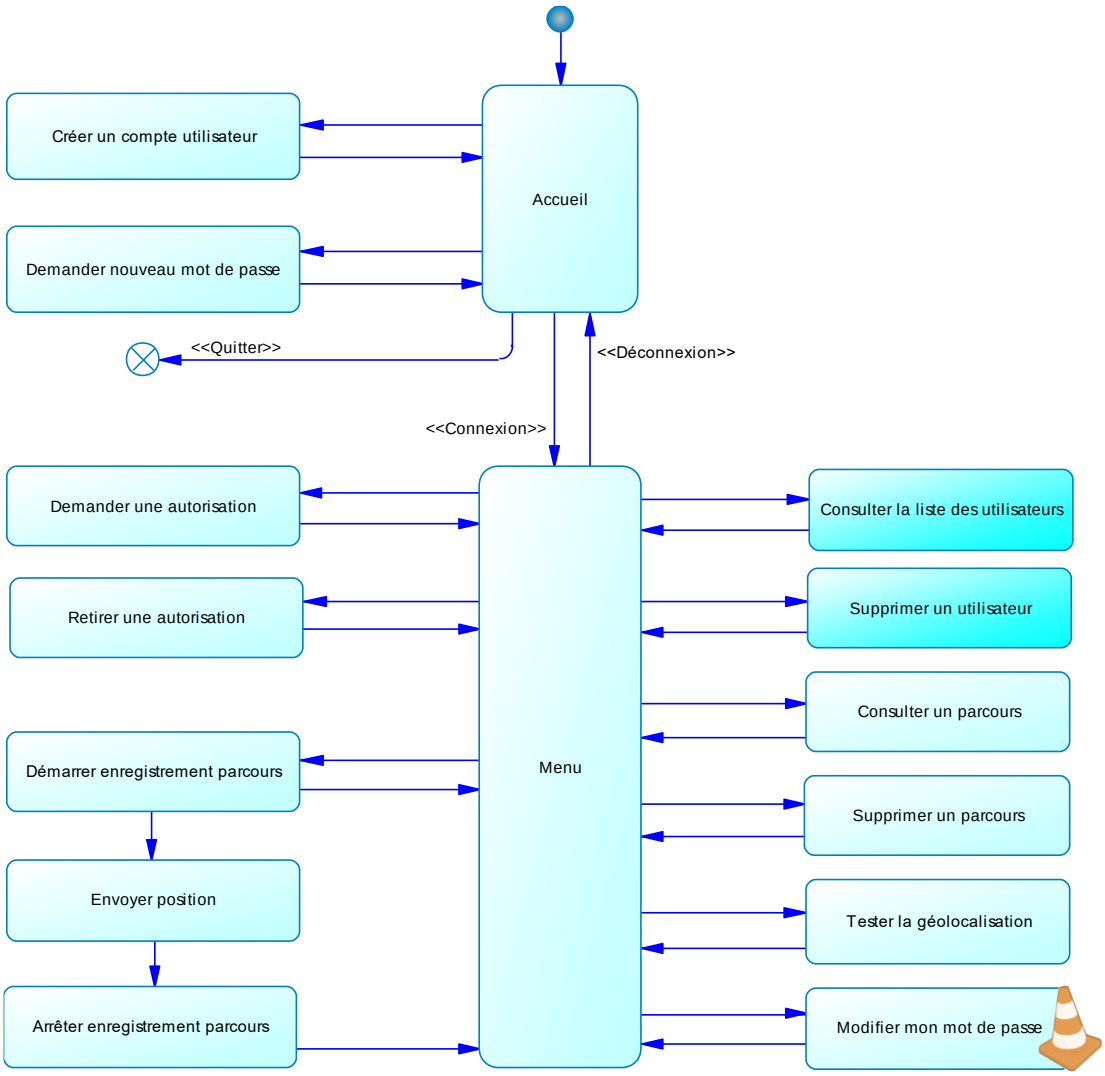
5- Développement de l'application mobile Android

5-3 ChangerMdp (page de changement de mot de passe)

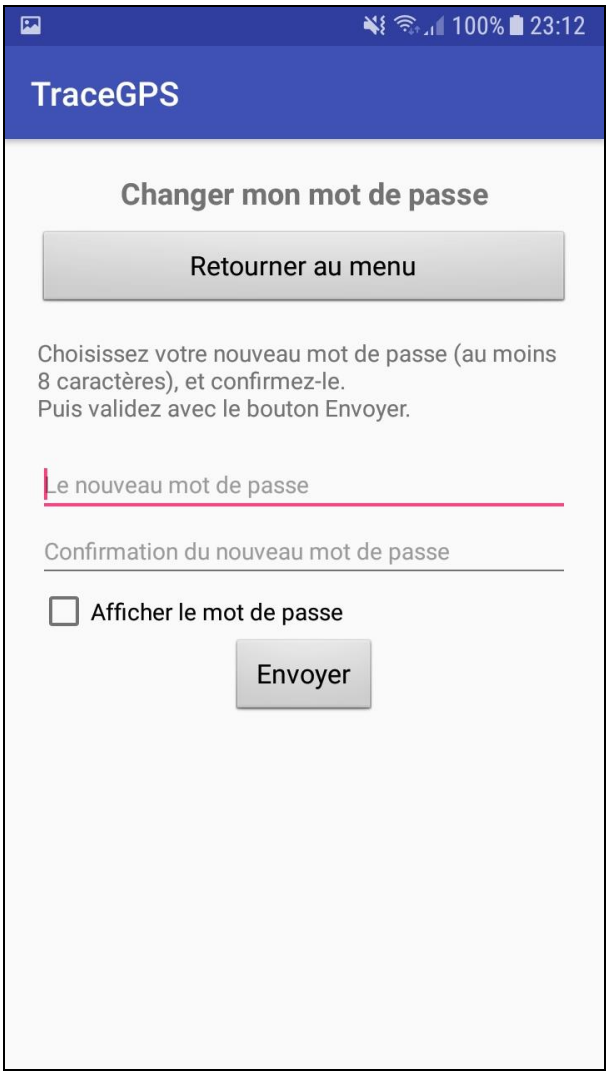
1- Situation de l'activité dans la structure de l'application
2- Modification du fichier strings.xml
3- Création de l'activité
4- Création de l'interface graphique
5- Modification de la programmation Java de MenuGeneral.java
6- Programmation Java de l'activité ChangerMdp.java

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



2- Modification du fichier strings.xml

L'interface graphique à créer :

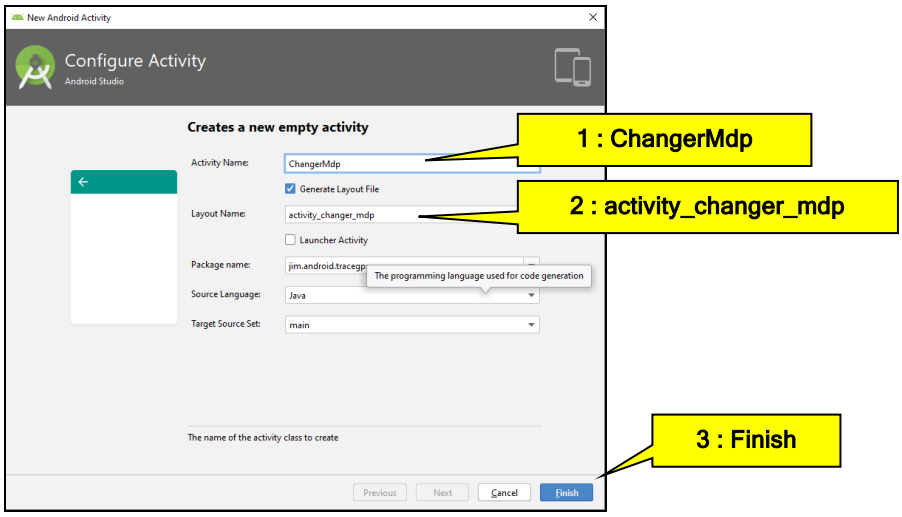


Dans le dossier **res/values**, complétez le fichier **strings.xml** avec le code suivant :

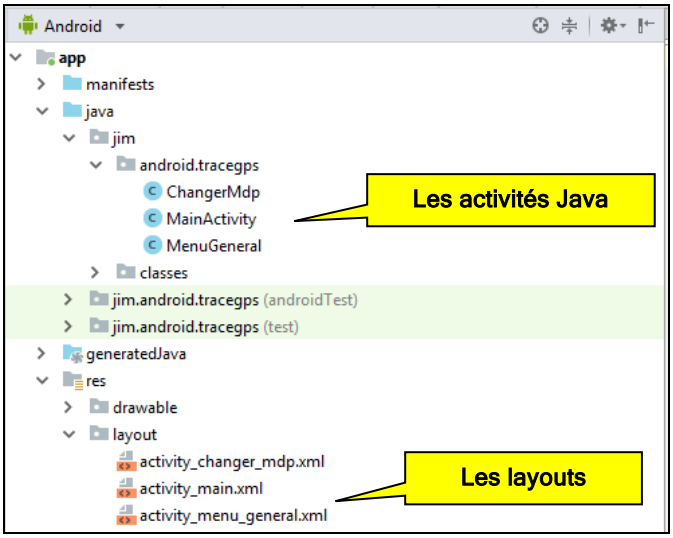
```
<!-- Les textes de la page de changement du mot de passe -->
<string name="changer_mdp_titre1">Changer mon mot de passe</string>
<string name="changer_mdp_bouton_retourner">Retourner au menu</string>
<string name="changer_mdp_saisie_entrer">Le nouveau mot de passe</string>
<string name="changer_mdp_saisie_confirmer">Confirmation du nouveau mot de passe</string>
<string name="changer_mdp_case_mdp_visible">Afficher le mot de passe</string>
<string name="changer_mdp_bouton_envoyer">Envoyer</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 **ChangerMdp.java** et le layout **activity_changer_mdp.xml** sont alors créés :



On peut constater que le fichier **AndroidManifest.xml** a été automatiquement complété :

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme"
    android:usesCleartextTraffic="true">
    <activity android:name=".ChangerMdp"></activity>
    <activity android:name=".MenuGeneral"></activity>
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

L'activité ajoutée

L'activité principale

4- Création de l'interface graphique

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="jim.android.tracegps.ChangerMdp">

</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

Component Tree

- LinearLayout (vertical)
  - **changer_mdp_textView1** (TextView) - "@string/changer_mdp_titre1"
  - **changer_mdp_buttonRetourner** (Button) - "@string/changer_mdp_bouton_retourner"
  - **changer_mdp_textViewGuide** (TextView) - "@string/texte_provisoire"
  - **changer_mdp_editTextMdp** (EditText)
  - **changer_mdp_editTextConfirmationMdp** (EditText)
  - **changer_mdp_checkBoxMdpVisible** (CheckBox) - "@string/changer_mdp_case_mdp_visible"
  - **changer_mdp_buttonEnvoyer** (Button) - "@string/changer_mdp_bouton_envoyer"
  - **changer_mdp_textViewMessage** (TextView) - "@string/texte_provisoire"
  - **changer_mdp_progressBar** (ProgressBar)

Le code XML de l'activité :

```
<?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.ChangerMdp">

   <TextView
      android:id="@+id/changer_mdp_textView1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:paddingBottom="10dp"
      android:text="@string/changer_mdp_titre1"
      android:textAlignment="center"
      android:textSize="18sp"
      android:textStyle="bold" />

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

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

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

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

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

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

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

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

</LinearLayout >
```

5- Modification de la programmation Java de MenuGeneral.java

La page de menu doit permettre d'appeler la page de changement de mot de passe, **mais elle doit ensuite récupérer le nouveau mot de passe** lors du retour au menu.

La connaissance du nouveau mot de passe est importante pour pouvoir continuer à appeler les différents services web.

Commencez par ajouter les déclarations suivantes à la suite des déclarations existantes :

```
// codes utilisés avec la méthode startActivityForResult
// ces nombres peuvent être quelconques, mais doivent être différents les uns des autres
private final int CODE_RESULTAT_CHANGEMENT_MDP = 1;
```

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

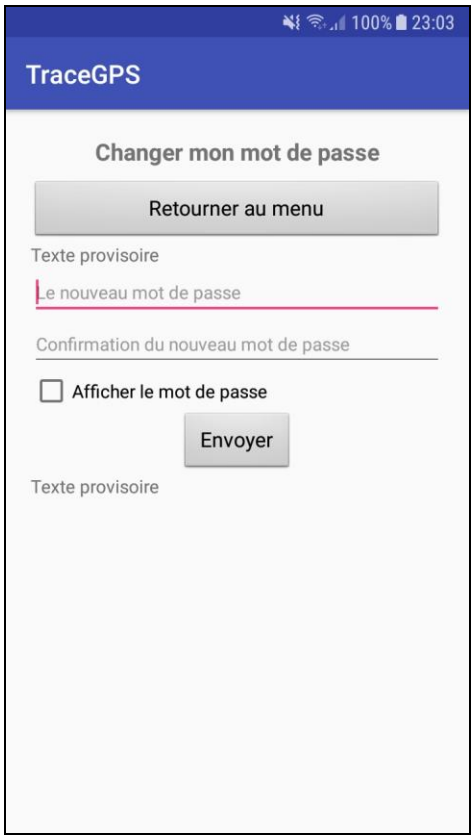
```
/** classe interne pour gérer le clic sur le bouton buttonChangerMdp. */
private class buttonChangerMdpClickListener implements View.OnClickListener{
    public void onClick(View v) {
        // crée une Intent pour lancer l'activité
        Intent uneIntent = new Intent(MenuGeneral.this, ChangerMdp.class);
        // 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 et attend un résultat (le nouveau mdp)
        startActivityForResult(uneIntent, CODE_RESULTAT_CHANGEMENT_MDP);
    }
}
```

Et ajoutez la fonction **onActivityResult** qui devra gérer le retour d'une activité appelée avec **startActivityForResult** (c'est-à-dire une activité qui retourne des données) :

```
/** récupére les données fournies par une activité appelée avec startActivityForResult
 *   ici, une seule activité concernée : celle qui permet de changer le mot de passe
 *   il faut alors récupérer ce nouveau mot de passe pour pouvoir l'utiliser lors des appels de services web
 */
protected void onActivityResult(int requestCode, int resultCode, Intent uneIntent) {
    super.onActivityResult(requestCode, resultCode, uneIntent);

    switch (requestCode) {
        case CODE_RESULTAT_CHANGEMENT_MDP :
            mdp = uneIntent.getStringExtra(EXTRA_MDP); break;
    }
}
```

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

Le bouton Modifier mon mot de passe doit activer l'activité **ChangerMdp** :



6- Programmation Java de l'activité ChangerMdp.java

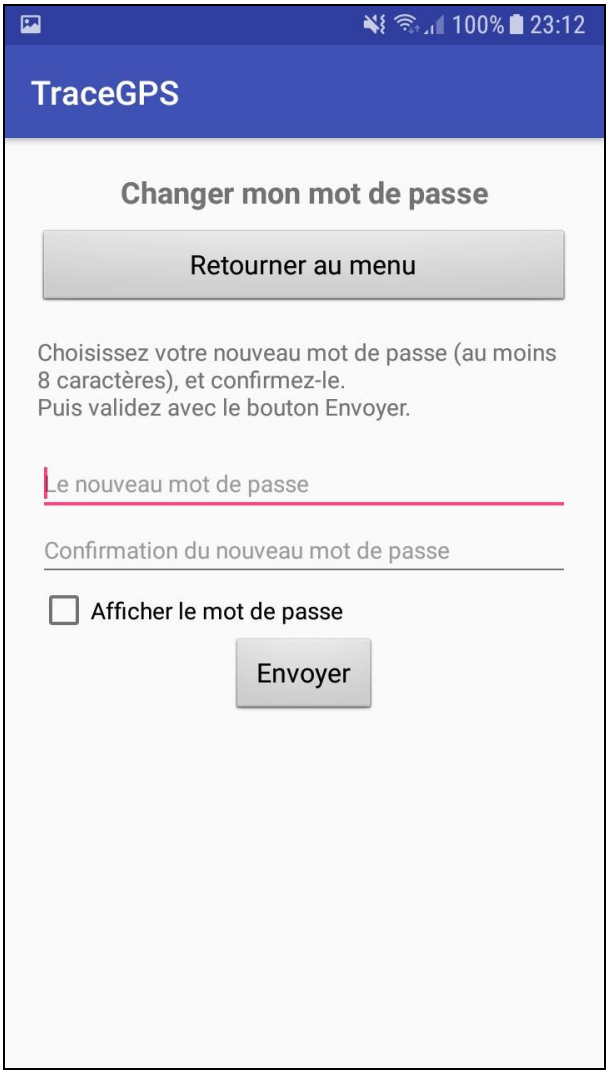
6-1 Déclarations diverses et initialisation des objets

Dans le fichier **ChangerMdp.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.ProgressBar;
import android.view.View;
import android.content.Intent;

public class ChangerMdp extends AppCompatActivity {

    // les zones de saisie et les variables associées
    private EditText editTextMdp;
    private EditText editTextConfirmationMdp;

    private String nouveauMdp;
    private String confirmation;

    // les 2 boutons
    private Button buttonEnvoyer;
    private Button buttonRetourner;

    // le ProgressBar pour afficher le cercle de chargement
    private ProgressBar progressBar;

    // les zones d'affichage de message
    private TextView textViewGuide;
    private TextView textViewMessage;

    // 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 nom;
    private String mdp;

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

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

        // récupération des EditText grâce à leur ID
        editTextMdp = (EditText) findViewById(R.id.changer_mdp_editTextMdp);
        editTextConfirmationMdp = (EditText) findViewById(R.id.changer_mdp_editTextConfirmationMdp);

        // récupération des Button grâce à leur ID
        buttonEnvoyer = (Button) findViewById(R.id.changer_mdp_buttonEnvoyer);
        buttonRetourner = (Button) findViewById(R.id.changer_mdp_buttonRetourner);

        // récupération des TextView grâce à leur ID et initialisations des textes affichés
        textViewGuide = (TextView) findViewById(R.id.changer_mdp_textViewGuide);
        textViewMessage = (TextView) findViewById(R.id.changer_mdp_textViewMessage);
        String msg = "\nChoisissez votre nouveau mot de passe (au moins 8 caractères), et confirmez-le.\n";
        msg += "Puis validez avec le bouton Envoyer.\n";
        textViewGuide.setText(msg);
        textViewMessage.setText("");
```

```
        progressBar = (ProgressBar) findViewById(R.id.changer_mdp_progressBar);
        // arrête le cercle de chargement
        progressBar.setVisibility(View.GONE);

        // association d'un écouteur d'évenement à chaque bouton
        buttonEnvoyer.setOnClickListener ( new buttonEnvoyerClickListener());
        buttonRetourner.setOnClickListener ( new buttonRetournerClickListener());
    } // fin de onCreate

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

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

Exécutez et testez :



6-2 Gestion du bouton Retourner au menu

Complétez l'écouteur **buttonRetournerClickListener** :

```
/** classe interne pour gérer le clic sur le bouton buttonRetourner. */
private class buttonRetournerClickListener implements View.OnClickListener {
    public void onClick(View v) {
        // crée une Intent pour retourner nouveauMdp à l'activité MenuGeneral
        Intent uneIntent = new Intent();
        // passe les données à l'Intent
        uneIntent.putExtra(EXTRA_MDP, mdp);
        setResult(RESULT_OK, uneIntent);
        finish();
    }
}
```

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

6-3 Gestion du bouton Changer mon mot de passe

On fait appel ici au service web **ChangerDeMdp** pour changer le mot de passe du compte utilisateur.

Commencez par ajouter les **import** suivants aux **import** existants :

```
import android.os.AsyncTask;
import android.widget.Toast;
import jim.classes.Outils;
import jim.classes.PasserelleServicesWebXML;
```

Complétez l'écouteur **buttonEnvoyerClickListener** pour appeler la tâche **TacheChangerDeMdp** :

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

        Mémoriser les valeurs saisies dans les variables nouveauMdp et confirmation.

        Si le nouveau mot de passe comporte moins de 8 caractères, afficher le message "Vous devez saisir un mot de passe d'au moins 8 caractères." dans l'objet textViewMessage et dans un toast fugitif.

        Si le nouveau mot de passe et sa confirmation sont différents, afficher le message "Erreur : le nouveau mot de passe et sa confirmation sont différents." dans l'objet textViewMessage et dans un toast fugitif.

        Si les saisies sont correctes, lancez l'exécution de la tâche asynchrone TacheChangerDeMdp qui sera écrite à l'étape suivante et qui appellera le service web ChangerDeMdp.


    }
}
```

En vous inspirant des tâches asynchrones déja écrites dans les activités précédentes, ajoutez la tâche asynchrone **TacheChangerDeMdp** :

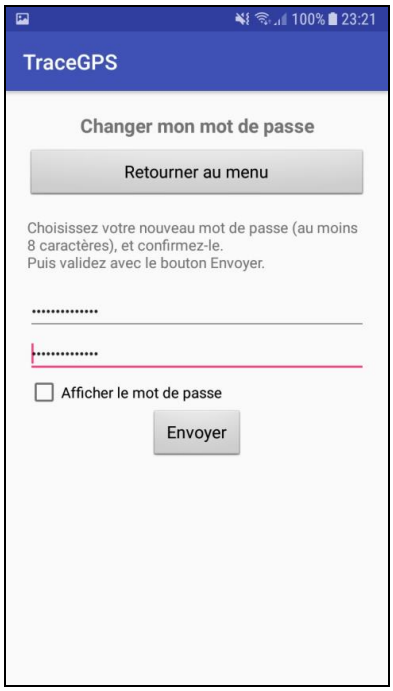
```
// ----------------------------------------------------------------------------------------------
// ------------------------------ Tâche asynchrone TacheChangerDeMdp ------------------------------
// ----------------------------------------------------------------------------------------------

// appel du service web ChangerDeMdp
private class TacheChangerDeMdp extends AsyncTask<Void, Void, String> {


} // fin tâche asynchrone TacheChangerDeMdp
```

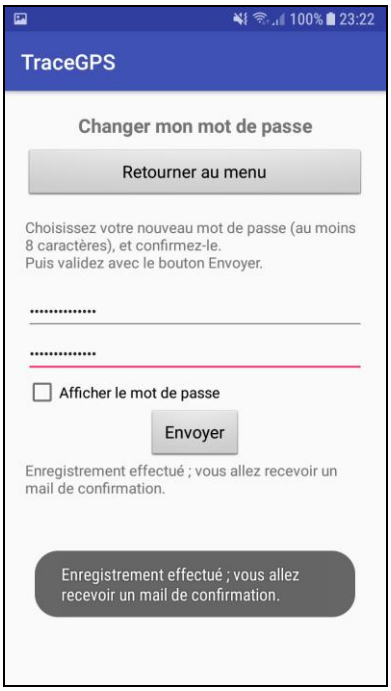
La fonction **onPreExecute** doit désactiver le bouton **buttonRetourner** et démarrer l'affichage de l'objet **progressBar**.

La fonction **doInBackground** doit appeler le service web **ChangerDeMdp** en utilisant une des méthodes statiques de la classe **PasserelleServicesWebXML** et en lui passant les paramètres nécessaires.

La fonction **onPostExecute** doit réactiver le bouton **buttonRetourner,** arrêter l'affichage de l'objet **progressBar** et afficher dans l'objet **textViewMessage** et dans un toast fugitif le message retourné par la méthode ;
si le message retourné par la méthode ne commence pas par le mot "**Erreur**", le nouveau mot de passe est

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





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

Cher(chère) jim12345

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

Votre nouveau mot de passe est : 12345678

6-4 Gestion de la case à cocher permettant d'afficher (ou non) le mot de passe

Vous êtes chargés de coder la possibilité d'afficher en clair les mots de passe.

La technique à utiliser a déjà été mise en oeuvre dans la page de connexion. En voici une description :

```
protected void onCreate(Bundle savedInstanceState) {
```

Dans la fonction **onCreate**, créer un objet Java nommé **caseMdpVisible** correspondant à l'objet XML du layout, puis associer un écouteur d'événement **caseMdpVisibleClickListener** pour gérer le **Click** sur l'objet **caseMdpVisible**

```
} // fin de onCreate
```

Et ajoutez cet écouteur **caseMdpVisibleClickListener** dans votre code :

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

Si la case **caseMdpVisible** est cochée
    les contenus des 2 objets **editTextMdp** et **editTextConfirmationMdp** sont rendus visibles
sinon
    les contenus des 2 objets **editTextMdp** et **editTextConfirmationMdp** sont rendus invisibles

```
    }
}
```

L'affichage obtenu sera du type :