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可维护性

当日后代码需要修改逻辑的时候，只需要修改某些类或者增加某些类，对主程序和大部分代码不需要修改，体现为可维护性。

1.2.2可复用性

将一定的代码进行封装，在多处可以重复使用。

1.2.3可扩展性

当增加新的功能模块时，只需要增加某些代码，对整体代码架构不需要做太多修改。

1.2.4高灵活性

通过封装、继承、多态把程序的耦合度降低，耦合度降低后，使代码的使用更加灵活，更容易修改，易于复用。

1.3 参考资料

1.4修订版本记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **描述** | **作者** |
| <2/6/2021> | <1.0> | 关于宿舍管理系统的概要设计 | 小组所有成员 |
| <9/6/2021> | <2.0> | 对系统的逻辑架构、物理架构和开发技术架构做了一定修改 | 小组所有成员 |
|  |  |  |  |
|  |  |  |  |

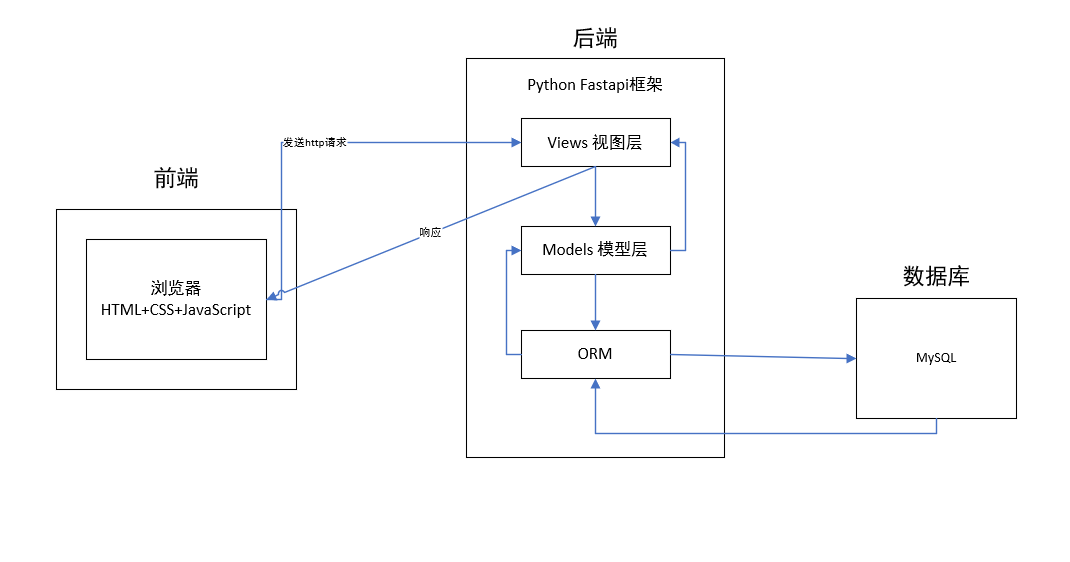
2 术语表

请参考需求分析文档

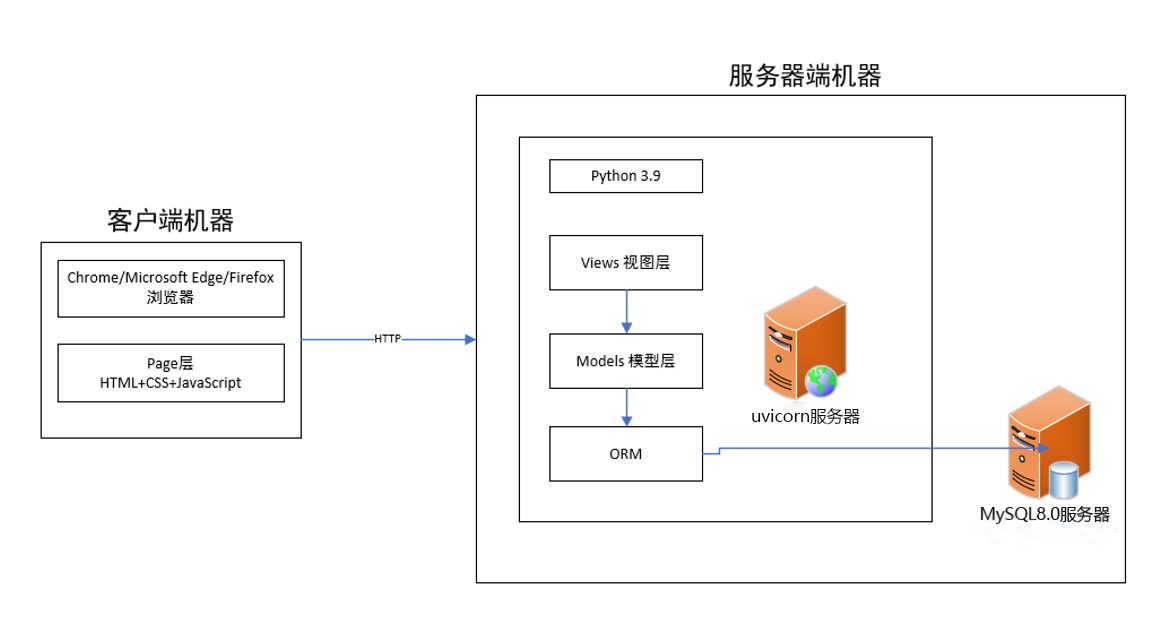
3 设计概述

3.1 系统架构设计

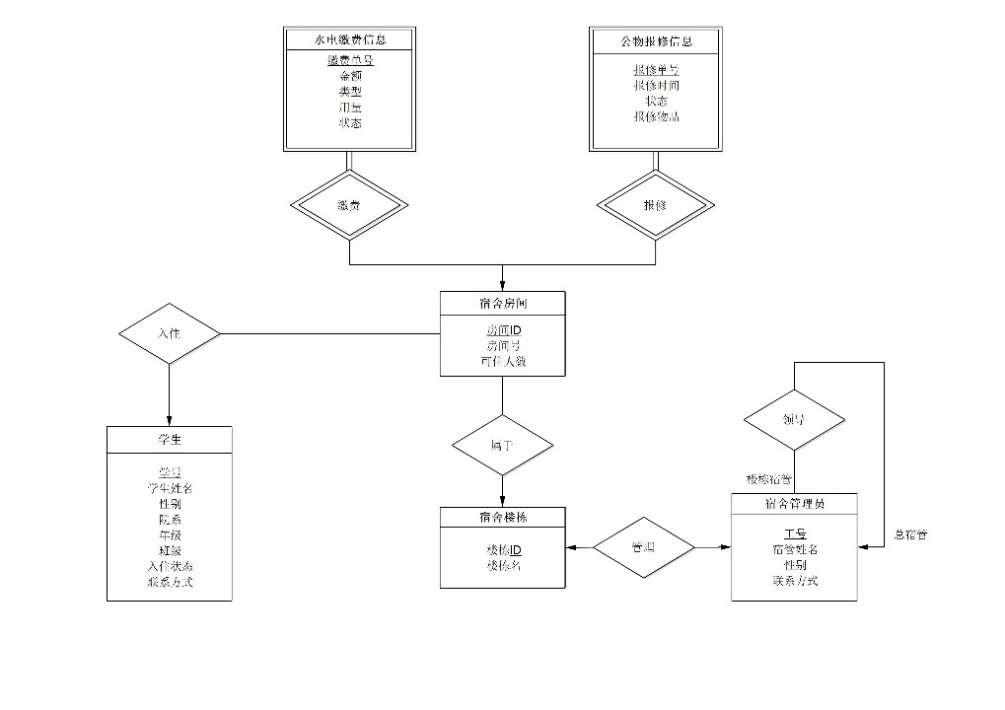
1.系统逻辑架构（软件的技术架构）：软件运行技术架构



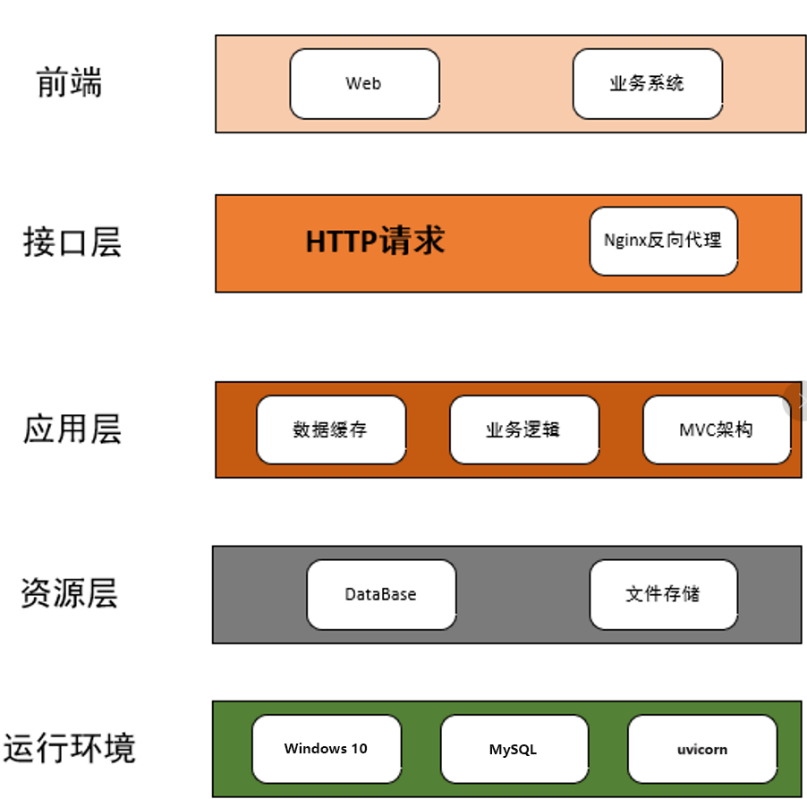
2.系统物理架构（物理设备架构图：软件如果部署到物理设备上）



3.系统数据模型（系统ER模型）



4.系统开发架构：系统开发技术架构



3.2 系统接口设计

**用户接口**：

1.登录界面：用户用键盘输入学号或者工号，系统输出成功登录后的界面

2.查询操作：用户用鼠标点击查询按钮，系统输出查询信息

3.修改操作：用户用鼠标需要点击需要修改属性，再键盘输入修改值，用鼠标点击确认，系统输出修改成功信息

4.删除操作：用户用鼠标选择删除的信息以及确认，系统输出删改成功的信息

5.添加数据操作：用户用鼠标点击添加数据按钮，用键盘输入数据，系统输出添加成功信息  
**外部接口：**

硬件接口：无

软件接口：

1.使用Sqlalchemy与数据库MYSQL连接

2.与uvicorn服务器接口连接

3.与第三方支付（微信/支付宝）连接

**内部接口**：程序内的各个模块之间采用函数调用、参数传递、返回值得方式进行信息传递。

3.3 约束和假定 （描述系统设计中最主要的约束，如果没有可以不写等等）。

本系统主要针对学生宿舍情况进行设计,其最终用户是宿舍系统的管理员、宿舍管理员以及宿舍学生，他们都具备一定的计算机使用能力。同时系统的开发人员是计算机专业人员,熟悉数据库、操作系统。该系统需要在6月底之前开发完成，保证系统的按时交付,实现用户的全部需求。

编程语言：python

硬件条件:

客户端：Intel Core-i5、GTX 1650（或同等配置的其他产品）、8G内存、256G磁盘空间

Tomcat服务器：Intel Core-i7、GTX 1660ti、8G内存、256G磁盘空间

MySQL服务器：Intel Core-i7、GTX 1660ti、8G内存、256G磁盘空间

运行环境:Windows10

数据库：MySQL

3.4非功能性设计

3.4.1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。

3.4.2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。

3.4.3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）。