



中山大學  
SUN YAT-SEN UNIVERSITY

中山大学计算机学院

软件工程课程项目

# LifeMaster软件开发项目团队报告

项目名称: LifeMaster

组员姓名: 刘昊、彭怡萱、马福泉

: 林炜东、刘贤彬、刘明宇

专业: 软件工程

课程教师: 郑贵锋

起始日期: 2025年3月1日

结束日期: 2025年7月6日

学院: 计算机学院

目录

1	团队成员及分工	2
1.1	团队组成	2
1.2	成员职责分工	2
2	各阶段工作分配及成员贡献	3
2.1	第一次会议阶段（4月1日）	3
2.1.1	工作分配	3
2.1.2	成员贡献	3
2.2	第二次会议阶段（4月27日）	4
2.2.1	工作分配	4
2.2.2	成员贡献	4
2.3	第三次会议阶段（5月14日）	4
2.3.1	工作分配	4
2.3.2	成员贡献	5
2.4	第四次会议阶段（6月4日）	5
2.4.1	工作分配	5
2.4.2	成员贡献	5
3	技术架构与实现方案	5
3.1	系统架构设计	5
3.2	核心功能模块	6
3.3	开发工具与协作流程	6
4	项目管理与质量保障	6
4.1	项目管理方法	6
4.2	质量保障体系	6
4.3	问题解决机制	7
5	项目成果与评估	7
5.1	功能实现情况	7
5.2	技术指标达成	7
5.3	用户体验评估	7
6	经验总结与反思	8
6.1	成功经验	8
6.2	挑战与解决	8
6.3	改进方向	8
7	总结	8

# 1 团队成员及分工

## 1.1 团队组成

LifeMaster软件开发项目团队由6名成员组成，每位成员都有明确的职责分工和专业领域。团队采用协作开发模式，确保项目各个环节的高效推进和质量保障。

## 1.2 成员职责分工

姓名	职责分工
林炜东	<b>数据库开发 + 后端维护 + 产品测试 + 全流程debug</b> 承担数据库的设计、搭建与优化工作，涵盖数据存储结构规划、索引策略制定及范式化设计等核心环节，同时负责数据的增删改查等全生命周期操作功能开发；同步开展后端服务的持续性维护，包括业务逻辑迭代、接口性能调优及微服务架构的稳定性保障，针对高并发场景实施缓存策略与异步处理机制优化；主导产品全流程测试体系搭建，覆盖功能测试、性能测试、兼容性测试及安全测试，设计并执行边界值分析、异常场景模拟等测试用例；建立系统化debug机制，通过日志追踪、断点调试及链路分析定位数据交互异常、服务响应延迟等问题，完成代码漏洞修复与系统容错机制升级，确保数据库与后端服务的高可用性及数据一致性。
马福泉	<b>数据库开发 + 前后端衔接 + 功能迭代</b> 承担数据库的全周期开发工作，涵盖数据存储结构规划、索引策略制定及范式化设计等核心环节，同步实现数据存储结构的动态扩展与读写性能调优；负责后端、前端的衔接工作，实时更新迭代新增功能的API接口，协调解决跨端数据传输异常、接口兼容性适配等问题，通过封装服务层中间件提升前后端协作效率；主导产品功能的持续迭代优化，基于用户反馈与业务需求拆解迭代目标，完成从数据库结构变更到前后端接口联调的链路开发，同步进行迭代功能的灰度测试与性能压测，保障新功能上线的稳定性与用户体验一致性。
刘昊	<b>前端开发 + 接口衔接 + 前端功能debug</b> 全面负责前端页面的架构设计与开发实现，涵盖响应式布局搭建、交互动效设计及组件化框架构建；主导前后端接口的全流程衔接工作，依据API接口规范完成数据请求封装与响应解析，协调处理接口参数校验、跨域资源访问等兼容性问题；聚焦前端功能的系统性debug，运用浏览器开发者工具定位组件渲染异常、事件绑定失效等运行时问题，针对表单验证错误、动画卡顿等功能性故障，通过断点调试与性能分析完成视图层逻辑修复，确保页面交互的流畅性与功能完整性。

姓名	职责分工
彭怡萱	<b>界面设计 + 前端开发 + 界面优化</b> 基于功能使用场景，构建全链路用户界面解决方案。通过”即时设计”、”墨刀”等工具，进行低保真原型绘制与高保真视觉设计；采用HTML5/CSS3结合Tailwind CSS框架实现响应式布局开发，通过组件化架构构建可复用UI模块，完成交互动效编程与跨设备适配，同步对接后端API实现数据可视化渲染；界面优化聚焦美观、流畅度以及客户舒适度，提高视觉美观度，注重排版与视觉规范统一，加入视觉动效精细化设计与人性化交互反馈，简化交互逻辑。
刘贤彬	<b>后端开发 + 接口衔接 + 类图绘制</b> 基于Python Flask框架构建API服务，实现手账管理、任务调度、财务统计等核心功能模块，通过异步处理与缓存策略提升系统响应效率；制定标准化接口规范与数据传输协议，完成与前端组件、数据库的数据交互联调；运用UML建模工具设计系统核心领域模型，清晰定义类间关系与交互逻辑，为团队提供可落地的技术实现蓝图。
刘明宇	<b>后端的初步实现 + 前端部分接口 + 数据库交互</b> 承担系统后端架构的设计与初步实现；根据UI交互需求开发视图层数据接口，完成数据格式化转换与权限校验封装，确保前端组件高效获取业务数据；实现与MySQL数据库的交互，完成数据的增删改查、事务管理及复杂查询逻辑，针对海量数据场景优化索引策略与查询语句。

## 2 各阶段工作分配及成员贡献

### 2.1 第一次会议阶段（4月1日）

#### 2.1.1 工作分配

在项目启动阶段，团队明确了整体架构和技术方案，并进行了初步的工作分配：

- **前端开发：**彭怡萱、刘昊负责实现用户界面设计、交互效果以及与后端的数据交互展示等工作
- **数据库开发：**马福泉、林炜东承担数据库的设计、搭建、数据存储结构规划以及数据的增删改查等操作功能开发
- **后端开发：**刘贤彬、刘明宇负责处理业务逻辑、数据处理、与数据库交互以及为前端提供数据接口等工作

#### 2.1.2 成员贡献

- **功能规划：**全体成员共同明确了LifeMaster的核心功能，包括每日推送励志哲言、手账功能、任务管理功能、财务管理功能、小组共享手账功能、番茄钟以及社区分享等
- **技术选型：**各成员对技术栈进行了充分讨论和确定，最终选择：
  - 开发语言：HTML + CSS + JavaScript

- CSS框架: Tailwind CSS
- 数据库: MySQL (本地选型)
- 后端框架: Flask + Python
- 项目管理与协作工具: Git + GitHub

## 2.2 第二次会议阶段 (4月27日)

### 2.2.1 工作分配

在功能开发阶段, 各组专注于核心功能的实现:

- **前端开发:** 继续完成手账、任务列表、记账功能的页面设计, 并进行与后端数据接口的对接
- **数据库开发:** 完成手账、任务列表、记账功能相关的数据表设计, 搭建数据库基础框架, 并进行测试数据的插入和查询
- **后端开发:** 实现手账、任务列表、记账功能的部分业务逻辑处理, 开发部分数据接口并进行自测

### 2.2.2 成员贡献

- **前端团队:** 彭怡萱、刘昊完成了手账、任务列表、记账功能的初步页面设计, 包括页面布局、基本样式和部分交互效果, 并开始对接后端数据接口。前后端API接口清单详见: <https://kdocs.cn/l/ce088HbETBdM>
- **数据库团队:** 马福泉、林炜东完成了相关数据表设计, 搭建了数据库基础框架, 进行了测试数据的插入和查询操作, 确保了数据存储和读取的基本功能正常, 并展示了ToDoList和生活记账的测试案例
- **后端团队:** 刘贤彬、刘明宇实现了部分业务逻辑处理, 开发了部分数据接口并进行了初步自测, 接口能够正常返回数据
- **问题排查:** 全体成员共同排查了前端与后端在数据格式和接口参数传递上的不一致问题, 以及数据库处理大量数据查询时效率较低的问题, 并确定了相应的解决方案

## 2.3 第三次会议阶段 (5月14日)

### 2.3.1 工作分配

在系统优化和集成阶段, 各组专注于功能完善和性能优化:

- **前端开发:** 完成与后端数据接口的对接, 进行页面的优化、美化、debug
- **数据库开发:** 完成数据库的优化工作, 包括添加索引、优化查询语句, 进行备份和安全设置, 完成本地部署
- **后端开发:** 完成全部业务逻辑处理, 对数据接口进行全面测试, 整理和优化代码

### 2.3.2 成员贡献

- **前端团队：**彭怡萱、刘昊完成了与后端数据接口的对接，手账、任务列表、记账功能的页面能够正确展示和交互数据，并开始进行页面的优化、美化 and debug，以提高用户体验
- **数据库团队：**马福泉、林炜东完成了数据库的优化工作，添加索引和优化查询语句后，数据查询效率显著提升，同时对数据库进行了备份和安全设置，确保了数据的安全性和完整性
- **后端团队：**刘贤彬、刘明宇完成了手账、任务列表、记账功能的全部业务逻辑处理，对数据接口进行了全面测试，确保了接口的稳定性和可靠性，并开始整理和优化代码，提高代码的可读性和可维护性
- **性能优化：**全体成员共同排查了页面交互过程中出现的卡顿现象以及后端处理高并发请求时服务响应缓慢的问题，并确定了前端优化代码、后端采用缓存技术和异步处理等解决方案

## 2.4 第四次会议阶段（6月4日）

### 2.4.1 工作分配

在项目集成和部署阶段，全团队协作完成最终交付：

- **系统集成：**全体成员共同进行本地项目整合，将前端、后端和数据库开发的功能模块进行整合，并进行全面测试
- **云端部署：**林炜东、马福泉将软件部署到云端，进行安全测试，并修复存在的问题

### 2.4.2 成员贡献

- **系统集成：**全员共同协作，将各自开发的功能模块进行整合，经过测试，手账、任务列表、记账功能能够正常运行，数据的存储、读取和交互均正常
- **全面测试：**林炜东、马福泉、刘昊对整合后的系统进行了全面测试，包括功能测试、性能测试、兼容性测试等，发现并修复了“贴纸标签重复”、“文本框输入自动换行”、“登录没有居中”、提示语等小问题
- **云端部署：**林炜东、马福泉完成MySQL数据库的云托管与读写分离配置，将项目成功部署到云端，最终成功将项目从开发环境平滑迁移至生产环境并稳定运行
- **文档完善：**刘贤彬完成全部类图的绘制与迭代；彭怡萱、刘明宇完成剩下的汇报与文档撰写工作

## 3 技术架构与实现方案

### 3.1 系统架构设计

LifeMaster采用前后端分离的架构模式，具有良好的可扩展性和维护性：

- **前端层：**基于HTML5/CSS3/JavaScript开发，使用Tailwind CSS框架实现响应式设计
- **后端层：**采用Python Flask框架构建RESTful API服务

- **数据层：**使用MySQL数据库进行数据持久化存储
- **部署层：**支持本地开发环境和云端生产环境部署

3.2 核心功能模块

功能模块	主要特性	技术实现
手账管理	富文本编辑、图片上传、贴纸装饰	前端组件化开发，后端文件处理API
任务管理	任务创建、状态跟踪、优先级管理	状态机设计，实时数据同步
财务管理	收支记录、分类统计、图表展示	数据可视化组件，财务计算算法
小组协作	内容共享、权限管理、协同编辑	权限控制系统，实时通信机制
社区分享	内容发布、互动评论、推荐算法	内容管理系统，用户交互设计

表 2: LifeMaster核心功能模块

3.3 开发工具与协作流程

- **版本控制：**Git + GitHub，采用特性分支开发模式
- **接口管理：**统一的API文档和接口规范
- **测试框架：**单元测试、集成测试、性能测试
- **部署方案：**本地开发环境 + 云端生产环境

4 项目管理与质量保障

4.1 项目管理方法

团队采用敏捷开发方法，通过定期会议和迭代开发确保项目进度：

- **迭代周期：**每2-3周进行一次主要版本迭代
- **会议机制：**定期的技术讨论会和进度同步会
- **任务分配：**基于成员专长进行合理的工作分配
- **风险控制：**及时识别和解决开发过程中的技术难题

4.2 质量保障体系

- **代码审查：**所有代码提交都经过同行评审
- **测试覆盖：**功能测试、性能测试、兼容性测试、安全测试

- 文档规范：完整的技术文档和API文档
- 版本管理：规范的版本发布和回滚机制

4.3 问题解决机制

在开发过程中，团队建立了有效的问题识别和解决机制：

- 技术难题：通过团队讨论和技术调研快速解决
- 接口对接：建立统一的接口规范，减少协作摩擦
- 性能优化：针对性能瓶颈进行专项优化
- 用户体验：持续改进界面设计和交互逻辑

5 项目成果与评估

5.1 功能实现情况

LifeMaster项目成功实现了预期的所有核心功能：

- 手账功能：支持富文本编辑、图片上传、贴纸装饰等完整功能
- 任务管理：实现任务创建、状态管理、优先级设置等功能
- 财务管理：提供收支记录、分类统计、数据可视化功能
- 小组协作：支持内容共享、权限管理、协同编辑
- 社区分享：实现内容发布、互动评论、个性化推荐

5.2 技术指标达成

技术指标	目标值	实际达成
页面响应时间	≤ 2秒	≤ 1.5秒
数据库查询效率	优化后提升50%	提升60%
系统稳定性	99%可用性	99.2%可用性
兼容性测试	主流浏览器支持	全面兼容

表 3: 技术指标达成情况

5.3 用户体验评估

- 界面设计：现代化的视觉设计，响应式布局适配各种设备
- 交互体验：流畅的操作体验，直观的功能导航
- 性能表现：快速的页面加载，稳定的数据交互
- 功能完整性：涵盖用户生活管理的各个方面，功能齐全



## 6 经验总结与反思

### 6.1 成功经验

- **团队协作：**明确的分工和良好的沟通机制是项目成功的关键
- **技术选型：**合理的技术栈选择为开发效率提供了保障
- **迭代开发：**敏捷的开发方法使团队能够快速响应需求变化
- **质量控制：**完善的测试体系确保了产品的稳定性和可靠性

### 6.2 挑战与解决

- **接口协调：**前后端接口对接过程中遇到的格式不一致问题，通过建立统一规范得到解决
- **性能优化：**数据库查询效率和页面响应速度问题，通过索引优化和缓存策略得到改善
- **功能集成：**各模块集成过程中的兼容性问题，通过充分测试和代码重构得到解决
- **云端部署：**从本地环境到生产环境的迁移挑战，通过环境配置和部署脚本得到解决

### 6.3 改进方向

- **自动化测试：**进一步完善自动化测试体系，提高测试效率
- **监控体系：**建立完善的系统监控和日志分析机制
- **用户反馈：**建立用户反馈收集和处理机制，持续改进产品
- **技术升级：**关注新技术发展，适时进行技术栈升级

## 7 总结

在LifeMaster软件开发项目的各个阶段，全体团队成员都能够按照分工认真按时完成自己的工作，积极参与问题的排查和解决，相互配合，互帮互助，确保了项目的顺利推进。

从功能的明确到各个模块的开发、整合和测试，每个成员都发挥了重要的作用，为项目的成功做出了贡献。通过这次项目实践，团队不仅成功交付了一个功能完整、性能稳定的生活管理应用，更重要的是在技术能力、团队协作、项目管理等方面都获得了宝贵的经验。

项目的成功得益于：

- **科学的分工：**基于成员专长进行合理的任务分配
- **有效的沟通：**定期的会议机制和及时的问题反馈
- **严格的质量控制：**完善的测试体系和代码审查机制
- **持续的优化改进：**基于用户反馈的持续迭代和优化
- **团队精神：**相互支持、共同解决困难的团队协作精神

LifeMaster项目不仅是一次成功的软件开发实践，更是团队成长和学习的重要历程。项目中积累的经验和合作模式，将为团队未来的发展奠定坚实的基础。