|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| 课设项目开发报告 | | | | |
| 高级程序开发（双语） | | | | |
|  | | | | |
|  | 组 员 姓 名： | ： |  |  |
| 指 导 教 师： | ： | 鹿旭东 |
| 学 院 名 称： | ： | 软件学院 |
| 班 级： | ： | 23级1班 |
| 二〇二三 年 五 月 | | | | |
|  | | | | |

目录

[第一章 系统总体要求 3](#_Toc15491)

[1、系统总体功能要求 3](#_Toc26407)

[2、开发平台介绍 3](#_Toc21785)

[3、开发进度表 3](#_Toc23666)

[4、小组成员分工及比例 4](#_Toc23666)

[第二章 学生信息管理系统分析与设计基本过程 4](#_Toc16822)

[1、学生信息系统用例 4](#_Toc23857)

[（1）编写目的与项目背景 5](#_Toc12634)

[（2）各类管理用例 5](#_Toc26562)

[（3）项目说明 9](#_Toc30673)

[2、学生信息管理系统领域模型 9](#_Toc2553)

[（1）UML类图 9](#_Toc14326)

[（2）学生管理系统总时序图（以管理员为例） 1](#_Toc15954)0

[第三章 系统设计 1](#_Toc304)1

[1、系统概要设计 1](#_Toc25076)1

[（1）系统总体框架 1](#_Toc23933)1

[（2）系统工作流程 1](#_Toc2362)1

[（3）系统功能模块的划分](#_Toc23933) 14

[2、系统详细设计 2](#_Toc16398)1

[（1）实体关系图（部分） 2](#_Toc23933)1

[（2）数据库表 2](#_Toc2362)1

[第四章 系统实现 2](#_Toc11985)7

[1、实体类基本逻辑 2](#_Toc24947)8

[2、控制逻辑 3](#_Toc25302)0

[（1）注解的作用 3](#_Toc13699)0

[（2）增删查改功能的实现](#_Toc4333) 31

[（3）课程中心相关功能的实现 3](#_Toc15501)7

[（4）个人画像与个人简历功能 4](#_Toc6016)3

[（5）验证码功能和密码修改功能 4](#_Toc26337)8

[（6）时间功能 4](#_Toc11014)9

[3、数据表维护查询 5](#_Toc15517)0

**第一章 系统总体要求//这一章我只有“1、系统总体功能要求”没动，其他的改好了**

1、系统总体功能要求//光改这个就行了

本次设计要求利用Java实现一个学生信息管理平台，是一个前后台分离的应用，前端基于Java FX 开发GUI图形交互界面，后端基于Java String Boot 开发，实现数据的操作与处理。主要功能为：

（1）学生基本信息、联系方式、入学前信息、家庭信息、社会关系等基本信息的管理

（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

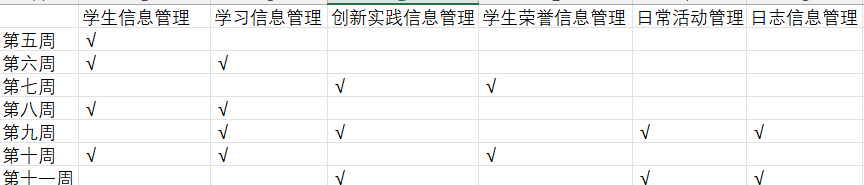
（8）学生各种信息的查询统计、综合绩分的计算（可自行设计公式）和学生个人画像、个人简历的生成打印

（所谓信息管理就是信息的CRUD——添加、查询、修改、删除）

2、开发平台介绍

IDEA，即Intellij IDEA——是高度集成的Java语言开放环境。被公认为是功能最强大，使用最广泛的Java开放工具之一。

3、开发进度表



4、小组成员分工及比例

|  |  |
| --- | --- |
| 小组成员分工 | |
|  | 1.设计表间关系，编写modle中的实体类。 |
| 2.编写前后端Controller中的方法。 |
| 3.前后端联调和新功能的设计和实现。 |
| 4.部分界面的制作和修改。 |
| 5.完成项目报告中所有UML图的绘画制作。 |
|  | 1.学生端相关前端Controller代码编写 |
| 2.运用Javafx进行项目的前端界面设计 |
| 3.项目完成后的测试和数据编写。 |
| 4.实验报告部分内容的编写。 |
| 5. 设计项目功能，参与表的结构设计。 |
|  | 1.实现系统页面调用的功能，修改页面部分bug。 |
| 2.对前端Controller进行编写。 |
| 3.编写项目报告中有关前端设计的部分。 |
| 4.进行前后端的对接。 |
| 5.项目报告中大部分内容的编写。 |
|  | 1.运用JavaFX进行项目的前端界面设计。 |
| 2.编写css层叠样式文件对前端界面进行美化。 |
| 3.测试系统功能的正确性与完善性。 |
| 4.编写项目报告内容。 |
| 5.改善报告排版 |

个人贡献比例：

=1：1：1：1

**第二章 学生信息管理系统分析与设计基本过程**

1、学生信息系统用例

**（1）编写目的与项目背景//不用改**

编写目的：使本系统的使用者对本系统各部分架构认识更为清晰，明确本系统的工作流程及具体功能实现的原理，从而更好的通过本系统实现对学生信息的统一管理。

项目背景：本项目是一个学生信息管理系统，其主要用户群体为老师与学生，主要功能实现以学生为主。可以实现老师和学生用户自身对相关信息的统一集中管理。具有较强的实用性和泛用性。

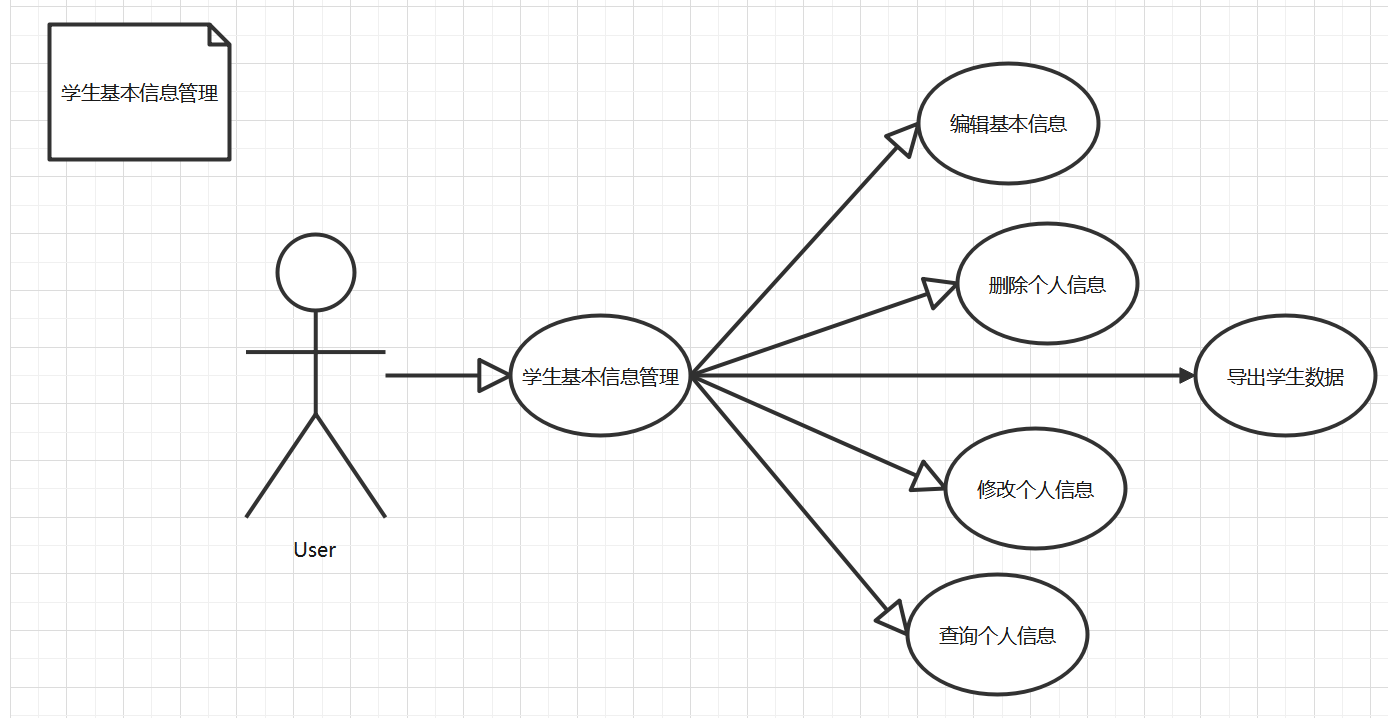
**（2）各类管理用例//从a到e是按咱们界面的字条写的框架，内容有的和咱一样的保留了学长的（文字+图片），但大部分都是空着的**

**a.总用例：**

管理员使用此系统可完成对教师，学生的多项管管理任务，包括：学生相关信息管理。教师相关信息管理，创新实践信息管理，学生荣誉信息管理，日常活动信息管理，日志信息管理等多项功能。

**b.人员管理**

**【1】学生管理**



1）管理员可通过此功能完成对学生相关信息的增，删，改，查等基本操作。

2）基本信息包括学生照片，姓名，学号，性别，专业，年级，班级，生日，联络方式，父母姓名与电话等。

3）可通过姓名，学号两种信息搜索学生。

4）一键导出学生数据(pdf格式)

5）对教师的管理与之类似。

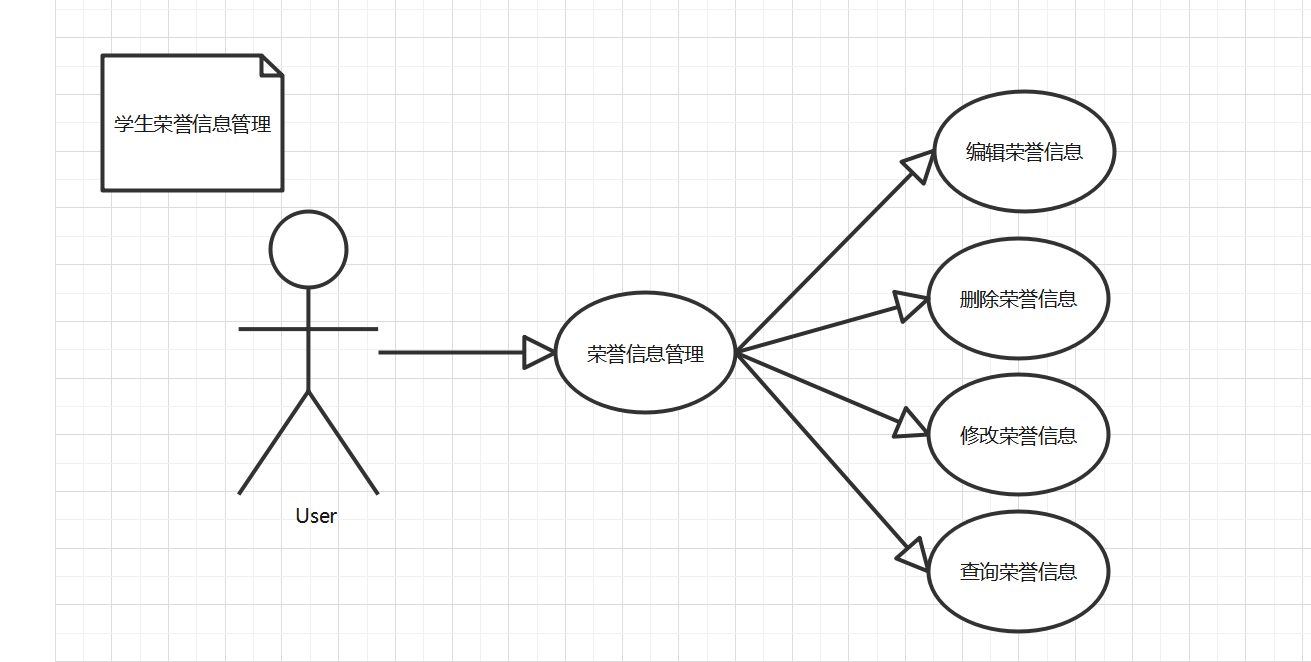
**【2】教师管理**

**【3】家庭信息**

**c.教务管理**

**【1】课程管理**

**【2】学生荣誉**



1）通过此功能完成对学生的各类荣誉信息管理（增删改查）；

2）搜索可通过姓名，荣誉名称，学号。

**【3】作业管理**

**【4】创新实践管理**

1）通过此功能完成对学生创新实践信息，包括社会实践，竞赛，科研等信息的增删改查操作；

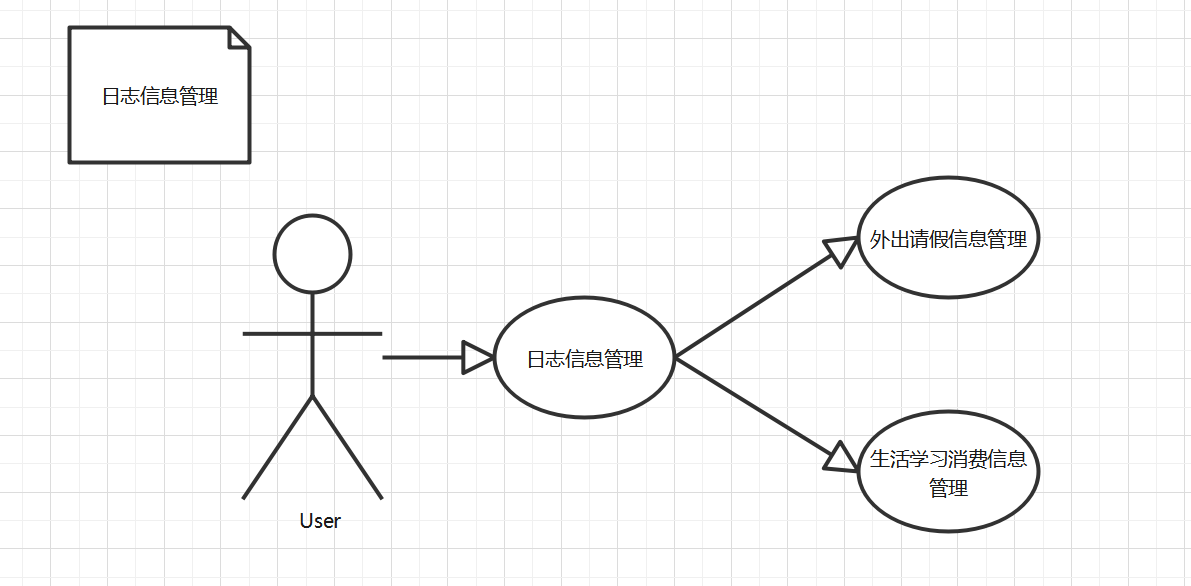
2）可通过学号，项目名称来搜索。

**d.日常活动管理**

1）通过此功能完成对各类日常活动信息的增删改查等操作；

2）可通过学号，姓名搜索。

**e.日志信息管理**



1）通过外出请假信息管理完成对学生请假信息的增删改查等基本操作，以及审批操作；

2）通过生活消费信息管理完成对消费信息的增删改查的基本操作。

**（3）项目说明//不知道咱有没有简历导出啥的，这里你们看着改吧我没动**

a.学生是此系统的主要用户，可以方便的完成各类信息（个人信息，学习学习，活动信息，荣誉信息等）的管理，以及个人画像的生成，简历的生成导出，绩点的计算等功能。

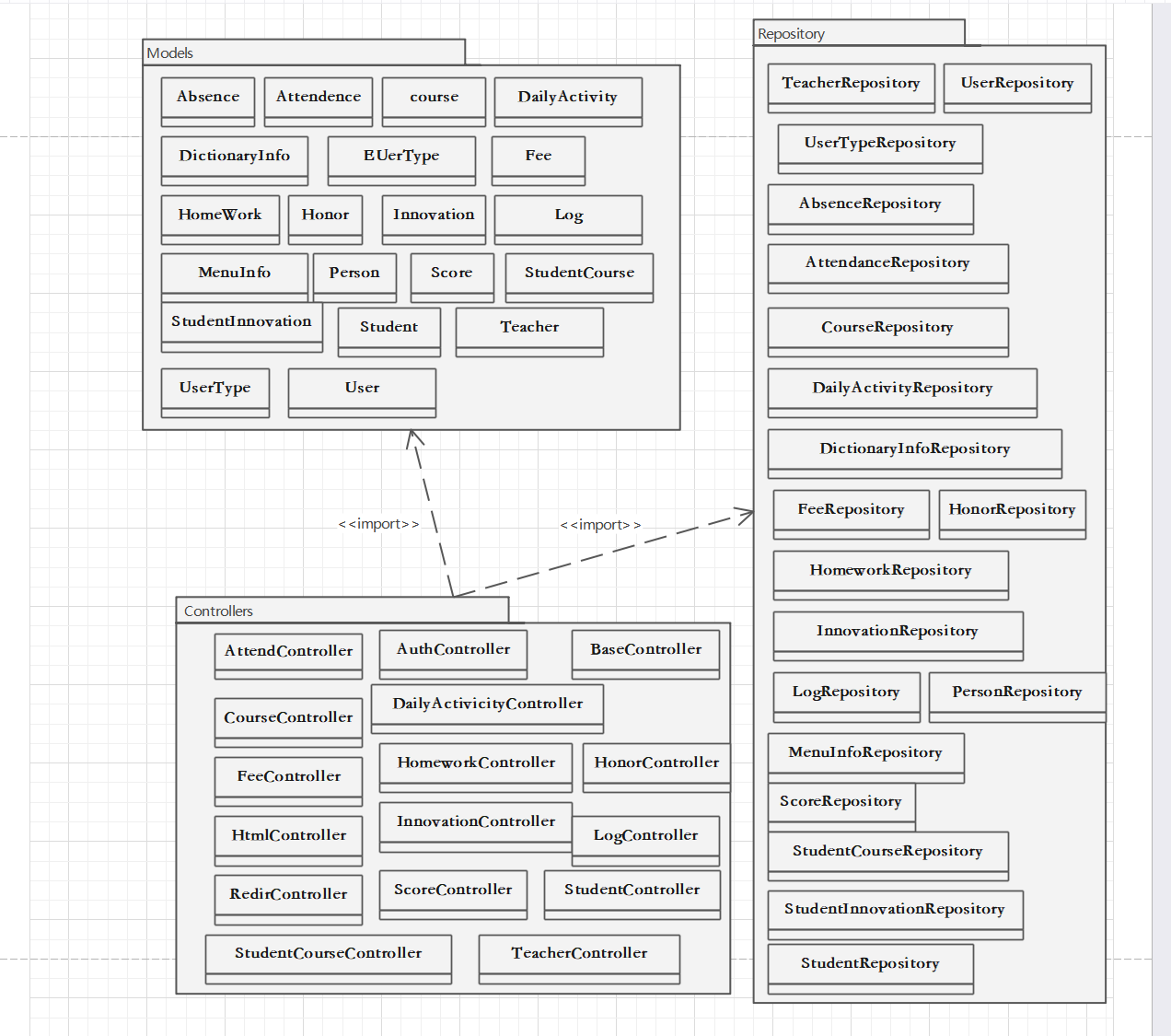
b.教师端与学生端类似，增加功能有所任课程信息的管理（资料上传下载，教科书编辑，课程信息编辑等），不作为展示重点。

c.管理员admin拥有最高权限，对学生和教师的信息进行统一管理，以及开放/删除选课，审批请假，查询消费信息等高级功能。

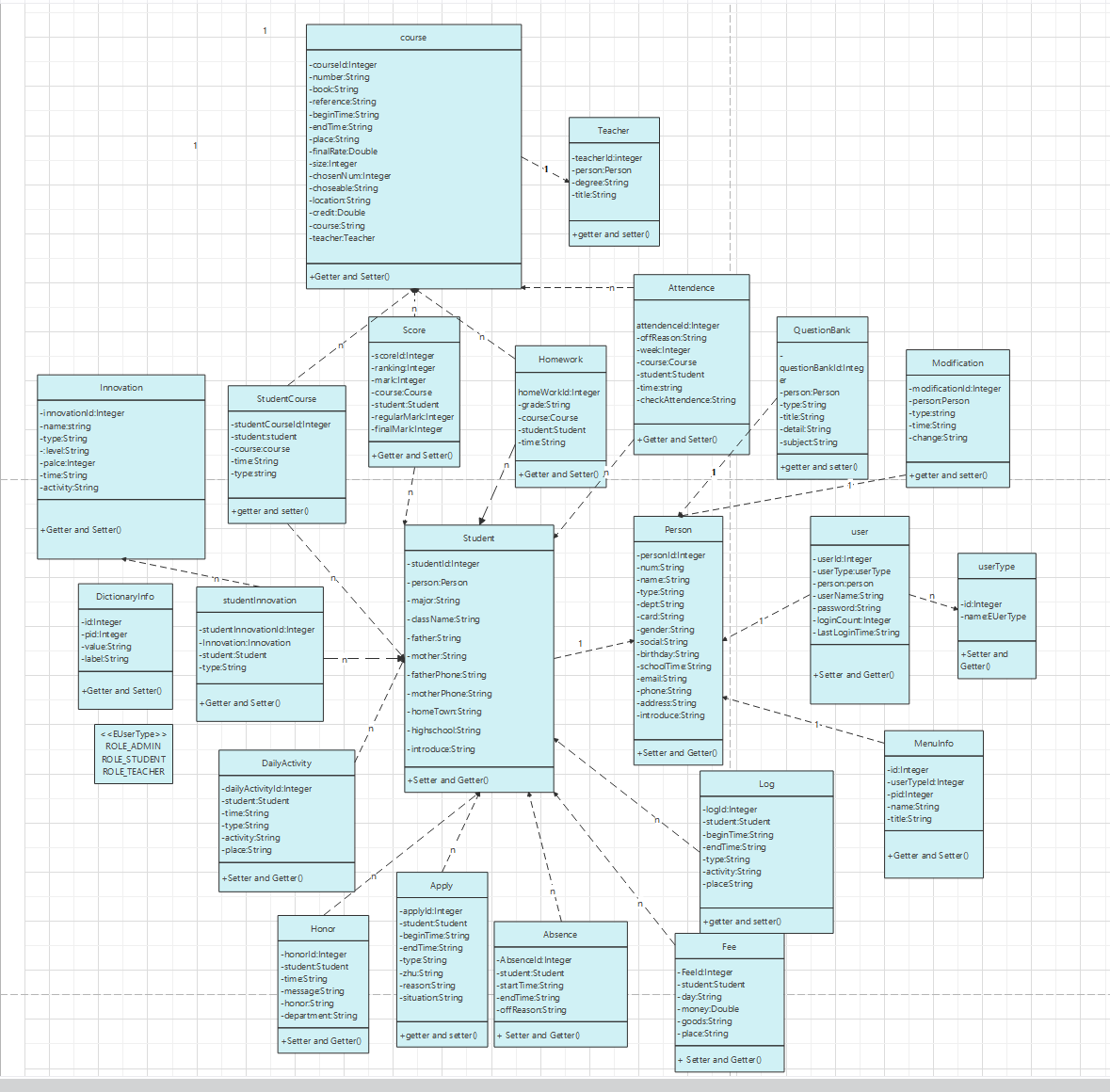
2、学生信息管理系统领域模型

**（1）UML类图//我没动，我看不明白**

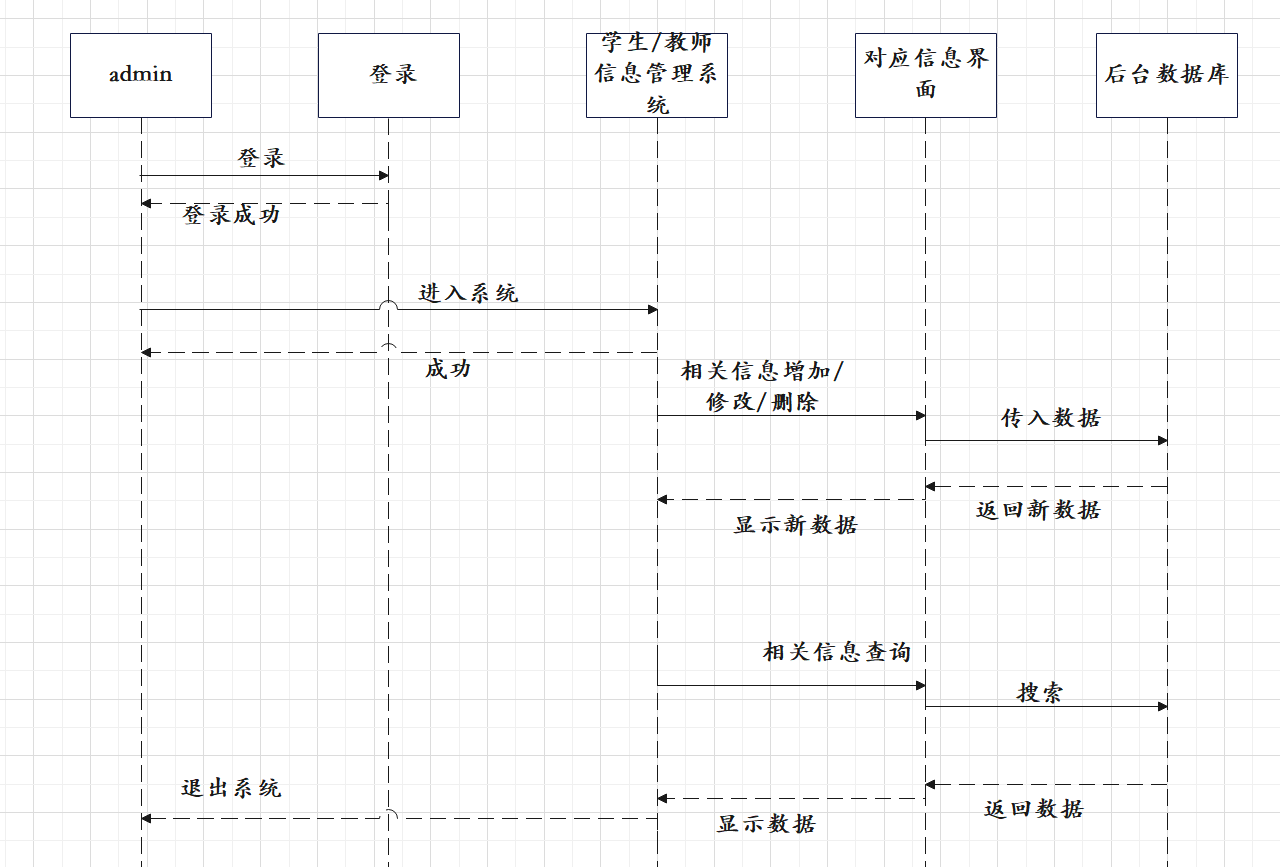
**a.包图**



**b.类图**



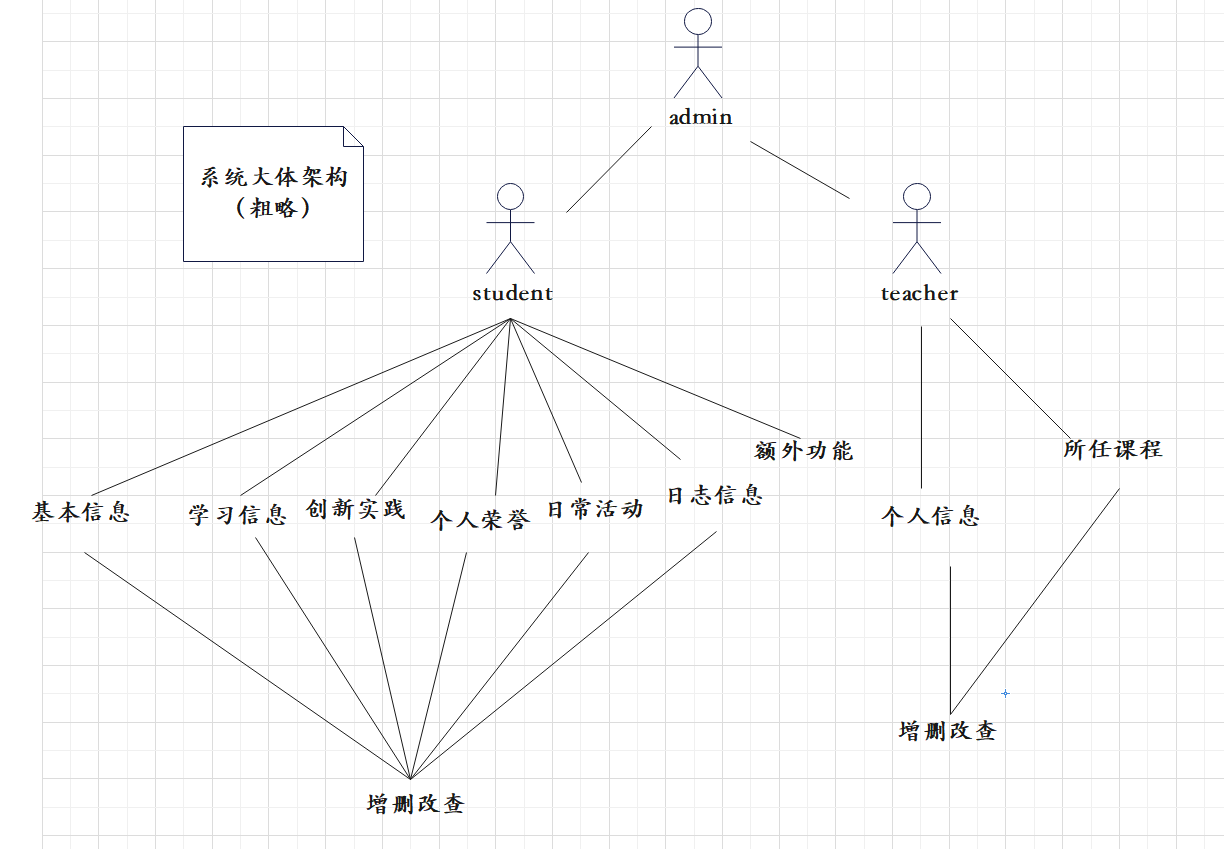
**（2）学生管理系统总时序图（以管理员为例）**



**第三章 系统设计**

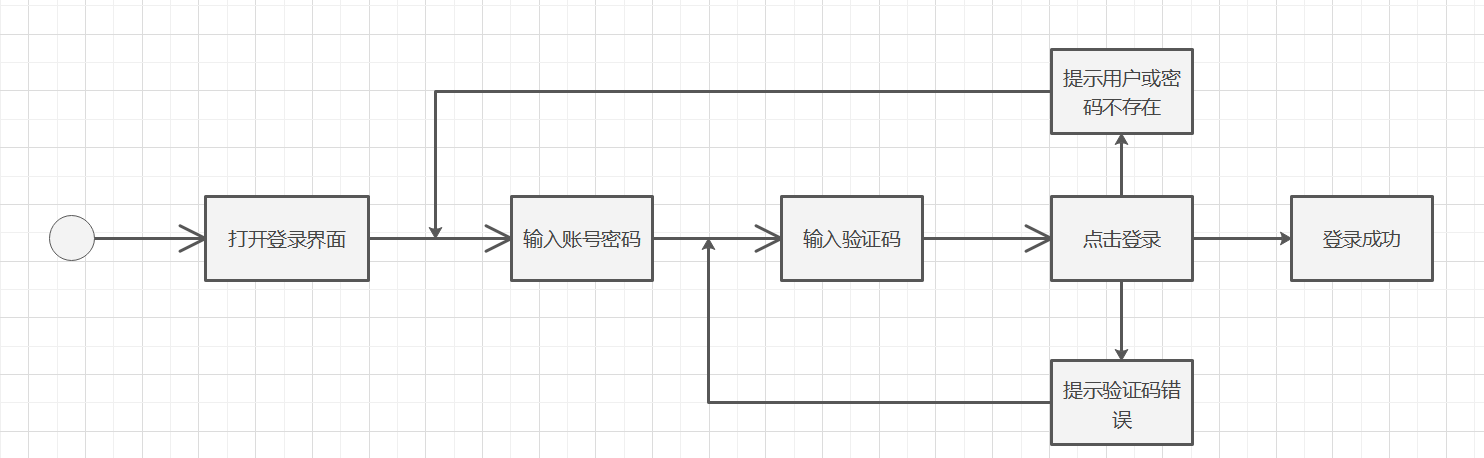
**1、系统概要设计**

**（1）系统总体架构//和咱的界面字条不太一样啊**

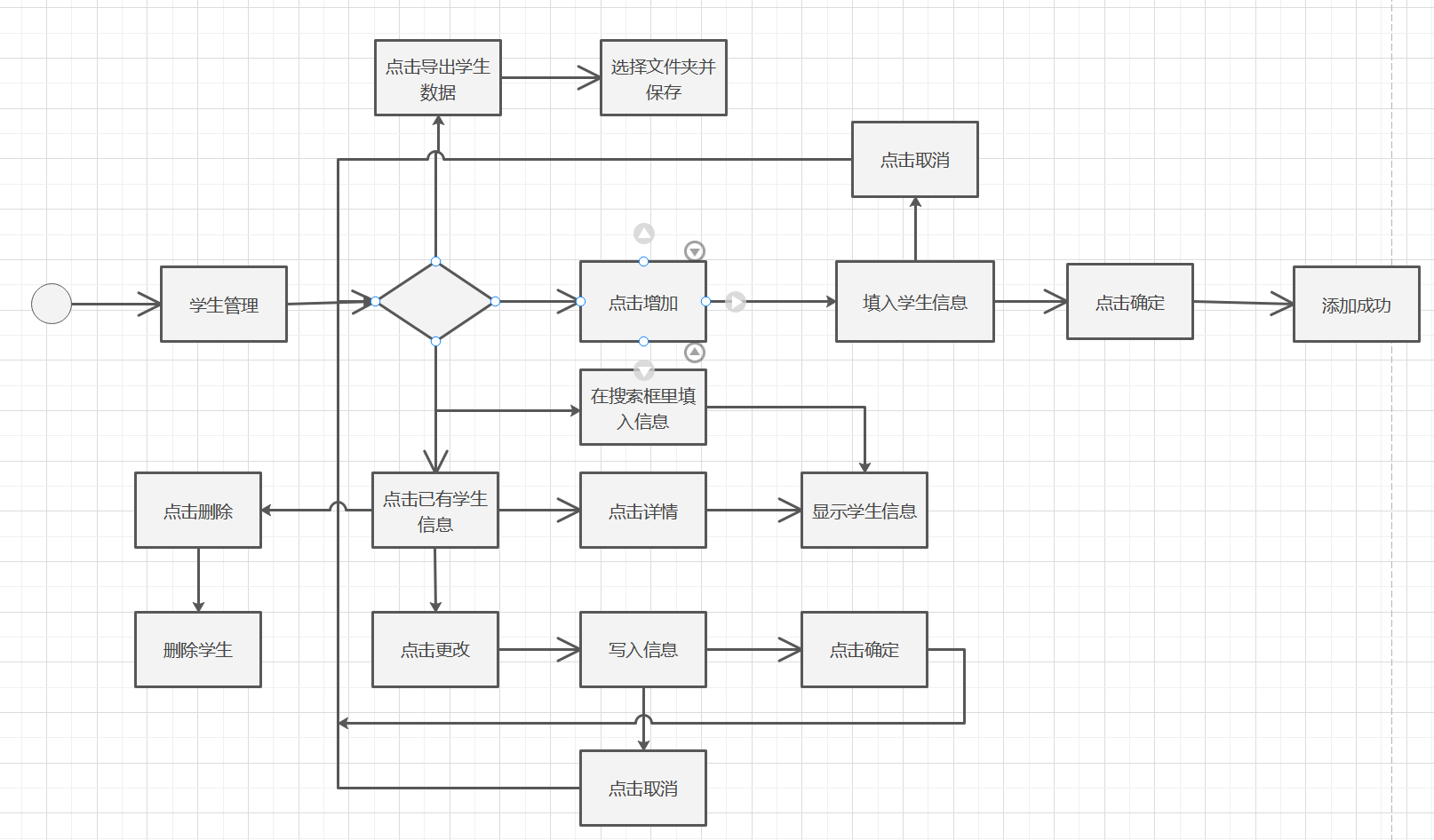


**（2）系统工作流程//我没动，框架也没动**

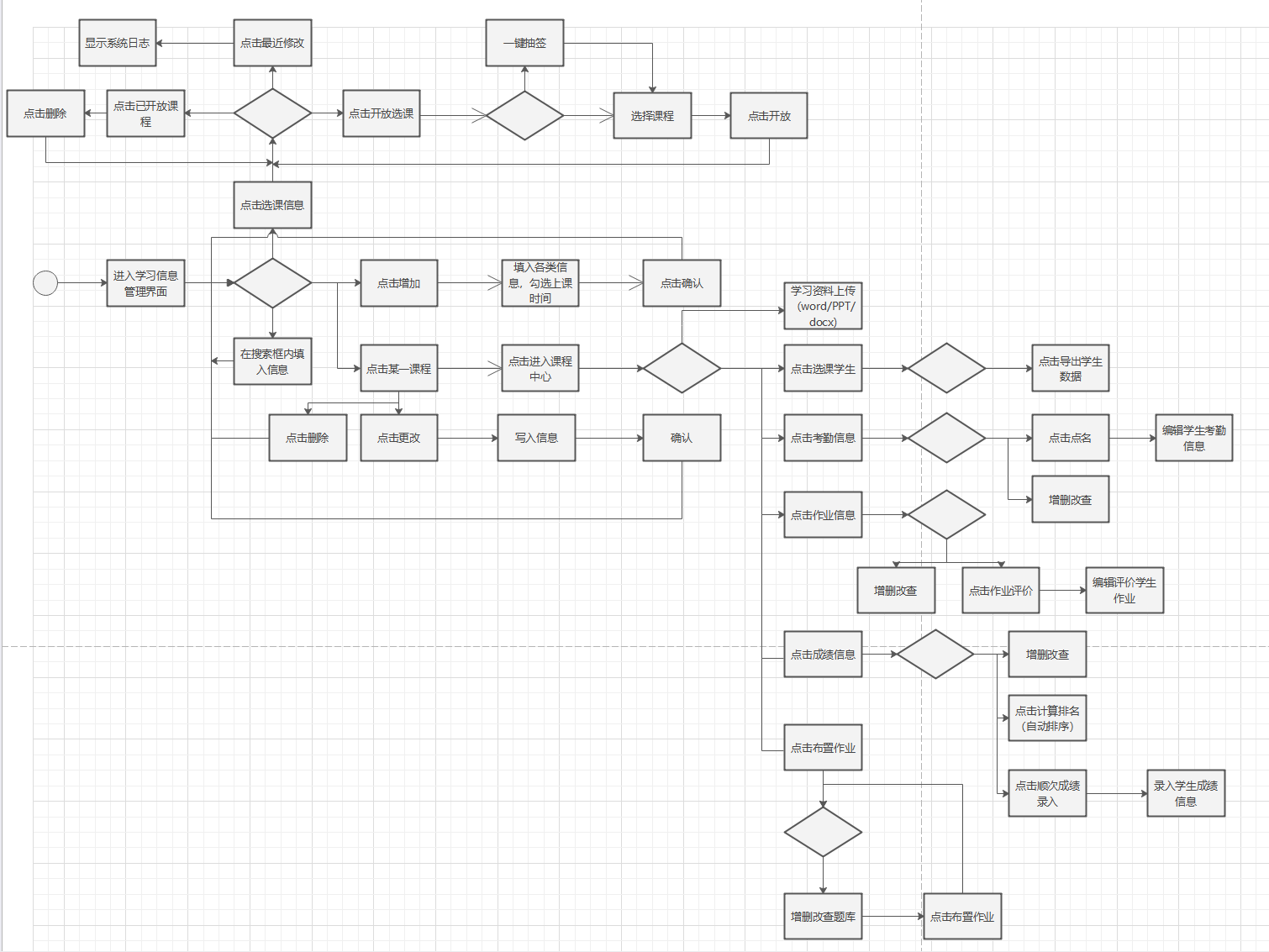
**a.登录流程**



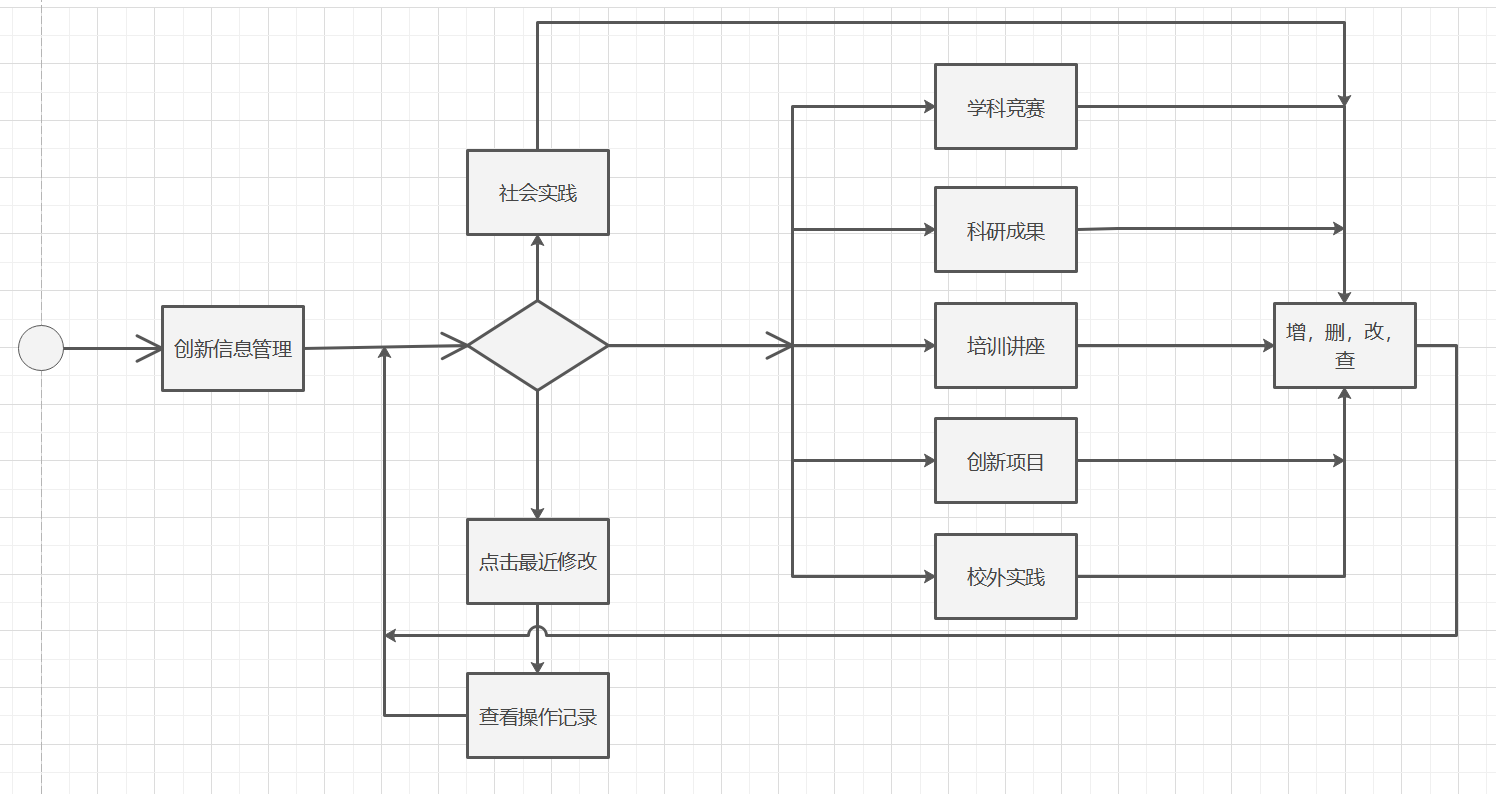
**b.学生管理流程图**



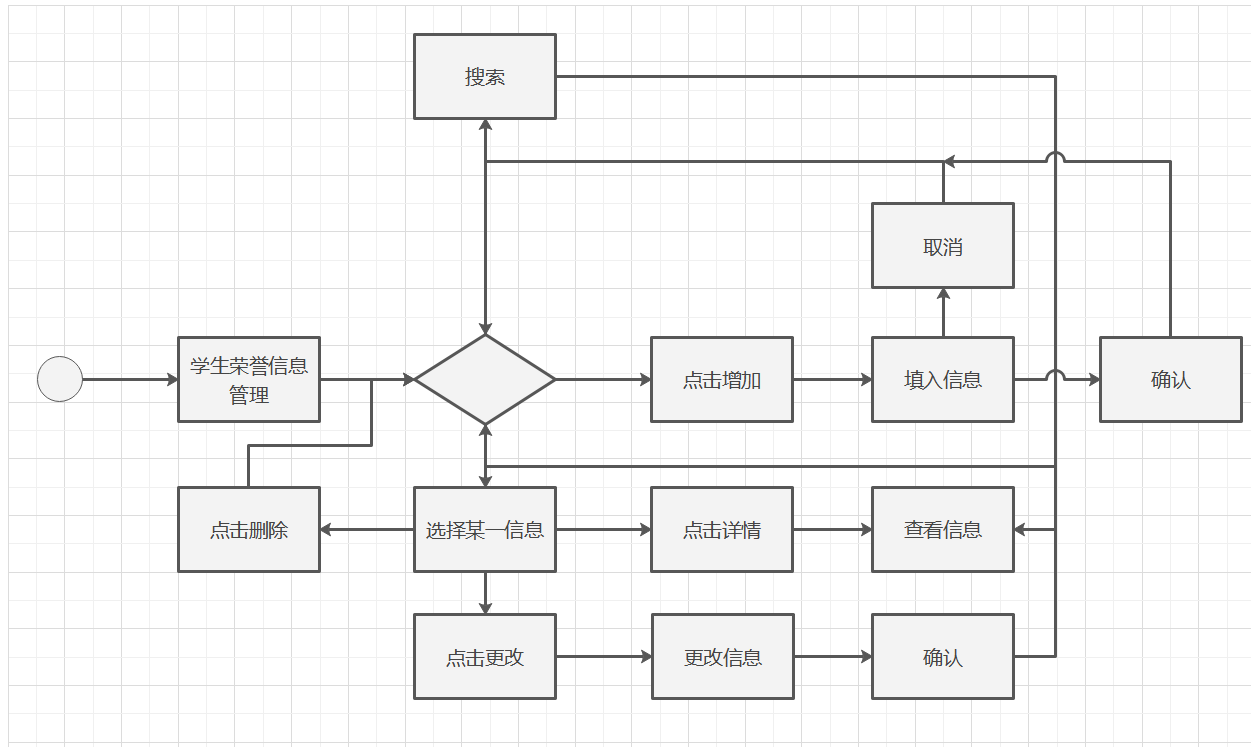
**c.学习信息管理流程**



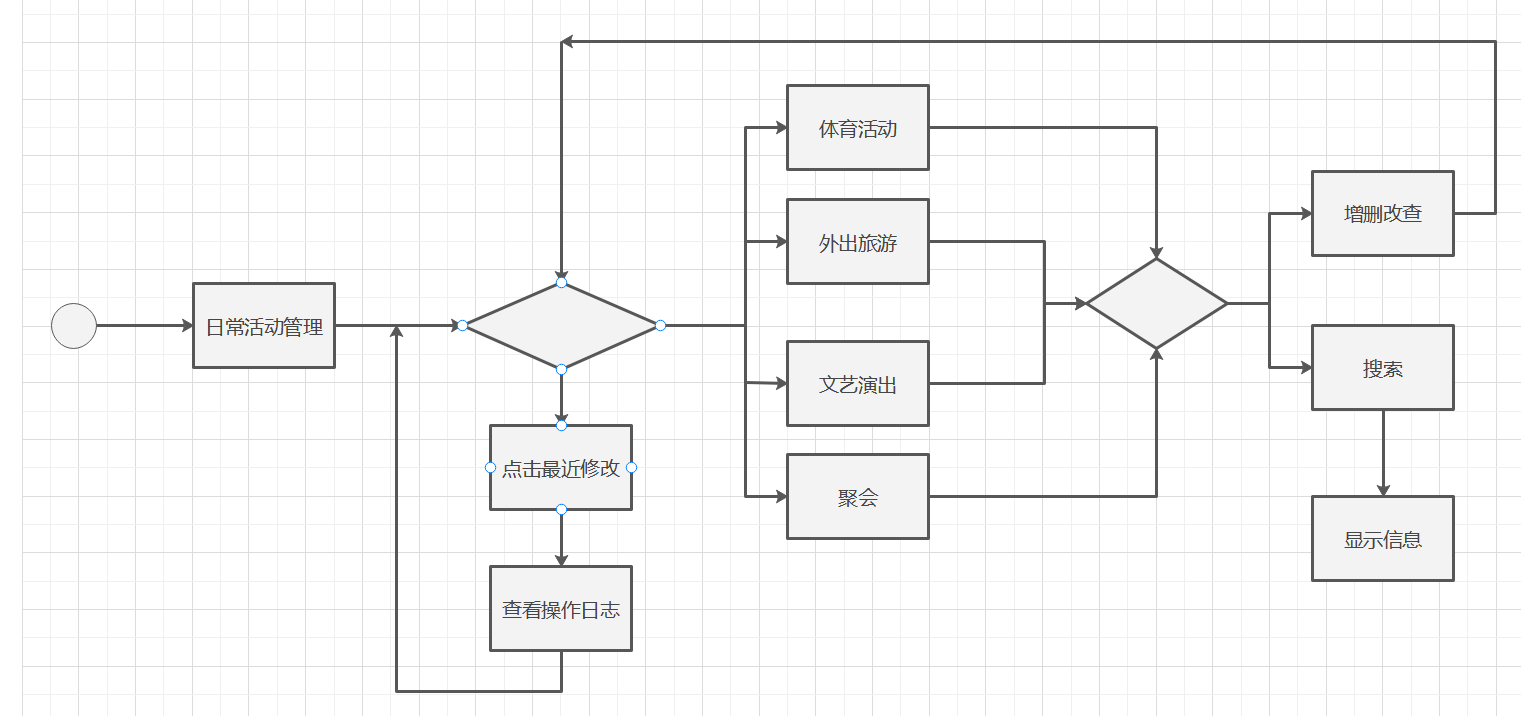
**d.创新信息管理流程**



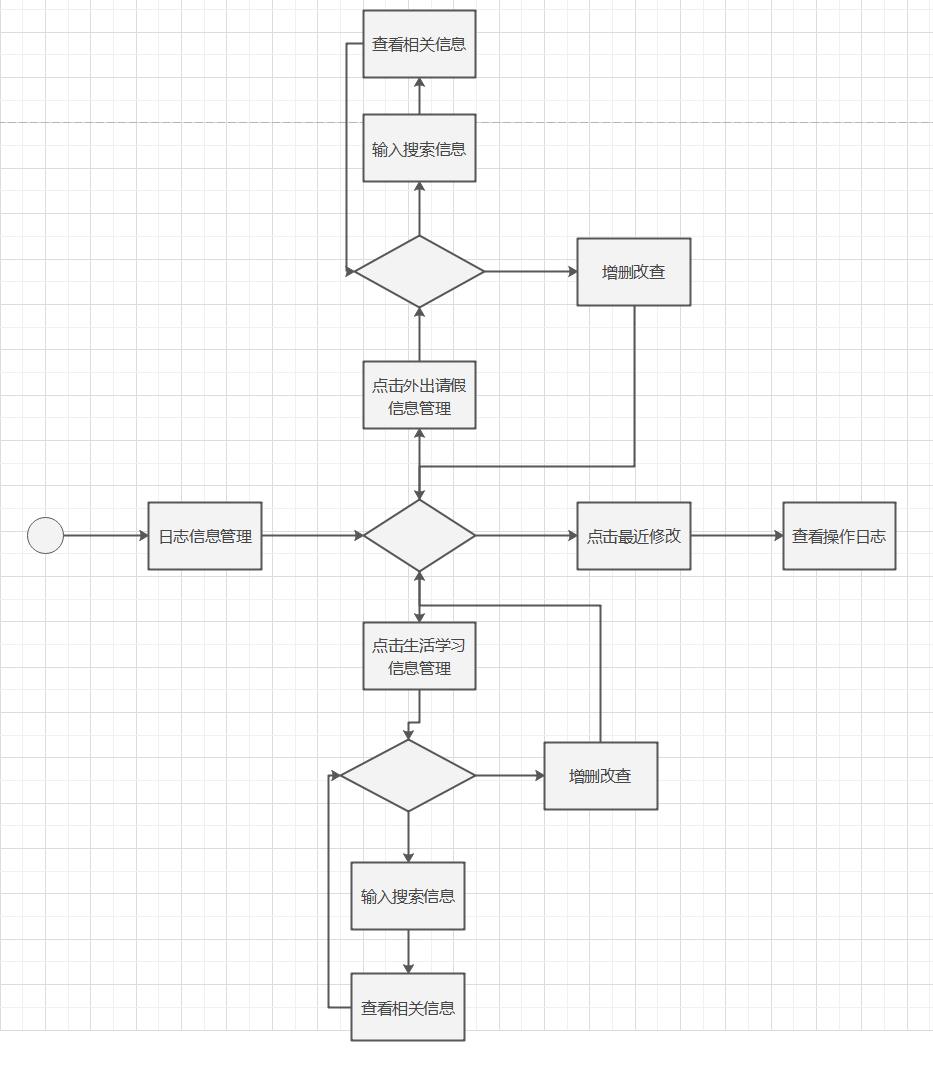
**e.学生荣誉信息管理流程**



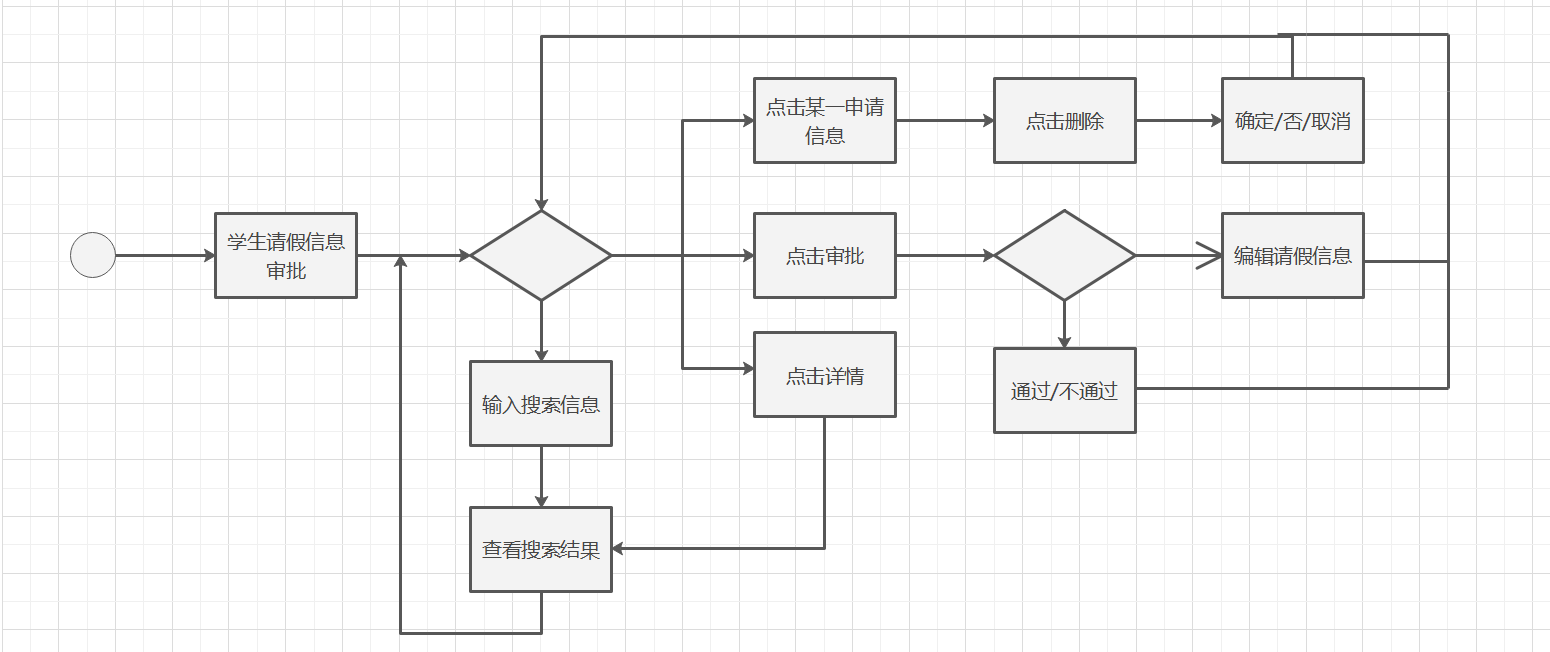
**f.日常活动管理流程**



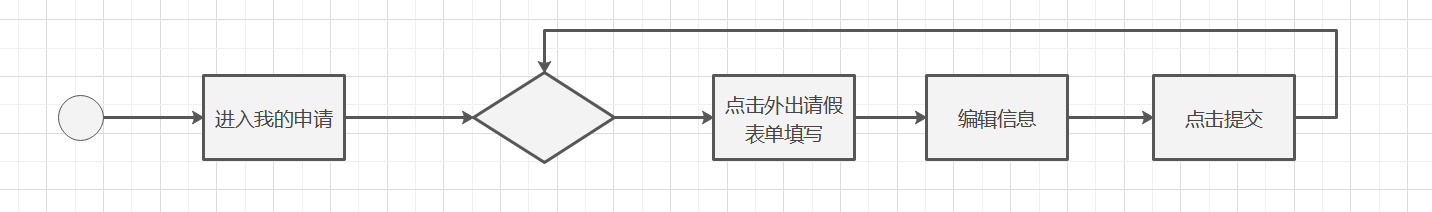
**g.日志信息管理流程**



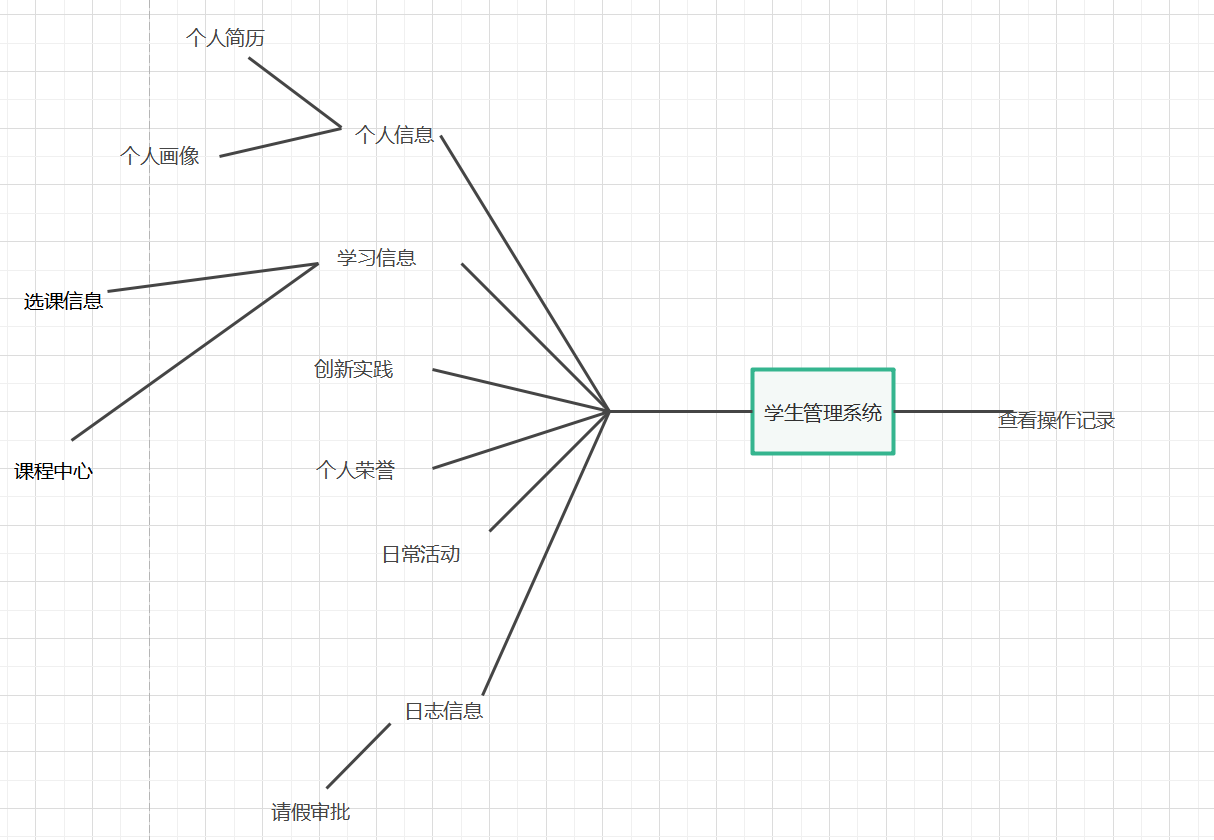
**h.请假信息审批流程**



**i.学生请假信息申请**



**（3）系统功能模块的划分//从这里一直到第四章我只删了学长的课设界面截图，其他的没动**



功能模块主要分为：**个人信息，学习信息，创新实践，个人荣誉，日常活动，日志信息**六大类的各类操作。

**a.个人信息**

个人简历由个人基本信息，荣誉信息等信息综合生成，可一键导出到本地。

**b.学习信息**

学习信息中有“课程中心”模块，可进行的学习资料下载、绩点计算、生成错题本、生成课表；老师/管理员则可以通过此模块上传添加题库，布置作业，录入成绩，导出学生名单，点名，批改作业，在选课信息中可进行开放/取消选课等。

**1）学生下载资料**

**2）计算绩点**

先选中下面的课程（勾选）然后点成绩可视化，出现下面的界面展示绩点：

**3）生成错题本**

**4）生成课表**

课表也是自动生成，学生选中课程生成课表信息。

**5）教师可录入成绩，计算排名，布置作业**

**6）管理员对学生请假信息的审批**

流程为：学生编辑申请，管理员可编辑（同意/不同意），操作完成后信息返回学生端，学生请假信息界面会显示（同意/不同意）。

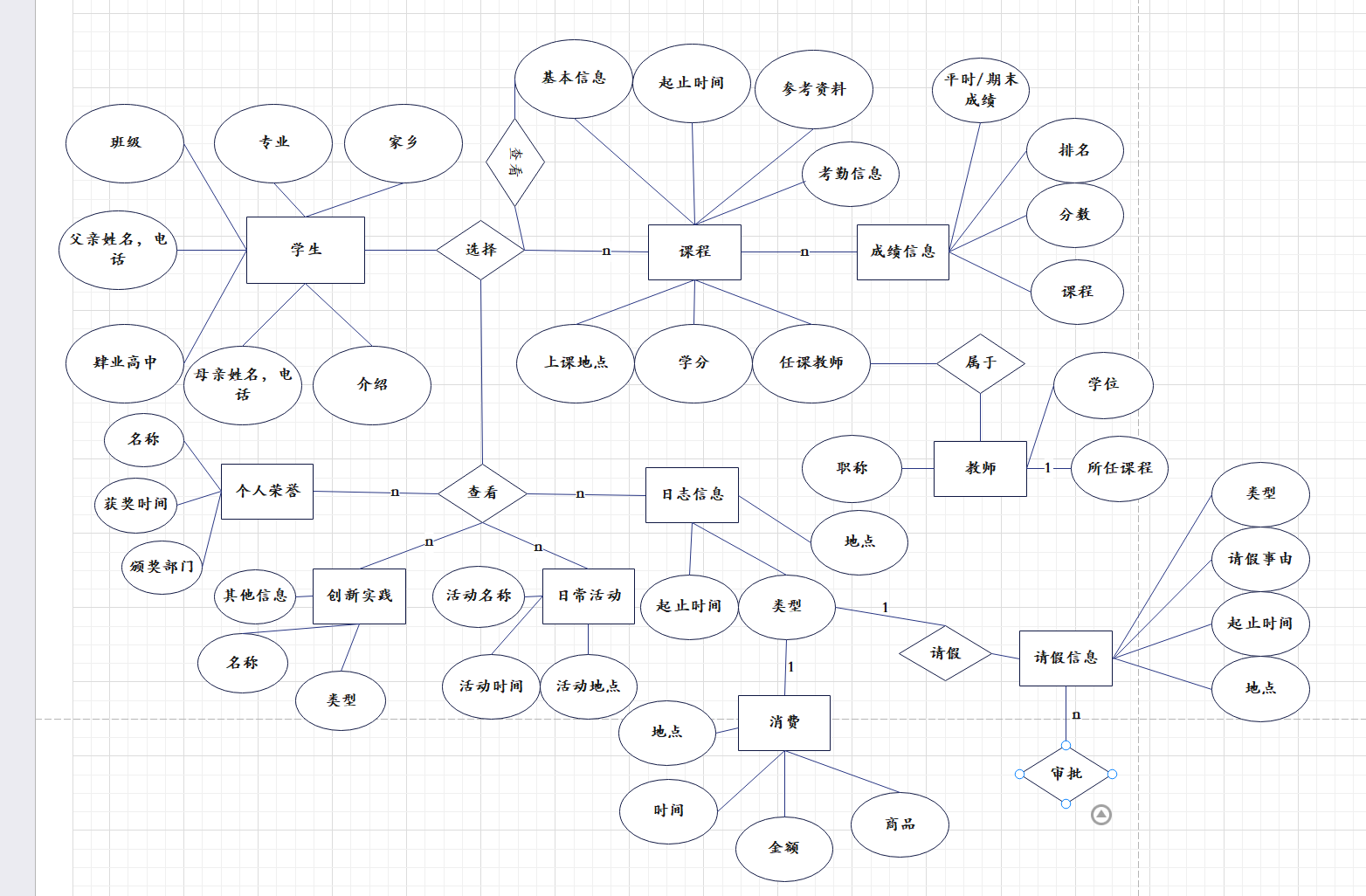
**7）管理员开放选课**

具体流程为：首先任课教师通过本系统编辑添加一门课程，然后管理员进行开放选课，最后学生才可以开始选课。

**2、系统详细设计**

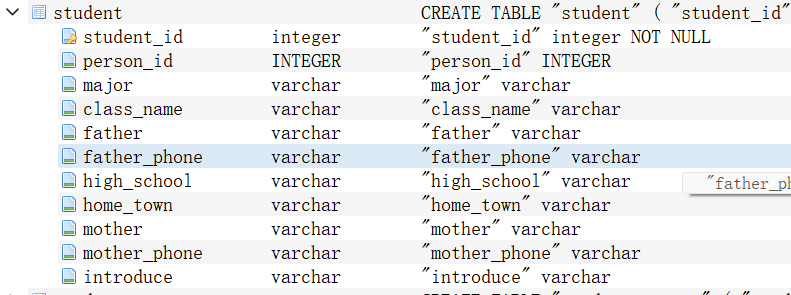
**（1）实体关系图（部分）**

以学生实体为中心，其他实体围绕学生展开。



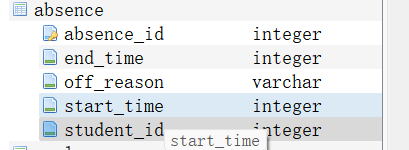
**（2）数据库表**

**a.student表**



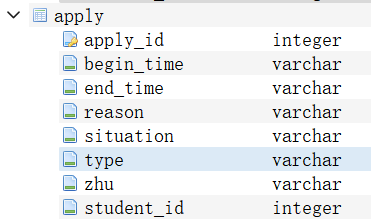
student表是最核心的表，用于保存学生基本信息，主键为student\_id，是很多其他表的外键。

**b.absence表**



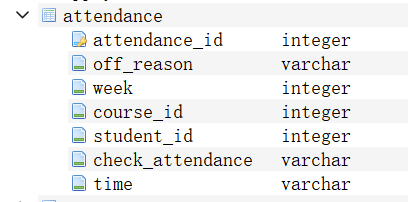
记录学生请假信息：包括请假事由、起止时间，主键为absence\_id,外键为 student\_id,二者为ManyToOne关系。

**c.apply表**



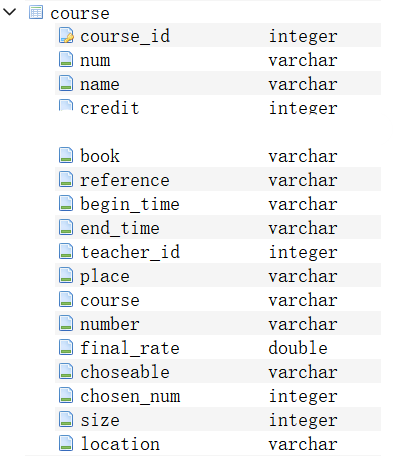
记录学生申请信息：包括起止时间、原因、类型等，主键为apply\_id，外键为student\_id，二者为ManyToOne关系。

**d.attendance表**



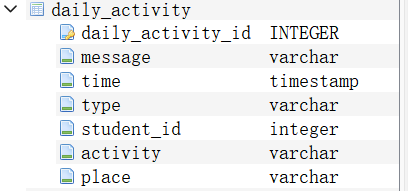
记录到场信息：student与course的中间表，主键为attendance\_id，外键为course\_id和student\_id，对二者均为多对一关系。

**e.course表**



保存课程信息：包括名称、上课时间、教师、上课地点、选课人数、容量、学分等，主键为course\_id，对外键teacher\_id为一对一。

**g.daily\_activity 表**

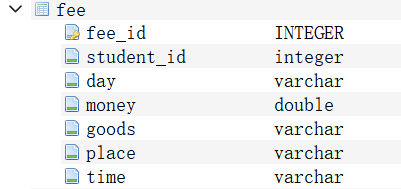


保存日常活动信息：包括基本信息、类型、地点，主键为 daily\_activity\_id,对外键student\_id为多对一关系。

**h.dictionary 表**

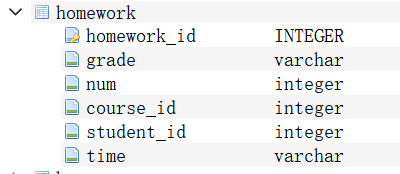


**i.fee表**



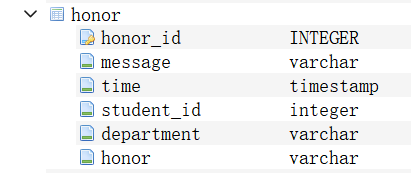
保存消费信息：包括时间，金额，商品，地点等，主键为fee\_id,对外键student\_id为多对一关系。

**j.homework表**



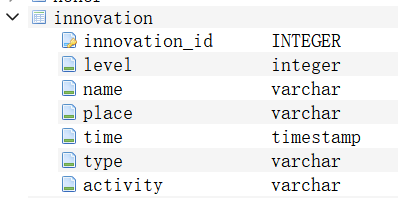
保存作业信息：包括等级、时间。主键为homework\_id，对外键student\_id与 course\_id为多对一关系。

**k.honor表**



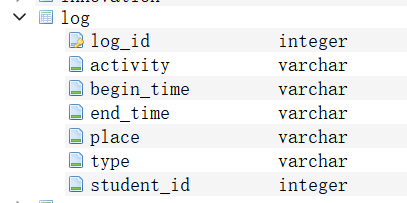
保存学生荣誉信息：包括基本信息、时间、颁发部分，主键为honor\_id,对外键student\_id为多对一关系。

**l.innovation表**



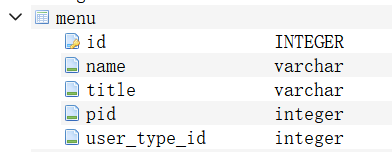
创新实践信息：包括级别、地点、时间、类型、活动。主键为innovation\_id。

**m.log表**



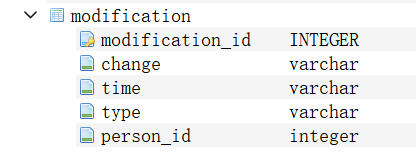
保存学生登录信息：包括起止时间、地点、类型等，主键为log\_id, 对外键student\_id为多对一。

**n.menu表**



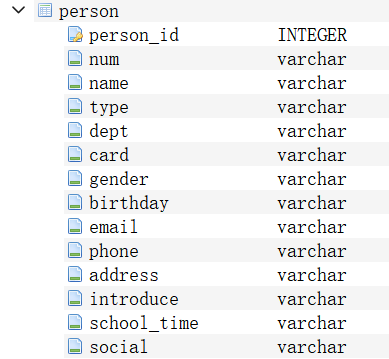
保存不同的用户类型的不同界面，主键为id，与外键person是一对一关系。

**o.modification表**



用于记录用户操作：包括变化描述、时间、类型，主键为modification\_id，与外键person\_id为一对一关系。

**p.person表**



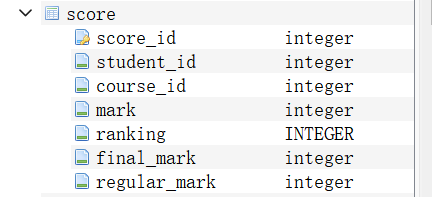
Person保存用户的基础信息，num是人员编号是唯一的。信息包括名字，类型，性别，系别，邮箱，电话，校园卡号等。每一位学生管理系统用户都是person类中的实例。

**q.question\_bank表**



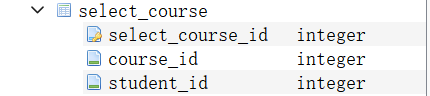
这个类保存的是学习信息下的题库信息，可由管理员和教师编辑。保存有内容、学科、标题、类型（选择/填空/判断）。主键为question\_bank\_id，与外键person\_id为多对一关系。

**r.score表**



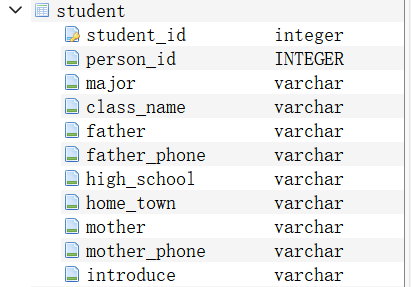
保存学生课程成绩信息：包括考试成绩、平时成绩、最终成绩、排名，主键为score\_id，与外键student\_id和course\_id为多对一关系，是二者的中间表。

**s.select\_course表**



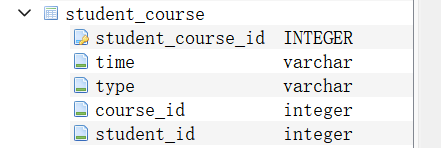
学生和课程的中间表。实现学生和课程的多对多关系。

**t.student表**



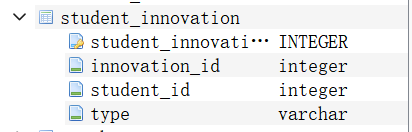
学生类，保存专业、班级、父母名字和电话、家乡、肄业高中等基本信息。主键为student\_id，与外键 person\_id为一对一关系。

**u.student\_course表**



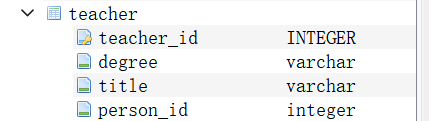
student和course的中间表，主要起连接作用。保存时间和类型信息，对外键course\_id 和student\_id为多对一关系。

**v.student\_innovation表**



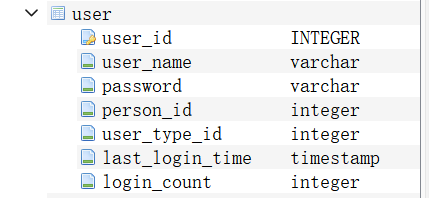
student和innovation的中间表，起连接作用。保存类型信息，对外键student\_id和innovation\_id为多对一关系。

**w.teacher表**



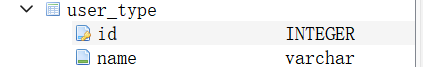
保存教师信息，包括职称和学位（因为基本信息已在person中保存）。主键为teacher\_id ,与外键person\_id为一对一关系。

**x.user表**



保存的是用户账号信息，包括用户名、密码、登录时间、登录次数的信息。主键为user\_id，一个person对应一个user，所以对外键person\_id为一对一关系；几个user对应同一用户类型，所以对外键user\_type\_id为多对一关系。

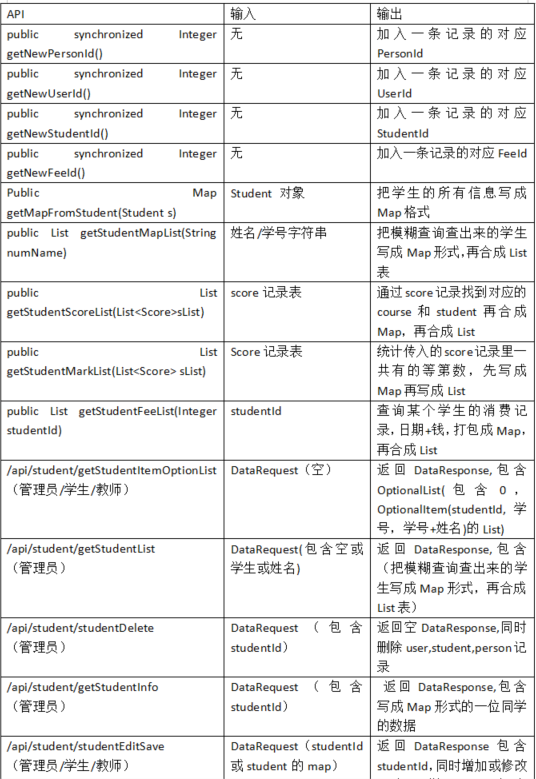
**y.user\_type表**



保存三种用户类型，主键为id.

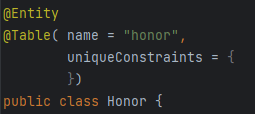
**第四章 系统实现**

首先根据考核要求，我们在分析完项目需求的基础上对功能及操作流程进行了设计，然后在此基础上编写了前后端接口对应的表格，然后我们进行了前后端分离制作，负责前端的同学通过SceneBuilder进行页面的设计，负责后端的同学在明确需求后设计表间关系（一对一，一对多，多对多），编写model中的实体类建表，再根据我们约定的表格编写repository和后端Controller实现对应的API，最后在前后端联调中进行前端Controller的编写和修改。



**1、实体类基本逻辑**

通过**Java Spring JPA**开发框架对数据库表的自动映射，我们可以通过编写model中的实体类实现直接数据库的自动建表。

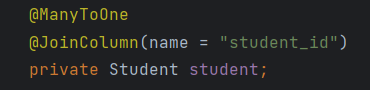


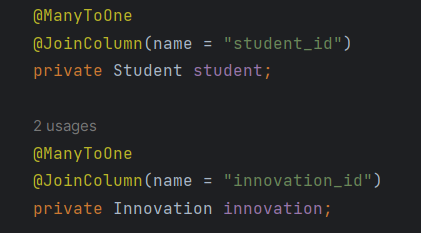
@Entity: 这个注解指示Honor类是一个**JPA实体类**，用于持久化到数据库中。

@Table(name = "honor", uniqueConstraints = {}): 这个注解用于指定实体类对应的数据库表的相关信息。name = "honor"：指定数据库表的名称为honor。

uniqueConstraints = {}：指定表的唯一约束，此处为空。







@Id：这个注解指示courseId属性是实体类的主键。

@GeneratedValue(strategy = GenerationType.IDENTITY)：这个注解指示courseId属性的值是自动生成的，根据底层数据库的自增策略生成唯一标识符。

private Integer courseId：课程的唯一标识符，作为主键。

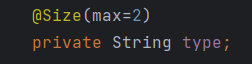
@OneToOne：这个注解表示课程与教师之间的关联关系是一对一关系。在我们的设计中，相同课程名，不同教师的课被设定为使用不同的课序号，因此课序号唯一标识课程，课程与教师一一对应。

@JoinColumn(name="teacher\_id")：这个注解指定了与教师关联的外键列名为teacher\_id。

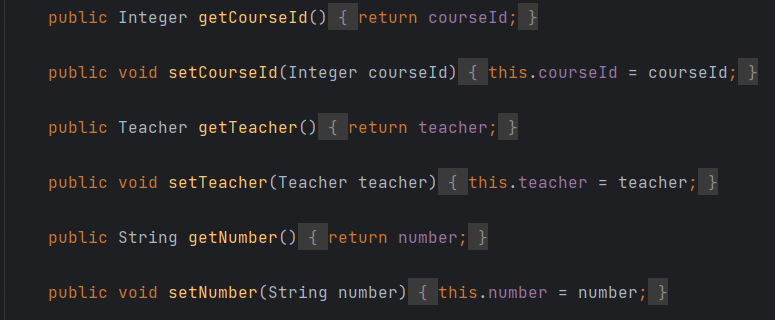
@NotBlank：对于有关绩点计算的学分"credit"关键属性，我们用@NotBlank指示credit属性不能为空。

@Size(max = 50)：这个注解指定了course属性的最大长度为50。

@ManyToOne:这个注解指定了表间的一对多关系，其中多对多关系通过在中间表中设置两个一对多属性来实现。



在Innovation和DailyActivity类中，我们通过设置type属性实现一个表对多种相似信息的管理，比如在Innovation表中，我们通过1-6分别对应社会实践、学科竞赛、科研成果、学科竞赛、培训讲座、创新项目、学科竞赛的管理，避免因为表过多引起程序开发过程混乱。

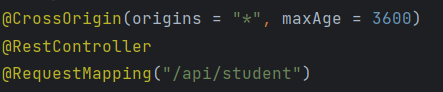


通过java自带的generate工具生成对应的get、set方法，实现对对象属性的管理。

**2、控制逻辑**

大多数Controller，一般由CRUD（增删改查）构成，部分Controller（如StudentController）里面还包括了html转换成pdf的方法，计算综合绩点的方法，以及为个人画像提供数据的后端方法等。

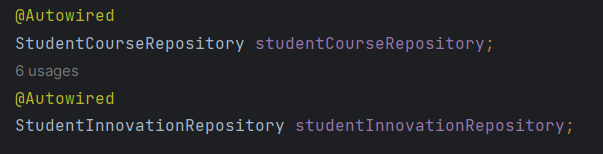
**（1）注解的作用**



@CrossOrigin(origins = "\*", maxAge = 3600)：用于启用跨域资源共享支持。origins = "\*"表示允许来自任何源的请求访问该接口，maxAge = 3600表示在3600秒内不再发送预检请求（OPTIONS请求）。

@RestController：将该类声明为一个RESTful风格的控制器，用于处理HTTP请求。

@RequestMapping("/api/student")：这是一个注解，用于将请求映射到指定的URL路径。在这种情况下，所有以/api/student开头的请求将由该控制器处理。



@Autowired是一个Spring框架的注解，用于自动装配依赖关系Spring会自动在应用上下文中查找匹配类型的实例，并将其注入到相应的变量中。我们最初对这种分离操作的目的感到疑惑，在对Springboot框架的学习中，我认识到这样由框架管理对象的方式可以降低代码的耦合度，提高代码的可维护性、可测试性和可扩展性。

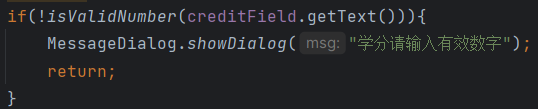
**（2）增删查改功能的实现**

当前端选中一条学生记录后，点击对应的操作按钮，就会跳出单个学生信息所在的页面，数据表所在的Controller将通过监听获取当前选中学生的主键，并向弹出界面所在的Controller传入主键与用command属性标记的操作类型（1表示增加，2表示查看，3表示修改），当用户点击确认按钮后，弹出界面的controller会将根据command类型将该页面的数据传给对应后端方法，实现数据操作。

**a.增加与修改功能**



增加与删除功能在后端对应同一个URL请求，如果前端传过来的主键id为空，则调用getNewStudentId()方法获得对应表的最大id，然后直接在表中添加新的记录。如果主键id不为空，则在表中查询对应的记录，如果查询成功则将该学生数据覆盖，并向前端传递正确信息，查询失败则向前端传送错误信息，由前端的对应方法完成返回信息的页面展示。

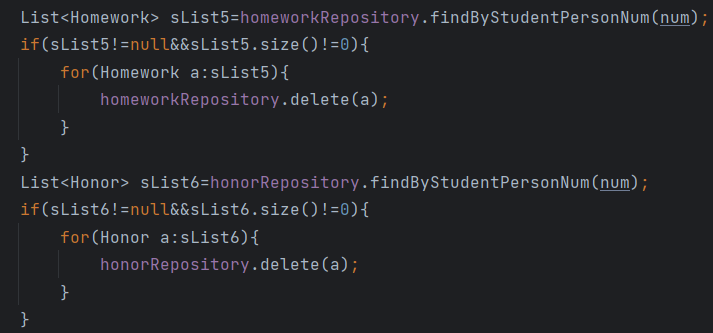
、

对于关键数据（如与**绩点计算**有关的学分），程序对输入数据进行了检查，若数据不符合要求，则会提示数据类型错误，不能成功保存，提高程序的健壮性。

**b.删除功能**



在执行删除操作时，前端会发出弹窗提醒，如果用户点击确认则执行删除操作，在后端方法中先根据主键查找对应学生，再由**repository**执行删除操作。



在后端的StudentController中，执行学生删除操作时对应删除了其他学生信息表（如Honor表）中关于此学生的所有信息，在一对多关系中，如Teacher，Course表，均实现了这样的级联删除，这样避免了数据库中脏数据的出现。

IMG_256

在数据的传输过程中，我们使用**正则表达式**对数据进行处理，使得数据符合我们数据储存的规范。

//从第四章到这里介绍了一些注解以及增删的代码实现，应该可以之间拿来用，没删

**c.学生照片的上传//从这到末尾我只删掉了学长的课设界面截图，各种功能的代码截图我没删，框架也没动，不知道哪些功能是咱有的哪些咱没有的。。。**





在学生信息修改界面设置了拖动事件的监听，只需要拖动照片到照片显示的窗口上并松开，便会自动实现照片的上传，并将数据传给后端，在photo文件夹里用学生的学号为名唯一标识。

**d.显示与查询功能**





IMG_256

当查询内容传到后端之后，后端的getStudentListByQuery方法调用getHonorMapListByQuery方法（调用honorRepository进行模糊查询），并进一步调用getMapFromStudent将查出来的数据打包成Map的List形式，方便其以json格式在前后端之间传输。



查询框设置了**回车监听**，可以通过回车键方便地进行一键查询。要查询所有信息，则只要保持查询框为空即可。

**e.查询记录数的统计**



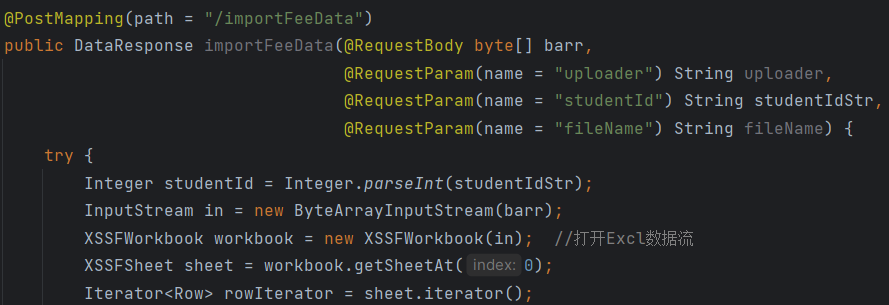
此方法一方面实现了给表中数据的自动标号，另一方面在标号的同时更新当前表中数据记录数的最大值，将查询记录赋值为这个最大值。



我们建立了InformationService类用来记录前端当前的状态，比如当前登录用户的用户名，进入课程中心后对应的该课程主键，以及每个表的大小，在上面方法的赋值结束后，将根据这个最大值判断是否要更新InformationService类中的数据。

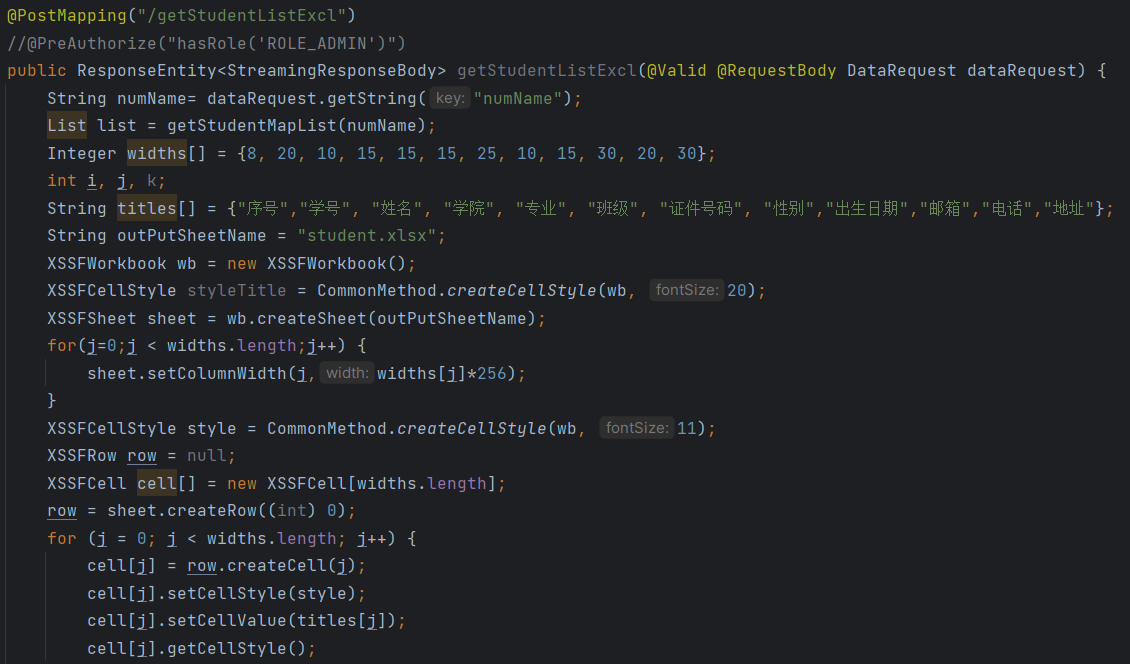
**f.学生消费记录的EXCEL表上传**





方法主体部分是对Excel文件的解析和数据处理逻辑。它使用Apache POI库来读取Excel文件（XSSFWorkbook），然后迭代遍历每一行的数据。每一行包含消费记录的日期、金额、地点和商品信息。代码将这些数据提取出来，并创建一个Fee对象，设置对应的属性值，然后将其保存到数据库中。

**g.学生、教师信息的EXCEL表下载**



遍历学生列表数据，创建数据行，并将学生信息填充到对应的单元格中。最终，将生成的Excel文件通过响应流返回给客户端。

**h.最近修改记录的查看功能**



在每次修改操作被确认后，我们会再向后端增加一条修改记录，里面包含了修改人，修改类型，修改时间等信息，管理员可以通过点击最近修改按钮，查询最近数据的编辑信息，提高了系统的安全性和实用性。

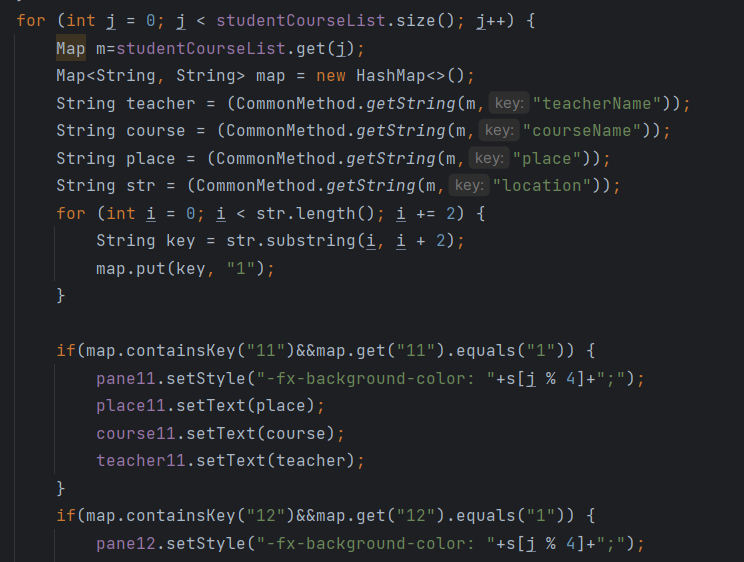
**i.学生请假功能的实现**



学生可以在学生端填写请假表单，记录在申请表中，管理员可以对申请表进行通过或者是不通过审批，但是不能更改申请的数据，一旦学生的请假信息被管理员批准，学生的situation属性将被赋值为“已通过”。这一条请假申请信息便会自动变为请假信息，汇入学生请假信息管理表中。学生可以在学生界面得知自己的请假是否成功。

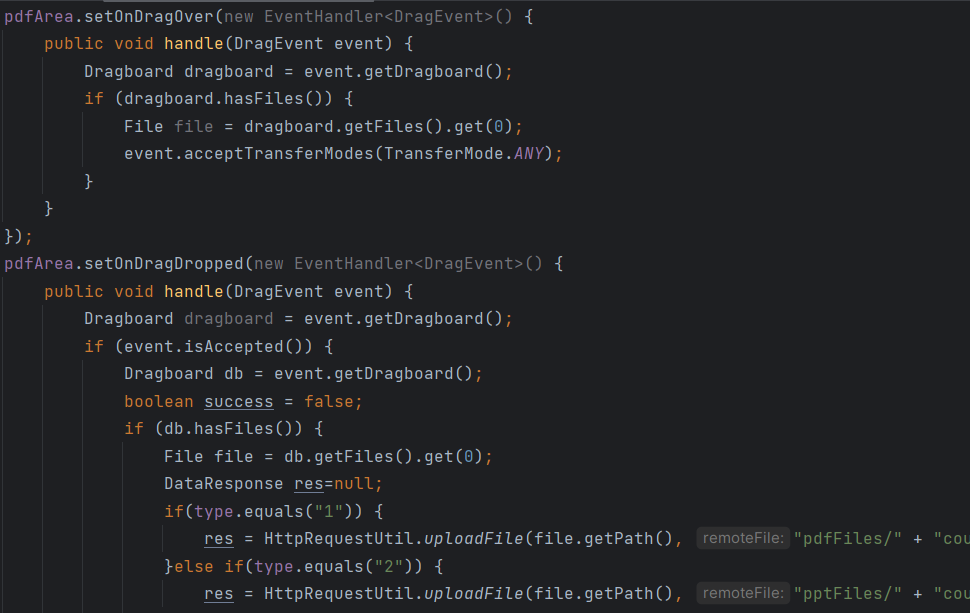
**（3）课程中心相关功能的实现**

**a.课表功能**



在课程添加界面可以更改这门课的上课时间，我们用一个数字串来保存一门课的上课时间（如周一第二节课和周三第五节课将被保存为location=1235），保存在course表的location属性中，前端的课表将对这个字符串进行解码，将对应的位置标上课名和上课时间地点。同时通过表示这周第几节课的变量j对4取模，来决定这门课被标记为颜色数组中的哪一种颜色，使得课表更有层次感。如s[0]为天蓝色。

**b.课件管理**



课件的上传与图片的上传下载类似，一共支持（pdf,pptx,docx）三种格式的课件，保存在用课序号命名的文件夹中，学生下载课件时只需要遍历这个文件夹就可下载该门课的全部课件。

**c.选课系统**



管理员可以在开放选课界面勾选想要开放的课程列表，一旦开放，课程表中，该课的choseable属性将从0被修改成1，学生端便可以看到这门课并开始选课，如果某门课的选课人数大于课容量，这门课的状态（situation）属性将被修改为待抽签，在管理员端选中该门课并点击抽签按钮后，选课表中的记录将被修改为（选中/未选中）。选课的实现完全随机，如果课容量为10，选课人数为20，系统将会把选课记录用Collections.shuffle()方法打乱，取前10条记录改为选中，剩下的改为未选中。未选中和待抽签的课不会进入学生的课程中心。

**d.图表的制作**



成绩和消费记录显示采用了图表，以成绩等第的饼状图为例，后端先查询学生的成绩表记录，根据学生的期末成绩给出成绩等地的个数统计,分为title和value分别打包成List传给前端，前端将此数据add给pieChart进行画图。

**e.出题功能-作业布置-错题本功能**



**在教师端和管理员端，**每个教师都有属于自己的**题库**，包含增删查改功能，教师可以选中题目记录，输入作业标题，点击生成作业按钮，系统会在前端根据题目类型对题目进行分类，生成作业对应的html字符串，传给后端，再生成作业的pdf文件（和学生简历共用pdf的下载功能），如果点击布置作业的按钮，该pdf文件将用日期和课程主键组合命名，传给后端，保存在后端的homework文件夹中。在这种命名格式下，学生端可以通过日期选择器下载对应日期和科目的作业。**在学生端，**这个功能被更改为**错题本功能**，学生也可以拥有自己的题库，生成错题卷，但是不能布置作业。

**f.顺次点名，批改作业，录入成绩**



为了让老师依次录入自己课上所有学生的作业，成绩，点名信息，我们设计了rollCall方法，该方法遍历老师的所有学生，附带计次功能，一键完成所有学生的信息录入，避免缺漏。

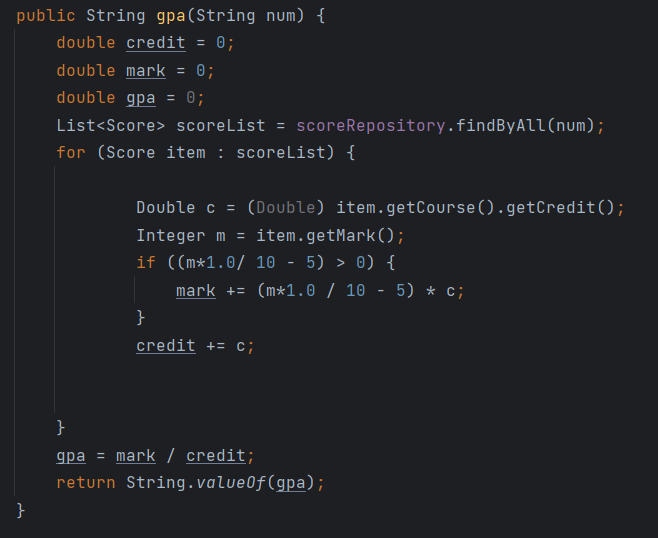
**g.自动排名**



当教师录入完学生成绩信息后，调用这个方法，使用冒泡排序对课程对应学生成绩的排名。首先，通过调用scoreRepository的findStudentListByNumName方法获取学生列表。然后，使用冒泡排序算法对学生列表进行排序，按照分数从低到高的顺序进行排列。接下来，通过迭代学生列表，为每个学生设置排名。使用一个计数器cnt来跟踪当前的排名，初始值为学生列表的大小。对于每个学生对象，根据其scoreId从scoreRepository中获取相应的学生实体对象，并对他们的ranking进行赋值。

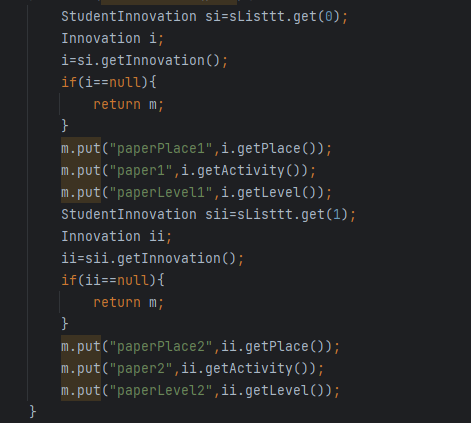
**（4）个人画像与个人简历功能**

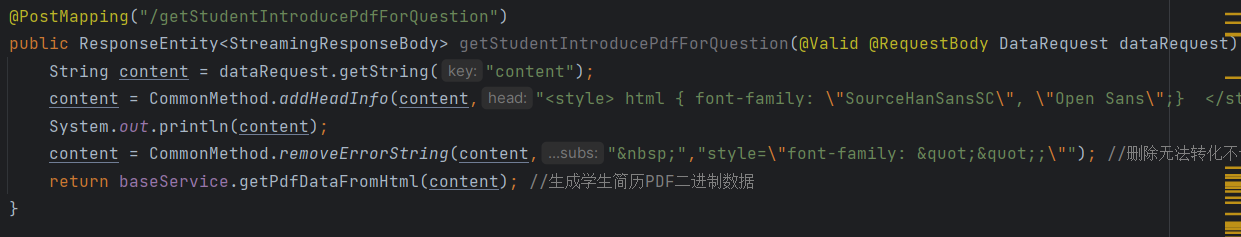
**a.综合绩分的计算**



代码中的参数num表示学生的学号，通过该学号查询数据库中该学生的成绩信息。通过scoreRepository.findByAll(num)从数据库中查询到该学生的成绩列表，并使用一个循环遍历每一门课程的成绩。在循环中，首先获取每门课程的学分c和成绩m。然后，通过判断(m\*1.0/10 - 5) > 0，来确定该门课程是否及格（成绩大于等于60分）。如果及格，则计算加权分数(m\*1.0/10 - 5) \* c，并将其累加到mark变量中。同时，将每门课程的学分c累加到credit变量中。循环结束后，通过mark / credit计算绩点，并将其赋值给变量Gpa。最后，将Gpa转换为字符串类型，并返回作为函数的结果。

**b.学生个人简历的生成和下载**







student的每一条记录中保存有个人简历的html模板，模板中用如${name}充当占位符，在生成pdf文件的过程中，StudentController里的方法通过各种repository查询各个表中的记录并打包成Map，其中Map的key与模板中的名字保持一致，从而达到占位符的替换。再调用PDFBox库将html字符串转为pdf文件。

**主要难点：**

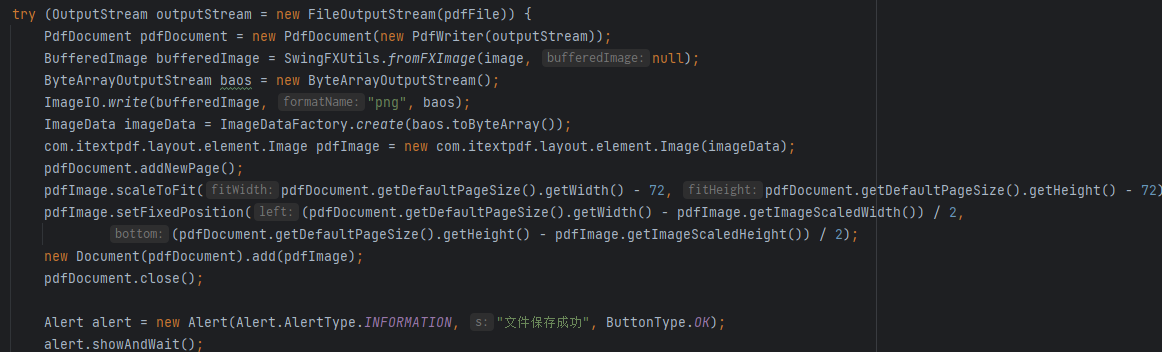
1）在这次课设中我们使用了PDFBox库进行html文件的转换，但是我们前几次尝试均不成功，通过查阅资料得知此方法对html的代码格式要求相当高以及不支持我们使用的许多css代码。比如标签必须闭合，最后我们严格按照html语法规则编写闭合标签的代码解决了这一问题。

2）另一个问题是我们的<img/>标签在转换时一直无法显示相应的图片，后来我们将图片url的路径改成从磁盘开始的绝对路径，最终转换成功。

**c.学生个人画像的生成和下载**



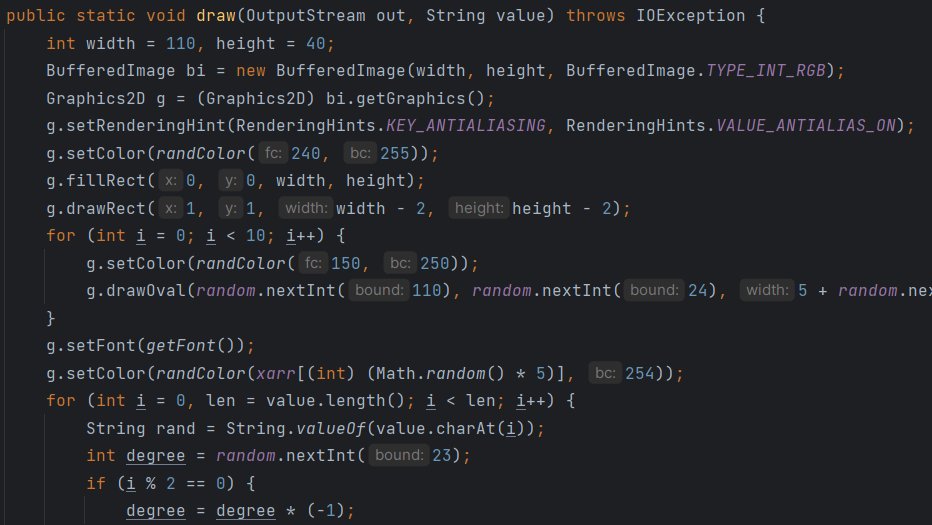
个人画像的数据与个人简历的数据共用一个数据查询方法，不同的是个人简历是一个html文件，而个人画像为了实现画图功能以及更加多彩的页面展示采用fxml格式。个人画像的图表与学生端的成绩可视化图表相同。



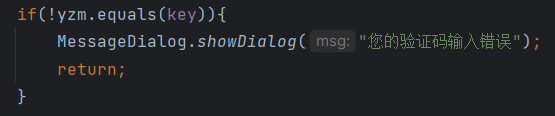
为了实现个人画像的保存，我们用这个方法将JavaFX界面截图，并将截图保存为PDF文件。方法中的代码逻辑包括将界面截图转换为WritableImage对象，然后使用FileChooser让用户选择保存PDF文件的路径和文件名。如果用户选择了有效的文件路径，就将截图转换为BufferedImage对象，并使用iText库将图像添加到PDF文档中。最后，将PDF文档保存到用户选择的文件路径，并显示相应的保存结果提示对话框。如果保存过程中发生异常，将显示保存出错的错误消息提示对话框。

**（5）验证码功能和密码修改功能**

**a.验证码功能**



QQ截图20230520141647



创建一个BufferedImage对象，用于表示验证码图片，指定了图片的宽度和高度。获取该图片的Graphics2D对象，用于进行绘图操作，并设置了抗锯齿渲染的提示。绘制背景，使用随机生成的颜色填充整个图片。绘制边框，使用固定的颜色绘制一个矩形边框。随机生成若干个椭圆，并使用随机的颜色绘制这些椭圆，增加验证码图片的干扰效果。设置绘制文本的字体，并随机选择颜色和位置，旋转绘制每个字符，形成扭曲的效果。绘制干扰线，使用固定的颜色绘制一条曲线增加验证码图片的干扰效果。完成绘图后，释放绘图资源。最后将生成的验证码图片保存到本地。如果用户输入的验证码与key值不符，则会弹出窗口显示“验证码错误”，不能成功登录。

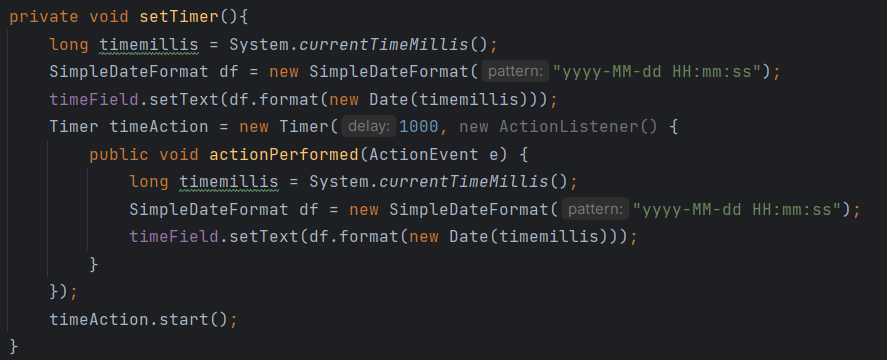
**b.密码修改功能**



密码修改是对user表的修改操作，若两次输入密码不一致或为空，则不能成功修改。

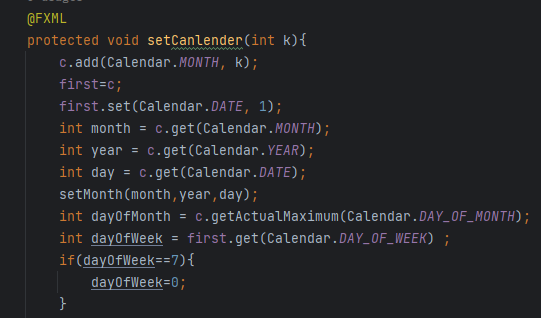
**（6）时间功能**

**a.系统时间的显示**



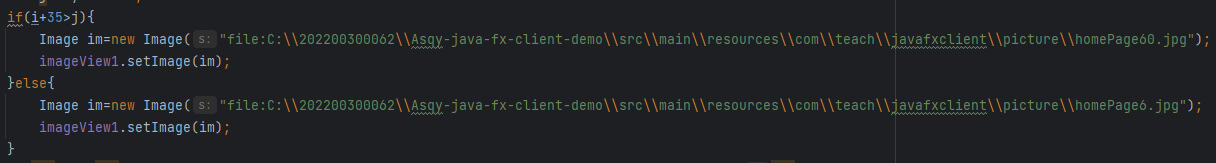
这个方法获取当前的时间戳，并使用SimpleDateFormat格式化为指定的日期时间格式。将格式化后的时间设置到文本字段中，用于显示当前时间。创建一个定时器对象，每隔1秒触发一次动作。在定时器的动作中，获取当前的时间戳，再次使用SimpleDateFormat格式化为日期时间字符串，并更新到文本字段中。启动定时器，开始定时更新时间字段的显示。通过在initialize（）方法中添加setTimer()我们实现了系统时间的显示以及每隔一秒刷新一次。

**b.日历系统**



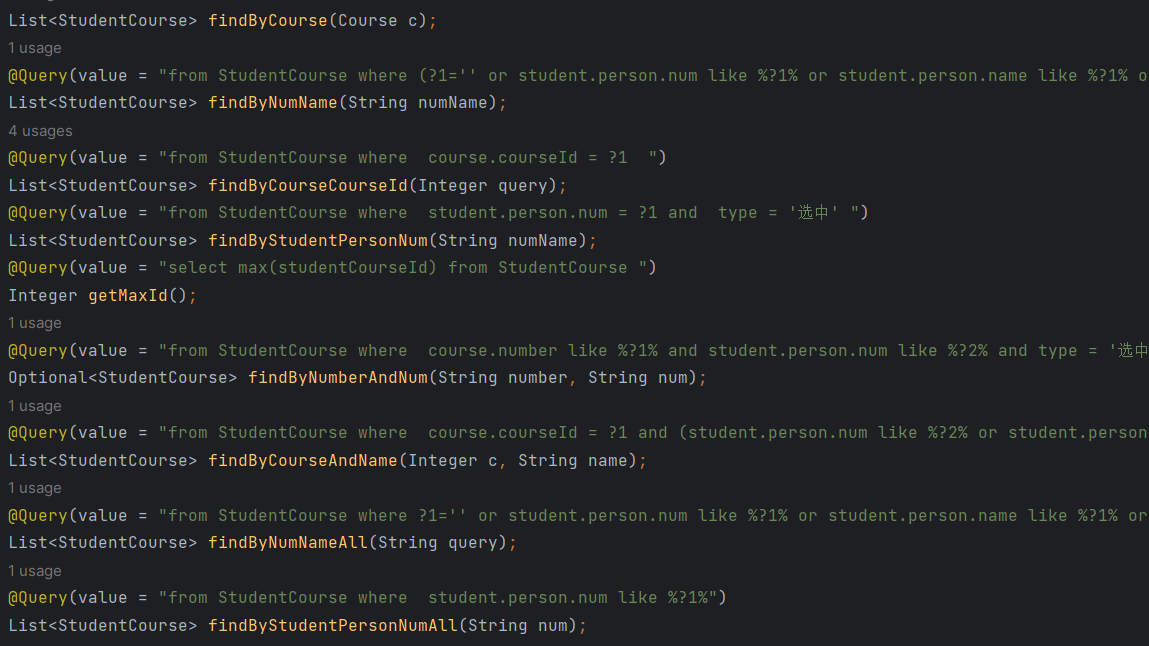
**主要难点：**闰年2月日期设置，日历行数变化

学生主界面的日历支持动态查询，用户可以通过点击页面最上方的标签来切换月份，在更改时间时调用javaCalender类，生成对应的时间而不是直接调用数组数据，这样可以避免例如2.29日期错误的问题。



一个月的日历有时候是五行，有时候是六行。我们通过对不同情况的判断，更改不同样式的背景图片，使得页面更加整齐美观。

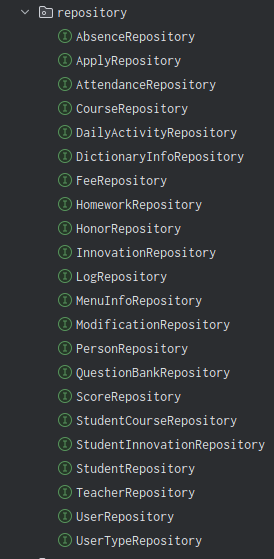
**3、数据表维护查询**



这是一个用于与数据库进行交互的接口类，名为StudentCourseRepository。它被标注为@Repository，表示该接口用于持久化层的操作，将其标识为Spring框架的Bean。该接口继承自JpaRepository<StudentCourse, Integer>，这是Spring Data JPA提供的一个通用接口，用于简化数据库的CRUD操作。JpaRepository接口提供了许多常用的数据库操作方法，如保存、删除、查询等。

JPA可以通过驼峰命名规范直接实现查询，比如findByCourse(Course c)表示根据课程查询，findByPersonNum（String num）表示根据学生学号查询。

JPA还可以通过@Query注解用JPQL语句更加灵活地设计查询规则，包括模糊查询，如student.person.num like %?2%表示根据传入的第二个参数实现对学生学号的模糊查询。



AbsenceRepository ->学生请假的数据库查询

ApplyRepository->学生申请的数据库查询

AttendanceRepository->学生出勤的数据库查询

CourseRepository->课程的数据库查询

DailyActivityRepository->日常活动的数据库查询

FeeRepository->消费数据的数据库查询

HomeworkRepository->学生作业的数据库查询

HonorRepository->学生荣誉信息的数据库查询

InnovationRepository->创新实践信息的数据库查询

LogRepository->学生日志的数据库查询

ModificationRepository->修改记录的数据库查询

QuestionRepository->题库的数据库查询

ScoreRepository->学生成绩的数据库查询

StudentCourseRepository->学生课程的数据库查询

StudentInnovationRepository->学生创新信息的数据库查询

StudentRepository->学生信息的数据库查询

TeacherRepository->教师信息的数据库查询

UserRepository->用户的数据库查询

UserTypeRepository->用户类型的数据库查询