# 系统设计文档

该项目从前端到后端使用了多种技术实现业务功能，包含前端技术和后端技术。

**前端技术栈：**

Vue.js + 生态 ，Vue 3。使用 Vue 3 的组合式 API 和 <script setup> 语法构建组件。同时模块化开发，如 components、views、layouts 等目录结构清晰分离 UI 组件和页面逻辑。使用TypeScript强类型语言，提升代码可维护性和 IDE 支持。使用Vue Router实现 SPA 内部路由跳转。使用Pinia 进行状态管理，替代 Vuex，轻量高效的状态管理库。同时自定义自定义 Composables（组合函数），用于封装复用逻辑，实现对不同微服务API的交互。

**后端技术栈：**（Spring Boot + 微服务生态）

Spring Boot快速搭建 RESTful Web 服务。自动配置机制、内嵌 Tomcat。Spring Cloud Gateway微服务网关，利用Nacos，统一处理请求转发和权限控制。

项目划分为多个独立的服务模块（comment-service, content-service, gateway-service, novel-service, user-service），每个服务专注于单一职责，易于扩展和维护。网关服务（gateway-service）负责请求路由、负载均衡、鉴权等功能。各业务服务（如 comment-service, novel-service）处理各自领域的业务逻辑。

数据访问层中使用了MyBatis -puls+ XML 映射以及结合 Spring Data JPA 提供灵活的数据访问能力。

安全方面，对于Spring Security / JWT，通过拦截器实现了对用户的认证授权。

对于不同的微服务，使用YML 配置管理，实现在多台服务器上部署不同的微服务。每台微服务使用Maven 构建工具进行依赖管理。