

# 基于 Struts 框架的企业办公自动化系统

## 摘 要

企业办公自动化系统是指企业在日常办公管理中通过一些技术手段提高办公效率，从而实现自动化办公处理的系统。近年来，J2EE 平台被广泛应用于各大办公系统中，大部分是基于 B/S 架构的 Web 应用。J2EE 设计中有很多开源框架，使得系统的开发成本得到了有效的控制。在各大开源框架中，Struts、Spring、Hibernate 三大框架是目前市场上主流的开源框架。

本文主要以 Struts+Spring+Hibernate 的框架集来构建基于 B/S 架构的企业办公自动化系统。根据软件工程的开发流程，对系统进行分析、实现。系统主要包含有人事管理、日程安排、工作日志、任务管理、消息传递、公告管理、请假管理、常用工具、附加功能九大功能模块。用户用例包含有经理和员工，不同角色的用户，操作权限也不同。

**关键词：**J2EE B/S架构 Struts Spring Hibernate

# Based on Struts framework enterprise office automation system

## Abstract

Enterprise office automation system is a system that enterprise uses some technical means to improve office efficiency the day-to-day office management. So that enterprise could achieve office automation processing. In recent years, J2EE platform has been widely used in major office systems, most of them are based on the Web application of the B/S architecture. There are many open-source framework while using J2EE to design, so that the cost of development of the system has been effectively controlled. Among in the major open-source framework, The three frameworks of Struts、 Spring and Hibernate is the main open-source framework now.

In this thesis, we use the frameworks of Struts、 Spring and Hibernate to build enterprise office automation system which is based on the Web application of the B/S architecture. According to the software engineering development process, we could analyze and achieve the system. There are nine functional modules in this system, which are Personnel Management, Schedule, Work Log, Task Manager, Messaging, Notice Management, Leave Management, Common Tools, and Additional Features. There are managers and employees in Use Cases. To different users, operating authority is different.

**Key Words:** J2EE ; B/S architecture ; Struts ; Spring ; Hibernate

# 目录

1. 引言.....	1
1.1 课题背景.....	1
1.2 研究目的和意义.....	1
1.3 国内外研究现状.....	1
1.4 主要研究内容.....	2
1.5 系统的实现.....	3
2. 系统相关技术介绍与分析.....	4
2.1 SSH 技术.....	4
2.1.1 Struts2 的工作原理.....	4
2.1.2 Spring 的工作原理.....	4
2.1.3 Hibernate 的工作原理.....	5
2.2 其他技术.....	5
2.2.1 JSP 技术.....	5
2.2.2 JS 脚本语言.....	5
3. 系统的需求分析.....	6
3.1 功能需求.....	6
3.2 其它具体需求.....	8
4. 系统的概要设计.....	9
4.1 系统用例图.....	9
4.2 请假管理功能的建模.....	10
5. 系统的实现.....	14
5.1 开发环境的搭建.....	14
5.2 系统数据库设计.....	14
5.2.1 数据库实体关系图.....	14
5.2.2 数据库表设计.....	15
5.3 SSH 框架搭建.....	19
5.4 具体功能实现.....	21
5.4.1 数据持久层的实现.....	21
5.4.2 业务层的实现.....	22
5.4.3 表示层的实现.....	23
结束语.....	26
参考文献.....	27
致谢.....	28

# 1. 引言

## 1.1 课题背景

随着科学技术的发展，特别是近十年信息技术的迅猛发展，传统的手工办公方式已逐渐走向末路，被企业所摒弃，取而代之的是新兴的办公自动化技术。办公自动化实现了“一人一机”的办公方式，革新了繁琐、复杂的传统手工办公方式。由于种种的优越性，大部分企业选择办公自动化系统。

## 1.2 研究目的和意义

办公自动化系统是利用技术的手段提高办公的效率，进而实现办公自动化处理的系统<sup>1</sup>。使企业工作人员之间实现信息互通，高效、快捷的协同合作，进行日常工作的处理。

办公自动化系统通过信息化技术，提高企业办公效率以减少企业处理日常工作的成本。实现企业之间、部门之间、岗位之间的信息互通，企业之间或者企业内部资源的共享。整合单位内、外各类信息系统和信息资源，实现日常办公流程的高效管理，同时用户通过系统提供的丰富的流程定制工具，可以迅速搭建起面向具体业务的办公环境<sup>2</sup>。

## 1.3 国内外研究现状

在上世纪 90 年代信息技术的迅猛发展之下，办公自动化也得到了很大程度的发展。目前，国内的办公自动化系统主要经过了三个阶段的发展。第一代办公自动化是以数据为处理中心的传统 MIS 系统，第二代是以工作流为中心的办公自动化系统，第三代办公自动化系统建立在企业 Internet 平台之上<sup>3</sup>。以国内几篇硕士研究论文为例，说明目前的研究现状。

文尹斐的《基于 Struts 和 Hibernate 的办公自动化系统的设计与实现》一文运用了 Struts1 和 Hibernate 框架来构建一个开源的轻量级 Web 应用框架，然后基于开源技术的发展现状，选择适合企业应用的开源软件进行企业级开发部署，将应用框架和开源软件

---

<sup>1</sup> 赵媛媛,杨涛.浅谈办公自动化系统在煤炭企业管理中的应用[J].科技与企业, 2012:1~2

<sup>2</sup> 王双业,刘波,张晶姝.办公自动化信息系统研究与企业应用现状[J].数字石油和化工, 2009:158

<sup>3</sup> 李正涛.OA 系统发展历程与趋势[J].办公自动化, 2008:14~15

应用于重庆华日有限公司办公自动化系统项目的开发和部署<sup>1</sup>。Struts1 经过多年的发展完善，超过四分之一的市场份额，使其成为市场上最热门的开源框架之一。由于其稳定性，使得在开发过程中拥有丰富经验的软件开发工程师，已经成为企业开发的工业标准。然而，随着时间的推移，技术的进步，Struts1 的局限性和制约性逐渐地暴露出来，约束了 Struts1 的进一步发展。Struts1 框架中，与 JSP、Servlet 的高度耦合，使其本身具有很大的缺陷。首先，Struts1 视图层中可运用的技术相当有限，不能与 FreeMarker、Velocity 等这些视图层的模版技术进行整合。其次，Struts1 与 Servlet API 的严重耦合，使得应用程序很难测试。最严重的是 Struts1 代码很大程度上依赖于 Struts1 API，属于一种侵入性框架。

容宇的《企业级的办公自动化系统的分析与实现》一文是通过 SOA 架构+MVC 三层模式+J2EE 进行系统架构的<sup>2</sup>。SOA 服务结构，在开发过程中虽然有着诸多优点但也并非十全十美，目前市场上主流实现方式接口很多，从而使整体结构松散、脆弱，如果接口标准选择有问题，如主流的 Webservice 之类，会使整个系统不稳定并且还会增加额外的开销。其次，底层代码在与数据库进行交互时，需要手动配置 JDBC，每次进行数据操作时都需要建立连接、编写 SQL、执行 SQL、关闭连接。在代码编写时，不断重复，代码冗余。应对数据操作进行简单的方法封装，或者使用市场上一些开源的框架。

李锋的《基于 SSH 架构的办公自动化系统的设计与实现》一文采用 SSH (Struts、Spring、Hibernate 的缩写) 三层架构体系进行设计<sup>3</sup>。三层架构通常包含有：数据持久层、业务层、视图层，运用三层结构设计的系统可降低耦合度、增强系统的内聚性，利用 Struts 来充当表示层中的视图及控制，即 MVC 开发模式中的 V 和 C，利用 J2SE 来实现相关的业务逻辑，Hibernate 负责底层对数据库的相互及操作，Spring 通过配置文件，负责对象实例的创建。

通过多篇文献的研究比较，采用目前主流的 Struts2+Spring+Hibernate 即 SSH 框架，进行系统的开发效率比较高并且具有良好的稳定性和可扩展性。Struts2 负责表示层中的视图及控制层，Hibernate 负责底层对数据库的相互及操作，Spring 通过配置文件，负责对象实例的创建。

---

<sup>1</sup> 文尹斐.基于 Struts 和 Hibernate 的办公自动化系统的设计与实现：[硕士学位论文].成都：电子科技大学，2007:62

<sup>2</sup> 容宇.企业级的办公自动化系统的分析与实现：[硕士学位论文].南京：南京理工大学，2011:8

<sup>3</sup> 李锋.基于 SSH 架构的办公自动化系统的设计与实现：[硕士学位论文].湖南：中南大学，2010:4

## 1.4 主要研究内容

### A. SSH 框架介绍以及其他涉及技术的简介

分别对SSH三大框架的工作原理进行阐述，然后对本文另外所涉及到的JSP、JavaScript、CSS样式表进行简要介绍。

### B. 系统的需求分析

由于没有真正的实际用户群，不能通过用户调研进行用户需求分析，只能通过同类软件比较，来构建软件的业务需求。

### C. 系统的概要设计

在需求分析的基础上对各个功能模块进行建模，利用UML的图形化开发模式来描述业务需求，并根据该模式设计系统的总体架构以及各模块包含的功能。以请假功能为例，通过用例图、时序图、活动图描述业务模型的整个建模过程。

### D. 系统的实现

对系统开发环境的搭建，数据库的设计以及SSH框架的搭建进行介绍。以邮件收发功能为例，详细介绍邮件管理模块功能的实现。

## 1.5 系统的实现

系统实现中包括环境搭建、数据库设计、编码及运行界面等相关工作，在编码部分将每一个层次的功能代码进行解析，强化对轻量级框架分层结构的认识及实现过程。

## 2. 系统相关技术介绍与分析

### 2.1 SSH 技术

#### 2.1.1 Struts2 的工作原理

A. 客户端初始化一个指向 Servlet 容器（例如 Tomcat）的请求。

B. 请求经过一系列的过滤器（Filter）（这些过滤器中有一个叫做 ActionContextCleanUp 的可选过滤器，这个过滤器对于 Struts2 和其他框架的集成很有帮助）。

C. 接着 FilterDispatcher 被调用，FilterDispatcher 询问 ActionMapper 决定这个请求是否需要调用 Action。

D. 如果 ActionMapper 决定需要调用某个 Action，FilterDispatcher 把请求的处理交给 ActionProxy。

E. ActionProxy 通过 Configuration Manager 询问框架的配置文件，找到需要调用的 Action 类。

F. ActionProxy 创建一个 ActionInvocation 的实例。

G. ActionInvocation 实例使用命名模式来调用，在调用 Action 的过程前后，调用相关的拦截器。

H. Action 执行结束后，ActionInvocation 负责根据 Struts.xml 中的配置找到对应的返回结果，返回结果通常是一个需要被表示的 JSP 或者 FreeMarker 的模版<sup>1</sup>。在表示的过程中可以使用 Struts2 框架中继承的标签。在这个过程中需要涉及到 ActionMapper。

#### 2.1.2 Spring 的工作原理

Spring 框架的核心是 IOC/AOP。IOC(Inversion of control)控制反转，是指控制权反转给容器，容器去创建各对象实例并且创建各对象之间的关系。动态注入，对象的实例不再通过 new 方法来实现，而是由 Spring 生成对象实例，这其实就是 J2SE 里的反射，运行时根据 Spring 的配置文件动态的去创建、调用对象以及对象里的方法的。

AOP (Aspect-Oriented Programming) 面向面的编程，针对一个方面编写一个

---

<sup>1</sup> Donald Brown, Chad Michael Davis, Scott Stanlick. Struts 2 in Action[M]. 人民邮电出版社, 2010

InvocationHandler，然后通过 JDK 反射包中的定义的 Proxy 类为各种接口动态生成相应的代理类，可以对某一类对象进行监督和控制（也就是在调用这类对象的具体方法的前后去调用你指定的模块）从而达到对一个模块扩充的功能<sup>1</sup>。

### 2.1.3 Hibernate 的工作原理

Hibernate 是一个开源代码的对象关系映射框架，对 JDBC 对象进行了封装，使得 Java 程序员可以随心所欲的使用面向对象的编程思维对数据库进行操作<sup>2</sup>。Hibernate 通过 Java 类到数据表的映射关系，程序员可以直接对对象进行查询或操作实现对数据库相应的操作。相比使用 JDBC 和 SQL 手动操作数据库，Hibernate 可以大大减少开发人员对数据库的操作量。Hibernate 还可以通过代理模式来简化类的装载过程，这将大大降低使用 HibernateQL 从数据库中提取数据的代码，从而节省开发时间和开发成本。

## 2.2 其他技术

### 2.2.1 JSP 技术

JSP（Java Server Pages）技术是一种用来开发动态 Web 应用程序由 Sun 发布的技术，凭借其简单易学，跨平台的特性，在众多动态 Web 应用程序编程语言中异军突起，在短短几年内，已经形成了一套完整的规格，广泛应用于电子商务等领域<sup>3</sup>。随着 JSP 技术的发展，JSP 的优点已经深入人心，得到了越来越多的动态网站开发者的认可。JSP 技术相比传统的 HTML，能够很好的呈现动态界面，网页生成更加快速、简单。在 Web 应用程序构建中，JSP 能够与各种服务器、浏览器兼容。

### 2.2.2 JS 脚本语言

JS（JavaScript）是一种用于嵌入客户端浏览器的脚本语言。Netscape 最初在 1995 年提出的一个 JavaScript，但当时它叫 LiveScript，后来 Netscape 迅速地将 LiveScript 改名为 JavaScript。Java 的开发商 Sun 与他们在同一年发表了一项声明，声明中指出 Java 和 JavaScript 将互相补充，但它们是截然不同的技术，从而打消了很多人对这两种技术错误的理解<sup>4</sup>。JavaScript 提供了一种脚本语言，用于创建用户界面控件。JavaScript 在浏览器

---

<sup>1</sup> 陈雄华.Spring 3.x 企业应用开发实战[M].电子工业出版社，2012

<sup>2</sup> 孙卫琴.精通 Hiberbate[M].电子工业出版社，2005

<sup>3</sup> 张新曼.精通 JSP-WEB 开发技术与典型应用[M].人民邮电出版社，2007

<sup>4</sup> 夏天.JavaScript 高级应用与实践[M].电子工业出版社，2008

中插入了代码逻辑，使它可以支持 Web 页面上的一些验证，或者在移动鼠标过程中实现某些特殊效果。

### 3. 系统的需求分析

#### 3.1 功能需求

本办公管理系统在使用的过程中所需具备的功能如下：人事管理、日程安排、工作日志、任务管理、消息传递、公告管理、请假管理、常用工具、附加功能九个功能板块。其详细结构如图3-1所示。



图 3-1 系统功能结构图

下面针对各功能板块进行详细说明：

人事管理包括个人信息、职工信息、部门信息。每个登录系统的用户都能查看自己的个人信息详情，并且能够对信息进行维护（修改），还能够查看所有职工的信息列表和详情以及所有部门的信息列表和详情。经理在此基础上还能在职工信息中对所有职工的信息进行维护（包括增加、修改、删除），在部门信息中对所有部门进行维护（包括增加、修改、删除）。

日程安排包括个人日程、部门日程。每个登录系统的用户都能查看自己的个人日程列表和详情以及所属部门的部门日程列表和详情。经理能查看自己的个人日程列表和详情以及所有部门的部门日程列表和详情，并且能够对所有员工的个人日程进行维护（包括增加、修改、删除）以及所有部门的部门日程进行维护（包括增加、修改、删除）。

工作日志仅包括个人工作日志。每个登录系统的用户都能查看自己的工作日志列表以及详情，并且能够对工作日志进行维护（包括增加、修改、删除）。

任务管理包括任务分配、任务查看。每个登录系统的用户都能通过任务查看来查看自己所分配到的任务（完成状态：未完成）列表和详情，并且能够对任务进行完成操作，一旦完成后，任务将不再显示在任务列表中。经理在此基础上还有任务分配操作，任务分配列表能够查看所有已发布的任务（完成状态：未完成和已完成）列表和详情，并且能够对任务分配进行维护（包括增加、修改、删除）。

消息传递包括新建邮件、个人信箱、垃圾箱。每个登录系统的用户都能够通过新建邮件来新建信息发送给其它员工，并能够添加不大于 9M 的附件。个人信箱显示收到邮件的列表，用户能够查看详情并且能够下载其附件。还能对邮件进行删除操作。所有删除的邮件将显示在垃圾箱中，用户能够将垃圾箱中的邮件进行恢复或者永久删除，垃圾箱中的邮件仅显示邮件列表，无法查看详情。

公告管理包括发布公告、查看公告。每个登录系统的用户都能查看公告列表和详情。经理不仅能查看公告还能对公告进行维护（包括修改、删除），还可以通过发布公告来发布新的公告。

请假管理包括请假申请、请假审批。每个登录系统的用户都能查看自己的请假条列表和详情，状态为待审核、已批准、已驳回，并且能够对待审核的请假条进行抽回，取消申请。经理能够查看待审核的请假条列表和详情，并且能够对请假条进行审核（批准或者驳回）。

常用工具包括计算器、万年历。每个登录系统的用户都能使用常用工具。

附加功能包括资料、密码修改、系统注销。每个登录系统的用户都能查看一些企业

内部的共享资料，并且能够对登录密码进行修改。系统注销，退出当前系统返回到登录界面。

### 3.2 其它具体需求

系统角色设置：由于该系统目前尚处于初期开发阶段，角色权限限定为：经理、员工。员工能够进行日常办公中的基础操作，高一级的维护都交由经理完成。

系统界面要求。基于易于使用，美观，统一的原则，对于操作比较复杂的步骤能有相关的提示信息，以帮助用户尽快熟悉系统的使用，尽可能让每个界面仅包含相关信息，无需额外的冗余信息，以减少对工作人员的干扰。不同用户登录有相应的操作模块，应显示在页面左侧的导航栏上，以使用户操作。

## 4. 系统的概要设计

### 4.1 系统用例图

在面向对象的程序设计中，首先要搞清楚设计的对象。设计办公系统，首先就要分析系统具有哪些模块，操作模块的用户以及用户需要什么这是我们分析模块的来源。下面描述了该办公系统的用户以及他们之间的关系（如图4-1所示）。

系统用户关系图显示了系统中不同用户（在建模中叫做参与者）之间的关系，其中“系统用户”为所有用户，位于顶端。箭头表示“泛化”，它包括“经理”和“员工”。通过系统用户用例图可以直观的看到该系统涉及到的用户，然后在系统的业务功能设计中，就针对这两类用户进行设计。

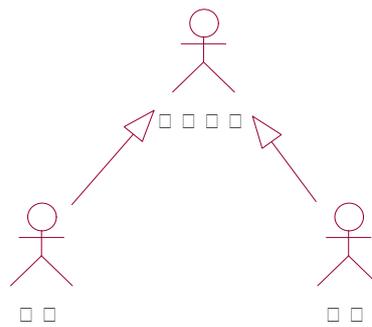


图 4-1 用户用例图

#### 1) 确定系统用户

通过对办公自动化系统需求分析得到：两个系统用户

(1) 经理：对系统内的所用子系统进行操作。

(2) 员工：对系统内的部分功能进行操作。

#### 2) 确定用例

在办公自动化系统顶层用例图中（如图4-2所示），在系统内共有8个功能用例，系统外有两类系统用户。

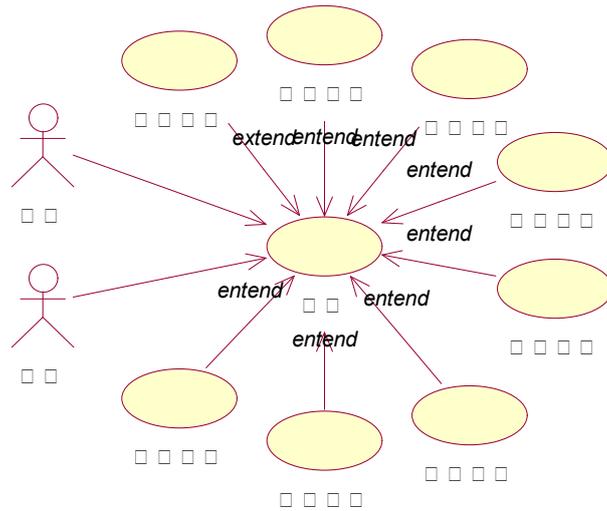


图 4-2 系统顶层用例图

## 4.2 请假管理功能的建模

本系统中共有 8 个功能用例，各功能用例中根据不同的系统用户又分为 2~3 个子功能用例。由于篇幅的限制，UML 建模过程又是需要对功能业务进行详细分析说明的过程，所以在本课题中不可能对系统中各个功能业务一一进行 UML 建模过程的说明。下面就以请假管理功能为例，说明其 UML 建模过程。

上文已经从系统整体角度描述了整个系统的用例图，下面对请假管理功能用例进行详细的用例描述，如图 4-3 所示。

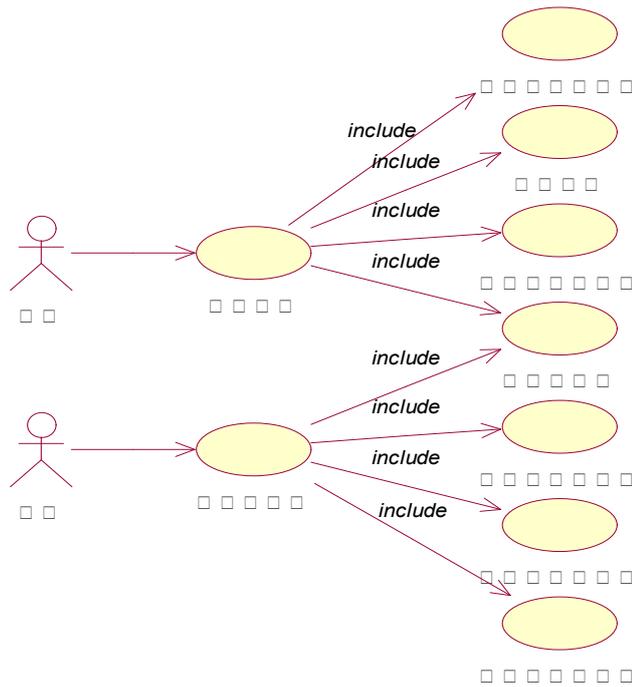


图4-3 请假管理用例图

时序图是用来描述对象之间传递消息的时间顺序，它主要用来表示用例的行为顺序<sup>1</sup>。通过需求分析中功能需求的分析，请假管理功能流程包括请假申请、请假审批。请假申请流程：员工登陆后获取个人请假单列表界面，点击新建申请，获取请假单空白表单，填写表单，提交表单，提交后，若单据为未审核状态，则可进行抽回操作。请假审批流程：经理登陆后获取请假单申请列表界面，点击列表中的每一条单据请求单据详细界面，详细界面能够对单据进行批准和驳回操作。将上述流程转化为时序图来表示，如图4-4、图4-5所示。

<sup>1</sup> Sherry X. Sun, J. Leon Zhao. Developing a Workflow Design Framework Based on Dataflow Analysis. Proceedings of the 41st [J]. Hawaii International Conference on System Sciences, 2008:1530-1605

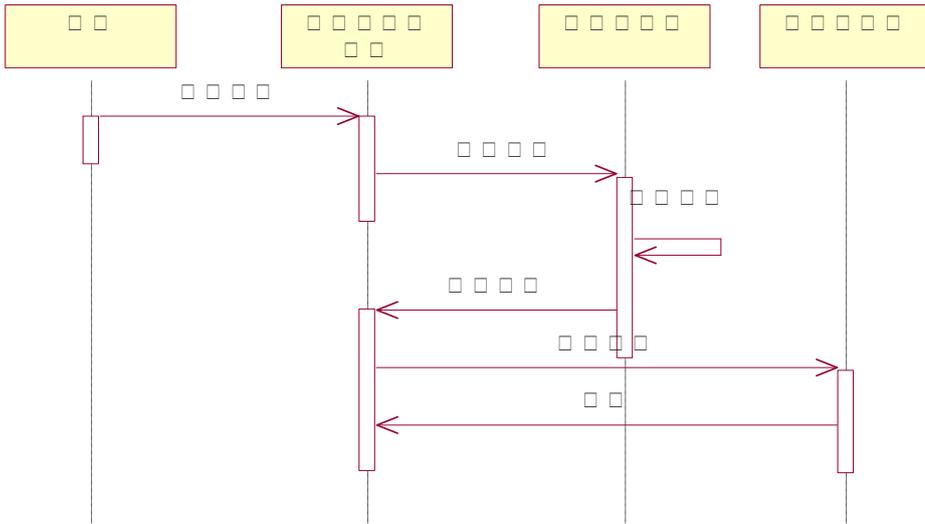


图4-4 请假申请时序图

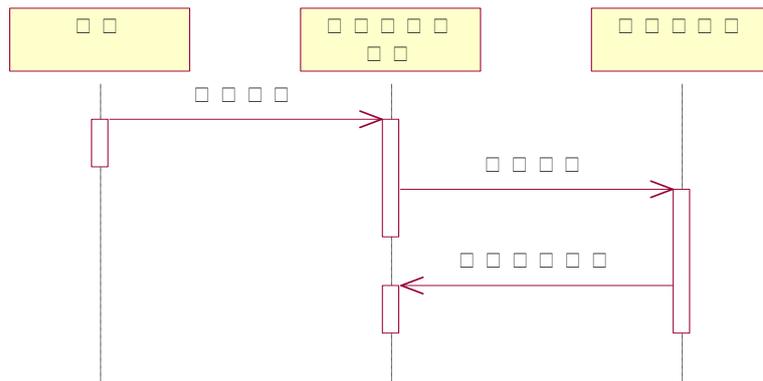


图4-5 请假审批时序图

活动图用于描述需要做的活动以及执行这些活动的顺序<sup>1</sup>。如图4-6所示，就是请假管理的活动图。

<sup>1</sup> Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modelng Language User Guide(Second Edition) [J].人民邮电出版社, 2009

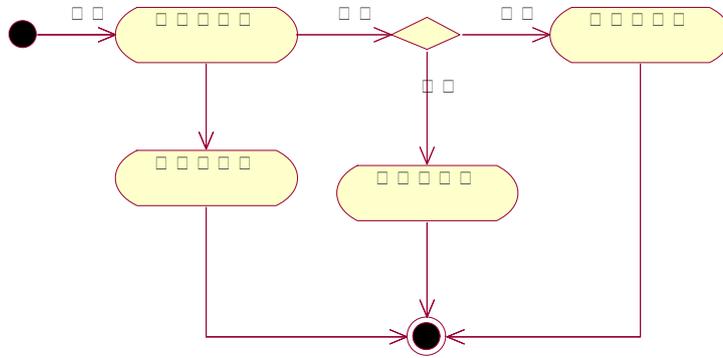


图4-6 请假管理活动图

## 5. 系统的实现

### 5.1 开发环境的搭建

本系统选用Java语言开发，开发软件选用MyEclipses8.5，服务器选用Tomcat7.0，后台数据库使用的是MySQL5.1.30，数据库管理是用的Navicat for MySQL8.0，客户端可采用任何基于Web应用的浏览器，如Explorer、Firefox等。

### 5.2 系统数据库设计

本系统选用的数据库是MySQL，选择它是因为它具有速度快、易用性好、支持SQL、广泛支持各种网络环境、可移植性强等特点。相比于Oracle、DB2等大型数据库来说MySQL的操作、管理上更为简单和高效。MySQL还可以运行在Linux、Unix、Solaris、Windows等各种平台上，因此有着更强的移植性；开源的使用是完全免费的，这样又可以减少相应的成本，通过网络论坛，我们可以获取相关的技术支持，方便日后的维护工作。

#### 5.2.1 数据库实体关系图

本系统数据实体关系图，如下图5-1所示：

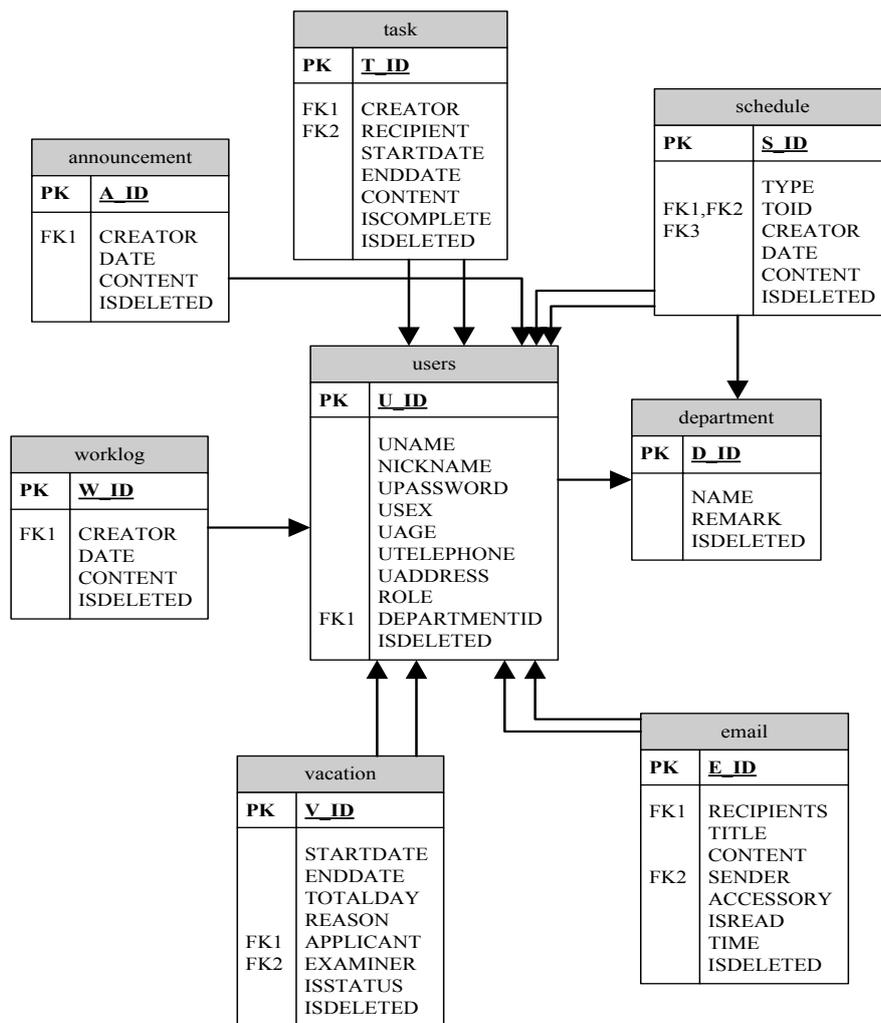


图 5-1 数据库实体关系图

### 5.2.2 数据库表设计

用户表（users）如下表5-1所示：

字段名	数据类型	长度	是否主键	默认值	描述
U_ID	int	20	YES		ID(自动编号)
UNAME	varchar	20		NULL	姓名
NICKNAME	varchar	20		NULL	昵称
UPASSWORD	varchar	16		NULL	密码
USEX	varchar	10		NULL	性别
UAGE	int	10		NULL	年龄
UTELEPHONE	varchar	20		NULL	电话

UADDRESS	varchar	100		NULL	地址
ROLE	varchar	20		NULL	角色（经理、 员工）
DEPARTMENTI D	varchar	20		NULL	部门 ID
ISDELETED	int	10		NULL	是否删除

表 5-1 用户表

部门表（department）如下表5-2所示：

字段名	数据类型	长度	是否主键	默认值	描述
D_ID	int	20	YES		ID(自动编号)
NAME	varchar	20		NULL	名称
REMARK	varchar	200		NULL	备注
ISDELETED	int	10		NULL	是否删除

表 5-2 部门表

工作日志表（worklog）如下表 5-3 所示：

字段名	数据类型	长度	是否主键	默认值	描述
W_ID	int	20	YES		ID(自动编号)
CREATOR	int	20		NULL	创建者 ID
DATE	varchar	50		NULL	时间
CONTENT	varchar	200		NULL	内容
ISDELETED	int	10		NULL	是否删除

表 5-3 工作日志

日程表（schedule）如下表5-4所示：

字段名	数据类型	长度	是否主键	默认值	描述
S_ID	int	20	YES		ID(自动编号)
TYPE	varchar	20		NULL	类型（部门日

					程、个人日程)
TOID	int	50		NULL	所属 ID
CREATOR	int	20		NULL	创建者 ID
DATE	varchar	50		NULL	时间
CONTENT	varchar	200		NULL	内容
ISDELETED	int	10		NULL	是否删除

表 5-4 日程表

任务表 (task) 如表下5-5所示:

字段名	数据类型	长度	是否主键	默认值	描述
T_ID	int	20	YES		ID(自动编号)
CREATOR	int	20		NULL	创建者 ID
RECIPIENT	int	50		NULL	所属者 ID
STARTDATE	varchar	50		NULL	开始日期
ENDDATE	varchar	50		NULL	结束日期
CONTENT	varchar	200		NULL	内容
ISCOMPLETE	int	10		NULL	是否完成
ISDELETED	int	10		NULL	是否删除

表 5-5 任务表

公告表 (announcement) 如表下 5-6 所示:

字段名	数据类型	长度	是否主键	默认值	描述
A_ID	int	20	YES		ID(自动编号)
CREATOR	int	20		NULL	创建者 ID
DATE	varchar	50		NULL	时间
CONTENT	varchar	200		NULL	内容
ISDELETED	int	10		NULL	是否删除

表 5-6 公告表

请假申请单表（vacation）如表下5-7所示：

字段名	数据类型	长度	是否主键	默认值	描述
V_ID	int	20	YES		ID(自动编号)
STARTDATE	varchar	50		NULL	开始日期
ENDDATE	varchar	50		NULL	结束日期
TOTALDAY	int	10		NULL	天数
REASON	varchar	200		NULL	理由
APPLICANT	int			NULL	申请人 ID
EXAMINER	int	20		NULL	审核人 ID
ISSTATUS	varchar	20		NULL	状态（待审核、已通过、已驳回）
ISDELETED		10		NULL	是否删除

表 5-7 请假申请单表

邮件表（email）如表下5-8所示：

字段名	数据类型	长度	是否主键	默认值	描述
E_ID	int	20	YES		ID(自动编号)
RECIPIENTS	int	20		NULL	接收方 ID
TITLE	varchar	50		NULL	标题
CONTENT	varchar	4000		NULL	内容
SENDER	int	20		NULL	寄件人 ID
ACCESSORY	varchar	200		NULL	附件
ISREAD	int	10		NULL	是否读取
TIME	varchar	50		NULL	发送时间
ISDELETED		10		NULL	是否删除

表 5-8 邮件表

### 5.3 SSH 框架搭建

首先，建立一个 Web 项目，加入 Spring 的需要的 jar 跟 spring 的配置文件 applicationContext.xml。然后在项目的 web.xml 里面加入 spring 的信息：

```
<context-param>

    <param-name>contextConfigLocation</param-name>

    <param-value>classpath:applicationContext.xml</param-value>

</context-param>

<listener>

    <listener-class>

        org.springframework.web.context.ContextLoaderListener

    </listener-class>

</listener>
```

然后导入 Struts2、Spring、Hibernate 以及 MySQL 数据库连接驱动的相关架包。

然后在 applicationContext.xml 里面加入数据源的配置信息（本系统采用的是 MySQL 数据库）。

```
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">

    <propertyname="driverClassName" value="com.mysql.jdbc.Driver"/>

    <propertyname="url" value="jdbc:mysql://localhost:3306/oa"/>

    <propertyname="username" value="root"/>
```

```

    <propertyname="password" value="123"/>

</bean>

<beanid="sessionFactory" class="org.springframework.orm.hibernate3.LocalSe
ssionFactoryBean">

    <propertyname="dataSource" ref="dataSource"/>

    <propertyname="hibernateProperties">

        <props>

            <propkey="hibernate.hbm2ddl.auto">update</prop>

            <propkey="dialect">org.hibernate.dialect.MySQLDialect</prop>

            <propkey="current_session_context_class">thread</prop>

            <propkey="cache.provider_class">

                org.hibernate.cache.NoCacheProvider

            </prop>

        </props>

    </property>

</bean>

```

建立实体类 xx.java 跟 hibernate 的映射文件 xx.hbm.xml。然后在 applicationContext.xml 加上引入映射文件配置。

```

</property>

    <propertyname="mappingResources">

```

```
<list>

    <value>dto/xx.hbm.xml</value>

</list>

</property>
```

至此，整个 SSH 框架就搭建完毕了，开发过程中只需写入相应的 DAO 层、Service 层以及 Action 类。

## 5.4 具体功能实现

本系统具体功能实现一一例举篇幅较大，这里就以消息管理功能即邮件的收发举例进行说明。

### 5.4.1 数据持久层的实现

在 src/com.entity 目录下，创建一个实体类，Email 对象。Email.java 部分代码如下：

```
public class Email implements Serializable {
    private static final long serialVersionUID = 1L;
    private Integer EId;
    private Integer recipients;
    private String title;
    private String content;
    private Integer sender;
    private String accessory;
    private Integer isread;
    private String time;
    private Integer isdeleted;
}
```

有了实体类之后下面就是创建和配置实体类和数据库表的关系映射文件

Email.hbm.xml，在 Email.java 的同一级目录下创建文件 Email.hbm.xml。需要注意的是，一般我们都是手动配置 xml 文件，但是在 MyEclipse 中提供了一种更为便捷的方法，由数据表到实体类的反向工程。通过这种反向工程，系统会帮你自动生成实体类以及实体类的映射文件，还可以选择系统自动生成相应的 Dao 层接口以及 Dao 层实现类的代码，不过一般我们只要生成实体类和映射文件就够了，Dao 层设计由开发人员自行设计编写。

实体类的映射文件配置完成后，在 applicationContext.xml 中引入该映射文件，配置如下：

```
<property name="mappingResources">
    <list>
        <value>com/entity/Email.hbm.xml</value>
    </list>
</property>
```

对象以及关系映射都配置好后，接下来就是设计 dao 层接口以及实现接口。在 src/com.dao 目录下新建 Email 的 dao 层接口 EmailDao.java。在 src/com.daoimpl 目录下新建 EmailDaoImpl.java 来实现我们设计的 dao 层接口。接口设计部分代码如下：

```
public interface EmailDao {
    public int add(Email email);
    public int delete(Integer EId, int isdeleted);
    public int update(Integer id);
    public List<Email> emailList(int UId);
    public Email select(Integer EId);
    public int saveFile(String filepath, String accessory);
    public int downloadFile(String filepath, String file);
}
```

#### 5.4.2 业务层的实现

业务层（service 层）就是通过调用 dao 层方法，以实现具体的业务功能。在 src/com.service 新建 Email 的 dao 层接口 EmailService.java。在 src/com.serviceimpl 目录下新建 EmailServiceImpl.java 来实现我们设计的 service 层接口。接口设计部分代码如下：

```

public interface EmailService{
    public int add(Email email);
    public int delete(Integer EId, int isdeleted);
    public int update(Integer id);
    public List<Email> emailList(int UId);
    public Email select(Integer EId);
    public int saveFile(String filepath, String accessory);
    public int downloadFile(String filepath, String file);
}

```

### 5.4.3 表示层的实现

这里不仅要设计 JSP 界面，还得写 Action 类。JSP 页面通过 Action 请求，调用 Action 类中相应的方法，方法返回值放入 request 中传递给前台 JSP。所以表示应该是包含两部分内容，一是 JSP 页面，二是 Action 类。而将这两者联系起来的就是 Struts 的配置文件 Struts.xml。

配置完 Struts.xml 后，在 src/com.action 目录下新建 EmailAction.java 控制 JSP 的跳转。部分代码如下：

```

//新建邮件方法
public String add() {
    Date date = new Date();
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
    String time = df.format(date);
    email.setTime(time);
    String str = email.getAccessory();
    if (str != null && !str.equals("")) {
        String fileName = email.getAccessory().substring(
            email.getAccessory().lastIndexOf("\\") + 1,
            email.getAccessory().length());
        String filepath = getSavePath() + fileName;
        File f = new File(getSavePath());
    }
}

```

```

    f.mkdir();// 创建文件夹

    int i = emailDao.saveFile(filepath, email.getAccessory());
    email.setAccessory(filepath);
    if (i == 1 && emailDao.add(email) != 0) {
        return "write_success";
    } else if (i == 2) {
        return "write_success";
    } else {
        return "fail";
    }
} else {
    if (emailDao.add(email) != 0) {
        return "write_success";
    } else {
        return "fail";
    }
}
}

//附件下载方法
public String download() {
    request = ServletActionContext.getRequest();
    String accessory = request.getParameter("accessory");
    int EId = Integer.parseInt(request.getParameter("EId"));
    String fileName = accessory.substring(accessory.lastIndexOf("/") + 1,
    accessory.length());
    String filepath = getDownloadPath() + fileName;
    File f = new File(getDownloadPath());
    f.mkdir();
    int i = emailDao.downloadFile(filepath, accessory);
    e = emailDao.select(EId);
}

```

```
if (i != 0) {
    if (e != null) {
        request.setAttribute("email", e);
        Users user = usersDao.findById(e.getSender());
        request.setAttribute("senderName", user.getUsername());
        return "detail_success";
    } else {
        return "fail";
    }
} else {
    if (e != null) {
        request.setAttribute("email", e);
        Users user = usersDao.findById(e.getSender());
        request.setAttribute("senderName", user.getUsername());
        return "detail_success";
    } else {
        return "fail";
    }
}
}
```

## 结语

五个多月的努力，论文终于在导师悉心的指导下完成了。从一开始的选题到后期的系统实现，一步步过来遇到了很多的问题。埋怨过，也想过放弃，可是在周围同学的鼓励中顺利的完成了论文系统。在这个过程中，不仅学会了怎样通过有限的资源更好的去整合自己的需求，更学习到了很多以前不会的技术，也让我对办公自动化系统有了新的认识。在选题完成初期，对系统需求的了解一无所有，通过查阅多方资料，以及在公司实习过程中积累的经验，终于对系统的需求有了一个较为全面的认识。在这里我要特别感谢我的导师，唐志强老师在不方便返校完成论文的情况下，给了我很多指导，还提供给我了大量的参考文献。在概要设计阶段，由于对以前学习过得软件工程、UML 建模设计与分析的逐渐淡忘，让我不得不拿起教科书重新学习。在系统实现方面也遇到了诸多问题，可是这些问题也无法通过查阅资料来解决，我就在公司实习之余，请教那些高级软件工程师。这种系统的实现对他们真可谓是小菜一碟，由于我虚心的求教，他们也很乐意为我解答。

通过毕业设计，让我认识到要做好任何一件事情都不是轻而易举的。在未动手之前想想是很简单，不过是纸上谈兵而已，可是真正解决起来就会有诸多问题，所以没有亲身体味，不要对任何一件事妄下定论。对待问题，我们要保持足够的耐心，善于运用身边的资源去解决它。

## 参考文献

- [1]赵媛媛, 杨涛. 浅谈办公自动化系统在煤炭企业管理中的应用[J]. 科技与企业, 2012:1~2
- [2]王双业, 刘波, 张晶姝. 办公自动化信息系统研究与企业应用现状[J]. 数字石油和化工, 2009:158
- [3]李正涛. OA 系统发展历程与趋势[J]. 办公自动化, 2008:14~15
- [4]文尹斐. 基于 Struts 和 Hibernate 的办公自动化系统的设计与实现: [硕士学位论文]. 成都: 电子科技大学, 2007:62
- [5]容宇. 企业级的办公自动化系统的分析与实现: [硕士学位论文]. 南京: 南京理工大学, 2011:8
- [6]李锋. 基于 SSH 架构的办公自动化系统的设计与实现: [硕士学位论文]. 湖南: 中南大学, 2010:4
- [7]Donald Brown, Chad Michael Davis, Scott Stanlick. Struts 2 in Action[M]. 人民邮电出版社, 2010
- [8]陈雄华. Spring 3. x 企业应用开发实战[M]. 电子工业出版社, 2012
- [9]孙卫琴. 精通 Hiberbate[M]. 电子工业出版社, 2005
- [10]张新曼. 精通 JSP-WEB 开发技术与典型应用[M]. 人民邮电出版社, 2007
- [11]李刚. 轻量级 Java EE 企业应用实战 (第 3 版) [M]. 电子工业出版社, 2011
- [12]夏天. JavaScript 高级应用与实践[M]. 电子工业出版社. 2008
- [13]Sherry X. Sun, J. Leon Zhao. Developing a Workflow Design Framework Based on Dataflow Analysis. Proceedings of the 41st [J]. Hawaii International Conference on System Sciences, 2008:1530-1605
- [14] Grady Booch, James Rumbaugh, Ivar Jacobson. The Unified Modeling Language User Guide (Second Edition) [M]. 人民邮电出版社, 2009