

# 中山大学数据科学与计算机学院本科生实验报告

## ( 2017 年秋季学期 )

课程名称：手机应用平台开发

任课教师：刘宁

年级	15 级	专业 ( 方向 )	计应
学号	15331421	姓名	郑佳锋
电话	13719325472	Email	1773643139@qq.com
开始日期	12.1	完成日期	12.3

### 一、 实验题目

Lab9 : File Editor

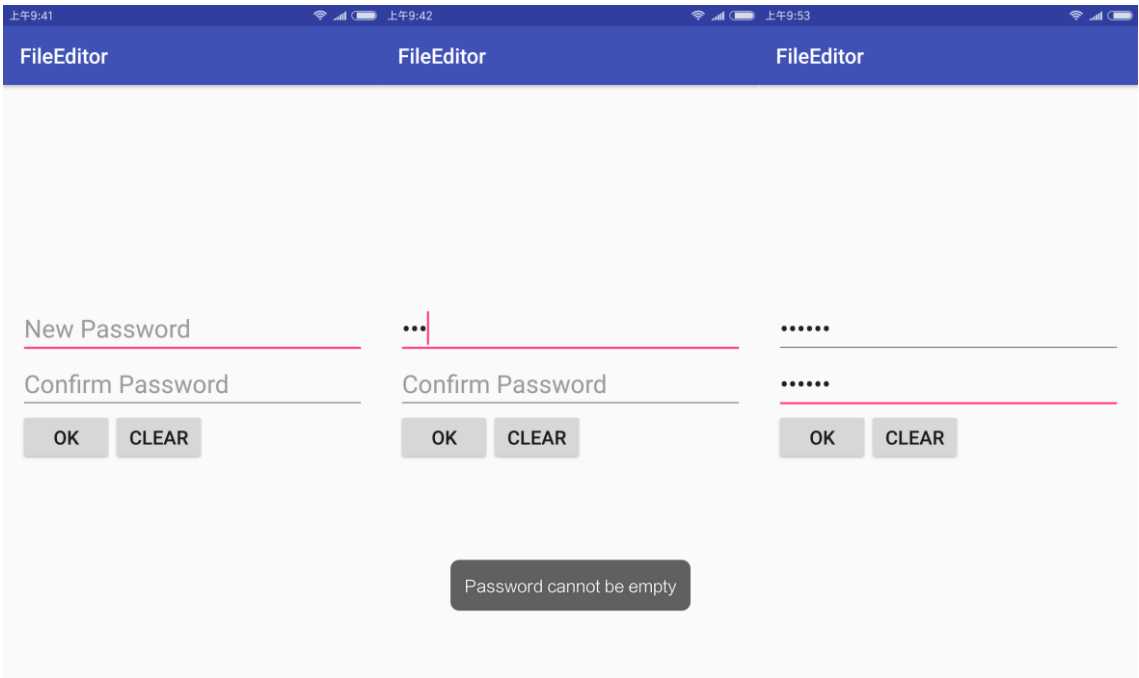
### 二、 实现内容

- 1.使用 SharedPreferences 实现密码的保存。
- 2.实现 android 内存存储文件的增删改查。

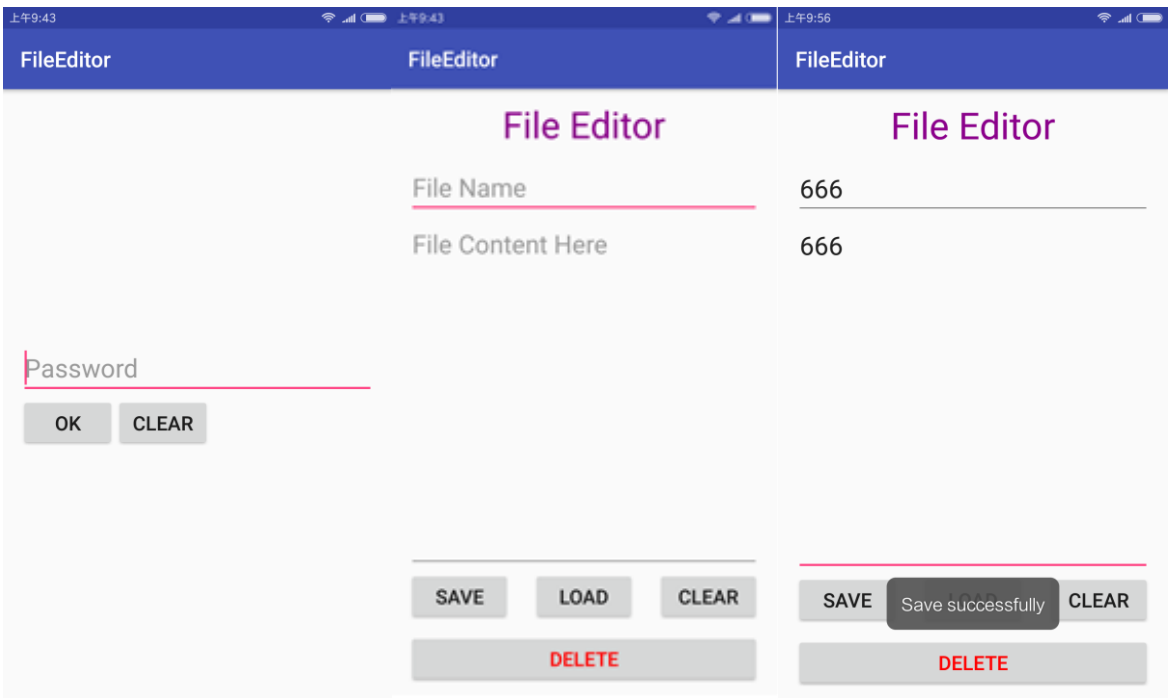
### 三、 实验结果

#### ( 1 ) 实验截图

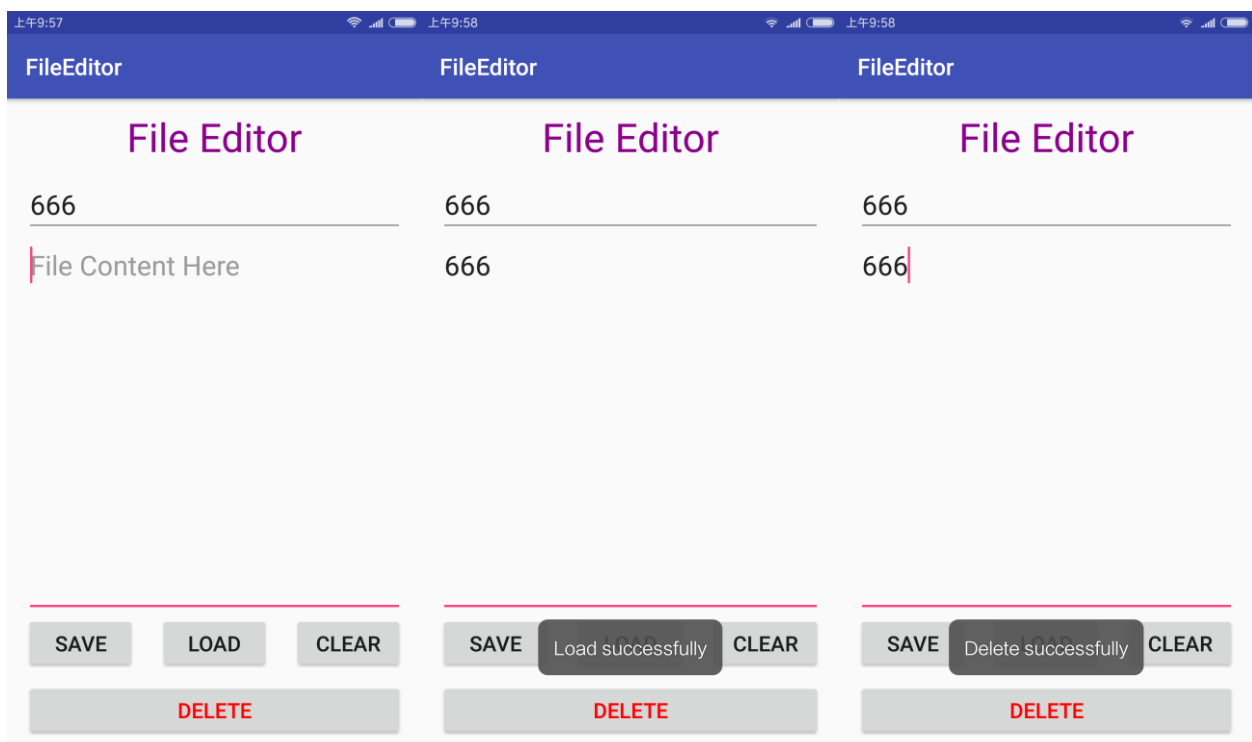
第一次启动应用：                      只输入一次密码：                      输入两次一样的密码后跳转页面：



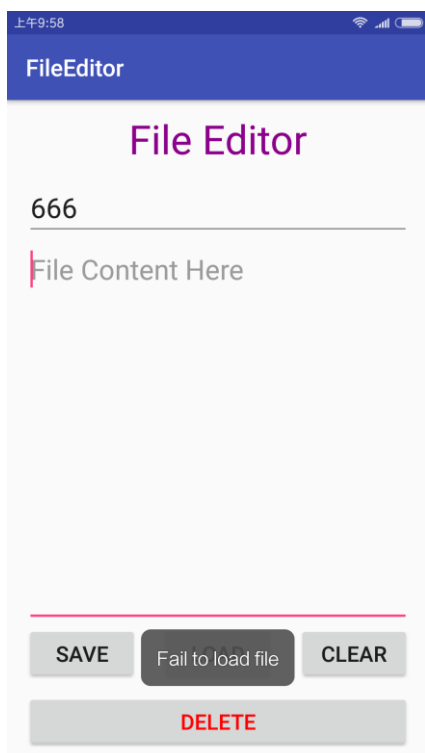
退出应用后再次打开：                      输入相同的密码再次跳转页面    保存文件



点击 clear 按钮                      点击 load 按钮                      点击 delete 按钮



再次点击 load



(2) 实验步骤以及关键代码

## 1.首先是登录界面的设计：

整体使用垂直的 `LinearLayout` 布局，两个按钮水平居中使用嵌套的水平 `LinearLayout` 布局。页面顶部和底部的留白使用 `layout_weight` 来指定。第一个 `EditText` 设置 `layout_weight` 为 1，页面最底部增加一个空白的 `view`，`layout_weight` 也设为 1。这样两个空间就平分剩余的空间。

```
1. <EditText
2.     android:id="@+id/newpwd"
3.     android:layout_width="match_parent"
4.     android:layout_height="wrap_content"
5.     android:hint="@string/new_pwd"
6.     android:inputType="numberPassword"
7.     android:layout_weight="1"
8.     android:gravity="left|bottom"
9. />
10.
11. <View
12.     android:layout_width="match_parent"
13.     android:layout_height="0dp"
14.     android:layout_weight="1"
15. >
16. </View>
```

## 2.接着设计 File Editor 界面：

同样使用线性布局，中间三个按钮使用嵌套的水平的线性布局。设置 `layout_weight` 都为 1 使得它们平分所有的空间。同样，文本内容框设置 `layout_weight` 为 1，使得它占有页面所有的剩余空间。

```
1. <EditText
2.     android:id="@+id/content"
3.     android:layout_width="match_parent"
4.     android:layout_height="wrap_content"
5.     android:hint="@string/file_content"
6.     android:layout_weight="1"
7.     android:gravity="left|top"
8. />
```

## 3.然后实现登录界面的逻辑：

首先我利用一个变量 `valid` 来记录用户是否已经设置过密码，默认为 `false`。所以第一次打开应用时需要输入两次密码。然后用 `SharedPreferences` 保存下来，同时也保存 `valid` 的值为 `true`。这样下次打开应用时可以读取到 `valid` 值为 `true` 时。可以将第二个密码输入框 `setVisibility` 为 `Gone`，表示设置组件不可见并移除它不占空间，两个按钮就可以往上移。接着将用户输入的密码跟保存的密码比对，若匹配，就跳转界面，否则提示密码错误。

页面加载时：

```
1. private static int MODE = MODE_PRIVATE;
2. public static final String PREFERENCE_NAME = "SavePassword";
3. SharedPreferences share;
4. share = getSharedPreferences(PREFERENCE_NAME, MODE);
5. boolean valid = share.getBoolean("valid", false);
6. if(valid == false) {
7.     setListener();
8. }
9. else{
10.     conPwd.setVisibility(View.GONE);
11.     newPwd.setHint("Password");
12.     setListenerForPass();
13. }
```

比对密码：

```
1. newPassword = newPwd.getText().toString();
2. if(newPassword.equals("")){
3.     Toast.makeText(MainActivity.this, "Password cannot be empty", Toast.LENGTH_SHORT).show();
4. }
5. else if(!newPassword.equals(share.getString("password", "default"))){
6.     Toast.makeText(MainActivity.this, "Password Mismatch", Toast.LENGTH_SHORT).show();
7. }
8. else{
9.     Intent intent = new Intent(MainActivity.this, FileEdit.class);
10.    MainActivity.this.startActivity(intent);
11. }
```

#### 4. 最后设置 File Editor 的逻辑。

首先是文件的保存。使用 `FileOutputStream` 写入文件。Close 前使用 `flush` 将缓冲区的内容全部写入。因为文件存取容易出错，所以使用了异常捕捉。

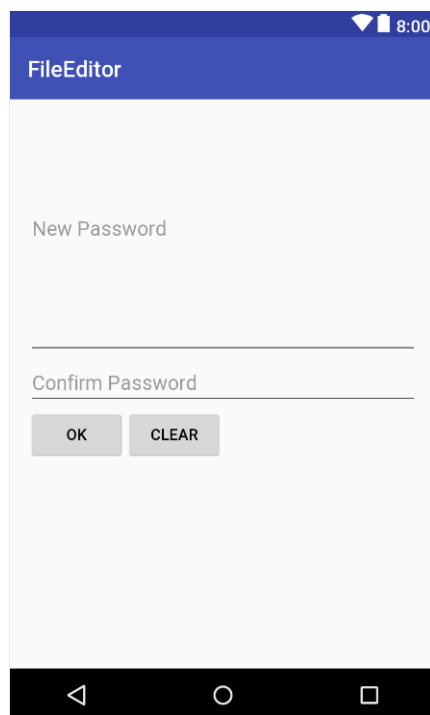
```
1. try (FileOutputStream fileOutputStream = openFileOutput(name_str, MODE_PRIVATE)) {
2.     fileOutputStream.write(content_str.getBytes());
3.     fileOutputStream.flush();
4.     fileOutputStream.close();
5.     Toast.makeText(FileEdit.this, "Save successfully", Toast.LENGTH_SHORT).show();
6. } catch (IOException ex) {
7.     Toast.makeText(FileEdit.this, "Fail to save file", Toast.LENGTH_SHORT).show();
```

```
8. }
```

文件的读取使用 `FileInputStream`，删除使用 `deleteFile`，都是类似的，不再贴代码。

### (3) 实验遇到困难以及解决思路

遇到的问题是处理页面留白的问题。登录界面最开始打算使用两个空白的 `View`，`layout_weight` 为 1 来实现。这样做的问题在于当移除一个第一个密码输入框的时候，第二个输入框会处在页面比较底部的地方，影响美感。所以后来就将删除掉了顶部的 `View`，转而将第一个 `EditText` 的 `layout_weight` 设为 1。这样做就可以解决问题。每次删除掉第二个密码框，下面的两个按钮会自动往上移，布局就比较正常。但是，又有一个问题，就是 `EditText` 占用的空间很大，使得它的提示文字也悬浮在中间。如下图：



后来查了一下 `EditText` 有一个属性叫 `gravity`，将它设置成 `"left|bottom"`（左下角）就完美了问题。

#### 四、 实验思考及感想

本次实验比较简单，主要学习了 android 其中的两种保存信息的方法：SharedPreferences 和 内部文件存储。