

**计算机与信息科学学院**

**Android项目报告**

**课 程：Android课程设计**

**项目名称：人脸识别签到**

**专业年级：计算机科学与技术2013级**

**姓 名：梁 轩**

**学 号：20130720080**

**指导老师：曾 智**

**2016年6月20日**

**计科2班第组任务分配情况**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号** | **姓名** | **角色** | **任务** | **打分** |
| 20130720080 | 梁轩 | 组长 | 统筹全局，各个功能模块的整合，代码实现 |  |
| 20130720065 | 陈克融 | 组员 | 信息校验和存储模块 |  |
| 20130516102 | 刘刈 | 组员 | 人脸识别模块 |  |
| 20130516119 | 马鑫 | 组员 | 实现对整个项目界面的布局，规划，排版。 |  |
| 20130516135 | 秦凯 | 组员 | 签到与活动模块 |  |
| 20120516121 | 盛润浩 | 组员 | 登录注册模块 |  |

目录

[摘要： 4](#_Toc454265636)

[1 项目背景说明 4](#_Toc454265637)

[2 项目概述 4](#_Toc454265638)

[2.1项目设计的目的和意义 4](#_Toc454265639)

[2.2 项目设计的主要内容 4](#_Toc454265640)

[2.3 项目设计实现的主要功能 5](#_Toc454265641)

[3 需求分析 5](#_Toc454265642)

[3.1 系统的总体需求 5](#_Toc454265643)

[3.2 系统需求分析 5](#_Toc454265644)

[3.3 系统功能的需求说明 6](#_Toc454265645)

[3.4 系统的可行性分析 6](#_Toc454265646)

[4 系统设计与实现 7](#_Toc454265647)

[4.1 系统的设计目标 7](#_Toc454265648)

[4.2 系统的总体设计 7](#_Toc454265649)

[4.3 系统总体流程图 8](#_Toc454265650)

[4.4 系统功能模块图 9](#_Toc454265651)

[4.5 发布签到活动详细设计 10](#_Toc454265652)

[4.6 签到活动详细设计 10](#_Toc454265653)

[5 项目演示 12](#_Toc454265654)

[5.1 界面的实现与效果 12](#_Toc454265655)

[6 总结 16](#_Toc454265656)

**人脸识别签到**

## 摘要：

本项目是一个利用人脸识别进行签到的项目，主要使用了高德地图定位，一登的SDK，Bmob的SDK，BaseRecyclerViewAdaptHelper框架等技术实现，提高了java的使用熟悉程度，本app界面简单且简洁大方，且高度实用。

**关键字：**安卓平台，人脸识别，地图定位，签到

## 1 项目背景说明

我们小组注意到老师上课总有学生逃课不来或者找人代课。学生活动中此类事件也时有发生。为了杜绝此类事件的发生，可以开发一款人脸识别签到APP。通过APP老师和学生可以发布课程或者活动，而参与者也可通过APP签到确认是本人到达了现场参与了活动。

## 2 项目概述

### 2.1项目设计的目的和意义

随着移动硬件配置的不断提高，手机的功能也是越发的强大。而人们对于手机的依赖也不只是打个电话、发个短信而已，许多的手机软件的开发，大大的丰富了现代人的生活。而随着Android智能平台的出现，人们也需要更多、更好的手机软件来满足自己的需求。由于传统点名的繁琐费时，我们就想开发一个可以快速签到的系统代替点名。这样可以提高效率，也可以让我们更加熟悉Java环境的开发。

### 2.2 项目设计的主要内容

在对本课题进行设计的时候，首先要熟悉Android平台的结构和开发环境，也就是要了解Android SDK软件开发中的一些类库和熟悉工具包的使用及Android平台提供的多个接口函数。

在做好了准备后就是对签到系统流程的研究，虽然想法容易，但是实现出来还是需要流程、框架的设计等。

### 2.3 项目设计实现的主要功能

该项目用到的主要功能:

* 正则表达式校验提示注册登录信息
* 人脸绑定、登录与人脸校验
* 地理围栏提示签到功能
* 发布活动功能
* 签到活动功能
* 位置时间校验功能

## 3 需求分析

需求分析是软件工程中非常重要的一步，充分的需求分析是设计好程序的前提。这里主要包括系统的总体需求、系统的需求分析、系统的需求说明和系统的可行性分析。

### 3.1 系统的总体需求

现在无论是在公司还是在学校，很多地方都在使用考勤系统，一般来说使用的是指纹考勤机，打卡考勤机等，价格昂贵且麻烦，开发一个手机上面的签到APP既可以减少成本，也可以方便大家。

本次的课程设计的课题就是在此基础上提出总体的功能需求。主要就是能够实现签到功能。

### 3.2 系统需求分析

本次的课程设计主要就是在基于Android平台下设计并开发一个签到软件。运用当前高德提供的Android SDK来实现对定位模块。使用的数据库是Bomb后台的数据库。系统的总体的功能业务可以主要描述为以下几点：

* 登陆界面功能

登陆界面是在运行该软件时的第一个要显示的界面，出现在签到界面之前。主要就是注册登陆模块。只要用户开启该软件，就会首先出来欢迎界面。

* 主界面菜单选项功能

主界面菜单的功能主要是让用户解除绑定、查看签到列表、发起签到活动、查看活动列表功能。界面的主要需求就是简单明了，在加上清新明目的背景图片，一定会给用户视觉上的快感。主界面这里拒绝那些繁琐的操作，给用户以方便。

* 使用者定位功能

定位功能主要是让用户定位，并查看周围的签到活动。通过地理围栏判断是否满足签到条件。

### 3.3 系统功能的需求说明

根据对本课题的主要需求的分析做如下说明：

1）登陆界面主要就是用户的注册、登陆、录入人脸特征。

2）主界面上主要就是主要是让用户解除绑定、查看签到列表、发起签到活动、查看活动列表功能。

3）定位功能主要是让用户定位，并查看周围的签到活动。通过地理围栏判断是否满足签到条件。

4）签到活动功能主要是让发起签到者设置签到活动名称、活动说明、签到时间、限时、范围限制。

### 3.4 系统的可行性分析

* 复杂分析

本系统的数据库文件是选用网上已经存在的数据库文件，所以进行设计与开发时候都要进行数据库方面的设计，来满足本系统的功能实现。要对已经存在的数据库文件进行分析，了解数据库文件中的表名和表的结构，字段等属性。在导入单词库要实用Eclipse里面打开DDMS功能窗口的File Explorer功能。

而在单词的查询和模糊查询也需要对数据库进行操作，需要用到数据库帮助类，还需要写出相应的方法，实现查询的功能，特别是模糊查询，模糊查询时候要进行列举多个单词，对数据库操作时候要使用DatabaseHelper中的相应方法。

* 可行性

随着Android的快速发展和不断完善，Android的应用是越发的广泛。所以基于Android平台的手机小程序也是越来越多、越来越受欢迎。而一款好用且功能强大的签到软件可以为企业、教师用户提供随时随地的服务，相信很多的用户都会选择享受这种服务。因而从服务的角度来说，具有可行性。

通过以上的需求分析和对系统复杂性的分析，基于Android的人脸识别签到的开发具有现实性。综述，该课题的设计开发具有可行性。

## 4 系统设计与实现

### 4.1 系统的设计目标

通过对现有网上的签到应用的了解，在现有签到应用的基础上，设计一个加入人脸识别和地理位置双验证的签到系统，这个创新主要体现在人脸识别上。也就是在实现定位签到的同时，实现人脸识别的功能。所以这也是此次课程设计所要达到的设计目标。

### 4.2 系统的总体设计

签到软件需要具备的功能主要就是身份识别，发布活动，定位签到。所以我们主要是围绕以下主要功能来实现的。该课题主要是在Windows的操作系统下搭建，在Android平台上开发和运行。总体设计图如图4–1：

签到app

发布活动

签到活动

Android平台

Java编程语言

Android模拟器

Android SDK

Eclipse

Windows系统

图4.1 系统的总体设计

### 4.3 系统总体流程图

在启动Android的模拟器以后，系统首先显示注册登陆页面，选择注册按钮可以输入注册信息，录入人脸信息，并获取注册码完成注册。选择找回密码功能可以通过发送邮件到注册邮箱完成找回。

启动模拟器

注册登陆结果显示

引导界面

注册登陆找回密码界面

开始

结束

签到功能模块

图4–2系统总体流程图

### 4.4 系统功能模块图

系统的功能框架，主要分为对使用者的定位、签到活动。本课题模块结构示意如图4–5：

基于Android的签到app

活动签到功能

发布活动功能

输出活动

输出签到

登录模块功能

登录校验

图4–5系统的功能模块图

### 4.5 发布签到活动详细设计

发布签到活动就是在用户选择签到活动功能后，进入发布签到活动界面，输入活动名称、活动说明、签到时间、结束时间、设置位置和位置范围。其主要的流程如图4–3：



图4–3发布签到活动流程图

### 4.6 签到活动详细设计

签到发布活动就是用户在活动列表或者地图上点击签到的活动，进入签到界面，输入签到信息、签到说明、人脸识别，后台自动校验签到时间、签到位置和人脸是否匹配。其主要的流程如图4–4：



图4–4签到活动流程图

## 5 项目演示

### 5.1 界面的实现与效果

图5-1 引导界面和登录模块界面

登录功能代码：

*/\*\*  
 \* 用户登录  
 \*  
 \** ***@param*** *mEmail\_login 电子邮件  
 \** ***@param*** *mPassword\_login 密码  
 \*/*public void Login(EditText mEmail\_login, EditText mPassword\_login) {  
 String email = mEmail\_login.getText().toString().trim();  
 String passwd = mPassword\_login.getText().toString().trim();  
 final StringBuffer s=new StringBuffer("");  
 if (TextUtils.*isEmpty*(email)){  
 s.append(context.getString(R.string.*email\_no\_null*));  
 }  
 if (TextUtils.*isEmpty*(passwd)){  
 s.append(context.getString(R.string.*pwd\_no\_null*));  
 }  
 if (!TextUtils.*isEmpty*(s.toString())){  
 Toast.*makeText*(context, s.toString(), Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 final Dialog dialog = DialogView.*createLoadingDialog*(context, context.getString(R.string.*loginrun*));  
 dialog.show();  
  
  
 BmobUser.*loginByAccount*(context, email, passwd, new LogInListener<MyUser>() {  
 @Override  
 public void done(MyUser myUser, BmobException e) {  
 if (e != null) {  
 Log.*e*("TAG", e.toString()+e.getErrorCode());  
 Toast.*makeText*(context,context.getString(R.string.*usernameorpassworderror*), Toast.*LENGTH\_SHORT*).show();  
 } else {  
 LogUtils.*e*(LogUtils.*getLineInfo*(),"done");  
 intent = new Intent(context, MainActivity.class);  
 context.startActivity(intent);  
 activity.overridePendingTransition(android.R.anim.*slide\_in\_left*, android.R.anim.*slide\_out\_right*);  
 activity.finish();  
 }  
 dialog.dismiss();  
 }  
 });  
  
}

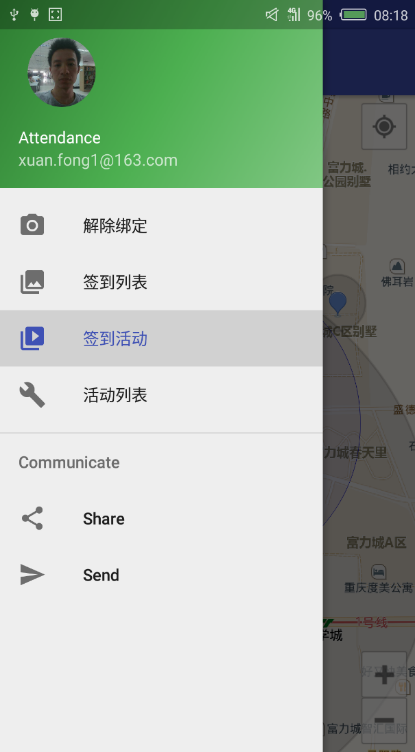
 

图5-2 地图和菜单界面

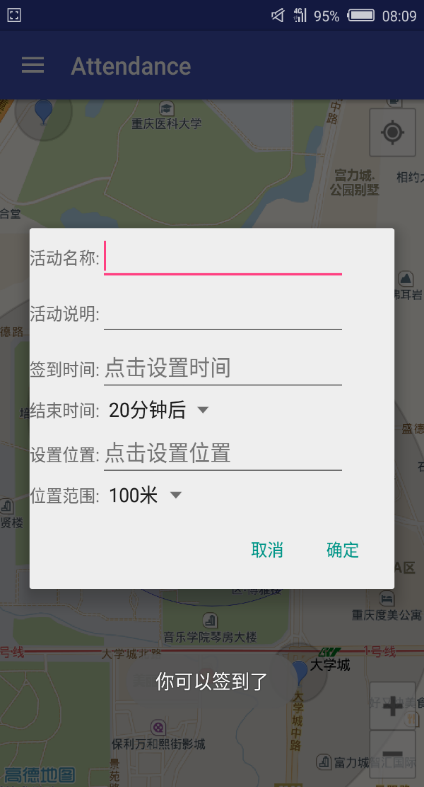
 

图5-3 发布活动信息和签到信息录入界面

签到信息校验功能代码：

public void attendance\_proess(final Event mEvent, final Attendance mAttendance) {  
 // 计算量坐标点距离  
 Location location = new Location(context);  
 location.setlocationlist(new Location.LocationCallback() {  
 @Override  
 public void onLocationlist(final AMapLocation aMapLocation) {  
 float distance = AMapUtils.*calculateLineDistance*(new LatLng(aMapLocation.getLatitude(), aMapLocation.getLongitude()), mEvent.getLocation());  
  
 if (distance <= mEvent.getLocationrangr()) {  
  
 View event\_dialog = View.*inflate*(context, R.layout.*attendance\_dialog*, null);  
 final EditText ed\_time = (EditText) event\_dialog.findViewById(R.id.*ed\_time\_attend*);  
 final EditText ed\_name = (EditText) event\_dialog.findViewById(R.id.*ed\_name\_show*);  
 final EditText ed\_name\_attend = (EditText) event\_dialog.findViewById(R.id.*ed\_name\_atted*);  
 final EditText ed\_msg = (EditText) event\_dialog.findViewById(R.id.*ed\_msg\_show*);  
 final EditText ed\_msg\_attend = (EditText) event\_dialog.findViewById(R.id.*ed\_msg\_attend*);  
  
 ed\_time.setText(mEvent.getTime() + "开始，" + mEvent.getTimerange() + "后结束");  
 new MyCountDownTimer(aMapLocation.getTime(), mEvent.getTime(), mEvent.getTimerange(), 1000, ed\_time, "验证手机").start();  
 ed\_msg.setText(mEvent.getMsg());  
 ed\_name.setText(mEvent.getName());  
 android.app.AlertDialog.Builder builder = new android.app.AlertDialog.Builder(context);  
 builder.setView(event\_dialog);  
 builder.setCancelable(false);  
 builder.setNegativeButton("取消", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 }  
 });  
 builder.setPositiveButton("确定", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 String name\_attend = ed\_name\_attend.getText().toString().trim();  
 String msg\_attend = ed\_msg\_attend.getText().toString().trim();  
 final StringBuffer s = new StringBuffer("");  
 if (TextUtils.*isEmpty*(name\_attend)) {  
 s.append("名字不能为空");  
 }  
 if (TextUtils.*isEmpty*(msg\_attend)) {  
 s.append("信息不能为空");  
 }  
 if (!TextUtils.*isEmpty*(s.toString())) {  
 Toast.*makeText*(context, s.toString(), Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 mAttendance.setName(name\_attend);  
 mAttendance.setEvent(mEvent);  
 mAttendance.setMsg(msg\_attend);  
 mAttendance.setLocation(new LatLng(aMapLocation.getLatitude(), aMapLocation.getLongitude()));  
 mAttendance.setUser(BmobUser.*getCurrentUser*(context));  
 SuperID.*faceVerify*(activity, 3); //刷脸验证  
 }  
 });  
 builder.create().show();  
  
 } else {  
 Toast.*makeText*(context, "你不在签到范围", Toast.*LENGTH\_SHORT*).show();  
 }  
 Toast.*makeText*(context, "距离为：" + distance + "距离为" + mEvent.getUser\_name(), Toast.*LENGTH\_SHORT*).show();  
  
 }  
 });  
 location.start();  
}

图5-4 活动列表和签到列表界面

## 6 总结

基于Android平台的人脸识别签到的基本功能都已实现，但是也有很多的不完善之处。经过几周的开发，对我们组做的课程设计做一个小结。

此次课程设计主要针对Google推出的Android平台下开发签到软件，并设计软件的界面和流程。这个系统的数据库基于Bomb后台。功能还有一些缺陷比如发布签到活动人员的权限，自定义签到限时，过多签到活动后的筛选等需要完善。

总的来说通过这次Android项目开发学到了很多知识，学会了小组分工合作，明白了团队合作很重要。