[【关键词】 B/S模式 可视化 Node.js React.js RethinkDB 异步请求 III](#_Toc480546901)

[**Abstract** IV](#_Toc480546902)

[绪论 1](#_Toc480546903)

[1 在线的可视化网站设计概述 1](#_Toc480546904)

[1.1 在线的可视化网站设计的概念 1](#_Toc480546905)

[1.2 在线的可视化网站设计的发展和趋势 1](#_Toc480546906)

[1.3 可视化网站设计的必要性 1](#_Toc480546907)

[1.4 课题的背景及意义 1](#_Toc480546908)

[1.5 本文的主要工作 1](#_Toc480546909)

[2 可视化网站设计相关技术的研究 2](#_Toc480546910)

[2.1 可视化网站设计的开发模式 2](#_Toc480546911)

[2.1.1 系统结构的选择 2](#_Toc480546912)

[2.1.2 开发环境的选择 2](#_Toc480546913)

[2.2 Node.js 技术 3](#_Toc480546914)

[2.2.1 Node.js工作原理 3](#_Toc480546915)

[2.2.2 Node.js Express框架 3](#_Toc480546916)

[2.2.3 Node.js访问数据库 5](#_Toc480546917)

[2.3 React.js技术简介 6](#_Toc480546918)

[2.3.1 React.js工作原理 6](#_Toc480546919)

[2.3.2 React.js的组件化 7](#_Toc480546920)

[2.4 RethinkDB数据库简介 8](#_Toc480546921)

[3 系统分析 8](#_Toc480546922)

[3.1 用户需求分析 8](#_Toc480546923)

[3.2 系统性能分析 8](#_Toc480546924)

[3.2.1 实现办公规范化和制度化 8](#_Toc480546925)

[3.2.2 系统设计的先进性 8](#_Toc480546926)

[3.3 系统结构设计 8](#_Toc480546927)

[3.4 系统功能分析 9](#_Toc480546928)

[3.5 系统配置分析 10](#_Toc480546929)

[4 系统主要模块的实现与技术关键 10](#_Toc480546930)

[4.1 系统功能模块图 10](#_Toc480546931)

[4.2 数据库设计 10](#_Toc480546932)

[4.2.1数据库需求分析 11](#_Toc480546933)

[4.2.2 数据库逻辑结构设计 11](#_Toc480546934)

[4.2.3 数据库连接设计 13](#_Toc480546935)

[4.3 系统模块设计 13](#_Toc480546936)

[4.3.1 系统登录模块设计 13](#_Toc480546937)

[4.3.2 用户信息管理模块设计 15](#_Toc480546938)

[4.3.3 文件管理模块设计 18](#_Toc480546939)

[4.3.4 内部邮件模块设计 20](#_Toc480546940)

[4.3.5 公告管理模块设计 22](#_Toc480546941)

[4.3.5.1 添加公告 22](#_Toc480546942)

[4.3.5.2 修改和删除公告 23](#_Toc480546943)

[结论 23](#_Toc480546944)

[参考文献 25](#_Toc480546945)

在线的可视化网站设计

摘 要

随着Web应用所处的环境越来越开放，构成系统的组件也越来越复杂。使得系统的维护与演化的成本不断增加。传统的软件开发方法不能很好地满足 Web 应用软件的设计需求l,。如何在开放和动态环境下快速响应需求, 构建可扩展的 Web 应用，降低应用开发成本成为了新的挑战。

本文对在线的可视化网站设计进行了详细设计并给出实现方案。基于Node.js和React.js，设计应用开发环境的总体框架，将各个功能模块封装成React组件，插拔灵活，扩展方便。基于该系统框架，集成可视化拖拽式界面设计、预览工具，提供以方法绑定的方式给界面元素添加事件绑定。基于事件驱动机制，设计后台业务执行环境。

【关键词】 B/S模式 可视化 Node.js React.js RethinkDB 异步请求

**Online Visual Website Design**

**Abstract**

The rapid development of information technology, especially computer technology, network communication technology, security technology and automation technology provides a wider space for the development of Office Automation. Nowadays the speed-up of information technology in our country makes it necessary for both government and enterprises to apply Office Automation systems in order to improve their efficiency and speed of information transmission.

This paper analyses the context and implications of the program by giving a general overview of the definition, current situation, and trend of development and workflow technology. And it also analyzes the design scheme and implementation principles of Office Automation System in accordance with the special features of Office Management for small and medium sized enterprises, formulates a design scheme of office automation based on Web, adopts Internet and ASP technology and B/S mode, and implements the basic function modules of office automation system.

**【Key words】**office automation system B/S mode ASP

绪论

可视化WEB设计器提供了良好的解决方案. 基于可复用组件库的可视化界面建模、代码生成及实时预览的快速开发工具, 是提高 Web 应用系统开发效率与产品质量的一种重要手段, 其关键技术是 Web 组件复用方法和可视化建模设计方法. 实现 Web 组件复用的主要方法是Web 系统架构与Web 组件开发技术, 通过Web 系统架构设计把系统各模块组件化, 将Web 组件组装成一个完整的 Web 应用系统. 传统的可视化设计器侧重于仿真界面原型设计, 适用于用户需求确认,对应用开发工作帮助有限, 特别是在应用开发的前后端联调测试时实用化程度较弱.本文设计的可视化 WEB 设计器分离了界面原型设计与实际界面设计开发的过程, 其中基于浏览器的界面原型设计器支持原型设计和本地模拟数据, 用作需求分析; Eclipse 插件版设计器支持与真实业务模型服务对接, 方便开发人员调试和集成. 从四个方面对Web 应用的柔性架构及界面交互体验提供了支持: (1)借鉴标准化界面交互流程模型, 提高 Web 应用标准化水平; (2)离线设计支持和面向资源的 REST 服务支持;(3)支持单页应用程序(SPA)设计; (4)模式抽取功能, 抽取的典型场景可作为建模知识和界面模式库复用.

1 在线的可视化网站设计概述

1.1 在线的可视化网站设计的概念

可视化网站设计，它的发展历史不长，随着技术的进步和人们认识过程的深化，可视化网站设计工具也在不断地增强功能。这类工具大多是集[网页](http://baike.baidu.com/item/%E7%BD%91%E9%A1%B5)制作和管理网站于一身的所见即所得网页编辑器，可以轻而易举地制作出跨越平台限制和跨越[浏览器](http://baike.baidu.com/item/%E6%B5%8F%E8%A7%88%E5%99%A8)限制的充满动感的网页。

1.2 在线的可视化网站设计的发展和趋势

进入21 世纪以后，由于国内移动互联网的飞速发展，以及政府大力推行“互联网+”，这导致政府和企业对网站设计需求快速增长，可视化网站设计又有了新的生命和机遇。国内专门从事互联网软件开发的公司也逐渐成长起来。由于各个政府部门、公司需求不同，导致需求量急剧增加，从程序员的角度来看，日复一日的编码，未免有重复造轮子之嫌。可视化网站设计工具虽多，却少有创新的设计。

目前可视化网站设计的发展比较成熟，大名鼎鼎的，如Adobe Dreamweaver和Microsoft Visual Studio。它们的出现都大大提高了网站的开发的效率。这些工具提供的控件拖拽，数据绑定，语法检查等功能，简化了开发流程，降低了web开发的门槛。但这些产品的开发效率还可以被进一步挖掘，前有微信小程序，后有Google Android Instant Apps。

综上所述，未来网站设计将向着可视化、组件化和平台化的方向发展。

1.3 可视化网站设计的必要性

1111111111111111111111111111

1.4 课题的背景及意义

2222222222222222222222222222222222222222222222222222

1.5 本文的主要工作

本系统的设计基于B/S架构，有别于传统的IDE，采用所见即所得，所见即可用的开发模式，在本系统的平台上开发网站，省去了传统开发模式的部署步骤，可大大提高开发效率。采用组件组合式开发模式，大大减少了开发出错的概率。

文所做的主要工作有：

① 进行系统的整体分析、设计，确定了系统的B/S模式体系结构和开发工具。

② 通过比较，确定了系统的开发工具和后台数据库。

③ 进行用户需求分析，确定系统的总体功能结构。

④ 本文实现了在线的可视化网站设计的部分基本功能模块，包括：

2 可视化网站设计相关技术的研究

2.1 可视化网站设计的开发模式

2.1.1 系统结构的选择

（1）B/S模式

B/S模式的系统以服务器为核心，把所有的功能都实现在Web服务器上，并就不同的功能为开发者和用户设置相对应的权限。开发者通过调用服务器上的API和自定义组件完成网站设计，用户通过HTTP请求在权限范围内调用Web服务器上不同API完成相应的工作。B/S模式易升级，易维护，易使用，可移植性强。同时，B/S模式灵活性较差、应用模式简单。目前B/S模式是被广泛应用的系统模式。

（2）C/S模式

C/S模式系统用服务器作为数据处理和存储的核心，在客户机设计相应的应用，必须使用应用才能对数据进行操作。C/S模式专业化程度高、安全性好、开发手段灵活、运行速度快、交互性强，但开发成本高，维护复杂，升级麻烦。

（3）单体式应用

单体式应用为人熟知，IDE友好，便于共享，易于测试，容易部署。在本系统的设计中，缺点也十分明显：不够灵活，开发周期长，维护复杂，学习成本高。现有的Eclipse、Visual Studio均属于这种架构，没有必要重复造轮子。

（4）结论

本系统将开发与生产环境融为一体，具备在线协同开发，无需部署，开发完成即可投入使用。根据系统的属性确定，将在线的可视化网站设计的开发结构确定为B/S(浏览器/服务器)模式。

2.1.2 开发环境的选择

（1）Node.js

Node.js是服务端的JavaScript运行时环境，基于Google Chrome V8 JavaScript引擎。它具有无阻塞和事件驱动的特色。Node.js也实现了类似Apache和Nginx的web服务。结合React.js和Angular.js等前端框架，可以使用JavaScript开发全栈应用。

（2）ASP.NET

ASP.NET框架具有很好的扩展性和灵活性，它和Windows操作系统平台进行结合，可以与各类应用程序方便地进行数据交换。ASP.NET框架下有很多开发工具，主要是VB、 VC、ASP、C#，它一种通用的开发手段，ASP.NET框架应用广泛，但它的专业性不强，同时在规模大的应用中，稳定性表现的不够好。

（3）JAVA EE

JAVA适用于大规模数据处理，具有很强稳定性和安全性，适合开发浏览器形式的应用系统，具有很好的灵活性，在开发模式上与ASP.NET相似。

Node.js能够应付大量的数据请求，而不会造成大量的阻塞。Node.js高效利用并行I/O，解决了服务端的性能瓶颈。

鉴于Node.js的上述优点，本课题采用Node.js进行在线的可视化网站设计的软件开发。

2.2 Node.js 技术

Node.js是一个基于Google Chrome JavaScript运行时建立的平台， 用于方便地搭建响应速度快、易于扩展的网络应用。Node.js 使用事件驱动， 非阻塞I/O 模型而得以轻量和高效，非常适合在分布式设备上运行数据密集型的实时应用。

2.2.1 Node.js工作原理

（1）V8引擎解析JavaScript脚本。

（2）解析后的代码，调用Node API。

（3）[LIBUV库](https://github.com/joyent/libuv)负责Node API的执行。它将不同的任务分配给不同的线程，形成一个Event Loop（事件循环），以异步的方式将任务的执行结果返回给V8引擎。

（4）V8引擎再将结果返回给用户。

Node.js的工作原理如图2.1所示：

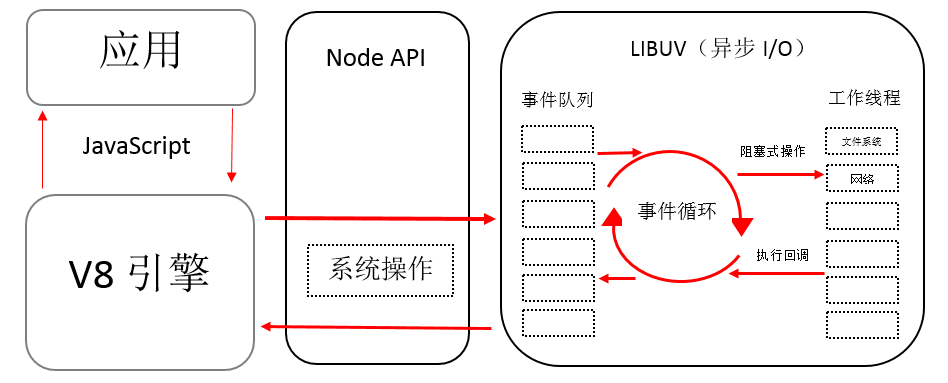


图 2.1 Node.js 工作原理

2.2.2 Node.js Express框架

Express 是一个简洁而灵活的 node.js Web应用框架, 提供了一系列强大特性帮助你创建各种 Web 应用，和丰富的 HTTP 工具。使用 Express 可以快速地搭建一个完整功能的网站。

2.2.2.1 Request对象

 Request 对象表示 HTTP 请求，包含了请求查询字符串，参数，内容，HTTP 头部等属性。常见属性如下：

(1) req.app：当callback为外部文件时，用req.app访问express的实例

(2) req.baseUrl：获取路由当前安装的URL路径

(3) req.body / req.cookies：获得「请求主体」/ Cookies

(4) req.fresh / req.stale：判断请求是否还「新鲜」

(5) req.hostname / req.ip：获取主机名和IP地址

(6) req.originalUrl：获取原始请求URL

(7) req.params：获取路由的parameters

(8) req.path：获取请求路径

(9) req.protocol：获取协议类型

(10) req.query：获取URL的查询参数串

(11) req.route：获取当前匹配的路由

(12) req.subdomains：获取子域名

(13) req.accepts()：检查可接受的请求的文档类型

(14) req.acceptsCharsets / req.acceptsEncodings / req.acceptsLanguages：返回指定字符集的第一个可接受字符编码

(15) req.get()：获取指定的HTTP请求头

(16) req.is()：判断请求头Content-Type的MIME类型

2.2.2.2 Response对象

  Response 对象表示 HTTP 响应，即在接收到请求时向客户端发送的 HTTP 响应数据。常见属性如下：

1. res.app：同req.app一样
2. res.append()：追加指定HTTP头
3. res.set()在res.append()后将重置之前设置的头
4. res.cookie(name，value [，option])：设置Cookie
5. opition: domain / expires / httpOnly / maxAge / path / secure / signed
6. res.clearCookie()：清除Cookie
7. res.download()：传送指定路径的文件
8. res.get()：返回指定的HTTP头
9. res.json()：传送JSON响应
10. res.jsonp()：传送JSONP响应
11. res.location()：只设置响应的Location HTTP头，不设置状态码或者close response
12. res.redirect()：设置响应的Location HTTP头，并且设置状态码302
13. res.send()：传送HTTP响应
14. res.sendFile(path [，options] [，fn])：传送指定路径的文件 -会自动根据文件extension设定Content-Type
15. res.set()：设置HTTP头，传入object可以一次设置多个头
16. res.status()：设置HTTP状态码
17. res.type()：设置Content-Type的MIME类型

2.2.2.3 路由

路由决定了由谁(指定脚本)去响应客户端请求。在HTTP请求中，我们可以通过路由提取出请求的URL以及GET/POST参数。下面是路由示例：

var express = require('express');

var app = express();

// 主页输出 "Hello World"

app.get('/', function (req, res) {

console.log("主页 GET 请求");

res.send('Hello GET');

})

// POST 请求

app.post('/', function (req, res) {

console.log("主页 POST 请求");

res.send('Hello POST');

})

var server = app.listen(8081, function () {

var host = server.address().address

var port = server.address().port

console.log("应用实例，访问地址为 http://%s:%s", host, port)

})

2.2.3 Node.js访问数据库

2.2.3.1 Node.js访问RethinkDB数据库的原理

当用户通过浏览器发起一个HTTP请求时，Node.js响应该请求并解析相关参数，调用RethinkDB数据库模块，通过RethinkDB数据库模块，结合相关参数。如果脚本指令中含有访问数据库的请求，则通过ODBC与后台数据库相连，ASP通过数据库存取控件使用ADO（ActiveX Data Objects）与数据库对话，并将执行结果动态生成一个HTML页面返回服务器端，以响应浏览器的请求。

2.2.3.2 Node.js访问RethinkDB数据库的主要步骤

（1）引入RethinkDB模块

Port用于指定需要连接的rethinkDB数据库的端口，host用于指定需要连接的rethinkDB数据库的IP，db用于指定需要连接的数据库的名称。

var r = require('rethinkdbdash')({

port: 28015,

host: 'localhost'

db:’myDB’

});

（2）创建数据库

r.dbCreate('myDB')

.run()

.then(function(response){

console.log(response);

})

.error(function(err){

console.log('error occured ', err);

});

（3）创建表

r.tableCreate('Employee')

.run()

.then(function(response){

console.log(response);

})

.error(function(err){

console.log('error while creating table ', err);

})

（4）在rethinkDB数据库中插入数据

r.table("Employee")

.insert({

name: "Jay",

company: "SitePoint"

})

.run()

.then(function(response){

console.log('Success ',response);

})

.error(function(err){

console.log('error occurred ',err);

})

（5）实时推送与更新数据

r.table('Employee')

.changes()

.run()

.then(function(cursor){

cursor.each(console.log);

})

.error(function(err){

console.log(err);

});

2.3 React.js技术简介

2.3.1 React.js工作原理

在Web开发中，我们总需要将变化的数据实时反应到UI上，这时就需要对DOM进行操作。而复杂或频繁的DOM操作通常是性能瓶颈产生的原因（如何进行高性能的复杂DOM操作通常是衡量一个前端开发人员技能的重要指标）。

React引入了虚拟DOM（Virtual DOM）的机制：在浏览器端用Javascript实现了一套DOM API。基于React进行开发时所有的DOM构造都是通过虚拟DOM进行，每当数据变化时，React都会重新构建整个DOM树，然后React将当前整个DOM树和上一次的DOM树进行对比，得到DOM结构的区别，然后仅仅将需要变化的部分进行实际的浏览器DOM更新。而且React能够批处理虚拟DOM的刷新，在一个事件循环（Event Loop）内的两次数据变化会被合并，例如你连续的先将节点内容从A变成B，然后又从B变成A，React会认为UI不发生任何变化，而如果通过手动控制，这种逻辑通常是极其复杂的。尽管每一次都需要构造完整的虚拟DOM树，但是因为虚拟DOM是内存数据，性能是极高的，而对实际DOM进行操作的仅仅是Diff部分，因而能达到提高性能的目的。这样，在保证性能的同时，开发者将不再需要关注某个数据的变化如何更新到一个或多个具体的DOM元素，而只需要关心在任意一个数据状态下，整个界面是如何Render的。

通常，服务器端所要做的就是根据数据Render出HTML送到浏览器端。如果这时因为用户的一个点击需要改变某个状态文字，那么也是通过刷新整个页面来完成的。服务器端并不需要知道是哪一小段HTML发生了变化，而只需要根据数据刷新整个页面。换句话说，任何UI的变化都是通过整体刷新来完成的。而React将这种开发模式以高性能的方式带到了前端，每做一点界面的更新，你都可以认为刷新了整个页面。至于如何进行局部更新以保证性能，则是React框架要完成的事情。

2.3.2 React.js的组件化

虚拟DOM(virtual-dom)不仅带来了简单的UI开发逻辑，同时也带来了组件化开发的思想，所谓组件，即封装起来的具有独立功能的UI部件。React推荐以组件的方式去重新思考UI构成，将UI上每一个功能相对独立的模块定义成组件，然后将小的组件通过组合或者嵌套的方式构成大的组件，最终完成整体UI的构建。例如，Facebook的instagram.com整站都采用了React来开发，整个页面就是一个大的组件，其中包含了嵌套的大量其它组件。如果说MVC的思想让你做到视图-数据-控制器的分离，那么组件化的思考方式则是带来了UI功能模块之间的分离。下面是一个评论框组件的示例：

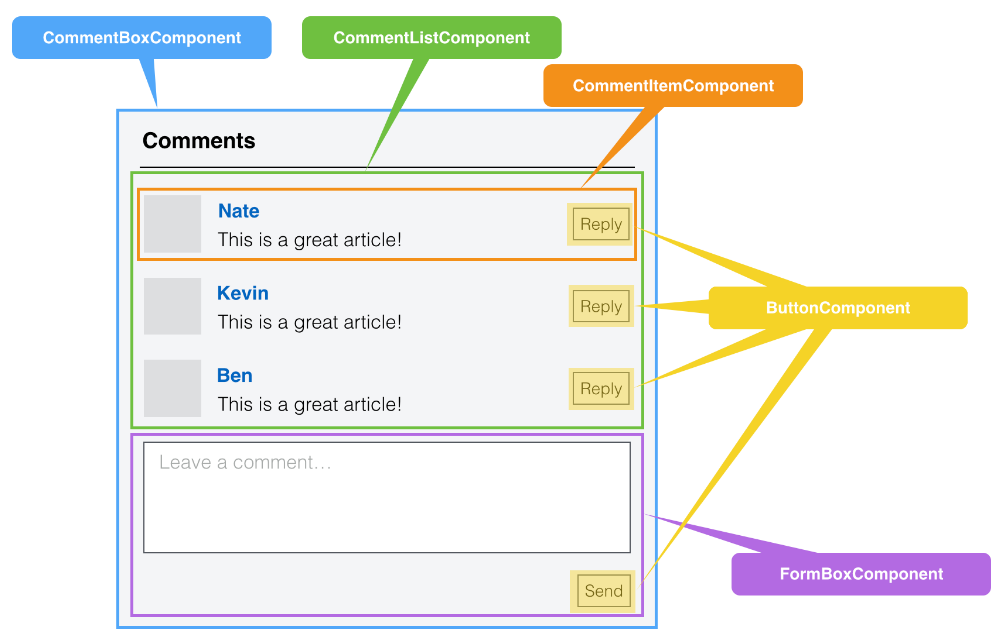


图2.2 典型的评论框组件

React认为一个组件应该具有如下特征：

（1）可组合（Composeable）：一个组件易于和其它组件一起使用，或者嵌套在另一个组件内部。如果一个组件内部创建了另一个组件，那么说父组件拥有（own）它创建的子组件，通过这个特性，一个复杂的UI可以拆分成多个简单的UI组件；

（2）可重用（Reusable）：每个组件都是具有独立功能的，它可以被使用在多个UI场景；

（3）可维护（Maintainable）：每个小的组件仅仅包含自身的逻辑，更容易被理解和维护。

2.4 RethinkDB数据库简介

RethinkDB是第一个基于实时Web构建的开源，可扩展的JSON数据库。通过暴露出令人兴奋的新访问模式，它反转了传统的数据库体系结构，而不是轮询更改，开发人员可以告诉RethinkDB不断将更新的查询结果实时地推送到应用程序。

3 系统分析

系统分析的基本思想[16]是从系统的观点出发，通过对事务进行综合分析，找出可行的方案，为系统设计提供理论依据。在系统分析中分为用户需求分析、系统性能分析、系统功能分析及系统配置分析几个步骤。

3.1 用户需求分析

用户需求分析就是对项目进行全面调研，以确定在线的可视化网站设计的功能和目的，这是构建在线的可视化网站设计的基础。

中小企业需求规模相对较小，大体可分为：一方面是通过收发电子邮件为主的方式，与其它员工交流各种信息，员工管理个人日程安排，各种图书资料的查阅等日常办公功能。另一方面则是面向公司办公业务方面的业务管理功能。即完全采用计算机技术处理办公业务，使企业内部人员能够方便快捷地共享信息、交流信息，高效地协同工作，既兼顾个人办公效率的提高，又可以实现群体协同工作。

在软、硬件方面对系统的需求，中小企业信息化由于资金实力并不雄厚。所以其对硬件的配置要求是尽可能的低，对软件配置的要求是系统最好基于日常办公软件如OFFICE系列上的应用系统，这样既不用花大量资金添置软件，员工又较容易学会使用应用系统。

3.2 系统性能分析

3.2.1 实现办公规范化和制度化

大多数中小企业由于办公自动化起步较慢，办公方式还停留在使用没有联网的计算机和MS Office系列应用软件来提高个人办公效率，没有从根本上改变过去复杂、低效的手工办公方式，远远没有实现规范化和制度化。因此，中小企业建立办公自动化系统就是要使办公方式规范化和制度化，增强企业的竞争力，使企业能够在激烈的市场竞争中取胜，不断发展壮大。

3.2.2 系统设计的先进性

办公自动化系统在开发出来时在技术上应具有一定的先进性，代表甚至是超出当时办公自动化信息系统的先进水平，这样才能使系统在建成后一段时间内，不会因技术的落后而需要大规模的调整，并且能够随着逐步的升级而保持整个系统的先进性。同时要求系统对计算机硬件要求不能太高，运行速度要快。

3.3 系统结构设计

由于中小企业一般资金短缺，为了降低开发成本和维护费用，我们采用基于B/S模式的网络结构体系，来开发本办公自动化系统。图3.1显示了典型的B/S体系结构。

B/S结构将办公自动化系统中的三要素(数据、功能、行为)分离，形成前端客户层，负责可移植的逻辑表达层；中间的应用层，允许用户通过将其与设计应用隔离而共享和控制业务逻辑；后端的设计隔离和服务层，提供对专门数据服务的访问，处理客户端与数据库间的数据流。B/S层次分离的优势体现在界面风格统一为浏览器，并具有统一的语言格式、统一的传输协议，系统管理简单，能够优化资源，可支持异种数据库和方便信息发布等。

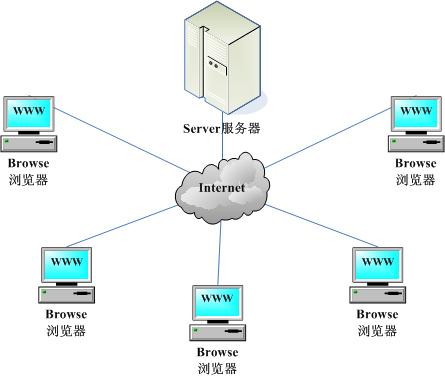


图3.1 B/S体系结构

3.4 系统功能分析

本系统是采用ASP结合SQL Server数据库平台开发的一个办公系统。本系统具有以下功能：

① 系统登录。一般系统都具有的功能，用来验证用户的合法性。

② 文件管理。这是本系统中最常用的功能，用来查看收件箱、发送文件给其他人、对文件进行答复或批复。

③ 公告发布。该栏目以一个公告板，任何权限的人都可以通过单击查看信息。查看的内容包括主题、发布人、发布时间、公告内容等。

④ 内部邮件管理。单位中各级工作人员可以通过该项功能。相互之间以信件的方式发送相关的内容，达到交流的目的。

⑤ 议题讨论管理。议题讨论包括查看议题和发表议题。用户可以通过议题讨论栏共享信息和交流心得。

⑥ 系统管理。系统管理包括公告信息、用户信息、文件信息、议题信息等方面的功能。

在系统安全操作方面需要说明的是，员工只能根据自身的权限访问和发送相关的内容。不同权限的用户有不同的功能限制：管理权限可以进行读、写、修改、删除等功能；写权限可以进行读取、写入功能；读权限只能查看，不能进行其他操作。而有些功能是不分权限的，任何用户都可以使用，如阅读公告等。

3.5 系统配置分析

本文采用浏览器/服务器B/S (Browser/Server)模式，易于维护，并使前台客户机的负载减到最少。实现该模式的开发手段采用Microsoft的ASP (ActiveX Server Pages)技术，数据库采用SQL Server。为节省设备投资，我们决定采用单个物理服务器结构，也可采用多个物理服务器结构支持大型数据事务。

系统的软硬件需求如下：

（1）服务器端

硬件配置：内存：≥512MB，CPU：Pentium4 2. 9G以上

软件配置：操作系统：Windows NT4. 0或Win2000 SERVER

安装程序：IIS (Internet Information Server)；Microsoft SQL Server7.0；

IE5. 0；分辨率800\*600

（2）客户端

硬件配置：内存：≥256MB，CPU：2.0G以上

软件配置：操作系统：Windows2000/XP

安装程序：IE5.0以上版本；分辨率800\*600

4 系统主要模块的实现与技术关键

4.1 系统功能模块图

根据需求分析和功能分析，对系统各功能进行集中、分块，按照结构化程序设计的要求，本系统的系统功能模块图如图4.1所示。



图4.1 系统功能模块图

4.2 数据库设计

本课题设计时数据库采用Microsoft SQL Server 2000，它是一种可扩展的、高性能的关系型数据库管理系统。其主要特点如下：

（1）高性能设计，可充分利用Windows的优势；

（2）系统管理先进，支持Windows图形化管理工具，支持本地和远程的系统管理和配置；

（3）强大的事务处理功能，采用各种方法保证数据的完整性；

（4）支持对称多处理器结构、存储过程、ODBC，并具有自主的SQL语言。

SQL Server以其内置的数据复制功能、强大的管理工具、与Internet的紧密集成和开放的系统结构为广大的用户、开发人员和系统集成商提供了一个出众的数据库平台。

4.2.1数据库需求分析

用户需求具体体现在各种信息的提供、保存、更新和查询上，这就要求数据库结构能充分满足各种信息的输入和输出。通过收集基本数据、数据结构以及数据处理的流程，组成一份详尽的数据词典，为后面的具体设计打下基础。

在仔细分析和调查系统的基础上，得到如图4.2所示的系统处理数据的流程图。



图4.2 系统数据流程图

4.2.2 数据库逻辑结构设计

针对无纸化网络办公系统的需求，通过对办公过程的内容和数据分析设计数据库中的表格以及各个表格之间的关系：

公告信息表（noticeinfo表）包括公告编号、公告主题、公布人、公布日期、公告内容等，其结构如表4.1所示：

表4.1 公告信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| NoticeID | Int | 4 |  | 公告编号（主键） |
| NoticeSubject | nvarchar | 50 |  | 公告主题 |
| NoticeMan | nvarchar | 50 | √ | 公布人 |
| NoticeDate | smalldatetime | 4 | √ | 公布日期 |
| NoticeContent | ntext | 16 |  | 公告内容 |

用户信息表（userinfo表）包括用户编号、用户名、密码、权限、职务、所属部门、性别、电子邮件地址、联系电话、住址等，其结构如表4.2所示：

表4.2 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| UserID | nvarchar | 50 |  | 用户编号（主键） |
| UserName | nvarchar | 50 |  | 用户名 |
| Sex | char | 10 | √ | 性别 |
| LoginID | nvarchar | 20 | √ | 登录编号 |
| Password | nvarchar | 30 |  | 密码 |
| Email | nvarchar | 50 |  | 电子邮件地址 |
| Telephone | nvarchar | 50 |  | 电话 |
| Mobile | nvarchar | 20 | √ | 手机 |
| Address | ntext | 16 | √ | 住址 |
| JoinDate | datetime | 8 | √ | 进入公司日期 |
| Online | bit | 1 |  | 是否在线 |
| Remark | text | 16 |  | 注释 |
| Department | text | 16 |  | 部门 |
| Business | nvarchar | 20 |  | 职务 |
| LeaveDate | datetime | 8 | √ | 离开日期 |

文件信息表（fileinfo表）包括文件编号、文件名、文件大小、上传人及上传时间等，其结构如表4.3所示：

表4.3 文件信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| FileID | int | 4 |  | 文件编号（主键） |
| FileName | nvarchar | 50 | √ | 文件名 |
| FileSize | nvarchar | 50 | √ | 文件大小 |
| UploadMan | nvarchar | 50 | √ | 上传人 |
| UploadDate | smalldatetime | 4 | √ | 上传时间 |
| Description | ntext | 16 | √ | 文件描述 |

回复议题表（reply表）包括回复编号、回复议题编号、回复标题、回复人、回复时间、回复内容等，其结构如表4.4所示：

表4.4 回复议题表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| ReplyID | int | 4 |  | 回复编号 |
| ReplySubID | nvarchar | 50 |  | 回复议题编号 |
| ReplyTitle | nvarchar | 50 |  | 回复标题 |
| ReplyMan | nvarchar | 50 | √ | 回复人 |
| ReplyDate | smalldatetime | 4 | √ | 回复时间 |
| ReplyContent | ntext | 16 | √ | 回复内容 |

私人信件表（message表）包括信件编号、信件主题、收信人、发信人、发信时间、信件内容、新旧等，其结构如表4.5所示：

表4.5 私人信件表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| MessageID | int | 4 |  | 信件编号 |
| MessageSubject | nvarchar | 50 |  | 信件主题 |
| Receiver | nvarchar | 50 | √ | 收信人 |
| Sender | nvarchar | 50 | √ | 发信人 |
| SendDate | smalldatetime | 4 | √ | 发信时间 |
| MessageContent | nvarchar | 50 | √ | 信件内容 |
| IsNew | bit | 1 |  | 新旧 |

此外还有系统日志表、议题讨论表等。

4.2.3 数据库连接设计

本系统的数据库连接设计定义了一个包含文件sysinc.asp，便于将来需要连接数据库时方便地调用。代码如下：

<%

strConnString="Provider=SQLOLEDB.1;Server=admin;UID=WebAdmin;PWD=12345"

set my\_Conn=Server.CreateObject("ADODB.Connection")

my\_Conn.open strConnString

set my\_rs=Server.Create.Object("ADODB.Recordset")

%>

4.3 系统模块设计

4.3.1 系统登录模块设计

系统登录页面如图4.3所示,用户需要在页面中输入用户名和密码，然后单击“登录”按钮。接下来系统会检查登录用户身份是否合法，如果合法，则进入到系统主页面。

设计系统登录模块，首先要获取登录页面中的用户名和密码这两个表单元素，并查询数据库中的用户信息表，判断该用户是否合法。当用户输入不正确或登录失败时，应给出输入错误提示或系统登录失败提示。



图4.3 系统登录页面示例

该页面的主要处理代码如下：

<% '验证用户

if session("Username")="" then

UserName=trim(request("Username"))

Password=trim(request("Password"))

sqlstr="select \* from userinfo where UserName='"&Username&"'"

my\_rs.open sqlstr,my\_Conn,1,3

if my\_rs.eof then

Response.Redirect ("login.asp?err=1")

else

if Password=my\_rs("Password") then

session("Username")=Username

session("权限")=my\_rs("Purview")

my\_rs.update

else

Response.Redirect ("login.asp?err=2")

end if

my\_rs.close

end if

end if

%>

登录到本系统后，系统自动识别用户，给用户一个权限，系统再根据该项权限提供相应的功能。例如用户消息管理、文件管理、公告管理等等。

4.3.2 用户信息管理模块设计

用户信息管理[17]模块实现的主要功能包括：添加用户、修改用户和删除用户的功能。

4.3.2.1 添加用户

首先设计一个添加新用户的表单，如图4.4所示：

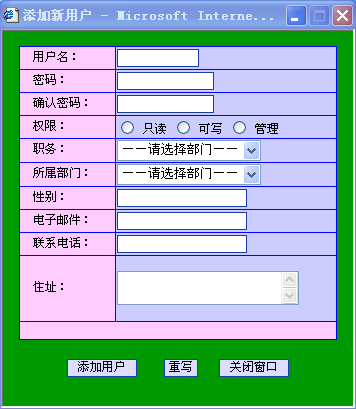


图4.4 添加新用户的表单

在“添加用户”的表单中输入相应的用户信息，单击【添加用户】按钮，向服务器提交用户信息的数据，用Request对象获取表单元素，并用SQL语句[18]插入数据库的用户信息表中。主要实现代码如下：

UserName=trim(Request("username"))

password1=trim(request("password1"))

password2=trim(request("password2"))

Purview=trim(request("purview"))

Business=trim(request("business"))

Department=trim(request("department"))

Sex=trim(request("sex"))

Email=trim(request("Email"))

Phone=trim(request("phone"))

Address=trim(request("place"))

if sex="" then

sex="未知"

end if

if Email="" then

Email="未知"

end if

if phone="" then

phone="未知"

end if

if place="" then

place="未知"

end if

'Response.Write UserName&password1&password2&Purview&Business&Department&Sex&Email&Phone&Adress

if UserName="" or password1="" or password2="" or Purview="" or Business="" or Department="" then

Response.Write "相应信息未填写完整　请确认！" & "<input type='button' value='返回' name='B2' class='buttonface' onclick='javascript：history.back()'>"

else

if password1=password2 then

sqlstr="insert into userinfo( UserName, Password, Purview, Business, Department,Sex,

Email,Phone,Address)values('"&UserName&"','"&password1&"','"&Purview&"','"&Business&"','"&Department&"','"&Sex&"','"&Email&"','"&Phone&"','"&Adress&"')"

'Response.Write sqlstr

my\_rs.open sqlstr,my\_Conn

Response.Write "用户添加成功！" & "<input type='button' value='返回' name='B2' class='buttonface' onclick='javascript：history.back()'>"

else

Response.Write "密码不一致 请确认！" & "<input type='button' value='返回' name='B2' class='buttonface' onclick='javascript：history.back()'>"

end if

end if

4.3.2.2 用户修改

使用本系统修改用户分为两类操作，第一类是登录用户修改自己的个人信息，任何用户都可以修改自己的信息。用户修改自己的个人信息后，用Request对象提交给服务器，然后用Update语句更新数据库中的个人信息。修改部分的主要代码如下：

Kusername=trim(Request("username")) '用户名

oldpassword=trim(request("oldpassword")) '旧密码

password1=trim(request("password1")) '密码

password2=trim(request("password2")) '确认密码

sex=trim(request("sex")) '性别

Email=trim(request("Email")) '电子邮件

phone=trim(request("phone")) '联系电话

place=trim(request("place")) '住址

if sex="" then

sex="未知"

end if

if Email="" then

Email="未知"

end if

if phone="" then

phone="未知"

end if

if place="" then

place="未知"

end if

sqlstr="select \* from userinfo where UserName='"&Kusername&"'"

my\_rs.open sqlstr,my\_Conn,1,1

if my\_rs("Password") <> oldpassword then

Response.Write "旧密码错　请确认！" & "<input type='button' value='返回' name='B2' class='buttonface' onclick='javascript：history.back()'>"

else

if password1=password2 then

my\_rs.close

StrSQL = "Select \* from userinfo where UserName='"&Kusername&"'"

my\_rs.Open StrSQL,my\_Conn,1,3

my\_rs("Password") = password1

my\_rs("Sex") = sex

my\_rs("Email") = Email

my\_rs("Phone") = phone

my\_rs("Address") = place

my\_rs.Update

第二类修改是系统管理员修改用户的权限，这项功能只能是拥有管理权限的用户才能执行。管理员用户修改用户信息后，用Request对象提交给服务器，用Update语句更新数据库中的用户信息表。

4.3.2.3 删除用户

若管理员要删除用户，则在用户信息管理页中单击“删除”链接，选择要删除的用户。删除用户操作的主要代码如下：

Kid=request("id")

if session("Purview")="管理" and id<> session("UserID") then

sqlstr="delete from userinfo where UserID="&Kid

my\_rs.open sqlstr,my\_Conn

my\_rs.close

set my\_rs=nothing

response.write "删除成功"

else

response.write "权限不够，与管理员联系！"

end if

4.3.3 文件管理模块设计

用户可以在某台计算机上把文件上传到服务器上，到了别的工作地点，可以把它下载下来，这样就方便了数据的流动，尤其是异地工作的用户。系统对此功能也设置了权限，一般用户无法删除和下载文件，只有文件的拥有者或管理员才可以对文件进行操作。

文件管理包括三个方面，即上传、下载和删除。

4.3.3.1 文件上传

文件上传组件比较多，常用的有Microsoft的Posting Acceptor组件、Software Artisans的SA-FileUp 组件、Persits Software的ASPUpload组件和国产的LyfUpload 组件。本系统采用了ASPUpload组件，这是一个功能很强大的COM组件。它能够实现以下许多功能，如：限制上传文件的大小和修改文件属性，还能够同时上传多个文件、设置用户的权限和将文件保存到数据库等。

实现文件上传功能的主要代码如下：

Dim objTool

Set objTool = Server.CreateObject("ASPSoft.Upload") ' 建立和初始化上传对象

strFileName = objTool.Form("file1").Properties("SourceFileName")

If Len(strFileName) > 0 Then

sqlstr = "Select \* FROM fileinfo"

my\_rs.Open sqlstr,my\_Conn,1,3

my\_rs.Addnew

filename=objTool.Form("file1").Properties("SourceFileName")

my\_rs("FileName")=filename

my\_rs("FileSize")=objTool.Form("file1").Properties("Filesize")

my\_rs("UploadMan")=session("Username")

my\_rs("Description")=objTool.Form("textarea1")

my\_rs.update

my\_rs.close

set my\_rs=nothing

ff=server.mappath("downupfile")

objTool.Form("file1").SaveToFile ff&"\", CStr(strFileName), True

Response.Write "文件上传成功！！"

else

Response.Write "没有文件上传！！"

end if

用户要实现文件上传功能，只需单击主页上的“文件上传”即可，上传文件的页面如图4.5所示：

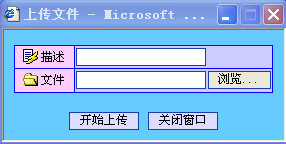


图4.5 上传文件的页面

4.3.3.2 文件下载

只有具有管理权限、写权限的用户或者上传该文件的用户才具有下载文件的权利，首先单击“系统管理”菜单中的“文件信息”选项，之后进入文件信息管理的页面中，如图4.6所示。



图4.6 文件信息管理页面

在文件信息管理页面中单击“下载”链接，之后打开下载的对话框，然后确定文件的保存地点，过一段时间，系统会提示下载成功，说明该文件被成功下载。

4.3.3.3 文件删除

把鼠标移到主页的“系统管理”上，在弹出的菜单中选择“文件信息”选项，单击“删除”链接，如图4.6所示。如果文件出现删除成功，则出现删除成功提示的页面。关闭窗口后对页面执行“刷新”操作，即可出现删除后剩余文件的信息。

实现文件删除和删除成功与否的验证程序的主要代码如下：

sqlstr="select \* from fileinfo where FileID="&request("id")

my\_rs.open sqlstr,my\_Conn,1,1

filename=my\_rs("FileName")

if session("Purview")="管理" or my\_rs("UploadMan")=session("UserName") then

my\_rs.close

set my\_rs=nothing

Set my\_rs= Server.CreateObject("ADODB.Recordset")

sqlstr="delete from fileinfo where FileID="&request("id")

my\_rs.open sqlstr,my\_Conn,1,3

response.write "成功删除文件!!"

my\_rs.close

set my\_rs=nothing

else

response.write "你没有权限删除该文件，请与管理员联系!!"

end if

4.3.4 内部邮件模块设计

各类用户均可以通过发送邮件通信的功能，相互之间以信件的方式发送相关的内容，达到交流的目的。

4.3.4.1 发送邮件

首先是设计发送邮件的表单，如图4.7所示：

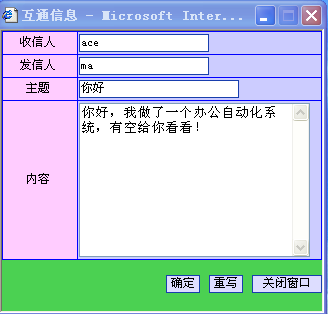


图4.7 邮件发送表单

然后要Request对象获取该表单，将邮件信息保存在“私人信件”表中。处理部分的主要代码如下：

sxr=trim(Request("sxr")) '收信人

fxr=trim(request("fxr")) '发信人

zt=trim(request("zt")) '主题

nr=trim(request("nr")) '内容

sqlstr="select \* from userinfo where UserName='"&sxr&"'"

my\_rs.open sqlstr,my\_Conn,1,3

if my\_rs.eof then

Response.Write sxr&"用户不在公司成员内，请与管理员联系！" & "<input type='button' value=' 返回' name='B2' class='buttonface' onclick='javascript：history.back()'>"

else

if zt="" then

zt="未知"

end if

if nr="" then

nr="未知"

end if

if sxr="" then

Response.Write "未填收信人　请确认！" & "<input type='button' value='返回' name='B2' class='buttonface' onclick='javascript：history.back()'>"

else

my\_rs.close

sqlstr="select \* from message"

my\_rs.open sqlstr,my\_Conn,1,3

my\_rs.addnew

my\_rs("MessageSubject")=zt

my\_rs("Receiver")=sxr

my\_rs("Sender")=fxr

my\_rs("MessageContent")=nr

my\_rs("IsNew")="Yes"

my\_rs.update

Response.Write "发信成功！"

end if

end if

4.3.4.2 接收邮件

如果收信人执行登录，进入主页后就可以发现新收到的这封信，如图4.8所示。

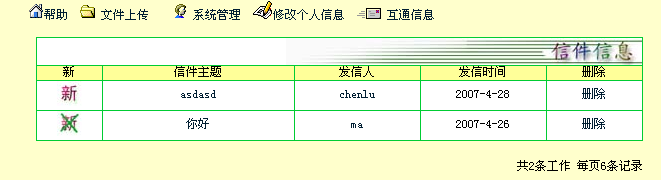


图4.8 接收新邮件页面示例

在收到的来信中，单击发信人可回信,单击信件主题可查看信件内容，如图4.9所示。

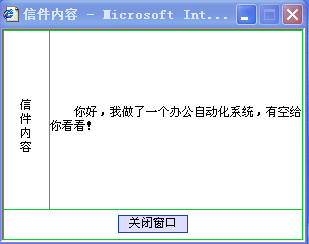


图4.9 查看信件内容页面

4.3.5 公告管理模块设计

在公告信息中，具有可写、管理权限的用户可以删除信息、修改信息、查看公告信息、添加公告信息等权限，其他用户只能查看公告信息。

4.3.5.1 添加公告

添加公告时先要设计添加公告内容的表单，如图4.10所示：

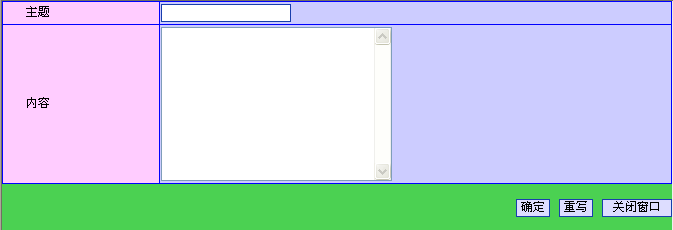


图4.10 添加公告内容的表单

当单击“确定”按钮时，将表单数据加入数据库，主要代码如下：

zhuti=trim(Request("zhuti"))

neirong=trim(Request("neirong"))

if zhuti<>"" and neirong<>"" then

sqlstr = "Select \* FROM noticeinfo"

my\_rs.Open sqlstr,my\_Conn,1,3

my\_rs.Addnew

my\_rs("NoticeSubject")=zhuti

my\_rs("NoticeMan")=session("Username")

my\_rs("NoticeContent")=neirong

my\_rs.update

my\_rs.close

esponse.write "添加公告信息成功！！"

else

response.write "公告信息不全　请确定！！"

end if

4.3.5.2 修改和删除公告

修改公告和删除公告都要验证是否具有“管理权限”，如果是则执行修改或删除操作，从公告信息表中用Delete语句删除指定的公告，Update语句更新公告信息。当用户要进行删除或修改操作时，判断用户是否具有管理权限的部分代码如下：

if Kaction="删除" then

if session("Purview")="管理" then

sqlstr="delete from noticeinfo where NoticeID="&Kid

my\_rs.open sqlstr,my\_Conn

else if Kaction="修改" then

if session("Purview ")="管理" then

sqlstr="select \* from noticeinfo where NoticeID ="&Kid

my\_rs.open sqlstr,my\_Conn

在修改公告时先获取修改公告页面表单中的公告编号、公告主题及公告内容，根据公告编号查询数据库，再更新公告的主题和内容。主要代码如下：

id=trim(request("id"))

zhuti=trim(Request("zhuti"))

neirong=trim(Request("neirong"))

if zhuti<>"" and neirong<>"" then

sqlstr = "Select \* FROM noticeinfo where NoticeID="&id

my\_rs.Open sqlstr,my\_Conn,1,3

my\_rs("NoticeSubject")=zhuti

my\_rs("NoticeContent")=neirong

my\_rs.update

my\_rs.close

如果用户的身份不是具有删除权利的管理员，在进行公告删除的时候应给出不具有删除公告权限的提示。

结论

本文提出了一个基于Web的办公自动化系统的开发方案，以B/S的模式来实现办公自动化系统和数据库后台支持。在对基于Web的办公自动化系统的研究和实际开发过程中，通过阅读各种教材、查阅大量的中英文资料、和老师同学进行探讨，对办公自动化系统相关理论和技术有了深入的了解和认识。可以说对这一课题的研究让我受益匪浅。由于时间及本人水平所限，所做的工作不可避免的存在有很多不足。最后总结全文，前段时期的工作取得的成果主要有以下几点：

① 了解并掌握了当前国内外办公自动化软件的应用情况和发展趋势及系统架设过程。增强了提出问题、分析问题和解决问题的科研能力。

② 获得了有关网络、数据库和编程方面的知识。

③ 采用动态服务器网页ASP技术，完成了办公自动化系统的分析和整体设计，包括数据库的设计，各个功能模块的划分，并最终实现了系统。

本文所设计的办公自动化系统主要具有以下几个特点：

① 本系统具有相关提示，不需要特别训练，管理员或普通用户就可以很快地掌握如何使用系统中提供的大多数功能。

② 本系统具有安全性，通过采用设置口令、设置权限等手段，保证系统中数据的完整性和安全性，防止各种非法的操作和意外的破坏。

③ 本系统完全基于浏览器的Web方式为使用者提供了方便，客户端只需安装大众的IE浏览器即可。

④ 本系统维护在服务器端执行，实现客户端零维护，并可随时实现异地管理，远程办公，避免了C/S模式麻烦的系统维护过程。

⑤ 本系统合理利用各类办公信息，提高企业的工作效率、工作质量及事务管理水平，增强信息资源管理、指导及信息导向的能力，以适应信息化社会的发展。

参考文献

[1] 王波，桑军等．办公自动化系统设计[M]．北京：人民邮电出版社，2003．

[2] 张锐昕．办公自动化概论[M]．北京：清华大学出版社，2004．

[3] 姚卿达．办公自动化的发展与实现技术[J]．现代电子计算机，2002，17(9)：83～921

[4] 张有生，陈松乔．C/S与B/S混合软件体系结构模型[J]．计算机应用，2002，2(3)：138～140

[5] 杨振山．中国办公自动化及解决方案[J]．电子计算机与外部设备，2000，9(2)：38～41

[6] 王健南．ASP．NET网络办公项目方案精解[M]．北京：航空工业出版社，2006．

[7] 陈娴．ASP信息化系统建设案例[M]．北京：人民邮电版社，2006．

[8] 马照亭，郭月强，焦祝军．ASP Web编程实例教程[M]．北京：北京希望电子出版社，2002．

[9] 邓文渊，陈惠贞，陈俊荣．ASP与网络数据库技术[M]．北京：中国铁道出版社，2003．

[10] 刘瑞新．ASP动态网站开发[M]．北京：机械工业出版社，2005．

[11] 丁贵广等．ASP 动态网站建站实例与技巧：ASP与Web数据库结合[M]．西安：西安电子科技大学出版社，2001．

[12] 李存斌，樊建平．ASP高级编程及其项目应用开发[M]．重庆：中国水利水电出版社，2003．

[13] 余波．动态Web应用高级开发指南[M]．北京：人民邮电出版社，1999．

[14] 范玉顺．工作流管理技术基础[M]．北京：清华大学出版社，2001．

[15] Zhuge H, Cheung TY, Pung HK.A timed workflow process model [J]. Journal of System and Software, 2001, 55(3): 231~243

[16] Roger S. Software engineer: A Practioner’s Approach [M]. Fifth Edition. New York: McGraw Hill Company, 2002.

[17] McLeod, Raymond. Management Information Systems [M]. Third Edition. Chicage: Science Research Associates, Inc. 1986.

[18] Ryan K.Stephens, Ronald R. Plew. Sams Teach Yourself SQL In21 Days [M]. Third Edition. New Jersey: The MeGraw Hill Comapany. 2005.

致 谢

在论文即将完成之际，向所有给我支持、帮助和指导的人表示衷心的感谢。尤其要感谢我的论文指导老师陈端荣副教授，他在学习方面给了我很多帮助，在毕业设计上为我们作了认真的指导，使我从中学到了很多知识。他严谨求实的治学态度、高度的敬业精神、兢兢业业、孜孜以求的工作作风和大胆创新的进取精神对我产生了重要影响。他渊博的知识、开阔的视野和敏锐的思维给了我深深的启迪。在此祝愿他身体健康，全家幸福！

其次我要特别感谢身边的同学对我论文写作的指导，在系统设计过程中感谢他们提供的资料及建议，他们为我完成这篇论文提供了巨大的帮助。还要感谢大学四年来的所有老师，为我们打下了软件开发专业知识的基础。

最后，感谢我的母校湖北大学。通过大学阶段的学习，提高了我们适应社会和自我发展的能力，为我们提供了一个很好的知识平台。