

聊天室系统需求规格

V0.2

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 10 月 15 日

聊天室系统需求规格.....	1
1 引言.....	3
1.1 目的.....	3
1.2 范围.....	3
1.3 缩写说明.....	3
1.4 术语定义.....	4
1.5 引用标准.....	5
1.6 参考资料.....	5
1.7 版本更新信息.....	5
2 系统定义.....	5
2.1 项目来源及背景.....	5
2.2 项目要达到的目标.....	5
2.3 系统整体结构.....	6
3 应用环境.....	7
3.1 系统运行网络环境.....	7
3.2 系统运行硬件环境.....	8
3.3 系统运行软件环境.....	8
4 功能规格.....	9
4.1 结构化分析方法.....	9
4.1.1 数据流图之顶层.....	9
4.1.2 数据流图之中间层.....	10
4.1.3 数据流图之底层.....	10
4.1.4 数据词典.....	12
4.2 E-R 图分析.....	13
4.3 面向对象分析方法.....	14
4.3.1 角色 (Actor) 定义.....	14
4.3.2 系统主 Use Case 图.....	15
4.4 用户端子系统.....	17
4.4.1 用户登录.....	18
4.4.2 用户注册.....	18
4.4.3 发送聊天信息.....	18
4.4.4 显示聊天信息.....	19
4.5 管理员端子系统.....	19
4.5.1 管理员登录.....	20
4.5.2 查看用户是否在线.....	20
4.5.3 控制用户.....	21
4.5.4 查看用户聊天记录.....	21
5 性能需求.....	21

5.1	界面需求.....	21
5.2	响应时间需求.....	22
5.3	可靠性需求.....	22
5.4	开放性需求.....	22
5.5	可扩展性需求.....	22
5.6	系统安全性需求.....	22
6	产品提交.....	22
7	实现约束.....	23
8	签字.....	错误！未定义书签。

1 导言

1.1 目的

该文档是关于用户对于聊天室系统的功能和性能的要求，重点描述了聊天室的设计需求，将作为对该工具在概要设计阶段的设计输入。

本文档的预期读者是：

- 设计人员
- 开发人员
- 项目管理人员
- 测试人员
- 用户

1.2 范围

该文档是借助于当前系统的逻辑模型导出目标系统的逻辑模型，解决整个项目系统的“做什么”的问题。在这里，对于开发技术并没有涉及，而主要是通过建立模型的方式来描述用户的需求，为客户、用户、开发方等不同参与方提供一个交流的渠道。

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

C# C#是微软公司发布的一种面向对象的、运行于.NET Framework 之上的高级程序设计语言。并定于在微软职业开发者论坛(PDC)上登台亮相。C#是微软公司研究员 Anders Hejlsberg 的最新成果。C#看起来与 Java 有着惊人的相似；它包括了诸如单一继承、接口、与 Java 几乎同样的语法和编译成中间代码再运行的过程。但是 C#与 Java 有着明显的不同，它借鉴了 Delphi 的一个特点，与 COM（组件对象模型）是直接集成的，而且它是微软公司 .NET windows 网络框架的主角。

C#是一种安全的、稳定的、简单的、优雅的，由 C 和 C++衍生出来的面向对象的编程语言。它在继承 C 和 C++强大功能的同时去掉了一些它们的复杂特性（例如没有宏以及不允许多重继承）。C#综合了 VB 简单的可视化操作和 C++的高运行效率，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为.NET 开发的首选语言。

C#是面向对象的编程语言。它使得程序员可以快速地编写各种基于 MICROSOFT .NET 平台的应用程序，MICROSOFT .NET 提供了一系列的工具和服务来最大程度地开发利用计算与通讯领域。

C#使得 C++程序员可以高效的开发程序，且因可调用由 C/C++ 编写的本机原生函数，因此绝不损失 C/C++原有的强大的功能。因为这种继承关系，C#与 C/C++具有极大的相似性，熟悉类似语言的开发者可以很快的转向 C#。

.NET Framework Microsoft .NET Framework（微软.NET 框架）是用于 Windows 的新托管代码编程模型。它将强大的功能与新技术结合起来，用于构建具有视觉上引人注目的用户体验的应用程序，实现跨技术边界的无缝通信，并且能支持各种业务流程。

托管代码 (managed code) 同受管制的代码，由公共语言运行库环境（而不是直接由操作系统）执行的代码。托管代码应用程序可以获得公共语言运行库服务，例如自动垃圾回收、运行库类型检查和安全支持等。这些服务帮助提供独立于平台和语言的、统一的托管代码应用程序行为。

XML 可扩展标记语言，标准通用标记语言的子集，一种用于标记电子文件使其具有结构性的标记语言。它可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。它非常适合万维网传输，提供统一的方法来描述和交换独立于应用程序或供应商的结构化数据。

ADO.NET ADO.NET 的名称起源于 ADO（ActiveX Data Objects），是一个 COM 组件库，用于在以往的 Microsoft 技术中访问数据。之所以使用 ADO.NET 名称，是因为 Microsoft 希望表明，这是在 .NET 编程环境中优先使用的数据库访问接口。

LINQ LINQ 语言集成查询是一组用于 C#和 Visual Basic 语言的扩展，它允许编写 C#或者 Visual Basic 代码以查询数据库相同的方式操作内存数据。

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 A-1。

表 A-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
001	2014. 10. 05	0. 1	全部	初始发布版本
002	2014. 10. 14	0. 2	1. 4, 3. 2, 3. 3, 4. 1. 4, 7	术语定义, 软硬件环境, 数据词典, 约束

2 系统定义

我们分别阐述一下项目的来源、背景和项目的目标。

2.1 项目来源及背景

随着信息流量的大量增加, 人们之间越来越需要一个可以交流和联系的平台。经常上网的人可能都去过聊天室, 在聊天室里可以和有共同话语的异地的陌生人交流彼此的认识和思想, 从而增强人际软关系。同时可以通过发布一些新动态去维系彼此的硬关系。由于绝大部分网友的喜爱, 聊天室模块是众多门户网站的必备模块。其实聊天室的开发方式有很多, 以前大多使用CGI程序或Java Applet来开发聊天室系统, 但使用这些方式开发过应用系统的程序员会感觉开发的过程非常复杂, 但使用ASP.NET来开发的话一切就会变得简单起来。

为了能够开发出一款基于Web、群聊功能专一且通用的聊天室系统, 该项目将应用C#语言更好地开发ASP.NET应用程序。

2.2 项目要达到的目标

本项目设定的目标如下:

1. 系统能够提供友好的用户界面，使操作人员的工作量最大限度的减少，并且让用户使用感觉良好。
2. 系统具有良好的运行效率，能够得到提高生产率的目的。
3. 系统应有良好的可扩充性，可以容易的加入其它系统的应用，尽量满足用户对该产品的期望。
4. 平台的设计实用且灵活，更好适应未来可能发生的配置变化。
5. 通过这个项目可以锻炼队伍，提高团队的开发能力和项目管理能力。

2.3 系统整体结构

根据用户的需求陈述，可以确定本项目分为用户和管理员两个实体。用户功能如下：

- 用户登录，提供一个接口，让用户提供用户名来登录聊天室，从而让系统获得要发言用户的标识。一个开放的聊天室通常指需要用户输入用户名即可，且不管当前在线用户的用户名是否重复。但为了方便聊天室管理，还是建议在开发聊天室时提供用户注册功能，只有注册后用户才能进入聊天室聊天。
- 发送消息，提供一个接口，让登录聊天室的用户来发送聊天信息。
- 显示信息，提供一个接口，显示所有在线用户发送的聊天信息。

以上3个功能是组成聊天室的基本功能。此外一个完整的聊天室还需要提供管理员功能以方便聊天室管理员对聊天室的控制。这里的聊天室管理员就像一个讨论会的主持，他需要具有以下几个功能：

- 查看用户（包括在线和不在线的）。
- 控制用户，能够随时踢出那些扰乱聊天室的用户，并停用这些用户名。
- 查看用户聊天纪录。

他们的关系如图A-1。

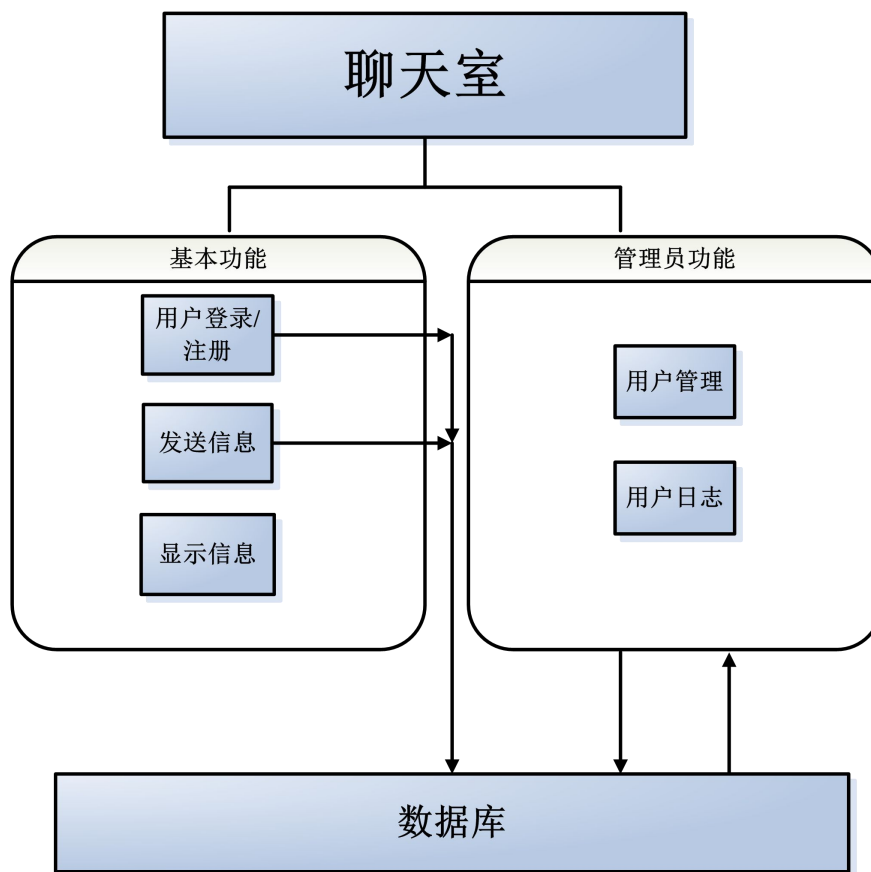


图 A-1：;聊天室功能关系图

3 应用环境

本项目的应用环境可以分硬件环境、软件环境和网络环境来描述。

3.1 系统运行网络环境

本系统的网络运行图如图 A-2，无论是用户还是管理员都可以通过网络登录到本系统中。用户可为一个或多个，进行登录、注册、发送信息等活动，管理员进行可以查看用户、控制用户等活动。

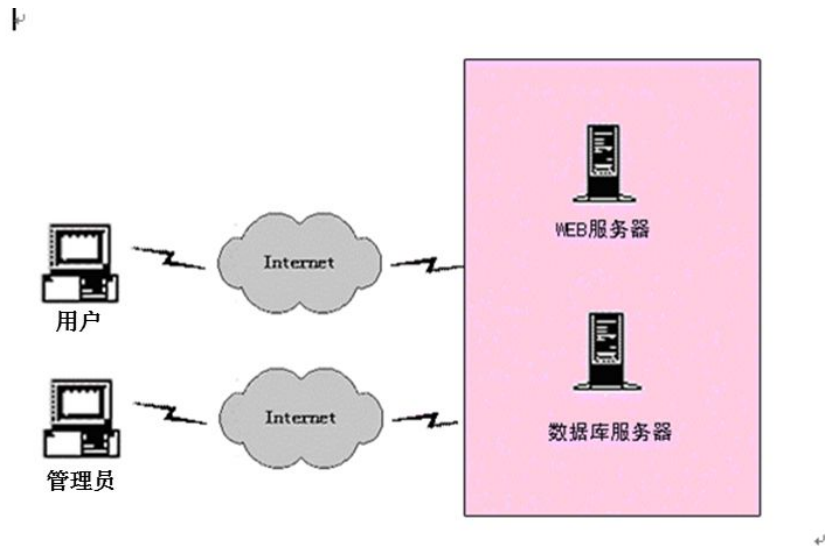


图 A-2：网络拓扑图

3.2 系统运行硬件环境

本系统的硬件环境如下：

- 客户机：普通 PC
 - CPU：P4 1.8GHz
 - 内存：1G 以上
 - 分辨率：推荐使用 1024*768 像素
- WEB 服务器
 - CPU：P4 1.8GHz
 - 内存：1G 以上
- 数据库服务器
 - CPU：P4 1.8GHz
 - 内存：1G 以上

3.3 系统运行软件环境

- 操作系统：Windows 7 或者 Windows 8
- 数据库：Oracle 11g
- 开发工具：Visual Studio 2010
- 浏览器：IE10（Internet Explorer 10）
- 服务器：ASP.NET

4 功能规格

由于功能需求分析供给不同群体使用，所以我们采用了多种需求分析方法。

4.1 结构化分析方法

首先，我们采用结构化分析方法，该方法是面向数据流的需求分析方法，主要根据内部的数据传递和变换关系，自顶向下逐层分解，描绘出满足功能要求的软件模型。

4.1.1 数据流图之顶层

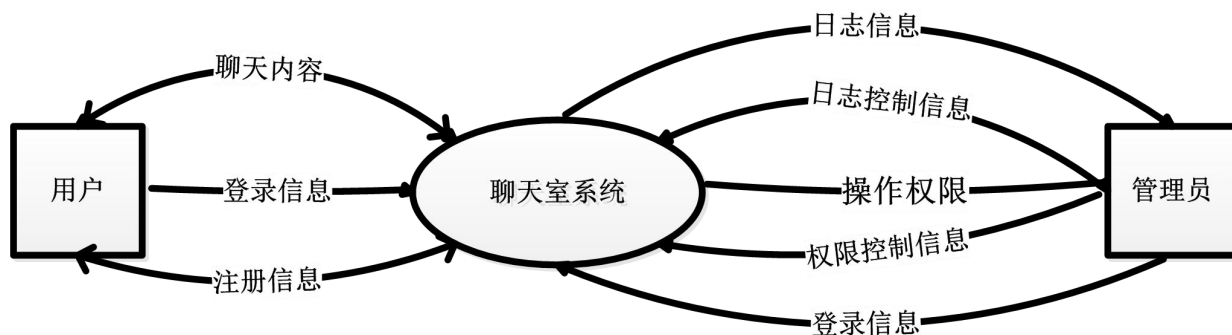


图 A-3：聊天室系统数据流图顶层图

4.1.2 数据流图之中间层

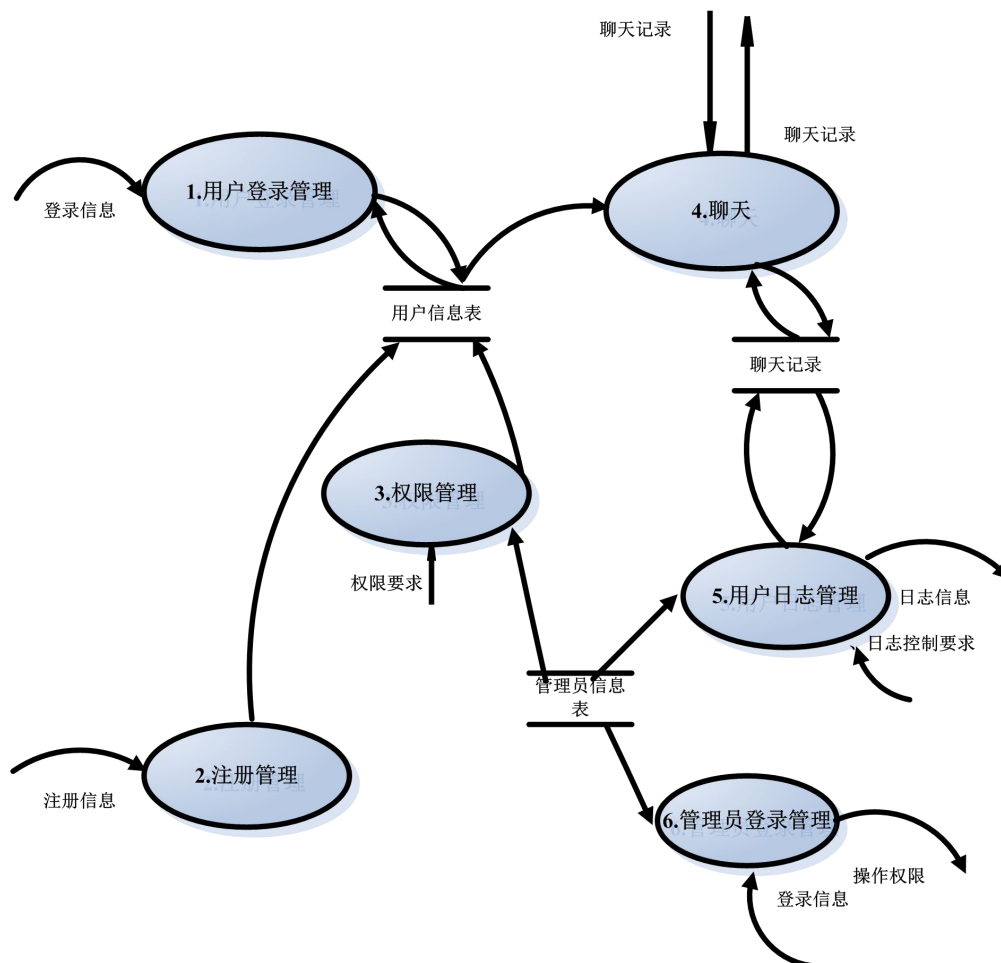


图 A-4：聊天室系统数据流图中间层

4.1.3 数据流图之底层

(1) 登录管理细化：

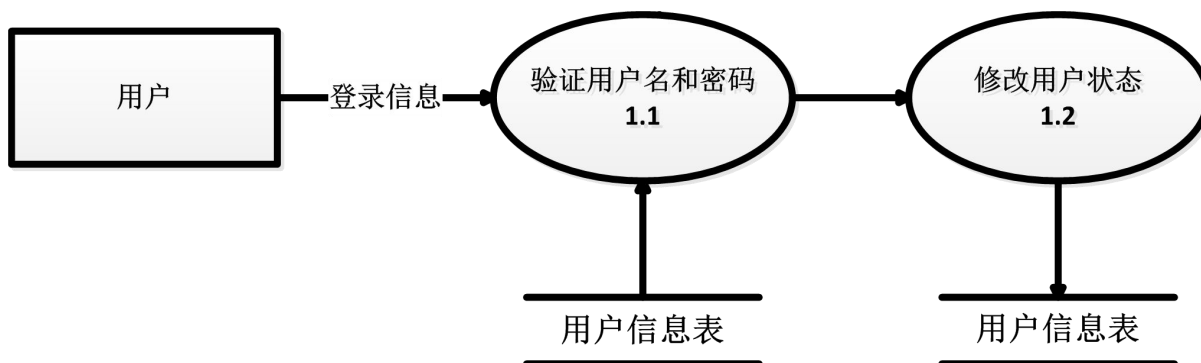


图 A-5：聊天室系统登录管理细化图

(2) 注册管理细化：



图 A-6: 聊天室系统注册管理细化图

(3) 权限管理细化:

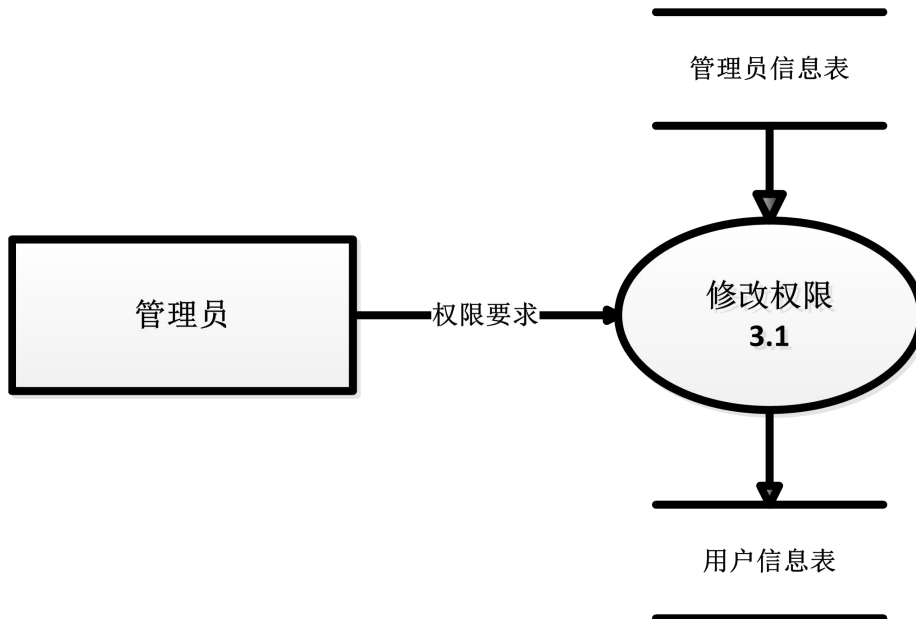


图 A-7: 聊天室系统权限管理细化图

(4) 聊天细化:

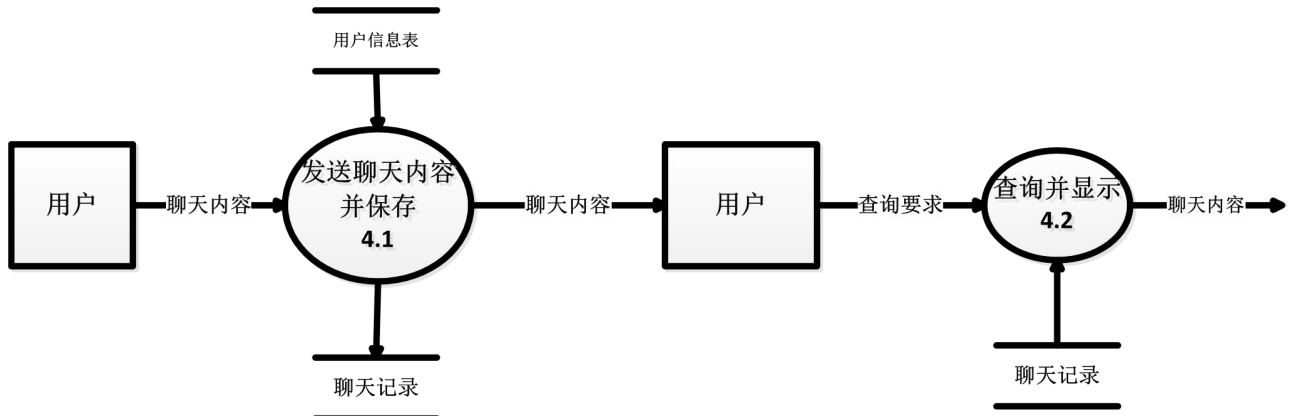


图 A-8: 聊天室系统聊天细化图

(5) 用户日志管理细化:

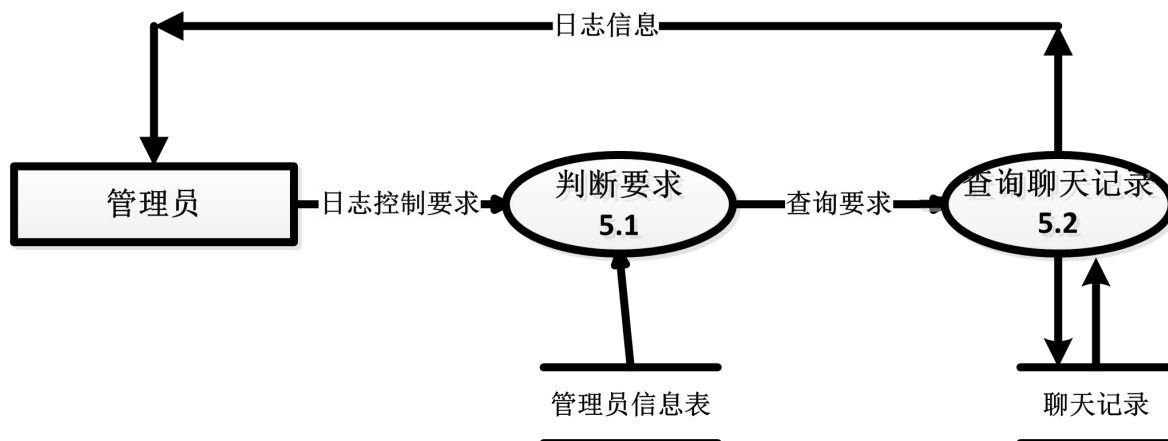


图 A-9: 聊天室系统用户日志管理细化图

(6) 管理员登录管理细化:

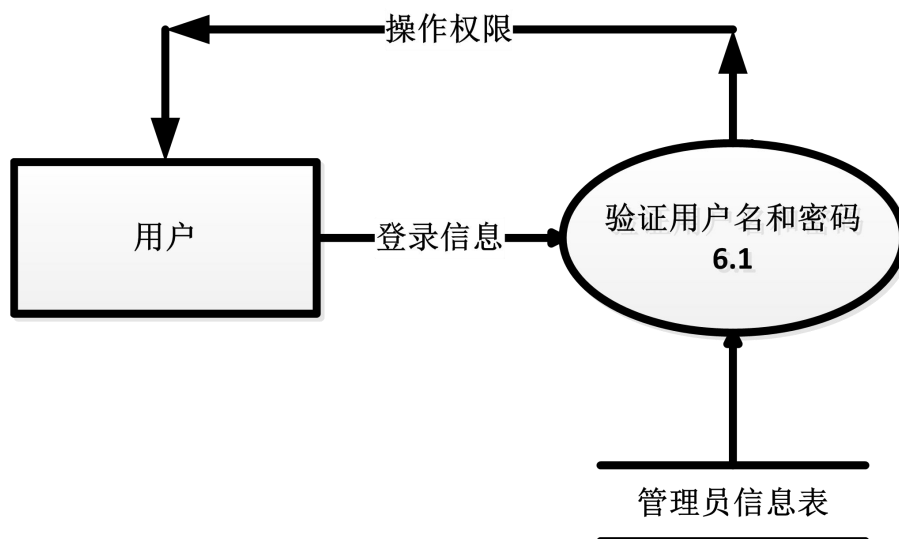


图 A-10: 聊天室系统管理员登录管理细化图

4.1.4 数据词典

(1) 数据流条目:

- ①登录信息 = 用户名(昵称) + 密码
 - ②注册信息 = [用户名 + 密码 + (电子邮箱 | 真实姓名 | 年龄 | 性别)]
 - ③查询要求 = [用户名 | 客户机 IP | 时间]
 - ④权限要求 = [用户名 | 允许用户发送信息的指示 | 不允许用户发送信息的指示 | 强制下线的指示]
 - ⑤日志控制要求 = [用户名 | 登录时间]
 - ⑥聊天内容 = 用户名 + 聊天信息 + 聊天时间 + 客户机 IP
 - ⑦(管理员)操作权限 = [是管理员 | 不是管理员]
- 注: 相当于有没有成功登录管理员账号

(2) 文件条目:

- ①用户信息表 = 用户名(昵称) + 密码 + 电子邮箱 + 真实姓名 + 年龄 + 性别 + 用

户是否在线 + 是否允许用户发送信息

②聊天记录表 = 索引 + 用户名 + 聊天信息 + 聊天时间 + 客户机 IP

③管理员信息表 = 用户名 + 密码

(3) 数据项条目：

①姓名、密码、电子邮箱、真实姓名、聊天时间：文本类型，大小为 50

②年龄、性别、：文本类型，大小为 10

③索引：数字类型，大小为 20

④用户是否在线、是否允许用户发送信息：布尔值，大小为 1

⑤客户机 IP：文本类型，大小为 15

⑥聊天信息：文本类型，大小为 10000000

(4) 加工条目（以登录/注册该部分功能举例）：

用户存在	✓	✓	
需要注册			✓
密码正确	✓		
可以收发信息	✓		

4.2 E-R图分析

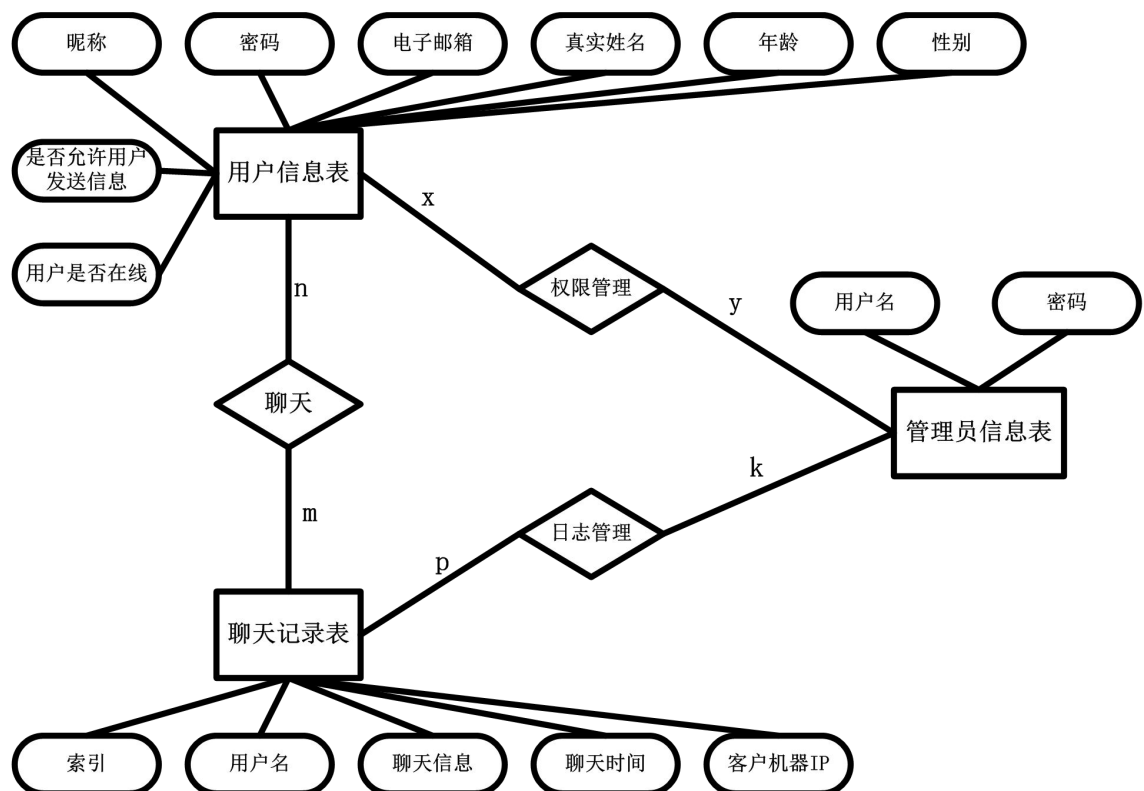


图 A-11：聊天室系统 E-R 图

其次，我们采用面向对象分析作为主要的系统建模方法，使用 UML (Unified Modeling

Language)作为建模语言。UML 为建模活动提供了从不同角度观察和展示系统的各种特征的方法。在 UML 中,从任何一个角度对系统所作的抽象都可能需要几种模型来描述,而这些来自不同角度的模型图最终组成了系统的映像。

Use Case 描述的是“actor”(用户、外部系统以及系统处理)是如何与系统交互来完成工作的。Use Case 模型提供了一个非常重要的方式来界定系统边界以及定义系统功能,同时,该模型将来可以派生出动态对象模型。

设计 Use-case 时,我们遵循下列步骤:

第一步,识别出系统的“actor”。Actor 可以是用户、外部系统,甚至是外部处理,通过某种途径与系统交互。重要的是着重从系统外部执行者的角度来描述系统需要提供哪些功能,并指明这些功能的执行者(Actor)是谁。尽可能地确保所有 Actor 都被完全识别出来。

第二步,描述主要的 Use Case。可以采取不断地问自己“这个 Actor 究竟想通过系统做什么?”来准确地描述 Use Case。

第三步,重新审视每个 Use Case,为它们下个详尽的定义。

4.3 面向对象分析方法

角色或者执行者(Actor)指与系统产生交互的外部用户或者外部系统。

4.3.1 角色(Actor)定义

用户是指在这个聊天室系统中通过这个平台进行聊天的人员,这个 Actor 主要参与登录账号、注册账号、发送消息、接受消息等功能。

管理员扮演着相当于聊天室的主持人的角色。它通过在后台的操作对用户进行权限设置、日志管理以及可以查看其聊天记录。

用户、管理员可为任意个,一个管理员可以管理多个用户,一个用户可以被多个管理员管理。它们的关系如图 A-12:

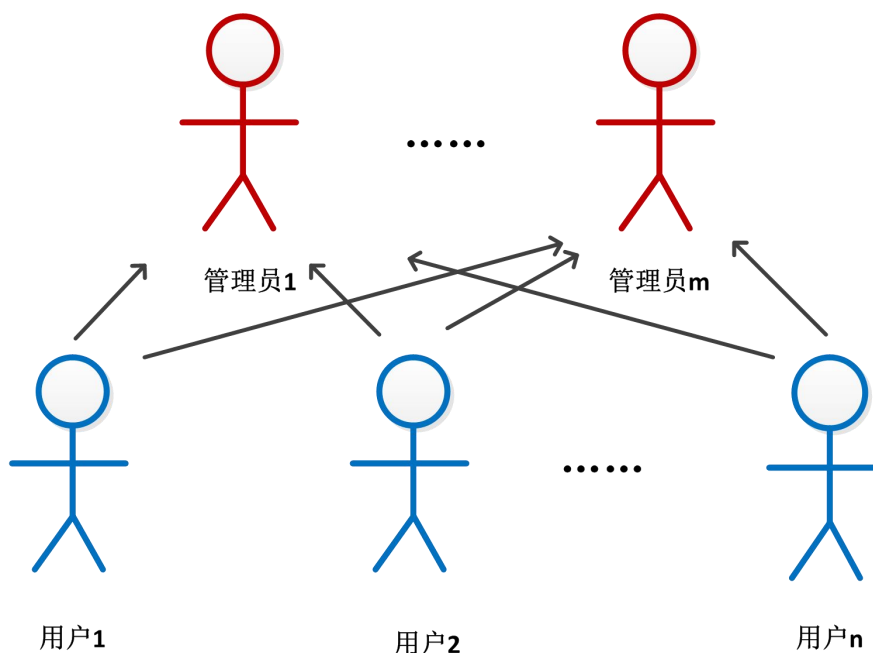


图 A-12：聊天室系统用户和管理员之间的关系图

数据库是一个与系统产生交互的外部系统，这个 Actor 负责系统的数据查询、增加、删除和修改等操作。用户和管理员的操作都需要数据库的参与。

4.3.2 系统主 Use Case 图

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。

系统的主 Use Case 图如图 A-13 所示。

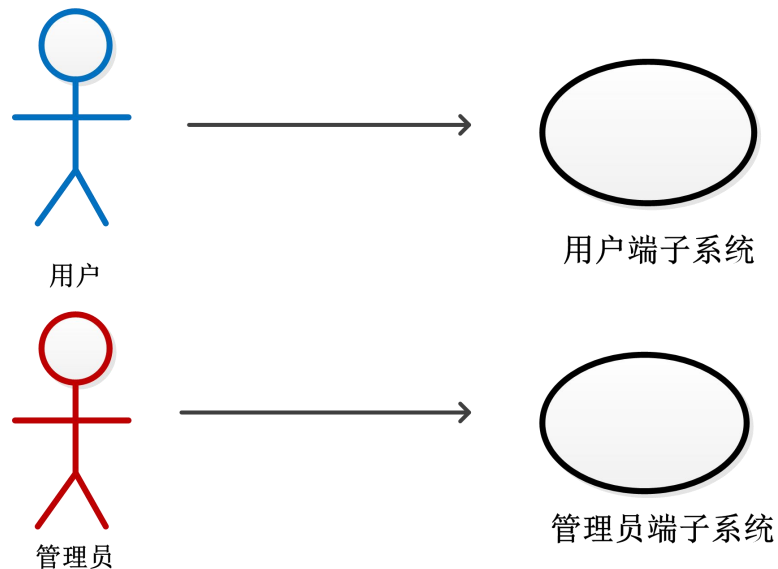


图 A-13：系统的主 Use Case 图

4.4 用户端子系统

用户端的功能主要包括登录/注册、发送信息、显示信息等功能，它的用例图如图 A-14。

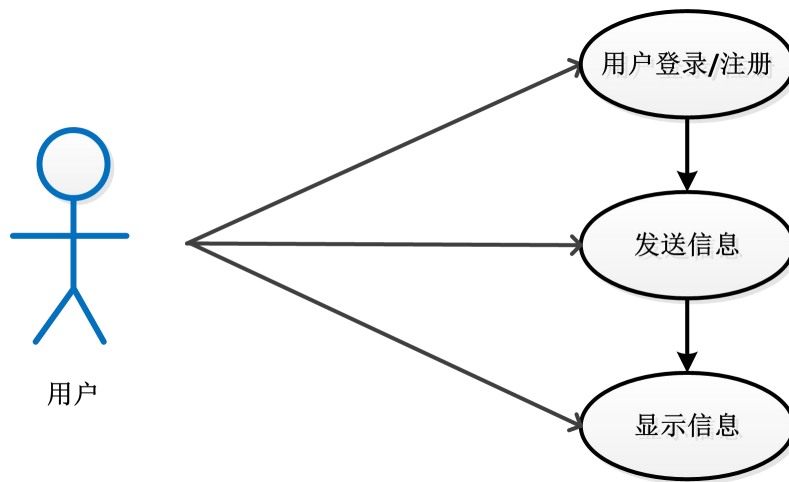


图 A-14：用户端的功能用例图

用户端管理的功能描述如下：

F-C-1：用户登录

用户进入登录的页面，在该页面会要求用户填写用户名（昵称）和密码，然后提交。

F-C-2：用户注册

用户进入注册的页面，在该页面会要求用户填写昵称、密码、电子邮箱、真实姓名、年龄、性别等。只有注册过的用户才可以登录进入聊天室。

F-C-3：发送信息

用户编写信息，然后发送。

F-C-4：显示信息

显示当前所有用户在系统内发送的信息以及系统自己发送的信息。

4.4.1 用户登录

登录功能为用户进入聊天室系统的大门。具体描述如下：

用例描述：用户登录

执行者：用户

前置条件：用户已事先注册；

后置条件：填写完用户名和密码后，进行提交。

基本路径：

- a) 用户打开登录的页面，显示两个文本框，分别是用户名和密码；
- b) 按要求进行填写用户名和密码；
- c) 按下提交；
- d) 如果系统验证用户名和密码成功，即可进入聊天室；如果验证失败，要么离开页面退出，要么点击注册。

4.4.2 用户注册

用户注册功能是为了满足没有身份的用户向系统注册身份。具体描述如下：

用例描述：用户注册

执行者：用户

前置条件：用户没有拥有系统的身份；

后置条件：按要求正确填写注册信息，注册成功即可。

基本路径：

- a) 基本信息输入，包括用户名、密码、电子邮箱、真实姓名、年龄、性别等。
- b) 提交后，若没有重复用户，即可拥有系统身份。

4.4.3 发送聊天信息

用户把自己想发的话发送出去。具体功能描述如下：

用例描述：发送聊天信息

执行者：用户

前置条件：用户已经成功登录；

后置条件：用户编辑消息然后发送。

基本路径：

- a) 用户在聊天框下编辑文字；
- b) 若没有什么需要修改后，就点击发送。

4.4.4 显示聊天信息

显示当前所有用户在系统内发送的信息以及系统自己发送的信息。

具体功能描述如下：

用例描述：显示聊天信息

执行者：用户

前置条件：已经有用户发送聊天信息；

后置条件：聊天框显示当前所有用户在系统内发送的信息以及系统自己发送的信息。

基本路径：

- a) 当前所有用户的聊天信息不断更新。

- b) 在屏幕上稳定显示当前所有用户在系统内发送的信息以及系统自己发送的信息。

4.5 管理员端子系统

管理员端主要负责用户日志和用户权限两方面。它的功能涉及到看查看用户是否在线、控制用户从而可以决定是否停用该用户以及查看用户聊天记录。

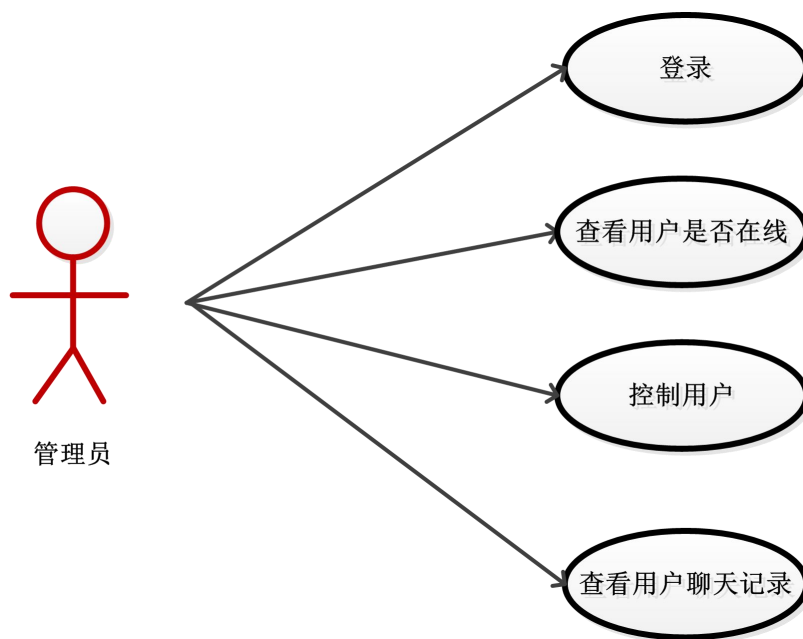


图 A-15：管理员端用例图

管理员端的这些 Use case（用例）描述如下：

F-M-1 管理员登录

管理员端的人员要登录到管理端必须经过登录界面，输入自己的用户名和密码。

F-M-2 查看用户是否在线：

管理员可以查看所有用户的在线情况，以便了解当前在线人数。

F-M-3 控制用户

管理员可以在接到用户投诉后将用户踢出聊天室，以及有权力停用该用户。

F-M-4 查看用户聊天记录：

管理员可以在接到用户投诉后查看其聊天记录，以此确认情况是否属实。

4.5.1 管理员登录

管理员自身也是需要登录，否则没有权限管理用户。具体描述如下：

用例描述：管理员登录

执行者：登录前也认为是用户，登录后身份变为管理员

前置条件：该管理者身份存在；

后置条件：管理员登录后，即可进行用户日志和用户权限方面的管理。

基本路径：

- a) 进入登录页面，按要求进行填写用户名和密码；
- b) 按下提交；
- c) 如果系统验证用户名和密码成功，即可进入聊天室，拥有管理用户的权限；如果验证失败，则不能成功拥有管理员身份。

4.5.2 查看用户是否在线

管理员查看用户状态，分为在线和离线两个状态，从而及时掌握用户在线情况。具体描述如下：

用例描述：查看用户是否在线

执行者：管理员

前置条件：管理员已登录系统；

后置条件：掌握用户在线情况。

基本路径：

- a) 打开用户信息表；
- b) 查看用户信息表；
- c) 掌握用户在线信息。

4.5.3 控制用户

管理员接到用户投诉后，限制被投诉用户的发言权等。具体功能描述如下：

用例描述：控制用户

执行者：管理员

前置条件：管理员已登录系统；

后置条件：对于扰乱聊天秩序的用户，可强制使其下线，或者停用其账号的使用。

基本路径：

- a) 接到投诉；
- b) 可通过用户名或者其他关键字，查找被投诉用户的相关信息。
- c) 强制被投诉用户下线或者停用其账号。

4.5.4 查看用户聊天记录

管理员查看所有用户聊天记录，如聊天时间、聊天内容等。具体功能描述如下：

用例描述：查看用户聊天记录

执行者：管理员

前置条件：管理员已登录系统；

后置条件：所有用户的聊天记录可以被查看。

基本路径：

- a) 输入要被查看的用户名或者其他关键字；
- b) 系统完成搜索，得到相应的聊天记录。

5 性能需求

根据用户对本系统的要求，确定系统在响应时间、可靠性、安全等方面有较高的性能要求。

5.1 界面需求

系统的界面要求如下：

- 1) 页面内容：聊天室的主题突出，站点定义、术语和行文格式统一、规范、明确，栏目、菜单设置和布局合理，传递的信息准确、及时。内容丰富，文字准确，语句通顺；专用术语规范，行文格式统一规范
- 2) 导航结构：页面具有明确的导航指示，且便于理解，方便用户使用。如提交、发送、注册等提示。
- 3) 技术环境：页面大小适当，能用各种常用浏览器以不同分辨率浏览；无错误链接和空链接；采用 CSS 处理，控制字体大小和版面布局。
- 4) 艺术风格：聊天室系统的页面应该简约不杂糅。界面、版面形象清新悦目、布局合理，字号大小适宜、字体选择合理，前后一致，美观大方；动与静搭配恰当，动静效果好；色彩和谐自然，与主题内容相协调。

5.2 响应时间需求

无论是客户端和管理端，当用户登录，进行任何操作的时候，系统应该及时的进行反应，反应的时间在 5 秒以内。系统应能监测出各种非正常情况，如与设备的通信中断，无法连接数据库服务器等，避免出现长时间等待甚至无响应。

5.3 可靠性需求

系统应保证 7X24 内不当机，保证 20 人可以同时在客户端登录，系统正常运行，正确提示相关内容。

5.4 开放性需求

系统应具有十分的灵活性，以适应将来功能扩展的需求。

5.5 可扩展性需求

系统设计要能够体现扩展性要求，以适应将来功能扩展的需求。

5.6 系统安全性需求

系统有严格的权限管理功能，各功能模块需有相应的权限方能进入。系统需能够防止各类误操作可能造成的数据丢失，破坏。防止用户非法获取网页以及内容。

6 产品提交

提交产品为：

- a) 应用系统软件包
 - b) 数据库初始数据
 - c) 系统开发过程文档
 - d) 系统使用维护说明文档
- 提交方式：CD 介质 或电子版文档

7 实现约束

系统的实现约束如下：

- a) 操作系统为 Windows 7 或者 Windows 8
- b) 开发平台为：Visual Studio 2010
- c) 数据库为：Oracle 11g
- d) 架构：二层架构 UI→数据库

例图如下：

