

同济·宿



济宿团队

2252042 周政宇 2253722 张扬
2250695 奥泉瑞 2252144 张立伟

目录

1 项目介绍(INTRODUCTION)	4
1.1 项目目标	4
1.2 需求建模阶段的进展	4
1.3 功能优化与改进	4
1.4 当前状态	5
2 总体架构图	6
2.1 总体架构设计	6
2.2 架构图不同层次	7
2.3 逻辑架构	11
3 分析模型	13
3.1 用户管理子系统	13
3.2 宿舍信息子系统	15
3.3 设备和维修管理子系统	17
3.4 缴费管理子系统	18
3.5 公共资源管理子系统	19
3.6 公告与安全管理子系统	23
4 需求更新	25

5 快照更新	29
6 待解决的问题(OPEN ISSUES)	31
参考文献	33
团队成员贡献	35



1 项目介绍(INTRODUCTION)

1.1 项目目标

JISU 智能宿舍管理系统旨在解决当前高校宿舍管理系统功能分散、缺乏集成与智能化的问题。系统的核心目标包括实现智能化的宿舍分配、便捷的设施维护管理、自动化的费用管理、精确的门禁安全监控以及可视化的数据分析，通过整合这些功能，打造一个提升管理效率并优化学生居住体验的综合性平台。

1.2 需求建模阶段的进展

在需求分析阶段，我们完成了详细的系统建模工作：

- 采用敏捷开发方法，通过用户画像和用户故事深入理解用户需求
- 构建了六大核心子系统：用户管理、宿舍信息、设备和维修管理、缴费管理、公共资源管理以及公告与安全管理
- 完成了详细的用例模型，包括 13 个主要用例的规约和描述
- 绘制了 8 个关键业务流程的活动图，清晰展示了系统的运作机制
- 设计了 6 个核心功能模块的界面原型

1.3 功能优化与改进

通过需求分析，我们对系统功能进行了多项优化：

- 宿舍分配模块增加了基于生活习惯、学习时间和个人偏好的智能匹配算法
- 设施维护模块优化了进度追踪机制
- 费用管理模块增加了多渠道支付集成能力

- 🌀 公共资源管理模块新增了预约系统和使用状态实时展示
- 🌀 各模块增加了实时通知功能，提升用户交互体验

1.4 当前状态

项目目前已完成需求分析和初步设计阶段，正在进入分析模型设计阶段。

接下来将重点推进系统分层架构设计和基于类的子系统分析模型设计，确保系统的模块化架构具备良好的可扩展性和可维护性。



2 总体架构图

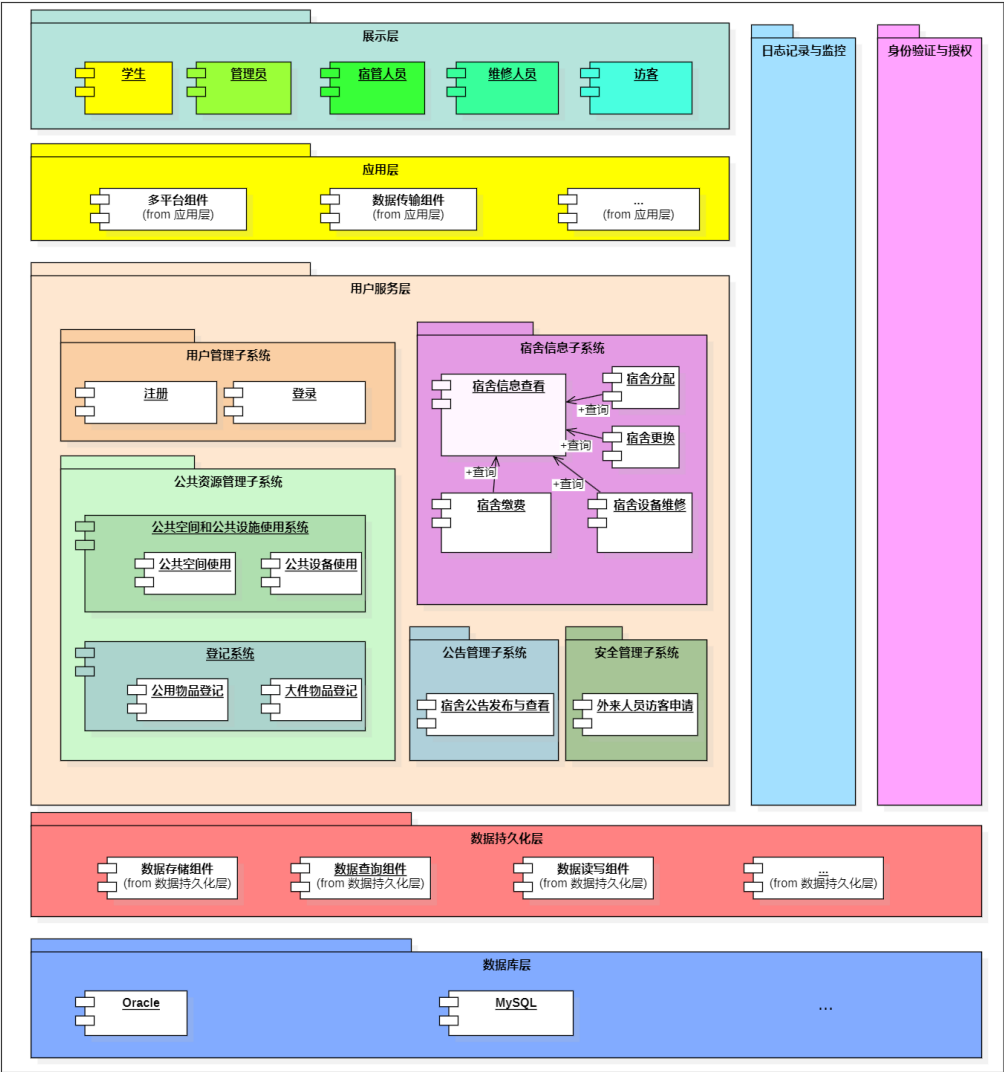
此架构方案将应用系统精确地划分为多个层次。每一层均聚焦于特定的问题域，通过协同合作，共同构建起全面的解决方案，既有利于团队成员进行高效的并行开发，又极大地方便了各层之间的独立测试。

2.1 总体架构设计

本项目对系统进行了细致的划分，具体包括展示层、应用层、用户服务层、数据持久化层以及数据库层。

- ✿ 展示层，作为用户交互的窗口，旨在为用户、管理员及运维人员提供直观、友好的图形用户界面（UI）。
- ✿ 应用层通过用户服务层所提供的接口能够向用户服务端发送请求，以获取所需数据。
- ✿ 用户服务层，作为系统的核心大脑，封装了 UI 适配模块及实现各类业务逻辑的子模块，旨在降低模块间的耦合度，提高代码的复用性。UI 适配模块负责与表展示层进行交互，而其他子模块的划分则依据项目的用例模型及子系统划分，这一层次的设计充分考虑了系统的可维护性及可扩展性极大地提升了系统的灵活性和响应速度。
- ✿ 数据持久化层，主要进行数据访问与读写等，这一层次的设计充分考虑了数据的可管理性、可扩展性及高性能需求。
- ✿ 数据库层，作为系统的数据存储中心，存储着系统的各类数据。

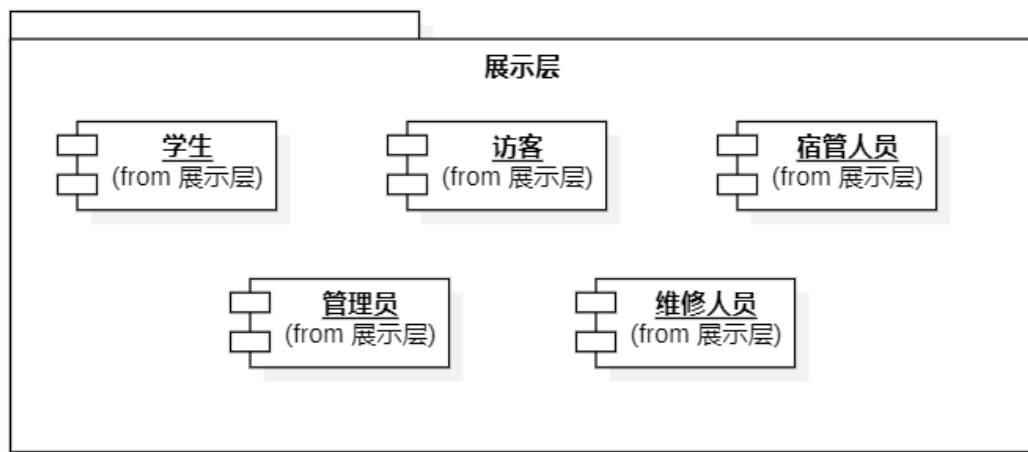
总的来说，这一架构方案为系统的稳定性、可扩展性及可维护性提供了坚实的基础。



2.2 架构图不同层次

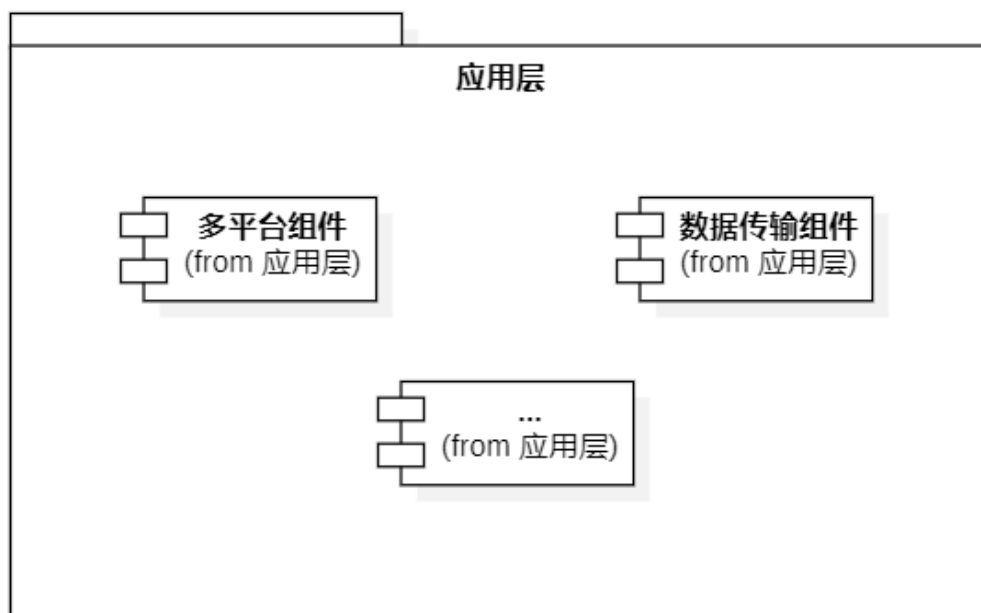
2.2.1 展示层

展示层是用户界面中的视觉呈现部分，负责将信息以直观、吸引人的方式展现给用户。它利用图形、文本、色彩等元素，确保内容清晰易懂，提升用户体验。优秀的展示层设计能引导用户操作，增强互动性，使信息传递更加高效。



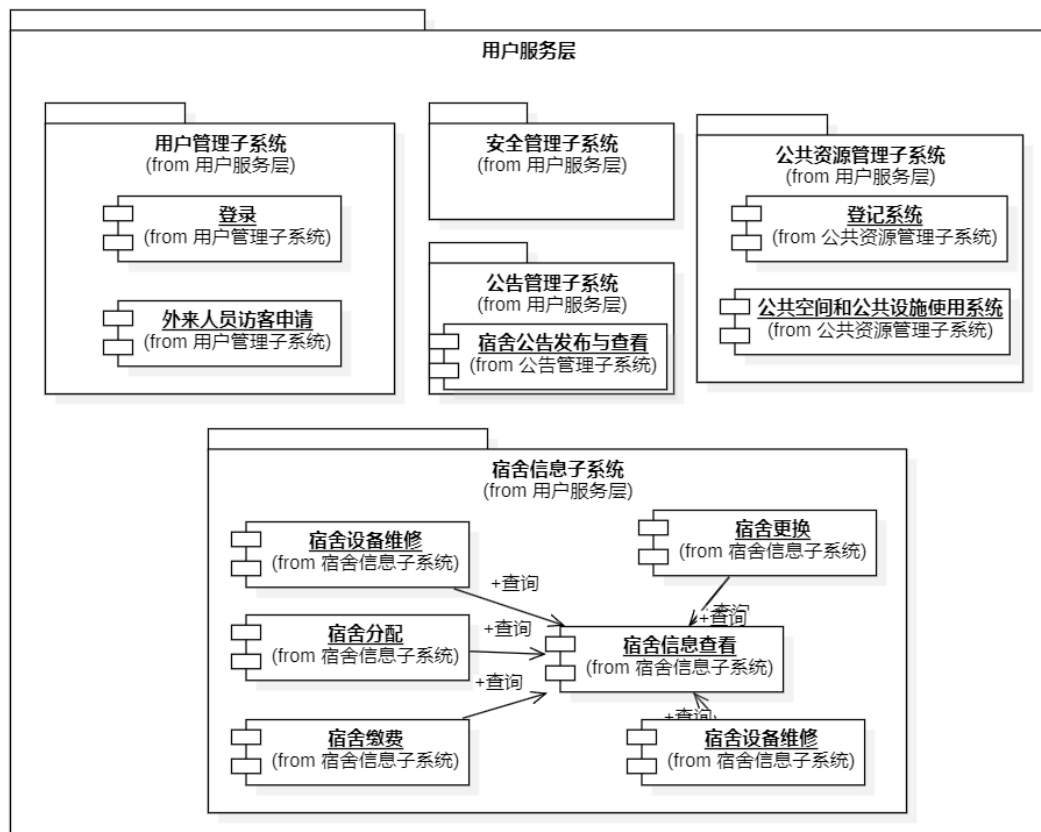
2.2.2 应用层

应用层是软件架构中直接与用户交互的部分，提供丰富的功能和服务。它接收用户输入，处理业务逻辑，并通过展示层呈现结果。应用层的设计需考虑用户需求、业务规则及交互体验，确保功能实用、操作便捷且用户友好。



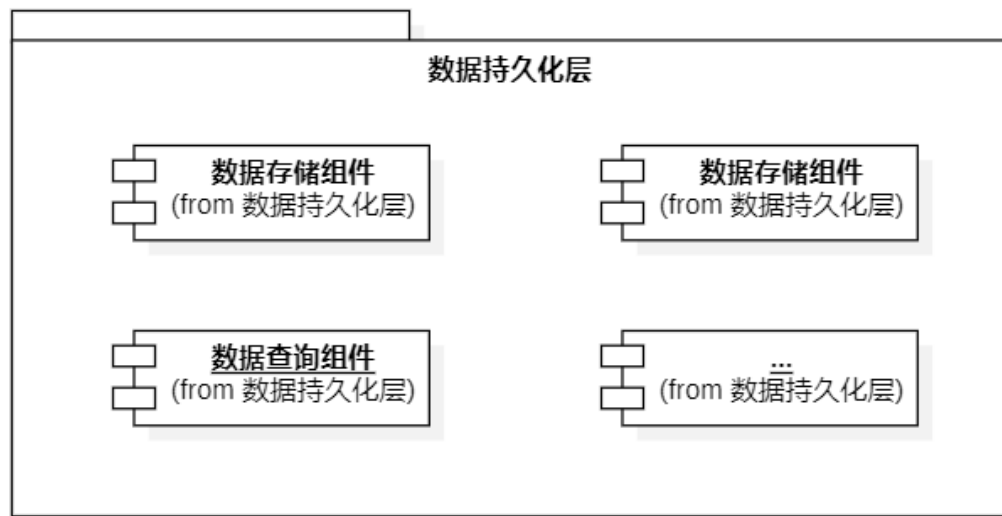
2.2.3 用户服务层

用户服务层是软件架构中专门负责处理用户请求和提供服务的层级。它接收来自应用层的用户指令，执行相应的业务逻辑，并与后台系统交互以获取所需数据，最终将处理结果反馈给用户，确保用户获得高效、流畅的服务体验。



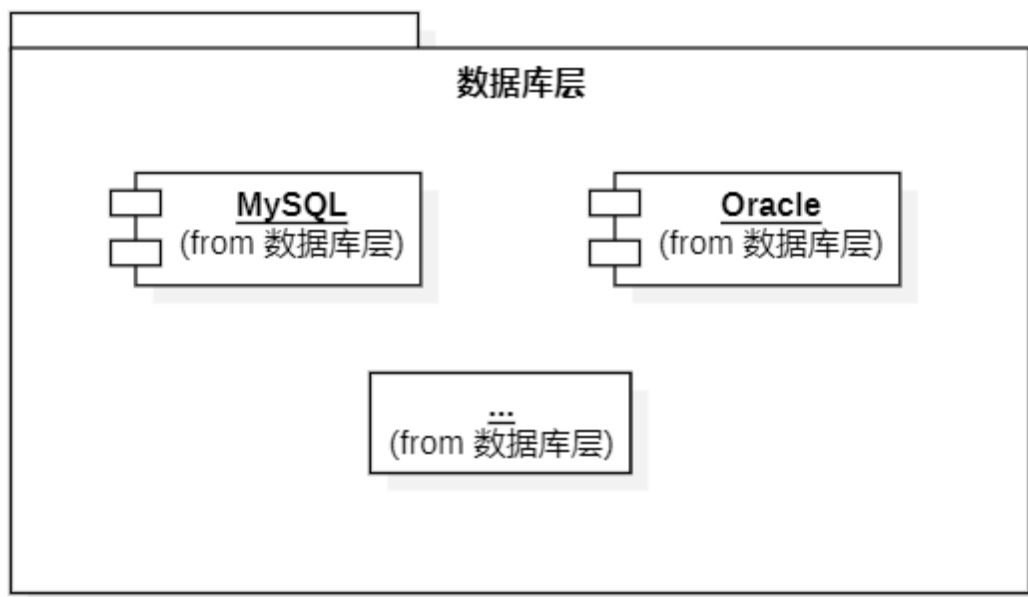
2.2.4 数据持久化层

数据持久化层是软件架构中专注于数据存储和持久化的层级。它负责将内存中的数据保存到数据库中，确保数据在系统重启或故障后仍能恢复。通过提供统一的数据访问接口，数据持久化层简化了数据管理工作，提高了系统的可靠性和数据安全性。



2.2.5 数据库层

数据库层是软件架构中的关键部分，负责处理所有与数据库相关的请求，提供数据的存储、检索、更新和删除功能，并保证数据的安全性、完整性和一致性。



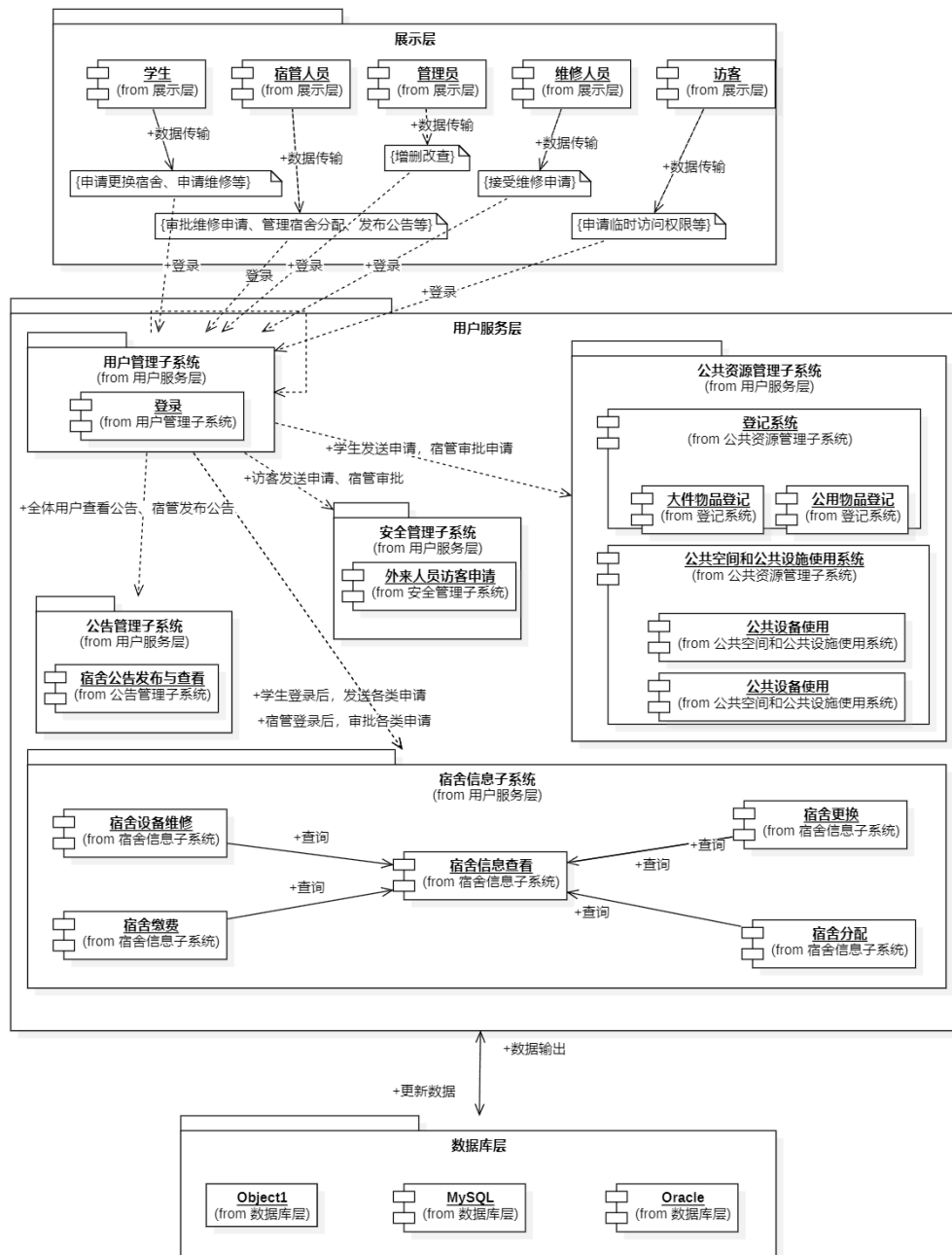
2.3 逻辑架构

在本宿舍管理系统中，采用微服务架构能显著提升系统的可扩展性、可维护性和灵活性。各个微服务之间通过定义清晰的接口进行通信，实现了松耦合和高内聚。

核心服务包括用户管理、宿舍信息、公告管理、安全管理和公共资源管理。

- ⊗ 用户微服务负责处理用户信息的增删改查，确保数据的一致性和安全性。
- ⊗ 宿舍信息微服务则综合管理宿舍信息，根据算法和规则，为用户分配宿舍，同时管理宿舍的更换、缴费和设备维修。
- ⊗ 公告管理微服务允许宿管发送宿舍公告，同时允许学生查看。
- ⊗ 安全管理微服务负责处理访客的申请，严格把控宿舍进出人员的身份，确保宿舍安全。
- ⊗ 公共资源管理微服务负责把控公共空间和公共功能设备的使用，同时包含登记系统，管理公共物品以及大件物品的登记。

这些微服务之间通过轻量级的通信协议（如 HTTP/HTTPS）和数据格式（如 JSON）进行交互，实现了高效的协作。同时，每个微服务都可以独立部署和扩展，提高了整个系统的灵活性和响应速度。

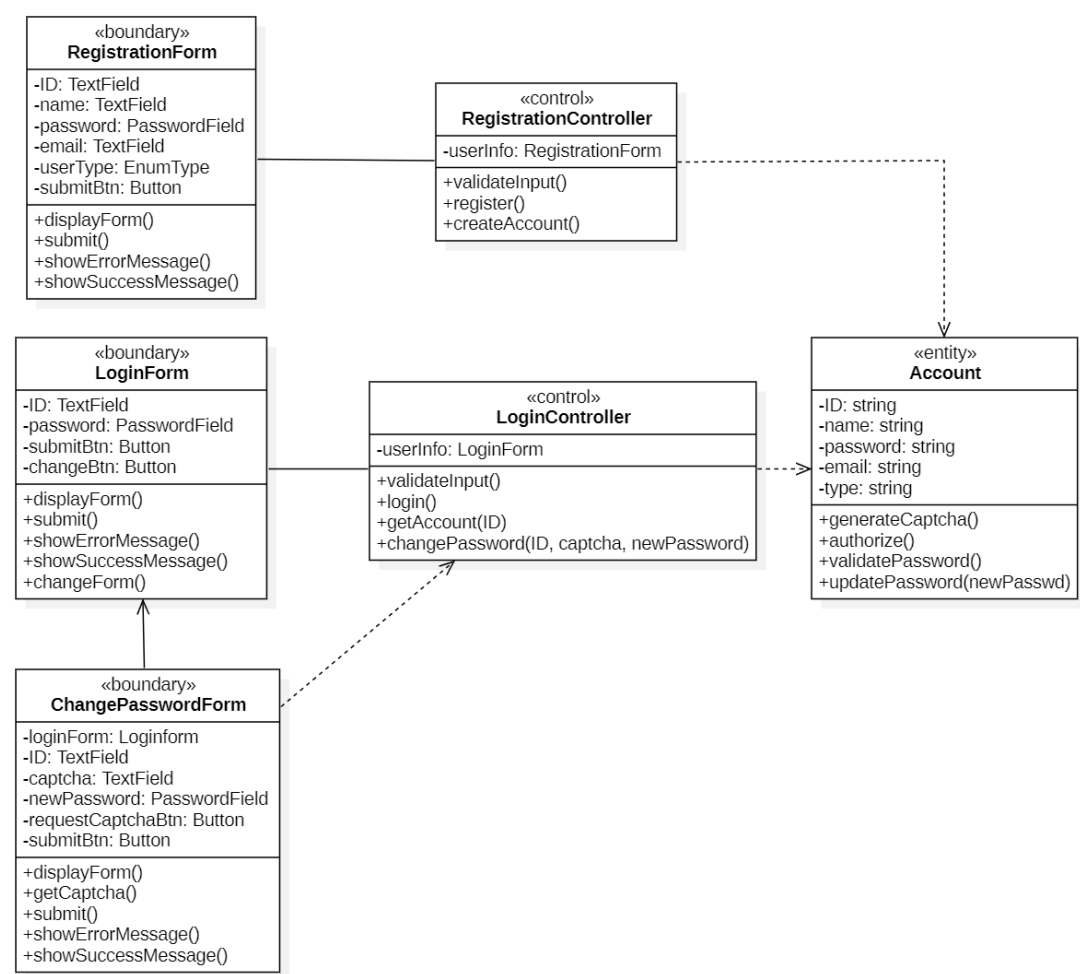


3 分析模型

3.1 用户管理子系统

3.1.1 类图

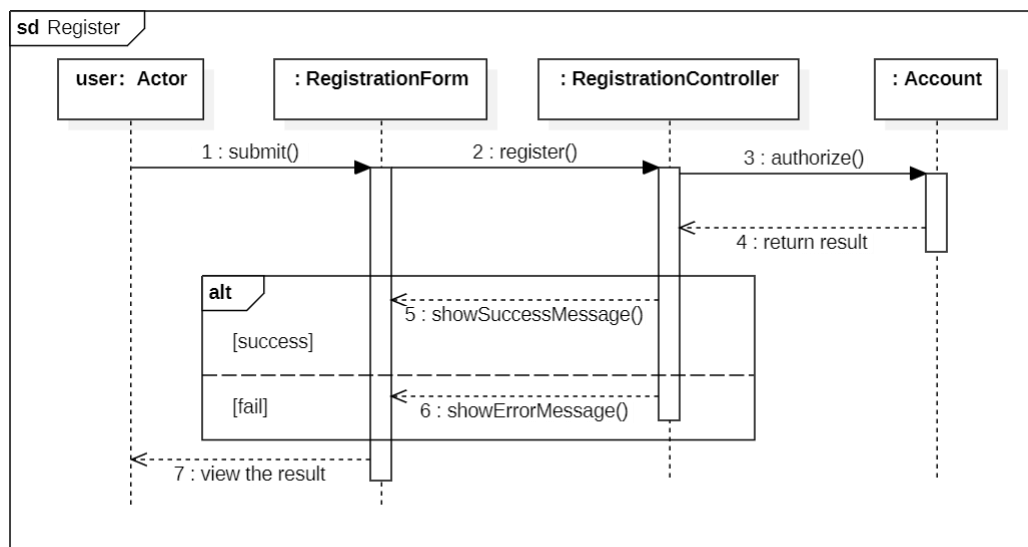
以下是用户管理子系统的类图，边界类为登录、注册、修改密码的表单；控制类负责管理登录和注册行为；实体类为 Account 用于存储账户信息。



3.1.2 交互图

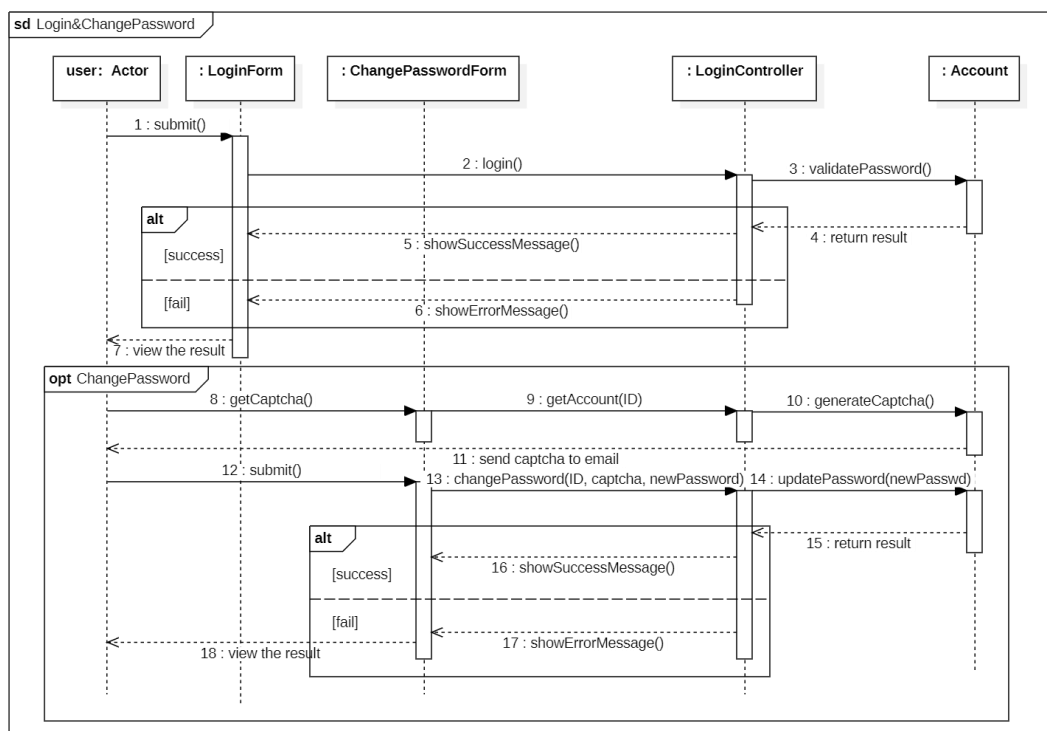
3.1.2.1 注册

这张时序图描述了内部用户进行注册的流程，用户信息将由 Account 类进行授权，并由 RegistrationForm 类负责显示注册结果。



3.1.2.2 登录和修改密码

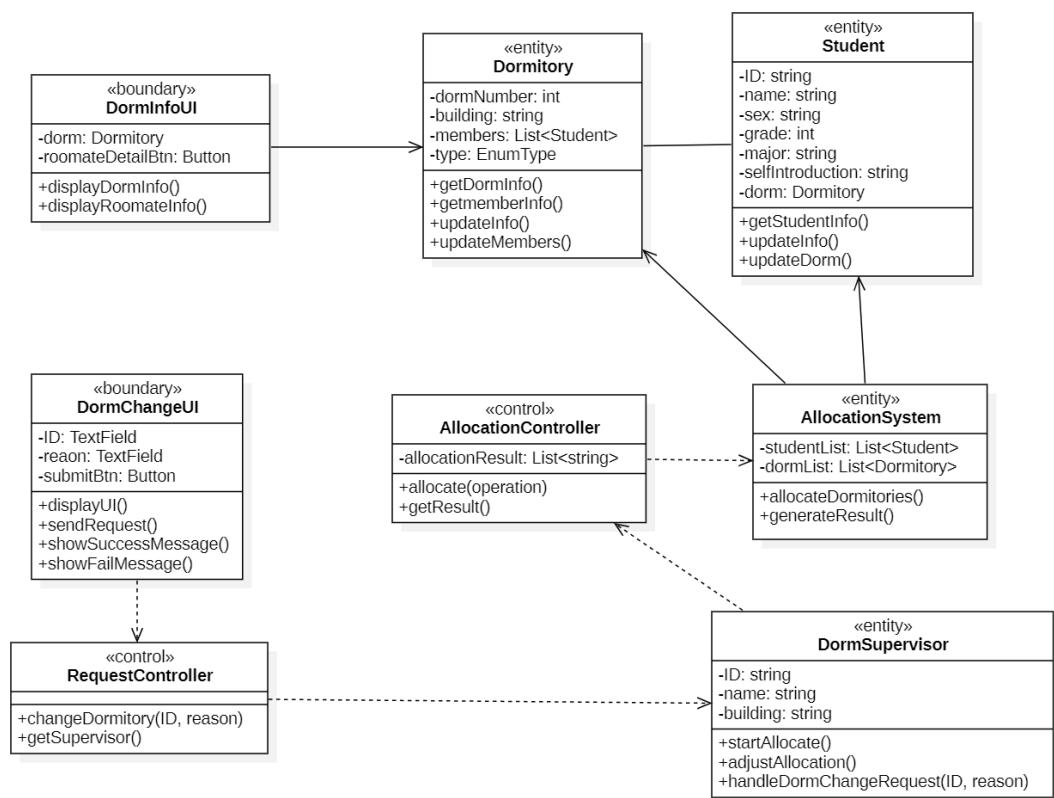
这张时序图描述了用户登录以及修改密码的流程。用户登录过程与注册类似，由 Account 类进行验证。修改密码作为可选项，需要用户填写验证码和新密码以进行修改。



3.2 宿舍信息子系统

3.2.1 类图

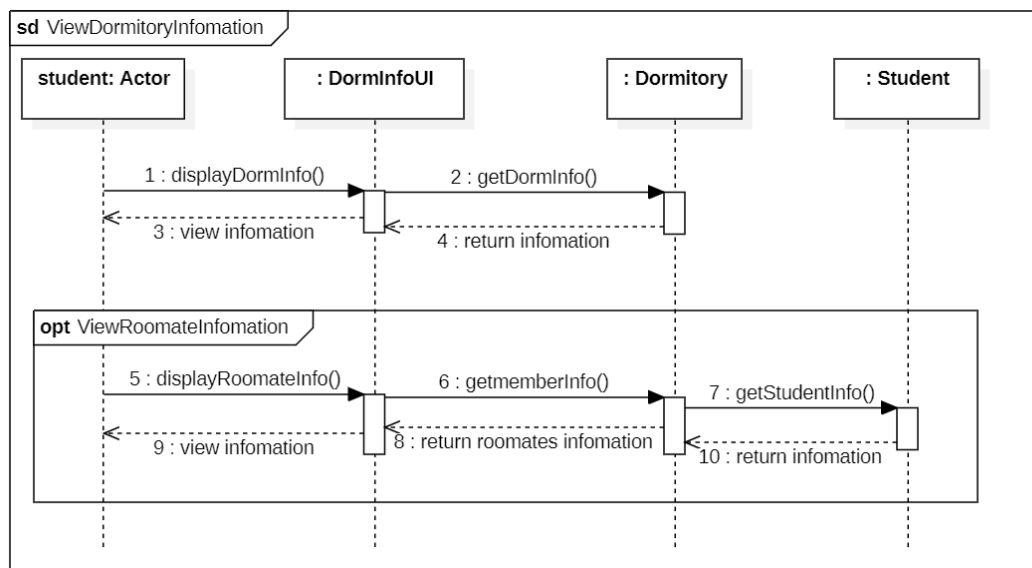
以下是宿舍信息子系统的类图，边界类负责显示宿舍信息界面和申请宿舍更换的表单；控制类负责管理换宿请求和宿舍分配；实体类有 Dormitory、Student、AllocationSystem 和 DormSupervisor，分别存储各自的信息和相应的处理操作。



3.2.2 交互图

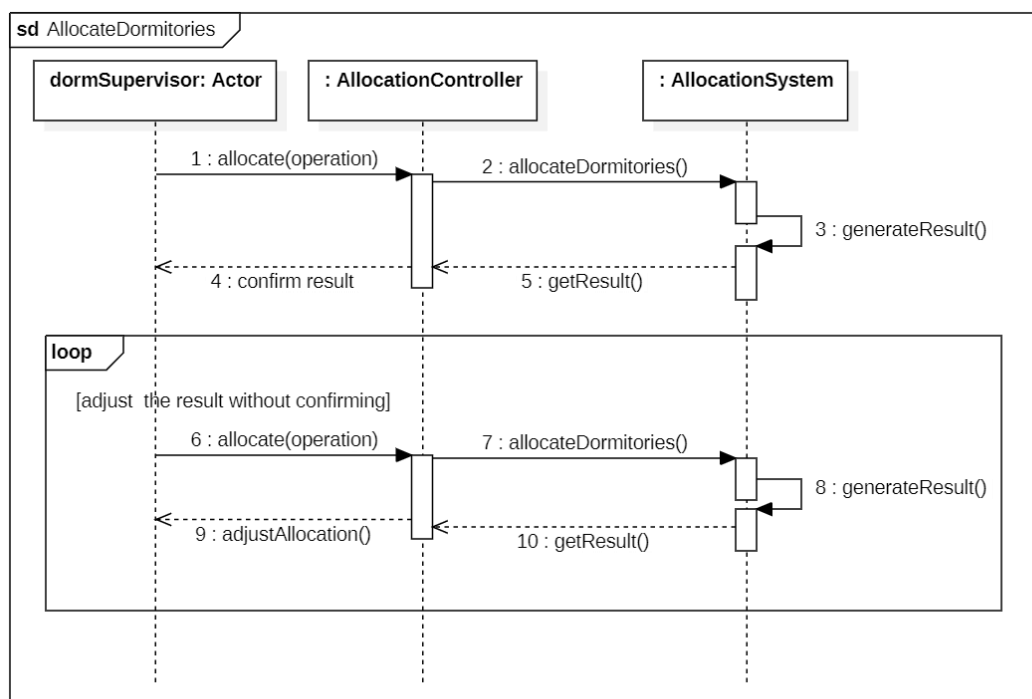
3.2.2.1 查看宿舍信息

这张时序图描述了学生如何查看宿舍信息。控制类收到请求后，将从 Dormitory 类获取与宿舍相关的信息，并且可以从 Student 类获取舍友信息。



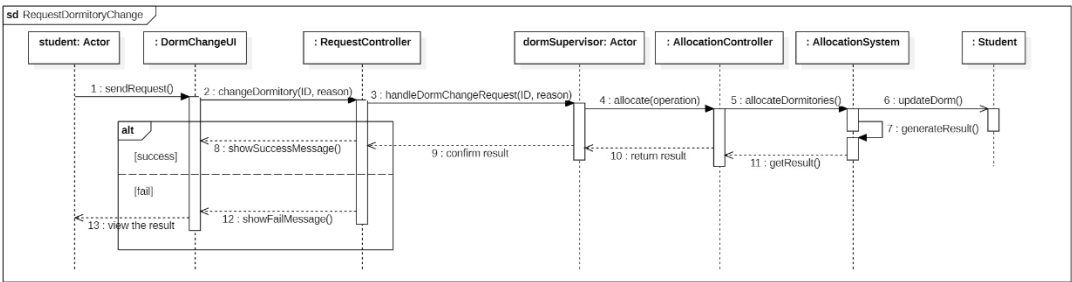
3.2.2.2 分配宿舍

这张时序图描述了系统分配宿舍的流程。有宿管启动分配，分配系统收到请求后进行宿舍分配，再由宿管调整分配结果，直至宿管确认结果。



3.2.2.3 申请更换宿舍

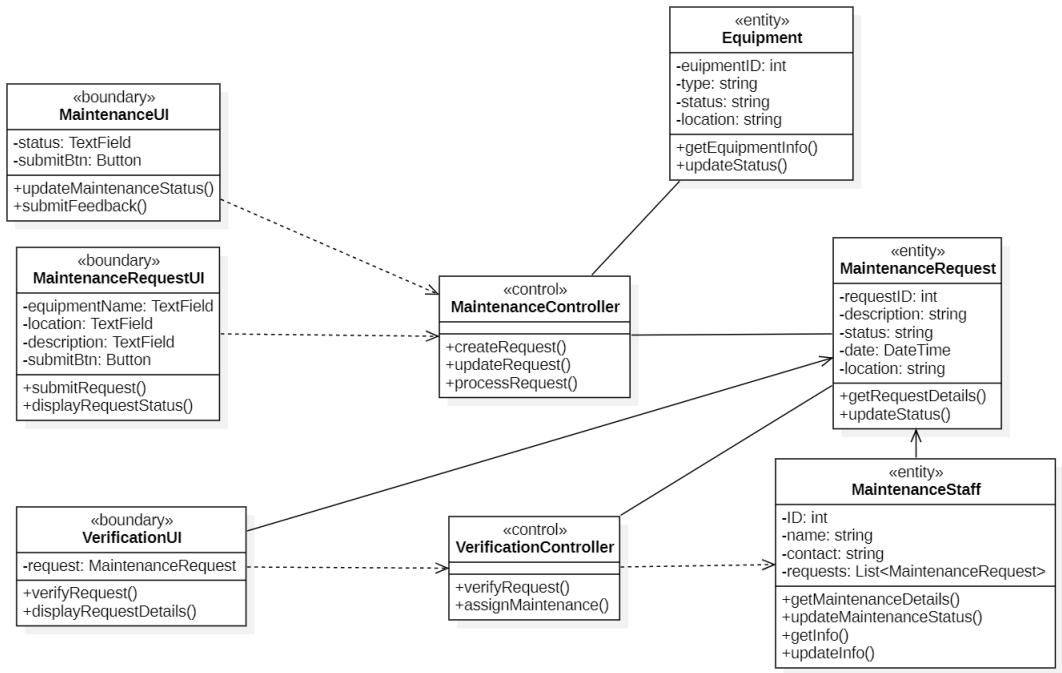
这张时序图描述了学生申请换宿的流程。换宿申请将由宿管进行审核，并由分配系统分配新的宿舍，同时 Student 类更新对应的宿舍信息。



3.3 设备和维修管理子系统

3.3.1 类图

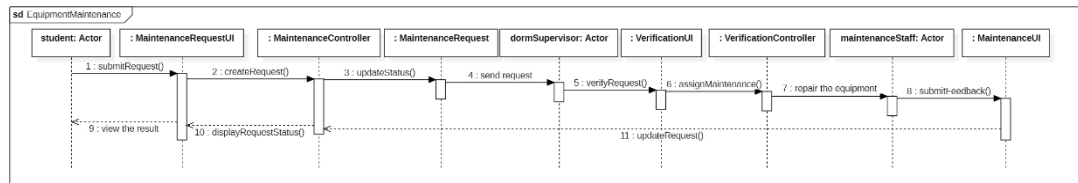
以下为设备和维修管理子系统的类图。边界类有学生查看维修结果、申请维修和 UI 界面和宿管审核维修申请的 UI 界面；控制类主要负责维修的申请和分配；实体类 `Equipment` 管理设备信息、`MaintenanceRequest` 管理维修信息、`MaintenanceStaff` 管理维修人员信息。



3.3.2 交互图

3.3.2.1 宿舍设备报修

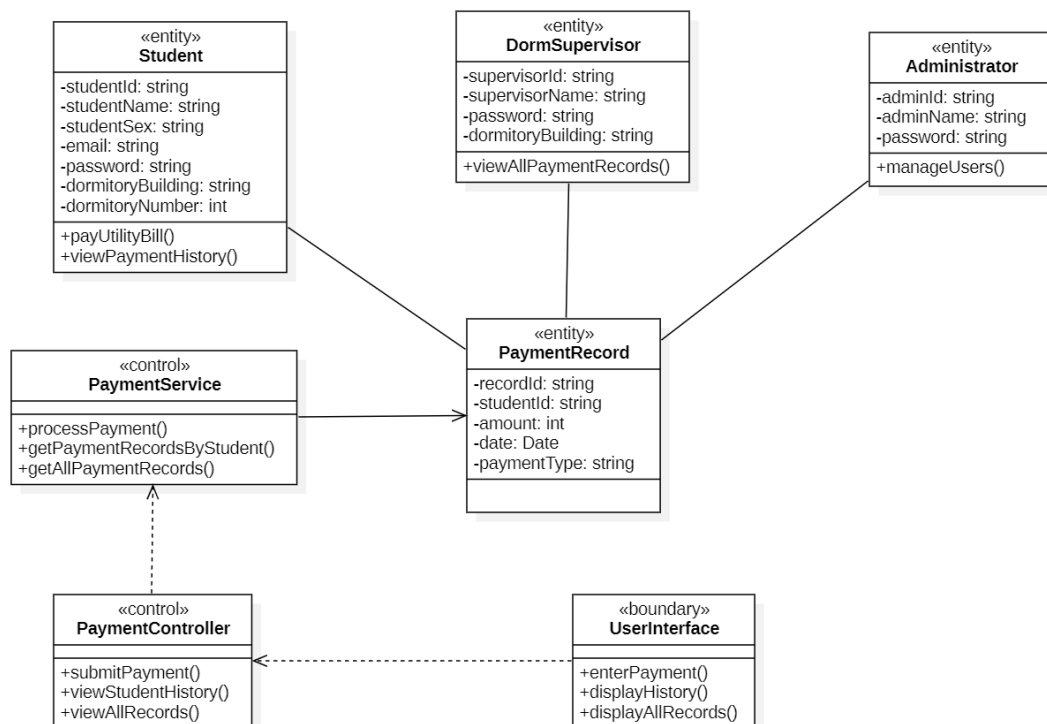
这张交互图描述了学生进行宿舍报修的流程。报修申请将交由宿管审核并分配对应的维修人员，维修人员维修结束后在平台进行反馈，反馈结果将显示在学生端的相应界面上。



3.4 缴费管理子系统

3.4.1 类图

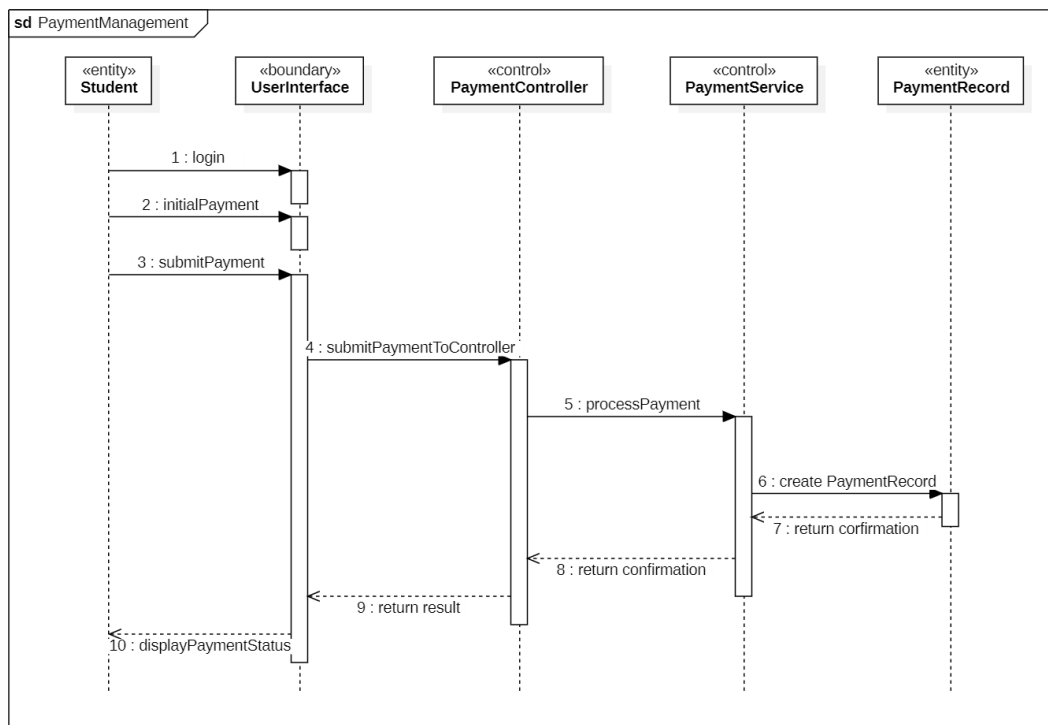
以下为缴费管理子系统的类图。边界类有用户接口的 UI 界面；控制类主要负责缴费的提交和查看；实体类 PaymentRecord 管理缴费信息。



3.4.2 交互图

3.4.2.1 缴费管理

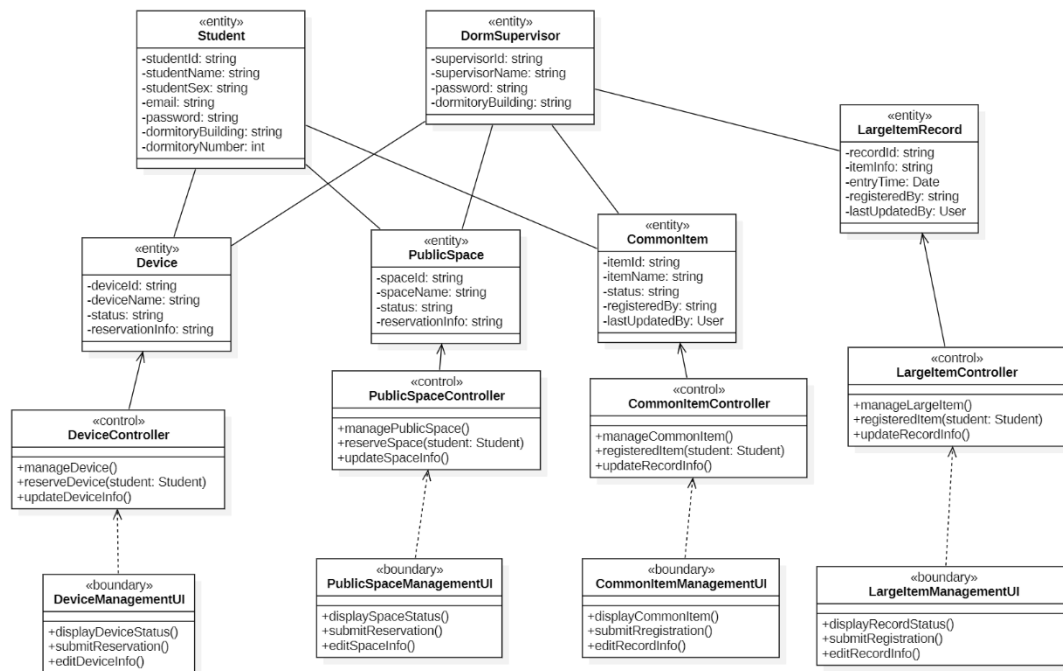
这张交互图描述了学生进行宿舍缴费的流程。学生申请缴费之后会调用支付相关的接口并创建缴费记录，缴费成功与否会在平台进行反馈，反馈结果将显示在学生端的相应界面上。



3.5 公共资源管理子系统

3.5.1 类图

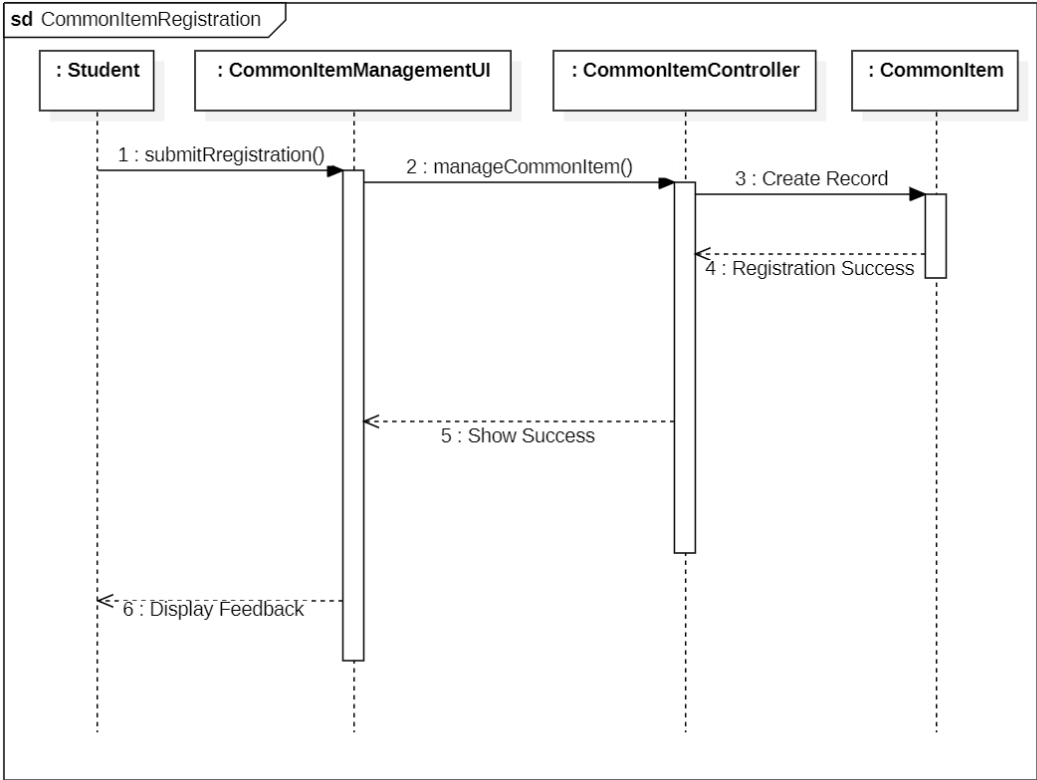
以下为公共资源管理子系统的类图。边界类有公用设备管理、公共空间管理、公用物品登记、大件物品登记的 UI 界面；控制类主要负责公用设备、公共空间的申请和分配以及公用物品和搭建物品的登记；实体类 Device、PublicSpace、CommonItem 和 LargeItemRecord 分别负责管理公用设备、公共空间、公用物品以及大件物品。



3.5.2 交互图

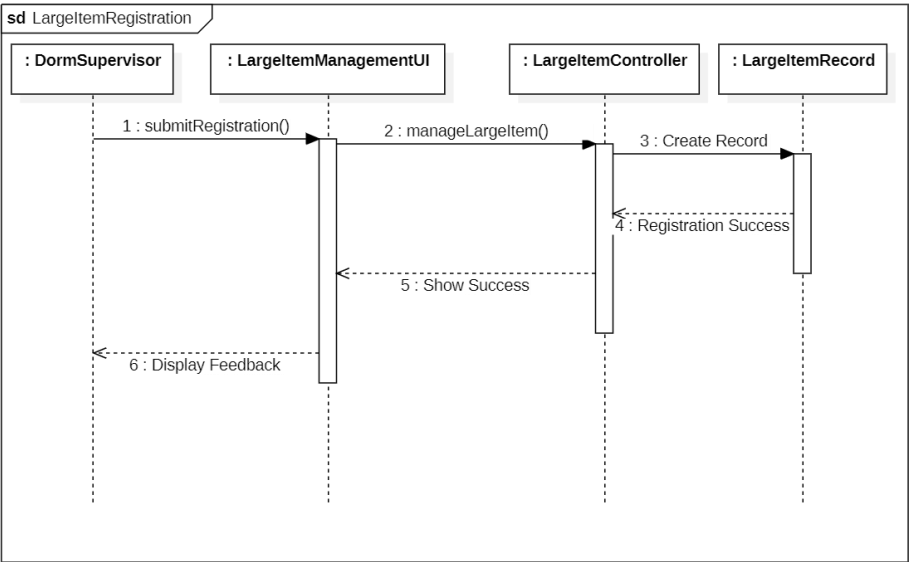
3.5.2.1 公共物品登记

这张交互图描述了学生进行公共物品登记的流程。公共物品的申请将交由宿管审核并登记，学生归还之后仍然需要宿管进行归还登记，反馈结果将显示在学生端的相应界面上。



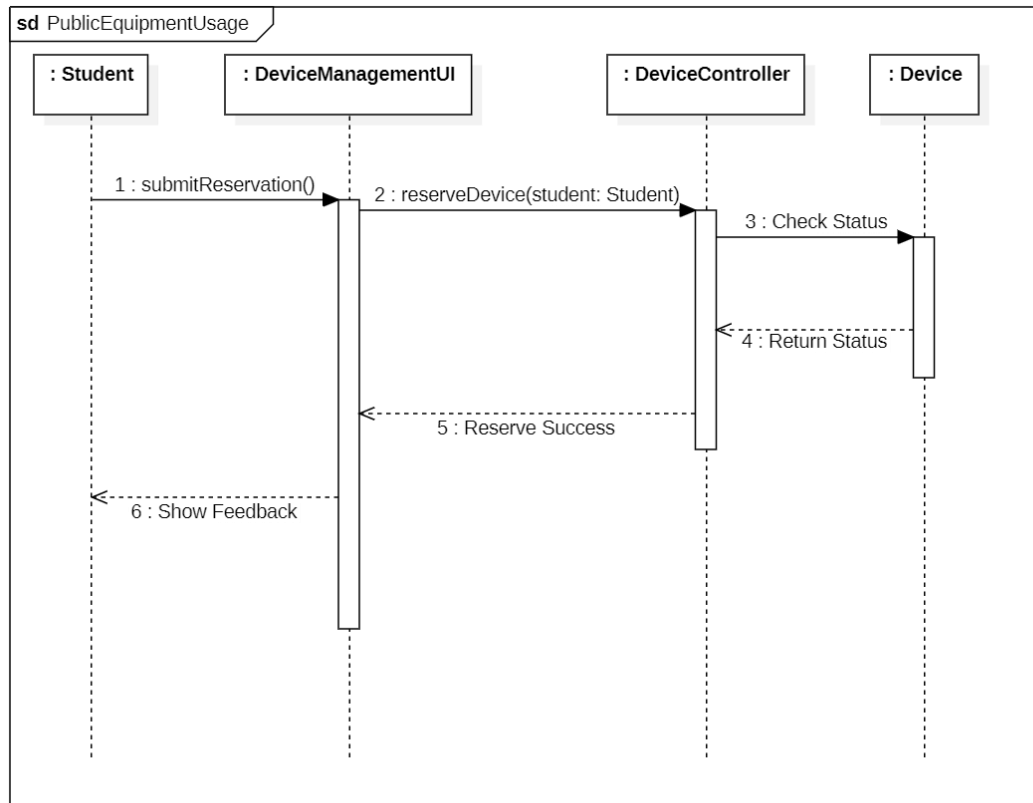
3.5.2.2 大件物品 登记

这张交互图描述了学生进行大件物品带入宿舍的登记流程。大件物品带入宿舍需要宿管登记，反馈结果将显示在学生端的相应界面上。



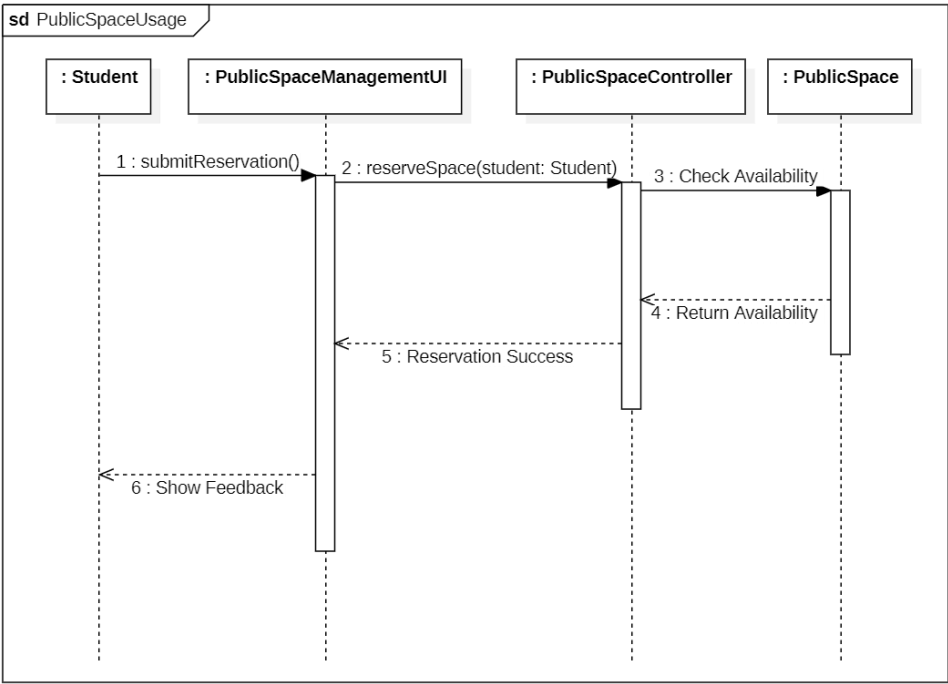
3.5.2.3 公用设备使用

这张交互图描述了学生进行公用设备预约使用的登记流程。公用设备的预约需要学生查看可用设备的状态以及确定使用时间，成功预约之后的反馈结果将显示在学生端的相应界面上。



3.5.2.4 公共空间使用

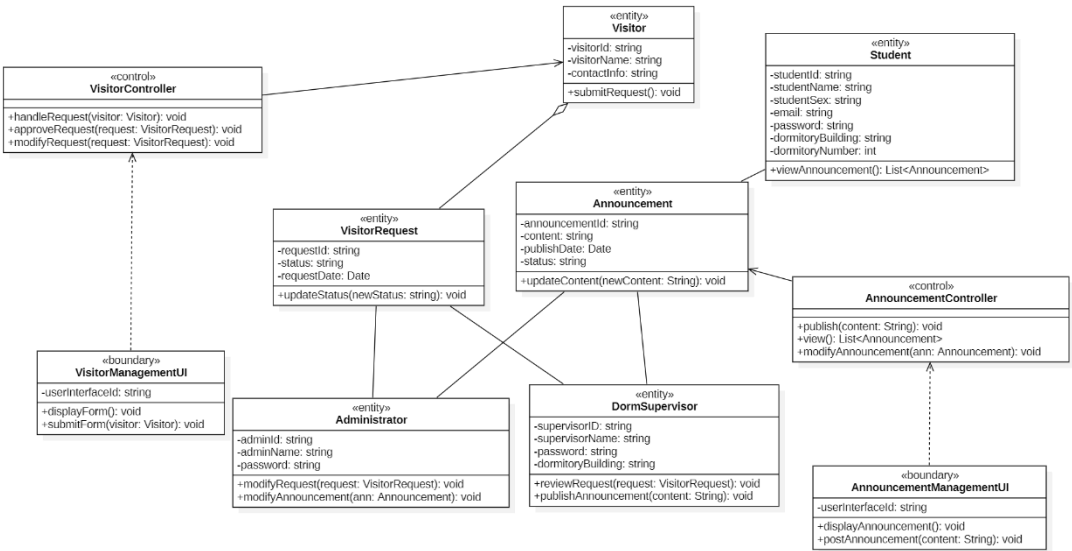
这张交互图描述了学生进行公共空间预约使用的登记流程。公共空间的预约需要学生查看可用空间的状态以及确定使用时间，成功预约之后的反馈结果将显示在学生端的相应界面上。



3.6 公告与安全管理子系统

3.6.1 类图

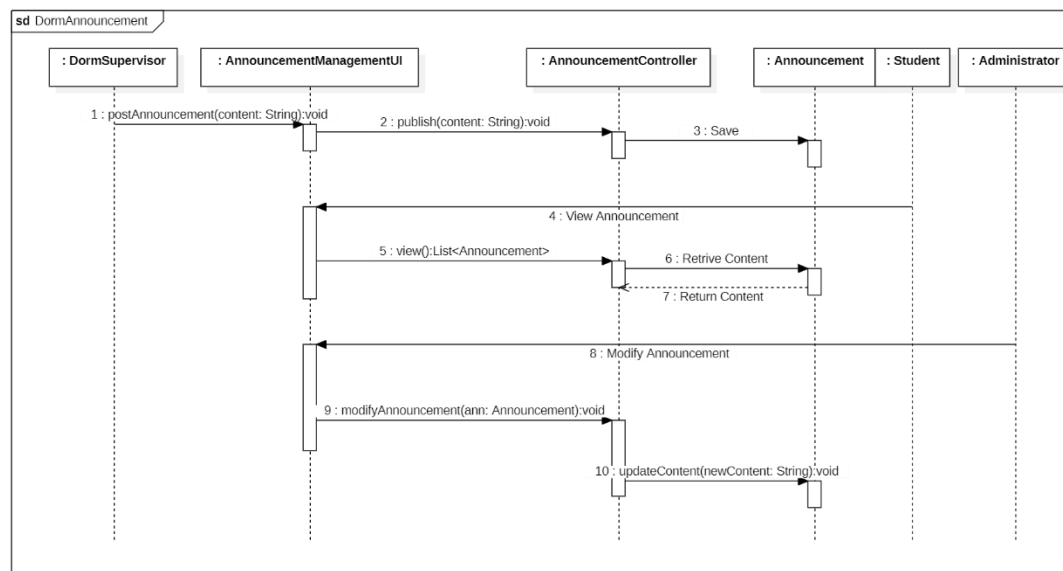
以下为公告与安全管理的类图。边界类有访客管理和公告管理的 UI 界面；控制类主要负责访客的申请以及公告的发布和查看；实体类 VisitorRequest 和 Announcement 分别负责管理访客申请和宿舍公告。



3.6.2 交互图

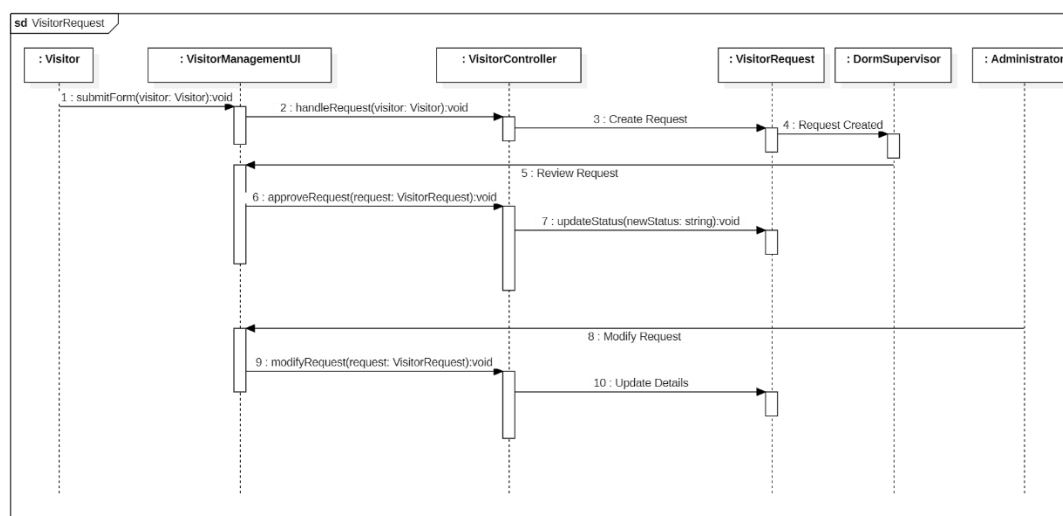
3.6.2.1 宿舍公告

这张交互图描述了宿舍公告的发布与查看的流程。宿舍公告由宿管发出，可以被宿管、学生以及管理员查看，管理员具有更高的权限，可以对所有的公告进行增删改查。



3.6.2.2 外来访客申请

这张交互图描述了外来访客申请的流程。外来访客需要填写表单，提交相关信息，宿管在收到申请之后决定是否允许访客进入宿舍，并将决定显示到访客的界面上。管理员具有更高的权限，可以对访客的申请记录进行增删改查。



4 需求更新

原用例表缺少扩展事件流、业务规则以及非功能性需求，为了使用例的描述更加准确，将部分关键用例表进行扩充如下。

4.1 “注册”用例表的更新

属性	描述
用例名	注册
编号	UC01
参与者	全体用户
前置条件	无
基本事件流	1. 用户进入注册页面 2. 填写必要的注册信息（如用户名、密码、邮箱） 3. 系统校验输入信息格式是否正确，若正确则创建用户账号 4. 默认注册为外来人员，完成注册
扩展事件流	3a. 若信息校验失败，系统提示用户重新输入 4a. 管理员可通过批量导入注册其他类型用户（如宿管）
后置条件	新用户账号存储在后台数据库中，状态为“已注册”
业务规则	1. 用户名和邮箱必须唯一，不得重复 2. 密码长度不少于 8 个字符，且必须包含字母和数字
非功能性需求	系统需在 1 秒内完成信息校验，注册页面加载时间不超过 2 秒

4.2 “宿舍信息查看”用例表的更新

属性	描述
用例名	宿舍信息查看
编号	UC03
参与者	学生、宿管
前置条件	用户已登录，且用户类型为学生或宿管

基本事件流	<ol style="list-style-type: none"> 1. 用户进入宿舍信息查看页面 2. 通过搜索功能定位目标宿舍 3. 查看宿舍的详细信息（如宿舍号、床位状态、入住人员）
扩展事件流	2a. 如果输入的宿舍信息无效，系统提示用户重新输入
后置条件	无直接数据修改，用户仅查看信息
业务规则	<ol style="list-style-type: none"> 1. 学生只能查看与自身相关的宿舍信息，不得访问他人宿舍信息 2. 宿管可以查看所有宿舍信息，包括空余床位状态
非功能性需求	查询功能响应时间不超过 3 秒，确保快速获取结果

4.3 “宿舍分配”用例表的更新

属性	描述
用例名	宿舍分配
编号	UC04
参与者	学生、宿管
前置条件	用户已登录，且用户类型为学生或宿管；分配的宿舍必须有空余床位
基本事件流	<ol style="list-style-type: none"> 1. 宿管开启宿舍分配功能，并设置分配规则（如按性别、年级等） 2. 学生进入宿舍分配页面，查看可用宿舍列表 3. 学生选择目标宿舍，并提交分配申请 4. 系统校验宿舍空余床位，若符合条件则完成分配，并通知学生和宿管
扩展事件流	<ol style="list-style-type: none"> 3a. 若目标宿舍无空余床位，系统提示学生重新选择 4a. 若分配规则冲突（如性别或年级不符），系统拒绝分配并通知学生原因
后置条件	<ul style="list-style-type: none"> - 后台宿舍信息更新（包括分配的学生信息、床位号） - 学生和宿管收到分配完成的通知
业务规则	<ol style="list-style-type: none"> 1. 宿舍只能分配给符合规则的学生，避免违规分配 2. 每个学生在一个学期内只能提交一次宿舍分配申请
非功能性需求	系统需在 10 秒内完成分配校验与信息更新，保证高效响应

4.4 “宿舍设备维修”用例表的更新

属性	描述
用例名	宿舍设备维修
编号	UC06
参与者	学生、宿管、维修人员、系统管理员
前置条件	用户已登录，且用户类型为学生、宿管或维修人员；目标设备必须已经在系统中登记
基本事件流	<ol style="list-style-type: none"> 1. 学生选择需要维修的设备，并提交报修申请，填写故障描述 2. 系统验证输入信息是否完整，若不完整提示补全 3. 宿管收到维修申请并核实设备故障是否属实 4. 核实后，宿管将任务分配给维修人员，指定维修时间 5. 维修人员接受任务并完成维修后，在系统中提交维修完成状态及备注 6. 系统通知学生维修完成并显示详情
扩展事件流	<ol style="list-style-type: none"> 3a. 若设备状态正常，宿管可直接拒绝维修申请，系统通知学生原因 4a. 若维修人员因故无法接受任务，宿管可以重新分配任务
后置条件	<ul style="list-style-type: none"> - 设备维修记录更新（包括维修时间、维修人员、故障描述、维修状态） - 学生和宿管收到维修完成的通知
业务规则	<ol style="list-style-type: none"> 1. 未完成的维修任务在分配维修人员后 24 小时内必须处理 2. 维修任务只能由宿管分配，维修人员不能主动接单
非功能性需求	系统响应时间不得超过 3 秒，任务分配和状态更新须实时同步

4.5 “宿舍缴费”用例表的更新

属性	描述
用例名	宿舍缴费
编号	UC07
参与者	学生、宿管、系统管理员
前置条件	用户已登录，且用户类型为学生或宿管
基本事件流	<ol style="list-style-type: none"> 1. 学生进入宿舍缴费页面，查看需缴费用明细（如水电费）

	2. 学生选择支付方式并完成缴费 3. 系统生成缴费记录，并更新数据库
扩展事件流	2a. 如果支付失败，系统提示用户重新选择支付方式并重试
后置条件	- 系统更新缴费记录，包括缴费时间、金额、支付状态等 - 宿管可在后台查看所有学生的缴费记录
业务规则	1. 缴费完成后，支付状态必须为“成功”，否则缴费无效 2. 宿管不得修改学生的缴费记录，仅能查看
非功能性需求	系统需在 5 秒内完成支付校验，支付结果需实时同步


4.6 “公共空间的使用”用例表的更新

属性	描述
用例名	公共空间的使用
编号	UC09
参与者	学生、宿管、系统管理员
前置条件	用户已登录，且用户类型为学生、宿管或系统管理员
基本事件流	1. 用户进入公共空间预约页面，查看可用时间段 2. 学生选择目标时间段并提交预约申请 3. 系统校验预约冲突，若无冲突则生成预约记录
扩展事件流	2a. 若时间段已被占用，系统提示用户选择其他时间段 3a. 系统管理员可根据特殊情况手动修改预约记录
后置条件	- 预约记录更新至系统数据库，包括时间段、申请人等信息 - 系统通知申请人预约结果
业务规则	1. 每位学生一天内只能预约两次公共空间，避免资源被垄断 2. 宿管和管理员可随时查看所有预约记录
非功能性需求	系统响应时间不超过 3 秒，预约冲突检测精确且高效

5 快照更新

5.1 访客登录界面快照的更新

访客登录页面是一个专为临时访客设计的简化登录界面，它允许用户无需复杂注册即可快速访问特定内容。该页面记录了访客登录时的基本信息和状态，确保访问安全的同时，也提供了便捷的一键登录体验，让内容分享更加流畅无阻。

该图展示了访客登录界面的快照。界面顶部有一个蓝色的标题栏，上面写着“访客申请”。下方是三个灰色的输入框，分别用于填写“您的姓名：”、“您的联系方式：”和“您的进楼原因：”。在输入框下方有一个黑色的“提交”按钮。

访客申请

您的姓名：

您的联系方式：

您的进楼原因：

提交

5.2 登记界面快照的更新

该系统页面集大件物品管理、公共物品监控与晚归晚出登记于一体，提供便捷的在线操作平台，实现物品借用审批、状态追踪及晚归晚出人员信息录入，确保秩序井然，提升管理效率与用户体验。

登记

公共物品使用登记

借用物品名称:

预计归还时间:

大件行李登记

大件行李件数:

大件行李出楼原因:

晚归晚出登记

属于晚归/出:

晚归/出时间:

具体原因:

提交

5.3 换寝申请界面快照的更新

学生换宿舍系统页面，提供一站式申请与审批流程。学生可轻松提交换宿请求，查看进度，页面设计直观，操作简便，确保宿舍资源高效配置，满足学生住宿需求，提升管理效率。

宿舍更换申请

您当前的宿舍:

更换宿舍原因:

是否与原室友已沟通:

您的意向宿舍:

选择此宿舍原因:

是否与新室友已沟通:

提交

6 待解决的问题(OPEN ISSUES)

6.1 智能分配算法的优化

目前的宿舍分配算法初步考虑了学生的生活习惯、学习时间和个人偏好，但算法的准确性和效率有待提升。下一阶段需要探索更先进的机器学习和数据分析技术，优化匹配算法，提高宿舍分配的公平性和满意度。

6.2 数据安全性与隐私保护

系统处理大量的学生个人信息和敏感数据，数据安全和隐私保护是必须解决的问题。需要设计和实施完善的安全策略，包括数据加密、访问控制和安全认证，确保符合相关法律法规要求。

6.3 系统性能与可扩展性

随着用户数量和数据量的增长，系统需要具备高性能和可扩展性。需要在架构上考虑负载均衡、分布式处理和缓存机制，确保系统在高并发场景下的稳定性和响应速度。

6.4 多平台兼容性和用户体验

为满足不同用户的使用习惯，系统需要支持多种设备和平台。需要优化前端设计，确保在 PC、手机和平板等设备上都有良好的用户体验，同时提升界面的友好性和易用性。

6.5 实时监控与预警机制的实现

门禁安全和设施设备的状态需要实时监控和预警。需要研究高效的数据采集和处理方法，建立实时报警系统，及时发现和处理异常情况，保障宿舍安全。

6.6 第三方服务集成

系统涉及多渠道支付、短信通知等功能，需要集成第三方服务。需要解决与第三方平台的接口兼容性、稳定性和安全性问题，确保服务的顺利对接和可靠运行。

6.7 用户反馈机制和迭代更新

建立有效的用户反馈渠道，收集用户在使用过程中的问题和建议。根据反馈不断优化和迭代系统功能，提升系统的实用性和用户满意度。

参考文献

[1]. 《微服务架构设计模式》 - 作者：CHRIS RICHARDSON

本书深入探讨了微服务架构的设计原则和实践模式，对于我们的项目有直接的指导意义。书中详细介绍了如何将大型单体应用拆分为一系列小型、独立部署的微服务，这与我们系统的分层架构设计高度契合。通过学习微服务的通信方式、数据管理和部署策略，我们能够有效地降低系统耦合度，提高开发和维护效率。此外，书中对于微服务安全性和事务管理的讨论，为我们处理跨服务的数据一致性和安全问题提供了宝贵的参考。

[2]. 《用户体验要素：以用户为中心的产品设计》 - 作者：JESSE JAMES GARRETT

该书全面阐述了用户体验的五大要素，从战略层到表现层，为我们的展示层和应用层设计提供了理论基础。通过理解用户需求和行为，我们在界面设计中注重信息架构和交互设计，提升系统的易用性和美观度。书中强调的以用户为中心的设计理念，指导我们在功能实现的同时，注重用户的情感体验和满意度，为系统的成功奠定了基础。

[3]. 《高性能 MySQL（第 3 版）》 - 作者：BARON SCHWARTZ 等

本书深入探讨了 MySQL 数据库的性能优化、架构设计和集群管理，对我们数据持久化层和数据库层的设计具有重要意义。通过学习索引优化、查询优化和数据库架构，我们能够提升数据读写效率，满足高并发场景的需求。书中关



于分区、复制和集群的内容，为我们设计可扩展性强、可靠性高的数据库解决方案提供了实用指导。

[4]. 《RESTFUL WEB APIS》 - 作者：LEONARD
RICHARDSON, MIKE AMUNDSEN, SAM RUBY

该书详细介绍了 RESTful 架构风格的 API 设计原则和实践方法，对于我们用户服务层与其他层之间的接口设计有直接的参考价值。通过遵循 RESTful 的设计原则，我们能够创建统一、简洁和易于维护的 API 接口，促进各模块之间的解耦和协作。书中对于状态管理、安全性和性能优化的讨论，帮助我们设计出高效、安全的服务接口。

[5]. 《软件工程（第 10 版）》 - 作者：IAN SOMMERVILLE

这本经典的教材涵盖了软件工程的各个方面，包括需求分析、系统设计、实现、测试和维护。它为我们的项目提供了全方位的方法论指导。特别是在需求建模、架构设计和测试策略方面，书中提供的实践准则和案例分析，帮助我们规范项目流程，提升软件质量和团队协作效率。

团队成员贡献

在完成这项任务的过程中，所有团队成员都积极参与讨论，并努力完成各自的任务。我们的团队成员合作融洽，遇到问题时能及时沟通和解决。小组内的分工均匀且明确，具体如下：

- ✿ 项目简介：张扬
- ✿ 架构设计：周政宇、张立伟
- ✿ 分析模型：
 - ◇ 用户管理子系统：张扬
 - ◇ 宿舍信息子系统：张扬
 - ◇ 设备和维修管理子系统：张扬
 - ◇ 缴费管理子系统：奥泉瑞
 - ◇ 公共资源管理子系统：奥泉瑞
 - ◇ 公告与安全管理子系统：奥泉瑞
- ✿ 需求更新：奥泉瑞
- ✿ 快照更新：张立伟
- ✿ 待解决的问题：周政宇
- ✿ 团队成员贡献：周政宇
- ✿ 文档排版：周政宇

成员分配权重如下：

学号	姓名	权重
2252042	周政宇	100%
2253722	张扬	100%
2252144	张立伟	100%
2250695	奥泉瑞	100%