

Android 开发实验 6 报告

13331251 王嘉威

一.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

二.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。

4)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(相信大家都用过类似功能的东西吧):

(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(因为文件不存在嘛)

(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

三.实验步骤

1.创建 EditFileActivity 和 LoginActivity,按照实验要求编写对应 xml 布局界面.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:textColor="#8027df"
            android:textSize="12pt"
            android:text="Edit File"/>
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content">
            <TextView
                android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:text="File Name"/>
            <AutoCompleteTextView
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:id="@+id/fileName"/>
        </LinearLayout>
    </LinearLayout>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:text="File Content"/>
        ...
    </LinearLayout>
</LinearLayout>
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Welcome to Login"
            android:layout_gravity="center"
            android:id="@+id/header"
            android:textColor="#8027df"
            ...
        </TextView>
    </LinearLayout>
</RelativeLayout>
```

2.完成 EditFileActivity 和 LoginActivity 的控件初始化,注意 EditFileActivity 中的文件名自动补全需要为 AutoCompletedTextview 控件配置适配器:

```
private ArrayAdapter<String> av;

public void updateFileList() {
    av = new ArrayAdapter<String>(this, android.R.layout.simple_dropdown_item_1line, this.fileList());
    fileName.setAdapter(av);
}
```

3.创建 FileUtils 类,该类为文件操作的类,完成其对文件的读写删以及结果信息提示(Toast 实现)的方法接口.

```
public class FileUtils {
    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();
        }
    }

    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("");
        }
    }
}
```

4.在 EditFileActivity 中创建 FileUtils 对象,绑定控件事件,完成对文件的读写删的时间调用,注意每次读写删后需要更新文件目录适配器.

```
private FileUtils fileReader = new FileUtils();
```

```

save.setOnClickListener((v) -> {
    fileReader.saveContent(EditFileActivity.this, fileName.getText().toString(), content.getText().toString());
    updateFileList();
});

read.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        content.setText(fileReader.getContent(EditFileActivity.this, fileName.getText().toString()));
        updateFileList();
    }
});

del.setOnClickListener((view) -> {
    fileReader.deleteFile(EditFileActivity.this, fileName.getText().toString());
    fileName.setText("");
    content.setText("");
    updateFileList();
});
}
}

```

5.在 LoginActivity 中完成注册和登录并显示结果信息事件的绑定,注册和登录通过 SharedPreferences 来保存用户信息:

```

private SharedPreferences regInfo;

regBtn.setOnClickListener((v) -> {
    registInfo();
    Toast.makeText(getBaseContext(), "Register Success", Toast.LENGTH_SHORT).show();
});

logBtn.setOnClickListener((v) -> {
    if ((username.getText().toString().compareTo(regInfo.getString("user", "")) == 0) && (password.getText().toString().compareTo(regInfo.getString("password", "")) == 0) {
        startActivity(intent);
    } else {
        Toast.makeText(LoginActivity.this, "Login Error", Toast.LENGTH_SHORT).show();
    }
});
}
}

```

而 Remember password 通过在 onStart()方法中判断 checkbox 状态而决定是否从 SharedPreferences 中提取用户名和密码自动填写.

```

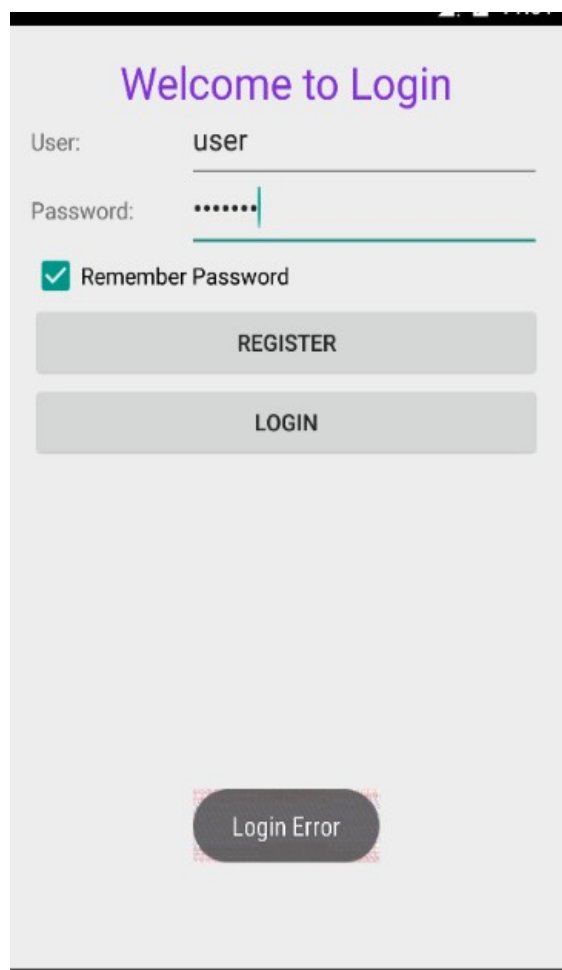
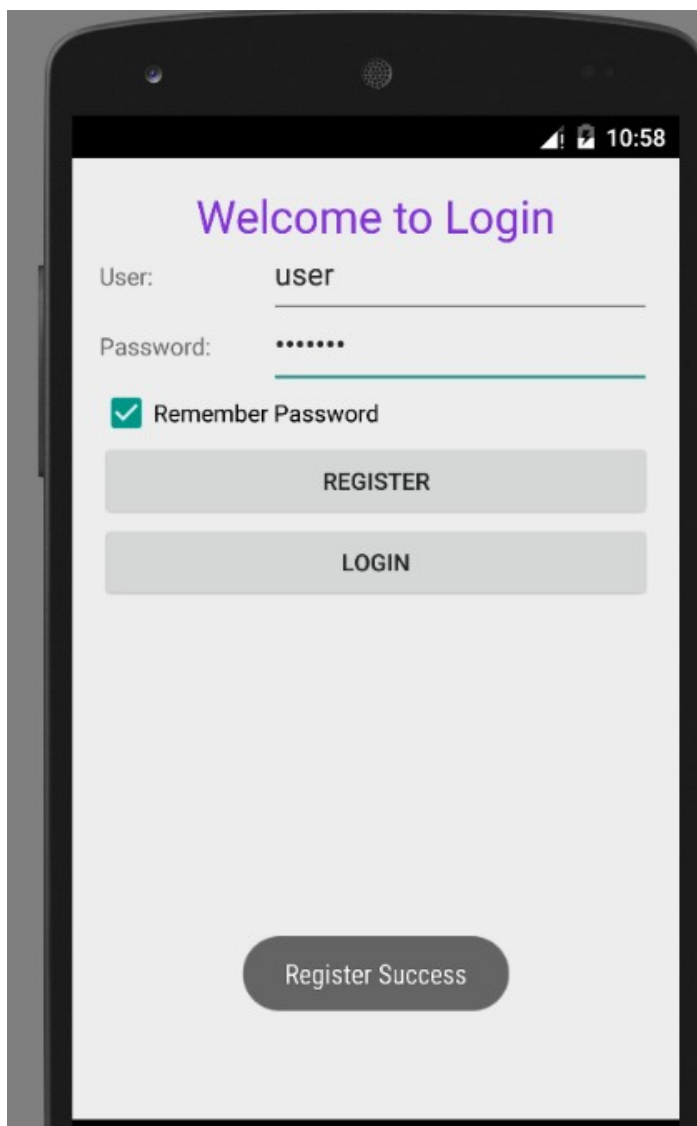
@Override
protected void onStart() {
    super.onStart();
    username.setText("");
    password.setText("");

    if (regInfo.getBoolean("remember", false)) {
        username.setText(regInfo.getString("user", ""));
        password.setText(regInfo.getString("password", ""));
        repass.setChecked(true);
    } else {
        repass.setChecked(false);
    }
}
}
}

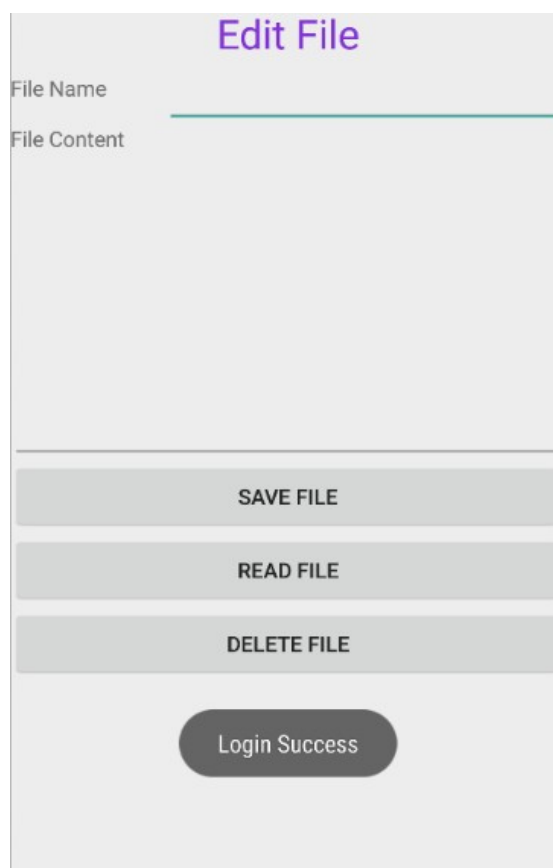
```

四.实验结果

1.注册和登录错误:



2.登录成功,进入文件操作界面:



3.保存文件及自动补全:

Edit File

File Namefile1

File Content

HAHAHA

SAVE FILE

READ FILE

DELETE FILE

Save Content Success

Edit File

File Namefil

File Content

file1

SAVE FILE

READ FILE

DELETE FILE

4.读取文件

Edit File

File Namefile1

File Content

HAHAHA

SAVE FILE

READ FILE

DELETE FILE

Read Content Success

5.删除文件(同时清空显示):

The screenshot shows the 'Edit File' app interface. At the top, the title 'Edit File' is in purple. Below it, there are two input fields: 'File Name' and 'File Content'. The 'File Name' field is empty, and the 'File Content' field is also empty. Below the input fields, there are three buttons: 'SAVE FILE', 'READ FILE', and 'DELETE FILE'. The 'DELETE FILE' button is highlighted. At the bottom, there is a dark gray rounded rectangle with the text 'Delete File Success' in white.

The screenshot shows the 'Edit File' app interface. At the top, the title 'Edit File' is in purple. Below it, there are two input fields: 'File Name' and 'File Content'. The 'File Name' field contains the text 'file1', and the 'File Content' field is empty. Below the input fields, there are three buttons: 'SAVE FILE', 'READ FILE', and 'DELETE FILE'. The 'READ FILE' button is highlighted. At the bottom, there is a dark gray rounded rectangle with the text 'Read Content Failed.' in white.

五.实验总结

这次的实验要点在对 SharedPreferences 和 Java 文件操作类的了解和使用,总体难度不大,实验的说明已经很详细了,实验的过程中也没有出现什么问题,要

注意的是 AutoCompletedTextView 控件在每一次文件操作的更新.而对文件的操作完全封装成了一个类来实现,体现了模块化的特点,这对以后开发的层次设计以及分工是很重要的经验.

参考资料:

1.<http://blog.csdn.net/wxyyxc1992/article/details/17222841>

手机:18819473218