

BÁO CÁO THỰC HÀNH

Môn học: **BẢO MẬT WEB VÀ ỨNG DỤNG**

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

GVHD: *Ngô Khánh Khoa*

1. THÔNG TIN CHUNG:

Lớp: NT213.021.ANTN.1

STT	Họ và tên	MSSV	Email
01	Phạm Nguyễn Hải Anh	21522010	21522010@gm.uit.edu.vn
02	Nguyễn Nhật Quân	21522497	21522497@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Nội dung
01	Yêu cầu 1 - Sinh viên tiếp tục sửa lỗi Broadcast Receivers.
02	Yêu cầu 2 - Sinh viên xây dựng ứng dụng Android gồm 3 giao diện chức năng chính
03	Yêu cầu 3 - 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.
04	Yêu cầu 4 - Điều chỉnh mã nguồn để password được lưu và kiểm tra dưới dạng mã hash thay vì plaintext.
05	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.
06	Yêu cầu 6 - Tìm hiểu và sử dụng công cụ ProGuard để tối ưu hóa mã nguồn. Trình bày khác biệt trước và sau khi sử dụng?
Điểm tự đánh giá	

BÁO CÁO CHI TIẾT

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

Source Code: https://github.com/PNg-HA/BMW_Lab6

Yêu cầu 1 - Sinh viên tiếp tục sửa lỗi Broadcast Receivers. Vì ban đầu Broadcast Receivers `android:exported="true"` trong tập tin `AndroidManifest.xml`, nên có thể lắng nghe các lời gọi từ 1 app khác, nên ta chỉnh lại `android:exported="false"` để vá lại lỗ hổng Broadcast Receivers trong InsecureBankv2:

```
<receiver android:exported="false"
android:name="com.android.insecurebankv2.MyBroadCastReceiver">
    <intent-filter>
        <action android:name="theBroadcast"/>
    </intent-filter>
</receiver>
```

Yêu cầu 2 - Sinh viên xây dựng ứng dụng Android gồm 3 giao diện chức năng chính:

Giao diện đăng nhập:

- Code:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#D6D6D6"
tools:context=".MainActivity">

    <View
        android:id="@+id/view2"
        android:layout_width="412dp"
        android:layout_height="85dp"
        android:background="#1EA69A"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:text="Login"
        android:textColor="#FFFFFF"
        android:textSize="34sp"
        app:layout_constraintBottom_toBottomOf="@+id/view2"
        app:layout_constraintStart_toStartOf="@+id/view2"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/username"
```

```
        android:layout_width="300dp"
        android:layout_height="52dp"
        android:layout_marginTop="120dp"
        android:ems="10"
        android:hint="Username"
        android:inputType="text"
        android:password="false"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/view2" />

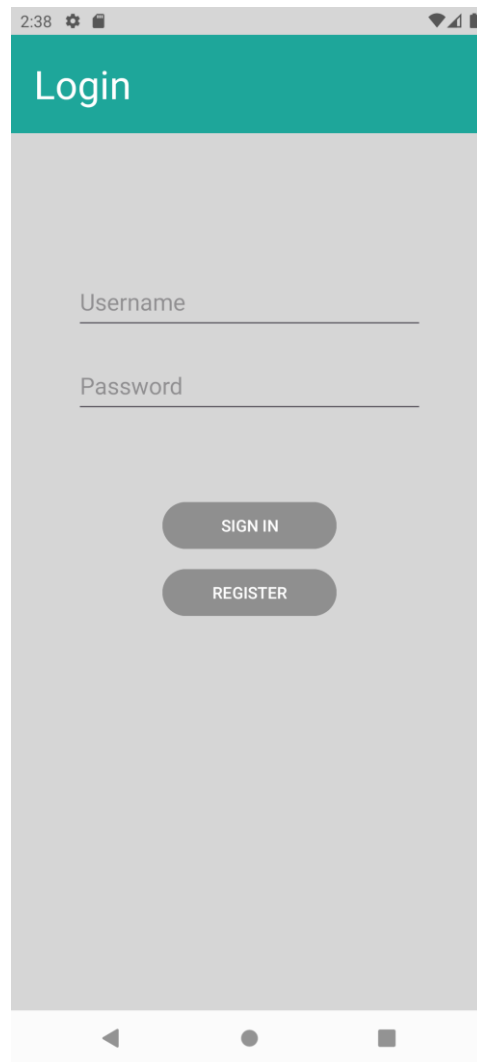
<EditText
    android:id="@+id/password"
    android:layout_width="300dp"
    android:layout_height="52dp"
    android:layout_marginTop="20dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="text"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/username" />

<Button
    android:id="@+id/button"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="70dp"
    android:backgroundTint="#8F8F8F"
    android:text="SIGN IN"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/password" />

<Button
    android:id="@+id/button3"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:backgroundTint="#8F8F8F"
    android:text="REGISTER"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

- Giao diện mình họa:



Giao diện đăng ký:

- Code:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#D6D6D6"
    tools:context=".register">

    <View
        android:id="@+id/view2"
        android:layout_width="412dp"
        android:layout_height="85dp"
        android:background="#1EA69A"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
```

```
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:text="Register"
        android:textColor="#FFFFFF"
        android:textSize="34sp"
        app:layout_constraintBottom_toBottomOf="@+id/view2"
        app:layout_constraintStart_toStartOf="@+id/view2"
        app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/username"
    android:layout_width="300dp"
    android:layout_height="52dp"
    android:layout_marginTop="120dp"
    android:ems="10"
    android:hint="Username"
    android:inputType="text"
    android:password="false"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/view2" />

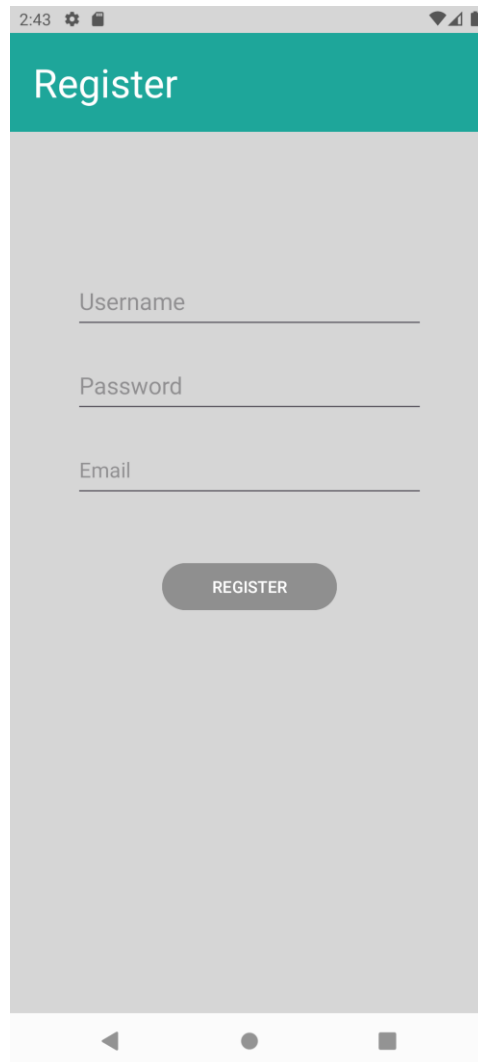
<EditText
    android:id="@+id/password"
    android:layout_width="300dp"
    android:layout_height="52dp"
    android:layout_marginTop="20dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="text"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/username" />

<Button
    android:id="@+id/button3"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:backgroundTint="#8F8F8F"
    android:text="REGISTER"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/email" />

<EditText
    android:id="@+id/email"
    android:layout_width="300dp"
    android:layout_height="52dp"
    android:layout_marginTop="20dp"
    android:ems="10"
    android:hint="Email"
    android:inputType="text"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/password" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Giao diện minh họa:



Giao diện hiển thị thông tin người dùng:

- Code:

```
- <?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#D6D6D6"
    tools:context=".HelloUser">

    <View
        android:id="@+id/view2"
        android:layout_width="412dp"
        android:layout_height="85dp"
        android:background="#1EA69A"
```

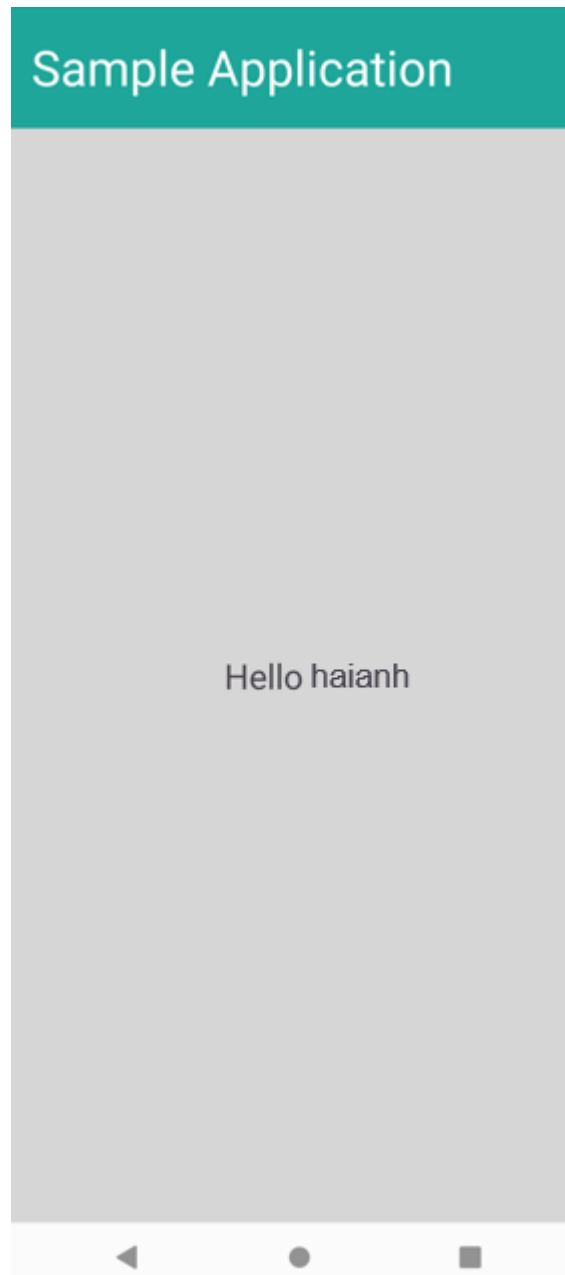
```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:text="Sample Application"
    android:textColor="#FFFFFF"
    android:textSize="34sp"
    app:layout_constraintBottom_toBottomOf="@+id/view2"
    app:layout_constraintStart_toStartOf="@+id/view2"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/view2" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

- Giao diện minh họa:



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.

3.1a Thêm thông tin gồm email, username và password vào cơ sở dữ liệu khi người dùng dùng chức năng Đăng ký

Để thêm thông tin người dùng vào cơ sở dữ liệu, ta tạo một đối tượng của lớp SQLiteConnector sử dụng để tương tác với cơ sở dữ liệu SQLite:

```
db = new SQLiteConnector( context: this);
```


Thêm các thông tin của người dùng cho 1 đối tượng thuộc class User, sau đó thêm User đó vào database thông qua phương thức addUser(u):

```
User u = new User();
u.setName(username);
u.setPassword(password);
u.setEmail(email);
db.addUser(u);
```

Đoạn code cụ thể như sau:

```
package com.example.myapplication;

import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.view.View;
import android.widget.Toast;
import android.content.Intent;
import android.util.Log;

public class register extends AppCompatActivity {
    private EditText etUsername, etPassword, etMail;
    private Button btnRegister;
    private SQLiteConnector db;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_register);

        etUsername = findViewById(R.id.username);
        etPassword = findViewById(R.id.password);
        etMail = findViewById(R.id.email);
        btnRegister = findViewById(R.id.button3);

        db = new SQLiteConnector(this);

        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String username = etUsername.getText().toString().trim();
                String password = etPassword.getText().toString().trim();
                String email = etMail.getText().toString().trim();

                if (username.isEmpty() || password.isEmpty() || email.isEmpty())
                {
                    Toast.makeText(register.this, "Please enter all the fields",
                    Toast.LENGTH_SHORT).show();
                } else {
```

```

        User u = new User();
        u.setName(username);
        u.setPassword(password);
        u.setEmail(email);
        db.addUser(u);
        Log.d("Password", HashPassword.hashPassword(password));
        Intent i = new Intent(register.this, MainActivity.class);
        startActivity(i);
    }
}

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;
});
}
}

```

3.1b. Truy vấn thông tin username và password cho chức năng đăng nhập

Trước tiên ta cũng tạo một đối tượng của lớp SQLiteConnector sử dụng để tương tác với cơ sở dữ liệu SQLite:

```
db = new SQLiteConnector(context, this);
```

Sau đó ta kiểm tra thông tin đăng nhập của người dùng bằng phương thức checkUser:

```

boolean isValid = db.checkUser(username, password);
if (isValid) {
    Intent i = new Intent(packageContext, MainActivity.this, HelloUser.class);
    i.putExtra(name, "username", username);
    startActivity(i);
} else {
    Log.d(tag, "Error", msg, "Loggin Fail");
}
}

```

Đoạn code cụ thể như sau:

```

package com.example.myapplication;

import android.os.Bundle;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

```

```

import android.widget.Toast;
import android.content.Intent;
import android.util.Log;

public class MainActivity extends AppCompatActivity {

    private EditText etUsername, etPassword;
    private Button btnLogin, btnRegister;
    private SQLiteConnector db;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        etUsername = findViewById(R.id.username);
        etPassword = findViewById(R.id.password);
        btnLogin = findViewById(R.id.button);
        btnRegister = findViewById(R.id.button3);

        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(MainActivity.this, register.class);
                startActivity(i);
            }
        });

        db = new SQLiteConnector(this);
        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String username = etUsername.getText().toString().trim();
                String password = etPassword.getText().toString().trim();

                if (username.isEmpty() || password.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please enter all the
fields", Toast.LENGTH_SHORT).show();
                } else {
                    boolean isValid = db.checkUser(username, password);
                    if (isValid) {
                        Intent i = new Intent(MainActivity.this,
HelloUser.class);

                        i.putExtra("username", username);
                        startActivity(i);
                    } else {
                        Log.d("Error", "Loggin Fail");
                    }
                }
            }
        });

        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;
        });
    }
}

```

Yêu cầu 4 - Điều chỉnh mã nguồn để password được lưu và kiểm tra dưới dạng mã hash thay vì plaintext. Trước tiên, ta thêm 1 class HashPassword vào project:

```
package com.example.myapplication;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class HashPassword {
    public static String hashPassword(String password) {
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-256");
            byte[] hash = md.digest(password.getBytes());
            StringBuilder hexString = new StringBuilder();
            for (byte b : hash) {
                String hex = Integer.toHexString(0xff & b);
                if (hex.length() == 1) hexString.append('0');
                hexString.append(hex);
            }
            return hexString.toString();
        } catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Trong class này, ta sử dụng SHA-256 làm hàm băm thông qua câu lệnh:

```
MessageDigest md = MessageDigest.getInstance("SHA-256");
```

Sau đó thực hiện băm mật khẩu dưới dạng mảng byte:

```
byte[] hash = md.digest(password.getBytes());
```

Cuối cùng ta chuyển đổi kết quả băm sang chuỗi hex:

```
StringBuilder hexString = new StringBuilder();
for (byte b : hash) {
    String hex = Integer.toHexString(0xff & b);
    if (hex.length() == 1) hexString.append('0');
    hexString.append(hex);
}
```

Điều chỉnh code tại addUser, tại nơi nhập giá trị password vào database ta gọi hàm hashPassword của class HashPassword để password nhập vào được lưu dưới dạng mã hash:

```

public void addUser(User user) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(COLUMN_USER_NAME, user.getName());
    values.put(COLUMN_USER_EMAIL, user.getEmail());
    values.put(COLUMN_USER_PASSWORD, HashPassword.hashPassword(user.getPassword()));

    // Inserting Row
    db.insert(TABLE_USER, nullColumnHack null, values);
    db.close();
}

```

Tương tự với updateUser và checkUser:

- updateUser:

```

public void updateUser(User user) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(COLUMN_USER_NAME, user.getName());
    values.put(COLUMN_USER_EMAIL, user.getEmail());
    values.put(COLUMN_USER_PASSWORD, HashPassword.hashPassword(user.getPassword()));

    // updating row
    db.update(TABLE_USER, values, whereClause: COLUMN_USER_ID + " = ?",
        new String[]{String.valueOf(user.getId())});
    db.close();
}

```

- checkUser:

```

public boolean checkUser(String username, String password) {

    // array of columns to fetch
    String[] columns = {
        COLUMN_USER_ID
    };
    SQLiteDatabase db = this.getReadableDatabase();
    // selection criteria
    String selection = COLUMN_USER_NAME + " = ?" + " AND " + COLUMN_USER_PASSWORD + " = ?";

    // selection arguments
    String[] selectionArgs = {username, HashPassword.hashPassword(password)};
}

```

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.

Để kết nối với Microsoft SQL server, ta thêm 2 permission sau:

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
```

Sau đó, ta tạo 1 class tên là ConSQL để tạo kết nối giữa ứng dụng với database:

```
package com.example.myapplication;
import android.annotation.SuppressLint;

import java.sql.Connection;
import java.sql.DriverManager;

import android.os.StrictMode;
import android.util.Log;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class ConSQL {
    Connection con;
    @SuppressLint("NewApi")
    public Connection conclass() {
        StrictMode.ThreadPolicy a = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(a);
        String ConURL=null;
        try {
            Class.forName("net.sourceforge.jtds.jdbc.Driver");
            ConURL =
            "jdbc:jtds:sqlserver://192.168.133.1:1433;databaseName=UserDatabase;user=haianh;
            password=13062013;";
            con= DriverManager.getConnection(ConURL);
        }
        catch (Exception e) {
            Log.e("Error", e.getMessage());
        }
        return con;
    }
}
```

Trong đoạn code trên chuỗi ConURL là chuỗi chứa thông tin cần thiết để thiết lập kết nối đến cơ sở dữ liệu SQL Server từ một ứng dụng Java bằng JDBC.

Chi tiết của chuỗi ConURL:

- jdbc:jtds:sqlserver://: Đây là tiền tố của URL kết nối JDBC để chỉ định loại cơ sở dữ liệu (trong trường hợp này là SQL Server) và sử dụng giao thức kết nối (trong trường hợp này là jtds).
- 192.168.133.1: Đây là địa chỉ IP máy chủ của cơ sở dữ liệu SQL Server.
- 1433: Đây là cổng mà máy chủ SQL Server lắng nghe yêu cầu kết nối.
- databaseName=UserDatabase: Đây là tên của cơ sở dữ liệu mà ta muốn kết nối đến.
- user=haianh: Đây là tên người dùng được sử dụng để xác thực kết nối đến cơ sở dữ liệu.

- password=13062013: Đây là mật khẩu được sử dụng để xác thực kết nối đến cơ sở dữ liệu.

Để tạo user mới trong màn hình đăng ký, trước hết ta khởi tạo kết nối đến cơ sở dữ liệu:

```
ConSQL c = new ConSQL();
con = c.conclass();
```

Tạo câu lệnh SQL để thêm 1 user vào database:

```
String sqlstatement = "INSERT INTO [User] (Username, Password, Email) VALUES ('" + username + "', '" + password + "', '" + email + "')";
```

Và thực thi câu lệnh:

```
Statement smt = con.createStatement();
smt.executeUpdate(sqlstatement);
```

Để kiểm tra user trong màn hình đăng nhập, ta cũng thực hiện các bước tương tự như trên, nhưng sẽ thay câu lệnh SQL trên thành câu lệnh SQL sau:

```
String sqlstatement = "SELECT ID FROM [User] WHERE Username = '" + username + "' AND Password = '" + password + "'";
```

Đoạn code chi tiết:

- Đăng ký:

```
- package com.example.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;

public class Register extends AppCompatActivity {
    private EditText etUsername, etPassword, etEmail;
    private Button btnRegister;

    Connection con;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_register);
        btnRegister = findViewById(R.id.button3);
```

```

        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                etUsername = findViewById(R.id.username);
                etPassword = findViewById(R.id.password);
                etEmail = findViewById(R.id.email);
                String username = etUsername.getText().toString().trim();
                String password = etPassword.getText().toString().trim();
                String email = etEmail.getText().toString().trim();
                if (username.isEmpty() || password.isEmpty()) {
                    Toast.makeText(Register.this, "Please enter all the
fields", Toast.LENGTH_SHORT).show();
                } else {
                    ConSQL c = new ConSQL();
                    con = c.conclass();
                    if (c != null) {
                        try {
                            String sqlstatement = "INSERT INTO [User]
(Username, Password, Email) VALUES ('" + username + "', '" + password +
"', '" + email + "')";
                            Statement smt = con.createStatement();
                            smt.executeUpdate(sqlstatement);
                            Intent i = new Intent(Register.this,
MainActivity.class);
                            startActivity(i);
                            con.close();
                        } catch (Exception e) {
                            Log.e("Error: ", e.getMessage());
                        }
                    }
                }
            }
        });
        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;
        });
    }
}

```

- Đăng nhập:

```

- package com.example.myapplication;

import android.os.Bundle;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.content.Intent;

```



```

import android.util.Log;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;

public class MainActivity extends AppCompatActivity {

    private EditText etUsername, etPassword;
    private Button btnLogin, btnRegister;
    Connection con;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        etUsername = findViewById(R.id.username);
        etPassword = findViewById(R.id.password);
        btnLogin = findViewById(R.id.button);
        btnRegister = findViewById(R.id.button3);

        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(MainActivity.this, Register.class);
                startActivity(i);
            }
        });

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String username = etUsername.getText().toString().trim();
                String password = etPassword.getText().toString().trim();

                if (username.isEmpty() || password.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please enter all
the fields", Toast.LENGTH_SHORT).show();
                } else {
                    ConSQL c = new ConSQL();
                    con = c.conclass();
                    if (c!=null){
                        try{
                            String sqlstatement = "SELECT ID FROM [User]
WHERE Username = '" + username + "' AND Password = '" + password + "'";
                            Statement smt = con.createStatement();
                            ResultSet set =
smt.executeQuery(sqlstatement);
                            if (set.next()) {
                                Intent i = new Intent(MainActivity.this,
HelloUser.class);

                                i.putExtra("username", username);
                                startActivity(i);
                            } else {
                                Toast.makeText(MainActivity.this,
"Incorrect username or password", Toast.LENGTH_SHORT).show();
                            }

                            con.close();
                        }
                    }
                }
            }
        });
    }
}

```

```

        catch (Exception e) {
            Log.e("Error: ", e.getMessage());
        }
    }
}

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;
});
}
}

```

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?

Bật ProGuard trong build.gradle.kts:

```

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

```

Viết các rule trong proguard-rules.pro:

```

-assertionerror class android.util.Log {
    public static *** d(...);
    public static *** v(...);
}
-keepattributes Signature
-optimizations code/allocation/variable

```

Ta nhận thấy có một số điểm khác nhau cơ bản giữa ứng dụng có sử dụng và không sử dụng ProGuard như sau:

- Không sử dụng ProGuard:
 - o Mã nguồn của ứng dụng sẽ được giữ nguyên vẹn trong file APK.
 - o Ứng dụng sẽ chiếm dung lượng lớn hơn do mã nguồn không bị thu gọn.
- Có sử dụng ProGuard:

- ProGuard sẽ thực hiện các tác vụ như: Loại bỏ các lớp, các phương thức không sử dụng, xóa các comments không cần thiết,...
- Ứng dụng sẽ có kích thước file APK nhỏ hơn

---HẾT---