校友社区网站[需求分析 概要设计]

1. 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。

- 网站社区版块管理

  社区管理员可以添加、删除、调整网站版块。

- 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，也可以对本版块论坛中的文章进行置顶、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已建社区，注册用户可以查找已创建的班级社、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2-1）

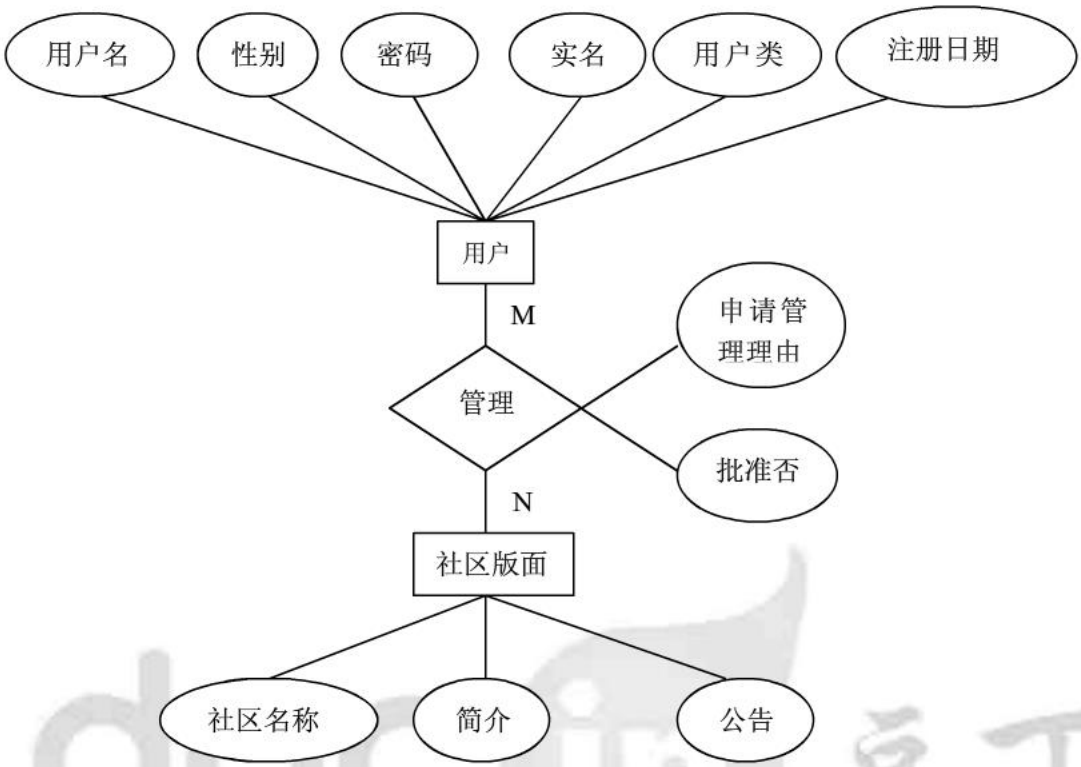
图 2-1

1.2 数据描述

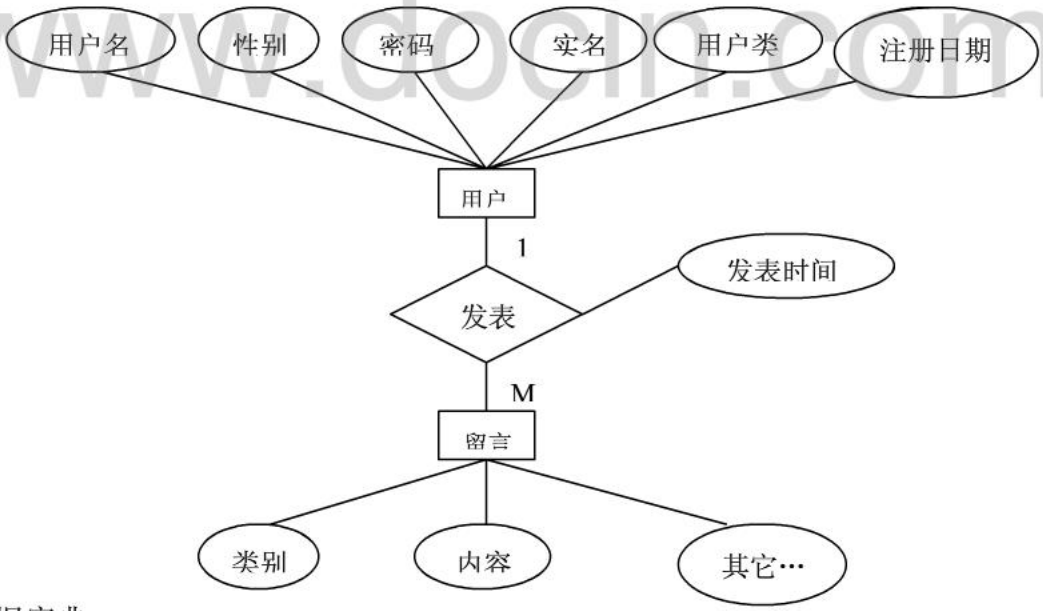
本系统实体之间主要有以下几类联系（Relation）:用户与网站页面之间存在管理联系（M：N)、用户与留言发表之间（1：M)、用户与友情链接（1：M)、用户与社区版块（1：M)、社区版面与内容（1：M)。

1.2.1 基本实体-联系图（ER 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。

1.3.1 网络需求

1）服务器要求能承载 1000 用户同时在线。

2）网络带宽要求 100MB 以上。

1.3.2 响应需求

1）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。

2）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。一般操作的响应时间应在 1-2 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。

3） 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——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 1GB 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 400MHz 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

1.4.4 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。

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，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，大部分 SNS 由教育机构、研究机构或商业机构管理，大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..JSP（JavaServer Pages）JSP 技术使用 Java 编程语言编写类 XML 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 MVC 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 ActionServlet/RequestProcessor 和我们定制的 Action,也为 V(View)提供了一系列定制标签（Custom Tag)。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

4.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）

2.1 需求规定

2.1.1 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。同时保证后台的管理页面和前台的服务页面保持风格的一致。

2.1.2 时间要求

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。

2.1.3 灵活性要求

灵活性:当用户需求，如操作方式，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 2-1

表 2-1 主机类型

类别	服务器标准配置
CPU	Intel i3 2340　2.0GHz 以上
内存	1G 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

1．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

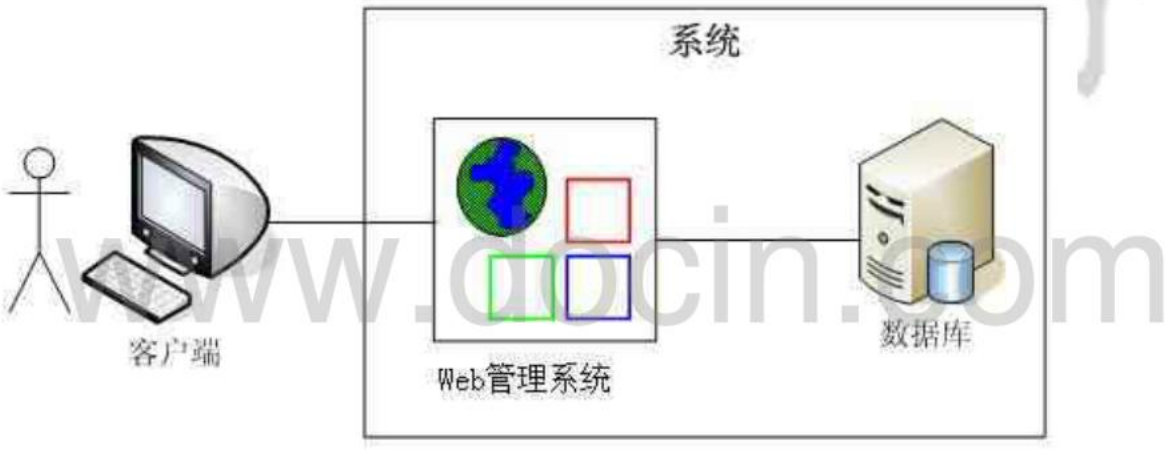
4．应用服务器：Tomcat 7.0 以上

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：



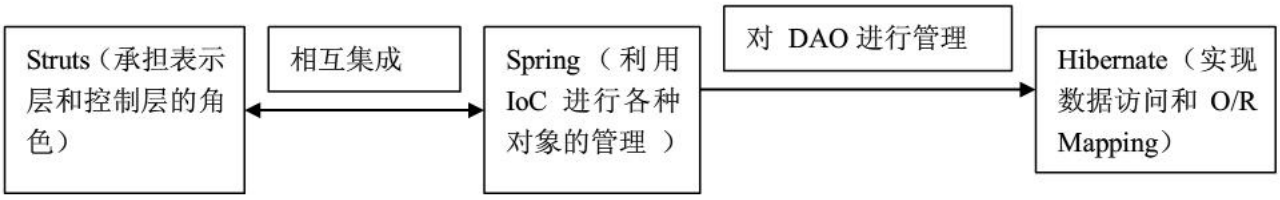
2、为什么对本项目要应用轻量级的框架技术

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

2.3.2 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



（1）对于表示层

经验表明，最好的方法是选择已存在的并已得到证明了的 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 Struts。

（2）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 接口的 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 IoC 和 AoP 的特性，将是实现业务逻辑层的更好选择。

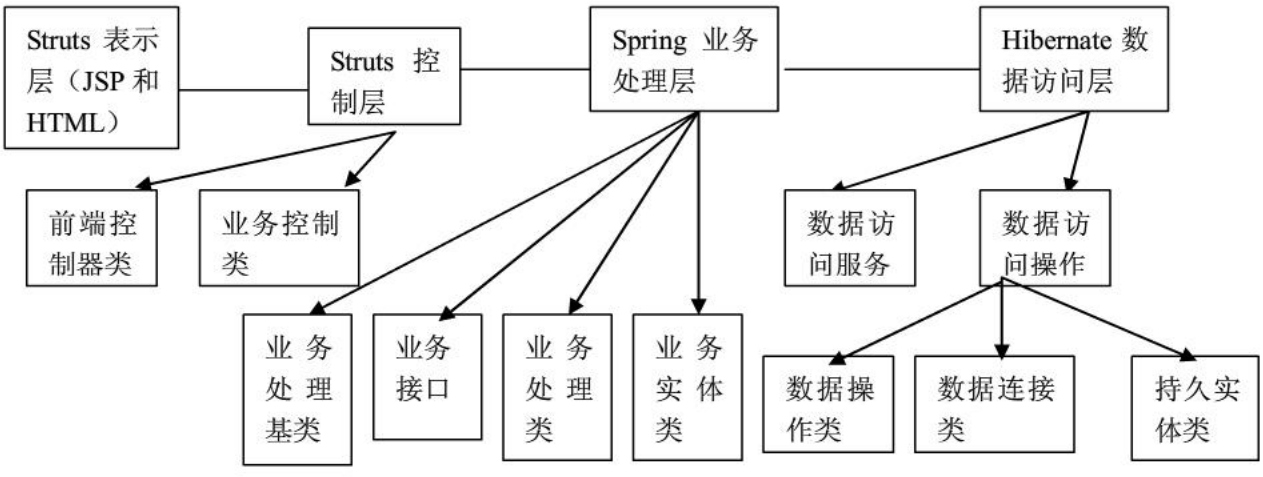
（3）在持久层中由于需要利用关系型数据库实现数据的持续化，但在应用中可以存在多种方法可用来实现：

JDBC：这是最为灵活的方法，然而，低级的 JDBC 难以使用，而且质量差的 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法，即 Entity bean 不是将应用与某种数据库类型而是 EJB 容器约束在一起。

O/R Mapping 框架：一个 ORM 框架采用以对象为中心的方法实现数据持续化，一个以对象为中心的应用易于开发并具有高度的可移植性----在该领域中存在几个框架可用—JDO、Hibernate、TopLink 以及 iBATIS 和 CocoBase 等。在本项目中我们选用 Hibernate。

2、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。

3、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，JDO 可替换 Hibernate。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，从而使得系统易于测试；便于移植；“开发-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。

- 前端控制层：　　ActionServlet 类，并且对它加以扩展。
- 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

业务处理基类：将各个业务功能模块中共同的部分抽象出，从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：针对应用系统中的各个数据库表提供对应的 POJO 类

2.3.3 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、MVC、成熟技术、ActionForm 技术等

2、业务逻辑层 Spring 框架完成如下工作

- 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；
- 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。

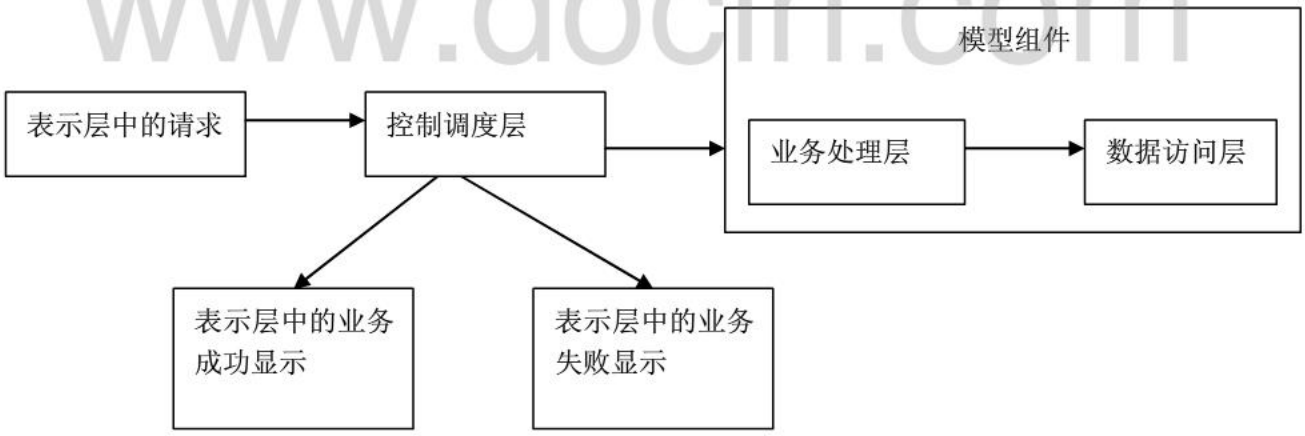
其它：

- 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”-----技术性的问题）
- IoC（对象的管理由容器完成）
- POJO（普通 JavaBean）-----不继承框架中某个类
- 容器服务（事务、数据库连接池）---Spring 中已经提供了
- 包装其它的框架（简化）

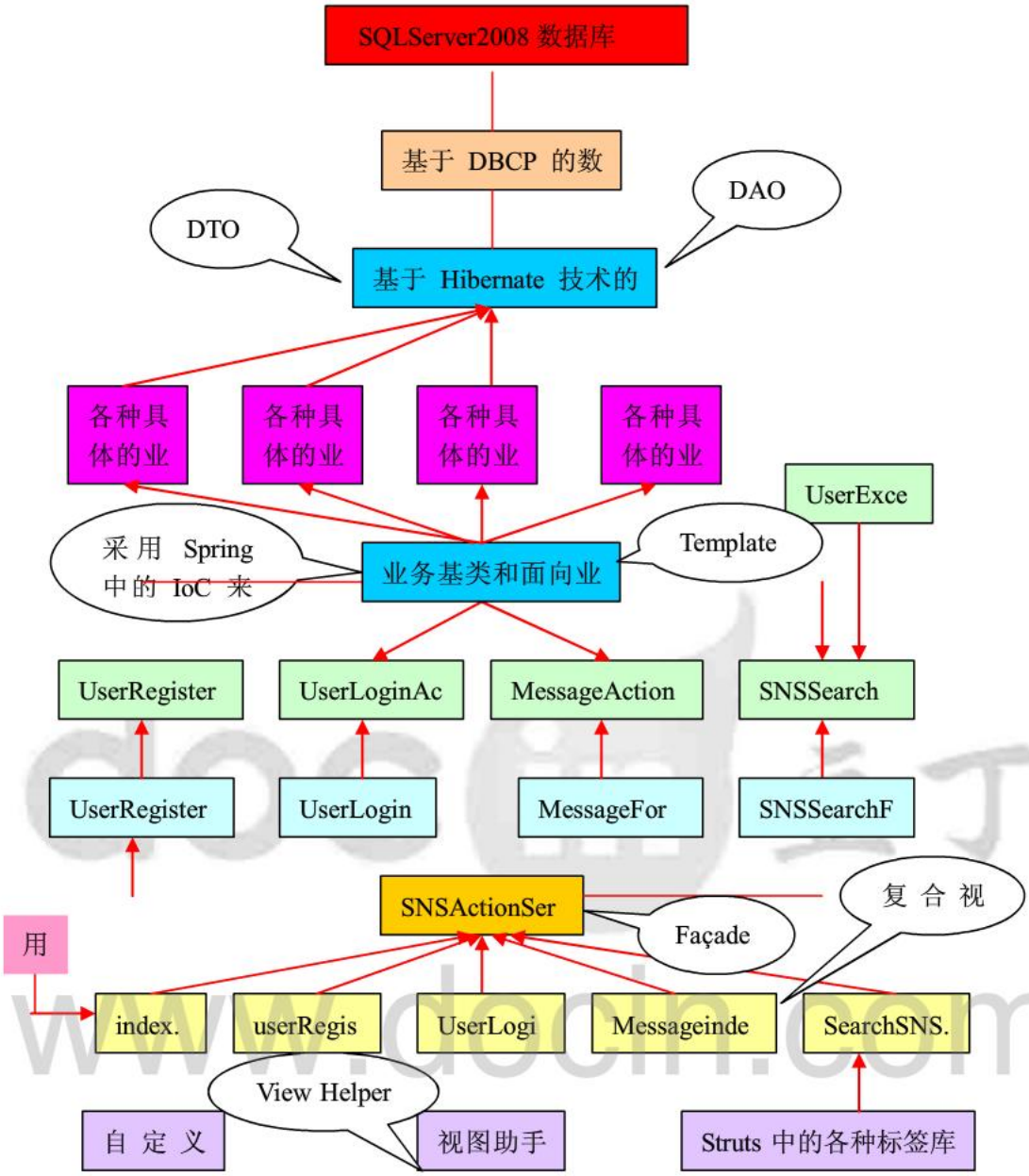
3、持久层 Hibernate 框架完成如下工作

- 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。

2.3.4 系统基于 MVC 设计

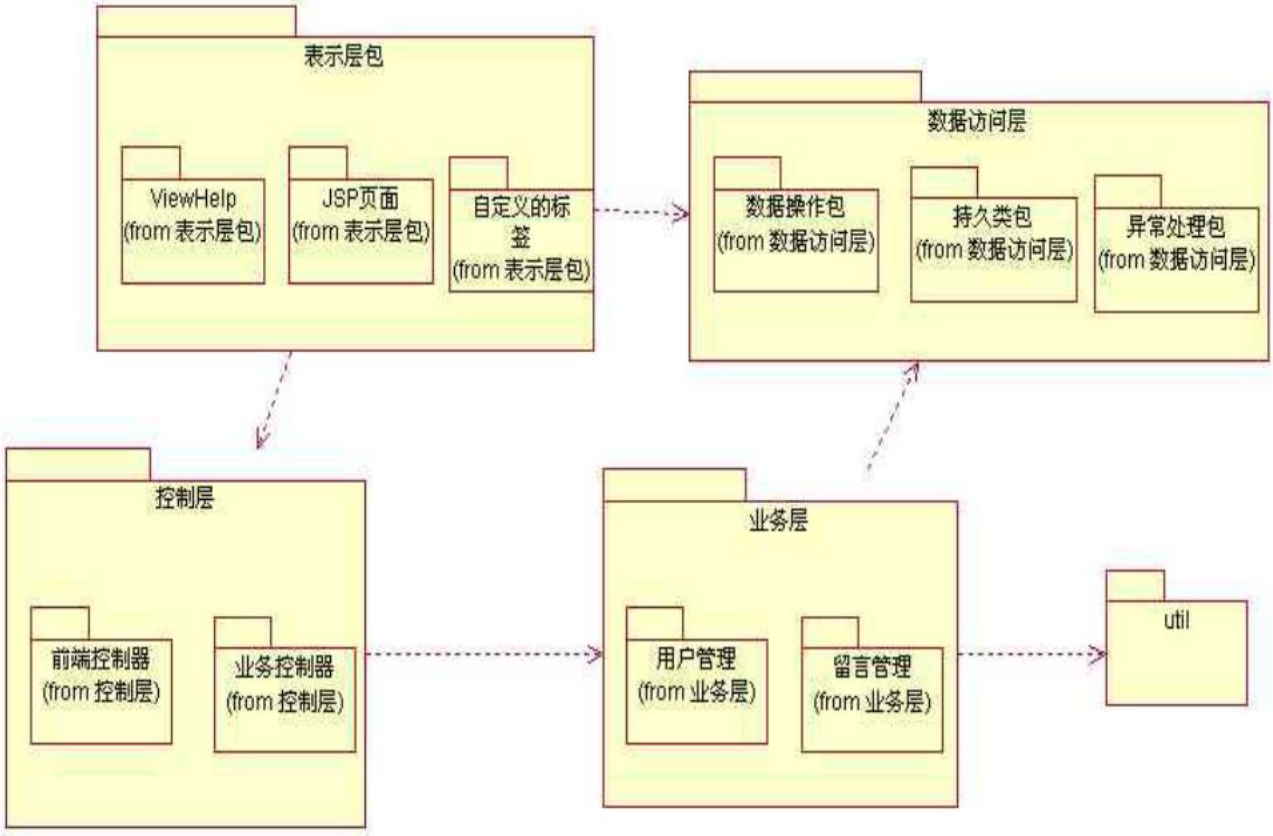


2.3.5 总体架构设计

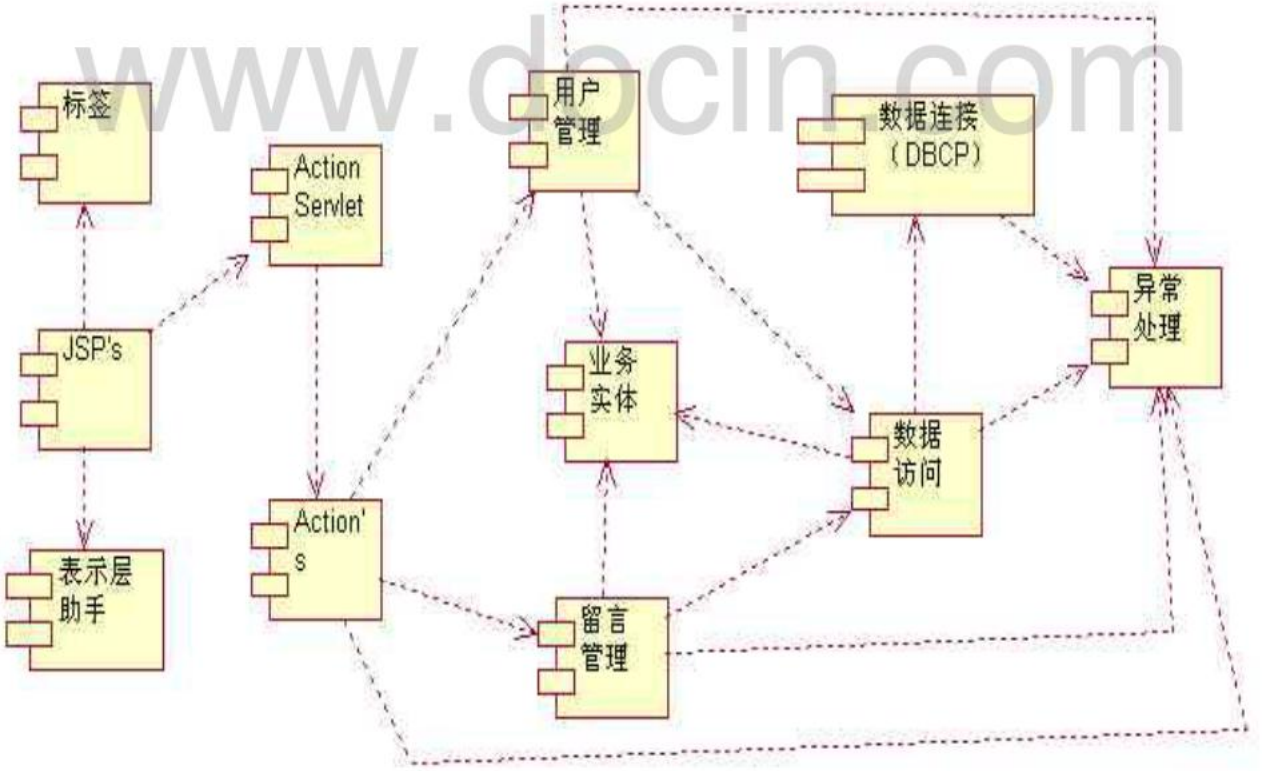

SQLServer2008 数据库
基于 DBCP 的数
DAO
DTO
基于 Hibernate 技术的
各种具体的业
各种具体的业
各种具体的业
各种具体的业
UserExce
采用 Spring 中的 IoC 来
Template
业务基类和面向业
UserRegister
UserLoginAc
MessageAction
SNSSearch
UserRegister
UserLogin
MessageFor
SNSSearchF
SNSActionSer
Façade
复合视
用
index.
userRegis
UserLogi
Messageinde
SearchSNS.
View Helper
自定义
视图助手
Struts 中的各种标签库

2.4 系统结构（系统各个组件设计）

2.4.1 体系结构包图（架构包图）



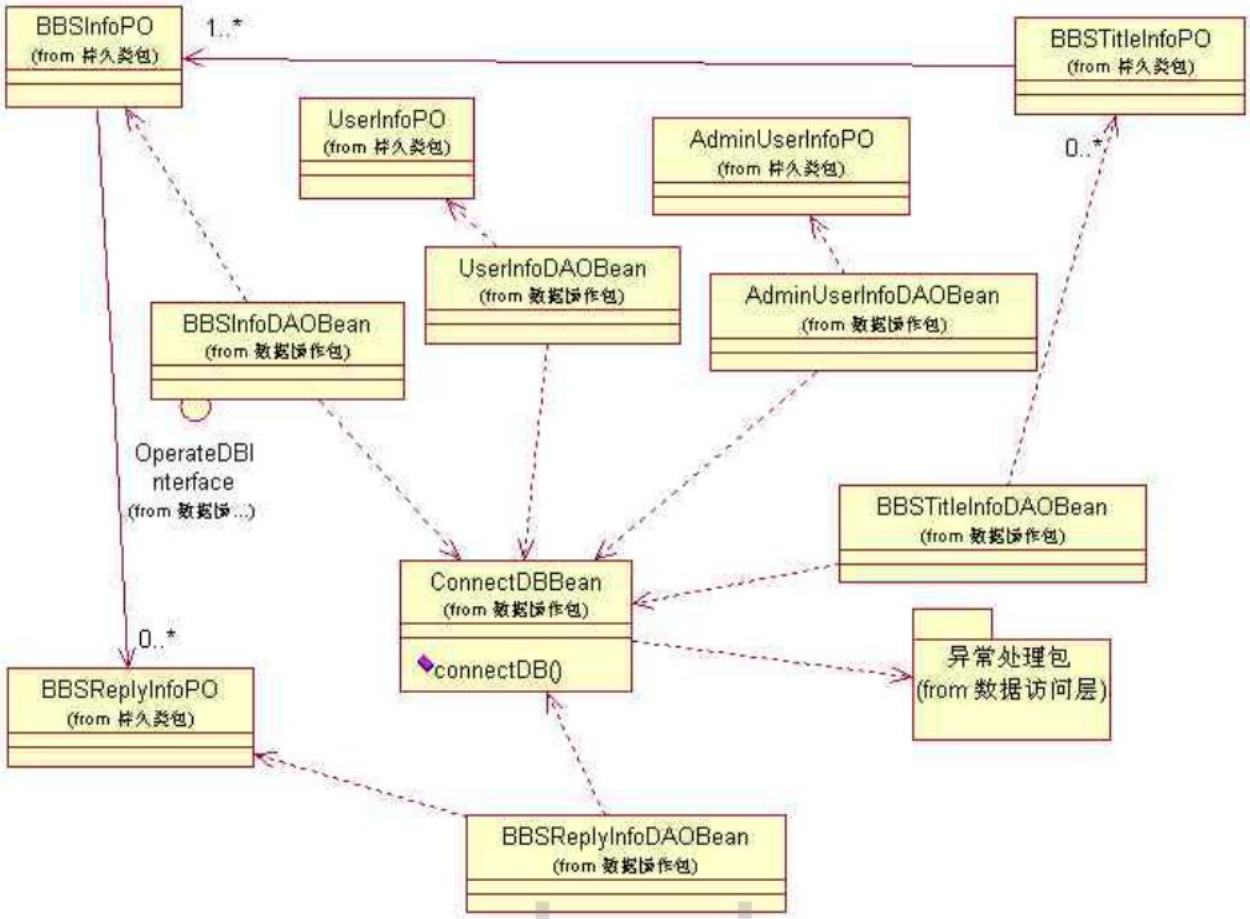
2.4.2 组件设计图（系统中的各个组件）



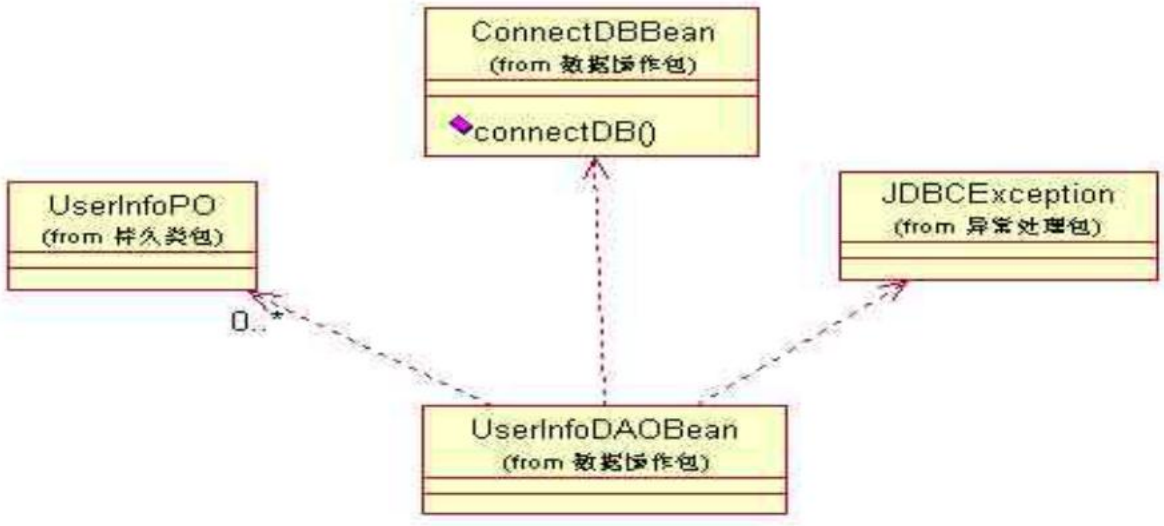
2.4.3 类图与接口设计（各个组件中的相关的类和接口）

1．数据访问层组件

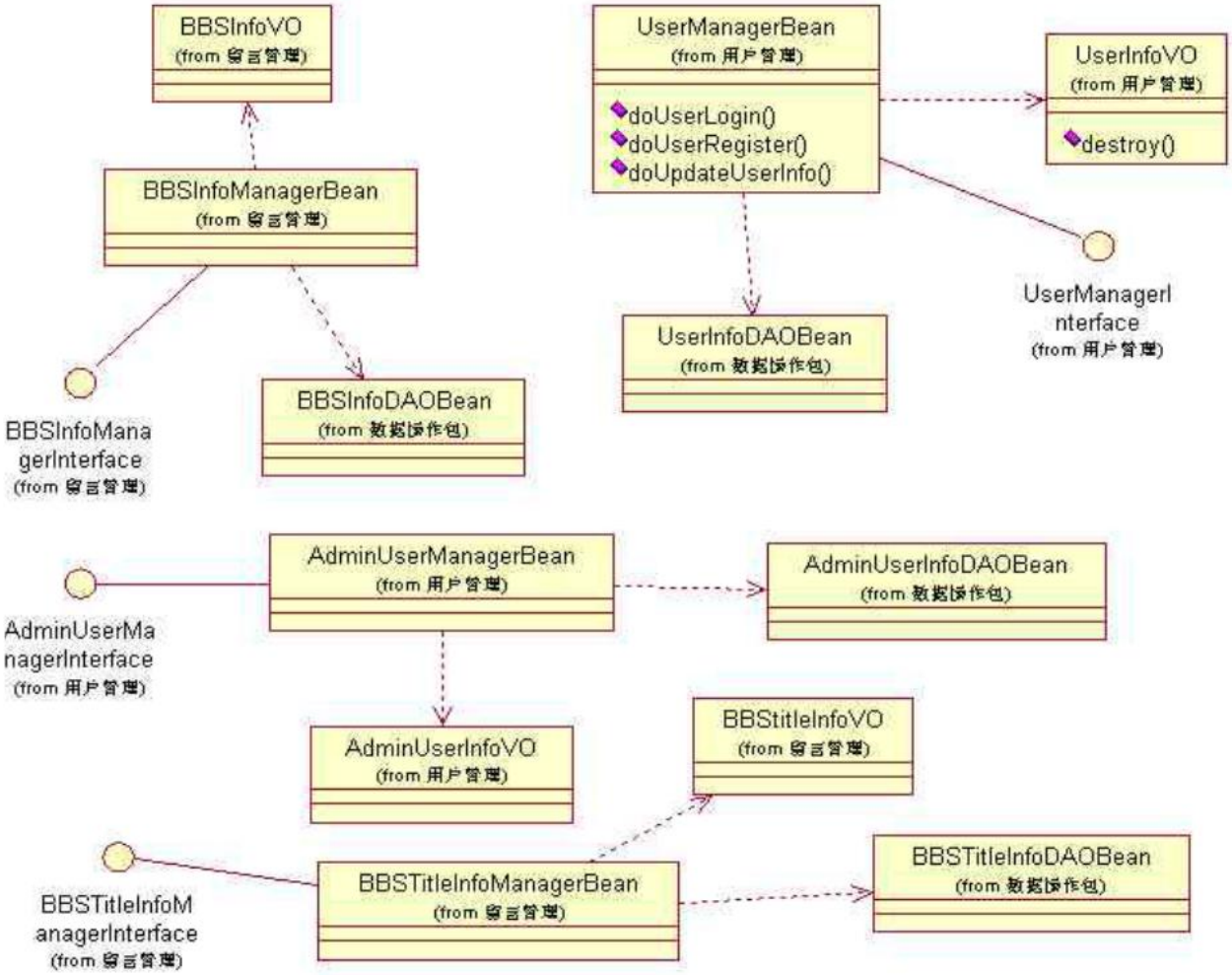
采用一个 DAO 组件实现数据访问操作



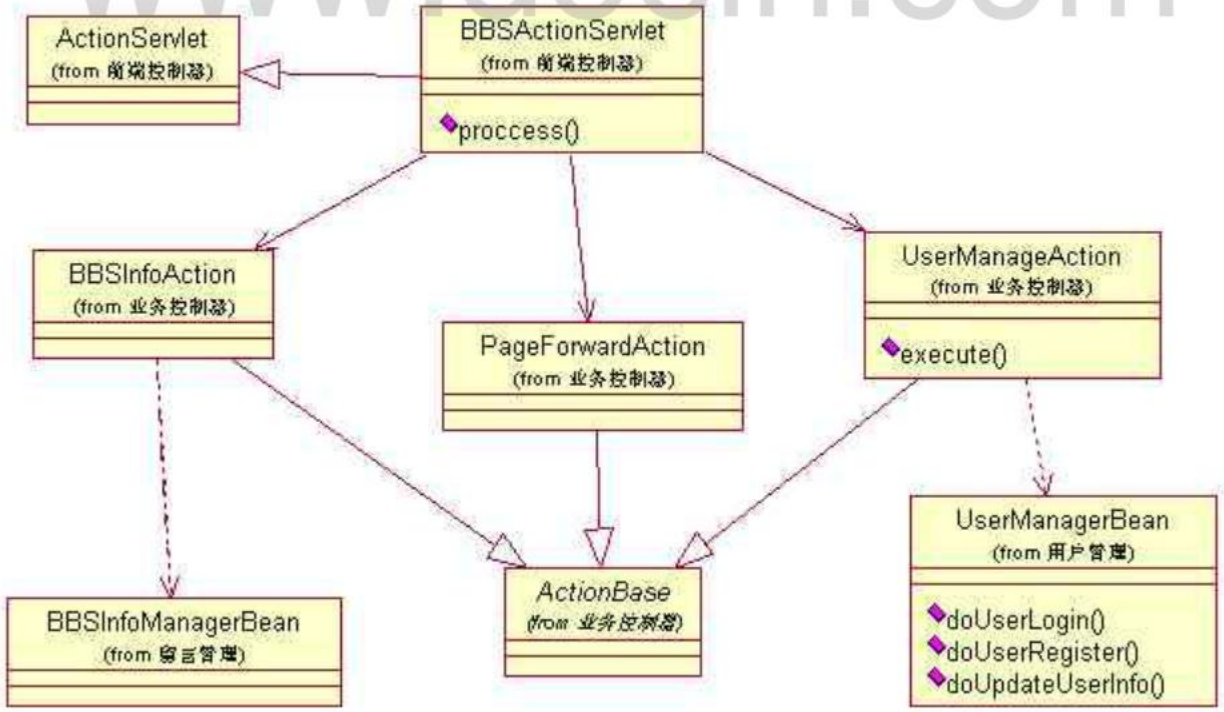
分别采用不同的 DAO 组件实现数据访问操作



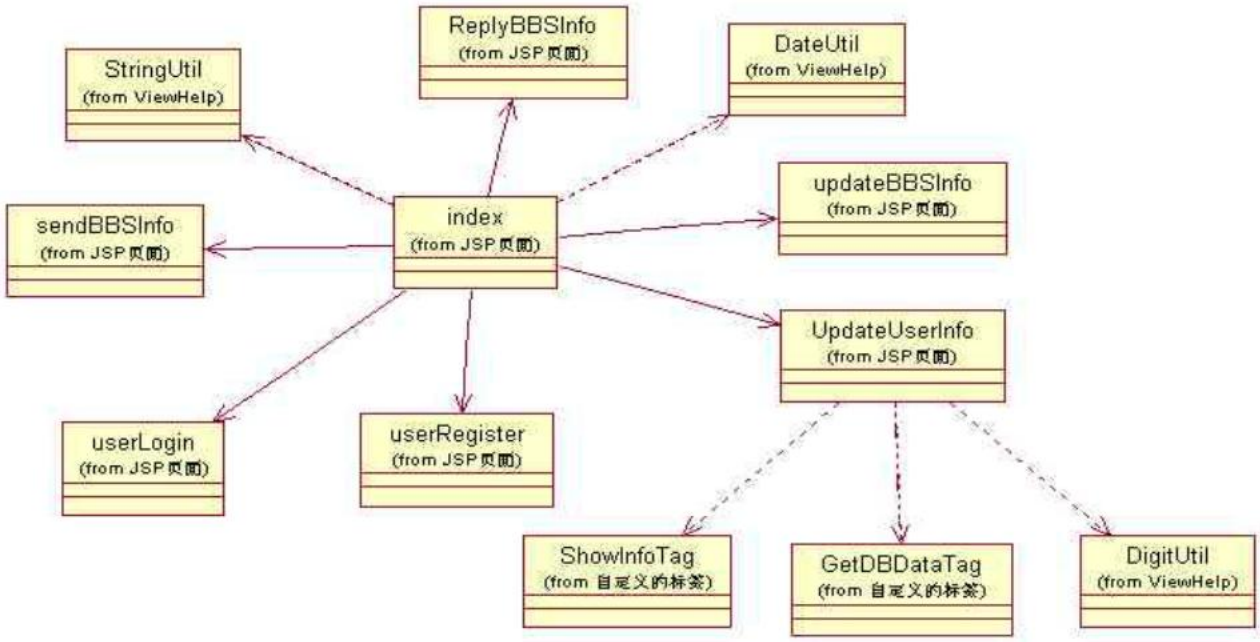
2、业务处理层组件



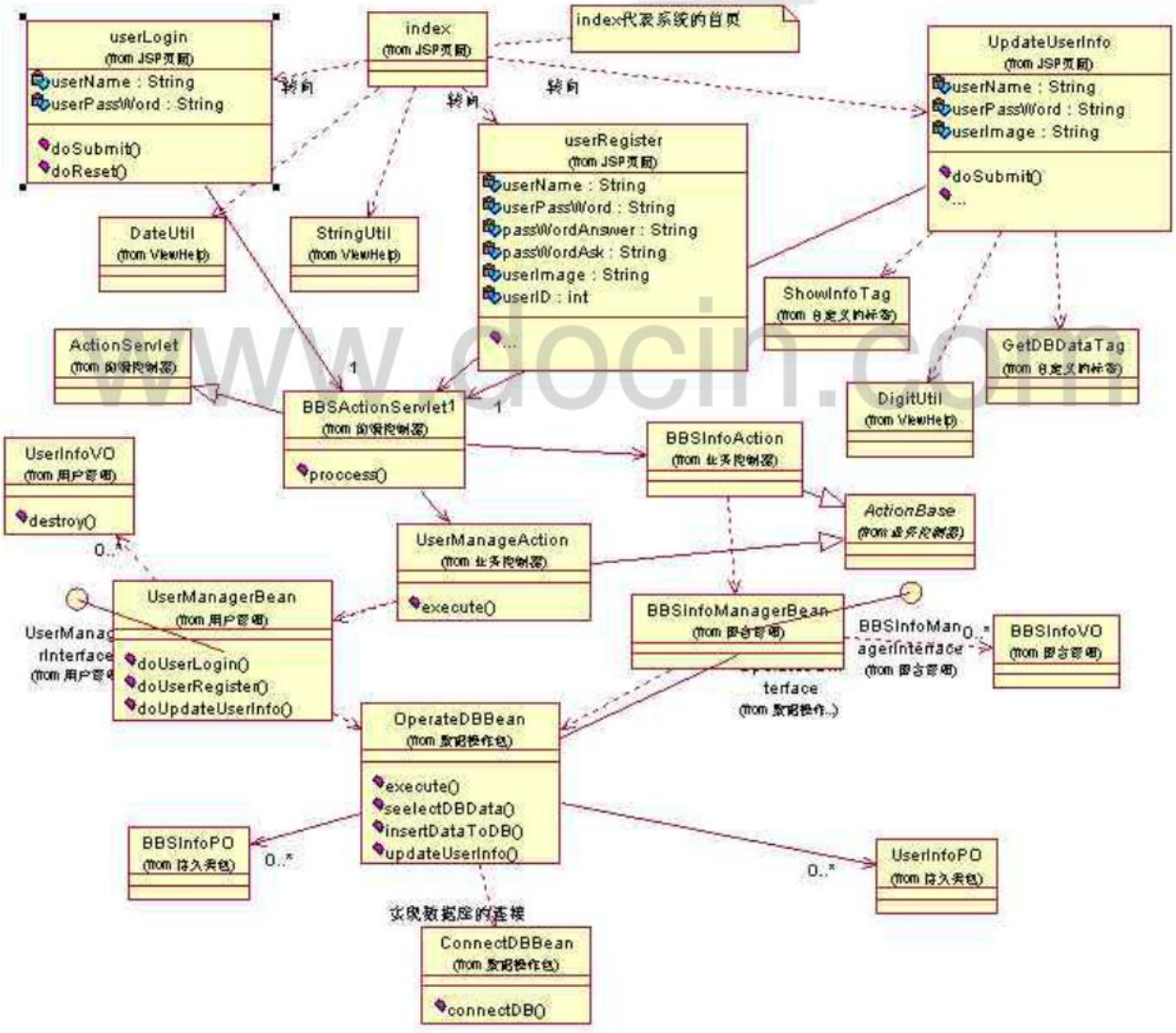
3、控制层组件



4．表示层组件



2.4.4 系统总体类图（以体现类之间的关系）

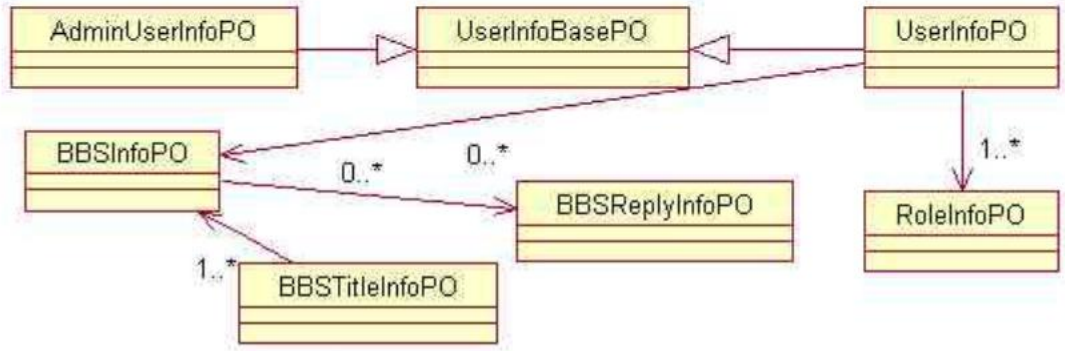


第三章 系统数据结构设计

3.1 数据库逻辑结构设计

3.1.1 实体关系（Entity-Relationship）图

1. 逻辑图（Logic diagram）



3.1.2 数据库表的逻辑设计

1、SNS 信息数据库表结构及数据字典定义表

	列名	数据类型	长度	允许空
🔑	bbsID	int	4	
	author	nvarchar	80	✓
	title	nvarchar	255	✓
	replay	int	4	✓
	hits	int	4	✓
	sendInfoTime	nvarchar	50	✓
	content	ntext	16	✓
	mailto	smallint	2	✓
	abstract	ntext	16	✓
	lastUpdateTime	nvarchar	50	✓
	bbsIconID	int	4	✓
	bbsTypeID	int	4	✓
	bbsTitleID	int	4	✓
	userID	nvarchar	50	✓

注意：对数据库表中的结构设计，最后应该给出下面的对每个字段的详细说明。

字 段 名	标 识 符	类型及长度	有无空值	主键
SNS 的 ID 标识	id	int（自动编号）	无	PK
SNS 作者	author	vchar	无	
SNS 标题	title	vchar	无	

字段名	标识符	类型及长度	是否允许为空	主键
主题 ID	SNSID	int 4	Not null	PK
主题作者	author	nvarchar(20)	Not null	
主题标题	SNSTitle	nvarchar(20)	Not null	
回复数	reply	int 4	Not null	

发表时间	createTime	nvarchar(20)	Not null	
最后回复时间	lastUpdateTime	nvarchar(20)	Not null	
内容	content	ntext(16)	Not null	
点击数	hits	int 4	Not null	
SNS 表情图片	iconID	int 4		
所属类型 ID	SNSTypeID	int 4	Not null	
所属版块 ID	boardID	int 4	Not null	
用户 ID	userID	int 4	Not null	
是否为精华帖	SNSElite	int 4		
是否置顶主题	SNSTop	int 4		
是否禁止回复	SNSLock	int 4		

所属类型：原创、转贴、问题、灌水、建议、种子、资料、下载

2、SNS 回复信息表结构及数据字典定义表

字段名	标识符	类型及长度	是否允许为空	主键
回复帖 ID	replyID	int 4	Not null	PK
所属主题 ID	SNSID	int 4	Not null	
回复作者	replyAuthor	nvarchar(20)	Not null	
回复时间	replyTime	nvarchar(20)	Not null	
回复表情 ID	replyIconID	int 4		
回复标题	replyTitle	nvarchar(50)	Not null	
回复内容	replyContent	ntext	Not null	

3、SNS 分类标题数据库表结构及数据字典定义表

	列名	数据类型	长度	允许空
🔑	bbsTitleID	int	4	
	bbsTitle	nvarchar	50	✓
	titleLeaderName	nvarchar	50	✓
	totalTopicNumber	int	4	✓
	todayTopicNumber	int	4	✓
	lastSendTime	nvarchar	50	✓
	lastTopicAuthor	nvarchar	50	✓
	newTopic	int	4	✓
	titleAbstractText	nvarchar	255	✓
	userID	nvarchar	50	✓

字段名	标识符	类型及长度	是否允许为空	主键
版块 ID	boardID	int 4	Not null	PK
版块题目	boardTitle	nvarchar(20)	Not null	
版块内容	boardContent	nvarchar(255)	Not null	
版主	leaderName	nvarchar(20)		
副版主	secLeaderName	nvarchar(20)		
版块主题总和	allTopicNumber	int 4		
版块当天帖子总和	todayTopicNumber	int 4		
最后回复的时间	lastReplyTime	nvarchar(20)	Not null	
最后回复的作者	lastReplyAuthor	nvarchar(20)	Not null	

4、用户信息数据库表结构及数据字典定义表

列名	数据类型	长度	允许空
userName	nvarchar	50	✓
userPassWord	nvarchar	50	✓
userType	int	4	✓
aliaoName	nvarchar	50	✓
passWordAsk	nvarchar	100	✓
userImage	nvarchar	100	✓
registerTime	nvarchar	50	✓
userID	nvarchar	50	
passWordAnswer	nvarchar	100	✓
userMail	nvarchar	50	✓
userSex	smallint	2	✓
userBirthDay	nvarchar	50	✓
userComeFrom	nvarchar	50	✓
userQQCode	nvarchar	50	✓
userICQCode	nvarchar	50	✓
userMSNCode	nvarchar	50	✓
userResume	ntext	16	✓
userSign	nvarchar	255	✓
emailVisible	smallint	2	✓
acceptAdvise	smallint	2	✓

字段名	标识符	类型及长度	是否允许为空	主键
用户 ID	userID	int 4	否	PK
用户名	username	nvarchar(20)	否	
用户性别(男 0，女 1)	userSex	smallint(2)	否	

用户密码	userPassWord	nvarchar(16)	否	
用户 Email	userEmail	nvarchar(50)	否	
用户取回密码问题	userPassAsk	nvarchar(30)	否	
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	userComeFrom	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)	否	
用户发表的主题数总和	userTopicCount	int 4		
用户回复的主题数总和	userReTopicCount	int 4		
用户被删的帖子总和	userDelTopicCount	int 4		
用户被推荐精华帖总和	userEliteTopicCount	int 4		
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。