

SAE 302 : Application Android, TCP/UDP

Par : Théo DUFOUR

Réseaux & Télécommunication

Semestre 3 - Janvier 2023

Table des matières

1 Introduction	3
1.1 Enoncé du problème	3
1.2 La solution	4
2 Frontend	4
2.1 Design	4
2.2 Établir une connexion TCP / ICMP / UDP / FTP	6
2.3 Gestion des permissions	7
3 UI	9
3.1 Design	9
3.2 Ecran principal	12
3.3 Les différentes activités	14
4 FTP	19
4.1 Liste des fichiers	19
4.2 Téléchargements	20
5 Conclusion	22
6 Documentation	23
7 Annexe	24

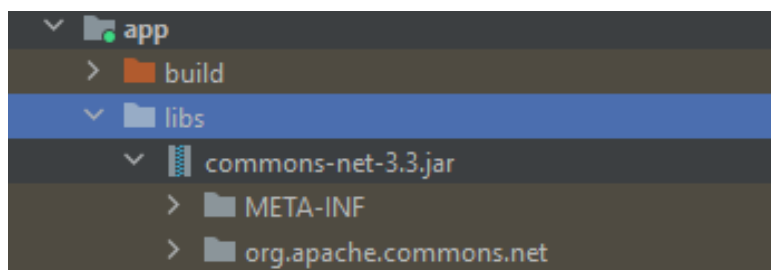
1 Introduction

1.1 Enoncé du problème

Ce projet vise à concevoir et implémenter une application utilisant le protocole TCP/IP, UDP sur mobile pour un système d'exploitation Android qui accompagne également un serveur ftp hackcesss.org . L'application utilise la bibliothèque Java .net d'Apache Commons pour effectuer trois fonctions :

- Se connecter à distance à un serveur FTP
- Parcourir les listes de répertoires et télécharger

Cette Application est réservée à un usage personnel et ne sera pas publiée sur le Google Play Store.



Capture 1 : Librairie commons-net-3.3.jar

1.2 Solution

Pour résoudre ce problème, une application Android doit être créée pour agir comme le client mobile ou serveur, transférer des fichiers vers l'appareil ou en recevoir. L'application doit être créée à l'aide d'Android Studio car elle utilise Java, qui est largement utilisé et facile à trouver (notamment pour les nombreuses bibliothèques), mon téléphone fonctionne sous Android, je vais donc pouvoir faire les tests directement dessus. En outre, l'application doit utiliser l'un des protocoles de communications (TCP/UDP) inclus dans le système d'exploitation du serveur et client. Etant donné que cette application sera utilisée sur les réseaux publics, la sécurité est aussi une préoccupation.

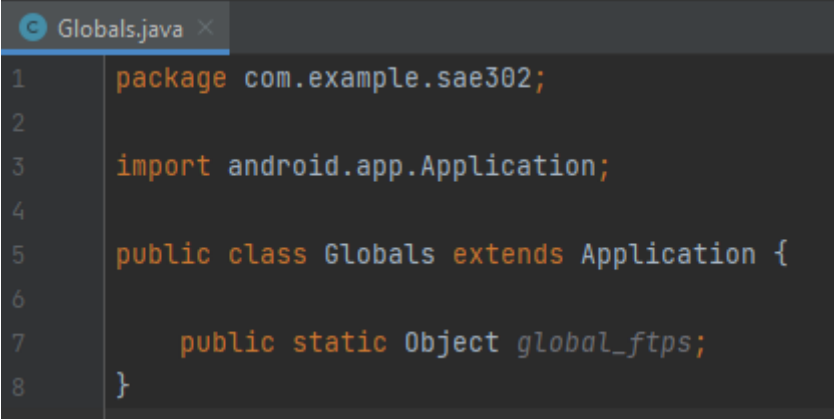
2 Frontend

2.1 Design

J'ai décidé d'utiliser FTPS comme solution réseau pour mon FTP ainsi que TCP en raison de sa simplicité et sécurité par rapport à UDP. La simplicité vient du jeu de commandes de base que FTP utilise pour établir une connexion, naviguer dans les répertoires et gérer les transferts. Cependant, le ftp pur manque de tout, notamment de sécurité dans sa connexion, en transmettant des informations de connexion sensibles en clair et laissant les données non cryptées, permettant ainsi à quiconque entre le client et le serveur de facilement s'immiscer (type : attaque [MITM](#)). Je vais donc résoudre ce problème avec un simple ajout de la sécurité TLS / SSL, ceci utilise une combinaison de chiffrement à clé publique/privée avec un certificat

signé pour permettre une connexion sécurisée entre deux appareils. Après avoir choisis le protocole de transfert sécurisé à utiliser, il faut aussi la bonne bibliothèque. J'ai choisi Apache Commons .net en raison de leur documentation approfondie. L'utilisation des bibliothèques Apache Commons tourner autour d'un seul objet FTPS qui maintient le statut, la connexion, les paramètres, etc...

Le point négatif est que la programmation empêche cet objet commun d'être transmis entre les écrans de l'interface utilisateur (activité), donc je choisis d'utiliser une référence globale pour s'assurer que l'objet resterait statique et commun à travers tout le code Android.



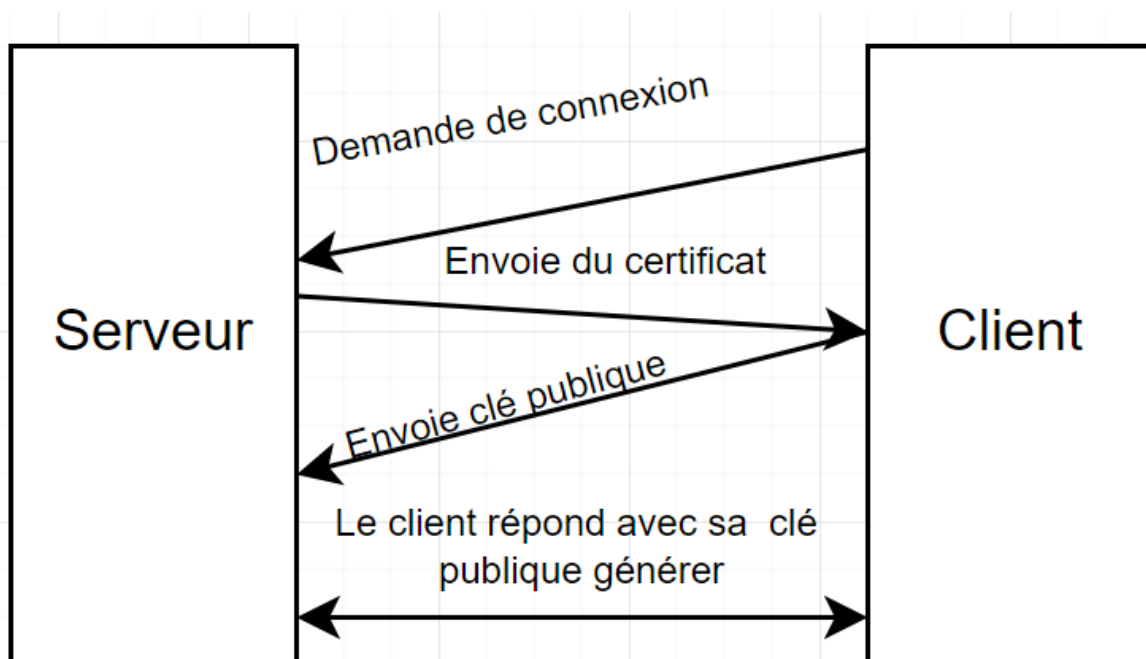
```
1 package com.example.sae302;
2
3 import android.app.Application;
4
5 public class Globals extends Application {
6
7     public static Object global_ftps;
8 }
```

Capture 2 : Référence variables globals en Java

2.2 Établir une connexion TCP / ICMP / UDP / FTP

La connexion à un client ou un serveur nécessite un ensemble d'informations comprenant le nom d'hôte ou l'adresse IP du serveur, un nom d'utilisateur et un mot de passe par exemple pour une connexion FTP. Cela implique généralement l'échange de protocole tcp, icmp, udp. Pour ce qui est du FTP l'échange s'effectue avec un certificat qui doit être validé en externe par un tiers, ou être accepté comme valide par l'utilisateur.

Malheureusement, en raison des limitations de la bibliothèque choisie, cette vérification de certificat n'est en aucun cas effectué, il faut donc être prudent lors de la connexion à des serveurs que nous ne sommes pas propriétaires.



Capture 3 : Processus de configuration d'une connexion sécurisé (FTP)

2.3 Gestion des permissions

Il est indiqué dans le sujet que le serveur doit gérer plusieurs clients en parallèle (multithreading) ainsi que gérer les droits d'accès entre les différents utilisateurs. Par exemple si une application doit envoyer des messages SMS, elle doit inclure cette permission dans le fichier manifeste :

```
<manifest xmlns:android = "http://schemas.android.com/apk/res/android"
  package = "com.example.sae302" >
  <uses-permission android: name = "android.permission.SEND_SMS" />
  <application ... >
  ...
  </ application>
</ manifest>
```

Capture 4 : Exemple permission

Si l'autorisation demandée dans le fichier manifeste ne présente pas un risque élevé pour la confidentialité de l'utilisateur ou le fonctionnement du périphérique, le système accorde automatiquement cette autorisation à votre application. Si par contre l'autorisation demandée représente un risque, affecte la confidentialité de l'utilisateur ou le fonctionnement normal du périphérique, tel qu'envoyer un SMS par exemple, l'utilisateur doit explicitement accorder cette autorisation. Dans notre cas nous n'avons pas besoin de permissions du type confidentiel, nous n'aurons besoin que du réseau de l'utilisateur.

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

Capture 5 : Les permissions dans le fichier AndroidManifest.xml

Lorsque vous travaillez sur une application qui souhaite accéder à Internet, vous devez ajouter une autorisation supplémentaire dans androidmanifest.xml, sinon votre application ne pourra pas accéder à Internet. Vous devez donc mettre le code ci-dessous :

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

Dans le fichier AndroidManifest.xml pour accéder à Internet dans votre application.

Pour vérifier l'état général du réseau et surveiller les connexions réseau en général on utilise cette permission :

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

Pour gérer tous les aspects de la connectivité Wi-Fi :

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

Et enfin pour la dernière permission :

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

Cette autorisation « Lire l'état et l'identité du téléphone » peut être utilisée pour effectuer une gamme de tâches, des plus simples aux plus invasives :

- désactiver le son d'une application lorsqu'un appel arrive
- vérification de l'utilisateur/téléphone avec les informations IMEI.

Ceux-ci sont des exemples et ne seront pas forcément utilisés dans mon application.

3 UI

3.1 Design

Android étant au cœur du développement de cette application, son utilisation généralisée et son accessibilité signifie qu'un monde entier de documentation, de tutoriels et de conseils utiles existe sur internet pour les personnes qui débutent avec Android. La programmation sous Android se fait entièrement en format Java ou plus récemment depuis 2011 en Kotlin, dans notre cas nous devons réaliser l'application sous Java qui sera associé également à un fichier xml et nécessite l'utilisation d'un développement intégré, grâce à l'IDE (environnement de développement intégré)

Android Studio on peut gérer la structuration, la compilation et l'exécution et le débogage de chaque projet.

Il existe deux principaux IDE à prendre en compte lors du démarrage du développement Android :

Eclipse ou Android Studio. J'ai choisi d'utiliser les deux, mais l'application Android sera codée uniquement sous Android Studio car il est indiqué dans le sujet de la SAE 302. Une fois l'espace de travail préparé, la création d'un projet aboutit à un modèle générique qui contient tous les éléments requis ressources nécessaires pour exécuter une application de base. Cela inclut le code de démarrage requis,

Android bibliothèques, ressources, y compris les fichiers d'image et la mise en page et également un manifest contenant des informations importantes sur l'utilisation et la structure de l'application. La programmation d'Android est centrée sur l'utilisation d'activités, qui divisent les fonctionnalités de l'application dans différentes activités (écran) vues par l'utilisateur. Chaque activité sert son propre objectif, qu'il s'agisse de collecter des informations via des zones de texte, d'afficher des éléments dans une liste déroulante, ou simplement afficher l'écran de chargement (SplashScreenActivity) pendant le traitement des informations. Ces activités consistent en un corps de code contenant la méthode onCreate() par défaut, qui est appelé lors du démarrage ou de la transition vers une autre activité, et un fichier de mise en page au format xml détaillant l'apparence et la fonction de l'interface utilisateur. Lorsqu'une activité est lancée, elle est passée en mode Intent contenant des informations contextuelles de l'activité précédente.

```
//CHANGER D'ACTIVITE AVEC UN CLICK SUR LE BOUTON FTP OU PING ACTIVITY OU CLIENT / SERVER
ftp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent ftp = new Intent( packageContext MainActivity.this, FtpActivity.class);
        startActivity(ftp);
    }
});
ping.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent ping = new Intent( packageContext MainActivity.this,PingActivity.class);
        startActivity(ping);
    }
});
tcpudp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent tcpudp = new Intent( packageContext MainActivity.this,TcpUdpActivity.class);
        startActivity(tcpudp);
    }
});
```

Capture 6 : Gestion des activités via le fichier MainActivity.java

La gestion du projet se trouve dans le fichier AndroidManifest.xml, qui contient une liste de toutes les activités et leurs filtres d'utilisation, des informations requis par Google Play Store, tels que l'identification de la version et le minimum/cible. Version Android et autorisation spéciales requises pour des choses comme l'écriture en mémoire et connexion à internet.

```
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="true"
    android:theme="@style/Theme.SAE302"
    tools:targetApi="31">
    <activity
        android:name=".ServerActivity"
        android:parentActivityName=".MainActivity"
        android:exported="false">
        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
    <activity
        android:name=".TcpUdpActivity"
        android:parentActivityName=".MainActivity"
        android:exported="false">
        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
    <activity
        android:name=".SplashScreenActivity"
        android:exported="true">
        <intent-filter>
```

Capture 7 : AndroidManifest.xml

Ainsi j'ai créé six activités pour gérer la partie client TCP, serveur TCP, effectué des PING, serveur FTP, ainsi que l'écran de démarrage dans des écrans indépendants. La faite de diviser en six catégories signifie également simplifier la conception en permettant à chaque mise en page d'être relativement statique sans difficulté supplémentaire ou d'inclure des fenêtres de pop-up ou animation complexes, n'étant pas évalué dans le design de l'application je vais donc faire au mieux pour qu'elle soit user friendly.

3.2 Ecran Principal

La première activité qui s'ouvre par défaut au premier lancement de l'application est la SplashScreenActivity.java qui n'est autre qu'un écran de chargement avant l'écran principale puis quatre boutons apparaissent. Une fois que l'utilisateur a fait son choix et appuyé sur un des quatre boutons un rappel de la fonction associée à l'action du clic sur le bouton commence à exécuter le code de redirection vers une autre activité voire capture six pour le code.



Capture 8 : SplashScreenActivity.java



Capture 9 : Ecran Principale

3.3 Les différentes activités

Lorsque vous cliquez sur un des quatre boutons, il va donc vous renvoyer dans une des quatre activités détaillées au-dessus. J'ai mis en premier l'accès FTP puis EFFECTUER UN PING et TCP CLIENT enfin TCP SERVEUR. Pour l'accès FTP on se retrouve avec cette interface :



Capture 10 : Activité FTP

On a donc trois zones de texte pour entrer l'adresse du serveur, le nom d'utilisateur et son mot de passe associé, ainsi qu'un bouton « connexion » pour confirmer que l'utilisateur a fini de saisir ses informations. Ces zones de texte sont des vues de texte modifiables permettent au texte par défaut de montrer à l'utilisateur avec les informations qu'il doit entrer et où.

```

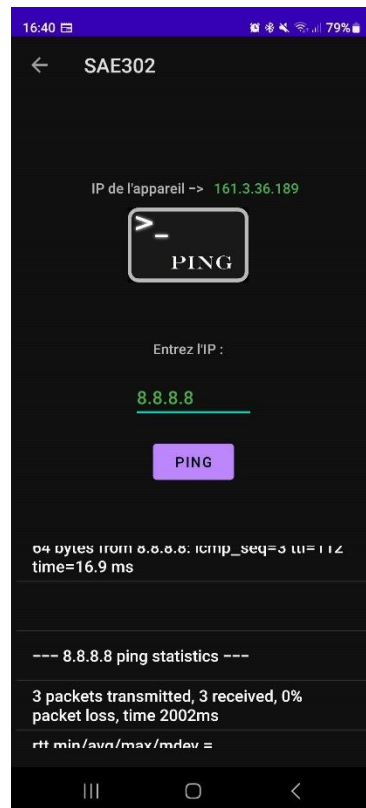
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SAE 302 - THEO DUFOUR"
    android:textColor="#4CAF50"/>
<EditText
    android:id="@+id/server"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="IP SERVEUR FTP"
    android:layout_marginTop="240dp"
    android:layout_marginEnd="10dp"
    android:layout_marginBottom="2dp"
    android:inputType="text" />
<EditText
    android:id="@+id/username"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="NOM D'UTILISATEUR"
    android:inputType="text" />
<EditText
    android:id="@+id/password"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="MOT DE PASSE"
    android:inputType="textPassword" />
<Button
    android:id="@+id/myButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#5E35B2"

```

Capture 11 : Fichier XML associé à l'activité FTP

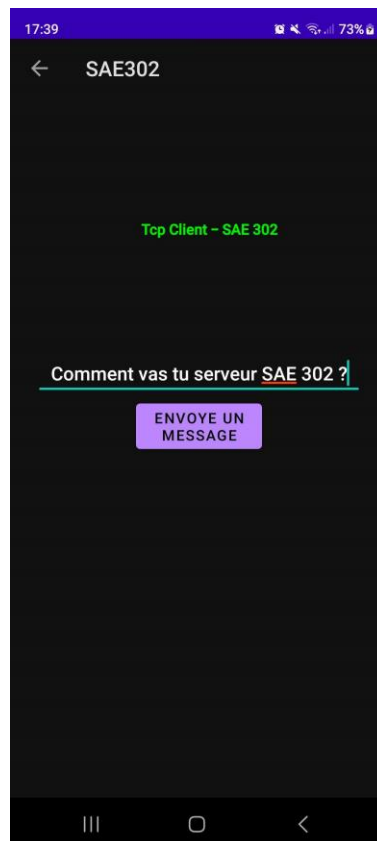
Une fois que l'utilisateur a terminé et appuyé sur le bouton « connexion », un rappel de la fonction associée à l'action de click sur le bouton commence à exécuter le code FTPS pour connectez le client au serveur FTP. Si la connexion est acceptée l'objet FTPS est enregistré à un état global et une autre activité est démarrée, sinon l'application reste sur le même écran (même activité) et invite l'utilisateur à ressaisir ses informations.

En deuxième bouton nous avons « effectué un ping » celle-ci renvoie l'utilisateur vers une nouvelle activité où il peut renseigner une adresse IP avec un texte modifiable, pour réaliser des tests de connectivité, également j'ai rajouté la possibilité de voir l'Adresse IP de notre appareil



Capture 12 : Effectuer un ping

Puis nous avons un troisième bouton qui renvoie vers un client TCP, avec un texte pour envoyer sur un serveur que j'ai créé via l'IDE Eclipse et qui reçoit les messages émis depuis l'Application.

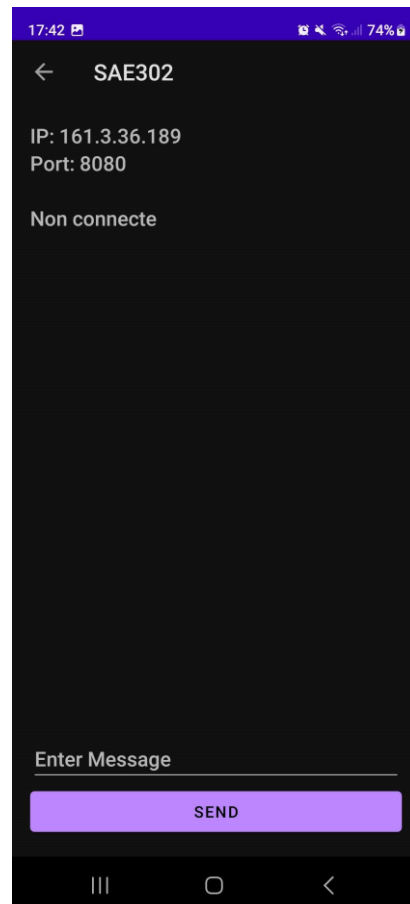


Capture 13 : TCP Client

```
Problems Javadoc Declaration Console Coverage
Server [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (11
Server started. Listening to the port 4444
hello
Comment vas tu serveur SAE 302 ?
```

Capture 14 : Réception Message sur Eclipse

Et pour finir la dernière activité c'est tout simplement un serveur TCP, il nous affiche L'IP ainsi que le port pour qu'un client puisse instancier une connexion sur le serveur Android.

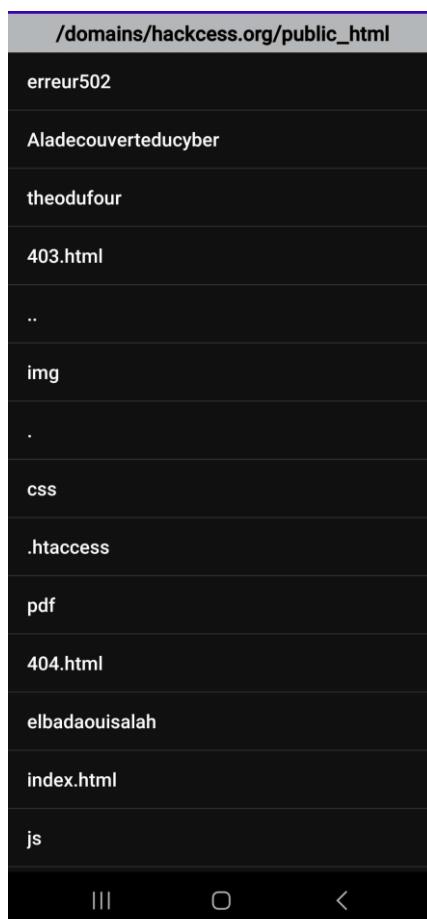


Capture 15 : Serveur TCP

4 FTP

4.1 Liste des fichiers

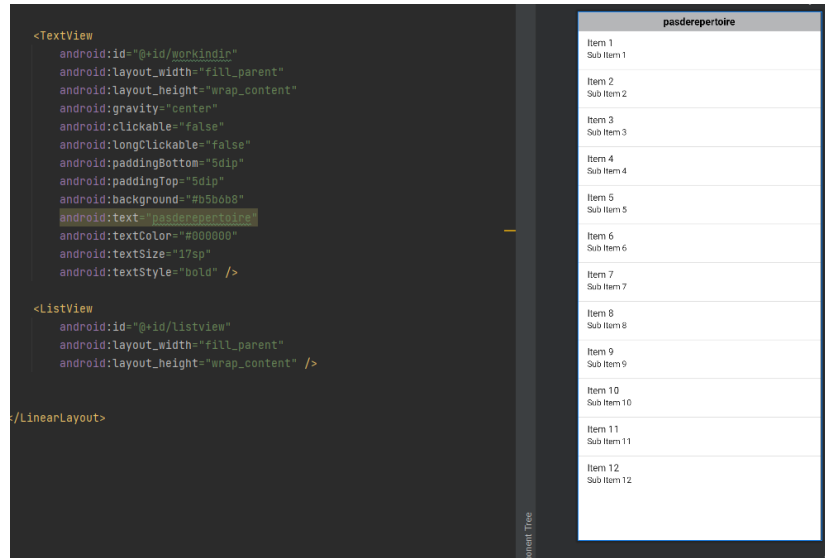
Lorsque vous vous connectez avec l'activité FTP la liste des fichiers de votre serveur FTP apparaît :



Capture 16 : Liste Fichier FTP Hackcess

J'ai mis le chemin du répertoire en haut de l'écran comme en-tête de la liste afin qu'elle soit facilement visible et toujours présente. Ceci est fait en utilisant une vue de texte standard dont le texte est défini dynamiquement à chaque lancement de l'activité. Le contenu du répertoire est présenté dans une vue de liste déroulante, où chaque élément peut être soit pressé appui régulier ou long selon ce que vous souhaitez comme par exemple changer de répertoire.

La vue de liste est dynamiquement remplie par un tableau de chaînes fourni par la méthode LIST à l'aide d'un adaptateur de tableau qui articule chaque donnée en tableau avec son propre élément de liste cliquable.



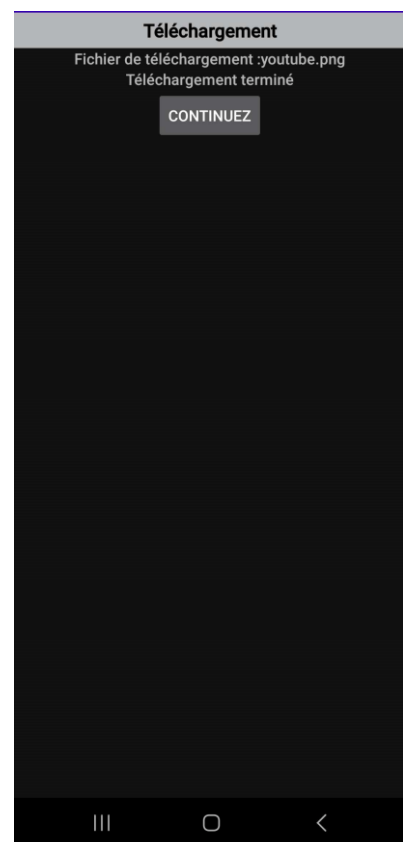
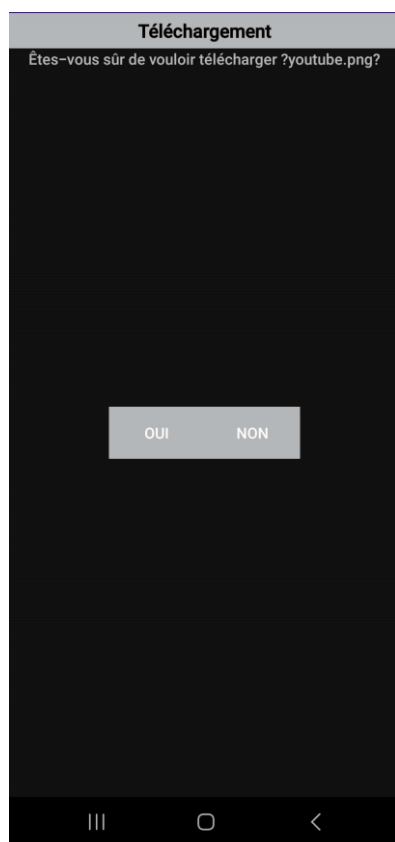
Capture 17 : Code XML Liste Fichier.

Lorsque l'utilisateur clique sur un dossier, le changement de répertoires est envoyé à l'activité Fichiers qui est relancée afin que l'écran puisse être mis à jour avec le nouveau contenu de ce répertoire. Une fois que l'utilisateur a trouvé le fichier ou dossier qu'ils veulent télécharger, un appui long provoque le déclenchement de l'activité de Téléchargement d'un fichier ou dossier.

4.2 Téléchargements

Après le lancement de l'activité de téléchargement, le nom du fichier ou du dossier choisi s'affiche dans une question demandant si l'utilisateur est sûr de vouloir commencer le téléchargement de son fichier ou dossier choisit au préalable. J'ai fait cela pour empêcher tout transfert accidentel de clics erronés ou involontaires. Les deux boutons marqués oui

et non permettent à l'utilisateur de continuer et démarrer le transfert ou de revenir à parcourir les fichiers ce qui relance l'activité Fichiers. En appuyant sur le bouton oui lance les transferts et supprime le texte et les boutons de confirmation, en les remplaçant par des confirmations textuelles à la fin du téléchargement de chaque élément indiquant également la progression de l'utilisateur si plusieurs transferts sont à effectué.



Capture 18 & 19 : Activité Téléchargement.

5 Conclusion

Ce projet m'a donné l'occasion d'en savoir plus sur Android et comment coder une application dédiée au système d'exploitation Android. J'ai l'impression d'avoir acquis une meilleure compréhension de la programmation Java. Devoir apprendre la programmation Android quasiment à partir de zéro était parfois difficile plus d'heure de TP aurait pu être nécessaire mais je sens que je suis devenu un « programmeur » meilleure et plus complet pour mes efforts.

L'application dans son état actuel est plus un prototype fonctionnel qu'une version final. Il fait tout ce qu'il était prévu de faire sur le sujet, mais pas dans le meilleure ou du moins pas de manière la plus efficace. Actuellement, toutes les fonctions de mise en réseau s'exécutent dans le thread principal qui peut empêcher l'interface utilisateur de se mettre à jour ou de répondre aux entrées de l'utilisateur.

Également beaucoup plus de temps aurait pu être consacré à rendre l'interface utilisateur plus attrayante, mais l'application remplit son objectif je suis satisfait de ce que j'ai accompli dans le temps impartit.

Temps passé environ : 26 heures.

6 Documentation

1. *Eclipse*

Disponible sur : <https://eclipse.org/>

2. *Android Studio*

Disponible sur : <https://developer.android.com/studio>

3. GitHub permissions Android

Disponible sur : <https://github.com/316k/IFT1155-Permissions>

4. Guide permission Android

Guide : <https://developer.android.com/guide/topics/permissions/overview>

5. Apache Commons Net 3.9.0

Disponible sur : [Apache Commons Net 3.9.0](#)

6. Pourquoi ActionBarActivity est obsolète

Disponible sur : [stackoverflow](#)

7. Pourquoi ai-je besoin de prolonger une activité

Disponible sur : [stackoverflow](#)

8. Probleme rencontrer lors du code

Disponible sur : [stackoverflow](#)

7 Annexe

SplashScreenActivity.java

```
//IMPORT
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
//IMPORT
public class SplashScreenActivity extends AppCompatActivity {

    private final int splashscreen = 2000; //TEMPS CHARGEMENT

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);

        //Rediriger vers la page principale "MainActivity" après 2 secondes
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                //demarrer une page
                Intent intent = new
Intent(getApplicationContext(),MainActivity.class);
                startActivity(intent);
                finish();
            }
        }, splashscreen);

    }
}
```

MainActivity.Java

```
package com.example.sae302;
////// IMPORT ///
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.view.View;

import android.widget.Button;

import android.os.Bundle;
//////      ////

public class MainActivity extends AppCompatActivity {
```



```

//VARIABLE ICI :

Button ftp,ping,tcpudp,server; // Les différents boutons .

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); //Le fichier est associé à
l'écran Activity_Main.xml

    ping = findViewById(R.id.ping); //Pas besoin de préciser ping =
(Button) findViewById(R.id.ping)
    ftp = findViewById(R.id.ftp); //Pas besoin de préciser ftp =
(Button) findViewById(R.id.ftp)
    tcpudp = findViewById(R.id.tcpudp); //Pas besoin de préciser
tcpudp = (Button) findViewById(R.id.tcpudp)
    server = findViewById(R.id.server); //Pas besoin de préciser
server = (Button) findViewById(R.id.server)

    ///
    //CHANGER D'ACTIVITE AVEC UN CLICK SUR LE BOUTON FTP OU PING
ACTIVITY OU CLIENT / SERVER
    ftp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent ftp = new Intent(MainActivity.this,
FtpActivity.class);
            startActivity(ftp);
        }
    });
    //LANCEMENT DE L'ACTIVITE Ping Activité au click
    ping.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent ping = new
Intent(MainActivity.this,PingActivity.class);
            startActivity(ping);
        }
    });
    //LANCEMENT DE L'ACTIVITE TcpUdpActivity (Client FTP) au click
    tcpudp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent tcpudp = new
Intent(MainActivity.this,TcpUdpActivity.class);
            startActivity(tcpudp);
        }
    });
    //LANCEMENT DE L'ACTIVITE ServerActivity (Serveur FTP) au click
    server.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View view) {
            Intent server = new
Intent(MainActivity.this, ServerActivity.class);
            startActivity(server);

        }
    });
}
}
}

```

FtpActivity.Java

```

package com.example.sae302;
//IMPORT ///
import java.io.IOException;
import java.io.PrintWriter;
import org.apache.commons.net.PrintCommandListener;
import org.apache.commons.net.ftp.FTP;
import org.apache.commons.net.ftp.FTPConnectionClosedException;
import org.apache.commons.net.ftp.FTPReply;
import org.apache.commons.net.ftp.FTPSClient;
import android.content.Intent;
import android.os.Bundle;
import android.os.Environment;
import android.os.StrictMode;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

////////////////////////////////////

public class FtpActivity extends AppCompatActivity implements
OnClickListener {
    private EditText username, mdp, serveur; /// EDIT TEXT (Nom utilisateur
, mot de passe , ip serveur)
    private Button Connexion; //Bouton connexion
    boolean error = false;
    String protocol = "SSL";
    FTPSClient ftps;
    public static final String SAE_302 = "SAE302";
    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ftp);

    StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);
    ftps = new FTPSClient(protocol);
    ftps.addProtocolCommandListener(new PrintCommandListener(new
        PrintWriter(System.out)));

    //VARIABLES ICI :

    serveur = findViewById(R.id.server); // Pas besoin de preciser
par exemple server= (EditText) car preciser au dessus .
    username = findViewById(R.id.username); // Pas besoin de preciser
par exemple username = (EditText) car preciser au dessus .
    mdp = findViewById(R.id.password); // Pas besoin de preciser par
exemple password = (EditText) car preciser au dessus .
    Connexion = findViewById(R.id.myButton); // Pas besoin de preciser
par exemple theButton = (Button) car preciser au dessus .
    Connexion.setOnClickListener(this);
}

public void onClick(View v) {
    //respond to click
    if (v.getId() == Connexion.getId()) {
        // init connection
        try
        {
            int reply;
            ftps.connect(serveur.getText().toString());
            // System.out.println("Connecté à " +
server.getText().toString() + ".");
            Toast.makeText(this,"Connecté à " +
serveur.getText().toString() + ". ",Toast.LENGTH_SHORT).show();
            reply = ftps.getReplyCode();

            if (!FTPReply.isPositiveCompletion(reply))
            {
                ftps.disconnect();
                // System.err.println("Le serveur FTP a refusé la
connexion.");
                Toast.makeText(this,"Le serveur FTP a refusé la
connexion. " ,Toast.LENGTH_SHORT).show(); //Renvoie message sur le mobile
                System.exit(1);
            }
        }
        catch (IOException e)
        {
            if (ftps.isConnected())
            {
                try
                {
                    ftps.disconnect();
                }
                catch (IOException f)
                {
                    // do nothing
                }
            }
        }
    }
}

```

```

        //System.err.println("Le client FTP n'a pas pu se connecter
au serveur.");
        Toast.makeText(this, "Le serveur FTP a refusé la connexion.
", Toast.LENGTH_SHORT).show(); //Renvoie message sur le mobile
        e.printStackTrace();
        System.exit(1);
    }
    // Connexion
    try
    {
        ftps.setBufferSize(1000);
        if (!ftps.login(username.getText().toString(),
            mdp.getText().toString()))
        {
            ftps.logout();
            error = true;
            System.exit(1);
        }
        ftps.setFileType(FTP.BINARY_FILE_TYPE);
        ftps.enterLocalPassiveMode();
        ftps.sendCommand("OPTS UTF8 ON");
        Globals.global_ftps = ftps;
        Toast.makeText(this, "Connexion
effectue", Toast.LENGTH_SHORT).show(); //Renvoie message sur le mobile

        Intent i = new
Intent(getApplicationContext(), FilesActivity.class);
        startActivity(i);
    }
    catch (FTPConnectionClosedException e)
    {
        error = true;
        //System.err.println("Connexion au serveur fermée.");
        Toast.makeText(this, "Connexion
effectue", Toast.LENGTH_SHORT).show(); //Renvoie message sur le mobile
        e.printStackTrace();
    }
    catch (IOException e)
    {
        error = true;
        e.printStackTrace();
    }
    System.out.println(Environment.getDataDirectory());
    System.out.println(Environment.getExternalStorageDirectory());
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Gonflez le menu, cela ajoute des éléments à la barre d'action si
elle est présente.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {

    // Gérer les clics sur les éléments de la barre d'action ici. La
barre d'action
    // gère automatiquement les clics sur le bouton Home/Up, aussi
longtemps

```

```

        // lorsque vous spécifiez une activité parente dans
        AndroidManifest.xml.

        //Plus de detail :
https://stackoverflow.com/questions/28816725/mainactivity-java-action-
settings-cannot-be-resolved-or-is-not-a-field

        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}

```

FilesActivity.Java

```

package com.example.sae302;
//IMPORT
import android.os.Bundle;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import org.apache.commons.net.ftp.FTPFile;
import org.apache.commons.net.ftp.FTPSClient;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
////

public class FilesActivity extends Activity {

    FTPSClient ftps;
    ArrayList<String> folders = new ArrayList<String>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_files);
        String[] fileNames = null;
        FTPFile[] files;
        ftps = (FTPSClient)Globals.global_ftps;
        // Recuperer le dossier courant et liste les fichiers
        try
        {
            String pwd = ftps.printWorkingDirectory();
            if (pwd != null) {
                TextView myTextView = (TextView)
                findViewById(R.id.workindir);
                myTextView.setText(pwd);
            }
            fileNames = ftps.listNames();

```

```

        files = ftps.listDirectories();
        for (FTPFile file : files) {
            folders.add(file.getName());
        }
    }
    catch (IOException e)
    {
        e.printStackTrace();
        Intent i = new Intent(getApplicationContext(),
FtpActivity.class);
        startActivity(i);
    }
    final ListView listView = (ListView) findViewById(R.id.listview);
    final ArrayList<String> list = new ArrayList<String>();
    for (int i = 0; i < fileNames.length; ++i) {
        list.add(fileNames[i]);
    }
    final StableArrayAdapter adapter = new StableArrayAdapter(this,
        android.R.layout.simple_list_item_1, list);
    listView.setAdapter(adapter);
    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, final View view,
            int position, long id) {
            final String item = (String)
parent.getItemAtPosition(position);
            view.animate().setDuration(1000).alpha((float)0.5)
                .withEndAction(new Runnable() {
                    @Override
                    public void run() {
                        for (String folder : folders) {
                            if (item.equals(folder)) {
                                try
                                {
                                    ftps.cwd(item);
                                }
                                catch (IOException e)
                                {
                                    e.printStackTrace();
                                    Intent i = new
Intent(getApplicationContext(),
                                    FtpActivity.class);
                                    startActivity(i);
                                }
                                Intent i = new
Intent(getApplicationContext(), FilesActivity.class);
                                startActivity(i);
                            }
                        }
                    }
                })
        }
    });
    listView.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(AdapterView<?> parent, final
View view, int position, long id) {
            final String item = (String)
parent.getItemAtPosition(position);

```

```

        view.animate().setDuration(1000).alpha((float)0.5)
            .withEndAction(new Runnable() {
                @Override
                // SI APPUIE LONG ALORS LANCER L'ACTIVITE
DownloadActivity.class
                public void run() {
                    Intent i = new
Intent(getApplicationContext(), DownloadActivity.class);
                    i.putExtra("file", item);
                    for (String folder : folders) {
                        if (item.equals(folder)) {
                            i.putExtra("isDir", true);
                        }
                    }
                    startActivity(i);
                }
            });
        return true;
    }
}

@Override
//Fonction pour revenir en arriere
public void onBackPressed() {
    try
    {
        ftps.changeToParentDirectory();
    }
    catch (IOException e)
    {
        e.printStackTrace();
        Intent i = new Intent(getApplicationContext(),
FtpActivity.class);
        startActivity(i);
    }
    Intent i = new Intent(getApplicationContext(),
FilesActivity.class);
    startActivity(i);
}

private class StableArrayAdapter extends ArrayAdapter<String> {
    HashMap<String, Integer> mIdMap = new HashMap<String, Integer>();
    public StableArrayAdapter(Context context, int textViewResourceId,
        List<String> objects) {
        super(context, textViewResourceId, objects);
        for (int i = 0; i < objects.size(); ++i) {
            mIdMap.put(objects.get(i), i);
        }
    }
    @Override
    public long getItemId(int position) {
        String item = getItem(position);
        return mIdMap.get(item);
    }
    @Override
    public boolean hasStableIds() {
        return true;
    }
}
}

```

DownloadActivity.java

```
package com.example.sae302;

//IMPORT
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import org.apache.commons.net.ftp.FTPFile;
import org.apache.commons.net.ftp.FTPSClient;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.LinearLayout.LayoutParams;
import android.widget.TextView;
import android.widget.Toast;
///

public class DownloadActivity extends Activity {
    //Variables ici :
    FTPSClient ftps;
    Boolean isDir = false;
    String fname = null;
    //
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_download);
        //RATACHEZ AU LAYOUT ACTIVITY DOWNLOAD

        ftps = (FTPSClient)Globals.global_ftps;
        Intent myIntent = getIntent();
        fname = myIntent.getStringExtra("file");
        isDir = myIntent.getBooleanExtra("isDir", false);
        TextView tv = (TextView) findViewById(R.id.yousure);
        tv.setText("Êtes-vous sûr de vouloir télécharger " + fname + "?");
        Button btn_yes = (Button) findViewById(R.id.button_yes);
//BUTTON_YES
//SI POURSUIVRE LE TELECHARGEMENT
        btn_yes.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                LinearLayout parent = (LinearLayout)
findViewById(R.id.download_layout);
                View child = (View) findViewById(R.id.button_bar);
                parent.removeView(child);
                downloadFiles();
            }
        });
//SI BOUTON APPUYE SUR NON RENVOIE VERS ACTIVITE FILES , retour au
repertoire
    }
}
```



```

        Button btn_no = (Button) findViewById(R.id.button_no);
        btn_no.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(),
FilesActivity.class);
                startActivity(i);
            }
        });
    }

    private void downloadFiles() {
        String[] fileNames = null;
        ArrayList<String> folders = new ArrayList<String>();
        FTPFile[] files;
        // Layout/view stuff
        LinearLayout linlay = (LinearLayout)
findViewById(R.id.download_layout);
        LayoutParams lparams = new LayoutParams(
            LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT);
        TextView tv = (TextView) findViewById(R.id.yousure);
        tv.setText("tmp");
        File path = Environment.getExternalStoragePublicDirectory(
            Environment.DIRECTORY_DOWNLOADS);
        if (!isDir) {
            File file = new File(path, fname);
            FileOutputStream out;
            tv.setText("Fichier de téléchargement : " + fname);
            try {
                out = new FileOutputStream(file);
                if (ftps.retrieveFile(fname, out) == false) {
                    System.err.println("Erreur lors du téléchargement du
fichier");
                    out.close();
                    Intent i = new Intent(getApplicationContext(),
FilesActivity.class);
                    startActivity(i);
                }
                out.close();
            }
            catch (IOException e) {
                e.printStackTrace();
                Intent i = new Intent(getApplicationContext(),
FilesActivity.class);
                startActivity(i);
            }
        }
        else {
            File directory = new File(path, fname);
            FileOutputStream out;
            tv.setText("Changement repertoire " + fname);
            if (!directory.exists()) {
                directory.mkdir();
            }
            //get files and directories
            try
            {
                ftps.cwd(fname);
                fileNames = ftps.listNames();
                files = ftps.listDirectories();
                for (FTPFile file : files) {
                    folders.add(file.getName());
                }
            }
            catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

    }
}
catch (IOException e)
{
    e.printStackTrace();
    Intent i = new Intent(getApplicationContext(),
FilesActivity.class);
    startActivity(i);
}
TextView textView2 = new TextView(this);
textView2.setLayoutParams(lparams);
textView2.setText("");
linlay.addView(textView2);
isDir = false;
for (String f : fileNames) {
    for (String folder : folders) {
        if (f.equals(folder)) {
            isDir = true;
        }
    }
    if (!isDir) {
        textView2.append("Fichier de téléchargement : " + f +
"\n");
        try {
            out = new FileOutputStream(new File(directory, f));
            if (ftps.retrieveFile(f, out) == false) {
                System.err.println("Erreur lors du
téléchargement du fichier");
                out.close();
                Intent i = new Intent(getApplicationContext(),
FilesActivity.class);
                startActivity(i);
            }
            out.close();
        }
        catch (IOException e) {
            e.printStackTrace();
            Intent i = new Intent(getApplicationContext(),
FilesActivity.class);
            startActivity(i);
        }
    }
    isDir = false;
}
try
{
    ftps.changeToParentDirectory();
}
catch (IOException e)
{
    e.printStackTrace();
    Intent i = new Intent(getApplicationContext(),
FtpActivity.class);
    startActivity(i);
}
}
TextView textView3 = new TextView(this);
textView3.setLayoutParams(lparams);
textView3.setText("Téléchargement terminé");
//AFFICHE TELECHARGEMENT TERMINE
Toast.makeText(this, "Téléchargement terminé

```

```

", Toast.LENGTH_SHORT).show();
    linlay.addView(textView3);
    Button btn = new Button(this);
    btn.setId(666);
    btn.setText("Continuez");
    //SI APPUIE SUR BOUTON CONTINUEZ RENVOYEZ VERS ACTIVITE
FILESACTIVITY.CLASS
    linlay.addView(btn, lparams);
    btn.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(),
FilesActivity.class);
            startActivity(i);
        }
    });
}
}
}

```

PingActivity.java

```

package com.example.sae302;
//IMPORT
import androidx.appcompat.app.AppCompatActivity;

import android.net.wifi.WifiManager;
import android.text.format.Formatter;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;

import java.util.List;
import android.text.Editable;
import android.os.Bundle;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.ArrayList;

///

public class PingActivity extends AppCompatActivity {

    TextView ipadd;
    Button btnIP;
    ListView listePing;
    EditText edtIP;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

        setContentView(R.layout.activity_ping);

        // POUR RECUP L'IP

        ipadd = findViewById(R.id.ipadd);

        WifiManager wifiManager = (WifiManager)
getApplicationContext().getSystemService(WIFI_SERVICE);

ipadd.setText(Formatter.formatIpAddress(wifiManager.getConnectionInfo().get
IpAddress())); //PERMET DE RECUPERER L'ADRESSE LORS D'UNE CONNEXION A UN
RESEAU

        //INSTANCE

        btnIP = findViewById(R.id.btn_ping); // BOUTON POUR PING
        listePing = findViewById(R.id.listView_ping) ;// VOIR LES PING
        edtIP = findViewById(R.id.edit_ip) ; // CHANGER L'IP

    }

    //COMMANDE PING
    public void fExecuterPing(View view){

        Editable host = edtIP.getText(); //Host = Adresse IP Saisi par
l'utilisateur
        List<String> listeReponsePing = new ArrayList<String>();
        ArrayAdapter<String> adapterListe = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1,listeReponsePing);

        try {

            String cmdPing = "ping -c 3 "+host; //Ping 3 fois si plus
changer l'argument ping -c x
            Runtime r = Runtime.getRuntime();
            Process p = r.exec(cmdPing);
            BufferedReader in = new BufferedReader(new
InputStreamReader(p.getInputStream()));
            String inputLinhe;

            Toast.makeText(this,"Commande ping
execute",Toast.LENGTH_SHORT).show(); //Renvoie Commande Ping Execute

            //AFFICHE RESULTAT PING

            while((inputLinhe = in.readLine())!= null){
                listeReponsePing.add(inputLinhe);
                listePing.setAdapter(adapterListe);
            }
        }
    }

```

```

        } catch (Exception e) {

            Toast.makeText(this, "Erreur : "+
e.getMessage().toString(), Toast.LENGTH_SHORT).show(); // Si erreur lors du
test de connectivite

        }

    }

}

```

TcpUdpActivity.java

```

package com.example.sae302;
//IMPORT
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
////////

public class TcpUdpActivity extends AppCompatActivity {
    //declaration des variables ici :
    private Socket client;
    private PrintWriter printwriter;
    private EditText textField;
    private Button button;
    private String message;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tcp_udp); //ASSOCIE A L'ECRAN
ACTIVITY_TCP_UDP.xml
        //VARIABLES ICI :

        // référence au champ de texte
        textField = findViewById(R.id.editText1);

        // reference au champ bouton
        button = findViewById(R.id.button1);
    }
}

```

```

        // Pour écouter
        button.setOnClickListener(new View.OnClickListener() {

            public void onClick(View v) {

                // récupère le message texte sur le champ de texte
                message = textField.getText().toString();

                // démarre le Thread pour se connecter au serveur
                new Thread(new ClientThread(message)).start();

            }

        });

    }

    class ClientThread implements Runnable {
        private final String message;

        ClientThread(String message) {
            this.message = message;
        }

        @Override
        public void run() {
            try {
                // l'adresse IP et le port doivent être corrects pour
                // établir une connexion
                // Crée un socket de flux et le connecte au numéro de port
                // spécifié sur l'hôte nommé.
                client = new Socket("192.168.72.240", 4444); // connecte
                // au serveur

                printwriter = new
                PrintWriter(client.getOutputStream(), true);

                printwriter.write(message); // écrit le message dans le
                // flux de sortie

                printwriter.flush();
                printwriter.close();

                // fermeture de la connexion
                client.close();

            } catch (IOException e) {
                e.printStackTrace();
            }

            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    textField.setText("");
                }
            });
        }
    }
}

```

ServerActivity.java

```
package com.example.sae302;
//IMPORT
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.annotation.SuppressLint;
import android.net.wifi.WifiInfo;
import android.net.wifi.WifiManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.UnknownHostException;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;

@SuppressLint("SetTextI18n")
///

public class ServerActivity extends AppCompatActivity {
//VARIABLES ICI :
    ServerSocket serverSocket;
    Thread Thread1 = null;

    TextView tvIP, tvPort;
    TextView tvMessages;

    EditText etMessage;
    Button btnSend;

    public static String SERVER_IP = "";
    public static final int SERVER_PORT = 8080; //PORT PAR DEFALT DU
SERVEUR A MODIFIER ICI

    String message;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_server);

        tvIP = findViewById(R.id.tvIP); //IP
        tvPort = findViewById(R.id.tvPort); //PORT
        tvMessages = findViewById(R.id.tvMessages); //Message
        etMessage = findViewById(R.id.etMessage);
        btnSend = findViewById(R.id.btnSend); //Envoie

        try {
            SERVER_IP = getLocalIpAddress();
        } catch (UnknownHostException e) {
```

```

        e.printStackTrace();
    }

    Thread1 = new Thread(new Thread1());
    Thread1.start();

    btnSend.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            message = etMessage.getText().toString().trim();
            if (!message.isEmpty()) {
                new Thread(new Thread3(message)).start();
            }
        }
    });
}

private String getLocalIpAddress() throws UnknownHostException {
    WifiManager wifiManager = (WifiManager)
getApplicationContext().getSystemService(WIFI_SERVICE);
    assert wifiManager != null;
    WifiInfo wifiInfo = wifiManager.getConnectionInfo();
    int ipInt = wifiInfo.getIpAddress();
    return InetAddress.getByAddress(
ByteBuffer.allocate(4).order(ByteOrder.LITTLE_ENDIAN).putInt(ipInt).array()
)
        .getHostAddress();
}

private PrintWriter output;
private BufferedReader input;

class Thread1 implements Runnable {

    @Override
    public void run() {
        Socket socket;
        try {
            serverSocket = new ServerSocket(SERVER_PORT);
            runOnUiThread(new Runnable() {

                @Override
                public void run() {
                    tvMessages.setText("Non connecte");
                    tvIP.setText("IP: " + SERVER_IP);
                    tvPort.setText("Port: " +
String.valueOf(SERVER_PORT));
                }
            });
        } catch {
            socket = serverSocket.accept();

            output = new PrintWriter(socket.getOutputStream());
            input = new BufferedReader(new
InputStreamReader(socket.getInputStream()));

            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    tvMessages.setText("Connecté\n");

```



```

        }
        });
        new Thread(new Thread2()).start();
    } catch (IOException e) {
        e.printStackTrace();
    }
} catch (IOException e) {
    e.printStackTrace();
}
}

private class Thread2 implements Runnable {
    @Override
    public void run() {
        while (true) {
            try {
                final String message = input.readLine();
                if (message != null) {
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            tvMessages.append("client:" + message +
"\n");
                        }
                    });
                } else {
                    Thread1 = new Thread(new Thread1());
                    Thread1.start();
                    return;
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

class Thread3 implements Runnable {
    private String message;

    Thread3(String message) {
        this.message = message;
    }

    @Override
    public void run() {
        output.write(message);
        output.flush();
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                tvMessages.append("server: " + message + "\n");
                etMessage.setText("");
            }
        });
    }
}
}

```

activity_splashscreen.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SplashScreenActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/splashscreen"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="0dp" />

    <ProgressBar
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="42dp"
        android:layout_marginTop="-78dp"
        android:layout_marginEnd="500dp"
        android:indeterminate="true"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.425"
        app:layout_constraintStart_toEndOf="@+id/imageView"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />

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

activity_main.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    tools:layout_editor_absoluteX="-20dp"
    tools:layout_editor_absoluteY="-7dp">

    <Button
        android:id="@+id/ftp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="112dp"
        android:layout_marginTop="240dp"
        android:layout_marginEnd="112dp"
        android:layout_marginBottom="2dp"
        android:text="ACCES FTP"
        app:layout_constraintBottom_toTopOf="@+id/ping"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.457"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/ping"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="112dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="112dp"
        android:layout_marginBottom="21dp"
        android:text="EFFECTUER UN PING"
        app:layout_constraintBottom_toTopOf="@+id/tcpudp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ftp" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="135dp"
        android:layout_marginTop="126dp"
        android:layout_marginEnd="135dp"
        android:layout_marginBottom="22dp"
        android:text="SAE 302 - Theo Dufour"
        android:textColor="#FF0000"
        android:textColorHighlight="#000000"
        app:layout_constraintBottom_toTopOf="@+id/ftp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/tcpudp"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="112dp"
        android:layout_marginTop="24dp"
        android:layout_marginEnd="112dp"
        android:text="TCP Client"
        app:layout_constraintBottom_toTopOf="@+id/server"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.492"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ping" />

<Button
    android:id="@+id/server"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="112dp"
    android:layout_marginTop="22dp"
    android:layout_marginEnd="122dp"
    android:layout_marginBottom="365dp"
    android:text="TCP Serveur"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tcpudp" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

activity_ftp.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/workindir"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:clickable="false"
        android:longClickable="false"
        android:paddingBottom="5dip"
        android:paddingTop="5dip"
        android:background="#b5b6b8"
        android:text="pasderepertoire"
        android:textColor="#000000"
        android:textSize="17sp"
        android:textStyle="bold" />

    <ListView
        android:id="@+id/listview"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

</LinearLayout>

```

activity_files.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/workindir"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:clickable="false"
        android:longClickable="false"
        android:paddingBottom="5dip"
        android:paddingTop="5dip"
        android:background="#b5b6b8"
        android:text="pasderepertoire"
        android:textColor="#000000"
        android:textSize="17sp"
        android:textStyle="bold" />

    <ListView
        android:id="@+id/listview"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

activity_download.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/download_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/downloading"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:clickable="false"
        android:longClickable="false"
        android:paddingBottom="5dip"

        android:paddingTop="5dip"
        android:background="#b5b6b8"
        android:text="Téléchargement"
        android:textColor="#000000"
        android:textSize="17sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/yousure"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="êtes-vous sûr ?" />

    <LinearLayout
        android:id="@+id/button_bar"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="horizontal" >

        <Button
            android:id="@+id/button_yes"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="OUI"
            android:background="#b5b6b8"
            android:padding="5dp" />

        <Button
            android:id="@+id/button_no"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="NON"
            android:background="#b5b6b8"
            android:padding="5dp" />

    </LinearLayout>
</LinearLayout>
```

activity_ping.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PingActivity">

    <TextView
        android:id="@+id/ipadd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="200dp"
        android:layout_marginTop="84dp"
        android:layout_marginEnd="143dp"
        android:text="Hello World !"
        android:textColor="#4CAF50"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="104dp"
        android:layout_marginTop="84dp"
        android:layout_marginEnd="27dp"
        android:text="IP de l'appareil ->"
        app:layout_constraintEnd_toStartOf="@+id/ipadd"
        app:layout_constraintHorizontal_bias="0.594"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtIp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="140dp"
        android:layout_marginTop="140dp"
        android:layout_marginEnd="140dp"
        android:text="Entrez l'IP :"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/ipadd" />

    <EditText
        android:id="@+id/edit_ip"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="120dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="120dp"
```

```

        android:layout_marginBottom="17dp"
        android:textColor="#4CAF50"
        app:layout_constraintBottom_toTopOf="@+id/btn_ping"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtIp">

</EditText>

<Button
    android:id="@+id/btn_ping"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="140dp"
    android:layout_marginTop="17dp"
    android:layout_marginEnd="140dp"
    android:onClick="fExecuterPing"
    android:text="PING"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edit_ip" />

<ListView
    android:id="@+id/listView_ping"
    android:layout_width="350dp"
    android:layout_height="200dp"
    android:layout_marginStart="42dp"
    android:layout_marginTop="40dp"
    android:layout_marginEnd="42dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btn_ping" />

<ImageView
    android:id="@+id/imageView3"
    android:layout_width="119dp"
    android:layout_height="300dp"
    android:layout_marginStart="-250dp"
    android:layout_marginEnd="22dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.547"
    app:layout_constraintStart_toEndOf="@+id/edit_ip"
    app:srcCompat="@drawable/ping"
    tools:layout_editor_absoluteY="24dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

```


activity_tcp_udp.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".TcpUdpActivity">

    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="120dp"
        android:layout_marginTop="240dp"
        android:hint="Ecrire un message"
        android:layout_marginStart="24dp"
        android:layout_marginEnd="24dp"
        android:layout_marginRight="120dp"
        android:textAlignment="center" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="120dp"
        android:layout_marginRight="120dp"

        android:text="Envoye un message" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="125dp"
        android:layout_marginTop="-225dp"
        android:layout_marginRight="100dp"
        android:text="Tcp Client - SAE 302"
        android:textStyle="bold"
        android:textColor="#00f100"
        android:textSize="14sp" />

</LinearLayout>
```

activity_server.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_margin="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tvIP"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium" />

    <TextView
        android:id="@+id/tvPort"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tvIP"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium" />

    <TextView
        android:id="@+id/tvConnectionStatus"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tvPort"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium" />

    <TextView
        android:id="@+id/tvMessages"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@+id/etMessage"
        android:layout_below="@+id/tvConnectionStatus"
        android:inputType="textMultiLine"

android:textAppearance="@style/Base.TextAppearance.AppCompat.Medium" />

    <EditText
        android:id="@+id/etMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@+id/btnSend"
        android:hint="Enter Message"
        android:inputType="text" />

    <Button
        android:id="@+id/btnSend"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:text="ENVOYER" />

</RelativeLayout>
```

Android Manifest.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">
    <!-- PERMISSION -->
    <!-- //////////////////////////////////////// -->
    <uses-permission android:name="android.permission.INTERNET" />    <!--
ACCES A INTERNET -->
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>    <!-- ACCES RESEAU -->
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<!-- ACCES WIFI -->
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" /><!-- ACCES WIFI
-->
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <!-- //////////////////////////////////////// -->
    <!-- PARTIE APPLICATION -->
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SAE302"
        tools:targetApi="31">
        <activity
            android:name=".DownloadActivity"
            android:exported="false">
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity
            android:name=".FilesActivity"
            android:exported="false">
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity
            android:name=".ServerActivity"
            android:exported="false"
            android:parentActivityName=".MainActivity">
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity
            android:name=".TcpUdpActivity"
            android:exported="false"
            android:parentActivityName=".MainActivity">
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity
```

```

        android:name=".SplashScreenActivity"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER"
/>

        </intent-filter>

        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
    <activity
        android:name=".PingActivity"
        android:parentActivityName=".MainActivity">
        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
    <activity
        android:name=".FtpActivity"
        android:exported="true"
        android:parentActivityName=".MainActivity">
        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
    <activity
        android:name=".MainActivity"
        android:exported="true">
        <meta-data
            android:name="android.app.lib_name"
            android:value="" />
    </activity>
</application>
</manifest>

```