移动平台应用开发

实
验
报
告

学　　院　计算机学院

年　　级　16 级

班　　级　3 班

学　　号　1613415

姓　　名　潘巧巧

2019 年 **4** 月 **15** 日

目录

一、实验目的

本次实验的目的是熟悉 Android 开发中的 UI 设计，包括了解和熟悉常用控件的使用、界面布局和事件处理等内容。具体包括：

1. 熟悉和掌握界面控件设计
2. 了解 Android 界面布局
3. 掌握控件的事件处理

二、实验内容

（1）常用控件（本次实验用到的控件红色加粗，共涉及 3 类、6 种、12 个控件）

1、文本框：**TextView**、**EditText**

2、按钮：**Button**、**RadioButton**、**RadioGroup**、CheckBox、ImageButton

3、列表：List、ExpandableListView、Spinner、AutoCompleteTextView、GridView、ImageView

4、进度条：**ProgressBar**、ProgressDialog、SeekBar、RatingBar

5、选择器：DatePicker、TimePicker

6、菜单：Menu、ContentMenu

7、对话框：Dialog、ProgressDialog

（2）常用界面布局（本次实验用到的控件红色加粗，共 2 个、2 种）

1、FrameLayout：最简单的一个布局对象。它里面只显示一个显示对象。Android 屏幕元素中所有的显示对象都将会固定在屏幕的左上角，不能指定位置。但允许有多个显示对象，但后一个将会直接在前一个之上进行覆盖显示，把前一个部份或全部挡住（除非后一个是透明的）。

2、**LinearLayout**：以单一方向对其中的显示对象进行排列显示，如以垂直排列显示，则布局管理器中将只有一列；如以水平排列显示，则布局管理器中将只有一行。同时，它还可以对个别的显示对象设置显示比例。

3、TableLayout：以拥有任意行列的表格对显示对象进行布局，每个显示对象被分配到各自的单元格之中，但单元格的边框线不可见。

4、AbsoluteLayout：允许以坐标的方式，指定显示对象的具体位置，左上角的坐标为(0, 0)，向下及向右，坐标值变大。这种布局管理器由于显示对象的位置定死了，所以在不同的设备上，有可能会出现最终的显示效果不一致。

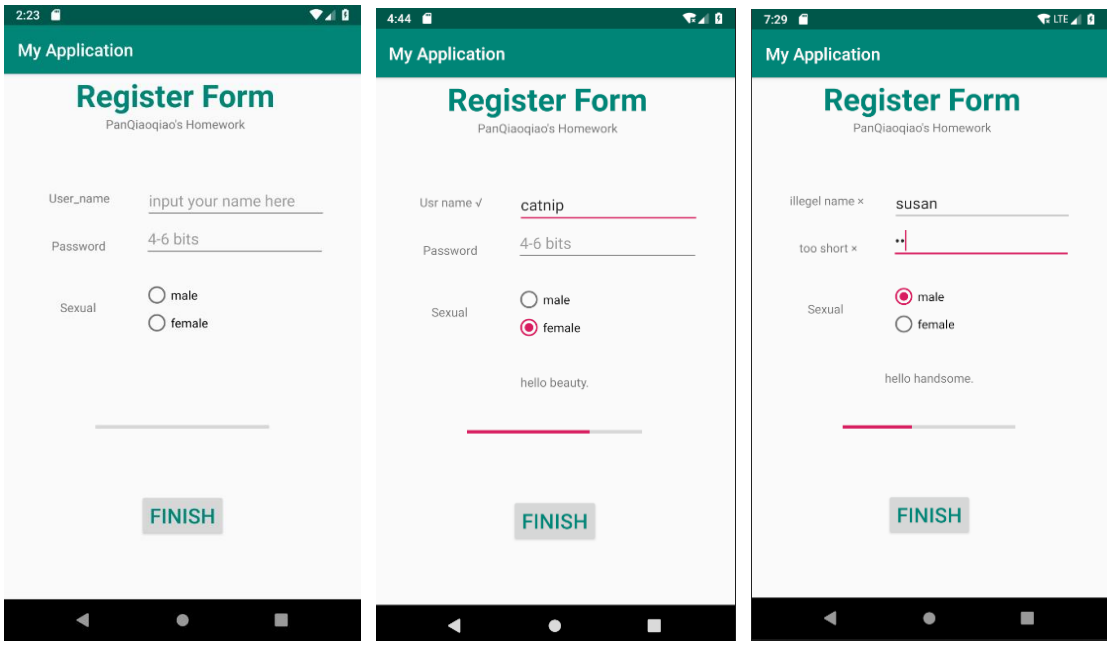
5、RelativeLayout：允许通过指定显示对象相对于其它显示对象或父级对象的相

对位置来布局。如一个按钮可以放于另一个按钮的右边，或者可以放在布局管理器的中央。

6、**ConstraintLayout**：它的出现是为了解决复杂布局时，布局嵌套（布局内的布局）过多的问题（嵌套布局会增加绘制界面所需的时间）。它可以根据同级视图和父布局的约束条件为每个视图定义位置，类似于 RelativeLayout 所有视图都是根据兄弟视图和父级布局之间的关系来布局的，但是与 RelativeLayout 相比，它更加灵活，更易于使用。

三、实验步骤及实验结果

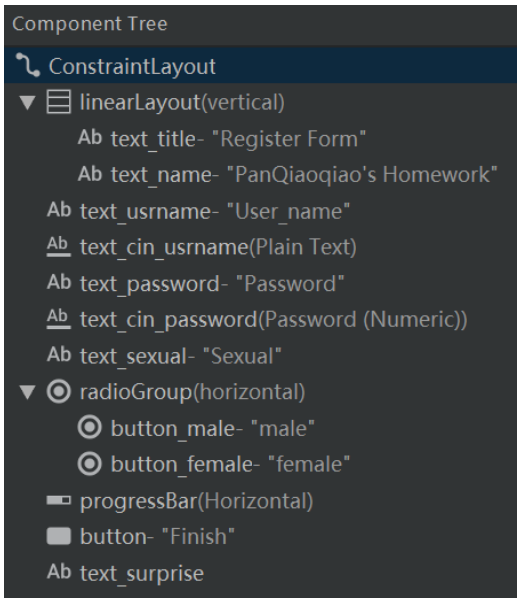
【实验成果图】



（左 1 为初始界面，左 2 左 3 为进行一定操作后的界面。）

【本次实验结果+步骤】

实验报告结构：各个控件的功能+实现方法



（所有控件及布局）

```
public class MainActivity extends AppCompatActivity {
    Vector usr = new Vector();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        usr.add("susan");
        usr.add("sam");
```

（全局数组 usr 存储用户名信息）

（1） **布局**

实验总体布局为 ConstraintLayOut，其中标题和署名放在一个 LinearLayout 中。

（2） **控件 1（TextView）：text_title —— 界面标题**

```
<TextView
    android:id="@+id/text_title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Register Form"
    android:textAlignment="center"
    android:textColor="@color/colorPrimary"
    android:textSize="36sp"
    android:textStyle="bold" />
```

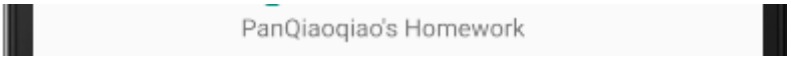
（XML 代码）

（成果展示）

（3） **控件 2（TextView）：text_name —— 作者署名**

```
<TextView
    android:id="@+id/text_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="PanQiaoqiao's Homework"
    android:textAlignment="center" />
```

（XML 代码）



（成果展示）

（4） **控件 3（TextView）：text_usrname —— 引导用户注册用户名**

功能：①引导用户注册，一开始 text 为 *“user_name”*；

②当用户输入自定义的用户名时，与数据库中的已有用户名进行对比，如果已有别的用户注册该用户名（如预置的 susan 和 sam），则 text 变成 *“illegel name ×”*；

③如果用户输入了名字之后又删除清空，则 text 变成 *“cannot be NULL ×”*；

以上三种情况不会使屏幕下方注册进度条增加。

④如果用户输入合法用户名，则 text 变成 *“Usr name ✓”*，并且屏幕下方注册进度条增加 30（满分为 100）。

具体实现通过监听输入框，Java 代码见控件 4。

```
<TextView
    android:id="@+id/text_usrname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="55dp"
    android:layout_marginTop="41dp"
    android:layout_marginEnd="44dp"
    android:text="User_name"
    app:layout_constraintEnd_toStartOf="@+id/text_cin_usrname"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout" />
```

（XML 代码）

（5） 控件 4（Edit Text）text_cin_usrname —— 用户输入框

功能：用户在该输入框中输入自定义的用户名。该用户输入框在 Java 代码中被监听，以实现控件 3（TextView）和控件 11（ProgressBar）根据用户输入实现变化。已经被其他用户注册过的用户名存放在 usr 中，不能再被注册。

```
<EditText
    android:id="@+id/text_cin_usrname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="31dp"
    android:layout_marginEnd="40dp"
    android:ems="10"
    android:hint="input your name here"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout" />
```

（XML 代码）

```java
//监听用户名输入框
EditText name = (EditText) findViewById(R.id.text_cin_usrname);
name.addTextChangedListener(new TextWatcher() {
    @Override   public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    @Override   public void onTextChanged(CharSequence s, int start, int before, int count) { }
    @Override
    public void afterTextChanged(Editable s) { //屏蔽回车 中英文空格
        String name1=s.toString();
        int usr_size=usr.size();
        boolean flag=false;
        for(int i=0;i<usr_size;i++) {
            if (name1.equals(usr.elementAt(i))) {
                TextView name = (TextView) findViewById(R.id.text_usrname);
                name.setText("illegel name ×");
                add();
                flag = true;
            }
        }
        if(flag==false) {
            if (!name1.isEmpty()) {
                TextView name = (TextView) findViewById(R.id.text_usrname);
                name.setText("Usr name √");
                add();
            } else {
                TextView name = (TextView) findViewById(R.id.text_usrname);
                name.setText("cannot be NULL ×");
                add();
```

（Java 代码实现对控件的监听）

（6） 控件 5（TextView）：text_password —— 引导用户设置密码

功能：初始化时该控件的 text 显示为 *“password”*，当用户在控件 6 中设置密码时，如果用户设置的密码低于 4 位，该控件的 text 将变为 *“too short ×”*，并且下方进度条不会变化。当用户设置的密码合法时，该控件的 text 变为 *“Password √”*，下方进度条增加 30（满分 100）。相关实现需要监控控件 6。

```
<TextView
    android:id="@+id/text_password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="55dp"
    android:layout_marginTop="36dp"
    android:layout_marginEnd="43dp"
    android:text="Password"
    app:layout_constraintEnd_toStartOf="@+id/text_cin_password"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/text_usrname" />
```

（XML 代码）

（7） 控件 6（Edit Text）：text_cin_password —— 用户密码输入框

功能：用户在此输入密码。通过 maxLength 属性将密码位数限制在 6 位以内。预设密码应大于 4 位，以及实现控件 5 的变化和控件 11 进度条变化的相关功能均通过在 Java 中对该控件进行监控实现。

```
<EditText
    android:id="@+id/text_cin_password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="41dp"
    android:layout_marginBottom="31dp"
    android:clickable="false"
    android:ems="10"
    android:hint="4-6 bits"
    android:inputType="numberPassword"
    android:maxLength="6"
    app:layout_constraintBottom_toTopOf="@+id/radioGroup"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/text_usrname" />
```

（XML 代码）

```
    //监听密码输入
    EditText password = (EditText) findViewById(R.id.text_cin_password);
    password.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {
        }
        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
        }
        @Override
        public void afterTextChanged(Editable s) { //屏蔽回车 中英文空格
            String psw=s.toString();
            if(psw.length()>=4) {
                TextView name= (TextView) findViewById(R.id.text_password);
                name.setText("Password √");
                add();
            }
            else
            {
                TextView name= (TextView) findViewById(R.id.text_password);
                name.setText("too short ×");
                add();
            }
        }
    });
};
```

（Java 实现监听控件 6）

（8） 控件 7（TextView）：text_sexual —— 引导用户选择性别

```
<TextView
    android:id="@+id/text_sexual"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="55dp"
    android:layout_marginTop="52dp"
    android:layout_marginEnd="46dp"
    android:text="Sexual"
    app:layout_constraintEnd_toStartOf="@+id/radioGroup"
    app:layout_constraintHorizontal_bias="0.555"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/text_password" />
```

（XML 代码）

（9） 控件 8（RadioGroup）—— 容纳两个性别选项，使他们只能单选

```
<RadioGroup
    android:id="@+id/radioGroup"
    android:layout_width="210dp"
    android:layout_height="64dp"
    android:layout_marginTop="31dp"
    android:layout_marginEnd="41dp"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/text_password">
```

（XML 代码）

（10） 控件 9、10（RadioButton）：button_male、button_female —— 性别选项

在 RadioGroup 中的两个 RadioButton 只能单选。当用户选择其中一个时，会分别触发两个 Java 代码中的函数 onClick_male()或 onClick_female，从而触发控件 11 进度条和控件 13TextView 的变化。

```
<RadioButton
    android:id="@+id/button_male"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:onClick="onClick_male"
    android:text="male" />

<RadioButton
    android:id="@+id/button_female"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:onClick="onClick_female"
    android:text="female" />
```

（XML 代码）

```
public void onClick_male(View v)
{
    TextView name= (TextView) findViewById(R.id.text_surprise);
    name.setText("hello handsome.");
    ProgressBar bar=(ProgressBar)findViewById(R.id.progressBar);
    add();
    if(bar.getProgress()==60)
        bar.setProgress(100);
}

public void onClick_female(View v)
{
    TextView name= (TextView) findViewById(R.id.text_surprise);
    name.setText("hello beauty.");
    ProgressBar bar=(ProgressBar)findViewById(R.id.progressBar);
    add();
    if(bar.getProgress()==60)
        bar.setProgress(100);
}
```

（Java 代码）

（11） 控件 11（ProgressBar）—— 告诉用户填写注册信息的进度。

总长度为 100，合法填写用户名增加 30，合法填写密码增加 30，选择性别增加 40。通过 Java 中的函数 add()调用控件 3、5、13 查看用户当前的填写状态并计算相关值。该函数在用户填写用户名、密码、性别时被监听函数和 click 函数调用。

```
<ProgressBar
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="200dp"
    android:layout_height="55dp"
    android:layout_marginStart="105dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="106dp"
    android:max="100"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/text_surprise" />
```

（XML 代码）

```
public void add() {
    int sum = 0;

    TextView usrname = (TextView) findViewById(R.id.text_usrname);
    String usrname_copy = usrname.getText().toString();
    if (usrname_copy.equals("Usr name √"))
        sum += 30;

    TextView psw = (TextView) findViewById(R.id.text_password);
    String psw_copy = psw.getText().toString();
    if (psw_copy.equals("Password √"))
        sum += 30;

    TextView surprise = (TextView) findViewById(R.id.text_surprise);
    String surprise_copy = surprise.getText().toString();
    if (surprise_copy.equals("hello handsome.") || surprise_copy.equals("hello beauty."))
        sum += 40;

    ProgressBar bar = (ProgressBar) findViewById(R.id.progressBar);
    bar.setProgress(sum);
}
```

（Java 代码实现进度条更新）

（12） 控件 12（Button）—— 用户点击确认提交 register form

用户通过点击该button提交注册表，该button与Java中的函数onClick_btn()绑定。函数读取控件 11 中进度条的值，如果进度条满则弹出一个*“Register successful!”*的 toast，否则弹出 *“Register failed!”*的 toast。注意此处保存用户信息到 usr 中并清空和复原所有控件。

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="158dp"
    android:layout_marginTop="48dp"
    android:layout_marginEnd="165dp"
    android:layout_marginBottom="117dp"
    android:onClick="onClick_btn"
    android:text="Finish"
    android:textColor="@color/colorPrimary"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.181"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/progressBar"
    app:layout_constraintVertical_bias="0.0" />
```

（XML 代码）

```
public void onClick_btn(View v)
{
    ProgressBar bar = (ProgressBar) findViewById(R.id.progressBar);
    if(bar.getProgress()==100) {
        Toast.makeText(v.getContext(), text: "Register successful!", Toast.LENGTH_LONG).show();
        TextView name= (TextView) findViewById(R.id.text_cin_usrname);
        String name_copy=name.toString();
        usr.add(name_copy);
        name.setText("");
        TextView psw= (TextView) findViewById(R.id.text_cin_password);
        psw.setText("");
        TextView spr= (TextView) findViewById(R.id.text_surprise);
        spr.setText("");
        TextView name_left= (TextView) findViewById(R.id.text_usrname);
        name_left.setText("User_name");
        TextView psw_left= (TextView) findViewById(R.id.text_password);
        psw_left.setText("Password");
        RadioGroup sexual=(RadioGroup)findViewById(R.id.radioGroup);
        sexual.clearCheck();
        bar.setProgress(0);
    }
    else
        Toast.makeText(v.getContext(), text: "Register failed!",Toast.LENGTH_LONG).show();
}
```

（Java 代码）

（13） 控件 13（TextView）：text_suprise —— 对用户的性别选择做出反馈

初始时该控件的 Text 为空，若用户勾选性别 male，Text 赋为 *“hello handsome.”*，若用户勾选性别为 female，Text 赋为 *“hello beauty.”*。具体实现见控件 9、10 对应的 Java 代码。

```
<TextView
    android:id="@+id/text_surprise"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="176dp"
    android:layout_marginTop="37dp"
    android:layout_marginEnd="176dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/radioGroup" />
```

（XML 代码）

四、实验遇到的问题及其解决方法

（1）**问题**：点击运行时，程序持续处于 installing APK 状态无法运行。

解决方案：查找资料后发现网上调节 setting 的办法均无效，有一篇帖子说真机调试遇到该问题重启真机即可，但处于模拟机状态下一直死机无法重启。然后下载了新的 Virtual Device，重新运行可以继续进行实验。但原来的 VD 仍无法启动。

（2）**问题**：点击运行时，程序持续处于 waiting-for-target-device-to-come-online 状态无法运行。

解决方案：界面右上角 AVD Manager→Action→三角符号→Cold Boot Now→重新运行程序。

（3）**问题**：调试程序时无法在控制台打印调试信息

解决方案：代码中使用 System.out.println();函数，运行后在左下角找到 Logcat，搜索框中输入 println 可以过滤掉多余信息并展示打印信息。

五、实验结论

实验感想：

① 模拟机太难用了，各种各样的问题，下次试试真机调试也许不会这么麻烦。

② Design 模式拖控件设计 UI 效率很高。

③ 学编程语言最好的方法是实践，作为 Java 的初学者上手实践更好理解。