



太原科技大学
TAIYUAN UNIVERSITY OF SCIENCE & TECHNOLOGY

学士学位论文

题目：电厂安全培训管理系统

——人员及资料管理设计与实现

设计人：雷龙

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 142002

学 号：201420020208

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	雷龙	学号	201420020208
专业班级	软件工程 142002 班	同组人	乔利强
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——人员及资料管理系统设计与实现		
设计目的	通过设计开发一套安全培训管理系统，对电厂外来人员的安全培训进行更加有效的管理。系统需要实现的功能包括：内部部门管理，员工管理，短委员工管理以及检修项目管理，黑名单管理；题库管理和培训内容管理等。		
设计主要内容	1. 系统分析，对电厂安全培训管理系统进行需求分析，建立相关模型； 2. 总体设计，包括培训系统的功能模块设计以及电厂安全系统项目对应的系统架构和结构设计 3. 详细设计，包括数据库设计、人员及资料模块的界面设计、接口设计 4. 系统实现，人员及资料管理模块的编码实现。 5. 系统测试，包括单元测试和集成测试。		
设计要求	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. 撰写用户使用手册一套。		
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

目前整个电力行业的安全施工问题非常严峻，急需一套完备的安全培训管理系统来对外来入厂施工人员和厂内正式员工的安全培训进行科学有效的管理。此系统主要实现电厂题库资料管理、考试及试卷管理、短委员工管理、黑名单管理等。

本文按照软件工程的思想，首先对电厂安全培训管理系统进行了可行性和需求分析，并以此建立了需求模型、业务流程图和数据流图等，还对系统的功能性和非功能性需求进行了分析。然后对该系统进行了总体设计和详细设计，确定了 MVC 架构和 B/S 模式，并进行了数据库设计、接口设计和用户界面设计等。接着系统利用 Java 语言以及 Spring、Struts2、Mybatis 框架来进行整合开发。开发完成后对核心功能模块进行了详细的测试。

该系统上线后，真正意义上实现了对电厂人员信息的统一管理，方便公司领导管理决策。同时该系统将用户操作的权限进行了详细的划分，用户管理职责更加的清晰明确。

关键词：电厂安全培训；个人培训档案；黑名单；题库；Spring+Struts2+Mybatis

The Power Plant Safety Training Management System —Design and Implementation Of Personnel And Data Management

Athor: Lei Long Tutor: Xie Binghong

Abstract

At present, the safety construction problem in the entire power industry is very serious, and a complete set of safety training management system is urgently needed to scientifically and effectively manage the safety training for external construction workers and regular employees in the factory. This system mainly implements data management of power plant item database, examination and test paper management, short-term employee management, and blacklist management.

According to the idea of software engineering, this paper first analyzes the feasibility and needs of the power plant safety training management system, and establishes a demand model, business flow charts, and data flow charts, as well as the functional and non-functional requirements of the system. Analyzed. Then the system was designed and designed in detail. The MVC architecture and B/S model were determined, and database design, interface design and user interface design were performed. Then the system uses Java language and Spring, Struts2, Mybatis framework for integrated development. The core functional modules were tested in detail after the development was completed.

After the system was put on line, it truly realized the unified management of power plant personnel information and facilitated the company's leadership management decisions. At the same time, the system divides the authority of user operations in detail, and the user management responsibilities are more clearly defined.

Key words: Plant Safety Training; Personal Training Files; Blacklists; Question Bank; Spring+Struts2+Mybatis

目 录

第一章 系统概述..... 1

1.1 开发背景 1

1.2 开发意义 1

1.3 开发内容 2

1.4 推广前景 2

1.5 开发目标 3

1.6 论文结构 3

1.7 本章小结 3

第二章 开发技术..... 4

2.1 后端开发相关技术 4

2.1.1 Spring 介绍..... 5

2.1.2 Struts2 介绍 5

2.1.3 Mybatis 主要概述..... 6

2.1.4 Shiro 主要概述 6

2.1.5 Encache 缓存技术简介 7

2.1.6 其他相关技术简介 7

2.2 前端开发相关技术 8

2.2.1 Bootstrap 概述 8

2.2.2 JQuery 简介 8

2.2.3 EasyUI 简介..... 8

2.2.4 zTree 简介	9
2.2.5 Kindeditor 简介	9
2.3 项目管理工具	9
2.3.1 Maven 简介	9
2.3.2 Git 简介	10
2.4 数据库简介	11
2.5 本章小结	11
第三章 系统分析	12
3.1 可行性分析	12
3.2 需求分析	13
3.2.1 需求模型	13
3.2.2 业务流程模型	16
3.2.3 数据流图	19
3.2.4 系统用例分析	21
3.2.5 非功能性需求	24
3.3 本章小结	25
第四章 系统总体设计	27
4.1 系统功能模块图	27
4.2 系统架构设计	28
4.2.1 逻辑设计	29
4.2.2 三层架构	29
4.2.3 软件框架设计	30

4.3 系统结构设计	30
4.3.1 前端结构设计	30
4.3.2 后端结构设计	32
4.4 本章小结	34
第五章 系统详细设计	35
5.1 数据库设计	35
5.1.1 概念数据模型设计	35
5.1.2 物理数据模型设计	37
5.1.3 数据库视图设计	38
5.1.4 存储过程和事件计划设计	41
5.1.5 数据库表结构设计	42
5.2 接口设计	53
5.2.1 题库管理模块	53
5.2.2 试题管理模块	54
5.2.3 培训内容管理模块	55
5.2.4 内部部门管理模块	57
5.2.5 员工管理模块	59
5.2.6 检修项目管理	62
5.2.7 黑名单管理	65
5.3 人机对话设计	66
5.3.1 人机对话设计的原则	66
5.3.2 系统界面设计	67

5.4 本章小结	70
第六章 系统编码和实现	71
6.1 题库管理模块	71
6.1.1 题库管理功能实现界面	71
6.1.2 类结构图	72
6.1.3 关键代码	72
6.2 试题管理模块	73
6.2.1 试题管理功能实现界面	73
6.2.2 类结构图	74
6.2.3 关键代码	74
6.3 培训内容管理模块	76
6.3.1 培训内容管理功能实现界面	76
6.3.2 类结构图	77
6.3.3 关键代码	77
6.4 内部部门管理模块	78
6.4.1 内部部门管理功能实现界面	78
6.4.2 类结构图	79
6.4.3 关键代码	79
6.5 检修项目管理	80
6.5.1 检修项目管理功能实现界面	80
6.5.2 类结构图	81
6.5.3 关键代码	81

6.6 员工管理模块	82
6.6.1 员工管理功能实现界面	82
6.6.2 类结构图	83
6.6.3 关键代码	83
6.7 黑名单管理	84
6.7.1 黑名单管理功能实现界面	84
6.7.2 类结构图	85
6.7.3 关键代码	86
6.8 本章小结	86
第七章 系统测试	87
7.1 测试意义	87
7.2 测试环境	87
7.3 功能模块测试	87
7.3.1 题库统计功能测试	87
7.3.2 题库导出功能测试	88
7.3.3 员工培训档案查看导出功能测试	89
7.4 本章小结	90
结束语	91
致谢	92
参考文献	93
附录 I 中英文翻译	94
英文原文	94

中文翻译.....	97
-----------	----

第一章 系统概述

本章主要介绍电厂安全培训管理系统的开发背景、意义、开发内容以及项目推广前景，并对本论文结构安排做简要描述。

1.1 开发背景

火力发电厂生产是一个复杂的能量转换过程，期间存在很多的安全隐患。参照《企业职工伤亡事故分类》（GB6441-1986），火力发电厂至少存在物体打击、车辆伤害、容器爆炸等 13 类危险因素。因此，对新入厂的作业人员进行安全生产教育培训工作是非常重要的。

但是目前整个电力行业培训员工的方式仍旧十分的原始和落后，浪费大量的人力物力且对于外来单位、人员信息的无法做到统一集中管理，效率十分低下。因此，目前电力行业急需一套完备的安全培训管理系统来协助培训管理人员对入厂施工人员与厂内正式员工的安全培训进行管理。

1.2 开发意义

使用本系统后首先可以对外来单位信息和人员信息进行统一管理，同时可以将需要培训的视频上传到系统随时观看。考试培训试卷由系统生成，利用系统维护考试题库，更加的方便灵活。判卷评分交由判卷机完成，更加方便和高效。同时本系统配有违章管理的功能，可以对施工人员的违章情况及时统计查看，更好的管理施工人员，降低安全隐患。

1.3 开发内容

利用 Java 语言，以及 Spring+Struts2+Mybatis+Shiro 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。

经过需求调研和分析，确定该系统主要功能模块包括人员管理、检修管理、资料管理和系统管理。

1、人员管理主要实现各类员工信息的添加、删除、修改和查询等功能。为了减轻操作用户的负担，提供信息的精确性，人员信息需要利用身份证识别仪读取身份证信息并将其保存到系统中。此外，当员工违章时需将员工违章信息进行记录，还可以根据员工违章信息计算所在单位的违章总积分和加权积分。

2、检修管理主要实现对检修项目、参与项目的单位和员工信息进行统一有效的管理。

3、资料管理主要实现对系统资料信息的管理，如题库试题资料，培训视频资料等。

4、系统权限管理可以按照用户、角色和权限模式对系统中的所有功能包括界面中的每个按钮进行灵活分配和管理，确保每个人履行自己的职责。

1.4 推广前景

该项目一经推广，可在很大程度上提高电厂安全培训的工作效率，对被培训人员的要求也更加严格且效果更加显著。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，对于减少电厂安全操作隐患问题有显著效果。总之，电厂安全培训管理系统的前景十分广阔。

1.5 开发目标

本次开发的电厂安全培训管理系统主要目标就是为了解决单位信息和人员信息不能统一集中管理，考试与阅卷过程的繁琐，违章信息难于管理的问题，降低电厂安全隐患，为整个电力行业进行安全培训管理提供一个解决方案。

1.6 论文结构

本文共有七章，第一章围绕电厂安全培训从背景、意义和前景等方面展开描述。第二章主要介绍开发所使用的技术，包括系统前端开发技术与后端开发相关技术。第三章分析了系统的可行性并对其业务需求进行详细的分析和设计，做好准备工作。第四章则根据需求分析的结果为系统划分功能模块，并根据所划分的模块设计电厂培训系统的前后端结构。第五章从数据库设计，业务层接口设计以及用户界面设计三个方面对电厂安全培训系统进行详细设计，为系统编码实现打好基础。第六章着重介绍了人员管理和资料管理模块功能实现过程。第七章对已完成的功能模块进行测试，为系统上线做准备。

1.7 本章小结

本章首先分析了电厂安全培训系统的开发背景、开发意义、开发内容、开发目标及项目实施后的推广前景，紧接着介绍的本论文的目录结构安排。对电厂培训系统的开发有了明确的认识，为后期的开发奠定了基础。

第二章 开发技术

本系统采用 B/S 结构，前端主要是 HTML5，CSS3，JavaScript 等常用技术，后端开发语言 Java 语言，采用 SSM 框架进行开发。

本系统相关的开发环境如下：

- 1) 系统环境：Win 7。
- 2) 开发工具：Eclipse。
- 3) 后端框架：Spring3.2、Struts2.3、Mybatis3.2、Shiro1.3。
- 4) 前端相关：JQuery、Ajax、Bootstrap、EasyUI、zTree、KindEditor。
- 5) 数据库：Mysql5.5。
- 6) 服务器：Tomcat。
- 7) 源码管理工具：Git、Maven。
- 8) 建模工具：PowerDesigner12.6。
- 9) 测试工具：Junit。
- 10) JDK：JDK1.7。

本系统所需配套的硬件如下：

- 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生成 Excel 成绩单导入系统。

2.1 后端开发相关技术

本系统后端开发使用的核心技术是 Spring，Struts2，Mybatis 以及 Shiro，缓存技术主要是 Encache，也用到一些 POI、Jsoup 等相关技术。本节主要介绍电厂安全培训管理系统用到的核心技术的原理以及在系统中的具体应用。

2.1.1 Spring 介绍

Spring 是一个开源的轻量级（直接使用不依赖其他东西）的框架，它的主要思想是面向切面编程（扩展功能不是修改源代码实现）和控制反转（对象的创建不是通过 new 方式实现，而是交给 Spring 配置创建类对象）。它主要由七个模块组成，具体模块结构图如图 2.1 所示。

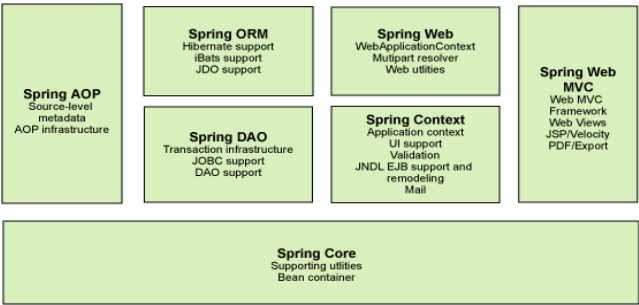


图 2.1 Spring 核心模块结构图

2.1.2 Struts2 介绍

Struts2 框架是一个基于 MVC 设计模式的 Web 应用框架，它对应的是 JavaEE 三层结构控制层的框架，Struts2 框架是在 Struts1 和 Webwork 基础之上发展的全新的框架，使用它可以很便捷的进行前台页面与服务器端的数据交互，让代码编写人员能够专注于系统业务逻辑。其工作原理如图 2.2 所示：

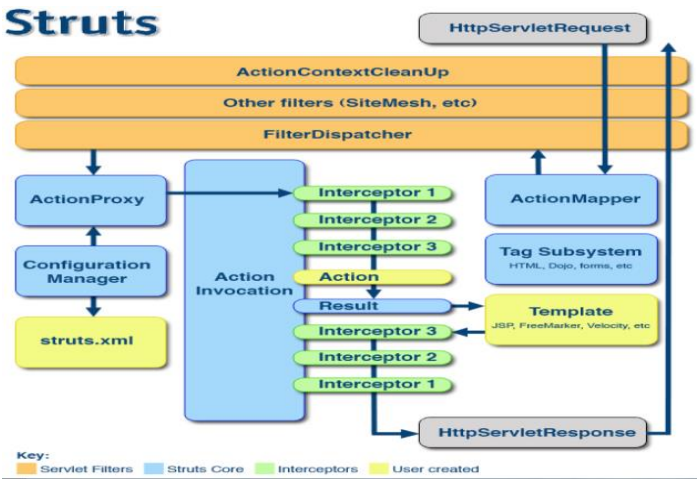


图 2.2 Struts2 工作原理图

2.1.3 Mybatis 主要概述

Mybatis 是一个数据持久层框架,它让程序员将主要精力放在 SQL 和业务逻辑上,通过 Mybatis 提供的映射方式,可自由灵活生成(半自动化,大部分需要程序员编写 SQL)满足业务需要 SQL 语句。Mybatis 框架的架构如图 2.3 所示:

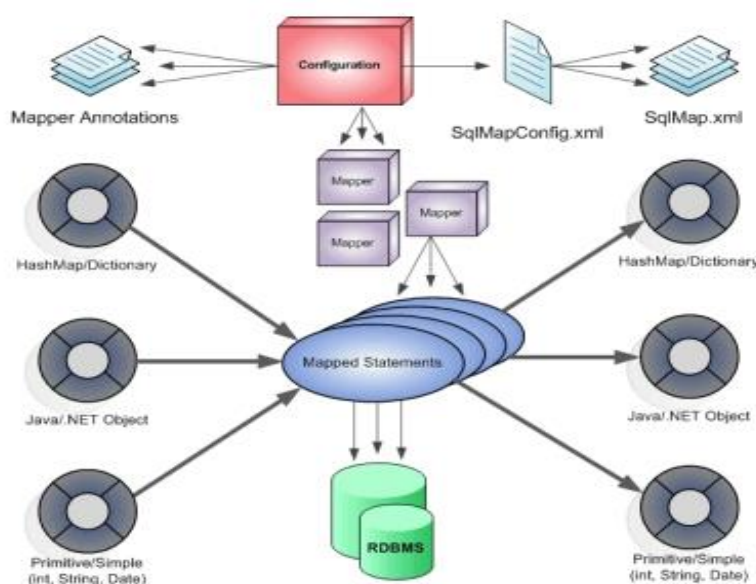


图 2.3 Mybatis 框架架构图

2.1.4 Shiro 主要概述

Shiro 是 Apache 下一个开源的权限管理的框架,常用来实现用户认证、用户授权等系统权限操作。它不依赖于 Spring,不仅可以实现 Web 应用的权限管理,还可以实现 C/S 系统、分布式等系统的权限管理,因此很受企业项目的欢迎。

在电厂安全培训管理系统中就使用 Shiro 框架来实现该系统的用户权限管理,简化了许多开发的操作,大大降低了开发成本,为系统开发节省了不少时间。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 能够很好的支持缓存以及分布式 session,对电厂培训系统的性能也有很大提高。其架构如图 2.4 所示:

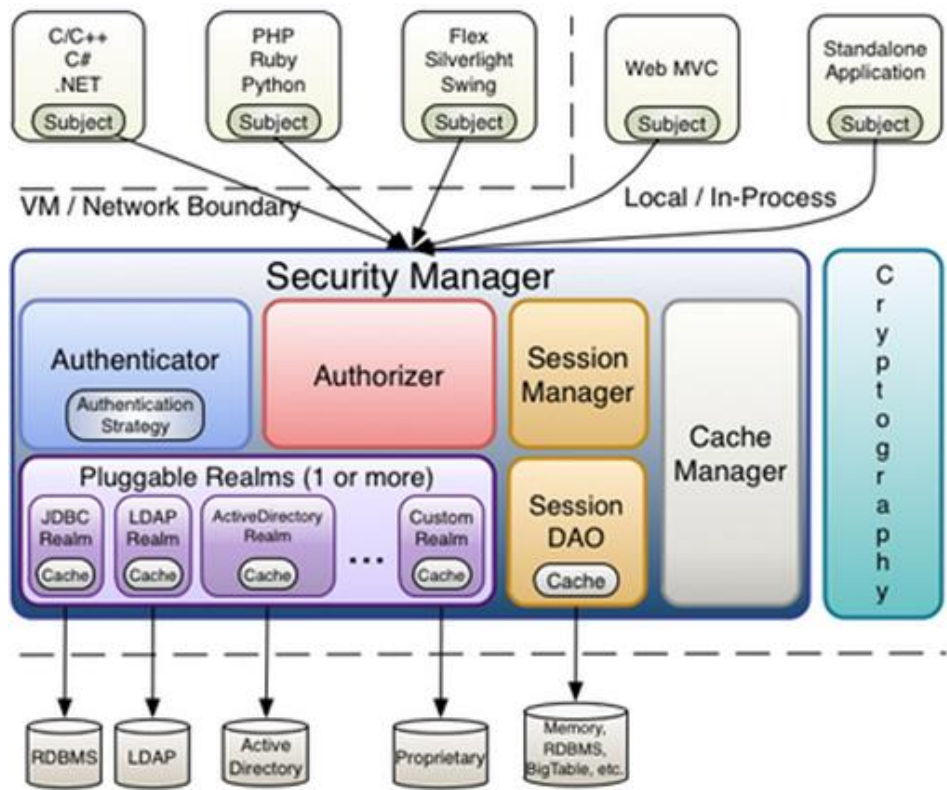


图 2.4 Shiro 架构图

2.1.5 Encache 缓存技术简介

Ehcache 是一种在开发行业广泛使用的开源 Java 分布式缓存框架。主要面向通用缓存, Java EE 和轻量级容器。

在本电厂安全培训管理系统中,其主要应用了 Mybatis 的二级缓存和 Shiro 的缓存来对一些常用查询如工种信息, 知识点信息和用户角色权限信息等进行缓存处理, 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是一个开放源码函式库,它提供了一系列的方法供开发人员进行 Office 办公软件的相关操作。在本系统中主要用来操作 Excel 文件, 如批量导入题库试题, 导出个人培训档案。

Jsoup 是一个用于在系统后台处理 HTML 标签的 Java 库。它提供了一个非常方便的 API 来提取和操作数据，它有时也可以用来进行 Java 爬虫项目的开发。本系统中主要用在后台处理试题编辑框中的相关标签。

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 是基于 HTML、CSS、JavaScript 的开源工具包，它的特点是简洁灵活，可以加快界面开发的速度，节省系统开发中前台界面绘制的时间。大致包含 Bootstrap 基本结构、Bootstrap CSS、Bootstrap 布局组件和 Bootstrap 插件几个部分。本系统的前台功能模块界面就是使用它来进行搭建的。

2.2.2 JQuery 简介

JQuery 是 JavaScript 的轻量级的库。JQuery 是一个 JavaScript 框架，它兼容 css3，还兼容各种浏览器。与传统的 JavaScript 相比有以下优势：

- 1) JS 传统方式页面加载会存在覆盖问题整个页面加载完毕(包括里面的其他内容，比如图片等才开始执行) JQuery 不存在覆盖问题，加载的时候是顺序执行的。
- 2) JQuery 的加载比 JS 加载更快（当整个 DOM 树结构绘制完毕就会加载）

2.2.3 EasyUI 简介

EasyUI 是一种基于 JQuery 的用户界面插件库，里面有丰富的前台功能插件，使用它可以极大的减少前台页面开发的工作量，在本系统中主要使用了 EasyUI 的分页插件。

2.2.4 zTree 简介

zTree 是一个依靠 JQuery 实现的多功能“树插件”。具有性能优异、配置灵活、功能丰富等优点。在官方网站提供了详细的 API 开发手册和许多小巧精炼的 Demo 供使用者参考，易于学习和使用。本系统所有有关树结构的操作都使用的是这个插件，如部门树，检修单位树，权限树等。

2.2.5 Kindeditor 简介

KindEditor 是一套开源的 HTML 可视化编辑器，支持常见的编辑操作，兼容 IE、火狐、谷歌等主流浏览器。KindEditor 使用 JavaScript 编写，继承 JS 的所有特性，可以和 Java、.NET 等程序完美接合。在本系统中主要使用 KindEditor 插件来进行试题编辑修改功能的实现。

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 是一个项目管理和整合工具。它是用纯 Java 进行开发的，并且只是用来管理 Java 项目。它有以下特点：

- 1) 实现了对项目中的 jar 包的统一管理，可以节省空间
- 2) 一个命令就可以将源码运行起来 (tomcat:run)
- 3) 有 Java 跨平台的特性
- 4) 应用于大型项目分模块开发，可以提高开发效率

Maven 的核心概念图如图 2.5 所示。

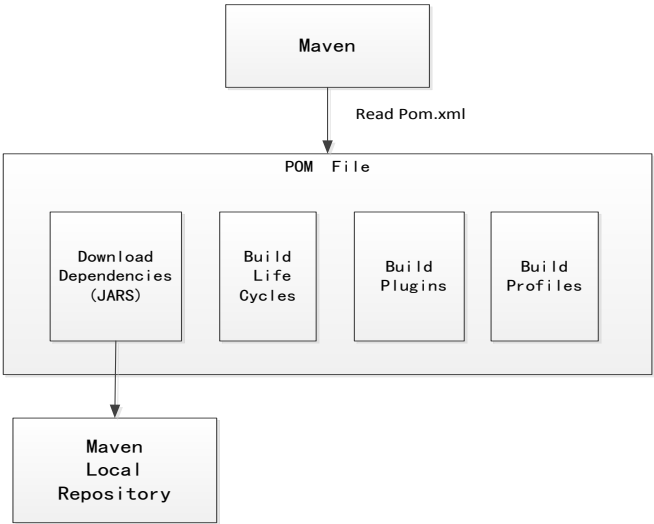


图 2.5 Maven 核心概念图

2.3.2 Git 简介

Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件，目前被广泛用于项目开发的版本控制。

本系统采用 Git 客户端进行项目版本的管理，它相对于 Eclipse 中的 Git 插件有不受限于开发环境，可与任意开发环境紧密结合，进行项目版本的控制的特点。

Git 的工作流程如图 2.6 所示：

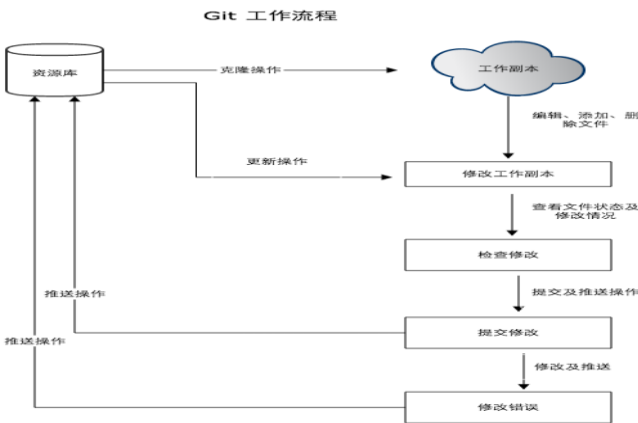


图 2.6 Git 工作流程图

2.4 数据库简介

MySQL 是一种关系数据库管理系统，它所使用的 SQL 语言是用于访问数据库的最常用标准化语言。关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

相比较 Oracle 数据库而言，Mysql 数据库更加的小巧灵活，学习成本很低，易于操作，配合图形化工具 Navicat 使用十分的便捷，对开发人员技能要求很低，从另一方面来讲也提高了开发效率。

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，主要包括后端开发技术介绍、前端开发技术介绍以及在电厂系统中的具体应用场景。还介绍了系统项目的管理工具和开发所使用的数据库，基本上确定了该电厂安全培训管理系统的技术选型。

第三章 系统分析

系统分析的任务是确定系统应该“做什么”。系统分析的全过程可以划分为三个阶段：可行性分析阶段、需求分析阶段以及需求定义阶段。

3.1 可行性分析

从电厂安全培训管理系统的实际情况出发，本小节将从经济、技术、时间、操作以及社会等方面来对电厂安全培训系统进行全面分析，判断该系统开发是否可行。

1) 经济可行性

采用电厂安全培训管理系统在很大程度上提高了电厂培训管理的工作效率和管理水平，极大的减轻了各级安全培训员的工作量。同时还实现管理方式的网络化、数字化和规范化，总体来说，开发这个系统利大于弊。

2) 技术可行性

小组成员具有一定的 Web 项目开发经验，有多个完整系统开发经验，基础扎实，无论是对信息系统开发方法的理解还是对于技术的熟练操作程度都没有任何问题。

3) 时间可行性

系统开发周期大概需要 90 多天的时间。15 天用于系统的需求分析，15 天用于数据库和系统架构设计，一个多月的时间进行编程和功能的实现，剩下的时间用于测试和维护。

4) 操作可行性

功能开发完毕后为系统专门配有相对应的用户开发手册，并在系统试用期内对使用者进行相关的培训，确保用户能够熟练的使用该系统进行培训管理。

5) 社会可行性

火力发电厂生产是一个复杂的能量转换过程，期间存在很多的安全隐患。参照《企业职工伤亡事故分类》（GB6441-1986），火力发电厂至少存在物体打击、车辆伤害、

容器爆炸等 13 类危险因素。因此，对新入厂的作业人员进行安全生产教育培训工作是非常重要的，十分需要这样一套安全培训管理系统来帮助电厂降低安全隐患和风险。

6) 总结

通过经济、技术、时间以及操作等方面的可行性分析，可以确定电厂安全培训系统的开发是十分必要的，而且对电厂行业意义十分重大，建议立即展开系统相关的开发工作。

3.2 需求分析

需求分析应准确地回答“系统必须做什么”。需求分析需要对目标系统提出完整、准确、清晰、具体的要求。

3.2.1 需求模型

需求模型（Requirement Model, RQM）描述系统需要完成的任务，通过和用户的需求沟通得到本系统的需求模型。系统的总需求文档模型如图 3.1 所示

Title ID	Full Description	Code	Priority	Workload	Risk	Status	
1.	人员管理 (包括内部部门、外部部门、内部员工、外部员工管理)	REQ_ 1	5		High	Draft	
2.	培训管理 (包括考试、成绩、试卷管理)	REQ_ 2	5		High	Draft	
3.	资料管理 (包括题库管理，试题管理与视频管理)	REQ_ 3	2.5		Medium	Draft	
4.	系统管理 (包括用户、角色、权限、字典管理)	REQ_ 4	1		Low	Draft	

图 3.1 系统总需求文档模型

部门管理需求文档模型如图 3.2 所示，主要包括部门基础信息的维护，部门违章积分的统计以及加权积分计算（部门总人数与总违章积分的比值）以及所有二级部门和长委单位的人数统计和多条件查询。

	→	1.1.1	部门管理 (对部门信息进行管理，对部门违章积分按年度统计)	REQ_ 17	5	0	High
4		1.1.1.1	部门基础信息维护 对电厂内部部门和长委单位的基础信息进行增删改查操作	REQ_ 25	5		High
5		1.1.1.2	部门违章积分统计 统计内部部门和长委单位员工的违章积分和加权积分	REQ_ 26	5		High
6		1.1.1.3	部门人数统计 对所有二级部门和长委单位的人数进行统计并分页显示方便公司领导查看	REQ_ 27	5		High
7		1.1.1.4	多条件查询 根据部门名称、违章时间和单位类型等部门进行查询	REQ_ 28	5		High

图 3.2 部门管理需求文档模型

员工管理需求文档模型如图 3.3 所示，主要包括员工基础信息的维护，通过身份识别仪接口读取身份证信息，个人培训档案的查看和导出，个人违章信息的增删改查操作。

	→	1.1.2	员工管理 (对员工信息进行管理，对员工违章积分按年度统计)	REQ_ 18	5	0	
9		1.1.2.1	员工基础信息维护 对电厂内部员工和长委单位的员工基础信息进行增删改查操作	REQ_ 29	5		
10		1.1.2.2	个人培训档案管理 可以对电厂员工的培训档案进行查看和导出的操作	REQ_ 30	5		
11		1.1.2.3	个人违章信息管理 可以对个人违章信息进行增删改查操作	REQ_ 31	5		

图 3.3 员工管理需求文档模型

检修项目管理需求文档模型如图 3.4 所示，主要包括检修基础信息的维护，检修标段维护统计，短委单位的增删改查，已发放工作证人数统计分析。

14	→	1.2.1	检修项目管理 (对大修、小修以及检修单位进行管理)	REQ_ 20	5	0	
15		1.2.1.1	检修基础信息维护 对电厂的检修基础信息进行增删改查操作	REQ_ 32	5		
16		1.2.1.2	检修标段管理 对检修中的标段信息进行增删改查和统计	REQ_ 33	5		
17		1.2.1.3	短委单位管理 对检修项目中的单位基础信息进行增删改查操作	REQ_ 34	5		
18		1.2.1.4	人数统计 对已发放工作证的短委员工进行统计分析	REQ_ 35	5		

图 3.4 检修项目管理需求文档模型

黑名单管理需求文档模型如图 3.5 所示，主要包括黑名单单位基础信息的维护以及黑名单员工违章信息的查看和黑名单记录的删除操作。

20	1.2.3	黑名单管理 (对部门与员工黑名单进行增删改查)	REQ_ 22	5
21	1.2.3.1	黑名单单位管理 对黑名单单位信息进行增删改查操作	REQ_ 36	5
22	1.2.3.2	黑名单员工管理 黑名单员工的违章信息的查看和黑名单记录的删除操作	REQ_ 37	5

图 3.5 黑名单管理需求文档模型

题库管理需求文档模型如图 3.6 所示，主要包括题库的基础信息维护，试题的批量导入，试题批量导入模版下载，题库试题统计分析，题库导出。

31	3.1	题库管理 (对电厂的培训题库进行管理)	REQ_ 23	2.5
32	3.1.1	题库基础信息维护 对电厂题库基础信息进行增删改查操作	REQ_ 38	2.5
33	3.1.2	题库试题统计 对题库中的试题按照题型进行统计分析	REQ_ 39	2.5
34	3.1.3	试题批量操作 对题库中的试题进行批量导入和导出的操作	REQ_ 40	2.5
35	3.1.4	试题导入模版管理 可以下载试题批量导入模版准备题库试题进行导入操作	REQ_ 41	2.5

图 3.6 题库管理需求文档模型

试题管理需求文档模型如图 3.7 所示，主要包括试题的基础信息维护，试题的详情预览，试题的批量删除和批量移动题库试题。

36	3.2	试题管理 (主要是对题库中的试题进行管理)	REQ_ 24	2.5
37	3.2.1	试题基础信息管理 对单个试题进行增删改查操作	REQ_ 43	2.5
38	3.2.2	试题详情查看 可以查看某个试题的题干、选项和答案	REQ_ 44	2.5
39	3.2.3	批量管理 可以实现试题的批量删除和批量移动题库试题的操作	REQ_ 45	2.5

图 3.7 试题管理需求文档模型

培训内容管理需求文档模型如图 3.8 所示，主要包括培训类别树的维护，视频资料的基础信息维护，视频的播放，以及批量删除操作。

40	3.3	培训内容管理 (主要是对培训视频进行线上管理)	REQ_ 42	2.5
41	3.3.1	培训资料基础信息维护 可以实现对培训资料基本信息的增删改查	REQ_ 46	2.5
42	3.3.2	资料类别树管理 对培训资料的类别树进行增删改查操作	REQ_ 47	2.5
43	3.3.3	批量管理 可以实现对培训资料进行批量删除的操作	REQ_ 48	2.5

图 3.8 培训内容管理需求文档模型

3.2.2 业务流程模型

业务流程模型（Business Process Model，BPM）是专为描述业务流程建立的模型，特点是使业务用户更容易理解系统的相关功能。它采用图形化的方法描述软件系统的业务流程，对软件系统的分析、设计、开发、维护和优化具有重要的作用。

该系统的业务流程根据实际情况和用户需要可以大致分成四大部分，分别是人员管理业务流程包括内部员工和短委员工的相关业务；培训管理业务流程包括试卷、考试、成绩、员工培训安排的相关业务；资料管理业务流程包括题库、试题以及培训内容（视频资料）的相关业务；最后是系统管理业务流程包括用户、角色、权限和字典的相关业务。

下面主要介绍电厂安全培训系统中人员管理和资料管理的相关业务流程图。

1) 内部员工管理的业务流程图

电厂安全培训管理系统的内部员工管理主要包括：部门管理中内部单位和长委单位两种单位的基本信息的增删改查，以及所有二级部门和长委单位的总人数和违章积分信息的统计，员工管理中通过选择部门树中的一个部门进行员工的添加操作，可以对内部和长委员工和基本信息和违章信息进行增删改查，员工个人培训档案的查看和导出，按照年龄段和违章时间等条件对员工信息进行统计查询。

内部员工管理的业务流程图如图 3.9 所示：

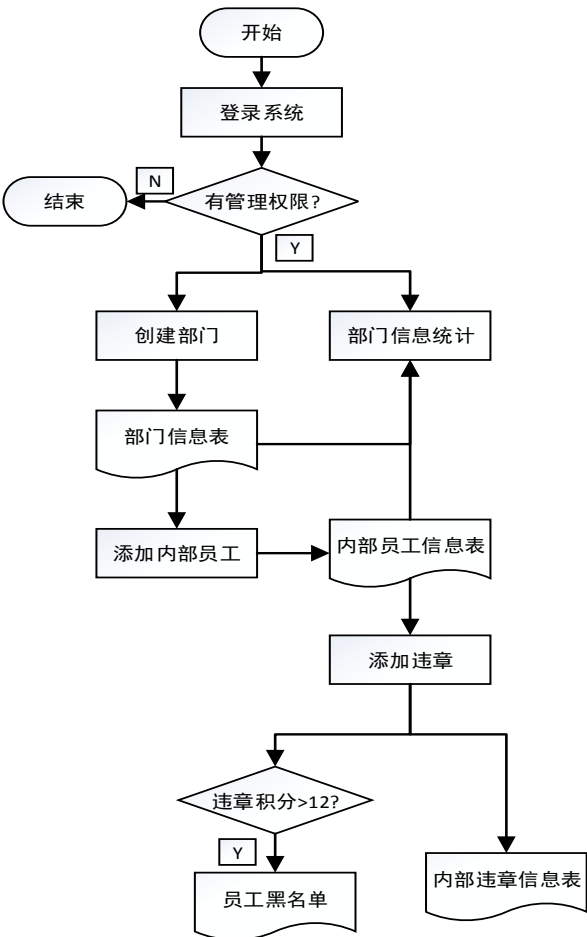


图 3.9 内部员工管理业务流程图

2) 短委员工管理的业务流程图

短委员工管理主要包括：检修项目管理中检修基础信息的维护，检修标段信息的维护统计，每个检修项目中短委单位基本信息增删改查和单位违章积分和人数的统计，已发放工作证人数统计分析。短委员工管理中检修单位树的查询显示，选择检修单位树中的一个单位进行员工的添加操作，可以对员工基本信息和违章信息进行增删改查，员工个人培训档案的查询。黑名单管理中黑名单单位基础信息的维护以及拉黑人员违章信息的查看和黑名单记录的删除操作。

短委员工管理的业务流程图如图 3.10 所示：

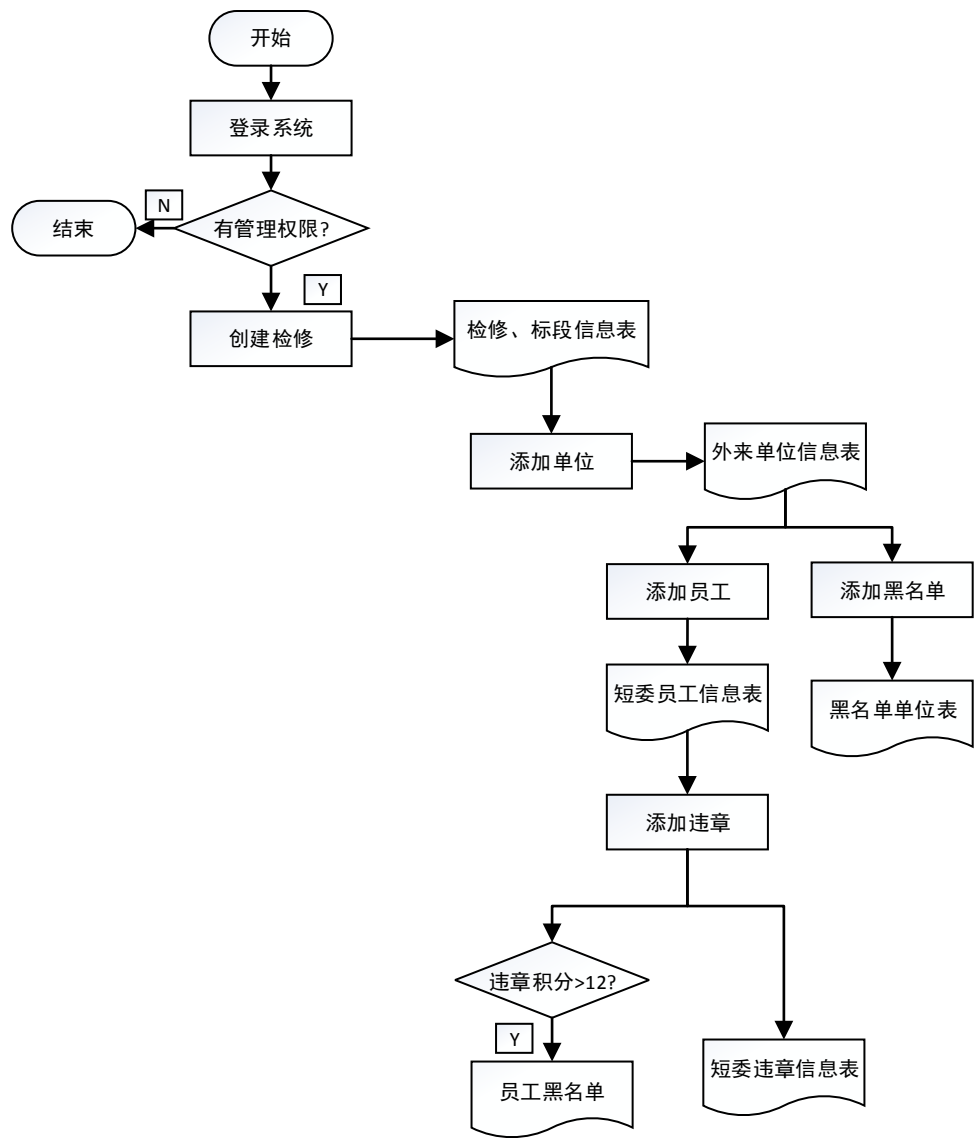


图 3.10 短委员工管理业务流程图

3) 资料管理的业务流程图

资料管理主要是对安全培训系统中的相关资料信息进行统一管理，主要包括题库试题信息以及培训视频资料信息。对于试题而言，只有创建了题库才能进行试题的相关操作，当题库删除时，其对应的试题也需删除。培训视频资料有所属的资料类别，只有先选择一个培训类别才能进行视频的相关操作，类别不存在时对应的视频也全部删除。

资料管理的业务流程图如图 3.11 所示：

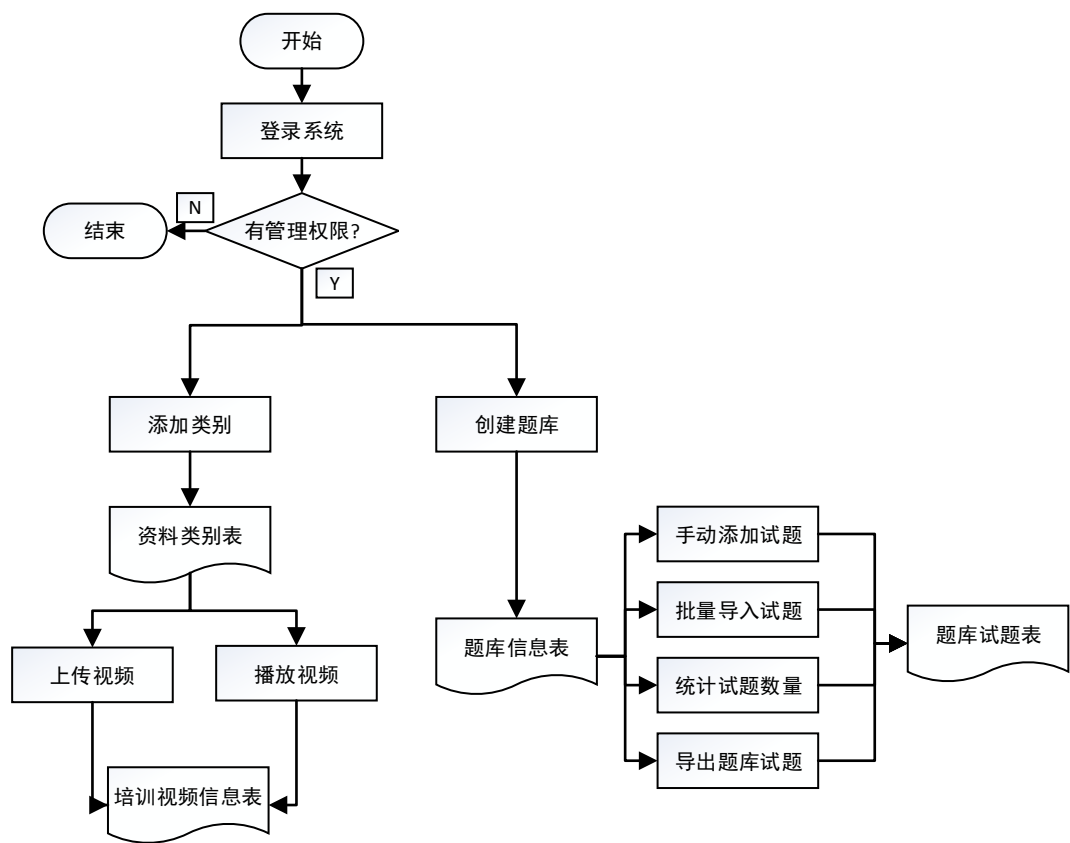


图 3.11 资料管理业务流程图

3.2.3 数据流图

数据流图描述数据流动、存储、处理的逻辑关系。主要用到四个基本符号，即外部实体，数据处理、数据流和数据存储。绘制数据流图可以帮助设计人员分析系统的主要的操作者，可以判断不同的用户在系统中扮演的角色，关注的功能等，还可以帮助设计人员检查系统设计逻辑是否准确合理，数据源是否有遗漏等。

设计良好的数据流图可以清晰的向用户和开发人员展示系统的功能，以及系统中数据的交互信息，及早的发现系统设计中存在的问题，从而降低系统开发中存在的风险。

电厂安全培训管理系统的顶层数据流图如图 3.12 所示

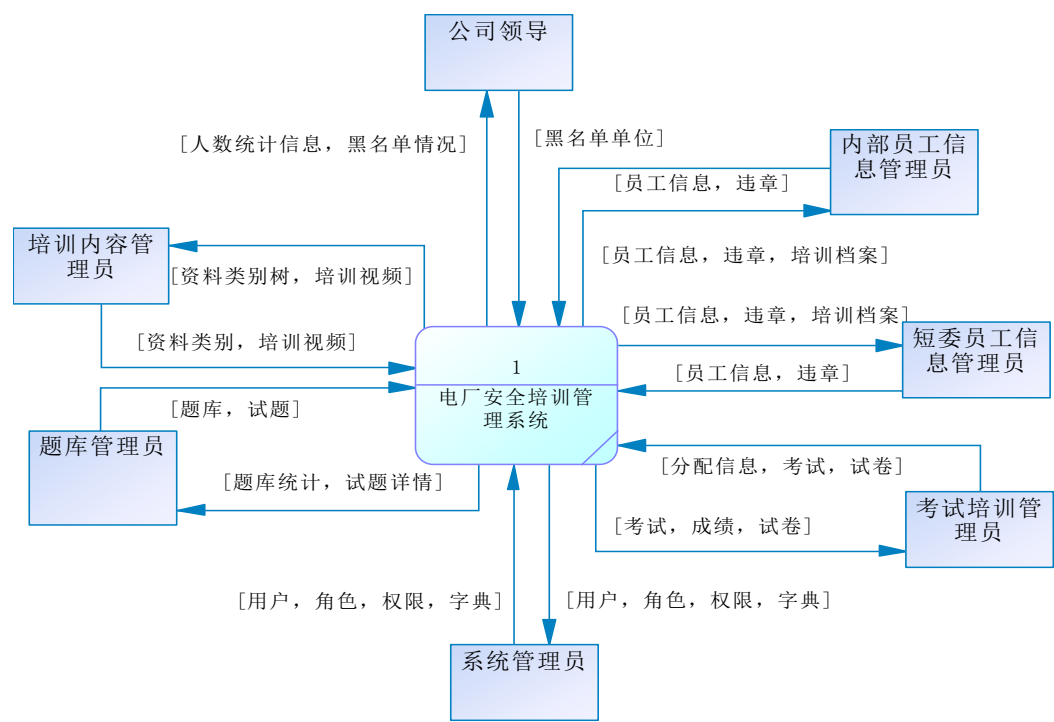


图 3.12 顶层数据流图

为了表达数据处理过程中数据的加工情况，用一个数据流图往往是不够的。稍微复杂的实际问题，在数据流图上常常出现十几个甚至几十个处理，这样的数据流图看起来很不清楚。层次结构的数据流图能很好地解决这个问题。按照系统的层次结构进行逐步分解，并以分层的数据流图反映这种关系，能清楚地表达整个系统。

根据系统顶层数据流图和需求分析设计电厂安全培训管理系统的一层数据流图如图 3.13 所示。

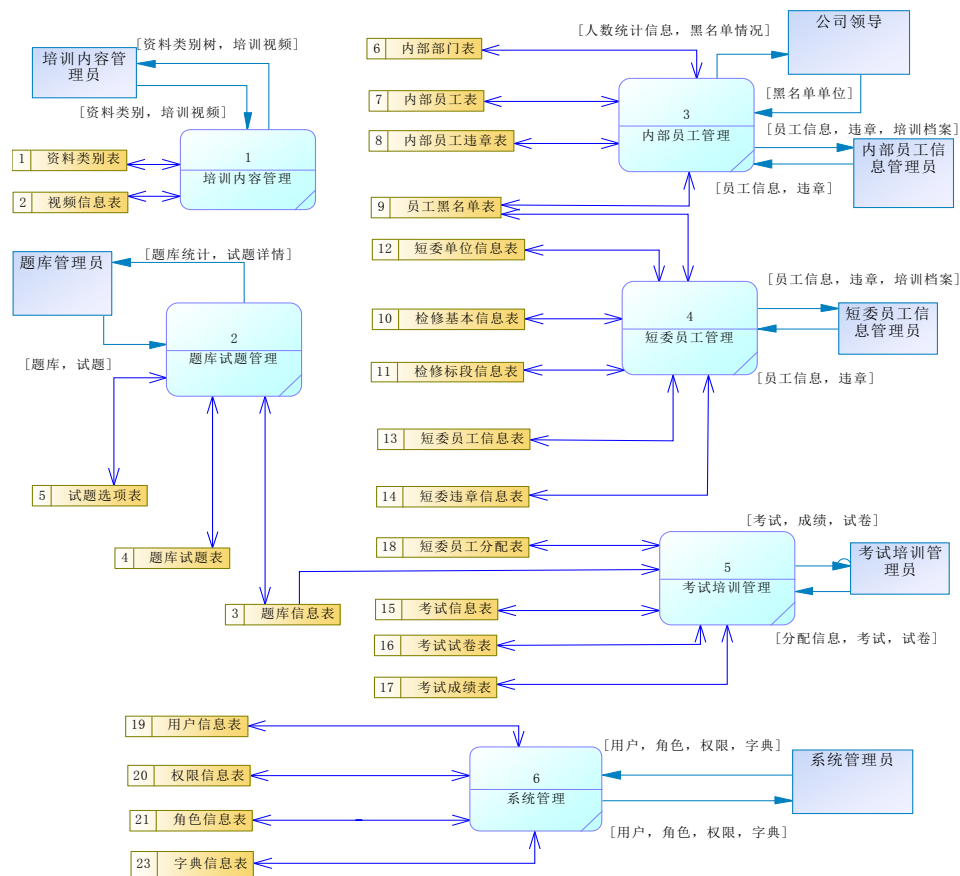


图 3.13 一层数据流图

3.2.4 系统用例分析

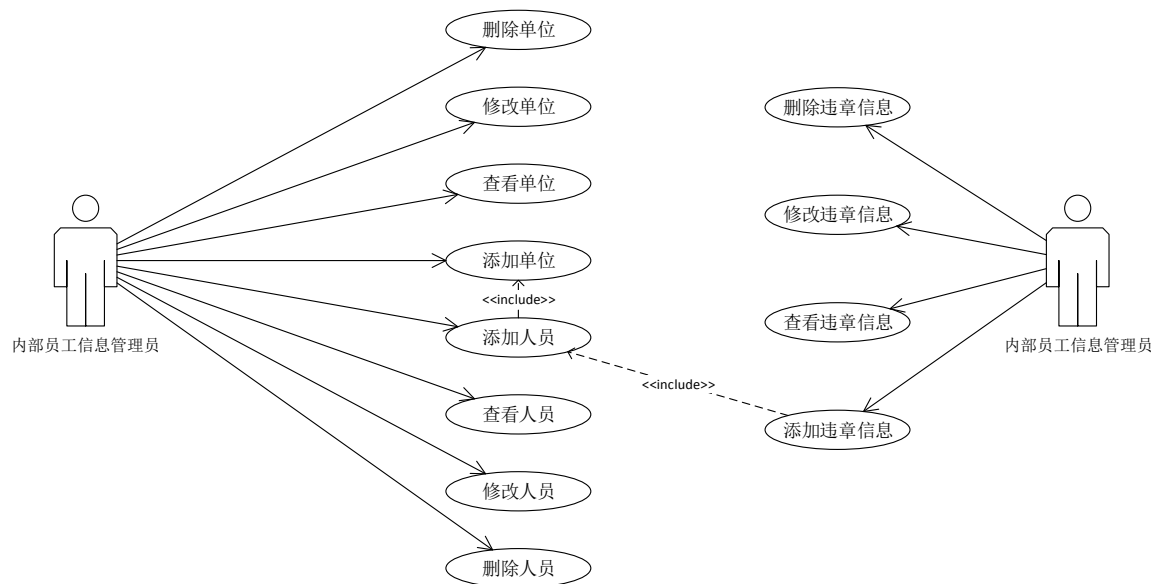


图 3.14 内部员工信息管理员用例图

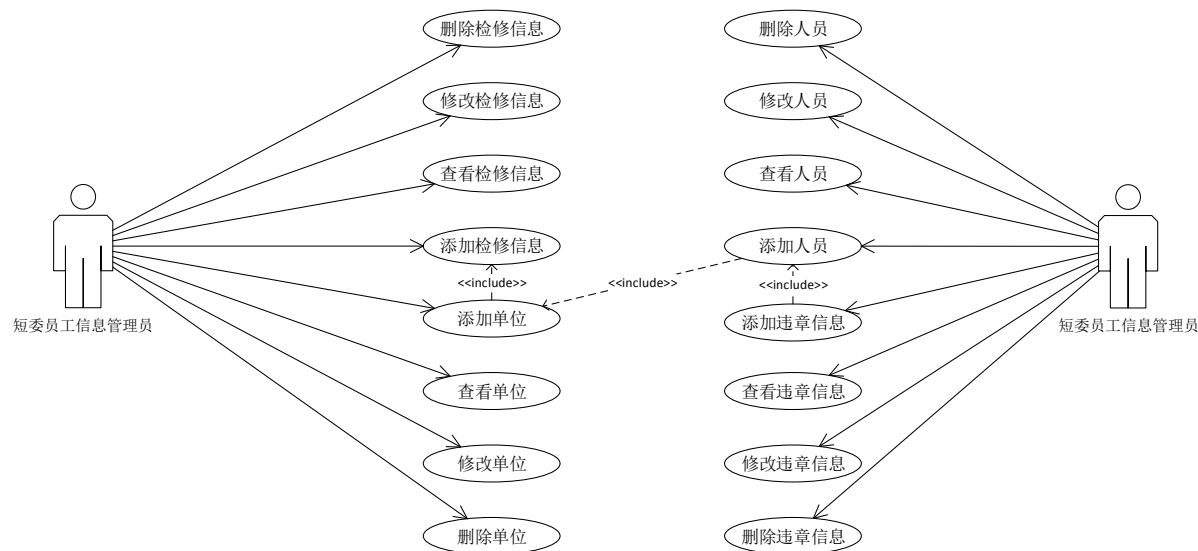


图 3.15 短委员工信息管理员用例图

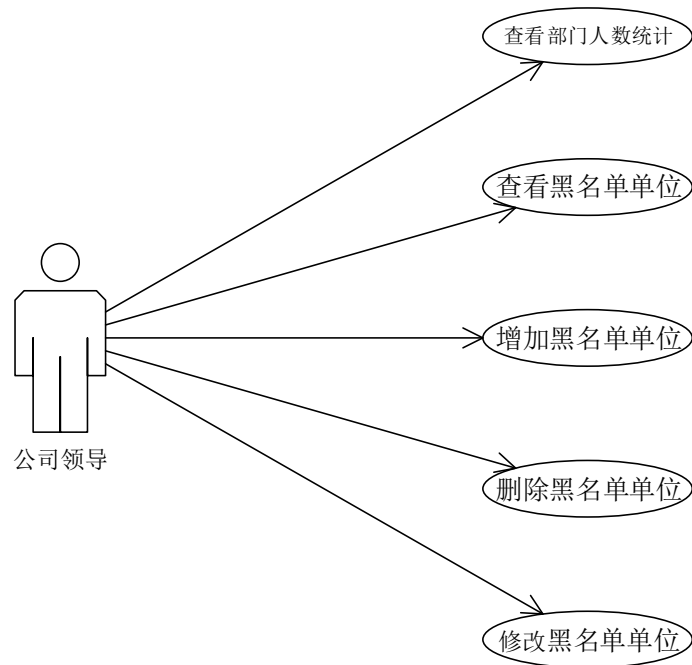


图 3.16 公司领导用例图

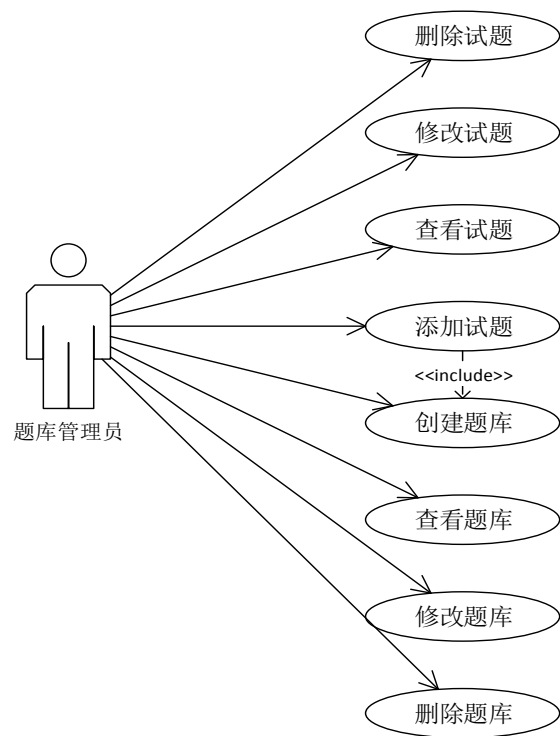


图 3.17 题库管理员用例图

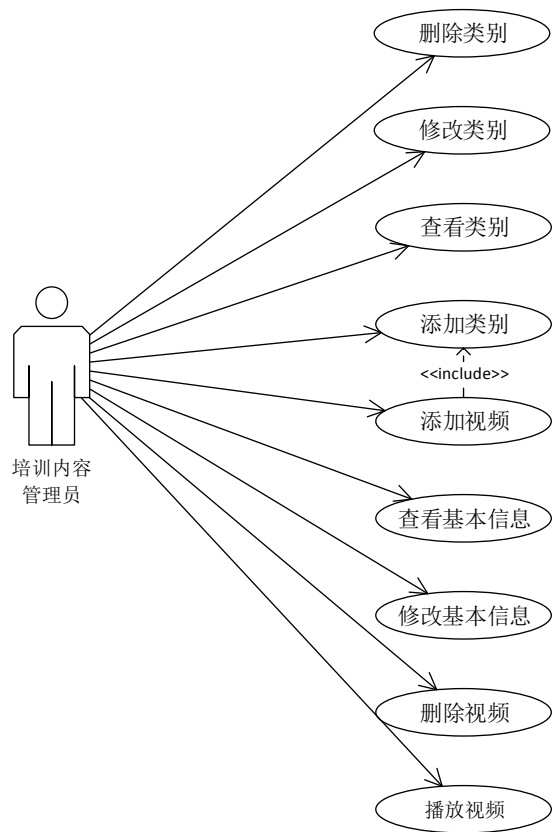


图 3.18 培训内容管理员用例图

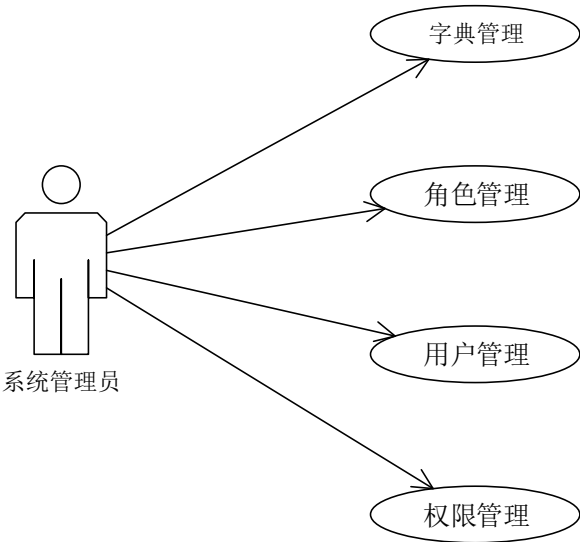


图 3.19 系统管理员用例图

3.2.5 非功能性需求

在进行系统分析时，除了需要考虑上述的功能性需求外，非功能性的需求也是设计人员必须要分析考虑的一项内容。比如说系统的运行环境、性能、成本、功率消耗等因素，这些都是设计人员需要考虑的内容。

1) 电厂培训系统的运行环境，其中服务器端环境如表 3-1 所示，客户端环境如表 3-2 所示。

表 3-1 服务器端环境

系统所使用的数据库	MySQL，Redis
JDK	1.7
Web 服务器	Tomcat7.0

表 3-2 客户端环境

操作系统	Win7 及以上
浏览器	IE8，谷歌火狐等常用浏览器
分辨率	1314*737

2) 可保障性需求

(1) 可维护性

服务器端采用 bat 脚本文件和任务调度，定时备份系统数据，如果用户因操作不当而造成数据丢失，可以选择时间节点进行恢复。同时本系统有详细的开发文档资料和清晰的代码注释，开发代码规范，易于理解。

(2) 可扩展性

系统功能结构设计时需要考虑需求变更导致的扩展性问题，需求变化时能快速对系统进行扩展且不影响其他功能。

(3) 可移植性

采用 Java 语言开发，具有跨平台的特性，并且提供数据查询接口，能够快速与其他系统对接。

3) 易用性需求

系统菜单功能模块划分简洁清晰，交互界面设计友好，开发指导手册内容详实，操作十分简便。

4) 安全需求

(1) 容错性

如果用户录入了非法的数据，系统给出友好的提示，让用户核对数据的准确性，不会因为用户操作的失误导致系统崩溃。

(2) 数据安全性

系统中有多层次的数据参数校验，一些重要的操作都配置有相关的权限，安全性较高。同时在服务器中利用计划调度定期对系统的关键数据进行备份，能够快速恢复系统误删除的数据。

3.3 本章小结

本章主要是对电厂安全培训管理系统的前期可行性和需求的分析。首先从软件系统开发常用的几个方面对培训系统进行分析，确保电厂安全培训管理系统的开发是十

分必要的。接着按照软件工程的开发流程对培训系统进行需求业务以及其他非功能性需求等方面的分析，为培训系统的总体设计做好准备工作。

第四章 系统总体设计

系统总体设计专注于如何实现系统功能。本章主要是用来阐述电厂安全培训管理系统的系统功能模块和主要的框架结构,从系统层面对电厂培训系统进行分析和设计。

4.1 系统功能模块图

经过系统需求分析,可以将电厂安全培训管理系统划分为四个主要模块,分别是人员管理模块、培训管理模块、资料管理模块和系统管理模块。整个电厂安全培训管理系统的功能模块图如图 4.1 所示:

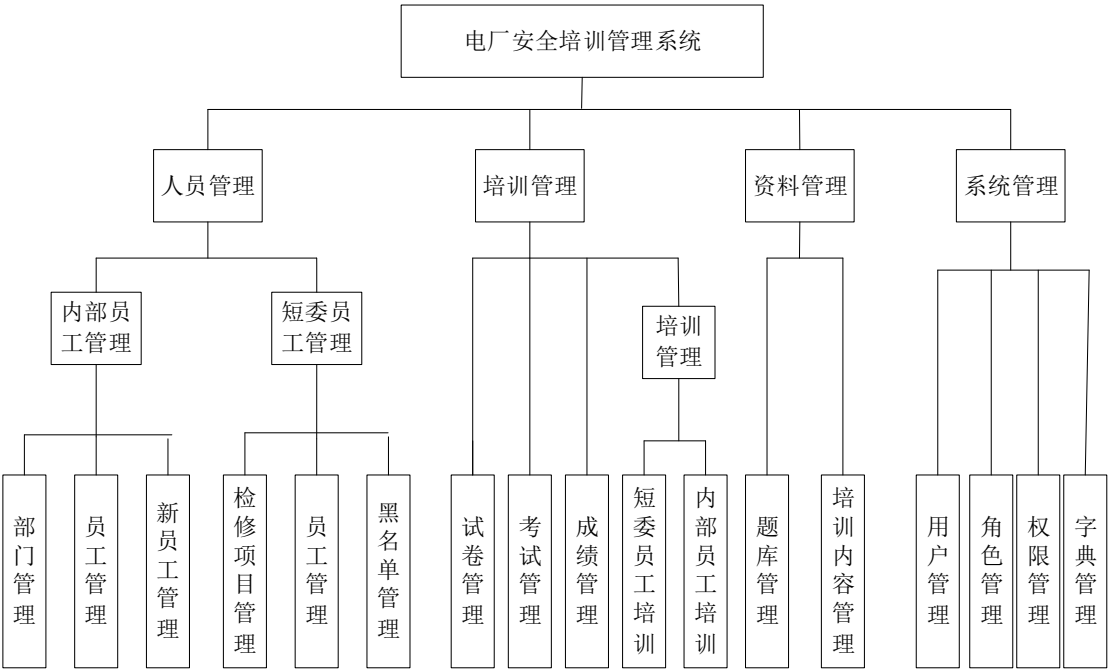


图 4.1 电厂安全培训系统功能模块图

下面对电厂安全培训系统中的人员管理和资料管理相关模块的功能进行简要的描述。

人员管理模块

1) 内部员工管理

内部员工管理主要有部门管理中内部单位和长委单位两种单位的基本信息的增删改查，以及所有二级部门和长委单位的总人数和违章积分信息的统计，员工管理中内部和长委员工和基本信息和违章信息的增删改查，员工个人培训档案的查看和导出，按照年龄段和违章时间等条件对员工信息进行统计查询。新员工管理中对新入职的员工进行基础信息维护等功能。

2) 短委员工管理

短委员工管理主要有检修项目管理中检修基础信息的维护，检修标段信息的维护统计，每个检修项目中短委单位基本信息增删改查和单位违章积分和人数的统计，已发放工作证人数统计分析。短委员工管理中检修单位树的查询显示，员工基本信息和违章信息的增删改查，员工个人培训档案的查询。黑名单管理中黑名单单位基础信息的维护以及拉黑人员违章信息的查看和黑名单记录的删除操作等功能。

资料管理模块

1) 题库试题管理

题库试题管理主要有题库的基础信息维护，试题的批量导入，试题批量导入模版下载，题库试题统计分析，题库导出以及试题的基础信息维护，试题的详情预览，试题的批量删除和批量移动等功能。

2) 培训内容管理

培训内容管理主要有培训类别树的维护，视频资料的基础信息维护，上传视频的详情查看和播放，以及批量删除操作等功能。

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。现在大多数的 Java Web 系统开发采用的 MVC 模式。MVC 分别代表 model、controller、view。

- 1) 模型 (model): 代表数据, 就是实体类的对象及其相关属性。
- 2) 控制器 (controller): 相当于管理者。
- 3) 视图 (view): 相当于系统向用户的展示效果。

它的逻辑结构如下如图 4.2 所示:

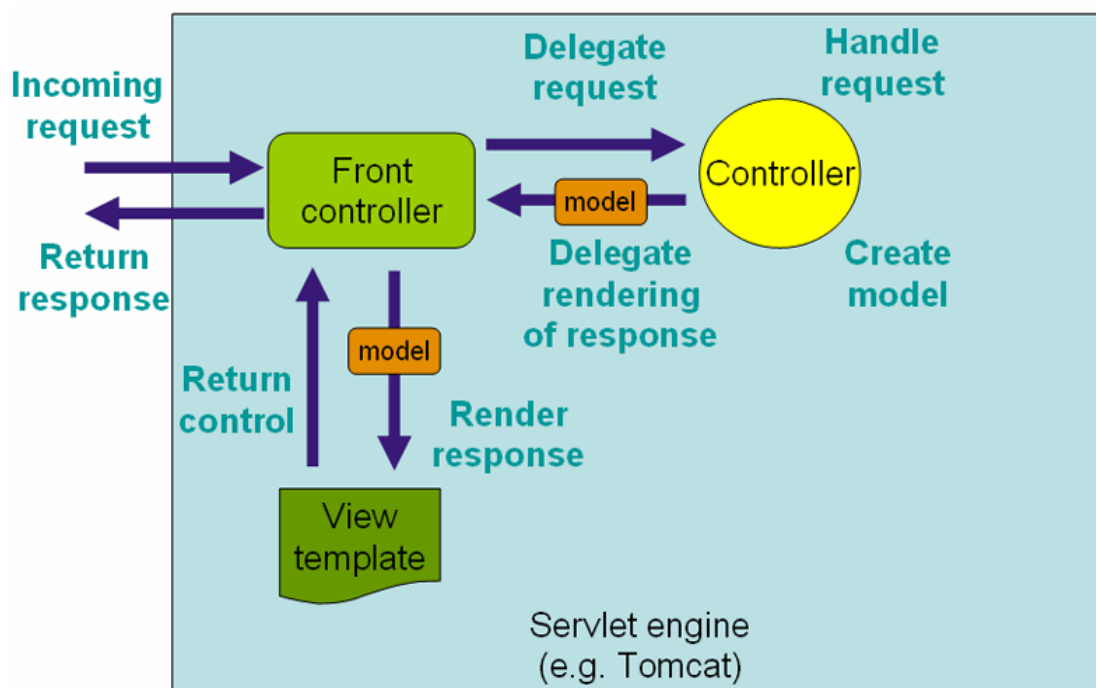


图 4.2 MVC 逻辑结构图

4.2.2 三层架构

通常意义上的三层架构就是将整个系统应用划分为：界面层、业务逻辑层、数据访问层。划分层次的目的主要是为了体现软件设计“高内聚低耦合”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。其逻辑结构图如图 4.3 所示：

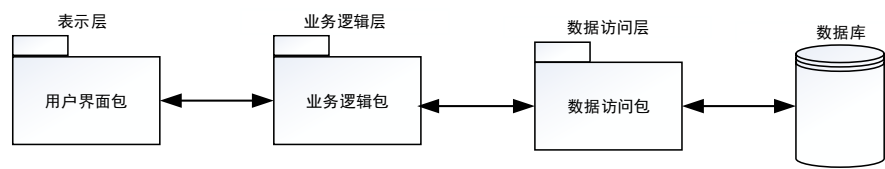


图 4.3 三层逻辑结构图

4.2.3 软件框架设计

开发框架选型时不仅需要考虑框架在市面上的受欢迎程度，还要考虑本项目小组成员的能力水平和开发习惯，选择适合的软件框架进行开发。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+Struts2+Mybatis+Shiro，后端的这四个框架已经在大量的软件开发项目中使用，是极其优秀的开源框架，且本小组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。

4.3.1 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：

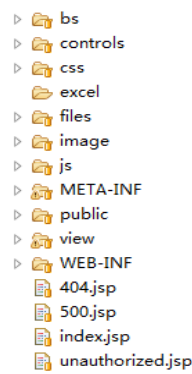


图 4.4 前端结构图

表 4-1 前端结构目录描述

目录结构名称	具体描述
bs	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	系统 css 的目录
files	用于存放系统一些临时生成的一些文件如个人培训档案，题库试题信息等
images	存放一些电厂安全培训系统常用的图标
js	各个模块的 Javascript 脚本
public	存放公共的 JS 和 JSP 文件
view	存放的是和电厂系统模块相关的 JSP 页面
其他 jsp 文件	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:

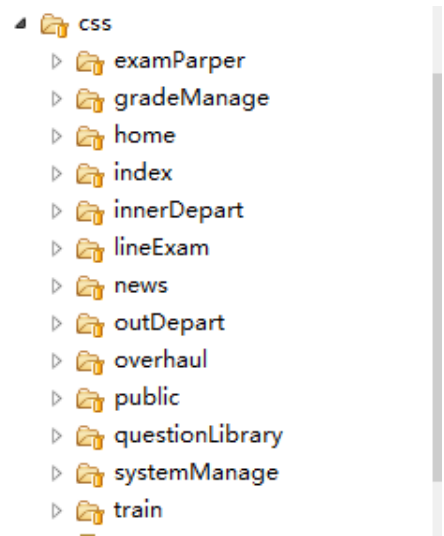


图 4.5 CSS 目录结构图

表 4-2 CSS 结构目录描述

目录结构名称	具体描述
examParper	存放考试试卷模块界面的相关样式
gradeManage	存放成绩管理模块界面的相关样式
home	存放系统的公共部分的相关样式
index	存放首页的相关样式
innerDepart	存放内部部门管理模块界面的相关样式
lineExam	存放线上考试模块界面的相关样式
news	存放新闻管理模块界面的相关样式（该模块在后期被取消）
outDepart	存放外部部门管理模块界面的相关样式
overhaul	存放检修管理模块界面的相关样式
public	公共的 CSS
questionLibrary	存放考试试卷模块界面的相关样式
systemManage	存放系统管理模块界面的相关样式
train	存放培训资料管理模块界面的相关样式

4.3.2 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:

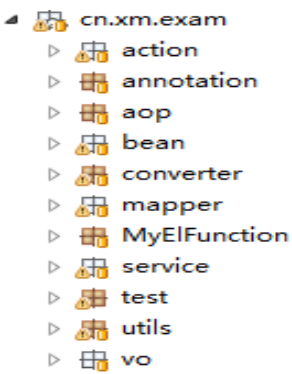


图 4.6 后端总体结构图

表 4-3 后端总体结构目录描述

目录结构名称	具体描述
action	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	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	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
annotation	自定义注解,主要是为了自定义注解实现日志功能
aop	存放 Spring 的 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	自定义 EL 表达式
utils	存放系统的一些工具类,如日期处理与 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:

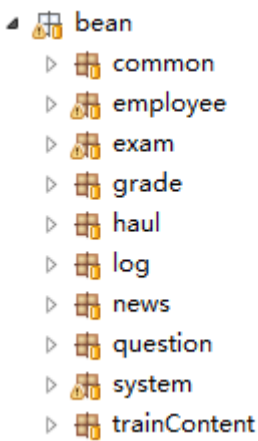


图 4.7 bean 具体结构图

表 4-4 bean 结构目录描述

目录结构名称	具体描述
common	存放系统公共信息的实体类,比如字典和消息的实体类

employee	存放内部员工和外部员工的实体类
exam	存放考试管理的实体类
haul	存放检修管理的实体类
log	存放日志管理的实体类
new	存放新闻管理的的实体类(后来此功能被取消)
question	存放试题和题库管理的实体类
system	存放系统管理的实体类
trainContent	存放培训内容管理的实体类

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

详细设计阶段主要是对总体设计的具体分析，用来阐述系统的具体实现思路和方法。本章主要从系统的数据库设计、模块接口设计、人机交互设计等方面来介绍电厂安全培训管理系统的详细设计。

5.1 数据库设计

数据库是系统的核心，它对于系统开发至关重要。该系统采用的是MySQL数据库来存储电厂培训系统的相关信息。接下来将根据系统分析的结果，详细介绍电厂培训系统的数据库设计。

5.1.1 概念数据模型设计

概念数据模型（Conceptual Data Model, CDM）也称为信息模型，它是以实体-联系（Entity-Relationship, E-R）理论为基础，并对其进行扩充产生的一种模型，从用户的观点出发对信息进行建模，主要用于数据库的概念级设计。

通过系统需求分析后设计的 CDM 如下图 5.1、5.2 所示。

5.1.2 物理数据模型设计

物理数据模型（Physical Data Model, PDM）用于定义自己系统的数据库的数据结构。建立 PDM 的过程，实际上是用图形化的方法建立数据定义语言、数据操纵语言和数据控制语言等 SQL 语句的过程。在 PDM 中，最主要的概念有业务规则、表、列、主键、候选键、引用等。

遵循规范化理论设计的 PDM 可以解决表中数据的数据冗余等问题。但是由于表的规范化过程实际上是表的拆分过程，因此规范化程度越高，系统的运行速度越慢。通过上面的概念数据模型并结合反规范化设计生成的物理数据模型如图 5.3、图 5.4 所示。

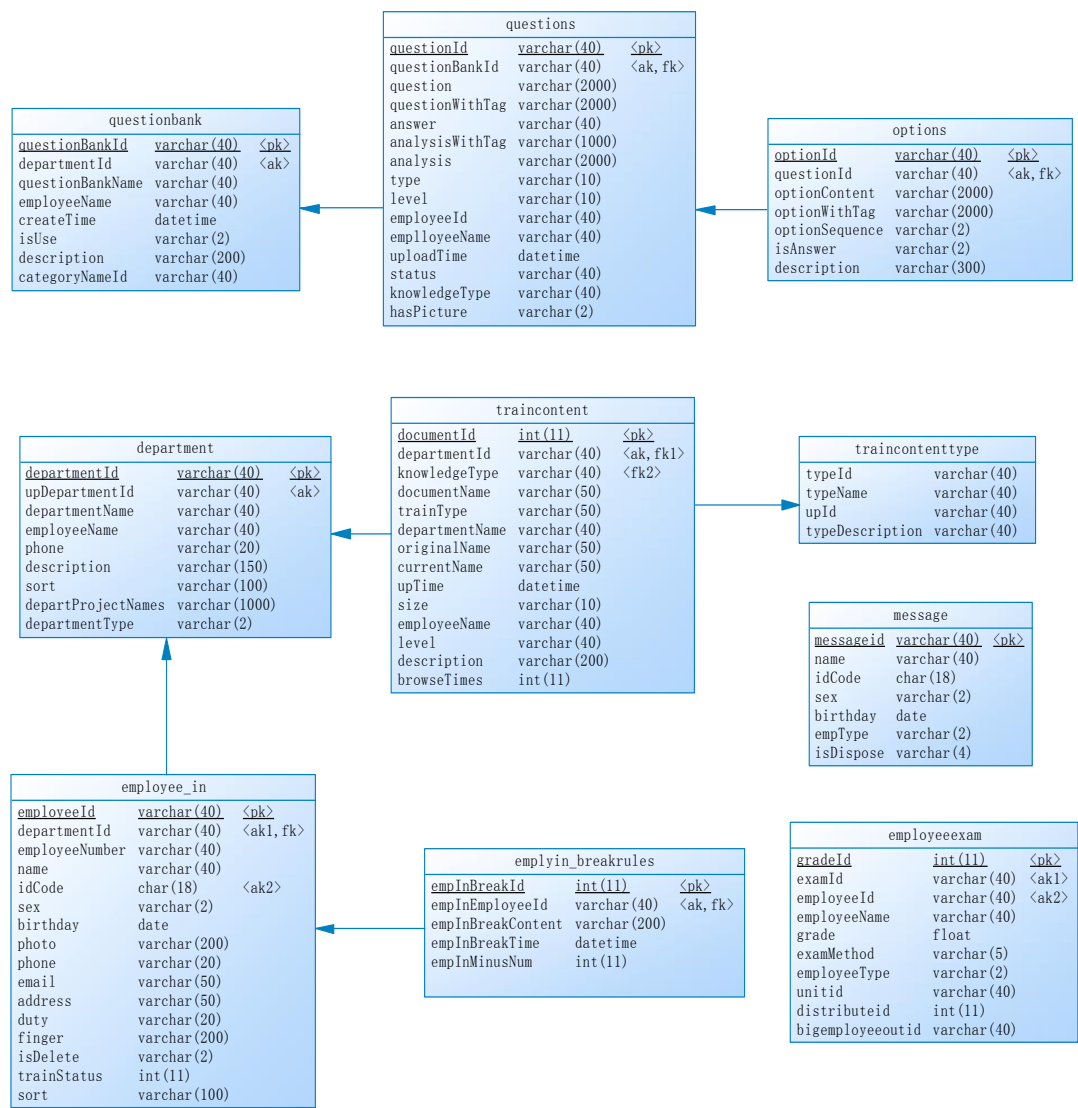


图5.3 电厂安全培训管理系统物理数据模型图分图1

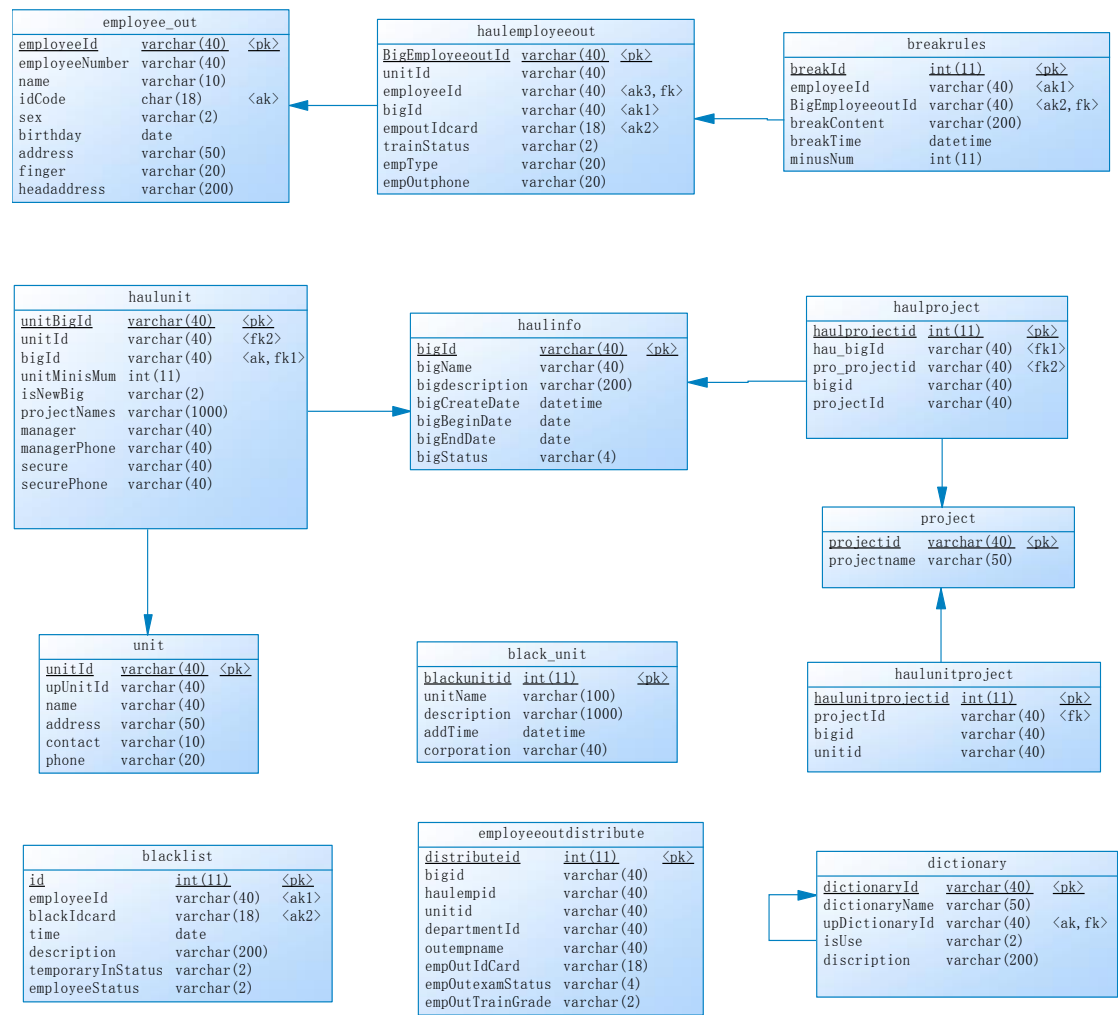


图5.4 电厂安全培训管理系统物理数据模型图分图2

5.1.3 数据库视图设计

视图（View）是建立在一个或多个表上的虚表，它由来自多个表中的列组成，代表一组 SQL 语句。具有数据安全性等特点，可以实现对原始数据的保护。

根据需求分析和业务需要，本系统设计了一些视图来简化对数据库的查询操作，提高开发效率，下面就检修部门视图和题库统计视图进行详细介绍。

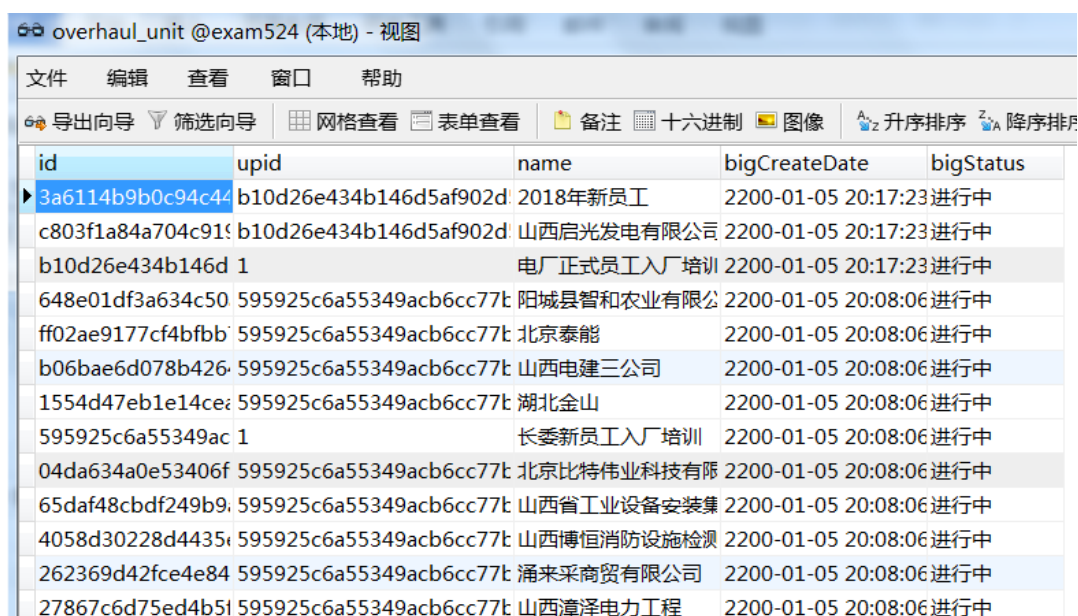
1) 检修部门视图

在短委员工管理中添加外来员工需要在检修单位树上选择一个外来单位,该视图通过查询检修基本信息和检修部门信息拼接查询结果实现检修部门树的操作。

实现 SQL 语句

```
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER
VIEW `overhaul_unit` AS SELECT
    `haulinfo`.`bigId` AS `id`,
    (CASE WHEN 1 THEN 1 ELSE 2 END) AS `upid`,
    `haulinfo`.`bigName` AS `name`,
    `haulinfo`.`bigCreateDate` AS `bigCreateDate`,
    `haulinfo`.`bigStatus` AS `bigStatus`
FROM
    `haulinfo`
UNION
SELECT
    `haulunit`.`unitId` AS `id`,
    `haulunit`.`bigId` AS `upid`,
    `unit`.`name` AS `name`,
    `haulinfo`.`bigCreateDate` AS `bigCreateDate`,
    `haulinfo`.`bigStatus` AS `bigStatus`
FROM
    ((`haulunit` JOIN `unit`) JOIN `haulinfo`)
WHERE
    ((`unit`.`unitId` = `haulunit`.`unitId`) AND (`haulinfo`.`bigId` = `haulunit`.`bigId`))
ORDER BY
    `bigCreateDate` DESC
```

检修单位视图查询结果如下图 5.5 所示



id	upid	name	bigCreateDate	bigStatus
3a6114b9b0c94c44	b10d26e434b146d5af902d	2018年新员工	2200-01-05 20:17:23	进行中
c803f1a84a704c91f	b10d26e434b146d5af902d	山西启光发电有限公司	2200-01-05 20:17:23	进行中
b10d26e434b146d 1		电厂 正式员工入厂培训	2200-01-05 20:17:23	进行中
648e01df3a634c50	595925c6a55349acb6cc77f	阳城县智和农业有限公	2200-01-05 20:08:06	进行中
ff02ae9177cf4bfbf	595925c6a55349acb6cc77f	北京泰能	2200-01-05 20:08:06	进行中
b06bae6d078b426	595925c6a55349acb6cc77f	山西电建三公司	2200-01-05 20:08:06	进行中
1554d47eb1e14ce	595925c6a55349acb6cc77f	湖北金山	2200-01-05 20:08:06	进行中
595925c6a55349ac 1		长委新员工入厂培训	2200-01-05 20:08:06	进行中
04da634a0e53406f	595925c6a55349acb6cc77f	北京比特伟业科技有限	2200-01-05 20:08:06	进行中
65daf48cbdf249b9	595925c6a55349acb6cc77f	山西省工业设备安装集	2200-01-05 20:08:06	进行中
4058d30228d4435f	595925c6a55349acb6cc77f	山西博恒消防设施检测	2200-01-05 20:08:06	进行中
262369d42fce4e84	595925c6a55349acb6cc77f	涌来采商贸有限公司	2200-01-05 20:08:06	进行中
27867c6d75ed4b5f	595925c6a55349acb6cc77f	山西漳泽电力工程	2200-01-05 20:08:06	进行中

图 5.5 检修单位视图查询结果

2) 题库统计视图

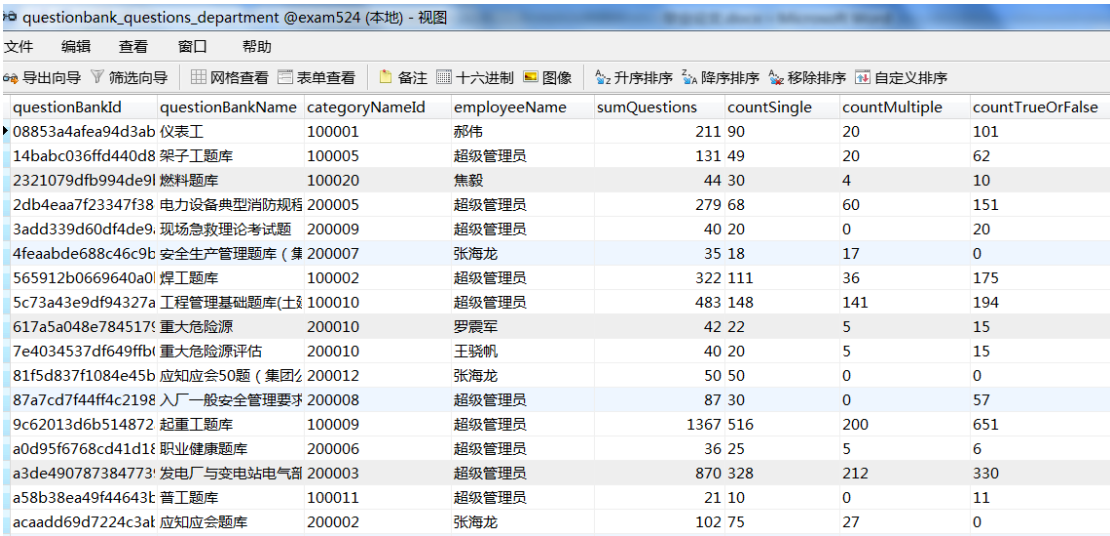
题库管理模块中有对题库试题的统计分析,需要统计出题库中各类试题的数量,该视图通过关联查询题库和实体表结合内置函数实现对题库试题数量的统计功能。

实现 SQL 语句

```
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER
VIEW `questionbank_questions_department` AS SELECT
    `questionbank`.`questionBankId` AS `questionBankId`,
    `questionbank`.`questionBankName` AS `questionBankName`,
    `questionbank`.`departmentId` AS `departmentId`,
    `questionbank`.`employeeName` AS `employeeName`,
    `questionbank`.`createTime` AS `createTime`,
    `questionbank`.`isUse` AS `isUse`,
    `questionbank`.`description` AS `description`,
    `questionbank`.`categoryNameId` AS `categoryNameId`,
    `dictionary`.`dictionaryName` AS `typeName`,
    `dictionary`.`upDictionaryId` AS `upTypeId`,
    count(`questions`.`questionId`) AS `sumQuestions`,
    sum((CASE `questions`.`type` WHEN '单选题' THEN 1 ELSE 0 END)) AS `countSingle`,
    sum((CASE `questions`.`type` WHEN '多选题' THEN 1 ELSE 0 END)) AS `countMultiple`,
    sum((CASE `questions`.`type` WHEN '判断题' THEN 1 ELSE 0 END)) AS `countTrueOrFalse`
FROM
    ( `questionbank` JOIN `dictionary` ON((`dictionary`.`dictionaryId` =
```

```
`questionbank`.`categoryNameId`)))  
LEFT JOIN `questions` ON ((`questionbank`.`questionBankId` =  
`questions`.`questionBankId`)))  
GROUP BY  
`questionbank`.`questionBankId`
```

题库统计视图查询结果如下图 5.6 所示



questionBankId	questionBankName	categoryNameId	employeeName	sumQuestions	countSingle	countMultiple	countTrueOrFalse
08853a4afea94d3ab	仪表工	100001	郝伟	211	90	20	101
14bab036ffd440d8	架子工题库	100005	超级管理员	131	49	20	62
2321079dfb994de9f	燃料题库	100020	焦毅	44	30	4	10
2db4eaa7f23347f38	电力设备典型消防规程	200005	超级管理员	279	68	60	151
3add339d60df4de9	现场急救理论考试题	200009	超级管理员	40	20	0	20
4feaabde688c46c9b	安全生产管理题库(集团)	200007	张海龙	35	18	17	0
565912b0669640a0f	焊工题库	100002	超级管理员	322	111	36	175
5c73a43e9df94327a	工程管理基础题库(土木)	100010	超级管理员	483	148	141	194
617a5a048e784517f	重大危险源	200010	罗震军	42	22	5	15
7e4034537df649ffb	重大危险源评估	200010	王骁帆	40	20	5	15
81f5d837f1084e45b	应知应会50题(集团)	200012	张海龙	50	50	0	0
87a7cd7f44ff4c2198	入厂一般安全管理要求	200008	超级管理员	87	30	0	57
9c62013d6b514872	起重工题库	100009	超级管理员	1367	516	200	651
a0d95f6768cd41d1f	职业健康题库	200006	超级管理员	36	25	5	6
a3de490787384773f	发电厂与变电站电气部分	200003	超级管理员	870	328	212	330
a58b38ea49f44643b	普工题库	100011	超级管理员	21	10	0	11
acaadd69d7224c3a4	应知应会题库	200002	张海龙	102	75	27	0

图 5.6 题库统计视图查询结果

5.1.4 存储过程和事件计划设计

存储过程（Stored Procedure）是存储在数据库中的预先编译好的 SQL 语句集合，相当于一个 DBMS 的自定义函数。在应用程序中能够调用预先定义好的存储过程。它实质上就是一组完成特定功能的 SQL 语句，可以与 MySQL 的事件计划结合使用完成业务上需要的功能。

本系统因为业务需要也设计了存储过程来完成相应的功能，下面以修改数据状态存储过程为例做简单的介绍。

本系统有许多和时间有关的状态，如安排的考试有未开始，正在答题，已结束三种状态，创建的检修也有未开始，进行中，已结束三种状态。这些状态需要系统随着时间的变化动态的进行修改。在本系统中通过事件计划和存储过程结合使用来实现。

存储过程 SQL 语句

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `updateStatus`()
BEGIN
    UPDATE exam SET `status`="已结束" WHERE `status` != "已结束" AND (NOW() -
endTime)>0 ;
    UPDATE exam SET `status`="正在答题" WHERE `status` != "已结束" AND (NOW() -
startTime)>=0
    AND (NOW() - endTime)<=0;
    UPDATE haulinfo SET bigStatus="已结束" WHERE bigStatus != "已结束" AND
(CURDATE() - bigEndDate)>0;
    UPDATE haulinfo SET bigStatus="进行中" WHERE (CURDATE() - bigEndDate)<=0
    AND (CURDATE() - bigBeginDate)>=0;
    UPDATE exam SET bigStatus=(SELECT bigStatus FROM haulinfo WHERE
bigId=exam.bigId);
END
```

事件计划 SQL 语句

```
CREATE DEFINER=`root`@`localhost` EVENT `eventUpdateStatus` ON SCHEDULE EVERY 1
SECOND STARTS '2018-5-21 00:12:44' ON COMPLETION PRESERVE ENABLE DO call
updateStatus()
```

5.1.5 数据库表结构设计

1) black_unit 是黑名单单位表，用于存放黑名单单位基本信息。

表 5-1 black_unit（黑名单单位表）

代码	名称	数据类型	主键	强制	注释
blackunitid	blackunitid	int(11)	TRUE	TRUE	黑名单单位编号
unitName	unitName	varchar(100)			单位名称
description	description	varchar(1000)			拉黑原因
addTime	addTime	datetime			拉黑时间
corporation	corporation	varchar(40)			公司法人

2) blacklist 是黑名单表，用于存放进入黑名单的员工信息。

表 5-2 blacklist （黑名单表）

代码	名称	数据类型	主键	强制	注释
id	id	int(11)	TRUE	TRUE	编号
employeeId	employeeId	varchar(40)			员工编号
time	time	date			拉黑时间
description	description	varchar(200)			描述
blackIdcard	blackIdcard	varchar(18)			身份证号
temporaryInStatus	temporaryInStatus	varchar(2)			是否是临时进入黑名单字段
employeeStatus	employeeStatus	varchar(2)			员工状态是否是内部员工表中的员工

3) breakrules 是违章表，用于存放外来单位临时员工的违章信息

表 5-3 breakrules （违章表）

代码	名称	数据类型	主键	强制	注释
breakId	breakId	int(11)	TRUE	TRUE	外部员工违章编号
employeeId	employeeId	varchar(40)			员工编号
BigEmployeeoutId	BigEmployeeoutId	varchar(40)			大修员工编号
breakContent	breakContent	varchar(200)			违章内容
breakTime	breakTime	datetime			违章时间
minusNum	minusNum	int(11)			减分

4) department 是内部部门表，用于存放电厂内部部门信息。

表 5-4 department （内部部门表）

代码	名称	数据类型	主键	强制	注释
departmentId	departmentId	varchar(40)	TRUE	TRUE	部门编号(UUID 生成)
upDepartmentId	upDepartmentId	varchar(40)			上级编号

departmentName	departmentName	varchar(40)		TRUE	部门名称
employeeName	employeeName	varchar(40)			负责人姓名
phone	phone	varchar(20)			负责人电话
description	description	varchar(150)			描述
sort	sort	varchar(100)			排序用的(日期的字符串形式)
departProjectNames	departProjectNames	varchar(1000)			长期外来单位的工程信息
departmentType	departmentType	varchar(2)			部门类型0代表内部部门，1代表长期外来部门

5) dictionary 是字典表，用于存放系统字典信息。

表 5-5 dictionary（字典表）

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

6) employee_in 是内部员工基本信息表，用于存放内部员工的基本信息。

表 5-6 employee_in（内部员工基本信息表）

代码	名称	数据类型	主键	强制	注释
employeeId	employeeId	varchar(40)	TRUE	TRUE	编号
employeeNumber	employeeNumber	varchar(40)		TRUE	员工号(没用上)
name	name	varchar(40)		TRUE	姓名

idCode	idCode	char(18)		TRUE	身份证号
sex	sex	varchar(2)			性别
birthday	birthday	date			生日
photo	photo	varchar(200)			图片(存放路径)
phone	phone	varchar(20)			电话
email	email	varchar(50)			邮箱(没用)
address	address	varchar(50)			住址
duty	duty	varchar(20)			职务
departmentId	departmentId	varchar(40)			部门编号
finger	finger	varchar(200)			指纹
isDelete	isDelete	varchar(2)			是否删除
trainStatus	trainStatus	int(11)			培训状态
sort	sort	varchar(100)			排序

7) employee_out 表是外部员工基本信息表，用于存放短委员工基本信息。

表 5-7 employee_out（外部员工基本信息表）

代码	名称	数据类型	主键	强制	注释
employeeId	employeeId	varchar(40)	TRUE	TRUE	短委员工 ID
employeeNumber	employeeNumber	varchar(40)			编号
name	name	varchar(10)			员工名称
idCode	idCode	char(18)			身份证号
sex	sex	varchar(2)			性别
birthday	birthday	date			生日
address	address	varchar(50)			地址
finger	finger	varchar(20)			指纹
headaddress	headaddress	varchar(200)			头像地址

8) employeeexam 是员工成绩表，用于存放员工的考试成绩信息。

表 5-8 employeeexam（员工成绩表）

代码	名称	数据类型	主键	强制	注释
gradeId	gradeId	int(11)	TRUE	TRUE	成绩 ID
examId	examId	varchar(40)			考试 ID
employeeId	employeeId	varchar(40)			员工身份证号
employeeName	employeeName	varchar(40)			员工名称
grade	grade	float			成绩
examMethod	examMethod	varchar(5)			考试方式
employeeType	employeeType	varchar(2)			员工类型
unitid	unitid	varchar(40)			外来单位 ID
distributeid	distributeid	int(11)			分配表的主键
bigemployeeoutid	bigemployeeoutid	varchar(40)			外来大修员工编号

9) employeeoutdistribute 是外部员工分配表，用于存放培训分配信息。

表 5-9 employeeoutdistribute（外部员工分配表）

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

10) employin_breakrules 是内部员工违章表，用于存放内部员工违章信息。

表 5-10 employin_breakrules（内部员工违章表）

代码	名称	数据类型	主键	强制	注释
empInBreakId	empInBreakId	int(11)	TRUE	TRUE	内部违章 ID
empInEmployeeId	empInEmployeeId	varchar(40)			内部员工表中的 员工 ID
empInBreakContent	empInBreakContent	varchar(200)			违章内容
empInBreakTime	empInBreakTime	datetime			违章时间
empInMinusNum	empInMinusNum	int(11)			违章积分

11) haulemployeeout 是检修员工表，用于存放检修员工的基本信息。

表 5-11 haulemployeeout（检修员工表）

代码	名称	数据类型	主键	强制	注释
BigEmployeeoutId	BigEmployeeoutId	varchar(40)	TRUE	TRUE	检修员工 ID
unitId	unitId	varchar(40)			单位 ID
employeeId	employeeId	varchar(40)			短委员工 ID
bigId	bigId	varchar(40)			检修 ID
empoutIdcard	empoutIdcard	varchar(18)			身份证号
trainStatus	trainStatus	varchar(2)			培训状态
empType	empType	varchar(20)			工种
empOutphone	empOutphone	varchar(20)			联系电话

12) haulinfo 是检修基本信息表，用于存放检修基本信息。

表 5-12 haulinfo（检修基本信息表）

代码	名称	数据类型	主键	强制	注释
bigId	bigId	varchar(40)	TRUE	TRUE	检修 ID
bigName	bigName	varchar(40)			检修名称

bigdescription	bigdescription	varchar(200)			描述
bigCreateDate	bigCreateDate	datetime			创建时间
bigBeginDate	bigBeginDate	date			开始时间
bigEndDate	bigEndDate	date			结束时间
bigStatus	bigStatus	varchar(4)			检修状态

13) haulproject 是检修项目表，用于存放检修标段信息。

表 5-13 haulproject（检修项目表）

代码	名称	数据类型	主键	强制	注释
haulprojectid	haulprojectid	int(11)	TRUE	TRUE	检修项目 ID
bigid	bigid	varchar(40)			检修 ID
projectId	projectId	varchar(40)			项目 ID

14) haulunit 是检修单位表，用于存放检修单位信息。

表 5-14 haulunit（检修单位表）

代码	名称	数据类型	主键	强制	注释
unitBigId	unitBigId	varchar(40)	TRUE	TRUE	检修单位 ID
unitId	unitId	varchar(40)			单位 ID
bigId	bigId	varchar(40)			检修 ID
unitMinisMum	unitMinisMum	int(11)			单位违章记分
isNewBig	isNewBig	varchar(2)			是否最新
projectNames	projectNames	varchar(1000)			工程名称
manager	manager	varchar(40)			负责人
managerPhone	managerPhone	varchar(40)			负责人电话
secure	secure	varchar(40)			安全员
securePhone	securePhone	varchar(40)			安全员电话

15) haulunitproject 是检修单位工程表，用于存放检修单位负责的标段信息。

表 5-15 haulunitproject（检修单位工程表）

代码	名称	数据类型	主键	强制	注释
haulunitprojectid	haulunitprojectid	int(11)	TRUE	TRUE	检修单位工程 ID
bigid	bigid	varchar(40)			大修 ID
unitid	unitid	varchar(40)			单位 ID
projectId	projectId	varchar(40)			工程编号

16) message 超龄员工信息表，用于存放超龄的员工信息。

表 5-16 message（超龄员工信息表）

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

17) options 是试题选项表，用于存放试题选项信息。

表 5-17 options（试题选项表）

代码	名称	数据类型	主键	强制	注释
optionId	optionId	varchar(40)	TRUE	TRUE	选项 ID

questionId	questionId	varchar(40)			试题 ID
optionContent	optionContent	varchar(2000)			选项内容
optionWithTag	optionWithTag	varchar(2000)			带标签的内容
optionSequence	optionSequence	varchar(2)			选项序号
isAnswer	isAnswer	varchar(2)			是否是答案
description	description	varchar(300)			描述

18) project 是工程基本信息表，用于存放工程基本信息。

表 5-18 project（工程基本信息表）

代码	名称	数据类型	主键	强制	注释
projectid	projectid	varchar(40)	TRUE	TRUE	工程 ID
projectname	projectname	varchar(50)			工程名称

19) questionbank 是题库信息表，用于存放题库的基本信息。

表 5-19 questionbank（题库表）

代码	名称	数据类型	主键	强制	注释
questionBankId	questionBankId	varchar(40)	TRUE	TRUE	题库 ID
questionBankName	questionBankName	varchar(40)			题库名称
departmentId	departmentId	varchar(40)			部门 ID
employeeName	employeeName	varchar(40)			创建人姓名
createTime	createTime	datetime			创建时间
isUse	isUse	varchar(2)			是否使用
description	description	varchar(200)			描述
categoryNameId	categoryNameId	varchar(40)			类别名称编号

20) questions 是试题表，用于存放试题基本信息。

表 5-20 questions（试题表）

代码	名称	数据类型	主键	强制	注释
questionId	questionId	varchar(40)	TRUE	TRUE	试题 ID
questionBankId	questionBankId	varchar(40)			题库 ID
question	question	varchar(2000)			题干
questionWithTag	questionWithTag	varchar(2000)			带标签的题干
answer	answer	varchar(40)			答案
analysisWithTag	analysisWithTag	varchar(1000)			带标签答案
analysis	analysis	varchar(2000)			解析
type	type	varchar(10)			类型
level	level	varchar(10)			等级
employeeId	employeeId	varchar(40)			上传人 ID
employeeName	employeeName	varchar(40)			上传人姓名
uploadTime	uploadTime	datetime			上传时间
status	status	varchar(40)			状态
knowledgeType	knowledgeType	varchar(40)			知识点
hasPicture	hasPicture	varchar(2)			是否有图片

21) traincontent 是培训内容基本信息表，用于存放培训内容的基本信息。

表 5-21 traincontent（培训内容基本信息表）

代码	名称	数据类型	主键	强制	注释
documentId	documentId	int(11)	TRUE	TRUE	培训内容 ID
departmentId	departmentId	varchar(40)			部门 ID
documentName	documentName	varchar(50)			资料名称
trainType	trainType	varchar(50)			类型
departmentName	departmentName	varchar(40)			部门名称
knowledgeType	knowledgeType	varchar(40)			知识点

originalName	originalName	varchar(50)			原始名称
currentName	currentName	varchar(50)			最新名称
upTime	upTime	datetime			上传时间
size	size	varchar(10)			大小
employeeName	employeeName	varchar(40)			上传人姓名
level	level	varchar(40)			等级
description	description	varchar(200)			描述
browseTimes	browseTimes	int(11)			浏览次数

22) traincontenttype 是培训内容类别表，用于存放培训内容类别信息。

表 5-22 traincontenttype（培训内容类别表）

代码	名称	数据类型	主键	强制	注释
typeId	typeId	varchar(40)		TRUE	培训内容类别编号
typeName	typeName	varchar(40)			培训内容类别名称
upId	upId	varchar(40)			培训内容类别上级编号
typeDescription	typeDescription	varchar(40)			描述(保留字段)

23) unit 是外来单位基本信息表，用于存放外来单位基本信息。

表 5-23 unit（外来单位基本信息表）

代码	名称	数据类型	主键	强制	注释
unitId	unitId	varchar(40)	TRUE	TRUE	部门 ID
upUnitId	upUnitId	varchar(40)			上级部门 ID
name	name	varchar(40)			部门名称
address	address	varchar(50)			地址
contact	contact	varchar(10)			联系人
phone	phone	varchar(20)			联系电话

5.2 接口设计

本系统采用的是面向接口编程的方式进行系统开发，所谓面向接口编程就是先把客户的业务逻辑抽取出来，设计成暴露在外的接口，而业务的具体实现通过该接口的实现类来完成，与设计的接口分隔开来。当电厂安全培训管理系统的需求发生变化时，只需要修改相对应的实现类即可，减少对系统功能结构的影响。

下面主要针对自己所负责的电厂人员管理和电厂资料管理的相关功能模块的接口进行介绍。

5.2.1 题库管理模块

QuestionbankService（题库接口类）关键方法

```
/**
 * 增加题库(id 使用 UUID 工具类生成)
 * @param questionBank
 * @return
 * @throws Exception
 */
public boolean addQuestionBank(Questionbank questionBank) throws Exception;

/**
 * 修改题库基本信息
 * @param questionBank 修改后的对象
 * @return
 * @throws Exception
 */
public boolean updateQuestionBanK(Questionbank questionBank) throws Exception;

/**
 * 获取所有的题库名称和 ID，初始化下拉列表使用
 * @return
 * @throws Exception
 */
public List<Map<String,Object>> getQuestionBankNameList() throws Exception;

/**
```

```

    * 根据 id 删除题库
    * (实现层首先删除题库中的试题和选项，然后再删除题库)
    * @param questionBankId
    * @return
    * @throws Exception
    */
    public boolean deleteBankInfoByQuestionBankId(String questionBankId) throws Exception;

    /**
    * 分页查询题库:根据题库名称，题库类别组合查询
    * @param currentPage 当前页，默认第一页
    * @param currentTotal 当前页的数量
    * @param condition 查询条件
    * @return
    * @throws Exception
    */
    public PageBean<QuestionbankQuestionsDepartment> findQuestionBankWithCondition(int
    currentPage, int currentCount,
        Map<String, Object> condition) throws Exception;

```

5.2.2 试题管理模块

QuestionsService（试题选项接口类）关键方法

```

    /**
    * 添加试题，同时将选项解析添加到选项表
    * @param questions 试题对象
    * @param option 选项对象集合
    * @return
    * @throws Exception
    */
    public boolean saveQuestionsAndOptions(Questions questions, List<Options> options) throws
    Exception;

    /**
    * 批量添加试题
    * @param list 试题集合，一个 map 封装一道题的试题 questions 与选项 options
    * @return 导入的试题数量
    * @throws Exception
    */
    public int saveQuestionBatch(List<HashMap<String, Object>> list) throws Exception;

```



```

/**
 * 通过 id 删除试题
 * @param id 要删除的试题 questionId
 * @return 是否删除成功
 * @throws Exception
 */
public boolean deleteQuestionInfoByQuestionId(String questionId) throws Exception;

/**
 * 批量删除试题
 * @param questionIds 要删除的试题 id 集合(先批量删除选项表中的对应数据)
 * @return 影响行数
 * @throws Exception
 */
public int deleteQuestionBatch(List<String> questionIds) throws Exception;

/**
 * 批量移动试题，从一个题库移到另一个题库(将题的题库 id 修改为新的题库 id)
 * @param questionIds 需要移动的试题的 id
 * @param newQuestionBankId 移动到的题库的 id
 * @return
 * @throws Exception
 */
public boolean updateQuestionBatch(List<String> questionIds, String newQuestionBankId) throws
Exception;

```

5.2.3 培训内容管理模块

1) TraincontenttypeService（培训内容类别接口类）关键方法

```

/**
 * 增加培训类别
 * @param trainContentType
 * @return
 */
public boolean addTraincontenttype(Traincontenttype trainContentType)throws SQLException;

/**
 * 删除培训内容类别(删除自己以及自己的下级的视频以及删除类别)
 * @param typeId
 * @return
 */
public boolean deleteTraincontenttypeById(String typeId) throws SQLException;

```

```

/**
 * 修改培训内容类别
 * @param trainContentType
 * @return
 */
public boolean updateTraincontenttypeById(Traincontenttype trainContentType)throws
SQLException;
/**
 * 获取培训内容类别树
 * @return
 * @throws SQLException
 */
public List<Map<String,Object>> getTraincontenttypeTree()throws SQLException;
/**
 * 根据编号查询培训内容类别
 * @param id
 * @return
 * @throws SQLException
 */
public Traincontenttype getTraincontenttypeById(String id)throws SQLException;

```

2) TraincontentService（培训内容接口类）关键方法

```

/**
 * 增加培训内容
 * @param trainContent 要增加的培训记录
 * @return 是否增加成功
 * @throws Exception
 */
public boolean addTrainContent(Traincontent trainContent) throws Exception;
/**
 * 批量删除培训内容
 * @param trainIds 删除的培训内容的 id 集合
 * @return 是否成功
 * @throws Exception
 */
public boolean deleteTrainContentBatch(List<Integer> trainIds) throws Exception;

/**
 * 修改培训内容

```

```

    * @param trainContent 修改后的对象
    * @return
    * @throws Exception
    */
    public boolean updateTrainContent(Traincontent trainContent) throws Exception;
    /**
    * 分页查询:根据组合条件进行分页查询
    * @param currentPage 当前页，默认第一页
    * @param currentTotal 当前页的数量
    * @param condition 查询条件
    * @return
    * @throws Exception
    */
    PageBean<Traincontent> findTraincontentWithCondition(int currentPage, int
    currentTotal,Map<String, Object> condition) throws Exception;

```

5.2.4 内部部门管理模块

DepartmentService（内部部门 service 接口）关键方法

```

    /**
    * 统计内部正式单位的信息分页显示
    * @param condition
    * @return
    * @throws Exception
    */
    public PageBean<Map<String, Object>> getDepartmentInFormalCountInfo(int currentPage,
    int currentTotal,Map<String, Object> condition) throws Exception;

    /**
    * 统计内部长委单位的信息分页显示
    * @param condition
    * @return
    * @throws Exception
    */
    public PageBean<Map<String, Object>> getDepartmentInToDoCountInfo(int currentPage, int
    currentTotal,Map<String, Object> condition) throws Exception;

    /**
    * 获取内部正式单位部门和员工数

```

```

    * @return
    * @throws Exception
    */
    public Map<String,Object> getFormalDepartmentAndEmpNum();

    /**
     * 获取内部常委单位单位和部门数
     * @return
     * @throws Exception
     */
    public Map<String,Object> getToDoDepartmentAndEmpNum();

    /**
     * 公共树的查询
     * @param departmentId
     * @return
     * @throws Exception
     */
    public List<Map<String, Object>> getDepartmentTreeCommon(String departmentId) throws
    SQLException;

    /**
     * 根据上级编号获取数据库中下级部门最大 id 值，在 service 实现类中转为 Integer 加 1 后再
     变为 String 传到控制层作为添加时的部门 id
     * @param upDepartmentId 上级部门(根据这个查询下级部门最大的 Id 值)
     * @return
     * @throws Exception
     */
    public String getNextDepartmentId(String upDepartmentId);

    /**
     * 修改部门信息
     * @param department 修改过的单位对象重新传到 dao 层, dao 自动根据对象的 id 重新赋值(dao
     层修改其关联的子表)
     * @return 是否修改成功
     * @throws Exception
     */
    public boolean updateDepartment(Department department);

```

5.2.5 员工管理模块

1) EmployeeOutService (短委员工 service 接口)

```
/**
 * 查询大修和部门的树结构
 * 参数: 培训标记类型, 正式员工和外来单位长期员工的大修 ID
 * @return
 * @throws Exception
 */
public List<Map<String, Object>> getDepartmentAndOverHaulTree(Map<String, Object>
condition) throws Exception ;

/**
 * 根据外来单位员工的身份证号查询该员工所有的考试信息
 * @param idCard
 * @return
 * @throws Exception
 */
public List<Map<String, Object>> getExamsInfoByEmployeeOutIdCard(String idCard) throws
Exception;

/**
 * 根据身份证号查询外来单位员工的所有考试信息, 分页显示
 * @param currentPage
 * @param currentCount
 * @param idCard
 * @return
 * @throws Exception
 */
public PageBean<Map<String, Object>> getExamsInfoByEmployeeOutIdCard(int currentPage,
int currentCount, String idCard) throws Exception;

/**
 * 根据员工 ID 和参加员工大修 ID 查询员工的违章信息
 * @param condition
 * @return
 * @throws Exception
 */
public List<Breakrules> getBreakRulesInfoByCondition(Map<String, Object> condition) throws
Exception;
```

```

/**
 * 根据条件查询符合条件的员工信息用于生成工作证
 * @param condition
 * @return
 * @throws Exception
 */
public List<EmployeeOutBaseInfo> getEmpInfoForCertificateWithCondition(Map<String,Object>
condition) throws Exception;

/**
 * 批量加入外部员工的基本信息及参加大修外部员工的基本信息
 * @param employeeOutList
 * @param haulemployeeoutList
 * @return
 * @throws Exception
 */
public int addEmployeeOutBatch(List<EmployeeOut> employeeOutList,List<Haulemployeeout>
haulemployeeoutList) throws Exception;

/**
 * 根据身份证号查询该员工的状态
 * 返回：1、表示没有来过 2、表示来过 3、表示进入黑名单 4、表示已经添加到这次大
修的其他部门中
 * @param idCard bigId
 * @return
 * @throws Exception
 */
public int findEmployeeOutStatus(String idCard,String bigId,String unitId) throws Exception;

/**
 * 根据参加大修员工身份证集合查询员工的 id 集合
 * @param idCards
 * @return
 * @throws Exception
 */
public List<String> findEmployeeOutIdByIdCards(List<Haulemployeeout> haulemployeeoutList)
throws Exception;

/**
 * 根据员工身份证号和大修 ID 删除一次大修中的员工信息

```

```

    * @param condition
    * @return
    * @throws Exception
    */
    public boolean deleteHaulEmployeeOutInfoByCondition(Map<String,Object> condition) throws
    Exception;

    /**
     * 根据大修 ID 和身份证号修改外来单位员工的信息
     * @param condition
     * @return
     * @throws Exception
     */
    public boolean updateHaulEmployeeOutInfoByCondition(Map<String,Object> condition) throws
    Exception;

    /**
     * 初始化外来单位员工分配表
     * @param distributeInfoList
     * @return
     * @throws Exception
     */
    public int addEmpOutDistributeInfoList(List<Employeeoutdistribute> distributeInfoList) throws
    Exception;

    /**
     * 根据大修员工 ID 集合和状态码修改大修员工表中的培训状态字段
     * trainStatus bigEmployeeOutIds
     * @param condition
     * @return
     * @throws Exception
     */
    public int updateHaulEmployeeOutTrainStatusByCondition(Map<String,Object> condition) throws
    Exception;

```

2) BreakrulesService (个人违章信息 service 接口)

```

    /**
     * 添加违章信息
     * @param breakRules 违章信息的 javabean
     * @return 是否添加成功

```

```
*/  
public int addBreakrules(Breakrules breakRules);  
  
/**  
 * 修改违章信息 (根据 javabean 修改违章信息)  
 * @param breakRules 修改后的违章信息  
 * @return 是否修改成功  
 */  
public int updateBreakrules(Breakrules breakRules);  
  
/**  
 * 删除违章信息(根据违章信息 id 删除)  
 * @param breakId 违章信息 id  
 * @return 是否删除成功  
 */  
public int delBreakrules(int breakId);
```

5.2.6 检修项目管理

HaulinfoService（检修信息 service）关键方法

```
/**  
 * 增加大修基本信息  
 * @param haulInfo  
 * @param projectList 工程名字  
 * @return  
 * @throws SQLException  
 */  
public boolean addHaulinfo(Haulinfo haulInfo,List<String> projectList) throws SQLException;  
  
/**  
 * 根据大修 ID 删除大修  
 * @param haulId  
 * @return  
 * @throws SQLException  
 * @throws Exception  
 */  
public boolean deleteHaulinfoByHaulId(String haulId) throws SQLException, Exception;
```



```
/**
 * 根据大修 id 查询大修信息
 * @param haulId
 * @return
 * @throws SQLException
 */
public Haulinfo getHaulinfoByHaulId(String haulId) throws SQLException;

/**
 * 分页查询大修信息
 * @param currentPage 当前页
 * @param currentCount 页大小
 * @param condition 组合条件
 * @return
 * @throws SQLException
 */
public PageBean<Map<String, Object>> getHaulinfoPageByCondition(int currentPage, int
currentCount, Map<String, Object> condition) t;

/**
 * 查询大修的基本信息与参加大修的部门的信息
 * @param haulId 大修 ID
 * @return map 中包含大修基本信息与参加大修的部门的信息
 * @throws SQLException
 */
public Map<String, Object> getHaulinfoWithUnitInfo(String haulId) throws SQLException;

/**
 * 查询大修和部门的树结构
 * @return
 * @throws SQLException
 */
public List<Map<String, Object>> getDepartmentAndOverHaulTree() throws SQLException;

/**
 * 根据大修 ID 查询所有的大修部门 ID
 * @param haulId
 * @return
 * @throws SQLException
 */
```

```
public List<String> getHaulUnitByHaulid(String haulId) throws SQLException;

/**
 * 查询未结束的大修的名字与 ID
 * @return
 * @throws SQLException
 */
public List<Map<String, Object>> getHaulNameAndIdsForExam(String departmentId) throws
SQLException;
/**
 * 根据大修 ID 查找标段信息
 * @return
 * @throws SQLException
 */
public PageBean<Map<String, Object>> getProjectInfoByBigId(int currentPage,int
currentCount,Map<String, Object> condition) throws SQLException;
/**
 * 根据大修 ID 查找标段、单位、人数信息
 * @return
 * @throws SQLException
 */
public PageBean<Map<String, Object>> getProjectUnitPerNumInfoByBigId(int currentPage,int
currentCount,Map<String, Object> condition)throws SQLException;
/**
 * 增加单个标段
 * @param bigId
 * @param projectname
 * @return
 * @throws SQLException
 */
public boolean addOnebiaoduan(String bigId,String projectname)throws SQLException;
/**
 * 修改单个标段
 * @param bigId
 * @param projectname
 * @return
 * @throws SQLException
 */
public boolean updateOnebiaoduan(String projectId,String projectname)throws SQLException;
/**
```

```

* 获取所有正在进行的检修的统计信息
* @param condition 过滤掉的检修 ID，一个是内部员工培训，一个是常委培训
* @return
* @throws SQLException
*/
public Map<String,Object> getAllHaulInfo(Map<String,Object> condition)throws SQLException;

```

5.2.7 黑名单管理

1) BlackUnitService（黑名单单位 service）关键方法

```

/**
* 添加单位黑名单
* @param BlackUnit
* @return
* @throws SQLException
*/
public boolean addBlackUnit(BlackUnit blackUnit)throws SQLException;
/**
* 修改黑名单单位
* @param blackUnit
* @return
* @throws SQLException
*/
public boolean updateBlackUnitByBlackId(BlackUnit blackUnit)throws SQLException;
/**
* 分页查询黑名单单位
* @param currentPage
* @param currentCount
* @return
* @throws SQLException
*/
public PageBean<BlackUnit> getBlackUnitPage(int currentPage,int currentCount)throws
SQLException;

```

2) BlackListEmpOutService（黑名单人员 service）关键方法

```

/**
* 分页查询黑名单人员信息
* @param currentPage
* @param currentCount
* @return

```

```

    * @throws SQLException
    */
    public PageBean<Map<String,Object>> getBlackEmployeePage(int currentPage,int
    currentCount)throws SQLException;

    /**
     * 根据黑名单 ID 删除黑名单信息
     * @param blackId
     * @return
     * @throws SQLException
     */
    public boolean deleteBlackListInfoById(String blackId) throws SQLException;

    /**
     * 根据员工 ID 和员工类型查询违章信息
     * @param employeeId
     * @param employeeType
     * @return
     * @throws SQLException
     */
    public Map<String,Object> getBreakRulesInfoList(String employeeId,String employeeType)
    throws SQLException;

```

5.3 人机对话设计

人机对话设计本质上来讲就是人与计算机进行信息的交流，它也称为用户界面（user interface，UI）设计。

5.3.1 人机对话设计的原则

人机对话设计应从用户的角度出发来进行相关界面操作的设计，着眼于用户，多为用户考虑，与用户充分进行交流和沟通。设计电厂安全培训界面时应考虑以下几个方面：

- 1) 操作用词应符合业务需求，及时征询用户意见。
- 2) 操作方式应尽量简单明了，考虑不同水平用户的用户体验。

3) 对用户的失误性操作提示应该友善，简洁清楚。

5.3.2 系统界面设计

遵循上述的基本原则和注意事项，设计出系统的人机交互界面。

1) 部门管理界面



图 5.7 部门管理界面

2) 员工管理界面



图 5.8 员工管理界面

3) 检修管理界面

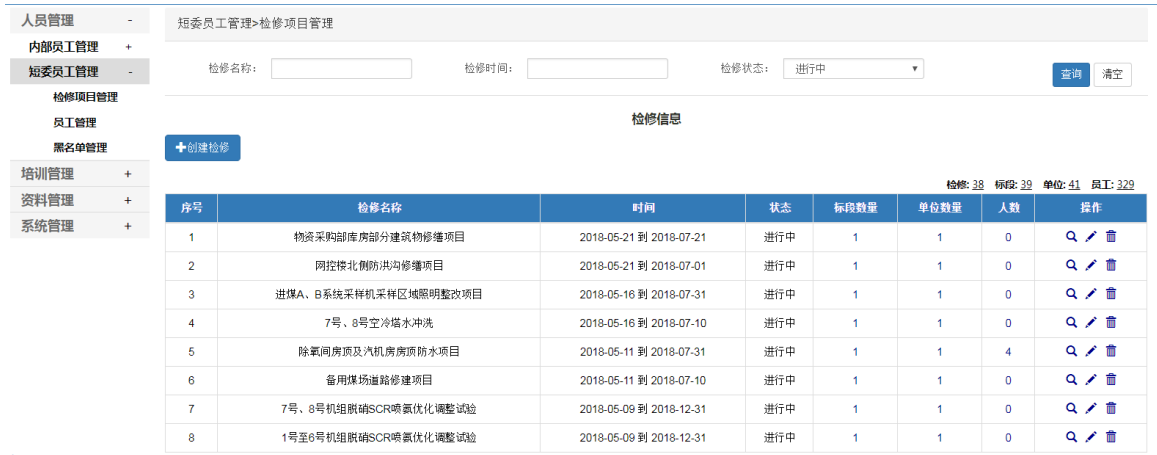


图 5.9 检修管理界面

4) 短委员工管理界面



图 5.10 短委员工管理界面

5) 黑名单管理界面



图 5.11 黑名单管理界面

6) 题库管理界面



图 5.12 题库管理界面

7) 试题管理界面



图 5.13 试题管理界面

8) 培训内容管理界面



图 5.14 培训内容管理界

5.4 本章小结

本章首先对电厂安全培训系统的数据库设计进行了详细的介绍,包括 CDM、PDM、视图和存储过程等的设计;紧接着对系统后台的 service 层接口做了详尽的描述,然后设计了电厂培训系统的主要模块界面,基本上完成了开发前期的所有准备工作,为接下来的编码实现打下了基础。

第六章 系统编码和实现

6.1 题库管理模块

本节主要介绍题库管理模块的实现页面以及功能实现的关键代码。

6.1.1 题库管理功能实现界面

题库批量导入模版如图 6.1 所示

单选题模板					
题干	第一个选项	第二个选项	第三个选项	第四个选项	答案
弧光中的红外线可造成对人眼睛的伤害,引起()	畏光	眼睛流泪	白内障	电光性眼炎	3
国家标准规定,工作企业噪声不应超过()	50dB	85dB	85dB	120dB	2
焊接场地应保持必要的通道,且车辆通道宽度	1m	2m	3m	5m	3
焊接场地应保持必要的通道,且人行通道					
宽度不小于()。	1m	1.5m	3m	5m	2
焊工应有足够的作业面积,一般不小于()	2m	4m	6m	8m	2
工作场地要有良好的自然采光或局部照明,以保证工作面照度达()lx	30-50	50-100	100-150	150-200	2
焊接场地周围()范围内,各类易燃易爆					
物品应清理干净。	3m	5m	10m	15m	3
用于紧固装配零件的是()。	夹紧工具	压紧工具	拉紧工具	撑具	1
扩大或撑紧装配零件的工具是()。	夹紧工具	压紧工具	拉紧工具	撑具	4
将所装配的零件的边缘拉到规定的尺寸应					
该用()。	夹紧工具	压紧工具	拉紧工具	撑具	3
焊条的直径是以()来表示的。	焊条直径	焊条外径	药皮厚度	焊芯直径和、药皮厚度之和	1
要求韧性好,冲击初度高的焊缝应选用()焊条。	酸性	碱性	不锈钢	铸铁	2
碱性焊条的主要优点之一是()。	脱渣性好	电弧稳定期	对偏长无要求	良好的抗裂性	4
根据GB/T1557-1995,碳钢焊条型号是以指					
戴金属的力学性能,()、药皮类型和焊	化学成份	焊接位置	扩散氢含量	抗裂性能	2
接电流种类来划分的。					
"E4303"是碳钢焊条型号完整的表示方	药皮类型	电流类型	焊接位置	化学成份	3
法,其中第二位阿拉伯数字表示的是()					

图 6.1 题库批量导入模版

题库批量导入试题功能的实现页面如图 6.2 所示

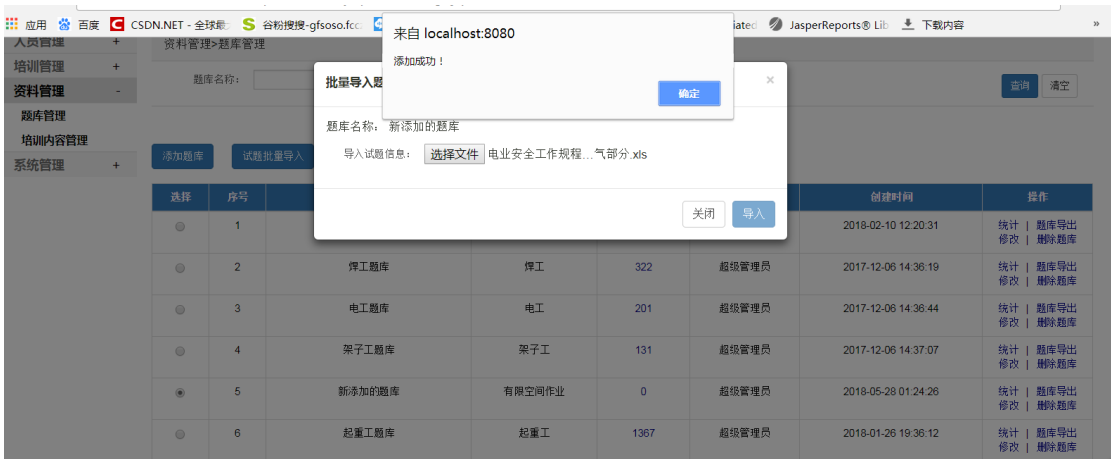


图 6.2 题库批量导入试题功能页面

6.1.2 类结构图

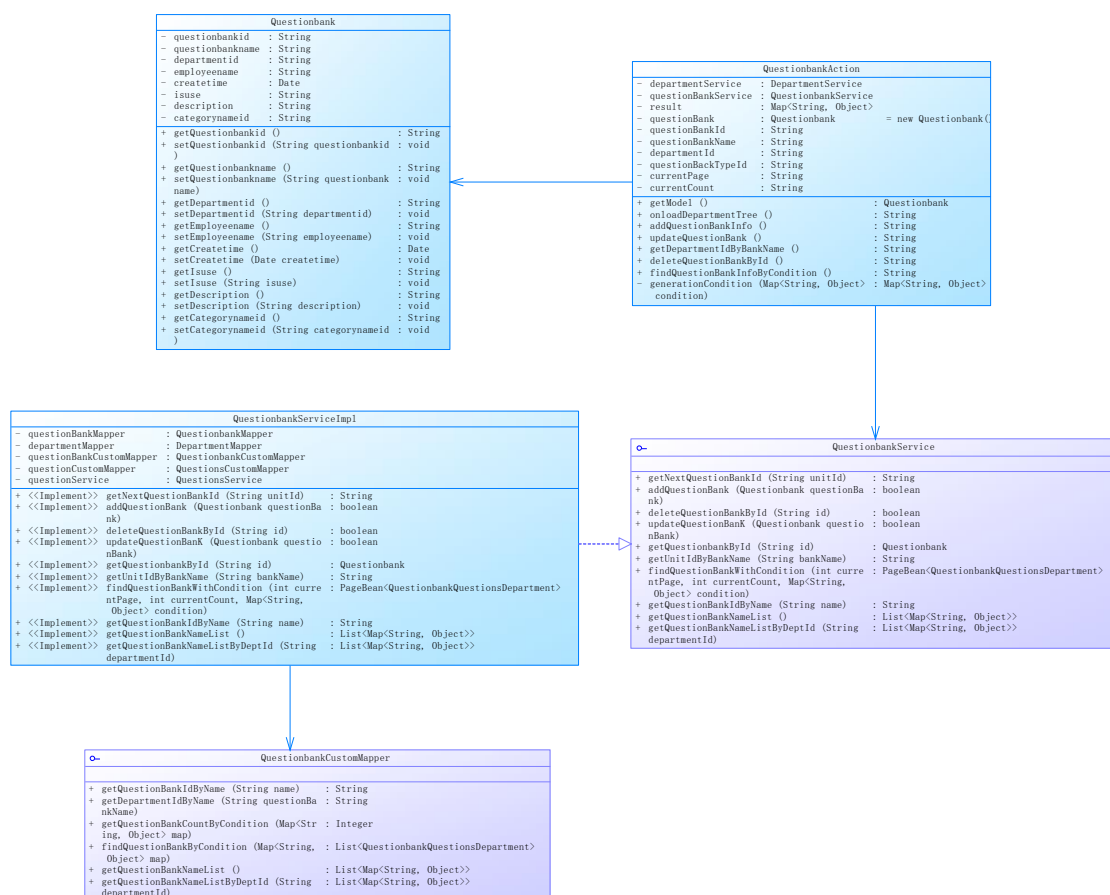


图 6.3 题库管理功能类结构图

6.1.3 关键代码

```

/**
 * 批量添加试题
 * @param list 试题集合，一个 map 封装一道题的试题 questions 与选项 options
 * @return 导入的试题数量
 * @throws Exception
 */
public int saveQuestionBatch(List<HashMap<String, Object>> list) throws Exception {
    for (HashMap<String, Object> hashMap : list) {
        Questions question = (Questions) hashMap.get("question");
        @SuppressWarnings("unchecked")
        List<Options> options = (List<Options>) hashMap.get("options");
    }
}

```

保存 关闭

保存 关闭

6.2.2 类结构图

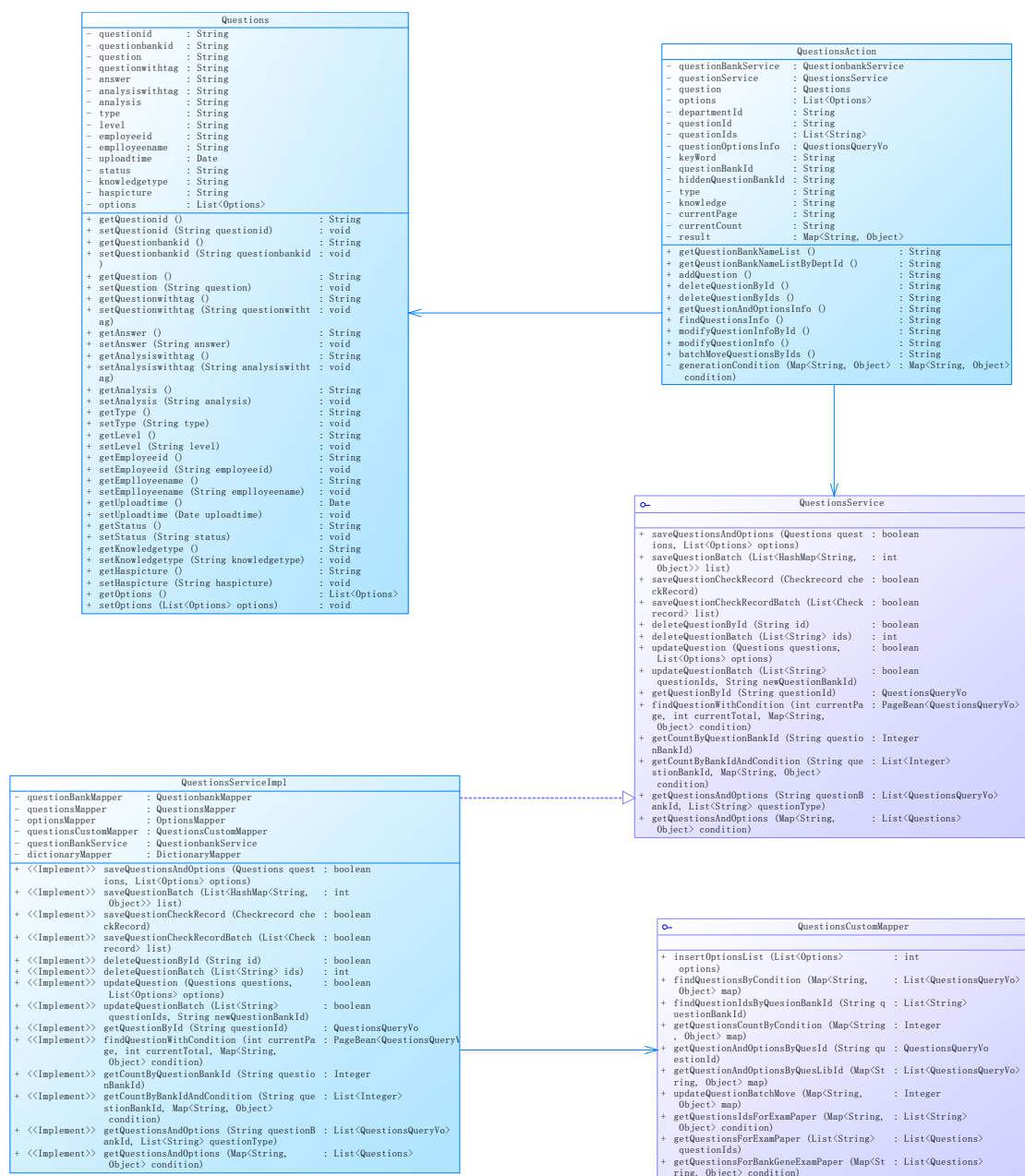


图 6.6 试题管理类结构图

6.2.3 关键代码

//点击修改试题的保存按钮执行的操作
function modify_question(){

```
if(isNotNull.form()){
    var optionsIsNotNull = ;
    //定义一个数组,存放试题答案
    var answer = [];
    var obj = document.getElementsByName("question_answer");
    for(var i=0; i<obj.length; i++){
        if(obj[i].checked){
            answer.push(i+1)
        }
    }
    var optionListObj = $(".input_wenbk");
    //判断当长度为 0 时证明是判断题, 不做处理
    if(optionListObj.length==0){
        optionsIsNotNull = true;
    }
    //拼接 options 的<input>标签信息
    var optionListInfo = "";
    for(var i=0; i<optionListObj.length; i++){
        if(optionListObj.eq(i).val().length>0){
            var j = i+1;
            optionListInfo += "<input name='options["+i+"]' optioncontent' type='hidden'
value='"+optionListObj.eq(i).val()+">"
            + "<input name='options["+i+"]' optionsequence' type='hidden' value='"+j+">";
            optionsIsNotNull = true;
        }else{
            alert("第"+(i+1)+"个选项不能为空! ");
            optionsIsNotNull = ;
            optionListInfo = "";
            break;
        }
    }
    if(optionsIsNotNull){
        //判断当存储答案的数组中有值时进行拼接操作
        if(answer.length>0){
            //拼接试题答案
            optionListInfo += "<input name='question.answer' type='hidden'
value='"+answer+">";
        }
        $("#updateOptions_content").append(optionListInfo);
    }
}
```

6.3 培训内容管理模块

本节主要介绍培训内容管理模块的实现页面以及功能实现的关键代码。

6.3.1 培训内容管理功能实现界面

培训内容添加的功能的实现界面如图 6.7，6.8 所示

人员管理

+

培训管理

+

资料管理

+

系统管理

+

培训管理>培训内容管理>添加培训资料

添加培训资料

资料名称:

选择文件:

选择文件

未选择任何文件

上传部门:

阳城电厂

资料类型:

mp4

资料类别:

YJ

资料说明:

上传人:

超级管理员

上传时间:

2018-06-19 22:20:56

保存

取消

图 6.7 培训内容添加界面（1）

人员管理

+

培训管理

+

资料管理

+

系统管理

+

培训管理>培训内容管理>添加培训资料

添加培训资料

资料名称:

电厂安全培训视频测试上传

选择文件:

选择文件

2f23b1f24...0df4.mp4

上传部门:

阳城电厂

资料类型:

mp4

资料类别:

YJ

资料说明:

电厂安全培训视频测试上传

上传人:

超级管理员

上传时间:

2018-06-19 22:19:05

培训资料正在上传，请稍后...

图 6.8 培训内容添加界面（2）

6.3.2 类结构图

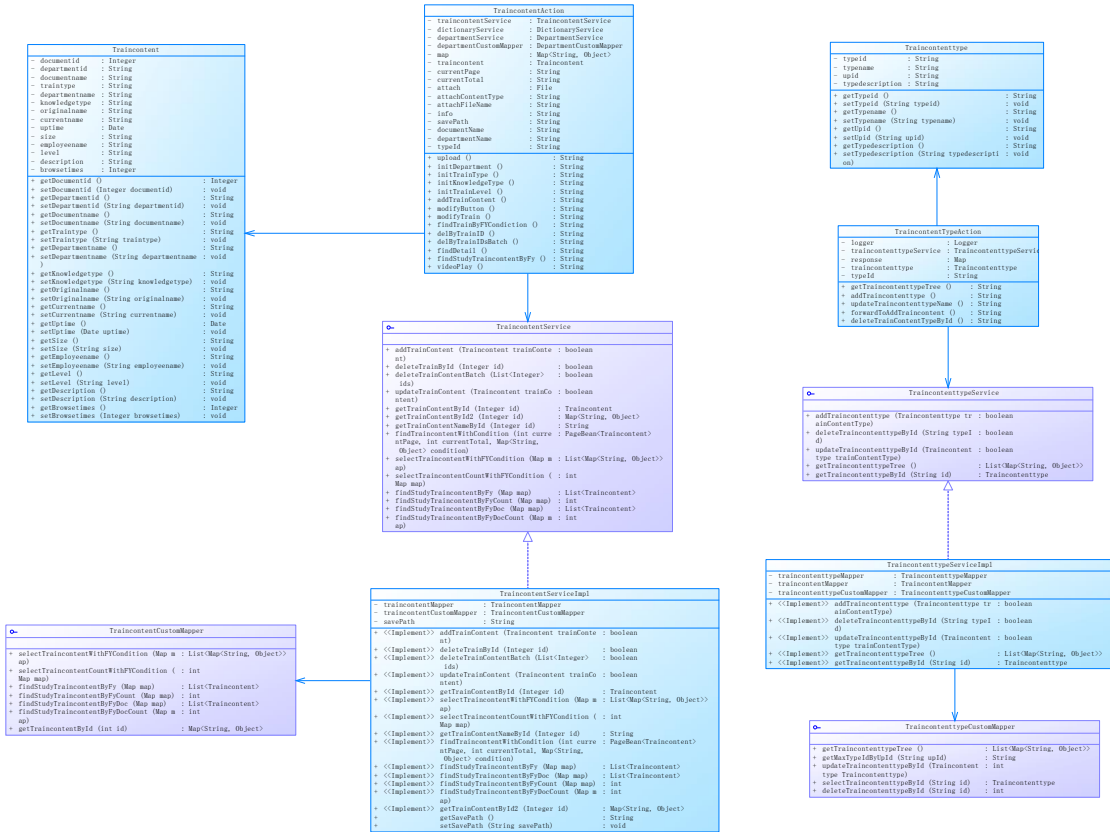


图 6.9 培训内容管理类结构图

6.3.3 关键代码

```

/**
 * 增加培训资料
 * @return
 * @throws Exception
 */
public String addTrainContent() throws Exception {
    // 判断文件是否存在，也就是判断用户是否上传了文件
    if (attach != null) {
        // 获取上传过来的文件的后缀名
        int lastIndexOf = attachFileName.lastIndexOf('.');
        String endName = attachFileName.substring(lastIndexOf); // 获取文件后缀名 .mp4
        if (bufferSize < 1024) {

```

```
// 文件大小小于 1KB 的，单位则用字节表示
size = String.valueOf(bufferSize) + "Byte";
} else if (bufferSize < 1048576) {
// 文件大小小于 1MB 的，单位则用 KB 表示
size = String.valueOf(bufferSize / 1024) + "KB";
} else {
// 文件大小大于 1MB 的，单位则用 MB 表示
double buSize = bufferSize;
double sizeMB = buSize / 1024 / 1024;
DecimalFormat df = new DecimalFormat("#####0.00");
size = df.format(sizeMB) + "MB";
}
traincontent.setSize(size);// 将文件大小保存在 javabean 中
// 将上传过来的文件保存在指定的硬盘目录中
// attachFileName 原文件名 currentName 当前文件名
FileUtils.copyFile(attach, new File(savePath + currentName));
// 将培训资料保存在培训资料表中
flag = traincontentService.addTrainContent(traincontent);}
return "ok";}
```

6.4 内部部门管理模块

本节主要介绍内部部门管理模块的实现页面以及功能实现的关键代码。

6.4.1 内部部门管理功能实现界面

部门统计功能的实现界面如图 6.10 所示



图 6.10 部门统计功能界面

6.4.2 类结构图

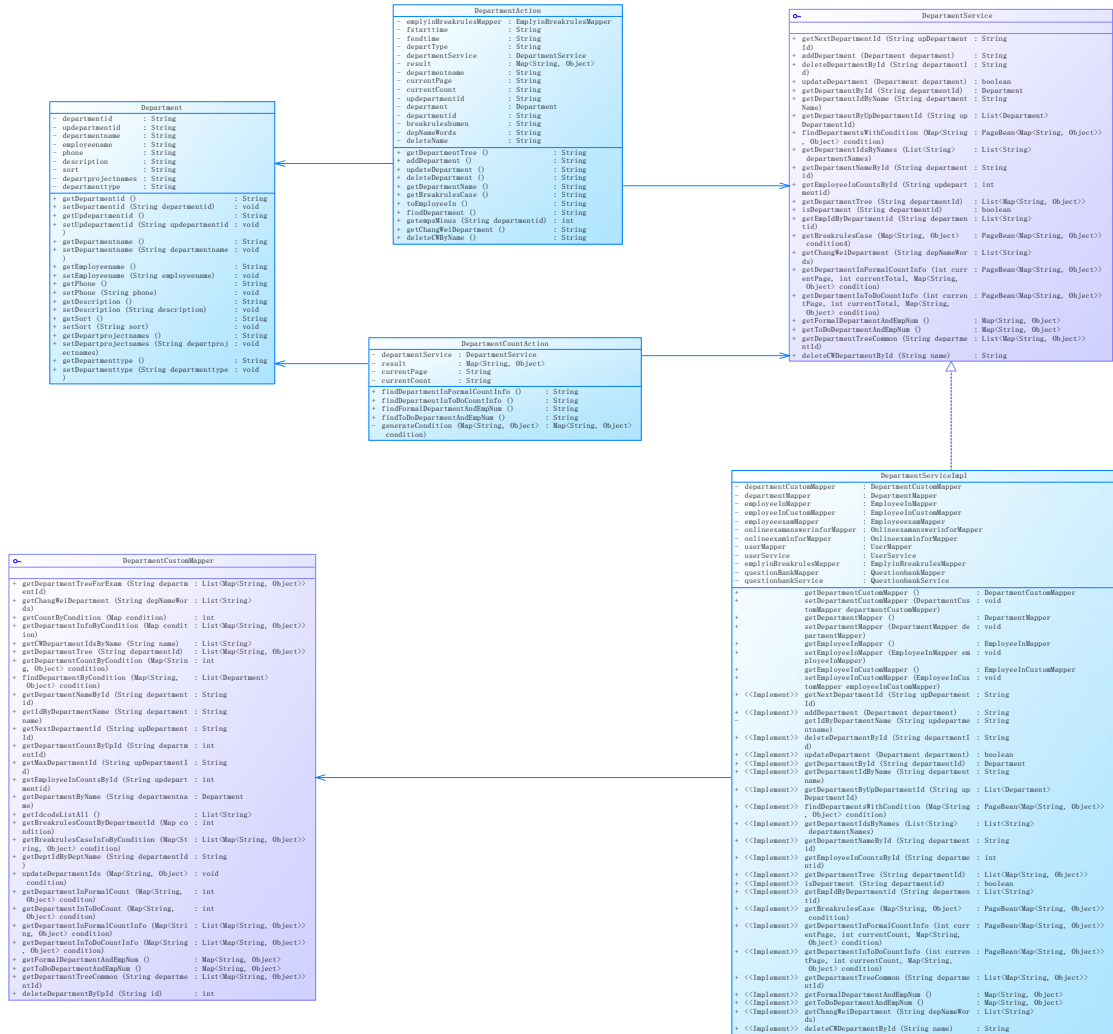


图 6.11 内部部门管理类结构图

6.4.3 关键代码

```
<!-- 查询内部正式单位和员工数 -->
```

```
<select id="getFormalDepartmentAndEmpNum" resultType="map">
```

```
    SELECT count(departmentId) departmentNum,sum(employeeNum) employeeNum
```

```
    FROM (
```

```
        SELECT
```

```
            departmentId,
```

```
            departmentName,
```

```
            employeeName,
```

```
phone,
(
    SELECT
        COUNT(*)
    FROM
        employee_in empin,
        department dep
    WHERE
        empin.departmentId LIKE CONCAT(department.departmentId, "%")
        AND empin.departmentId = dep.departmentId
        AND dep.departmentType = "0"
    ) employeeNum
FROM
    department
WHERE
    departmentType = "0"
    AND upDepartmentId = "01"
)p
</select>
```

6.5 检修项目管理

本节主要介绍检修项目管理模块的实现页面以及功能实现的关键代码。

6.5.1 检修项目管理功能实现界面

添加检修功能的实现界面如图 6.12 所示

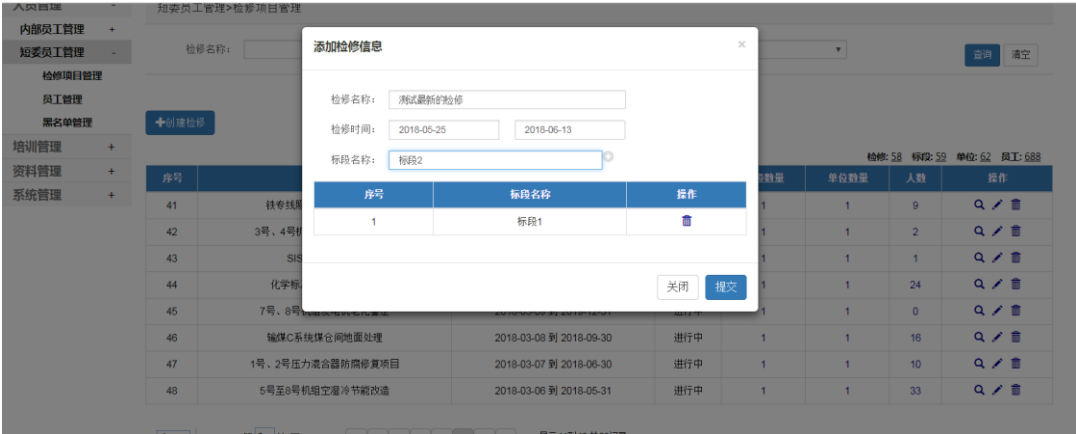


图 6.12 添加检修功能界面

80

6.5.2 类结构图

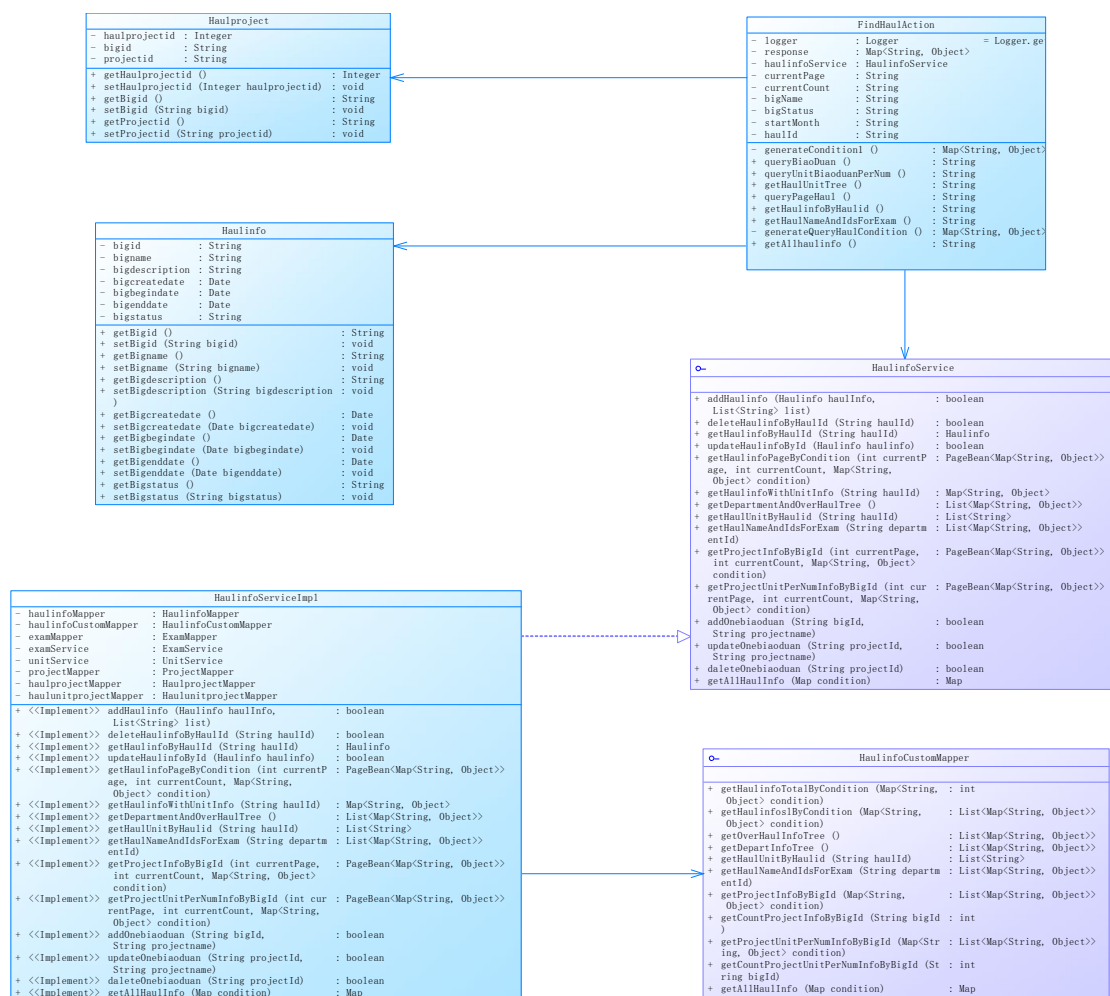


图 6.13 检修项目管理类结构图

6.5.3 关键代码

```

public String execute() {
    response = new HashMap<String, Object>();
    String result = "";
    try {
        // 将标段名称拆分为集合
        List list = null;
        if (ValidateCheck.isNull(hidden_biaoduan)) {
            list = Arrays.asList(hidden_biaoduan.split(","));
        }
    }
}

```

```
result = haulinfoService.addHaulinfo(haulinfo,list) ? "添加成功!": "添加失败!";
} catch (SQLException e) {
    logger.error("添加大修出错", e);
}
response.put("result", result);
return SUCCESS;
}
```

6.6 员工管理模块

本节主要介绍员工管理模块的实现页面以及功能实现的关键代码。

6.6.1 员工管理功能实现界面

添加员工功能的实现界面如图 6.14,6.15 所示



图 6.14 添加员工功能界面（1）



图 6.15 添加员工功能界面（2）

6.6.2 类结构图

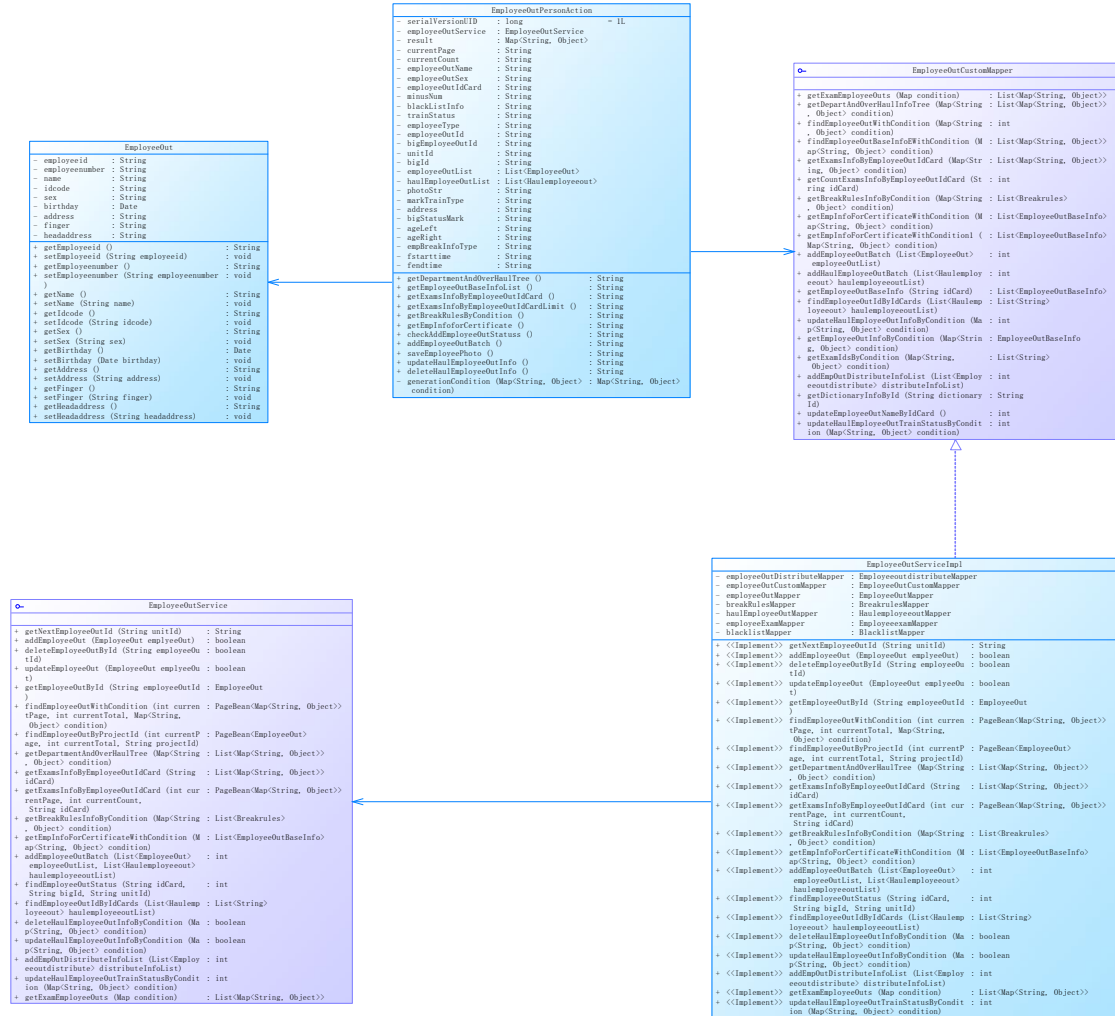


图 6.16 短委员工管理类结构图

6.6.3 关键代码

//批量导入外来单位员工的基本信息

```
public String addEmployeeOutBatch(){
    result = new HashMap<String,Object>();
    try {
```

//员工信息有可能为空

```
if(employeeOutList !=null&&employeeOutList.size()>0){
```

```
    employeeOutList.removeAll(Collections.singleton(null));
```

```
//设置员工的 id 和头像路径
```

```

        for (EmployeeOut employeeOut : employeeOutList) {
            employeeOut.setEmployeeid(UUIDUtil.getUUID2());

employeeOut.setHeadaddress("/employeeOutPhotos/"+employeeOut.getIdcode()+".jpg");
        }}
//去除集合中的所有 null 元素
haulEmployeeOutList.removeAll(Collections.singleton(null));
//设置员工大修 ID 和培训状态
for (HaulEmployeeOut haulEmployeeOut : haulEmployeeOutList) {
    haulEmployeeOut.setBigemployeeoutid(UUIDUtil.getUUID2());
    haulEmployeeOut.setTrainstatus("0");
}
//批量导入外来单位员工的基本信息
int count =
employeeOutService.addEmployeeOutBatch(employeeOutList,haulEmployeeOutList);
    if(count==haulEmployeeOutList.size()){
        result.put("result", "添加成功! ");
    }else{
        result.put("result", "添加失败! ");
    }
} catch (Exception e) {
    e.printStackTrace();
}
return SUCCESS;
}

```

6.7 黑名单管理

本节主要介绍黑名单管理模块的实现页面以及功能实现的关键代码。

6.7.1 黑名单管理功能实现界面

黑名单员工查询功能的实现界面如图 6.17 所示

人员管理

内部员工管理

短委员工管理

检修项目管理

员工管理

黑名单管理

培训管理

资料管理

系统管理

人员管理>短委员工管理>黑名单管理

黑名单单位管理

黑名单人员管理

序号	姓名	性别	员工类型	拉黑时间	违法积分	身份证号	操作
1	吕晋东	男	短委	2018-05-02	12	140522197503031019	删除
2	郭建云	男	短委	2018-04-20	12	132521197904010810	删除
3	赵金锋	男	短委	2018-04-20	12	130721198005052615	删除
4	刘红昌	男	短委	2018-03-31	12	130425198711023052	删除
5	宋付喜	男	短委	2018-03-27	12	410521198302043535	删除

8 第 1 页 共 1 页 显示 1 到 5 共 5 记录

图 6.17 黑名单员工查询功能界面

6.7.2 类结构图

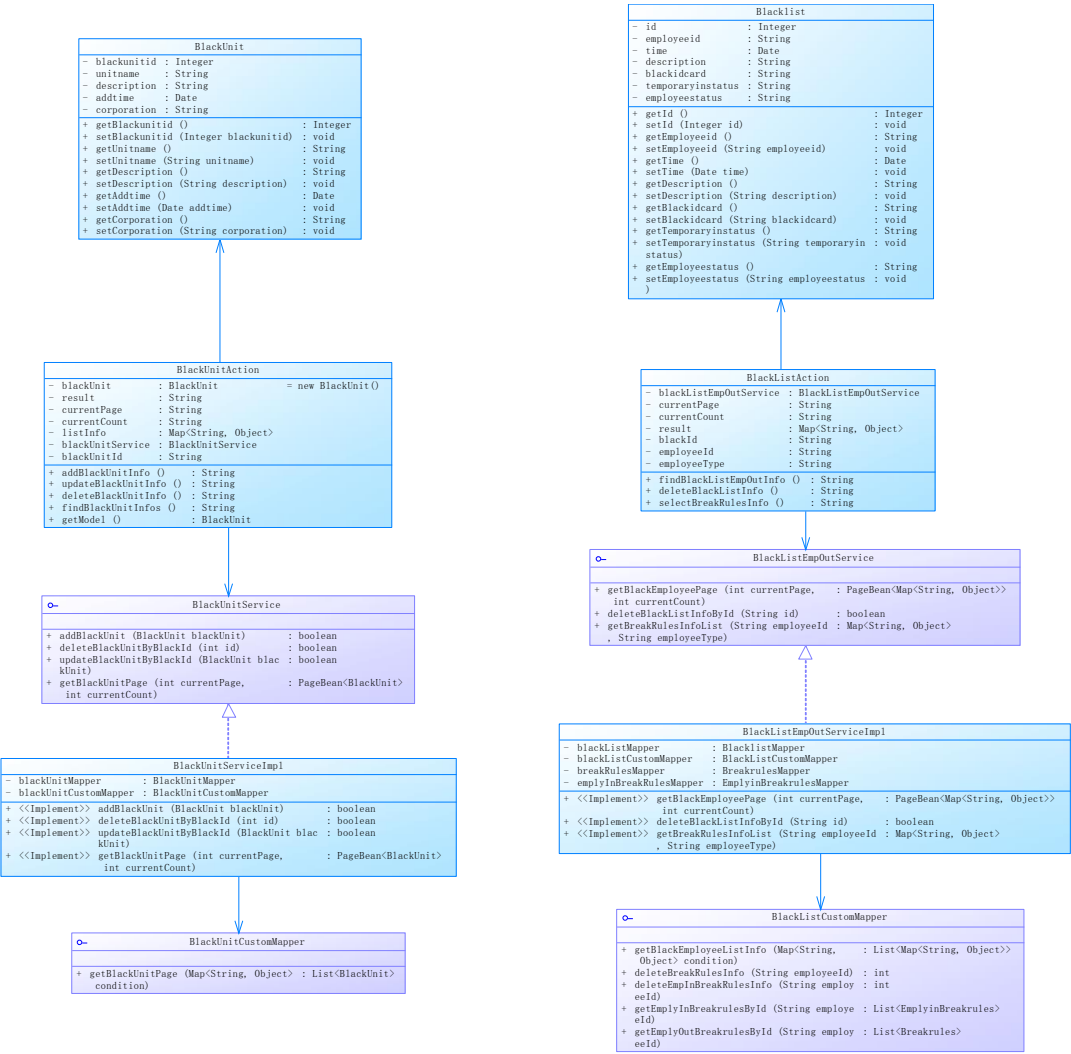


图 6.18 黑名单管理类结构图

6.7.3 关键代码

```
<!-- 查询黑名单员工信息分页显示 -->
<select id="getBlackEmployeeListInfo" parameterType="map" resultType="map">
    SELECT
        blacklist.*, CASE employeeStatus = '1'
        WHEN TRUE THEN
            (SELECT `name` FROM employee_out WHERE blackIdcard = idCode)
        ELSE
            (SELECT `name` FROM employee_in WHERE blackIdcard = idCode)
        END AS `name`,
        CASE employeeStatus = '1'
        WHEN TRUE THEN
            (SELECT `sex` FROM employee_out WHERE blackIdcard = idCode)
        ELSE
            (SELECT `sex` FROM employee_in WHERE blackIdcard = idCode)
        END AS `sex`
    FROM
        `blacklist`
    <where>
        <include refid="query_blackListInfo_where"></include>
    </where>
    ORDER BY time DESC
    <include refid="query_blackList_limit"></include>
</select>
```

6.8 本章小结

本章主要是对人员管理和资料管理模块的相关功能的具体编码实现和功能界面的展示，通过本章可以对电厂安全培训管理系统的人员管理和资料管理模块的相关功能有更好的了解和认识。

第七章 系统测试

系统测试的目的就是在系统软件在投放到市场上之前，对系统可能出现的错误进行排查删选，尽可能的降低系统运行中存在的安全隐患，对软件的功能做最后的审核。

7.1 测试意义

系统测试可以降低软件程序的出错率，因此进行系统测试时十分重要的，是保证软件质量的关键步骤。但是，由于种种原因，软件测试完后也不一定会不存在错误。而有效地测试，可以发现并修改一些较为严重的错误，提高软件的质量，确保此次系统对客户需求的满足程度，以及系统在使用中的流畅性、容错性、可靠性。

7.2 测试环境

系统环境:Win 7 服务器端: Tomcat 7 浏览器: 谷歌浏览器 65.0

7.3 功能模块测试

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电厂安全培训管理的核心模块如题库试题模块，短委员工管理模块等进行测试。下面以题库管理的统计导出和培训档案管理为例进行详细介绍。

7.3.1 题库统计功能测试

点击题库统计，查看题库基本信息以及各类题型所占的比重，数量与数据库中对应的试题数量是否一致。测试结果如图 7.1 所示。



图 7.1 题库统计功能测试图

7.3.2 题库导出功能测试

选择导出格式为 Word 导出题型包括单选题和判断题进行导出。测试结果如图 7.2，7.3 所示



图 7.2 题库导出功能测试图

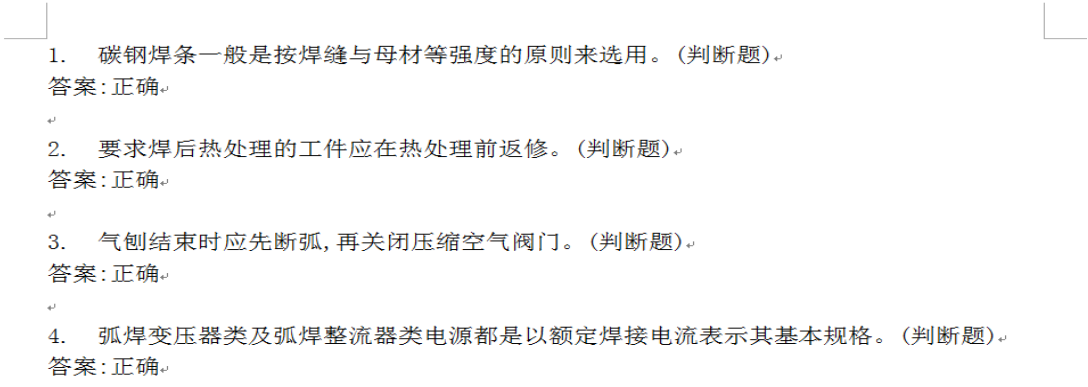


图 7.3 题库导出的题库信息截图

7.3.3 员工培训档案查看导出功能测试

选择一个员工点击查看培训档案并点击导出培训档案测试结果如下图 7.3，7.4 所示：



图 7.3 员工培训档案查看测试图



图 7.4 员工培训档案导出功能测试图

7.4 本章小结

本章主要是对电厂安全培训管理系统相关模块的核心功能进行测试，排查培训管理系统中存在的隐患和问题，为系统上线做最后的准备。

结束语

本文通过对用户需求的分析，确定了系统的功能模块。同时，通过数据库概念设计和数据库逻辑设计，确定了数据库各个实体的属性以及实体间的联系。使用 Java 常用开源框架和 MySQL 数据库，实现了后台业务逻辑功能的实现；使用 Bootstrap 框架和 CSS 等前端技术，实现了前端页面的搭建，两者结合，完成了系统设计与开发。经过测试后该系统也正式上线开始试运行，作为自己第一个实际上线的项目，看着自己开发的系统真正的为用户服务，还是十分激动的。

在这次开发中我学到许多新的知识，比如前台 zTree 树的初始化，KindEditors 的同步和校验问题，后台 Jsoup 工具包处理 HTML 标签，POI 的相关操作等。吸取上次项目手工整合代码的教训，使用 Git 进行源码管理，切实体会到了 Git 的强大，提高了自己的效率。同时通过书写详细设计文档，帮助自己理清设计思路，方便以后的开发，还降低了自己的开发风险。另外我还体会到了需要从用户实际操作的角度出发去实现自己的功能，要多考虑实际因素，做真正能用的功能。

到目前为止，电厂安全培训管理系统基本上达到了前期预想的效果。当然，由于自己开发经验和技術等方面的不足，该系统还有一些待完善的地方，如代码重构和优化，公共方法抽取，SQL 查询优化等。这些问题还需在今后的系统更新和维护中进一步解决和完善。

致谢

大学四年生活即将结束，毕业设计是对我过去所学知识的综合性应用，一次检验自己大学四年的学习成果的机会。感谢我的导师谢斌红教授，您的启迪，让我有了设计思路，您的支持，让我有了不断前进的动力。您给予了我许多的帮助。不仅教导我理论上的知识，更是给了我许多来自精神上的鼓励。感谢您的教诲，因为您的辛勤付出，我的论文才能不断地完善。

感谢计算机学院的各位老师，大学四年中，我学到的专业知识离不开你们的悉心教导。是你们让我开启了一个新的人生目标和方向。

最后我还要感谢我的家人，我的成长离不开他们的鼓励和陪伴。正是因为他们的无私奉献和默默支持，我才能在遇到困难挫折时更加的从容淡定，冷静应对。

再次感谢所有帮助过我的人，谢谢大家！

参考文献

- [1] 孙卫琴. Tomcat 与 Java Web 开发技术详解.北京: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] 金敏,周翔. 高级软件开发过程.北京:清华大学出版社, 2005.
- [17] 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.
- [18] 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

英文原文

Computational thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone. Computational thinking confronts the riddle of machine intelligence: What can humans do better than computers? And What can computers do better than humans? Most fundamentally it addresses the question: What is computable? Today, we know only parts of the answers to such questions.

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the three Rs, what is appropriately incestuous about this vision is that computing and computers facilitate the spread of computational thinking.

Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science. Having to solve a particular problem, we might ask: How difficult is it to solve? and What's the best way to solve it? Computer science rests on solid theoretical underpinnings to answer such questions precisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our advantage, and whether false positives or false negatives are allowed. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation.

Computational thinking is thinking recursively. It is parallel processing. It is interpreting code as data and data as code. It is type checking as the generalization of dimensional analysis. It is recognizing both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect addressing and procedure call. It is judging a program not just for correctness and efficiency but for aesthetics, and a system's design for simplicity and elegance.

Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. It is

using invariants to describe a system's behavior succinctly and declaratively. It is having the confidence we can safely use, modify, and influence a large complex system without understanding its every detail. It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. modularizing something in anticipation of multiple users or prefetching and caching in anticipation of future use.

Computational thinking is thinking in terms of prevention, protection, and recovery from worst-case scenarios through redundancy, damage containment, and error correction. It is calling gridlock deadlock and contracts interfaces. It is learning to avoid race conditions when synchronizing meetings with one another.

Computational thinking is using heuristic reasoning to discover a solution. It is planning, learning, and scheduling in the presence of uncertainty. It is search, search, and more search, resulting in a list of Web pages, a strategy for winning a game, or a counterexample. Computational thinking is using massive amounts of data to speed up computation. It is making trade-offs between time and space and between processing power and storage capacity.

Consider these everyday examples: When your daughter goes to school in the morning, she puts in her backpack the things she needs for the day; that's prefetching and caching. When your son loses his mittens, you suggest he retrace his steps; that's backtracking. At what point do you stop renting skis and buy yourself a pair?; that's online algorithms. Which line do you stand in at the supermarket?; that's performance modeling for multi-server systems. Why does your telephone still work during a power outage?; that's independence of failure and redundancy in design. How do Completely Automated Public Turing Test(s) to Tell Computers and Humans Apart, or CAPTCHAs, authenticate humans?; that's exploiting the difficulty of solving hard AI problems to foil computing agents.

Computational thinking will have become ingrained in everyone's lives when words like algorithm and precondition are part of everyone's vocabulary; when nondeterminism and garbage collection take on the meanings used by computer scientists; and when trees are drawn upside down.

We have witnessed the influence of computational thinking on other disciplines. For example, machine learning has transformed statistics. Statistical learning is being used for problems on a scale, in terms of both data size and dimension, unimaginable only a few years ago. Statistics departments in all kinds of organizations are hiring computer scientists. Schools of computer science are embracing existing or starting up new statistics departments.

Computer scientists' recent interest in biology is driven by their belief that biologists can benefit from computational thinking. Computer science's contribution to biology goes beyond the ability to search through vast amounts of sequence data looking for patterns. The hope is that data structures and algorithms—our computational abstractions and methods—can represent the structure of proteins in ways that elucidate their function. Computational biology is changing the way biologists think. Similarly, computational game theory is changing the way economists think; nanocomputing, the way chemists think; and quantum computing, the way physicists think.

This kind of thinking will be part of the skill set of not only other scientists but of everyone else. Ubiquitous computing is to today as computational thinking is to tomorrow. Ubiquitous computing was yesterday's dream that became today's reality; computational thinking is tomorrow's reality.

Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction.

WHAT IT IS, AND ISN'T

Computer science is the study of computation— what can be computed and how to compute it. Computational thinking thus has the following characteristics: Conceptualizing, not programming. Computer science is not computer programming. Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction; *Fundamental, not rote skill*. A fundamental skill is something every human being must know to function in modern society. Rote means a mechanical routine. Ironically, not until computer science solves the AI Grand Challenge of making computers think like humans will thinking be rote; *A way that humans, not computers, think*. Computational thinking is a way humans solve problems; it is not trying to get humans to think like computers. Computers are dull and boring; humans are clever and imaginative. We humans make computers exciting. Equipped with computing devices, we use our cleverness to tackle problems we would not dare take on before the age of computing and build systems with functionality limited only by our imaginations; *Complements and combines mathematical and engineering thinking*. Computer science inherently draws on mathematical thinking, given that, like all sciences, its formal foundations rest on mathematics. Computer science inherently draws on engineering thinking, given that we build systems that interact with the real world. The constraints of the underlying computing device force computer scientists to think computationally, not just mathematically.

Being free to build virtual worlds enables us to engineer systems beyond the physical world; *Ideas, not artifacts*. It's not just the software and hardware artifacts we produce that will be physically present everywhere and touch our lives all the time, it will be the computational concepts we use to approach and solve problems, manage our daily lives, and communicate and interact with other people; and For everyone, everywhere. Computational thinking will be a reality when it is so integral to human endeavors it disappears as an explicit philosophy.

Many people equate computer science with computer programming. Some parents see only a narrow range of job opportunities for their children who major in computer science. Many people think the fundamental research in computer science is done and that only the engineering remains. Computational thinking is a grand vision to guide computer science educators, researchers, and practitioners as we act to change society's image of the field. We especially need to reach the pre-college audience, including teachers, parents, and students, sending them two main messages:

Intellectually challenging and engaging scientific problems remain to be understood and solved. The problem domain and solution domain are limited only by our own curiosity and creativity; and One can major in computer science and do anything. One can major in English or mathematics and go on to a multitude of different careers. Ditto computer science. One can major in computer science and go on to a career in medicine, law, business, politics, any type of science or engineering, and even the arts.

Professors of computer science should teach a course called “Ways to Think Like a Computer Scientist” to college freshmen, making it available to non-majors, not just to computer science majors. We should expose pre-college students to computational methods and models. Rather than bemoan the decline of interest in computer science or the decline in funding for research in computer science, we

should look to inspire the public's interest in the intellectual adventure of the field. We'll thus spread the joy, awe, and power of computer science, aiming to make computational thinking commonplace.

中文翻译

计算思维

计算机思维代表着一种普遍的认识和一类普适的技能，每一个人，不仅仅是计算机科学家，都应热心于它的学习和运用。

计算思维建立在计算过程的力量和限制之上，无论它们是由人还是由机器执行。计算方法和模型使我们有勇气解决问题和设计系统，我们没有人能够独自处理。计算思维面临机器智能的谜语：人类比电脑做得更好？电脑比人类做得更好？最根本的是它解决了一个问题：什么是可计算的？今天，我们只知道这些问题的部分答案。

计算思维是每个人的基本技能，不仅仅是计算机科学家。对于阅读，写作和算术，我们应该为每个孩子的分析能力增加计算思维。正如印刷机促进了三卢比的扩张一样，对于这一愿景来说，适当地令人不安的是，计算机和计算机便于计算思维的传播。

计算思维涉及解决问题，设计系统和理解人借鉴计算机科学基础的概念。计算思维包括一系列反映计算机科学领域广度的心理工具。

要解决一个特殊的问题，我们可能会问：解决难度多大？什么是最好的解决方法？计算机科学依赖于坚实的理论基础，以精确地回答这些问题。说明问题的难点解释了机器的基本功能 - 将运行解决方案的计算设备。我们必须考虑机器的指令集，其资源限制及其操作环境。

在有效解决问题时，我们可能会进一步询问一个近似解决方案是否足够好，我们是否可以使用随机化来获取优势，以及是否允许假阳性或假阴性。计算思维正在将一个看似困难的问题重新形成一个我们知道如何解决的问题，也许是通过减少，嵌入，转换或模拟来解决的。

计算思维是递归思考的。它是并行处理。它将代码解释为数据和数据作为代码。它是类型检查作为维度分析的泛化。它确认了别名的优点和危险，或者给予某人或某事不止一个名字。它承认间接寻址和程序调用的成本和功能。它正在判断一个程序，不仅仅是为了正确性和效率，而是美学，而且系统的设计简单而优雅。

计算思维在攻击大型复杂任务或设计大型复杂系统时正在使用抽象和分解。这是关切的分离。它正在为一个问题选择一个适当的代表，或者对问题的相关方面进行建模以使其易于处理。它使用不变量来简洁和声明地描述系统的行为。我们有信心我们可以安全地使用，修改和影响大型复杂系统，而无需了解其每一个细节。

它代表了一个普遍适用的态度和技能，每个人，而不仅仅是计算机科学家，将渴望学习和使用。模糊化预期多个用户或预期的预取和缓存的东西未来使用。计算思维正在考虑通过冗余，损害遏制和纠错等最坏情况的预防，保护和恢复。它正在调用网络锁死锁和接口。正在学习避免种族同步会议时的条件。

计算思维在使用启发式推理来发现解决方案。它正在规划，学习，并在存在不确定性的情况下进行调度。它是搜索，搜索和更多的搜索，产生一个网页列表，一个赢得一个游戏的策略，或一个反例。计算思维正在使用大量的数据来加速计算。它在时间和空间以及处理能力和存储容量之间进行权衡。

考虑这些日常的例子：当你女儿早上上学的时候，她把背包里的东西放在一起，这是预取和缓存。当你的儿子失去手套，你建议他回溯他的步骤；这是回溯。你什么时候停止租用滑雪板，买一双呢？那就是在线算法。你在超市里站哪条线？这是多服务器系统的性能建模。为什么你的电话在停电期间仍然工作？这是设计中的失败和冗余的独立性。全面自动化公共图灵测试如何告诉计算机和人类，或CAPTCHAs认证人类？这是利用解决硬AI问题来解决计算机代理的困难。

计算思维在每个人的生活中都会变得根深蒂固，像algo这样的词汇算术和前提条件是每个人的词汇的一部分；当非确定性和垃圾收集取决于计算机科学家所使用的含义时；当树被颠倒的时候。我们目睹了计算思维对其他学科的影响。例如，机器学习已经转变了统计数据。在数据大小和维度方面，统计学习正在大规模地用于问题，仅在几年前就难以想象。各类组织的统计部门正在聘请计算机科学家。计算机科学学院正在拥抱现有或启动新的统计部门。计算机科学家最近对生物学的兴趣是由于他们相信生物学家可以从计算思维中获益。计算机科学对生物学的贡献超出了搜索大量序列数据寻找模式的能力。希望数据结构和算法（我们的计算抽象和方法）能够以阐明其功能的方式来表示蛋白质的结构。计算生物学正在改变生物学家的想法。同样，计算博弈理论正在改变经济学家思考的方式；纳米计算，化学家的想法；和量子计算，物理学家认为的方式。这种思考将不仅是其他科学家，而且是其他人的技能组合的一部分。无处不在的计算是今天的计算思维是明天。无处不在的计算机是昨天的梦想，成为今天的现实；计算思维是明天的现实。

像计算机科学家一样思考意味着不仅仅能够对计算机进行编程。它需要思考多层次的抽象。它是不是计算机科学是计算的研究 - 可以计算什么以及如何计算它。计算思维因此具有以下特征：概念化，而不是编程。计算机科学不是电脑编程。像计算机科学家一样思考意味着不仅仅是能够对计算机进行编程。它需要思考多层次的抽象；基础，而不是技能。一个基本的技能就是每个人都必须知道的现代社会的功能。Rote表示机械程序。具有讽刺意味的是，直到电脑科学解决了AI大挑战，使得电脑思考像人类会思考的一样；人类而不是电脑思考的方式。计算思维是人类解决问题的一种方式；它不是想让人类像电脑一样思考。电脑沉闷无聊；人类是聪明而富有想象力的。我们人类让电脑激动人心。

配备了计算机设备，我们运用我们的聪明才智来解决我们不敢在计算时代之前所遇到的问题，并建立具有仅受我们想像力限制的功能的系统；完成并结合了数学和工程思维。计算机科学本质上依靠数学思维，因为像所有科学一样，它的正式基础依赖于数学。考虑到我们建立与现实世界相互作用的系统，计算机科学本质上依赖于工程思维。计算机设备的限制迫使计算机科学家在计算上思考，而不仅仅是数学上的。自由构建虚拟世界，使我们能够设计超越物理世界的系统；想法而不是工件。不仅仅是我们生产的软件和硬件工件将会随处可见，并且随时触及我们的生活，它将是我們用来处理和解决问题，管理我们的日常生活以及与其他人交流和交流的计算概念；对于大家，无处不在。

计算思维将成为一个现实，当它与人类的努力是不可分割的，它消失为明确的哲学。许多人将计算机科学与计算机编程相提并论。一些家长对于计算机科学专业的孩子来说，只看到很少的就业机会。许多人认为计算机科学的根本研究是完成的，只有工程学才能存在。

智力上极有挑战性并且引人入胜的科学问题依旧亟待理解和解决。这些问题的范围和解决方案的范围之唯一局限就是我们自己的好奇心和创造力；同时一个人可以主修计算机科学并且干什么都行。一个人可以主修英语或者数学，接着从事各种各样的职业。计算机科学也一样。一个人可以主修计算机科学，接着从事医学、法律、商业、政治，以及任何类型的科学和工程，甚至艺术工作。

计算机科学的教授应当为大学新生开一门称为“怎么像计算机科学家一样思维”的课，使其适用于非专业，而不仅仅是计算机专业。我们应该让大学前学生学习计算方法和模式。我们应当使大学之前的学生接触计算的方法和模型。我们应当设法激发公众对于计算机领域中的科学探索之兴趣，而不是悲叹对其兴趣的衰落或者哀泣其研究经费的下降。因此，我们应当传播计算机科学的趣味性、高尚性和潜藏的巨大能量，致力于计算思维的常识化。