1 概述

1.1 系统简述

本系统是基于HTML, CSS, JavaScript和Python Fastapi框架开发的Web应用程序。其中，前端的浏览器在客户端供用户使用，前端发送的HTTP请求被后端接收后，先由视图层处理再转发至模型层，通过ORM调用数据库中的数据，再反馈给前端。

系统包含以下两个子系统：

* + 用户信息管理
  + 宿舍信息管理

1.2 软件设计目标

1.2.1 使学生更快捷地申请宿舍业务

1.2.2 使宿舍管理员更方便地处理宿舍业务

1.2.3 支持数据的批量处理

1.3 参考资料

1.4修订版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **描述** | **作者** |
| <9/6/2021> | <1.0> | 关于宿舍管理系统的初步详细设计 | 小组所有成员 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

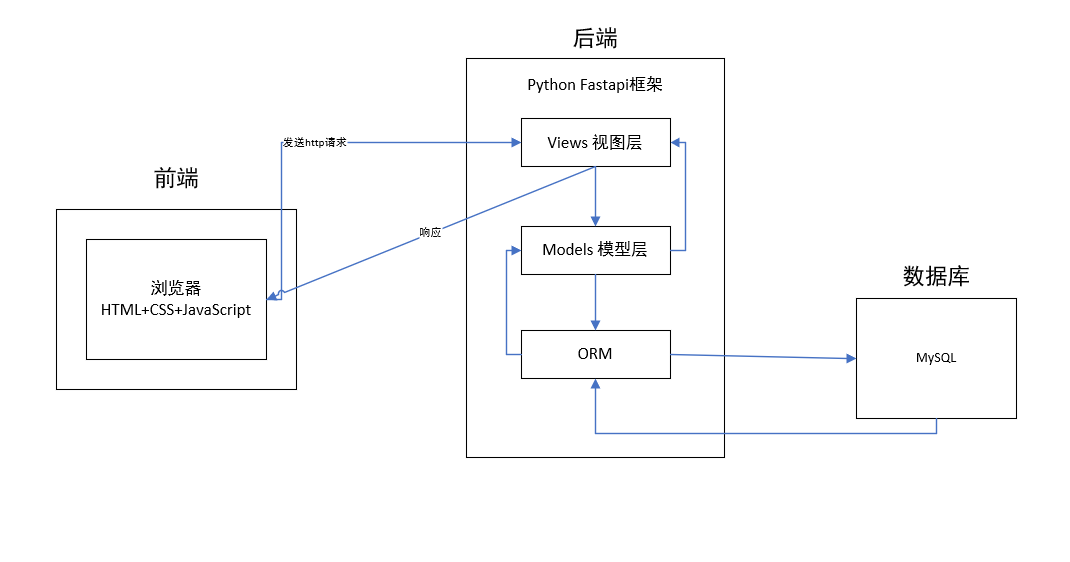
2 术语表

请参考需求分析文档。

3 设计概述

3.1 系统的复用计划：库、框架、模式、构件等方面的复用（如果没有就不写了）

Python Fastapi框架



3.2系统接口设计(根据概要设计进一步细化设计)

• 各种提供给用户的界面

**1. 用户信息管理系统登录界面**

用户（学生/宿舍管理员/系统管理员）在该界面输入用户名及密码，点击“登陆”按钮后进入系统主界面。点击“重置”按钮可以清空输入框内已输入的内容。



**2.学生界面**

本界面为用户信息管理系统的主界面，本系统的各种功能以按钮的形式直观地呈现给使用者，其中包括：“查询信息”、“宿舍业务”、“申请入住”、“申请离校”、“查看申请进度”和“退出登录”。点击“退出登录”按钮即跳转回登录界面



2.1查询信息界面——学生

在系统主界面点击“查询信息”按钮进入本界面。学生可在本界面查询到自身的学号、学院、班级、宿舍等基本信息。点击“返回”按钮即可回到系统主界面。



进入界面后点击“查询”按钮，学生信息就会显示在内联框架中。



2.2宿舍业务界面

在系统主界面点击“宿舍业务”按钮进入本界面。点击“返回”按钮回到系统主界面。



学生通过下拉列表选择要办理的宿舍业务：水电缴费或公物报修。



如果在下拉列表中选择了“水电缴费”，界面中出现缴费类型的下拉列表，可选择缴纳水费或电费。点击“返回”按钮即回到系统主界面。



如在下拉列表中选择了“公务报修”，界面中显示输入框让学生输入公物名称。



2.2.1宿舍业务申请成功界面

在宿舍业务界面选择要办理的业务后点击“确认”按钮即跳转到本界面。点击“返回”按钮即返回宿舍业务界面。



2.3申请入住界面

学生可在未住宿的状态下点击系统主界面的“申请入住”按钮进入本界面。点击“提交申请”即可提交宿舍入住申请。点击“返回”按钮即回到系统主界面。



2.3.1申请入住成功界面

在申请入住界面点击“提交申请”按钮即进入本界面。点击“返回”按钮即返回系统主界面。



2.4申请离校界面

学生点击系统主界面的“申请离校”按钮进入本界面。学生点击“提交申请”即可提交离校申请。点击“返回”按钮即回到系统主界面。



学生需在本界面通过下拉列表选择离校原因：包括毕业、请假和其他。



如学生选择了“其他”，界面中会出现输入框让学生填写离校原因。



学生需通过下拉列表选择离校时间。



2.4.1离校申请提交成功页面

在申请离校界面填写好离校信息后点击“提交申请”按钮即可进入本界面。点击“返回”按钮回到系统主界面。



2.5查看申请进度界面

学生点击系统主界面的“查看申请进度”按钮进入本界面。点击“返回”按钮回到系统主界面。



学生可在业务类型的下拉列表中选择想查看的业务进度，包括离校申请和宿舍申请。点击“查看”按钮后即可在内联框架中查看到业务进度。

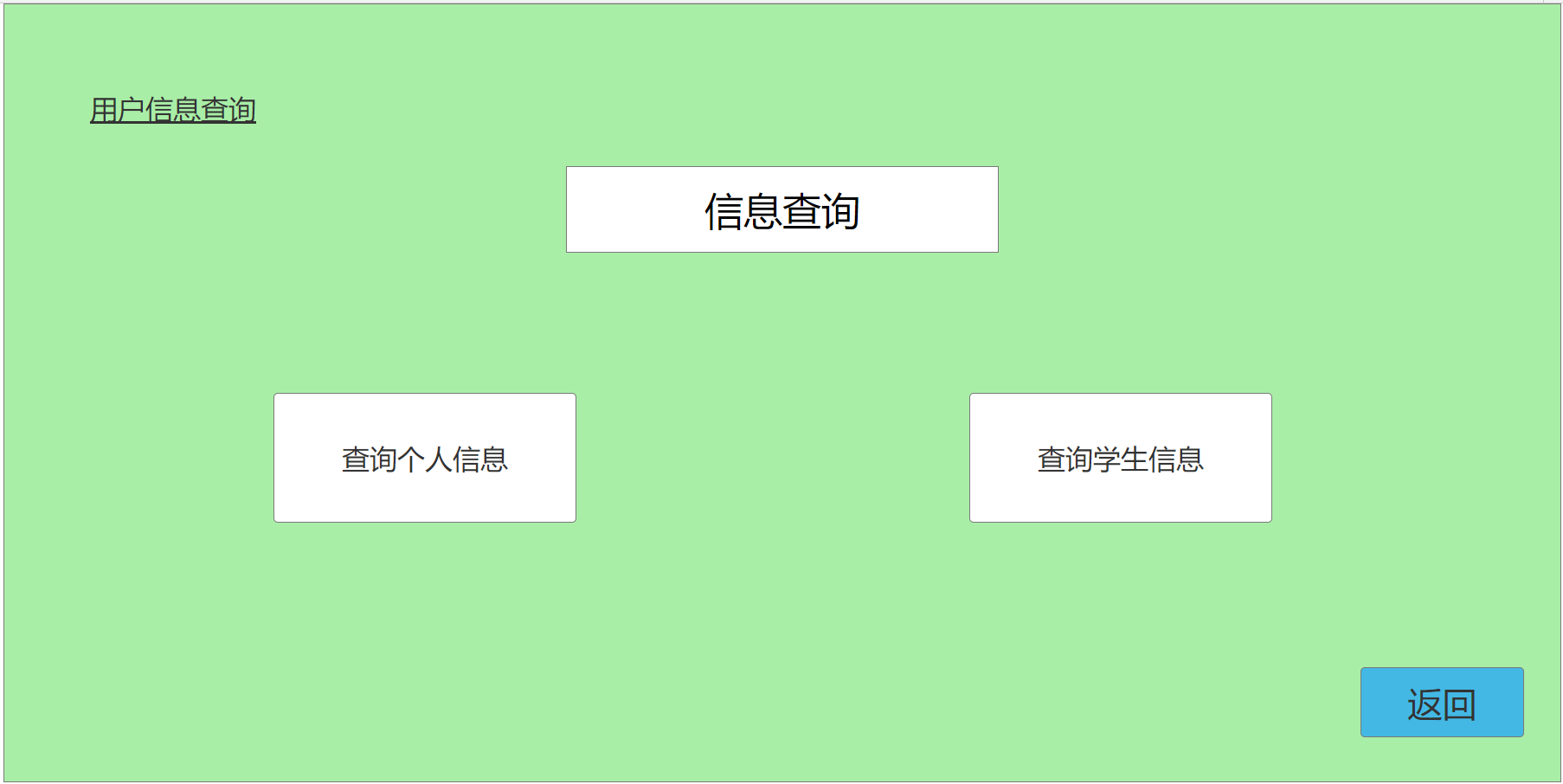


**3．宿舍管理员界面**

3.1 宿管管理的总界面：普通宿管可以用鼠标点击查询宿舍信息、查询用户信息、处理宿舍业务按钮来进入对应界面；点击返回按钮返回至**登录界面**；总宿管能额外有处理离校申请、处理入住申请按钮。



3.2 查询用户信息界面：宿管点击查询个人信息或查询学生信息来进入对应界面；点击返回按钮返回至**宿管管理总界面**。



3.3 查询个人信息界面：宿管可以点击返回按钮返回至**查询用户信息界面。**



3.4 查询学生信息界面：用户可以用鼠标选择年级、学院、性别、班级或用键盘输入学号、姓名；然后点击查询按钮，系统将返回查询的学生信息在下方。

点击返回按钮可以返回至**查询用户信息界面**。



3.5 处理入住申请界面：宿管点击确认分配按钮可以为单个学生分配宿舍，系统将弹出分配成功或者分配失败的信息；点击一键分配可以为所有申请学生分配宿舍；点击返回按钮将返回至**宿舍管理总界面**。



3.6 处理离校申请界面：宿管可以点击拒绝按钮，然后在弹窗中用键盘输入拒绝原因；点击返回按钮可以返回至**宿舍管理总界面**。



3.7宿舍管理员处理宿舍业务界面

该查询模块的输出数据为固定结构的数据项，将管理员填写的查询条件或填写的信息封装在固定的数据类型中传递给被调用的数据库模块。结构基本就是数据的存储结构,具体可见需求分析报告。



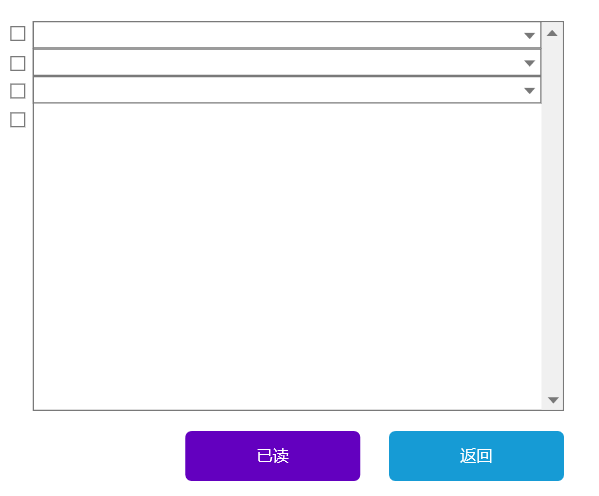
3.7.1宿舍管理员查询与处理报修请求界面：宿舍管理员选中需要处理的报修请求，点击“批量审批”按钮，请求提交给第三方，等待报修；宿舍管理员点击“批量完成按钮”，系统提示相关宿舍报修已完成。



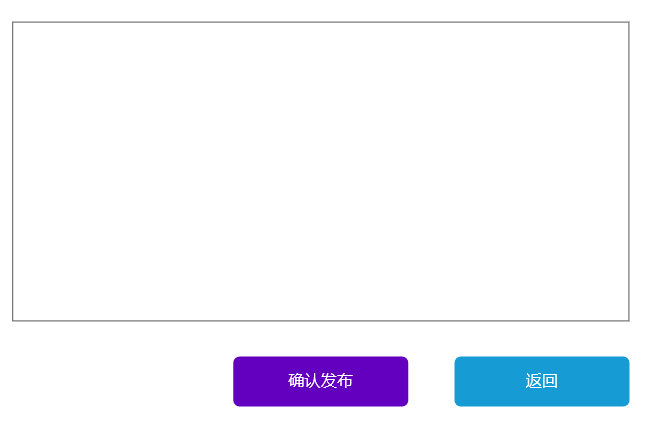
3.7.2宿舍管理员查询学生考勤信息界面：页面显示当天23：30时所有学生的归寝信息，宿管可选中未归寝学生批量导出信息上报。



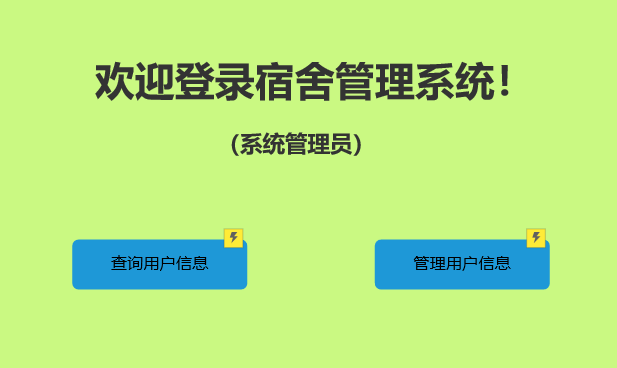
3.7.3通知界面：宿舍管理员在该界面可看到通知消息，并批量选中为已读状态。



3.7.4发布公告界面：宿舍管理员编辑发布的公告内容，编辑完成后点击“确认发布”。



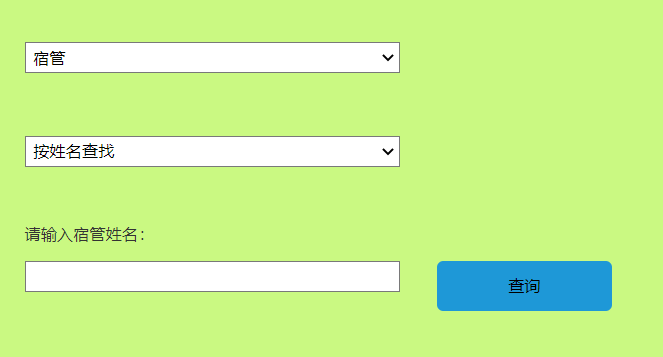
**4.系统管理员界面**



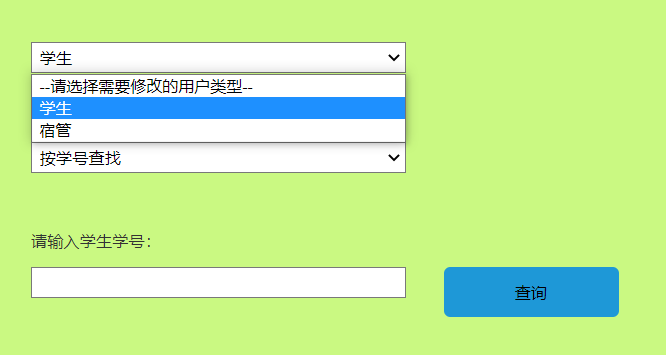
4.1 系统管理员总界面：系统管理员登录后，系统跳转至此界面，可以执行查询用户信息和管理用户信息操作。

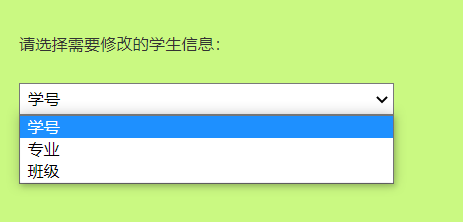
4.2 查询用户信息界面：系统管理员可以选择查询的用户类型，并根据用户类型的具体属性进行查询。



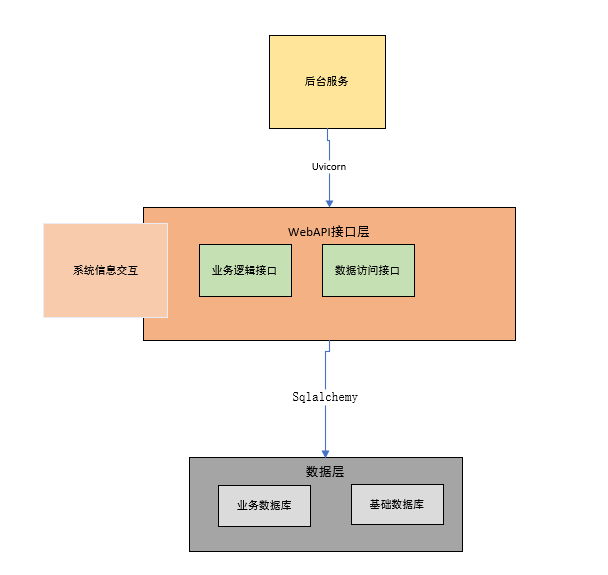


4.3 修改用户信息界面：系统管理员先查找到需要修改的用户，再选择具体属性进行修改。





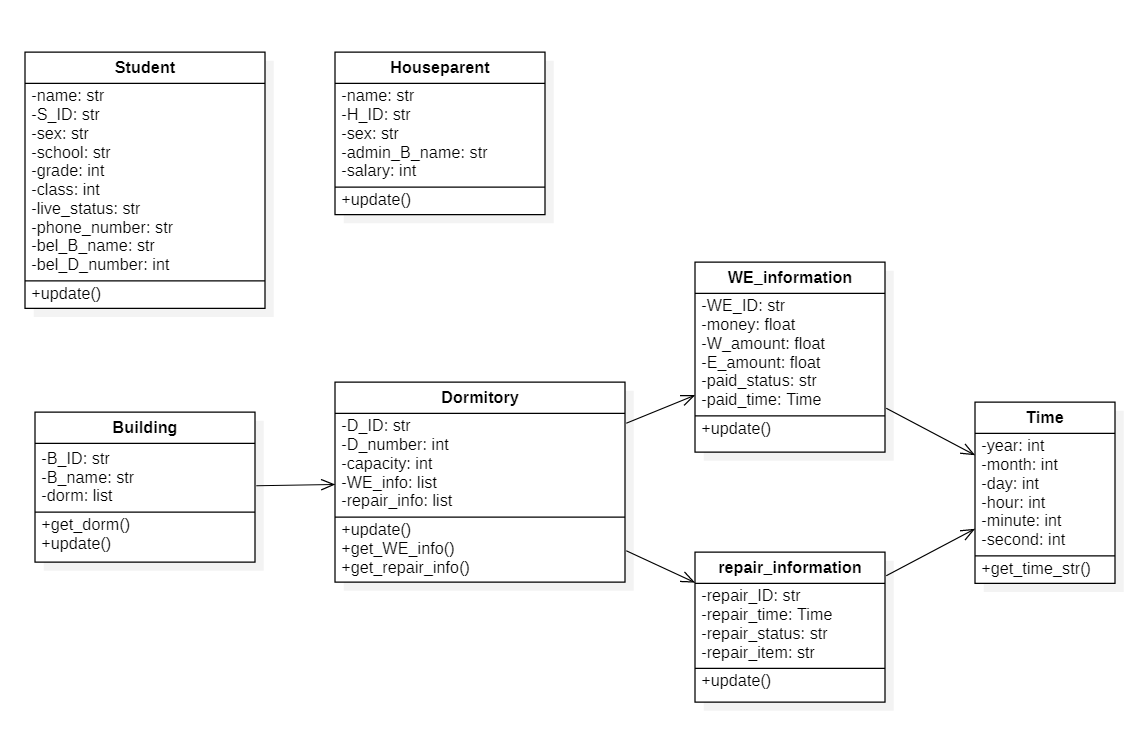
• 系统外部接口设计：与外部系统的交互设计



3.3 对象模型设计

• 对象模型（类图或对象图）：提供整个系统的对象模型，在其中应该包含

所有的系统对象。所有对象之间的关联必须被确定并且必须指明联系的基数。

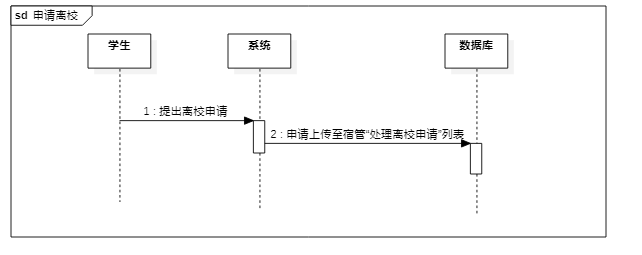


• 对象描述: 在这个部分叙述每个对象的细节，它的属性、它的方法。对每个对象的每个属性详细说明：名字、类型; 对每个对象的每个方法详细说明：方法名，返回类型，返回值，参数，用途以及使用的算法的简要说明。

3.4系统用例实现详细设计

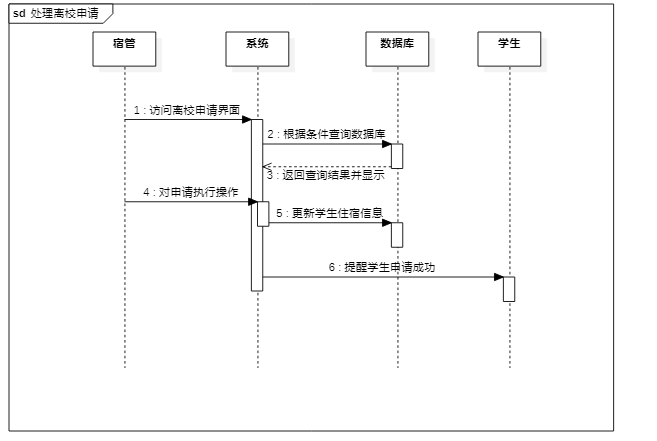
3.4.1 申请离校：

– 顺序图：



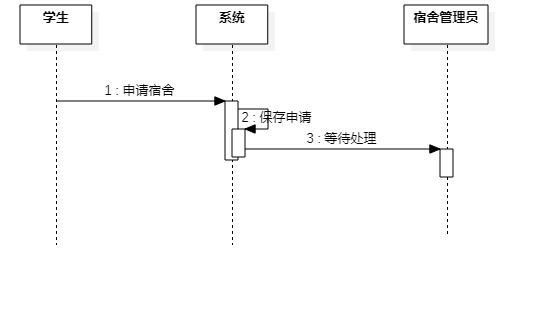
3.4.2 处理离校申请：

– 顺序图：



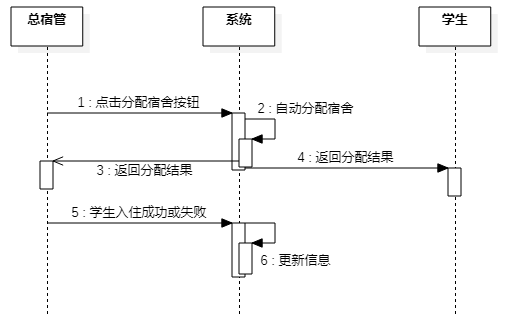
3.4.3 申请入住

– 顺序图：



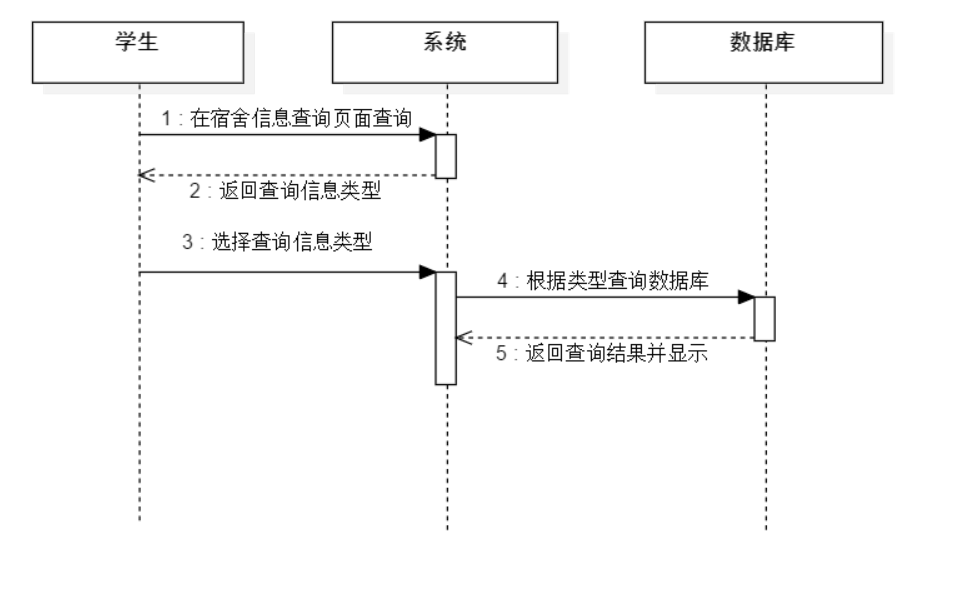
3.4.4处理入住申请

– 顺序图：



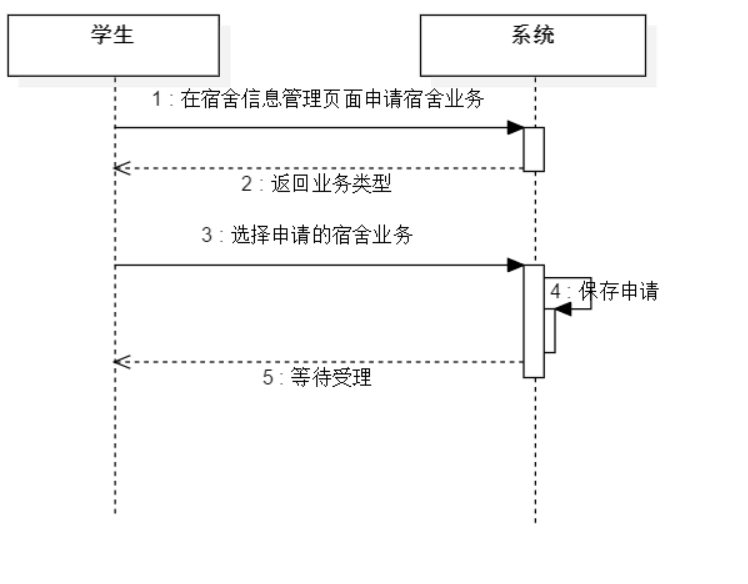
3.4.5 查询宿舍信息

– 顺序图：



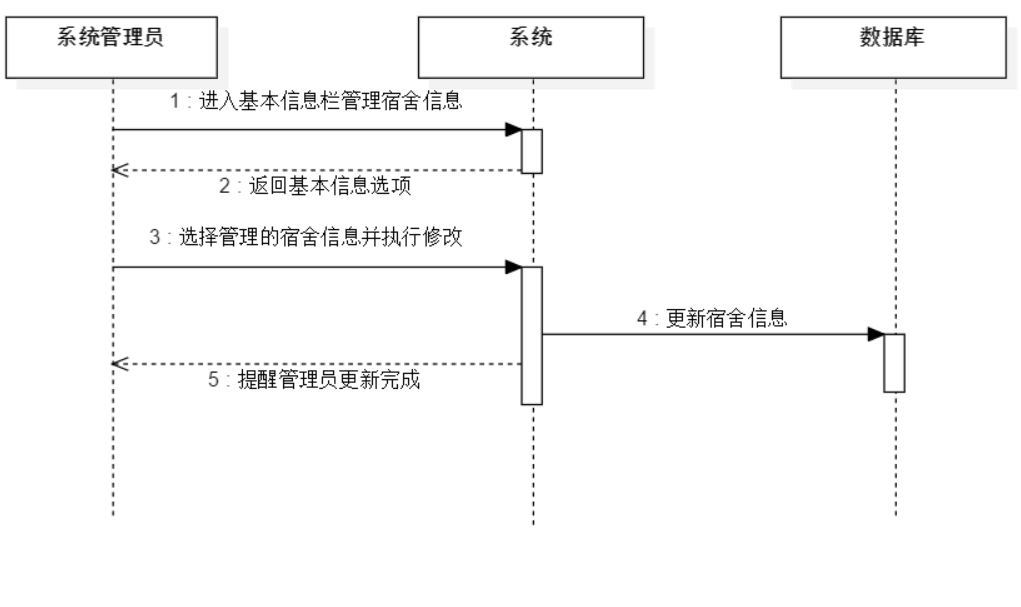
3.4.6 申请宿舍业务

– 顺序图：

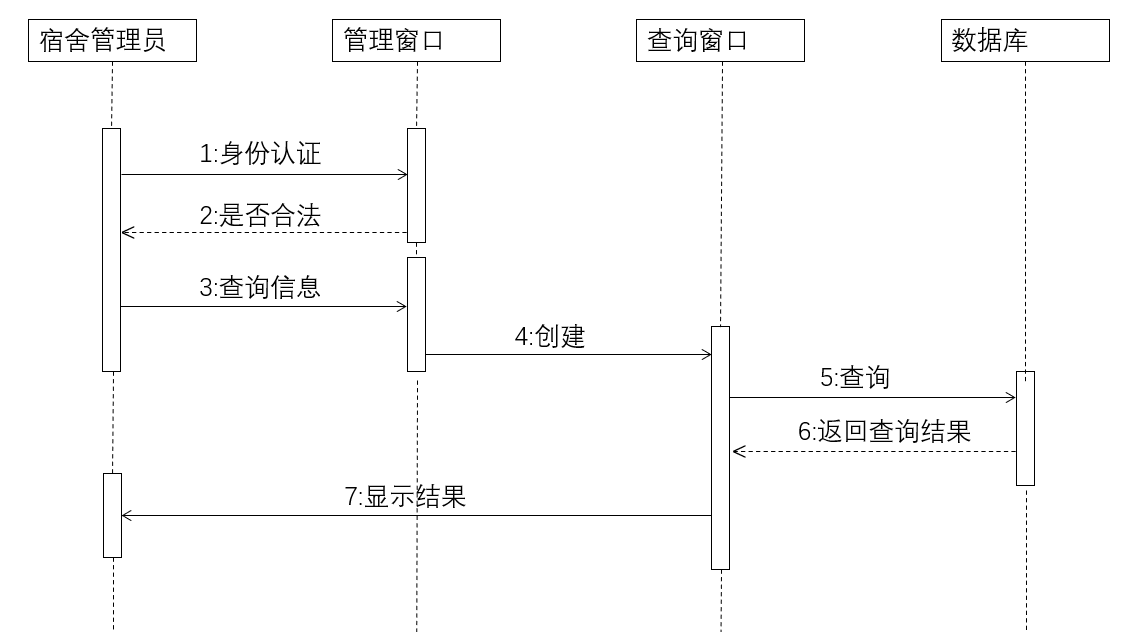


3.4.7 管理宿舍信息

– 顺序图：

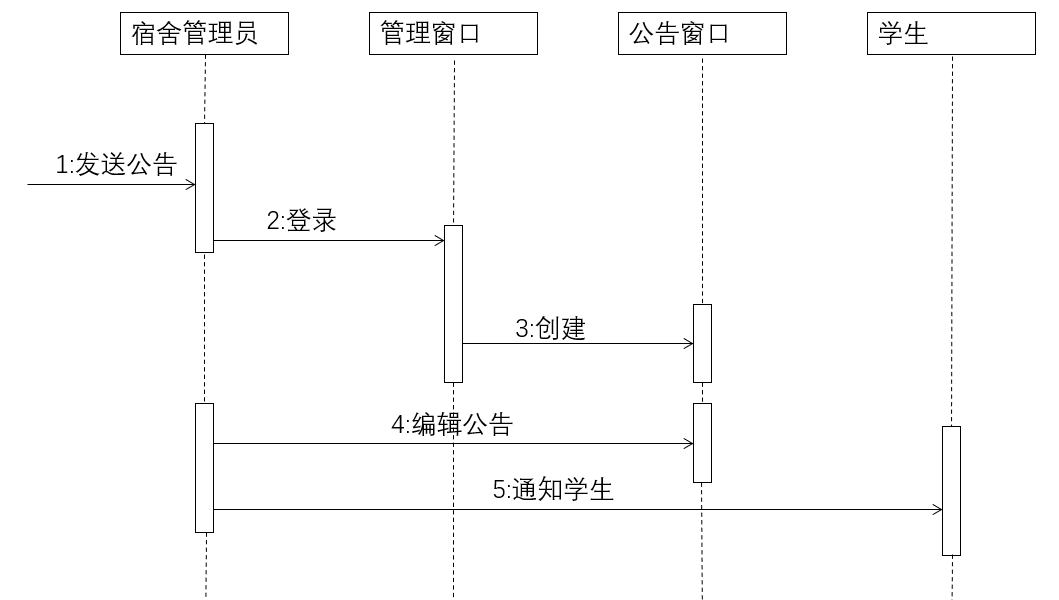


3.4.8 查询宿舍信息：

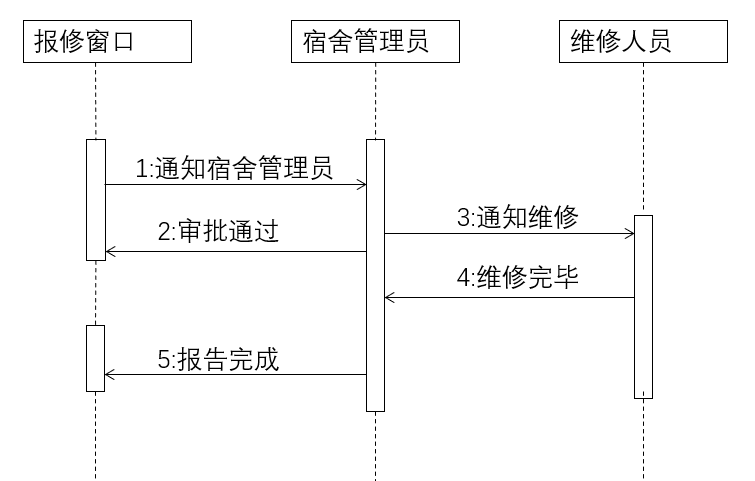


3.4.9 处理宿舍业务：

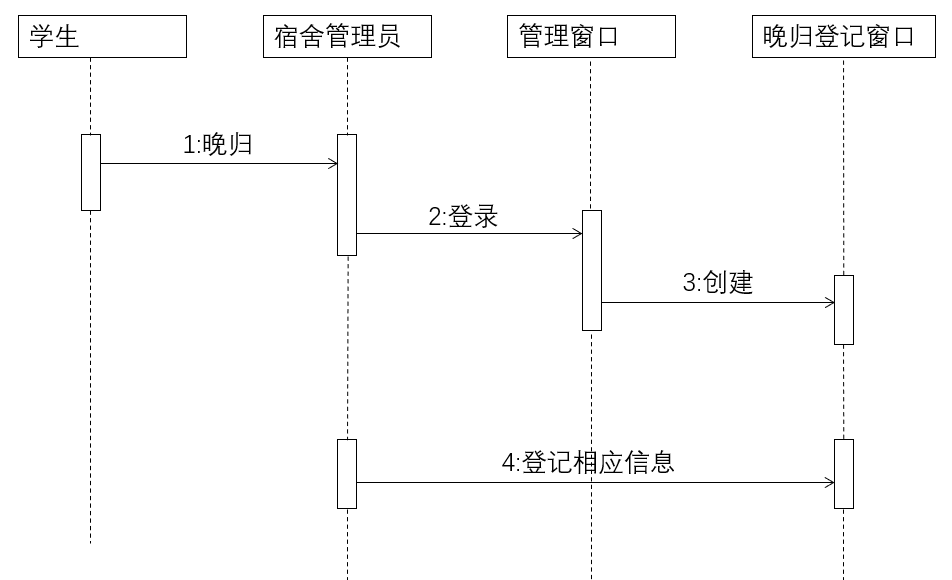
3.4.9.1发布公告



3.4.9.2 处理报修信息



3.4.9.3 登记晚归



3.5系统非功能设计

3.5.1满足系统性能需求：

3.4.1.1使用连接池。连接池技术能够重用到数据库的连接，而不是每次请求都建立新的TCP连接，新连接仅在于连接池中得不到连接时才建立。当连接被关闭时，它被返回到连接池中。

3.4.1.2使用缓存技术。

3.5.2满足系统可靠性需求：

3.4.2.1在软件需求分析阶段，制定详细的需求分析计划和需求分析方法，保证需求分析彻底、完整、准确、可靠。

3.4.2.2设计过程中保证需求分析和说明定义的一致性、完整性、准确性和无歧义性，进而保证开发过程中对需求容易辨识。

3.4.2.3进行详细的数据组织规划和逻辑设计，保证软件设计过程中的数据组织的准确性和逻辑的可靠性，进而防止软件开发设计过程中出现过多的逻辑错误和数据描述错误。

3.4.2.4选择可靠有效的建模语言，系统设计整个过程中保证先建模后验证再编码，有效保证每个编码的质量。

3.5.3满足系统易用性需求：

3.4.3.1在系统开发过程中遵循迭代优化式开发模式，通过详细设计与代码开发是作用与反作用的关系来提高系统易用性。

3.4.3.2采取界面内容可见即可用的设计原则，在详细设计时，将系统各项操作与权限角色进行关联，无权操作内容不应出现在用户界面上，且不应出现用户需根据个人角色信息选择不同按键的操作（用户有双重角色除外）。