

《JavaEE软件综合实践》实习报告

指导教师：姜伟杰、张俊三、张培颖

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 姓名 | 学号 | 班级 | 工作量占比% | 成绩 |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 小组名称 |  | | | | |
| 项目名称 |  | | | | |
| 教师评语 |  | | | | |
| 教师签字 |  | | | | |

青岛软件学院、计算机与科学与技术学院

软件工程

目录

[一、项目概述 3](#_Toc45315133)

[二、需求分析 3](#_Toc45315134)

[2.1总体目标 3](#_Toc45315135)

[2.2具体目标 4](#_Toc45315136)

[2.3图示分析 5](#_Toc45315137)

[三、概要设计 5](#_Toc45315138)

[3.1接口设计 5](#_Toc45315139)

[3.2数据库设计 7](#_Toc45315140)

[3.3运行设计 12](#_Toc45315141)

[3.4出错处理设计 12](#_Toc45315142)

[四、模块分工 13](#_Toc45315143)

[五、实习收获(心得体会) 15](#_Toc45315144)

[六、致谢 17](#_Toc45315145)

# 项目概述

本项目的开发背景是我们刚刚在小学期的学习阶段学习了SSM框架，在后续的实训阶段中将以小组为单位实际运用所学，我们小组选择的是虚拟社区系统，后端采用SSM框架，前端采用Bootstrap框架，最终开发的虚拟社区系统功能完善，并已成功部署到云端可以直接在线访问，其核心功能页面是系统首页和管理员页面。

在系统首页中，用户可以浏览全部帖子，还能看到热门文章，也可以根据关键字或板块搜索查看想看的帖子，在注册并登录后，便可进行发帖、评论、收藏、关注等操作，注册中实现了邮箱验证码验证，发帖时可以进行图片或视频的上传，在首页中还可以进入个人主页，在个人主页中用户可以进行编辑个人资料，查看动态、回答、关注、收藏等信息。

在管理员页面中，管理员在登录后可以进行用户的删除、帖子的审核以及板块的增删改，还能看到每个板块的访问量。

# 需求分析

## 2.1总体目标

依托网络技术的迅速发展，网络虚拟社区系统的概念随之初现。网络虚拟社区是指一个网络板块，主要内容包含不同的人围绕同一主题引发的讨论，如天涯社区等。具有活力的论坛拥有庞大核心用户群体，主体覆盖人们生活中的方方面面。

本项目的总体目标是做出一基于WEB的虚拟社区系统，总体思路是通过采用Spring、SpringMVC和Mybatis三大框架及Bootstrap前端页面框架，搭建起包含着基本功能和良好的交互界面的该类系统。

## 2.2具体目标

虚拟社区系统的基本功能应实现的模块应包含一下几个模块并包含一定的功能，模块应具有良好的交互界面，给予用户良好体验。

社区首页模块：社区首页模块作为用户访问的第一界面，应具备展示论坛中帖子及评论的功能、用户登入入口、热门帖子的展示功能、搜索帖子功能、用户发帖功能、查看不同板块下的帖子功能和良好的界面设计。

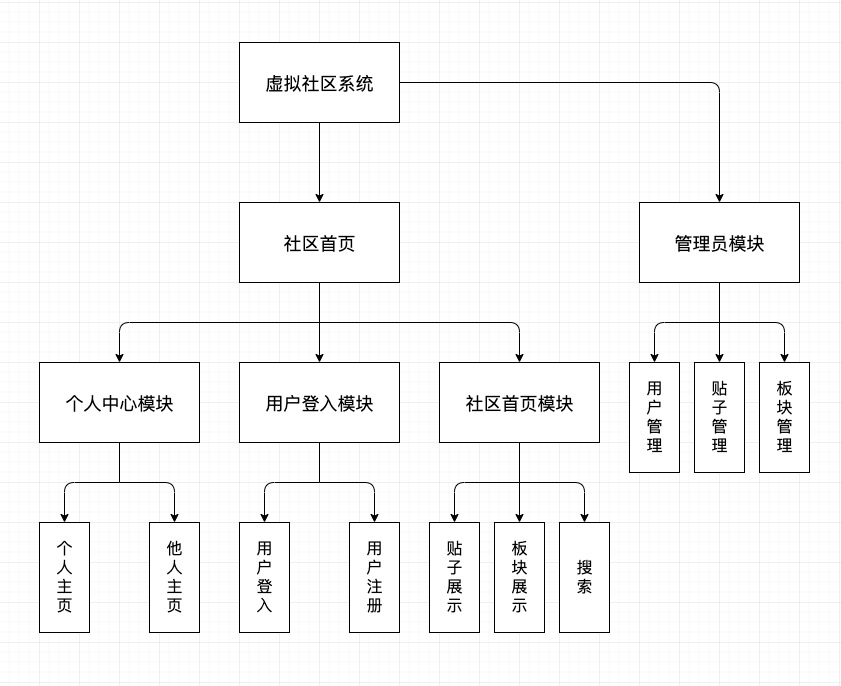
个人中心模块：个人中心模块作为用户主要的编辑和查看对象，该模块由不同主体分为两相似的页面——个人主页和他人主页。个人主页应依托着前端界面进行展示个人信息，并应提供用户更换头像的功能和更新个人资料的功能。此外，个人主页中应展示该用户的发帖、评论、关注和收藏信息。他人主页模块和个人主页功能大致相同，区别主要集中于用户只能观察到其他用户的信息和动态，无法查看他人的关注和收藏。

管理员模块：该模块作为社区管理员的使用主体，首先应具备登入功能。此外管理员在登入后可以进行三项功能操作，用户管理、帖子管理和板块管理。用户管理功能中，管理员可以查看用户的各项信息，并可以删除特定的用户。帖子管理中，管理员可以审核帖子的各项信息和其发布者，并根据审核结果对帖子进行通过或拒绝的操作，只有通过审核的帖子才能展示在社区首页。

用户登录注册模块：该模块作为用户进入本系统的主要入口。已注册的用户可以根据用户名和密码进行登入，而新用户可以通过注册来成为本论坛的一员，注册和登入时系统应审查用户的合法性。

## 2.3图示分析

分别从管理端、普通用户端设计图示



# 概要设计

## 3.1接口设计

### 3.1.1设计概述

在本次的JavaEE实训项目中，我们采取了SpringMvc框架以及Mybatis框架两者的结合的方式，由于本项目分为用户管理模块，帖子管理模块、管理员模块以及分区管理模块，因此想让这些模块之间互相协调工作，就必须在每个模块上开放以便其他模块调用的接口。从而使得这些模块能相对独立但又有条不紊的担任自己的职责。接口的数据传递基本都是由前端->Controller层->Service层->Dao层。

### 3.1.2用户管理模块

该模块是对用户的信息进行保存以及修改的模块，是整个项目的重要组成部分。几乎整个项目的功能模块都会与该模块存在依赖关系，例如在用户登录的时候就需要调用该模块去查询有没有存在此用户，该用户输入的账号和密码是否正确。对于注册功能，也需要调用该模块去添加用户。该模块提供对用户的多种查询接口，例如按用户昵称查询用户信息（selectByUserAll），按用户的在数据库中的id（唯一标识主键）来查询用户信息（getUserId）等，当然还向前端提供对于用户信息进行修改的接口，开发人员可根据项目内的Controller而文件夹下的各个Controller类中所写的接口进行调用便可完成相应的功能而无需担心底层的实现。

### 3.1.3 帖子管理模块

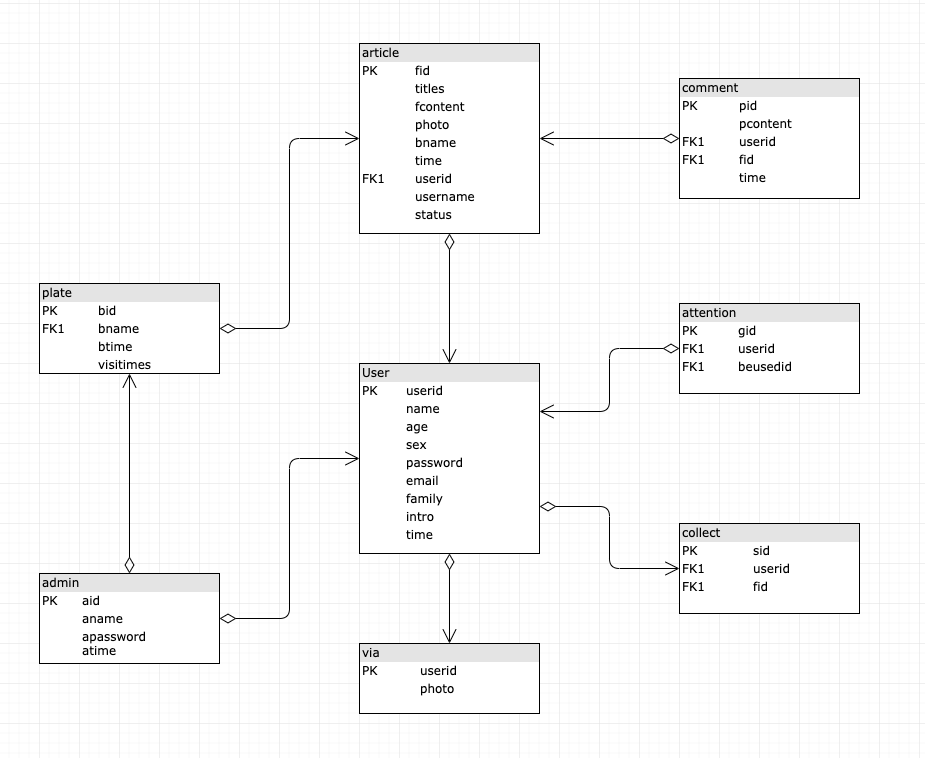
该模块与数据库之间的交流频繁，且每一次与数据库之间交互的数据量较大。该模块主要包括对帖子的内容的管理，对于发帖操作，我们在前端设置了一个发帖的界面，当用户提交帖子的时候，用户的基本信息、帖子内容甚至是多媒体文件都会被传到服务器端，服务器会将一条帖子的各个组成部分分开并行处理，对于上传的多媒体文件，我们在.jsp页面就已经根据依赖“commons-fileupload”提供了一个文件上传的接口，文件便会上传至服务器指定的资源路径中，对于帖子的文字信息内容，我们首先会根据发帖用户的昵称调用上述用户管理模块所提供的查询接口对用户的Id进行查询，得到id之后，便会将id与帖子文字数据一同打包并通过“ArticleController”提供的添加帖子接口与部署在服务器上的数据库进行交互，并将帖子内容、发帖人id等信息写入数据库，并返回操作结果。对于帖子的删除，与添加帖子的过程基本相同，在ArticleController层中提供deleteArticle接口，系统便会根据request对象来对相应帖子进行删除。当然最为重要的接口是对帖子的查询，我们在此提供了多个查询接口，可以利用getArticlebBname接口来取得指定板块下的所有帖子，也可以无条件式的利用getArticle接口来取得所有的帖子。该模块还提供帖子编辑的操作，updateArticle接口会满足您的要求。

### 3.1.4 分区管理模块

该模块较为简单，其实质上是属于管理员模块的一个子模块，但由于其对整个项目的实际意义以及价值发挥着不可忽略的作用，因此对其上接口的设计也需考虑周全。该模块的主要功能是对模块的添加、编辑以及删除（都需要管理员权限）。对于模块的添加，我们在前端的.jsp下编写了友好的页面专门供给用户使用，会要求管理员用户填写分区的标题。当管理员点击确认按钮时，分区的标题会流向两个方向，一个是流向数据库，通过生成的plate对象逐层传入数据库中，还有一个就是picgetter（图像抓取功能模块），即利用模块的名称从网上自动抓取一个图片显示为该模块的标识图片。

## 3.2数据库设计

### 3.2.1 E-R图



### 3.2.2关系模式

管理员（管理员id，管理员用户名，管理员密码，管理员注册时间）

帖子（帖子id，帖子标题，帖子内容，帖子图片，所属板块名称，发帖时间，发帖用户id，发帖人用户名，审核状态）

关注（关注表id，关注者id，被关注者id）

收藏（收藏id，收藏用户id，被收藏帖子id）

评论（评论id，评论内容，发表评论的用户id，被评论的帖子id，评论时间）

板块（板块id，板块名称，板块建立时间，板块访问次数）

用户（用户id，用户名，年龄，性别，用户密码，邮箱，住址，个人简介，注册时间）

头像（用户id，头像）

### 3.2.3数据库管理系统

Mysql 5.7

### 3.2.4数据库命名

bbs\_test

### 3.2.5数据库表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 管理员信息表：admin | | | | |
| 字段名称 | 数据类型 | 可空 | 默认值 | 说明 |
| aid | int(11) | 不可 |  | 主键，管理员id，每次自增1 |
| aname | varchar(255) | 不可 |  | 管理员用户名 |
| apassword | varchar(255) | 不可 |  | 管理员密码 |
| atime | timestamp | 不可 |  | 管理员注册时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 帖子信息表：article | | | | |
| 字段名称 | 数据类型 | 可空 | 默认值 | 说明 |
| fid | int(11) | 不可 |  | 主键，帖子id，每次自增1 |
| titles | varchar(100) | 不可 |  | 帖子标题 |
| fcontent | varchar(2000) | 不可 |  | 帖子内容 |
| photo | varchar(255) | 不可 |  | 帖子图片 |
| bname | varchar(255) | 不可 |  | 所属板块名称 |
| time | timestamp | 不可 |  | 发帖时间 |
| userid | int(11) | 不可 |  | 发帖用户id |
| username | varchar(45) | 不可 |  | 发帖人用户名 |
| status | int(11) | 不可 |  | 审核状态(0-待审核，1-通过审核，2-未通过审核) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 关注信息表：attention | | | | |
| 字段名称 | 数据类型 | 可空 | 默认值 | 说明 |
| gid | int(11) | 不可 |  | 主键，关注id，每次自增1 |
| userid | int(11) | 不可 |  | 关注者id |
| beuserid | int(11) | 不可 |  | 被关注者id |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 收藏信息表：collect | | | | |
| 字段名称 | 数据类型 | 可空 | 默认值 | 说明 |
| sid | int(11) | 不可 |  | 主键，收藏id，每次自增1 |
| userid | int(11) | 不可 |  | 收藏用户id |
| fid | int(11) | 不可 |  | 被收藏帖子id |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评论信息表：comment | | | | |
| 字段名称 | 数据类型 | 可空 | 默认值 | 说明 |
| pid | int(11) | 不可 |  | 主键，评论id，每次自增1 |
| pcontent | varchar(400) | 不可 |  | 评论内容 |
| userid | int(11) | 不可 |  | 发表评论的用户id |
| fid | int(11) | 不可 |  | 被评论的帖子id |
| time | timestamp | 不可 |  | 评论时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 板块信息表：plate | | | | |
| 字段名称 | 数据类型 | 可空 | 默认值 | 说明 |
| bid | int(11) | 不可 |  | 主键，板块id，每次自增1 |
| bname | varchar(255) | 不可 |  | 板块名称 |
| btime | timestamp | 不可 |  | 板块建立时间 |
| visitimes | int(11) | 不可 |  | 板块访问次数 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户信息表：user | | | | |
| 字段名称 | 数据类型 | 可空 | 默认值 | 说明 |
| userid | int(11) | 不可 |  | 主键，用户id，每次自增1 |
| name | varchar(45) | 不可 |  | 用户名 |
| age | int(11) | 可空 |  | 年龄 |
| sex | varchar(255) | 可空 |  | 性别 |
| password | varchar(45) | 不可 |  | 用户密码 |
| email | varchar(45) | 不可 |  | 邮箱 |
| family | varchar(45) | 可空 |  | 住址 |
| intro | varchar(255) | 可空 |  | 个人简介 |
| time | timestamp | 不可 |  | 注册时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 头像信息表：via | | | | |
| 字段名称 | 数据类型 | 可空 | 默认值 | 说明 |
| userid | int(11) | 不可 |  | 主键，用户id |
| photo | varchar(255) | 不可 |  | 头像 |

## 3.3运行设计

项目模块：用户管理模块、分区管理模块、帖子管理模块。

运行关系：相互交织，围绕着帖子管理模块运行。

运行时间分析：

用户管理模块的相应时间相对较短，在整个项目中所调用的次数较少，而且查询出来的信息会被存储在用户与服务器的会话域（session）中，加之用户（User）类所包含的数据对象较少，故该模块的响应时间比较少，速度较快。

分区管理模块的运行时间在其他两者之间，首先其在项目中所被调用的次数较为平凡，每当用户进入或者返回主页面的时候，该模块都会被调用，而且该模块还需要加载一些图片信息，所以响应时间会相对较长。

帖子管理模块是运行时间最长的，在整个项目中处于核心地位，而且其对数据库的操作涉及到多个表之间的联接查询，而且其数据量比较大，记录着不仅仅是用户本身的帖子信息，还包含着其他用户的帖子信息，在加上对评论文字信息的查询及加载，因此其对整个项目的运行复杂度以及用户体验起到了一个举足轻重的作用。

## 3.4出错处理设计

3.4.1 概述

在项目在运行的时候总避免不了出现各种各样的错误，其中有可能是机器故障，程序异常，也有可能是人为的异常。但评判一个程序的好坏的最主要依据就是这个程序是否稳定。那么我们在设计项目的时候就需要考虑多种可能出现的意外情况，并策划相应的应对机制。

3.4.2 设计介绍

首先是登陆模块，用户的登陆是一个较为敏感的操作，因此这里的安全性要相对较高。若用户没有填写用户名或者密码时，登陆窗口便会给出相应的提示，若用户名或者密码错误，则会提示用户用户名或密码错误。

在注册的时候，系统要求用户必须输入用户名、密码以及邮箱，否则注册请求不会被通过并会给出提示，确认密码与密码两栏的输入必须一致，对邮箱一栏的输入也会有检测，若不符合上述要求，都会给出提示。当注册请求被服务器接收到时，便会给用户所提供的邮箱发送一个带有注册验证码的邮件，网页上会出现让用户输入去邮箱验证码的对话框，若输入的验证码与实际不符，则会提示验证码错误。

在图片的爬取操作同样有容错机制，当由于网络原因导致目标src请求不到时，系统便会放弃这个操作转而执行后面的操作。总而言之，该系统的容错机制以提示或者跳过为主，当然一些重要的错误会反馈到服务器管理者的控制台上。

# 模块分工

## 4.1登录页面

在首页可以从右上角进行用户的登录，输入用户名和密码，再点击登录框后，后端采用ajax进行登录的验证，正确则成功登录，可以继续进行其它登录后的操作，错误则返回错误信息“用户名或密码错误”，提示用户重新输入，还提供了清除框，可以清除用户输入的信息，未注册过的用户还可以点击立即注册进行用户的创建。

## 4.2注册页面

注册页面要求用户输入用户名、密码、确认密码和电子邮箱，包含以下完整的验证，首先，用户名不得重复，若重复会提示“该用户名已存在！”，其次，要求两次密码输入必须一致，以防用户因误输入密码导致以后无法登录，若两次密码不一致则会提示“两次密码不一致”，然后对邮箱的输入也有限制，必须符合邮箱的格式，格式正确后会往指定邮箱发送验证码，验证正确后才可成功注册，同样地，也提供了清除框清除用户输入的全部信息。

## 4.3发帖页面

在用户登录后，即可进行发帖，一个完整的帖子包括标题、所属板块、内容以及附件，板块只能选择系统中有的板块，附件可以是图片或者视频，视频只支持.mp4和.avi格式，用户发帖后，帖子不会立刻在首页显示，而是需要管理员在后台审核后再决定是否通过，同时，管理员可以随时撤销用户所发的帖子。

## 4.4个人设置

个人设置出现在两处，一处是首页，一处是个人主页，我负责编写的是首页的设置，用户根据需要可以修改自己的用户名、密码和电子邮箱。

## 4.5数据库设计

项目开工第一天领导组员完成了数据库的设计，共设计了八个表，涵盖了管理员、帖子、关注、收藏、评论、板块、用户、头像所需要的数据表，详细内容在概要设计的数据库设计中均已写出。

## 4.6前端Bootstrap框架

本次项目开发，我们后端使用的是老师所教的SSM框架，在前端框架的选择上，我最终确定使用Bootstrap，它来自Twitter，基于HTML、CSS、JAVASCRIPT，通过使用Bootstrap，可以轻松创建一个美观的前端网页，不需要在前端布局上花费太多功夫，使得我能够更专注于后端的开发。

## 4.7页面适配

本系统不仅可以在电脑上使用，还能够在移动端如手机、平板电脑上使用，考虑到手机屏幕没有电脑屏幕那么大，如果采用一致的布局，会给用户的使用带来不少麻烦。因此，面向移动端针对页面显示进行了优化，方便用户的操作，其实现方法是系统根据窗口大小判定，所以如果在电脑上把网页窗口缩小，也会得到和手机端一样的界面。

# 实习收获(心得体会)

队员1：

队员2：

队员3：

# 致谢