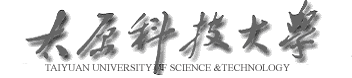
****

**学士学位论文**

**题目：电厂安全培训管理系统**

**——培训管理与系统管理**

**设 计 人：乔利强**

**指导教师：谢斌红**

**所属系部：计算机科学与技术学院**

**专业班级****：软件工程142001**

**学 号：201420020123**

2018年 6 月 15 日

太原科技大学毕业设计（论文）任务书

学院: 计算机科学与技术学院

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学 生 姓 名 | | 乔利强 | 学号 | | 201420020123 |
| 专业班级 | | 软件工程142001班 | 同组人 | | 雷龙 |
| 任务下发时间 | | 2018.3.6 | 任务完成时间 | | 2018.6.6 |
| 设计（论文）题目 | | 电厂安全培训管理系统——培训管理与系统管理 | | | |
| 设计  目的 | 通过设计电厂安全培训管理系统，实现对电厂员工信息的线上管理。将员工的培训管理由线下改为线上，包括线上出题、线上考试、以及线上成绩管理。 | | | | |
| 设计  主要  内容 | 1. 系统分析，对系统进行可行性分析和需求分析，明确系统的开发背景以及意义; 2. 总体设计，包括模块划分、技术选型以及开发环境搭建; 3. 详细设计，包括数据库设计、界面设计、处理过程设计（功能结构设计、类图设计、时序图设计）； 4. 系统实现，完成后台开发功能，与硬件对接; 5. 系统测试，包括单元测试和集成测试。 | | | | |
| 设计  要求  资料 | 1. 完成毕业论文撰写和英文文献翻译。  2. 开发电厂安全培训管理系统一套。  3. 撰写系统操作手册一份。 | | | | |
| 学生签名 |  | | 指导教师签名 |  | |
| 系主任签名 |  | | 主管院长签名 |  | |

中文摘要

安全教育培训是安全生产管理工作中一项十分重要的内容，然而目前存在诸如对于单位信息、人员信息的管理方式无法做到统一集中管理；培训资料只能由安全培训员收集，且在培训过程中使用培训资料比较困难等问题。因此电厂急需要运用现代化的计算机技术、网络技术等信息化技术开发一套基于企业内网的安全培训管理系统。

本文首先对系统的开发背景、开发意义、开发内容等进行了简要的概述，对技术选型进行了分析并确定开发采用SSM框架+Bootstrap前端框架。接着对系统进行了分析，确定了系统的需求模型与用例模型，并确定了系统的总体架构与功能结构。然后设计了系统的CDM、PDM与界面原型。最后采用SSM框架对系统进行了实现，并采用黑盒测试以及抓包技术对开发的系统进行了测试。

本系统主要解决了电厂培训线下管理培训信息与员工部门信息浪费人力物力等问题，为整个电力行业提供了极大的便利，极大的减少了电力行业的安全事故的发生。

关键词: 电厂；培训，Bootstrap；抓包

**Safety training management system of power plant**

**——Training management and system management**

**Author: Qiao Liqiang Tutor: Xie Binhong**

**Abstract**

The training of safety education is a very important content in the work of safety production management. However, there are some problems, such as the management of unit information and personnel information, which can not be unified and centralized management; the training data can only be collected by the safety trainers, and it is difficult to use the training data during the training process. Therefore, power plants urgently need to use modern computer technology, network technology and other information technology to develop a safety training management system based on Intranet.

This paper first gives a brief overview of the development background, development significance, and development content of the system, analyzes the selection of technology and determines the development of the SSM framework +Bootstrap front-end framework. Then we analyze the system, determine the demand model and use case model of the system, and determine the overall architecture and function structure of the system. Then the system's CDM, PDM and interface prototypes are designed. Finally, the system is implemented by SSM framework, and the system is tested by black box testing and packet capture technology.

This system mainly solves the problems of the management training information under the training line of the power plant and the manpower and material resources of the Staff Department, which provides great convenience for the whole power industry and greatly reduces the occurrence of safety accidents in the power industry.

**Key words**: power plant; training; Bootstrap; grab bag

**目录**

[第一章 系统概述 1](#_Toc517021752)

[1.1 开发背景 1](#_Toc517021753)

[1.2 开发意义 2](#_Toc517021754)

[1.3 开发内容 2](#_Toc517021755)

[1.4 开发目标 3](#_Toc517021756)

[1.5 论文结构 3](#_Toc517021757)

[1.6 小结 3](#_Toc517021758)

[第二章 技术选型 4](#_Toc517021759)

[2.1后端开发相关技术 4](#_Toc517021760)

[2.1.1 Spring介绍 4](#_Toc517021761)

[2.1.2 Struts2介绍 5](#_Toc517021762)

[2.1.3 Mybatis主要概述 6](#_Toc517021763)

[2.1.4 Shiro主要概述 7](#_Toc517021764)

[2.1.5 Encache缓存技术简介 8](#_Toc517021765)

[2.1.6 其他相关技术简介 9](#_Toc517021766)

[2.2 前端开发相关技术 9](#_Toc517021767)

[2.2.1 Bootstrap概述 9](#_Toc517021768)

[2.2.2 jQuery简介 9](#_Toc517021769)

[2.2.3 Easyui简介 10](#_Toc517021770)

[2.3项目管理工具 10](#_Toc517021771)

[2.3.1 Maven简介 10](#_Toc517021772)

[2.3.2 Git简介 11](#_Toc517021773)

[2.4 建模技术 12](#_Toc517021774)

[2.5 本章小结 13](#_Toc517021775)

[第三章 系统分析 14](#_Toc517021776)

[3.1 可行性分析 14](#_Toc517021777)

[3.1.1经济可行性 14](#_Toc517021778)

[3.1.2技术可行性 14](#_Toc517021779)

[3.1.3时间可行性 14](#_Toc517021780)

[3.1.4操作可行性 15](#_Toc517021781)

[3.1.5社会可行性 15](#_Toc517021782)

[3.1.6总结 15](#_Toc517021783)

[3.2 需求分析 15](#_Toc517021784)

[3.2.1 需求模型 15](#_Toc517021785)

[3.2.2 功能性需求 17](#_Toc517021786)

[3.2.3 非功能性需求 20](#_Toc517021787)

[3.3 本章小结 21](#_Toc517021788)

[第四章 系统总体设计 22](#_Toc517021789)

[4.1 系统功能结构 22](#_Toc517021790)

[4.2 系统架构设计 23](#_Toc517021791)

[4.2.1 逻辑设计 23](#_Toc517021792)

[4.2.2 三层架构 24](#_Toc517021793)

[4.2.2 软件框架设计 25](#_Toc517021794)

[4.3 系统结构设计 25](#_Toc517021795)

[4.2.1前端结构设计 25](#_Toc517021796)

[4.2.2后端结构设计 27](#_Toc517021797)

[4.2.3 后端包依赖 29](#_Toc517021798)

[4.4 本章小结 29](#_Toc517021799)

[第五章 系统详细设计 30](#_Toc517021800)

[5.1 数据库设计 30](#_Toc517021801)

[5.1.1数据库设计规则 30](#_Toc517021802)

[5.1.2 概念数据模型设计 31](#_Toc517021803)

[5.1.3 物理数据模型设计 32](#_Toc517021804)

[5.1.4 数据库表清单 33](#_Toc517021805)

[5.1.5 数据库主要表的列清单 34](#_Toc517021806)

[5.1.6 视图设计 40](#_Toc517021807)

[5.1.7存储过程和事件设计 41](#_Toc517021808)

[5.2 界面设计 43](#_Toc517021809)

[5.2.1 人机对话设计的原则 44](#_Toc517021810)

[5.2.2 人机对话的方法设计 44](#_Toc517021811)

[5.2.3界面原型 44](#_Toc517021812)

[5.3处理过程设计 48](#_Toc517021813)

[5.3.1 试卷管理模块 48](#_Toc517021814)

[5.3.2 考试管理模块 50](#_Toc517021815)

[5.3.3 成绩管理模块 51](#_Toc517021816)

[5.3.4 员工培训模块 53](#_Toc517021817)

[5.3.5 系统管理模块 54](#_Toc517021818)

[5.4 本章小结 57](#_Toc517021819)

[第六章 系统编码与实现 58](#_Toc517021820)

[6.1 试卷管理模块页面与实现 58](#_Toc517021821)

[6.1.1生成试卷页面 58](#_Toc517021822)

[6.1.2 主要代码 58](#_Toc517021823)

[6.2考试管理模块页面与主要代码 67](#_Toc517021824)

[6.2.1添加考试的页面 67](#_Toc517021825)

[6.2.2核心代码 67](#_Toc517021826)

[6.3成绩管理模块页面与主要代码 72](#_Toc517021827)

[6.3.1分析成绩界面 72](#_Toc517021828)

[6.3.2主要代码 73](#_Toc517021829)

[6.4 培训管理页面与主要代码 74](#_Toc517021830)

[6.4.1 分配员工界面 74](#_Toc517021831)

[6.4.2主要代码 74](#_Toc517021832)

[6.5用户管理模块界面与主要代码 75](#_Toc517021833)

[6.5.1 分配角色界面 75](#_Toc517021834)

[6.5.2主要代码 75](#_Toc517021835)

[6.6 角色管理主要页面与代码 76](#_Toc517021836)

[6.6.1 给角色配置权限页面 76](#_Toc517021837)

[6.6.2 主要代码: 77](#_Toc517021838)

[6.7 字典管理界面以及代码 79](#_Toc517021839)

[6.7.1 添加字典主要界面 79](#_Toc517021840)

[6.7.2 核心代码 79](#_Toc517021841)

[第七章 系统测试与操作手册 81](#_Toc517021842)

[7.1系统测试 81](#_Toc517021843)

[7.1.1试卷管理模块测试 81](#_Toc517021844)

[7.1.2考试管理模块测试 83](#_Toc517021845)

[7.1.3 成绩管理模块测试 84](#_Toc517021846)

[7.1.4 用户管理模块测试 85](#_Toc517021847)

[7.1.5 角色管理模块测试 86](#_Toc517021848)

[7.1.6 字典管理测试 87](#_Toc517021849)

[7.2本章小结 88](#_Toc517021850)

[结束语 89](#_Toc517021851)

[致 谢 90](#_Toc517021852)

[参考文献 91](#_Toc517021853)

[附录Ⅰ 中英文翻译 92](#_Toc517021854)

## 第一章 系统概述

本章着重分析了本系统的开发背景、意义、内容、以及开发目标，最后介绍了论文的结构安排。

### 1.1 开发背景

安全教育培训是安全生产管理工作中一项十分重要的内容，是贯彻安全生产“安全第一、预防为主”的方针，实现生产经营单位安全生产、文明生产、提高员工安全意识和素质、防止发生不安全行为、减少事故伤亡的重要途径。国家相关法律、法规明确规定，电力系统的生产经营单位对员工进行安全生产教育培训成为法定职责。对于新入厂的员工，必须进行更系统的安全教育培训。对于外来临时人员、外来调试人员都应进行对应的系统安全教育。

但是目前电厂安全培训过程中还存在许多问题待解决，例如对于单位信息、人员信息的管理方式无法做到统一集中管理；培训资料只能是由安全培训员收集，且在培训过程中使用培训资料比较困难；安全培训考试的题库、试题、试卷、考试以及阅卷和成绩管理等环节都是完全手工或半手工方式进行，费时费力，效率低且易出错。当员工违章时，安全培训人员要从巨量纸质文档中搜到违章人员档案，录入违章信息后再重新归档。当安全培训员需要统计某单位的违章积分时就要把该单位下的所有员工的违章积分统计出来重新计算才能完成统计。违章制度的不完善使得施工人员在施工过程中不谨慎，经常因为细节施工问题造成巨大的安全施工事故的发生。施工人员入厂方式采用证件识别方式，造成了有外来人员混入厂区的情况，给厂区的安全作业带来了巨大的安全隐患。对违章施工人员无法做到有效的管理，施工人员多次违章却可以多次入厂作业。

总之，电厂急需要运用现代化的计算机技术、网络技术等信息化技术开发一套基于企业内网的安全培训管理系统。

### 1.2 开发意义

电厂安全事故的频发令人担忧，造成这一现象的根本原因是电厂对入厂工作的员工培训不足，致使员工的安全意识不足。而本系统的开发可以解决现在的对员工培训不足的现状，同时改变电厂内部的管理方式。增强入厂工作的员工的安全意识，减少安全事故的发生。本系统的建立，为电厂内部管理提供了极大的便利。更重要的是，将来系统可以利用大数据技术对员工的培训信息进行数据分析，进而针对性的对员工进行培训，减少安全事故的发生。

### 1.3 开发内容

利用Java语言，以及Spring、Struts、Mybatis框架进行整合开发。采用JavaEE三层架构对项目进行整体架构设计，按照系统所需功能划分模块，分模块进行系统开发，易于后期的修改和维护。

经过需求调研分析和设计，确定该系统主要功能模块有：人员管理、检修管理、考试管理、资料管理、新人管理和系统管理。

1)人员管理对员工的个人信息以及培训档案进行自动存档，利用身份证识别仪自动将人员录入到系统中，录入员工违章信息后自动计算员工所在单位的违章信息并管理。检修管理对检修、检修下的单位、单位拥有的员工的信息进行管理。员工的录入同样采用身份证识别仪自动将人员录入到系统中。利用大数据技术对员工的培训情况进行分析，分析后得出的结果数据可得出员工的培训状态，为进一步安排员工的培训提供更加精确的参考信息，由分析出的培训数据预测员工的安全操作的隐患。可以作为下一步的工作。

2)考试管理对每个员工参加的培训进行考核，内部正式员工采用在线考试的方式进行答题，成绩由系统自动阅出;外部的员工采用导出试卷答题卡答题并用判卷机判卷，确保了考试的严谨性。

3)系统管理采用shiro框架进行管理，对不同的人赋予不同的权限，可精确到每个操作按钮，确保每个人履行自己的职责。

### 1.4 开发目标

本系统的开发、实施和应用可从根本上解决了发电企业内部部门和人员信息，以及外来单位和人员不能统一集中管理、试卷与题库管理方式繁琐、考试与判卷过程的繁琐等问题，从根本上解决了违章信息的管理问题，从根本上解决了员工因安全操作意识不强而造成的安全隐患问题。为整个电力行业提供了极大的便利。

### 1.5 论文结构

本文共有七章，第一章首先介绍了系统的背景以及意义。第二章重点分析了系统的前端与后端开发所使用的相关技术。第三章对分析了系统的可行性。第四章对系统进行整体设计，包括功能设计、架构设计与结构设计。第五章进行详细设计，包括系统的数据模型、用户界面以及模块接口。第六章介绍了系统主要功能的编码实现。第七章简要进行了测试。

### 1.6 小结

本章分析了系统的开发背景、开发意义、开发内容、开发目标及项目推广前景，介绍了论文结构安排。对系统的开发内容有了明确的认识，为后期的开发奠定了基础。

## 第二章 技术选型

本系统是一个B/S结构的系统，浏览器端即前端开发主要使用了HTML5、CSS3和JavaScript等技术，服务器端即后端主要使用了Java语言及Spring、Struts、Mybatis与shiro等框架进行开发。

本系统所需的硬件如下:

1) 研腾身份证识别仪：用于扫描身份证录入员工信息。

2) 南昊判卷机：用于对线下考试的答题卡进行阅卷，并导出成绩单。

### 2.1后端开发相关技术

系统后端开发使用的核心技术是Spring，Struts，Mybatis以及Shiro，缓存技术主要是Encache，也用到一些POI、Jsoup等相关技术。本节主要介绍这些核心技术的特点以及原理。

#### 2.1.1 Spring介绍

Spring是一个一站式的开源框架，它可以IOC方式管理DAO、Service与Controller等对象。此外，还可以以AOP方式，解决普通的OOP的代码重复问题，例如，在本项目中通过使用AOP技术主要使用其进行日志管理和事务控制。

任何应用程序开发都可以Spring从中受益。Spring的核心是控制反转（IoC）和面向切面（AOP）。简单来说，spring是一个分层的JavaEE/SE full-stack（一站式）轻量级开源框架。Spring 框架是一个包含7个核心模块的分层架构。Spring框架的七个模块如图2.1所示。

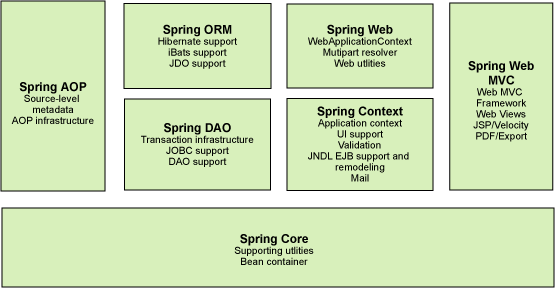


图2.1 spring核心模块图

#### 2.1.2 Struts2介绍

Struts2是一个基于MVC设计模式的Web应用框架，它本质上相当于一个servlet，在MVC设计模式中，Struts2作为控制器(Controller)来建立模型与视图的数据交互。Struts 2是Struts的下一代产品，是在 struts 1和WebWork的技术基础上进行了合并的全新的Struts 2框架。其全新的Struts 2的[体系结构](https://baike.baidu.com/item/%E4%BD%93%E7%B3%BB%E7%BB%93%E6%9E%84)与Struts 1的体系结构差别巨大。Struts 2以WebWork为核心，采用拦截器的机制来处理用户的请求，这样的设计也使得业务逻辑控制器能够与[Servlet](https://baike.baidu.com/item/Servlet)API完全脱离开，所以Struts 2可以理解为WebWork的更新产品。虽然从Struts 1到Struts 2有着太大的变化，但是相对于WebWork，Struts 2的变化很小。项目中主要使用其做MVC转发器，用于接收前端请求且调用service获取数据。其工作原理如图2.2所示:



图2.2 struts2工作原理图

#### 2.1.3 Mybatis主要概述

MyBatis 本是[Apache](http://baike.baidu.com/view/28283.htm)的一个开源项目[iBatis](http://baike.baidu.com/view/628102.htm), 2010年这个项目由Apache software foundation 迁移到了google code，并且改名为MyBatis，实质上Mybatis对ibatis进行一些改进。

MyBatis是一个优秀的持久层框架，它对jdbc的操作数据库的过程进行封装，使开发者只需要关注SQL本身，而不需要花费精力去处理例如注册驱动、创建connection、创建statement、手动设置参数、结果集检索等jdbc繁杂的过程代码。

Mybatis通过xml或注解的方式将要执行的各种statement（statement、preparedStatemnt、CallableStatement）配置起来，并通过java对象和statement中的sql进行映射生成最终执行的sql语句，最后由mybatis框架执行sql并将结果映射成java对象并返回。其主要结构如图2.3所示:



图2.3 Mybatis框架的架构

#### 2.1.4 Shiro主要概述

Shiro是Apache旗下的一个开源的权限管理框架，可以实现用户认证、用户授权。Shiro不依赖于Spring，Shiro不仅可以实现 web应用的权限管理，还可以实现c/s系统，分布式系统权限管理，Shiro属于轻量框架，越来越多企业项目开始使用Shiro。

使用shiro实现系统的权限管理，有效提高开发效率，从而降低开发成本。Shiro不仅能够实现普通的基于菜单的权限管理，也可以实现对界面的按钮的控制，如果再细致一点，可以精确到对某个表的操作进行权限控制。而且shiro可以很好的支持缓存以及分布式session，在中小型系统中都得到重用。其架构如图2.4所示:

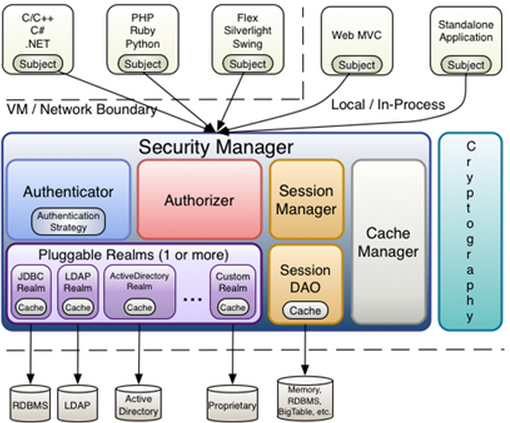


图2.4 Shiro主要架构图

#### 2.1.5 Encache缓存技术简介

Ehcache是一种纯Java语言开发的分布式缓存。主要面向通用缓存,Java EE和轻量级容器。它具有内存和磁盘存储，缓存加载器,缓存扩展,缓存异常处理程序,一个gzip缓存servlet过滤器,支持REST和SOAP api等特点。

在本系统中，其主要用于Mybatis的二级缓存和Shiro的缓存，极大的提高了系统的性能，避免多次访问数据库造成的性能损失。

#### 2.1.6 其他相关技术简介

Apache POI是一个Java语言开发的处理Office文档的开源jar包，可以对Office进行读写操作，其API也是非常的详细。

jsoup是一个Java语言编写的处理HTML数据的开源jar包。它提供了一个非常方便的API来提取和操作数据，使用最好的DOM，CSS和类似jquery的方法。jsoup也被广泛应用于Java爬虫技术。

### 2.2 前端开发相关技术

前端开发主要是基于bootstrap，jQuery，Easyui，zTree以及Kindeditor等，下面对这些技术进行简单的介绍。

#### 2.2.1 Bootstrap概述

Bootstrap是Github上的一个开源项目，其使用简单、方便，可以快速的搭建响应式布局的界面。也目前流行的前端框架之一，其有好多组件可以在项目中快速的使用，包括其模态框、轮播等组件。

#### 2.2.2 jQuery简介

jQuery 是一个 JavaScript 函数库。jQuery 极大地简化了 JavaScript 编程。

jQuery 库包含以下特性：

1)方便敏捷，jQuery有很多的选择器，包括基本选择器、派生选择器以及一些其他的属性选择器。

2)aJax封装，jQuery对aJax提供了三种封装，包括$.post、$.get、$.ajax，在开发中极大的方便了开发。

3)对DOM元素的操作简单易懂，我们可以快速的操作DOM元素。

4)详细的API手册，其API手册非常详细，可以方便开发人员。

#### 2.2.3 Easyui简介

Easyui是一种基于jQuery的用户界面插件集合。Easyui为创建现代化，互动，JavaScript应用程序，提供必要的功能。使用easyui你不需要写很多代码，你只需要通过编写一些简单HTML标记，就可以定义用户界面。Easyui是个完美支持HTML5网页的完整框架。Easyui节省您网页开发的时间和规模。Easyui很简单但功能很强大。在项目中主要使用了其分页插件。

### 2.3项目管理工具

本系统后端构建项目使用的工具是Maven，整个系统的前后端开发使用的分布式版本控制工具是Git。

#### 2.3.1 Maven简介

Maven 是一个项目管理和构建工具。Maven 为开发者提供了一套完整的构建生命周期框架。开发团队几乎不用花多少时间就能够自动完成工程的基础构建配置，因为 Maven 使用了一个标准的目录结构和一个默认的构建生命周期。

Maven可以快速构建项目的工程结构，而且可以使用其插件进行编译运行项目。且其Jar包管理饱受Java开发人员的青睐，构建一个项目无需下载jar包，只需从中央仓库查找对应的位置坐标即可。Jar的冲突解决也是极其的方便，开发人员可以选择第一声明优先原则、路径近者优先原则、排除原则、版本锁定原则等原则解决冲突。Maven 可以让开发者的工作变得更简单。Maven的核心概念图如图2.5所示:



图2.5 Maven核心概念图

#### 2.3.2 Git简介

Git是一个分布式的版本控制工具，对于多人开发的项目使用git可以方便的管理代码。传统的合并代码的方式是手工合并，git可以解决手工合并代码的一系列问题，高效且准确。git可以集成到Eclipse中，也可以单独使用。git最好的使用方式是单独作为工具使用，使用git的客户端工具在里面输入git命令即可。git操作简单且容易学习。是SVN的替代产品。

Git 的工作流程如图2.6所示：

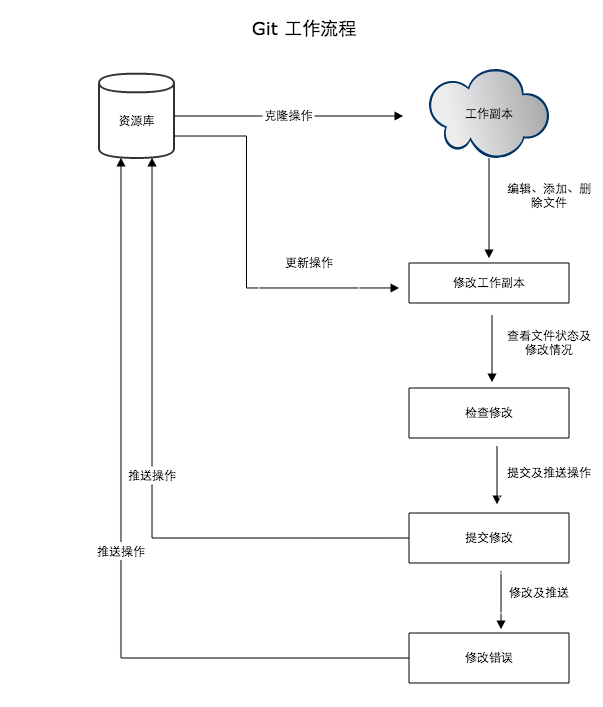


图2.6 Git工作流程图

### 2.4 建模技术

在本系统中，建模技术主要涉及到是业务建模、数据建模以及应用程序建模或者面向对象建模。业务建模主要包括RQM。数据建模主要包括CDM和PDM。应用程序建模主要包括用例图，类图，时序图以及包结构图。用到的建模工具主要是Eclipse和PowerDesigner。

在CDM、RQM、PDM、用例图以及时序图的设计过程中用到了PowerDesigner，在类图与包结构图的设计过程中用到了Eclipse。二者结合使用可以更清晰的了解本系统。

### 2.5 本章小结

本章对于系统开发的语言以及用到的相关技术做了简要的概述，包括后端、前端框架，项目管理工具。

## 第三章 系统分析

本章主要从系统的可行性、以及功能性需求与非功能性需求等方面对系统进行分析，进一步明确系统的需求。

### 3.1 可行性分析

从电厂目前的实际问题出发，本节主要从经济、技术、时间、操作性等方面分析系统是否可行。

#### 3.1.1经济可行性

项目运行后可在行业全面推广，目前整个电力行业的安全施工问题仍然非常严峻，电力行业急需一套完备的安全培训管理系统来支撑入厂施工人员的安全培训。该项目一经推广，可在很大程度上提升安全培训的工作效率，对被培训人员的要求也更加严格且培训效果更加显著。施工人员的安全操作意识可以进一步增强，违章管理制度的加入让施工人员在施工过程中更加小心谨慎，对于减少整个行业因为施工人员的操作而造成的安全隐患问题有显著效果。总之，该项目的经济效益非常乐观。

#### 3.1.2技术可行性

开发小组之前已经有一定的Web项目经验，而且之前的系统的开发均采用的是Spring+Struts+Mybatis+Shiro+Bootstrap，所以该系统的开发从技术方面考虑是完全可行的。

#### 3.1.3时间可行性

整个系统的开发预计需要90天，10天分析需求，10天设计数据库，10天设计界面，40天编写后台完成功能，20天进行整个测试和部署安装。所以从时间角度分析是可行的。

#### 3.1.4操作可行性

系统采用B/S架构开发，目前市面上流行的火狐、谷歌、IE等浏览器均可使用本系统。而且本系统的界面设计从用户的角度考虑，统一采用蓝色背景，给人一种舒适简洁的感觉，所以从这个角度分析也是可行的。

#### 3.1.5社会可行性

电厂安全事故的发生令人堪忧，安全问题成为了大家的关注的焦点。本系统从根本上解决了单位信息和人员信息不能统一集中管理问题，从根本解决了试卷与题库管理方式繁琐的问题，从根本上解决考试过程与判卷过程的繁琐问题，从根本上解决了违章信息的管理问题。从根本上解决了员工因安全操作意识不强而造成的安全隐患问题。为整个电力行业提供了极大的便利。

#### 3.1.6总结

通过经济、技术、时间、操作以及社会方面的可行性分析，可以确定本系统的开发完全必要，而且是可行的，可以立项开发。

### 3.2 需求分析

需求分析准确地回答“系统必须做什么”。需求分析对目标系统提出完整、准确、清晰、具体的要求。本文从业务、功能以及非功能需求三方面来对本系统进行分析。

#### 3.2.1 需求模型

根据用户的业务描述可以得到本系统的需求模型RQM。系统的总需求文档模型如下图3.1所示。

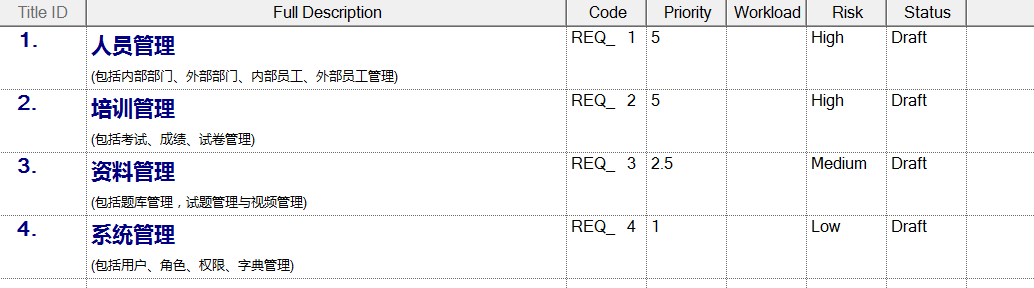


图3.1 总的 RQM图

人员管理又包括内部员工管理和外部员工管理，其RQM如图3.2所示：



图3.2 人员管理RQM

培训管理的 RQM如图3.3所示:



图3.3 培训管理莫快递RQM图

资料管理的 RQM如图3.4所示:



图3.4 资料管理RQM

系统管理的 RQM如图3.5所示:

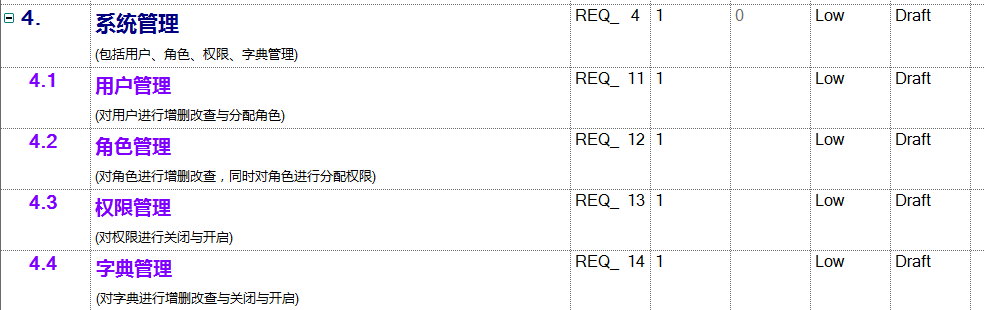


图3.5 系统管理RQM图

#### 3.2.2 功能性需求

功能需求分析了系统必须要完成的功能，使得用户可以通过该系统完成相应的业务需求。

1)角色分析

在电厂安全培训管理系统中，根据实际需求业务分析，主要功能和职责如下表3-1所示。其中角色主要有三个，分别是普通用户，管理员以及超级管理员。

表3-1 系统角色信息描述

|  |  |
| --- | --- |
| 角色 | 主要职责和功能 |
| 普通用户 | 在线考试，查看自己的考试信息，违章信息 |
| 管理员 | 人员管理，培训管理，培训内容管理中的部分功能 |
| 超级管理员 | 人员管理，培训管理，培训内容管理，系统管理 |

2)用例建模

通过用例可以获取用户需求，规划和控制项目。获取用例是需求分析阶段的主要工作之一，而且是首先要做的工作。用例模型描述的是外部行为者所理解的系统功能。用例模型是开发者和用户反复讨论的结果，它描述了开发者和用户对需求规格所达成的共识。

普通用户的用例图如图3.6所示，管理员的用例图如图3.7所示，超级管理员的用例图如图3.8所示:



图3.6 普通用户用例图



图3.7 管理员用例图



图3.8 超级管理员用例图

#### 3.2.3 非功能性需求

在整个系统运行中，除了功能性需求外，一个系统的运行必不可少的一项需求为非功能性需求，包括产品必须遵守的标准，规范以及合约。

1. 行环境约束包括服务器端环境和客户端环境，其中服务器端环境如表3-2所示，客户端境如表3-3所示。

表3-2 服务器端环境

|  |  |
| --- | --- |
| 系统所使用的数据库 | Mysql |
| JDK | 1.7 |
| Web服务器 | Tomcat7.0 |

表3-3 客户端环境

|  |  |
| --- | --- |
| 操作系统 | window7以及以上 |
| 浏览器 | IE8以上即可直接使用 |
| 分辨率 | 1366\*768 |

1. 可保障性需求
2. 可维护性

假如系统的数据的被意外删除，系统仍有对策保证系统的正常运行。同时开发过程中要管理并不断更新详细的文档资料。

1. 可扩展性

系统功能设计要保证在需求发生变化时容易扩展。

1. 可移植性

系统的数据格式符合行业标准，可以实现快速的移植。

1. 易用性需求

操作界面简单，符合的使用习惯，不需要专门的培训即可上手。

1. 安全需求
2. 容错性

假如用户录入了无效或者错误的数据，系统给出友好的提示，不会因为用户操作的失误导致系统崩溃。

1. 系统环境的安全

系统多层次的数据校验，系统中特定的操作需要有相应的权限。系统需要支持数据备份，以免系统运行出错导致崩溃。包括数据库的备份和图片文件的备份。

### 3.3 本章小结

本章主要从系统的可行性和系统的需求两个方面对系统进行分析。可行性分析主要从开发系统的经济、技术、时间、操作以及社会等几个方面论述了系统实施的可行性。需求分析从系统的业务、功能性、非功能性三个方面论述了系统的需求分析。

## 第四章 系统总体设计

本章主要从架构层面对系统进行设计，为详细设计以及开发奠定基础。

### 4.1 系统功能结构

经过前期的分析，大致将系统分为四个主要模块，各模块下又有子模块，主要功能模块如图4.1所示:



图4.1 功能模块图

系统各功能模块功能的简要描述如表4-1所示。

表4-1 系统功能模块介绍表

|  |  |  |  |
| --- | --- | --- | --- |
| 一级功能模块 | 二级模块 | 三级模块 | 功能概述 |
| 人员管理 | 内部员工管理 | 部门管理 | 对内部部门基本信息以及违章信息进行管理 |
| 员工管理 | 内部正式员工基本信息以及违章信息管理 |
| 新员工管理 | 内新入厂的员工进行管理 |
| 外部员工管理 | 检修项目管理 | 对大修以及小修进行管理 |
| 员工管理 | 对外部临时入厂员工进行管理 |
| 黑名单管理 | 对黑名单员工和黑名单单位进行管理 |
| 培训管理 |  | 试卷管理 | 从题库抽取试题形成试卷并对试卷进行修改归档等管理 |
|  | 考试管理 | 从内部部门或者外部部门选取员工进行考试，并可以以Excel格式导出参考名单 |
|  | 成绩管理 | 可以导入员工的成绩与对考试成绩进行分析查看 |
| 员工培训管理 | 短委员工培训 | 临时入厂员工三级考试与分配考试部门 |
| 内部员工培训 | 对新入厂的员工进行三级培训 |
| 资料管理 |  | 题库管理 | 对题库以及试题进行增删改查 |
|  | 培训内容管理 | 对培训视频进行上传与在线播放 |
| 系统管理 |  | 用户管理 | 用用户角色以及密码进行管理 |
|  | 角色管理 | 对角色基本信息进行管理以及分配权限 |
|  | 权限管理 | 对权限进行开启与关闭 |
|  | 字典管理 | 对字典进行增删改查 |

### 4.2 系统架构设计

系统架构设计主要分析系统采用的架构模式与采用的结构模式，进而决定开发步骤。

#### 4.2.1 逻辑设计

本系统采用的软件架构模式是MVC模式，以实现业务逻辑、数据、和表示的代码分离。MVC分别代表model、controller、view。

1) 模型（model）：代表数据，就是业务领域的对象及其属性。

2) 控制器（controller）：是模型与视图的联系纽带。

3) 视图（view）：是模型的外在表现形式。

它的基本逻辑结构如图4.2所示:

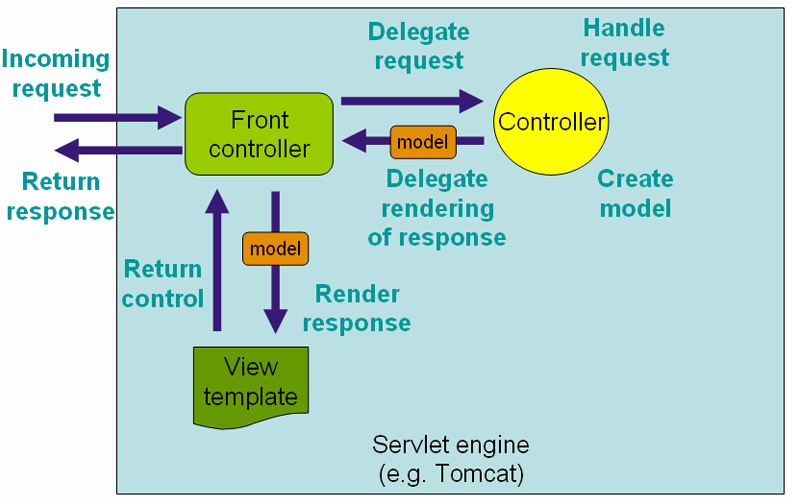


图4.2 MVC逻辑结构图

#### 4.2.2 三层架构

[三层架构](http://baike.baidu.com/item/%E4%B8%89%E5%B1%82%E6%9E%B6%E6%9E%84)(3-tier architecture) 通常意义上的三层架构就是将整个业务应用划分为：界面层（User Interface layer）、业务逻辑层（Business Logic Layer）、数据访问层（Data access layer）。区分层次的目的即为了“[高内聚低耦合](http://baike.baidu.com/item/%E9%AB%98%E5%86%85%E8%81%9A%E4%BD%8E%E8%80%A6%E5%90%88)”的思想。在软件体系架构设计中，分层式结构是最常见，也是最重要的一种结构。微软推荐的分层式结构一般分为三层，从下至上分别为：数据访问层、业务逻辑层（又或称为领域层）、表示层。其主要思想如图4.3所示:



图4.3 三层架构思想图

#### 4.2.2 软件框架设计

由于现在流行的框架较多，且框架能极大地提高开发速度，所以需要从现有的框架中选择优秀的框架。经过分析，确定前端所使用的主要框架是Bootstrap，其响应式布局基本能满足本系统的界面需求。后端采用Spring+Struts+Mybatis+Shiro，后端的这四个框架经过多少年的考研也是极其优秀的开源框架，且本项目的开发人员有这些框架的开发经验，因此选定这些框架。

### 4.3 系统结构设计

系统的结构设计主要确定前端开发的目录结构以及后端开发的目录结构以及相应的模块结构。

#### 4.2.1前端结构设计

前端主要是JS和CSS和JSP页面的组织结构，前端主要结构目录如图4.4所示:

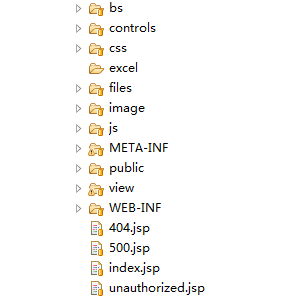


图4.4 前端结构图

bs目录主要存放的是Bootstrap的相关插件的CSS和JS，controls主要存放其他Easyui、zTree等相关插件，CSS是系统css的目录，files用于存放系统一些临时生成的excel等文件，images存放一些图标，js是开发人员写的javascript脚本，public存放公共的JS和jsp文件，view存放的是相关的JSP页面。下面的404.jsp和500.jsp是系统发生404错误和500错误对应的页面，index.jsp是系统的首页，unauthorized.jsp是系统未授权的提示页面。

前端的CSS和JS和View又按照功能模块划分为具体的目录结构。每个目录对应一个响应的功能模块。例如CSS目录下的结果图如图4.5所示:

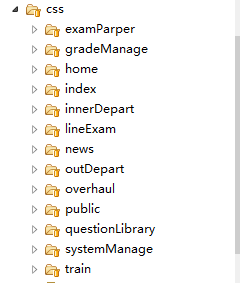


图4.5 CSS目录结构图

examParper目录对应的是考试试卷模块，gradeManage对应的是成绩管理模块，home对应的是系统的公共部分的css，index对应首页的css，innerDepart对应内部部门管理模块，lineExam对应线上考试模块。news对应的新闻管理模块（该模块在后期被取消），outDepart对应的是外部部门管理模块，overhaul对应检修历模块。public是公共的css，questionLibrary是题库与视图管理模块，systemManage是系统管理模块，train是培训资料管理模块。

#### 4.2.2后端结构设计

后端结构设计主要是组织后端的包结构以及按照功能模块图划分具体的包，其总体的包结构如图4.6所示:

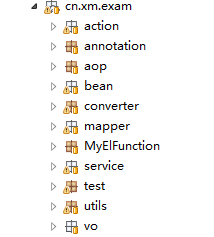


图 4.6 后端总体结构图

action包主要是存放控制层(controller层)代码，主要是接收页面传回来的参数调用service获取数据然后返回给前台页面;service包存放业务层的接口和接口的实现，调用dao层获取数据之后返回给controller层；mapper是存放mapper接口以及对应的sql实现，mapper也就是我们常说的dao层，主要是操作数据库，并返回操作结果；annotatio是自定义注解，主要是为了自定义注解实现日志功能；aop存放的是AOP切面，主要是为了编写通知以及将通知织入切入点形成切面；bean存放基本是实体类，利用mybatis的逆向工程生成的实体类，满足ORM的条件；converter存放一些转换器，比如日期转换器；MyELFunction是自定义El表达式；utils存放系统的一些工具类，包括word转pdf等工具类；vo存放封装一些查询条件的java文件。

具体的每个包下又按模块划分对应的目录，例如bean目录下的结构如图4.7所示:

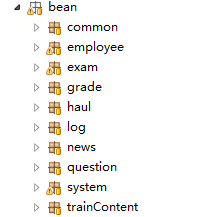


图4.7 bean具体结构图

common是系统的公共包，比如字典和消息；employee存放内部员工和外部员工的基本信息；exam是考试管理的包；grade是成绩管理的包；haul是检修管理的包；log是日志管理的包；new是新闻管理的包(后来此功能被取消)；question是试题和题库管理的包；system是系统管理的包；trainContent是培训内容管理的包。

#### 4.2.3 后端包依赖

后端的包结构设计主要是安装三层架构的思想进行划分，同时也按照功能模块进行划分子包，action包对应三层架构的控制层，action包主要依赖于service包;service包对应三层架构的业务层，service包依赖于mapper包;mapper包对应三层架构的持久层。包依赖关系如图4.8所示:

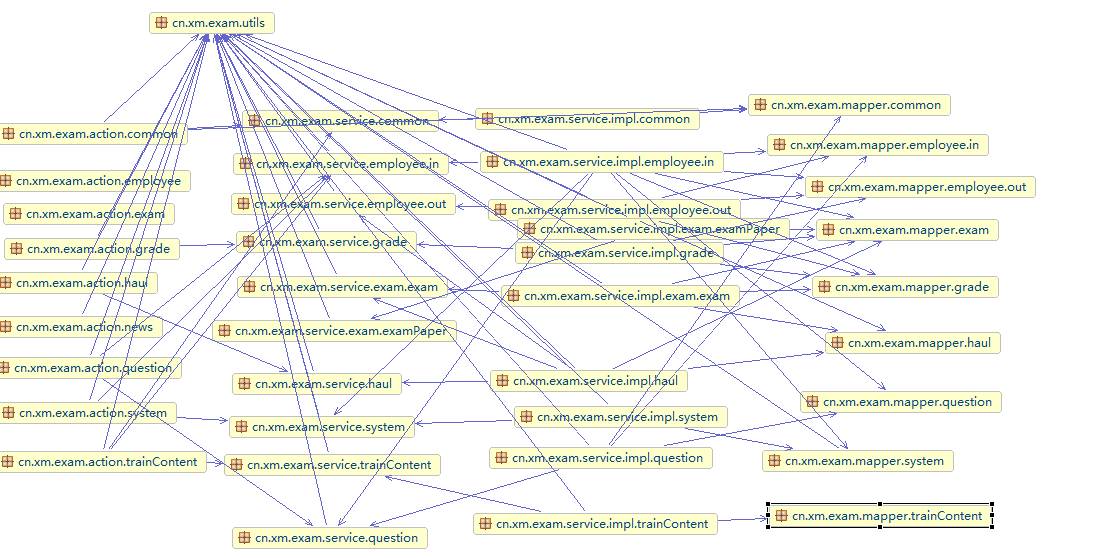


图4.8 后端包依赖图

### 4.4 本章小结

本章确定了系统的功能模块，大致分为四个一级模块；接着确定了系统采用的MVC架构以及三层架构；确定了前端后端所采用的技术框架；最后确定了前端后端的目录结构，为详细设计奠定了基础。

## 第五章 系统详细设计

详细设计阶段的根本目的是确定应该怎么样具体地实现所要求的系统，是对目标系统的精确描述。总体设计负责构件系统整体骨架，详细设计则要考虑各个方面和部件内部细节的方案，诸如系统的输入输出设计、用户界面设计、数据库设计等方面的内容。

### 5.1 数据库设计

数据库是系统的核心，所有的数据都来自于数据库，它的优劣会对系统产生决定性的影响。本系统采用的是mysql数据库，根据前期对系统的需求分析，设计了数据库的CDM，PDM以及数据库脚本。

#### 5.1.1数据库设计规则

主要遵循下列6项规则:

1、数据库命名规范

采用26个英文字母(区分大小写)和0-9的自然数(经常不需要)加上下划线'\_'组成;

2、数据库表名命名规范

采用26个英文字母(区分大小写)和0-9的自然数(经常不需要)加上下划线'\_'组成;

3、数据库表字段名命名规范

采用26个英文字母(区分大小写)和0-9的自然数(经常不需要)加上下划线'\_'组成;

命名简洁明确,多个单词用下划线'\_'分隔;

表与表之间的相关联字段名称要求尽可能的相同;

4、数据库表字段类型规范

用尽量少的存储空间来存数一个字段的数据;

最好给每个字段一个默认值,最好不能为null;

5、数据库表索引规范

命名简洁明确,例如：user\_login表user\_name字段的索引应为user\_name\_index唯一索引;

为每个表创建一个主键索引;

为每个表创建合理的索引;

6、遵循的数据库范式

第一范式(1NF)：字段值具有原子性,不能再分(所有关系型数据库系统都满足第一范式);

第二范式(2NF)：一个表必须有主键,即每行数据都能被唯一的区分;

第三范式(3NF)：一个表中不能包涵其他相关表中非关键字段的信息,即数据表不能有沉余字段;

#### 5.1.2 概念数据模型设计

经过分析与设计，系统管理和培训管理的概念数据模型分别如图5.1和图5.2所示:



图5.1 系统管理模块概念数据模型图



图5.2 培训管理CDM

#### 5.1.3 物理数据模型设计

经过分析与设计，系统管理和培训管理的物理数据模型分别如图5.3和图5.4所示:



图5.3 系统管理模块物理数据模型图



图5.4 培训管理模块PDM

#### 5.1.4 数据库表清单

经过分析与设计，系统初步设计了38张表，系统管理和培训管理模块设计17张表，主要清单如表5-1所示:

表5-1 数据库表清单

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 代码 |
| 1 | 试卷大题表 | bigquestion |
| 2 | 字典表 | dictionary |
| 3 | 成绩表 | employeeexam |
| 4 | 员工培训分配表 | employeeoutdistribute |
| 5 | 考试基本信息表 | exam |
| 6 | 试卷基本信息表 | exampaper |
| 7 | 试卷选项表 | exampaperoption |
| 8 | 试卷题目表 | exampaperquestion |
| 9 | 日志表 | logtable |
| 10 | 消息表 | message |
| 11 | 线上考试答题信息表 | onlineexamanswerinfor |
| 12 | 线上考试登录信息 | onlineexaminfor |
| 13 | 权限表 | permission |
| 14 | 角色表 | role |
| 15 | 角色权限表 | rolepermission |
| 16 | 用户表 | user |
| 17 | 用户角色表 | userrole |

#### 5.1.5 数据库主要表的列清单

数据库培训管理与系统管理主要表与表对应的列清单如下所示

表5-2 bigquestion 的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| bigQuertionId | bigQuertionId | varchar(40) | TRUE | TRUE | 编号 |
| paperId | paperId | varchar(40) | FALSE | FALSE | 试卷编号 |
| bigQuestionName | bigQuestionName | varchar(2000) | FALSE | FALSE | 大题题干 |
| bigQuestionSequence | bigQuestionSequence | int(11) | FALSE | FALSE | 大题顺序 |

表5-3 dictionary的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| dictionaryId | dictionaryId | varchar(40) | TRUE | TRUE | 字典编号 |
| dictionaryName | dictionaryName | varchar(50) | FALSE | TRUE | 字典名称 |
| upDictionaryId | upDictionaryId | varchar(40) | FALSE | FALSE | 上级字典编号 |
| isUse | isUse | varchar(2) | FALSE | TRUE | 是否启用 |
| discription | discription | varchar(200) | FALSE | FALSE | 描述 |

表5-4 employeeexam的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| gradeId | gradeId | int(11) | TRUE | TRUE |  |
| examId | examId | varchar(40) | FALSE | FALSE |  |
| employeeId | employeeId | varchar(40) | FALSE | FALSE |  |
| employeeName | employeeName | varchar(40) | FALSE | FALSE |  |
| grade | grade | float | FALSE | FALSE |  |
| examMethod | examMethod | varchar(5) | FALSE | FALSE |  |
| employeeType | employeeType | varchar(2) | FALSE | FALSE |  |
| unitid | unitid | varchar(40) | FALSE | FALSE | 外来单位ID |
| distributeid | distributeid | int(11) | FALSE | FALSE | 分配表的主键 |
| bigemployeeoutid | bigemployeeoutid | varchar(40) | FALSE | FALSE | 外来大修员工编号 |

表5-5 employeeoutdistribute的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| distributeid | distributeid | int(11) | TRUE | TRUE | 编号 |
| bigid | bigid | varchar(40) | FALSE | FALSE | 大修id |
| haulempid | haulempid | varchar(40) | FALSE | FALSE | 大修员工ID |
| unitid | unitid | varchar(40) | FALSE | FALSE | 单位ID |
| departmentId | departmentId | varchar(40) | FALSE | FALSE | 内部部门ID |
| outempname | outempname | varchar(40) | FALSE | FALSE | 员工姓名 |
| empOutIdCard | empOutIdCard | varchar(18) | FALSE | FALSE | 身份证号 |
| empOutexamStatus | empOutexamStatus | varchar(4) | FALSE | FALSE | 考试状态 |
| empOutTrainGrade | empOutTrainGrade | varchar(2) | FALSE | FALSE | 培训等级 |

表5-6 exam的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| examId | examId | varchar(40) | TRUE | TRUE |  |
| examName | examName | varchar(2000) | FALSE | FALSE |  |
| paperId | paperId | varchar(40) | FALSE | FALSE |  |
| startTime | startTime | datetime | FALSE | FALSE |  |
| endTime | endTime | datetime | FALSE | FALSE |  |
| traincontent | traincontent | varchar(1000) | FALSE | FALSE |  |
| status | status | varchar(40) | FALSE | FALSE |  |
| xueshi | xueshi | varchar(40) | FALSE | FALSE |  |
| bigId | bigId | varchar(40) | FALSE | FALSE | 大修编号 |
| bigStatus | bigStatus | varchar(5) | FALSE | FALSE | 大修状态 |
| examLevel | examLevel | varchar(5) | FALSE | FALSE | 考试等级 |
| employeeName | employeeName | varchar(40) | FALSE | FALSE |  |
| employeeNum | employeeNum | int(11) | FALSE | FALSE |  |
| createTime | createTime | datetime | FALSE | FALSE |  |
| examType | examType | varchar(8) | FALSE | FALSE | 考试类型(内部考试，外部考试) |
| AnswerTime | AnswerTime | int(11) | FALSE | FALSE | 考试答题时间 |
| departmentId | departmentId | varchar(40) | FALSE | FALSE | 部门ID |

表5-7 exampaper的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| paperId | paperId | varchar(40) | TRUE | TRUE |  |
| paperScore | paperScore | float | FALSE | FALSE |  |
| makeTime | makeTime | datetime | FALSE | FALSE |  |
| level | level | varchar(10) | FALSE | FALSE |  |
| employeeName | employeeName | varchar(40) | FALSE | FALSE |  |
| title | title | varchar(100) | FALSE | FALSE |  |
| paperAnswer | paperAnswer | varchar(2) | FALSE | FALSE |  |
| useTimes | useTimes | int(11) | FALSE | FALSE |  |
| description | description | varchar(200) | FALSE | FALSE |  |
| departmentId | departmentId | varchar(40) | FALSE | FALSE | 部门ID |

表5-8 exampaperoption的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| optionId | optionId | int(11) | TRUE | TRUE |  |
| questionId | questionId | varchar(40) | FALSE | FALSE |  |
| optionContent | optionContent | varchar(300) | FALSE | FALSE |  |
| optionSequence | optionSequence | varchar(2) | FALSE | FALSE |  |
| description | description | varchar(2000) | FALSE | FALSE |  |
| isAnswer | isAnswer | varchar(2) | FALSE | FALSE | 是否是答案 |

表5-9 exampaperquestion的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| questionId | questionId | varchar(40) | TRUE | TRUE |  |
| paperId | paperId | varchar(40) | FALSE | FALSE |  |
| bigQuertionId | bigQuertionId | varchar(40) | FALSE | FALSE |  |
| questionContent | questionContent | varchar(200) | FALSE | TRUE |  |
| questionSequence | questionSequence | int(11) | FALSE | TRUE |  |
| type | type | varchar(10) | FALSE | FALSE |  |
| answer | answer | varchar(20) | FALSE | FALSE |  |
| analysis | analysis | varchar(2000) | FALSE | FALSE |  |
| score | score | float | FALSE | FALSE |  |
| questionSource | questionSource | varchar(2) | FALSE | FALSE |  |

表5-10 logtable的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| logid | logid | int(11) | TRUE | TRUE |  |
| operatorName | operatorName | varchar(5) | FALSE | FALSE |  |
| operateDescription | operateDescription | varchar(100) | FALSE | FALSE |  |
| operateResult | operateResult | varchar(2) | FALSE | FALSE |  |
| operateClass | operateClass | varchar(50) | FALSE | FALSE |  |
| operatorId | operatorId | varchar(30) | FALSE | FALSE |  |
| operateMethod | operateMethod | varchar(50) | FALSE | FALSE |  |
| remark | remark | varchar(20) | FALSE | FALSE |  |

表5-11 message的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| messageid | messageid | varchar(40) | TRUE | TRUE | 编号 |
| name | name | varchar(40) | FALSE | FALSE | 姓名 |
| idCode | idCode | char(18) | FALSE | FALSE | 身份证号 |
| sex | sex | varchar(2) | FALSE | FALSE | 性别 |
| birthday | birthday | date | FALSE | FALSE | 生日 |
| empType | empType | varchar(2) | FALSE | FALSE | 员工类型(0:短委,1:内部) |
| isDispose | isDispose | varchar(4) | FALSE | FALSE | 是否处理(0未处理，1处理) |

表5-12 onlineexamanswerinfor的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| answerInfoId | answerInfoId | varchar(40) | TRUE | TRUE |  |
| questionId | questionId | varchar(40) | FALSE | TRUE |  |
| paperId | paperId | varchar(40) | FALSE | FALSE |  |
| questionType | questionType | varchar(40) | FALSE | FALSE |  |
| employeeId | employeeId | varchar(40) | FALSE | FALSE |  |
| answer | answer | varchar(20) | FALSE | FALSE |  |
| selectOption | selectOption | varchar(30) | FALSE | FALSE |  |
| questionScore | questionScore | float | FALSE | FALSE |  |
| score | score | float | FALSE | FALSE |  |
| onlineAnswerExamid | onlineAnswerExamid | varchar(40) | FALSE | FALSE | 考试ID |

表5-13 onlineexaminfor的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| onLineExamId | onLineExamId | varchar(40) | TRUE | TRUE |  |
| examId | examId | varchar(40) | FALSE | FALSE |  |
| employeeId | employeeId | varchar(40) | FALSE | FALSE |  |
| loginTime | loginTime | datetime | FALSE | FALSE |  |
| startTime | startTime | datetime | FALSE | FALSE |  |
| endTime | endTime | datetime | FALSE | FALSE |  |
| ipAddress | ipAddress | varchar(50) | FALSE | FALSE |  |
| examStatus | examStatus | varchar(20) | FALSE | FALSE |  |

表5-14 permission 的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| permissionId | permissionId | varchar(20) | TRUE | TRUE |  |
| name | name | varchar(40) | FALSE | TRUE |  |
| status | status | varchar(2) | FALSE | TRUE |  |
| type | type | varchar(20) | FALSE | FALSE |  |
| url | url | varchar(100) | FALSE | FALSE |  |
| permissionCode | permissionCode | varchar(40) | FALSE | FALSE |  |
| parentId | parentId | varchar(40) | FALSE | FALSE |  |
| parentsId | parentsId | varchar(40) | FALSE | FALSE |  |
| sortNo | sortNo | varchar(40) | FALSE | FALSE |  |
| description | description | varchar(150) | FALSE | FALSE |  |

表5-15 rolepermission 的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| roleId | roleId | varchar(40) | TRUE | TRUE |  |
| permissionId | permissionId | varchar(40) | TRUE | TRUE |  |

表5-16 user 的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| userId | userId | varchar(40) | TRUE | TRUE |  |
| userIDCard | userIDCard | varchar(18) | FALSE | FALSE |  |
| password | password | varchar(40) | FALSE | FALSE |  |
| username | username | varchar(40) | FALSE | FALSE |  |
| departmentName | departmentName | varchar(40) | FALSE | FALSE |  |
| employeeId | employeeId | varchar(40) | FALSE | FALSE |  |
| userPhoto | userPhoto | varchar(40) | FALSE | FALSE |  |
| isUse | isUse | varchar(2) | FALSE | FALSE |  |
| departmentId | departmentId | varchar(40) | FALSE | FALSE |  |
| phone | phone | varchar(20) | FALSE | TRUE |  |
| time | time | datetime | FALSE | FALSE |  |

表5-17 userrole 的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| roleId | roleId | varchar(40) | TRUE | TRUE |  |
| userId | userId | varchar(40) | TRUE | TRUE |  |

表5-18 role 的列清单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 代码 | 名称 | 数据类型 | 主要的 | 强制 | 注释 |
| roleId | roleId | varchar(40) | TRUE | TRUE |  |
| roleName | roleName | varchar(40) | FALSE | TRUE |  |
| roleStatus | roleStatus | varchar(2) | FALSE | TRUE |  |
| description | description | varchar(200) | FALSE | FALSE |  |
| remark | remark | varchar(200) | FALSE | FALSE |  |
| departmentId | departmentId | varchar(40) | FALSE | FALSE |  |
| time | time | datetime | FALSE | FALSE |  |

#### 5.1.6 视图设计

视图是数据库中的虚表，从一个表或者视图中提取出来的数据的一种逻辑的表现形式。视图在数据库中是不存在的，是一个虚表。视图的作用是可以提高数据的安全性，隐藏数据的复杂性，简化查询语句，分离应用程序与基表，保存复杂查询等。

为了简化对员工成绩以及考试情况的查询，创建了employee\_exam\_grade视图，创建的语句如下:

|  |
| --- |
| DELIMITER $$  USE `exam9`$$  DROP VIEW IF EXISTS `employee\_exam\_grade`$$  CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `employee\_exam\_grade` AS  SELECT `exam`.`examId` AS `examId`,`exam`.`examName` AS `examName`,`exam`.`startTime` AS `startTime`,`exam`.`endTime` AS `endTime`,`exam`.`status` AS `status`,`exam`.`paperId` AS `paperId`,`exam`.`bigStatus` AS `bigStatus`,`exam`.`bigId` AS `bigId`,`exam`.`departmentId` AS `departmentId`,`exampaper`.`level` AS `level`,`exampaper`.`paperScore` AS `paperScore`,`employeeexam`.`employeeId` AS `employeeId`,`employeeexam`.`employeeName` AS `employeeName`,`employeeexam`.`employeeType` AS `employeeType`,`employeeexam`.`examMethod` AS `examMethod`,`employeeexam`.`grade` AS `grade`,(CASE (`employeeexam`.`grade` >= (`exampaper`.`paperScore` \* 0.9)) WHEN 1 THEN '是' ELSE '否' END) AS `isPass`,(CASE `employeeexam`.`employeeType` WHEN '0' THEN (SELECT `employee\_in`.`sex` FROM `employee\_in` WHERE (`employee\_in`.`idCode` = `employeeexam`.`employeeId`)) ELSE (SELECT `employee\_out`.`sex` FROM `employee\_out` WHERE (`employee\_out`.`idCode` = `employeeexam`.`employeeId`)) END) AS `sex`,(CASE `employeeexam`.`employeeType` WHEN '0' THEN (SELECT `employee\_in`.`idCode` FROM `employee\_in` WHERE (`employee\_in`.`idCode` = `employeeexam`.`employeeId`)) ELSE (SELECT `employee\_out`.`idCode` FROM `employee\_out` WHERE (`employee\_out`.`idCode` = `employeeexam`.`employeeId`)) END) AS `idCode`,(CASE `employeeexam`.`employeeType` WHEN '0' THEN (CASE (`exam`.`status` = '已结束') WHEN 1 THEN '已交卷' ELSE (SELECT `onlineexaminfor`.`examStatus` FROM `onlineexaminfor` WHERE ((`onlineexaminfor`.`employeeId` = `employeeexam`.`employeeId`) AND (`onlineexaminfor`.`examId` = `employeeexam`.`examId`))) END) ELSE '已交卷' END) AS `answerStatus`,(CASE `employeeexam`.`employeeType` WHEN '0' THEN (SELECT `department`.`departmentName` FROM (`department` JOIN `employee\_in`) WHERE ((`employee\_in`.`idCode` = `employeeexam`.`employeeId`) AND (`employee\_in`.`departmentId` = `department`.`departmentId`))) ELSE (SELECT `unit`.`name` FROM `unit` WHERE (`unit`.`unitId` = `employeeexam`.`unitid`)) END) AS `departmentName` FROM ((`exam` JOIN `exampaper`) JOIN `employeeexam`) WHERE ((`exam`.`examId` = `employeeexam`.`examId`) AND (`exam`.`paperId` = `exampaper`.`paperId`))$$  DELIMITER ; |

#### 5.1.7存储过程和事件设计

存储过程用于执行特定的操作，不需要返回值。在调用时，存储过程可以作为一个独立的表达式使用。有时候我们需要在数据库中周期性的执行一些SQL语句，就可以将这些需要执行的SQL语句设计为一个存储过程，然后创建事件周期性的调用存储过程。事件是mysql周期性的执行协议SQL语句，类似于java的Timer与quartz周期性任务调度。例如通过mysql的事件可以实现每分钟查询年龄大于55岁的人。mysql的事件 本项目设计的存储过程是每天检查员工表，判断员工的年龄是否大于55岁，如果大于55岁需要将大于55岁的插入到message表中，并在界面中作为一个消息提醒相关人员该员工大于55岁。

存储过程设计如下:

|  |
| --- |
| DELIMITER $$  USE `exam9`$$  DROP PROCEDURE IF EXISTS `queryOlderPerson`$$  CREATE DEFINER=`root`@`localhost` PROCEDURE `queryOlderPerson`()  BEGIN  /\*将外部55岁以上插入消息表\*/  INSERT INTO message  (messageid,  NAME,  idCode,  sex,  birthday,  empType,  isDispose)  SELECT DISTINCT  emp.employeeId,  emp.name,  emp.idCode,  emp.sex,  emp.birthday,  '0',  '0'  FROM employee\_out emp,  haulemployeeout haulemp,  haulinfo haul  WHERE emp.idCode = haulemp.empoutIdcard  AND haulemp.bigid = haul.bigid  AND haul.bigstatus = '进行中'  AND emp.employeeId NOT IN(SELECT  messageid  FROM message)  AND TIMESTAMPDIFF(YEAR,emp.birthday,NOW()) > 54;  /\*将内部55岁以上插入消息表\*/  INSERT INTO message  (messageid,  NAME,  idCode,  sex,  birthday,  empType,  isDispose)  SELECT DISTINCT  emp.employeeId,  emp.name,  emp.idCode,  emp.sex,  emp.birthday,  '1',  '0'  FROM employee\_in emp  WHERE emp.employeeId NOT IN(SELECT  messageid  FROM message)  AND TIMESTAMPDIFF(YEAR,emp.birthday,NOW()) > 54;  END$$  DELIMITER ; |

事件设计如下:

|  |
| --- |
| DELIMITER $$  ALTER DEFINER=`root`@`%` EVENT `queryOlderPer` ON SCHEDULE EVERY 1 MINUTE STARTS '2018-01-25 15:54:33' ON COMPLETION NOT PRESERVE ENABLE DO CALL queryOlderPerson()$$  DELIMITER ; |

### 5.2 界面设计

界面设计住户要从用户角度出发，考虑界面的可操作性、舒适性以及简约性。友好的用户界面是信息系统成功的条件之一。

#### 5.2.1 人机对话设计的原则

人机对话设计原则的首要原则是以用户未导向，一切偶从用户的角度考虑。本系统人机对话设计主要遵从下列原则:

1)界面友好性，普通用户可以自己操作界面而不需要培训；

2)界面风格一致性，主要是界面的色调以及按钮等的搭配要做到统一。

3)提示性。对用户的操作亚奥给出提示，使用户知道自己的操作结果。对关键的删除、修改操作要给出提示以及确认。

4)人性化。对需要用户输入的信息进行验证，对一些可选的输入给予初值减少用户的工作量。

#### 5.2.2 人机对话的方法设计

人机对话主要是用户输入信息然后通过鼠标点击或者回车进行数据交互，本系统的主要设计原则如下:

1. 简化输入操作，尽可能使用鼠标而不是键盘，降低键盘输入的数量和错误。
2. 保持信息显示和数据输入之间的一致性。显示的视觉特征应该与输入域一致。

#### 5.2.3界面原型

在这一节展示一些主要功能模块的界面原型。

1.系统首页界面原型如图5.5所示:



图5.5 系统首页图

2.试卷管理界面原型如图5.6所示:



图5.6 试卷管理界面原型图

3.考试管理界面原型如图5.7所示



图5.7 考试管理界面原型图

4.成绩管理界面原型如图5.8所示:



图5.8 成绩管理界面原型

5.员工培训界面原型:



图5.9 员工培训界面原型

6.用户管理界面原型如图5.10所示:



图5.10 用户管理界面原型

7.角色管理界面原型如图5.11所示:



图5.11 角色管理界面原型

8.权限管理界面原型如图5.12所示:



图5.12 权限管理界面原型

9.字典管理界面原型:



图5.13 字典管理界面原型

### 5.3处理过程设计

处理过程主要从功能结构设计、类图、以及时序图三方面对系统进行详细的介。

1. 功能结构设计

在本文的第四章已经对系统的功能结构做了初步的介绍，在此进一步对每个模块进行详细的介绍。

1. 类图设计

类图主要是用来描述模块中具体相关类的联系，还要具体描述每一个类涉及到的变量和方法以及类与类之间的调用关系。

1. 时序图设计

时序图主要用来表示用例中具体的行为过程，同时显示对象之间的交互。通过时序图的设计，可以明确发送消息的内容和参数。

#### 5.3.1 试卷管理模块

试卷管理模块主要对试卷进行增删改查，也要完成对试卷的归档操作。增加试卷有三种方式:从题库抽题、随机抽题、按工种与知识点抽题。

1)主要功能结构如图5.14所示:



图5.14 试卷管理功能模块图

2)主要类图设计如图5.15所示:

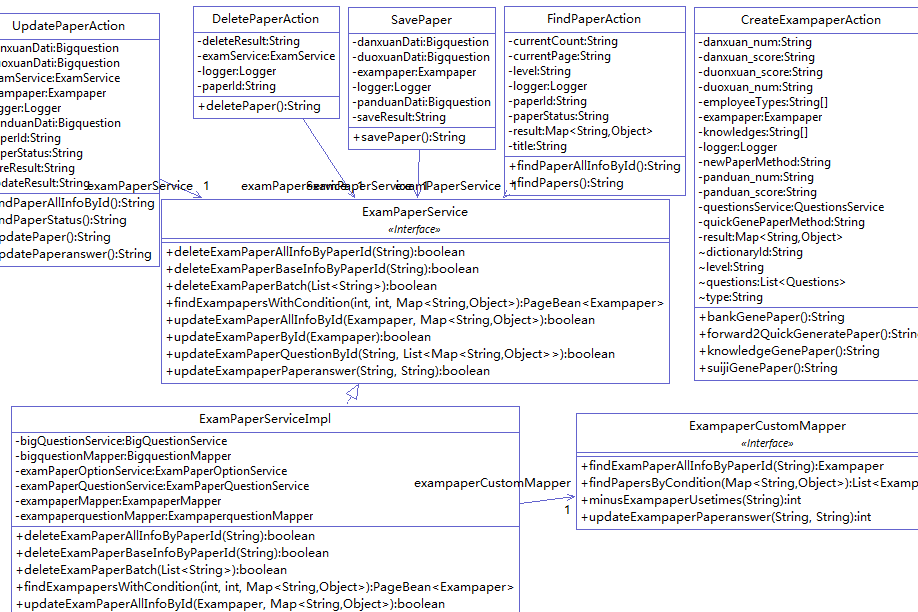


图5.15 试卷管理类图

3)时序图

主要分析了添加试卷的时序图，如图5.16所示:



图5.16 添加试卷时序图

#### 5.3.2 考试管理模块

考试管理主要是增加考试、删除考试、修改考试、查询考试。增加考试需要选择参考人员、选择试卷与填写考试基本信息；修改考试亚奥判断考试状态未开始；删除考试也只能删除未开始的考试；查询考试可以查看参考人员信息以及试卷信息与答案信息。

1)功能结构图如图5.17所示:



图5.17 考试管理功能结构图

2)类图如图5.18所示:

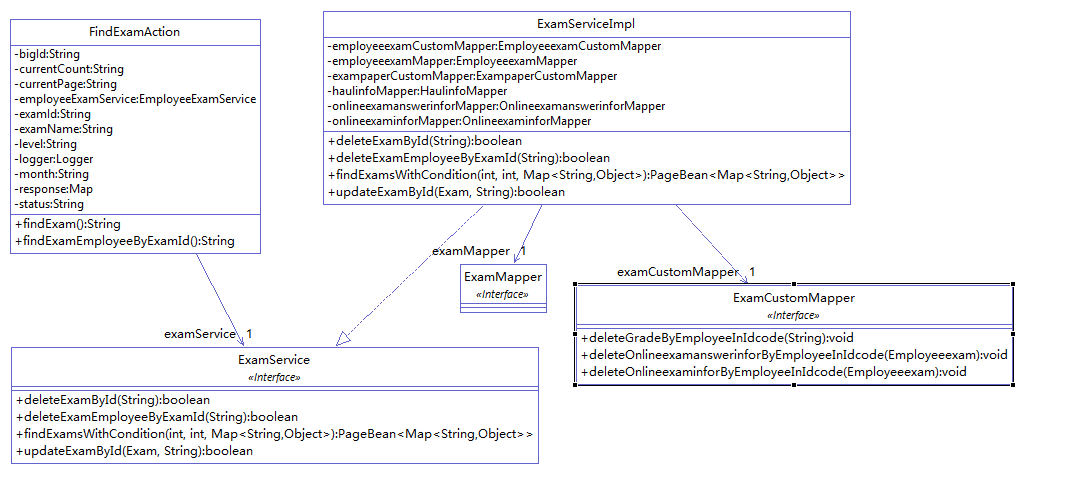


图5.18 考试管理类图

3)时序图

主要分析了添加考试的时序图，如图5.19所示:



图5.19 添加考试时序图

#### 5.3.3 成绩管理模块

成绩管理主要是对线下考试的成绩进行导入，对成绩进行分析，按部门查看成绩等操作。

1)功能模块图如图5.20所示:



图5.20 成绩管理功能结构图

2)类图如图5.21所示:

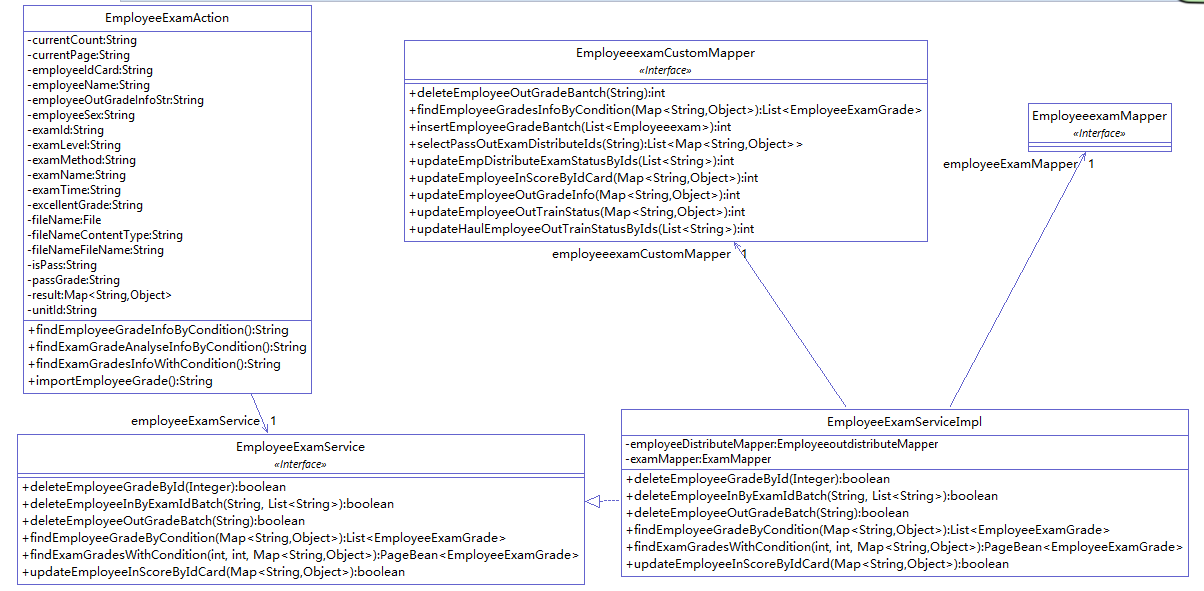


图5.21 成绩管理类图

3)时序图

主要分析了导入成绩的时序图，如图5.22所示

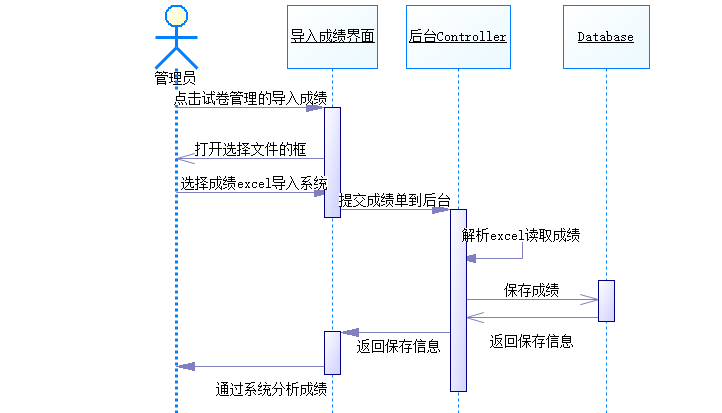


图5.22 导入成绩时序图

#### 5.3.4 员工培训模块

员工培训管理主要是对培训合格的员工进行分配，如果三级都通过可以发放工作证，可以回收工作证。可以也查看员工的培训档案。

1)功能结构图如图5.23所示:



图5.23员工培训功能结构图

2)类图如图5.24所示:

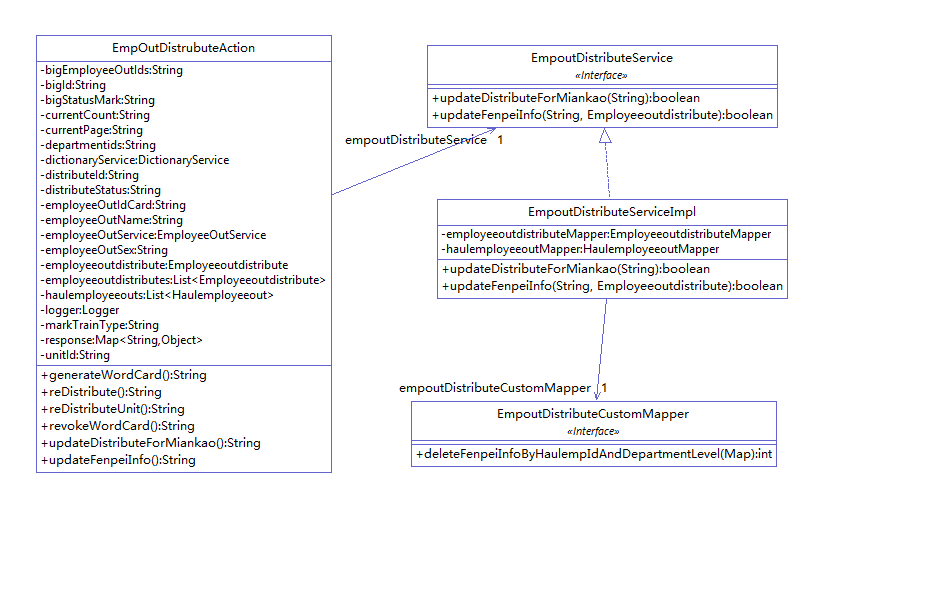


图5.24 员工培训类图

3)时序图

主要分析了员工培训的时序图，如图5.25所示:

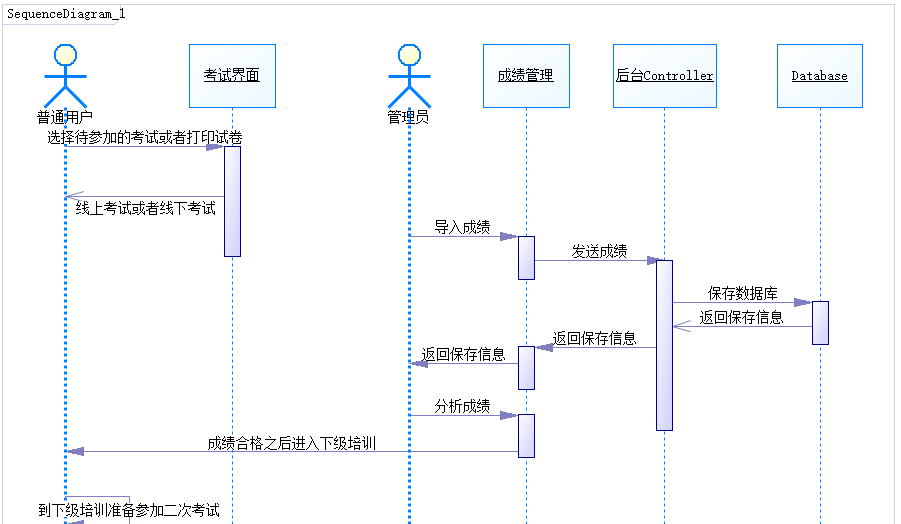


图5.25 员工培训时序图

#### 5.3.5 系统管理模块

系统管理下又分为用户管理、角色管理、字典管理。分别画出了其功能结构图。

1)用户管理可以对用户账户进行开启与锁定，可以对用户密码进行修改，可以给用户分配对应的角色。对应的功能结构图如图5.26所示:



图5.26用户管理功能结构图

角色管理主要是对角色进行增删改查，同时能给角色赋予一定的权限，可以关闭与启用角色。对应的功能结构图如图5.27所示:



图5.27 角色管理功能结构图

字典管理主要是对字典进行增删改查，同时操作的时候应判断字典的等级。对应的功能结构图如图5.28所示:



图5.28 字典管理功能结构图

2)系统管理的类图如图5.29所示:

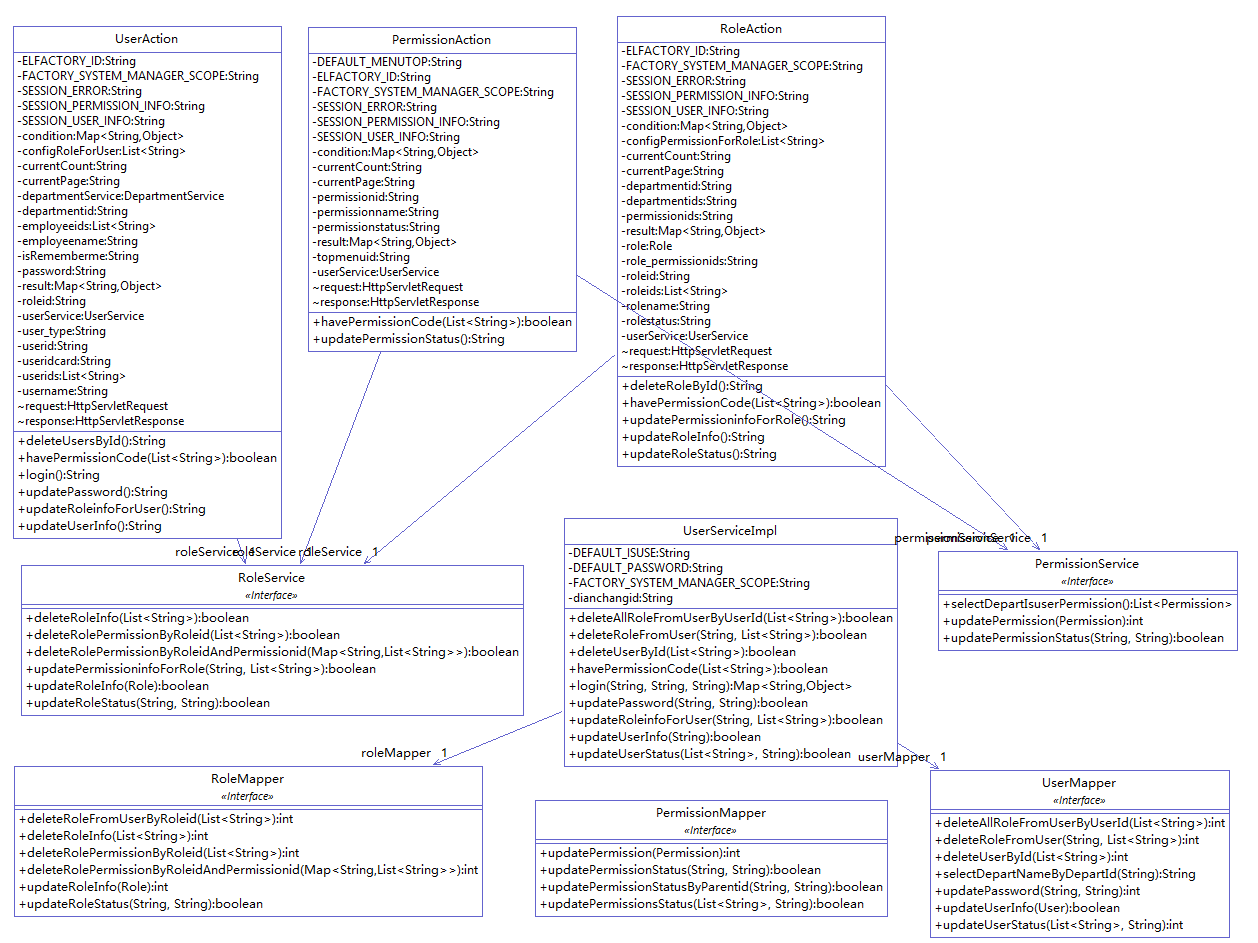


图5.29系统管理类图

3)时序图

主要分析了修改角色的时序图，如图5.30所示

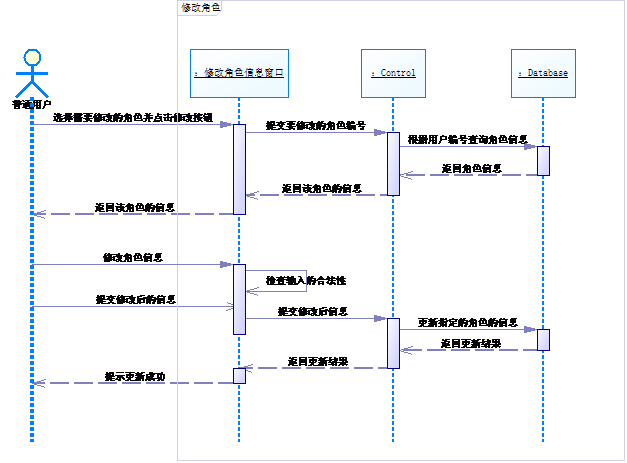


图5.30 修改角色时序图

### 5.4 本章小结

本章首先分析了数据库的设计过程，完成了CDM、PDM以及sql脚本的设计;然后进行了界面设计；最后对模块进行进一步的细化，对系统的业务进行了进一步的了解。为接下来的功能完成奠定了基础。

## 第六章 系统编码与实现

本章主要对系统的培训管理与系统管理的编码实现的核心代码，主要包括核心的JS代码以及SQL语句的编写。对于mybatis逆向工程生成的 sql就不再列出。

### 6.1 试卷管理模块页面与实现

本节主要介绍试卷管理模块的页面的页面与代码实现。

#### 6.1.1生成试卷页面

主要界面如图6.1所示



图6.1 题库抽题界面

#### 6.1.2 主要代码

主要JS代码

|  |
| --- |
| /\*\*  \* qlq写的 根据题库生成试卷的JS 用于生成查询所有右边带复选框的题选择题  \*/  // 页面加载后执行的函数  var isLoad = false;// 全局变量记录是否已经加载  $(function() {  /\*\*  \* 题库试题的提交按钮的点击事件  \*/  $("#queryBankBtn").click(function() {  searchQuestions();  });  });  // 查询试题  var searchQuestions = function() {  $.ajax({  url : contextPath + '/createPaper\_bankGenePaper.action',  data : {  'level' : $("[name='exampaper.level']").val(),  'type' : $("select[name='type'] option:selected").val(),  'dictionaryId' : $("select[name='dictionaryId'] option:selected")  .val()  },  async : true,  dataType : 'json',  type : 'POST',  success : showQuestions,  error : function() {  alert("ajax查询试题失败！")  }  });  }  // 显示到右边题库试题栏  var showQuestions = function(questions) {  $("#bankQuestions").html("");// 清空题  for (var i = 0, length = questions.length; i < length; i++) {  // 拼接单选题  if (questions[i].type == '单选题') {  var danxuan = "<li class='list-group-item el\_drag' style='height: 30px;'>"  + "<input type='checkbox' onclick='el\_tiku\_checkedButtonF(this)' class='el\_tiku\_checkedButton'> "  + "<input type='hidden' id='"  + questions[i].questionid  + "' class='ques\_id'> <!--放 id -->"  + "<!--单选题 class = 'dan'--><div class='movie\_box dan'><div class='tm\_btitlt'>"  + questions[i].question + "</div><ul class='wjdc\_list'>";  // 拼接单选题选项  for (var j = 0, options\_length = questions[i].options.length; j < options\_length; j++) {  danxuan += "<li><label> <input type='radio' value=''><span>"  + questions[i].options[j].optioncontent  + "</span></label></li>";  }  danxuan += '</ul>'  + '答案: <input type="hidden" class="ques\_answer" value="'  + questions[i].answer + '">'  + updateAnswerFormat(questions[i].answer, "12345", "ABCDE")  + '<br />'  /\*+ '解析:<input type="hidden" class="ques\_analy" value="'  + questions[i].analysis + '">' + questions[i].analysis\*/  + '</div> <span class="el\_unflod"> &or;</span></li>';  $("#bankQuestions").append(danxuan);  }  // 拼接多选题  if (questions[i].type == '多选题') {  var duoxuan = "<li class='list-group-item el\_drag' style='height: 30px;'>"  + "<input type='checkbox' onclick='el\_tiku\_checkedButtonF(this)' class='el\_tiku\_checkedButton'>"  + "<input type='hidden' id='"  + questions[i].questionid  + "' class='ques\_id'> <!--放 id -->"  + "<!--单选题 class = 'dan'--><div class='movie\_box duo'><div class='tm\_btitlt'>"  + questions[i].question + "</div><ul class='wjdc\_list'>";  // 拼接多选题选项  for (var j = 0, options\_length = questions[i].options.length; j < options\_length; j++) {  duoxuan += "<li><label> <input type='checkbox' value=''><span>"  + questions[i].options[j].optioncontent  + "</span></label></li>";  }  duoxuan += '</ul>'  + '答案: <input type="hidden" class="ques\_answer" value="'  + questions[i].answer + '">'  + updateAnswerFormat(questions[i].answer, "12345", "ABCDE")  + '<br />'  /\*+ '解析:<input type="hidden" class="ques\_analy" value="'  + questions[i].analysis + '">' + questions[i].analysis\*/  + '</div> <span class="el\_unflod"> &or;</span></li>';  $("#bankQuestions").append(duoxuan);  }  // 拼接判断题  if (questions[i].type == '判断题') {  var panduan = "<li class='list-group-item el\_drag' style='height: 30px;'>"  + "<input type='checkbox' onclick='el\_tiku\_checkedButtonF(this)' class='el\_tiku\_checkedButton'> "  + "<input type='hidden' id='"  + questions[i].questionid  + "' class='ques\_id'><!--放 id -->"  + "<!--单选题 class = 'dan'--><div class='movie\_box pan'><div class='tm\_btitlt'>"  + questions[i].question + "</div><ul class='wjdc\_list'>";  // 拼接判断题选项  for (var j = 0, options\_length = questions[i].options.length; j < options\_length; j++) {  panduan += "<li><label> <input type='radio' value=''><span>"  + questions[i].options[j].optioncontent  + "</span></label></li>";  }  panduan += '</ul>'  + '答案: <input type="hidden" class="ques\_answer" value="'  + questions[i].answer + '">'  + updateAnswerFormat(questions[i].answer, "12345", "ABCDE")  + '<br />'  /\*+ '解析:<input type="hidden" class="ques\_analy" value="'  + questions[i].analysis + '">' + questions[i].analysis\*/  + '</div> <span class="el\_unflod"> &or;</span></li>';  $("#bankQuestions").append(panduan);  }  }  validateCheckedQuestion();// 验证哪些选中打上对勾  }  /\*\*  \* 将source字符串按照s1-s2替换，例如:s1:1234,s2:ABCD则为将source中1换为A，2换为B```  \*/  function updateAnswerFormat(source, s1, s2) {  for (var i = 0, length\_1 = s1.length; i < length\_1; i++) {  source = source.replace(s1.charAt(i), s2.charAt(i));  }  return source;  }  // 验证已经选择的题，如果为已经选中的打上对勾  function validateCheckedQuestion() {  var $leftIds = $(".el\_addEPleft").find(".ques\_id")// 查到左边隐藏的ID元素  var leftIds = [];// 获取到左边已经选择的试题ID数组  for (var i = 0, length\_1 = $leftIds.length; i < length\_1; i++) {  leftIds[i] = $($leftIds[i]).attr("id");  }  // 根据左边的元素与右边查找如果已经选中的话打上对勾  for (var k = 0, length\_2 = leftIds.length; k < length\_2; k++) {  $("#bankQuestions").find("[id='" + leftIds[k] + "']").siblings(  ".el\_tiku\_checkedButton").attr("checked", "true");  }  } |

后台保存试卷代码:

|  |
| --- |
| @Override  public boolean addExamPaper(Exampaper examPaper, Map<String, Object> bigQues) throws SQLException {  if (examPaper == null) {  return false;  }  // 获取试卷ID，如果为空产生一个ID  String paperId = examPaper.getPaperid();  if (ValidateCheck.isNull(paperId)) {  paperId = UUIDUtil.getUUID2();// 产生试卷ID  examPaper.setPaperid(paperId);  examPaper.setUsetimes(0);// 第一次增加的时候将试卷使用次数设置为0  }  // 修改试卷的答案，用来存放试卷的归档与否:0代表未归档，1代表归档  if(ValidateCheck.isNull(examPaper.getPaperanswer())){  examPaper.setPaperanswer("0");  }  // 如果部门的ID为空的话就设置部门ID  if (ValidateCheck.isNull(examPaper.getDepartmentid())) {  User user = (User) ServletActionContext.getRequest().getSession().getAttribute("userinfo");  String departmentIdSession = user == null ? null : user.getDepartmentid();// 获取到部门ID  examPaper.setDepartmentid(departmentIdSession);  }  // 如果试卷描述没写就写一个空的试卷描述  if (ValidateCheck.isNull(examPaper.getDescription())) {  examPaper.setDescription("无");  }  boolean result = this.addExamPaperBaseInfo(examPaper);// 添加试卷的基本信息  if (result) {  Bigquestion danxuanDati = (Bigquestion) bigQues.get("danxuanDati");// 获取单选大题  Bigquestion duoxuanDati = (Bigquestion) bigQues.get("duoxuanDati");// 获取单选大题  Bigquestion panduanDati = (Bigquestion) bigQues.get("panduanDati");// 获取单选大题  // 添加单选题  if (danxuanDati != null) {  danxuanDati.setPaperid(paperId);  String danxuanDatiId = UUIDUtil.getUUID2();// 产生单选大题ID  danxuanDati.setBigquertionid(danxuanDatiId);  String bigTitle = danxuanDati.getBigquestionname();// 获取到单选大题的题干  Document doc = Jsoup.parse(bigTitle);// 转为HTML  String val = doc.select(".el\_modifiedGrade").get(0).val();// 获取到分值  // 添加单选大题  if (bigQuestionService.addBigQuestion(danxuanDati)) {  // 添加题与选项  List<Exampaperquestion> questions = danxuanDati.getQuestions();  for (int i = 0, size = questions.size(); i < size; i++) {  Exampaperquestion question = questions.get(i);  question.setType("单选题");  String questionid = UUIDUtil.getUUID2();  question.setQuestionid(questionid);  question.setPaperid(paperId);  question.setBigquertionid(danxuanDatiId);  question.setScore(Float.valueOf(val));// 设置每道题的分值  // 添加试题  if (examPaperQuestionService.addExamPaperQuestion(question)) {  List<Exampaperoption> options = question.getOptions();  for (int j = 0, optionSize = options.size(); j < optionSize; j++) {  Exampaperoption exampaperoption = options.get(j);  exampaperoption.setQuestionid(questionid);  // 添加选项  examPaperOptionService.addPaperOption(exampaperoption);  }  }  }  }  }  // 添加多选题  if (duoxuanDati != null) {  duoxuanDati.setPaperid(paperId);  String duoxuanDatiId = UUIDUtil.getUUID2();// 产生单选大题ID  duoxuanDati.setBigquertionid(duoxuanDatiId);  String bigTitle = duoxuanDati.getBigquestionname();// 获取到多选大题的题干  Document doc = Jsoup.parse(bigTitle);// 转为HTML  String val = doc.select(".el\_modifiedGrade").get(0).val();// 获取到每道题分值  // 添加单选大题  if (bigQuestionService.addBigQuestion(duoxuanDati)) {  // 添加题与选项  List<Exampaperquestion> questions = duoxuanDati.getQuestions();  for (int i = 0, size = questions.size(); i < size; i++) {  Exampaperquestion question = questions.get(i);  question.setType("多选题");  String questionid = UUIDUtil.getUUID2();  question.setQuestionid(questionid);  question.setPaperid(paperId);  question.setBigquertionid(duoxuanDatiId);  question.setScore(Float.valueOf(val));// 设置每道题的分值  // 添加试题  if (examPaperQuestionService.addExamPaperQuestion(question)) {  List<Exampaperoption> options = question.getOptions();  for (int j = 0, optionSize = options.size(); j < optionSize; j++) {  Exampaperoption exampaperoption = options.get(j);  exampaperoption.setQuestionid(questionid);  // 添加选项  examPaperOptionService.addPaperOption(exampaperoption);  }  }  }  }  }  // 添加判断题  if (panduanDati != null) {  panduanDati.setPaperid(paperId);  String panduanDatiId = UUIDUtil.getUUID2();// 产生单选大题ID  panduanDati.setBigquertionid(panduanDatiId);  String bigTitle = panduanDati.getBigquestionname();// 获取到单选大题的题干  Document doc = Jsoup.parse(bigTitle);// 转为HTML  String val = doc.select(".el\_modifiedGrade").get(0).val();  // 添加单选大题  if (bigQuestionService.addBigQuestion(panduanDati)) {  // 添加题与选项  List<Exampaperquestion> questions = panduanDati.getQuestions();  for (int i = 0, size = questions.size(); i < size; i++) {  Exampaperquestion question = questions.get(i);  question.setType("判断题");  String questionid = UUIDUtil.getUUID2();  question.setQuestionid(questionid);  question.setPaperid(paperId);  question.setBigquertionid(panduanDatiId);  question.setScore(Float.valueOf(val));// 设置每道题的分值  // 添加试题  if (examPaperQuestionService.addExamPaperQuestion(question)) {  List<Exampaperoption> options = question.getOptions();  for (int j = 0, optionSize = options.size(); j < optionSize; j++) {  Exampaperoption exampaperoption = options.get(j);  exampaperoption.setOptionsequence(String.valueOf(j));  exampaperoption.setQuestionid(questionid);  // 添加选项  examPaperOptionService.addPaperOption(exampaperoption);  }  }  }  }  }  }  return true;  } |

### 6.2考试管理模块页面与主要代码

本节主要介绍考试管理的页面与主要代码。

#### 6.2.1添加考试的页面

界面如图6.2所示:



图6.2 添加考试界面

#### 6.2.2核心代码

JS代码

|  |
| --- |
| /\*\* \*S 保存考试 \*/  var saveExam = function() {  // 进行数据验证  $.validator.methods.compareDate = function(value, element, param) {  var startDate = $("#inpstart0").val(); // 开始日期  var d1 = new Date(startDate);  var d2 = new Date(value);  return d1 <= d2;  };  var isNotNull = $("#examForm").validate({  ignore : [],  rules : {  "exam.paperid" : "required",// 发现日期  "exam.examname" : {  required : true  },  "exam.traincontent" : {  required : true  },  "exam.xueshi" : {  required : true,  digits : true  },  "exam.answertime" : {  required : true,  digits : true,  range : [ 20, 400 ]  },  "exam.address" : "required",// 验证文本框的。前边是 name 属性  "exam.examlevel" : "required",// 验证文本框的。前边是 name 属性  "exam.starttime" : "required",  "exam.endtime" : {  required : true/\*,  compareDate : "#inpend0"\*/  },  "exam.employeename" : {  required : true  },  "employeeFlag" : "required"  },  messages : {  "exam.paperid" : {  required : "必须选择一份考试试卷"  },// 下边与上边对应  "exam.examname" : {  required : "考试名称不能为空"  },  "exam.traincontent" : {  required : "培训内容不能为空"  },  "exam.xueshi" : {  required : "培训时长不能为空",  digits : "必须输入整数"  },  "exam.examlevel" : {  "required" : "考试等级不能为空"  },// 验证文本框的。前边是 name  "exam.answertime" : {  required : "考试时长不能为空",  digits : "考试时长必须是整数",  range : "考试时长必须在20到400分钟范围内"  },  "exam.address" : "考试地址不能为空",// 验证文本框的。前边是 name 属性  "exam.starttime" : "开始时间不能为空",  "exam.endtime" : {  required : "结束时间不能为空"/\*,  compareDate : "结束日期不能小于开始日期"\*/  },  "exam.employeename" : {  required : "创建人不能为空"  },  "employeeFlag" : "必须选择参考员工"  }  });  // 如果验证通过可以提交数据  if (isNotNull.form()) {  if (confirm("确认保存考试并退出?")) {  $("#saveExamBtn").prop("disabled", true);  $.ajax({  url : contextPath + '/exam\_addExam.action',  data : $("#examForm").serialize(),  type : 'POST',  dataType : 'json',  async : true,  success : function(response) { //  alert(response.result);  if (response.result == '添加成功！') {  window.location.href = contextPath  + '/view/examParper/exam/examManage.jsp';  }  },  error : function() {  alert("保存试卷失败！！！")  }  });  }  }  }  /\*\* \*E 保存考试 \*/ |

保存考试的 后台代码:

|  |
| --- |
| @Override  public boolean addExam(Exam exam,String examMethod) throws Exception {  if (exam == null) {  return false;  }  // 获取内外部员工  List<Employeeexam> employeeInExams = exam.getEmployeeInExams();// 内部员工  List<Employeeexam> employeeOutExams = exam.getEmployeeOutExams();// 外部员工  boolean insertEmployeeResult = false;  int in\_size, out\_size;  in\_size = employeeInExams != null ? employeeInExams.size() : 0;  out\_size = employeeOutExams != null ? employeeOutExams.size() : 0;  // 获取参考人数  Integer employeeNum = in\_size + out\_size;  String examId = exam.getExamid();  // 如果考试的id为空就UUID生成一个ID(与修改考试用的同一个Action)  if (ValidateCheck.isNull(examId)) {  examId = UUIDUtil.getUUID2();// 考试ID  exam.setExamid(examId);  }  // 如果部门ID为空就获取当前session中的用户的部门ID并设置考试的部门ID  if (ValidateCheck.isNull(exam.getDepartmentid())) {  User user = (User) ServletActionContext.getRequest().getSession().getAttribute("userinfo");  String departmentIdSession = user == null ? null : user.getDepartmentid();// 获取到部门ID  exam.setDepartmentid(departmentIdSession);  }  if (ValidateCheck.isNull(exam.getExamtype())) {  exam.setExamtype("外部考试");  }  // 创建时间为空就创建一个创建时间  if (exam.getCreatetime() == null) {  exam.setCreatetime(new Date());// 设置创建时间  }  // 大修ID不为空的话证明是外部的单位参加的考试，调用大修ID去获取大修状态(为空或者不为无证明是外部员工)  if (ValidateCheck.isNotNull(exam.getBigid()) && !"无".equals(exam.getBigid())) {  exam.setBigstatus(haulinfoMapper.selectByPrimaryKey(exam.getBigid()).getBigstatus());// 设置考试的大修状态  }  // 大修ID为空的话证明是内部的单位参加的考试。设置一个大修ID的默认值与大修状态  if (ValidateCheck.isNull(exam.getBigid()) || "无".equals(exam.getBigid())) {  exam.setBigstatus("无");  exam.setBigid("无");  exam.setExamtype("内部考试");  }  exam.setStatus("未开始");// 设置考试状态为未考  exam.setEmployeenum(employeeNum);// 设置参考人数  insertEmployeeResult = examMapper.insert(exam) > 0 ? true : false;  if (employeeInExams != null && employeeInExams.size() > 0) {  for (Employeeexam employeeIn : employeeInExams) {  employeeIn.setExamid(examId);  employeeIn.setExammethod(examMethod);  employeeIn.setGrade(0f);  //将外部员工独有的置为空  employeeIn.setDistributeid(0);  employeeIn.setBigemployeeoutid("0");  // employeeIn.setUnitid("无");  employeeIn.setEmployeetype("0");// 0代表内部员工  }  // 插入内部员工  insertEmployeeResult = employeeexamCustomMapper.addEmployeeExam(employeeInExams) > 0 ? true : false;  }  if (employeeOutExams != null && employeeOutExams.size() > 0) {  for (Employeeexam employeeOut : employeeOutExams) {  employeeOut.setExamid(examId);  employeeOut.setExammethod(examMethod);  employeeOut.setGrade(0f);  employeeOut.setEmployeetype("1");// 1代表外部员工  }  // 插入外部员工  insertEmployeeResult = employeeexamCustomMapper.addEmployeeExam(employeeOutExams) > 0 ? true : false;  }  // 试卷的使用次数增加一  exampaperCustomMapper.addExampaperUsetimes(exam.getPaperid());// 试卷使用次数增加一次  return insertEmployeeResult;  } |

### 6.3成绩管理模块页面与主要代码

本节主要介绍成绩管理的页面与主要代码。

#### 6.3.1分析成绩界面

如图6.3所示:

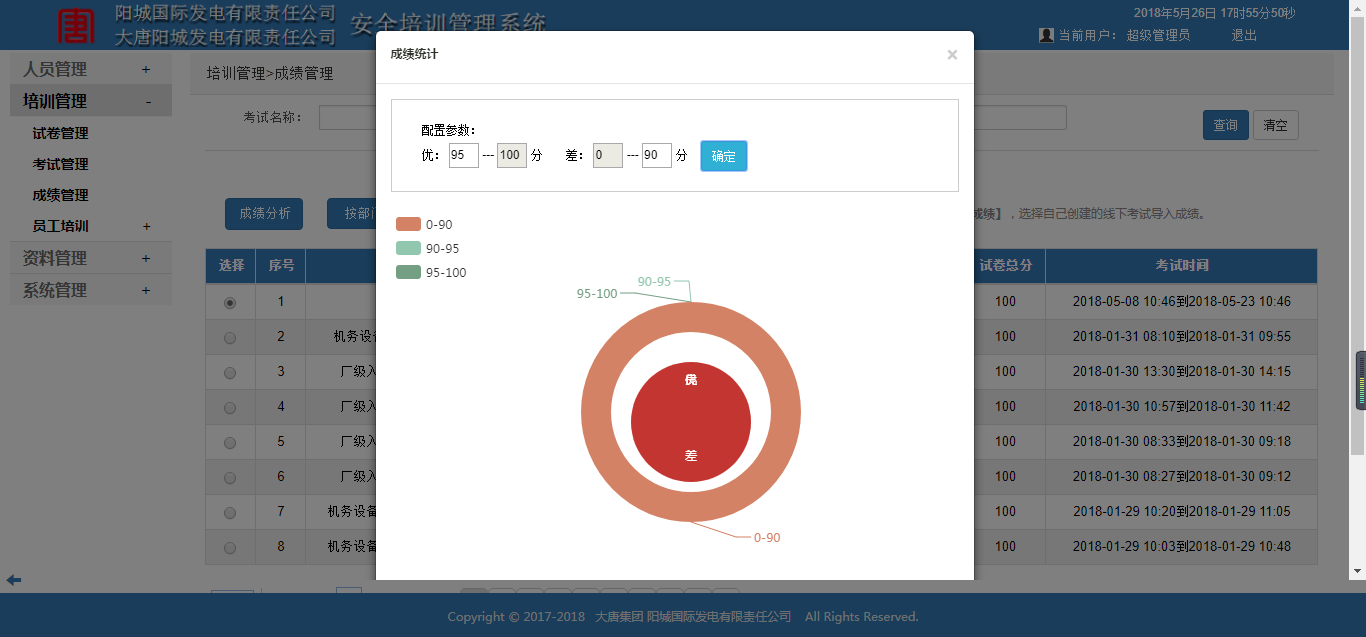


图6.3 成绩分析界面

#### 6.3.2主要代码

核心JS代码:

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*成绩分析\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //点击页面的成绩分析执行的操作  function gradeAnaly() {  //累计选择的个数，若等于1，才执行，否则提示  var nums = 0;  $.each($(".el\_checks"), function (i, el\_check) {  if ($(this).prop("checked")) {  //将隐藏域中的考试ID传入成绩分析的模态框中  $("#gradeAnalyse\_examId").val($(this).val());  var unitId = $(this).parents("tr").find(".query\_unitid").val();  $("#gradeAnalyse\_unitId").val(unitId);  nums++;  }  });  if (nums == 1) {  //先将图表的div清空  $("#main").empty();    $('#el\_scoreStatistisModel').modal();  } else {  alert("请先选择一个考试！")  }  } |

### 6.4 培训管理页面与主要代码

本节主要介绍培训管理模块的页面与主要代码。

#### 6.4.1 分配员工界面

如图6.4所示:

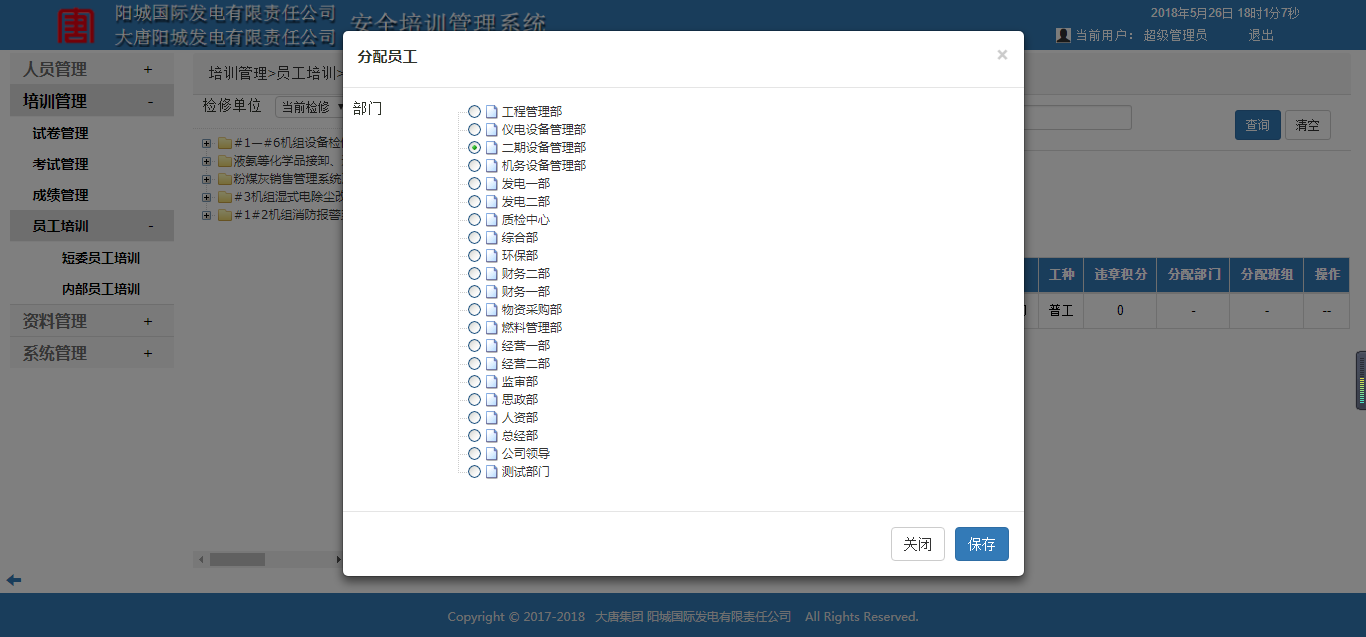


图6.4 分配员工界面

#### 6.4.2主要代码

核心JS代码

|  |
| --- |
| /\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*分配员工\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  function el\_empTrainDoc() {  var chooseEmpNum = 0;// 判断是否有员工被选中  $(".el\_checks").each(function() { // 获取选择的员  if ($(this).prop("checked")) {// 如果选中。。。  chooseEmpNum++;  }  }  if (chooseEmpNum != 0) {  $("#el\_empTrainDoc").modal()  } else {  alert("请先选择员工！")  }  } |

### 6.5用户管理模块界面与主要代码

本节主要介绍用户管理模块的核心代码与主要界面。

#### 6.5.1 分配角色界面

如图6.5所示：



图6.5 分配角色界面

#### 6.5.2主要代码

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*S 设置单个用户权限相关操作\*\*\*\*\*/  function setUserRole(obj){  //0.将角色列表的多选框清空  $(".role4oneUser").prop("checked",false);//清空多选的选择  //1.判断是否已经配置角色  //1.1未配置的直接清空多选框然后打开模态框  var text = $(obj).parents("tr").children("td:eq(8)").text();  var userId = $(obj).parents("tr").find(":checkbox").val();  $("#hidden\_one\_userId").val(userId);//向模态框隐藏的userId赋值  if(text === "已配置"){  $.post(contextPath+'/user/getUserRoleByUserId.do',{"userId":userId},function (response) {  for(var i=0,length\_1 = response.length;i<length\_1;i++){  $(".role4oneUser[value='"+response[i].roleID+"']").prop("checked",true);  }  },'json');  }  //2.打开模态框  var width=($(window).width()\*0.60);  var index = layer.open({  title:'配置用户角色',  area: width+'px',//大小  fixed: true, //不固定  maxmin: true,  zIndex:500,  shadeClose: false,  shade:0.4,  type:1,  content:$('#set\_oneUserRole\_modal')  });  //向页面隐藏index  $("#hidden\_set\_oneUserRole\_index").val(index);  } |

### 6.6 角色管理主要页面与代码

#### 6.6.1 给角色配置权限页面

如图6.6所示:

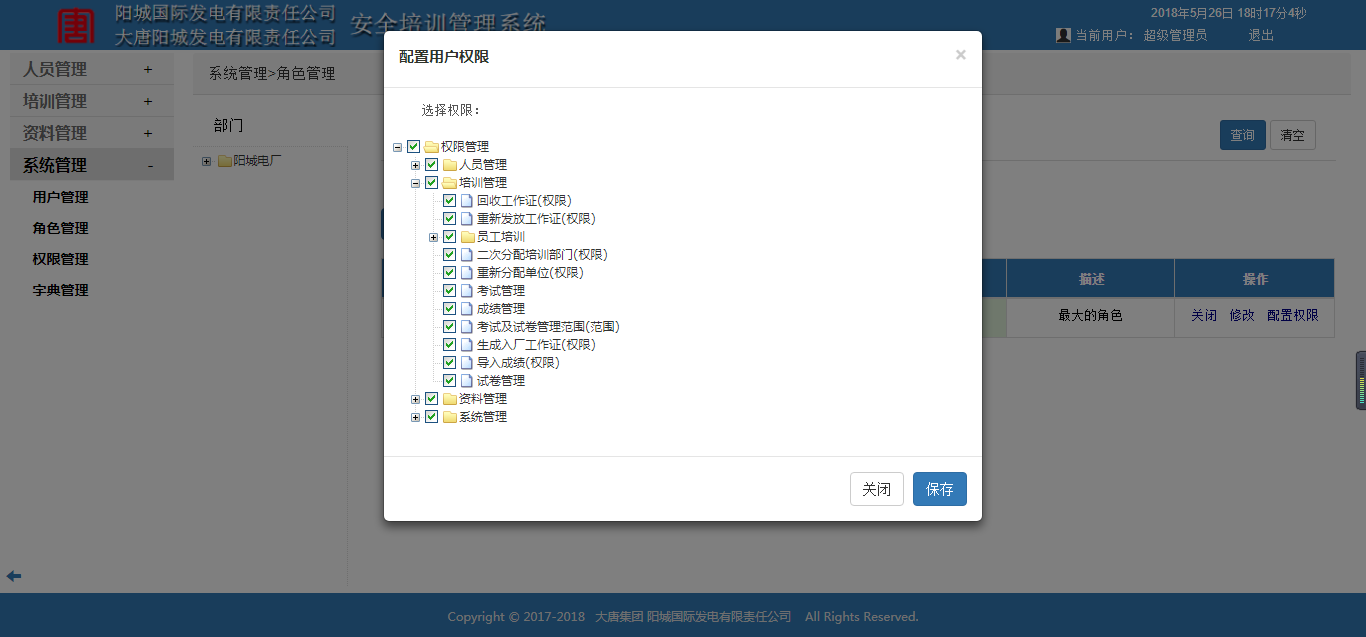


图6.6角色配置界面

#### 6.6.2 主要代码:

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*S 修改角色信息相关操作\*\*\*\*\*\*\*\*\*\*\*\*/  /\*\*  \* 修改角色信息  \* @param obj  \*/  function updateRole(obj) {  var tr = $(obj).parent().parent();//获取到tr元素  //1.初始化模态框的值  var roleId = tr.find(":checkbox").val();//角色ID  var roleName = tr.children("td:eq(2)").text();//角色名称  var roleStatus = tr.children("td:eq(3)").text();//角色状态  var roleDescription = tr.children("td:eq(5)").text();//角色描述  $("#update\_roleid").val(roleId);  $("#update\_rolename").val(roleName);  $("#update\_description").val(roleDescription);  //设置状态  $(".update\_rolestatus").removeProp("checked");  if("启用"==roleStatus){  $(".update\_rolestatus[value='1']").prop("checked","checked");  }else{  $(".update\_rolestatus[value='0']").prop("checked","checked");  }  layui.use('form',function () {  layui.form.render('radio');  })  //2.打开模态框  var width=($(window).width()\*0.80);  var height=($(window).height()\*0.70);  var index = layer.open({  title:'修改角色信息',  area: [width+'px', height +'px'],//大小  fix: true, //不固定  maxmin: true,  zIndex:500,  shadeClose: false,  shade:0.4,  type:1,  content:$('#updateRoleModal')  });  //向页面隐藏index  $("#hidden\_updateRole\_index").val(index);  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*E 修改角色信息相关操作\*\*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\*\*S 生成权限树相关操作\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\*\*  \* 查询权限树的函数  \*/  function getPermissionTree(){  $.ajax({  url : contextPath + '/permission/getPermissionTree4Role.do',  async : true,  dataType : 'json',  success : genePermissionTree,  error : function() {  alert("查询权限树失败！！！")  }  });  } |

### 6.7 字典管理界面以及代码

本节主要介绍字典管理的界面以及核心代码。

#### 6.7.1 添加字典主要界面

如图6.7所示：



图6.7 添加字典界面

#### 6.7.2 核心代码

核心JS

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*S 添加字典相关方法\*\*\*\*\*\*\*\*\*\*\*\*/  function openAddModal(){  layui.use(['layer','form'],function () {  /\*如果没有选择上级字典就提示先选择上级字典\*/  var layer = layui.layer,form = layui.form;  if($("[name='updictionaryid']").val()==''){  layer.alert("请先选择上级字典")  return;  }  //如果选择了上级字典就可以添加  var width=($(window).width()\*0.80);  var height=($(window).height()\*0.70);  var index = layer.open({  title:'添加字典',  area: [width+'px', height +'px'],//大小  fix: true, //不固定  maxmin: true,  zIndex:500,  shadeClose: false,  shade:0.4,  type:1,  content:$('#addModal')  });  //向页面隐藏index  $("#hidden\_add\_index").val(index);  })  }  //监听提交事件  layui.use(['layer','form'],function () {  var layer = layui.layer,form = layui.form;  form.on('submit(addDictionary)', function(data){  $.ajax({  url:contextPath+"/dictionary/addDictionary.do",  data:data.field,  type:'post',  datatype:'text',  success:function (response) {  layer.msg(response,{time:1500},function () {  if("添加成功"==response){  layer.close($("#hidden\_add\_index").val()); //关闭当前窗口  window.location.reload();//刷新页面  }  });  }  });  });  })  /\*\*\*\*\*\*\*\*\*E 添加字典相关方法\*\*\*\*\*\*\*\*\*\*\*\*/ |

## 第七章 系统测试与操作手册

本章主要介绍系统测试与系统的操作手册，从两方面完善系统。

### 7.1系统测试

本节主要按模块对系统进行功能测试，主要采用黑盒测试方法对系统的核心模块进行了测试。同时利用抓包工具对请求进行了分析。

#### 7.1.1试卷管理模块测试

本模块主要对试卷的生成与下载进行了测试。

1)试卷生成测试

生成试卷测试如图7.1所示:



图7.1 生成试卷测试结果图

利用抓包工具查看生成试卷向后台发送的数据结果如图7.2所示：

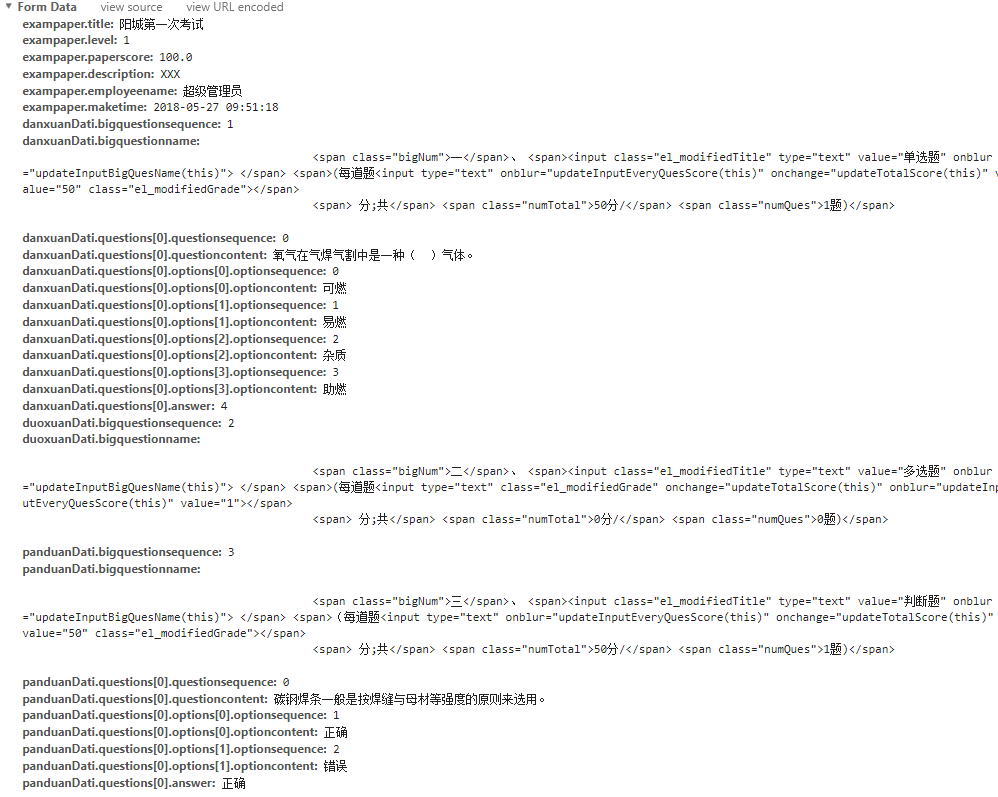


图7.2 生成试卷发送数据图

2)试卷下载测试

点击导出进行下载试卷的测试结果如图7.3所示:



图7.3 下载试卷测试结果图

利用抓包工具查看下载试卷向后台发送的请求如图7.4所示:



图7.4 下载试卷发送数据图

#### 7.1.2考试管理模块测试

主要对安排考试与查看考试参考人员进行测试。

1)安排考试的测试结果如图7.5所示:



图7.5 安排考试测试结果图

2)查看考试人员的测试结果如图7.6所示:



图7.6 查看参考人员结果图

#### 7.1.3 成绩管理模块测试

主要测试成绩分析与按部门统计成绩。

1)成绩分析的测试结果如图7.7所示



图7.7成绩分析测试结果图

2)按部门查看成绩的测试结果如图7.8所示:



图7.8 按部门查看成绩测试结果图

#### 7.1.4 用户管理模块测试

本节主要测试给用户分配角色。测试结果如图7.9所示:



图7.9 分配角色测试结果图

利用抓包工具查看发送的数据请求如图7.10所示:

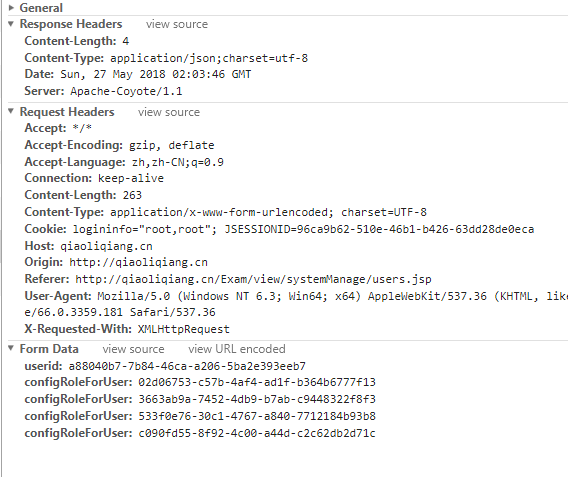


图7.10 分配角色发送请求图

#### 7.1.5 角色管理模块测试

本节主要测试了给角色配置权限。测试结果如图7.11所示:



图7.11 分配权限测试结果图

利用抓包工具查看请求结果如图7.12所示:

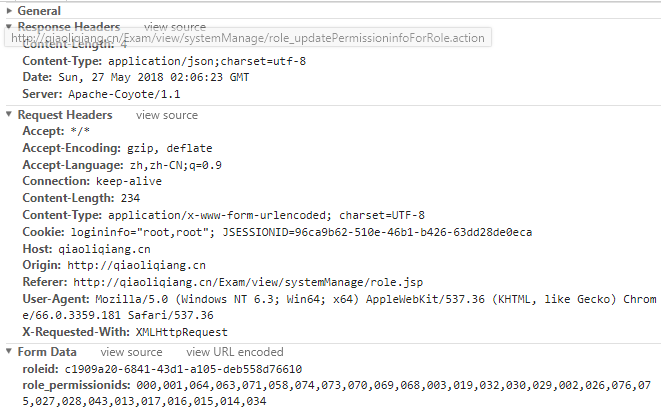


图7.12 分配权限请求结果图

#### 7.1.6 字典管理测试

本节测试了添加字典项。测试结果如图7.13所示:



图7.13 添加字典测试结果图

利用抓包工具查看请求结果如图7.14所示:



图7.14 添加字典请求结果图

### 7.2本章小结

本章主要对培训管理和系统管理两大模块进行测试，测试可以了解系统的不足，发现系统存在哪些问题，从而不断对系统进行改进。

## 结束语

本次毕业设计临近结束，在这个过程中，本人收获颇多。在毕业设计开始时，对毕业设计课题方面的知识一无所知，经过各种途径获取相关的资料后，开始渐渐了解。随后和团队队员一起相互协助，完成了从对系统的设计到系统的编码实现。

在整个设计与也开发、部署过程中，自己主要完成的工作是开发环境的搭建，完成开发环境搭建以及 Git 环境的搭建，数据库的设计，service 层接口的设计，以及模块的划分以及任务分工，同时负责项目的部署与安装以及 Tomcat 调优等工作。自己完成的模块是系统管理和培训管理。经过五个月的开发，加深了对 MVC 架构的理解，也加深了对三层架构的理解，同时也了解到数据库的设计原则，熟练了 SSM+Shiro 框架开发环境的搭建过程以及对工具类的封装。也熟练了 Encache 缓存的使用，Mybatis 与 Shiro 都与 Encache 缓存技术很好的集成在 一起。熟练了对 mysql 触发器、存储过程以及任务调度的使用，同时了解到数据库索引的重要性以及 sql 语句优化的方法。通过部署项目，学会了 Tomcat 优化 JVM 以及并发优化的配置，以及使用 Java 自带的性能分析工具监测 Tomcat 性能，同时学会了 Tomcat 日志的分析与查看以及利用 http\_load 对系统进行压力测试。了解到团队合作的重要性，同时了解到团队沟通的重要性以及文档书写能力的重要性以及源码管理的重要性。

在整个过程中将大学四年里从课本学到的知识综合起来灵活的应用到实际的开发工作中。总之，此次毕业设计让我受益匪浅。

## 致 谢

四年的大学生活接近尾声，四年来自己在科大遇到了很多的人和事。感谢学院老师这四年来对我的悉心栽培、教育以及陪伴我四年的同学们对我的热心的帮助。感谢那些一起学习、一起打球、一起做项目的同学，有了他们我的大学生活才更加的丰富多彩。

本论文是在谢斌红老师的细心指导下完成的。期间遇到许许多多的困难，谢老师总是耐心的为我分析各种难题并时常给出一些建议，总能让我从困难中获得灵感，找到解决问题的方法。开发过程中也遇到许多技术上的困难，谢老师总是能在第一时间给予指导性建议，团队成员也总是耐心的为我分析问题产生的原因，交给我快速解决问题的方法，谨在此表示由衷的感谢！本系统也是自己开发的第一个上线的系统，在项目部署调试过程中遇到了诸多的困难，感谢谢老师在第一时间给予帮助，本系统的操作手册也是在谢老师的指导下完成，再次感谢谢老师对这个系统的付出以及对我们这个开发团队的指导。

## 参考文献

1. 孙卫琴. Tomcat与Java Web开发技术详解.北京:ss电子工业出版社，2008.
2. 袁绍欣. Java面向对象程序设计.北京：清华大学出版社，2014.
3. 王晓敏,邝孔武. 信息系统分析与设计.北京：清华大学出版社，1998.
4. 张海藩. 软件工程导论.北京：清华大学出版社，2008.
5. 白尚旺,党伟超. 软件分析建模与PowerDesigner实现.北京：清华大学出版社，2010.
6. 成林. Bootstrap实战.北京：机械工业出版社，2013.
7. 李超. CSS网站布局实录.北京：科学出版社，2007.
8. 刘京华. Java Web整合开发王者归来.北京：清华大学出版社，2010.
9. 孟祥旭. 人机交互基础课程.北京：清华大学出版社，2010.
10. 秦航,杨强. 软件质量保证与测试.北京：清华大学出版社，2014.
11. 王珊,陈红. 数据库系统原理教程.北京：清华大学出版社，1998.
12. 王少锋. 面向对象技术UML教程.北京：清华大学出版社，2004.
13. 谢钧,谢希仁. 计算机网络教程.北京：人民邮电出版社，2014.
14. 谢郁. CSS高效开发实战.北京：电子工业出版社，2014.
15. 徐涛. 深入理解Bootstrap.北京：机械工业出版社，2014.
16. 张少军,刘增杰. PostgreSQL9从零开始学.北京：人民邮电出版社，2013.
17. 金敏,周翔. 高级软件开发过程.北京：清华大学出版社，2005.
18. 孙玉山,刘旭东. 软件设计模式与体系结构.北京：高等教育出版社，2013.
19. 李刚.疯狂HTML5/CSS3/JavaScript讲义.北京：电子工业出版社，2012.

### 附录Ⅰ 中英文翻译

**Web MVC with the Spring Framework**

Juergen Hoeller

**1. Introduction: Spring the Application Framework**

When first confronted with the Spring Framework, one might be tempted to think: "Oh no, not yet another web framework". This article will outline why Spring isn't particularly a web framework but a generic lightweight application framework with dedicated web support, and show the architectural differences to Struts and WebWork.

In contrast to Struts or WebWork, Spring is an application framework for all layers: It offers a bean configuration foundation, AOP support, a JDBC abstraction framework, abstract transaction support, etc. It is a very non-intrusive effort: Your application classes do not need to depend on any Spring classes if not necessary, and you can reuse every part on its own if you like to. From its very design, the framework encourages clean separation of tiers, most importantly web tier and business logic: e.g. the validation framework does not depend on web controllers. Major goals are reusability and testability: Unnecessary container or framework dependencies can be considered avoidable evils.

Of course, Spring's own web support is nicely integrated with the framework's general patterns. Nevertheless, replacing the web solution with Struts, WebWork, or the like is easy. Both with Spring's web support or a different one, Spring allows for building a true dedicated middle tier in the web container, with the option to reuse exactly the same business logic in test environments or standalone applications. And within J2EE, your business logic will not unnecessarily depend on container services like JTA or EJB - allowing complex, well-architected web applications to run in a "simple" container like Tomcat or Resin.

Note that Spring doesn't generally aim to compete with existing solutions. It rather fosters seamless integration with standards like Servlet, JSP, JTA, JNDI, JDBC, and JDO, and well-suited tools like Hibernate, Velocity, Log4J, and Caucho's Hessian/Burlap. The framework is designed to grow with the needs of your applications, in terms of technology choice: For example, you will probably use JTA via Spring's JtaTransactionManager if you need distributed transactions - but only then, as there are perfect replacements for single databases, like DataSourceTransactionManager or HibernateTransactionManager.

**2. Web MVC: The Design of Spring's Web Framework**

Spring's web framework is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mappings, view resolution, and locale and theme resolution. The default handler is a very simple Controller interface, just offering a "ModelAndView handleRequest(request,response)" method. This can already be used for application controllers, but you will prefer the included implementation hierarchy, consisting of AbstractController, AbstractCommandController, MultiActionController, SimpleFormController, AbstractWizardFormController. Application controllers will typically be subclasses of those. Note that you can choose an appropriate base class: If you don't have a form, you don't need a FormController. This is a major difference to Struts.

You can take any object as command or form object: There's no need to implement an interface or derive from a base class. Spring's data binding is highly flexible, e.g. it treats type mismatches as validation errors that can be evaluated by the application, not as system errors. So you don't need to duplicate your business objects' properties as Strings in your form objects, just to be able to handle invalid submissions, or to convert the Strings properly. Instead, it's often preferable to bind directly to your business objects. This is another major difference to Struts which is built around required base classes like Action and ActionForm - for every type of action. Compared to WebWork, Spring has more differentiated object roles: It supports the notion of a Controller, an optional command or form object, and a model that gets passed to the view. The model will normally include the command or form object but also arbitrary reference data. Instead, a WebWork Action combines all those roles into one single object. WebWork does allow you to use existing business objects as part of your form, but just by making them bean properties of the respective Action class. Finally, the same Action instance that handles the request gets used for evaluation and form population in the view. Thus, reference data needs to be modelled as bean properties of the Action too. These are arguably too many roles in one object.

Regarding views: Spring's view resolution is extremely flexible. A Controller implementation can even write a view directly to the response, returning null as ModelAndView. In the normal case, a ModelAndView instance consists of a view name and a model Map, containing bean names and corresponding objects (like a command or form, reference data, etc). View name resolution is highly configurable, either via bean names, via a properties file, or via your own ViewResolver implementation. The abstract model Map allows for complete abstraction of the view technology, without any hassle: Be it JSP, Velocity, or anything else - every renderer can be integrated directly. The model Map simply gets transformed into an appropriate format, like JSP request attributes or a Velocity template model.

**3. Integration: Using a Different Web Framework with Spring**

Many teams will try to leverage their investments in terms of know-how and tools, both for existing projects and for new ones. Concretely, there are not only a large number of books and tools for Struts but also a lot of developers that have experience with it. Thus, if you can live with Struts' architectural flaws, it can still be a viable choice for the web layer. The same applies to WebWork and other web frameworks.

If you don't want to use Spring's web MVC but intend to leverage other solutions that Spring offers, you can integrate the web framework of your choice with Spring easily. Simply start up a Spring root application context via its ContextLoaderListener, and access it via its ServletContext attribute (or Spring's respective helper method) from within a Struts or WebWork action. Note that there aren't any "plugins" involved, therefore no dedicated integration: From the view of the web layer, you'll simply use Spring as a library, with the root application context instance as entry point. All your registered beans and all of Spring's services can be at your fingertips even without Spring's web MVC. Spring doesn't compete with Struts or WebWork in this usage, it just addresses the many areas that the pure web frameworks don't, from bean configuration to data access and transaction handling. So you are able to enrich your application with a Spring middle tier and/or data access tier, even if you just want to use e.g. the transaction abstraction with JDBC or Hibernate.

**4. Feature Check List**

If just focussing on the web support, some of Spring's unique features are: .Clear separation of roles: controller vs validator vs command object vs form object vs model object, DispatcherServlet vs handler mapping vs view resolver, etc. .Powerful and straightforward configuration of both framework and application classes as JavaBeans, including easy in-between referencing via an application context, e.g. from web controllers to business objects and validators. .Adaptability, non-intrusiveness: Use whatever Controller subclass you need (plain, command, form, wizard, multi action, or a custom one) for a given scenario instead of deriving from Action/ActionForm for everything. .Reusable business code, no need for duplication: You can use existing business objects as command or form objects instead of mirroring them in special ActionForm subclasses. .Customizable binding and validation: type mismatches as application-level validation errors that keep the offending value, localized date and number binding, etc instead of String-only form objects with manual parsing and conversion to business objects. Customizable handler mapping, customizable view resolution: flexible model transfer via name/value Map, handler mapping and view resolution strategies from simple to sophisticated instead of one single way. Customizable locale and theme resolution, support for JSPs with and without Spring tag library, support for JSTL, support for Velocity without the need for extra bridges, etc. Simple but powerful tag library that avoids HTML generation at any cost, allowing for maximum flexibility in terms of markup code.

**Spring的web MVC 构架模式**

**Juergen Hoeller**

**1.介绍：Spring 应用构架**

当你第一次看到Spring的时候，你一定会说:"哦不，又一种web 构架".这篇文章将告诉你Spring明显区别于其他轻量级application framework, 它将专注于web的支持，与struts 和 webwork有着明显的区别。

在和struts 和 webwork的对比上,Spring是一个服务于所有层面的application framework:提供了bean的配置基础，AOP的支持，JDBC的提取框架，抽象事务支持，等等。它有一个非常显著的特点：在某个层面上如果你不需要Spring的支持，你就可以不使用String的class，只使用它的某一部分的功能。 从它的设计理念，你可以看到String 帮助你实现了真正的逻辑层和web层的分离：例如。一个校验应用将不用依靠controllers，就可以实现。这样的目标是 更好的重用和易测：过分依靠不必要的容器和框架将不能实现这一点。

当然,Spring 的自己的web支持和通常框架模式的细致完整.然而，Spring替换struts,webwork或者其他的web方案非常的容易.对于Spring的web支持或者不同的地方，Spring 允许你在web容器里面建立一个中间层，在测试环境或者标准独立的应用里面来设置重用你的商务逻辑.还有在J2EE环境里面，你的商务逻辑不必依靠容器提供的服务，像JTA ,EJB的支持.良好的构架的web应用可以运行在任何容器上，如，Tomcat 或者 Resin. 值得注意的是，Spring 不是和已经存在的解决方案进行竞争. 我们鼓励结合标准的技术，如， Servlet, JSP, JTA, JNDI, JDBC, and JDO, 和非常匹配的工具，如，Hibernate, Velocity, Log4J, and Caucho's Hessian/Burlap.这个框架的的设计思想是在你的应用需要改良的时候，你将会做一些技术的选择:例如,如果你需要分布式事务处理，你可能需要用Spring的Jta TransactionManager 来实现JTA服务.或者，用DataSourceTransactionManager 来实现美妙完美的单个数据库交换。

**2. Web MVC:Spring web 框架的设计思想**

Spring 框架通过配置操作mappings,展示resolution,本地化和模版集成围绕着分派请求操作的servlet - DispatcherServlet设计的.缺省的操作是一个非常简单的控制接口, 他只提供了ModelAndView handleRequest(request,

response)方法.这将用于一个应用的控制,但是,如果你想包含多个层次的控制，AbstractController, AbstractCommandController, MultiActionController, SimpleFormController, AbstractWizardFormController 将帮助你完成.应用控制将代表那些子系统.注意，你可以选择一个适当的基类:如果你没有 web form,你就不必用FormController.这就是和Struts最大的不同. 你可以通过命令或者form对象来操作任何对象:这不需要接口工具或者一个基础类的驱动.Spring的数据邦定是非常的灵活的.举例来说,它描述了具有在应用范围内的校验错误的输入机制,但不是系统错误.所以在你的form对象里面你不必复制你的业务对象的string属性,只操作出错的子任务,或者适当地转换string.换句话说,它通常可以很好的直接邦定你的业务对象. 这也是和struts围绕请求基础类Action 和 ActionForm (每个action操作类型) 建立主要的不同之一. 对比WebWork,Sping更多的区别在于对象角色:Sping支持控制器的感念,一个操作命令或者form对象,和得到数据传递给视图的模式.这个模式通常包含命令和form对象,但有时也包含任意的参考数据.换句话说,一个WebWork Action联合所有这些角色到一个单独的对象.WebWork 允许你用已经存在的业务对象作为你 form 的一部分,但是只生成各自Action 的 bean 属性. 最后,操作请求的Action 实例在一个视图里面获得付值和form population. 然而,参考数据也需要作为Action 的属性被模拟. 一个类里面有太多的角色是值得讨论的. 关于视图:Spring的视图方案非常的灵活. 一个控制器执行可以通过response 返回ModelAndView对象null,就可以直接写到一个视图.在通常的状况下,一个ModelAndView实例结合了 一个view 和一个 model Map,包含了bean name 和 通讯对象(像命令或者form,参考数据等等). View名称是非常高端的的配置，不是通过bean name, 一个properties 文件就是通过你自己的ViewResolver.这个抽象的model Map 允许你在视图层面完成提取,没有任何的争辩JSP/Velocity或者其他,每一种都可以直接完整使用.这个model Map 还可以简单得得到适当的格式化数据的转换,像JSP 请求属性或者Velocity 模版模式。

**3.集成:用Spring一个不同web 框架**

许多开发团队将为他们已有的项目或者新的项目已经获得的期限和工具进行投资.这里没有像Struts那样大量的图书和工具，但是同样我们有大量的拥有Spring开发技巧的开发人员.然而,如果你愿意生活在Struts的构架瑕疵中的话，他将是你在web层开发不错的选择.当然，其他应用也是一样. 如果你不想用Spring的 web MVC ，但是想借用Spring嫁接其他的解决方案,你可以非常简单地通过Spring 继承你自己的web 框架.你可以非常简单地通过ContextLoaderListener 启动一个Spring root application context, 并且,通过Struts 或者WebWork 的action 利用ServletContext 属性(或者Spring 的helper方法)存取它. 值得注意的是, 这里没有任何的 "plugins"被调用,因此没有专门的集成:来自web层的视图, 你可以简单的将Spring作为一个管理application context 实例入口点的类库. 所有你注册bean和Spring服务都可以在不需要Spring的web MVC的情况下都可以被你轻松掌握.Spring 不是和Struts ,WebWork这些应用进行竞争,它将作为一个纯web框架应用于 很多领域,因为它不需要配置数据存储和事务操作.所以,你可以利用Spring提供的中间层和数据存储层来丰富你的应用,甚至,你想用JDBC或者Hibernate进行事务抽象.

**4.特点核对列表**

如果聚焦于web 支持,Spring的一些显著特点是: 清楚地角色分离:controller,validator,command object,form object , model object,和 DispatcherServlet , handler mapping vs view resolver, 等等 强大而且直接的框架和作为JavaBeans的应用配置,包括简单的参照和应用内容,例如,从web控制器到业务对象和数据校验. 适应性,外挂:无论什么样的控制器你都需要得到代替Action/ActionForm所做的每件事情的方案(简单,命令,form,范例,多重action,或者定制一个) 重用业务逻辑代码,不需要复制:你可以用已经有的业务逻辑对象作为命令或则form对象代替反射特定的ActionForm子类. 可订制的邦定和数据校验:作为应用级的输入机制的错误校验,固定日期和数字邦定,例如,通过手动检查和转换业务对象来替换单个的string。 可订制的操作,可订制的视图方案:通过name/value Map灵活的模型传送,用一个简单的方法从简单到复杂操作映射和视图方案策略.