



中山大學  
SUN YAT-SEN UNIVERSITY

# 《手机平台应用开发》

## 数据存储（一）

学 院 名 称 ： 数据科学与计算机学院

---

成 员 ： 陈伟宸

---

时 间 ： 2016 年 11 月 9 日

---

## 一、实验环境

操作系统：Windows 10

IDE：Android Studio 2.1.2

## 二、实验过程

1. 新建一个工程，添加 MainActivity.java 及其布局文件 activity\_file\_editor.xml，FileEditorActivity.java 及其布局文件 activity\_file\_editor.xml

2. 设计 MainActivity 的布局。

添加两个 EditText，用于输入密码和确认密码

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:id="@+id/new_password"
    android:layout_gravity="center_horizontal"
    android:hint="New Password"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"/>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:id="@+id/confirm_password"
    android:hint="Confirm Password"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"/>
```

添加 OK 按钮和 CLEAR 按钮

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_gravity="left">
    <Button
        android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="OK"
        android:layout_marginLeft="10dp"/>
    <Button
        android:id="@+id/clear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CLEAR"/>
</LinearLayout>

```

### 3. 设计 FileEditorActivity 的布局。

添加一个 EditText 用于输入文本 ,将 gravity 设为 left|top 使光标出现在左上角

```

<EditText
    android:id="@+id/notes"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textMultiLine"
    android:gravity="left|top"
    android:lines="25"/>

```

使用 RelativeLayout 对三个 Button 排版

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp">
    <Button
        android:id="@+id/save"
        android:layout_alignParentLeft="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SAVE"/>
    <Button
        android:id="@+id/load"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="LOAD"/>
    <Button
        android:id="@+id/editor_clear"
        android:layout_alignParentRight="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CLEAR"/>
</RelativeLayout>

```

#### 4. 实现 MainActivity。

获得所有控件并新建一个 Toast 和取得 SharedPreferences

```

linearLayout = (LinearLayout)findViewById(R.id.layout);
new_password = (EditText)findViewById(R.id.new_password);
confirm_password = (EditText)findViewById(R.id.confirm_password);
ok = (Button)findViewById(R.id.ok);
clear = (Button)findViewById(R.id.clear);
toast = Toast.makeText(MainActivity.this, "", Toast.LENGTH_SHORT);
sharedPref = getApplicationContext().getSharedPreferences("MY_PREFERENCE", Context.MODE_PRIVATE);

```

判断在 SharedPreferences 中是否已经存在 PASSWORD 的 key ,  
如果存在 , 删除用于确认密码的输入框

```

if (sharedPref.contains("PASSWORD")) {
    linearLayout.removeView(confirm_password);
    new_password.setHint("Password");
}

```

实现点击 OK 按钮事件的回调函数，如果 SharedPreferences 中没有 PASSWORD 的 key，判断输入是否正确

```
if (!sharedPref.contains("PASSWORD")) {
    if (new_password.getText().toString().isEmpty() || confirm_password.getText().toString().isEmpty()) {
        toast.setText("Password can not be empty.");
        toast.show();
    } else if (!new_password.getText().toString().equals(confirm_password.getText().toString())) {
        toast.setText("Password Mismatch.");
        toast.show();
    } else {
        SharedPreferences.Editor editor = sharedPref.edit();
        editor.putString("PASSWORD", new_password.getText().toString());
        editor.commit();
        Intent intent = new Intent(MainActivity.this, FileEditorActivity.class);
        startActivity(intent);
    }
}
```

如果已经存在，在 SharedPreferences 中新建一个 PASSWORD 的 key 用于保存输入的密码并跳转到 FileEditorActivity

```
else {
    if (new_password.getText().toString().equals(sharedPref.getString("PASSWORD", null))) {
        Intent intent = new Intent(MainActivity.this, FileEditorActivity.class);
        startActivity(intent);
    } else {
        toast.setText("Invalid Password.");
        toast.show();
    }
}
```

实现点击 CLEAR 按钮事件的回调函数，将输入框都清空

## 5. 实现 FileEditorActivity。

获得所有控件并新建一个 Toast

```
save = (Button)findViewById(R.id.save);
load = (Button)findViewById(R.id.load);
clear = (Button)findViewById(R.id.editor_clear);
notes = (EditText)findViewById(R.id.notes);
toast = Toast.makeText(FileEditorActivity.this, "", Toast.LENGTH_SHORT);
```

实现点击 SAVE 按钮事件的回调函数，尝试将输入的文本保存到文件

中

```
save.setOnClickListener((view) -> {  
    try (FileOutputStream fileOutputStream = openFileOutput(FILE_NAME, MODE_PRIVATE)) {  
        fileOutputStream.write(notes.getText().toString().getBytes());  
        toast.setText("Save successfully.");  
        toast.show();  
    } catch (IOException ex) {  
        Log.e("TAG", "Fail to save file.");  
        toast.setText("Fail to save file");  
        toast.show();  
    }  
});
```

实现点击 LOAD 事件的回调函数，尝试从文件中读出已保存的内容

```
load.setOnClickListener((view) -> {  
    try (FileInputStream fileInputStream = openFileInput(FILE_NAME)) {  
        byte[] note = new byte[fileInputStream.available()];  
        fileInputStream.read(note);  
        notes.setText(new String(note));  
        toast.setText("Load successfully.");  
        toast.show();  
    }  
    catch (IOException ex) {  
        Log.e("TAG", "Fail to read file.");  
        toast.setText("Fail to load file.");  
        toast.show();  
    }  
});
```

实现点击 CLEAR 事件的回调函数，将输入框清空

```
clear.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        notes.setText("");  
    }  
});
```

6. 在 AndroidManifest.xml 中注册各 Activity 并设置 MainActivity 的 windowsSoftInputMode 为 adjustResize

```

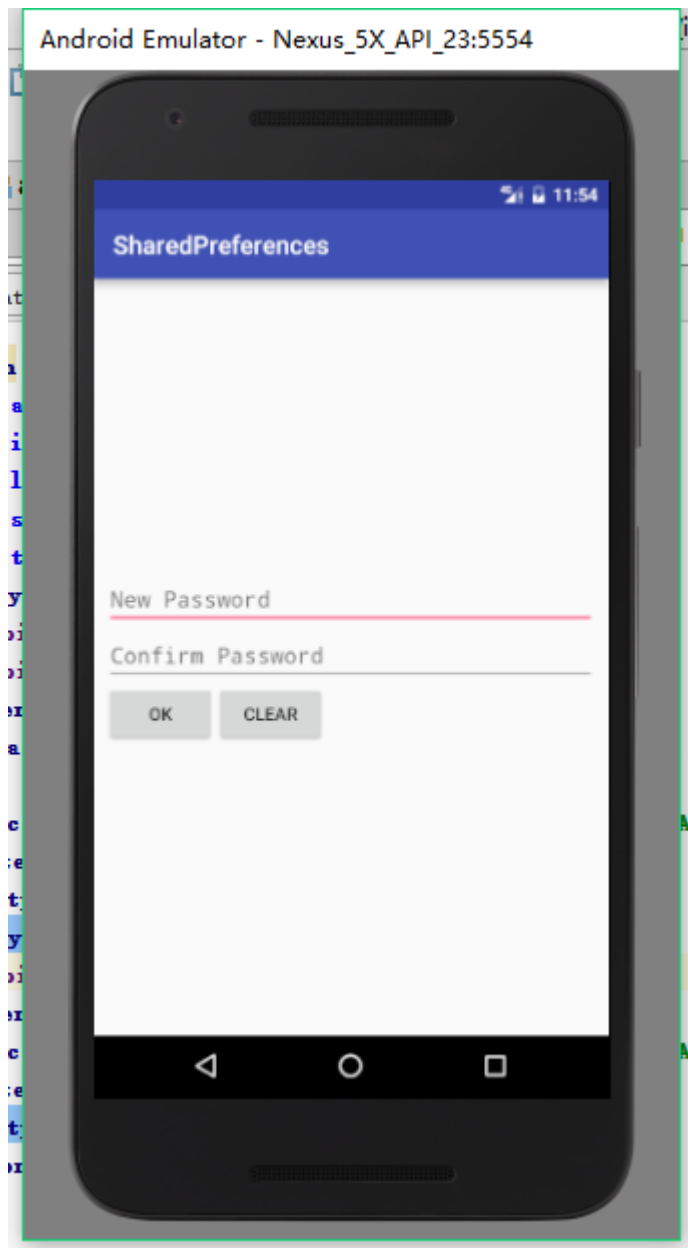
<activity android:name=".MainActivity"
    android:noHistory="true"
    android:windowSoftInputMode="adjustResize">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".FileEditorActivity"
    android:noHistory="true">
    <intent-filter>
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>

```

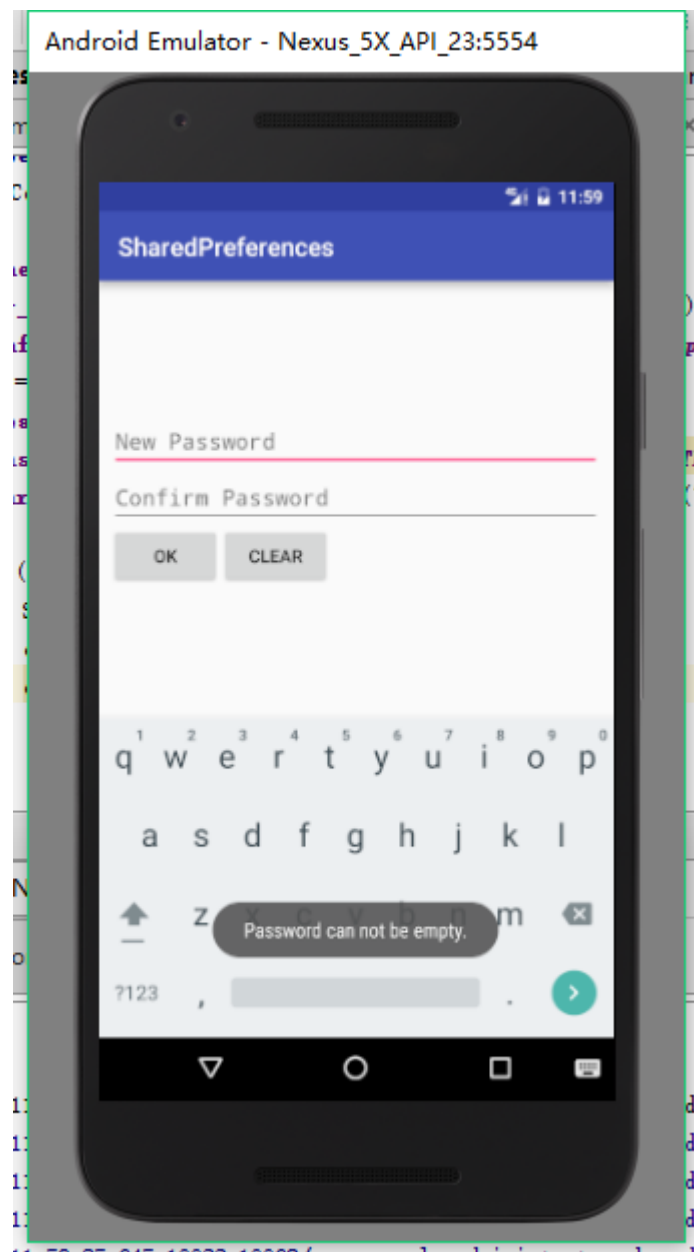
### 三、实验结果

#### 1. 运行程序

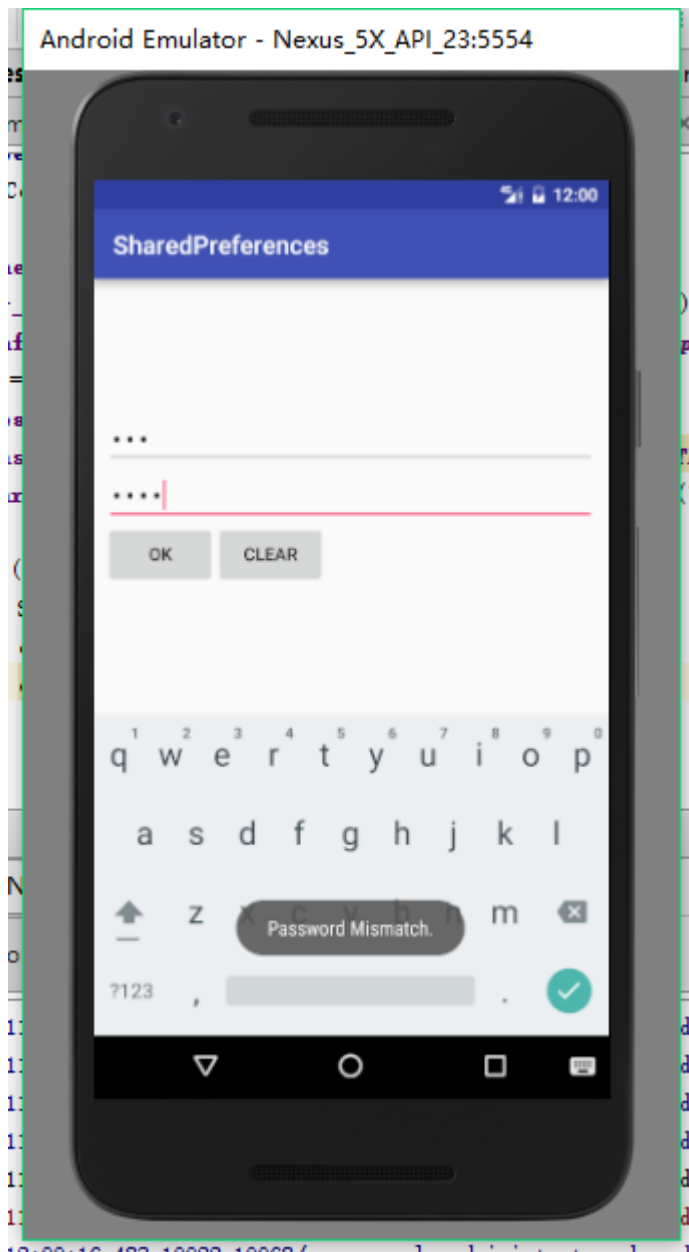


2. 直接点击 OK 按钮，提示 Password can not be empty

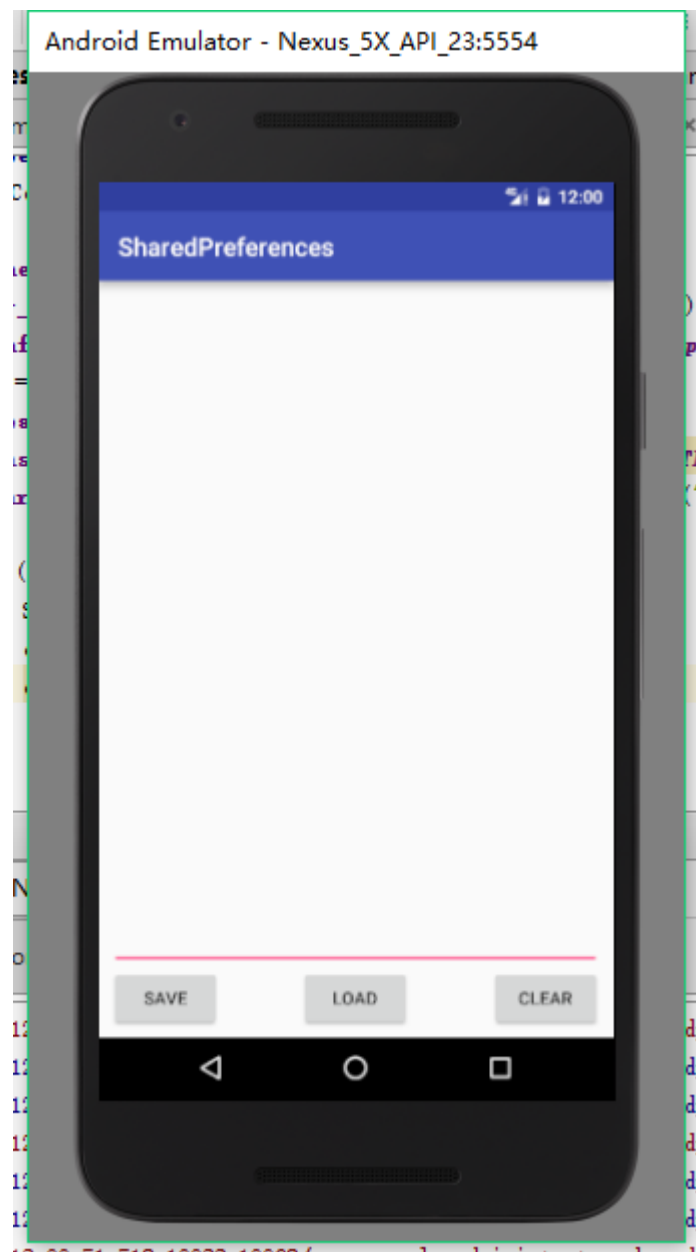




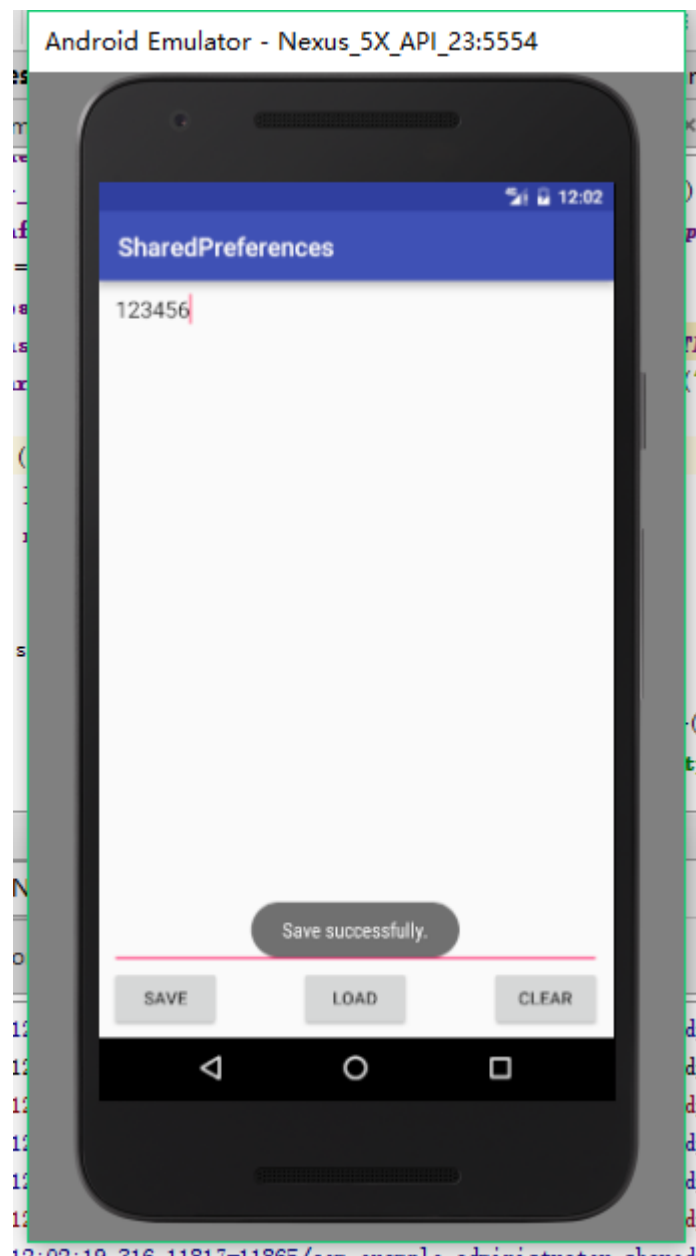
3. 输入不同的密码，提示 Password Mismatch



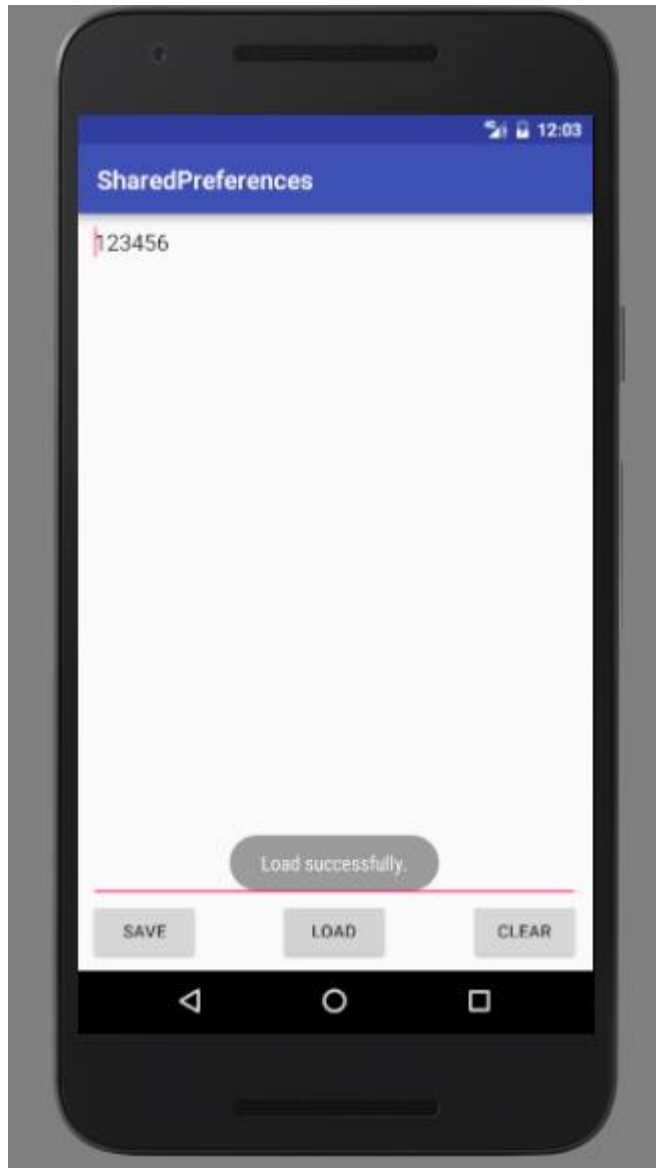
4. 输入相同的密码，点击 OK，成功跳转



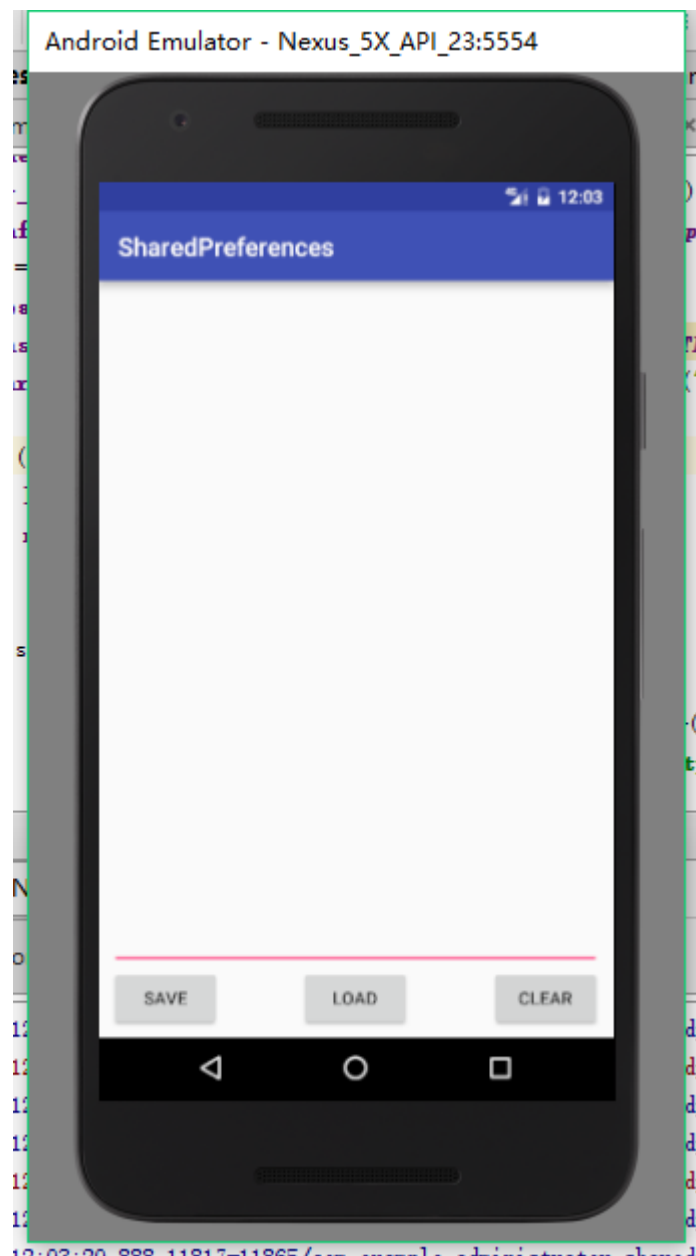
5. 输入 123456 , 并点击 SAVE , 提示 Save Successfully



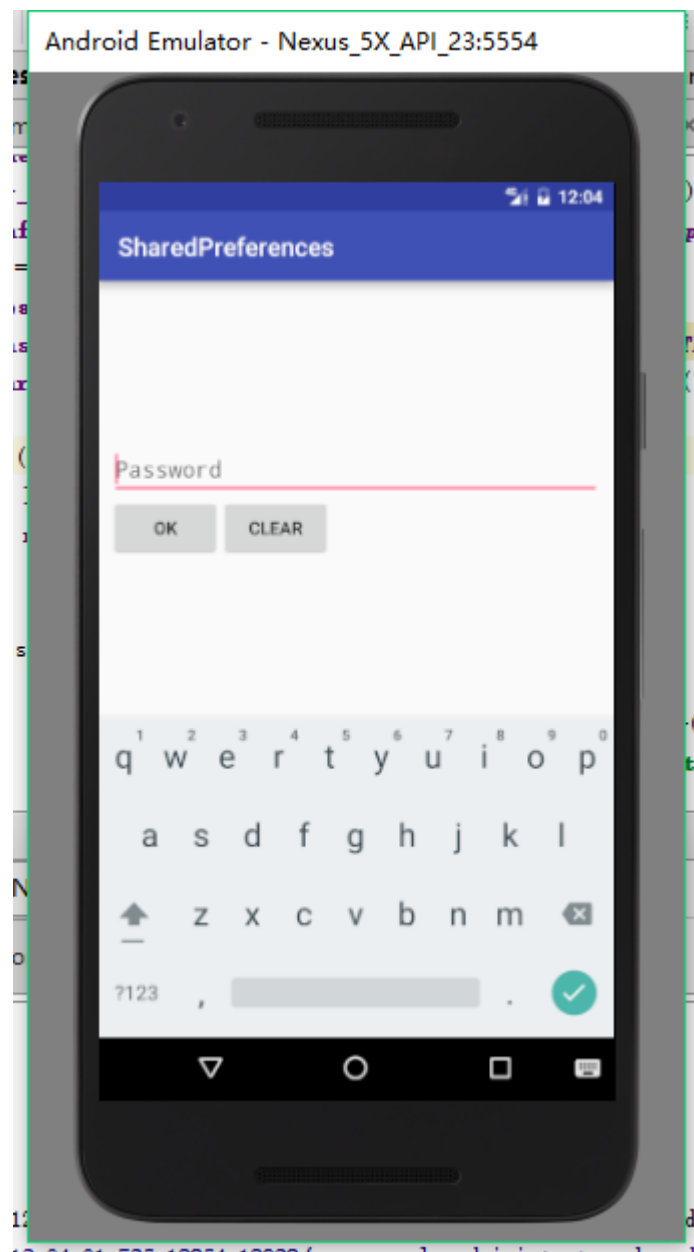
6. 将 123456 删除，再点击 LOAD，LOAD 成功



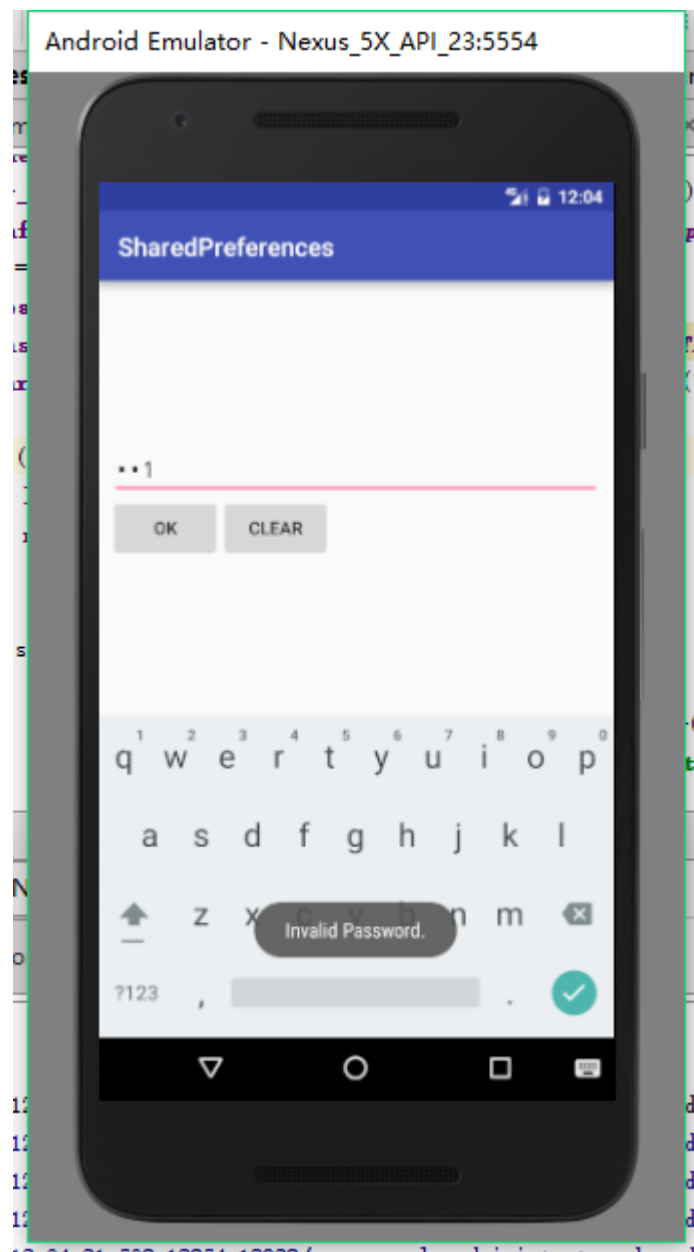
7. 点击 CLEAR , 成功清除文本



8. 重新运行程序，程序主页面只剩下一个输入框

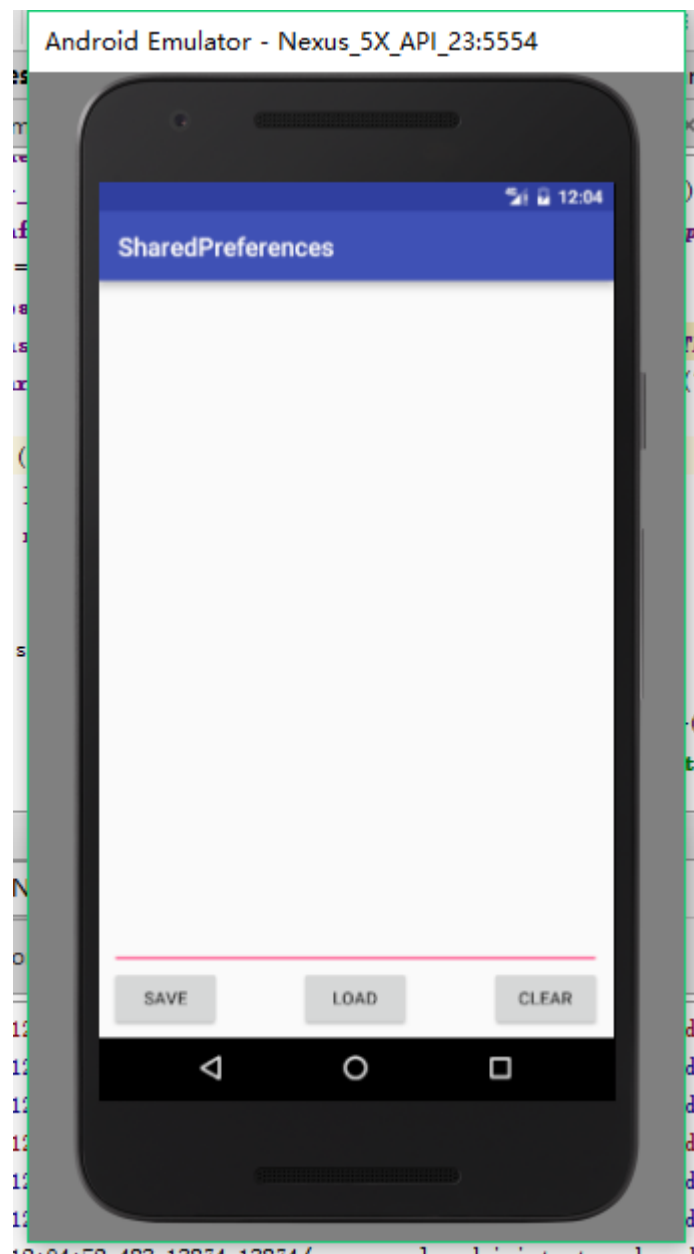


9. 输入错误的密码，提示 Invalid Password

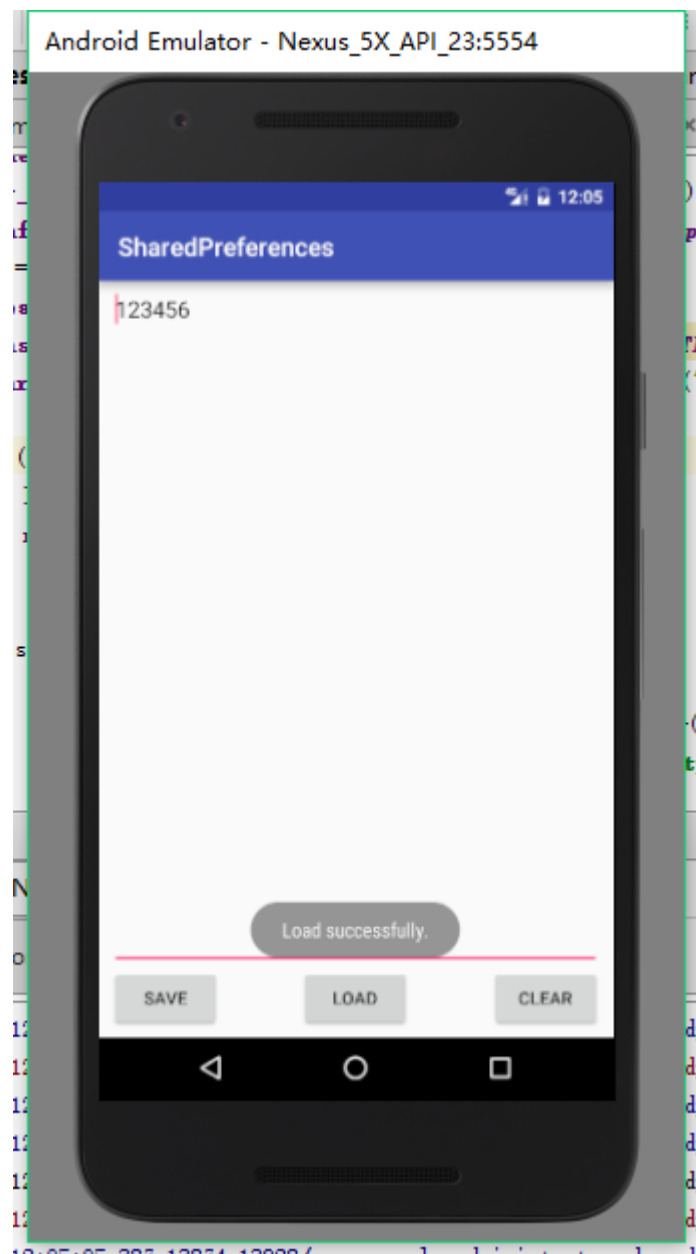


10. 输入正确的密码，成功跳转





11. 点击 LOAD , 加载成功



#### 四、实验心得

将 EditText 的光标定位在左上的方法是将 gravity 设置为 left|top。  
此次实验较为简单，只要懂得如何使用 SharedPreferences 的读写方法和 Android 中对文件的读写方法就可顺利完成此次实验。