**[同城校友网]**

**毕业论文**

专业：[计算机科学与技术]

班级：[2011级1班]

学生姓名：[张沛]

指导教师：[李磊军]

完成时间： 2015年4月12日



[1.摘要 5](#_Toc416640045)

[2.绪论 5](#_Toc416640046)

[2.1选题背景 5](#_Toc416640047)

[2.2国内外发展情况 6](#_Toc416640048)

[2.3研究意义 8](#_Toc416640049)

[2.4论文的组织结构 8](#_Toc416640050)

[2.5小结 8](#_Toc416640051)

[3系统需求分析 9](#_Toc416640052)

[3.1系统总体目标 9](#_Toc416640053)

[3.2功能需求 9](#_Toc416640054)

[3.3性能需求 9](#_Toc416640055)

[3.3.1数据精确度 9](#_Toc416640056)

[3.3.2页面设计 9](#_Toc416640057)

[3.3.3页面自适应性 10](#_Toc416640058)

[3.4：系统功能模块结构 10](#_Toc416640059)

[3.4.1功能结构图 10](#_Toc416640060)

[3.4.2后台管理系统 10](#_Toc416640061)

[3.4.3前台 11](#_Toc416640062)

[3.5数据需求（根据实体类来决定） 12](#_Toc416640063)

[3.5.1用户信息 12](#_Toc416640064)

[3.5.3学校信息 12](#_Toc416640065)

[3.5.3地址信息 13](#_Toc416640066)

[3.5.4群组信息 13](#_Toc416640067)

[3.5.5帖子信息 13](#_Toc416640068)

[3.7小结 13](#_Toc416640069)

[4数据库设计 13](#_Toc416640070)

[4.1数据库的概念 13](#_Toc416640071)

[4.2数据库的选择 14](#_Toc416640072)

[4.3数据库表的设计 14](#_Toc416640073)

[4.3.1 概念数据模型 14](#_Toc416640074)

[4.3.2 物理数据模型 15](#_Toc416640075)

[4.3.3 表格清单 15](#_Toc416640076)

[4.3.3 表格acct\_user 16](#_Toc416640077)

[4.3.4 表格fr\_college 16](#_Toc416640078)

[4.3.5 表格fr\_group 16](#_Toc416640079)

[4.3.6 表格fr\_message 17](#_Toc416640080)

[4.3.7 表格fr\_msgbox 18](#_Toc416640081)

[4.3.8 表格fr\_msgtype 18](#_Toc416640082)

[4.3.9 表格fr\_school 19](#_Toc416640083)

[4.3.10 表格fr\_user 19](#_Toc416640084)

[4.3.11表格fr\_user\_group 20](#_Toc416640085)

[4.3.12表格sys\_region 20](#_Toc416640086)

[5.系统开发要点 21](#_Toc416640087)

[5.1系统开发环境及开发工具的使用及介绍 21](#_Toc416640088)

[5.1.1eclipse 21](#_Toc416640089)

[5.1.2mysql 21](#_Toc416640090)

[5.1.3SVN 21](#_Toc416640091)

[5.1.4Maven 21](#_Toc416640092)

[5.1.5JDk1.7以上 21](#_Toc416640093)

[5.1.6tomcat7.0以上 21](#_Toc416640094)

[5.2系统开发相关技术 22](#_Toc416640095)

[5.2.1Html 22](#_Toc416640096)

[5.2.2Css 22](#_Toc416640097)

[5.2.3javascript 22](#_Toc416640098)

[5.2.4Servlet 23](#_Toc416640099)

[5.2.5Jsp 23](#_Toc416640100)

[5.2.6Mysql 24](#_Toc416640101)

[5.2.7JQuery 24](#_Toc416640102)

[5.2.8：Bootstrap 26](#_Toc416640103)

[5.2.9：nutz 26](#_Toc416640104)

[5.2.10：SeaJS 27](#_Toc416640105)

[5.3小结 27](#_Toc416640106)

[6系统实现 28](#_Toc416640107)

[6.1后台管理系统 28](#_Toc416640108)

[6.1.1用户管理 29](#_Toc416640109)

[6.1.2学校管理 31](#_Toc416640110)

[6.1.3地址管理 33](#_Toc416640111)

[6.1.4群组管理 36](#_Toc416640112)

[6.1.5帖子类型管理 37](#_Toc416640113)

[6.1.6帖子管理 39](#_Toc416640114)

[6.2前台 40](#_Toc416640115)

[6.2.1首页 41](#_Toc416640116)

[6.2.2全部群组 44](#_Toc416640117)

[6.2.3群组详情页 45](#_Toc416640118)

[6.2.4帖子详情页 49](#_Toc416640119)

[6.2.5注册页 52](#_Toc416640120)

[6.2.6登录页 55](#_Toc416640121)

[6.2.7个人中心 56](#_Toc416640122)

[6.3公共技术组件 60](#_Toc416640123)

[6.3.1分页组件 60](#_Toc416640124)

[6.3.2提示组件 61](#_Toc416640125)

[6.3.2基础的服务类 63](#_Toc416640126)

[6.4权限认证 63](#_Toc416640127)

[6.4.1 管理员登录 63](#_Toc416640128)

[6.4.2 前台用户登录 63](#_Toc416640129)

[7系统安装配置 64](#_Toc416640130)

[7.1 JDK安装 64](#_Toc416640131)

[7.2 tomcat安装 65](#_Toc416640132)

[7.3 mysql安装 65](#_Toc416640133)

[7.4 mysql客户端软件安装 78](#_Toc416640134)

[7.5 初始化数据 79](#_Toc416640135)

[结束语 80](#_Toc416640136)

[参考文献 80](#_Toc416640137)

[附录 80](#_Toc416640138)

[致谢 80](#_Toc416640139)

# 1.摘要

随着中国的社会迅速发展，越来越多的人走出家门，奔向各大城市努力奋斗，除了能获得更多的经济收入之外，还获得更多的工作机会和见识，基于这样的趋势，越来越多的学校同学加入这个洪流，在新的环境里，衣食住行都会在一个新的陌生的环境中慢慢成型，在新的环境中，由于各方面的不熟悉，身边也没有什么可以信赖的群体，导致会遇到各种问题，影响自己的发展。

本项目的开发目的，就是为了解决这样的学生群体遇到的问题，校友网主要是用于已毕业的校友之间的人际交流，尤其是同城之间的学习，工作经验交流，及生活等沟通，通过校友网这个社交网站达到日常交流。

**【关键词】：学生，陌生，交流，新环境，信赖，社交网站**

# 2.绪论

## 2.1选题背景

**【社交网络服务（亦称社交网站）】**主要作用是为一群拥有相同兴趣与活动的人创建在线社区。这类服务往往是基于互联网，为用户提供各种联系、交流的交互通路，如电子邮件、实时通信服务等。此类网站通常通过朋友，一传十、十传百地把网络展延开去，就像树叶的脉络，华语地区一般称之为“社交网站”。

多数社交网络会提供多种让用户交互起来的方式，可以为聊天、寄信、影音、文件分享、博客、新闻组等。

社交网络为信息的交流与分享提供了新的途径。作为社交网络的网站一般会拥有数以百万的登记用户，使用该服务已成为了用户们每天的生活。社交网络服务网站当前在世界上有许多，知名的包括Google+、Myspace、Plurk、Twitter、Facebook等。在中国大陆地区，社交网络服务为主的流行网站有人人网、QQ空间、百度贴吧、微博等。

## 2.2国内外发展情况

**【社交网络服务的发展】**在早期的互联网上，有许多提供用户间交互交谈的服务，例如：BBS，新闻组等。早期社交网络的服务网站呈现为在线社区的形式。用户多通过聊天室进行交流。

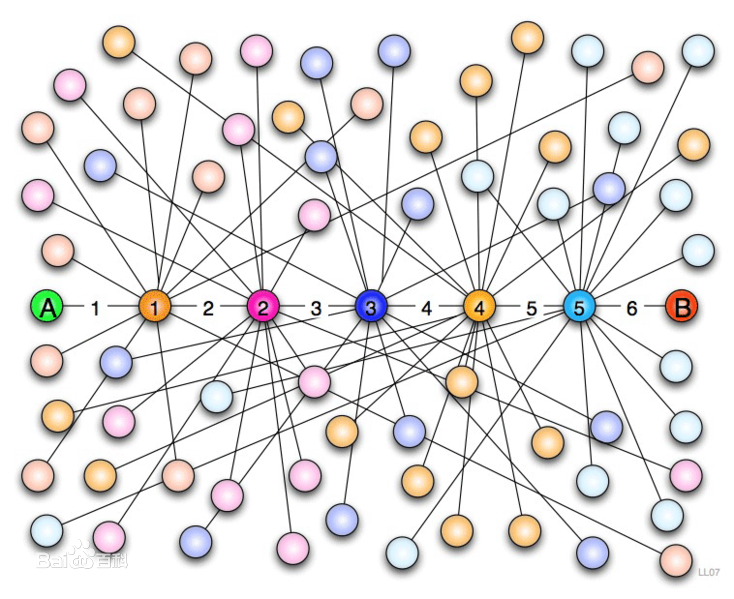
随着博客等新的网上交际工具的出现用户可以通过网站上创建的个人主页来分享喜爱的信息。

2002年至2004年间，世界上三大最受欢迎的社交网络服务类网站是Friendster、MySpace、Bebo。

在2005年之际MySpace成为了世上最巨大的社交网络服务类网站。传闻当时其页面访问量超越了作为著名搜索引擎的Google。

2006年第三方被允许开发基于Facebook的网站API的应用，使得Facebook随后一跃成为全球用户量增长最快的网站。有更多的网站随后加入自己或是第三方开发者的API。

社交服务网站的发展验证了**【六度分隔理论】**，即“人际关系脉络方面你必然可以通过不超出六位中间人间接与世上任意先生女士相识”。个体的社交圈会不断地扩大和重叠并在最终形成大的社交网络。在此类通过对“朋友的朋友是朋友”原则的实现而得到发展的线上社交网络中，Friendster具备一定的代表性。



但“朋友的朋友”，只是社交拓展的一种方式，而并非社交拓展的全部。因此，一般所谓的SNS，则其含义还远不止“朋友的朋友”这个层面。比如根据相同话题进行凝聚（如贴吧）、根据爱好进行凝聚（如Fexion网）、根据学习经历进行凝聚（如Facebook，人人网）、根据工作经历进行凝聚（如尚班族、大街网）、根据周末出游的相同地点进行凝聚、根据中国农民应用网络的方式凝聚（如农享网）等，都被纳入“SNS”的范畴。

社交服务类网站最早出现商业盈利目的是在2005年3月雅虎对雅虎360°的推出。在2005年6月新闻集团成功收购MySpace。随后在2005年12月，英国ITV购得Friends Reunited。此后在世界的各地涌现出各种不同语言的社交网络服务类网站。

**【社交网络的基本功能】**

* 纪录个人数据
* 私讯功能
* 用户相互链接的功能
* 用户检索的功能
* 日记（博客）的功能
* 社区的功能（包括公开的社区(open group)、不公开的社区(not open group)、秘密社区(closed group)等。）

**【社交网络的高级功能】**

* 提供地点打卡的功能。
* 可以针对某一议题设立可讨论的专页，惟其权限只限于开设版面的创版人与其所允许的管理员。

## 2.3研究意义

随着人们的沟通方式的多样化，信息化水平的不断提高，网络上出现了越来越多的各种各样的交流平台，有专业技术交流，也有日常生活的点点滴滴，例如，58同城，赶集类的网站侧重解决越来越多的人们工作难，就业难的问题，饿了么，最近火热的订餐网站，。。。。我们会发现越来越多的交流网站的目标在不断的细化，不断的切合实际，解决人们日益出现的问题，对于毕业生找工作的，或者马上也毕业，准备实习的同学来说，他们的压力也很大，除了要面临更大的毕业生多，就业岗位少的现实外，还要顶着各种没有工作经验，没有实践能力的公司苛刻，所以缓解大学生就业压力很重要，虽然有人人网等一些校内网站在解决各自问题，但是对于大部分毕业生的就业问题的的解决需要全面的把握！此社交网站在解决毕业生矛盾的同时，对于维护社会的安定也有潜在的意义

## 2.4论文的组织结构

**用visio来描述**

## 2.5小结

本绪论先从国内国外关于社交网络的发展做了概要的分析，然后抓住当前社会中部分社交网站涉及到的问题，和社会中日益凸显的毕业生就业难的矛盾关系，基于此，抓住这个矛盾，针对于这个问题，设计了这个面向解决这样的问题的社交网站，以此来弥补社会上已经出现的类似网站的不足，突出解决这样的社会问题！

# 3系统需求分析

## 3.1系统总体目标

实现用户在某个地区可以在所在的学校的群组中进行信息共享，和信息交流沟通，达到通过同校的较高信任度来完成在新的环境中快速适应和更好的工作和生活的目的，通过同校之间的互帮互助，让学校的温暖在需要的地方为需要的校友送去幸福，以此增进校友之间的友谊。

## 3.2功能需求

前台功能：用户可以注册登录此校友网，可以加入群组，退出群组，发表帖子，上传文件

后台功能：实现全国主要城市的学校的信息的采集和维护(lucene),角色管理，权限管理，用户管理，群组管理，帖子管理，文件管理

## 3.3性能需求

### 3.3.1数据精确度

在搜索时，可以进行模糊查询

省市显示的时候异步处理数据

### 3.3.2页面设计

操作界面简单明了，易于操作，对格式和数据类型限制的数据，进行验证，包括客户端验证和服务器验证，并采用错误提醒机制，提示用户输入正确数据和密码的 安全保密性

只有合法用户才能登录使用系统，对每个用户都有权限设置。对登录名、密码、以及用户重要信息进行加密，保证账号信息安全。 可维护性

### 3.3.3页面自适应性

采用自适应框架，页面在浏览器窗口大小改变的情况下，可以做出相应的调整，满足页面效果人性化的需求

## 3.4：系统功能模块结构

### 3.4.1功能结构图

### 3.4.2后台管理系统

后台管理系统，由系统管理员登陆后使用，主要针对系统各个模块数据进行统一维护、查看，包括用户管理，学校管理，地址管理。群组管理，帖子管理，帖子类型管理。

#### 3.4.2.1用户管理

查看已经注册用户列表信息，点击单个用户查看用户详细信息。统计总的注册用户数。

#### 3.4.2.3学校管理

查看所有学校列表信息，提供初始化学校功能，点击初始化学校，去新浪网站（<http://edu.sina.com.cn/gaokao/2014-08-08/1903430534.shtml>）采集全国高效名单：

#### 3.4.2.4地址管理

查看各个省份信息，包括编码，名称，经纬度，点击省份名称，查看该省份下的所有地市信息。

#### 3.4.2.5群组管理

查看所有群组列表信息，对群组进行审批操作，包括审批为活动群组，解散群组，关闭群组，解散和关闭群组的时候需要填写操作原因，并发送消息给群组创建人。

#### 3.4.2.6帖子管理

查看所有帖子列表信息，点击可以查看帖子详细内容。

#### 3.4.2.7帖子类型管理

查看所有帖子类型列表信息，增加、修改、删除帖子类别。

### 3.4.3前台

#### 3.4.3.1首页

网站首页，展示最新创建的若干群组，展示最热（按照帖子回复数量作为帖子热度的依据）的若干群组。显示最新发布的若干帖子。显示评论量最大的若干帖子。

#### 3.4.3.2群组列表首页

点击帖子的“全部群组”进入群组列表页面。该列表页，以列表形式展示所有群组。点击群组链接可以进入群组详情页面

#### 3.4.3.3群组详情页面

展示群组详细信息，包括群组创建信息，群组成员。该群组下的帖子列表。并提供发布帖子和加入（或者退出）该群组的链接功能。

#### 3.4.3.4帖子详情页

在任何地方点击帖子链接，进入该页面，该页面展示帖子详细信息，以及帖子的所有回复信息，并提供帖子的回复功能。

#### 3.4.3.5用户注册

用户在该页面输入详细注册信息，即可注册为网站用户。

#### 3.4.3.5用户登录

用户输入正确的登陆账号和密码，即可成功登录该网站，登录后举报发布帖子，回复帖子，加入群组等功能。

#### 3.4.3.6个人中心

用户登陆后可以看到个人中心链接，进入该页面，可以看到该用户在网站的所有信息，包括加入的群组，创建的群组，发布的帖子，查看以及完善用户个人信息。并具备修改密码和查看收件箱功能。

## 3.5数据需求（根据实体类来决定）

### 3.5.1用户信息

用户信息包括用户名，密码，昵称，真实姓名，Email，联系电话，QQ,所属学校，所属学院。

### 3.5.3学校信息

学校信息包括学校名称，地址。

### 3.5.3地址信息

地址信息包括地址编码，名称，经度，纬度。

### 3.5.4群组信息

群组信息包括群组名称，群组描述，创建者，床架日期，群组状态（审核中，活动群组，已解散，已关闭）。

### 3.5.5帖子信息

帖子信息包括标题，创建日期，发帖人、帖子类型。

## 3.7小结

通过对系统的整体把握，系统的大致框架和数据来源即最终的系统功能效果基本成型，为后期的系统实现绘制蓝图

# 4数据库设计

## 4.1数据库的概念

数据库（Database）是按照数据结构来组织、存储和管理数据的仓库，它产生于距今六十多年前，随着信息技术和市场的发展，特别是二十世纪九十年代以后，数据管理不再仅仅是存储和管理数据，而转变成用户所需要的各种数据管理的方式。数据库有很多种类型，从最简单的存储有各种数据的表格到能够进行海量数据存储的大型数据库系统都在各个方面得到了广泛的应用。

在信息化社会，充分有效地管理和利用各类信息资源，是进行科学研究和决策管理的前提条件。数据库技术是管理信息系统、办公自动化系统、决策支持系统等各类信息系统的核心部分，是进行科学研究和决策管理的重要技术手段。

## 4.2数据库的选择

本系统采用MySql搭建关系型数据库

MySQL是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策（本词条“授权政策”），它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。由于其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境。

与其他的大型数据库例如 Oracle、DB2、SQL Server等相比，MySQL 自有它的不足之处，但是这丝毫也没有减少它受欢迎的程度。对于一般的个人使用者和中小型企业来说，MySQL提供的功能已经绰绰有余，而且由于 MySQ L是开放源码软件，因此可以大大降低总体拥有成本。Linux作为操作系统，Apache和 Nginx作为 Web 服务器，MySQL 作为数据库，PHP/Perl/Python作为服务器端脚本解释器。由于这四个软件都是免费或开放源码软件（FLOSS)，因此使用这种方式不用花一分钱（除开人工成本）就可以建立起一个稳定、免费的网站系统，被业界称为“LAMP“或“LNMP”组合。

## 4.3数据库表的设计

### 4.3.1 概念数据模型

简称概念模型(Conceptual Data Model)，是面向数据库用户的现实世界的模型，主要用来描述世界的概念化结构，它使数据库的设计人员在设计的初始阶段，摆脱计算机系统及DBMS的具体技术问题，集中精力分析数据以及数据之间的联系等，与具体的数据库管理系统（Database Management System，简称DBMS）无关。概念数据模型必须换成逻辑数据模型，才能在DBMS中实现。

其只描述信息的特征和强调语义，而不涉及信息在计算机中的表示，是现实世界到信息世界的第一层抽象。最常用的是实体联系模型（Entity Relationship Model)。

### 4.3.2 物理数据模型

物理数据模型（Physical Data Model）PDM，提供了系统初始设计所需要的基础元素，以及相关元素之间的关系。即用于存储结构和访问机制的更高层描述，描述数据是如何在计算机中存储的，如何表达记录结构、记录顺序和访问路径等信息。使用物理数据模型，可以在系统层实现数据库。数据库的物理设计阶段必须在此基础上进行详细的后台设计，包括数据库的存储过程、操作、触发、视图和索引表等；至2013年，还没有太多的物理数据模型，最常用的物理数据模型如下：

统一模型。

框架存储模型。

### 4.3.3 表格清单

* 考虑到数据库是在项目的开发中不断的完善的，所以就没有采用powerdesigner工具来设计数据库，后期可以在实体关系完全确定后用powerdesigner来设计数据库
* 这里数据库表的设计几乎都是和项目中的Java实体Bean对象一一对应的，所以以下表的内容会把数据库中的表和实体对象纳入到一起来对比显示
* 注：是否为空列项：(0表示不允许为空，-1允许为空）

### 4.3.3 表格acct\_user

#### 4.3.3.1表格acct\_user的卡片

|  |  |
| --- | --- |
| 名称 | acct\_user |
| 注释 | 管理员表 |

#### 4.3.3.2表格acct\_user的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id |  | bigint(20) | 20 |
| login\_name | 管理员帐号 | varchar(64) | 64 |
| name | 管理员姓名 | varchar(64) | 64 |
| password | 管理员密码 | varchar(255) | 255 |
| register\_date | 注册日期 | timestamp |  |

### 4.3.4 表格fr\_college

#### 4.3.4.1 表格fr\_college的卡片

|  |  |
| --- | --- |
| 名称 | fr\_college |
| 注释 | 学校表 |

#### 4.3.4.2 表格fr\_college的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id |  | bigint(20) | 20 |
| cname | 学校名称 | varchar(32) | 32 |
| addr\_code | 所属地区编码 | varchar(6) | 6 |
| addr | 所属地区名称 | varchar(62) | 62 |

### 4.3.5 表格fr\_group

#### 4.3.5.1 表格fr\_group的卡片

|  |  |
| --- | --- |
| 名称 | fr\_group |
| 注释 | 群组表 |

#### 4.3.5.2 表格fr\_group的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id |  | bigint(20) | 20 |
| gname | 群组名称 | varchar(32) | 32 |
| descript | 描述 | varchar(512) | 512 |
| status | 群组状态：1：活动小组；2：已关闭;3:已关闭 | int(11) | 11 |
| college | 所属学校 | bigint(20) | 20 |
| addr\_code | 群组所在地区编码 | bigint(20) | 20 |
| createtime | 创建日期 | date |  |
| creator\_id | 创建者id | bigint(20) | 20 |

### 4.3.6 表格fr\_message

#### 4.3.6.1 表格fr\_message的卡片

|  |  |
| --- | --- |
| 名称 | fr\_message |
| 注释 | 帖子表 |

#### 4.3.6.2 表格fr\_message的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id |  | bigint(20) | 20 |
| title | 帖子标题 | varchar(128) | 128 |
| descript | 帖子内容 | text |  |
| createtime | 发布日期 | date |  |
| status | 状态：0：发布中；1：已关闭 | int(11) | 11 |
| tid | 所属消息类型id | bigint(20) | 20 |
| uid | 发布者id | bigint(20) | 20 |
| gid | 所属群组id | bigint(20) | 20 |
| replay\_id | 如果该字段有值，说明该消息是一个回复消息，该字段表示回复的那个消息的id | bigint(20) | 20 |
| istop | 是否置顶：0是，1：不制定 | int(11) | 11 |

### 4.3.7 表格fr\_msgbox

#### 4.3.7.1 表格fr\_msgbox的卡片

|  |  |
| --- | --- |
| 名称 | fr\_msgbox |
| 注释 | 收件箱 |

#### 4.3.7.2 表格fr\_msgbox的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id |  | bigint(20) | 20 |
| send\_user | 发送用户 | varchar(32) | 32 |
| send\_date | 发送日期 | date |  |
| if\_read | 是否读过，0：未读，1：已读 | int(11) | 11 |
| receive\_user | 消息接受者 | bigint(20) | 20 |
| descript | 描述 | varchar(512) | 512 |

### 4.3.8 表格fr\_msgtype

#### 4.3.8.1 表格fr\_msgtype的卡片

|  |  |
| --- | --- |
| 名称 | fr\_msgtype |
| 注释 | 消息类型 |

#### 4.3.8.2 表格fr\_msgtype的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id |  | bigint(20) | 20 |
| name | 消息类型名称 | varchar(32) | 32 |
| descript | 描述 | text |  |

### 4.3.9 表格fr\_school

#### 4.3.9.1 表格fr\_school的卡片

|  |  |
| --- | --- |
| 名称 | fr\_school |
| 注释 | 院系表 |

#### 4.3.9.2 表格fr\_school的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id |  | bigint(20) | 20 |
| sname | 院系名称 | varchar(32) | 32 |
| cid | 所属学校id | bigint(20) | 20 |

### 4.3.10 表格fr\_user

#### 4.3.10.1 表格fr\_user的卡片

|  |  |
| --- | --- |
| 名称 | fr\_user |
| 注释 | 用户表 |

#### 4.3.10.2 表格fr\_user的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id |  | bigint(20) | 20 |
| uname | 帐号 | varchar(32) | 32 |
| pwd | 密码 | varchar(32) | 32 |
| nick\_name | 昵称 | varchar(32) | 32 |
| real\_name | 真实姓名 | varchar(32) | 32 |
| email | Email | varchar(64) | 64 |
| tel | 电话 | varchar(16) | 16 |
| qq | Qq | varchar(16) | 16 |
| cid | 所属学校id | bigint(20) | 20 |
| sid | 所属院系id | bigint(20) | 20 |

### 4.3.11表格fr\_user\_group

#### 4.3.11.1 表格fr\_user\_group的卡片

|  |  |
| --- | --- |
| 名称 | fr\_user\_group |
| 注释 | 用户群组关系表 |

#### 4.3.11.2 表格fr\_user\_group的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id |  | bigint(20) | 20 |
| fr\_group\_id | 群组id | bigint(20) | 20 |
| fr\_user\_id | 用户id | bigint(20) | 20 |

### 4.3.12表格sys\_region

#### 4.3.12.1 表格sys\_region的卡片

|  |  |
| --- | --- |
| 名称 | sys\_region |
| 注释 | 地区编码表 |

#### 4.3.12.2表格sys\_region的列清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 注释 | 数据类型 | 长度 |
| id | 地区编码，后四位为0的地区是省份 | bigint(20) | 20 |
| name | 地区名称 | varchar(100) | 100 |
| description | 描述 | varchar(128) | 128 |
| lng | 经度 | varchar(255) | 255 |
| lat | 纬度 | varchar(255) | 255 |

# 5.系统开发要点

## 5.1系统开发环境及开发工具的使用及介绍

### 5.1.1eclipse

开源，轻便，可以以插件的方式方便加入和去除需要的插件功能

### 5.1.2mysql

开源，小巧，功能较齐全的数据库

### 5.1.3SVN

项目版本控制管理工具

Svn（Subversion）是近年来崛起的版本管理工具，在当前的开源项目里(J2EE)，几乎95%以上的项目都用到了 SVN。Subversion 项目的初衷是为了替换当年开源社区最为流行的版本控制软件 CVS，在 CVS的功能的基础上有很多的提升同时也能较好的解决 CVS 系统的一些不足。

### 5.1.4Maven

翻译为“专家”，“内行”

Maven是跨平台的项目管理工具。主要服务于基于Java平台的项目构建，依赖管理和项目信息管理。

### 5.1.5JDk1.7以上

### 5.1.6tomcat7.0以上

## 5.2系统开发相关技术

### 5.2.1Html

超文本标记语言或超文本链接标示语言（标准通用标记语言下的一个应用）HTML（HyperText Mark-up Language）是一种制作万维网页面的标准语言，是万维网浏览器使用的一种语言，它消除了不同计算机之间信息交流的障碍。

它是目前网络上应用最为广泛的语言，也是构成网页文档的主要语言。HTML文件是由HTML命令组成的描述性文本，HTML命令可以说明文字、图形、动画、声音、表格、链接等。HTML文件的结构包括头部（Head）、主体（Body）两大部分，其中头部描述浏览器所需的信息，而主体则包含所要说明的具体内容。

优点：打开速度快，操作简单，无需下载其他阅读器。

缺点：改变格式，无法直接引用，界面不友好，阅读不方便。

### 5.2.2Css

级联样式表是一种用来表现HTML（标准通用标记语言的一个应用）或XML（标准通用标记语言的一个子集）等文件样式的计算机语言。

CSS目前最新版本为CSS3，是能够真正做到网页表现与内容分离的一种样式设计语言。相对于传统HTML的表现而言，CSS能够对网页中的对象的位置排版进行像素级的精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力，并能够进行初步交互设计，是目前基于文本展示最优秀的表现设计语言。CSS能够根据不同使用者的理解能力，简化或者优化写法，针对各类人群，有较强的易读性。

### 5.2.3javascript

JavaScript一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器被称为JavaScript引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在HTML（标准通用标记语言下的一个应用）网页上使用，用来给HTML网页增加动态功能。

在1995年时，由Netscape公司的Brendan Eich，在网景导航者浏览器上首次设计实现而成。因为Netscape与Sun合作，Netscape管理层希望它外观看起来像Java，因此取名为JavaScript。但实际上它的语法风格与Self及Scheme较为接近。

为了取得技术优势，微软推出了JScript，CEnvi推出ScriptEase，与JavaScript同样可在浏览器上运行。为了统一规格，因为JavaScript兼容于ECMA标准，因此也称为ECMAScript。

是一种解释性脚本语言（代码不进行预编译）。

主要用来向HTML（标准通用标记语言下的一个应用）页面添加交互行为。

可以直接嵌入HTML页面，但写成单独的js文件有利于结构和行为的分离。

跨平台特性，在绝大多数浏览器的支持下，可以在多种平台下运行（如Windows、Linux、Mac、Android、iOS等）。

### 5.2.4Servlet

[servlet](http://baike.baidu.com/view/25169.htm) 是在[服务器](http://baike.baidu.com/view/899.htm)上运行的小[程序](http://baike.baidu.com/view/17674.htm)。这个词是在 Java [applet](http://baike.baidu.com/view/150834.htm) 的环境中创造的。虽然后者已很少被使用，但 servlet 却发展的很好。是一般面试都会常考的知识

### 5.2.5Jsp

JSP全名为Java Server Pages，中文名叫java服务器页面，其根本是一个简化的Servlet设计，它[1] 是由Sun Microsystems公司倡导、许多公司参与一起建立的一种动态网页技术标准。JSP技术有点类似ASP技术，它是在传统的网页HTML（标准通用标记语言的子集）文件(\*.htm,\*.html)中插入Java程序段(Scriptlet)和JSP标记(tag)，从而形成JSP文件，后缀名为(\*.jsp)。 用JSP开发的Web应用是跨平台的，既能在Linux下运行，也能在其他操作系统上运行。

它实现了Html语法中的java扩张（以 <%, %>形式）。JSP与Servlet一样，是在服务器端执行的。通常返回给客户端的就是一个HTML文本，因此客户端只要有浏览器就能浏览。

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的显示分离，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。 JSP(JavaServer Pages)是一种动态页面技术，它的主要目的是将表示逻辑从Servlet中分离出来。

Java Servlet是JSP的技术基础，而且大型的Web应用程序的开发需要Java Servlet和JSP配合才能完成。JSP具备了Java技术的简单易用，完全的面向对象，具有平台无关性且安全可靠，主要面向因特网的所有特点。

### 5.2.6Mysql

MySQL是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策（本词条“授权政策”），它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。由于其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境。

### 5.2.7JQuery

#### 5.2.7.1 JQuery简介

Jquery是继prototype之后又一个优秀的Javascript库。它是轻量级的js库 ，它兼容CSS3，还兼容各种浏览器（IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+），jQuery2.0及后续版本将不再支持IE6/7/8浏览器。jQuery使用户能更方便地处理HTML（标准通用标记语言下的一个应用）、events、实现动画效果，并且方便地为网站提供AJAX交互。jQuery还有一个比较大的优势是，它的文档说明很全，而且各种应用也说得很详细，同时还有许多成熟的插件可供选择。jQuery能够使用户的html页面保持代码和html内容分离，也就是说，不用再在html里面插入一堆js来调用命令了，只需要定义id即可。

jQuery是一个兼容多浏览器的javascript库，核心理念是write less,do more(写得更少,做得更多)。jQuery在2006年1月由美国人John Resig在纽约的barcamp发布，吸引了来自世界各地的众多JavaScript高手加入，由Dave Methvin率领团队进行开发。如今，jQuery已经成为最流行的javascript库，在世界前10000个访问最多的网站中，有超过55%在使用jQuery。

jQuery是免费、开源的，使用MIT许可协议。jQuery的语法设计可以使开发者更加便捷，例如操作文档对象、选择DOM元素、制作动画效果、事件处理、使用Ajax以及其他功能。除此以外，jQuery提供API让开发者编写插件。其模块化的使用方式使开发者可以很轻松的开发出功能强大的静态或动态网页。

jQuery，顾名思议，也就是JavaScript和查询（Query），即是辅助JavaScript开发的库。

#### 5.2.7.2 特点

* 动态特效
* AJAX
* 通过插件来扩展
* 方便的工具 - 例如浏览器版本判断
* 渐进增强
* 链式调用
* 多浏览器支持，支持Internet Explorer6.0+、Opera9.0+、Firefox2+、Safari2.0+、Chrome1.0+（在2.0.0中取消了对Internet Explorer6,7,8的支持）

### 5.2.8：Bootstrap

#### 5.2.8.1Bootstrap简介

Bootstrap，来自 Twitter，是目前最受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态[CSS](http://baike.baidu.com/view/15916.htm)语言Less写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，包括NASA的MSNBC（微软全国广播公司）的Breaking News都使用了该项目。

### 5.2.9：nutz

#### 5.2.9.1 nutz简介

Nutz是对于Java程序员来说，除SSH之外的另一个选择。当然，它是开源的，并且是完全免费的。同时也是商业友好的(Licensed under the Apache License, Version 2.0)。

#### 5.2.9.2特点

* **轻**

当前最新版，整个jar文件共910kB -- 针对JDBC的薄封装，无缓存。

* **全**

提供了Dao(ORM, SQL管理)，Ioc, Aop, Mvc, Json解析等必要功能。

* **活**

各个部分可以独立使用，比如在Spring里采用Nutz.Dao，又比如在Nutz.Ioc里使用Hibernate等。

* **整**

它所有功能均不依赖第三方jar文件。这就意味着：如果一个Web应用，你在WEB-INF/lib下只需要放置一个nutz.jar就够了。当然，你要使用其它的连接池、数据库驱动、打印PDF支持等功能，还需要自行添置jar包。

### 5.2.10：SeaJS

#### 5.2.10.1SeaJS简介

SeaJS是一个遵循CommonJS规范的JavaScript模块加载框架，可以实现JavaScript的模块化开发及加载机制。与jQuery等JavaScript框架不同，SeaJS不会扩展封装语言特性，而只是实现JavaScript的模块化及按模块加载。SeaJS的主要目的是令JavaScript开发模块化并可以轻松愉悦进行加载，将前端工程师从繁重的JavaScript文件及对象依赖处理中解放出来，可以专注于代码本身的逻辑。SeaJS可以与jQuery这类框架完美集成。使用SeaJS可以提高JavaScript代码的可读性和清晰度，解决目前JavaScript编程中普遍存在的依赖关系混乱和代码纠缠等问题，方便代码的编写和维护。

SeaJS的作者是前淘宝UED,现支付宝前端工程师玉伯。

SeaJS本身遵循KISS（Keep It Simple, Stupid）理念进行开发，其本身仅有个位数的API，因此学习起来毫无压力。在学习SeaJS的过程中，处处能感受到KISS原则的精髓——仅做一件事，做好一件事.

#### 5.2.10.1SeaJS特点

Sea.js 追求简单、自然的代码书写和组织方式，具有以下核心特性：

简单友好的模块定义规范：Sea.js 遵循 CMD 规范，可以像 Node.js 一般书写模块代码。

自然直观的代码组织方式：依赖的自动加载、配置的简洁清晰，可以让我们更多地享受编码的乐趣。

Sea.js 还提供常用插件，非常有助于开发调试和性能优化，并具有丰富的可扩展接口。

### 5.3小结

好的框架和技术如果很好利用的话，总能达到很好的效果，这里面的技术多多少少都用到过！

# 6系统实现

整个系统分为后台管理系统和前台系统。后台管理系统负责对系统的相关模块、数据进行统一的维护，由网站的管理员角色来完成。前台主要向相关用户展示功能页。

## 6.1后台管理系统

后台管理系统负责对系统的相关模块、数据进行统一的维护，由网站的管理员角色来完成。主要包括用户管理，学校管理，地址管理，群组管理，帖子管理，帖子类型管理。

### 6.1.1用户管理

#### 6.1.1.1用户列表页

* 设计思路

用户在后台管理页面点击“用户管理”链接，进入用户列表页面，页面展示系统总注册用户数，分页展示用户列表信息。

* 效果图



* 关键代码

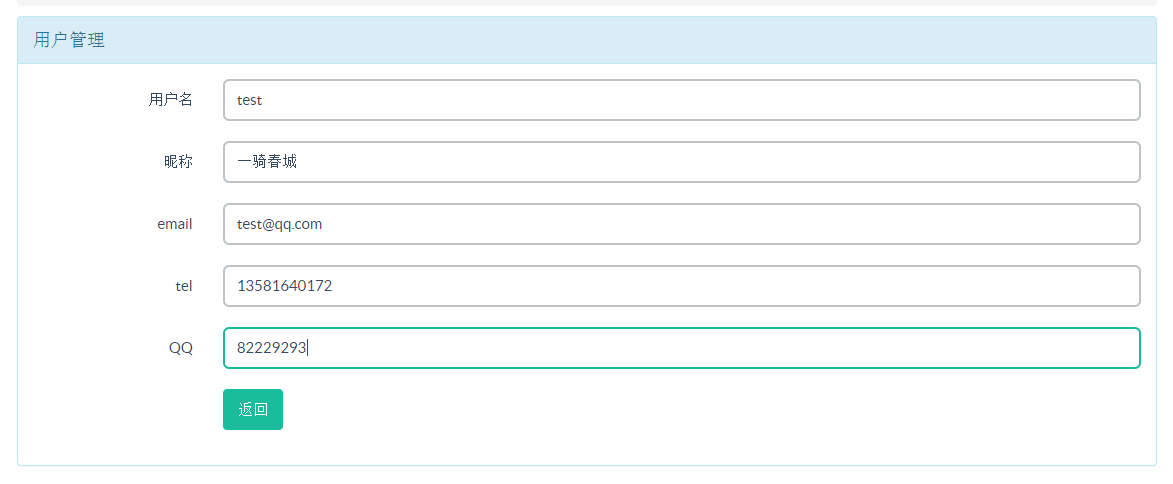
|  |
| --- |
| @At("/admin/user")  @Ok("jsp:/page/admin/user/user.jsp")  **public** **void** index(@Param("ps") Integer pageSize,  @Param("pn") Integer pageNumber, HttpServletRequest request) {  Pager pager = **new** Pager();  **if** (pageNumber == **null**) {  pageNumber = 1;  }  **if** (pageSize == **null**) {  pageSize = 10;  }  pager.setPageNumber(pageNumber);  pager.setPageSize(pageSize);  List<User> list = userService  .searchByPage(User.**class**, Cnd.*NEW*(), pager);  **int** count = userService.searchCount(User.**class**, Cnd.*NEW*());  pager.setRecordCount(count);  request.setAttribute("pager", pager);  request.setAttribute("list", list);  } |

#### 6.1.1.2用户详情页

* 设计思路

在用户列表页面，点击用户名称链接，进入用户详情页面，用户详情页根据用户id查询用户详细信息并展示在页面上

* 效果图



* 关键代码

|  |
| --- |
| 后台代码：  @At("/admin/userform")  @Ok("jsp:/page/admin/user/userForm.jsp")  **public** User form(@Param("id") Long id) {  User user = userService.find(id, User.**class**);  **return** user;  } |
| 前台代码：  <form class=*"form-horizontal"* role=*"form"* >  <input type=*"hidden"* name=*"user.id"* value=*"*${obj.id}*"*/>  <div class=*"form-group"*>  <label class=*"col-lg-2 control-label"*>用户名</label>  <div class=*"col-lg-10"*>  <input type=*"text"* name=*"user.uname"* class=*"form-control"* value=*"*${obj.uname}*"*>  </div>  </div>  <div class=*"form-group"*>  <label class=*"col-lg-2 control-label"*>昵称</label>  <div class=*"col-lg-10"*>  <input type=*"text"* name=*"user.nickName"* class=*"form-control"* value=*"*${obj.nickName}*"*>  </div>  </div>  <div class=*"form-group"*>  <label class=*"col-lg-2 control-label"*>email</label>  <div class=*"col-lg-10"*>  <input type=*"text"* name=*"user.email"* class=*"form-control"* value=*"*${obj.email}*"*>  </div>  </div>  <div class=*"form-group"*>  <label class=*"col-lg-2 control-label"*>tel</label>  <div class=*"col-lg-10"*>  <input type=*"text"* name=*"user.email"* class=*"form-control"* value=*"*${obj.tel}*"*>  </div>  </div>  <div class=*"form-group"*>  <label class=*"col-lg-2 control-label"*>QQ</label>  <div class=*"col-lg-10"*>  <input type=*"text"* name=*"user.email"* class=*"form-control"* value=*"*${obj.qq}*"*>  </div>  </div>  <div class=*"form-group"*>  <div class=*"col-lg-offset-2 col-lg-10"*>  <a href=*"*${ctx }*/admin/user"* class=*"btn btn-primary"*>返回</a>  </div>  </div>  </form> |

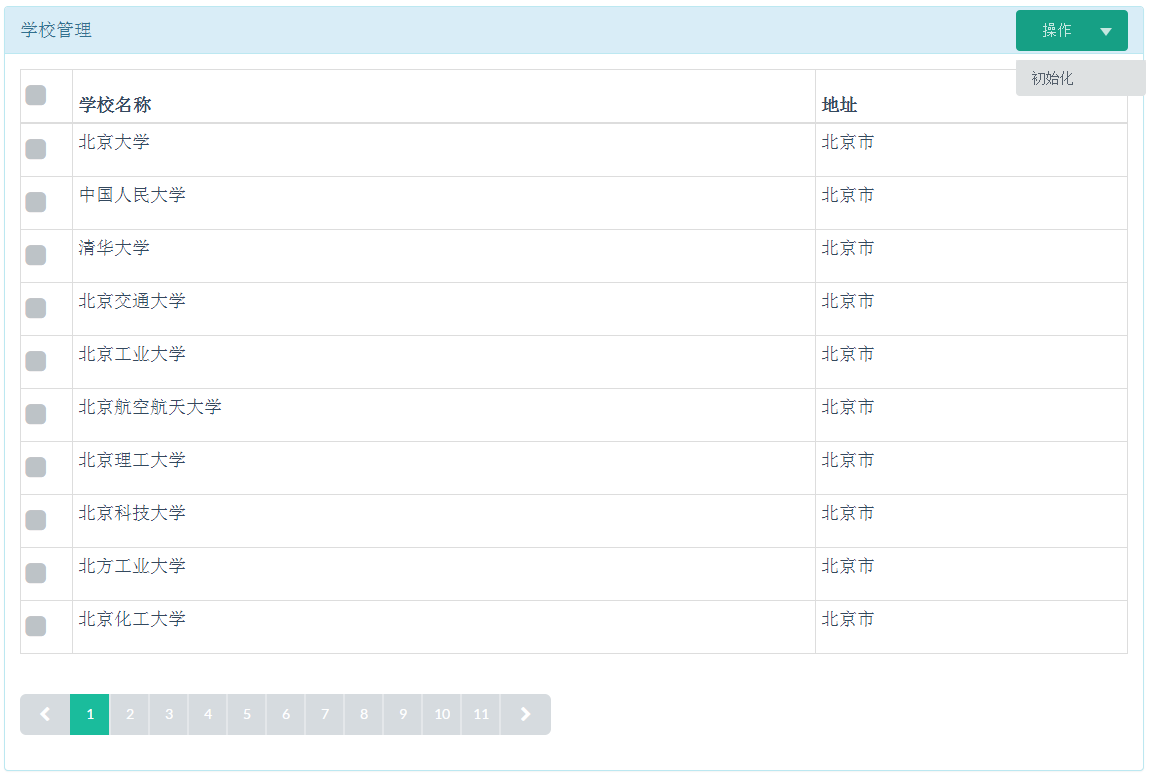
### 6.1.2学校管理

#### 6.1.2.1学校列表页

* 设计思路

在学校列表页，分页查询学校信息，并展示在页面，并提供了初始化学校功能。初始化的逻辑为：先清空学校表中的所有学校数据，然后从新浪教育频道讲全国几千所高效抓取下来并解析后存放到学校表中。

* 效果图



* 关键代码

采集学校使用到了开源的Jsoup组件，该组件可以方便的根据url从网上抓去数据，并提供了强大、易用的api对抓取到的页面文档进行解析、处理。

|  |
| --- |
| 初始化学校：  **public** **void** collect() {  // 清空学校表中的原有学校信息  **this**.clear();  String url = "http://edu.sina.com.cn/gaokao/2014-08-08/1903430534.shtml";  **try** {  Document doc = Jsoup.*connect*(url).timeout(100000).get();  Elements es = doc.getElementsByTag("tbody").get(0)  .getElementsByTag("a");  **for** (Element e : es) {  Document doc2 = Jsoup.*connect*(e.attr("href")).timeout(100000)  .get();  Elements es2 = doc2.getElementsByTag("tbody").get(0).children();  **for** (**int** i = 2; i < es2.size(); i++) {  Element e2 = es2.get(i);  Elements es3 = e2.children();  College c = **new** College();  c.setCname(es3.get(1).text().trim());  Region r = findByCondition(Region.**class**, "name", es3.get(3)  .text().trim());  c.setAddrCode(r != **null** ? r.getId() : **null**);  c.setAddr(es3.get(3).text().trim());  save(c);  }  }  } **catch** (Exception e2) {  e2.printStackTrace();  }  } |

### 6.1.3地址管理

#### 6.1.3.1省份列表页

* 设计思路

分页展示省份信息，包括省份的编码、名称，经度、纬度。

* 效果图



* 关键代码

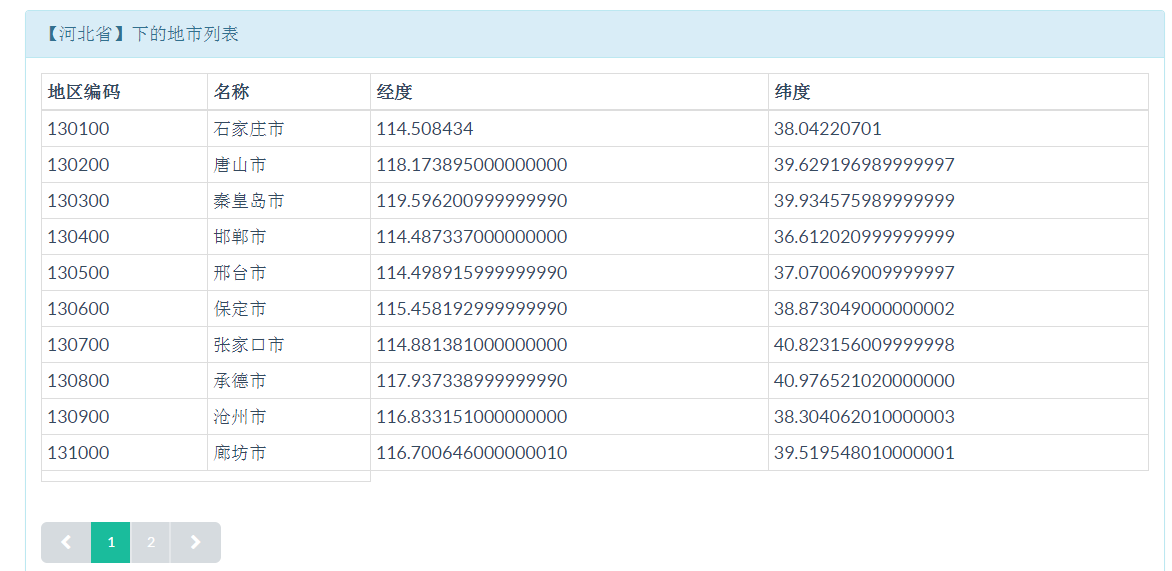
|  |
| --- |
| @At("/admin/region")  @Ok("jsp:/page/admin/region/province.jsp")  **public** **void** index(@Param("ps") Integer pageSize,  @Param("pn") Integer pageNumber, HttpServletRequest request) {  Pager pager = **new** Pager();  **if** (pageNumber == **null**) {  pageNumber = 1;  }  **if** (pageSize == **null**) {  pageSize = 10;  }  pager.setPageNumber(pageNumber);  pager.setPageSize(pageSize);  List<Region> list = regionService.searchByPage(Region.**class**,  Cnd.*where*("id", "like", "%0000"), pager);  **int** count = regionService.searchCount(Region.**class**,  Cnd.*where*("id", "like", "%0000"));  pager.setRecordCount(count);  request.setAttribute("pager", pager);  request.setAttribute("list", list);  } |

#### 6.1.3.1城市列表页

* 设计思路

在省份页面点击省份，进入城市列表，该页面根据根据省份编码查询该省份下所有城市列表并分页展示，如果该省份为直辖市，则列表中显示“该省份下没有数据！”

* 效果图





* 关键代码

|  |
| --- |
| @At("/admin/region/cities")  @Ok("jsp:/page/admin/region/cities.jsp")  **public** **void** index(@Param("ps") Integer pageSize,  @Param("pn") Integer pageNumber, @Param("pid") String provinceId,  HttpServletRequest request) {  Pager pager = **new** Pager();  **if** (pageNumber == **null**) {  pageNumber = 1;  }  **if** (pageSize == **null**) {  pageSize = 10;  }  pager.setPageNumber(pageNumber);  pager.setPageSize(pageSize);  List<Region> list = regionService.searchByPage(  Region.**class**,  Cnd.*where*("id", "like", provinceId.substring(0, 2) + "%").and(  "id", "<>", provinceId), pager);  **int** count = regionService.searchCount(  Region.**class**,  Cnd.*where*("id", "like", provinceId.substring(0, 2) + "%").and(  "id", "<>", provinceId));  pager.setRecordCount(count);  request.setAttribute("province",  regionService.find(Long.*valueOf*(provinceId), Region.**class**));  request.setAttribute("pid", provinceId);  request.setAttribute("pager", pager);  request.setAttribute("list", list);  } |

### 6.1.4群组管理

#### 6.1.4.1群组列表页

* 设计思路

分页展示群组列表，并提供群组的操作功能，包括针对新建群组的审核通过，解散群组，关闭群组功能。解散或者关闭群组的时候需要通知群组的创建者。并且如果群组的状态为非活动状态（审核中，已解散，已关闭），则前台不能展示该群组。

* 效果图



* 关键代码

|  |
| --- |
| /\*\*  \* 更新群组状态  \*  \* **@param** id  \* id  \* **@param** status  \* 要更改为的状态  \* **@param** statusMsg  \* 更改群组原因  \*/  **public** **boolean** updateGroupStatus(**long** id, **int** status, String statusMsg) {  Group entity = find(id, Group.**class**);  entity.setStatus(status);  **try** {  update(entity);  entity.getCreator();  MsgBox msg = **new** MsgBox();  msg.setSendUser("系统管理员");  msg.setDescript(statusMsg);  msg.setSendDate(**new** Date());  msg.setIfRead(MsgBox.***NO\_READ***);  msg.setReceiveUser(entity.getCreatorId());  save(msg);  **return** **true**;  } **catch** (Exception e) {  e.printStackTrace();  }  **return** **false**;  } |

#### 6.1.4.2群组详情页

* 设计思路

群组详情页和前台的群组相同，具体设计思路、效果图，关键代码请参考章节[6.2.3](#_6.2.3群组详情页)

### 6.1.5帖子类型管理

#### 6.1.5.1帖子类型列表页

* 设计思路

分页展示帖子类型信息，显示帖子的名称和描述

* 效果图



* 关键代码

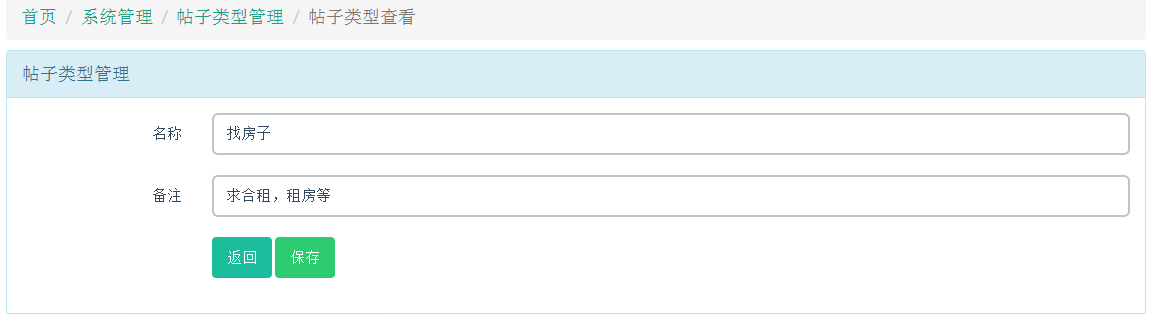
|  |
| --- |
| @At("/admin/msgtype")  @Ok("jsp:/page/admin/msgtype/msgtype.jsp")  **public** **void** index(@Param("ps") Integer pageSize,  @Param("pn") Integer pageNumber, HttpServletRequest request) {  Pager pager = **new** Pager();  **if** (pageNumber == **null**) {  pageNumber = 1;  }  **if** (pageSize == **null**) {  pageSize = 10;  }  pager.setPageNumber(pageNumber);  pager.setPageSize(pageSize);  List<MsgType> list = msgTypeService.searchByPage(MsgType.**class**,  Cnd.*NEW*(), pager);  **int** count = msgTypeService.searchCount(MsgType.**class**, Cnd.*NEW*());  pager.setRecordCount(count);  request.setAttribute("pager", pager);  request.setAttribute("list", list);  } |

#### 6.1.5.2帖子类型详情页

* 设计思路

在帖子列表页，点击帖子名称超链接，进入帖子详情页，该页面展示帖子的名称和描述信息，并可以进行编辑，并保存以更新帖子类型信息。

* 效果图

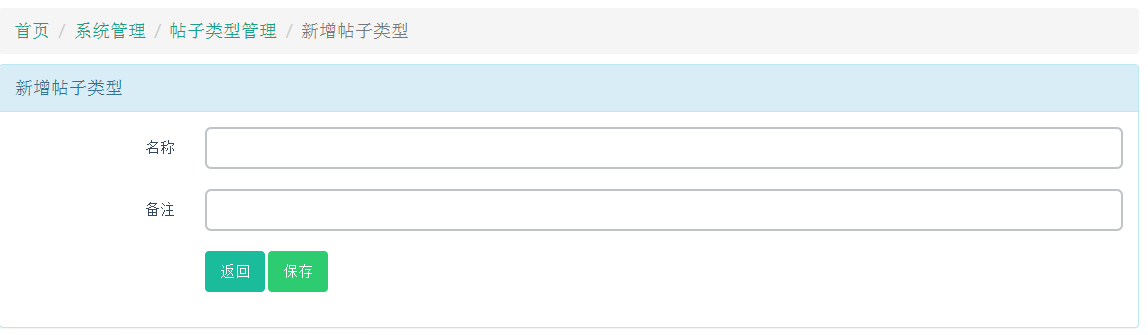


#### 6.1.5.3帖子类型添加页

* 设计思路

在帖子类型列表页，打开帖子类型添加页面，输入帖子名称和描述，点击保存按钮，保存一条帖子类型信息。

* 效果图



### 6.1.6帖子管理

#### 6.1.6.1帖子列表页

* 设计思路

分页显示帖子列表信息，并提供封帖功能。针对违反网站规定的帖子，对其进行封帖，封帖的时候需要输入封帖原因，并且发送一条消息给发帖者，告知其封帖原因，封的帖子。

* 效果图



* 关键代码

|  |
| --- |
| @At("/admin/message")  @Ok("jsp:/page/admin/message/message.jsp")  **public** **void** index(@Param("ps") Integer pageSize,  @Param("pn") Integer pageNumber, HttpServletRequest request) {  Pager pager = **new** Pager();  **if** (pageNumber == **null**) {  pageNumber = 1;  }  **if** (pageSize == **null**) {  pageSize = 10;  }  pager.setPageNumber(pageNumber);  pager.setPageSize(pageSize);  List<Message> list = messageService.searchByPage(Message.**class**,  Cnd.*where*("replayId", "=", 0), pager);  **int** count = messageService.searchCount(Message.**class**,  Cnd.*where*("replayId", "=", 0));  pager.setRecordCount(count);  request.setAttribute("pager", pager);  request.setAttribute("list", list);  } |

#### 6.1.6.1帖子详情页

* 设计思路

帖子详情页和前台的帖子详情页相同，具体设计思路、效果图，关键代码请参考章节[6.2.4](#_6.2.4帖子详情页)

## 6.2前台

网站前台即针对普通用户提供的页面功能，包括首页，群组列表页，群组详情页，帖子详情页，注册页，登录页，个人中心页。

### 6.2.1首页

* 设计思路

首页主要提供各个功能的链接入口，包括全部群组，登录，注册。

首页提供最热特定数量的最热群组和新建群组，最热群组根据该群组下的帖子数量来决定，新建群组则按照群组创建日期进行倒序排列。

首页展示也展示最新发布的20个帖子，以及最热的帖子，最新帖子按照发布时间来决定，最热帖子根据该帖子的回复量来决定。

* 效果图



* 关键代码

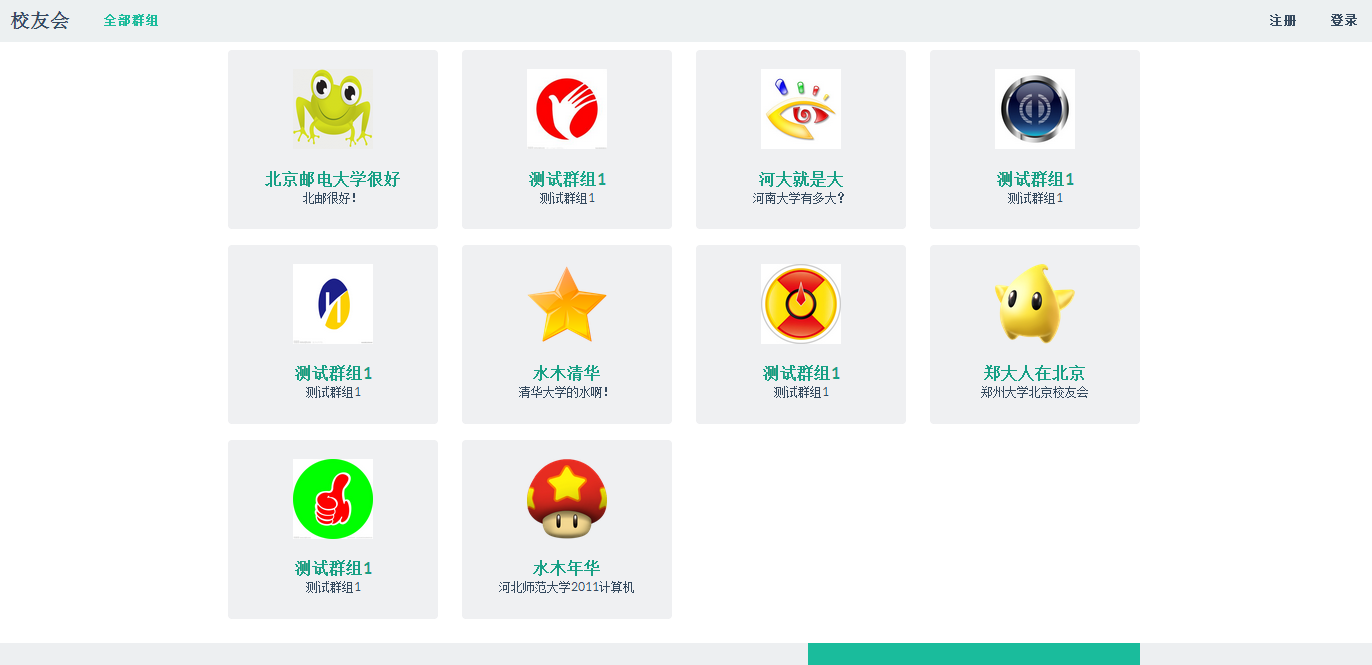
|  |
| --- |
| Controller：  @At("/")  @Ok("jsp:/page/front/index.jsp")  **public** **void** index(HttpServletRequest request) {  initService.init();  // 最热的四个群组  List<Group> hotGroupList = groupService.findHot();  // 最新的四个群组  List<Group> newGroupList = groupService.findNew();  // 最新的帖子  List<Message> newMessageList = messageService.findNew();  // 最热的帖子  List<Message> hotMessageList = messageService.findHot();  request.setAttribute("hotGroupList", hotGroupList);  request.setAttribute("newGroupList", newGroupList);  request.setAttribute("newMessageList", newMessageList);  request.setAttribute("hotMessageList", hotMessageList);  }  Service：  /\*\*  \* 查询活跃度最高的小组  \*  \* **@return**  \*/  **public** List<Group> findHot() {  **return** (List<Group>) search(  Group.**class**,  Cnd.*wrap*("WHERE status="  + Group.***STATUS\_HUODONG***  + " and id IN(select gid from fr\_message GROUP BY gid ORDER BY count(\*) DESC) order by id limit 0,4"));  }  /\*\*  \* 查询最近创建的的小组  \*  \* **@return**  \*/  **public** List<Group> findNew() {  **return** (List<Group>) search(  Group.**class**,  Cnd.*wrap*("where status=" + Group.***STATUS\_HUODONG***  + " order by createtime desc limit 0,4"));  }  /\*\*  \* 查询最新的20条帖子  \*  \* **@return**  \*/  **public** List<Message> findNew() {  List<Message> messageList = (List<Message>) search(  Message.**class**,  Cnd.*NEW*()  .*wrap*("title is not null order by createtime desc limit 0,20"));  **for** (Message msg : messageList) {  dao.fetchLinks(msg, "user");  }  **return** messageList;  }  /\*\*  \* 查询最热的10条帖子  \*  \* **@return**  \*/  **public** List<Message> findHot() {  Sql sql = Sqls  .*create*("select replay\_id from fr\_message group by replay\_id having replay\_id is not null order by count(\*) desc limit 0,10");  sql.setCallback(**new** SqlCallback() {  **public** Object invoke(Connection conn, ResultSet rs, Sql sql)  **throws** SQLException {  List<Long> list = **new** LinkedList<Long>();  **while** (rs.next())  list.add(rs.getLong("replay\_id"));  **return** list;  }  });  dao.execute(sql);  List<Long> idList = sql.getList(Long.**class**);  **if** (idList != **null** && idList.size() > 0) {  String sqlStr = "id in(" + idList.get(0);  **for** (**int** i = 1; i < idList.size(); i++) {  sqlStr += "," + idList.get(i);  }  sqlStr += ")";  List<Message> messageList = search(Message.**class**, Cnd.*wrap*(sqlStr));  **for** (Message msg : messageList) {  dao.fetchLinks(msg, "user");  }  **return** messageList;  }  **return** **null**;  } |

### 6.2.2全部群组

* 设计思路

该页面按照时间顺序列出所有创建的群组，（后期考虑参考帖子的那种分页形式。用户通过这里可以快速找到自己感兴趣的群组，并点进去查看。

* 效果图



* 关键代码

|  |
| --- |
| @At("/group/list")  @Ok("jsp:/page/front/group/list.jsp")  **public** List<Group> list(HttpServletRequest request) {  List<Group> groupList = groupService.search(Group.**class**, Cnd.*NEW*()  .*where*("status", "=", Group.***STATUS\_HUODONG***).desc("createtime"));  **return** groupList;  } |

### 6.2.3群组详情页

#### 6.2.3.1展示群组详情页信息

* 设计思路

在其他任何地方点击群组链接，根据群组id后台查询群组详情相关信息，在该页面展示，页面主要信息包括：群组详细信息，加入该群组的成员列表，分页展示该群组下的帖子列表。

* 效果图



* 关键代码

|  |
| --- |
| @At("/group/detail")  @Ok("jsp:/page/front/group/detail.jsp")  **public** **void** showGroupDetail(@Param("gid") Long gid,  HttpServletRequest request) {  String msg = request.getParameter("msg");  **if** ("joinfirst".equals(msg)) {  request.setAttribute("msg", "请先申请加入该群组");  }  Group group = groupService.find(gid);  Object obj = request.getSession().getAttribute("user");  **boolean** inGroup = **false**;  **if** (obj != **null**) {  User user = (User) obj;  inGroup = groupService.checkIfInGroup(user.getId(), gid);  }  request.setAttribute("inGroup", inGroup == **true** ? 1 : 0);  List<User> userList = groupService.findAllUserByGroupId(gid);  Pager pager = **new** Pager();  pager.setPageNumber(1);  pager.setPageSize(10);  List<Message> messageList = messageService.searchByPage(  Cnd.*where*("gid", "=", gid).and("replayId", "=", 0).desc("createtime"),  pager);  request.setAttribute("messageList", messageList);  request.setAttribute("group", group);  request.setAttribute("userList", userList);  } |

#### 6.2.3.2加入或者退出群组

* 设计思路

群组详情页面提供了加入和退出群组功能，进入该页面后，判断该用户是否在该群组，如果不在，则显示“加入”按钮，以便用户可以申请加入；如果用户已经加入该群组，则显示“退出”按钮，以便用户可以退出该群组。

* 效果图



* 关键代码

|  |
| --- |
| <a href='${ctx}/group/${inGroup=="1"?"quit":"join"}?gid=${group.id}' class="btn btn-info">${inGroup=="1"?"退出":"加入" } |

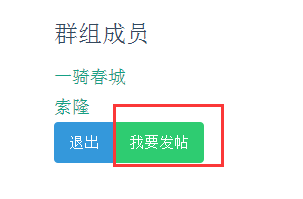
#### 6.2.3.3发布帖子

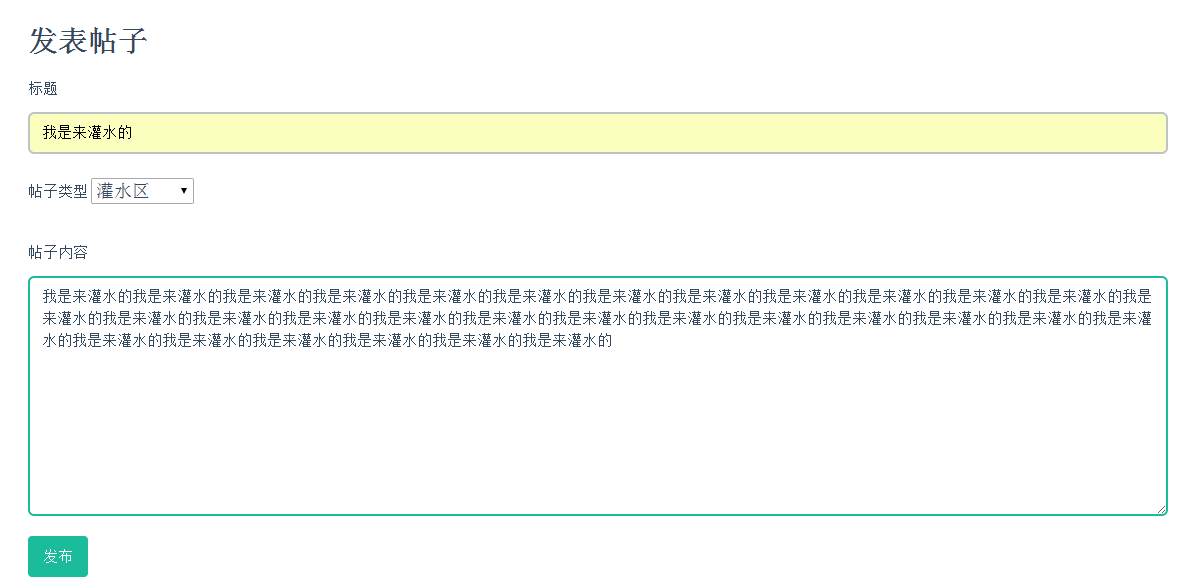
* 设计思路

用户在群组详情界面，点击“我要发帖”，如果该用户属于该群组，则跳转到发帖界面，如果不属于该群组，则提醒用户不属于该群组，需要先加入该群组再发帖。

在发帖界面，输入帖子标题，帖子类型，帖子内容，点击“发布”，则发布一条帖子。

* 效果图





* 关键代码

|  |
| --- |
| /\*\*  \* 跳转至发布帖子页面  \* **@param** gid  \* **@param** request  \* **@return**  \*/  @At("/message/add")  **public** View toAdd(@Param("gid") Long gid, HttpServletRequest request) {  Object user = request.getSession().getAttribute("user");  **if** (user == **null**) {  // 跳转至登录页面  **return** **new** JspView("/page/front/login.jsp");  }  // 判断当前登录用户是否是gid群组的，如果不是，跳转至该群组页面，提醒申请加入  **boolean** flag = groupService.checkIfInGroup(((User) user).getId(), gid);  **if** (!flag) {  **return** **new** ViewWrapper(**new** ServerRedirectView("/group/detail?gid="  + gid + "&msg=" + "joinfirst"), **null**);  }  List<MsgType> typeList = msgTypeService  .search(MsgType.**class**, Cnd.*NEW*());  request.setAttribute("typeList", typeList);  request.setAttribute("gid", gid);  **return** **new** JspView("/page/front/message/message\_form.jsp");  }  /\*\*  \* 发布一条帖子并跳转到新发布的帖子详情页  \* **@param** msg  \* **@param** request  \* **@return**  \* **@throws** Exception  \*/  @At("/message/save")  **public** View save(@Param("::msg.") Message msg, HttpServletRequest request)  **throws** Exception {  msg.setCreatetime(**new** Date());  User user = (User) request.getSession().getAttribute("user");  msg.setUid(user.getId());  messageService.save(msg);  **return** **new** ViewWrapper(**new** ServerRedirectView(  "/message/message\_detail?id=" + msg.getId()), **null**);  } |

### 6.2.4帖子详情页

#### 6.2.4.1帖子详情展示

* 设计思路

在帖子界面，展示帖子详细信息，包括帖子的回复信息。

* 效果图



* 关键代码

|  |
| --- |
| @At("/message/message\_detail")  @Ok("jsp:/page/Front/message/message\_detail.jsp")  **public** **void** showMessageDetail(HttpServletRequest request) {  String mid = request.getParameter("id");  Message message = messageService.find(Long.*parseLong*(mid));  request.setAttribute("message", message);  }  /\*\*  \* 根据帖子id查询帖子详细信息，并且查询该帖子的所有回复  \*  \* **@param** id  \* 帖子id  \* **@return**  \*/  **public** Message find(**long** id) {  Message msg = dao.fetch(Message.**class**, id);  dao.fetchLinks(msg, "user");  dao.fetchLinks(msg, "group");  dao.fetchLinks(msg, "msgType");  dao.fetchLinks(msg, "replayList");  List<Message> replayList = msg.getReplayList();  **for** (Message replay : replayList) {  dao.fetchLinks(replay, "user");  }  **return** msg;  } |

#### 6.2.4.2回复帖子

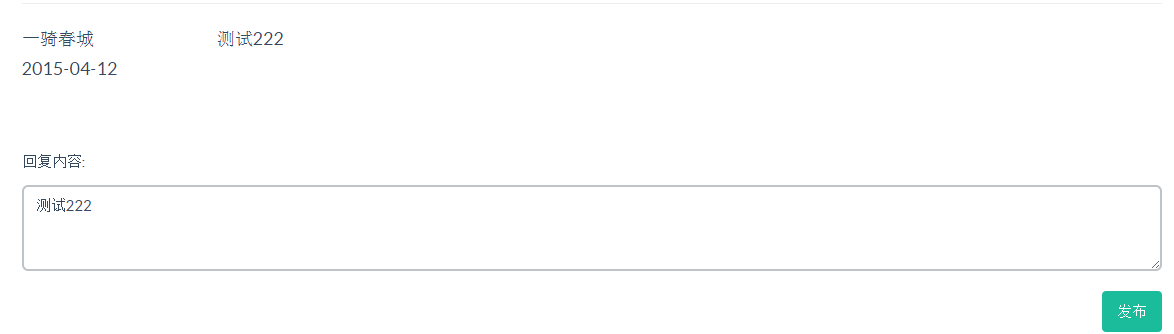
* 设计思路

在帖子详情界面，输入回复内容，则后台保存一条回复信息，如果该用户没有登录，则自动跳转到登录界面要求用户登录后再发表回复信息。成功回复一条信息后，会通过ajax自动在帖子回复列表的最后一行追加最新的回复

* 效果图







* 关键代码

|  |
| --- |
| $('#replayBtn').click(**function**() {  **var** content = $('#replayTxt').val().trim();  **if** (content == '') {  tooltip.tooltip('回复内容不能为空');  **return**;  }  $.post(ctx + '/message/addreplay', {  'msg.descript' : content,  'msg.gid' : gid,  'msg.replayId' : rid  }, **function**(result) {  **var** message = eval('(' + result + ')');  console.log(**typeof** message);  // 将json格式的字符串转换为json对象  **if** (message!=**null**&&(**typeof** message!='object')&&message.indexOf('error:')>-1) {  tooltip.tooltip(message.split(':')[1]);  **return**;  }    **if** (message == **null**) {  tooltip.tooltip('提交数据发生错误！');  **return**;  }  // 将返回的帖子添加到帖子顶部  **var** divmessage = '<hr><div class="row"><div class="col-md-2">';  divmessage += message.user.nickName + '<br>';  divmessage += message.createtime.substring(0, 10);  divmessage += '</div><div class="col-md-10">';  divmessage += message.descript;  divmessage += '</div></div>'  $('#replayDiv').before(divmessage);  });  }); |

### 6.2.5注册页

* 设计思路

点击顶部导航的“注册”链接，打开注册页面，在该页面输入必要的注册信息，点击“注册”则会成功注册一个用户

* 效果图



* 关键代码

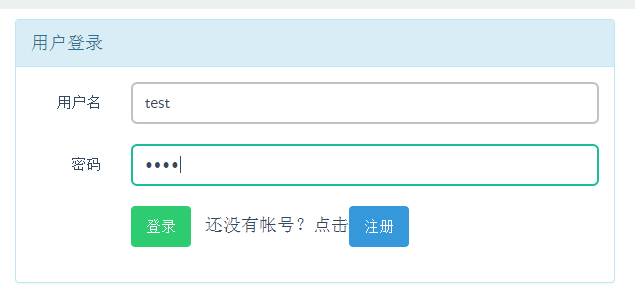
|  |
| --- |
| @At("/user/")  @Ok("jsp:/page/front/login.jsp")  **public** **void** reg(@Param("::user.") User user) **throws** Exception {  userService.save(user);  }  前台验证相关代码：  define(**function**(require, exports, module) {  **var** tooltip = require('tooltip');  tooltip.show(msg);  $('#province').change(  **function**(p1, p2, p3) {  **var** pid = **this**.value;  console.log(ctx);  $.post(ctx + '/front/querycity', {  pid : pid  }, **function**(result) {  $('#city').html(  '<option selected="selected">---请选择---</option>')  **var** resultObj = $.parseJSON(result);  **if** (resultObj.length == 0) {  } **else** {  **for** (**var** i = 0; i < resultObj.length; i++) {  $('#city').append(  '<option value="' + resultObj[i]['id']  + '">' + resultObj[i]['name']  + '</option>');  }  }  });  });  $('#confirmPwd').blur(**function**(){  **var** pwd = $('#pwd').val();  **var** pwd2 = $('#confirmPwd').val();  **if**(pwd!=pwd2){  $('#lblConfirmPwd').show().html('密码不一致');  $('#confirmPwd').focus();  **return**;  }  $('#lblConfirmPwd').show().html('');    });  $('#name').blur(**function**() {  **var** name = **this**.value;  **if** (name == **null** || name == "") {  $('#lblUser').show().html('用户名不能为空');  } **else** {  $.post(ctx + '/front/checkname', {  name : name  }, **function**(result) {  **if** (result == 'false') {  $('#lblUser').show().html('该用户名已经存在，请使用其他用户名');  } **else** {  $('#lblUser').html('').hide();  }  });  }  });  $('#regBtn').click(**function**(){  **var** success = **true**;  $("input").each(**function**(){  **var** valid = $(**this**).attr('valid');  **if**(valid){  **if**(valid=='notnull'){  **if**(!$(**this**).val()){  tooltip.show('请输入必填项');  success =**false**;  }  }  }  });  **return** success;    })  }) |

### 6.2.6登录页

* 设计思路

用户在需要登录的时候打开该页面，输入帐号和密码，点击登录，登录成功后自动跳转到网站首页，如果失败，则还跳转到该页面，并提示错误信息。

* 效果图



* 关键代码

|  |
| --- |
| @At("/login")  **public** View login(@Param("uname") String uname, @Param("pwd") String pwd,  HttpServletRequest request) {  User user = userService.login(uname, pwd);  **if** (user != **null**) {  request.getSession().setAttribute("user", user);  request.setAttribute("user", user);  **return** **new** ServerRedirectView("/");  } **else** {  request.setAttribute("msg", "用户名或密码错误");  **return** **new** JspView("/page/front/login.jsp");  }  } |

### 6.2.7个人中心

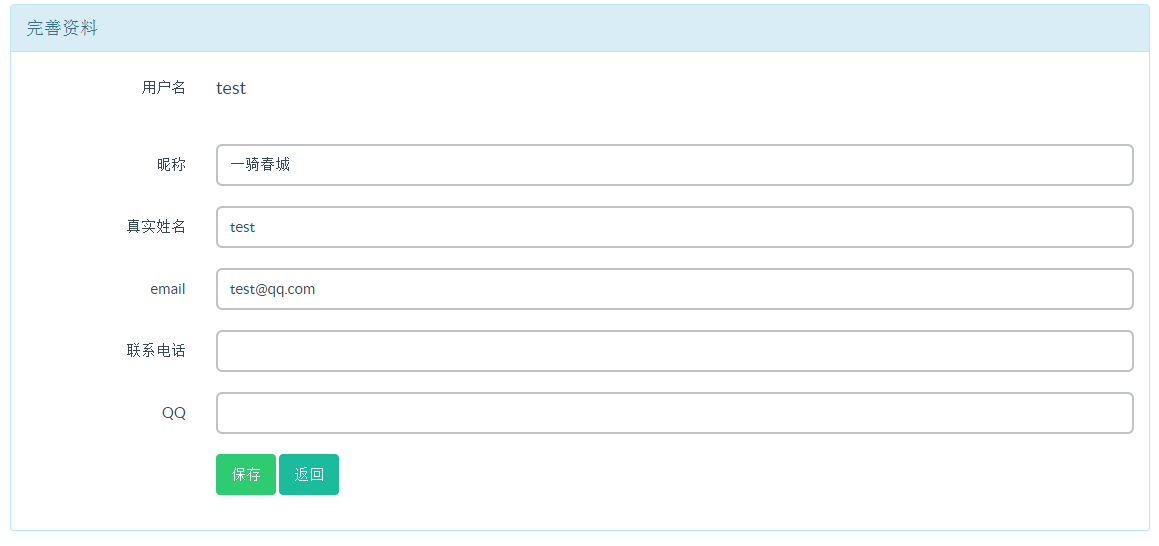
用户登录后可以进入自己的个人中心界面，其他用户也可以查看别人的个人中心界面，区别在于，个人可以修改自己的相关信息，其他人看别人的个人中心界面只能查看，不能更新相关内容。

#### 6.2.7.1基本资料

* 设计思路

显示个人的详细信息，可以编辑进行修改。用户名不能修改

* 效果图



* 关键代码

|  |
| --- |
| @At("user/update")  @Ok("redirect:/profile")  **public** **void** save(@Param("::user.") User param, HttpServletRequest request)  **throws** Exception {  User user = (User) request.getSession().getAttribute("user");  user.setNickName(param.getNickName());  user.setRealName(param.getRealName());  user.setEmail(param.getEmail());  user.setTel(param.getTel());  user.setQq(param.getQq());  userService.update(user);  } |

#### 6.2.7.2个人帖子管理

* 设计思路

列出自己发布的所有帖子，并且可以进行编辑和删除，删除的时候会把该帖子一级帖子下的所有回复都删除。

* 效果图



#### 6.2.7.3个人群组管理

* 设计思路

该功能包括“我创建的群组”和“我加入的群组”，每个模块都列出特定的群组列表，在“我创建的群组”里面，可以解散该群组，解散的时候，将该群组标记为解散状态，并且在前台也看不到这个群组。

* 效果图

我加入的群组：



我创建的群组：



#### 6.2.7.4修改密码

* 设计思路

在修改密码页面，需要输入原有密码，新密码和重复密码才能进行提交，保存新密码的时候需要判断原有密码是否正确，新密码和重复密码是否一致。否则不能成功修改密码。

* 效果图



关键代码

|  |
| --- |
| 前台js代码：  $('#oldpwdTxt').blur(**function**( ) {  **var** oldPwd = $(**this**).val();  $.post(ctx + '/user/checkoldpwd', {  oldPwd : oldPwd  }, **function**(result) {  console.log(result);  **if** (result == 'false') {  $('#oldpwdTxt').focus();  tooltip.tooltip('原有密码输入不正确');  }  });  });    $('#changePwdBtn').click(**function**(){  **var** pwd = $('#pwdTxt').val();  **var** rePwd = $('#rePwdTxt').val();  **if**(pwd!=rePwd){  tooltip.tooltip('前后密码不一致');  **return** **false**;  }  });  后台代码：  /\*\*  \* 验证旧密码是否正确  \*  \* **@param** oldPwd  \* 旧密码  \* **@param** request  \* **@return**  \*/  @At("/user/checkoldpwd")  @Ok("json")  **public** **boolean** checkOldPwd(@Param("oldPwd") String oldPwd,  HttpServletRequest request) {  User user = (User) request.getSession().getAttribute("user");  **if** (user.getPwd().equals(oldPwd)) {  **return** **true**;  }  **return** **false**;  }  /\*\*  \* 修改密码  \*  \* **@param** pwd  \* **@param** rePwd  \* **@param** request  \*/  @At("/user/changepwd")  @Ok("redirect:/profile")  **public** **void** ChangePwd(@Param("pwd") String pwd, HttpServletRequest request) {  User user = (User) request.getSession().getAttribute("user");  user.setPwd(pwd);  **try** {  userService.update(user);  } **catch** (Exception e) {  }  } |

#### 6.2.7.5收件箱

* 设计思路

该模块主要展示当前用户收到所有的短消息

* 效果图



## 6.3公共技术组件

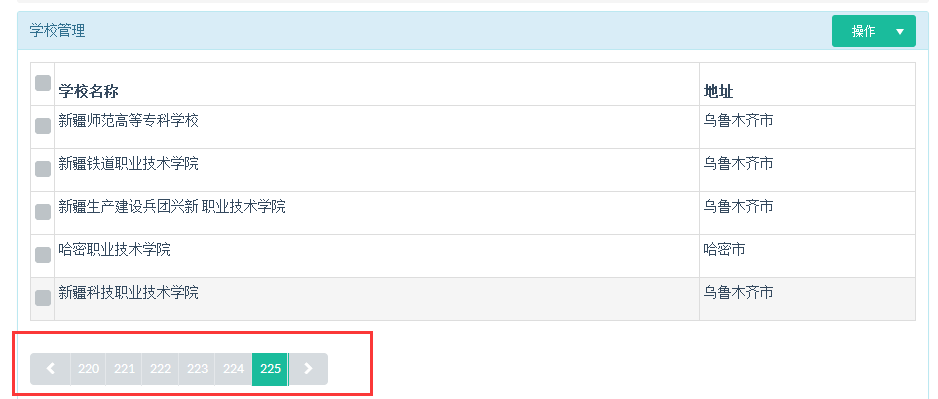
### 6.3.1分页组件

* 设计思路

由于在后台管理系统中几乎每个模块都涉及到分页功能，且后台使用了nutz框架封装的分页组件org.nutz.dao.pager.Pager。那么前台的分页效果可以进行封装供所有同样的分页场景使用。

分页组件关键用的几个属性：pageSize（每页显示记录数），pageNumber（当前页数），pageCount（总页数）在后台返回的org.nutz.dao.pager.Pager对象中都具备了，所以使用这几个关键值画出一组分页按钮即可。

* 效果图





* 关键代码

关键的分页代码封装在page.js中，。在分页的页面定义放置分页组件的dom元素：

<div class="pagination" id="page" url="${ctx}/admin/user"></div>

需要引入该文件，然后定义三个基本属性：

|  |
| --- |
| <script type=*"text/javascript"*>  **var** pageSize = '${pager.pageSize}';  **var** pageNumber = '${pager.pageNumber}';  **var** pageCount = '${pager.pageCount}';  </script> |

最后在js中初始化分页功能：

|  |
| --- |
| **var** Page = require('page');//引入page.js组件  **new** Page(pageSize, pageNumber, pageCount);//初始化page功能 |

### 6.3.2提示组件

* 设计思路

该功能主要针对后台使用较多，很多地方都用到了提示功能，所以进行了封装。主要封装了两个接口，一个tooltip，一个show，前者为提示性信息，会在提醒后三秒内自动隐藏，而show则会一直提醒。

* 效果图

比如点击删除功能的时候，如果没有选择要删除的记录，会提醒：



* 关键代码

tooltip代码封装在tooltip.js中，

|  |
| --- |
| define(**function**(require, exports, module) {  module.exports = {  container : $('#tooltip'),  show : **function**(text) {  $("html,body").animate({  scrollTop : 0  }, 200);  **this**.container.html(text);  **this**.container.css('width', (text.length \* 20) + 'px');  **this**.container.show();  },  tooltip:**function**(text){  **this**.show(text);  **var** me = **this**;  setTimeout(**function**(){  me.hide();    },3000);  },  hide : **function**() {  **this**.container.hide();  **this**.show(text);  **var** me = **this**;  setTimeout(**function**() {  me.hide();  }, 3000);  }  };  }); |

使用示例：



### 6.3.2基础的服务类

* 设计思路

该服务类封装了针对数据库的基本的增上该查功能，其他所有service都必须继承该类，然后根据自己的特殊需要再扩展对应的方法。

* 关键代码

关键代码，封装在GeneralService.java中，

## 6.4权限认证

本系统分为后台管理和前台网站， 所以登录也分为两方面内容，后台管理员登录，前台用户登录

### 6.4.1 管理员登录

管理员登录url为：<http://ip:port/appname/admin>。默认的管理员账户和密码为admin/admin，暂时阿弥有提供管理员的密码修改功能

管理员登录后可以操作后台管理系统中的所有功能，使用前台账户的账户和密码是无法成功登录后台管理系统的。

### 6.4.2 前台用户登录

任何网站用户如果想需要创建、加入群组、发帖，会贴，必须登录账户才能进行。前台用户登录参考章节[6.2.6](#_6.2.6登录页)

# 7系统安装配置

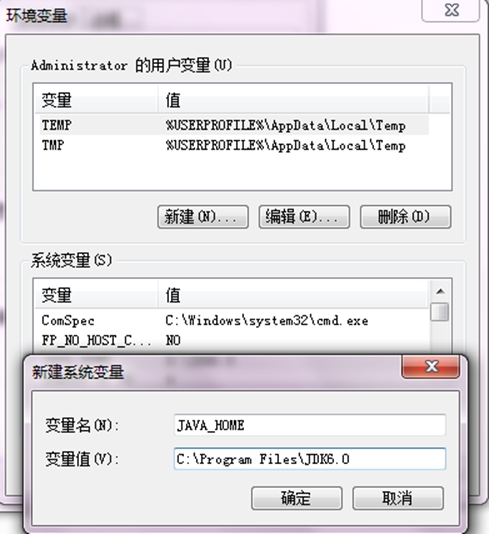
## 7.1 JDK安装

直接下载解压版jdk1.6

在监控服务器上打开Windows环境变量设置界面，

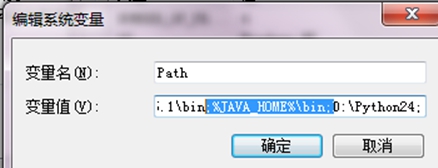
一、新建一个系统变量， 变量名：JAVA\_HOME

变量值：C:\Program Files\JDK6.0 如下图所示：



二、在系统变量里，找到变量 Path

在最后面加入： ;%JAVA\_HOME%\bin;



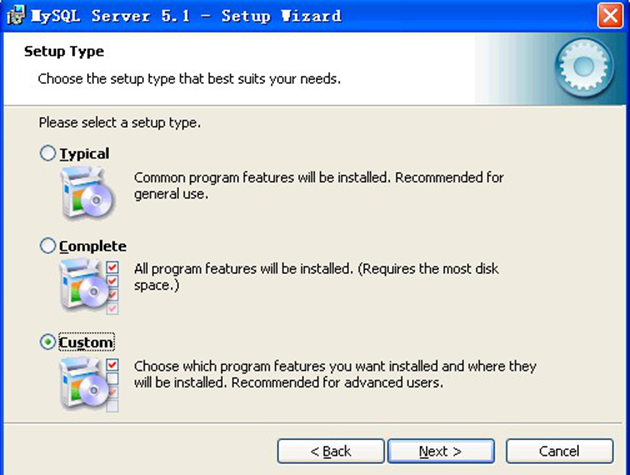
最后点击确定保存。

## 7.2 tomcat安装

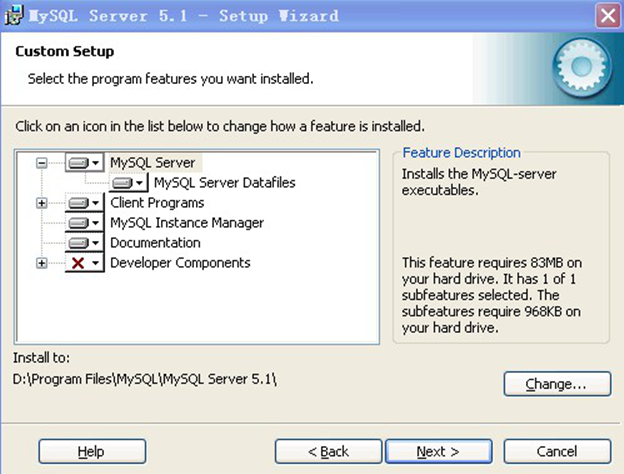
在官网下载tomcat解压即可。

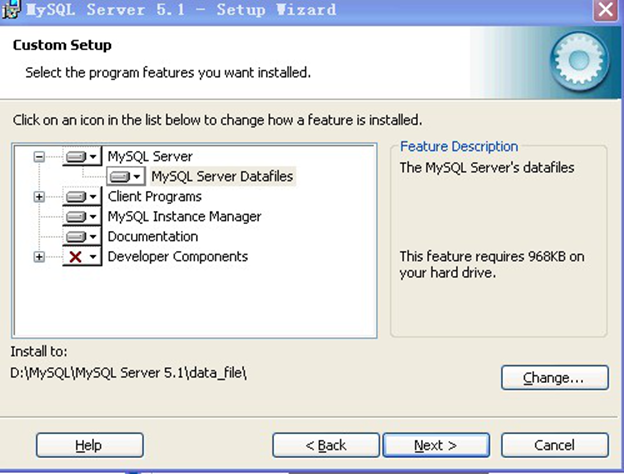
## 7.3 mysql安装

mysql安装向导启动，按“Next”继续, 选择安装类型，有“Typical（默认）”、“Complete（完全）”、“Custom（用户自定义）”三个选项，我们选择“Custom”，有更多的选项，也方便熟悉安装过程



2)更改安装路径和数据文件存放路径。

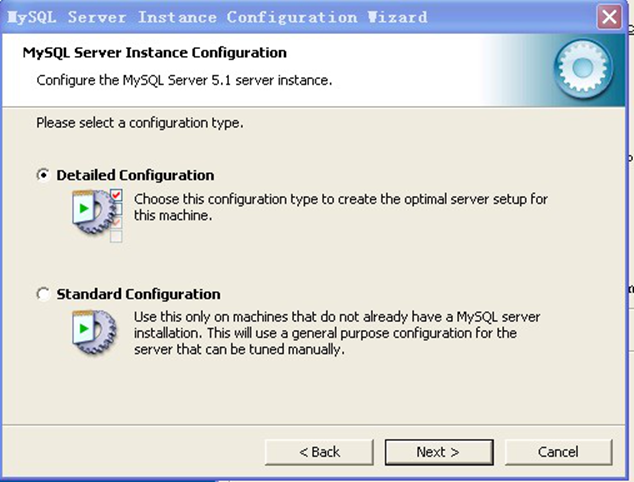




3）执行完安装后，开始出现配置数据库实例界面，如下图：



4）选择选择配置方式，“Detailed Configuration（手动精确配置）”、“Standard Configuration（标准配置）”，我们选择“Detailed Configuration”，方便熟悉配置过程。



5）选择服务器类型，“Developer Machine（开发测试类，mysql占用很少资源）”、“Server Machine（服务器类型，mysql占用较多资源）”、“Dedicated MySQL Server Machine（专门的数据库服务器，mysql占用所有可用资源）”，大家根据自己的类型选择了，一般选“Server Machine”，不会太少，也不会占满。如下图



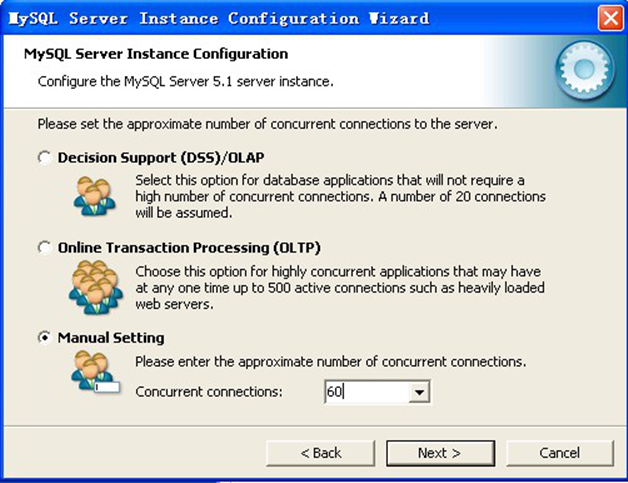
6）选择mysql数据库的大致用途，“Multifunctional Database（通用多功能型，好）”、“Transactional Database Only（服务器类型，专注于事务处理，一般）”、“Non-Transactional Database Only（非事务处理型，较简单，主要做一些监控、记数用，对MyISAM数据类型的支持仅限于non-transactional），随自己的用途而选择了，我这里选择“Transactional Database Only”，按“Next”继续。



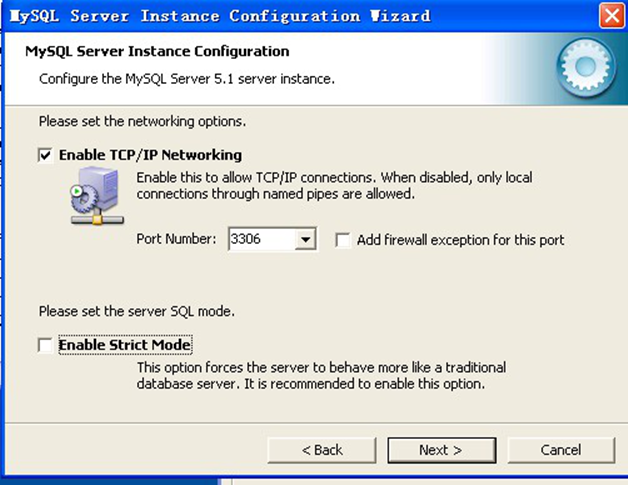
6）对InnoDB Tablespace进行配置，就是为InnoDB 数据库文件选择一个存储空间，如果修改了，要记住位置，重装的时候要选择一样的地方，否则可能会造成数据库损坏，当然，对数据库做个备份就没问题了，这里不详述。我这里没有修改，使用用默认位置，直接按“Next”继续



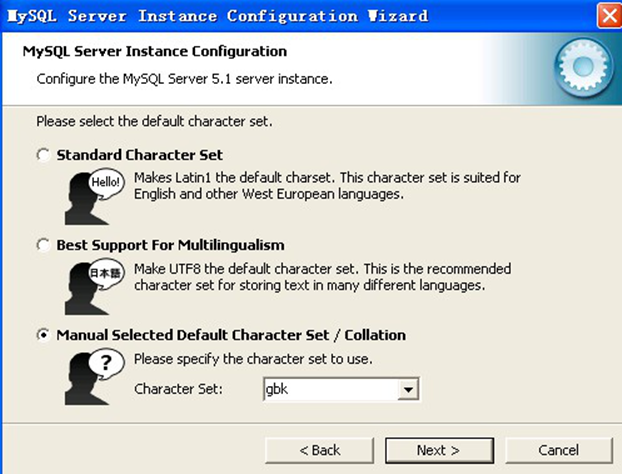
7）选择您的网站的一般mysql访问量，同时连接的数目，“Decision Support(DSS)/OLAP（20个左右）”、“Online Transaction Processing(OLTP)（500个左右）”、“Manual Setting（手动设置，自己输一个数 60）”，我这里选“Online Transaction Processing(OLTP)”，自己的服务器，应该够用了，按“Next”继续



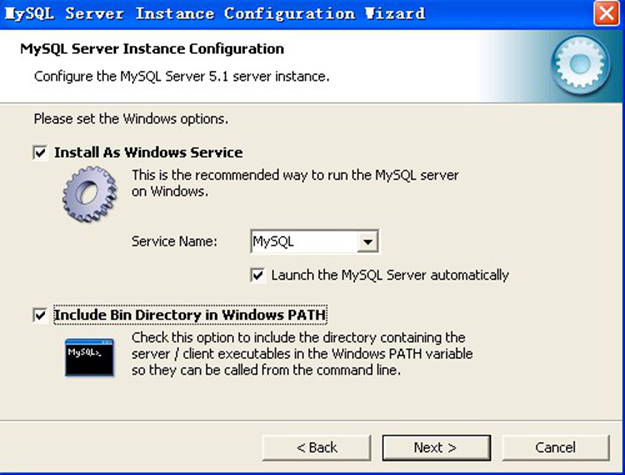
8）是否启用TCP/IP连接，设定端口，如果不启用，就只能在自己的机器上访问mysql数据库了，我这里启用，把前面的勾打上，Port Number：3306，在这个页面上，您还可以选择“启用标准模式”（Enable Strict Mode），这样MySQL就不会允许细小的语法错误。如果您还是个新手，我建议您取消标准模式以减少麻烦。但熟悉MySQL以后，尽量使用标准模式，因为它可以降低有害数据进入数据库的可能性。按“Next”继续



9）这个比较重要，就是对mysql默认数据库语言编码进行设置，第一个是西文编码，第二个是多字节的通用utf8编码，都不是我们通用的编码，这里选择第三个，然后在Character Set那里选择或填入“gbk”，当然也可以用“gb2312”，区别就是gbk的字库容量大，包括了gb2312的所有汉字，并且加上了繁体字、和其它乱七八糟的字——使用mysql的时候，在执行数据操作命令之前运行一次“SET NAMES GBK;”（运行一次就行了，GBK可以替换为其它值，视这里的设置而定），就可以正常的使用汉字（或其它文字）了，否则不能正常显示汉字。按 “Next”继续。



10）选择是否将mysql安装为windows服务，还可以指定Service Name（服务标识名称），是否将mysql的bin目录加入到Windows PATH（加入后，就可以直接使用bin下的文件，而不用指出目录名，比如连接，“mysql.exe -uusername -ppassword;”就可以了，不用指出mysql.exe的完整地址，很方便），我这里全部打上了勾，Service Name不变。按“Next”继续。



11）这一步询问是否要修改默认root用户（超级管理）的密码（默认为空），“New root password”如果要修改，就在此填入新密码（如果是重装，并且之前已经设置了密码，在这里更改密码可能会出错，请留空，并将“Modify Security Settings”前面的勾去掉，安装配置完成后另行修改密码），“Confirm（再输一遍）”内再填一次，防止输错。   
“Enable root access from remote machines（是否允许root用户在其它的机器上登陆，如果要安全，就不要勾上，如果要方便，就勾上它）”。 (一定要勾选上，允许root其它机器上登陆)

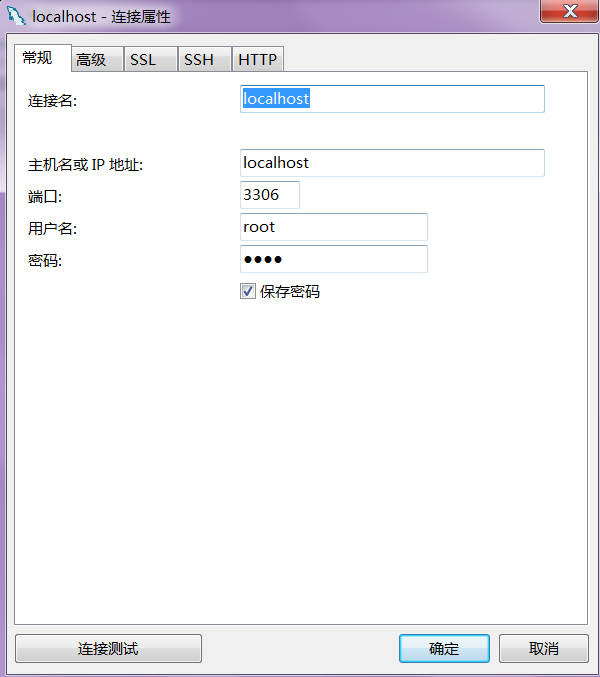
最后“Create An Anonymous Account（新建一个匿名用户，匿名用户可以连接数据库，不能操作数据，包括查询）”，一般就不用勾了，设置完毕，按“Next”继续。



## 7.4 mysql客户端软件安装

在本地安装mysql客户端， 打开“Navicat for MySQL.exe”, 默认安装即可。

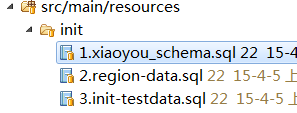
安装后，打开改程序，添加和远程数据库的连接，如图：



## 7.5 初始化数据

新建数据库，数据库名“xiaoyou”,字符集“UTF-8”

按照下面数字顺序执行三个sql文件，初始化校友数据库：



将打包好的该项目war包复制到tomcat/webapps目录下，启动tomcat即可。

# 结束语

# 参考文献

# 附录

# 致谢