

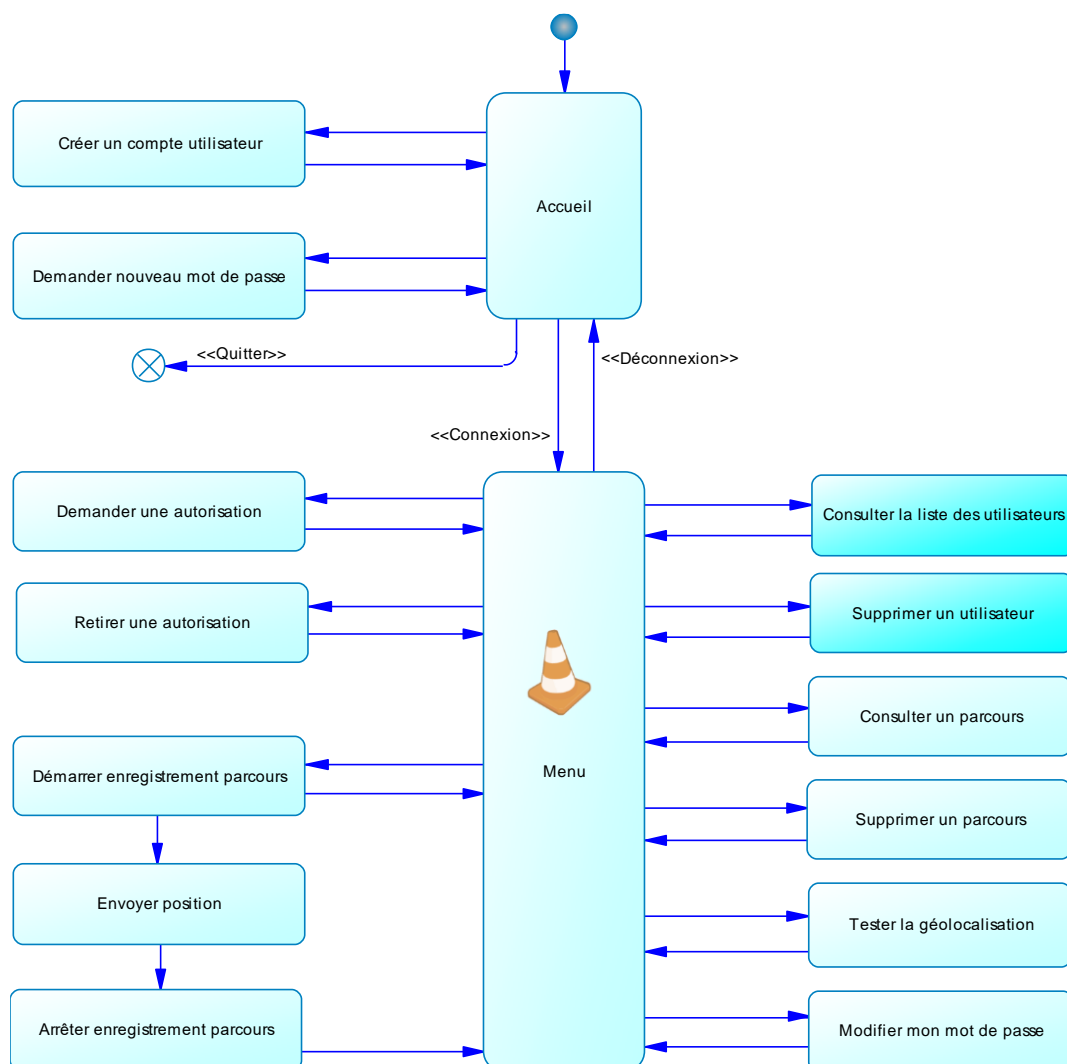


5- Développement de l'application mobile Android

5-2 MenuGeneral (page de menu)

- 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 MainActivity.java
- 6- Programmation Java de l'activité MenuGeneral.java

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



2- Modification du fichier strings.xml

L'interface graphique à créer (**utilisateur** à gauche, **administrateur** à droite) :

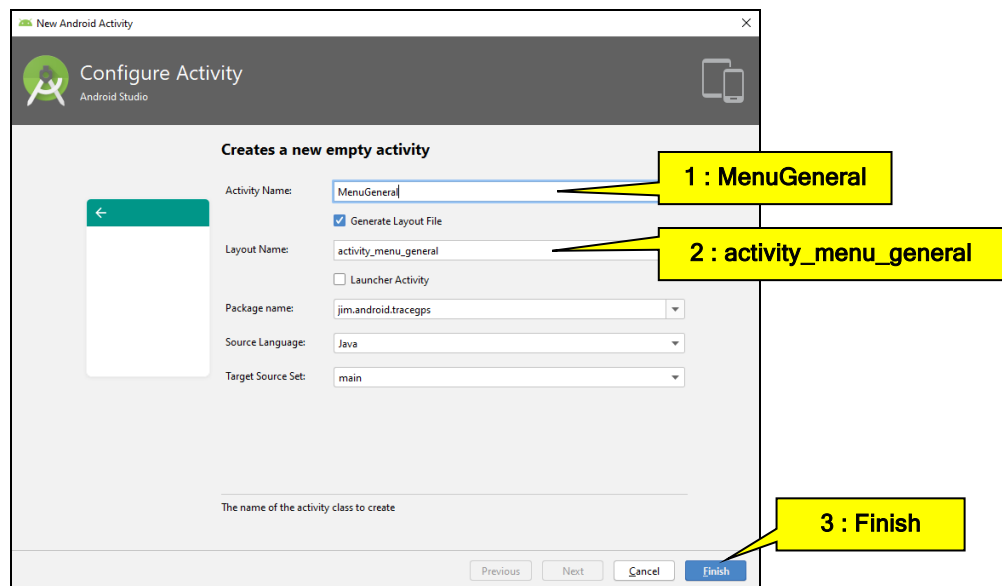


Dans le dossier **res/values**, complétez le fichier **strings.xml** avec le code suivant :

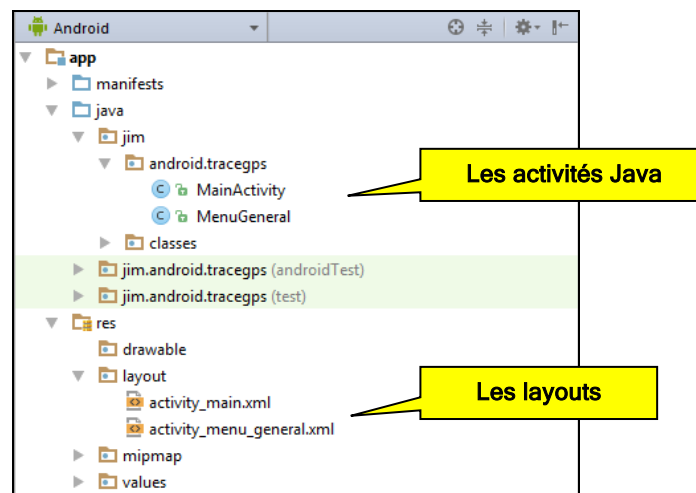
```
<!-- Les textes de la page de menu -->
<string name="menu_titre1">Menu</string>
<string name="menu_bouton_tester_geolocalisation">Tester la géolocalisation</string>
<string name="menu_bouton_demander_autorisation">Demander une autorisation</string>
<string name="menu_bouton_retirer_autorisation">Retirer une autorisation</string>
<string name="menu_bouton_demarrer_enregistrement">Démarrer enregistrement parcours</string>
<string name="menu_bouton_consulter_parcours">Consulter un parcours</string>
<string name="menu_bouton_supprimer_parcours">Supprimer un parcours</string>
<string name="menu_bouton_changer_mdp">Modifier mon mot de passe</string>
<string name="menu_bouton_lister_utilisateurs">Consulter la liste des utilisateurs</string>
<string name="menu_bouton_supprimer_utilisateur">Supprimer un utilisateur</string>
<string name="menu_bouton_deconnecter">Se déconnecter</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 **MenuGeneral.java** et le layout **activity_menu_general.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:supportRtl="true"
    android:theme="@style/AppTheme"
    android:usesCleartextTraffic="true">
    <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>
```

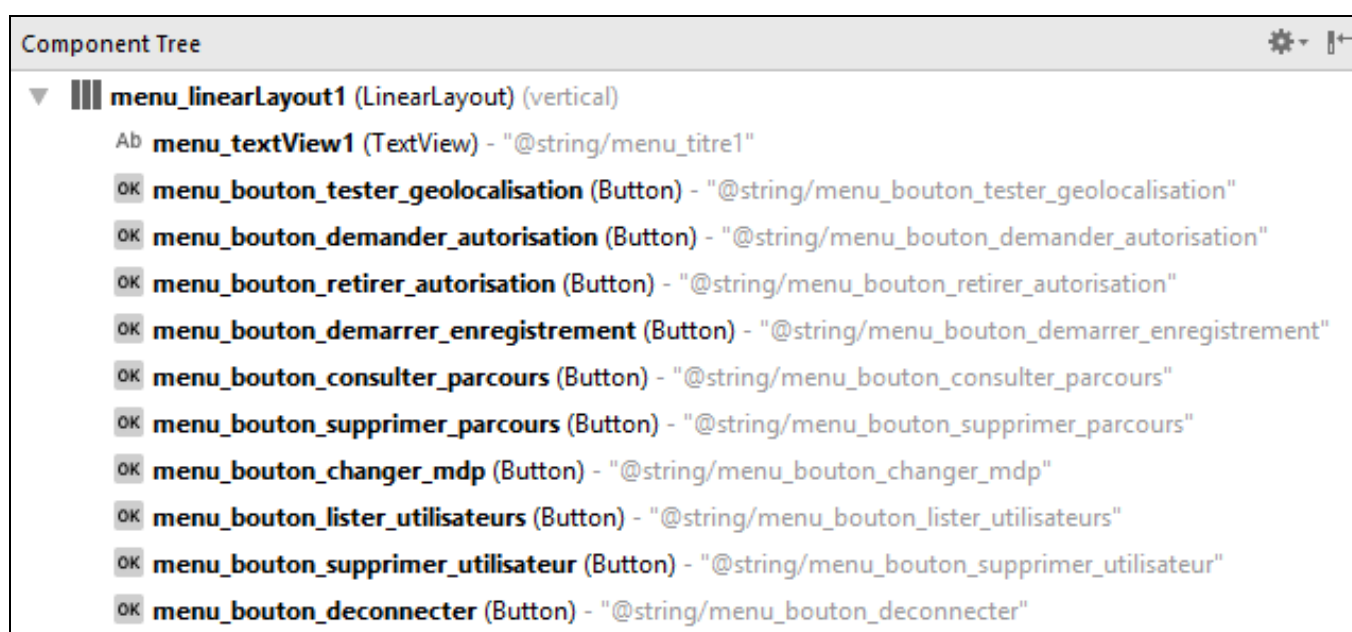
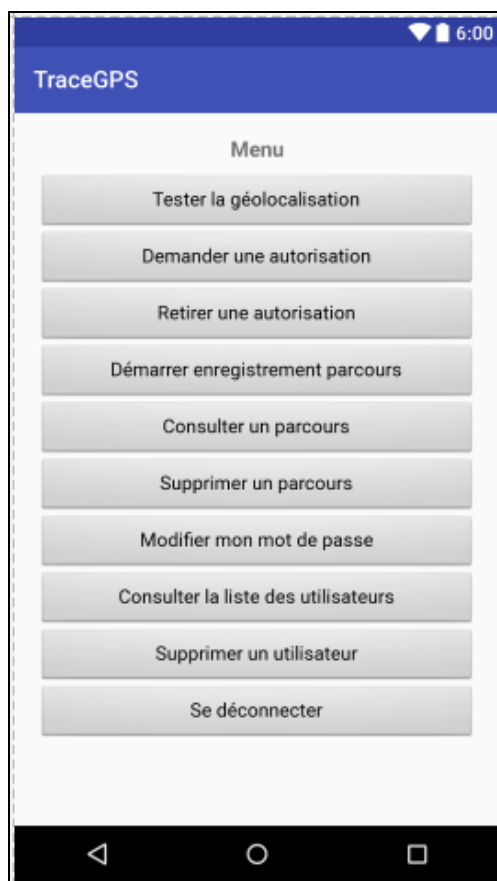
Callouts in the image:

- L'activité ajoutée:** Points to the `<activity android:name=".MenuGeneral"></activity>` line.
- L'activité principale:** Points to the `<activity android:name=".MainActivity">` line.

4- Création de l'interface graphique

Le fichier **activity_menu_general.xml** du dossier **res/layout** contient la description de l'interface graphique. Dès la création d'une activité, 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 :



Le code XML (à recopier) 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:id="@+id/menu_linearLayout1"
    android:orientation="vertical"
    android:padding="@dimen/tailleMarges"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="jim.android.tracegps.MenuGeneral">

    <TextView
        android:id="@+id/menu_textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="10dp"
        android:text="@string/menu_titre1"
        android:textAlignment="center"
        android:textSize="18sp"
        android:textStyle="bold" />

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

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

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

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

    <Button
        android:id="@+id/menu_bouton_consulter_parcours"
        style="@android:style/Widget.Button"
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/menu_bouton_consulter_parcours"
android:textSize="16sp" />
```

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

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

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

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

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

```
</LinearLayout >
```

5- Modification de la programmation Java de MainActivity.java

Nous allons commencer par modifier la page d'identification (**MainActivity.java**) pour activer la page de menu (**MenuGeneral.java**) quand l'authentification est réussie.

Nous allons également utiliser des "extras" pour transmettre à l'activité **MenuGeneral.java** les 3 données suivantes (qui seront également retransmises aux autres activités pour pouvoir appeler les différents services web) :

- le pseudo de connexion
- le mot de passe de connexion
- le type d'utilisateur ("utilisateur" ou "administrateur")

Commencez par ajouter l' **import** suivant à la suite des **import** existants :

```
import android.content.Intent;
```

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

```
// 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 3 ici, dont voici les 3 clés :
private final String EXTRA_PSEUDO = "pseudo";
private final String EXTRA_MDP = "mdp";
private final String EXTRA_TYPE_UTILISATEUR = "typeUtilisateur";
```

Remplacez les 2 lignes provisoires de la méthode **onPostExecute** de la tâche **TacheConnecter** :

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

// appel du service web Connecter
private class TacheConnecter extends AsyncTask<Void, Void, String> {
    .....
    .....
    .....
    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);
        }
        else {
            // détermine le type d'utilisateur
            if (msg.startsWith("Utilisateur authentifié")) typeUtilisateur = "utilisateur";
            if (msg.startsWith("Administrateur authentifié")) typeUtilisateur = "administrateur";

            // efface le dernier message
            textViewMessage.setText("");

            // crée une Intent pour lancer le menu général
            Intent unelIntent = new Intent(MainActivity.this, MenuGeneral.class);
            // passe nom, mdp et typeUtilisateur à l'Intent
            unelIntent.putExtra(EXTRA_PSEUDO, pseudo);
            unelIntent.putExtra(EXTRA_MDP, mdp);
            unelIntent.putExtra(EXTRA_TYPE_UTILISATEUR, typeUtilisateur);
            // démarre l'activité à partir de l'Intent
            startActivity(unelIntent);
        }
    }
} // fin tâche asynchrone TacheConnecter
```

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

En cas d'authentification correcte, l'activité **MenuGeneral** prend la main :



6- Programmation Java de l'activité MenuGeneral.java

6-1 Déclarations diverses et initialisation des objets

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

```
package jim.android.tracegps;

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

import android.view.View;
import android.widget.Button;

public class MenuGeneral extends AppCompatActivity {

    // les boutons
    private Button boutonTesterGeolocalisation;
    private Button boutonDemanderAutorisation;
    private Button boutonRetirerAutorisation;
    private Button boutonDemarrerEnregistrement;
    private Button boutonConsulterParcours;
    private Button boutonSupprimerParcours;
    private Button boutonChangerMdp;
    private Button boutonConsulterUtilisateurs;
    private Button boutonSupprimerUtilisateur;
    private Button boutonDeconnecter;

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

        // récupération des Button grâce à leur ID
        boutonTesterGeolocalisation = (Button) findViewById(R.id.menu_bouton_tester_geolocalisation);
        boutonDemanderAutorisation = (Button) findViewById(R.id.menu_bouton_demander_autorisation);
        boutonRetirerAutorisation = (Button) findViewById(R.id.menu_bouton_retirer_autorisation);
        boutonDemarrerEnregistrement = (Button) findViewById(R.id.menu_bouton_demarrer_enregistrement);
        boutonConsulterParcours = (Button) findViewById(R.id.menu_bouton_consulter_parcours);
        boutonSupprimerParcours = (Button) findViewById(R.id.menu_bouton_supprimer_parcours);
        boutonChangerMdp = (Button) findViewById(R.id.menu_bouton_changer_mdp);
        boutonConsulterUtilisateurs = (Button) findViewById(R.id.menu_bouton_lister_utilisateurs);
        boutonSupprimerUtilisateur = (Button) findViewById(R.id.menu_bouton_supprimer_utilisateur);
        boutonDeconnecter = (Button) findViewById(R.id.menu_bouton_deconnecter);

        // association d'un écouteur d'événement à chaque bouton
        boutonTesterGeolocalisation.setOnClickListener ( new boutonTesterGeolocalisationClickListener());
        boutonDemanderAutorisation.setOnClickListener ( new boutonDemanderAutorisationClickListener());
        boutonRetirerAutorisation.setOnClickListener ( new boutonRetirerAutorisationClickListener());
        boutonDemarrerEnregistrement.setOnClickListener ( new boutonDemarrerEnregistrementClickListener());
        boutonConsulterParcours.setOnClickListener ( new boutonConsulterParcoursClickListener());
        boutonSupprimerParcours.setOnClickListener ( new boutonSupprimerParcoursClickListener());
        boutonChangerMdp.setOnClickListener ( new boutonChangerMdpClickListener());
        boutonConsulterUtilisateurs.setOnClickListener ( new boutonConsulterUtilisateursClickListener());
        boutonSupprimerUtilisateur.setOnClickListener ( new boutonSupprimerUtilisateurClickListener());
        boutonDeconnecter.setOnClickListener ( new boutonDeconnecterClickListener());
    }

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

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

```

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

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

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

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

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

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

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

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

} // FIN DE LA CLASSE

```

6-2 Gestion du bouton Déconnecter

Complétez l'écouteur **boutonDeconnecterClickListener** :

```

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

```

Testez l'application et le bon fonctionnement du bouton **Déconnecter**.

6-3 Gestion des 2 types de menus en fonction du type d'utilisateur

Cette activité tient compte des données envoyées par l'activité précédente pour choisir d'afficher le menu utilisateur ou le menu administrateur.

Commencez par ajouter l' **import** suivant à la suite des **import** existants :

```
import android.content.Intent;
```

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

```
// 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 3 ici, dont voici les 3 clés et les 3 variables associées :
private final String EXTRA_PSEUDO = "pseudo";
private final String EXTRA_MDP = "mdp";
private final String EXTRA_TYPE_UTILISATEUR = "typeUtilisateur";
private String pseudo;
private String mdp;
private String typeUtilisateur; // "utilisateur" ou "administrateur"
```

Complétez la fonction **onCreate** :

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

    // récupération des Button grâce à leur ID
    buttonTesterGeolocalisation = (Button) findViewById(R.id.menu_bouton_tester_geolocalisation);
    buttonDemanderAutorisation = (Button) findViewById(R.id.menu_bouton_demander_autorisation);
    buttonRetirerAutorisation = (Button) findViewById(R.id.menu_bouton_retirer_autorisation);
    buttonDemarrerEnregistrement = (Button) findViewById(R.id.menu_bouton_demarrer_enregistrement);
    buttonConsulterParcours = (Button) findViewById(R.id.menu_bouton_consulter_parcours);
    buttonSupprimerParcours = (Button) findViewById(R.id.menu_bouton_supprimer_parcours);
    buttonChangerMdp = (Button) findViewById(R.id.menu_bouton_changer_mdp);
    buttonConsulterUtilisateurs = (Button) findViewById(R.id.menu_bouton_lister_utilisateurs);
    buttonSupprimerUtilisateur = (Button) findViewById(R.id.menu_bouton_supprimer_utilisateur);
    buttonDeconnecter = (Button) findViewById(R.id.menu_bouton_deconnecter);

    // association d'un écouteur d'événement à chaque bouton
    buttonTesterGeolocalisation.setOnClickListener ( new buttonTesterGeolocalisationClickListener());
    buttonDemanderAutorisation.setOnClickListener ( new buttonDemanderAutorisationClickListener());
    buttonRetirerAutorisation.setOnClickListener ( new buttonRetirerAutorisationClickListener());
    buttonDemarrerEnregistrement.setOnClickListener ( new buttonDemarrerEnregistrementClickListener());
    buttonConsulterParcours.setOnClickListener ( new buttonConsulterParcoursClickListener());
    buttonSupprimerParcours.setOnClickListener ( new buttonSupprimerParcoursClickListener());
    buttonChangerMdp.setOnClickListener ( new buttonChangerMdpClickListener());
    buttonConsulterUtilisateurs.setOnClickListener ( new buttonConsulterUtilisateursClickListener());
    buttonSupprimerUtilisateur.setOnClickListener ( new buttonSupprimerUtilisateurClickListener());
    buttonDeconnecter.setOnClickListener ( new buttonDeconnecterClickListener());

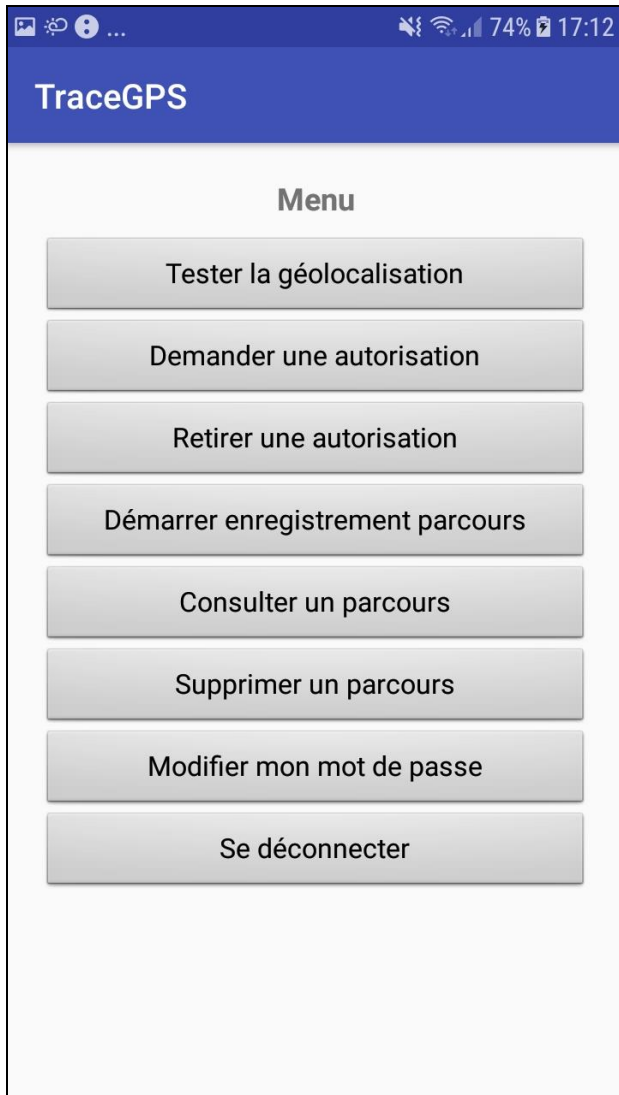
    // récupération du nom, du mot de passe et du type d'utilisateur passés par l'activité précédente
    Intent unelIntent = getIntent();
    pseudo = unelIntent.getStringExtra(EXTRA_PSEUDO);
    mdp = unelIntent.getStringExtra(EXTRA_MDP);
    typeUtilisateur = unelIntent.getStringExtra(EXTRA_TYPE_UTILISATEUR);

    // boutons buttonConsulterUtilisateurs et buttonSupprimerUtilisateur visibles seulement pour un administrateur
    if ( typeUtilisateur.equals("administrateur") ) {
        // masquer les 2 boutons
        buttonConsulterUtilisateurs.setVisibility(View.GONE);
        buttonSupprimerUtilisateur.setVisibility(View.GONE);
    }
}
```

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

Pour le test, vous vous connecterez en utilisateur et en administrateur.

***Menu obtenu après une connexion
en utilisateur***



***Menu obtenu après une connexion
en administrateur***

