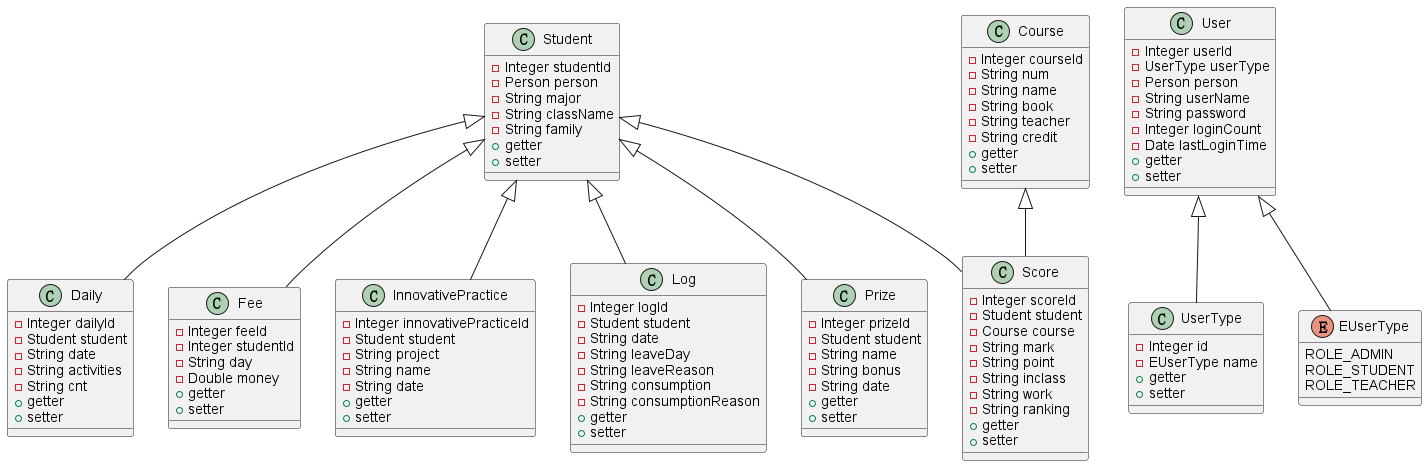
|  |
| --- |
|  |
| **学生管理系统**  **课设报告** |
| 程前，姚仁广  童慎豪，王子成，陈方圆 |
| 二〇二三年 5 月 |



系统实体关系图

**第一章**

1. **系统总体功能要求**

（1）学生学院班级，联系方式、入学前信息、家庭信息等基本信息的管理

（2）学习信息管理，包括课程基本信息，课程中心（教材、课件、参考资料等），选课信息、考勤信息、作业信息、成绩信息等

（3）学生社会实践、学科竞赛、科技成果、培训讲座、创新项目、校外实习等创新实践信息管理

（4）学生荣誉信息管理，包括获得的各种称号奖励等

（5）学生体育活动、外出旅游、文艺演出、聚会、等日常活动管理。

（6）学生外出请假信息和生活学习消费等日志信息管理

（7）学生个人信息的统计汇入统计数据库

（8）学生各种信息的查询统计、综合绩分的计算（可自行设计公式）和学生个人画像、个人简历的生成打印

1. **软件开发平台**

我们课题开发框架采用SpringBoot，它由三部分组成：model(实体)、repository（存储）、service（服务）。

model映射数据库表， repository负责数据的查询修改操作，service负责业务逻辑处理，由于业务逻辑处理比较简单，本课程的框架没有单独设计Service，它的作用由controller代码实现。

View通常由前端技术实现，我们课程提供的是基于JavaFx的实现， 前端框架根据根据后端描述的页面结构和命名匹配规则，将后端传过来的数据按照YAML描述进行渲染，显示需要展现的数据，这部分开发框架自动完成。

Controller层是Model与View之间沟通的桥梁，负责分派用户请求并调用service， repository进行业务处理，并把结果返回View（前端框架）。

1. **小组成员分工表：**

|  |  |
| --- | --- |
| 程前 | 负责后端的功能实现和与前端的一些链接等等 |
| 姚仁广 | 负责前端界面的绘制优化修改 |
| 童慎豪 |  |
| 王子成 |  |
| 陈方圆 |  |

1. **系统需求分析**
2. **需求背景**

每个学生都是一个独立的个体，有个人信息、课程信息、学生荣誉、日常活动、创新实践和日志等多种信息。为了方便记录管理，我们开发了一个学生管理系统，旨在记录学生的各种信息，同时允许管理者对信息进行实时编辑和修改。

学生系统是学生和老师之间的桥梁，便于老师对学生的信息进行管理，提高了效率，同时允许对信息进行实时更新。

1. **价值评估**

此系统是在一定的架构下进行开发的，开发周期较长，但是系统的价值高，很大程度上提高了学生信息管理的效率和准确性，同时附带个人简历的生成，提供下载功能。

经过评估，该系统可以较好的保证网页转化率，因为信息是处于不断更新状态，所以系统也支持可持续更新以及修改的状态。

综合评估，该需求的价值满足产品上线条件。

1. **产品简介**

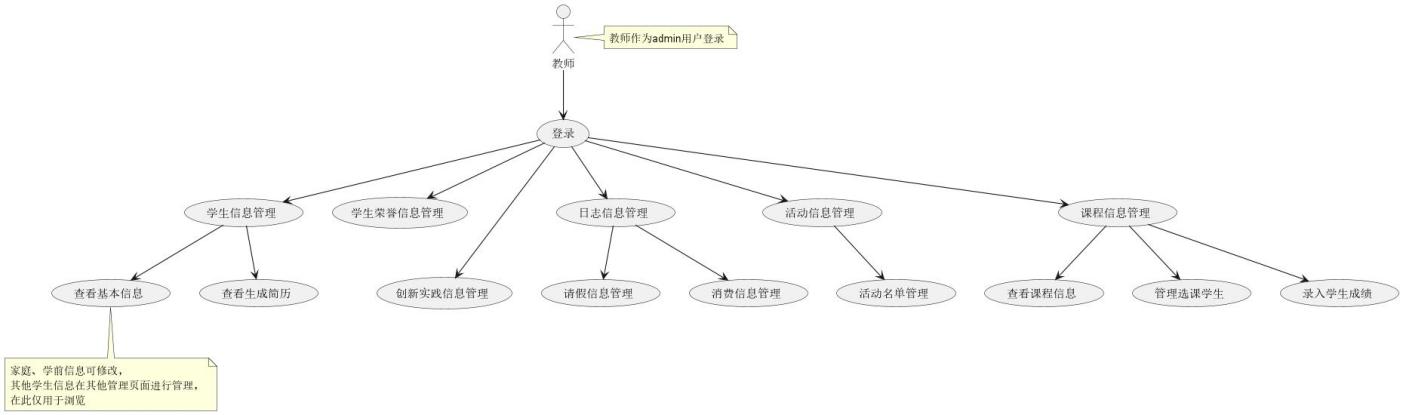
* 产品定位

面向所有山大师生，为山大师生的信息管理提供更便利的服务。

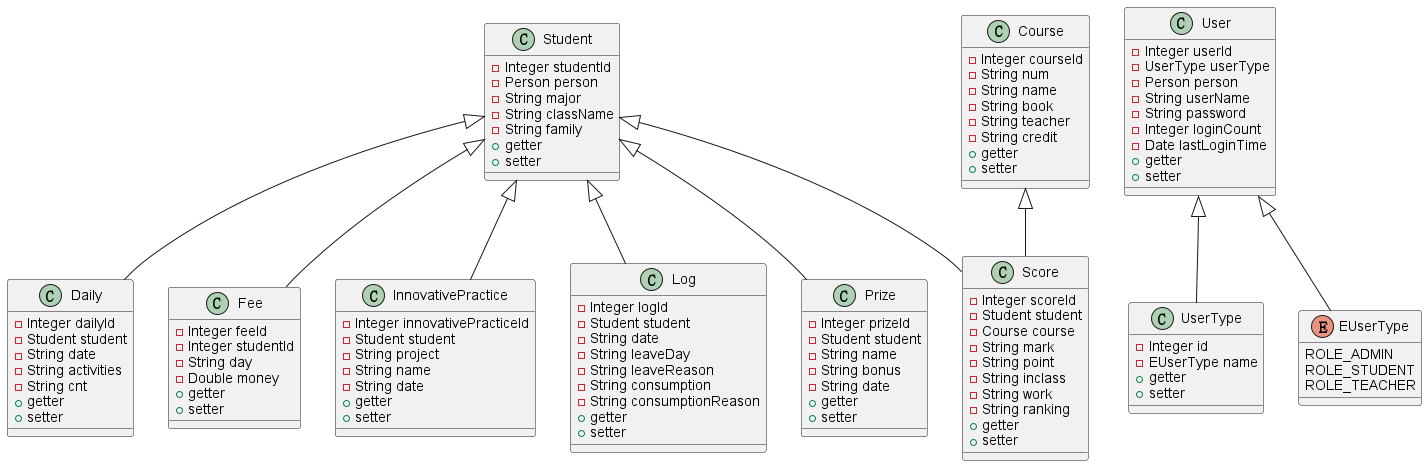
* 产品综述

|  |  |
| --- | --- |
| 产品名称 | 学生管理系统 |
| 产品定位 | 学生个人信息管理及动态更新 |
| 产品特色 | 1. 向学生用户提供查看个人信息的平台 2. 涉及信息全面且准确 3. 为系统管理人员提供更便利的信息编辑以及动态更新平台 4. 构建学生与老师的联系平台，提高学生信息管理的效率 5. 提供学生个人简历的生成以及下载，服务于学生 |
| 目标人群 | 山大全体师生 |

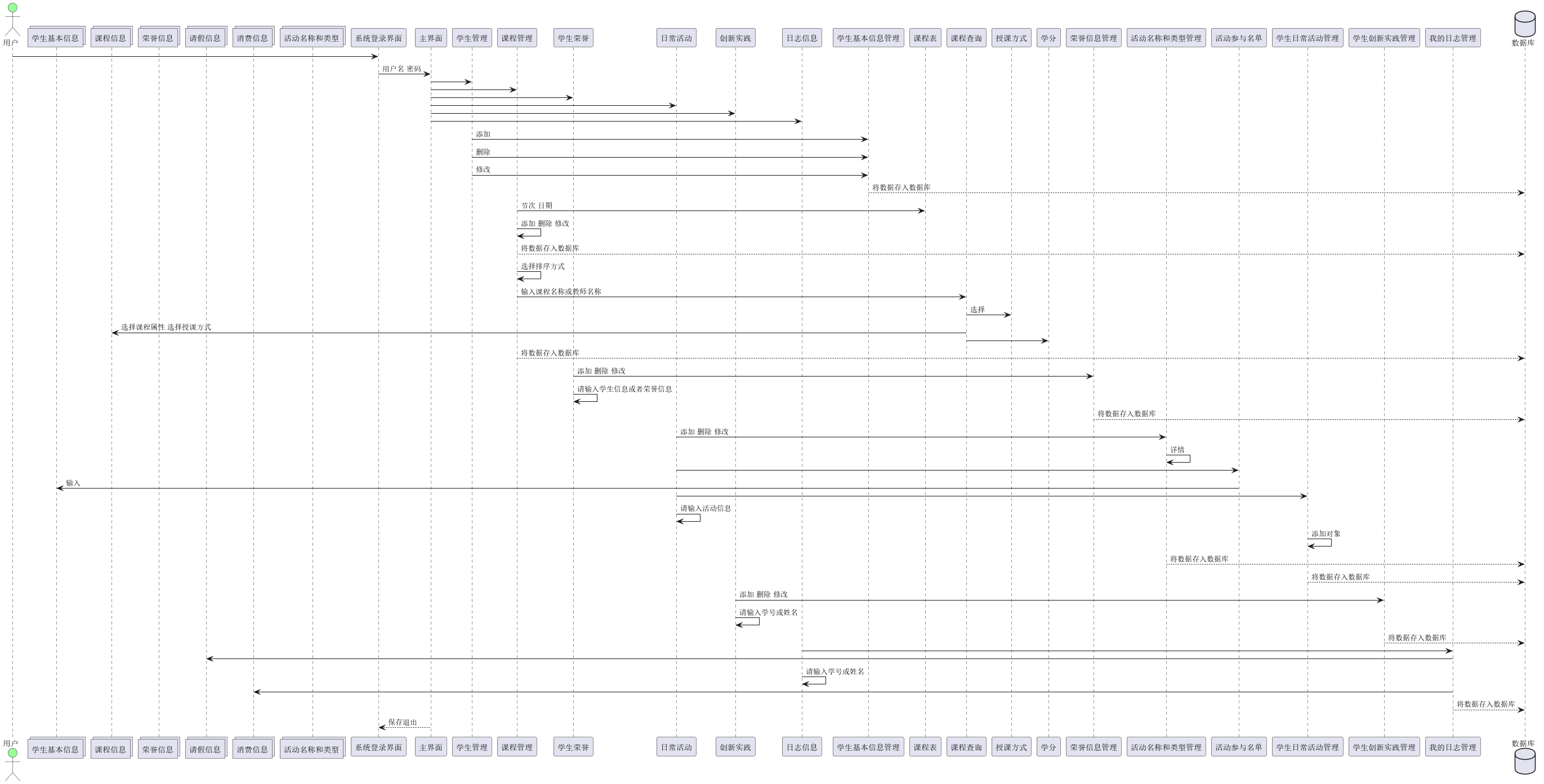
**用例图**



**类图**



**时序图**



1. **系统概要设计**
2. **系统概要设计**

系统的总体架构：

基于Spring boot框架，通过model(实体)、repository（存储）、service（服务）。

model映射数据库表， repository负责数据的查询修改操作，service负责业务逻辑处理，由于业务逻辑处理比较简单，本课程的框架没有单独设计Service，它的作用由controller代码实现，只有在个人简历生成的时候使用了IntroduceService 。

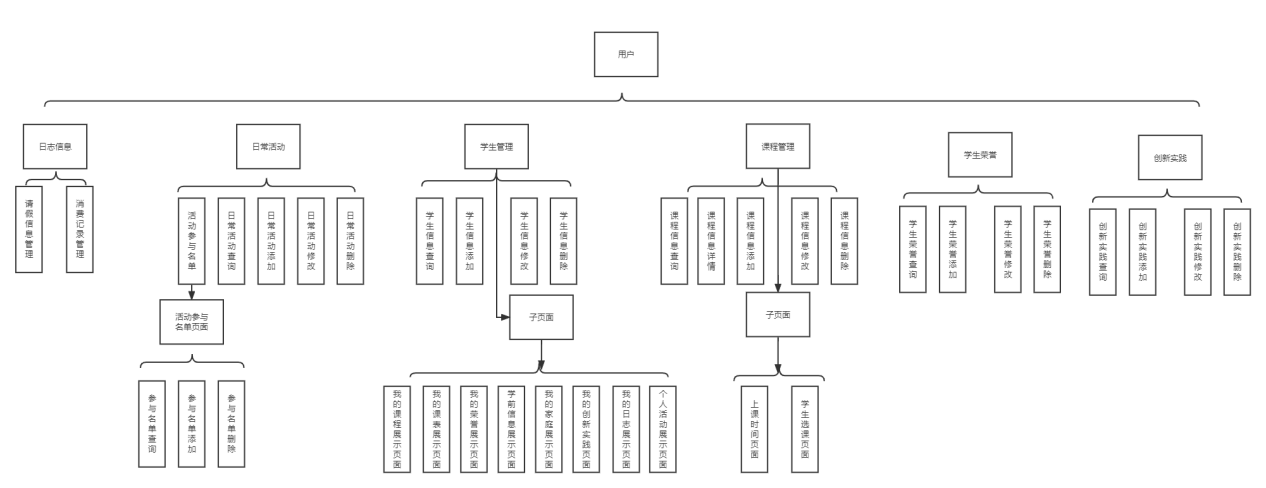
View通常由前端技术实现，我们课程提供的是基于JavaFx（用来搭建用户界面的渐进式框架）的实现， 前端框架根据根据YAML描述的页面结构和命名匹配规则，将后端传过来的数据按照YAML描述进行渲染，显示需要展现的数据，这部分开发框架自动完成。

Controller层是Model与View之间沟通的桥梁，负责分派用户请求并调用service， repository进行业务处理，并把结果返回View（前端框架）

工作流程

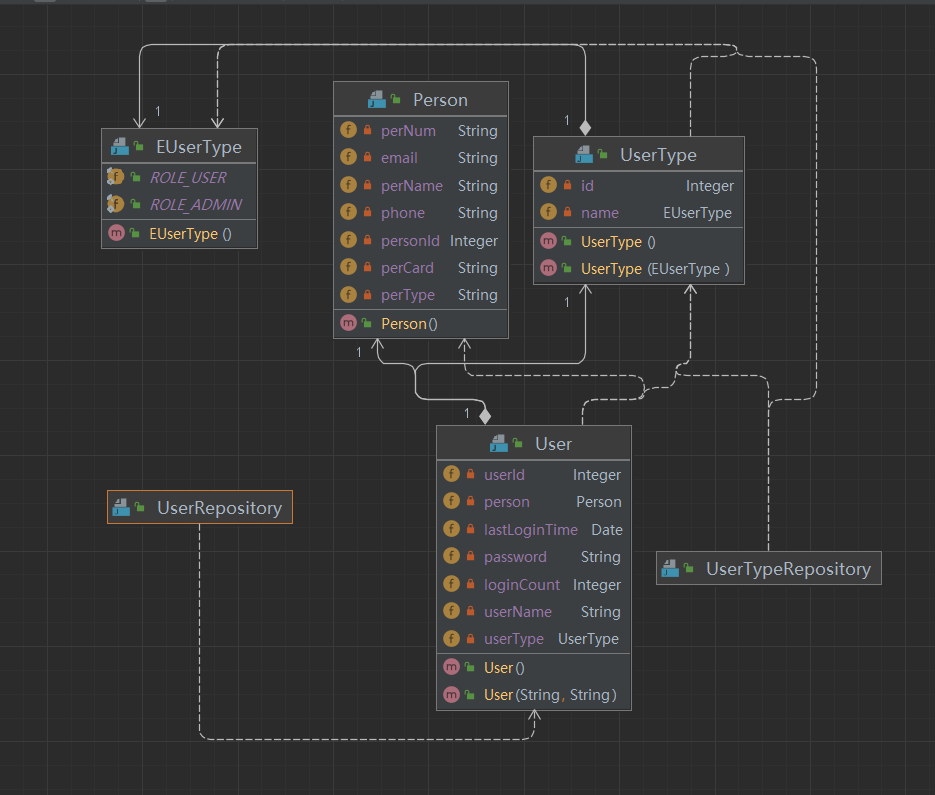
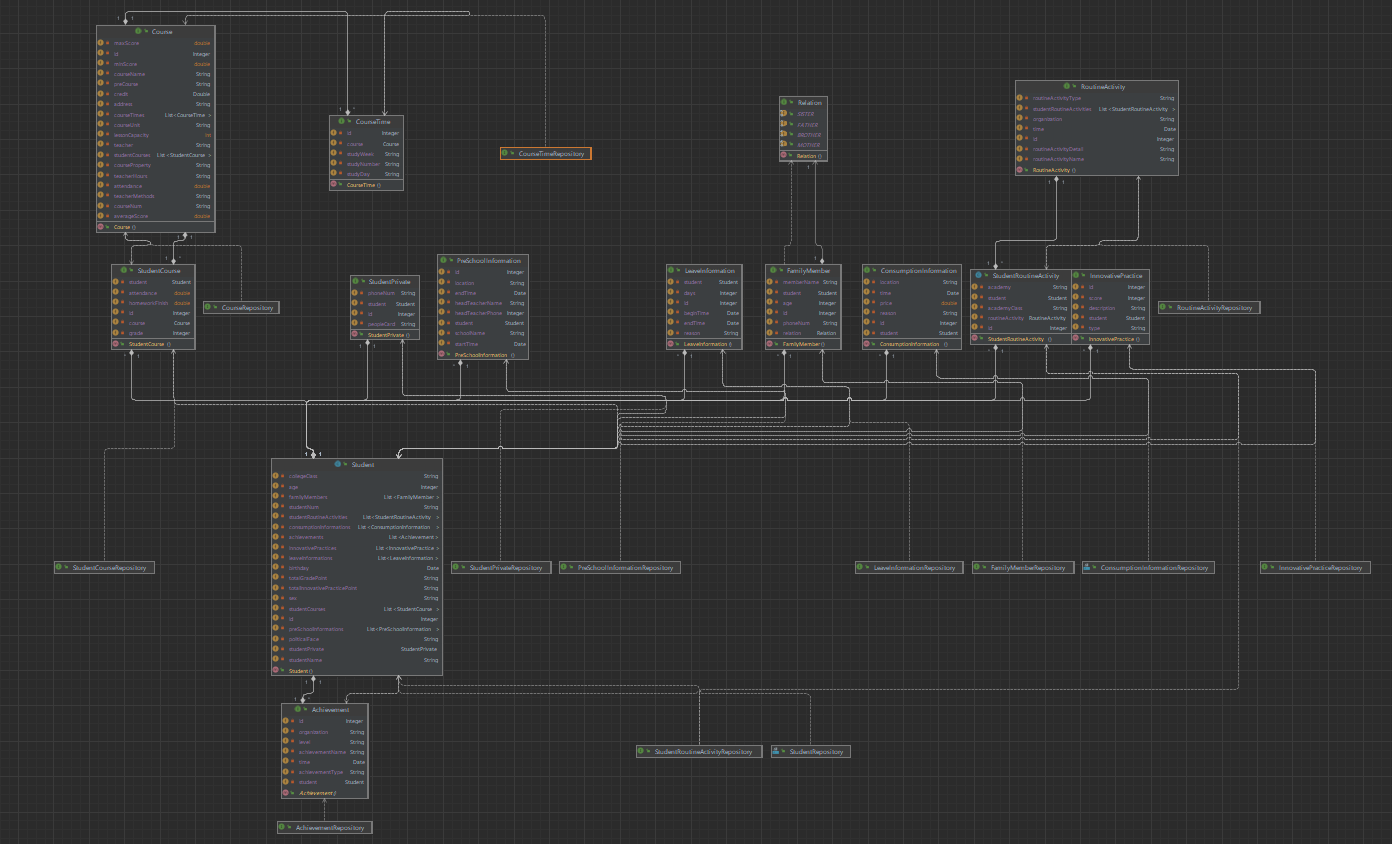


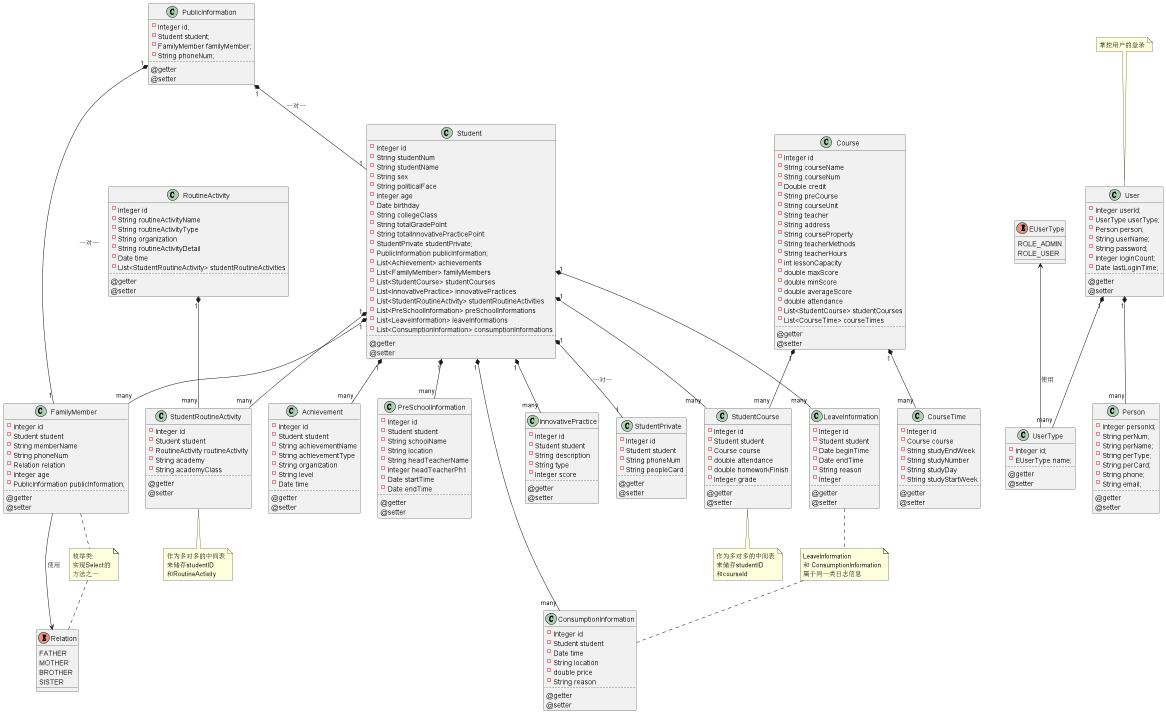
功能模块划分



1. **系统的详细设计**

类图：

、



CourseController 课程控制方法

StudentController 学生的各类信息控制方法

DailyController 日常活动控制方法

ScoreController 学生分数课程控制方法

LogController 日志控制方法

PrizeController 学生各种奖项控制方法

InnovativePracticeController 创新实践信息控制方法

HtmlController Html语言编辑控制方法

**Student类 ：**

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer studentId;

@OneToOne

@JoinColumn(name="person\_id")

private Person person;//一对一的personId

@Size(max = 20)

private String major; //专业

@Size(max = 50)

private String className; //班级

@Size(max = 50)

private String family; //家庭信息

**Course类：**

@Id

private Integer courseId;

@Size(max = 50)

private String name;//课程名

private String num;//课序号

private Double credit;//学分

@Size(max = 50)

private String courseUnit;//开设学院

@Size(max = 50)

private String teacher;//授课教师

**Prize类:**

@Id

private Integer prizeId;

private Student student;//多对一的外键（student\_id)

private String name;//荣誉名称

private String bonus;//奖金

private String date;//获得时间

**Fee类：**

@Id

private Integer feeid;

private Student student;//多对一的外键（student\_id)

private String day; //消费时间

private double money; //金额

**InnovativePractice类：**

private Integer id;

private Student student;///多对一的外键（student\_id)

private String project;//项目类型

private String name;//名字

private String date;//时间

**Log类：**

private Integer id;

private Student student;//多对一的外键（student\_id)

private String date;//时间

private String leaveDay;//请假时间

private String leaveReason;//请假原因

private String consumption;//消费

private String consumptionReason;//消费原因

**Score类：**

private Integer scoreid;

private Student student;// 多对一的外键（student\_id)

private String course;// 多对一的外键（course\_id)

private String mark;//成绩

private String ranking;//排名

private String point;//绩点

private String inclass;//出勤率

private String work;//作业完成度

**Daily类:**

private Integer dailyId;

private Student student;//关联学生

private String activities;//活动类型

private String cnt;//活动人数

private String date;//时间

###### 使用说明

**具体业务功能的执行给出一个使用说明；**

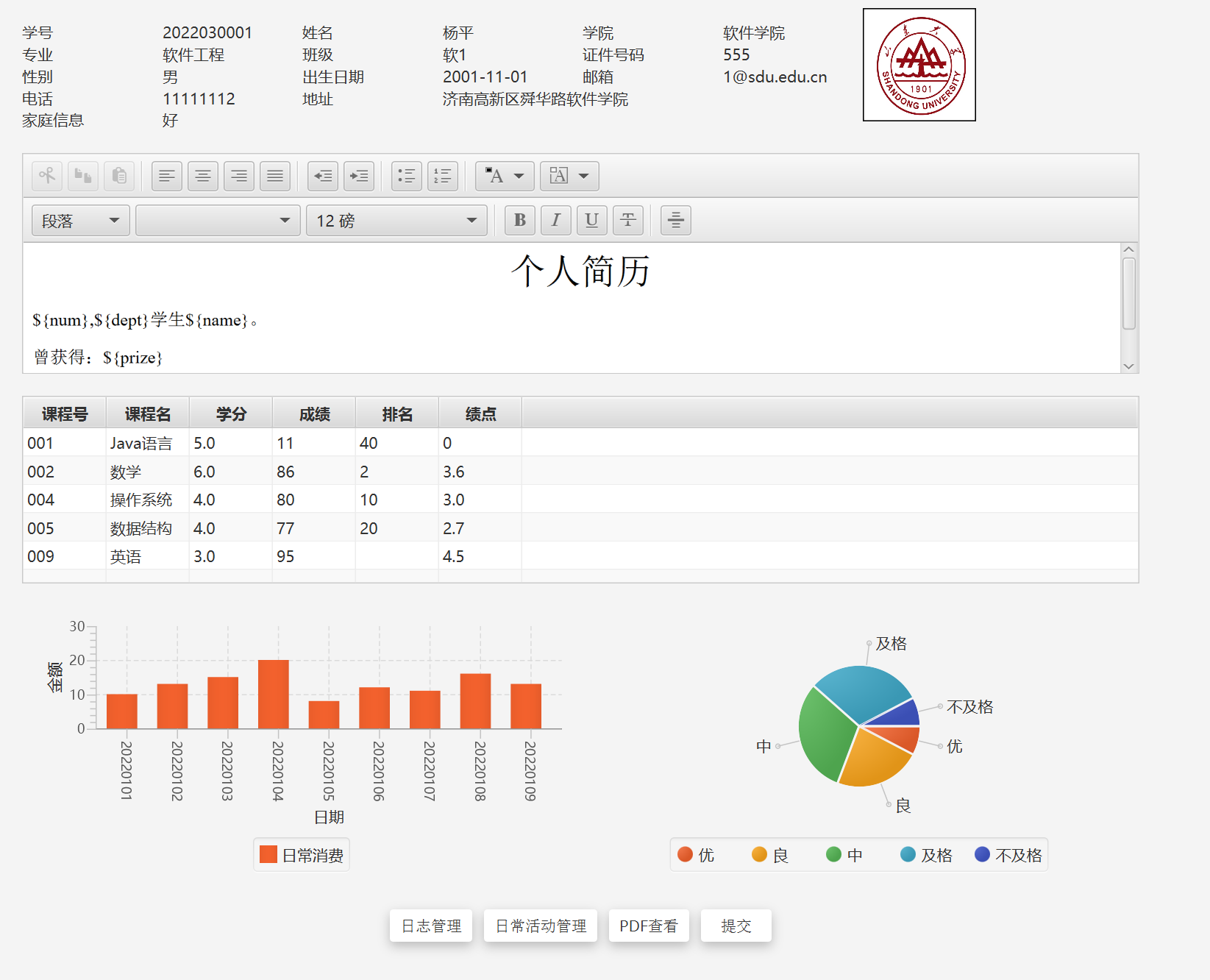
页面已按照实验提供条件完成所有功能

此外，通过研究、利用提供代码与修改前端代码，实现了以下功能

* 增删改查之后自动查询一次，不必重新打开
* 自动填充编号列，让界面清晰易懂
* 简历中生成学生课程能力图、创新实践饼图
* 学生课程带有课程编号
* 多种查询、添加方式，添加页面根据数据库提供选项
* 对逻辑进行控制，对不符合逻辑与要求特别是部分重复的输入信息（如手机号、身份证号、学号等）都会采取不录入数据的方式处理并进行提示
  + **学生管理页面**

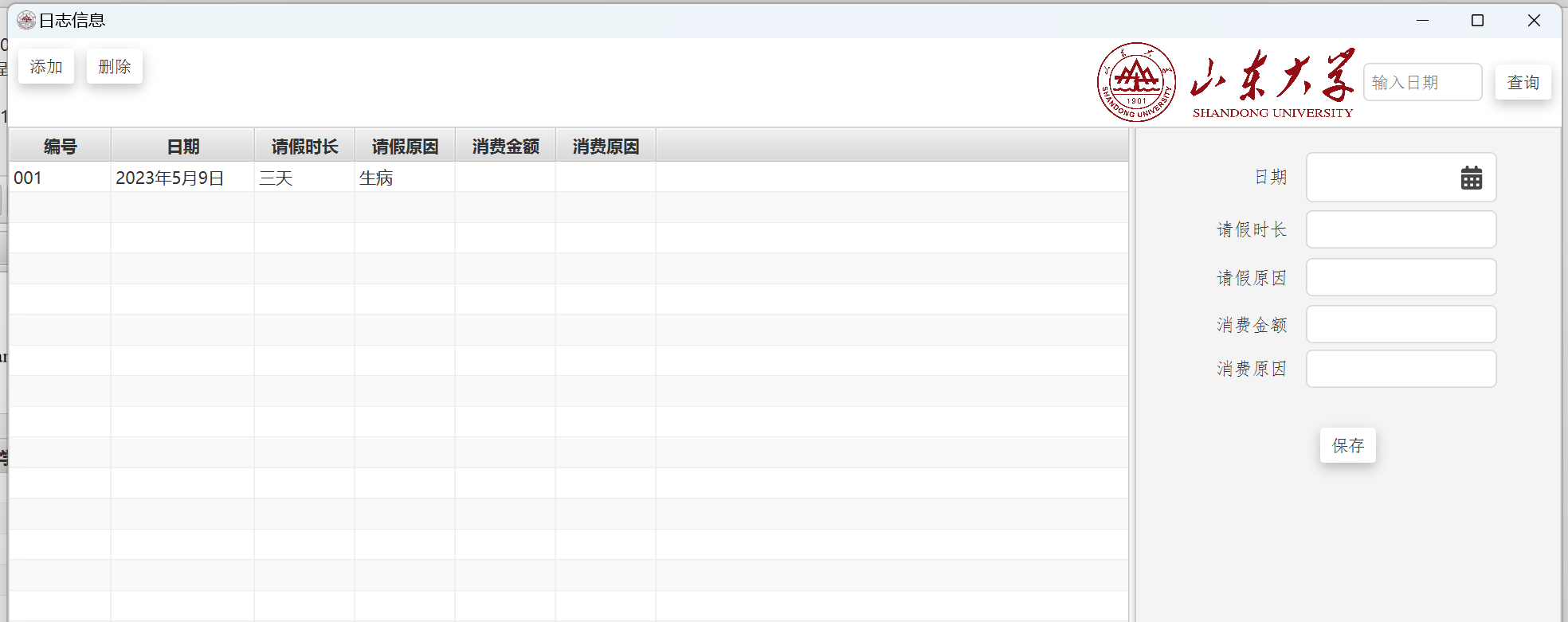
页面概述：学生管理页面用于统一了解、管理学生,并修改部分学生信息，以下以杨平同学为例进行介绍

页面：



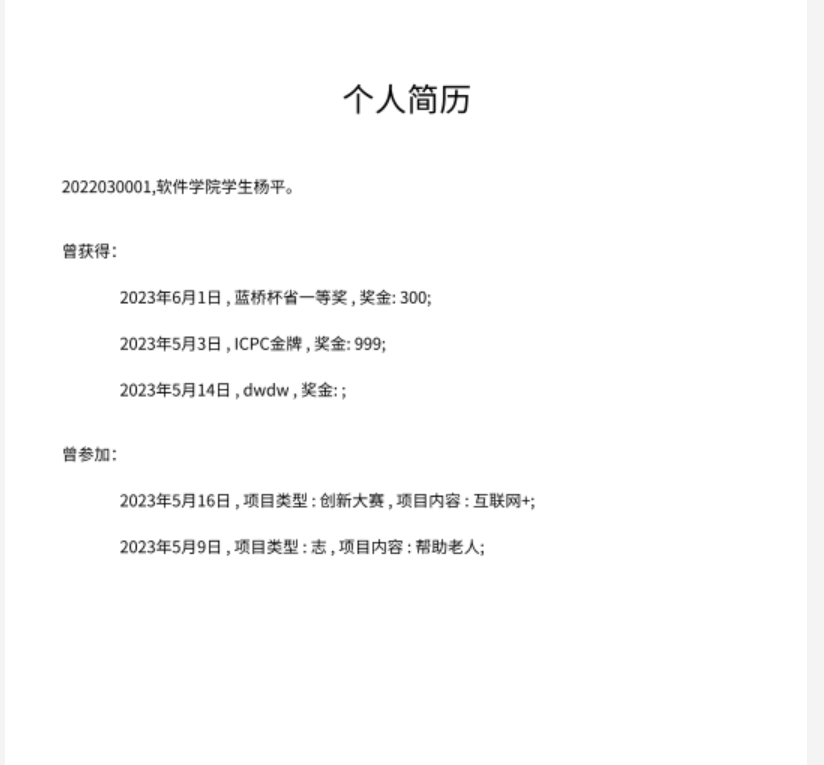
说明：上面显示个人信息和头像（支持修改），中间部分是个人简历的编辑功能，可以自主添加想要的部分（不推荐），下面是消费直方图和成绩饼图。最下方四个按钮，前两个支持学生自己的日志管理和日常活动管理（可修改），第三个查看简历，第四个提交简历中的修改。

添加编辑页面



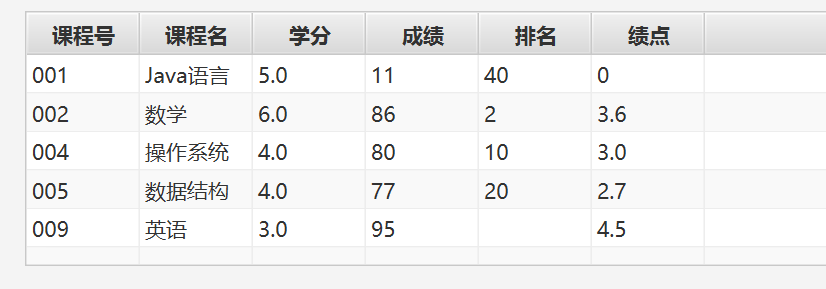
说明：和教师端保持一致，可进行修改，添加，删除等学生对自己日常活动管理的一些功能。

* + **简历页面**

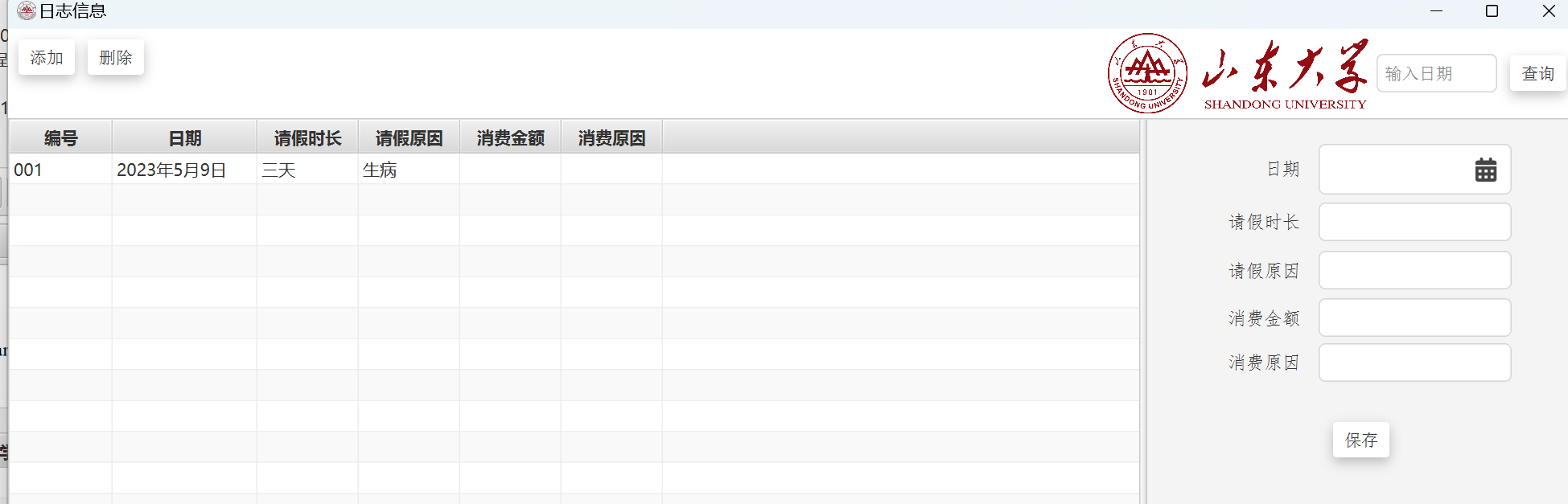


说明：此处上面显示学好，学院，等信息。然后以后端数据库查询的方式展示学生曾获得的奖项及奖金，学生曾参与的创新实践信息和项目内容。这里也表明了这两项内容只支持查看，而不支持修改，这也符合它作为一个学生管理系统的逻辑。同时信息的输出符合格式，由Html语言进行控制。

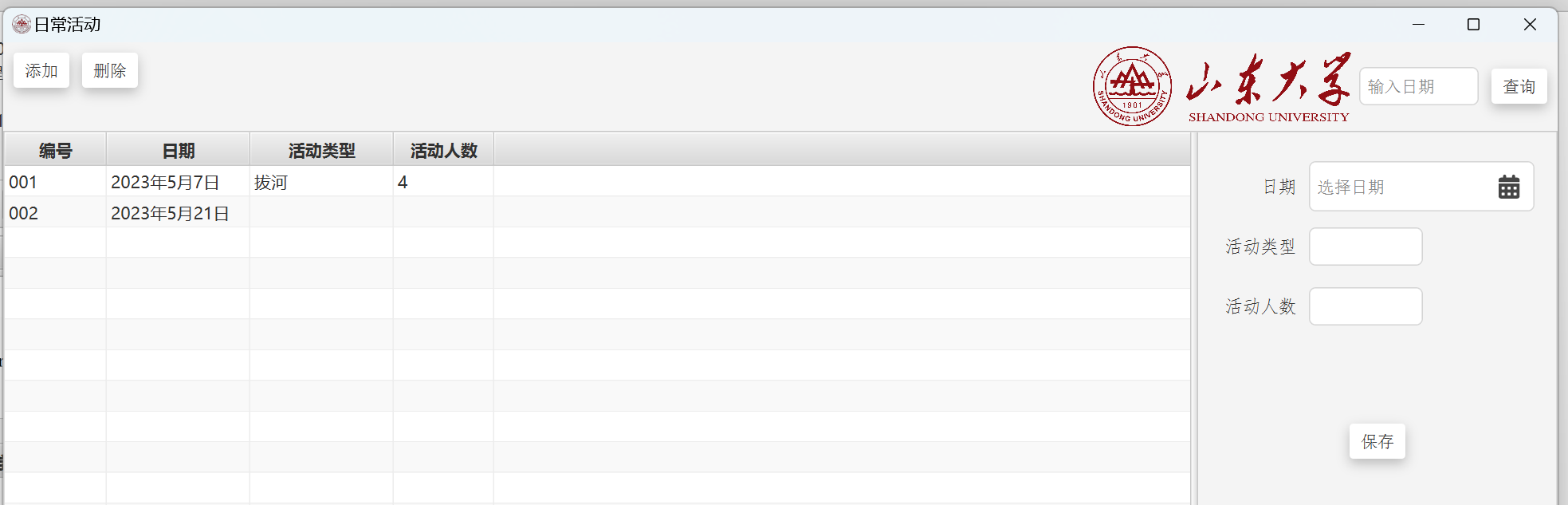
* + 我的选课页面



* + **日志信息页面**



* + **日常活动页面**



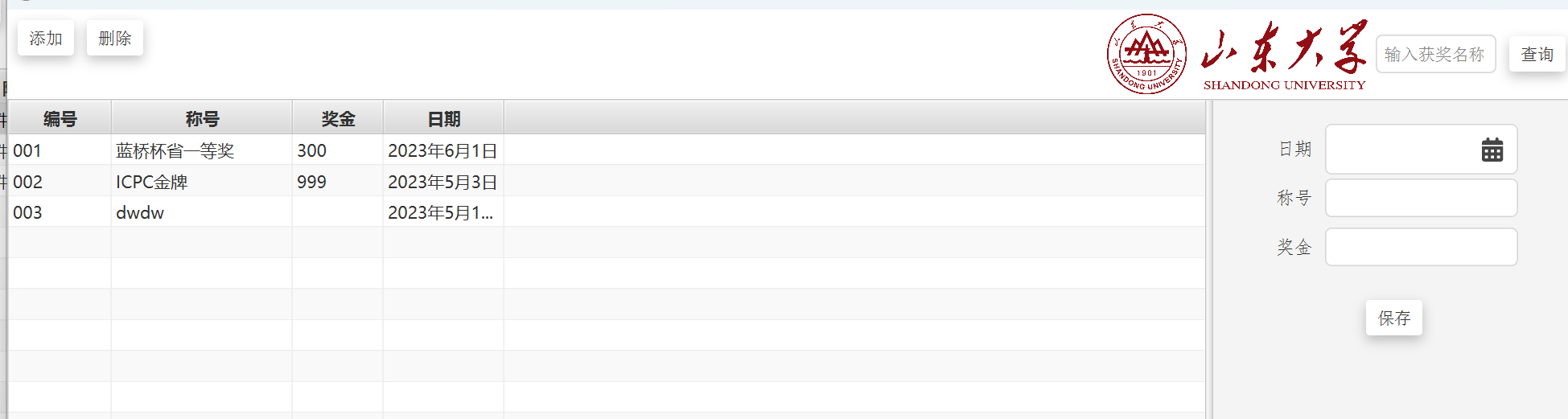
* + **课程管理页面**

页面概述：课程管理页面用于增添、修改、删除课程，管理添加、删除选课学生，以及录入学生成绩

页面

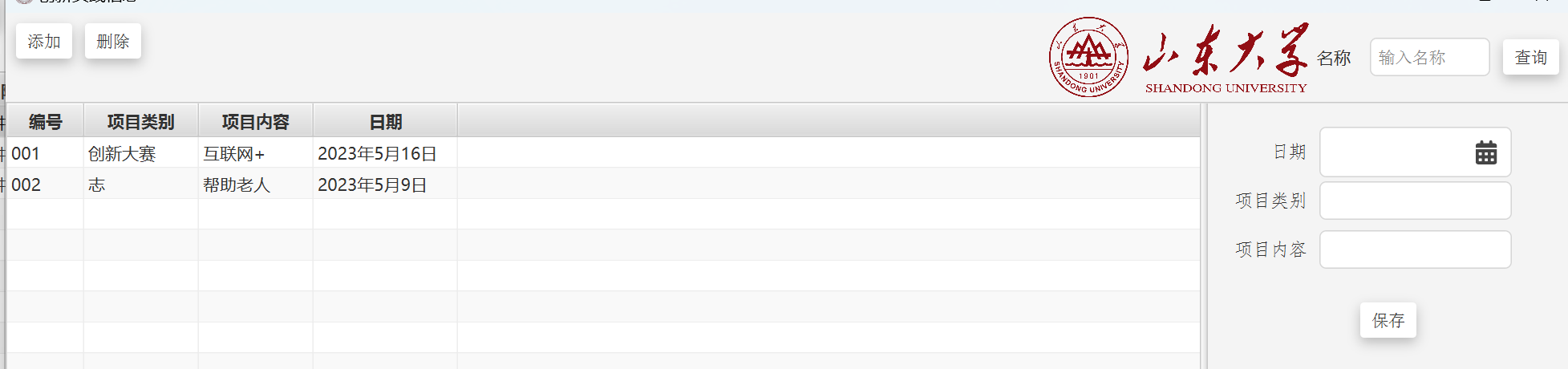


* + **学生荣誉页面**



提供细化的查询方式，方便查找

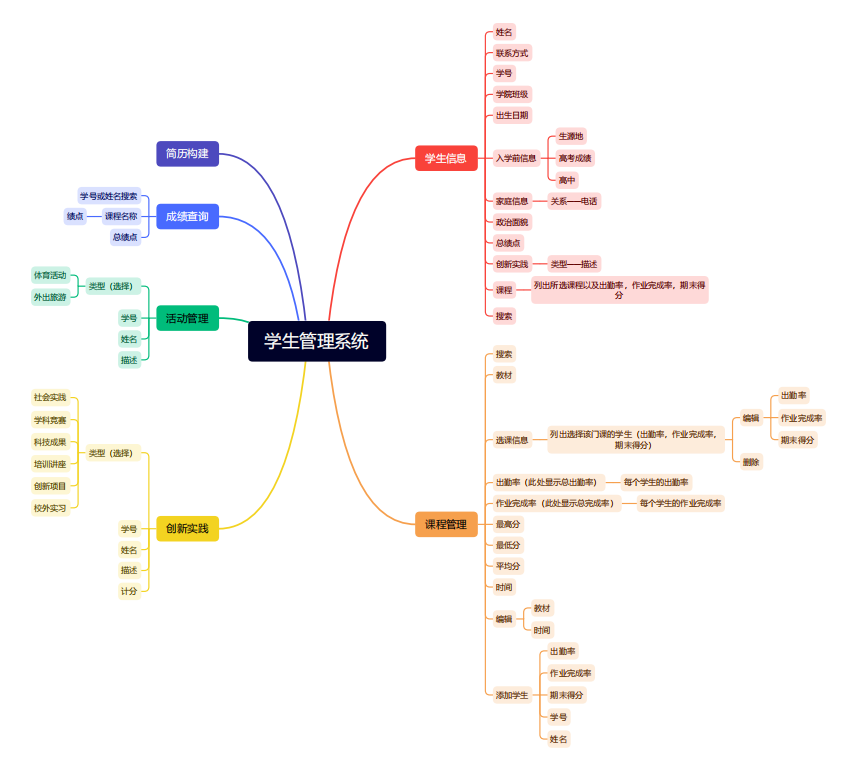
* + **创新实践页面**



可通过项目类型查询

1. **系统实现**

首先根据题目即考核要求，我们先对其功能进行了分析，制作了相应的简单的pdf来划分相应的菜单模块，然后分工进行页面制作。



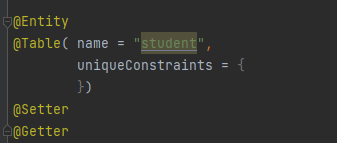
1. **实体类基本逻辑**：

成员包括基本属性和部分一对多和一对一的属性。

在Java Spring JPA开发框架对Java对象操作映射为数据库中记录以及老师对前后端的封装下，我们只需要通过相应的注释和小驼峰命名法（例如：studentNum->student\_num）就可以实现model类属性和数据库中列名的对应。

1：@Entity--->实体类

1. @Table (name = “数据库名” ) --> 通过该注解直接生成相应的数据库表
2. @Date（@Setter 和 @Getter） 通过插件lombok 来实现get set方法通过注解自动生成

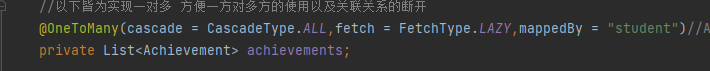


基本属性：String，Integer，double，Date等，其中id是主键 不可省略（但也无需通过前端输入，但对于最重要两个类Student和Course来说，学号，姓名以及课程名不可省略（@NotBlank）

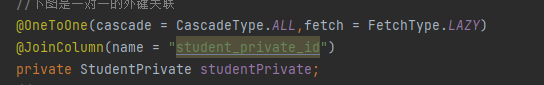


其他成员

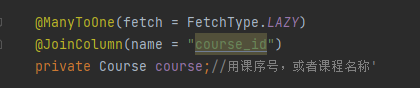
一对多的成员（@OneToMany)



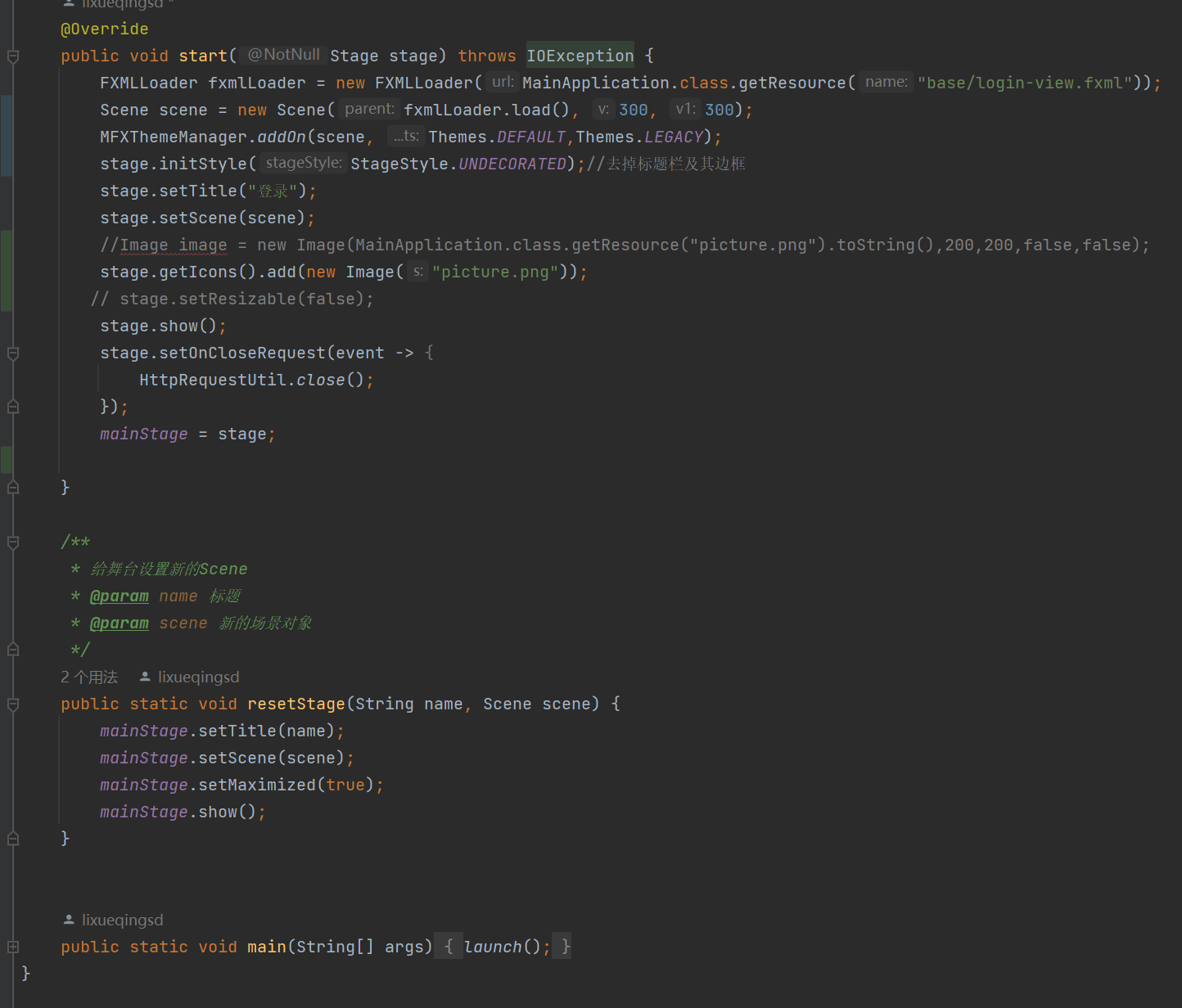
一对一的成员（@OneToOne）



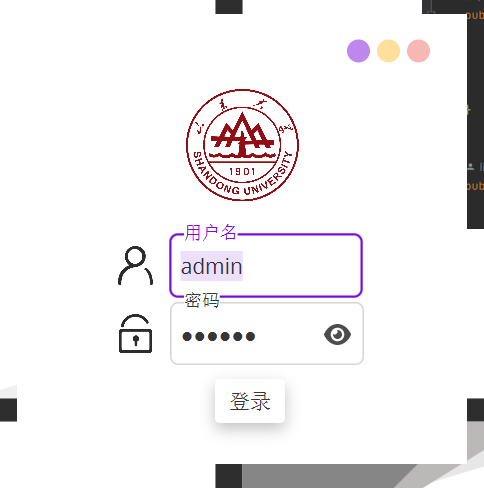
多对一的成员（@ManyToMany）



1. **控制逻辑：**



前端打开login表，展示登录界面



经由前端组件优化，登录界面可以拖动且比较优美。

初始密码为123456.

用户名有两张：admin和学生

学生的话用户名为学号

Admin为管理员端

**Admin端：**



平平无奇四个菜单

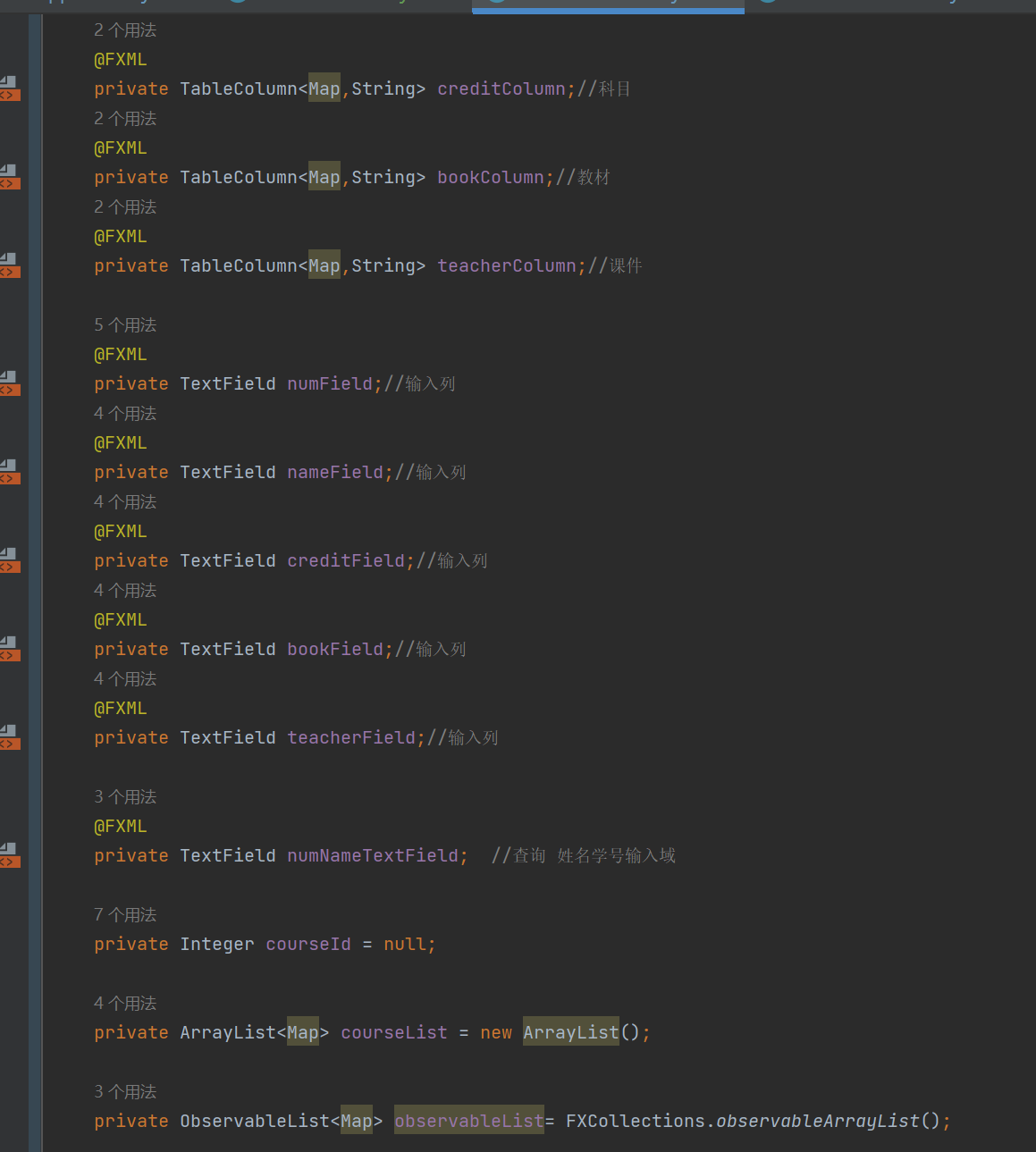
个人信息里为修改密码等操作

编辑里面为修改一些系统属性等操作，与我们的学生管理无关

**课程管理：**

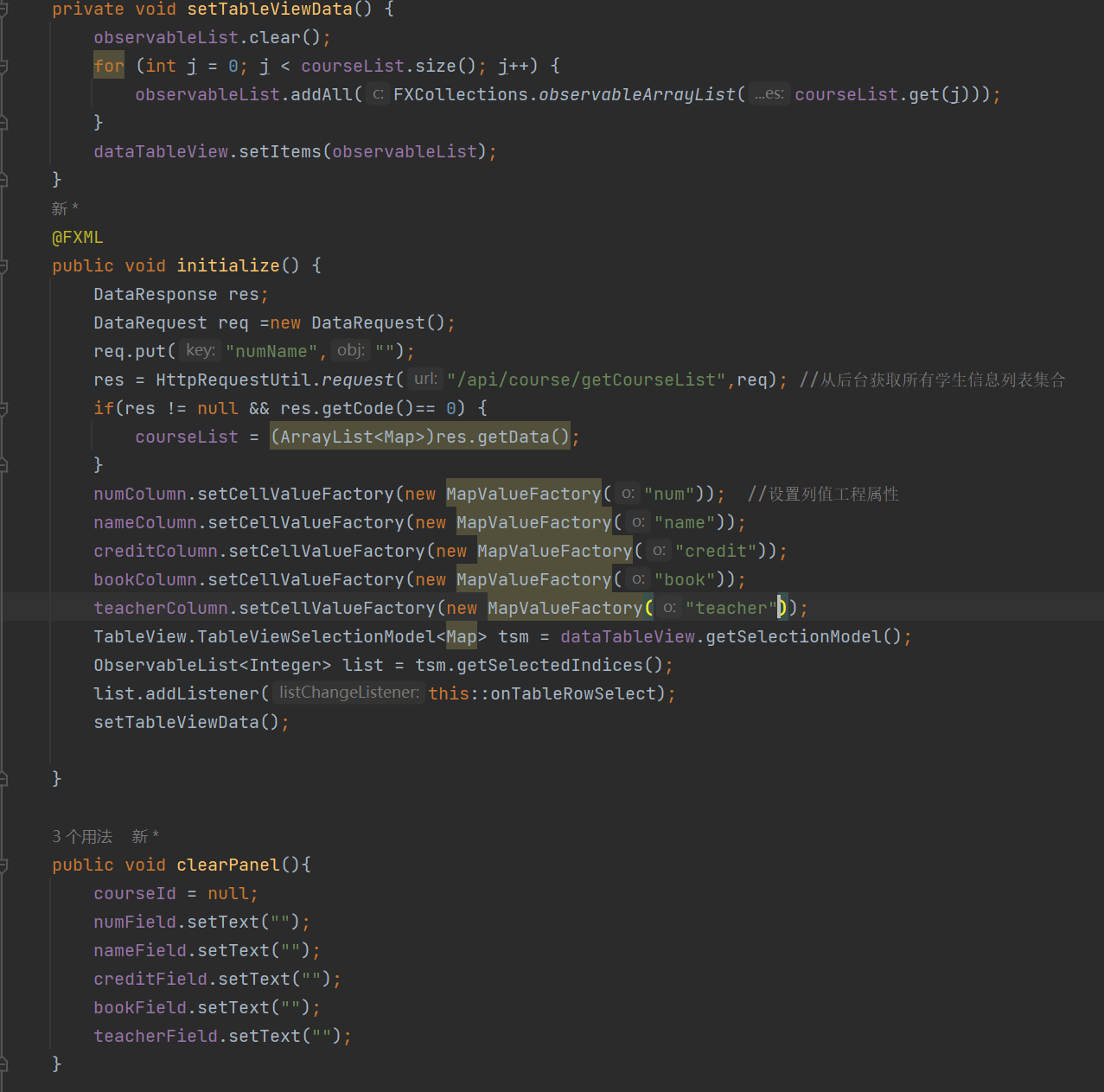


由CourseController设置

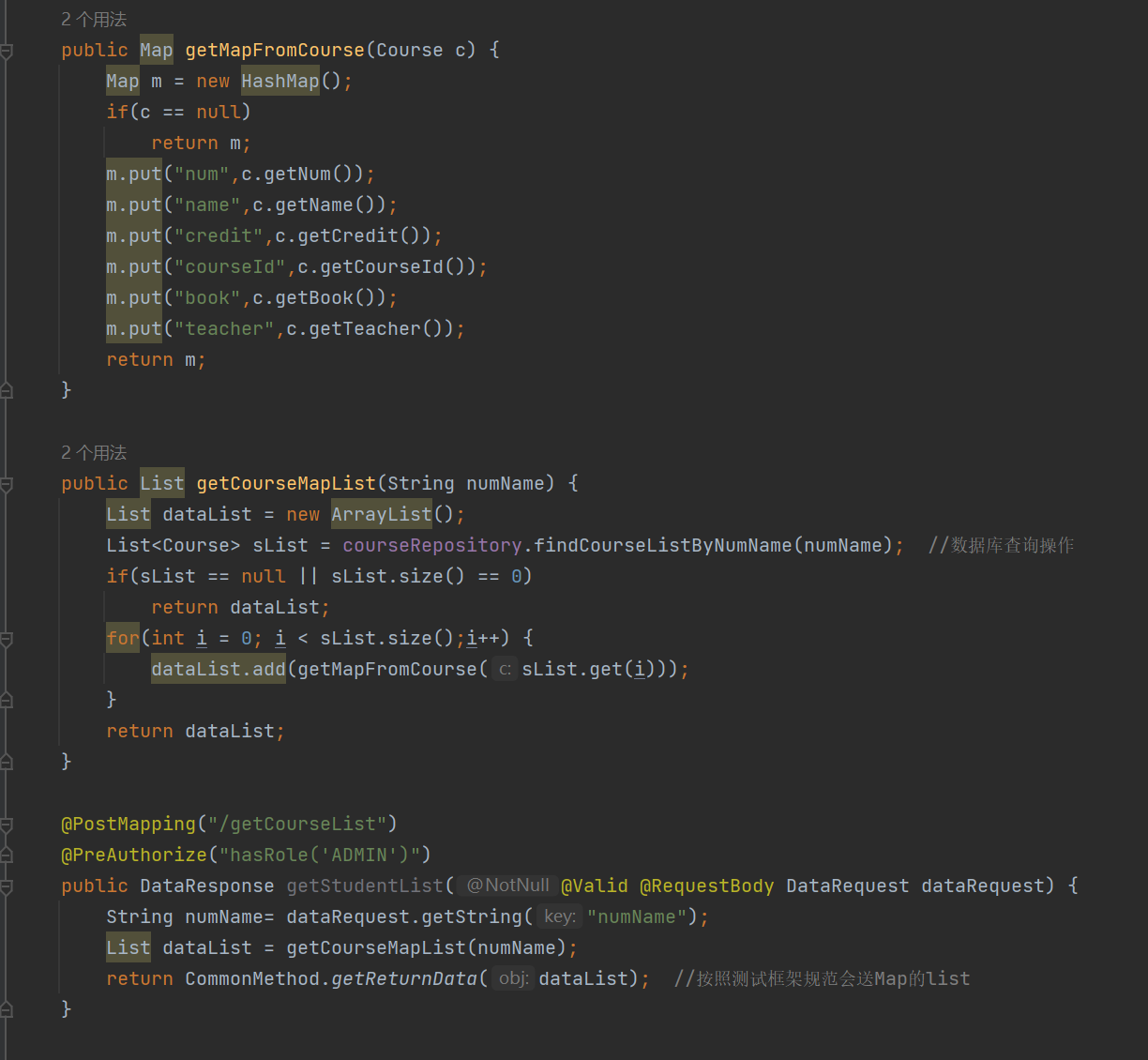


在CourseController中设置了表，列等控件

界面初始化时会调用相关方法：

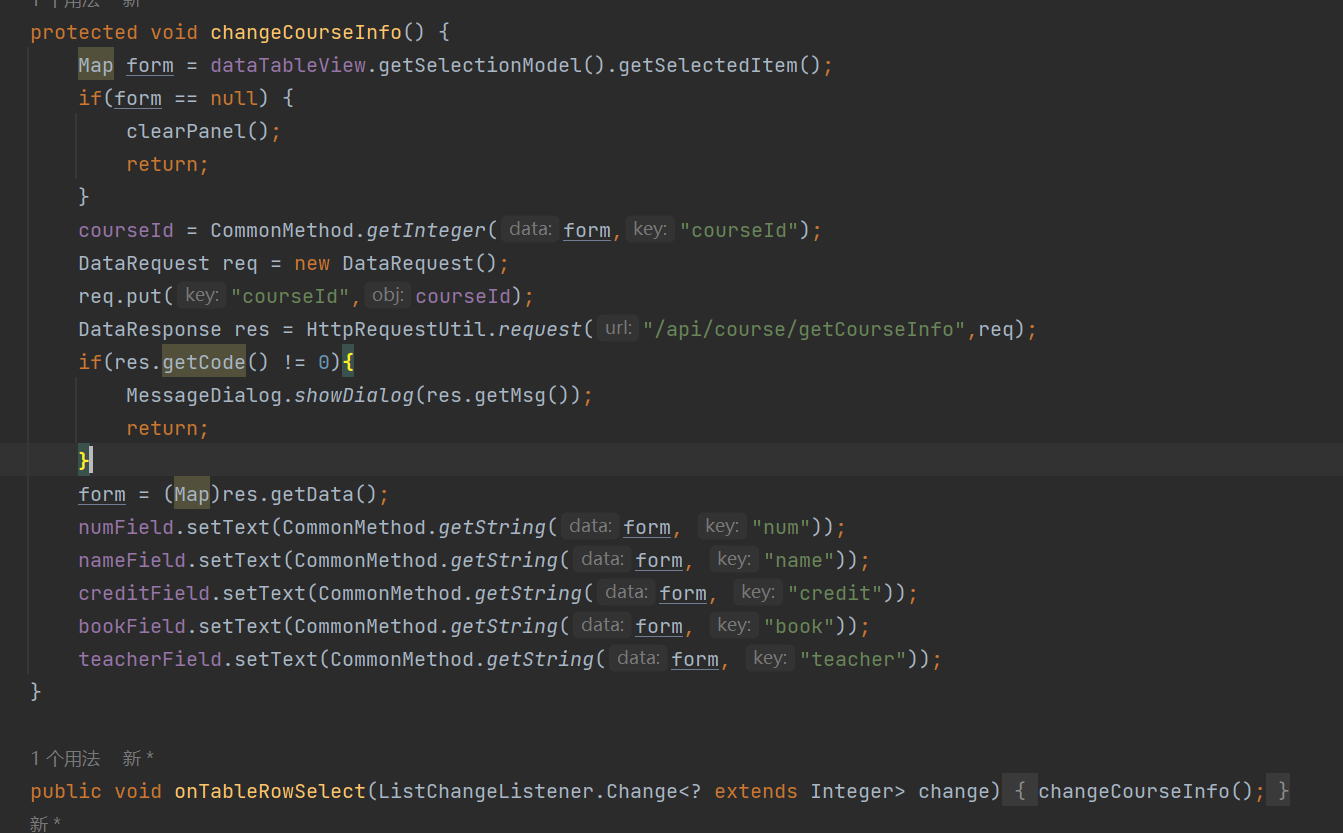


通讯传到后端：

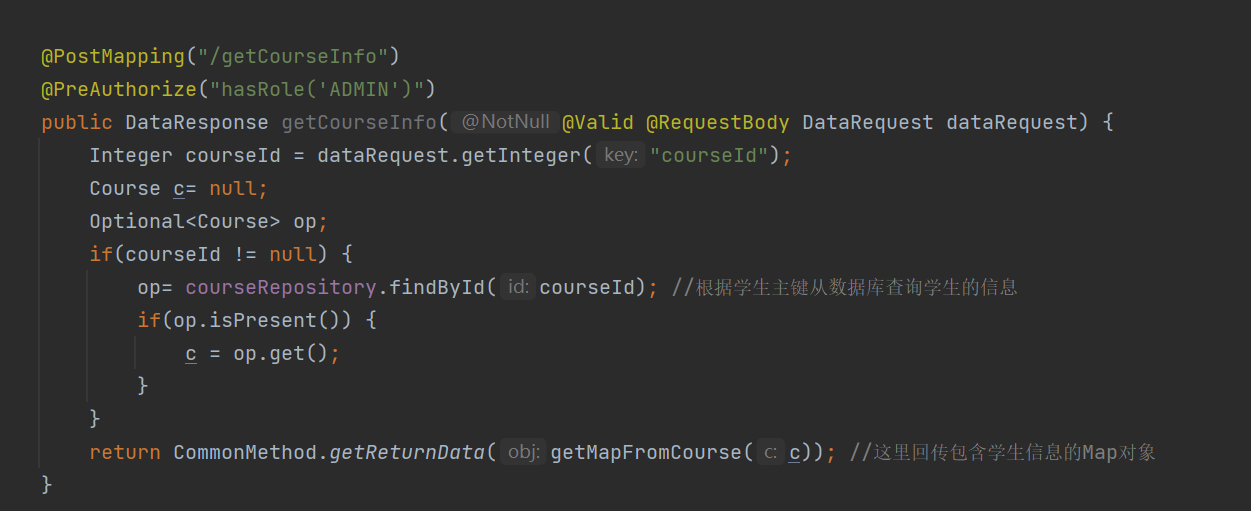


通过API通讯，前端可以把一个空的字符串作为查询启动器传到后端，后端会根据数据库查询方法把数据库里的所有课程信息查询出来，同时通过map，list等结构储存起来返回前端。前端收到信息后可通过setTableViewDate（）方法，自动填充表的各列及相关信息。

选择：



每当用户选择某一列时，系统会监测到鼠标的选择时间，通过req向后端请求：



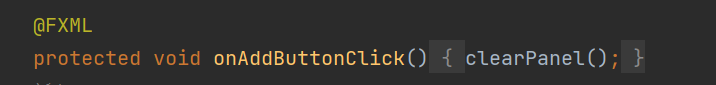
后端接受到参数courseId,会通过findById方法查询，如果查询的到，就把它的map信息返回前端，前端再进行方法吧它显示出来。

查询按钮的响应：



与刚才别无二致，这里略过。

添加按钮的响应：



添加的逻辑是清空显示列，至于真正的添加其实是保存。

删除按钮的响应：



如果没有选择，则弹出消息：没有选择，不能查看。

如果有选择，则弹出确认消息，是否要删除。

如果仍要删除，则把CourseId传进req里，传入后端。



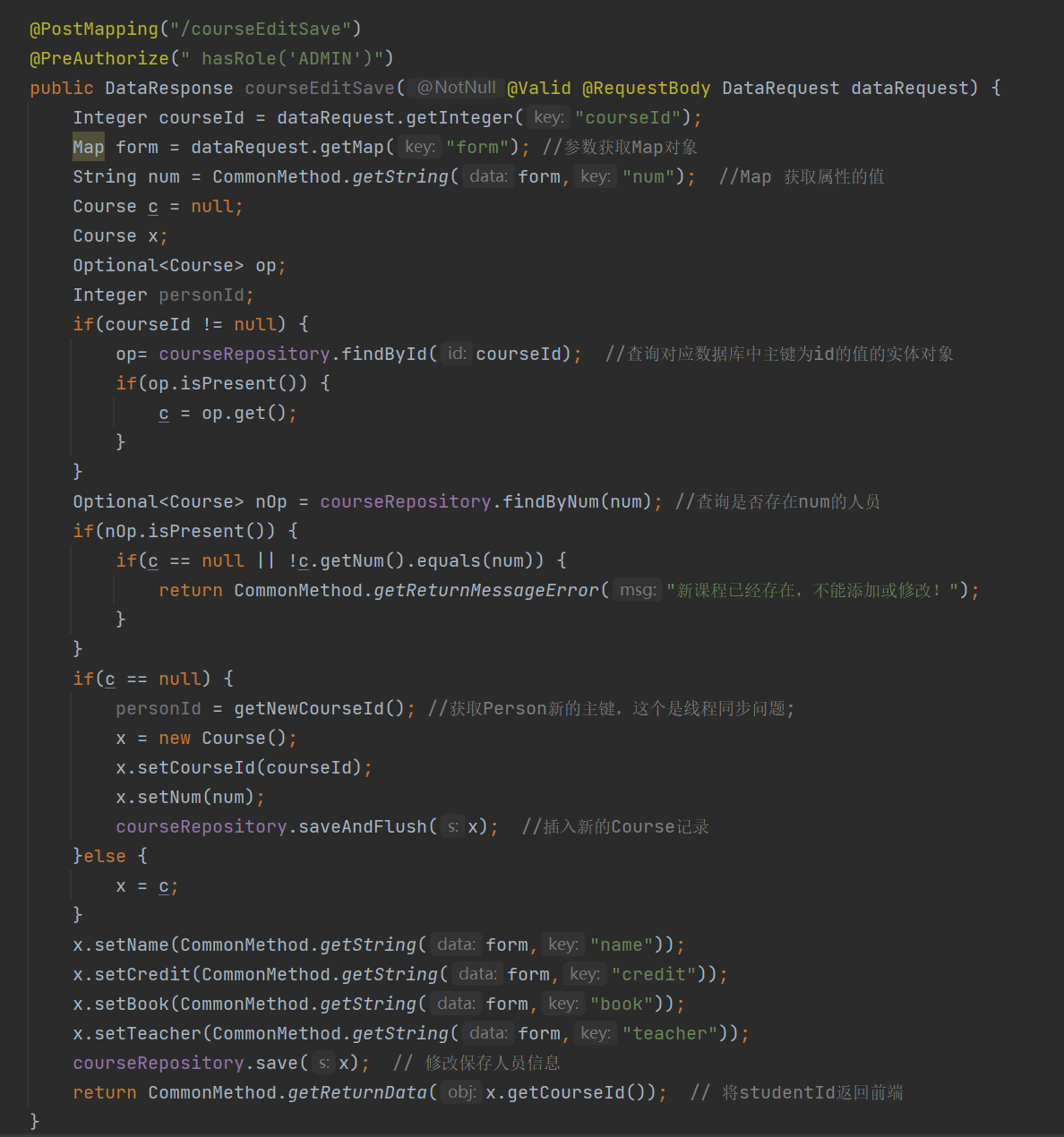
后端会删除数据库信息。

保存按钮的响应：



保存显然支持两个功能，添加和修改

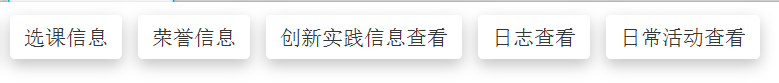
所以在后端把各个field的文本提取出来放到map里，传入后端



后端会进行判断，如果存在此数据，即为要实现修改功能，就单单改一下数据就可以了；如果不存在，即为要实现添加功能，先saveAndFlush，再添加信息保存。

**学生信息管理：**

除学生基本信息以外有如下功能

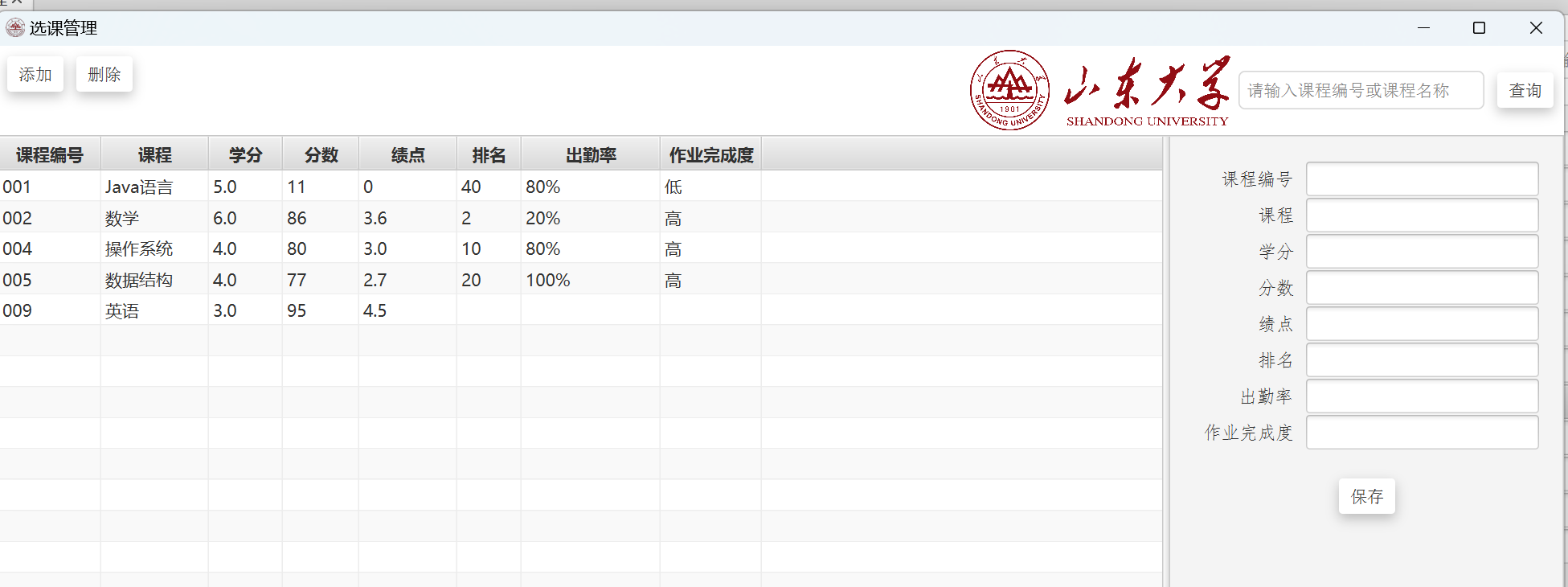


关于基本信息：

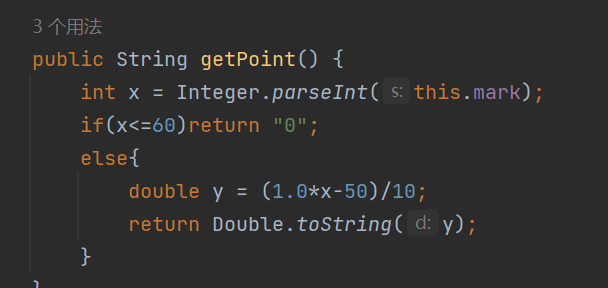


Student表与person，user联动，其增删改查的实现与课程无太大区别

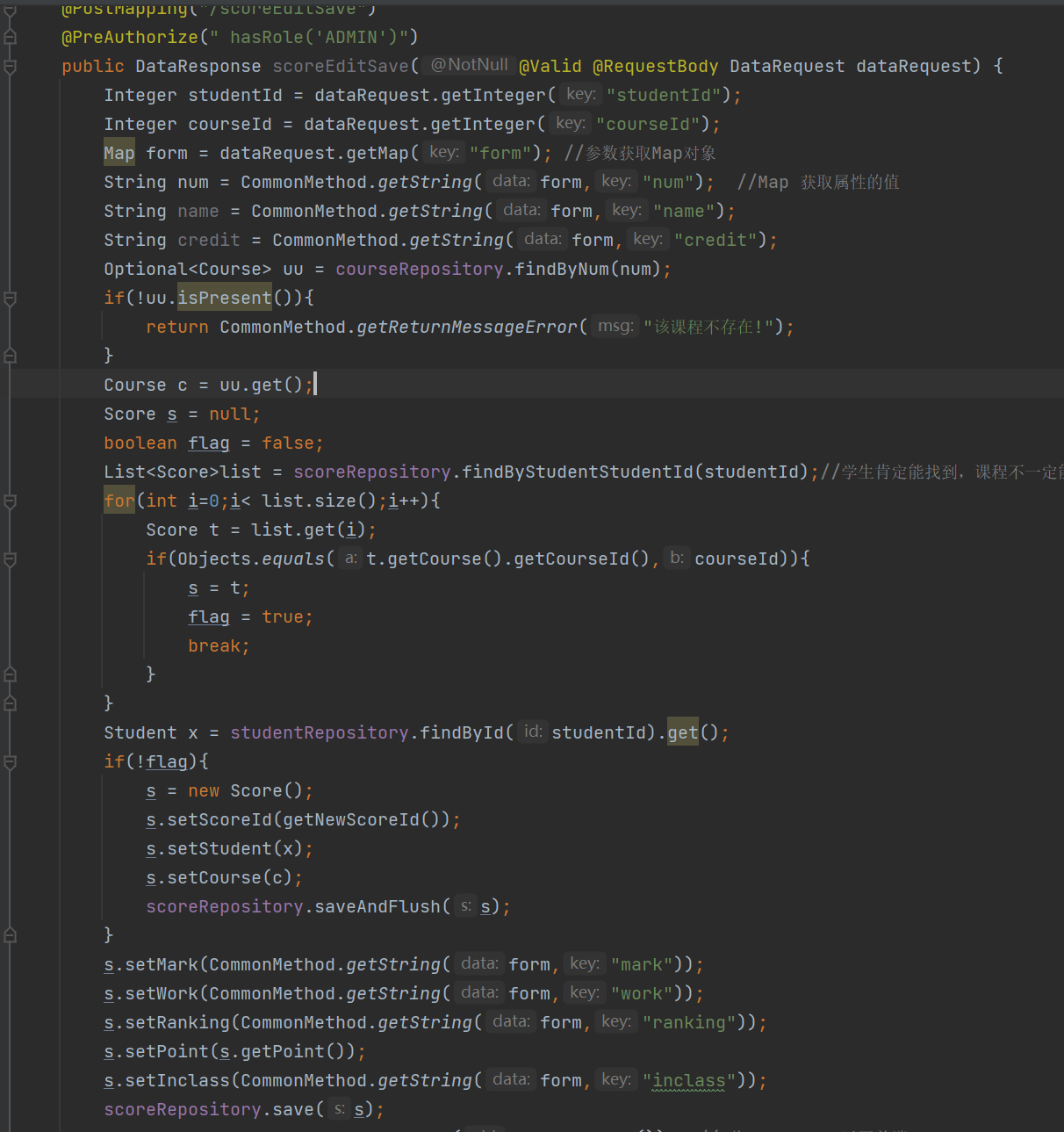
选课信息：



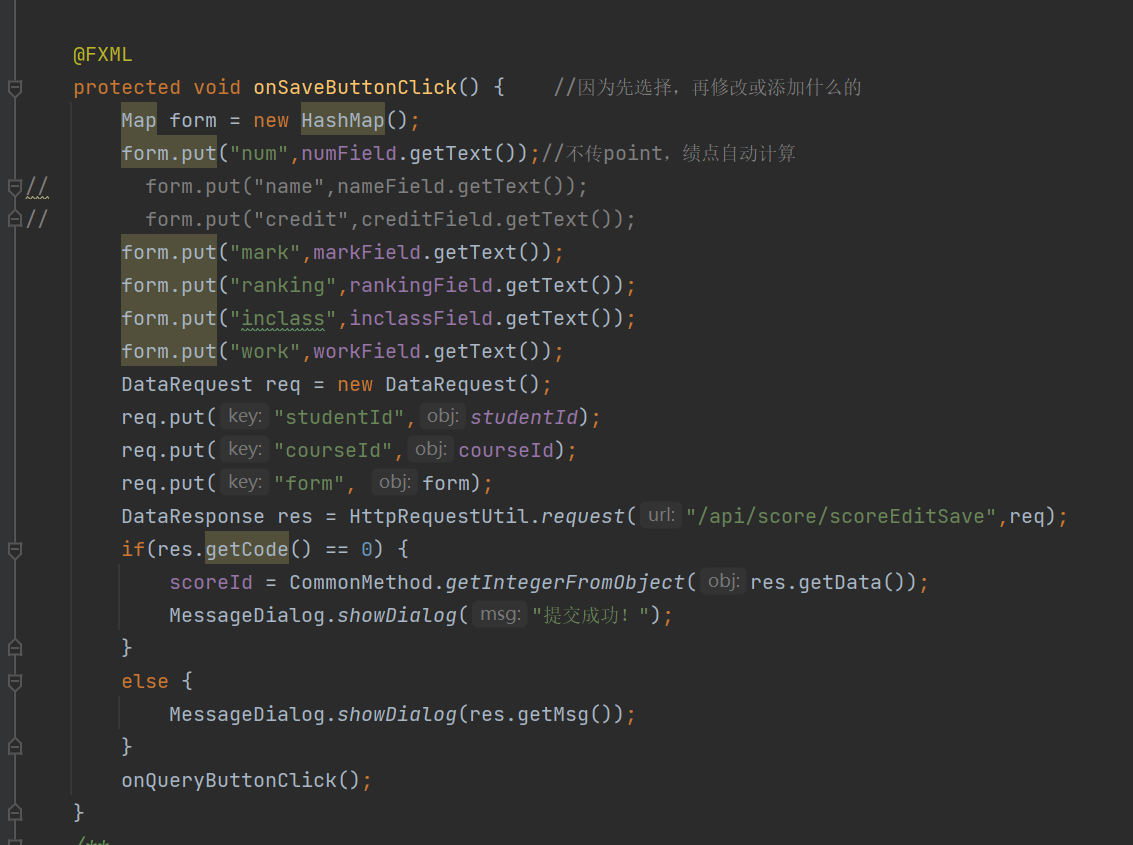
其中，绩点无需手动修改或添加，而是按照一定的规则自动填充，这样的话要求修改时分数在绩点前



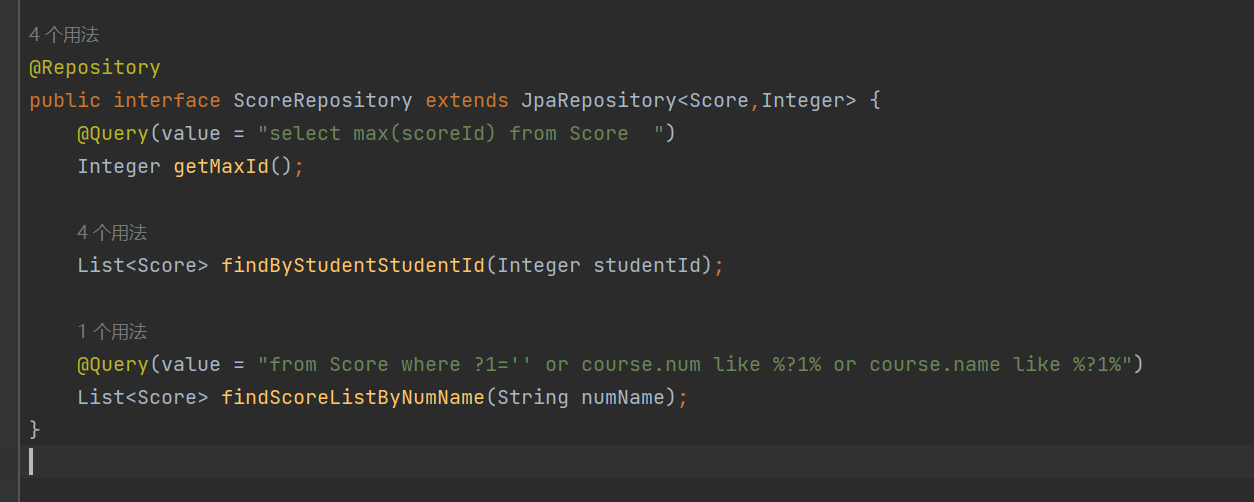
相关方法的使用：



其中setpoint方法的参数仅仅写getpoint就可以，同时对于前端，也做了相应修改



此表与student表，course表强关联，在Repository里也有体现



相应的数据库查询语言进行技术支持。

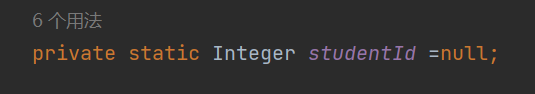
在前端中：



前端在此地方添加onLookButtonClick响应事件，在studentController中添加同名方法，并在此进行查空检测，如果非空，检测studentId，并传入ScoreController的同名方法

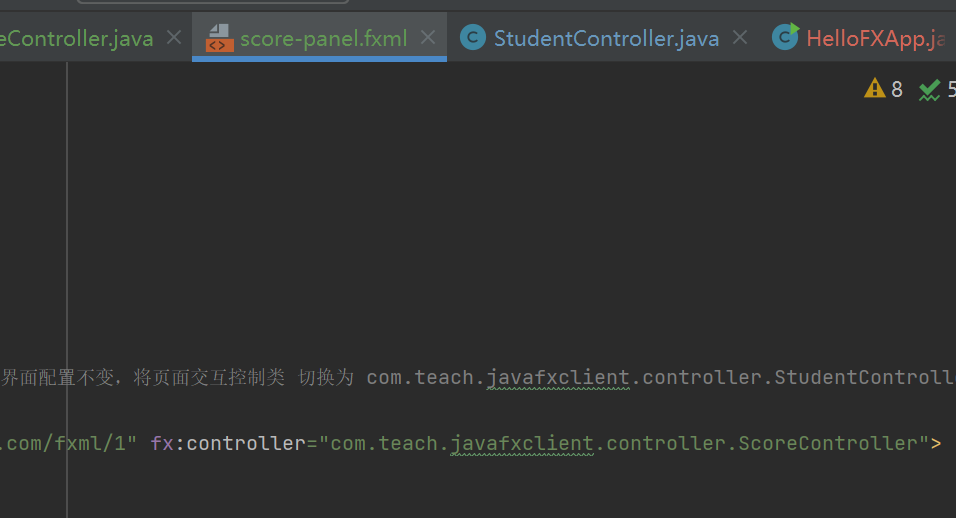


为了让接收的studentId起作用，决定设置其为static



以此，完美实现studentId串联一切的设计

在ScoreController的onLookButtonClick中，打开一个新的stage，里面装一个新的scene，打开scorefxml文件，同时在里面关联ScoreController，从而直接转到该类的Initially方法





Initialize方法中会实现数据库查询并填充表格来显示信息。

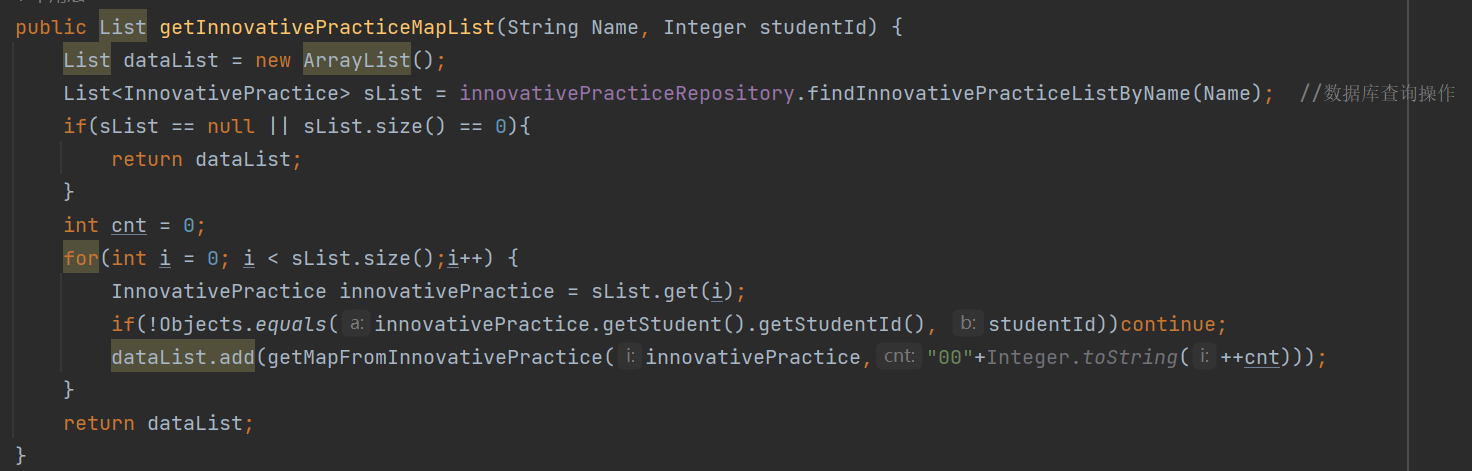
增删改查功能与上述别无二致，不再赘述。

荣誉信息，创新实践信息，日志管理，日常活动：

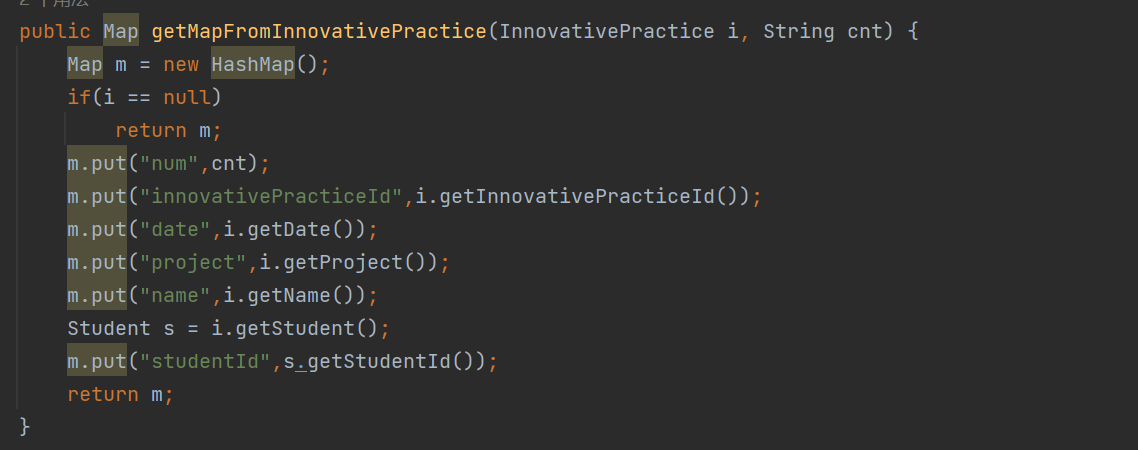
此四类具有很大共同点，皆是以名称或日期为查询方式，且都加上了编号列，实现如下：



注意到前端的num信息，来自后端传输：



++cnt，以此实现编号列永远是从1到n的顺序排列，每次查询，添加，删除后查询时都不会变更此顺序

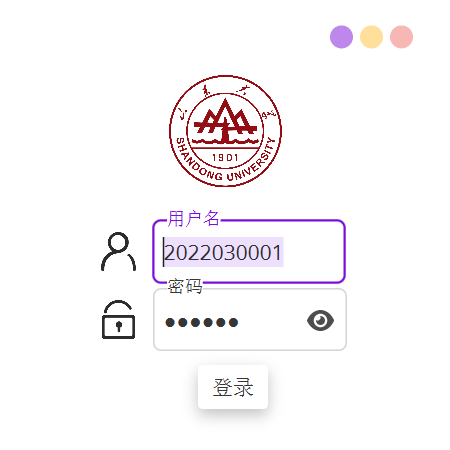


把cnt信息传入map，传入前端!



因其传输的特殊性，右侧状态栏便不再显示其编号

**学生端：**



输入学号+密码来登录



主页面，上方是一些信息栏显示信息，往下是一个Html

编辑器，可以控制简历输出信息，往下是学生选课信息及分数查看，不支持修改，往下是消费直方图和成绩饼图展示，

下方前两个按钮是相关信息，支持修改。

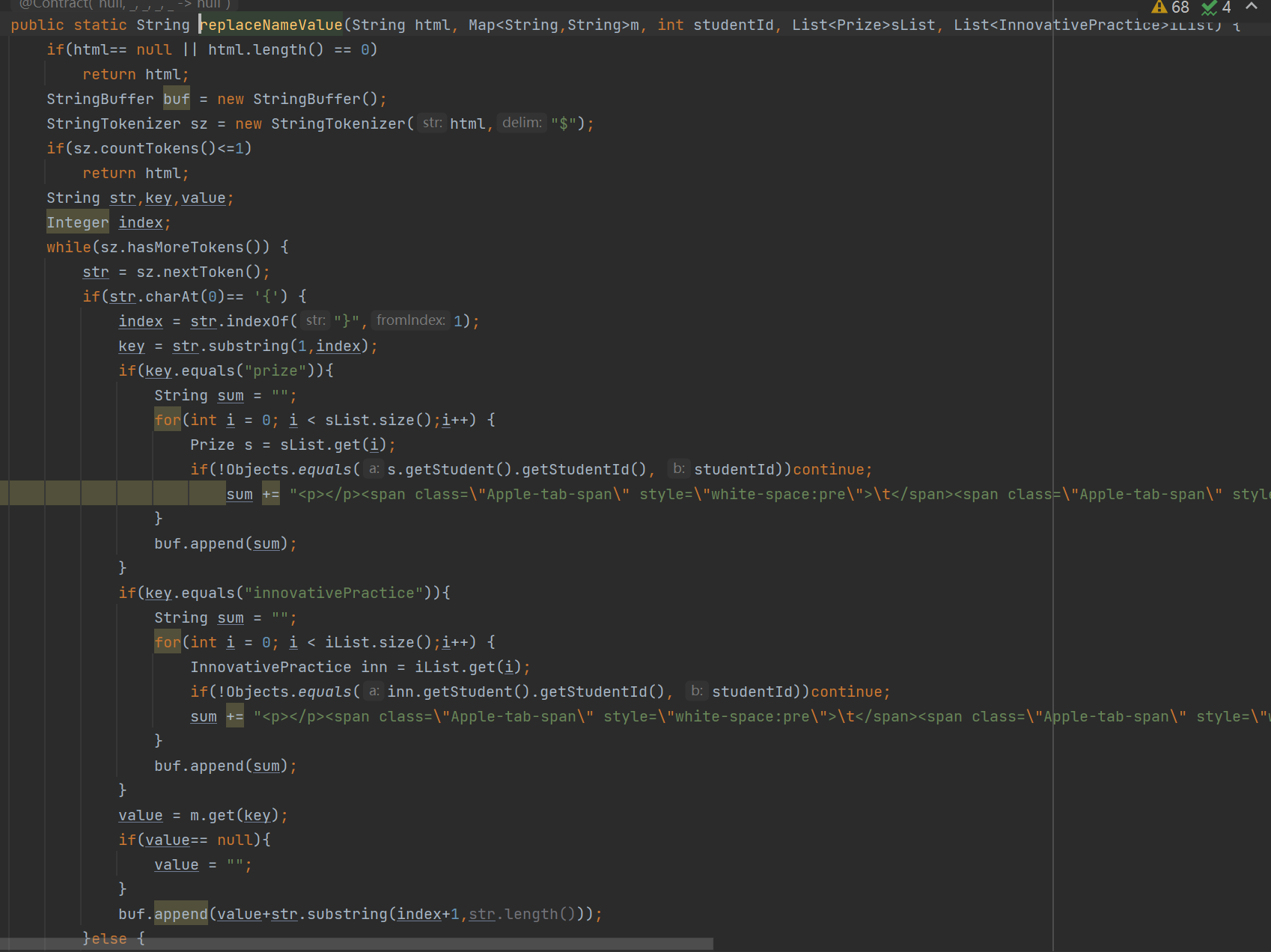
第四个按钮是Html信息提交。

第三个按钮查看个人简历。

其后端功能如下：



主要操作点在replaceNameValue()方法:





此方法5个参数：html字符串，学生信息的map，studentId，Prize的list和InnovativePractice的list。

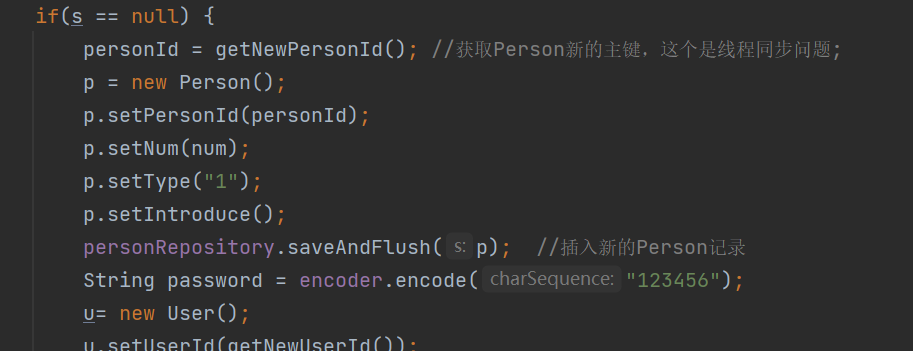
后两个在getStudentIntroducePdf（）方法中通过prizeRepository.findPrizeListByName("")和innovativePracticeRepository.findInnovativePracticeListByName("")查询得到。

然后在该方法中，遍历list，筛选与传入的studentId相同的Prize或InnovativePractice对象，并从html语言中找到&{prize}或&{innovativePractice}并将之转换为相信的信息，同时每找到一个就打印回车+两个tap控制格式。

这一切做完之后，在Person类中重写setIntroduce方法：



在新增学生时自动设置：



避免了手动操作，以后添加学生后可以自动打印了。

其余功能与前面亦别无二致，不再赘述。

1. **个人总结和体会**

**姓名：程前**

**主要负责部分**：负责后端的功能实现和与前端的一些链接等等

**个人总结：**

刚开始的时候，什么也不会，一头雾水，不知道怎么做。于是就拿着老师的代码研究，并随时翻阅资料，查找信息，终于越来越有眉目，知道了如何去做，也开始一点点写东西。

中间遇到了不少困难，第一个完整的方法实现用了两天，不过之后就好了很多，速度也越来越快，在与队友的配合下，最后终于堪堪完成了项目，收获了很多。

不过对于整个项目的逻辑功能，还是有很多不熟练的地方，很多地方因为我的考虑不周，也出现了bug，不过，吸取了这次的教训，我相信对于以后的项目，或是任何学习的相关事情，都能多考虑一点，都能更快，更好！

“莫道春光难揽取，浮云过后艳阳天”。最后看着我们的成果，心里还是很自豪的，毕竟，这是我们的努力。

**姓名**：姚仁广

**主要负责部分**：负责前端界面的绘制优化修改

**个人总结**：我自己比较擅长前端，在网络上也找到了遵循谷歌Material Design的JavaFX组件库，所以就写了前端，好在组件库的作者给了演示程序和wiki，在阅读教程的过程中对于我英语能力提升了不少，但也只是大改了登陆界面，其他界面只是把组件修改了一下，主要还是因为JavaFX教程过少，找不太到相关教程，不太会改menubar，所以其他界面大致与老师给的原项目保持一致。相信以后学习其他前端框架我能运用的更好，也希望我在后端也能有所进步。