



成都學院  
CHENGDU UNIVERSITY

# 本科毕业（设计）论文

题    目 基于 JavaWeb 的会议管理系统的  
设计与实现

学    院 信息科学与工程学院

专    业 计算机科学与技术

学生姓名 杨宽

学    号 201410411221 年 级 2014 级

校内指导教师 游磊 职 称 副教授

校外指导教师          职 称 高级工程师

二〇一八年五月

# 基于 JavaWeb 的会议管理系统的设计与实现

专    业： 计算机科学与技术        学    号： 201410411221

学    生： 杨宽                        校内指导教师： 游磊

校外指导教师： 何宙兴

**摘要：**在当今参加会议是非常普遍的现象，会议管理保证了会议的顺利进行也决定了工作的走向。对于公司或高校，会议管理还存在着很多问题，由于处于手工处理阶段的会议管理还有很多，不仅导致效率低容易出错，而且还非常不利于管理。该系统采用 Eclipse 和 MySQL 数据库管理系统作为工具开发的基于 JavaWeb 的会议管理系统。该系统能够提供更方便的管理，整个系统的窗口清新，其功能有登录、注册、对员工信息的增删查改、发布会议、申请会议、会议座次的选择以及新闻管理功能。整个系统操作简单、效率高、不易出错、易于管理、维护与扩展。

**关键词：**会议管理；B/S 结构；WebSocket 协议；Shiro 权限

# **Design and Implementation of Conference Management System Based on Web**

Specialty: Computer Science and Technology

Student Number: 201410411221

Supervisor: You Lei

Student: Yang Kuan

Enterprise Supervisor: He Zhouxing

**Abstract:**At present, it is a very common phenomenon to attend a meeting, and the conference management ensures the smooth progress of the meeting and determines the direction of the work. For companies or universities, it still exists many problems, for lots of conference managements are in the stage of manual work , which results in not only low efficiency and error-prone tendency, but also bad management. This system uses the Eclipse and MySQL database management system as tool and bases on JavaWeb. The system can provide a more convenient management, and the window of the whole system is pure and fresh, which has the function of login, register, employees' change and deletion, conference releasing, conference applying, seats choosing, and news managing. The whole system is simple, efficient, less error-prone, and easy to manage, maintain and expand.

**Key words:** Conference management; B/S structure; WebSocket agreement; Shiro permissions

## 目 录

|       |                       |           |
|-------|-----------------------|-----------|
| 1     | 绪论.....               | 1         |
| 1.1   | 研究目的与意义 .....         | 1         |
| 1.2   | 国内外研究现状 .....         | 2         |
| 1.3   | 课题研究内容 .....          | 3         |
| 2     | 开发工具及相关技术.....        | 4         |
| 2.1   | 开发工具 .....            | 4         |
| 2.1.1 | Eclipse 集成开发环境 .....  | 4         |
| 2.1.2 | MySQL 关系型数据管理系统 ..... | 错误!未定义书签。 |
| 2.2   | 相关技术 .....            | 错误!未定义书签。 |
| 2.2.1 | Java 开发语言 .....       | 错误!未定义书签。 |
| 2.2.2 | MVC 设计模式 .....        | 错误!未定义书签。 |
| 2.2.3 | JSP 技术 .....          | 错误!未定义书签。 |
| 2.2.4 | Spring 框架 .....       | 错误!未定义书签。 |
| 2.2.5 | B/S 体系结构 .....        | 错误!未定义书签。 |
| 2.2.6 | Ajax 技术 .....         | 错误!未定义书签。 |
| 2.2.7 | WebSocket 协议技术 .....  | 错误!未定义书签。 |
| 2.2.8 | Shiro 安全框架 .....      | 错误!未定义书签。 |
| 2.2.9 | MyBatis 持久层框架 .....   | 错误!未定义书签。 |
| 3     | 需求分析.....             | 5         |
| 3.1   | 系统定义 .....            | 5         |
| 3.2   | 系统功能需求 .....          | 5         |
| 3.3   | 系统性能需求 .....          | 6         |
| 4     | 数据库设计.....            | 7         |
| 4.1   | 实体关系 E-R 图 .....      | 7         |
| 4.2   | 表设计 .....             | 9         |
| 4.2.1 | 用户表 (y_user) .....    | 9         |
| 4.2.2 | 部门表 (y_dept) .....    | 9         |

|       |                        |           |
|-------|------------------------|-----------|
| 4.2.3 | 角色表 (y_role)           | 错误!未定义书签。 |
| 4.2.4 | 权限表 (y_permission)     | 错误!未定义书签。 |
| 4.2.5 | 会议信息表 (y_meetinginfo)  | 错误!未定义书签。 |
| 4.2.6 | 申请会议信息表 (y_askmessage) | 错误!未定义书签。 |
| 4.2.7 | 会议室表 (y_meetingRoom)   | 错误!未定义书签。 |
| 4.2.8 | 会议文件表 (y_file)         | 错误!未定义书签。 |
| 5     | 功能的设计与实现               | 10        |
| 5.1   | 系统结构设计                 | 10        |
| 5.2   | 系统功能模块与系统流程图           | 10        |
| 5.2.1 | 系统流程图                  | 10        |
| 5.2.2 | 内部管理模块图                | 12        |
| 5.2.3 | 个人中心模块图                | 12        |
| 5.2.4 | 新闻模块图                  | 13        |
| 5.3   | 功能的详细设计                | 13        |
| 5.3.1 | 登录与注册                  | 13        |
| 5.3.2 | 会议发布与会议申请              | 错误!未定义书签。 |
| 5.3.3 | 会议查看与删除                | 错误!未定义书签。 |
| 5.3.4 | 会议室选座                  | 错误!未定义书签。 |
| 5.3.5 | 密码修改                   | 错误!未定义书签。 |
| 5.3.6 | 员工的增删改查                | 错误!未定义书签。 |
| 5.3.7 | 角色权限的分配                | 错误!未定义书签。 |
| 5.3.8 | 新闻的发布、查看               | 错误!未定义书签。 |
| 6     | 系统测试                   | 错误!未定义书签。 |
| 7     | 总结                     | 17        |
|       | 参考文献                   | 18        |
|       | 致谢                     | 20        |
|       | 附录                     | 21        |

## 1 绪论

### 1.1 研究目的与意义

随着当今社会竞争的日益激烈，企业和机构的内部会议也越来越多，会议信息的数量也越来越多。现在几乎是所有的企业都需要通过会议进行沟通，得出工作的结论，作出决定。在现阶段，企事业单位每天或每周都会有非常多的会议进行处理，如果像传统会议一样，通过手工对会议进行操作，这样的效率非常低，并且难于管理，非常容易出错。

据以往的调查显示，每一家公司的大部分人员每周花在会议上面的时间几乎有四分之一。又有另一项调查发现，80%的员工的工作效率与他们进行会议时的表现有很大的关系。因此，如何让会议更加高效，如何进行会议的召开，如何管理会议的资料，对员工和企业都具有重要意义。

对于管理来说，会议进行的内容与沟通得出的结论和反馈都是非常重要的。因此，会议管理系统肯定能成为管理人员的好帮手，它能提供更高效率的查询与记录。但在很长一段时间内，会议的管理存在许多不足之处。例如，会议记录查询需要根据会议记录的大量会议记录来安排。这是管理人员和用户的负担，效率低下，容易出错，且保密程度低。

随着中国改革开放的深入和经济的持续快速发展，中国出现了大、中、小型企业。在这个满是信息的时代，企业对信息的管理是尤为重要的。如果想要好的收益，就一定需要一个好的商业管理者，尤其是在今天的信息技术中。因此，开发一个“会议管理系统”是非常必要的。这样的系统可以提供高效的查找、高精度、低成本和高效率。这些优点可以大大提高会议管理的效率。

对企业来说，需要创新创造能力，而一个好的想法需要集结所有头脑进行总结。会议就是头脑风暴的一个渠道，会议是聚会的工具，是集思广益的通道。通过会议，不同的人 and 不同的想法被聚集在一起，相互碰撞，从而产生“黄金思想”。由于会议的召开，会议记录应该以方便快捷的方式进行管理，所以会议管理系统是一个很好的选择。

## 1.2 国内外研究现状

针对如何提高会议的效率,国内的知名公司或软件开发商研究并开发出了许多不同的会议管理系统<sup>[1]</sup>。例如,康拓普公司推出的一套无纸化会议系统,该系统通过文件的网络传输实现会议的无纸化,其具有会议信息的上传、会议批注、会场签到、资料的发放、表决等功能。该系统现如今已被中信集团、中国人民银行等企事业单位使用。国内的发展较慢,产品也主要面向国内企业,在国内中小型企业占大多数,因此会议管理系统得到迅速的发展<sup>[2]</sup>。国内现在的会议管理系统非常之多,会鸽所研究的会议周期管理系统,主要功能包括通过电子邮件将资料发送到参会人员、会场的快速签到、资料的索取、投票、抽奖与信息交换等,其会场签到的核心技术是 NFC 通讯技术来完成的,使会议简单高效,会议的管理更方便。

目前,智能的会议管理系统已经开始全面普及,包括国家与省机关处<sup>[3]</sup>。如今全国各地城市与区县级国家机关也慢慢开始起步。国家机关对会议管理系统的需求持续增长,促使会议系统的应用不断发展,不断加强其功能,会议管理系统慢慢成为了国家利润的一个增长点。从 2013 年开始,国家的两会便开始使用无纸化会议系统,省去了会议组织者会前繁琐的资料准备工作,会后更是可以一键销毁以确保会议资料的安全性,或者可以将会议记录和会议文件进行加密<sup>[4]</sup>,不仅省时省力,还可以节约大量纸张。由康拓普提供的无纸化会议管理系统有力的支撑了国家关于加强节能工作的安排,高效落实了节能减排的政策措施,履行中央企业带头节约资源和保护环境,带头完成节能减排任务的社会责任<sup>[5]</sup>。

在 70 年代,国外就开始研究网络会议管理系统,到现在有许多大公司(如 VCON、VTEL 等)都在从事这方面的研究工作<sup>[6]</sup>。在国外,据以往调查数据表明,2004 年的上半年,在欧洲网络会议管理系统就已经迅速发展,网络会议管理系统的总收入就已经占据了各大运营商 20% 的份额<sup>[7]</sup>。在 2005 年亚太地区的网络会议管理系统得到了长足的发展。在国际上,有很多国家在会议的在线管理方面已经有了相当成熟的系统,这其中包括商业的和一些开源的系统<sup>[8]</sup>。如 VSIS ConfTool、MyReview 是一些开源的免费系统,这些软件都相当成熟,并且已经应用于学术会议管理中<sup>[9]</sup>。

无论是国内外,会议系统的发展是必不可挡的趋势,从国内近几年的发展速度来观察,会议系统成为了国家重点关注的对象。会议系统已经在一些外资企业和高新技术企业广泛应用,未来必定成为企业会议形式的主流<sup>[10]</sup>。

### 1.3 课题研究内容

为了解决传统会议的不足，本课题探究的是将会议的管理从传统会议中抽取出来，并加以实现，课题研究主要的目的是让其担任一个对会议进行管理的角色，采用 Spring+SpringMVC+MyBatis 框架对会议管理系统进行开发，将传统的会议整理、会议资料、发布会议、会议通知、会议召开等繁琐步骤加以简化，研究开发一套安全性强、实用性强、容易扩展并以会议为中心的会议管理系统。



## 2 开发工具及相关技术

### 2.1 开发工具

#### 2.1.1 Eclipse 集成开发环境

Eclipse 现在由非营利软件供应商联盟 Eclipse 基金会管理，早期是由 IBM 公司开发的。

Eclipse 是基于 Java 的一个开发平台，现如今还是大部分 Java 项目开发的首选工具，原因在于它本身是开放源代码的，并且 Eclipse 是跨平台的自由开发环境。它本身自带了一组标准的组件，并且还能集成其它插件，因此它的功能非常强大，不仅仅只限于本身插件的功能。但是由于它能集成的插件非常多，导致其灵活性降低，所以有很多软件开发商开发了自己的 IDE 集成开发环境。

选择 Eclipse 作为该系统的开发环境的原因在于：一是 Eclipse 集成了许多的插件，并且自带提示功能，极大的提高了开发的效率；二是 Eclipse 集成了 Maven 和 Git 插件，这使得项目的构建与项目版本的管理更方便，开发效率因此也得以提高。

### 3 需求分析

#### 3.1 系统定义

会议管理系统基于 B/S 结构, 采用 Java、Jsp、Spring、Spring MVC、Mybatis 和 MySQL 等工具及技术。该系统按模块划分为会议管理模块、会议发布与申请模块、员工管理模块和新闻管理模块四个部分。论文主要阐述了会议管理系统采用的技术、功能的详细设计与实现。其功能包括: 会议信息查看, 会议信息的发布、会议室座位的选择、会议信息删除、员工信息的增删改查、权限的管理和新闻的管理等。

#### 3.2 系统功能需求

会议管理系统的的一个最主要的目的就是为会议组织者提供方便快捷的服务, 会议管理系统基于 B/S 架构, 会议申请者可以通过 IE 或其它浏览器对该系统进行访问, 会议申请者提交申请后由管理员进行审批。

会议管理系统主要实现以下功能:

- (1) 登录系统: 用户登录、用户注册、用户注销。
- (2) 用户的查询: 实现用户的查询功能, 可以按用户 ID 和姓名进行查询。
- (3) 用户的添加: 添加一个新用户, 具体字段: 登录 ID, 密码, 用户姓名, 性别, 地址, 出生日期, 电话, 部门等信息。
- (4) 用户信息的修改: 实现对用户的资料修改功能, 具体字段: 登录 ID, 密码, 用户姓名, 性别, 地址, 出生日期, 电话, 部门等信息。
- (5) 显示用户的详细信息: 能够显示用户的个人详细信息, 具体字段: 登录 ID, 密码, 用户姓名, 性别, 地址, 出生日期, 电话, 部门等信息。
- (6) 用户的删除: 现在相应的用户, 对其进行删除操作。
- (7) 修改密码: 可以修改用户和管理员的登录密码。具体字段: 原密码, 新密码, 重新输了新密码。
- (8) 会议的申请: 实现对会议的通知, 具体字段: 会议 ID, 会议主题, 会议简介, 会议时间, 会议内容, 会议文件资料。

（9）会议的删除：对已经完成过的会议进行删除操作。

（10）位置的选择：选择会议室里的想坐的座位。

（11）会议的审批：管理员能够对前台用户提交的会议申请进行审批做“同意”或“不同意”的判断。

（12）新闻管理：由管理员发布新闻，员工可以发表对新闻的看法并评论。

### 3.3 系统性能需求

通过功能需求分析能够分析出来系统需要的性能：

（1）系统运行速度：实时推送，实时查询。当管理者发布会议之后，在线员工会及时收到会议信息；当不在线员工在登录系统时，马上查询到会议的信息。

（2）系统的存储容量：该系统会存储大量的会议信息、会议文件、用户评论等记录，因此会需要大量的存储空间。

（3）系统的安全：因为有关会议的所有信息，包括内容、文件和评论等，都属于单位的内部信息，所以安全性是非常重要的，因此需要设置用户权限等。

（4）系统的运行环境：硬件：CPU 选用 Intel Core i3 及以上；内存（RAM）：选用 4G 及以上。软件：Windows7 操作系统及以上，MySQL5.7，Tomcat8。

## 4 数据库设计

数据库设计是为了指明各个表的设计以及它们之间的关联，然后为之建立数据库。数据库设计是为了建立一个高速运行的环境，给用户提供良好的体验。在本章中，通过各个实体的分析画出各个实体的 E-R 图，再根据实体分析得出实体数据表。

### 4.1 实体关系 E-R 图

本系统涉及到的实体与 E-R 图如下：

(1) 用户：（编号，登录编号，登录密码，用户名，性别，出生日期，联系方式，住址，部门编号），如下图 4-1 所示：

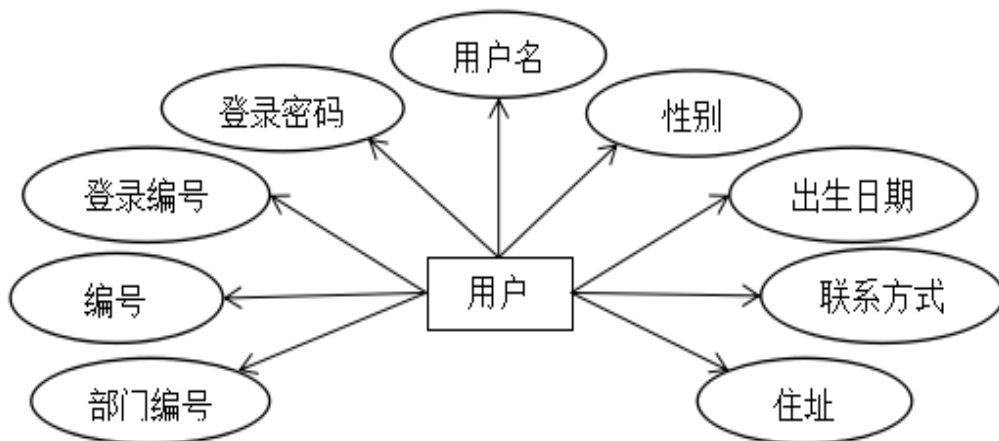


图 4-1 用户 E-R 图

(2) 部门：（部门编号，部门名，部门电话），如下图 4-2 所示：

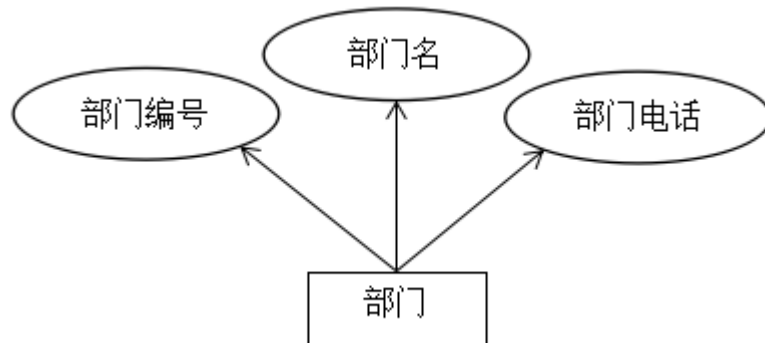


图 4-2 部门 E-R 图

(3) 角色：（角色编号，角色名，角色描述），如下图 4-3 所示：

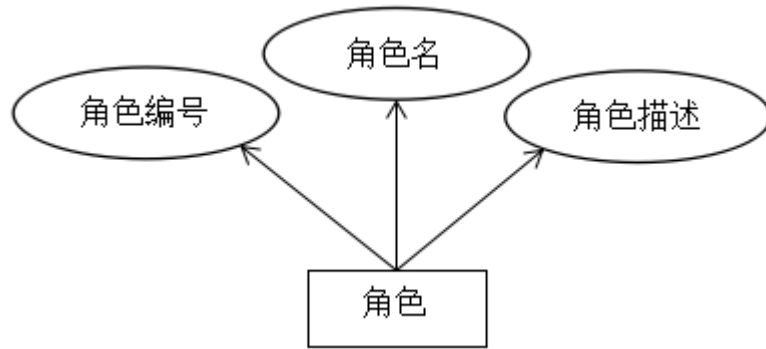


图 4-3 角色 E-R 图

(4) 权限：（权限编号，权限名），如下图 4-4 所示：

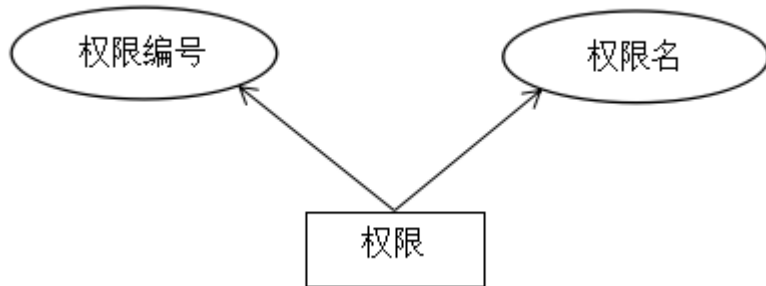


图 4-4 权限 E-R 图

(5) 会议信息：（编号，会议编号，主题，简介，会议时间，会议地点，会议内容，部门编号，发布者），如下图 4-5 所示：

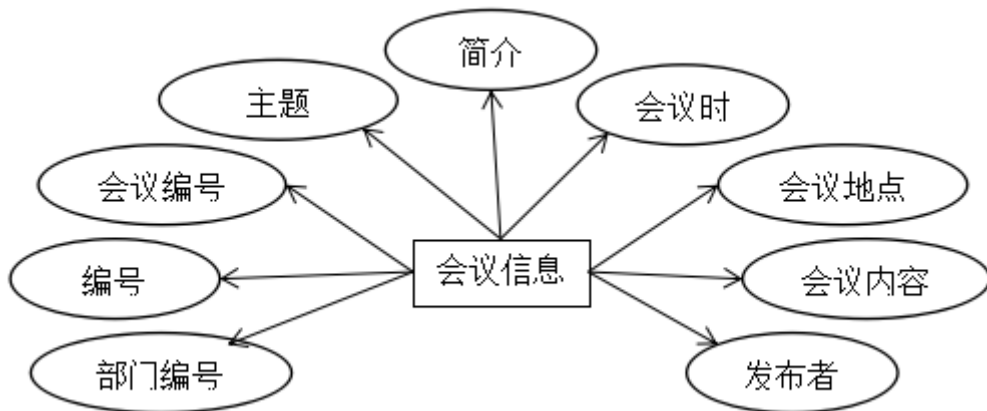


图 4-5 会议信息 E-R 图

(6) 申请会议：（部门编号，申请人编号，经理编号，用户接收，是否同意，会议编号，主题，简介，会议内容，申请人姓名），如下图 4-6 所示：

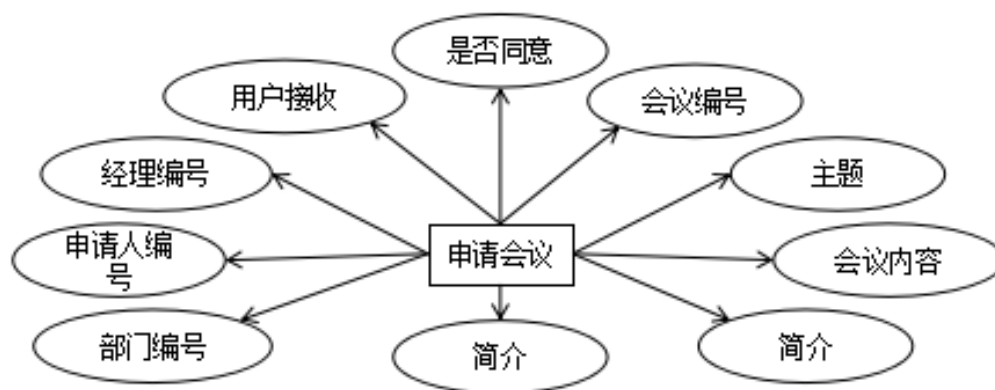


图 4-6 申请会议 E-R 图

## 4.2 表设计

### 4.2.1 用户表（y\_user）

该表主要用来存储用户的信息。该表可用于用户登录，用户查询与修改信息，存放角色用于操作具有权限的功能，还可以用于发布会议或申请会议。该表的数据字典见表 4-1 所示：

表 4-1 用户表（y\_user）

| 标题   | 字段名称     | 类型      | 大小 | 描述       |
|------|----------|---------|----|----------|
| 序号   | id       | int     | 11 | 序号       |
| 登录编号 | loginId  | int     | 11 | 用户登录的 id |
| 用户名  | username | varchar | 10 | 用户的姓名    |
| 密码   | password | varchar | 16 | 用户登录的密码  |
| 性别   | sex      | varchar | 2  | 用户的性别    |
| 生日   | birthday | date    | 0  | 用户的出生日期  |
| 电话   | tel      | varchar | 11 | 用户的电话    |
| 住址   | addr     | varchar | 20 | 用户的家庭住址  |
| 部门编号 | deptId   | int     | 11 | 部门编号     |

### 4.2.2 部门表（y\_dept）

该表用于存放部门信息。该表可以用于用户部门的查询与修改，还可用于标记部门

## 5 功能的设计与实现

本章主要介绍了系统功能的设计与实现，首先介绍了系统的总体架构设计说明，接下来展现了整个系统的流程图并描述了各个模块的功能。其次展现了每一个功能的界面显示与流程的走向，详细的说明了功能的设计思路，并附有功能的核心代码与代码的注释。

### 5.1 系统结构设计

系统总体结构设计的目的是首先使用抽象的方式确定系统要完成的任务，然后根据结构确定要使用的软件结构，使其与总体结构一样，模块之间相互独立，让模块间的接口尽量简单。系统体系设计结构：浏览器-服务器，体系结构如下图 5-1 所示：

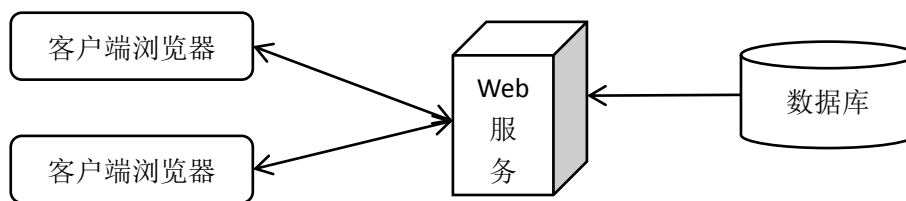


图 5-1 B/S 体系结构

该系统使用 B/S 结构有一个很大的好处，因为 B/S 结构叫浏览器/服务器结构，故只要有一台能上网的电脑，就能随时随地的访问该系统，完全不需要安装任何客户端软件，完全不需要再客户端进行维护。

现如今 B/S 结构的使用越来越多，通过浏览器发出请求，服务器对其进行处理，因此这会大量使用到 Ajax 技术，于是推动了 Ajax 技术的发展。在客户端能处理的程序非常少，从而极大的降低了客户端承受的压力。

### 5.2 系统功能模块与系统流程图

#### 5.2.1 系统流程图

在程序分析与设计中，流程图是最基本，但是也是最重要的分析技术，它展现出来是整个程序的完整操作。着眼于整个流程，可以发现它简单明了，因此可以对每一个步

骤提取出来,对其中的步骤进行更详细的研究并加以分析,以减少开发过程中不必要的操作,因此开发与维护的效率都能够大大提高。

在程序设计中,开发者需要着眼于整个项目的开发流程,因此流程图给予了一个特别方便的地方,它能直观的描绘出控制流程,并且易于掌握,这样简洁明确。

管理员系统流程图用来表示管理员具有的操作,当管理员登录后可以对会议信息进行处理,例如处理会议申请、处理员工信息等信息。如图 5-2 所示:

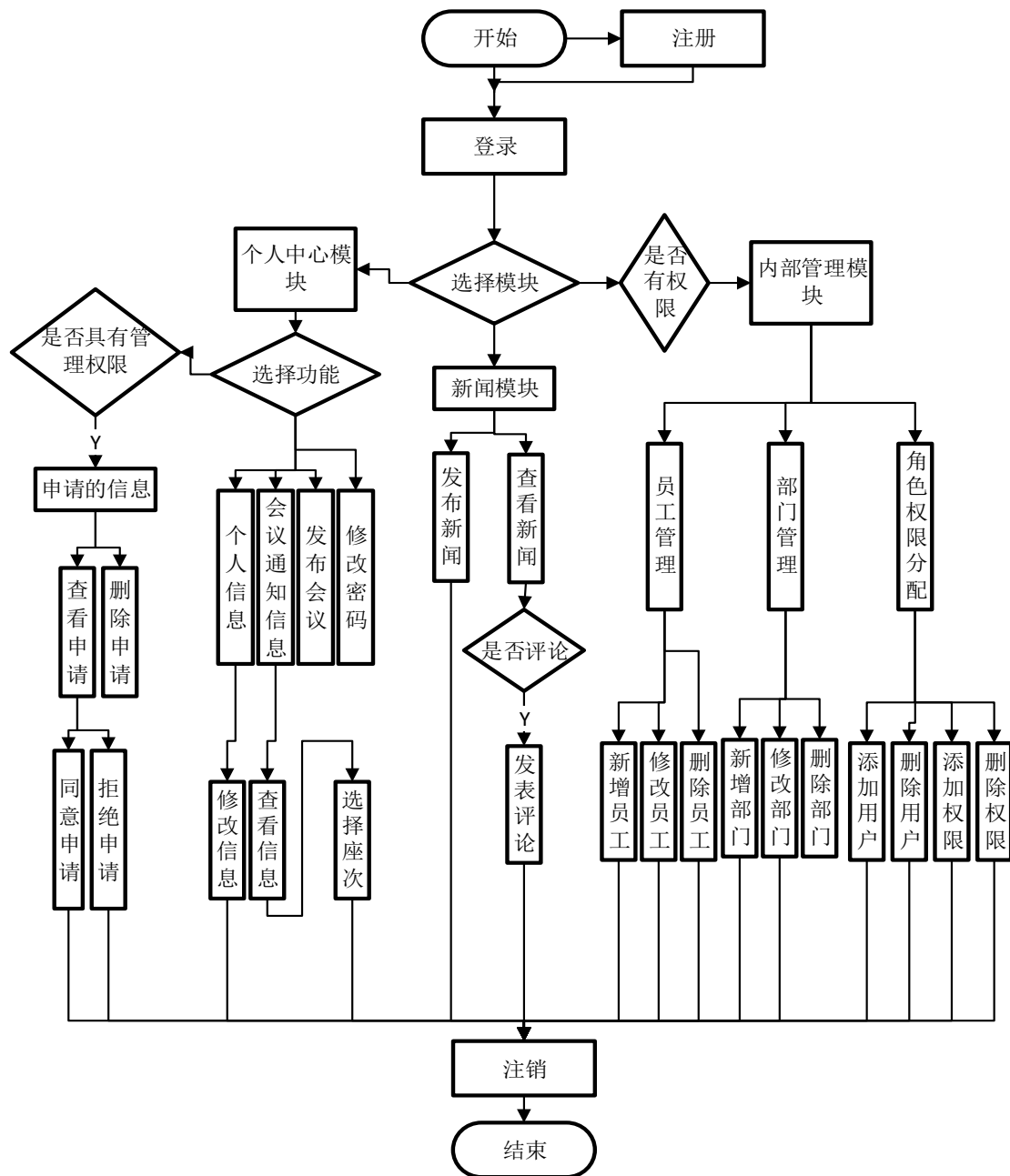


图 5-2 系统流程图



### 5.2.2 内部模块图

内部模块是提供给管理员对员工管理的一个模块，主要包括员工信息的查看、新员工信息的添加、员工信息的修改、员工密码的重置、删除员工、部门管理和给员工分配角色与权限。内部模块图如图 5-3 所示：

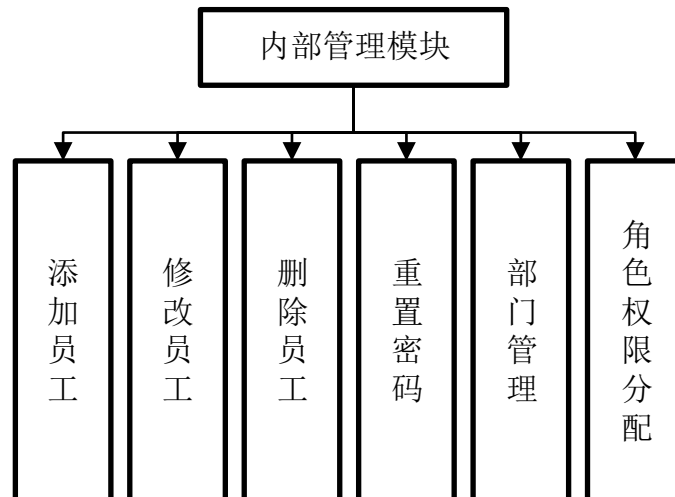


图 5-3 内部模块图

### 5.2.3 个人中心模块图

个人中心模块是用户管理个人信息以及会议信息的中心，主要由个人信息、会议信息、会议申请信息、会议发布和修改密码五个部分组成。个人信息支持修改功能。会议信息支持查看、删除会议与会议座次的选择。个人中心模块图如下图 5-4 所示：

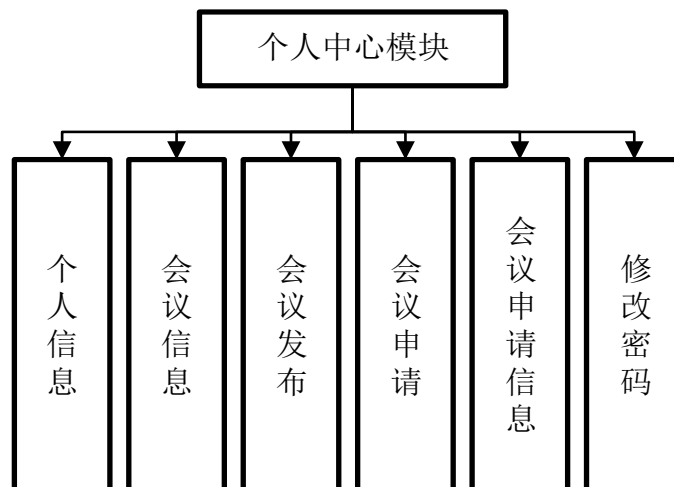


图 5-4 个人中心模块图

#### 5.2.4 新闻模块图

新闻模块是单独独立出来的一个模块，这个模块是针对所有用户（员工与经理）开放的，新闻模块不仅仅是发布新闻信息，对于会议的总结也可以发布，然后通过参会人员对于会议进行总结发布评论。新闻模块包括发布新闻、显示新闻条目、查看新闻详情三大功能。新闻模块功能如下图 5-5 所示：

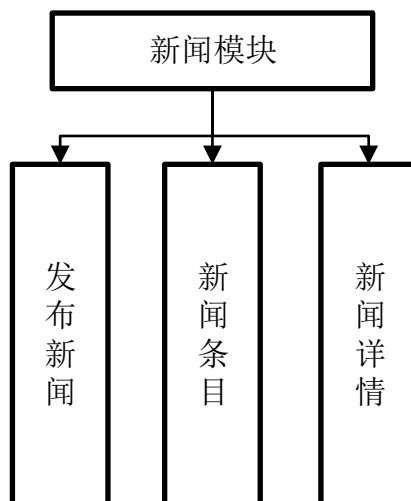


图 5-5 新闻模块图

### 5.3 功能的详细设计

详细设计阶段的根本目标是叙述如何具体地实现系统的功能，通过用户界面的展示详细的叙述了设计思路，说明完设计思路后，通过详细设计展现了如何设计程序，如何实现功能的，最后通过此功能的流程图来概括每一个功能的总体流程。

#### 5.3.1 登录与注册

##### （1）登录功能

所有用户均需要通过登录才能使用此系统，首先用户需要进入登录界面，输入用户名与密码，只有当用户名与密码相匹配时，方可进入此系统进行后续的操作。登录页面如下图 5-6 所示：



图 5-6 登录页面

设计思路：用户在登录页面输入登录 ID 与密码后，通过点击登录按钮将登录 ID 与密码发送到后台进行密码的比对，通过 Shiro 权限框架将用户登录信息封装，Shiro 从数据库获取对应用户的信息，然后通过设置其 MD5 加密方式将用户输入的密码进行加密，然后将其进行比对，只有当密码比对成功之后才能进入此会议管理系统。

详细设计：前台将用户输入的登录 ID 与密码发送到后台，后台封装成一个 UsernamePasswordToken 对象，对其进行认证，认证通过则进入系统，否则提示密码错误。登录功能部分后台核心代码如下：

```
//将用户输入的登录 ID 和密码封装成 UsernamePasswordToken 对象
```

```
UsernamePasswordToken token = new
```

```
    UsernamePasswordToken(loginId.toString(), password);
```

```
//得到主体
```

```
Subject subject = SecurityUtils.getSubject();
```

```
try {
```

```
    //通过 Realm 进行认证
```

```
    subject.login(token);
```

```
//记住认证信息
token.setRememberMe(true);
//从 Shiro 的 session 中获得用户的信息
User user = (User) subject.getPrincipal();
}

Realm 认证:
//1、把 AuthenticationToken 转换成 UsernamePasswordToken
UsernamePasswordToken upToken = (UsernamePasswordToken) token;
//2、从 token 中获取用户名 username
String loginId = upToken.getUsername();
//3、调用 Dao 层的方法，从数据库中获取 username 对应的记录
User userInfo = loginService.getUserInfo(Integer.parseInt(loginId));
//4、若用户不存在，则可以抛出异常
if (userInfo == null) {
    //抛出“用户不存在”的异常
    throw new UnknownAccountException("用户不存在");
}
//5、构建 AuthenticationInfo 对象并将其返回
//1)principal:认证的实体信息
Object principal = userInfo;
//2)credentials:密码
Object credentials = userInfo.getPassword();
//3)realmName:当前 realm 对象的 name
String realmName = getName();
//4)盐值加密，加密方式为 MD5
ByteSource credentialsSalt = ByteSource.Util.bytes(loginId);
SimpleAuthenticationInfo info = new SimpleAuthenticationInfo(principal,
    credentials, credentialsSalt, realmName);
```

登录流程如下图 5-7 所示：

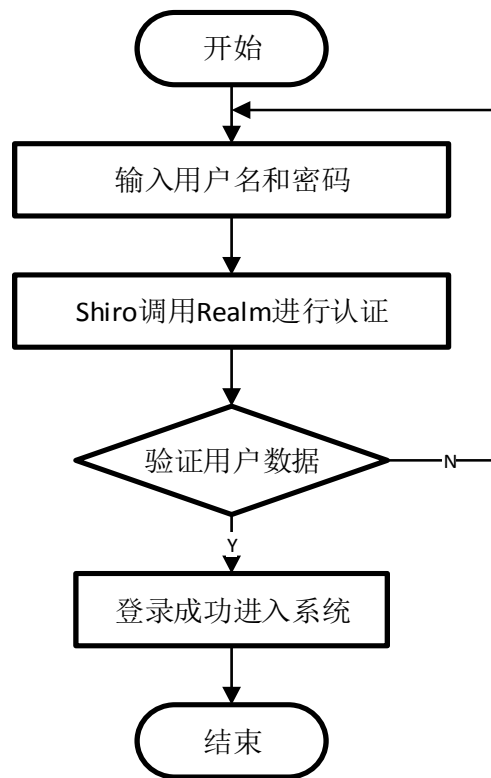


图 5-7 登录流程

## 7 总结

论文主要介绍了项目的开发背景、整个设计的流程与功能点的详细设计。项目总体采用 B/S 结构的模式，使用 SSM 作为框架搭建项目的架构，并使用 Maven 作为构建项目的工具和 Git 作为项目的版本管理工具。

在整个设计过程中，遇到的难点也不少，首先一开始就被困住，在会议发送阶段遇到了一个难点——如何实现会议信息的实时发送与接收。最初的想法是通过 Ajax 轮询来实现，Ajax 轮询的思路是通过每隔一段时间发送一个 HTTP 请求来判断是否有新的会议信息发布，然后无限循环的发送请求。这样的做法会引发一个非常大的问题，由于频繁的发送 Ajax 请求，如果当用户较多的时候，会发送大量的 Ajax 请求，这可能会造成 I/O 的阻塞或更严重的是导致系统的崩溃，因此这种方法不太适用。终于又寻找到另一种解决方法，那就是使用 WebSocket。WebSocket 协议其实是 HTTP 协议的升级版本，它实现了全双工通信，即服务器可以主动给浏览器发送消息。WebSocket 是一个长连接，虽然也会降低系统的性能，但相对于 Ajax 轮询要好很多。因此通过 WebSocket 便能轻松实现会议的实时发送与接收。虽然所遇到的问题层出不穷，而且每次问题一出现都会引发一大堆的问题，但最终也都解决了每一个难题。

毕业设计的整个过程，首先需要明确此项目的需求分析，再根据需求分析与设计的数据字典决定项目开发采用的浏览器模式与框架的选择，然后根据模块划分功能，最后进行项目的开发。

这次毕业设计将给大学生涯画上一个句号，在这个过程中，它加强了我的专业知识。该系统中界面的设计、逻辑的处理、数据库查询的效率还存在诸多不足，后期也会不断对界面进行美化，对逻辑处理和数据库查询进行不断的优化，并根据需要增加更多的功能点。

## 参考文献

- [1] 范晓玲. 基于 J2EE 的会议室管理系统的设计与实现[J]. 电脑与电信, 2016(10): 22-29
- [2] 万洁. 会议管理的信息化系统建设[J]. 中国电信业, 2013(6): 64-65
- [3] 陈建. 智能会议系统在会议管理中的应用[J]. 电子世界, 2017(18): 197-198
- [4] 姜刚, 杜宏伟. 一种基于透明加密文件技术的电子会议管理系统研究与实现[J]. 电子测试, 2017(13): 83-84
- [5] 钱俊凤, 付鋈, 陆岫昶. 无纸化会议系统建设及应用实例[J]. 贵州电力技术, 2017, 20(3): 61-63
- [6] Kenneth C, Laudon Jane. Management Information System [J]. California: Higher Education Press, 2009(6): 50-52
- [7] Zhiwen Yu, Yuichi Nakamura. Smart meeting systems[J]. ACM Computing Surveys(CSUR), 2010(13): 10-20
- [8] Teresa Ostrowska. Management Information in Administration Systems[J]. Foundations of Management, 2009(8): 95-110
- [9] DesHarnais S, Marshall B, Dulski J. Information management in the age of managed competition[J]. The Joint Commission journal on quality improvement, 1994(9): 36-38
- [10] 高辉. 如何提高会议管理水平[J]. 中国商贸, 2009(7): 116-117
- [11] 谭力, 杨宗源, 谢瑾奎. Ajax 技术的数据响应优化[J]. 计算机工程, 2010(7): 77-79
- [12] 肖在昌, 杨文晖. 基于 WebSocket 的实时技术[J]. 电脑与电信, 2012(8): 54-57
- [13] 笛斯特曼, 王浩. Java 核心技术[M]. 北京: 机械工业出版社, 2008
- [14] 姜晓铭, 陈武. JSP 程序设计与实例分析教程[M]. 北京: 清华大学出版社, 2001
- [15] 孙卫琴. Tomcat 与 Java Web 开发技术详解[M]. 北京: 电子工业出版社, 2009
- [16] 任泰明. 基于 B/S 结构的软件开发技术[M]. 西安: 西安电子科技大学出版社, 2006
- [17] 张宇, 王映辉, 张翔南. 基于 Spring 的 MVC 框架设计与实现[J]. 计算机工程, 2010(4): 88-90
- [18] 徐雯, 高建华. 基于 Spring MVC 及 MyBatis 的 Web 应用框架研究[J]. 微型电脑应用, 2012(7): 65-68

- [19] Yue Hua Ding, Chang Hua Liu, Yu Xin Tang. MVC Pattern Based on JAVA[J]. Applied Mechanics and Materials, 2012(12): 537-541
- [20] Jing Ying Zhao, Hai Guo. Development of the JSP Basic Knowledge Learning Software[J]. Advanced Materials Research, 2014(5): 481-485
- [21] Sabah Currim, Sudha Ram, Alexandra Durcikova. Using a knowledge learning framework to predict errors in database design[J]. Information Systems, 2014(5): 11-31
- [22] Mauricio J.V. Silva, C.Robert Carlson. MOODD, a method for object-oriented database design[J]. Data & Knowledge Engineering, 1995(16): 159-181
- [23] Dragos-Paul Pop, Adam Altar. Designing an MVC Model for Rapid Web Application Development[J]. Procedia Engineering, 2014(18): 1172-1179
- [24] Douglass, Bruce Powel. UML Class Diagrams [J]. F.nibedded Sysiems Programming, 2009(2): 30-40



## 致谢

经过几个月的时间终于将毕业设计与论文一并完成，在这期间遇到了许多的困难，但是都在老师与同学们的帮助下解决了，非常感谢我的毕业设计指导老师——游磊老师。在毕业设计期间，他不厌其烦的对我们的论文进行修改，对我们的设计进行指导，让我在这几个月中养成了能坚持做好一件事的耐心。

其次，对我的大学生涯我还得感谢每一位出现在我面前的老师，感谢您们给了我很多的教导，不仅给我专业知识有所指导，也对我们的未来规划和要走的方向给予了指导。感谢您们对我的悉心教导和倾心的培养，让我们避开了很多的弯路，这会让我在未来的道路走得更快更远，奠定了坚实的基石。

最后，衷心感谢在百忙之中参加答辩及评阅论文的老师，所有论文的不足之处，恳请各位老师给予批评和指正。

## 附录

登录:

```
@RequestMapping(value="/login",method=RequestMethod.POST)
```

```
    public String login(Integer loginId, String password, Map<String,Object> map) {  
        UsernamePasswordToken token = new UsernamePasswordToken(loginId.toString(),  
password);  
  
        Subject subject = SecurityUtils.getSubject();  
        System.out.println("subject:" + subject);  
        try {  
            subject.login(token);  
            token.setRememberMe(true);  
            User user = (User) subject.getPrincipal();  
            int flagCount = loginService.isHasFlag(loginId);  
            map.put("flagCount", flagCount);  
            map.put("userName", user.getUsername());  
            map.put("user", user);  
            map.put("loginId", loginId);  
            int askMeetingCount = 0;  
            String meetingResult = "";  
            if(subject.hasRole("manager")) {  
                askMeetingCount = askMeetingService.isHasManagerFlag(loginId);  
                if(askMeetingCount > 0){  
                    askMeetingService.setManagerFlag(loginId,0);  
                }  
            }else{  
                meetingResult = askMeetingService.getAskMeetingResult(loginId);  
            }  
            map.put("meetingResult",meetingResult);  
        }  
    }  
}
```

```

        map.put("askMeetingCount", askMeetingCount);
        return "head";
    } catch (AuthenticationException e) {
        e.printStackTrace();
    }
    return "redirect:/index.jsp";
}

```

自定义 Realm 认证授权：

/\*授权 \*/

@Override

protected AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principal) {

    User user = (User) principal.getPrimaryPrincipal();

    List<String> permissionList = null;

    try {

        permissionList = loginService.getPermissionListByLoginId(user.getLoginid());

    } catch (Exception e) {

        e.printStackTrace();

    }

    if(permissionList != null) {

        SimpleAuthorizationInfo simpleAuthorizationInfo = new

SimpleAuthorizationInfo();

        simpleAuthorizationInfo.addStringPermissions(permissionList);

        return simpleAuthorizationInfo;

    }

    return null;

}

/\*认证\*/

@Override

protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken token)

throws AuthenticationException {

```
UsernamePasswordToken upToken = (UsernamePasswordToken) token;
String loginId = upToken.getUsername();
User userInfo = loginService.getUserInfo(Integer.parseInt(loginId));
if (userInfo == null) {
    throw new UnknownAccountException("用户不存在");
}
Object principal = userInfo;
Object credentials = userInfo.getPassword();
String realmName = getName();
ByteSource credentialsSalt = ByteSource.Util.bytes(loginId);
SimpleAuthenticationInfo info = null;
info = new SimpleAuthenticationInfo(principal, credentials, credentialsSalt,
realmName);
return info;
}
```