**分类号：TP311.52 U D C：D10621-408-(2017)2877-0**

**密 级：公 开 编 号：2013121158**

**成都信息工程大学**

**学位论文**

**任务管理系统的设计与实现**

|  |  |
| --- | --- |
| **论文作者姓名:** | **胡志鹏** |
| **申请学位专业:** | **网络工程** |
| **申请学位类别:** | **工学学士** |
| **指导教师姓名（职称）:** | **苟智坚（讲师）** |
| **论文提交日期:** | **2016年06月10日** |

**任务管理系统的设计与实现**

**摘 要**

企业的任务管理涉及到企业的每个部门，伴随着时代的发展，企业的任务也越来越复杂，而传统的任务管理模式已经逐渐满足不了现代企业的需求，企业应该使用信息化手段来对工作任务进行管理。

本文的任务管理系统旨在满足当代企业的任务管理需求。本系统使用Java作为后端编程语言，MySQL作为数据库，HTML作为前端编程语言，并使用异步的方式进行数据交互。使用这一任务管理系统可以帮助企业管理人员对任务进行实时管理，提高任务管理的准确性，减少任务管理的人力成本。本系统提供完善的任务发布，任务修改，任务划分，任务反馈，进度跟踪，消息中心，个人中心等功能。系统充分利用计算机技术和信息化的优势，帮助企业提高运行效率，降低运行成本。

**关键词:**任务管理；信息化；计算机技术 ；MySQL；Java

**Design and Implementation of Task Management System**

**Abstract**

The task management of the enterprise involves every department of the enterprise. With the development of the times, the task of the enterprise is more and more complicated. The traditional task management mode has gradually failed to meet the needs of the modern enterprise. The enterprise should use the information means to Work tasks to manage.

The task management system of this paper aims to meet the task management needs of contemporary enterprises. The system uses Java as a back-end programming language, MySQL as a database, HTML as a front-end programming language, and uses asynchronous methods for data interaction. The uses of this task management system can help enterprise management personnel to manage the task in real time, improve the accuracy of task management, and reduce the manpower cost of task management. The system provides a complete mission release, task modification, task division, task feedback, progress tracking, message center, personal center and other functions. System to take full advantage of computer technology and information technology to help enterprises improve operating efficiency and reduce operating costs.

**Key words:** task management; enterprise information; computer technology;

MySQL; Java

**目 录**

论文总页数:23页

[1 引言 1](#_Toc484885267)

[1.1 课题背景 1](#_Toc484885268)

[1.2 国内外研究现状 1](#_Toc484885269)

[1.3 本课题研究意义 1](#_Toc484885270)

[1.4 本课题研究目的 2](#_Toc484885271)

[2 实现技术 2](#_Toc484885272)

[2.1 系统开发环境 2](#_Toc484885273)

[2.1.1 开发工具 2](#_Toc484885274)

[2.1.2 开发环境 2](#_Toc484885275)

[2.1.3 数据库 2](#_Toc484885276)

[2.2 系统技术介绍 2](#_Toc484885277)

[2.2.1 Spring技术介绍 2](#_Toc484885278)

[2.2.2 Mybatis技术介绍 3](#_Toc484885279)

[3 需求分析 4](#_Toc484885280)

[3.1 可行性需求 4](#_Toc484885281)

[3.1.1 技术可行性分析 4](#_Toc484885282)

[3.1.2 社会可行性分析 4](#_Toc484885283)

[3.1.3 经济可行性分析 4](#_Toc484885284)

[3.1.4 法律可行性分析 4](#_Toc484885285)

[3.2 功能需求 4](#_Toc484885286)

[3.3 功能模块图 5](#_Toc484885287)

[3.4 系统流程图 5](#_Toc484885288)

[4 系统设计 6](#_Toc484885289)

[4.1 系统模块设计 6](#_Toc484885290)

[4.1.1 员工登陆模块 6](#_Toc484885291)

[4.1.2 我的模块 6](#_Toc484885292)

[4.1.3 任务中心模块 6](#_Toc484885293)

[4.1.4 反馈模块 6](#_Toc484885294)

[4.1.5 消息中心模块 6](#_Toc484885295)

[4.2 数据库设计 7](#_Toc484885296)

[4.2.1 数据库实体图 7](#_Toc484885297)

[4.2.2 实体关系图 8](#_Toc484885298)

[4.2.3 数据表设计 9](#_Toc484885299)

[5 系统实现 11](#_Toc484885300)

[5.1 登陆实现 11](#_Toc484885301)

[5.2 我的模块实现 12](#_Toc484885302)

[5.3 任务中心实现 12](#_Toc484885303)

[5.4 反馈实现 13](#_Toc484885304)

[5.5 消息中心实现 14](#_Toc484885305)

[6 系统测试 15](#_Toc484885306)

[6.1 登陆测试 15](#_Toc484885307)

[6.2 我的模块测试 16](#_Toc484885308)

[6.3 任务中心测试 17](#_Toc484885309)

[6.4 反馈模块测试 18](#_Toc484885310)

[6.5 消息中心测试 19](#_Toc484885311)

[6.6 测试结语 19](#_Toc484885312)

[结 语 20](#_Toc484885313)

[参考文献 21](#_Toc484885314)

[致 谢 22](#_Toc484885315)

[声 明 23](#_Toc484885316)

**1 引言**

**1.1 课题背景**

在信息化高度发达的今天，企业迎来了新的机遇也迎来了新的挑战。当下是一个多元化的时代，人们逐渐从以前的集体性的需求转变成为现在的个性化的需求，这对企业来说意味着可以有更多的业务可以扩展也意味着企业需要随时应对市场和客户的需求变化，而传统的任务管理方式无法高效快速的让企业对策略做出调整，这对于现代企业来说是一个致命的缺点，现代企业需要对市场有着快速的应变能力，企业不能浪费过多的时间在任务的发布和分配上，企业需要将管理人员的任务策略快速地分发给普通员工，普通员工对自己所负责的任务有问题可以及时地将问题反馈给管理人员以便问题能够及时地得到解决。任务管理系统旨在满足现代企业对任务管理的需求，它可以帮助企业应对市场的变化，从而使企业能够在竞争中获得优势。

**1.2 国内外研究现状**

二十世纪七十年代后，欧美发达国家的任务管理自动化系统得到了飞速的发展，欧美等发达国家都在大力推进任务管理系统的自动化研究，并且他们在理论上得到了创新，在实践上取得了成果，达到了较高的水平。

我国计算机技术发展时间较短，企业信息化和自动化的水平较低，受到国内资源和政策的影响，目前国内大部分企业的任务下达还处在书面管理的形式，甚至有些企业还处于口头下达任务的状态，下达任务的方式过于简单。对完成任务所需的人力，物力以及任务状态都没有很好的描述，这会让刚进入企业的新员工感到无从下手。所以我国企业当前的首要工作是实现任务管理的自动化和信息化，把先进的计算机技术和科学的管理理念结合，运用到企业的实际任务管理过程中，是提高企业任务管理效率，降低任务管理成本的主要途径。

**1.3 本课题研究意义**

企业的任务分配是一个包含大量人员、资源和信息的过程，在这个过程中获取信息和管理过程是两大关键因素。获取信息主要影响任务能否顺利地进行，管理过程影响企业是否具备与其他企业竞争的核心竞争力。大型项目的完成过程是一个高度复杂非线性的过程，局部不协调可能引起蝴蝶效应导致全局的混乱，甚至于导致无法完成项目的预期目标。

根据科学的管理理论，一个项目的任务划分得越精细就越能提高工作效率，减轻员工工作负担，使员工能够有更高的激情去工作。本文的研究通过建立任务管理系统，对整个项目的任务流程进行管理，可以使管理人员清晰地了解各个任务的执行状态，同时，也可以使任务的负责人清楚自己的任务目标。本系统是企业任务管理的优化和实用工具，因此，本课题的研究不仅具有科学意义而且具有实用价值。

**1.4 本课题研究目的**

任务管理是企业正常运作的核心之一，目前一些企业仍使用传统落后的任务管理方式，而传统的任务管理方式已经无法跟上现代社会快节奏的脚步，这严重地限制企业的发展，影响企业的核心竞争力。任务管理系统就是为了帮助企业更好地进行任务管理，使用它，企业管理人员可以将任务信息更好地展示出来，可以将任务划分成为多级的子任务分配给任务的负责人，可以清晰地了解到各个任务的状态，方便管理人员做出战略调整，而任务的负责人可以清楚地了解到自己所负责任务的任务目标，可以更快捷地将自己的任务进度展示给任务发布人，可以快速地将任务的问题反馈给任务发布人。使用任务管理系统作为企业的任务管理核心工具可以提高企业的工作效率，降低企业的运行成本，让企业能够在竞争中取得优势。

**2 实现技术**

**2.1 系统开发环境**

**2.1.1 开发工具**

本系统开发工具为MyEclipse，MyEclipse是一个Eclipse的插件强化版本，它主要应用于JavaWeb的开发，是一个很优秀的Java开发工具。

**2.1.2 开发环境**

本系统开发环境为JDK1.7，JDK是Java的开发工具包，其中包括Java的编译器，Java的虚拟机以及Java的基础类。

**2.1.3 数据库**

本系统数据库为MySQL, MySQL是一个轻型的开源关系型数据库，作为最好的网页应用的数据库，它具有体积小，速度快，成本低等特点。

**2.2 系统技术介绍**

**2.2.1 Spring技术介绍**

Spring框架是一个开源的轻量级的企业级Java应用开发框架，它被设计成无入侵的方式，即开发者可以根据自己的需求选择它必要的组件，而忽略自己不需要的部分，以做到最小的依赖。Spring框架拥有许多特性，它包含七个模块，这七个模块分别是Spring Core模块、Spring Context模块、Spring DAO模块、Spring ORM模块、Spring Web模块、Spring AOP模块和Spring MVC模块。

Spring Core模块是Spring框架的核心模块，它封装了Spring的核心包，主要是实现反转控制容器。反转控制是Spring实现功能的基础，其他一切的功能都是基于反转控制。

Spring Context模块主要为开发者提供访问Spring中的对象的方式，并包含国际化和事件传播等特性。

Spring DAO模块是对JDBC的一个高度集成封装包，它避免了JDBC繁琐冗长的代码，为开发者开发数据库提供便捷，同时，它还具有声明事物管理的功能。

Spring ORM模块提供集成常用的ORM框架的封装包，使用它可以更快速地使用ORM框架进行开发，同时，它也可以使用Spring的其他特性。

Spring Web模块主要提供Web开发的一些基础特性，如文件上传等功能。同时也提供与常用Web框架集成的封装包。

Spring AOP模块是Spring中的另一个核心模块，AOP又称面向切面编程，它并不是Spring提出的技术，但它已成为Spring中除反转控制特性外的又一核心特性，简单来说面向切面编程就是当很多模块中需要实现相同的功能，如日志功能等，这些相同的功能被称为“切面”，面向切面编程就是把这些通用功能与业务模块分离出来。

Spring MVC模块是Spring框架提供的一个MVC框架，通过MVC模式将业务、展现和数据分离开来，它主要由DispatcherServlet、处理器映射、处理器(控制器)、视图解析器、视图组成。其原理如图1:

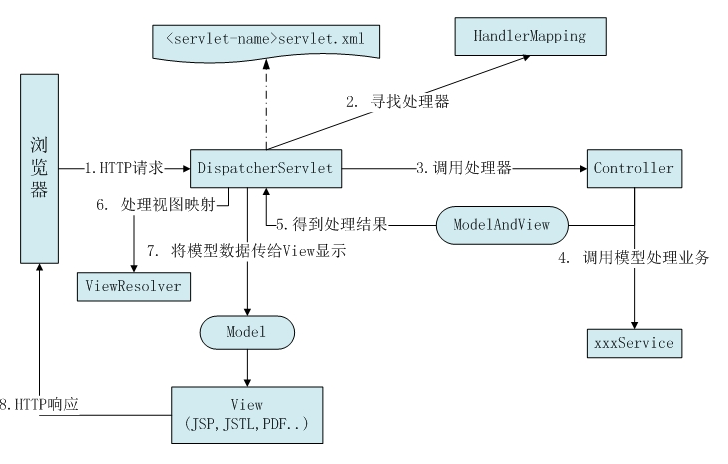


图1 Spring MVC原理图

**2.2.2 Mybatis技术介绍**

Mybatis源自于Apache的一个开源项目iBatis，它是一个优秀的持久层的框架，它主要对JDBC的数据库操作进行了高度的封装，让开发者可以只关注SQL语句本身而不用去写繁琐的JDBC代码。它通过xml配置或注解的方式将要执行的各种JDBC操作配置起来，并通过Java对象和Statement中的SQL语句进行映射生成最终的SQL语句，最后将SQL语句执行的结果映射成Java的对象进行返回。

**3 需求分析**

**3.1 可行性需求**

**3.1.1 技术可行性分析**

任务管理系统采用B/S三层体系架构，编程语言采用JAVA，数据库采用MySQL，前端采用HTML+Jquery，服务器采用Tomcat，以上技术都是当下比较成熟的计算机技术。其中采用B/S架构可以使用户在任意电脑上使用浏览器对本系统进行访问和操作，同时B/S架构在需要添加新的功能时只需更新服务器功能并添加网页即可实现系统更新，所以它具有方便维护和兼容性强等优点，而JAVA编程语言的特点之一就是跨平台，可以在任意安装了JAVA环境的系统里运行。所以在技术上分析任务管理系统是完全可行的。

**3.1.2 社会可行性分析**

当今社会在互联网+的大时代背景下，企业如果还是使用传统的任务管理方式就跟不上时代的步伐，从而不能够在竞争中获得优势，选择任务管理系统对企业任务实施管理，可以节省企业在任务管理上的时间和成本，从而提高企业运行的效率，降低公司运营的成本。而任务管理的信息化、自动化已成为当代企业所必需具备的特性。

**3.1.3 经济可行性分析**

对企业来讲，满足客户需求是首要任务。而在当今社会背景下客户的需求是多变的，企业需要随时根据市场和客户的需求对策略做出必要的调整。而一个任务管理系统可以缩短企业的策略调整的时间，从而提升企业竞争力，为企业节省大量的不必要的开销。

**3.1.4 法律可行性分析**

任务管理系统是本人在指导老师的指导下完成的，不涉及任何私人或公司的个人信息，不存在任何侵权和知识产权等问题。

**3.2 功能需求**

任务管理系统的总目标是:为企业实现任务的分配，任务的在线发布，任务的划分，任务的反馈、任务进度的跟踪以及消息的提醒。经过可行性的分析，结合现有的实际情况，任务管理系统采用B/S结构开发，开发语言为JAVA，数据库采用MySQL。系统的主要模块有员工登陆模块、我的模块、任务中心模块、任务反馈模块、消息中心模块五个方面。

我的:查看个人信息、修改头像、修改密码。

任务中心:发布任务、我发布的任务、我负责的任务。

任务反馈:提交反馈、查看反馈、删除反馈。

消息中心:未查看消息数量、消息提醒、删除消息。

**3.3 功能模块图**

根据需求分析任务管理系统主要由用户登陆模块，我的模块，任务中心模块，反馈模块和消息中心模块组成，每一模块完成特定的功能，最终由这些模块构成完整的任务管理系统，其具体功能模块图如图2:



图2 功能模块图

**3.4 系统流程图**

根据功能需求设计出系统流程，其流程图如图3:

图3 系统流程图

**4 系统设计**

**4.1 系统模块设计**

**4.1.1 员工登陆模块**

员工输入邮箱，密码，验证码进行登陆，后台通过邮箱和密码进行登陆验证，并且通过对当前用户的权限判断来确定用户可以进行哪些操作。

**4.1.2 我的模块**

我的模块包括个人信息查看、修改头像、修改密码等功能，主要实现用户对某些个人信息的修改。

1. 个人信息查看:主要显示用户姓名、性别、部门、职位、邮箱、头像等信息。
2. 修改头像:用户点击修改头像按钮后选择本地图片上传后头像显示为上传图片。
3. 修改密码:用户输入旧密码和两次新密码若所有密码均正确则更新用户密码。

**4.1.3 任务中心模块**

任务中心模块主要包括发布任务、我负责的任务、我发布的任务等功能主要实现了对任务的一系列管理。

1.发布任务:用户输入任务名称、任务截止日期、任务描述、负责人、关联任务（可选）等信息，点击确定发布信息。

2.我负责的任务:显示我负责任务的相关信息、对任务进度进行更新、提交任务反馈、将任务划分成子任务。

3.我发布的任务:显示我发布的任务的相关信息、将任务划分成子任务、将任务删除，对任务进行修改。

**4.1.4 反馈模块**

反馈模块主要包含提交反馈、查看反馈、删除反馈功能，任务反馈主要是为当用户在执行任务过程中遇到自身权限不可解决的问题时，可以向更高权限的用户提出请求和帮助。

1.提交反馈:用户在我负责的任务模块中选择自己所负责的任务点击反馈将自己所遇到的问题提交给任务负责人。

2.查看反馈:任务发布人点击反馈按钮可以查看自己所发布的任务的相关反馈。

3.删除反馈:任务发布人可以将已解决了问题的反馈删除。

**4.1.5 消息中心模块**

消息中心模块主要有未查看消息数量、消息提醒、删除消息功能。消息中心主要是当用户有了新的任务或当用户的任务信息变更时，及时地将消息传递给用户。

1.未查看消息数量:用户在登陆成功后会显示用户未查看消息数量来提醒用户注意查看消息。

2.消息提醒:当任务发布或修改时，将任务的相关信息发送给任务负责人。

3.删除消息:当用户读取消息后，用户可以选择删除消息。

**4.2 数据库设计**

**4.2.1 数据库实体图**

员工（ID，邮箱，密码，姓名，性别，职位，权限，头像，状态）

员工

图4 员工实体图

任务（ID，名称，描述，发布者，负责人，发布日期，截止日期，完成度）

任务

图5 任务实体图

任务关系（父任务，子任务）

任务关系

图6 任务关系实体图

部门（ID，名称）

部门

图7 部门实体图

消息（ID，日期，内容，状态，类型）

消息

图8 消息实体图

任务反馈（ID，内容，反馈人，任务）

任务反馈

图9 任务反馈实体图

**4.2.2 实体关系图**

实体间的关系如图10:

员工

部门

属于

1

N

任务

发布

1

N

反馈

提交

属于

1

N

N

1

任务关系

属于

N

N

员工

消息

属于

N

1

图10 实体关系图

**4.2.3 数据表设计**

员工表主要记录可登录的员工信息。

员工表employee，如表1:

表1 员工表: employee

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| ID | int | 11 | 主键 |
| Dep\_ID | int | 11 | 部门 |
| email | char | 32 | 邮箱 |
| password | char | 32 | 密码 |
| name | char | 6 | 姓名 |
| sex | char | 1 | 性别 |
| position | char | 16 | 职位 |
| power | int | 11 | 权限 |
| head | char | 32 | 头像 |
| state | char | 4 | 状态码 |

任务表主要记录任务相关信息

任务表task，如表2:

表2 任务表: task

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| ID | int | 11 | 主键 |
| name | char | 64 | 名称 |
| description | varchar | 256 | 描述 |
| publisher | int | 11 | 发布者 |
| executor | int | 11 | 负责人 |
| releaseDate | datetime |  | 发布日期 |
| closingDate | datetime |  | 截止日期 |
| completionDegree | decimal | 3 | 进度 |

任务关系表主要记录任务之间的关系

任务关系表taskrelation，如表3:

表3 任务关系表: taskrelation

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| ID | int | 11 | 主键 |
| parentTask | int | 11 | 父任务 |
| childTask | int | 11 | 子任务 |

部门表主要记录部门信息。

部门表department，如表4:

表4 部门表: department

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| ID | int | 11 | 主键 |
| name | char | 16 | 名称 |

消息表主要记录消息通知信息。

消息表news，如表5:

表5 消息表: news

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| ID | int | 11 | 主键 |
| Emp\_ID | int | 11 | 员工 |
| date | datetime |  | 日期 |
| subject | varchar | 256 | 内容 |
| state | tinyint | 1 | 状态 |
| type | varchar | 64 | 类型 |

任务反馈表主要记录任务反馈信息。

任务反馈表taskfeedback，如表6:

表6 任务反馈表:taskfeedback

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 说明 |
| ID | int | 11 | 主键 |
| Emp\_ID | int | 11 | 员工 |
| Task\_ID | int | 11 | 任务 |
| content | varchar | 256 | 内容 |
| date | datetime |  | 日期 |

**5 系统实现**

**5.1 登陆实现**

登陆界面如图11:



图11 登陆界面

用户输入邮箱，密码和验证码，后台对用户输入信息进行验证，若无此用户则返回1，若密码错误则返回2，若验证码是错误的则返回3，若验证通过则返回0，当验证通过时，将用户ID存进会话中。具体代码如下:

int flag = 0;

Employee employee = service.getEmployee(email);

if (employee == null)flag = 1;

else if (!pwd.equals(employee.getPassword())) {flag = 2;

} else if (!BuildCodeHelper.*INSTANCE*.check(session, code)) {flag = 3;

} else {flag = 0;

session.setAttribute("id", employee.getId());

session.removeAttribute("code");}

**5.2 我的模块实现**

我的界面如图12:

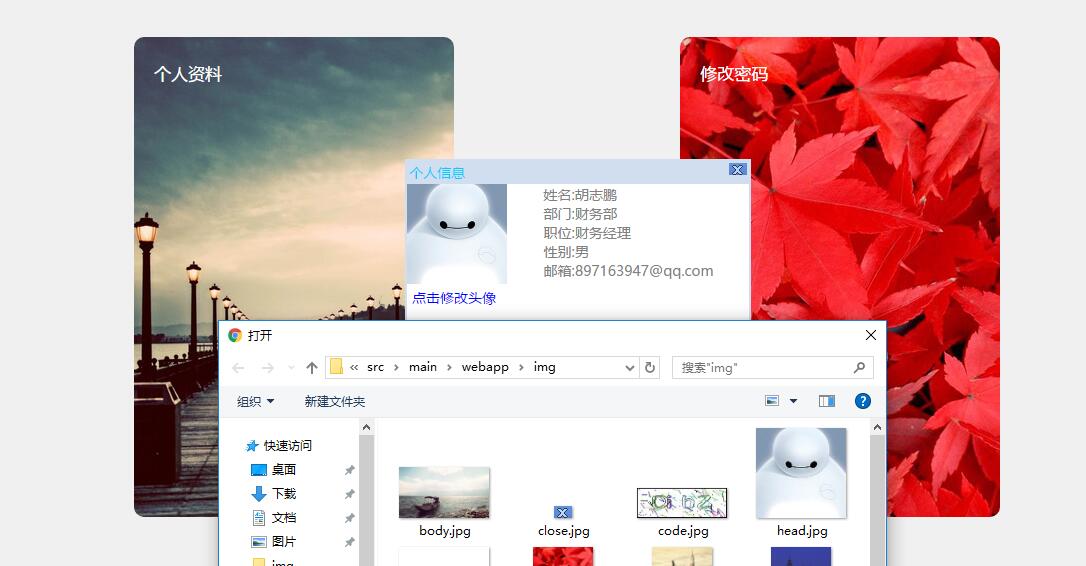


图12 我的界面

用户登陆成功后将ID存进会话里，通过会话里的ID可以查询出用户相关的信息，用户点击修改头像选择本地图片可将图片上传至服务器。具体实现代码如下:

public UserInfo getUserInfo(HttpSession session) {

UserInfo userInfo = service.getUserInfo((Integer) session.getAttribute("id"));

return userInfo;

}

public void uploadHead(@RequestParam CommonsMultipartFile file,

HttpServletRequest request) throws IOException {

String filename = file.getOriginalFilename();

UploadFileHelper helper = UploadFileHelper.*getUploadFileHelper*();

String path = "head/" + request.getSession().getAttribute("id")

+ "\_head" + filename.substring(filename.lastIndexOf("."));

String realpath = request.~~getRealPath~~("") + "/" + path;

helper.saveFile(realpath, file);

service.resetHead(path,(Integer) request.getSession().getAttribute("id"));

}

**5.3 任务中心实现**

任务中心界面如图13:

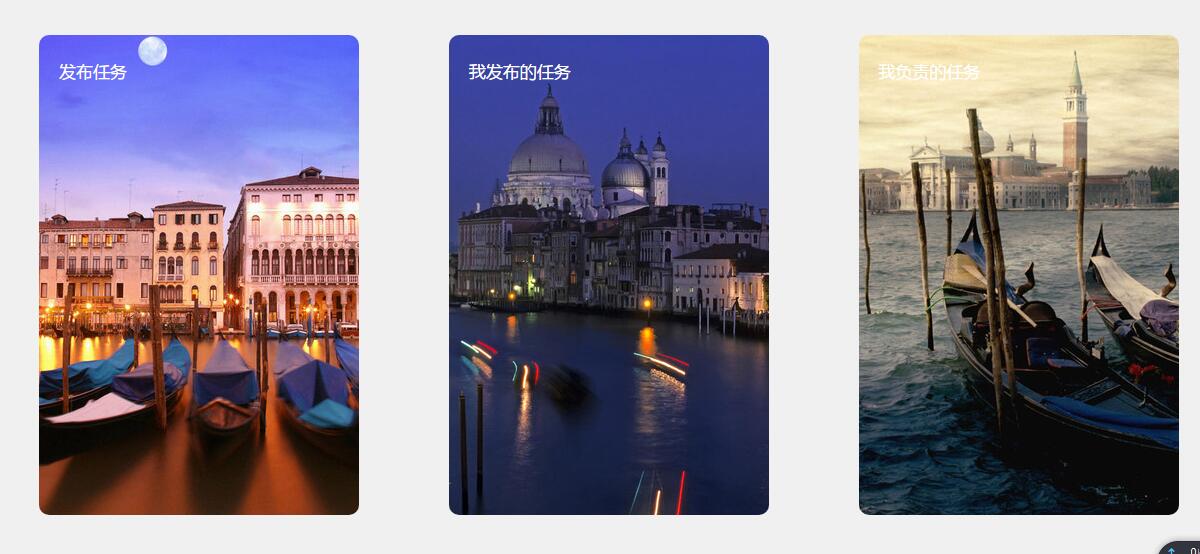


图13 任务中心界面

任务中心主要提供发布任务，我发布的任务，我负责的任务等功能，主要通过存在会话里的用户ID来查询用户发布的任务和用户负责的任务，同时通过用户ID查询用户权限来提供当用户发布任务时可选择的负责人。代码如下:

public int addTask(Task task, Integer father, HttpSession session)

throws IOException {

String date = DateFormat.*format*(new Date());

task.setPublisher((Integer) session.getAttribute("id"));

task.setReleasedate(date);

task.setCompletiondegree((short) 0);

int flag = service.addTask(task);

if (father != null && !"".equals(father)) {

TaskRelation relation = new TaskRelation();

relation.setChildtask(task.getId());

relation.setParenttask(father);

service.addRestion(relation);

}

}

public Page<?> getTasks(HttpSession session, @PathVariable int now) {

return service.getTasks((Integer) session.getAttribute("id"), now);

}

public Page<?> getExecutorTasks(HttpSession session, @PathVariable int now) {

return service.getExecutorTasks((Integer) session.getAttribute("id"),now);}

**5.4 反馈实现**

反馈界面如图14:



图14 反馈界面

反馈模块主要是用户在其负责的任务中向发布人提交反馈，任务发布人可以在反馈中看到自己发布的任务的反馈，并可以删除反馈。代码如下:

@ResponseBody

@RequestMapping(value = "/addFeedBack", method = RequestMethod.*POST*)

public int addFeedBack(HttpSession session, TaskFeedback feedback) {

feedback.setEmpId((Integer) session.getAttribute("id"));

return service.addFeedBack(feedback);}

@ResponseBody

@RequestMapping(value = "/ getFeedBackInfos/{now}", method = RequestMethod.*GET*)

public Page<FeedBackInfo> getFeedBackInfos(HttpSession session,@PathVariable Integer now){

return service.getFeedBackInfos((Integer)session.getAttribute("id"),now);}

@ResponseBody

@RequestMapping(value = "/deleteFeedBack/{id}", method = RequestMethod.*GET*)

public int deleteFeedBack(@PathVariable Integer id){

return service.deleteFeedBack(id);}

**5.5 消息中心实现**

消息中心界面如图15:



图15 消息中心界面

消息中心主要是通过WebSocket进行消息的在线发送，当用户登陆时将会话分享给WebSocket,当建立WebSocket连接时会产生一个WebSocket会话，并将两个会话用Map保存起来，当执行会产生消息的操作时，消息中心可以通过Map找到对应用户的WebSocket会话将消息发送给相应的用户。代码如下:

public boolean beforeHandshake(ServerHttpRequest request,,ServerHttpResponse response, WebSocketHandler handler,Map<String, Object> map) throws Exception {

if (request instanceof ServletServerHttpRequest) {

ServletServerHttpRequesthttpRequest=(ServletServerHttpRequest) request;

HttpSession session=httpRequest.getServletRequest().getSession();

if (session!=null) {Integer id=(Integer) session.getAttribute("id");

map.put("id", id);}return true;}}

public void afterConnectionEstablished(WebSocketSession session)

throws Exception {

Integer id=(Integer) session.getHandshakeAttributes().get("id");

*usersMap*.put(id, session);

}

public void registerWebSocketHandlers(WebSocketHandlerRegistry registry) {

registry.addHandler(webSocketHandler(), "/news")

.addInterceptors(newWebSocketHandshakeInterceptor())

.withSockJS();}

**6 系统测试**

**6.1 登陆测试**

登陆模块测试过程及结果如表7:

表7 登陆模块测试

|  |  |
| --- | --- |
| 测试名称 | 登陆模块 |
| 测试目的 | 可以实现登陆信息的验证 |
| 测试方法 | 黑盒测试 |
| 测试权限 | 任意用户 |
| 测试步骤 | 1. 输入不存在的邮箱，其余信息正确填写。 2. 输入错误密码，其余信息正确填写。 3. 输入错误验证码，其余信息正确填写。 4. 所有信息填写正确。 |
| 预期结果 | 1. 提示用户名错误。 2. 提示密码错误。 3. 提示验证码错误。 4. 跳转到主页。 |
| 测试结论 | 测试通过 |
| 测试界面 | |
| 图16邮箱验证    图17密码验证    图18验证码验证 | |

**6.2 我的模块测试**

我的模块测试步骤和结果如表8:

表8 我的模块模块测试

|  |  |
| --- | --- |
| 测试名称 | 我的模块模块测试 |
| 测试目的 | 可以实现个人资料查询和头像上传 |
| 测试方法 | 黑盒测试 |
| 测试权限 | 任意用户 |
| 测试步骤 | * + - 1. 点击我的信息。       2. 点击修改头像。       3. 选择本地图片。       4. 点击确定。 |
| 预期结果 | 显示正确的个人资料。  头像上传成功。 |
| 测试结论 | 测试通过 |
| 测试界面 | |
| 图19 个人信息    图20 头像修改 | |

**6.3 任务中心测试**

任务中心测试步骤和结果如表9:

表9 任务中心测试

|  |  |
| --- | --- |
| 测试名称 | 任务中心模块 |
| 测试目的 | 可以实现任务中心的相关功能 |
| 测试方法 | 黑盒测试 |
| 测试权限 | 权限大于1的用户 |
| 测试步骤 | 1. 点击发布任务。  2. 输入任务相关信息  3. 点击确定。  4. 点击我发布的任务查看。  5. 换任务负责人登陆。  6. 点击我负责的任务查看。 |
| 预期结果 | * + - 1. 发布任务成功。       2. 能查看到自己发布的任务。       3. 能查看到负责任务。 |
| 测试结论 | 测试通过 |
| 测试界面 | |
| 图21 发布任务    图22 我发布的任务    图23 我负责的任务 | |

**6.4 反馈模块测试**

反馈模块测试和步骤如表10:

表10 反馈模块测试

|  |  |
| --- | --- |
| 测试名称 | 反馈模块测试 |
| 测试目的 | 可以成功发布反馈并且发布人可以看到反馈信息 |
| 测试方法 | 黑盒测试 |
| 测试权限 | 任意用户 |
| 测试步骤 | 任务负责人登陆。  向任务发布人提交反馈。  任务发布人登陆。  任务发布人查看反馈。 |
| 预期结果 | 1. 提交反馈成功。 2. 任务发布人能看到任务负责人提交的反馈。 |
| 测试结论 | 测试通过 |
| 测试界面 | |
| 图24 发布反馈    图25 查询反馈信息 | |

**6.5 消息中心测试**

消息中心测试步骤和结果如表11:

表11 消息中心测试

|  |  |
| --- | --- |
| 测试名称 | 消息中心模块 |
| 测试目的 | 可以实现消息发送功能 |
| 测试方法 | 黑盒测试 |
| 测试权限 | 任意用户 |
| 测试步骤 | 两个权限不同的账号同时登陆。  权限高的用户向权限低的用户发布任务。  低的一方查看消息。 |
| 预期结果 | 1. 发布任务成功。 2. 权限低的一方收到有新任务的消息。 3. 点击消息可以看到消息内容。 |
| 测试结论 | 测试通过 |
| 测试界面 | |
| 图26 发布任务    图27 消息提醒    图28 消息内容 | |

**6.6 测试结语**

以上测试均已通过，任务管理系统的功能基本完成，可以正常运行。

**结 语**

经过近6个月地研究与开发，终于完成了任务管理系统的任务目标，在这6个月里首先研究分析任务计划书，然后去网上查看现有的任务管理系统，收集它们信息，整理后开始构思自己的任务管理系统，然后设计出任务管理系统的功能模块，设计功能模块后，开始使用Power Designers设计数据库的概念模型，然后使用HBuilder设计出系统的前端，最后使用My Eclipse搭建出后台的Spring+SpringMVC+MyBatis后台框架，并完成后台程序。在这个过程中我学到了很多的东西，首先就是前端的知识如样式表和jQuery,然后是JavaEE的知识。在这次开发中我遇到过很多的问题，但最后都一一解决了，在解决问题的过程中我获得了我以前所没有掌握的知识和技巧，在这里我要谢谢我的指导老师，他在我的研究和开发过程中给予了我很大的帮助，我在他的身上学到了很多的东西，最后任务管理系统的完成极大地提升了我的技术水平，我在完成它的过程中获益良多。

[**参考文献**](论文格式摸板(2007).doc)

[1] (美)Roger S.Pressman.软件工程实践者的研究方法[M].机械工业出版社，2011.04

[2] 孙卫琴.Java面向对象编程[M].电子工业出版社，2006.07

[3] 王小东.高性能MySQL[M].电子工业出版社，2010.02

[4] (美)Joshua Bloch.Java高效编程(第二版)[M].机械工业出版社，2009.01

[5] 贾素玲，王强.HTML网页设计[M].清华大学出版社，2007.05

[6] (美)弗拉纳根.Javascript权威指南[M].机械工业出版社，2007.08

[7] (美)奥特罗,(美)劳伦斯.jQuery高级编程[M].清华大学出版社，2013.04

[8] (比利时)戴克.Spring MVC学习指南[M].人民邮电出版社，2015.05

[9] 杨开振.深入浅出MyBatis技术原理与实战[M].电子工业出版社，2016.08

[10] 罗时飞.精通Spring-深入JavaEE开发核心技术[M].电子工业出版社，2008.10

**致 谢**

本文是在苟智坚老师的热情关心和指导下完成的，他渊博的知识和严谨的治学态度使我受益匪浅，对顺利完成本课题起到了极大的作用。在此向他表示我最衷心的感谢！

在论文完成过程中，本人还得到了李享梅老师和刘伟同学的热心帮助，本人向他们表示深深的谢意！

最后向在百忙之中评审本文的各位专家、老师表示衷心的感谢！

作者简介:成都信息工程大学网络工程专业学生

姓 名:胡志鹏 性别:男

出生年月:1994.06.05 民族:汉

E-mail:897163947@qq.com

**声 明**

本论文的工作是2016年12月至2017年 06月在成都信息工程大学网络空间安全学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明:

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括:

（1）学校有权保管并向有关部门递交学位论文的原件与复印件。

（2）学校可以采用影印、缩印或其他复制方式保存学位论文。

（3）学校可以学术交流为目的复制、赠送和交换学位论文。

（4）学校可允许学位论文被查阅或借阅。

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

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

特此声明！

作者签名：

2017年06月10日