Lab 13. Ứng dụng tải tập tin (4 tiết)

I. Yêu cầu

- Sinh viên sử dụng công cụ Android Studio để làm bài. Kết quả bài làm cần được chụp lại và giữ lại toàn bộ dự án để sử dụng khi cần thiết.

- Mỗi người làm trên một dự án khác nhau. Mỗi sinh viên chỉ làm trên một dự án

- Khi có yêu cầu, sinh viên nộp qua email giáo viên hoặc một kênh khác.

II. Luyện tập

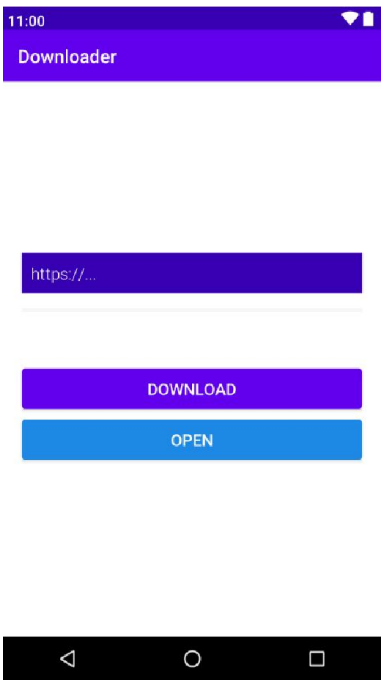
Xây dựng giao diện download file

Phạm vi kiến thức vận dụng

- Các kiến thức căn bản về Design UI
  - Drawable
  - Color Schemes
  - String constants
  - Style & themes
  - Layout constraint rules (padding, margin…)
- Các đối tượng View System
  - TextView, ImageView
    - Các ViewGroup gồm: LinearLayout, FrameLayout, TableRow

Nội dung bài thực hành

1. Yêu cầu: Tạo 1 ứng dụng Downloader và xây dựng giao diện layout sau



Mô tả: Đây là 1 giao diện hiển thị ô nhập link để download file, sau khi download thành công, có thể mở được file đó lên.

Yêu cầu giao diện:

- Có progressbar bên dưới ô nhập link download
- Thiết kế giao diện như hình mô tả.

2. Gợi ý: Các kiến thức sử dụng

- Tạo 1 layout xml trong thư mục res/layout
- Sử dụng các ViewSystem:
  - TextView, ProgressBar
- Sử dụng các ViewGroup:
  - LinearLayout hoặc FrameLayout

3. Hướng dẫn

Tạo file **res/layout/activity_main.xml** có nội dung sau:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_vertical"
    android:orientation="vertical"
    android:padding="20dp"
    tools:context=".MainActivity">
    <EditText
        android:id="@+id/edt_link"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/design_default_color_primary_dark"
        android:fontFamily="sans-serif-light"
        android:hint="@string/hint_link"
        android:padding="10dp"
        android:textColor="@color/white"
        android:textColorHint="@color/white"
        android:textSize="18sp" />
    <ProgressBar
        android:id="@+id/progress_bar"
        style="@style/Widget.AppCompat.ProgressBar.Horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:backgroundTint="@color/design_default_color_primary_dark"
        android:progressTint="@color/design_default_color_primary_dark" />
    <Button
        android:id="@+id/bt_download"
        android:layout_marginTop="50dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="10dp"
        android:text="@string/txt_download"
        android:textSize="18sp" />
    <Button
        android:id="@+id/bt_open"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:backgroundTint="@color/blue"
        android:padding="10dp"
        android:text="@string/txt_open"
        android:textSize="18sp" />
</LinearLayout>
```

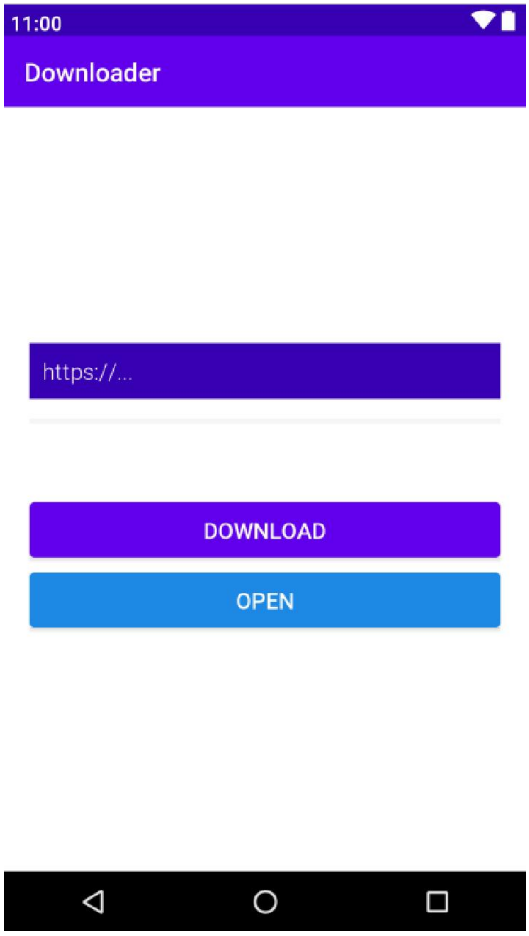
Thực hiện xử lý download file và lưu trong DataStorage

Phạm vi kiến thức vận dụng

- Tiếp nối Lab 7.1 - Xây dựng giao diện download file
- Các kiến thức về logic
    - Activity
    - OnClickListener
    - Intent
    - User permission
    - Thread
    - Provider

Nội dung bài thực hành

Yêu cầu: Tạo 1 ứng dụng Downloader và xây dựng giao diện layout sau



Mô tả: Đây là 1 giao diện hiển thị ô nhập link để download file, sau khi download thành công, có thể mở được file đó lên.

Yêu cầu giao diện:

- Có progressbar bên dưới ô nhập link download
- Thiết kế giao diện như hình mô tả.

2. Gợi ý: Các kiến thức sử dụng

- Tạo 1 layout xml trong thư mục res/layout
- Sử dụng các ViewSystem:
    - TextView, ProgressBar
- Sử dụng các ViewGroup:

LinearLayout hoặc FrameLayout

Hướng dẫn triển khai code logic trong activity

Bước 1: Viết mã cho file **AndroidManifest.xml**

- Khai báo quyền người dùng trong Android Manifest
  - Đọc ghi bộ nhớ
  - Truy cập internet để download File
- Cung cấp quyền cho việc đọc file từ bộ nhớ **external**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.techja.downloader">
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE"
        tools:ignore="ScopedStorage" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.Downloader"
        tools:ignore="AllowBackup">
        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="${applicationId}.provider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/provider_paths" />
        </provider>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Bước 2: Tạo file **res/xml/provider_paths.xml** để cho phép truy cập vùng nhớ **external** của ứng dụng.

```
<?xml version="1.0" encoding="utf-8"?>
<paths>
    <external-path name="external_files" path="."/>
</paths>
```

Bước 3: Xử lý logic trong file MainActivity như sau:

- Ánh xạ các View trong layout ra môi trường code
- Yêu cầu quyền người dùng để có thể truy cập bộ nhớ để đọc/ghi file
- Xử lý click vào các button download, open để tải file, mở file.
- Sử dụng Thread để việc truy cập và download file tối ưu hơn.
- Sử dụng Intent để mở file sau khi tải.

```
package com.techja.downloader;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.FileProvider;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.net.URL;
import java.net.URLConnection;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
   private EditText edtLink;
   private ProgressBar progressBar;
   private int size = 0;
   private Button btOpen;
   private String savePath;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      initViews();
   }
   private void initViews() {
      progressBar = findViewById(R.id.progress_bar);
      edtLink = findViewById(R.id.edt_link);
      findViewById(R.id.bt_download).setOnClickListener(this);
      btOpen = findViewById(R.id.bt_open);
      btOpen.setEnable(false);
      btOpen.setOnClickListener(this);
   }
```

```
    @Override
    public void onClick(View v) {
        if (!checkPermission()) return;

        if (v.getId() == R.id.bt_download) {
            downloadFile(edtLink.getText().toString());
        } else if (v.getId() == R.id.bt_open) {
            openFile();
        }
    }
    private boolean checkPermission() {
        boolean isAllow = checkSelfPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED;
        if (!isAllow) {
            requestPermissions(new String[]{
                    Manifest.permission.READ_EXTERNAL_STORAGE,
                    Manifest.permission.WRITE_EXTERNAL_STORAGE
            }, 101);
        }
        return isAllow;
    }
    private void openFile() {
        if (savePath == null || savePath.isEmpty()) return;
        File file = new File(savePath);
        Uri uri = FileProvider.getUriForFile(this, getPackageName() + ".provider", file);
        String mime = getContentResolver().getType(uri);

        Intent intent = new Intent();
        intent.setAction(Intent.ACTION_VIEW);
        intent.setDataAndType(uri, mime);
        intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
        startActivity(intent);
    }
    private void downloadFile(String link) {
        new Thread() {
            @Override
            public void run() {
                try {
                    URLConnection conn = new URL(link).openConnection();
                    InputStream in = conn.getInputStream();
                    savePath = getExternalFilesDir(null).getPath() + "/" + new File(link).getName();
                    FileOutputStream out = new FileOutputStream(new File(savePath));
                    byte[] buff = new byte[1024];
                    int len = in.read(buff);
                    runOnUiThread(() -> {
                        progressBar.setMax(conn.getContentLength());
                        progressBar.setProgress(0);
                        btOpen.setEnabled(false);
                    });
                    size = 0;
                    while (len > 0) {
```

```
                out.write(buff, 0, len);
                size += len;
                runOnUiThread(() -> progressBar.setProgress(size));
                len = in.read(buff);
            }
            out.close();
            in.close();
            runOnUiThread(() -> btOpen.setEnabled(true));
        } catch (Exception e) {
            e.printStackTrace();
            runOnUiThread(() -> Toast.makeText(MainActivity.this, e.getMessage(),
Toast.LENGTH_SHORT).show());
        }
      }
    }.start();
  }
}
```

4. Lưu ý:

- Phương thức kiểm tra quyền người dùng xem đã được cấp hay chưa:

```
private boolean checkPermission() {
    boolean isAllow = checkSelfPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED;
    if (!isAllow) {
      requestPermissions(new String[]{
          Manifest.permission.READ_EXTERNAL_STORAGE,
          Manifest.permission.WRITE_EXTERNAL_STORAGE
      }, 101);
    }
    return isAllow;
  }
```

- Phương thức để download file

```
private void downloadFile(String link) {
  new Thread() {
    @Override
    public void run() {
      try {
        URLConnection conn = new URL(link).openConnection();
        InputStream in = conn.getInputStream();
        savePath = getExternalFilesDir(null).getPath() + "/" + new File(link).getName();
        FileOutputStream out = new FileOutputStream(new File(savePath));
        byte[] buff = new byte[1024];
        int len = in.read(buff);
        runOnUiThread(() -> {
          progressBar.setMax(conn.getContentLength());
          progressBar.setProgress(0);
          btOpen.setEnabled(false);
        });
        size = 0;
```

```
            while (len > 0) {
                out.write(buff, 0, len);
                size += len;
                runOnUiThread(() -> progressBar.setProgress(size));
                len = in.read(buff);
            }
            out.close();
            in.close();
            runOnUiThread(() -> btOpen.setEnabled(true));
        } catch (Exception e) {
            e.printStackTrace();
            runOnUiThread(() -> Toast.makeText(MainActivity.this, e.getMessage(),
Toast.LENGTH_SHORT).show());
        }
    }
  }.start();
}
```

- Phương thức để mở file sau khi tải

```
private void openFile() {
  if (savePath == null || savePath.isEmpty()) return;
  File file = new File(savePath);
  Uri uri = FileProvider.getUriForFile(this, getPackageName() + ".provider", file);
  String mime = getContentResolver().getType(uri);

  Intent intent = new Intent();
  intent.setAction(Intent.ACTION_VIEW);
  intent.setDataAndType(uri, mime);
  intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
  startActivity(intent);
}
```

III. Bài tập

Sinh viên đọc, hiểu code và comment lại các đoạn code tương ứng.

--Hết--