract từ file .apk số lượng các thư mục file tăng lên



* Các thư viện, class đều bị đổi thành các kí tự a,b,c,… 🡺 khiến cho attacker dịch ngược đọc hiểu sẽ rất khó.





Các thư mục file, class đều bị xáo trộn gây khó hiểu nếu dịch ngược để tấn công.

-- Hết --

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-2)