

浙江大学

本科实验报告

课程名称: B/S 体系软件设计

实验名称: 物联网管理系统

姓 名: 王伟杰

学 院: 计算机学院

系: 软件工程

专 业: 软件工程

学 号: 3210106034

指导教师: 胡晓军

2023 年 12 月 20 日

浙江大学实验报告

课程名称: B/S 体系软件设计 实验类型: _____
实验项目名称: 物联网管理系统
学生姓名: 王伟杰 专业: 软件工程 学号: 3210106034
同组学生姓名: 无 指导老师: 胡晓军
实验地点: _____ 实验日期: 2023 年 12 月 20 日

开发心得

在完成这次物联网设备管理网站的课程项目中,我从零开始,独立承担了整个开发过程。这对我来说不仅是一个挑战,也是一个宝贵的学习机会。以下是我在这个项目的每个阶段的心得和收获

1 系统设计阶段

1.1 技术栈的选择

前后端分离:我采用了前后端分离的方法,这是一种现代的主流 Web 开发模式,提高了系统的灵活性和可维护性。

Vue.js 作为前端框架的选择:Vue.js 以其轻量级和灵活性闻名。我选择 Vue 主要是因为它提供了响应式和组件化的解决方案,使得构建用户界面变得更加高效和简洁。它拥有一个活跃的社区和丰富的插件生态系统,这大大促进了开发流程。此外,Vue 的学习曲线相对平缓,适合快速开发和原型设计。

Koa 作为后端框架的选择：Koa 以更轻量、更富表现力的中间件架构而著称。它的简洁性和灵活性使得创建高效且易于维护的服务器应用成为可能。利用了 Node.js 的异步特性，通过 `async/await` 语法简化了错误处理和中间件的编写，提供了更好的控制流和错误管理。

MQTT 服务器与数据库的选择：我将 MQTT 服务器集成到后端中，用一个 `route` 来监听消息，极大的方便了代码开发和构建。数据库我使用 MySQL 数据库，作为一个成熟的关系数据库管理系统，它提供了我所需的所有特性，包括稳定性、灵活性和广泛的社区支持。

1.2 系统设计的挑战与调整

在系统设计阶段，我进行了深入的需求分析。这包括了用户分类、登录注册功能、个人信息管理、设备配置、设备消息记录、运动轨迹展示、用户设备统计等方面。基于这些需求，我设计了相应的数据模型和后端 API。这个过程中，我意识到最初的设计并不完美，需要在实际开发过程中进行调整。

我在开发过程中不断地回顾和修改最初的设计。这个过程教会了我，即使是最初的设计也需要不断地迭代和改进，以适应项目的实际情况。

在这个过程中，我明白了技术选型的重要性，选择合适的技术和工具对于项目的成功至关重要。这不仅影响了开发效率，也直接关系到最终产品的性能和用户体验。面对新技术，我不仅要学习它们的基础，还要学会如何将它们应用于实际项目中。这个过程提高了我的技术能力和问题解决能力。

2 开发阶段

2.1 前端

组件化开发：我充分利用了 Vue 的组件化特性，将界面分解为可重用的组件。这种方法提高了代码的可维护性，也使得功能模块更加清晰。

在组件设计时，我注重界面的用户体验和交互设计，确保每个组件不仅功能完备，而且易于操作。

状态管理：对于复杂的应用状态，我采用了 Vuex 进行状态管理。这帮助我在组件间有效地共享和管理状态，特别是在处理用户认证和设备数据时非常有用。通过 Vuex，我能够更方便地跟踪和调试应用状态，特别是在应用规模变大时，它显得尤为重要。

响应式和动态数据绑定：Vue 的响应式系统使得数据的显示和更新变得非常简单。我只需关注数据本身，Vue 会自动处理 DOM 的更新。动态数据绑定极大地减少了 DOM 操作的代码，提高了开发效率。

样式和布局：我在 Vue 项目中使用了现代 CSS 框架（如 Bootstrap 或 Tailwind CSS），这加速了响应式布局的开发，并提供了一致的界面样式。对于复杂的布局和动画，我利用了 Vue 的过渡和动画系统，增强了用户交互体验。

2.2 后端

异步处理：利用 Koa 的 `async/await` 支持，我能够以更直观和更少出错的方式编写异步代码。这在处理数据库操作和外部 API 调用时尤为重要。这种异步模式让错误处理变得更加简单，增强了代码的稳定性和可读性。

RESTful API 设计：我使用 Koa 设计和实现了 RESTful API，这些 API 为前端 Vue 应用提供了所需的数据服务。在 API 设计中，我注重其可扩展性和安全性，确保了 API 既易于维护又安全可靠。

数据库交互：我选择了适合项目需求的数据库（MySQL），并使用合适的 ORM（Sequelize）与数据库交互。确保了数据库操作的效率 and 安全性，特别是在处理用户数据和设备信息时。

3 测试阶段

测试计划与用例设计：我首先制定了详细的测试计划，确定了测试的范围和目标。这包括了对所有功能点的覆盖，如用户登录、设备配置、数据展示等。为每个功能点编写了详细的测试用例，包括正常场景和边缘情况。

手动测试与自动化测试：对于一些关键的用户界面和交互，我进行了手动测试，以确保用户体验的流畅性和直观性。我也实施了自动化测试，特别是对 API 的测试。使用了 Postman 来模拟请求和响应，确保后端逻辑的正确性。

错误处理和边缘情况：我特别关注了错误处理和边缘情况的测试，如输入验证、异常处理等。这帮助我发现并修复了可能导致系统崩溃或不正常行为的问题。通过这些测试，我增强了系统的鲁棒性和用户体验。

4 总结

通过这个项目，我巩固了软件工程的系统设计知识，还熟悉了一套技术栈，包括 Vue、Element Plus、高德地图 SDK、Node.js、Koa 和 Docker，同时对 MQTT 协议和物联网设备有了更深入的了解，对前后端分离的 B/S 设计模式也有了深入的了解，收获巨大。