****

**宿舍管理系统**

设计模型

小组成员： 雷泓 1851202 王文政1853928

肖博阳1852672 张明哲1852452

指导教师： 孙萍

专 业： 软件工程

同济大学软件学院

School of Software Engineering, Tongji University

目录

[一、简介 3](#_Toc44003008)

[1.1 项目简介 3](#_Toc44003009)

[1.2 项目进度 4](#_Toc44003010)

[1.3 项目工具使用介绍 4](#_Toc44003011)

[1.3.1 UML建模工具 4](#_Toc44003012)

[1.3.2 Scrum开发过程 5](#_Toc44003013)

[二、用例模型及更新 6](#_Toc44003014)

[2.1 用例图 6](#_Toc44003015)

[2.2 用例模型 6](#_Toc44003016)

[三、设计模型 8](#_Toc44003017)

[3.1体系结构改进 8](#_Toc44003018)

[3.2用例实现示例 10](#_Toc44003019)

[3.2.1 宿舍人员信息查询 10](#_Toc44003020)

[3.2.2 注册 12](#_Toc44003021)

[3.2.3 缴纳寝室费用 13](#_Toc44003022)

[3.2.4 宿舍设施报修 15](#_Toc44003023)

[3.2.5 信息更新与检索 16](#_Toc44003024)

[3.2.6 行为检测 18](#_Toc44003025)

[3.3子系统设计 20](#_Toc44003026)

[3.3.1 数据库系统 20](#_Toc44003027)

[3.3.2 学生信息系统交互 20](#_Toc44003028)

[3.3.3 检查学生学号 21](#_Toc44003029)

[3.3.4 分派维修任务系统 21](#_Toc44003030)

[3.3.5 设施检查子系统 23](#_Toc44003031)

[3.3.6 快递快件查询子系统 25](#_Toc44003032)

[3.4 类设计 25](#_Toc44003033)

[3.4.1 ReportData（报表信息）类from财务统计系统 25](#_Toc44003034)

[3.4.2 OrderContorller（订单控制）类from 宿舍设施报修 27](#_Toc44003035)

[3.4.3 AssetDataForm（资产数据）类from 宿舍资产管理 28](#_Toc44003036)

[3.4.4 MachineManager（机器管理员）类from宿舍自动贩卖机货物整理 28](#_Toc44003037)

[3.4.5 GoodsData（货物信息）类from宿舍自动贩卖机货物整理 30](#_Toc44003038)

[3.4.6 PersonalInformation（学生个人信息）类 from 学生信息管理 32](#_Toc44003039)

[3.4.7 RegisterControl（注册控制）类 from 注册 33](#_Toc44003040)

[3.4.8 StudentLeaveInformation(学生离校信息)类 from 学生长时间离宿 34](#_Toc44003041)

[3.4.9 IssueAnnounceInteraction（发布通知交互）from发布公告通知 35](#_Toc44003042)

[3.4.10 QueryStudentInCampusControl(查询学生在校控制)from学生在校查询 36](#_Toc44003043)

[3.4.11 Payment(寝室费用账单)from缴纳寝室费用 36](#_Toc44003044)

[3.4.12 QueryPaymentInterFace (查询费用交互)from缴纳寝室费用 37](#_Toc44003045)

[3.4.13 FunctionRoom(功能室)from功能室申请 38](#_Toc44003046)

[3.4.14 Housemaster(宿管)from申请更换宿舍&功能室申请 39](#_Toc44003047)

[3.4.15 Mail(快递、快件)from快递快件查询 39](#_Toc44003048)

[四、组员分工 40](#_Toc44003049)

# 一、简介

## 1.1 项目简介

技术迅速发展的今天，为了方便工作生活各式各样的数据被保存下来以供之后的安排和统计。数据膨胀的时代，逼迫着高校以更加有效率和简便的方式来管理和组织这些数据。本系统以常见的高校宿舍管理系统为框架进行补充和改善，实现对宿舍人员和设备的有效管理。

本系统基于宿舍系统网络的架构，为学生、宿舍管理员和生活老师等提供便捷的服务，达到更高效的管理。主要功能有为宿舍学生提供查询及缴纳寝室费用，查询学生信息等常用服务，为宿舍管理员及生活老师提供宿舍信息查询、宿舍内设施检查保修以及查看设施信息服务，同时为数据库管理员提供清晰的接口以维护和更新数据库。

本系统分为7个子系统：登陆子系统；学生服务子系统；学生信息子系统；设施信息子系统；财务子系统；寝室成绩子系统；数据库管理子系统。

登陆子系统主要参与者为用户，该系统为数据库中已有用户提供登陆服务，验证其身份给予其使用服务权力，为未来用户提供账户注册服务。

学生服务子系统主要参与者为学生，该系统为学生提供一系列如查询缴纳费用、查看公告及申请更换宿舍等常见宿舍服务。

学生信息子系统主要参与者为宿舍管理员和生活老师，该系统提供了查询宿舍各方面信息、发布公告及学生晚归登记等一系列学生信息管理和登记服务，同时提供发布宿舍公告服务。

设施信息子系统主要参与者为宿舍管理员和生活老师，该系统提供了查询宿舍内各设施状态、功能室借用以及备用钥匙借用等宿舍内设施信息管理服务。

财务子系统主要参与者为财务人员，该系统提供了宿舍内总体开销费用和宿舍资产登记服务。

寝室成绩子系统主要参与者为学生、宿舍管理员和生活老师，该系统为学生和生活老师提供查询宿舍得分情况服务；为宿舍管理员提供各宿舍违规登记，卫生情况登记以及录入成绩等管理功能。

数据库管理子系统主要参与者为数据库管理员，该系统提供了系统配置功能，允许数据库管理员对数据库维护和更新，并进行异常行为检测等活动。

整个宿舍管理系统基本涵盖了学生及老师管理员在管理宿舍时需要进行的各项活动，在真实使用中能够方便使用者。如在网络上可以直接填写申请更换宿舍表，省去大部分与管理员和生活老师的对接工作，而管理员也只需查看系统中的信息，再与老师简单沟通即可完成。且通过数据库的管理和整合，学生和宿舍内各样的数据都能够方便快捷的取出进行统计。

## 1.2 项目进度

之前的文档已完成了项目的基本规划、用例图及用例规约、活动图、术语表 和补充规约、架构分析、分析机制、模型分析、类图及部分时序图和协作图。本文档将进行项目的体系结构改进、用例实现、类设计、子系统设计及本系统的功能和特点及其操作必须遵守的约束条件。本文档供系统的开发者和利益相关者阅读。

## 1.3 项目工具使用介绍

### 1.3.1 UML建模工具

Unified Modeling Language (UML)又称统一建模语言或标准建模语言，是一个支持模型化和软件系统开发的图形化语言，为软件开发的所有阶段提供模型化和可视化支持，包括由需求分析到规格，到构造和配置。UML规范用来描述建模的概念有，类(对象的)、对象、关联、职责、行为、接口、用例、包、顺序、协作，以及状态。UML从考虑系统的不同角度出发，定义了用例图、类图、对象图、状态图、活动图、序列图、协作图、构件图、部署图等9种图。这些图从不同的侧面对系统进行描述。系统模型将这些不同的侧面综合成一致的整体，便于系统的分析和构造。尽管UML和其它开发工具还会设计出许多派生的视图，但上述这些图和其它辅助性的文档是软件开发人员所见的最基本的构造。

在本宿舍管理系统的分析与设计中，我们使用得最多且最广泛的即UML建模，具体的软件工具为starUML。构造的图有：用例图、类图、对象图、状态图、活动图、序列图、协作图、构件图、部署图。用例图从用户角度描述系统功能，并指各功能的操作者，用于系统参与者、包含需要实现的用例以及用例与参与者之间关系的直观展示。类图描述系统中类的静态结构，用于模型分析。包图是包和类组成的，表示包与包之间的关系，用于描述系统的分层结构。行为图描述系统动态模型和对象组成的交换关系，包括状态图和活动图。活动图描述了业务实现用例的工作流程，用于展示用例的基本流与备选流。状态图是描述状态到状态控制流，用于动态特性建模。最后还有包括顺序图和合作图的交互图，用于描述对象之间的交互关系。顺序图展示对象之间的动态合作关系，强调对象发送消息的顺序，同时显示对象之间的交互。合作图描述对象之间的协助关系。

此外，还有一些架构的实现也在uml建模工具的使用得到体现。

### 1.3.2 Scrum开发过程

Scrum是迭代式增量软件开发过程，通常用于敏捷软件开发。Scrum是一个包括了一系列的实践和预定义角色的过程骨架(是一种流程、计划、模式，用于有效率地开发软件)。Scrum中的主要角色包括同项目经理类似的Scrum主管角色负责维护过程和任务，产品负责人代表利益所有者，开发团队包括了所有开发人员。Scrum开发过程的核心是冲刺，在每一次需求被分派下来后，团队会产生一个冲刺目标。在每一次冲刺，开发团队创建可用的软件目标增量，即本次冲刺需完成的任务。每一个冲刺所要实现的特性来自产品目标(product backlog，本处指作业任务)，产品目标是指按照优先级排列的需要完成的工作的概要的需求(目标)。最终形成产品目标、冲刺订单、燃尽图三个成果文档。

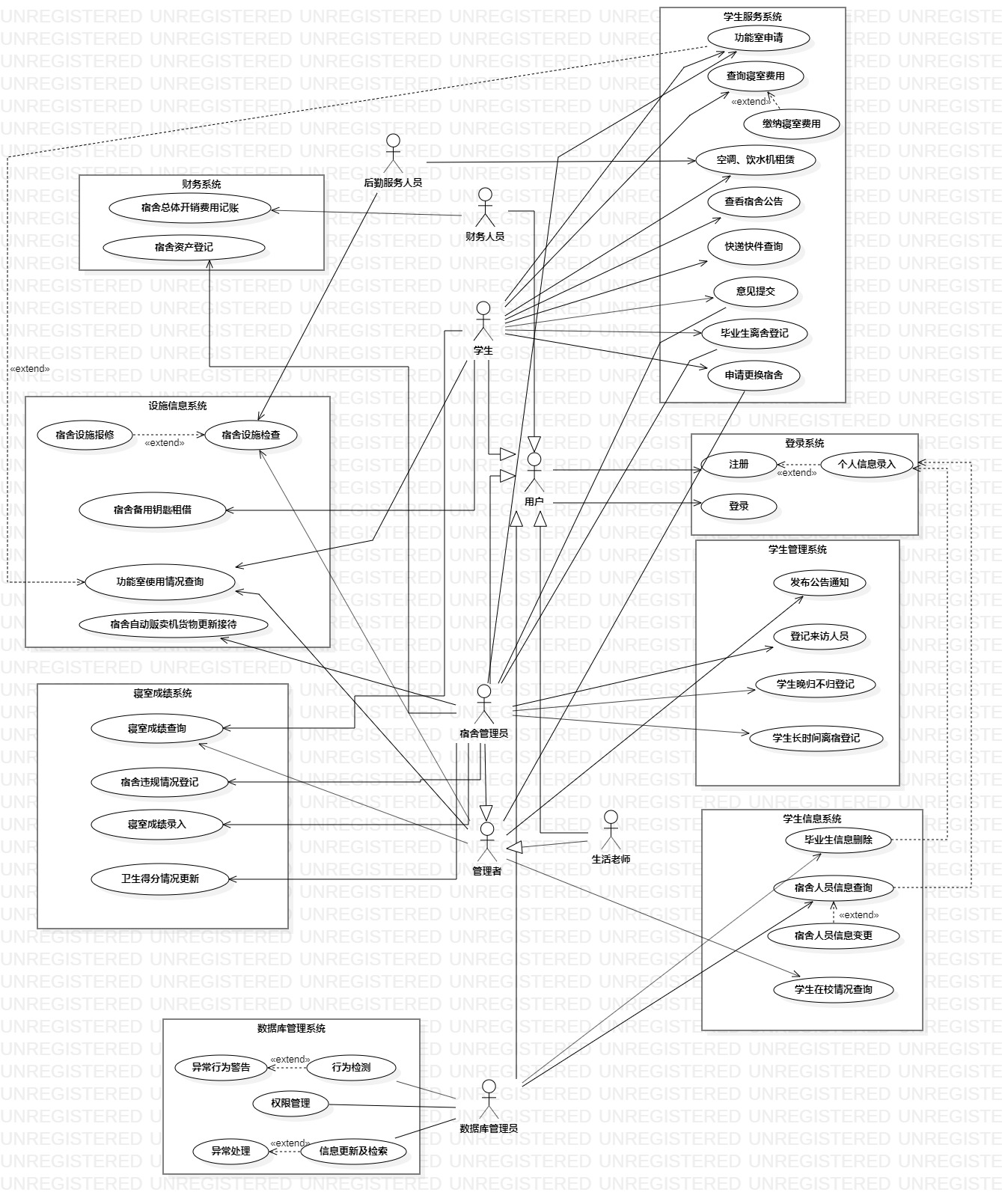
本次系统分析与设计中，我们首先接收每一次的作业信息，产生一个项目的概要文档（即产品目标），包括所有所需特性的粗略的描述。产品目标是关于将要创建的什么产品。产品目标是开放的，每位组员都可以编辑。产品目标包括粗略的估算，通常以天为单位，估算将帮助团队衡量时间表和优先级。

其次我们将制作冲刺目标（sprint backlog），细分化文档，包含团队如何实现下一个冲刺的需求的信息。任务被分解为以小时为单位，没有任务可以超过16个小时。如果一个任务超过16个小时，那么它就应该被进一步分解。冲刺目标上的任务不会被分派，而是由团队成员签名认领他们喜爱的任务。

当任务临近截至，团队会总结一幅燃尽图(burn down chart)，包含一个公开展示的图表，显示当前冲刺中未完成的任务数目，或在冲刺目标上未完成的目标项的数目，以便于团队更好地抓住工作重点。燃尽图可以使'冲刺(sprint)'平稳的覆盖大部分的迭代周期，且使项目仍然在计划周期内。

# 二、用例模型及更新

## 2.1 用例图



## 2.2 用例模型

对象：学生、宿舍管理员、生活老师、数据库系统管理员、服务提供人员

1.登录系统：

1）注册-<拓展>-信息录入

2）登录

2.学生服务系统：

1）查询寝室费用

2）缴纳寝室费用

3）空调、饮水机租贷

4）查看宿舍公告

5）快递快件查询

6）功能室申请

7）意见提交

8）毕业生离舍登记

9）申请更换宿舍

3.学生信息系统：

1）宿舍人员信息查询

2）宿舍人员信息变更管理

3）学生在校情况查询

4）来访人员登记

5）学生晚归/不归登记

6）学生请假/长时间离宿登记

7）毕业生信息删除

8）发布宿舍公告通知

4.设施信息系统：

1）宿舍设施检查

2）宿舍设施报修

3）宿舍备用钥匙租借登记

4）功能室使用情况查询

5）宿舍自动贩卖机货物更新接待

5.财务系统：

1）宿舍总体开销费用记账

2）宿舍资产登记

6.寝室成绩系统：

1）卫生情况管理

2）宿舍违规情况登记

3）寝室成绩导入

4）寝室成绩情况查询

7.数据库管理系统：

1）行为监测 -<拓展>-异常行为警告

2）系统配置

3）信息更新和检索 -<拓展>- 异常处理

由于项目开始时对系统功能用例考虑较为周全全面，暂时并未对系统用例模型进行增添、删除或修改等更新操作。

# 三、设计模型

## 3.1体系结构改进

本此设计模型中体系架构沿用了上次模型分析中的逻辑架构分析以及物理架构分析。逻辑架构采取常用的软件架构——分层模式为基础，分为用户界面层，业务服务层，中间件层以及数据库层；

**·用户界面层**：用户界面，负责视觉和用户互动

**·业务服务层**：实现业务逻辑

**·中间件层**：为业务服务提供数据等中间件

**·数据库层**：保存数据

这种架构将软件分成若干个水平层，每一层都有清晰的角色和分工，不需要知道其他层的细节。每一层都能够独立测试，其他层的接口通过模拟进行解决，使得整个系统具有高内聚低耦合特点。



本次体系结构改进主要体现在将类图中的设计元素进行分类，并分别根据属性、功能归类到对应的分层结构中，同时为一些操作设立了接口。本次改进将更有利于数据和操作的分离，在对操作功能进行调整的时候能尽可能少的牵扯其他有关操作，减少了函数修改时的复杂性和冗余性，使系统调用相关功能时尽可能使用接口，同时保证了当某一功能出现错误时不涉及其他功能的崩溃。

用户界面层包含：

|  |
| --- |
| 宿舍成绩查询界面 |
| 数据库管理界面 |
| 财务统计界面 |
| 宿舍资产管理界面 |
| 分派任务界面 |
| 订单界面 |
| 自动贩卖机货物整理界面 |

业务服务层包括：

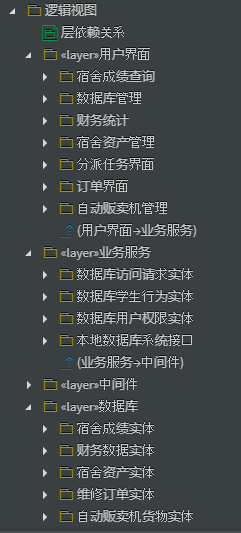
|  |
| --- |
| 数据库访问请求实体 |
| 数据库学生行为实体 |
| 数据库用户权限实体 |
| 本地数据库系统接口 |

中间件层包括：

|  |
| --- |
|  |

数据库层包括：

|  |
| --- |
| 宿舍成绩实体 |
| 财务数据实体 |
| 宿舍资产实体 |
| 维修订单实体 |
| 自动贩卖机货物实体 |



## 3.2用例实现示例

### 3.2.1 宿舍人员信息查询

设计类图



实现

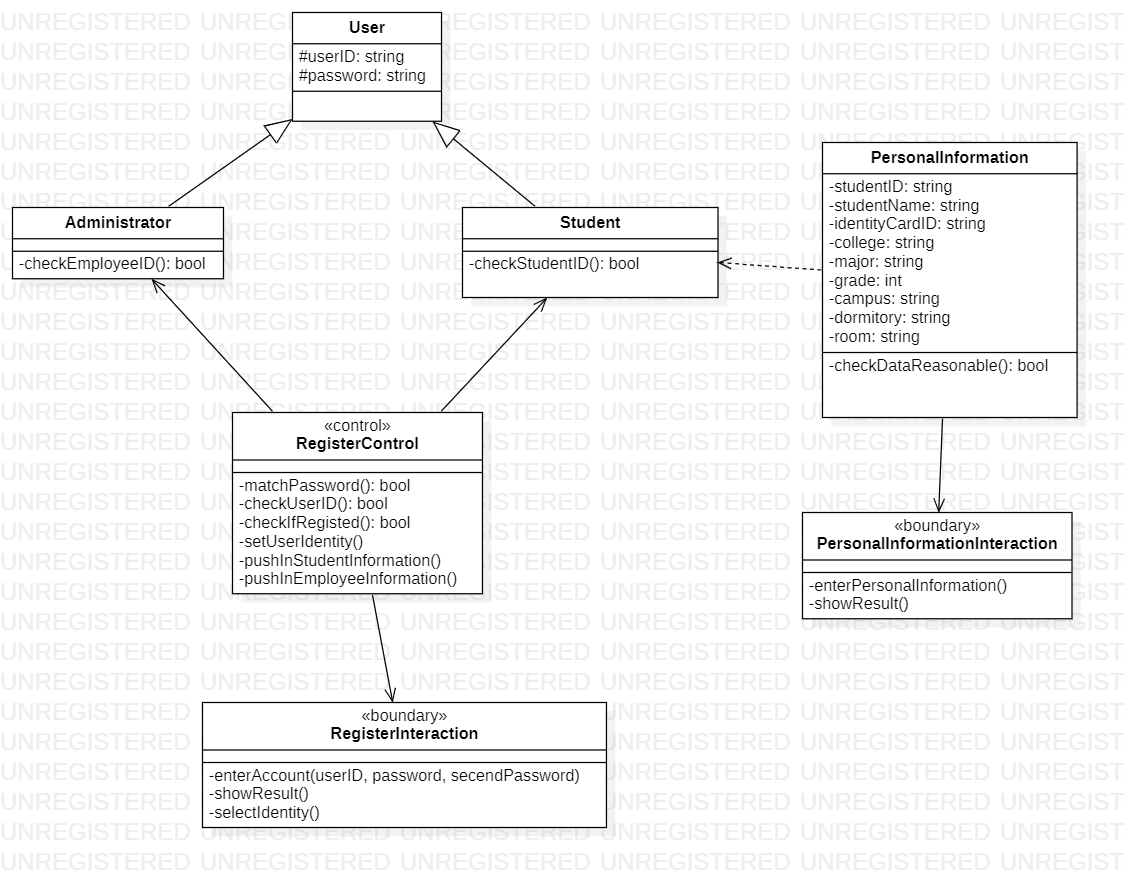


时序图



### 3.2.2 注册

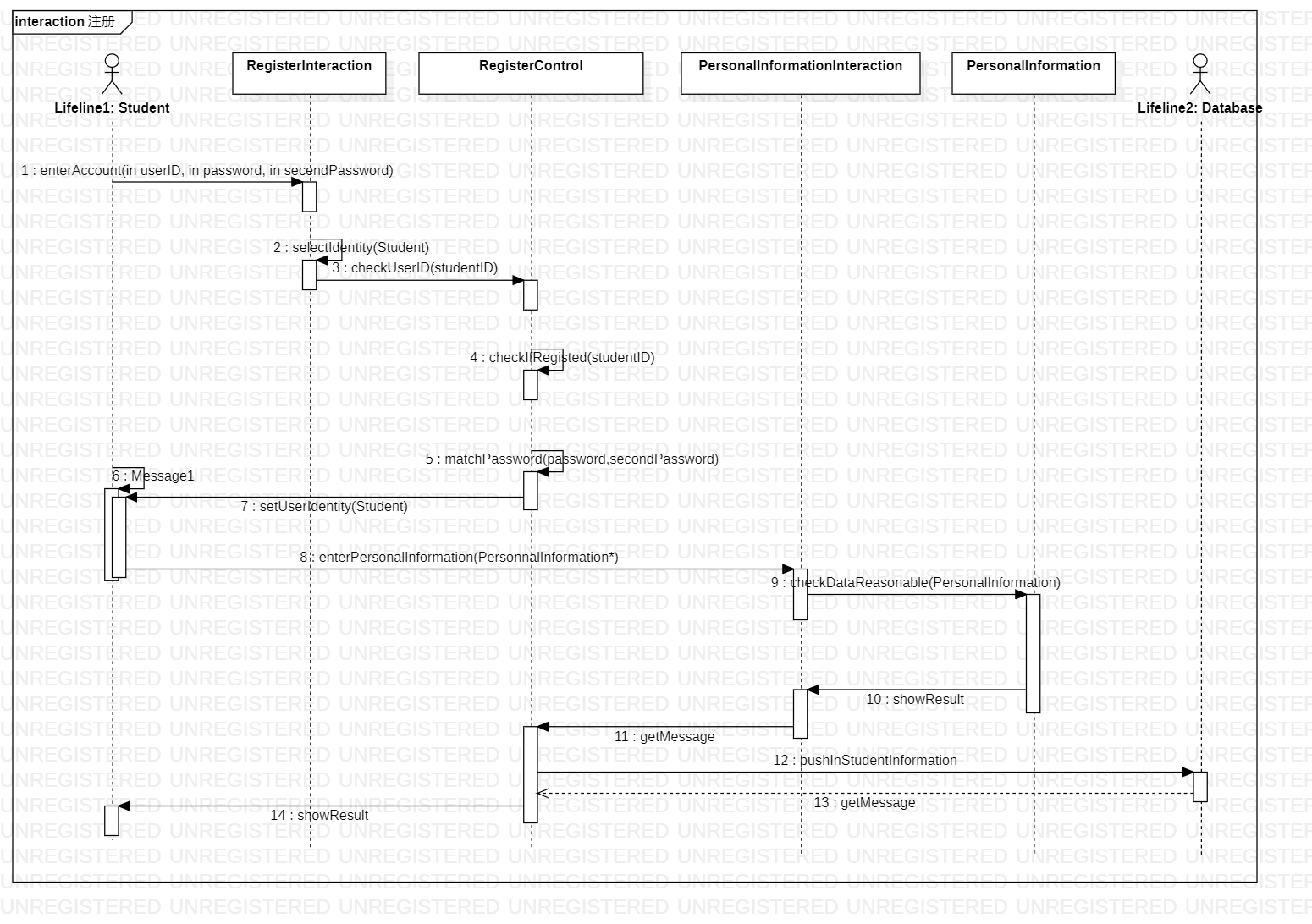
设计类图



实现

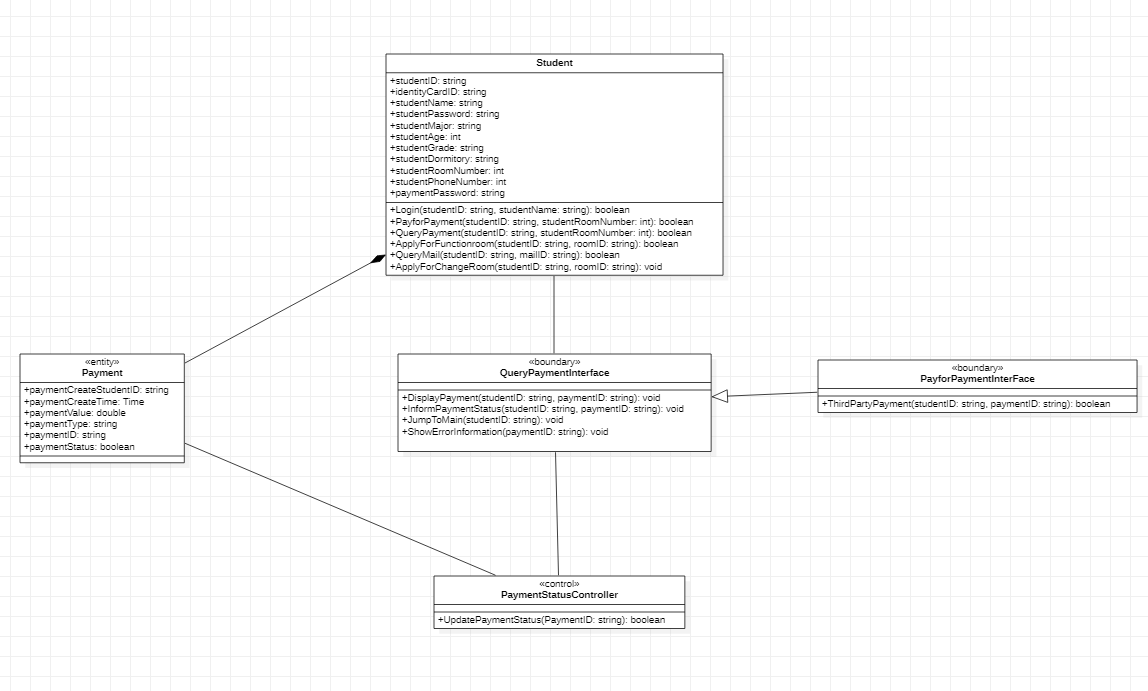


时序图

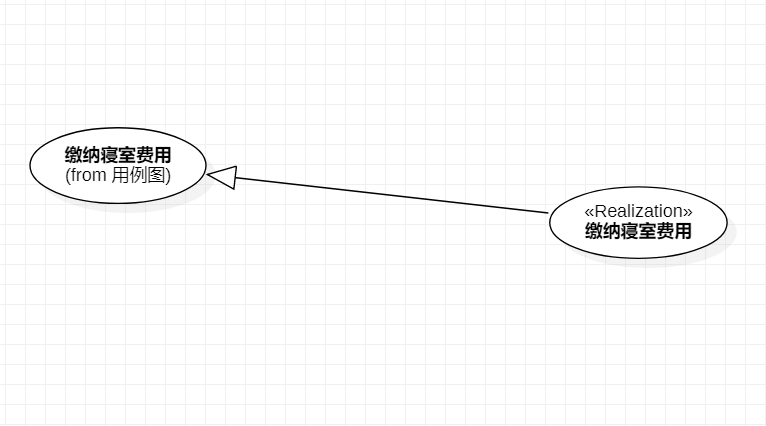


### 3.2.3 缴纳寝室费用

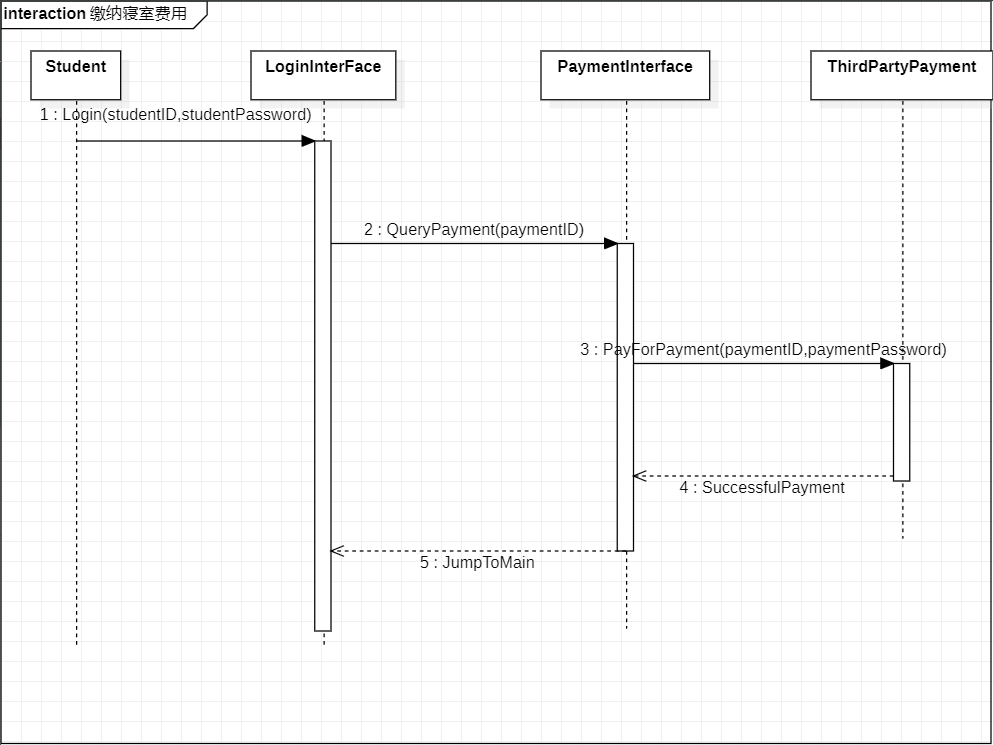
设计类图



实现

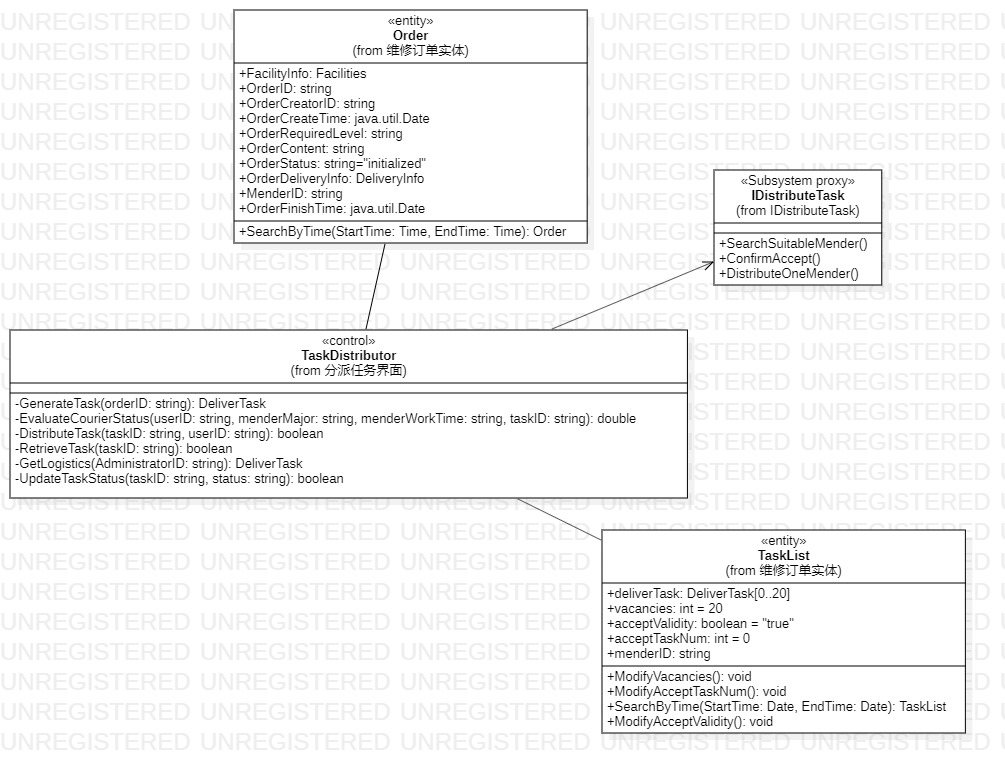


时序图



### 3.2.4 宿舍设施报修

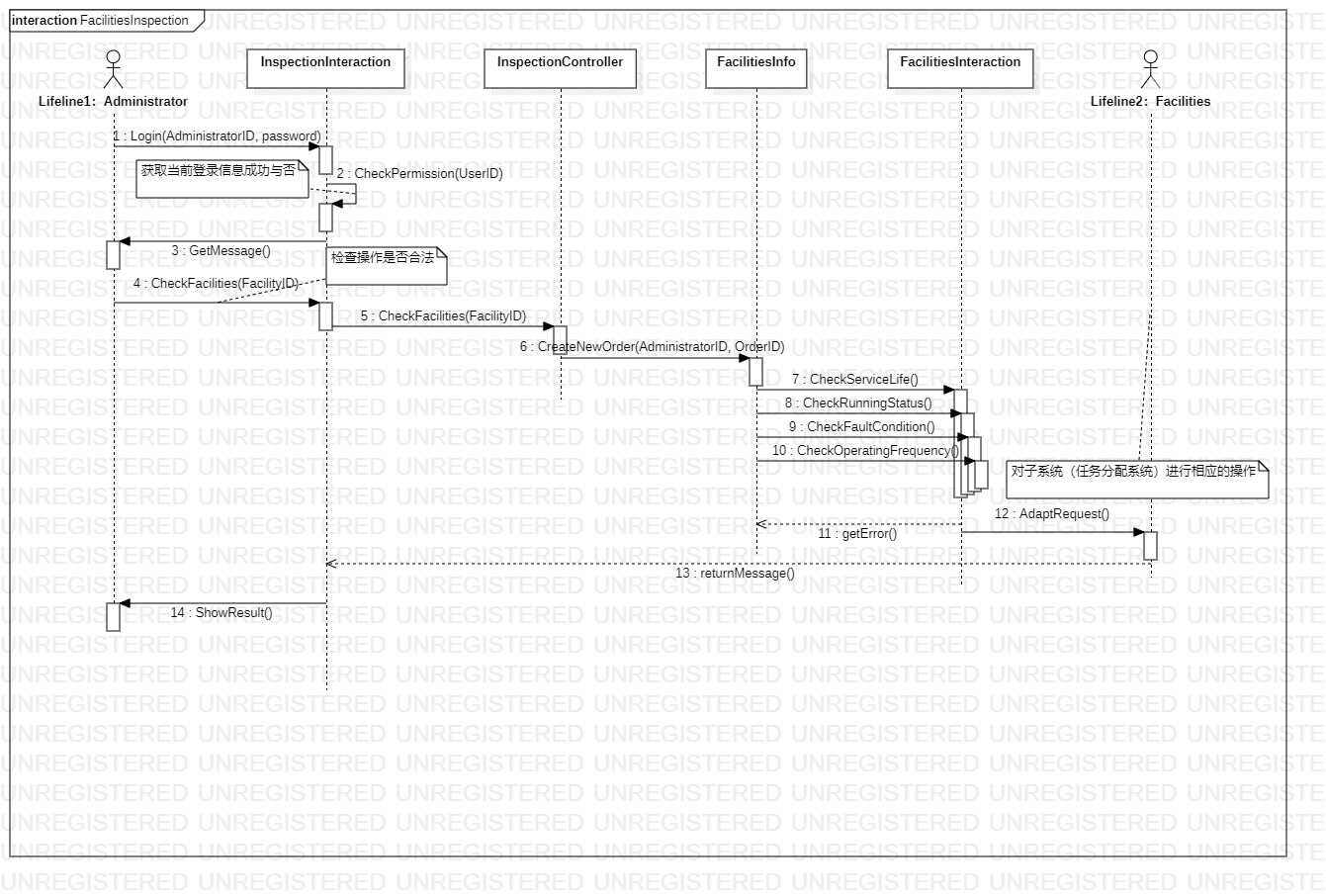
设计类图



实现



时序图

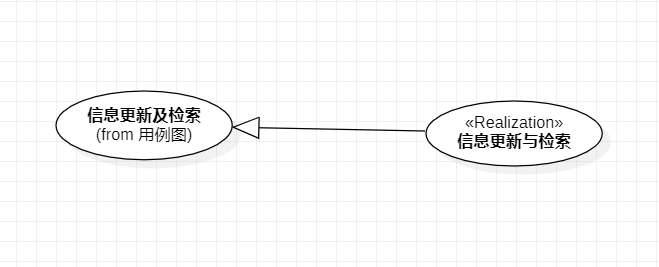


### 3.2.5 信息更新与检索

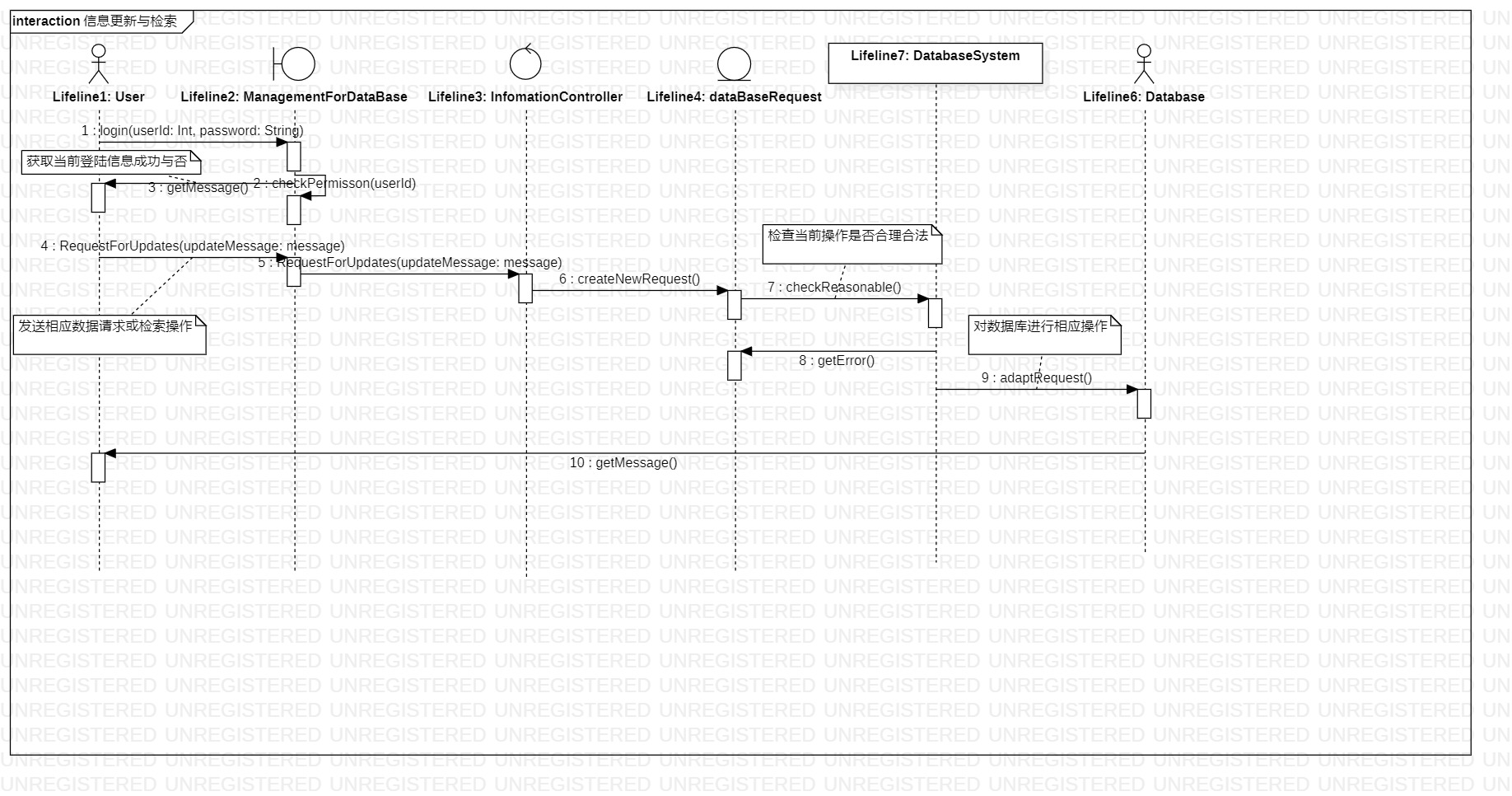
设计类图



实现



时序图

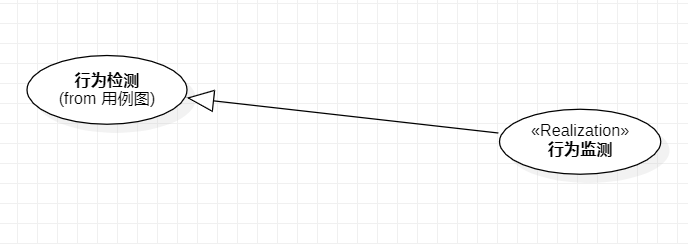


### 3.2.6 行为检测

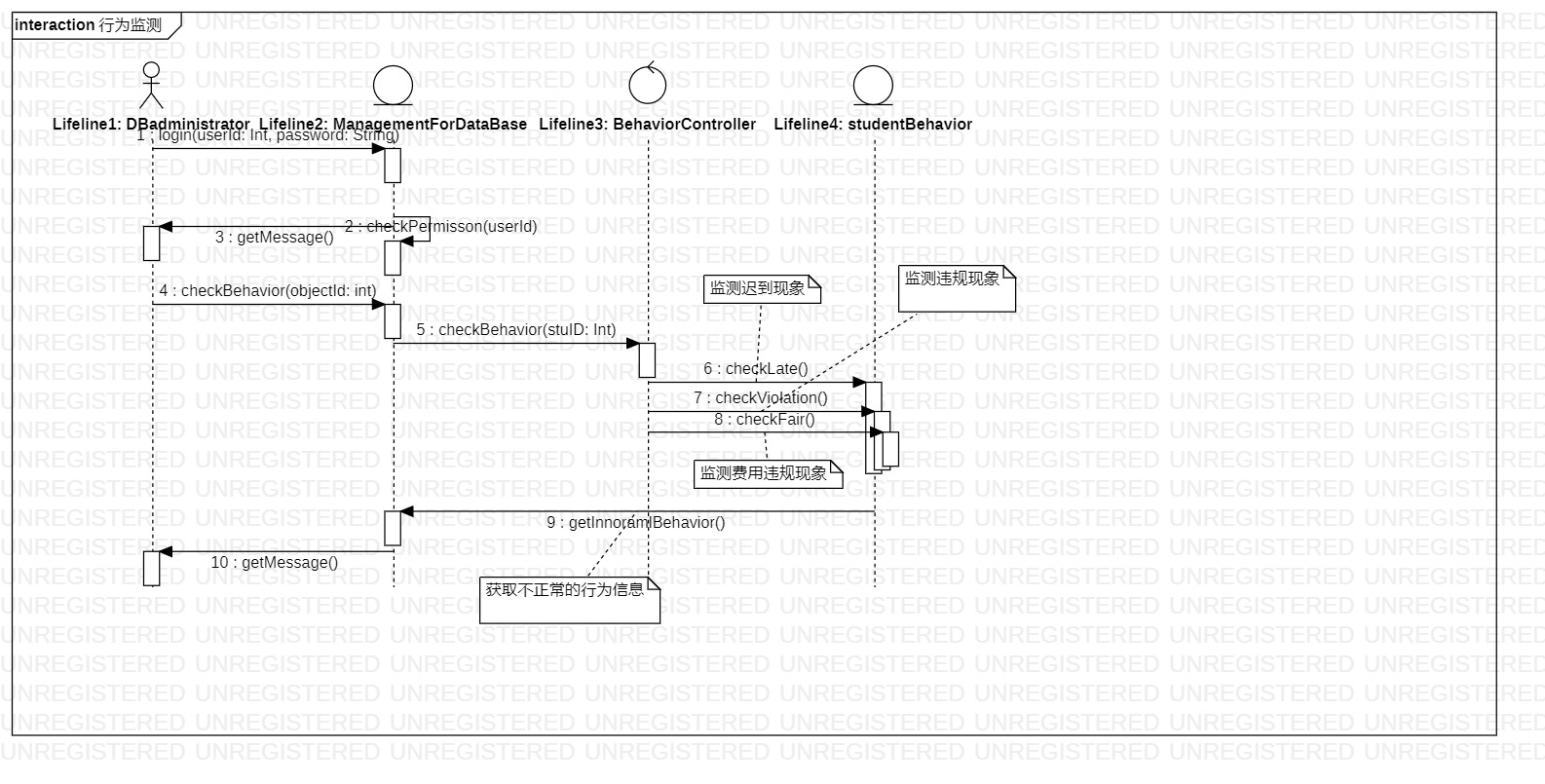
设计类图



实现关系



时序图



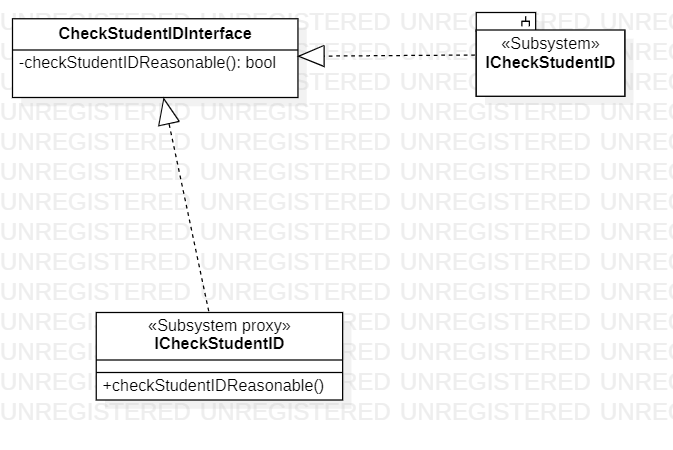
## 3.3子系统设计

### 3.3.1 数据库系统

### 3.3.2 学生信息系统交互

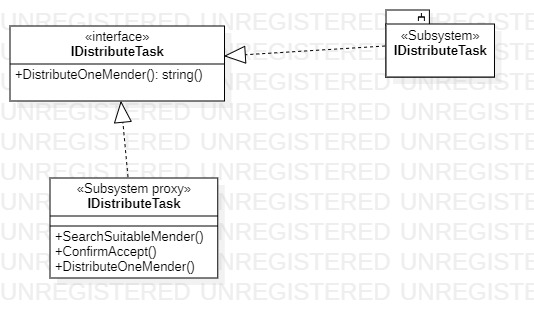


### 3.3.3 检查学生学号



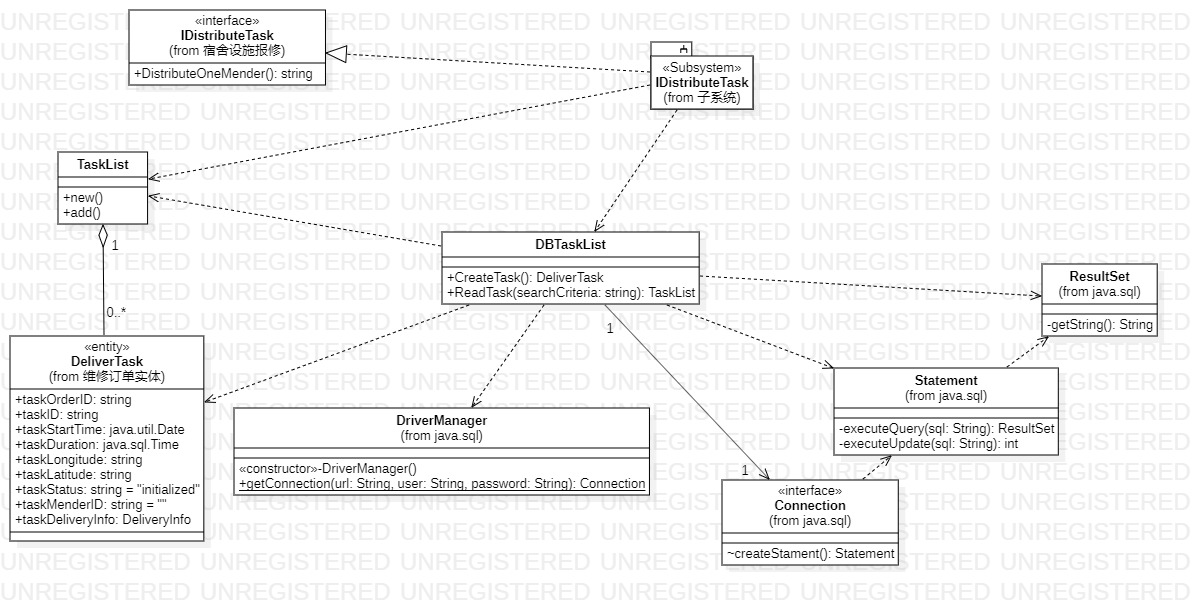
### 3.3.4 分派维修任务系统

子系统关系图：

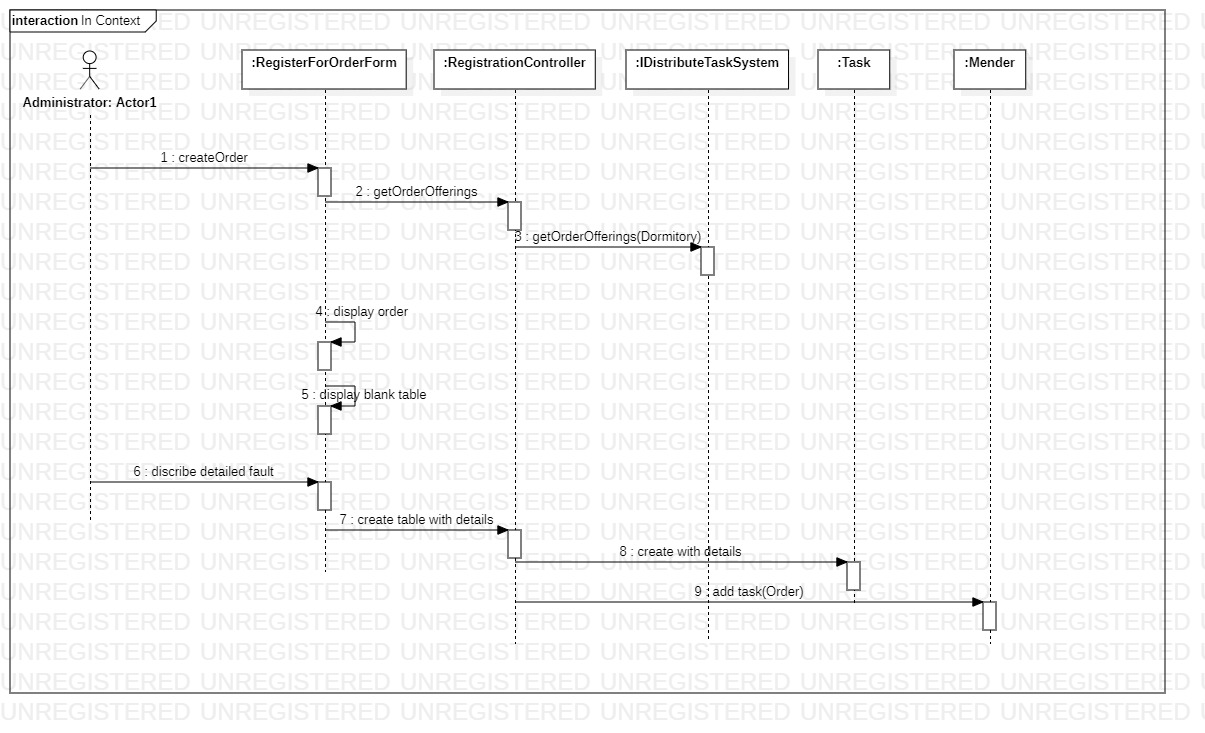


说明：该系统提供一个子系统，用于在管理员发出报修请求后对此次报修订单进行分配，分配到合适的维修人员手上。同时，该系统提供一个接口IDistributeTask供子系统实现，接口实现函数DistributeOneMender()向系统返回一个维修人员的ID信息，表示由该维修人员负责完成修理任务。最后系统提供一个代理服务器供相关维修人员使用，包括根据算法查询合适的维修员、维修员确认执行修理任务以及返回一个可执行任务的修理员。

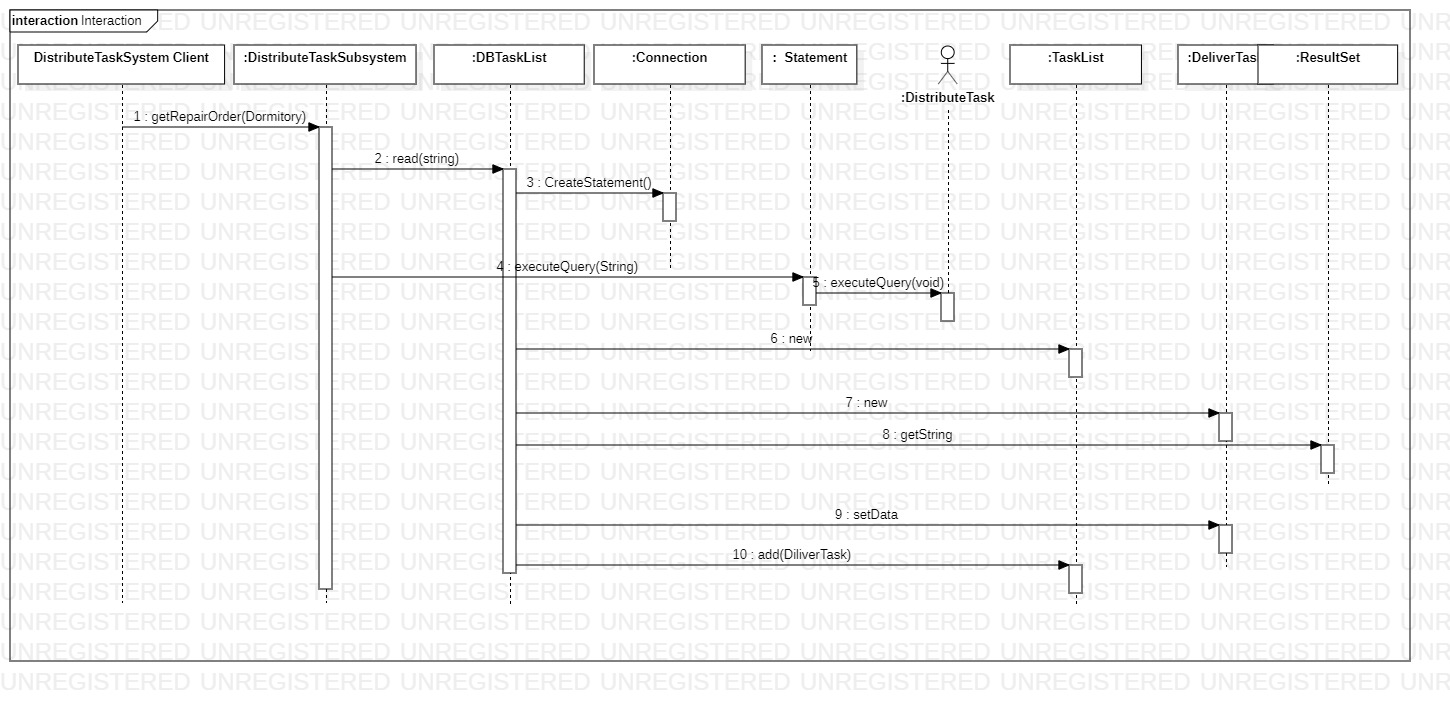
子系统元素实现机制：



上下文：

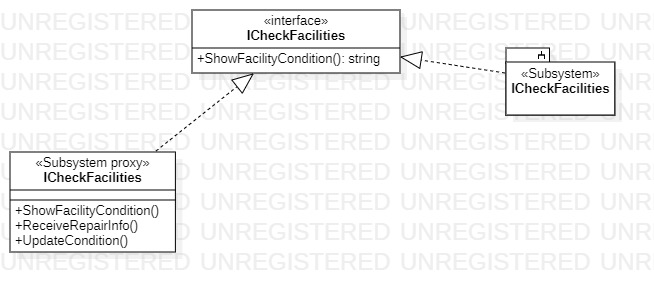


交互图：



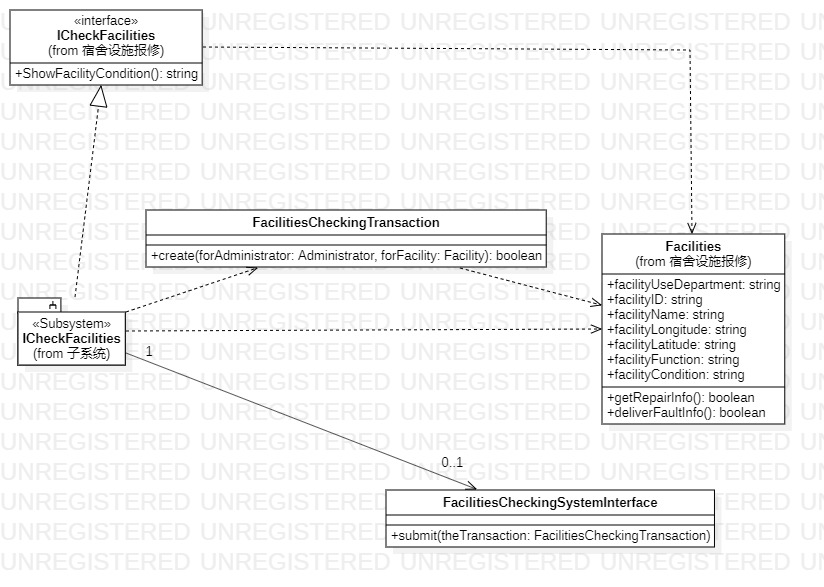
### 3.3.5 设施检查子系统

子系统关系图：

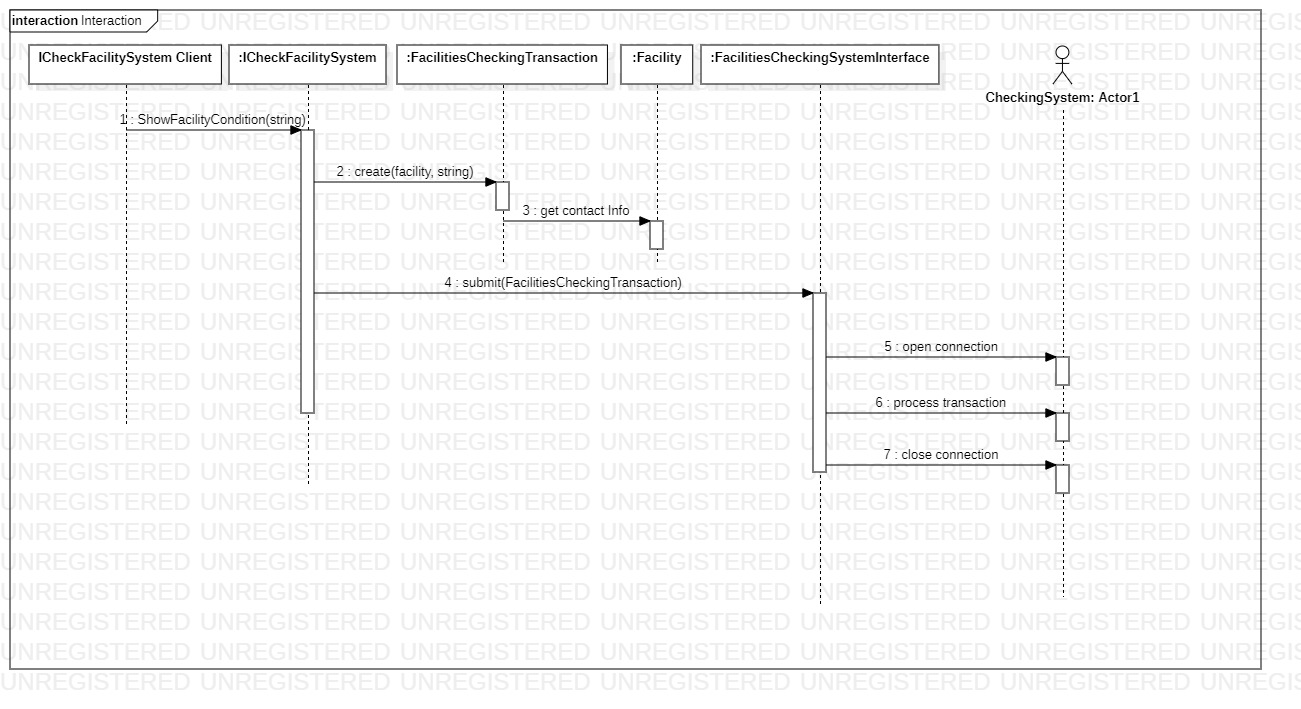


说明：该系统提供一个子系统，用于帮助管理员检查设备故障情况，并在设备出现故障维修完成后将故障信息同步到该设备。同时，该系统提供一个接口ICheckFacilities供子系统实现，接口实现函数ShowFacilityCondition()向系统返回一台设备的故障情况。最后系统提供一个代理服务器供相关人员使用，包括查看设备状况、返回维修信息以及在维修完成后更新同步故障信息到该设备。

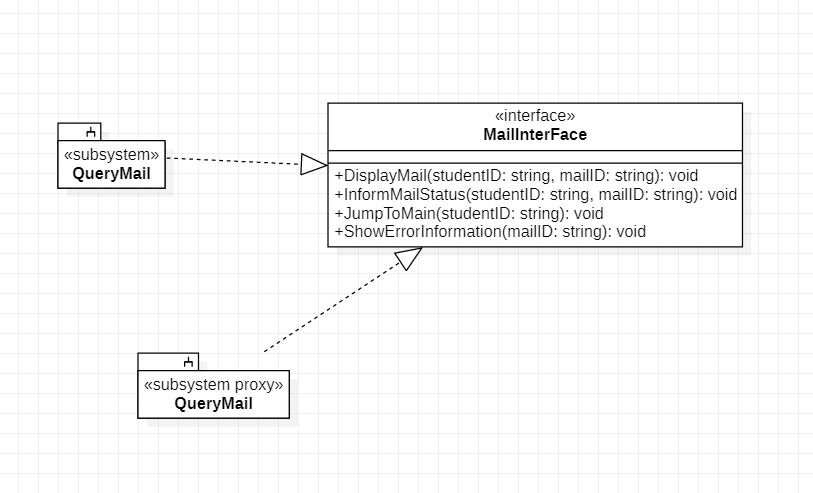
子系统元素实现机制：



交互图：



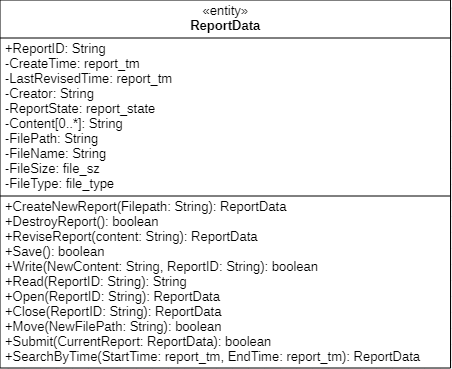
### 3.3.6 快递快件查询子系统



## 3.4 类设计

### 3.4.1 ReportData（报表信息）类from财务统计系统

类总览：



**Class stereotype:** Entity

**Architectural mechanisms:** Persistency, Security

**Attributes Table:**

|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| ReportID | String | 报表ID |
| CreateTime | report\_tm | 报表创建时间 |
| LastRevisedTime | report\_tm | 最后更改时间 |
| Creator | String | 报表创建者 |
| ReportState | report\_state | 报表状态（未修改、已修改、已提交） |
| Content[0..\*] | String | 报表内容 |
| FilePath | String | 报表文件路径 |
| FileName | String | 报表文件名 |
| FileSize | file\_sz | 报表文件大小 |
| FileType | file\_type | 报表文件类型（表格、文档） |

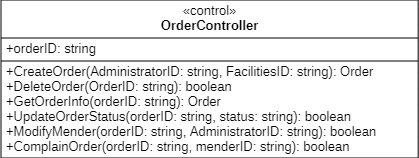
**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| CreateNewReport | 创建一个新的报表，输入文件路径开辟一个报表数据的存储空间 |
| DestroyReport | 删除一个报表，返回boolean类型值表示删除成功与否 |
| ReviseReport | 修改报表信息 |
| Save | 文件操作，保存报表文件 |
| Write | 文件操作，将报表文件写入存储器 |
| Read | 文件操作，从存储器中读取报表文件信息 |
| Open | 文件操作，从存储器打开一个报表文件进行操作 |
| Close | 文件操作，关闭已打开的报表文件 |
| Move | 文件操作，移动报表文件存储位置 |
| Submit | 提交报表 |
| SearchByTime | 根据创建时间查询对应报表信息 |

**Association:** Associate with <<control>>StatisticsControler

### 3.4.2 OrderContorller（订单控制）类from 宿舍设施报修

类总览：



**Class stereotype:** Control

**Architectural mechanisms:** Legacy Interface, Inter-Process Communication **Attributes Table:**

|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| orderID | String | 订单ID |

**Operations Table:**

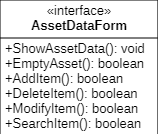
|  |  |
| --- | --- |
| **OpName** | **Description** |
| CreateOrder | 创建一个新的订单，传入参数管理员ID及要维修设备的ID，读取设备信息输出对应的报修订单 |
| DeleteOrder | 删除一个订单，返回boolean类型值表示删除成功与否 |
| GetOrderInfo | 获取一段时间内的订单信息 |
| UpdateOrderStatus | 更新订单状态（已接受、已发出、正在进行、已完成） |
| ModifyMender | 用户可自主更改派送的维修人员，选择合适的维修人员 |
| ComplainOrder | 提供通道给用户对维修人员进行投诉 |

**Association:** Associate with <<interface>>OrderForm

Aggregation from <<entity>>Order to OrderController

### 3.4.3 AssetDataForm（资产数据）类from 宿舍资产管理

类总览：



**Class stereotype:** Interface

**Architectural mechanisms:** Legacy Interface

**Operations Table:**

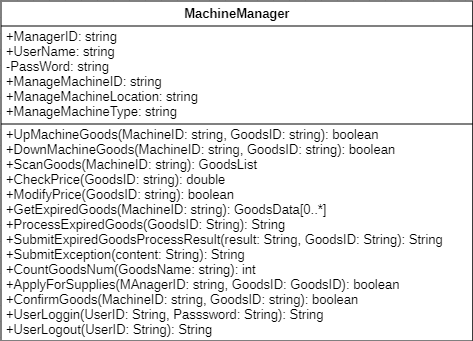
|  |  |
| --- | --- |
| **OpName** | **Description** |
| ShowAssetData | 显示当前资产信息。 |
| EmptyAsset | 弹出清空学生资产界面，返回布尔值表示清空资产清空成功与否。 |
| AddItem | 弹出添加资产至总资产界面，返回布尔值表示用户成功操作或终止操作。 |
| DeleteItem | 弹出从总资产删除资产的界面，返回布尔值表示用户成功操作或终止操作。 |
| ModifyItem | 弹出修改总资产中资产数量界面，返回布尔值表示用户成功操作或终止操作。 |
| SearchItem | 弹出在总资产中查询资产界面，返回布尔值表示用户成功操作或终止操作。 |

**Association:** Associate with <<control>>AssetManagementController

Associate with Administrator

### 3.4.4 MachineManager（机器管理员）类from宿舍自动贩卖机货物整理

类总览：



**Class stereotype:** Role

**Architectural mechanisms:** Persistency, Security

**Attributes Table:**

|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| ManagerID | String | 管理员ID |
| UserName | String | 管理员用户名 |
| PassWord | String | 密码（private） |
| ManageMachineID | String | 管理的贩卖机ID |
| ManageMachineLocation | String | 管理的贩卖机地点 |
| ManageMachineType | String | 管理的贩卖机类型 |

**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| UpMachineGoods | 货物上架操作 |
| DownMachineGoods | 货物下架操作 |
| ScanGoods | 供应商品信息扫描 |
| CheckPrice | 检查商品货物价格 |
| ModifyPrice | 修改该商品货物价格 |
| GetExpiredGoods | 过期商品信息获取 |
| ProcessExpiredGoods | 过期商品处理 |
| SubmitExpiredGoodsProcessResult | 过期商品处理提交结果 |
| SubmitException | 提交异常报告 |
| CountGoodsNum | 统计贩卖机内商品货物总数 |
| ApplyForSupplies | 补货请求，连接货物供应商子系统 |
| ConfirmGoods | 管理员确认贩卖机中商品信息 |
| UserLoggin | 用户登录系统 |
| UserLogout | 用户登出系统 |

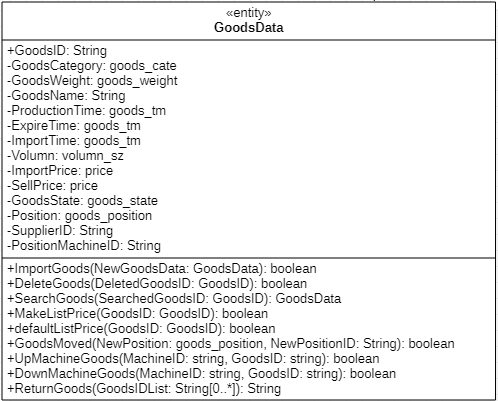
**Association:** Associate with <<interface>>GoodsDataManipulateInterface

Associate with <<control>>MachineManageController

Coposition by <<entity>>VendingMachineTotalGoods （multiplicity from 1…\* to 1）

### 3.4.5 GoodsData（货物信息）类from宿舍自动贩卖机货物整理

类总览：



**Class stereotype:** Entity

**Architectural mechanisms:** Persistency

**Attributes Table:**

|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| GoodsID | String | 货物ID |
| GoodsCategory | goods\_cate | 货物类型 |
| GoodsWeight | goods\_weight | 货物重量 |
| GoodsName | String | 货物名称 |
| ProductionTime | goods\_tm | 生产日期 |
| ExpireTime | goods\_tm | 保质期 |
| ImportTime | goods\_tm | 进货时间 |
| Volumn | volumn\_sz | 货物体积 |
| ImportPrice | price | 货物进价 |
| SellPrice | price | 当前出售价格 |
| GoodsState | goods\_state | 货物状态（在售、售罄、补货中、下架） |
| Position | goods\_position | 货物产地 |
| SupplierID | String | 供货商ID |
| PositionMachineID | String | 储存该货物的贩卖机ID |

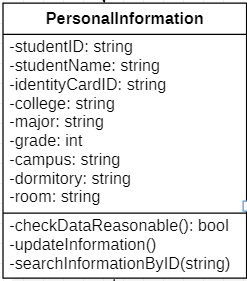
**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| ImportGoods | 货物入仓操作 |
| DeleteGoods | 货物出仓操作（删除该货物信息） |
| SearchGoods | 查询特定货物信息 |
| MakeListPrice | 货物确定标价 |
| defaultListPrice | 初始化货物标价 |
| GoodsMoved | 货物移仓/转移操作 |
| UpMachineGoods | 货物上架到自动贩卖机 |
| DownMachineGoods | 货物从自动贩卖机处下架 |
| ReturnGoods | 返回货物商品信息 |

**Association:** Associate with <<entity>>GoodsList（multiplicity from 1...\* to 1）

### 3.4.6 PersonalInformation（学生个人信息）类 from 学生信息管理

**类总览**



**Class stereotype:** Entity

**Architectural mechanisms:** Persistency

**Attributes Table:**

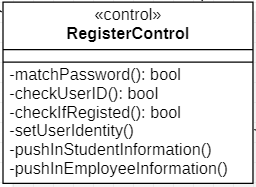
|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| studentID | string | 学生学号 |
| studentName | string | 学生姓名 |
| identityCardID | string | 学生身份证号 |
| college | string | 学院 |
| major | string | 专业 |
| grade | int | 年级 |
| campus | string | 校区 |
| dormitory | string | 宿舍楼 |
| room | string | 宿舍房间号 |

**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| CheckDataReasonable | 检查输入数据是否合理且合规 |
| updateInformation | 将原信息更新为新输入的信息 |
| searchInformationByID | 根据学号查询学生信息 |

### 3.4.7 RegisterControl（注册控制）类 from 注册

**类总览**



**Class stereotype:** Control

**Architectural mechanisms:** Security 、Inter-Process Communication

**Attributes Table:**

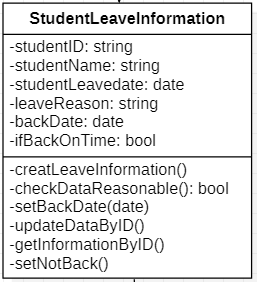
|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
|  |  |  |

**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| matchPassword | 检查输入数据是否合理且合规 |
| checkUserID | 检查用户账号是否在学生及教工中存在 |
| checkIfRegistered | 检查该账号是否已经注册过 |
| setUserIdentity | 设置账号的身份 |
| pushInStudentInformation | 将学生信息传至数据库 |
| pushInEmployInformation | 将职工信息传至数据库 |

### 3.4.8 StudentLeaveInformation(学生离校信息)类 from 学生长时间离宿

**类总览**



**Class stereotype:** Entity

**Architectural mechanisms:** Persistency

**Attributes Table:**

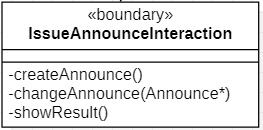
|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| studentID | string | 学生学号 |
| studentName | string | 学生姓名 |
| studentLeaveDate | date | 学生离开时间 |
| leaveReason | string | 学生离校理由 |
| backdate | date | 预计返回时间 |
| ifBackOnTime | bool | 标记学生是否按时返回 |

**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| createLeaveInformation | 创建新的学生离校登记信息 |
| checkDataReasonable | 检查输入数据是否合理且合规 |
| setBackDate | 更新返回时间 |
| updateDataByID | 根据输入的学号查找信息并更新数据 |
| getInformationByID | 根据学号查询该学生的离校信息 |
| setNotBack | 若学生未按时返回则将学生标记为未按时返回 |

### 3.4.9 IssueAnnounceInteraction（发布通知交互）from发布公告通知

**类总览**



**Class stereotype:** boundary

**Architectural mechanisms:** Inter-Process Communication

**Attributes Table:**

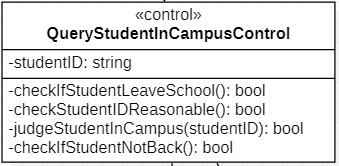
|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
|  |  |  |

**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| createAnnounce | 创建新的通知 |
| changeAnnounce | 修改已发布的通知 |
| showResult | 展示结果（通知发布成功或失败） |

### 3.4.10 QueryStudentInCampusControl(查询学生在校控制)from学生在校查询

**类总览**



**Class stereotype:** control

**Architectural mechanisms:** Legacy Interface、 Inter-Process Communication

**Attributes Table:**

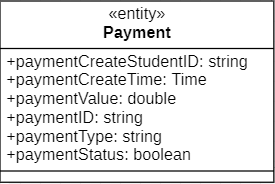
|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| studentID | string | 学生学号 |

**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| checkIfStudentLeaveSchool | 检查学生是否在长时间离宿状态 |
| checkStudentIDReasonable | 检查学生学号是否在系统中注册 |
| judgeStudentInCampus | 判断学生是否在校内 |
| checkIfStudentNotBack | 检查学生是否在晚归不归状态 |

### 3.4.11 Payment(寝室费用账单)from缴纳寝室费用

**类总览**



**Class stereotype:** entity

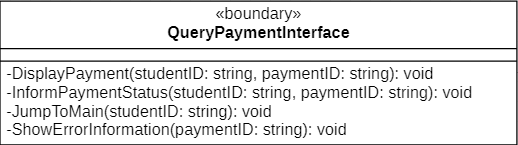
**Architectural mechanisms:** Persistency

**Attributes Table:**

|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| paymentCreateStudentID | string | 创建账单学生学号 |
| paymentCreateTime | Time | 创建账单时间 |
| paymentValue | double | 账单数额 |
| paymentID | string | 账单流水号 |
| paymentType | string | 账单类型 |
| paymentStatus | boolean | 标记账单是否已支付 |

### 3.4.12 QueryPaymentInterFace (查询费用交互)from缴纳寝室费用

**类总览**



**Class stereotype:** boundary

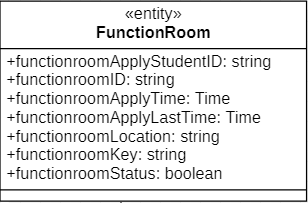
**Architectural mechanisms:** Inter-Process Communication

**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| DisplayPayment | 展示费用账单 |
| InformPaymentStatus( | 通知账单状态信息 |
| JumpToMain | 返回主界面 |
| ShowErrorInformation | 显示错误信息 |

### 3.4.13 FunctionRoom(功能室)from功能室申请

**类总览**



**Class stereotype:** Entity

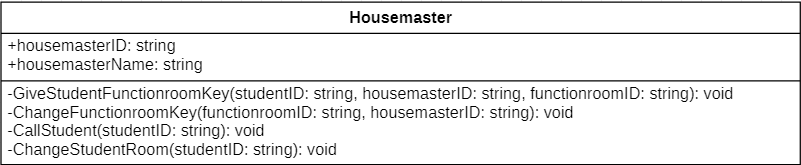
**Architectural mechanisms:** Persistency

**Attributes Table:**

|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| functionroomApplyStudentID | string | 申请功能室学生学号 |
| functionroomID | string | 功能室房间号 |
| functionroomApplyTime | time | 功能室申请时间 |
| functionroomApplyLastTime | time | 功能室使用时长nag |
| functionroomLocation | string | 功能室地点 |
| functionroomKey | string | 功能室密码 |
| functionroomStatus | Boolean | 标记功能室是否可申请 |

### 3.4.14 Housemaster(宿管)from申请更换宿舍&功能室申请

**类总览**



**Class stereotype:** role

**Architectural mechanisms:** Persistency

**Attributes Table:**

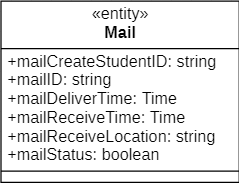
|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| housemasterID | string | 宿管工号 |
| housemasterName | string | 宿管姓名 |

**Operations Table:**

|  |  |
| --- | --- |
| **OpName** | **Description** |
| GiveStudentFunctionroomKey | 给予学生功能室密码 |
| ChangeFunctionroomKey | 更改功能室密码 |
| CallStudent | 呼叫学生 |
| ChangeStudentRoom | 更改学生宿舍 |

### 3.4.15 Mail(快递、快件)from快递快件查询

**类总览**



**Class stereotype:** entity

**Architectural mechanisms:** Persistency

**Attributes Table:**

|  |  |  |
| --- | --- | --- |
| **Name** | **DataType** | **Description** |
| mailCreateStudentID | string | 快递接收学生学号 |
| mailID | string | 快递单号 |
| mailDeliverTime | time | 快递分发时间 |
| mailReceiveTime | time | 快递接收时间 |
| mailReceiveLocation | time | 快递接收地点 |
| mailStatus | boolean | 标记快递是否已接收 |

# 四、组员分工

|  |  |
| --- | --- |
| 雷泓  1851202 | 项目工具使用介绍、用例实现（宿舍设施报修）、子系统设计（分派维修任务系统、设施检查子系统）、类设计3.4.1-3.4.5 |
| 王文政  1853928 | 用例实现（宿舍人员信息查询、注册）、子系统设计（学生信息系统交互、检查学生学号）、类设计3.4.6-3.4.10 |
| 肖博阳  1852672 | 用例实现（缴纳寝室费用）、子系统设计（快递快件查询子系统）、类设计3.4.11-3.4.15 |
| 张明哲  1852452 | 用例实现（信息更新与检索、行为检测）、子系统设计（数据库系统）、类设计3.4.16-3.4.20 |