基于J2EE的尾矿库坝体安全性能检测平台

**摘 要：**随随着科技的不断发展，信息化进程不断加快，互联网传输能力不断增强，现如今已逐步发展为将数据和各类软件聚合存储于“网络云”（大型处理中心）中统一处理，用户只需要一个上网终端即可完成一系列的需求。

设计了一套web尾矿库安全云管理系统，为矿区管理者进行尾矿安全性能检测、日常维护等提供决策支持。系统主要实现的功能有：尾矿库温湿度，浸润线，位移，水位，有毒有害气体，重金属元素数据监测，异常报警，视频监控，企业用户管理，系统信息系统异常记录处理，生成报表等功能。平台使用，前后端分离的开发方式，后端主要采用Java语言开发，主要应用SpringBoot框架，使用mysql + mybatis + jpa的方式进行存储数据，使用POI技术开发报表功能，前端使用了Vue + Element-ui + iView + axios + Echarts等前端框架和技术开发用户交互页面。通过这些功能和技术实现一个高效的、稳定的、实用的云管理系统。

**关键字:** J2EE；尾矿库；数据可视化；前后端分离；安全检测；

# 1 绪论

## 1.1 选题背景及意义

尾矿库是指筑坝阻挡谷口或围地组成的，用以堆存金属或非金属矿山举行矿石选别后排挤尾矿或其他工业废渣的场所[1]。尾矿库是一个具有高势能的人造泥石流伤害源，存在溃坝伤害，一旦出事，极易造成重特大事故[2]。美国克拉克大学公害评定小组的研究表明，尾矿库事故造成的危害，在世界93种事故、公害隐患中，名列第18位。1962年9月26日，云南个旧云锡火谷都尾矿库事故，造成171人死亡、92人受伤，11个村寨及1座农场被毁。1985年8月25日，牛角垅尾矿库事故，造成49人死亡，直接经济损失达1300多万元[3]。2008年9月8日，襄汾尾矿溃坝造成277人死亡、4人失踪、33人受伤，直接经济损失达9619万元[4]。鉴于安全事故频发，迫切需要采用现代化手段对尾矿库进行监控、管理、和事故的预警处理。

随着信息化进程的不断加快，互联网传输能力的不断增强，现如今已逐步发展为将数据和各类软件聚合存储于“网络云”（大型处理中心）中统一处理，用户只需要一个上网终端即可完成一系列的需求。本次研究与开发的这个基于J2EE的尾矿库安全云管理系统，使得尾矿库的管理从之前本地、固定的管理模式转化为云端、灵活的管理模式，用户不在再受地域和终端的限制，只需要任意一款浏览器即可完成一系列的管理、监控、异常处理和生成报表等工作。云端部署，24小时不间断监控，发现异常，即刻预警等待处理，全年保障尾矿库的安全运行。该系统简明易用，减少用户学习成本，在提高用户工作效率同时，最大程度的保障了人民的生命和财产安全。

## 1.2 尾矿库管理系统概述

本系统以Java语言为基础，利用SpringBoot、jpa、Mybatis、POI等最新的技术进行开发，主要是以尾矿库的安全数据监控，异常预警，异常处理，人员管理功能为主，实现最大程度的保障人民的生命和财产安全。

## 1.3 云平台的定义

云平台是指搭建在ESC服务器中的开发环境，是提供给开发人员使用的。软件开发的云平台给开发者提供RDS数据库、ESC服务器、OSS对象存储以及CDN内容分发网络四大部分的开发基础组件。使用云平台后。开发者不需要在服务器的硬件的配置维护，数据库的管理，文件的存储上浪费巨大的时间和资源。

## 1.4 基于J2EE的尾矿库安全云管理系统的特点

健壮性：系统在输入部分做了过滤，针对用户名、密码电子邮箱等有规则的输入进行了正则判定，过滤非法输入，对错误输入进行容错提示。

可扩展性：尾矿库安全管理系统采用前后端分离的方式开发，前端后台解耦，对前端文件也进行了模块化封装，增强了代码的可扩展性，新增功能只需添加新的模块后引用即可。

兼容性：系统是基于云平台的，只存在前端页面的兼容性问题，本系统前端页面可兼容大部分浏览器，已适配谷歌、火狐、360等主流浏览器基本无兼容性问题。

可移植性：系统的前端部分采用的是Vue框架实现，对电脑环境没有任何要求，有浏览器即可。

易用性：系统再设计之初就进行了多方调研，力争真正解决尾矿库安全管理的及时性和人力难以控制等问题。在代码实现前期进行了充分的原形和交互逻辑的设计，保证系统的易用性。

## 1.5 本课题要解决的问题

1. 解决尾矿库温度、湿度、浸润线等安全数据的实时监测上报。
2. 帮助管理人员进行全方位管理，系统集成实时视频监控和监控地图显示等。
3. 解决用人单位对人员进行管理的问题，并可实现权限控制等功能。

# 2 相关开发技术与开发、运行环境

## 2.1 Vue框架

Vue是一套构建用户界面的 渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。另一方面，Vue 完全有能力驱动采用[单文件组件](http://doc.vue-js.com/v2/guide/single-file-components.html)和[Vue生态系统支持的库](http://github.com/vuejs/awesome-vue#libraries--plugins)开发的复杂单页应用。Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。为前端开发人员提供便捷的服务，减少开发周期。使用Vue可以将前端项目工程化，提供更加规范的web开发。

## 2.2 Element-ui iview组件

Element-UI 和iViev（基于 Vue.js 的高质量UI 组件库）优点：

一致性 Consistency：与现实生活一致：与现实生活的流程、逻辑保持一致，遵循用户习惯的语言和概念；在界面中一致：所有的元素和结构需保持一致，比如：设计样式、图标和文本、元素的位置等。

反馈 Feedback：控制反馈：通过界面样式和交互动效让用户可以清晰的感知自己的操作；页面反馈：操作后，通过页面元素的变化清晰地展现当前状态。

效率 Efficiency：简化流程：设计简洁直观的操作流程；清晰明确：语言表达清晰且表意明确，让用户快速理解进而作出决策；帮助用户识别：界面简单直白，让用户快速识别而非回忆，减少用户记忆负担。

可控 Controllability：用户决策：根据场景可给予用户操作建议或安全提示，但不能代替用户进行决策；结果可控：用户可以自由的进行操作，包括撤销、回退和终止当前操作等。

## 2.2 MyBatis框架

MyBatis起源于apache开源项目iBatis, 是一款优秀的持久层框架，MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以使用简单的 XML 或注解来配置和映射原生类型、接口和 Java 的 POJO（Plain Old Java Objects，普通老式 Java 对象）为数据库中的记录。

## 2.3 JPA框架

简单方便：JPA的主要目标之一就是提供更加简单的编程模型：在JPA框架下创建实体和创建Java 类一样简单，没有任何的约束和限制，只需要使用 javax.persistence.Entity进行注释，JPA的框架和接口也都非常简单，没有太多特别的规则和设计模式的要求，开发者可以很容易地掌握。JPA基于非侵入式原则设计，因此可以很容易地和其它框架或者容器集成。查询能力： JPA的查询语言是[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1" \t "_blank)而非面向数据库的，它以面向对象的自然语法构造查询语句，可以看成是Hibernate HQL的等价物。JPA定义了独特的[JPQL](https://baike.baidu.com/item/JPQL" \t "_blank)（Java Persistence Query Language），JPQL是EJB QL的一种扩展，它是针对实体的一种查询语言，操作对象是实体，而不是关系数据库的表，而且能够支持批量更新和修改、JOIN、GROUP BY、HAVING 等通常只有 SQL 才能够提供的高级查询特性，甚至还能够支持[子查询](https://baike.baidu.com/item/%E5%AD%90%E6%9F%A5%E8%AF%A2" \t "_blank)。

## 2.4 Druid技术

数据库连接是一种关键的有限的昂贵的资源，这一点在多用户的网页应用程序中体现得尤为突出。对数据库连接的管理能显著影响到整个应用程序的伸缩性和健壮性，影响到程序的性能指标。数据库连接池正是针对这个问题提出来的。数据库连接池负责分配、管理和释放数据库连接，它允许应用程序重复使用一个现有的数据库连接，而不是再重新建立一个;释放空闲时间超过最大空闲时间的数据库连接来避免因为没有释放数据库连接而引起的数据库连接遗漏。这项技术能明显提高对数据库操作的性能。 Druid是阿里巴巴技术团队开发的一款优秀的数据库连接池，并且已经在GitHub上开源，它与其他国外优秀的数据库连接池用法类似，性能更好，同时加入了日志监控，可以让开发者明确的看到自己的sql语句的执行情况。可以说 Druid是目前国内互联网企业中使用范围最广的数据库连接池。

Druid首先是一个数据库连接池。Druid是目前最好的数据库连接池，在功能、性能、扩展性方面，都超过其他数据库连接池，包括DBCP、C3P0、BoneCP、Proxool、JBoss DataSource。Druid已经在阿里巴巴部署了超过600个应用，经过一年多生产环境大规模部署的严苛考验。    Druid是一个JDBC组件，它包括三个部分： 基于Filter－Chain模式的插件体系。 DruidDataSource 高效可管理的数据库连接池。SQLParser。Druid的功能如下：

1、替换DBCP和C3P0。Druid提供了一个高效、功能强大、可扩展性好的数据库连接池。

2、可以监控数据库访问性能，Druid内置提供了一个功能强大的StatFilter插件，能够详细统计SQL的执行性能，这对于线上分析数据库访问性能有帮助。

3、数据库密码加密。直接把数据库密码写在配置文件中，这是不好的行为，容易导致安全问题。DruidDruiver和DruidDataSource都支持PasswordCallback。

4、SQL执行日志，Druid提供了不同的LogFilter，能够支持Common-Logging、Log4j和JdkLog，你可以按需要选择相应的LogFilter，监控你应用的数据库访问情况。

5、扩展JDBC，如果你要对JDBC层有编程的需求，可以通过Druid提供的Filter机制，很方便编写JDBC层的扩展插件。

## 2.5 SpringBoot框架

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。Springboot包含spring和SpringMVC的所有功能， Spring是一个开放源代码的设计层面框架，他解决的是业务逻辑层和其他各层的松耦合问题，因此它将面向接口的编程思想贯穿整个系统应用。Spring是于2003 年兴起的一个轻量级的Java 开发框架，由Rod Johnson创建。简单来说，Spring是一个分层的JavaSE/EE **full-stack(一站式)** [轻量级](https://baike.so.com/doc/585215-619452.html)开源框架。Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。 使用 Spring 可插入的 MVC 架构，可以选择是使用内置的 Spring Web 框架还是 Struts 这样的 Web 框架。通过策略接口，Spring 框架是高度可配置的，而且包含多种视图技术，例如 JavaServer Pages（JSP）技术、Velocity、Tiles、iText 和 POI。Spring MVC 框架并不知道使用的视图，所以不会强迫您只使用 JSP 技术。Spring MVC 分离了[控制器](https://baike.so.com/doc/422704-447609.html" \t "_blank)、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行[定制](https://baike.so.com/doc/6757325-6971924.html)。其他特点如下：

1. 创建独立的Spring应用程序

2. 嵌入的Tomcat，无需部署WAR文件

3. 简化Maven配置

4. 自动配置Spring

5. 提供生产就绪型功能，如指标，健康检查和外部配置

6. 绝对没有代码生成并且对XML也没有配置要求

## 2.6 ECharts开源技术

Echarts 基于H5的Canvas类库 ZRender,来源于百度的开源项目，Echar提供丰富的js图表库，并且支持高度定制，Echar主要的三大特点是支持拖拽后重算，数据驱动视图和值域的漫游，提高了使用者的体验性和良好的交互。

### 2.6.1 ECharts提供丰富的图表类型

ECharts 提供折线/柱状/饼/散点/雷达/k线/热力等多种图表类型，并且不同类型的图可以混搭使用。

### 2.6.2 ECharts数据驱动特点

ECharts图表得益于数据驱动改变，后台提供的数据改变，图表便轻松发生更新。开发者只需要关注输入图表数据的变换，图表会自动检查最新数据数据与当前数据的差异，并更新图表。

### 2.7 webSocket技术

WebSocket协议是基于TCP的一种新的网络协议。它实现了浏览器与服务器全双工(full-duplex)通信--允许服务器主动发送信息给客户端。WebSocket通信协议于2011年被IETF定为标准RFC 6455，并被RFC7936所补充规范。长久以来, 创建实现客户端和用户端之间双工通讯的web app都会造成HTTP轮询的滥用: 客户端向主机不断发送不同的HTTP呼叫来进行询问。这会导致一系列的问题:

1.服务器被迫为每个客户端使用许多不同的底层TCP连接:一个用于向客户端发送信息，其它用于接收每个传入消息。

2.有些协议有很高的开销，每一个客户端和服务器之间都有HTTP头。

3.客户端脚本被迫维护从传出连接到传入连接的映射来追踪回复。

一个更简单的解决方案是使用**单个TCP连接**双向通信。 这就是WebSocket协议所提供的功能。 结合WebSocket API ，WebSocket协议提供了一个用来替代HTTP轮询实现网页到远程主机的双向通信的方法。

WebSocket协议被设计来取代用HTTP作为传输层的双向通讯技术,这些技术只能牺牲效率和可依赖性其中一方来提高另一方，因为HTTP最初的目的不是为了双向通讯。[编辑本段](https://baike.so.com/create/edit/?eid=6840040&sid=7057314&secid=3)实现原理：

在实现websocket连线过程中，需要通过浏览器发出websocket连线请求，然后服务器发出回应，这个过程通常称为"握手" 。在 WebSocket API，浏览器和服务器只需要做一个握手的动作，然后，浏览器和服务器之间就形成了一条快速通道。两者之间就直接可以数据互相传送。在此WebSocket 协议中，为我们实现即时服务带来了两大好处:

1. Header：互相沟通的Header是很小的-大概只有 2 Bytes

2. Server Push：服务器的推送，服务器不再被动的接收到浏览器的请求之后才返回数据，而是在有新数据时就主动推送给浏览器。

### 2.8 POI技术

Apache POI是一种流行的API，它允许程序员使用Java程序创建，修改和显示MS Office文件。这由Apache软件基金会开发使用Java分布式设计或修改Microsoft Office文件的开源库。它包含类和方法对用户输入数据或文件到MS Office文档进行解码。

### 2.9基于JWT的Token认证机制

JSON Web Token（JWT）是一个非常轻巧的规范。这个规范允许我们使用JWT在用户和服务器之间传递安全可靠的信息。它由三部分组成，头部、载荷与签名。优点如下：

支持跨域访问: Cookie是不允许垮域访问的，这一点对Token机制是不存在的，前提是传输的用户认证信息通过HTTP头传输.

无状态(也称：服务端可扩展行):Token机制在服务端不需要存储session信息，因为Token 自身包含了所有登录用户的信息，只需要在客户端的cookie或本地介质存储状态信息.

更适用CDN: 可以通过内容分发网络请求你服务端的所有资料（如：javascript，HTML,图片等），而你的服务端只要提供API即可.

去耦: 不需要绑定到一个特定的身份验证方案。Token可以在任何地方生成，只要在你的API被调用的时候，你可以进行Token生成调用即可.

更适用于移动应用: 当你的客户端是一个原生平台（iOS, Android，Windows 8等）时，Cookie是不被支持的（你需要通过Cookie容器进行处理），这时采用Token认证机制就会简单得多。

CSRF:因为不再依赖于Cookie，所以你就不需要考虑对CSRF（跨站请求伪造）的防范。

性能: 一次网络往返时间（通过数据库查询session信息）总比做一次HMACSHA256计算 的Token验证和解析要费时得多.

不需要为登录页面做特殊处理: 如果你使用Protractor 做功能测试的时候，不再需要为登录页面做特殊处理.

基于标准化:你的API可以采用标准化的 JSON Web Token (JWT). 这个标准已经存在多个后端库（.NET, Ruby, Java,Python, PHP）和多家公司的支持（如：Firebase,Google, Microsoft）.

### 2.10 开发工具

### 2.10.1 IntelliJ IDEA

IntelliJ IDEA，是用于java语言开发的集成环境(也可用于其他语言)，IntelliJ在业界被公认为最好的java开发工具之一，尤其在智能代码助手、代码自动提示、重构、[J2EE](https://baike.so.com/doc/901087-952427.html" \t "_blank)支持、[Ant](https://baike.so.com/doc/5418621-10440907.html" \t "_blank)、[JUnit](https://baike.so.com/doc/1858643-1965718.html" \t "_blank)、[CVS](https://baike.so.com/doc/1774040-1876053.html" \t "_blank)整合、代码审查、 创新的[GUI设计](https://baike.so.com/doc/7065585-7288495.html)等方面的功能可以说是超常的。

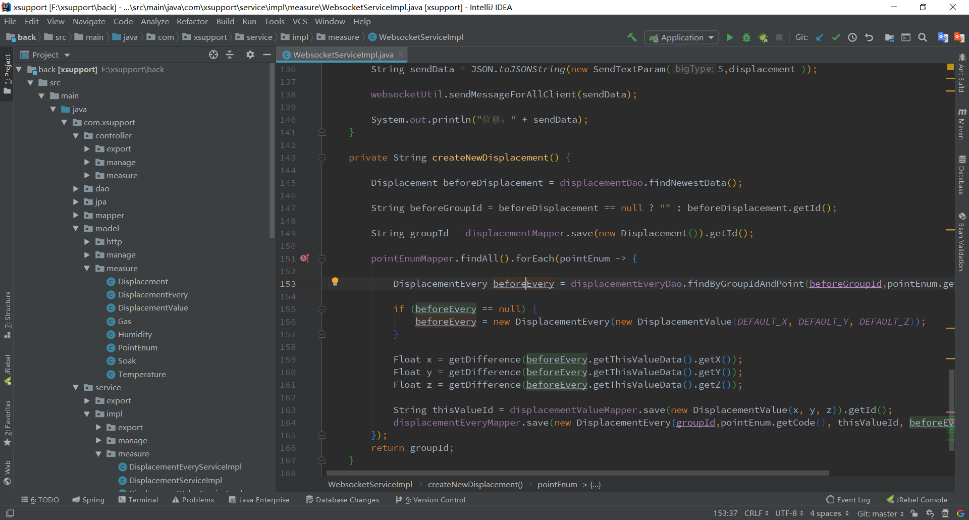


图2-3 IntelliJ IDEA工作界面图

### 2.10.2 WebStorm开发工具

WebStorm 是[jetbrains](https://baike.baidu.com/item/jetbrains" \t "_blank)公司旗下一款JavaScript 开发工具。目前已经被广大中国JS开发者誉为“Web前端开发神器”、“最强大的HTML5编辑器”、“最智能的JavaScript IDE”等。与[IntelliJ IDEA](https://baike.baidu.com/item/IntelliJ%20IDEA" \t "_blank)同源，继承了IntelliJ IDEA强大的JS部分的功能。

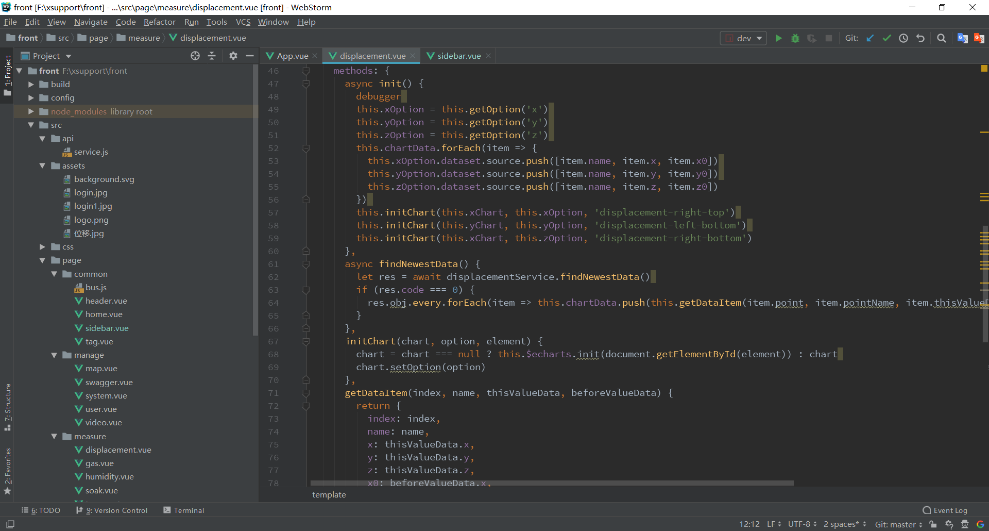


图2-3 WebStorm工作界面图

# 3项目需求分析

## 3.1系统的需求分析

该云管理系统主要是以尾矿库的安全数据监控，异常预警，异常处理，人员管理功能，以及导出数据报表等功能为主。该系统的需求如下：

测量中心，下设温度测量，湿度测量，水位测量，浸润测量，位移测量，金气测量等模块，实时获取后台数据，并在前台做可视化展示，如果数据异常则及时预警。

视频监控，实时采集摄像头可见范围，帮助管理人员更好的掌握尾矿库的实际环境。

监测地图，对周围地形地貌进行全方位展示，方便在突发情况来临前做战略部署或在特殊时期做应急指挥。

数据报表，能够生成特定时间内人员信息和各种监测数据的报表，以excel方式打印，方便管理上报。

用户管理：系统配置了管理员和普通用户的权限，实现不同的用户拥有不同的权限和可操作控制。

系统管理：测量异常集中处理。并支持在线查看系统运行时异常信息方便迅速解决系统bug。

## 3.2 系统的总体需求图

为了更加明确的对系统需求进行分析，这里使用一张尾矿库安全管理系统需求图进行一个基本的阐述。

如图3-1所示，本图展示了尾矿库安全云管理系统的总体需求，管理员或普通用户登录以后可以进入系统进行监控数据查看，异常预警查看，异常处理，人员管理，以及导出数据报表等功能的使用。

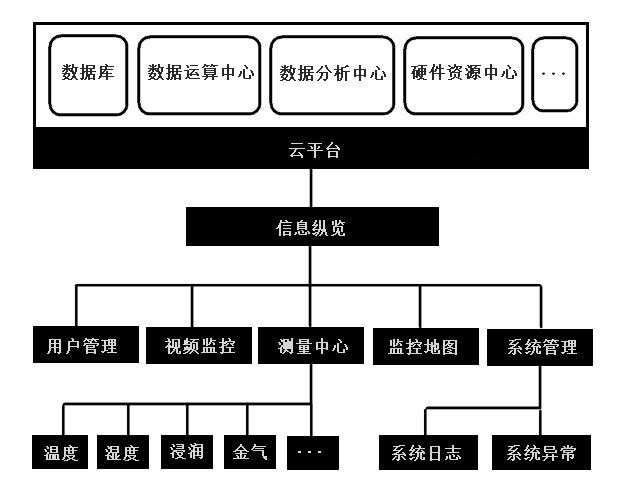


图3-1 项目总体需求图

## 3.3 系统的性能需求

根据现在互联网的开发模式下，一个能够上线的项目需要满足以下几个方面：

1. 按需求分析完成系统设计，编码和实现，没有影响系统正常运行的致命bug。
2. 确保设计功能能正常使用，确定上线的功能没有功能缺失，在功能上可以形成一个闭环。
3. 提供完整的项目文档和接口文档，让新加入的开发人员能够快速加入开发。
4. 提供并建立完备的问题处理机制和响应体系，拥有完整的运维体系，可以随时修复线上问题和进行回滚。

# 4 系统设计

## 4.1系统设计的原则

良好的系统设计原则与项目的完美实现密切相关，在程序设计之中要遵循以下总结的设计原则：

1. 避免重复性（Avoid repeatability），避免重复性是程序设计的最基本的原则。在程序中必然拥有很多数据、存储结构，例如：函数、循环体、对象等。一旦重复就会造成代码冗杂，影响效率。
2. 简单原则（Simple principle），程序的简单是一个程序设计中不可缺少的，使用最少的代码，实现一个很重要的功能，并且简单代码的空间大小和执行时间更少，发现漏洞也会更加及时，对于维护更加便利。
3. 低耦合原则（Low coupling principle），这是所有的框架和程序设计都在遵循的原则和追求的目标，需要在代码的任何一个地方都减少对其他代码的依赖，底耦合往往是一个优秀系统设计的标准。

## 4.2 系统概要设计

该系统web层的MVC使用SpringMVC实现，其具体的执行流程如图4-1：

1. 配置单一控制器 WEB.xml 接收请求；
2. HandlerMapping 将请求派发到 子控制器；
3. 某个Controller 会调用业务层进行业务逻辑处理，处理后返回视图；
4. 视图处理器处理视图；
5. 前端控制器解析和渲染数据和视图，通过response返回给客户端展示。

映射视图图

发送请求图

View（视图层）

1、渲染数据2、向Controller发送请求

Controller（控制层）

1、定义行为和动作2、解析视图传递数据

3、将动作映射到

Model

封装应用功能

模型数据

管理数据的存储一致性

修改model

图4-1 MVC模型图

项目的开发模型为现在流行的MVC，即模型-视图-控制模型。模型图如图4-1所示。在该模型下前端的静态页面和数据渲染为视图层，mybatis将数据库表自动生成为一个个单独的JavaBean对象作为模型层，控制层有springMVC中每一个controller组成。

## 4.3 系统模块设计

通过对项目中的具体需求分析以及结合实际情况，该系统主要需要完成四大模块，具体模块内容如图4-2系统应用功能图所示。

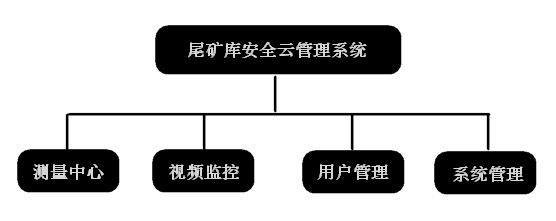


图4-2系统应用功能图

设计系统前端页面的时候以简洁易上手为主，让用户一目了然。在设计过程中，不在一个页面放很多内容，让网页间接为主。同时尽量采用数据可视化展示，避免用户对大量监测数据的视觉疲惫，并帮助其更快更好的掌握和处理相关数据反映出来的问题。

### 4.3.1 测量中心模块设计

测量中心模块是整套系统中最重要的一个模块，在设计上对数据的展示做了图表对应的页面效果，从图表上即可看到各类监测出数据的详细数据，并且采用后台动态更新的方式进行数据获取。同时右侧列出相关数据和数据的状态和相关处理操作，并可以设计各个模块的阈值，测量值大于阈值会记录测量记录。

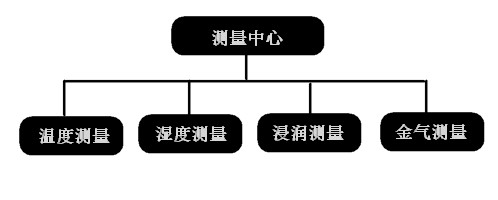


图4-4测量中心 模块图

### 4.3.2 视频监控模块设计

视频监控模块采用多视角设计，与外界硬件摄像头进行实时通信，获取监控范围内的视频画面，并且可监控多个方位。

### 4.3.4 人员管理模块设计

本系统的人员管理模块只要有人员的增加，删除，修改，查询，禁用，人员的权限分配等功能构成，人员的增加，删除，修改，查询是对一个管理系统最基本的要求，这样系统才有了它设计的意义，帮助用户更好的处理工作。而权限分配则进一步对用户的可操作范围有了明确的限制，本系统主要设计管理员和普通用户，两种角色，管理员角色可以进行用户的增加、删除、修改，查询和禁用，可进行密码修改，等功能的使用，而普通用户不能删除，修改其他人的信息。

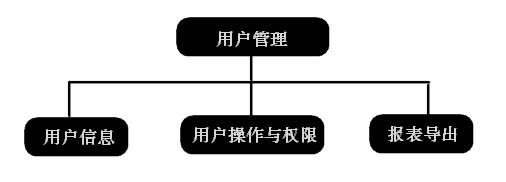


图4-6人员管理 模块图

### 4.3.5 系统管理模块设计

系统管理是整个系统总管理的重要模块，系统管理用来记录测量中心记录的一些异常数据，即超过阈值的数据。并可以增加备注，修改异常状态，报表导出等功能。

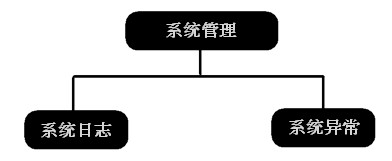
****

图4-7 系统管理 模块图

## 4.4数据库设计

无论哪一个系统都离不开数据库，它是所有数据的家，没有数据库任何系统就无法使用，在软件设计中，数据库设计的好坏一定程度决定程序设计的好坏。所以在程序设计中数据库的设计尤为的重要。数据库设计中需要设计好数据存储结构，好的数据存储结构对系统的功能和性能都会产生非常大的影响[8]。同样，优秀的数据存储结构可以使代码拥有更好的扩展性，对后期需求增加提供一个良好基础。

根据系统需求分析和具体实现，本系统的教务管理建立了一个名为xsupport的数据库，编码为UTF-8，xsupport数据库中包含下面几张数据表：

1. 用户表（user），用于存储系统注册用户的相关信息，具体的结构如用户表4-1用户表所示：

表4-1用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| id | id | Varchar(3) | Yes |
| 创建时间 | createTime | datetime | No |
| 账号 | username | varchar(20) | No |
| 密码 | password | varchar(20) | No |
| 是否管理员 | isAdmin | tinyint | No |
| 是否禁用 | isForbidden | tinyint | No |
| 姓名 | name | Varchar5) | No |
| 性别 | sex | tinyint | No |
| 年龄 | age | int | No |
| 家庭住址 | address | varchar(50) | No |
| 联系电话 | phone | varchar(11) | No |
| 备注 | remark | text |  |

2．湿度表（humidity），该表用于存储监测湿度的相关信息，具体的结构，如表4-3湿度监控表所示：

表4-3湿度监控表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| Id | id | varchar (32) | Yes |
| 采集时间 | createTime | datetime | No |
| 采集值 | value | float(5,2) | No |
| 阈值 | limitValue | float(5,2) | No |
| 湿度状态 | state | int (1) | No |

3．温度表（temperature），该表用于存储监测温度的相关信息，具体的结构，如表4-3温度监控表所示：

表4-3温度监控表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| Id | id | varchar (32) | Yes |
| 采集时间 | createTime | datetime | No |
| 采集值 | value | float(5,2) | No |
| 阈值 | limitValue | float(5,2) | No |
| 湿度状态 | state | int (1) | No |

4．浸润监控表（soak），该表用于存储监测湿度的相关信息，具体的结构，如表4-3浸润监控表所示：

表4-3浸润监控表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| id | id | varchar (32) | Yes |
| 采集时间 | createTime | datetime | No |
| 采集值 | value | float(5,2) | No |
| 阈值 | limitValue | float(5,2) | No |
| 湿度状态 | state | int (1) | No |

5．有毒气体和重金属监控表（gas），该表用于存储监测有毒气体和重金属的相关信息，具体的结构，如表4-3有毒气体和重金属监控表所示：

表4-3有毒气体和重金属监控表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| 湿度ID | id | varchar (32) | Yes |
| 采集时间 | createTime | datetime | No |
| 类型 | bigType | Int(1) | No |
| 序号 | subIndex | Int(1) | No |
| 采集值 | value | float(5,2) | No |
| 阈值 | limitValue | float(5,2) | No |
| 湿度状态 | state | int (1) | No |

6．位移采集记录表，有四张表构成，位移采集汇总记录表（displacement），移采集单条记录表（displacement\_every），移采集值表（displacement\_value），测量点记录表（point\_enum），该表用于每次位移采集相关信息，具体的结构，如表下所示：

表4-3位移采集汇总记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| 湿度ID | id | varchar (32) | Yes |
| 采集时间 | createTime | datetime | No |
| 备注 | remark | text | No |

表4-3移采集单条记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| id | id | varchar | Yes |
| 组ID | groupId | varchar | No |
| 测量点 | point | int | No |
| 本次测值 | thisValue | varchar | No |
| 上次测值 | beforeValue | varchar | No |

表4-3移采集值表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| id | id | varchar | Yes |
| x值 | x | float | No |
| y值 | y | float | No |
| z值 | z | float | No |

表4-3测量点记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| id | id | varchar | Yes |
| code | code | int | No |
| 测量线 | line | int | No |
| 测量点 | point | int | No |
| 测点名称 | name | varchar | No |

7．系统测量异常记录（sys\_warn），该表用于存储系统测量异常记录，具体的结构，如表4-4系统测量异常记录表所示：

表4-4系统测量异常记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段含义 | 字段名称 | 类型及长度 | 主键 |
| id | id | varchar | Yes |
| 创建时间 | createTime | datetime | No |
| 修改时间 | updateTime | datetime | No |
| 操作人 | optionUser | varchar | No |
| 异常类型 | exceptionLocation | int | No |
| 异常描述 | exceptionDescription | int | No |
| 异常状态 | exceptionState | int | No |
| 异常值 | exceptionValue | float | No |
| 阈值 | limitValue | float | No |
| 备注 | remark | varchar | No |

# 5 系统的实现

## 5.1系统部署方案

本系统的部署全部基于云平台，服务器使用的是腾讯云的ECS和RDS。并且系统集成Nginx 、Tomcat、 JDK、 MySql等JAVA运行环境。

服务端部署：

1．安装配置MySql数据库

2. 打包项目并传到Linux机器上

为解决Chrome浏览系调用摄像头兼容问题，本系统全站使用https协议，HTTPS 协可以理解为HTTP+SSL/TLS， 即 HTTP 下加入 SSL 层，HTTPS 的安全基础是 SSL，因此加密的详细内容就需要 SSL，用于安全的 HTTP 数据传输[9]。

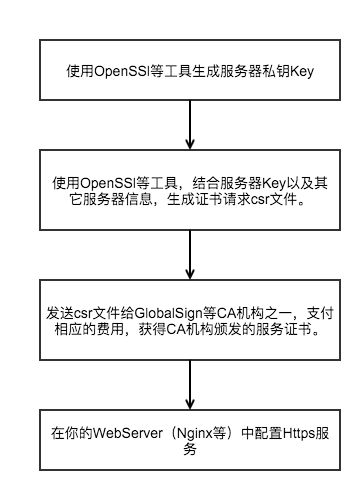


图5-1 Https使用流程的流程

## 5.2系统功能实现

依据需求分析，本系统实现了全部功能，并且保证了每一个功能点可用性和实用性，主要完成了测量中心模块、视频监控模块、人员管理模块、系统关系模块等。

各个角色登录时，输入用户账号、用户密码，点击登录，就可以向后台发送请求。

### 5.2.1主页模块

主页模块主要登陆页面、系统信息，测量中心等组成。登录页面如5-4所示。

图5-4 尾矿库安全云管理系统 登录页

### 5.2.2 测量中心模块

测量中心模块主要有温度测量、湿度测量功能以水位测量，有毒气体检测，重金属含量测量，位移测量等。

测量数据可动态更新，在客户端第一次访问该服务器时，会建立一个长连接，服务器保存所有客户端的会话，当测量值发生变化时，将变化信息推送给所有客户端。在每次传输时，只传递数据，在客户端监听该数据变化，发生改变时动态更新表格极大地减少了不必要的的header等信息的发送，减轻服务器的压力。

报表导出使用了POI的Java的api进行操作Excel，通过读取模板文件，将数据写入指定单元格，并一一对应。

支持对单个测量类别设置阈值，测量值草除阈值，在系统管理里自动存入异常数据。

（1）温度测量模块负责展示实时采集来的温度数据。并可设置阈值线。

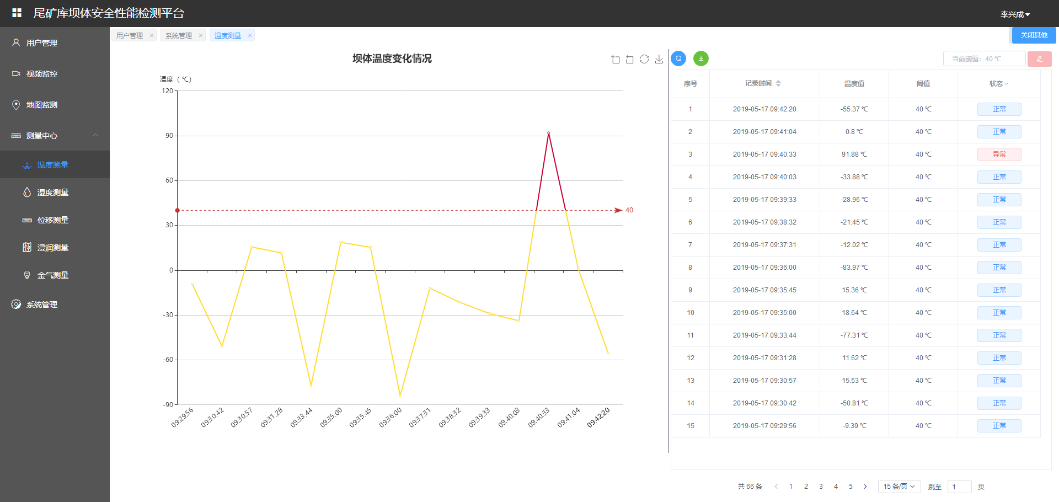


图5-11 温度测量

（2）湿度测量模块负责展示实时采集来的湿度数据，并可设置阈值线。



图5-12 湿度测量

（3）浸润测量模块负责展示坝体水位线和尾矿所在地区的降雨量和水流量信息，方便管理人员及时开展防洪防汛工作。

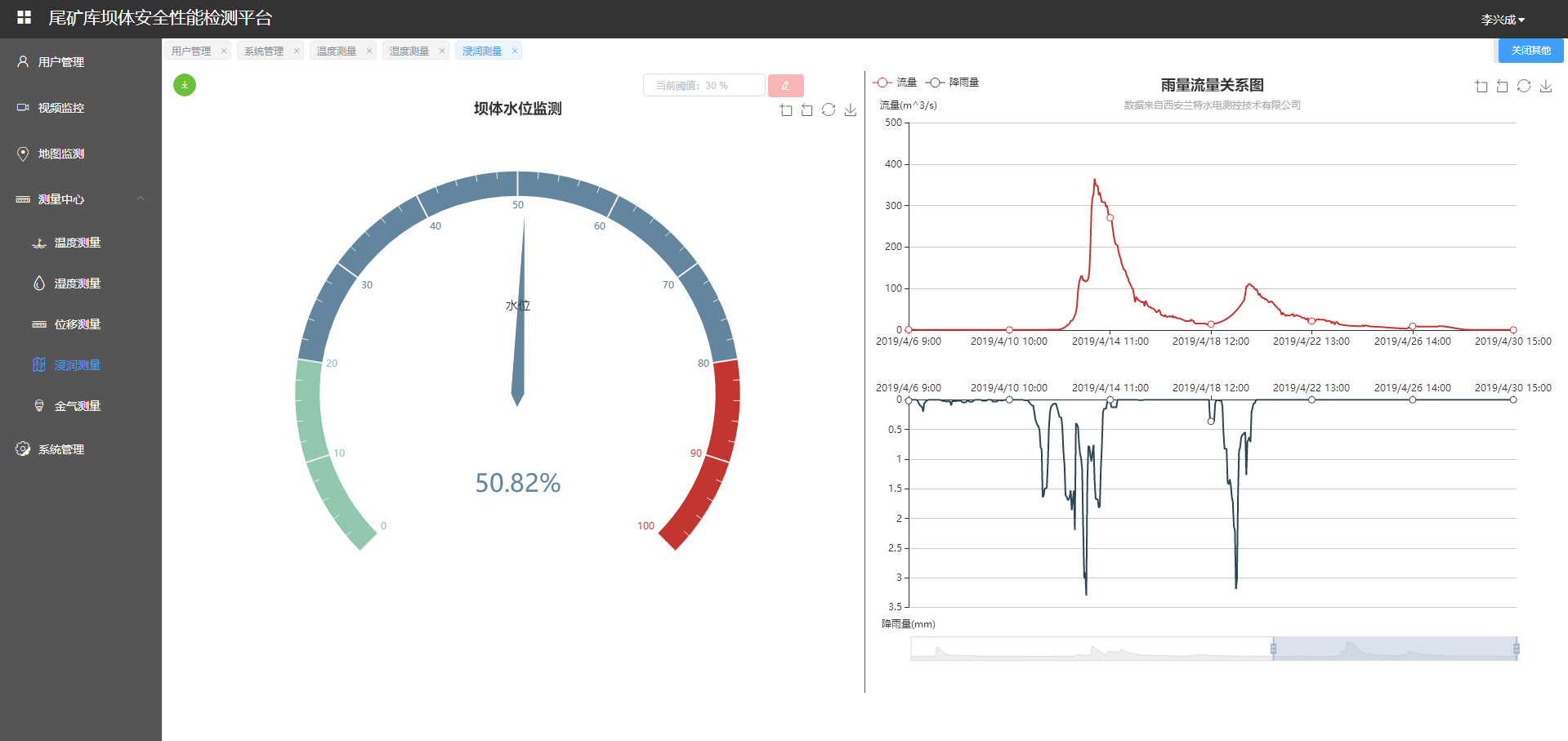


图5-13 浸润测量

（4）金气测量模块负责展示坝体有毒有害气体占比和重金属元素含量，帮助工作人员及时排除危险，避免不必要的人员伤害。



图5-13 金气测量

（5）位移测量模块负责展示坝体多个点的坝体位移x，y，z各个轴线上位移的变化情况。

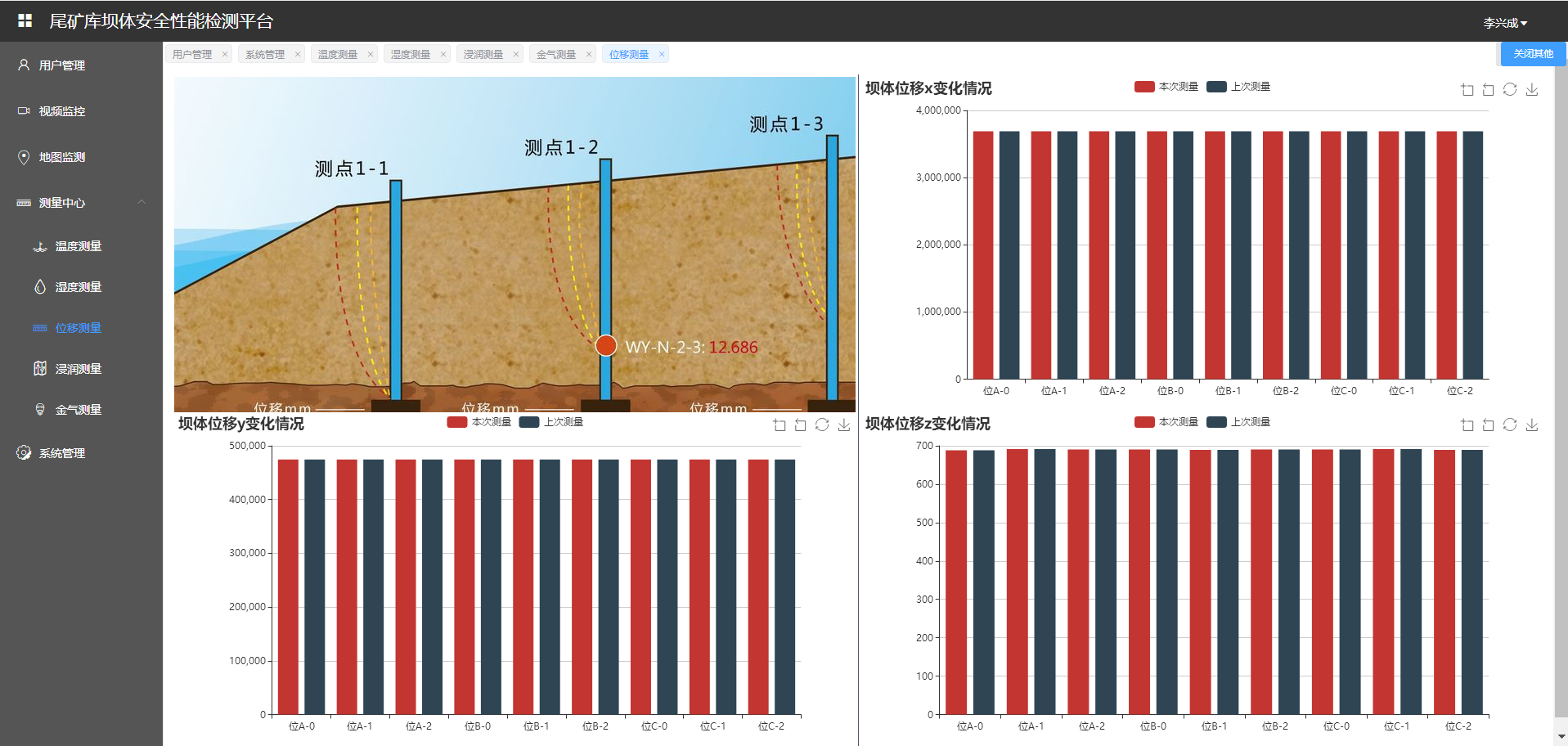


图5-13 位移测量

### 5.2.4视频监控模块

视频监控模块的功能主要有实时监控区域变化情况。并支持多视角显示。

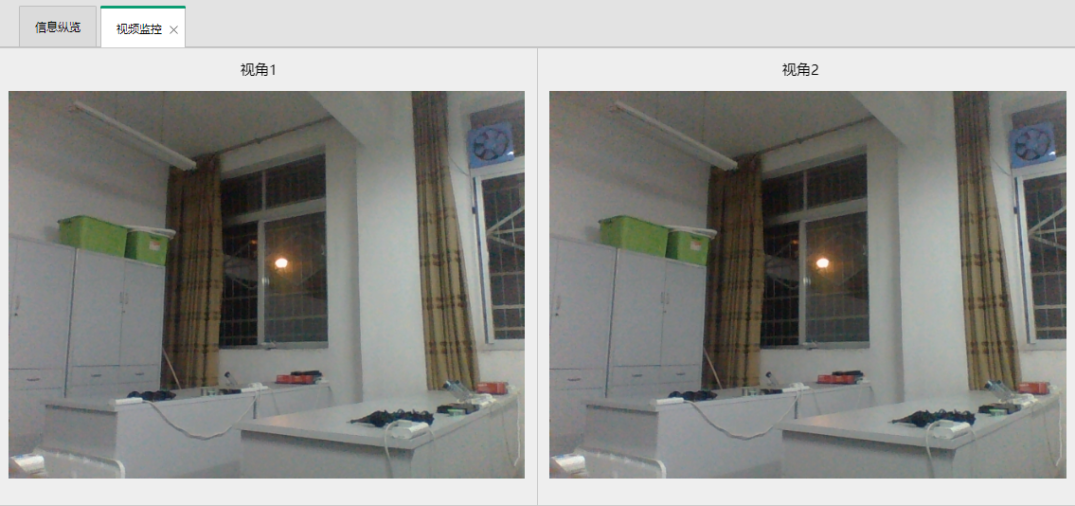


图5-11视频监控页面图

### 5.2.5 监控地图模块

用户管理模块，根据需求被实际通过调用API实现周围环境的地图监控功能，通过百度地图进行实时定位，将GPS采集到的数据可视化展示给用户[10]，方便在突发情况来临前做战略部署或在特殊时期做应急指挥，如图5-12所示

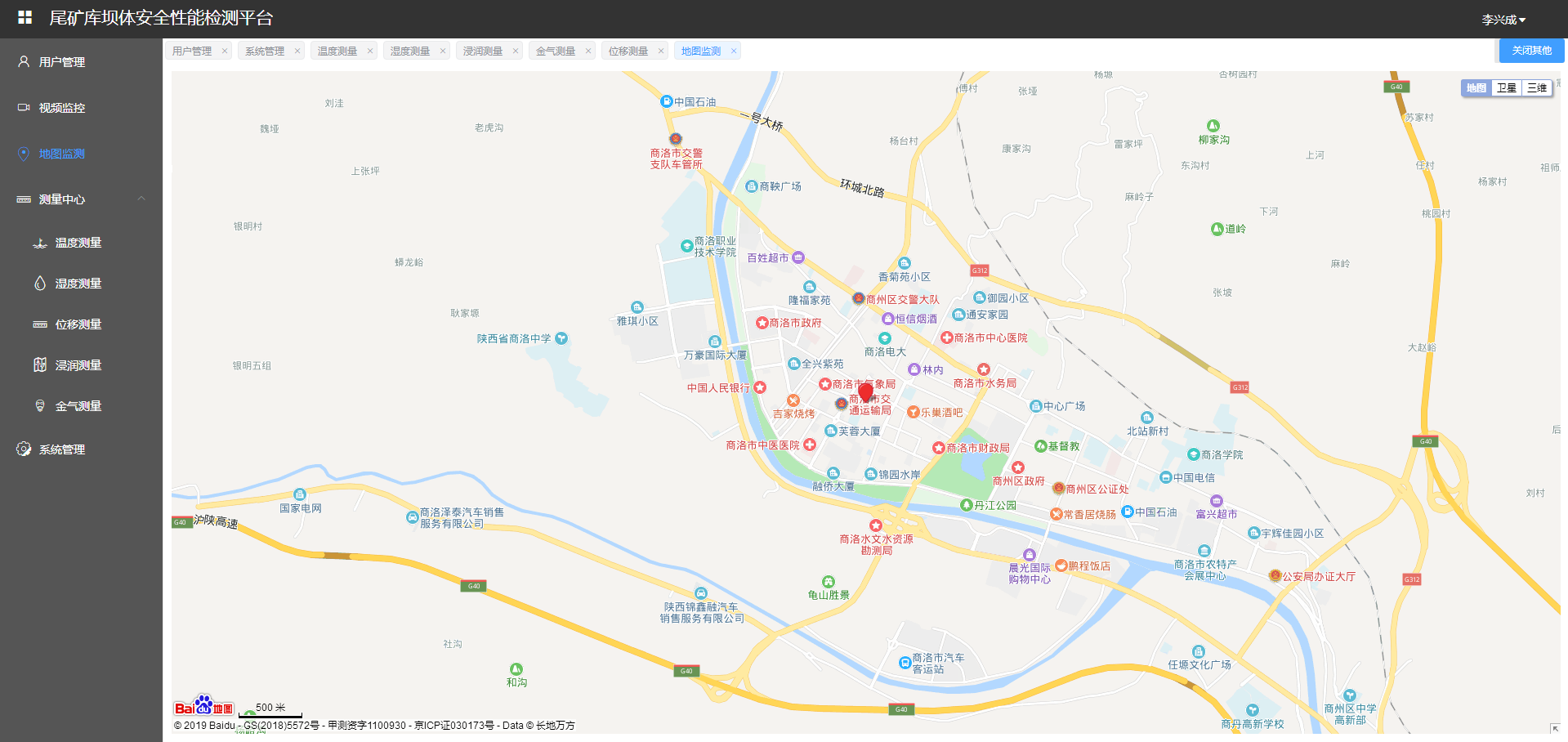


图5-12监控地图页面

### 5.2.6用户管理模块

用户管理模块，根据需求本项目实现了用户的增加，删除，修改，查询和权限操作等功能。普通用户只能查询管理员以进行其他的所有操作，具体的实现页面如图5-12所示。

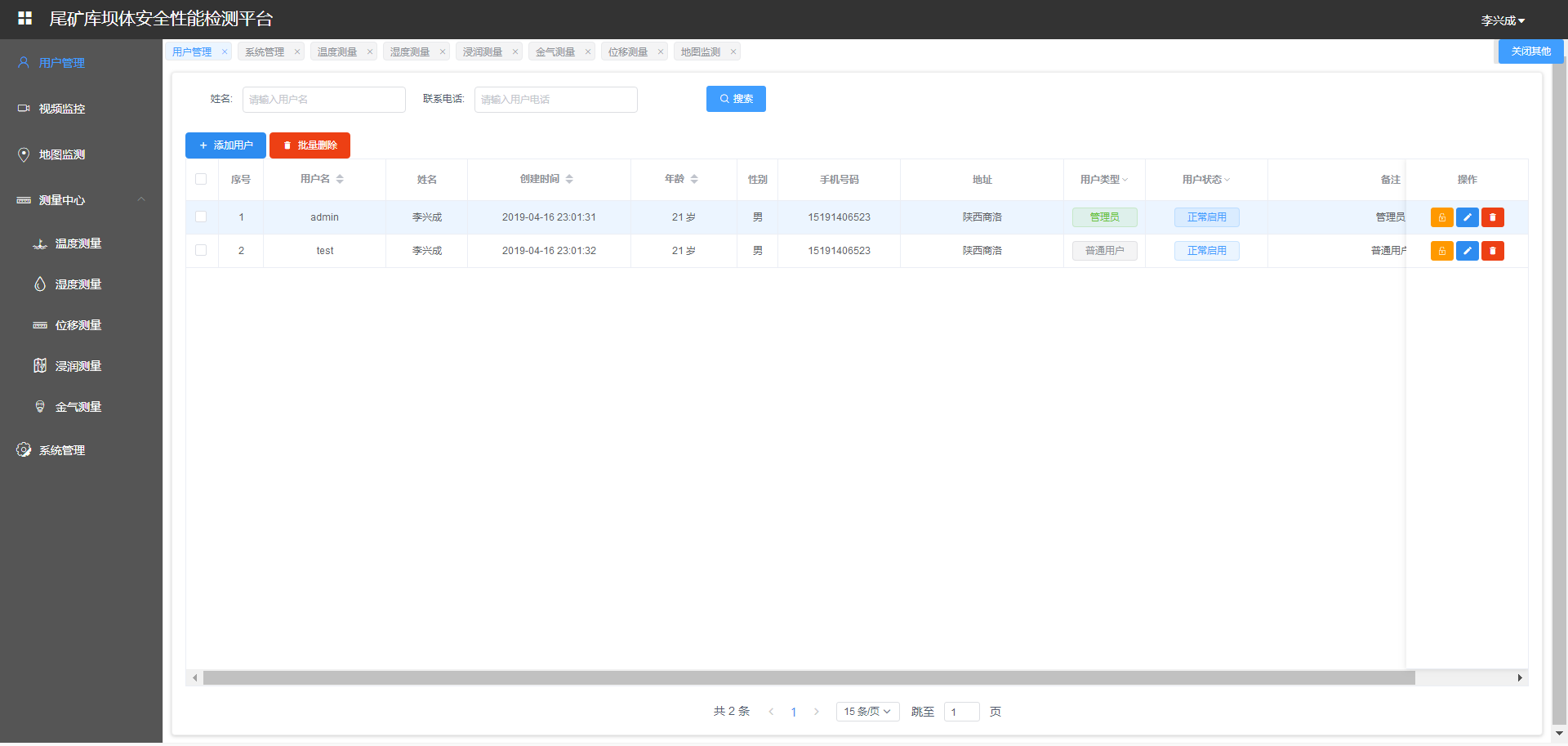


图5-13用户管理页面图

### 5.2.7系统管理模块

系统管理模块主要记录了系统运行时测量到的异常值，并可以进行状态标记和备注，

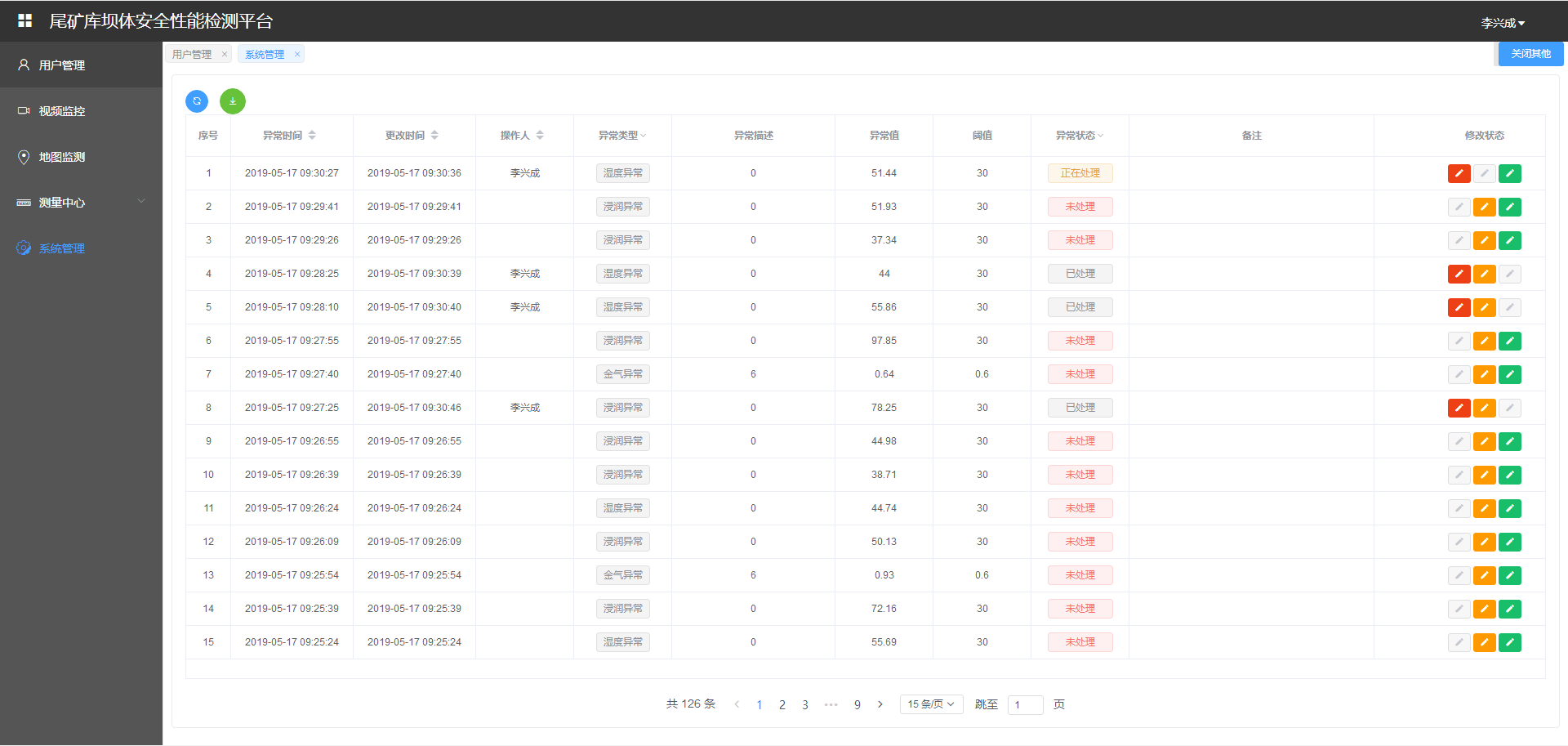


图5-14 系统管理 页面图

# 6 系统的测试

本项目的代码编写历时一个多月时间，为了能使系统达到可以线上部署的要求，就必须对系统进行各种各样的测试和BUG修改工作，所以很大一部分时间都用在了测试和完善bug中。

常用的测试方法有：单元测试、黑白盒测试、压力测试、动态测试等。本项目中使用了单元测试和Jmeter压力测试。

## 6.1单元测试

单元测试（Unit Testing），是最微小的测试方法，最大的特征是单元测试是有程序员来编写的而不是由专业的测试人员编写。

## 6.2 OneAPM CPT云压测

压力测试又称负载测试，是一个应用程序能够承担最大的的负荷量。压力测试与服务器的硬件设备密不可分，所以不同的代码在不同的硬件机器上可能会出现不能的结果。

OneAPM CPT 压测节点遍布全球,节点分部图如6-2所示

图6-2 OneAPM CPT压测节点

在压测过程中，可对事务耗时深入到各应用组件, 自定义事务深入到执行线, 数据库监控下钻至 SQL 执行计划, 误定位到代码行及错误堆栈信息，文章的篇幅有限，这里只展示了菜品列表的页面，该页面的数据量最大，能够更好的体现系统的性能，测试结果图如图6-3所示。



图6-3 OneAPM CPT 压测结果

压力测试结果说明：

本次测试每秒启动1193个用户线程访问，相当于每秒的访问量。在此环境下所有

线程错误率0.03%，90%的线程在2秒内完成，95%在5秒内完成，平均在1秒内完成，平均每秒内8221.5次点击。测试结果通过，本尾矿库安全管理系统完全可以满足一般企业正常业务使用。