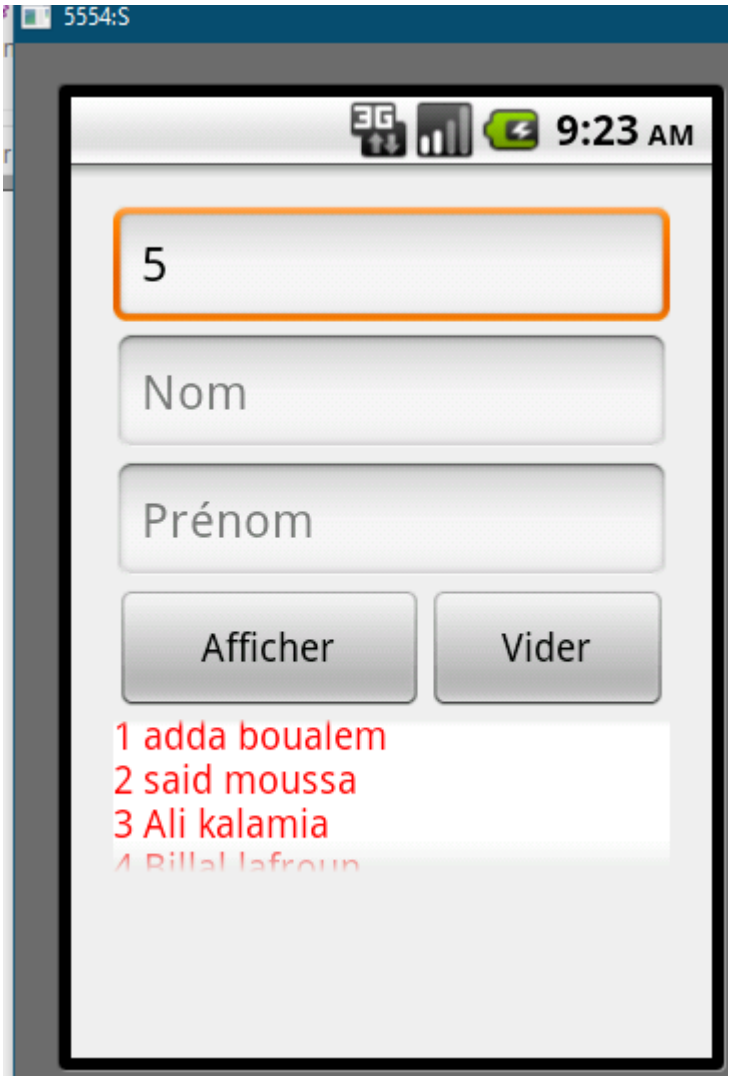
Applications Mobile :TP 3 (3ème année, Licence SI)

Objectif : Utiliser les informations obtenues lors des travaux pratiques précédents pour créer une application complète.

Contexte :

Dans ce TP nous allons saisir et afficher les données dans un widget « TextView » à la place d'une liste.



Travail demandé :

1. Créer le Projet « AffichageDonnees ».

2. Partie IHM (XML)

a. Reproduire l'IHM est définie dans la figure suivante

b. Types de données :
   i. Matricule de type number

ii. Nom de type Capcaracters
iii. Prénom de type Capword
iv. Changer la propriété **Hint** des **EditText** correspondants par : Matricule, Nom et prénom respectivement.
v. Changer la propriété **Text** des Buttons respectivement par : **Afficher** et **Vider L.**
vi. Ajouter un **scrollView** et mettre le **TextView** dedans.
vii. Changer la couleur de la propriété «background » du **TextView** par la couleur Blanc « #ffffffff »
viii. Ajouter la propriété « `android:singleLine="true"` » dans toutes les zones de saisies.

3. Partie Source

i. Supprimer la partie source du Menu
ii. Créer les objets qui peuvent subir l'interaction de l'utilisateur

```
public class MainActivity extends Activity {
EditText mat, nom, prenom;
TextView liste;
Button aff, fermer;
int i;
String j;
```

iii. Instancier les objets crées (déclarés)

```
  mat = (EditText) findViewById(R.id.editText1);
  nom = (EditText) findViewById(R.id.editText2);
  prenom = (EditText) findViewById(R.id.editText3);
  liste = (TextView) findViewById(R.id.textView1);
  aff = (Button) findViewById(R.id.button1);
  fermer = (Button) findViewById(R.id.button2);
  mat.setText("1");
  fermer.setOnClickListener(new OnClickListener() {
```

iv. Vider et Incrémenter le Matricule à chaque affichage (enregistrement) par 1

Soit la méthode suivante :

```
 private void vider(){
     j = mat.getText().toString();
     mat.setText(null);
     nom.setText(null);
     prenom.setText(null);

     Log.i("Connexion", j);
     i=Integer.valueOf(j);
 j=String.valueOf(i);
 i=Integer.valueOf(j)+1;
 j=String.valueOf(i);
     Log.i("Connexion", String.valueOf(i));  Log.i("Connexion", j);
 //  i=Integer.valueOf(mat.getText().toString())+1;
     mat.setText(j);
     mat.requestFocus();
 }
```

v. Vider la liste d'affichage (initialisation)

Soit la méthode suivante :

```
private void viderliste(){
    liste.setText(null);
    mat.setText("1");
    mat.requestFocus();
}
```

vi. La méthode pour afficher les données saisies dans la liste d'affichage est définie comme suit :

```
aff.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        liste.setTextColor(Color.RED);
        liste.append(mat.getText()+" "+nom.getText()+" "+ prenom.getText()+"\n");
        vider();
    }
});

}
```

Questions :

a. Comment redéfinir les méthodes des évènements **OnClick** pour permettre les appelées par XML.

b. Quelle est l'objectif d'utilisation les instructions suivantes :

i. La méthode `setContentView(R.layout.activity_main);`
ii. La méthode append() ;
iii. La méthode setTextColor() ;
iv. La méthode log.i() ;
v. La méthode Color ;
vi. L'instruction (propriété) : `android:background="#ffffffff"`
vii. L'instruction (propriété) : `android:singleLine="true"`
viii. L'instruction (propriété) : `<requestFocus />`
ix. Le Conteneur : `ScrollView`