****

计算机科学与技术学院

Java框架编程技术

大作业

|  |  |
| --- | --- |
| 班 级： | **计科1507** |
| 姓 名： | **刘晓达** |
| 学 号： | **15110543038** |

2018年 12 月 24日

**大作业任务书及成绩评定**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **课题名称** | | Javaweb框架编程技术 | | | **学生姓名** | **刘晓达** | | | |
| **班级** | | 计科1507 | | | **学号** | **15110543038** | | | |
| **序号** | **评分标准** | | | | | | **满分** | | **得分** |
| 1 | 系统使用SSM框架实现增、删、改、查最基本的功能，且能正常运行。增、删、改、查少一项扣10分。 | | | | | | 60 | |  |
| 2 | 系统设计合理，界面美观，易于操作，代码规范。 | | | | | | 10 | |  |
| 3 | 系统功能完善。 | | | | | | 10 | |  |
| 4 | 答辩过程回答问题准确，基础知识掌握牢固。 | | | | | | 20 | |  |
| **合 计** | | |  | | | | | | |
| **教师签字** | | |  | **成绩等级** | | | |  | |

**目 录**

[《Java框架编程技术》考核说明 1](#_Toc533460405)

[第一章 系统分析与总体设计 5](#_Toc533460406)

[1.1功能需求分析 5](#_Toc533460407)

[2.2 业务流程分析 7](#_Toc533460408)

[2.2.1学生教师业务流程分析 7](#_Toc533460409)

[1.2 系统体系结构设计 8](#_Toc533460410)

[1.3 数据库设计 10](#_Toc533460411)

[第二章 系统的实现 13](#_Toc533460412)

[2.1 登录窗体和系统主界面的设计 13](#_Toc533460413)

[2.2学生模块设计与实现 14](#_Toc533460414)

[2.3 管理员模块的设计与实现 17](#_Toc533460415)

# 第一章 系统分析与总体设计

## 1.1功能需求分析

毕业生选题系统，角色分为管理员，教师和学生三个子系统，分别做出其用例图，状态图用例图

根据对系统目标及需求的分析，建立系统用例模型。用例图如图2-1所示。

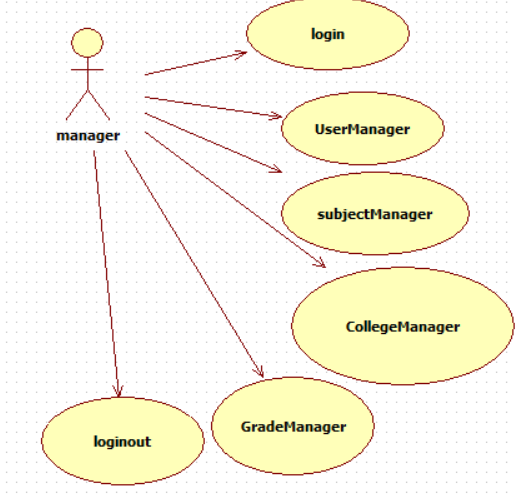
管理员用例图及描述

图2-1 毕业设计选题系统用例图

用例的简要介绍如表2-1所示。

表2-1 管理员用例

|  |  |  |
| --- | --- | --- |
| **参与者** | **用例编号及名称** | **用例简介** |
| Manager | Login | 用户登录进行秘密账号验证 |
| Manager | UserManager | 用户管理对于用户的增加删除 |
| Manager | SubjectManager | 课题查询以及删除不可修改 |
| Manager | CollegeManager | 学院管理 |
| Manager | GradeManager | 年级管理 |
| Manager | LoginOUt | 登出。退出登录 |

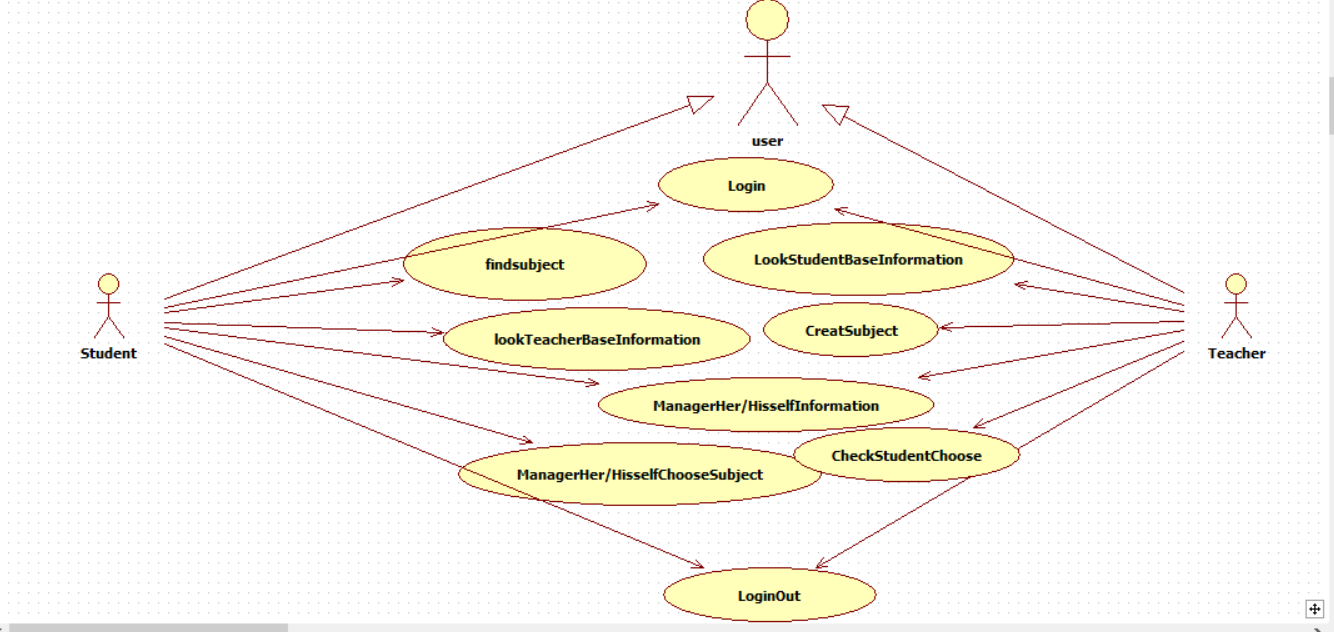
用户用例图及描述

图 2-2 毕业设计选题系统用户用例图

用例的简要介绍如表2-2所示。

表2-2用户例

|  |  |  |
| --- | --- | --- |
| **参与者** | **用例编号及名称** | **用例简介** |
| User-Student | Login | 用户登录进行秘密账号验证 |
| User-Student | FindSubject | 查看课题 |
| User-Student | LookTeacherBaseInformation | 通过课题查看老师信息 |
| User-Student | ManagerHer/HisselfInformation | 个人信息管理 |
| User-Student | ManagerHer/HisselfChooseSubject | 选择课题提交审核 |
| User-Student | LoginOUt | 登出。退出登录 |
| User-Teacher | Login | 用户登录进行秘密账号验证 |
| User-Teacher | CreateSubject | 创建课题 |
| User-Teacher | LookStudentBaseInformation | 通过课题查看学生信息 |
| User-Teacher | ManagerHer/HisselfInformation | 个人信息管理 |
| User-Student | CheckStudentChoose | 审核学生提交课题申请 |
| User-Student | LoginOUt | 登出。退出登录 |

## 2.2 业务流程分析

### 2.2.1学生教师业务流程分析

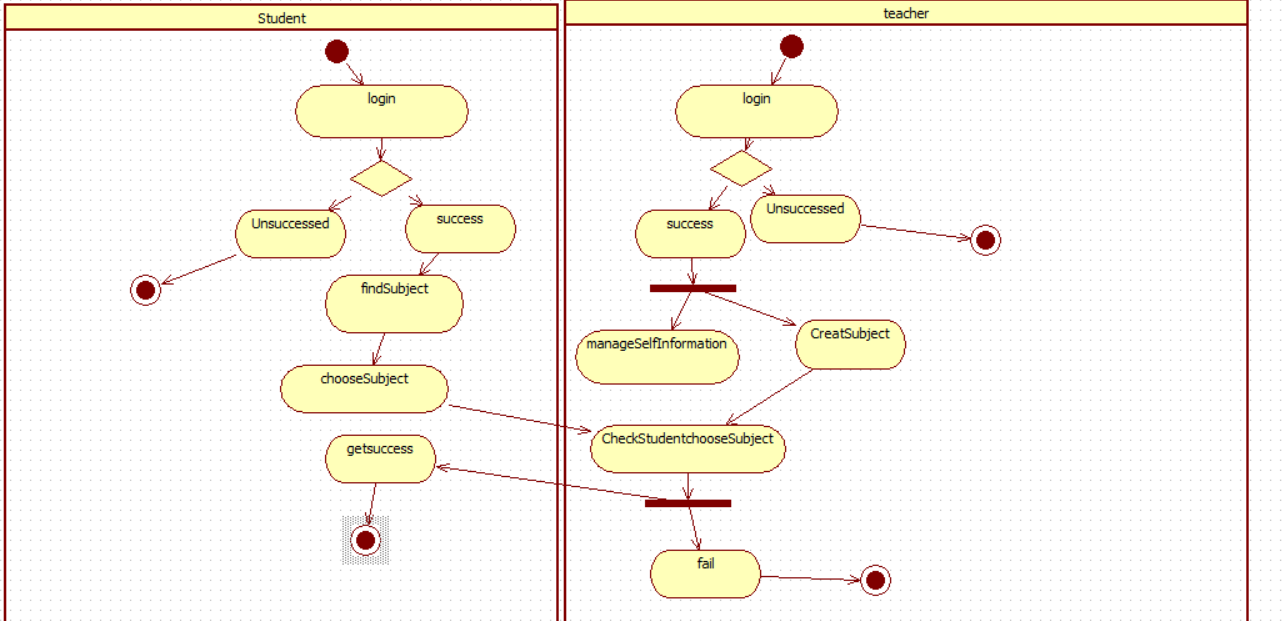
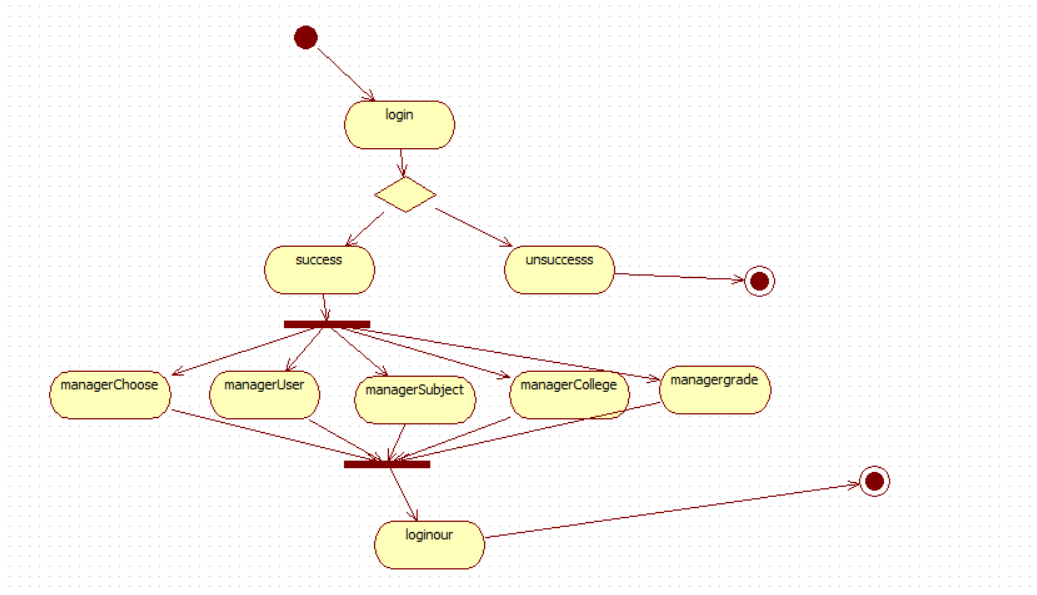


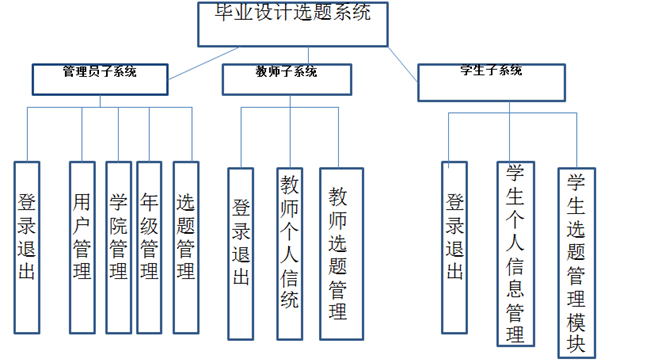
图2-2.1是学生教师流程的活动图。



2.2.2管理员业务流程分析

## 1.2 系统体系结构设计

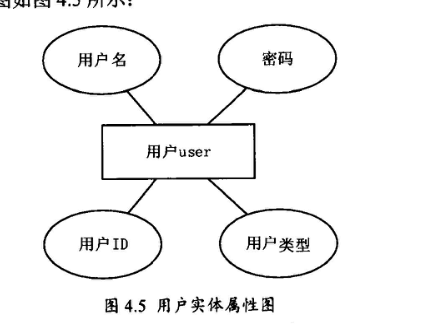
本节将根据需求分析中总结的用户需求设计系统的体系结构。系统的主要功能模块图如图3-1所示：



3-1系统功能模块图

在系统功能模块图的结构中，每一个结点都是一个功能模块。

每一个功能模块都需要根据不同的表完成相同的数据库操作：增加记录、删除记录、修改记录及查询显示记录信息等。

在本系统中，可以把功能模块分为以下主要的功能集合来描述。

（1）用户user保存了系统用户的相关信息，比如：用户ID、用户名、密码、用户类型。

（2）学院college保存了系统中相关学院的信息比如：学院的Id和学院名称。学院college的实体

学院college

（3）学生实体

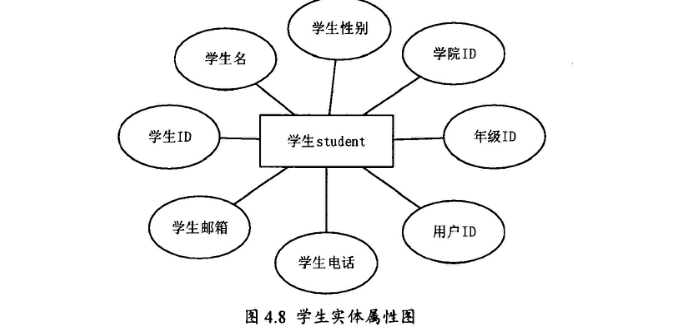


图 3-4 查询功能模块结构图

## 1.3 数据库设计

本系统采用SQL Server 2008数据库作为后台数据库。整个系统共使用了11张数据表，分别为：

（1）user表

user表用于用户信息如表3-2所示。

表3-2 user表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **为空** | **约束** | **备注** |
| Userid | integer（11） | 否 | 主键 | id |
| Username  Password | varchar (15)  varchar (40) | 否  否 |  | 姓名  密码 |
| Usertype  userstate | varchar (36)  tinyint（1） | 否  否 |  | 类型  是否激活 |

（2）teacher表

teacher表用于teacher信息如表3-2所示。

表3-2 user表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **为空** | **约束** | **备注** |
| Teacherid | integer（11） | 否 | 主键 | id |
| Teachername  Teachersex | varchar (15)  varchar (40) | 否  否 |  | 姓名  密码 |
| collegeid  Teacheremail  Teachertelephone  Teacherfaild  Teacherpost  Userid | varchar (36)  varchar (36)  varchar（21）  varchar（23）  varchar(40)  varchar(40) | 否  否  否  否  否  否 | 外键  外键 | 类型  Email  电话  研究方向  职称  用户id |

（3）grade表用于年级信息如表3-2所示。

表3-2 grade表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **为空** | **约束** | **备注** |
| gradeid | integer（11） | 否 | 主键 | id |
| gradename | varchar (15) | 否 |  | 名字 |

（4）Student表

Student表用于Student信息如表3-2所示。

表3-2 user表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **为空** | **约束** | **备注** |
| Studentid | integer（11） | 否 | 主键 | id |
| Studentname  Studentsex | varchar (15)  varchar (40) | 否  否 |  | 姓名  密码 |
| collegeid  Studentemail  Studentelephone  Userid | varchar (36)  varchar (36)  varchar（21）  varchar(40) | 否  否  否  否 | 外键  外键 | 类型  Email  电话  用户id |

（4）subject表

subject表用于subject信息如表3-2所示。

表3-2 user表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **为空** | **约束** | **备注** |
| subjectid | integer（11） | 否 | 主键 | id |
| Subjectname  Subjectcontext | varchar (15)  varchar (40) | 否  否 | 外键  外键 | 课题名字  课题内容 |
| Teacherid  collegeid | varchar (36)  varchar (36) | 否  否 | 外键  外键 | 老师id  学院 |

（4）choose表

choose表用于choose信息如表3-2所示。

表3-2 user表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **为空** | **约束** | **备注** |
| chooseid | integer（11） | 否 | 主键 | id |
| collegeid  subjectid | varchar (15)  varchar (40) | 否  否 | 外键  外键 | 学院  课题 |
| teacherid  Studentid  ischoose | varchar (36)  varchar (36)  varchar（21） | 否  否  否 | 外键  外键 | 老师  学生  是否选择 |

（5）College表

College表用于记录相关学院信息，每个学院具有唯一的学院id，如表3-3所示。

表3-3 College表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **数据类型** | **为空** | **约束** | **备注** |
| collegeid | Integer(10) | 否 | 主键 | 学院编号 |
| collegename | varchar (50) | 否 |  | 学院名字 |

# 

# 第二章 系统的实现

下面主要介绍了登录窗体的设计与实现、系统主界面的设计等、信息管理功能模块的设计与实现、信息查询模块的设计与实现、留言板功能模块的设计与实现等。

## 2.1 登录窗体和系统主界面的设计

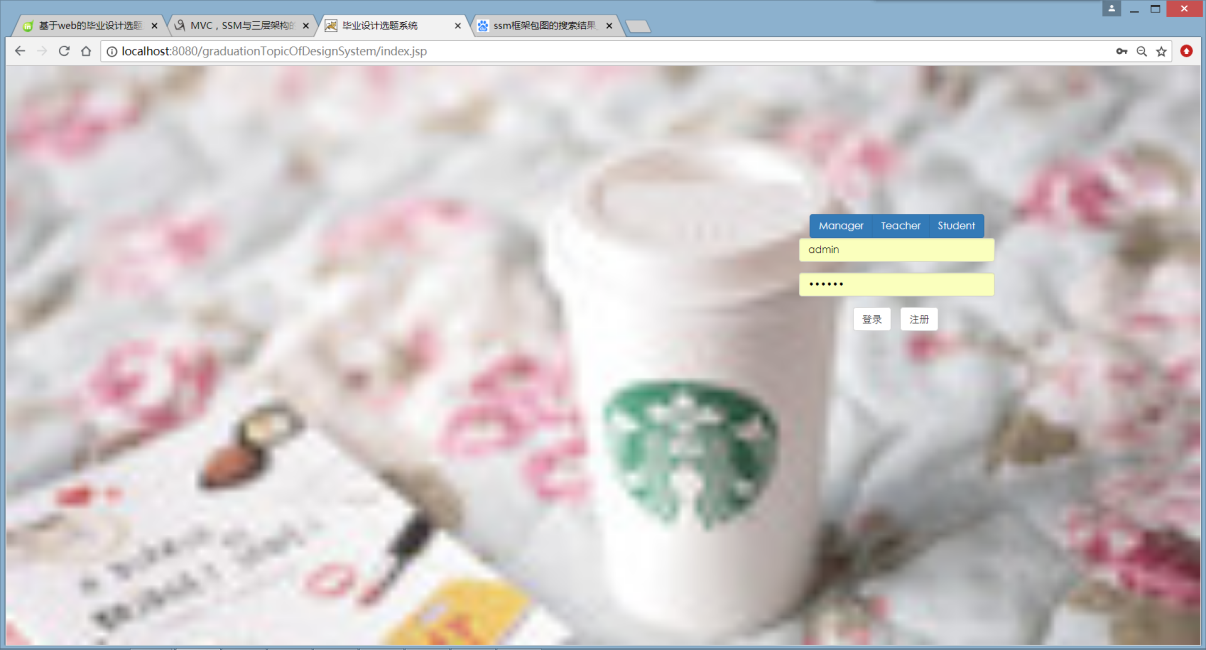
当系统运行时，首先打开的是登录窗体，只有有权限的用户才能进入系统。登录成功后将显示系统主界面。后台数据库中有用户表，此表有四个字段学号、用户类型、密码，此表包含了登录此系统的所有的用户。如果登录用户不能通过用户表的身份验证，此用户就不能进入系统。登录系统的用户有两种身份即教师和学生，用户登录系统，建立与数据库中用户表的连接，通过用户表中的“权限”字段来区用户的身份。用户输入自己的用户名和密码，按回车键进入系统相应的主界面，系统的登录界面如图4-1所示:

图4-1系统登录界面

登录窗体中用到的主要的代码如下：

User user = userService.findByPrimaryKey(username, password,usertype);

**if**(user != **null**){

// 将用户对象添加到Session

session.setAttribute("USER\_SESSION", user);

// 跳转到主页面

// return "customer";

**if**(usertype=="Manager"||usertype.equals("Manager"))//manager

**return** "redirect:manageruser/userList.action";

**else** **if**(usertype=="Teacher"||usertype.equals("Teacher"))//teacher

**return** "redirect:teacher/subjectListByTeacher.action";

**else** **if**(usertype=="Student"||usertype.equals("Student"))//student

**return** "redirect:student/subjectList.action";

}

model.addAttribute("msg", "账号或密码错误，请重新输入！");

// 返回到登录页面

**return** "Login";

## 2.2学生模块设计与实现

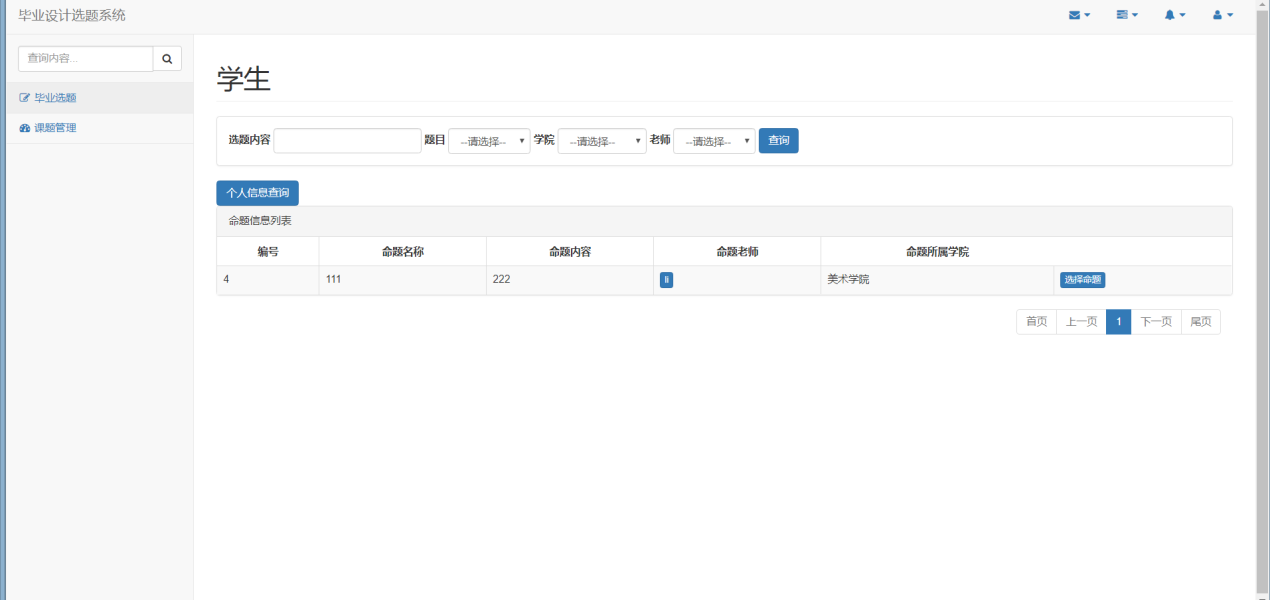
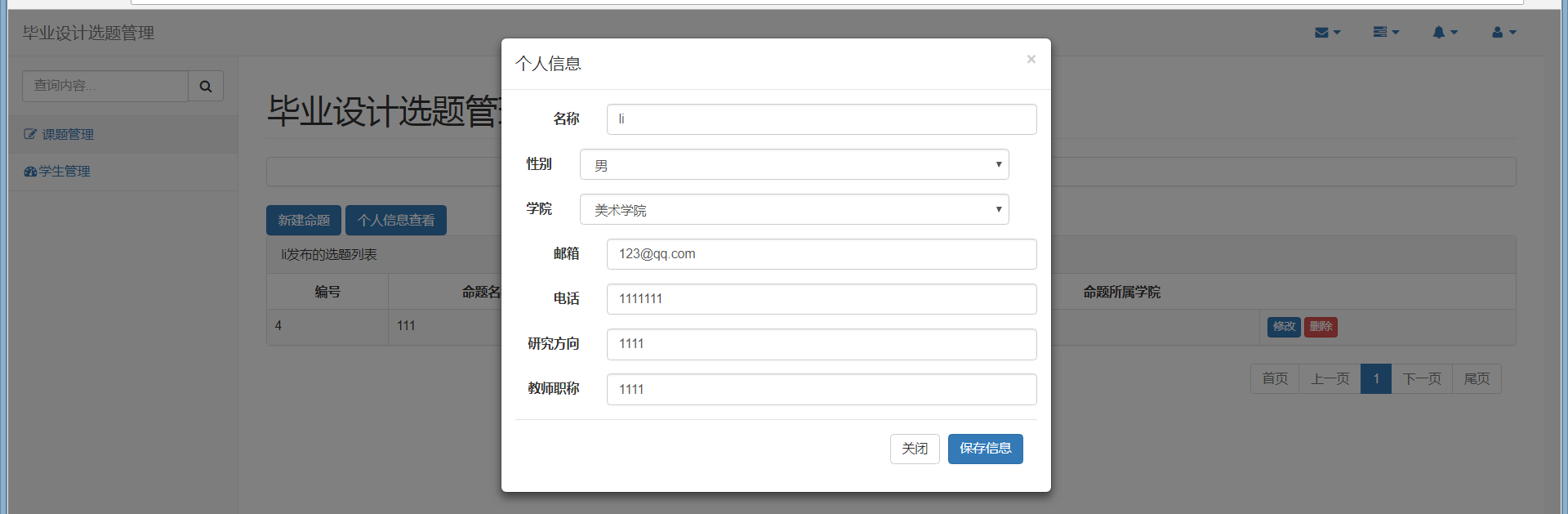


图4-2 学生界面

 4-3教师操作界面

点击添加按钮，实现添加新纪录，所实现功能的主要代码如下：

**public** String list(@RequestParam(defaultValue="1")Integer page,

@RequestParam(defaultValue="10")Integer rows,

Model model,HttpSession session) {

User user = (User) session.getAttribute("USER\_SESSION");

// 将当前用户id存储在客户对象中

**int** id = teacherService.findByUserID(user.getUserid()).getTeacherid();

Page <SubjectView> subjects = subjectService.findSubjectListByTeacherID(page, rows, String.*valueOf*(id));

model.addAttribute("page", subjects);

// 学院

List<College> collegeType = collegeService.findCollegeList(**null**);

//年级

List<Grade> gradeType = gradeService.findGradeList(**null**);

// 老师

List<Teacher> teachers= teacherService.findTeacherList(**null**);

Teacher teacher= teacherService.findByUserID(user.getUserid());

List<Subject> subjectType = subjectService.findAllSubjectList(**null**);// 添加参数

model.addAttribute("subjectType", subjectType);//subjectType

model.addAttribute("collegeType", collegeType);//fromType

model.addAttribute("gradeType", gradeType);//industryType

model.addAttribute("teachers", teachers);//levelType

model.addAttribute("teacher", teacher);//levelType

**return** "TeacherSubjectManage";

}

@RequestMapping("/teacher/updateSubject.action")

**public** String subjectUpdate(Subject subject, HttpSession session) {

String inforError="";

**if**(subject.getSubjectid() ==**null** || subject.getSubjectid()==0)

{

**int** rows =subjectService.creatSubject(subject);

**if**(rows > 0){

**return** "redirect:subjectListByTeacher.action";

}**else**{

**return** "error";

}

}

**else**

{ **int** rows = subjectService.updateByPrimaryKey(subject);

**if**(rows > 0){

**return** "redirect:subjectListByTeacher.action";

}**else**{

**return** "error";

}}

}

@RequestMapping("/teacher/updateTeacherInfor.action")

**public** String teacherInforUpdate(Teacher teacher, HttpSession session) {

String inforError="";

**if**(teacher.getTeacherid() ==**null** || teacher.getTeacherid()==0)

{

**int** rows =teacherService.creatTeacher(teacher);

**if**(rows > 0){

**return** "redirect:subjectListByTeacher.action";

}**else**{

**return** "error";

}

}

**else**

{ **int** rows = teacherService.updateByPrimaryKey(teacher);

**if**(rows > 0){

**return** "redirect:subjectListByTeacher.action";

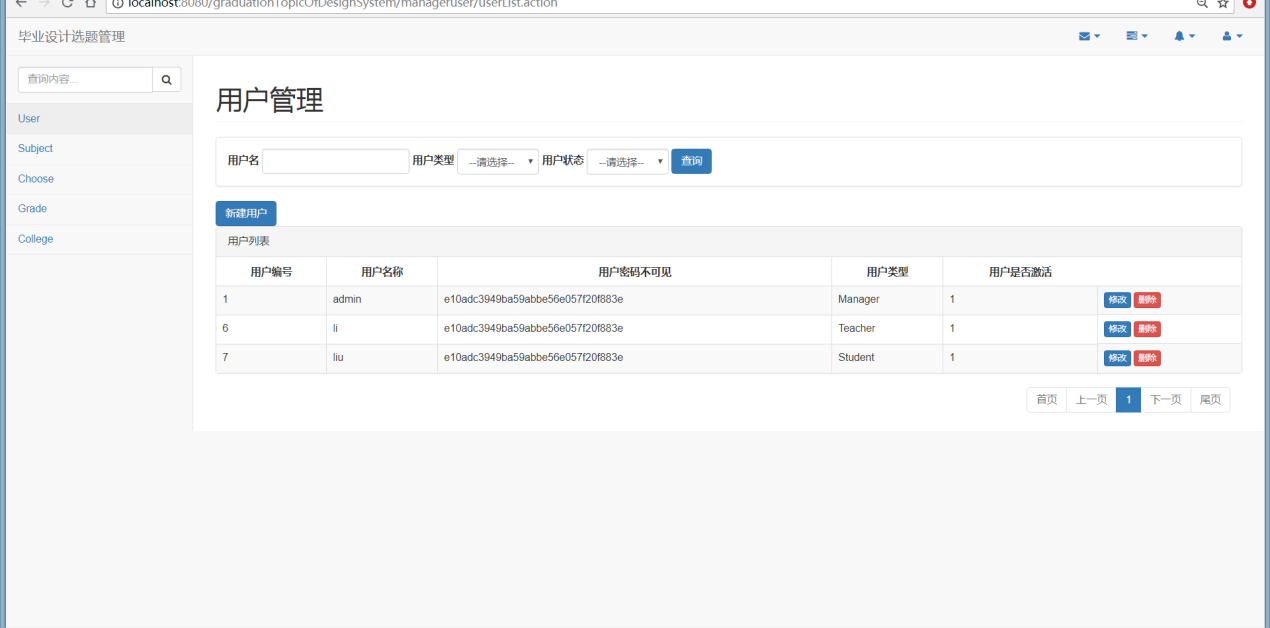
}**else**{

**return** "error";

}}

}

## 2.3 管理员模块的设计与实现

核心代码实现如下：

@RequestMapping(value = "/manageruser/userList.action")

**public** String userList(@RequestParam(defaultValue="1")Integer page,

@RequestParam(defaultValue="10")Integer rows,

String AlltimeUsertype, String allTimeUsername, String AlltimeUserstate,

Model model,HttpSession session) {

// 条件查询所有客户

User user = (User) session.getAttribute("USER\_SESSION");

// 将当前用户id存储在客户对象中

Page <User> userpage = userService.findUserListPage(page, rows, allTimeUsername, AlltimeUsertype, AlltimeUserstate);

model.addAttribute("page", userpage);

// 学院

model.addAttribute("AlltimeUsertype", AlltimeUsertype);//custName

model.addAttribute("allTimeUsername", allTimeUsername);//custSource

model.addAttribute("AlltimeUserstate", AlltimeUserstate);//custIndustry

**return** "ManageUser";

}