

# 浙江大学

## 本科实验报告

课程名称:	B/S 体系软件设计
姓 名:	徐文皓
学 院:	计算机科学与技术学院
系:	软件工程系
专 业:	软件工程
学 号:	3210102377
指导教师:	胡晓军

2023 年 11 月 19 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： 综合

实验名称： 物联网设备管理平台

学生姓名： 徐文皓 专业： 软件工程 学号： 3210102377

同组学生姓名： / 指导老师： 胡晓军

实验地点： 线上 实验日期： 2023 年 11 月 19 日

## 物联网设备管理平台 需求规格和系统设计说明书

### 目录

1. 引言	4
1.1. 项目背景	4
1.2. 编写目的	4
1.3. 文档描述	4
1.4. 术语解释	5
2. 总体描述	6
2.1. 项目信息	6
2.1.1. 项目名称	6
2.1.2. 开发者	6
2.1.3. 用户类及其特征	6
2.2. 设计和实现上的约束	6
2.3. 假设和依赖	7
2.4. 开发计划	7
3. 需求分析	8
3.1. 功能性需求	8
3.1.1. 系统总用例图	8
3.1.2. 功能性需求	8
3.1.3. 需求优先级分析	10
3.2. 非功能性需求	10
3.2.1. 性能与并发需求	10
3.2.2. 可视化需求	10
3.2.3. 安全需求	10
3.2.4. 可维护性需求	11
3.3. 数据字典	11

<b>4. 系统设计</b>	13
4.1. 项目技术选型	13
4.1.1. 项目技术选型	13
4.1.2. 主要技术介绍	13
4.2. 系统架构设计	15
4.2.1. 系统总体架构图	15
4.2.2. 前端架构图	15
4.2.3. 后端架构图	16
4.2.4. 系统运行环境	16
4.3. 数据库设计	17
4.3.1. 数据表设计	17
4.3.2. ER 图	18
4.4. 系统接口设计	18
4.4.1. 用户相关接口	18
4.4.2. 设备相关接口	19
4.4.3. 消息相关接口	20
4.5. 运行设计	20
<b>5. 系统界面原型</b>	22
5.1. 用户注册	22
5.2. 用户登录	22
5.3. 设备总览	23
5.4. 设备配置	23
5.5. 数据统计	24
5.6. 位置信息	24
<b>6. 附录</b>	25

# 1. 引言

## 1.1. 项目背景

本项目，即物联网设备管理平台，是浙江大学 2023-2024 学年秋冬学期 B/S 体系软件设计的课程项目，旨在设计一个物联网设备的管理平台。课程提出要求，本项目需要具备用户账号信息相关功能，允许对设备进行配置、查询等操作，提供统计信息等。搭建 mqtt 服务器，Web 技术允许自行进行技术选型，要求独立完成、界面友好、提供必要的过程文档。

## 1.2. 编写目的

为了明确、细化本项目提出的具体需求，使其更加具有可行性、可操作性，同时对项目开发过程中的技术选型、底层逻辑、系统架构等做出具体规划，合理安排开发周期、跟进开发进度，编写本文档。

## 1.3. 文档描述

本文档是本项目的需求规格和系统设计说明书。

根据实际需要，我们将本文档分为引言、总体描述、需求分析、系统设计、系统界面原型和附录共 6 个部分。其中，引言部分的主要描述对象是文档，其余部分的描述对象是本项目。

引言部分对文档的情况进行了说明，介绍了项目产生的原因和开发的背景，阐述了便携本文档的目的，对文档逻辑结构进行了概述，同时说明了文档使用到的一些术语，便于本文档的潜在读者进行阅读。

总体描述部分给出了项目的名称、开发者、用户等基本特征，提出了项目在设计和实现上所受到的限制和约束，说明了项目开发所基于的假设和依赖，并对项目的总体开发流程和里程碑进行了规划。

需求分析部分从功能性需求和非功能性需求两部分细化分析了本项目的需求。功能性需求部分，给出了系统总用例图，基于用例地描述了项目的功能性需求，并且对其优先级进行了分析。为了对课程提出的要求进行适当补充、更加全面地明确项目要求，在非功能性需求部分，我们从性能和并发控制、可视化、安全性和可维护性等方面提出了非功能性需求。之后，为了便于之后数据库的设计，我们根据需求初步制定数据字典。

系统设计部分，对项目的技术选型、总体架构、数据库、系统接口和运行逻辑进行

了构思。确定技术选型后，对前端、后端及服务器相关的关键技术进行了简要介绍。考虑到系统架构设计的必要性和重要性，给出了系统总体架构图，展示了系统各模块之间的联系，并提出了前端、后端以及运行环境相关的情况。数据库是项目的重要组成部分，我们提出了对数据库的初步构想。此外，系统接口设计和运行设计也从前后端交互、用户和前端交互这两种场景对需求进行了可操作化。

系统界面原型部分给出了前期开发取得的阶段性成果，进行了系统的界面设计。

附录部分陈述了一些其他需要说明的问题。

#### 1.4. 术语解释

本项目是物联网设备管理平台，是提供给使用者查看和管理设备配置和查看设备反馈信息的系统。项目主要设计 3 个术语：用户、设备、消息。此处，就本项目中的 3 个核心术语作解释。

用户，指本系统的使用者，即对物联网设备相关信息有查询和操作需求的人。

设备，指接入本系统、允许被用户管理并且向系统反馈消息的物联网设备。

消息，指设备返回给用户，报告自身运行情况和数据的信息。

## 2. 总体描述

### 2.1. 项目信息

#### 2.1.1. 项目名称

物联网设备管理平台

#### 2.1.2. 开发者

徐文皓，3210102377

#### 2.1.3. 用户类及其特征

我们将产品的使用方定义为 1 个用户类，即对物联网设备相关信息有查询和操作需求的人，下文简称“用户”。用户类特征如下表示：

表 1 用户类及其特征

用户类	主要特征
用户	可以进行账号的注册和登录操作
	可以对个人账号信息进行管理
	可以查看设备总量、在线总量、设备分布、接收的数据量等总览信息图表
	可以对设备进行新建、删除、修改配置等操作
	可以查询设备上传的信息
	可以在地图界面查看设备信息，部分设备可以看到轨迹等
	可以在移动端友好操作

### 2.2. 设计和实现上的约束

约束条件，主要是服务器运行环境限制和开发过程中代码规范和工具版本等限制。

服务器运行环境方面，项目需要在特定的服务器运行环境下进行部署和运行，包括操作系统要求、硬件要求以及必要的软件依赖。我们需要确保服务器满足这些限制条件，以确保项目能够正常运行。

代码规范方面，为了保持代码的一致性、可读性和可维护性，需要遵守一定的代码规范，包括命名规范、代码缩进、注释规范、函数和类的设计等。遵守代码规范有助于团队成员之间的协作，并提高代码的质量和可维护性。

工具版本限制方面，我们需要使用特定版本的开发工具、框架或库来开发项目，这些版本限制来自于由于项目依赖的其他软件组件或框架的要求，或者是兼容性需求。同

时，在开发过程中，我们需要确保我们使用的数据库版本符合项目的要求。某些 SQL 语句在一定范围的数据库版本中才能执行，因此需要指定好数据库的版本。

安全和合规性方面，对于涉密信息，需要从完整性、保密性以及可用性三个特性的角度来保护用户的数据安全。

2.3. 假设和依赖

我们项目的可行性基于以下假设和依赖。

表 2 假设和依赖

假设与依赖编号	假设与依赖描述
AS-1	系统可以得到足够的服务器等外部支持
AS-2	服务器提供的数据满足系统实现相关功能的需求
AS-3	用户掌握了计算机的基本操作
AS-4	系统的迭代开发过程具有足够的精力支持
AS-5	开发者具有足够的学习和开发能力
AS-6	无不可控的突发意外情况

2.4. 开发计划

本项目的完成期限是 2023-2024 学年冬学期第 8 周。为此，我们制定如下计划：

在冬 2 周前，根据整体需求，搭建前端框架、初步完成前端静态页面开发，实现大部分系统界面原型。考虑开发过程中的理论技术限制，以在满足全部需求的前提下保障项目的可行性、可操作性为原则，细化需求，拟定设计方案，形成项目需求规格和系统设计说明书。

在冬 4 周前，完成前端开发，向后端给出接口要求。

在冬 5 周前，初步完成数据库和后端开发，实现前端所需要的接口。如果有技术上无法实现的接口，视情况调整前端逻辑。

在冬 6 周前，完成服务器搭建，完成后端开发，对项目进行集成测试。

在冬 8 周前，形成项目相关文档。

### 3. 需求分析

#### 3.1. 功能性需求

##### 3.1.1. 系统总用例图

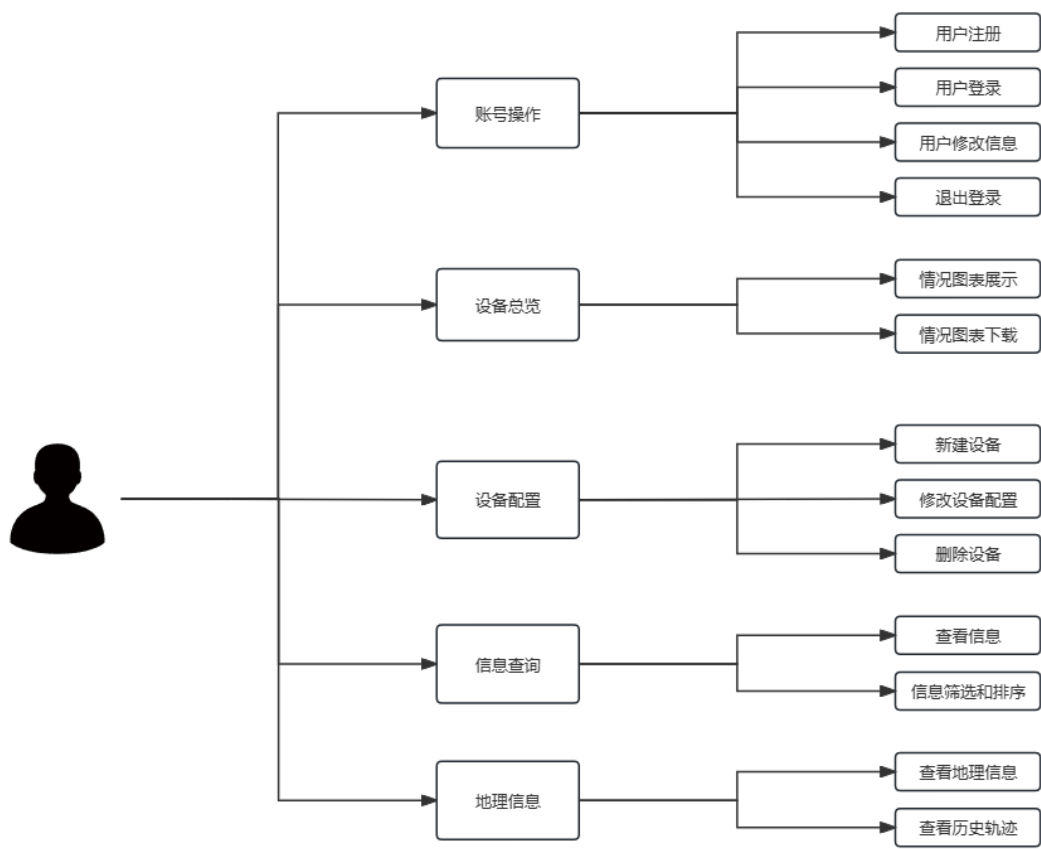


图 1 系统总用例

##### 3.1.2. 功能性需求

本项目是一个 B/S 架构的 Web 应用，对于每个用户需要实现相同的功能性需求。考虑到后续开发迭代的需要，我们在此处以较粗粒度陈列功能性需求。

###### 3.1.2.1. 用户注册

用户可以通过输入用户名、密码、电子邮箱、手机号等信息创建一个新账户。其中，用户名和电子邮箱均不可重复，用户名和密码需在 6 字节以上，用户名、密码和电子邮箱必填。系统自动进行以上规则的验证，即重复性验证、格式正确性验证和非空验证。

验证无误后，系统向数据库申请创建新账号，若创建成功则自动生成用户 ID，向用户反馈，否则反馈失败原因。



#### 3.1.2.2. 用户登录

用户可以通过输入用户名或电子邮箱作为唯一标识，并输入密码进行登录。系统进行非空验证后，将密码进行加密，向数据申请登录，若登录成功则进入主界面，否则反馈失败原因。

#### 3.1.2.3. 用户查看和修改个人信息

在主界面，用户可以将鼠标移动至用户名处，系统显示用户 ID、用户名、电子邮箱和手机号。用户可以在此处退出登录。

用户可以对电子邮箱和手机号进行修改。系统进行重复性验证、格式正确性验证和非空验证后，向数据库申请修改个人信息，若修改成功则更新显示，否则反馈失败原因。

#### 3.1.2.4. 设备总览

进入主界面后，展示设备总览信息。系统通过图表的形式展示设备总量、设备在线情况、数据接受量和设备分布情况等。允许用户对图表进行下载。

#### 3.1.2.5. 设备信息配置

展示当前该用户的设备信息，包括设备名称、设备 ID、设备类型和在线状态等。允许根据设备类型进行筛选、根据设备名称进行搜索。允许用户变更设备的在线状态、修改设备名称和设备类型、删除设备等。删除设备时，要求用户确认。系统向数据库申请修改设备名称和设备类型，修改成功则更新显示，否则反馈失败原因。系统向数据库申请删除设备，删除成功则更新显示，否则反馈失败原因。

允许用户新建设备，需要指定设备名称和设备类型。系统进行非空验证后，向数据库申请新建设备，新建成功后，自动为设备分配 ID 并更新显示，否则反馈失败原因。

#### 3.1.2.6. 设备上报信息查询

展示当前用户的所有设备的上报信息，包括信息 ID、接收时间、设备 ID、设备类型、信息等级和信息具体内容等，允许根据各项进行排序，允许用户进行筛选。

#### 3.1.2.7. 设备地理信息查询

展示地图，地图上标点显示设备信息，区分不同类型的信息。部分类型的设备数据可以展示成轨迹。

### 3.1.3. 需求优先级分析

依据风险价值四象限矩阵，对各需求用例做出以下评级：

表 3 需求优先级分析

用例编号	用例名称	风险	价值	优先级
SE-01	用户注册	中	高	中
SE-02	用户登录	中	高	高
SE-03	用户查看和修改个人信息	低	中	低
SE-04	设备总览	高	高	高
SE-05	设备信息配置	低	高	高
SE-06	设备上报信息查询	中	高	高
SE-07	设备地理信息查询	高	高	高

## 3.2. 非功能性需求

### 3.2.1. 性能与并发需求

系统应保证运行稳定，避免出现崩溃，在一周内不超过一次维护与重启。

系统应能及时检测出各种非正常情况，如与设备的通信中断断开，无法连接数据库服务器等情况，及时反馈避免用户长时间等待。

系统应实现多 Web 浏览器支持，即在各主流 Web 浏览器中均能正确显示和执行，包括 Firefox、Chrome、Edge 等。

当用户进行任何操作时，系统应该能及时进行反应，Web 响应时间在 1s 以内，涉及数据库的操作时间在 2s 以内；高峰情况下，Web 响应时间在 2s 以内，涉及数据库的操作时间在 5s 以内。

系统应能保证至少 100 人的并发访问。

### 3.2.2. 可视化需求

统计图表、设备信息、地理位置和轨迹等都能以友好、可视化的方式呈现。

### 3.2.3. 安全需求

个人账户信息应存储在安全的数据库中，并采取适当的加密措施，以保护数据的机密性和完整性。

对用户输入的数据进行严格的验证和过滤，防止恶意用户通过注入攻击来执行恶意数据库查询。

### 3.2.4. 可维护性需求

作为一个成熟的系统，在开发初期就应该充分考虑系统的可维护性。对此，我们提出以下几点要求：

高内聚、低耦合的系统模块划分。开发者需要充分考虑模块内部结构的紧密型及模块间联系的独立性。

完备、清晰、可读的文档。文档是影响软件可维护性的一个决定因素，一个好的文档应具有简明性和书写风格的一致性，从而提高系统的可读性和可修改性。设计系统时应准备好各类相关文档，方便操作人员的对功能的快速查阅及维护人员的对架构的系统掌握。交付时应文档齐全，说明详尽，且文档描述符合相关标准。

良好的编程风格。程序内部应有详细的注释和统一的编程格式，结构清晰、注释明确，使调试、测试人员能快速定位各种错误。对编程风格的具体要求如下：不使用令人捉摸不定或含糊不清的代码；使用有意义的数据名和过程名；适当的、格式正确的注释；使用模块化、结构化的设计方法；具有正确、一致和完整的文档。

严谨的单元测试。对核心模块应编写单元测试，在交互时保证各子模块和系统整体的正常运作。对可测试性的要求如下：模块化、具有良好的结构；具有可理解性、可靠性；能显示任意的中间结果；以清楚的描述方式说明系统输出，根据要求显示所有输入。

### 3.3. 数据字典

为便于数据库设计，初步给出数据字典。此处讨论数据项，对数据流等暂不展开。

表 3 数据项字典

编号	名称	类型	长度	备注
DI-01	用户 ID	整数	6	自动生成，递增
DI-02	用户名	字符串	>=6, <=64	不允许使用@字符
DI-03	密码	字符串	>=6, <=64	密文传递存储
DI-04	电子邮箱	字符串	非空, <=64	
DI-05	手机号码	字符串	11	
DI-06	设备数量	整数	非空	
DI-07	在线数量	整数	非空	
DI-08	数据接收量	整数	非空	
DI-09	设备 ID	整数	非空	自动生成，递增

DI-10	设备类型	整数	1	0,1,2,3 分别代表 4 种设备类型
DI-11	设备名称	字符串	非空, <=16	
DI-12	设备状态	布尔值	1	true, false 分别代表在线和离线
DI-13	信息 ID	整数	非空	自动生成, 递增
DI-14	信息接收时间	时间	时间	自动生成, 递增
DI-15	信息等级	布尔值	1	true, false 分别代表警告和一般
DI-16	信息内容	字符串	非空, <=256	
DI-17	经度	实数	不限	
DI-18	纬度	实数	不限	

## 4. 系统设计

### 4.1. 项目技术选型

#### 4.1.1. 项目技术选型

项目采用前后端分离的 Web 开发技术，在后端编写一系列接口供前端调用，实现前后端的协作。同时实现自己实现了一个独立于前后端的 mqtt 服务器，用于接收模拟客户端发送的数据并存储在数据库里。

前端方面，选用 Vue3 作为主要框架，npm 作为包管理工具，element plus 作为主要组件库，结合 echarts 等组件，并调用高德地图 JS API 等。

后端方面，选用 Java SpringBoot 作为主要框架，MyBatis 作为持久层框架。

数据库方面，选用 MySQL 数据库管理系统。

Mqtt 服务器方面，选用 EMQX。

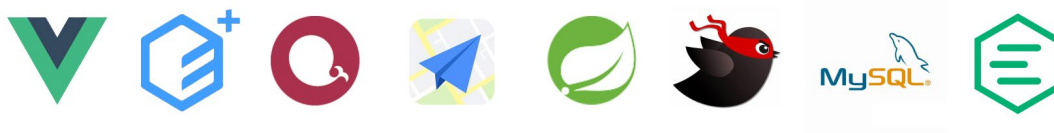


图 2 项目技术选型

#### 4.1.2. 主要技术介绍

##### 4.1.2.1. 前端技术框架

Vue3 是一款用于构建用户界面的 JavaScript 渐进式框架。它基于标准 HTML、CSS 和 JavaScript 构建，并提供了一套声明式的、组件化的编程模型，帮助开发者高效地开发用户界面。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

Element Plus 是一个基于 Vue3 的高质量 UI 组件库。它包含了丰富的组件和扩展功能，例如表格、表单、按钮、导航、通知等，让开发者能够快速构建高质量的 Web 应用。Element Plus 的设计理念是：提供开箱即用的 UI 组件和扩展功能，帮助开发者快速构建应用程序，同时提供详细的文档和教程，让开发者更好地掌握和使用 Element Plus。

ECharts 是一个使用 JavaScript 实现的开源可视化库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器，底层依赖矢量图形库 ZRender，提供直观，交互丰

富，可高度个性化定制的数据可视化图表。

地图 JS API 2.0 是高德开放平台免费提供的第四代 Web 地图渲染引擎，以 WebGL 为主要绘图手段，本着“更轻、更快、更易用”的服务原则，广泛采用了各种前沿技术，交互体验、视觉体验大幅提升，同时提供了众多新增能力和特性。

#### 4.1.2.2. 后端技术框架

Spring Boot 是一个基于 Spring 框架的快速开发框架，它使用了约定大于配置的方式，可以帮助开发人员快速搭建基于 Spring 的 Web 应用程序。Spring Boot 的特点在于自动化配置、嵌入式 Web 容器、依赖管理等方面，可以帮助开发人员轻松地构建高质量的 Web 应用程序。Spring Boot 的主要目标是简化 Spring 应用程序的配置和部署过程，同时提供更加灵活的开发方式。它使用了约定大于配置的方式，减少了开发人员需要编写的配置文件和代码量，并提供了自动化配置功能，可以根据应用程序的需求自动配置 Spring 框架中的各个组件。基于这些特点，Spring Boot 可以帮助开发人员快速搭建基于 Spring 的 Web 应用程序，同时提高开发效率和代码质量。

MyBatis 是一款用于持久层的、轻量级的半自动化 ORM 框架，封装了所有 JDBC 操作以及设置查询参数和获取结果集的操作，支持自定义 SQL、存储过程和高级映射。

#### 4.1.2.3. 数据库管理系统

MySQL 是一个关系型数据库管理系统，由瑞典 MySQLAB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 Web 应用方面，MySQL 是最好的 RDBMS (关系数据库管理系统) 应用软件之一。关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型和大型网站的开发都选择 MySQL 作为网站数据库。

#### 4.1.2.4. MQTT 服务器

EMQX 是一款开源的大规模分布式 MQTT 消息服务器，功能丰富，专为物联网和实时通信应用而设计。EMQX5.0 单集群支持 MQTT 并发连接数高达 1 亿条，单服务器的传输与处理吞吐量可达每秒百万级 MQTT 消息，同时保证毫秒级的低时延。EMQX 支持多种协议，包括 MQTT(3.1、3.1.1 和 5.0)、HTTP、QUIC 和 WebSocket 等，保证各种网络环境和硬件设备的可访问性。EMQX 还提供了全面的 SSL/TLS 功能支持，比如双向认证以及多种身份验证机制，为物联网设备和应用程序提供可靠和高效的通信基础

设施。内置基于 SQL 的规则引擎，EMQX 可以实时提取、过滤、丰富和转换物联网数据。此外，EMQX 采用了无主分布式架构，以确保高可用性和水平扩展性，并提供操作友好的用户体验和出色的可观测性。

4.2. 系统架构设计

4.2.1. 系统总体架构图

系统的总体架构如下图所示，其中客户端支持 PC 和手机等多种终端访问，服务端主要有后端服务器和 MQTT 服务器，分别单独和数据库进行交互，后端服务器可以接受网页发出的 HTTP 请求并处理,MQTT 可以接受设备发出的报文并存储在数据库中。

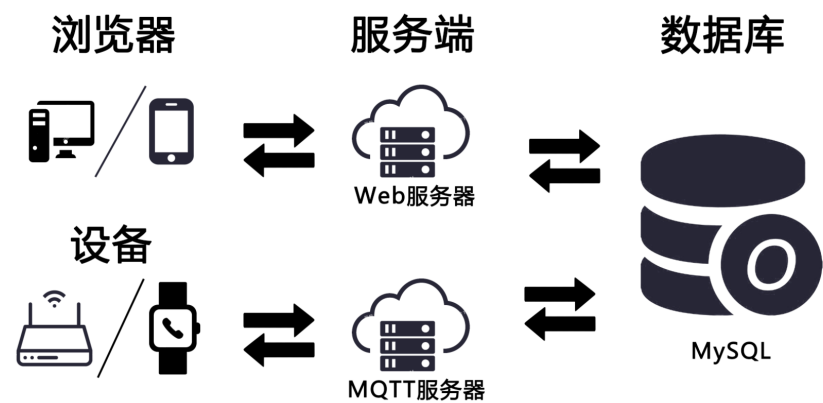


图 3 系统总体架构

4.2.2. 前端架构图

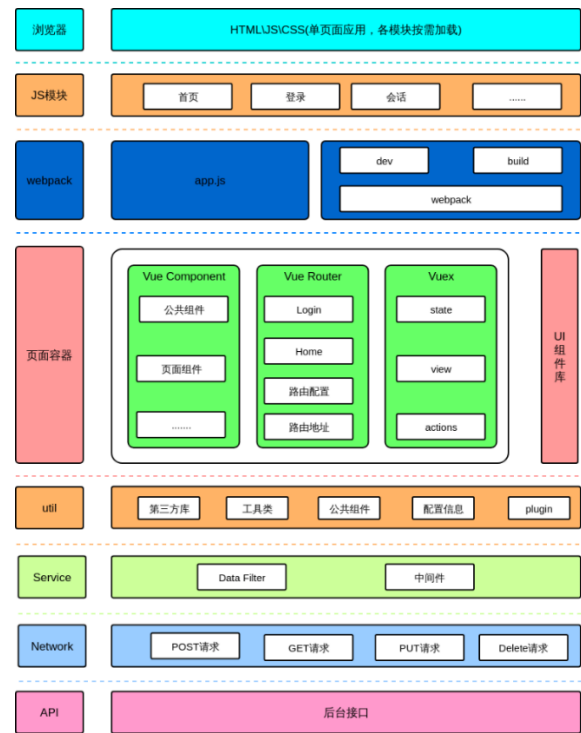


图 4 前端架构

4.2.3. 后端架构图

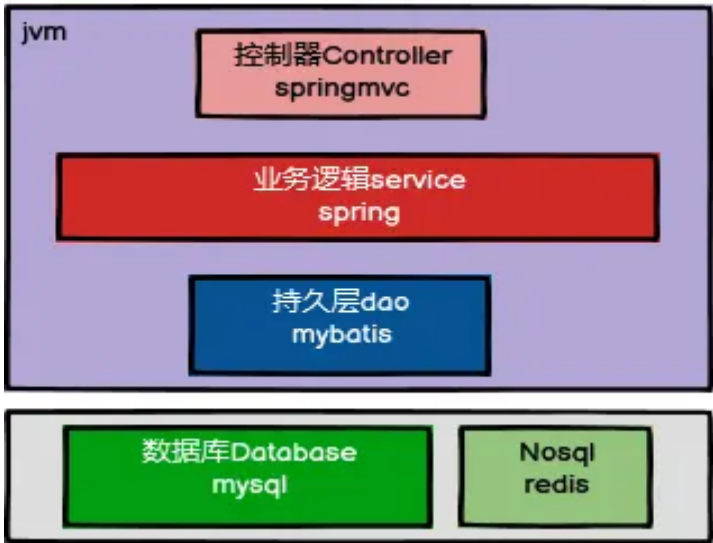


图 5 后端架构

4.2.4. 系统运行环境

4.2.4.1. 软件环境

表 3 软件环境

操作系统	Windows 11
网站服务器	Nginx
数据库服务器	Linux Socket
数据库服务器类型	MySQL
浏览器	Edge

4.2.4.2. 硬件环境

表 4 硬件环境

处理器	2 核 1.0GHz 以上
内存	2GB 以上
存储	128GB 以上
网卡	速率 1Mbps 以上



### 4.3. 数据库设计

#### 4.3.1. 数据表设计

本项目的数据库共设计了 3 个数据表，分别用于存储用户信息、设备配置信息和设备发送的消息，具体的表结构设计如下：

表 5 用户信息表

user 表			
字段名	类型	描述	备注
userid	int	用户 ID	主键，自增
username	varchar(64)	用户名	非空，唯一
password	varchar(128)	密码	非空，密文存储
email	varchar(64)	电子邮箱	非空，唯一
phonenum	char(11)	手机号码	

表 6 设备配置表

device 表			
字段名	类型	描述	备注
id	int	设备 ID	主键，自增
name	varchar(32)	设备名称	非空
category	int	设备种类	非空
userid	int	用户 ID	非空，外键
online	boolean	设备在线状态	非空

表 7 设备消息表

message 表			
字段名	类型	描述	备注
id	int	消息 ID	主键，自增
deviceid	int	设备 ID	非空，外键
time	date	产生时间	非空
type	int	告警状况	非空
info	varchar(128)	消息内容	非空
lon	numeric	经度	非空
lat	numeric	纬度	非空

### 4.3.2. ER 图

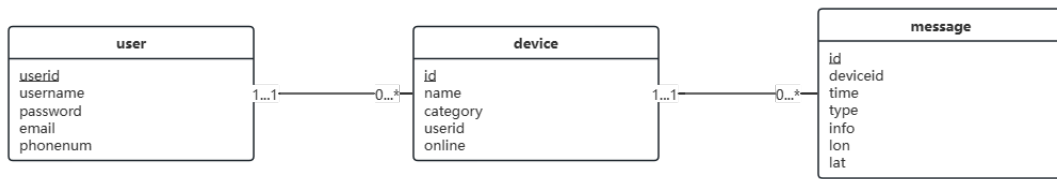


图 6 数据库实体关系

## 4.4. 系统接口设计

系统的接口主要是前后端进行交互时候的接口，这里前端采用 axios 库与后端进行异步的交互，而后端则设置若干个 Controller、Service 以及 Repository 并定义一系列路由映射和 RESTful 的增删查改 API 供前端调用，具体的 API 设计如下所示。

### 4.4.1. 用户相关接口

#### 4.4.1.1. 用户注册

URL: /user/register

主要参数: username, password, email, phonenum

返回值: status,

userid(if succeed),

message(if fail)

#### 4.4.1.2. 用户登录

URL: /user/login

主要参数: identifier, password

返回值: status,

userid & username & email & phonenum(if succeed),

message(if fail)

#### 4.4.1.3. 用户修改个人信息

URL: /user/modify

主要参数: tomodify, newvalue

返回值: status,

message(if fail)

#### 4.4.2. 设备相关接口

##### 4.4.2.1. 获取设备信息

URL: /device/get

主要参数: deviceid

返回值: status,

category &name &online(if succeed),

message(if fail)

##### 4.4.2.2. 修改设备配置信息

URL: /device/modiify

主要参数: deviceid, newname, newcategory

返回值: status,

message(if fail)

##### 4.4.2.3. 查询用户某类型设备数量

URL: /device/count

主要参数: userid

返回值: status,

count(if succeed),

message(if fail)

##### 4.4.2.4. 查询用户设备

URL: /device/have

主要参数: userid

返回值: status,

deviceid[] (if succeed),

message(if fail)

4.4.3. 消息相关接口

4.4.3.1.查询设备消息

URL: /message/get

主要参数: deviceid

返回值: status,

id &time &type &lat &lon &info(if succeed),

message(if fail)

4.4.3.2.查询设备轨迹

URL: /message/path

主要参数: deviceid

返回值: status,

location[](if succeed),

message(if fail)

4.4.3.3.查询用户消息数量

URL: /messaage/count

主要参数: userid

返回值: status,

count(if succeed),

message(if fail)

4.5. 运行设计

表 8 运行设计

操作	行为	预期结果
注册	未登录时，在登录界面点击注册按钮，跳到注册界面，输入基本信息并点击注册按钮	获得一个新账号
登录	未登录时，在登录界面输入用户名或邮箱，并输入密码，点击登录按钮	登录成功，并跳转到主界面
查看个人信息	登录后，鼠标移至用户名位置	显示基本信息

修改个人信息	在查看个人信息弹窗点击修改信息，填写新信息并确认修改	提示修改成功，显示立即变更
退出登录	在查看个人信息弹窗点击退出登录并确认	跳转到登录界面
切换页面	在导航栏选择目的页面	跳转到相应页面
查看设备总览	跳转到设备总览页面	显示统计图表
下载统计图表	在相应统计图表点击下载按钮	下载相应图表
查看设备配置	跳转到设备配置页面	显示当前用户设备配置
修改设备配置	点击对应设备修改按钮，修改后确认	提示修改成功，显示立即变更
新建设备	点击新建设备按钮，填写信息后确认	提示新建成功，显示立即变更
删除设备	点击对应设备删除按钮并确认	提示删除成功，显示立即变更
设备筛选	点击筛选框	设备按照条件被筛选，显示立即变更
设备查找	在搜索框精确输入待查找设备名称的子串	设备按照条件被查找，显示立即变更
查看数据统计	跳转到数据统计页面	显示数据统计信息
数据排序	选择排序关键字及排序类型	信息按照条件被排序，显示立即变更
查看位置信息	跳转到位置信息界面	显示地图，并根据设备信息标注地图点
查看地图点信息	点击地图点	显示设备信息
查看设备轨迹	在设备信息页面选择查看轨迹	显示该设备历史轨迹

## 5. 系统界面原型

以下给出了初期开发成果，作为系统界面原型。

### 5.1. 用户注册



The image shows a user registration interface prototype. It features a light blue rectangular form centered on a dark blue background with a network of glowing blue lines and nodes. The form has a title "用户注册" (User Registration) at the top. Below the title are five input fields, each with a label and a red asterisk: "用户名" (Username), "密码" (Password), "确认密码" (Confirm Password), "电子邮箱" (Email), and "手机号码" (Mobile Number). At the bottom of the form are two buttons: "注册" (Register) in blue and "返回" (Return) in white.

图 7 用户注册界面原型

### 5.2. 用户登录



The image shows a user login interface prototype. It features a light blue rectangular form centered on a dark blue background with a network of glowing blue lines and nodes. On the left side of the form is a green icon consisting of three circles connected by lines. To the right of the icon is the title "用户登录" (User Login). Below the title are two input fields, each with a label and a red asterisk: "用户名" (Username) and "密码" (Password). At the bottom of the form are two buttons: "登录" (Login) in blue and "用户注册" (User Registration) in white.

图 8 用户登录界面原型

5.3. 设备总览



图 9 设备总览界面原型

5.4. 设备配置

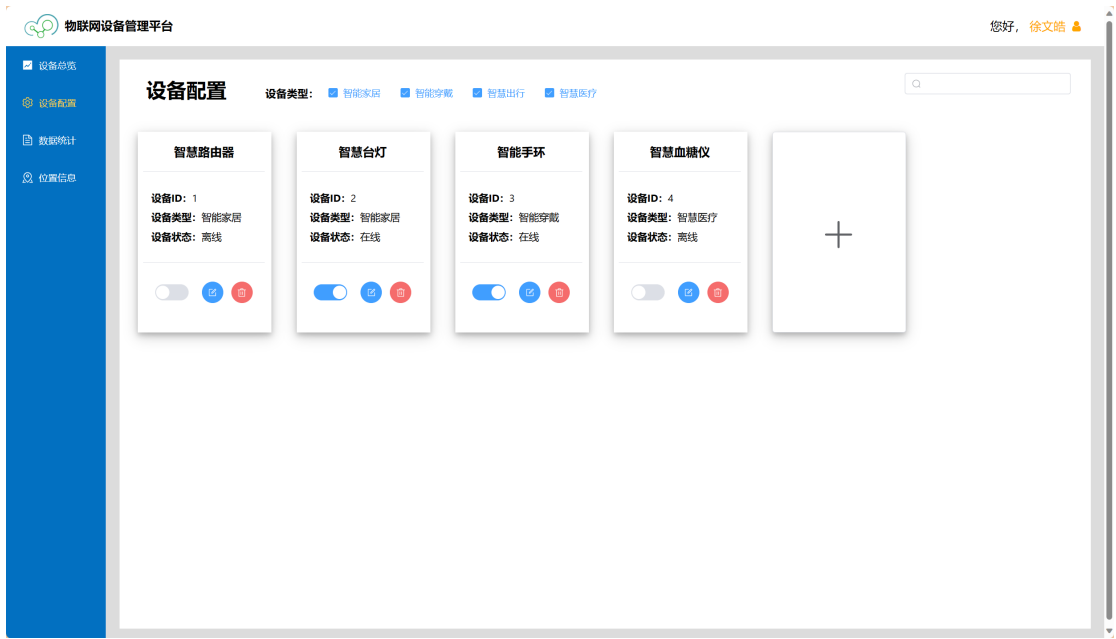


图 10 设备配置界面原型

5.5. 数据统计



图 11 数据统计界面原型

5.6. 位置信息

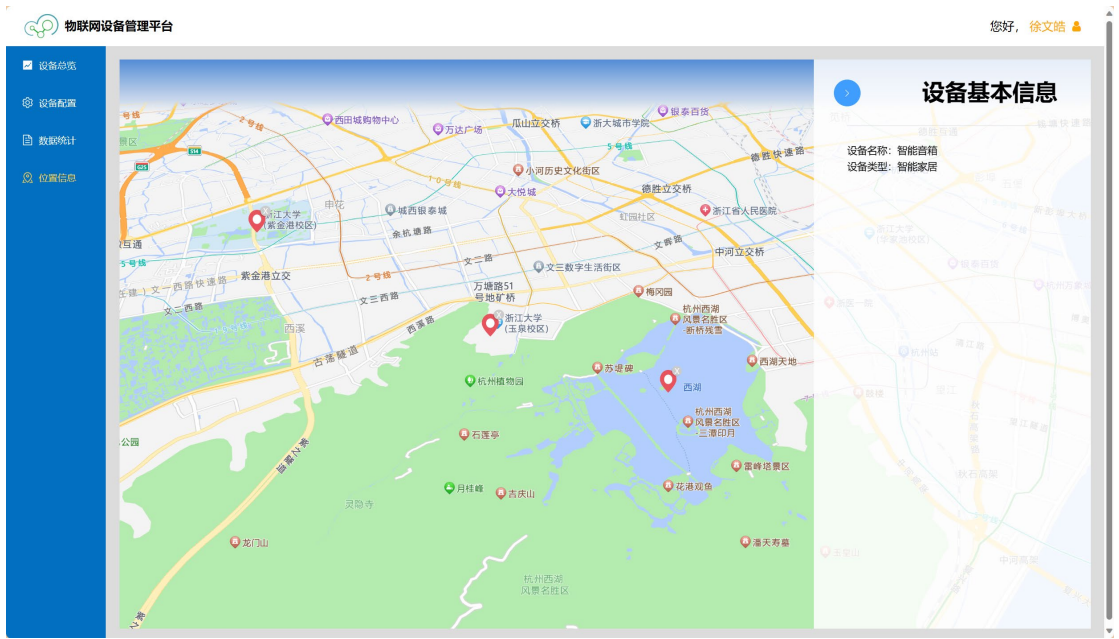


图 12 位置信息界面原型



## 6. 附录

在此说明，本需求规格和系统设计说明书仅用作规划和参考。考虑到实际情况和在实现中遇到的具体问题，项目的实现细节可能还会做出调整。文档中的一切描述及未竟之事，以最终交付的系统为准。