**项目总结报告**

日期：2024/06/20

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 15 | 项目名称 | EMall |
| 编程语言 | Java | 开发平台和框架 | Vue+ Spring Boot+Mysql |

|  |
| --- |
| **项目工作小结** |
| 1.采用哪种架构风格？哪些设计模式？  **分层架构（Layered Architecture） 微服务架构（Microservices Architecture）**  **MVC模式（Model-View-Controller）工厂模式（Factory Pattern）**  **单例模式（Singleton Pattern）观察者模式（Observer Pattern）**  2.技术方案有哪些亮点？   * **读写分离**：将数据库的读操作和写操作分离到不同的数据库实例，提高数据库的并发处理能力和性能。 * **水平分片**：对数据进行水平分片，将数据分散到多个数据库实例，减少单个数据库的压力，提高系统的可扩展性。 * **统一入口：**所有外部请求通过API网关进入系统，简化了客户端的调用逻辑，提供统一的接口管理。 * **安全管理**：在API网关层面实现认证、授权和请求过滤，提高系统的安全性。 * **智能客服**：利用大模型构建智能客服系统，能够快速、准确地解答用户问题，提高客服效率和用户满意度。 * **内容生成**：借助大模型自动生成商品描述、营销文案等，提高内容生产效率，确保高质量的文本输出。   3.是否做了单元测试？是否做了系统的功能测试、性能测试、兼容性测试和易用性测试等？  4.是否采用大模型来辅助开发？对开发生产率约有百分之几的提升？  **代码生成与优化**：通过大模型自动生成代码片段和优化已有代码，减少重复性劳动，提高开发速度。  **文档撰写：**利用大模型自动生成技术文档、API文档和用户手册，提高文档撰写的效率和质量。  **需求分析与设计**：大模型帮助分析用户需求和系统设计，生成初步的需求文档和设计方案，减少前期调研时间。  **错误检测与修复：**大模型在代码审查和单元测试过程中，帮助检测潜在错误并给出修复建议，提升代码质量。  引入大模型辅助开发对开发生产率的提升约为30%至50%。具体提升比例因项目复杂度和团队熟练程度而有所不同。 |
| **项目组成员对项目的贡献度（%）** |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 姓名 | 需求 | 设计 | 编码 | 测试 | 项目管理\* | 小计 | | 刘嘉毅 | 60 | 40 | 50 | 30 | 70 | 50 | | 李若彬 | 40 | 60 | 50 | 70 | 30 | 50 | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |   注： 项目管理包括项目计划、报告、沟通与协调等。 |

|  |  |
| --- | --- |
| **软件规模** | |
| 前端的代码行数（不包括注解行、空行和复用代码）： | 4019 |
| 后端的代码行数（不包括注解行、空行和复用代码）： | 2136 |

|  |
| --- |
| **经验、教训和建议** |
| * + 项目管理：采用敏捷开发或其他适合的项目管理方法，确保开发进度可控。   + 里程碑设定：设定明确的里程碑和阶段性目标，定期评估进展，调整计划。   + 风险管理：识别并管理开发过程中的风险，及时采取措施降低风险影响   + 整体视角：在开发初期，首先要从整体角度规划系统架构，确保设计合理、可扩展。   + 模块清晰：明确划分各个模块的功能与边界，确保模块之间的接口清晰、通信顺畅。   + 架构评审：定期进行架构评审和改进，及时发现并解决潜在问题。   + 团队协作：根据每个成员的技能和优势，合理分配任务，促进团队协作。   + 激励机制：建立有效的激励机制，如奖励制度、技能提升机会等，激发团队成员的积极性。   + 反馈机制：定期进行团队反馈和评估，及时解决问题，提升团队效率 |

项目组各成员签字：刘嘉毅 李若彬