



本科生毕业论文（设计）

基于 J2EE 的尾矿库安全云管理系统

The Tailings Security Cloud Management System
Based on J2EE

专 业_____电子信息工程_____

姓 名_____封 劲_____

学 号_____14037304_____

指 导 教 师_____王 博_____

完 成 时 间_____2018.6_____

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包括其他人已经发表或撰写过的研究成果，也不包含为获得商洛学院或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：_____ 日期：_____

关于论文使用授权的说明

本人完全了解商洛学院有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

☐公开 ☐保密（_____年_____月）（保密的学位论文在解密后应遵守此协议）

签名：_____ 导师签名：_____ 日期：_____

基于 J2EE 的尾矿库安全云管理系统

摘 要：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。

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等级评估、日常维护等提供决策支持。系统主要实现的功能有：尾矿库信息数据纵览，尾矿库温湿度，浸润线，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。主要采用 Java 语言开发，应用 Spring 框架和 SpringMvc 分层的设计思想，采用 Mysql 关系型数据库存储数据，使用 JQuery,Ajax,EasyUI,Echarts 等前端技术开发用户交互页面，使用 POI 技术开发报表功能。通过这些功能实现了一个高效的、稳定的、实用的云管理系统。

关键字：管理系统；J2EE；尾矿库；数据可视化；

The Tailings Security Cloud Management System Based on J2EE

Abstract: With the informationization advancement speeding up unceasingly, the Internet transmission capacity increasing, now gradually developed for data aggregation and all kinds of software stored in the "network cloud" (a large processing center) in unified processing, users need only a terminal can complete a series of requirements.

A set of safety cloud management system of web tailings pond is designed to provide decision support for mine managers to carry out tailings grade evaluation and daily maintenance. Information system is mainly the functions are: tailings data survey, tailings, the temperature of saturation line, water level, the poisonous and harmful gas, heavy metal element data monitoring, abnormal alarm, video monitoring, user management, system information system exception logging process, generate reports, and other functions. Mainly USES the Java language development, application and the Spring framework for SpringMvc layered design ideas, using Mysql relational database to store data, using JQuery, Ajax, EasyUI, Echarts front page technology to develop user interaction, such as using POI technology development reporting features. Through these functions, an efficient, stable and practical cloud management system is realized.

Key words: Management System; J2EE; Tailings; Data Visualization;

目 录

1 绪论.....	1
1.1 选题背景及意义.....	1
1.2 尾矿库管理系统概述.....	1
1.3 云平台的定义.....	1
1.4 基于 J2EE 的尾矿库安全云管理系统的特点.....	2
1.5 本课题要解决的问题.....	2
2 相关开发技术与开发、运行环境.....	2
2.1 JQuery EasyUI 技术.....	2
2.2 MyBatis 技术.....	3
2.3 Druid 技术.....	3
2.4 Spring mvc 框架.....	4
2.5 ECharts 开源技术.....	4
2.5.1 ECharts 提供丰富的图表类型.....	5
2.5.2 ECharts 数据驱动特点.....	5
2.6 Eclipse 开发工具.....	5
2.7 关系型数据库 RDS.....	5
2.8 开发环境.....	6
2.9 运行环境.....	6
3 项目需求分析.....	7
3.1 系统的需求分析.....	7
3.2 系统的总体需求图.....	7
3.3 系统的性能需求.....	8
4 系统设计.....	8
4.1 系统设计的原则.....	8
4.2 系统概要设计.....	9
4.3 系统模块设计.....	9

4.3.1 主页模块设计.....	10
4.3.2 测量中心模块设计.....	10
4.3.3 视频监控模块设计.....	10
4.3.4 人员管理模块设计.....	11
4.3.5 系统管理模块设计.....	11
4.4 数据库设计.....	11
5 系统的实现.....	14
5.1 系统部署方案.....	14
5.2 系统功能实现.....	14
5.2.1 主页模块.....	15
5.2.2 测量中心模块.....	17
5.2.3 测量中心模块数据采集接口（http）.....	20
5.2.4 视频监控模块.....	21
5.2.5 监控地图模块.....	21
5.2.6 用户管理模块.....	21
5.2.7 系统管理模块.....	22
6 系统的测试.....	22
6.1 单元测试.....	22
6.2 OneAPM CPT 云压测.....	23
7 总结和展望.....	24
参考文献.....	25
致 谢.....	26

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,用户不再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 Spring、Spring MVC、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 Chrome、Firefox 以及 IE9 等市面上常见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 Easyui 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 JQuery EasyUI 技术

Jquery EasyUI 用 HTML 标记定义页面，并且它完美的支持 H5 页面的完整框架。使用 JQuery EasyUI 可以减少用户大量代码编辑的重复工作，做到与 jQuery，“多做一点，少写一点”相同的理念。为前端开发人员提供便捷的服务，减少开发周期。

Jquery EasyUI 的优点：

Jquery EasyUI, 是一个基于 jQuery 的前端开发的 JS 插件库, 为应用开发提供必备基础。为前端开发者, 提供了大多数 UI 控件, 如:

panel,tabs,layout,menu,link/menubutton,form,comobox,calendar>window,datagrid,tree 等等。

Jquery EasyUI 官方还提供几种不同的主题风格, 供开发者选用。

Jquery EasyU 主要有以下优点:

1. 基于 jQuery 用户界面插件的集合;
2. 为一些当前用于交互的 JS 应用提供必要的功能;
3. EasyUI 支持 html 和 javascript 两种渲染方式;
4. 通过 data-options 属性可以完美支持 HTML5;
5. 开发产品时可节省时间和资源;
6. 简单, 但很强大;
7. 支持扩展, 可根据自己的需求扩展控件;
8. 不断的版本递进来修复各种 Bug。

2.2 MyBatis 技术

MyBatis 起源于 apache 开源项目 iBatis, 是一款优秀的持久层框架, MyBatis 支持定制 SQL、指出存储过程以及高级映射^[5]。MyBatis 封装了 JDBC 的所有功能, 可以使用简单的 XML 标记, 或注解来配置和映射原生信息, 将接口(interface)和 Java 的 POJOs(Plain Old Java Objects)映射成数据库中的数据^[6]。

2.3 Druid 技术

Java 在开发的 web 应用程序的时候, 数据库的操作是必须的, 而 JDBC 是一定会被用到。但是 JDBC 是一个开销很大的线程, 在程序中不可能对 JDBC 进行无限制的创建, 所以一般都会选择一款数据库连接池, 本尾矿库安全云管理系统使用了 Druid 数据库连接池。

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。

2.4 Spring mvc 框架

MVC 是一种设计模式，SSM 架构项目^[7]中通过 mvc 强制性地 把应用系统拆分为视图展示，逻辑处理和流程控制三个环节。

MVC 将应用系统分成三个核心模块：model-模型、view-视图和 controller-控制器，他们相互连接又分别承担不同的任务和处理逻辑，如图 2-1 所示。

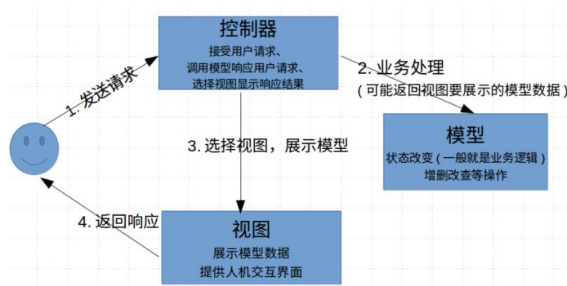


图 2-1 MVC 模型图

Spring MVC 是一个模型 - 视图 - 控制器（MVC）的 Web 框架建立在中央前端控制器 servlet（DispatcherServlet），它负责发送每个请求到合适的处理程序，使用视图来最终返回响应结果的概念。Spring MVC 是 Spring 产品组合的一部分，它享有 Spring IoC 容器紧密结合 Spring 松耦合等特点，因此它有 Spring 的所有优点。

SpringMVC 架构图 2-2 所示：

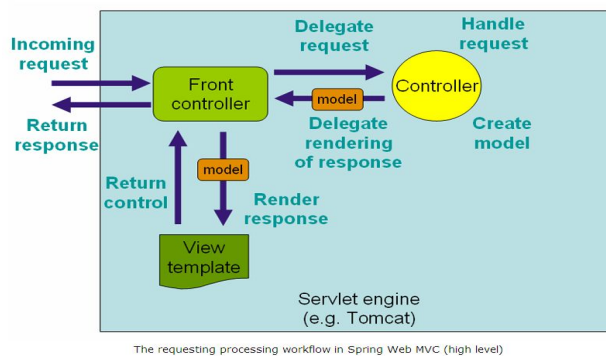


图 2-2 SpringMVC 架构图

2.5 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。

2.5.1 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。

2.5.2 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。

2.6 Eclipse 开发工具

Eclipse 是跨平台的著名自由集成开发环境（IDE），最初主要用来 Java 语言开发。Eclipse 通过安装不同的插件可以支持不同的计算机语言，如 C++和 Python 等 IDE。Eclipse 的本身只是一个框架平台，但是因为众多插件的支持使得 Eclipse 拥有其他功能相对固定的开发软件很难具有的灵活性，许多软件开发商以 Eclipse 为框架开发定制自己的 IDE。

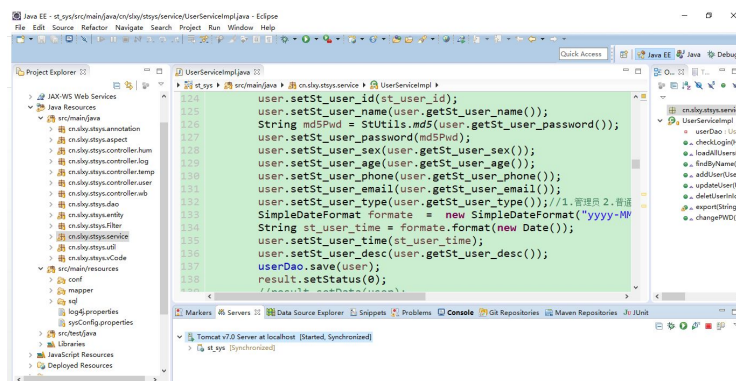


图 2-3 Eclipse 工作界面图

2.7 关系型数据库 RDS

本项目中使用的数据库，为腾讯云提供的 RDS（基于 mysql5.7），没有使用本地物理数据库原因是本地数据库需要搭建，而且没有优秀的数据库管理员进行维护，数据库的正常运行将非常困难，使用腾讯云 RDS 的优势：

1. 数据库的日常维护不再需要开发者进行，节省人力成本。
2. 任何系统数据的备份是必须的，腾讯云数据库支持自动备份功能。使用腾讯云后数据库的备份不用人工进行备份。

3. 支持设置 ip 黑白名单的功能，可以从数据库层将非法的 ip 地址访问阻挡起来，大大的降低了非法访问对服务器和数据的压力。

2.8 开发环境

为更好的开发系统，需要提前配置好前端后端和运行端的相关环境，前端配置如表 2-1 所示，后端配置表如表 2-2 所示

表 2-1 前端开发环境

前端开发环境
处理器：2.6 GHz Intel Core i7
内存：16 GB 2133 MHz LPDDR3
系统：win10
开发 IDE：HBuilder

表 2-2 后端开发环境

后端开发环境
处理器：2.6 GHz Intel Core i7
内存：16 GB 2133 MHz LPDDR3
开发 IDE：Eclipse oxygen Release
数据库：MySQL5.7
JDK：JDK 1.7

2.9 运行环境

测试运行环境使用腾讯云的 ECS 和 RDS，其规格如下：

机器配置：

操作系统 CentOS 6.5 64 位

CPU：1 核

内存：1GB

系统盘：disk-6coxybzh 50 GB

公网带宽：1Mbps

软件：

系统：CentOS 7.0 64 位

nginx 代理：nginx/1.8.0

数据库：腾讯云 RDS（MySQL 5.6.26）

JDK：JDK 1.7

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：

1. 测量中心，下设温度测量，湿度测量，水位测量，浸润测量，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。
2. 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。
3. 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。
4. 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。
5. 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。
6. 系统管理：本系统可在线查看系统日志，包括登陆，修改删除等信息的记录，绑定到操作用户自身。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。

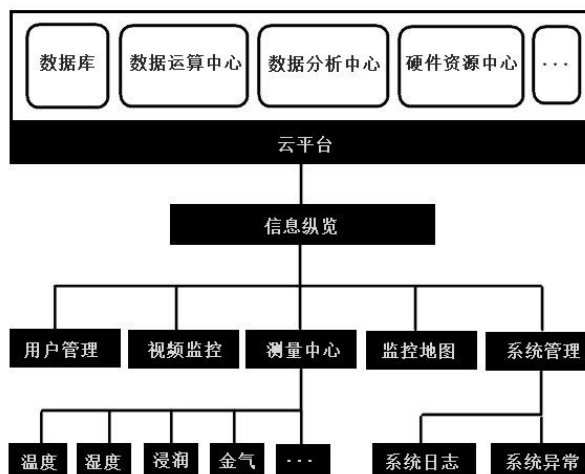


图 3-1 项目总体需求图

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，低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。

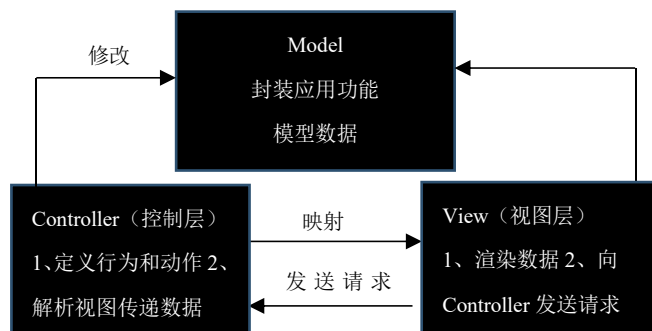


图 4-1 MVC 模型图

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。

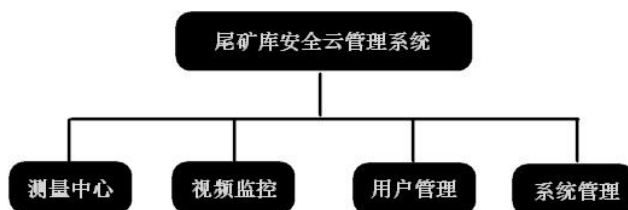


图 4-2 系统应用功能图

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 主页模块设计

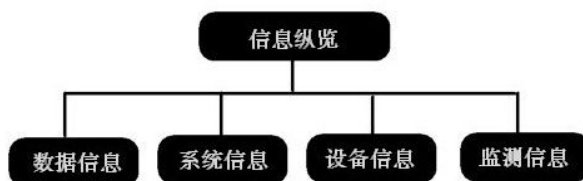


图 4-3 主页模块图

信息纵览模块即为主页模块，主要成上下两个区域暂时，上边区域主要展示了监测数据的异常和已处理异常的状态，展示系统的人员注册数量，展示软硬件连接的状态和一个日历插件。下边区域采用全图表化展示了系统运行时候的重要数据，采用图表的方式一目了然，提升用户体验。

4.3.2 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個模块，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最小值数据，最大值数据和平均值数据，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。

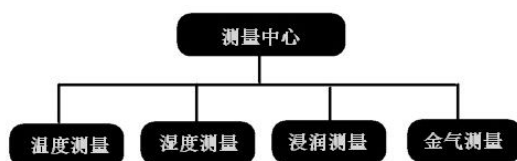


图 4-4 测量中心 模块图

4.3.3 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。



图 4-5 视频监控 模块图

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删改查，人员的权限分配，人员的报表报导出三个主要的功能构成，人员的增删改查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、修改和查询，可进行密码修改，报表导出等功能的使用，而普通用户只能对系统中的人员进行查询而不能做任何变更的操作。

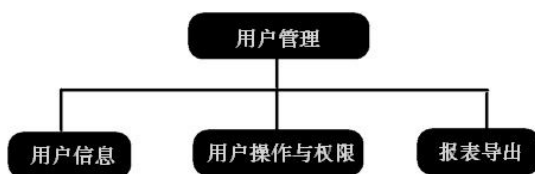


图 4-6 人员管理 模块图

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 log 用来记录用户的登录信息，记录用户的行为信息和行为过程，做到对系统操纵留痕的要求，并在用户操作时候获取其 ip 地址和所在城市，提升系统安全性校验。系统异常部分记录系统在运行时候所出现的异常信息，并将其记录，方便开发人员迅速定位系统 bug，从而迅速的处理，将用户的损失降到最低，快速恢复系统的正常使用，保障使用者的人力和财力安全。

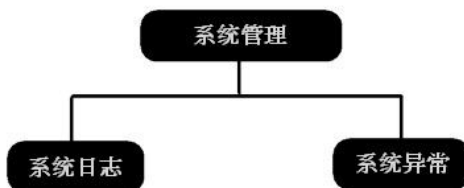


图 4-7 系统管理 模块图

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 st_sys 的数据库,编码为 UTF-8, st_sys 数据库中包含下面几张数据表:

1. 用户表 (st_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
用户 ID	st_user_id	varchar (100)	Yes
帐号	st_user_name	varchar(100)	No
姓名	st_user_nick	varchar(50)	No
密码	st_user_password	varchar (70)	No
性别	st_user_sex	varchar (30)	No
年龄	st_user_age	varchar(30)	No
电话	st_user_phone	varchar(80)	No
邮箱	st_user_email	varchar(70)	No
类别	st_user_type	varchar(50)	No
注册时间	st_user_time	varchar(30)	No
备注信息	st_user_desc	varchar(100)	No

表中的用户类别为一个枚举, 其枚举含义为 1: 男 2: 女

2. 湿度表 (st_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
湿度 ID	st_humidity_id	varchar (100)	Yes
湿度值	st_humidity_data	varchar (30)	No
采集时间	st_humidity_time	varchar (50)	No
湿度状态	st_humidity_state	varchar (30)	No
备注信息	st_humidity_desc	varchar (100)	No

表中的湿度状态为一个枚举, 其枚举含义为 0: 正常 1: 异常 2: 已处理

3. 系统状态信息（st_syslog），该表用于存储系统运行时候数据，具体的结构，如表 4-4 系统 log 表所示：

表 4-4 系统 log 表

字段含义	字段名称	类型及长度	主键
主键	st_sysLog_id	Varchar50)	Yes
操作方法	st_sysLog_operMethod	varchar(100)	No
方法描述	st_sysLog_operDesc	varchar(30)	No
操作人	st_sysLog_operName	varchar(20)	No
操作时间	st_sysLog_operTime	varchar(50)	No
IP 地址	st_sysLog_ipAdder	varchar(20)	No
操作城市	st_sysLog_cname	varchar(100)	No

4. 系统异常信息表（st_sysexclg），该表用于系统运行时异常信息，具体的结构，如表 4-5 系统异常表所示：

表 4-5 系统异常表

字段含义	字段名称	类型及长度	主键
异常 ID	st_sysExcLog_id	varchar (50)	Yes
异常类型	st_sysExcLogg_type	varchar(30)	No
异常信息	st_sysExcLog_info	varchar(30)	No
异常定位	st_sysExcLog_method	varchar(50)	No
方法描述	st_sysExcLog_methodDesc	varchar(30)	No
操作人	st_sysExcLog_operName	varchar(20)	No
操作时间	st_sysExcLog_operTime	varchar(50)	No
IP 地址	st_sysExcLog_ipAdder	varchar(20)	No
操作城市	st_sysExcLog_cname	varchar(100)	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 协议可以理解为 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]。

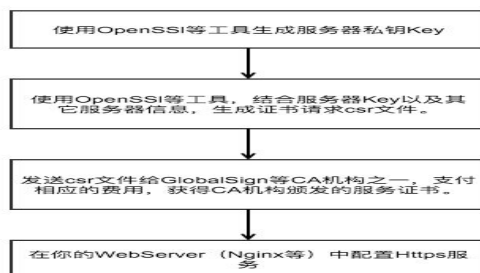


图 5-1 Https 使用流程的流程

5.2 系统功能实现

依据上一章的需求分析，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，主要完成了主页模块、测量中心模块、视频监控模块、人员管理模块、系统关系模块。本系统的欢迎页面如图 5-2 所示：



图 5-2 尾矿库安全云管理系统 欢迎页

各个角色登录时，输入用户账号、用户密码和随机生成的验证码，点击登录，就可以向后台发送请求。

数据库的操作使用 DBCP 数据库连接池进行处理，采用 mybatis 自动根据数据库生成 model，使用自己编写 sql 进行数据库操作。

操作数据库之前首先要连接数据库，并且将数据库连接对象交 spring 管理，具体配置如图 5-3 所示：

```
<!-- 定义dbcp组件DataSource -->
<bean id="dbcp" class="org.apache.commons.dbcp.BasicDataSource"
    destroy-method="close">
    <property name="username" value="root"></property>
    <property name="password" value="497555212"></property>
    <property name="driverClassName" value="com.mysql.jdbc.Driver"></property>
    <property name="url" value="jdbc:mysql://localhost:3306/st_sys?useUnicode=true&characterEncoding=utf8"></property>
</bean>
<!-- 定义SqlSessionFactory -->
<bean id="ssf" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dbcp"></property>
    <property name="mapperLocations" value="classpath:mapper/*.xml"></property>
<!-- 配置分页插件 -->
    <property name="plugins">
        <array>
            <bean class="com.github.pagehelper.PageHelper">
                <property name="properties">
                    <value>
                        dialect=mysql
                        reasonable=true
                    </value>
                </property>
            </bean>
        </array>
    </property>
</bean>
<!-- 定义MapperScannerConfigurer -->
<bean id="mapperScanner" class="org.mybatis.spring.mapper.MapperScannerConfigurer">
    <property name="basePackage" value="cn.slxy.stsys.dao"></property>
</bean>
</beans>
```

图 5-3 DBCP 连接池 mybatis 配置

在连接好数据库后，就可以直接取到数据库的连接对象，然后根据数据库生成的 model 就可以操作数据库的增删改查。

5.2.1 主页模块

主页模块主要登陆页面、系统数据全局纵览，系统信息，日历等组成。登录页面如 5-4 所示。



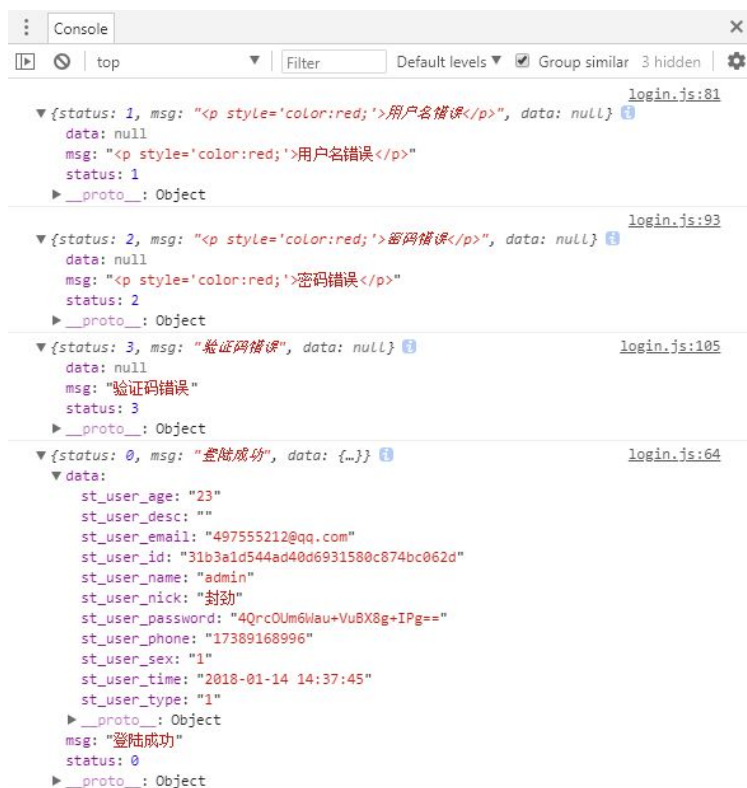
图 5-4 尾矿库安全云管理系统 登录页

各个角色登录时，输入账号、密码、选择角色类型，点击登录，就可以向后台发送请求如图 5-5

```
// 用户登陆
function userLogin() {
    if ($('#login_form').form('validate')) {
        // 获取请求数据
        var name = $("#count").val().trim();
        var password = $("#pwd").val().trim();
        var vCode = $("#vCode").val().trim();
        var cip = returnCitySN.cip;
        var cname = returnCitySN.cname;
        setCookie("cip", cip, "h1");
        setCookie("cname", cname, "h1");
        // 发送ajax请求
        $.ajax({
            url : path + "/user/login.do",
            type : "post",
            //async:"false",
            cache : false,
            data : {
                "name" : name,
                "password" : password,
                "vCode" : vCode,
                "cip":cip,
                "cname":cname
            },
            dataType : "json",
            success : function(result) {
```

图 5-5 向后台发送请求

后台返回不同的信息，服务器会自动序列化成一个 json 返回前端处理，这个 json 的格式如图 5-6：



```
{status: 1, msg: "<p style='color:red;'>用户名错误</p>", data: null}
{status: 2, msg: "<p style='color:red;'>密码错误</p>", data: null}
{status: 3, msg: "验证码错误", data: null}
{status: 0, msg: "登陆成功", data: {...}}
```

图 5-6 返回 json 格式数据

登陆后即可看到系统工作面板如图 5-7 所示：

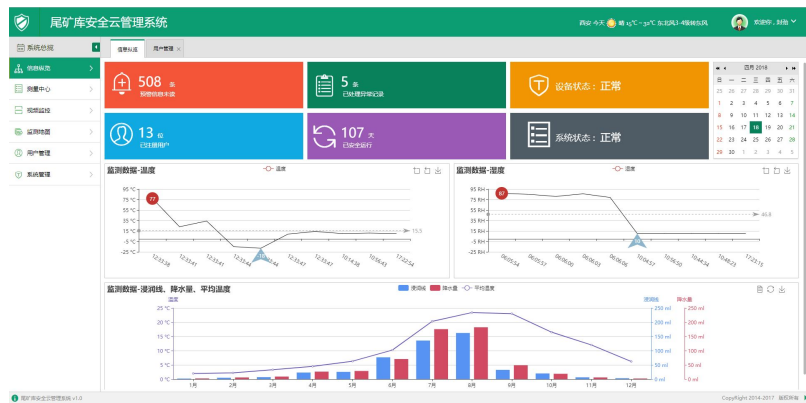


图 5-7 尾矿库安全云管理系统 主页

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等功能。页面的实现主要以左右分布为主。上部分为标题栏、左部为功能操作栏、右部为功能显示区。

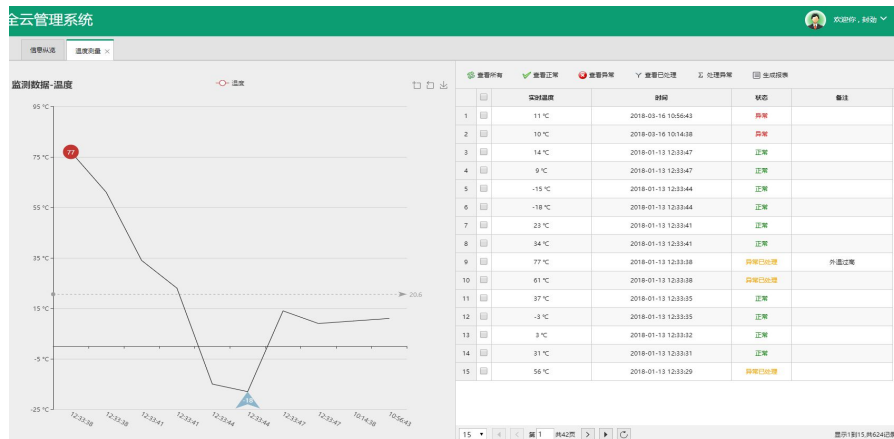


图 5-8 测量中心模块页面图

功能显示区左侧为右侧数据的可视化视图，并采用动态更新，每过 30s 向后台请求一次最新数据并做页面更新。

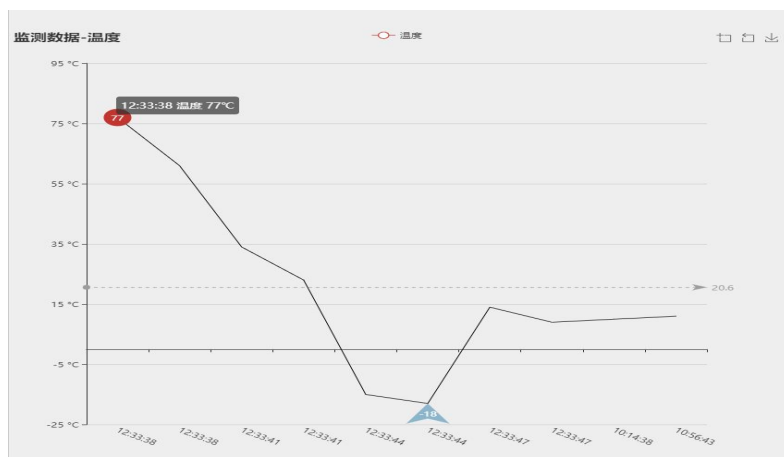


图 5-9 监测数据可视化

功能显示区右侧采用上下布局的方式，上侧为可操作区，可以进行查看所有数据，查看正常数据，查看异常数据，查看已处理数据，处理异常和生成报表等一系列操作。下侧为监测数据列表，具体表现方式如图 5-10 所示。

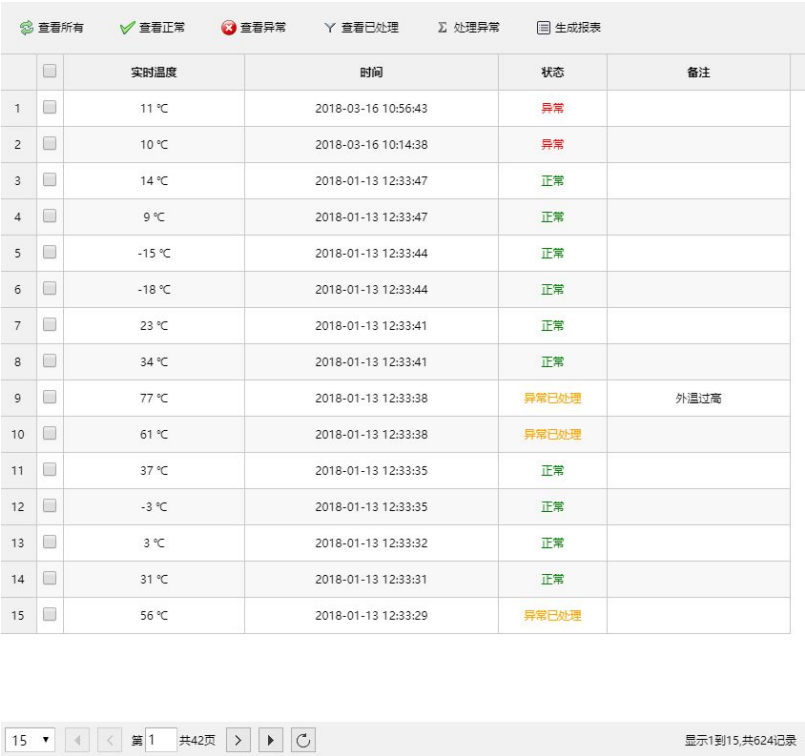


图 5-10 监测数据操作区页面图

具体的测量模块如图 5-11 温度测量，图 5-12 湿度测量，图 5-13 浸润测量，图 5-14 金气测量等：

温度测量模块负责展示实时采集来的温度数据，并突出展示实时最高温度，最低温度和平均温度。

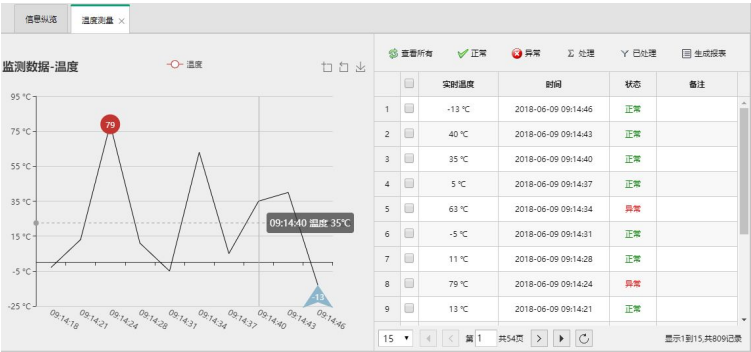


图 5-11 温度测量

温度测量模块负责展示实时采集来的湿度数据，并突出展示实时最高温度，最低温度和平均湿度。

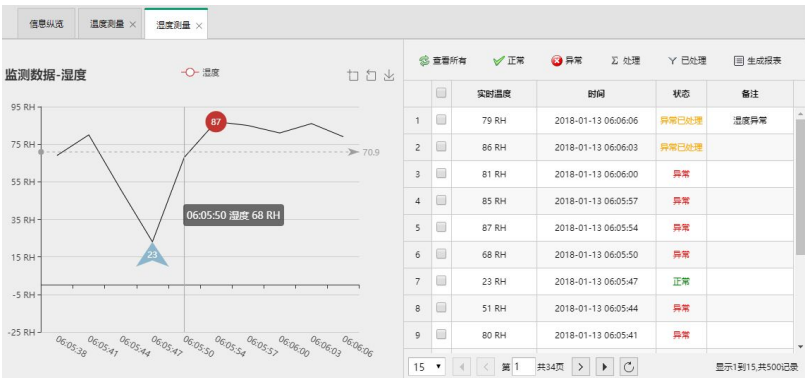


图 5-12 温度测量

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。

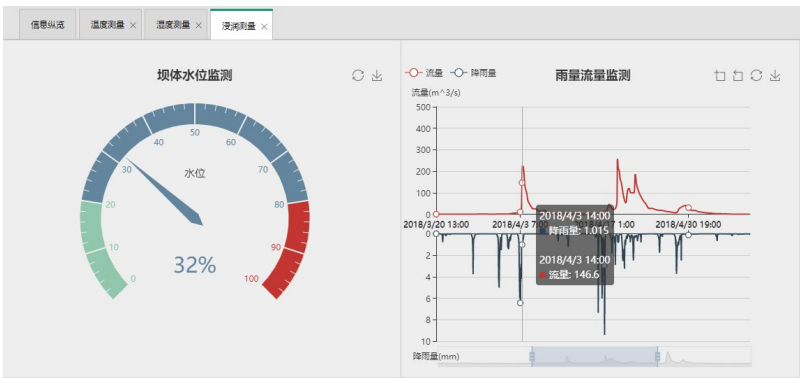


图 5-13 浸润测量

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。

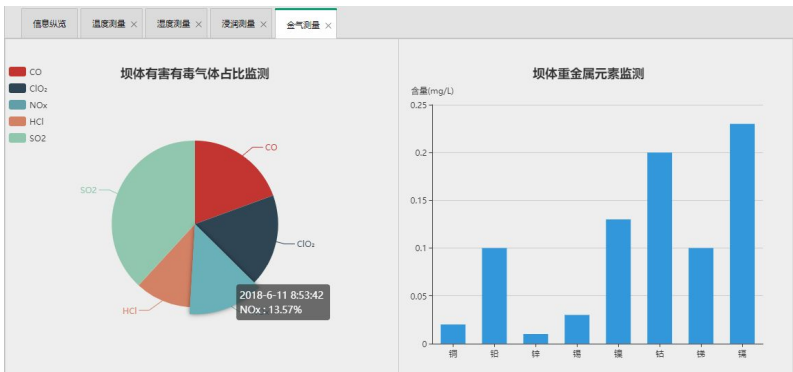


图 5-13 金气测量

5.2.3 测量中心模块数据采集接口（http）

本系统开放接口供第三方硬件设备接入，采集数据，采集视频，采集状态等内容。
以采集温/湿度数据为例做以接口说明：

温度采集：

接口：temp/addTempDates.do?st_temperature_data=10&st_temperature_state=1

参数：

参数	类型	说明
st_temperature_data	Int	温度值
st_temperature_state	Int	状态 0：正常 1：异常

返回值：{"status":0,"msg":"添加温度数据成功",
"data":{"st_temperature_id":"01d7f0a7acff403b09dc70581",
"st_temperature_data":"10",
"st_temperature_time":"2018-04-12 10:56:36",
"st_temperature_state":"1",
"st_temperature_desc":null}}

湿度采集：

接口：hum/addHumDates.do?st_humidity_data=10&st_humidity_state=1

参数：

参数	类型	说明
st_humidity_data	Int	湿度值
st_humidity_state	Int	状态 0：正常 1：异常

返回值：{"status":0,"msg":"添加湿度数据成功",
"data":{"st_humidity_id":"35fc94cece0c4cceb5b552a89",
"st_humidity_data":"10",
"st_humidity_time":"2018-04-12 11:48:52",
"st_humidity_state":"0",
"st_humidity_desc":null}}

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。

布局采用双视角页面，每次同时接入两个监控摄像头进行监控，具体的实现页面如 5-11 图所示。

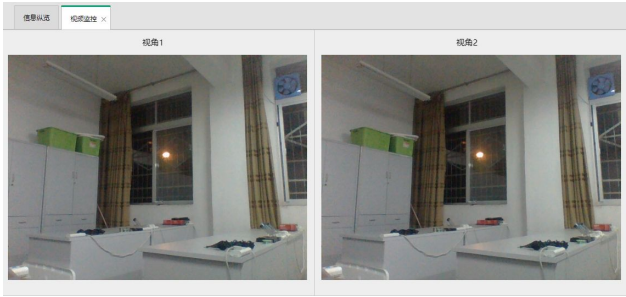


图 5-11 视频监控页面图

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，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

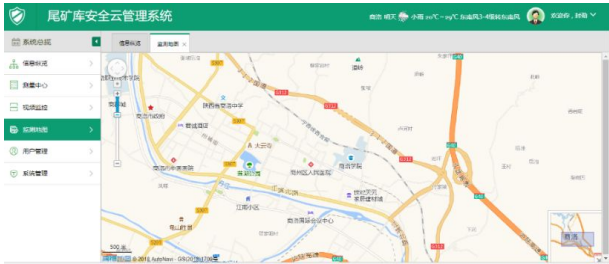


图 5-12 监控地图页面

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删改查，生成数据报表和权限操作等功能。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。

序号	姓名	性别	年龄	电话	地址	备注	操作
1	admin	男	35	1562584215	1562584215	系统管理员	删除
2	test	男	18	132124145	132124145	普通用户	删除
3	user	男	18	1562584215	1562584215	普通用户	删除
4	user	男	30	1888888888	1562584215	普通用户	删除
5	user	男	12	132124145	132124145	普通用户	删除
6	user	男	12	132124145	132124145	普通用户	删除
7	user	男	32	132124145	132124145	普通用户	删除
8	user	男	13	1562584215	1562584215	普通用户	删除
9	user	男	12	132124145	132124145	普通用户	删除
10	user	男	12	132124145	132124145	普通用户	删除
11	user	男	3	132124145	132124145	普通用户	删除
12	user	男	23	1562584215	1562584215	普通用户	删除

图 5-13 用户管理页面图

5.2.7 系统管理模块

系统管理模块主要实现了系统运行是 LOG 记录和系统异常时 LOG 记录的功能。

系统日志精确记录操作 ID，操作那个方法，操作方法的详细描述，操作人信息，操作时间点，操作时用户 IP 地址，操作人所在城市等信息。

操作ID	操作人	方法描述	操作时间	IP地址	操作城市
1	e4e15917b1740d6b6c2386a7923c	用户登陆	2018-04-02 10:33:49	123.138.87.158	陕西省西安市
2	77778d27b675488fa1b58f7a3b8b771	用户登陆	2018-03-21 14:39:52	123.138.87.158	陕西省西安市
3	6a06275e0f8757a107a88208a6d	退出用户登录	2018-03-19 13:47:03	123.138.87.158	陕西省西安市
4	d52a2720e13e4f93a75a038d6d1d8a	用户登陆	2018-03-19 09:08:18	123.138.87.158	陕西省西安市
5	4a4f9127a543a3a6a75a038d6d1d8a	更新用户信息	2018-03-07 09:03:29	123.138.87.158	陕西省西安市
6	a46921d5a6a3a75a038d6d1d8a	用户登陆	2018-03-02 21:46:05	115.183.33.140	中国
7	733a4a7b6a75a038d6d1d8a	用户登陆	2018-03-05 10:06:05	81.165.143.11	陕西省西安市
8	d5891d10898a6a75a038d6d1d8a	用户登陆	2018-03-05 09:45:04	123.138.87.158	陕西省西安市
9	c7b9374a021a2877b5a2177597	用户登陆	2018-03-02 21:28:13	115.183.33.140	中国
10	b192d467b34a6a75a038d6d1d8a	创建新用户登录	2018-03-02 15:07:28	123.138.87.158	陕西省西安市
11	4b1a41a469a48993a75a038d6d1d8a	用户登陆	2018-03-02 15:02:05	123.138.87.158	陕西省西安市
12	d8171779a6a3a75a038d6d1d8a	用户登陆	2018-03-02 15:16:05	45.32.43.146	日本
13	a3056a74714239a209f1b57b3a8	用户登陆	2018-03-02 15:13:23	123.138.87.158	陕西省西安市
14	71a8a2039a6a75a038d6d1d8a	退出用户	2018-03-02 15:13:08	123.138.87.158	陕西省西安市
15	72a079a1b6a4a6a75a038d6d1d8a	用户登陆	2018-03-02 15:07:42	123.138.87.158	陕西省西安市

图 5-14 系统日志 页面图

系统异常日志精确记录异常 ID，异常类型，异常信息，异常定位，异常方法描述，操作人，操作时间，操作时 IP 地址，操作人所在城市等信息。一旦系统出现问题，方便开发人员及时修改相关 BUG，保障系统正常运行

异常ID	异常类型	异常信息	异常定位	方法描述	操作人	操作时间	IP地址	操作城市
1	05Bc092a3647420f8ab07a09529333	java.lang.ArithmeticException	/ by zero	UserService	退出用户	2018-02-06 11:10:50	123.138.87.158	陕西省西安市

图 5-15 系统异常日志 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。

在本项目中的所有服务器端的接口全部都编写了规范的单元测试代码，这里列举一个测试用户修改密码的完整 unit 单元测试代码。

```

public class TestUserService extends TestBase {
    private UserService userService;
    @Before
    public void init() {
        userService = getContext().getBean("userService", UserService.class);
    }
    // 修改密码测试用例1
    // 期望结果: 1, 原始密码错误, null 测试结果: 1, 原始密码错误, null
    @Test
    public void test1() {
        StResult<User> result = userService.changePwd("31b3a1d544ad48d6931580c874bc062d", "admin", "123457",
            "admin123");
        System.out.println(result.getStatus() + ", " + result.getMsg() + ", " + result.getData());
    }
    // 修改密码测试用例2
    // 期望结果: 0, 密码修改成功, null 测试结果: 0, 密码修改成功, null
    @Test
    public void test2() {
        StResult<User> result = userService.changePwd("31b3a1d544ad48d6931580c874bc062d", "admin", "123456",
            "admin123");
        System.out.println(result.getStatus() + ", " + result.getMsg() + ", " + result.getData());
    }
}

```

图 6-1 单元测试代码

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同的结果。

OneAPM CPT 压测节点遍布全球,节点分部图如 6-2 所示

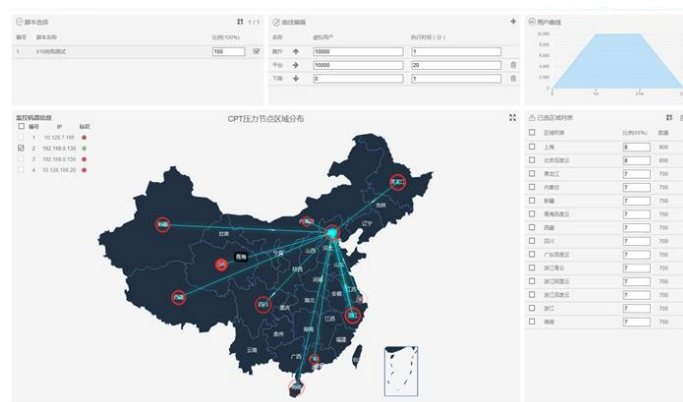


图 6-2 OneAPM CPT 压测节点

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。

7 总结和展望

本文遵守程序设计的原则和规范，运用了当前互联网建站的主流技术和资源，最终完成了基于 J2EE 的尾矿库安全云管理系统。

尾矿库安全云管理系统每一个尾矿企业相关企业都应该配备，每一个网站设计都要有自身的特色，本项目中也体现出来自己的特色，本项目在如下的两个方面有新颖的地方。

1. 采用数据可视化展示，管理人员不用面对大量的数据，只需要关注可视化变化和系统自动预警即可完成日常的管理工作。
2. 本系统与传统的管理网站不一样，传统的网站只拥有 web 系统，但是本项目中添加连接硬件设备的接口，可以从任何可以采集数据的硬件端导入相应数据。

当然，在上面两个之外，本项目还有许多创新之处，由于文章有限，这里就不一一举例说明了。

通过一个多月的编码实现功能，最终完成了这个管理项目，自己从需求分析、系统设计、代码实现到测试运行最到达上线部署，自己了解了一整套软件项目开发流程。但是每一件事情都肯定不是完美的，在这一个多月中也发现了很多不足的地方，比如：

1. 网站的某些页面交互不够好看，自己前端技术不够完善。
2. 由于时间的问题，里面很多安全性的问题没有十分完善，比如数据传输中的加密问题等。
3. 测试工作太少，只进行了单元测试和 OneAPM CPT 云压测，没有进行完整测试。

总结完本次项目，有收获也有不足的地方，对今后的工作有了明确的目标，就是完善自己提出的项目中还存在问题，完善自己的学习体系，争取将本项目能够完成的更好。

参考文献

- [1] 于广明,宋传旺,潘永战,李亮,李冉,路世豹.尾矿坝安全研究的国外新进展及我国的现状和发展态势[J].岩石力学与工程学报,2014,33(S1):3238-3248.
- [2] 杜通, 浑宝炬, 张大伟,等. 尾矿库的危害和安全管理措施[J]. 河北联合大学学报(自然科学版), 2009, 31(2):9-11.
- [3] 敬小非. 尾矿坝溃决泥沙流动特性及灾害防护研究[D]. 重庆大学, 2011.
- [4] 吴宗之,梅国栋.尾矿库事故统计分析及溃坝成因研究[J].中国安全科学学报,2014,24(09):70-76.
- [5] 荣艳冬.关于 Mybatis 持久层框架的应用研究[J].信息安全与技术,2015,6(12):86-88.
- [6] 黄艳秀.基于 mybatis 的面向数据库自动生成技术[J].河南科技,2014(04):21-22.
- [7] 黄啸. 基于 SSM 软件框架的 web 管理系统的研究与实现[D]. 暨南大学, 2016.
- [8] 陈佳. 交通路况预测模拟系统数据库设计[J]. 电脑知识与技术, 2016, 12(2):1-2.
- [9] 李志芳, 潘军, 孙辉. SSL 协议及 WEB 安全实现[J]. 煤炭技术, 2011, 30(5):116-117.
- [10] 蔚元方, 郑秋生, 李向东. 基于 D3 的地图信息可视化研究[J]. 中原工学院学报, 2015, 26(4):59-64.
- [11] Ingo Wölbern, Georg Rümpker. Crustal thickness beneath Central and East Java (Indonesia) inferred from P receiver functions[J]. Journal of Asian Earth Sciences, 2016, 115.
- [12] Y.S. Kong, S. Abdullah, M.Z. Omar, S.M. Haris. Failure assessment Of A Leaf Spring Eye Design under various load cases[J]. Engineering Failure Analysis, 2016, :.

致 谢

在完成本次论文的撰写过程中，我首先要感谢的就是我的第一指导人王博老师。作为一名本科生，第一次面对毕业论文，对一切都感觉陌生和疑惑，但是经过与老师的多次交流后，老师非常认真的分析我的毕业设计的总体框架、软件设计的总体需要，即基于云平台的尾矿库安全管理系统的思路，在王博老师细心指导下，我顺利的完成了开题报告。在编码的过程中，王博老师多次指导我完善自己的代码质量、修改 bug，最后才真正的实现了系统的全部需求。

编写代码的实际过程中，导出数据报表采用的 POI 技术自己以前没有接触过，在恶补 POI 基础之后，王博老师又给予我关键点的指导，经过老师的指点和帮助，自己才可以使用 POI 进行完整的报表功能的开发。

四年的大学时光匆匆走过，在此，我再次感谢包括此次论文指导老师王博在内的所有商洛学院的教师，感谢你们四年孜孜不倦的教诲。