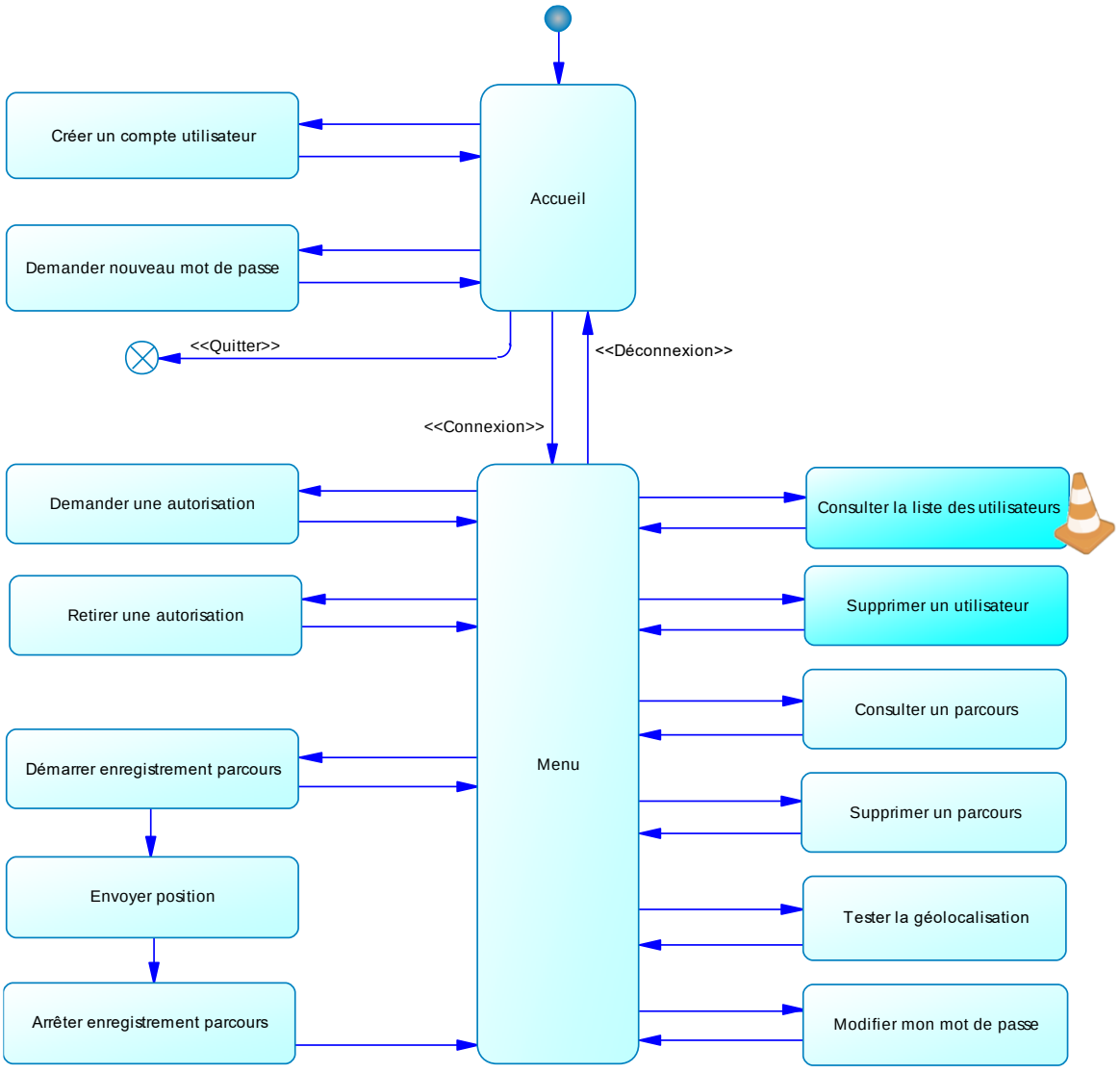
5- Développement de l'application mobile Android

5-4 ConsulterUtilisateurs (consultation de la liste des utilisateurs)

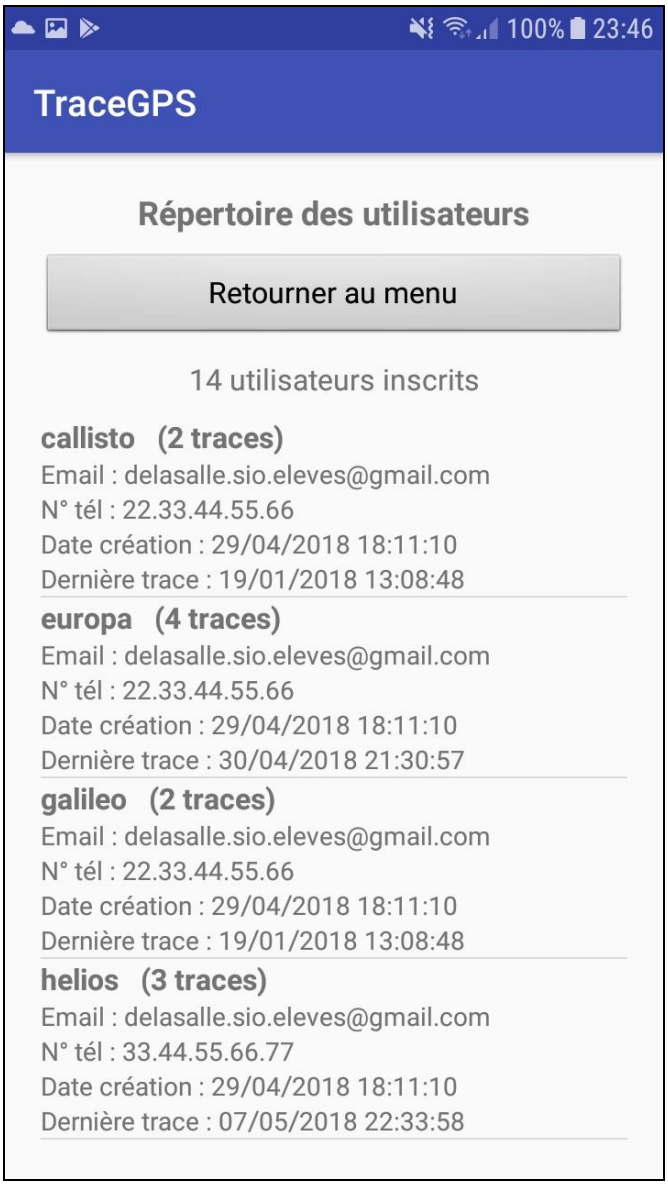
1- Situation de l'activité dans la structure de l'application
2- Modification du fichier strings.xml
3- Création de l'activité
4- Création de l'interface graphique
5- Création d'un layout pour l'affichage des données de chaque utilisateur
6- Modification de la programmation Java de MenuGeneral.java
7- Programmation Java de l'activité ListerUtilisateurs.java

1- Situation de l'activité dans la structure de l'application



2- Modification du fichier strings.xml

L'interface graphique à créer :

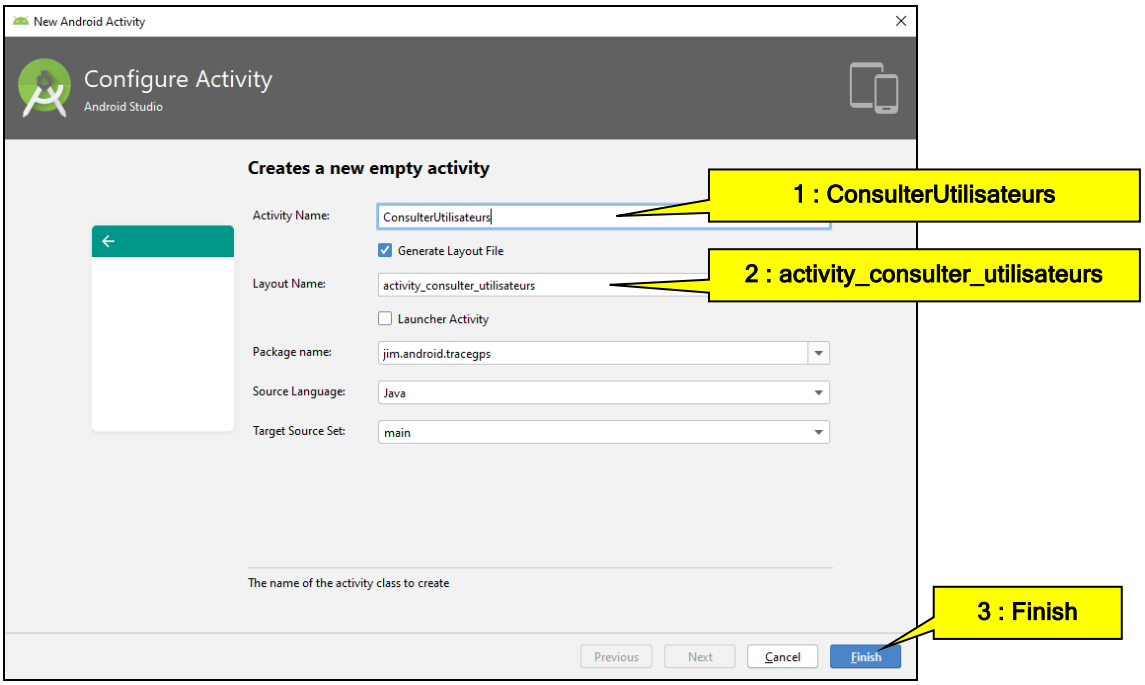


Dans le dossier **res/values**, complétez le fichier **strings.xml** avec le code suivant :

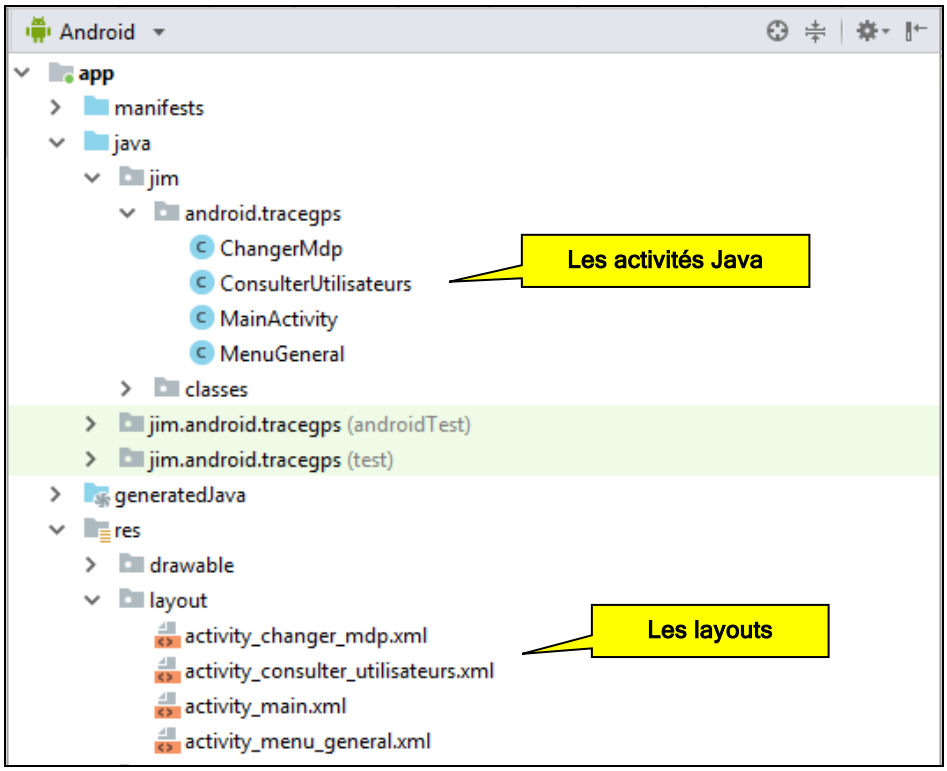
```
<!-- Les textes de la page de consultation de la liste des utilisateurs -->
<string name="lister_utilisateurs_titre1">Répertoire des utilisateurs</string>
<string name="lister_utilisateurs_bouton_retourner">Retourner au menu</string>
```

3- Création de l'activité

Créer une nouvelle activité en faisant un clic droit sur la racine **app** du projet et en choisissant la commande **New / Activity / Empty Activity** :



L'activité **ConsulterUtilisateurs.java** et le layout **activity_consulter_utilisateurs.xml** sont alors créés :



4- Création de l'interface graphique de l'activité

A la création d'une nouvelle activité, l'interface comporte automatiquement un **ConstraintLayout** vide.

Comme d'habitude, nous allons commencer par remplacer le **ConstraintLayout** proposé par un **LinearLayout (vertical)** qui est beaucoup plus souple pour positionner les objets graphiques.

Le **ConstraintLayout** ne pouvant être ni modifié ni supprimé en mode **Design**, on va donc le modifier en mode **Text** :

Remplacez ConstraintLayout par LinearLayout

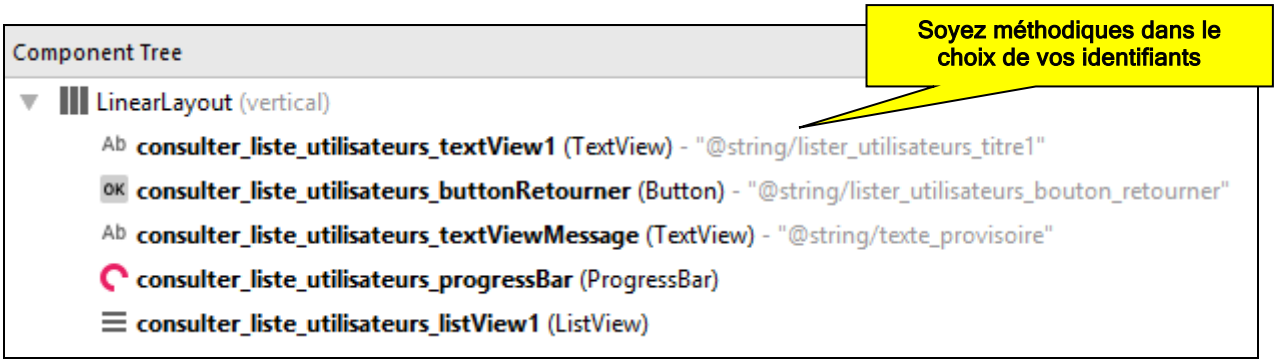
```
<?xml version="1.0" encoding
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:padding="@dimen/tailleMarges"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ConsulterUtilisateurs">

</LinearLayout>
```

Supprimez cette ligne

Ajoutez ces 2 lignes

Revenez maintenant en mode **Design** et placez les différents composants en suivant la structure suivante et en utilisant bien sûr les chaines du fichier **strings.xml** :

Soyez méthodiques dans le choix de vos identifiants



Le code XML du layout :

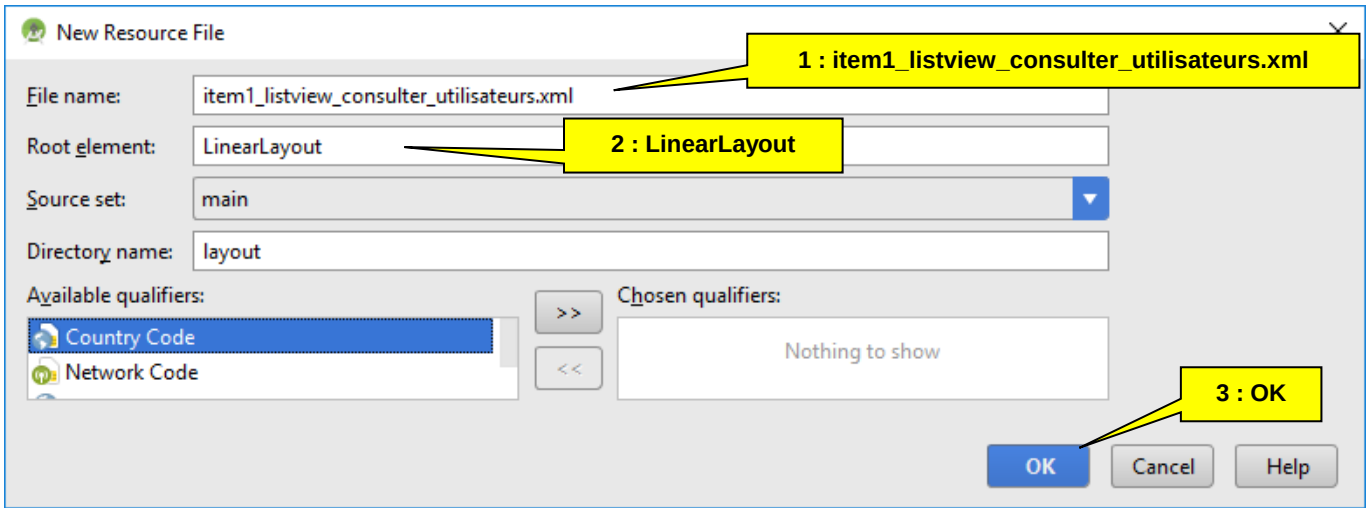
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:padding="@dimen/tailleMarges"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="jim.android.tracegps.ConsulterUtilisateurs">

    <TextView
        android:id="@+id/consulter_liste_utilisateurs_textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="10dp"
        android:text="@string/lister_utilisateurs_titre1"
        android:textAlignment="center"
        android:textSize="18sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/consulter_liste_utilisateurs_buttonRetourner"
        style="@android:style/Widget.Button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/lister_utilisateurs_bouton_retourner"
        android:textSize="16sp" />

    <TextView
        android:id="@+id/consulter_liste_utilisateurs_textViewMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:text="@string/texte_provisoire"
        android:textAlignment="center"
        android:textSize="16sp" />

    <ProgressBar
        android:id="@+id/consulter_liste_utilisateurs_progressBar"
        style="?android:attr/progressBarStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal" />

    <ListView
        android:id="@+id/consulter_liste_utilisateurs_listView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

5- Création d'un layout pour l'affichage des données de chaque utilisateur

Créez un nouveau layout pour afficher les données de chaque item du ListView, en visualisant pour chaque item :

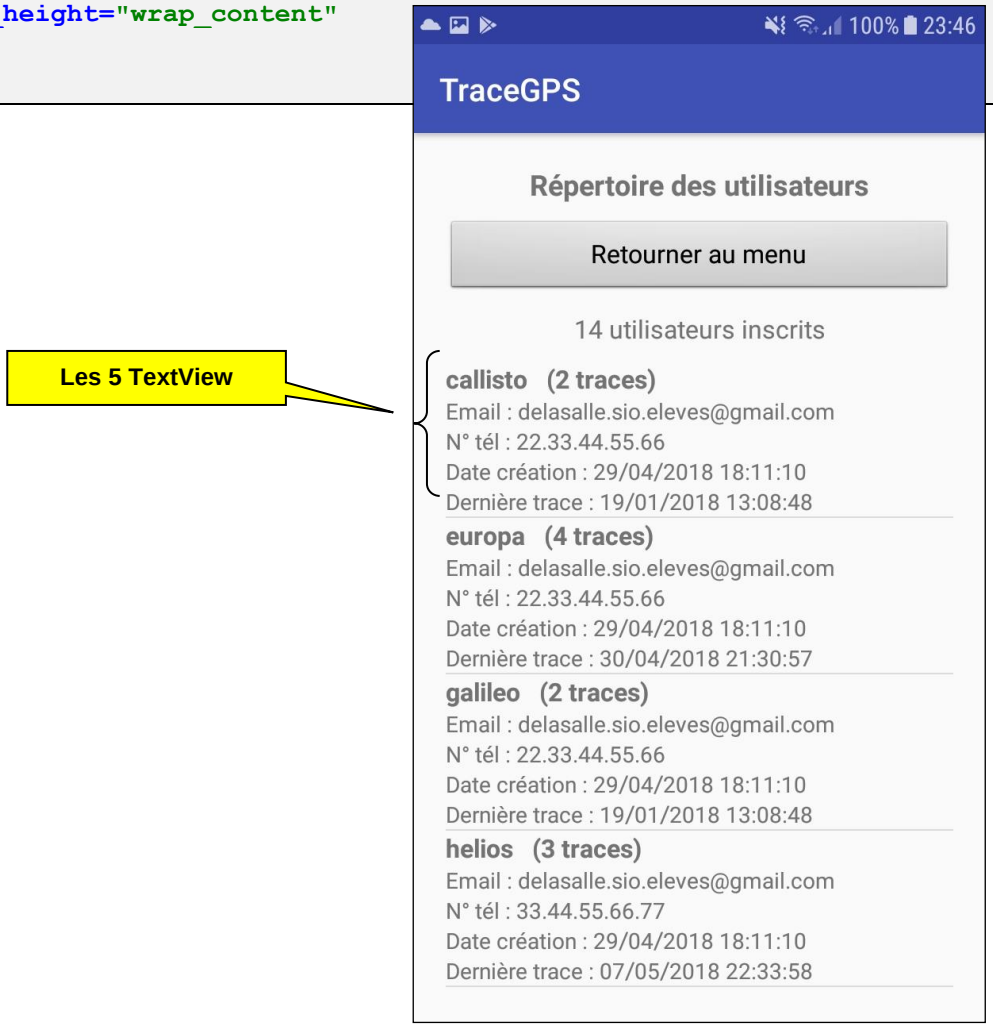
- le pseudo de l'utilisateur et le nombre de traces (en gras)
- son adresse mail
- son numéro de téléphone
- la date de création du compte utilisateur
- la date de sa dernière trace

Créer ce nouveau layout en faisant un clic droit sur le dossier **res / layout** du projet **app** et en choisissant la commande **New / Layout resource file** :



Ce layout contiendra 5 contrôles **TextView** ; vous pouvez le créer directement en mode **Text** :

```
<?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="wrap_content"
    android:orientation="vertical">

    <TextView android:id="@+id/item1_textView_pseudo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textStyle="bold"
        />
    <TextView android:id="@+id/item1_textView_adrmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />
    <TextView android:id="@+id/item1_textView_numtel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />
    <TextView android:id="@+id/item1_textView_date_creation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />
    <TextView android:id="@+id/item1_textView_date_derniere_trace"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

</LinearLayout>
```



6- Modification de la programmation Java de MenuGeneral.java

Complétez l'écouteur d'événement associé à **buttonConsulterUtilisateurs** :

```
/** classe interne pour gérer le clic sur le bouton buttonConsulterUtilisateurs. */
private class buttonConsulterUtilisateursClickListener implements View.OnClickListener{
    public void onClick(View v) {
        // crée une Intent pour lancer l'activité
        Intent uneIntent = new Intent(MenuGeneral.this, ConsulterUtilisateurs.class);
        // passe nom, mdp et typeUtilisateur à l'Intent
        uneIntent.putExtra(EXTRA_PSEUDO, pseudo);
        uneIntent.putExtra(EXTRA_MDP, mdp);
        uneIntent.putExtra(EXTRA_TYPE_UTILISATEUR, typeUtilisateur);
        // démarre l'activité à partir de l'Intent
        startActivity(uneIntent);
    }
}
```

Testez cette étape sur un mobile réel **en vous connectant avec un compte administrateur**, et corrigez les erreurs si besoin.

Le bouton **Consulter la liste des utilisateurs** doit activer l'activité **ConsulterUtilisateurs** :

7- Programmation Java de l'activité ConsulterUtilisateurs.java

7-1 Déclarations diverses et initialisation des objets

Dans le fichier **ConsulterUtilisateurs.java**, ajoutez le code indiqué en gras :

```
package jim.android.tracegps;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import android.widget.Button;
import android.widget.TextView;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.view.View;
import android.content.Intent;

public class ConsulterUtilisateurs extends AppCompatActivity {

    // les objets du layout
    private TextView textViewMessage;                   // le TextView pour afficher le message
    private Button buttonRetourner;                     // le Button pour retourner au menu
    private ProgressBar progressBar;                    // le ProgressBar pour afficher le cercle de chargement
    private ListView laListView;                // le ListView pour afficher les utilisateurs

    // le passage des données entre activités se fait au moyen des "extras" qui sont portés par les Intent.
    // un extra est une couple de clé/valeur
    // nous en utiliserons 2 ici, dont voici les 2 clés et les 2 variables associées :
    private final String EXTRA_PSEUDO = "pseudo";
    private final String EXTRA_MDP = "mdp";
    private String pseudo;
    private String mdp;

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

        // récupération du nom, et du mot de passe passés par l'activité précédente
        Intent uneIntent = getIntent();
        pseudo = uneIntent.getStringExtra(EXTRA_PSEUDO);
        mdp = uneIntent.getStringExtra(EXTRA_MDP);

        // récupération des objets du layout grâce à leur ID
        buttonRetourner = (Button) findViewById(R.id.consulter_liste_utilisateurs_buttonRetourner);
        textViewMessage = (TextView) findViewById(R.id.consulter_liste_utilisateurs_textViewMessage);
        laListView = (ListView) findViewById(R.id.consulter_liste_utilisateurs_listView1);
        progressBar = (ProgressBar) findViewById(R.id.consulter_liste_utilisateurs_progressBar);
        // arrête le cercle de chargement
        progressBar.setVisibility(View.GONE);

        // association d'un écouteur d'évenement au bouton
        buttonRetourner.setOnClickListener ( new buttonRetournerClickListener());
    }

    /** classe interne pour gérer le clic sur le bouton buttonRetourner. */
    private class buttonRetournerClickListener implements View.OnClickListener {
        public void onClick(View v) {
            finish();
        }
    }

} // fin de l'activité
```

Testez l'application et le bon fonctionnement du bouton **Retourner au menu**.

7-2 Mise en place de la tâche asynchrone

A la suite des **import** existants, ajoutez les **import** suivants :

```
import jim.classes.*;
import java.util.ArrayList;
import android.os.AsyncTask;
```

A la suite des déclarations existantes, ajoutez la déclaration suivante :

```
private ArrayList<Utilisateur> lesUtilisateurs;          // contient la collection des utilisateurs
```

A la fin de l'activité, ajoutez la tâche asynchrone **TacheGetTousLesUtilisateurs** et la fonction provisoire **afficherLesUtilisateurs** :

```
// ------------------------------------------------------------------------------------------------------
// ---------------------- tâche asynchrone pour PasserelleServicesWebXML.getTousLesUtilisateurs ----------
// ------------------------------------------------------------------------------------------------------

private class TacheGetTousLesUtilisateurs extends AsyncTask<ArrayList<Utilisateur>, Void, String> {
    protected void onPreExecute() {
        // cette méthode exécute un traitement initial avant de lancer la tâche longue
        progressBar.setVisibility(View.VISIBLE);    // démarre le cercle de chargement
    }

    protected String doInBackground(ArrayList<Utilisateur>... params) {
        // cette méthode permet de lancer l'exécution de la tâche longue
        // ici, on met à jour la collection d'utilisateurs passée en paramètre à partir du service web
        lesUtilisateurs = params[0];
        String msg = PasserelleServicesWebXML.getTousLesUtilisateurs(pseudo, Outils.sha1(mdp), lesUtilisateurs);
        // et on retourne la réponse fournie par le service web
        return msg;
    }

    protected void onPostExecute(String param) {
        // cette méthode est automatiquement appelée quand la tâche longue se termine
        progressBar.setVisibility(View.GONE);               // arrête le cercle de chargement

        if ( param.startsWith("Erreur"))
            // il s'agit d'un message d'erreur ; on l'affiche
            textViewMessage.setText(param);
        else
        {   // afficher la liste des utilisateurs
            afficherLesUtilisateurs();
        }
    }
}
```

```
// afficher la liste des utilisateurs
public void afficherLesUtilisateurs() {
    // on affiche le nombre d'utilisateurs
    textViewMessage.setText(lesUtilisateurs.size() + " utilisateurs inscrits");
} // fin de la fonction afficherLesUtilisateurs
```

Complétez la fonction **onCreate** pour appeler la tâche asynchrone :

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

    // récupération du pseudo, et du mot de passe passés par l'activité précédente
    Intent uneIntent = getIntent();
    pseudo = uneIntent.getStringExtra(EXTRA_PSEUDO);
    mdp = uneIntent.getStringExtra(EXTRA_MDP);

    // récupération des objets du layout grâce à leur ID
    buttonRetourner = (Button) findViewById(R.id.consulter_liste_utilisateurs_buttonRetourner);
    textViewMessage = (TextView) findViewById(R.id.consulter_liste_utilisateurs_textViewMessage);
    laListView = (ListView) findViewById(R.id.consulter_liste_utilisateurs_listView1);

    // association d'un écouteur d'évenement au bouton
    buttonRetourner.setOnClickListener ( new buttonRetournerClickListener());

    // chargement de la liste des utilisateurs à partir du service web à l'aide d'une tâche asynchrone
    lesUtilisateurs = new ArrayList<Utilisateur>();
    new TacheGetTousLesUtilisateurs().execute(lesUtilisateurs);
}
```

Testez l'application ; vous devez obtenir un affichage provisoire similaire à celui-ci :

7-3 Gestion de l'affichage des utilisateurs dans la ListView

A la suite des **import** existants, ajoutez les **import** suivants :

```
import java.util.HashMap;
import android.widget.SimpleAdapter;
```

Complétez la fonction définitive **afficherLesUtilisateurs** :

```
// afficher la liste des utilisateurs
public void afficherLesUtilisateurs() {
    // on affiche le nombre d'utilisateurs
    textViewMessage.setText(lesUtilisateurs.size() + " utilisateurs inscrits");

    // création de la ArrayList qui permettra de remplir la listView
    ArrayList<HashMap<String, String>> lesElementsDuListView = new ArrayList<HashMap<String, String>>();

    for (int i = 0; i < lesUtilisateurs.size(); i++)
    {   Utilisateur unUtilisateur = lesUtilisateurs.get(i);
        // création d'une HashMap pour insérer les informations d'un utilisateur
        HashMap<String, String> element = new HashMap<String, String>();
        element.put("pseudo", unUtilisateur.getPseudo() + "   (" + unUtilisateur.getNbTraces() + " traces)");
        element.put("adrmail", "Email : " + unUtilisateur.getAdrMail());
        element.put("numtel", "N° tél : " + unUtilisateur.getNumTel());
        element.put("creation", "Date création : " + Outils.formaterDateHeureFR(unUtilisateur.getDateCreation()));
        if (unUtilisateur.getDateDerniereTrace() != null)
            element.put("derniere", "Dernière trace : " +
Outils.formaterDateHeureFR(unUtilisateur.getDateDerniereTrace()));
        else
            element.put("derniere_trace", "Dernière trace : " );
        // ajoute le HashMap dans le ArrayList
        lesElementsDuListView.add(element);
    }

    // le SimpleAdapter met les items de la liste (lesElementsDuListView) dans la vue item1_listview_consulter_utilisateurs
    SimpleAdapter monAdapter = new SimpleAdapter (getBaseContext(), lesElementsDuListView,
            R.layout.item1_listview_consulter_utilisateurs,
            new String[] {
                    "pseudo",
                    "adrmail",
                    "numtel",
                    "creation",
                    "derniere"},
            new int[] {
                    R.id.item1_textView_pseudo,
                    R.id.item1_textView_adrmail,
                    R.id.item1_textView_numtel,
                    R.id.item1_textView_date_creation,
                    R.id.item1_textView_date_derniere_trace});

    // attribuer au listView l'adapter que l'on vient de créer
    laListView.setAdapter(monAdapter);
} // fin de la fonction afficherLesUtilisateurs
```

Testez l'application ; vous devez obtenir un affichage définitif similaire à celui-ci :