

实验六：数据存储（一）

13331388 庄梓嘉

实验目的

- （1）学会使用 Android SharedPreferences 保存和读取数据
- （2）学会使用 Android 文件操作保存和读取数据
- （3）学会使用 AutoCompletedTextView 实现历史记录提示

实验内容

- （1）登录界面：

使用 SharedPreferences 记录登录状态，点击 Register 按钮，能够将 User 和 Password 写入 SharedPreferences，写入后使用 Toast 提示写入成功。

- （2）注册账号成功后，输入账号和密码，点击登陆按钮，若账号与密码正确，则跳转到文件操作界面，否则使用 Toast 提示登录错误

- （3）文件操作界面返回登录界面时，如果 Remember Password CheckBox 没有勾上，则 User 和 Password 控件不保存相应的信息，否则将 SharedPreferences 保存的信息自动填上相应的控件。使用 AutoCompletedTextView 实现文件自动提示功能，如果文件存在，则该匹配的文件自动在 AutoCompletedTextView 下生成。

- （5）在 File Content 下输入文件的内容，点击 Save File 后能够自动保存文件，并且文件名自动提示已经更新；点击 Read File 按钮能够自动读取文件保存的数据，并将内容显示在 File Content 控件下面。

- （6）点击 Delete File 控件能够删除文件，并且文件名自动提示已经更新，重新点击 Read File，发现已经不能够读取文件内容了。

(7) 检查是否成功生成文件 (用模拟器)

保存文件路径 : /data/data/你的程序包名/files/*.txt

SharedPreferences 文件路径 : /data/data/你的程序包名/shred_prefs/*.xml

【拓展项】

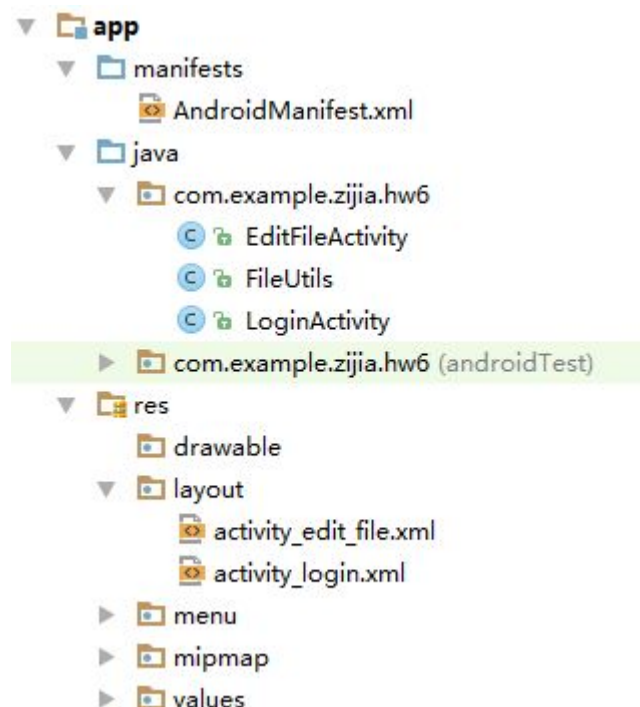
Android SharedPreferences 能够将 K-V 保存在本地文件之中 ,但是直接将密码已明文的形式保存在文件中是非常不安全的 , 很容易就会被别人盗取。

针对这种情况 , 请提出一个改进方案 , 并实现。

报告中请说明清楚思路 , 并附加核心代码辅助说明。

实验步骤

实验文件目录 :



(1) 设计 Edit File 界面 , 效果看实验结果。

其中 File Name 的 EditText 需要用 AutoCompleteTextView 代替。

```
<AutoCompleteTextView
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_width="0dp"
    android:hint=""
    android:id="@+id/fileName"/>
```

(2) 创建 FileUtils 类文件，在里面添加存储文件，读取文件，删除文件函数。

//存储文件

```
public void saveContent(Context context, String fileName, String fileText){
    try {
        FileOutputStream fos = context.openFileOutput(fileName, Context.MODE_PRIVATE);
        fos.write(fileText.getBytes());
        fos.close();
        Toast.makeText(context, "Save Content Success", Toast.LENGTH_SHORT).show();
    } catch (IOException e) {
        Toast.makeText(context, "Save Content Failed", Toast.LENGTH_SHORT).show();
        e.printStackTrace();
    }
}
```

//读取文件

```
public String getContent(Context context, String fileName) {
    try {
        FileInputStream fis = context.openFileInput(fileName);
        byte[] contents = new byte[fis.available()];
        fis.read(contents);
        fis.close();
        Toast.makeText(context, "Read Content Success", Toast.LENGTH_SHORT).show();
        return new String(contents);
    } catch (IOException e) {
        Toast.makeText(context, "Read Content Failed", Toast.LENGTH_SHORT).show();
        e.printStackTrace();
        return new String("");
    }
}
```

//删除文件

```
public void deleteFile(Context context, String fileName){
    context.deleteFile(fileName);
    Toast.makeText(context, "Delete File Success", Toast.LENGTH_SHORT).show();
}
```

(3) 在 EditFileActivity 里将三个函数分别对应到相应按钮的监听事件。

```

//存储文件
saveFile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        fileUtils.saveContent(EditFileActivity.this, fileName.getText().toString(),
            fileText.getText().toString());
        fileName.setAdapter(new ArrayAdapter<String>(EditFileActivity.this,
            R.layout.support_simple_spinner_dropdown_item, EditFileActivity.this.fileList()));
    }
});

//读取文件
readFile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        fileText.setText(fileUtils.getContent(EditFileActivity.this, fileName.getText().toString()));
    }
});

//删除文件
deleteFile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        fileUtils.deleteFile(EditFileActivity.this, fileName.getText().toString());
        fileName.setAdapter(new ArrayAdapter<String>(EditFileActivity.this,
            R.layout.support_simple_spinner_dropdown_item, EditFileActivity.this.fileList()));
        fileName.setText("");
        fileText.setText("");
    }
});

```

(4) 设计 Login 界面，效果看实验结果，编辑注册和登录按钮监听事件。

```

//注册事件
register.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SharedPreferences.Editor editor = getSharedPreferences("data", MODE_PRIVATE).edit();
        editor.putString("user", user.getText().toString());
        editor.putString("password", password.getText().toString());
        editor.commit();
        Toast.makeText(LoginActivity.this, "Register success", Toast.LENGTH_SHORT).show();
    }
});

//登录事件
login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SharedPreferences sharedPreferences = getSharedPreferences("data", MODE_PRIVATE);
        String pre_user = sharedPreferences.getString("user", "");
        String pre_password = sharedPreferences.getString("password", "");
        if (pre_user.equals(user.getText().toString()) && pre_password.equals(password.getText().toString())) {
            startActivity(new Intent(LoginActivity.this, EditFileActivity.class));
        } else {
            Toast.makeText(LoginActivity.this, "Login Error", Toast.LENGTH_SHORT).show();
        }
    }
});

```

(5) 添加判断 CheckBox 及对应事件，若 isChecked 是真就保存到 data 里，若是假，则清空密码栏。

```
//判断CheckBox是否选上及对应事件
if (getSharedPreferences("data", MODE_PRIVATE).getBoolean("isChecked", false)) {
    user.setText(getSharedPreferences("data", MODE_PRIVATE).getString("user", ""));
    password.setText(getSharedPreferences("data", MODE_PRIVATE).getString("password", ""));
    checkBox.setChecked(true);
}

//记录密码就保存密码数据，不记录则清空密码栏。
checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            SharedPreferences.Editor editor = getSharedPreferences("data", MODE_PRIVATE).edit();
            editor.putBoolean("isChecked", isChecked);
            editor.commit();
        } else {
            password.setText("");
        }
    }
});
```

【拓展项】

(1) 首先在 LoginActivity 里添加加密解密函数。

```
//简单的密码加密解密算法 异或运算
private static final String key0 = "FECOI()*&<MNCXZPKL";
private static final Charset charset = Charset.forName("UTF-8");
private static byte[] keyBytes = key0.getBytes(charset);
public static String encode(String enc){
    byte[] b = enc.getBytes(charset);
    for(int i=0,size=b.length;i<size;i++){
        for(byte keyBytes0:keyBytes){
            b[i] = (byte) (b[i]^keyBytes0);
        }
    }
    return new String(b);
}
public static String decode(String dec){
    byte[] e = dec.getBytes(charset);
    byte[] dee = e;
    for(int i=0,size=e.length;i<size;i++){
        for(byte keyBytes0:keyBytes){
            e[i] = (byte) (dee[i]^keyBytes0);
        }
    }
    return new String(e);
}
```

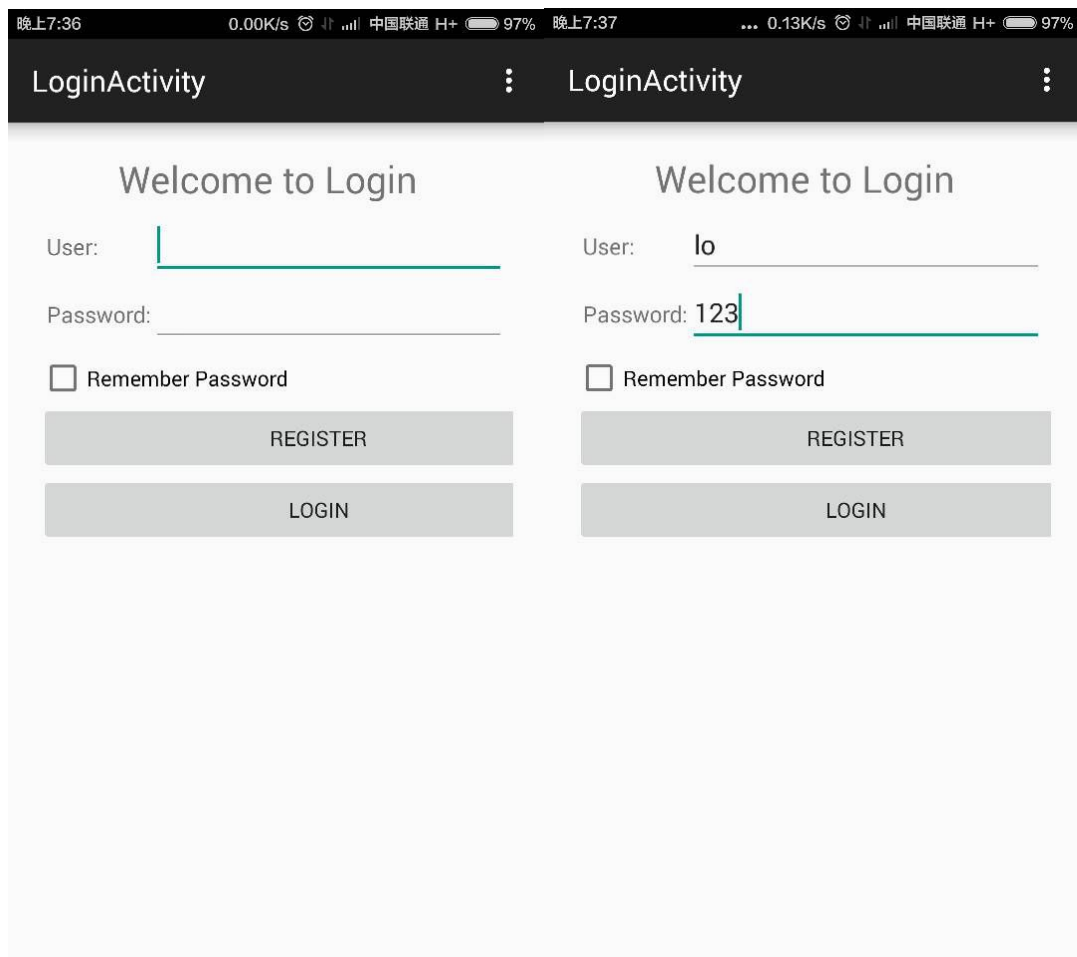
(2) 在注册按钮监听事件里添加密码加密。

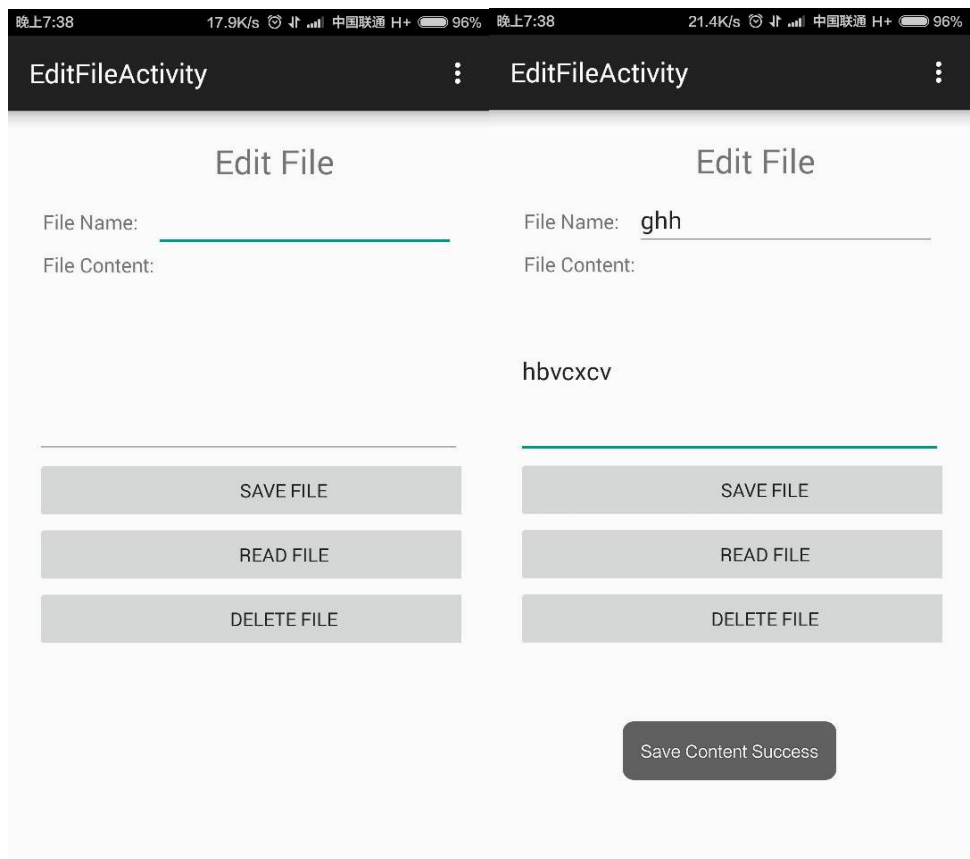
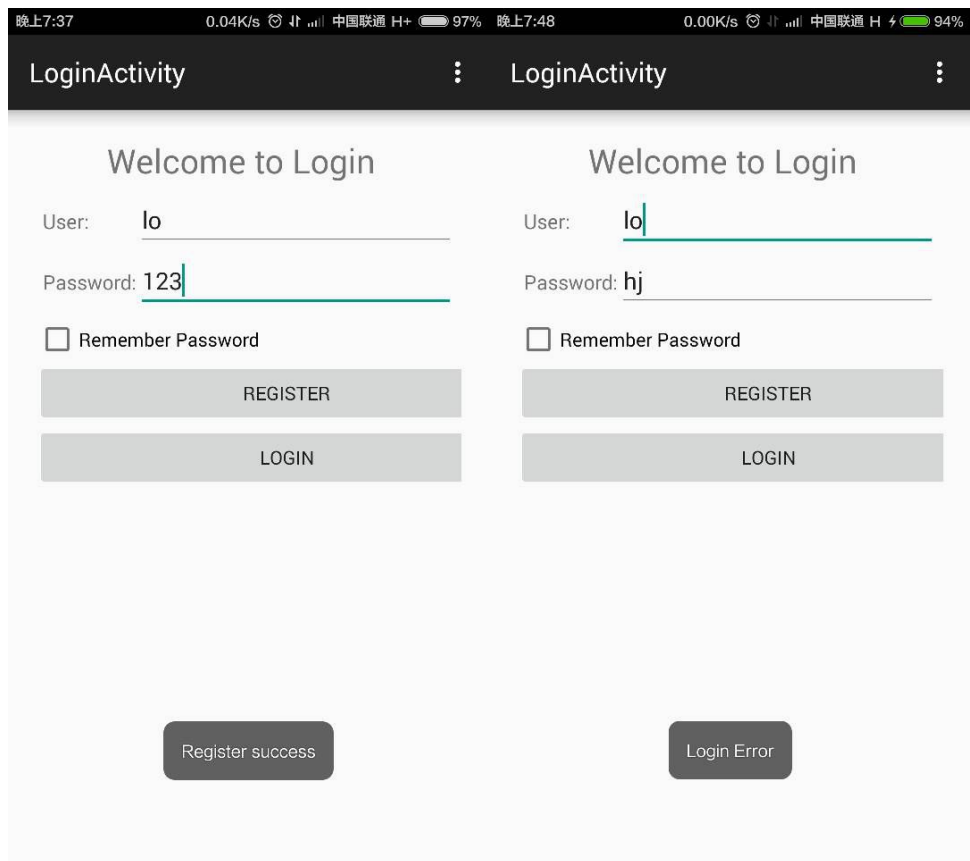
```
//密码加密
//enpassword = encode(password.getText().toString());
//editor.putString("password", enpassword);
```

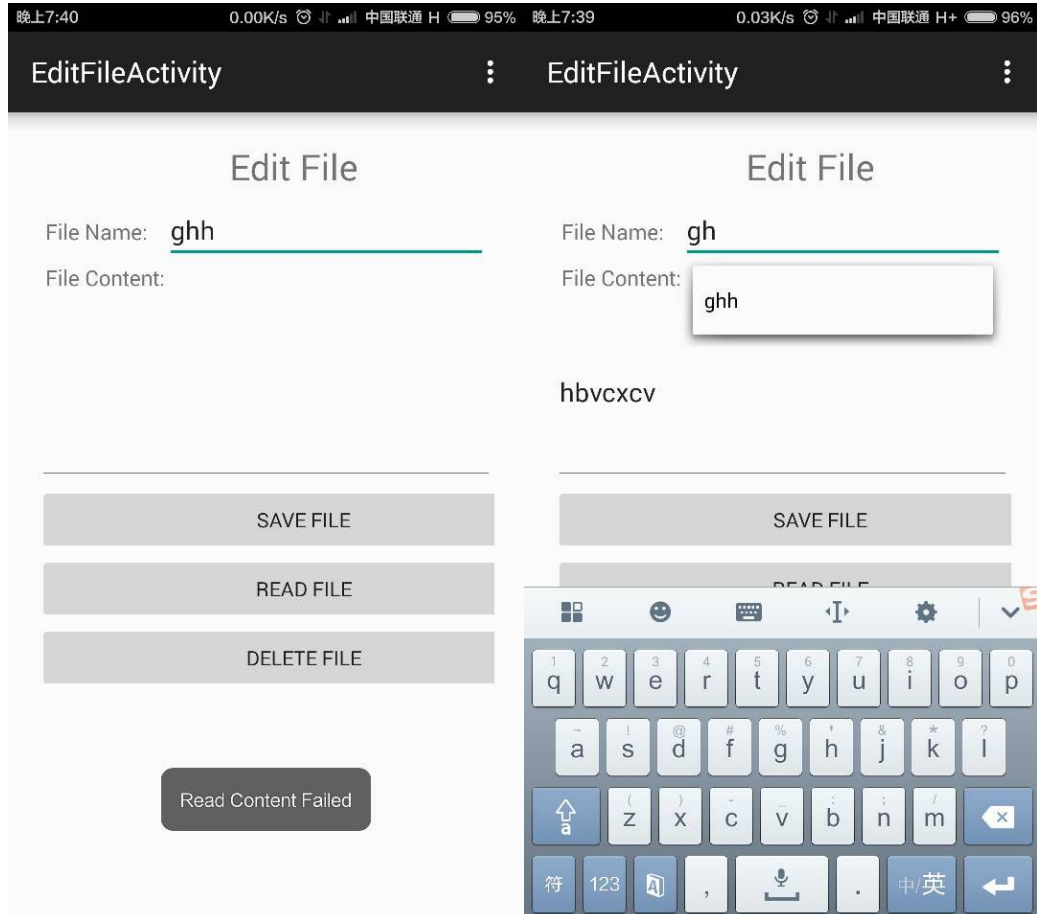
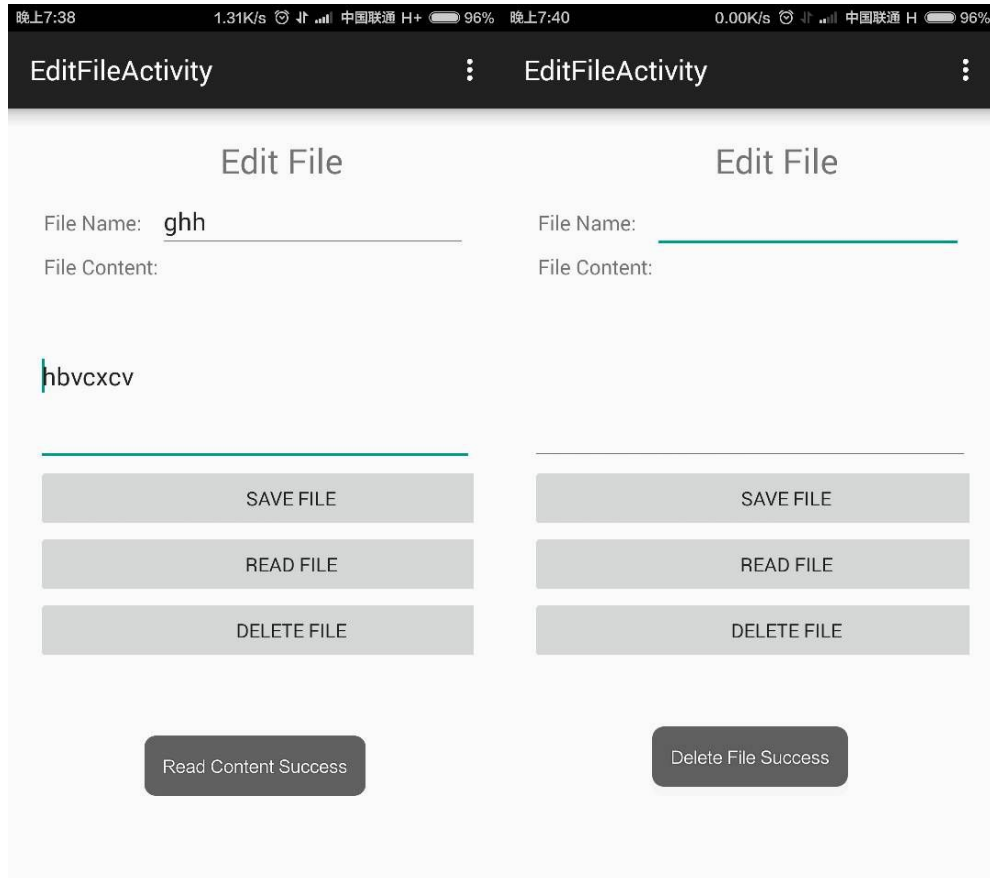
(3) 在登录按钮监听事件里添加密码解密。

```
//密码解密
//depassword = sharedPreferences.getString("password", "");
//String pre_password = decode(depassword);
```

实验结果







可以找到存储用户的数据文件和保存文件的数据。

| | | | | |
|---|-----------------------|-----|------------|-------|
| ▼ | com.example.zijia.hw6 | | 2015-11-28 | 05:02 |
| > | cache | | 2015-11-28 | 04:21 |
| ▼ | files | | 2015-11-28 | 04:57 |
| | lo | 3 | 2015-11-28 | 04:38 |
| | lib | | 2015-11-28 | 05:02 |
| ▼ | shared_prefs | | 2015-11-28 | 05:02 |
| | data.xml | 194 | 2015-11-28 | 05:02 |

参考资料

Java 简单的加密解密算法

<http://szitjiang.iteye.com/blog/1662997>

Android 入门：File 文件存储

<http://blog.csdn.net/xiazdong/article/details/7687439>

Android 之 Adapter 用法总结

<http://blog.csdn.net/fznpcy/article/details/8658155/>

实验总结

这次实验要求简单使用数据存储方面知识。根据实验文档进行，难度不大，需要关注一些地方，如自动补全的功能，需要把 EditText 换成 AutoCompletedTextView，以及存储文件和删除文件的时候都需要更新一下 adapter。至于拓展项的内容，密码加密之后再存储起来有很多种方法，挑选了比较简单的异或运算，存之前加密，取之后解密，这样存储在文件里的就是加密后的密码，比明文密码要好一点。要查看到加密后的密码需要打开保存用户数据的文件，本来在虚拟机上进行的，后来好像 Genymotion 要收费没法用了，只能在真机上进行，真机上找不到那个数据文件，只好作罢。当然，有更好的加密算法，如 MD5 之类，不过之前只写过 C++ 的 MD5 加密算法，java 的还没有时间钻研，所以就没使用了。