

文章编号: 2096-1472(2017)-01-40-02

WebService在基于.Net架构的招聘报名管理系统向 Android系统拓展开发中的设计与应用

魏 晋

(长治医学院计算机教学部, 山西 长治 046000)

摘 要: 虽然大多数高校已有网站形式的招聘报名管理系统, 但是如何能让管理人员随时随地、更加灵活的管理报名工作环节, 让报考人员不受电脑的限制在手机上完成报名、查看审核结果、及时关注报考进度和消息等操作。在此背景下提出了, 引入WebService并对API进行自定义重新编程的方法, 实现了在与已有的招聘报名管理系统共用底层数据库的基础上, 再拓展开发了基于Android平台的招聘报名管理系统, 满足了为管理和报考人员提供在Android平台上完成相关工作和业务的需求。

关键词: Android平台; WebService; API编程

中图分类号: TP311.5 **文献标识码:** A

Design and Application of Web Service in the Extension Development of the .Net-Based Recruitment Registration Management System to Android System

WEI Jin

(Department of Computer Teaching, Changzhi Medical College, Changzhi 046000, China)

Abstract: With the current web-based recruitment registration management systems in most colleges and universities, it is still not flexible enough for administrators to manage the recruitment process and not convenient enough for applicants to apply, register or check information through mobile phones. Adopting Web Service technology and reprogramming the APIs, this study extends the existing recruitment registration management system to Android system, based on the shared underlying database, which effectively meets the requirements of both management and application operation.

Keywords: android system; web service; API programming

1 引言(Introduction)

当前国内大部分高校的招聘报名工作都已经采用了依托网络的在线招聘报名方式。依托于网络的在线招聘报名方式, 因为实现了工作环节自动化, 所以在节省了大量人力和财力投入的同时, 也提高了高校招聘报名工作的效率。随着智能手机的普及、手机功能的多样化, 以及移动互联网的飞速发展, 人们更习惯于使用手机连接互联网, 在手机上进行各种互联网的业务操作^[1,2]。因此结合现有招聘报名工作的实际情况, 在已有的在线招聘报名管理系统的基础上, 再开发一套基于Android平台的招聘报名管理系统, 为管理和报考人员提供第二种完成相关管理工作与报考操作的渠道, 对于高校招聘报名工作来说具有非常大的现实意义。

2 系统需求分析(System requirements analysis)

2.1 系统功能需求

作者单位现已有一套基于.Net架构的在线招聘报名管理系统, 该系统采用了技术较为成熟的三层架构, 可以满足报考人员通过电脑连接互联网实现在线报名的需求, 但是如果通过手机浏览器访问在线招聘报名系统的表示层WEB页面

时, 手机就会出现页面乱码、页面不完整、功能按钮点击失效等情况^[3]。因此为了能够实现让报考人员能够通过手机完成报名的需求, 就必须开发一套基于Android平台的招聘报名系统。为了不影响已有的在线招聘报名系统的继续使用, 以及保证招聘报名工作数据的统一性, 基于Android平台的招聘报名系统要通过公共数据访问模块远程调用已有的在线招聘报名系统的底层数据库, 也就是说两个不同操作系统的程序要共用一个数据库。本系统采用对WebService中API的重新编程来完成这个远程调用任务。

基于Android平台的招聘报名系统提供了报名端和管理端两个访问端口, 系统的数据支持由一个公共数据访问模块提供。报名端提供了完成所有报名流程的各个功能, 以及查看审核结果、报考进度等功能。为了让工作人员能够及时掌握报名工作情况、处理报名工作中出现的问题, 管理端为工作人员提供了查询、维护和管理报名数据信息的功能^[4]。

2.2 系统主要工作流程

本系统工作流程如图1所示。

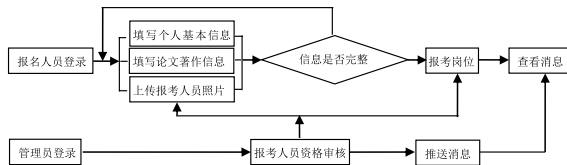


图1 系统主要工作流程图

Fig.1 Main work flowchart of the system

3 系统架构设计(Design of software architecture)

3.1 模块划分

在Android平台进行软件开发与Windows平台不同,在系统模块的划分中要充分考虑到Android平台的特性,以及手机硬件环境对程序运行的影响等问题^[5]。本系统的模块划分如图2所示。

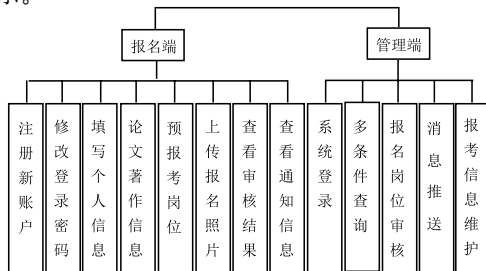


图2 模块划分图

Fig.2 System module partition

3.2 公共数据访问模块的架构设计

公共数据访问模块是系统的核心。本系统实现的难点是如何在与已有的在线招聘报名系统共用一个底层数据库,也就是如何处理两个不同的操作系统、不同的系统之间多用户并发访问延迟的问题。为了解决这些问题,本系统引入了WebService技术。WebService可以兼容任何一种操作系统,而且与实现编程语言无关^[6,7]。在Android系统的开发中,并不能直接对WebService进行编程,需要调用KSOAP2开发包来辅助完成。在WebService中提供了许多可以自定义的API接口,我们通过对API接口的编程就能够实现对多个用户的访问请求、远程数据库的响应进行协调与管理^[8,9]。系统数据架构设计模型如图3所示。



图3 数据架构图

Fig.3 Data structure diagram

4 核心功能模块的设计与实现(Design and implementation of core functional modules)

4.1 报名端数据录入的实现

报名端要完成报考人员的所有信息录入,在这些信息中科研论文与著作信息的输入是实现的难点,因为这些信息具有特殊符号多、字符长度长、数据记录多等特点,所以本系统在对这些数据进行录入的时候,首先会对这些数据进行了特殊符号校验、字符长度限制、数据记录分段等处理,并且还使用了哈希表来构造这些数据的结构形式,主要实现方法如下所示。

```
String ServiceUrl="http://ip:8080"
```

```
String webservicename=WebService;
SoapObject rq=new SoapObject(ServiceUrl,
webservicename);
HashMap<String, Object> ifmps=new HashMap<
String, Object>();
ifmps.put( "bk1" ,edit1);
ifmps.put( "bk2" ,edit2);
for ( i=0;i<ifmps.length;i++,){
ifmps.hasNext()
request.addProperty(ifmps.getKey().toString());}
```

4.2 管理端推送信息模块的实现

招聘管理人员可以通过管理端的信息推送功能,将报考人员的资格审核结果、或是在报考工作中需要发布的通知、公告等信息推送出去。管理端推送信息的主要实现方法如下所示。

```
URL ServiceUrl=new URL( "http://ip:8080" );
SoapObject webrpc=new SoapObject(webservicename);
webrpc.addProperty(webservicename, tx1);
String vf=result.getProperty("webservicetext").
toString();
```

```
Intent webnet=new Intent(this,ifActivity.class);
webnet.putExtra("message",vf);
```

4.3 上传报名信息文件的模块实现

报考人员通过手机上传报名信息照片到远端服务器中。为了保证上传的可靠性,系统对照片的体积大小做了相应的限制。主要实现方法如下所示。

```
URL WURL=new URL("http://ip:8080");
File imagefile=new File(imagetextpath);
StringBuffer stringbuffer=new StringBuffer();
InputStream inputstream;
try {inputstream=new FileInputStream(imagefile);
byte[] imagebuff=new byte[1024*1024];
int len=-1;
while (-1 !=(len=is.read(imagebuff,0, imagebuff.
length)))
{stringbuffer.append(Base64.encode(imagebuff,0,
len));}
is.close();}
```

4.4 系统实现

本系统报名端登录、填报个人信息和管理端消息推送运行效果如图4所示。



图4 系统效果图

Fig.4 System effect diagram

(下转第21页)