

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật Web và Ứng dụng

Kỳ báo cáo: Buổi 05 (Session 05)

Tên chủ đề: Basic Android Secure Programming

GVHD: Ngô Đức Hoàng Sơn

Ngày báo cáo: 14/5/2024

Nhóm: 09

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT213.O22.ATCL.2

STT	Họ và tên	MSSV	Email
1	Hồ Ngọc Thiện	21522620	21522620@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Kịch bản 01	100%
2	Kịch bản 02	100%
3	Kịch bản 03	100%
4	Kịch bản 04	100%
5	Kịch bản 05	100%
6	Kịch bản 06	100%
7	Kịch bản 07	100%

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

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

Yêu cầu 1: Sinh viên tiếp tục sửa lỗi Broadcast Receiver

Cũng giống như sửa lỗi ở activity PostLogin, ta cũng chuyển thuộc tính exported của Broadcast Receiver của ta thành “false”

```
<provider android:authorities="com.android.insecurebankv2.TrackerContentProvider" android:exported="true" android:grantUriPermissions="true">
    <receiver android:exported="false" android:name="com.android.insecurebankv2.MyBroadcastReceiver">
        <intent-filter>
            <action android:name="theBroadcast"/>
        </intent-filter>
    </receiver>
</provider>
```

Tiến hành build và kí.

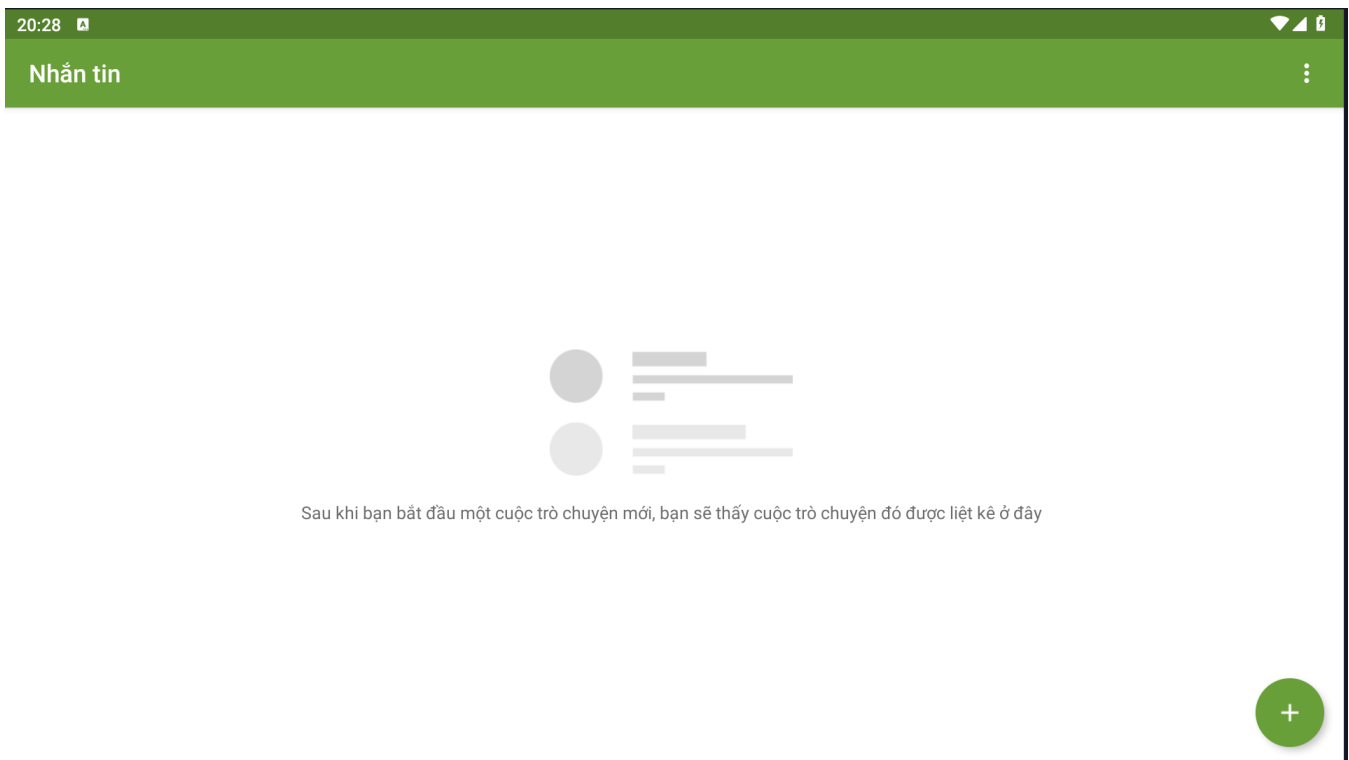
```
PS C:\Users\Ngoc Thien\Downloads> apktool b .\InsecureBankv2\ -o FixBroadcast.apk
I: Using Apktool 2.9.3
I: Checking whether sources has changed...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk into: FixBroadcast.apk
Press any key to continue . . .
```

```
Press any key to continue . . .
PS C:\Users\Ngoc Thien\Downloads> keytool -genkey -v -keystore FixBroadcast.keystore -alias FixBroadcast -keyalg RSA -ke
ysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
Enter the distinguished name. Provide a single dot (.) to leave a sub-component empty or press ENTER to use the default
value in braces.
What is your first and last name?
[Unknown]:
What is the name of your organizational unit?
[Unknown]:
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
[no]: yes
Generating 2,048 bit RSA key pair and self-signed certificate (SHA384withRSA) with a validity of 10,000 days
for: CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
[Storing FixBroadcast.keystore]
```

```
PS C:\Users\Ngoc Thien\Downloads> apksigner sign --ks .\FixBroadcast.keystore .\FixBroadcast.apk
Keystore password for signer #1:
```

Sau đó thử chạy lại đoạn code exploit và kiểm tra phần tin nhắn xem còn có thể tương tác với Broadcast Receiver này nữa không

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
    Button btnExploit = findViewById(R.id.btnExploit);
    Intent intent = new Intent( action: "theBroadcast");
    intent.putExtra( name: "phonenumber", value: "+84111111111");
    intent.putExtra( name: "newpass", value: "aaaaaa");
    sendBroadcast(intent);
}
```



Phần tin nhắn không hiện tin nhắn đổi password nữa, vậy ta đã sửa lỗi thành công.

Yêu cầu 2 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 vào ứng dụng (username, password).
- 3) Hiển thị thông tin người dùng (một lời chào có tên người dùng).

Giao diện của register

Register

Username

Password

Confirm Password

Sign up

Phần code:

Ta sử dụng SQLite để tạo 1 database local (lưu trữ trong thiết bị của người dùng)

```
EditText edtUsername = findViewById(R.id.edtUsername);
EditText edtPassword = findViewById(R.id.edtPassword);
EditText edtEmail = findViewById(R.id.edtEmail);
edtConfirmPassword = findViewById(R.id.edtConfirmPassword);
btnSignup = findViewById(R.id.btnSignup);
SQLiteDatabase mydatabase = openOrCreateDatabase( name: "login.db", MODE_PRIVATE, factory: null);
try {
    String sql = "CREATE TABLE users(username TEXT primary key, password TEXT, email TEXT)";
    mydatabase.execSQL(sql);
}
catch(Exception e){
    Log.e( tag: "Error", msg: "Table is already exists!!!");
}
```

Dưới đây là phần kiểm tra định dạng của username và password đang đăng kí, đồng thời INSERT thông tin đăng nhập vào database của người dùng.

```
btnSignup.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String username = edtUsername.getText().toString();
        String password = edtPassword.getText().toString();
        String confirmPassword = edtConfirmPassword.getText().toString();
        String email = edtEmail.getText().toString();
        if(password.length() < 8){
            Toast.makeText( context: SignupActivity.this, text: "Password needs at least 8 characters! Please try again.", Toast.LENGTH_SHORT).show();
        }
        else{
            if(password.equals(confirmPassword)){
                ContentValues myvalue = new ContentValues();
                myvalue.put("username",username);
                myvalue.put("password", password);
                myvalue.put("email", email);
                String msg = "";
                Cursor cursor = mydatabase.rawQuery( sql: "SELECT username FROM users WHERE username = ?", new String[]{username});
                if(cursor.getCount() > 0){
                    msg = "User is already exist!";
                }
                else{
                    long insertResult = mydatabase.insert( table: "users", nullColumnHack: null, myvalue);
                    if(insertResult != -1){
                        msg = "Sign up successfully!";
                        Intent intent = new Intent( packageContext: SignupActivity.this, MainActivity.class);
                        startActivity(intent);
                    }
                    else{
                        msg = "Fail to sign up!";
                    }
                }
            }
        }
    }
});
```

```
        else{
            msg = "Fail to sign up!";
        }

        Toast.makeText(context: SignupActivity.this, msg, Toast.LENGTH_SHORT).show();
    }
    else{
        String msg = "Your password doesn't match! Please try again.";
        Toast.makeText(context: SignupActivity.this, msg, Toast.LENGTH_SHORT).show();
    }
}
```

Giao diện của trang đăng nhập

Login

Username

Password

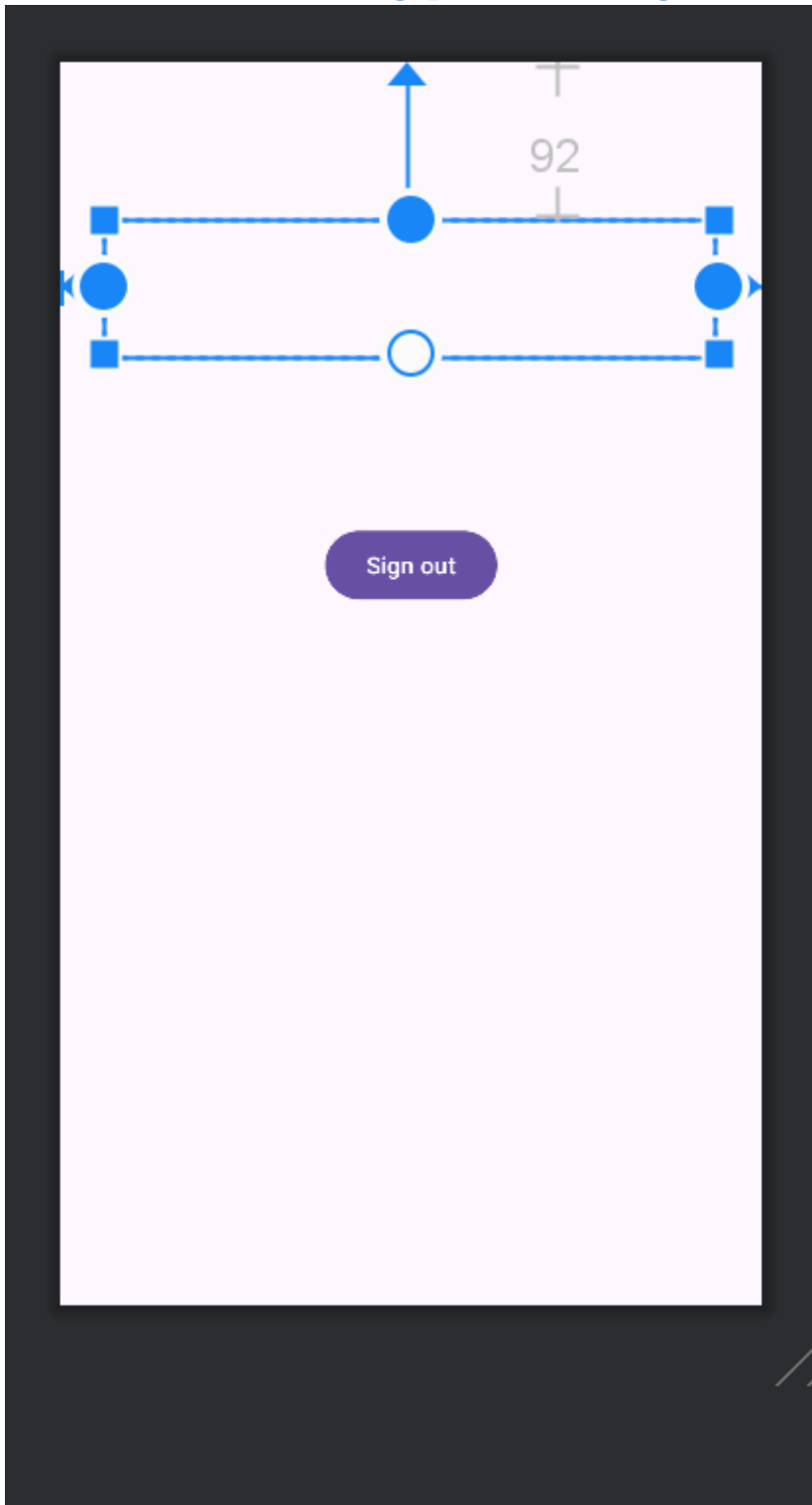
Sign in

Sign up

Còn đây là phần code xử lý đăng nhập, trong đó bao gồm kiểm tra thông tin đăng nhập có chứa trong database hay không, nếu có thực hiện truy cập và gửi username đến trang chào người dùng.

```
private boolean checkUserExist(String username, String password) {  
    // Mở kết nối đến cơ sở dữ liệu  
    SQLiteDatabase db = openOrCreateDatabase("Login.db", MODE_PRIVATE, null);  
  
    // Thực hiện truy vấn  
    Cursor cursor = db.rawQuery("SELECT * FROM users WHERE username = ? AND password = ?", new String[]{username, password});  
  
    // Kiểm tra xem có kết quả hay không  
    if (cursor.moveToFirst()) {  
        // User tồn tại  
        cursor.close();  
        db.close();  
        return true;  
    } else {  
        // User không tồn tại  
        cursor.close();  
        db.close();  
        return false;  
    }  
}
```

Giao diện của trang chào người dùng



Phần code xử lý bao gồm việc nhận dữ liệu và 1 nút để trở về màn hình chính

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_completed_login);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
    TextView txvNofi = findViewById(R.id.txvTitle);
    btnSignout = findViewById(R.id.btnSignout);
    btnSignout.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent( packageContext: CompletedLoginActivity.this, MainActivity.class);
            startActivity(intent);
            finish();
        }
    });
    Intent intent = getIntent();
    String username = intent.getStringExtra( name: "username");
    if (username != null && !username.isEmpty()) {
        String message = "Hello " + username;
        txvNofi.setText(message);
    }
}
```

Cấu hình file AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="LoginApp"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LoginApp"
        tools:targetApi="31">
        <activity
            android:name=".CompletedLoginActivity"
            android:exported="false" />
        <activity
            android:name=".SignupActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Kết quả

Hello nt1208

Sign out

Yêu cầu 3 Sinh viên viết mã nguồn Java cho chức năng đăng nhập và đăng ký, sử dụng tập tin **SQLiteConnector** được giảng viên cung cấp để thực hiện kết nối đến cơ sở dữ liệu SQLite với các yêu cầu bên dưới.

Thêm dòng sau để import SQLiteConnector cho trang login và register

```
import com.example.loginapp.sqliteconnector.SQLiteConnector;
```

Code của trang register, ta tạo đối tượng SQLiteConnector để thực hiện các thao tác trên database.

```
EditText edtUsername = findViewById(R.id.edtUsername);
EditText edtPassword = findViewById(R.id.edtPassword);
EditText edtEmail = findViewById(R.id.edtEmail);
edtConfirmPassword = findViewById(R.id.edtConfirmPassword);
btnSignup = findViewById(R.id.btnSignup);

SQLiteConnector mydb = new SQLiteConnector(this.getApplicationContext());
```

Sau đó kiểm tra định dạng và sử dụng hàm **addUser(User user)** để thêm user mới vào database.

```
public void onClick(View v) {
    User user = new User();
    String username = edtUsername.getText().toString();
    String password = edtPassword.getText().toString();
    String confirmpassword = edtConfirmPassword.getText().toString();
    String email = edtEmail.getText().toString();
    if(password.length() < 8){
        Toast.makeText(context: SignupActivity.this, text: "Password needs at least 8 characters! Please try again.", Toast.LENGTH_SHORT).show();
    }
    else{
        if(password.equals(confirmpassword)){
            user.setName(username);
            user.setPassword(password);
            user.setEmail(email);
            String msg = "";
            try{
                mydb.addUser(user);
                msg = "Sign up successfully!";
                Intent intent = new Intent( packageContext: SignupActivity.this, MainActivity.class);
                startActivity(intent);
            }
            catch (Exception e){
                msg = "Fail to sign up!";
            }
            Toast.makeText(context: SignupActivity.this, msg, Toast.LENGTH_SHORT).show();
        }
        else{
            String msg = "Your password doesn't match! Please try again.";
            Toast.makeText(context: SignupActivity.this, msg, Toast.LENGTH_SHORT).show();
        }
    }
}
```

Còn ở trang Login, ta sử dụng hàm **checkUser(String username, String password)** để kiểm tra xem user có tồn tại ở trong database hay không.

```
SQLiteConnector mydb = new SQLiteConnector(this.getApplicationContext());

btnSignin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String username = edtUsername.getText().toString();
        String password = edtPassword.getText().toString();
        if(username.isEmpty() || password.isEmpty()){
            Toast.makeText(context: MainActivity.this, text: "Invalid login! Please try again.", Toast.LENGTH_SHORT).show();
        }
        else if(password.length() < 8){
            Toast.makeText(context: MainActivity.this, text: "Invalid Password!", Toast.LENGTH_SHORT).show();
        }
        else{
            if (mydb.checkUser(username, password)) {
                // Chuyển đến CompletedSigninActivity
                Intent intent = new Intent( packageContext: MainActivity.this, CompletedLoginActivity.class);
                intent.putExtra( name: "username", username);
                startActivity(intent);
            } else {
                // Hiển thị thông báo "User is not exist!"
                Toast.makeText(context: MainActivity.this, text: "User does not exist!", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

Kết quả

Register

1234

.....

.....

1234@gmail.com

Sign up

Hello 1234

Sign out

Yêu cầu 4 Điều chỉnh mã nguồn để password được lưu và kiểm tra dưới dạng mã hash thay vì plaintext.

Trong build.gradle, ta thêm dòng sau để có thể sử dụng mã hóa bcrypt cho yêu cầu này

```
androidTestImplementation(libs.espresso.core)
implementation("at.favre.lib:bcrypt:0.10.2")
```

Đồng thời import dòng sau vào các file cần thiết

```
import at.favre.lib.crypto.bcrypt.BCrypt;
```

Ở file SQLiteConnector, ta thêm 1 hàm mới như sau để lấy mật khẩu đã được hash về để kiểm tra:

```
@SuppressWarnings("Range")
public String getHashPassword(String username){
    String[] columns = {
        COLUMN_USER_PASSWORD
    };
    SQLiteDatabase db = this.getReadableDatabase();
    String selection = COLUMN_USER_NAME + " = ?";
    // query the user table
    String[] selectionArgs = {username};
    Cursor cursor = db.query(TABLE_USER, columns, selection, selectionArgs, null, null, null, null);

    if (cursor.moveToFirst()) {
        return cursor.getString(cursor.getColumnIndex(COLUMN_USER_PASSWORD));
    }
    cursor.close();
    db.close();
    return cursor.getString(cursor.getColumnIndex(COLUMN_USER_PASSWORD));
}
```

Ở Register, ta sửa như hình dưới.

```
if(password.equals(confirmpassword)){
    String bcryptHashString = BCrypt.withDefaults().hashToString(12, password.toCharArray());
    user.setName(username);
    user.setEmail(email);
    user.setPassword(bcryptHashString);
    String msg = "";
```

Ở Login, ta sửa như hình dưới, sử dụng hàm **getHashPassword(String username)** để lấy thông tin mật khẩu trên database, sau đó sử dụng **verifyer()** để kiểm tra tính chính xác.

```
String hashPassword = mydb.getHashPassword(username);
BCrypt.Result result = BCrypt.verifyer().verify(password.toCharArray(), hashPassword);
if (result.verified){
    // Chuyển đến CompletedSignInActivity
    Intent intent = new Intent(packageContext, CompletedLoginActivity.class);
    intent.putExtra(name: "username", username);
    startActivity(intent);
} else {
    // Hiển thị thông báo "User is not exist!"
    Toast.makeText(context, MainActivity.this, "User does not exist!", Toast.LENGTH_SHORT).show();
}
```


Hello hash2

Sign out

Yêu cầu 5 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.

Gợi ý: Sinh viên có thể tận dụng CSQL MySQL và các tập tin xử lý PHP đã thực hiện ở bài thực hành trước và kết nối sử dụng Web REST API. Lưu ý, có cần điều chỉnh quyền hạn gì của ứng dụng hay không?

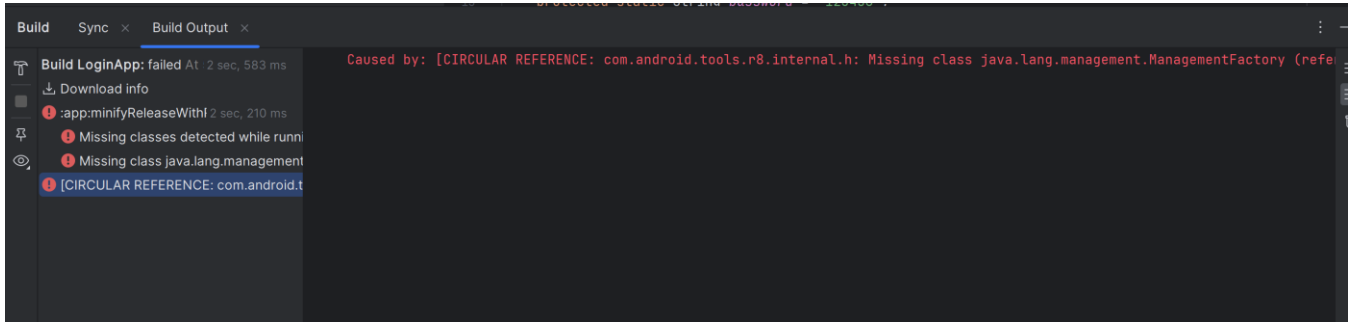
Thêm dòng sau để thêm thư viện của Mysql vào project

```
// https://mvnrepository.com/artifact/mysql/mysql-connector-java  
implementation("mysql:mysql-connector-java:8.0.23")
```

Sau đó, tạo 1 file java clas ConnecionDB như sau:

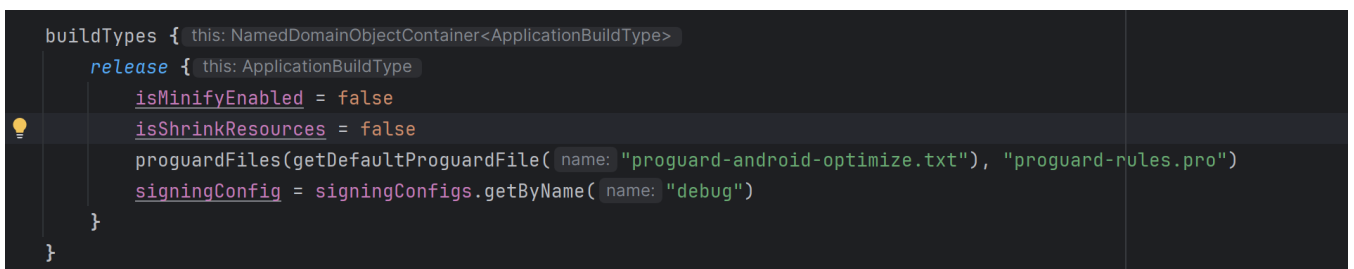
```
1 package com.example.loginapp;  
2 import android.util.Log;  
3  
4 import java.sql.Connection;  
5 import java.sql.DriverManager;  
6 import java.util.Objects;  
7  
8 3 usages  
9 public class ConnectionDB {  
10     1 usage  
11     protected static String db = "mydb";  
12     1 usage  
13     protected static String ip = "127.0.0.1";  
14     1 usage  
15     protected static String port = "3306";  
16     1 usage  
17     protected static String username = "root";  
18     1 usage  
19     protected static String password = "123456";  
20  
21     1 usage  
22     public Connection CONN(){  
23         Connection conn = null;  
24         try{  
25             Class.forName("com.mysql.jdbc.Driver");  
26             String connectionString = "jdbc:mysql://" + ip + ":" + port + "/" + db;  
27             conn = DriverManager.getConnection(connectionString, username, password);  
28         }  
29         catch (Exception e){  
30             Log.e("Error", Objects.requireNonNull(e.getMessage()));  
31         }  
32         return conn;  
33     }  
34 }
```

Tuy nhiên sau khi thử và biên dịch thì gặp lỗi như sau, em sẽ cố gắng khắc phục trong tương lai

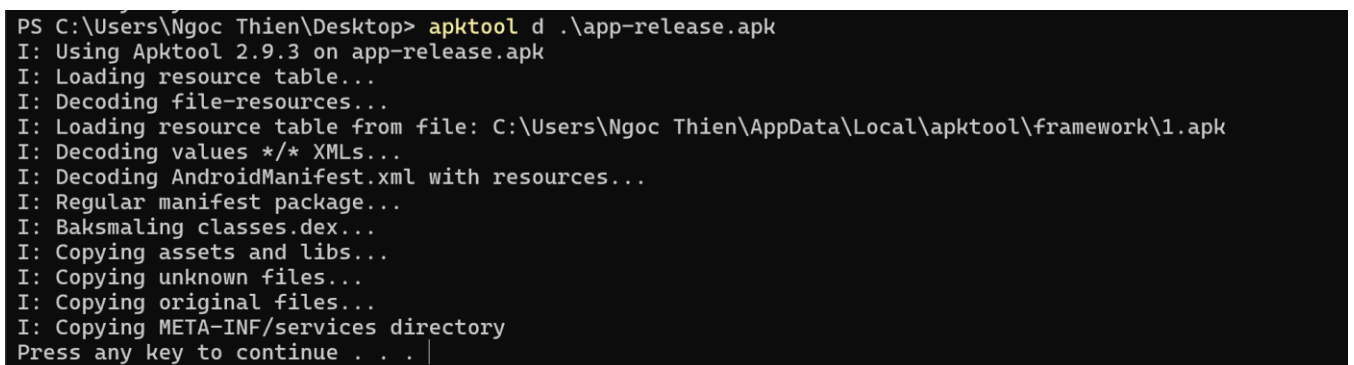


Yêu cầu 6 Với ứng dụng đã xây dựng, tìm hiểu và sử dụng công cụ ProGuard để tối ưu hóa mã nguồn. Trình bày khác biệt trước và sau khi sử dụng?

Khi không bật tính năng Proguard:



Thuộc tính **isMinifyEnabled** được set là false, ta thử decompile bằng apktool xem ta nhận được gì



Đây là thư mục ta có được sau khi decompile

Name	Date modified	Type	Size
assets	5/15/2024 4:32 PM	File folder	
kotlin	5/15/2024 4:32 PM	File folder	
META-INF	5/15/2024 4:32 PM	File folder	
original	5/15/2024 4:32 PM	File folder	
res	5/15/2024 4:32 PM	File folder	
smali	5/15/2024 4:32 PM	File folder	
unknown	5/15/2024 4:32 PM	File folder	
AndroidManifest.xml	5/15/2024 4:32 PM	XML Source File	3 KB
apktool.yml	5/15/2024 4:32 PM	Yaml Source File	3 KB

Ta dễ dàng có được các file smali, chính là mã nguồn của chương trình thực tế

Name	Date modified	Type	Size
sqliteconnector	5/15/2024 4:32 PM	File folder	
CompletedLoginActivity\$\$ExternalSyntheticLa...	5/15/2024 4:32 PM	SMALI File	1 KB
CompletedLoginActivity\$1.smali	5/15/2024 4:32 PM	SMALI File	2 KB
CompletedLoginActivity.smali	5/15/2024 4:32 PM	SMALI File	5 KB
ConnectionDB.smali	5/15/2024 4:32 PM	SMALI File	4 KB
MainActivity\$\$ExternalSyntheticLambda0.smali	5/15/2024 4:32 PM	SMALI File	1 KB
MainActivity\$\$ExternalSyntheticLambda1.smali	5/15/2024 4:32 PM	SMALI File	1 KB
MainActivity\$1.smali	5/15/2024 4:32 PM	SMALI File	6 KB
MainActivity\$2.smali	5/15/2024 4:32 PM	SMALI File	2 KB
MainActivity.smali	5/15/2024 4:32 PM	SMALI File	8 KB
R\$color.smali	5/15/2024 4:32 PM	SMALI File	1 KB
R\$drawable.smali	5/15/2024 4:32 PM	SMALI File	1 KB
R\$id.smali	5/15/2024 4:32 PM	SMALI File	1 KB
R\$layout.smali	5/15/2024 4:32 PM	SMALI File	1 KB
R\$mipmap.smali	5/15/2024 4:32 PM	SMALI File	1 KB
R\$string.smali	5/15/2024 4:32 PM	SMALI File	1 KB
R\$style.smali	5/15/2024 4:32 PM	SMALI File	1 KB
R\$xml.smali	5/15/2024 4:32 PM	SMALI File	1 KB
R.smali	5/15/2024 4:32 PM	SMALI File	1 KB
SignupActivity\$\$ExternalSyntheticLambda0.smali	5/15/2024 4:32 PM	SMALI File	1 KB

Khi sử dụng Proguard, ta chuyển trạng thái của thuộc tính isMinifyEnabled thành true:

```
buildTypes { this: NamedDomainObjectContainer<ApplicationBuildType>
    release { this: ApplicationBuildType
        isMinifyEnabled = true
        isShrinkResources = false
        proguardFiles(getDefaultProguardFile( name: "proguard-android-optimize.txt"), "proguard-rules.pro")
        signingConfig = signingConfigs.getByName( name: "debug")
    }
}
```

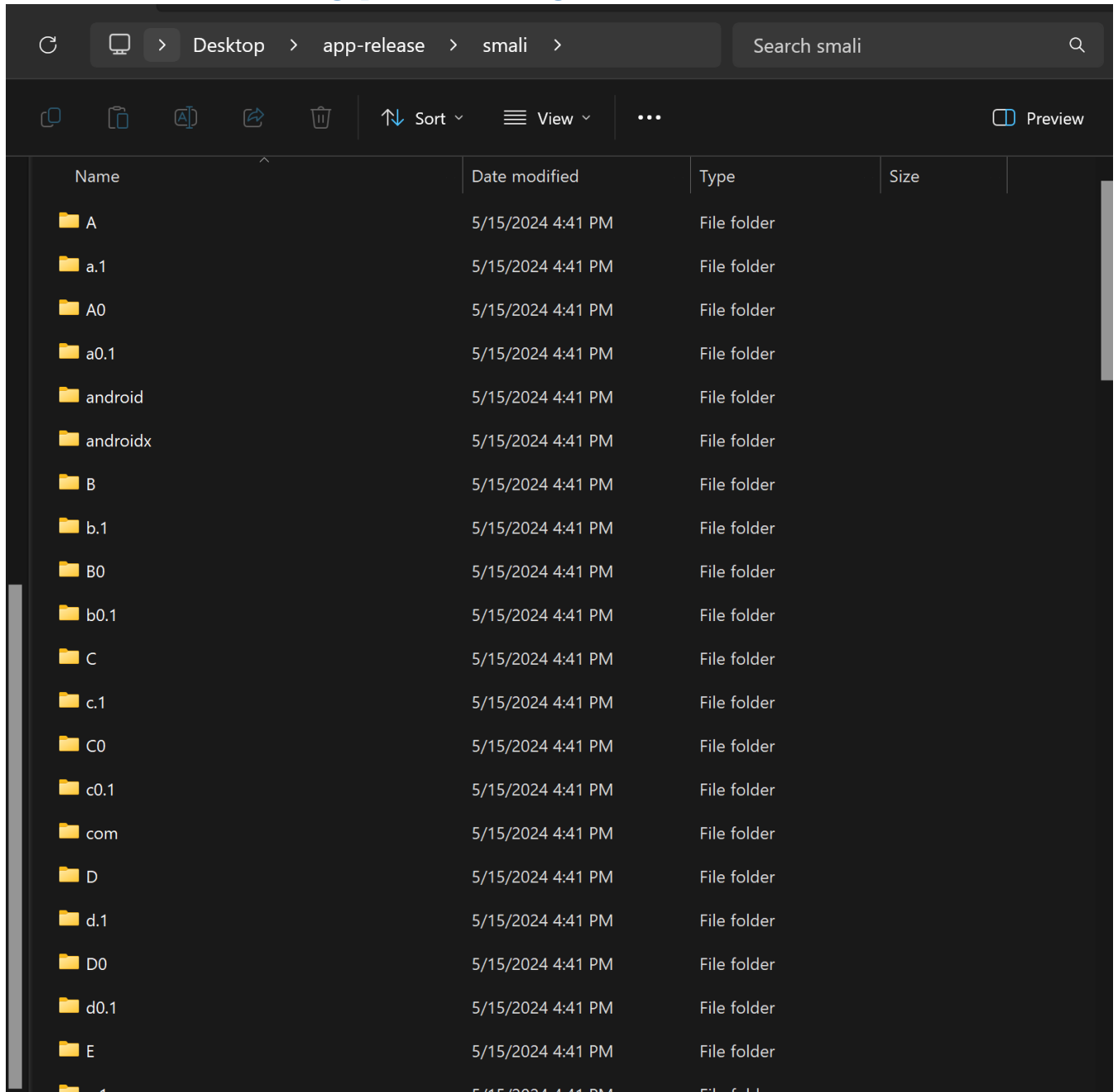
Ta có thể cấu hình file proguard-rules.pro như sau:

```
1  # Giữ lại các lớp và phương thức cần thiết
2  -keep class com.example.loginapp.** { *; }
3  -keep interface com.example.loginapp.** { *; }
4
5  # Giữ lại các file R
6  -keep class **R$* { *; }
7
8  # Giữ lại các file Manifest
9  -keep class **Manifest* { *; }
10
11 # Giữ lại các file activity
12 -keep public class * extends android.app.Activity {
13     public void **();
14 }
```

Thử decompile lại file apk, ta nhận được thư mục như sau:

Name	Date modified	Type	Size
assets	5/15/2024 4:41 PM	File folder	
kotlin	5/15/2024 4:41 PM	File folder	
original	5/15/2024 4:41 PM	File folder	
res	5/15/2024 4:41 PM	File folder	
smali	5/15/2024 4:41 PM	File folder	
unknown	5/15/2024 4:41 PM	File folder	
AndroidManifest.xml	5/15/2024 4:41 PM	XML Source File	3 KB
apktool.yml	5/15/2024 4:41 PM	Yaml Source File	1 KB

Tuy nhiên, khi truy cập vào folder smali, ta không còn thấy đường dẫn com.example.loginapp nữa, thay vào đó là các thư mục được Proguard tạo ra để tránh decompile



Như vậy, nhà phát triển có thể sử dụng proguard như 1 công cụ để che giấu mã nguồn, đồng thời giúp tinh giảm dung lượng của chương trình.

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File **.DOCX và .PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng **Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.**
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành đã đăng ký với GVHD-TH).
Ví dụ: [NT101.K11.ATCL]-Session1_Group3.
- Nếu báo cáo có nhiều file, nén tất cả file vào file **.ZIP** với cùng tên file báo cáo.
- **Không đặt tên đúng định dạng – yêu cầu, sẽ KHÔNG chấm điểm bài Lab.**
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

Bài sao chép, trể, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT