

实验报告

课程名称： 智能终端软件开发技术

APP名称： 秋学期智能终端APP

专业： 信息安全

成员1姓名： 许多 学号： 3160103958

成员2姓名： 吕慕凡 学号： 3160102463

电子邮件地址： 3160103958@zju.edu.cn

手机： 17342019969

任课老师： 张 寅

目录

实验报告 1

一．系统简介 4

二．总体架构及功能划分 4

2.1 相关activity及功能 4

（1）LoginActivity 4

（2） SignupActivity 4

（3） MainMenuAvtivity 4

2.2 相关界面及功能 5

（1）activity\_login 5

（2） activity\_signup 6

（3） content\_main\_menu 7

（4） activity\_main\_menu 8

2.3 相关类及功能 8

（1）User 8

（2） MyMessageTools 8

（3） EntitiesStructure 8

（4） TriplesStructure 9

2.4 附加控件及功能 9

（1）selection\_action\_menu 9

（2） TextRatingBar 9

三．关键数据结构/算法 9

3.1 向指定服务器传输、返回数据 9

3.2 解析服务器传回的json数据 10

3.3 Unicode转码为中文字符 11

3.4 与服务器交互状态的显示 12

3.5 长按文字后出现的菜单的功能改写 12

（1）重新定义一个callback类 13

（2）重新定义长按后的菜单 13

（3）清除系统自带的“复制”、“全选”等菜单内容 13

（4）处理我们自定义的点击事件 13

四．开发困难及解决方案 14

4.1 登录成功以后的界面无法跳转 14

4.2 实体标注采用输入框输入标注文段中所有相同字符串 14

五．分工 14

六．总结 15

### 

### 一．系统简介

### 二．总体架构及功能划分

#### 2.1 相关activity及功能

###### （1）LoginActivity

主要实现登录功能，将用户输入的用户名和密码发送至服务端，在服务端进行登录。发送请求后接收服务器返回登录相关信息并将信息显示在屏幕上。登录的有效时长为 30min，若超过有效时长并为进行任何操作则自动登出。登录成功后跳转至MainMenuActivity。点击注册按钮则跳转至SignupActivity。

###### SignupActivity

主要实现注册功能，将用户输入的用户名、邮箱和密码发送至服务端，在服务端进行注册。发送请求后接收服务器返回的注册相关信息并将信息显示在屏幕上。点击取消按钮跳转回LoginActivity。

###### MainMenuAvtivity

进入时为欢迎界面，用户可点击左侧任务菜单栏查看用户个人信息以及选择进行实体标注或段落标注，点击下方退出按钮实现登出功能。

当用户选择实体标注时，界面获取的将会是网站上的实体相关信息，点击“下一页”会刷新为另一篇实体标注相关文本段落。长按段落中的文字并在长按后弹出的菜单选择将字体标注为红色作为人名还是标注为绿色作为职位。标注结束后可点击“提交”按键将信息提交至服务器，若提交成功，将有浮动窗口显示提交成功字样。

当用户选择关系标注时，界面获取的将会是网站上的关系标注相关信息，点击“下一页”会刷新为另一篇关系标注相关文本段落。文本段落下方为标注了的关系信息。点击“提交”按键可将信息提交至服务器，若提交成功，将有浮动窗口显示提交成功字样。

#### 2.2 相关界面及功能

###### （1）activity\_login

用户在用户名处输入用户名，在密码处输入密码，点击登录可进行登录，点击注册跳转至注册界面。

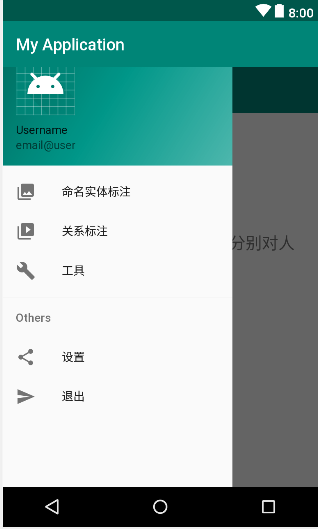
###### activity\_signup

用户在三个输入框中分别输入对应信息，点击注册可进行注册，已有账户的用户可点击取消返回登录界面。

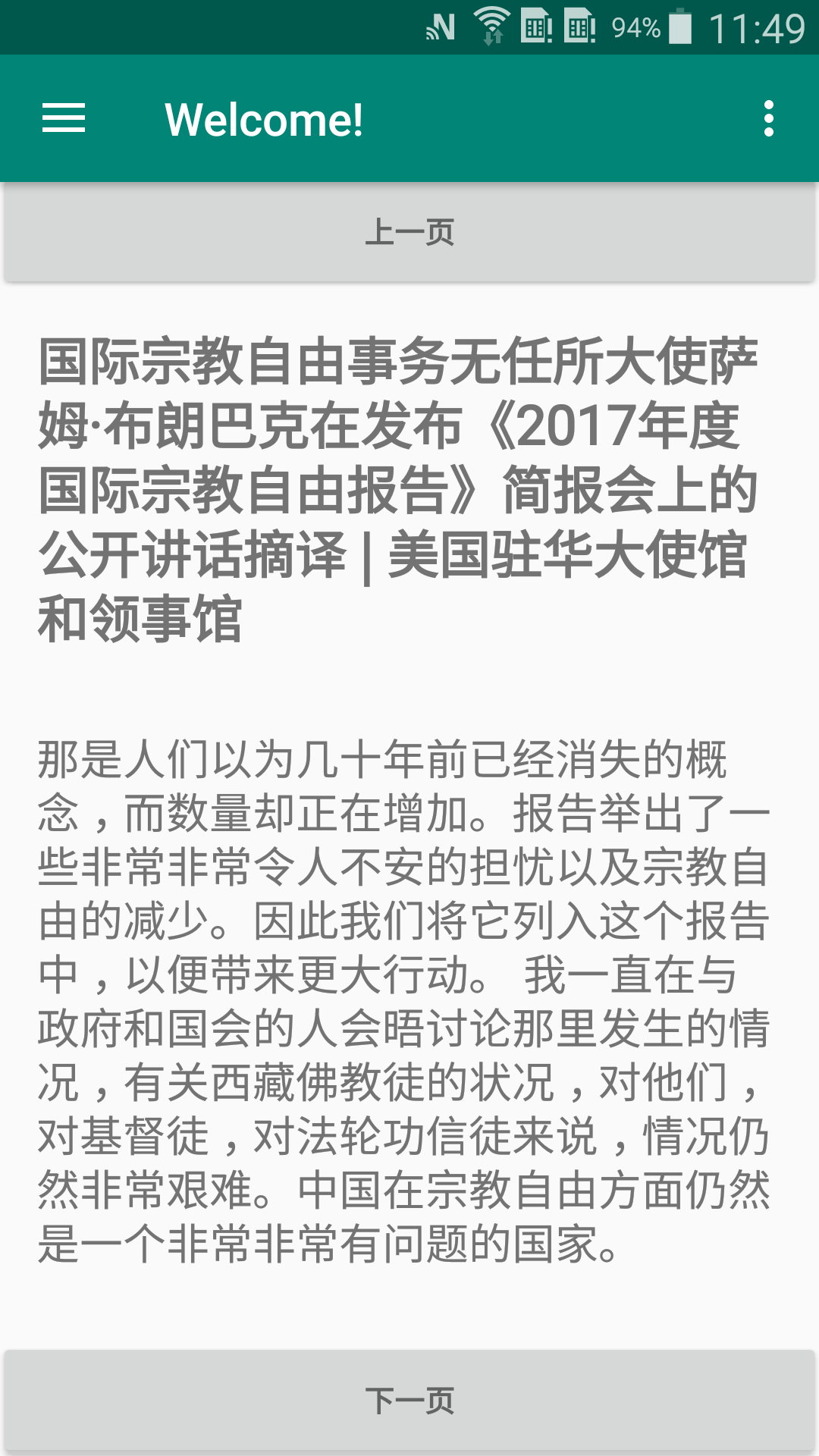
###### content\_main\_menu



###### activity\_main\_menu



点击命名实体标注可在content\_main\_menu看到已有实体信息



点击关系标注可在content\_main\_menu看到已有关系标注信息，点击退出可以登出服务器，返回至登录界面

#### 2.3 相关类及功能

###### （1）User

存储了用户的登录、注册等相关个人信息。用以实现用户登录成功后的token存储以及后续的token校验问题。

###### MyMessageTools

存储部分在APP各功能中重复使用的函数。

###### EntitiesStructure

存储实体标注相关信息，为后续上传做准备。

###### TriplesStructure

存储关系标注相关信息，为后续上传做准备。

#### 2.4 附加控件及功能

###### （1）selection\_action\_menu（此功能在6.0版本通过测试）

其功能为将长按文字后系统弹出的菜单自定义为我们想要的菜单名称，在本APP中应用在实体标注的部分，使长按文字后弹出的菜单为“人名（红）”和“职位（绿）”，用户选择对应的选项即可完成对实体的标注。



###### TextRatingBar

### 三．关键数据结构/算法

#### 3.1 向指定服务器传输、返回数据

在整个APP中的数据传输都使用了okhttp3的网络架构，用formbody的形式向服务器post数据并接收返回的json数据，在此仅用登录部分代码进行详细解释。其作用是向指定服务器以formbody形式post用户名和密码，并反馈登录信息。  
 OkHttpClient login\_client = new OkHttpClient();*//用okhttp的网络架构进行登录* RequestBody postBody = new FormBody.Builder()*//用formbody的形式向服务器传输用户名和密码* .add("username", *user*.getUsername())  
 .add("password", *user*.getPassword())  
 .build();  
 Request request = new Request.Builder()  
 .url("http://10.15.82.223:9090/app\_get\_data/app\_signincheck")*//指定访问的服务器地址* .post(postBody)  
 .build();  
 Call call = login\_client.newCall(request);  
 setProgressDialouge();*//显示联网状态* call.enqueue(new Callback() {  
 @Override  
 public void onFailure(Call call, IOException e) {  
 Log.*d*("CONNECTION", "请求失败 !!") ;  
 }*//对服务器请求失败* @Override  
 public void onResponse(Call call, final Response response) throws IOException {  
 progressDialog.dismiss();  
 Log.*d*("CONNECTION", "请求成功");  
 final String responseData = response.body().string() ;  
 parseJSONWithJSONObject(responseData);*//调用parseJSONWithJSONObject()方法来解析数据* }*//对服务器请求成功*

#### 3.2 解析服务器传回的json数据

服务器将json格式的数据返回后，要输出到用户界面中，必须要对其进行解析和Unicode转换。我们构造了一个函数来实现这个过程。

**private void** parseJSONWithJSONObject(String responseData){  
 String msg;  
 **try**{  
 JSONObject jsonObject = **new** JSONObject(responseData.substring(responseData.indexOf(**"{"**), responseData.lastIndexOf(**"}"**) + 1)) ;  
 String message = jsonObject.getString(**"msg"**);  
 msg = Message.*unicodeToUtf8*(message);*//对数据进行Unicode转码为中文字符* }**catch** (Exception e){  
 e.printStackTrace();  
 }  
}

#### 3.3 Unicode转码为中文字符

由于服务器返回的数据为Unicode编码，因此要正常显示到用户界面之前要对其进行转码。

**static** String unicodeToUtf8(String theString) {  
 **char** aChar;  
 **int** len = theString.length();  
 StringBuffer outBuffer = **new** StringBuffer(len);  
 **for** (**int** x = 0; x < len;) {  
 aChar = theString.charAt(x++);  
 **if** (aChar == **'\\'**) {  
 aChar = theString.charAt(x++);  
 **if** (aChar == **'u'**) {**int** value = 0;  
 **for** (**int** i = 0; i < 4; i++) {  
 aChar = theString.charAt(x++);  
 **switch** (aChar) {  
 **case '0'**:  
 **case '1'**:  
 **case '2'**:  
 **case '3'**:  
 **case '4'**:  
 **case '5'**:  
 **case '6'**:  
 **case '7'**:  
 **case '8'**:  
 **case '9'**:  
 value = (value << 4) + aChar - **'0'**;  
 **break**;  
 **case 'a'**:  
 **case 'b'**:  
 **case 'c'**:  
 **case 'd'**:  
 **case 'e'**:  
 **case 'f'**:  
 value = (value << 4) + 10 + aChar - **'a'**;  
 **break**;  
 **case 'A'**:  
 **case 'B'**:  
 **case 'C'**:  
 **case 'D'**:  
 **case 'E'**:  
 **case 'F'**:  
 value = (value << 4) + 10 + aChar - **'A'**;  
 **break**;  
 **default**:  
 **throw new** IllegalArgumentException(  
 **"Malformed \\uxxxx encoding."**);  
 }  
 }  
 outBuffer.append((**char**) value);  
 } **else** {  
 **if** (aChar == **'t'**)  
 aChar = **'\t'**;  
 **else if** (aChar == **'r'**)  
 aChar = **'\r'**;  
 **else if** (aChar == **'n'**)  
 aChar = **'\n'**;  
 **else if** (aChar == **'f'**)  
 aChar = **'\f'**;  
 outBuffer.append(aChar);  
 }  
 } **else** outBuffer.append(aChar);  
 }  
 **return** outBuffer.toString();  
}

#### 3.4 与服务器交互状态的显示

在传递数据到服务器时，如果有一个对话框能显示等待、正在连接诸如此类的信息，会让联网状态更加直观。

**private void** setProgressDialouge(){*//联网状态的显示* **progressDialog** = **new** ProgressDialog(LoginActivity.**this**) ;  
 **progressDialog**.setTitle(**"Please Wait"**);  
 **progressDialog**.setMessage(**"Connecting to server..."**);  
 **progressDialog**.setIndeterminate(**true**);  
 **progressDialog**.setCancelable(**false**);  
 **progressDialog**.show();  
}

#### 3.5 长按文字后出现的菜单的功能改写

在实体标注的时候，我们希望用户的标注方法可以是长按文字并选中后，点击文字上方浮现的菜单栏对应的标注信息来对实体进行标注，而此时则需要对菜单的选项和对应功能进行改写。

###### （1）重新定义一个callback类

**private** ActionMode.Callback **callback2**= **new** ActionMode.Callback()

（2）重新定义长按后的菜单

重新定义长按后的菜单为我们单独编写的“***selection\_action\_menu***”，并且使用getSelectionStart()、getSelectionEnd()两个函数对长按选中文字的位置信息进行标记，为后续的操作做准备。  
 **public boolean** onCreateActionMode(ActionMode actionMode, Menu menu) {  
 MenuInflater menuInflater = actionMode.getMenuInflater();  
 menuInflater.inflate(R.menu.***selection\_action\_menu***,menu);  
 **int** start = **responseText**.getSelectionStart();*//获取选择部分的起始信息* **int** end = **responseText**.getSelectionEnd();  
 System.***out***.print(start);  
 System.***out***.print(end);  
 **return true**;*//返回false则不会显示弹窗* }

（3）清除系统自带的“复制”、“全选”等菜单内容  
 **public boolean** onPrepareActionMode(ActionMode actionMode, Menu menu) {  
 MenuInflater menuInflater = actionMode.getMenuInflater();  
 menu.clear();  
 menuInflater.inflate(R.menu.***selection\_action\_menu***,menu);  
 **return true**;  
 }

（4）处理我们自定义的点击事件

本部分功能为当用户点击了对应想标注的类型，如人名时，界面上会浮现弹窗“人名”，对应选中文字变为红色，并且相关实体标注信息将被存储在实体结构中，为后续上传做准备。  
 **public boolean** onActionItemClicked(ActionMode actionMode, MenuItem menuItem) {  
 *//根据item的ID处理点击事件* **int** start = **responseText**.getSelectionStart();*//获取选择部分的起始信息* **int** end = **responseText**.getSelectionEnd();  
 **switch** (menuItem.getItemId()){  
 **case** R.id.***name***:  
 Toast.*makeText*(MainMenuActivity.**this**, **"人名"**, Toast.***LENGTH\_SHORT***).show();  
 SpannableStringBuilder styled1 = **new** SpannableStringBuilder(**responseText**.getText());  
 styled1.setSpan(**new** ForegroundColorSpan(Color.***RED***), start,end,

Spannable.***SPAN\_EXCLUSIVE\_EXCLUSIVE***);  
 String EntityName = **responseText**.getText().toString().substring(start,end);  
 **Entities**.getEntityName().add(EntityName);  
 **Entities**.getStart().add(start);  
 **Entities**.getEnd().add(end);  
 **Entities**.getNerTag().add(**"PERSON"**);  
 **responseText**.setText(styled1);*//更改选中部分的颜色* actionMode.finish();*//收起操作菜单* **break**;  
 **case** R.id.***job***:  
 Toast.*makeText*(MainMenuActivity.**this**, **"职位"**, Toast.***LENGTH\_SHORT***).show();  
 SpannableStringBuilder styled2 = **new** SpannableStringBuilder(**responseText**.getText());  
 styled2.setSpan(**new** ForegroundColorSpan(Color.***GREEN***), start,end,

Spannable.***SPAN\_EXCLUSIVE\_EXCLUSIVE***);  
 String EntityName1 = **responseText**.getText().toString().substring(start,end);  
 **Entities**.getEntityName().add(EntityName1);  
 **Entities**.getStart().add(start);  
 **Entities**.getEnd().add(end);  
 **Entities**.getNerTag().add(**"TITLE"**);  
 **responseText**.setText(styled2);  
 actionMode.finish();  
 **break**;  
 }  
 **return true**;*//返回true则系统的"复制"、"搜索"之类的item将无效，只有自定义item有响应* }

### 四．开发困难及解决方案

#### 4.1 登录成功以后的界面无法跳转

解决方案：界面跳转是在UI线程之外的，在之前的输出信息部分，我们采用把Toast放在Looper之间在进行成功输出的方法，但是在Looper之外的界面跳转功能仍然无法实现，最后我们将Looper换成runonUIthread的方法，跳转成功。

#### 4.2 实体标注采用输入框输入标注文段中所有相同字符串

解决方案：放弃输入框输入标注的做法，采用改写callback的方式，让用户可以采用长按选中一段文字的方式并点击长按后菜单中对应选项对这段文字进行标注，使用户的标注方式更加自由化，并且提高用户标注的准确性和交互体验。

### 五．分工

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 完成任务 |
| 许多 | 3160103958 | 登录、注册、实体标注、屏幕滚动 |
| 吕慕凡 | 3160102463 | 内容获取、结果上传、关系标注、登出、字体大小 |

### 六．总结