中山大学数据科学与计算机学院软件工程（移动信息工程）本科生实验报告

移动应用开发实验报告

姓名	学号	班级	电话	邮箱	日期
蔡烨	15352010	1501班	15989010669	caiy29@mail2.sysu.edu.cn	2017/12/11

1. 实验题目

数据存储（一）

2. 实现内容

使用SharedPreferences进行密码存储，实现密码识别；

使用internal storage进行文件存储，实现增、删、改、加载功能。

3. 实验过程

1. 写好两个xml文件，file_edit.xml中，通过设置weightSum和layout_weight来保证编辑框的大小，同时使用android:gravity="top"使得编辑框的内容从左上角开始显示。
2. 写MainActivity.java文件，主要对ok按钮和clear按钮进行监听，以及重写onResume函数。
    1. onResume函数首先判断sharedpreferences中是否已经有密码，对getString("password")的结果进行判断，如果不为空，意味着不是第一次打开这个应用，已经设置过密码，因此将第一个输入框的visibility设置为invisible，第二个输入框的提示设为password。

```
@Override
protected void onResume(){
    super.onResume();
    SharedPreferences sharedPreferences = getSharedPreferences("password",MODE_PRIVATE);
    if(!sharedPreferences.getString("password","").equals("")){
        psw.setVisibility(View.INVISIBLE);
        confirm.setHint("Password");
```

    2. ok按钮被点击时，如果还没设置过密码，需要对两个输入框都进行判断，只有两个输入框的内容都一样时，才调用sharedpreferences的editor中，用putString("password","<第一个输入框的内容>")写进密码，然后跳转到编辑文件的页面。

```
public void onClick(View v) {
    SharedPreferences sharedPreferences = getSharedPreferences("password",MODE_PRIVATE);
    if(sharedPreferences.getString("password","").equals("")){// 还没注册过
        if(psw.getText().toString().equals(""))
            Toast.makeText(MainActivity.this,"password cannot be empty", Toast.LENGTH_SHORT).show();
        else if(confirm.getText().toString().equals(""))
            Toast.makeText(MainActivity.this,"please confirm your password", Toast.LENGTH_SHORT).show();
        else if(!confirm.getText().toString().equals(psw.getText().toString()))
            Toast.makeText(MainActivity.this,"passwords do not match", Toast.LENGTH_SHORT).show();
        else if(confirm.getText().toString().equals(psw.getText().toString())){
            SharedPreferences.Editor editor =  sharedPreferences.edit();
            editor.putString("password",psw.getText().toString());
            editor.commit();
            Intent intent = new Intent(MainActivity.this,FileEdit.class);
            startActivity(intent); //页面跳转
```

    如果sharedpreferences中已经存在密码，则取出，判断与第二个输入框中的内容是否一致，如果一致，跳转到编辑文件页面。

```
} else { //注册过了
    if(!confirm.getText().toString().equals(sharedPreferences.getString("password","")))
        Toast.makeText(MainActivity.this,"the password is false", Toast.LENGTH_SHORT).show();
    else{
        Intent intent = new Intent(MainActivity.this,FileEdit.class);
        startActivity(intent); //页面跳转
```

3. clear按钮被点击时，用setText()将两个输入框的内容都设置为空即可。

3. 写FileEdit.java文件，对四个按钮进行监听。

    1. save按钮被点击时，获取第一个输入框的内容作为文件名fileName，用openFileOutput方法打开文件，然后用write方法将第二个编辑框的内容写进去。

```
//保存按钮
save.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String fileName = name.getText().toString();
        try(FileOutputStream fileOutputStream = openFileOutput(fileName,MODE_PRIVATE)){
            String str = content.getText().toString();
            fileOutputStream.write(str.getBytes());
            Toast.makeText(FileEdit.this,"Save successfully", Toast.LENGTH_SHORT).show();
        }catch (IOException e){
            Toast.makeText(FileEdit.this,"Fail to save", Toast.LENGTH_SHORT).show();
```

    2. load按钮被点击时，获取fileName，用openFileInput方法打开文件，如果不存在该文件，会抛出异常，在catch的时候toast。如果存在，用read()方法读取文件内容，然后用setText将内容写进第二个编辑框。

```
//加载按钮
load.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String fileName = name.getText().toString();
        try(FileInputStream fileInputStream = openFileInput(fileName)){
            byte[] contents = new byte[fileInputStream.available()];
            fileInputStream.read(contents);
            content.setText(new String(contents));
            Toast.makeText(FileEdit.this,"Load successfully", Toast.LENGTH_SHORT).show();
        }catch(IOException e){
            Toast.makeText(FileEdit.this,"Fail to load file", Toast.LENGTH_SHORT).show();
        }
```

    3. clear按钮被点击时，将第二个编辑框setText("")，然后toast。

```
//清除按钮
clear.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        content.setText("");
        Toast.makeText(FileEdit.this,"Clear successfully", Toast.LENGTH_SHORT).show();
    }
```

    4. delete按钮被点击时，用deleteFile(<fileName>)方法将文件删除，然后将第二个编辑框的内容置空，再toast。

```
//删除按钮
delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String fileName = name.getText().toString();
        boolean b = deleteFile(fileName);
        content.setText("");
        if(b)
            Toast.makeText(FileEdit.this,"Delete successfully", Toast.LENGTH_SHORT).show();
        else
            Toast.makeText(FileEdit.this,"Failed to delete", Toast.LENGTH_SHORT).show();
    }
```

4. 为了使得按下返回键时不再回到输入密码的界面，以及注册FileEdit类，在注册文件中修改如下：

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

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".FileEdit"/>
```

5. internal storage和external storage的区别

1. internal storage：

   内部存储适用于，当不想被用户或者其他app访问到文件。

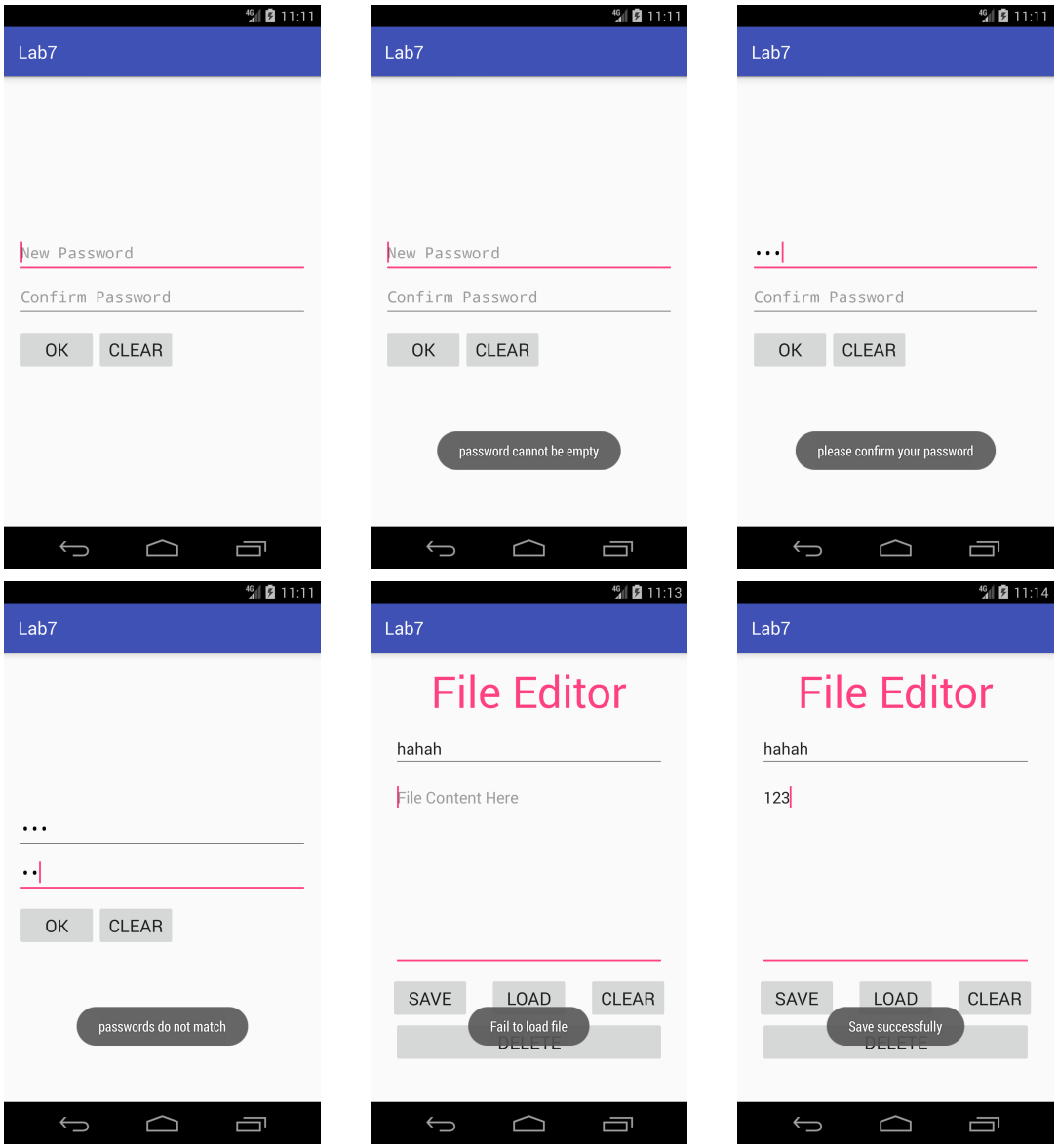
   内部存储的文件只能被当前app访问到，当用户卸载app时，这些文件也会被删掉。

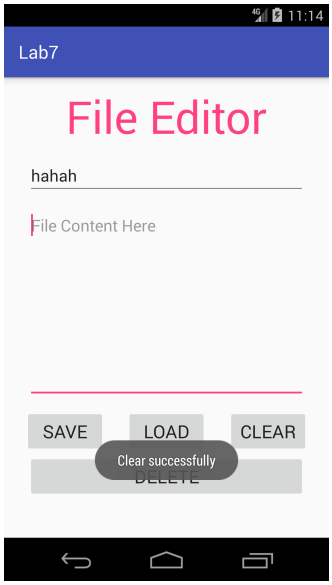
2. external storage：

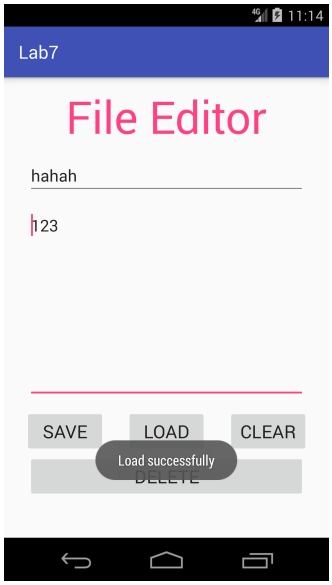
   外部存储适用于，不需要严格的访问权限且希望文件能够被其他app所共享或者用户可以通过电脑访问到。

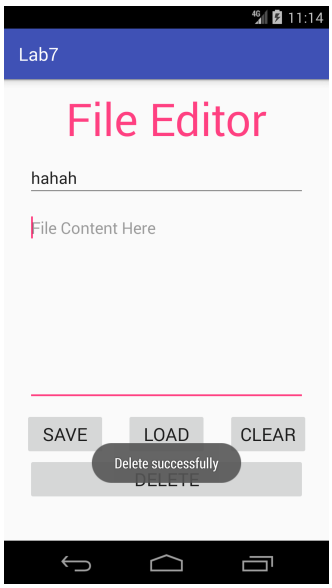
   外部存储的文件可以被其他程序应用或者用户访问到，并且不一定总是可用的，比如，如果文件被放在了外部sdcard上面，当sdcard被移走时，文件就无法访问到了。而当卸载app时，external根目录（getExternalFileDir()）下的文件会被删除，但是getExternalStorageDirectory()下的文件则不会被删除，即使应用被卸载了，这些文件还是依然存在并且可用的。

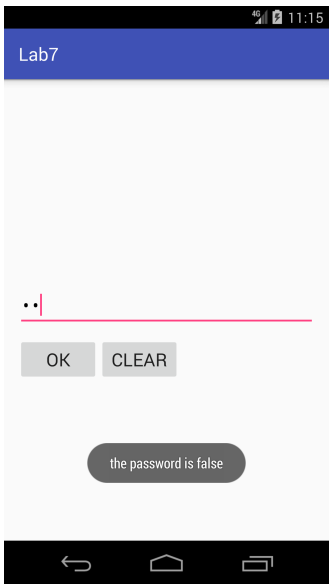
4. 实验结果截图











5. 实验思考及感想

这一次实验整体上来说比较顺利，主要遇到的问题有，一开始 用来判断是否已经设置过密码时，是用布尔类型的flag，虽然是定义为全局变量，在onCreate函数中写入sharedpreferences时已经改变了flag的值，但是返回再打开时，在onResume函数判断得到的flag并不是改变之后的值，导致无法按照需求来设置页面。猜想是因为返回的时候已经把这个activity结束掉了，因此再打开时flag又被初始化，导致即使已经存过密码，但页面并不会改变。后来的做法就是不用flag，而直接获取sharedpreferences里面的内容，根据是否为空来判断是否设置过密码。