

# 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：手机应用开发

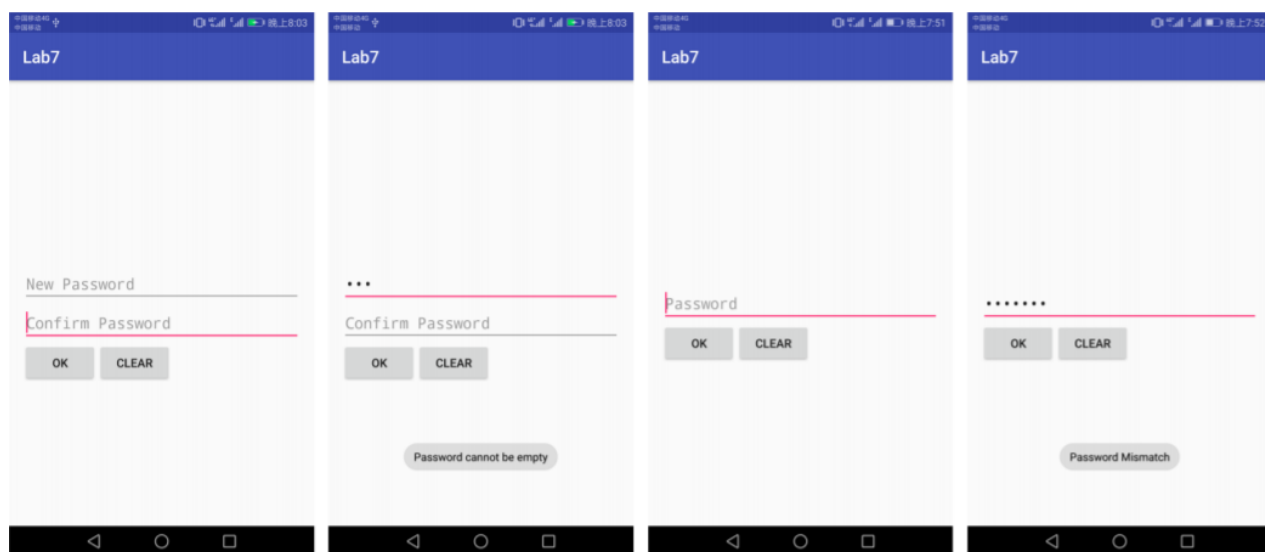
任课教师：刘宁

年级	2015	专业 ( 方向 )	计应
学号	14970011	姓名	宋思亭
电话	18826073511	Email	1056284551@qq.com
开始日期	2017.11.27	完成日期	2017.11.28

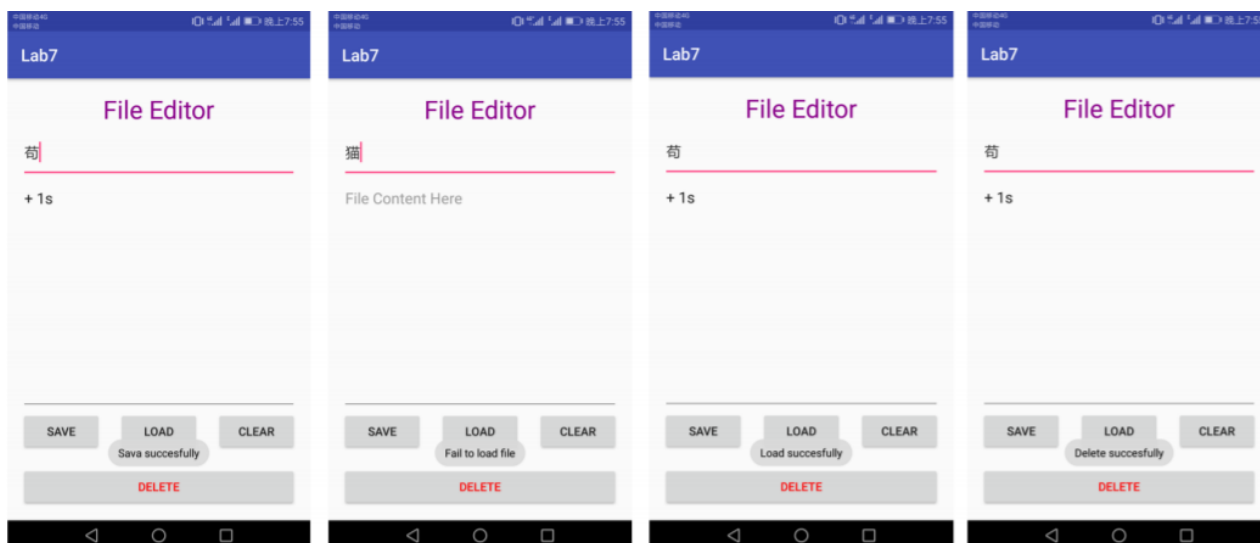
### 一、 实验目的

- 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



从左至右，依次为：初始密码界面、密码为空提示、密码匹配后重新进入界面、密码错误提示。



从左至右，依次为：保存成功提示、写入失败提示、写入成功提示、删除成功提示。

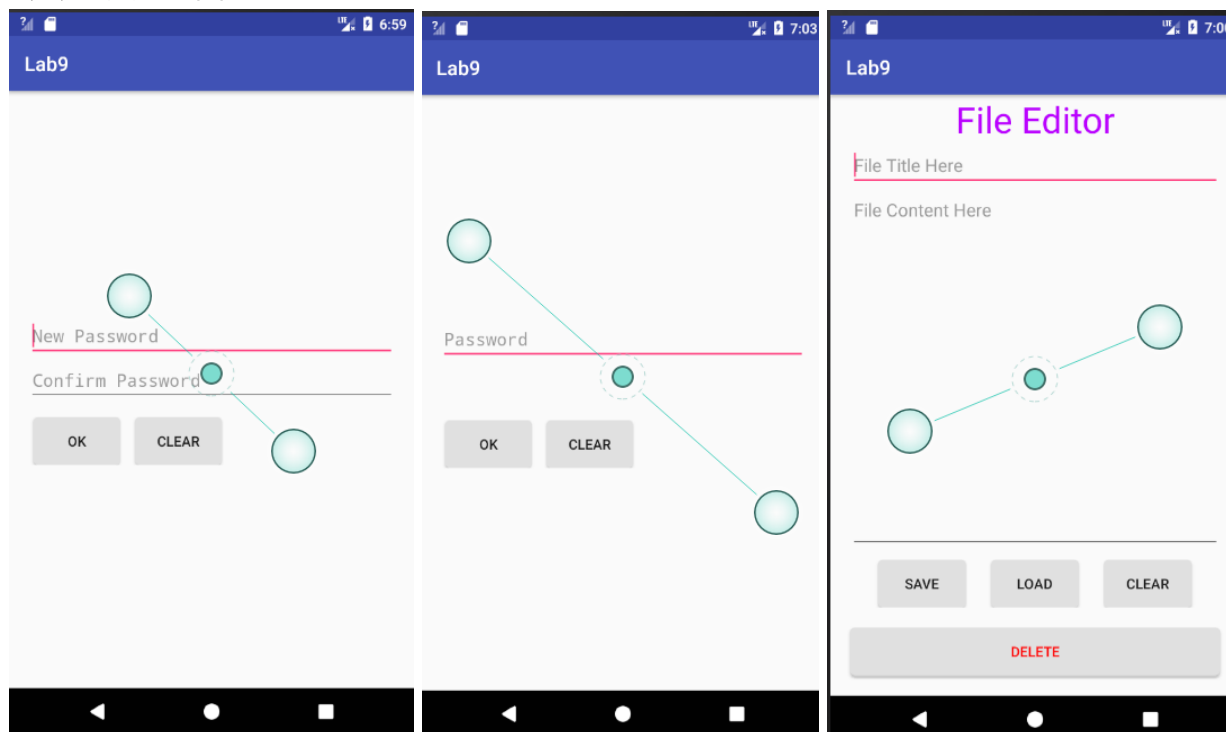
1. 如图所示，本次实验需要实现两个 activity；
2. 首先，需要实现一个密码输入 activity：
  - a) 如果应用首次启动，则界面呈现出两个输入框，分别为新密码输入和确认密码输入框；
  - b) 输入框下方有两个按钮：
    - OK 按钮，点击之后：
      - 若 new password 为空，则弹出密码为空的提示；
      - 若 new password 与 comfirm password 不匹配，则弹出不匹配的提示；
      - 若密码不为空且互相匹配，则保存密码，进入文件编辑界面。
    - CLEAR 按钮，点击之后清除所有输入框的内容。
  - 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。
  - d) 出于学习的目的，我们使用 SharedPreferences 来保存密码，但是在实际应用中我们会用更加安全的机制来保存这些隐私信息；
3. 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。

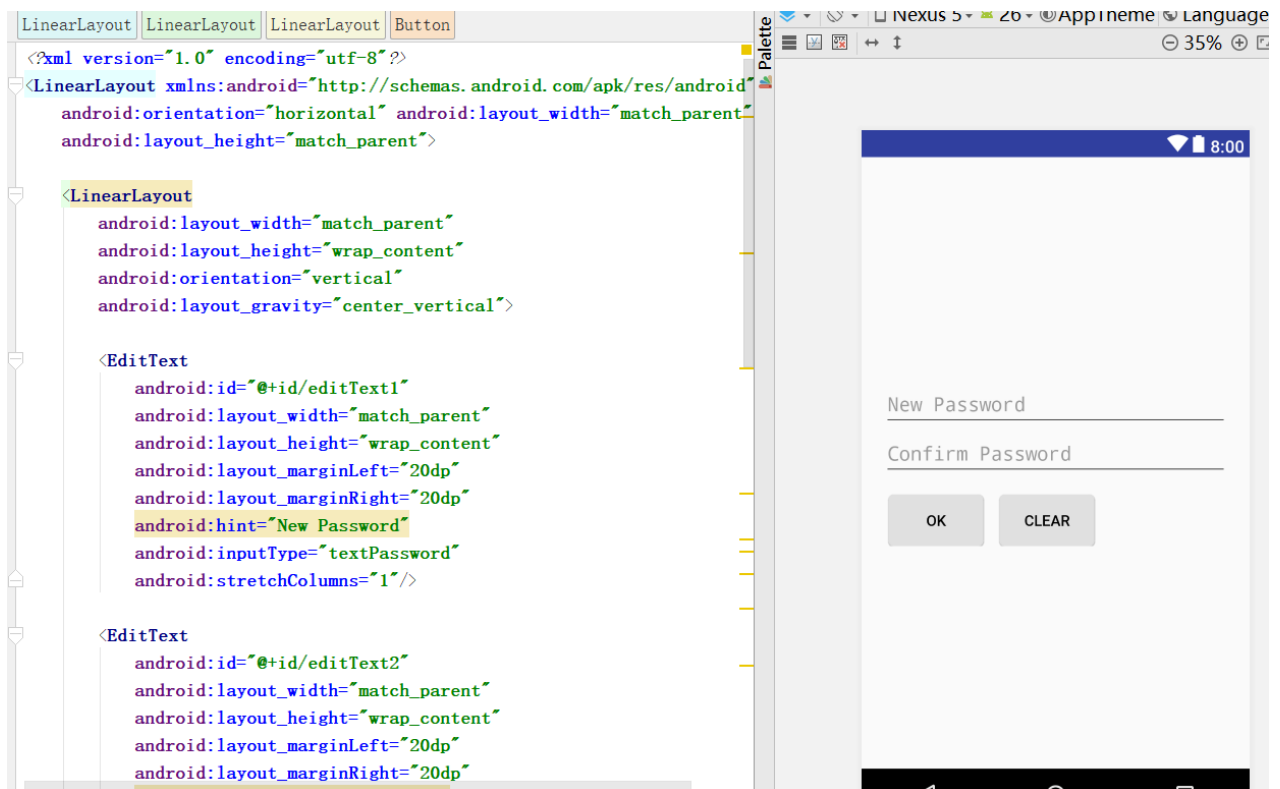
### 三、 课堂实验结果

#### (1) 实验截图

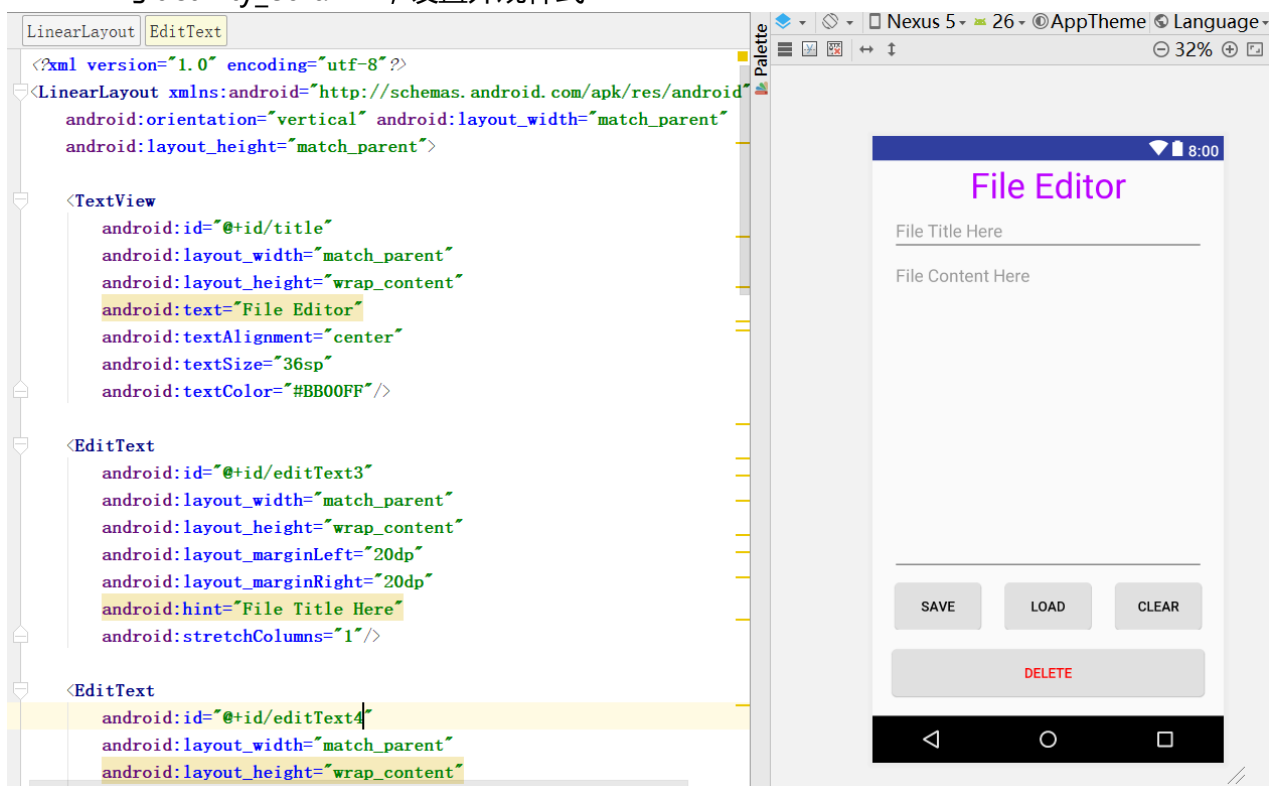


#### (2) 实验步骤以及关键代码

1. 写 activity\_main.xml，设置外观样式



写 activity\_edit.xml，设置外观样式



写 shape.xml，设置按钮样式

```

<solid android:color="#E0E0E0" />
<corners android:radius="5dip" />
<padding
    android:left="10dp"
    android:right="10dp"
    android:bottom="0dp"
    android:top="0dp"/>

```

## 2. 写 MainActivity 类

onCreate 函数中先读取本地密码，判断是否是第一次打开应用

```

//读取保存在本地密码
public void readAccount() {
    //创建SharedPreferences对象
    SharedPreferences sp = getSharedPreferences( name: "info", MODE_PRIVATE);

    //获得保存在SharedPreferences中的密码和确认密码
    String password1 = sp.getString( key: "password1", defValue: "");
    String password2 = sp.getString( key: "password2", defValue: "");

    //在输入框中显示密码和确认密码
    et_password.setText(password1);
    et_password2.setText(password2);

    if(!"".equals(password2)) {
        et_password2.setVisibility(View.INVISIBLE);
        et_password.setHint("Password");
        et_password.setText("");
        tag = false;
    }
}

```

使用 SharedPreferences 保存提交的密码

```

ok.setOnClickListener((v) -> {
    //获得用户输入的用户名和密码
    String password1 = et_password.getText().toString();
    String password2 = et_password2.getText().toString();

    //创建sharedPreference对象, info表示文件名, MODE_PRIVATE表示访问权限为私有的
    SharedPreferences sp = getSharedPreferences( name: "info", MODE_PRIVATE);

    //获得sp的编辑器
    SharedPreferences.Editor ed = sp.edit();

    //以键值对的显示将用户名和密码保存到sp中
    ed.putString("password1", password1);
    ed.putString("password2", password2);

    //提交用户名和密码
    ed.commit();

    // 提交以及失败判定逻辑
    if ("".equals(password1) || "".equals(password2)) {
        Toast.makeText( context: MainActivity.this, text: "Password cannot be empty.", Toast.LENGTH_LONG).show();
    } else if (!password1.equals(password2)) {
        Toast.makeText( context: MainActivity.this, text: "Password Mismatch", Toast.LENGTH_LONG).show();
    } else if (password1.equals(password2)) {
        Intent intent = new Intent();
        intent.setClass( packageContext: MainActivity.this, FileEdit.class);
        MainActivity.this.startActivity(intent);
    }
}

```

### 设置 CLEAR 按钮点击监听

```

clear.setOnClickListener((v) -> {
    if (tag == true) {
        et_password.setText("");
        et_password2.setText("");
    } else {
        et_password.setText("");
    }
});

```

## 3. 写 FileEdit 类

### 监听 SAVE

```

save.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String title = editTitle.getText().toString();
        try (FileOutputStream fileOutputStream = openFileOutput( name: title + ".txt", MODE_PRIVATE)) {
            String str = "";
            str = editContent.getText().toString();
            fileOutputStream.write(str.getBytes());
            Toast.makeText( context: FileEdit.this, text: "Save successfully", Toast.LENGTH_LONG).show();
            return ;
        } catch (IOException ex) {
            Log.e( tag: "TAG", ex.toString());
        }
    }
});

```

## 监听 LOAD

```
load.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String title = editTitle.getText().toString();
        try (FileInputStream fileInputStream = openFileInput( name: title + ".txt")) {
            byte[] contents = new byte[fileInputStream.available()];
            fileInputStream.read(contents);
            editContent.setText(new String(contents));
            Toast.makeText( context: FileEdit.this, text: "Load successfully", Toast.LENGTH_LONG).show();
            return ;
        } catch (IOException ex) {
            Log.e( tag: "TAG", ex.toString());
            Toast.makeText( context: FileEdit.this, text: "Fail to load file.", Toast.LENGTH_LONG).show();
            return ;
        }
    }
});
```

## 监听 CLEAR

```
clear.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        editTitle.setText("");
        editContent.setText("");
    }
});
```

## 监听 DELETE，注意设置文件路径，并重新载入文件

```
delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String title = editTitle.getText().toString();

        //获取绝对路径
        File file = new File( pathname: getApplicationContext().getFilesDir().getAbsolutePath() + "/" + title + ".txt");
        //删除文件
        if (file.exists()) {
            file.delete();
            Toast.makeText( context: FileEdit.this, text: "Delete successfully", Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText( context: FileEdit.this, text: "Fail to delete file", Toast.LENGTH_LONG).show();
        }

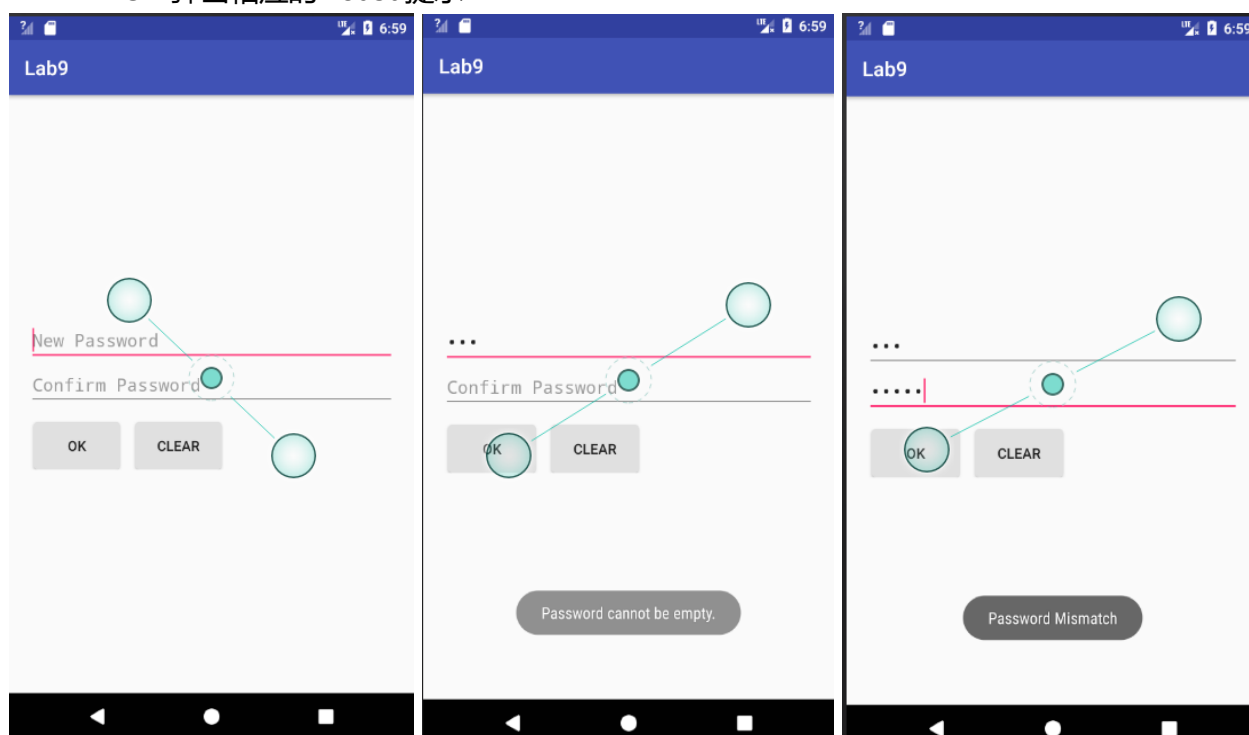
        //重新载入文件
        try (FileInputStream fileInputStream = openFileInput( name: title + ".txt")) {
            byte[] contents = new byte[fileInputStream.available()];
            fileInputStream.read(contents);
            editContent.setText(new String(contents));
            Toast.makeText( context: FileEdit.this, text: "Load successfully", Toast.LENGTH_LONG).show();
            return ;
        } catch (IOException ex) {
            Log.e( tag: "TAG", ex.toString());
            Toast.makeText( context: FileEdit.this, text: "Fail to load file.", Toast.LENGTH_LONG).show();
            return ;
        }
    }
});
```

### (3) 实验遇到困难以及解决思路

1. 删除文件时使用相对路径导致找不到文件，删除失败，更换为绝对路径  
`getApplicationContext().getFilesDir().getAbsolutePath() + "/" + title + ".txt"`解决
2. 实现进入文件编辑的 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，在 MainActivity 中设置静态变量 `mainActivity = this`;并在 FileEdit 创建时执行 `mainActivity.finish()`;终结 Activity 生命，则按返回键不会返回到密码输入界面，而是直接退出。

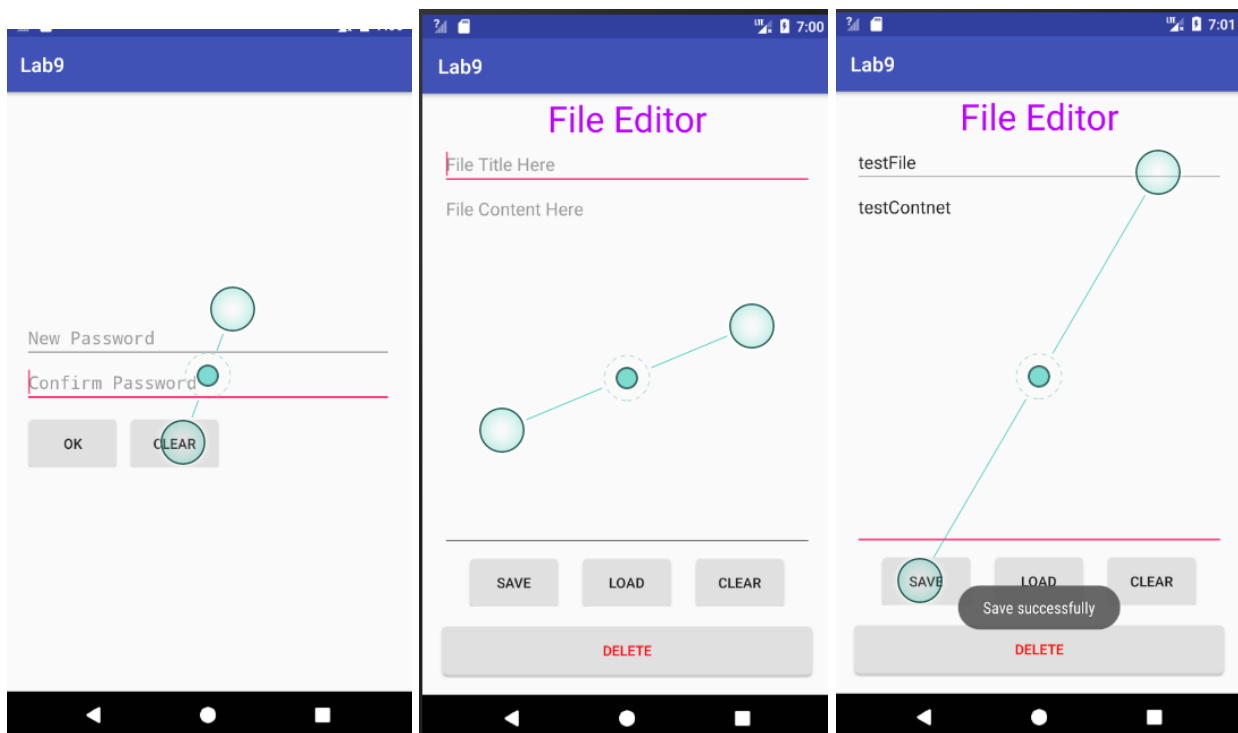
## 四、 课后实验结果

1. 播放、首次打开程序，显示两个密码框，当有空密码或密码与确认密码不匹配时，点击 OK 弹出相应的 Toast 提示



2. 点击 CLEAR 按钮清空输入，输入匹配的密码和确认密码点击 OK 保存并进入编辑界面，输入标题和内容点击 SAVE 储存文件并弹出 Toast 提示保存成功





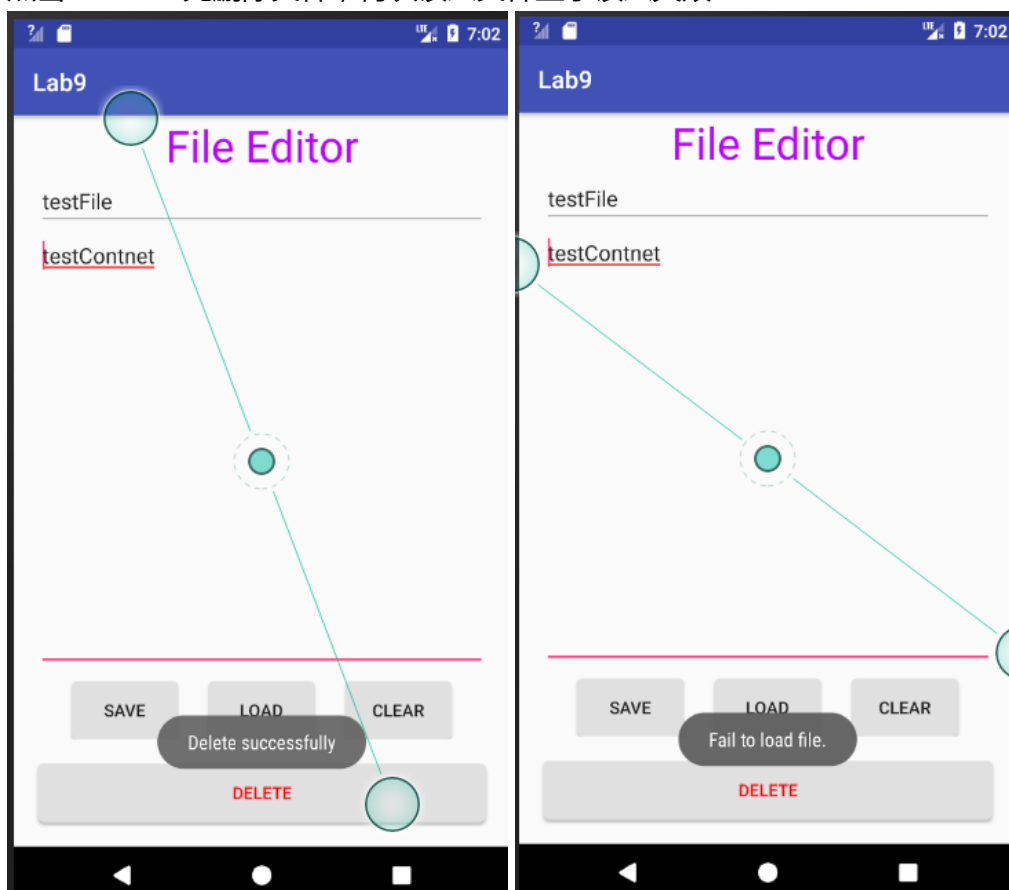
文件夹截图证明该文件已保存

Threads Heap Allocation Tracker Network Statistics File Explorer Emulator Control					
Name	Size	Date	Time	Permissions	
> com.example.gypc.e_dictionary	4096	2017-11-21	07:23	drwx-----	
▼ com.example.stellasong.lab9	4096	2017-11-28	07:01	drwx-----	
> cache	4096	2017-11-28	06:58	drwxrwx--x	
> code_cache	4096	2017-11-28	06:58	drwxrwx--x	
▼ files	4096	2017-11-28	07:01	drwxrwx--x	
testFile.txt	11	2017-11-28	07:01	-rw-rw----	
> shared_prefs	4096	2017-11-28	07:00	drwxrwx--x	

3. 点击 CLEAR 清空输入框，输入文件名后点击 LOAD 载入文件内容



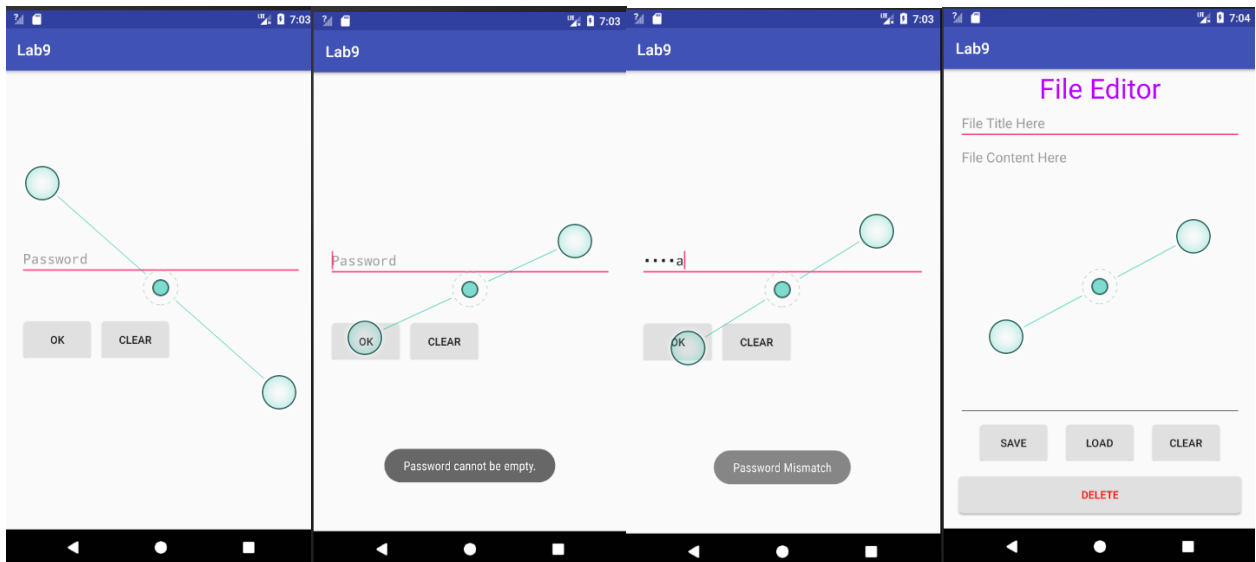
4. 点击 DELETE 先删除文件，再次读入文件显示读入失败



## 文件夹截图证明文件已删除

Threads   Heap   Allocation Tracker   Network Statistics   File Explorer   Emulator Control				
Name	Size	Date	Time	Permissions
> com.example.gypc.e_dictionary	4096	2017-11-21	07:23	drwx-----
▼ com.example.stellasong.lab9	4096	2017-11-28	07:01	drwx-----
> cache	4096	2017-11-28	06:58	drwxrwx--x
> code_cache	4096	2017-11-28	06:58	drwxrwx--x
files	4096	2017-11-28	07:02	drwxrwx--x
> shared_prefs	4096	2017-11-28	07:00	drwxrwx--x

5. 关闭应用后重新打开，显示密码输入框，测试输入为空或密码错误的提示，当输入密码正确时，跳转到文件编辑页面



## 五、 实验思考及感想

1. 安卓开发过程中有很多坑等我们去跳，需要多用搜索引擎多看文档
2. 遇到奇怪的问题不知道怎么解决时，可以重启试试