**成都信息工程大学**

学士学位论文

中小企业在线办公系统的设计与实现

****

**姓 名： 张航**

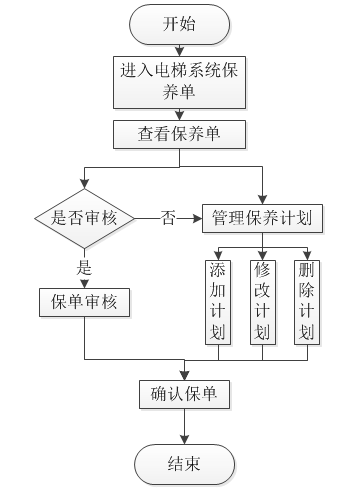
**学 号： 2014081139**

**系 别： 软件工程学院**

**专 业： 软件工程**

**指导老师： 唐聃**

**地址：四川省成都市西南航空港经济开发区学府路一段24号**

中小企业在线办公系统的设计与实现

摘要

随着社会的进步，不管是政府部门还是企业都有许多的文案，流程需要处理。在纷乱复杂的事物之间做出一个优秀的决定显得尤为重要。拥有一个专业化，智能化的办公系统，对政府部门或者企业来说，都可以大幅提高工作上的效率。随着科技的发展，在线办公系统使得不同地方或者不同部门之间的协同办公成为可能。如今的中国正朝着信息化建设的道路前进，而一套智能的在线办公系统将是企业或者政府部门实现信息化改革的第一步。

在线办公系统，顾名思义是指办公信息处理的智能化，它利用先进的技术，强悍的逻辑处理，帮助办公人员提高办公效率，简化一些复杂的办公流程，能办公人员更能专注于一些特别的任务，从而提高整个部门的生产率。

本次所开发的中小企业的在线办公系统，基于企业内部管理，应用软件工程技术和计算机设备实现中小企业办公的智能化以及企业内部管理的简单化。这次的系统采用面向对象设计的系统开发方式。整个系统包括身份验证，员工信息管理，消息管理，公告管理，部门管理，权限管理，系统管理，这七个模块和三类用户（普通员工，部门经理，人力资源部人员）。系统基于Node.js开发，使用容易维护的B/S框架，同时使用当下热门的非关联性数据库Mongodb作为数据的存储和处理，这些使得在线办公系统的公文发放，查看变得极为的方便与简单。

使用该系统能够极大的简化办公流程和时间，不需要发送一则公告在各部门传来传去，也不需要为了一则文件跑完所有的办公室。而这些统统都可以使用在线办公系统在网上得到解决；不仅如此，系统的运用还可以加强企业内部的管理，使得工作变得规范化，同时也可减省一大笔管理的开销费用；尤为重要的是该系统还可以加强上级与员工的交流，提高企业内部之间的凝聚力，更好的发挥员工的创造力和工作的积极性。

**关键词：**办公自动化；Node.js；Mongodb；管理信息;

Design and implementation of online office system for small and medium-sized enterprises

With the progress of society, both government departments and enterprises have a lot of copywriting, and the process needs to be dealt with. It is important to make a good decision between complex and complex things. With a specialized, intelligent office system, it can greatly improve the efficiency of the work for government departments or enterprises. With the development of technology, the online office system makes it possible to work together in different places or departments. Today's China is moving towards informationization, and a smart online office system will be the first step towards informatization reform in enterprises or government departments.

Online office system, just as its name implies is refers to the office of intelligent information processing, it USES advanced technology, strong logical processing, help office workers improve office efficiency, simplify the complex office process, can office staff can focus on some special task, so as to improve the productivity of the whole department.

The development of small and medium-sized enterprise's online office system, based on the enterprise internal management, application software engineering techniques and computer equipment to implement the intelligent of the small and medium-sized enterprise office as well as the enterprise internal management simple. This system adopts the system development method of object - oriented design. The whole system including authentication, employee information management, news management, bulletin management, department management, rights management, system management, the seven modules and three types of users (ordinary employees, department managers, hr personnel). System based on the Node. Js development, using B/S framework of easy to maintain, and use the current decoupling mongo database for data storage and processing, which makes online office system of official document, view becomes very convenient and simple.

The use of the system can greatly simplify office processes and time, without sending a notice to the departments, or running all the offices for a single file. All of these can be solved using the online office system; Moreover, the application of the system can also strengthen the internal management of the enterprise, so as to standardize the work and reduce the cost of a large amount of management. It is particularly important that the system can also strengthen the communication between superiors and employees, enhance the cohesion within the enterprise, and give better play to employees' creativity and enthusiasm for work.

**Key words:** Office Automation; Cooperative Office; Nodejs; Mongodb

**目录**

论文总页数：50页

[1 引言 1](#_Toc509944534)

[1.1 本论文组织结构简介 1](#_Toc509944535)

[1.2 课题背景 1](#_Toc509944536)

[1.3 企业面临的挑战 2](#_Toc509944537)

[1.4 在线办公系统的发展趋势 2](#_Toc509944538)

[1.5 在线办公系统开发的现实意义 3](#_Toc509944539)

[1.6 在线办公系统概述 3](#_Toc509944540)

[2 设计技术及开发环境 4](#_Toc509944541)

[2.1 设计技术 4](#_Toc509944542)

[2.1.1 Node.js技术简介和具有的优势 4](#_Toc509944543)

[2.1.2 Mongodb数据库简介和优势 6](#_Toc509944544)

[2.1.3 Angular技术简介和优势 7](#_Toc509944545)

[2.2 开发环境 8](#_Toc509944546)

[2.2.1 Node.Js 8](#_Toc509944547)

[2.2.2 Mongodb 8](#_Toc509944548)

[2.2.3 Angular 8](#_Toc509944549)

[2.2.4 Express 8](#_Toc509944550)

[2.2.5 Ejs 8](#_Toc509944551)

[2.2.6 集成开发工具WebStorm 8 9](#_Toc509944552)

[3 系统分析 9](#_Toc509944553)

[3.1 可行性分析 9](#_Toc509944554)

[3.1.1 技术可行性 9](#_Toc509944555)

[3.1.2 经济可行性 9](#_Toc509944556)

[3.2 需求分析 10](#_Toc509944557)

[3.2.1 用户注册、登入、身份验证功能 10](#_Toc509944558)

[3.2.2 个人信息管理 10](#_Toc509944559)

[3.2.3 员工信息管理 10](#_Toc509944560)

[3.2.4 部门管理 10](#_Toc509944561)

[3.2.5 公告发表 10](#_Toc509944562)

[3.2.6 公告管理 10](#_Toc509944563)

[3.2.7 发表留言 11](#_Toc509944564)

[3.2.8 云盘管理 11](#_Toc509944565)

[3.2.9 权限管理 11](#_Toc509944566)

[3.2.10 资源回收 11](#_Toc509944567)

[3.3 数据流分析 11](#_Toc509944568)

[3.4 系统用列分析 12](#_Toc509944569)

[3.4.1 普通员工角色用例 12](#_Toc509944570)

[3.4.2 部门经理角色用例 13](#_Toc509944571)

[3.4.3 管理层管理员角色用例 14](#_Toc509944572)

[3.5 系统功能结构图 15](#_Toc509944573)

[4 系统概要设计与实现 16](#_Toc509944574)

[4.1 功能模块 16](#_Toc509944575)

[4.1.1 用户身份验证模块 17](#_Toc509944576)

[4.1.2 个人信息管理模块 17](#_Toc509944577)

[4.1.3 员工信息管理模块 18](#_Toc509944578)

[4.1.4 公告管理模块 19](#_Toc509944579)

[4.1.5 消息管理模块 19](#_Toc509944580)

[4.1.6 部门管理模块 20](#_Toc509944581)

[4.1.7 云盘管理模块 20](#_Toc509944582)

[4.1.8 权限管理模块 21](#_Toc509944583)

[4.2 数据库设计 21](#_Toc509944584)

[4.2.1 概念结构设计 22](#_Toc509944585)

[4.2.2 关系模式设计 25](#_Toc509944586)

[5 系统编码的实现 27](#_Toc509944587)

[5.1 系统架构 27](#_Toc509944588)

[5.1.1 系统架构图 28](#_Toc509944589)

[5.1.2 程序结构图 28](#_Toc509944590)

[5.2 模块实现 29](#_Toc509944591)

[5.2.1 数据库的链接与实现 29](#_Toc509944592)

[5.2.2 用户登入模块 30](#_Toc509944593)

[5.2.3 主页面 33](#_Toc509944594)

[5.2.4 消息管理 34](#_Toc509944595)

[5.2.5 公告 35](#_Toc509944596)

[5.2.6 部门管理 39](#_Toc509944597)

[5.2.7 个人云盘管理 40](#_Toc509944598)

[6 系统测试 42](#_Toc509944599)

[6.1 测试设计与测试结果 43](#_Toc509944600)

[6.1.1 登录功能的测试 43](#_Toc509944601)

[6.1.2 注册模块测试 43](#_Toc509944602)

[6.1.3 公告管理功能模块测试 44](#_Toc509944603)

[6.1.4 公告查看测试内容 44](#_Toc509944604)

[6.1.5 留言模块测试内容 44](#_Toc509944605)

[6.1.6 个人信息管理模块测试内容 44](#_Toc509944606)

[6.1.7 部门人员查看模块测试内容 45](#_Toc509944607)

[6.1.8 部门人员管理模块测试内容 45](#_Toc509944608)

[6.1.9 发表公告管理模块测试内容 45](#_Toc509944609)

[6.1.10 个人公告管理模块测试内容 45](#_Toc509944610)

[6.1.11 存档管理模块测试内容 46](#_Toc509944611)

[6.1.12 标签管理模块测试内容 46](#_Toc509944612)

[6.1.13 上传文件模块测试内容 46](#_Toc509944613)

[6.1.14 下载文件模块测试内容 46](#_Toc509944614)

[6.1.15 消息查看模块测试内容 47](#_Toc509944615)

[6.1.16 消息发表模块测试内容 47](#_Toc509944616)

[6.1.17 消息状态模块测试内容 47](#_Toc509944617)

[6.1.18 阅读统计模块测试内容 47](#_Toc509944618)

[6.1.19 头像管理模块测试内容 48](#_Toc509944619)

[6.1.20 员工审核模块测试内容 48](#_Toc509944620)

# 引言

## 本论文组织结构简介

本论文主要分为六个章节，下面是对各个章节内容的简要介绍：

1. 引言部分。主要介绍本系统的开发背景与现状。
2. 设计技术与开发环境。主要对本系统的用到的工具和环境进行介绍。
3. 系统分析。主要对本系统的可行性和数据流进行分析
4. 系统概要设计。主要介绍了系统的功能和数据库进行介绍
5. 系统编码的实现。主要介绍了系统的各个模块的功能内容。
6. 系统测试。主要对系统的各个功能模块进行测试做介绍

## 课题背景

本随着科技的不断前进，计算机的发展越来越快，促进了整个人类社会的进步和繁荣，软件作为各种信息流动的载体和信息处理的一般工具，在如今的数据大爆炸的时代扮演了不可或缺的角色。在各个政府部门、企业的工作室中，是采用Internet/Intranet技术，利用数据流的知识，开始以软件为中心，把一系列办公设备和先进的通信机制链接在一起，广泛、全面、快速地整理、加工、存储和使用有效信息，使企业内部人员快速的分享有用信息，高效地协同工作；改变以前复杂、低效、拖沓的办公方式，从而达到提高生产效率的目的，为科学管理和决策服务提供最大的帮助。

一个优秀的企业已经把是否全面实现公司办公自动化作为衡量自己企业是否达到现代化企业的标准。企业对信息需求的增长，使通信与网络技术已经渗透到企业的日常工作中。传统的企业内信息的交流方式复杂，早已不能满足企业对大量信息的快速交流与处理。在线办公系统的自动化管理的应用满足了企业需求，使得企业的办公网络化、自动化，提高了企业内部的管理水平和办公的效率化，进而全面提升了企业在市场竟争中的综合竞争力。在信息技术飞速发展的今天，在线办公系统的技术的内涵和扩展都发生了根本的改变。各种技术的引入使的企业办公人员能够处理更加广泛的业务和信息，比如工作日程安排、公告发放、会议安排、公文发布等。伴随服务器端的优化，分布式处理加强和数据库技术的发展，在线办公系统使的企业现代化的无纸办公成为现实。在线办公系统逐步转向以信息流为中心，实现了对员工、公告、消息和会议的一系列管理。在线办公系统的基本功能是使企业内部信息的发布与传递、工作流管理和公告管理变得极为方便快捷。

当今社会已经进入信息大爆炸时代，信息技术的改革使社会的各个领域，各个行业都发生了翻天覆地的变化，每个企业都必须紧跟信息时代化的步伐，加强企业的信息处理能力，不断提高竞争力和提升现代化企业的管理能力，以适应整个社会的不断发展。

## 企业面临的挑战

面对信息化大爆炸时代的到来和日益激烈的市场竞争环境，企业也迎来了新的机遇和挑战。不断加强企业的信息基础建设，提高企业管理效率和生产率是企业获得优势竞争力的最佳方法。在线办公系统作为提高企业办公管理效率的平台，近年来，受到各中小企业高度重视，他们纷纷构建起适合于自身企业的应用特色的在线办公系统，进而逐步实现企业的流程自动化，从而提高工作效率并提升管理质量。企业通过构建高效实用的企业在线办公系统，使企业内外实现高效的信息沟通联络、网络协同办公的无纸化；帮助企业最终实现规范管理，信息资源高效传递；使企业从彼此独立的被动的复杂管理模式转向一体化，信息资源共享的统一管理模式，并最终提高企业在社会上的强大竞争力。

## 在线办公系统的发展趋势

在线办公系统自20世纪70年代中期以来，发展至今已经经历了三个阶段：以数据处理为中心的文件型在线办公系统；以工作流为中心的流程性在线办公系统；以知识管理为核心的知识型在线办公系统。传统在线办公系统无论是独立的应用软件还是集成的解决方案，大多是独立性的应用，或者是业务功能模块的简单叠加。它们具有的信息封闭问题不仅体现在系统本身，还体现在与其他系统的不兼容。信息封闭的存在给企业的运营带来了很大的负面影响，使得许多系统的数据只能在本系统内使用，而不能与其他系统实现数据交换、文件流转和信息共享，造成企业内部各业务系统相互独立、数据不一致、信息共享程度不高、管理分散、维护工作量大的后果。

导致这些问题的出现，实质在于缺少一个统一、综合、开放、规范的应用平台。开发自主的中小企业在线办公系统网络平台软件，以统一平台，协同应用为特征，是在线办公自动化系统的综合解决方案。这个平台必须具有充分的开放性和灵活性，允许用户方便的自定义各种业务流程和事物，和其他系统进行数据整合，共同处理信息等一系例问题。传统的Java技术开发的在线办公系统因为不具有良好的维护性而正在逐渐被淘汰，目前以Node.js开发的在线办公系统正在逐渐进入人们的视野，在这种在线办公系统平台上，在管理中做事务，而不是在事物中做管理，信息交流、业务、部门管理都是协同一致的。

## 在线办公系统开发的现实意义

在线办公系统以计算机为中心，采用一系列的办公设备和先进的信息交流技术，广泛、全面、迅速地整理、加工、存储和使用信息资源，使企业各个部门之间的内部人员方便快捷地互相共享信息，高效地协同工作；在线办公系统不但使个人办公效率迅速提高，更重要的是可以实现企业各个部门之间的协同工作。协同工作意味着要进行信息资源的交流互换，工作的协调与合作。由于网络的逐渐发展的，这种交流与沟通几乎可以在瞬间完成，并且不必担心对方是否在电话旁边或是否身边有传真机可用，这对于以前的有纸化办公大大的提高了生产率而且节约了很多资源。

在线办公系统还可以根据不同企业自身的特色，实现企业定制的在线办公系统，可以定制出属于企业自身发展最适合的的一套系统。操作人员不需要懂计算机和软件开发的任何知识，就可以轻松操作该系统，并达到预期的效果，大大的方便企业领导的决策和人员管理。

改变以往复杂、低效的无沟通的手工办公方式，转入为科学管理和决策服务的在线办公，从而达到提高行政效率的目的。对于企业来讲高效的工作效率就是企业的最大竞争力。

## 在线办公系统概述

办公自动化(OA－Office Automation)是将计算机网路功能和现代化办公结合起来的一种新型的办公方式，是当前信息技术革命中一个具有很强生命力的技术应用领域，是信息化社会的产物。通过计算机，企业内部的人员可不同时间，不同地点协同工作。通过在线办公系统所实施的交换式网络应用，使信息的传递快速、高效、方便，从而极大地扩展了办公手段，实现了办公和管理高效率。

同时随着科学技术的发展和企业使用人员的办公方式和习惯、管理思想的变化，对于在线办公明确的定义也不断的变化着，在技术发展过程中的每一个时代，也赋予给在线办公不同的内容和新的内涵，同时不同行业、不同层次的人员对在线办公的看法和理解也各有不同。

在线办公系统就功能方面而言，就是一个企业各部门汇聚一切信息资源，并在一起共同处理与管理的集合。它面向企业不同类型的使用者，便有不同的功能；

对于企业高层领导而言，在线办公系统是领导查看企业内部不同部门运行状况的工具。同时在线办公运用科学的逻辑模型，结合企业内部的信息为条件，为企业领导提供决策上的参考和依据；

对于各个部门的管理者而言：在线办公系统给管理者提供一个信息发放的平台，同时也搭建起管理者与员工沟通的桥梁，使得管理者与员工的信息交流更加的及时，高效。从而提高整个部门的办事效率；

对于普通职员而言：在线办公系统能搞逻辑的处理好办事的流程，同时能够方便的查看部门的通告，使之准确、高效，愉快地工作。

在以前传统的认知和了解中，认为在线办公系统仅仅不过是诸如公告流转、收发文管理、档案管理、会议安排、文献检索等等这些非结构化数据的处理过程，面向的用户群也只是机关办公室或企业的职能部门、文秘部门、其实，在今天看来，在线办公系统拥有更丰富的内容和层面，更广泛的用户群。

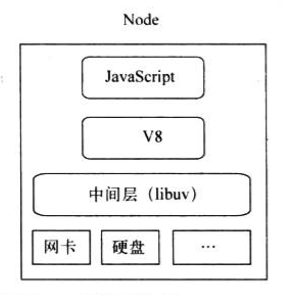
# 设计技术及开发环境

## 设计技术

### Node.js技术简介和具有的优势

1. Node.js简介

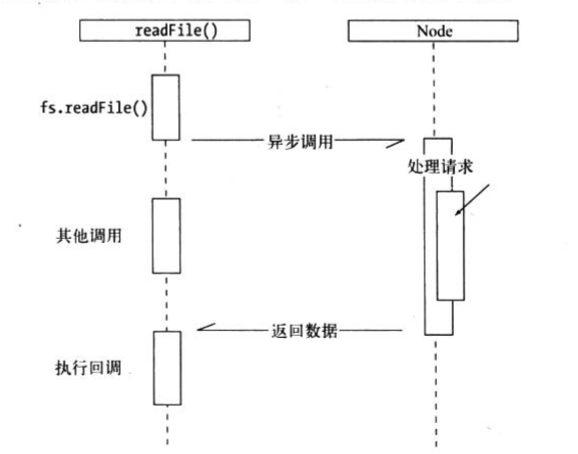
Node.js是一项服务器技术。我们都知道客户端提出服务请求，而服务器端负责处理请求并提供服务。而对于互联网来说，在Node.js之前JavaScript是一项完全的客户端技术，被用于浏览器中实现各种动画，对DOM的操作等等。而后端，即服务端则是由PHP、Python、Ruby、Java等等语言来实现。Node.js的出现，使得前后端使用同一种语言，统一模型的梦想得以实现。Node.js拥有比JavaScript的更好的组织代码能力，大幅提升复用性。同时在ES6中这一点也得到了很大的提升。处理文件与数据库。与互联网进行沟通，以标准化的格式处理请求并发送回答。同时，Node.js还带来了许多别的后端技术所不具备，或是不完善的优点，如其他人回答中的事件驱动，异步编程，非阻塞式io等等。JavaScript本身语言的特性，以及其的流行程度与社区活跃度给Node.js带来了各种意义上的优势。



图片 1 Node的组件构成

1. Node.js良好的异步I/O

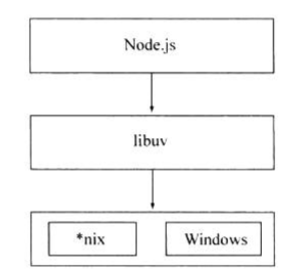
在Node中，绝大多数的操作都以异步的方式进行，在Node中，我们可以从语言层面很自然的进行并行I/O操作。每个调用之间无须等待之间的I/O调用结束。在编程模块上可以极大的提升效率。



图片 2 Node经典的异步操作

1. Node.js跨平台好

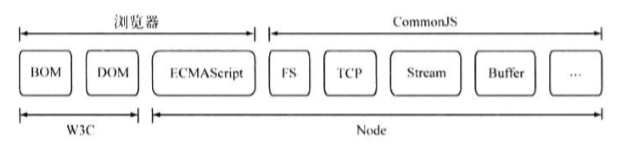
起初，Node只能在Linux平台上运行，后来随着Node的发展，微软注意到了它的存在，并投入一个团队帮助Node实现window平台的兼容。现在Node能够直接在window平台上运行。得益与Node在架构层面的改动，它在操作系统与Node上层模块构建了libuv架构，使得Node在不同平台上的兼容。



图片 3 Node基于libuv实现跨平台

1. Node.js对CommonJs规范成熟的表现

CommonJs规范的提出，主要是为了弥补JavaScript当前没有标准的缺陷，已到达像Java等开发大型应用的能力而不是停留在小脚本程序的阶段。Node能以一种比较成熟的姿态出现，离不开CommonJs规范的影响。在服务器端，CommonJs能以一种寻常的姿态写进各个公司的代码中，也离不开Node的优异表现。



图片 4 Node与CommonJs,ECMAScript之间的关系

### Mongodb数据库简介和优势

1. Mongodb的简介

MongoDB 是一个基于分布式文件存储的数据库。由C++语言编写。旨在为Web应用提供可扩展的高性能数据存储解决方案。

MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。他支持的数据结构非常松散，是类似Json的Bson格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是他支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。

1. Mongodb的优秀的文档存储

数据存储以BSON/JSON文档，这对于Web应用程序有很大的意义。开发者API喜欢以JSON形式传输，这使得整个项目的数据表示可采用统一的模型。所有这一切都无需任何前期架构设计。

1. Mongodb的可扩展性

MongoDB被用在一些规模庞大的环境中，FourSquare/Craiglist都在使用它。通过分片数据缩放处理理论上可实现更高的吞吐量。

1. Mongodb的数据库易于设计

MongoDB以文档的形式存储数据，不支持事务和表连接。因此查询的编写、理解和优化都容易得多。简单查询设计思路不同于SQL模式，嵌入文档在特定的环境下可得到更好的查询，然而这需要先加入集合。如果需要执行多个请求到数据库则需要加入其到客户端。在MongoDB时ODM工具（如Doctrine2）将发挥自身的优势。

1. Mongodb的安全性高

由于MongoDB客户端生成的查询为BSON对象，而不是可以被解析的字符串，所以可降低受到SQL注入的攻击的危险。最常见的攻击类型为针对Web应用程序的攻击，在MongoDB上使用Doctrine2 ODMs 查询语言可减轻攻击风险。

### Angular技术简介和优势

1. Angular的简介

AngularJS诞生于2009年，由Misko Hevery 等人创建，后为Google所收购。是一款优秀的前端JS框架，已经被用于Google的多款产品当中。AngularJS有着诸多特性，最为核心的是：MVC、模块化、自动化双向数据绑定、语义化标签、依赖注入等等。

AngularJS 是一个 JavaScript框架。它是一个以 JavaScript 编写的库。它可通过 <script> 标签添加到HTML 页面。

AngularJS通过指令扩展了 HTML，且通过表达式绑定数据到 HTML。

AngularJ是以一个JavaScript 文件形式发布的，可通过 script 标签添加到网页中。

1. Angular的优势

Angular模板功能强大丰富，并且是声明式的，自带了丰富的Angular指令；

Angular是一个比较完善的前端MV\*框架，包含模板，数据双向绑定，路由，模块化，服务，过滤器，依赖注入等所有功能；

Angular自定义Directive，比jQuery插件还灵活，但是需要深入了解Directive的一些特性，简单的封装容易，复杂一点官方没有提供详细的介绍文档，我们可以通过阅读源代码来找到某些我们需要的东西，如：在directive使用 $parse；

Angular的ng模块化比较大胆的引入了Java的一些东西（依赖注入），能够很容易的写出可复用的代码，对于开发的团队来说非常有帮助，即使项目从上线到目前，UI变化很大，在摸索中迭代产品，但是js的代码基本上很少改动。

## 开发环境

### Node.Js

从Node.js网站（http://nodejs.org）下载最新的node-v8.9.4-x64版本.

安装过后配置Node.js环境变量：Path：D:\nodejs\；

Mongodb

从Mongodb官网（http://docs.mongodb.org/manual/installation/）下载最新的mongodb-win32-x86\_64-2008plus-ssl-3.6.1-signed版本.

安装完后配置Mongodb的环境变量：Path: D:\mongodb\bin;

创建数据文件目录<mongo\_data\_location>/data/db;

在控制台提示符下使用如下命令启动：

Mongod -dbpath<mongo\_data\_location>/data/db;

Angular

Angular是一个Javascript客户端库，为了在Node.js环境中实现，我们需要在HTML模板中使用<script>标记，我们可以从官网（http:angularjs.org）下载下来并引用，同时也可使用在线的第三方。

Express

Express 是一个简洁而灵活的 Node.js Web应用框架, 提供一系列强大特性帮助你创建各种Web应用。Express 不对Node.js 已有的特性进行二次抽象，我们只是在它之上扩展了Web应用所需的功能。丰富的HTTP工具以及来自Connect框架的中间件随取随用，创建强健、友好的API变得快速又简单.

同时Express 支持各种模板引擎，Express 命令行可以添加对 Jade，EJS，JSHTML 和 Hogan 的支持。根据 Express 文档，任何模版引擎只要符合特定的签名就可以生效。推荐从 Consolidate.js 库（模版引擎集合）里面查找喜欢的模版引擎。

在安装好Node.js后，我们在package.json 中手动添加 Express 依赖，在相同的目录运行：npm install 就可搭建好Express环境，或者我们也可以直接使用命令行安装：npm install -g express；

安装完后可以用 Express 在命令行生成应用。

Ejs

Ejs是一个JavaScript模板库，用来从Json数据中生成Html字符串。与最初的JavaScript相比较，一些不太了解你的代码的人可以更容易地通过EJS模板代码看得懂你的代码。而且可以让代码更加干净整洁，让人易懂。使用Ejs可以来找回你的明确、维护性良好的Html代码结构。

集成开发工具WebStorm 8

WebStorm是一款强大的HTML5/JavaScript Web前端开发工具，被广大JS开发者誉为“Web前端开发神器”。WebStorm 8全新特性中包括对AngularJS的支持，能够高效准确地智能感知Angular语法、指令。WebStorm是一款强大的HTML5/JavaScript Web前端开发工具，被广大JS开发者誉为“Web前端开发神器”。

同时WebStorm 8全新特性中包括对AngularJS的支持，能够高效准确地智能感知Angular语法、指令。WebStorm还完美支持Spy-js，合并了这款JavaScript调试利器，大大提高了开发者们的工作效率。

本系统中我们选用WebStorm 8作为我们的扩展开发工具。

# 系统分析

## 可行性分析

技术可行性

基于中小型企业的在线办公系统的开发采用了目前比较流行、先进成熟、并且符合实际应用的开发方法。且广泛应用于网站开发的网页开发工具Dream weaver，Webstorm，JavaScript模板库Ejs。还采用良好的后台处理Node.js和MVC架构技术，功能强大的非关联行数据库Mongodb，在结合自己的实际开发能力，可以开发出较为完善的办公自动化系统。

经济可行性

中小型企业的在线办公系统通过对企业内部信息进行全面的管理，大大提高了企业各部门的办公效率。通过系统对企业各部门运行过程中的数据进行全面的管理，避免人为处理各类数据时所产生的各种问题，提高了企业的办事效率，为企业提供了大量的、可靠的数据，使企业的管理更加的系统化、科学化、流程化。

开发中小型企业的在线办公系统在硬件方面主要是计算机，且开发系统所需资料书籍可以到图书馆翻阅或者个人购买，我们的硬件情况可以满足开发任务，因此整个系统的所需投资符合我们的目前预见的支付状况，经济可行性好。

本系统包括PC端浏览器端、Web服务器和服务端数据库。他们之间的物理架构如图3.1所示。

## 需求分析

中小型企业的在线办公系统主要实现的功能有：

用户注册、登入、身份验证功能

中小型企业的在线办公系统的出发点就是注册，登入功能，只有实现了这个功能才能依次使用其它的功能。新加入公司的员工可在系统注册页面进行注册功能，每个注册的用户的账户名都是唯一的，在进行注册时，系统会对用户注册的账户名进行验证以保证其唯一性，用户注册成功后便可在登入页面登入系统。

中小型企业的在线办公系统根据不同身份的用户进入系统都会呈现不同的功能，所以在进入系统之前必须要经过身份验证，本系统主要分三类身份的用户登录使用：管理层、部门经理、部门员工。每类用户都有不同的身份，在登入页面，用户登入后，系统通过后台查询数据库反馈相应的身份信息，通过反馈的身份信息给不同的用户呈现不同的页面。

个人信息管理

用户通过身份验证登录系统之后，可以在个人信息管理页面进行自己信息的查看和更改。用户可以对自己的姓名，年龄，邮箱，以及籍贯进行更改。而账户名，所在部门，入职时间，管理权限则只可查看，无法变更。

员工信息管理

用户通过身份验证登录系统之后，如果身份是部门经理，则可以在部门员工管理页面查看部门员工的个人信息，还可以修改、删除本部门员工的信息；如果身份是管理层，则可以在企业员工管理页面查看整个企业员工的个人信息，还可以修改、删除不同部门的员工的信息。

部门管理

用户通过身份验证登录系统之后，如果身份是普通职员则不会显示部门管理功能；如果身份是管理层则可以在部门管理页面进行增加部门、删除部门、修改部门功能。在新增部门时，每个部门都会有唯一的id确保每个部门的独立性，使其每个部门在企业尤其只有一个。

公告发表

根据登录用户的身份，不同的用户可以发送不同的公告。如果身份是普通员工，则可以发表在本部门的普通公告；如果身份是部门经理，则可以在部门内发表带有标签的公告，方便部门员工及时查看和注意；如果身份是管理层，则可以发表在本部门的带有标签的公告，同时还可以决定是否发表的公告在系统的主页展示。

公告管理

用户通过身份验证登录系统之后，可以在我的公告页面查看自己所发表的所有公告。点击每篇公告，则进入该公告的详细页面，可以对该公告进行删除和修改。点击部门公告，可以查看部门里的所有公告，同时可以搜索部门内的公告进行查看。如果身份是部门经理，还可以对部门的公告进行统一的管理，不但可以查看部门内所有公告，同时也可以删除部门内的公告。如果身份是管理层，则可以在公告管理页面查看所有部门的公告，也可以搜索、删除企业不同部门的公告，同时可以通过特有标签查看不同部门重要的公告，方便管理层及时知道每个部门的重要信息。

发表留言

用户通过身份验证登录系统之后，进入所查看公告的详细页面后，可以在公告底部对该公告进行留言或评论。这功能可以使得部门经理及时知道部门重要公告，已被哪些员工所知道，方便信息在部门的传达。

云盘管理

用户通过身份验证登录系统之后，每个用户都会被分配云盘，用户可以在自己的云盘页面上传重要的文件或资料，同时也可以进行下载，方便用户及时的处理企业重要的文档，方便快速的办公。同时在我的文件页面，可以对上传的文件进行管理，可以删除自己不像保存的文件。每个用户的云盘是独立的，其余用户没有权限进行查看。

权限管理

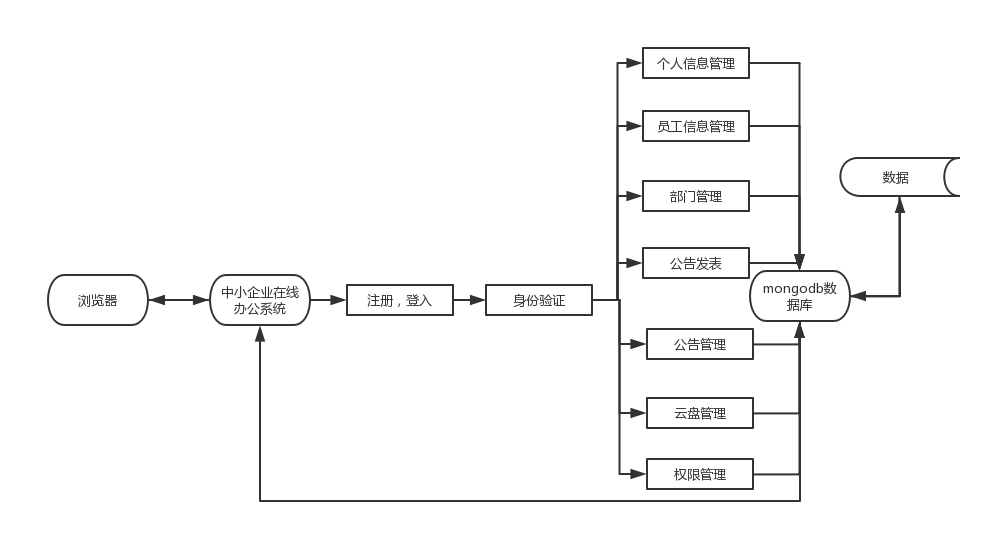
权限管理在本系统扮演着重要的角色，权限的设定关系到用户登录系统会呈现怎样的界面，权限管理主要是对职员角色的设定来实现。只有管理层才有此权限管理此功能。实现对用户的审核和修改用户的权限。

资源回收

用户通过身份验证登录系统之后，每个用户都会被分配云盘，这会导致后台的文件越来越多，同时用户在发表公告时，所附带的附件资料，也会所随的公告的增加而变多。同时企业员工之间的流动性很大，员工的去留就会导致后台系统的文件的堆积和冗余，所以一个良好的资源管理会起到一个稳定后台的作用。如果身份是管理层，则可以在员工辞退后，删除此员工。这时资源回收功能就会删除该用户的云盘里所有的文件以及该部门内该用户的文件分区，同时用户所发表的公告，以及公告里面附带的文件也会在公告的文件分区被删除，大大的节省系统所占的内存大小。

## 数据流分析

系统数据通过客户端浏览器到业务系统的各个接口，根据登录角色的身份不同，不同数据流向的目标接口点也不同，数据从浏览器开始，经过系统的不同逻辑处理，最终流向目标数据库,系统数据流向图，见下图所示：系统中的事件发生过程如下：



图片 5 系统数据流图

## 系统用列分析

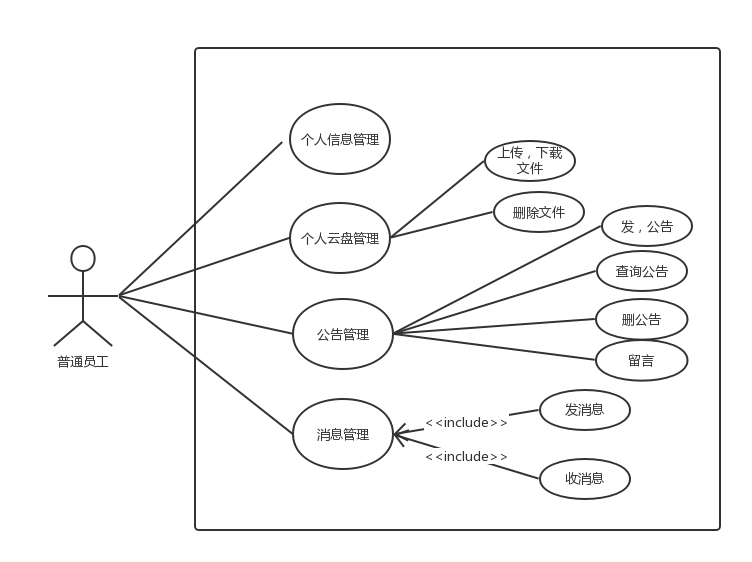
用例是一个UML中非常重要的概念，在使用UML的整个软件开发过程中，用例处于一个中心地位。用例是对一组动作序列的抽象描述，系统执行这些动作序列，产生相应的结果。这些结果要么反馈给参与者，要么作为其他用例的参数，用例被认为是第二代面向对象技术的标志普通职员角色用例。

参与者和用例从用户的角度来看，他们并不想了解系统的内部结构和设计，他们所关心的是系统所能提供的服务，也就是被开发出来的系统将是如何被使用的。而用例的内容用例图使我们对系统的功能有了一个整体的认知，我们可以知道有哪些参与者会与系统发生交互，每一个参与者需要系统为它提供什么样的服务。用例包括参与者和用例，参与者是系统用户的一个标签，也就是系统的一个角色，参与者可以是与系统、子系统或类发生交互外部用户、进程或其他系统也可以是人、另一个计算机系统或一些可运行进程。在 UML 中，用一个简单的椭圆来表示用例，而参与者则用一个带有名称的小人表示，参与者与用例的关系用箭图线表示，同时用例与用例也有两种关系包括和扩展也可以没有关系，参与者与参与者有泛化关系也可以没有关系。

根据系统要求，可知该系统有三类角色也就是三种参与者：一是普通职员，二是部门经理，三是管理层；不同角色有不同的权限，根据角色的权限可以得到该系统的用例图如下：

普通员工角色用例

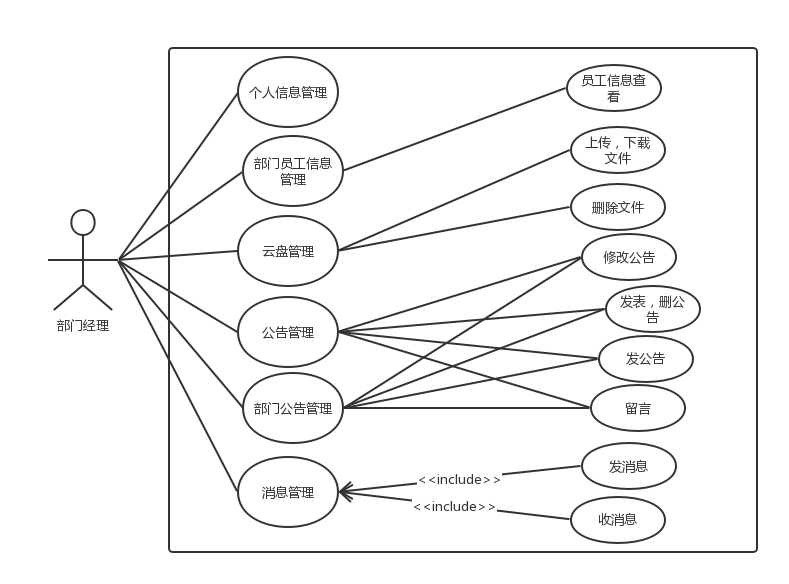
职员角色用例主要有：个人信息管理、公告管理、云盘管理、信息管理。个人信息管理主要是本人的一些基本信息的查看和修改；公告管理，就是发送公告，搜索公告，阅读公告，修改公告，删除公告等一系列功能；云盘管理就是上传、下载、删除自己的文档资料；信息管理就是发送信息，查看信息，发送信息可以根据其他职员的账号名发送给特定职员，目前短消息不支持群发；普通员工的权限在中小企业在线办公系统里是最小的，员工角色用例如下图所示：



图片 6 普通员工用例图

部门经理角色用例

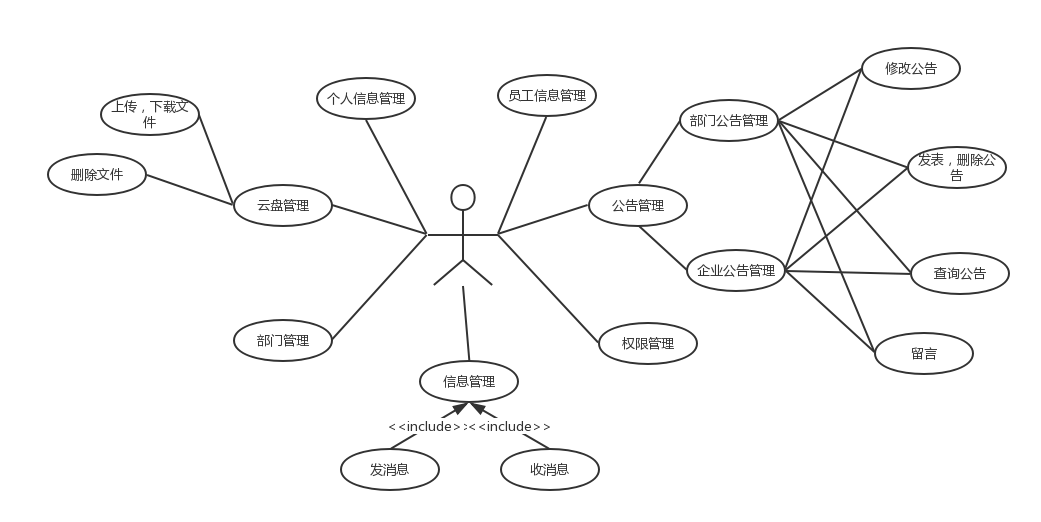
部门经理角色用例主要有：个人信息管理、公告管理、云盘管理、信息管理，员工信息管理，部门公告管理；部门经理角色继承与职员角色，有职员的所有功能，同时可以对本部门员工进行管理，如查看可以删除员工信息，修改员工信息；同时在公告上，部门经理不但可以查看部门内所有的公告，还可以删除部门内的所有公告部门经理对部门内的权限可以进行修改，但对部门和部门外的权限信息的修改删除只有管理层角色有此权限，部门经理角色用例图如下图所示：



图片 7 部门经理用例图

管理层管理员角色用例

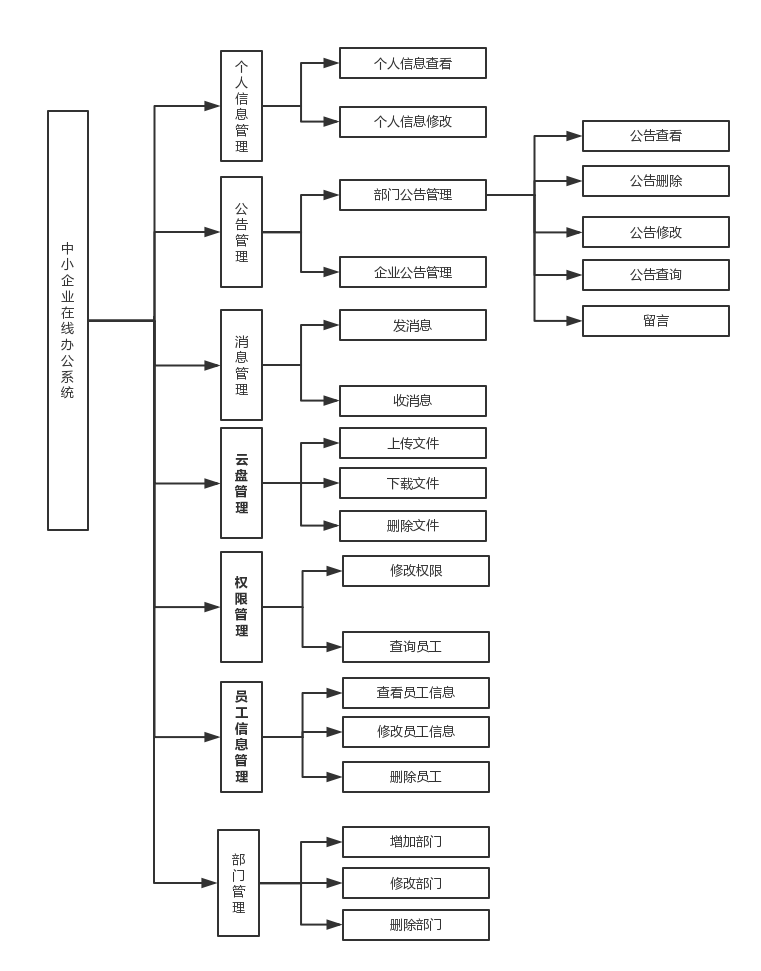
管理层角色用例主要有：个人信息管理、公告管理、云盘管理、信息管理，员工信息管理，企业公告管理，权限管理；管理层是系统的最高级用户，拥有系统的最高权限。管理员在员工信息管理上除了部门经理的功能，还可以对企业内的任何部门内的员工进行管理。在公告管理上除了查阅，修改，删除自己部门的公告，还可以查看，删除所有部门的公告，同时还可以通过特有标签进行公告查看。管理层，还可以给员工进行权限管理。给员工增加额外的权限。管理层用例图见下图所示：



图片 8 管理层用例图

## 系统功能结构图

系统功能结构图可以很好的描述整个系统所实现的功能，形象直观。如下图系统功能结构图，主要是有7个主功能，主功能之后有又些小功能。详细如下图所示：



图片 9 系统功能结构图

# 系统概要设计与实现

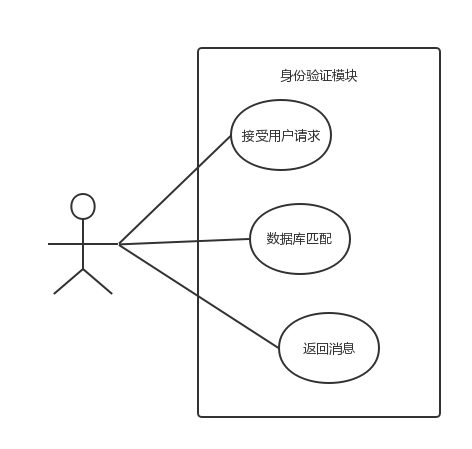
## 功能模块

该系统是基于 B/S(Browser/Server，B/S)架构的三层体系结构，即浏览器和服务器结构。它是随着Internet技术的兴起，对C/S结构的一种变化或者改进的结构。在这种结构下，用户工作界面是通过浏览器来实现，极少部分事务逻辑在前端(Browser)实现，但是主要事务逻辑在服务器端(Server)实现，形成所谓三层结构。这样就大大简化了客户端电脑载荷，减轻了系统的维护与升级的成本和工作量，降低了用户的总体成本。特别是在JAVA这样的跨平台语言出现之后，B/S架构管理软件更是方便、快捷、高效。如果能将该系统划分的用户身份验证模块和其他模块再分解成一些相互独立、功能简单、易于理解和修改的模块，这样的系统就具备了系统设计首要考虑的可变更性。由模块组合构成的系统一般称为模块化结构系统。在这样的系统中，由于各个模块之间基本上是相互独立的，所以每个模块都可以独立的被理解、编程、调试和修改，使其复杂的系统设计工作变得相对简单。在遵循模块化结构设计原则的基础上，根据实现的目标，用户身份验证只需要一个模块，而其他模块则分成了八个不同的子模块，各子模块既相互配合，又各自具有一定的独立功能。这样就形成了一个树形的结构。

### 用户身份验证模块

用户在登入页面，通过表单登入进入到系统主页面，身份验证模块主要又包括三个用例，接收用户请求、数据库验证、返回验证消息。用户在登录界面的时候输入用户名和密码提交数据后，信息会提交给系统，系统接收这些信息传输到后台，后台则从数据库去调用数据并进行匹配，匹配成功后，则开始用户身份验证，验证用户在该系统中的角色，从而展示不同的系统页面。此系统主要是通过用户登录模块来辅助实现管理，通过登录的用户名从用户表中获取用户的权限角色。

身份验证用例图见下图所示：

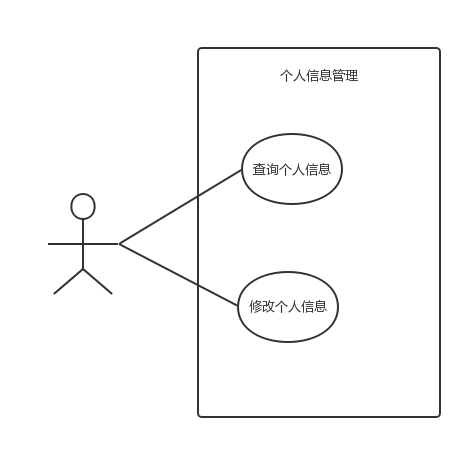


图片 10 身份验证用例图

个人信息管理模块

个人信息管理模块主要包括两个用例：查看个人信息，个人信息修改。用户登入系统后，点击个人信息管理页面，系统从后台调用数据库中的个人信息，返回给系统展示出来。用户修改个人信息后，系统在把数据传递给后台，后台再把数据转递给数据库实现信息的更新。

个人信息管理用例图见下图所示：



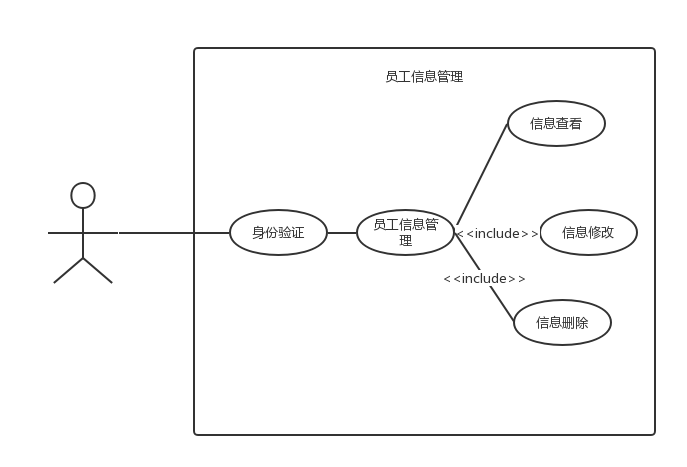
图片 11 个人信息管理模块用例

### 员工信息管理模块

员工信息模块主要包括三个用例：修改职员、删除职员、查看职员。但是本模块又对不同的用户有不同的权限：

1. 部门经理进入该模块会有删、改的对应按钮，但是对应的部门经理只能对本部门的职员进行操作也就是说部门经理只能查看本部门成员的基本信息，和删除、修改部门内成员信息，但是不能修改员工的所在部门。
2. 管理层成员进入该模块可以对企业的全体的职员的信息进行查看、修改和删除，甚至修改部门和权限。

员工信息管理模块用例图见下图所示：

****

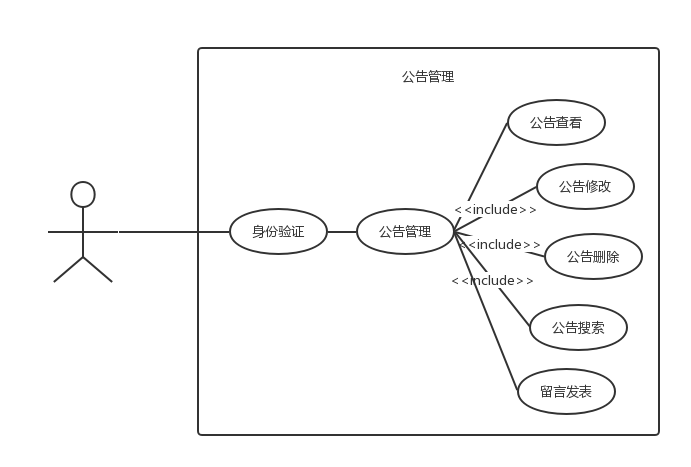
图片 12 员工信息管理模块用例

### 公告管理模块

公告管理模块主要包括四个用例：发表公告、删除公告、查看公告、搜索公告、发表留言。但是本模块又对不同的用户有不同的权限：

1. 普通员工进入该模块只能发表普通公告，只能搜索部门内的公告，只能删除自己的公告。
2. 部门经理进入该模块不仅可以删除自己的公告，还可以删除部门内的任一公告。同时部门经理可以发表带有标签的公告。
3. 管理层成员进入该模块可以对企业的全体的员工的公告进行查看和删除，还可以决定自己发的公告是否放在系统的首页。同时可以对各个部门内的公告进行标签查看。

公告管理模块用例图见下图所示：

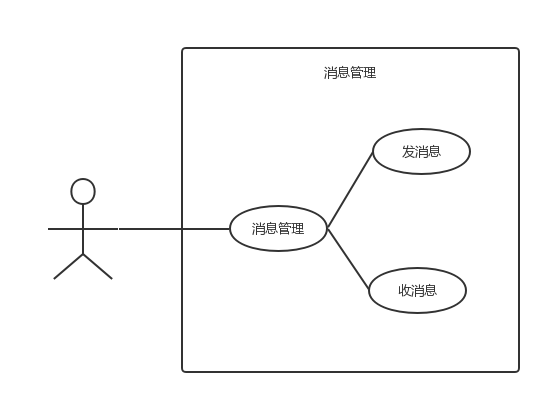


图片 13 公告管理模块用例

### 消息管理模块

短消息管理模块主要包括两个用例：发送消息、接收消息。当前的登录用户给系统内的其他用户发送短消息，并可以查看其他用户发来的消息。

消息管理模块用例图见下图所示：



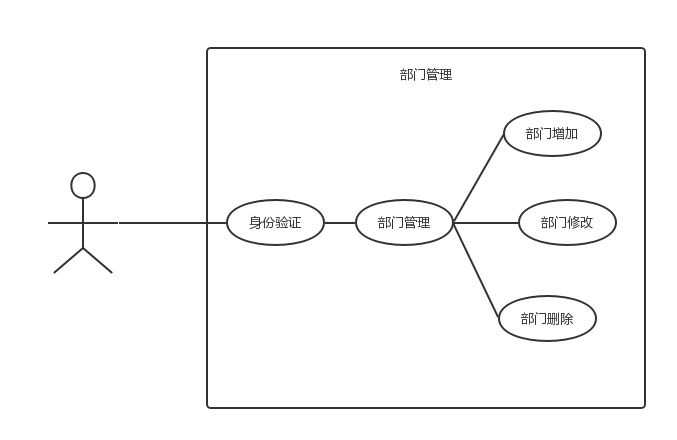
图片 14 消息管理模块用例

### 部门管理模块

部门管理模块主要包括三个用例：增加部门、删除部门、修改部门。用户通过身份验证登录系统之后，不同的角色有不同的功能：

1. 如果身份是普通员工和部门经理则不会有此模块。
2. 如果身份是管理层成员则增加部门、删除部门、修改部门都会有相应的按钮去实现相应的功能。

部门管理模块用例图见下图所示：

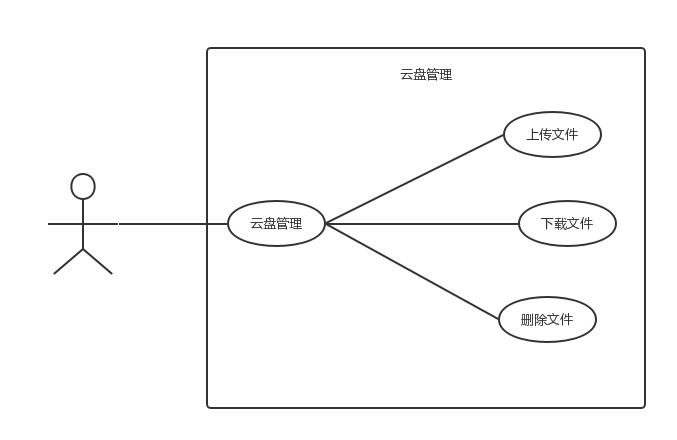


图片 15 部门管理用例

### 云盘管理模块

云盘管理模块主要包括四个用例：上传文件、下载文件、删除文件、查看文件。用户登入系统后，点击云盘，就会进入个人云盘查看自己的文件，同时点击上传按钮，可以上传需要的文档。同时相应的下载，删除功能也有相应的按钮去实现。

云盘管理模块用例图见下图所示：



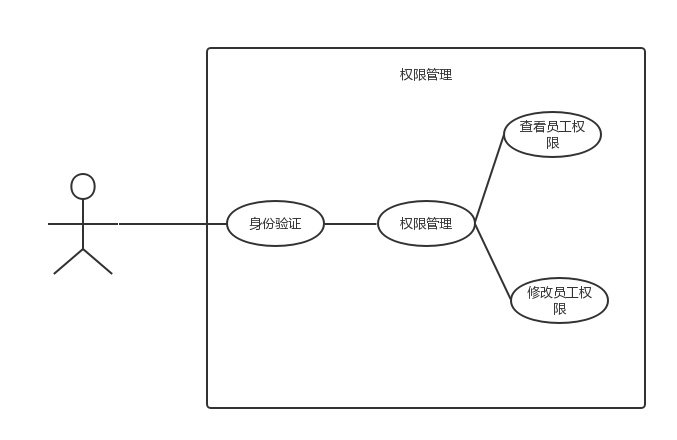
图片 16 云盘管理模块用例

### 权限管理模块

权限管理模块有两个用例：查看用户权限、修改用户权限。权限管理在本系统扮演着重要的角色，权限的设定关系到用户登录系统会呈现怎样的界面，权限管理主要是对员工在不同部门，以及在部门内不同的角色的设定来实现。

1. 部门经理可以对本部门的员工修改部门内的权限。
2. 管理层可以修改企业内不同员工的部门权限。

权限管理模块用例见下图所示：



图片 17 权限管理模块用例

## 数据库设计

数据库设计(Database Design)是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求（信息要求和处理要求）。数据库设计还是项目开发和系统设计中非常重要的一个环节，没有一个好的数据库设计，在后来的系统维护、变更和功能扩充时，甚至在系统开发过程中都会引起比较大的问题。

中小企业在线办公系统使用非关联性的Mongodb作为系统数据库，以便于日后对系统的修改和维护。

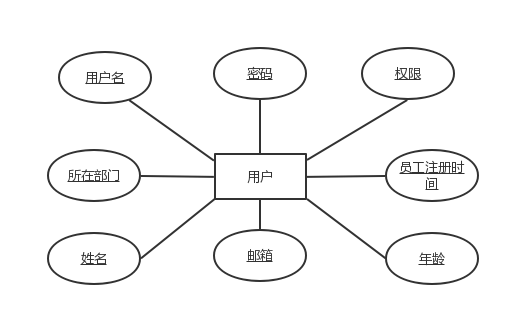
在整个系统数据库设计包含了用户表，部门表，公告表，文件表，消息表等。

### 概念结构设计

概念结构设计的任务是在需求分析阶段产生的需求说明书的基础上，按照特定的方法把它们抽象为一个不依赖于任何具体机器的数据模型，即概念模型。概念模型使设计者的注意力能够从复杂的实现细节中解脱出来，而只集中在最重要的信息的组织结构和处理模式上。

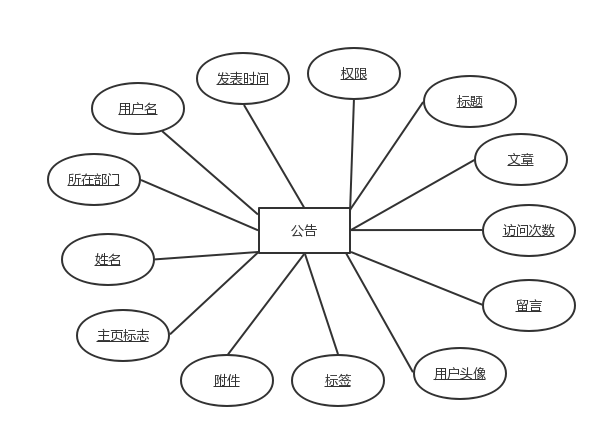
在本系统数据库的概念结构设计上，我们采用E-R图来描述数据库的实体关系。E-R图也称实体-联系图(Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。它是描述现实世界关系概念模型的有效方法，是表示概念关系模型的一种方式。用“矩形框”表示实体型，矩形框内写明实体名称；用“椭圆图框”表示实体的属性，并用“实心线段”将其与相应关系的“实体型”连接起来。

1. 员工信息管理模块的E-R，见下图所示：



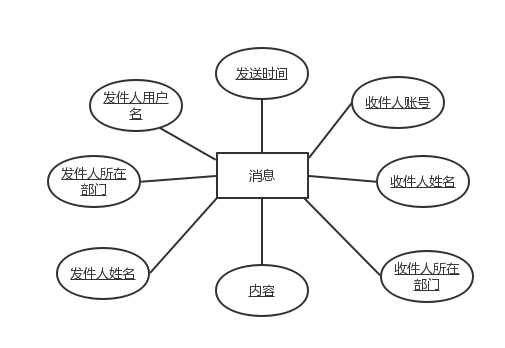
图片 18 员工信息管理模块

1. 公告管理模块的E-R，见下图所示：



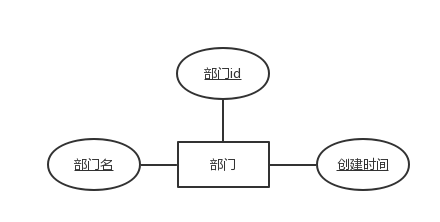
图片 19 公告管理模块

1. 短消息模块的E-R图，见下图所示：



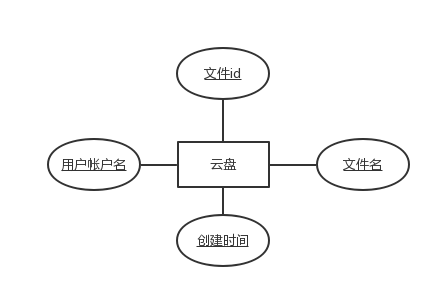
图片 20 短消息模块

1. 部门管理模块的E-R图，见下图所示：



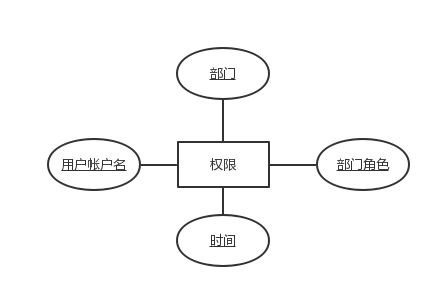
图片 21 部门管理模块

1. 云盘管理模块的E-R，见下图所示：



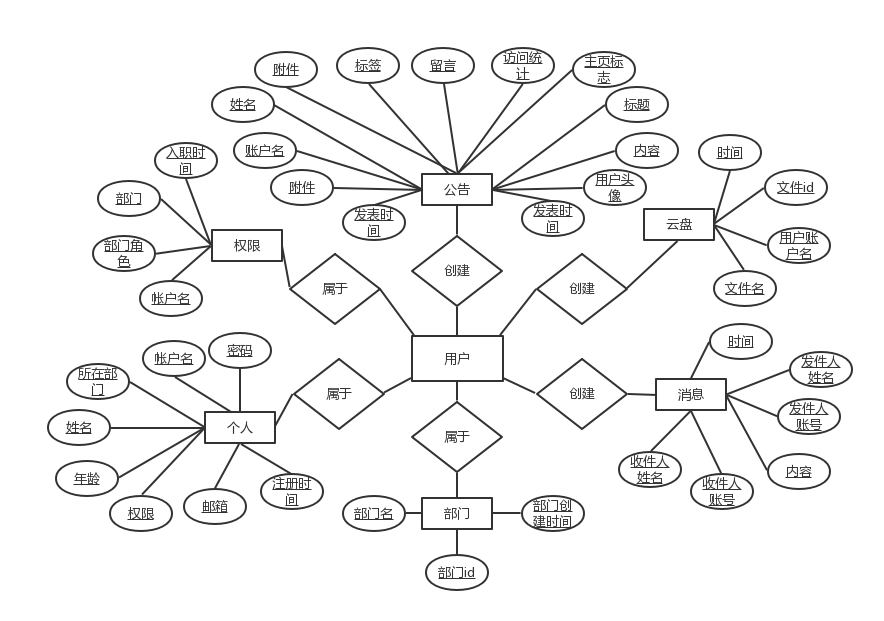
图片 22云盘管理模块

1. 权限管理模块的E-R，见下图所示：



图片 23权限管理模块

1. 整体的E-R，见下图所示：



图片 24 整体E-R图

### 关系模式设计

综合以上分析的概念结构设计和各个模块的E-R图，我们可以得到以下的关系模式:

1) 员工信息（员工帐户名，员工姓名，账号密码，员工邮箱，员工籍贯，员工注册时间，员工年龄，员工所在部门，员工头像，员工部门权限）

2) 部门（部门ID，部门名称，部门创建时间）

3) 云盘（文件id，员工帐户名，文件名，文件创建时间）

4) 消息（发送者姓名，接收者姓名，发送者帐户名，接收机帐户名，消息内容，发送时间）

5) 公告（员工帐户名，员工姓名，员工所在部门，员工头像，员工部门权限，发表时间，标题，主页标志，标签，公共内容，留言，附件，访问统计）

详细的数据库表见以下表格：

1. 员工信息表：用于存储员工的基本信息，见下表所示：

表格 1员工信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 可否为空 | 默认值 | 详细说明 |
| \_id | ObjectId | 是 | 否 | 无 | 数据库分配唯一id |
| iid | String | 否 | 否 | 无 | 帐户名 |
| name | String | 否 | 否 | 无 | 姓名 |
| password | String | 否 | 否 | 无 | 密码 |
| email | String | 否 | 否 | 无 | 邮箱 |
| head | String | 否 | 否 | 无 | 头像 |
| age | Int | 否 | 否 | 无 | 年龄 |
| idepartment | String | 否 | 否 | 无 | 所在部门 |
| idate | Object | 否 | 否 | 无 | 创建时间 |
| place | String | 否 | 否 | 无 | 籍贯 |
| irole | Int | 否 | 否 | 无 | 部门权限 |
| Iphone | String | 否 | 否 | 无 | 手机号 |

1. 部门信息表：用于存储部门的基本信息，见下表所示：

表格 2部门信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 可否为空 | 默认值 | 详细说明 |
| \_id | ObjectId | 是 | 否 | 无 | 数据库分配唯一id |
| iid | String | 否 | 否 | 无 | 部门id |
| name | String | 否 | 否 | 无 | 部门名称 |
| time | String | 否 | 否 | 无 | 创建时间 |

1. 个人云盘信息表：用于存储部门的基本信息，见下表所示：

表格 3个人云盘信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 可否为空 | 默认值 | 详细说明 |
| \_id | ObjectId | 是 | 否 | 无 | 数据库分配唯一id |
| iid | String | 否 | 否 | 无 | 文件id |
| name | String | 否 | 否 | 无 | 员工账户名称 |
| filename | String | 否 | 否 | 无 | 文件名 |
| time | Object | 否 | 否 | 无 | 创建时间 |

1. 消息信息表：用于存储部门的基本信息，见下表所示：

表格 4消息信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 可否为空 | 默认值 | 详细说明 |
| \_id | ObjectId | 是 | 否 | 无 | 数据库分配唯一id |
| reid | String | 否 | 否 | 无 | 接收者帐户名 |
| rename | String | 否 | 否 | 无 | 接收者姓名 |
| seid | String | 否 | 否 | 无 | 发送者帐户名 |
| sename | String | 否 | 否 | 无 | 发送者姓名 |
| message | String | 否 | 否 | 无 | 信息内容 |
| time | String | 否 | 否 | 无 | 创建时间 |

1. 公告信息表：用于存储公告的基本信息，见下表所示：

表格 5 公告信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否主键 | 可否为空 | 默认值 | 详细说明 |
| \_id | ObjectId | 是 | 否 | 无 | 数据库分配唯一id |
| iid | String | 否 | 否 | 无 | 帐户名 |
| name | String | 否 | 否 | 无 | 姓名 |
| title | String | 否 | 否 | 无 | 公告标题 |
| mainpost | String | 否 | 否 | null | 主页标志 |
| head | String | 否 | 否 | 无 | 头像 |
| tags | Array | 否 | 是 | null | 标签 |
| idepartment | String | 否 | 否 | 无 | 所在部门 |
| time | Object | 否 | 否 | 无 | 创建时间 |
| files | String | 否 | 是 | null | 附件 |
| irole | Int | 否 | 否 | 2 | 部门权限 |
| post | String | 否 | 是 | null | 公告内容 |
| comments | Array | 否 | 是 | null | 留言 |
| pv | Int | 否 | 是 | null | 访问统计 |

# 系统编码的实现

## 系统架构

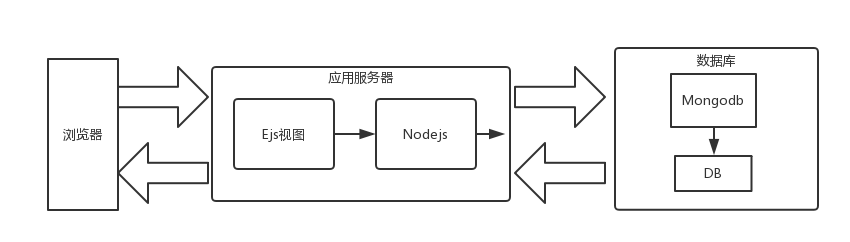
本系统以Nodejs+Mongodb架构为基础，加上Angularjs作为必要的前端框架，按照实际的开发流程来进行开发。包括如下：

Angularjs功能开发：主要是给用户提供一个完美的UI界面，供用户实现科学化的人机交互，用户可以通过Ejs页面进行交互。

Mongodb功能开发：开发系统需要的数据模式和数据结构，并对系统要用的数据进行检验，保证数据的安全性，使得系统健壮的运行。

Nodejs功能开发：为系统搭建一个可靠的后台服务器，使得前端到后端的交互更加的快速。

### 系统架构图



图片 25 系统架构图

### 程序结构图

本系统源程序包的目录结构见下图所示。

OA为项目的名称，也是项目的根目录。

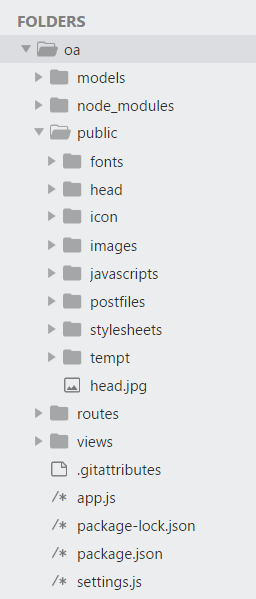
public为系统存放目录，包括images、fonts、stylesheets、javascript、icon、head等目录。

views目录存放本系统的视图。

routs目录存放本系统的路由以及后台文件。

models目录存放本系统的数据模型和数据库的访问文件。

Node\_models目录存放本系统的需要用到的各种模块文件。



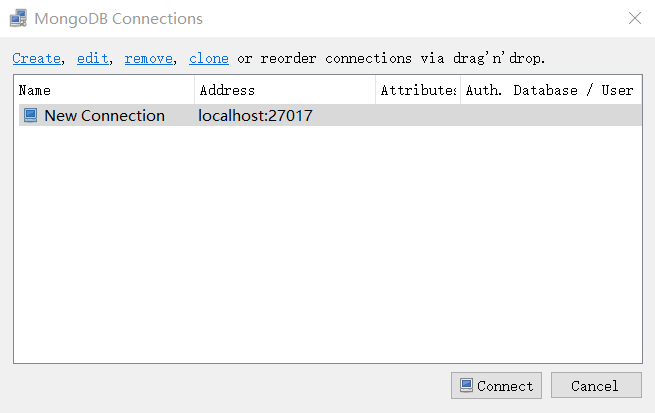
图片 26程序结构图

## 模块实现

### 数据库的链接与实现

本系统采用的是Mongodb数据库，以Robo3T作为Mongodb的界面操作，在Mongodb数据中建立数据库名为：oa，用户名和密码分别为root、123456。

数据库的链接见下图所示。



图片 27数据库链接

Nodejs连接Mongodb数据库的代码实现：

|  |
| --- |
| / module.exports={  cookieSecret:'myoa',  db:'oa',  host:'localhost'  };  var settings = require('../settings'),  Db = require('mongodb').Db,  Connection = require('mongodb').Connection,  Server =require('mongodb').Server;  module.exports = new Db(settings.db,new Server(settings.host,27017),{safe:true}); |

用户登入模块

用户首先进入系统是通过登录界面如下图所示的，用户登陆时如果用户没有自己的账号则可以点击“注册”按钮进入注册界面；如果用户已注册但是未通过管理员的审核会弹出提示“请等待审核”；如果用户是已审核过的用户会直接进入主界面。



图片 28登入界面



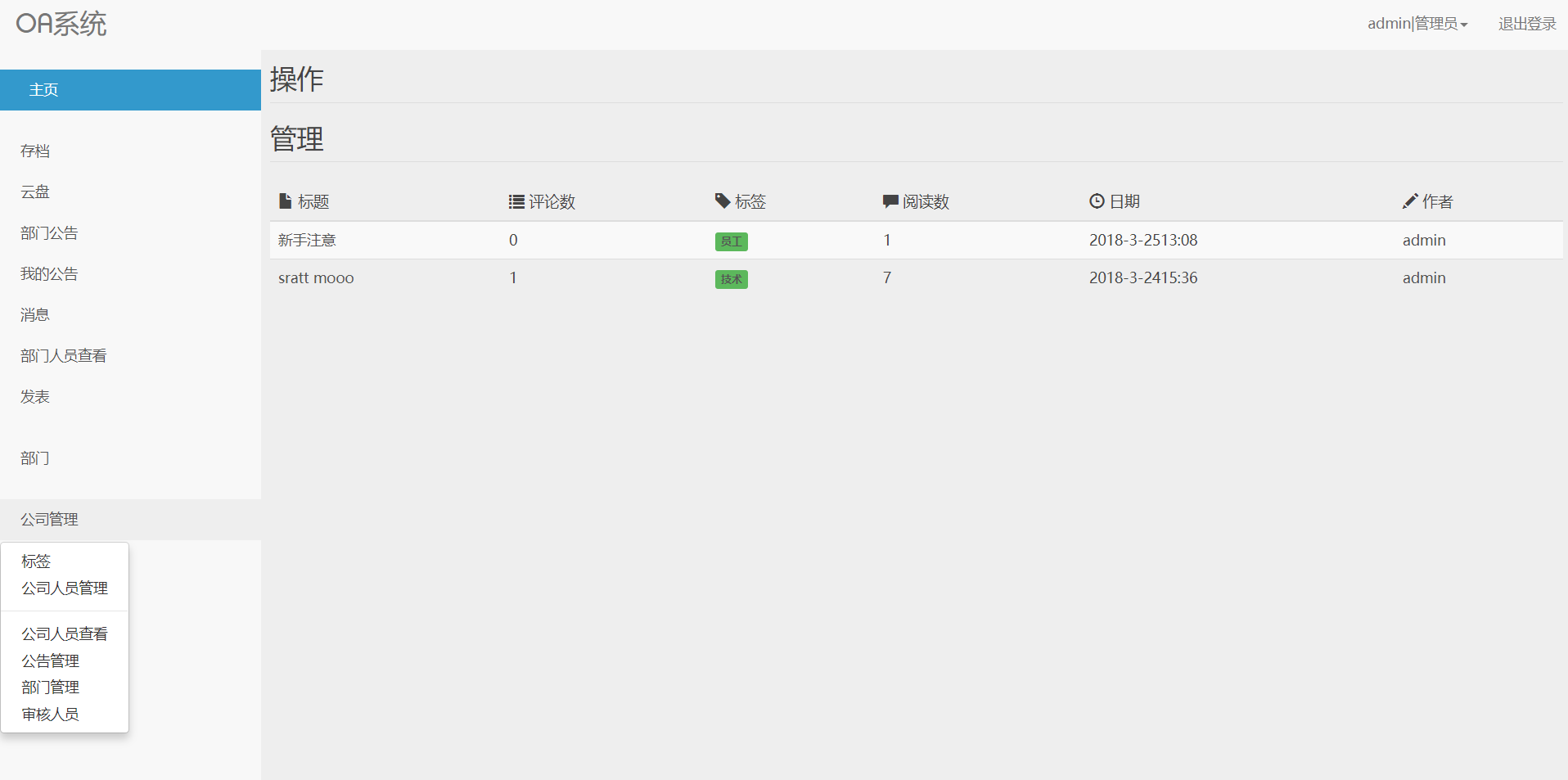
图片 29注册界面

用户登录的核心代码：

|  |
| --- |
| / app.post('/login',function(req,res){  //生成密码的MD5值  var md5=crypto.createHash('md5'),  password=md5.update(req.body.password).digest('hex');  //检查用户名是否存在  User.get(req.body.id,function(err,user,department,role){  if (!user) {  req.flash('error','用户名已存在');  return res.redirect('/login');  }  //检查两次输入密码一样否？  if (user.password !=password) {  req.flash('error','密码错误');  return res.redirect('/login');  }  if (department =='待定') {  req.flash('error','请等待管理员审核通过');  return res.redirect('/login');  }  var path='./public/head/'+req.body.id;  fs.exists(path,function(exists){  if(exists){  var pathnamefile='./public/head/'+req.body.id+'/'+'head.png';  var target\_path ='./public/'+'head.jpg';  //转移文件  fs.writeFileSync(pathnamefile, fs.readFileSync(target\_path));  req.session.department=department;  req.session.role=role;  req.session.user=user;  req.flash('success','登入成功');  res.redirect('/');  }  if(!exists){  fs.mkdir(path,function(err){  if (err) {  console("创建失败"+err);  }  var pathnamefile='./public/head/'+req.body.id+'/'+'head.png';  var target\_path ='./public/'+'head.jpg';  //转移文件  fs.writeFileSync(pathnamefile, fs.readFileSync(target\_path));  req.session.department=department;  req.session.role=role;  req.session.user=user;  req.flash('success','登入成功');  res.redirect('/');  });  }  });    });  }); |

### 主页面

不同的用户角色会进入系统，则会到看到不同的的主界面，所以需要获取当前用户的角色来判断是该用户有什么样的权限。而用户登录系统时后台会保存当前用户的信息到session中，系统在session获取用户的信息后，则得到用户的权限，前端界面则呈现不同的信息。角色分为三类会看到三种不同的主界面，有管理层管理员的，部门经理的，普通员工的。主界面见下图所示：



图片 30 主页面

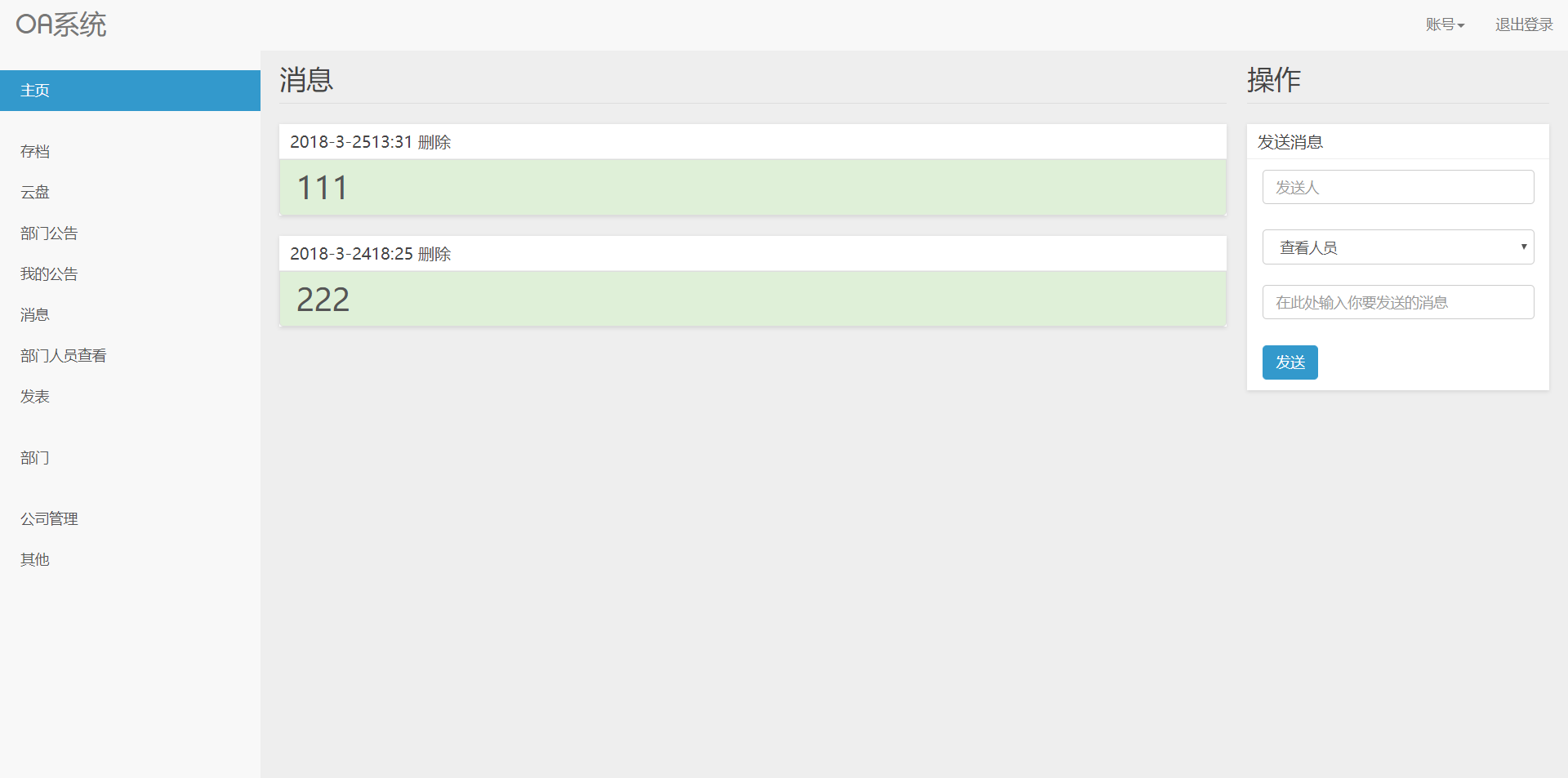
用户主界面相关核心代码：

|  |
| --- |
| app.get('/',checkLogin);  app.get('/',function(req,res){  var page=req.query.p ? parseInt(req.query.p) :1;  Post.getmainpost(page,function(err,posts,total){  if (err) {  posts=[];  }  res.render('index',{  title:'主页',  posts:posts,  page:page,  isFirstPage:(page-1)==0,  isLastPage:((page-1)\*10+posts.length)==total,  department:req.session.department,  role:req.session.role,  user:req.session.user,  success:req.flash('success').toString(),  error:req.flash("error").toString()  });  });  }); |

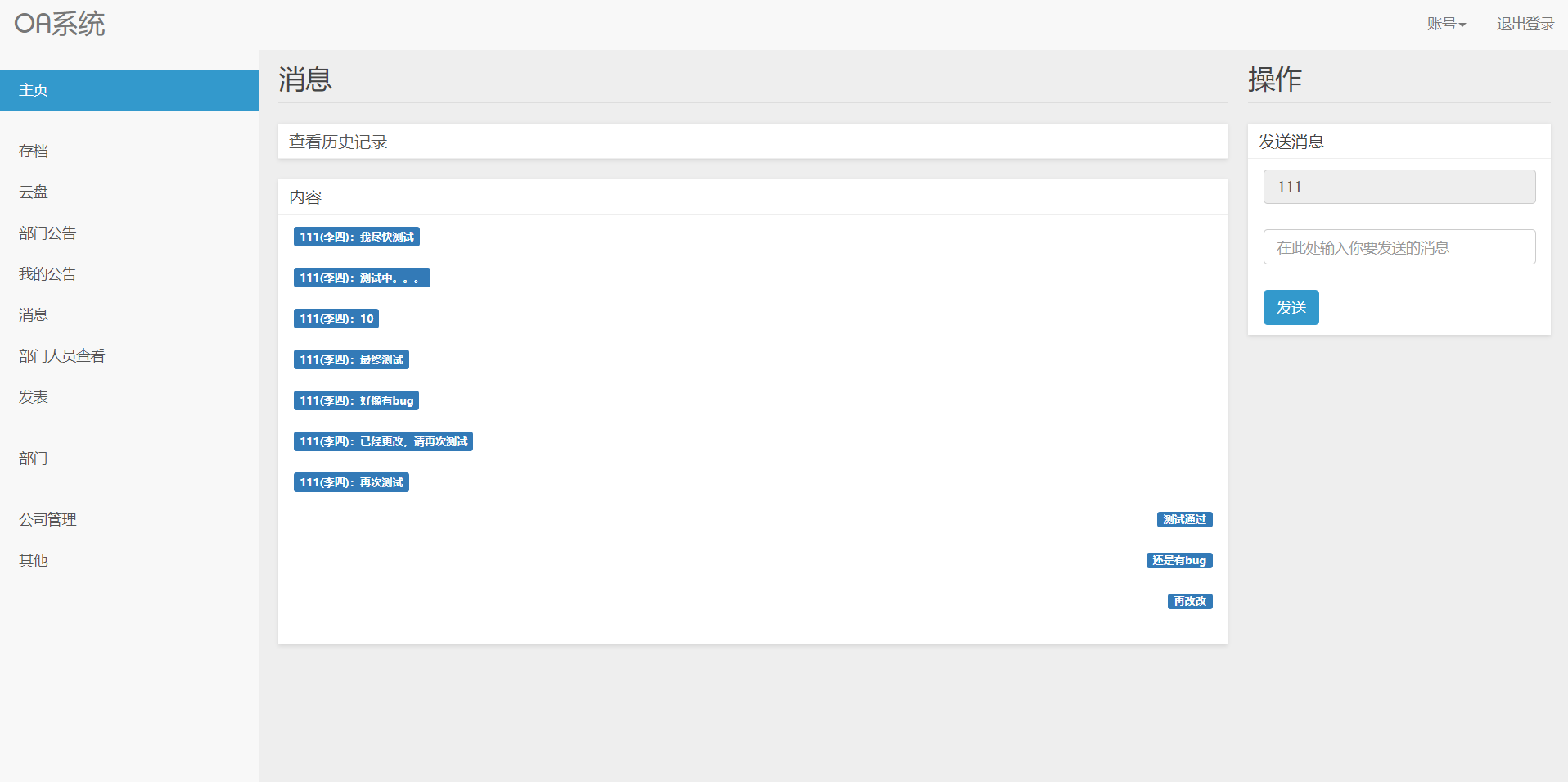
消息管理

用户点击消息按钮后，则会进入消息页面。在消息页面会根据不同用户发出消息的时间进行排序，时间越近的排在前面，同时会根据不同消息的性质显示不同的颜色。未读消息为红色，已读消息为绿色。同时用户点击用户名，显示用户之间的详细对话信息。

消息页面和对话页面见下图：



图片 31消息界面



图片 32 对话页面

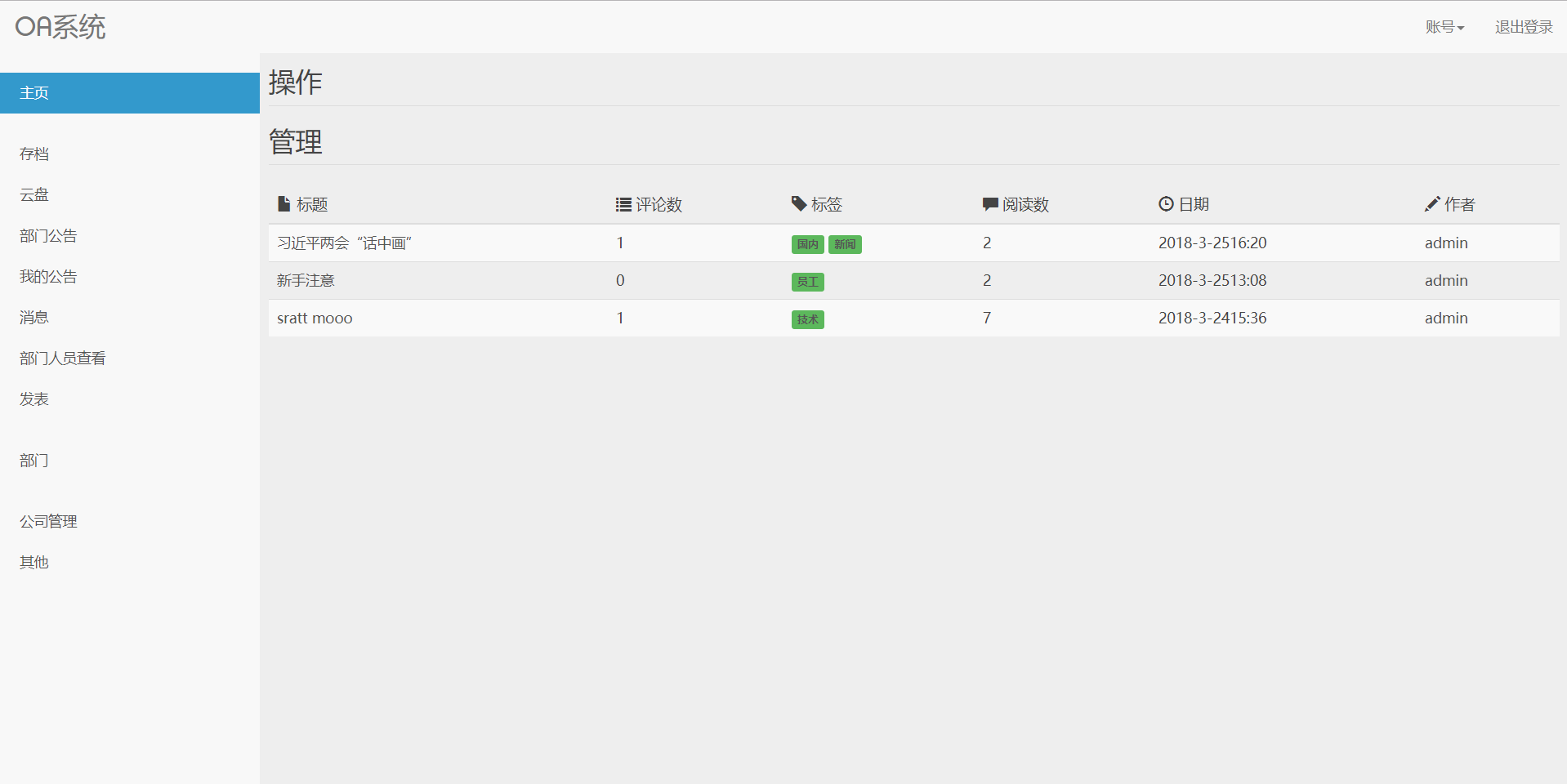
消息页面核心代码：

|  |
| --- |
| / app.get('/imessage',checkLogin);  app.get('/imessage',function(req,res){  var page=req.query.p ? parseInt(req.query.p) :1;  User.getall(function(err,alluser){  Messages.getTen(page,req.session.user.iid,function(err,messages,total){  if (err) {  posts=[];  }  res.render('message',{  title:'主页',  messages:messages,  alluser:alluser,  page:page,  isFirstPage:(page-1)==0,  isLastPage:((page-1)\*10+messages.length)==total,  department:req.session.department,  role:req.session.role,  user:req.session.user,  success:req.flash('success').toString(),  error:req.flash("error").toString()  });  });  });  }); |

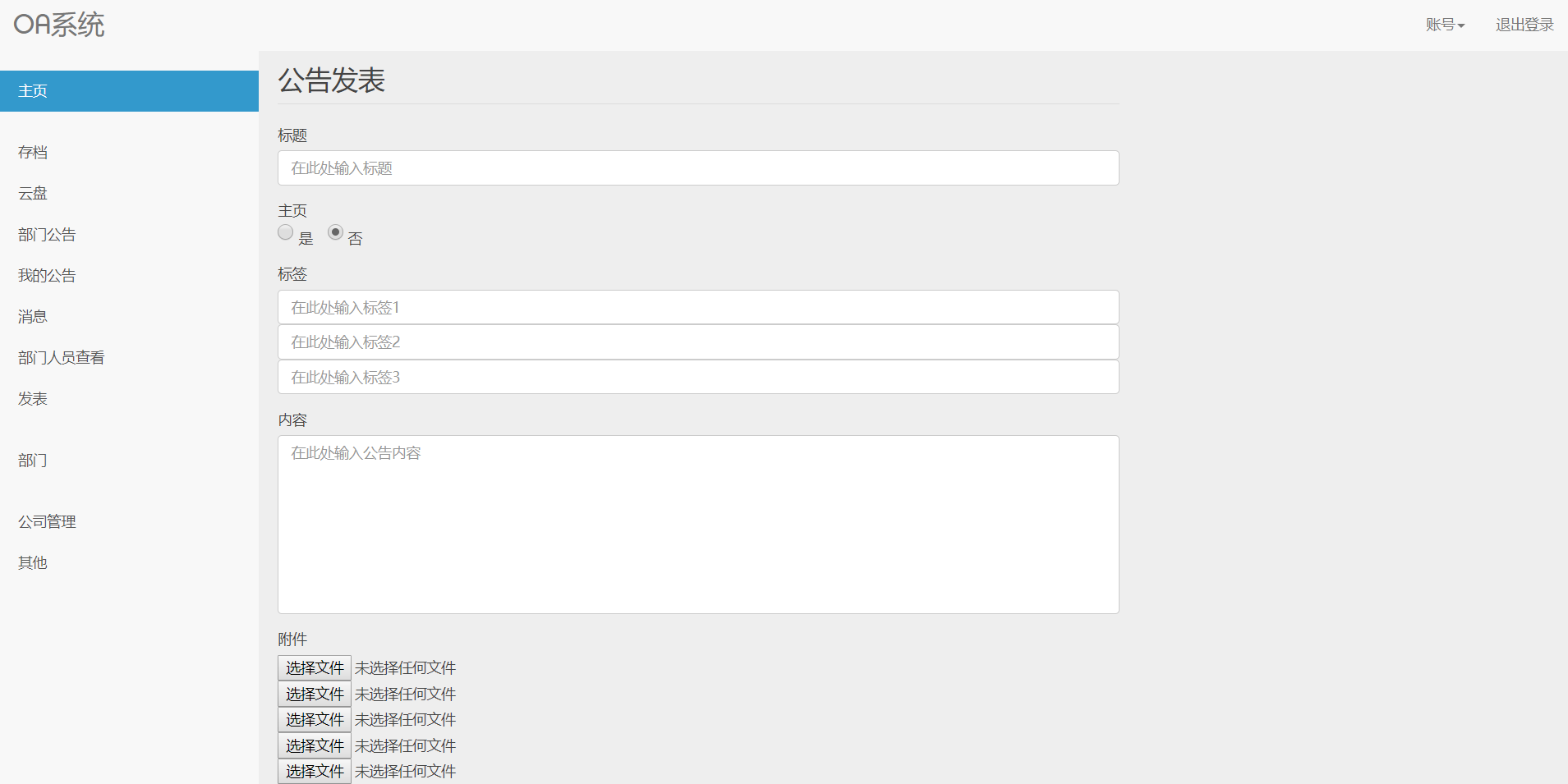
### 公告

用户登入后，就会进入主页看到主页的公告。同时点击部门公告，也可看到部门公告。用户点击发表就可发表自己的公告。但系统会根据用户的权限，对应显示不同的用户公告发表的页面。发表页面会xianshi1三种不同权限的页面，分别是管理层管理员的，部门经理的，个普通员工的。用户点击公告的标题后，就会进入到公告的详细页面，在详细页面，用户可以阅读公告的详细信息，下载公告附带的文件，也可对该公告进行留言。

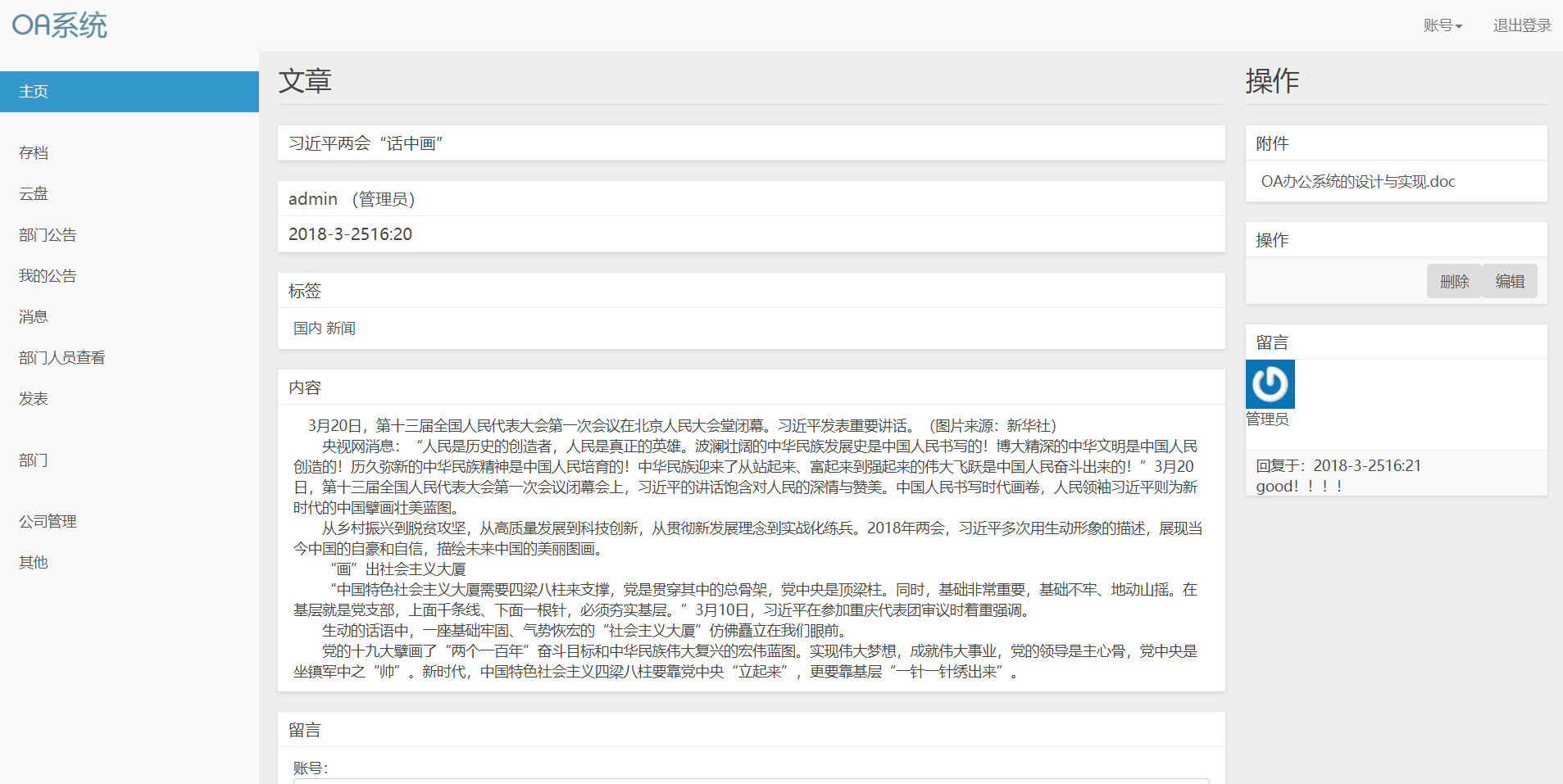
详细的部门公告，公告发表页面见下图：



图片 33 部门公告



图片 34 发表



图片 35 发表

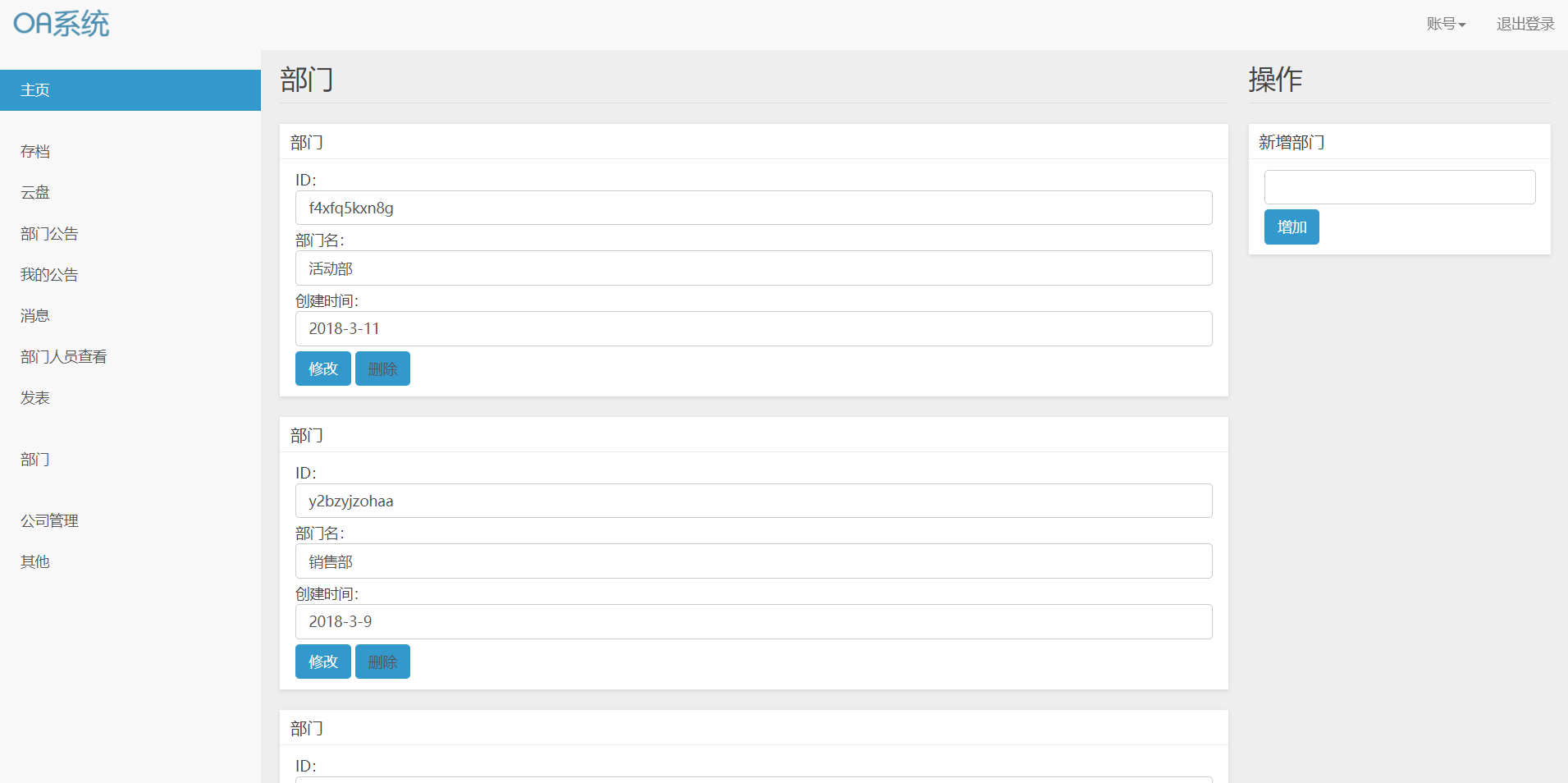
发公告的核心代码：

|  |
| --- |
| app.post('/post',function(req,res){ var currentUser=req.session.user,  tags=[req.body.tag1,req.body.tag2,req.body.tag3],  post= new Post(currentUser.iid,currentUser.name,req.session.department,req.session.role,currentUser.head,req.body.title,req.body.mainpost,tags,req.body.post);  post.save(function(err){  if (err) {  req.flash('error',err);  return res.redirect('/');  }  var path2='./public/postfiles/'+req.session.user.iid+'/'+req.body.title; //检查目录，不存在就创建  var files=[];  var j=0;  console.log('start');  console.log(path2);  fs.exists(path2,function(exists){  if(exists){  console.log("目录存在")  }  if(!exists){  fs.mkdir(path2,function(err){  if (err) {  console.log("创建失败"+err);  }  for (var i in req.files) {  if (req.files[i].size == 0) {  //使用同步方式删除一个文件  fs.unlinkSync(req.files[i].path);  console.log('success removed an empty file!');  }else{  console.log('start add filename'+i);  files[j]=req.files[i].name;  console.log(files[j]+j);  var pathnamefile='./public/postfiles/'+req.session.user.iid+'/'+req.body.title+'/'+req.files[i].name;  var target\_path ='./public/tempt/'+req.files[i].name;  //使用同步方式重命名一个文件  fs.renameSync(req.files[i].path,target\_path);  //转移文件  fs.writeFileSync(pathnamefile, fs.readFileSync(target\_path));  //删除文件  fs.unlink(target\_path,function(err){  if (err) {  console.log("删除失败"+err);  }  });  };  j++;  };  var filename={  files1:files[0],  files2:files[1],  files3:files[2],  files4:files[3],  files5:files[4],  }  var newPostfiles=new Postfiles(currentUser.iid,req.body.title,req.body.post,filename);  newPostfiles.save(function(err){  console.log(currentUser.iid+req.body.title+req.body.post+files);  if (err) {  req.flash('error',err);  return res.redirect('back');  }  req.flash('success','发表成功');  return res.redirect('/department');  });  });  };  });  });  }); |

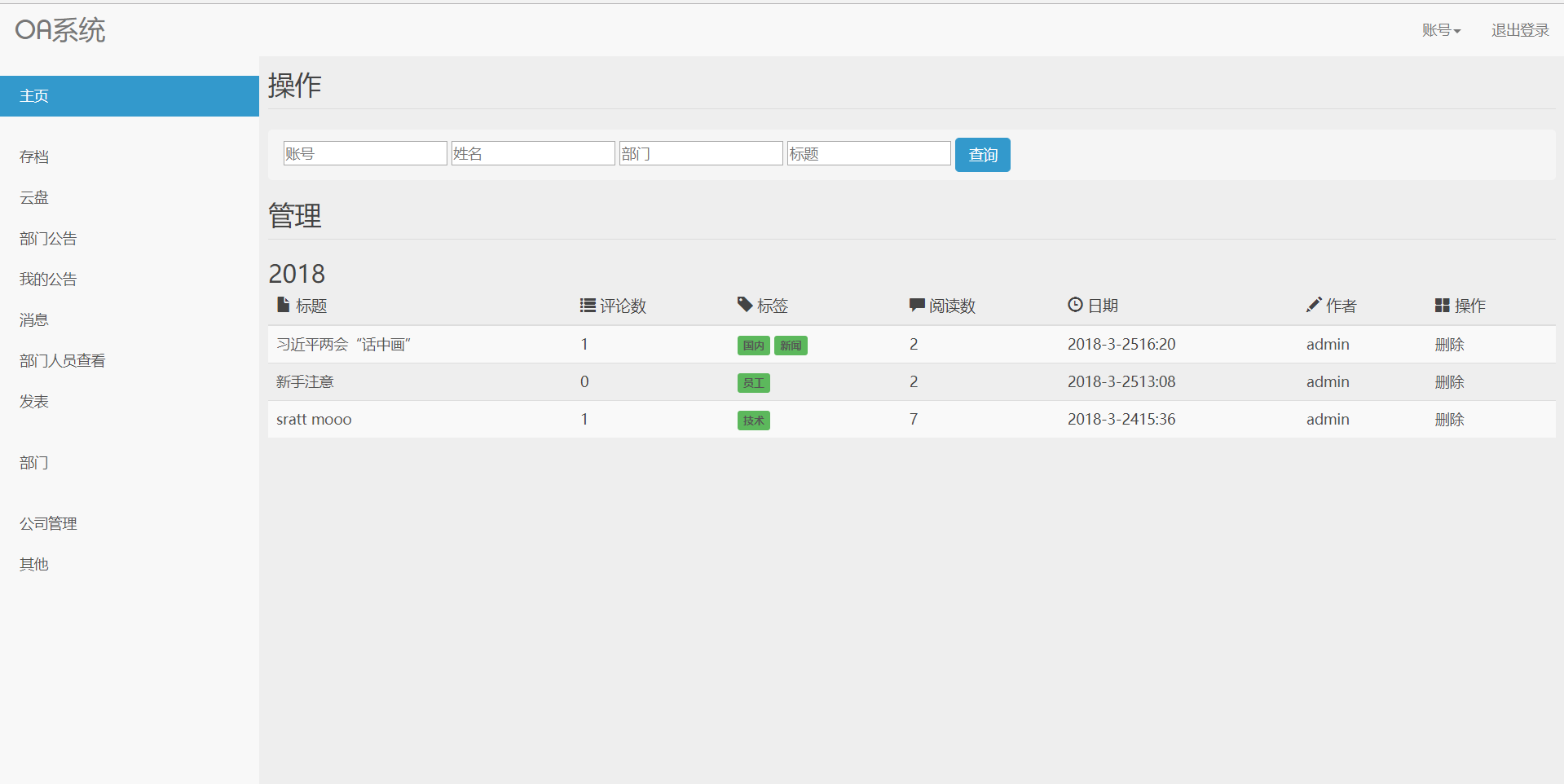
### 部门管理

用户进入系统后，如果权限是最高级，也就是管理层管理员的身份，在主页面会显示部门管理的功能。进入部门管理就会对企业的部门进行修改和删除。如果权限允许，还可对部门的人员进行管理，以及对部门人员的公告进行删除等操作。

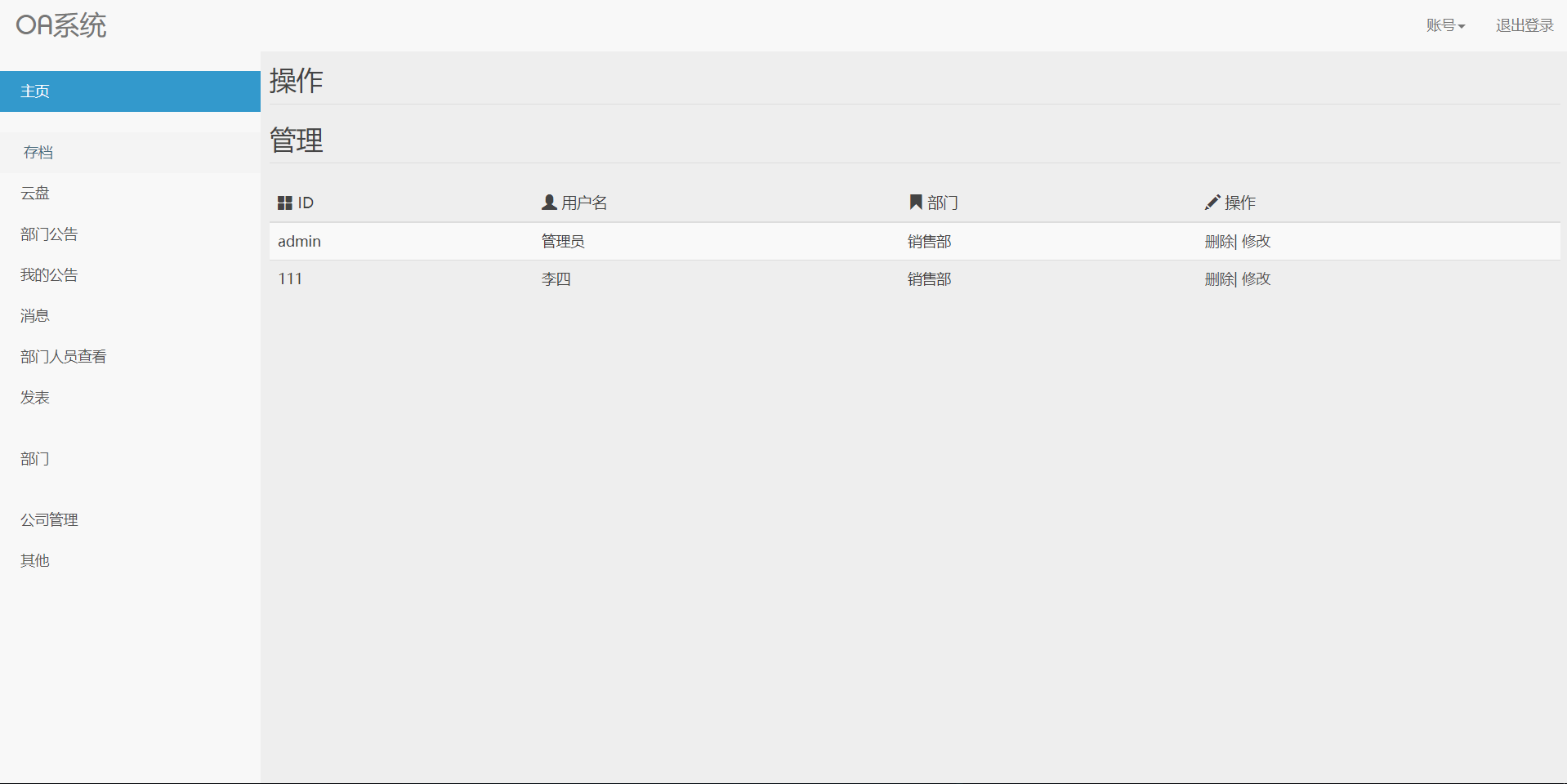
部门管理，员工管理，公告管理的页面如下图：



图片 36部门管理



图片 37 部门公告管理



图片 38部门人员管理

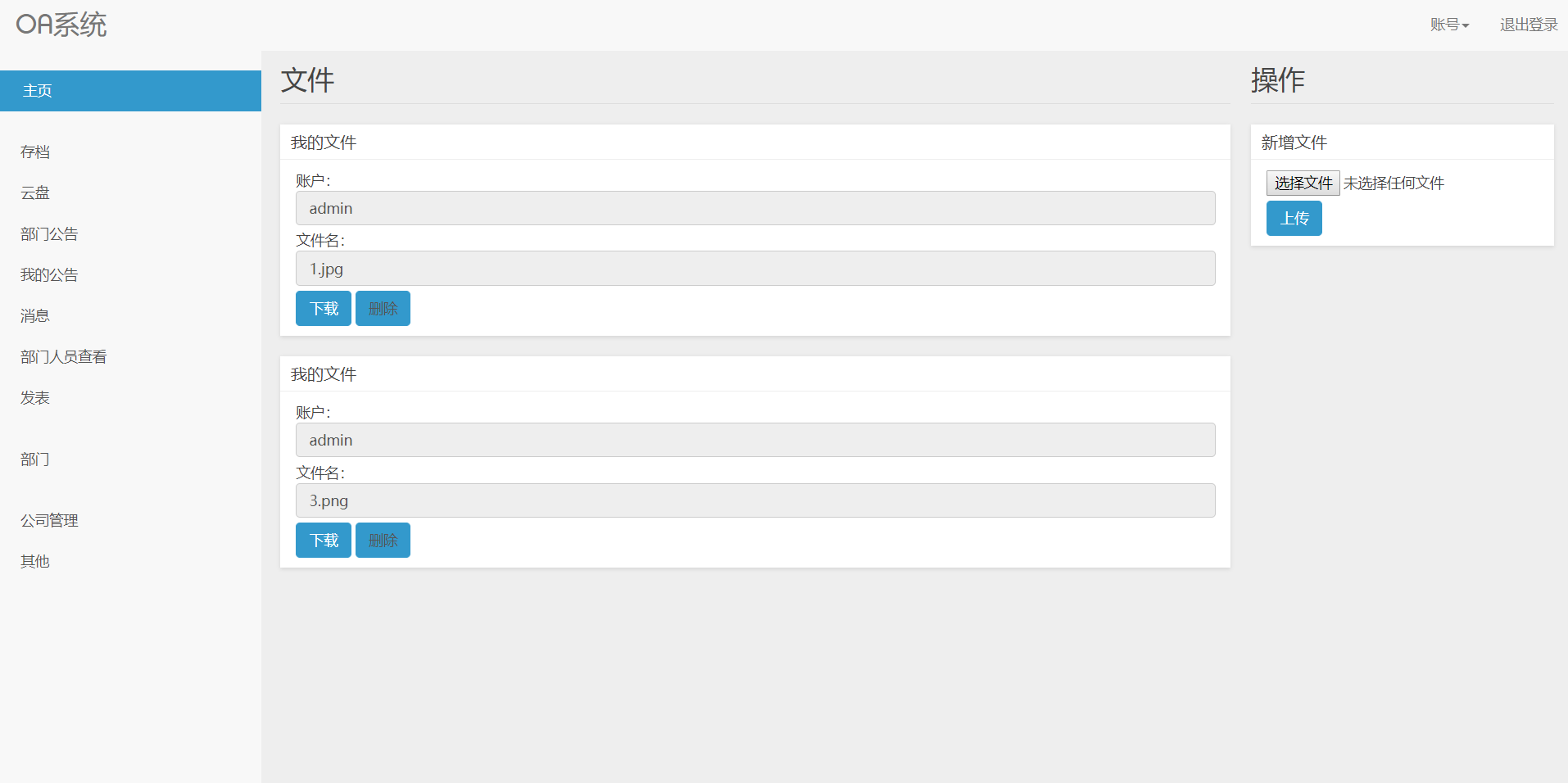
部门管理核心代码如下;

|  |
| --- |
| app.get('/departmentmanage',function(req,res){  var page=req.query.p ? parseInt(req.query.p) :1;  Department.getTen(page,function(err,departments,total){  if (err) {  departments=[];  }  res.render('departmentmanage',{  title:'主页',  page:page,  isFirstPage:(page-1)==0,  isLastPage:((page-1)\*10+departments.length)==total,  departments:departments,  user:req.session.user,  department:req.session.department,  role:req.session.role,  success:req.flash('success').toString(),  error:req.flash("error").toString()  });  });  }); |

### 个人云盘管理

用户进入系统后，系统会为用户单独建立文件夹，方便用户上传自己的文件，方便用户的办公。用户点击云盘就会进入到自己的个人云盘，在哪里用户可以管理自己的文件，包括上传、下载，删除等功能。

云盘的界面如下图所示;



图片 39云盘

云盘的核心代码如下：

|  |
| --- |
| app.post('/upload',checkLogin);  app.post('/upload',function(req,res){  for (var i in req.files) {  if (req.files[i].size == 0) {  //使用同步方式删除一个文件  fs.unlinkSync(req.files[i].path);  console.log('success removed an empty file!');  }else{  var filename=req.files[i].name;  var newFiles =new Files({  name:req.session.user.iid,  filename:req.files[i].name  });  //检查用户名是否存在  Files.get(req.session.user.iid,filename,function(err,file){  if (file) {  req.flash('error','文件名已存在');  return res.redirect('/');  }  //如果不存在则加入新用户  newFiles.save(function(err,filename){  if (err) {  req.flash('error',err);  return res.redirect('/upload');  }  });  var pathnamefile='./public/images/'+req.session.department+'/'+req.session.user.iid+'/'+req.files[i].name;  var target\_path ='./public/tempt/'+req.files[i].name;  //使用同步方式重命名一个文件  fs.renameSync(req.files[i].path,target\_path);  //转移文件  fs.writeFileSync(pathnamefile, fs.readFileSync(target\_path));  //删除文件  fs.unlink(target\_path,function(err){  if (err) {  console.log("删除失败"+err);  }  })  req.flash('success','文件上传成功');  return res.redirect('/myfiles');  });  }    };  }); |

# 系统测试

软件测试（英语：software testing），描述一种用来促进鉴定软件的正确性、完整性、安全性和质量的过程。软件测试的经典定义是：在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。

软件测试主要分为：黑盒测试、白盒测试、灰盒测试。

黑盒测试：

黑盒测试也称功能测试或数据驱动测试，它是在已知产品所应具有的功能，通过测试来检测每个功能是否都能正常使用，在测试时，把程序看作一个不能打开的黑盆子，在完全不考虑程序内部结构和内部特性的情况下，测试者在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数锯而产生正确的输出信息，并且保持外部信息（如数据库或文件）的完整性。

白盒测试：

白盒测试也称结构测试或逻辑驱动测试，它是知道产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都有能按预定要求正确工作，而不顾它的功能，白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。 “白盒”法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。

灰盒测试：

灰盒测试是介于二者之间的，灰盒测试关注输出对于输入的正确性，同时也关注内部表现。

本系统主要采用的是黑盒测试。

## 测试设计与测试结果

登录功能的测试

输入账号和密码，点击登录，观察账号登录情况。

登录模块测试内容如下表所示。

表格 6 登入模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 01 | 测试用例名称 | 登录模块 |
| 测试功能描述 | 输入账号密码登录系统，根据权限进入相应的主界面 | | |
| 测试步骤 | 1.输入账号  2.输入密码  3.点击登录 | | |
| 期待输出结果 | 1. 当账号不存在时，提示账号错误。 2. 当账号存在但是密码不正确时，提示检查账号密码 3. 若输入账号密码正确，但没被审核通过，提示请等待审核员通过。。 4. 若输入的是管理层管理员账号，当账号密码对应且正确时，跳转至管理层管理员主界面。 5. 若输入的是部门经理人员账号，当账号密码对应且正确时，跳转至部门经理主界面。 6. 若输入的是普通员工账号，当账号密码对应且正确时，跳转至普通员工主界面。 | | |

### 注册模块测试

输入账号、密码和邮箱，点击注册，观察账号注册情况。

注册模块测试内容如下表所示。

表格 7 注册模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 02 | 测试用例名称 | 注册模块 |
| 测试功能描述 | 注册账号，看系统给出的反应 | | |
| 测试步骤 | 1.输入账号  2.输入密码  3.输入邮箱  4.点击注册 | | |
| 期待输出结果 | 1. 数据查找没有相同的账户，提示注册成功，请等待管理员审核 2. 数据查找有相同的账户，提示该账号存在，请重新注册 | | |

### 公告管理功能模块测试

登入系统后，点击部门公告管理，观察系统情况。

公告管理模块测试内容如下表所示。

表格 8 公告管理模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 03 | 测试用例名称 | 公告管理模块测试 |
| 测试功能描述 | 获取数据库中的公告，并对公告进行删除，修改操作 | | |
| 测试步骤 | 点击公告管理，进入公告展示页面 | | |
| 期待输出结果 | 公告显示，并有公告操作界面 | | |

公告查看测试内容

登入系统后，点击部门公告，观察系统情况。

公告查看模块测试内容如下表所示。

表格 9 公告查看测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 04 | 测试用例名称 | 公告查看模块 |
| 测试功能描述 | 获取数据库中公告的详细内容，及附带的文件。 | | |
| 测试步骤 | 点击部门公告，进入公告详情页面 | | |
| 期待输出结果 | 公告详情显示成功 | | |

### 留言模块测试内容

进入公告详情页面，输入留言内容，点击留言按钮，观察系统情况。

留言模块测试内容如下表所示。

表格 10 留言模块测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 05 | 测试用例名称 | 留言模块 |
| 测试功能描述 | 进行正常的公告留言 | | |
| 测试步骤 | 进入公告详情页面，输入留言内容，点击留言 | | |
| 期待输出结果 | 页面展示留言内容 | | |

个人信息管理模块测试内容

进入系统后，点击个人信息，看是否加载个人信息，并可修改个人情况。

个人信息管理模块测试内容如下表所示。

表格 11 个人信息管理

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 06 | 测试用例名称 | 个人信息管理模块 |
| 测试功能描述 | 正常加载个人信息，并可对信息进行操作 | | |
| 测试步骤 | 1. 进入个人信息页面 2. 修改个人信息 3. 再次查看个人信息 | | |
| 期待输出结果 | 正常显示个人信息，并修改后信息正常加载。 | | |

### 部门人员查看模块测试内容

进入系统，点击部门人员查看，观察是否正常加载员工信息。

部门人员查看模块测试内容如下表所示。

表格 12 部门人员查看模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 07 | 测试用例名称 | 部门人员查看模块 |
| 测试功能描述 | 加载员工信息 | | |
| 测试步骤 | 点击部门人员查看按钮 | | |
| 期待输出结果 | 员工信息展示成功 | | |

### 部门人员管理模块测试内容

进入系统，点击人员管理，观察是否能正常操作员工信息。

部门人员管理模块测试内容如下表所示。

表格 13 部门人员管理测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 08 | 测试用例名称 | 部门人员管理模块 |
| 测试功能描述 | 正常操作员工信息。 | | |
| 测试步骤 | 1. 点击员工管理 2. 点击删除 3. 点击修改 | | |
| 期待输出结果 | 员工信息操作无错 | | |

### 发表公告管理模块测试内容

进入发表页面，发表公告，看公告是否正常显示。

发表公告模块测试内容如下表所示。

表格 14 发表公告管理模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 09 | 测试用例名称 | 发表公告模块 |
| 测试功能描述 | 正常发表公告 | | |
| 测试步骤 | 1. 进入发表页面 2. 输入公告内容，点击发表 3. 查看发表情况 | | |
| 期待输出结果 | 正常发表公告 | | |

个人公告管理模块测试内容

进入系统，点击个人公告，查看个人公告信息显示是否正常。

个人公告管理模块测试内容如下表所示。

表格 15 个人公告管理模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 10 | 测试用例名称 | 个人公告管理模块 |
| 测试功能描述 | 加载个人发表的公告 | | |
| 测试步骤 | 进入个人公告页面 | | |
| 期待输出结果 | 公告正常显示 | | |

### 存档管理模块测试内容

进入系统，点击存档，查看历年公告显示情况。

存档管理模块测试内容如下表所示。

表格 16 存档管理模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 11 | 测试用例名称 | 存档模块 |
| 测试功能描述 | 查看存档信息 | | |
| 测试步骤 | 进入存档页面 | | |
| 期待输出结果 | 每年公告按序正常显示 | | |

### 标签管理模块测试内容

进入系统，点击标签，查看标签公告显示情况。

标签管理模块测试内容如下表所示。

表格 17标签管理模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 12 | 测试用例名称 | 标签管理模块 |
| 测试功能描述 | 查看标签公告信息 | | |
| 测试步骤 | 进入标签公告页面 | | |
| 期待输出结果 | 标签公告正常显示 | | |

### 上传文件模块测试内容

进入云盘页面，点击上传文件，查看文件是否正常上传。

上传文件管理模块测试内容如下表所示。

表格 18 上传文件模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 13 | 测试用例名称 | 上传文件模块 |
| 测试功能描述 | 上传个人文件 | | |
| 测试步骤 | 1. 进入云盘页面 2. 上传文件 | | |
| 期待输出结果 | 提示文件上传成功 | | |

下载文件模块测试内容

进入云盘页面，点击下载文件，查看文件是否正常下载。

下载文件管理模块测试内容如下表所示。

表格 19下载文件模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 14 | 测试用例名称 | 下载文件模块 |
| 测试功能描述 | 下载个人文件 | | |
| 测试步骤 | 1. 进入云盘页面 2. 点击文件下载 | | |
| 期待输出结果 | 文件下载成功 | | |

### 消息查看模块测试内容

进入消息模块，点击员工账号，查看员工之间的消息。

消息查看管理模块测试内容如下表所示。

表格 20消息查看模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 15 | 测试用例名称 | 消息查看模块 |
| 测试功能描述 | 消息查看 | | |
| 测试步骤 | 1. 进入我的消息页面 2. 点击员工账户 3. 查看员工最近消息 | | |
| 期待输出结果 | 消息显示正常 | | |

消息发表模块测试内容

进入消息模块，点击员工账号，输入消息信息，点击发送。

消息发表管理模块测试内容如下表所示。

表格 21消息发表模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 16 | 测试用例名称 | 消息发表模块 |
| 测试功能描述 | 员工之间能正常发送消息 | | |
| 测试步骤 | 1.进入消息页面  2.输入员工账号，消息  3.点击发送 | | |
| 期待输出结果 | 员工之间能正常收到消息 | | |

消息状态模块测试内容

进入消息模块，查看员工之间消息的状态显示。

消息状态管理模块测试内容如下表所示。

表格 22消息状态模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 17 | 测试用例名称 | 消息状态测试模块 |
| 测试功能描述 | 查看员工之间消息状态 | | |
| 测试步骤 | 进入消息页面 | | |
| 期待输出结果 | 1. 若是未读消息，则消息显示为红色 2. 若是已读消息，则消息显示为绿色 3. 时间最近的消息显示在最前面。 | | |

阅读统计模块测试内容

进入主页面，查看各公告阅读情况。

阅读统计管理模块测试内容如下表所示。

表格 23 阅读统计模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 18 | 测试用例名称 | 阅读统计模块 |
| 测试功能描述 | 统计阅读次数 | | |
| 测试步骤 | 进入主页面 | | |
| 期待输出结果 | 正常加载公告阅读次数 | | |

### 头像管理模块测试内容

进入个人信息页面，修改个人头像，观察系统显示情况。

头像管理模块测试内容如下表所示。

表格 24 头像管理测试模块

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 19 | 测试用例名称 | 头像管理模块 |
| 测试功能描述 | 修改头像 | | |
| 测试步骤 | 1. 进入个人信息页面 2. 点击修改头像 3. 上传头像 4. 剪裁头像 5. 显示新头像 | | |
| 期待输出结果 | 头像修改成功 | | |

### 员工审核模块测试内容

进入主页面，点击审核人员，观察待审核人员名单。

员工审核管理模块测试内容如下表所示。

表格 25 员工审核模块测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例序号 | 20 | 测试用例名称 | 员工审核模块 |
| 测试功能描述 | 查看待审核人员名单，并能修改待审核人员的部门 | | |
| 测试步骤 | 1. 进入审核页面 2. 查看待审核人员名单 3. 选择下拉列表中的部门 4. 点击修改 | | |
| 期待输出结果 | 待审核人员修改成功。 | | |

**结论**

本次所做的OA办公系统主要是采用面向对象的方式去设计，整个系统最主要的就是权限管理，当用一个账号去登陆的时候在身份验证的过程中就需要验证这个用户是属于什么级别的（普通员工、部门经理、管理层管理员），通过身份验证用户会进入不同的系统主界面。OA系统为公司内部人员使用，进入主页就可以根据自己身份的权限，开始一些基本操作。用户可以发表公告，可以查看公告，可以上传重要的文件，方便办公。也可以和员工之间进行交流，保证工作的基本交流。但本系统涉及的公司企业管理还不是很全面，只是截取了一个企业需要的基本的日常流程，功能不够丰富，系统样式固定，只能供员工完成一些简单的日常工作流程。

通过对这个系统的设计与开发，我慢慢熟悉了Nodejs+Mongodb网站开发的思路和方法。对于HTML、CSS、JavaScript、Ejs等技术有一定的理解。本系统主要完成了OA系统的分析、研究、设计与实现，基本满足了项目的要求。在完成系统的工程中我丰富了自己的知识，锻炼了自己的实践能力，为以后的工作积累了宝贵的经验。

**参考文献**

1. 泽卡斯 (Zakas. Nicholas C.) 深入浅出Node.js [D]. 人民邮电出版社,2013.
2. 坎特伦 (Mike Cantelon) (作者), Marc Harter (作者), T.J. Holowaychuk (作者), Nathan Rajlich (作者), 吴海星 (译者). Node.js实战[J]. 人民邮电出版社,2013.
3. 石丹. 基于Web服务和工作流技术的企业协同办公系统（OA）的设计与实现[D].吉林大学,2016.
4. 杜鹏，中小企业办公自动化系统的模型设计[J]. 科协论坛(下半月),2008,(10):48-49.
5. 史冠卿，黄莺. 浅析OA系统在公文管理中的应用[J]. 科技信息,2013,(22):261-263.
6. 徐珂. 关于办公自动化的发展及主流技术的探析[J]. 信息与电脑(理论版),2014,(02):65-66.
7. 李沛霖. OA网络办公管理系统的设计与实现[D].吉林大学,2011.
8. 姜丽萍,刘慧媛. OA办公系统发展现状与实践困惑[J]. 办公室业务,2016,(20):171-172.
9. 许文智. OA系统的发展历程与新一代OA系统开发方案[J]. 中国金融电脑,2008,(01):57-64.
10. 戴维·豪斯 (David Hows) (作者), Peter Membrey (作者), Eelco Plugge (作者), Tim Hawkins (作者), 周连科 (译者)，MongoDB大数据处理权威指南(第3版) [M]，清华大学出版社; 第1版，2017
11. 坎特伦 (Mike Cantelon) (作者), Marc Harter (作者), T.J. Holowaychuk (作者), Nathan Rajlich (作者), 吴海星 (译者). MongoDB权威指南(第2版) [J]. 人民邮电出版社;,2014.