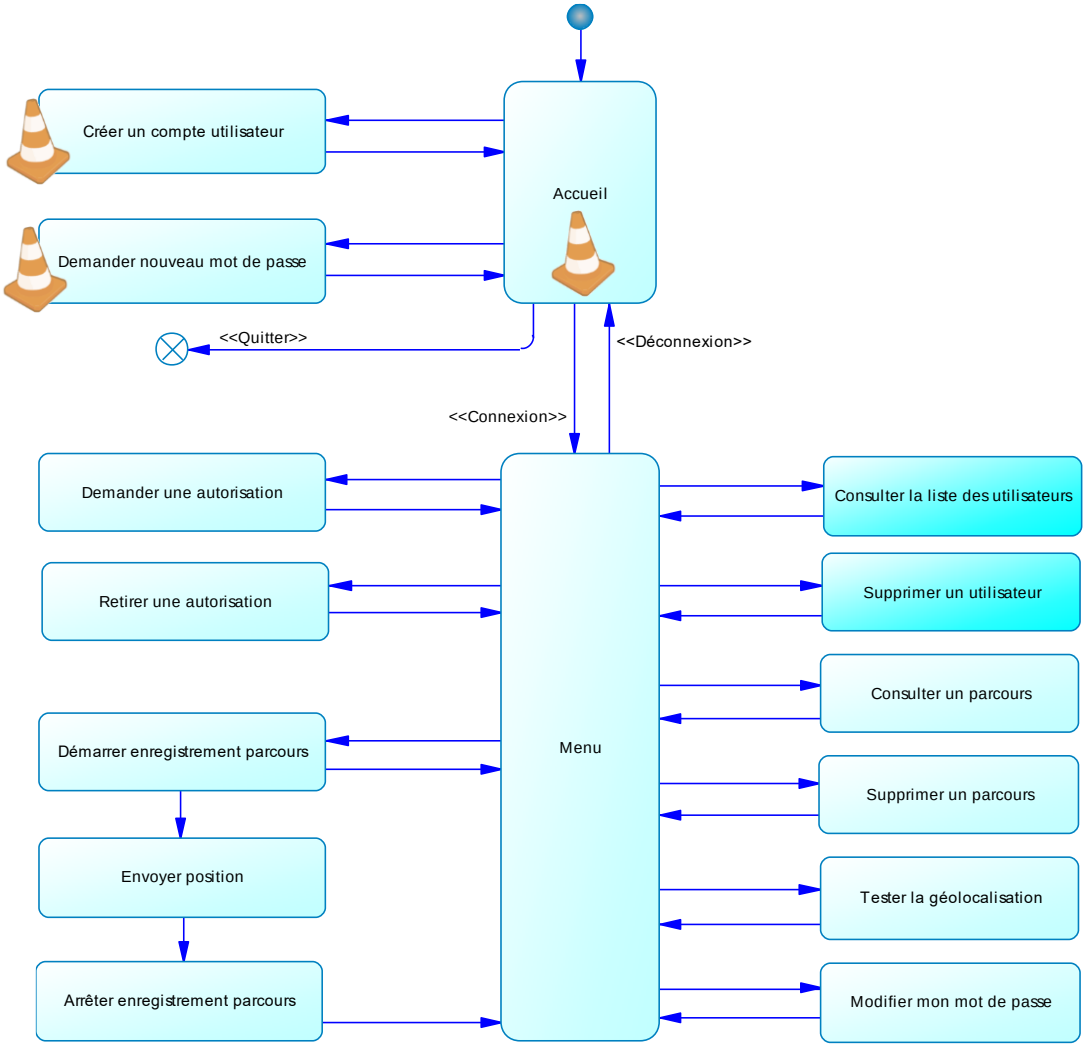
AP

5- Développement de l'application mobile Android

5-1 MainActivity
(Identification, création de compte et mot de passe oublié)

1- Situation de l'activité dans la structure de l'application
2- Création du projet
3- Importation des classes Java développées à l'étape précédente
4- Modification du manifeste
5- Création des fichiers ressources
6- Création de l'interface graphique
7- Programmation Java de l'activité MainActivity.java

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

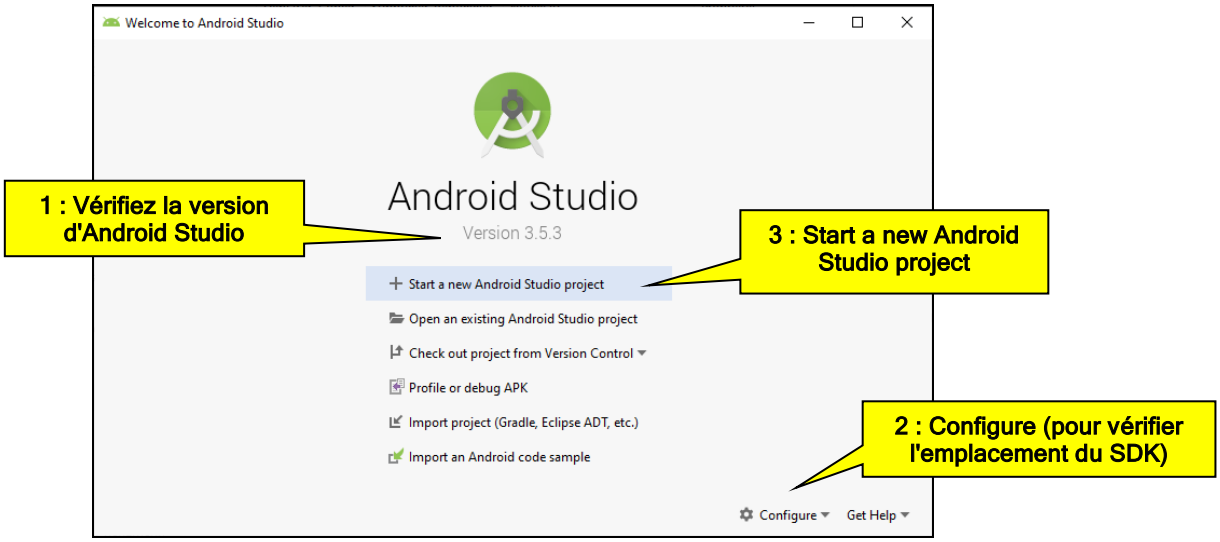


2- Création du projet

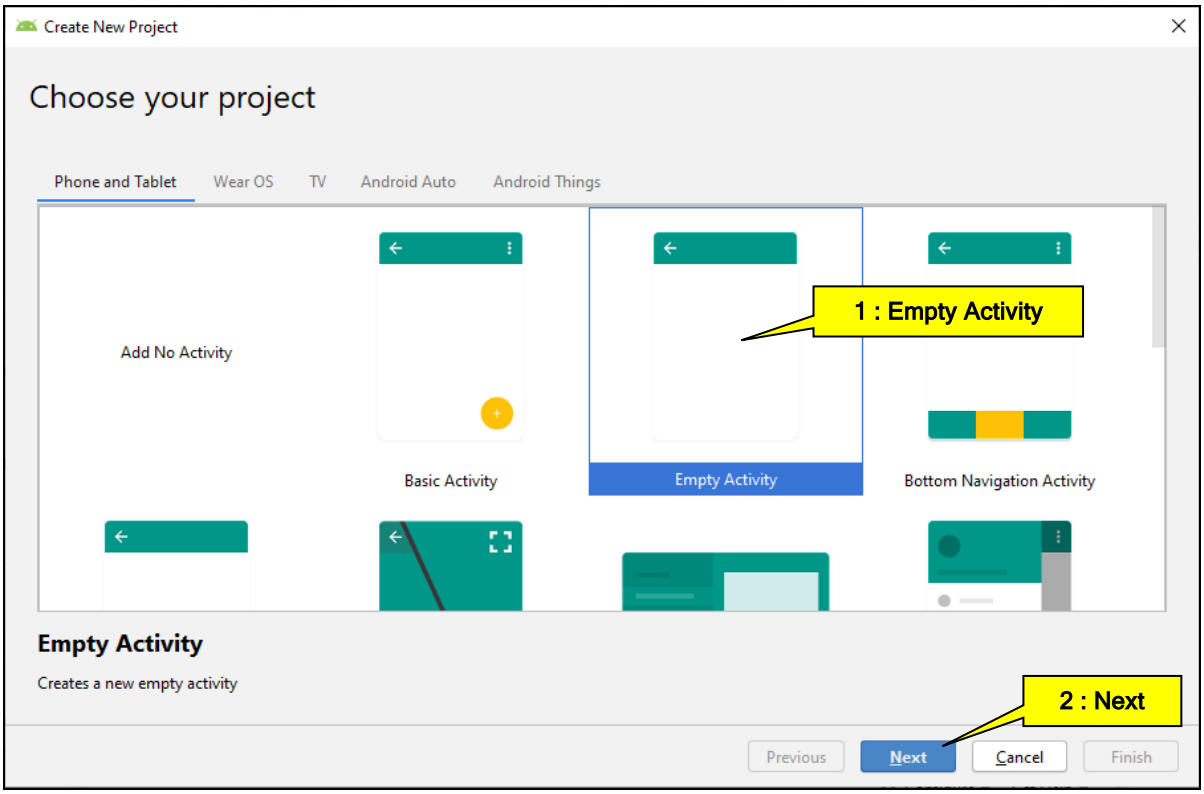
> ⚠ Si ce n'est pas encore fait, créez un espace de travail (workspace) réservé à Android Studio**.** Cet emplacement sera ***D:\ WorkspacesAndroid\ws-android-xxx*** (où xxx est votre nom).

> ⚠ Vérifiez ensuite que Android Studio connaît bien l'emplacement du SDK Android avec le menu **Configure / SDK Manager.** Cet emplacement est ***D:\Programmes\android-sdks***.

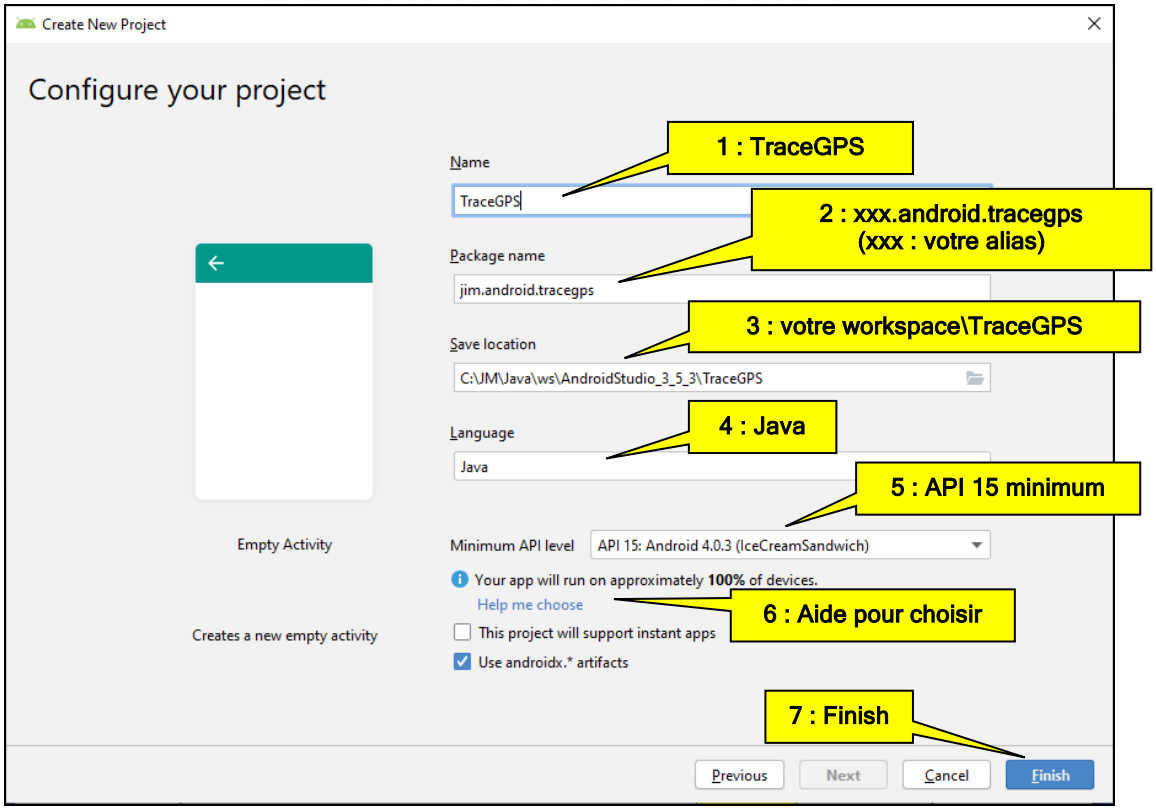
Une fois cette vérification faite, créez un projet (si le précédent projet est ouvert, fermez-le avec la commande **File / Close project**) :



Choisissez le type de projet (**Empty Activity**) :

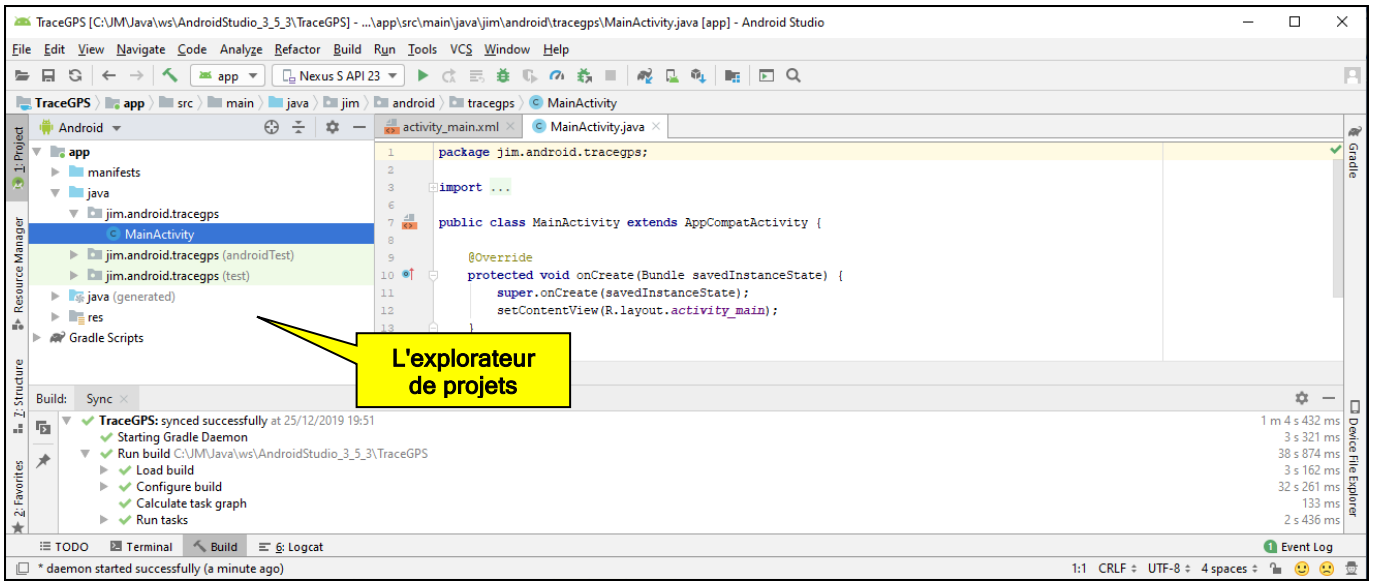


Donnez un nom au projet et au package, choisissez l'emplacement du projet, le langage et la version minimum d'Android permettant l'exécution de l'application :



Au bout d'un temps variable, on obtient un écran dans lequel on peut distinguer :

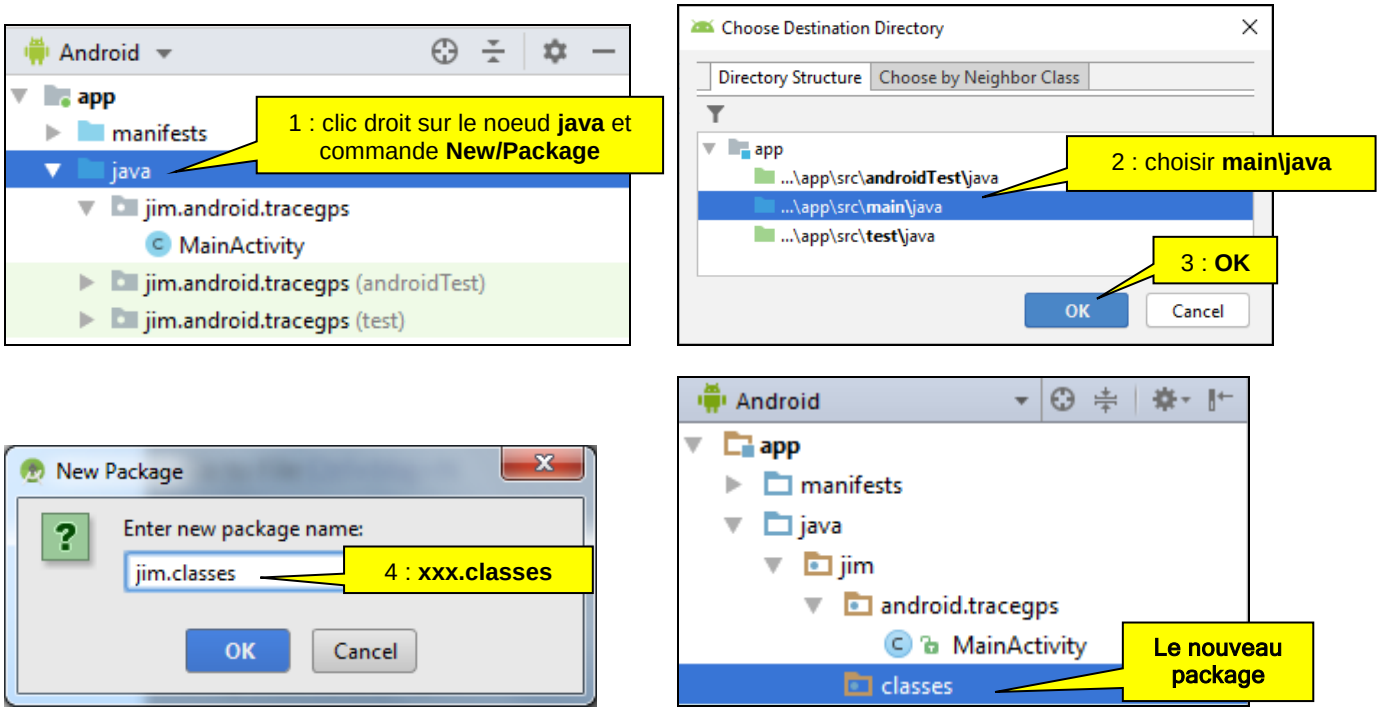
- la structure du projet (à gauche, dans l'explorateur de projets)
- le fichier **activity_main.xml** décrivant l'interface graphique
- le fichier **MainActivity.java** décrivant les traitements de l'application



3- Importation des classes Java développées à l'étape précédente

Ces classes ont été développées et testées dans la phase précédente du PPE.

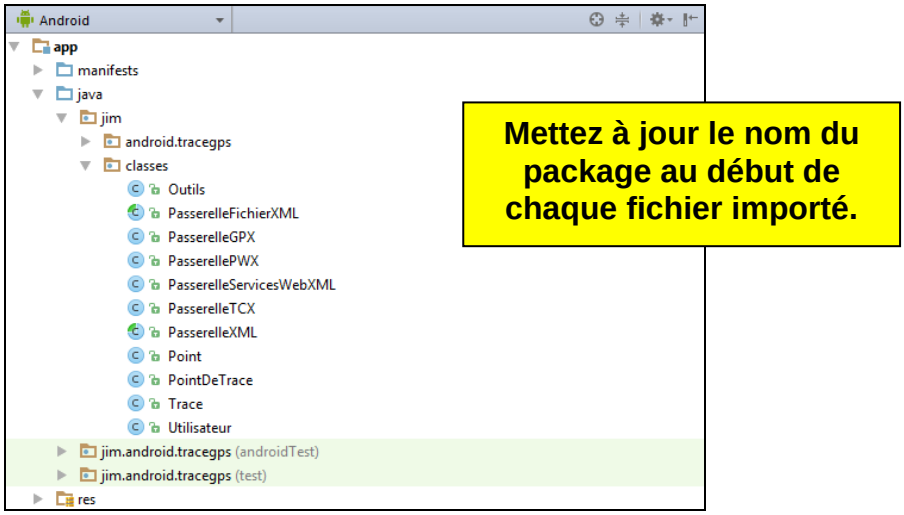
Avant de les importer, ajoutez un package **xxx.classes** (ou **xxx** représente votre alias) dans le dossier **...\app\src\main\java** (clic droit sur le nœud **java** et commande **New / Package**) :



Android Studio ne possède pas de commande pour importer des fichiers de code source. Cette opération se fera donc avec **l'explorateur Windows**. Recopiez vos classes dans le dossier :

votre_workspace/TraceGPS/app/src/main/java/votre_alias/classes

Sous Android Studio, la structure du projet devient :



⚠ Vous penserez à indiquer l'adresse choisie pour l'hébergeur des services web en mettant à jour l'attribut **_adresseHebergeur** dans la classe **PasserelleServicesWebXML.java**

4- Modification du manifeste

Une fois le projet créé, ouvrez le fichier ***AndroidManifest.xml*** et rajoutez les 2 lignes suivantes pour autoriser l'accès Internet (pour appeler les services web) :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="jim.android.tracegps">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        android:usesCleartextTraffic="true">
        <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>
```

La première ligne à ajouter (respectez bien l'écriture)

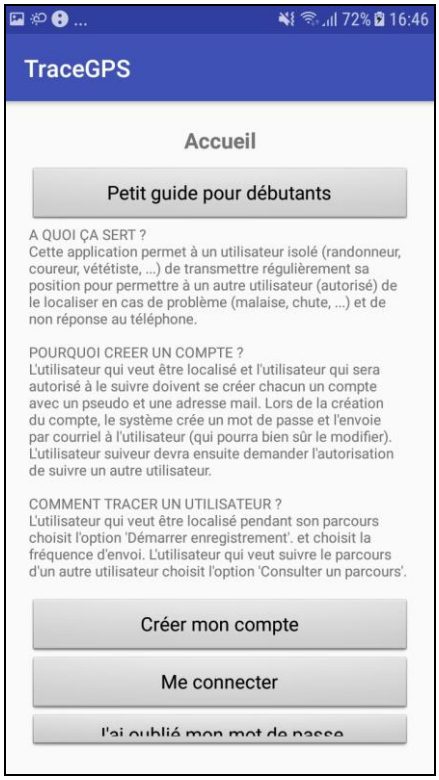
La seconde ligne à ajouter (à partir de la version 9 d'Android)

La liste des activités sera automatiquement complétée ici quand on créera les autres activités du projet

5- Création des fichiers ressources

5-1 Modification du fichier strings.xml

L'interface graphique à créer :



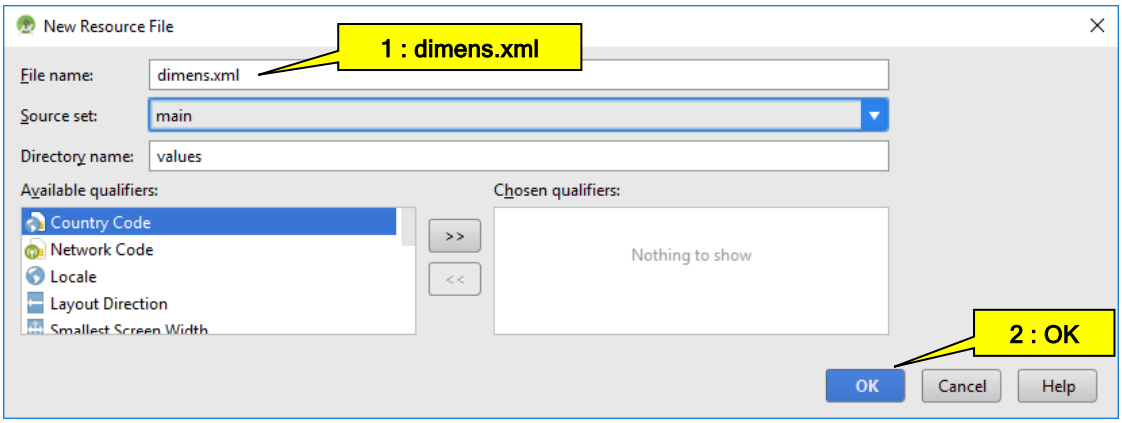
Dans le dossier **res/values**, complétez le fichier **strings.xml** pour préparer le développement de la première vue (**MainActivity**).

Chaque activité possèdera un paragraphe contenant les différents textes utilisés dans la vue.
Les identifiants doivent être uniques dans l'application, on utilise un préfixe représentatif de chaque vue :

```
<resources>
    <string name="app_name">TraceGPS</string>

    <!-- Texte provisoire pour certains TextView (à modifier par programmation) -->
    <string name="texte_provisoire">Texte provisoire</string>

    <!-- Les titres des activités -->
    <string name="title_activity_main">TraceGPS</string>
    <string name="title_activity_menu">TraceGPS</string>
    <string name="title_activity_creer_utilisateur">TraceGPS</string>
    <string name="title_activity_demander_nouveau_mdp">TraceGPS</string>
    <string name="title_activity_tester_geolocalisation">TraceGPS</string>
    <string name="title_activity_demander_autorisation">TraceGPS</string>
    <string name="title_activity_retirer_autorisation">TraceGPS</string>
    <string name="title_activity_demarrer_enregistrement">TraceGPS</string>
    <string name="title_activity_consulter_parcours">TraceGPS</string>
    <string name="title_activity_supprimer_parcours">TraceGPS</string>
    <string name="title_activity_changer_mdp">TraceGPS</string>
    <string name="title_activity_consulter_utilisateurs">TraceGPS</string>
    <string name="title_activity_supprimer_utilisateur">TraceGPS</string>

    <!-- Les textes de la page d'accueil -->
    <string name="main_titre1">Accueil</string>
    <string name="main_bouton_guide">Petit guide pour débutants</string>
    <string name="main_bouton_creer_compte">Créer mon compte</string>
    <string name="main_creer_compte_saisie_pseudo">Mon pseudo</string>
    <string name="main_creer_compte_saisie_adrmail">Mon adresse mail (obligatoire)</string>
    <string name="main_creer_compte_saisie_numtel">Mon n° de téléphone (facultatif)</string>
    <string name="main_creer_compte_bouton_envoyer">Envoyer</string>
    <string name="main_bouton_connecter">Me connecter</string>
    <string name="main_connecter_saisie_pseudo">Mon pseudo</string>
    <string name="main_connecter_saisie_mdp">Mon mot de passe</string>
    <string name="main_connecter_case_mdp_visible">Afficher le mot de passe</string>
    <string name="main_connecter_case_memoriser_parametres">Mémoriser les paramètres</string>
    <string name="main_connecter_bouton_envoyer">Envoyer</string>
    <string name="main_bouton_mdp_oublie">J\'ai oublié mon mot de passe</string>
    <string name="main_mdp_oublie_message">Pour confirmer que vous souhaitez obtenir un nouveau mot de
passe, saisissez votre pseudo.</string>
    <string name="main_mdp_oublie_saisie_pseudo">Mon pseudo</string>
    <string name="main_mdp_oublie_bouton_envoyer">Envoyer</string>
    <string name="main_bouton_quitter">Quitter l\'application</string>

</resources>
```

5-2 Ajout du fichier dimens.xml

Faites un clic droit dans le répertoire **res/values** et choisissez la commande **New / Values resource file** pour ajouter le fichier **dimens.xml** destinés à contenir les dimensions utilisées par l'application.



Complétez le fichier avec la taille des marges :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="tailleMarges">20dp</dimen>
</resources>
```

6- Création de l'interface graphique

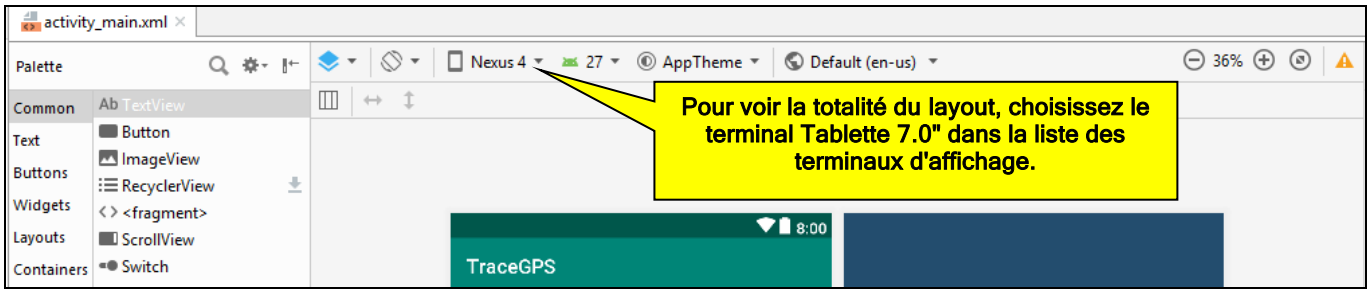
Le fichier **activity_main.xml** du dossier **res/layout** contient la description de l'interface graphique.

Dès la création d'un nouveau projet, l'interface comporte automatiquement un objet **ConstraintLayout** et un objet **TextView**.

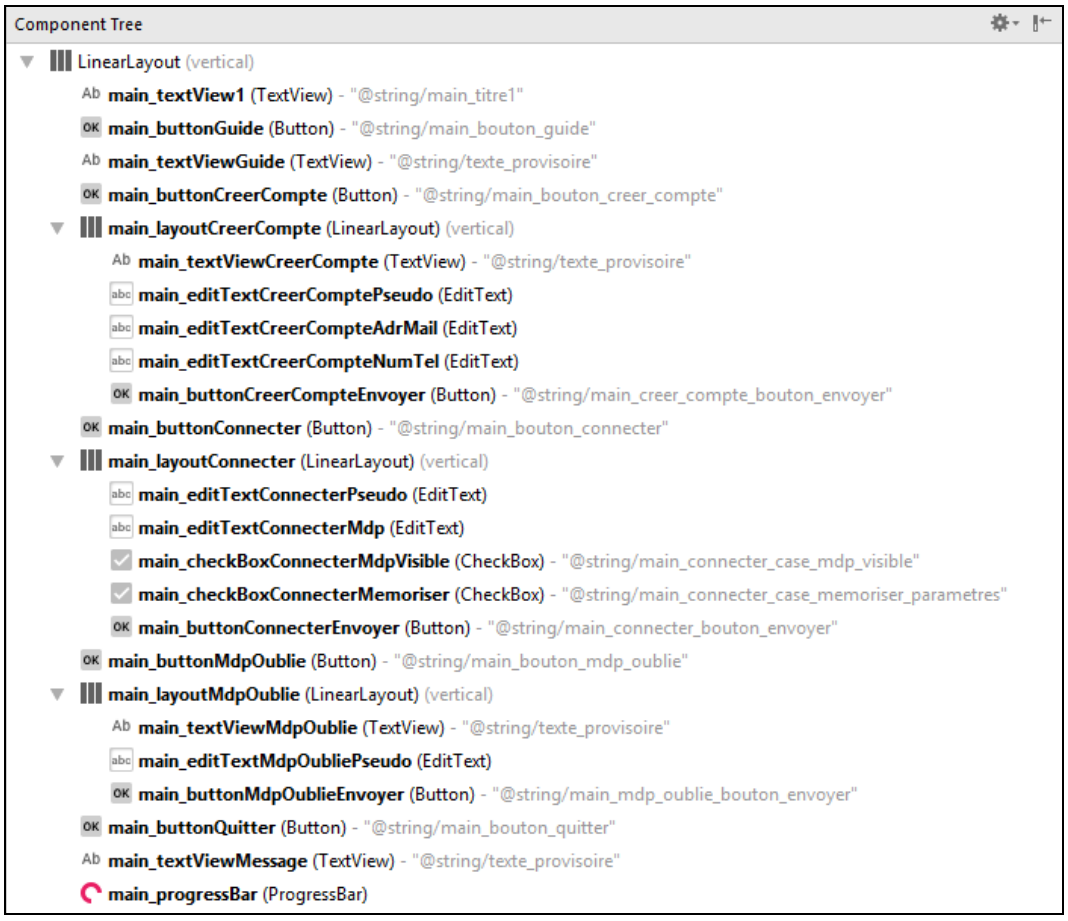
En mode **Text**, vous remplacerez le code existant par le code fourni plus loin afin d'obtenir la structure suivante, dans laquelle on trouve **1 TextView** et **3 LinearLayout** qui seront affichés ou masqués afin d'obtenir un **effet "accordéon"** :

- **main_textViewGuide** : affiché ou masqué avec le bouton **Petit guide pour débutants**
- **main_layoutCreerCompte** : affiché ou masqué avec le bouton **Créer mon compte**
- **main_layoutConnecter** : affiché ou masqué avec le bouton **Connecter**
- **main_layoutMdpOublié** : affiché ou masqué avec le bouton **J'ai oublié mon mot de passe**

La page affichée en mode **Design** occupe beaucoup d'espace en vertical, mais en exécution, elle en occupera moins car sur les 4 éléments précédents, un seul sera affiché à la fois.




8:00
TraceGPS
Accueil
Petit guide pour débutants
Texte provisoire
Créer mon compte
Texte provisoire
Mon pseudo
Mon adresse mail (obligatoire)
Mon n° de téléphone (facultatif)
Envoyer
Me connecter
Mon pseudo
Mon mot de passe
Afficher le mot de passe
Mémoriser les paramètres
Envoyer
J'ai oublié mon mot de passe
Texte provisoire
Mon pseudo
Envoyer
Quitter l'application
Texte provisoire


Component Tree
LinearLayout (vertical)
main_textView1 (TextView) - "@string/main_titre1"
main_buttonGuide (Button) - "@string/main_bouton_guide"
main_textViewGuide (TextView) - "@string/texte_provisoire"
main_buttonCreerCompte (Button) - "@string/main_bouton_creer_compte"
main_layoutCreerCompte (LinearLayout) (vertical)
main_textViewCreerCompte (TextView) - "@string/texte_provisoire"
main_editTextCreerComptePseudo (EditText)
main_editTextCreerCompteAdrMail (EditText)
main_editTextCreerCompteNumTel (EditText)
main_buttonCreerCompteEnvoyer (Button) - "@string/main_creer_compte_bouton_envoyer"
main_buttonConnecter (Button) - "@string/main_bouton_connecter"
main_layoutConnecter (LinearLayout) (vertical)
main_editTextConnecterPseudo (EditText)
main_editTextConnecterMdp (EditText)
main_checkBoxConnecterMdpVisible (CheckBox) - "@string/main_connecter_case_mdp_visible"
main_checkBoxConnecterMemoriser (CheckBox) - "@string/main_connecter_case_memoriser_parametres"
main_buttonConnecterEnvoyer (Button) - "@string/main_connecter_bouton_envoyer"
main_buttonMdpOublie (Button) - "@string/main_bouton_mdp_oublie"
main_layoutMdpOublie (LinearLayout) (vertical)
main_textViewMdpOublie (TextView) - "@string/texte_provisoire"
main_editTextMdpOubliePseudo (EditText)
main_buttonMdpOublieEnvoyer (Button) - "@string/main_mdp_oublie_bouton_envoyer"
main_buttonQuitter (Button) - "@string/main_bouton_quitter"
main_textViewMessage (TextView) - "@string/texte_provisoire"
main_progressBar (ProgressBar)

```
<?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:orientation="vertical"
    android:padding="@dimen/tailleMarges"
    tools:context="jim.android.tracegps.MainActivity">

    <TextView
        android:id="@+id/main_textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="10dp"
        android:text="@string/main_titre1"
        android:textAlignment="center"
        android:textSize="18sp"
        android:textStyle="bold" />

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

    <TextView
        android:id="@+id/main_textViewGuide"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/texte_provisoire"
        android:textSize="12sp" />

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

    <LinearLayout
        android:id="@+id/main_layoutCreerCompte"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="10dp"
        android:orientation="vertical">

        <TextView
            android:id="@+id/main_textViewCreerCompte"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/texte_provisoire"
            android:textSize="14sp" />

        <EditText
            android:id="@+id/main_editTextCreerComptePseudo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/main_creer_compte_saisie_pseudo"
            android:inputType="textPersonName"
            android:textSize="14sp" />

        <EditText
            android:id="@+id/main_editTextCreerCompteAdrMail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="14sp"
            android:hint="@string/main_creer_compte_saisie_adrmail"
```

```
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/main_editTextCreerCompteNumTel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:hint="@string/main_creer_compte_saisie_numtel"
        android:inputType="textEmailAddress" />

    <Button
        android:id="@+id/main_buttonCreerCompteEnvoyer"
        style="@android:style/Widget.Button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="@string/main_creer_compte_bouton_envoyer"
        android:textSize="16sp" />

</LinearLayout>

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

<LinearLayout
    android:id="@+id/main_layoutConnecter"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingBottom="10dp"
    android:orientation="vertical">

    <EditText
        android:id="@+id/main_editTextConnecterPseudo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:hint="@string/main_connecter_saisie_pseudo"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/main_editTextConnecterMdp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:hint="@string/main_connecter_saisie_mdp"
        android:inputType="textPassword" />

    <CheckBox
        android:id="@+id/main_checkBoxConnecterMdpVisible"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/main_connecter_case_mdp_visible"
        android:textSize="14sp" />

    <CheckBox
        android:id="@+id/main_checkBoxConnecterMemoriser"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/main_connecter_case_memoriser_parametres"
        android:textSize="14sp" />

    <Button
        android:id="@+id/main_buttonConnecterEnvoyer"
        style="@android:style/Widget.Button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_gravity="center_horizontal"
        android:text="@string/main_connecter_bouton_envoyer"
        android:textSize="16sp" />
    </LinearLayout>
    <Button
      android:id="@+id/main_buttonMdpOublie"
      style="@android:style/Widget.Button"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:textSize="16sp"
      android:text="@string/main_bouton_mdp_oublie" />
    <LinearLayout
      android:id="@+id/main_layoutMdpOublie"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:paddingBottom="10dp"
      android:orientation="vertical">
      <TextView
        android:id="@+id/main_textViewMdpOublie"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/texte_provisoire"
        android:textSize="14sp" />
      <EditText
        android:id="@+id/main_editTextMdpOubliePseudo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:hint="@string/main_mdp_oublie_saisie_pseudo"
        android:inputType="textPersonName" />
      <Button
        android:id="@+id/main_buttonMdpOublieEnvoyer"
        style="@android:style/Widget.Button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="@string/main_mdp_oublie_bouton_envoyer"
        android:textSize="16sp" />
    </LinearLayout>
    <Button
      android:id="@+id/main_buttonQuitter"
      style="@android:style/Widget.Button"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="@string/main_bouton_quitter"
      android:textSize="16sp" />
    <TextView
      android:id="@+id/main_textViewMessage"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="@string/texte_provisoire"
      android:textSize="14sp" />
    <ProgressBar
      android:id="@+id/main_progressBar"
      style="?android:attr/progressBarStyle"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center_horizontal" />
</LinearLayout>
```

7- Programmation Java de l'activité MainActivity.java

7-1 Déclarations diverses et initialisation des objets

Dans le fichier **MainActivity.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.EditText;
import android.widget.TextView;
import android.widget.CheckBox;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.Toast;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    // les zones de saisie et les variables associées
    private EditText editTextConnecterPseudo;
    private EditText editTextConnecterMdp;
    private EditText editMdpOubliePseudo;
    private EditText editTextCreerComptePseudo;
    private EditText editTextCreerCompteAdrMail;
    private EditText editTextCreerCompteNumTel;
    private String pseudo;
    private String mdp;
    private String typeUtilisateur;      // "utilisateur" ou "administrateur"
    private String adrMail;
    private String numTel;
    // les cases à cocher
    private CheckBox caseMdpVisible;
    private CheckBox caseMemoriser;
    // les boutons
    private Button buttonGuide;
    private Button buttonCreerCompte;
    private Button buttonCreerCompteEnvoyer;
    private Button buttonConnecter;
    private Button buttonConnecterEnvoyer;
    private Button buttonMdpOublie;
    private Button buttonMdpOublieEnvoyer;
    private Button buttonQuitter;
    // les layouts
    private LinearLayout layoutConnecter;
    private LinearLayout layoutCreerCompte;
    private LinearLayout layoutMdpOublie;
    // le ProgressBar pour afficher le cercle de chargement
    private ProgressBar progressBar;
    // les zones d'affichage de message
    private TextView textViewGuide;
    private TextView textViewCreerCompte;
    private TextView textViewMdpOublie;
    private TextView textViewMessage;

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

```
        // récupération des objets XML grâce à leur ID
        layoutConnecter = (LinearLayout) findViewById(R.id.main_layoutConnecter);
        layoutMdpOublie = (LinearLayout) findViewById(R.id.main_layoutMdpOublie);
        layoutCreerCompte = (LinearLayout) findViewById(R.id.main_layoutCreerCompte);

        editTextConnecterPseudo = (EditText) findViewById(R.id.main_editTextConnecterPseudo);
        editTextConnecterMdp = (EditText) findViewById(R.id.main_editTextConnecterMdp);
        editMdpOubliePseudo = (EditText) findViewById(R.id.main_editTextMdpOubliePseudo);
        editTextCreerComptePseudo = (EditText) findViewById(R.id.main_editTextCreerComptePseudo);
        editTextCreerCompteAdrMail = (EditText) findViewById(R.id.main_editTextCreerCompteAdrMail);
        editTextCreerCompteNumTel = (EditText) findViewById(R.id.main_editTextCreerCompteNumTel);

        caseMemoriser = (CheckBox) findViewById(R.id.main_checkBoxConnecterMemoriser);
        caseMemoriser.setChecked(true);       // case cochée au démarrage

        caseMdpVisible = (CheckBox) findViewById(R.id.main_checkBoxConnecterMdpVisible);
        caseMdpVisible.setChecked(false);  // case décochée au démarrage

        buttonGuide = (Button) findViewById(R.id.main_buttonGuide);
        buttonCreerCompte = (Button) findViewById(R.id.main_buttonCreerCompte);
        buttonCreerCompteEnvoyer = (Button) findViewById(R.id.main_buttonCreerCompteEnvoyer);
        buttonConnecter = (Button) findViewById(R.id.main_buttonConnecter);
        buttonConnecterEnvoyer = (Button) findViewById(R.id.main_buttonConnecterEnvoyer);
        buttonMdpOublie = (Button) findViewById(R.id.main_buttonMdpOublie);
        buttonMdpOublieEnvoyer = (Button) findViewById(R.id.main_buttonMdpOublieEnvoyer);
        buttonQuitter = (Button) findViewById(R.id.main_buttonQuitter);

        textViewGuide = (TextView) findViewById(R.id.main_textViewGuide);
        textViewCreerCompte = (TextView) findViewById(R.id.main_textViewCreerCompte);
        textViewMdpOublie = (TextView) findViewById(R.id.main_textViewMdpOublie);
        textViewMessage = (TextView) findViewById(R.id.main_textViewMessage);

        progressBar = (ProgressBar) findViewById(R.id.main_progressBar);

        // masque le cercle de chargement et certains éléments
        progressBar.setVisibility(View.GONE);
        layoutConnecter.setVisibility(View.GONE);
        layoutMdpOublie.setVisibility(View.GONE);
        layoutCreerCompte.setVisibility(View.GONE);
        textViewGuide.setVisibility(View.GONE);
        textViewMessage.setText("");

        // association d'un écouteur d'évenement à chaque bouton
        buttonGuide.setOnClickListener ( new buttonGuideClickListener());
        buttonCreerCompte.setOnClickListener ( new buttonCreerCompteClickListener());
        buttonCreerCompteEnvoyer.setOnClickListener ( new buttonCreerCompteEnvoyerClickListener());
        buttonConnecter.setOnClickListener ( new buttonConnecterClickListener());
        buttonConnecterEnvoyer.setOnClickListener ( new buttonConnecterEnvoyerClickListener());
        buttonMdpOublie.setOnClickListener ( new buttonMdpOublieClickListener());
        buttonMdpOublieEnvoyer.setOnClickListener ( new buttonMdpOublieEnvoyerClickListener());
        buttonQuitter.setOnClickListener ( new buttonQuitterClickListener());

        // association d'un écouteur d'évenement à la case caseMdpVisible
        caseMdpVisible.setOnClickListener ( new caseMdpVisibleClickListener());
    }

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

    /** classe interne pour gérer le clic sur le bouton buttonCreerCompte. */
    private class buttonCreerCompteClickListener implements View.OnClickListener {
        public void onClick(View v) {
```

```
    }
  }

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

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

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

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

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

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

  /** classe interne pour gérer le clic sur la case caseMdpVisible. */
  private class caseMdpVisibleClickListener implements View.OnClickListener {
    public void onClick(View v) {
    }
  }
} // fin de l'activité
```

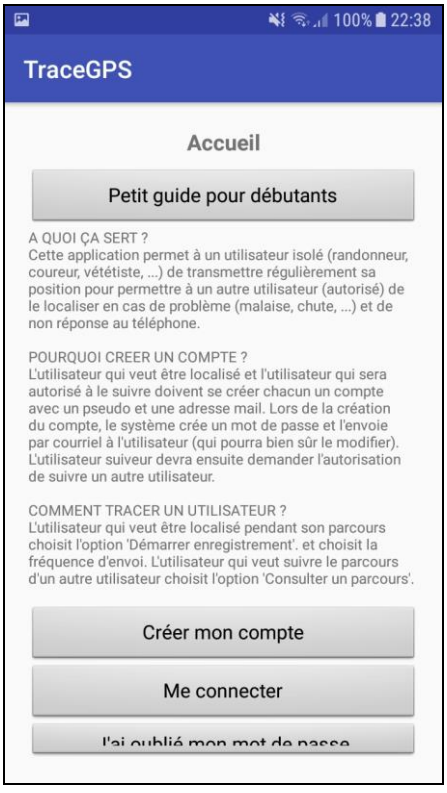
Testez l'application et le bon fonctionnement du bouton **Quitter l'application** (qui est le seul à être programmé pour l'instant).

7-2 Gestion du bouton Guide

Complétez l'écouteur **buttonGuideClickListener** :

```
/** classe interne pour gérer le clic sur le bouton buttonGuide. */
private class buttonGuideClickListener implements View.OnClickListener {
   public void onClick(View v) {
       layoutConnecter.setVisibility(View.GONE);
       layoutCreerCompte.setVisibility(View.GONE);
       layoutMdpOublie.setVisibility(View.GONE);
       textViewMessage.setText("");

       if (textViewGuide.getVisibility() == View.GONE)
          textViewGuide.setVisibility(View.VISIBLE);  // afficher le texte explicatif
       else
          textViewGuide.setVisibility(View.GONE);     // masquer le texte explicatif

       String msg = "A QUOI ÇA SERT ?\n";
       msg += "Cette application permet à un utilisateur isolé (randonneur, coureur, vététiste, ...) de transmettre régulièrement sa position pour permettre";
       msg += " à un autre utilisateur (autorisé) de le localiser en cas de problème (malaise, chute, ...) et de non réponse au téléphone.\n\n";
       msg += "POURQUOI CREER UN COMPTE ?\n";
       msg +="L'utilisateur qui veut être localisé et l'utilisateur qui sera autorisé à le suivre doivent se créer chacun un compte avec un pseudo et une adresse mail.";
       msg += " Lors de la création du compte, le système crée un mot de passe et l'envoie par courriel à l'utilisateur (qui pourra bien sûr le modifier).";
       msg += " L'utilisateur suiveur devra ensuite demander l'autorisation de suivre un autre utilisateur.\n\n";
       msg += "COMMENT TRACER UN UTILISATEUR ?\n";
       msg += "L'utilisateur qui veut être localisé pendant son parcours choisit l'option 'Démarrer enregistrement'. et choisit la fréquence d'envoi.";
       msg += " L'utilisateur qui veut suivre le parcours d'un autre utilisateur choisit l'option 'Consulter un parcours'.\n";
       textViewGuide.setText(msg);
   }
}
```

Testez l'application et le bon fonctionnement du bouton **Petit guide pour débutants**.



7-3 Gestion du bouton Créer mon compte

On fait appel ici au service web **CreerUtilisateur** pour créer le compte utilisateur.

Commencez par ajouter les **import** suivants à la suite des **import** existants :

```
import jim.classes.*;
import android.os.AsyncTask;
```

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

```
/** classe interne pour gérer le clic sur le bouton buttonCreerCompte. */
private class buttonCreerCompteClickListener implements View.OnClickListener {
    public void onClick(View v) {
        layoutConnecter.setVisibility(View.GONE);
        layoutMdpOublie.setVisibility(View.GONE);
        textViewGuide.setVisibility(View.GONE);
        textViewMessage.setText("");
        String msg = "Choisissez un pseudo (au moins 8 caractères), indiquez votre adresse mail";
        msg += " (obligatoire) votre n° de téléphone (facultatif) et validez avec le bouton Envoyer.";
        textViewCreerCompte.setText(msg);

        if (layoutCreerCompte.getVisibility() == View.GONE)
            layoutCreerCompte.setVisibility(View.VISIBLE);    // afficher le layout
        else
            layoutCreerCompte.setVisibility(View.GONE);       // masquer le layout
    }
}
```

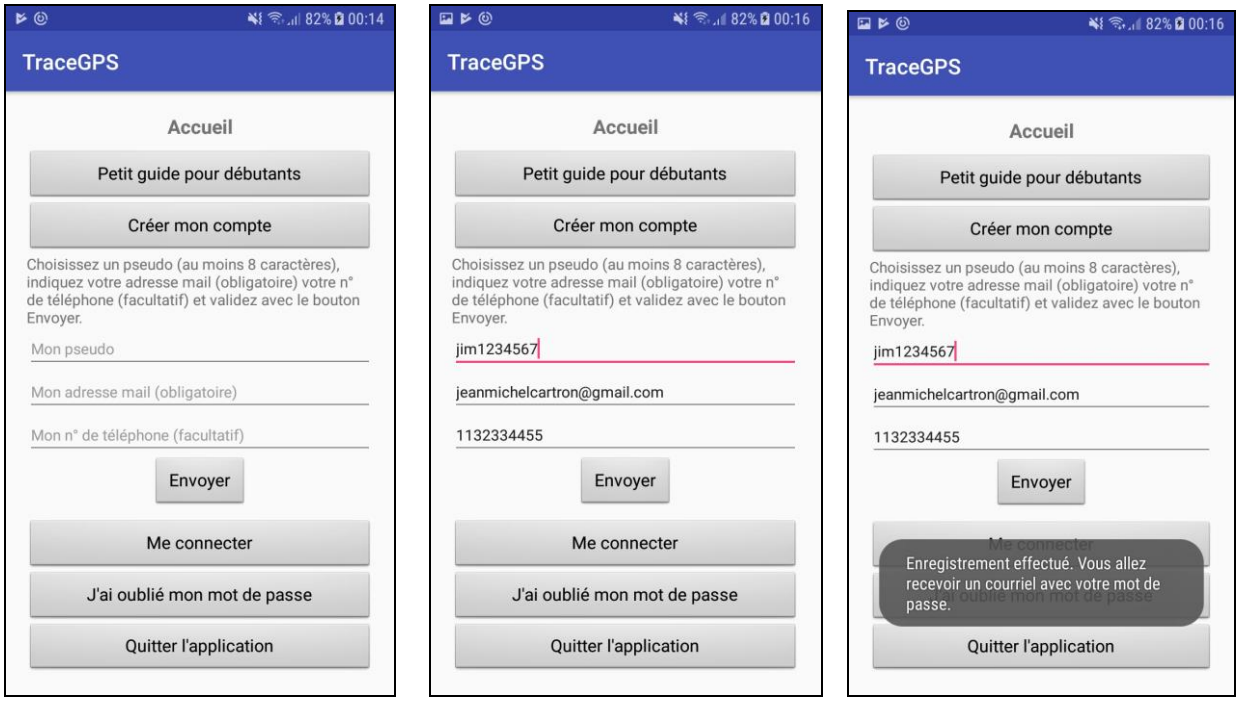
Complétez l'écouteur d'événement associé à **buttonCreerCompteEnvoyer** pour appeler la tâche asynchrone **TacheCreerUtilisateur** :

```
/** classe interne pour gérer le clic sur le bouton buttonCreerCompteEnvoyer. */
private class buttonCreerCompteEnvoyerClickListener implements View.OnClickListener {
    public void onClick(View v) {
        // récupération des données saisies
        pseudo = editTextCreerComptePseudo.getText().toString().trim();
        adrMail = editTextCreerCompteAdrMail.getText().toString().trim();
        numTel = editTextCreerCompteNumTel.getText().toString().trim();

        // test des données obligatoires
        if ( pseudo.equals("") || adrMail.equals("") ) {
            String msg = "Données incomplètes";
            textViewMessage.setText(msg);
            Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_LONG).show();
        }
        else {
            // appel du service web CreerUtilisateur avec une tâche asynchrone
            new TacheCreerUtilisateur().execute();
        }
    }
}
```

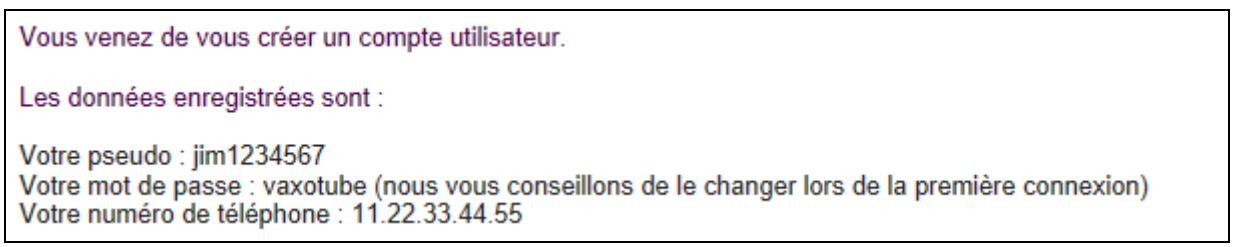
Ajoutez la tâche asynchrone **TacheCreerUtilisateur** juste avant la fin de l'activité :

```
// ------------------------------------------------------------------------------------------------------
// ---------------------------- Tâche asynchrone TacheCreerUtilisateur ----------------------------------
// ------------------------------------------------------------------------------------------------------

// appel du service web CreerUtilisateur
private class TacheCreerUtilisateur extends AsyncTask<Void, Void, String> {
    @Override
    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(Void... params) {
        // cette méthode permet de lancer l'exécution de la tâche longue
        String msg = PasserelleServicesWebXML.creerUnUtilisateur(pseudo, adrMail, numTel);
        return msg;
    }
    protected void onPostExecute(String msg) {
        // cette méthode est automatiquement appelée quand la tâche longue se termine
        textViewMessage.setText(msg);
        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_LONG).show();
        progressBar.setVisibility(View.GONE);  // arrête le cercle de chargement
    }
} // fin tâche asynchrone TacheCreerUtilisateur
```

Testez cette étape sur un mobile réel et corrigez les erreurs si besoin :



Vérifiez qu'un courriel similaire à celui-ci est bien envoyé avec le mot de passe :



7-4 Gestion du bouton "Me connecter" et de l'appel du service web

On fait appel ici au service web **Connecter** pour vérifier l'authentification de l'utilisateur.

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

```
/** classe interne pour gérer le clic sur le bouton buttonConnecter. */
private class buttonConnecterClickListener implements View.OnClickListener {
    public void onClick(View v) {
        layoutMdpOublie.setVisibility(View.GONE);
        layoutCreerCompte.setVisibility(View.GONE);
        textViewGuide.setVisibility(View.GONE);
         textViewMessage.setText("");

        if (layoutConnecter.getVisibility() == View.GONE)
            layoutConnecter.setVisibility(View.VISIBLE);    // afficher le layout
        else
            layoutConnecter.setVisibility(View.GONE);       // masquer le layout
    }
}
```

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

```
/** classe interne pour gérer le clic sur le bouton buttonConnecterEnvoyer. */
private class buttonConnecterEnvoyerClickListener implements View.OnClickListener {
    public void onClick(View v) {
        pseudo = editTextConnecterPseudo.getText().toString().trim();
        mdp = editTextConnecterMdp.getText().toString().trim();
        // appel du service web Connecter avec une tâche asynchrone
        new TacheConnecter().execute();
    }
}
```

Ajoutez la tâche **TacheConnecter** juste avant la fin de l'activité :

```
// ------------------------------------------------------------------------------------------------------
// ---------------------------------- Tâche asynchrone TacheConnecter -----------------------------------
// ------------------------------------------------------------------------------------------------------

// appel du service web Connecter
private class TacheConnecter extends AsyncTask<Void, 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(Void... params) {
        // cette méthode permet de lancer l'exécution de la tâche longue
        String msg = PasserelleServicesWebXML.connecter(pseudo, Outils.sha1(mdp));
        return msg;
    }
    protected void onPostExecute(String msg) {
        // cette méthode est automatiquement appelée quand la tâche longue se termine
        progressBar.setVisibility(View.GONE);// arrête le cercle de chargement

        if (msg.startsWith("Erreur")) {
            textViewMessage.setText(msg);
            Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_LONG).show();
        }
        else {
            // détermine le type d'utilisateur
            if (msg.startsWith("Utilisateur authentifié")) typeUtilisateur = "utilisateur";
            if (msg.startsWith("Administrateur authentifié")) typeUtilisateur = "administrateur";

            // traitement provisoire
            textViewMessage.setText(msg);
        }
    }
} // fin tâche asynchrone TacheConnecter
```

Ces 2 lignes provisoires seront remplacées par l'appel de l'activité MenuGeneral (quand celle-ci sera développée).

Testez cette étape sur un mobile réel et corrigez les erreurs si besoin.

Voici 3 exemples de captures d'écran illustrant cette activité :

7-5 Gestion de la mémorisation des paramètres

Cette option permet à l'application de mémoriser les paramètres de connexion pour ne pas les ressaisir à la prochaine exécution.

Commencez par ajouter les **import** suivants à la suite des **import** existants :

```
import android.content.SharedPreferences;
```

Ajoutez les **déclarations** suivantes à la suite des **déclarations** existantes :

```
// la mémorisation des données se fait au moyen de couples clé/valeur :
private final String KEY_PSEUDO = "pseudo";
private final String KEY_MDP = "mdp";
```

Ajoutez le code suivant à la fin de la fonction **onCreate** :

```
// Récupération des données de connexion dans le cas où elles ont été mémorisées lors de la dernière connexion
SharedPreferences mesDonnees = getPreferences(MODE_PRIVATE);
pseudo = mesDonnees.getString(KEY_PSEUDO, "");
mdp = mesDonnees.getString(KEY_MDP, "");
// si des données avaient été mémorisées, elles sont proposées dans les zones de saisies
if ( ! pseudo.equals("") ) editTextConnecterPseudo.setText(pseudo);
if ( ! mdp.equals("") ) editTextConnecterMdp.setText(mdp);
```

Complétez l'écouteur **buttonConnecterEnvoyerClickListener** :

```
/** classe interne pour gérer le clic sur le bouton buttonConnecterEnvoyer. */
private class buttonConnecterEnvoyerClickListener implements View.OnClickListener {
    public void onClick(View v) {
        pseudo = editTextConnecterPseudo.getText().toString().trim();
        mdp = editTextConnecterMdp.getText().toString().trim();

        // Mémorisation des données de connexion si la case est cochée
        SharedPreferences mesDonnees = getPreferences(MODE_PRIVATE);
        SharedPreferences.Editor monEditor = mesDonnees.edit();
        if (caseMemoriser.isChecked()) {
            monEditor.putString(KEY_PSEUDO, pseudo);
            monEditor.putString(KEY_MDP, mdp);
        }
        else {
            monEditor.putString(KEY_PSEUDO, "");
            monEditor.putString(KEY_MDP, "");
        }
        monEditor.commit();

        // appel du service web Connecter avec une tâche asynchrone
        new TacheConnecter().execute();
    }
}
```

Testez cette étape sur un mobile réel et corrigez les erreurs si besoin.

Exemple de connexion sans mémorisation lors de l'exécution précédente

Exemple de connexion avec mémorisation lors de l'exécution précédente

7-6 Gestion de la case à cocher permettant d'afficher (ou non) le mot de passe

Complétez l'écouteur **caseMdpVisibleClickListener** pour modifier le type d'affichage de l'objet **editTextConnecterMdp** :

```
/** classe interne pour gérer le clic sur la case caseMdpVisible. */
private class caseMdpVisibleClickListener implements View.OnClickListener {
    public void onClick(View v) {
        if (caseMdpVisible.isChecked())
            editTextConnecterMdp.setInputType(android.text.InputType.TYPE_CLASS_TEXT |
                                              android.text.InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD);
        else
            editTextConnecterMdp.setInputType(android.text.InputType.TYPE_CLASS_TEXT |
                                              android.text.InputType.TYPE_TEXT_VARIATION_PASSWORD);
    }
```

Testez cette étape sur un mobile réel et corrigez les erreurs si besoin :

7-7 Gestion du bouton "J'ai oublié mon mot de passe"

On fait appel ici au service web **DemanderMdp** pour recevoir un courriel contenant le mot de passe.

Pour éviter de demander par erreur un nouveau mot de passe, l'utilisateur doit saisir son pseudo dans la zone de saisie.

Complétez l'écouteur **buttonMdpOublieClickListener** :

```
/** classe interne pour gérer le clic sur le bouton buttonMdpOublie. */
private class buttonMdpOublieClickListener implements View.OnClickListener {
    public void onClick(View v) {
        layoutCreerCompte.setVisibility(View.GONE);
        layoutConnecter.setVisibility(View.GONE);
        textViewGuide.setVisibility(View.GONE);
        textViewMessage.setText("");
        String msg = "Pour confirmer que vous souhaitez obtenir un nouveau mot de passe, ";
        msg += "saisissez votre pseudo.";
        textViewMdpOublie.setText(msg);

        if (layoutMdpOublie.getVisibility() == View.GONE)
            layoutMdpOublie.setVisibility(View.VISIBLE);    // afficher le layout
        else
            layoutMdpOublie.setVisibility(View.GONE);       // masquer le layout
    }
}
```

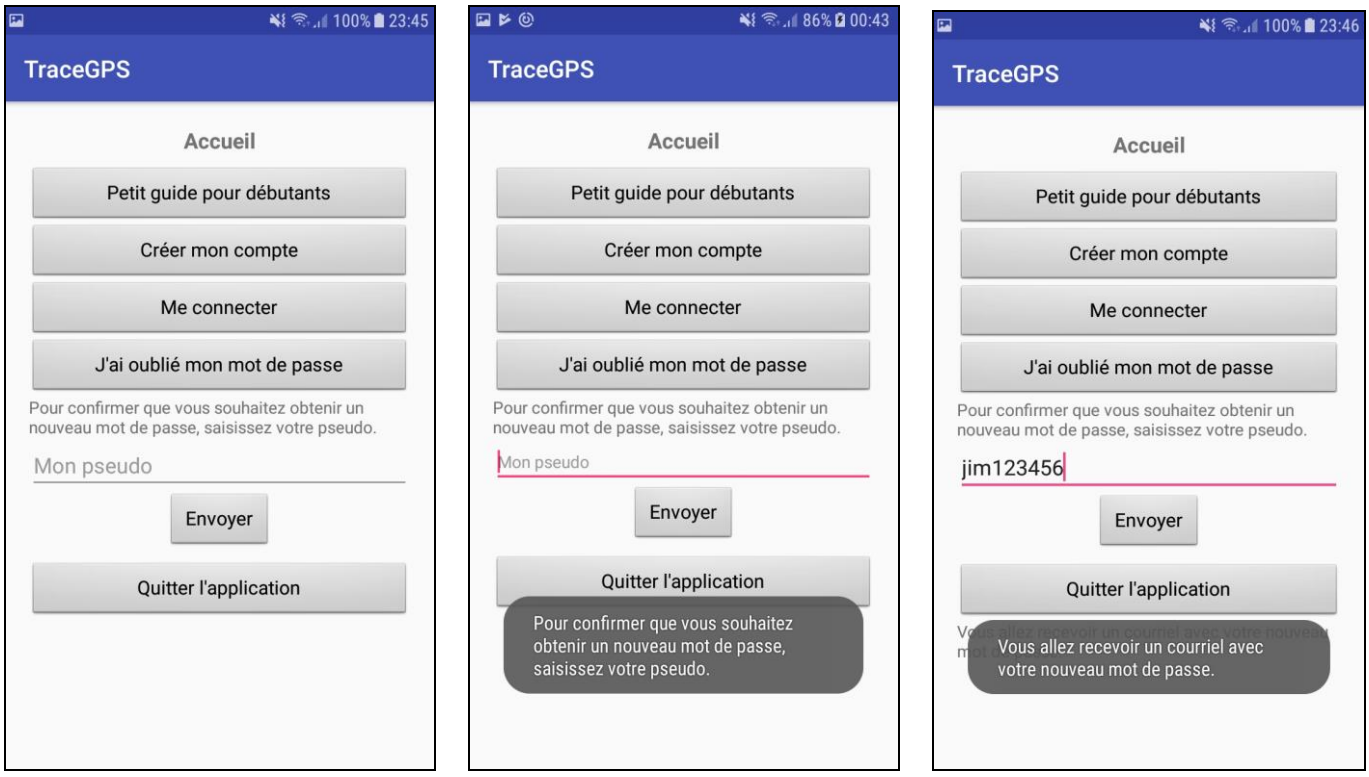
Complétez l'écouteur **buttonMdpOublieEnvoyerClickListener** pour appeler la tâche asynchrone **TacheDemanderMdp** :

```
/** classe interne pour gérer le clic sur le bouton buttonMdpOublieEnvoyer. */
private class buttonMdpOublieEnvoyerClickListener implements View.OnClickListener {
    public void onClick(View v) {
        pseudo = editMdpOubliePseudo.getText().toString();
        if ( pseudo.equals("")) {
            String msg = "Pour confirmer que vous souhaitez obtenir un nouveau mot de passe, ";
            msg += "saisissez votre pseudo.";
            Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_LONG).show();
            editMdpOubliePseudo.setHint("Mon pseudo");
        }
        else {
            // appel du service web DemanderMdp avec une tâche asynchrone
            new TacheDemanderMdp().execute();
        }
    }
}
```

Ajoutez la tâche asynchrone **TacheDemanderMdp** juste avant la fin de l'activité :

```
// ----------------------------------------------------------------------------------------------------
// -------------------------------- Tâche asynchrone TacheDemanderMdp ---------------------------------
// ----------------------------------------------------------------------------------------------------

// appel du service web DemanderMdp
private class TacheDemanderMdp extends AsyncTask<Void, 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(Void... params) {
        // cette méthode permet de lancer l'exécution de la tâche longue
        String msg = PasserelleServicesWebXML.demanderMdp(pseudo);
        return msg;
    }
    protected void onPostExecute(String msg) {
        // cette méthode est automatiquement appelée quand la tâche longue se termine
        textViewMessage.setText(msg);
        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_LONG).show();

        progressBar.setVisibility(View.GONE);        // arrête le cercle de chargement
    }
} // fin tâche asynchrone TacheDemanderMdp
```

Testez cette étape sur un mobile réel et corrigez les erreurs si besoin :



Vérifiez qu'un courriel est bien envoyé avec le nouveau mot de passe :

Cher(chère) jim123456

Votre mot de passe d'accès au service service TraceGPS a été modifié.

Votre nouveau mot de passe est : gexyzafy