

浙江大学

本科实验报告

课程名称： B/S 体系软件设计

姓 名： 祝广程

学 院： 计算机科学与技术学院

系： 计算机科学与技术系

专 业： 软件工程

学 号： 3210105954

指导教师： 胡晓军

2023 年 11 月 6 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: 个人实验

实验项目名称: 物联网设备管理平台

学生姓名: 祝广程 专业: 软件工程 学号: 3210105954

同组学生姓名: 无 指导老师: 胡晓军

实验地点: 玉泉曹光彪西-503 实验日期: 2023 年 11 月 6 日

设计报告

1. 项目介绍

1.1. 项目说明

项目名称：物联网设备管理平台

任务提出者：胡晓军

开发者：祝广程

用户群：需要监控管理物联网设备的人员

1.2. 项目背景

在当今数字化时代，物联网（Internet of Things, IoT）技术的迅速发展已经在各行各业产生了深远的影响。物联网允许物理世界中的设备和传感器通过互联网连接和交互，从而实现数据的实时监测、收集和分析。为了更好地管理和利用这些物联网设备，我们需要一个强大的物联网设备管理平台。本项目是浙江大学“B/S 体系软件设计”课程的大作业，旨在帮助学生深入理解 Web 开发技术，并应用这些技术来构建一个全功能的物联网设备管理平台。

2. 需求概述

2.1. 功能需求

1. MQTT 服务器

- 搭建一个 MQTT 服务器，用于接收来自物联网终端模拟器发送的数据。
- 确保服务器能够处理大量的并发连接，以支持多个设备同时连接并发送数据。
- 实现数据的实时推送，确保数据能够即时传递到平台，以供后续处理和分析。

2. 用户管理

- 提供用户注册功能，用户需要填写必要的信息，包括用户名、密码、邮箱地址等。
- 实现用户信息的验证和保护，包括密码的安全存储和加密及邮箱地址的格式验证。
- 确保用户名和邮箱在系统中唯一，以避免冲突和重复。

3. 设备管理

- 提供设备配置界面，使用户能够创建和修改设备信息。
- 设备信息应包括设备 ID、设备名称、设备类型等基本信息。
- 确保设备 ID 的唯一性，以防止重复的设备标识符。

4. 数据查询和统计

- 提供数据查询界面，允许用户查询设备上报的数据。
- 实现数据的统计和分析功能，例如按照时间范围、设备类型、地理位置等进行数据过滤和汇总。
- 可能需要支持数据导出功能，以使用户将数据导出到其他分析工具中。

5. 地图展示

- 提供交互式地图界面，用于展示设备信息。
- 在地图上标记设备的位置，并根据设备状态进行颜色区分（正常、告警等）。
- 部分设备类型的历史数据应以轨迹的形式展示在地图上，以使用户了解设备的运动轨迹。

6. 数据可视化

- 在首页上展示统计信息，包括设备总量、在线总量、接收的数据量等。
- 使用图表（如柱状图、折线图等）以可视化的方式呈现这些统计数据，以帮助用户更好地理解平台的整体性能和趋势。

7. 移动友好性

- 确保平台的界面和布局能够适配移动设备，使用户可以在手机浏览器或内置浏览器中友好地使用平台。
- 考虑响应式设计，以确保在不同屏幕尺寸上的良好显示和交互体验。

2.2. 性能需求

1. 并发连接处理

- MQTT 服务器应该能够支持大量的并发连接，以处理多个物联网设备同时连接和发送数据。
- 最小并发连接数的要求应适应系统规模，确保平台能够处理数百到数千个设备的同时连接。

2. 实时数据处理

- 平台需要实时接收、处理和存储来自物联网设备的数据，确保数据能够实时传递到系统，以供用户访问和分析。
- 数据传输的延迟应保持在合理范围内，以确保数据的实时性。

3. 用户请求响应时间

- 平台应该迅速响应用户操作，包括登录、查询数据、配置设备等。
- 响应时间应该在用户进行操作后的短时间内完成，以提高用户体验。

4. 数据库性能

- 数据库系统需要能够高效地存储和检索大量数据。
- 查询性能应该在大规模数据集下仍然快速和高效，以支持数据的检索和分析。

5. 地图数据渲染

- 地图界面的性能应足够强大，以支持大量设备的地理位置数据的实时渲染。
- 用户在地图上浏览设备位置和状态时，界面应该保持流畅，而不会出现延迟或卡顿。

6. 数据统计和可视化

- 数据统计和可视化图表的生成应该快速完成，即使对于大规模的数据集也应该在短时间内生成。
- 用户可以交互式地探索数据并切换视图，同时保持高性能。

7. 移动设备兼容性

- 平台在移动设备上的性能应该足够好，以确保用户可以在手机和平板电脑上流畅地使用。
- 移动设备上的界面和功能应该能够适应不同屏幕尺寸和分辨率，而不会影响性能。

2.3. 安全需求

1. 身份验证和访问控制

- 用户身份应受到严格的身份验证和访问控制，以确保只有授权的用户可以访问系统。
- 系统应支持多因素身份验证（如用户名/密码和验证码、令牌等）以提高安全性。
- 用户和设备的访问权限应根据其角色和级别进行明确定义。

2. 数据加密

- 所有数据传输应使用安全的加密协议（如 TLS/SSL）来保护数据的机密性。
- 用户密码和敏感数据应在存储和传输过程中进行加密。

3. 漏洞管理

- 定期对系统进行安全漏洞扫描和评估，以及时发现并修复潜在的安全漏洞。
- 安全漏洞的修复应该迅速进行，以减少潜在的威胁。

4. 备份和灾难恢复

- 数据应定期备份，并实施可靠的灾难恢复计划，以应对数据丢失或系统中断的情况。
- 备份数据应存储在安全的位置，并定期测试以确保可用性。

3. 系统技术选型与架构设计

3.1. 项目技术选型

本项目采用前后端分离的架构，主要采用的技术栈包括：

- 开发工具：IDEA+VSCode
- 前端：Vue2+ElementUI+VueRouter+VueX+axios+百度地图 SDK+npm
- 后端：Java Springboot2+MyBatisPlus+MySQL+Maven

- 接口调试工具：Swagger+Apipost
- 身份认证：JWT

3.2. 项目主要技术介绍

3.2.1. 开发工具

- **IDEA**：一款由 JetBrains 开发的集成开发环境，主要用于 Java 开发。它提供了智能代码编辑、强大的调试功能、版本控制集成、丰富的插件生态系统和内置的构建工具等主要功能，帮助开发人员提高开发效率、管理代码质量和简化项目构建过程。
- **VSCode**：一款由微软开发的免费开源的跨平台代码编辑器。它具有智能代码编辑、强大的调试功能、丰富的扩展生态系统和集成版本控制等主要功能。开发人员可以通过 VSCode 进行代码编写、调试、版本控制和任务管理等，提高开发效率和代码质量。

3.2.2. 前端技术框架

- **Vue**：一种功能强大、灵活且易于使用的 JavaScript 框架，用于构建用户界面。它被设计为一种渐进式框架，可以逐步应用到现有项目中，也可以作为一个完整的单页应用程序开发。Vue 的核心思想是通过使用组件化的方式构建应用程序，使得代码更加模块化、可维护性更高。
- **ElementUI**：基于 Vue.js 的组件库，提供了丰富的 UI 组件和交互式的界面元素，用于快速构建现代化的 Web 应用程序。它包含了诸如按钮、表单、对话框、导航菜单、数据展示等各种常用的 UI 组件，同时还提供了一些高级组件和功能，如日期选择器、分页、表格等。ElementUI 的设计风格简洁大方，易于使用和定制，适用于各种类型的 Web 应用程序开发。
- **VueRouter**：Vue.js 官方的路由管理器，用于构建单页面应用程序。它允许开发者通过定义路由映射关系、支持嵌套路由、提供导航守卫和动态路由等功能，来管理应用程序的 URL 和页面之间的切换。VueRouter 与 Vue.js 深度集成，为开发者提供了一种优雅的方式来管理应用程序的路由，使得构建单页面应用程序变得更加简单和灵活。
- **VueX**：Vue.js 的官方状态管理库，用于管理应用程序的状态。它提供了一个集中式的状态管理方案，使得不同组件之间可以共享和访问状态。这样可以更好地管理应用程序的状态，避免了状态分散和混乱的问题，使得状态的变化和更新更加可控和可预测。
- **Axios**：一个基于 Promise 的现代化 HTTP 客户端，可在浏览器和 Node.js 环境中发送 HTTP 请求。它提供了简洁的 API 和强大的功能，包括 Promise API、拦截器、取消请求等，使得处理异步操作和发送 HTTP 请求变得更加便捷和灵活。
- **百度地图 SDK**：由百度提供的一套地图应用程序接口，用于在 Web 和移动应用程序中集成地图功能。它提供了丰富的地图展示、定位、导航、地点搜索等功能，可以帮助开发者构建具有地图展示和地理位置服务的应用程序。它支持多种平台，包括 Web、iOS 和 Android，开发者可以根据自己的需求选择合适的平台进行集成。

- **npm:** 用于 Node.js 平台的包管理工具，它允许开发者从注册表中下载并安装 JavaScript 软件包，也允许开发者将自己的包发布到注册表中。除了作为包管理工具外，npm 还提供了一些命令行工具，用于初始化项目、安装依赖、运行脚本等。

3.2.3. 后端技术框架

- **Springboot:** 一个用于构建基于 Spring 框架的快速应用程序的开源 Java 框架。它的设计目标是简化 Spring 应用程序的开发过程，通过提供约定优于配置的原则和自动化配置，减少开发者的工作量。Spring Boot 内置了多种常用的容器，如 Tomcat、Jetty 等，使得开发者可以将应用程序打包成一个可执行的 JAR 文件，方便部署和运行。此外，Spring Boot 提供了一组用于监控、管理和维护应用程序的工具，如健康检查、指标收集、配置管理等，使得应用程序的管理变得更加便捷。
- **MyBatisPlus:** 一个为简化开发而设计的 MyBatis 增强工具，提供了丰富的 CRUD 方法、内置代码生成器、Lambda 表达式查询、分页插件和逻辑删除功能等实用功能，极大地简化了数据库操作和提高了开发效率，适用于各种规模的 Java 应用程序开发。
- **MySQL:** 一种流行的开源关系型数据库管理系统，被广泛应用于 Web 应用程序开发中。它具有开源免费、跨平台性、高性能、可扩展性和丰富的社区支持等特点，使得它成为许多开发者和组织的首选数据库解决方案。MySQL 能够处理大规模的数据和高并发的访问请求，适用于各种类型的应用程序。
- **Maven:** 一个用于 Java 项目管理和构建的强大工具，它能够帮助开发者管理项目的构建、依赖、文档和报告等方面。Maven 通过提供一致的项目结构、标准化的构建流程和丰富的插件生态系统，使得项目的构建和管理变得更加简单和高效。开发者可以通过 Maven 定义项目的依赖关系、构建过程和发布流程，从而实现项目的自动化构建和管理。

3.2.4. 接口调试工具

- **Swagger:** 一个用于设计、构建和文档化 API 的开源工具。它允许开发者在编写 API 时使用标准的注解来描述 API 的各种方面，包括请求和响应的数据格式、参数、错误码等信息。通过 Swagger，开发者可以生成交互式的 API 文档，方便团队成员和其他开发者查阅和理解 API 的使用方式。
- **Apipost:** 一款专业的接口调试与管理工具，它提供了丰富的功能，包括接口测试、接口管理、Mock 数据、自动化测试等。通过 Apipost，开发者可以方便地创建和管理接口，进行接口测试和调试，并且支持团队协作和版本控制。

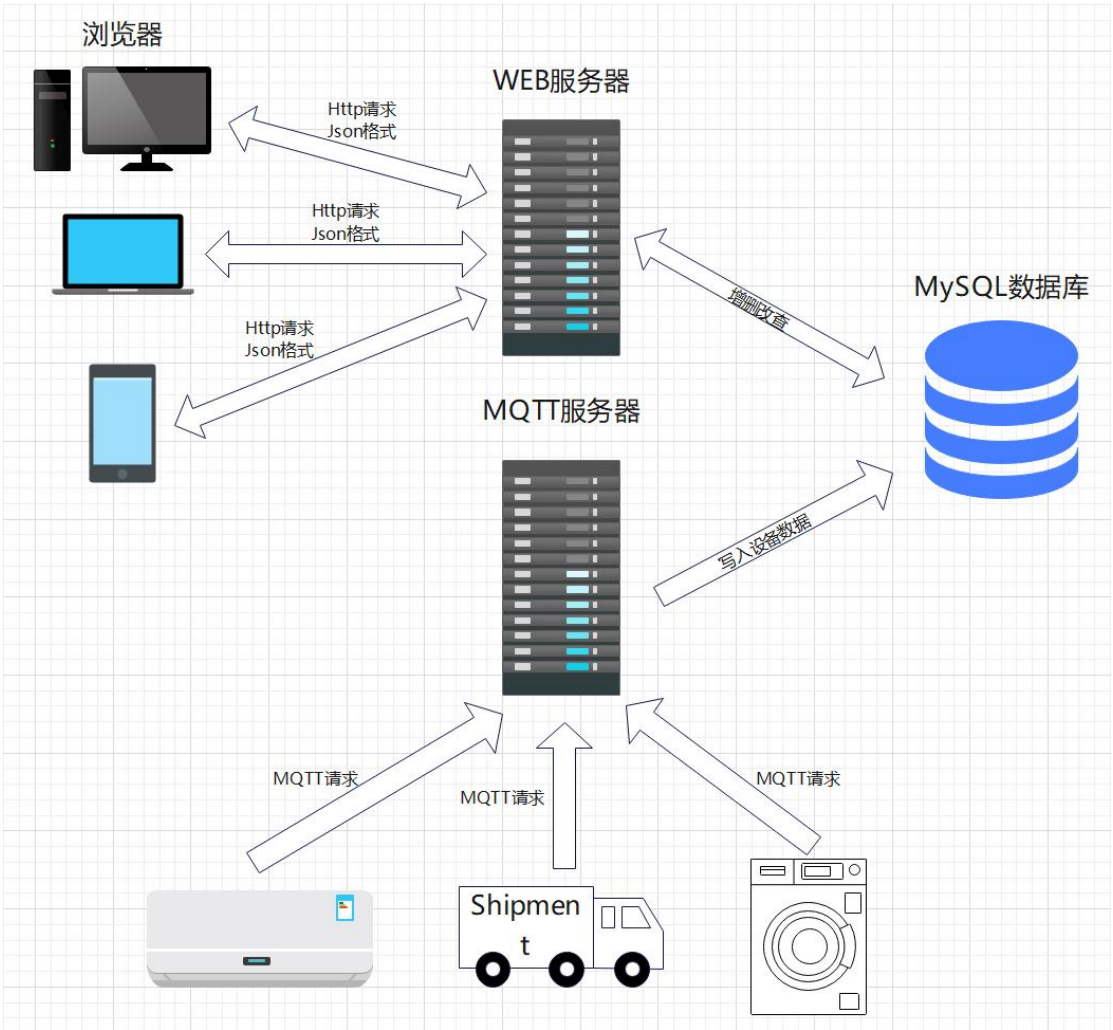
3.2.5. 身份认证

- **JWT:** 一种开放标准（RFC 7519），用于在各方之间安全地传输信息。JWT 通过使用数字签名或加密来验证信息的可靠性，可以在用户和服务器之间传递声明信息。这些声

明可以被验证和信任，因为它们是经过数字签名的。JWT 通常被用于身份验证和信息交换，特别是在分布式网络中。

3.3. 项目总体架构图

项目的总体架构图如下所示，其中浏览器前端页面的显示支持 PC 和手机等多种终端，服务器包括一个 WEB 服务器和一个 MQTT 服务器，其中 WEB 服务器用于接收前端浏览器页面发来的 Http 请求，并通过 Json 格式的数据与前端进行交互；而 MQTT 服务器用于接收指定的物联网终端模拟器发送的数据，并将其写入 MySQL 数据库中。WEB 服务器通过对数据库中物联网设备的数据记录进行增删改查的操作，实现对物联网设备的分析与控制。



3.4. 前端架构图

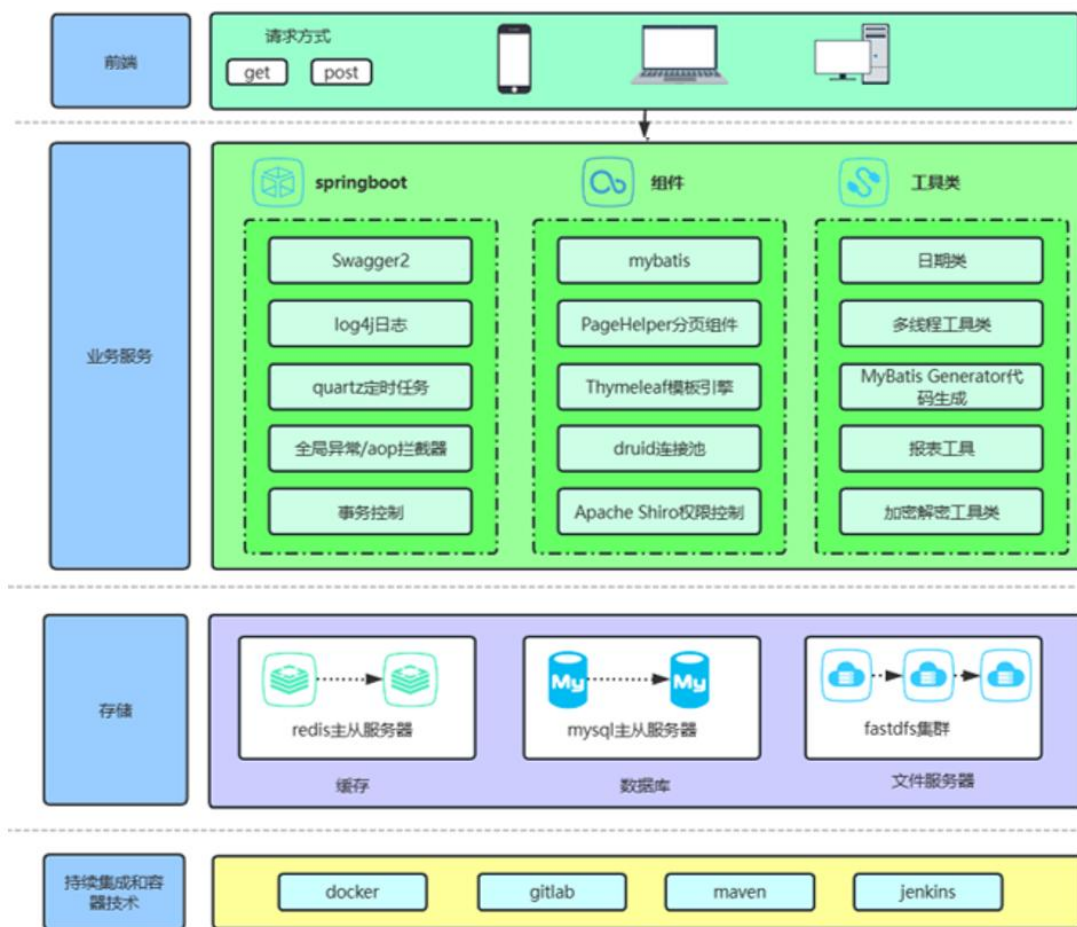
本项目前端采用的是基于 Node.js 的 Vue 框架，使用 ElementUI 组件库对前端页面进行绘制，通过 VueRouter 实现路由跳转，VueX 进行状态管理，并使用 Axios 向后端服务器发送 Http 请求，实现前后端交互。同时，本项目的前端还使

用到了百度地图 SDK 作为设备的经纬度地理位置可视化工具，并通过 npm 进行项目环境的配置和包管理。前端架构图如下所示：



3.5. 后端架构图

本项目后端采用的是针对 Java 的 Springboot 框架，通过 Json 格式的数据与前端交互并完成数据的收发，使用 MyBatisPlus 来与 MySQL 数据库交互，并简化对数据库的增删改查操作。同时，本项目的后端还使用到了 JWT 以生成和解析 token，实现对用户身份的认证，并通过 Maven 进行项目环境的配置和包管理。后端架构图如下所示：



3.6. 系统运行环境

3.6.1. 软件环境

操作系统	Windows7 及以上、Linux	
网站服务器	Nginx	1.15.8
数据库服务器类型	MySQL	8.0.31
数据库服务器	Linux Socket	
浏览器	Chrome、Edge、Firefox 等	

3.6.2. 硬件环境

操作系统	CPU: CORE i5 及以上
	内存: 2G 及以上
	硬盘: 100G 及以上
	硬盘转速: 7200rpm 及以上
应用服务器 数据库服务器 文件服务器	内存: 512M 及以上
	硬盘: 50G 及以上

通讯设备	网线：正常联通且数据传输能力良好
	网卡：100M 及以上

4. 数据库设计与 ER 图

4.1. 数据库表设计

4.1.1. user 表

用于保存用户信息

编号	属性名	字段名	数据类型	长度	备注
1	用户 ID	id	int		primary key auto increase
2	用户名	username	varchar	10	not null unique
3	密码	password	varchar	10	not null
4	邮箱地址	email	varchar	30	not null unique

4.1.2. device 表

用于保存设备信息

编号	属性名	字段名	数据类型	长度	备注
1	设备 ID	device_id	int		primary key auto increase
2	设备名称	name	varchar	20	not null unique
3	设备类型	type	varchar	20	{ 传感器设备 执行器设备 嵌入式设备 网关设备 智能终端设备 工业设备 }
4	设备状态	state	int		not null 0:关闭 1:开启 2:报警 3:严重报警

5	设备描述信息	description	varchar	256	
6	经度	longitude	double		not null
7	纬度	latitude	double		not null
8	设备所属用户的 ID	id	int		foreign key not null

4.1.3. message 表

用于保存设备上报数据的消息信息

编号	属性名	字段名	数据类型	长度	备注
1	消息 ID	msg_id	int		primary key auto increase
2	消息内容	data	varchar	256	not null
3	报警信息	alert	int		not null 0: 正常 1: 一般报警 2: 严重报警
4	时间戳	stamp	varchar	128	not null
5	发送消息的设备 ID	device_id	int		foreign key not null

4.1.4. alarm 表

用于保存设备的报警条件

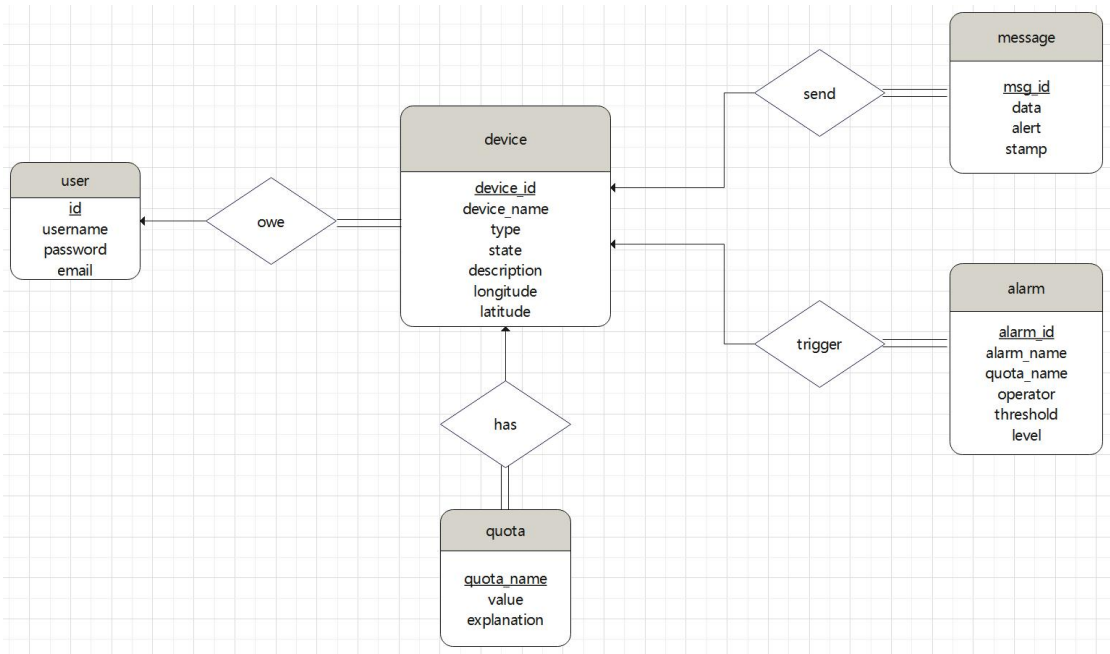
编号	属性名	字段名	数据类型	长度	备注
1	报警 ID	alarm_id	int		primary key auto increase
2	报警名称	alarm_name	varchar	20	not null
3	报警的指标名称	quota_name	varchar	10	foreign key not null
4	运算符	operator	char	1	{> 或 < 或 =}
5	阈值	threshold	double		not null
6	报警级别	level	int		0:一般报警 1:严重报警
7	该报警条件对应的设备 ID	device_id	int		foreign key not null

4.1.5. quota 表

用于保存设备的指标信息

编号	属性名	字段名	数据类型	长度	备注
1	指标名称	quota_name	varchar	10	primary key
2	该指标对应的设备 ID	device_id	int		primary key foreign key
3	指标值	value	double		not null
4	对该指标的解释	explanation	varchar	30	not null

4.2. ER 图



5. 系统接口设计

5.1. 用户登录

URL	/user/login
请求方法	POST
请求格式	{ "username": "zhangsan", "password": 123456 }
响应格式	{ "flag": true, "msg": "认证成功", }

	<pre>"code": 20000, "data": { "token": "a782741f667201b54880c923b005ff734....." } }</pre>
备注	"code" 等于 20000 表示请求成功，20001 表示请求失败；返回的 token 为校验令牌，前端会将其存在本地，此后用户的请求需要在请求头带上 token 以表示已经登录到系统当中，token 有效期为七天。

5.2. 用户注册

URL	/user/register
请求方法	POST
请求格式	<pre>{ "username": "zhangsan", "password": "123456", "email": "3301825092@qq.com" }</pre>
响应格式	<pre>{ "flag": true, "msg": "注册成功", "code": 20000 }</pre>
备注	用户在注册时还需要重新输入密码以进行确认，而第二次输入的密码会在前端与第一次输入的密码进行一致性校验，故不会通过请求传递给后端。

5.3. 查找用户名是否存在

URL	/user/find
请求方法	GET
请求格式	<pre>{ "username": "zhangsan" }</pre>
响应格式	<pre>{ "flag": true, "msg": "用户名存在", "code": 20000 }</pre>
备注	登录时，当用户输入用户名，且光标指向别的输入框时，会自动向后端发送这个请求，如果后端查询发现用户名不存在，则

	会在前端页面进行错误提示。
--	---------------

5.4. 校验用户名是否可用

URL	/user/checkUsername
请求方法	GET
请求格式	{ "username": "zhangsan" }
响应格式	{ "flag": true, "msg": "用户名可用", "code": 20000 }
备注	注册时，当用户输入用户名，且光标指向别的输入框时，会自动向后端发送这个请求，如果后端查询发现用户名重复，则会在前端页面进行错误提示。

5.5. 校验邮箱是否可用

URL	/user/checkEmail
请求方法	GET
请求格式	{ "email": "3301825092@qq.com" }
响应格式	{ "flag": true, "msg": "邮箱地址可用", "code": 20000 }
备注	注册时，当用户输入邮箱地址，且光标指向别的输入框时，会自动向后端发送这个请求，如果后端查询发现邮箱地址重复，则会在前端页面进行错误提示。

5.6. 获取用户信息

URL	/user/info
请求方法	GET
请求格式	{ "token": "a782741f667201b54880c923b005ff734....." }
响应格式	{ "flag": true,

	<pre> "msg": "用户信息获取成功", "code": 20000, "data": { { "username": "zhangsan", "password": "123456", "email": "3301825092@qq.com" } } </pre>
备注	获取用户信息需要提供登录时得到的 token 信息，以确保只有登录的用户才能获取个人信息。

5.7. 用户登出

URL	/user/logout
请求方法	POST
请求格式	{}
响应格式	<pre> { "flag": true, "msg": "登出成功", "code": 20000 } </pre>
备注	注销的同时会清除 token 信息。

5.8. 获取用户持有设备列表

URL	/device/list
请求方法	GET
请求格式	<pre> { "token": "a782741f667201b54880c923b005ff734....." } </pre>
响应格式	<pre> { "flag": true, "msg": "获取成功", "code": 20000, "data": { "devices": [{ "device_id": 1, "device_name": "lock", "type": "智能家居", </pre>

	<pre> "state": 1, "description": "用于控制门锁的开关", "longitude": 30.11, "latitude": 119.20, "alarm": { "alarm_id": 3, "alarm_name": "开门报警", "quota_name": "开关", "operator": '=', "threshold": 1, "level": 0 }, "quota": { "quota_name": "开关", "value": "1", "explanation": "门锁开关，1 表示开，0 表示关" } }, { "device_id": 2, }] } </pre>
备注	<p>获取用户持有设备列表需要提供登录时得到的 token 信息，以确保只有登录的用户才能获取相关信息，"devices"为用户所持设备的列表，其中包括设备的基本信息、指标信息和报警信息。</p>

5.9. 创建设备

URL	/device/create
请求方法	POST
请求格式	<pre> { "token": "a782741f667201b54880c923b005ff734.....", "device_name": "lock", "type": "智能家居", "state": 1, "description": "用于控制门锁的开关", "quota": { </pre>

	<pre> "quota_name": "开关", "value": "1", "explanation": "门锁开关, 1 表示开, 0 表示关" }, "alarm": { "alarm_name": "开门报警", "quota_name": "开关", "operator": '=', "threshold": 1, "level": 0 } } </pre>
响应格式	<pre> { "flag": true, "msg": "设备创建成功", "code": 20000 } </pre>
备注	用户创建设备，需要填写该设备的基本信息，包括设备名称、设备类型、设备状态和对设备的描述，同时还需要设置该设备包含的指标和对应的报警信息。

5.10. 删除设备

URL	/device/delete
请求方法	POST
请求格式	<pre> { "token": "a782741f667201b54880c923b005ff734.....", "device_id": 3 } </pre>
响应格式	<pre> { "flag": true, "msg": "设备删除成功", "code": 20000 } </pre>
备注	用户需要提供要删除的设备 ID，以完成设备的删除。

5.11. 修改设备信息

URL	/device/modify
请求方法	POST
请求格式	{

	<pre> "token": "a782741f667201b54880c923b005ff734.....", "device_id": 3, "device_name": "lock1" } </pre>
响应格式	<pre> { "flag": true, "msg": "设备信息修改成功", "code": 20000 } </pre>
备注	用户需要提供要修改信息的设备 ID 和要修改的表项, 以完成设备相应信息的修改。

5.12. 获取指定设备信息

URL	/device/find
请求方法	GET
请求格式	<pre> { "token": "a782741f667201b54880c923b005ff734.....", "device_id": 1 } </pre>
响应格式	<pre> { "flag": true, "msg": "获取成功", "code": 20000, "data": { "device_name": "lock", "type": "智能家居", "state": 1, "description": "用于控制门锁的开关", "longitude": 30.11, "latitude": 119.20, "alarm": { "alarm_id": 3, "alarm_name": "开门报警", "quota_name": "开关", "operator": '=', "threshold": 1, "level": 0 }, "quota": { "quota_name": "开关", </pre>

	<pre> "value": "1", "explanation": "门锁开关，1 表示开，0 表示关" } } } } </pre>
备注	获取指定设备信息需要提供登录时得到的 token 信息，以确保只有登录的用户才能获取相关信息，包括设备的基本信息、指标信息和报警信息。

5.13. 获取用户设备总量

URL	/device/sum
请求方法	GET
请求格式	<pre> { "token": "a782741f667201b54880c923b005ff734....." } </pre>
响应格式	<pre> { "flag": true, "msg": "用户设备总量获取成功", "code": 20000, "data": { "total_devices": 100 } } </pre>
备注	获取用户持有设备总量需要提供登录时得到的 token 信息，以确保只有登录的用户才能获取该信息

5.14. 获取用户在线设备数量

URL	/device/online
请求方法	GET
请求格式	<pre> { "token": "a782741f667201b54880c923b005ff734....." } </pre>
响应格式	<pre> { "flag": true, "msg": "用户在线设备数量获取成功", "code": 20000, "data": { "online_devices": 30 } } </pre>

	}
备注	获取用户在线设备数量需要提供登录时得到的 token 信息，以确保只有登录的用户才能获取该信息

5.15. 获取用户一般报警设备数量

URL	/device/alarm_normal
请求方法	GET
请求格式	{ "token": "a782741f667201b54880c923b005ff734....." }
响应格式	{ "flag": true, "msg": "用户一般报警设备数量获取成功", "code": 20000, "data": { "alarm_normal_devices": 5 } }
备注	获取用户一般报警设备数量需要提供登录时得到的 token 信息，以确保只有登录的用户才能获取该信息

5.16. 获取用户严重报警设备数量

URL	/device/alarm_serious
请求方法	GET
请求格式	{ "token": "a782741f667201b54880c923b005ff734....." }
响应格式	{ "flag": true, "msg": "用户严重报警设备数量获取成功", "code": 20000, "data": { "alarm_serious_devices": 2 } }
备注	获取用户严重报警设备数量需要提供登录时得到的 token 信息，以确保只有登录的用户才能获取该信息

5.17. 获取用户持有设备的上报消息列表

URL	/device/message_list
请求方法	GET
请求格式	{ "token": "a782741f667201b54880c923b005ff734....." }
响应格式	{ "flag": true, "msg": "用户持有设备上报消息列表获取成功", "code": 20000, "data": { "messages": [{ "msg_id": 20, "data": { "longitude": 30.11, "latitude": 119.20, "开关": 0 } }, { "msg_id": 23, }] } }
备注	设备上报的消息包括设备的经纬度和所有指标的数据，报警信息，以及时间戳

5.18. 查询指定设备的历史上报消息

URL	/device/message_list
请求方法	GET
请求格式	{ "token": "a782741f667201b54880c923b005ff734.....", "device_id": 3 }
响应格式	{

	<pre> "flag": true, "msg": "设备 {device_id} 历史上报消息获取成功", "code": 20000, "data": { "messages": [{ "msg_id": 20, "data": { "longitude": 30.11, "latitude": 119.20, "开关": 0 } }, { "msg_id": 23, }] } </pre>
备注	设备上报的消息包括设备的经纬度和所有指标的数据，报警信息，以及时间戳

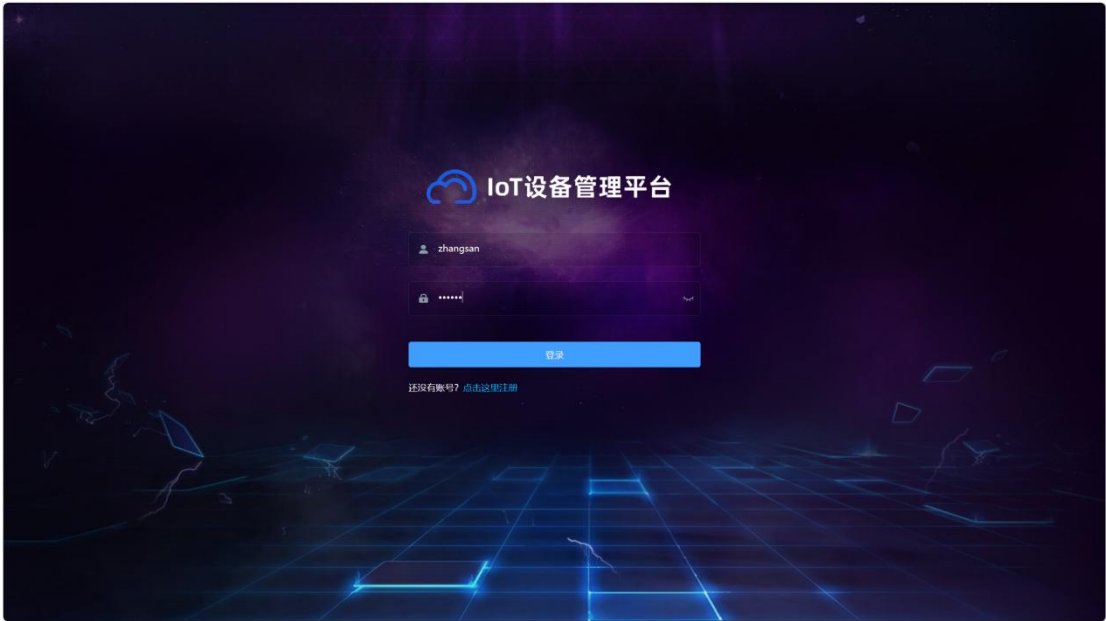
5.19. 查询设备指定指标值的变化

URL	/device/message_list
请求方法	GET
请求格式	<pre> { "token": "a782741f667201b54880c923b005ff734.....", "device_id": 3 } </pre>
响应格式	<pre> { "flag": true, "msg": "设备 {device_id} 历史上报消息获取成功", "code": 20000, "data": { "switch": [{ "value": 1, "stamp": "2023-11-15-10:59:23" }] } } </pre>

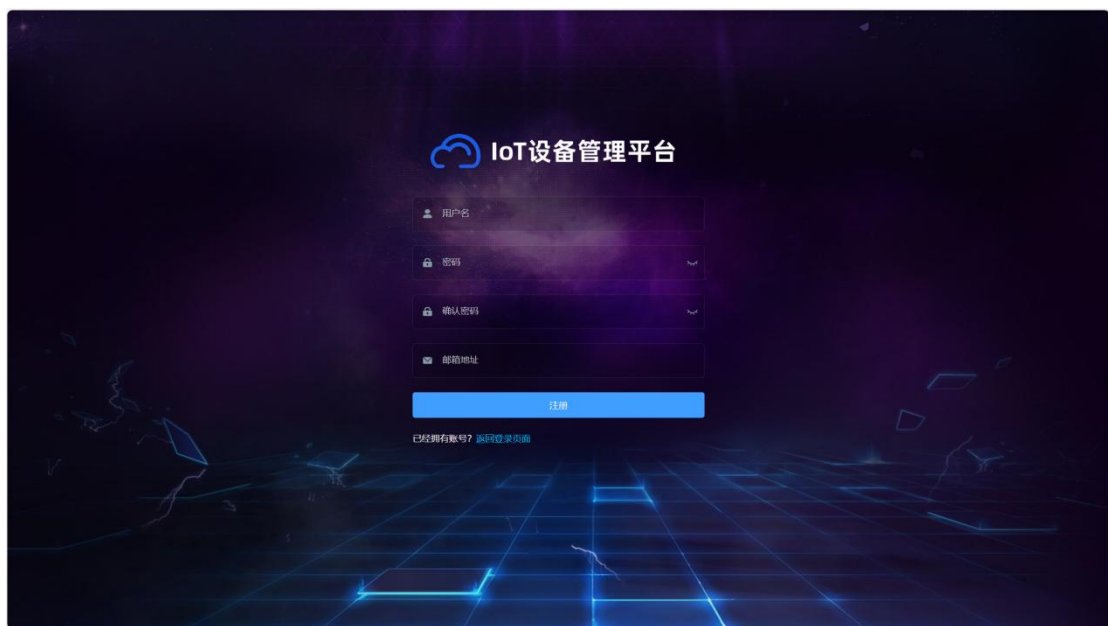
	<pre> }, { "value": 1, "stamp": "2023-11-15-11:01:23" }, { "value": 0, "stamp": "2023-11-15-11:03:23" }, ] }</pre>
备注	获取到的结果包括设备每次上传消息时指定指标的值和对应的时间戳。

6. 系统界面原型

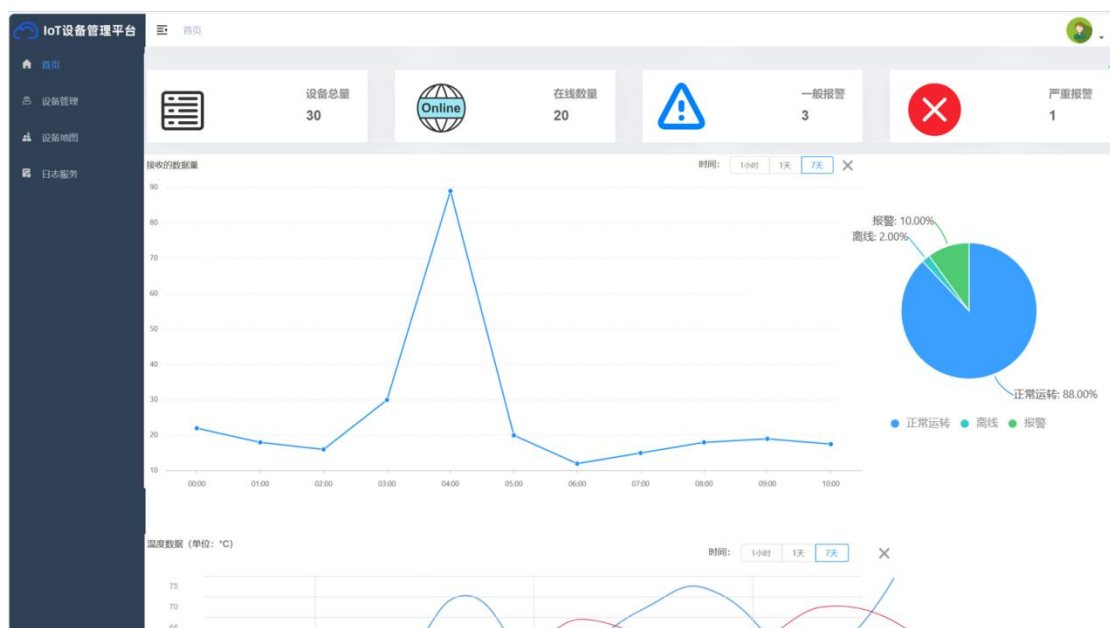
6.1. 登录界面



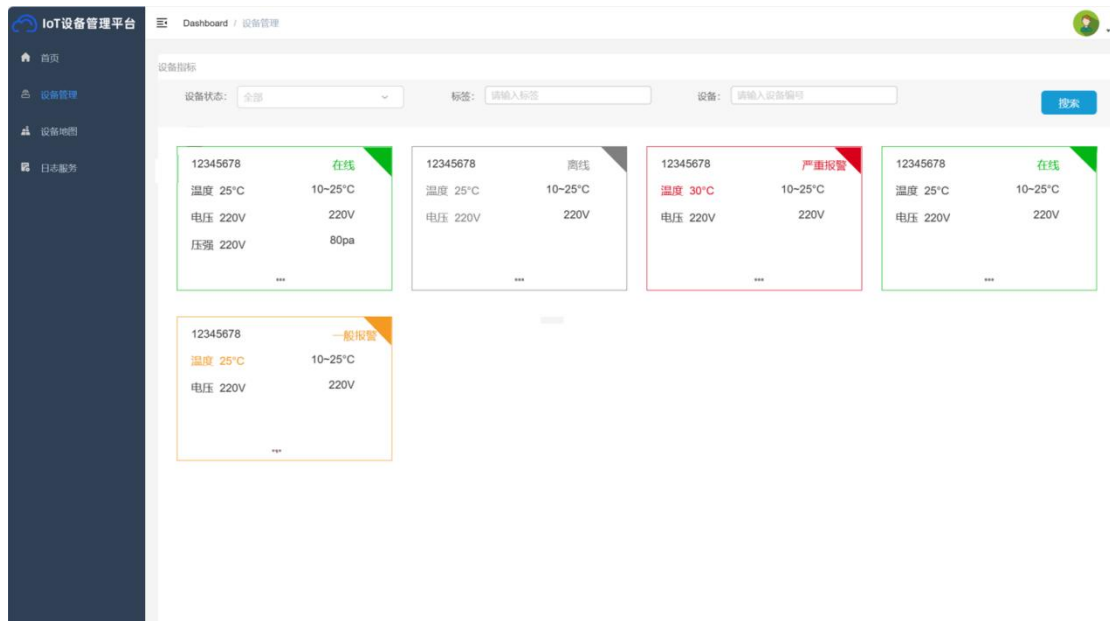
6.2. 注册界面



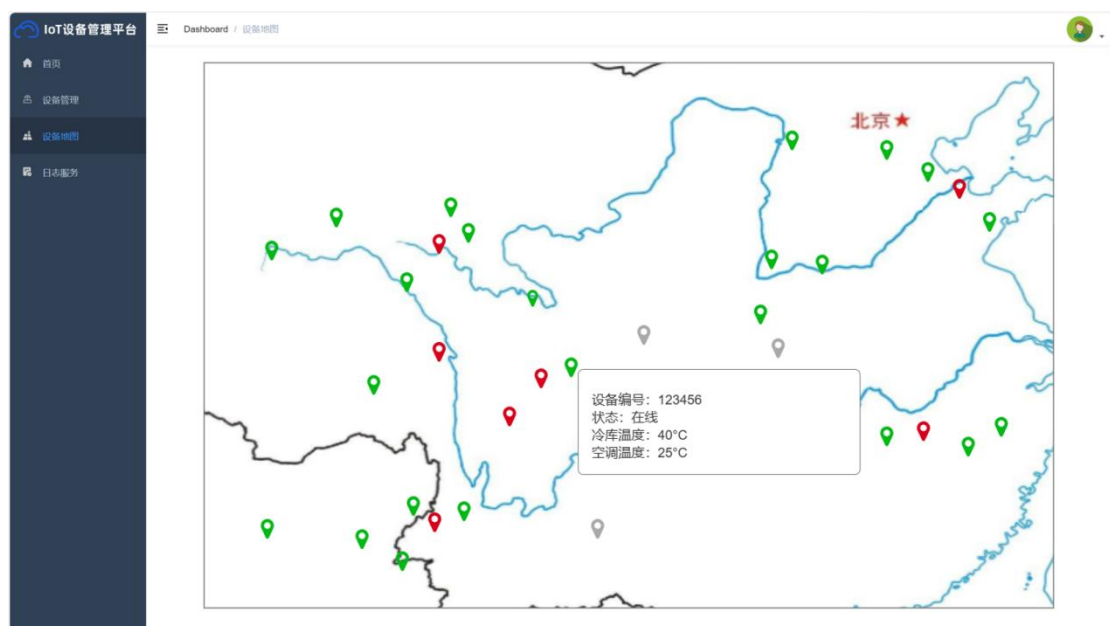
6.3. 系统首页



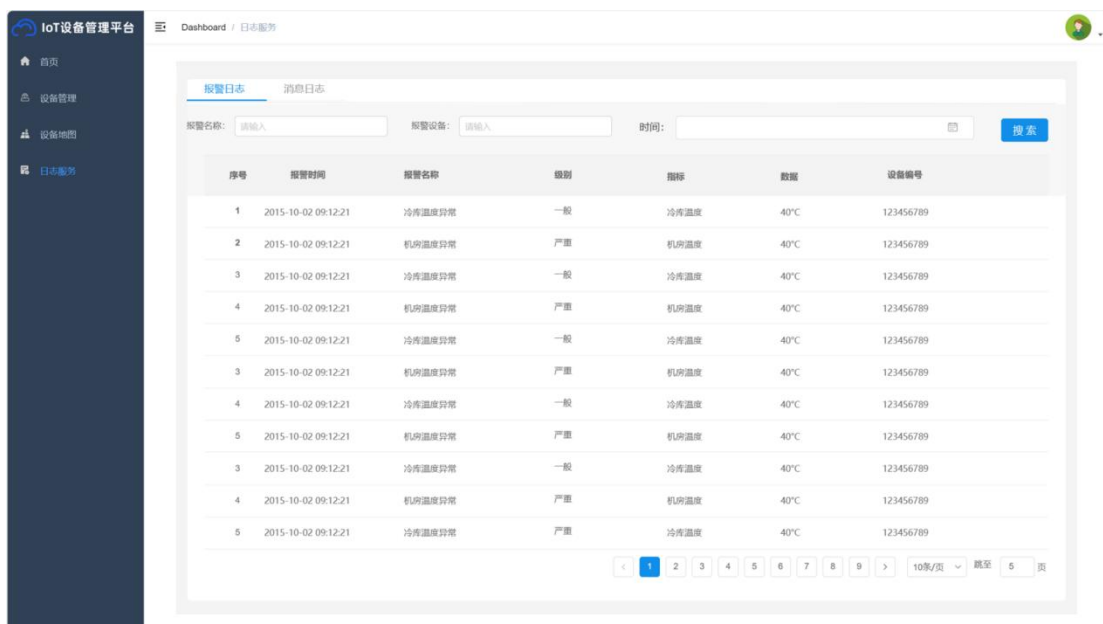
6.4. 设备管理界面



6.5. 设备地图



6.6. 日志服务界面



7. 附录

7.1. 状态码设计

状态码	解释
20000	操作成功
20001	操作失败
50008	非法 token
50012	其他客户登录
50014	token 过期
404	访问的资源不存在

7.2. 项目进度安排

时间段	计划进度
2023.11.6-2023.11.18	完成系统的总体设计和框架开发
2023.11.19-2023.12.19	完成系统各功能模块的开发
2023.12.20-2023.12.22	对系统进行集成和测试
2023.12.23-2023.12.26	完成测试报告、用户手册等文档的填写
2023.12.27	录制视频，项目验收及上传