**NIIT教务系统**

**项目设计说明书**

**（重要资料）**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： | NIITES-WEBAPP- JAN-QD-DETAIL |
| 当前版本： | 1.4 |
| 作 者： | Java5-Team10 |
| 完成日期： | 2018.1.3 |

**开发小组：QDU-2014级-JAVA05-第10组**

**TeamLeader: 刘国政**

**成员: 葛礼明 刘欣 仇鑫迪 李帅**

**目 录**

[**一、引言 1**](#_Toc504058976)

[**1.1编写目的 1**](#_Toc504058977)

[**1.2范围 1**](#_Toc504058978)

[**1.2定义 2**](#_Toc504058979)

[**1.3参考文献 2**](#_Toc504058980)

[**二、系统框架结构** 3](#_Toc504058981)

[**三、系统的结构** 7](#_Toc504058982)

[**四、本软件各模块的设计说明** 8](#_Toc504058983)

[ 4.1.1系统管理 8](#_Toc504058984)

[ 4.1.2用户管理 8](#_Toc504058985)

[ 4.1.3学生管理 10](#_Toc504058986)

[ 4.1.4考勤管理 11](#_Toc504058987)

[ 4.1.4教务管理 14](#_Toc504058988)

[ 4.1.5成绩表现管理 15](#_Toc504058989)

[**五、本软件业务关系** 16](#_Toc504058990)

[**六、系统用例具体分析** 18](#_Toc504058991)

[**七、其他模块活动** 21](#_Toc504058992)

[**八、系统实现** 25](#_Toc504058993)

[**8.1 用户登录** 25](#_Toc504058994)

[**8.2 详细模块实现** 26](#_Toc504058995)

[**九、开发总结** 29](#_Toc504058996)

[**9.1 课题总结** 29](#_Toc504058997)

[**9.2** **心得体会** 29](#_Toc504059003)

# **一、引言**

## 1.1编写目的

由于NIIT目前的教学管理工作比较繁琐，使用传统的人工管理方式已经很难满足目前的教学工作，学生数据、教学数据、成绩数据、就业数据等很多数据无法共享，导致数据混乱，因此为了让NIIT的教学管理工作变得更简单、更科学，通过对NIIT日常教学工作的了解，梳理出了相关的业务需求，以便开发该教务管理系统。

## 1.2范围

该软件系统主要应用于NIIT的教学管理，主要包括以下业务：

该系统主要有 系统管理、学生管理、考勤管理、教务管理、学习管理、成绩管理、就业管理等10大模块；

其中：

系统管理包括：系统菜单类型（login/public）：{登录后用户菜单/未登录时访客菜单}、系统菜单管理、角色管理（维护角色、角色分配菜单）、用户管理、权限管理、系统参数、系统公告等子功能。

学生管理包括：学生注册、学生档案管理。

考勤管理包括：考勤打卡、请假管理、请假类型管理。

教务管理包括：专业方向管理、班级信息管理、课程类型管理、课程管理。

学习管理：课程笔记。

成绩管理包括：考勤报表、成绩汇总、成绩查询。

就业管理包括：简历管理。

## 1.2定义

**类 图：**[类图](http://doc.qiuhao.com/phrase/200603071659325.html)(Class diagram)是显示了模型的静态结构，特别是模型中存在的[类](http://doc.qiuhao.com/phrase/200603090857555.html)、类的内部结构以及它们与其他类的关系等，用于描述系统的结构化设计。

**对象图：**[对象图](http://doc.qiuhao.com/phrase/200604161646065.html)(Object Diagram) 是显示了一组[对象](http://doc.qiuhao.com/phrase/200603090845215.html)和他们之间的关系的图示，一个对象图可看成一个类图的特殊[用例](http://doc.qiuhao.com/phrase/200604240937105.html)，

**状态图：**状态图(State Diagram)是描述一个实体基于事件反应的动态行为，显示了该实体如何根据当前所处的状态对不同的时间做出反应的，用于显示状态机（它指定对象所在的状态序列）、使对象达到这些状态的事件和条件、以及达到这些状态时所发生的操作。

## 1.3参考文献

《中华人民共和国国家标准计算机软件产品开发文件编写指南》中华人民共和国电子工业部

《中华人民共和国国家标准**信息处理-数据流程图、程序流程图、系统流程图、程序网络图和系统资源图的文件编制符号及约定**》中华人民共和国机械电子工业部

《中华人民共和国国家标准计算机软件产品开发文件编写指南》中华人民共和国电子工业部

《中华人民共和国国家标准**信息处理-数据流程图、程序流程图、系统流程图、程序网络图和系统资源图的文件编制符号及约定**》中华人民共和国机械电子工业部

GB8566-88计算机软件开发规范

GB9385-88计算机软件需求说明编制指南

GB9386-88计算机软件测试文件编制规范

BG/T12504-90计算机软件质量保证计划规范

GB/T12505-90计算机软件配置管理计划规范

**二、系统框架结构**

本系统采用三层架构设计，三层架构就是将整个业务应用划分为：表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL）。区分层次的目的即为了“高内聚，低耦合”的思想。

１、表现层（UI）：通俗讲就是展现给用户的界面，即用户在使用一个系统的时候他的所见所得。

２、业务逻辑层（BLL）：针对具体问题的操作，也可以说是对数据层的操作，对数据业务逻辑处理。

３、数据访问层（DAL）：该层所做事务直接操作数据库，针对数据的增添、删除、修改、更新、查找等。

3个层次中，系统主要功能和业务逻辑都在业务逻辑层进行处理。三层体系将业务规则、数据访问及合法性校验等工作放在中间层处理。客户端不直接与数据库交互，而是通过组件与中间层建立连接，再由中间层与数据库交互。中间层采用的是流行的Spring + Hibernate+Struts，为了将控制层与业务逻辑层分离，又细分为以下几种。

**Web层：**就是MVC模式里面的“C”（controller），负责控制业务逻辑层与表现层的交互，调用业务逻辑层，并将业务数据返回给表现层作组织表现，该系统的MVC框架采用Struts。

**Service层（就是业务逻辑层）**：负责实现业务逻辑。业务逻辑层以DAO层为基础，通过对DAO组件的正面模式包装，完成系统所要求的业务逻辑。

**DAO层：**负责与持久化对象交互。该层封装了数据的增、删、查、改的操作。

**PO（持久化对象）：**通过实体关系映射工具将关系型数据库的数据映射成对象，很方便地实现以面向对象方式操作数据库，该系统采用Hibernate作为ORM框架，将Strust2作为流程控制。

Spring的作用贯穿了整个中间层，将Web层、Service层、DAO层及PO无缝整合，其数据服务层用来存放数据。

其中使用的mybatis框架详细如下图所示：

SqlMapConfig.xml配置了数据源、事务等mybatis运行环境

配置映射文件（配置sql语句）

mapper.xml（映射文件）

SqlSessionFactory（会话工厂），根据配置文件创建工厂

作用：创建SqlSession

msyql

SqlSession（会话），是一个接口，面向用户（程序员）的接口

作用：操作数据库（发出sql增、删、改、查）

Executor（执行器），是一个接口（基本执行器、缓存执行器）

作用：SqlSession内部通过执行器操作数据库

mapped statement（底层封装对象）

作用：对操作数据库存储封装，包括 sql语句，输入参数、输出结果类型

输出结果 类型

java简单类型

hashmap

pojo自定义

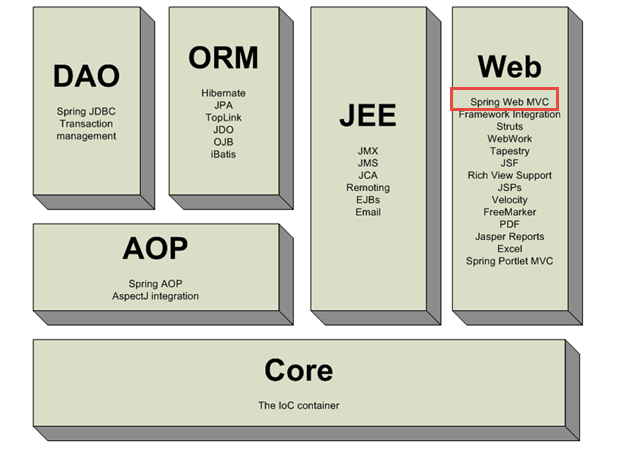
输入参数类型

java简单类型

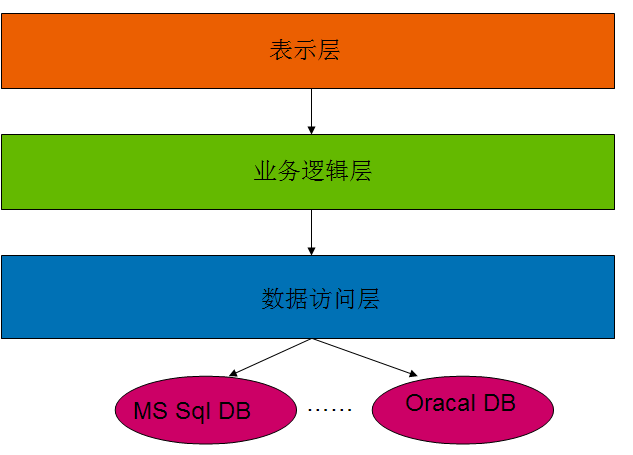
hashmap

pojo自定义

SpringMVC

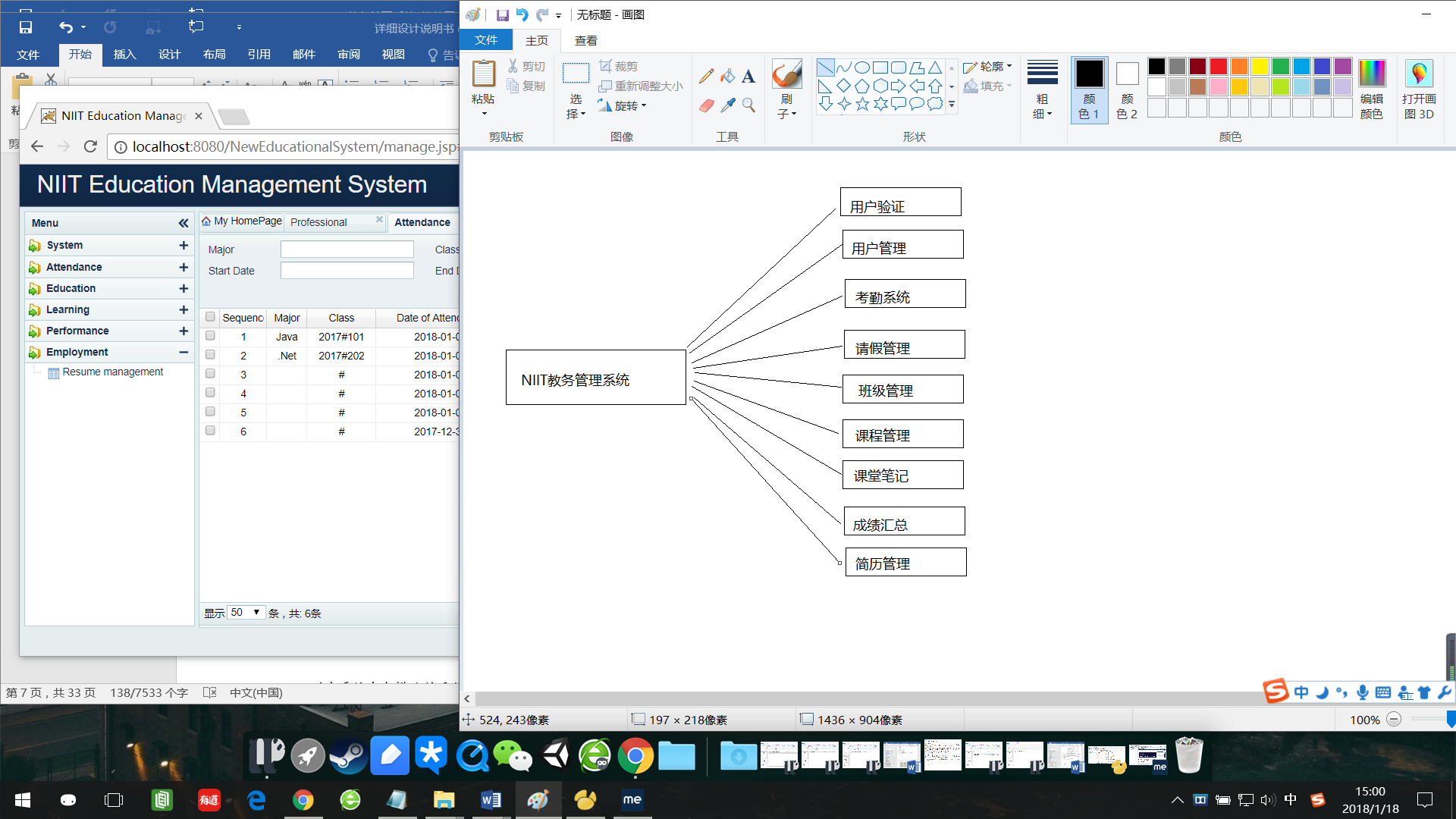


这三层的结构层次简单表示如下图：



# **三、系统的结构**

系统的结构主要由两方面构成，一是系统的软件结构框图，二是系统的硬件结构框图。软件结构框图中将给出系统所应分层的顶级模块；而硬件结构框图将给出系统运行的硬件结构示意图；本软件采用传统的软件开发生命周期的方法，采用自顶向下，逐步求精的结构化的软件设计方法，图表如下：





# **四、本软件各模块的设计说明**

### 4.1.1系统管理

**模块描述：**通过此功能维护完菜单信息后，当用户进入系统后看到的

菜单可以根据权限动态展示不同的菜单。同时显示系统公告以及系统版本信息。

**功 能：**实现对系统菜单信息的添加，删除，修改，查询和查看

功能，系统需要维护菜单的信息。

**性 能：**页面风格与主页面风格一致。

页面操作要方便用户使用。

**输 出 项：**根据不同用户显示不同系统菜单选项

**设计方法：**设计算法。检测用户id，根据用户id判断当前登录用户类型

从而显示不同的用户界面

### 4.1.2用户管理

**模块描述：**系统用户信息的添加，删除，修改，查询和查看功能

**功 能：**管理员可以维护系统的用户账号、密码等信息。

**性 能：**页面风格与主页面风格一致。

页面操作要方便用户使用。

**输 入 项：**用户id、真实姓名、登录密码、角色、其他个人信息

**输 出 项：**管理员可以管理所有用户增删改查、老师管理学生

**设计方法：**添加、修改、删除成功后都要返回默认列表页，查询时根据角色进行查询，多种查询方法（按时间、类型等），查询所有角色，得到列表页

**流程图：**

管理界面

用户名

真实姓名

密码

角色

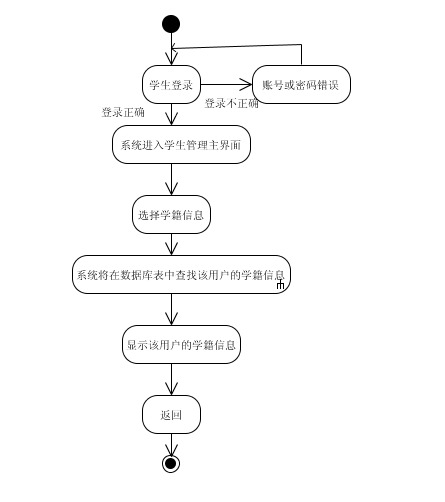
其他信息

否

更新到数据库

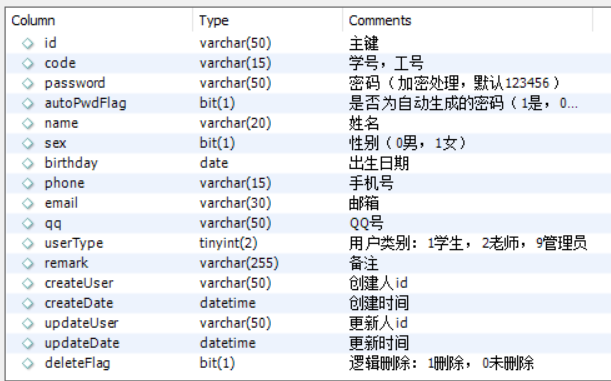
合法用户

结束



**数据库设计表：**

用户管理

****

### 4.1.3学生管理

**模块描述：**学生注册管理、学生档案管理

**功 能：**注册学生信息，系统自动使用该学生学号作为登录账号，

并生成一个初始密码，自动分配默认权限，在学生注册成功

后，系统可以发送邮件提示初始密码。

提供学生信息的管理功能，比如管理员老师可以通过该功能查看、修改学生档案等。

**性 能：**页面风格与主页面风格一致。

页面操作要方便用户使用。

**输 入 项：**数据库读取到的当前学生详细信息

**设计方法：**添加、修改、删除成功后都要返回默认列表页，查询时根据角色进行查询，多种查询方法（按时间、类型等），查询所有角色，得到列表页

**流程图：**

管理界面

学生档案信息

否

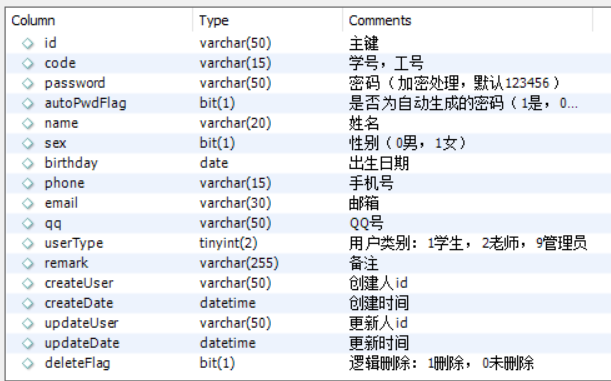
数据操作库

合法操作

结束

**数据库设计表：**

学生\_用户管理

****

### 4.1.4考勤管理

**模块描述：**考勤管理、请假类型管理、请假管理

**功 能：**考勤管理：上课前登录系统打卡，下课后也要登录系统

打卡，每天需要打两次卡，如果只打一次，算早退； 打了多次，按照第一次和最后一次算。没打卡算旷课。

请假类型：系统中记录请假信息，请假分为事假、病假等不同类型。

请假管理: 通过该系统填写请假信息，并将假条照片作为附件上传，提交等待审核，审核通过后请假生效

**性 能：**页面风格与主页面风格一致。

页面操作要方便用户使用。

**输 入 项：**打卡次数记录、打卡时间记录、请假提交

**设计方法：**初始登录系统自动提醒打卡，同事记录到数据库当前打卡时间，当前打卡次数。申请请假时不仅更新学生记录，同步更新到老师界面。

**流程图：**

打卡

是

初始打卡

否

标记迟到或旷工

更新打卡信息

更新到数据库并通知教师端

结束

请假申请

请假类型

同步到数据库

通知教师端审核

审核失败

返回学生端

审核成功

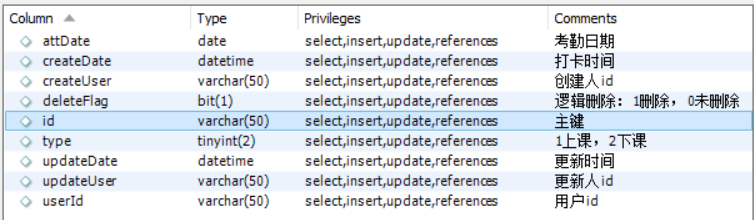
填充请假标识

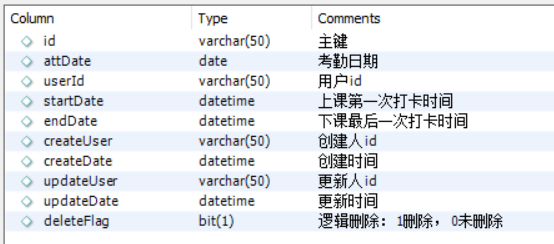
结束

结束

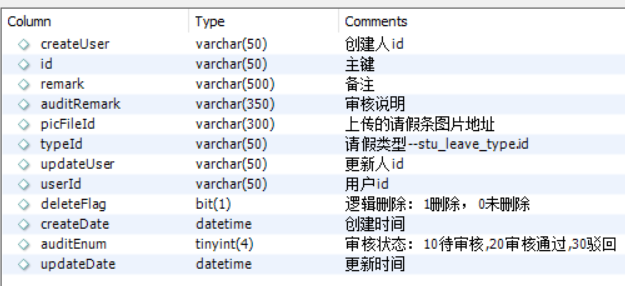
**数据库设计表：**

考勤管理

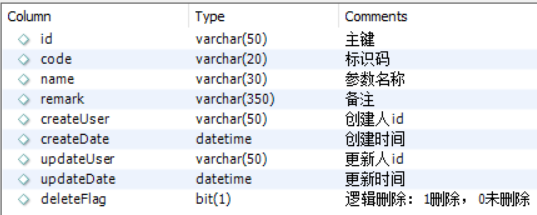


****

请假记录表

****

请假类型记录

****

### 4.1.4教务管理

**模块描述：**课程管理、班级管理、课堂笔记管理、

**功 能：**针对不同的课程可能会采取不同的授课方式，因此我们将

课程分成不同的类型。

按照不同的学年、专业方向，各自对学生进行分班管理，因此班级信息需要在该系统中进行管理。

通过课程表，每个学生都可以知道自己所学课程，在该系统中，学生可以对自己所学课程的各个章节记录笔记。

**性 能：**页面风格与主页面风格一致。

页面操作要方便用户使用。

**设计方法：**初始登录系统自动提醒打卡，同事记录到数据库当前打卡时间，当前打卡次数。申请请假时不仅更新学生记录，同步更新到老师界面。

### 4.1.5成绩表现管理

**模块描述：**考勤报表、成绩汇总

**功 能：**完成成绩录入之后，可以对每个学生按照课程占比，以及

出勤率计算总成绩，并展示，管理员老师可以通过该功能查看

每个学生的最终成绩。

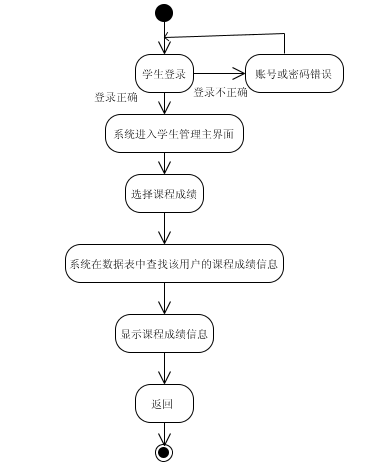
统计学生在指定时间段内的出勤情况，可以查看一个专业方向或一个班级下所有学生各自的出勤情况，也可以查看具体某一个学生的出勤情况，可以通过指定不同的统计条件来实现统计。

**性 能：**页面风格与主页面风格一致。

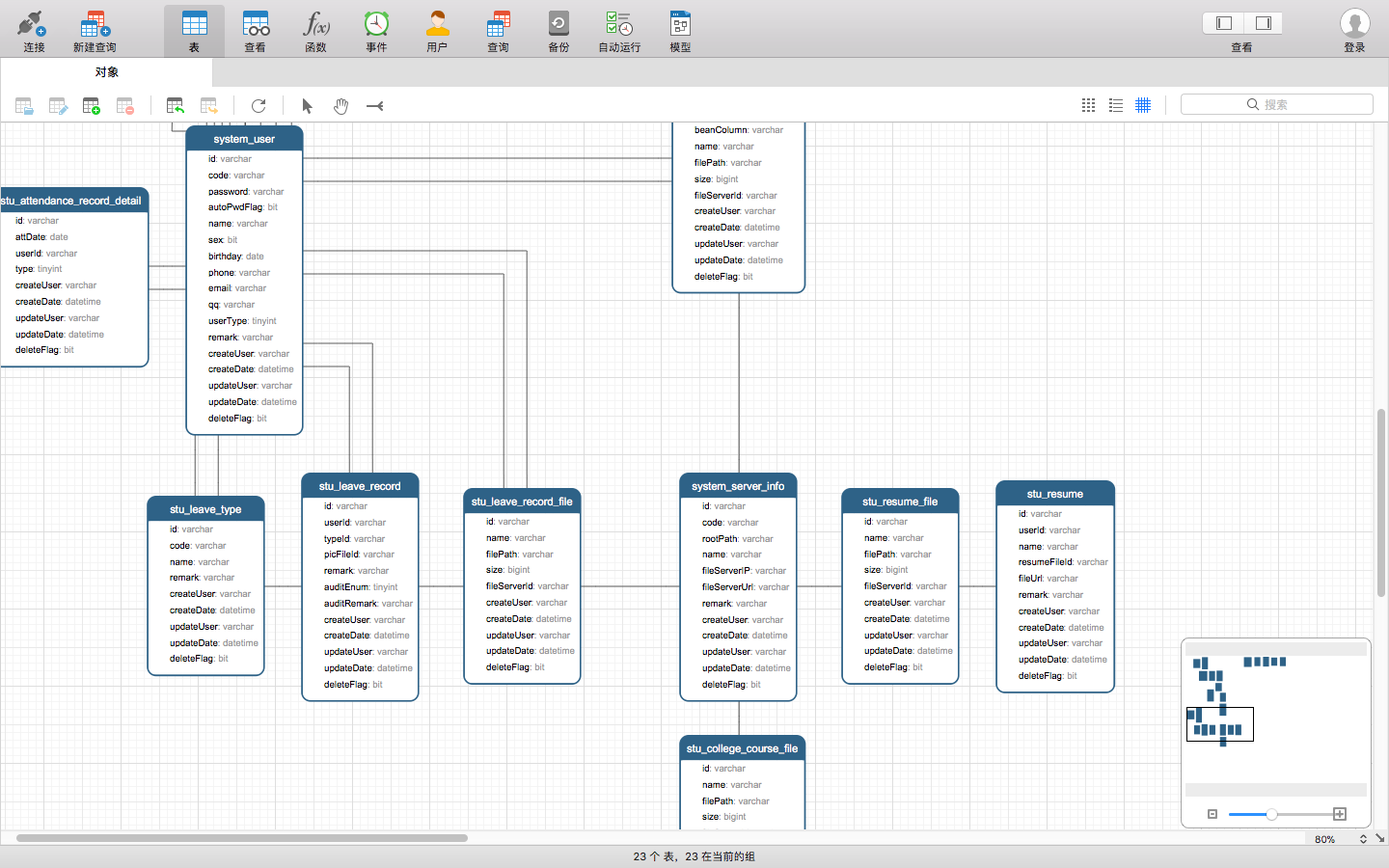
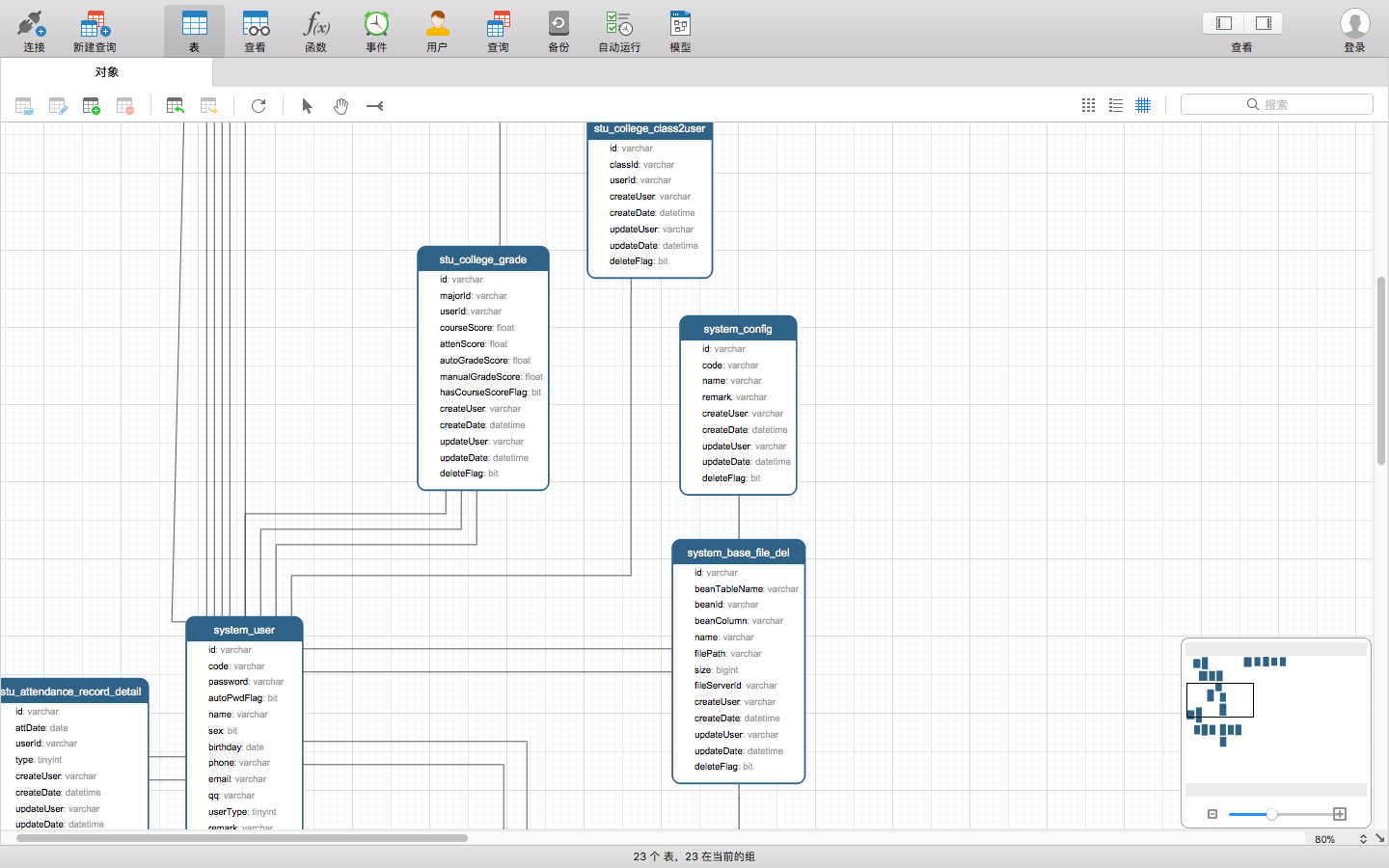
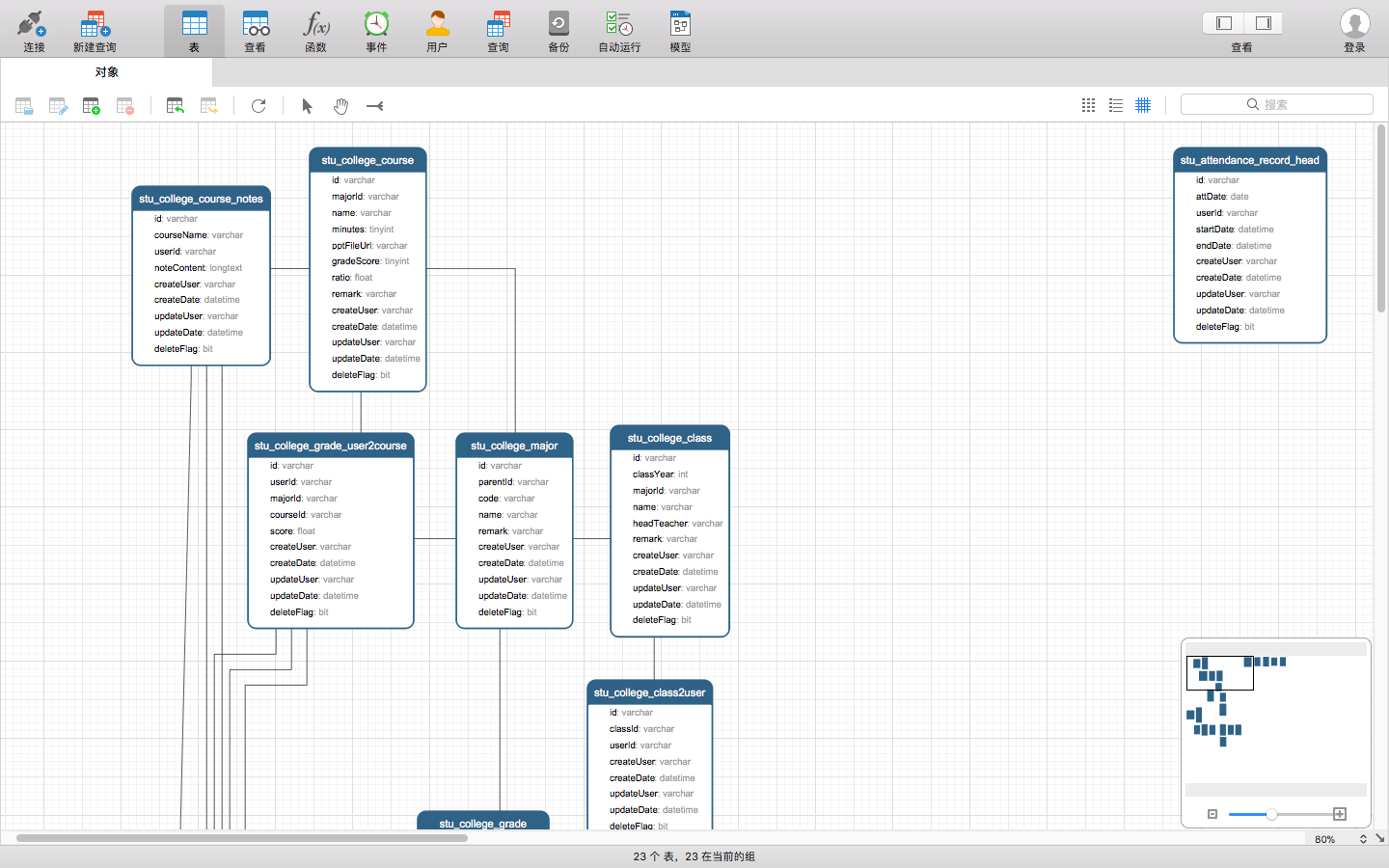
页面操作要方便用户使用。

**设计方法：**根据学生表关联的成绩以及考勤记录生成报表。

**流程图：**

成绩查询

# **五、本软件业务关系**



# **六、系统用例具体分析**

1. 教务处管理员管理用例
2. 系统管理：包括添加教务人员及管理员自己的密码修改，添加教务人员，完成新教务人员的基本信息，输入教务人员的编号，密码，确认密码及姓名密码修改，完成管理员本身的密码更新，输入密码，确认密码。
3. 学生信息管理：包括学生查询及添加学生

学生查询，完成学生基本信息的查询，可根据类型，匹配模式，关键值查看指定的学生信息，也可完成学生信息的修改及删除；

添加学生，完成学生的添加，输入对应学生的基本信息。

1. 教师信息管理：包括教师查询及添加教师

教师查询，完成老师基本信息的查询，可根据类型，匹配模式，关键值查看指定的教师信息，也可完成教师信息的修改及删除

添加教师，完成教师的添加，输入对应教师的基本信息

1. 课程信息管理：包括课程查询，添加课程，课程表安排，成绩录入，院系管理

课程查询，完成课程基本信息的查询，可根据类型，匹配模式，关键值查看指定的课程信息，也可完成课程信息的修改及删除

添加课程，完成课程的添加，输入对应课程的编号，课程名，学分等相关信息

课程表安排，完成课程表的快速查询，可根据院第，年级，班级及课程的类型来快速查询课程，完成课程表的显示，可对课程表中的内容进行插入，修改及删除，完成指定班级的所有课程安排的显示成绩录入，完成学生对所先课程成绩的录入及修改

院系管理，完成所有院系基本信息的显示及添加

1. 通知系统管理：包括通知查询及发布通知

通知查询，完成通知公告的基本信息内容及删除

发布通知，完成发布公告，输入通知标题，具体内容从而发布消息

1. 设置选课标志：完成学生选课标志的设置，“是”或“否”
2. 查看平均成绩：完成指定院系平均成绩统计查看
3. 重新录入：退出本界面，显示登录界面重新登录
4. 退出：退出本系统

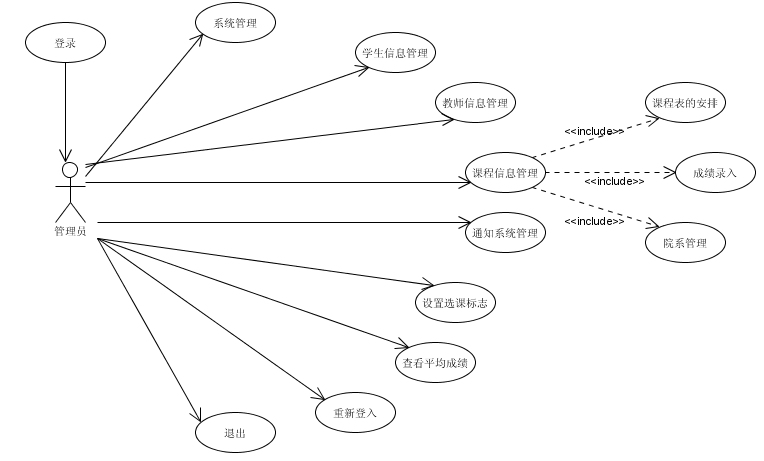


图3-1 管理员管理用例图

1. 教师管理用例
2. 教师信息设置：包括个人信息的查看，密码修改，重新登入及退出。

个人信息，完成个人信息的查看

密码修改，完成个人的密码修改，输入密码，确认密码

重新登入，退出本界面，显示登录界面重新登录

退出，退出本系统

1. 学生信息管理：完成学生基本信息的查询，可根据类型，匹配模式，关键值查询指定的学生。
2. 课程信息管理：包括课程信息的查询，课程表的查询，平均成绩的查看及成绩查询。

课程信息的查询，完成课程基本信息的查询，可根据类型，匹配模式，关键值查询指定的课程

课程表的查询，完成课程基本信息的查询，可根据类型，匹配模式，关键值查看指定的课程信息

平均成绩的查看，完成指定院系平均成绩统计查看

成绩查询，完成指定课程的学生的成绩查询

1. 通知管理：包括发送信息，查看收件箱及通知查询。

发送信息，完成发布信息，输入收件人，通知标题，具体内容从而发布消息

查看收件箱，完成本收件邮箱的查看及删除

通知查询，完成管理员所发信息的查询

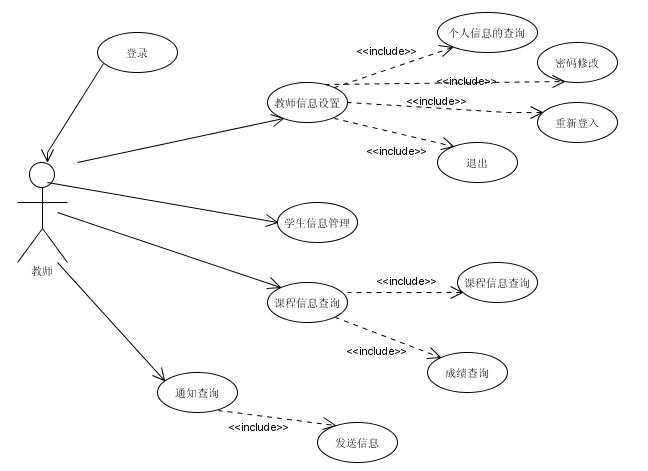
****

图3-2 教师管理用例

1. 学生管理用例
2. 学生信息设置：包括学籍信息，修改密码，重新登入及退出

学籍信息，完成本学生基本信息的查询

修改密码，完成个人的密码修改，输入密码，确认密码

重新登入，退出本界面，显示登录界面重新登录

退出，退出本系统

1. 学生信息设置：包括学籍信息，修改密码，重新登入及退出

课程查询选定，完成课程的查询及课程的选定

课表显示，完成本学生选课情况的课表显示

已选课程，完成本学生已选课程的列表查询及删除

课程成绩，完成课程成绩列表的显示

1. 通知管理：包括发送信息，查看收件箱及通知查询

通知管理：包括发送信息，查看收件箱及通知查询

发送信息，完成发布信息，输入收件人，通知标题，具体内容从而发布消息

查看收件箱，完成本收件邮箱的查看及删除

通知查询，完成管理员所发信息的查询

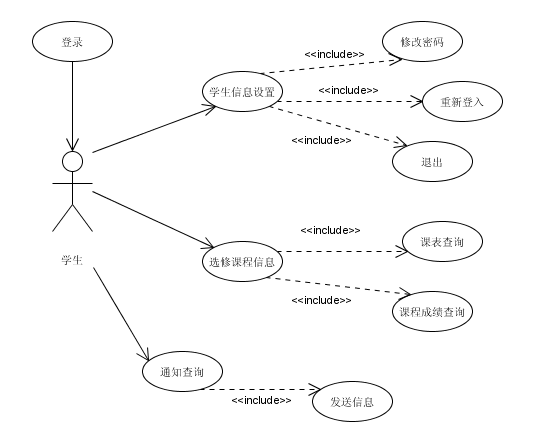
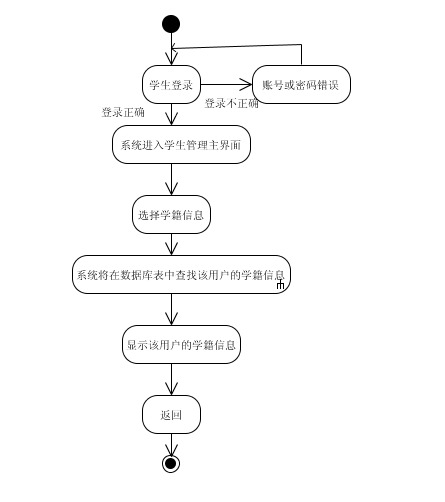


图3-3 学生管理用例

# **七、其他模块活动**

* 1. 学生信息查看

描述：用户成功登录系统，进入学生管理主界面，选择学籍信息一栏，系统则在用户信息数据表中查找该用户的学籍信息，并将其显示到学籍信息界面中。



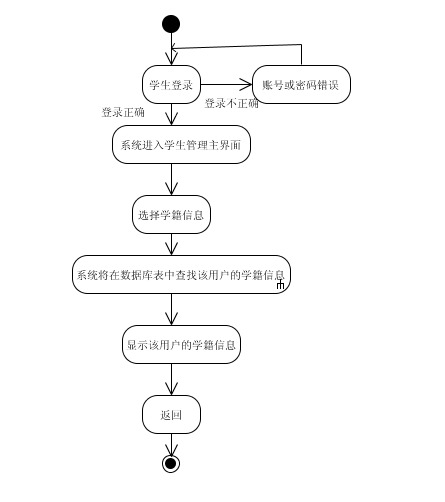


图3-5 学生信息查看活动

* 1. 密码修改

描述：用户成功登录系统，进入学生管理主界面，选择密码修改，系统显示本用户的密码修改界面，然后输入用户新密码及确认密码，如果选择清除，则系统将自动清除用户则输入的信息；如果选择提交则系统将判断两次密码是否一致，如果不一致，则重新输入密码；如果一致，系统将新密码更新到数据库中，最终显示密码修改成功。

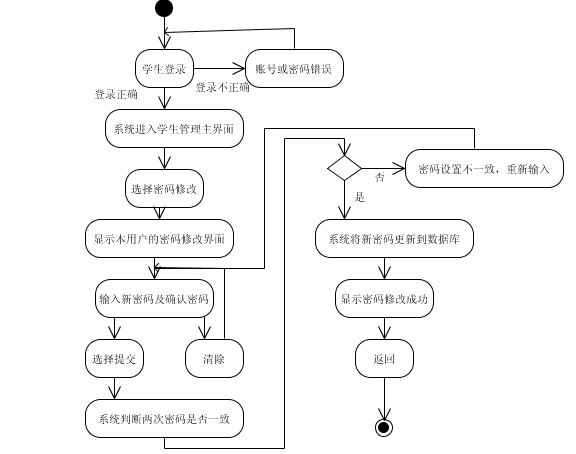


图3-6 密码修改活动

* 1. 重新登录

描述：用户成功登录系统后，进入学生管理主界面，选择重新登录后，系统将退出本界面，直接返回到原始的系统登录界面。

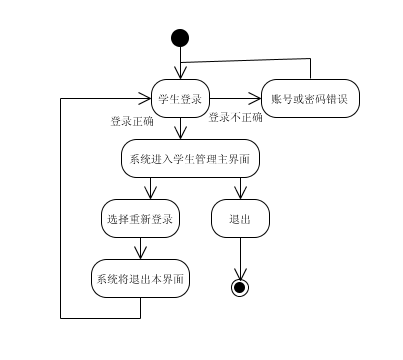


图3-7 重新登录活动

4）通知公告查看

描述：用户成功登录系统后，进入学生管理主界面，选择查看通知，系统将从数据库中查找通知信息，显示所有通知信息的列表。

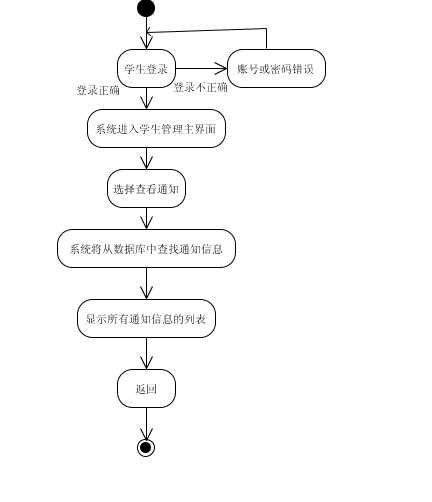


图3-16 通知公告查看活动

5）管理员通知公告删除

描述：用户成功登录系统后，进入学生管理主界面，选择查看通知，系统将从数据库中查找通知信息，显示所有通知信息的列表，选择删除，系统将从数据库表中清除已选择的记录，然后显示删除成功。

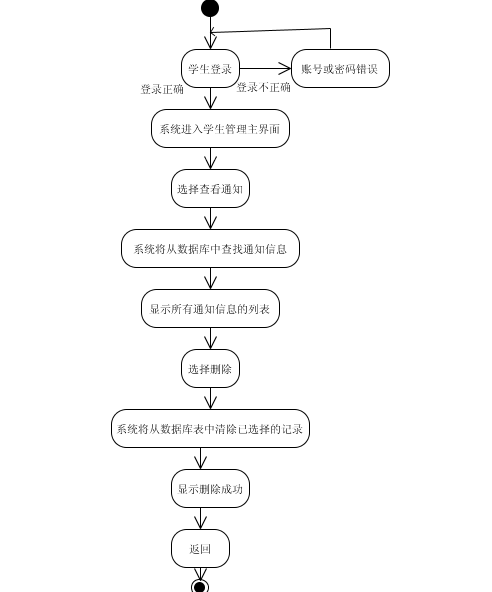


图3-17 管理员公告删除活动

# **八、系统实现**

## **8.1 用户登录**

用户登录界面如图所示：

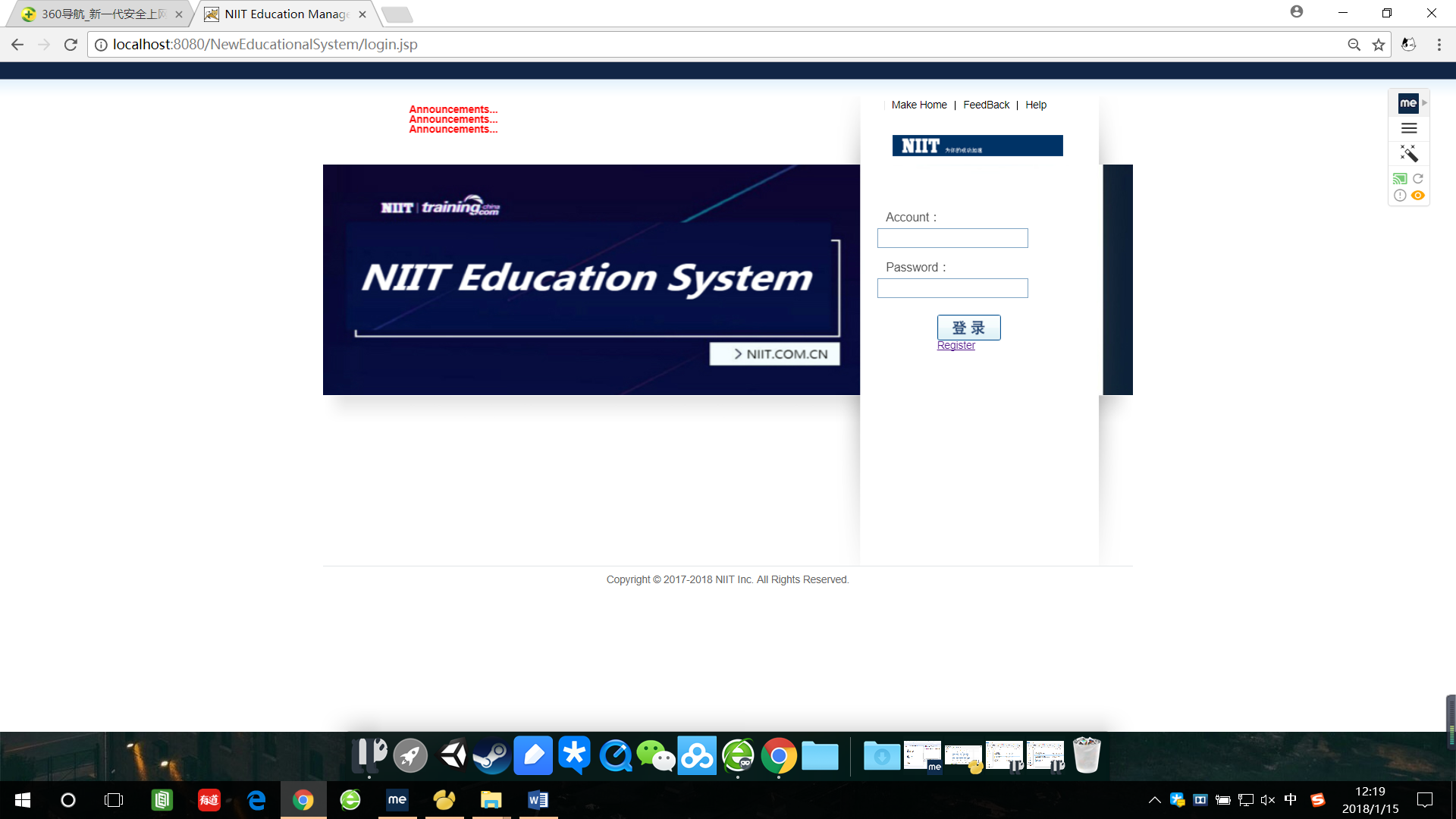


图8-1 用户登录界面

## **8.2 详细模块实现**

1. 学生管理主界面：

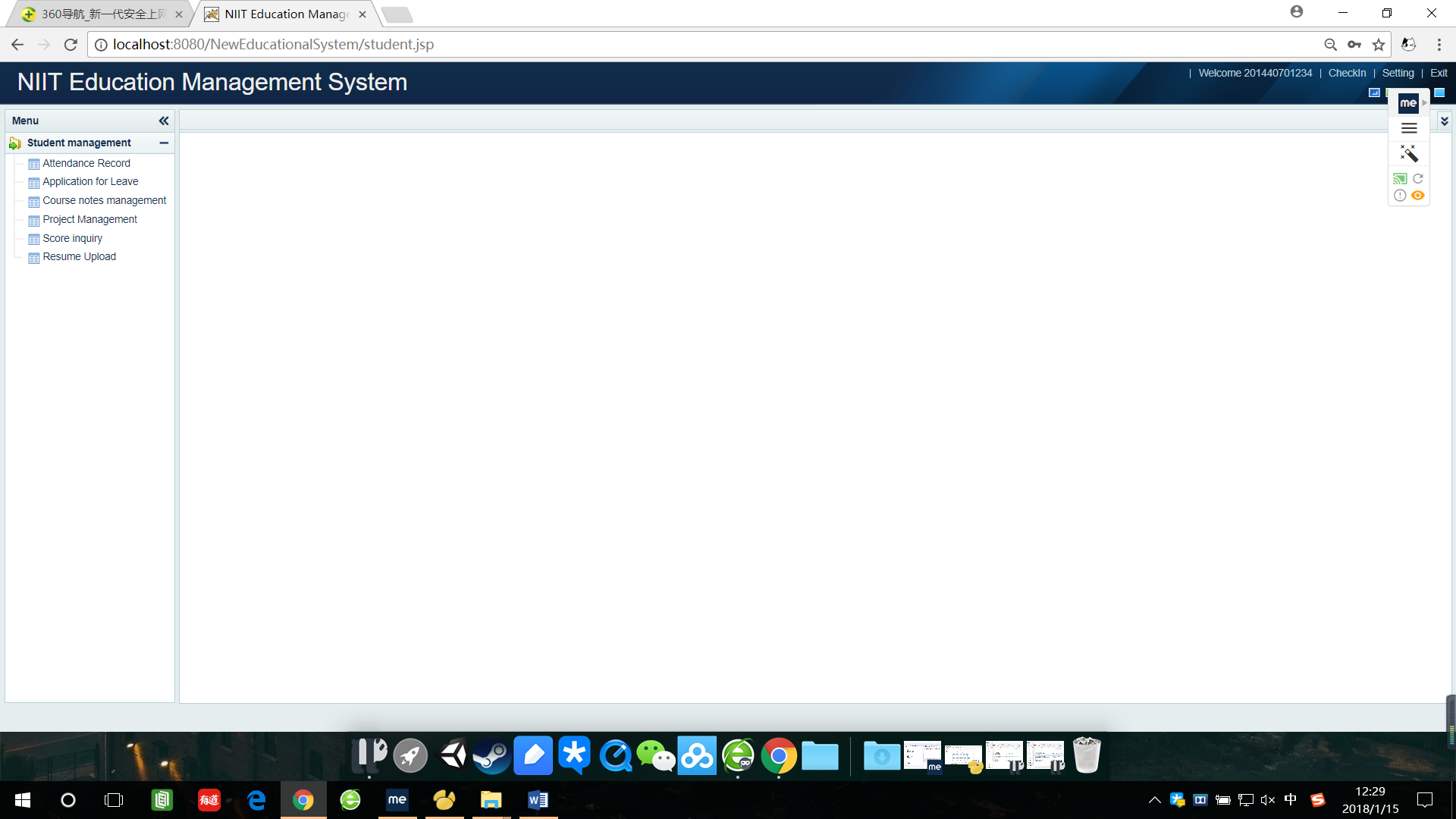


图 学生管理界面

1. 课程查询，如图所示：

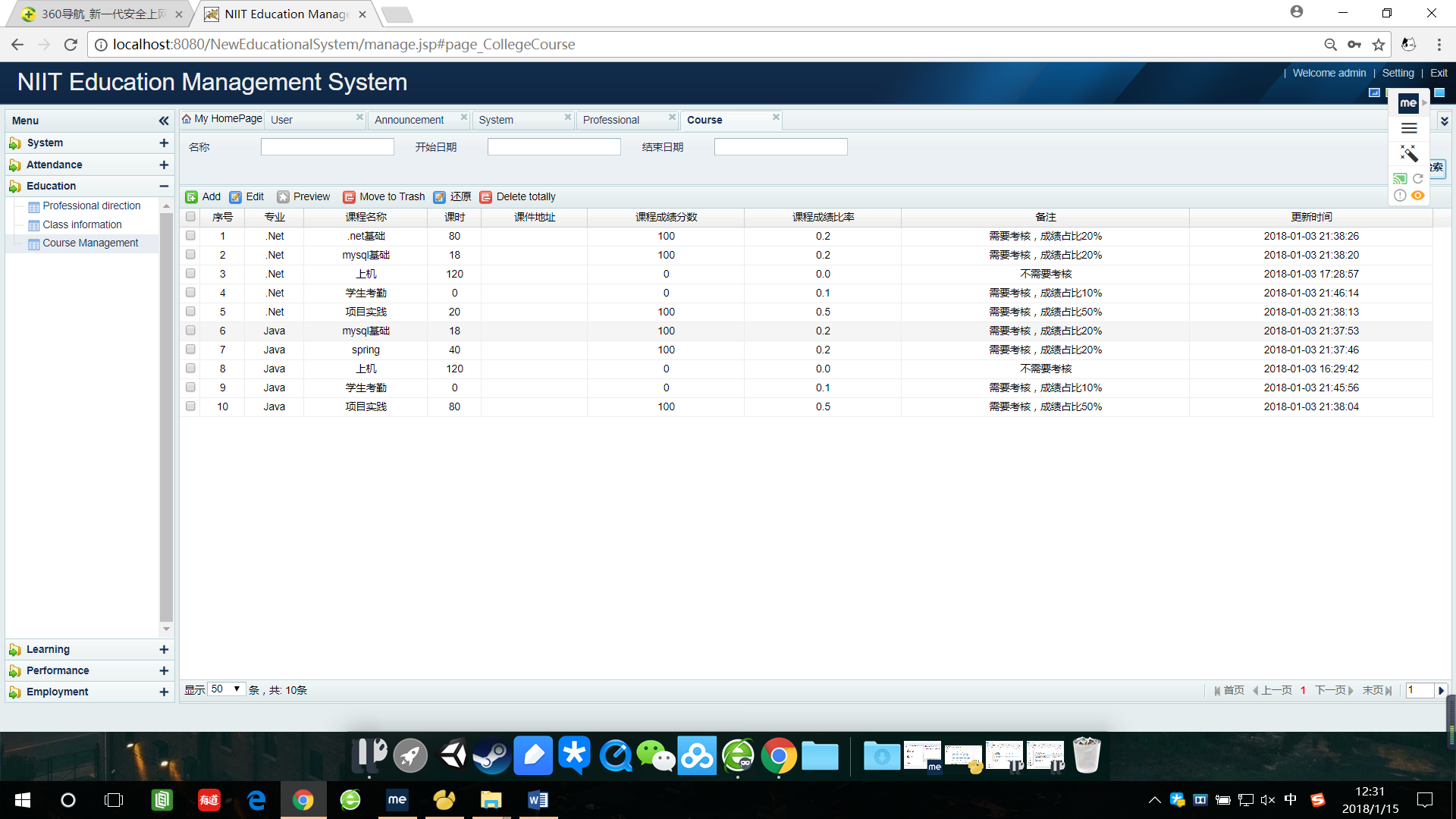


图 课程查询及选定

1. 课程成绩查询界面：

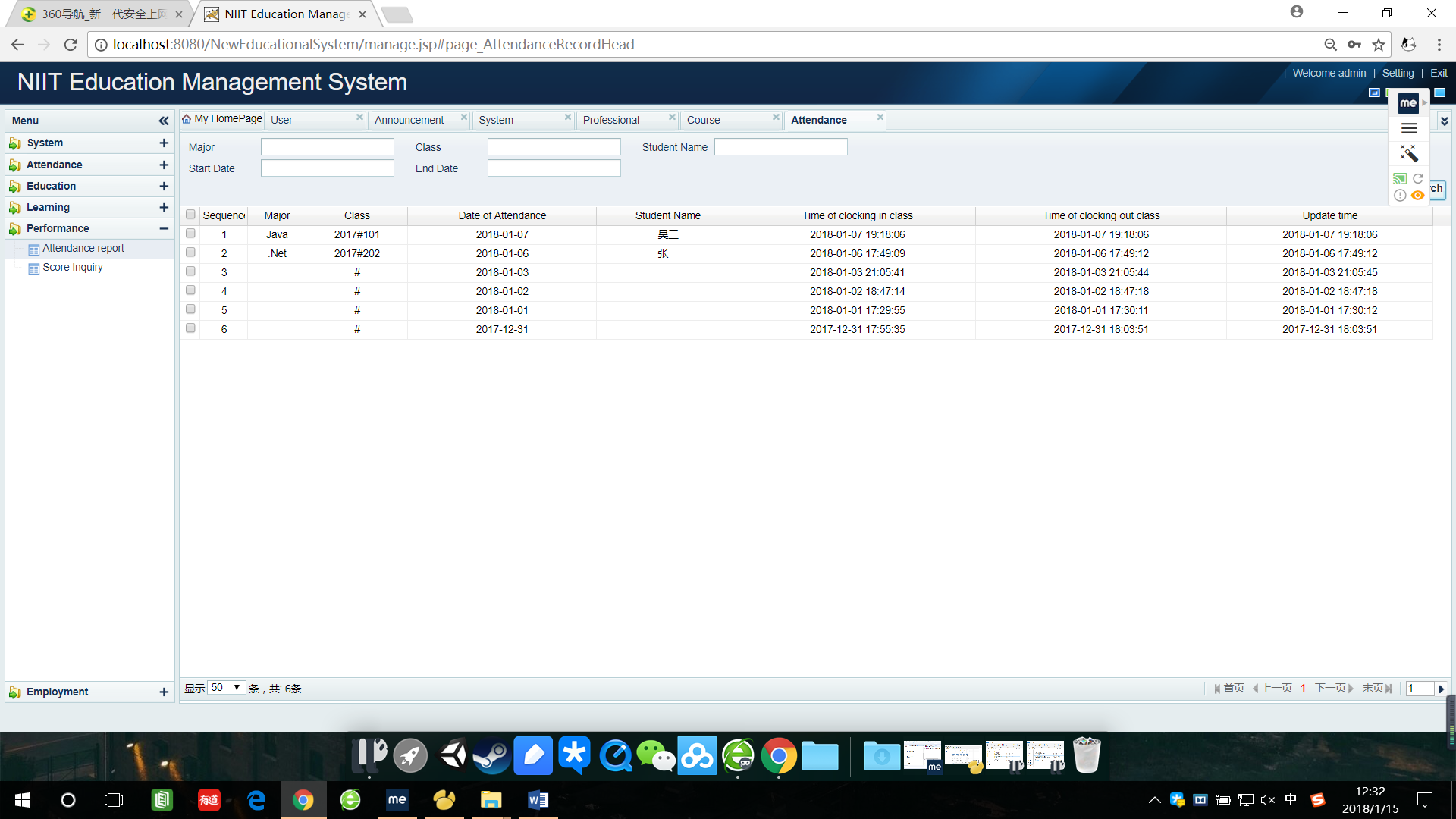


图 课程成绩查询界面

# **九、开发总结**

## **9.1 课题总结**

教务信息管理系统是一种融合管理科学、信息科学、系统科学和计算机技术为一体的综合性先进管理手段。系统是建立在WINDOWS操作系统上的基于B/S模式的教务信息管理系统，基于J2EE的Spring + Mybatis + Mybatis数据库进行开发。本系统主要分为三大模块教务管理模块、教师管理模块和学生管理模块，通过该系统的应用，可以满足高校教务管理工作的高效化、无纸化和网络化的需求，提高高校教务的工作质量与效率。



## **心得体会**

教务管理系统开发完成后，由于开发设计的时间较短，所以说本系统的存在的问题之一是作为一个学生管理能够运行的系统可能功能不够完善；存在的问题之二，学生成绩管理系统中的数据的不够安全。

本次实验设计使我深刻理解了在《NIIT教务管理系统》中所学的软件开发方法与开发技术，也使我懂得了一些软件文档资料写法的规范性。开发一个系统首先进行用户的调查、进行可行性研究，然后对系统进行的分析、进行系统设计、系统的实现，最后进行系统的测试并形成各种文档。

在本次项目设计中遇到了许多的问题，带着这些问题，自己去寻找相应的资料，与同学相互讨论，最终问题得以解决，为自己以后进一步开发系统做了很好的示范。在这次实验设计和文档的写作过程中得到了指导老师大力帮助，在此深表谢意。

在以后的学习或工作的道路中，我会以最好的状态去面对每一个问题，以结课项目的态度及方式来分析及解决相应的问题，把每件事都做到最好，同时也给自己留下宝贵的经验。