

数据储存（一）

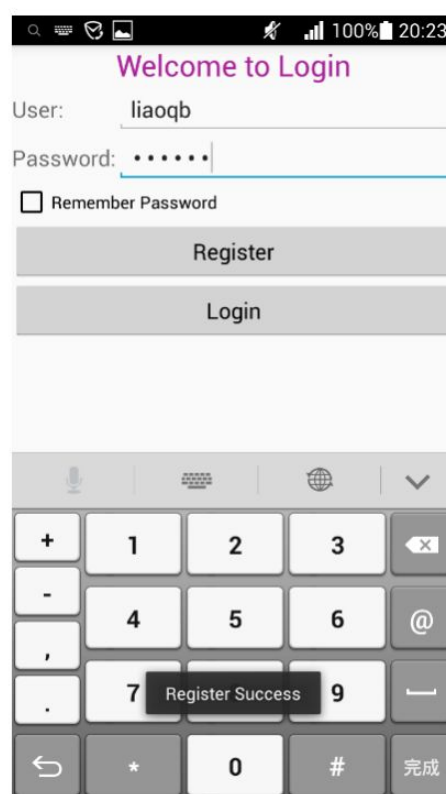
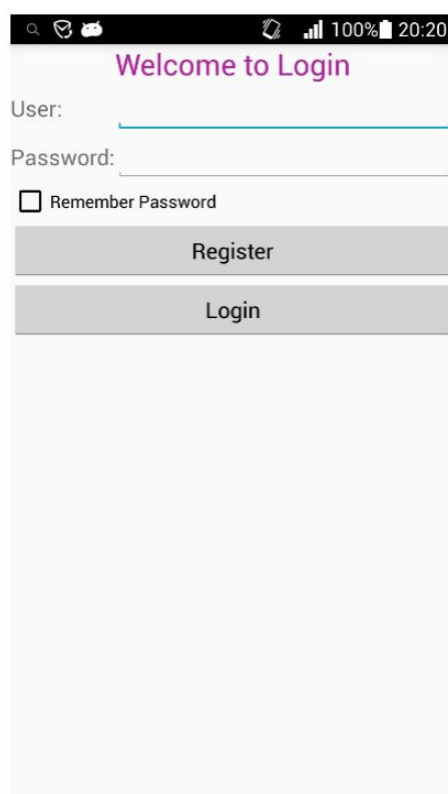
【实验目的】

- （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

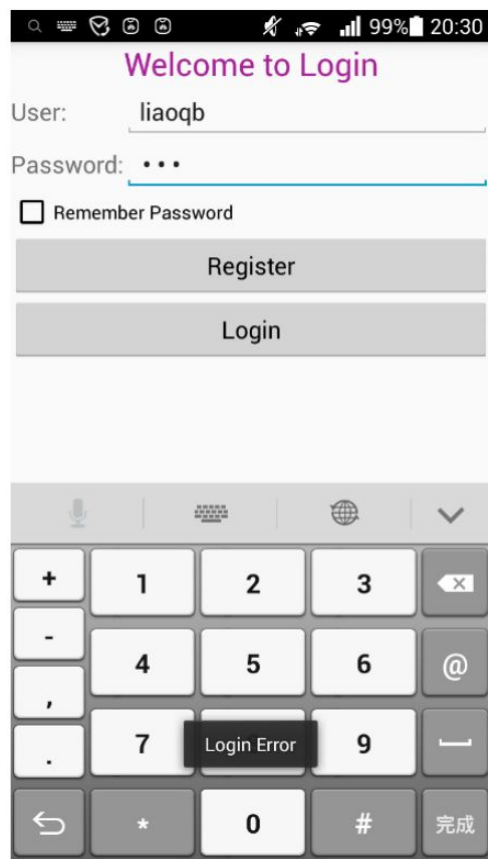
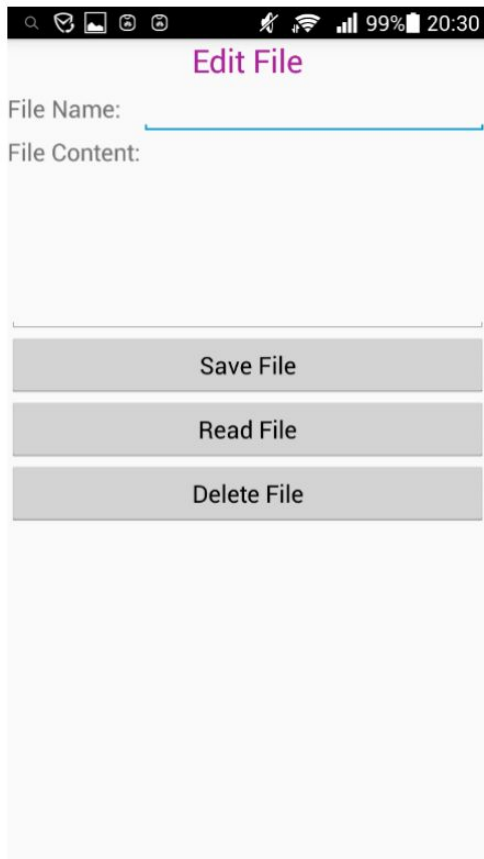
【实验内容】

- （1）登录界面：

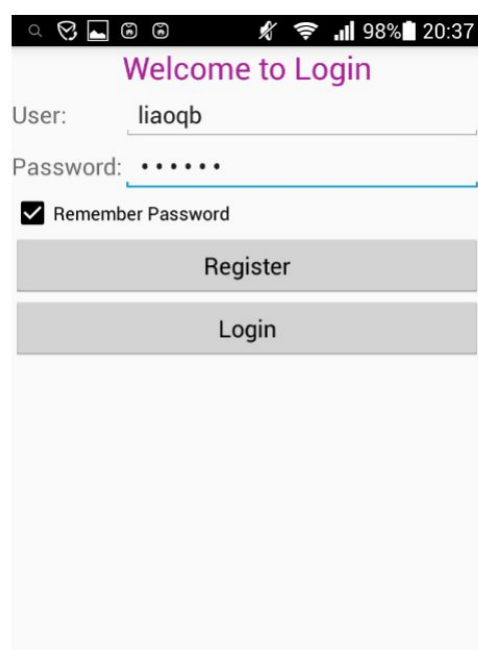
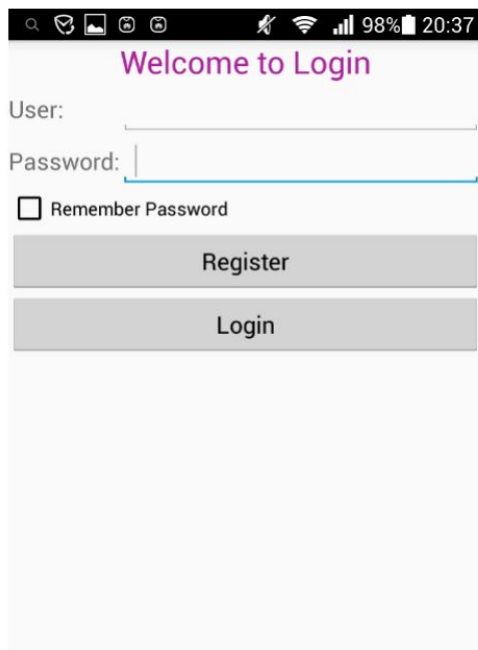
使用 SharedPreferences 记录登录状态，点击 Register 按钮，能够将 User 和 Password 写入 SharedPreferences，写入后使用 Toast 提示写入成功。



- （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

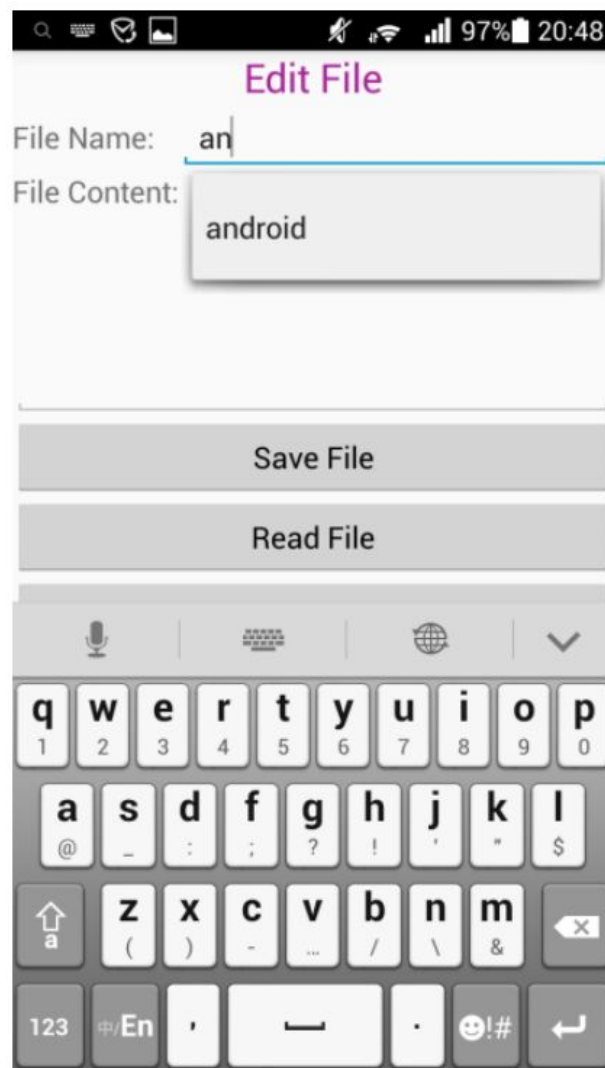


(3) 文件操作界面返回登录界面时，如果 Remember Password CheckBox 没有勾上，则 User 和 Password 控件不保存相应的信息，否则将 SharedPreferences 保存的信息自动填上相应的控件。

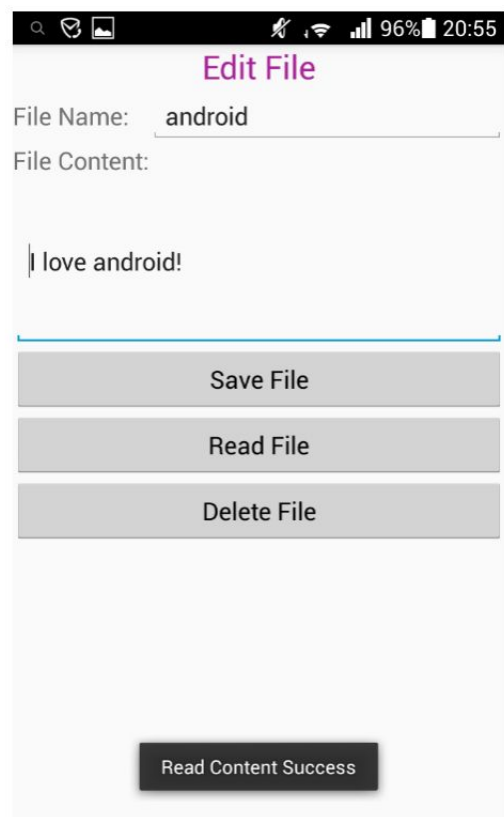
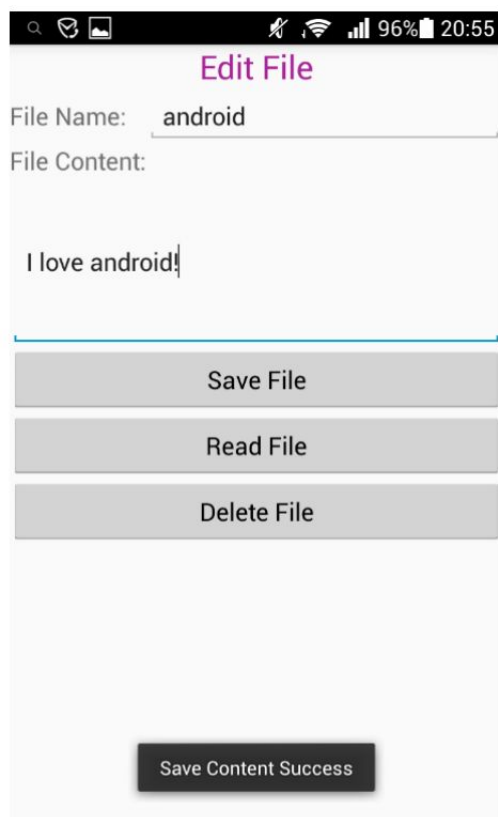


(4) 使用 AutoCompletedTextView 实现文件自动提示功能，如果文件存在，则该匹配的文件

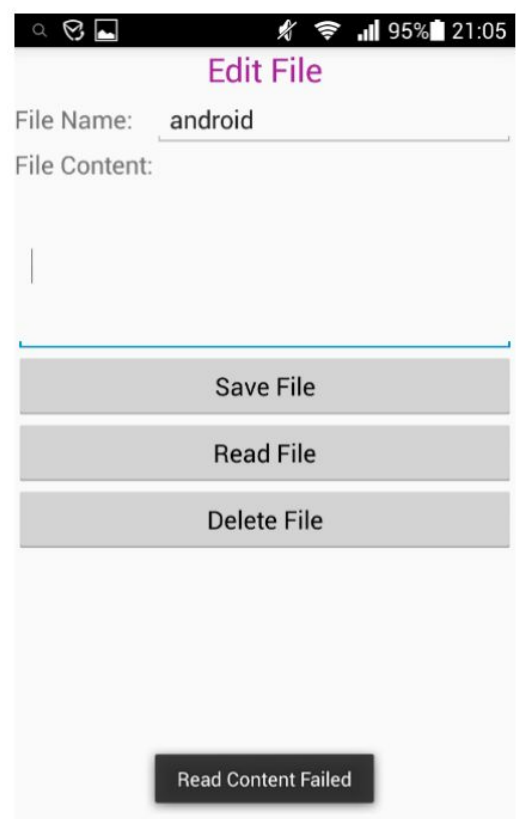
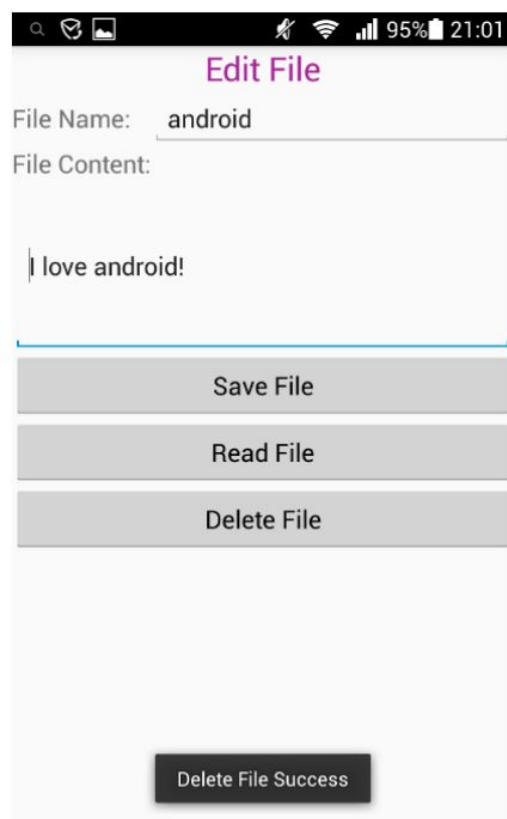
自动在 `AutoCompletedTextView` 下生成（相信大家都用过类似功能的东西吧）：



（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）

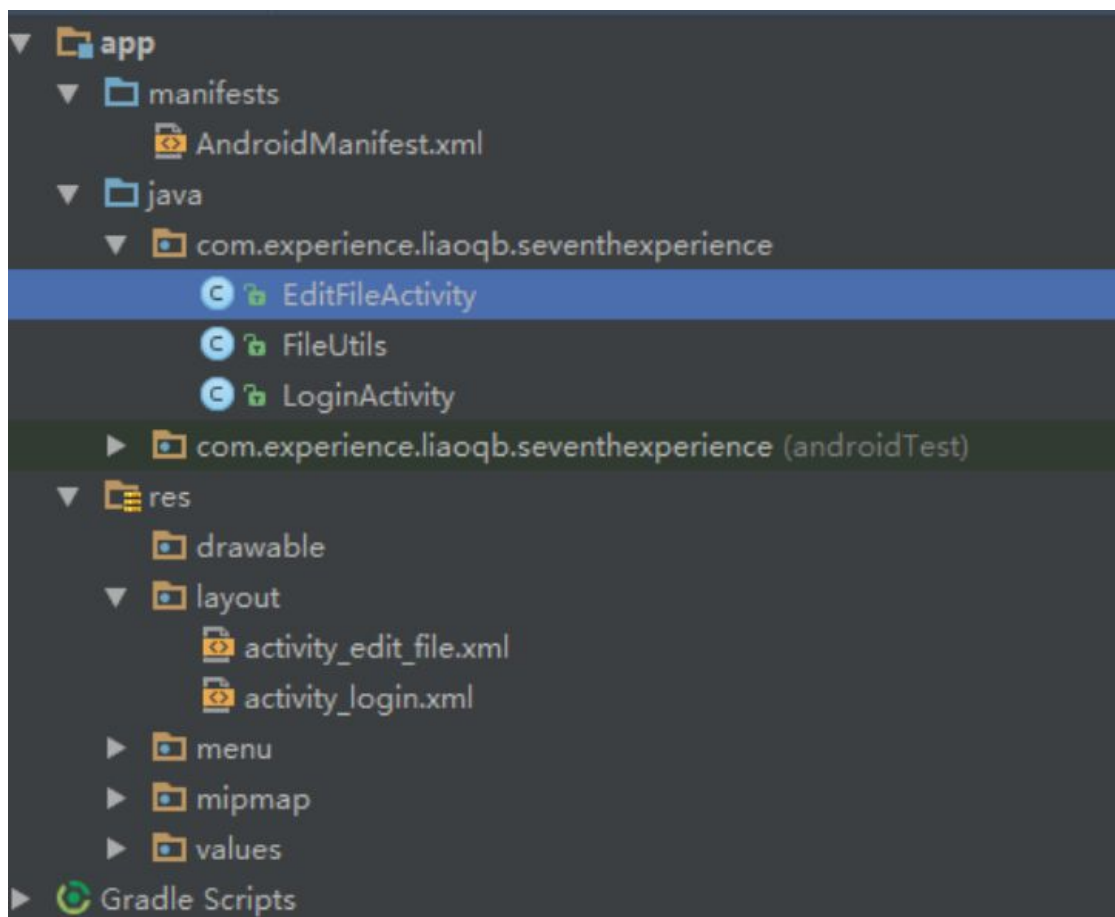
保存文件路径：/data/data/你的程序包名/files/*.txt

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

com.experience.liaoqb.seventhexperience	2015-09-12	13:28	drwxr-x--x
> cache	2015-09-12	13:27	drwxrwx--x
> files	2015-09-12	13:28	drwxrwx--x
android	16	2015-09-12	-rw-rw----
> lib	2015-09-12	13:27	drwxr-xr-x
> shared_prefs	2015-09-12	13:27	drwxrwx--x
data.xml	188	2015-09-12	-rw-rw----

【实验参考】

(1) 项目参考文件目录如下：



(2) AutoCompletedTextView 使用方法：

a) 可以看成是一般的控件

```
fileName = (AutoCompleteTextView)findViewById(R.id.file_name);
```

b) 设置适配器，并且将该适配器传递到控件之中：

```
fileName.setAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_dropdown_item_1line, this.fileList()));
```

其中，fileList 获取 Files 文件夹下所有文件名，返回类型为 String 数组（其实，联想到原来的 ListView 容器，不难发现容器类都需要设置适配器）

- c) 每一次文件目录的改变，都需要重新设置适配器，比如实验中的 Save File 和 Delete File，每次保存和删除文件后，都需要更新文件名数组和适配器；
- d) 其中，FileUtils 类为文件操作的类（又是可恶的面向对象）：

```
public class FileUtils {  
    public void saveContent(Context context, String fileName, String fileText) {...}  
  
    public String getContent(Context context, String fileName) {...}  
  
    public void deleteFile(Context context, String fileName) {...}  
}
```

比如其中的保存函数：

```
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();  
    }  
}
```

以及删除函数：

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

以及读取文件（这次实验重点是 SharedPreferences）

```

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("");
    }
}

```

其中，FileUtils 类被其他类调用，比如保存文件

```

saveFile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        fileUtils.saveContent(EditFileActivity.this, fileName.getText().toString(),
            fileContent.getText().toString());

        fileName.setAdapter(new ArrayAdapter<String>(EditFileActivity.this,
            android.R.layout.simple_dropdown_item_1line, EditFileActivity.this.fileList()));
    }
});

```

其中 File Content 控件可以使用下面语句，使其能够支持多行（如果文件很大，EditText 存不下，就需要这样写）

```

<EditText
    android:id="@+id/file_content"
    android:layout_width="wrap_content"
    android:minLines="5"
    android:inputType="textMultiLine" />

```

(3) SharedPreferences 使用方法:

- a) 获取当前 SharedPreferences:

```
SharedPreferences sharedPreferences = getSharedPreferences("data", MODE_PRIVATE);
```

(关于后面的模式,有兴趣的可以去查)

- b) 通过 SharedPreferences 保存的都是 K-V (键值对) 类型, 如果要编辑其中内容, 需要使用 Editor:

```
SharedPreferences.Editor editor = getSharedPreferences("data", MODE_PRIVATE).edit();

editor.putString("user", user.getText().toString());
editor.putString("password", password.getText().toString());

editor.commit();
```

编辑完成后记得调用 commit 方法写入文件之中

- c) 当 CheckBox 状态改变时, 需要调用 CheckBox 中的监听器:

```
isChecked.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
```

【扩展】

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。