**武汉大学**

**计算机学院实训**

**WHUClubSynapse项目关闭总结报告**

**项目成员：黄凯琳，刘思哲，王子安，张佳宇，马之涵**

**撰 写 人： 马之涵**  2025 年 7 月8 日

**评 审 人：**  年 月 日

武汉凡诺软件技术有限公司

2025年

**修订记录**

| 日期 | 修订版本 | 修改章节 | 修改描述 | 修订人 |
| --- | --- | --- | --- | --- |
| 7.8 | V1.0 | 一，二，三，四，五 | 初版定稿 | 马之涵 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Catalog目 录

[一、项目基本情况 4](#_Toc269733113)

[二、项目的完成情况 4](#_Toc269733114)

[三、学员任务及其工作量总结 4](#_Toc269733115)

[四、项目进度 5](#_Toc269733116)

[五、经验教训及改进建议 5](#_Toc269733117)

# 一、项目基本情况

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称： | WHUClubSynapse | | |
| 开发环境： | WSLUbuntu24.04 | 运行平台： | WEB浏览器、Ubuntu |
| 涉及技术： | RAG,LoRA,MCP,FastAPI | Iris、GORM、Redis Streams、gRPC | vue3 |
| 项目起止时间： | 6月23日-7月8日 | 项目地点： | B301实验室 |
| 项目组成员： | 黄凯琳，刘思哲，张佳宇，王子安，马之涵 | | |
| 项目描述： | WHUClubSynapse是一个基于AI的社团管理程序，其基于实现RAG和模型微调的AI与本地部署的AI实现了多样的AI辅助功能体验，用户交互模块使用VUE和NODE.JS实现现代化用户友好界面，并完整实现所有功能；业务逻辑模块使用GO实现各模块之间的数据转发与数据库调用，并通过redis实现消息队列进行数据同步；智能AI模块通过RAG实现智能搜索，通过模型微调实现专业性问答AI，本地部署的AI实现智能助手，助力社团活力增强。 | | |

# 二、项目的完成情况

（说明：描述项目总体完成情况、各模块完成情况，及总体代码规模和个人代码量）

项目总体实现了所有基础功能和部分附加功能，同时在此基础上，基于AI进行了大幅的功能拓展。项目分三个模块：用户交互模块，业务逻辑模块，体验增强模块。

用户交互模块完全完成并实现其功能。王子安：技术架构与接口适配 1. 接口对接：采用 HTTP + SSE（Server-Sent Events）组合，适配小组内全部 AI 接口的前端调用；针对不同接口特性封装统一函数库，降低耦合度，提升代码复用性。 2. Markdown 渲染：引入 markdown-it、showdown 等解析库，将 AI 返回的 Markdown 数据解析为 HTML，实现富文本展示，确保多终端美观易读。 3. 实时流传输：基于 SSE 建立服务器到前端的单向长连接，AI 响应数据以流形式推送，前端通过 EventSource 监听并动态更新页面（如实时显示生成内容）；处理中断、重连等异常，保障传输稳定。 二、AI 能力与智能交互 1. 多维度信息处理：AI 可读取页面信息、用户个人信息、输入内容及聊天历史，进行综合思考与决断。 2. 深度思考渲染：解析后端标签，对 gong根据社团帖子内容，为用户提供交流建议。 ◦ 智能审核支持：整合社团信息与用户申请信息，为管理员提供审核意见。 ◦ 经费管理辅助：自动解析活动介绍，生成结构化账单记录。 ◦ 智能聊天助手：以悬浮球形式存在，支持共性问题解答（新手引导、规则指南）、社团问题查询（基于社团数据），聊天记录可检索，回复带打字机特效，适配 SSE 流传输。 三、社团核心功能 1. 社团详情页：展示基本信息（名称、简介、成立时间），采用响应式布局；支持社团动态、公告的编辑与显示，动态和公告支持标签及特性渲染。 2. 社团互动区域： ◦ 独立发帖区：所有人可查看，仅社团成员可发帖 / 回帖，支持多层级回帖及 MD 格式输入与渲染。 ◦ 分类与搜索：支持社团按类别筛选展示；主页搜索集成 AI 智能体，辅助用户快速获取信息。 3. 社团管理功能：管理员可进入经费管理页面，查看账单、录入经费来源与去向。 四、用户与社团交互 1. 个人信息管理：支持存储、在线编辑个人信息（学号、姓名、专业等通用信息；兴趣、标签等独立信息），个人标签可自定义或选择；偏好设置中含社团推荐功能，基于用户信息与社团数据推荐适配社团及理由。 2. 社团申请与审核： ◦ 用户可发起社团申请，状态分为 “审核中 / 申请 / 已满员”，申请需填写理由。 ◦ 管理员可查看申请人信息及申请理由，参考 AI 审核意见做决策，拒绝时可填写原因。纯vue代码add33000行diffgong28000处

用户交互模块。张佳宇：负责构建了基于Vue 3 + TypeScript + Element Plus的系统架构。采用模块化设计理念，将功能按领域进行清晰划分，建立了可维护、可扩展的代码结构。在App.vue中实现了应用的核心初始化逻辑，使用Vue 3的Composition API和Pinia状态管理，通过useAuthStore和useClubStore管理全局状态。应用启动时并行初始化认证状态和社团分类数据，确保数据的完整性和一致性。设计了基于角色的路由守卫系统，实现了多级权限控制。构建了完整的路由架构，包括主布局路由、公共页面、用户功能路由、认证页面和管理员路由。游客模式创新：实现了游客模式功能。在App.vue的onMounted生命周期中实现了游客模式的自动初始化逻辑，当检测到用户未登录时，系统会自动调用authStore.login创建游客会话，并设置authStore.isGuest标识游客状态。主页社团展示系统：在主页中实现了高效的社团信息渲染机制，实现分类导航，使用网格布局展示社团卡片。设计了智能的分页系统，在ClubCard组件中通过computed属性实现动态状态管理。社团详情页功能：实现了与后端API的完整对接，包括社团详情获取、收藏功能、申请加入、发帖功能和评论系统。在auth.ts store中实现了完整的JWT token管理流程，通过localStorage存储token，使用computed属性isLoggedIn实时反映登录状态。实现了完整的个人数据管理功能，包括收藏列表管理、申请记录跟踪、社团成员管理和社团管理功能。并实现管理员后台系统

在Admin/index.vue中实现了严格的管理员权限控制，在onMounted生命周期中通过authStore.user?.role检查用户角色，非管理员用户会被重定向到首页并显示权限提示，使用基于角色的访问控制（RBAC）确保只有管理员可以访问后台功能。用户管理功能通过authStore.fetchAllUsers方法获取用户列表，支持分页和搜索功能，用户信息展示包括头像、用户名、角色等，支持用户状态的编辑和权限管理，通过el-table组件实现用户列表的展示，支持排序和筛选功能。社团管理功能通过clubStore获取所有社团信息，支持社团的查看、编辑、删除操作，使用el-card组件展示社团信息，包括社团封面、名称、成员数等关键信息，实现了社团数据的统计和分析功能。申请审核系统通过clubStore获取待审核的社团申请，支持通过和拒绝操作，使用el-dialog组件实现审核详情查看，包括申请信息和审核表单，实现了申请状态的实时更新和审核结果的自动通知功能。

业务逻辑模块完全完成并实现其功能。已完整实现社团管理系统的后端核心功能，包括数据库设计、基础服务和邮箱验证服务开发部署。数据库模块完成全部8张核心数据表及关联逻辑；基于Go的基础服务实现全部58个API接口，集成鉴权、文件处理和AI服务代理；Node.js开发的邮箱验证服务实现核心发件功能并通过gRPC通信；系统完成云端部署并投入测试运行。总体代码量约5500行，其中个人独立完成包括约4700行业务逻辑（含数据库操作、API实现和通信等）、约300行邮箱服务代码，约300行为protoc生成代码。此外为一部分配套技术文档，总提交行数约11,000行。

体验增强模块完全完成并实现其功能。其中核心RAG体验增强服务，使用微服务架构，搭建了FastAPI。从零开始构建了生产级的RAG服务，实现了MCP协议接口，涵盖从文档处理到智能问答的完整流程，并进行了单元测试和功能测试，使用RAGAS评估了系统效果，代码行数约2000行。离线部署Qwen3-8B-AWQ模型并使用vllm提供高效推理服务，并使用LoRA进行微调，针对社团管理下游任务场景进行专门化适配，训练代码约2000行（包含数据）。此外配套相关接口、技术文档。全部提交行数总计约8000行。核心AI代理服务架构使用微服务架构，基于FastAPI搭建了完整的AI代理服务器。从零开始构建了生产级的AI服务，实现了20+个专业化API端点，涵盖从基础聊天到智能社团管理的完整流程，并进行了全面的单元测试和功能测试。构建了混合推荐系统，结合内容推荐和大语言模型分析。实现了完整的数据处理与整合流程，支持从JSONL文件加载社团信息并进行智能匹配。离线部署Qwen3-8B-AWQ模型并使用vLLM提供高效推理服务，同时集成通义千问API提供文本摘要功能。针对社团管理下游任务场景进行了专门化适配。实现了AI驱动的训练数据生成系统，支持生成高质量的社团管理场景训练数据，包含多角色视角的指令-响应对。此外配套了详细的API文档、使用示例和技术文档。全部提交行数总计约8000+行，包含：核心服务器代码：4000+行，测试客户端：1000+行，训练数据：3000+行，配置与启动脚本：200+行，详细技术文档：800+行，系统采用模块化设计，支持高并发异步处理，具备完善的错误处理机制和CORS跨域支持，达到了生产级别的代码质量和架构标准。

项目总规模约30000行左右。

# 三、学员任务及其工作量总结

|  |  |  |
| --- | --- | --- |
| 姓名 | 职责 | 负责模块 |
| 黄凯琳 | 离线部署模型服务与LoRA微调，RAG系统构建 | 体验增强模块，8121行 |
| 刘思哲 | 服务器基架、API及网关服务器、数据库表设计、服务器及数据库云端部署 | 业务逻辑及数据库操作模块，Go+NodeJS纯代码约5000行 |
| 王子安 | 全部前端ai接口对接，部分页面展示，社团公告动态，详情，帖子，个人主页的个人信息，社团申请等 | 部分用户交互模块和ai模块，vue代码32000+行修改28000处 |
| 张佳宇 | 完成了系统页面架构的整体搭建，采用模块化设计理念，实现了路由导航、视图组件的分层管理。 | 用户交互模块框架设计和基础功能实现，与服务器接口进行对接，代码修改26000处 |
| 马之涵 | 本地LLM部署与服务器搭建 | 体验增强模块，8066行代码。 |

# 四、项目进度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目阶段 | 计划 | | 实际 | |
| 开始日期 | 结束日期 | 开始日期 | 结束日期 |
| 立项 | 6月23日 | 6月24日 | 6月23日 | 6月24日 |
| 需求 | 6月24日 | 6月25日 | 6月24日 | 6月25日 |
| 设计 | 6月25日 | 6月26日 | 6月25日 | 6月26日 |
| 编码 | 6月26日 | 7月7日 | 6月26日 | 7月7日 |
| 测试 | 7月7日 | 7月8日 | 7月7日 | 7月8日 |

# 五、经验教训及改进建议

在项目开发过程中，我们发现一个重要的疏漏：未能充分识别并纳入帖子审核和删帖的需求。这表明我们在需求分析阶段存在不足。

1. 需求分析阶段的不足：

缺乏用户场景的深入思考： 我们可能过于侧重于核心发帖和浏览功能，而没有充分模拟实际运营中可能出现的各种情况。例如，如果用户发布了违规内容，或者出现恶意灌水的情况，如何进行管理？这些都是用户场景中不可或缺的环节。运营流程的考虑不周： 在初期规划时，我们没有充分考虑到产品上线后的日常运营和内容管理需求。审核和删帖是任何UGC（用户生成内容）平台不可或缺的运营工具，旨在维护社区秩序和内容质量。沟通不足： 可能是与产品经理、运营团队或潜在用户之间的沟通不够深入，未能充分挖掘出这些隐含的需求。有时候，一些看似“非功能性”的需求，实则是维系平台健康运行的关键。风险预估不足： 没有充分预估到不加审核和管理可能带来的风险，例如内容质量下降、用户体验受损、法律合规性问题等。

2. 改进建议：

强化需求访谈和用户调研：在需求收集阶段，要更深入地与产品经理、潜在用户以及运营团队进行沟通。可以通过用户故事（User Story）或用户旅程图（User Journey Map）等方法，更全面地模拟用户和管理员在不同场景下的行为，从而发现潜在需求。开展头脑风暴会议，邀请多方参与，共同发掘可能被忽视的需求点。

引入“运营前置”思维：在项目启动初期，就应将运营团队纳入到需求讨论中。他们的经验能有效帮助我们预见和避免未来可能遇到的内容管理问题。构建最小可行性产品（MVP）时，除了核心功能，也应考虑最基本的运营和管理功能，例如一个简单的后台管理界面。

建立需求评审机制：在需求分析完成后，组织正式的需求评审会议，邀请所有相关方（包括开发、测试、产品、运营等）参与，对需求文档进行详细的审查和讨论。重点关注潜在的遗漏点、模糊之处和可能存在的风险。

通过上述改进措施，我们希望能够避免类似问题再次发生，确保在项目初期就能更全面、准确地识别和满足所有关键需求，从而构建出更加健壮和完善的产品。