

# 浙江大学

## 本科实验报告

课程名称: B/S 体系软件设计

姓 名: 张溢弛

学 院: 计算机科学与技术学院

系: 软件工程

专 业: 软件工程 1801

学 号: 3180103772

指导教师: 胡晓军

2021 年 4 月 25 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: 综合型  
实验项目名称: 物联网应用网站开发  
学生姓名: 张溢弛 专业: 软件工程 学号: 3180103772  
同组学生姓名: 无 指导老师: 胡晓军  
实验地点: 玉泉一舍 376 实验日期: 2021 年 4 月 25 日

## 物联网应用网站-设计报告

### 目 录

一、项目背景 .....	5
二、系统需求分析 .....	5
2.1 功能性需求分析 .....	5
2.2 非功能性需求分析 .....	6
2.2.1 性能需求 .....	6
2.2.2 输入输出需求 .....	7
2.2.3 数据管理需求 .....	7
三、系统技术选型与架构设计 .....	8
3.1 项目技术选型 .....	8
3.2 项目主要技术介绍 .....	8
3.2.1 前端技术框架 .....	8
3.2.2 后端技术框架 .....	9
3.2.3 MQTT 服务器 .....	9
3.3 系统总体架构图 .....	9
3.4 前端架构图 .....	10
3.5 后端架构图 .....	11
3.6 系统运行环境 .....	11
3.6.1 软件环境 .....	11

3.6.2 硬件环境 .....	12
四、数据库设计与 ER 图 .....	13
4.1 数据表设计 .....	13
4.1.1 user 表 .....	13
4.1.2 device 表 .....	13
4.1.3 message 表 .....	14
4.2 ER 图 .....	14
五、系统接口设计 .....	15
5.1 用户信息相关接口 .....	15
5.1.1 用户登录 .....	15
5.1.2 用户注册 .....	16
5.1.3 用户修改密码 .....	16
5.1.4 用户修改绑定邮箱 .....	16
5.1.5 用户修改手机号 .....	17
5.2 设备配置信息接口 .....	17
5.2.1 获取单个设备信息 .....	17
5.2.2 修改设备配置信息 .....	17
5.2.3 查询用户设备总数 .....	18
5.2.4 查询用户持有设备列表 .....	18
5.2.5 查询用户不同类型设备数量 .....	18
5.2.6 查询当前活跃设备数量 .....	19
5.3 设备消息相关接口 .....	19
5.3.1 查询设备历史轨迹 .....	19
5.3.2 查询设备历史消息 .....	19
5.3.3 查询设备 value 属性的变化情况 .....	20
5.3.4 查询当前消息总数 .....	20
六、系统界面原型 .....	21
6.1 登陆界面 .....	21
6.2 注册界面 .....	21
6.3 网站首页 .....	22
6.4 个人信息界面 .....	23

6.5 设备配置界面 .....	24
6.6 设备轨迹查看界面 .....	25
6.7 设备统计信息界面 .....	27
七、附录 .....	28
7.1 项目进度安排 .....	28
7.2 备注 .....	28

# 一、项目背景

本项目是 2020-2021 春夏学期《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，用户在登陆之后可以修改个人信息和物联网设备配置信息，并查看设备信息的可视化界面，包括运动轨迹和设备的统计信息，同时需要网站界面对用户友好，样式适配手机移动端，可以在手机浏览器和微信等应用内置的浏览器中友好显示，同时也需要提供必要的软件项目文档，使自己了解并掌握一套 web 应用开发技术和开发的总体流程。

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，以及数据库的设计和系统接口、界面原型的设计等内容，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，mqtt 服务器，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

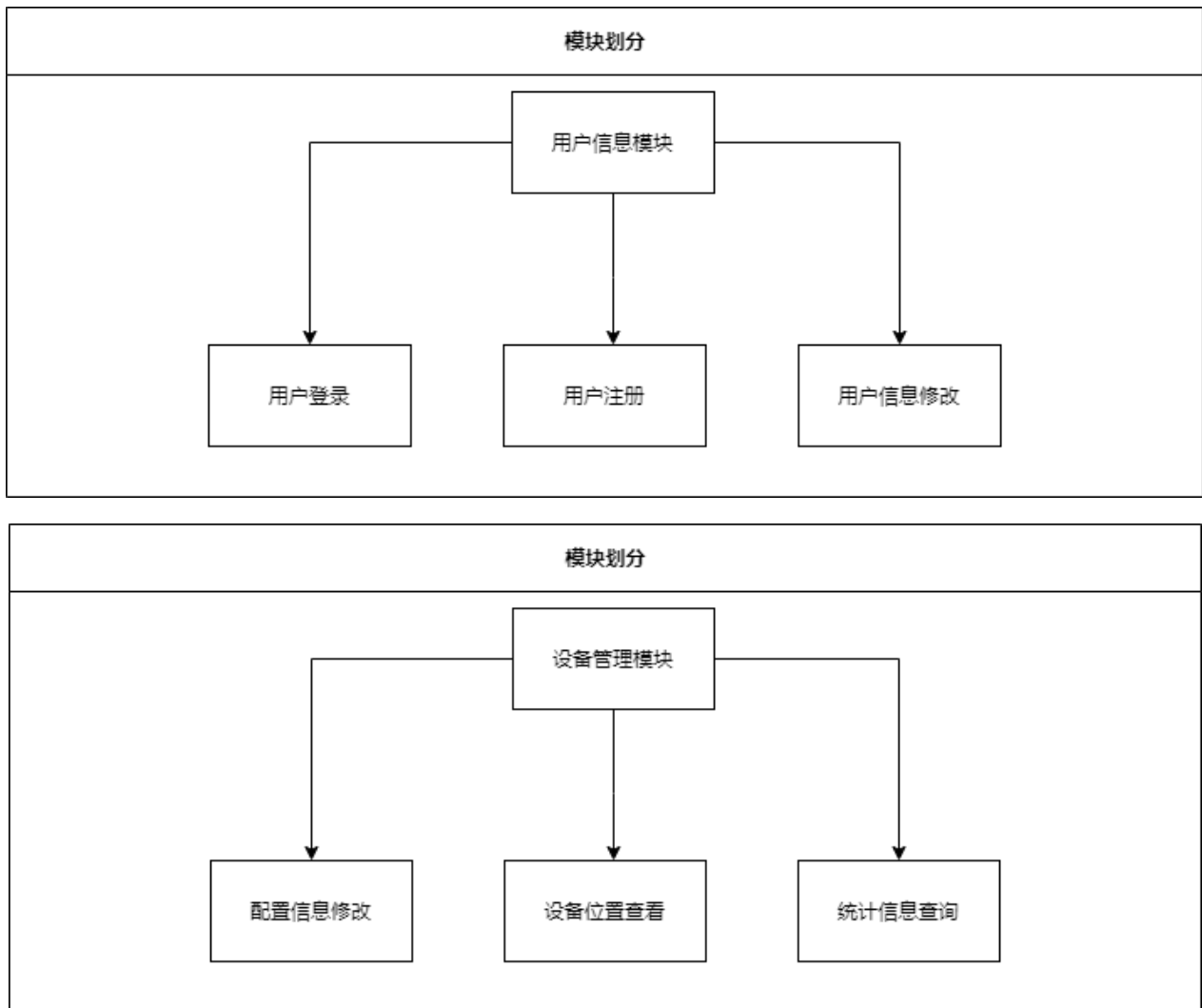
### 2.1 功能性需求分析

该项目主要是一个 B/S 架构的 web 应用，对于**每个用户**需要实现如下功能性需求：

- 用户注册(需要用户名，密码，手机号，邮箱等关键信息)
- 用户登录
- 用户信息修改
- 设备位置信息查看(可视化界面，显示设备的运动轨迹)
- 设备配置信息修改
- 设备统计数据查看(可视化界面，显示基于图表的统计信息)

该系统主要用户有**两种**，一种是**普通用户**，一种是**管理员**，其中普通用户可以查看到的信息仅包含该用户持有的一系列设备，而管理员可以查看到所有存在于数据库中的设备的一系列信息。

因此，该项目的功能性需求主要可以分成两个模块，一个是用户信息模块，另一个是设备管理模块，这些功能性需求的关系和模块划分用图可以表示为：



其中每个功能下还有若干子功能，比如用户信息修改包括密码修改和邮箱手机号的修改，配置信息修改包括新增设备和修改已有设备，设备位置查看包括地图轨迹查看和消息记录查看，统计信息查询包括设备数查询，消息总数查询和设备种类查询等等。

## 2.2 非功能性需求分析

该系统的非功能性需求包括性能需求，输入输出需求，数据管理需求等，具体的要求如下：

### 2.2.1 性能需求

- 系统应保证运行稳定，避免出现崩溃；
- 当前主流浏览器均能正常访问本系统；
- 系统应能保证至少 100 人的并发访问；
- 当用户登录以及进行任何操作时，系统应该能及时进行反应，反应的时间 在 1s 以内；

- 系统应该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，避免用户长时间等待；
- 每个页面一般情况下应在 1s 内加载完毕，高峰期应在 5s 内加载完毕；
- 系统保证在一周内不超过一次维护与重启。

### 2.2.2 输入输出需求

- 在用户输入账号密码时，应对数据输入进行数据有效性检查，同时应该确保其安全性；
- 设备的运动轨迹和统计信息必须以可视化的图标等方式呈现，需要美观而准确显示数据库中的数据
- 此外，系统应通过程序控制出错几率，减少系统因用户人为的错误引起的破坏，开发者应当尽量周全地考虑到各种可能发生的问题，使出错的可能降至最小。

### 2.2.3 数据管理需求

- 系统既要与其他系统有接口，又必须保证本系统的独立性与完整性。即应防止未经授权的各类人员对本系统进行设置和修改或访问系统内部数据。
- 系统服务器软件必须提供可靠的数据备份和恢复手段，在服务器软件或硬件出现严重故障时，能够根据备份的数据和账户信息迅速恢复正常运行环境。同时，软件开发者不得在系统中预留任何特殊账户和密码，保证其安全性。
- 除此之外，系统应具备加密登录、数据加密传输等安全方面的保障，保证数据在不用系统间传输过程中的保密性与安全性。

## 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 RESTful 风格的 api 供前端调用，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，整个项目技术选型如下所示

- 前端：React + Ant Design 组件库+Ant Design Charts+React-Router+百度地图 SDK+yarn 包管理工具
- 后端：Java Spring Boot 框架+Maven+MyBatis+Redis+Jackson
- 数据库：MySQL
- mqtt 服务器：Python3.8+mysql-connector+Paho(额外 python 库)
- 测试工具：ApiPost 用于测试后端接口数据正确性

### 3.2 项目主要技术介绍

#### 3.2.1 前端技术框架

项目中主要使用了 React + Ant Design 组件库+Ant Design Charts+React-Router+百度地图 SDK+yarn 包管理工具作为前端开发的技术栈。

React 是 FaceBook 公司开发的一套 Web 开发框架，基于 Javascript 来构建用户界面，React 的设计是 Web 前端开发中的革命性创新，采用虚拟 DOM 的技术和声明式设计，可以非常高效而灵活地开发 Web 应用。

Ant Design 是蚂蚁集团开发的开源的企业级产品 UI 框架，提供了大量美观而功能性强的 UI 组件，而 Ant Design Charts 是 Ant Design 的衍生品，提供了一系列优质的统计图表模板用于数据的可视化。

百度地图 SDK 是百度开发的一款基于百度地图的前端开发工具包，并提供了对 React 的支持，百度地图 SDK 提供了地图组件，并且可以在地图中进行点和线的标注和绘制，可以用于设备位置的可视化。



### 3.2.2 后端技术框架

本项目的后端主要采用了 Java Spring Boot 框架+Maven+MyBatis+Redis+Jackson 的技术栈，同时使用了 MySQL 作为关系型数据库。其中 Maven 是 Java 工程项目的包管理工具，MyBatis+Redis+Jackson 是一系列中间件和对应的 Spring Boot 集成开发包。

Spring 是一个基于 Java 的开源应用框架，提供具有控制反转特性的容器，并且具有面向切面编程(AOP)的特性，利用容器管理对象的生命周期，可以使用 XML 文件进行一系列配置。Spring Boot 是基于 Spring 的轻量级框架，继承了 Spring 框架原有的优秀特性，并且简化了 Spring 应用的搭建开发过程，并且集成了一系列框架解决了依赖包的版本冲突问题。

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。MyBatis 可以通过简单的 XML 或注解来配置和映射原始类型、接口和 Java POJO（Plain Old Java Objects，普通老式 Java 对象）为数据库中的记录。

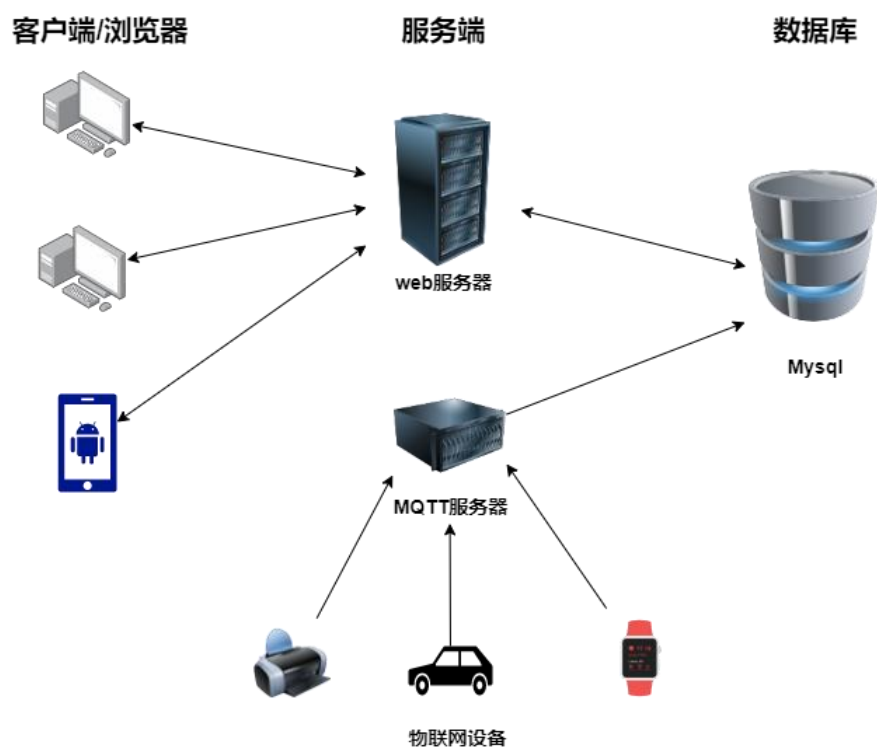
Redis 是一个运行在内存中的 key-value 数据库，可以用于存储数据结构和缓存以及消息队列等内容，本项目中使用 Redis 来存储 token，并在 SpringBoot 项目中集成了 Redis，可以提高 token 存取的效率。

### 3.2.3 MQTT 服务器

本项目的 MQTT 服务器采用了 mosquitto，但是因为 mosquitto 不能存储消息，因此本项目使用 Python 编写了一个服务端脚本进行了二次开发，采用 Paho 库来接收设备模拟器客户端发送过来的消息，并存储到数据库中。

## 3.3 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。



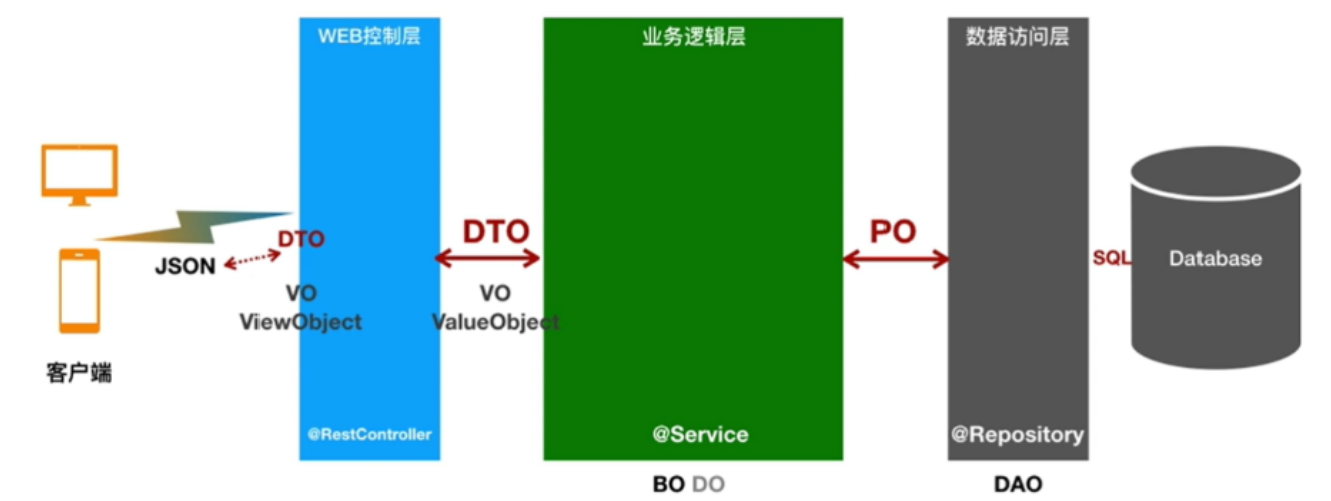
### 3.4 前端架构图

该项目的前端主要采用基于 Node.js 的 React 框架，并采用阿里主导的开源项目 Ant Design 作为 UI 组件框架，百度地图 SDK 作为设备的经纬度地理位置可视化工具，采用 React 全家桶开发，前端总体架构可以用下面的图来表示：



### 3.5 后端架构图

该项目后端主要使用 Spring Boot 框架编写而成，并集成一系列中间件，Jackson、MyBatis 和 Redis 等等来完善后端的功能，采用 json 格式进行数据的收发，其主要的架构可以用下图表示：



### 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 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	Linux socket	
数据库服务器类型	MySQL	8.0
浏览器	Chrome	

### 3.6.2 硬件环境

项目	名称
操作系统	CPU: CORE i5 及以上
	内存: 2G 及以上
	硬盘: 500G 及以上
应用服务器	内存: 512M 及以上
数据库服务器	硬盘: 50G 及以上
邮件服务器	
文件服务器	
通讯设备	网线: 具有良好的数据传输能力

# 四、数据库设计与 ER 图

## 4.1 数据表设计

本项目的数据库共设计了三个数据表，分别用于存储用户信息、设备配置信息和 mqtt 服务器接收到的消息，具体的表结构设计情况如下：

### 4.1.1 user 表

- 该表用来保存用户的信息

字段名	类型	描述	备注
id	int	用户的 id	主键，非空，自增
name	varchar(128)	用户名	非空
password	varchar(128)	账号密码	非空
email	varchar(128)	用户邮箱	非空，并且 UNIQUE
phone	varchar(128)	用户手机号	非空

### 4.1.2 device 表

- 该表用于保存设备的配置信息

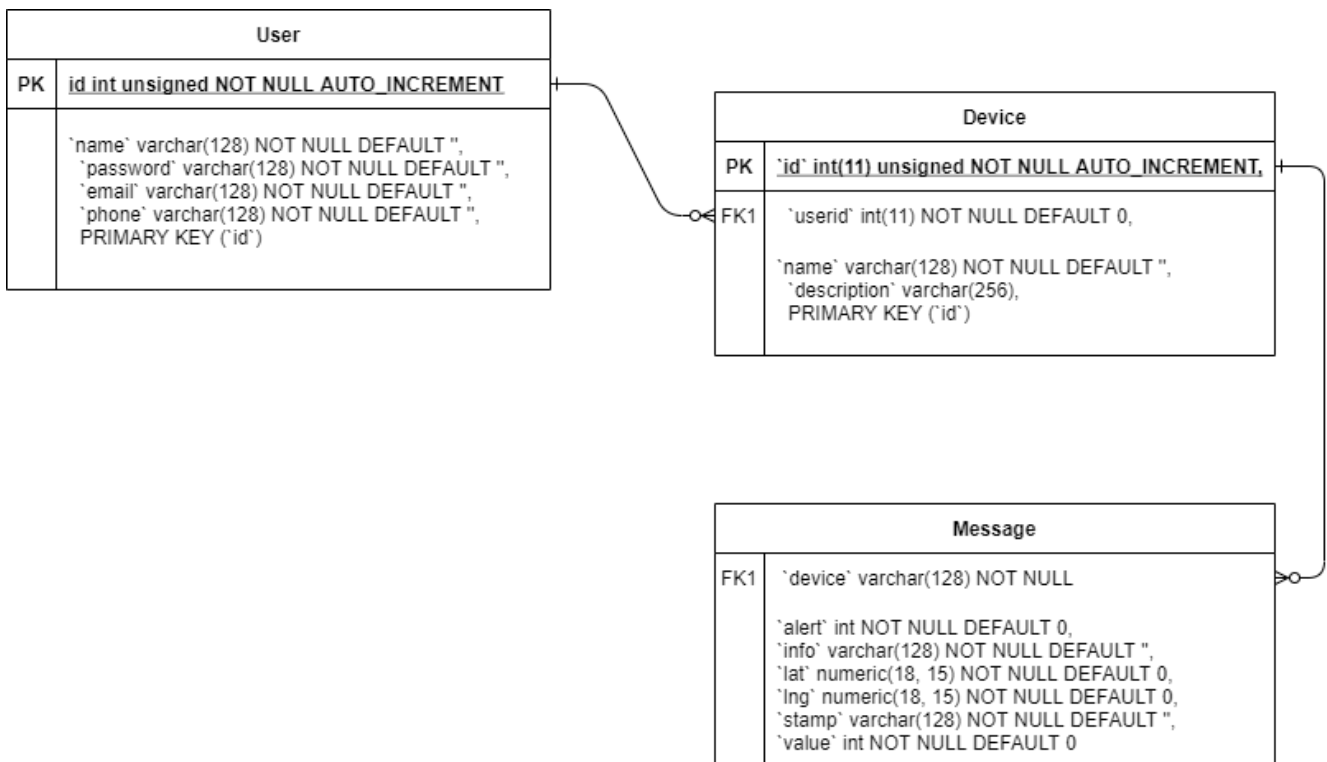
字段名	类型	描述	备注
id	int	设备的 id	主键，非空，自增
name	varchar(128)	设备的名称	非空，Unique
description	varchar(256)	设备的描述信息	非空
userid	int	设备的用户的 id	非空，作为外键
kind	int	设备的类型	非空，默认是 1，目前暂定的类型有车载设备、智能家居、可穿戴设备、基础设施和其他设备等类型，在统计设备种类分布情况的时候起到比较重要的作用。
activate	varchar(128)	上次活跃的时间	非空，默认”

### 4.1.3 message 表

- 该表用于保存 mqtt 服务器接收到的设备状态消息

字段名	类型	描述	备注
device	varchar(128)	设备名	非空，可作为外键
alert	int	是否触发警报	0 表示正常，1 表示警报
info	varchar(128)	设备发送的消息	默认是”
lat	numeric	设备发送的经度	默认是 0
lng	numeric	设备发送的纬度	默认是 0
stamp	varchar(128)	发送时候的时间戳	
value	int	设备发送的一个值	默认是 0

### 4.2 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、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。

不过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。

此外，由于本系统的后端使用了 Spring-Boot-Starter-json 模块，因此一些以 Map 类型和已定义 model 类型作为返回值的 api 的返回值将被自动转换成 JSON 格式的数据，以下接口设计表中给出的都是原本的返回值类型。

## 5.1 用户信息相关接口

用户信息类接口主要使用 User 表的数据，并对 User 表进行增删查改，提供的接口 URL 均以 /user 开头，具体的 api 主要有：

### 5.1.1 用户登录

接口 URL	/user/login
主要参数列表	Map<string, Object> form 包含登陆界面提交的表单中的信息，如用户名和密码等
返回值类型	Map 类型，其中“code”属性是整数 1 表示成功，0 表示用户不存在，2 表示账号或者密码错误，“token”属性包含服务器为其生成的 token
接口简介	用于网站的登录认证，在登陆成功后会生成 token 并返回给用户保存在浏览器端。

### 5.1.2 用户注册

接口 URL	/user/register
主要参数列表	Map<string, Object> form 包含注册界面提交的表单中的信息，如用户名，密码，邮箱，手机号等等，其中 <b>邮箱和手机号的正确性以及密码的安全性已经在前端进行了验证，因此不需要后端再进行检验</b>
返回值类型	整型，1 表示注册成功，0 表示注册失败用户已经存在
接口简介	用于用户注册新账号时使用，在数据表中添加新用户的信息

### 5.1.3 用户修改密码

接口 URL	/user/change/password
主要参数列表	Map<string, Object> form 包含新密码和旧密码
返回值类型	整型，1 表示修改成功，0 表示修改失败
接口简介	用户修改密码

### 5.1.4 用户修改绑定邮箱

接口 URL	/user/change/email
主要参数列表	Map<string, Object> form 包含新邮箱和旧邮箱
返回值类型	整型，1 表示修改成功，0 表示修改失败
接口简介	用户修改邮箱



### 5.1.5 用户修改手机号

接口 URL	/user/change/phone
主要参数列表	Map<string, Object> form 包含新手机号和旧手机号
返回值类型	整型，1 表示修改成功，0 表示修改失败
接口简介	用户修改手机号

## 5.2 设备配置信息接口

设备配置信息相关的接口主要提供对设备配置信息的增删查改功能，这一类接口的 URL 都用/device 开头，具体设计如下：

### 5.2.1 获取单个设备信息

接口 URL	/device/{#id}
主要参数列表	String id 从 GET 请求的 URL 中提取
返回值类型	Device 对象，包含一台设备的信息
接口简介	用于获得单台设备的信息

### 5.2.2 修改设备配置信息

接口 URL	/device/config
主要参数列表	Map 类型，存储需要修改的设备信息字段和对应的信息
返回值类型	整型，1 表示成功，0 表示失败
接口简介	修改设备的配置信息，比如设备名称等等。

### 5.2.3 查询用户设备总数

接口 URL	/device/query/all
主要参数列表	Map<String, Object>存储了用户名和 token，需要先验证 token 的有效性
返回值类型	整型，返回当前用户持有的设备总数
接口简介	用于查询某个用户拥有的设备总数

### 5.2.4 查询用户持有设备列表

接口 URL	/device/query/list
主要参数列表	Map<String, Object>存储了用户名和 token，需要先验证 token 的有效性
返回值类型	数组，存储了当前用户的一系列设备编号
接口简介	用于查询某个用户当前持有的所有设备的列表。

### 5.2.5 查询用户不同类型设备数量

接口 URL	/device/query/kind
主要参数列表	Map<String, Object>存储了用户名和 token，需要先验证 token 的有效性
返回值类型	Map 类型，存储了每种类型对应的设备数量，用类型做 key，数量作为 Value
接口简介	用于查询用户的不同类型设备总数

## 5.2.6 查询当前活跃设备数量

接口 URL	/device/query/active
主要参数列表	Map<String, Object>存储了用户名和 token，需要先验证 token 的有效性
返回值类型	整型，返回当前活跃的设备数量，当 token 认证无效的时候返回-1
接口简介	用于查询当前活跃设备数量，暂定查询当天有消息记录的设备为活跃设备

## 5.3 设备消息相关接口

涉及到设备消息的接口全部使用/message 作为 URL 的开头，一般来说这一部分只提供查询的接口，具体的接口定义有如下内容：

### 5.3.1 查询设备历史轨迹

接口 URL	/message/path/{#id}
主要参数列表	String id 从 GET 请求 URL 中得到参数
返回值类型	Message 数组，包含一系列该设备的 message，当数目较多的时候仅显示最活跃的 20 条记录
接口简介	用于获取一个设备提交的 Message 的时间和提交时候的经纬度等信息，并在前端地图组件上进行渲染。

### 5.3.2 查询设备历史消息

接口 URL	/message/info/{#id}
主要参数列表	String id 从 GET 请求 URL 中得到参数

返回值类型	Message 数组，包含一系列该设备的 id 和对应的 info 字段中的信息
接口简介	用于查询设备发送给 mqtt 服务器的历史消息

### 5.3.3 查询设备 value 属性的变化情况

接口 URL	/message/value/{#id}
主要参数列表	String id 从 GET 请求 URL 中得到参数
返回值类型	Message 数组，包含一系列该设备的 id 和对应的 value 字段中的信息
接口简介	用于查询设备发送给 mqtt 服务器的历史消息

### 5.3.4 查询当前消息总数

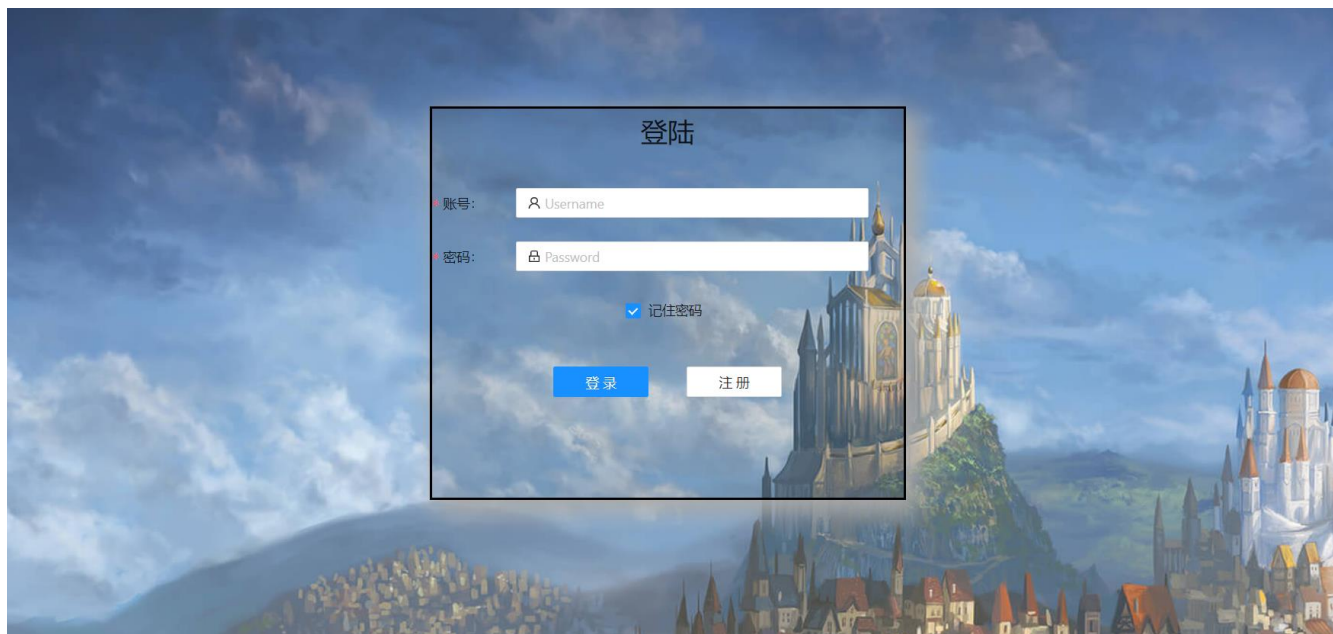
接口 URL	/message/all/
主要参数列表	Map<String, Object> 从请求体中获得的用户名和 token 构成的 Map，需要先验证 token 的有效性，返回当前用户持有设备
返回值类型	整型，返回当前用户持有设备的消息总数
接口简介	查询用户当前持有设备发送过来的消息总数

## 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。**

注：该设计报告完成于最终网站成形之前，因此最终许多地方都与系统界面原型设计有较大的差异。以下界面原型仅供参考。

### 6.1 登陆界面



### 6.2 注册界面

注册

\*电子邮箱:

e-mail

\*用户账号:

Username

\*输入密码:

Password

\*确认密码:

\*手机号码:

+86

☐

我已阅读并同意

用户须知

注册

## 6.3 网站首页

zyc

退出

物联网应用平台

首页

个人信息

设备配置

查看设备

设备趋势图

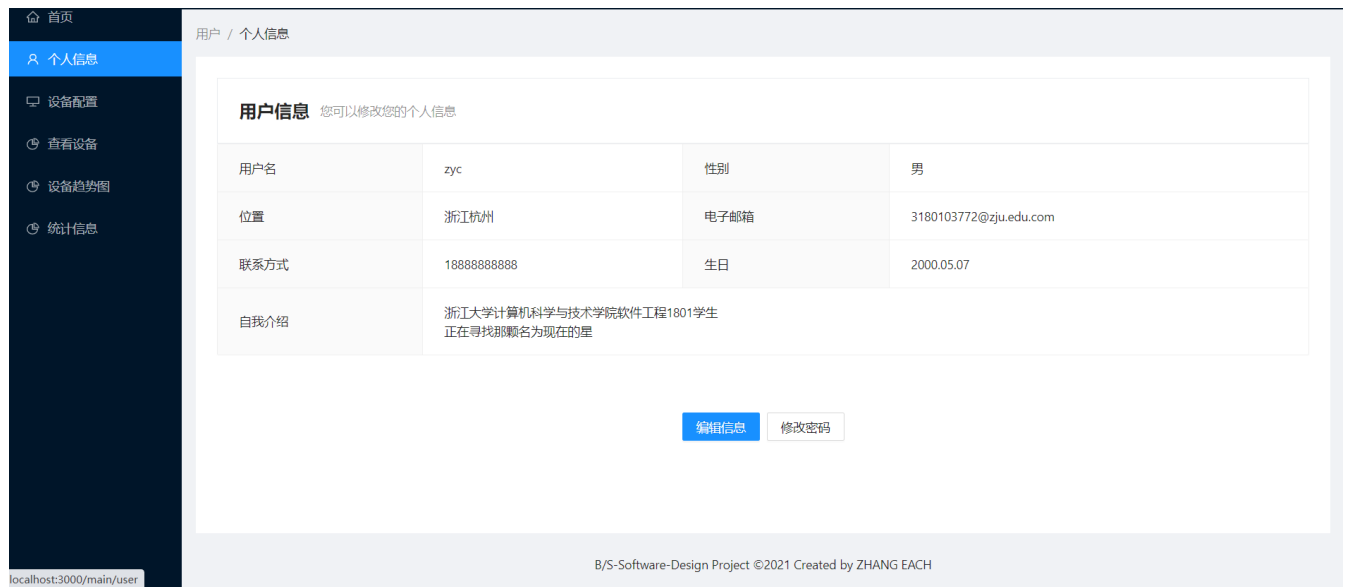
统计信息

首页 / 网站介绍

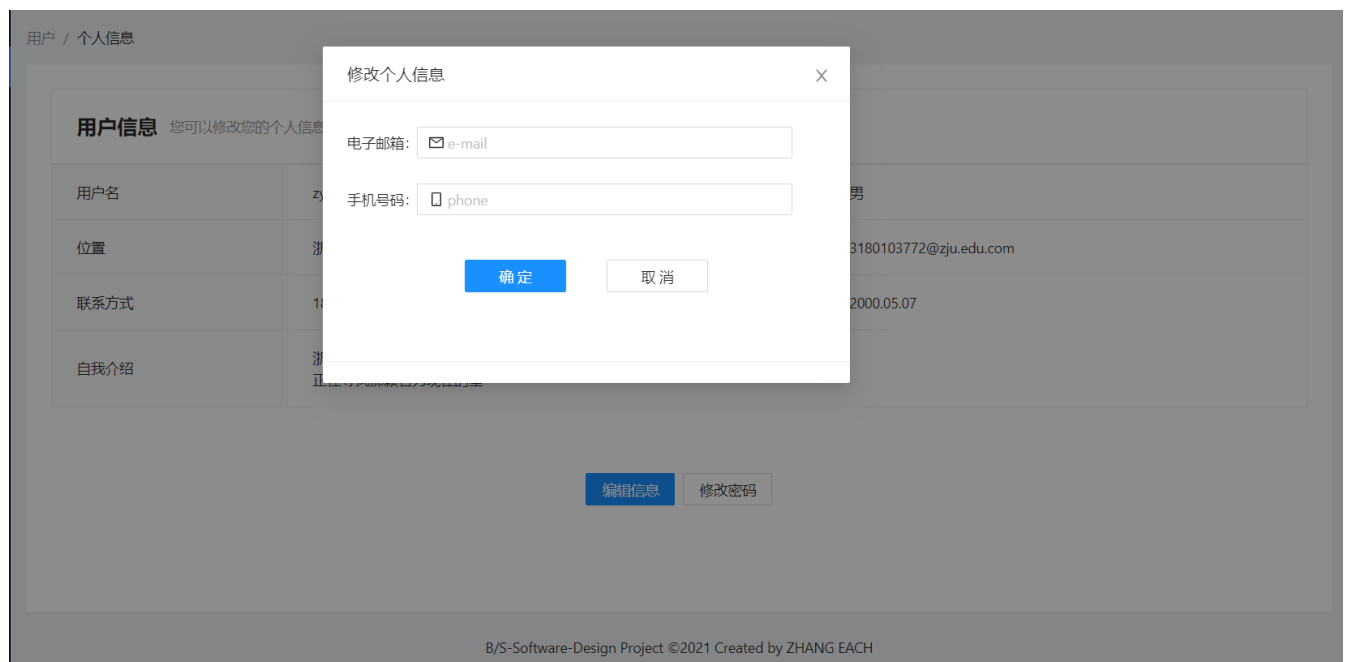
欢迎来到物联网应用网站! 您可以在本网站中修改个人信息, 并修改设备配置, 查看您的物联网设备的信息

localhost:3000/main

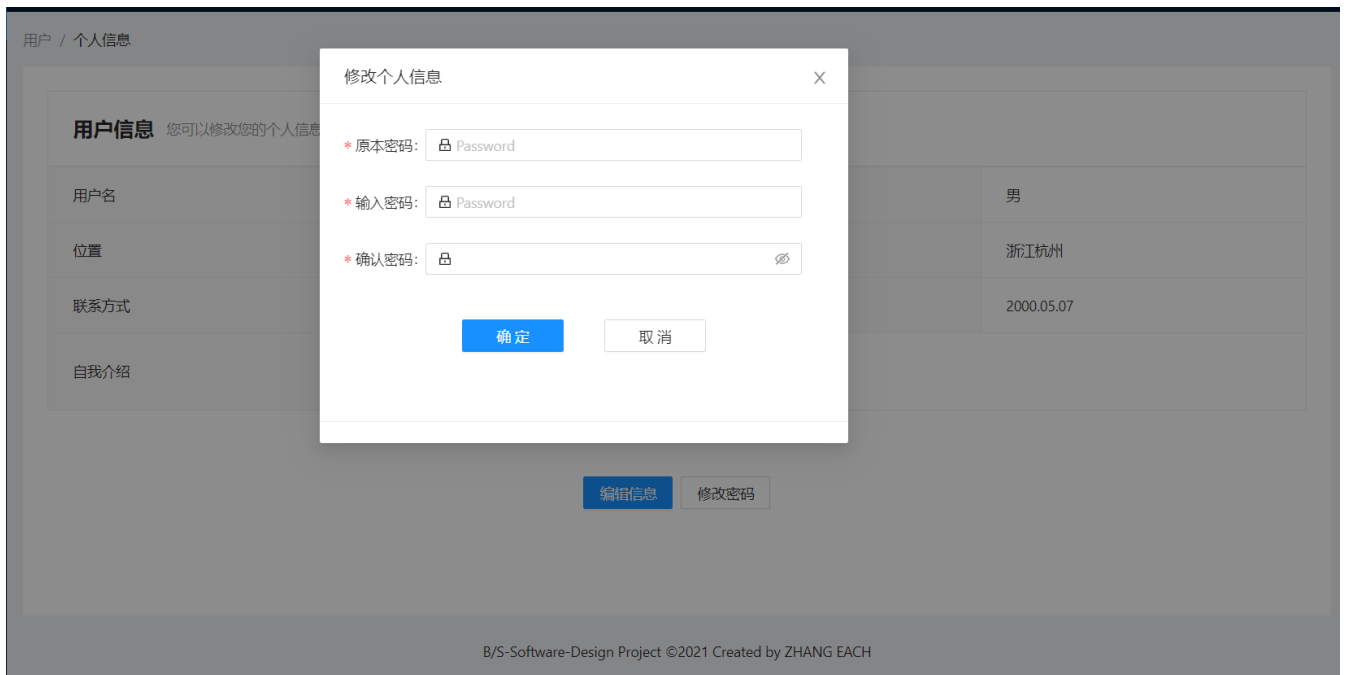
## 6.4 个人信息界面



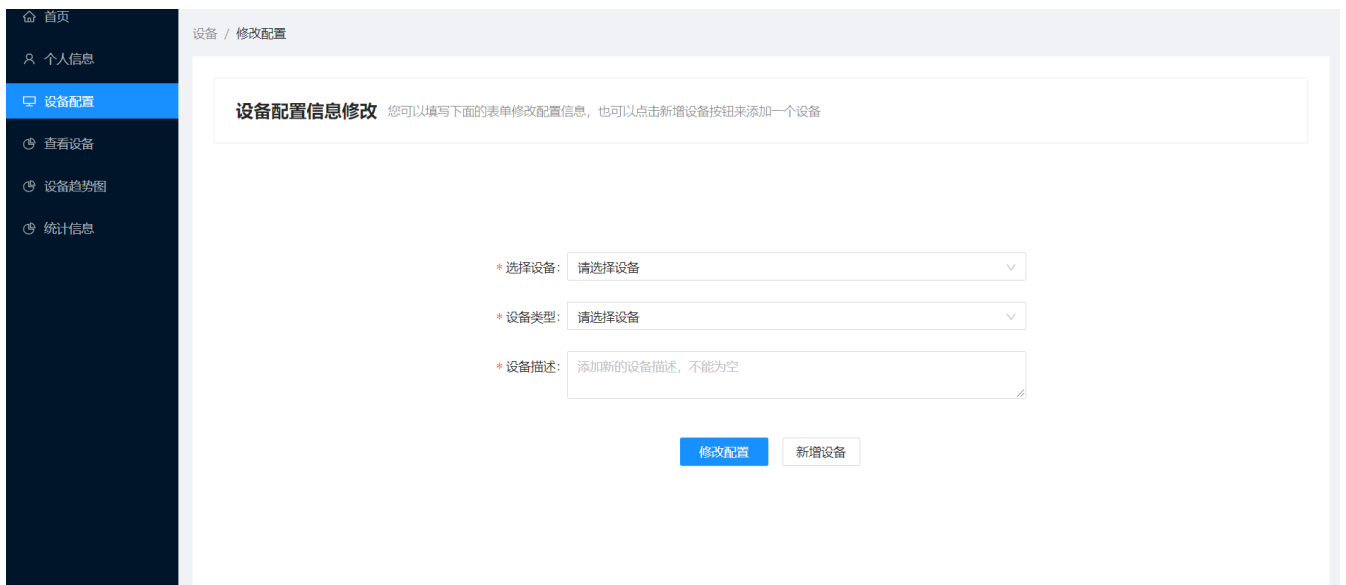
### ● 修改个人信息的界面



### ● 修改密码的界面

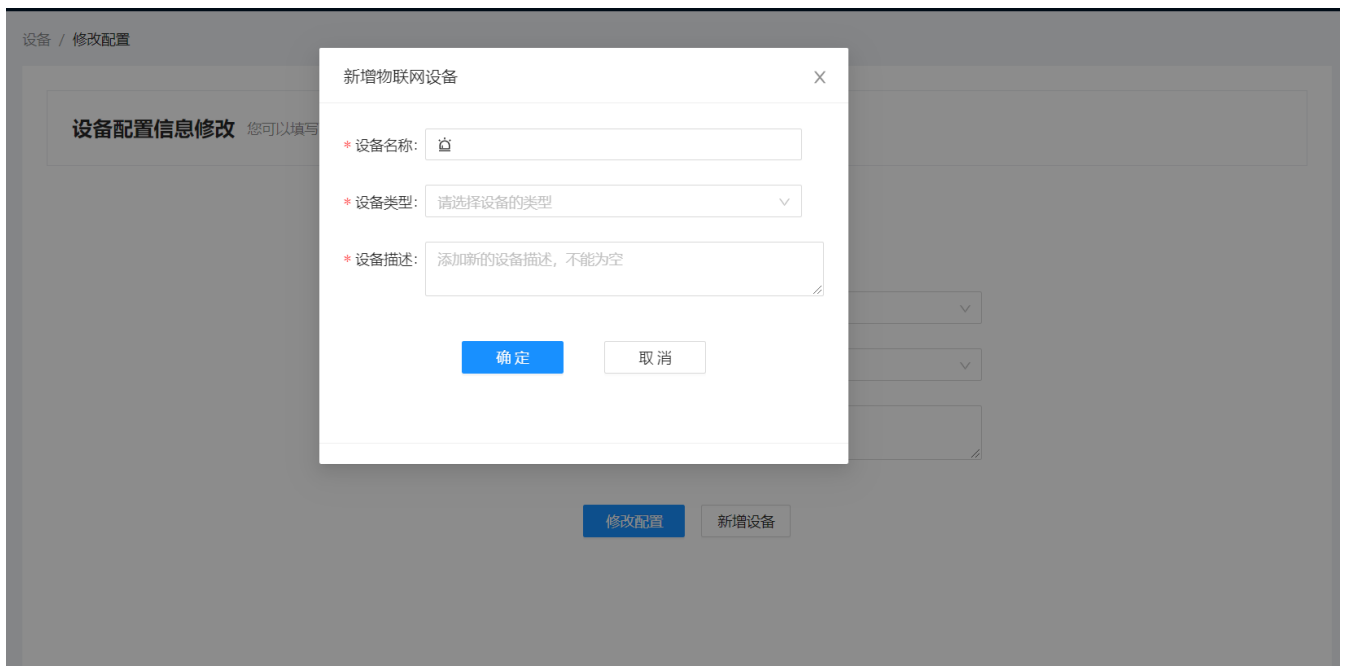


## 6.5 设备配置界面

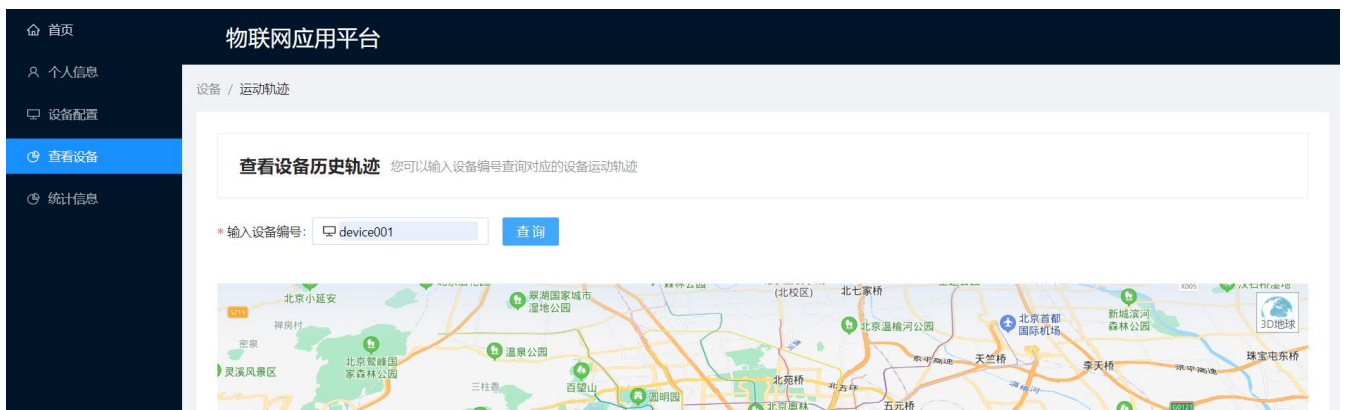


### ● 新增设备子界面



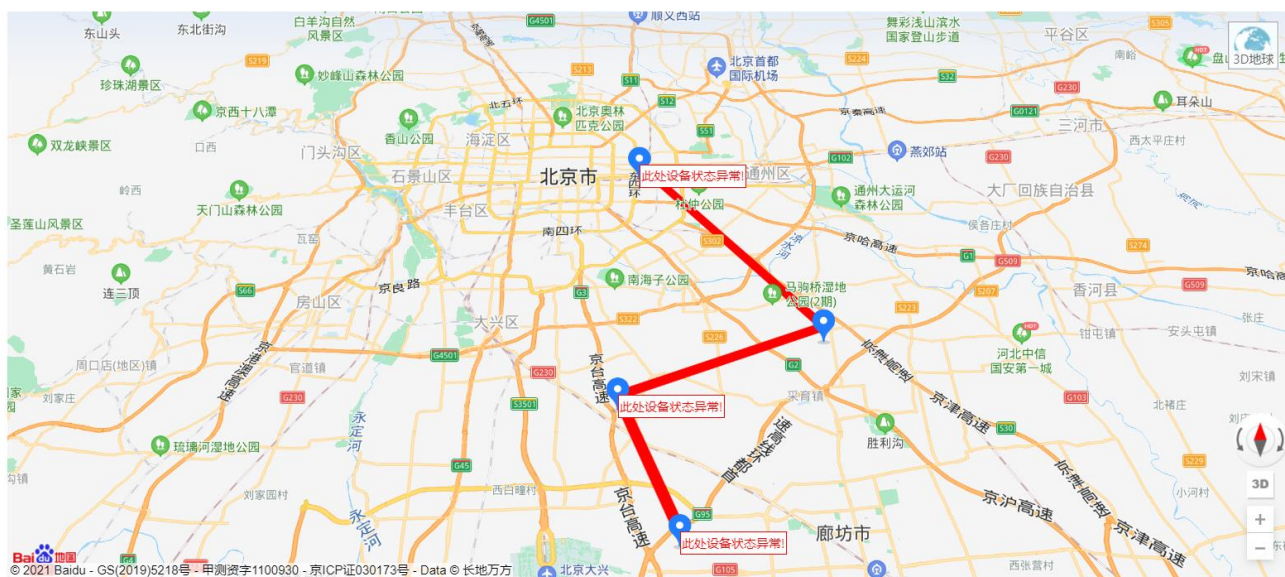


## 6.6 设备轨迹查看界面



- 具体的地图效果（因为比较大上面一张放不下），采用百度地图 SDK 开发而成，可以显示设备经过的点和路径，并在 **alert** 处提示消息异常

\* 输入设备编号:



B/S-Software-Design Project ©2021 Created by ZHANG EACH

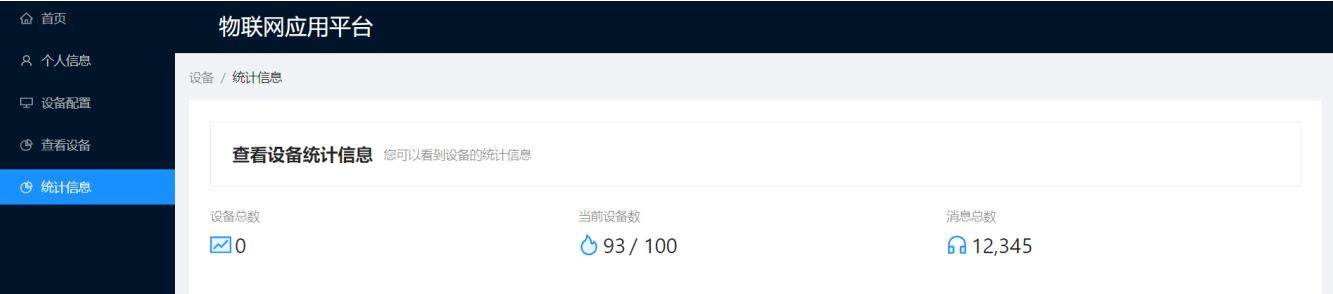
## ● 设备的消息记录界面

### 设备消息记录

★	消息发送时间: 2021-04-29 23:45:01 消息内容: Device Data 2021/04/29 23:45:01
★	消息发送时间: 2021-04-29 23:44:37 消息内容: Device Data 2021/04/29 23:44:37
★	消息发送时间: 2021-04-29 23:44:37 消息内容: Device Data 2021/04/29 23:44:37
★	消息发送时间: 2021-04-29 23:44:26 消息内容: Device Data 2021/04/29 23:44:26
★	消息发送时间: 2021-04-29 23:44:19 消息内容: Device Data 2021/04/29 23:44:19
★	消息发送时间: 2021-04-29 23:44:10 消息内容: Device Data 2021/04/29 23:44:10
★	消息发送时间: 2021-04-29 23:44:04 消息内容: Device Data 2021/04/29 23:44:04
★	消息发送时间: 2021-04-29 23:43:44 消息内容: Device Data 2021/04/29 23:43:44

# 6.7 设备统计信息界面

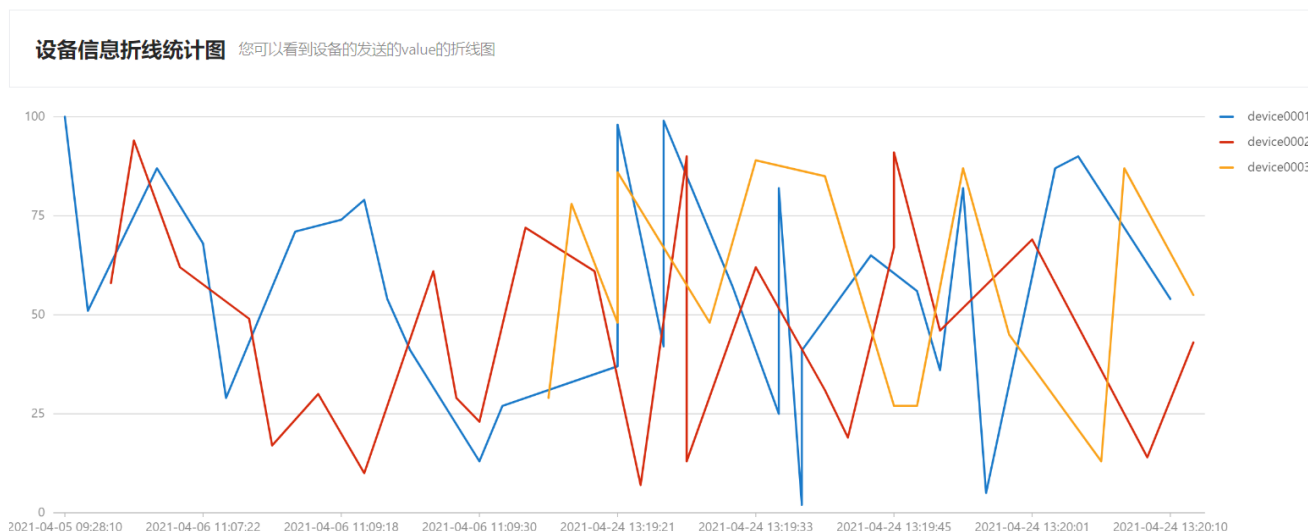
- 元数据的统计信息



- 可以显示用户持有的设备种类的扇形统计图：



- 设备 value 变化的统计图，采用多折线统计图来呈现（目前仅为实例，最终呈现结果可能略有差异）



## 七、附录

### 7.1 项目进度安排

时间段	计划进度
<b>2021.04</b>	完成系统的设计与初步开发
<b>2021.05 上旬</b>	基本完成系统的前后端代码开发
<b>2021.05 中下旬</b>	完成系统的集成与测试，编写系统的各个文档

### 7.2 备注

本设计文档中未竟之事均以最后提交的系统源代码和文档为准。