**数据库需求分析：**

用户可以登录与注册，数据库需要保存用户名（昵称）与密码、性别、所在城市、VIP等级、签到积分、签名的基本信息；当用户忘记密码是可以通过邮件找回，所以需要记录用户的邮箱；用户登录可通过输入邮箱登录。用户登录后可发帖、查看自己发表的帖子，并可以重新编辑、删除等；用户可以回复别人发的帖；系统记录用户发帖、回帖时间（精确到秒）；用户可以收藏帖子，页面可以显示收藏时间，并可以删除收藏；每个帖子记录访问量；用户可以上传头像，所以需要记录用户上传的图片路径；系统记录每个用户每次登陆的时间；一个用户可以发表多个帖子，所以用户和帖子是一对多关系。

**Java Web需求分析：**

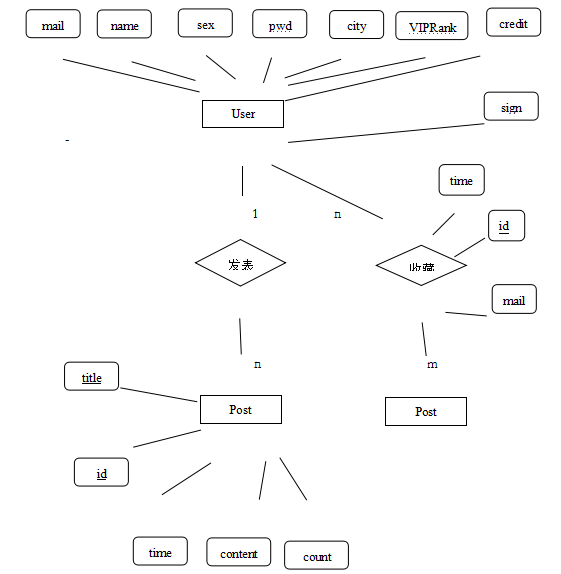
用户进入网站后，首先进入首页，在页面顶端有交流、案例等选项，点击则跳转至相应功能页面。在网站的右上角，有登录、注册、QQ、微博快速登录的图案及字样，用户点击后即进入相应的登录注册页面，当用户鼠标悬停在某些字样上时对应的字体变亮，达到较好的交互效果。其中QQ、微博登录要求已经安装了QQ、微博。在登录页面，用户输入邮箱、密码、验证码进行登录，并自动检测邮箱格式，如果用户输入的不是邮箱的格式则提示用户输入邮箱格式不正确，如果用户输入的邮箱数据库中不存在则提示该用户名不存在，如果密码输入错误则提示密码错误，验证码输入错误则提示验证码错误，其中验证码由服务器端随机产生。如果忘记密码可以点击忘记密码字样进入找回密码界面。在登录页面的登录字样旁边有注册字样，点击即进入注册界面，用户也可以通过点击右上方的注册字样进入注册界面，在注册页面用户输入邮箱、昵称、密码、确认密码、验证码进行注册，提示用户邮箱将会成为您唯一的登入名，密码格式为6到16个字符，如果不符合则提示用户“密码格式为6到16个字符”，如果验证码输入错误则提示验证码错误，自动检测邮箱格式，如果用户输入的不是邮箱的格式则提示用户输入邮箱格式不正确，如果用户输入的邮箱在数据库已存在则提示用户该邮箱已注册，为保证密码只让用户个人知道需要将密码进行加密再存入数据库，当用户注册成功后自动跳转至登录界面。当用户登录成功后有将跳转至首页，但此时右上角不再显示登录、注册按钮，取而代之的是显示该登录的用户的昵称、等级、头像。其中用户的等级根据用户发帖、回帖的数量、发帖的阅读量、点赞数、读别人帖子的数量、登录次数、时长、天数等自动增加，刚注册时用户等级为0。当用户鼠标悬停在头像或昵称上时将出现关于用户的菜单，用户可以选择基本设置、我的消息、我的主页、退出，点击分别进入相应页面。在基本设置页面也有进入我的消息、我的主页的链接，方便用户在这些页面间的跳转。在基本设置页面中用户可以更改自己的昵称、所在城市、签名、头像、密码等基本信息，还可以绑定邮箱。当用户修改密码时需要输入旧密码，修改后的密码同样要遵守密码格式，否则提示错误。在用户主页页面，显示用户的头像、昵称、性别（用图标显示）、VIP等级、认证信息、积分数、加入时间、城市等基本信息，并显示最近发的帖、回的帖。在我的消息界面显示收到的信息，提供单个删除和删除所有的操作，点击该用户名进入该用户的详情界面。在用户详情页面可以看最近发过的帖和收藏的帖，点击相应的帖子则进入该帖子页面。在网站主页，当用户登录后显示我发的帖和我收藏的帖链接，点击即进入相应页面。登录后还显示签到模块，点击“今日签到”按钮即进行签到，签到后可增加5积分，用户点击说明时则显示签到规则说明，点击活跃榜显示活跃的用户。在主页分类、分页显示用户的帖子，其中置顶、较火的帖子显示在前面，用户点击某个帖子即进入该帖子的详细页面。主页对帖子进行分类，分为提问、分享、讨论、建议、公告等，并提供这些类别的链接，当用户点击某一类即显示该类别的帖子。另外，帖子还有“未结”、“以结”、“热帖”等标签。在主页的右侧，提供其他网站优秀帖子的链接，方便读者阅读、减少查找所需信息的时间。对于每一个用户，系统将记录用户所有浏览帖子的类型以推测该用户喜欢的类型以便智能推荐。在帖子主页，显示该帖子的内容、作者昵称、悬赏的积分数等信息，当用户点击作者昵称时进入该作者的详情页面，在该页面显示作者基本信息、可向该作者发信息、加好友等。用户可收藏帖子、回复帖子。在网站主页，用户可以点击发表帖子按钮，如果用户未登录则跳转则登录界面，如果已登录则直接跳转至发帖界面，用户必须编写标题、帖子内容才可发表成功，否则提示错误，用户可以选择帖子类型、悬赏积分、插入表情、图片等，发表帖子也需要验证码。

**数据库设计：**

数据库的访问、数据的存取、持久化对象与关系间的映射用Hibernate框架实现。其每一张数据库表与面向对象的持久化对象PO对应。本应用访问的数据库名为forum；一个用户可以发表多个帖子，所以用户和帖子是一对多关系。

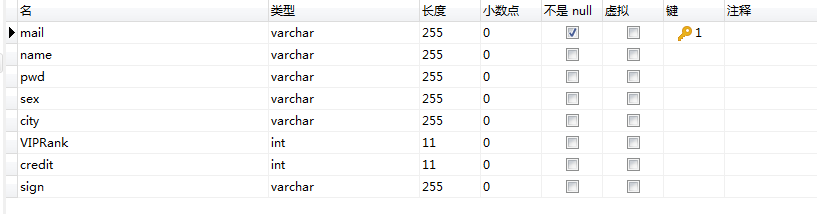
**ER模型：**

User与Post、Collection的关系：

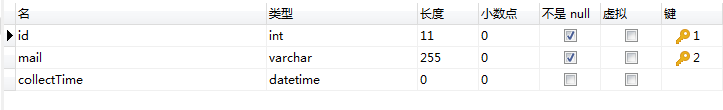


**数据库逻辑结构设计：**

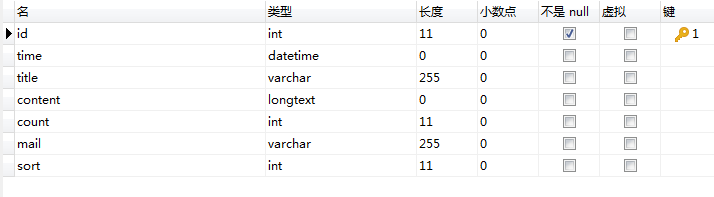
User：



Collection：



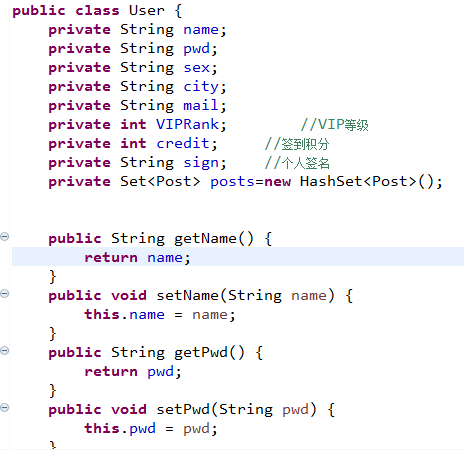
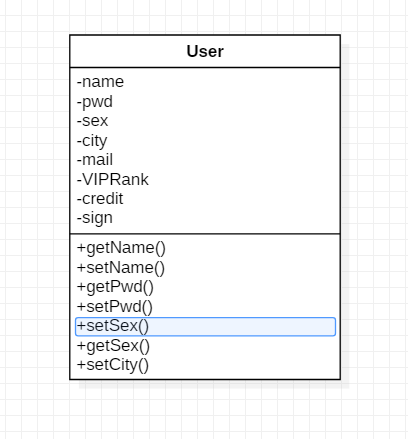
Post：



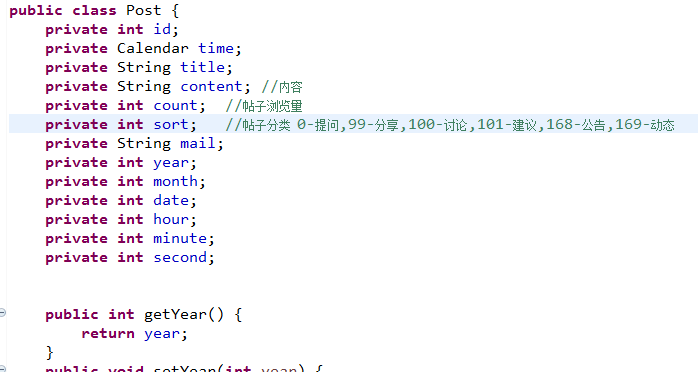
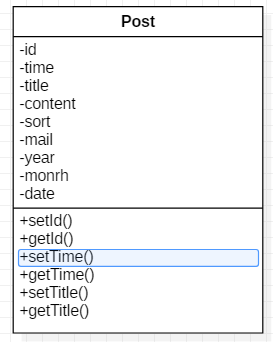
**Bean设计：**

PO设计（持久化对象）：

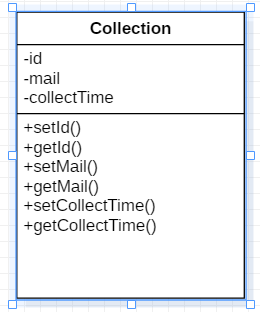
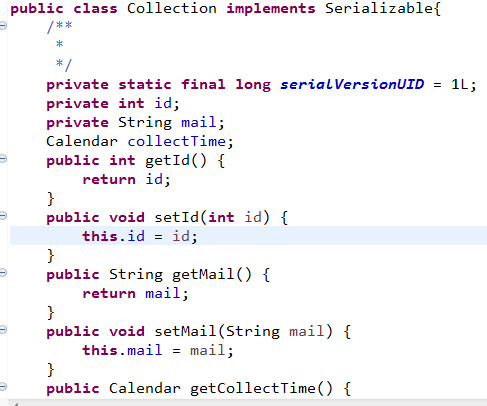
User（省略了一些set和get方法）：



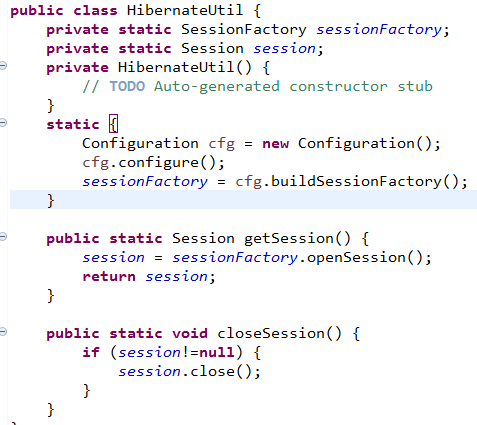
Post（省略了一些set和get方法）：



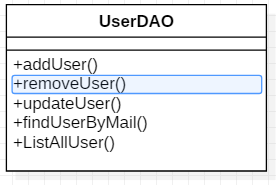
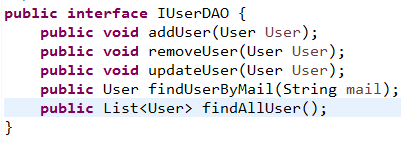
Collection：

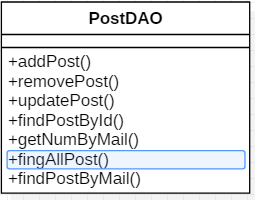
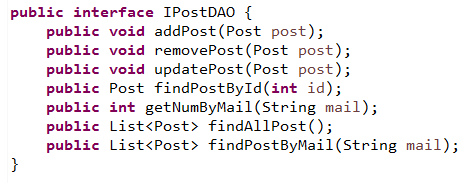
**DAO设计（bean对应数据库操作方法）：**



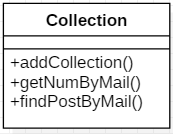
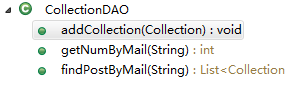
UserDAO：

PostDAO：

CollectionDAO：

**业务处理：**

