

**本科实验报告**

|  |  |
| --- | --- |
| 课程名称： | B/S体系软件设计 |
| 姓 名： |  |
| 学 院： | 计算机科学与技术学院 |
| 系： | 计算机 |
| 专 业： | 计算机科学与技术 |
| 学 号： |  |
| 指导教师： |  |

2023年 11 月 19 日

目录

[1. 项目背景 3](#_Toc151337597)

[2. 系统设计 4](#_Toc151337598)

[2.1 需求分析 4](#_Toc151337599)

[2.1.1 功能性 4](#_Toc151337600)

[2.1.2 非功能性 6](#_Toc151337601)

[2.2 运行环境 7](#_Toc151337602)

[2.2.1 软件环境 7](#_Toc151337603)

[2.2.2 硬件环境 7](#_Toc151337604)

[3. 技术选型 7](#_Toc151337605)

[3.1 项目技术选型 7](#_Toc151337606)

[3.2 项目主要技术介绍 9](#_Toc151337607)

[4. 项目架构图 10](#_Toc151337608)

[5. 数据库与ER图设计 11](#_Toc151337609)

[5.1 数据表 11](#_Toc151337610)

[5.1.1 user 11](#_Toc151337611)

[5.1.2 device 12](#_Toc151337612)

[5.1.3 message 13](#_Toc151337613)

[5.2 ER图 14](#_Toc151337614)

[6. API LIST 14](#_Toc151337615)

[6.1 用户信息接口 14](#_Toc151337616)

[6.1.1 用户登录 14](#_Toc151337617)

[6.1.2 用户注册 15](#_Toc151337618)

[6.1.3 用户修改密码 15](#_Toc151337619)

[6.1.4 用户修改邮箱 15](#_Toc151337620)

[6.2 设备配置信息接口 15](#_Toc151337621)

[6.2.1 获取单个设备信息 15](#_Toc151337622)

[6.2.2 修改设备配置信息 16](#_Toc151337623)

[6.2.3 查询用户设备总数 16](#_Toc151337624)

[6.2.4 查询用户持有设备列表 16](#_Toc151337625)

[6.2.5 查询不同类型设备数量 16](#_Toc151337626)

[6.2.6 查询当前活跃设备数量 16](#_Toc151337627)

[6.3 设备消息相关接口 17](#_Toc151337628)

[6.3.1 查询设备历史轨迹 17](#_Toc151337629)

[6.3.2 查询设备历史消息 17](#_Toc151337630)

[6.3.3 查询value属性变化 17](#_Toc151337631)

[6.3.4 查询当前消息总数 17](#_Toc151337632)

[7. 项目开发时间轴 18](#_Toc151337633)

# 1. 项目背景

物联网（IoT）是一种基于现有互联网技术，通过扩展边缘设备来构建网络的概念，它通过将多种信息感知设备与网络相结合，实现了人、机器和物体在任何时间和地点的互联互通。

物联网的应用广泛，覆盖了工业、农业、环境保护、交通、物流、安全等多个基础设施领域。它促进了这些领域的智能化发展，使资源分配更加合理高效，从而提升了行业的效率和盈利能力。此外，在家庭、医疗健康、教育、金融服务业和旅游业等与日常生活紧密相关的领域，物联网也大大改善了服务的范围、方式和质量，显著提高了人们的生活水平。

该项目是2023-2024年秋冬学期《B/S体系软件设计》课程的一部分，目的是开发一个物联网应用网站。该网站允许用户登录后修改个人信息和物联网设备的配置，并提供设备信息的可视化展示，如运动轨迹和统计数据。网站界面需用户友好，能够适配手机端，以便在手机浏览器和微信等应用中良好显示，并提供必要的项目文档，以帮助用户理解和掌握Web应用开发的技术和流程。

《设计报告》涵盖了系统的需求分析、总体架构设计、数据库设计以及系统接口和界面原型的设计。报告详细介绍了系统的实现，阐述了设计计划，明确了系统概览和结构设计，包括产品和项目的总体结构、架构设计和错误处理设计，还制定了具体的时间表，以便后续开发跟进。

# 2. 系统设计

## 2.1 需求分析

### 2.1.1 功能性

该项目主要是一个 B/S 架构的 web 应用，需要实现以下需求：

* 用户注册

提供用户名、密码、邮箱等注册，保证邮箱唯一性，同时验证其格式的正确性。尝试增加邮箱验证码功能，提高系统安全性。

* 用户登录

要求用户提供邮箱和密码，验证后登录。增加登录的验证码功能，提高系统安全性，防止恶意软件爬虫。

* 用户信息修改

用户可以修改自己的用户名和密码，密码修改需要提供原密码，原密码正确才可以修改为新密码。

* 设备位置信息查看

通过地图SDK显示设备位置信息。

* 设备配置信息修改

提供对物联网设备信息修改的功能。

* 设备统计数据查看

可视化呈现设备统计信息。

在我们设计的系统中，用户模块分为普通用户和管理用户（admin），其中普通用户可以查看到自己的设备，而管理用户可以所有存在于数据库的设备信息。

另外还有设备管理模块，这包括信息修改、注册设备、查看设备信息等等，为这两种用户提供不同的权限管理。

### 2.1.2 非功能性

* 性能需求：

1. 系统需稳定运作，避免崩溃现象。
2. 应兼容主流网页浏览器，如微信浏览器、手机端等，确保系统的可访问性。
3. 系统需支持1000名用户的同步访问能力。
4. 对用户登录及操作应迅速响应，在1秒内完成。
5. 系统应自动检测异常状况，如设备通信断开或数据库连接失败，以减少用户等待时间。
6. 页面加载时间应在1秒内，但在流量高峰期间应不超过5秒。
7. 系统维护和重启的频率不应超过每周一次。

* 输入输出需求：

1. 用户输入账户密码时，系统需验证数据的有效性并确保安全性。
2. 设备的运动轨迹和统计数据应通过直观的图表美观准确地展示。
3. 系统应通过程序减少由用户操作错误引起的问题，开发者需全面预见并最小化错误发生的可能性。
4. 系统应当在用户登录时提供验证码验证功能，提高系统安全性

* 数据管理需求：

1. 系统应有与其他系统的接口，同时维持自身的独立性和完整性，防止未授权人员进行设置、修改或访问内部数据。
2. 系统服务器软件应提供可靠的数据备份与恢复方案，以便在严重故障发生时快速恢复运行环境，并且开发者不应预留任何特殊账户和密码，确保系统安全。
3. 系统还应有加密登录和数据传输的安全措施，确保数据在系统间传输的保密性和安全性。

## 2.2 运行环境

### 2.2.1 软件环境

该系统需展现良好的负载性能，确保在高并发环境下至少支持1000名用户的同时在线访问，涵盖数据存储、网络吞吐和数据安全等方面的能力。此外，网站须适应Android和iOS等移动操作系统的终端设备，并提供必要的网络安全防护措施来保障对外服务的安全性。

在客户端兼容性方面，网站应确保在包括IE、Chrome、Firefox、Opera在内的主流浏览器上均可顺畅访问。即使用户选择其他非主流浏览器，也应保证能够访问网站的核心功能。网站的界面设计需适应移动设备，确保在手机浏览器和应用程序如微信内置的浏览器上也能提供良好的用户体验。

### 2.2.2 硬件环境

CPU：主频大于2GHz

硬盘:硬盘容量大于 200GB、硬盘转速大于等于 5400 转/分钟

内存：大于等于2G及以上

# 3. 技术选型

## 3.1 项目技术选型

本项目采用了现代的前后端分离架构，前端通过RESTful API与后端进行通信。此外，项目还包括一个独立的MQTT服务器，它负责接收模拟客户端的数据，并将数据存储到数据库中。项目的技术栈如下：

* **前端**

React：用于构建用户界面的JavaScript库。

Ant Design：一个基于React的UI组件库。

Ant Design Charts：一套基于Ant Design的图表库。

React-Router：用于在React中实现路由功能。

百度地图SDK：用于集成百度地图提供地图展示和服务。

yarn：一个快速、可靠、安全的依赖管理工具。

* **后端**

golang：轻量级、规范的后端编程语言

gin：golang流行web框架

* **数据库**

MySQL：一个广泛使用的开源关系数据库管理系统。

* **MQTT服务器**

Python 3.8：高级编程语言，用于MQTT服务器的开发。

mysql-connector：Python的MySQL驱动程序。

Paho：提供了多种语言的MQTT客户端库。

* **测试工具**

Postman：流行的api接口测试工具

该技术选型的目标是创建一个既高效又稳定的Web应用，同时也注重提供卓越的用户体验。使用React框架结合Ant Design可以快速构建出美观的用户界面。此外，采用golang的gin框架作为后端开发工具，能够加快API的开发速度并确保其稳定运行。MQTT服务器的集成进一步增加了处理设备数据通信的灵活性。在开发过程中，利用Postman进行接口测试，从而保障后端服务的数据处理准确无误。这样的技术组合有助于打造一个可靠且用户友好的Web服务平台。

## 3.2 项目主要技术介绍

* **在前端技术框架方面**，本项目选择了React结合Ant Design组件库、Ant Design Charts、React-Router、百度地图SDK以及yarn作为包管理工具作为开发栈。React是由Facebook开发的用于构建用户界面的JavaScript库，其采用虚拟DOM技术和声明式编程理念，使得Web应用的开发既高效又灵活。Ant Design是由蚂蚁金服开发的企业级UI设计语言，它提供了一整套设计资源和开发工具，旨在提升用户界面的美观度和功能性。Ant Design Charts为数据可视化提供了丰富的图表模板。百度地图SDK支持React，并提供地图组件以及点线的标注和绘制功能，适用于实现设备位置的可视化展示。
* **在后端技术框架方面**，本项目采用了golang语言流行的web框架gin，它是Go世界里最流行的Web框架，Github上有32K+star， 是基于httprouter开发的Web框架。同时它也是 中文文档齐全，简单易用的轻量级框架。
* **在数据库方面**，本项目使用了MySQL，它是一个关系型数据库管理系统，由瑞典MySQL AB 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 (Relational Database Management System，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。
* **在MQTT服务器方面**，项目使用了mosquitto作为基础，但由于mosquitto本身不具备消息存储能力，项目中使用Python脚本进行了扩展，使用Paho库接收由设备模拟器发送的消息，并将这些消息存储在数据库中。这样的处理提高了MQTT服务器在物联网通信中的应用能力和灵活性。
* **在测试工具方面**，postman是一款支持http协议的接口调试与测试工具，其主要特点就是功能强大，使用简单且易用性好 。无论是开发人员进行接口调试，还是测试人员做接口测试，postman都是首选工具之一 。

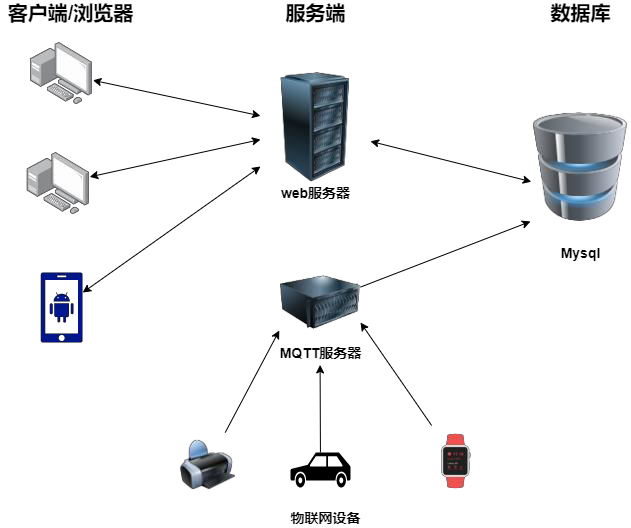
# 4. 项目架构图

系统的总体架构设计以支持灵活性和可扩展性，以满足不同终端用户的需求。客户端设计兼容PC和移动设备，确保用户可以通过多种设备访问系统。服务端架构分为后端服务器和MQTT服务器两部分，每部分都与数据库独立交互。

后端服务器负责处理客户端通过网页发送的HTTP请求，包括用户认证、数据处理、逻辑运算等，然后将结果返回给客户端。这部分通常涉及到用户界面的请求处理和业务逻辑的执行。

MQTT服务器专注于处理来自设备的报文。设备通过MQTT协议发送状态信息、传感器数据等，MQTT服务器将这些信息存储在数据库中供后端服务器进一步处理。这样的设计使得设备数据的收集和传输更加高效，也方便了设备状态监控和数据分析。

这种架构分离了用户交互和设备通信两个主要功能，使系统可以更加灵活地进行横向扩展，以支持更多的用户和设备，同时也便于维护和升级各个独立组件。



# 5. 数据库与ER图设计

！ 以下设计为初次构想产生，可能和最终版本有所区别。

## 5.1 数据表

### 5.1.1 user

此表用于存放用户信息

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 备注 |
| id | int | primary key，not null，auto\_increment |
| name | varchar(128) | not null |
| password | varchar(128) | not null |
| email | varchar(128) | not null, unique |

CREATE TABLE user (

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

name VARCHAR(128) NOT NULL,

password VARCHAR(128) NOT NULL,

email VARCHAR(128) NOT NULL UNIQUE

);

### 5.1.2 device

此表用于存放设备信息

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 备注 |
| id | int | primary key，not null，auto\_increment |
| name | varchar(128) | not null, unique |
| description | varchar(256) | not null |
| userid | int | not null,foreign key |
| kind | int | not null, 0 in default |
| last\_activate\_time | timestamp | not null |

CREATE TABLE device (

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

name VARCHAR(128) NOT NULL UNIQUE,

description VARCHAR(256) NOT NULL,

userid INT NOT NULL,

kind INT NOT NULL DEFAULT 0,

last\_activate\_time TIMESTAMP NOT NULL,

FOREIGN KEY (userid) REFERENCES user(id)

);

### 5.1.3 message

此表用于存放mqtt服务器收到的设备消息

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 备注 |
| device | varchar(128) | not null,foreign key |
| status | int | 0 -> error ，1 -> non error |
| info | varchar(128) |  |
| latitude | numeric | 0 in default |
| longitude | numeric | 0 in default |
| time\_stamp | time\_stamp |  |
| value | int | 0 in default |

CREATE TABLE message (

device VARCHAR(128) NOT NULL,

is\_alert INT,

info VARCHAR(128),

lat NUMERIC DEFAULT 0,

lng NUMERIC DEFAULT 0,

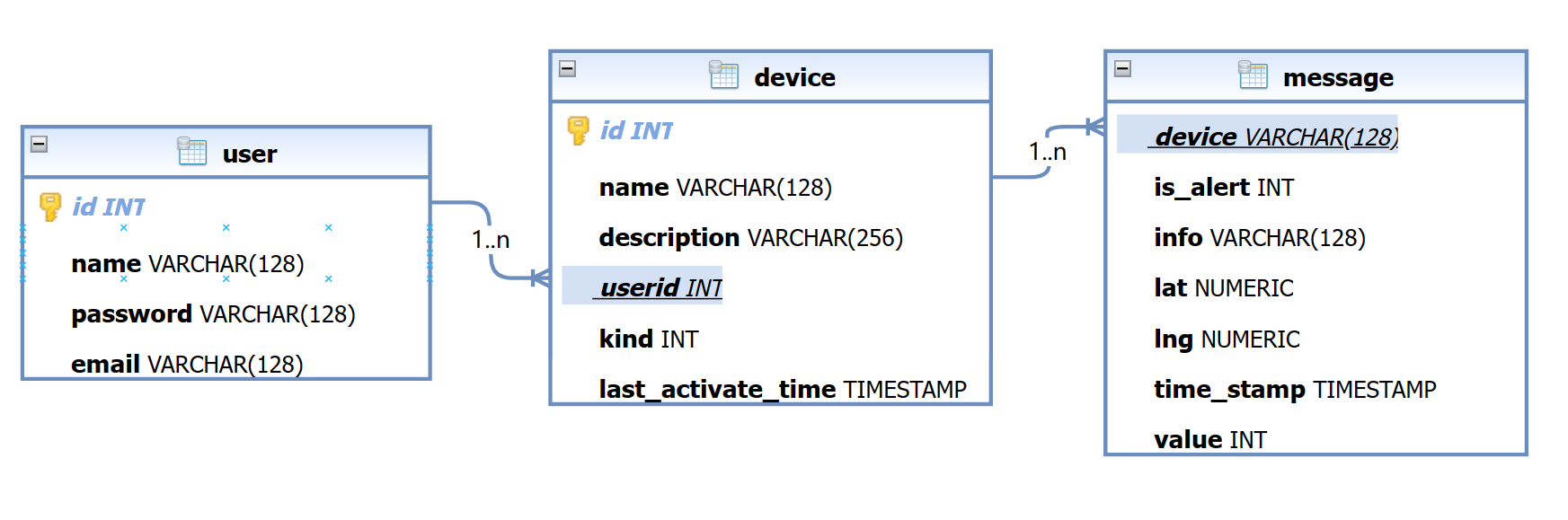
time\_stamp TIMESTAMP,

value INT DEFAULT 0,

FOREIGN KEY (device) REFERENCES device(name)

);

## 5.2 ER图



# 6. API LIST

## 6.1 用户信息接口

这些接口处理与用户表(User)相关的数据操作，所有接口URL以/user为前缀。

### 6.1.1 用户登录

URL: /user/login

参数: 包含登录表单信息的json，包含邮箱和密码。

返回: json类型，包含“code”和“token”。Code=1表示成功，0表示用户不存在，2表示账号或密码错误。

简介: 用于用户认证，登录成功后返回token。

### 6.1.2 用户注册

URL: /user/register

参数: 包含注册信息的json，如用户名、密码、邮箱等。

返回: 整型，1表示成功，0表示用户已存在。

简介: 用于用户注册，向用户表中添加新记录。

### 6.1.3 用户修改密码

URL: /user/change/password

参数: 包含新旧密码的json

返回: 整型，1表示成功，0表示失败。

简介: 用于用户修改密码。

### 6.1.4 用户修改邮箱

URL: /user/change/email

参数: 包含新旧邮箱的json

返回: 整型，1表示成功，0表示失败。

简介: 用于用户修改绑定邮箱。

## 6.2 设备配置信息接口

这些接口提供对设备配置信息的操作，URL前缀为/device。

### 6.2.1 获取单个设备信息

URL: /device/{#id}

参数: 设备ID。

返回: Device对象。

简介: 查询单台设备的信息。

### 6.2.2 修改设备配置信息

URL: /device/config

参数: 需要修改的设备信息字段json。

返回: 整型，1表示成功，0表示失败。

简介: 修改设备配置。

### 6.2.3 查询用户设备总数

URL: /device/query/all

参数: 包含用户名和token的json。

返回: 用户持有的设备总数的整型值。

简介: 查询用户所有设备数量。

### 6.2.4 查询用户持有设备列表

URL: /device/query/list

参数: 包含用户名和token的json。

返回: 设备编号数组。

简介: 查询用户持有的设备列表。

### 6.2.5 查询不同类型设备数量

URL: /device/query/kind

参数: 包含用户名和token的json。

返回: 各类型设备数量的json。

简介: 查询用户不同类型的设备数量。

### 6.2.6 查询当前活跃设备数量

URL: /device/query/active

参数: 包含用户名和token的json。

返回: 整型，表示活跃设备数量。

简介: 查询当前活跃设备。

## 6.3 设备消息相关接口

设备消息接口以/message为URL前缀，主要用于查询。

### 6.3.1 查询设备历史轨迹

URL: /message/path/{#id}

参数: 设备ID。

返回: Message数组。

简介: 获取设备历史轨迹。

### 6.3.2 查询设备历史消息

URL: /message/info/{#id}

参数: 设备ID。

返回: Message数组。

简介: 查询设备历史消息。

### 6.3.3 查询value属性变化

URL: /message/value/{#id}

参数: 设备ID。

返回: Message数组。

简介: 查询value属性历史变化。

### 6.3.4 查询当前消息总数

URL: /message/all/

参数: 用户名和token组成的Map。

返回: 整型，表示消息总数。

简介: 查询用户设备的消息总数。

# 7. 项目开发时间轴

