校友社区网站[ 概需求要分设析计]

# 需求分析

## 1．1功能需求

1．1．1描述

本网站是在为了满足学校校友之间交流的情况下开发的，随着信息时代的发展，社会普遍对网络的需求，该网站能够提供更多实用功能，本着实用，美观，高效的目的，该网站能够提供如下功能需求：1、校友交流平台2、校友信息发布平台3、院系班级交流功能4、社区新闻公告发布5、社区查询等功能。

1．1．2系统功能

1)校友社区网站由如下功能组成：

>用户注册

新用户注册，提供用户信息，检验用户信息的有效性，并将用户信息持久化。

>用户登陆

提供用户凭证，验证用户信息，基于角色授权。

》用户管理 

管理员由系统初始化分配一个，管理员可以对用户信息进行部分更改，主要包括用户角色调整，版主调整，删除用户等。

1 0

>《网站社区版块管理

衽区管理员可以添加、删除、调整网站版2 块 I ll ℃ 0 rl"l

》留言管理

社区管理员可以对所有留言进行转移、置顶、删除等操作，社区管理员可以转移本版块留言，也可以对本版块论纭中的文章进行置顶、删除等操作。

》留言发表

注册用户可以在注册的社区版块中发表新留言信息。

>留言回复

用户可以对自己感兴趣的社区版块留言簿中发表留言回复。

》页面浏览

用户可以浏览所有权限范围内的页面。

》社区查找

用户可以提供标题关键字查找所有己建社区，注册用户可以查找己创建的班级社、年级社区、个性社区。

》相关链接

用户可以进入其它校友社区浏览等。

2）网上校友社区系统总体功能需求框架图如下所示：（图2一1）

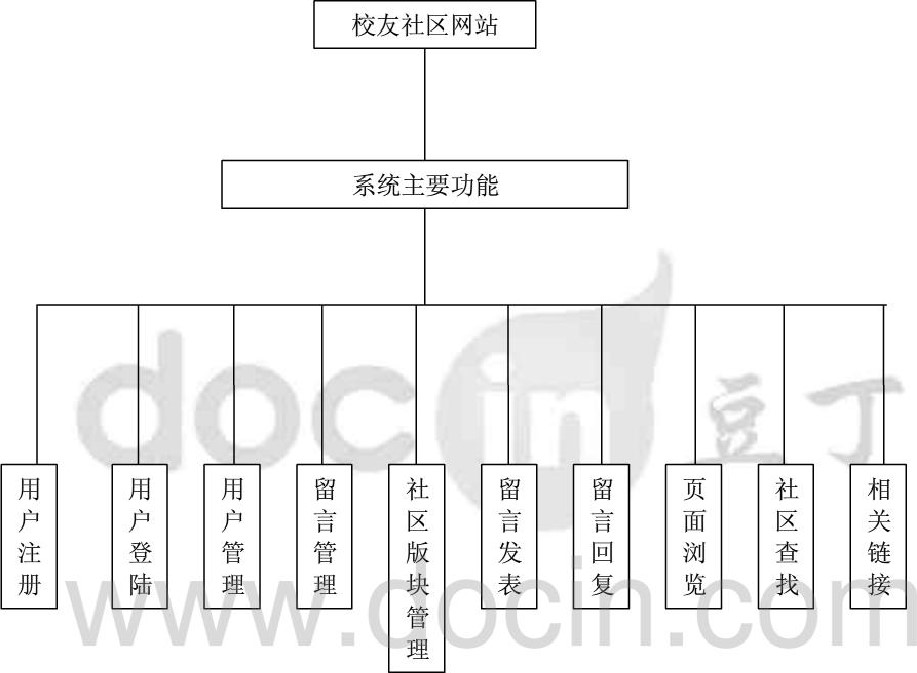
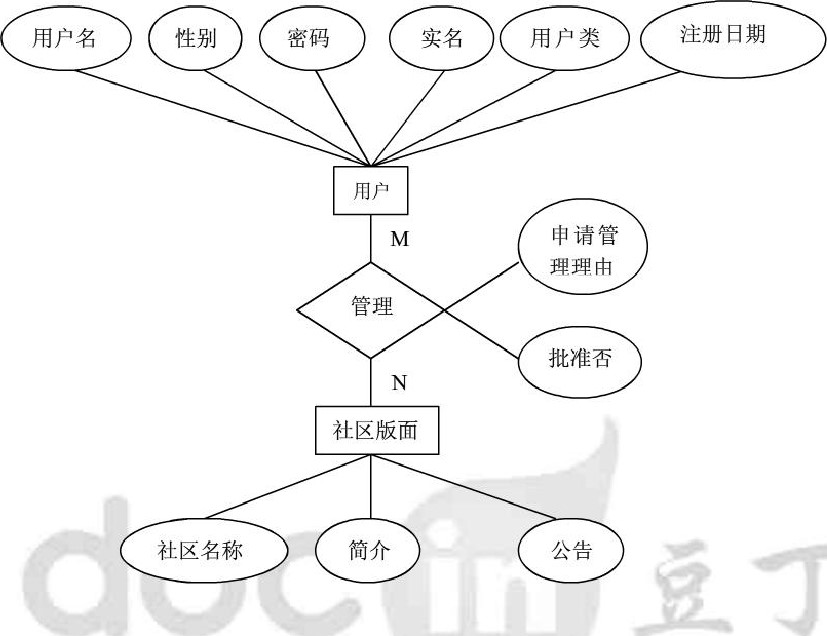


图2一1

## 1．2数据描述

本系统实体之间主要有以下几类联系(Relation)：用户与网站页面之间存在管理联系 (M：N)、用户与留言发表之间（1：M)、用户与友情链接（1：M)、用户与社区版块（1：M)、社区版面与内容（1：

1基本实体一联系图( )R图）



1

）

用

户

与

版

面

之

间

管

理

关

系

ER

图

2

）

用

户

与

留

言

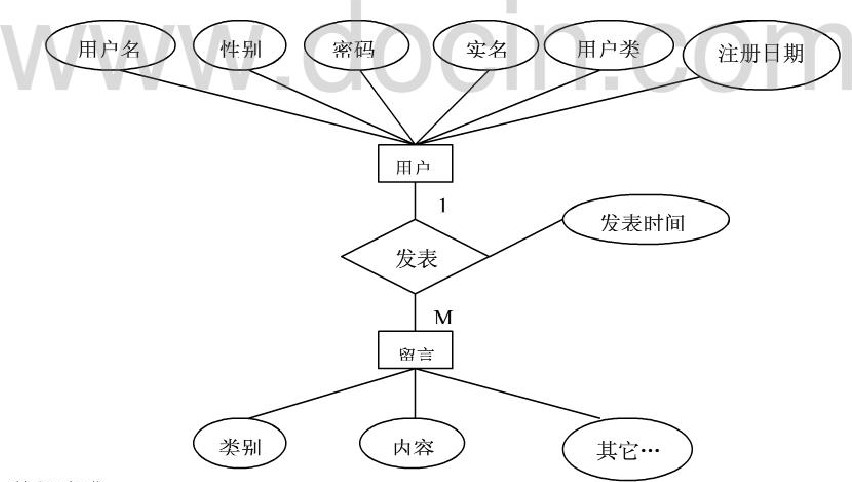
之

间

的

ER

图



1.2

．

2

数

据

字

典

详见概要设计系统数据结构设计

## 1．3性能需求

本节将较完整地描述系统的性能需求。

1． 1网络需求

D服务器要求能承载1000用户同时在线。

2）网络带宽要求100朋以上。

1．3．2响应需求

D数据精确度：查询时应保证查全率，所有相应域包含查询关键字的记录都应能查到。

1. 时间特性：进行查询时以秒为单位，以达到实时性；而进行增加，删除和修改等操作时，可以根据数据的多少分别以秒和分为单位，原则是操作人员不因时间而影响效率。

般操作的响应时间应在1一2秒内，对磁盘和打印机服务响应时间应在可接受的时间内完成。

1. 适应性：对前面提到的运行环境要求不应存在困难。

1．3．3安全性需求

1)客户端与服务器两者之间通信的数据必须加密。

2）数据库的管理员只能有一名，只有他（她）可以对数据库的所有信息作任何操作。

## 1．4接口需求

本节将提供可确保系统正确地与外部部件进行通信的信息。 

1．4．1用户接口需求

提供用户登录框，进行权限认证，有相关版块一一一班级社区、年级社区、个性社区、新闻公告、友情链接、意见建议等。利用菜单界面驱动方式，对用户友好，必须对鼠标和键盘单独支持。

1．4．2硬件接口需求

1)服务器：

主频要求2．8GHz以上，内存IGB以上，显卡的颜色配置需要设置为24位增强色或32

位真彩色。（具体视学校所提供的服务器而定）

2）终端设备：

主频要求400鼎z以上，内存512MB以上，显卡的颜色配置要设置为24位增强色或32 位真彩色，显示器分辨率设为1024\*768。

1．4．3软件接口需求

1)操作系统：

支持Windows XP、 Windows 7、Windows 8

2）应用软件：

数据库应用软件：My SQL

4通信接口需求

客户端和服务器端的一般通信采用socket通信方式，数据传输采用SSL加密机制。

## 1．5出错处理需求

1 )本系统可能有出错的情况：

》用户身份认证时可能出错。

》信息输出出错。

2）出错处理方法及补救措施：

》根据出错的种类提示身份认证重新输入。

>系统给出出错提示。

>系统给出正确的操作序列。

## 1．6设计和实现上的限制（约束）

项目的第一个版本必须在按时交付交付，所以，对网站的功能实现上采取实用的原则。

力求开发一个出功能精简的网站。

。六d0C10 ， corn

2、概要设计

# 第一章：概要设计

## 1. 1编写目的

本文档作为SNS的概要设计说明文档，用于与用户确定最终的目标，并成为协议文本的一部分，同时也是本系统设计人员的基础文档。

1. 1. 1概要设计说明书目的

本概要设计说明书说明了SNS校友社区系统设计的整体结构。

1. 1. 2预期读者

本系统开发人员及维护人员。

# 1. 2背景

SNS：Social Network Service,社交网络服务，是电子商务网站中一种常见功能，也是互联网上一种新兴起的互动交流服务。它为上网用户提供了也各自由的讨论区。通过论坛可以向用户提供开放性的分类专题讨论区服务，同时注册的用户可以根据需要在论坛上发表文章，交流技术经验，或者提出问题并表达自己的观点。不仅如此0上网的用户还可以在论坛中看到他人发表的文章，并且能够对该文章进行评论。

一般情况下，SNS按不同主题分为多个布告栏，其设立多是依据使用者的要求和喜好，但多具有信件交流、软件交流、信息发布等功能。

目前，木部分SNS由教育机构、研究机构或商业机构管理，丛多有自己的拨入电话号码，用户只需电脑、调制解调器和电话线就可通过电话拨号登录SNS站点。

1. 2．1待开发软件系统的名称

SNS校友社区系统

# 1. 3定义

1. 3．1 本文档中涉及的专业词汇

1. GB：中华人民共和国国家标准的英文缩写字母
2. 构件：具有某种功能的可重用的软件模版单元，表示了系统中主要的计算元素和数据存储。
3. 逻辑视图：描述支持系统的功能需求的视图。
4. 开发视图：也称模块视图，主要侧重于软件模块的组织和管理描述。

1. 3．2 名词说明

1. SNS：Social Network Service

2一JSP (JavaServer Pages）JSP技术使用Java编程语言编写类WL的tags和 scriptlets，来封装产生动态网页的处理逻辑。网页还能通过tags和scriptlets 访问存在于服务端的资源的应用逻辑。JSP将网页逻辑与网页设计和显示分离，支持可重用的基于组件的设计，使基于Web的应用程序的开发变得迅速和容易

3．Struts只是一个WC框架(Framework)它用于快速开发Java Web应用。Struts实现的重点在C(Controller),包括ActionServ1et/RequestProcessor和我们定制的 Action，也为v（View）提供了一系列定制标签(Custom Tag)o但struts几乎没有涉及M (Model)，所以Struts可以采用JAVA实现的任何形式的商业逻辑。

## 1. 4参考资料

1.教材：软件体系结构实用教程付燕等西安电子科技大学出版社2009年9月

2．参考书籍：软件体系结构原理/实践方法张友生清华大学出版社2006年11月

3．软件架构设计温昱电子工业出版社2007年3月

生软件体系结构设计第二版张友生等清华大学出版社2006年11月第二章总体设计（系统架构设计）

2口需求规定

2就1输入输出要求

界面风格：要求整体界面美观，有清晰的层次感，布局简洁、合理。同时保证后台的管

### 2口．2时间要求



0

一

理

页

面

和

前

台

时间需求：在软件方面，响应时间，更新处理时间都比较快且迅速，系统响应时间不能超过20秒。

2就3灵活性要求

灵活性：当用户需求，加操作方式，运行环境，结果精度，数据结构等其他软件接口等发生变化时，设计的软件能做出适当调整，灵活性非常大。

2· 2运行环境

2 · 2．1设备

1.主机类型如表2一1 表2一1主机类型

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | | 服务器标准配置 | |
| CPU | | Intel i3 2340 2．OGHz以上 | |
| 内存 | | IG以上 | |
| 硬盘 | | 350G以上 | |

2．网络类型：百兆高速局域网

3．存贮器容量：大容量存贮器

4．其他特殊设备：网络打印机，复印机

2．2 2支撑软件

I .操作系统：Windows XP Windows7及以上

2．数据库管理系统：MySQL3. 0以上版本数据库

3．其他支撑软件：J2SDK1. 5及以上版本

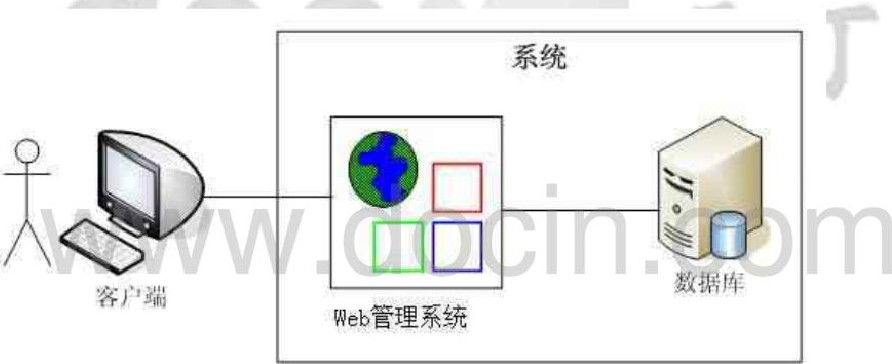
4．应用服务器：Tomcat 7．0以上

### 2．3基本设计概念和处理流程

2．3，1系统概述

1.系统采用基于J2EE的轻量级B/S架构体系

SNS校友社区系统采用B/S架构（浏览器/服务器）模式来实现。考虑到系统应用性、安全性、可扩展性与可维护性，决定采用基于J2EE的轻量级架构体系。其体系结构图如下所示：



2、为什么对本项目要应用轻量级的框架技术

轻量级容器的设计目标是为了能够避免如下所有这些麻烦事情，基于以下的各个优点，我们决定在本项目中采用轻量级的框架技术。

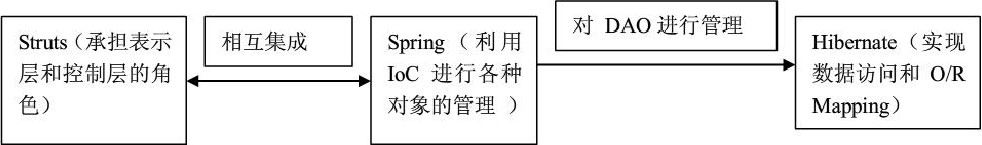
 侵略性的API（代码依赖于EJB)

 对容器的依赖（代码不能在EJB容器之外工作）  只提供固定的一组功能，不具备配置能力  启动时间长

 部署过程取决于特定的产品，无法通用

3．2 系统架构示图

1、本论坛系统的整体架构设计为Struts \*Spring +hibernate架构组成



（D对于表示层

经验表明，最好的方法是选择己存在的并己得到证明了的web应用框架，而不是自己去设计和开发新的框架。我们拥有多个可选择的框架，如Struts, WebWork和JSF等，在本项目中，我们选择采用Strutso

1. EJB和POJO都可以用来创建业务逻辑层

如果应用是分布式的，采用具有remote接囗的EJB是一个好的选择：由于本系统是一个典型的不需要远程访问的web应用，因此选用POJO,并充分利用Spring框架的loc和AoP 的特性，将是实现业务逻辑层的更好选择。

1. 在持久层中由于需要利用关系型数据库实现数据的持续化，但在应用中可以存在多种方法可用来实现：

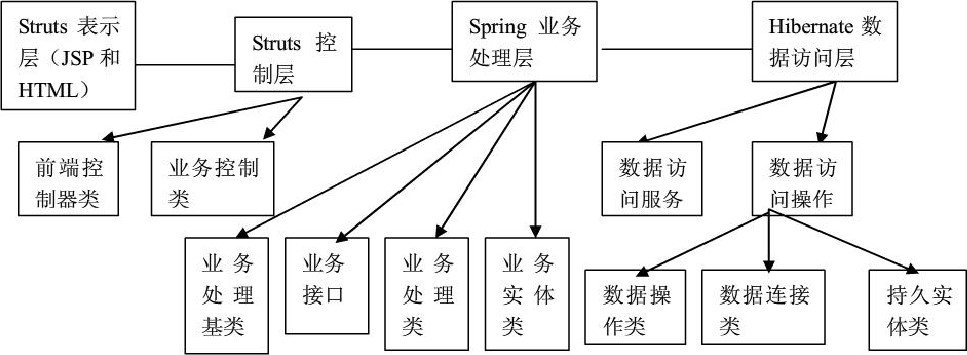
JDBC：这是最为灵活的方法，然而，低级的JDBC难以使用，且质量差的JDBC代码很难运转良好

EJB Entity beans：CMP的Entity bean是一种分离数据访问代码和处理ORM的昂贵的方法，它是以应用服务器为中心的方为即Entity yean不是将应用与某种数据库类型而是 EJB容器约束在。

O/R Mapping框架：一'亠个ORM框架采用以对象为中心的方法实现数据持续化，丨以对象为中心的应用易于开发并具有高度的可移植性一一在该领域中存在几个框架可用一JDC)、

Hibernate、TopLink以及iBATIS和CocoBase等。在本项目中我们选用Hibernateo

1. 架构示图



本系统采用了多层非分布式的构架，上图展示了系统的分层以及每一层中所采用的技术和对应的框架，并且各层将存在于同一个web容器中。

1. 该形式的总体架构设计的主要特点

(1)遵循sun J2EE中两个主要的原则：“多层架构、松藕合"。

由于采用分层的设计方式，各个模块功能相互独立封装，层与层之间关联少，保持松耦合连接，稳定性高，便于扩展和维护。

（2）本项目中的每一层所采用的技术都是可替换的。

例如struts可以被JSF或者Tapestry替换掉，JDC)可替换Hibernateo在每个层中都不同程度地应用了J2EE中常用的设计模式。使用基于POJO的轻量级架构，从而使得原统易于测试；便于移植；“开发一发布"周期短。

4、各层中的组件

(1)表示层由Struts JSP组件实现，利用了Struts中的构造标签技术，在用户浏览界面利用表单构造网页的整体结构

（2）控制层由Struts中的ActionServlet和Action组件实现，并利用ActionForm封装JSP页面中的表单。将页面整体作为对象处理，在相应的Action了中调用业务逻辑，完成业务功能。

龙前端控制层：ActionServ1et类，并且对它加以扩展。

#### 龙业务中心控制层：各个业务Action类（标准Action类和DispatchAction„类）（3）业务处理层由spring中的IOC来管理 0



业务处理塞类：将各个业务功能模块中共同的部分抽象出，丛而完成一些共同的功能。

各个业务处理类：完成具体的应用功能的各个模块

（4）数据访问层由Hibernate框架来提供技术支持

数据库操作(DAO)类：完成对数据库数据的相关操作（增、删、解、查询等）。

数据持久(PO)类：针对应用原统中的各个数据库表提供对应的POJO类

2 3各层中应用了相应的主流的J2EE框架技术

1. 服务器端表示层Struts框架完成如下工作

 客户端表单进入的验证；  管理请求和响应；

 提供控制器来完成页面流转和向业务逻辑层的委托：  返回到客户端页面显示。

 其它：标签技术、wc、成熟技术、ActionForm技术等

1. 业务逻辑层spring框架完成如下工作龙为服务器段表示层提供松散的耦合：

 处理真实的企业级应用：  事务管理的选择：

龙 协调各种业务逻辑对象之间的依赖关系；  为持久层和业务逻辑层之间提供松散耦合：  实现持久层的业务逻辑。

其它：

 解藕（类与类、系统本身脱离容器）

 AOP（统一地解决系统中一些“切面" 一一技术性的问题）

 IOC（对象的管理由容器完成）

 POJO（普通JavaBean)一一一不继承框架中某个类

 容器服务（事务、数据库连接池）---spring中己经提供了  包装其它的框架（简化）

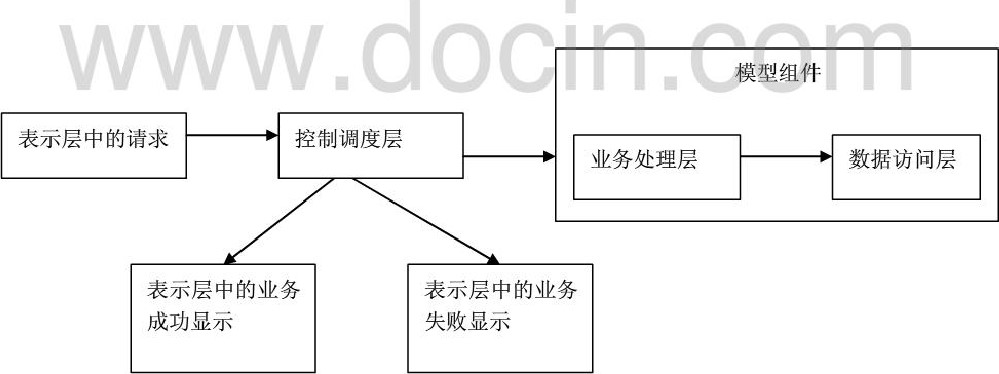
3、持久层Hibernate框架完成如下工作

》对数据库进行查询，得到持久化对象PO,

》对数据库进行添加、删除、修改的动作并以PO来进行。



>域模型层VO完成如下工作：为各层之间数据交互服务，、同时也在持久层部分可  以描述一个实体，并与PO进行转换。



0

2

．

3

．

4

系

统

基

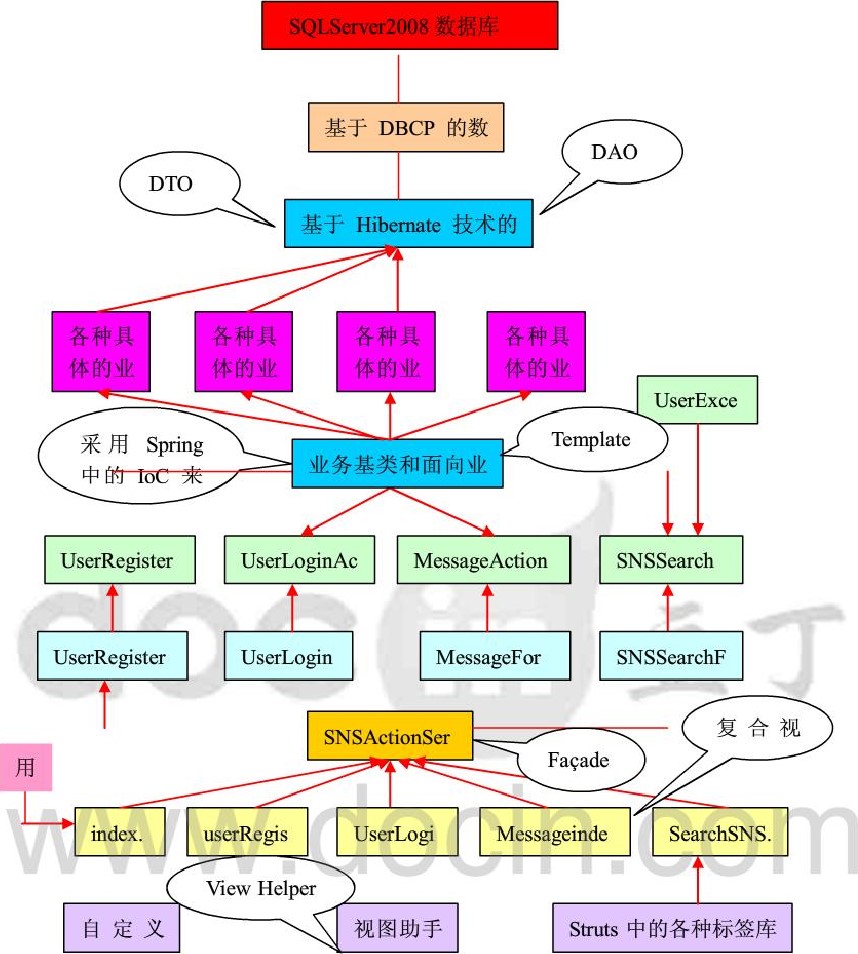
于

wc

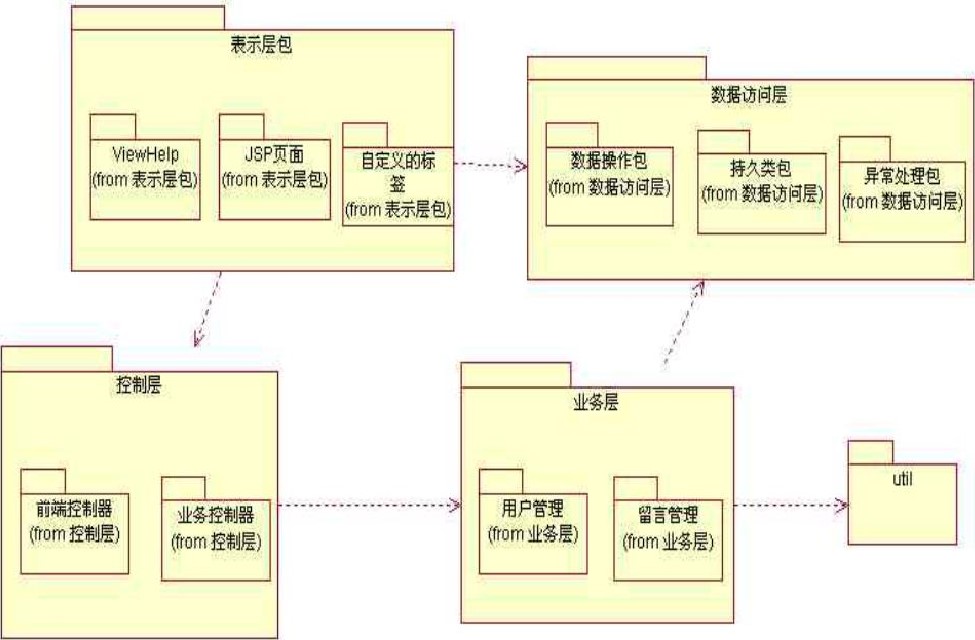
设

计

## 2. 3. 5



0



2

．

4

．

1

体

系

结

构

包

图

（

架

构

包

图

）

2

．

4

系

统

结

构

（

系

统

各

个

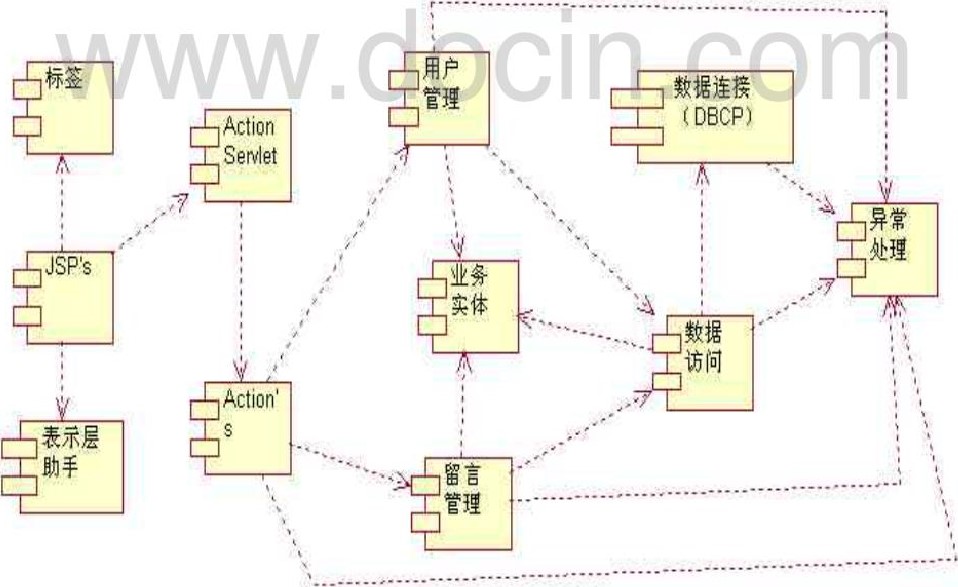
组

件

设

计

）



2

．

4

．

2

组

件

设

计

图

（

系

统

中

的

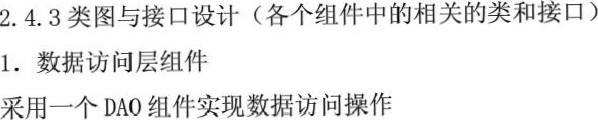
各

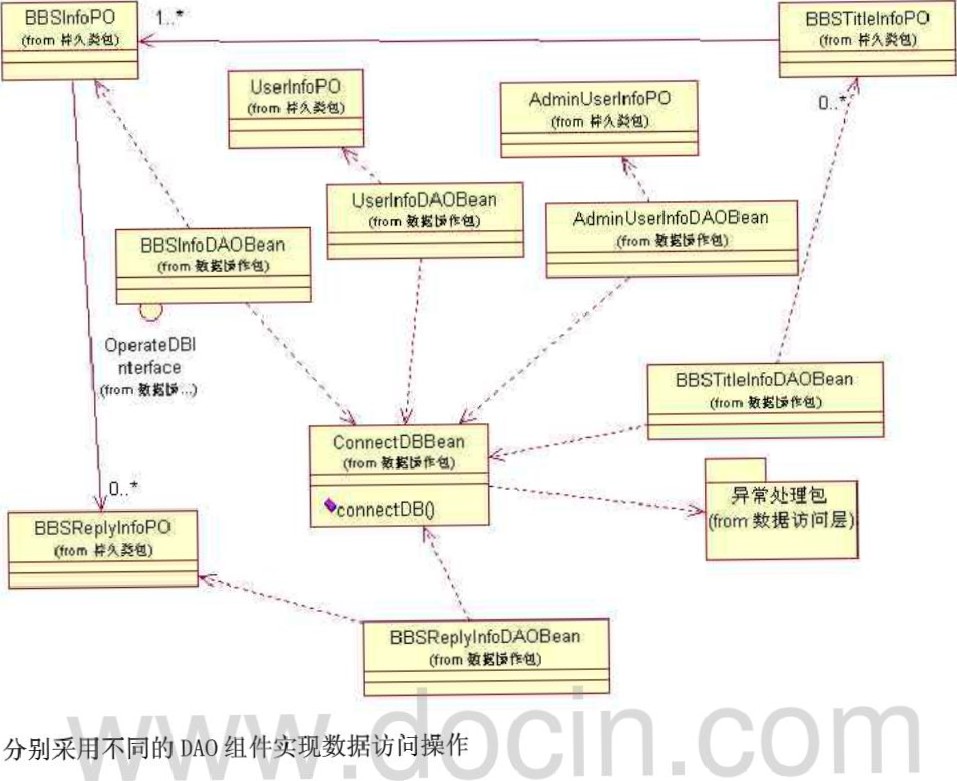
个

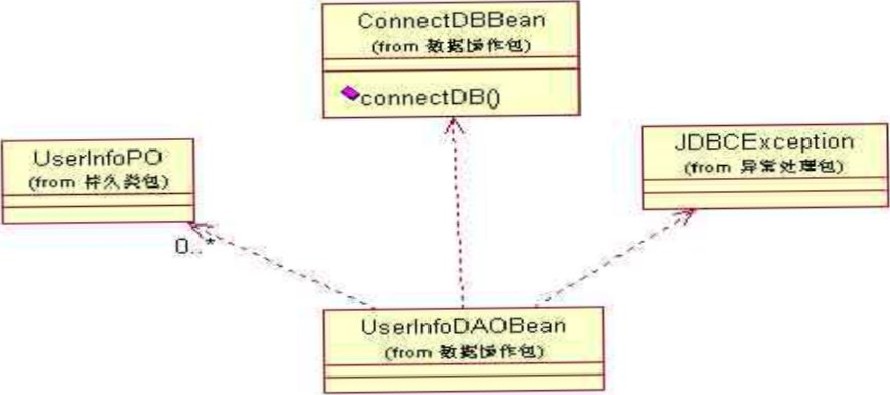
组

件

）



jcnmcorn

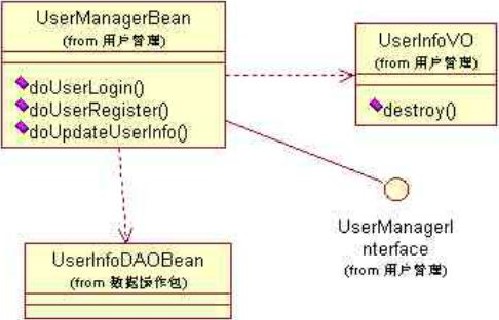




|  |
| --- |
| BBSlnfoVO  (from |
|  |

|  |
| --- |
| BBSlnfoManagerBean  (from g |
|  |

|  |  |
| --- | --- |
|  |  |
| BBSlnfoDAOBean  (from •X Eliftg) | |
|  | |
|  | |

BBSlnfoMana gerlntefface

(from

|  |  |
| --- | --- |
|  | AdminUserManagerBean  (trom TE) |
|  |  |

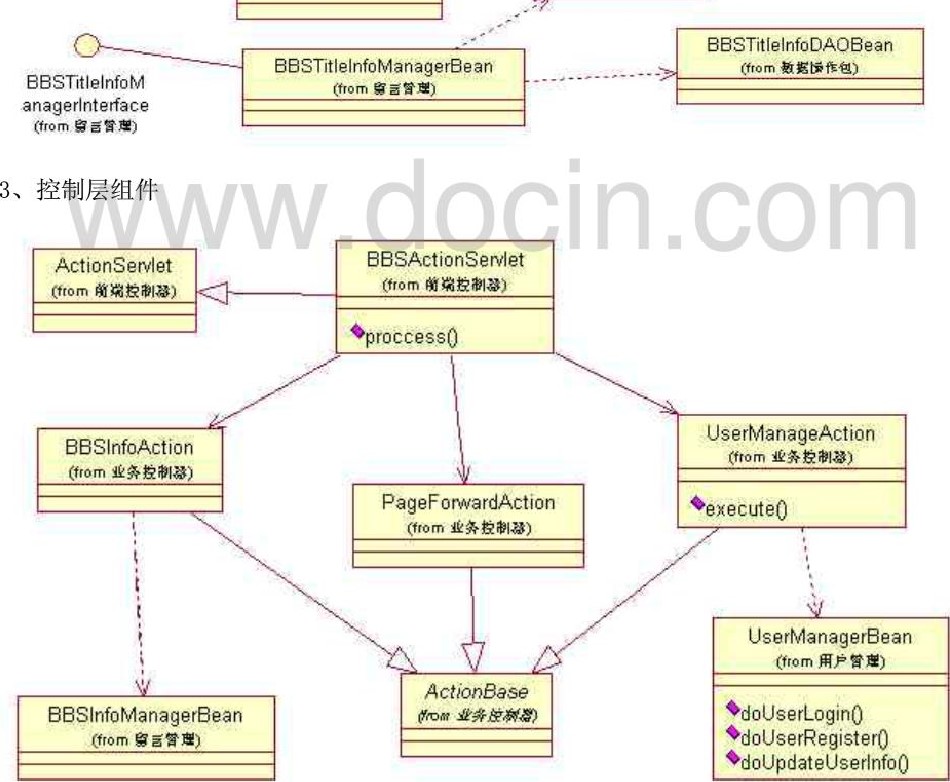
|  |
| --- |
| AdminUserlnfoDAOBean  (from |
|  |

AdminlJsertv1a nagerlnterface

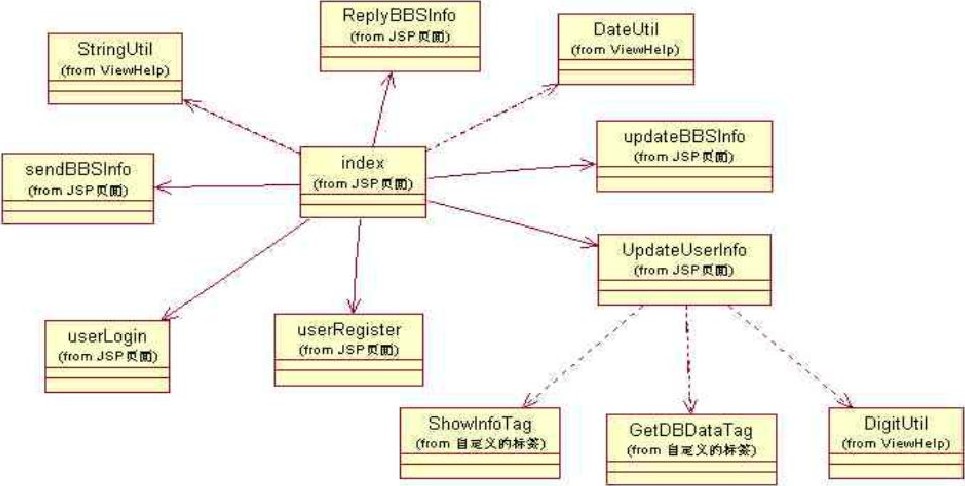
(from

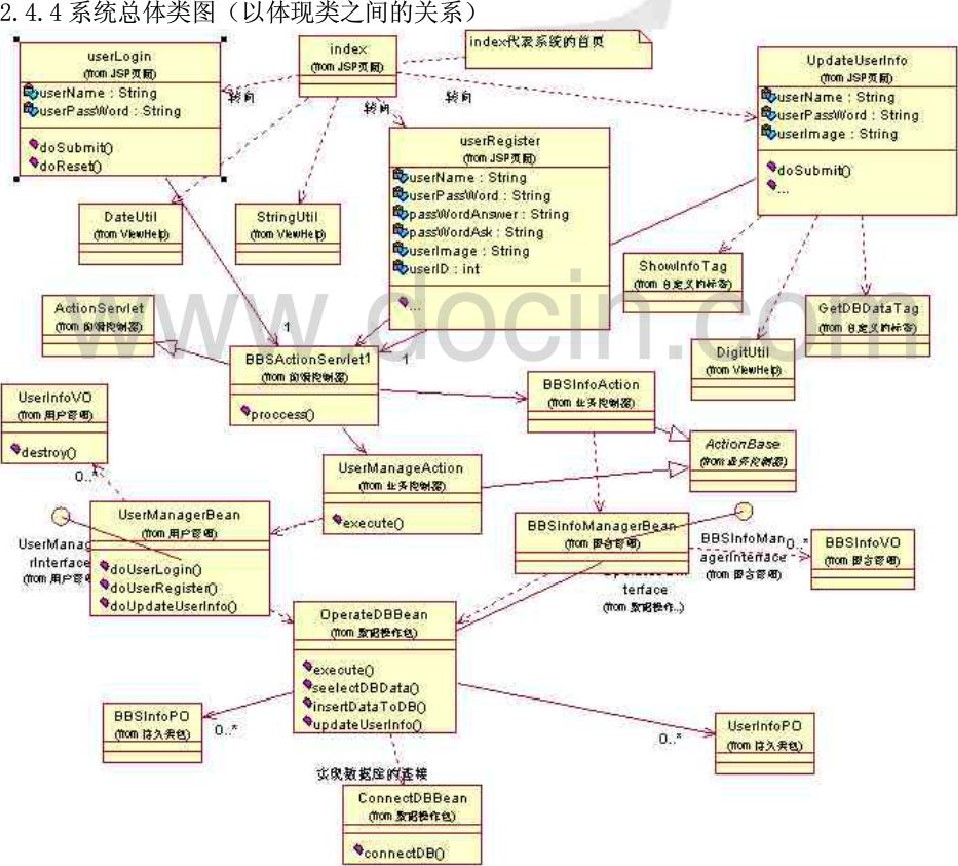
|  |
| --- |
| BBStitlelnf0VO  (from g |
|  |

|  |
| --- |
| AdminUserlnfoVO  (from |
|  |

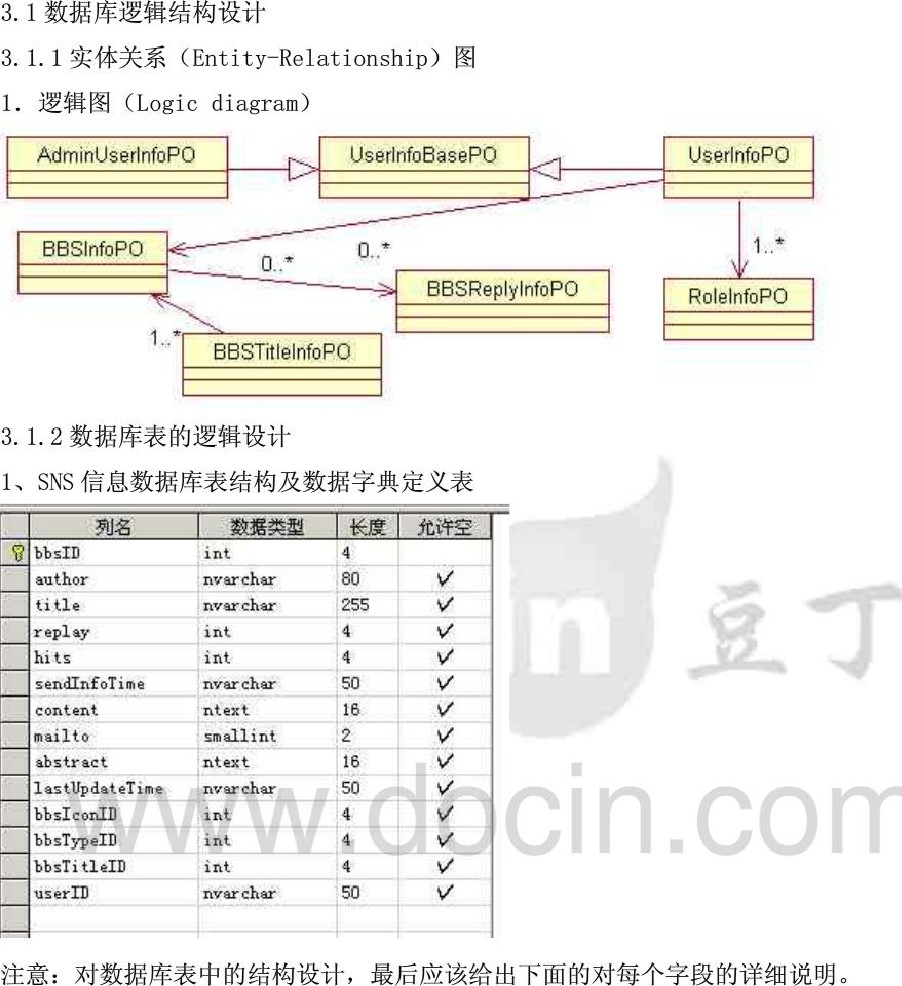
BBSTltlelnfoM anagerlnterface







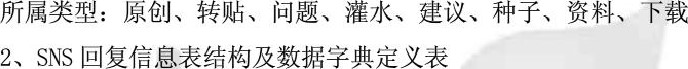




tin-corn

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | |  | |  | |  |
| SNS fitJ ID | |  | id | | int ( | |  | |  |
| SNS | |  | author | | vchar | |  | |  |
| SNS | |  | title | | vchar | |  | |  |
|  |  | |  |  | | ZT5titF)kJåi | |  | |
| ID | SNSID | |  | int 4 | | Not null | |  | |
|  | author | |  | nvarchar (20) | | Not null | |  | |
|  | SNSTit1e | |  | nvarchar (20) | | Not null | |  | |
|  | reply | |  | int 4 | | Not null | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | createTime | nvarchar (20) | Not null |  |
|  | | lastUpdateTim e | nvarchar (20) | Not null |  |
|  | | content | ntext (16) | Not null |  |
|  | | hi ts | int 4 | Not null |  |
| SNS | | iconID | int 4 |  |  |
|  | | SNSType1D | int 4 | Not null |  |
| FTrJdJfiEk ID | | boardID | int 4 | Not null |  |
| ID | | userID | int 4 | Not null |  |
|  | | SNSEI | int 4 |  |  |
|  | | SNSTop | int 4 |  |  |
|  |  | SNSLock | int 4 |  |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| ID | replyID | int 4 | Not null |  |
| ID | SNSID | int 4 | Not null |  |
|  | replyAuthor | nvarchar (20) | Not null |  |
|  | replyTime | nvarchar (20) | Not Ihull |  |
|  | replyIconID | int 4 |  |  |
|  | replyTitle | nvarchar (50) | Not null |  |
|  | replyContent | ntext | Not null |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | boardID | int 4 | Not null |  |
|  | boardTitle | nvarchar (20) | Not null |  |
| YdiEkFkJ\* | boardContent | nvarchar (255 | Not null |  |
| hJi± | leaderName | nvarchar (20) |  |  |
| MlJhJi± | secLeaderName | nvarchar (20) |  |  |
|  | all TopicNumber | int 4 |  |  |
|  | todayTopi cNumber | int 4 |  |  |
|  | lastReplyTime | nvarchar (20) | Not null |  |
|  | lastReplyAuthor | nvarchar (20) | Not null |  |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  | | | |
|  | | |  |
|
| userResume userSi | | |
| emailVizible  acceptÅdVi | | |

cinscorn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | LTftif)kJQ |  |
| ID | user ID | int 4 |  |  |
|  | username | nvarchar (20) |  |  |
| 1) | userSex | smallint (2) |  |  |
| 用户密码 | userPassWord | nvarchar(16) | 否 |  |
| 用户Email | userEmai ]一 | nvarchar（50） | 否 |  |
| 用户取回密码问题 | userPas sAsk | nvarchar（30） |  |  |
| 用户取回密码答案 | userPassAnswer | nvarchar（30） | 否 |  |
| 用户角色 | userType | tinyint (1) | 否 |  |
| 用户生日 | userBirthday | nvarchar(10) |  |  |
| 用户地址 | us erComeFrOln | nvarchar(50) |  |  |
| 注册时间 | userRegister | nvarchar（20） | 否 |  |
| 用户头像 | userSign | nvarchar(255) |  |  |
| QQ号 | userQQCode | nvarchar（15） |  |  |
| 用户签名档 | userIdiograph | ntext |  |  |
| 最后登录时间 | userLastTime | nvarchar（20） |  |  |
| 用户发表的主题数总和 | userTop i cCount | int 4 |  |  |
| 用户回复的主题数总和 | userReTop i cCount | int 4 |  |  |
| 用户被删的帖子总 | userDelTopicCount | int 4 |  |  |
| 用户被推荐精华帖总和 | userEliteTopicCou  fit | int 4 0 |  |  |
| 用户登录次数 | userLoadDegree | int 4 |  |  |
| 是否被禁言 | userLock | int 4 |  |  |

数据库设计要遵循一些规则，一个好的数据库满足一些严格的约束和要求。尽量分离各实，体对应的表，一个实体对应一个表，分析该实体有哪些属性，对应有些什么字段，以及各实体之间的联系。实体、属性与联系是进行概念设计时要考虑的三个元素，也是一个好的数据库设计的核心。