**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**

随着我国教育事业的高速发展,高等院校的规模和师生队伍的不断扩大,为以往传统的考勤管理工作带来了很大的困扰。基于Android系统的学生考勤系统将教师从繁重的费时费力的考勤中解放出来,使他们更专心教学。考勤系统利用移动平台中应用最广泛的Andriod系统，分为两个部分，一部分为教师客户端，主要是通过扫描学生图书证上的二维码记录考勤学生信息。另一部分为Web服务端，主要提供考勤数据管理、院系管理、学生管理服务，用于多角度统计学生出勤信息，便于教师和教学管理者及时掌握教学情况，为教学决策提供依据。此系统界面友好,易于操作,实现了预期的所有功能,将为全院师生的考勤管理工作提供了一个科学便利的平台。

**关键词：**Android，学生，考勤，条形码扫描

**Abstract**

With the rapid development of Chinese education, the scale of higher education and the ranks of teachers and students are expanding,which has brought great distrouble to the traditional attendance management in the past.Student attendance system that based on Android will liberate teachers from heavy laborious attendance out, making them concentrate more on teaching. Attendance system which is used widely among the mobile platform is divided into two parts. One is for teacher clients, mainly through scanning two-dimensional code on student library card to record their attendance information.Another part is the Web server, the main provider of time and attendance data management, faculty and student management services.Moreover,it is used to count student attendance information from multi-angle and so convenient for teachers and educational administrators to grasp the teaching situation, which will the basis for instructional decisions. This system is user-friendly, easy to operate and achieve all the expected functions. Therefore, it will provide a scientific and convenient platform for managing attendance of all students and teachers.

**Keywords:** Android, students, student attendance, barcode scanning

**目录**

[1 前言 1](#_Toc32304)

[1.1选题背景 1](#_Toc1184)

[1.2相关研究现状及前景 1](#_Toc13285)

[1.3研究目的 1](#_Toc1863)

[1.4可行性分析 1](#_Toc11971)

[1.5论文的主要内容 2](#_Toc17371)

[2 需求分析 3](#_Toc20637)

[2.1引言 3](#_Toc14883)

[2.2学生考勤系统的特点 3](#_Toc12897)

[2.2.1用户特点 3](#_Toc16044)

[2.2.2系统的特点 3](#_Toc16490)

[2.2.3系统功能 3](#_Toc30273)

[2.2.4运行环境 4](#_Toc22195)

[2.2.5软件支持 4](#_Toc21993)

[2.3功能需求分析 4](#_Toc17238)

[2.3.1教师方面 4](#_Toc5077)

[2.3.2学生方面 4](#_Toc2610)

[2.3.3教务处方面 4](#_Toc6393)

[2.4需求分析信息整理 5](#_Toc23228)

[2.5考勤系统结构示意图 5](#_Toc16411)

[3 概要设计 7](#_Toc28986)

[3.1考勤系统功能结构 7](#_Toc22167)

[3.2系统总体流程 7](#_Toc9983)

[3.3服务端流程 8](#_Toc13371)

[3.4客户端流程 9](#_Toc20832)

[4 数据库设计 9](#_Toc10964)

[4.1 Mysql数据库简介 10](#_Toc19835)

[4.2 数据库概念结构设计 10](#_Toc21359)

[4.3 数据库物理结构设计 12](#_Toc18182)

[5 详细设计 17](#_Toc11867)

[5.1考勤客户端信息采集模块实现 17](#_Toc31420)

[5.1.1考勤客户端信息采集模块的流程图 17](#_Toc29799)

[5.1.2考勤记录创建设计 17](#_Toc31021)

[5.2条形码扫描模块实现 18](#_Toc11001)

[5.2.1条形码扫描模块流程图 18](#_Toc21742)

[5.2.2条形码扫描模块设计 19](#_Toc23021)

[5.3考勤数据模块的实现 19](#_Toc16266)

[5.3.1缺勤用户统计模块流程图 19](#_Toc31069)

[5.3.2缺勤用户统计模块设计 20](#_Toc29333)

[5.4班级考勤数据模块的实现 21](#_Toc7877)

[5.4.1班级考勤数据模块流程图 21](#_Toc28379)

[5.4.2班级考勤数据设计 22](#_Toc27676)

[6 系统的测试运行与维护 23](#_Toc10623)

[6.1系统的测试 23](#_Toc6359)

[6.1.1兼容性测试 23](#_Toc16251)

[6.1.2用户界面测试 23](#_Toc27640)

[6.1.3 功能测试 23](#_Toc18934)

[6.1.4 压力测试 23](#_Toc30608)

[6.2系统的维护 24](#_Toc564)

[6.2.1．系统维护的内容 24](#_Toc10593)

[6.2.2．系统维护的类型 24](#_Toc6714)

[6.2.3．系统维护的作用 24](#_Toc8581)

[7 总结 25](#_Toc5973)

[附录 26](#_Toc1852)

[参考文献 36](#_Toc9774)

[致谢 37](#_Toc5465)

# 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，致力于为教师提供便利的考勤方式，以及考勤数据能及时反馈考勤方面问题带来的便利，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。

## 1.2相关研究现状及前景

传统的考勤方式，叫号点名，难以避免代考勤、考勤数据零散的现状，致使教师、教务秘书处无法即时知道当次考勤的一些数据，也无法通过现代的一些通讯工具，如短信、邮件等，对缺勤学生进行提醒、警告等消息通知。考勤效率成本高，达到的效益低。

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，通过便捷的考勤方式，可迅速暴露出已存在的问题，相关人员可及时跟进并解决。

## 1.3研究目的

通过考勤系统进行考勤可以快速得到缺勤学生人数、比例、名单等信息，而且对于缺勤学生，可以实时对学生进行邮件、短信等方式通知其来上课补勤，并且数据会即时出现在系统后台的报表管理内，教务秘书、老师可对数据进行实时分析，发现严重缺勤情况可即时找相关人员进行谈心或教育，这样有效提高了考勤的效率，以最低的成本达到最佳的考勤效果。

## 1.4可行性分析

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[1]。

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[2]，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[3]。

## 1.5论文的主要内容

论文主要从论文主要从系统设计、数据库设计介绍了考勤系统的设计，通过介绍主要功能的流程图，简述实现过程。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，主要是通过扫描图书证上的二维码记录考勤学生信息。另一部分为web服务端，主要提供考勤数据管理、院系管理、学生管理服务。

2 需求分析

## 2.1引言

基于Android系统的学生考勤系统是以学生考勤工作为出发点，通过普遍的安卓客户端，降低考勤成本，使用数据库技术将所有的考勤管理数据统一管理起来，从而形成一个有规范、有条理的系统化系统资源。

考勤系统就是把分散、凌乱的考勤信息集中到一起，实现系统集中、统一、规范的管理模式。实现考勤数据分类的管理功能，考勤数据长久存储功能，以及信息检索、修改、存取等功能[4]。

## 2.2学生考勤系统的特点

### 2.2.1用户特点

使用本系统的最终用户为学校的众多教职工人员。从高校教职工人员对计算机相关技术的熟悉和掌握程度等方面考虑，本考勤系统力争做到功能完备，操作方法简单明了，系统本身易于维护管理，极大限度的满足需求的变更。

### 2.2.2系统的特点

该系统与传统的考勤系统相比，具有使用方便、功能完善、省时省力的特点外，还具有以下优点:

(1)利用图书证作为考勤卡，重复使用体现了物尽其用的价值，并且节约成本，通过扫描图书证上的条形码，具有强抗干扰能力，考勤速度也快。

(2)图书证一旦丢失可以立即挂失补办，不影响考勤，不丢失任何考勤数据。

(3)考勤设备利用安卓作为客户端运行平台，随着安卓手机的普及，极大的降低了使用成本，以及既有利于在校园内推广使用。

(4)采集完考勤数据，即可生成相应的报表，即时性强。

### 2.2.3系统功能

本系统不仅适合于众高校使用，如果做得完善，也同样适用于相类似的单位。

“基于Android系统的学生考勤系统”可方便实现如下功能：

(1)可实现联网操作，保证了数据的实时性传输、统计和存储，保证数据不丢失。

(2)实现对教职工账号进行冻结操作，防止客户端产生虚假数据，避免对部分涉及汇总数据的功能造成数据不准确的影响。

(3)实现联网实时控制Android考勤客户端。

(4)系统的存储容量大，可长时期保存考勤数据。

(5)为教师、教务秘书等人员提供方便快捷的查询、管理界面，实现考勤记录备份、处理、查询等操作，实现人性化服务。

### 2.2.4运行环境

基于Android系统的学生考勤系统客户端适用于Android平台版本在2.0以上的Android机子，服务端可运行在windows/linux操作系统的Nginx服务器中，采用Nginx作为Web服务器，是因为Nginx在同等访问量的情况下，系统资源的使用远比Apache要低，并且响应更快[5]。考勤系统不需要依赖特殊的第三方组件，服务端具有一定的跨平台能力。

### 2.2.5软件支持

手机客户端采用Java进行开发，服务端采用PHP+Mysql进行开发，PHP最大的特色是简单并与MySQL天生的结合性，采用PHP+Mysql进行开发软件，缩短了研发周期，并且具有良好的跨平台能力[6]。

手机客户端:手机平台需要是Android2.0以上版本，并且支持JDK1.6+。

服务端：支持PHP/Mysql等服务的运行。

## 2.3功能需求分析

### 2.3.1教师方面

通过手机客户端扫描学生证条形码，而不需要再叫号登记考勤，方便了考勤的过程，更方便教师统计关于自己课程方面考勤数据统计，以达到可以及时制定、调整自己的教学计划和方案。不同职能的教师关注数据的维度不同，考勤系统提供多维度的数据，更直观得看到目前存在的问题。

### 2.3.2学生方面

学生参与考勤只需要携带图书证即可，避免了考勤时，代同学签到、喊到的情况，对于缺勤学生，系统会下发短信、邮件进行通知，可以让部分学生在一定时间内迅速赶来上课，在一定程度上提高了学生的出勤率。

### 2.3.3教务处方面

可以通过每天各位老师创建的考勤记录，及时得知每个学生、班级的缺勤情况，从而能更快的通知学院、辅导员对缺勤率大的班级、经常缺勤的学生，进行教育、谈心。

## 2.4需求分析信息整理

图2-1是考勤系统设计的业务流程分析，原始的数据是某次考勤的学生考勤记录信息，系统要求统计学生考勤的记录信息，并以班级为单位根据计算出某次考勤的缺勤学生名单、缺勤率及班级考勤曲线图。

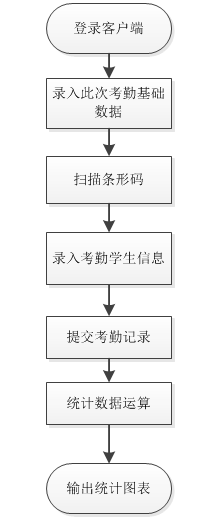


图2-1 考勤系统数据流

## 2.5考勤系统结构示意图

基于Android系统的学生考勤系统可以直接以互联网为传输媒介，进行考勤信息的传输。将考勤系统的主机和众多Android考勤设备接入互联网，即可马上进行工作。考勤系统通过互联网连接实现对考勤设备的基础数据下发，Android考勤设备通过互联网连接实现对考勤数据的上传。考勤系统的结构示意图如图2-2所示。

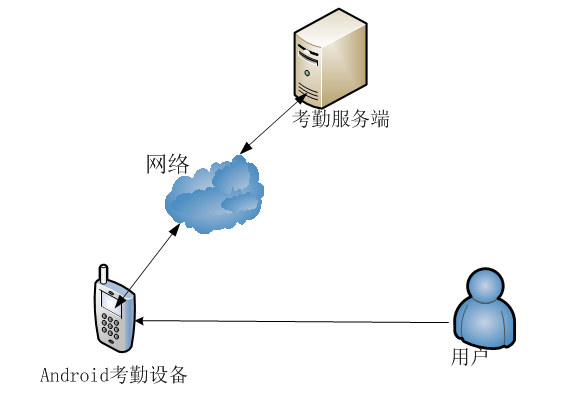


图2-2 考勤系统结构示意图

# **3 概要设计**

## 3.1考勤系统功能结构

考勤系统主要实现对学生考勤数据的管理，主要功能是进行以学生为单位、以班级为单位的数据统计，并生成考勤数据报表。系统结构分为5个模块，系统层次结构图如图3-1所示。

(1)系统管理：清除缓存、冻结账号、重置密码、权限控制。管理员可以通过此功能对教师账号、普通管理员账号进行权限控制，以达到保证系统数据的一致性及准确性。

(2)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。数据管理，记录着所有课程每一次考勤的考勤记录，通过此功能，可以查看、导出班级的考勤记录及图表。

(3)院系管理：课程信息、教师信息、专业信息等增删改查管理。不同学期、不同教师、不同专业的课程相关信息是变化的，管理员可以通过此功能进行动态调整，以保证基础数据的准确性。

(4)学生管理：管理学生数据、缺勤学生数据、缺勤邮件通知。保存每个学生缺勤数据，通过系统设置，可及时通知缺勤学生，达到警惕学生作用。

(5)系统帮助：为本系统的操作提供技术指导。

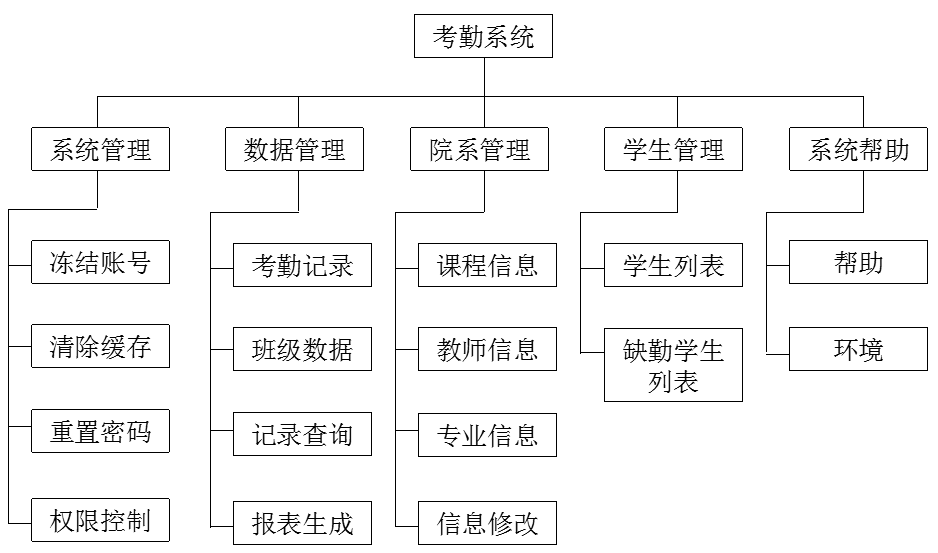


图3-1 考勤系统层次图

## 3.2系统总体流程

由图3-2的系统总体流程图可以看出，本系统在录入学生考勤记录后，会得到考勤的数据报表，维度包括班级报表、学院报表、学生报表。系统主要的流程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，服务端通过考勤记录的ID查询出结果集并生成报表的过程。

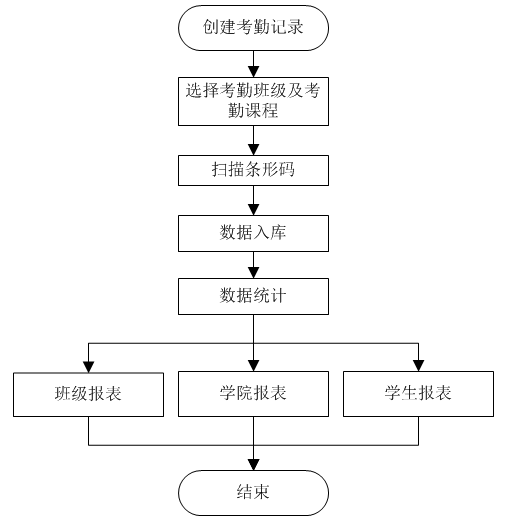


图3-2 系统总体流程

## 3.3服务端流程

服务端通过考勤记录的ID查询出结果集并生成报表，流程如图3-3所示。

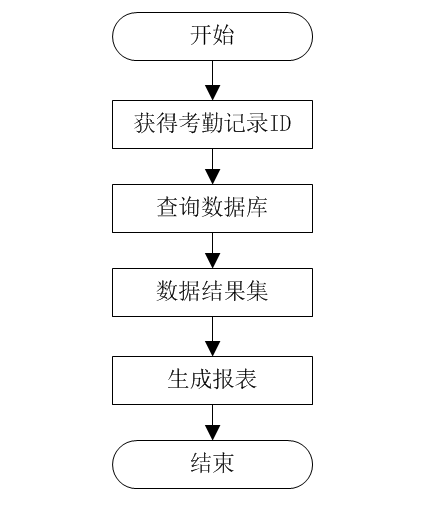
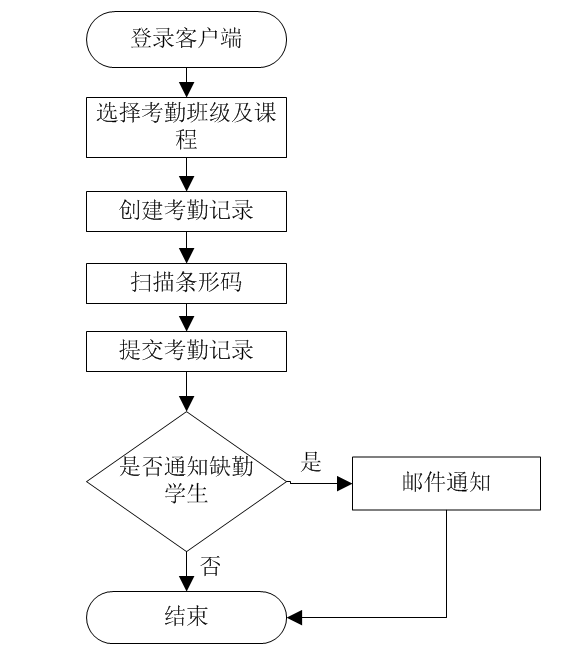


图3-3 服务端操作流程

## 3.4客户端流程

客户端流程描述数据的产生以及传递过程，如图3-4所示。



# **4 数据库设计**

图3-4 客户端操作流程

## 4.1 Mysql数据库简介

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)[7]。

数据库设计的质量如何直接影响到考勤管理系统运行结果的正确性和产品的质量，因此，数据库设计一定要科学、严谨、灵活，还要严格遵照数据库的范式理论，数据库设计时还要充分依据本项目的实际情况，反复评估、修改，根据本项目的实际情况，采用MySQL作为系统的数据库管理系统。

## 4.2 数据库概念结构设计

考勤系统数据库概念结构设计的实体主要有教师、学生、考勤客户端、学生图书证等，这些实体间通过考勤规则关系关联在一起，每个实体间都对应一定的关系。考勤系统的部分E-R图如图4-1至图4-4所示。

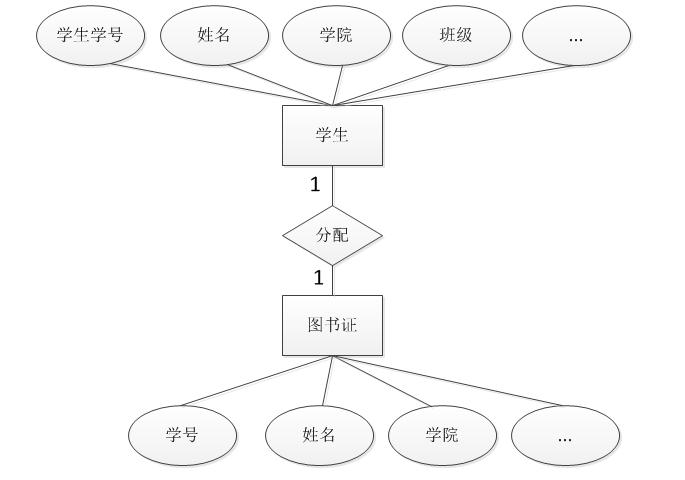


图4-1 学生与图书证实体E-R图

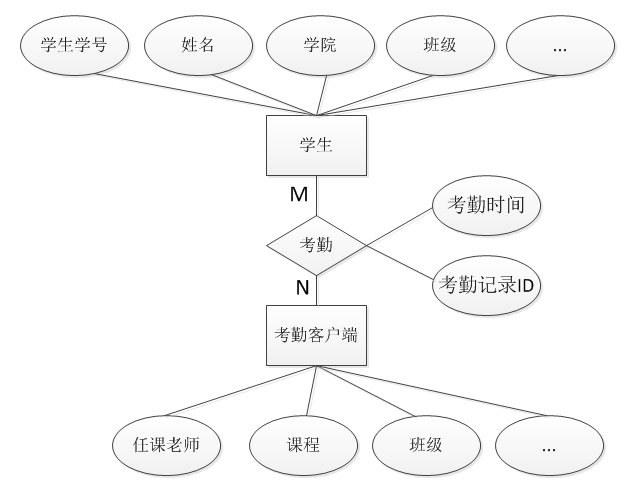


图4-2 学生与考勤客户端实体E-R图

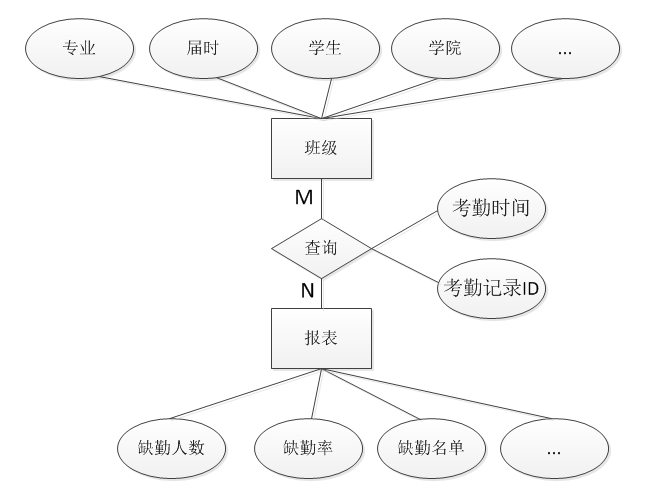


图4-3 学生与考勤客户端实体E-R图

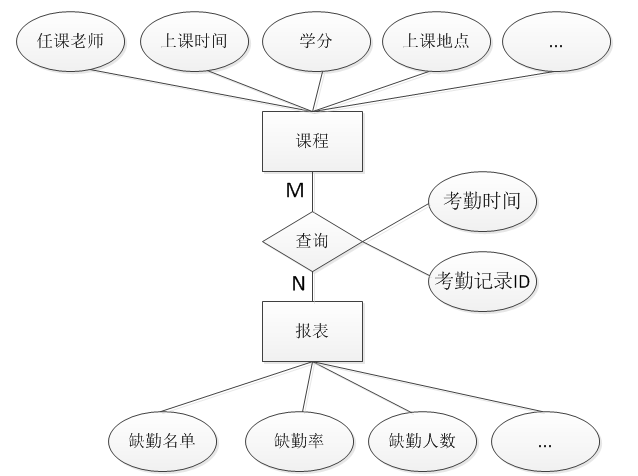


图4-4 学生与考勤客户端实体E-R图

## 4.3 数据库物理结构设计

数据库物理结构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。

(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。

数据库表结构设计如表4-5、4-6、4-7、4-8、4-9、4-10、4-11、4-12所示:

表4-5管理员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否允许为空 | 是否主键 |
| id | 管理员自增ID | int | 11 | 否 | 是 |
| name | 姓名 | varchar | 32 | 否 | 否 |
| password | 密码 | varchar | 32 | 否 | 否 |
| last\_login\_ip | 上次登录IP地址 | varchar | 20 | 否 | 否 |
| last\_login\_time | 上次登录时间 | datetime |  | 否 | 否 |

表4-6课程表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否允许为空 | 是否主键 |
| lesson\_id | 课程ID | int | 8 | 否 | 是 |
| lesson\_name | 课程名称 | varchar | 255 | 否 | 否 |
| lesson\_teacher | 课程老师 | varchar | 64 | 否 | 否 |
| lesson\_xuefen | 课程学分 | int | 8 | 否 | 否 |
| lesson\_address | 上课地点 | varchar | 255 | 否 | 否 |
| xy\_id | 学院ID | int | 8 | 否 | 否 |

表4-7考勤记录表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否允许为空 | 是否主键 |
| record\_id | 考勤记录自增ID | int | 8 | 否 | 是 |
| tech\_id | 考勤教师ID | int | 8 | 否 | 否 |
| lesson\_id | 课程ID | int | 8 | 否 | 否 |
| record\_time | 考勤时间 | datetime |  | 否 | 否 |
| record\_class | 考勤班级 | varchar | 32 | 否 | 否 |
| record\_come | 签到人数 | tinyint | 3 | 否 | 否 |
| record\_uncome | 未签到人数 | tinyint | 3 | 否 | 否 |
| record\_info | 考勤记录信息 | text |  | 否 | 否 |

表4-8学院列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否允许为空 | 是否主键 |
| xy\_id | 学院自增ID | int | 8 | 否 | 是 |
| xy\_name | 学院名称 | varchar | 64 | 否 | 否 |

表4-9专业课程表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否允许为空 | 是否主键 |
| id | 自增ID | int | 8 | 否 | 是 |
| l\_teach | 校区 | varchar | 32 | 否 | 否 |
| l\_teachnum | 教学计划号 | varchar | 16 | 否 | 否 |
| l\_zhuanye | 专业名称 | varchar | 32 | 否 | 否 |
| l\_classname | 班级名称 | varchar | 32 | 否 | 否 |
| l\_person\_num | 人数 | varchar | 32 | 否 | 否 |
| l\_lesson\_code | 课程代码 | varchar | 32 | 否 | 否 |
| l\_lesson\_name | 课程名称 | varchar | 32 | 否 | 否 |
| l\_teacher\_num | 教师职工号 | varchar | 32 | 否 | 否 |
| l\_teacher\_name | 教师姓名 | varchar | 32 | 否 | 否 |
| l\_type | 职称 | varchar | 12 | 否 | 否 |
| l\_lesson\_num | 选课课号 | varchar | 128 | 否 | 否 |
| l\_per\_week\_time | 周学时 | varchar | 32 | 否 | 否 |
| l\_during | 起止周 | varchar | 12 | 否 | 否 |
| l\_mark | 学分 | varchar | 12 | 否 | 否 |

表4-10专业信息表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | | | 类型 | 长度 | | 是否允许为空 | 是否主键 |
| z\_id | 专业课自增ID | | | int | 8 | | 否 | 是 |
| z\_teachnum | 教学计划号 | | | varchar | 10 | | 否 | 否 |
| z\_xiaoqu | 校区 | | | varchar | 16 | | 否 | 否 |
| z\_year | 开设时间 | | | varchar | 8 | | 否 | 否 |
| z\_zhuanye | 专业名称 | | | varchar | 32 | | 否 | 否 |
| z\_type | 层次 | | | varchar | 12 | | 否 | 否 |
| z\_class\_num | 班级数 | | | varchar | 8 | | 否 | 否 |
| z\_lesson\_num | 正式课程数 | | | varchar | 8 | | 否 | 否 |
| xy\_id | 学院ID | | | int | 8 | | 否 | 否 |
| l\_sum\_time | | 总学时 | varchar | | 32 | 否 | | 否 |
| l\_lesson\_type | | 课程性质 | varchar | | 32 | 否 | | 否 |
| l\_lesson\_kind | | 课程类别 | varchar | | 32 | 否 | | 否 |
| l\_lesson\_do | | 上课时间 | varchar | | 255 | 否 | | 否 |
| l\_lesson\_where | | 上课地点 | varchar | | 255 | 否 | | 否 |
| l\_student\_xueyuan | | 学生学院 | varchar | | 32 | 否 | | 否 |
| l\_lesson\_k | | 课程类别 | varchar | | 32 | 否 | | 否 |

表4-11学生信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否允许为空 | 是否主键 |
| uid | 学号 | int | 10 | 否 | 是 |
| name | 姓名 | varchar | 32 | 否 | 否 |
| sex | 性别 | varchar | 8 | 否 | 否 |
| xueyuan | 学院 | varchar | 64 | 否 | 否 |
| zhuanye | 专业 | varchar | 65 | 否 | 否 |
| class | 行政班 | varchar | 32 | 否 | 否 |
| grade | 年级 | varchar | 8 | 否 | 否 |
| year | 学制 | varchar | 8 | 否 | 否 |
| zynum | 专业代码 | varchar | 8 | 否 | 否 |
| type | 层次 | varchar | 32 | 否 | 否 |
| long\_num | 长号 | varchar | 12 | 否 | 否 |
| email | 邮箱地址 | varchar | 255 | 否 | 否 |

表4-12教师信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 描述 | 类型 | 长度 | 是否允许为空 | 是否主键 |
| tech\_id | 教师ID | int | 8 | 否 | 是 |
| tech\_name | 学院ID | varchar | 32 | 否 | 否 |
| token | 校验码 | varchar | 32 | 否 | 否 |
| password | 密码 | varchar | 32 | 否 | 否 |
| tech\_status | 状态 | int | 3 | 否 | 否 |

# **5 详细设计**

本章从系统设计的角度阐明考勤系统中各个组成模块的功能和具体的实现方式，通过模块流程图描述，通俗易懂描述了系统功能的业务流程。通过本章节，你将会更了解系统的功能模块业务逻辑和技术实现的方式。

## 5.1考勤客户端信息采集模块实现

### 5.1.1考勤客户端信息采集模块的流程图

考勤客户端信息采集模块流程图如图5-1所示。教师通过登录考勤客户端，进行考勤信息（如考勤课程、考勤班级）设置，即可开始扫描学生图书证上得条形码进行考勤，待考勤结束，提交考勤记录，即为一次有效考勤。

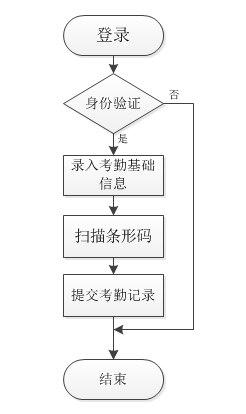


图5-1 考勤记录创建模块流程图

### 5.1.2考勤记录创建设计

通过5-2图中的登录操作验证后，即可进入图5-3所示界面，进行考勤基础信息的填写，考勤基础信息内容，会根据教师用户ID实时从考勤服务端获得有效的基础信息给用户选择。



图5-2 登录界面 图5-3 考勤基础信息界面

## 5.2条形码扫描模块实现

### 5.2.1条形码扫描模块流程图

客户端条形码扫描流程图如5-4所示。客户端调用摄像头设备，对条形码图

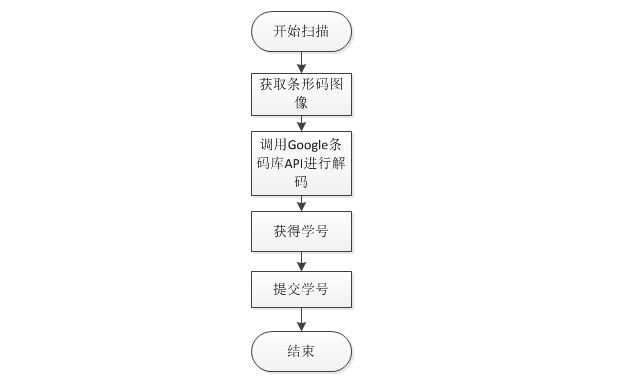


图5-4 条形码扫描流程图

案进行扫描，然后调用Google条形码库进行解码处理，得到学号信息，并将该学号提交至服务端。

### 5.2.2条形码扫描模块设计

客户端条形码扫描如5-5所示，用户点击“学号扫描”后，客户端调用摄像头设备，对条形码图案进行拍摄，如图5-5（a），并通过Google条形码库接口对拍摄的图片进行解码处理[8]，得到学号信息，如图5-5（a），并将该学号提交至服务端进行有效性验证。



（a）正在进行条形码扫描 （b）条形码扫描结果

图5-5 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。通过获取考勤记录ID，从数据库中取出对于数据，进行统计方面计算，输出用户期望的报表数据，并以图表的形式展示。

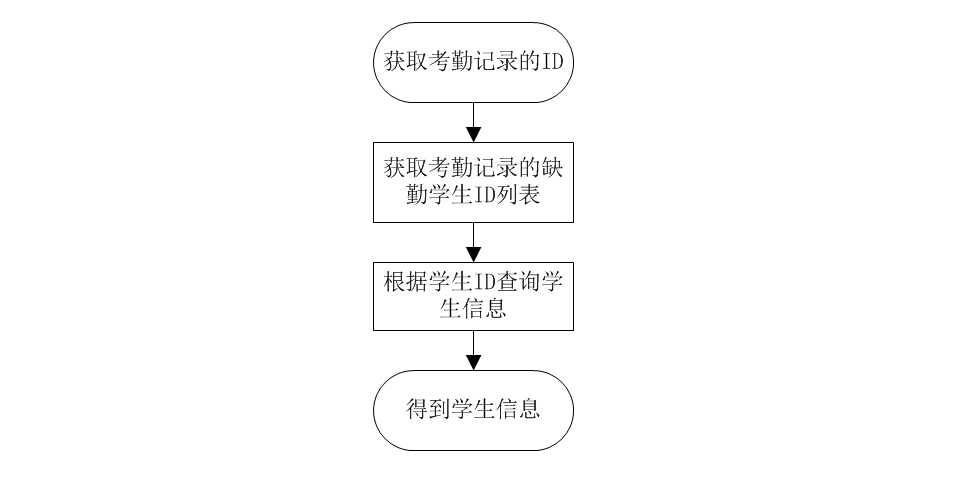


图5-5 缺勤用户统计流程图

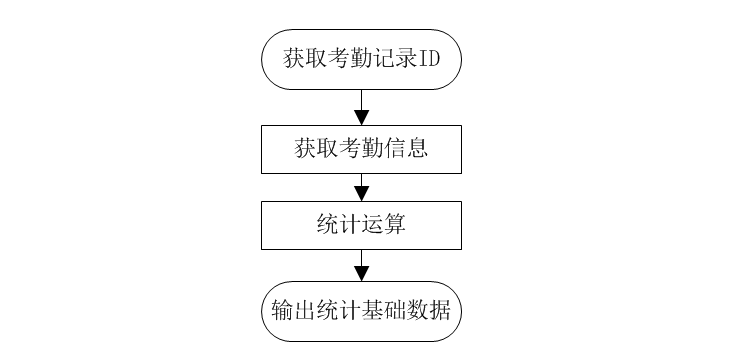


图5-6 缺勤比例统计流程图

### 5.3.2缺勤用户统计模块设计

本系统的考勤数据列表如图5-6所示。该功能模块提供所有考勤记录的查询、关键字检索、查看缺勤名单、查看缺勤比例、删除等功能。



图5-6 考勤数据列表

缺勤学生列表功能如图5-7所示。根据班级ID，从数据库中取出班级学生列表集合，与参加考勤学生集合作交集运算，便得到该次考勤该班级的缺勤学生名单。



图5-7 考勤数据缺勤学生列表

缺勤比例功能如图5-8所示。通过获得缺勤学生人数与班级总人数比例输出数据，以饼状图输出数据展示。

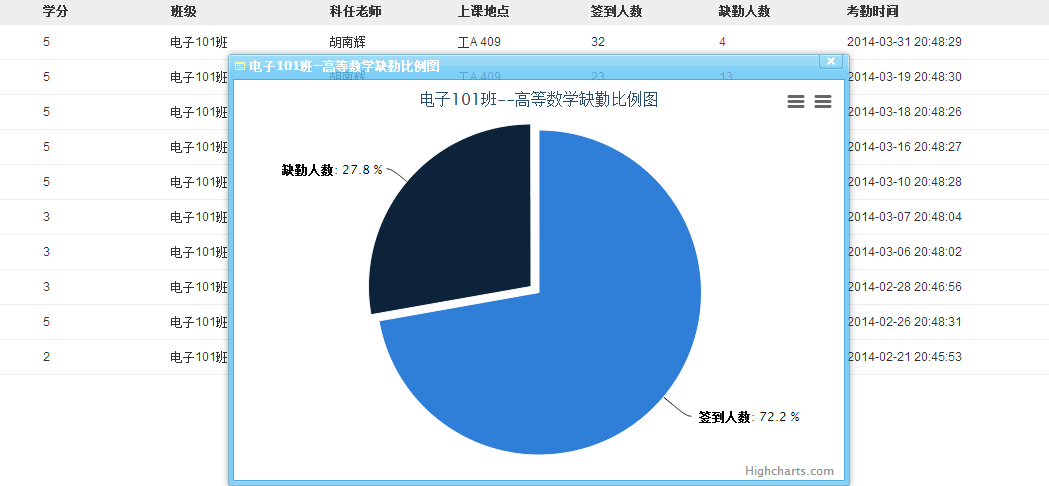


图5-8 考勤数据缺勤学生比例

## 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5-9所示。通过汇总班级所有考勤记录，输出缺勤比例的一个折线统计图，为班主任、教务处等查看班级缺勤率变化提供精确、有效、实时的数据，通过图表展示，更是加强了可读性。

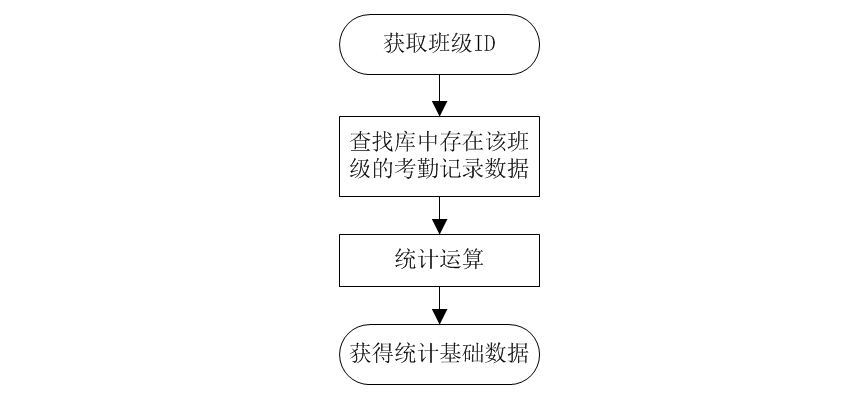


图5-9班级考勤数据模块流程图

### 5.4.2班级考勤数据设计

本系统的班级考勤数据模块如图5-11、5-12所示。



图5-11班级考勤数据列表

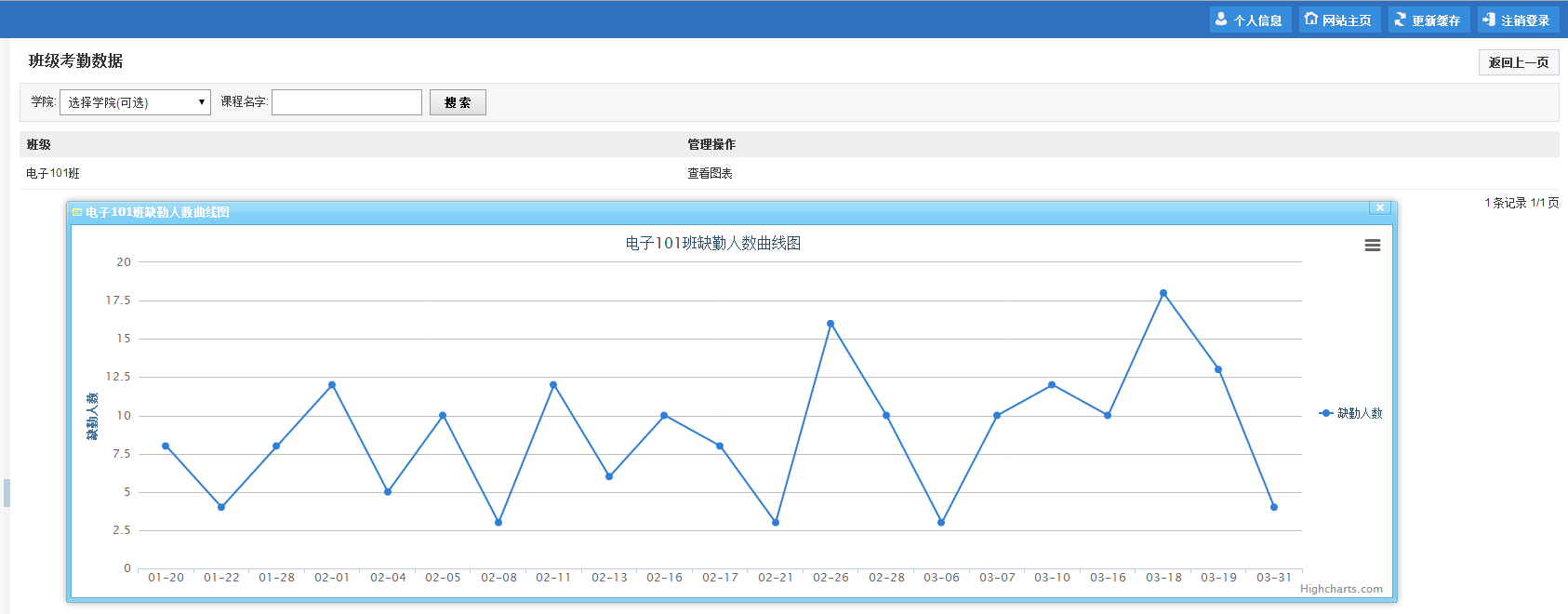


图5-11班级考勤数据折线图

# 6 系统的测试运行与维护

系统进行测试的目的是发现系统中存在的逻辑和物理上得错误，为开发人员及时修正软件提供依据，并且越早发现软件中的错误，对软件的质量越有利。系统测试应是贯穿于软件开发的整个开发周期，从需求分析阶段开始就要对需求文档进行测试[9]。

## 6.1系统的测试

### 6.1.1兼容性测试

经测试，本系统服务端可成功的在windows及Linux系统中运行，Android可运行在Android平台2.0以上，具有良好的兼容性。

### 6.1.2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。

### 6.1.3 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。

### 6.1.4 压力测试

为了保证系统在一定压力下能正常运行，需要知道系统的性能瓶颈，重要接口每秒的处理事务个数是多少，对于服务端设备以后是否需要扩容提供重要的数据参考。压力测试结果如图6-1所示。

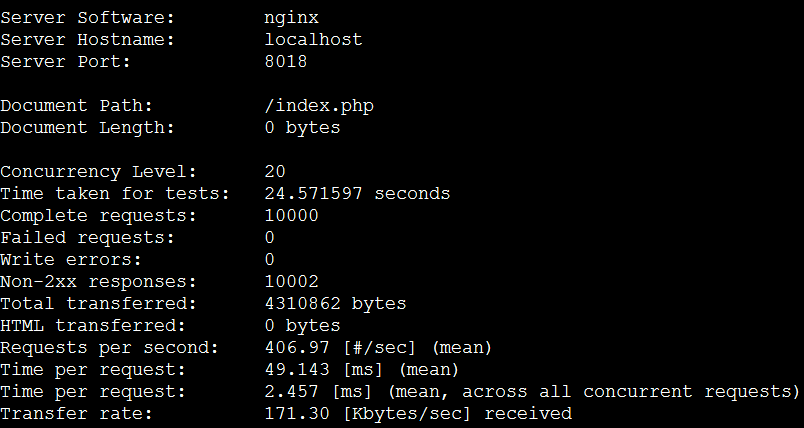


图6-1压力测试结果

系统重要接口TPS(Transaction Per Second，每秒处理事务数量)为407，符合性能要求。

## 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。

### 6.2.1．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。

### 6.2.2．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[10]。

### 6.2.3．系统维护的作用

系统维护的任务是改正软件系统在使用过程中发现的隐含错误，扩充在使用过程中用户提出的新的功能及性能要求，其目的是维护软件系统的"正常运作"。这阶段的文档是软件问题报告和软件修改报告，它记录发现软件错误的情况以及修改软件的过程[10]。

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，进一步明确了开发信息管理系统的方法和思想；既加深了对考勤方面的认识，又是对书本上所学知识的一个实践，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编程方面，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，出错处理不够完善等多方面问题。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未能找到最佳的方案进行解决，只是做了临时的处理，使业务使用正常。

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

[1]吴宇宁，电子技术与软件工程[J]，2013年第20期,软件工程的可行性研究.

[2]杨芙清，面向对象的系统设计[M],北京：清华大学出版社，2003年.

[3][李宁](http://searchb.dangdang.com/?key=&key2=%C0%EE%C4%FE&medium=01&category_path=01.00.00.00.00.00)，[Android开发权威指南](http://product.dangdang.com/product.aspx?product_id=22491464" \t "_blank" \o "Android开发权威指南)[M]，北京：机械工业出版社.2011年.

[4]高猗男，高校综合考勤管理系统的设计与实现[D],大连海事大学，2010年.

[5]凌质亿，刘哲星，曹蕾，计算机系统应用，2013年第06期，高并发环境下Apache与Nginx的I/O性能比较.

[6]肖维明，物流工程与管理[J],2009年第06期，基于PHP+Mysql的网站开发.

[7]萨师煊、王珊，数据库系统概论[M]，北京：高等教育出版社,2005年:1～121.

[8]李小立，车生兵，电脑编程技巧与维护[J]，2013年第02期，二维条形码识别技术研究.

[9]朱少民，软件测试方法和技术[M],北京：清华大学出版社，2007年.

[10]张海藩，软件工程导论[M]，北京：清华出版社，2003年，1-258.

# 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。在论文的写作过程中遇到了很多的困难和障碍，都在同学和老师的帮助下度过了。尤其要强烈感谢我的论文指导老师——杨久红老师，她对我进行了无私的指导和帮助，[不厌其烦](http://zhidao.baidu.com/search?word=不厌其烦&fr=qb_search_exp&ie=utf8)的帮助进行论文的修改和改进，在此向杨老师表示最衷心的感谢！

感谢这篇论文所涉及到的各位学者。本文引用了数位学者的研究文献，如果没有各位学者的研究成果的帮助和启发，我将很难完成本篇论文的写作。

感谢我的同学和朋友，在我写论文的过程中给予了我很多建议，还在论文的撰写和排版等过程中提供热情的帮助。

由于我的学术水平有限，所写论文难免有不足之处，恳请各位老师和同学批评和指正！