

# 聊天室系统概要设计

V0.1

学院： 计算机科学与工程学院

项目： 聊天室系统

—

评 审 日 期： 2014 年 11 月 17 日

# 聊天室系统概要设计

<b>1. 导言</b>	<b>1</b>
1.1 目的	1
1.2 范围	1
1.3 缩写说明	1
1.4 术语定义	2
1.5 引用标准	2
1.6 参考资料	2
1.7 版本更新信息	2
<b>2. 系统分析</b>	<b>2</b>
<b>3. 界面设计</b>	<b>4</b>
<b>4. 体系结构</b>	<b>7</b>
4.1 体系结构	7
4.1.1 浏览器/服务器模型体系结构	7
4.1.2 系统体系结构	8
4.2 系统运行环境	9
4.2.1 网络结构图	9
4.2.2 硬件环境	9
4.2.3 软件环境	10
<b>5. 数据模型</b>	<b>10</b>
5.1 数据库的概念结构模型设计	10
5.2 数据库的逻辑结构模型设计	11
5.3 数据库接口程序设计	13
5.3.1 ConfigManager 类	13
5.3.2 StoredProcedure 类	13
5.3.3 DAO 类	13
5.3.4 ExecuteSql 类	13
<b>6. 模块设计</b>	<b>14</b>
6.1 用户登录和管理员登录	15
6.2 用户注册	17
6.3 发送聊天内容	18
6.4 显示聊天内容	19
6.5 查看用户日志	19
6.6 用户管理	20

# 1. 导言

## 1.1 目的

该文档的目的是描述网上招聘系统项目的概要设计，其主要内容包括：

- 系统功能简介
- 系统结构设计
- 系统接口设计
- 数据设计
- 模块设计
- 界面设计

本文档的预期的读者是：

- 开发人员
- 项目管理人员
- 测试人员

## 1.2 范围

该文档定义了系统的结构和单元接口，但未确定单元的实现方法，这部分内容将在详细设计/实现中确定。

## 1.3 缩写说明

*XML*（Extensible Markup Language）可扩展标记语言。

*ADO.NET*（ActiveX Data Objects）是一个 **COM** 组件库。

*LINQ*（Language Integrated Query）语言集成查询是一组用于 [c#](#)和 [Visual Basic](#) 语言的扩展。

*ASP.NET*（Active Server Page .NET）是.NET Framework 的一部分，是一项[微软公司](#)的技术，是一种使嵌入网页中的[脚本](#)可由因特网服务器执行的服务器端脚本技术。

## 1.4 术语定义

ASP.NET 是 .NET Framework 的一部分，是一项[微软公司](#)的技术，是一种使嵌入网页中的[脚本](#)可由因特网服务器执行的服务器端脚本技术，它可以在通过 HTTP 请求文档时再在 Web 服务器上动态创建它们。指 Active Server Pages（动态服务器页面），运行于 IIS（Internet Information Server 服务，是 Windows 开发的 Web 服务器）之中的程序。

## 1.5 引用标准

## 1.6 参考资料

- [1] 《软件工程——理论与实践（第 2 版）》  
高等教育出版社
- [2] 《Oracle 11g 管理与编程基础》  
人民邮电出版社
- [3] 《Web 程序设计——ASP.NET 实用网站开发》  
清华大学出版社

## 1.7 版本更新信息

本文档的更新记录如表 B-1 所示。

表 B-1 版本更新记录

修改编号	修改日期	修改后版本	修改位置	修改内容概述
000	2014. 10. 20	0. 1	全部	初始发布版本

# 2. 系统分析

聊天室系统可以分为两个主要的组成部分，一个是用户端子系统，一个是管理员子系统。用户端子系统主要是进行一些基本功能，如用户登录/注册、发送信息、显示信息。管理员端子系统与用户管理和用户日志密切相关。

图 B-1 说明了用户与管理员之间的关系，图 B-2 说明了该系统主要分成哪几个子系统，图 B-3 和 B-4 为用户端和管理员端的组成结构图。

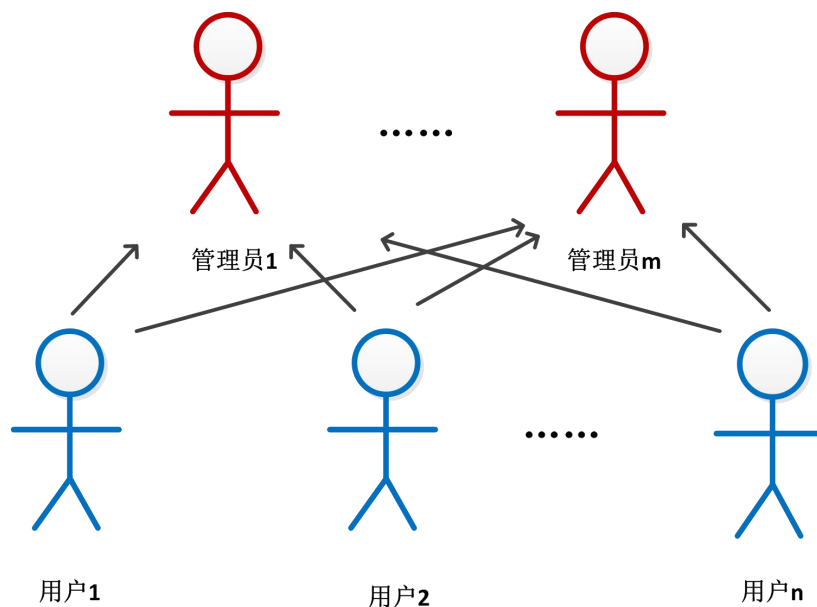


图 B-1：用户与管理员关系图

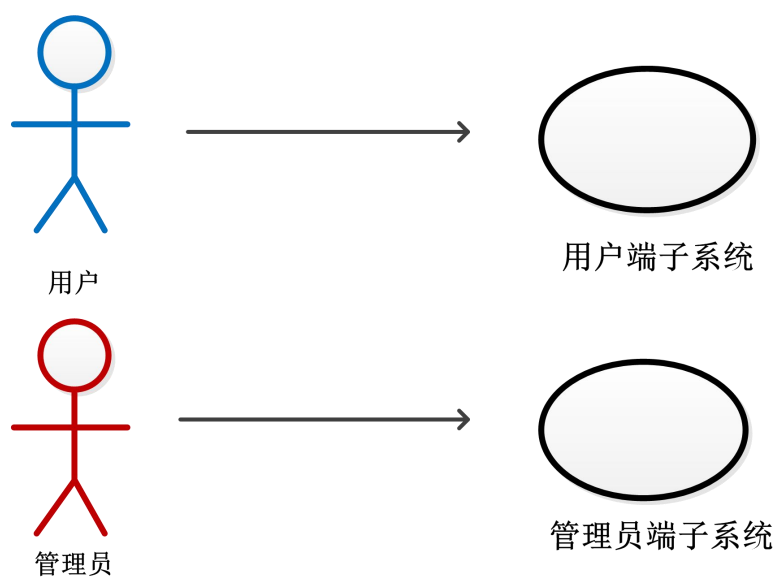


图 B-2：系统的主 Use Case 图

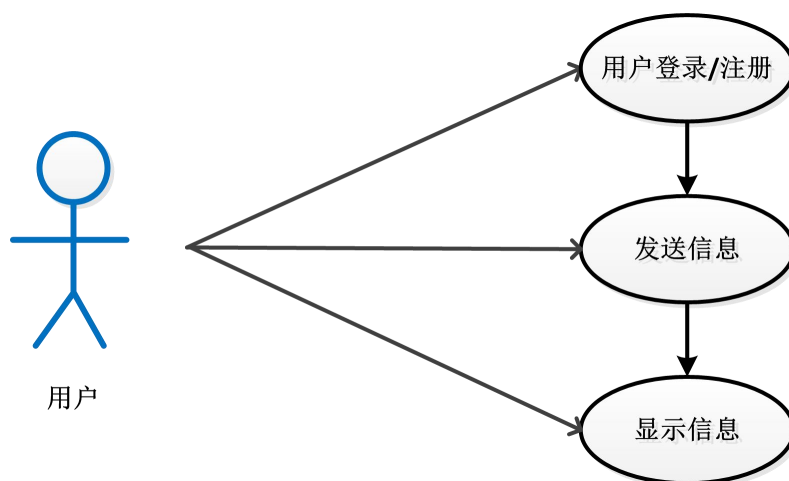


图 B-3：用户端子系统图示

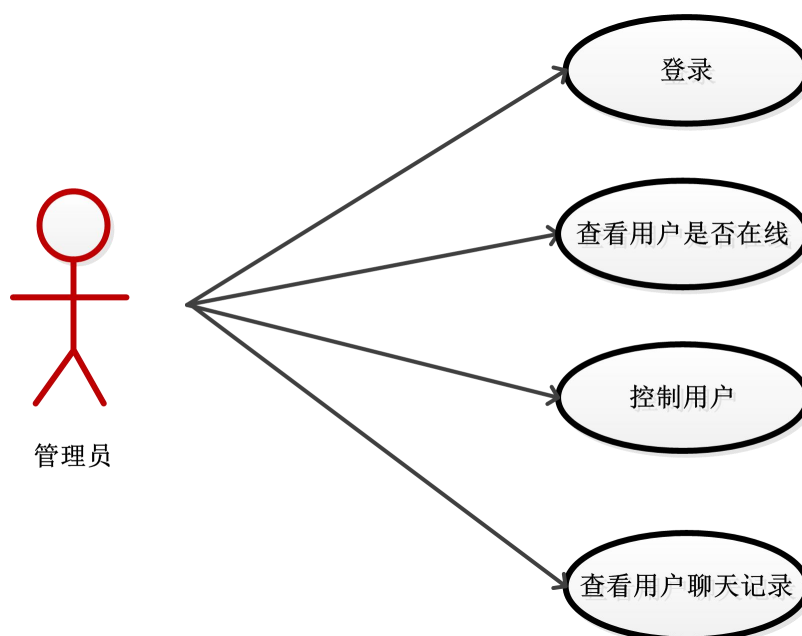


图 B-4：管理员端子系统

### 3.界面设计

聊天室系统的框架包括四个主要页面。

其一是登录界面中，界面上有两个文本框，分别是用户名框和密码框，有两个按钮：一个是提交钮，用来提交用户登录信息到服务器，另一个是注册钮，用来链接到用户注册信息页面。

## 请输入用户名

name:

password:

## 请输入用户名

name:

password:

## 用户名或密码不正确！请注册！

name:

password:

图 B—5：登录界面初步设计

其二是注册界面，有六个文本框，分别是用户名、密码、电子邮件、姓名、年龄、性别，下方会有一个注册键。想要注册的用户按要求填入相应的个人信息，其中用户名和密码不能为空，若为空会提示“用户名不能为空”和“密码不能为空”，若满足要求即可成功注册。

### 注册账号

用户名

密码

电子邮箱

姓名

年龄

性别

图 B—6：注册界面初步设计

其三是聊天室主界面，提供用户聊天的环境，聊天室主界面分为上下两部分，上面部分用来显示聊天的信息，下面部分用来让想要聊天的用户发送信息。当然想要在该聊天室聊天的用户必须先登录，单击“登录”链接转到登录界面登录。

你还没有登录，请先[登录](#)

```
welcom asdf
asdfsay:
asdfsay:fdsa
welcom zmy
zmysay:
zmysay:我来了
asdfsay:查吧, 应该有了
```

消息:

图 B-7: 聊天室界面初步设计

其四是管理员界面，管理员有自己的登录页面、查看聊天日志和用户管理界面。因为管理员身份是系统实现设置，因此登录界面下方没有注册选择。

请输入用户名

name:

password:

图 B-8: 管理员登录界面初步设计

查看聊天日志

姓名

查看聊天日志

姓名

ID	NAME	MESSAGE	CHATTIME	IPADDRESS
4	zmy		2014/11/14 22:45:27	10.154.62.27
5	zmy	我来了	2014/11/14 22:45:35	10.154.62.27

图 B-9: 管理员查看聊天日志界面初步设计



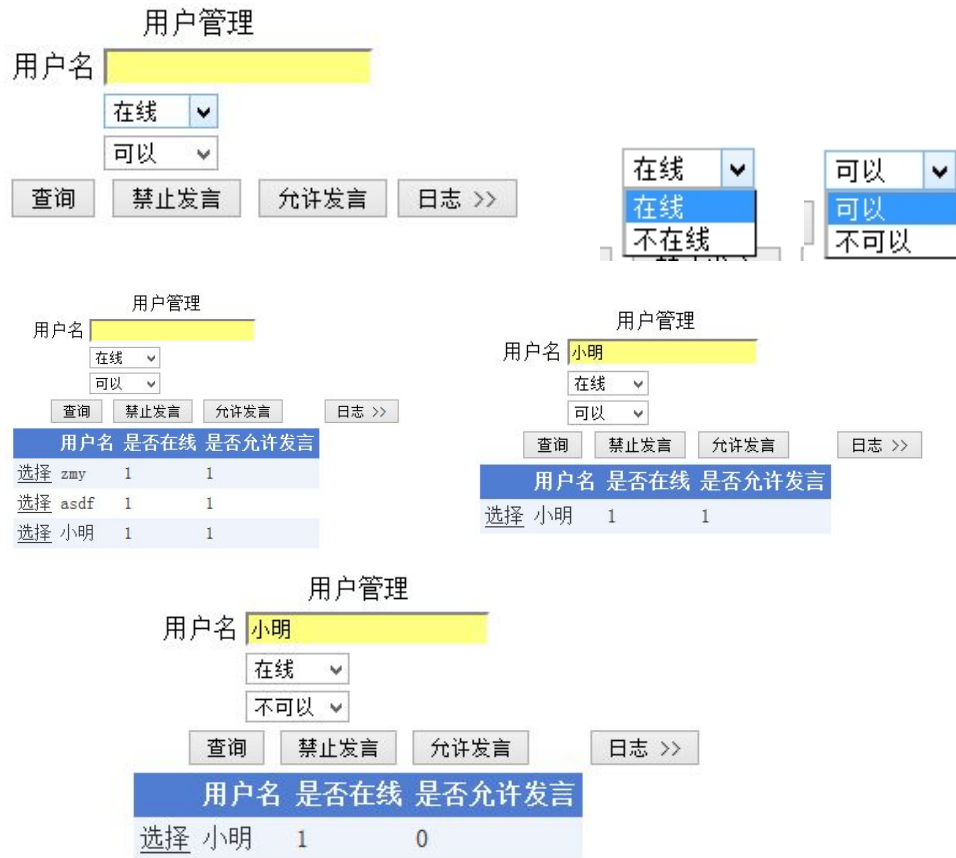


图 B-10：管理员用户管理界面初步设计（组图）

## 4. 体系结构

### 4.1 体系结构

分布型结构：浏览器/服务器（B/S）模型

#### 4.1.1 浏览器/服务器模型体系结构

基于 Web 的 B/S 风格其实也是一种客户机/服务器风格，只不过它的客户端是浏览器。为了区别传统的 C/S 风格，才特意称其为 B/S 风格。它实际上是一种三层应用结构的一种实现方式，即：浏览器/Web 服务器/数据库服务器。

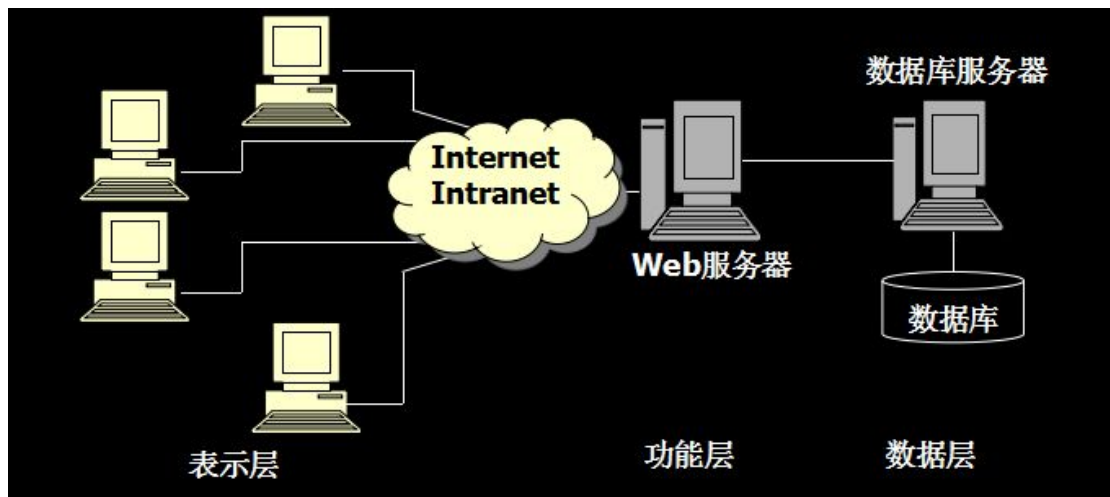


图 B-11: 浏览器/服务器模型

### 4.1.2 系统体系结构

根据系统分析结果，聊天室系统系统从结构上应满足：

- 基于浏览器进行显示以方便用户使用；
- 采用 ASP.NET 技术与数据库通讯以便于数据库的转换；

表示层是用户界面或者管理员界面； 功能层是相当于三层标准架构中的 Web 应用服务层，除了响应用户和管理员的请求，并且进行逻辑处理，处理的结果反馈给客户或者发送到数据库中。涉及到如下图所示的功能；数据库层即是数据库服务器。

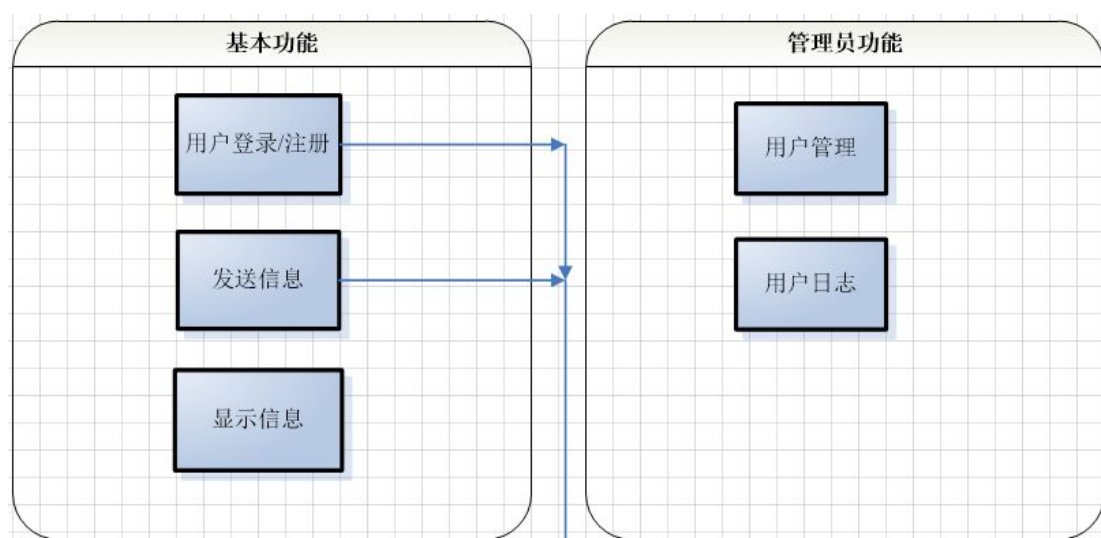


图 B-12: 功能层涉及到的功能

## 4.2 系统运行环境

系统运行的网络结构图、硬件软件环境图如下。

### 4.2.1 网络结构图

本系统的网络拓构图如图 B — 8：

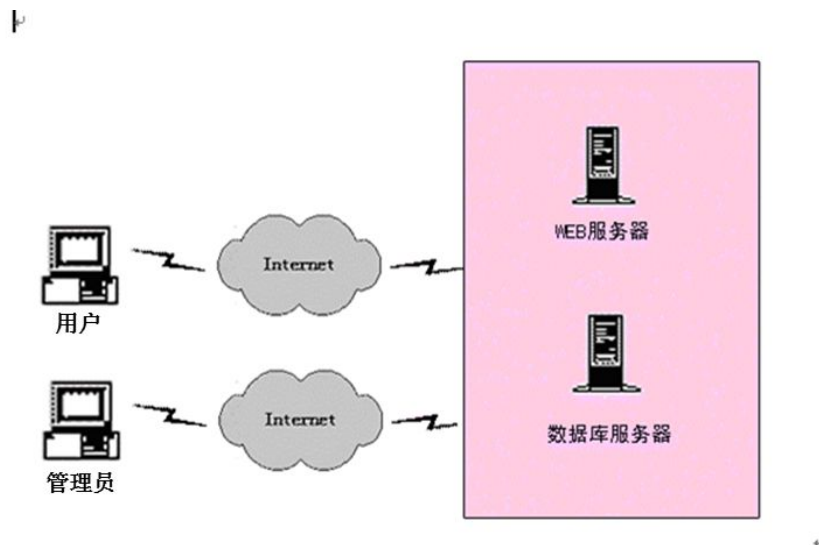


图 B — 13：网络拓扑图

### 4.2.2 硬件环境

本系统的硬件环境如下：

- 客户机：普通 PC
  - CPU：P4 1.8GHz 以上
  - 内存：256MB 以上
  - 能够运行 IE5.0 以上或者 Netscape4.0 以上版本的机器
  - 分辨率：推荐使用 1024\*768 像素
- WEB 服务器
  - CPU：P4 2.0GHz
  - 内存：1G 以上
  - 硬盘：80G 以上
  - 网卡：千兆

- 数据库服务器
  - CPU: P4 2.0GHz
  - 内存: 1G 以上
  - 硬盘: 80G 以上

### 4.2.3 软件环境

- 操作系统: Windows 7 或者 Windows 8
- 数据库: Oracle11g
- 开发工具: Visual Studio 2010
- 浏览器: IE10 (Internet Explorer 10)
- 服务器: ASP.NET

## 5.数据模型

本系统的数据模型主要是进行数据库的设计。

### 5.1 数据库的概念结构模型设计

概念设计以反映现实世界中的实体、属性和它们之间的关系等的原始数据形式,建立数据库的每一幅用户视图。图 B-14 是系统 E-R 图。

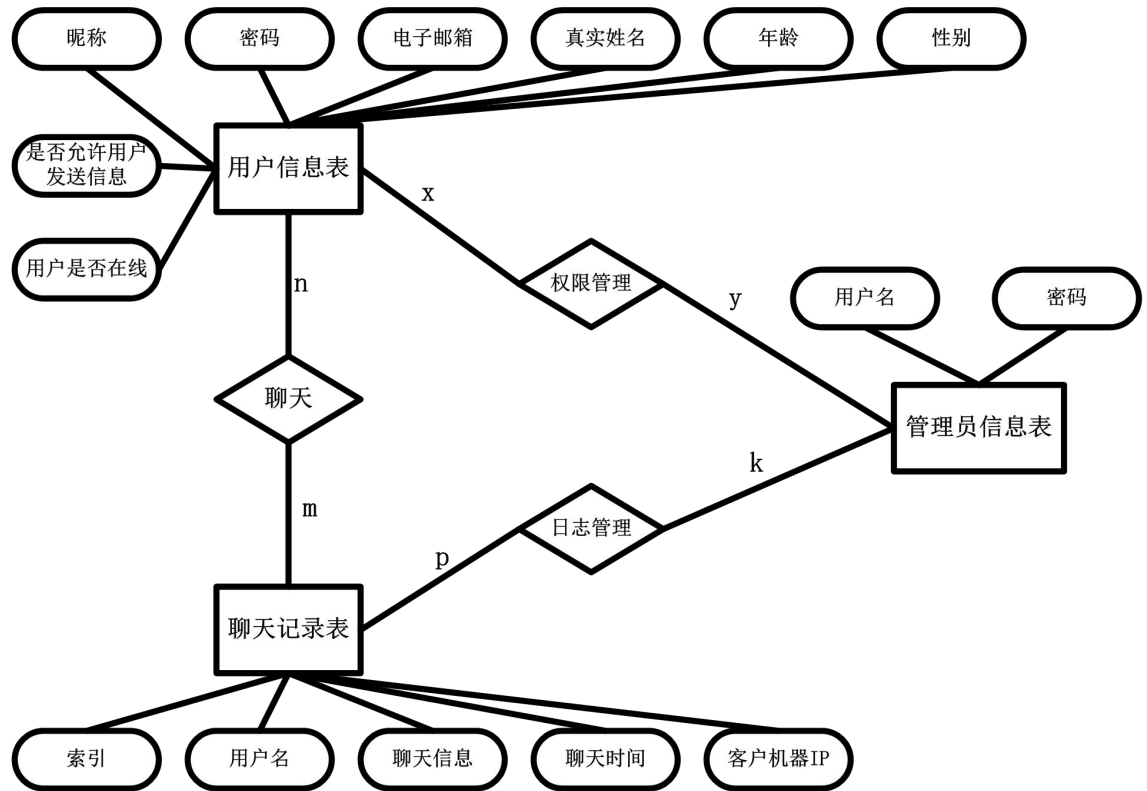


图 B-14: 系统的 E-R 图

## 5.2 数据库的逻辑结构模型设计

- 数据库的逻辑设计是将各局部的 E-R 图进行分解、合并后重新组织起来形成数据库全局逻辑结构，包括所确定的关键字和属性、重新确定的记录结构、所建立的各个数据之间的相互关系。

表 C-1: UseInformation (用户信息表) 表

字段名	字段代码	字段类型	字段大小	关键字	可否为空	代码字典表
用户名	name	VARVARCHAR	50	Y	N	
密码	password	VARVARCHAR	50		N	
电子邮箱	email	VARVARCHAR	50			
真实姓名	realname	VARVARCHAR	50			
年龄	age	VARVARCHAR	10			
性别	sex	VARVARCHAR	10			
用户是否在线	IsOnline	VARVARCHAR	1			
是否允许用户发送信息	IsAllowMessage	VARVARCHAR	1			

表 C-2: Admin (管理员信息表) 表

字段名	字段代码	字段类型	字段大小	关键字	可否为空	代码字典表
用户名	name	VARVARCHAR	50	Y	N	
密码	password	VARVARCHAR	50		N	

表 C-3: ChatRoomLog (聊天记录表) 表

字段名	字段代码	字段类型	字段大小	关键字	可否为空	代码字典表
索引	id	number	20	Y	N	
用户名	name	VARVARCHAR	50	Y	N	
聊天信息	message	VARVARCHAR	50		N	
聊天时间	chattime	VARVARCHAR	50	Y	N	
客户机器 IP	ipaddress	VARVARCHAR	10	Y	N	

/\*

创建用户信息表脚本

\*/

```
create table userinformation(
    name varchar(50),
    password varchar(50),
    email varchar(50),
    realname varchar(50),
    age varchar(10),
    sex varchar(10),
    isonline varchar(1),
    isallowmessage varchar(1)
);
```

/\*

创建管理员信息表脚本

\*/

```
create table admin(
    name varchar(50),
    password varchar(50)
);
```

/\*

创建聊天记录表脚本

\*/

```
create table chatroomlog(
    id number(20),
    name varchar(50),
    message varchar(50),
```

```
    chattime varchar(50),  
    ipaddress varchar(10)  
);
```

### 5.3 数据库接口程序设计

为了方便访问数据库，需要开发名称为 DAL 的类空间来提供访问 SQL 语句的接口。DAL 空间包括以下几个类：

- ConfigManager 类，提供读取 Web.config 文件中数据库连接字符串的功能。
- StoredProcedure 类，封装数据库访问过程。
- DAO 类，执行数据库访问应用程序的基类。
- ExecuteSql 类，提供执行 SQL 语句的方法。

#### 5.3.1 ConfigManager 类

ConfigManager 类，提供读取 Web.config 文件中数据库连接字符串的功能，包含一个公开的属性 DALConnectionString，通过该属性应用程序可以获得数据库连接字符串，包含一个私有 dalConnectionString，以及一个构造函数的定义，并在构造函数里实现配置文件的读取。

#### 5.3.2 StoredProcedure 类

StoredProcedure 类，封装数据库访问过程。该类提供 3 个公开函数，分别是 Dispose()，用来释放数据库访问过程中占有的资源；Run()，执行不返回数据的 SQL 命令，Run(DataTable dataTable)是 Run()的一个重载函数，执行返回数据的 SQL 命令。公开变量 ErrorMessage 来存放数据访问过程中出现的错误信息。

StoredProcedure 类被定义为衍生类，且是内置的，并为命名空间外的应用程序调用。

#### 5.3.3 DAO 类

DAO 类，执行数据库访问应用程序的基类。该类只包含一个内置的 StoredProcedure 类的对象 sproc 的定义。

#### 5.3.4 ExecuteSql 类

ExecuteSql 类，提供执行 SQL 语句的方法。该类提供一个公共的方法 run，该方法有两个重载版本，其一为 run(string sqlText)，包含一个参数接受 SQL 命令的文本，执行不返回数据的 SQL 的命令；另外一个 run(DataTable table, string sqlText)，参数 table 用来存储执行 SQL 命令后返回的数据，参数 sqlText 接受 SQL 命令的文本。此外该类还提供一个公开

变量 ErrorMessage 用来存储 SQL 命令执行过程出现的错误信息。

该类继承于数据库访问基类 DAO。

## 6.模块设计

按照功能分解，本系统分为用户端和管理员端。

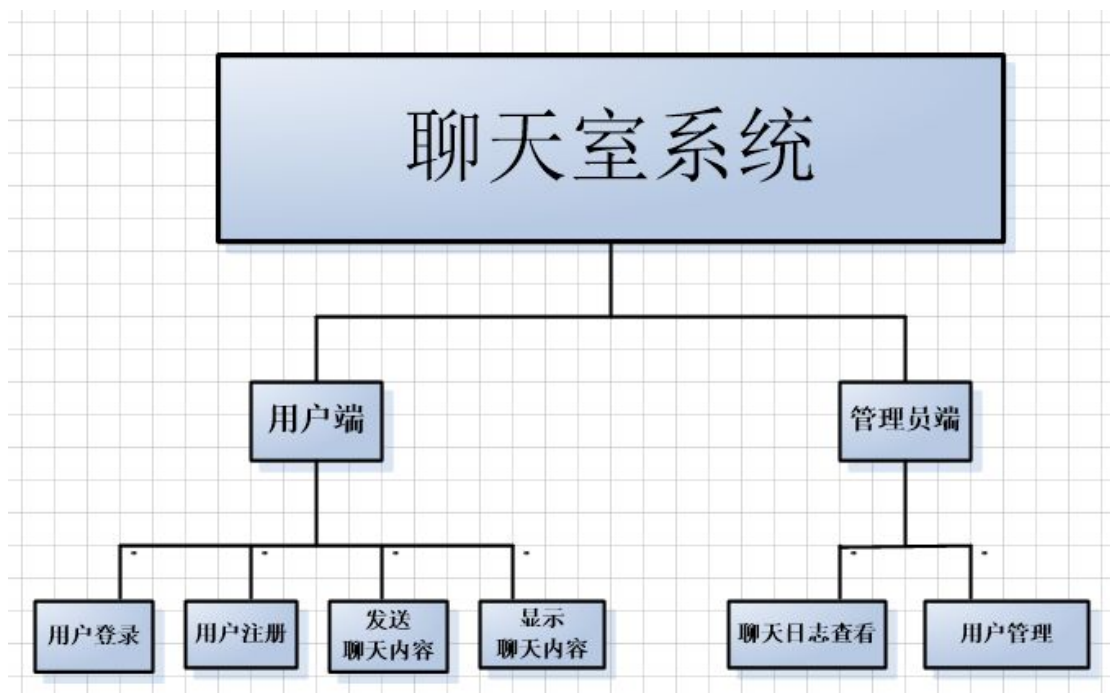


图 B-15: 模块分析图



## 6.1 用户登录和管理员登录

### ● Default.aspx

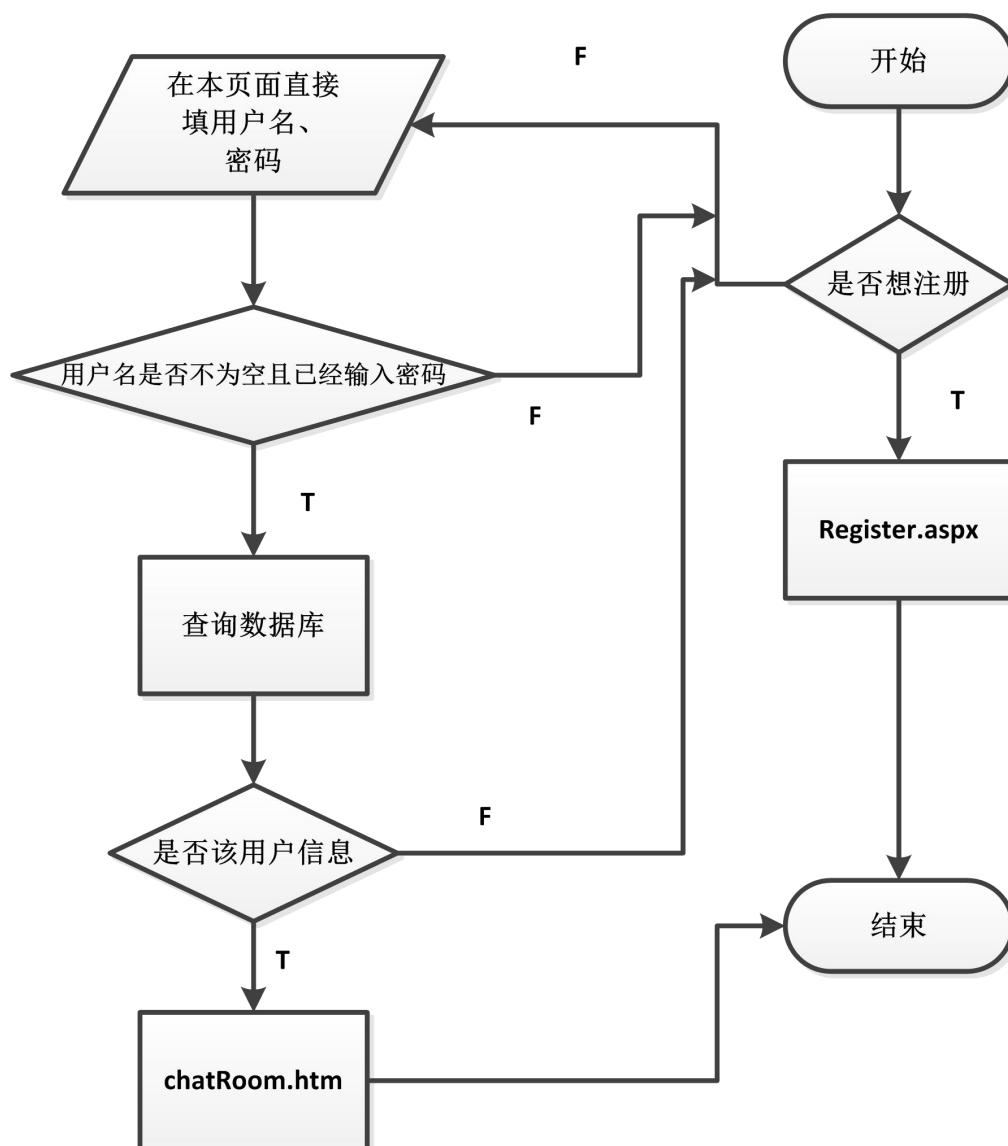


图 B-16: 用户登录流程图

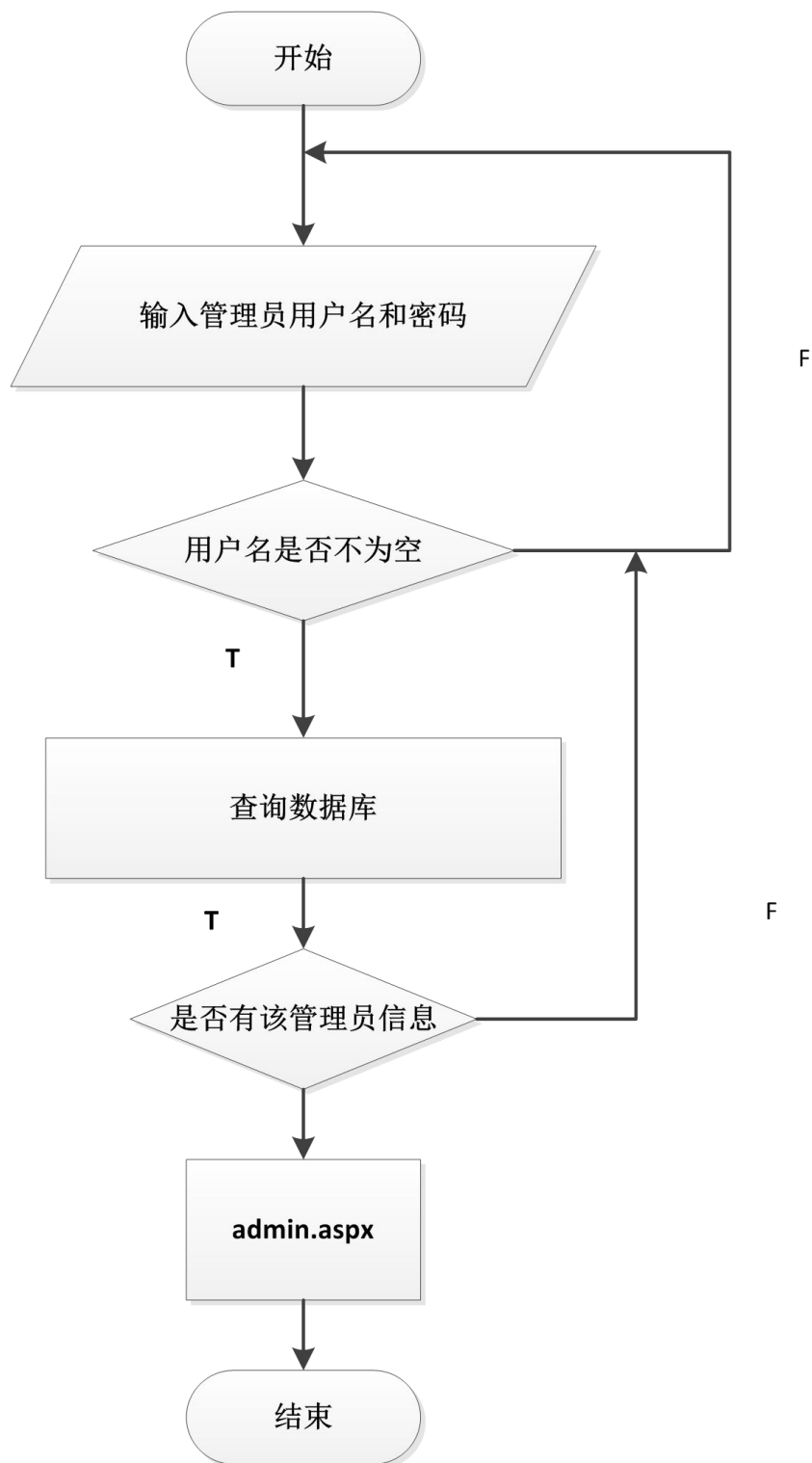


图 B —17：管理员登录流程图

## 6.2 用户注册

### ● Register.aspx

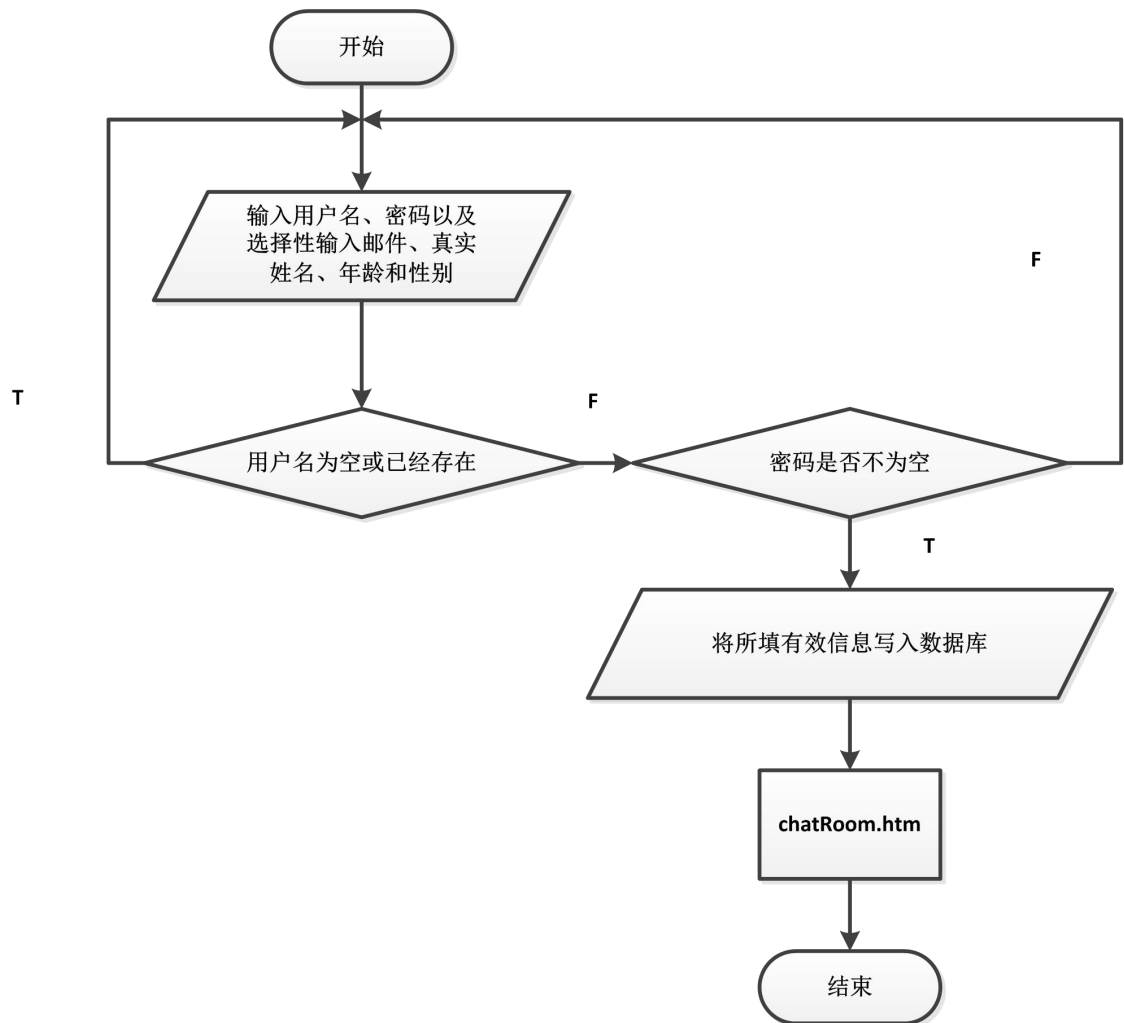


图 B—18: 用户注册流程图

## 6.3 发送聊天内容

### ● chatRoom.htm

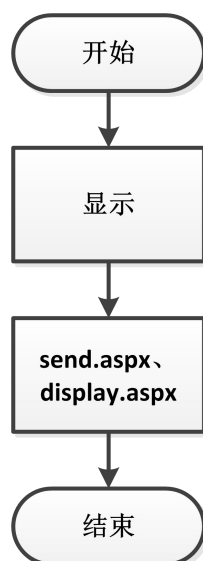


图 B-19: 聊天主页面流程图

### ● send.aspx

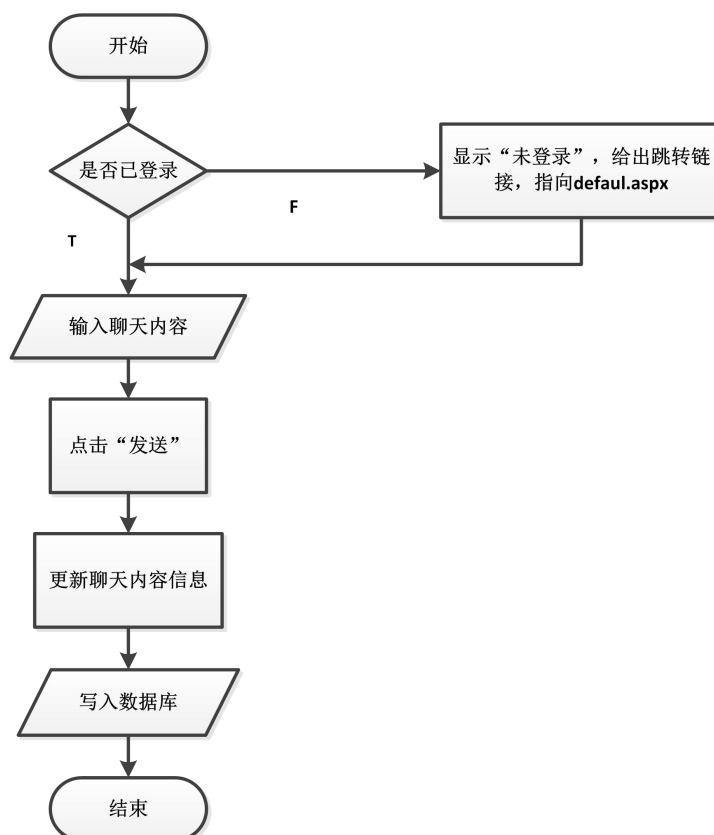


图 B-20: 发送聊天内容流程图

## 6.4 显示聊天内容

- display.aspx

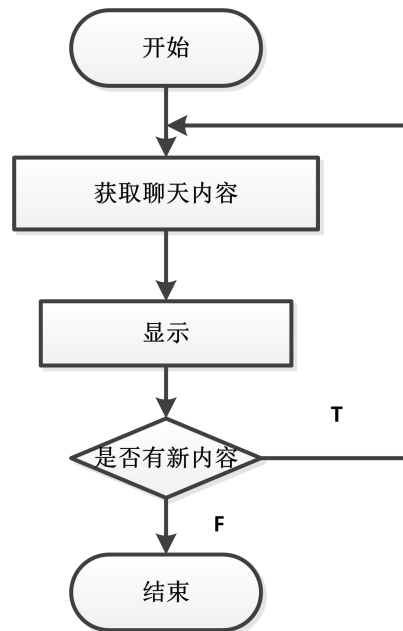


图 B-21：发送聊天内容流程图

## 6.5 查看用户日志

- Admin.aspx

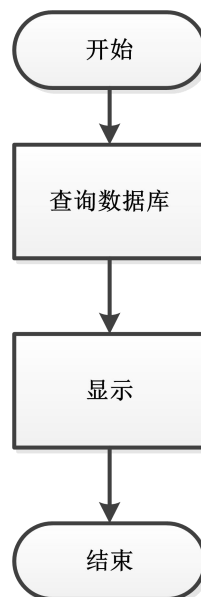


图 B-22：查看用户日志流程图

## 6.6 用户管理

- usercontrol.aspx

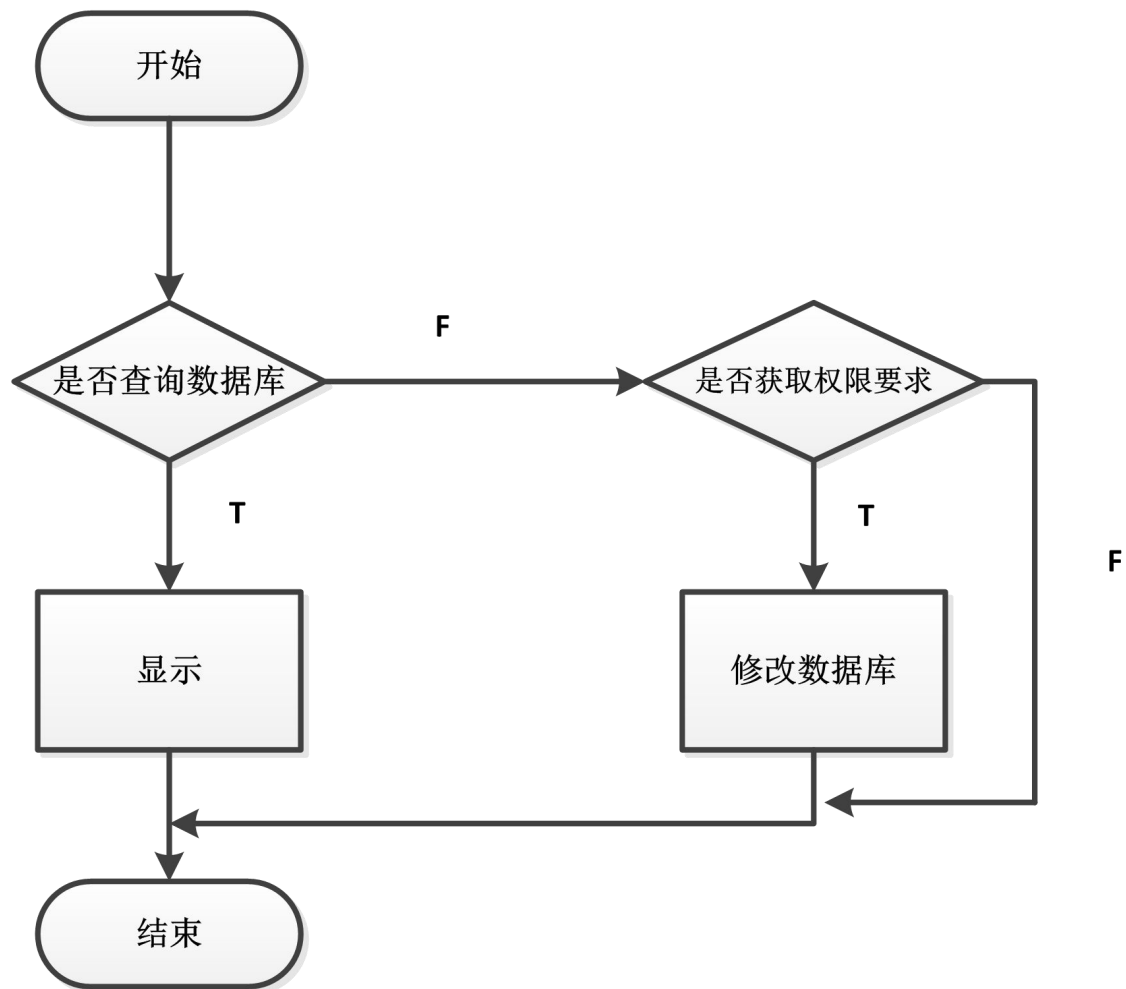


图 B—23：用户管理流程图