# 中山大学移动信息工程学院本科生实验报告

## （2017 年秋季学期）

课程名称：移动应用开发　　任课教师：

| | | | |
|---|---|---|---|
| 年级 | 15352272 | 专业（方向） | 软件工程（移动信息工程） |
| 学号 | 15352272 | 姓名 | 彭国栋 |
| 电话 | 15626066058 | Email | 578291308@qq.com |
| 开始日期 | | 完成日期 | |

## 一、 实验题目

a、 学习 SharedPreferences 的基本使用；

b、 学习 Android 中常见的文件操作方法；

c、 复习 Android 界面编程。

## 二、 实现内容

1、首先，需要实现一个密码输入 activity：

a、 如果应用首次启动，则界面呈现出两个输入框，分别为新密码输入和确认密码输入框；

b、 输入框下方有两个按钮：

- OK 按钮，点击之后：

- 若 new password 为空，则弹出密码为空的提示；

- 若 new password 与 comfirm password 不匹配，则弹出不匹配的提示；

- 若密码不为空且互相匹配，则保存密码，进入文件编辑界面。

- CLEAR 按钮，点击之后清除所有输入框的内容。

c、 完成创建密码后，退出应用再进入应用，则只呈现一个密码输入框；

- 点击 OK 按钮后，如果输入的密码与保存的密码不匹配，则弹出 Toast 提示；

- 点击 CLEAR 按钮后，清除密码输入框的内容

2、然后，实现一个文件编辑 activity：

a、 界面底部有两行四个按钮， 第一行三个按钮高度一致，顶对齐，按钮水平均匀分布。 按钮上方除了 ActionBar 和 StatusBar 之外的空间由标题和两个 EditText 占据， 文件内容编辑的 EditText 需要占据除去其他控件的全部屏幕空间，且内部文字竖直方向置顶，左对齐；

b、 在文件名输入框内输入文件名， 在文件内容编辑区域输入任意内容，点击 SAVE 按钮后能够保存到指定文件，成功保存后弹出 Toast 提示；

c、 点击 CLEAR 按钮，能够清空文件内容编辑区域内的内容；

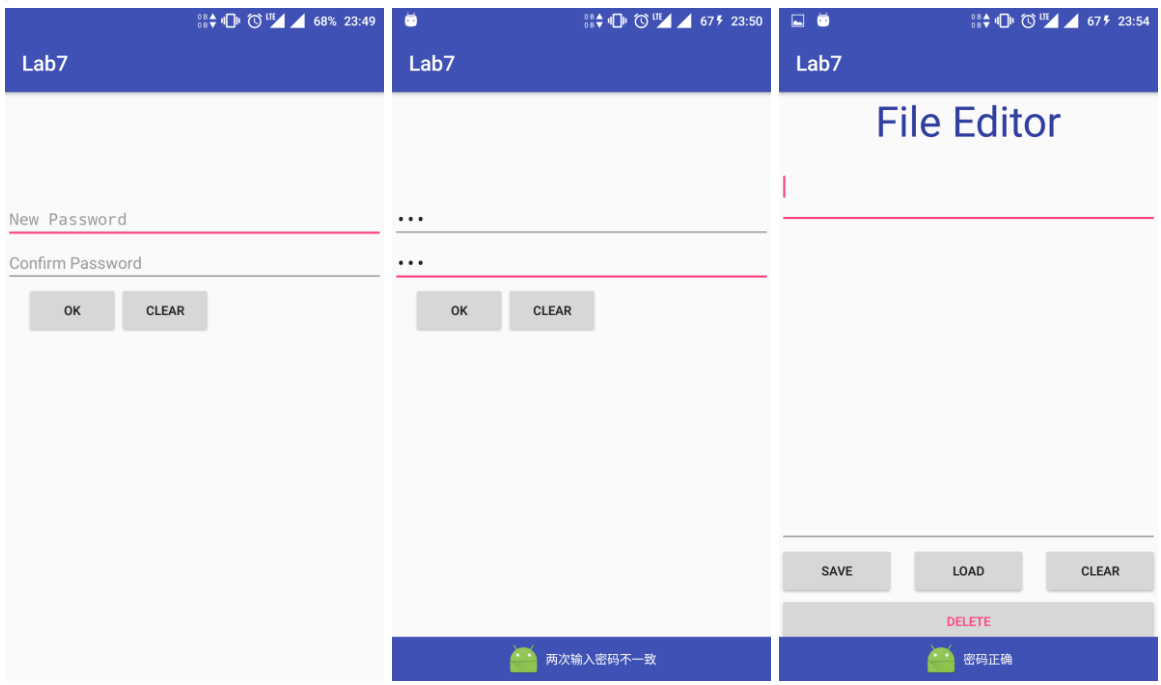
d、 点击 LOAD 按钮，能够按照文件名从内存中读取文件内容，并将文件内容写入到编辑框中。如果成功导入，则弹出成功的 Toast 提示， 如果导入失败（例如：文件不存在），则弹出读取失败的 Toast 提示。

e、 点击 DELETE 按钮，能够按照文件名从内容中删除文件，删除文件后再载入文件，弹出导入失败的 Toast 提示。

4、 特殊要求：进入文件编辑的 Activity 之后，如果点击返回按钮，则直接返回 Home 界面，不再返回密码输入界面。

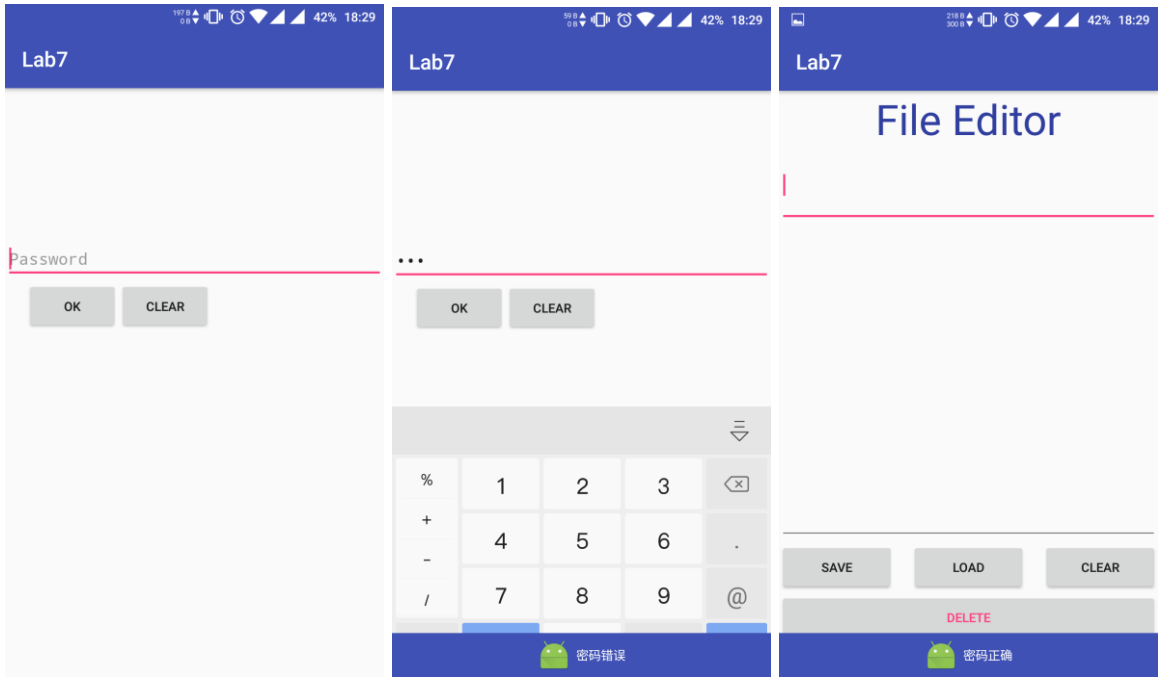
## 三、 课堂实验结果

（1） 实验截图

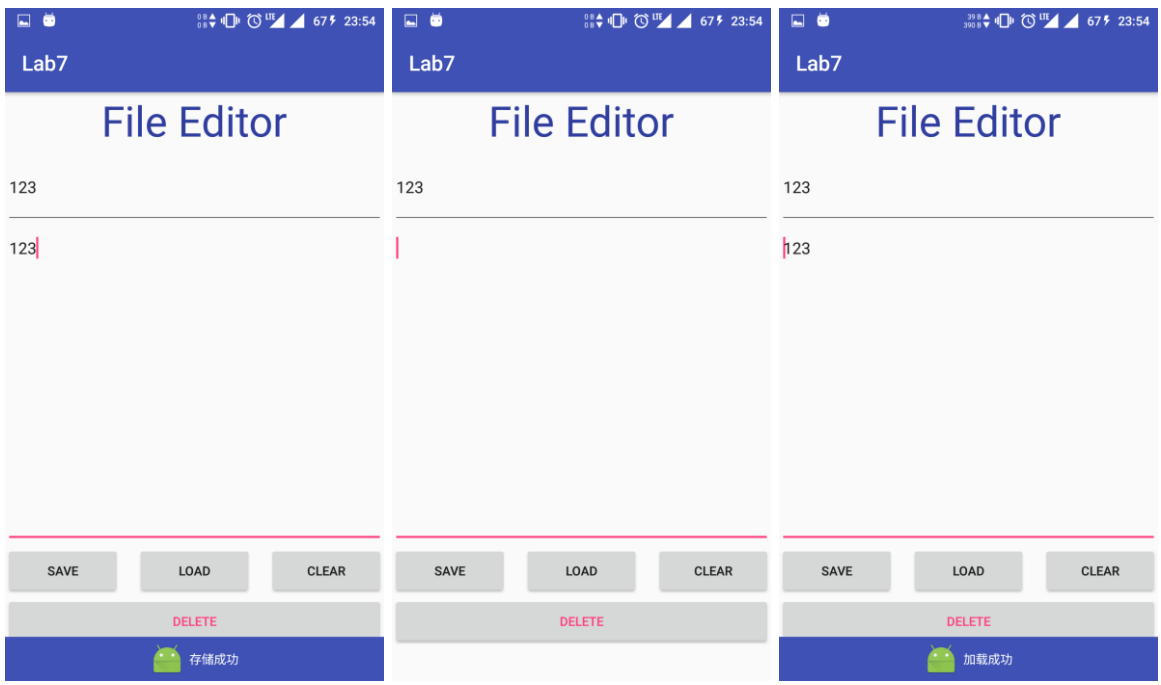


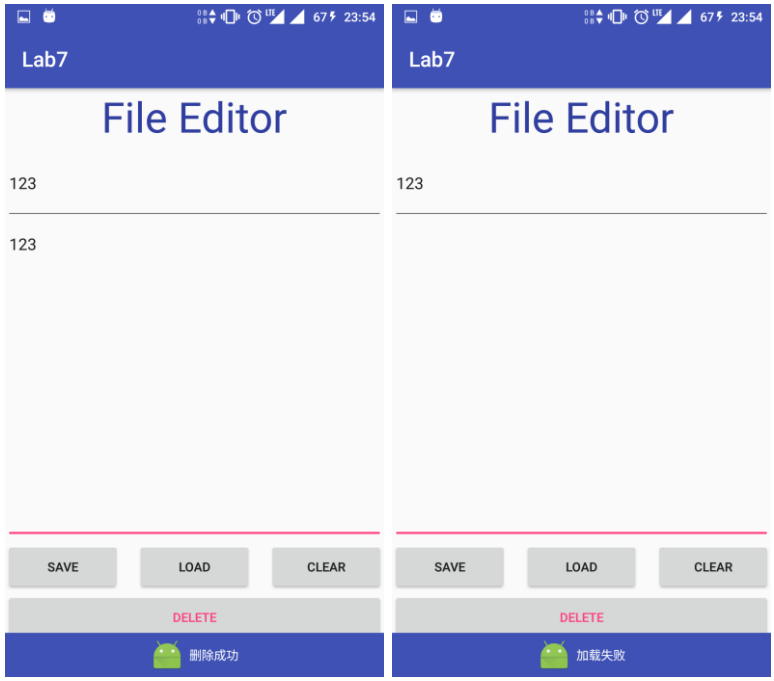

① 首次使用该软件，需要输入新密码和确认密码，当两者相同且不为空，

即可登录成功。




② 再次使用该软件，只需要输入上次所存储的密码即可。






③ 分别执行存储，清空，加载，删除，清空，加载操作。

（2） 实验步骤以及关键代码

从界面上看，主要分为两个界面，分别为输入密码和文件操作界面。

①、MainActivity:

这里定义了三个 EditText 部件，通过判断是否存在密码来显示不同的部件，当不存在密码的时候，则显示 2 个 EditText 供用户输入，当存在密码的时候，只显示一个 EditText 来供用户输入。

为了存储密码和是否存在密码，采用将 flag 和 curpasswrod 保存在一个 sharedpreference 里。

```
static Boolean flag;//判断是否存在密码
static String curpassword;//判断当前密码是什么
SharedPreferences sharedPref;
SharedPreferences.Editor editor;
final static String PREFERENCE_NAME="SAVE_PASSWORD";
```

初始化的时候：通过读取文件里面的值来给 flag 和 curpassword 赋值。

```
private void init()
{

    new_password=(EditText)findViewById(R.id.new_password);
    confirm_password=(EditText)findViewById(R.id.confirm_password);
    password=(EditText)findViewById(R.id.password);
    ok_button=(Button)findViewById(R.id.ok_button);
    clear_button=(Button)findViewById(R.id.clear_button);
    sharedPref=getSharedPreferences(PREFERENCE_NAME,MODE_PRIVATE);
    editor=sharedPref.edit();
    flag=sharedPref.getBoolean("flag",false);//每次读取flag
    curpassword=sharedPref.getString("current_password",null);//每次读取密码
}
```

当密码不存在的时候，则显示两个 EditText，当输入的密码匹配并且长度不为 0 的时候，证明密码已经正确，就要设置 flag 和 curpassword 的值，并存储在 SAVE_PASSWORD 里。

```
if(flag==false)
{
    password.setVisibility(EditText.INVISIBLE);
    new_password.setVisibility(EditText.VISIBLE);
    confirm_password.setVisibility(EditText.VISIBLE);
}
```

```
if(flag==false)//密码不存在
{
    String np=new_password.getText().toString();
    String cp=confirm_password.getText().toString();
    if(np.equals(cp)&&np.length()!=0)
    {
        editor.putBoolean("flag",true);//存储密码标志
        editor.putString("current_password",cp);//存储密码
        editor.commit();

        Toast.makeText(getApplicationContext(), "密码正确", Toast.LENGTH_SHORT).show();

        Intent intent=new Intent(MainActivity.this,FileEdit.class);
        startActivity(intent);
    }
```

当密码存在的时候，则只显示一个 EditText，并且当密码正确的时候则跳转。

```
else
{
    password.setVisibility(EditText.VISIBLE);
    new_password.setVisibility(EditText.INVISIBLE);
    confirm_password.setVisibility(EditText.INVISIBLE);
}
```

```
String pass=password.getText().toString();
if(pass.equals(curpassword))
{
    Toast.makeText(getApplicationContext(), "密码正确", Toast.LENGTH_SHORT).show();

    Intent intent=new Intent(MainActivity.this,FileEdit.class);
    startActivity(intent);
}
```

②、FileEdit

·setSave：存储文件，建立一个叫 filename 的文件，并写入内容。

```
try
{
    FileOutputStream fileOutputStream=getApplicationContext().openFileOutput(filename,MODE_PRIVATE);
    fileOutputStream.write(filecontent.getBytes());
    fileOutputStream.flush();
    fileOutputStream.close();
    Toast.makeText(getApplicationContext(),"存储成功",Toast.LENGTH_SHORT).show();
} catch (FileNotFoundException e) {
    e.printStackTrace();
}catch (IOException e)
{
    e.printStackTrace();
}
```

·setLoad：加载文件，打开文件名为 filename 的文件，读取里面的内容并写入 file_content 的部件里面。

```
load_button.setOnClickListener((v) → {
        String filename=file_name.getText().toString();
        String filecontent;
        try
        {
            FileInputStream fileInputStream=openFileInput(filename);
            //打开filename
            byte[] readBytes=new byte[fileInputStream.available()];
            fileInputStream.read(readBytes);
            //读取
            filecontent=new String (readBytes);
            //设置内容
            file_content.setText(filecontent);
            Toast.makeText(getApplicationContext(),"加载成功",Toast.LENGTH_SHORT).show();
```

·setClear：清空内容，通过直接设置 file_content 的部件的内容为” ”。

```
private void setClear()
{
    clear_button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            file_content.setText("");
        }
    });
}
```

·setDelete：通过查询 deleteFile 的定义，发现其返回一个 Boolean 的值，那么删除成功的时候就返回 true，所以直接使用 deleteFile 即可。

```
private void setDelete()
{
    delete_button.setOnClickListener((v) → {
            String filename=file_name.getText().toString();
            if(deleteFile(filename))
            {
                Toast.makeText(getApplicationContext(),"删除成功",Toast.LENGTH_SHORT).show();

            }
            else
            {
                Toast.makeText(getApplicationContext(),"未找到该文件",Toast.LENGTH_SHORT).show();
            }

    });
}
```

·按返回键回到 home。

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK) {


    //跳转到主界面
        Intent intent = new Intent();
        intent.setAction("android.intent.action.MAIN");
        intent.addCategory("android.intent.category.HOME");
        startActivity(intent);
        finish();
    }
    onDestroy();
    return super.onKeyDown(keyCode, event);
}
```

（3） 实验遇到困难以及解决思路

在实验过程中，对于如何在第二次打开应用的时候保存密码。一开始以为直接使用 static 的变量 flag 即可。但是发现当 activity 被销毁后，并不能保存为上一次的值。后面通过借助 SharedPreferences 来实现，将输入的密码保存在一个文件里，每次进入的时候先读取文件内的信息，从而实现了在第二次以后进入能够保存密码的操作。

## 四、 实验思考及感想

这次实验总的来说比起上次的更为简单，而实际实现过程中也没遇到特别多的困难，但还是学到了通过将数据保存在本地再访问的思想。感觉只要能将程序的每个功能去细化，再逐一实现，更有利于可读性和可修改性。这次实验还学到了可以使用/** */的方式进行注释，而 AS 也能够自动帮助我们进行这样的注释，这种注释的好处是如果使用/** */注释的话，你再调用类和方

法的时候会出现提示，内容就是你写的注释。就好像文档帮助一样。