分类号：TP315UDC：D10621-408-(2018)2146-0

密　级：公开编号：**2014051016**

成都信息工程大学

学位论文

基于Hadoop自适应光学分析系统的设计与实现

|  |  |
| --- | --- |
| 论文作者姓名： | 吴光宇 |
| 申请学位专业： | 计算机科学与技术 |
| 申请学位类别： | 工学学士 |
| 指导教师姓名（职称）： | 张欢（讲师） |
| 论文提交日期： | 2018年5月23日 |

基于Hadoop自适应光学分析系统的设计与实现

**摘　要**

建立数据结构课程管理系统，实现数据结构课程的数字化管理，搭建师生之间的在线交流平台，对于提高学校教学水平，改善教学环境具有重大意义。以Oracle为数据库，采用Struts2+Ibatis架构建立基于B/S模式的数据结构课程管理系统，实现了教师对数据结构这门课程的日常教务管理，提供了师生之间的在线交流平台。系统的主要功能包括：教师可以在线维护课程的教学大纲，发布课程公告，维护个人信息，管理教师课件，在线布置学生作业，记录学生出勤，维护管理试卷库，编辑学生上传的学习资源，发布考试安排信息，录入考试成绩等日常教务管理；学生可以在线查询课程相关信息，下载教师课件，查看作业内容，上传作业，查询考试成绩，共享学习资源，在线交流学习。系统还为师生之间的学习交流提供了一个在线交流平台—BBS，方便了教师的答疑解惑，学生的学习交流，为传统的面授交流提供了一种很好的辅助。

**关键词：**Struts2；Ibatis；数据结构；课程管理；Oracle；BBS

Thesis Title

**Abstract**

Developing a data structure course management system, implementing data structure course on digital management, and setting up an online communication platform between students and teachers is of great significance for improving teaching and improving teaching and learning environment. With the database Oracle, the system is developed under the integration of Struts2 and Ibatis framework and based on B/S mode. It provides data structure of daily educational administration to teachers, and an online communication platform between students and teachers. System's main features include: teachers can maintain the course syllabus , post course announcements online, protect personal information, manage teachers courseware, assign homework online, record students attendance, maintain and manage the paper library, edit resources uploaded by students, publish information of Exam schedule, record students achievement and do other daily educational administration; students can check course information online, download teachers courseware, view homework, upload assignments, check examination results, share learning resources, take exchange of learning online. In order to facilitate teachers’ FAQ, and students’ learning, it also provides an online learning platform-BBS between teachers and students. So it is a supplementary system for the traditional face to face communication.

**Key words:** Struts2; Ibatis; Data Structure; Course Management; Oracle; BBS

**目 录**

论文总页数：插入页数页

[1 引言 - 1 -](#_Toc514960762)

[1.1 课题背景 - 1 -](#_Toc514960763)

[1.2 国内外研究现状 - 1 -](#_Toc514960764)

[1.3 本课题研究的意义 - 1 -](#_Toc514960765)

[1.4 本课题的研究方法 - 2 -](#_Toc514960766)

[2 需求分析及项目方案 - 2 -](#_Toc514960767)

[2.1 需求分析 - 2 -](#_Toc514960768)

[2.1.1 功能需求 - 2 -](#_Toc514960769)

[2.1.2 系统约束条件 - 4 -](#_Toc514960770)

[2.2 项目方案 - 4 -](#_Toc514960771)

[2.2.1 项目特色 - 4 -](#_Toc514960772)

[2.2.2 Spring技术简介 - 5 -](#_Toc514960773)

[2.2.3 Spring Security技术简介 - 7 -](#_Toc514960774)

[2.2.4 Hadoop技术简介 - 8 -](#_Toc514960775)

[2.2.5 Hbase技术简介 - 8 -](#_Toc514960776)

[2.2.6 开发工具和运行环境 - 8 -](#_Toc514960777)

[2.2.7 项目实施计划 - 9 -](#_Toc514960778)

[3 项目设计及实现 - 10 -](#_Toc514960779)

[3.1 数据库设计 - 10 -](#_Toc514960780)

[3.1.1 表结构设计 - 10 -](#_Toc514960781)

[3.1.2 关于数据库设计的一些说明 - 12 -](#_Toc514960782)

[3.2 功能模块的设计与实现 - 13 -](#_Toc514960783)

[3.2.1 系统整体框图 - 13 -](#_Toc514960784)

[3.2.2 数据管理模块的设计与实现 - 13 -](#_Toc514960785)

[3.2.3 系统管理模块的设计与实现 - 17 -](#_Toc514960786)

[4 特殊问题及解决方案 - 19 -](#_Toc514960787)

[4.1 Hbase行键设计问题 - 19 -](#_Toc514960788)

[4.1.1 问题描述 - 19 -](#_Toc514960789)

[4.1.2 解决方案 - 20 -](#_Toc514960790)

[4.1.3 结果 - 20 -](#_Toc514960791)

[4.2 中文乱码问题 - 20 -](#_Toc514960792)

[4.2.1 问题描述 - 20 -](#_Toc514960793)

[4.2.2 解决方案 - 20 -](#_Toc514960794)

[4.2.3 结果 - 20 -](#_Toc514960795)

[4.3 前端图表重新渲染问题 - 20 -](#_Toc514960796)

[4.3.1 问题描述 - 20 -](#_Toc514960797)

[4.3.2 解决方案 - 20 -](#_Toc514960798)

[4.3.3 结果 - 20 -](#_Toc514960799)

[5 结果测试及性能分析 - 21 -](#_Toc514960800)

[5.1 测试概要 - 21 -](#_Toc514960801)

[5.1.1 测试环境 - 21 -](#_Toc514960802)

[5.1.2 测试方法 - 22 -](#_Toc514960803)

[5.1.3 测试范围 - 22 -](#_Toc514960804)

[5.1.4 数据上传模块测试 - 22 -](#_Toc514960805)

[5.1.5 数据可视化模块测试 - 22 -](#_Toc514960806)

[5.2 性能分析 - 23 -](#_Toc514960807)

[结 束 语 - 24 -](#_Toc514960808)

[参考文献 - 25 -](#_Toc514960809)

[致　　谢 - 26 -](#_Toc514960810)

[声　　明 - 27 -](#_Toc514960811)

# 引言

## 课题背景

自适应光学是在1953年由Horace W.Babcock提出，主要构想是用闭环矫正波前误差来补偿天文视宁度[1]。造成波前误差的主要原因是大气湍流，而波前误差会极大的影响最后的成像质量，被动光学以及早期的主动光学无法很好的解决这个问题，而自适应光学的出现为解决这个问题提供了可能。

自适应光学系统主要由波前传感器，波前控制器以及波前矫正器组成。其中波前传感器主要负责捕捉光因大气湍流和其他原因造成的波前误差，波前控制器通过分析、计算波前传感器传输的数据，并将其转化为波前矫正器的电压参数，而波前矫正器负责控制控制矫正单元，使望远镜镜面发生形变，以矫正相位，从而获得成像质量较好的图形。

在自适应光学系统短时间运行后，将累计大量的数据，这些数据主要由波前传感器捕获的斜率数据和波前控制器计算获得电压数据组成，而如何存储、解析和可视化这些实验数据是一个值得考虑的问题。

## 国内外研究现状

当前存在的软件通常使用关系型数据库作为数据存储的媒介，这在软件早期或者实验数据累计量较小的时候是一个不错的解决方案，但是随着实验数据的增多，数据库的扩展是一个必然会遇到的问题。通常来说数据库的扩展方式有两种，一种是纵向扩展，通过提高单机的性能来提高数据库的存储能力上限，但是这种方式成本十分高昂。第二种方式，也是目前采用较为多的方式，横向扩展数据库，这种方式只需要通过添加节点即可完成，但是这种方式对关系型数据库并不友好，主要的问题在于横向扩展后如何维护表之间关系，目前没有很好的方式解决这个问题。

所以本文尝试使用非关系型数据库来存储实验数据，并提供配套的权限体系，数据可视化等功能。

## 本课题研究的意义

基于Hadoop自适应光学分析系统是一个人，计算机，数据组成的有机整体，它主要解决了以下一些问题。

数据的存储问题。针对前文提出的现有软件的关于数据存储存在的一些问题，通过使用非关系型数据库，可以有效解决上述问题。随着数据量的增加，只需要通过横向扩展的方式增加数据节点即可，而如果出现数据量减少的情况，则只需要适当减少数据节点即可。而物理环境的变动几乎不会对软件的实际性能造成显著影响。

数据可视化问题。通过多元化的方式将数据渲染成不同的图像，诸如表格，折线图，散点图等。可以让数据表现的更为直观，方便使用人员对数据进行对比，分析。

## 本课题的研究方法

本系统是一个以Hbase为数据库，搭建在Spring mvc框架下的基于B/S结构的数据分析系统。服务器版本是Tomcat7，集群版本是HDP-2.6.0.3，服务器操作系统版本为Centos7，在Intellij Idea环境下进行开发，编程语言是java，js。

# 需求分析及项目方案

## 需求分析

### 功能需求

本系统的用户有普通用户、数据管理员、权限管理员和系统管理员。通过本系统，普通用户可以查看数据，提交评论，上传实验数据，数据管理员可以删除数据，权限管理员可以管理权限，分配角色，系统管理员可以维护系统的所有功能，保证系统稳定运行。总共分为4个模块：数据管理、角色管理、用户管理、日志查看。

1) 数据管理

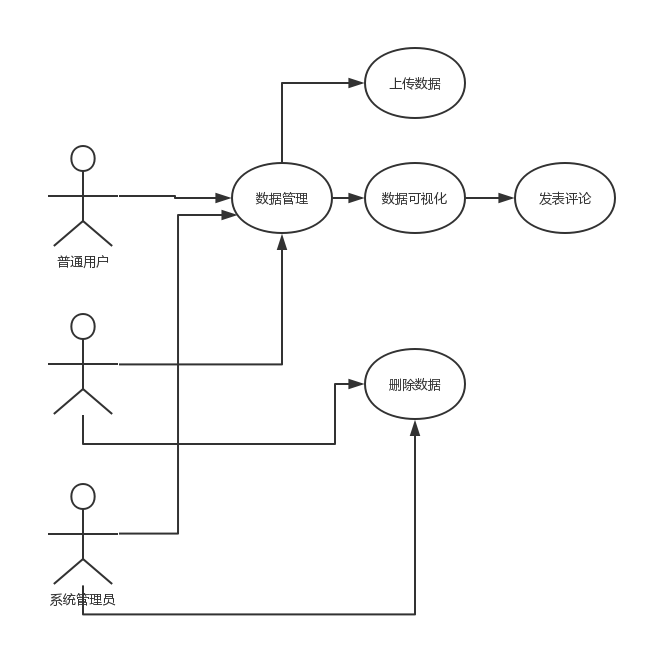
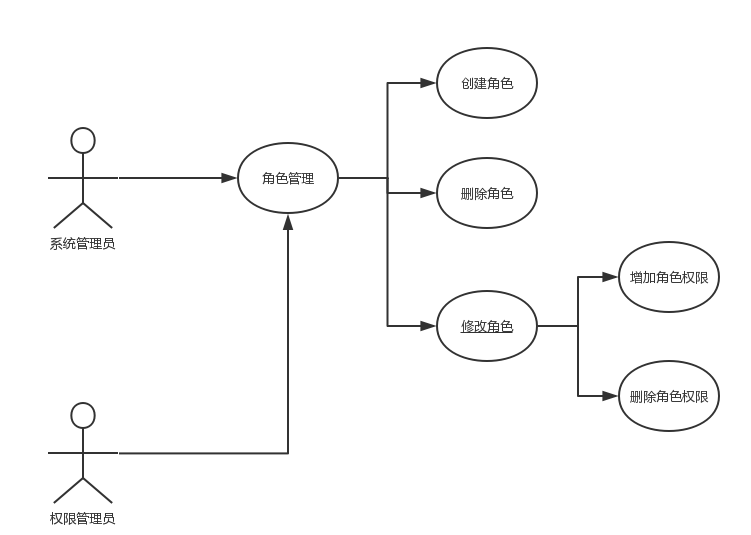
在此模块中，用户可以上传自己的实验数据，查看系统已存在数据，浏览某一数据的详细信息，针对某一信息进行评论。本模块的用例图如图2‑1所示。

图2‑1 数据管理的用例图

2) 角色管理

角色是一组权限的集合，代表着一种于系统进行交互的身份。通过这个功能可以创建不同的角色，增加、删除或者修改角色，通过这些操作，可以实现对系统访问权限的细粒度控制。本模块的用例图如图2-2所示。

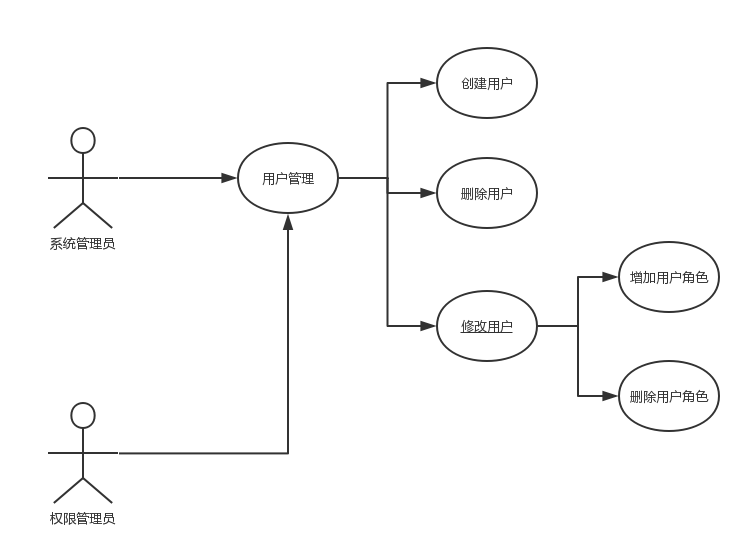
图2‑2 角色管理的用例图

图2‑3 用户管理的用例图

3) 用户管理

用户是实际与系统进行交互的人，通过授予用户不同的角色来控制用户对系统访问的权限等级，可以有效的维护系统和系统数据的安全。在该模块，系统管理员可以增加用户，删除用户，以及修改用户信息。本模块的用例图如图2-3所示。

3) 日志管理

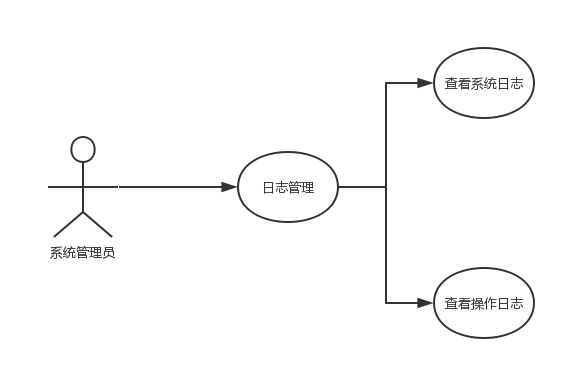
日志是系统运行过程中信息的记录，日志分为两种，一种是系统日志，系统管理员可以通过查看系统日志来判断系统的运行状态，从而防止系统出现问题。第二种是操作日志，系统管理员可以通过操作日志来跟踪用户的操作记录，当用户出现违规操作时可以通过操作日志来回复系统的数据。本模块的用例如如图2-4所示。

图2‑4 日志管理的用例图

### 系统约束条件

本系统的用户需要具备基础的计算机操作技能

访问本系统需要一台可以访问网络的PC，浏览器。

## 项目方案

### 项目特色

1. 多平台适用，本系统使用Java语言进行开发，得益于Java语言的平台无关性，本系统可以在Windows平台上良好运行，也可以在Linux的平台上良好运行。

2) 数据库的横向扩展十分容易。得益于Hbase本身的设计，数据库的扩展或者缩减只需要对数据节点进行增加或者减少即可。

3) 项目后端采用了合理的架构，考虑了系统的可维护性和可扩展性，位系统的升级维护提供了便利。

4) 使用了前后端分离的技术。前后端通信统一为Json字符串，这样可以方便前后端分离部署，以便前后端采用不同的负载均衡技术。

5) 使用了Echarts进行前端渲染，通过数据可视化技术，使人和数据交互变得更为方便自然。

### Spring技术简介

支撑spring的两大核心原理，一是控制反转(依赖注入)，在传统的Java程序中，系统模块之间必然存在这各种各样的依赖关系，过强的耦合会使软件结构复杂、维护困难[2]。而Spring的控制反转(依赖注入)则可以解决这个问题。另一个是面向切面编程，面向切面编程主要实现的目的是针对业务处理过程中的切面进行提取，它所面对的是处理过程中的某个步骤或阶段，以获得逻辑过程中各部分之间低耦合性的隔离效果[3]。这些切面，也叫关注点，一个关注点，就是一个特定的目的，或者说一个系统需要实现的目标[4]，而这个关注点可能横跨几个模块。

在Spring中实现控制反转(依赖注入)的方式是Jvm级别维护一个Ioc容器，该Ioc容器通过配置文件来生成、管理、销毁类实例，而在程序中需要获取类实例都在Ioc容器中进行获取，即组件之间的依赖关系由容器在运行期间决定，形象的说，就是容器在运行期间将对象实例注入到组件中，而组件只需要关心自己的业务逻辑，而不需要关心使用的资源来自何方。

而面向切面编程主要解决代码重复和关注点分散的问题，面向切面编程比较常见的例子就是系统的日志接口，几乎每个类都需要使用日志接口，如果将日志功能单独抽象为一个模块，然后在每个类中都对其进行实例化，这样会造成所有模块都和日志模块耦合，而如果将日志功能的代码嵌入到每个需要使用的地方则会造成该段代码重复率极高，而对于原本的类来说，日志功能的实现却不是该模块应该关注的内容，为了解决这些问题，Spring提供了面向切面编程的功能，通过声明的方式将功能模块应用到需要他们的模块中去，进而改变他们的行为。

Spring是基于配置来实例化对象、应用对象、销毁对象，随着项目规模的不断扩大，Spring需要的配置文件越来越繁杂，以至于维护配置信息变得极其困难，Spring Boot的出现就是为了解决这个问题，Spring Boot默认习惯优于配置，即Spring默认配置好了符合大多数项目的配置文件，而使用者无需关心，直接使用即可，而调整默认配置也变得极其简单。

Spring MVC是一个MVC框架，通过实现Model-View-Controller模式来很好地将数据、业务与展现进行分离[5]。Spring MVC的核心思想继承了Struts、Struts2，即采用中心Dispatch来管理所用前端请求，同时支持返回多种格式文件，比如视图，或者字符串等。同时与Spring很好的结合，将这个核心思想通过Spring的方式实现，即通过依赖注入，注解等手段来实现整个框架的部署和使用。

1. Spring 框架的引入，在pom.xml中添加如下

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring-webmvc.version}</version>

</dependency>

1. Spring配置如下：

通过Java的方式进行配置，先初始化web相关配置：

public class WebAppInitializer extends AbstractAnnotationConfigDispatcherServletInitializer {

protected Class<?>[] getRootConfigClasses() {

return new Class[] {RootConfig.class, SecurityConfig.class};

}

protected Class<?>[] getServletConfigClasses() {

return new Class[] {WebConfig.class};

}

protected String[] getServletMappings() {

return new String[] {"/"};

}

@Bean

public MultipartResolver multipartResolver() throws IOException {

return new StandardServletMultipartResolver();

}

@Override

protected Filter[] getServletFilters() {

// 默认不是utf-8，还得自己转

return new Filter[] {new CharacterEncodingFilter("UTF-8", true, true)};

}

// 文件上传相关配置

@Override

protected void customizeRegistration(ServletRegistration.Dynamic registration) {

registration.setMultipartConfig(

// 默认上传到程序的发布目录

new MultipartConfigElement(""));

}

}

1. 不与前端交互的bean的配置：

@Configuration

@ComponentScan(basePackages = {"com.graduation.log", "com.graduation.logic"})

public class RootConfig {}

1. controller配置：

@Configuration

@EnableWebMvc

@ComponentScan("com.graduation.module")

public class WebConfig extends WebMvcConfigurerAdapter {

// 现在把html当静态资源处理，如果是在是要做html的解析，还要在这里配置html的视图解析器

// 如果需要对静态资源进行访问，可以在下面进行配置

@Override

public void configureDefaultServletHandling(DefaultServletHandlerConfigurer configurer) {

configurer.enable();

}

}

### Spring Security技术简介

Spring security是为基于spring的应用程序提供声明式安全保护的安全性框架[6]。

认证和权限控制是Spring Security要处理的两个主要领域[7]。认证需要处理这样一些问题：

1.用户具有哪些权限

2.访问对应资源需要什么权限

3.用户是否具有访问该资源的权限

授权需要解决这些问题：

1.该系统权限种类

2.如何授予用户权限

Spring security支持广泛的认证模型，其中大部分既支持标准机构发布的模型，也支持第三方发布的模型。

Spring security的权限认证主要集中在以下三个方面：

1.WEB请求的认证

2.方法执行的认证

3.独立域资源的授权

在Web程序中通常使用WEB请求认证，即对所有可能的请求规定相应的权限，只有拥有该权限的用户发送的请求才是合法请求，其他请求都视为不合法的请求。

为了实现上述过程，需要实现这样一些类。首先需要实现AccessDecisionManager这样一个类，在该类中决定了用户将以何种方式通过权限认证，在该类中可以获取到用户具有的权限以及访问请求需要的权限，之后需要明确以何种方式来认证请求合法，通常来说，是采用或的方式，即用户拥有该请求要求的任一权限即认证通过，本系统采用的就是这种方式，当然，也可以时且的方式，即必须用有访问该请求的所有权限才算认证成功。当然除了这两种通用的方式，还可以按照实际需求，组合不同的方式来完成权限认证。

还需要实现MySecurityContext，这个类主要是指明如何获取访问不同请求需要的权限，通常来说，这些信息可以放在数据库中，对于数据量不太多的情况，也可以放在xml或者property之类的文件中，本系统采用的方式是将之放在property文件中。

之后需要实现的类是MySecurityMetadataSource，该类指明了访问一个特定请求后，如何获取它需要的权限。该类是建立在实现了MySecurityContext的基础上的。

UserDetails是一个用户实体，Java Bean类，是认证体系中用于存放用户信息的类。

UserDetailsService指明了如何根据用户名从数据库中加载该用户具有的权限的。在本系统中，需要先通过用户名加载用户的角色信息，然后通过这些角色信息加载权限，然后组装成相关的集合。

### Hadoop技术简介

Apache Hadoop软件库是一个框架，允许使用简单的编程模型跨分布式计算机集群处理大型数据集[8]。

Hadoop是Google分布式系统的开源版本，Hadoop有三个核心：

HDFS文件系统，该文件系统直接参照了Google的GFS文件系统，在设计上有诸多相似之处。HDFS运行在廉价的商业软件上，因此HDFS文件系统是高度容错的[9]。HDFS通过存放多份数据来避免因为硬件错误造成的数据丢失，通过相应的算法来保证数据的一致性。

第二个核心是Mapreduce计算框架，Hadoop的开发人员在HDFS文件系统的基础上，实现了Google的Mapreduce计算框架，使Hadoop真正成为了一个可以实际运用的分布式计算框架，Mapreduce的本质在于并行运算，Mapreduce将耗时的计算任务进行划分，并将该任务传送到计算数据所在的节点，然后多节点并行运算，然而这种并行几乎不需要付出代价，除了需要满足该编程模型的一些要求之外，这使得没有任何并行和分布式系统经验的程序员可以轻松利用大型分布式系统的资源[10]。

最后一个核心是Hadoop的资源调度框架Yarn，可以保证Hadoop集群的负载均衡。

### Hbase技术简介

Hbase是一个开源的非关系型分布式数据库（NoSQL），它参考了google的bigTable[11]建模，实现的语言为java[12]。

Hbase适合于存储大数据，支持单条记录的快速查询，在任意指定位置单条或批量添加，删除数据[13]。Hbase将数据作为字节块存储在分布式文件系统中，如果列值为空，将不会被存储，这将大大节省存储空间。而行键按LSM结构进行存储，进而支持单条记录的快速查询。

分布式的非SQL数据库提供了人们十分需要的可伸缩特性[14]。这使得横向扩展变得容易，从而能够存储更多的数据。

### 开发工具和运行环境

1. 系统开发工具和开发环境如表2‑1所示。

表2‑1 开发工具表

|  |  |
| --- | --- |
| 开发工具名称 | 软件用途 |
| JDK 1.8 | 系统环境 |
| Intellij Idea 2018.1.1 | 开发平台 |
| Tomcat7.0 | 服务器 |
| Microsoft Visio 2003 | 系统设计，创建流程图等 |
| HDP-2.6.0.3 | 集群版本 |
| Centos7 | 集群服务器操作系统版本 |

### 项目实施计划

# 项目设计及实现

## 数据库设计

### 表结构设计

1. 用户表

表3‑1 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名称** | | 用户表 | | |
| **物理表名** | | graduation\_user | | |
| **行键** | | username | | |
| **业务主键** | | 无 | | |
| **字段列表** | | | | |
| **序号** | **中文名称** | | **列族名** | **列名** |
| 1 | 密码 | | userinfo | password |
| 2 | 电话 | | userinfo | phone |
| 3 | 描述 | | userinfo | description |

1. 角色表

表3‑2 角色表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名称** | | 角色表 | | |
| **物理表名** | | graduation\_role | | |
| **行键** | | rolename | | |
| **业务主键** | | 无 | | |
| **字段列表** | | | | |
| **序号** | **中文名称** | | **列族名** | **列名** |
| 1 | 描述 | | roleinfo | description |

1. 权限表

表3‑3 权限表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名称** | | 权限表 | | |
| **物理表名** | | graduation\_ jurisdiction | | |
| **行键** | | permission | | |
| **业务主键** | | 无 | | |
| **字段列表** | | | | |
| **序号** | **中文名称** | | **列族名** | **列名** |
| 1 | 描述 | | jurisdictioninfo | description |

1. 用户角色表

表3‑4 用户角色表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名称** | | 用户角色表 | | |
| **物理表名** | | graduation\_user\_role | | |
| **行键** | | username-rolename | | |
| **业务主键** | | 无 | | |
| **字段列表** | | | | |
| **序号** | **中文名称** | | **列族名** | **列名** |
| 1 | 描述 | | info | description |

1. 权限角色表

表3‑5 权限角色表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名称** | | 角色权限表 | | |
| **物理表名** | | graduation\_role\_jurisdiction | | |
| **行键** | | role-jurisdiction | | |
| **业务主键** | | 无 | | |
| **字段列表** | | | | |
| **序号** | **中文名称** | | **列族名** | **列名** |
| 1 | 描述 | | info | description |

1. 通用数据表

表3‑6 通用数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名称** | | 通用数据表 | | |
| **物理表名** | | graduation\_data | | |
| **行键** | | filename | | |
| **业务主键** | | 无 | | |
| **字段列表** | | | | |
| **序号** | **中文名称** | | **列族名** | **列名** |
| 1 | 上传者名字 | | datainfo | username |
| 2 | 上传时间 | | datainfo | time |
| 3 | x偏移量统计信息 | | datainfo | jsonStatisticsX |
| 4 | x偏移量统计信息 | | datainfo | jsonStatisticsY |
| 5 | 描述 | | datainfo | description |

1. 详细数据表

表7‑7 详细数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名称** | | 详细数据表 | | |
| **物理表名** | | graduation\_data\_all | | |
| **行键** | | filename | | |
| **业务主键** | | 无 | | |
| **字段列表** | | | | |
| **序号** | **中文名称** | | **列族名** | **列名** |
| 1 | 上传者名字 | | datainfo | username |
| 2 | 上传时间 | | datainfo | time |
| 3 | x偏移量统计信息 | | datainfo | jsonStatisticsX |
| 4 | x偏移量统计信息 | | datainfo | jsonStatisticsY |
| 5 | x实验数据 | | datainfo | jsonX |
| 6 | y实验数据 | | datainfo | jsonY |
| 7 | 序列 | | datainfo | sequence |

### 关于数据库设计的一些说明

数据库遵循高表设计的原则，每张表的列族数量不多于3个，在关联表，诸如角色权限表，或者用户角色表中，行键设计遵循等长，避免热点问题的原则，详细情况将在后文说明。

## 功能模块的设计与实现

### 系统整体框图

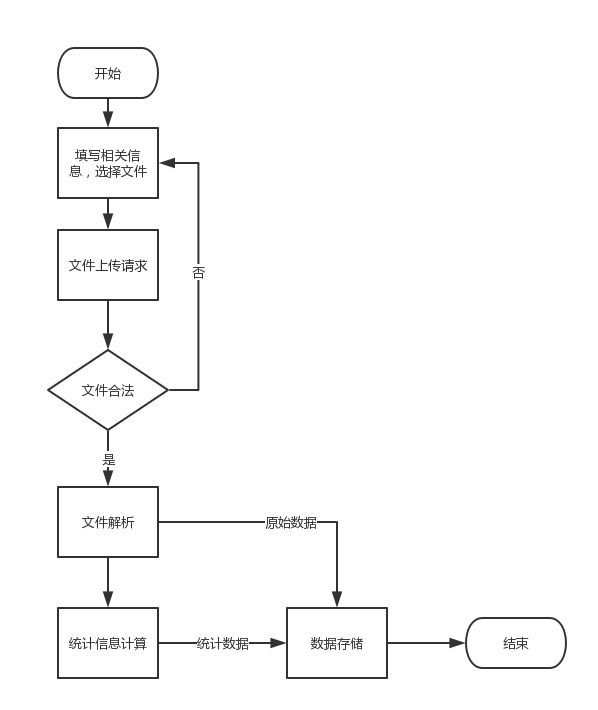
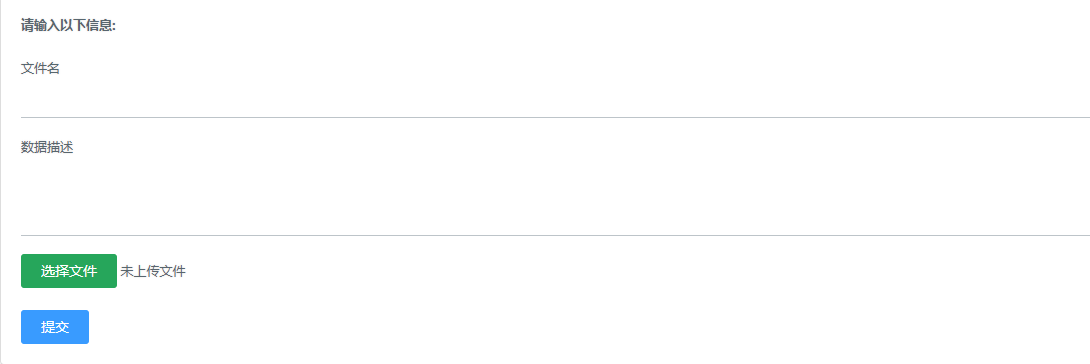
### 数据管理模块的设计与实现

本模块主要实现对用户上传数据的处理和展示。主要包括：数据导入和数据可视化两个子模块，各个模块的设计与实现如下。

1. 数据导入
   1. 功能描述

用户进入系统主页面，点击文件上出按钮，弹出文件上传界面，用户填写相关信息，选择上传文件，发送上传请求，后端接受上传的文件到指定位置，检查文件是否合法，如果不合法发送相关状态码给前端，前端提示用户并要求用户重新提交文件，如果文件合法，则返回成功状态码，后端开始解析文件，计算该实验数据的相关统计信息，最后将原始数据和相关统计信息存储到数据库中。

具体流程图如下：

* 1. 上传功能截图如下：
  2. 主要代码

public String upload(……)

throws IOException {

JSONObject res = new JSONObject();

if (!multipartFile.isEmpty()) {

String suffix = ".ocamdq\_gv";

FileUtils.copyInputStreamToFile(

multipartFile.getInputStream(), new File("target/upload", filename + suffix));

HttpSession session = request.getSession();

String url = "target/upload/" + filename + suffix;

// 这个实在太慢了，所以另起一个线程进行操作

Thread thread =

new Thread(

() -> {

try {

dataManager.saveDataToDB(

dataManager.formatGData(url),

(String) session.getAttribute("username"),

description,

filename + suffix);

} catch (IOException e) {

e.printStackTrace();

}

});

thread.start();

……

} else {

res.put("msg", "上传失败");

res.put("status", "fail");

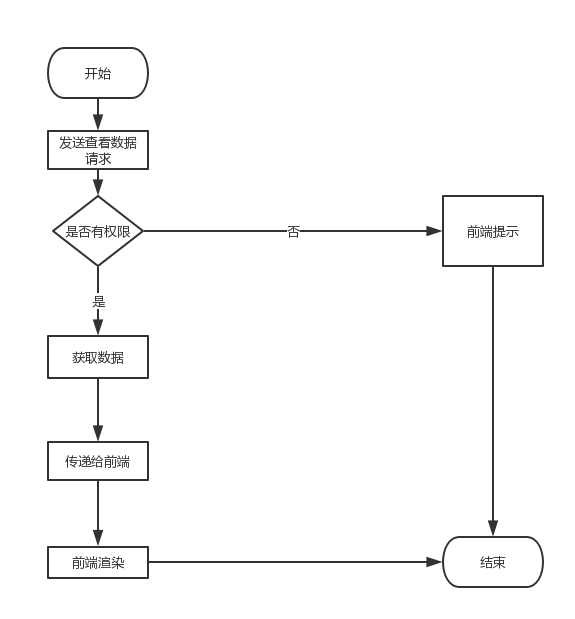
}

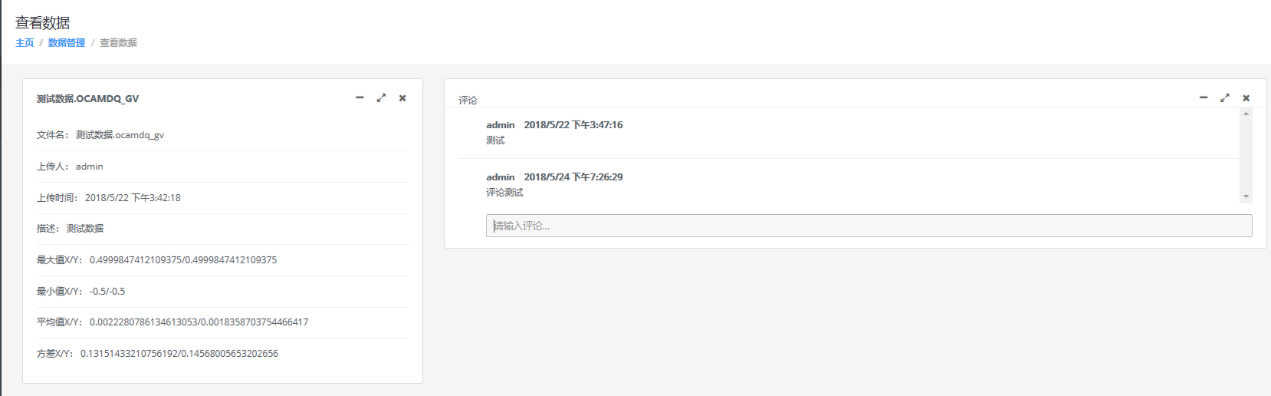
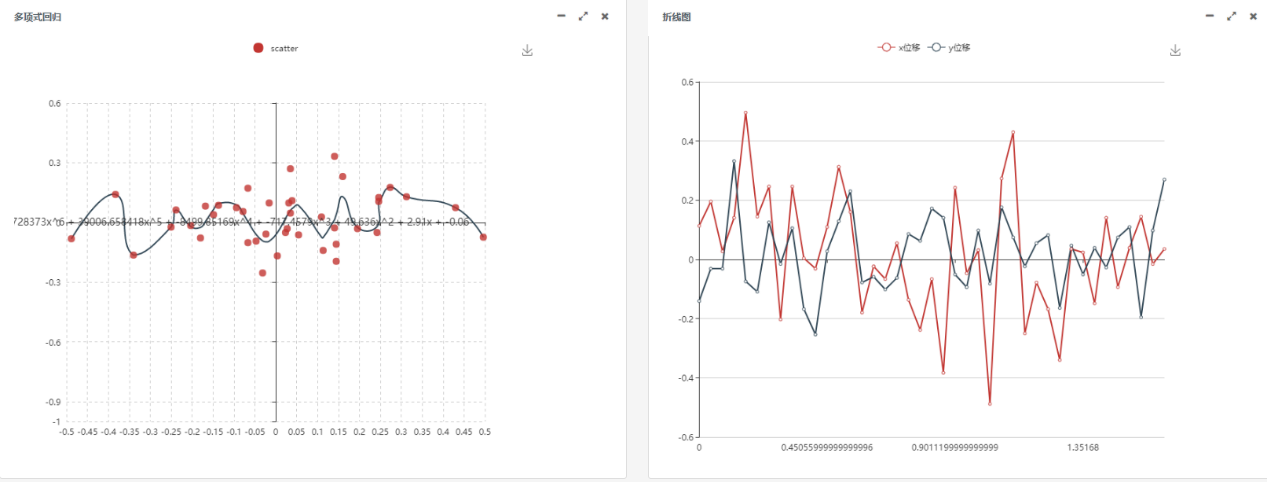
return res.toString();

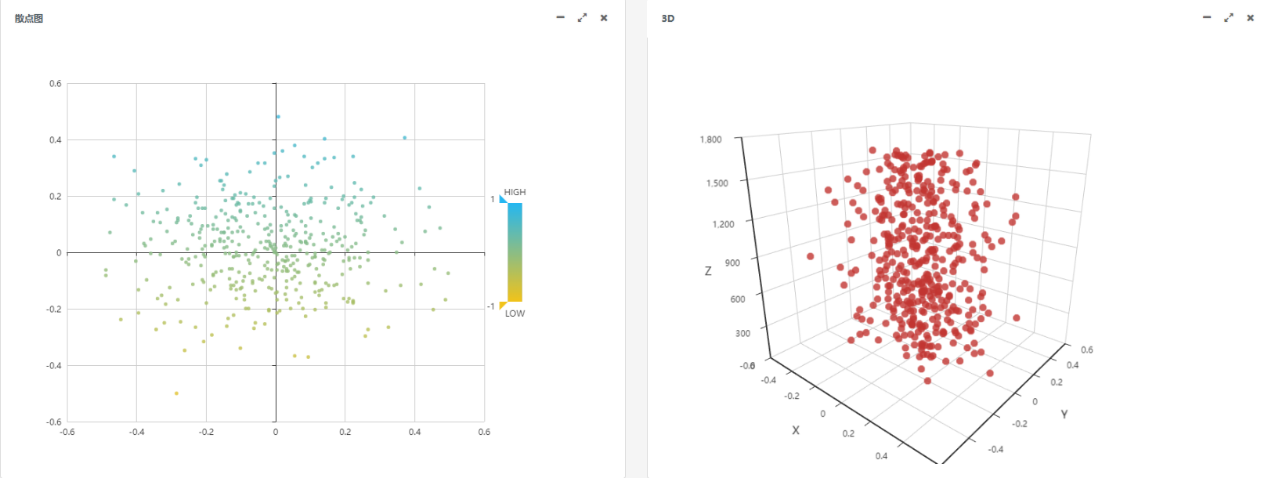
}

1. 数据可视化
   1. 功能描述

用户选择相应的数据集，发送查看该数据集的请求，检查用户是否具相关权限，如果用户具有相关权限，后端查询数据库，将相关信息发送给前端，前端渲染该数据的可视化界面。用户可以在该界面查看数据的基础统计信息，查看相关图表，发表对于该数据集的评论。

具体流程如下：

* 1. 功能截图如下：



* 1. 主要代码如下：

@RequestMapping(value = "/getDataByFileName", produces = "application/json;charset=UTF-8")

@ResponseBody

public String getDataByFileName(HttpServletRequest request) {

String fileName = (String) request.getSession().getAttribute("fileName");

DataBean dataBean = dataManager.getSimpleData(fileName);

if (dataBean == null) {

return "{\"status\":\"fail\"}";

}

return JSON.toJSONString(dataBean);

}

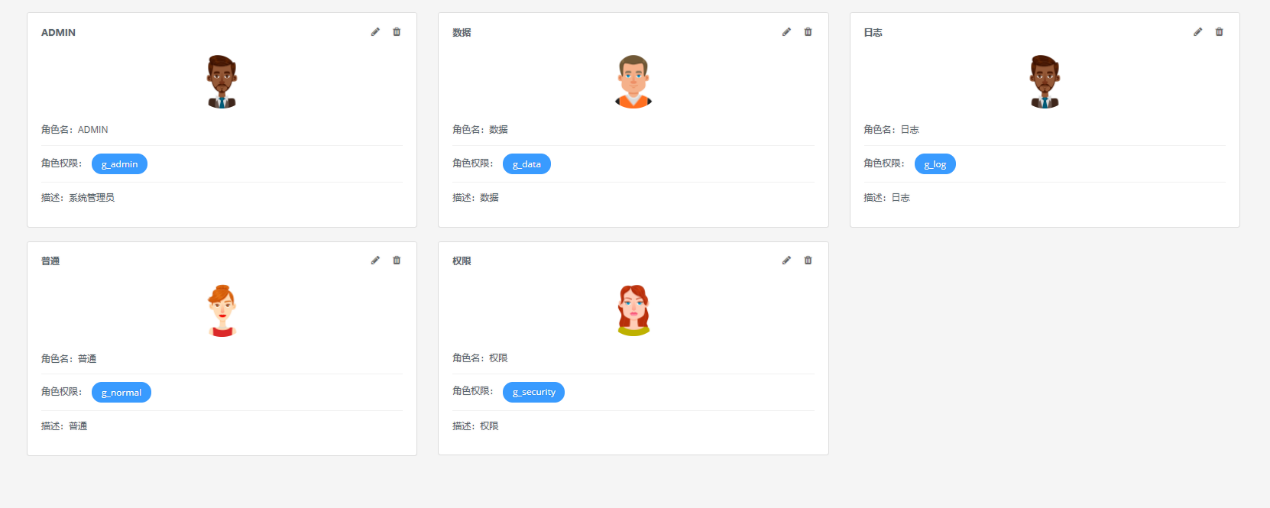
### 系统管理模块的设计与实现

本模块主要实现系统的权限管理，主要包括两个模块：角色管理和用户管理，各模块设计与实现如下。

1. 角色管理
   1. 功能描述

系统管理员在系统管理页面点击角色管理打开角色管理页面，通过角色管理页面系统管理员可以增加角色，删除角色，修改角色权限。

* 1. 角色管理界面截图如下：



* 1. 主要代码

function getRoleData() {

var roleDataTmpl = $('#role-data').html()

$.template("roleData", roleDataTmpl)

$.ajax({

type: "post",

url: "/dataAnalysis/role/getAllRoleData",

async: false,

success: function (data) {

$.tmpl("roleData", data).appendTo("#role-row")

},

error: function (data) {

alert(data["status"])

}

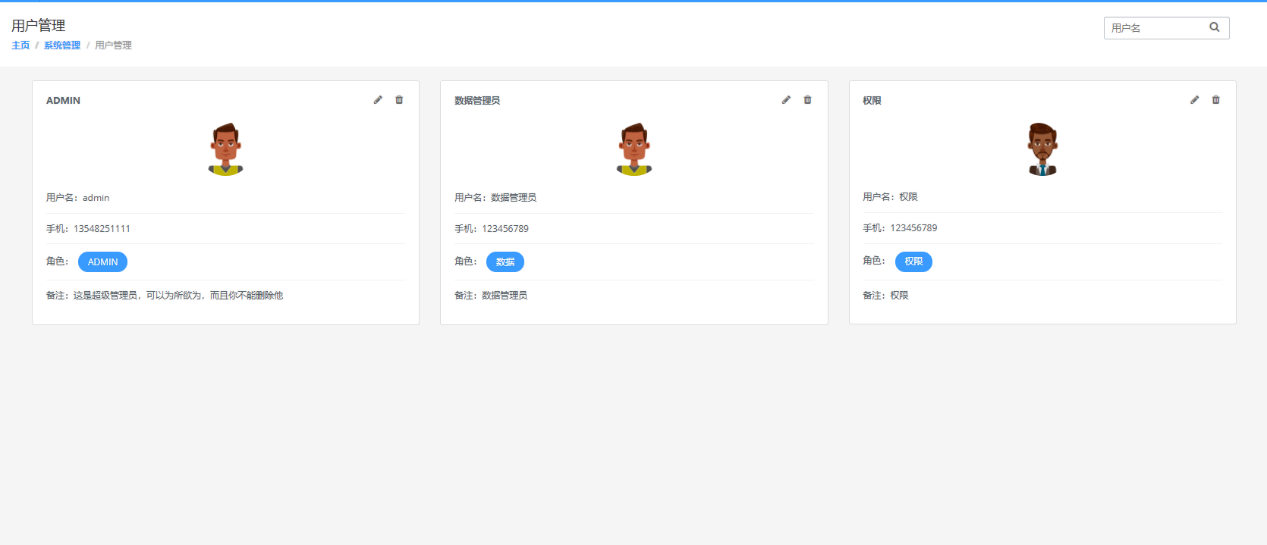
}

)

}

1. 用户管理
   1. 功能描述

系统管理员在系统管理页面点击用户管理打开用户管理页面，通过用户管理页面系统管理员可以增加用户，删除用户，修改用户信息。

* 1. 用户管理界面截图如下：
  2. 主要代码

function getUserData() {

var userDataTmpl = $('#user-date').html()

$.template("userData", userDataTmpl)

$.ajax({

type: "post",

url: "/dataAnalysis/user/getUserData",

async: false,

success: function (data) {

$.tmpl("userData", data).appendTo("#row")

},

error: function (data) {

alert(data["status"])

}

}

)

}

# 特殊问题及解决方案

## Hbase行键设计问题

### 问题描述

Hbase的行键设计对整个系统的数据库访问效率至关重要，在设计行键的过程中应该遵循一些原则。

行键内容在可接受的范围内应该包含更多的内容。

行键长度的问题，在本系统中，以用户角色表为例子，在行键中需要包含用户和角色两种信息，但是两种字段的长度是没有办法控制的，这可能会出现过长的字段与两种字段匹配的情况。

另外一个问题就是行键的热点问题，如果在字符上相关联的行键在同一时间会大量出现的话，将造成热点问题，即单一时间内某台物理机负载远高于其他物理机。

### 解决方案

针对行键长度的问题，将单一字段扩展为该字段允许的长度上限，不足以指定字符填充，这样整个行键和行键中的每一个字段都是等长且可控的，这样有利于后面的按条件查询。

针对行键的热点问题，这种问题常常出现在行键中包含时间戳的情况，对于程序中包含时间戳的情况，通常采取的做法是将时间错放在最后，或者将字符无关联的信息放在行键的开始部分。

### 结果

不仅解决了问题，还有效提高了数据库行键的查询效率和容错性。

## 中文乱码问题

### 问题描述

在本程序中前后端固定采用Json格式的字符串作文消息交换的格式，在程序运行过程中发现如果消息中含有中文，在前端会出现乱码。

### 解决方案

在服务端返回Json字符串时指定字符串的编码格式为“UTF-8”。

### 结果

前端正确显示中文。

## 前端图表重新渲染问题

### 问题描述

当图表绑定的数据变更的时候，需要重新渲染图表，但是实际结果显示渲染失败，通过查询知道，在使用echarts进行图表渲染的时候，会给容器设置一个属性，在进行图表渲染的时候，需要通过这个属性进行定位，当对当前容器再次进行渲染时，会发现该容器已经具有这个属性了，而没有办法用新的属性覆盖就得属性，导致再次渲染得时候没有办法定位到需要渲染得容器。

### 解决方案

在渲染前将容器得旧属性移除，保证新的属性可以增加上去。

### 结果

可以正确渲染新的图表。

# 结果测试及性能分析

## 测试概要

### 测试环境

表5‑1 数据库服务器配置

|  |  |
| --- | --- |
| CPU | E3-1230 V3 |
| 内存 | 8G |
| 硬盘（可用空间大小） | 1T |
| 操作系统 | Centos7 |
| 机器网络名 | master,slave1,slave2 |
| 局域网地址 | 192.168.0.100、192.168.0.101、192.168.0.102 |

表5‑2 应用服务器配置

|  |  |
| --- | --- |
| CPU | E3-1230 V3 |
| 内存 | 8G |
| 硬盘（可用空间大小） | 1T |
| 操作系统 | Centos7 |
| 应用软件 | Tomcat7 |
| 机器网络名 | master |
| 局域网地址 | 192.168.0.100 |

表5‑3 客户端器配置

|  |  |
| --- | --- |
| CPU | i5-8250U |
| 内存 | 8G |
| 硬盘（可用空间大小） | 250G |
| 操作系统 | Windows 10 |
| 应用软件 | Chorme |
| 机器网络名 | MC |
| 局域网地址 | 192.168.0.103 |

### 测试方法

黑盒测试

### 测试范围

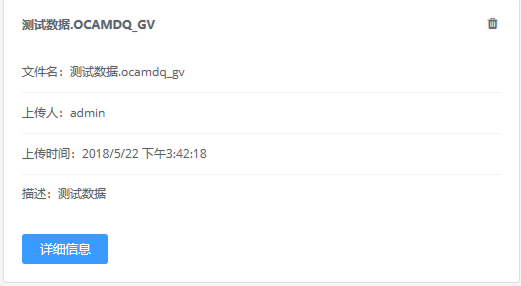
表5‑4 测试范围

|  |  |  |  |
| --- | --- | --- | --- |
| 模块编号 | 模块名称 | 主要功能简述 | 测试结果 |
| 1.1 | 数据上传 | 上传实验数据 | 得到预期结果 |
| 1.2 | 数据可视化 | 展示数据得各项特征 | 得到预期结果 |
| 1.3 | 发表评论 | 查看和发表结论 | 得到预期结果 |
| 2.1 | 用户管理 | 用户的增删改查 | 得到预期结果 |
| 2.2 | 角色管理 | 角色的增删改查 | 得到预期结果 |
| 3.1 | 系统日志 | 查看系统日志 | 得到预期结果 |
| 3.2 | 操作日志 | 查看操作日志 | 得到预期结果 |

### 数据上传模块测试

1. 测试描述

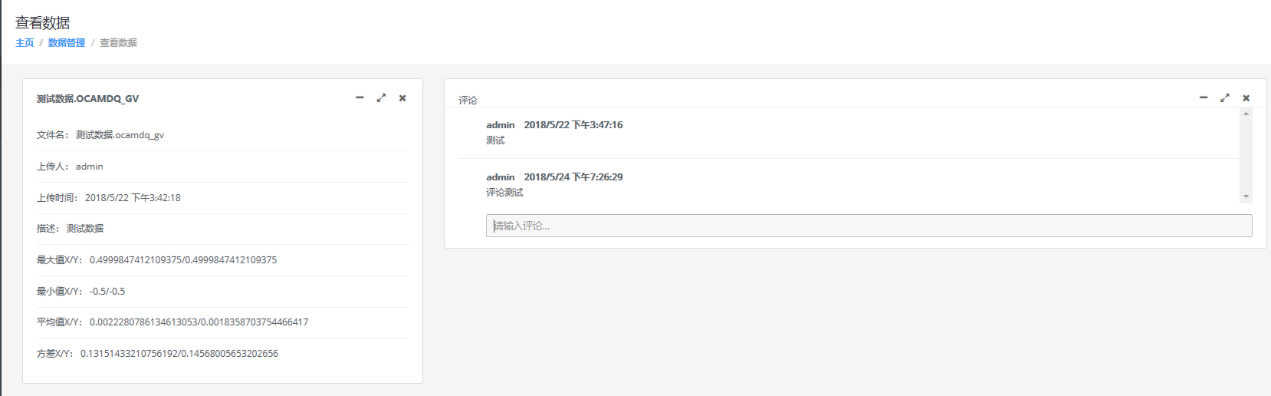
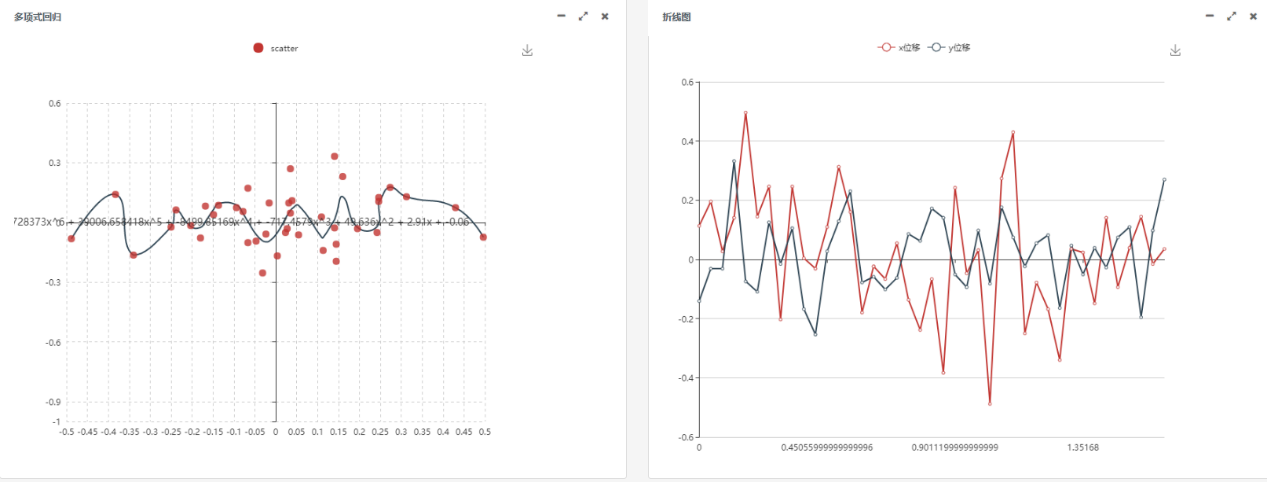
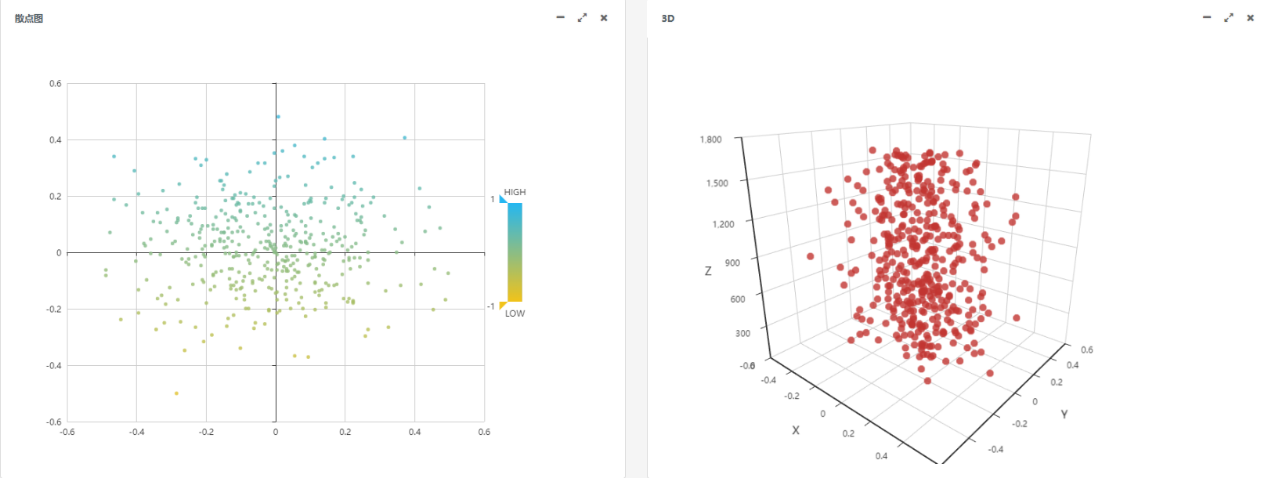
测试使用数据为：16年07月18日\_00时29分23秒.ocamdq-gv

1. 测试结果

### 数据可视化模块测试

1. 测试描述

查看上传数据的各项详细信息

1. 测试结果

## 性能分析

1. 软件能力

经过测试表明，需求分析中提出的所有需求都已完成且程序功能运行正常。

1. 缺陷和限制

角色和用户界面在描述过长的时候会影响整个页面的布局。

在数据量增大之后，前端渲染速度将变得极为缓慢。

1. 测试结论

对比测试结果，本系统可以完成自适应光学分析系统的基础功能。功能较为完备。良好的系统架构为程序的稳定性和健壮性提供了保障，同时为程序的维护和扩展提供了便利。

结 束 语

本系统根据自适应光学实验数据存储和可视化的实际需求，在Spring MVC+Hbase框架下开发而成。系统提供了多样的数据可视化界面，稳定的数据存储方式，快捷的数据查询功能，同时配套了完备的权限认证体系，可追溯的日志查看功能，为数据分析人员提供了一个方便的平台。本系统无需安装，可以通过浏览器直接访问，可以高效的浏览和处理分析实验数据。系统的主要功能包括：数据的导入，数据可视化，发表评论，用户管理，权限管理，日志管理。

但是对于大量数据的可视化，渲染时间将成倍增加，这十分显著的增加了浏览器的负担。同时在前端布局方面，如果字段出现过长的情况，将影响整个界面的布局，使界面显得无序杂乱。

在本项目执行的过程中，项目进度控制存在以下问题。首先是时间划分不合理，主要体现在基础环境的搭建所花费的时间在项目总时间中占比较高，同时预留的用于解决技术问题的时间过少，导致项目进度有时会因为技术难题而不得不延期。

通过本系统，数据分析人员可以高效的存储，查询数据，通过多样化的数据展现方式，处理、分析实验数据，并可以通过本系统与其他用户交流自己的分析结论或发现的问题，充分提高了用户的工作效率。

参考文献

1. 林旭东,薛陈,刘欣悦等.自适应光学波前矫正器技术发展现状[J].中国光学,2012年,第5卷,第4期.
2. 薄奇,许林英.Spring框架中IoC的实现[J].微处理机,2008年,第1期.
3. 百度百科：面向切面编程[EB/OL]. https://baike.baidu.com/item/面向切面编程.
4. 张国平,万仲保,刘高原.Spring Aop框架在J2EE中的应用[J].微计算机信息,2007,第23卷，第12-3期.
5. wawlian：Spring MVC简介[EB/OL]. <http://www.cnblogs.com/wawlian/archive/2012/11/17/2775435.html>.
6. Craig Walls. Manning.Spring.in.Action[M].4th.Edition.北京市丰台区成寿寺路11号：人民邮电出版社.
7. 蒋丛萃,史卓.Spring Security在电子商务中的设计应用[J].电脑知识与技术,2016，第5x期.
8. Apach Hadoop:what is apache hadoop[EB/OL]. <http://hadoop.apache.org/>.
9. BORTHAKUR D. HDFS architecture guide[EB/OL]. (2013T04T04) [2014 T09 T04]. http://hadoop.apache.org/docs/r1.2.1/hdfs\_design.html.Performance of PVM with the MOSIX Preemptive Process Migration Scheme.
10. Dean J, Ghemawat S. MapReduce: Simplified data process-ing on large clusters/ / Proceedings of the 6th Symposium onOperating System Design and Implementation (OSDI 04) .San Francisco, California, USA, 2004: 137-150Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen†, Eric Jul†, Christian Limpach, Ian Pratt, Andrew Warfield Live Migration of Virtual Machines NSDI'05 Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2 2005
11. F. Chang, J. Dean, S. Et al, Bigtable: A Distributed Storage System for Structured Data, OSDI 2006.
12. 维基百科：Apache Hbase[EB/OL]. <https://zh.wikipedia.org/wiki/Apache_HBase>.
13. 张叶,许国艳,花青.基于HBase的矢量空间数据存储与访问优化[J].计算机应用,2015,35(11):3102-3105.
14. Chao-Hsien Lee, Yu-Lin Zheng, ‘ Automatic SQL-to-NoSQL Schema Transformation over the MySQL and HBase Databases’, 2015International Conference on Consumer Electronics.

致　　谢

在论文完成过程中，本人还得到了张欢老师和同课题组同学的热心帮助，在遇到技术难题的时候，张欢老师给予了无私的帮助。在遇到一些论文编写细节问题时，得到了同课题组同学的详细指导。同时在论文编写的这段时间里，父母在生活上给予了诸多帮助，感谢各位无私帮助过我的人，父母，老师和同学！

最后向在百忙之中评审本文的各位专家、老师表示衷心的感谢！

**作者简介：**

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名： | 吴光宇 | 性别： | **男** |
| 出生年月： | 1996年1月20日 | 民族： | **汉** |
| E-mail： | wgy199601@gmail.com |  |  |

声　　明

本论文的工作是2017年10月至2018年6月在成都信息工程大学计算机学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明：

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括：

（1）学校有权保管并向有关部门递交学位论文的原件与复印件。

（2）学校可以采用影印、缩印或其他复制方式保存学位论文。

（3）学校可以学术交流为目的复制、赠送和交换学位论文。

（4）学校可允许学位论文被查阅或借阅。

（5）学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

除非另有科研合同和其他法律文书的制约，本论文的科研成果属于成都信息工程大学。

特此声明！

作者签名：

年 月 日