|  |  |
| --- | --- |
| **学 号：** | 0122010870323 |

****

**软件工程实践（二）**

|  |  |
| --- | --- |
| **学 院** | **计算机与人工智能学院** |
| **专 业** | **软件工程** |
| **班 级** | **软件2002** |
| **姓 名** | **周盟** |
| **编 号** | **0122010870323** |
| **指导教师** | **唐祖锴** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2023 | 年 | 6 | 月 | 19 | 日 |

**目录**

[1 任务概述 1](#_Toc91495513)

[2 任务分析 1](#_Toc91495514)

[3 开发计划 1](#_Toc91495515)

[4 软件配置计划 1](#_Toc91495516)

[5 测试计划 1](#_Toc91495517)

[6 实施情况 1](#_Toc91495518)

[7 实施过程问题记录与分析 2](#_Toc91495519)

[8 任务总结 2](#_Toc91495520)

[9 参考文献 2](#_Toc91495521)

# 1 任务概述

## 任务目的

* 巩固强化软件编程规范
* 提高面向对象软件建模与抽象能力
* 培养小组协同开发能力
* 掌握基于Maven的软件项目管理机制
* 掌握基于Github的小组协同开发工具和平台
* 了解DevOps软件开发流程

## 任务内容

* 创建软件开发小组
* 针对样例代码工程进行小组讨论，确定功能扩充需求点
* 基于Github中的issue管理功能明确工作任务并为组员分配工作任务
* 基于小组商定的分支模型进行软件功能开发，并按开发流程进行代码测试、提交、归并和同步
* 代码提交到远程仓库后，应进行自动化代码格式规范检查和测试以确保功能符合需求设计
* 完成前述各项任务后，可尝试进行代码自动化打包，自动生成可供执行的jar文件

## 任务要求

1. 创建软件开发小组
   * 每个开发小组人数3-5人，推选一人作为组长，负责组织、协调和领导团队开发；
   * 所有小组成员应按操作步骤在github开发平台上加入同一小组，共用同一代码仓库；
2. 开展小组讨论，确定功能扩充点
   * 样例工程“world-of-zuul”具备最基本的程序功能，该项目具有极大的扩展空间，开发小组内可进行沟通讨论，确定系统结构优化需求或功能扩充需求，结构优化或功能扩充项不能少于5项；

可供参考的结构优化或功能扩充项包括但不限于以下内容：

1. 扩展游戏，使得一个房间里可以存放任意数量的物件，每个物件可以有一个描述和一个重量值，玩家进入一个房间后，可以通过“look”命令查看当前房间的信息以及房间内的所有物品信息；
2. 在游戏中实现一个“back”命令，玩家输入该命令后会把玩家带回上一个房间；
3. 在游戏中实现一个更高级的“back”命令，重复使用它就可以逐层回退几个房间，直到把玩家带回到游戏的起点；
4. 在游戏中增加具有传输功能的房间，每当玩家进入这个房间，就会被随机地传输到另一个房间；
5. 在游戏中新建一个独立的Player类用来表示玩家，并实现下列功能需求：
   * 一个玩家对象应该保存玩家的姓名等基本信息，也应该保存玩家当前所在的房间；
   * 玩家可以随身携带任意数量的物件，但随身物品的总重量不能操过某个上限值；
   * 在游戏中增加两个新的命令“take”和“drop”，使得玩家可以拾取房间内的指定物品或丢弃身上携带的某件或全部物品，当拾取新的物件时超过了玩家可携带的重量上限，系统应给出提示；
   * 在游戏中增加一个新的命令“items”, 可以打印出当前房间内所有的物件及总重量，以及玩家随身携带的所有物件及总重量；
   * 在某个或某些房间中随机增加一个magic cookie（魔法饼干）物件，并增加一个“eat cookie”命令，如果玩家找到并吃掉魔法饼干，就可以增长玩家的负重能力；
6. 扩充游戏基本架构，使其支持网络多人游戏模式，具备玩家登陆等功能；
7. 为单机或网络版游戏增加图形化用户界面，用过可以通过图形化界面执行游戏功能；
8. 可以为游戏增加数据库功能，用于保存游戏状态和用户设置；
9. 基于Github中的issue管理功能明确工作任务并为组员分配工作任务
   * 将工作任务拆分细化后，明确版本开发计划和里程碑时间节点；
   * 在github平台创建任务issue并为所有组员分配任务；
   * 每位组员可以分别承担不同的开发任务，也可以按照小组角色分别承担开发、测试、集成等工作任务；
   * 工作任务的划分是最终衡量小组成员工作量的重要依据；
10. 基于小组商定的分支模型进行软件功能开发，并按开发流程进行代码测试、提交、归并和同步
    * 小组成员按照小组商定的分支模型在各自的工作分支进行进行开发任务；
    * 工作分支在合并前应同步到远程仓库供教师检查每人的开发工作完成情况；
    * 提交代码时应按照小组约定的规范格式填写代码提交说明，代码提交说明也将作为评分的重要依据；
11. 代码提交到远程仓库后，应进行自动化代码格式规范检查和测试以确保功能符合需求设计；
    * 可以利用github平台的actions功能在代码提交时自动触发代码格式检查，对于不符合规范的代码系统将给出提交失败提示；
    * 可以利用github平台的actions功能在代码提交时自动触发测试用例检查，对于不能通过测试检查的代码系统将给出提交失败提示；
12. 可尝试进行代码自动化打包，自动生成可供执行的jar文件
    * 结合github平台的actions功能和maven编译脚本，在代码通过规范性检查和测试用例后，进行自动化打包，生成可供直接执行的jar文件用于系统发布

# 2 任务分析

本次系统开发的主要内容是一个冒险游戏。该游戏涉及到前端界面的设计和开发，包括登录、注册、忘记密码等功能的实现；后端的鉴权认证，包括用户身份验证和授权；数据库的设计，用于存储用户信息和游戏数据；前后端数据交互的格式设计，确保数据在前后端之间的有效传递；以及后端核心游戏逻辑的编写，包括游戏场景的展示、角色的操作与交互等。整个系统开发涵盖了前后端技术栈的应用和协同工作，旨在打造一个完整的冒险游戏平台。

本次实践任务的工作重点可以总结为以下几个方面：

1. 架构设计：重点是设计系统的整体架构，包括前端和后端的架构设计。这包括确定采用的技术栈、组织代码结构、定义模块和组件之间的关系等，以确保系统具有良好的可扩展性、可维护性和可重用性。

2. 任务分配与协调：重点是将任务拆分成可执行的子任务，并为每个成员分配相应的工作任务。协调团队成员的工作进度，确保任务按计划进行，并及时解决可能出现的问题和冲突。

3. 前端界面开发：重点是实现游戏的前端界面，包括登录、注册、忘记密码等功能的界面编写。关注用户体验和界面交互设计，确保界面美观、易用和符合用户期望。

4. 后端开发：重点是实现后端的鉴权认证功能和核心游戏逻辑。包括用户身份验证、授权管理、数据库设计和操作，以及游戏逻辑的编写。关注系统的安全性、性能和可靠性。

5. 数据交互格式设计：重点是定义前后端之间的数据交互格式，确保数据的准确传递和解析。设计合适的数据结构和协议，使得前后端能够高效地进行数据交换。

6. 测试与质量保证：重点是进行全面的测试，包括单元测试、集成测试和系统测试，以确保系统的功能符合需求并具备稳定性。关注代码质量和性能优化，确保系统的高质量交付。

本次任务采用了以下技术方案：

1. 前端技术方案：前端采用Vue框架和Element-UI组件库进行开发。Vue是一种流行的JavaScript框架，具有轻量级、灵活和高性能的特点，能够实现组件化开发和响应式数据绑定。Element-UI是基于Vue的UI组件库，提供了丰富的界面组件和样式，使得前端界面开发更加高效和便捷。

2. 后端技术方案：后端采用Spring Boot框架和MyBatis Plus持久化框架进行开发。Spring Boot是一种快速构建Java应用程序的框架，提供了自动化配置和约定优于配置的特性，简化了开发流程。MyBatis Plus是MyBatis框架的增强版，提供了更加便捷的数据库操作方式和强大的查询能力，提高了后端数据持久化的效率和开发效率。

3. 数据库技术方案：数据库采用关系型数据库MySQL进行存储。MySQL是一种常用的开源数据库管理系统，具有稳定性高、性能好和成本低的优点，适用于中小型应用程序的数据存储需求。

4. 接口测试工具：接口测试使用Postman进行。Postman是一种常用的API测试工具，能够模拟HTTP请求并验证接口的响应结果，方便开发人员进行接口调试和测试。

5. 版本控制工具：版本控制使用Git进行。Git是一种分布式版本控制系统，能够管理代码的版本和变更，方便团队协作和代码的追踪。

# 3 开发计划

项目开发计划如下：

1. 需求分析和规划阶段（0.5天）：

- 确定项目的整体需求和目标

- 分析用户需求，确定功能和特性

- 制定项目计划和时间表

2. 技术准备阶段（0.5天）：

- 搭建开发环境，包括配置开发工具和环境

- 确定使用的技术栈和框架

- 准备所需的资源和文档

3. 前端开发阶段（1天）：

- 实现登录界面的设计和开发

- 实现注册界面的设计和开发

- 实现忘记密码界面的设计和开发

- 进行表单字段校验的实现

4. 后端开发阶段（1天）：

- 实现用户登录功能的后端逻辑和接口

- 实现用户注册功能的后端逻辑和接口

- 实现忘记密码功能的后端逻辑和接口

- 实现游戏核心逻辑的后端逻辑和接口

5. 前后端联调和测试阶段（0.5天）：

- 进行前后端接口的联调测试

- 测试前端界面的功能和交互

- 进行全面的系统测试，包括用户场景测试和异常情况测试

6. 优化和修复阶段（0.2天）：

- 根据测试结果进行优化和修复

- 解决发现的bug和问题

- 进行性能优化和代码优化

7. 文档编写和整理阶段（0.2天）：

- 撰写架构设计文档

- 编写接口文档和使用说明文档

- 整理代码注释和文档

- 准备最终的项目交付物和报告

以上计划是一个初步的时间安排，具体的开发进度可能会根据实际情况进行微调和调整。在每个阶段的结束，团队成员将进行评审和复盘，确保项目按计划顺利进行，并及时发现和解决问题。

在本次项目开发中，我们计划采用以下主要的软件工程工具：

1. 版本控制工具：Git

- 使用Git进行团队协作和代码版本管理

- 创建Git仓库用于存储项目代码

- 使用分支管理进行并行开发和版本控制

- 提交代码、合并分支和解决冲突

2. 文档编写工具：Markdown

- 使用Markdown编写项目文档和报告

- 使用Markdown语法进行文本排版和格式化

- 在文档中插入代码块、表格、图像等元素

3. 编码规范工具：ESLint

- 使用ESLint进行JavaScript代码的静态检查，确保代码质量和一致性

- 配置代码规范和风格约定，统一团队的编码风格

4. 单元测试工具：JUnit

- 使用Jest进行前端代码的单元测试

- 使用JUnit进行后端代码的单元测试

- 编写测试用例，验证代码逻辑的正确性和稳定性

5. 接口测试工具：Postman

- 使用Postman进行后端接口的测试和调试

- 发送HTTP请求，验证接口的响应和数据格式

- 检查接口性能和异常处理

# 4 软件配置计划

在本项目中，我们遵循以下命名规范、分支管理规范和提交规范：

1. 命名规范：

- 分支命名规范：使用小写字母和下划线，描述分支的用途或所涉及的功能模块。例如：`dev`、`backend`、`front\_login`等。

- 提交消息命名规范：简要描述本次提交完成的内容，遵循清晰、准确和具体的命名规范。

2. 分支管理规范：

- `master` 分支：作为主分支，存储稳定和可发布的代码版本。仅用于合并来自 `dev` 分支的成熟功能和修复的bug。

- `dev` 分支：用于集成开发人员的工作成果和进行整体功能测试。

- 功能分支：每个人根据自己领取的 issue 在个人分支上进行开发，分支命名与领取的 issue 相关。

- 分支合并：开发人员在完成阶段性开发任务后，将个人分支提交合并请求（Pull Request）到 `dev` 分支进行代码审查和集成测试。

3. 提交规范：

- 提交消息应准确描述本次提交所完成的内容，遵循简明、清晰和具体的规范。

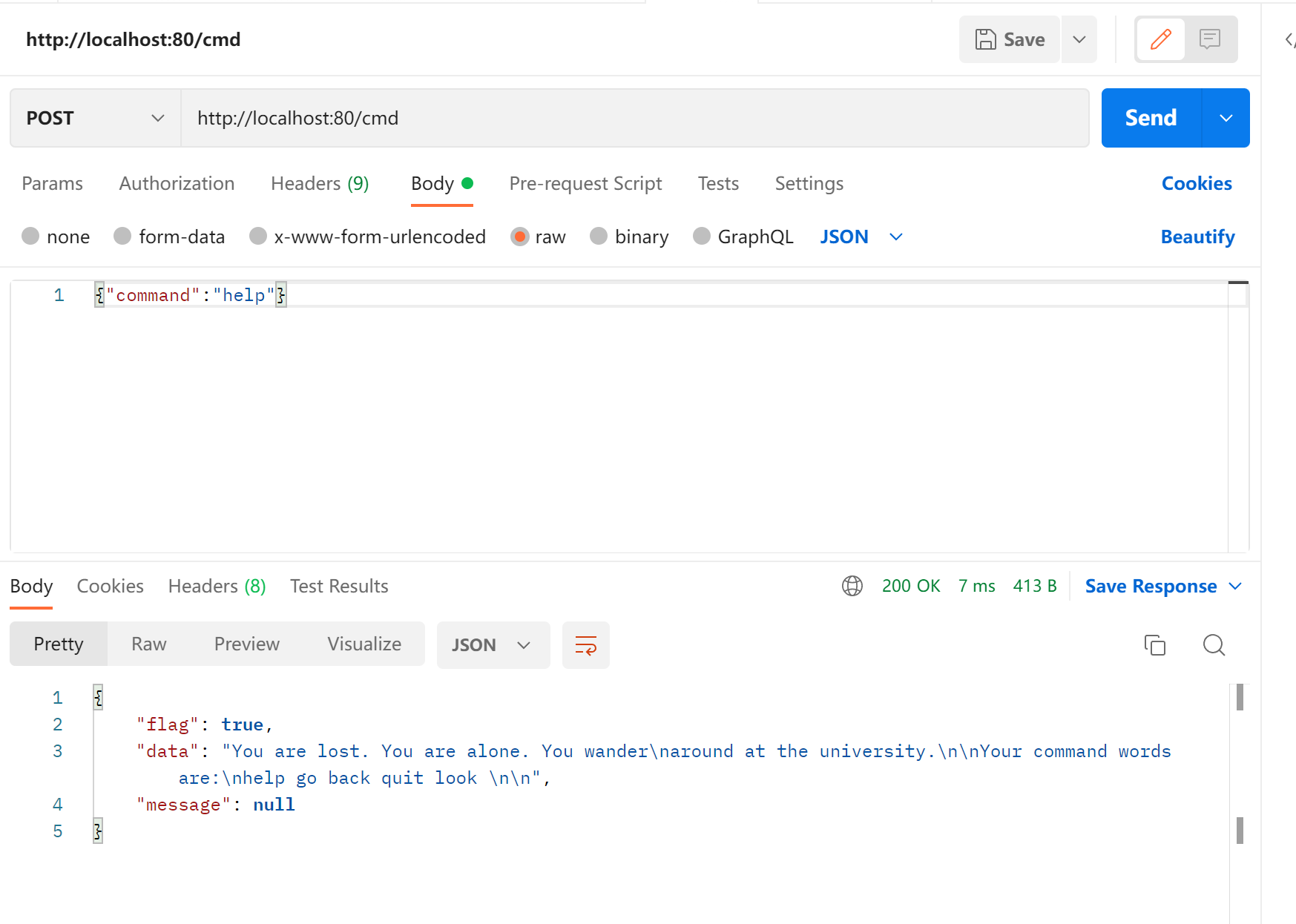
- 提交频率应根据开发进度和任务的复杂性进行合理的划分，以确保提交的代码是稳定和可测试的。

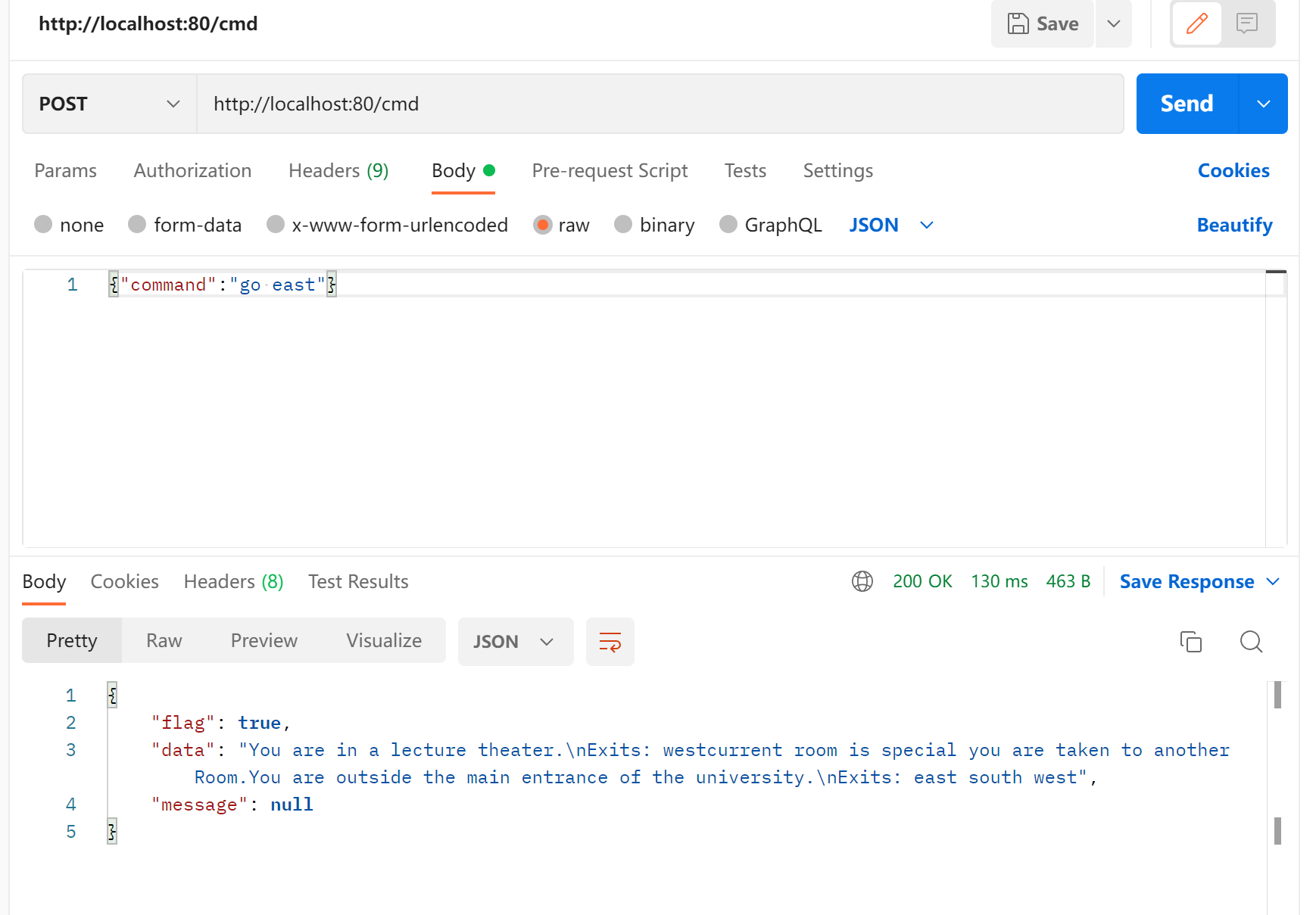
# 5 测试计划

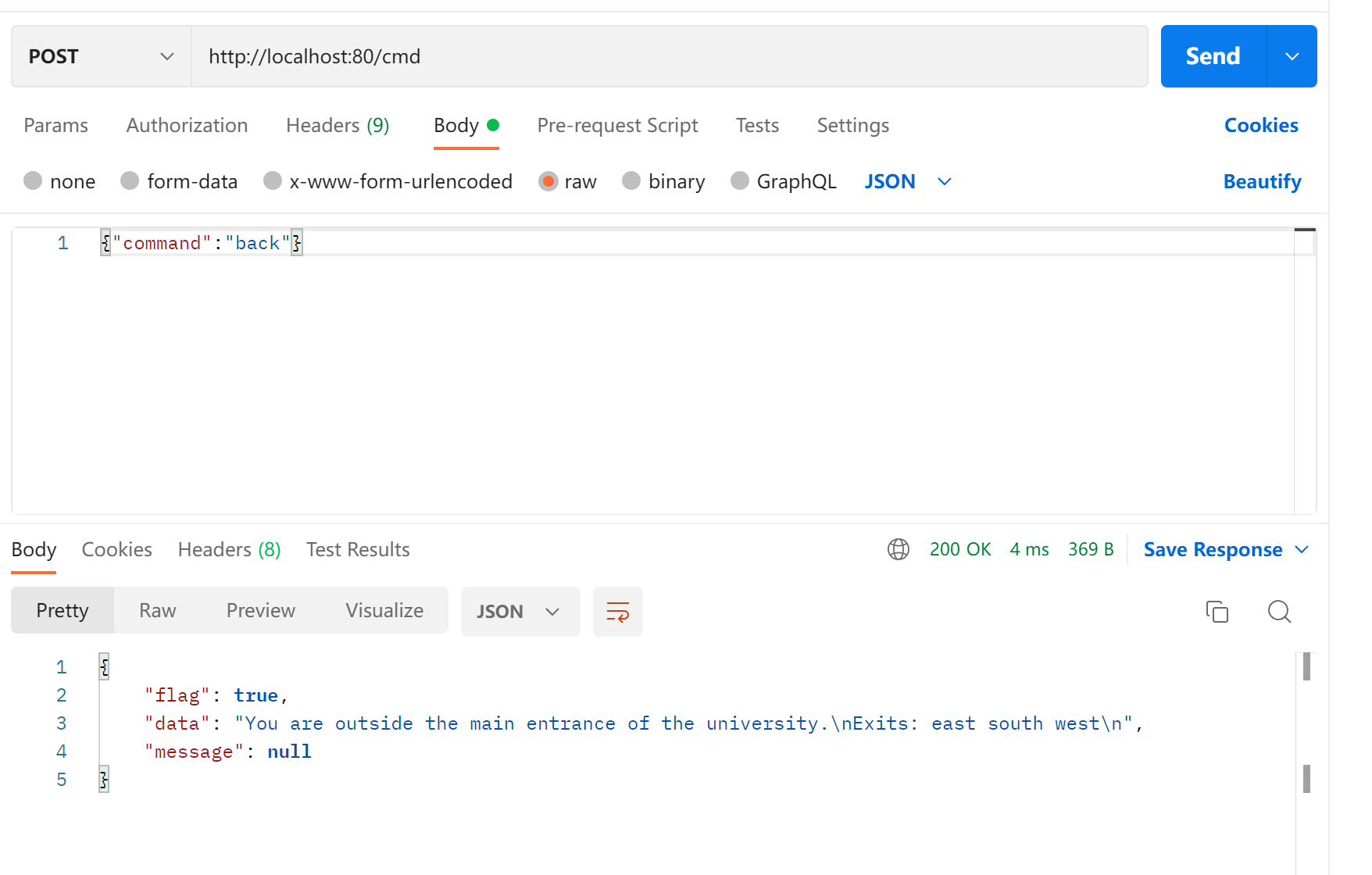
前端测试：测试响应式布局是否正常，测试页面跳转是否正确，测试数据格式校验是否正常

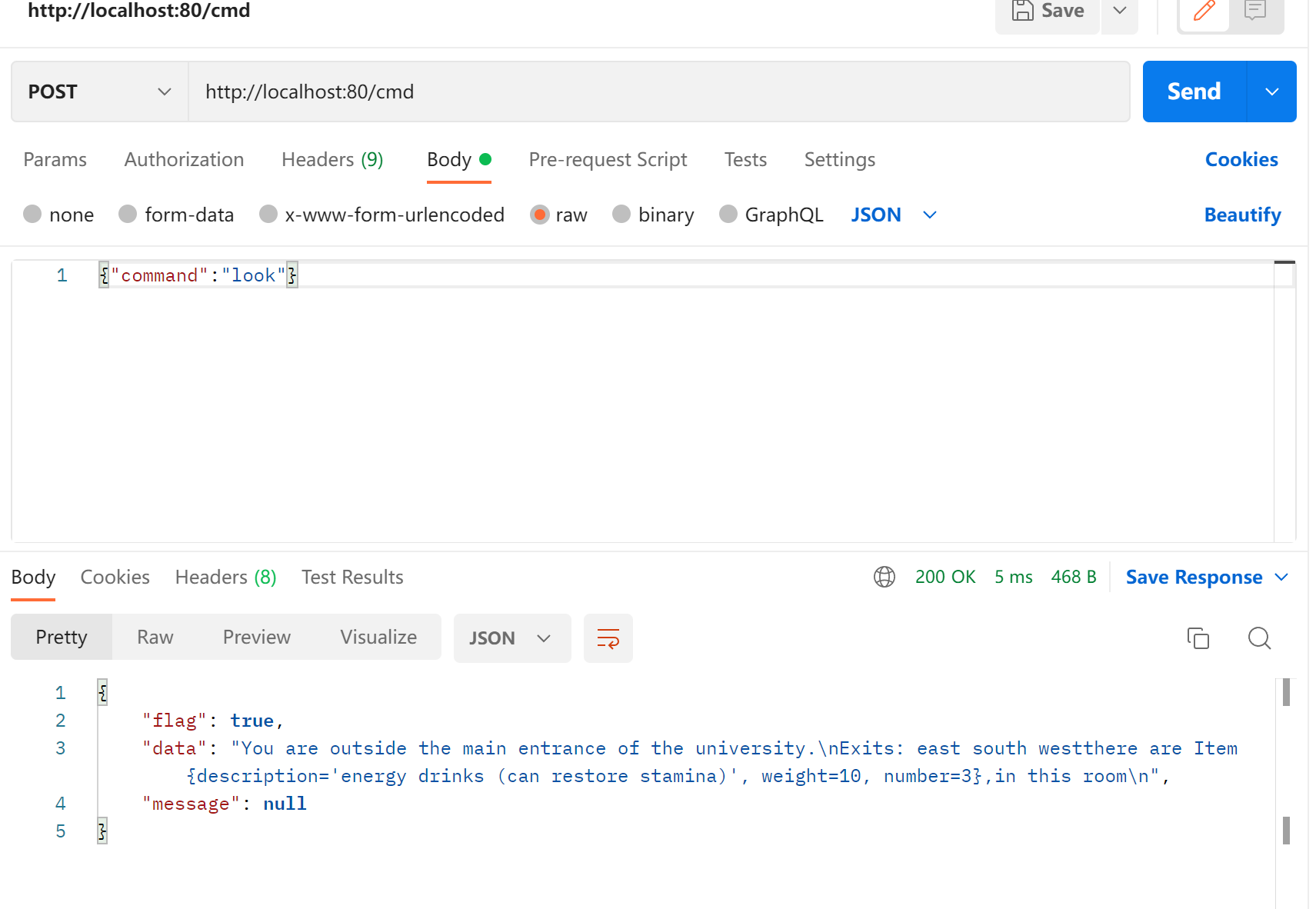
后端测试：测试接口返回数据是否正常

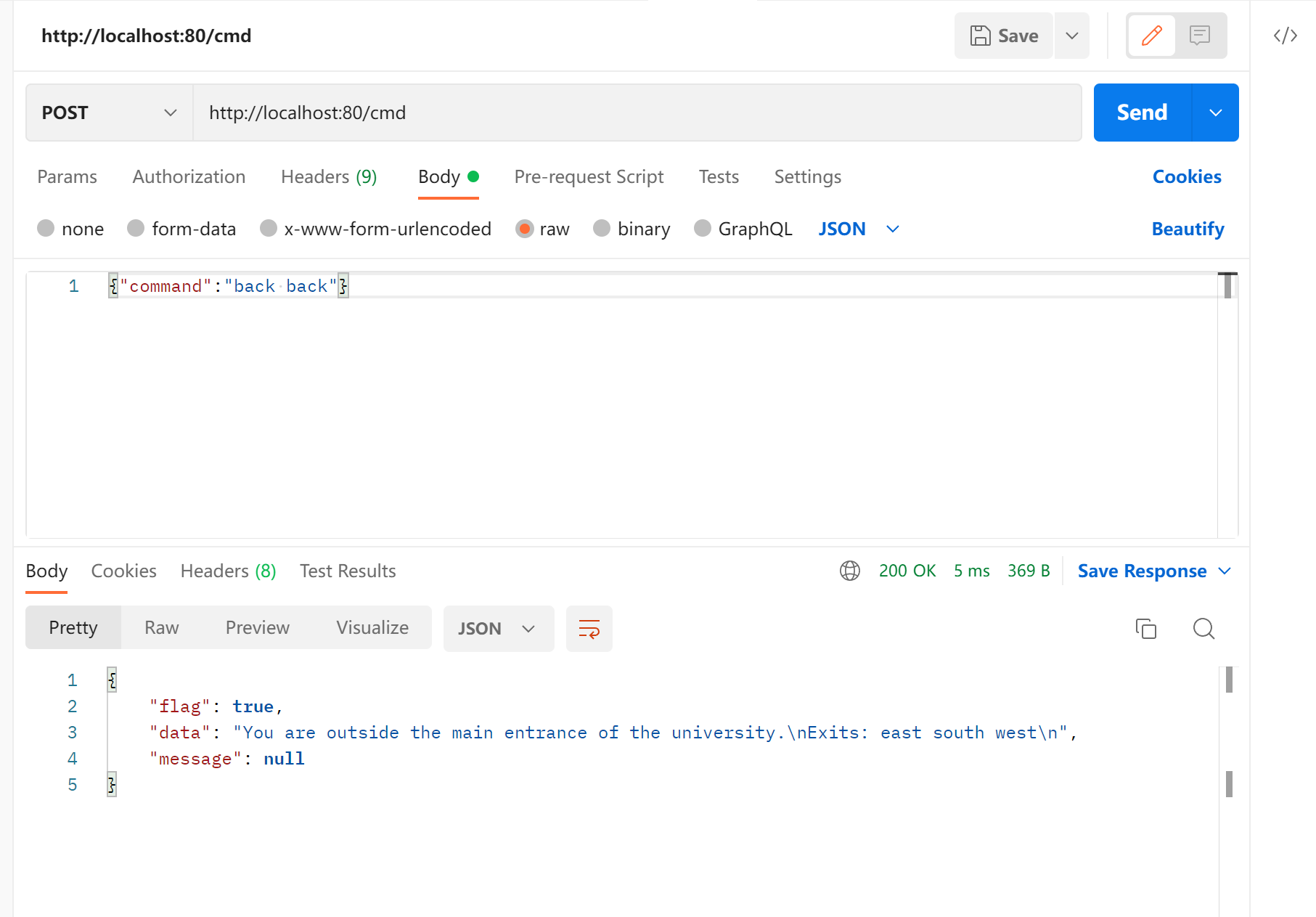
后端游戏逻辑测试截图：











# 6 实施情况

开发过程：

1、架构设计，输出架构设计文档、

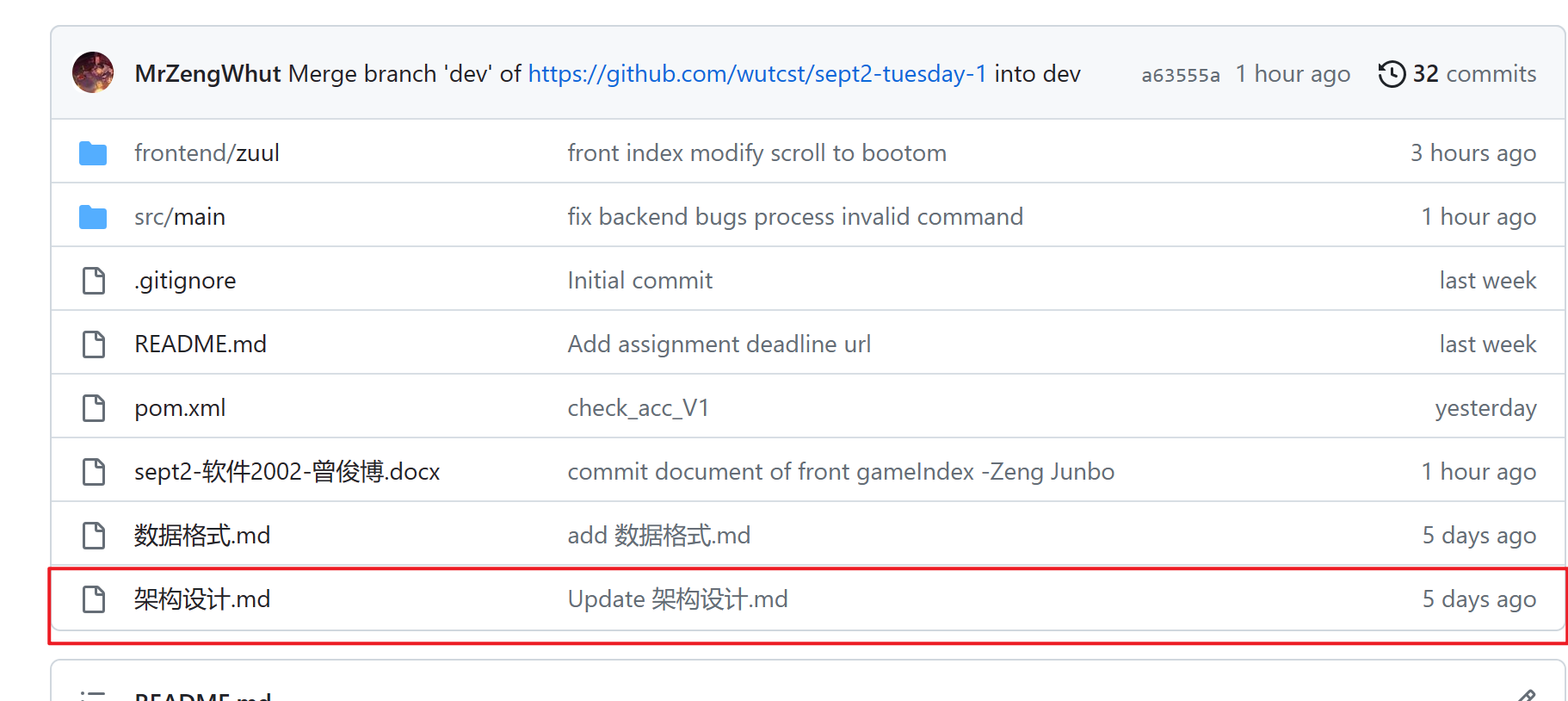
2、编写issue分发任务

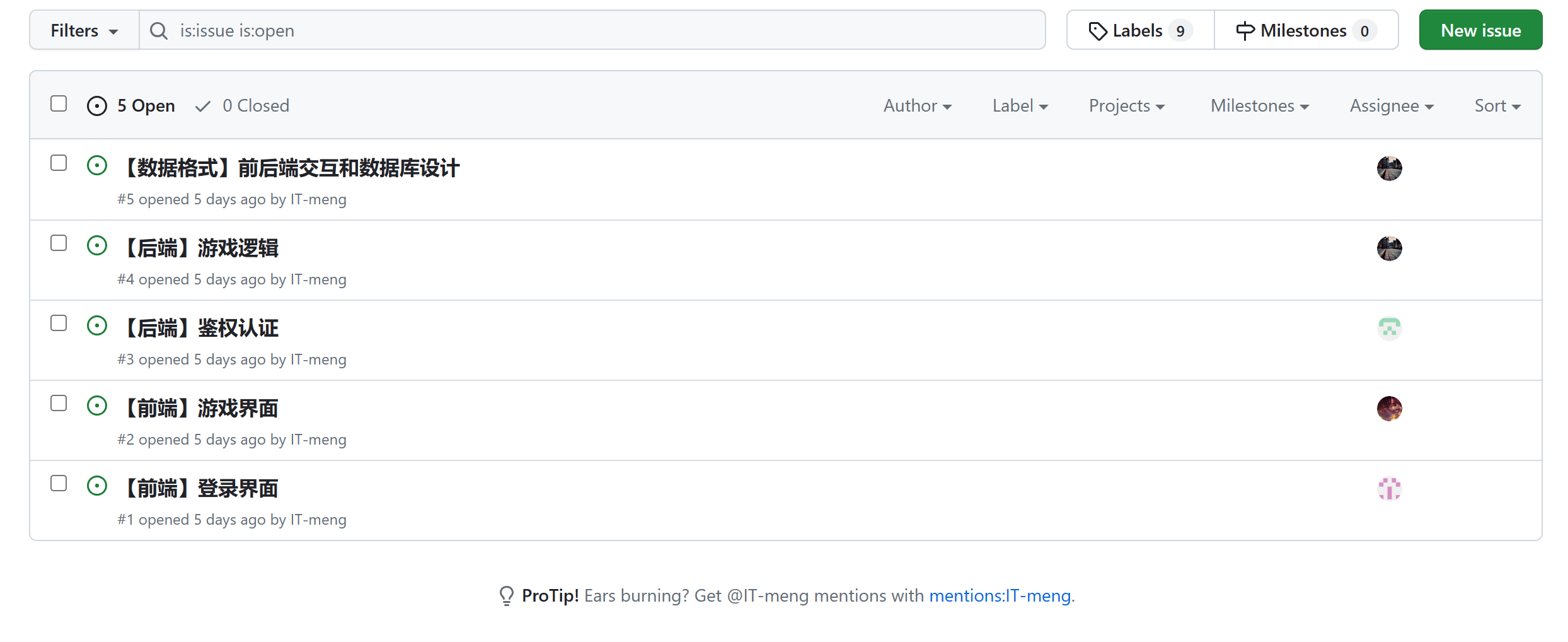
3、领取issue完成相关任务

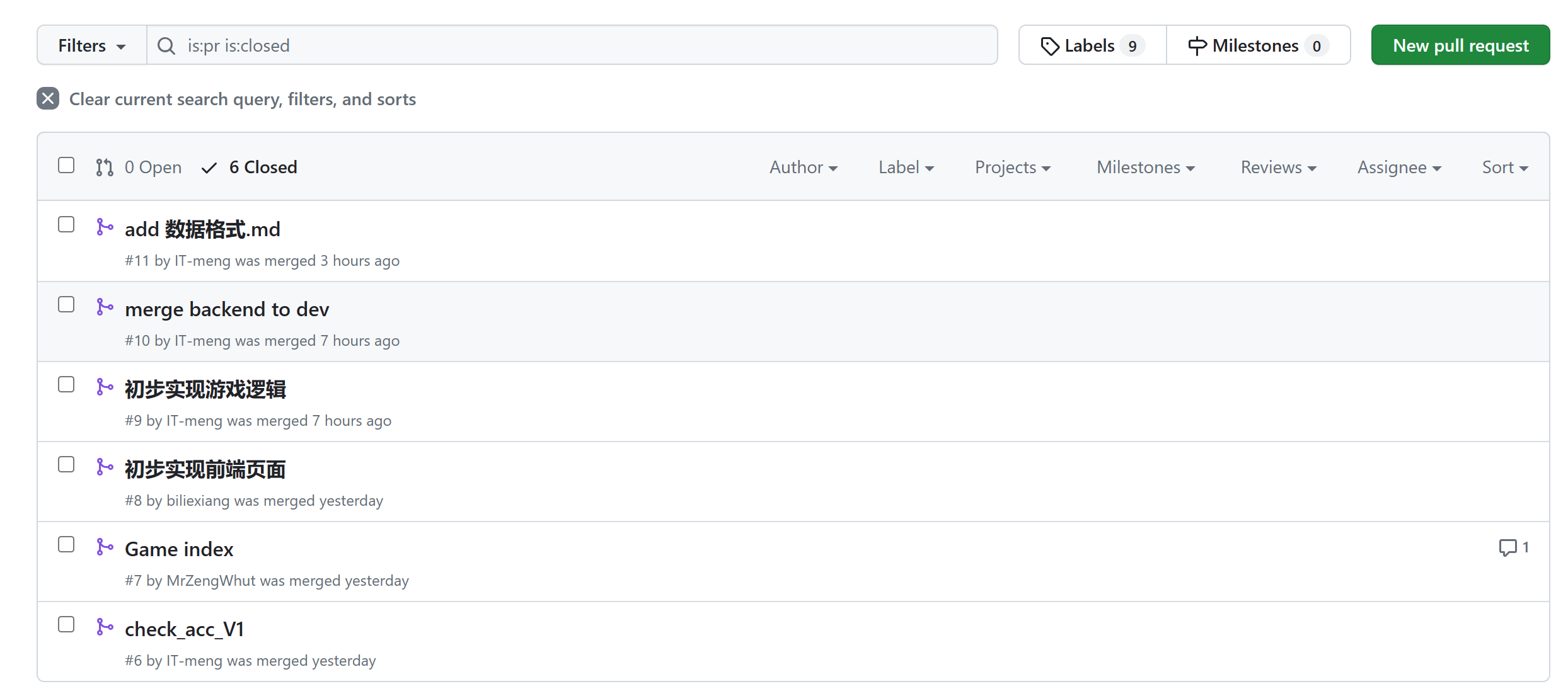
4、对自己的代码进行测试并提交pr

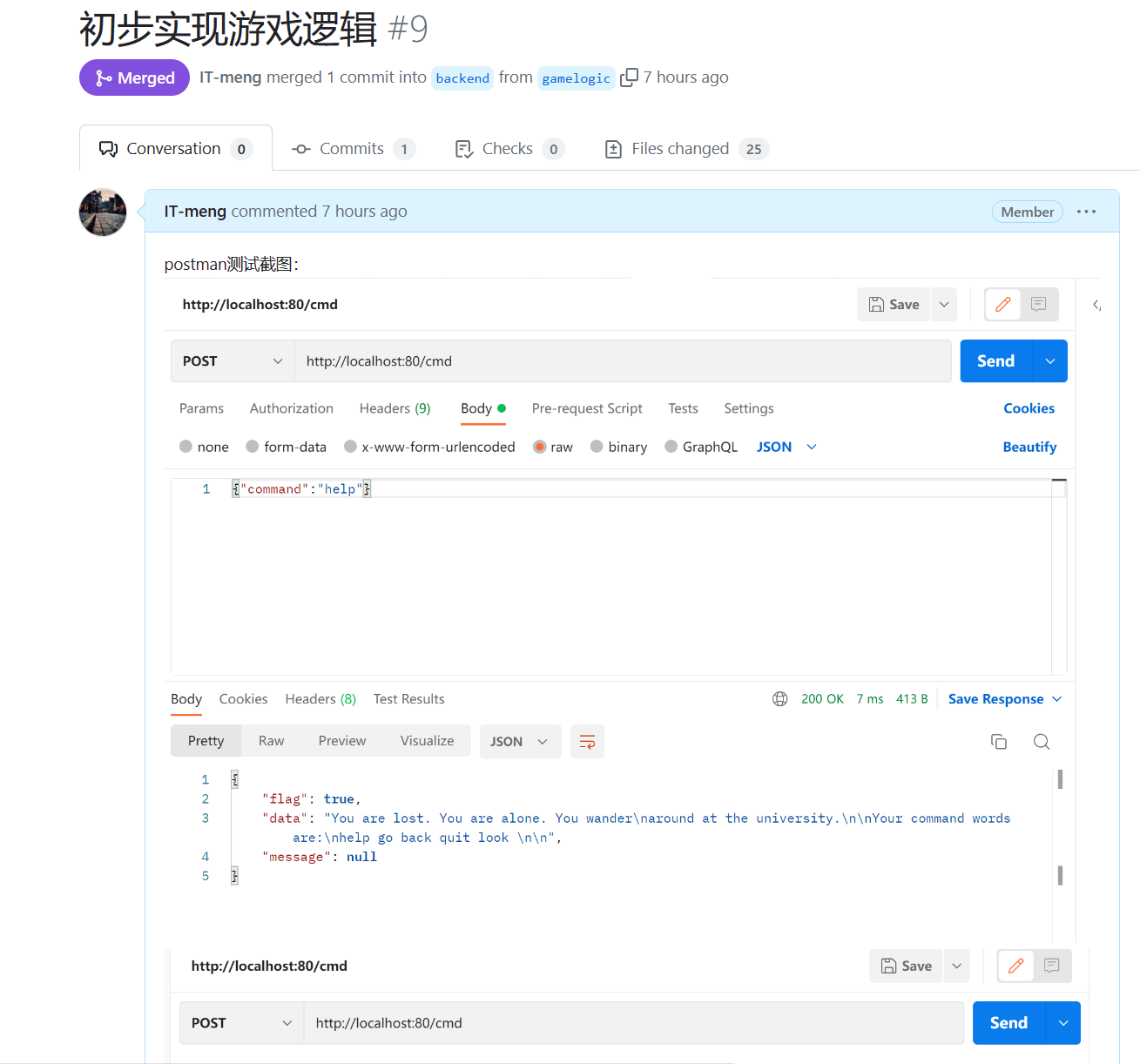
5、对合并后的代码进行整体测试

6、测试通过合并到master分支



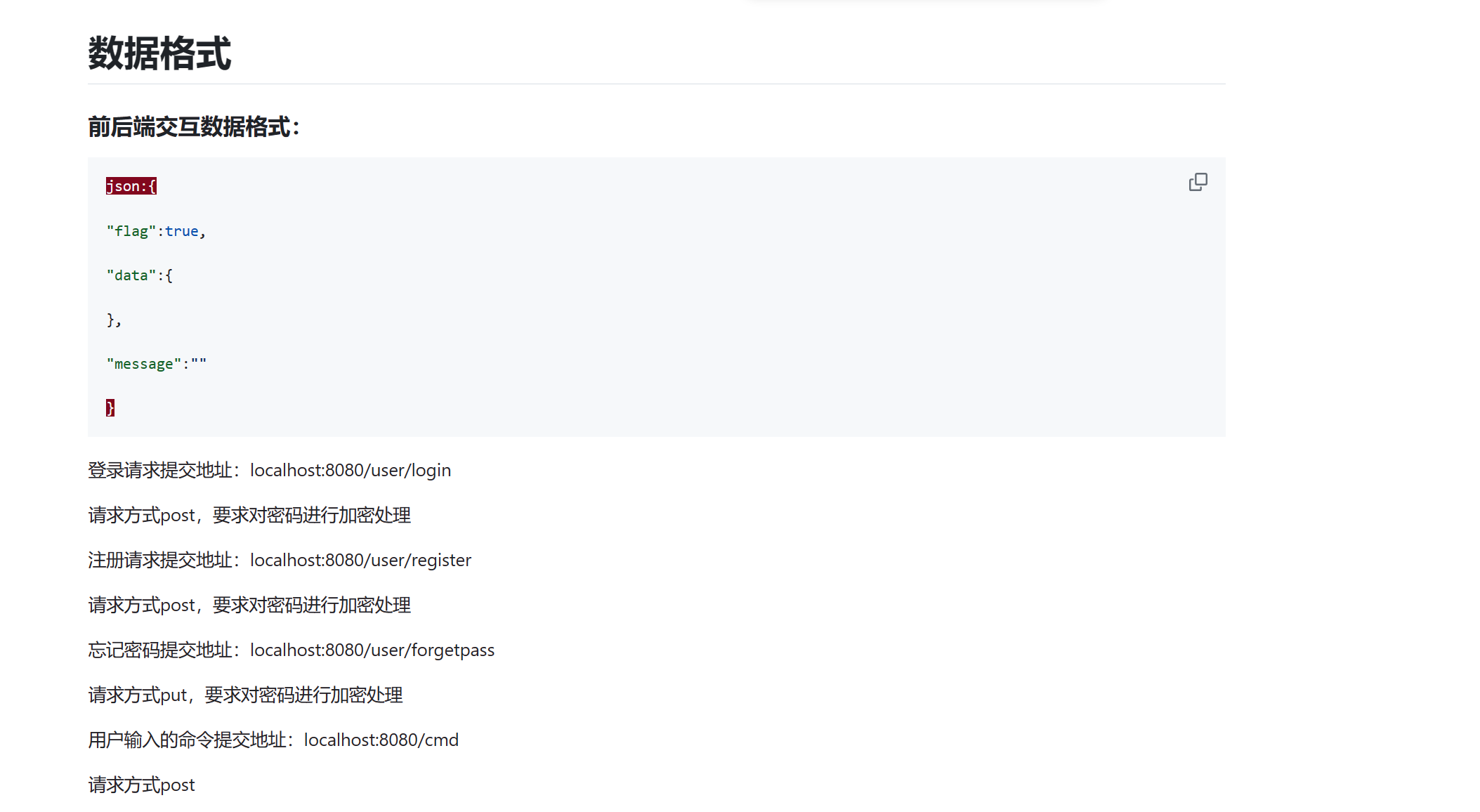






# 7 实施过程问题记录与分析

接口对接：在前后端的接口对接过程中，我们遇到了一些参数传递和数据格式的不一致问题。这导致前端无法正确解析后端返回的数据或后端无法获取前端传递的参数。为了解决这个问题，我们进行了详细的接口文档编写，并与前端团队成员进行密切的沟通和协作，以确保接口的准确性和一致性。



后端配置：



游戏核心逻辑接口：



# 8 任务总结

本次小组协同开发任务是一个具有挑战性的实践项目，我们成功地完成了系统开发并取得了一系列成果。通过这次任务，我们深入理解了团队合作的重要性，掌握了软件工程的实际应用，并提升了自身的技术能力和团队协作能力。

在任务概述中，我们明确了本次系统开发的主要内容，即开发一个冒险游戏系统，涉及前后端开发、数据库设计、数据交互等多个方面。

任务分析阶段，我们重点关注了游戏的前后端架构设计、技术方案选择和工程实施方案。通过团队的讨论和研究，我们制定了合理的开发计划，并采用了一系列软件工程工具和管理策略来支持项目开发。

在实施过程中，我们严格按照项目计划进行工作，各成员负责领取和完成相应的任务。通过合理的分支管理和提交规范，我们能够有效地协同开发，减少代码冲突，并及时跟踪和解决问题。

在测试阶段，我们制定了详细的测试计划，使用了适当的测试工具和规范，确保系统的功能和性能达到预期要求。同时，我们也进行了持续的代码提交和版本管理，保证了代码的可追溯性和可维护性。

在实施过程中，我们遇到了一些具体问题，例如数据库设计、接口对接、并行开发冲突和性能优化等。通过记录和分析这些问题，我们能够及时调整方案和解决困难，为项目的顺利实施提供了宝贵的经验教训。

总的来说，这次小组协同开发任务是一次富有挑战和收获的实践经验。我们充分发挥了团队的协作优势，解决了各种技术和管理问题，并最终成功地完成了项目的开发和交付。通过这次任务，我们不仅提升了自己的技术能力，还培养了团队协作和沟通的重要性，为今后的工作打下了坚实的基础。

# 9 参考文献

[1]秦不凡.大数据时代下软件工程技术的应用[J].农家参谋,2020(10):214.

[2]王永锋.计算机应用软件开发中编程语言的选择研究[J].计算机产品与流通,2020(05):24+38.

[3]张粦雁.企业软件项目建设管理应用系统实施经验浅谈[J].厦门科技,2020(02):43-46.

[4]苏红.软件工程方法在计算机软件开发中的应用解析[J].南方农机,2020,51(07):224.

[5]沈伟.信息系统集成在软件开发中的应用研究[J].农家参谋,2020(07):144.

[6]赵俊,石春.敏捷思想在软件开发中的应用与实践研究[J].电脑知识与技术,2020,16(09):98-99.

**《软件工程实践（一）》成绩评定表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓 名** | | **周盟** | **学 号** | **0122010870323** | |
| **专业、班级** | | **软件2002** | | | |
| **成绩评定：** | | | | | |
| 评价内容 | | | 满分 | 实得分 | |
| 得分 | 小计 |
| 实践任务  完成情况 | 软件项目设计、改进与扩充 | | 20 |  |  |
| 个人软件过程与项目管理 | | 15 |  |
| 代码版本管理 | | 25 |  |
| 代码注释与编码规范 | | 25 |  |
| 单元测试 | | 15 |  |
| 实践报告  总评情况 | 学习态度与考勤 | | 10 |  |  |
| 报告格式的规范性 | | 10 |  |
| 报告的逻辑结构与语言表达 | | 15 |  |
| 实践内容的正确性与合理性 | | 60 |  |
| 文献引用及标注 | | 5 |  |
| 总分 | | | 100 |  |  |
| **最终评定成绩（以优、良、中、及格、不及格评定）** | | | | | |

**指导教师签字：**

**年 月 日**