****

**WEB 高级编程**

**课程作业**

**Web高级编程大作业设计方案**

作业项目名称 12. 学生选课系统

指导教师及职称 张恒峰 讲师

开课学期 2018 至 2019 学年 第 2 学

[1 需求分析 1](#_Toc12480101)

[1.1需求概述 1](#_Toc12480102)

[1.2 用例分析与描述 2](#_Toc12480103)

[1.2.1 用例分析 2](#_Toc12480104)

[1.2.1 用例描述 2](#_Toc12480105)

[1.4 参考文献 6](#_Toc12480106)

[2 系统设计 6](#_Toc12480107)

[2.1 多层结构设计 6](#_Toc12480108)

[2.2 问题域设计 6](#_Toc12480109)

[2.2.1 域模型设计 6](#_Toc12480110)

[2.2.2 Service接口设计 7](#_Toc12480111)

[2.3 持久化设计 7](#_Toc12480112)

[2.3.1 数据库设计 7](#_Toc12480113)

[2.3.2 数据访问设计 11](#_Toc12480114)

[2.4 界面设计 11](#_Toc12480115)

[2.4.1 页面链接关系 11](#_Toc12480116)

[2.4.2 页面设计 12](#_Toc12480117)

[3 系统实现 14](#_Toc12480118)

[3.1 系统开发环境 14](#_Toc12480119)

[3.2 表达层 14](#_Toc12480120)

[3.3 业务层 15](#_Toc12480121)

[3.4 持久化层 15](#_Toc12480122)

# 1 需求分析

## 1.1需求概述

学生选课系统是一个学校不可缺少的部分，它的内容对于学校的决策者和管理者来说都至关重要，所以学生选课系统应该能够为用户提供充足的信息和快捷的査询手段。

## 1.2 用例分析与描述

## 1.2.1 用例分析

本学生选课系统需要满足来自四方面的需求，这三个方面分别是学生、教师、教务和机房。

学生的需求是査询课程，查询课表，查询成绩；

教师对选课系统学生选课情况进行操作，提交开课信息，申请实验，成绩录入；

教务需要审核开课信息，审核选课信息，安排授课时间、地点。

机房需要管理机房信息、安排实验。

图片包含 文字, 地图

描述已自动生成通过上述分析，可得到系统的用例图如下：

## 1.2.1 用例描述

|  |
| --- |
| 用例名称：申请选课 |
| 描述：学生通过本系统选择课程 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:学生 |
| 前置条件：学生已登录 |
| 主事件流:   1. 学生成功登录系统 2. 选择申请课程功能 3. 选择目标课程 4. 点击选课   A1：选课人数达到上限  A2：选课时间与已有课程产生冲突   1. 选课结束 |
| 其他事件流：  A1: 选课人数达到上限  A2：选课时间与已有课程产生冲突 |
| 后置条件：将向光选课信息添加到数据库里 |
| 特殊需求：无 |

|  |
| --- |
| 用例名称：查询课表 |
| 描述：学生通过系统查询课表 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:学生 |
| 前置条件：学生已登录 |
| 主事件流:  1.学生成功登录系统  2.选择查询课表功能  A1：未选择课程，无法显示课表  5.查询结束 |
| 其他事件流：  A1：未选择课程，无法显示课表 |
| 后置条件： |
| 特殊需求： |

|  |
| --- |
| 用例名称：录入学生成绩 |
| 描述：教授通过本系统录入学生的成绩 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:教授 |
| 前置条件：教授已登录 |
| 主事件流:   1. 教授成功登录本系统 2. 选择录入学生成绩功能   A1：未开设课程   1. 录入成绩 2. 结束 |
| 其他事件流：  A1：未开设课程 |
| 后置条件： |
| 特殊需求： |

|  |
| --- |
| 用例名称：安排授课时间和地点 |
| 描述：教务处通过本系统安排课程的授课时间和地点 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:教务 |
| 前置条件：成功登录 |
| 主事件流:   1. 成功登录系统 2. 选择安排授课时间和地点按钮 3. 选择需要安排的课程 4. 安排时间和地点   A1:时间冲突  A2：地点冲突  5.安排结束 |
| 其他事件流：  A1:时间冲突  A2：地点冲突 |
| 后置条件：将安排的信息储存到数据库中 |
| 特殊需求： |

|  |
| --- |
| 用例名称：审核选课信息 |
| 描述：教务处通过本系统审核学生的选课信息 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:教务处 |
| 前置条件：成功登录 |
| 主事件流:   1. 成功登录系统后 2. 选择审核学生选课信息按钮 3. 进行审核 4. 结束审核 |
| 其他事件流： 无 |
| 后置条件：无 |
| 特殊需求：无 |

|  |
| --- |
| 用例名称：安排实验 |
| 描述：机房通过本系统接受到安排试演的需求 |
| 标识符:uc1 |
| 优先级：A(高) |
| 角色:机房 |
| 前置条件：已登录 |
| 主事件流:   1. 机房成功登录系统 2. 选择安排实验按钮 3. 根据信息安排实验   A1:实验时间或地点冲突   1. 结束 |
| 其他事件流：  A1:实验时间或地点冲突 |
| 后置条件： |
| 特殊需求： |

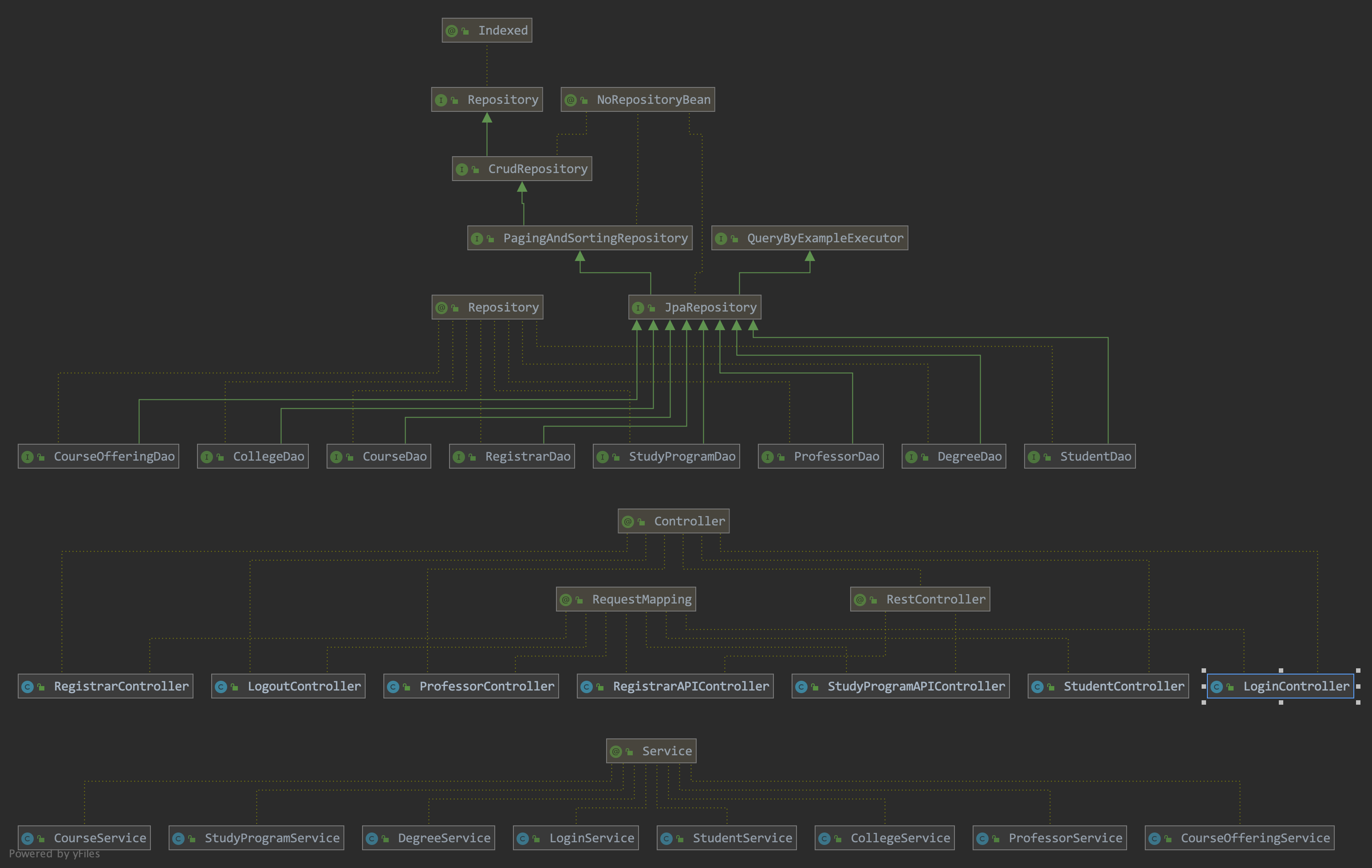
## 1.4 参考文献

[1] 齐志昌，谭庆平，宁洪.软件工程(第二版)[M].北京.高等教育出版社.2004

[2] 王珊，萨师煊.数据库系统概论（第四版）[M].北京.高等教育出版社.2006

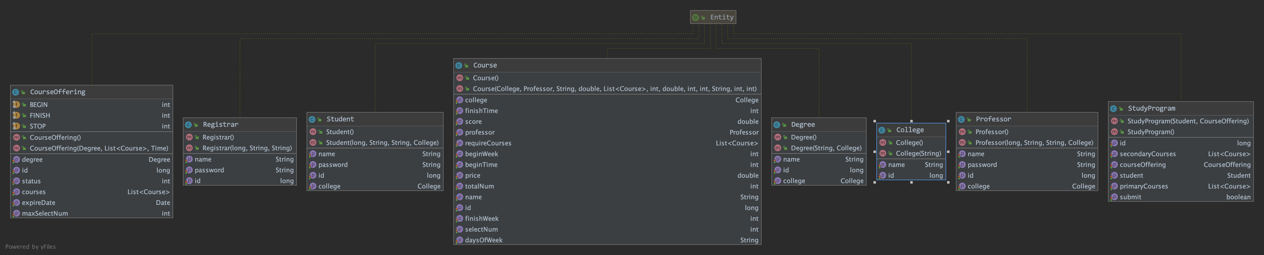
# 2 系统设计

## 2.1 多层结构设计

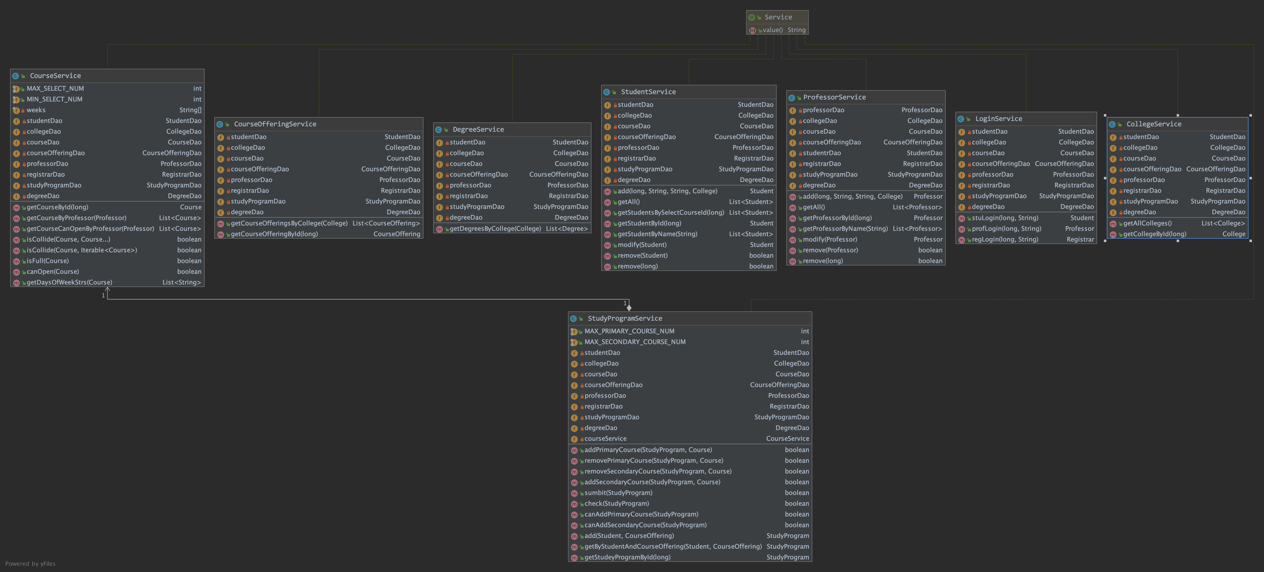


## 2.2 问题域设计

### 2.2.1 域模型设计



### 2.2.2 Service接口设计



## 2.3 持久化设计

### 2.3.1 数据库设计

根据以上给出的实体关系模型可以得到以下数据库物理物理模型。

表格详细情况如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| college | id | bigint(20) | auto\_increment | NULL | 否 |  |
| college | name | varchar(255) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| course | id | bigint(20) | auto\_increment | NULL | 否 |  |
| course | begin\_time | int(11) |  | NULL | 否 |  |
| course | begin\_week | int(11) |  | NULL | 否 |  |
| course | days\_of\_week | varchar(255) |  | NULL | 否 |  |
| course | finish\_time | int(11) |  | NULL | 否 |  |
| course | finish\_week | int(11) |  | NULL | 否 |  |
| course | name | varchar(255) |  | NULL | 否 |  |
| course | price | double |  | NULL | 否 |  |
| course | score | double |  | NULL | 否 |  |
| course | select\_num | int(11) |  | NULL | 是 |  |
| course | total\_num | int(11) |  | NULL | 否 |  |
| course | college\_id | bigint(20) |  | NULL | 否 |  |
| course | professor\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| course\_offering | id | bigint(20) | auto\_increment | NULL | 否 |  |
| course\_offering | expire\_date | datetime(6) |  | NULL | 否 |  |
| course\_offering | status | int(11) |  | NULL | 否 |  |
| course\_offering | degree\_id | bigint(20) |  | NULL | 否 |  |
| course\_offering | max\_select\_num | int(11) |  | NULL | 是 |  |
| course\_offering\_courses | course\_offering\_id | bigint(20) |  | NULL | 否 |  |
| course\_offering\_courses | courses\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| course\_require\_courses | course\_id | bigint(20) |  | NULL | 否 |  |
| course\_require\_courses | require\_courses\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| degree | id | bigint(20) | auto\_increment | NULL | 否 |  |
| degree | name | varchar(255) |  | NULL | 否 |  |
| degree | college\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| professor | id | bigint(20) |  | NULL | 否 |  |
| professor | name | varchar(255) |  | NULL | 否 |  |
| professor | password | varchar(255) |  | NULL | 否 |  |
| professor | college\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| registrar | id | bigint(20) |  | NULL | 否 |  |
| registrar | name | varchar(255) |  | NULL | 否 |  |
| registrar | password | varchar(255) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| student | id | bigint(20) |  | NULL | 否 |  |
| student | name | varchar(255) |  | NULL | 否 |  |
| student | password | varchar(255) |  | NULL | 否 |  |
| student | college\_id | bigint(20) |  | NULL | 是 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| study\_program | id | bigint(20) | auto\_increment | NULL | 否 |  |
| study\_program | submit | bit(1) |  | NULL | 否 |  |
| study\_program | course\_offering\_id | bigint(20) |  | NULL | 否 |  |
| study\_program | student\_id | bigint(20) |  | NULL | 是 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| study\_program\_primary\_courses | study\_program\_id | bigint(20) |  | NULL | 否 |  |
| study\_program\_primary\_courses | primary\_courses\_id | bigint(20) |  | NULL | 否 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表名 | 字段名 | 数据类型 | 主键 | 默认值 | 是否允许为空 | 字段说明 |
| study\_program\_secondary\_courses | study\_program\_id | bigint(20) |  | NULL | 否 |  |
| study\_program\_secondary\_courses | secondary\_courses\_id | bigint(20) |  | NULL | 否 |  |

### 2.3.2 数据访问设计

图片包含 黑色, 屏幕截图, 墙壁

描述已自动生成

## 2.4 界面设计

### 2.4.1 页面链接关系

表2.1 URL对应功能说明（pets）

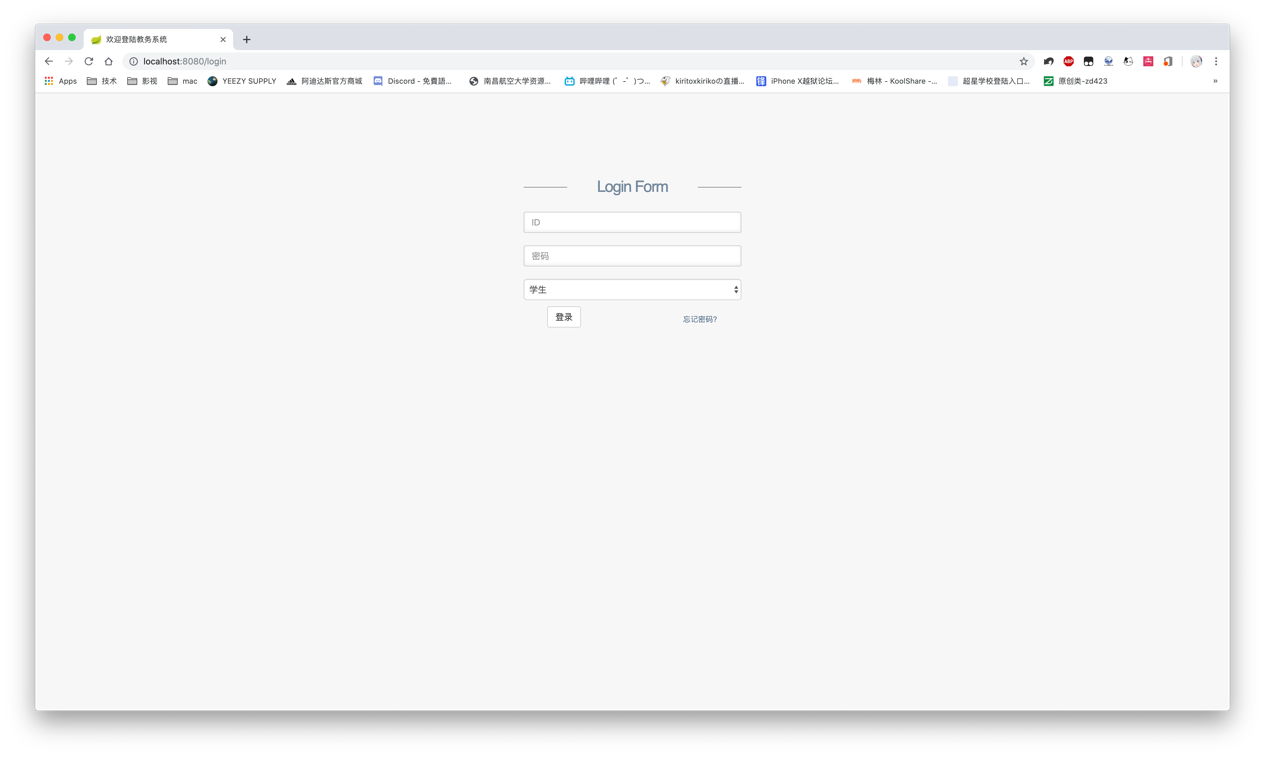
|  |  |  |  |
| --- | --- | --- | --- |
| HTTP verb | URL | action（Controller类的方法） | used for |
| GET | /pets/new | add | return an HTML form for creating a new pet |
| GET | /login | login | Login website |
| GET | /studentt | student | return an HTML web for student course list |
| POST | /login | Login | Request login form |
| GET | /teacher | teacher | Return an HTML web for teachers’ course list |

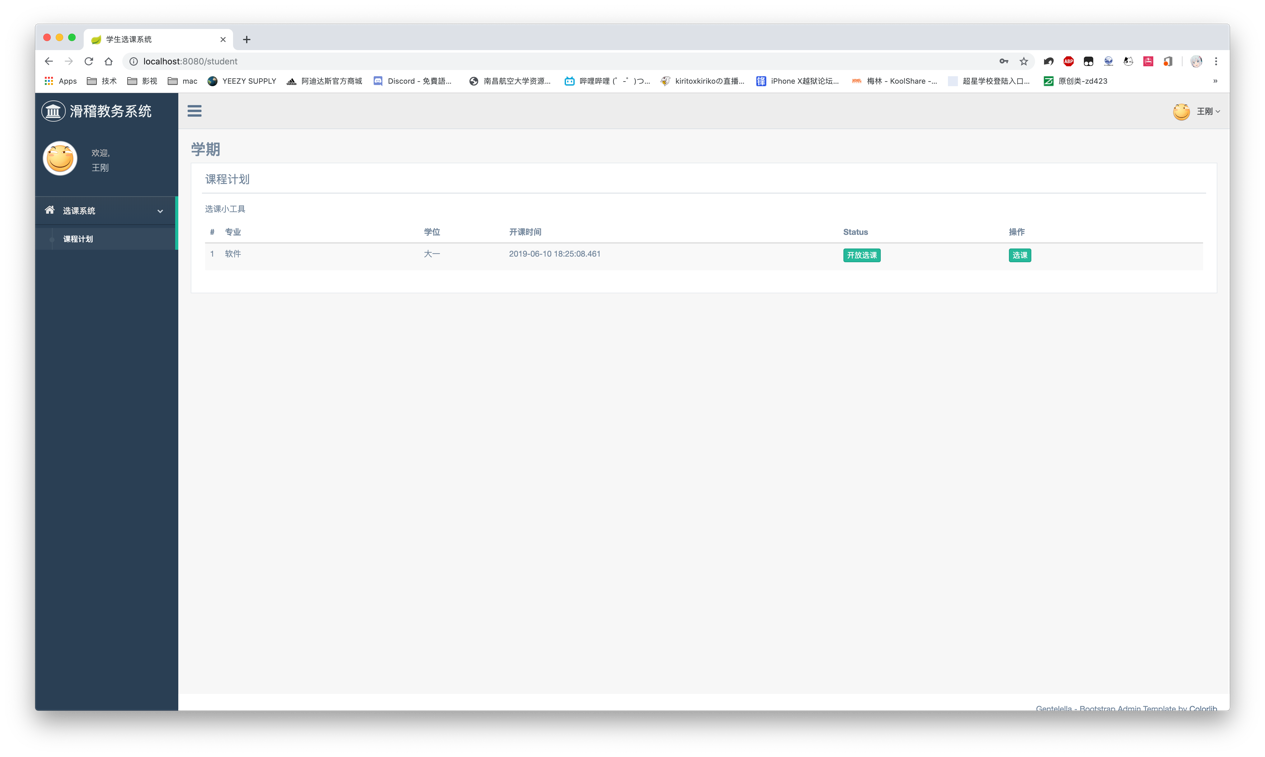
图片包含 屏幕截图

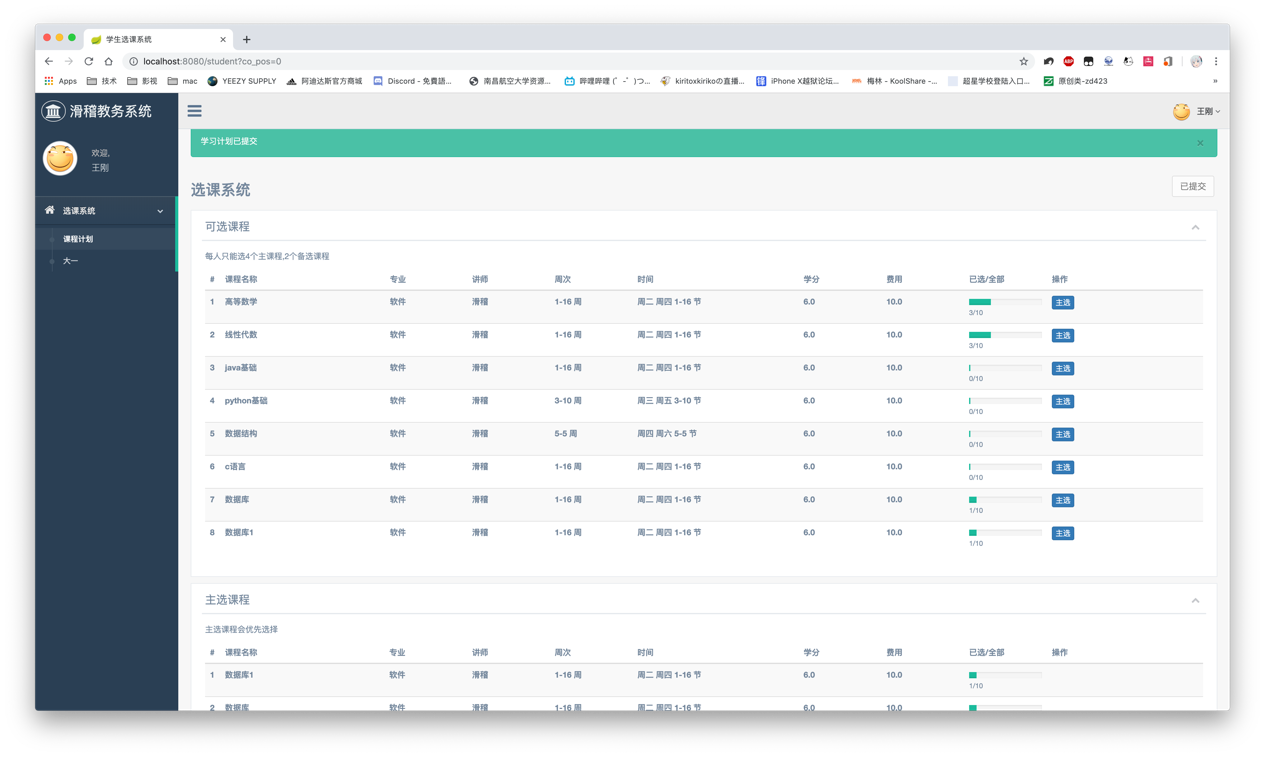
描述已自动生成

### 2.4.2 页面设计

（给出页面设计及其操作流程。）





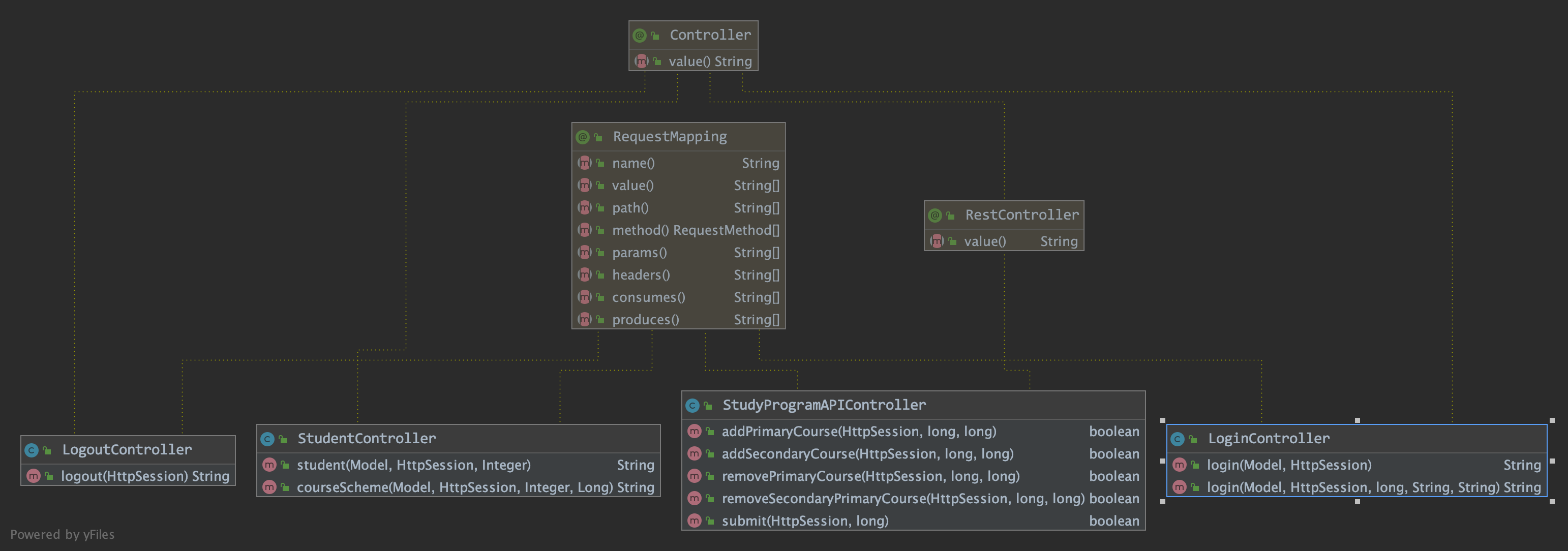


# 3 系统实现

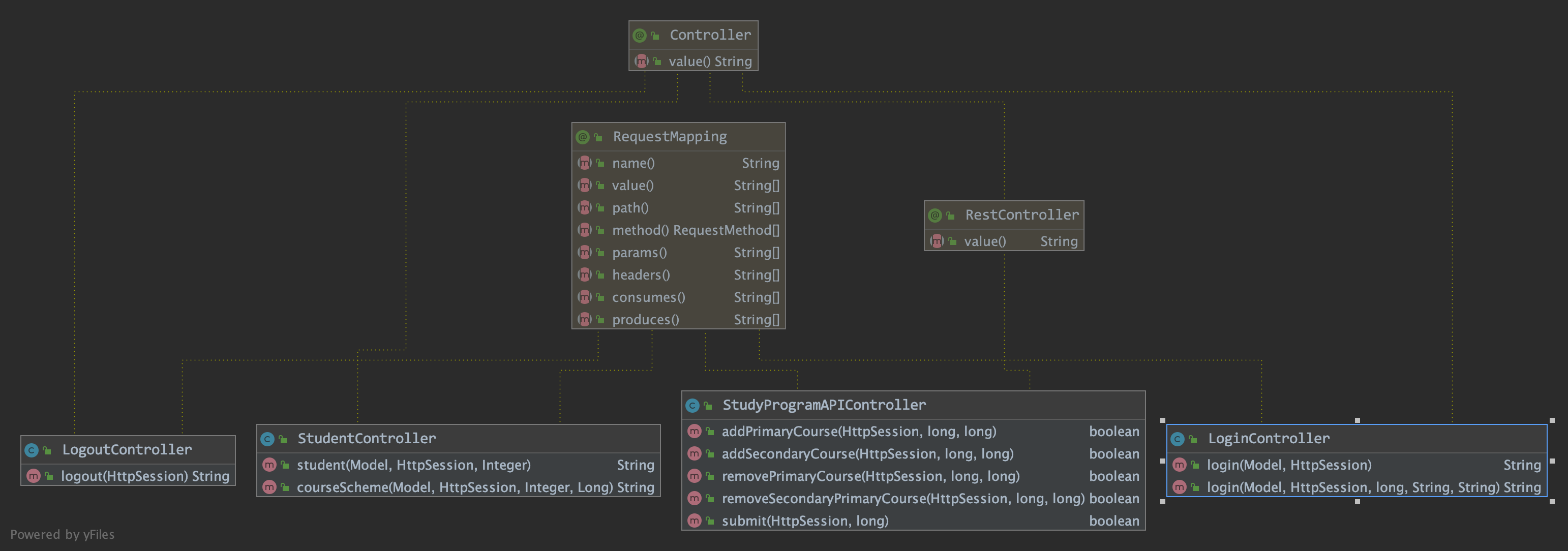
## 3.1 系统开发环境

* Java SDK 1.8
* spring boot 2.1.4
* (Optional) MySQL 5.x with MySQL Connector/J 5.x

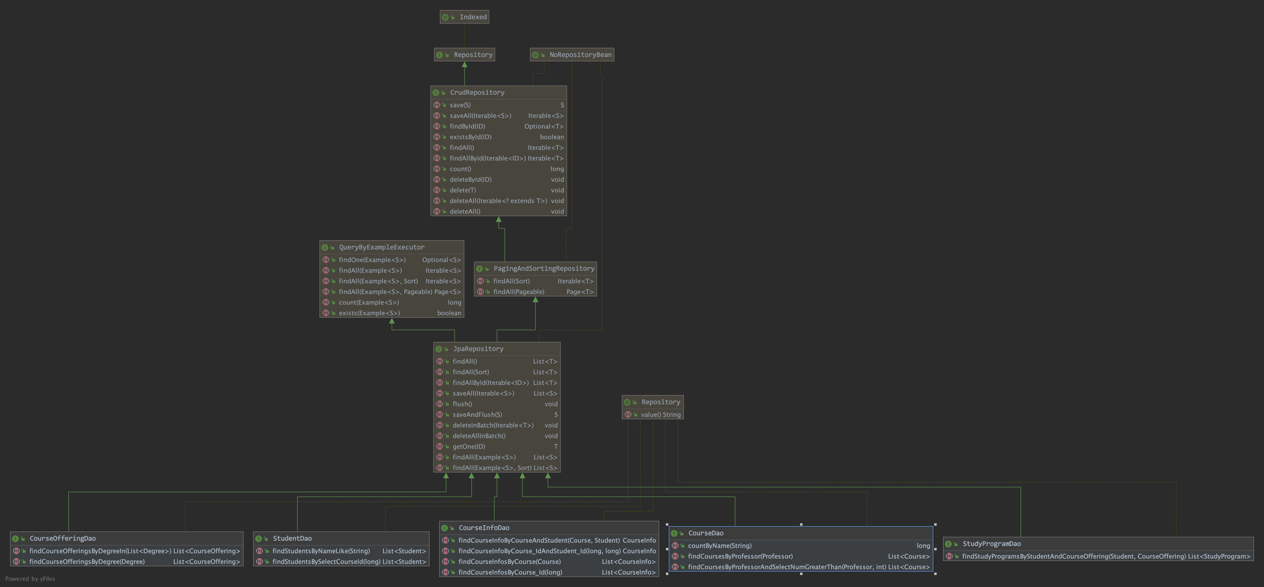
## 3.2 表达层



## 3.3 业务层



## 3.4 持久化层



**4 系统测试**

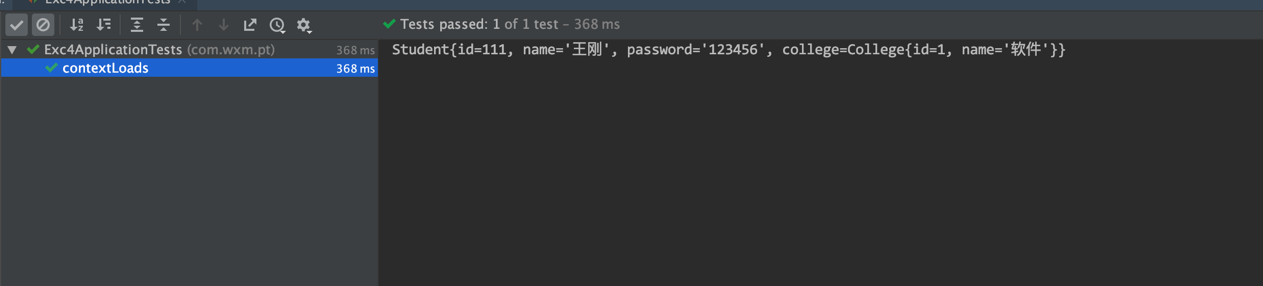
采用junit进行测试

对系统的每个方法都进行了测试

代码如下

package com.wxm.pt;  
  
import com.wxm.pt.dao.\*;  
import com.wxm.pt.entity.\*;  
import com.wxm.pt.service.CourseService;  
import com.wxm.pt.service.ProfessorService;  
import com.wxm.pt.service.StudentService;  
import com.wxm.pt.service.StudyProgramService;  
import org.junit.Test;  
import org.junit.runner.RunWith;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.boot.test.context.SpringBootTest;  
import org.springframework.test.context.junit4.SpringRunner;  
  
import java.sql.Time;  
import java.util.ArrayList;  
import java.util.List;  
  
@RunWith(SpringRunner.class)  
@SpringBootTest  
public class Exc4ApplicationTests {  
 @Autowired  
 private StudentDao studentDao;  
 @Autowired  
 private CollegeDao collegeDao;  
 @Autowired  
 private CourseDao courseDao;  
 @Autowired  
 private CourseOfferingDao courseOfferingDao;  
 @Autowired  
 private ProfessorDao professorDao;  
 @Autowired  
 private RegistrarDao registrarDao;  
 @Autowired  
 private StudyProgramDao studyProgramDao;  
 @Autowired  
 private DegreeDao degreeDao;  
 @Autowired  
 private StudentService studentService;  
 @Autowired  
 private ProfessorService professorService;  
 @Autowired  
 private StudyProgramService studyProgramService;  
 @Autowired  
 private CourseService courseService;  
 @Test  
 public void contextLoads() {  
 College college=new College("软件");  
 college=collegeDao.saveAndFlush(college);  
 Degree degree=new Degree("大一",college);  
 degree=degreeDao.save(degree);  
  
 List<Student> students=new ArrayList<>();  
 students.add(studentService.add(111,"王刚","123456",college));  
 students.add(studentService.add(222,"李强","123456",college));  
 students.add(studentService.add(333,"刘建华","123456",college));  
 students.add(studentService.add(444,"赵四","123456",college));  
 Professor professor=professorService.add(111,"滑稽","123456",college);  
 Registrar registrar=registrarDao.save(new Registrar(111,"admin","admin"));  
 //  
  
 List<Course> courses=new ArrayList<>();  
 courses.add(new Course(college,professor,"高等数学",6,null,10,10,1,16,"1 3",1,2,null));  
 courses.add(new Course(college,professor,"线性代数",6,null,10,10,1,16,"1 3",1,2,null));  
 courses.add(new Course(college,professor,"java基础",6,null,10,10,1,16,"1 3",5,6,null));  
 courses.add(new Course(college,professor,"python基础",6,null,10,10,3,10,"2 4",3,4,null));  
 courses.add(new Course(college,professor,"数据结构",6,null,10,10,5,5,"3 5",3,4,null));  
 courses.add(new Course(college,professor,"c语言",6,null,10,10,1,16,"1 3",7,8,null));  
 courses.add(new Course(college,professor,"数据库",6,null,10,10,1,16,"1 3",3,4,null));  
 courseDao.saveAll(courses);  
 List<Course> courses1=new ArrayList<>();  
 courses1.add(courseService.getCourseById(1));  
 courses.add(new Course(college,professor,"数据库1",6,courses,10,10,1,16,"1 3",3,4,null));  
 courseDao.saveAll(courses);  
 Time time=new Time(System.currentTimeMillis()+1000\*60\*10);//当前时间+10min  
 CourseOffering courseOffering=new CourseOffering(degree,courses,time);  
 courseOffering=courseOfferingDao.save(courseOffering);  
 // courses=courseDao.findAll();  
 //  
 //  
 Student student1=students.get(0);  
 System.out.println(student1);  
// Course tempC=courses.get(4);  
// tempC.setSelectNum(3);  
// courseDao.save(tempC);  
// for(Course c:courses){  
// c.setSelectNum(3);  
// courseDao.save(c);  
// }  
// StudyProgram studyProgram=studyProgramService.add(student1,courseOffering);  
// for(Course c:courses){  
// System.out.println("addP "+studyProgramService.addPrimaryCourse(studyProgram,c));  
// }  
// System.out.println("removeP "+studyProgramService.removePrimaryCourse(studyProgram,courses.get(0)));  
// System.out.println(courses.get(0));  
// System.out.println("addP "+studyProgramService.addPrimaryCourse(studyProgram,courses.get(0)));  
// System.out.println(courses.get(0));  
// //  
// System.out.println("addS "+studyProgramService.addSecondaryCourse(studyProgram,courses.get(4)));  
// studyProgram=studyProgramService.getByStudentAndCourseOffering(student1,courseOffering);  
// System.out.println(studyProgram);  
 //  
// System.out.println("submit "+studyProgramService.sumbit(studyProgram));  
// System.out.println("check "+studyProgramService.check(studyProgram));  
// System.out.println(studyProgram);  
 }  
  
}

测试结果如下



**5 总结与体会**

本项目使用了spingboot

spingboot使用Java或Groovy开发基于Spring的应用程序非常容易。

它减少了大量的开发时间并提高了生产力。

它避免了编写大量的样板代码，注释和XML配置。

Spring Boot应用程序与其Spring生态系统(如Spring JDBC，Spring ORM，Spring Data，Spring Security等)集成非常容易。

它遵循“自用默认配置”方法，以减少开发工作量。

它提供嵌入式HTTP服务器，如Tomcat，Jetty等，以开发和测试Web应用程序非常容易。

它提供CLI(命令行界面)工具从命令提示符，非常容易和快速地开发和测试Spring Boot(Java或Groovy)应用程序。

它提供了许多插件来开发和测试Spring启动应用程序非常容易使用构建工具，如Maven和Gradle。

它提供了许多插件，以便与嵌入式和内存数据库工作非常容易。

并且使用了spring data jpa+Hibernate作为数据持久层

jpa是Sun官方提出的Java持久化规范。它为Java开发人员提供了一种对象/关联映射工具来管理Java应用中的关系数据。它的出现主要是为了简化现有的持久化开发工作和整合ORM技术  
结束Hibernate、TopLink等ORM框架各自为营的局面。JPA是在充分吸收了现有Hibernate、TopLink等ORM框架的基础上发展起来的，具有易于使用，伸缩性强等优点。  
JPA是一套规范，不是一套产品。Hibernate是一套产品，如果这些产品实现了JPA规范，那么我们可以叫它们为JPA的实现产品。使用JPA，就可以把我们的应用完全从Hibernate中解脱出来

spring data jpa是在JPA规范下提供了Repository层的实现，但是使用哪一种ORM需要你来决定。虽然ORM框架都实现了JPA规范，但是在不同的ORM框架之间切换仍然需要编写不同的代码 ( Repository/Dao 层的业务代码 )，而通过使用Spring data jpa能够方便大家在不同的ORM框架之间进行切换而不要更改代码。并且spring data jpa 对Repository层封装的很好，也省去了不少的麻烦。

页面前段采用了jquery框架

jQuery是一个快速、简洁的JavaScript框架，是继Prototype之后又一个优秀的JavaScript代码库（或JavaScript框架）。jQuery设计的宗旨是“write Less，Do More”，即倡导写更少的代码，做更多的事情。它封装JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化HTML文档操作、事件处理、动画设计和Ajax交互。

并且和bootstrap一起使用

Bootstrap是美国Twitter公司的设计师Mark Otto和Jacob Thornton合作基于HTML、CSS、JavaScript 开发的简洁、直观、强悍的前端开发框架，使得 Web 开发更加快捷。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言Less写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，包括NASA的MSNBC（微软全国广播公司）的Breaking News都使用了该项目。 [1] 国内一些移动开发者较为熟悉的框架，如WeX5前端开源框架等，也是基于Bootstrap源码进行性能优化而来。