浙江大学

本科实验报告

课程名称：BS 体系软件设计

姓　　名：夏昕

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：3210102837

指导教师：胡晓军

2023 年　　11 月　　2 日

浙江大学实验报告

实验名称：物联网应用网站开发

电子邮件地址：1564549374@qq.com 手机：18367450573

实验地点：玉泉 32 舍 209 实验日期：2023 年 11 月 2 日

目录

一、项目背景

本项目是 2023-2024 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应用网站，用户可以注册账户、修改个人信息。用户可以在登录后修改个人信息和物联网设备配置信息，并通过可视化界面查看设备信息，包括运动轨迹和设备的统计信息。该网站将提供用户友好的界面设计，并适配手机移动端，以在手机浏览器和微信等应用内置的浏览器中友好显示。

本文档是该项目的系统设计文档，包含了系统的需求分析、系统的总体架构设计、数据库的设计和系统接口、界面原型的设计等内容，并详细介绍了物联网设备应用网站的设计情况。

该项目需要包含完整的 web 前后端，实现与 mqtt 服务器的连接，以及相关项目文档等内容，并且由一人独立完成。

二、系统需求分析

2.1 功能性需求分析

本项目是一个 B/S 架构的 web 应用程序，支持用户登录、管理用户信息。在登录后，支持用户管理自己的物联网设备，可以进行新建设备、删除设备、修改设备信息、查看设备收到的信息等功能。用户还可以在地图上看到自己的设备的位置信息、轨迹信息。

本项目的功能性需求主要可以分成用户信息管理模块和设备管理模块，我们可以用以下的图进行表示：

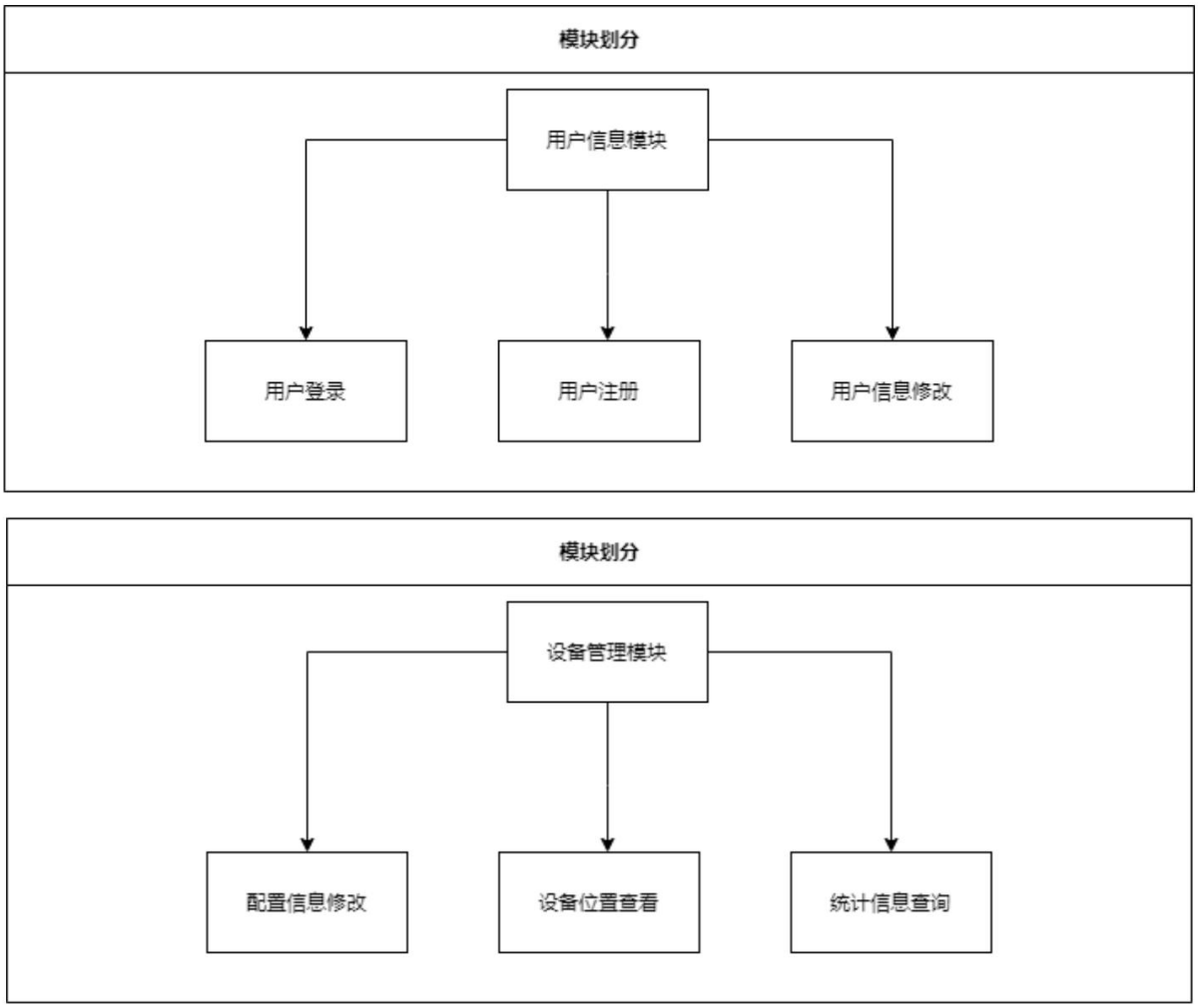

图 1 系统架构设计

该图只是我们的架构设计，涵盖了该项目的主要功能，每个主要功能下面会有其他的细分功能，如用户注册要求用户输入用户名、密码、邮箱等信息。

2.2 非功能性需求分析

本项目的非功能性需求分析主要分为以下两点：

2.2.1 性能要求：

1. 响应时间要求：系统对用户请求的响应时间应控制在可接受的范围内。例如，对于常规操作，系统应在 2 秒内响应。

2. 吞吐量要求：系统应能够处理并发请求并保持高吞吐量。例如，系统应能够每秒处理 100 个请求。

3. 并发性能要求：系统应能够同时处理多个并发请求。例如，系统应支持同时处理 100 个用户请求。

4. 稳定性要求：系统应具备高度稳定性，能够长时间运行而不发生严重故障。例如，系统应能够连续运行 30 天以上。

5. 可扩展性要求：系统应具备良好的可扩展性，能够在负载增加时保持稳定性和性能。例如，系统应能够支持每天新增 1000 个用户。

6. 资源利用要求：系统应有效利用资源，例如内存、存储和处理能力。例如，系统应在处理大量数据时能够高效利用内存资源。

7. 可靠性要求：系统应具备高度可靠性，能够在发生故障时快速恢复并保持正常运行。例如，系统应能够在 5 分钟内自动恢复正常运行。

8. 可用性要求：系统应具备高可用性，能够持续提供服务并减少停机时间。例如，系统应保证每月可用时间不低于 99.9%。

9. 可维护性要求：系统应易于维护和管理，包括日志记录、错误诊断和故障排除等功能。例如，系统应提供详细的日志记录和调试信息。

2.2.2 安全性要求：

1. 用户在登录界面输入密码时，密码应被加密，不能以明文显示，防止密码泄露。

2. 后端访问数据库时，应该使用使用参数化查询或预编译语句，使用安全的数据库访问层来访问，以避免恶意 SQL 代码的注入。

3. 使用 SSL/TLS 加密协议，确保数据在传输过程中受到保护。保护

用户与系统之间的数据传输，以防止中间人攻击。

4. 确保系统的软件更新和维护过程是安全的，不会引入漏洞。执行安全的软件更新流程，定期审查和修复潜在漏洞。

5. 实施安全审计和监测机制，以识别潜在的安全威胁和异常活动。使用安全信息和事件管理（SIEM）工具，建立日志，实时监控系统活动。

6. 实施定期数据备份和灾难恢复计划，以防止数据丢失。定期备份数据，确保备份存储在离线环境中，应对数据损坏丢失的情况。

2.2.3 输入输出需求：

1. 对用户输入的用户名/密码进行可用性和有效性的检查。

2. 首页以图表的方式展示统计信息。

3. 在地图上，设备的位置、轨迹要被清晰地展现出来。

2.2.4 可维护性需求

1. 采用模块化设计可以将软件系统划分为小而独立的模块，每个模块负责特定的功能，模块划分遵循高内聚、低耦合的原则。这样做可以提高代码的可读性和可重用性，使得项目的开发和维护变得更加容易。

2. 编写完备、清晰、可读的文档对于软件的可维护性至关重要。文档应该包括系统的功能描述、架构设计、模块之间的接口定义以及其他相关信息。文档应该简明扼要，符合相关标准，并且与实际代码保持同步，以便开发人员和维护人员能够快速理解系统的结构和功能。

3. 编程风格对于代码的可读性和可维护性有着重要影响。程序内部

应包含详细注释和统一的编程格式，结构清晰，注释明确，以便调试和测试人员能够快速定位错误。编程风格要求包括：避免使用令人困惑或模棱两可的代码；使用有意义的变量和函数名；适当且格式正确的注释；采用模块化和结构化的设计方法。

三、系统技术选型与架构设计

3.1 项目技术选型

该项目采用前后端分离的 Web 开发技术，并在后端编写一系列接口供前端调用，实现前后端的协作。以下是选用的技术：

1. 前端框架：Vue2 + Element UI2 +百度地图 SDK + Echarts

2. 后端框架：Spring Boot + mybatis + mysql5 + Maven

3. MQTT 服务器： Docker

3.2 开发技术分析

本项目使用 Vue2，Element UI2 作为前端框架，同时使用 vue-echart 制作首页统计信息图表。在前端还集成了百度地图的 SDK 来显示设备的位置。我们使用 Springboot 作为后端的框架，使用 Mybatis 技术配合 MySQL 作为后端来访问数据库。Springboot 还可以与 MQTT 服务器进行交互，实现设备数据的互通。以下是每种技术的详细介绍：

3.2.1 前端技术框架

3.2.1.1 Vue2

Vue 是一套用于构建用户界面的渐进式框架。它是一个轻量级的框架，专注于视图层的开发，可以与其他库或现有项目集成。Vue 采

用了组件化的架构，通过将用户界面划分为独立的组件，使开发者能够更好地组织和复用代码。

Vue 具有以下特点：

1. 响应式数据绑定：Vue 使用了响应式的数据绑定机制，开发者可以将数据与视图进行绑定，当数据发生变化时，视图会自动更新。这简化了开发过程，使得开发者无需手动操作 DOM 来更新视图。

2. 组件化开发：Vue 允许开发者将用户界面划分为独立组件，每个组件拥有自己的逻辑和模板。这种组件化的开发方式使得代码更加模块化、可复用，方便团队协作维护。在本项目中，一个界面就被包含在一个组件中，方便了界面的跳转。

3. 虚拟 DOM：Vue 使用虚拟 DOM 来提高性能。虚拟 DOM 是一个轻量级的 JavaScript 对象，它代表了真实 DOM 的映像。当数据发生变化时，Vue 会计算出最小的 DOM 操作，并将其应用于虚拟 DOM，然后通过高效的算法将变更更新到真实 DOM 上，从而提高了性能。

3.2.1.2 Element UI

Element UI 是一个基于 Vue.js 的开源 UI 组件库，旨在帮助开发人员快速构建现代化的 Web 界面。

Element UI 具有以下特点：

1. 提供了丰富的可复用的 UI 组件，包括按钮、表单、表格、对话框、菜单等，涵盖了常见的前端交互和展示需求。

2. 设计风格简洁、美观，符合现代化的用户界面设计趋势。Element UI

提供了灵活的主题定制和样式扩展机制，开发人员可以根据项目需求进行定制化的界面风格。

3. Element UI 的组件设计遵循了 Vue.js 的组件化开发理念，每个组件都是独立的、可复用的，可以根据需要进行组合和嵌套。通过使用 Element UI，开发人员可以快速搭建起一个功能完善、用户友好的前端界面，减少了开发时间和工作量。

4. Element UI 提供了一些高级组件和功能，如日期选择器、图标库、消息提示、弹窗等，方便开发人员在项目中添加更多的交互和功能特性。

3.2.1.3 百度地图 SDK

在 Vue 中使用百度地图 SDK 可以方便地集成百度地图的功能和服务，实现地图的展示、位置定位、导航等地理位置相关的功能。以下是百度地图 SDK 的特点：

1. 多种地图展示模式：百度地图支持多种地图展示模式，包括常见的卫星地图、交通地图、街景地图等。用户可以根据需求选择不同的地图模式进行查看。

地图标记和覆盖物：百度地图允许用户在地图上添加标记、绘制覆盖物等，方便用户进行地图标注和展示自定义信息。

2. 地图交互和事件处理：百度地图支持用户与地图进行交互，包括地图拖拽、缩放、点击等操作。同时，用户可以通过事件监听机制处理地图上的交互事件。

3. 地图 API 和 SDK 支持：百度地图提供了丰富的 API 和 SDK 支持，

开发人员可以根据自己的需求进行二次开发和定制，实现更多个性化的地图功能。

3.2.1.4 Echart

Vue-ECharts 是一个基于 Vue.js 框架的开源图表库，用于在 Vue 应用程序中创建和展示各种交互式图表。它是 ECharts（百度开源的数据可视化库）的 Vue.js 封装，为开发人员提供了在 Vue 项目中轻松集成和使用 ECharts 图表的能力。以下是 Vue-ECharts 的一些主要特点：

1. 强大的图表功能：Vue-ECharts 提供了丰富的图表类型，包括折线图、柱状图、饼图、雷达图、散点图等。开发人员可以根据需求选择合适的图表类型，展示数据和趋势。

2. 数据驱动：Vue-ECharts 采用数据驱动的方式，通过将数据传递给组件，自动生成相应的图表。开发人员可以通过更新数据来实现图表的动态更新和刷新。

3. 轻量级和易于使用：Vue-ECharts 是一个轻量级的图表库，易于学习和使用。它提供了简洁明了的 API 和组件，开发人员可以快速上手，并快速构建出各种图表。

3.2.2 后端技术框架

3.2.2.1 Spring Boot

Spring Boot 是一款基于 Spring 框架的开源应用程序开发工具，它旨在简化 Spring 应用程序的配置和开发过程。

Spring Boot 具有以下特点：

1. 简化配置：Spring Boot 采用约定优于配置的原则，通过自动配置和默认值，减少了繁琐的配置步骤。开发者可以快速启动一个 Spring 项目，并根据需要进行自定义配置。

2. 内嵌式容器：Spring Boot 内置了多个常用的 Web 容器（如 Tomcat、Jetty），使得开发者可以将应用程序打包成一个可执行的 JAR 文件，并直接运行，无需额外安装和配置外部容器。

3. 自动化依赖管理：Spring Boot 通过 Starter 模块来简化依赖管理。Starter 模块定义了一组相关的依赖，开发者只需要添加相应的 Starter 依赖，Spring Boot 就会自动引入所需的依赖库，简化了依赖管理的过程。

3.2.2.2 MyBatis

1. MyBatis 是一款优秀的持久层框架，它支持自定义 SQL、存储过程以及高级映射。MyBatis 免除了几乎所有的 JDBC 代码以及设置参数和获取结果集的工作。

2. MyBatis 可以通过简单的 XML 或注解配置和映射原始类型、接口和 Java POJO 为数据库的记录。

3.2.2.3 MySQL

MySQL 是一个开源的关系型数据库管理系统（RDBMS），它是最流行和广泛使用的数据库之一。MySQL 提供了可靠、高性能和可扩展的数据存储解决方案，被广泛应用于各种规模的应用程序和网站。

3.2.2.4 MQTT 服务器

在 Docker 中，MQTT（Message Queuing Telemetry Transport）服务器是一种基于发布/订阅模式的轻量级消息传输协议服务器。MQTT 是一种专为物联网（IoT）应用设计的协议，它具有低带宽、低功耗和小型代码库等特点，非常适合在资源受限的设备和网络环境中使用。

3.3 总体架构设计

本系统支持 PC 和移动端等多种终端访问，服务端主要分为前后端服务器和 MQTT 服务器.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。系统的总体架构如图 2 所示：

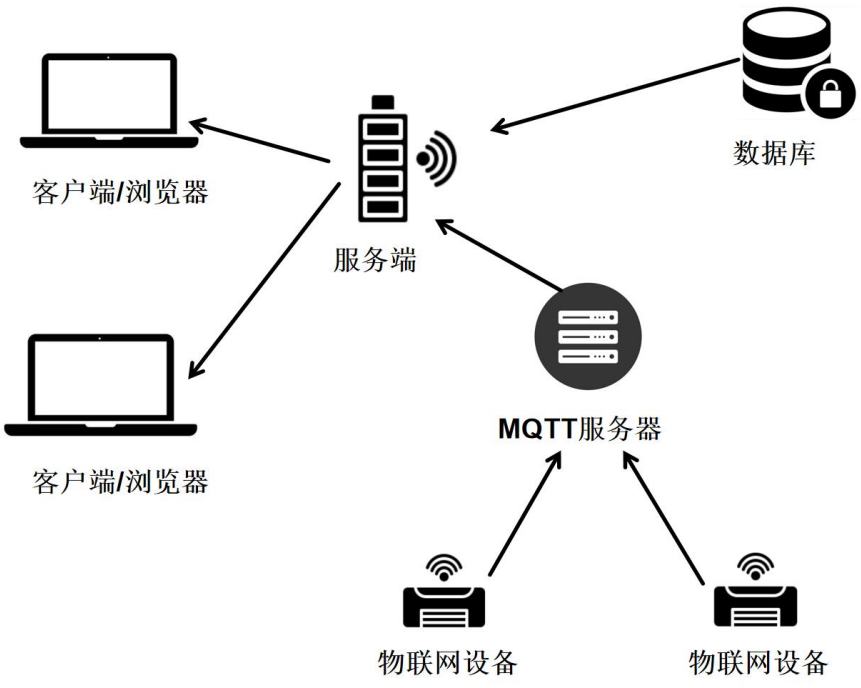

图 2 系统整体架构图

3.4 前端架构图

该项目的前端主要采用基于 Node.js 的 Vue2 框架，并采用

Element UI 作为 UI 组件框架，并辅以其他前端技术如百度地图 SDK、Echart 等。前端整体架构如图 3 所示：

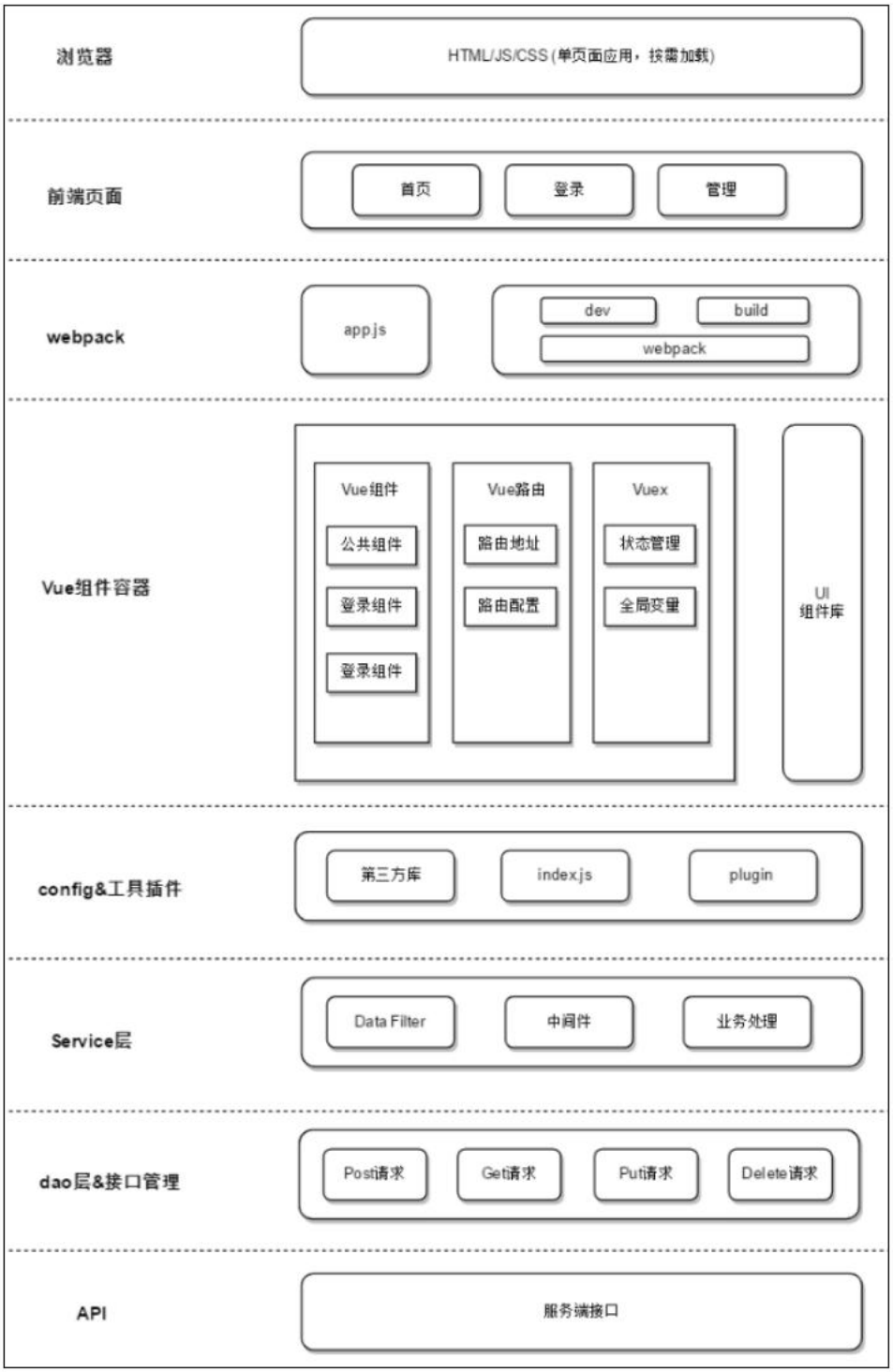

图 3 前端架构图

3.5 后端架构设计

本项目的后端主要采用 Java Spring boot 框架，使用 Maven 进行项目管理，并使用了 MyBatis 来访问数据库。后端采用 json 格式进行数据的收发，其主要的架构如图 3 所示：

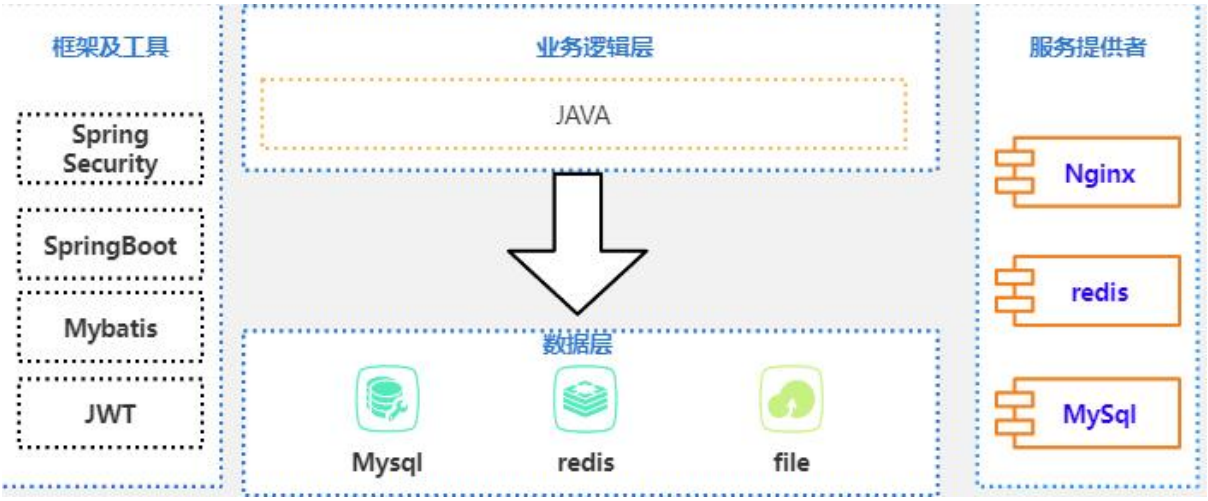

图 3 后端架构图

3.6 系统运行环境

系统运行环境包括硬件环境和软件环境。

3.6.1 硬件环境

3.6.1.1 操作系统要求：

- CPU：要求至少是 CORE i5 或更高的处理器。
- 内存：要求至少是 2GB 或更高的内存容量。
- 硬盘：要求至少是 500GB 或更高的存储容量。

3.6.1.2 应用服务器要求：

- 内存：要求至少是 512MB 或更高的内存容量。
- 硬盘：要求至少是 50GB 或更高的存储容量。

3.6.1.3 数据库服务器要求：

- 内存：要求至少是 512MB 或更高的内存容量。

• 硬盘：要求至少是 50GB 或更高的存储容量。

3.6.1.4 邮件服务器要求：

• 内存：要求至少是 512MB 或更高的内存容量。

• 硬盘：要求至少是 50GB 或更高的存储容量。

3.6.1.5 文件服务器要求：

• 内存：要求至少是 512MB 或更高的内存容量。

• 硬盘：要求至少是 50GB 或更高的存储容量。

3.6.1.6 通讯设备要求：

• 网线：要求具有良好的数据传输能力。

3.6.2 软件环境

• 操作系统: Windows 7 及以上版本、Linux

• 网站服务器: Nginx 1.15.8

•数据库服务器: 运行在 Linux 操作系统上，使用 Linux socket 进行通信

• 数据库服务器类型: MySQL 8.0

• 浏览器: Chrome

四、数据库设计与 ER 图

4.1 数据表设计

本项目使用了三张数据库表，分别用于存储用户信息、存储设备信息、存储设备收到的信息。

4.1.1 用户信息表（User）

字段名	类型	描述
id	int	用户的 id，自增
username	varchar(128)	用户名（unique）
password	varchar(128)	密码
email	varchar(128)	用户邮箱（unique）

表 1　user 表

4.1.2 设备信息表（device）

字段名	类型	描述
id	int	设备的 id，是该表的主键，自增
name	Varchar(128)	设备的名称
description	Varchar(128)	设备的描述信息
userid	int	设备所属的用户 id
kind	int	设备的类型，本项目中有五种类型，故该属性的取值范围为 1，2，3，4，5
activate	Varchar(128)	上次活跃的时间，如果现在仍活跃，其值为‘now’

表 2　device 表

4.1.3 设备发送信息表（message）

字段名	类型	描述
id	int	主键、自增
device_id	int	发送该消息的设备 id，使用 device 表的 id 作为外键
device_name	varchar(128)	发送该消息的设备名称，使用 device 表的 name 作为外键
alert	int	设备是否发出告警
info	varchar(128)	设备发送的信息
lat	Numeric（8，2）	设备发送的经度
lng	Numeric（8，2）	设备发送的纬度
stamp	timestamp	设备发送信息时的时间戳，默认值为当前的时间
value	int	设备发送的一个值

表 3　message 表

4.2 ER 图

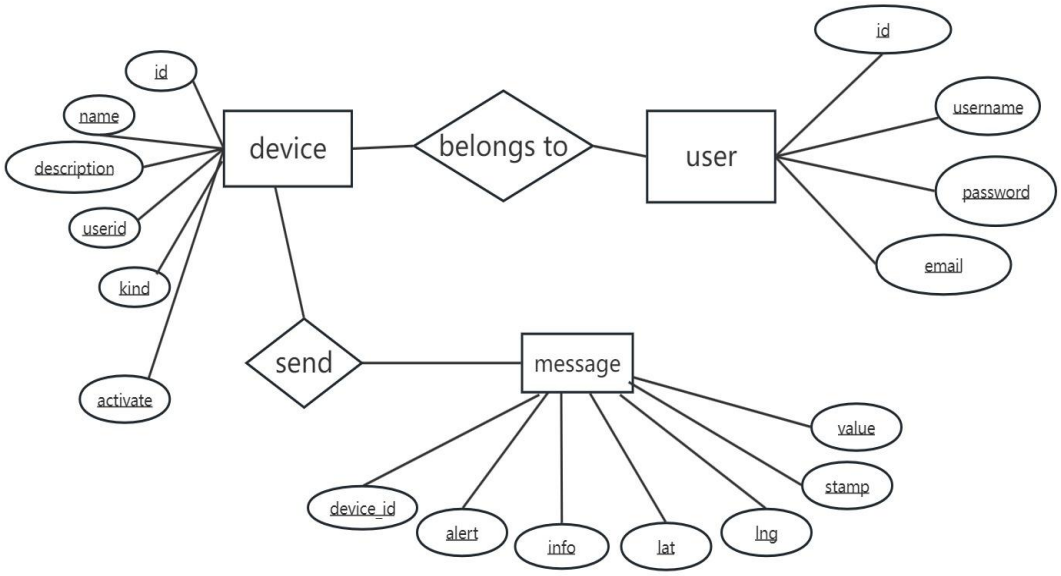

图 4　ER 图

五、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 的 api 供前端调用，具体的 api 设计如下所示。

由于当前系统仍处于开发阶段，最终成品的 api 的定义和数目可能和本设计文档中的接口定义略有差异。请以实物为准。

5.1 用户信息接口

5.1.1 注册接口

接口 URL	/register
传入参数	username, password, email, 传入的是注册用户的用户名、密码、邮箱

返回值类型	返回一个 int 型的参数，1 代表注册成功，0 代表注册失败
接口简介	用于网站的登录认证，在登陆成功后会将用户数据存入数据库表

表 4　register 接口

5.1.2 重置密码

接口 URL	/reset
传入参数	username, password, email, 传入的是用户的用户名、密码、邮箱。需要用户同时提供用户名、密码以验证身份
返回值类型	返回一个 int 型的参数，1 代表修改成功，0 代表修改失败
接口简介	用于用户的修改密码，修改成功后，数据库表中的用户密码会得到更新

表 5　reset 接口

5.1.3 登录

接口 URL	/login
传入参数	username, password, 传入的是用户的用户名(或邮箱)、密码。这里传入的参数 username 可以是用户的用户名或邮箱。本系统支持用户使用密码或者邮箱登录
返回值类型	返回 Pair<Integer, Pair<String, Integer>>。第一个 Integer 代表登录状态，0 代表登录失败，1 代表成功。后面的 pair 中，string 返回了用户名，integer 返回了用户在 user 表中的 id。后面的两个变量会保存在前端，作为用户的信息
接口简介	用于用户的登录。登录成功后将用户的用户名、id 保存在前端

表 6　login 接口

5.1.4 修改用户名

接口 URL	/resetUsername
传入参数	String oldname, String newName。oldname 代表原来的用户名，用于检索数据库中的对应条目。

	newName 是修改后的用户名。
返回值类型	返回一个 int 型的参数，1 代表修改成功，0 代表修改失败
接口简介	用于用户的修改用户名，修改成功后，数据库表中的用户名会得到更新。同时前端保存的用户名也会同步被修改

表 7　resetUsername 接口

5.1.5 修改用户邮箱

接口 URL	/resetemail
传入参数	String oldname, String newemail。oldname 代表原来的用户名，用于检索数据库中的对应条目。newemail 是修改后的用户邮箱。
返回值类型	返回一个 int 型的参数，1 代表修改成功，0 代表修改失败
接口简介	用于用户的修改用户邮箱，修改成功后，数据库表中的用户邮箱会得到更新

表 8　resetemail 接口

5.2 设备/消息接口

5.2.1 获取当前用户的所有设备

接口 URL	/getDevice
传入参数	Integer id。本接口需要传入用户的 id 作为后端查询数据库的依据
返回值类型	返回一个 List<Device>型的参数。Device 是自定义数据类，每个类对象包含了一条 device 表中的 entry。
接口简介	用于从数据库获取用户的 所有设备信息。该接口略经修改，可以更新成获取特定用户设备信息、获取用户

表 9　getDevice 接口

5.2.2 获取某个设备发送的所有信息

接口 URL	/getMessage
传入参数	Integer userid, Integer device_id。本接口需要传入用户的 id 和需要查询的设备 id，首先进行设备 id 与用户 id 的匹配，然后使用设备 id 作为后端查询数据库的依据。

返回值类型	返回 Pair<Integer,List<Message>>型的参数。第一个 Integer 代表是否匹配成功，当该参数为 1 时再从数据库查询数据，并将数据保存在第二个参数中
接口简介	用于从数据库获取某设备的所有信息，并要求用户 id 与设备 id 相匹配

表 10　getMessage 接口

5.3 设备修改接口

5.3.1 修改设备

接口 URL	/editDevice
传入参数	Integer device_id, String device_type, String device_name。本接口需要传入需要修改的设备 ID，希望设备修改的类型和设备名。
返回值类型	返回 Integer 型的参数。返回值为 1 代表修改成功，否则修改失败。
接口简介	用于修改用户指定设备的信息。

表 11　editDevice 接口

5.3.2 增加设备

接口 URL	/Adder
传入参数	Integer user_id, String device_name, String device_type, String device_des。本接口需要传入需要当前用户的 ID，设置的设备名，设备类型和设备描述。这些参数都是非空的。
返回值类型	返回 Integer 型的参数。返回值为 1 代表添加成功，否则添加失败。
接口简介	用于新建设备。

表 12　Adder 接口

5.3.1 修改设备

接口 URL	/delDevice
传入参数	Integer device_id。本接口需要传入需要删除的设备 ID,。
返回值类型	返回 Integer 型的参数。返回值为 1 代表删除成功，否则删除失败。
接口简介	用于删除用户的指定设备。

表 13　delDevice 接口

六、 系统界面原型

系统的界面原型设计如下所示，最终成品和界面原型略有差异,一切效果请以最终实物为准。

6.1 登录界面

图 4 登录界面

6.2 主页

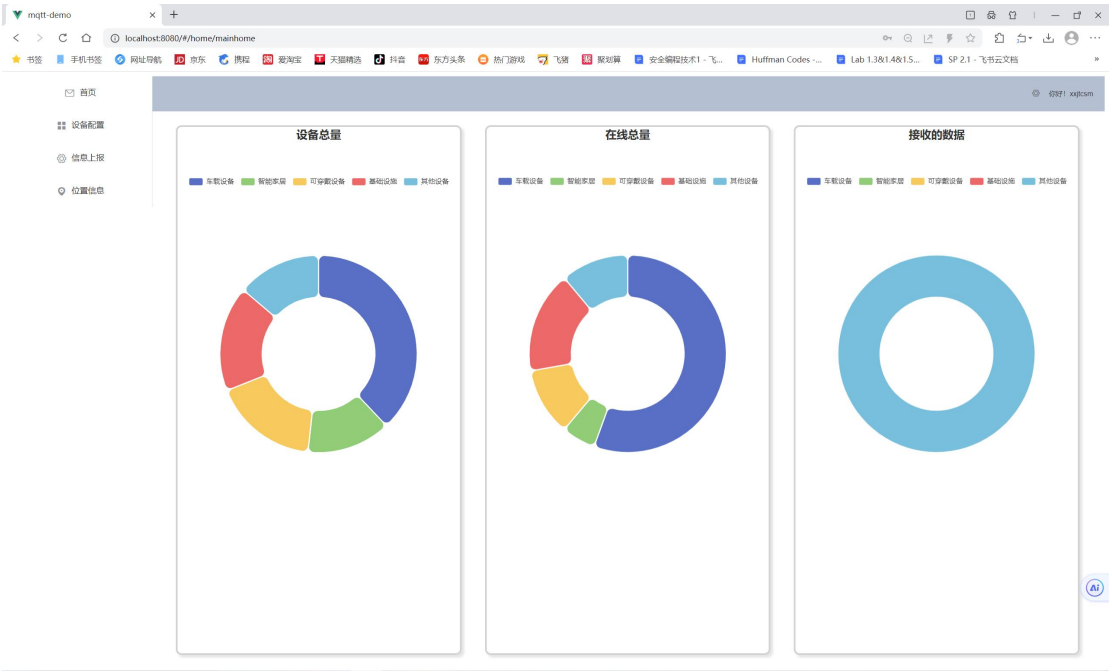

图 5 系统主页

6.3 设备配置界面

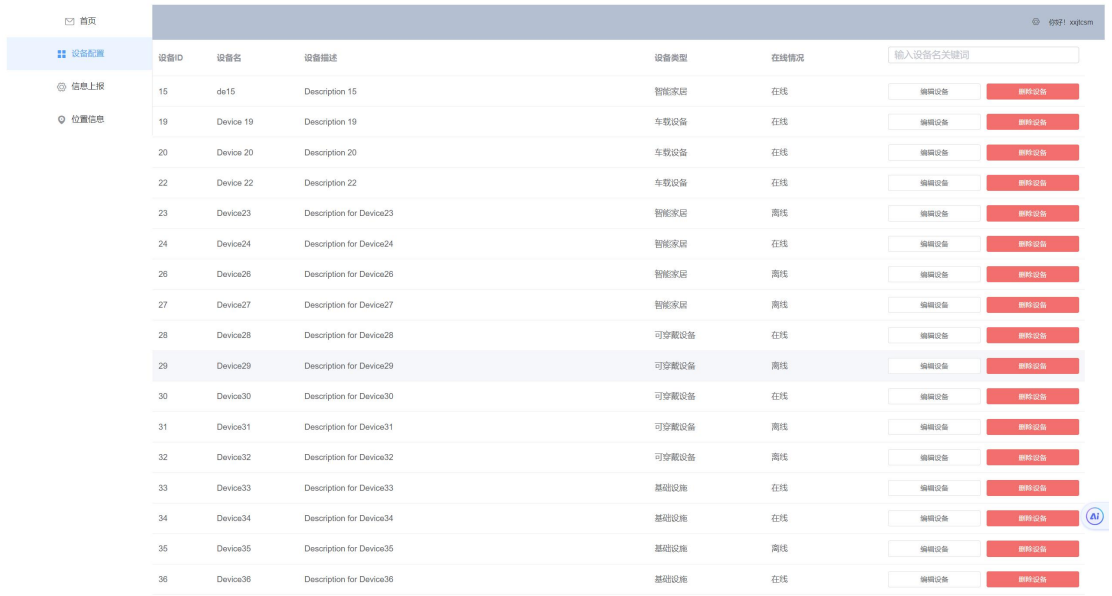

图 6 设备配置界面

6.4 信息上报界面

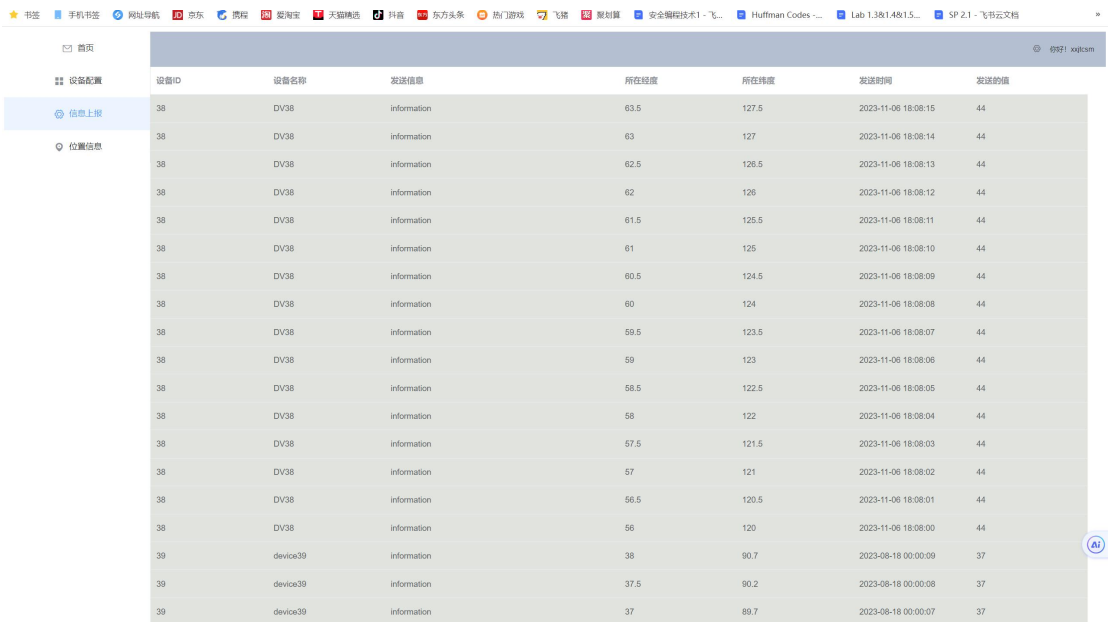

图 7 信息上报界面

6.5 设备轨迹查看界面

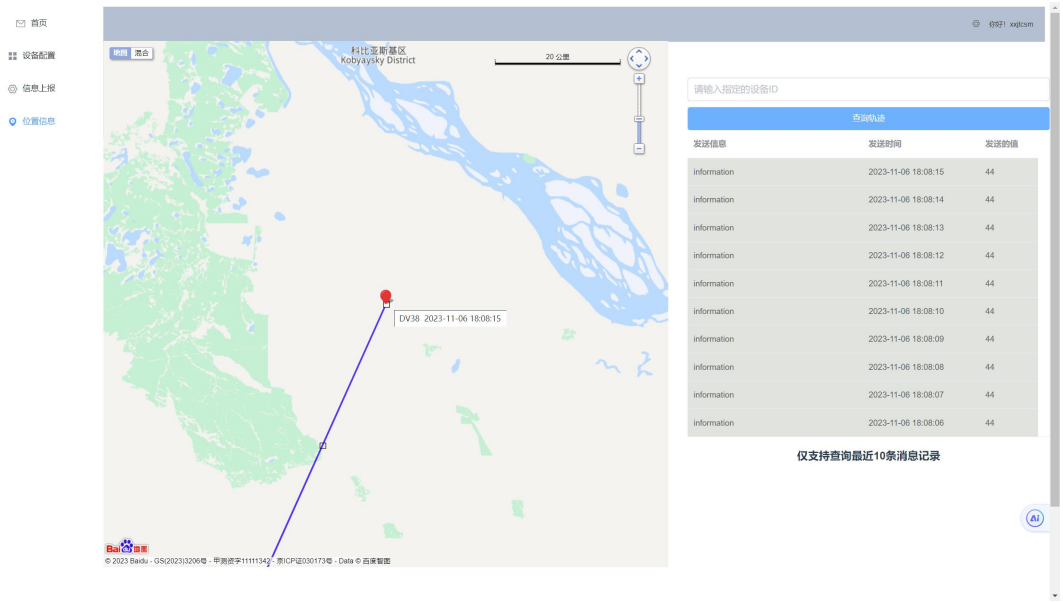

图 8 设备轨迹查看界面

七、附录

7.1 项目进度安排

图 9 开发计划甘特图

7.2 参考文档

- Vue：渐进式 JavaScript 开发框架

  https://v2.cn.vuejs.org/

- Element，一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库

  https://element.eleme.cn/#/zh-CN

- Spring Boot

  https://spring.io/projects/spring-boot

- BaiduMap:

- https://dafrok.github.io/vue-baidu-map/#/zh/index%C2%A0