

基于 Web 的企业排班管理系统设计与实现

林 野
(同济大学, 上海 200092)

摘 要: 随着我国企业的快速发展, 落后的排班方式正日益影响企业的工作效率。本文即探讨基于 Web 的企业排班管理。提出基于 SSH 开发框架并结合 MVC 设计模式来开发企业排班管理系统, 提高系统的安全性、易用性和可扩展性。

关键词: J2EE; MVC; Struts; Spring; Hibernate

中图分类号: TP302.1 **文献标识码:** A **文章编号:** 1003-9767 (2012) 09-0056-03

1. 引言

改革开放以来, 尤其在国际化、网络化、信息化趋势下, 我国企业快速成长, 我国经济高速发展。但是, 落后的信息管理水平, 制约着我国企业的发展。排班是许多企业的一项日常管理工作内容。一套公平、合理的排班系统不仅能调用员工积极性, 也能提高工作效率, 对企业具有非常重要意义。但目前很多企业都采用手工方式来完成排班, 费时费力也容易出错。在这种情况下, 基于 Web 排班管理系统应运而生。它利用计算机技术, 通过互联网, 将企业人员排班管理、人员信息查询等有机结合起来, 提高了企业员工的效率, 降低了企业的运营成本。本文就基于 Web 的企业排班管理系统设计与实现这一议题展开讨论和分析。

2. 基本需求分析

排班管理和用户管理是基于 Web 的企业排班管理系统的两个主要功能。

(1) 排班管理

排班管理是系统的主要功能, 也是最常用的功能。因此, 在设计该功能时最主要的考虑因素是易用性。系统应该至少提供列表和视图这两种展示方式来支持排班。列表方式展示信息量大, 而视图方式则以月或周为单位, 展示信息比较直观。

(2) 用户管理

用户管理是系统的基础模块, 主要用于对用户的基础信息进行管理。其中, 针对用户个人信息详细的特点, 需要设计良好的信息展示方式。同时, 由于用户管理中的角色管理和权限管理模块涉及到系统的安全性, 所以需要确保其稳定可靠。

由于系统是处于互联网这样的开放环境之中, 并且考虑到系统在以后会进行升级扩充, 所以除了以上的需求之外, 系统在安全性、可靠性、可扩展性和易操作性上都有很高的要求。这就需要在系统进行架构设计的时候仔细考虑。

3. 系统设计与实现

3.1 系统架构设计

系统基于 MVC (model-view-controller, 模型 - 视图 - 控制器) 模式构建, 采用经典的 SSH (struts + spring + hibernate) 框架。其框架结构如图 1 所示。

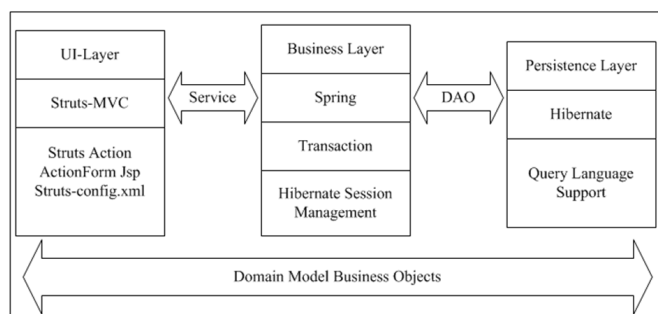


图 1 系统框架图

3.1.1 表示层

表示层主要采用 jQuery 技术实现, 并结合使用 Struts 提供的一系列页面标签。同时, 采用 CSS、Ajax 等技术手段来优化表示层, 完善应用系统与用户间的人机交互功能, 提升用户体验。

3.1.2 业务逻辑层

采用 spring 作为业务逻辑层框架, spring 是一个轻量级的容器框架。拥有控制反转 (IoC) 和面向切面 (AOP) 两大特性。它主要用来处理系统的业务逻辑和业务验证部分, 同时预留和其它层交互的接口, 管理从业务逻辑到实体层的实现, 管理业务层之间的依赖关系, 使表现层和实体层之间不直接发生联系, 增加它们的灵活性^[1]。

3.1.3 数据持久层

数据持久层采用 Hibernate 实现。通过把数据记录映射成 Java 对象, 充分利用了面向对象编程的优势, 对数据库的访问简单、高效和易于维护^[2]。大大简化了数据增、删、改、查等功能的开发过程, 同时为系统业务逻辑层提供服务。

3.2 系统安全设计

基于 Web 的系统会面临许多安全隐患。因为它运行在 Internet 之上, 所以可能会被黑客从外部入侵到企业的内部系统。同时也可能被不法分子冒充合法用户, 使用假冒的系统控制程序套取或修改使用权限、口令等信息, 然后, 利用这些信息登录系统, 进行自己的非法操作。或者也可能遇到非法访问的情况。用户不具备访问某些功能的权限, 却通过一些方法进行非法操作。这些问题都需要仔细考虑, 做好相关的防范。本系统通过完善的用户管理、日志管理以及多种方式的系统数据备份、恢复手段来实现对系统的保护。

3.2.1 用户管理

由于用户的角色不同, 所承担的职能也不同。因此, 为了确保系统的安全性和合理性, 需要给用户分配不同的权限。用户管理主要负责增加、删除用户, 修改用户密码并对用户进行分组, 对用户进行授权。权限主要分为单位权限和模块权限两种。

(1) 单位权限: 不同用户属于不同的单位 (部门), 只能操作该单位 (部门) 的数据, 对于某一用户可指定一个或多个操作单位, 该单位的下层部门数据同时可以进行操作。

(2) 模块权限: 分别对系统所需各项功能模块进行授权。如果该用户有此模块权限, 即能进入此模块; 如果无此模块权限, 即不能运行此模块。授权时, 应展开所有功能模块至最底层, 选中要授权的功能模块。

3.2.2 日志管理

系统日志是记录系统中软件和系统问题的信息, 同时还可以监视系统中发生的事件。用户可以通过它来检查错误发生的原因, 或者寻找受到攻击时攻击者留下的痕迹。为了增强系统的安全性, 需要追踪用户的行为, 即需要系统记录用户的敏感操作。为此, 系统提供了日志管理功能。日志记录了用户使用系统的一些基本信息, 记录用户的操作内容, 操作时间。系统管理员可以按操作时间、操作人员类别、操作类型等来查询系统日志, 并可将系统日志打印或输出。

3.2.3 身份认证和数据加密

不同的用户登录系统,需要对其身份进行认证,已确保此用户为合法用户。大多数系统采用用户名和密码进行身份认证。本系统也采用此方法。用户名和密码都存储在数据库中,当进行研究的时候通过用户名查找相关的密码进行比较即可。可是考虑到数据库数据可能泄露,因此,密码数据不能用明文存储,需要加密。

3.3 访问控制设计

访问控制是在身份认证基础上,依据授权对提出的资源访问请求加以控制。即限制已授权的用户、程序、进程或计算机网络中其他系统访问本系统资源的过程。访问控制技术是本系统的重要组成部分。主要用来防止非法的主体进入受保护的网路资源;允许合法用户访问受保护的网路资源;防止合法的用户对受保护的网路资源进行非授权的访问^[3]。

本系统采用基于角色的访问控制策略。同时,因为用户的操作很频繁,如果对于用户的每个动作,都去访问数据库的权限表,然后判断用户的权限。这种方式势必降低系统的性能。因此,在本系统中采用静态的访问权限。在用户会话开始时,将用户的权限信息都加载到内存中。对于以后用户的每次操作,都只需到内存中查找用户的权限即可,不再需要访问数据库。这种方式虽然在加载权限信息到内存中的这个阶段会需要较大的系统开销,但之后对于用户的每次操作,系统的开销就比较小。这样大大提高了系统的性能,同时保证了系统的安全性。

3.4 系统功能模块设计

3.4.1 排班管理

排班管理模块主要负责辅助用户进行排班信息管理,包括排班列表和排班视图两部分。排班列表实现添加排班、修改排班、删除排班;排班视图主要实现排班信息查看。排班管理模块功能划分如图2所示:

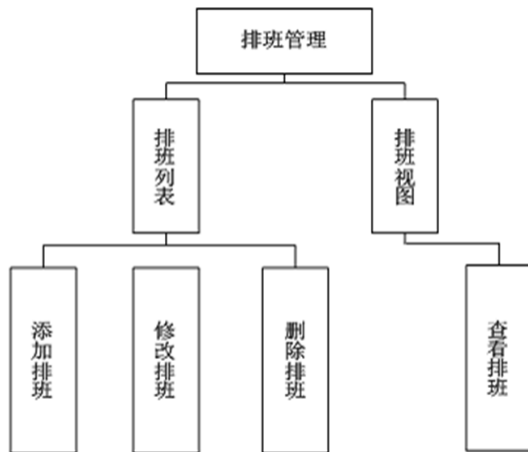


图2 排班管理功能图

3.4.2 用户管理

用户管理模块主要包括角色管理、权限管理、个人信息修改和员工管理等几部分功能。其中,角色管理实现添加角色、修改角色、删除角色和分配权限。权限管理实现添加权限、修改权限和删除权限。个人信息实现用户个人信息修改功能。员工管理实现新增员工、修改员工信息、删除员工、重置密码和分配角色。用户管理的功能划分如图3所示:

3.5 SSH 框架整合

要基于SSH框架开发,需要对Struts、Spring和Hibernate进行整合,将它们集成在一起。

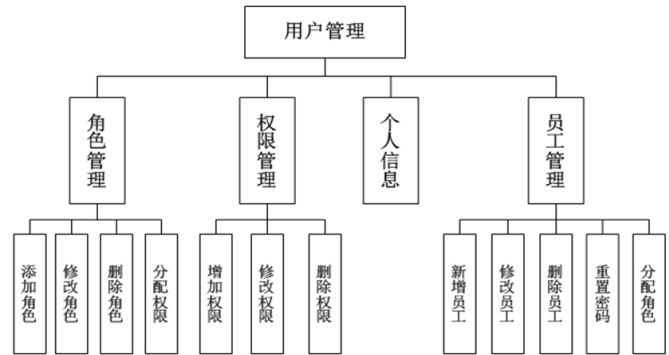


图3 用户管理功能图

3.5.1 集成 Struts

包括在Web部署文件中定义filter、复制Struts及其所需要的类库到项目的WebRoot\WEB-INF\lib目录中、创建Struts.xml配置文件。Web.xml文件配置如下:

```

<!-- 定义 Struts2 的核心控制器 FilterDispatcher -->
<filter>
    <filter-name>struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.
FilterDispatcher</filter-class>
</filter>
<filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
  
```

在这里,加载了Struts2的FilterDispatcher类。同时定义过滤器,指定该过滤器拦截所有的HTTP请求。当有HTTP请求过来时,会转向Struts来处理该request。

3.5.2 集成 Spring

首先配置web.xml,用以初始化Spring容器,以便项目运行时可以动态注入对象。

```

<!-- 指定以 Listener 方式启动 Spring 容器 -->
<listener>
    <listener-class>org.springframework.web.context.
ContextLoaderListener</listener-class>
</listener>
  
```

接着在web.xml里配置路径以便系统读取Spring配置文件。利用Spring的依赖注入特性来解耦系统,降低系统复杂性^[4]。

3.5.3 集成 Hibernate

配置好hibernate.cfg.xml文件,然后在Spring的配置文件中加载该配置文件,并且设置好hibernate的相关属性。

Spring的配置代码如下:

```

<!-- 加载 Hibernate 的配置 -->
<property name="configLocation" value="classpath:hibernate.cfg.
xml" />
<property name="configurationClass" value="org.hibernate.cfg.
AnnotationConfiguration" />
<!-- 设置 Hibernate 的相关属性 -->
<property name="hibernateProperties">
    <props>
        <prop key="hibernate.dialect">org.hibernate.dialect.
MySQLDialect</prop>
  
```

基于 .Net 的大客户管理信息系统的设计与实现

夏 飞¹ 王魁生¹ 李泽辉²

(1. 西安石油大学计算机学院, 陕西西安 710065; 2. 广东风华高新科技股份有限公司信息中心, 广东肇庆 526020)

摘 要:企业利用计算机管理信息系统进行客户资源管理是提高企业管理水平的有效手段。大客户管理信息系统主要针对企业的客户服务需要, 实现对大客户的个性化、高效的服务, 提高企业的管理水平。

关键词:数据库; 管理信息系统; 客户资料管理

中图分类号: TP302.1 **文献标识码:** A **文章编号:** 1003-9767 (2012) 09-0058-02

1. 前言

客户信息资料比较繁琐内容较多, 用户级别的不同, 故障响应时间的不同, 各基站点点的不同, 用户设备及连接方式的不同等等, 使得在用户的服务及故障响应有很大难度。因此, 以客户为中心的经营模式成为现代企业经营的标志, 也成为提高企业竞争力的重要方法。保持长期稳定的合作关系、把握客户信息、取得客户信赖、掌控客户真正需求、提高客户的忠诚度和满意度以及降低企业的运营成本, 客户信息资源的管理起了重要的支撑作用。

客户关系管理的出现, 改变了企业管理的重心: 由企业内部的运营管理转移到通过良好的客户关系及时发现并满足客户需求, 提高客户满意度和忠诚度上。企业通过构筑稳固的客户关系增强竞争实力, 它不但注重客户的满意程度, 更注重客户与企业建立一种长期的互利关系, 实现双赢目标。客户关系管理是企业的必然选择, 是管理理念的升华。另外, 随着计算机技术和互联网络技术的出现与普及, 数据库营销、一对一营销等基于先进技术的管理理念也被提出, 并融入到客户关系管理系统, 进一步提升了客户关系管理的理论和实践。

2. 系统功能组成

大客户是指消费到了一定数额资金的集团用户, 可以享受一定的优惠政策。在电信行业里, 大客户, 又称集团客户, 指大型厂矿企业、机关等等电信业务的大宗客户。大客户管理信息系统的设计, 主要是针对一些企事业单位、党政机关、金融等行业的通信线路租用业务和语音以及互联网业务。在以往的用户资料查找中经常遇到信息不全, 资料出错, 这不仅增加了用户业务维护的难度而且还会影响用户业务故障的定位判断和延长故障处理时间。大客户关系管理系统对大客户信息可以详细的记录和科学的管理, 这不仅可以提高对用户业务的维护质量还可以提高业务中断时维修的效率, 更有效的帮助故障定位大

大缩短故障处理时间。而且这个系统对全网的线路资源管理和线路规划也有一定帮助。

为有效管理大客户, 整个系统从总体上分为数据输入、信息查询、资料统计、系统维护四大部分:

(1) 资料输入部分

由用户档案管理 (包括用户档案的增加、修改、删除等)、用户消费信息管理 (包括用户各项消费信息的导入等)、用户服务记录管理 (包括客户经理走访或客户来访时间登记、走访服务内容、客户个人信息变更及客户意见的增加、修改、删除等) 组成。

(2) 资料查询部分

由用户档案查询 (按查询条件显示客户档案基本信息和消费信息)、走访记录查询 (按查询条件显示大客户享受的服务记录次数及相关内容)、纪念日查询 (客户经理可根据需要输入日期, 查询该日期是否为某大客户的特殊纪念日, 以便提供个性化服务) 组成。

(3) 资料统计部分

由消费统计 (主要是以输入客户编号为准, 查询客户或一个客户群近期的消费情况)、消费对比 (主要针对消费信息统计异常消费情况)、线路利用率 (主要统计线路租用状况和线路负载) 组成。

(4) 系统维护部分

由账户管理 (包括系统用户的增加、修改、删除和口令管理等)、系统编码、用户等级管理和数据库维护组成。

3. 系统结构

为了实现大客户的信息化管理, 方便管理人员的使用, 实现资源的共享和交流, 大客户管理信息系统采用 MVC 设计模式, 将系统分离成数据层、数据访问层、业务逻辑层和用户表示层, 其系统结构如图 1 所示。分层的架构使每层都有明确的责任, 这样可以降低整个系

```
<prop key="hibernate.hbm2ddl.auto">update</prop>
<!-- 设置 Hibernate 是否在控制台输出 SQL 语句, 开发调试阶段通常设为 true -->
<prop key="hibernate.show_sql">${showsql}</prop>
<!-- 设置 Hibernate 一个提交批次中的最大 SQL 语句数 -->
<prop key="hibernate.jdbc.batch_size">50</prop>
</props>
</property>
```

4. 结语

本文简要分析了基于 Web 的企业排班管理系统, 针对系统的应用需求, 提出了一个基于 J2EE 平台, 采用当前流行的 SSH 应用框架并结合 MVC 模式来完成系统设计的开发方案, 对系统的总体架构、

安全机制、访问控制设计以及系统功能进行了详细的阐述。同时, 探讨了 SSH 三大框架使用时的整合问题, 并给出了实现方法。

参考文献:

- [1] 计文柯. Spring 技术内幕: 深入解析 Spring 架构与设计原理 [M]. 北京: 机械工业出版社, 2012.
- [2] 郭克华. JavaEE 程序设计与应用开发 [M]. 北京: 清华大学出版社, 2011.
- [3] 李风华, 苏锐. 访问控制模型研究进展与发展趋势 [J]. 电子学报, 2012, 4 (9): 805-813.
- [4] 任振宇, 汪成曦. 基于 SSH 框架的企业信息管理系统设计研究 [J]. 科技资讯, 2012, (1): 15-16.