

2023秋DevOps课程第3次实践作业报告

第9组 LoveDevOps

角色	姓名	学号	QQ号	职责
组长	赵政杰	211250109	3239168744	代码检查，报告书写
组员2	谢其卓	211870187	1055069518	部署webhook，报告书写
组员3	陈盛	211250104	749810747	编写代码访问大模型
组员4	谷雨阳	211250108	1198890085	编写代码与大模型讨论

Code Review

使用了助教部署的大模型

承载平台

webhook部署在南软云

```
root@devops-student-run1 x + v
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\Misaka\Desktop\DevOps> ssh root@172.29.4.70
PS C:\Users\Misaka\Desktop\DevOps> ssh root@172.29.4.70
PS C:\Users\Misaka\Desktop\DevOps> ssh root@172.29.4.70
root@172.29.4.70's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

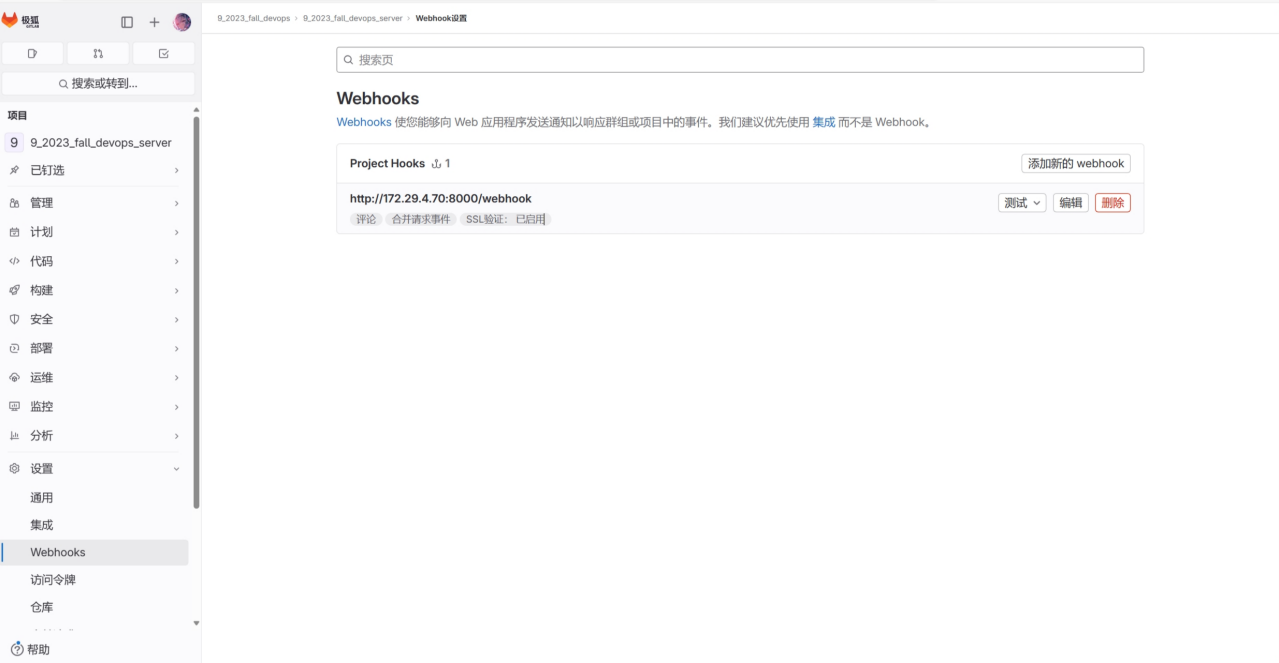
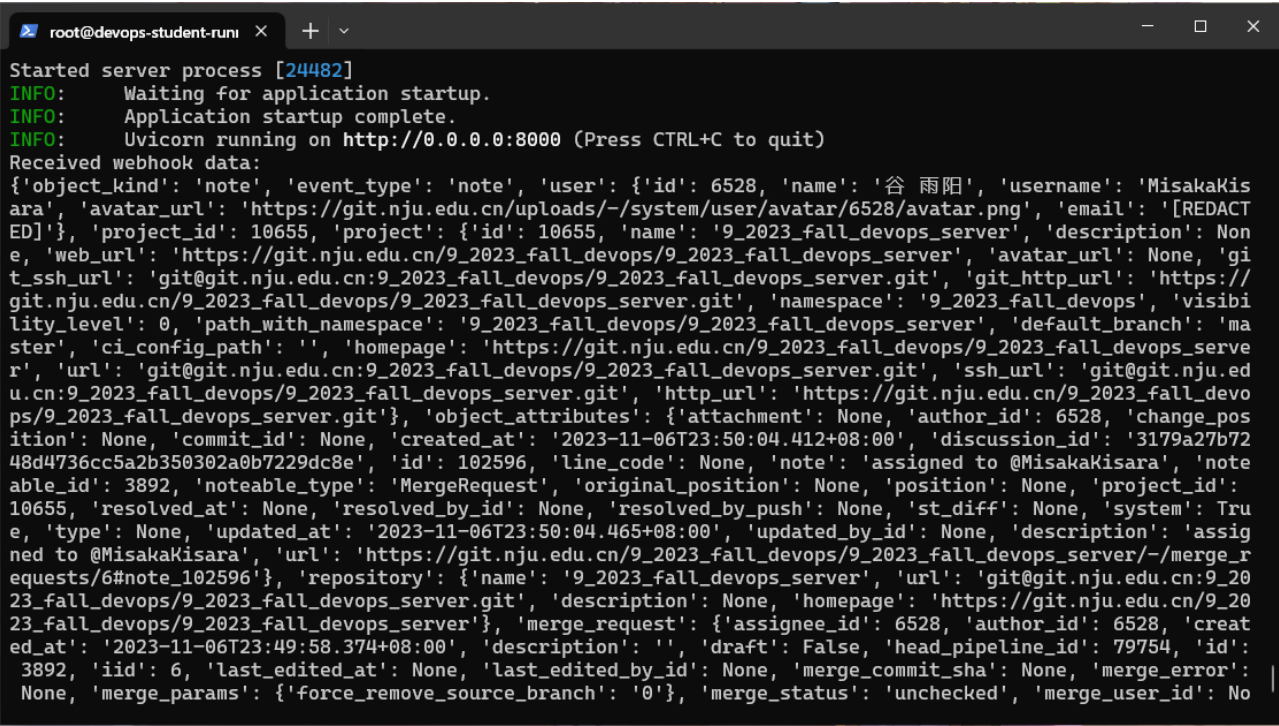
544 packages can be updated.
434 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Wed Nov  8 13:02:09 2023 from 172.31.98.180
root@devops-student-runner-9:~# ls
builds Desktop Documents Downloads Music Pictures Public snap Templates Videos
root@devops-student-runner-9:~# cd builds/
```

通用部分

实现Merge Request Hook 8/8分

截图与链接



关键代码展示

```
from fastapi import FastAPI, Request

app = FastAPI()

# 发起merge和comment时的webhook
@app.post("/webhook")
async def receive_webhook(request: Request):
    data = await request.json()

    print("Received webhook data:")
    print(data)

    if data["object_kind"] == 'merge_request':
        # 实现 merge_request 相关操作的代码
        pass

    if data["object_kind"] == 'note':
        # 实现 note 相关操作的代码
        pass

    return {"status": "ok"}
```

Code diff对应函数提取与PR规模限制 代码检查 16/16分

截图与链接

```
root@devops-student-run: ~
}

+   System.err.println("code diff test");
+
+   userDao.save(UserEntity.builder().username(username).password(BCrypt.hashpw(password))
+       .name(name).idn(idn).phone(phone).type(type).mileagePoints(0L).privilege(0L).build());
+   }

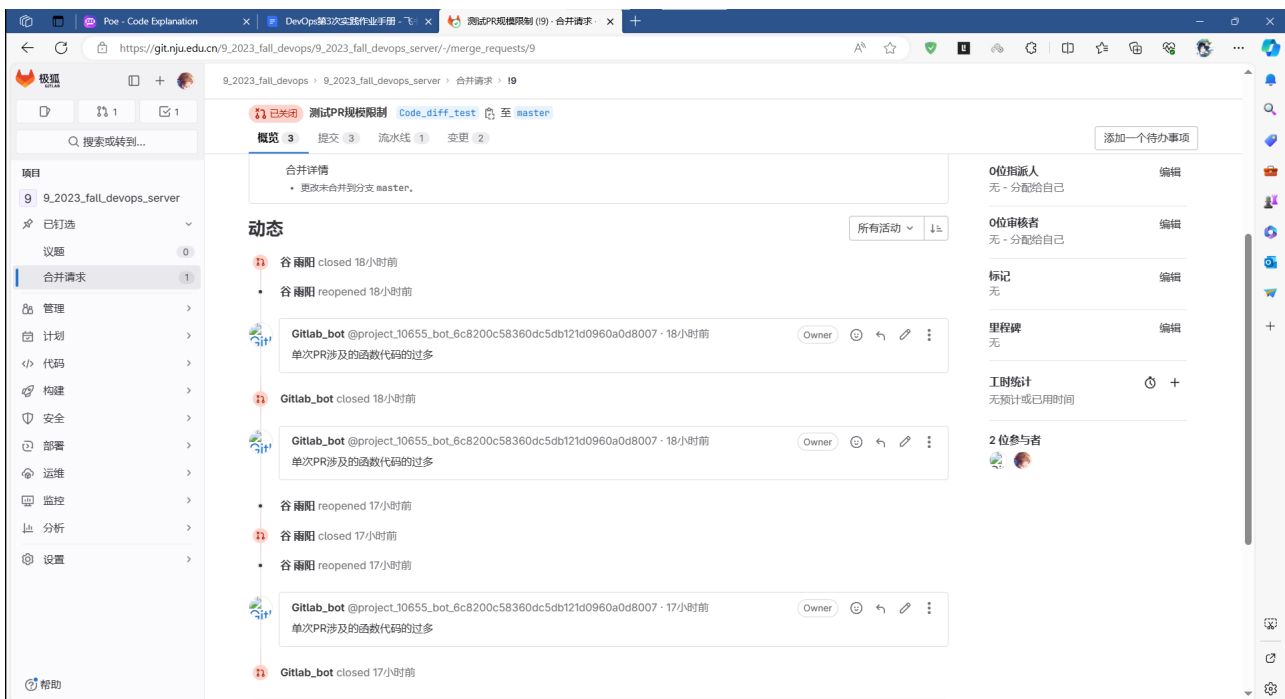
@@ -45,6 +47,9 @@ public class UserServiceImpl implements UserService {
+   /*
+   @Override
+   public UserEntity findByUserName(String username) {
+   +   System.err.println("code diff test");
+   +
+   +   return userDao.findByUsername(username);
+   }

origin_file_path: /tmp/tmpidzkinax.java
提取的函数在这里@Override
public UserEntity findByUserName(String username) {

    System.err.println("code diff test");

    return userDao.findByUsername(username);

}
除报错外所有函数提取成功!
INFO:      210.28.130.14:33374 - "POST /webhook HTTP/1.1" 200 OK
```



Gitlab_bot关闭的MR: https://git.nju.edu.cn/9_2023_fall_devops/9_2023_fall_devops_server/-/merge_requests/9

关键代码展示

```
if data["object_kind"] == 'merge_request':
    # 实现 merge_request 相关操作的代码
    project_id = data["project"]["id"]
    merge_request_iid = data["object_attributes"]["iid"]
    print("project_id: " + str(project_id))
```

```

print("merge_request_iid: " + str(merge_request_iid))
diffs = api_merge_diff(project_id, merge_request_iid)
length = 0
for diff in diffs:
    file_path = diff["old_path"]
    # 过滤非 java 文件的 diff
    if get_suffix(file_path) != '.java':
        continue
    file_diff = diff["diff"]
    # single_diffs = get_single_diff(file_diff) # 分离file-
diff至多个single-diff
    version = api_get_source_branch(project_id,
merge_request_iid)
    # temp_file_path = get_gitlab_file_content(project_id,
file_path, version) # 获取发生改变的文件的全部内容并生成临时文件保存服务器
    # get_new_func调用了get_single_diff和get_gitlab_file_content
    funcs = get_new_func(project_id, version, file_diff,
file_path) # 获取新增的函数
    for func in funcs:
        length += len(func[0])
    print(f'length: {length}')
    # 单次PR涉及的函数代码的字节数不应超过8192
    if length > 8192:
        comment_merge_request(project_id, merge_request_iid,
"单次PR涉及的函数代码的过多")
        close_merge_request(project_id, merge_request_iid)
    return {"status": "ok"}

```

实现对大模型的访问 代码检查 16/16分

截图与链接

```
root@devops-student-run: X + v
@@ -45,6 +47,9 @@ public class UserServiceImpl implements UserService {
    */
    @Override
    public UserEntity findByUserName(String username) {
+
+       System.err.println("code diff test");
+
+       return userDao.findByUsername(username);
    }

origin_file_path: /tmp/tmpiecl86_x.java
提取的函数在这里@Override
    public UserEntity findByUserName(String username) {

        System.err.println("code diff test");

        return userDao.findByUsername(username);
    }
}
除报错外所有函数提取成功!
length: 3310
AI评审中...
评审结果:
该函数实现了注册用户的功能，首先通过用户名查询数据库中是否存在该用户，如果存在则抛出异常，否则保存新用户的数据到数据库中。其中，密码使用BCrypt进行加密处理。该函数逻辑清晰，但未对可能出现的异常进行处理。建议在save方法中加入try-catch块，以防止数据库操作异常。
好的，代码评审如下：此方法为重写父类方法，主要功能为根据用户名查找UserEntity对象。代码逻辑较为简单，主要依赖于userDao对象的findByUsername方法。需要注意的是，打印语句可能会对性能产生一定影响，在实际使用中需要酌情处理。
INFO:      210.28.130.14:54786 - "POST /webhook HTTP/1.1" 200 OK
```

关键代码展示

```
for item in diff_list:
    content = ('请你用简要的中文语言评审以下代码，如果存在问题，给出具体的问题
    描述（注意不能是格式化或增加注释这类无意义的话）和解决方案。'
    + '用Markdown格式返回，并在最后建议是否接受这次merge
    request'
    + '（注意仅回答一个字“是”或者“否”并保证这是整片回答的最后一个
    字）\n'
    + '```待评审函数开始' + item[0] + '```待评审函数结束')
    rs = api_AI_post(content)
    # 处理异常
    if rs is None:
        comment_merge_request(project_id, merge_request_iid, "from
        GenAI: 大模型评审失败")
        close_merge_request(project_id, merge_request_iid)
    return {"status": "ok"}
```

构建符合任务需要的prompt 24/24分

截图与链接

```
for item in diff_list:
    content = ('请你用简要的中文语言评审以下代码，如果存在问题，给出具体的问题描述（注意不能是格式化或增加注释这类无意义的话）和解决方案。'
              + '用Markdown格式返回，并在最后建议是否接受这次merge request'
              + '（注意仅回答一个字“是”或者“否”并保证这是整片回答的最后一个字）\n'
              + '```待评审函数开始' + item[0] + '```待评审函数结束')
    rs = api_AI_post(content)
```

关键代码展示

```
for item in diff_list:
    content = ('请你用简要的中文语言评审以下代码，如果存在问题，给出具体的问题描述（注意不能是格式化或增加注释这类无意义的话）和解决方案。'
              + '用Markdown格式返回，并在最后建议是否接受这次merge request'
              + '（注意仅回答一个字“是”或者“否”并保证这是整片回答的最后一个字）\n'
              + '```待评审函数开始' + item[0] + '```待评审函数结束')
    rs = api_AI_post(content)
```

将大模型结果返回到Merge Request页面 8/8分

截图与链接



Gitlab_bot @project_10655_bot_6c8200c58360dc5db121d0960a0d8007 开启有关差异的主题 ^ 隐藏主题
2小时前

src/main/java/org/fffd/l23o6/service/impl/UserServiceImpl.java

```
33 33         throw new BizException(BizError.USERNAME_EXISTS);
34 34     }
35 35
36 +     System.err.println("code diff test");
37 +
```



Gitlab_bot

@project_10655_bot_6c8200c58360dc5db121d0960a0d8007

· 2小时前

from GenAI: 存在的问题:

1. 未进行password的加盐加密, 可能导致密码安全问题。
2. 使用System.err.println()打印代码差异测试信息, 这可能会暴露敏感信息。

解决方案:

1. 使用BCrypt等加盐加密库对密码进行加密。
2. 将打印语句删除或替换为更安全的打印方式, 如日志记录。

建议: 否

回复...

解决主题



关键代码展示

```
# 调用大模型
sha = api_get_sha(project_id, merge_request_id)
print("AI评审中...")
for item in diff_list:
    content = ('请你用简要的中文语言评审以下代码, 如果存在问题, 给出具体的问题描述 (注意不能是格式化或增加注释这类无意义的话) 和解决方案。'
              + '用Markdown格式返回, 并在最后建议是否接受这次merge request'
              + ' (注意仅回答一个字“是”或者“否”并保证这是整片回答的最后一个字)\n'
              + '```待评审函数开始' + item[0] + '```待评审函数结束')
    rs = api_AI_post(content)
    # 处理异常
    if rs is None:
        comment_merge_request(project_id, merge_request_id, "from GenAI: 大模型评审失败")
        close_merge_request(project_id, merge_request_id)
        return {"status": "ok"}
# 推送到评论区
```



```

flag = get_diffNote_postion(item[1])
tmp = api_post_diffNote(project_id, merge_request_id,
sha.get("base_commit_sha"), sha.get("head_commit_sha"),
sha.get("start_commit_sha"), new_path,
old_path, flag[0] if flag[1] == 1 else None,
flag[0] if flag[1] == 0 else None,
'from GenAI: \n' + rs)
print(rs)

```

Code Review 方向

大模型判断是否接受PR 16/16分

截图与链接

The screenshot displays a GitLab merge request (MR) for the file `src/main/java/org/fffd/I23o6/service/impl/TrainServiceImpl.java`. The merge request is titled "大模型测试" and is currently in the "Code diff test" state. The diff shows a new line of code being added:

```

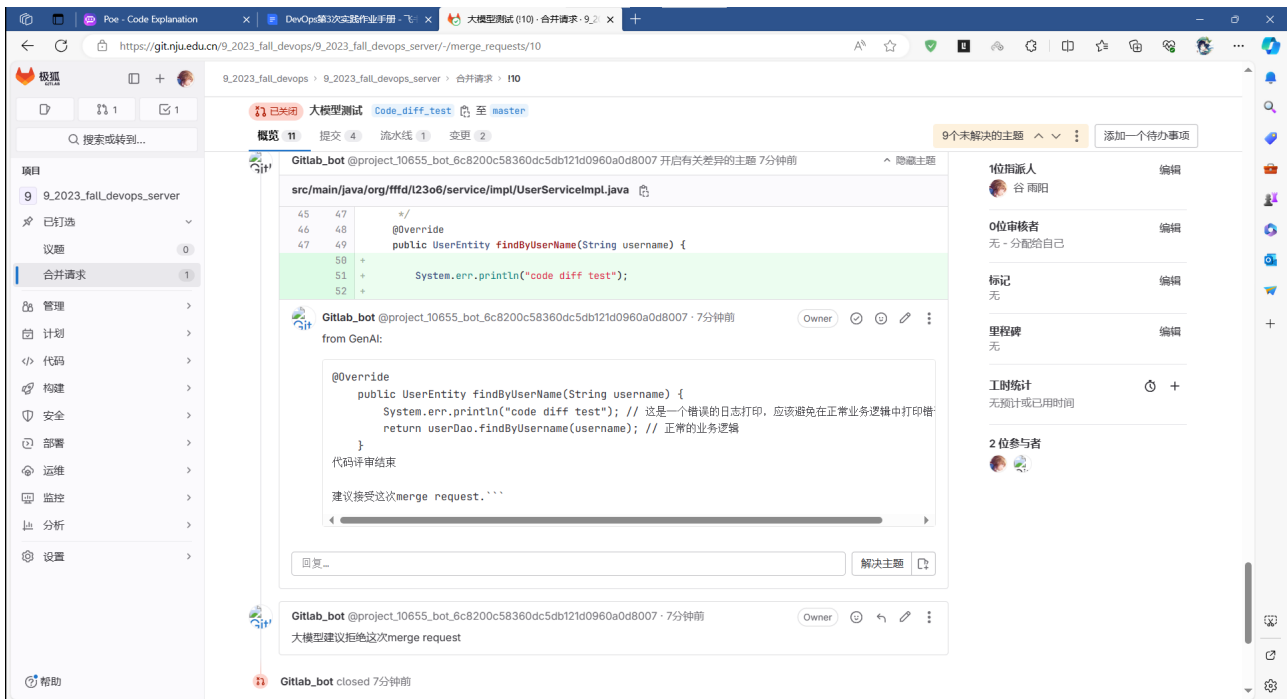
36 + System.err.println("code diff test");
37 +

```

A comment from the bot "Gitlab_bot" is visible, stating: "大模型建议接受这次merge request". Below the diff, there is a section for "从 GenAI: 代码评审:" (Code review from GenAI) with the following suggestions:

1. 在代码中使用了未初始化的变量 `System.err.println("code diff test");`, 建议进行初始化或者移除该语句。
2. 对于异常的处理, 如果 `user` 为 `null`, 抛出异常, 但未进行相应的处理, 建议进行处理。
3. 在 `UserEntity.builder` 中, `mileagePoints` 和 `privilege` 字段未进行赋值, 建议在调用 `build` 方法前进行赋值。

The right sidebar shows the merge request details, including the assignee (谷雨阳), reviewers (0), and participants (2).



根据大模型建议关闭的MR: https://git.nju.edu.cn/9_2023_fall_devops/9_2023_fall_devops_server/-/merge_requests/10

关键代码展示

```
# 调用大模型
sha = api_get_sha(project_id, merge_request_iid)
print("AI评审中...")
close = False
for item in diff_list:
    content = ('请你用简要的中文语言评审以下代码，如果存在问题，给出具体的问题描述（注意不能是格式化或增加注释这类无意义的话）和解决方案。'
              + '用Markdown格式返回，并在最后建议是否接受这次merge request'
              + '（注意仅回答一个字“是”或者“否”并保证这是整片回答的最后一个字）\n'
              + '```待评审函数开始' + item[0] + '```待评审函数结束')
    rs = api_AI_post(content)
    # 处理异常
    if rs is None:
        comment_merge_request(project_id, merge_request_iid, "from GenAI: 大模型评审失败")
        close_merge_request(project_id, merge_request_iid)
        return {"status": "ok"}
    # 推送到评论区
    flag = get_diffNote_postion(item[1])
```

```

tmp = api_post_diffNote(project_id, merge_request_id,
sha.get("base_commit_sha"),
                                sha.get("head_commit_sha"),
                                sha.get("start_commit_sha"), new_path,
                                old_path,

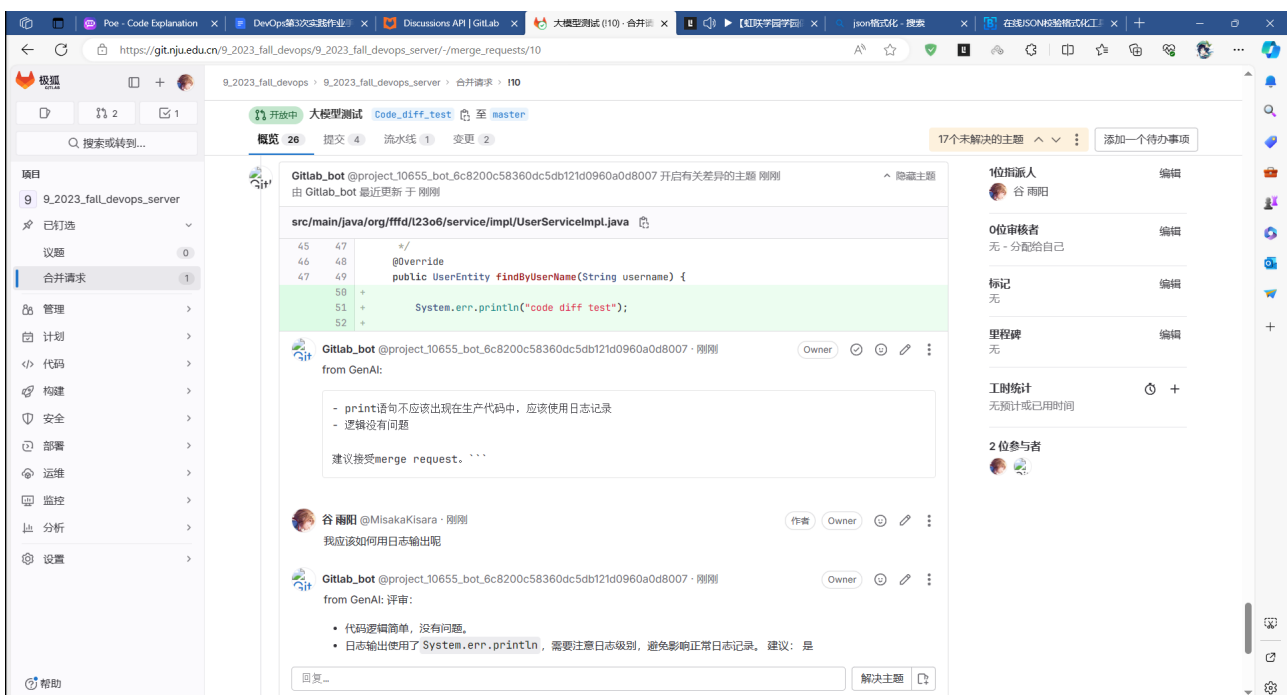
                                flag[0] if flag[1] == 1 else None,
                                flag[0] if flag[1] == 0 else None,

'from GenAI: \n' + rs)
    # 提取AI的建议
    if rs[-1] == '否':
        close = True
    print(rs)
if close:
    comment_merge_request(project_id, merge_request_id, "大模型建议
拒绝这次merge request")
    close_merge_request(project_id, merge_request_id)
else:
    comment_merge_request(project_id, merge_request_id, "大模型建议
接受这次merge request")
    reopen_merge_request(project_id, merge_request_id)

```

与大模型讨论PR 12/12分

截图与链接



与大模型讨论的MR: https://git.nju.edu.cn/9_2023_fall_devops/9_2023_fall_devops_server/-/merge_requests/10

关键代码展示

```
# 调用大模型是将变更加入缓存
set_list(json.loads(tmp).get("id"), content)

...

if data["object_kind"] == 'note':
    # 实现 note 相关操作的代码
    project_id = data["project_id"]
    merge_request_iid = data["merge_request"]["iid"]
    discussion_id = data["object_attributes"]["discussion_id"]
    description = data["object_attributes"]["description"]
    if match_string(description) == 1: # 过滤AI的回复
        print("Received note from AI")
        return {"status": "ok"}
    post_discussion_note(project_id, merge_request_iid,
                        discussion_id, description)
```

总结与感想

大模型理解代码的能力暂时比较薄弱，且不能从整体项目的角度出发。未来大模型的迅猛发展，这些问题得以解决，将为软件行业带来巨大变革。