

北京师范大学珠海分校 本科生毕业论文

论文题目： 考勤管理系统设计与实现

学 院 信息技术学院

专 业 软件工程

学 号 0901030130

学 生 姓 名 杨健东

指导教师姓名 杨林

指导教师职称 高级工程师

指导教师单位 信息技术学院

2013 年 01 月 04 日

北京师范大学珠海分校学位论文写作声明和使用授权说明

学位论文写作声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：

日期： 年 月 日

学位论文使用授权说明

本人完全了解北京师范大学珠海分校关于收集、保存、使用学位论文的规定，即：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务；学校可以采用影印、缩印、数字化或其它复制手段保存论文；在不以赢利为目的的前提下，学校可以将学位论文编入有关数据库，提供网上服务。（保密论文在解密后遵守此规定）

论文作者签名：

导师签名：

日期： 年 月 日

信息技术学院考勤管理系统

摘要

信息技术学院考勤管理系统主要是提供一个具有良好用户体验的在线点名功能，帮助教师从繁琐的纸质工作转移到简单的电子工作上。利用计算机强大的数据处理能力，帮助老师统计大量复杂的考勤数据并制作出格式规范、内容清晰的报表。通过信息化和自动化的管理技术，把复杂的工作流程，重复的人工操作交给系统管理，提高所有考勤工作的效率和准确性，使教师的精力更多地集中在知识传授和提高教学质量工作上。本系统使用 JAVA 编程语言，MySQL 数据库，基于 SpringMVC 的框架进行开发，整个开发流程遵循软件工程基本的开发周期，一共进行了需求分析、概要设计、详细设计、数据库设计、编码实现、测试和部署 7 个阶段。上线一个学期后，经统计，任课教师在课前考勤的时间减少了 50% 以上，教务人员考勤统计的工作效率提高了 80% 以上。从这项研究成果可以得出，使用计算机系统能够提高教学管理的质量和效率。

关键词：考勤；管理系统；SpringMVC

样本

仅供参考

严禁抄袭

Rollcall Mangement System Of Information Technology Academy

ABSTRACT

The rollcall management system of information technology academy is mainly developed to provide a great user interface online rollcall function that helps course teachers to work on computer instead of working on paper. Making good use of the high data processing ability of computer, educational administrations can statistics large amounts of data and create some reports with a good format and clear content. Let the management system to complete the complex workflows and repetitive manual work using the technology of informatization and automation which can make the teachers to deveote themselves into teaching and improving the quality of educational. The system is developed by Java programming language, the main framework is SpringMVC and the database is MySQL. The development is base on the basic life cycle of software engineering. The 7 phases are requirement analysis, summary design, detail design, database design, coding, testing and deploying. We found that the cost time of rollcall before course is reduced by 50% and the statistics work efficiency is increased by 80% after a term passed. The research found that, the use of computer systems to improve the quality and efficiency of teacher management.

Key words: Rollcall; Mangement system; SpringMVC

目 录

1. 绪论.....	1
2. 需求分析.....	2
2.1 用户分析.....	2
2.2 功能性需求及系统用例分析.....	2
2.3 非功能性需求.....	5
2.3.1 可用性可靠性需求.....	5
2.3.2 性能需求.....	5
2.3.3 可支持性需求.....	6
2.3.4 实现需求.....	6
2.3.5 接口需求.....	6
2.3.6 打包需求.....	6
2.3.7 合法需求.....	6
3. 概要设计.....	7
3.1 软件结构概要设计.....	7
3.1.1 程序模块划分及其层次结构.....	7
3.1.2 程序模块间调用关系.....	7
3.2 数据结构概要设计.....	8
3.3 界面原型设计.....	9
4. 详细设计.....	17
4.1 系统体系结构设计及决策.....	17
4.2 系统顶层包结构.....	18
4.3 模块详细设计.....	18
4.4 数据库详细设计.....	28
5. 系统实现.....	30
5.1 模块代码实现.....	30
6. 单元测试.....	34
结语.....	37
参考文献.....	38
致 谢.....	39

仅供参考
严禁抄袭

1. 绪论

当前的大背景是数字信息技术与互联网的飞速发展，因此，愈来愈多的企业、学校和个人都会采用计算机和成熟的互联网技术对数据进行自动化和信息化管理。例如企业的 ERP 系统，学校的教务系统和个人的博客系统等，这些系统都有助于管理人员对他们所关注的数据进行实时的更新、有力的监控和以更好地用户体验展示数据，并且从一定程度上这些管理系统能够减少人工劳动，从而节省人力成本上的开销和降低人为错误率。当把这些系统应用于互联网的平台上，就能更好地让不同的角色参与其中，让他们能够随时随地通过简单的操作对数据进行查看、存储和分享等。

适逢于当前的信息潮流，并通过一定的调查，得出的结论是，当前国内仍有一部分的学校对于学生的考勤管理缺乏自动化与信息化的管理，他们大部分都采用人工点名、统计和录入的方式进行学生考勤的监管并以邮件和即时通讯技术发布考勤报表。有鉴于此，如何开发一个高效可用的自动信息化的考勤管理系统就是一个很有意义和价值的研究课题。通过深入地研究这个课题并且着力开发该系统，就能帮助教师提高教室的考勤工作的效率，帮助教务管理人员方便地统计考勤数据，并借助当前计算机高速的计算能力和成熟的智能化数据分析技术，便能清晰地分析出老师教授课程的教学质量和学生对开设课程的接受程度、感兴趣程度和学习质量。

本文具体内容包括五个部分，即绪论、需求分析、系统设计、系统实现和软件测试。

绪论部分阐述了研究课题的背景和研究目的，对论文大体的情况的概括和总结，有利于对整篇文章的把握。

第二部分是需求分析，主要对主要解决的问题进行详细的分析，明确用户的业务环境中软件系统应该做什么和怎么做，确定输入输出和基本业务流程，并且通过用例模型语言等业务分析工具和技术进行对系统业务的分析和建模，包括用户角色建模、系统用例建模与描述和对主要的非功能性需求进行说明。

第三部分是概要设计，主要是进行软件结构、数据结构和界面原型的概要设计。软件结构设计包括系统的模块层次结构划分和建立程序模块间的调用关系。数据结构设计包括数据特征描述、数据结构关系并建立概念数据模型。

第四部分是详细设计是对概要设计的细化，首先要设计每个模块的实现逻辑，通过类图的形式设计模块的局部数据结构和功能行类结构，然后细化数据库的设计，建立物理数据模型。使用设计模式的技术进行系统架构，确定使用的技术和框架

第五部分是系统实现，进行系统功能点的编码。

第六部分是单元测试，该部分描述了什么是单元测试，单元测试对开发的帮助和如何使用相关的技术进行高效的单元测试。

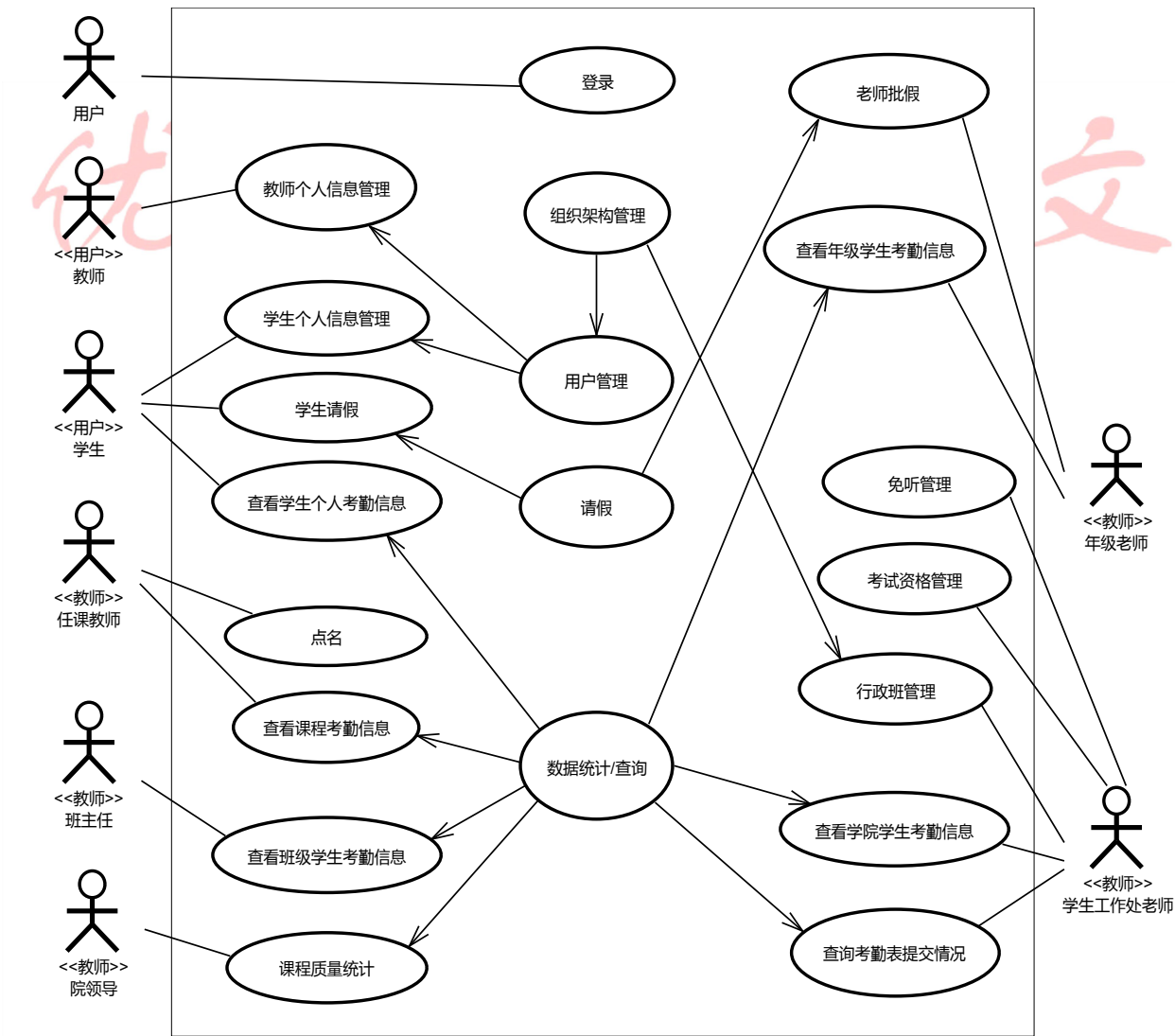


图 2 考勤管理系统用例图

主要系统用例详细分析如下表 1 至表 6 所示所示。

表 1 用例 UC1：教师点名

主要参与者：任课教师
前置条件：登录用户必须经过认证而且具有任课教师的角色。
后置条件：生成考勤表及考勤记录，准确记录每个学生的考勤类型。
基本流程： 1. 任课教师登录考勤管理系统并点击进入点名页面。 2. 任课教师选择需要进行考勤工作的课程安排。 3. 系统根据课程安排、学年学期和周获取考勤表，如果是第一次点名，则创建新的考勤表；否则，加载考勤表下的考勤项记录。 4. 系统查询当前课程的免听学生名单。 5. 系统查询当前课程的被取消考试资格的学生名单。 6. 系统查询当前课程的请假申请列表。 7. 教师点名，根据学生答到情况，可勾选该学生的考勤类型，分别有旷课、迟到和早退。迟到和早退可以双选，旷课和迟到、早退相互斥。如果学生已经免听、被取消考试资格或者请假申请已经批准，则该学生不需要参加当此考勤，选项按钮不可用。若全勤，则点击全勤按钮。

表 2 用例 UC2: 学生请假

主要参与者: 学生
前置条件: 登录用户必须经过认证而且具有学生的角色。
后置条件: 生成请假记录, 准确记录学生的请假项, 显示假条的当前状态为申请中。
基本流程: <ol style="list-style-type: none"> 1. 学生登录考勤管理系统并进入请假页面。 2. 系统根据周和学生查询。 3. 系统根据课程安排、学年学期和周获取学生课表。 4. 学生选择要请假的课程安排, 系统在已申请请假的课程列表中显示被选中的课程安排。已经申请请假的不能重复申请, 过期的课程安排不能进行请假申请。重新勾选, 则取消被选中的课程安排。 5. 学生填写请假理由并提交请假申请。

表 3 用例 UC3: 教师批假

主要参与者: 年级老师
前置条件: 登录用户必须经过认证而且具有年级老师的角色。
后置条件: 正确更改假条状态, 修改相对应的考勤记录。
基本流程: <ol style="list-style-type: none"> 1. 年级老师登陆考勤管理系统并进入批假页面。 2. 系统根据学年学期、周和假条状态查询假条列表。 3. 查看请假申请列表, 点击详细查看请假项。 4. 点击批准按钮表示该请假申请通过, 点击驳回按钮表示该请假申请不同过, 并填写审批备注。当假条的状态改变的时候, 要相应的改变对应学生的考勤记录。如果该生在课前成功通过请假审批, 则无需创建考勤记录; 否则, 要把旧的考勤类型重新标志为请假。

表 4 用例 UC4: 免听课程管理

主要参与者: 学生工作处老师
前置条件: 登录用户必须经过认证而且具有学生工作处老师的角色。
后置条件: 成功添加或取消免听记录, 正确地更改对应的学生选课信息。
基本流程: <ol style="list-style-type: none"> 1. 学生工作处老师登陆考勤管理系统并进入免听课程管理页面。 2. 系统根据学年学期查询免听记录列表。 3. 点击创建新记录按钮, 填写免听学生学号后, 系统查询并显示该生当前学年学期的已选课程, 教师可以选择一个或多个课程, 然后填写免听原因, 最后点击提交按钮提交免听记录。或者选中记录, 点击撤销。 4. 系统相对应地更改学生选课的信息。

表 5 用例 UC5: 考试管理

主要参与者: 学生工作处老师
前置条件: 登录用户必须经过认证而且具有学生工作处老师的角色。
后置条件: 成功添加或撤销取消考试资格记录, 正确地更改对应的学生选课信息。
基本流程: <ol style="list-style-type: none"> 1. 学生工作处老师登陆考勤管理系统并进入考试管理页面。 2. 系统根据学年学期查询取消考试资格记录列表。 3. 点击创建新记录按钮, 填写被取消考试资格的学生学号后, 系统查询并显示该生当前学年学期的已选课程, 教师可以选择一个或多个课程, 然后填写取消考试资格的

- 原因，最后点击提交按钮提交记录。或者选中记录，点击撤销。
4. 系统相对应地更改学生选课的信息。

表 6 用例 UC6：行政班管理

主要参与者：学生工作处老师
前置条件：登录用户必须经过认证而且具有学生工作处老师的角色。
后置条件：成功授权班主任或年级老师给行政班。
基本流程： <ol style="list-style-type: none"> 1. 学生工作处老师登陆考勤管理系统并进入行政班管理页面。 2. 系统根据学年学期查询行政班列表。 3. 选中班级，点击授权班主任，系统查询并显示出当前学年当前学期非班主任的教师名单，选中教师，点击授权，则该教师被任命为该班级的班主任；或者选中班级，点击授权年级老师，系统查询并显示所有教师名单，点击授权，则该教师被任命为该班级的年级老师。

除了上述主要功能需求，还有相关的数据统计与查询需求如下。

1. 学生查看个人考勤信息。
2. 任课教师查看课程考勤信息。
3. 班主任查看班级考勤信息。
4. 年级老师查看年级考勤信息。
5. 学生工作处老师查看考勤表提交情况。
6. 学生工作处老师查看全院学生考勤信息。
7. 院领导查看课程质量指标统计数据。

2.3 非功能性需求

2.3.1 可用性可靠性需求

考勤系统必须要在供电正常等正常情况下在学校上课的时间区间内能够正常运行并良好地执行所有功能点。在正常上课的时间内发生异常，例如断电、服务器挂机时等，必须能够在短时间内修复到正常运作而且保证数据的不丢失。系统的主要功能点要做到简单易用，并且要考虑到不同教室不同老师的差异情况，例如有的教室键盘的距离超出教师上课操作电脑时所能覆盖的正常范围，则要保证教师在使用鼠标的时候也能够很轻松的进行操作。

2.3.2 性能需求

用户对考勤系统的性能关注点主要在点名和信息统计查询的时候。点名的时候，系统要在短时间内呈现给用户事务正在执行或已经执行完毕，并允许用户不需等待事务执行完毕后才能继续执行下一个动作。在信息统计查询的时候，要使得在查询大量数据的时候仍能迅速的反馈信息给用户，例如使用分页来减少每页的数据量，或者使用异步数据加载或延迟加载等技术，使得信息是逐步呈现给用户，减少一次请求-响应的数据量，进而减少用户的等待时间，让用户对整个系统的整体性能感到满意。考虑到学院或学校服务器的数量有限，所以一台服务器可能会部署多个使用中的系统，所以在开发和部署的时候，必须时刻考虑到考勤系统的资源占用率。

2.3.3 可支持性需求

考勤系统应该是适用于所有学校的系统，学校中一般存在外籍教师，所以系统必须有国际化的支持，一般实现英文的国际化即可。不同学校的业务有一定的差异性，例如学生学号和老师教工号的位数等，这些都需要在系统初始化的时候让用户进行配置以适应特殊的业务需求。系统需要提供数据备份、数据导入等功能模块，数据导入也需要适应业务的差异性，在一些数据结构上，也要做到可配置的。

2.3.4 实现需求

因为在开发成本上的限制，本系统的开发基本是用开源工具和开发包进行开发。编程语言是 JAVA，基于 SpringMVC 框架和使用 Eclipse 集成开发工具进行开发。数据库使用的是 MySQL，服务器系统是 Ubuntu。

2.3.5 接口需求

考勤系统的基础数据是基于学校的教务管理系统的，因此教务管理系统必须提供一个数据访问接口，这个接口可以是一个有权限限制的数据库访问连接。考勤系统通过访问这个接口来获得源数据，并进行一定的整合，形成本系统的基础数据。

2.3.6 打包需求

后期的考勤系统开发上，需要考虑到一键打包和一键部署的功能，打包格式一般采用 .war。

2.3.7 合法需求

后期的考勤系统开发上，需要考虑到系统的授权方案，例如使用有时效的许可证。

3. 概要设计

3.1 软件结构概要设计

3.1.1 程序模块划分及其层次结构

考勤系统管理系统主要由访问控制模块和六大业务模块组成，业务模块包括点名模块，请假模块，数据统计模块，组织架构管理模块，课表管理模块和课业管理模块。点名模块实现教师点名功能。请假模块实现学生请假、教师批假。数据统计模块实现统计考勤数据功能，包括统计学生个人考勤数据、统计教学班考勤数据、统计行政班考勤数据、统计学院考勤数据、统计考勤表提交情况和统计教学质量数据。组织架构管理模块包括用户管理模块和行政班管理模块，用户管理模块又包括学生和教师的管理，该模块实现对用户的个人信息的管理，行政班管理模块实现班主任和年级老师的角色分配。课表管理模块，也称课程安排管理模块，实现教师和学生的课表管理以及课程安排有关信息的查询。课业管理实现学生工作处老师进行学生考试资格和课程免听的记录，有权取消旷课课时到达上限的学生的考试资格。程序模块层次结构如图 3 所示。

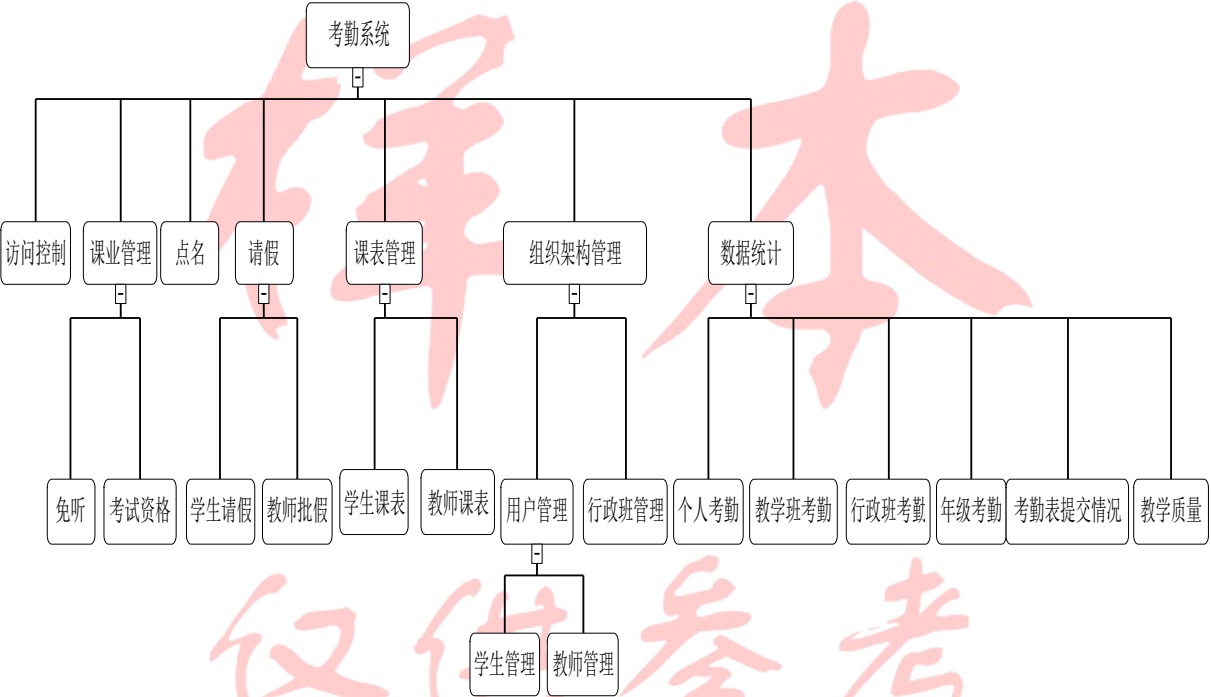


图 3 考勤管理系统程序模块层级结构

3.1.2 程序模块间调用关系

在系统分析的时候，考虑模块间调用的时候使用接口的方式往外暴露服务方法。如图 4 所示。接口定义如下：

1. 请假模块：定义假条查询接口。方法是根据课程安排和周来查询请假的学生名单。该接口是提供给点名模块使用。因为确定需要点名学生名单的时候，系统需要检查有哪些学生已经对当前课程安排申请请假并得到批准，以便在前台显示额外的信息

让教师注意到请假的学生和其他的一些业务控制。

2. 课表管理模块：

a) 定义课表查询接口，方法是当前周、当前学期和用户 id，取得课程安排。默认有两个实现，学生课表服务和教师课表服务，前者是列出学生的课程安排，后者是列出教师的课程安排。

b) 定义上课学生查询接口，方法是课程安排和周来查询上课学生。

3. 组织架构管理模块-用户管理模块：定义用户查询接口，根据学号查询学生，根据教工号查询教师。

4. 考勤模块：定义考勤更新接口。接收参数为学号、课程安排 Id 和周，当请假记录发生变更的时候，如学生的请假申请通过了审批，则需要触发考勤项的考勤类型的更新。

5. 课业管理模块：

a) 定义免听学生名单查询接口。考勤模块使用该接口的来查询免听学生的名单。

b) 定义被取消考试资格学生名单查询接口。考勤模块使用该接口的来查询被取消考试资格的学生名单。

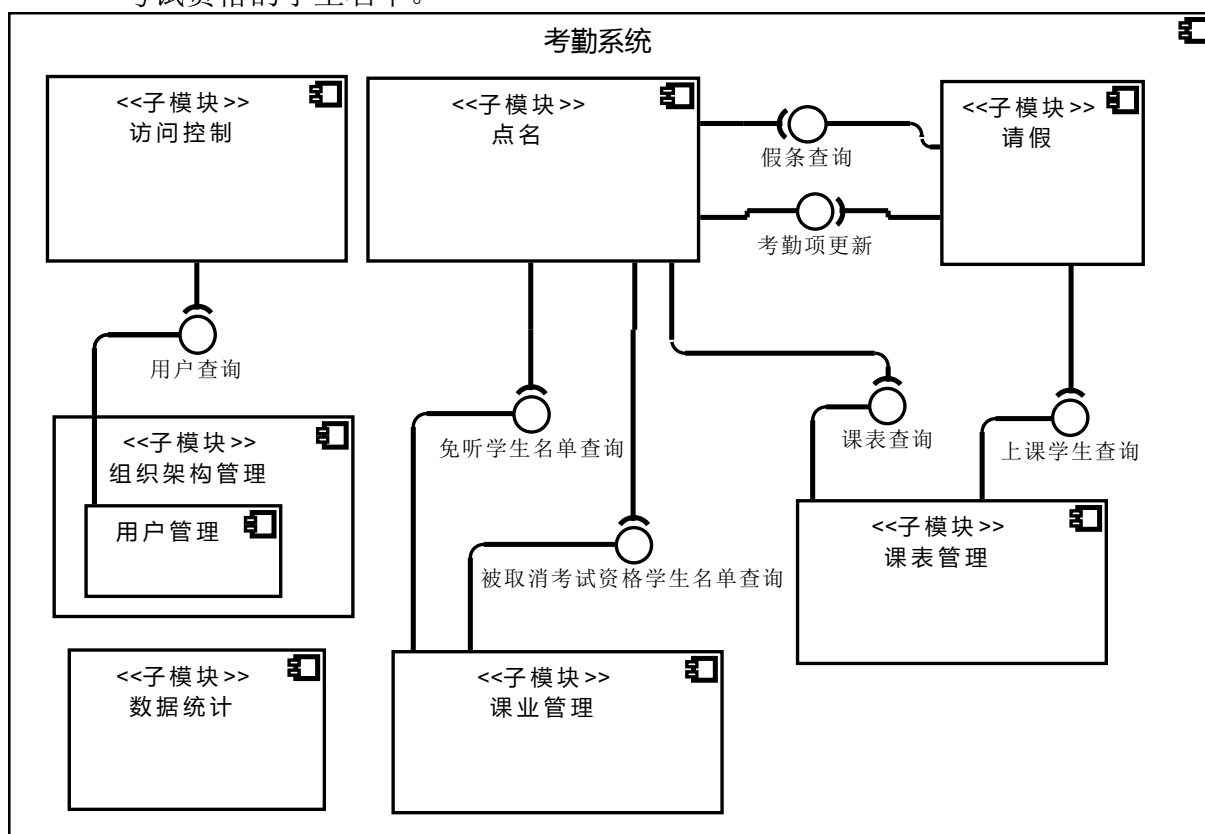
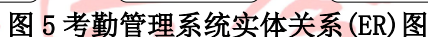


图 4 考勤管理系统程序模块间相互调用关系

3.2 数据结构概要设计

数据结构概要设计，即对考勤管理系统的领域模型的实体关系模型的分析与设计。系统的数据的内部模块是根据业务模块进行划分的，系统的实体关系 (ER) 图如图 5 所示。各领域模型的模块分布位置如下所示：

1. 组织架构模块：学生、老师、系、行政班。一个学生从属于一个行政班，一个行政班从属于一个系，行政班中分配有一个班主任（老师）和一个年级主任（老师）
2. 教学资管模块：课程、课程班、课程安排、专业、学年学期。课程班对应一个课程，



1. 登录页面

第 9 页 共 39 页

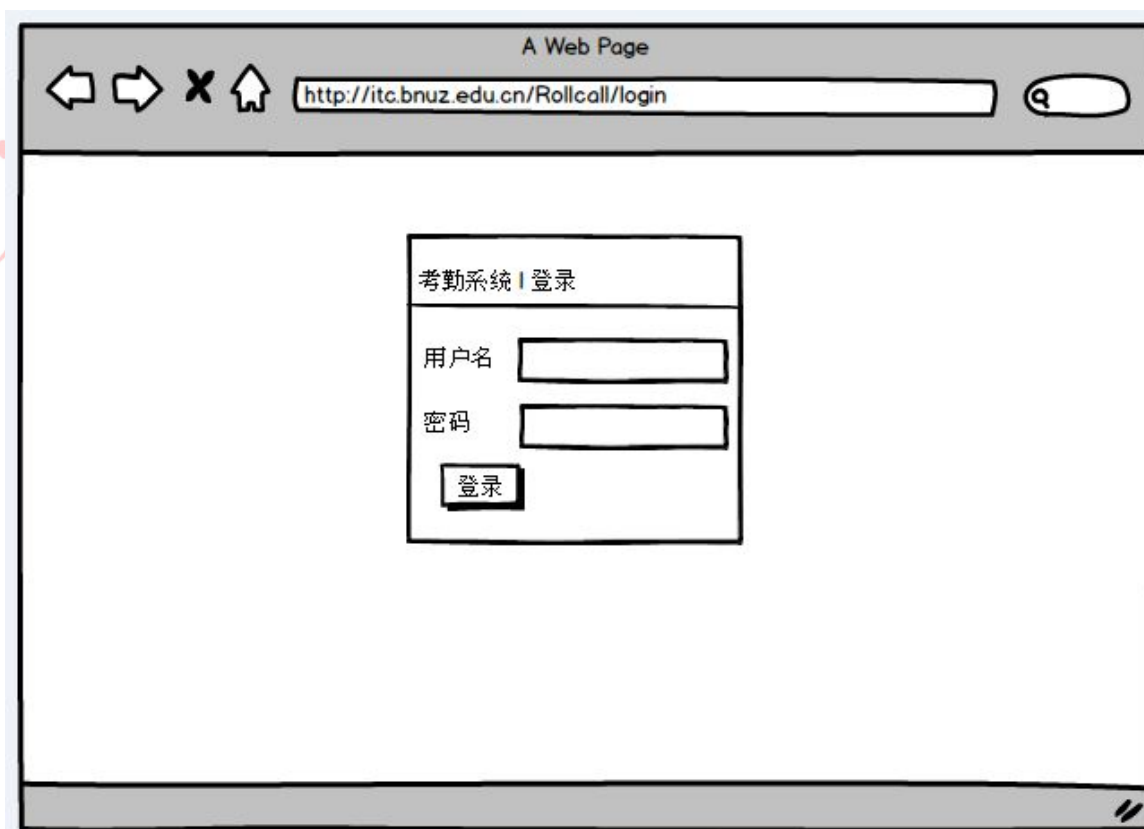


图 6 登录页面

2. 任课教师课表页面

此页面是任课教师的首页。首先包含一个菜单栏，菜单项分别有点名、教学班考勤情况查询、帐号设置和退出。菜单栏左边是考勤系统的商标。往下是一个下拉框，选择不同的周次，得到相应的课表数据。接着是一个表格控件，里面显示的是教师的课程在一个星期里的分布情况，即课表。设计图如图 7 所示。

仅供参考
严禁抄袭



图 7 任课教师课表页面

3. 任课教师点名页面

点击课表页面的其中一门课程，进入到点名页面。菜单栏下面有两个下拉框，第一个用于切换周次，第二个用于切换当前周的不同课程安排。接着是一个考勤表，考勤表的左起第一列有三个选择框，分别用于标志旷课、迟到和早退，表格的右半部分是考勤表的基础数据，如学生姓名、学号、历史考勤数据、专业和班级等。设计图如图 8 所示。

A Web Page

http://itc.bnuz.edu.cn/Rollcall/form/arrangementId/week

logo 首页 点名 教学班考勤情况 帐号设置 退出

第1周 周一 五, 六节 丽泽楼B301 C程序设计

考勤	姓名	学号	累计(旷)	专业	行政班
<input type="checkbox"/> 旷课 <input type="checkbox"/> 迟到 <input type="checkbox"/> 早退	张三	0901030130	9/1/0	软件工程	软件1班
<input type="checkbox"/> 旷课 <input type="checkbox"/> 迟到 <input type="checkbox"/> 早退	李四	0901030131	0/1/0	软件工程	软件1班

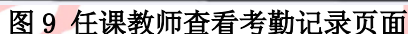
Copyright © 2012-2013 北京师范大学珠海分校信息技术学院 版权所有

图 8 点名界面

4. 任课教师查看考勤记录页面

菜单栏下方是下拉框，选择不同的学年学期，显示对应的考勤记录。接着是考勤记录表。第一列是每个课程安排的信息，第一至倒数第二列为周次，周次的最大值根据当前学年学期该任课教师的所有课程安排的最大周数来确定。可能是一至十八周，也可能是一至二十周。最后一列为累计，统计该行对应的课程安排的考勤记录统计。如 9/2/1/0，即共有 9 次旷课、2 次迟到、1 次早退和 0 次请假记录，这些记录分布在该课程安排的不同周次。设计图如图 9 所示。

仅供参考
严禁抄袭



点击菜单栏的请假菜单项,进入学生请假页面,菜单栏下方是一个请假申请表,申请人为当前登陆的学生用户,申请时间为当前时间,此两个输入框是设为不能更改,只供信息显示之用。接着是已选的课程列表和请假理由。请假申请表的下方是一个列表形式的课程表,学生选择不同的周次,列表显示对应的该学生的课表,勾选左边的选择框,即可添加请假课程,并对应的显示在已选课程列表中。学生确定请假申请信息无误后,点击提交申请按钮提交表单至系统。设计图如图 10 所示。

请假申请表

申请人 申请时间

已选课程列表

请假理由

第1周 ▼

<input type="checkbox"/>	课程	课室	时间	任课教师
<input checked="" type="checkbox"/>	C程序设计	综合楼A201	周二 第7、8、9节	杨林
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				

Copyright © 2012-2013 北京师范大学珠海分校信息技术学院 版权所有

图 10 学生请假页面

6. 年级老师批假页面

点击菜单栏的批假菜单项，进入批假页面。菜单栏下方有 3 个下拉框，它们作为请假记录列表的筛选条件，分别是学年学期、年级和假条状态，年级老师选择不同的筛选条件，点击查询按钮，系统查询出符合条件的请假申请记录。接着是请假申请记录列表，表的左半部分是申请记录的基本信息，右边最后一列是动作，分别有批准和驳回，年级老师点击批准则表示选中行的请假申请通过，点击驳回表示不通过。选中某一行并点击可以查看到每个申请表的申请项细节，包括课程列表和请假理由。设计图如图 11 所示。

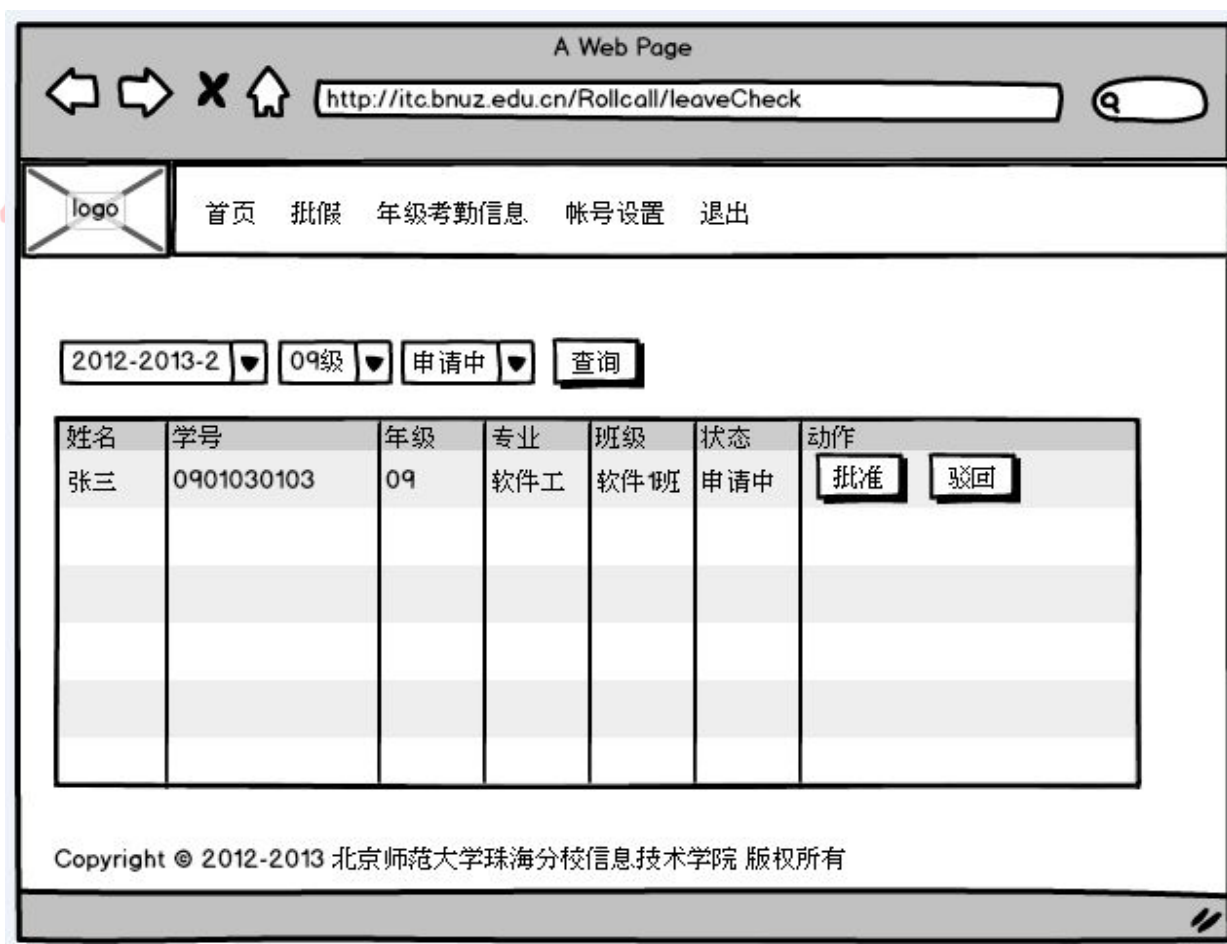


图 11 教师批假页面

7. 帐号设置页面

点击菜单栏的帐号设置项，进入帐号设置页面。帐号设置分为两个部分，个人资料和帐号安全，个人资料修改诸如联系电话和电子邮件等的基本信息，帐号安全是用于重置密码。设计图如图 12 和 13 所示所示。

仅供参考
严禁抄袭

A Web Page

http://itc.bnuz.edu.cn/Rollcall/account

logo 首页 点名 教学班考勤情况 帐号设置 退出

个人资料 帐号安全

联系电话(短号/长号) /

电子邮件

提交

Copyright © 2012-2013 北京师范大学珠海分校信息技术学院 版权所有

This screenshot shows the 'Personal Information' page of the account settings system. The browser window title is 'A Web Page' and the address bar shows 'http://itc.bnuz.edu.cn/Rollcall/account'. The navigation bar includes a logo and links to '首页', '点名', '教学班考勤情况', '帐号设置', and '退出'. The left sidebar has '个人资料' (Personal Information) and '帐号安全' (Account Security). The main content area contains two text input fields for '联系电话(短号/长号)' (Contact Number (Short/Long Number)) and '电子邮件' (Email), separated by a slash. A '提交' (Submit) button is at the bottom. The footer contains the copyright notice 'Copyright © 2012-2013 北京师范大学珠海分校信息技术学院 版权所有'.

图 12 帐号设置-个人资料页面

A Web Page

http://itc.bnuz.edu.cn/Rollcall/account

logo 首页 点名 教学班考勤情况 帐号设置 退出

个人资料 帐号安全

原密码

新密码

提交

Copyright © 2012-2013 北京师范大学珠海分校信息技术学院 版权所有

This screenshot shows the 'Account Security' page of the account settings system. The browser window title is 'A Web Page' and the address bar shows 'http://itc.bnuz.edu.cn/Rollcall/account'. The navigation bar includes a logo and links to '首页', '点名', '教学班考勤情况', '帐号设置', and '退出'. The left sidebar has '个人资料' (Personal Information) and '帐号安全' (Account Security). The main content area contains two text input fields for '原密码' (Original Password) and '新密码' (New Password). A '提交' (Submit) button is at the bottom. The footer contains the copyright notice 'Copyright © 2012-2013 北京师范大学珠海分校信息技术学院 版权所有'.

图 13 帐号设置-帐号安全页面

4. 详细设计

4.1 系统体系结构设计及决策

在开发工具上,考勤管理系统的后端主要采用Spring MVC框架进行开发.Spring MVC拥有Spring的依赖注入、Bean管理等优势,而且集成了流行的强大的Hibernate对象关系映射(ORM)框架,并且在Spring3.x很好地支持了RESTful的URL技术,因此系统的体系结构设计是基于SpringMVC框架。在数据库端,采用开源的、简单易用的和强大的MySQL数据库。系统前端采用了jQuery的JavaScript框架和Bootstrap的CSS框架。

考勤管理系统采用的是BS(浏览器—服务器)架构。在设计上,服务端使用了三层体系架构的设计模式,把整个系统的体系结构分为视图层、服务层和数据访问层。数据访问层使用Hibernate进行对象的持久化处理。服务层,即系统的业务模块层,按照系统的模块划分来架构服务层。视图层,在服务器端表现为控制器。前端与后端的数据交互上,采用比XML更加轻量级的JSON。为了降低服务器的压力和让用户得到更好的体验,会使用到AJAX进行数据的传送。系统前端使用FreeMarker模版引擎,模版引擎能够很好地降低前端与后端在开发上的相互依赖性,使得前端技术人员能够完全不需要编写后台代码,直接使用模版引擎提供的强大的标签即可。在访问控制层,使用了Spring Security框架,该框架简单引用并且具有很好的可配置性,考勤系统能根据自身的业务需求重写Spring Security,实现客户定制化的访问控制。考勤管理系统体系结构如图14所示。

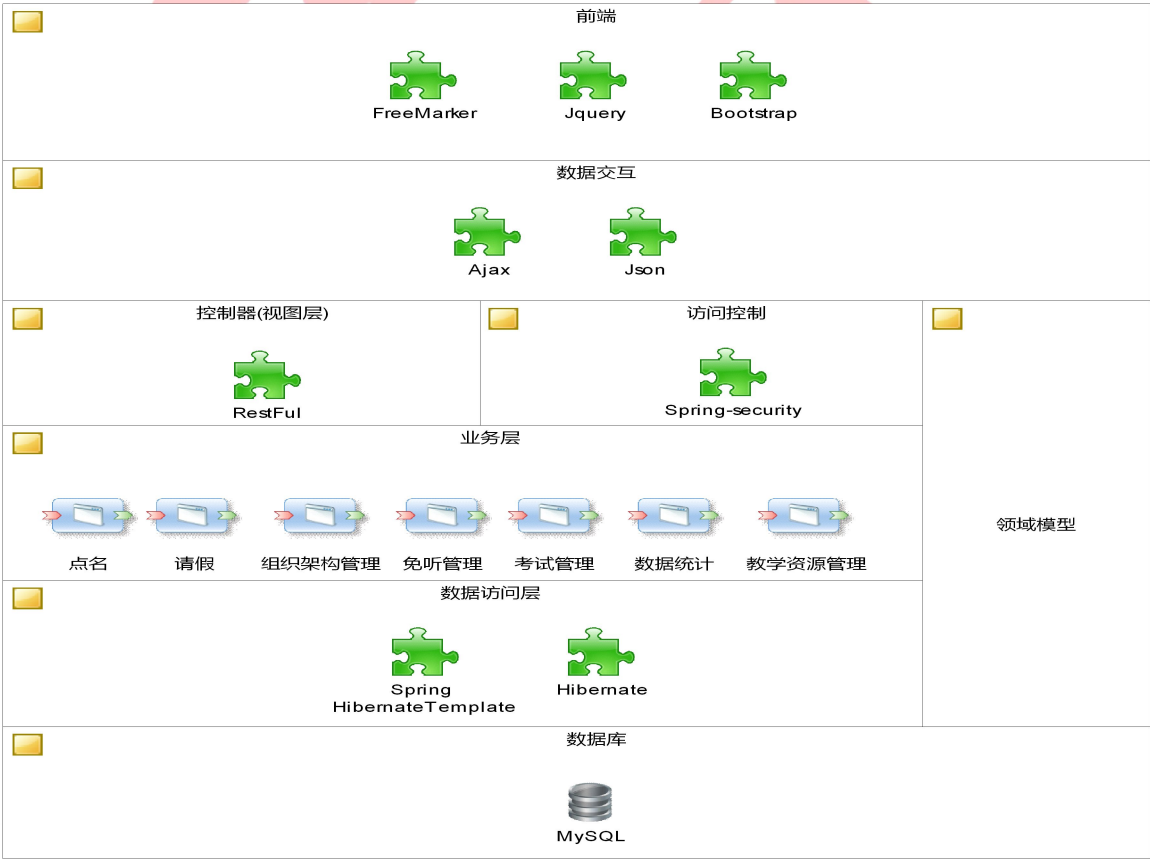


图 14 考勤管理系统体系结构

4.2 系统顶层包结构

系统顶层包结构按照 MVC、三层架构进行划分，Spring-security 作为系统访问控制模块单独建立一个包，util 工具包主要包含常用的日期工具类、json 数据转换工具类等。System 系统包主要包含除其他包外的系统帮助类，如菜单栏的构建类等。考勤系统顶层包结构如图 15 所示。

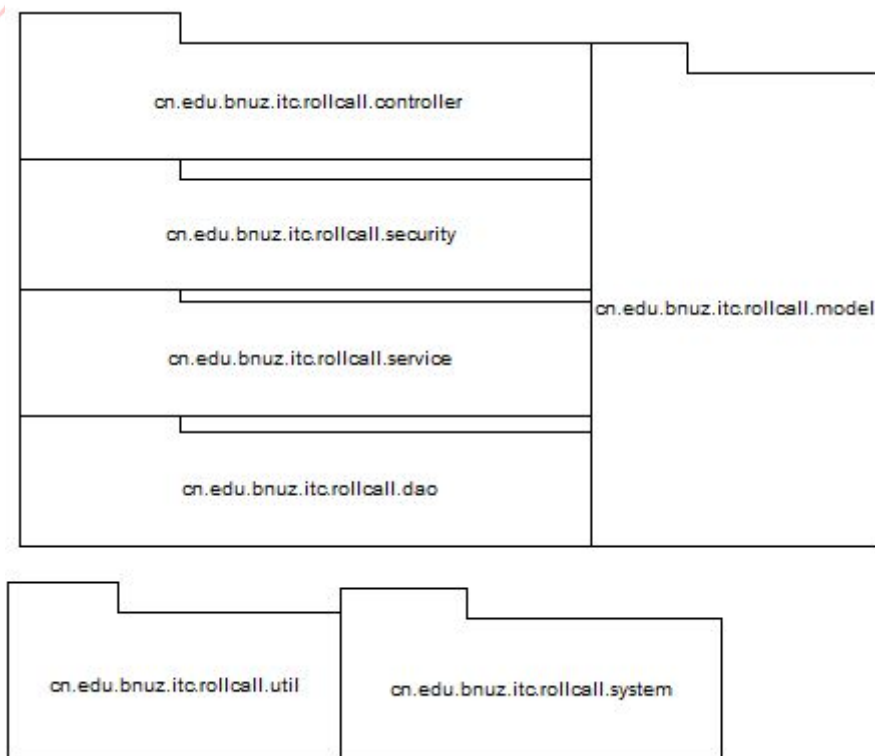


图 15 考勤系统顶层包结构

4.3 模块详细设计

在设计考勤管理系统类图的时候，必须考虑到 Hibernate 映射的问题，所以在实体关系图中某些类与类的双向关系将改成单向关系，在类级别上，体现于在多的一方反向关联一的一方。在一些多对多的关系上，为了代码实现的方便和查询的方便与高效，会建立中间表对象。例如学生与课程安排之间是多对多的关系，所以在数据库端会产生中间表，在类级别上，建立相应的中间表实体对象，引用学生与课程安排，使得数据访问层的查询代码更加精简，并且可以满足免听和考试资格管理的业务，也方便之后的扩展。考勤系统领域模型整体类结构关系如图 16。

以下是分模块描述考勤管理系统各模块的功能性类结构关系。

点名模块中主要的功能实现类是 RollcallService, 对 RollcallForm (考勤表) 的创建和获取, 对 RollcallItem (考勤记录) 的增删改, 这些数据的持久化操作都交给它们各自的数据访问对象 RollcallFormDao 和 RollcallItemDao 来完成。在点名模块中, 还需要获取参加考勤的学生列表、教师课程安排列表、免听的学生列表、被取消该门课考试资格的学生列表和该门课的所有申请中、驳回和批准的假条列表, 这些数据分别需要调用教学资源管理模块中的 ClassWorkService 和 ScheduleService。控制器 RollcallController 负责管理这一系列的服务, 调用它们的方法获取和组装数据。点名模块功能性类结构关系如图 17 所示。

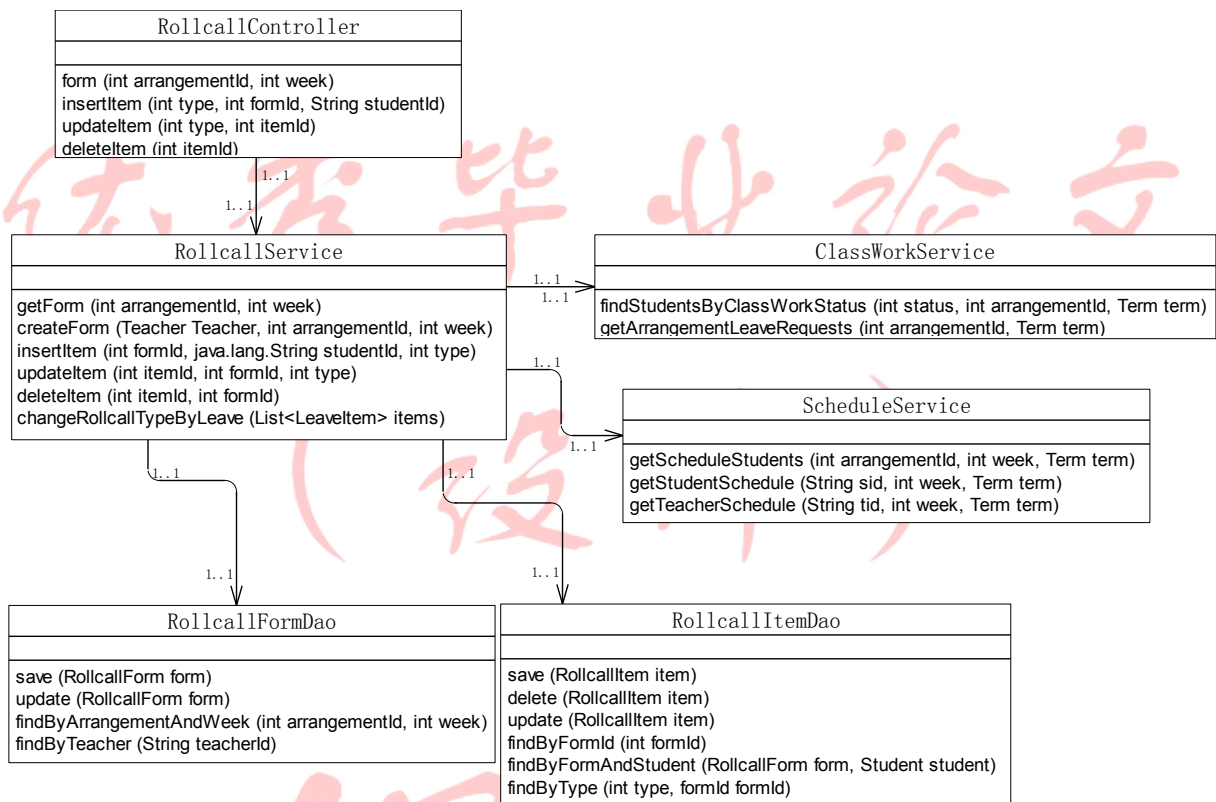


图 17 点名模块功能性类结构关系

考勤服务主要有显示考勤表、新建考勤项、更新考勤项和删除考勤项的方法。

显示考勤表的流程比较多，因为需要组织不同的数据以供前台展示之用。获取考勤表的时候，首先根据课程安排和周加载考勤表，若加载成功证明该教师并不是第一次进入同一时间同一个课程安排的点名页面。若加载失败，则证明该教师第一次参与该课程安排考勤工作，这时要新建一个考勤表并把基本的数据存入数据库，例如创建时间和创建人。获得考勤表后，查询所有有关的数据进行 json 数据的组装。显示考勤表的顺序图如图 18 所示。

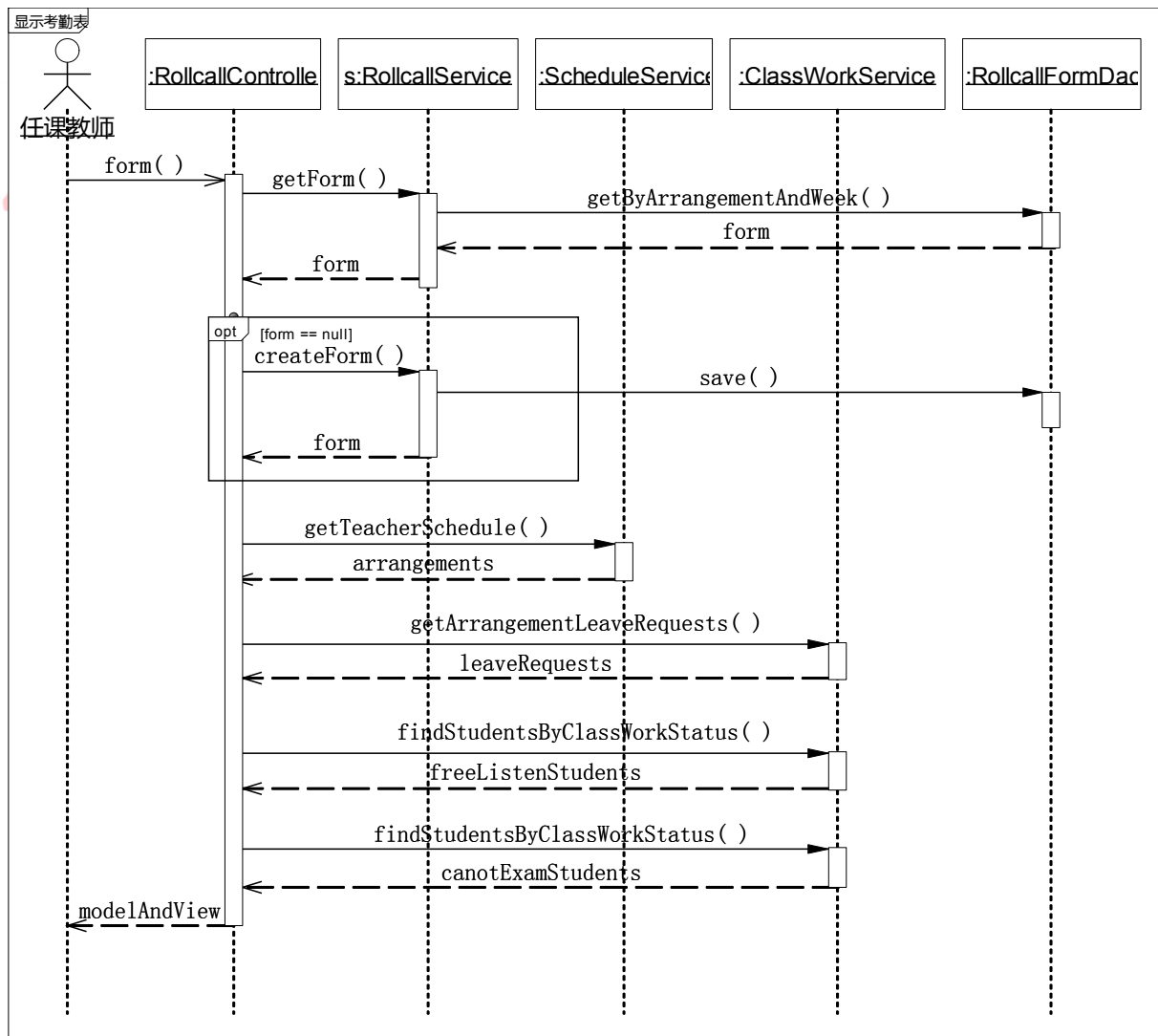


图 18 显示考勤表顺序图

新建、更新和删除考勤项的流程比较简单。从前台获取数据，建立对象，执行持久化操作。如果是新建考勤项，则需要返回考勤项，考勤项的 id 将会传到前台存储着，以便能够根据它的 id 进行更新和删除的操作。新建、更新和删除考勤项的顺序图如图 19、图 20 和图 21 所示。

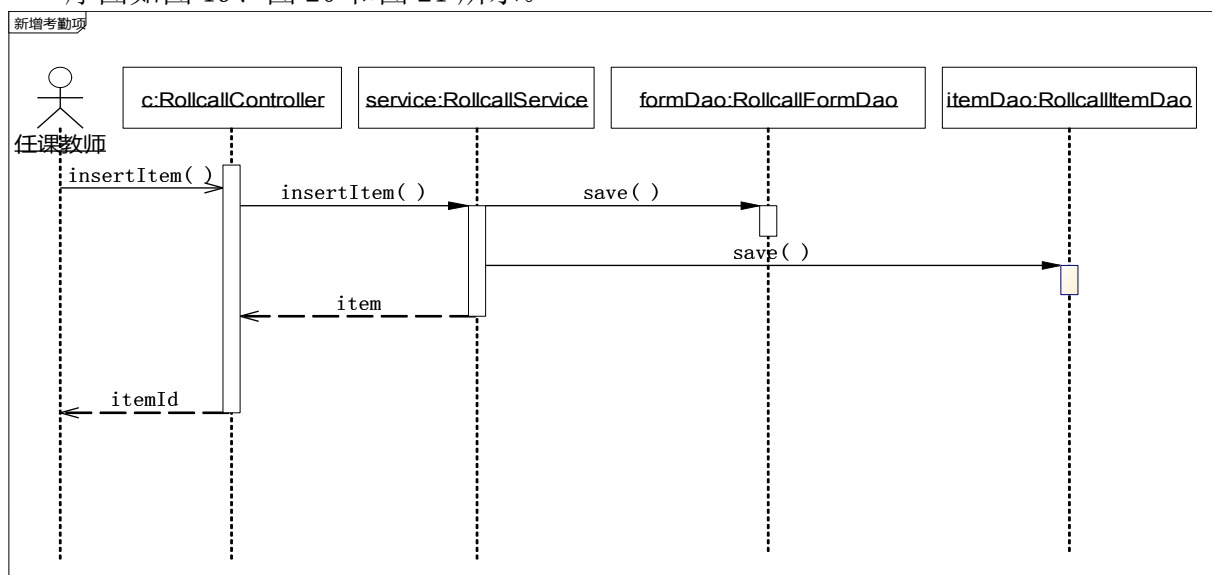


图 19 新建考勤项

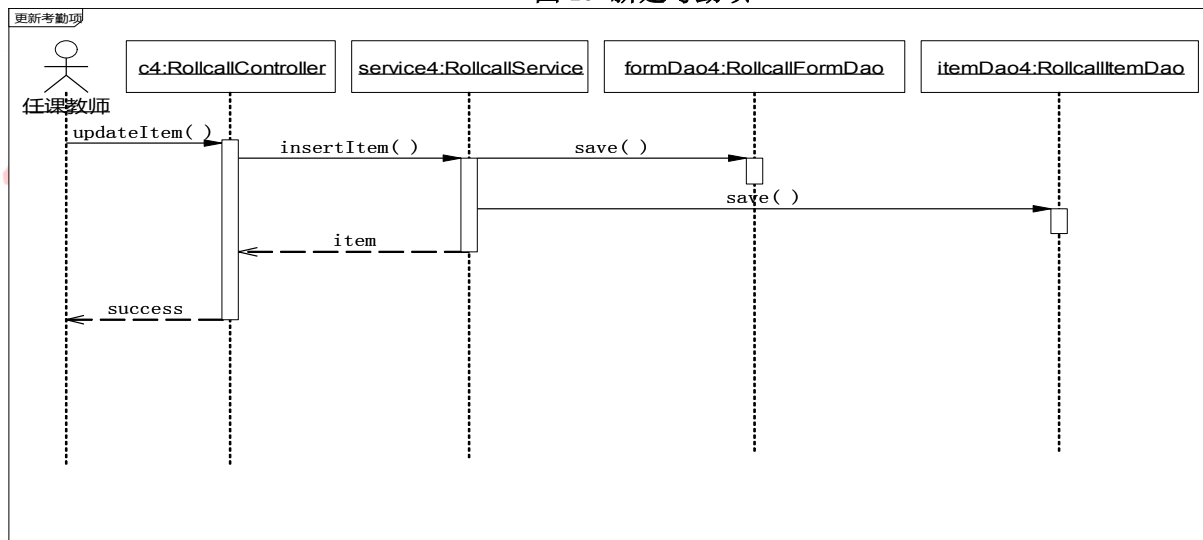


图 20 更新考勤项

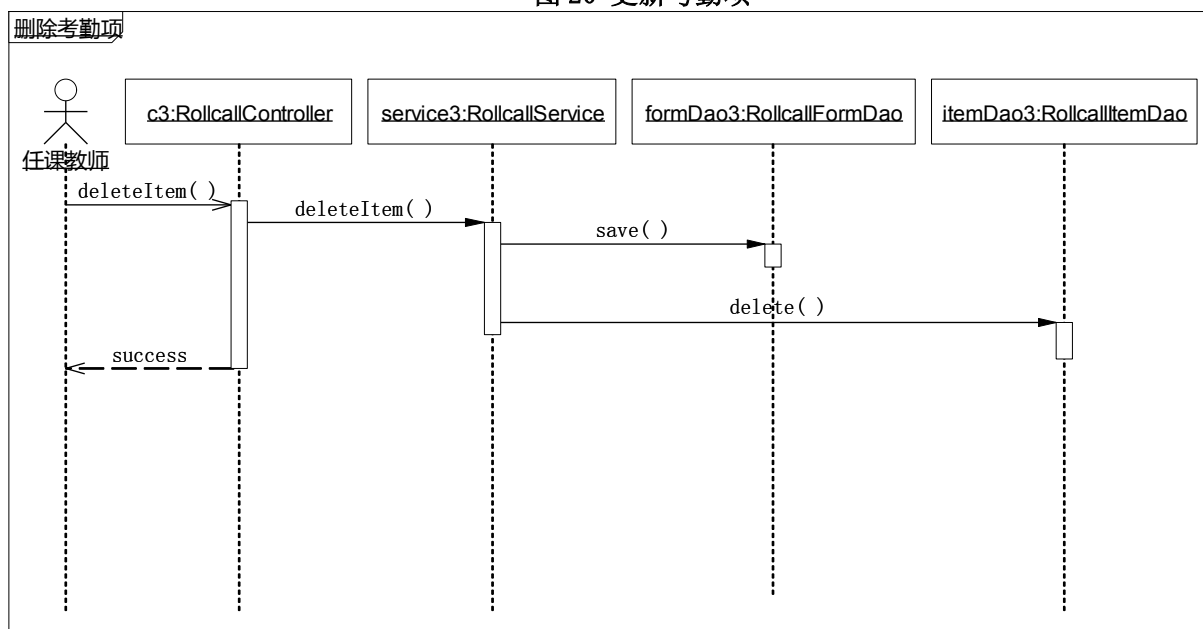


图 21 删除考勤项

2. 请假模块

请假模块中主要的功能实现类是 LeaveCheckService(教师批假服务), LeaveRequestService(学生请假服务)。学生请假模块包含两个个主要类、课程表服务接口和学生课表服务实现类, LeaveRequestController 和 LeaveRequestService。教师批假模块包含两个主要类、考勤项更新接口和考勤项更新服务实现类。数据持久交给 LeaveRequestDao(假条数据访问对象)和 LeaveItemDao(假条项数据访问对象)来处理。学生课表服务实现了获取学生课表的功能, 考勤项更新服务实现了当假条状态发生变化的时候, 触发对应的学生考勤记录的变化请假模块功能性类结构关系如图 22 所示。

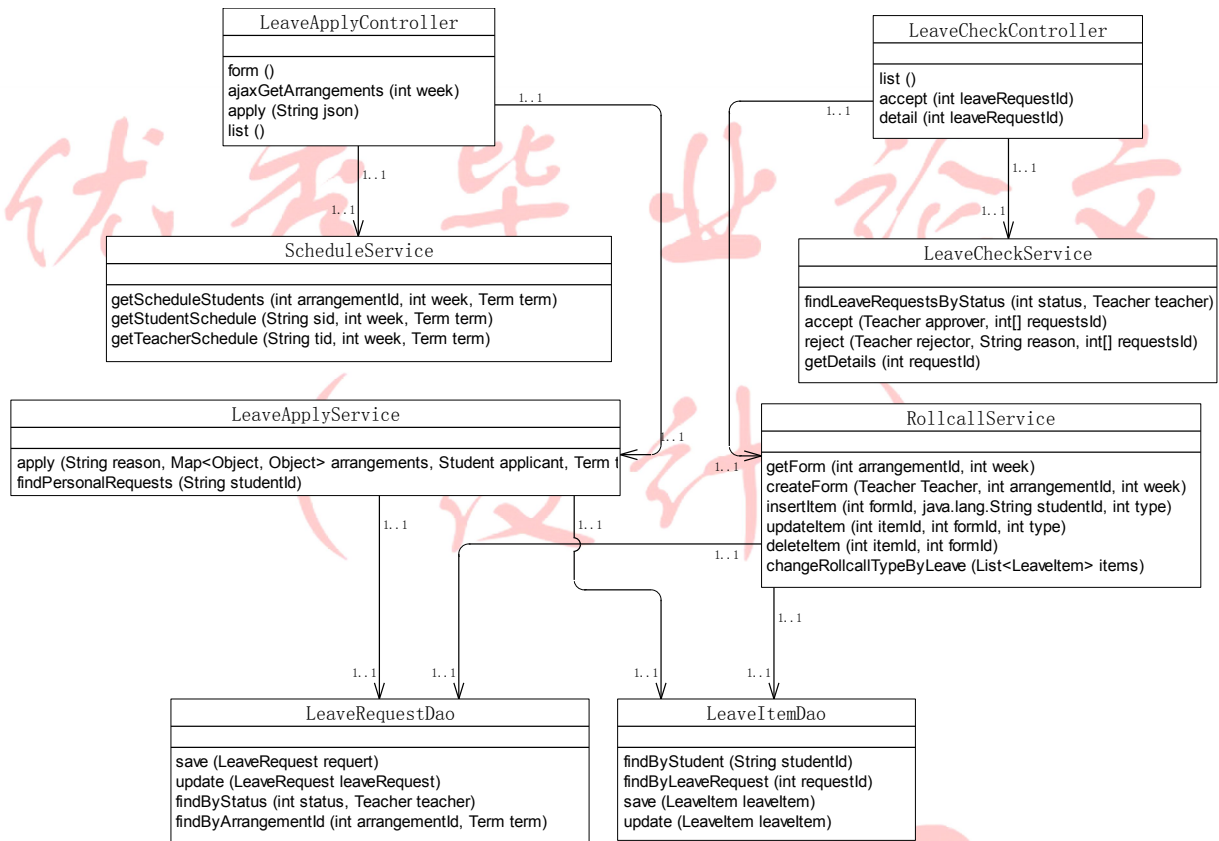


图 22 请假模块功能性类结构

学生请假流程简单，前台发送请假课程列表数据，请假理由数据到后台，后台把数据持久化到数据库。教师批假，当假条的状态发生变化的时候，调用RollcallService的changeRollcallTypeByLeave，改变对应学生的考勤项的更新。申请请假和老师批假的顺序图如图 23，图 24 所示。

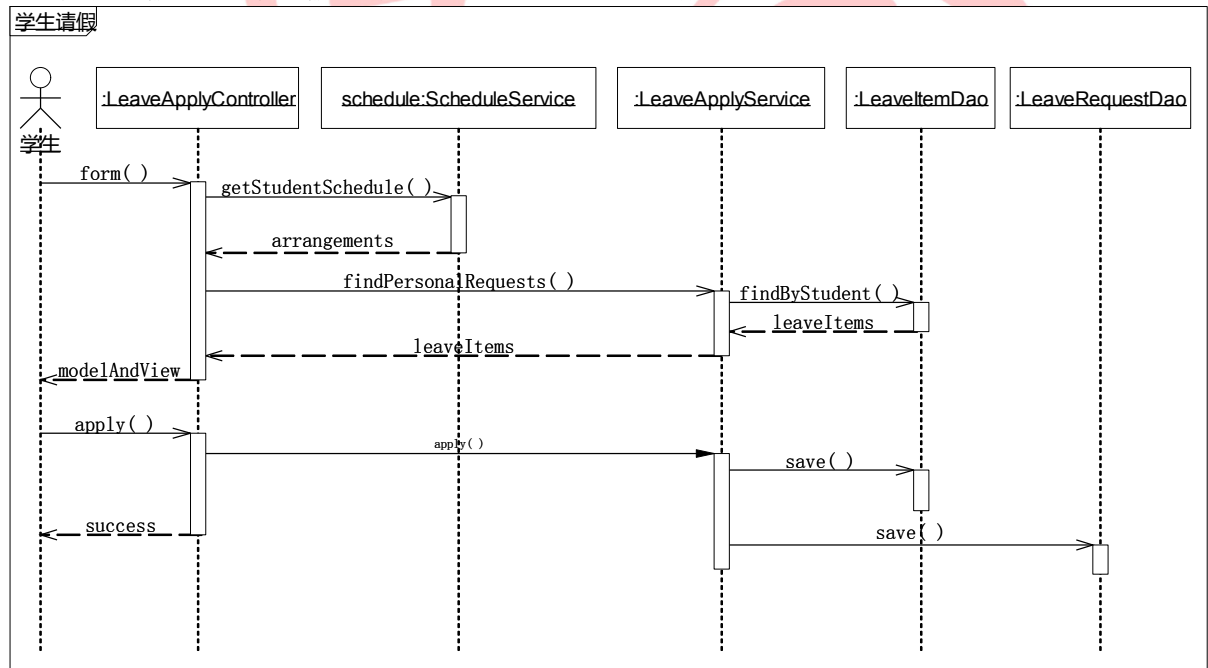


图 23 学生请假顺序图

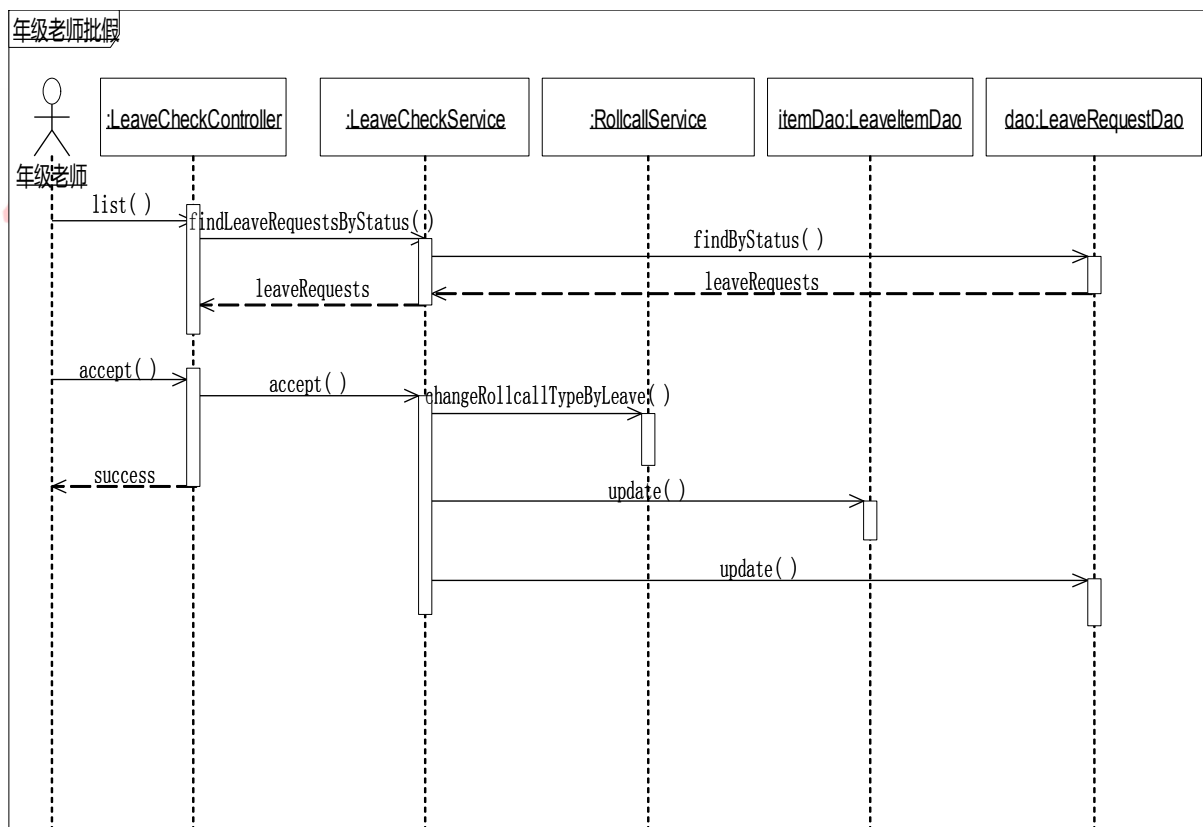


图 24 年级老师批假顺序图

3. 课业管理模块

课业管理模块主要由 ClassWorkController、ClassWorkService 两个主要类组成，这个模块实现对学生课程的管理。在 StudentCourseClass 类中，有一个 status 字段，提供两个标志属性分别是 FREE_LISTEN 和 CANOT_EXAM，表示免听和无考试资格的标志。课业管理管理功能性类结构如图 25 所示。

仅供参考
严禁抄袭

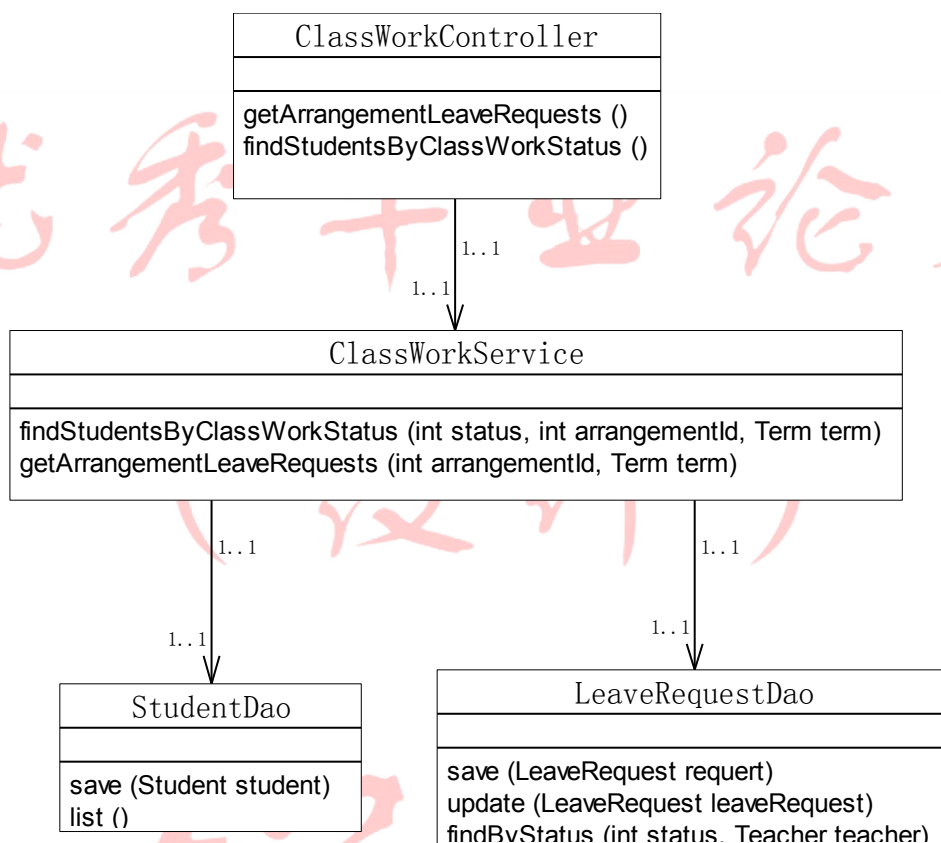


图 25 课业管理模块功能性类结构

4. 组织架构管理模块

组织架构管理模块包括行政班级管理、学生个人信息管理和教师个人信息管理。行政班级管理主要有 `AdminClassController`、`AdminClassService`、`AdminClassDao` 三个类。`AdminClassService` 实现了两个核心方法分别是 `grantAdminTeacher`（授权班主任）和 `grantGradeTeacher`（授权年级老师）。行政班级管理功能性类结构如图 26 所示。

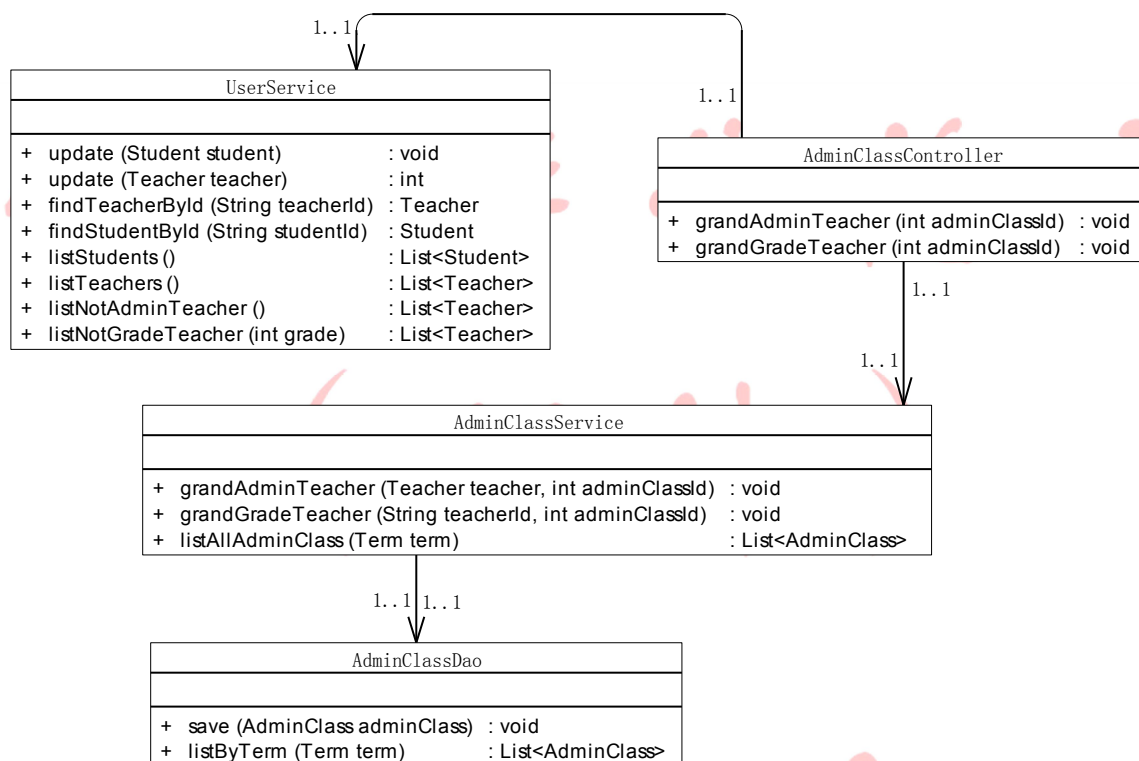


图 26 行政班管理功能性内结构图

授权班主任。调用 AdminClassService 的 listAllAdminClass 方法，把查询到的行政班列表发送到前台展示。学生工作处老师点击授权班主任按钮，界面弹出对话框，此时前台异步调用后台方法 listNotAdminTeacher，把还不是班主任的老师列出来。学生工作处老师选中其中一个老师，点击确定按钮，把 adminClassId 和 teacherId 作为参数传到后台，后台 AdminClassService 调用 grandAdminTeacher 方法，并将数据持久化到数据库中。授权年级老师。调用 AdminClassService 的 listAllAdminClass 方法，把查询到的行政班列表发送到前台展示。学生工作处老师点击授权年级老师按钮，界面弹出对话框，此时把选中的班级的所属年级作为参数，使用异步的方式调用后台方法 listNotGradeTeacher。把还不是年级老师和还未分配到其他年级担任年级老师的老师列出来。学生工作处老师选中其中一个老师，点击确定按钮，把 adminClassId 和 teacherId 作为参数传到后台，后台 AdminClassService 调用 grandGradeTeacher 方法，并将数据持久化到数据库中。

用户管理主要由 StudentController、TeacherController、UserService、StudentDao 和 TeacherDao 五个类组成。其功能性类结构如图 27 所示。

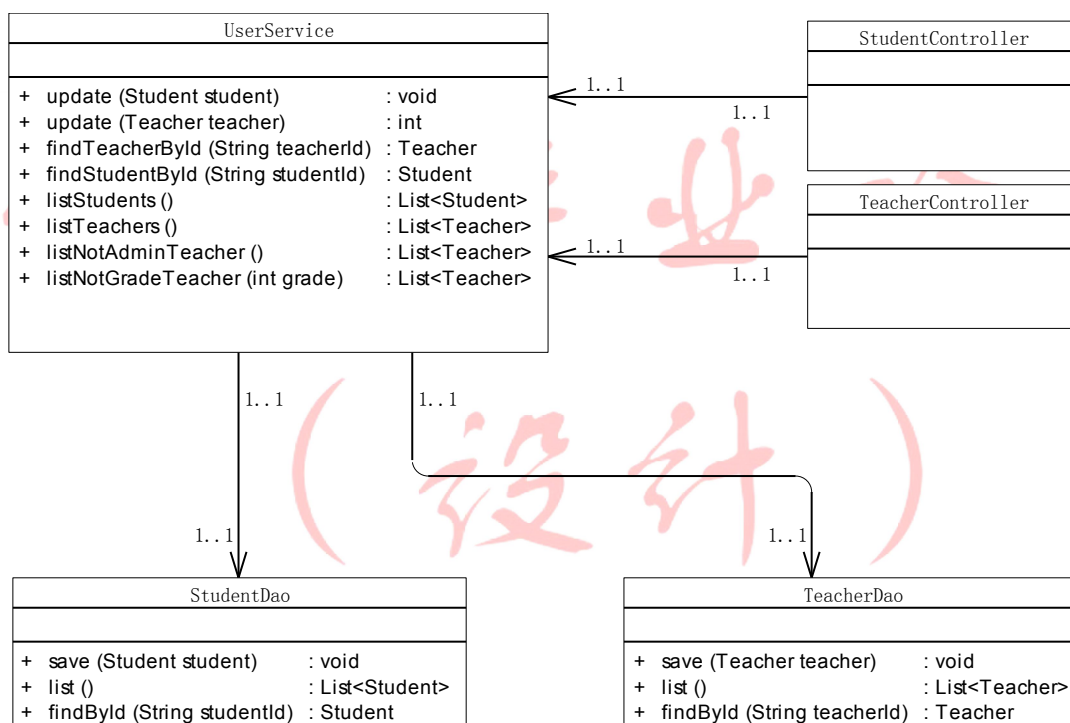


图 27 用户管理模块功能性类结构

5. 课表管理模块:

课表管理模块主要实现学生和老师的课表的查询功能。该模块主要由 ScheduleController 和 ScheduleService 两个类组成。ScheduleService 中提供了 getStudentSchedule 和 getTeacherSchedule 来分别获取学生和教师的课表。其功能性类结构如图 28 所示。

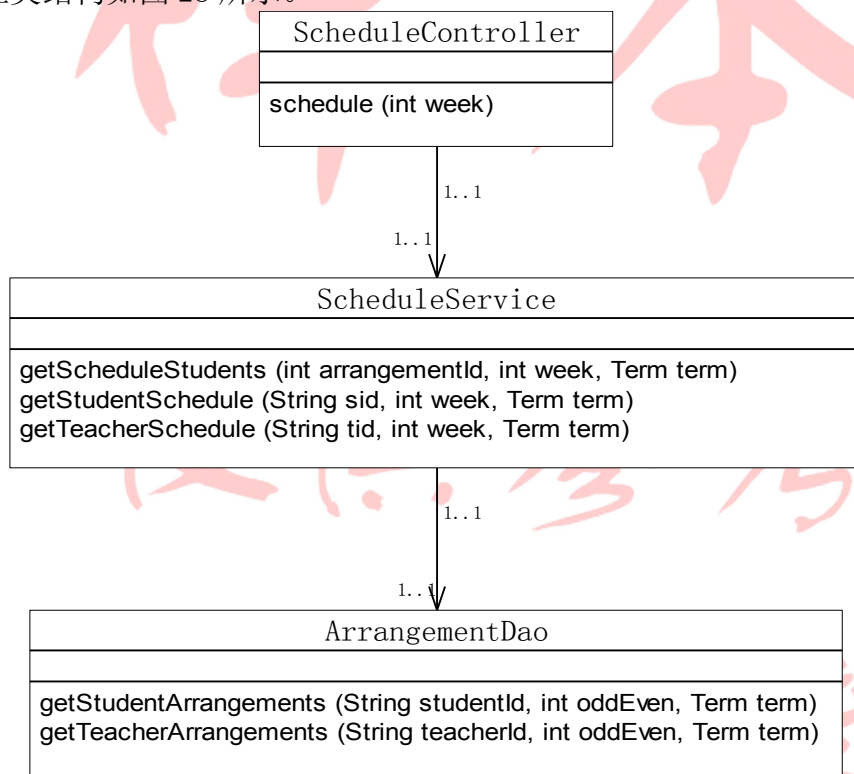


图 28 课表管理模块功能性类结构

4.4 数据库详细设计

通过工具把概要设计的数据结构设计中所得出的实体关系（ER）图生成物理数据模型（PDM）图。1、考虑到系统编码的时候，采用联合主键的方式将一定程度上增加持久层对象映射的设置复杂度，所以中间表都增加一个无意义的主键，把两个外键设置同时唯一约束。2、考虑到数据统计的时候需要连接许多表，并且查询出来的原始数据显示格式并不满足界面的需求，因此会增加系统编码时的复杂度，因此在数据库端，采取视图和触发器的方式进行这方面的优化。例如考勤数据的统计，参与过考勤活动的每个学生都会有旷课累计、迟到累计、早退累计和请假累计的数据，把这些数据组成一个视图或表，采用触发器的形式，监听考勤项表的增删改，触发统计视图或表的数据变更。3、对于复杂的统计查询，将采用存储过程的方式，这样能够降低系统编码的难度并且提高查询的效率和速度。4、考虑到一些数据的查询进场会用到某些字段作为条件，所以为这些字段建立索引能够提高查询的效率。物理数据模型如图 29 所示。

样本
仅供参考
严禁抄袭

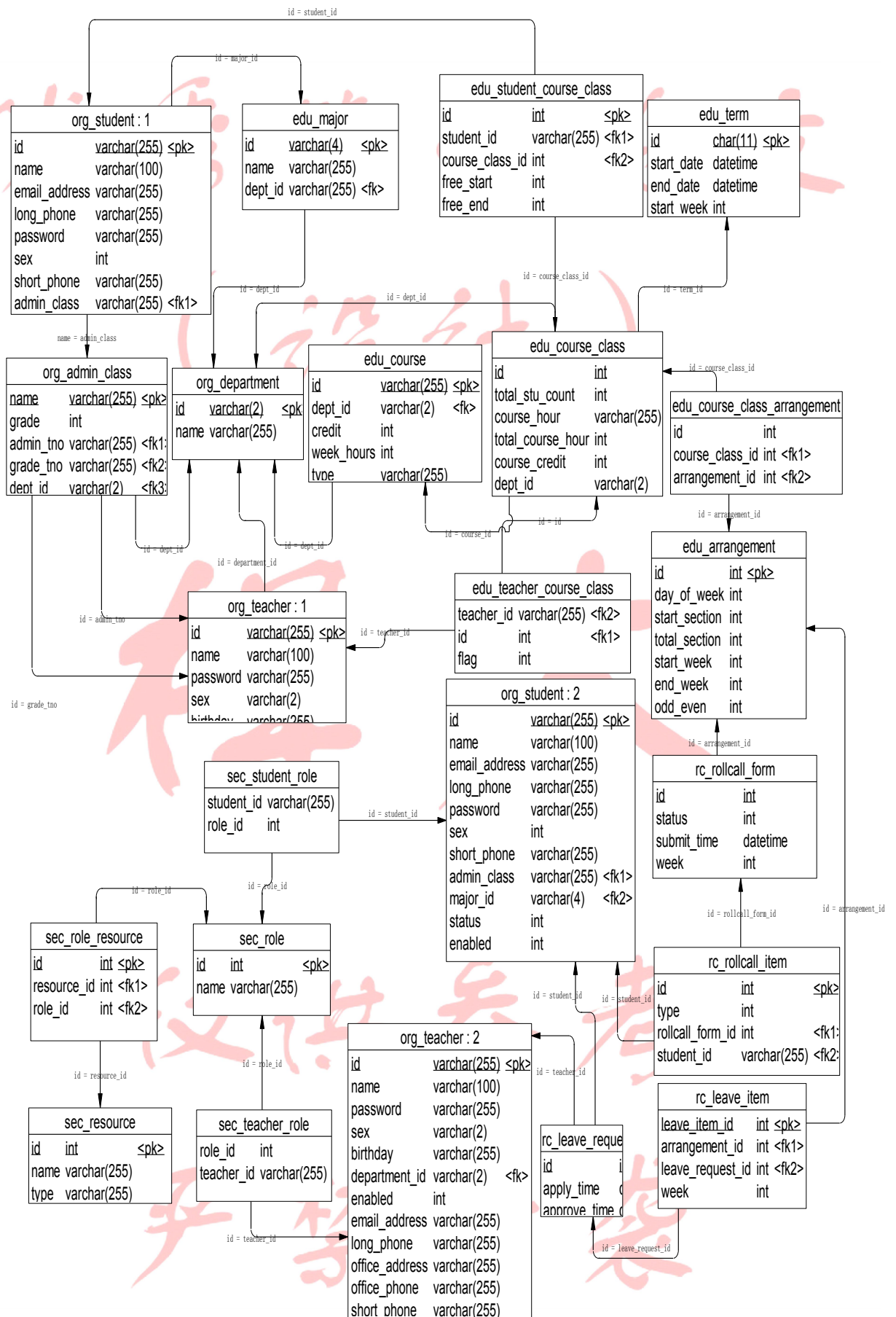


图 29 考勤管理系统数据物理模型

5. 系统实现

5.1 模块代码实现

分模块进行编码。

1. 点名模块

a) 显示考勤表

form 方法首先从 SecSecurityContext（安全性上下文）中获取当前登录的教师，然后加载考勤表，当加载的考勤表为空时，创建一个新的考勤表，然后通过调用其他 service 的方法获取需要的数据，然后通过调用 JsonUtil 的 buildJson 方法，传入需要被转成 json 的对象，组成 json 格式的字符串。通过 @RequestMapping(value = "/form/{arrangementId}/{week}") 的方式实现具有 Restful 风格的 url。主要代码如表 7 所示。

表 7 显示考勤表主要代码

```
@Controller
@RequestMapping(value = "/rollcall")
public class RollcallController {
    ...
    @RequestMapping(value = "/form/{arrangementId}/{week}")
    public ModelAndView form(@PathVariable("arrangementId") int
arrangementId, @PathVariable("week") int week) {
        Teacher teacher = SecSecurityContext.getTeacher();
        Term term = termService.currentTerm();
        RollcallForm form = rollcallService.getForm(arrangementId,
week);
        if (form == null) {
            form = rollcallService.createForm(teacher, arrangementId,
week);
        }
        List<RollcallItem> items =
rollcallService.getItemsByForm(form);
        List<Student> rollcallStudents = arrangementService
        .getArrangementStudents(arrangementId, week, term);
        List<Arrangement> teacherSchedule = teacherScheduleService
        .getCourseSchedule(teacher.getId(), week, term);
        List<LeaveRequest> leaveRequests = leaveRequestLookUp
        .getArrangementLeaveRequests(arrangementId, term);
        List<Student> freeListenStudents = freeListenLookUp
        .getFreeListStudents(arrangementId, term);
        List<Student> noExamQualiStudents =
examLookUp.getNoExamQualiStudents(
arrangementId, term);
    }
}
```

```

        Map<String, Object> jsonMap = new HashMap<String, Object>();
        jsonMap.put("form", form);
        jsonMap.put("items", items);
        jsonMap.put("leaveRequests", leaveRequests);
        jsonMap.put("teacherSchedule", teacherSchedule);
        jsonMap.put("rollcallStudents", rollcallStudents);
        jsonMap.put("freeListenStudents", freeListenStudents);
        jsonMap.put("noExamQualiStudents", noExamQualiStudents);
        String json = JsonUtil.buildJson(jsonMap);
        ModelAndView mv = new ModelAndView("rollcall/form.ftl",
"jsonObj", json);
        return mv;
    }
    ...
}

public class JsonUtil {
    public static String buildJson(Object obj) {
        ObjectMapper objectMapper = new ObjectMapper();
        String json = "";
        try {
            json =
objectMapper.defaultPrettyPrintingWriter().writeValueAsString(obj);
        } catch (JsonGenerationException e) {
            e.printStackTrace();
        } catch (JsonMappingException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return json;
    }
}

```

b) 新增，更新和删除考勤项

新增，更新和删除考勤项，主要是通过 jquery 的异步发送请求数据到后台，后台进行持久化处理并返回新增考勤项的 id 到前台。当请求处理成功并返回到前台后，前台改变相应 html 控件的显示内容。\$.post() 就是 jquery 提交数据的函数，参数是 Url 和请求数据，返回 onResult。后台控制器 insertRollcallItem 方法，接收从前台异步传过来的数据，调用 RollcallService 对应的方法，把数据持久化到后台，然后返回 rollcallItemId 到前台。主要代码如表 8 所示。

表 8 考勤项增删改主要代码

```
Form.js
$(document).ready(function(e) {
...
RowInfo.prototype = {
...
    insertItem : function(status, onResult) {
        $.post('/RollCall/rollcall/insert', {
            rcFormId : $("#rollcallFormID").val(),
            stuNum : this.stuNum,
            rcType : status
        }, onResult);
    },
    updateItem : function(status, onResult) {
        $.post('/RollCall/rollcall/update', {
            rcFormId : $("#rollcallFormID").val(),
            stuNum : this.stuNum,
            rcType : status
        }, onResult);
    },
    deleteItem : function(onResult) {
        $.post('/RollCall/rollcall/delete', {
            rcFormId : $("#rollcallFormID").val(),
            stuNum : this.stuNum
        }, onResult);
    }
...
}
...
}

RollcallController.java
public class RollcallController {
...
@RequestMapping(value = "/insert", method = RequestMethod.POST)
public @ResponseBody
String insertRollCallItem(
    @RequestParam(value = "type") int type,
    @RequestParam(value = "formId") int formId,
    @RequestParam(value = "studentId") String studentId) {
    RollcallItem item = rollcallService.insertItem(formId,
        studentId, type);
    return JsonUtil.buildJson(item.getId());
...
}
```

2. 请假模块

a) 年级老师批假

年级老师批假的主要代码集成在系统根据假条的审批状态更新对应学生的考勤记录的考勤类型。changeRollcallTypeByLeave, 传入参数是年级老师审批的假条所包含的假条项目。主要代码如表 9 所示。

表 9 年级老师批假主要代码

```
@Service("rollcallService")
public class RollcallService {
...
    public void changeRollcallTypeByLeave(List<LeaveItem> items) {
        for(LeaveItem leaveItem : items) {
            Student applier = leaveItem.getLeaveRequest().getApplier();
            int week = leaveItem.getWeek();
            Arrangement arrangement = leaveItem.getArrangement();
            RollcallForm form = getForm(arrangement.getId(), week);
            if(form != null) {
                RollcallItem rcItem =
rollcallItemDao.findByFormAndStudent(form, applier);
                boolean isRcItemNull = false;
                if(rcItem == null) {
                    isRcItemNull = true;
                    rcItem = new RollcallItem();
                    rcItem.setForm(form);
                    rcItem.setStudent(applier);
                }
                if(leaveItem.getLeaveRequest().getStatus() ==
LeaveRequest.STATUS_APPROVED) {
                    rcItem.setType(RollcallItem.TYPE_LEAVE);
                } else if(leaveItem.getLeaveRequest().getStatus() ==
LeaveRequest.STATUS_REJECTED) {
                    if(isRcItemNull) {
                        rcItem.setType(RollcallItem.TYPE_ABSENT);
                    }
                }
                rollcallItemDao.save(rcItem);
            }
        }
    }
...
}
```


6. 单元测试

对每一个需要测试的类建立单元测试，测试其中的每一个方法，测试方法的时候，必须保证每个测试方法互不依赖。方法中的数据验证采用断言的方式。[9]编写代码时，我们总是会做出一些假设，断言就是用于在代码中捕捉这些假设。断言表示为一些布尔表达式，程序员相信在程序中的某个特定点该表达式值为真，可以在任何时候启用和禁用断言验证，因此可以在测试时启用断言而在部署时禁用断言。同样，程序投入运行后，最终用户在遇到问题时可以重新启用断言。使用断言可以创建更稳定、品质更好且易于除错的代码。当需要在一个值为 FALSE 时中断当前操作的话，可以使用断言。单元测试必须使用断言（JUnit/JunitX）。除了类型检查和单元测试外，断言还提供了一种确定各种特性是否在程序中得到维护的极好的方法。使用断言使我们向按契约式设计更近了一步。在测试的时候不使用实际的数据库（MySQL），选择使用了内存数据库 h2db，然后配置测试数据的位置，测试数据采用 SQL 脚本的方式。例如测试 StudentDao 的 findById 的时候，首先建立 SQL 测试脚本，脚本里面编写插入语句，运行的时候，h2db 引擎会执行测试脚本的 SQL 语句，建立测试数据。

在测试 http 请求的时候，需要用到 mock 的测试思想。mock 测试，[9]就是在测试过程中，对于某些不容易构造或者不容易获取的对象，用一个虚拟的对象来创建以便测试的测试方法。这个虚拟的对象就是 mock 对象，它是真实对象在调试期间的代替品。使用 Mock Object 进行测试，主要是用来模拟那些在应用中不容易构造（如 HttpServletRequest 必须在 Servlet 容器中才能构造出来）或者比较复杂的对象（如 JDBC 中的 ResultSet 对象）从而使测试顺利进行的工具。并使用相关的开源类库，使用他们已经建好的 mock 对象。例如在测试 Controller 的时候，为了模拟 HttpServletRequest 就需要用到 MockHttpServletRequest，这样就不需要在 web 的环境下才能进行测试，有利于前台和后台的独立开发和测试。建立 MockServletContext、MockHttpServletRequest 和 MockHttpServletResponse 对象，就能很好地模拟了一次从前端发送请求到后台，后台处理好返回数据的一个测试流程，而不需要运行 web 服务器去测试一个业务，有效地提高了测试效率。主要代码如表 10 所示。

表 10 使用 Mock 测试 Controller 主要代码

```
public class JUnitControllerBase {
    private static HandlerMapping handlerMapping;
    private static HandlerAdapter handlerAdapter;
    private static XmlWebApplicationContext context;

    @BeforeClass
    public static void setUp() {
        if (handlerMapping == null) {
            String[] configs = { "classpath:test-applicationContext.xml" };
            context = new XmlWebApplicationContext();
            context.setConfigLocations(configs);
            MockServletContext msc = new MockServletContext();
            context.setServletContext(msc);
            context.refresh();
            msc.setAttribute(
```

```

        WebApplicationContext.ROOT_WEB_APPLICATION_CONTEXT_ATTRIBUTE,
            context);
        handlerMapping = (HandlerMapping) context
            .getBean(DefaultAnnotationHandlerMapping.class);
        handlerAdapter = (HandlerAdapter) context
            .getBean(context

        .getBeanNamesForType(AnnotationMethodHandlerAdapter.class)[0]);
    }
}

public ModelAndView excuteAction(HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    request.setAttribute(HandlerMapping.INTROSPECT_TYPE_LEVEL_MAPPING,
true);
    HandlerExecutionChain chain = handlerMapping.getHandler(request);
    ModelAndView model = null;
    try {
        model = handlerAdapter
            .handle(request, response, chain.getHandler());
    } catch (Exception e) {
        e.printStackTrace();
    }
    return model;
}
}

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations =
{ "classpath:test-applicationContext.xml" })
public class RollicallControllerTest extends JUnitControllerBase {

    @Autowired
    private SecUserDetailsService secUserDetailsService;

    @Test
    public void testForm() {
        //设置安全性上下文
        UserDetails userDetails =
secUserDetailsService.loadUserByUsername("01039");
        SecurityContextHolder.getContext().setAuthentication(new
UsernamePasswordAuthenticationToken(userDetails, "123"));

```

```
// 创建Mock对象
MockHttpServletRequest request = new MockHttpServletRequest();
MockHttpServletResponse response = new MockHttpServletResponse();
request.setServletPath("/rollcall/form/1/1");
try {
    ModelAndView mav = this.excuteAction(request, response);
    System.out.println(mav);
} catch (Exception e) {
    e.printStackTrace();
}
}
```

样本

仅供参考

严禁抄袭

结语

经过半年的时间的研究实践，研究结果说明，针对性、客户定制化、需求明确化的信息化和自动化管理系统对于高校的教学管理是非常重要的。在绝大多数高校中，基本都拥有自己的一套教务管理系统，这套管理系统是用于处理大部分常见的教学管理，例如学生选课和成绩查询，但是这套系统并不能处理一些其他看似不重要的教学管理，例如考勤管理、贫困生资助管理、实践活动学时管理等等。这些业务需要非常庞大的人力成本，而且工序流程不简单，琐碎的事非常多，教务管理的老师在没有一个需求定制化的信息管理系统的時候，只能通过一些字处理程序整理出不通用的不规范的数据，这样的办事效率非常低而且错误率非常高。所以分析不同岗位的工作任务，分析教师的对于教学管理的一些需求，定制一套或多套集成系统，能搞大大提高教学管理的效率和质量。

在这次的研究实践中，我们解决了学院考勤业务上的基本需求，让教师们更有效率的工作和教学。但本次研究的不足之处是，我们并没有太全面的考虑到系统的通用性，因为要加快工程进度，所以很多地方都采取了简单化，把一些短时间之内不需要用到的业务先搁置，所以导致后来其他学院要使用本系统的时候，系统的数据结构需要重新分析和建构，很多业务的处理上需要把更多的因素考虑进去，例如信息的筛选需要有学院的条件，还要去个别分析每个学院的课程的性质和特征，以便设计和做出一套更加通用的教务系统。

样本

仅供参考

严禁抄袭

参考文献

- [1] 李刚. 轻量级 JavaEE 企业应用实战(第3版)—— Struts 2+Spring 3+Hibernate 整合开发[M]. 北京:电子工业出版社, 2011.
- [2] 萨默维尔. 软件工程[M]. 北京:机械工业出版社, 2006.
- [3] 迪尔瓦. MySQL Cookbook[M]. 北京:电子工业出版社, 2008.
- [4] 布吕格, 迪图瓦. 面向对象软件工程: 使用 UML、模式与 Java[M]. 北京:清华大学出版社, 2011.
- [5] 斯宾耐立思, 郭西奥斯. 架构之美[M]. 北京:机械工业出版社, 2011.
- [6] 拉曼. UML 和模式应用[M]. 北京:机械工业出版社, 2006.
- [7] 弗里曼. Head First 设计模式 [M]. 北京:中国电力出版社, 2007.
- [8] 王大远. Div+CSS3.0 网页布局案例精粹[M]. 北京:电子工业出版社, 2011.
- [9] 佩腾. 软件测试[M]. 北京:机械工业出版社, 2006.
- [10] 梁. Java 语言程序设计 基础篇[M]. 北京:机械工业出版社, 2008.
- [11] 韦伯, 帕拉斯, 泰迪斯, 鲁滨逊. REST 实战[M]. 南京:东南大学出版社, 2011.
- [12] 弗拉纳根. JavaScript 权威指南[M]. 北京:机械工业出版社, 2007.

致 谢

暨论文完成之际，我特此向指导和帮助我的老师、同学、朋友及家人表示诚挚的谢意。首先在此，我特别要感谢 B308dot5 工作室的每一位成员所付出的努力，感谢杨俊雄，麦榕聪，没有你们的帮助，考勤系统无法完成得如此出色，尤其感谢我们的指导老师杨林博士，在他的精心指导下，我们完成了考勤系统，克服了开发过程中的大部分困难。老师对软件和编码的热爱和工作的热情，对学生的负责和扎实的编程功底、创新的学术思维都深深地影响了我。我从指导老师的、组员的身上学到了从书本上学不来的品质、精神和实践经验。正因为这个团队，这些组员和指导老师，我才能在学业的收官之路上有如此精彩的经历和优秀的成就。

其次，我要感谢在北京师范大学珠海分校信息技术学院里的所有教师的辛勤劳动和悉心指导，让我的理论知识和实践能力都有质的提升。感谢学校和学院提供一个非常棒的学习环境给我，让我能够心无旁骛的沉浸在这种无忧无虑的快乐的氛围中成长。再次，我要感谢软件工程学院实验班的每一位同学在四年的学习生活里对我的帮助我宽容，感谢你们给了我一个学习气氛如此浓厚的环境，让我感受竞争带来愉悦并促使自己快速进步。

再次，我要感谢我的室友，四年的大学生活，他们带给我太多的愉快的回忆。陪我度过难关，一起熬夜学习和工作，一起玩乐一起生活，他们是我在大学里的家人，是我无畏前进的有力屏障。

最后，感谢我的父母。没有你们的无私奉献和这么多年来对我的鼓励和支持，才使我能够顺利并出色完成大学四年的学业。你们在物质上和精神上都为我创造了优越的条件，支撑着我的过去、现在和将来。

感谢论文评审的各位老师！

仅供参考
严禁抄袭

北京师范大学珠海分校本科生毕业论文评定表

信息技术 学院(系) 2009 级 软件工程 专业 姓名 杨健东 学号 0901030130

题目	考勤管理系统设计与实现		
指导教师意见	(论文评语及给出初步成绩)		
	<div>(设计)</div> <div>指导教师签章</div> <div>201 年 月 日</div>		
答辩小组意见	(论文、答辩评语，成绩及是否推荐院级优秀论文)		
	<div>成绩(百分制) (四级分制) ; 推荐申报院级优秀论文投票: 赞成____人, 反对____人, 弃权____人。</div> <div>组长签章</div> <div>201 年 月 日</div>		
院级评优意见	(是否同意评为院级优秀论文及推荐校级优秀论文)		
	<div>推荐申报院级优秀论文投票: 赞成____人, 反对____人, 弃权____人。</div> <div>教学院长(主任)签章 院系章</div> <div>201 年 月 日</div>		
校级评优意见	<div>教务处长(签章)</div> <div>201 年 月 日</div>		

注：(1) 此表一式四份，一份存入学校档案，一份装入学生档案、一份装入论文封底，一份学院存档、长期保存。
填写时务必字迹工整，签章俱全。(2) *如系两位教师合作指导，应同时签名。