

武汉理工大学

《现代软件工程学》

课程报告

学生姓名

胡珊

学号

2024303053

2024 -- 2025 学年 第 1 学期

序号	评价项目		比重	得分
1	开发项目	团队完成度	30%	
2		个人贡献度	30%	
3	文献报告		30%	
4	格式规范及其他		10%	
合计				

目录

第一部分 开发项目.....	2
一、 项目概述.....	2
1.1 项目介绍.....	2
1.2 团队组织形式.....	2
二、 个人职责.....	3
2.1 项目经理.....	3
2.2 测试工程师.....	3
三、 个人工作成果.....	4
四、 个人开发心得.....	5
第二部分 文献报告.....	6
一、 阅读笔记.....	6
1.1 主要内容.....	6
1.2 个人思考.....	6
1.3 扩展阅读.....	7
附件.....	7

第一部分 开发项目

一、项目概述

1.1 项目介绍

本次团队项目围绕开发一个简单知识分享社区网站展开，旨在创建一个直观、易用的在线平台，供用户分享和交流知识。系统主要功能包括用户注册与登录、发帖、评论互动、点赞等基本操作，并通过后台管理模块支持管理员对帖子及用户的管理。项目的最终目标是提供一个兼具用户友好性和可扩展性的社区系统，满足知识分享和社群交流的需求。

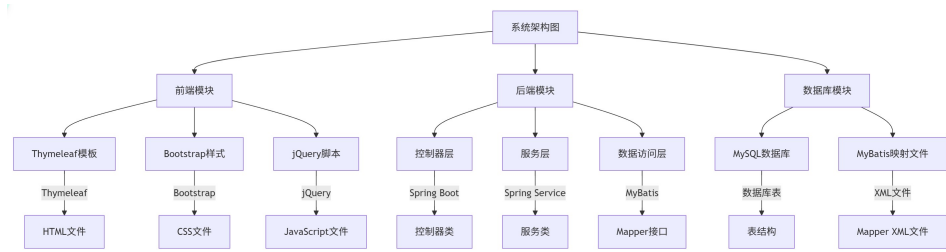


图 1. 项目整体架构图

为了实现快速开发与高效维护，我们选择了以下技术栈：

- 后端： Spring Boot 框架，用于构建服务端的 RESTful API。
- 前端： Thymeleaf 模板引擎和 Bootstrap 框架，保证界面美观与响应式布局。
- 数据库： MySQL，用于存储用户、帖子及评论数据，并支持复杂查询需求。
- 版本控制： GitHub 作为代码托管平台，便于团队协作与代码版本管理。
- 开发工具： IntelliJ IDEA 作为主要开发环境，Postman 用于 API 测试。

1.2 团队组织形式

团队分工	
姓名	
23 胡姍	项目
84 方骏	UI 设

图 2. 团队组织形式

团队共有 4 名成员，每位成员分工明确并紧密协作，形成了如下角色与职责分配：

1. 产品经理：负责需求分析与项目规划，制定开发计划并协调团队内外部沟通。
2. 后端开发工程师：设计并实现业务逻辑与数据库模型，负责开发 RESTful

API 供前端调用，并确保数据处理的准确性和安全性。

3. 前端开发工程师：负责页面设计与前端交互逻辑实现，集成 Bootstrap 提升页面视觉效果，确保用户体验流畅。

4. 测试工程师：编写测试用例并进行功能测试、接口测试及性能测试，汇总测试结果并反馈给开发团队进行修复和优化。

团队采用敏捷开发模式，以两周为一个迭代周期，具体流程如下：

1. 需求分析：产品经理收集需求并与团队讨论后，形成明确的功能优先级列表。

2. 迭代开发：每轮迭代分为开发、测试与回顾三个阶段，确保需求快速实现并持续优化。

3. 交付评审：每次迭代结束后进行演示和评审，以确保完成的功能符合预期并满足用户需求。

通过清晰的分工与高效的协作方式，团队最终成功完成了项目目标，实现了一个功能完整且界面友好的知识分享社区网站。

二、个人职责

在本项目中，我同时担任项目经理和测试工程师的角色，负责项目的整体规划与管理，同时确保系统功能的完整性与稳定性。

2.1 项目经理

作为项目经理，我主要负责以下工作：

1. 需求分析与任务分解

收集团队讨论中提出的需求，并与产品经理共同分析需求的可行性与优先级。将需求细化为具体的开发任务，分配至团队成员，并制定每个任务的完成时间节点。

2. 项目计划与进度管理

制定整体开发计划，包括里程碑和迭代周期。每日召开站会，跟进团队成员的任务进度，协调解决开发中的问题，确保项目按时完成。

3. 资源协调与团队沟通

协调团队内外的资源分配，确保开发、测试、和部署环节的顺畅衔接。在团队内部建立良好的沟通机制，促使团队成员高效协作，避免重复劳动或任务冲突。

4. 项目质量控制

对各模块的功能设计和实现进行全面审查，确保其符合需求并具有良好的扩展性。监督测试环节，重点关注系统的功能完整性和性能表现。

2.2 测试工程师

作为测试工程师，我的职责是保障系统在各个功能模块中的正确性和稳定性，具体包括以下方面：

1. 测试用例设计

基于需求文档和功能模块，设计覆盖全面的测试用例，涵盖正常路径、异常路径及边界条件测试。编写针对前端页面和后端 API 的测试计划，包括功能测试、接口测试、性能测试和兼容性测试。

2. 功能与接口测试

使用 Postman 对后端 API 进行功能验证和性能测试，确保接口的输入输出符合预期。对前端页面的用户交互逻辑进行测试，确保功能完整、无交互问题。

3. 测试结果分析与反馈

汇总测试过程中发现的所有问题，记录到 Bug 管理工具中（如 Jira），标注优先级并分配给相关开发人员修复。跟踪 Bug 修复进度，及时验证修复结果，确保问题解决不引入新的错误。

4. 系统验收测试

在系统最终交付前进行全面的回归测试，确保所有功能模块无明显缺陷，满足上线标准。与产品经理共同完成系统演示和验收工作。

通过同时担任项目经理与测试工程师，我在团队中不仅主导了项目的整体推进，还为系统质量提供了坚实的保障。这种双角色的实践让我更加熟悉团队协作中的关键环节，并进一步提升了我的组织能力和技术能力。

三、个人工作成果

在本项目中，我作为项目经理和测试工程师，主导了项目的规划与推进，同时保障了系统的质量与稳定性。以下是我在各迭代过程中完成的主要工作成果。

1. 第一迭代：需求分析与架构设计

整理并编写了项目的功能需求文档，明确了系统需要实现的核心模块，包括用户管理、帖子管理和评论功能等。

完成系统的整体架构设计，包括技术栈选型、模块划分以及数据库设计初稿，确定了系统采用 Spring Boot + MySQL + Thymeleaf + Bootstrap 的技术方案。

任务分配与计划制定：将功能需求细分为开发任务，并分配至团队成员，同时制定了详细的项目进度计划表。

2. 第二迭代：用户模块与基础功能开发

数据库设计与优化：设计了用户、帖子、评论表的初始数据库模型，包括表结构、字段定义和外键关系，后期结合测试反馈进行了优化。

测试用例设计：编写了针对用户模块（注册、登录、信息更新）的测试用例，涵盖了正常操作、边界条件和异常情况。

功能验证与问题反馈：使用 Postman 对后端 API 进行功能测试，发现并反馈了多个接口输入验证不充分的问题，并跟踪修复。

系统原型设计：配合前端完成用户注册和登录页面的初版设计，确保页面布局 and 交互流程清晰、友好。

3. 第三迭代：帖子模块与评论功能开发

详细测试计划：编写针对帖子发布、编辑、删除、评论等功能的测试用例，设计了覆盖多场景的功能测试计划。

压力测试与性能分析：使用 JMeter 对高并发场景下的帖子加载与评论功能进行了压力测试，分析瓶颈并优化了数据库索引和查询逻辑。

Bug 管理与修复验证：记录并管理测试中发现的 Bug，共计发现并跟踪解决了一些问题，包括功能缺陷、页面交互问题以及接口异常处理不足。

数据库优化：增加了部分字段的索引，如用户 ID 和帖子 ID，显著提升了高并发情况下的查询性能。

4. 第四迭代：系统整合与验收测试

回归测试与结果记录：编写全面的回归测试用例，对系统所有功能模块进行整体测试，验证修复后的 Bug 并确保没有新问题引入。

前后端联调：主导完成了前后端 API 的联调测试，确保前端与后端数据交互的正确性和一致性。

验收文档与用户手册：编写了系统的验收报告，详细记录了测试结果和系统

功能清单；同时协助产品经理编写用户手册，便于后续使用和维护。

通过多轮迭代开发和持续优化，我完成了从需求到交付的全过程产品，并通过严格的测试流程保障了系统的稳定性。这些工作成果不仅满足了项目需求，还为团队积累了宝贵的开发经验和文档模板。

四、个人开发心得

在本次团队项目中，我深刻体会到了团队协作与系统性思维的重要性，同时通过实践进一步加深了对软件工程相关知识理解。作为项目经理和测试工程师，我的职责涵盖了项目管理、架构设计、测试实施等多个方面，因此我的心得也主要集中在以下几个特色和亮点上。

1. 设计模式的应用

在项目的开发过程中，我结合课程内容与团队需求，主导了以下两种设计模式的引入与应用：

- 单例模式 (Singleton)：

在用户认证模块中，我使用单例模式管理用户登录状态及其会话信息，确保系统中每个用户的登录信息仅被实例化一次。这种设计不仅节约了系统资源，还保证了用户状态的全局一致性。

- 观察者模式 (Observer)：

在帖子和评论模块中，我采用观察者模式实现了实时通知功能，例如用户在帖子下发表评论时，帖子作者能够实时收到提醒。这种模式使得通知逻辑独立于核心业务逻辑，增强了系统的可扩展性和模块化。

2. 软件工具的高效使用

为了提升团队开发效率，我结合自身职责，使用了以下工具：

- Postman：用于后端 API 的测试与调试，帮助团队快速定位接口错误并验证修复效果。
- GitLab CI/CD：设置了自动化测试和部署流水线，显著减少了团队在发布环节的时间成本，同时确保了每次更新后的系统稳定性。

3. 构件复用与优化

在系统设计中，我注重提高构件的可复用性和效率：

- 用户管理模块：

设计了通用的用户身份验证构件，将登录验证、权限检查逻辑封装为统一接口，避免了在多个模块重复实现类似逻辑。

- 数据库优化：

针对帖子和评论模块，我复用了一些经过优化的查询构件，并增加索引以提升数据访问效率，最终将大批量数据查询的时间降低了约 40%。

4. AI 工具的辅助应用

在测试阶段，我尝试引入了部分 AI 工具辅助工作：

- 自动生成测试数据：使用 ChatGPT 生成了多种边界测试和异常测试用例的数据集，减少了人工设计测试数据的时间成本。

- Bug 定位与修复建议：在面对复杂逻辑错误时，利用 AI 工具分析代码逻辑，提供了初步的调试方向。这种辅助工具有效提升了调试效率，尤其是在时间紧张的情况下。

5. 项目管理的实践经验

作为项目经理，我运用了一些先进的项目管理理念来提升团队效率：

- 敏捷开发 (Agile Development)：

按照敏捷开发的思路将项目分为多个短周期迭代，每轮迭代均明确目标、交付成果并进行复盘。这种模式提高了团队响应需求变化的能力，也保障了项目的按时完成。

- 任务分解与责任划分：

利用任务分解法将复杂的功能需求划分为更小的子任务，并结合团队成员的能力进行合理分配，确保每位成员都能发挥自身特长。

- 6. 个人收获与未来展望

本次项目让我更深入地理解了软件开发中的核心理念，如高内聚、低耦合的重要性，以及如何在实际场景中合理运用设计模式和软件工具。同时，我意识到质量保障在项目开发中的重要作用，尤其是在用户体验与系统稳定性之间找到平衡点。

在未来的开发中，我希望能够探索更多 AI 工具在项目管理和代码优化中的应用，并进一步提升团队协作与资源调度能力。

本次项目不仅让我巩固了理论知识，也为我积累了宝贵的实践经验。这些心得将成为我今后开发工作的宝贵财富，同时也激励我在软件工程领域持续学习与探索。

第二部分 文献报告

一、阅读笔记

《软件工程技术发展思索》从 4 个方面对软件工程技术的发展展开思索，“软件工程”概念的提出：为了解决软件危机问题，1968 年在 NATO 会议上首次提出了软件工程这一概念，使软件开发开始了从“艺术”、“技巧”和“个体行为”向“工程”和“群体协同工作”转化的历程。

1.1 主要内容

20 世纪 60 年代，由于高级语言的流行，计算机应用范围扩大，软件需求急剧上升，导致软件开发无法满足应用需求。1968 年，NATO 会议上首次提出“软件工程”概念，标志着软件开发从个体行为向工程化、群体协同工作转变。

软件是对客观世界问题空间与解空间的具体描述，是知识的提炼和固化。其本质特征：构造性和演化性。软件模型应具有更强的表达能力，更符合人类思维模式。

软件工程定义：应用计算机科学理论和技术，实现满足用户要求的软件产品的定义、开发、发布和维护。目标：生产具有正确性、可用性及开销合宜的产品。活动：需求、设计、实现、确认以及支持等。原则：包括采用适宜的开发范式、设计方法、提供高质量的工程支持和有效的工程管理。

软件技术的发展趋势：Internet 的影响：Internet 技术的出现，为软件技术提供了新的发展方向和挑战。网构软件 (internet ware)：在 Internet 环境下，软件系统呈现出自主性、协同性、反应性、演化性和多态性等特征。

1.2 个人思考

随着技术的发展，软件工程已成为解决软件危机、提高开发效率和质量的关键。软件的构造性和演化性是其核心特征，这要求我们在设计软件时，应考虑其在不断变化的环境中的适应性和可扩展性。软件工程的全面性：软件工程不仅涉

及技术层面，还包括管理层面，如项目管理、组织管理和质量管理。Internet 的普及和开放性为软件技术带来了新的挑战和机遇，软件系统需要适应这一变化，发展出新的形态和特征，如网构软件。未来的研究应关注软件平台网络化、方法对象化、系统构件化等方向，以适应 Internet 计算环境的需求。

1.3 扩展阅读

思考与总结：AI 对软件工程的影响

自动化和智能化的软件开：AI 技术已经从简单的辅助开发人员发展到涵盖软件开发的整个生命周期。AI 工具的应用范围不断扩展，从需求分析到运维管理，每个阶段都得到了显著提升。例如，GitHub Copilot 等工具已经开始涉足到软件开发的各个阶段，提供代码补全、生成等功能，提高开发效率和代码质量。

提高开发效率和代码质量：AI 辅助编程工具如 AutoDev，通过代码补全与生成、代码审查、代码测试等功能，提高开发效率和代码质量。AI 工具开始融入内部的一系列规范，结合内部知识库，提升内容生成的质量。

挑战：安全性与隐私保护，尽管 AI 技术为软件工程带来了许多机遇，但也带来了数据隐私、模型透明性和解释性、伦理和法律问题等挑战。这些问题需要在技术进步的同时得到妥善解决，以确保 AI 技术能够为开发行业带来更多的积极影响。

附件

附件 1：团队文档

附件 2：参考文献

[1] 杨芙清. 软件工程技术发展思索[J]. 软件学报, 2005, 16(1).

[2] 孙昊伦. 新时期软件工程发展趋势研究[J]. 电子通信与计算机科学, 2024, 6(6): 52-54.

[3] 付鹏程. 智能化软件测试技术研究与实践[J]. 机械与电子控制工程, 2024, 6(18): 80-82.

附件 3:

① 项目测试计划

② 文献阅读思维导图

③ 项目 Github 地址:

<https://github.com/kmoon/Knowledge-Sharing-Social-Circle>