**嘉应学院**

**毕业设计论文**

**题目: 基于Andriod系统的学生考勤系统设计**

**指导老师： 杨久红**

**学院： 电子信息工程学院**

**专业： 电子信息工程**

**班级： 101**

**学号： 101100049**

**姓名： 刘兴**

**2014年2月17日**

**摘要**

高校中由于教学模式的原因，师生之间交流的时间极为短暂，而考勤占用了很多宝贵的教学时间，往往会影响正常的教学计划和教学质量，同样也会给学生的学习生活带来不必要的麻烦。简化教学工作，提高学生学习效率已经成为高校中亟待解决的问题。近些年移动互联网的不断壮大和移动平台的飞速发展为教学模式的改善提供了新的渠道和手段。利用移动平台中应用最广泛的Andriod系统，通过扫描学生证的条形码进行考勤，在手机移动端上搭建的便捷高效的考勤方式，方便老师进行教学考勤工作，也为学生提供查询课程签到的方式，优化了教学模式，增加了教学时间，也提高了学生的学习效率，并通过搭建移动端考勤系统，使得客户端扩大了传统考勤系统应用范围，降低了老师考勤的时间成本，极大改善了教学质量，让学生有更多的时间在课堂中进行学习。

**关键词**：Android、考勤、条形码扫描

**Abstract**

In colleges and universities，the communication between teachers and undergraduates is scrace, due to the reson of existing teaching mode.And the attendance checking system takes so much precious didactical hour, it usually takes a negative impact on both the quality of education and didactical schedule, moreover,it takes unnecessary troubles to undergraduates’ school life.How to simplify the work of teaching,and promote the undergraduats’learning efficiency has became a urgent issue.In recent years,the expending of mobile internet and the development of mobile platform provide new ways and means to improve the teaching mode.By making the most of the widely used Andriod system in mobile platform,and scaning bar code on Student ID Card,to set up a mobile terminal platform on which we have a convenient and efficient checking attendance system,that makes it easier for teachers to check attendance and the students can also have the approach to search their curriculum and attendance information.It optimizes the teaching mode,increases the didactical hour and encourages the learning efficiency. Meanwhile by setting up the mobile terminal platform of the work attendance checking system,it will extend the scope of traditional attendance system on client-side,decrease the time costs of teachers,perfect the quality of teaching.It helps undergraduats have more time to spend at class. **Keywords:** Android, Attendance, Barcode Scanning

**目录**

[1 前言 0](#_Toc21189)

[1.1选题背景 0](#_Toc3644)

[1.2可行性分析 0](#_Toc21553)

[2 需求分析 1](#_Toc13686)

[2.1需求分析目标及任务 1](#_Toc25514)

[2.2需求分析组织形式 1](#_Toc24757)

[2.3需求分析信息整理 1](#_Toc6549)

[2.4研发叙述说明 2](#_Toc18978)

[2.5研发标准与规范 2](#_Toc9571)

[3 概要设计 3](#_Toc4500)

[3.1系统功能 3](#_Toc17687)

[3.3系统主要流程 4](#_Toc11391)

[4 数据库设计 6](#_Toc29154)

[4.1 数据库Mysql简介 6](#_Toc27287)

[4.2 数据库设计 6](#_Toc13005)

[5.1考勤记录创建模块实现 10](#_Toc23643)

[5.2条形码扫描模块实现 10](#_Toc19851)

[5.3考勤数据模块的实现 12](#_Toc4898)

[5.4班级考勤数据模块的实现 14](#_Toc3803)

[6 系统的测试运行与维护 15](#_Toc11871)

[6.1系统的测试 15](#_Toc28472)

[6.2系统的维护 15](#_Toc2101)

[7 总结 17](#_Toc15765)

[附录 18](#_Toc9116)

[参考文献 27](#_Toc2432)

[致谢 28](#_Toc20636)

**1 前言**

**1.1选题背景**

随着科技的发展，移动通讯平台在不断地发展中愈来愈加成熟，安卓系统以其开放的平台，众多、免费的应用，华丽、新鲜的运行界面，简单、亲切的用户体验而备受消费者的喜爱。

在高校中，教师对于签到，很多时候都只能通过书面去记录，却不能方便、快捷、实时得汇总至教务秘书处。基于Android系统的教学管理系统则是一款建立在安卓平台的软件，致力于为广大师生工作与学习提供便利，系统突破传统交流的时空限制，实现师生之间的方便有效地交流。

基于Android系统的教学管理系统分为两个部分，一部分为教师客户端，为教师在对学生的签到方面上便利。另一部分为web服务端，教务秘书可以通过此客户端查看课程签到状况以及关于班级的一些考勤报表。

本论文将对学生客户端，包括安卓客户端与Web服务端的实现进行详细的阐述，实现系统的B/S模式。

**1.2可行性分析**

近些年，移动设备的高速发展，使得手机等移动设备高速智能化，移动网络通信正在以前所未有的惊人速度走入人们的生活、工作和学习中，提高了工作学习的效率，增加了获取信息的渠道，突破了相互交流的时空界限。考勤系统可以基于移动平台安卓系统进行开发，让考勤、统计变得更便捷。

可行性分析既不能以偏盖全，也不能过于追求细节，必须为决策提供有价值的证据。软件领域的可行性分析主要考虑四个因素：技术可行性分析，经济、社会效益分析，法律可行性分析，开发人员与进程可行性分析。

技术可行性分析：安卓系统是基于Java编写的，手机客户端采用AndroidSDK，开发人员具有一定Java功底，手机客户端在技术上是可行的。Web服务端用PHP编写，在LAMP环境下实现，PHP运用面向对象的编程思想进行Web开发，具有较高的开发效率和运行性能，后台数据库软件选用Oracle的Mysql，开发难度相对比较容易，因此在技术上是可行的。

经济、社会效益分析：主要进行成本-效益分析。从成本方面看，本系统的实现，计算机和网络硬件已经拥有，由于安卓是个开放的平台，AndroidSDK是免费提供的，网络客户端所用软件也都已经配备，因此成本很低；从效益方面看，本客户端主要是为广大师生提供便利。

法律可行性分析：从法律角度看，Android是个开放的平台，代码是开放的，加上基于Android的考勤系统主要为师生提供便利，作为毕业设计，并不以盈利为目的，且是自主开发设计，因此不会在社会上引起侵权或其它责任问题，在法律上是可行的。

开发人员与进程可行性分析：开发人员有一定的Java开发功底，并且学习能力较强以及指导老师的悉心教导完全可以胜任此次开发，因此开发进程方面是可行的。

**2 需求分析**

考勤系统对学校加强学生管理有着极其重要的作用，就一般的学校来说，它的设计内容非常复杂而且繁多，而且设计的模块也很多。

**2.1需求分析目标及任务**

1.基于Android的考勤系统的主要作用。

2.齐全，准确的找出本系统全部的功能，性能，限制。

3.找出全部的输入流，输出流。

**2.2需求分析组织形式**

1.通过调查研究,对本系统的功能进行分析。

2.通过访问，了解老师及教务处对本系统的需求情况。

**2.3需求分析信息整理**

下面是考勤系统设计的业务流程分析，原始的数据是学生基本的信息，系统要求统计学生的信息，并根据信息计算出的学生成绩基本情况导出学生综合测评分表单，成绩数据流如图2-1所示：

图2-1 综合测评成绩数据流

**2.4研发叙述说明**

本系统定位于学校等教育单位，C/S架构；操作系统服务端选择Linux系统，移动客户端选择Android系统。系统性质为MIS（管理信息系统）。

系统的非功能性需求

**1用户界面需求**

简洁、易用、易懂，美观、大方、标准，具备一定的兼容性。

**2软硬件环境需求**

软件环境：Linux操作系统。

硬件环境：386以上计算机。

**3软件质量需求**

易用、健壮、兼容性好、运行稳定、有一定安全保障。

**2.5研发标准与规范**

制定界面设计标准规范的目的是为了规范和统一软件界面设计制定软件界面设计标准与规范。

首先考虑标准化，在标准化的基础上进行界面的美工设计。

简单易用、简洁明了、兼容性好、标准、规范。

变量、方法名、类名采用驼峰写法。

代码设计方面，服务端采用MVC的代码架构方式。

MVC模式(Model-View-Controller)是软件工程中的一种软件架构模式，吧软件系统分为三个基本部分：模型(Model)、视图(View)和控制器(Controller)。

控制器层(Controller):负责转发请求、对请求进行处理。

视图层(View):界面设计人员进行图形界面设计。

模型(Model):程序员编写程序应有的功能(实现算法等等)、数据库专家进行数据管理和数据库设计(可以实现具体的功能)。

**3 概要设计**

**3.1系统功能**

考勤系统

系统管理

数据管理

院系管理

学生管理

冻结账号

清除缓存

重置密码

权限控制

考勤记录

班级数据

记录检索

报表导出

课程信息

教师信息

专业信息

信息修改

学生信息

缺勤管理

邮件通知

图3-1 考勤系统层次图

系统帮助

帮助

环境

由图3-1可以看出，该系统主要实现对学生考勤数据的管理，主要功能是录入并计算以学生为单位、以班级为单位、以学院为单位，进行多方面的数据统计，并生成考勤数据报表。系统结构分为5个模块：系统管理模块、数据管理模块、院系管理模块和学生管理模块。

* 系统管理：清除缓存、冻结账号、重置密码、权限控制。
* 数据管理：考勤记录、班级考勤数据、考勤数据检索及报表导出。
* 院系管理：课程信息、教师信息、专业信息等增删改查管理。
* 学生管理：管理学生数据、缺勤学生数据、缺勤邮件通知。
* 系统帮助：为本系统的操作提供技术指导。

**3.2系统总体流程**

由图3-2的系统总体流程图可以看出，本系统在录入学生测评明细前，首先需要选择综合测评的学期段，然后才进行录入工作，当录入完成所有学生的加分明细之后，执行计算操作，系统会自动统计出班内的各项能力的最高分以及计算出每个学生的综合测评和排名，最后在导出班级的排名表和学生的成绩表。

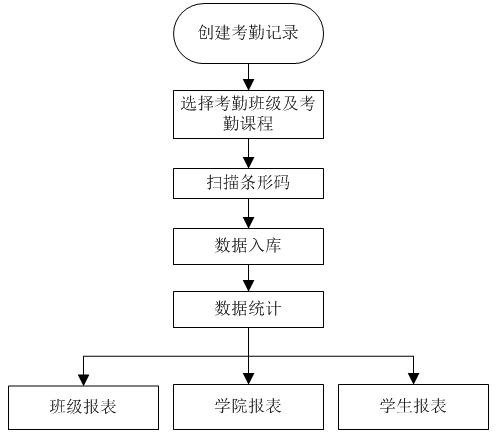


图3-2 系统总体流程

**3.3系统主要流程**

由图3-3和图3-4可知，系统主要的流程可分为客户端操作流程和服务端查询流程两部分。客户端操作流程描述数据的产生以及传递过程，服务端查询流程描述了报表的生成过程。

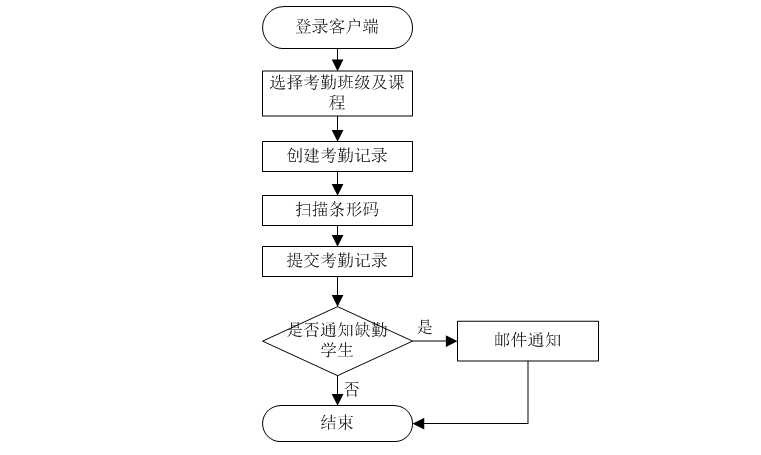


图3-3 客户端操作流程

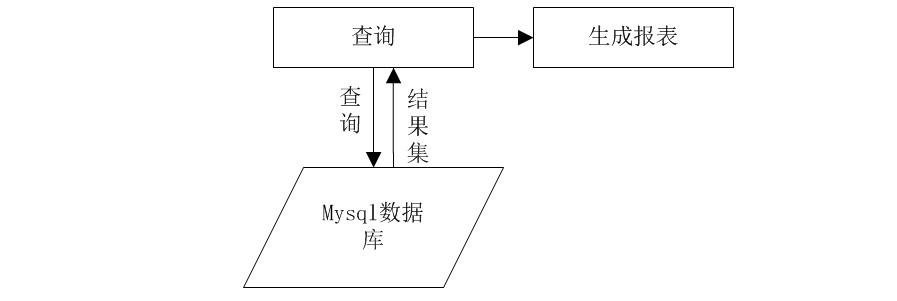


图3-4 服务端查询流程

**4 数据库设计**

**4.1 数据库Mysql简介**

MySQL是一个开放源码的小型关联式数据库管理[系统](http://baike.baidu.com/view/25302.htm)，开发者为[瑞典](http://baike.baidu.com/view/14445.htm)[MySQL AB](http://baike.baidu.com/view/2270659.htm)公司。MySQL被广泛地[应用](http://baike.baidu.com/view/220910.htm)在[Internet](http://baike.baidu.com/view/11165.htm)上的中小型网站中。由于其[体积](http://baike.baidu.com/view/274417.htm)小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站[数据库](http://baike.baidu.com/view/1088.htm)。

**4.2 数据库设计**

数据库设计主要是进行数据库的逻辑设计，即将数据按一定的分类、分组系统和逻辑层次组织起来，是面向用户的。数据库设计时需要综合各个业务逻辑，分析各个数据之间的关系，按照DBMS提供的功能和描述工具，设计出规模适当、正确反映数据关系、数据冗余少、存取效率高、能满足多种查询要求的数据模型。

数据库设计的步骤是：

1.数据库结构定义：目前的数据库管理系统（DBMS）有的是支持联机事务处理CLTP（负责对事务数据进行采集、处理、存储）的操作型DBMS，有的可支持数据仓库、有联机分析处理CLAP（指为支持决策的制定对数据的一种加工操作）功能的大型DBMS，有的数据库是关系型的、有的可支持面向对象数据库。针对选择的DBMS，进行数据库结构定义。

2.数据表定义：数据表定义指定义数据库中数据表的结构，数据表的逻辑结构包括：属性名称、类型、表示形式、缺省值、校验规则、是否关键字、可否为空等。关系型数据库要尽量按关系规范化要求进行数据库设计，但为使效率高，规范化程度应根据应用环境和条件来决定。数据表设计不仅要满足数据存储的要求，还要增加一些如反映有关信息、操作责任、中间数据的字段或临时数据表。  
 本系统数据库设计的模型图如4-1所示。

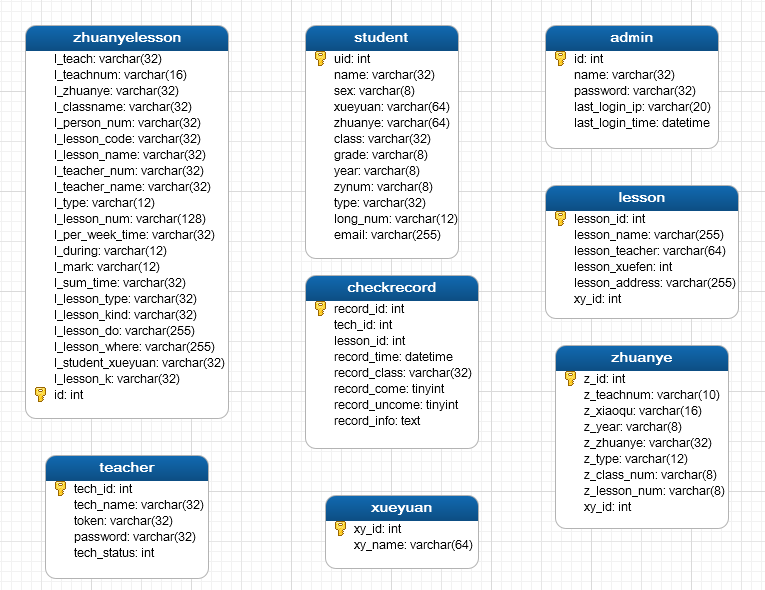


图4-1 考勤系统数据模型图

3.存储设备和存储空间组织：确定数据的存放地点、存储路径、存储设备等，备份方案，对多版本如何保证一致性和数据的完整性。

4.为了更好地组织数据和设计出实际应用数据库，应该注意如下问题：

关系数据结构的建立：在进行了数据基本结构的规范化重组后，还必须建立整体数据的关系结构。这一步设计完成后数据库和数据结构设计工作基本完成，只待系统实现时将数据分析和数据字典的内容代入到所设计的数据整体关系结构中，一个规范化数据库系统结构就建立起来了。

本系统设计的数据库总共包括8张表：

* 教师信息表：记录教师信息。
* 学生信息表：记录学生信息。
* 专业课程表：学校所有课程信息。
* 管理员表：管理员信息。
* 课程表：纳入考勤的课程信息。
* 考勤记录表：记录每次考勤的信息。
* 学院表：记录所有学院的信息。
* 专业信息表:记录各学院的专业信息。

主要数据表如表4-4、4-5、4-6、4-7、4-8所示:

表4-4 学生信息表(student)



表4-5 专业信息表(zhuanye)



表4-6 课程信息表(lesson)



表4-7 专业课程表(zhuanyelesson)

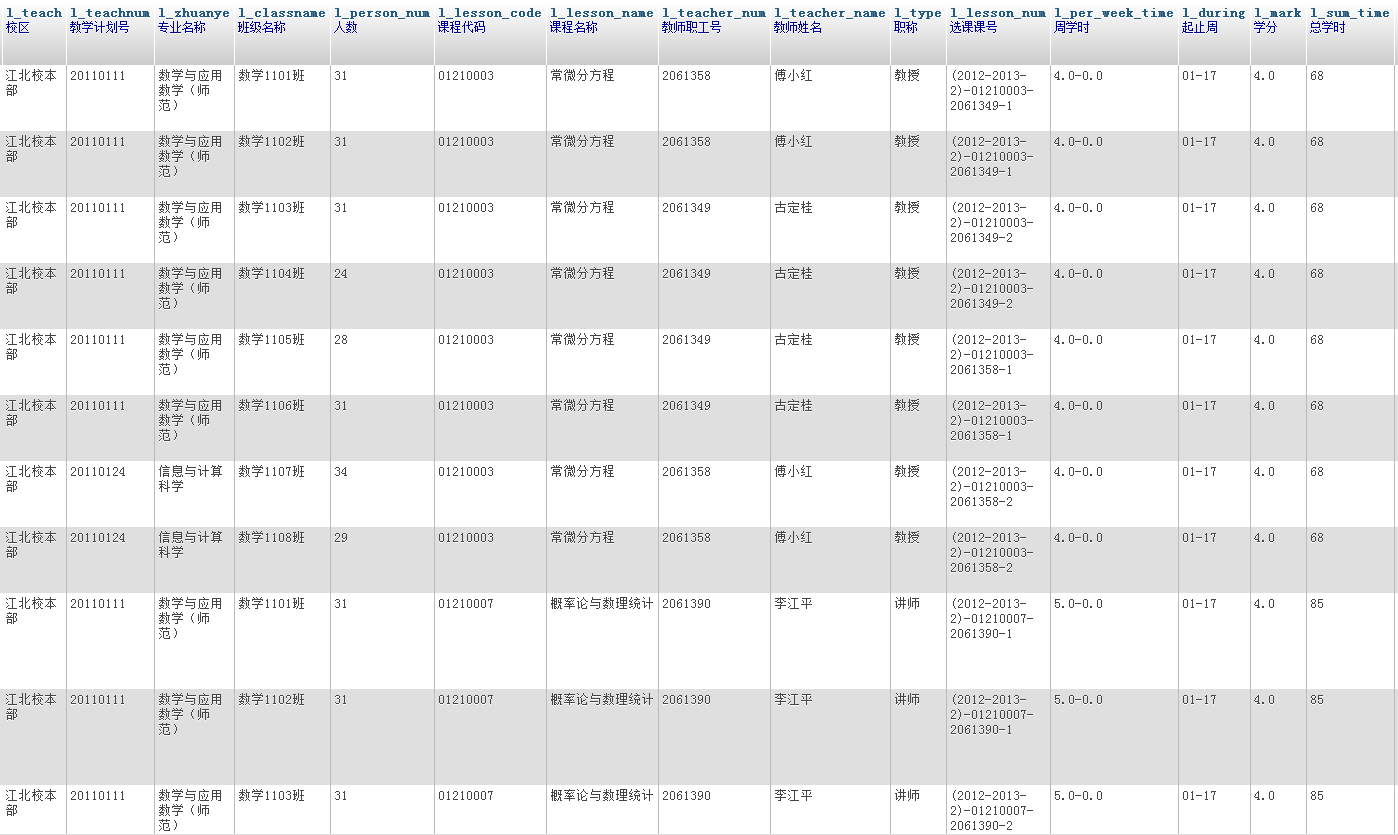


表4-8 考勤记录表(checkrecord)

**5 详细设计**



**5.1考勤记录创建模块实现**

**5.1.1考勤记录创建模块的流程图**

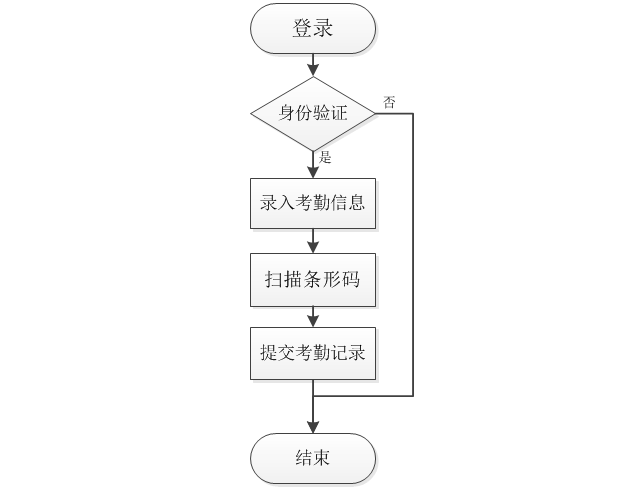


图5-1 考勤记录创建模块流程图

**5.1.2考勤记录创建设计界面**



图5-2 考勤记录创建界面

**5.2条形码扫描模块实现**

**5.2.1条形码扫描模块流程图**

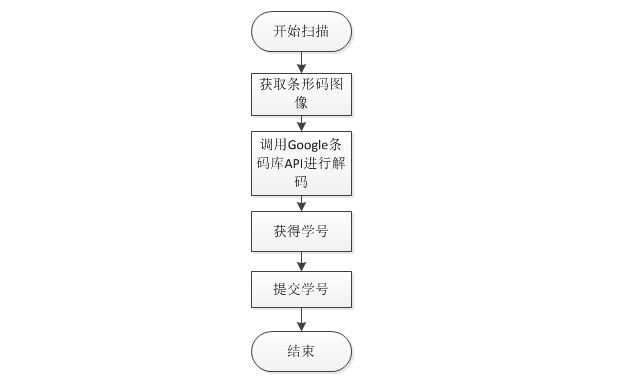


图5-3 条形码扫描流程图

**5.2.2条形码扫描模块的界面设计**



图5-4 条形码扫描界面

**5.3考勤数据模块的实现**

**5.3.1缺勤用户统计模块流程图**

缺勤用户统计模块的流程图如图5-5、5-6所示。

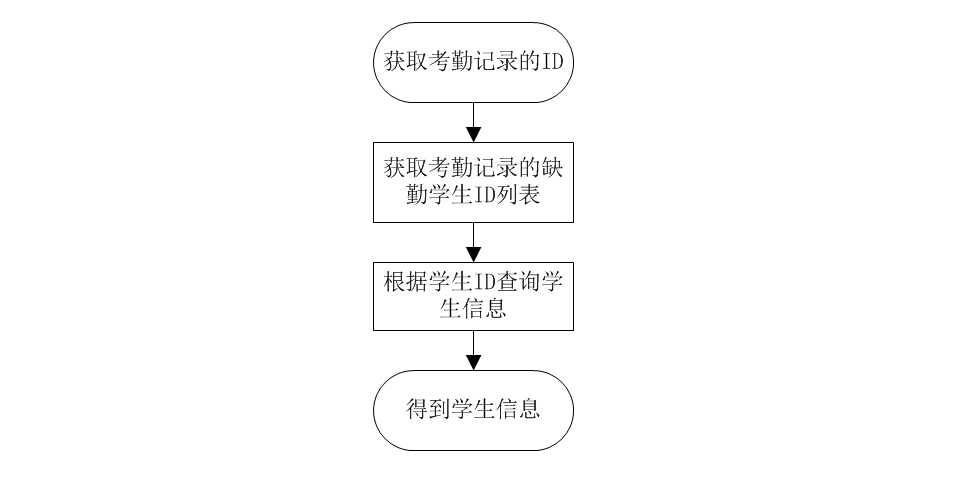


图5-5 缺勤用户统计流程图

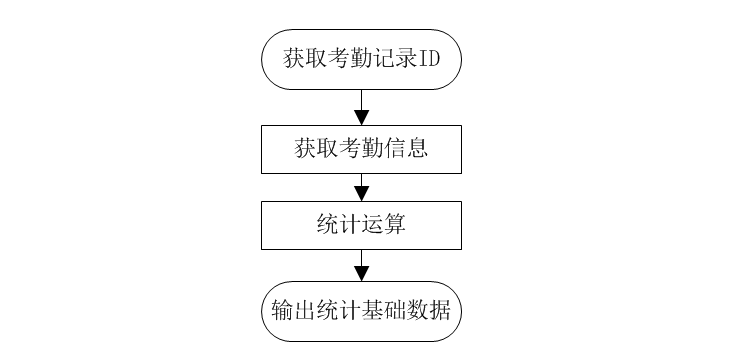


图5-6 缺勤比例统计流程图

**5.3.2缺勤用户统计模块的界面设计**

本系统的缺勤用户统计模块的界面设计如图5-6所示。

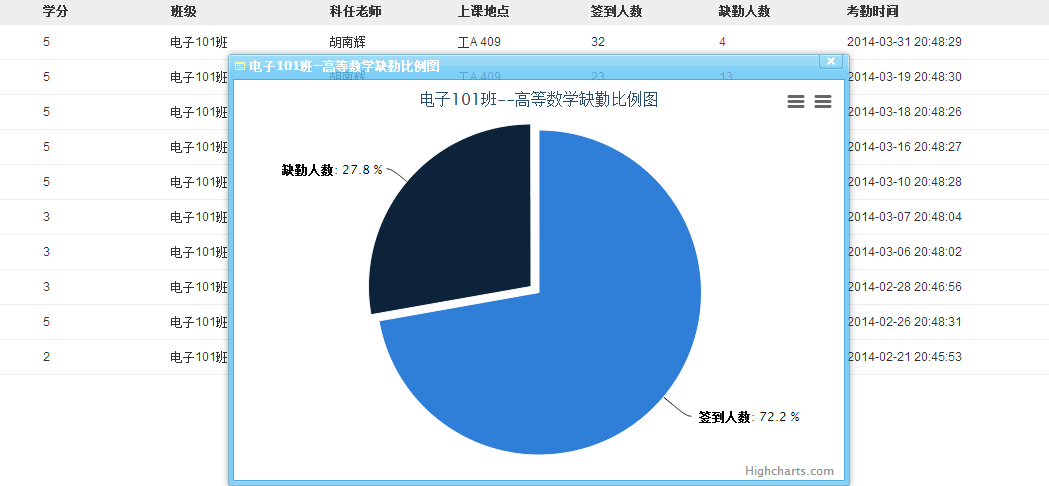
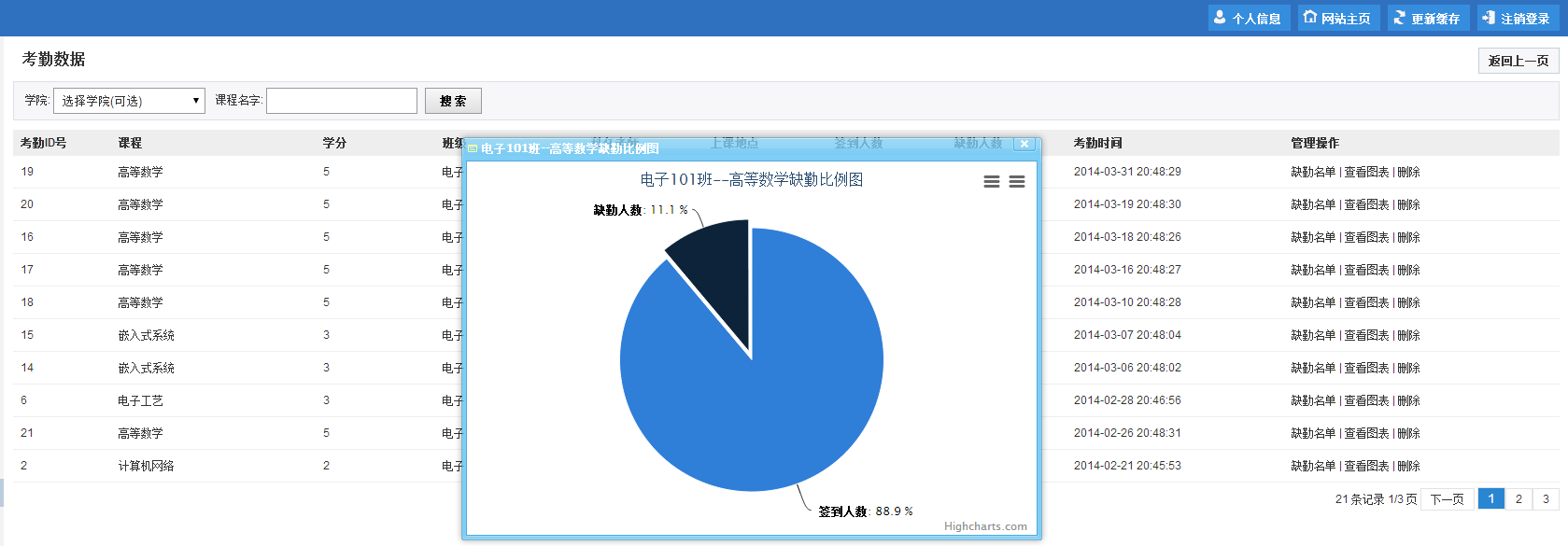


图5-6 考勤数据管理界面

**5.4班级考勤数据模块的实现**

**5.4.1班级考勤数据模块流程图**

班级考勤数据模块的流程图如图5-7所示

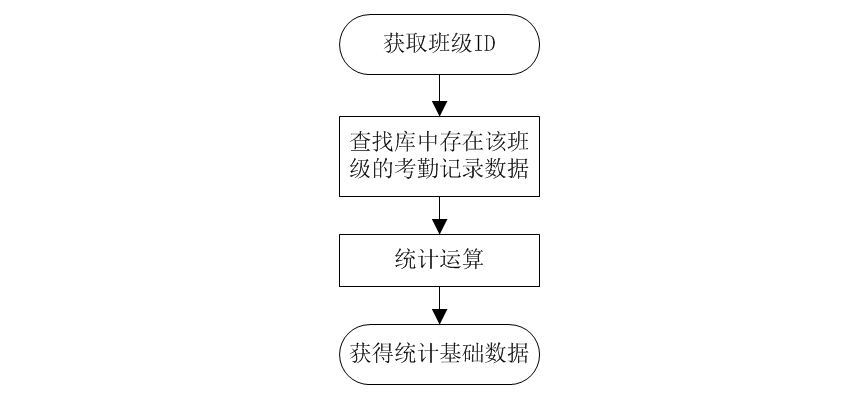


图5-7班级考勤数据模块流程图

**5.4.2班级考勤数据的界面设计**

本系统的班级考勤数据模块的界面设计如图5-8所示。

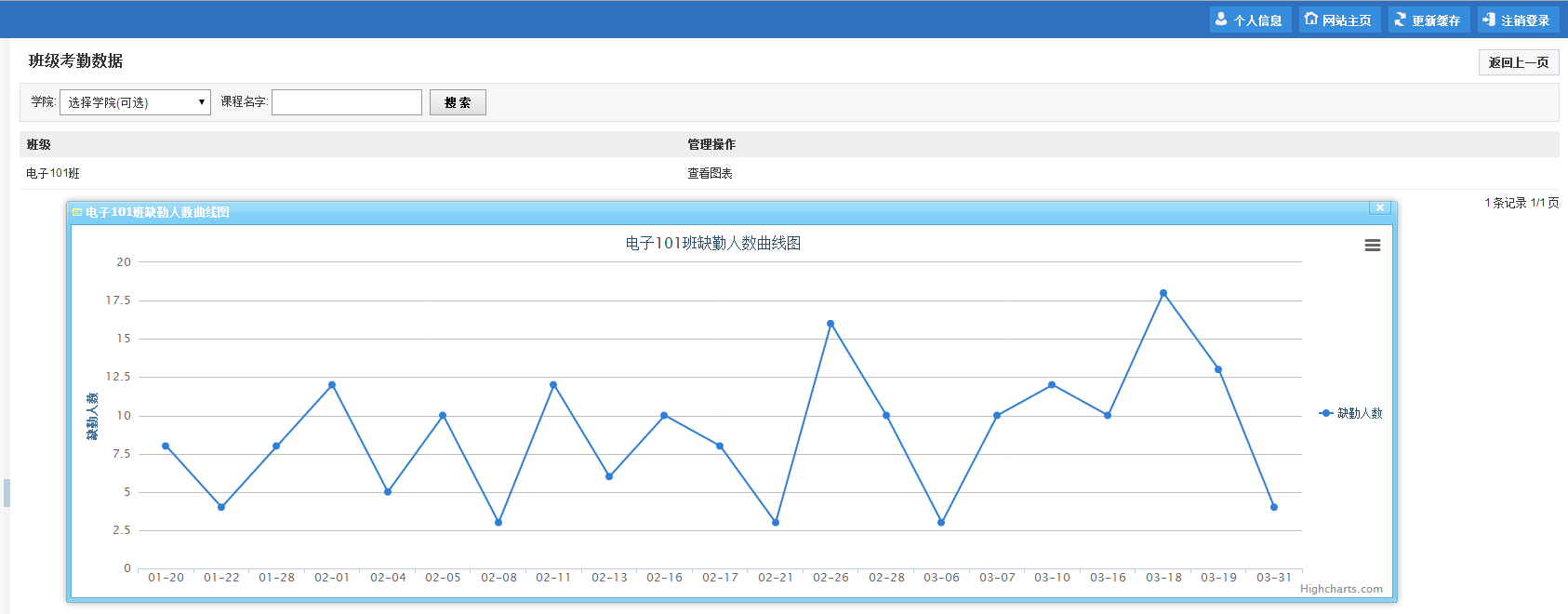


图5-8班级考勤数据模块的界面

**6 系统的测试运行与维护**

Android考勤系统在完成系统实施、投入正常运行之后，就进入了系统运行与维护阶段。

**6.1系统的测试**

1．兼容性测试

经测试，本系统可成功的在windows及Linux系统中运行，并具有良好的兼容性、跨平台性。

2．用户界面测试

本系统的用户界面风格满足客户要求，文字是显示正确，页面美观，文字图片组合完美，操作友好，能兼容各浏览器展示效果。

3. 功能测试

经过多次数据的录入计算，其最后结果与预期的结果一致，系统能成功的计算并统计出考勤数据报表。

**6.2系统的维护**

系统维护的目的是保证管理系统正常而可靠地运行，并能使系统不断得到改善和提高，以充分发挥作用。换言之，系统维护就是为了保证系统中的各个要素随着环境的变化始终处于最新的、正确的工作状态。

1．系统维护的内容

按照维护对象的不同，系统维护的内容可分为以下几类：

（1）系统应用程序维护

（2）数据维护

（3）代码维护

(4）硬件设备维护

2．系统维护的类型

系统维护的重点是系统应用软件的维护工作，按照软件维护的不同性质，可以划分为下面四种类型：

（1）纠错性维护 诊断和修正系统中遗留的错误。

（2）适应性维护 使系统适用环境的变化而进行的维护。

（3）完善性维护 为满足用户的需求而进行的维护。

（4）预防性维护 为将要发生的变化或调整而进行的主动性维护。

3．本系统的维护模块

本系统的维护功能主要是为保证系统能够正常的运行。

**7 总结**

通过此安卓考勤系统的开发，做到了理论联系实际，将书本上学到的知识与具体实践充分结合起来，进一步明确了开发管理信息系统的方法和思想；既加深了对考勤系统的认识，又进一步巩固了以前所学的计算机软件开发方面的知识，提高了软件开发的水平，从而为今后在实际工作中更好的应用计算机进行信息管理打下了坚实的基础。但是由于毕业设计时间较短，所以该系统还有许多不尽如人意的地方，比如没有考虑数据量大时候系统的运行情况，用户界面不够美观，出错处理不够等多方面问题。这些都有待进一步改善。提高科学管理水平首先要提高科学管理的意识，进行MIS的开发首先要具有科学的开发方法。成功MIS的基础是科学的管理加上科学的开发方式。企业的MIS开发必须结合实际，严格按照软件工程的思想进行才能保证MIS的高成功率和高效率。

总的来说，这次的毕业设计还是取到了较好的效果,实现了期望中的绝大部分功能，可以快速准确地统计学生的考勤情况，并给考勤人员清晰明了的报表,并将其导出电子档方便打印归档。

当然在这次的毕业设计中也遇到了不少的问题，大部分得到了很好的解决，但也有少部分遗留的问题。

在开发过程中我遇到了许许多多的问题，很多地方都不明白，经常有打退堂鼓的想法，我经常的请教同学和老师，最后的成果虽然不很完美，但我尽力了，并有所收获。

**附录**

**附录A考勤记录生成程序**

MVC分层中的Controller层

<?php

/\*

\*@Description:考勤系统API接口

\*@Author:学在囧途

\*/

class CheckApiAction extends Action{

/\*

\*@Description:创建考勤记录

\*/

public function CreateCheckRecord(){

$teacher\_id=trim($\_GET['teacher\_id']); //教师ID

$class\_name=trim($\_GET['class\_name']); //考勤班级

$lesson\_id=trim($\_GET['lesson\_id']); //考勤课程ID

if(!empty($teacher\_id)&&!empty($class\_name)&& !empty($lesson\_id)){

$record= D('CheckApi');

$record\_id=$record->getRecordId($teacher\_id, $class\_name, $lesson\_id);

echo $record\_id;

}else{

return false;

}

}

/\*

\*@Description:判断是否记录该次考勤记录，并返回该学生学生信息

\*/

public function CheckAction(){

$uid = trim($\_GET['student\_id']);

$record\_id = trim($\_GET['record\_id']);

if(!empty($uid) && !empty($record\_id)){

$info = $this->GetStudentInfo($uid);

if(!empty($info)){

$record = D('CheckApi');

$record->UpdataRecord($uid, $record\_id);

echo json\_encode($info);

}else{

$Msg['error'] = "该学生不存在";

echo json\_encode($Msg);

}

}else{

$Msg['error'] = "参数student\_id或record\_id不能为空";

echo json\_encode($Msg);

}

}

/\*

\*@Description:获取学生信息

\*/

private function GetStudentInfo($uid){

if(!empty($uid)){

$student = D('Student');

return $student -> getStudentInfo($uid);

}else{

return false;

}

}

/\*

\*@Description:手动提交此次考勤

\*/

public function OnSetCheckRecord(){

$record\_id = trim($\_GET['record\_id']);

if(!empty($record\_id)){

$record = D("CheckApi");

//执行提交事务

$status = $record->OnSetCheckRecord($record\_id);

return $status;

}

}

/\*

\*@Description:超过设置时间为提交考勤，系统自动提交考勤

\*/

public function AutoSetCheckRecord(){

$setTime = C("AutoCommitTime");

$record = D("CheckApi");

$record->AutoCommitCheckRecord($setTime);

}

}

?>

MVC分层中的Model层

<?php

/\*

\*@Description:考勤接口的Model层

\*@Author:学在囧途

\*/

class CheckApiModel extends Model{

private $dbhandle; //句柄

public function \_\_construct(){

$this->dbhandle = new model('checkrecord', '', '');

}

//创建record记录并返回记录ID

public function getRecordId($teacher\_id, $class\_name, $lesson\_id){

$this->dbhandle->query("INSERT INTO checkrecord(tech\_id, lesson\_id, record\_class, record\_time) VALUES('".$teacher\_id."', '".$lesson\_id."', '".$class\_name."', '".date("Y-m-d H:i:s", time())."')");

return mysql\_insert\_id();

}

//更新record

public function UpdataRecord($uid, $record\_id){

if(!empty($uid) && !empty($record\_id)){

$recordData = $this->dbhandle->query("SELECT record\_come, record\_info FROM checkrecord WHERE record\_id = '".$record\_id."'");

if(empty($recordData[0]['record\_info'])){

$recordinfo['come\_uidList'][0] = $uid;

$info = json\_encode($recordinfo);

}else{

$info = json\_decode($recordData[0]['record\_info'], true);

//判断该学生是否已经存在于考勤记录表中

if(!in\_array($uid, $info['come\_uidList'])){

$record\_come = $recordData[0]['record\_come'] + 1;

$info['come\_uidList'][] = $uid;

}else{

$Msg['error'] = "请勿重复扫描该学生图书证进行考勤";

echo json\_encode($Msg);

exit();

}

}

$info = json\_encode($info);

$this->dbhandle->query("UPDATE checkrecord SET record\_come = '".$record\_come."', record\_info = '".$info."' WHERE record\_id = '".$record\_id."'");

}

}

//commit考勤记录

public function OnSetCheckRecord($record\_id){

if(!empty($record\_id)){

//获取班级名称

$record = $this->dbhandle->query("SELECT record\_come ,record\_class, record\_info FROM checkrecord WHERE record\_id = '".$record\_id."'");

$record\_class = $record[0]['record\_class'];

$record\_come = $record[0]['record\_come'];

$record\_info = json\_decode($record[0]['record\_info'], true);

$record\_info = $record\_info['come\_uidList'];

//获取该班级总人数

$class\_num = $this->GetClassNum($record\_class);

//var\_dump($class\_num);

//获取班级所有人员的uid

$allUid = $this->GetClassAllUid($record\_class);

//入库record\_info

$info = array();

foreach ($allUid as $key => $value) {

if(!in\_array($value['uid'], $record\_info)){

$info['uncome\_List'][] = $value['uid'];

}else{

$info['come\_uidList'][] = $value['uid'];

}

}

//未签到人数

$uncomeNum = count($info['uncome\_List']);

//签到人数

$comeNum = count($info['come\_uidList']);

$info['comeNum'] = $comeNum;

$info['uncomeNum'] = $uncomeNum;

$info['sumNum'] = $class\_num;

$infoMsg = json\_encode($info);

$this->dbhandle->query("UPDATE checkrecord SET record\_uncome = '".$uncomeNum."', record\_info = '".$infoMsg."' WHERE record\_id = '".$record\_id."'");

$Msg['status'] = $record\_class."此次考勤结束并有效";

echo json\_encode($Msg);

}else{

$Msg['error'] = "所提交的考勤记录表ID为空，请联系管理员";

echo json\_encode($Msg);

}

}

//自动Commit考勤记录

public function AutoCommitCheckRecord($setTime){

//获取过期未Commit的Record

$List = $this->dbhandle->query("SELECT record\_id FROM checkrecord");

foreach ($List as $key => $value) {

$this->OnSetCheckRecord($value['record\_id']);

}

}

//根据班级名称获取班级总人数

private function GetClassNum($class\_name){

if(!empty($class\_name)){

$class\_num = $this->dbhandle->query("SELECT COUNT(uid) AS num FROM student WHERE class = '".$class\_name."'");

$class\_num = $class\_num[0]['num'];

return $class\_num;

}else{

return false;

}

}

//根据班级名称获取该班级所有人的uid

private function GetClassAllUid($class\_name){

if(!empty($class\_name)){

return $this->dbhandle->query("SELECT uid FROM student WHERE class = '".$class\_name."'");

}else{

return false;

}

}

}

?>

**附录B报表统计模块程序：**

MVC分层中的Controller层

//根据record\_id获取当次缺勤学生名字

public function getAllUncomeStudentName(){

$record\_id = trim($\_POST['record\_id']);

if(isset($record\_id)){

$record = D("CheckData");

$nameList = $record->getAllUncomeStudentName($record\_id);

echo json\_encode($nameList);

}else{

return false;

}

}

//根据record\_id获取该次考勤人数

public function getCheckNumInfo(){

$record\_id = trim($\_POST['record\_id']);

if(isset($record\_id)){

$record = D("CheckData");

$info = $record->getCheckNumInfo($record\_id);

$info[0]['sum'] = $info[0]['record\_come'] + $info[0]['record\_uncome'];

echo json\_encode($info);

}

}

//获取某个班级考勤数据

public function CheckClassData(){

$class\_name = trim($\_POST['class\_name']);

if(!empty($class\_name)){

$check = D("CheckData");

$data = $check->getClassData($class\_name);

echo json\_encode($data);

}

}

MVC分层中的View层

<SCRIPT type=text/javascript>

$(function(){

$('.showuncome').click(function(){

var record\_id = $(this).attr('id');

var infoShow = $(this).attr('name');

$.ajax({

url: "\_\_APP\_\_/CheckData/getAllUncomeStudentName",

type: "POST",

data: {record\_id:record\_id},

dataType: "json",

error: function(){

alert('异步查询表失败,请联系管理员');

},

success: function(data){

asyncbox.open({

html:'<div style="width:380px;height:280px;font-size:16px;">'+data.studentName+'</div>',

title:infoShow+'缺勤名单',

});

}

});

});

$(".show").click(function(){

var className = $(this).attr('name');

var showTitle = className + "缺勤人数曲线图";

$.ajax({

url: "\_\_APP\_\_/CheckData/CheckClassData",

type: "POST",

data: {class\_name:className},

dataType: "json",

error: function(){

alert('异步查询表失败,请联系管理员');

},

success: function(data){

var xline = [];

var yline = [];

for(var i in data){

xline.push(data[i].record\_time);

yline.push(parseInt(data[i].record\_uncome));

}

asyncbox.open({

html:'<div id="container" style="min-width: 1420px; height: 400px; margin: 0 auto"></div>',

title:showTitle

});

new Highcharts.Chart({

chart: {

renderTo:"container",

plotBackgroundColor: null,

plotBorderWidth: null,

plotShadow: false

},

title: {

text: showTitle,

x: -20 //center

},

subtitle: {

text: '',

x: -20

},

xAxis: {

categories: xline

},

yAxis: {

title: {

text: '缺勤人数'

},

plotLines: [{

value: 0,

width: 1,

color: '#808080'

}]

},

tooltip: {

valueSuffix: '人'

},

legend: {

layout: 'vertical',

align: 'right',

verticalAlign: 'middle',

borderWidth: 0

},

series: [{

name: '缺勤人数',

data: yline

}]

});

}

});

});

});

</SCRIPT>

**参考文献**

# 致谢

通过本次的毕业设计，我了解到了许多关于Android方面的知识。本设计是在杨老师的指导下完成的，在论文工作期间，经过老师耐心的指导和自己我讨论所遇到的各种问题，并指点出我的不足之处，使我顺利攻克很多知识盲区，拓宽了自己的知识面。他营造了一种轻松的学习氛围，使我顺利完成毕业论文的的设计，在此对他们表示由衷的敬意和至深的感谢！