

**本科毕业设计（论文）**

题 目 学生安全报送平台

学院名称 计算机科学与技术学院

专业名称 网络工程

年级班级 13级（1）班

学生姓名 赵宝旗

指导教师 闫玺玺

2017年 6 月

# 摘要

该APP主要是通过学生的平时的日常表现及时由宿舍长以及班长上报该学生异常信息给对应管辖的辅导员，相关人员借此查看所涉及学生的异常信息，有利于宿舍长、班长和辅导员及时地对学生的特殊情况进行紧急处理。同时对数据的安全性采取了一定的加密，以此确保学生信息的安全。

本次主要利用Java高级编程语言，数据交互是通过http协议的post请求，数据接收是使用了第三方的xutils，数据解析使用gson。其所涉及的后台则使用tomcat做为本地服务器提供数据的支持服务器端使用的是mybatis这个持久层框架，spring业务逻辑层框架，数据库使用的是mysql数据库。

实现APP的功能不仅能够协助学校加强对学生的管理，也能够为家长及时了解学生情况提供了一个便利通道。

**关键词：**学生；android；APP；http协议；C/S结构

# ABSTRACT

The APP is mainly through the students' everyday performance timely by the dormitory and the students reported long monitor abnormal information to the corresponding jurisdiction of the counselors, abnormal information related personnel to check the students involved in the dormitory, is conducive to the long, monitor and counselors timely special cases of students for emergency treatment. At the same time, the security of the data has been encrypted to ensure the safety of student information.

The main use of the Java high-level programming language, data interaction is through the HTTP protocol post request, data reception is the use of third party xutils, data analysis using gson. The background is done using the Tomcat server to provide data support for the local server is using the mybatis persistence layer framework, business logic layer, spring framework, database using MySQL database.

Through the realization of the APP function, not only can help schools to strengthen the management of students, but also for parents to understand the students in a timely manner to provide a convenient channel.

**Key words:** students;Android; APP; HTTP protocol; C/S structure

目 录

1.引言 1

1.1课题背景 1

1.2目的和意义 1

1.3开发工具及技术 1

1.3.1开发工具 1

1.3.2 开发技术 2

2.需求分析 3

2.1可行性分析 3

2.1.1技术的可行性 3

2.1.2经济的可行性 3

2.1.3操作可行性 3

2.1.4法律的可行性 3

2.2系统用户用例图 3

2.2.1宿舍长用例图 3

2.2.2班长用例图 4

2.2.3辅导员用例图 4

2.2.4学工办用例图 5

2.3功能模块需求分析 6

2.4设计的基本思想 6

2.5性能需求 7

2.5.1系统的安全性 7

2.5.2数据的完整性 7

2.6界面需求 7

2.7软硬件需求 7

3.数据库分析与设计 9

3.1数据库的分析与设计 9

3.1.1数据库的概念结构设计 9

3.1.2数据库的逻辑结构设计 12

3.1.3数据库的连接原理 14

3.2中文乱码问题处理 16

4.系统功能实现 17

4.1客户端登录页面 17

4.2宿舍长模块 19

4.2.1异常管理 19

4.2.2宿舍学生信息管理 20

4.2.3个人信息管理 22

4.3班长模块 24

4.3.1异常管理 24

4.3.2班级学生信息管理 24

4.3.3个人信息管理 25

4.4辅导员功能模块 25

4.4.1异常管理 25

4.4.2专业学生管理 26

4.4.3专业班级管理 26

4.5学工办功能模块 26

4.5.1异常管理 26

4.5.2辅导员管理 26

4.5.3定时短信提醒管理 28

5.系统测试 31

5.1软件测试目的与意义 31

5.2测试过程 31

5.2.1登录模块测试 31

5.2.2添加学生模块测试 32

5.3 Android自动化测试 32

5.4测试中遇到的错误 33

6.结论 35

致谢 37

参考文献 39

# 1.引言

## **1.1课题背景**

随着学生年龄的增长，独生子女政策这些让现代的大学生的心理承受压力的能力大打折扣，许许多多校园安全事故发生在我们身边，大学不像初高中一样那样严格，在大学主要是自我约束，但是在我这几年的大学生活所观察到，大学生的自控能力和抗压能力都差的很远，各种因为小事跳楼，自杀等等，最重要的一个原因就是当有啥事学生不能有及时沟通，及时采取措施解决问题。一方面为了能够快速了解学生的异常情况及时给予帮助，另一方面为了更好的配合学校进行学校进行更好的管理，在这方面的综合考虑我选择了本毕业设计通过学生的生活表现及时由宿舍长或班长上报该学生异常信息给对应管辖的辅导员，上报后该生的宿舍长班长辅导员，学工办和副院长可以查看上报的学生信息，同时班长和辅导员可以对学生的异常信息进行处理，在一周规定的时间辅导员和学工办收到学生有异常但是没有处理的学生信息，短信的提醒时间是由副院长设置，数据都是经过加密以确保学生信息的安全，请求也是通过多层验证。

## **1.2目的和意义**

目的意义在于一方面是希望通过该软件能够及时反馈学生异常信息给老师，做到早发现早解决，以免因为一件小事因缺乏及时给老师学生沟通而酿成大错，另一方面是通过这个毕业设计提高自己的动手实践能力，增强自己的逻辑思维能力，将大学四年在大学学到的东西真正的用到实处。

## **1.3开发工具及技术**

### 1.3.1开发工具

学生安全报送平台使用到的开发工具有：编写android程序的IDE Android Studio，编写服务器程序的IDE IDEA，使用的持久层数据库是MYSQL，使用的服务器Tomcat等等[1]。

（1）Android Studio

在刚刚过去三年的I/O大会上，谷歌公司推出了一个新的Android开发环境Android Studio。经过不断的升级不断的优化，现在Android Studio已经有了1.8.1版本了。先不说AS公司是谷歌公司的亲儿子，哪里好哪里好，但是谷歌最终宣布将在当年的年底前中止对其他IDE开发环境的支持—开发者是时候正式向Eclipse说再见。所以Android studio是Android开发的必然选择[2]。

（2）IDEA

IDEA 全称 IntelliJ IDEA，是java语言开发的集成环境，IntelliJ被当代的程序开发人员共同评价为最好的java编译器，在代码的整合和代码的提示，以及编程的功能上都十分强大，有很多eclipse不具备的功能[5]。

（3）MYSQL

MySQL是一个小型的 SQL数据库管理系统，虽然MySQL数据库的源码没有对外开放，但是对于我们开发人员的使用来说是不受太多的影响的，虽然数据库没有像sqlserver等那些大型数据库这么庞大但是功能上并不因为小而功能减弱，现在好多企业级都是有的是mysql特别是它与Apache和PHP以及java结合，为建立基于数据库的动态网站提供了强大动力。

（4）Tomcat

Tomcat是Apache 软件基金会的一个Jakarta 项目中的一个核心项目，Tomcat是由多个公司共同开发而成。其中一家公司是Sun，在它的帮助下最新的j2ee规范总是能在Tomcat 中得到体现，目前最新版本是8.0。

### 1.3.2开发技术

（5）Mybatis

 Mybatis是一种持久层框架，它和hibernate拥有着很大的差别，hibernate框架人家都把增删改查的方法都已经写好了，你只需要配置表与表之间的关系就行了，编程人员不需要知道太多的sql语句就行了，灵活性不太高，但是mybatis灵活性就高很多增删改查完全有编程人员自己编写。

（6）Spring

Spring框架是一个桥梁他让控制层和持久层，是现在企业级开发基本上是比用的，同时也有程序员的春天之称，运用他的控制反转和IOC容器实现业务逻辑，通过使用集成的很多工具比如控制字符编码，拦截器，定时任务等[7]。

（7）Spring MVC

Spring MVC是一种控制层框架，它主要负责前台jsp页面的渲染，以及前台和后台的交互。

# 2.需求分析

## **2.1可行性分析**

现在的软件开发行业，软件开发大多都是来源于初始的想法，但是有好多人只重视想法而忽略了其他方面的调研，一个好的软件肯定是要进行很多方面的可行性分析的，不然光靠一腔热血，要不软件夭折在开发过程用，要不就是开发出来前景不好没有人用。

### 2.1.1技术的可行性

该学生安全报送平台是一款由原生的android编写的，高级编程语言Java，使用的页面布局是android里面最常用的四大布局，使用的本地数据库是mysql数据库，同时使用的本地服务器是tomcat，客户端向服务器发出的请求是http请求。在大二的上学期我们学过高级编程语言java同时在大二的下学期我们学了网络编程，在大三上学期学过web框架技术，在大三下学期学过mysql，同时课余时间跟着学院老师做项目学习过Android开发[3]。

### 2.1.2经济的可行性

该客户端使用的工具包括一台电脑配置不需要太高3000元的电脑就可以，剩下的也没有太多，至于编写软件，因为所用到的技术都是都是开源的不需要任何费用，使用的mysql也是一样，所以满足经济可行性。

### 2.1.3操作可行性

本客户端使用前台和后台双层校验，客户端方面使用输入输出数据格式以及非空条件的校验，后台是对请求数据的权限进行校验以防通过网络爬虫请求到数据，客户端方面添加了好多提示以至于避免用户刚开始使用APP的时候就出现各种疑问提示，增强了用户的体验[8]。

### 2.1.4法律的可行性

此学生安全报送平台由我个人自行研发，没有抄袭或仿照别人，不存在侵权行为，本软件的数据是通过加密的能够保护用户信息的安全，不存在泄露隐私行为，同时使用的软件和技术都是开元的因此不存在侵权行为所以在法律上是可行的。

## **2.2系统用户用例图**

### 2.2.1宿舍长用例图

宿舍长主要功能包括登录、查看本宿舍的学生的异常信息、以及异常的的处理情况、同时寝室长可以上报寝室学生的异常情况和处理本宿舍学生的异常信息，添加学生修改学生信息，以及修改密码，注销登录操作。寝室长用例图如图2-1所示。



图2-1 宿舍长用例图

### 2.2.2班长用例图

班长主要功能包括登录、修改密码、查看本班学生的异常信息，上报本班学生的异常信息、处理本班学生的异常信息为给宿舍长设置权限，班长用例图如图2-2所示。



图2-2 班长用户用例图

### 2.2.3辅导员用例图

辅导员有登录、修改密码、查看宿舍长和班长上传上来学生的异常信息可以查看影长的处理情况，处理学生的异常信息，添加学生，通过导入Excel表格批量添加学生，定时收到没有处理的异常学生的信息，辅导员用例图如图2-3所示。

图2-3 辅导员用户用例图

### 2.2.4学工办用例图

学公办有登录、修改密码、辅导员管理对辅导员信息的增删改查、学生异常信息查看，学生异常信息的处理情况的查看，设置短信发送时间，定期收到有异常未处理的学生的信息，以及注销登录切换账号的功能，还可以根据选择的时间查看学生在一段时间里学生的异常信息以及对学生的异常信息的处理的情况，以便及时对问题进行处理，学工办用例图如图2-4所示。



图2-4 学工办用户用例图

## **2.3功能模块需求分析**

该APP分为五种用户：宿舍长、班长、辅导员、学工办和副书记，拥有严格的上下级关系上级给下级分配权限：

（1）宿舍长主要是登录、查看该宿舍的学生的异常信息、异常的的处理情况、同时寝室长可以上报寝室学生的异常情况，以及修改密码。

（2）班长主要功能包括登录、修改密码、查看本班学生的异常信息，上报本班学生的异常信息、处理本班学生的异常信息为给宿舍长设置权限。

（3）辅导员有登录、修改密码、查看宿舍长和班长上传上来学生的异常信息可以查看影长的处理情况，处理学生的异常信息，添加学生，通过导入Excel表格批量添加学生，定时收到没有处理的异常学生的信息。

（4）学公办有登录、修改密码、辅导员管理、学生异常信息查看，学生异常信息的处理情况的查看。

（5）副书记主要是对学生信息的查看以及对异常信息查看，以及对处理的异常信息查看功能。

## **2.4设计的基本思想**

设计思想遵循以下几点：

（1）本APP是采用的C/S架构进行开发，在客户端主要注重前台页面的设计，使用四大布局确保页面美观。

（2）本APP的服务器端主要是使用mvc开发模式进行设计，以确保业务逻辑的严谨。

（3）本APP严格根据权限设置用户的功能，上级设置下级的权限使得APP更加完善更加井井有条。

（4）为了提高数据的查询的速度采用了数据库优化技术，以及各种集函数的使用。

（5）为了保证学生信息的安全该APP采用的是MD5对学生信息进行加密，以防信息被盗取。

## **2.5性能需求**

### 2.5.1系统的安全性

本安全报送平台在管理权限上要严格进行控制，具体要求如下：

（1）该APP分5种权限，每个权限都有自己规定的职责，为了保证信息的及时性和可靠性，以及学生的信息的安全性有很高的要求[9]。

（2）该APP对学生信息的安全性要求很高，用户使用客户端来请求数据后台会进行身份验证如果口令错误访问数据就会失败，同时我们保存学生数据的时候进行数据的加密，以确保信息安全。

### 2.5.2数据的完整性

（1）要求必填的选项会给出提示，不填就不让数据提交到后台。

（2）对于一个流程五个用户都需要参与，根据不同权限显示不同数据。

## **2.6界面需求**

本APP拥有很强的用户体验，页面设计十分友好，在客户端页面使用Java语言进行页面数据格式以及非空判断，请求数据的提示特别的多，确保用户能够根据提示很快的做出操作。

## **2.7软硬件需求**

硬件需求：

CPU：Pentium以上计算机

内存: 4G以上

软件需求：

操作系统版本： Win7/Win8/Win10

开发工具：Android Studio

开发语言：Java

手机内核：Android 4.0以上版本

# 3.数据库分析与设计

## **3.1数据库的分析与设计**

数据库分析与设计是在数据库设计及数据库应用系统技术，是信息管理系统分析和设计中的核心部分。要想一个系统设计的合理，功能强大那数据库设计这一块一定要设计的十分得当才行，不然到系统开发出来会发现到处存在bug[6]。

### 3.1.1数据库的概念结构设计

数据库概念结构设计通常都会使用E-R图来体现出表的属性以及表与表的关联，E-R图有明确的规定方框代表的是表名，椭圆形代表的是属性，表与表之间的关系使用菱形，表与表之间的关系分为一对一、多对多、一对多。

（1）学生实体实体图如图3-1所示。

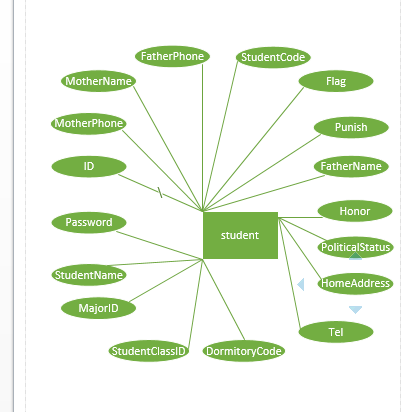


图3-1学生实体图

(2)辅导员实体实体图如图3-2所示。

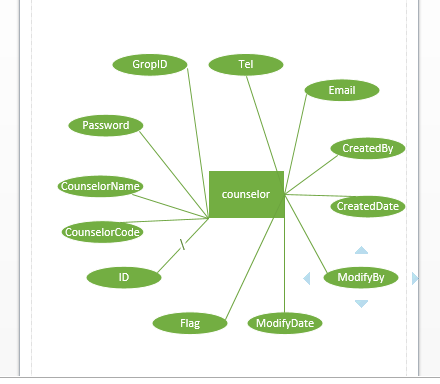


图3-2辅导员实体实体图

(3)学生异常信息实体图如图3-3所示。

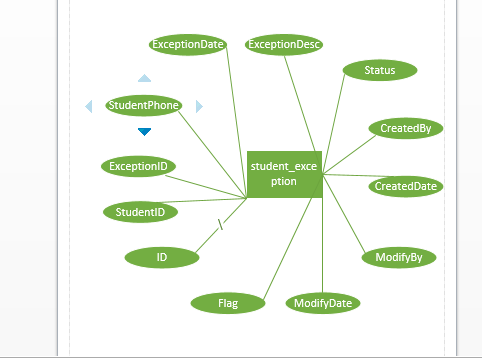


图3-3 学生异常信息实体图

(4)学生异常信息处理实体图如图3-4所示。

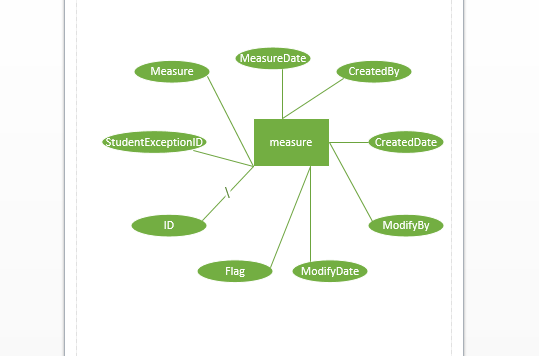


图3-4学生异常信息处理实体图

(5)辅导员专业表实体图如图3-5所示。

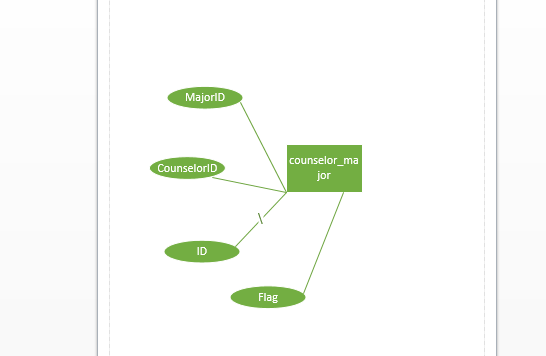


图3-5 辅导员实体实体图

(6)专业班级实体图如图3-6所示。

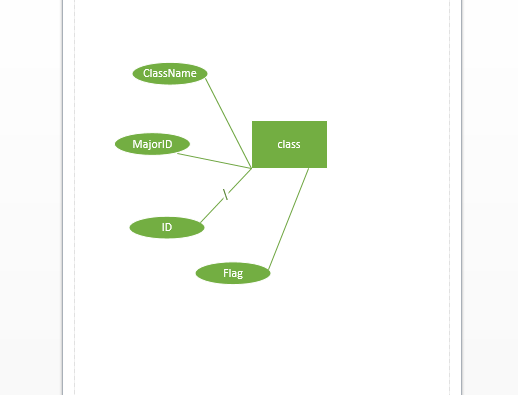


图3-6 专业班级实体实体图

### 3.1.2数据库的逻辑结构设计

逻辑结构是独立于任何一种数据模型的，在实际应用中，一般所用的数据库环境已经给定（如SQL Server或Oracle或MySql)。由于目前使用的数据库基本上都是关系数据库，因此首先需要将E-R图转换为关系模型，然后根据具体DBMS的特点和限制转换为特定的DBMS支持下的数据模型，最后进行优化。检查当前的E-R模型是否满足当前业务的场景。可以从某个实体开始，沿着当前E-R模型的各个节点去模拟业务场景。尤其需要和《需求规格说明书》去做校验。

到这里，也就完成了E-R模型建立的全过程，有时，对于比较复杂的E-R模型，一张图可能显得太过局促，可以建立全局、局部E-R模型图，以便于查看和分析。

（1）学生信息表主要是学生的基本信息，表结构如表3-1所示。

表3-1学生表(student)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 是否为空 | 是否主键 |
| ID | int | 50 | 否 | 是 |
| StudentCode | varchar | 50 | 否 | 否 |
| Password | varchar | 50 | 否 | 否 |
| StudentName | varchar | 20 | 否 | 否 |
| MajorID | int | 20 | 是 | 否 |
| StudentClassID | int | 20 | 是 | 否 |
| DormitoryCode | varchar | 20 | 是 | 否 |
| Tel | varchar | 20 | 是 | 否 |
| HomeAddress | varchar | 50 | 是 | 否 |
| Honor | varchar | 100 | 是 | 否 |
| PoliticalStatus | varchar | 20 | 是 | 否 |
| Punish | varchar | 20 | 是 | 否 |
| FatherName | varchar | 20 | 是 | 否 |
| FatherPhone | varchar | 20 | 是 | 否 |
| MotherName | varchar | 20 | 是 | 否 |
| MotherPhone | varchar | 20 | 是 | 否 |
| CreatedBy | varchar | 20 | 是 | 否 |
| CreateDate | timestamp | 0 | 是 | 否 |
| ModifyBy | varchar | 20 | 是 | 否 |

（2）辅导员表记录了辅导员的基本信息，表结构如表3-2所示。

表3-2辅导员表(counselor)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 是否为空 | 是否主键 |
| ID | int | 20 | 否 | 是 |
| CounselorCode | varchar | 20 | 否 | 否 |
| CounselorName | varchar | 20 | 否 | 否 |
| Password | varchar | 50 | 是 | 否 |
| GropID | int | 20 | 是 | 否 |
| Tel | varchar | 20 | 是 | 否 |
| Email | varchar | 20 | 是 | 否 |
| CreatedBy | varchar | 20 | 是 | 否 |
| CreatedDate | timestamp | 0 | 是 | 否 |
| ModifyBy | varchar | 20 | 是 | 否 |
| ModifyDate | timestamp | 0 | 是 | 否 |
| Flag | int | 10 | 是 | 否 |

（3）学生异常表，记录宿舍长班长上报的学生的异常信息，表结构如表3-3所示。

表3-3学生异常表(student\_exception)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 是否为空 | 是否主键 |
| ID | int | 50 | 否 | 是 |
| StudentID | int | 20 | 是 | 否 |
| ExceptionID | int | 20 | 是 | 否 |
| StudentPhone | varchar | 20 | 是 | 否 |
| ExceptionDate | timestamp | 0 | 是 | 否 |
| ExceptionDesc | varchar | 200 | 是 | 否 |
| Status | varchar | 20 | 是 | 否 |
| CreatedBy | varchar | 20 | 是 | 否 |
| CreatedDate | timestamp | 0 | 是 | 否 |
| ModifyBy | varchar | 20 | 是 | 否 |
| ModifyDate | timestamp | 0 | 是 | 否 |
| Flag | int | 10 | 是 | 否 |

（4）异常处理表，记录班长和辅导员对异常的处理情况，表结构如表3-4所示。

表3-4异常处理表表(measure)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 是否为空 | 是否主键 |
| ID | int | 20 | 否 | 是 |
| StudentExceptionID | int | 20 | 是 | 否 |
| Measure | varchar | 50 | 是 | 否 |
| MeasureDate | timestamp | 0 | 是 | 否 |
| CreatedBy | varchar | 20 | 是 | 否 |
| CreatedDate | timestamp | 0 | 是 | 否 |
| ModifyBy | varchar | 20 | 是 | 否 |
| ModifyDate | timestamp | 0 | 是 | 否 |
| Flag | int | 11 | 是 | 否 |

（5）异常的种类表，是学生异常属于哪一种，对异常进行分类，结构如表3-5所示。

表3-5异常类别表(exception)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 是否为空 | 是否主键 |
| ID | int | 20 | 否 | 是 |
| ExceptionName | varchar | 20 | 是 | 否 |
| CreatedBy | varchar | 20 | 是 | 否 |
| CreatedDate | timestamp | 0 | 是 | 否 |
| ModifyBy | varchar | 20 | 是 | 否 |
| ModifyDate | timestamp | 0 | 是 | 否 |
| Flag | int | 11 | 是 | 否 |

（6）用户权限表，对用户的账号进行授权，结构如表3-6所示。

表3-6异常类别表(exception)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 是否为空 | 是否主键 |
| ID | int | 20 | 是 | 否 |
| UserCode | varchar | 50 | 否 | 是 |
| GropID | int | 10 | 否 | 是 |
| Flag | int | 10 | 否 | 是 |

### 3.1.3数据库的连接原理

本客户端，在服务器上连接mysql是通过Druid数据库连接池，替换DBCP和C3P0。Druid提供了一个高效、功能强大、可扩展性好的数据库连接池。可以监控数据库访问性能，Druid内置提供了一个功能强大的StatFilter插件，能够详细统计SQL的执行性能，这对于线上分析数据库访问性能有帮助。数据库密码加密。直接把数据库密码写在配置文件中，这是不好的行为，容易导致安全问题。DruidDruiver和DruidDataSource都支持PasswordCallback。SQL执行日志，Druid提供了不同的LogFilter，能够支持Common-Logging、Log4j和JdkLog，你可以按需要选择相应的LogFilter，监控你应用的数据库访问情况。扩展JDBC，如果你要对JDBC层有编程的需求，可以通过Druid提供的Filter机制，很方便编写JDBC层的扩展插件。

连接数据库的代码如下：

<!-- 阿里 druid数据库连接池 -->

<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource"

destroy-method="close">

<!-- 数据库基本信息配置 -->

<property name="url" value="${url}" />

<property name="username" value="${username}" />

<property name="password" value="${password}" />

<property name="driverClassName" value="${driverClassName}" />

<property name="filters" value="${filters}" />

<!-- 最大并发连接数 -->

<property name="maxActive" value="${maxActive}" />

<!-- 初始化连接数量 -->

<property name="initialSize" value="${initialSize}" />

<!-- 配置获取连接等待超时的时间 -->

<property name="maxWait" value="${maxWait}" />

<!-- 最小空闲连接数 -->

<property name="minIdle" value="${minIdle}" />

<!-- 配置间隔多久才进行一次检测，检测需要关闭的空闲连接，单位是毫秒 -->

<property name="timeBetweenEvictionRunsMillis" value="${timeBetweenEvictionRunsMillis}" />

<!-- 配置一个连接在池中最小生存的时间，单位是毫秒 -->

<property name="minEvictableIdleTimeMillis" value="${minEvictableIdleTimeMillis}" />

<property name="validationQuery" value="${validationQuery}" />

<property name="testWhileIdle" value="${testWhileIdle}" />

<property name="testOnBorrow" value="${testOnBorrow}" />

<property name="testOnReturn" value="${testOnReturn}" />

<property name="maxOpenPreparedStatements" value="${maxOpenPreparedStatements}" />

<!-- 打开removeAbandoned功能 -->

<property name="removeAbandoned" value="${removeAbandoned}" />

<!-- 1800秒，也就是30分钟 -->

<property name="removeAbandonedTimeout" value="${removeAbandonedTimeout}" />

<!-- 关闭abanded连接时输出错误日志 -->

<property name="logAbandoned" value="${logAbandoned}" />

</bean>

## **3.2中文乱码问题处理**

在程序的测试阶段经常会出现这种问题就是请求数据到服务器端会出现乱码，之前通多好多手段但是都没有很好的解决这个问题，然后就询问上一届的学长，然后学长给出解决方案就是拦截所有的请求路径就请求数据转成utf-8格式这样就没有乱码的问题。在web.xml配置spring自己带的解决中文乱码问题的方法。

在web.xml中配置：

<!--中文过滤器-->

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>utf-8</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

# 4.系统功能实现

本客户端分为5种用户包括：宿舍长、班长、辅导员、学工办、副书记，宿舍长主要有查看该寝室的异常信息，上报异常处理异常，添加学生、查看学生详细信息、修改学生信息、删除学生、批量删除学生，班长主要是查看班级学生异常信息，上报学生异常修改密码查看个人信息，管理班级学生，辅导员主要是管理学生，管理班长查看学生异常处理异常，定时收到短信查看没有处理的学生异常信息。学工办查看学生异常信息，定时收到短信查看没有处理的学生异常信息。

## **4.1客户端登录页面**

（1）描述：客户端的登录一个是为了信息的安全，另一个就是为了获取登录人的权限来显示相关操作，以至于上下级关系仅仅有条。

（2）客户端登录页面效果图如图4-1所示：



图4-1 客户端登录页面设计

（3）核心代码。

Student student=userLoginService.getStudentByUserCode(userCode);

if (student != null){

if (StringUtils.isNotBlank(student.getSessionCode())){

//更新sessioncode

String sessionCode=CreateSessionCode.getSessionCode();

Session session = new Session();

session.setSessionCode(sessionCode);

session.setUserCode(student.getStudentCode());

userLoginService.updateUserSession(session);

}else {

//添加sessioncode

String sessionCode=CreateSessionCode.getSessionCode();

Session session = new Session();

session.setSessionCode(sessionCode);

session.setUserCode(student.getStudentCode());

userLoginService.addUserSession(session);

}

if (password.equals(student.getPassword())){

Student s=userLoginService.getStudentByUserCode(userCode);

resuletMap.put("code","200");

resuletMap.put("message","登录成功");

data.put("gropID",gropID);

data.put("userMessage",s);

resuletMap.put("data",data);

}else {

resuletMap.put("code","300");

resuletMap.put("message","密码错误");

}

}else {

resuletMap.put("code","300");

resuletMap.put("message","登录用户名不存在");

}

break;

因为该系统分为五种权限，宿舍长、班长、辅导员、学工办、副书记，每个人的操作都不同下面详细对每个权限分开说明。

## **4.2宿舍长功能管理**

### 4.2.1异常管理

（1）描述：第一张图是宿舍长查看本宿舍人的学生的异常情况，第二张图是宿舍长上传该寝室学生的异常信息，第三张是宿舍长查看学生异常处理情况，第四张是宿舍长对异常的处理。

（2）程序效果图如图4-2所示。

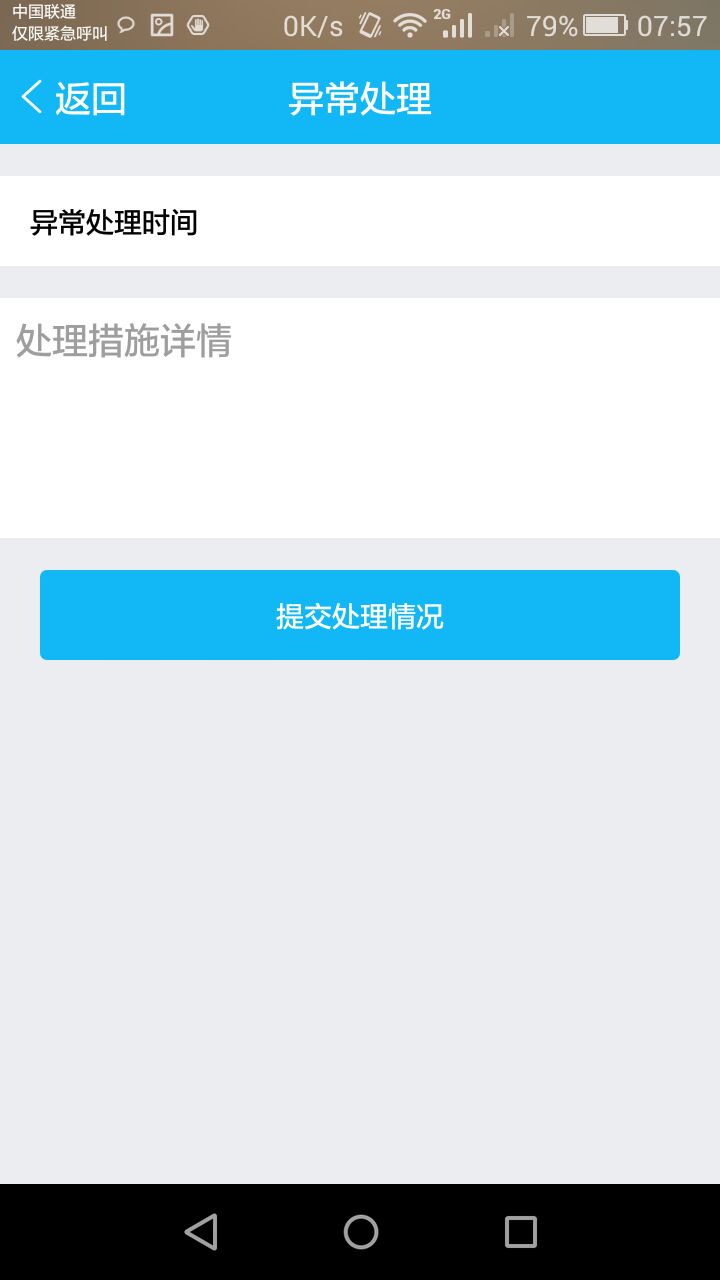
 

图4-2 异常管理

（3）核心代码。

@RequestMapping(value = "/getAllException.do")

@ResponseBody

public Map getAllException(HttpServletRequest request){

String sessionCode=request.getHeader("sessionCode");

Map<String,Object> resultMap = new HashMap<String,Object>();

int i=userLoginService.getSessionCodeCount(sessionCode);

if (i >0){

List<Exception> exceptions=exceptionService.getAllExceptions();

Map<String,Object> data = new HashMap<String,Object>();

resultMap.put("code","200");

resultMap.put("message","请求数据成功");

data.put("exceptionData",exceptions);

resultMap.put("data",data);

}else {

resultMap.put("code","500");

resultMap.put("message","你的账号可能在别处登录，请重新登录");

}

return resultMap;

}

### 4.2.2宿舍学生信息管理

（1）描述：宿舍长可以查看本人宿舍下的所有的学生的信息，以及学生的详细的信息，可以对学生的信息进行增删改查操作，图一查看本寝室学生列表，图二有对学生信息的相关操作，图三查看学生详细信息，图四修改学生信息。

（2）程序效果图如下图4-3所示。

图4-3学生信息管理

（3）核心代码：

//修改学生信息，并设置学生权限

@RequestMapping(value = "/updateStudentMessage.do",method = RequestMethod.POST)

@ResponseBody

public Map updateStudentMessage(String student,Integer gropID){

Map<String,Object> resultMap = new HashMap<String,Object>();

try {

Gson gson = new Gson();

Student s = gson.fromJson(student,Student.class);

studentManagerService.updateStudentByStudentCode(s);

if (gropID != null){

//判断该用户角色是否存在

Integer g=studentManagerService.getGropIDByUserCode(s.getStudentCode());

if (g!= null){

//修改

Map<String,Object> map = new HashMap<String,Object>();

map.put("userCode",s.getStudentCode());

map.put("gropID",gropID);

studentManagerService.updateGropIDByUserCode(map);

}else {

//给用户分配角色

Map<String,Object> map = new HashMap<String,Object>();

map.put("userCode",s.getStudentCode());

map.put("gropID",gropID);

studentManagerService.addUserGrop(map);

}

}

resultMap.put("code","200");

resultMap.put("message","修改用户信息成功");

} catch (Exception e) {

e.printStackTrace();

resultMap.put("code","300");

resultMap.put("message","修改用户信息失败");

}

return resultMap;

}

### 4.2.3个人信息管理

（1）描述：图一查看登录人的基本信息，图二是修改个人账户的密码。

（2）程序效果图如下图4-4所示

图4-4 修改密码

（3）核心代码：

case 1:

case 2:

Student student = new Student();

student.setStudentCode(userCode);

student.setPassword(password);

try {

userSettingService.updateStudentPasswordByUserCode(student);

resultMap.put("code","200");

resultMap.put("message","修改密码成功");

} catch (Exception e) {

resultMap.put("code","300");

resultMap.put("message","修改密码失败"+e.getMessage());

}

break;

case 3:

//修改辅导员的密码

Counselor counselor = new Counselor();

counselor.setCounselorCode(userCode);

counselor.setPassword(password);

try {

userSettingService.updateCounselorPasswordByUserCode(counselor);

resultMap.put("code","200");

resultMap.put("message","修改密码成功");

} catch (Exception e) {

resultMap.put("code","300");

resultMap.put("message","修改密码失败"+e.getMessage());

}

break;

case 4:

//创建一个学工办对象

SEO seo = new SEO();

seo.setsEOCode(userCode);

//设置新的密码

seo.setPassword(password);

try {

userSettingService.updateSEOPasswordByUserCode(seo);

//返回不同的状态值给客户端

resultMap.put("code","200");

resultMap.put("message","修改密码成功");

} catch (Exception e) {

resultMap.put("code","300");

resultMap.put("message","

修改密码失败"+e.getMessage());

}

break;

## **4.3班长功能管理**

### 4.3.1异常管理

班长的异常管理和宿舍长的异常管理十分相似，只是查看的数据不同宿舍长只能查看和管理本宿舍人的学生的异常的的信息而班长可以查看和管理本班的学生的异常的信息，班长可以处理全班学生的异常信息，宿舍长只能处理本宿舍人的学生的信息，班长可以对本班学生进行信息的增删改查，对学生进行授权但是授权只能是对学生进行受学生权限和宿舍长权限，班长也可以修改个人密码，切换账号，请参考宿舍长里面的异常管理。

### 4.3.2班级学生信息管理

（1）描述：班长和寝室长对学生的管理有两方面的区别是一方面就是查看学生的数量一个是只能查看本寝室几个人的学生异常信息，以及处理陈宿舍几个人学生的异常信息，班长可以查看本班学生的异常信息以及上报本班所有学生的异常的信息，以及查看本班所有学生异常的信息的处理情况，如图一还有一个就是修改的时候班长可以修改学生信息权限为班长一下权限，而宿舍长不能修改学生的权限。

（2）程序效果图如图4-5所示。

图4-5学生信息管理

3.核心代码：

<update id="updateGropIDByUserCode" parameterType="map">

UPDATE user\_grop SET GropID=#{gropID} WHERE UserCode=#{userCode}

</update>

### 4.3.3个人信息管理

（1）描述：班长的个人信息的管理和宿舍长的个人信息的管理相同，都包括查看个人信息，修改密码，切换账号等功能。

## **4.4辅导员功能管理**

### 4.4.1异常管理

（1）描述：辅导员的异常管理和宿舍长班长很相似都可以查看学生的异常信息，并且对学生的异常信息进行处理，但是最不同的是辅导员定期能够收到系统自动发送的短信信息，该信息主要是没有处理的学生的异常信息。

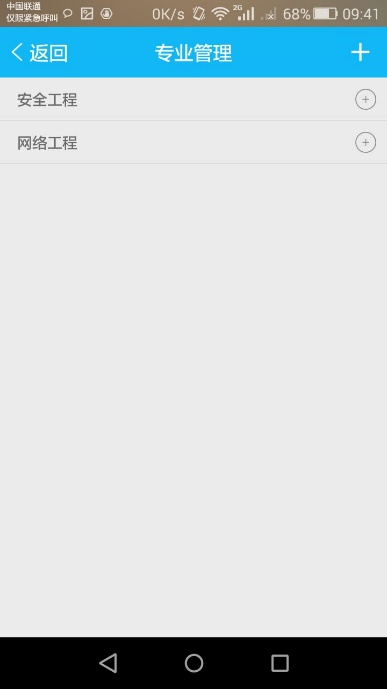
### 4.4.2专业学生管理

（1）描述：由于辅导员一个人可能要管理多个专业，相应的要管理的学生很多，因此辅导员学生管理除了比班长和宿舍长查看的学生信息多以外，还有就是多了一个批量导入学生的功能，通过导入excel表格数据到数据库批量新增学生信息。

### 4.4.3专业班级管理

（1）描述：辅导员拥有一个功能就是可以添加自己管辖的专业班级可以对专业班级进行管理宠儿管理专业班级下面的学生。

（2）如图4-6所示。

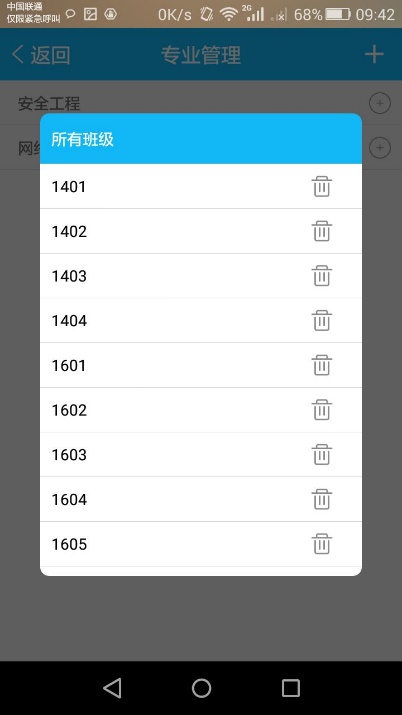
 

图4-6 辅导员专业班级管理

## **4.5学工办功能管理**

### 4.5.1异常管理

（1）描述：学工办关于异常描述和辅导员大体上相同处理比辅导员查看的异常信息多以外唯一不同的就是学工办不会对学生异常信息进行处理操作。

### 4.5.2辅导员管理

（1）描述：学工办关于异常描述和辅导员大体上相同处理比辅导员查看的异常信息多以外唯一不同的就是学工办不会对学生异常信息进行处理操作。

（2）如图4-7所示。

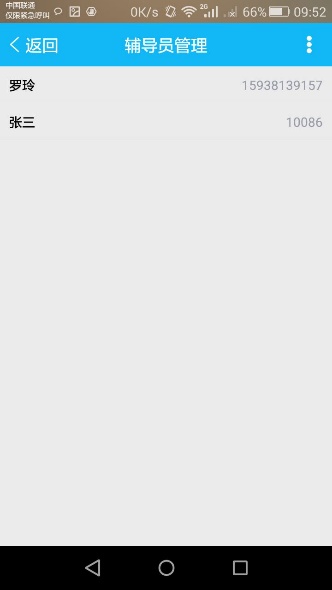
  

图4-7 辅导员管理

（3）核心代码

//添加辅导员

@RequestMapping(value = "/addCounselor.do",method = RequestMethod.POST)

@ResponseBody

public Map addCounselor(String counselor,Integer gropID){

Map<String,Object> resultMap = new HashMap<String,Object>();

try {

Gson gson = new Gson();

Counselor c = gson.fromJson(counselor,Counselor.class);

//获取该学生的学号看是否存在

Counselor co=studentManagerService.getCounselorByCode(c.getCounselorCode());

if (co==null){

studentManagerService.addCounselor(c);

resultMap.put("code","200");

resultMap.put("message","添加辅导员成功");

if (gropID !=null){

//判断该用户角色是否存在

Integer g=studentManagerService.getGropIDByUserCode(c.getCounselorCode());

if (g!= null){

//修改

Map<String,Object> map = new HashMap<String,Object>();

map.put("userCode",c.getCounselorCode());

map.put("gropID",gropID);

studentManagerService.updateGropIDByUserCode(map);

}else {

//给用户分配角色

Map<String, Object> map = new HashMap<String, Object>();

map.put("userCode", c.getCounselorCode());

map.put("gropID", gropID);

studentManagerService.addUserGrop(map);

}

}

}else {

//返回300状态码标识辅导员已经存在

resultMap.put("code","300");

resultMap.put("message","辅导员已经存在");

}

} catch (Exception e) {

e.printStackTrace();

resultMap.put("code","300");

resultMap.put("message","添加学生失败");

}

//将信息返回给客户端

return resultMap;

}

### 4.5.3定时短信提醒管理

(1)描述：学工办可以为系统设置时间来自动发送短信给辅导员学工办关于学生异常没有处理学生信息。

(2)如图4-8所示。



图4-8 设置短信发送时间

3.核心代码：

public static void send(Map map) {

HttpClient client = new HttpClient();

PostMethod method = new PostMethod(Url);

client.getParams().setContentCharset("GBK");

method.setRequestHeader("ContentType","application/x-www-form-urlencoded;charset=GBK");

int mobile\_code = (int)((Math.random()\*9+1)\*100000);

String content = new String("您的验证码是：" + mobile\_code + "。请不要把验证码泄露给其他人");

NameValuePair[] data = {//提交短信

new NameValuePair("account", "C91579494"), //查看用户名请登录用户中心->验证码、通知短信->帐户及签名设置->APIID

new NameValuePair("password", "0a18a7bf6a511c95362bfa5c8d03c9d6"), //查看密码请登录用户中心->验证码、通知短信->帐户及签名设置->APIKEY

//new NameValuePair("password", util.StringUtil.MD5Encode("密码")),

new NameValuePair("mobile", "18336821056"),

new NameValuePair("content", content),

};

method.setRequestBody(data);

try {

client.executeMethod(method);

String SubmitResult =method.getResponseBodyAsString();

//System.out.println(SubmitResult);

Document doc = DocumentHelper.parseText(SubmitResult);

Element root = doc.getRootElement();

String code = root.elementText("code");

String msg = root.elementText("msg");

String smsid = root.elementText("smsid");

System.out.println(code);

System.out.println(msg);

System.out.println(smsid);

if("2".equals(code)){

System.out.println("短信提交成功");

}

} catch (HttpException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (DocumentException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

# 5.系统测试

## **5.1****软件测试目的与意义**

软件测试的目的是在真实系统工作环境下通过与系统的需求定义作比较，检验完整的软件配置项能否和系统正确连接，发现软件与系统/子系统设计文档和软件开发合同规定不符合或与之矛盾的地方。系统测试是将通过确认测试的软件，作为整个基于计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合起来，在实际运行（使用）环境下，对计算机系统进行的测试。是为了发现缺陷并度量产品质量，按照系统的功能和性能需求进行的测试。而且，系统测试还要检验系统的文档等是否完整、有效。另外，系统测试的测试用例应根据需求分析说明书来设计，并在实际使用环境下来运行。最后，系统测试一般使用黑盒测试技术，并由独立的测试人员完成。

对于软件工作而言，系统测试是软件研制人员参加系统的综合测试，软件及计算机系统加入到系统中进行测试。应该一方面为系统测试提供必要的软、硬件及资料支持，另一方面从软件测试角度提出系统测试中关于软件的测试设计。

## **5.2测试过程**

在测试之初，由于将数据库中的数据是随便输入，没有实用性，为了使系统更加完善，在系统测试的过程中，本系统将一开始编程随便输入的数据全部清理后，输入了一些有效的测试数据，这样可以更真实的反映系统的功能实现情况。

这样不断发现问题，经过反复的测试、调试，把问题一个个的解决，最终系统可以正常运行。

### 5.2.1登录模块测试

测试流程：

（1）使用手机打开客户端，输入错误的账号和密码

（2）登录

（3）输入正确的账号密码信息

（4）登录

（5）测试结果

测试表格如表5-1所示。

表 5-1 客户端登录测试表格

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块名称 | 测试用例 | 预期结果 | 实际结果 | 是否合格 |
| 客户端登录 | 用户号：null 密码：null | 请输入用户号 | 请填写完整 | 合格 |
| 客户端登录 | 用户号：02  密码：null | 请输入密码 | 请填写完整 | 合格 |
| 客户端登录 | 用户名：02  密码：2018 | 用户名或者密码错误 | 密码错误 | 合格 |
| 客户端登录 | 用户名：02  密码：2017 | 登录成功到客户端首页 | 登录成功到客户端首页 | 合格 |

### 5.2.2添加学生模块测试

测试流程：

（1）使用辅导员账号，点击学生管理

（2）添加学生信息

（3）输入学生信息

（4）添加

（5）测试结果

测试表格如表5-2所示。

表 5-2客户端登录测试表格

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模块名称 | 测试用例 | 预期结果 | 实际结果 | 是否合格 |
| 添加学生 | 学号：null | 请输入学号 | 请填写完整 | 合格 |
| 添加学生 | 用户号：02 | 学生已经存在 | 学号已存在 | 合格 |
| 添加学生 | 用户名：\*\*b | 学号不合法 | 输入不进去 | 合格 |
| 添加学生 | 用户名：003 | 添加成功 | 添加成功 | 合格 |

## **5.3.Android自动化测试**

如今手机客户端的开发行业已经非常的流行，开发者也越来越多，对于android客户端的测试这一模块，很多公司还大部分使用测试人员手动进行软件测试。 由于手机客户端的开发，要适应不同的android内核，就比如一个android客户端要适应不同的android版本，不仅旧版本要支持最新的版本，还要支持比现有版本配置低的版本， 这就要求手动测试人员要使用很多部不同的Android系统下进行android 手机来手动测试。 要在很多部手机上进行测试毫无疑问测试的效率是非常低且会消耗大量的人力和物力。Android客户端除了在手机上测试还有可能要用到其他设备上，原本一个负责客户端测试的的测试人员应用的测试人员需要在不同的机器上测试，如果再加上开发人员不停的更新代码以及Android客户端的版本，我可以肯定测试人员要疯了。在这些情况的分析下，使用手机自动化测试不仅仅能解决上面的所有问题，使用手机测试自动化，每次版本更新只需要重新运行一下case, 然后添加一些新的case，对于不同的客户端版本，也不需要重新写case。 可以说一条自动化测试案例，解决了这些问题。 那么问题来了，手机自动化测试，哪家强？系统性能测试相对应的测试的原则是通过模拟生产运行的业务压力量以及使用系统的场景相结合，测试整个系统的性能是否满足特定用户需要的性能需求。也就是说，这种方法就是要在特定的运行条件下验证系统的能力状态[4]。

手机测试工具如下：

（1）Robotium Android Testing tool

Robotium是一个免费的android测试工具，它是由Java提供了一个第三方类库给测试人员调用， 它仅仅支持Java语言，所以它基本上用exclipse工具来开发代码。

写好的测试脚本可以在支持android客户端的不同版本测试。 因为Robotim测试工具是基于开发源码的基础上写的测试脚本，所以不能和application进行互动。

（2）UI Automator only for android 测试

这个测试工具是谷歌公司官方推出的的测试工具，对于测试android客户端,是一个不错的工具。它支持android 4.1以上的版本，Android客户端测试人员用这个工具编写的测试脚本，这个客户端可以支持很多不同的Android客户端平台，可以操作负责的用户流程。它能和TestNG集成，测试的方面比较多也就是说测试的东西比较全面，它可以客户端的UI元素。

（3）Appium Android 自动化框架

它是ios平台, android平台自动化测试的一个框架，支持多种高级编程语言: Java, C#, Ruby. 它可以调用selenium的API，所以可以利用selenium来测试在手机上面的网页应用，之前学过关于appium测试的相关知识,它支持云计算方面的设备它可以为软件测试人员提供许许多多虚拟手机进行测试，可以为客户提供许多虚拟手机来测试，但是这些测试都是付钱的，这块不过是付费的。所以你想要测试的越多，那你所付出的费用就越多。

（4）MonkeyRunner

下面主要介绍这个测试工具，这个工具目前是客户端测试行业一个比较流行的测试工具，但是它支持脚本语言比较单一比较拥有局限性，仅仅是python高级编程语言,因为因为有这个方面的特点导致不能像不要像Robotium一样处理源代码，它可以利用Record功能来录制测试脚本，虽然脚本是使用Python编写的但是她的执行是通过jython来解析执行，jython是Python的Java的实现，通过将代码解析到Java虚拟机的字节码进行执行，Python集成Java类型可以任意调用Java里面的api[10]。

## **5.4测试中遇到的错误**

在为客户端进行测试过程当中遇到的错误有很多，例如：

(1)辅导员批量导入学生数据的时候，如果存在两个学号相同的他也会再次添加到该系统里面导致查询或者登录的时候出现太多结果异常，导致整个软件崩溃，最终修改是当添加学生和或者修改学生信息的时候先根据学号查询如果存在就不执行操作。

(2)就是给用户授权的时候如果之前用户已经拥有一个权限，再给他授权的时候会出现一个用户拥有多个权限的问题，做的修改是根据用户号查询学生用户权限如果存在的话就做修改操作不存在的话做添加操作。

(3)定时发送短信功能，因为测试的短信的条数已经发送完毕，导致现在发送关心发送不出去，我充值了点钱现在可以发送短信提醒了。

(4)客户端提交学生信息会出现乱码问题，是因为客户端发送数据时没有设置发送格式，修改发送数据形式为utf-8就没有出现这种情况。

# 6.结论

通过对该客户端的开发，可以说我把所有的大学学到的关于编程方面的知识都用上了，我们专业在大二的时候学习了一年的就java，让我从一开始就对java语言产生浓厚的兴趣，在一个偶然的机会我认识认识我们学校的在软件方面特别有名气的老师，因为他们想在假期培养一些优秀的编程人才，我就踊跃的报名，并且通过了面试，成功成为ACM应用程序部的一员，在假期里学习java语言、Javaweb开发、数据库、Android开发等等好多东西。在一年的自学我结束了我的大三上学期的学习，感觉自己在编程方面提升了很多学到了好多知识，在大三的下学期我结识了另一位恩师，上一位是带我走向入门，而这以为让我提高自己，开始不再自己自学而是在真正的项目里面学习。在做这个学生安全报送平台的开发的过程当中，我运用了我学的大部分的知识，从Android开发到JavaWeb开发，到数据库设计，以及整个系统的规划设计，让我学习到了很多不仅把理论和实践结合在一起，而且大大增强了个人的编程思想，体格了个人的动手的能力。

# 致谢

光阴似箭，转眼间我们就来到了大四，我的大学四年收获了很多也失去了很多，大大学四年的生活我遇到了好多好多帮助我的老师，

首先要感谢我的指导老师闫玺玺老师，从一开始作为我的指导老师就为我们的毕设的事情操不完的心，帮我修改论文，查看毕业设计的不足之处，闫玺玺老师学识渊博，治学严谨，平易近人，给予我精心的指导和帮助，为我们付出辛勤的劳动，花费了大量的时间和精力，在这说声老师辛苦了。

在大二我认识了我们学校ACM协会的指导老师王建芳老师，在他那里开启了我的编程的路，学到了好多android和网络编程方面的知识，

在大三的一学年我认识到另一位恩师任建吉老师，在一个偶然的机遇下认识了任老师，总是感觉和他在一起时朋友而不是师生关系，在哪老师带着我们做项目，这个机会不是每个学生都拥有的，实战和知己自学差别还是挺大的，在哪一天的学习超过我自学一星期，现在我找的工作还是老师介绍的。

感谢大学期间一直帮助我的家人、老师、同学，你们的帮助让我一步一步完善了自己一步一步让自己走向成功成才，我的成功成才离不开你们的的帮助，忠诚的谢谢帮助我的每一个老师同学。

# 参考文献

1. 孙剑. Android 系统上应用程序按需加载机制的设计与实现[M]. 北京大学，2011:99-110.
2. 卢娜. 基于 Android 平台的手机桌面资讯系统的设计与实现[M]. 西安电子科技大学，2011:290-300.赵 辉，姚胤含．白领就业指南：Java Web开发设计师之路［Ｍ］．北京：电子工业出版社，2011.9．
3. 杜吉志，徐明昆. Android 系统内存管理研究及优化[J]. 软件，2012,24(5)：69-80.
4. 柯元旦. Android 内核剖析[M]. 北京：电子工业出版社，2011:59-70.
5. (美)Bronson．Java程学设计基础．北京大学出版社[M]，2005.1．
6. 伍俊良. 管理信息系统(MIS) 开发与应用. 北京: 科学出版社.
7. 陈雄华 Spring 企业级应用开发详解[M] 北京：电子工业出版社 2009赛奎春．JSP信息系统开发实例精选[M]．北京：机械工业出版社，2012.9．
8. Carmela Troncoso.Improving Secure Long-Term Archival of Digitally Signed Documents[M].Proceedings of the4th ACM international workshop on Storage secu-rity and survivability，2008：102-134.．
9. Aleksej Jerman Blai.Long-term trustedpreservation service using service interaction proto-col and evidence records[J].Computers and Standards，2007：23-29..
10. Watts D J.Small Worlds. The dynamics of networks between order and randomness[P].America: 56A2579BB6,l999-05-13.
11. Tsui，Frank F. JSP EM DASH A RESEARCH SIGNAL PROCESSOR IN JOSEPHSON TECHNOLOGY[C]. IBM Journal of Research and Development，Vol24， No2,1980：243-252.
12. Michael Gehond. Logic Programming and Nonmonotonic Reasoning [M]德国.Springer Berlin/Heidelberg. 2007