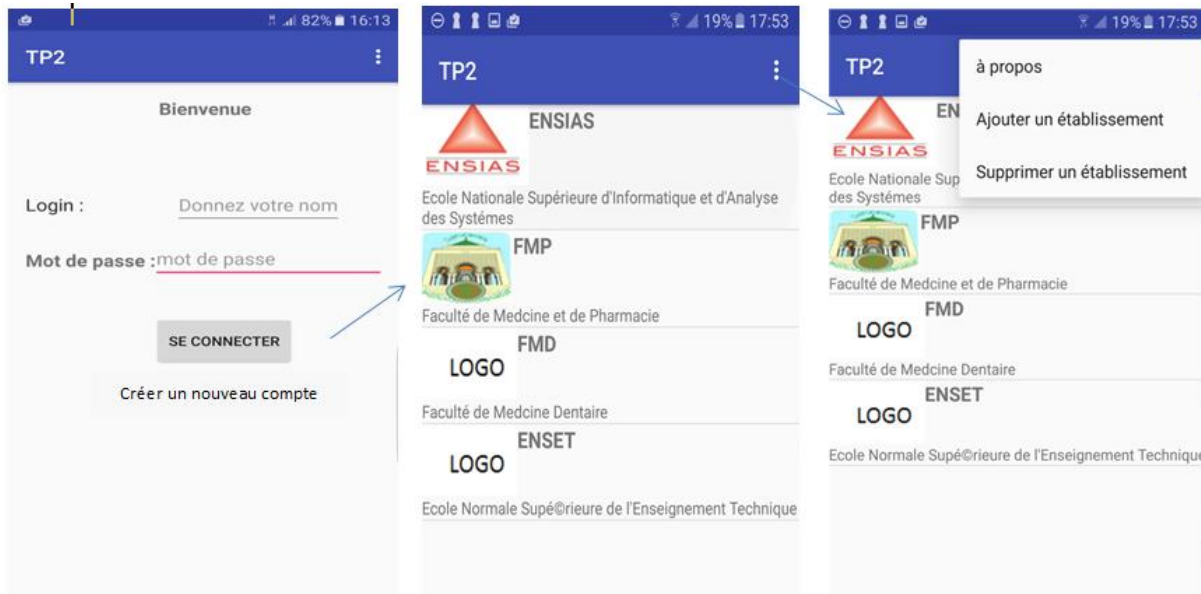


TP2



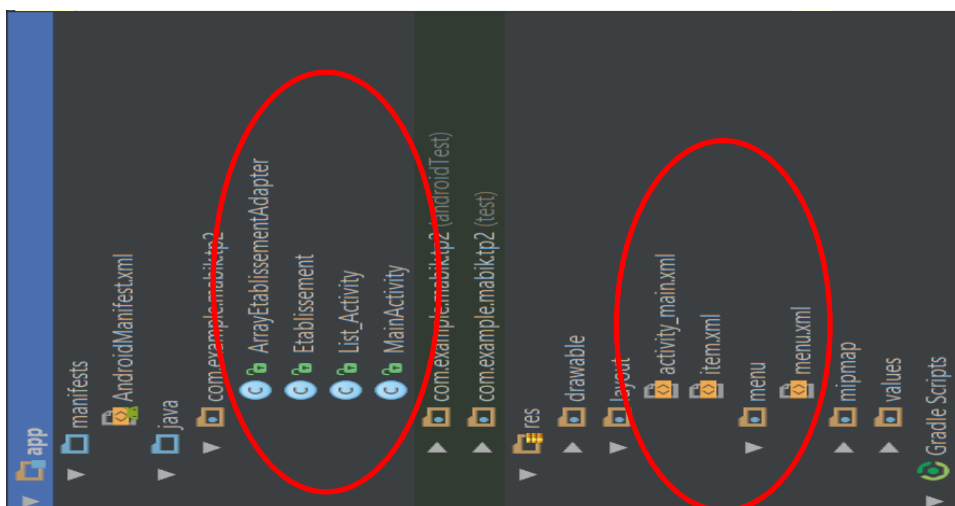
Interface 1

Interface 2 :ListView

Interface3 :Menu

L'objectif de ce TP est l'initiation à la création des interfaces graphiques en utilisant les layouts, vues, Listview, Menu et l'adaptateur personnalisé. Le résultat du TP est une application qui affiche la liste des établissements via l'IHM ListView.

La structure et les consignes présentées ci-dessous vous guident à réaliser ce TP.



NB : Vous pouvez procéder autrement pour avoir le même résultat.

Etape 1 :

1. Insérez les différents contrôles graphiques (TextView, EditText, Button) dans le layout principal "activity_main.xml"
2. Dans l'activité principale « Main_activity » associez le layout « activity_main.xml » par le setContentView
3. Ajoutez un nouveau layout "item.xml" contenant les Vues à afficher sur la listView. Les informations à afficher (logo, abréviation, nom) sont relatives aux établissements.
4. Ajouter la balise ListView dans le layout item.xml pour dérouler les éléments à afficher.

```
<ListView android:id="@+id/list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
/>
```

NB : mettez la ListView avant ou juste après le TextView

5. Créez la classe Etablissement.

```
public Etablissement (String label,String name, int im ){

    this.label= label;
    this.name= name;
    this.imag=im;

}
```

Ajoutez les méthodes pour récupérer les informations relatives à l'établissement (getLabel, getdesc,getimage) qui seront utilisées par l'adaptateur.

6. Ajouter un adaptateur personnalisé «ArrayEtablissementAdapter » qui va jouer le rôle d'interface entre les données et les Vues. Utilisez la classe déposée sur la plate-forme.
7. Ajouter l'activité List_Activity.java pour lier notre liste à l'adaptateur personnalisé. Cette activité est démarrée grâce au bouton « se connecter » de la classe Main_activity.main.

```
public class List_Activity extends AppCompatActivity {
    ListView l;
    List<Etablissement> Etablissements;
    Etablissement Etat;
    ArrayEtablissementAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        List<Etablissement> Etablissements = new
```

```

ArrayList<Etablissement>();
    Etablissements.clear();
    Etablissements.add(new Etablissement("ENSIAS", "Ecole Nationale
Supérieure d'Informatique et d'Analyse des Systèmes", R.drawable.ens));
    Etablissements.add(new Etablissement("FMP", "Faculté de Medcine
et de Pharmacie", R.drawable.md));
    .... //ajouter les autres établissements

    //liason entre la source des données (Etablissements) et les
    controles du layout item Ã affichÃ
    adapter = new ArrayEtablissementAdapter(this, R.layout.item,
    Etablissements);

    l= (ListView) findViewById(R.id.list);
    //// Binding resources Array to ListAdapter
    l.setAdapter(adapter);
    ...

```

8. Pour gérer les interactions avec les vues du ListView :

```

protected void onItemClick(ListView l, View v, int position, long id)
{
    Etablissement item = (Etablissement) l.getAdapter().getItem(position);
    Toast.makeText(this, item.getLabel() + "selected",
    Toast.LENGTH_LONG).show();
}

```

9. Pour gérer le menu à choix :

1. Créez un menu (via res/New/Android ressource file) puis ajoutez les items

```

<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item android:id="@+id/item1" android:title="à propos"/>
<item android:id="@+id/item2" android:title="Ajouter un établissement"/>
<item android:id="@+id/item3" android:title="Supprimer un établissement"/>
</menu>

```

2. Ajoutez la méthode « onCreateOptionsMenu » dans l'activité List_Activity

```

public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    return true;
}

```

3. Pour gérer les interactions avec le menu, ajouter la méthode

« onOptionsItemSelected »

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    switch(item.getItemId()) {
        case 1: Toast.makeText(this, "à props", 1000).show(); break;
        case 2: Toast.makeText(this, "Ajouter un etablissement", 1000).show(); break;
        case 3: Toast.makeText(this, "supprimer un etablissement", 1000).show(); break;
    }

    return super.onOptionsItemSelected(item);
}
```