**软件工程课程项目**

**——实验教学管理系统**

软件设计规格说明书

**Software DESIGNS Specification**

**小组成员：2051196刘一飞**

**2052348王杨乐**

**2050747赵帅涛**

**指导教师：黄杰**

目 录

[1 数据设计 6](#_Toc123334066)

[**1.1 数据库设计** 6](#_Toc123334067)

[**1.1.1 逻辑设计** 6](#_Toc123334068)

[**1.1.2 物理设计** 6](#_Toc123334069)

[**1.2 类设计** 7](#_Toc123334070)

[**1.2.1 用户登录子系统** 7](#_Toc123334071)

[**1.2.1.1 系统简述** 7](#_Toc123334072)

[**1.2.1.2 设计类图** 7](#_Toc123334073)

[**1.2.2 实验课程管理子系统** 7](#_Toc123334074)

[**1.2.2.1 系统简述** 7](#_Toc123334075)

[**1.2.2.2 设计类图** 8](#_Toc123334076)

[**1.2.3 实验项目管理子系统** 8](#_Toc123334077)

[**1.2.3.1 系统简述** 8](#_Toc123334078)

[**1.2.3.2 设计类图** 8](#_Toc123334079)

[**1.2.4 实验报告管理子系统** 9](#_Toc123334080)

[**1.2.4.1 系统简述** 9](#_Toc123334081)

[**1.2.4.2 设计类图** 9](#_Toc123334082)

[**1.2.5 课程资源管理子系统** 9](#_Toc123334083)

[**1.2.5.1 系统简述** 9](#_Toc123334084)

[**1.2.5.2 设计类图** 9](#_Toc123334085)

[**1.2.6 成绩管理子系统** 10](#_Toc123334086)

[**1.2.6.1 系统简述** 10](#_Toc123334087)

[**1.2.6.2 设计类图** 10](#_Toc123334088)

[**1.2.7 通知公告子系统** 10](#_Toc123334089)

[**1.2.7.1 系统简述** 11](#_Toc123334090)

[**1.2.7.2 设计类图** 11](#_Toc123334091)

[**1.2.8 个人信息子系统** 11](#_Toc123334092)

[**1.2.8.1 系统简述** 11](#_Toc123334093)

[**1.2.8.2 设计类图** 11](#_Toc123334094)

[**1.2.9 系统管理子系统** 12](#_Toc123334095)

[**1.2.9.1 系统简述** 12](#_Toc123334096)

[**1.2.9.2 设计类图** 12](#_Toc123334097)

[2 架构设计 12](#_Toc123334098)

[**2.1 总体架构** 12](#_Toc123334099)

[**2.1.1 前端架构设计** 12](#_Toc123334100)

[**2.1.2 后端架构设计** 13](#_Toc123334101)

[**2.2 微服务架构** 13](#_Toc123334102)

[**2.3 架构上下文** 13](#_Toc123334103)

[**2.4 系统内部架构** 14](#_Toc123334104)

[3 接口设计 14](#_Toc123334105)

[**3.1 图形用户界面** 14](#_Toc123334106)

[**3.1.1 登录界面** 15](#_Toc123334107)

[**3.1.2 学生导航界面** 15](#_Toc123334108)

[**3.1.3 教师导航界面** 15](#_Toc123334109)

[**3.1.4 管理员导航界面** 16](#_Toc123334110)

[**3.1.5 实验课程界面** 16](#_Toc123334111)

[**3.1.6 实验项目界面** 17](#_Toc123334112)

[**3.1.7 个人信息界面** 17](#_Toc123334113)

[**3.2 内部接口** 18](#_Toc123334114)

[**3.2.1 用户登录子系统** 18](#_Toc123334115)

[**3.2.1.1 用户账户激活** 18](#_Toc123334116)

[**3.2.1.2 用户登录** 19](#_Toc123334117)

[**3.2.1.3 忘记密码** 20](#_Toc123334118)

[**3.2.2 课程管理子系统** 22](#_Toc123334119)

[**3.2.2.1 用户获取课程列表** 22](#_Toc123334120)

[**3.2.2.2 责任教师添加课程** 23](#_Toc123334121)

[**3.2.2.3 责任教师修改课程** 24](#_Toc123334122)

[**3.2.2.4 责任教师删除课程** 25](#_Toc123334123)

[**3.2.3 实验管理子系统** 26](#_Toc123334124)

[**3.2.3.1 用户获取实验列表** 26](#_Toc123334125)

[**3.2.3.2 教师添加实验项目** 28](#_Toc123334126)

[**3.2.3.3 教师修改实验项目** 29](#_Toc123334127)

[**3.2.3.4 教师删除实验项目** 31](#_Toc123334128)

[**3.2.4 实验报告子系统** 32](#_Toc123334129)

[**3.2.4.1 教师上传实验模板** 32](#_Toc123334130)

[**3.2.4.2 学生获取实验模板** 33](#_Toc123334131)

[**3.2.4.3 学生提交实验报告** 34](#_Toc123334132)

[**3.2.4.4 教师/助教批阅实验报告** 35](#_Toc123334133)

[**3.2.5 课程文件子系统** 36](#_Toc123334134)

[**3.2.5.1 获取该课程的所有相关文件** 36](#_Toc123334135)

[**3.2.5.2 教师上传文件** 37](#_Toc123334136)

[**3.2.5.3 用户下载文件** 38](#_Toc123334137)

[**3.2.6 学生成绩子系统** 38](#_Toc123334138)

[**3.2.6.1 教师提交成绩** 38](#_Toc123334139)

[**3.2.6.2 学生查看成绩** 39](#_Toc123334140)

[**3.2.7 通知公告子系统** 40](#_Toc123334141)

[**3.2.7.1 获取公告列表** 40](#_Toc123334142)

[**3.2.7.2 新增公告信息** 41](#_Toc123334143)

[**3.2.7.3 更新公告状态** 42](#_Toc123334144)

[**3.2.8 个人信息子系统** 43](#_Toc123334145)

[**3.2.8.1 个人信息获取** 43](#_Toc123334146)

[**3.2.8.2 修改个人信息** 44](#_Toc123334147)

[**3.2.9 管理员子系统** 45](#_Toc123334148)

[**3.2.9.1 录入账号列表** 46](#_Toc123334149)

[**3.2.9.2 获取账号列表** 47](#_Toc123334150)

[**3.2.9.3 更新账号信息** 48](#_Toc123334151)

[**3.3 外部接口** 49](#_Toc123334152)

[**3.3.1 JavaMail邮件发送** 49](#_Toc123334153)

[4 构件设计 49](#_Toc123334154)

[**4.1 用户管理构件** 49](#_Toc123334155)

[**4.1.1 构件图** 49](#_Toc123334156)

[**4.1.2 核心操作** 50](#_Toc123334157)

[**4.1.2.1 login** 50](#_Toc123334158)

[**4.1.2.1.1 操作简述** 50](#_Toc123334159)

[**4.1.2.1.2 时序图** 50](#_Toc123334160)

[**4.1.2.2 register** 50](#_Toc123334161)

[**4.1.2.2.1 操作简述** 50](#_Toc123334162)

[**4.1.2.2.2 时序图** 51](#_Toc123334163)

[**4.2 课程管理构件** 51](#_Toc123334164)

[**4.2.1 构件图** 51](#_Toc123334165)

[**4.2.2 核心操作** 51](#_Toc123334166)

[**4.2.2.1 getCourseInfo** 51](#_Toc123334167)

[**4.2.2.1.1 操作简述** 52](#_Toc123334168)

[**4.2.2.1.2 时序图** 52](#_Toc123334169)

[**4.2.2.2 addNewCourse** 52](#_Toc123334170)

[**4.2.2.2.1 操作简述** 52](#_Toc123334171)

[**4.2.2.2.2 时序图** 52](#_Toc123334172)

[**4.2.2.3 modifyCourseInfo** 53](#_Toc123334173)

[**4.2.2.3.1 操作简述** 53](#_Toc123334174)

[**4.2.2.3.2 时序图** 53](#_Toc123334175)

[**4.2.2.4 deleteAbandonCourse** 53](#_Toc123334176)

[**4.2.2.4.1 操作简述** 53](#_Toc123334177)

[**4.2.2.4.2 时序图** 53](#_Toc123334178)

[**4.3 实验管理构件** 54](#_Toc123334179)

[**4.3.1 构件图** 54](#_Toc123334180)

[**4.3.2 核心操作** 54](#_Toc123334181)

[**4.3.2.1 getExperimentInfo** 54](#_Toc123334182)

[**4.3.2.1.1 操作简述** 54](#_Toc123334183)

[**4.3.2.1.2 时序图** 54](#_Toc123334184)

[**4.3.2.2 addNewExperiment** 55](#_Toc123334185)

[**4.3.2.2.1 操作简述** 55](#_Toc123334186)

[**4.3.2.2.2 时序图** 55](#_Toc123334187)

[**4.3.2.3 modifyExperiment** 55](#_Toc123334188)

[**4.3.2.3.1 操作简述** 55](#_Toc123334189)

[**4.3.2.3.2 时序图** 56](#_Toc123334190)

[**4.3.2.4 deleteAbandonExperiment** 56](#_Toc123334191)

[**4.3.2.4.1 操作简述** 56](#_Toc123334192)

[**4.3.2.4.2 时序图** 56](#_Toc123334193)

[**4.4 实验报告构件** 56](#_Toc123334194)

[**4.4.1 构件图** 57](#_Toc123334195)

[**4.4.2 核心操作** 57](#_Toc123334196)

[**4.4.2.1 getList** 57](#_Toc123334197)

[**4.4.2.1.1 操作说明** 57](#_Toc123334198)

[**4.4.2.1.2 时序图** 57](#_Toc123334199)

[**4.4.2.2 postReport** 58](#_Toc123334200)

[**4.4.2.2.1 操作说明** 58](#_Toc123334201)

[**4.4.2.2.2 时序图** 58](#_Toc123334202)

[**4.5 课程文件构件** 58](#_Toc123334203)

[**4.5.1 构件图** 58](#_Toc123334204)

[**4.5.2 核心操作** 59](#_Toc123334205)

[**4.5.2.1 checkResource** 59](#_Toc123334206)

[**4.5.2.1.1操作简述** 59](#_Toc123334207)

[**4.5.2.1.2时序图** 59](#_Toc123334208)

[**4.5.2.2 uploadResource** 59](#_Toc123334209)

[**4.5.2.2.1操作简述** 59](#_Toc123334210)

[**4.5.2.2.2时序图** 59](#_Toc123334211)

[**4.5.2.3 downloadResource** 60](#_Toc123334212)

[**4.5.2.3.1操作简述** 60](#_Toc123334213)

[**4.5.2.3.2时序图** 60](#_Toc123334214)

[**4.5.2.4 deleteResource** 60](#_Toc123334215)

[**4.5.2.4.1操作简述** 60](#_Toc123334216)

[**4.5.2.4.2时序图** 61](#_Toc123334217)

[**4.6 学生成绩构件** 61](#_Toc123334218)

[**4.6.1 构件图** 61](#_Toc123334219)

[**4.6.2 核心操作** 61](#_Toc123334220)

[**4.6.2.1 stuScore** 61](#_Toc123334221)

[**4.6.2.1.1 操作简述** 61](#_Toc123334222)

[**4.6.2.1.2 时序图** 62](#_Toc123334223)

[**4.6.2.2 courseScore** 62](#_Toc123334224)

[**4.6.2.2.1 操作描述** 62](#_Toc123334225)

[**4.6.2.2.2 时序图** 62](#_Toc123334226)

[**4.7 通知公告构件** 63](#_Toc123334227)

[**4.7.1 构件图** 63](#_Toc123334228)

[**4.7.2 核心操作** 63](#_Toc123334229)

[**4.7.2.1 noticeList** 63](#_Toc123334230)

[**4.7.2.1.1 操作简述** 63](#_Toc123334231)

[**4.7.2.1.2 时序图** 64](#_Toc123334232)

[**4.7.2.2 publishNotice** 64](#_Toc123334233)

[**4.7.2.2.1 操作简述** 64](#_Toc123334234)

[**4.7.2.2.2 时序图** 64](#_Toc123334235)

[**4.7.2.3 deleteNotice** 65](#_Toc123334236)

[**4.7.2.3.1 操作简述** 65](#_Toc123334237)

[**4.7.2.3.2 时序图** 65](#_Toc123334238)

[**4.8 个人信息构件** 65](#_Toc123334239)

[**4.8.1 构件图** 65](#_Toc123334240)

[**4.8.2 核心操作** 66](#_Toc123334241)

[**4.8.2.1 checkInfo** 66](#_Toc123334242)

[**4.8.2.1.1操作简述** 66](#_Toc123334243)

[**4.8.2.1.2时序图** 66](#_Toc123334244)

[**4.8.2.2 modifyInfo** 66](#_Toc123334245)

[**4.8.2.2.1操作简述** 66](#_Toc123334246)

[**4.8.2.2.2时序图** 66](#_Toc123334247)

[**4.9 管理员构件** 67](#_Toc123334248)

[**4.9.1 构件图** 67](#_Toc123334249)

[**4.9.2 核心操作** 67](#_Toc123334250)

[**4.9.2.1 addUsers** 67](#_Toc123334251)

[**4.9.2.1.1操作简述** 67](#_Toc123334252)

[**4.9.2.1.2时序图** 68](#_Toc123334253)

[**4.9.2.2 modifyUser** 68](#_Toc123334254)

[**4.9.2.2.1操作简述** 68](#_Toc123334255)

[**4.9.2.2.2时序图** 68](#_Toc123334256)

[**4.9.2.3 deleteUsers** 68](#_Toc123334257)

[**4.9.2.3.1操作简述** 68](#_Toc123334258)

[**4.9.2.3.2时序图** 69](#_Toc123334259)

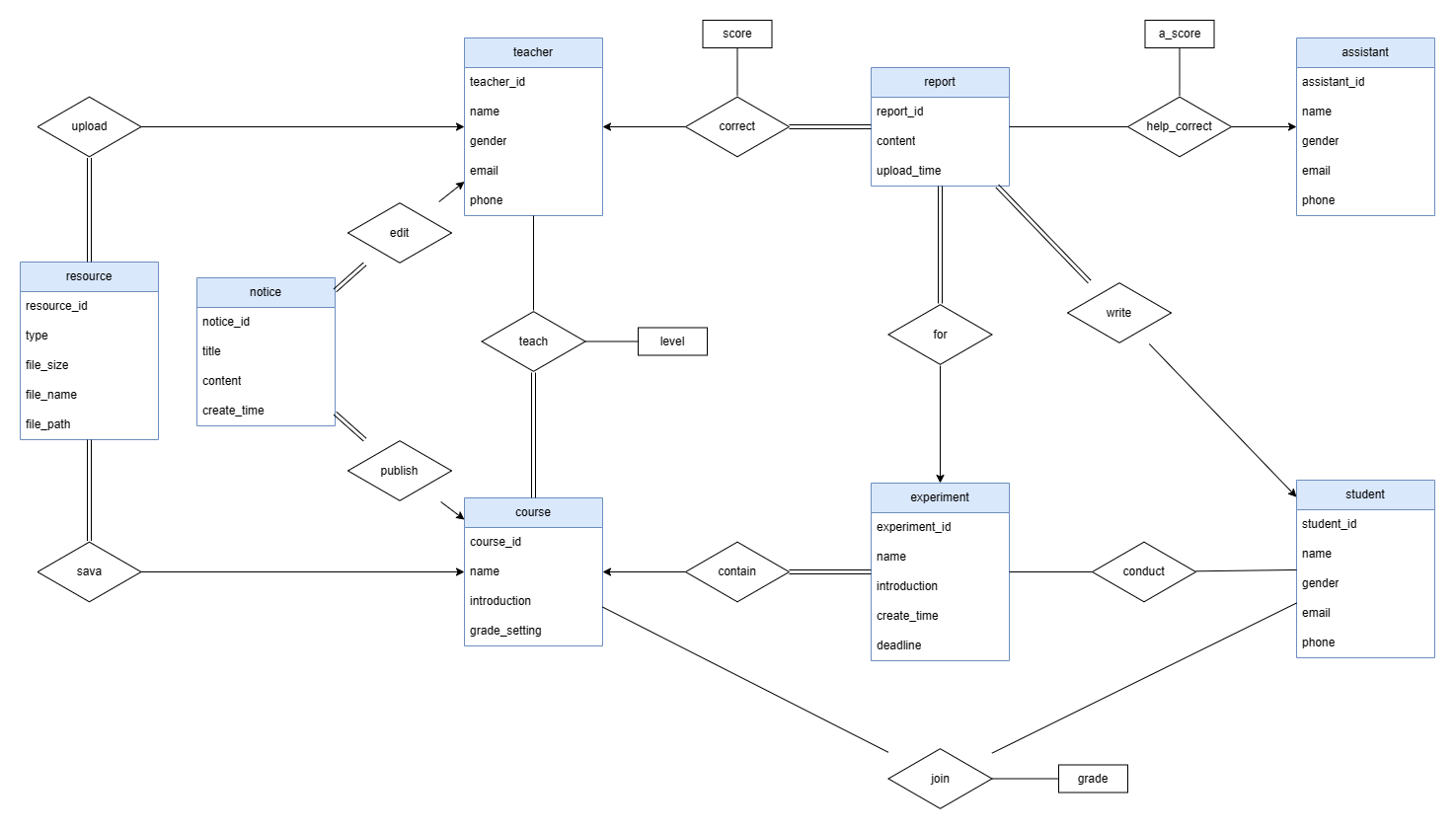
[5 项目代码 69](#_Toc123334260)

**1 数据设计**

**1.1 数据库设计**

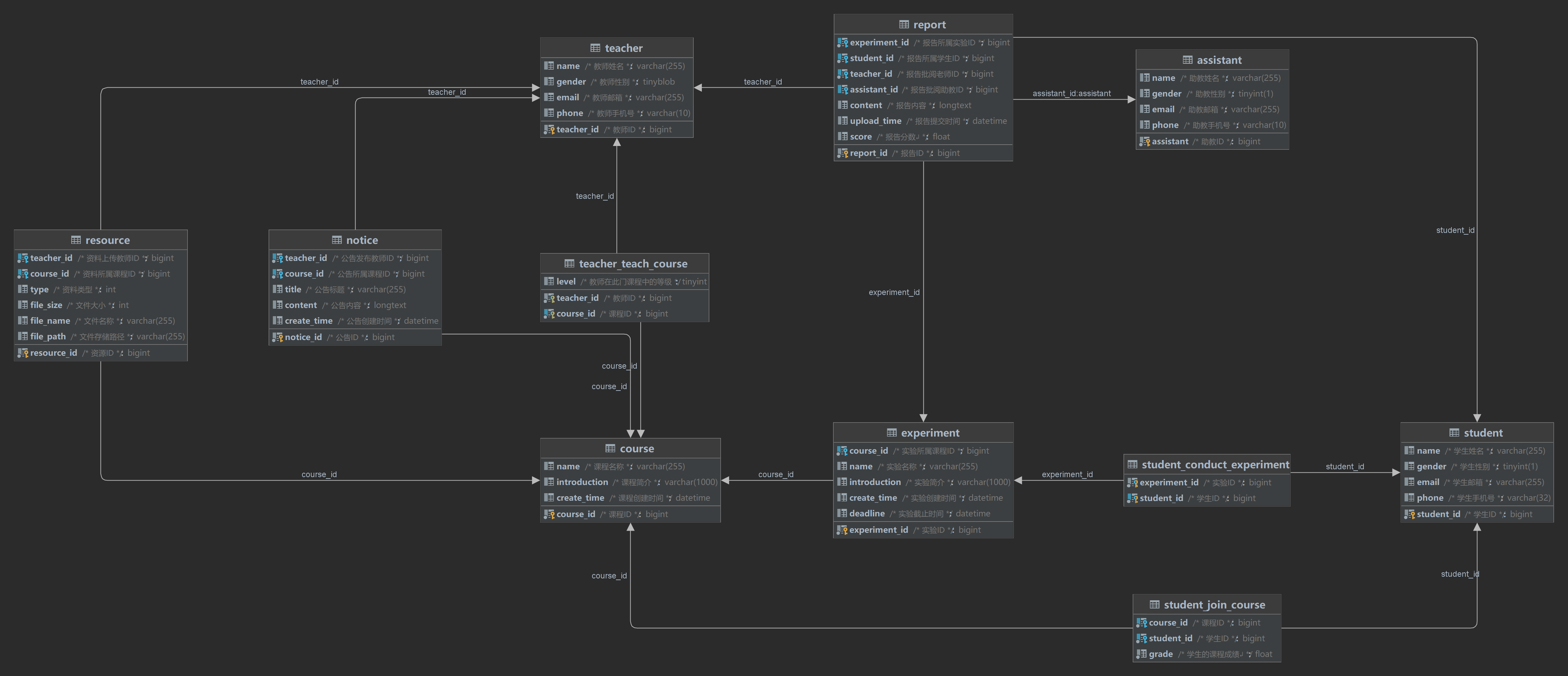
**1.1.1 逻辑设计**

针对支撑实验教学管理系统所需的数据模型，我们利用语法解析识别系统所涉及的实体，并根据其是否具有Retained information，Needed Services，Multiple Attribute等特征进行过滤，识别出系统所需数据实体，如教师，学生，实验报告，通知公告等，并根据具体的业务逻辑需求，设计出如下的ER图：



**1.1.2 物理设计**

在完成ER图的设计之后，我们根据ER图与数据库的关系，根据实体间一对一，多对一等关系，合理设计数据库表，通过增加数据库表或在表中增添字段，将各表之间的关系联系起来，为满足较高的性能需求，我们力求用最小的存储代价描述数据模型，为此数据库的设计严格遵循了第三范式，消除非主属性对于码的传递函数依赖，具体数据库表如下：



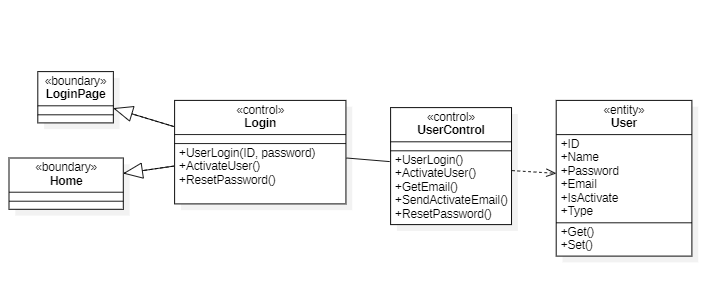
**1.2 类设计**

**1.2.1 用户登录子系统**

**1.2.1.1 系统简述**

该子系统主要承担用户在登录过程中的有关工作，包括了发送验证邮件、激活账户、用户登录、修改密码等操作。

**1.2.1.2 设计类图**

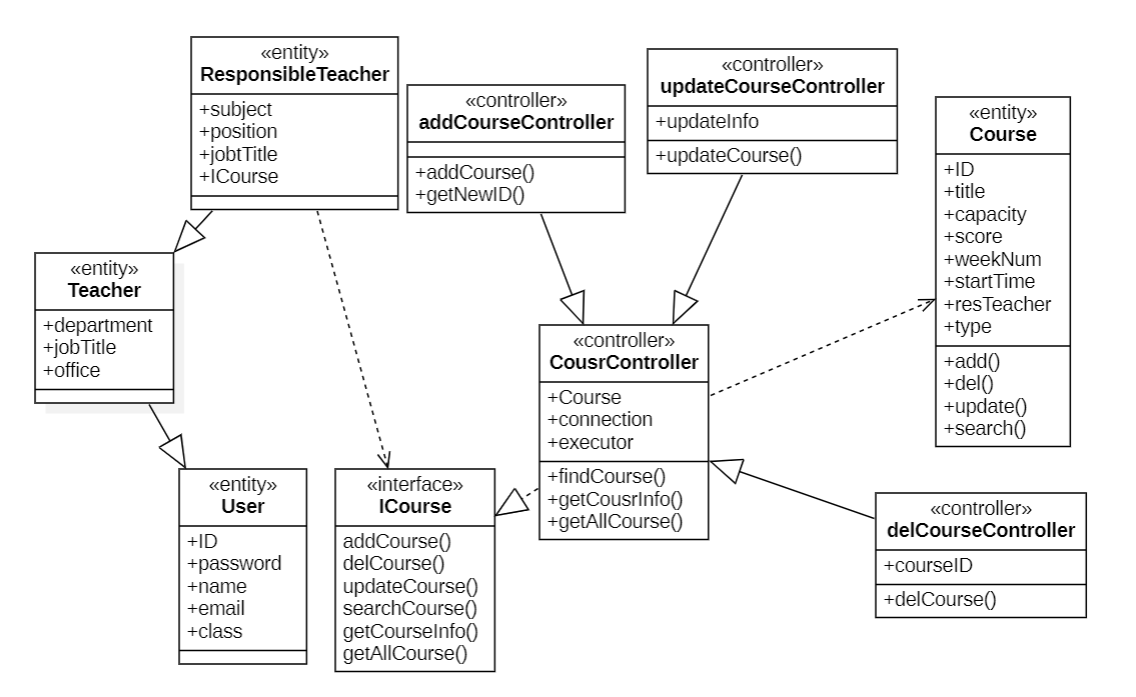


**1.2.2 实验课程管理子系统**

**1.2.2.1 系统简述**

该子系统主要承担与实验课程管理有关的所有工作，包括了教师添加、修改、删除课程等操作。

**1.2.2.2 设计类图**

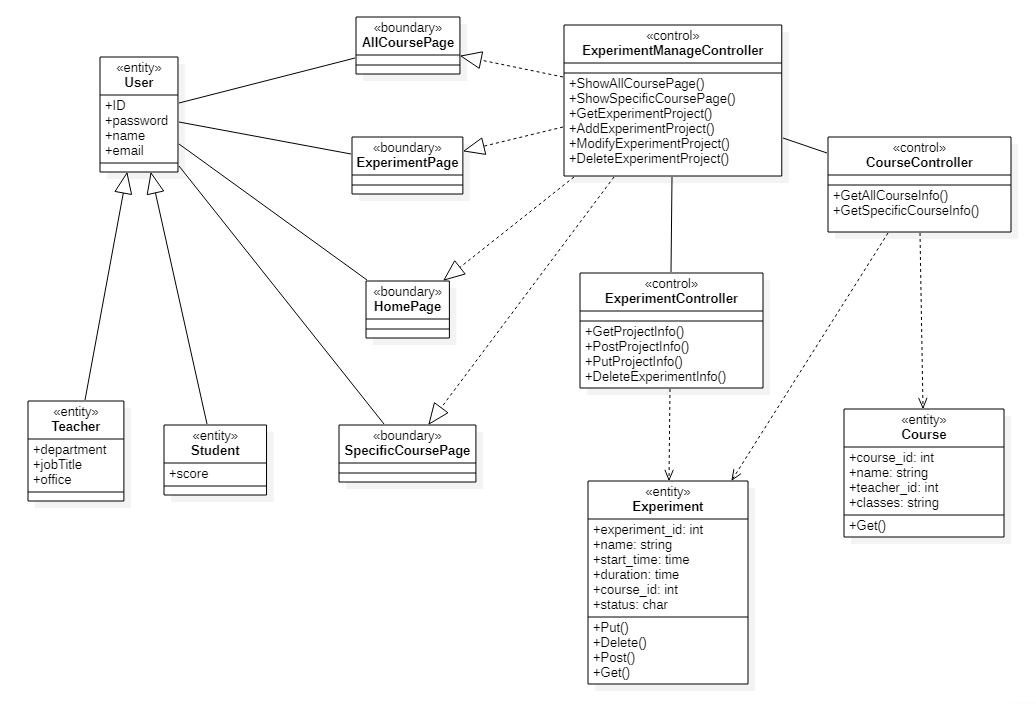


**1.2.3 实验项目管理子系统**

**1.2.3.1 系统简述**

该子系统主要承担与实验项目管理有关的所有工作，包括了教师添加、修改、删实验，学生参加实验等操作。

**1.2.3.2 设计类图**



**1.2.4 实验报告管理子系统**

**1.2.4.1 系统简述**

该子系统主要承担与实验报告管理有关的所有工作，包括了学生完成实验报告，教师和助教批改实验报告等操作。

**1.2.4.2 设计类图**

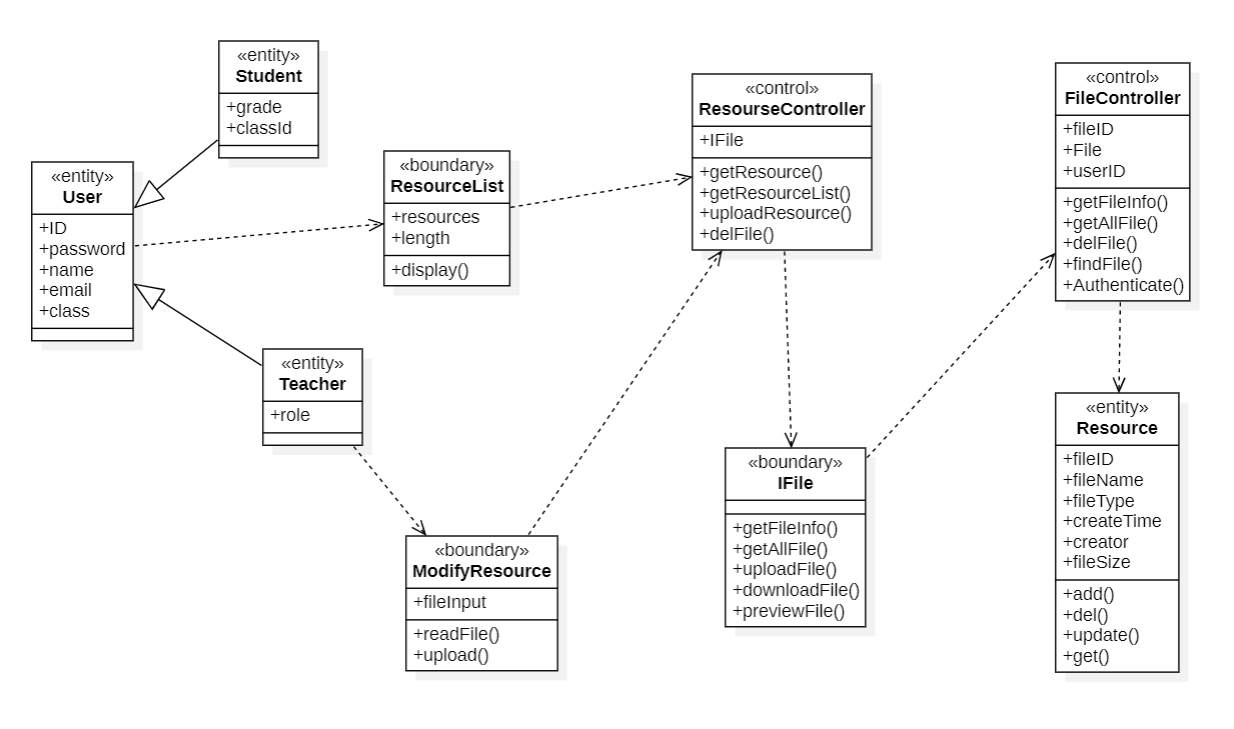


**1.2.5 课程资源管理子系统**

**1.2.5.1 系统简述**

该子系统主要承担与实验课程资源管理有关的所有工作，包括了教师上传实验资源、学生和教师下载实验资源等操作。

**1.2.5.2 设计类图**

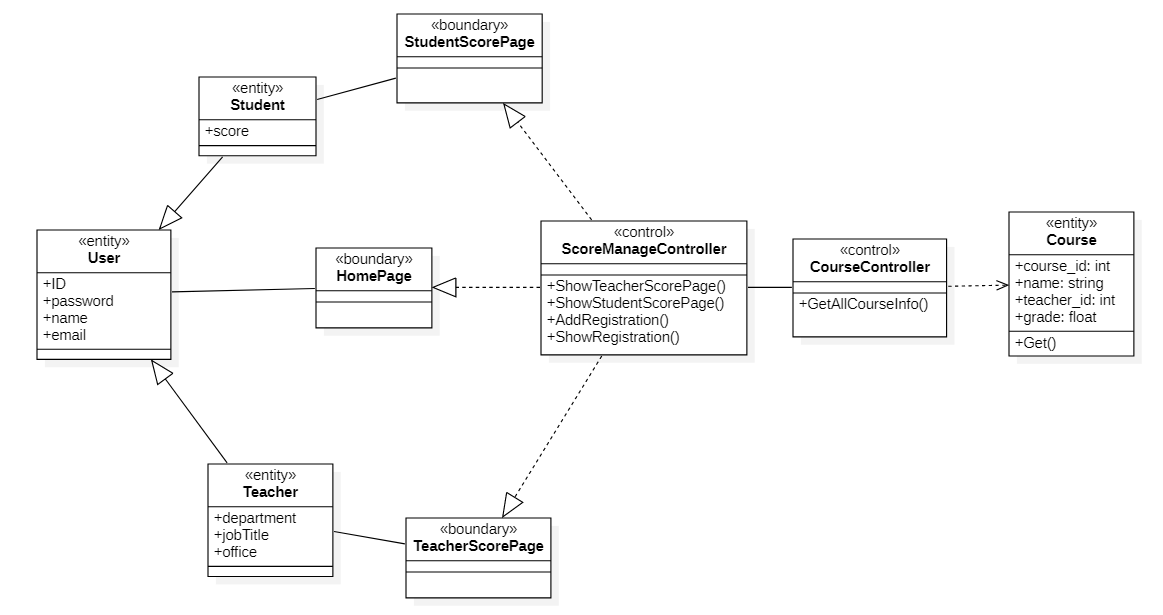


**1.2.6 成绩管理子系统**

**1.2.6.1 系统简述**

该子系统主要承担与实验成绩管理有关的所有工作，包括了教师发布签到、学生完成签到、教师和学生查看成绩等操作。

**1.2.6.2 设计类图**

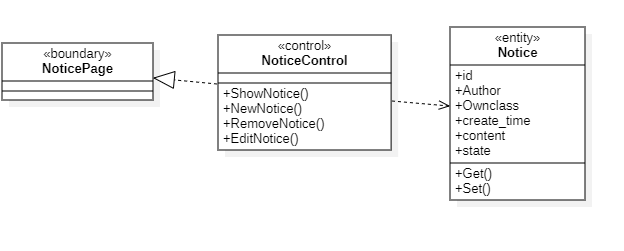


**1.2.7 通知公告子系统**

**1.2.7.1 系统简述**

该子系统主要承担与实验课程公告管理有关的所有工作，包括了教师发布、修改、删除公告等操作。

**1.2.7.2 设计类图**

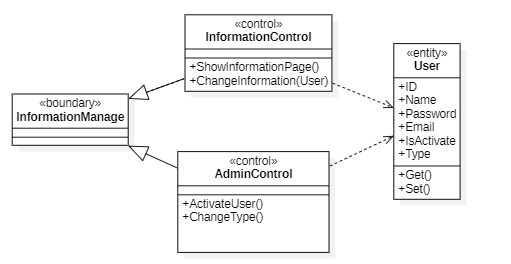


**1.2.8 个人信息子系统**

**1.2.8.1 系统简述**

该子系统主要承担与实验课程公告管理有关的所有工作，包括了用户修改个人信息等操作。

**1.2.8.2 设计类图**

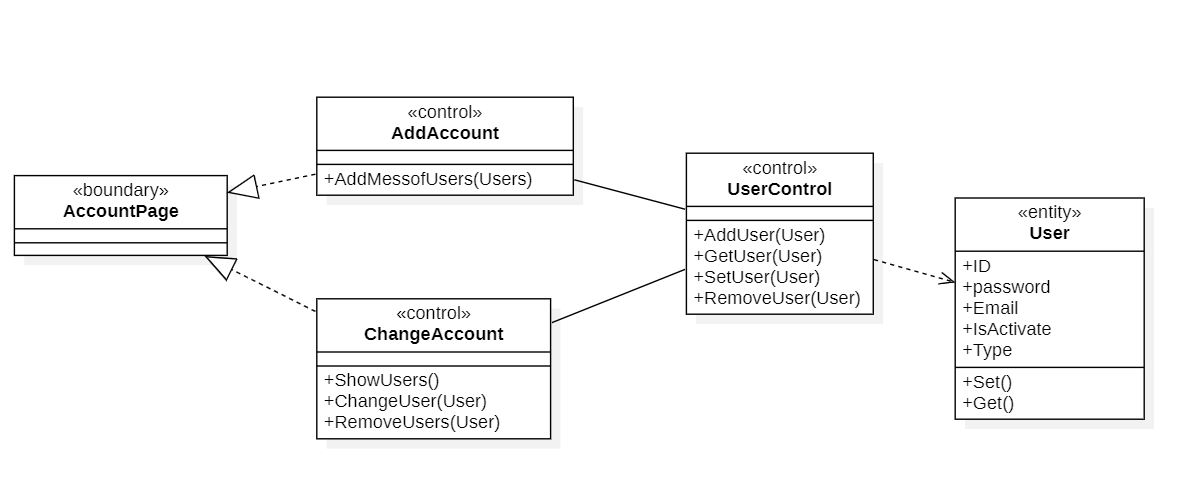


**1.2.9 系统管理子系统**

**1.2.9.1 系统简述**

该子系统主要承担与整个系统管理有关的所有工作，包括了用管理员批量添加用户、删除用户等操作。

**1.2.9.2 设计类图**



**2 架构设计**

**2.1 总体架构**

**2.1.1 前端架构设计**

为满足系统的跨平台要求和提高易用性，前端借助浏览器以web页面的形式呈现。以vue.js作为主体开发框架；为达到系统界面的美观性，使用了Element UI组件库，借助vue-router进行路由导航；vuex做前端存储管理；利用Ajax动态请求数据，更新页面，提升用户友好性，使用第三方库Axios对Ajax进行封装，降低开发难度。

vue是一款用于构建用户界面的JavaScript框架，其基于标准HTML，CSS和JavaScript，并提供了声明式，组件化的编程模型。借助其响应性特性，动态更新页面DOM，做到页面渲染和内部数据的双向同步。利用vue提供的单页面文件支持，模块化前端页面元素，降低系统耦合性。

Element UI将前端页面常用组件进行封装，提供了大量美观、易用的UI元素，借助该组件库，有效降低系统开发难度，提升系统观感。

vue-router为单页面应用而生，可方便的做到嵌套路由映射，动态路由导航，通过与vuex配合使用，动态生成路由表，做到细致的权限管理。

**2.1.2 后端架构设计**

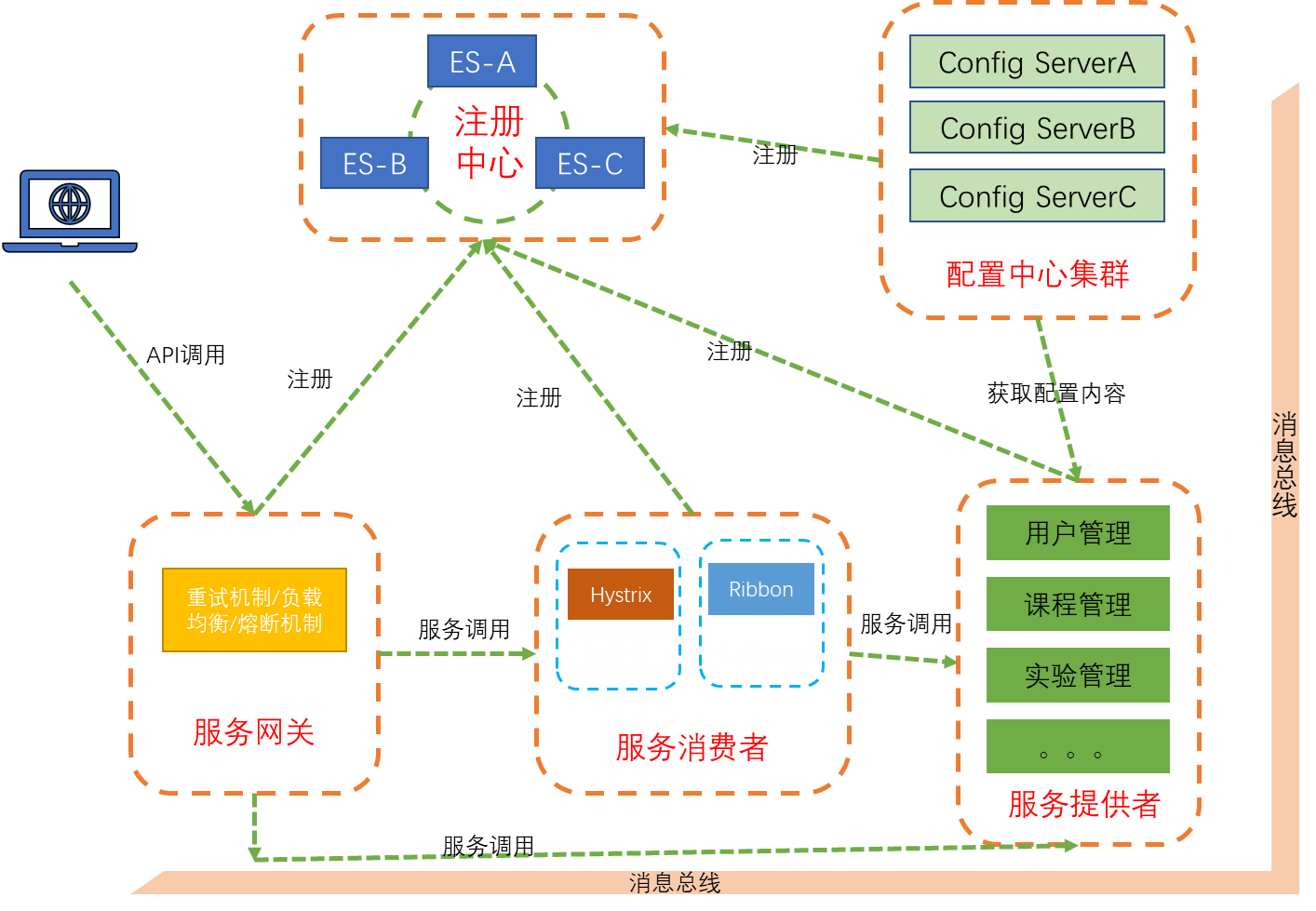
后端以Java作为开发语言，利用Spring生态，搭建微服务体系。微服务体系结构相比于单体服务，将业务模块进行了充分的解耦，提升了系统的健壮性，各模块也可按需扩容，业务切分同时降低了维护难度，便于融合其它技术栈，便于组织管理。

根据本实验教学管理平台的业务需求，将其分割为相对独立的7大模块，借助SpringCloud框架进行微服务体系的快速开发。SpringCloud借助Spring Boot对微服务架构中的技术框架进行了有序集成，简化了分布式系统中基础设施的开发，如服务注册发现，配置中心，负载均衡，熔断，数据监控等。

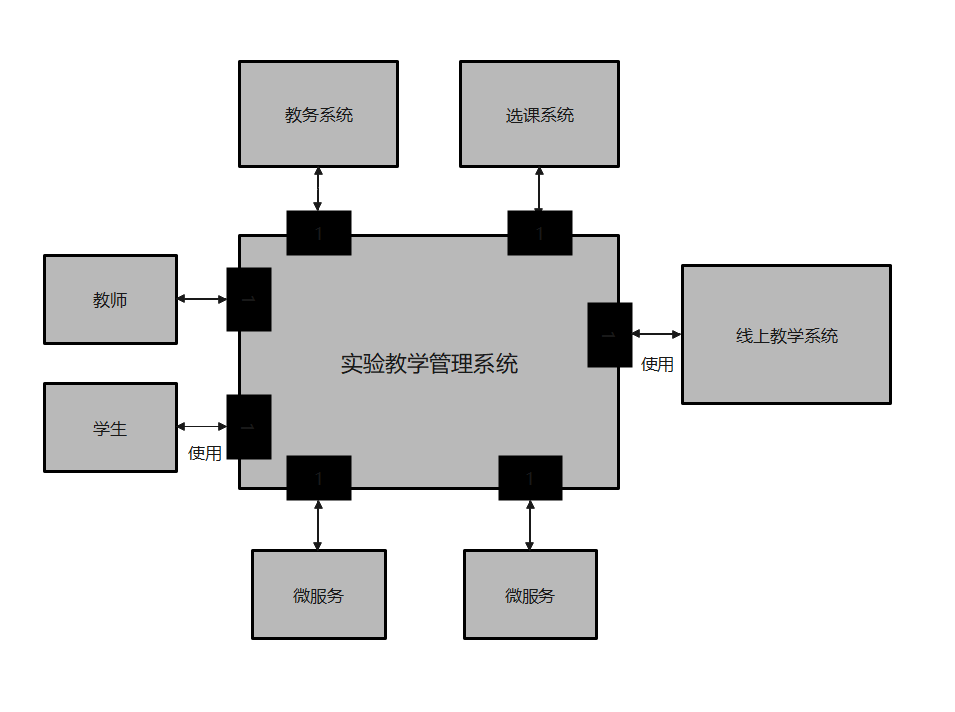
在本项目的开发中，采用Eureka进行服务的注册与发现，Feign进行声明式服务调用，GateWay作为API服务网关，为微服务的统一管理和相互调用提供支持。

**2.2 微服务架构**

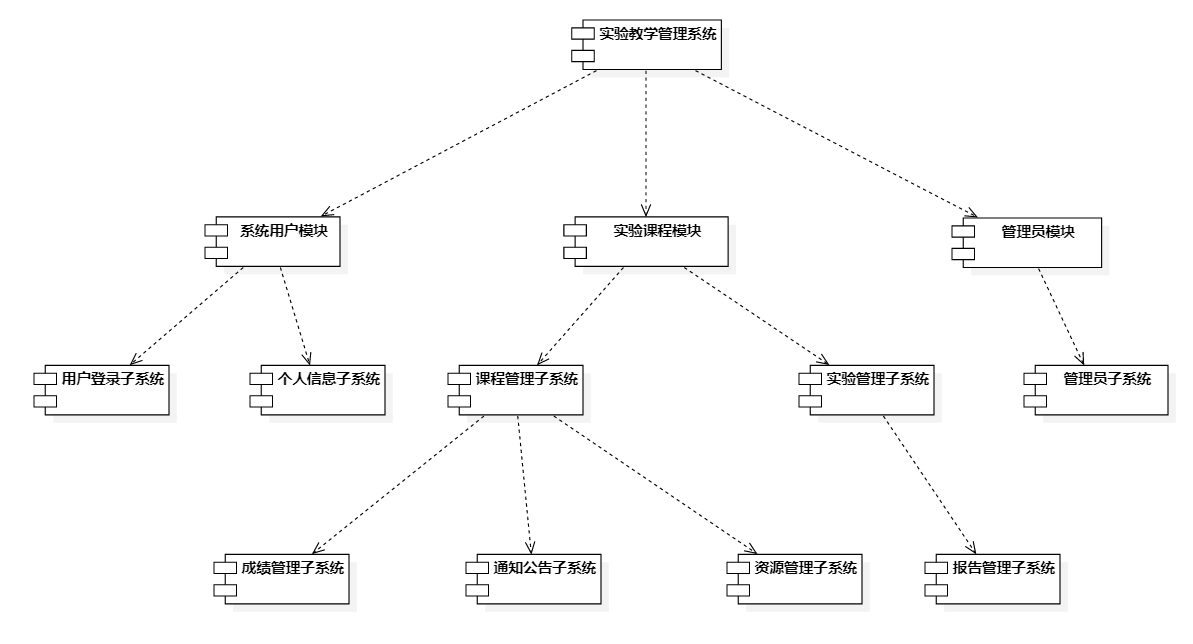
利用SpringCloud提供的工具链，我们搭建了适用于本套实验教学管理系统的微服务结构体系，具体包括用Nacos搭建的服务注册集群，以及服务配置集群，对于系统中可拆分的模块，我们另设一个微服务，并且在数据库层面上分表，拆分业务逻辑，尽可能降低各模块之间的耦合性，提高后期的可维护性，目前我们微服务架构体系可用下图简要描述。



**2.3 架构上下文**



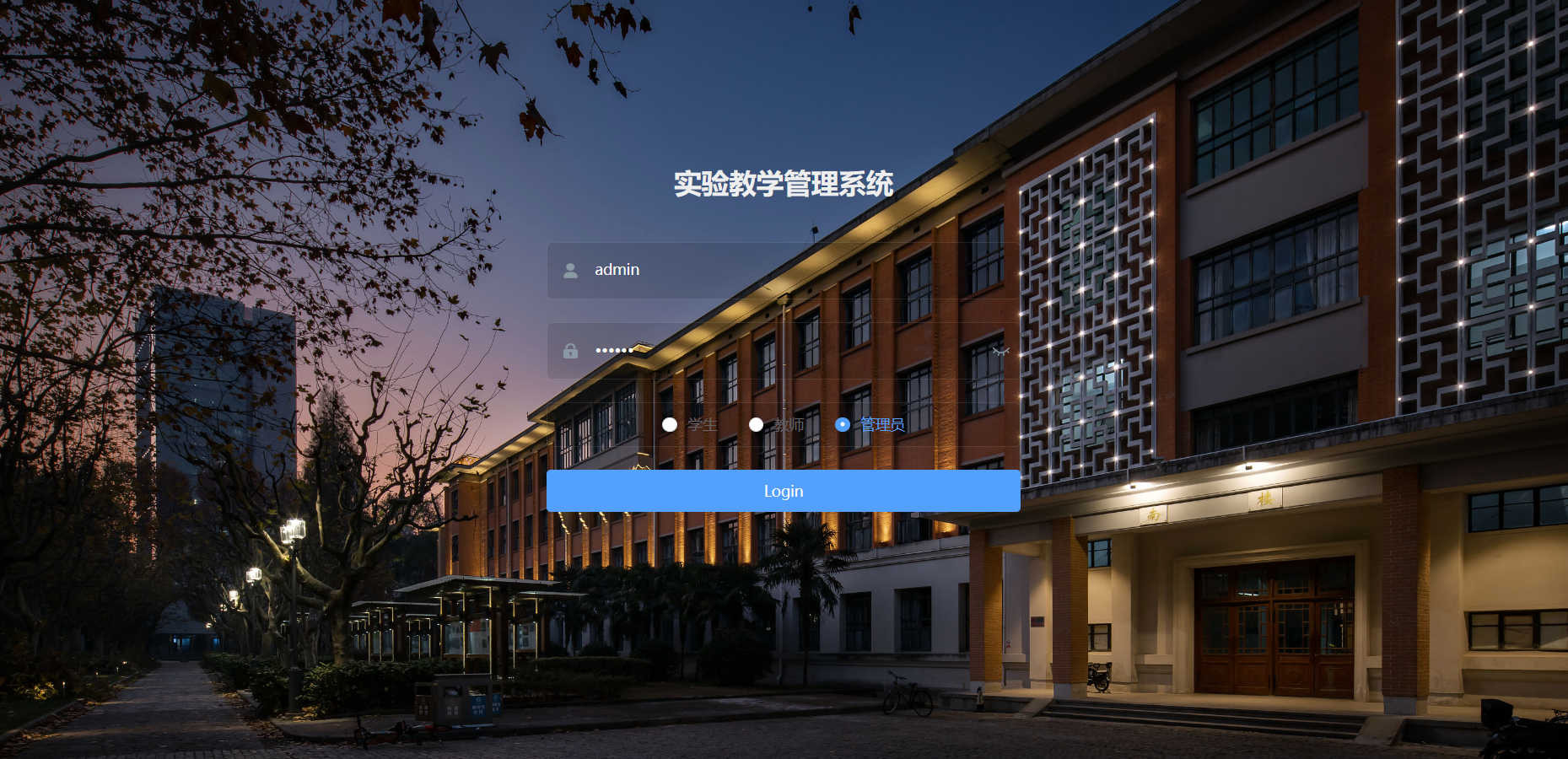
**2.4 系统内部架构**



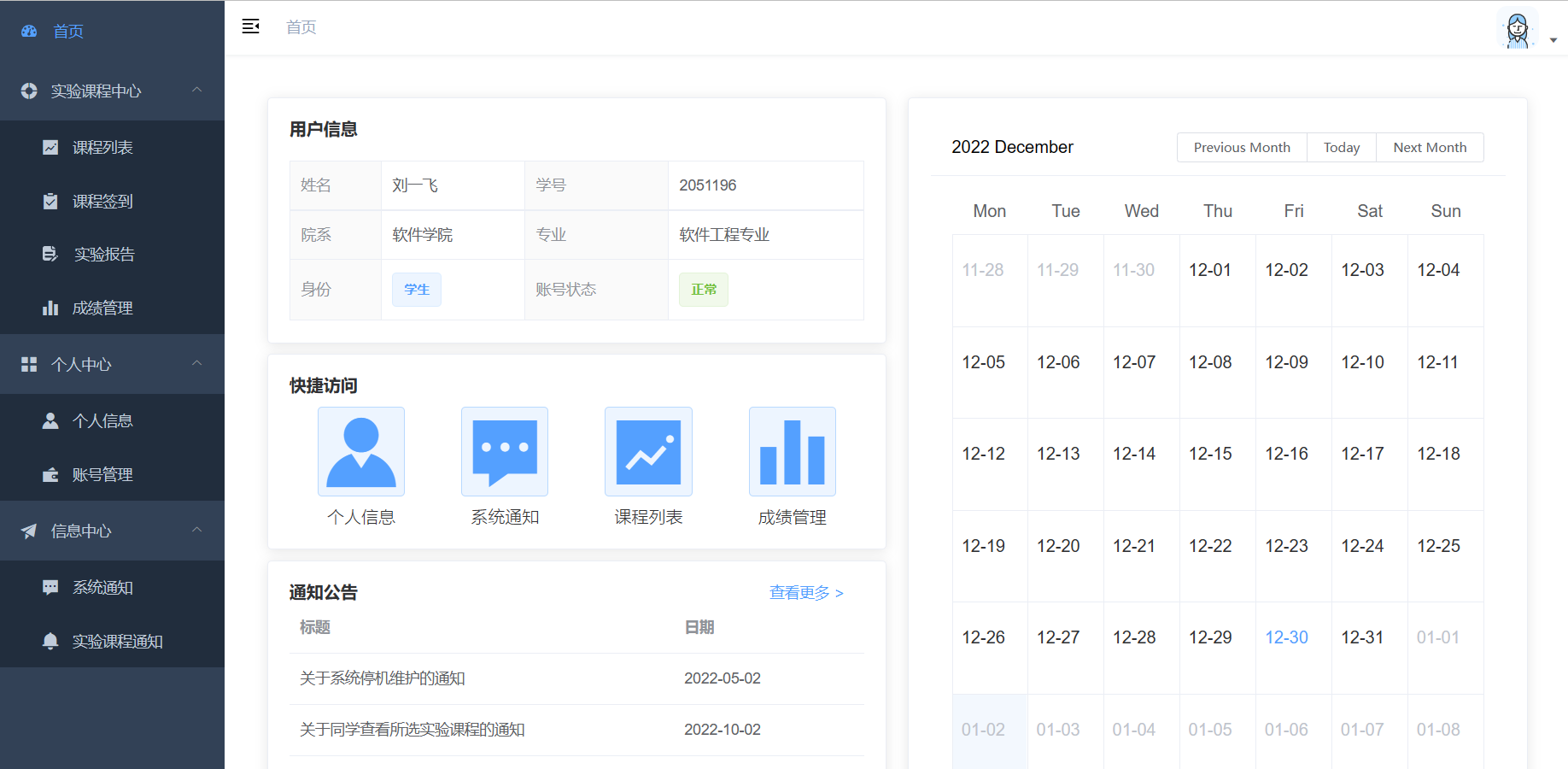
**3 接口设计**

**3.1 图形用户界面**

**3.1.1 登录界面**



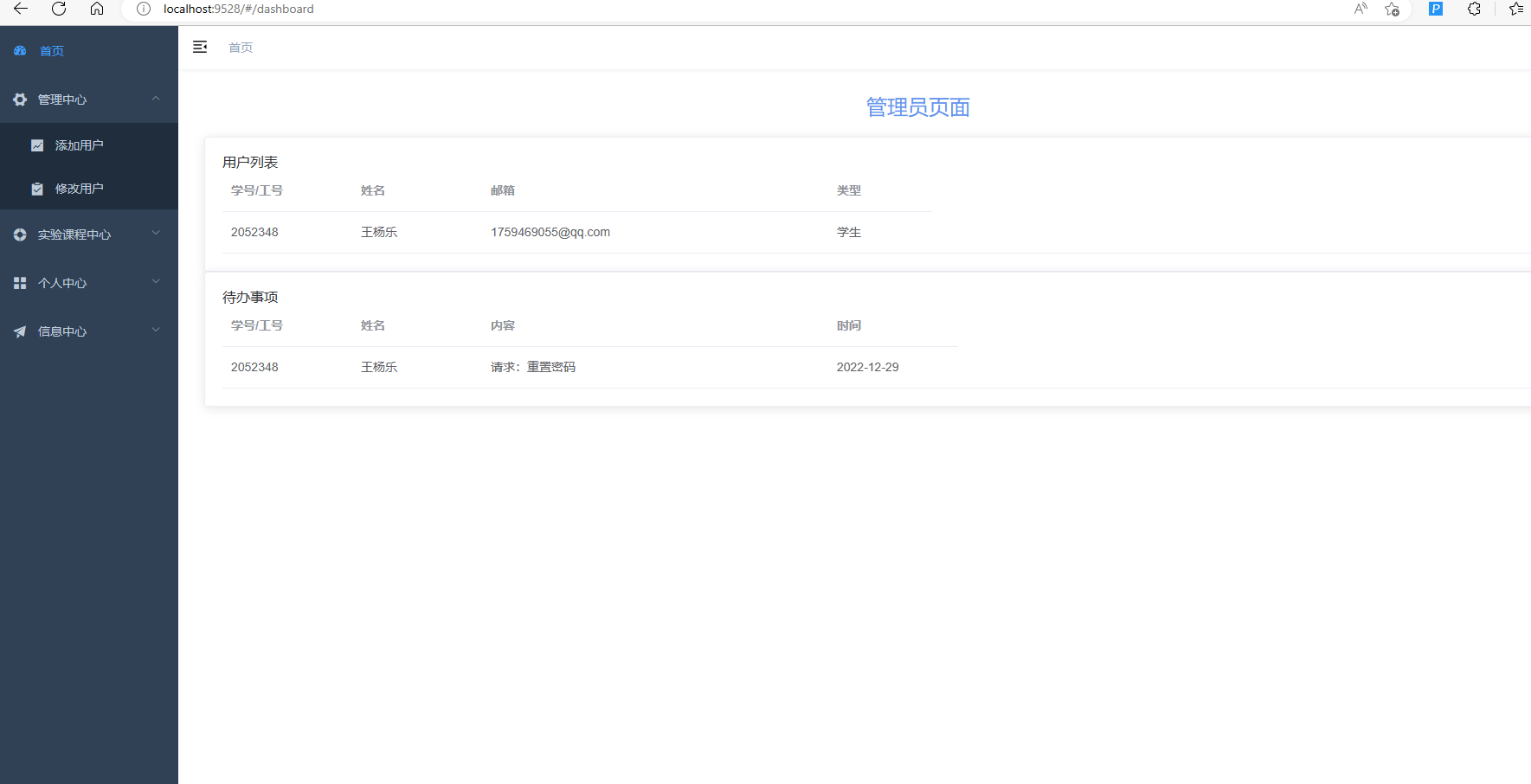
**3.1.2 学生导航界面**



**3.1.3 教师导航界面**



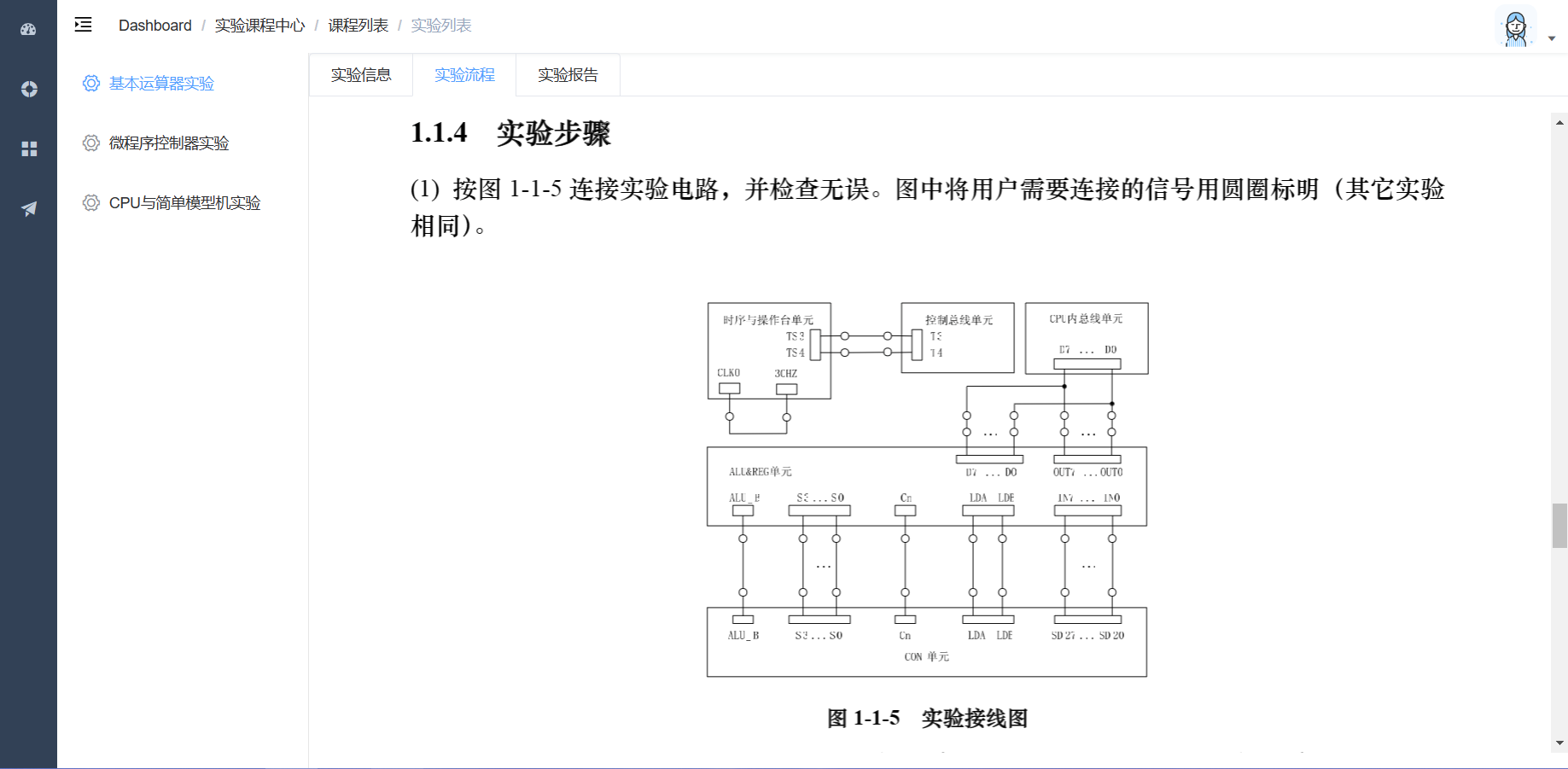
**3.1.4 管理员导航界面**



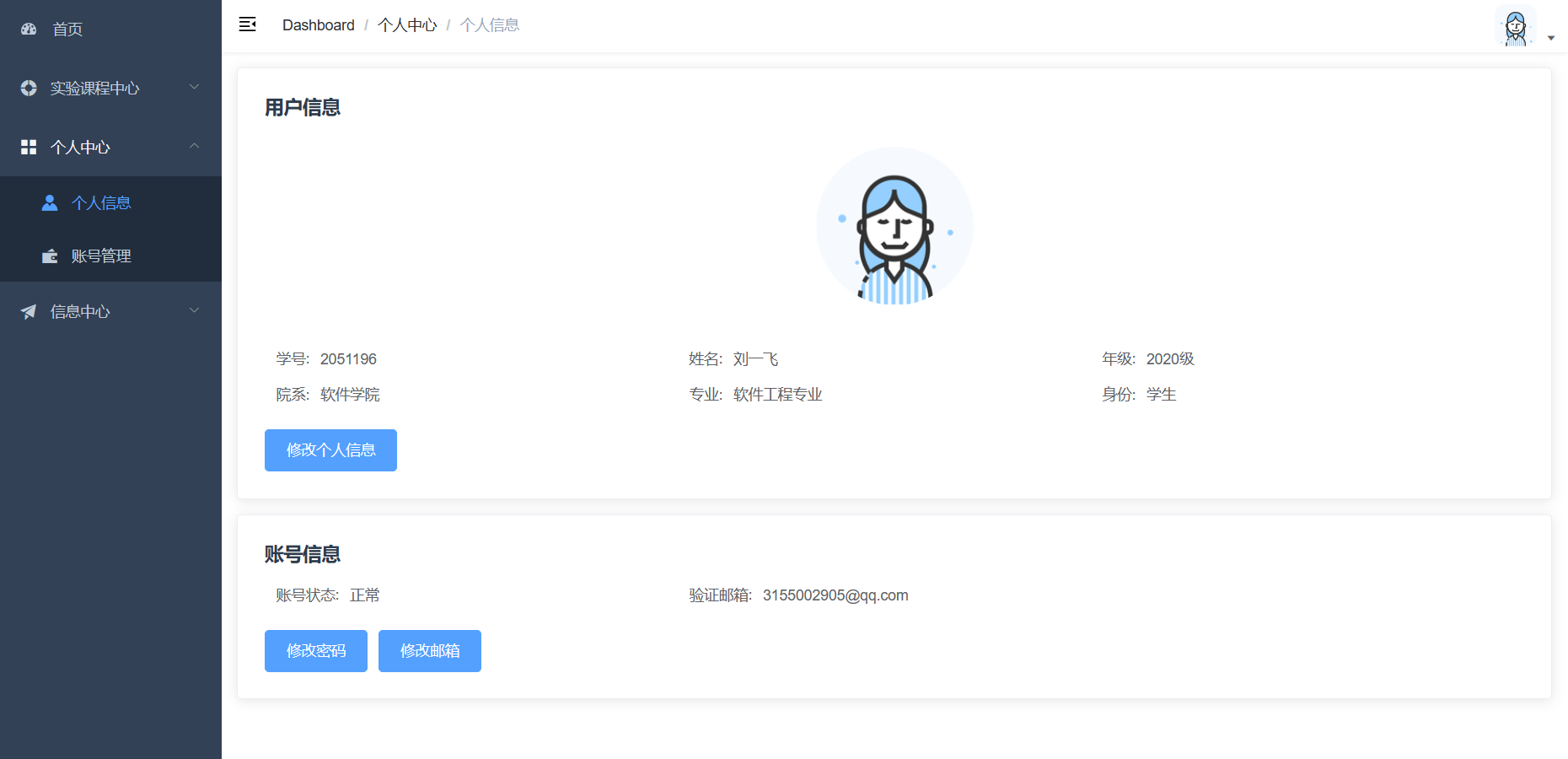
**3.1.5 实验课程界面**



**3.1.6 实验项目界面**



**3.1.7 个人信息界面**



**3.2 内部接口**

**3.2.1 用户登录子系统**

**3.2.1.1 用户账户激活**

|  |  |
| --- | --- |
| 接口地址 | /api/login/activate |
| 请求方式 | POST |
| 请求数据类型 | application/x-www-formrmurlencoded,application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于激活账号 |

* 请求示例:

|  |
| --- |
| JavaScript {  "id":"",  "email":"", } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| id | 学号或工号 | true | bigint |  |
| email | 邮箱 | true | string | 用户输入的验证邮箱 |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1,  "id":1,  "name":"",  "gender":"",  "phone":"" } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | status返回0表示失败，1表示成功 |
| id | 用户ID | bigint |  |
| name | 用户姓名 | string |  |
| gender | 用户性别 | string |  |
| phone | 用户手机号码 | string |  |

**3.2.1.2 用户登录**

|  |  |
| --- | --- |
| 接口地址 | /api/login/login |
| 请求方式 | POST |
| 请求数据类型 | application/x-www-formrmurlencoded,application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于验证登录请求 |

* 请求示例:

|  |
| --- |
| JavaScript {  "id":"",  "password":"" } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| id | 学号或工号 | true | bigint |  |
| password | 账户密码 | true | string |  |

* 响应示例

|  |
| --- |
| JavaScript {  "id":1,  "name":"",  "type":"",  "phone":"" } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| id | 用户ID | bigint |  |
| name | 用户姓名 | string |  |
| type | 用户类型 | string |  |
| phone | 用户手机号码 | string |  |

**3.2.1.3 忘记密码**

|  |  |
| --- | --- |
| 接口地址 | /api/login/login |
| 请求方式 | PUT |
| 请求数据类型 | application/x-www-formrmurlencoded,application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于用户忘记密码时重置密码 |

* 请求示例:

|  |
| --- |
| JavaScript {  "email":"",  "new\_passwd":"" } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| email | 邮箱名 | true | string | 系统会向邮箱中发送验证信息 |
| new\_passwd | 用户新设置的密码 | true | string |  |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1, } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | status返回0表示失败，1表示成功 |

**3.2.2 课程管理子系统**

**3.2.2.1 用户获取课程列表**

|  |  |
| --- | --- |
| 接口地址 | /api/course/getCourseList |
| 请求方式 | GET |
| 请求数据类型 | application/x-www-formrmurlencoded,application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于获取用户所有的课程列表 |

* 请求示例:

|  |
| --- |
| JavaScript {  "id":1, } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| id | 用户ID | true | bigint |  |

* 响应示例

|  |
| --- |
| JavaScript {  "courseList":[  {  "course\_id":1,  "name":"",  "introduction":"",  "create\_time":yyyy-MM-dd hh:mm:ss  }  ] } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| course\_id | 课程ID | bigint |  |
| name | 课程名称 | string |  |
| introduction | 课程简介 | string |  |
| create\_time | 课程创建时间 | date | 时间格式为yyyy-MM-dd hh:mm:ss |

**3.2.2.2 责任教师添加课程**

|  |  |
| --- | --- |
| 接口地址 | /api/course/addCourse |
| 请求方式 | POST |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于责任教师添加新的课程 |

* 请求示例:

|  |
| --- |
| JavaScript {  "teacher\_id":1,  "name":"",  "introduction":"",  "create\_time":yyyy-MM-dd hh:mm:ss  } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| teacher\_id | 责任教师ID | true | bigint |  |
| name | 课程名称 | true | string |  |
| introduction | 课程简介 | false | string |  |
| create\_time | 课程创建时间 | true | date | 时间格式为yyyy-MM-dd hh:mm:ss |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1, } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | status返回0表示失败，1表示成功 |

**3.2.2.3 责任教师修改课程**

|  |  |
| --- | --- |
| 接口地址 | /api/course/modifyCourse |
| 请求方式 | PUT |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于责任教师修改现有的课程信息 |

* 请求示例:

|  |
| --- |
| JavaScript {  "teacher\_id":1,  "course\_id":1,  "name":"",  "introduction":"",  "create\_time":yyyy-MM-dd hh:mm:ss  } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| teacher\_id | 责任教师ID | true | bigint |  |
| course\_id | 课程ID | true | bigint |  |
| name | 课程名称 | true | string |  |
| introduction | 课程简介 | false | string |  |
| create\_time | 课程创建时间 | true | date | 时间格式为yyyy-MM-dd hh:mm:ss |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1, } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | status返回0表示失败，1表示成功 |

**3.2.2.4 责任教师删除课程**

|  |  |
| --- | --- |
| 接口地址 | /api/course/removeCourse |
| 请求方式 | DELETE |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于责任教师删除课程 |

* 请求示例:

|  |
| --- |
| JavaScript {  "teacher\_id":1,  "course\_id":1, } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| teacher\_id | 责任教师ID | true | bigint |  |
| course\_id | 课程ID | true | bigint |  |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1, } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | status返回0表示失败，1表示成功 |

**3.2.3 实验管理子系统**

**3.2.3.1 用户获取实验列表**

|  |  |
| --- | --- |
| 接口地址 | /api/course/getExperimentList |
| 请求方式 | GET |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于用户获取实验项目列表 |

* 请求示例:

|  |
| --- |
| JavaScript {  "id":1,  "course\_id":1, } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| id | 用户ID | true | bigint |  |
| course\_id | 课程ID | true | bigint | 根据课程ID寻找该课程下属的所有实验 |

* 响应示例

|  |
| --- |
| JavaScript {  "experimentList":[  {  "experiment\_id":1,  "name":"",  "introduction":"",  "create\_time":yyyy-MM-dd hh:mm:ss,  "deadline":yyyy-MM-dd hh:mm:ss  }  ] } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| experiment\_id | 实验ID | bigint |  |
| name | 实验名称 | stirng |  |
| introduction | 实验简介 | string |  |
| create\_time | 实验创建时间 | date | 时间格式为yyyy-MM-dd hh:mm:ss |
| dead\_line | 实验截止时间 | date | 时间格式为yyyy-MM-dd hh:mm:ss |

**3.2.3.2 教师添加实验项目**

|  |  |
| --- | --- |
| 接口地址 | /api/course/addExperiment |
| 请求方式 | POST |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于教师添加新的实验项目 |

* 请求示例:

|  |
| --- |
| JavaScript {  "teacher\_id":1,  "name":"",  "introduction":"",  "create\_time":yyyy-MM-dd hh:mm:ss,  "dead\_line":yyyy-MM-dd hh:mm:ss } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| teacher\_id | 教师ID | true | bigint |  |
| name | 实验名称 | true | string |  |
| introduction | 实验简介 | false | string |  |
| create\_time | 实验创建时间 | true | date | 时间格式为yyyy-MM-dd hh:mm:ss |
| dead\_line | 实验截止时间 | true | date | 时间格式为yyyy-MM-dd hh:mm:ss |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1, } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | status返回0表示失败，1表示成功 |

**3.2.3.3 教师修改实验项目**

|  |  |
| --- | --- |
| 接口地址 | /api/course/modifyExperiment |
| 请求方式 | PUT |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于教师修改实验项目信息 |

* 请求示例:

|  |
| --- |
| JavaScript {  "teacher\_id":1,  "experiment\_id":1,  "name":"",  "introduction":"",  "create\_time":yyyy-MM-dd hh:mm:ss  } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| teacher\_id | 责任教师ID | true | bigint |  |
| experiment\_id | 实验ID | true | bigint |  |
| name | 实验名称 | true | string |  |
| introduction | 实验简介 | false | string |  |
| create\_time | 实验创建时间 | true | date | 时间格式为yyyy-MM-dd hh:mm:ss |
| dead\_line | 实验截止时间 | true | date | 时间格式为yyyy-MM-dd hh:mm:ss |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1, } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | status返回0表示失败，1表示成功 |

**3.2.3.4 教师删除实验项目**

|  |  |
| --- | --- |
| 接口地址 | /api/course/removeExperiment |
| 请求方式 | DELETE |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于用户获取实验项目列表 |

* 请求示例:

|  |
| --- |
| JavaScript {  "teacher\_id":1,  "experiment\_id":1, } |

* 请求参数：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 是否必须 | 数据类型 | 备注 |
| teacher\_id | 责任教师ID | true | bigint |  |
| experiment\_id | 实验ID | true | bigint |  |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1, } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | status返回0表示失败，1表示成功 |

**3.2.4 实验报告子系统**

**3.2.4.1 教师上传实验模板**

|  |  |
| --- | --- |
| 接口地址 | /api/report/uploadreporttemplate |
| 请求方式 | POST |
| 请求数据类型 | application/binary |
| 响应数据类型 | application/json |
| 接口描述 | 用于上传实验报告模板文件(.docx) |

请求示例：

|  |
| --- |
| JavaScript {  "teacher\_id": 1,  "course\_id": 1,  "experiment\_id": 1,  "template\_file":"",  } |

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| teacher\_id | 教师id | int |  |
| course\_id | 课程编号 | int |  |
| experiment\_id | 实验编号 | int |  |
| template\_file | 模板文件内容 | file |  |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1 } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | 0表示失败，1表示成功 |

**3.2.4.2 学生获取实验模板**

|  |  |
| --- | --- |
| 接口地址 | /api/report/downloadreporttemplate |
| 请求方式 | GET |
| 请求数据类型 | application/x-www-formrmurlencoded |
| 响应数据类型 | application/binary |
| 接口描述 | 用于下载实验报告模板文件(.docx) |

* 请求参数：无
* 响应示例：

|  |
| --- |
| JavaScript {  "teacher\_id": 1,  "course\_id": 1,  "experiment\_id": 1,  "template\_file":"",  } |

* 响应参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| teacher\_id | 教师id | int |  |
| course\_id | 课程编号 | int |  |
| experiment\_id | 实验编号 | int |  |
| template\_file | 模板文件内容 | file |  |

**3.2.4.3 学生提交实验报告**

|  |  |
| --- | --- |
| 接口地址 | /api/report/reporthandin |
| 请求方式 | POST |
| 请求数据类型 | application/binary |
| 响应数据类型 | application/json |
| 接口描述 | 用于学生提交实验报告至服务器 |

* 请求示例：

|  |
| --- |
| JavaScript {  "student\_id": 1,  "course\_id":1,  "experiment\_id":1,  "file":"",  } |

* 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| student\_id | 学生id | int |  |
| course\_id | 课程编号 | int |  |
| experiment\_id | 实验编号 | int |  |
| file | 报告文件内容 | file |  |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1 } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | 0表示失败，1表示成功 |

**3.2.4.4 教师/助教批阅实验报告**

|  |  |
| --- | --- |
| 接口地址 | /api/report/checkreport |
| 请求方式 | POST |
| 请求数据类型 | application/x-www-formrmurlencoded |
| 响应数据类型 | application/json |
| 接口描述 | 用于给学生的实验报告打分 |

* 请求示例：

|  |
| --- |
| JavaScript {  "student\_id": 1,  "course\_id": 1,  "experiment\_id": 1,  "score": 1,  } |

* 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| student\_id | 学生id | int |  |
| course\_id | 课程编号 | int |  |
| experiment\_id | 实验编号 | int |  |
| score | 实验报告分数 | int |  |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1 } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | 0表示失败，1表示成功 |

**3.2.5 课程文件子系统**

**3.2.5.1 获取该课程的所有相关文件**

|  |  |
| --- | --- |
| 接口地址 | /api/resource |
| 请求方式 | GET |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 获取课程的所有相关文件 |

* 请求示例

|  |
| --- |
| JavaScript {   "courseId":"", } |

* 响应示例

|  |
| --- |
| JavaScript {  "courseId":"",  "courseName":"",  "sources":[  {  "fileName":"",  "fileSzie":"",  "url":""  },  ] } |

**3.2.5.2 教师上传文件**

|  |  |
| --- | --- |
| 接口地址 | /api/resource/postone |
| 请求方式 | POST |
| 请求数据类型 | application/json、application/octet-stream |
| 响应数据类型 | \*/\* |
| 接口描述 | 上传文件至数据库 |

* 请求示例

|  |
| --- |
| JavaScript {   "courseId":"",  "courseName":"",  "source":  {  "fileName":"",  "fileSzie":"",  "filetype":"",  "content":"",  }, } |

* 响应示例

|  |
| --- |
| JavaScript {  "status": 1, } |

**3.2.5.3 用户下载文件**

|  |  |
| --- | --- |
| 接口地址 | /api/resource/getone |
| 请求方式 | GET |
| 请求数据类型 | application/json |
| 响应数据类型 | application/octet-stream |
| 接口描述 | 获取文件下载链接 |

* 请求示例

|  |
| --- |
| JavaScript {   "courseId":"",  "courseName":"",  "sourcepath":"", } |

* 响应示例

|  |
| --- |
| JavaScript {  "source":  {  "fileName":"",  "fileSzie":"",  "filetype":"",  "content":"",  }, } |

**3.2.6 学生成绩子系统**

**3.2.6.1 教师提交成绩**

|  |  |
| --- | --- |
| 接口地址 | /api/grade/submit |
| 请求方式 | POST |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 教师提交学生课程成绩总评 |

* 请求参数

|  |
| --- |
| JavaScript {  "courseId":"",  "teacherId":"",  "grade":[  {  "studentId":"",  "grade":""  }  ] } |

**3.2.6.2 学生查看成绩**

|  |  |
| --- | --- |
| 接口地址 | /api/grade/selfGrade |
| 请求方式 | GET |
| 请求数据类型 | application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 学生查看自己各个课程的成绩 |

* 请求示例

|  |
| --- |
| JavaScript {  "studentId":"" } |

* 响应示例

|  |
| --- |
| JavaScript {  "Id":"",  "grade":[  {  "courseId":"",  "courseName":"",  "score":"",  "grade":""  },  。。。  ] } |

**3.2.7 通知公告子系统**

**3.2.7.1 获取公告列表**

|  |  |
| --- | --- |
| 接口地址 | /api/notice/getNoticeList |
| 请求方式 | GET |
| 请求数据类型 | application/x-www-formrmurlencoded |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于获取公告列表 |

响应示例：

|  |
| --- |
| JavaScript {  "content": "",  "course\_id": "",   "create\_time": "",  "notice\_id": "",   "teacher\_id": "",  "title": "",  } |

响应参数：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 请求类型 | 是否必须 | 数据类型 | 备注 |
| notice | Notice | POST | true | body |  |
| content | 公告内容 |  | false | string |  |
| course\_id | 课号 |  | false | int |  |
| create\_time | 发布时间 |  | false | date |  |
| notice\_id | 公告id |  | false | int | 数据库在创建时生成 |
| teacher\_id | 发布者id |  | false | int |  |
| title | 公告标题 |  | false | string |  |

* 响应示例
* 响应参数

**3.2.7.2 新增公告信息**

|  |  |
| --- | --- |
| 接口地址 | /api/notice/newNotice |
| 请求方式 | POST |
| 请求数据类型 | application/x-www-formrmurlencoded,application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 新增公告 |

请求示例：

|  |
| --- |
| JavaScript {  "content": "",  "course\_id": 0,   "create\_time": "",  "teacher\_id": "",  "title": "",  } |

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| notice |  |  |  |
| content | 公告内容 | string |  |
| course\_id | 课号 | int |  |
| create\_time | 日期 | date | date格式 |
| publisher\_id | 发布者ID | int |  |
| title | 公告标题 | string |  |

* 响应示例
* 响应参数

**3.2.7.3 更新公告状态**

|  |  |
| --- | --- |
| 接口地址 | /api/admin/updateUser |
| 请求方式 | POST |
| 请求数据类型 | application/x-www-formrmurlencoded |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于更新某个特定公告 |

* 请求示例：

|  |
| --- |
| JavaScript {  "notice\_id": 0,   "state":0,  } |

* 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| notice\_id | 公告ID | int |  |
| state | 公告状态 | int |  |

* 响应示例

|  |
| --- |
| JavaScript {  "state":1 } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| state | 公告当前状态 | int |  |

**3.2.8 个人信息子系统**

**3.2.8.1 个人信息获取**

|  |  |
| --- | --- |
| 接口地址 | /api/person/getInformation/ |
| 请求方式 | GET |
| 请求数据类型 | application/x-www-formrmurlencoded |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于请求个人信息 |

请求示例:

|  |
| --- |
| JavaScript {  "id":"", } |

请求参数：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 请求类型 | 是否必须 | 数据类型 | 备注 |
| id | 学号或工号 |  | false |  |  |

响应示例：

|  |
| --- |
| JavaScript {  "id":"",  "name":"",  "gender":1,  "email":"",  "phone":"",  "avatar":"",  "schoolNum":"", } |

响应参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| user | 用户信息结构体 | body |  |
| id | 学号或工号 | int |  |
| name | 用户姓名 | string |  |
| gender | 性别 | boolean |  |
| email | 邮箱 | string |  |
| phone | 电话 | string |  |
| avatar | 头像 | string |  |

**3.2.8.2 修改个人信息**

|  |  |
| --- | --- |
| 接口地址 | /api/person/setInformation |
| 请求方式 | PUT |
| 请求数据类型 | application/x-www-formrmurlencoded,application/json |
| 响应数据类型 | \*/\* |
| 接口描述 | 用于更新个人信息 |

请求示例:

|  |
| --- |
| JavaScript {  "id":"",  "name":"",  "gender":1,  "email":"",  "phone":"", } |

请求参数：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数名称 | 参数说明 | 请求类型 | 是否必须 | 数据类型 | 备注 |
| user | 用户信息结构体 | body | true | body |  |
| id | 学号或工号 | int | false | int |  |
| name | 用户姓名 | string | false | string |  |
| gender | 性别 | boolean | false | int |  |
| email | 邮箱 | string | false | string |  |
| phone | 电话 | string | false | string |  |

* 响应示例
* 响应参数

**3.2.9 管理员子系统**

**3.2.9.1 录入账号列表**

|  |  |
| --- | --- |
| 接口地址 | /api/admin/uploadUser |
| 请求方式 | POST |
| 请求数据类型 | application/x-www-formrmurlencoded |
| 响应数据类型 | application/json |
| 接口描述 | 用于上传包含账户信息的excel文件(.xlsx) |

请求示例：

|  |
| --- |
| JavaScript {  "Userlist":[  "ID": 1,  "name":"",  "gender":1,  "email":"",  "type":"",  ]  } |

请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| Userlist | 存储在excel文件中上传 |  |  |
| ID | 学号/工号 | int |  |
| name | 用户姓名 | string |  |
| gender | 性别 | int |  |
| email | 邮箱 | string |  |
| type | 类型 | string |  |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1 } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | 0表示失败，1表示成功 |

**3.2.9.2 获取账号列表**

|  |  |
| --- | --- |
| 接口地址 | /api/admin/downloadUser |
| 请求方式 | GET |
| 请求数据类型 | application/x-www-formrmurlencoded |
| 响应数据类型 | application/json |
| 接口描述 | 用于获取账户信息 |

* 请求参数：无
* 响应示例：

|  |
| --- |
| JavaScript {  "Userlist":[  "ID": 1,  "name":"",  "gender":1,  "email":"",  "type":"",  ]  } |

* 响应参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| ID | 学号/工号 | int |  |
| name | 用户姓名 | string |  |
| gender | 性别 | int |  |
| email | 邮箱 | string |  |
| type | 类型 | string |  |

**3.2.9.3 更新账号信息**

|  |  |
| --- | --- |
| 接口地址 | /api/admin/updateUser |
| 请求方式 | POST |
| 请求数据类型 | application/x-www-formrmurlencoded |
| 响应数据类型 | application/json |
| 接口描述 | 用于更新某个特定用户信息 |

* 请求示例：

|  |
| --- |
| JavaScript {  "ID": 1,  "name":"",  "gender":"",  "email":"",  "type":"",  } |

* 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| ID | 学号/工号 | int | 禁止修改，只能注销后新增 |
| name | 用户姓名 | string |  |
| gender | 性别 | string |  |
| email | 邮箱 | string |  |
| type | 类型 | string |  |

* 响应示例

|  |
| --- |
| JavaScript {  "status":1 } |

* 响应参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数说明 | 类型 | 备注 |
| status | 操作成功与否 | int | 0表示失败，1表示成功 |

**3.3 外部接口**

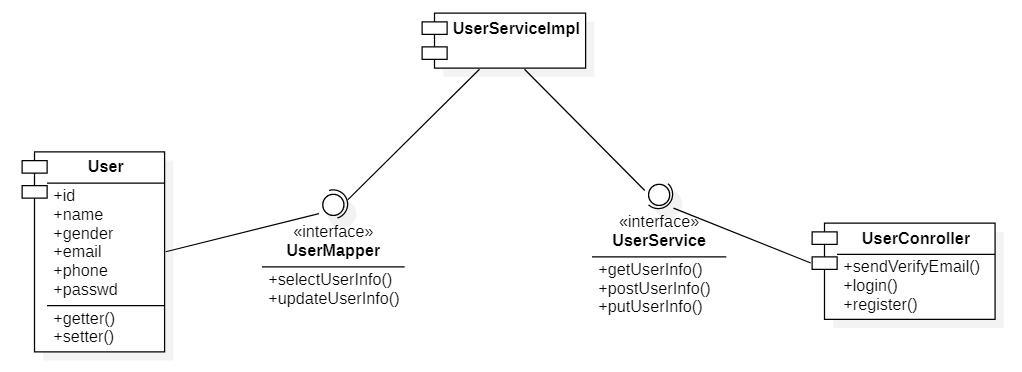
**3.3.1 JavaMail邮件发送**

通过调用JavaMail API实现用户在登录注册时需要完成的邮件验证功能，使用 SMTP 协议将邮件发送给对应用户。

**4 构件设计**

**4.1 用户管理构件**

**4.1.1 构件图**



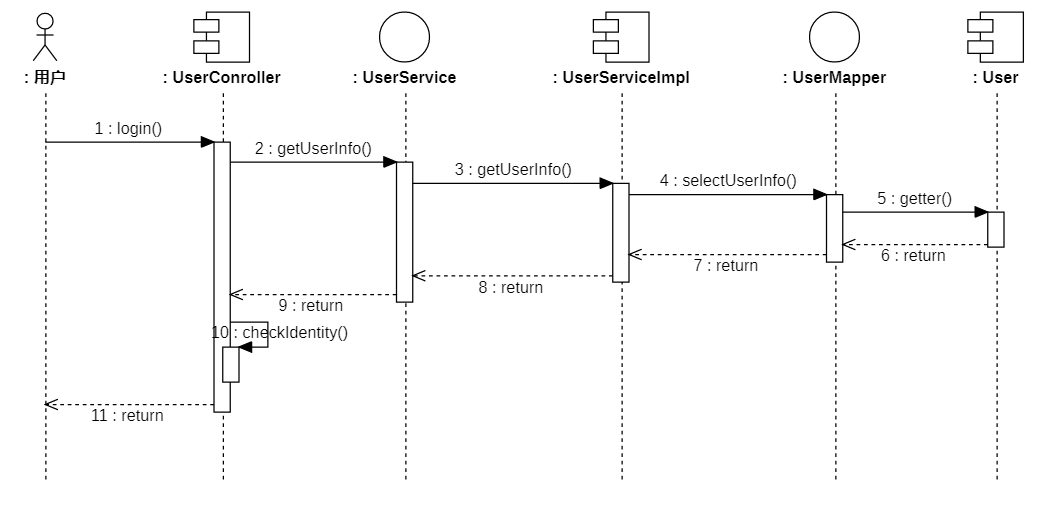
**4.1.2 核心操作**

**4.1.2.1 login**

**4.1.2.1.1 操作简述**

该操作对应用户正常登录，调用Controller中对外的login方法后，会调用业务层Service中的getUserInfo获取传入的登录信息中的用户ID，接着调用DAO层Mapper中的selectUserInfo根据用户ID在数据库中查找到对应的用户信息并返回，UserController会比对用户输入的信息和数据库中保存的信息，并根据结果判断是否登录成功。

**4.1.2.1.2 时序图**

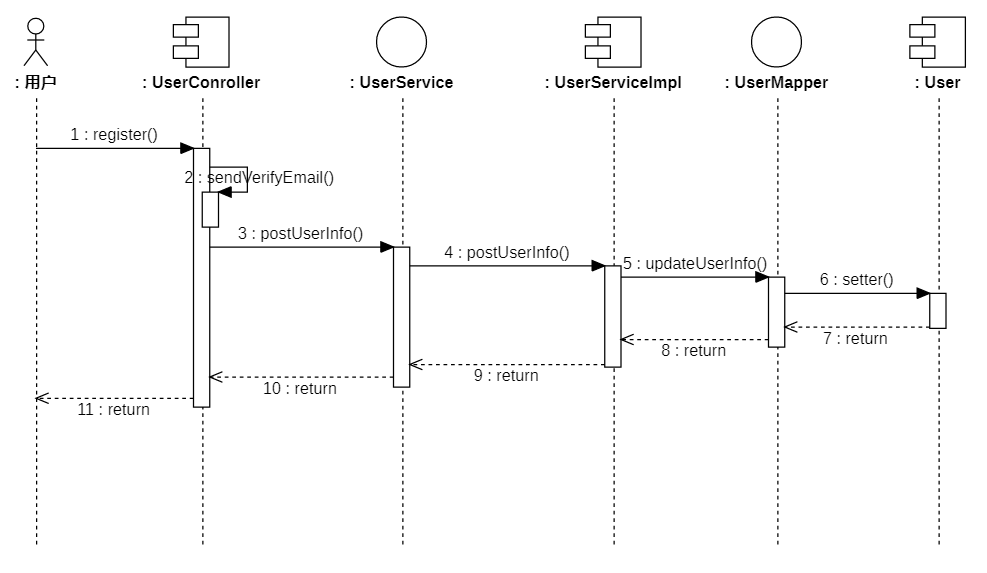


**4.1.2.2 register**

**4.1.2.2.1 操作简述**

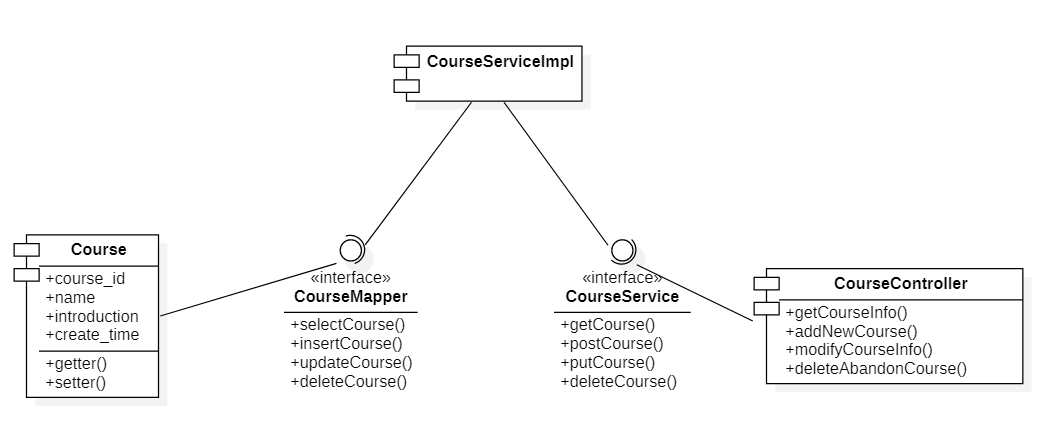
该操作对应用户的账户注册，调用Controller中对外的register方法后，会调用业务层Service中的sendVerifyEmail根据用户输入的邮箱信息向用户发送验证邮件，用户完成验证后，会调用Service中的postUserInfo将用户输入的信息组装成用户对象，接着调用DAO层Mapper中的updateUserInfo根据用户ID在数据库中更新对应的用户信息，从而完成账户注册流程。

**4.1.2.2.2 时序图**



**4.2 课程管理构件**

**4.2.1 构件图**



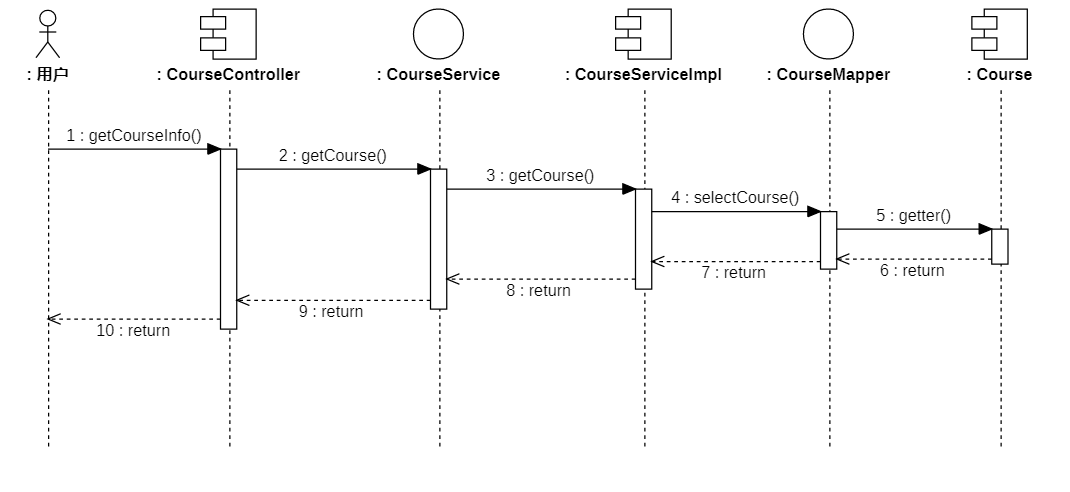
**4.2.2 核心操作**

**4.2.2.1 getCourseInfo**

**4.2.2.1.1 操作简述**

该操作对应用户获取课程信息，调用Controller中对外的getCourseInfo方法后，会调用业务层Service中的getCourse获取传入的课程ID，接着调用DAO层Mapper中的selectCourse根据课程ID在数据库中查找到对应的课程信息并返回。

**4.2.2.1.2 时序图**

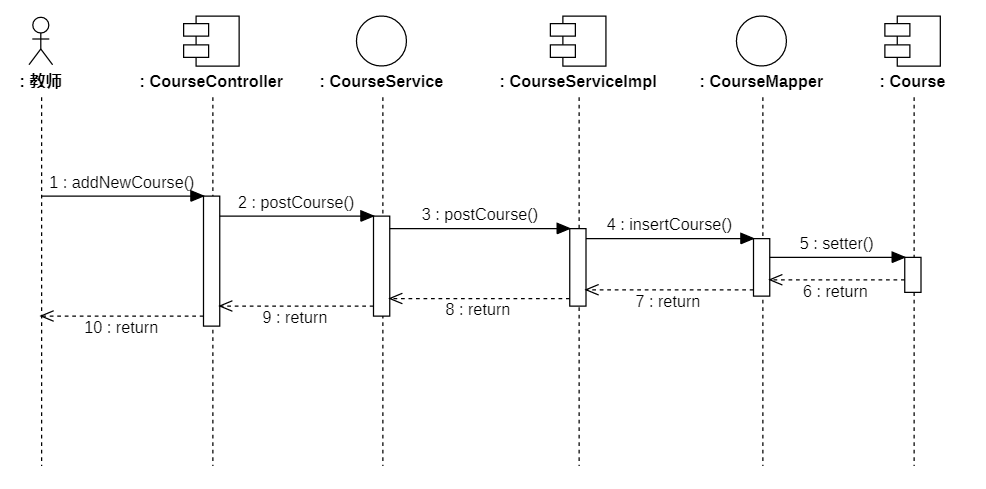


**4.2.2.2 addNewCourse**

**4.2.2.2.1 操作简述**

该操作对应教师添加新的实验课程，调用Controller中对外的addNewCourse方法后，会调用业务层Service中的postCourse对传入的课程信息封装为课程对象，接着调用DAO层Mapper中的insertCourse完成课程信息在数据库中的插入。

**4.2.2.2.2 时序图**

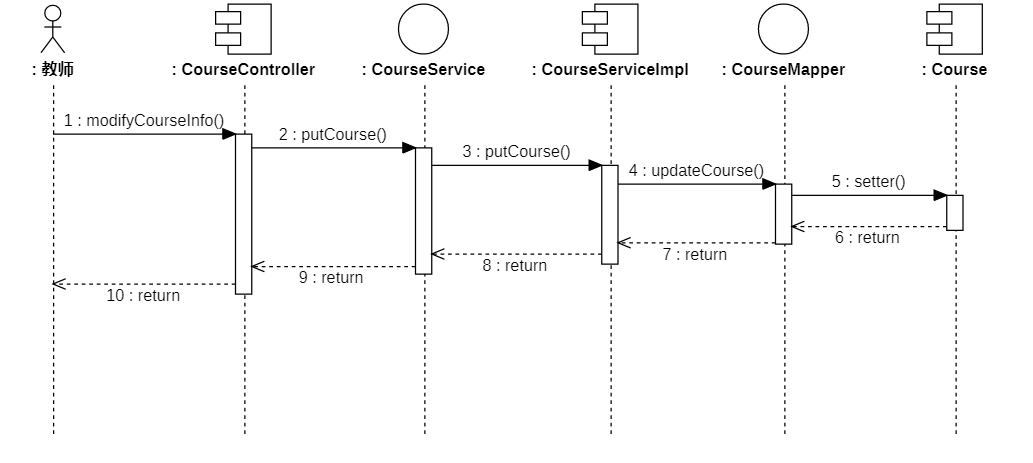


**4.2.2.3 modifyCourseInfo**

**4.2.2.3.1 操作简述**

该操作对应教师修改已有实验课程的信息，调用Controller中对外的modifyCourseInfo方法后，会调用业务层Service中的putCourse对传入的课程信息封装为课程对象，接着调用DAO层Mapper中的updateCourse完成对课程信息的更新。

**4.2.2.3.2 时序图**

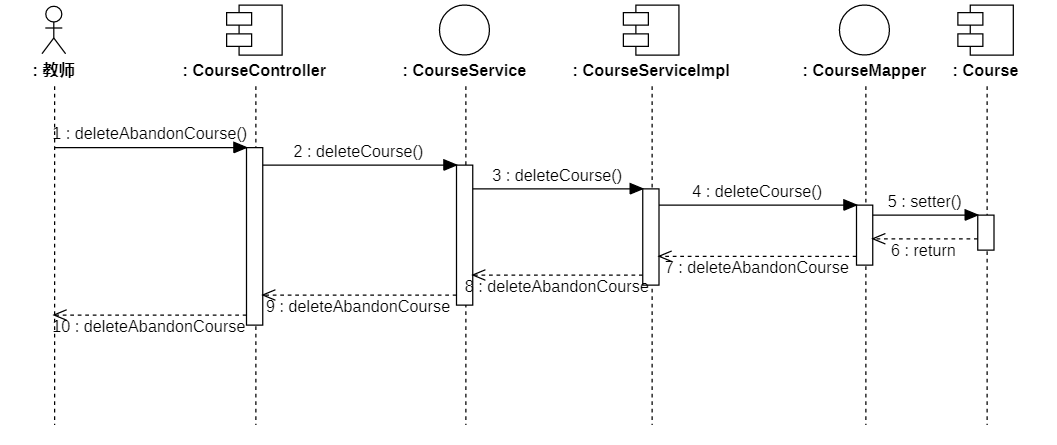


**4.2.2.4 deleteAbandonCourse**

**4.2.2.4.1 操作简述**

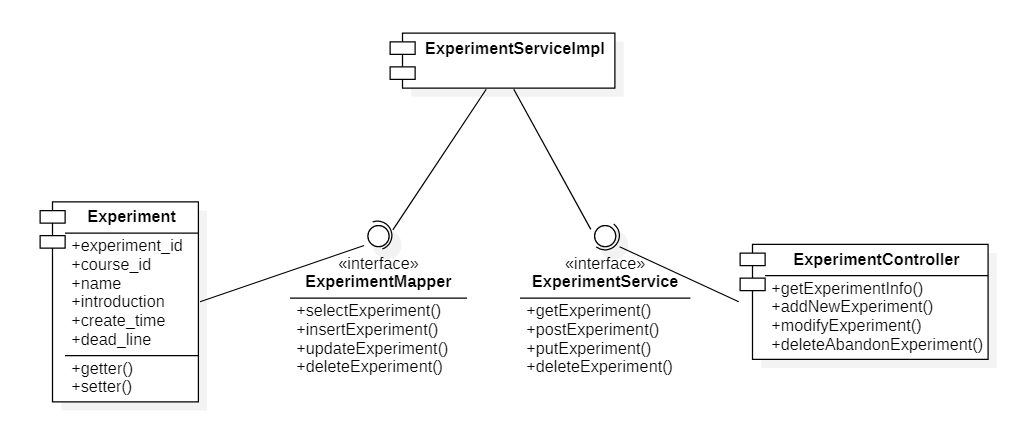
该操作对应教师删除已经废弃的实验课程，调用Controller中对外的deleteAbandonCourse方法后，会调用业务层Service中的deleteCourse获取要删除的实验课程的ID，接着调用DAO层Mapper中的deleteCourse根据指定的课程ID完成课程信息在数据库中的删除。

**4.2.2.4.2 时序图**



**4.3 实验管理构件**

**4.3.1 构件图**



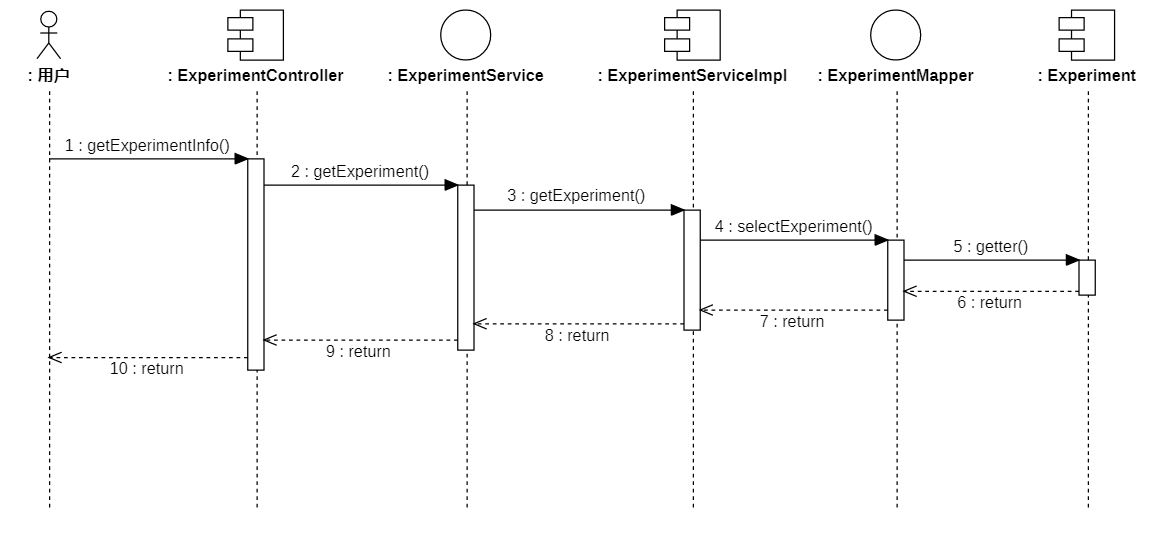
**4.3.2 核心操作**

**4.3.2.1 getExperimentInfo**

**4.3.2.1.1 操作简述**

该操作对应用户获取实验项目信息，调用Controller中对外的getExperimentInfo方法后，会调用业务层Service中的getExperiment获取传入的实验项目ID，接着调用DAO层Mapper中的selectExperiment根据实验项目ID在数据库中查找到对应的实验信息并返回。

**4.3.2.1.2 时序图**



**4.3.2.2 addNewExperiment**

**4.3.2.2.1 操作简述**

该操作对应教师添加新的实验项目，调用Controller中对外的addNewExperiment方法后，会调用业务层Service中的postExperiment对传入的实验信息封装为实验对象，接着调用DAO层Mapper中的insertExperiment完成实验项目信息在数据库中的插入。

**4.3.2.2.2 时序图**

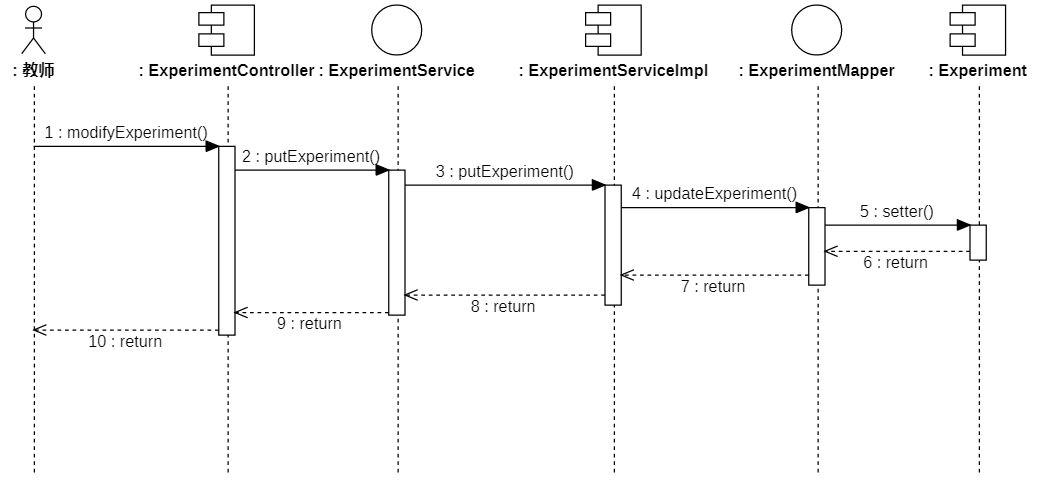


**4.3.2.3 modifyExperiment**

**4.3.2.3.1 操作简述**

该操作对应教师修改已有实验项目的信息，调用Controller中对外的modifyExperimentInfo方法后，会调用业务层Service中的putExperiment对传入的实验信息封装为实验对象，接着调用DAO层Mapper中的updateExperiment完成对实验项目信息的更新。

**4.3.2.3.2 时序图**

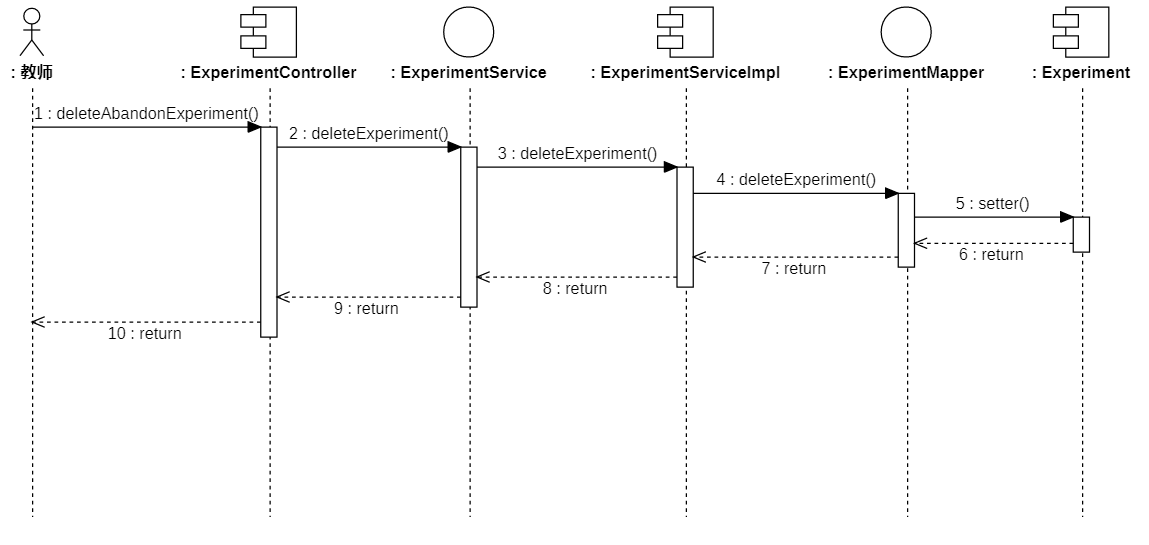


**4.3.2.4 deleteAbandonExperiment**

**4.3.2.4.1 操作简述**

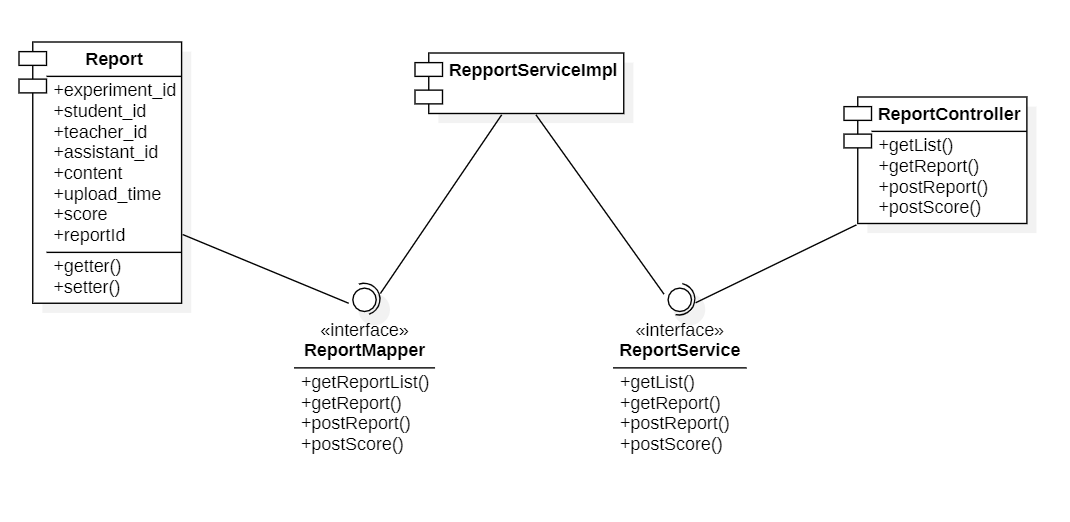
该操作对应教师删除已经废弃的实验项目，调用Controller中对外的deleteAbandonExperiment方法后，会调用业务层Service中的deleteExperiment获取要删除的实验项目的ID，接着调用DAO层Mapper中的deleteExperiment根据指定的实验ID完成实验项目信息在数据库中的删除。

**4.3.2.4.2 时序图**



**4.4 实验报告构件**

**4.4.1 构件图**



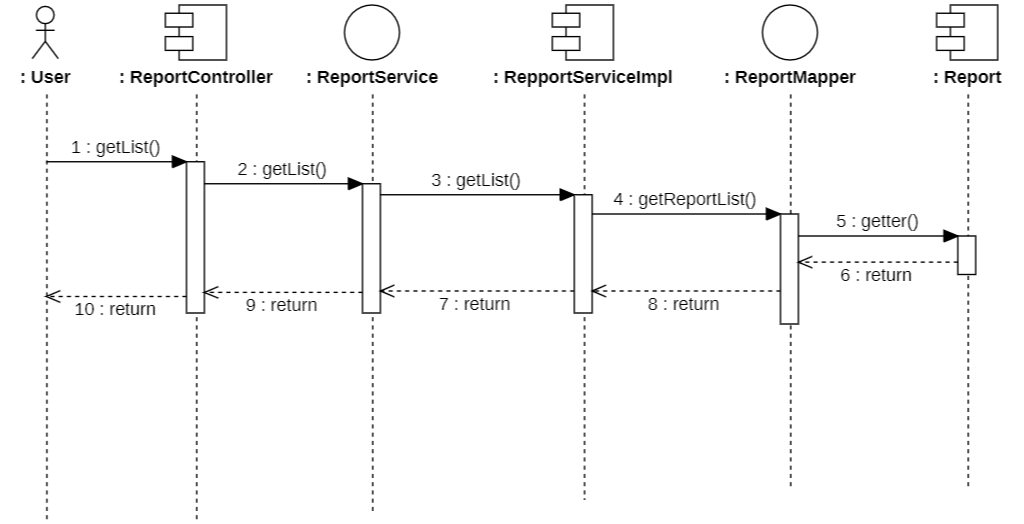
**4.4.2 核心操作**

**4.4.2.1 getList**

**4.4.2.1.1 操作说明**

该操作用于获取学生提交的实验报告列表，在controller层中触发该操作后，getList()方法会调用service层中的getList()方法，getList()方法检查参数合法性调用DAO层中的getReportList()方法访问数据库，将查询结果返回给上一层，在controller层中将结果封装为json返回给前端。

**4.4.2.1.2 时序图**

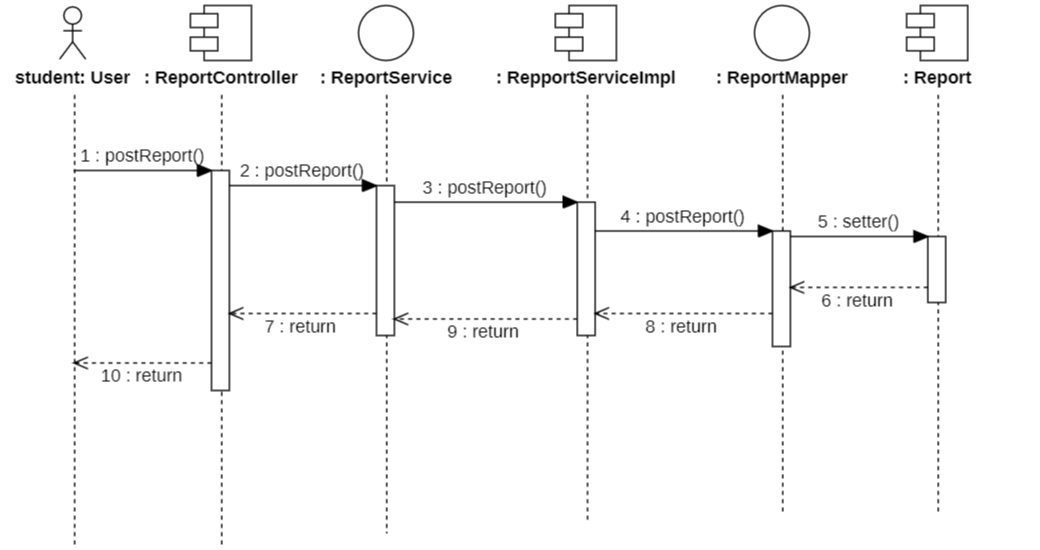


**4.4.2.2 postReport**

**4.4.2.2.1 操作说明**

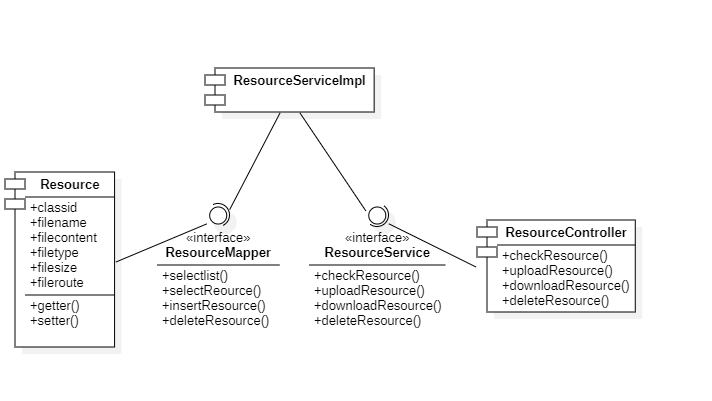
该操作用于学生提交实验报告，controller层从前端请求方法中获取到Report实体，依次向下传递给service层的postReport()，在service层中对字段进行非空检查再传递到DAO层中，DAO层中的postReport()方法访问数据库，对该Report进行持久化。

**4.4.2.2.2 时序图**



**4.5 课程文件构件**

**4.5.1 构件图**



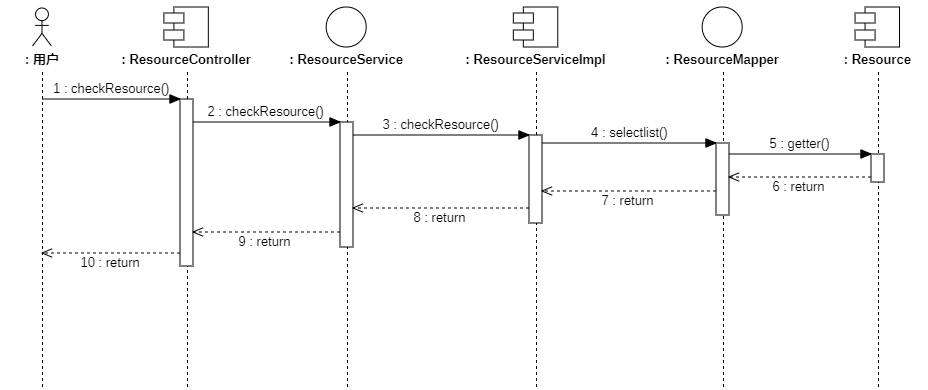
**4.5.2 核心操作**

**4.5.2.1 checkResource**

**4.5.2.1.1操作简述**

该操作对应老师或学生在课程页面中查看课程包含的文件，调用Controller中对外的checkResource方法后，会调用业务层Service中的checkResource对传入的课程信息进行封装，接着调用DAO层Mapper中的selectResource完成对课程文件的查询，并返回文件的信息列表json文件。

**4.5.2.1.2时序图**

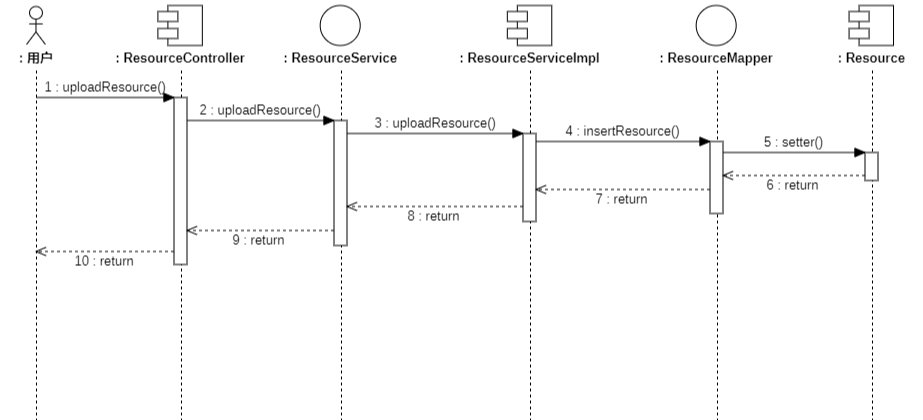


**4.5.2.2 uploadResource**

**4.5.2.2.1操作简述**

该操作对应老师在课程文件页面中上传本地文件，调用Controller中对外的uploadResource方法后，会调用业务层Service中的uploadResource对传入的文件信息封装为课程文件对象，接着调用DAO层Mapper中的insertResource完成对课程文件的插入。

**4.5.2.2.2时序图**

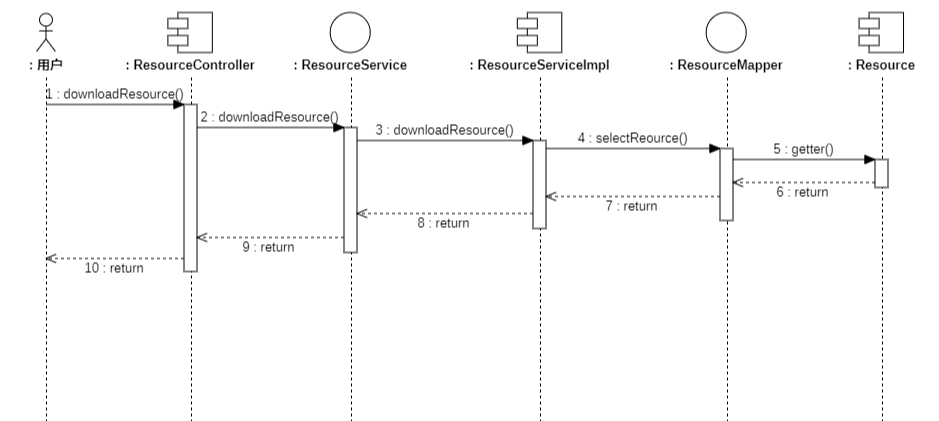


**4.5.2.3 downloadResource**

**4.5.2.3.1操作简述**

该操作对应老师或学生在课程文件页面中上传本地文件，调用Controller中对外的downloadResource方法后，会调用业务层Service中的downloadResource对传入的文件信息封装为文件对象，接着调用DAO层Mapper中的selectResource完成对课程文件的查询，并返回对应的课程文件。

**4.5.2.3.2时序图**

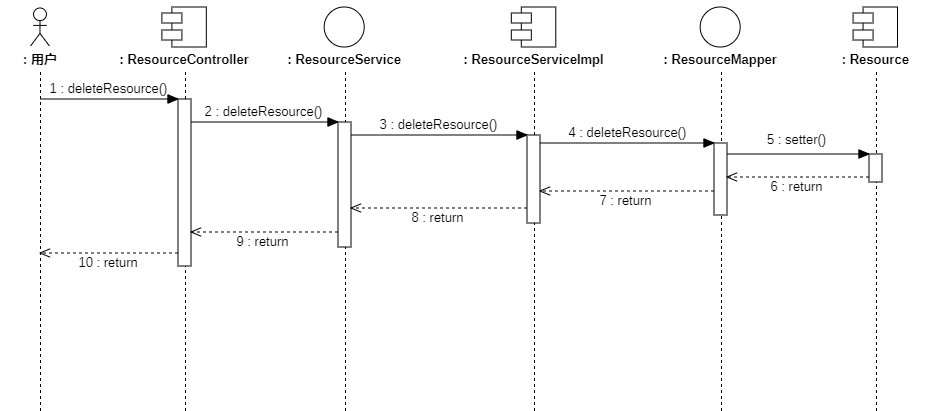


**4.5.2.4 deleteResource**

**4.5.2.4.1操作简述**

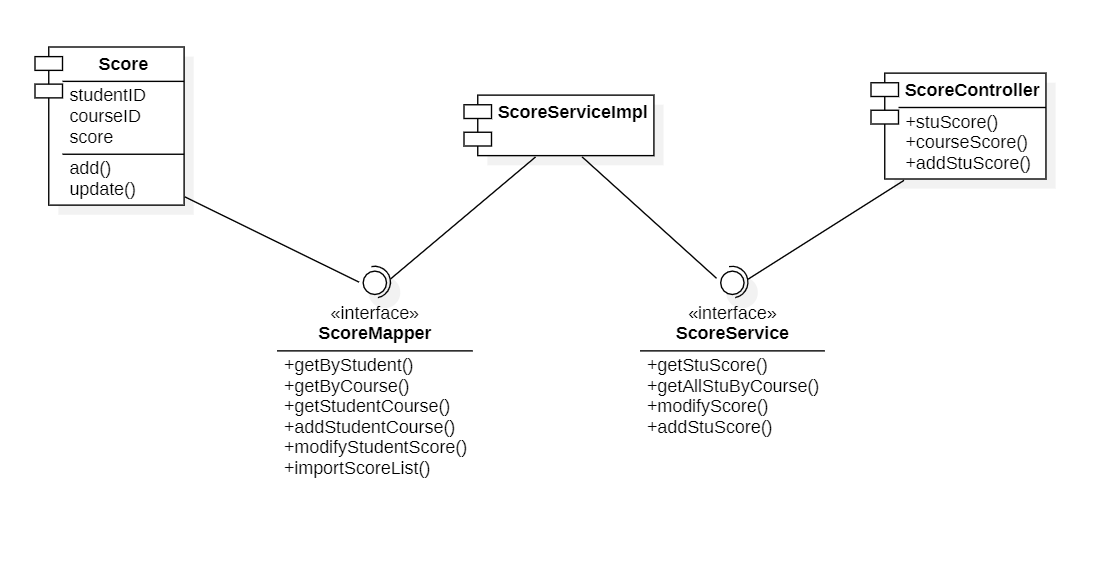
该操作对应老师在课程文件页面中删除服务器上的文件，调用Controller中对外的deleteResource方法后，会调用业务层Service中的deleteResource对传入的文件信息封装为文件对象，接着调用DAO层Mapper中的deleteResource完成对课程文件的删除。

**4.5.2.4.2时序图**



**4.6 学生成绩构件**

**4.6.1 构件图**



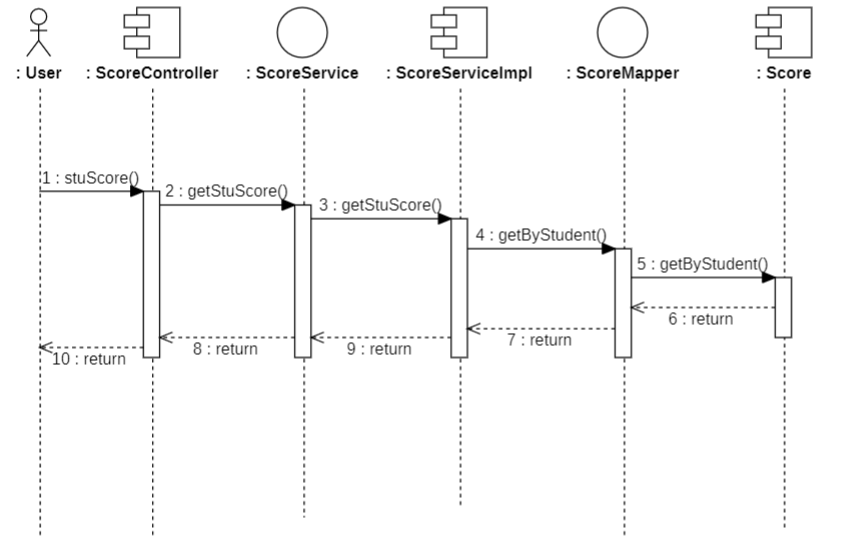
**4.6.2 核心操作**

**4.6.2.1 stuScore**

**4.6.2.1.1 操作简述**

该操作用于获取学生各已修读课程的成绩，当前端请求该接口时，stuScore()方法会调用service层中的getStuScore()方法，service层中getStuScore方法去调用DAO层中getByStudent()方法去访问数据库，获取学生成绩逐层返回到stuScore()方法中，以json形式返回给前端。

**4.6.2.1.2 时序图**

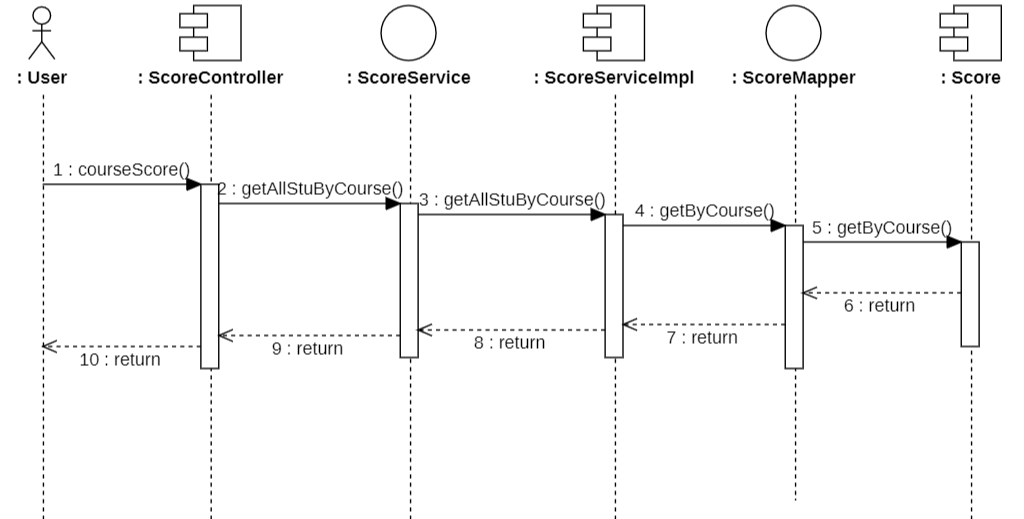


**4.6.2.2 courseScore**

**4.6.2.2.1 操作描述**

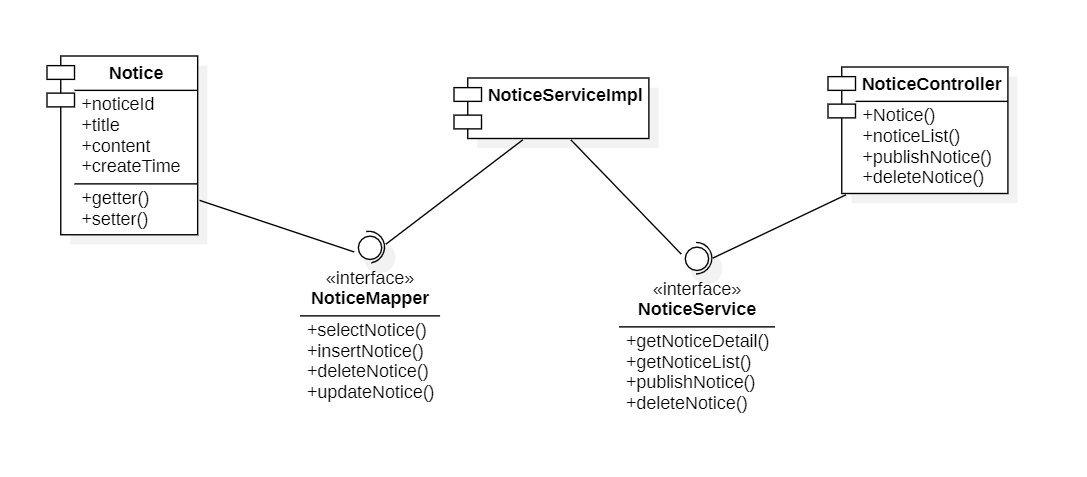
该操作是获取课程内所有学生的成绩，仅有相应的授课教师可调用，前端请求相应接口触发该操作后，controller层中的courseScore方法，该方法继而调用service层中的getAllStuByCourse()方法，该方法在检查参数之后会调用DAO层中的getByCourse()方法访问数据库获取相应数据返回给上一层，在getAllStuByCourse()方法中对结果封装为Map，返回给controller层，再以json形式返回给前端。

**4.6.2.2.2 时序图**



**4.7 通知公告构件**

**4.7.1 构件图**



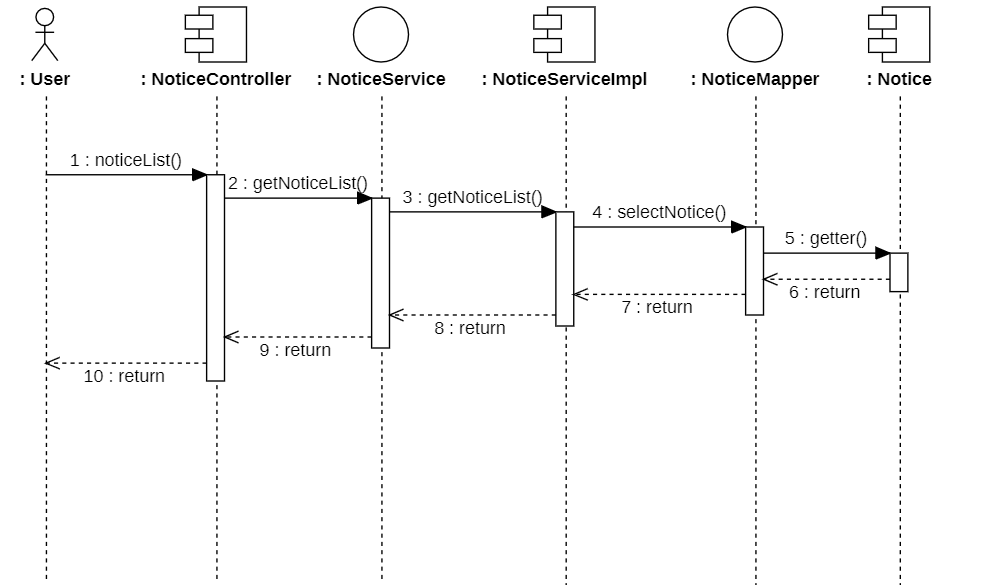
**4.7.2 核心操作**

**4.7.2.1 noticeList**

**4.7.2.1.1 操作简述**

该操作是获取当前课程中的所有通知公告，通过parameter可以实现分页展示，用户交互界面请求该API接口时，系统匹配该方法，该方法首先调用在service层中getNoticeList()方法，getNoticeList()方法会首先检查所给参数，然后调用DAO层中selectNotice()方法从数据库中查询结果，getNoticeLIst()方法对查询结果封装为Map，在Controller层中由noticeList()方法对结果封装为json返回给请求端。

**4.7.2.1.2 时序图**

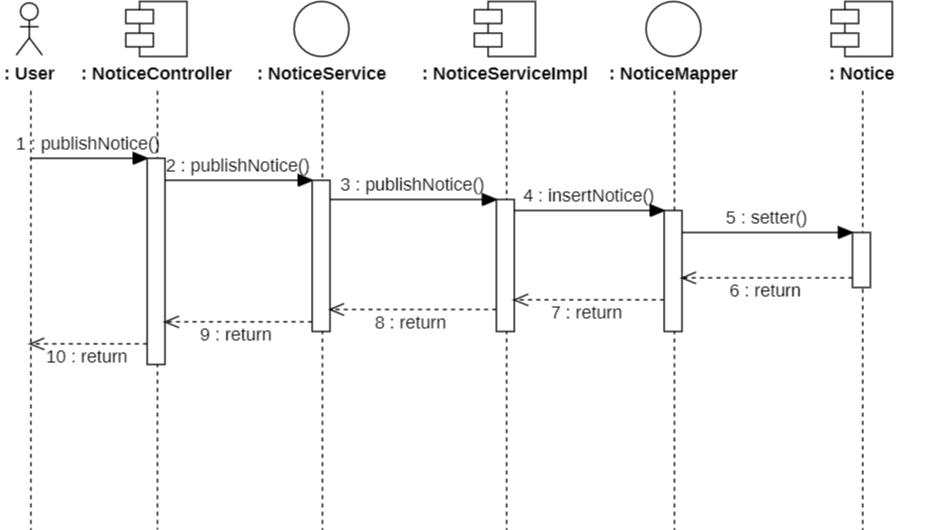


**4.7.2.2 publishNotice**

**4.7.2.2.1 操作简述**

该操作用于教师发布通知公告，前端请求相应URL接口，匹配到该操作后，controller层中的publishNotice()方法会接收到前端传送的Notice对象，传递给service层中的publishNotice()方法，service层中的该方法会对相应字段进行合法性检查，检查无误后，将其传递给DAO层中insertNotice方法()，该方法对Notice对象进行持久化。

**4.7.2.2.2 时序图**

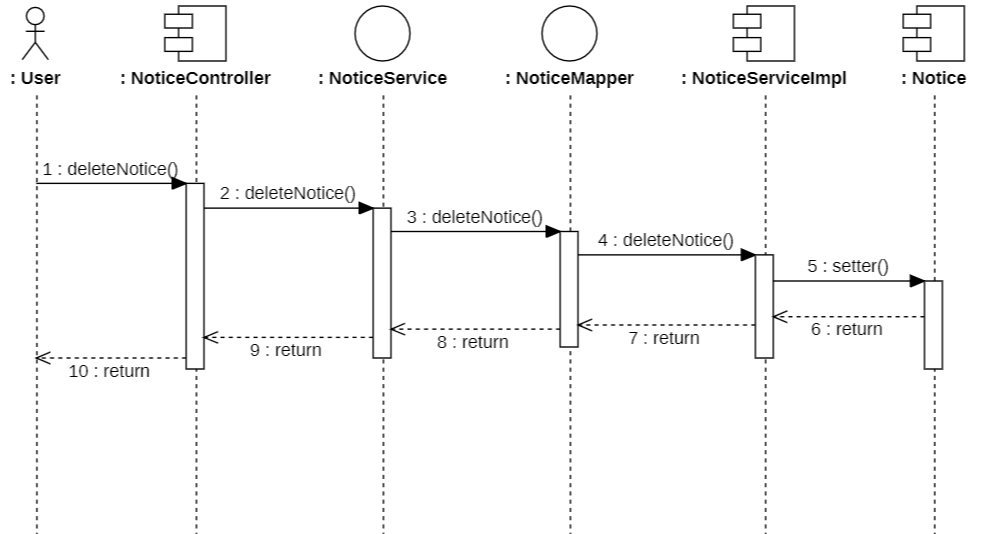


**4.7.2.3 deleteNotice**

**4.7.2.3.1 操作简述**

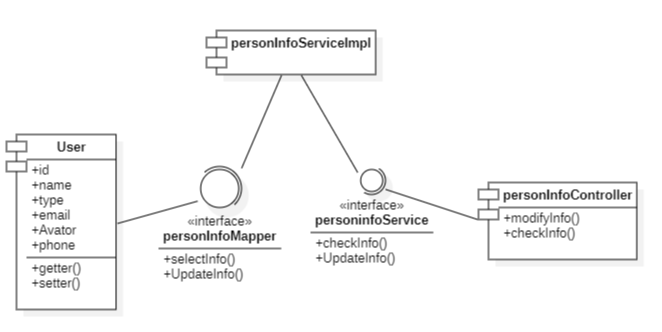
该操作用于删除已发布的通知公告，在controller层中触发该操作后，deleteNotice()方法会调用service层中的deleteNotice()方法，在service层中完成基本的业务逻辑，然后调用DAO层中的deleteNotice()方法对数据库中的数据进行删除。

**4.7.2.3.2 时序图**



**4.8 个人信息构件**

**4.8.1 构件图**



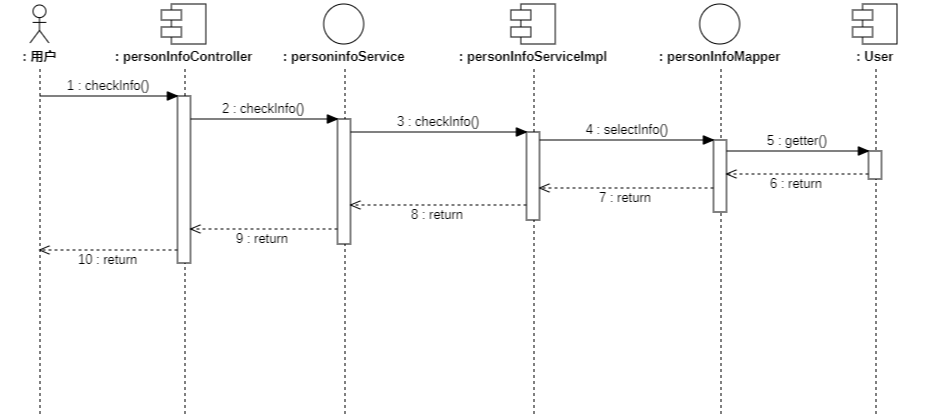
**4.8.2 核心操作**

**4.8.2.1 checkInfo**

**4.8.2.1.1操作简述**

该操作对应用户对自己的个人信息进行查询，调用Controller中对外的checkInfo方法后，会调用业务层Service中的checkInfo对传入的用户信息封装为用户对象，接着调用DAO层Mapper中的selectInfo完成用户信息在数据库中的查询，并返回json格式的用户信息。

**4.8.2.1.2时序图**

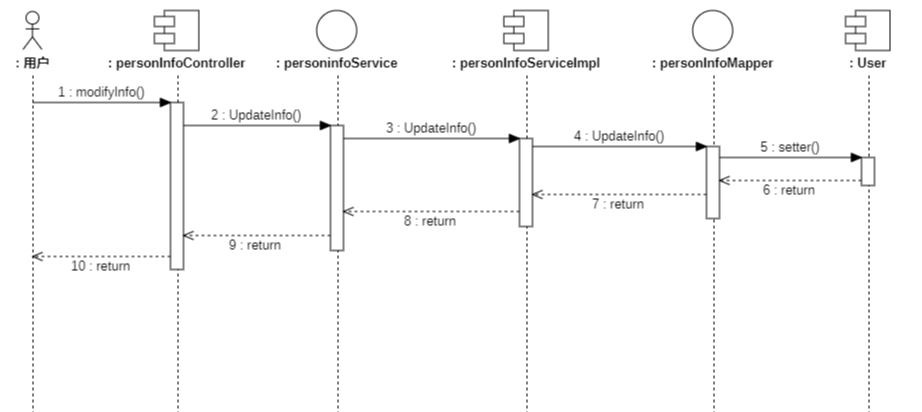


**4.8.2.2 modifyInfo**

**4.8.2.2.1操作简述**

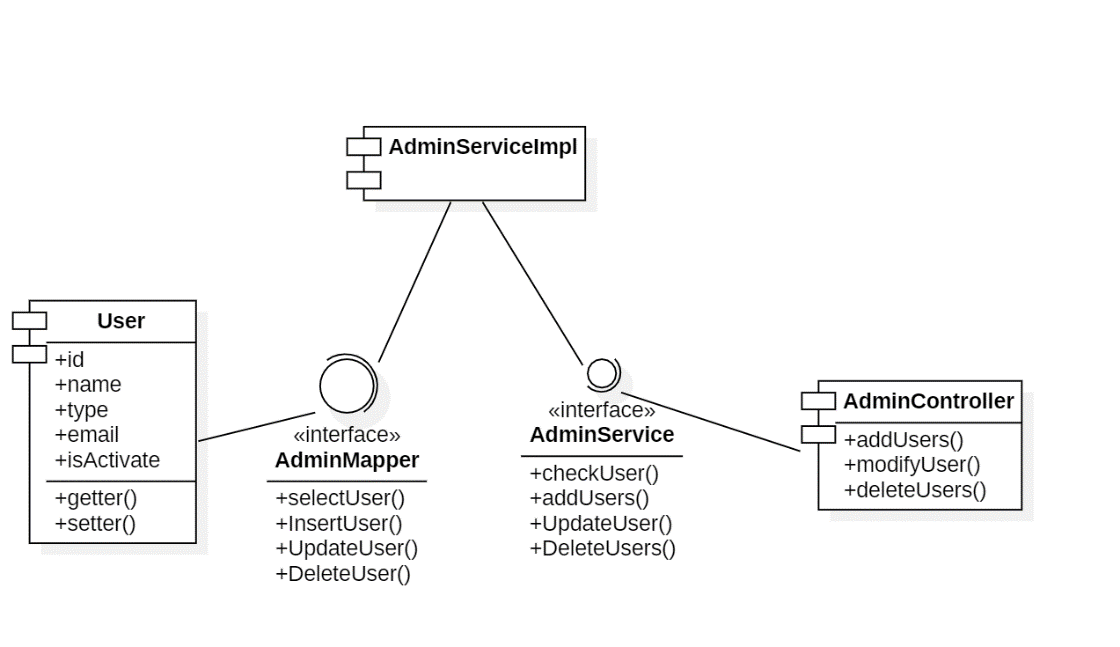
该操作对应用户对自己的个人信息进行修改，调用Controller中对外的modifyInfo方法后，会调用业务层Service中的UpdateInfo对传入的用户信息封装为用户对象，接着调用DAO层Mapper中的UpdateInfo完成用户信息在数据库中的查询，并返回json格式的用户信息。

**4.8.2.2.2时序图**



**4.9 管理员构件**

**4.9.1 构件图**



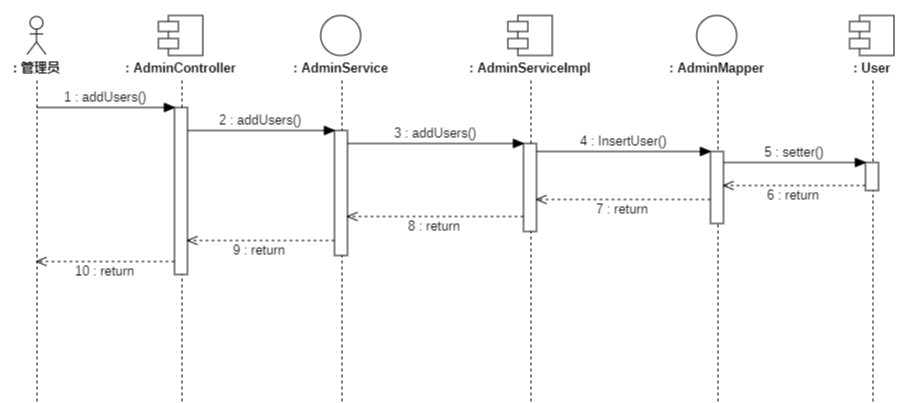
**4.9.2 核心操作**

**4.9.2.1 addUsers**

**4.9.2.1.1操作简述**

该操作对应管理员添加新的系统用户（包括学生和老师），调用Controller中对外的addUsers方法后，会调用业务层Service中的addUsers对传入的用户信息表格封装为多个用户对象，接着调用DAO层Mapper中的insertUser完成用户信息在数据库中的插入。

**4.9.2.1.2时序图**

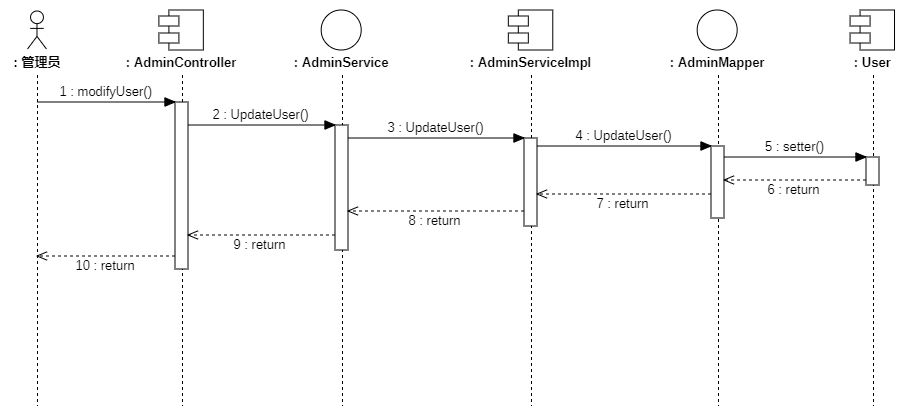


**4.9.2.2 modifyUser**

**4.9.2.2.1操作简述**

该操作对应管理员修改系统用户信息，包括个人信息和账户状态，调用Controller中对外的modifyUsers方法后，会调用业务层Service中的UpdateUser对传入的用户信息封装为用户对象，接着调用DAO层Mapper中的UpdateUser完成用户信息在数据库中的更新。

**4.9.2.2.2时序图**

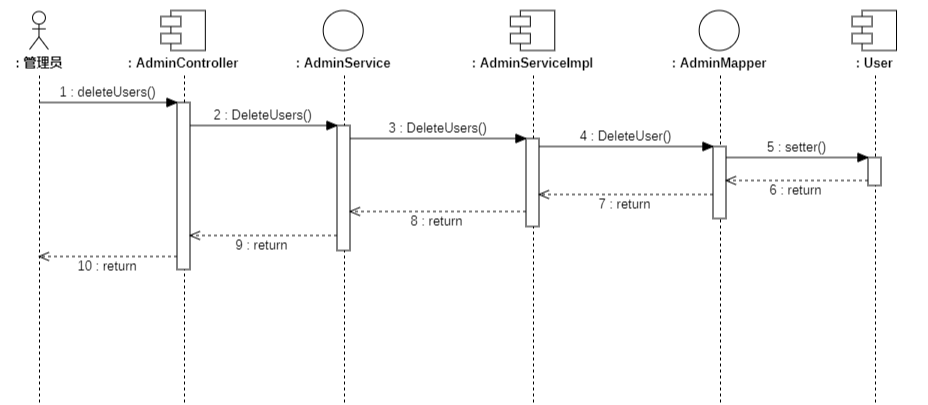


**4.9.2.3 deleteUsers**

**4.9.2.3.1操作简述**

该操作对应管理员注销系统用户，调用Controller中对外的deleteUsers方法后，会调用业务层Service中的DeleteUsers对传入的用户信息表格文件封装为多个用户对象，接着调用DAO层Mapper中的DeleteUser完成用户在数据库中的注销。

**4.9.2.3.2时序图**



**5 项目代码**

前端仓库：[happy-someone-wang/EMS-front (github.com)](https://github.com/happy-someone-wang/EMS-front)

后端仓库：[happy-someone-wang/EMS-back (github.com)](https://github.com/happy-someone-wang/EMS-back)