

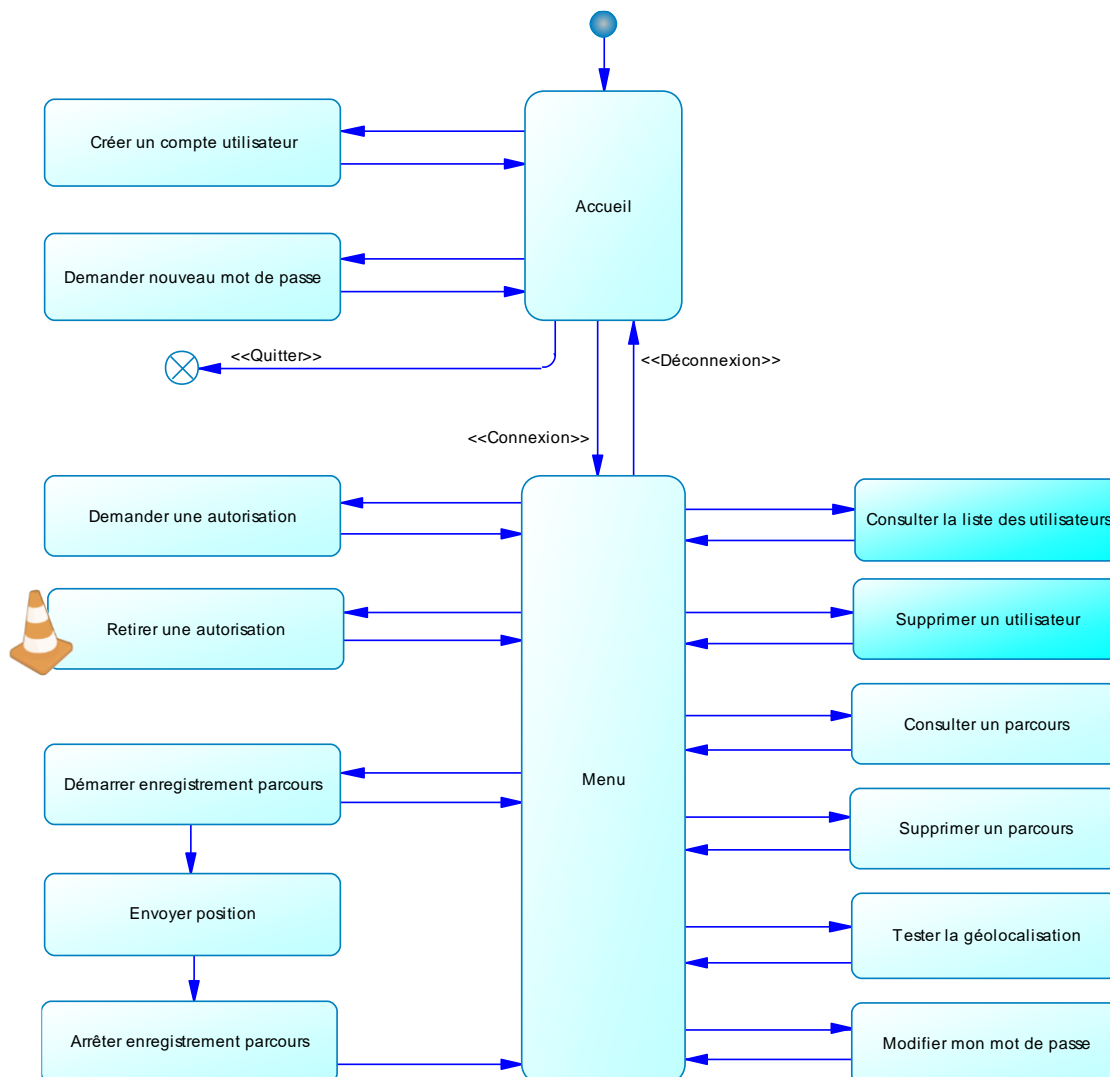


## 5- Développement de l'application mobile Android

### 5-6 RetirerAutorisation (retirer une autorisation)

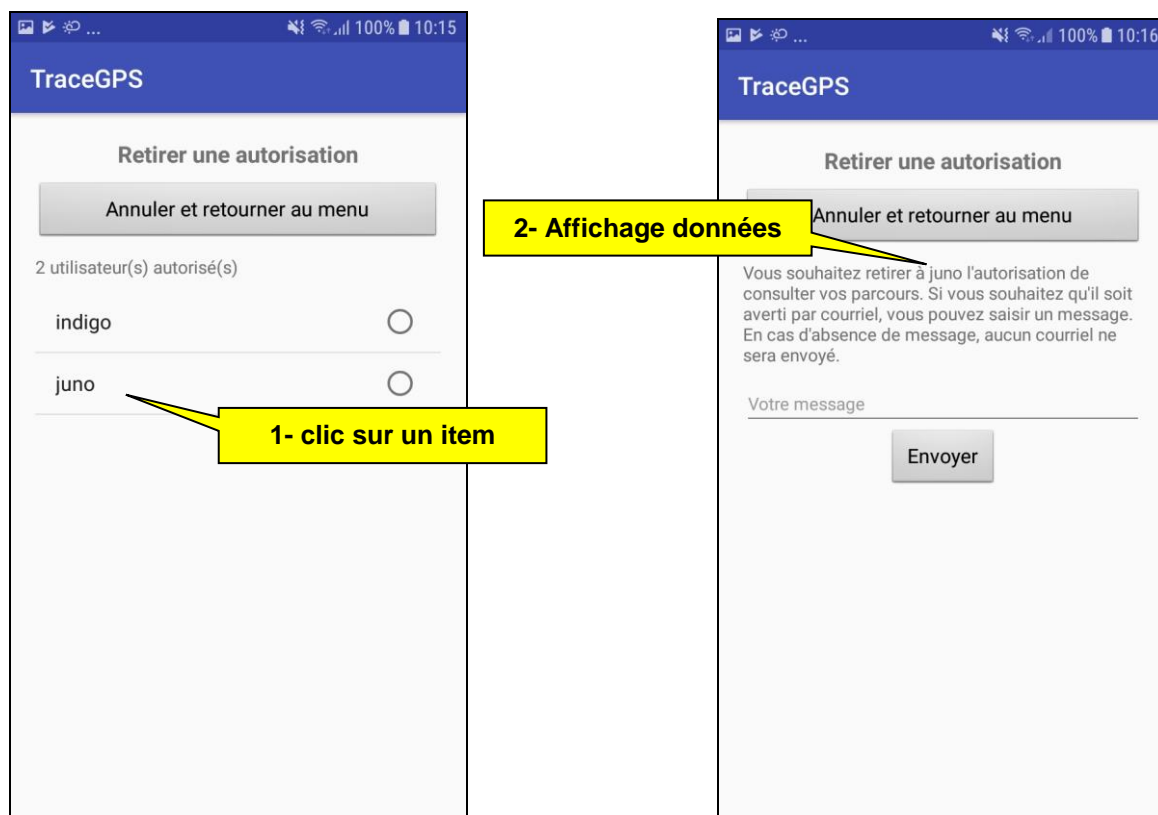
- 1- Situation de l'activité dans la structure de l'application
- 2- Modification du fichier strings.xml
- 3- Création de l'activité
- 4- Création de l'interface graphique
- 5- Modification de la programmation Java de MenuGeneral.java
- 6- Programmation Java de l'activité RetirerAutorisation.java

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



## 2- Modification du fichier strings.xml

L'interface graphique à créer :

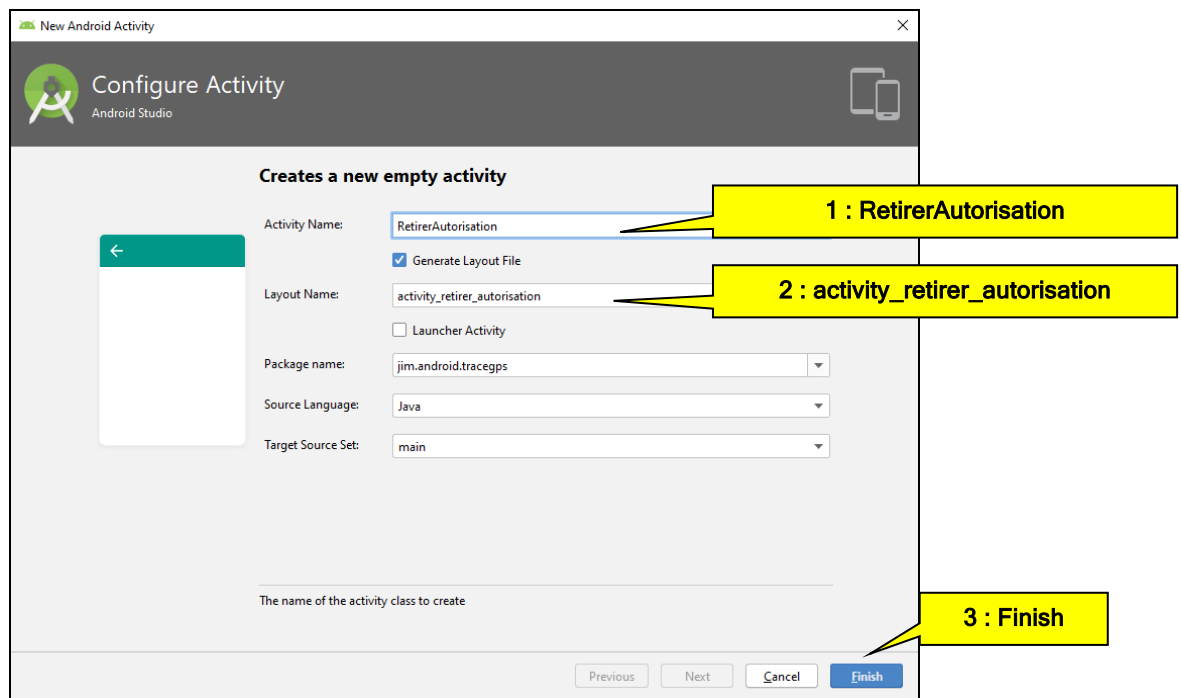


Dans le dossier **res/values**, complétez le fichier **strings.xml** avec le code suivant :

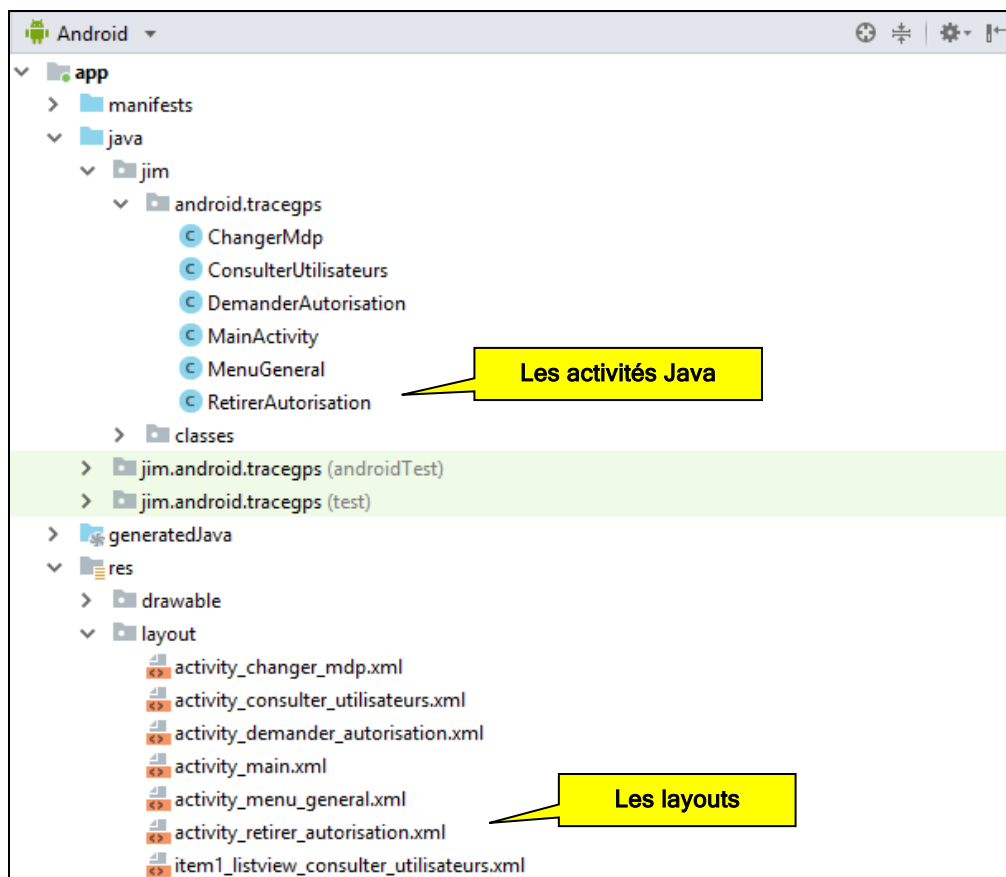
```
<!-- Les textes de la page retirer une autorisation -->
<string name="retirer_autorisation_titre1">Retirer une autorisation</string>
<string name="retirer_autorisation_bouton_retourner">Annuler et retourner au menu</string>
<string name="retirer_autorisation_bouton_envoyer">Envoyer</string>
<string name="retirer_autorisation_saisie_message">Votre message</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 **RetirerAutorisation.java** et le layout **activity\_retirer\_autorisation.xml** sont créés :



## 4- Création de l'interface graphique

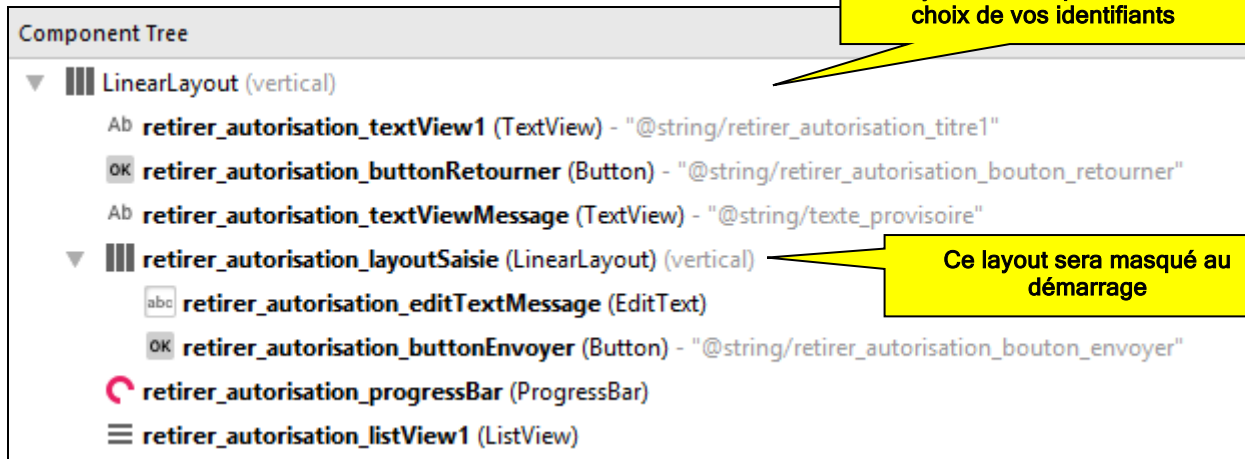
A la création d'une nouvelle activité, l'interface comporte automatiquement un **ConstraintLayout** vide.

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** :

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

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



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.RetirerAutorisation">
```

```
<TextView
    android:id="@+id/retirer_autorisation_textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingBottom="10dp"
    android:text="@string/retirer_autorisation_titre1"
    android:textAlignment="center"
    android:textSize="18sp"
    android:textStyle="bold" />

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

<TextView
    android:id="@+id/retirer_autorisation_textViewMessage"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:text="@string/texte_provisoire"
    android:textSize="14sp" />

<LinearLayout
    android:id="@+id/retirer_autorisation_layoutSaisie"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <EditText
        android:id="@+id/retirer_autorisation_editTextMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="14sp"
        android:hint="@string/retirer_autorisation_saisie_message" />

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

</LinearLayout>

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

<ListView
    android:id="@+id/retirer_autorisation_listView1"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

</LinearLayout>
```

## 5- Modification de la programmation Java de MenuGeneral.java

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

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

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

Le bouton **Retirer une autorisation** doit activer l'activité **RetirerAutorisation** :



## 6- Programmation Java de l'activité RetirerAutorisation.java

### 6-1 Déclarations diverses et initialisation des objets

Dans le fichier **RetirerAutorisation.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.ListView;
import android.widget.ProgressBar;
import android.widget.LinearLayout;
import android.widget.AdapterView;
import android.view.View;
import android.content.Intent;

public class RetirerAutorisation extends AppCompatActivity {

    // les objets du layout
    private TextView textViewMessage;           // le TextView pour afficher le message
    private EditText editTextMessage;          // l'EditText pour la saisie du message
    private Button buttonRetourner;            // le Button pour retourner au menu
    private Button buttonEnvoyer;             // le Button pour retirer la demande
    private ProgressBar progressBar;           // le ProgressBar pour afficher le cercle de chargement
    private LinearLayout layoutSaisie;         // le layout contenant les 2 EditText et les 2 Button
    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;

    private String texteMessage;              // pour la saisie du message

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

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

        // récupération des objets du layout grâce à leur ID
        textViewMessage = (TextView) findViewById(R.id.retirer_autorisation_textViewMessage);
        editTextMessage = (EditText) findViewById(R.id.retirer_autorisation_editTextMessage);
        buttonRetourner = (Button) findViewById(R.id.retirer_autorisation_buttonRetourner);
        buttonEnvoyer = (Button) findViewById(R.id.retirer_autorisation_buttonEnvoyer);
        progressBar = (ProgressBar) findViewById(R.id.retirer_autorisation_progressBar);
        layoutSaisie = (LinearLayout) findViewById(R.id.retirer_autorisation_layoutSaisie);
        laListView = (ListView) findViewById(R.id.retirer_autorisation_listView1);
        // arrête le cercle de chargement
        progressBar.setVisibility(View.GONE);
    }
}
```



```

// association d'un écouteur d'événement aux boutons
buttonRetourner.setOnClickListener ( new buttonRetournerClickListener());
buttonEnvoyer.setOnClickListener ( new buttonEnvoyerClickListener());

// association d'un écouteur d'événement à l'événement OnItemClickListener du ListView
laListView.setOnItemClickListener ( new laListViewOnItemClickListener());
}

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

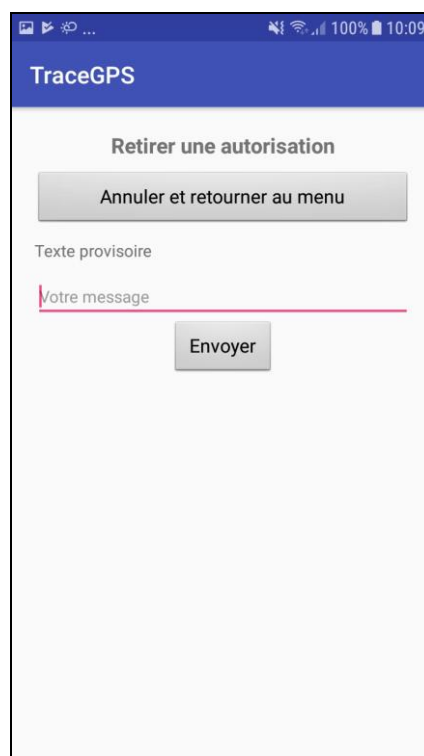
/** classe interne pour gérer le clic sur le bouton buttonEnvoyer. */
private class buttonEnvoyerClickListener implements View.OnClickListener {
    public void onClick(View v) {
        // A COMPLETER PLUS TARD
    }
}

/** classe interne pour gérer le clic sur un item du ListView. */
private class laListViewOnItemClickListener implements AdapterView.OnItemClickListener{
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        // A COMPLETER PLUS TARD
    }
}

} // fin de l'activité

```

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



## 6-2 Mise en place d'une tâche asynchrone

L'affichage des utilisateurs actuellement autorisés nécessite d'appeler le service web :

- **GetLesUtilisateursQueJautorise** : pour obtenir la liste des utilisateurs que j'autorise actuellement à suivre mes parcours

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> lesUtilisateursQueJautorise; // les utilisateurs que j'autorise
```

A la fin de l'activité, ajoutez la tâche asynchrone **TacheGetLesUtilisateursQueJautorise** (vous pouvez vous inspirer du document "**5-4 (0) Projet TraceGPS - Dév appli android - ConsulterUtilisateurs**") :

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

private class TacheGetLesUtilisateursQueJautorise extends AsyncTask<ArrayList<Utilisateur>, Void, String> {

    La fonction onPreExecute doit démarrer l'affichage de l'objet progressBar.

    La fonction doInBackground doit appeler le service web GetLesUtilisateursQueJautorise en utilisant une des
    méthodes statiques de la classe PasserelleServicesWebXML et en lui passant les paramètres nécessaires.
    Cette méthode devra remplir la collection lesUtilisateursQueJautorise.

    La fonction onPostExecute doit arrêter l'affichage de l'objet progressBar et tester le message retourné par le
    service web :

    • Si le message retourné par la méthode commence par le mot "Erreur", il faut afficher dans l'objet
      textViewMessage le message retourné par la méthode

    • Sinon, il faut exécuter la fonction afficherLesUtilisateurs dont le code provisoire est donné plus loin

}
```

A la fin de l'activité, ajoutez la fonction provisoire **afficherLesUtilisateurs** :

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

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

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

    // 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
    textViewMessage = (TextView) findViewById(R.id.retirer_autorisation_textViewMessage);
    editTextMessage = (EditText) findViewById(R.id.retirer_autorisation_editTextMessage);
    buttonRetourner = (Button) findViewById(R.id.retirer_autorisation_buttonRetourner);
    buttonEnvoyer = (Button) findViewById(R.id.retirer_autorisation_buttonEnvoyer);
    progressBar = (ProgressBar) findViewById(R.id.retirer_autorisation_progressBar);
    layoutSaisie = (LinearLayout) findViewById(R.id.retirer_autorisation_layoutSaisie);
    laListView = (ListView) findViewById(R.id.retirer_autorisation_listView1);
    // arrête le cercle de chargement
    progressBar.setVisibility(View.GONE);

    // association d'un écouteur d'événement aux boutons
    buttonRetourner.setOnClickListener ( new buttonRetournerClickListener());
    buttonEnvoyer.setOnClickListener ( new buttonEnvoyerClickListener());

    // association d'un écouteur d'événement à l'événement OnItemClickListener du ListView
    laListView.setOnItemClickListener ( new laListViewOnItemClickListener());
}
```

Masquer le layout **layoutSaisie**.

Instancier la collection **lesUtilisateursQueJautorise**.

Lancer l'exécution de la tâche asynchrone **TacheGetLesUtilisateursQueJautorise**.

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



### 6-3 Gestion de l'affichage des utilisateurs dans la ListView

A la suite des **import** existants, ajoutez l'**import** suivant :

```
import android.widget.AdapterView;
```

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

```
private ArrayList<String> listeChaines; // les libellés à placer dans le ListView
```

Complétez la fonction définitive **afficherLesUtilisateurs** :

```
// afficher la liste des utilisateurs autorisés
public void afficherLesUtilisateurs() {
    // on affiche le nombre d'utilisateurs autorisés
    textViewMessage.setText(lesUtilisateursQueJautorise.size() + " utilisateur(s) autorisé(s)");

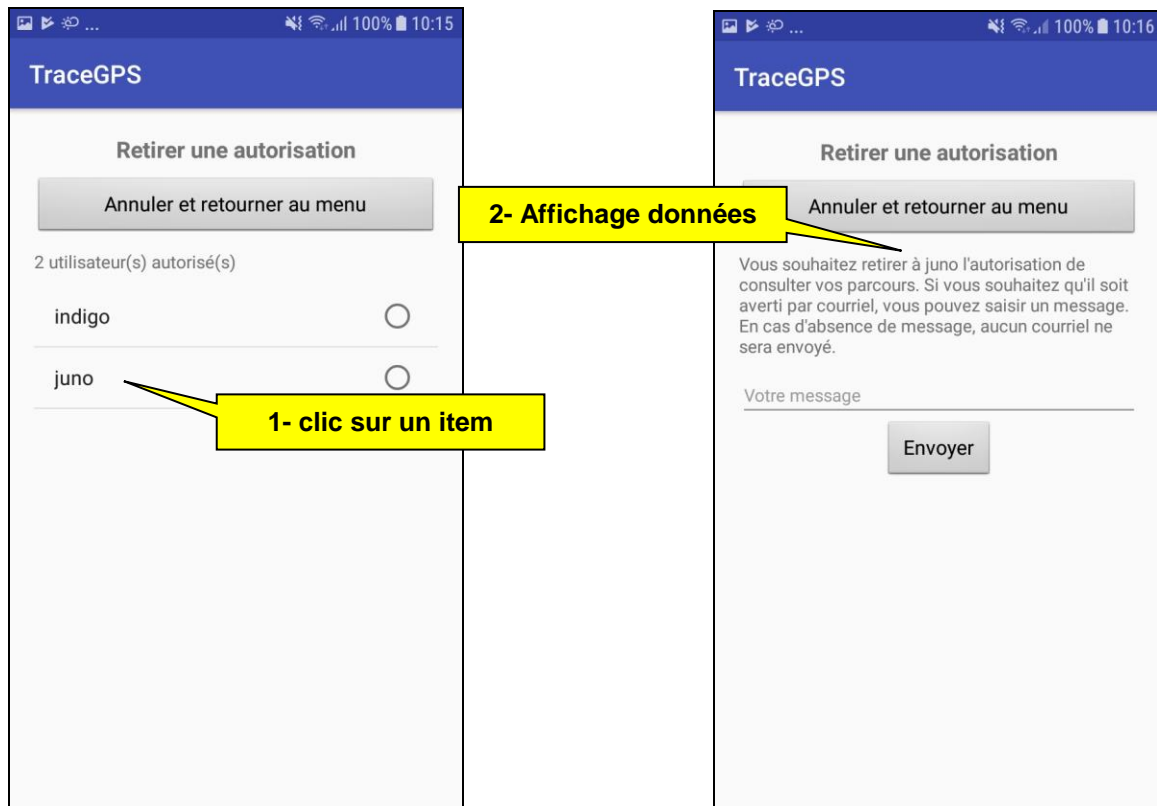
    // vidage de la liste pour afficher les utilisateurs
    listeChaines = new ArrayList<String>();
    // parcours de l'ensemble des utilisateurs contenus dans lesUtilisateursQueJautorise
    for (int i = 0; i < lesUtilisateursQueJautorise.size(); i++)
    { Utilisateur unUtilisateur = lesUtilisateursQueJautorise.get(i);
      // ajout de l'utilisateur à la liste
      listeChaines.add(unUtilisateur.getPseudo());
    }
    // le constructeur ArrayAdapter reçoit le contexte, le style des items, et les données à afficher :
    ArrayAdapter<String> monAdapter = new ArrayAdapter<String>(
        getBaseContext(),
        android.R.layout.simple_list_item_single_choice,
        listeChaines);
    // affichage de la liste
    laListView.setAdapter(monAdapter);
} // fin de la fonction afficherLesUtilisateurs
```

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



## 6-4 Gestion du clic sur un item du ListView

On va maintenant gérer le clic sur un item du ListView en masquant la ListView et en affichant le layout **layoutSaisie** :



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

```
private String pseudoARetirer; // le pseudo de l'utilisateur à retirer
```

Complétez l'écouteur d'événement **laListViewOnItemClickListener** :

```
/** classe interne pour gérer le clic sur un item du ListView. */
private class laListViewOnItemClickListener implements AdapterView.OnItemClickListener {
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        // recherche du pseudo choisi à partir de la position de l'item choisi
        pseudoARetirer = listeChaines.get(position);
        laListView.setVisibility(View.GONE);
        layoutSaisie.setVisibility(View.VISIBLE);
        String msg = "Vous souhaitez retirer à " + pseudoARetirer + " l'autorisation de consulter vos parcours.";
        msg += " Si vous souhaitez qu'il soit averti par courriel, vous pouvez saisir un message.";
        msg += " En cas d'absence de message, aucun courriel ne sera envoyé.";
        textViewMessage.setText(msg);
    }
}
```

A yellow callout box points to the `position` parameter in the `onItemClick` method, stating: "position indique le numéro de l'item ayant reçu le clic".

Exécutez et testez.

## 6-5 Gestion du clic sur le bouton de validation

Cette action nécessite d'appeler le service web :

- **RetirerUneAutorisation** : pour retirer une autorisation à un utilisateur

A la suite des **import** existants, ajoutez l'**import** suivant :

```
import android.widget.Toast;
```

A la fin de l'activité, ajoutez la tâche asynchrone **TacheRetirerAutorisation** (vous pouvez vous inspirer du document "**5-3 (1) Projet TraceGPS - Dév appli android - ChangerMdp**").

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

private class TacheRetirerAutorisation extends AsyncTask<Void, Void, String> {

    La fonction onPreExecute doit désactiver les boutons boutonRetourner et boutonEnvoyer et démarrer
    l'affichage de l'objet progressBar.

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

    La fonction onPostExecute doit réactiver le bouton boutonRetourner, arrêter l'affichage de l'objet progressBar
    et afficher dans l'objet textViewMessage et dans un toast fugitif le message retourné par la méthode.

}
```

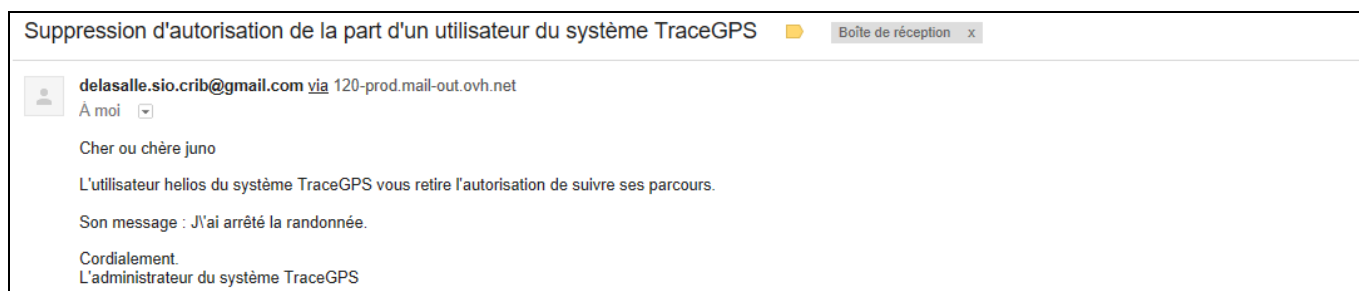
Complétez l'écouteur **buttonEnvoyerClickListener** pour appeler la tâche **TacheRetirerAutorisation** :

```
/** classe interne pour gérer le clic sur le bouton buttonEnvoyer. */
private class buttonEnvoyerClickListener implements View.OnClickListener {
    public void onClick(View v) {
        texteMessage = editTextMessage.getText().toString();
        // appel du service web RetirerAutorisation avec une tâche asynchrone
        new TacheRetirerAutorisation().execute();
    }
}
```

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



L'utilisateur visé par la suppression de l'autorisation doit recevoir un mail similaire à celui-ci :



Le demandeur peut vérifier en retournant sur la page de suppression d'autorisation :

