|  |  |  |
| --- | --- | --- |
|  | | |
| **本 科 毕 业 设 计** | | |
| **（ 2023届 ）** | | |
| **题目：基于Web的养老院云平台的**  **设计与实现** | | |
|  | | |
| **学生姓名** | **张冰如** |  |
| **学 号** | **1903010301** |  |
| **分院名称** | **软件工程** |  |
| **专业班级** | **2019级软件工程3班** |  |
| **指导教师** | **杨岚** |  |
| **填写日期** | **2023年03月** |  |

泉州信息工程学院本科毕业设计诚信承诺书

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 张冰如 | 二级学院 | 软件学院 | 指导教师 | 杨岚 |
| 学 号 | 1903010301 | 专业班级 | 2019级软件工程3班 | | |
| 毕业设计题目 | | 基于Web的养老院云平台的设计与实现 | | | |
| **学 生 承 诺**  我承诺在毕业设计活动中遵守学校有关规定，恪守学术规范。在本人的毕业设计中未剽窃、抄袭他人的学术观点、思想和成果，未篡改实验数据，未侵犯任何方面的知识产权，如有违规行为发生愿承担一切责任，接受学校的处理。  学生（签名）：  年 月 日 | | | | | |
| **指 导 教 师 承 诺**  我承诺在指导学生设计活动中遵守学校有关规定，恪守学术规范。经本人认真核查，该同学的设计中未发现有剽窃、抄袭他人的学术观点、思想和成果的现象，未发现篡改实验数据，未发现侵犯任何方面的知识产权。如果违反以上承诺，本人愿意接受学校的处理。  指导教师（签名）：  年 月 日 | | | | | |

基于Web的养老院云平台的设计与实现

# 摘 要

自改革开放以来，中国的经济高速发展，人民生活水平的不断提高，同时中国的老龄化问题也逐步突出。依据联合国对老龄化问题的规定，中国已经进入了老龄化社会，并且老龄化的速度还在不断的加快。老龄化问题急需解决，养老行业在今后势必会成为一大经济热点。本文首先分析了中国老龄化的背景，阐述了传统养老行业的痛点难点。然后从当下养老行业发展情况分析国内养老市场的竞争情况与新型养老行业在未来的发展前景。最后表明如何结合当下科学技术来提高养老行业水平，形成具有针对性的解决方案。基于此背景，本文设计并且实现了基于SpringCloud+Nacos+Rabbitmq分布式微服务架构的养老院云平台，对实际使用场景进行了需求分析，完成了平台业务与技术架构的设计。系统后台拆分成九个微服务模块且对各个微服务进行了具体的实现描述。在后台业务的支撑下，完成了包含用户、角色等公共后台服务与可视化地图监控为主的两个系统。

关键词：云开发养老；微服务架构；Spring Cloud

**Design and Implementation of Cloud Platform for Nursing Homes**

# **Abstract**

Since the reform and opening up, China's economy has developed rapidly and people's living standards have been improving. Meanwhile, the aging problem in China has become increasingly prominent. According to the Provisions of the United Nations on aging,China has entered an aging society, and the aging speed is still accelerating. The aging problem urgently needs to be solved, and the pension industry is bound to become a major economic hot spot in the future. This paper first analyzes the background of Aging in China, and expounds the pain points and difficulties of traditional pension industry. Then from the current pension industry development situation analysis of the domestic pension market competition and newpension industry in the future development prospects. Finally, it shows how to combine current science and technology to improve the level of the elderly care industry and form targeted solutions. Based on this background, this paper designed and implemented a nursing home cloud platform based on SpringCloud+Nacos+Rabbitmq distributed micro-service architecture,analyzed the actual use scenarios, and completed the design of the platform business and technical architecture. The system background is divided into nine micro-service modules and the realization of each micro-service is described in detail. Under the support of background services, two systems including public background services such as users androles and visual map monitoring are completed.

**Keywords:** Pension Industry; Microservice Architecture; SpringCloud

目录

[摘 要 I](#_Toc130047894)

**[Abstract](#_Toc130047895)** [II](#_Toc130047895)

[1 前言 1](#_Toc130047896)

[1.1 研究背景 1](#_Toc130047897)

[1.2 研究的目的与意义 1](#_Toc130047898)

[1.3 国内外研究现状 2](#_Toc130047899)

[1.4 研究内容 3](#_Toc130047900)

[2 开发工具与技术说明 4](#_Toc130047901)

[2.1 开发工具 4](#_Toc130047902)

[2.2 技术说明 4](#_Toc130047903)

[3 系统分析 5](#_Toc130047904)

[3.1 系统可行性分析 5](#_Toc130047905)

[3.2 系统角色分析 6](#_Toc130047906)

[3.3 系统角色分析 7](#_Toc130047907)

[3.4 系统用例规约 10](#_Toc130047908)

[3.5 系统主要类模型 15](#_Toc130047909)

[3.6 接口需求分析 15](#_Toc130047910)

[4 系统设计 16](#_Toc130047911)

[4.1 系统架构设计 16](#_Toc130047912)

[4.2 系统活动图设计 16](#_Toc130047913)

[4.3 系统时序图 19](#_Toc130047914)

[4.4 数据库设计 21](#_Toc130047915)

[5 系统实现 26](#_Toc130047916)

[5.1 登录功能实现 26](#_Toc130047917)

[5.2 用户管理功能实现 27](#_Toc130047918)

[5.3 角色管理功能实现 29](#_Toc130047919)

[5.4 网关管理功能实现 30](#_Toc130047920)

[5.5 IOT设备管理功能实现 31](#_Toc130047921)

[5.6 告警管理功能实现 33](#_Toc130047922)

[5.7 监控大屏功能实现 34](#_Toc130047923)

[6 系统测试 36](#_Toc130047924)

[6.1 系统测试概述 36](#_Toc130047925)

[6.2 系统的功能测试 37](#_Toc130047926)

[7 总结与展望 43](#_Toc130047927)

[7.1 总结 43](#_Toc130047928)

[7.2 安全与环保问题分析 43](#_Toc130047929)

[7.3 技术经济与成本分析 44](#_Toc130047930)

[7.4 展望 44](#_Toc130047931)

[参考文献 45](#_Toc130047932)

[致 谢 47](#_Toc130047933)

# 1 前言

## 1.1 研究背景

在这个经济发展高速腾飞的时代，中国的老龄化问题也逐步突出。根据联合国的定义，当一个国家或地区65岁及以上老年人口占总人口比例超过7%或60岁及以上老年人口占总人口比例超过10%时，该国家或地区就属于老龄化国家或地区。自2000年以来，中国已经进入了老龄化社会，并且老龄化速度在不断加快。到2019年，中国65岁及以上老年人口占总人口比例达到12.6%，60岁及以上老年人口占总人口比例达到18.1%。据预测，中国将在2032年左右进入超老龄化社会。按照这一增速，到2025年，中国60岁及以上人口将达到3亿；到2033年，将突破4亿；到2050年前，将达到5亿，占总人口的三分之一。

目前，中国的养老产业面临着供不应求的局面。为应对这一问题，政府正在大力推行社会福利社会化政策，旧有的养老服务模式也在不断变革中寻求生存。与此同时，各种新型养老服务也在不断涌现。目前，中国实行的养老模式是“9073”模式，即90%的老年人由家庭自我照顾，7%享受社区居家养老服务，3%享受机构养老服务。

随着老龄化问题的加剧，政府正在不断增加对养老机构的支持。在我国，养老机构按照承办主体分为公办、民办非企业性质和工商注册的养老机构；按照运营模式分为公办公营、公办民营和民办民营等多种类型。不同类型的养老机构享有不同的权利义务，并适用于不同的标准和政府补贴。[4]因此养老机构有动力、有需求的引入智能化的设备和服务，帮助提升服务水平和降低运营成本。

## 1.2 研究的目的与意义

不同类型、不同规模的养老机构，都面临着不同程度的运营成本压力，尤其是人力成本不断上涨的压力。同时随着生活水平的不断提高，养老退休机制逐渐完善，退休老年人均有较为可观的退休金。养老机构入住的长者及其家属又要求养老机构提供更好的服务。因此养老机构有需求、有动力地引入智能化的设备和服务来帮助提升服务水平和降低运营成本。

养老行业智慧化除了能够减少繁琐且重复性的工作、降低人力成本、减轻护工之外，还能吸引年轻人加入养老服务行业。养老服务从业者大多数为中老年人，年轻人不愿加入该行业，形成这个问题的原因有以下几点：一是现在虽然倡导就业平等，但仍存在着就业歧视，年轻人认为养老服务行业的社会地位低下，不愿加入养老服务行业；二是待遇低下，现在护工待遇处于社会平均水平或者更低，导致养老行业从业意愿低下。养老行业的智慧化，能够有效破除该局面，通过高科技设备，让养老行业转变为一个“新行业”，减轻传统护工行业枯燥、乏味的形象，为年轻人带来一丝新颖感。同时，因为国内生活水平的提高，老年人的退休金有所保障，部分机构结合科技设备加快向高端养老行业的转型，整体养老行业的收入增加，相关从业人员的待遇也能够相应的提高。

## 1.3 国内外研究现状

### 1.3.1 国内研究现状

中国进入21世纪以来，随着经济的快速增长和医疗水平的不断提高，老龄化问题也日益突出。目前，国内的智慧养老云平台尚不成熟，但发展趋势很好[。随着人口老龄化的加剧，政府和社会对智慧养老的需求和支持也在增加](https://www.qianzhan.com/analyst/detail/220/211101-c97d3dab.html" \t "https://www.bing.com/_blank)。智慧养老云平台利用信息技术，为老年人提供了多种、个性化和智能化的服务。根据统计数据，2019年，中国的智慧健康养老产业规模达到了近3.2万亿元。预计到2020年，该产业规模将超过4万亿元[。一些企业也在积极开发和推广智慧养老平台，例如九安医疗](https://new.qq.com/rain/a/20221114A046WL00" \t "https://www.bing.com/_blank)。未来，智慧养老云平台将更加注重用户体验、数据安全、服务质量等方面，打造更加完善和便捷的养老生态系统。[5]

发展老龄数字经济不仅是保障人民实现长寿梦想的手段，也是应对人口老龄化挑战和建设老年友好型社会的信心所在。目前，养老产业正面临着重大的历史机遇，信息通信技术在促进健康老龄化方面可以发挥更大的作用。基于Web的养老院云平台的设计与实现有助于建设智慧城市，确保老年人金融包容性，并为涉老服务机构提供支持。

### 1.3.2 国外研究现状

目前国外关于智慧型养老发展比国内更为迅速。国外知名企业Fountain Life建立了医疗信息化平台，提供预测、预防、个性化和数据驱动的健康服务。通过数据驱动方法能够在疾病造成伤害之前尽早发现疾病，包括癌症、心脏疾病、代谢问题和神经退行性疾病帮助老年人实现智慧养老健康监测。美国老年医疗护理初创公司Intus Care通过综合分析老年病患的各种财务、临床和行政等数据，制定和优化护理计划，以提高老年人的护理质量和效率。 医护人员短缺、护理成本高是困扰全球养老护理市场的痛点，而通过数字化方式分析和优化老年人的财务、疾病等各种数据，据此制定护理方案，这种方法不仅能够使护理更加精细化，保证护理质量，还能最大化医护资源的利用。在美国，通过数字化方式提供护理解决方案已经成为养老领域一大创业热点，但在国内这一模式并不多见，值得企业关注和挖掘。

## 1.4 研究内容

本课题设计并且实现了基于SpringCloud+Nacos+Rabbitmq分布式微服务架构的养老院云平台。系统后台拆分成九个微服务模块且对各个微服务进行了具体的实现描述，主要实现了登录、账号管理、角色管理、网关管理、IOT设备管理、告警管理、监控大屏。[智慧型养老院可以解决养老院的安全监护、健康监护、生活服务照料、员工工作效率等问题](https://zhuanlan.zhihu.com/p/441879123" \t "https://www.bing.com/_blank)[。智慧型养老院，为老人提供更加安全、舒适、便捷的养老环境和服务](https://new.qq.com/rain/a/20210803a07wfy00" \t "https://www.bing.com/_blank)。[1]

# 2 开发工具与技术说明

## 2.1 开发工具

### 2.1.1 IDEA介绍

IntelliJ IDEA是一款Java编程语言的集成开发环境，被公认为业界最佳的Java开发工具之一。它在智能代码助手、代码自动提示、重构、JavaEE支持、版本工具集成（如git、svn等）、JUnit、CVS整合、代码分析和创新的GUI设计等方面都表现出色。IntelliJ IDEA主要支持Java、Scala和Groovy等语言，并具备支持目前主流技术和框架的能力，擅长于企业应用、移动应用和Web应用的开发。

### 2.1.2 MySQL 介绍

MySQL是一种流行且免费的数据库系统，它以运行速度快而著称，被认为是目前最快的SQL语言数据库之一。与MongoDB等数据库相比，MySQL不仅具有独特的功能，还拥有成熟的体系。

## 2.2 技术说明

使用IDEA、MySQL数据库等开发软件进行开发，不断完善基于Web的养老院云平台的设计与实现的功能，利用SpringCloud框架进行开发，将业务拆分成更小粒度的架构，提高了资源的重复利用率，单个微服务的故障并不会影响其他微服务，可用性、健壮性更强，产品迭代周期更短，可扩展性更高。

# 3 系统分析

## 3.1 系统可行性分析

### 3.1.1 技术可行性

该项目使用SpringCloud框架进行开发，它利用Spring Boot的便利性简化了分布式系统基础设施的开发，并提供了诸如服务发现注册、配置中心、消息总线、负载均衡、断路器和数据监控等功能。Spring Cloud是一种成熟且经过实际考验的服务框架。项目的前端开发技术采用React，能够简化开发流程、减少代码量和降低开发成本，且React社区成熟。后台开发技术选择MySQL数据库，它是运行速度最快的SQL语言数据库。因此，该系统在技术上是可行的。[9]

### 3.1.2 经济可行性

该系统除开发平台需收费外，技术都是能够开源使用的，并且操作简单，可独立开发，并不需要人力资源，开发成本低。投入使用后利用网络进行操作，能够将现实繁杂的管理工作简单化，从而提高管理效率，节约管理成本。可见，该系统在经济上是可以实现。

### 3.1.3 操作可行性

用户只要按照菜单指示操作，就可以使用该系统的功能。该系统的界面整洁方便，功能实用高效，不需要用户掌握代码或数据库的知识。该系统采用分页加载数据的方式，减少了卡顿问题。所以，该系统在操作上是可实现的。

### 3.1.4 法律可行性

该系统由个人研发定制，不存在抄袭的纠纷，拥有所有的软件权利，所以在法律的基础上是可以进行的，不存在任何侵权的纠纷问题。可见，该系统在法律上是可以实现。

## 3.2 系统角色分析

本网站分为前端和后端两大部分，两部分分别开发互不影响。本平台为了在使用权限上对用户进行权限隔离，需要在前端实现两个平台入口，用来给不同角色登录。

（1）公共平台网页端：该平台供平台管理员使用，是养老云平台极其重要的权限管理端，是新增机构用户、设置角色、配置路由、运维等操作的主要场景。

（2）业务平台网页端：该平台面向机构人员使用，是物联设备管理、物联设备标记、大屏监控等的主要场景。

### 3.2.1 平台管理员描述

平台管理员涉及到的场景有：进行登录公告平台场景、进行注册场景、进行用户管理场景、进行角色管理场景描述、进行网关管理场景。具体描述如下：

（1）平台管理员进行登录公告平台场景描述：平台管理员进入公共平台登录页面，通过输入平台管理员专属的账号密码，点击登录按钮就可以进入到后台页面。也可以点击注册按钮进行跳转到注册界面。

（2）平台管理员进行注册场景描述：在该模块中用户通过输入用户姓名、用户名、手机号、密码等相应的个人信息，且保证手机号码唯一性的格式情况下，点击注册按钮完成平台管理员注册。

（3）平台管理员进行用户管理场景描述：平台管理员进入账号管理模块，可以对机构用户或者其他平台管理员进行新增并配置用户角色信息，可以对用户信息进行修改，查询用户，删除用户。

（4）平台管理员进行角色管理场景描述：在该模块中平台管理员可以创建或者修改角色，对角色进行资源授权，配置角色资源的可访问路径。[7]

（5）平台管理员进行网关管理场景描述：在该模块中平台管理员可以进行管理网关url路由，使用网关做智能路由转发（服务发现，负载均衡），只有通过配置好的路由才能够访问到其他的微服务，对所有经过网关的请求进行权限校验，过滤恶意请求。并且能通过只监控经过网关的请求，获取对应的性能指标。

### 3.2.2 机构用户场景描述

机构用户涉及到的场景有：登录业务平台场景、进入IOT设备维护场景、进入IOT设备标记场景、进入IOT设备告警处理场景、进入查看可视化监控大屏场景。具体描述如下：

（1）机构用户登录业务平台场景描述：机构用户无法自己注册账号，只能通过平台管理者录入账号信息，完成录入后的机构用户可以使用账号密码，点击登录按钮就可以进入到后台页面进入业务平台登录页面。

（2）机构用户进入IOT设备维护场景描述：机构用户进入物联网设备维护，可以查看智能IOT设备列表，对IOT设备信息的注册修改功能，查看IOT设备的运行情况。

（3）机构用户进入IOT设备标记场景描述：机构用户可以对在物联设备维护中注册好的IOT设备，通过设备标记功能在可视化地图上进行标记，从而能够在大屏监控地图上显示出来。

（4）机构用户进入IOT设备告警处理场景描述：机构用户可以进行告警处理，可查看需要处理的智能IOT设备，IOT设备发出告警消息后，需要在告警处理界面，消除告警警告。同时可以查看历史IOT设备的告警日志。

（5）机构用户进入查看可视化监控大屏场景描述：机构用户可在监控大屏中，看到在地图上显示各类设备的运行状态与告警消息通知。

## 3.3 系统角色分析

本系统主要角色主要分为平台管理者和机构用户两类，使用用例图进行说明。

平台管理员所具备的操作有：平台管理员通过输入用户名和密码进行登录功能，注册功能，进行用户管理包括新建机构用户等，进行角色管理，进行网关管理。如图3.2所示。

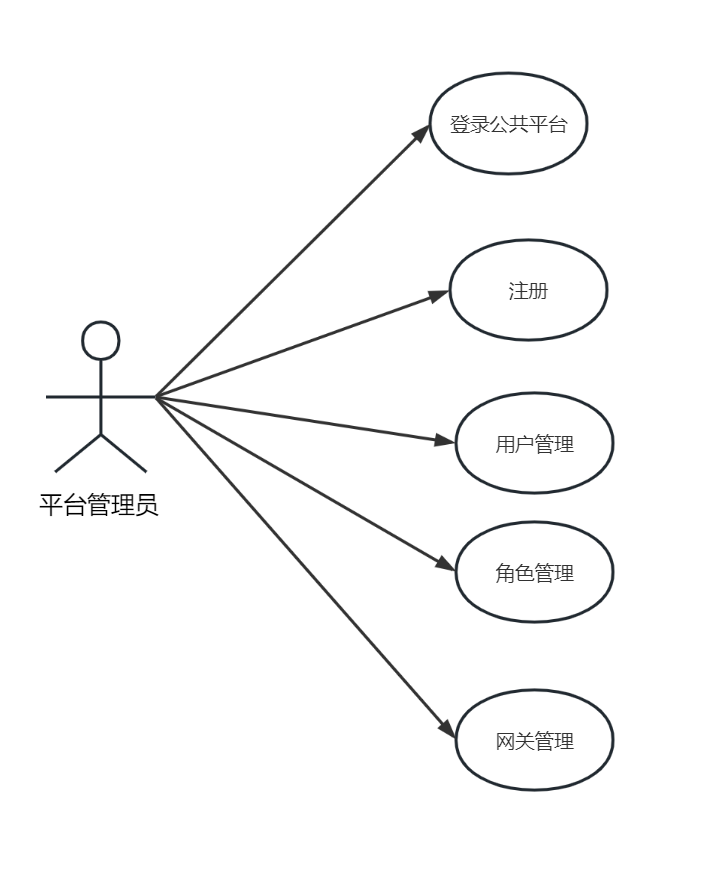


图3.2 平台管理员用例图

机构用户具有的操作有：机构用户能输入用户名密码登录功能，对IOT设备进行维护，对IOT设备进行标记，查看IOT设备告警信息，对IOT设备进行消警处理，查看监控大屏功能。如图3.1所示。

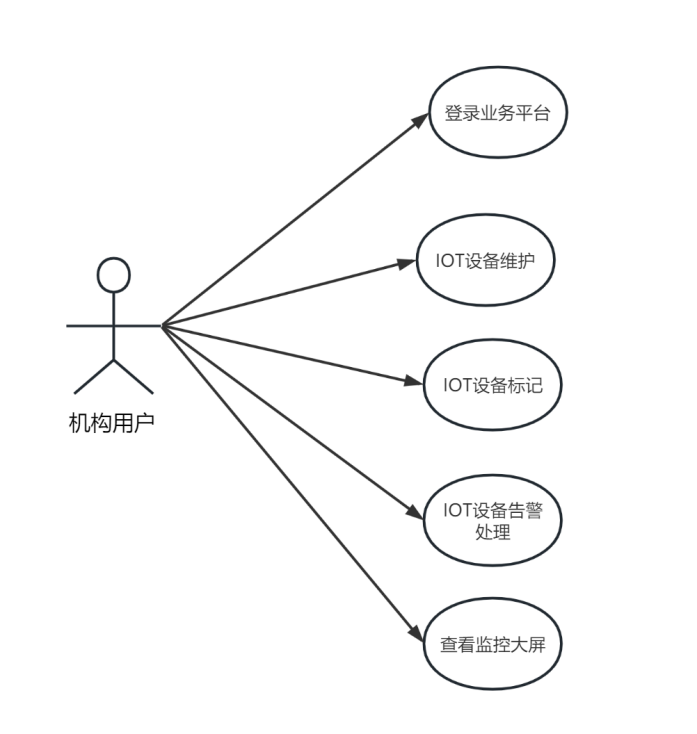


图3.1 机构用户用例图

## 3.4 系统用例规约

### 3.4.1 平台管理员系统用例规约

平台管理员登录模块的用例规约，见表3.1所示。

表3.1 登录模块系统用例规约表

|  |
| --- |
| 用例编号：0001 用例名称：登录系统 作者：张冰如 |
| 用例描述：本用例是平台管理员登录公共平台，在输入正确的账号和密码的情况下点击登录按钮，进入到对应的操作界面 |
| 执行者：平台管理员 |
| 前置条件：登录用户已经进行注册，并且成功持久化到数据库。 |
| 后置条件：登录者在完成相应的操作后，系统根据其登录者对应的身份跳转到不同的接口 |
| 基本路径：  （1）平台管理员进入公共平台的登录接口  （2）进行用户登录，填写账号密码等相关操作 |
| 备选路径：暂无 |
| 业务规则：系统所有平台管理员注册用户都能使用本用例 |

注册模块用例规约，见表3.2所示。

表3.2 注册模块用例规约表

|  |
| --- |
| 用例编号：0002 用例名称：用户注册 作者：张冰如 |
| 用例描述：本用例是平台管理员通过注册，进行创建新的账号密码 |
| 执行者：平台管理员 |
| 前置条件：平台管理员进入注册链接 |
| 后置条件：平台管理员注册成功后，可以进行登录 |
| 基本路径：  （1）在注册页面进行注册  （2）后台处理注册信息的创建  （3）注册成功进入登录界面进行登录 |
| 备选路径：  （1）以平台管理员身份登录公共平台  （2）进入账号管理  （3）添加平台管理员或者机构人员 |
| 业务规则：供使用者进行注册 |

用户管理模块用例规约，见表3.3所示。

表3.3 用户管理模块用例规约表

|  |
| --- |
| 用例编号：0003 用例名称：账号管理 作者：张冰如 |
| 用例描述：本用例是平台管理员进行账号管理 |
| 执行者：平台管理员 |
| 前置条件：平台管理员通过登录公共平台 |
| 后置条件：进入账号管理模块，进行新增、修改、删除用户 |
| 基本路径：  （1）平台管理员登录公共平台，进入账号管理  （2）对用户进行新增、修改、删除操作 |
| 备选路径：无 |
| 业务规则：供平台管理员管理机构用户或其他平台管理员使用 |

角色管理模块用例规约，见表3.4所示。

表3.4 角色管理模块用例规约表

|  |
| --- |
| 用例编号：0004 用例名称：角色管理 作者：张冰如 |
| 用例描述：本用例是平台管理员，进行角色管理 |
| 执行者：平台管理员 |
| 前置条件：平台管理员通过登录公共平台 |
| 后置条件：进入角色管理模块，对系统角色进行新增、修改、删除、为角色配置访问权限 |
| 基本路径：  （1）平台管理员登录公共平台，进入角色管理  （2）对系统角色进行新增、修改、删除操作、为角色配置访问权限 |
| 备选路径：无 |
| 业务规则：供平台管理员角色使用 |

网关管理模块用例规约，见表3.5所示。

表3.5 网关管理模块用例规约表

|  |
| --- |
| 用例编号：0005 用例名称：网关管理 作者：张冰如 |
| 用例描述：本用例是平台管理员，进行网关管理 |
| 执行者：平台管理员 |
| 前置条件：平台管理员通过登录公共平台 |
| 后置条件：进入网关管理模块，对网关进行增加、修改、删除网关策略 |
| 基本路径：  （1）平台管理员登录公共平台，进入角色管理  （2）对网关进行新增、修改、删除操作等操作 |
| 备选路径：无 |
| 业务规则：供平台管理员管理网关使用 |

### 3.4.2 机构系统用例规约

机构用户登录模块的用例规约，见表3.6所示。

表3.6 登录模块系统用例规约表

|  |
| --- |
| 用例编号：0006 用例名称：登录系统 作者：张冰如 |
| 用例描述：本用例是机构用户登录业务平台，在输入正确的账号和密码的情况下点击登录按钮，进入到对应的操作界面 |
| 执行者：机构用户 |
| 前置条件：登录用户已经进行注册，并且成功持久化到数据库。 |
| 后置条件：登录者在完成相应的操作后，系统根据其登录者对应的身份跳转到不同的接口 |
| 基本路径：  （1）机构用户进入业务平台的登录接口  （2）进行用户登录，填写账号密码等相关操作 |
| 备选路径：暂无 |
| 业务规则：系统所有机构用户注册用户都能使用本用例 |

IOT设备维护模块用例规约，见表3.7所示。

表3.7 IOT设备维护用例规约表

|  |
| --- |
| 用例编号：0007 用例名称：IOT设备维护 作者：张冰如 |
| 用例描述：本用例是机构用户进行设备管理，可以对设备进行设备查看，设备删除，设备信息修改 |
| 执行者：机构用户 |
| 前置条件：机构用户登录成功 |
| 后置条件：进入设备维护，设备查看，设备删除，设备信息修改 |
| 基本路径：  （1）机构用户登录业务平台  （2）进入设备管理模块 |
| 备选路径：无 |
| 业务规则：供机构账号管理IOT设备使用 |

IOT设备标记模块用例规约，见表3.8所示。

表3.8 IOT设备维护用例规约表

|  |
| --- |
| 用例编号：0008 用例名称：IOT设备标记 作者：张冰如 |
| 用例描述：本用例是机构用户进行设备管理，可以对设备标记 |
| 执行者：机构用户 |
| 前置条件：机构用户登录成功 |
| 后置条件：进入设备标记，对设备在地图的指定位置进行标记 |
| 基本路径：  （1）机构用户登录业务平台  （2）进入设备标记模块 |
| 备选路径：无 |
| 业务规则：供机构账号进行IOT设备标记使用 |

IOT设备告警模块用例规约，见表3.9所示。

表3.9 IOT设备维护用例规约表

|  |
| --- |
| 用例编号：0009 用例名称：IOT设备告警 作者：张冰如 |
| 用例描述：本用例是机构用户进行设备管理，可以对设备告警进行告警处理，告警历史查看 |
| 执行者：机构用户 |
| 前置条件：机构用户登录成功 |
| 后置条件：对设备告警进行告警处理，告警历史查看 |
| 基本路径：  （1）机构用户登录业务平台  （2）进入设备告警模块 |
| 备选路径：无 |
| 业务规则：供机构账号进行IOT设备告警处理，查看使用 |

监控大屏模块用例规约，见表3.10所示。

表3.10 监控大屏模块用例规约表

|  |
| --- |
| 用例编号：0010 用例名称：监控大屏 作者：张冰如 |
| 用例描述：本用例是机构用户查看大屏监控 |
| 执行者：机构用户 |
| 前置条件：机构用户登录成功 |
| 后置条件：进入监控大屏模块 |
| 基本路径：  （1）机构用户登录业务平台  （2）进入监控大屏 |
| 备选路径：无 |
| 业务规则：供机构用户查看监控大屏使用 |

## 3.5 系统主要类模型

本系统的主要类模型，主要包括User类（机构用户类），Admin类（平台管理员类），Device类（设备类），Alarm类（告警类），Map类（设备位置信息类），MuneVO(菜单信息)以及Role类（角色信息类）。如图3.4所示。

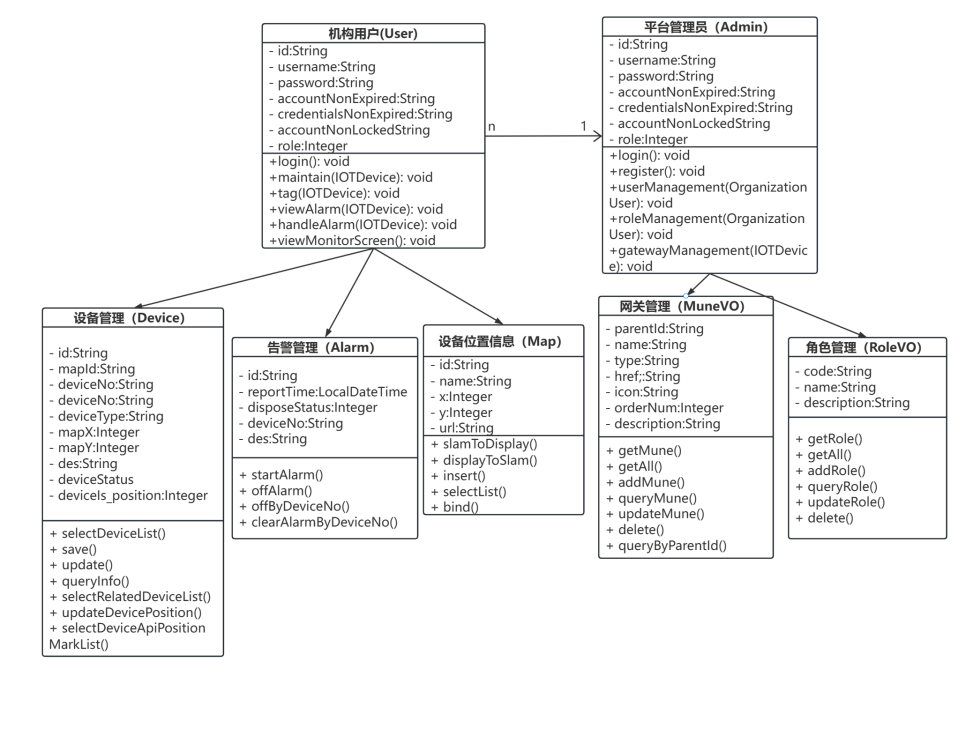


图3.4 系统主要类模型

## 3.6 接口需求分析

前端请求后端接口所使用的传输方式，采用的是目前非常流行的XML和JSON两种数据格式。前者规范性相对来说比较强，后者相对来说比较轻量简便。在基于Web的养老院云平台的设计与实现过程中，全部采用统一的JSON格式返回所需的相应数据。因为JSON更加简单纯洁，更加的便于系统的开发和后期的阅读。对系统的机器解析来说也是较为简洁方便的，因此极大的提升了数据的传输速度效率。

# 4 系统设计

## 4.1 系统架构设计

在结合实际业务需求和系统性能、可靠性、稳定性、扩展性等方面的要求，在对比了单体架构、分布式架构、微服务架构后，系统以SpringCloud + Nacos作为分布式微服务架构基础，同时结合Redis缓存、MySQL数据库、RabbitMQ消息队列等技术，搭建形成一个具有高可用性、高稳定性、易扩展性的系统架构。技术架构选型均为当下主流的架构设计。前端工程上同时选用了React，社区成熟的低代码开发框架，能够快速展示数据。

## 4.2 系统活动图设计

### 4.2.1 平台管理员活动图

基于Web的养老院云平台的平台管理员，可以进行用户管理，用户管理包含添加用户、编辑用户、查看用户、检索用户。平台管理员还进行角色管理，包含添加角色、编辑角色、查看角色权限、检索角色。平台管理员进行网关管理时，可以新建网关、编辑网关、查看网关、检索网关。如平台管理员活动图4.1和图4.2所示：

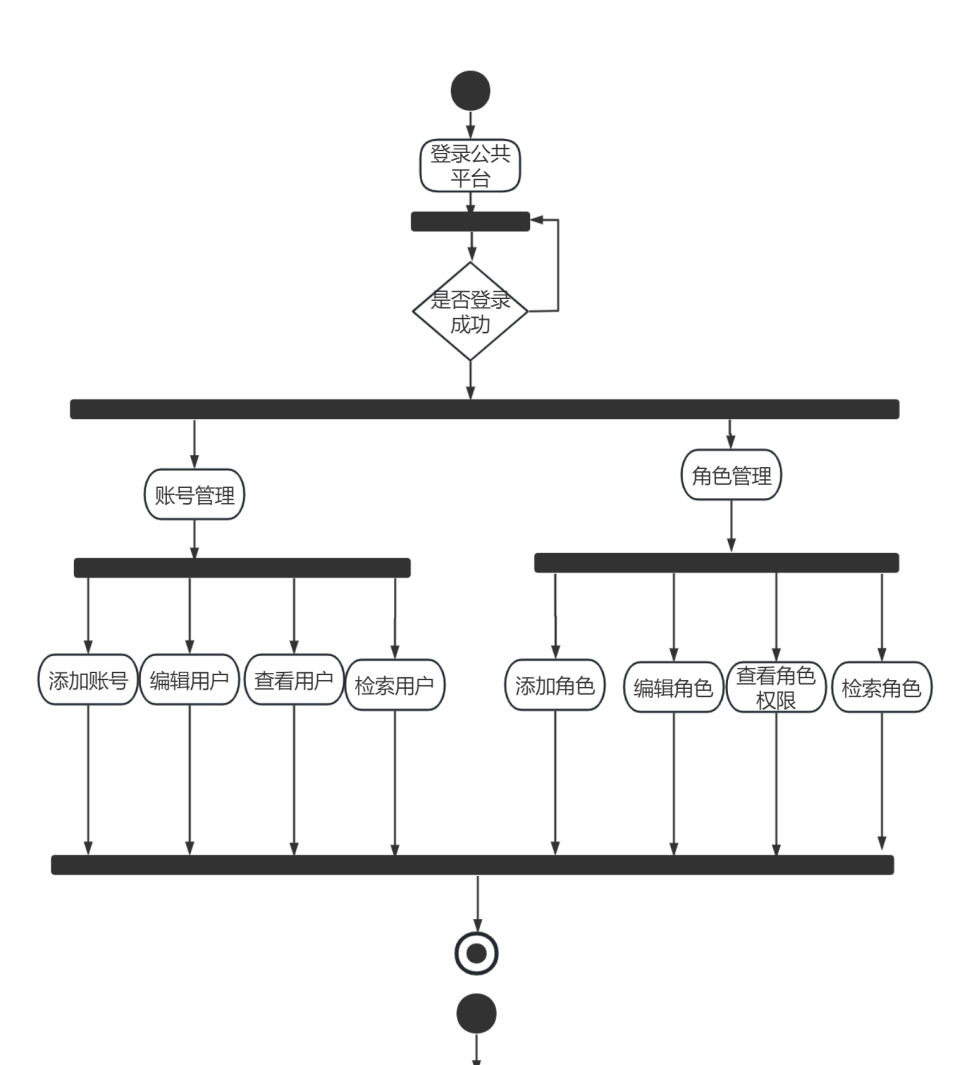
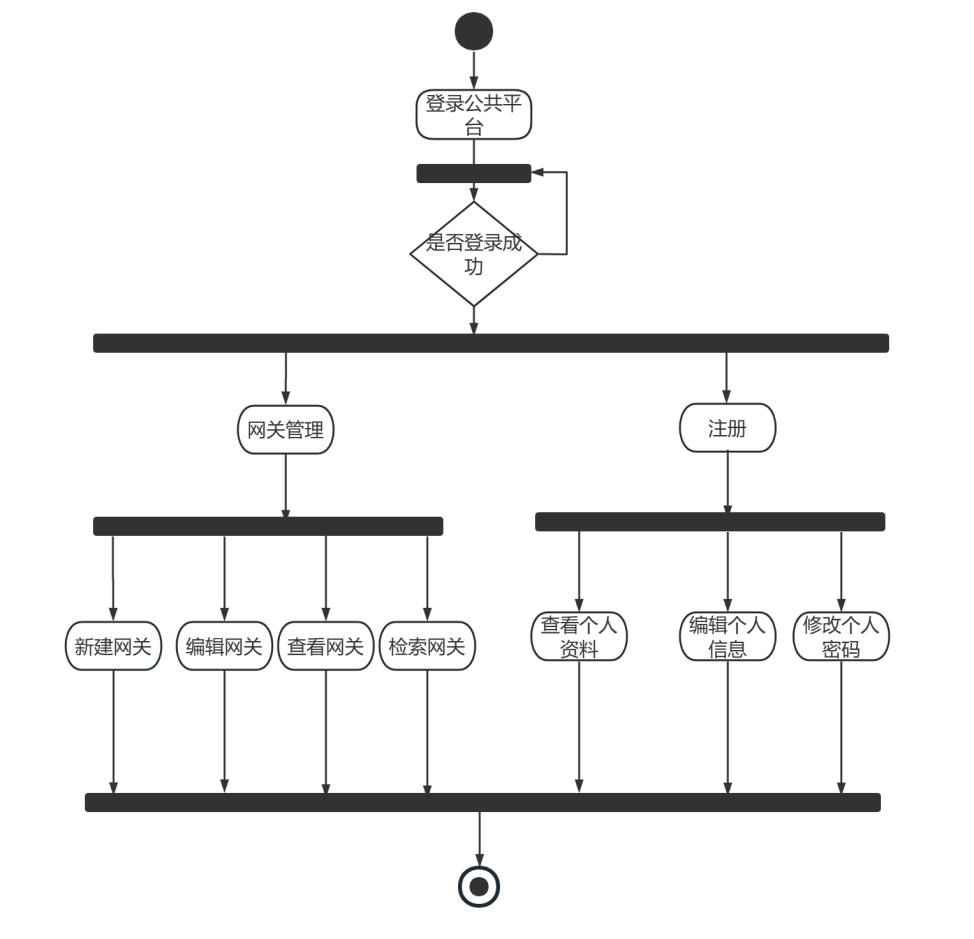


图4.1 平台管理员活动图（1）

图4.2 平台管理员活动图（2）

### 4.2.2 机构用户活动图

机构用户可以进行查看监控大屏，在监控大屏中可以看到IOT设备的位置，查看IOT设备告警列表。对IOT设备进行管理时可以对IOT设备进行新增、删除、编辑IOT设备信息、对IOT设备位置进行标记、进行告警查看和消警处理。如机构用户活动图4.3所示：

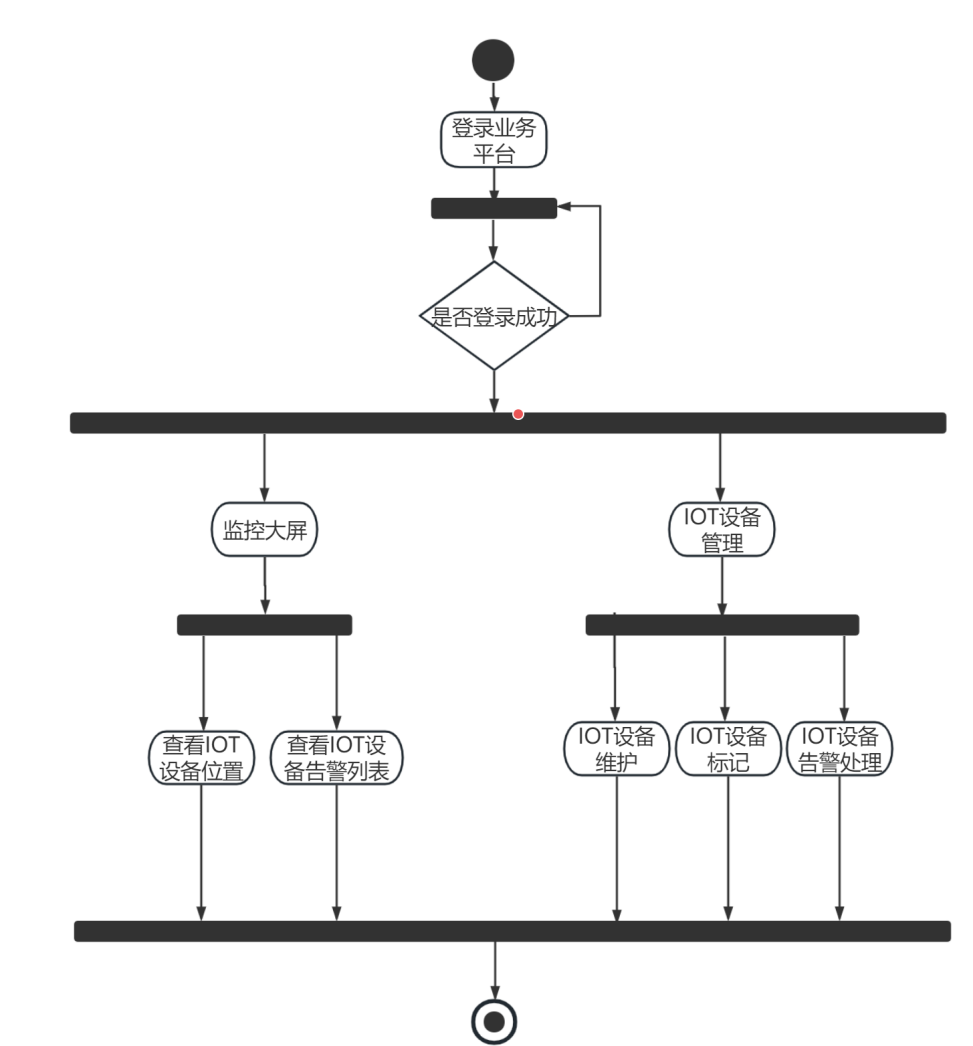


图4.3 机构用户活动图

## 4.3 系统时序图

### 4.3.1 平台管理员时序图

平台管理员通过输入姓名、用户名、手机号、密码进行注册，注册时会进行验证，验证成功则持久化到数据库。注册成功后可以使用注册的账户进行登录。进入系统后，可以进入用户管理模块，进行添加用户，删除用户，若成功则更新数据库，查询用户，检索等操作。平台管理员进入角色管理则看对角色进行新增角色，编辑角色，修改角色成功则更新数据库，查询角色等操作。进入网关管理，可以进行网关的新建、编辑网关成功则更新数据库，检索网关，查看网关。

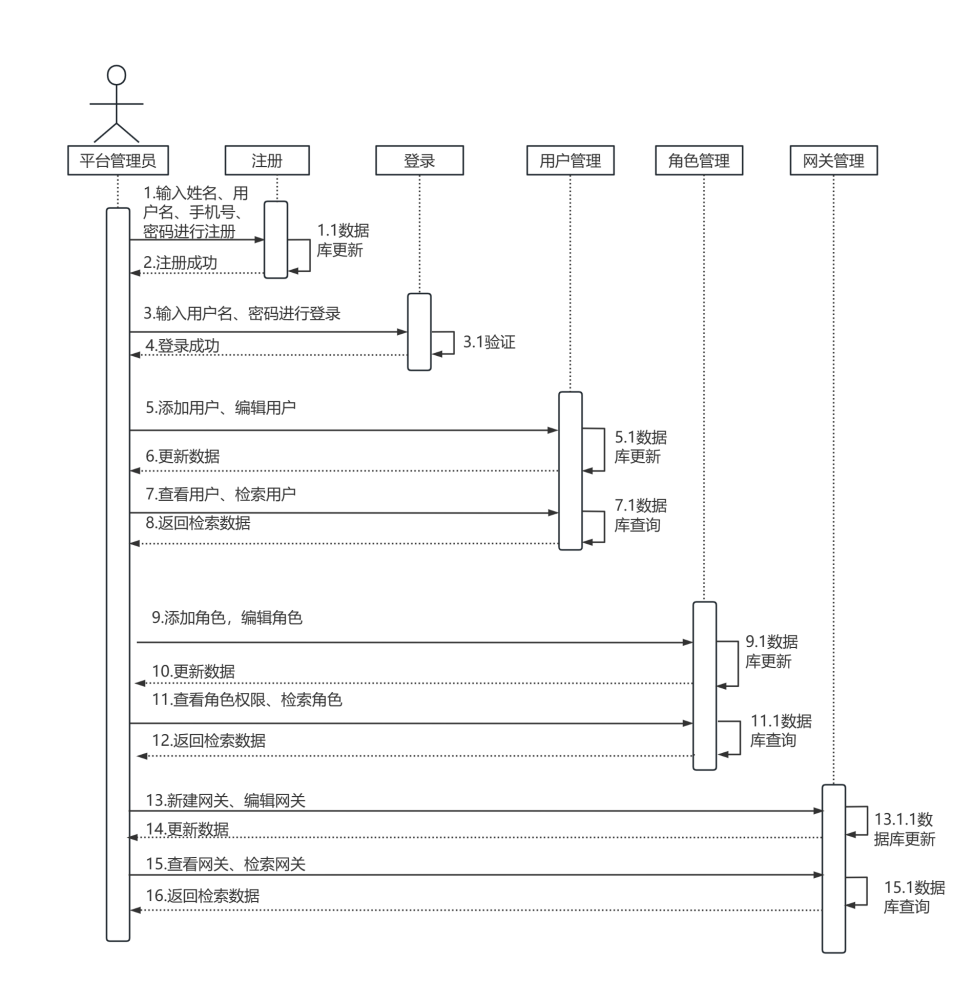


图4.4 平台管理员时序图

### 4.3.2 机构用户时序图

机构用户通过输入用户名和密码登录系统，验证成功后则进入系统。机构用户进入系统后可以进入监控大屏，查看到IOT设备的位置，查看IOT设备的告警列表。进入IOT设备管管理，对IOT设备进行维护、对IOT设备进行标记、对IOT设备进行告警处理。

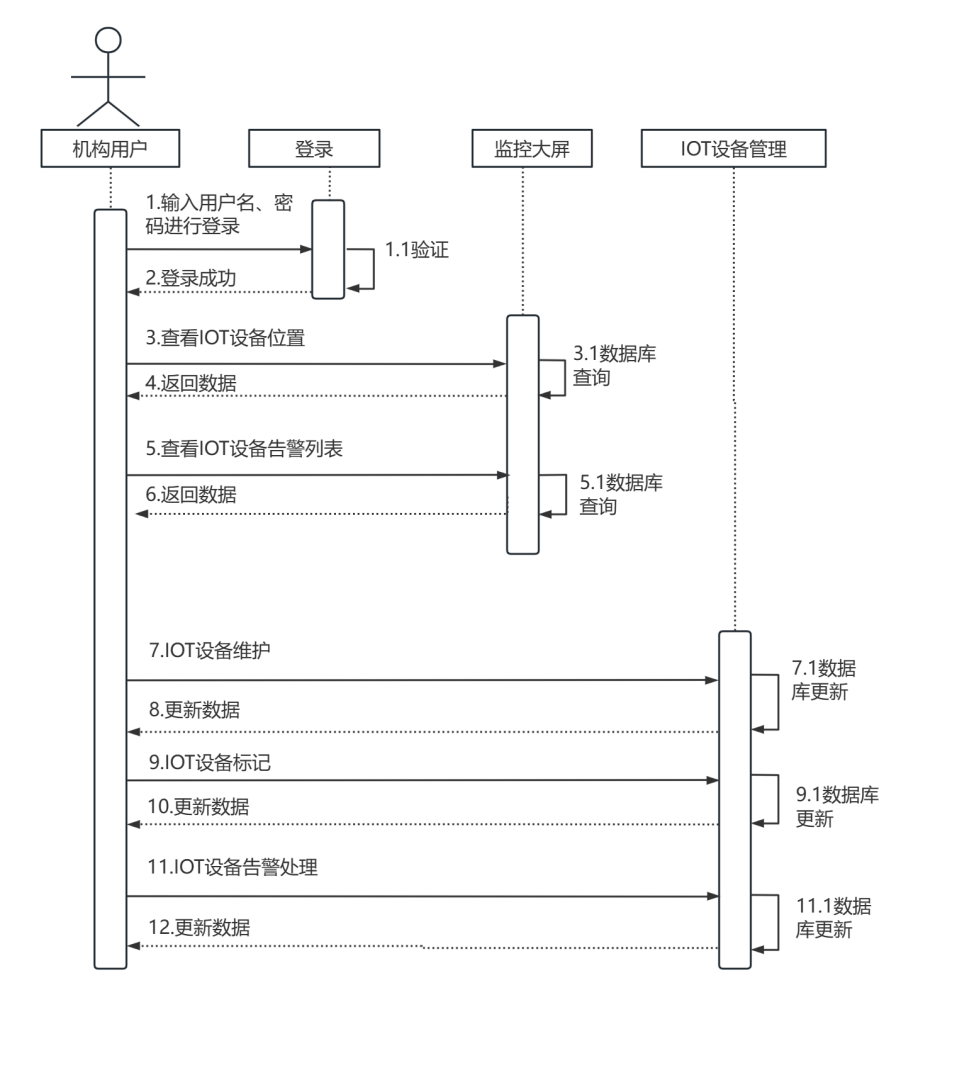


图4.5 机构用户时序图

## 4.4 数据库设计

用户表详细信息基本数据表分别有以下几个属性，分别是用户ID、用户名、用户密码、账号是否过期、密码是否过期、账号是否锁定，如表4.1所示。

表4.1 用户表详细信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 允许 | 约束 | 说明 |
| 1 | id | varchar(20) | 否 | 主键 | 用户ID |
| 2 | username | varchar(100) | 否 |  | 用户名 |
| 3 | password | varchar(100) | 否 |  | 用户密码 |
| 4 | account\_non\_expired | int(1) | 是 |  | 账号是否过期 |
| 5 | credentials\_non\_expired | int(1) | 是 |  | 密码是否过期 |
| 6 | account\_non\_locked | int(1) | 是 |  | 账号是否锁定 |

用户角色表详细信息基本数据表分别有以下几个属性，分别是角色ID、角色代码、图片，如表4.2所示。

表4.2 角色表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 允许空 | 约束 | 说明 |
| 1 | id | string | 否 | 主键 | 角色ID |
| 2 | code | number | 否 |  | 角色代码 |
| 3 | picUrl | string | 否 |  | 图片 |

菜单表详细信息基本数据表分别有以下几个属性，分别是菜单ID、菜单代码、菜单类型、菜单名称、菜单路径，如表4.3所示。

表4.3 菜单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 允许空 | 约束 | 说明 |
| 1 | id | varchar(20) | 否 | 主键 | 菜单ID |
| 2 | code | varchar(100) | 否 |  | 菜单代码 |
| 3 | type | varchar(100) | 否 |  | 菜单类型 |
| 4 | name | varchar(100) | 否 |  | 菜单名称 |
| 5 | url | varchar(256) | 否 |  | 菜单路径 |

用户与角色关联表数据表分别有以下几个属性，分别是ID、用户ID、角色ID，如表4.4所示。

表4.4 用户与角色关联表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 允许空 | 约束 | 说明 |
| 1 | id | varchar(20) | 否 | 主键 | ID |
| 2 | user\_id | varchar(20) | 否 |  | 用户ID |
| 3 | role\_id | varchar(20) | 否 |  | 角色ID |

角色与网关关联表数据表分别有以下几个属性，分别是ID、菜单ID、角色ID，如表4.5所示。

表4.5 角色与菜单关联表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 允许空 | 约束 | 说明 |
| 1 | id | varchar(20) | 否 | 主键 | ID |
| 2 | resource\_id | varchar(20) | 否 |  | 菜单ID |
| 3 | role\_id | varchar(20) | 否 |  | 角色ID |

信息数据库只存放网关表，主要支撑网关的管理服务，提供网关的动态路由的配置功能分别有以下几个属性，分别是ID、路由ID、uri路径、断言、过滤器、备注，如表4.6所示。

表4.6 网关表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 允许空 | 约束 | 说明 |
| 1 | id | varchar(20) | 否 | 主键 | ID |
| 2 | route\_id | varchar(100) | 否 |  | 路由ID |
| 3 | uri | varchar(100) | 否 |  | uri路径 |
| 4 | predicates | text | 否 |  | 断言 |
| 5 | filters | text | 是 |  | 过滤器 |
| 6 | description | varchar(100) | 是 |  | 备注 |

设备表详细信息表分别有以下几个属性，分别是ID、设备编号、设备名称、设备类型、设备所在X轴、设备所在Y轴、描述、设备状态、设备标记状态、地图ID，如表4.7所示。

表4.7 设备表详细信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 允许空 | 约束 | 说明 |
| 1 | id | int(20) | 否 | 主键 | ID |
| 2 | map\_id | varchar(20) | 否 |  | 设备编号 |
| 3 | device\_no | varchar(256) | 否 |  | 设备名称 |
| 4 | device\_no | varchar(256) | 否 |  | 设备类型 |
| 5 | device\_type | varchar(256) | 否 |  | 设备所在X轴 |
| 6 | map\_x | int(11) | 是 |  | 设备所在Y轴 |
| 7 | map\_y | int(11) | 是 |  | 描述 |
| 8 | des | varchar(256) | 是 |  | 设备状态 |
| 9 | device\_status | int(1) | 是 |  | 设备标记状态 |
| 10 | device\_is\_position | int(1) | 是 |  | 地图ID |

设备位置信息表详细信息表分别有以下几个属性，分别是主键、设备位置名称、设备位置X轴宽、设备位置Y轴高、设备位置地址，如表4.8所示。

表4.8 设备位置信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 允许空 | 约束 | 说明 |
| 1 | id | varchar(20) | 否 | 主键 | 主键 |
| 2 | name | varchar(100) | 否 |  | 设备位置名称 |
| 3 | x | int(11) | 否 |  | 设备位置X轴宽 |
| 4 | y | int(11) | 否 |  | 设备位置Y轴高 |
| 4 | url | varchar(100) | 否 |  | 设备位置地址 |

告警日志表详细信息表分别有以下几个属性，分别是主键ID、上报时间、处理状态、设备编号、描述，如表4.9所示。

表4.9 告警表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 允许空 | 约束 | 说明 |
| 1 | id | varchar(20) | 否 | 主键 | 主键ID |
| 2 | report\_time | datetime | 否 |  | 上报时间 |
| 3 | dispose\_status | int(1) | 否 |  | 处理状态 |
| 4 | device\_no | varchar(100) | 否 |  | 设备编号 |
| 5 | des | varchar(256) | 是 |  | 描述 |

# 5 系统实现

本章主要根据上述的需求分析介绍系统的详细设计与实现，系统主要模块为登录模块、账号管理模块、角色管理模块、路由管理模块、IOT设备管理模块、监控大屏模块。

## 5.1 登录功能实现

无论什么系统的开发，一定都会有登录的功能，这是最基础的功能。登录根据角色配置了两个登录页面，一个是平台管理员登录页面，即公共服务平台。一个是机构用户登录页面，即业务平台。用户输入账号密码进行验证，账号和密码都正确即为验证成功，后台跳转至相应的主页以显示菜单列表。[11]其公共管理平台登录界面如下图5.1所示，业务管理平台登录界面如图5.2所示。

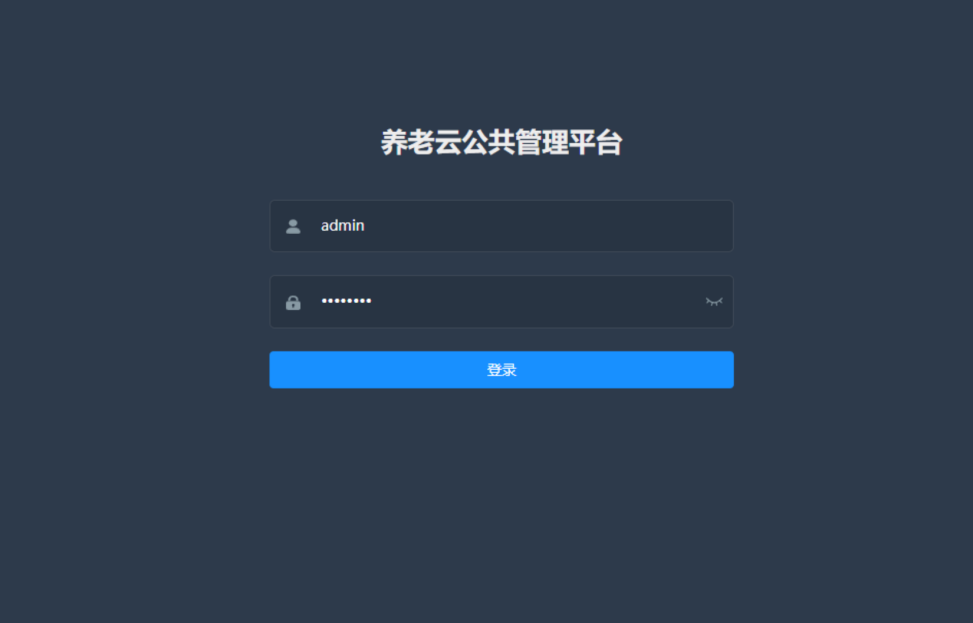


图5.1 公共管理平台登录入口

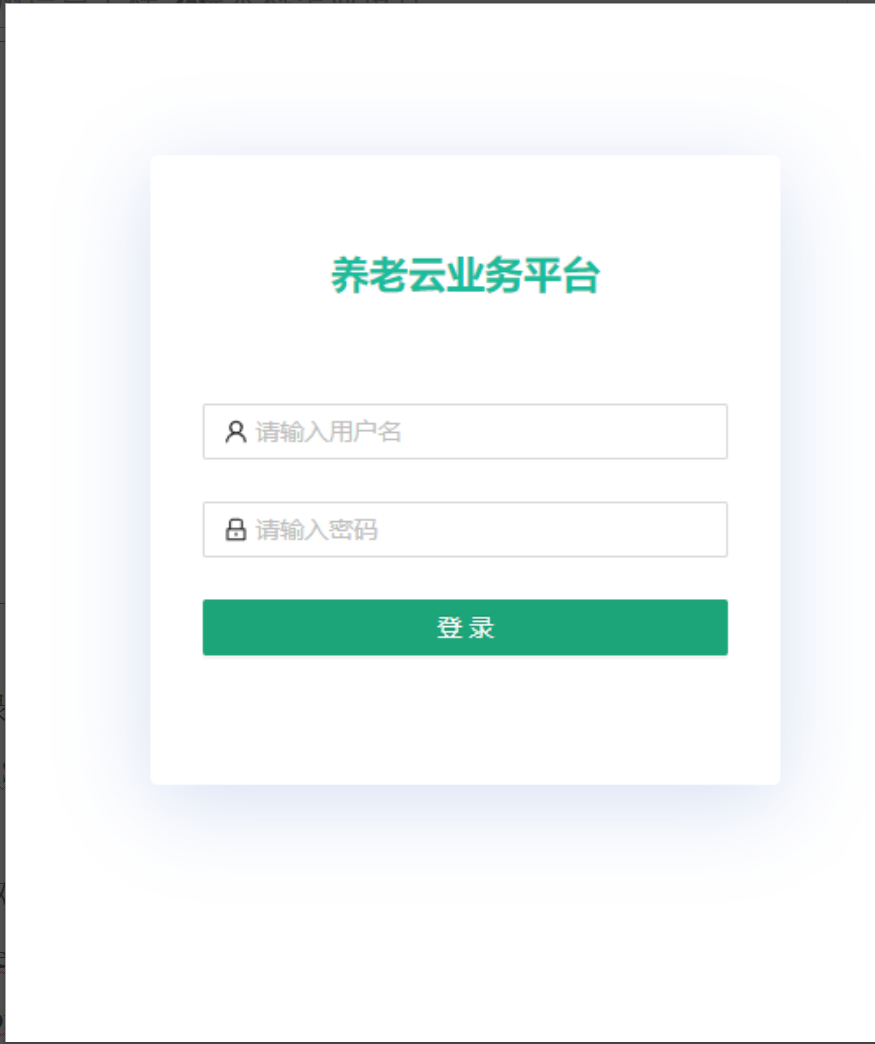


图5.2 业务平台登录入口

平台都需要用户先进行登录，首先用户点击登录按钮，通过token接口认证授权，获取返回的token数据，存到localStorage当中。实现前端自定义拦截方法request，前端所有接口请求的时候都会收到拦截，在请求头header当中拼接上token信息，实现权限认证。[10]其中token加密采用的是jwt对称加密算法，核心代码如下所示：

public JwtAccessTokenConverter accessTokenConverter() {

JwtAccessTokenConverter converter = new JwtAccessTokenConverter();

converter.setSigningKey(signingKey);

return converter;

}

并且token内容可通过继承spring-security-oauth2的TokenEnhancer类，重写enhance方法来实现自定义token信息，核心代码如下所示：

public JwtAccessTokenConverter accessTokenConverter() { JwtAccessTokenConverter converter = new JwtAccessTokenConverter(); converter.setSigningKey(signingKey); return converter;

}

## 5.2 用户管理功能实现

用户管理只给平台管理员使用，应用场景为平台管理员登录系统后，能够按照养老院方面注册机构用户，设置对应的角色。并且能对用户进行创建、搜索、修改、删除操作如图5.3所示。

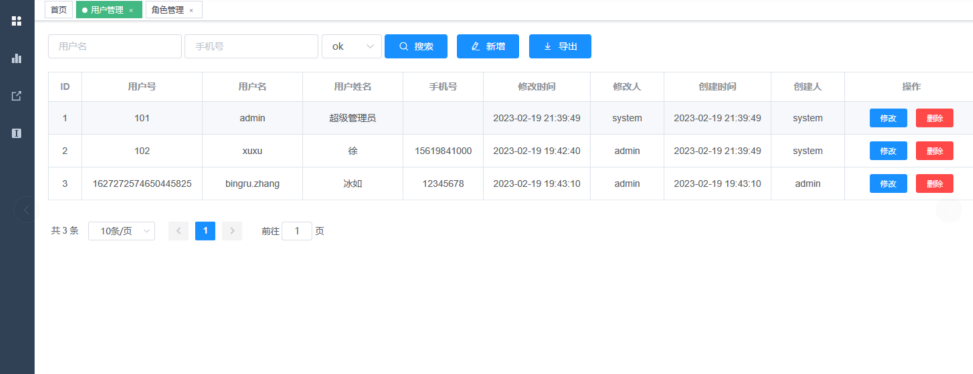


图5.3 用户管理界面

平台管理员在新增用户时配置用户角色信息，或是要移除某个用户权限，直接修改该的角色，便能让该用户无法访问资源。

用户新增核心代码如下所示：

@Transactional

public boolean add(User user) {

if (StringUtils.isNotBlank(user.getPassword()))

user.setPassword(passwordEncoder().encode(user.getPassword()));

boolean inserts = this.save(user);

userRoleService.saveBatch(user.getId()， user.getRoleIds());

return inserts;

}

用户修改核心代码如下所示：

@Transactional  
@CacheInvalidate(name = "user::"， key = "#user.id")  
public boolean update(User user) {  
 if (StringUtils.isNotBlank(user.getPassword()))  
 user.setPassword(passwordEncoder().encode(user.getPassword()));  
 boolean isSuccess = this.updateById(user);  
 userRoleService.saveBatch(user.getId()， user.getRoleIds());  
 return isSuccess;  
}

## 5.3 角色管理功能实现

角色管理只给平台管理员使用，应用场景为平台管理员在角色管理中创建或者修改角色，对角色进行资源授权，配置角色资源的可访问路径。[9]角色与资源的关系体现在角色与资源关联表中，一个角色可以拥有多个资源访问权限，如图5.4所示。

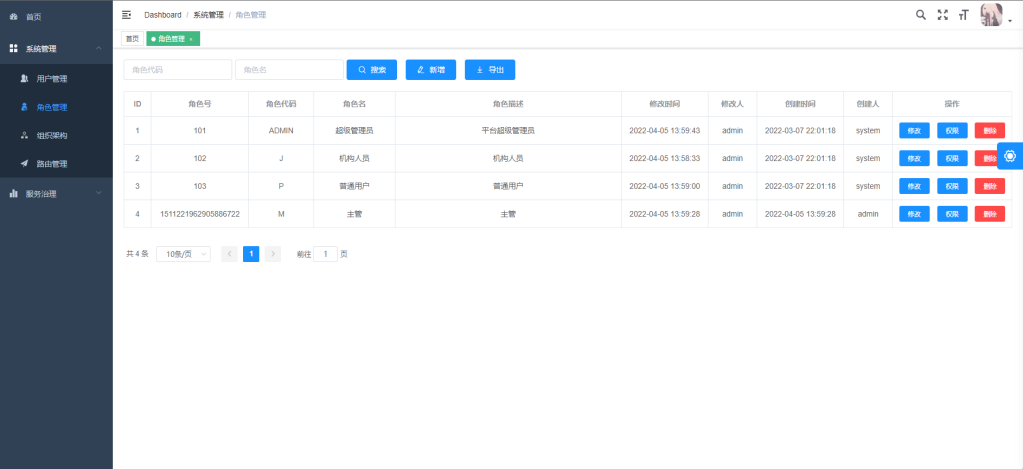


图5.4 角色管理界面

一种角色想要拥有资源访问权限需要平台管理员修改配置角色所拥有资源，之后该角色才拥有对应的资源访问权限。操作如图5.5所示。



图5.5 权限授权

权限授权核心代码如下所示：

@Transactional  
public boolean saveBatch(String roleId， Set<String> resourceIds) {  
 if (CollectionUtils.isEmpty(resourceIds))  
 return false;  
 removeByRoleId(roleId);  
 Set<RoleResource> userRoles = resourceIds.stream().map(resourceId -> new RoleResource(roleId， resourceId)).collect(Collectors.toSet());  
 return saveBatch(userRoles);}

## 5.4 网关管理功能实现

网关管理只给平台管理员使用，应用场景为平台管理员修改增加、修改、删除网关策略。网关服务主要实现是路由配置策略（限流、转发等网关动作），如图5.6所示。

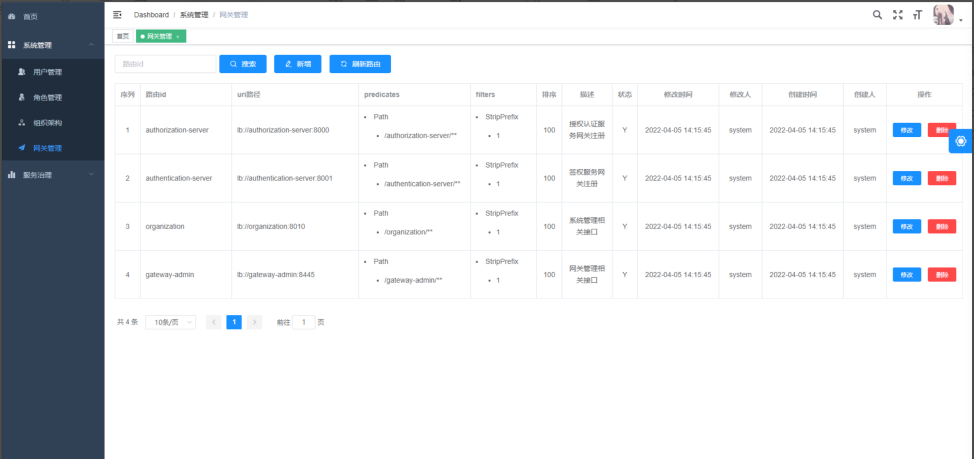


图5.6 网关管理

网关添加核心代码如下所示：

public boolean add(GatewayRoute gatewayRoute) {  
 boolean isSuccess = this.save(gatewayRoute);  
 // 转化为gateway需要的类型，缓存到redis， 并事件通知各网关应用  
 RouteDefinition routeDefinition = gatewayRouteToRouteDefinition(gatewayRoute);  
 gatewayRouteCache.put(gatewayRoute.getRouteId()， routeDefinition);  
 eventSender.send(BusConfig.ROUTING\_KEY， routeDefinition);  
 return isSuccess;  
}

## 5.5 IOT设备管理功能实现

IOT设备管理主要负责IOT设备维护、IOT设备标记功能。物联设备标记、告警日志与大屏监控功能都依赖物联设备维护。[6]

（1）物联设备维护

物联设备维护主要功能是对智能IOT设备进行录入、修改、删除操作，展示IOT设备的运行状态，如图5.7所示。



图5.7 物联设备维护界面

物联设备核心代码如下所示：

public DataResponse<Integer> updateDevicePosition(DeviceUpdatePositionRequest request) {

check(request);

DeviceDO deviceDO = deviceRepository.getById(request.getDeviceNo());

AssertUtil.notNull(deviceDO， BusinessErrorEnum.PARAM\_ERROR， "该设备不存");

deviceDO.setX(request.getX());

deviceDO.setY(request.getY());

boolean bool = false;

if (request.getX() == null && request.getY() == null) {

bool = deviceRepository.updateDevicePosition(deviceDO.getDeviceNo());

} else {

deviceDO.setPositionStatus(PositionStatusEnum.MARK.getCode());

bool = deviceRepository.updateById(deviceDO);

}

AssertUtil.isTrue(bool， BusinessErrorEnum.SAVE\_ERROR， "修改坐标失败");

return DataResponse.ok(1);

}

（2）物联设备标记

物联设备标记使用的前提，是需要在物联设备维护将IOT设备注册到对应的地图上，右侧的侧边框才能显示IOT设备，如图5.8所示。

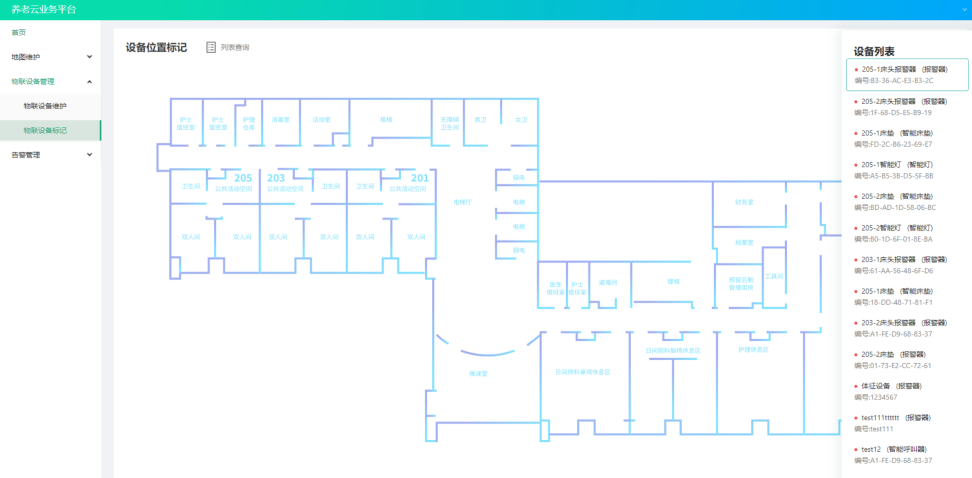


图5.8 物联设备标记界面

具体操作如下：(1)首先确定已经在物联设备维护中注册了对应的IOT设备；(2)从右侧的侧边栏将IOT设备拖动出来，移动到想要标记的位置点上，松开鼠标，这时便触发request请求，将标记的XY坐标系上传，修改数据库表中信息实施标注，右侧IOT设备状态由绿变红代表标记完成；(3)可拖动IOT设备图标进行重新标记。[12]

在物联设备标记中，具体操作逻辑集中在前端上，这一部分使用的是React框架，也充分体现了React框架处理复杂逻辑业务的能力。在地图标记这个功能上使用了AILabel这个轻量级图片标注框架，来实现地图图片缩放、平移、标注等功能。运用多图层渲染，实现3层图层层级来实现业务功能。

@Override

Public List<DeviceApiMapList>selectDevicePositionMarkList(DeviceApiRequest request) {

List<DeviceApiMapListResponse> deviceList = deviceRepository.selectDeviceApiPositionMarkList(request，InitOrganizationUtil.getCurrentUserOrganization());

convertDevice(deviceList);

return CollectionUtils.isEmpty(deviceList) ? Collections.emptyList() : deviceList;

}

## 5.6 告警管理功能实现

告警管理主要实现告警信息的查看、告警处理功能，告警日子查看，其中告警日志不可删除。需要在告警日志中处理告警设备，否则告警状态无法消除，大屏监控仍然提示告警消息。如图5.9所示。

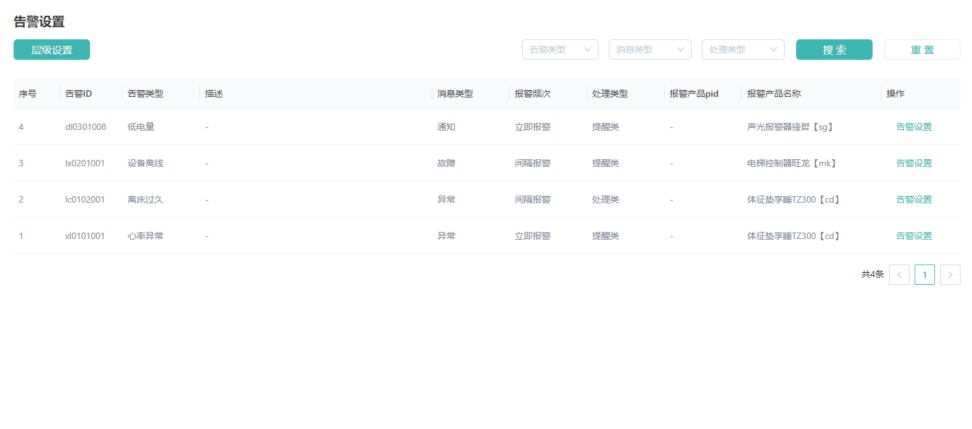


图5.9 告警界面

具体实现如下：

（1）IOT设备检测到自己发生异常，调用与后端协商好的接口上报；

（2）后端微服务收到上报消息，写入告警日志表当中，更改IOT设备的状态。

消警功能核心代码如下所示：

@Override

publicDataResponse<Integer> cancelAlarm(CancelAlarmRequest request) {

check(request);

DeviceDO deviceDO = deviceRepository.getById(request.getDeviceNo());

List<DeviceDO> list = deviceRepository.selectByDeviceIds(deviceDO.getdevice);

String[] deviceIds = request.getDeviceIds().split("，");

deviceService.doDeviceDPControl(deviceIds，request.getDpCode()， request.getDpId()， request.getDpType()， request.getDpValue()， 0);

return DataResponse.ok(1);

}

## 5.7 监控大屏功能实现

大屏监控是整个系统核心，作为项目中最为重要的部分，上述所有的功能都是为了支撑大屏监控。可通过实时展示在右侧地图上，让机构人员观察老人所在位置预防可能发生的走失、进入危险地带等。当IOT设备发生告警时，例如老人摔倒、设备电量不足、烟雾传感器报警等，IOT设备的颜色由绿变红，发生告警提示音，让机构人员快速处理。如图5.10所示。



图5.10 监控大屏

监控大屏核心代码如下所示

@Override

public void updateAlarmStatus(IotAlarmStatusUpdateParam iotAlarmStatusUpdateParam) {

String deviceNo = iotAlarmStatusUpdateParam.getDeviceNo();

String alarmType = iotAlarmStatusUpdateParam.getAlarmType();

String originalStatus = iotAlarmStatusUpdateParam.getOriginalStatus();

String targetStatus = iotAlarmStatusUpdateParam.getTargetStatus();

IotAlarmTypeEnum iotAlarmTypeEnum = IotAlarmTypeEnum.valueOf(alarmType);

QueryWrapper<AlarmRecord> queryWrapper = new QueryWrapper<>();

queryWrapper.eq("device\_no"， deviceNo);

queryWrapper.eq("alarm\_status"， originalStatus);

queryWrapper.eq("alarm\_type"， iotAlarmTypeEnum.getDescription());

queryWrapper.eq("status"， Status.ENABLED);

List<AlarmRecord> alarmRecords = selectList(queryWrapper);

SysAccountModel userDetails = (SysAccountModel) SessionLocal.getUserDetails();

UpdateWrapper<AlarmRecord> updateWrapper = new UpdateWrapper<>();

updateWrapper.in("id"，collect);

updateWrapper.set("alarm\_status"，targetStatus);

updateWrapper.set("updated\_time"，LocalDateTime.now());

updateWrapper.set("updated\_uid"，userDetails != null ? userDetails.getId(): -1L);

updateWrapper.set("processed\_uid"，userDetails != null ? userDetails.getId(): -1L);

updateWrapper.set("processed\_time"，LocalDateTime.now());

int num = update(null，updateWrapper);

alarmRecords.forEach(alarmRecord -> monitorEventPublisher

.publishEvent(new AlarmRelieveEvent(buildDeviceAlarmReliveVO(alarmRecord， targetStatus))));

}

# 6 系统测试

## 6.1 系统测试概述

### 6.1.1 系统测试目的

系统测试是对系统现有功能特性以及系统性能的验证。本文的前两章分别阐述了养老云平台的需求分析和设计与实现，在测试中将会通过对系统测试验证现在的系统是否满足设计需求，并且改正在测试阶段暴露出的问题。[5]

### 6.1.2 系统测试所需环境

进行系统测试的环境如表6.1所示。

表6.1 系统环境表

|  |  |
| --- | --- |
| 操作系统 | 版本 |
| CPU | AMD Ryzen 7 5800H 8核16线程 |
| 内存 | 16G |
| 带宽 | 本机部署，它机测试（内网环境，可忽略） |
| MySQL数据库 | 版本：MySQL 8 |
| Redis数据库 | 版本：Redis4.0 |

## 6.2 系统的功能测试

### 6.2.1 登录功能测试

登录功能测试是对登录页面进行一系列测试操作，如表6.2所示。

表6.2 用户登录测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 软件/项目 基于Web的养老院云平台的设计与实现 | | | |
| 功能模块 用户登录 | | | |
| 用例编号 001 | | | |
| 功能特性 养老云平台登录页面，进行基本常规操作 | | | |
| 测试目的 验证是否能登录成功 | | | |
| 预置条件 数据库提前储备好信息 | | | |
| 测试人 张冰如 | | | |
| 测试时间 2023/2/30 | | | |
| 操作步骤 | 操作描述 | 预期结果 | 实际结果 |
| 1 | 机构用户进入业务服务平台登录页面或平台管理员进入公共管理平台登录页面，输入正确的账号密码 | 成功登录 | （符合） |
| 2 | 机构用户进入业务服务平台登录页面或平台管理员进入公共管理平台登录页面，输入错误的账号密码 | 提示“账号或密码错误” | （符合） |

### 6.2.2 用户管理功能测试

用户管理功能的测试是平台管理员对用户进行创建、搜索、修改、删除操作，如表6.3所示。

表6.3 用户管理测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 软件/项目 基于Web的养老院云平台的设计与实现 | | | |
| 功能模块 账号管理 | | | |
| 用例编号 002 | | | |
| 功能特性 平台管理员对用户进行创建、搜索、修改、删除操作 | | | |
| 测试目的 验证创建、搜索、修改、删除是否能登录 | | | |
| 预置条件 数据库提前储备好信息 | | | |
| 测试人 张冰如 | | | |
| 测试时间 2023/2/30 | | | |
| 操作步骤 | 操作描述 | 预期结果 | 实际结果 |
| 1 | 点击新增，输入用户姓名，用户名，手机号，密码，角色，用户详情点击创建。 | 创建成功 | （符合） |
| 2 | 选择要修改的用户，点击修改，输入用户姓名，用户名，手机号，密码，角色，用户详情点击修改。 | 修改成功 | （符合） |
| 3 | 选择要删除的用户，点击删除 | 删除成功 | （符合） |
| 4 | 选择用户名或者手机号点击搜索。 | 搜索成功 | （符合） |

### 6.2.3 角色管理功能测试

角色管理功能的测试是平台管理员对角色进行创建、搜索、删改操作，如表6.4所示。

表6.4 角色管理测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 软件/项目 基于Web的养老院云平台的设计与实现 | | | |
| 功能模块 角色管理 | | | |
| 用例编号 003 | | | |
| 功能特性 平台管理员对角色进行创建、搜索、修改、删除操作 | | | |
| 测试目的 验证创建、搜索、修改、删除是否能登录 | | | |
| 预置条件 数据库提前储备好信息 | | | |
| 测试人 张冰如 | | | |
| 测试时间 2023/2/30 | | | |
| 操作步骤 | 操作描述 | 预期结果 | 实际结果 |

表6.4（续） 角色管理测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 操作步骤 | 操作描述 | 预期结果 | 实际结果 |
| 1 | 点击新增，输入角色号，角色代码，角色名，角色描述，点击创建。 | 创建成功 | （符合） |
| 2 | 选择要修改的用户，点击修改，输入角色号，角色代码，角色名，角色描述，点击修改。 | 修改成功 | （符合） |
| 3 | 选择要删除的角色，点击删除 | 删除成功 | （符合） |
| 4 | 选择角色代码或者角色名点击搜索。 | 搜索成功 | （符合） |

### 6.2.4 网关管理功能测试

网关管理功能的测试是平台管理员对网关进行新增、搜索、修改、删除操作，如表6.5所示。

表6.5 网关管理测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 软件/项目 基于Web的养老院云平台的设计与实现 | | | |
| 功能模块 网关管理 | | | |
| 用例编号 004 | | | |
| 功能特性 平台管理员对网关进行新增、搜索、修改、删除操作 | | | |
| 测试目的 验证创建、搜索、修改、删除是否能登录 | | | |
| 预置条件 数据库提前储备好信息 | | | |
| 测试人 张冰如 | | | |
| 测试时间 2023/2/30 | | | |
| 操作步骤 | 操作描述 | 预期结果 | 实际结果 |
| 1 | 点击新增，输入路由id，url路径，predicates，filters，排序，描述，状态，点击创建。 | 创建成功 | （符合） |
| 2 | 选择要修改的用户，输入路由id，url路径，predicates，filters，排序，描述，状态，点击修改。 | 修改成功 | （符合） |
| 3 | 选择要删除的网关，点击删除 | 删除成功 | （符合） |
| 4 | 选择路由id点击搜索。 | 搜索成功 | （符合） |

### 6.2.5 IOT设备管理功能测试

IOT设备管理功能的测试是机构用户对IOT设备进行添加、搜索、编辑、删除、标记操作，如表6.6所示。

表6.6 IOT设备管理测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 软件/项目 基于Web的养老院云平台的设计与实现 | | | |
| 功能模块 IOT设备管理 | | | |
| 用例编号 005 | | | |
| 功能特性 机构用户对IOT设备进行进行添加、搜索、编辑、删除、标记操作 | | | |
| 测试目的 验证创建、搜索、修改、删除是否能登录 | | | |
| 预置条件 数据库提前储备好信息 | | | |
| 测试人 张冰如 | | | |
| 测试时间 2023/2/30 | | | |
| 操作步骤 | 操作描述 | 预期结果 | 实际结果 |
| 1 | 点击添加输入设备编号，设备名称，设备类型，可视化位置点，备注，设备状态点击添加 | 添加成功 | （符合） |
| 2 | 选择要修改的设备，输入输入设备编号，设备名称，设备类型，可视化位置点，备注，设备状态，点击修改。 | 修改成功 | （符合） |
| 3 | 选择要删除的设备，点击删除 | 删除成功 | （符合） |
| 4 | 选择设备类型或设备状态点击搜索。 | 搜索成功 | （符合） |
| 5 | 机构人员进入标记界面，双击已标记设备。 | 标记成功 | （符合） |

### 6.2.6 告警管理功能测试

告警管理功能的测试是机构用户查看IOT设备告警日志，如表6.7所示。

表6.7 告警管理测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 软件/项目 基于Web的养老院云平台的设计与实现 | | | |
| 功能模块 告警管理 | | | |
| 用例编号 006 | | | |
| 功能特性 机构用户查看IOT设备告警日志 | | | |
| 测试目的 验证告警日志显示，告警消警 | | | |
| 预置条件 数据库提前储备好信息 | | | |
| 测试人 张冰如 | | | |
| 测试时间 2023/2/30 | | | |
| 操作步骤 | 操作描述 | 预期结果 | 实际结果 |
| 1 | 机构人员进入告警日志列表，点击条件查询按钮。 | 正常显示查询的数据。 | （符合） |
| 2 | 机构人员消除告警状态，回到大屏监控和物联设备管理查看告警状态。 | 列表告警状态消失，大屏监控与物联设备管理对应的告警状态也消失。 | （符合） |
| 3 | 机构人员进入标记界面，双击已标记设备。 | 标记成功 | （符合） |

### 6.2.7 监控大屏功能测试

监控大屏功能的测试是机构用户查看监控大屏，监控大屏展示IOT设备，显示IOT告警，如表6.8所示。

表6.8 告警管理测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 软件/项目 基于Web的养老院云平台的设计与实现 | | | |
| 功能模块 监控大屏 | | | |
| 用例编号 007 | | | |
| 功能特性 机构用户查看监控大屏 | | | |
| 测试目的 验证监控大屏显示IOT设备，显示IOT告警 | | | |
| 预置条件 数据库提前储备好信息 | | | |
| 测试人 张冰如 | | | |
| 测试时间 2023/2/30 | | | |
| 操作步骤 | 操作描述 | 预期结果 | 实际结果 |
| 1 | 机构人员进入监控大屏查看大屏IOT设备异常展示。 | 正常显示查询的数据。 | （符合） |
| 2 | 机构人员进入监控大屏查看IOT设备展示。 | 列表告警状态消失，大屏监控与物联设备管理对应的告警状态也消失。 | （符合） |

# 7 总结与展望

## 7.1 总结

本文在中国老龄化的背景下对新型养老模式进行了深入的探究，并对养老云平台的设计实现过程进行了详细的描述。本文在养老云平台的设计实现中主要完成了以下工作。

（1）完成了养老云平台课题研究背景以及国内养老现状的调研。在调研实际场景后，根据实际业务需求和业务流程，对权限模块、运维模块、地图模块、物联模块、告警模块进行了功能需求分析，并且进行了系统安全需求、易用需求、性能需求、可靠需求进行了非功能需求分析。

（2）在调研当下主流技术架构，基于系统需求分析，根据调研结果选用了MySQL、Redis、RabbitMQ、Nacos、SpringCloud、OAuth2等社区活跃、多家主流互联网公司使用的主流技术，保证了后续可不断升级优化系统，并且完成了养老云平台的系统架构设计与数据库设计。

（3）基于当下微服务设计思想，将上述需求模块拆分成权限服务、用户服务、网关服务、消息队列服务、监控服务、地图服务并且进行了详细的设计与实现描述。[13]同时根据前端框架React的独特个性实现了公共管理平台与业务平台两个平台。

（4）完成了系统功能性测试与非功能性测试。在系统功能性测试中以业务平台的地图服务进行了测试与描述，以及结果分析。非功能性测试中对系统安全需求、易用需求、性能需求、可靠需求进行了测试描述。

## 7.2 安全与环保问题分析

本系统在数据安全、电脑安全和环保方面都有优势，使用了加密、正规软件和分页加载等技术，提高了用户的体验和满意度。

（1）本系统在后端对密码进行了加密处理，这样即使数据库被盗，也无法轻易破解用户的密码。这样可以有效地保护用户的隐私和安全，避免了数据泄露的风险。

（2）本系统是在正规软件中开发的，没有恶意代码或病毒，保证了电脑运行的安全。这样可以防止电脑被感染或损坏，提高了电脑的性能和稳定性。

（3）本系统不会在无运行时占用电脑资源，节约了电能，有利于环保。这样可以减少电脑的功耗和发热，延长了电脑的寿命和效率，也减少了对环境的污染和影响。

## 7.3 技术经济与成本分析

本系统使用了开源技术、自动化工具和统一环境，降低了开发成本、简化了操作和提高了兼容性。

（1）本系统的开发成本低，因为除了正版软件外，所用的技术都是开源的，不需要额外付费。

（2）本系统的操作简单，可以独立开发，不需要雇用人力资源，节省了人力成本。例如，该系统使用了Maven作为项目管理工具，可以自动化构建和部署项目，无需手动配置和安装依赖包。。

（3）本系统的兼容性好，可以支持各种依赖包和语言，并且使用jdk8作为集成环境，避免了冲突和错误。例如，该系统使用了Spring Boot Starter Web来集成各种Web相关的依赖包，如Tomcat、Jackson、Hibernate Validator等，并且提供了统一的版本管理和配置文件。

## 7.4 展望

养老云平台因为有公司实际的项目参考，对需求的理解走在前沿，有着相对完善的需求分析，同时利用主流且成熟的技术进行开发，并且遵循良好的代码开发规范，但是由于自身技术上和时间上的限制，养老院平台仍有着不足之处，存在进一步完善的空间：

在分布式锁的实现上，因为用过使用Redis并且搭建集群实现分布式锁，在实际中存在着以下问题：当持有锁的Redis主节点宕机时，极端情况下没有一个Radis从节点同步到该锁，此时可能会导致业务流走不下去的过程，在这种情况要判断一个锁是否还存活，需要对其他从Redis节点进行轮询查看从节点是否同步到锁，当下主流的策略是有五分之三的从节点同步到锁即该锁存活。因此在未来可以将分布式锁换成ZooKeeper实现。[12]

在养老云平台使用过程中，会发现有时修改数据时效性并不一致，这是由于使用MySQL+Redis这种体系势必会面临双写不一致问题，这是不可避免、从原则上无法解决的问题，这种情况只能够依赖第三方组件实现强一致性效果，在未来可以嵌入阿里开源的Canal来解决这种双写问题。

# 参考文献

[1]张琴.养老院智能管理系统的设计与实现[J].信息与电脑(理论版),2022,34(13):158-160.

[2]段昊鹏.基于物联网技术的养老院智能管理系统设计[J].物联网技术,2022,12(02):99-100+105.DOI:10.16667/j.issn.2095-1302.2022.02.027.

[3]王丹,孙晓宇,杨路斌,高胜严.基于SpringBoot的软件统计分析系统设计与实现[J].软件工程,2019,22(03):40-42.DOI:10.19644/j.cnki.issn2096-1472.2019.03.012.

[4]李常迪,邵一川,陈醉涵,曹勇.大数据背景下智慧养老平台的设计与应用[J].科技创新与应用,2021,11(22):90-92.

[5]李常迪,邵一川,陈醉涵,曹勇.分布式智能养老系统的开发与应用[J].沈阳大学学报(自然科学版),2021,33(03):260-266.DOI:10.16103/j.cnki.21-1583/n.2021.03.011.

[6]喻佳,吴丹新.基于SpringBoot的Web快速开发框架[J].电脑编程技巧与维护,2021(09):31-33.DOI:10.16184/j.cnki.comprg.2021.09.013.

[7]吴昌政.基于前后端分离技术的web开发框架设计[D].南京邮电大学,2020.

[8]王绍婷. 网上商城管理系统的设计与实现[D].厦门大学,2017.

[9]宋馨来,张海涛.基于前后端分离和Spring Security的用户登录功能的设计与实现[J].科技创新与应用,2021,11(18):19-21，24.

[10]杜成龙,曹海平,吴亮.基于Spring Security的自定义用户认证和授权策略研究[J].信息系统工程,2022(09):75-78.

[11]王欢,李民,邓秀辉,焦宇,余开朝.基于Redis缓存数据库和Nginx负载均衡技术的购物网站性能优化[J].软件导刊,2022,21(08):114-119.

[12]邓秀辉,李民,方惠.基于分布式集群高可用管理信息系统设计[J].制造业自动化,2022,44(07):43-45+122.

[13]叶刚,王立河,王英明,谷国栋.基于Mybatis Plus的动态生成代码设计与实现[J].电脑编程技巧与维护,2019(07):7-8.DOI:10.16184/j.cnki.comprg.2019.07.002.

[14]B. F. Spencer, J. W. Park, and K. A. Mechitov, “Next generation wireless smart sensors toward sustainable civil infrastructure,” Procedia Engineering, vol. 171, pp. 5–13, 2017.

[15]R. Gonzalez-Usach, V. Collado, M. Esteve and C. E. Palau, "AAL open source system for the monitoring and intelligent control of nursing homes," 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), 2017, pp. 84-89, doi: 10.1109/ICNSC.2017.8000072.

# 致 谢

首先，我要感谢我的指导老师，老师知识渊博，在职业发展的道路上给予了我诸多教诲，在我求职路上遇到困难时为我寻求帮助且指引道路方向。同时老师在指导工作上严谨、认真、负责的工作态度，给我留下了深刻的印象。我希望能够以老师为榜样，在今后的工作与学习中砥砺前行。

其次，我要感谢深圳市优必选科技股份有限公司的平台技术部领导许锋刚，以及洪涛龙、徐琪涵等人，在他们的帮助下，我对养老平台业务的需求有了深刻的认知，本文的系统才能够顺利的开发下去，在这期间学到了诸多的技术知识，积累了宝贵的工作经验。

最后，我要感谢我的家人，我的家人永远都是我最好的避风港。在我成长的过程中，感谢你们的无私付出，感谢你们在我失落时对我的鼓励，感谢你们陪我一路成长。在今后的路程中，我将再接再厉，不辜负你们对我培养与期望。

总之，感谢所有评阅我拙文以及参加答辩的各位老师以及在这四年生活学习中给予过我微笑的人，希望所有的离别都是为了下一次更好的相遇。

会者定离，一期一祈，感恩所有的经历，感谢所有的遇见，再见了泉州信息工程学院，再见了博东路249号，山水有来路，早晚复相逢，愿此去经年，于万物中见顶峰。