****

课程设计报告书

**题目：网络应用开发**

**学 院 计算机科学与工程学院**

**专 业**

**学生姓名**

**学生学号**

**指导教师 布 社辉**

**课程编号**

**课程学分 1.0**

**提交日期 2024年5月15日**

**代码托管地址**

**部署在线网站**

**测试账号口令**

华南理工大学

**需求分析报告**

|  |
| --- |
| **概述** |
| 【基本要求】  根据选定的题目，任务清单，以及要达到的技术指标，确定满足开发需求所必须实现的功能，并选定整体技术路线，评估完成所需功能所需要的时间，指定整体开发规划。  【基本内容】  Step 1 : 根据所选择的应用领域，请认真分析业务的场景以及相关的业务流程，找出当前业务之中所存在的痛点需要解决的课题；  Step 2 : 根据所找出的痛点，提出可能的解决方案；  Step 3 :对所列出的解决方案进行可性能分析，提出解决方案；  Step 4 : 根据最优的解决方案，列出具体的实施方案以及所要达到的技术指标，并确定计划实施进度计划。 |
| **报告内容**   1. **业务场景分析与痛点识别**   在AI聊天平台，如通义千问、文心一言的应用领域中，业务主要围绕着提供智能化、自然语言驱动的交互体验展开。用户通过文本输入与AI进行交流，获取信息、解答问题或进行娱乐互动。然而，当前业务流程中存在两个显著痛点：   * 1. **个性化与定制化需求未得到满足：**尽管这些平台提供了高质量的对话体验，但用户界面和交互模式相对固定，不能根据个人偏好深度定制聊天机器人的回复风格、领域知识或专业水平。这限制了用户的个性化体验和某些专业场景的应用潜力。   2. **底层参数调优受限：**对于有高级需求的用户或开发者来说，平台隐藏了如top\_k、temperature等底层参数设置，限制了他们根据特定场景优化模型响应的能力，影响了模型的适应性和创造性输出。   3. **聊天记录管理不便：**用户难以高效回顾和管理与不同模型的聊天记录，影响了连贯对话体验和信息查找效率。  1. **可行解决方案的提出**    1. **多模型的实时切换**   **模型集成：**整合多个主流大语言模型，如通义千问、文心一言等，让用户在聊天界面轻松选择或即时切换模型。  **上下文传递**：实现模型间上下文的无缝传递，确保用户在切换模型时，对话连贯性不受影响。   * 1. **深化个性化定制功能**   **增强设置选项：**扩展个性化设置菜单，包括多维度的语气风格调控及模型回复偏好的学习。   1. **可行性分析与技术策略**    1. **技术可行性与架构设计**  * **多模型实时切换与适配器模式：**采用适配器设计模式，为不同模型设计接口适配层，统一对外提供标准化的API。这种方式可以有效封装模型差异，降低前端与后端的耦合度，实现模型的灵活切换。技术上，利用Spring Boot等框架快速构建服务架构，便于模型服务的管理和扩展。 * **深化个性化定制与动态界面：**Vue3配合Element-UI框架，不仅能够快速构建出美观且响应式的用户界面，还能够充分利用Vue3的Composition API特性，动态调整UI组件和样式，实现用户设置的即时反馈。此外，通过Vuex管理全局状态，使得用户个性化设置的同步与应用变得简单高效。 * **聊天记录与设置缓存策略：**   + **Redis缓存：**利用Redis的高性能特点，缓存聊天记录和用户个性化设置，支持快速读取和写入。特别是对于频繁查询的最近聊天记录，可以极大提高响应速度。   + **MySQL持久化存储：**作为数据持久化存储，MySQL负责存储所有的聊天记录和用户设置信息，确保数据的可靠性和完整性。通过定时任务或事务机制，保持Redis与MySQL数据的一致性。 * **单点登录**   + **单点登录：**借助Redis的会话共享机制，实现跨域的单点登录。用户认证信息在首次登录后存储于Redis中，后续访问其他子系统时直接验证Redis中的凭证，无需重复登录。 * **流量削峰策略**：   + **Redis旁路缓存**：利用Redis高速缓存特性，作为数据库访问的前置缓冲区，可以迅速响应大部分读请求，尤其是在高并发场景下，有效减轻数据库的压力，避免数据库成为瓶颈。   + **分布式缓存集群**：部署Redis集群，通过分片技术分散缓存数据，不仅可以提升缓存容量，还能增加系统的可用性和伸缩性，确保在高负载情况下仍能稳定服务。 * **缓存一致性策略**：   + **主动过期通知与备份键机制**：当Redis中的键接近过期时，可以触发一个事件通知机制，后台任务监听这些事件，及时将数据从Redis同步回MySQL数据库。同时，采用备份键策略，在主键过期前，预先复制一份数据到备份键，确保数据不因主键过期而立即丢失，实现平滑的数据迁移。   + **数据库异步写入**：对于写操作，可以先写入Redis，然后异步批量写入MySQL，减少对数据库的直接写压力，提高写入效率，同时通过消息队列保证最终一致性。 * **容灾与高可用设计**：   + **Redis集群与主备切换**：构建Redis哨兵或Cluster集群，实现主从复制和自动故障转移。当主节点发生故障时，系统能迅速切换到备用节点，确保服务不间断，防止数据丢失。   + **数据备份与恢复策略**：定期对Redis数据进行备份，并制定数据恢复计划，确保在极端情况下能够快速恢复服务，减少数据损失。  1. **具体实施方案**    1. **项目启动与需求分析阶段（1周）**    * 深入分析业务需求，细化功能模块，制定详细的功能需求文档。    1. **系统设计与架构搭建阶段（2周）**    * 设计系统架构图，确定技术栈，包括前端Vue3+Element-UI、后端Spring Boot、数据库MySQL+Redis。    * 开发模型适配层，确保各AI模型能够统一接入系统。    * 设计数据库表结构，制定缓存与持久化策略。    1. **开发与集成阶段（3-7周）**    * 前端开发：构建用户界面，实现动态设置调整、聊天记录展示等功能。使用Vuex管理状态，确保UI与数据同步。    * 后端开发：实现用户认证、多模型切换逻辑、上下文传递机制。开发缓存与数据库交互逻辑，确保数据一致性。    * 模型集成与测试：逐一集成AI模型，进行单元测试、集成测试，确保各模型与系统稳定交互。    1. **性能优化与安全加固阶段（7-9周）**    * 优化Redis与MySQL的交互逻辑，实施流量削峰策略，提升系统承载能力。 |
| **小结** |
| 本项目针对AI聊天平台，如通义千问、文心一言等，面临的三大核心痛点进行了深入分析与解决方案设计：个性化与定制化不足、底层参数调优受限、以及聊天记录管理不便。解决方案旨在通过技术创新提升用户体验和平台的灵活性，具体包括：   1. **多模型实时切换**：通过模型集成与上下文传递技术，用户能够根据需求即时切换聊天模型，享受连贯且多样的对话体验。 2. **深化个性化定制功能**：增强设置选项，使用户能够个性化定制聊天机器人的语气风格和回复偏好，提升个性化交互深度。   技术策略层面，项目采用了Spring Boot、Vue3+Element-UI等现代化技术栈，确保系统的高效开发与良好用户体验。利用Redis进行缓存管理与MySQL进行数据持久化，实现了高性能的数据存取与一致性保障。此外，通过单点登录、流量削峰、缓存一致性策略及容灾设计，提升了系统的稳定性和安全性。  实施方案分为四个阶段，从需求分析到系统设计、开发集成，再到性能优化与安全加固，步步为营，确保项目有序开展。在具体实施中，项目计划在9周内完成从基础架构搭建到性能优化的全过程，期间将重点解决前后端开发、模型集成测试、系统性能与安全加固等问题，确保最终平台能够高效、安全地服务于广大用户，满足其个性化与专业化的交互需求。通过这一系列的改进，项目有望显著提升AI聊天平台的市场竞争力和用户满意度。 |
| **指导教师评语及成绩** |
| 评语：  成绩：    指导教师签名：                                                  批阅日期： |

**系统设计报告**

|  |
| --- |
| **概述** |
| 【基本要求】  根据所必须实现的功能，设计系统整体，以及所需的模块，确定详细的开发计划。  【基本内容】  根据 《需求分析报告》所列出的具体实施方案以及计划进度， 提出整体框架设计方案；  根据整体框架设计方案，列出各个模块所需要实现的功能以及所需要达到的技术指标；  各个模块的具体实施方案；  必须要严格遵守软件工程的规范。  假设你作为PM(Project Manager)，在项目实施过程之中及时处理各种各样的问题，因此在项目开发的规划设计阶段中需要评估可能出现的各种风险。针对该项目，结合自己在完成实验时的经验请分析并评估可能出现的问题，以及针对这些问题所设定的可行的预案，并解释说明其预案的可行性。 |
| **报告内容**   1. **整体架构方案**    1. **系统架构概述**   本系统采纳前后端分离的架构设计，其中前端专注于用户交互与视觉呈现，后端则承担业务逻辑处理、数据管理及AI模型的集成重任。在技术选型上，我们利用Redis作为缓存和消息中间件，以提升响应速度和处理能力，同时MySQL作为数据持久化存储，确保数据的可靠性和完整性。后端架构遵循经典的MVC模式，并融合微服务思想，以确保系统的高度扩展性与维护便利性，为未来服务的多样化发展奠定坚实基础。   * 1. **前端设计**   前端采用Vue3与Element-UI框架，结合Vue3的Composition API，构建出既美观又响应式的用户界面，同时提升代码的模块化与复用性。Vue Router实现页面跳转逻辑管理，Vuex确保状态跨组件的统一管理与共享。Axios作为HTTP请求工具，封装网络交互，统一处理错误与异常，简化前后端通讯流程。   * 1. **后端架构**   后端架构深化为以下模块：   * 模型层(Model)：负责数据模型的定义与数据库交互，包括实体对象映射与DAO层，确保与MySQL的无缝对接。 * 视图层(View)：前端展示，后端提供JSON格式数据接口，由前端解析展示。 * 控制器层(Controller)：接收HTTP请求，执行业务逻辑，调度服务层处理，响应结果。 * 服务层(Service)：封装复杂业务逻辑，调用模型层处理数据，集成AI模型服务，通过适配器模式实现模型调用的灵活切换。 * 辅助类：引入ChatUtils、CookieUtil、HashUtil、JWTUtil等工具类，为系统提供实用功能支持，如处理聊天数据、管理Cookies、实现安全认证等。   **4. 中间件**   * Redis：   + 缓存策略：利用Redis缓存热点数据，如聊天记录，采用LRU/LFU策略自动淘汰机制优化存储。   + 会话管理：通过Redis存储用户会话信息，实现单点登录，跨域共享。   + 消息队列：采用Redis Pub/Sub 或 Stream 实现异步处理，如异步写库，提升性能。 * MySQL：   + 持久化存储：负责持久存储聊天记录、用户配置等。   + 数据一致性：利用事务确保操作原子性，通过定时任务或触发机制维持Redis与MySQL数据一致。     系统架构图 |
| **小结** |
|  |
| **指导教师评语及成绩** |
| 评语：  成绩：    指导教师签名：                                                  批阅日期： |

**系统实现报告**

|  |
| --- |
| **概述** |
| 【基本要求】  依据开发计划，进行代码实现，并在报告之中列出全部代码，解释说明部署过程步骤。  【基本内容】  对选用的开发环境以及选用的开发工具的说明；  必须要列出所有代码以及配置文件；  必须对在线部署过程进行说明。  假设你作为PM(Project Manager)，针对该项目在开发过程中所遇到的问题以及对应的解决方案，请评价在设计阶段所做的预估和预案的准确性和有效性，并总结在项目实施过程之中所获得的心得体会。 |
| **报告内容** |
| **小结** |
|  |
| **指导教师评语及成绩** |
| 评语：  成绩：    指导教师签名：                                                  批阅日期： |

**测试报告**

|  |
| --- |
| **概述** |
| 【基本要求】  需要进行黑盒测试，功能测试，将测试通过的代码共享至公共的代码托管平台，并将该网站部署上线并提供验证测试。  【基本内容】  测试方法（主要包括：黑盒测试，功能测试）；  测试的指标；  测试结果（需要提供截图）。 |
| **报告内容** |
| **小结** |
|  |
| **指导教师评语及成绩** |
| 评语：  成绩：    指导教师签名：                                                  批阅日期： |