成　绩	
评阅人	

复 旦 大 学

研 究 生 课 程 论 文

论文题目：　软件过程管理期末论文

修读课程：　软件过程管理（SOFT620011）

选课学期：　2023 年春季学期

选课学生：　杨子恒（22210240350）

完成日期：　20230620

目 录

一、 项目介绍及负责任务

1 项目介绍

本项目为自习室预约系统，其主要目的为以高等学校日常学习生活为背景，通过信息技术手段实现一个自习室座位预约系统，提升各类自习室座位使用率。项目实现的功能如图一展示。

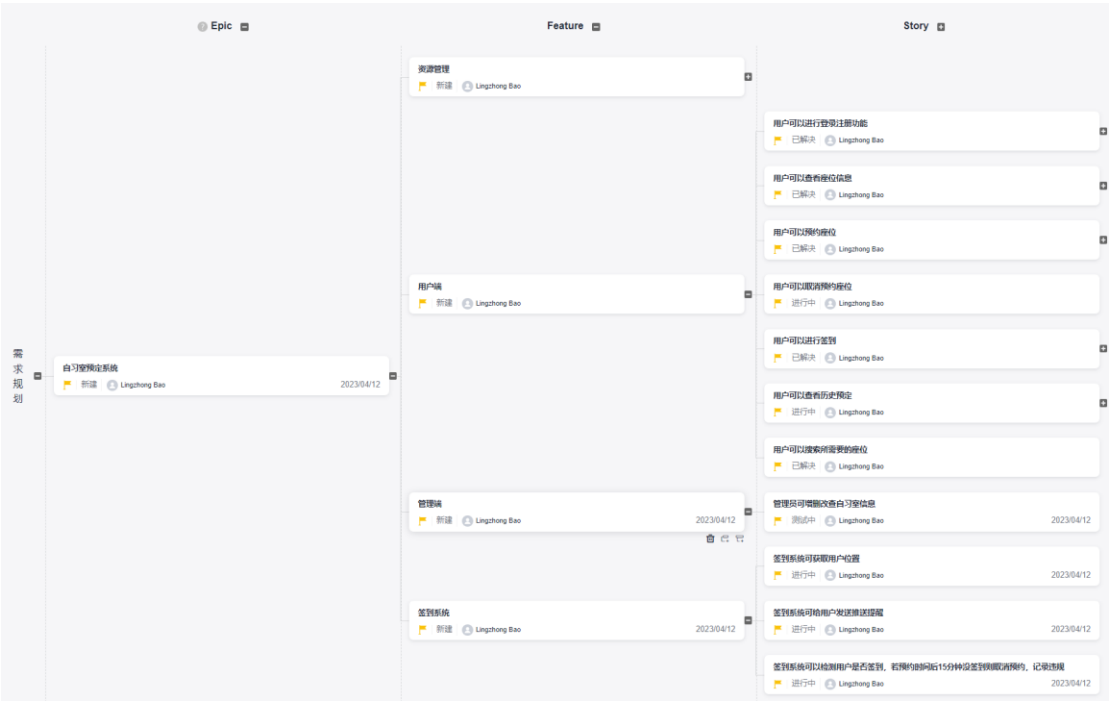

图 1 用户故事

本项目采用前后端分离开发的模式，前端使用 Vue 框架，后端使用 Flask 框架，数据库使用 MongoDB。使用 Docker 部署在阿里云服务器上，可通过 docker compose up 一键启动服务。通过 pytest 实现自动化测试。

2 本人负责的任务

本人主要负责两个任务，一是使用 docker 工具将项目部署到服务器端。二是负责管理端的前端页面开发。下面将详细介绍我的工作部分。

二、 Docker 部署部分

1 使用 Docker 的目的

Docker 是一种开源的容器化平台，可以帮助开发人员和系统管理员更轻松地构建、部署和运行应用程序。它具有许多好处，包括以下几点：

1、轻量和可移植性：Docker 容器非常轻量，因为它们与宿主操作系统共享内核，可以共享相同的系统资源，而无需为每个容器分配大量的内存。此外，Docker 容器是可移植的，可以在任何支持 Docker 的环境中运行，无论是开发机、测试环境还是生产服务器。

2、高度可扩展：使用 Docker，可以根据需求快速扩展应用程序的容量。通过 Docker 的自动化部署和管理功能，可以轻松地在集群中创建和部署多个容器实例，并根据需要进行水平扩展。

3、环境一致性：Docker 容器包含了应用程序的所有依赖项，包括操作系统、库和运行时环境。这意味着您可以确保应用程序在不同的环境中以相同的方式运行，避免了“在我的机器上可以工作”这样的问题。

4、快速部署和启动时间：与传统的虚拟机相比，Docker 容器的启动时间非常短。Docker 使用容器化技术，在几秒钟内启动一个容器，并且可以根据需要快速部署和销毁容器实例，从而加快了开发和部署的速度。

5、简化的管理：Docker 提供了一套强大的管理工具，使您可以轻松地管理和监控容器。您可以使用 Docker 命令行界面或图形用户界面来管理容器、映像、网络和数据卷等。

6、资源隔离和安全性：Docker 使用 Linux 内核的特性来提供容器间的资源隔离，使每个容器都运行在自己的独立环境中。这意味着即使一个容器中的应用程序发生故障或被攻击，它也不会影响其他容器或宿主操作系统的稳定性和安全性。

总的来说，Docker 提供了一种方便、灵活和高效的容器化解决方案，可以加速应用程序的开发、部署和扩展过程，并提供一致的运行环境和资源隔离，从而帮助开发人员和系统管理员更好地管理和交付软件。

2 使用 Docker 的过程

本项目把前端代码、后端代码、数据库分别 build 为三个镜像，其中前端代码和后端代码通过 Dockerfile 生成镜像，数据库使用官方镜像。部署时在三个容器中运行。下面展示了相关的代码信息。

```
FROM python:3.8
ADD . /classroom/
ADD requirements.txt .
ENV PYTHONPATH /classroom/
WORKDIR .
RUN pip install -r requirements.txt
```

图 2 后端 Dockerfile

```
FROM nginx

MAINTAINER yzh

RUN rm /etc/nginx/conf.d/default.conf

ADD default.conf /etc/nginx/conf.d/

COPY dist/ /usr/share/nginx/html/
```

图 3 前端 Dockerfile

在启动服务时可通过 docker compose up 一键启动服务。其 yml 文件如图 4 所示。

```
version: '3.1'
services:
  mongodb:                    # 服务名称，自定义，不能有重复，本服务是docker hub已创建服务
    container_name: mongo_classroom    # 容器名称，自定义，不能有重复
    image: mongo              # 镜像，不指定版本，从hub上拉取latest版本，指定版本mongo:版本号
    ports:                    # 宿主机port:docker容器port，外部访问要通过宿主机port
      - 8017:27017
    environment:              # 设置docker容器环境变量，根据hub 镜像说明
      - MONGO_INITDB_ROOT_USERNAME=root
      - MONGO_INITDB_ROOT_PASSWORD=123456
  flask:                              # 自建flask服务
    container_name: flask   # 自定义容器名称
    build: .                          # docker-compose.yml所在文件夹，去查找Dockerfile，构建镜像
    ports:
      - 5000:5000
    command: python classroom/serve.py    # docker启动后，运行flask程序
    depends_on:                                     # 需要依赖mongodb
      - mongodb
    environment:                                    # 通过环境变量为程序传入基础设置
      - mongo_uri=mongodb://root:123456@mongodb:27017
      #- mongo_uri=mongodb://localhost:27017
      - mongodb_name=ClassroomManager
    deploy:                                         # 设置该服务占用资源
      resources:
        limits:               # 设置容器的资源限制
          cpus: "0.5"         # 设置该容器最多只能使用 50% 的 CPU
          memory: 1G
  frontend: # 自建flask服务
    container_name: frontend   # 自定义容器名称
    build: /root/frontend
    ports:
      - 9090:80
```

图 4 yml 文件

3 实现效果

通过编写 Dockerfile 文件与 yml 文件，可实现在服务器上直接部署 Docker 容器来启动服务，图 5 展示了服务成功启动的界面。

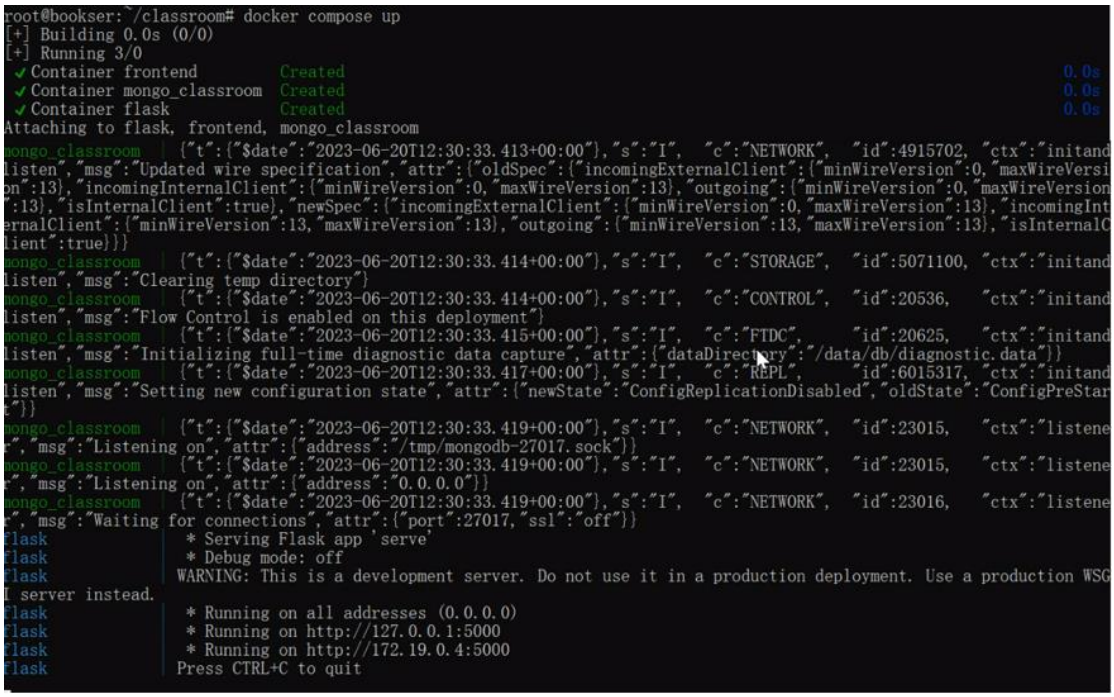

图 5 容器成功启动

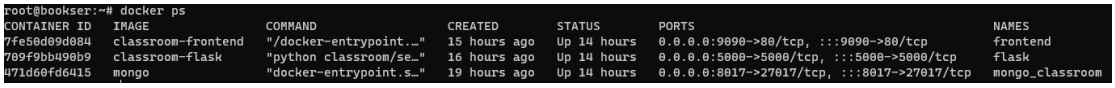

图 6 容器信息

4 遇到的问题及总结

由于之前软件开发经验较少，也没有使用过 docker，一切都是从零开始。在学习了 docker 的基本使用后就开始进行实践，采用的方法是在网上找到示例后进行阅读理解，再进行自己的操作。在实践过程遇到的问题包括但不限于：

1、 容器可以正常启动但是后端容器与数据库不能正常交互

2、 前后端容器启动后点击前端按钮没有反应

3、 前端容器启动后使用浏览器访问页面出现空白页面。

4、 在虚拟机中部署容器后功能正常，在部署到服务器上时访问不了前端页面的端口。

问题 1 的原因是数据库链接的地址有问题，问题 2 的原因是前端代码中与后端交互的地址有问题，问题 3 原因是需要修改 vue 中的路由模式。问题 4 是要设置安全组开放服务器的对应端口。还有一些问题如修改代码后没有保存对应文件直接生成镜像，导致问题一直卡在那里找不到原因。

总结下来以上问题的原因一是软件开发经验不足，很多编译器、工具的使用都不熟练导致效率较低，二是对于工具的工作原理没有完全理解清楚导致在配置一些信息时出错。所以要高效率的开发软件一是要多进行实际操作积累经验，二是对软件工作的底层原理要理解清楚，这才能达到事半功倍的效果。

此外在 pre 时根据老师的提醒发现真正的部署需要有一套固定的流程在保证项目工作

正常时在发布到服务器上，而自己在实践时只是自己测试通过后再发布，这样在大型的软件开发中如果忘记测试就发布可能导致项目功能不正常的问题，之后应该注意这个流程的规范，养成良好的开发习惯。

三、 前端管理端设计

1 Vue 介绍

Vue.js（通常简称为 Vue）是一款流行的开源 JavaScript 框架，用于构建用户界面。它专注于视图层，可以与其他库或现有项目进行集成，也可以作为单独的库使用。Vue.js 的目标是通过提供简单、灵活的方式来构建交互式的 Web 界面。Vue.js 是一款易学易用、灵活高效的前端框架，具有响应式数据绑定、组件化开发、虚拟 DOM 等特性，能够帮助开发人员构建现代化、交互式的 Web 应用程序。

2 管理端设计工作

前端代码框架如图 7 所示。主要为边侧栏与主页面两个文件。具体的代码细节不再赘述。

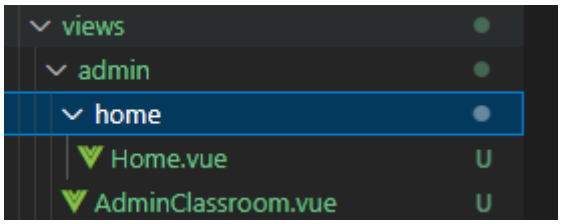

图 7 前端代码框架

3 前端实现效果

管理端的前端页面展示如图 8、图 9 所示。

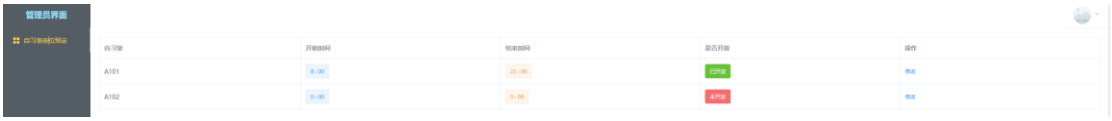

图 8 管理端前端展示

图 9 管理自习室开放时间

管理端页面的功能一是查询展示当前自习室的信息，包括自习室名称，开放时间，当前是否开放，以及修改开放时间的操作。

4 遇到的问题及总结

前端的学习及实现比较容易上手，在理解了 vue 的工作逻辑后根据需求编写代码即可。其中遇到过页面不能正常跳转的问题，查询 vue 路由机制后添加子路由后得以解决。

四、　本次作业的总结

在作业的实践过程中，预想比较简单的工作在实施起来遇到了各种各样的问题，一是自己开发经验不足，二是对工具的工作原理理解不清。导致工作量无意义的增大了很多，但也在这个过程中锻炼了心态，工程上总会遇到一个个问题，要耐心解决才能实现目标。

本次作业的目的之一也是希望通过规范的软件开发流程来熟悉开发流程，但这方面在本次作业中我们做的有所欠缺，只是为了实现功能而做，忽略了开发过程中的规范的合作流程，导致在后期合并代码时徒增了许多不必要的工作量，这也是这次作业的经验之一，合理运用规范的开发流程可能看起来会繁琐一点，但在大型项目中会减少很多不必要的麻烦。