项iang﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽﷽因此需要上网去查找，**Lab6 实验报告**

*13331231*

*孙圣*

一、实验目的

1.练习使用sharedPreferences对数据进行存储和读取。

2.练习使用Java的文件读写 。

3.学习使用自动填充TextView。

二、实验内容

实现一个注册和登录界面，并带有记住登录功能，利用Toast提示相应的状态。 当成功登录之后，进入到文件读写界面，可以读取，写入和删除文件。

三、实验步骤

1. 先实现出两个基本UI。

2. MainActivity：

对TA给出的实验要求进行了修改，原来只实现了单个用户的注册和登录，现在将名值对变为（用户名，密码），实现多个用户注册的功能。

注册函数：从sharedPreferences对用户所输入的用户名进行搜索，如果密码等于缺省值（$default$，假设用户不会将其设为密码，这里还是存在一定的漏洞），则认定该用户名还未被注册，将（用户名，密码）对写入sharedPreferences中去，利用editor.putString()方法，在最后还要commit()。

登录函数：同样从sharedPreferences搜索用户名，密码等于缺省值的时，登录失败，因为不存在这个用户。否则，将密码与用户输入的密码进行比对，密码不一样时也提示登录失败。登录成功后跳转到EditFileActivity。

对于记住用户，由于Android默认是将填充的数据记住，因此要反其道而行之，在onRestart()时判断checkbox是否被选中，如果没选中则清空数据域。

3.工具类FileUtils：

一共有三个函数。

SaveContent()利用Java提供的FileOutputStream类来实现保存数据，不过要注意将String转化为byte流。

ReadContent()利用FileInputStream，它所读出来的是字节流，所以要使用的时候需要转化为String。

以上两个方法都需要try和catch，进行IO异常处理。

最后一个方法较为简单，直接调用deleteFile()方法即可。

4.EditFileActivity：

这个类主要是调用刚才实现的FileUtils工具类。

在每个操作时，分别调用对应的函数。对于save和delete这种会改变文件目录的方法，需要重新改变适配器。

android.R.layout.simple\_dropdown\_item\_1line这个参数其实就实现了自动补全的功能，而fileList()获得了所有文件名。

附加项:

主要利用了Java提供的MessageDigest类，利用它我们可以实现很多加密方法，只需要在getInstance()方法中指明即可。在这里我使用的是MD5加密算法。

要得到加密好的信息需要以下两步：

1. 调用update()函数，参数为所要加密内容的字节

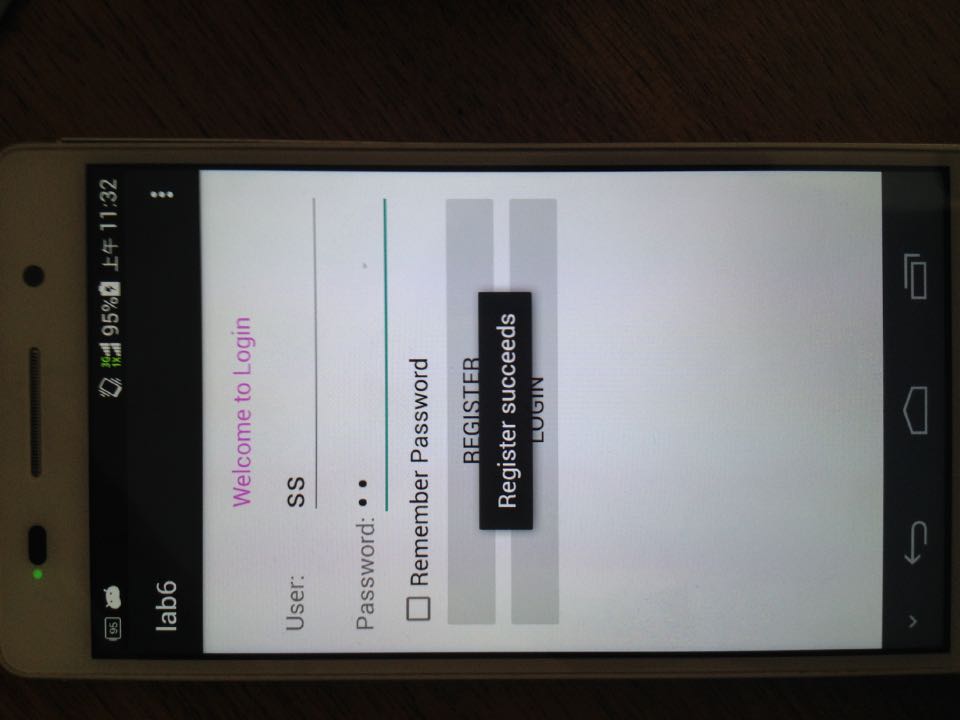
，即用要加密的字符串调用getBytes()方法。

2. 调用digest()函数，得到加密内容的字节数组。本次实验中，我将它又转为了String类型，便于存储在sharedPreferences中。

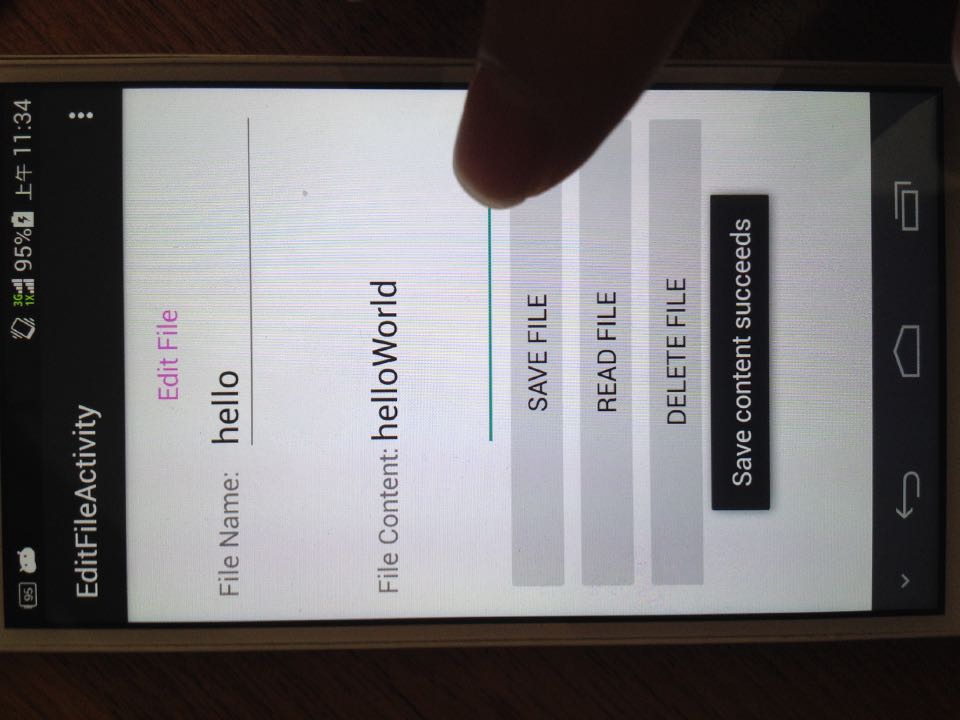
这样，之前明文的（用户名，密码）名值对变成了新的加密过后的（用户名，MD5算法（密码））的名值对。进行比对时，只需要比对MD5的值是否相等，即可知道密码是否正确。

四、实验结果

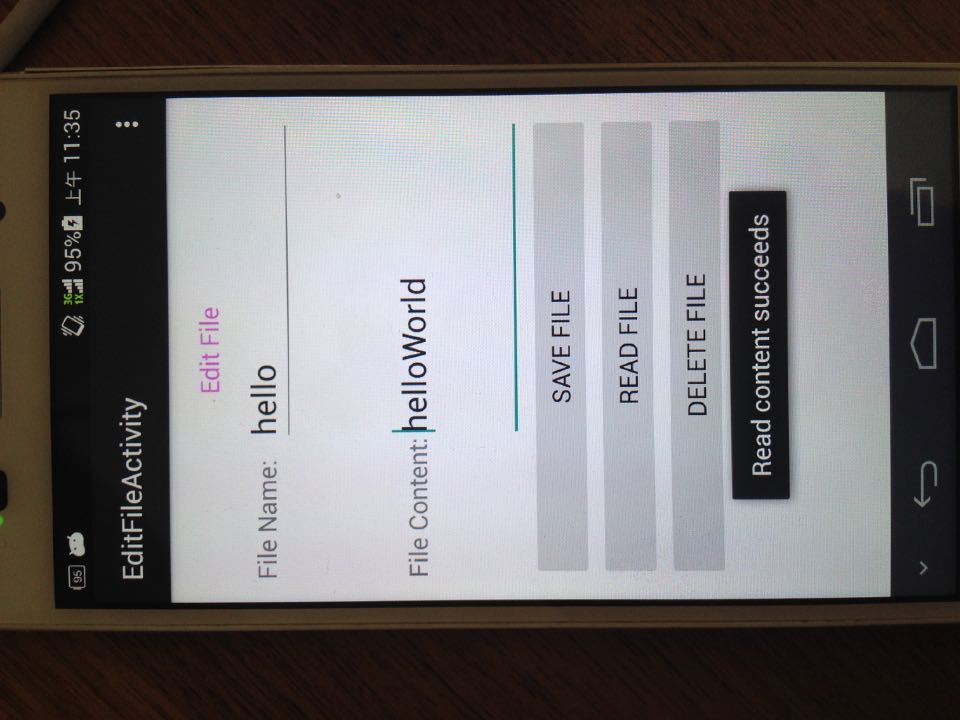
注册成功：



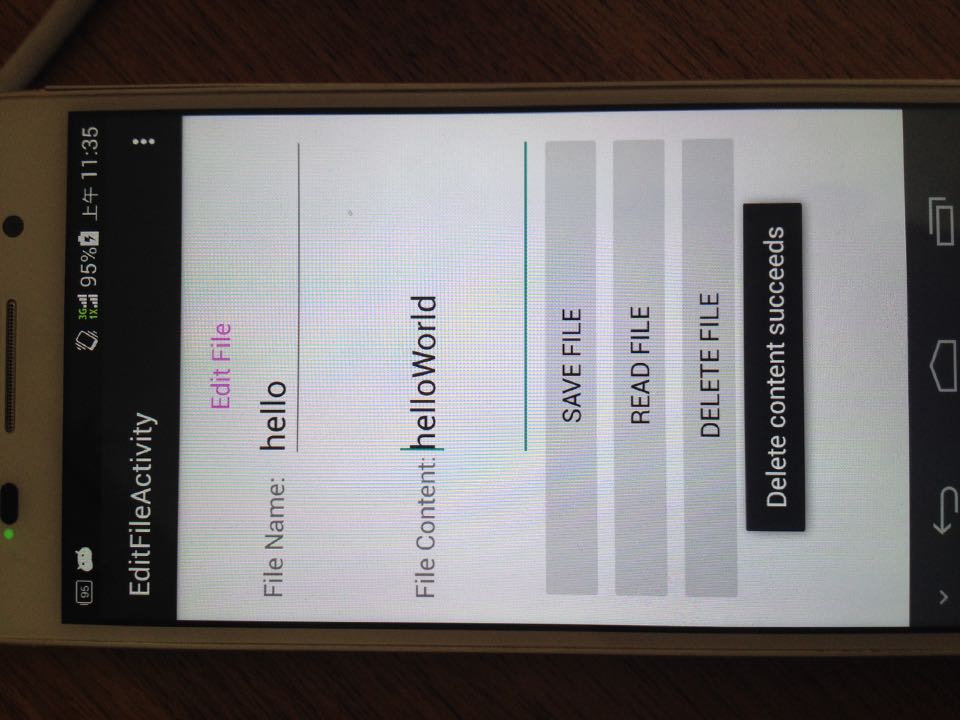
保存：



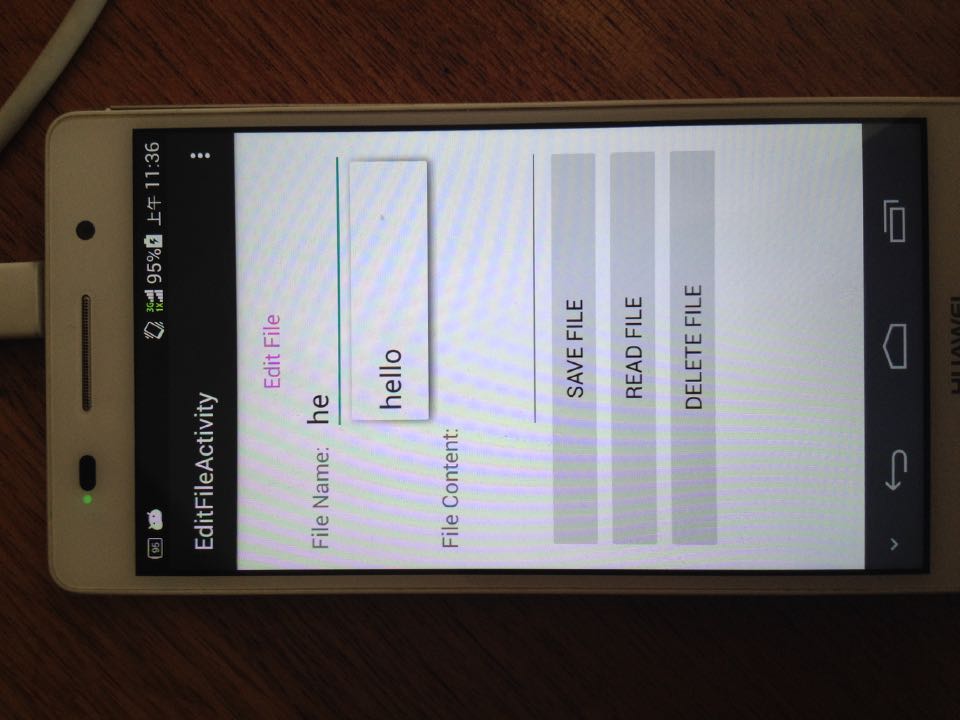
读取：



删除：



自动填充：



五、实验总结

一开始对照着实验要求文档向sharedPreferences中存储数据时发现是用了两个k-v对，一个存储用户名，一个存储密码。后来便对这个要求进行了修改，将key值设定为用户名，而value值设定成了密码，这样就可以允许多用户的注册。但这也带来了一个问题，从sharedPreferences中根据key获得value时默认的返回是一个字符串，这样就无法避免所设置的缺省的字符串与用户所输入的密码相同的情况。

后来，在实现记住用户的功能时，也遇到了一个小问题。原来Android是自动帮我们记住所填充的内容的，所以从其他Activity返回时，原来输入的数据还是被保留，这其实就是我们所需要的记住用户的功能。所以，实际上要做的其实是在没有选中记住用户时，将该数据清空。后来想到老师上课所说的Activity的生命周期，从Activity返回时，会调用restart()方法，因此我重写了那个方法，根据checkbox的值进行相应的操作。

参考资料：

1.Activity的生命周期

<http://developer.android.com/reference/android/app/Activity.html>

2. MessageDigest

<http://docs.oracle.com/javase/6/docs/api/java/security/MessageDigest.html>

<http://examples.javacodegeeks.com/core-java/security/java-security-messagedigest-example/>