# 目 录

1. [绪 论 1](#_bookmark0)
   1. [研究的背景及意义 1](#_bookmark1)
      1. [选题的背景 1](#_bookmark2)
      2. [国内外研究现状 1](#_bookmark3)
      3. [研究的意义 2](#_bookmark4)
   2. [系统目标 3](#_bookmark5)
2. [需求分析 4](#_bookmark6)
   1. [业务需求 4](#_bookmark7)
      1. [主要业务流程 4](#_bookmark8)
   2. [功能需求 6](#_bookmark9)
      1. [角色分析 6](#_bookmark10)
      2. [业务功能 7](#_bookmark11)
   3. [非功能需求 17](#_bookmark12)
      1. [环境需求 17](#_bookmark13)
      2. [性能需求 17](#_bookmark14)
      3. [安全需求 18](#_bookmark15)
3. [总体设计 19](#_bookmark16)
   1. [系统设计的原则 19](#_bookmark17)
   2. [系统体系结构设计 20](#_bookmark18)
   3. [系统功能结构设计 21](#_bookmark19)
4. [数据库设计 23](#_bookmark20)
   1. [概念结构设计 23](#_bookmark21)
      1. [设计思路 23](#_bookmark22)

[4.1.2 E-R 图 24](#_bookmark23)

* 1. [逻辑结构设计 28](#_bookmark24)
     1. [设计思路 28](#_bookmark25)
     2. [逻辑模型 28](#_bookmark26)
  2. [物理结构设计 31](#_bookmark27)
     1. [存取方式 31](#_bookmark28)
     2. [存储结构 31](#_bookmark29)

1. [界面设计 32](#_bookmark30)
   1. [界面关系图或工作流图 32](#_bookmark31)
      1. [用户端界面关系图 32](#_bookmark32)
      2. [后台管理界面关系图 32](#_bookmark33)
   2. [界面设计成果 33](#_bookmark34)
      1. [主界面 33](#_bookmark35)
      2. [子界面 35](#_bookmark36)
2. [详细设计 40](#_bookmark37)
   1. [系统主要功能模块介绍 40](#_bookmark38)
   2. [用户管理功能模块设计 41](#_bookmark39)
      1. [功能结构设计 41](#_bookmark40)
      2. [类图设计 41](#_bookmark41)
      3. [顺序图设计 44](#_bookmark42)
      4. [核心处理流程设计 44](#_bookmark43)
   3. [兼职职位管理功能模块设计 45](#_bookmark44)
      1. [功能结构设计 45](#_bookmark45)
      2. [类图设计 45](#_bookmark46)
      3. [顺序图设计 47](#_bookmark47)
      4. [核心处理流程设计 48](#_bookmark48)
   4. [论坛帖子管理模块 49](#_bookmark49)
      1. [功能结构设计 49](#_bookmark50)
      2. [类图设计 49](#_bookmark51)
      3. [顺序图设计 51](#_bookmark52)
      4. [核心处理流程设计 52](#_bookmark53)
   5. [广告管理模块 53](#_bookmark54)
      1. [功能结构设计 53](#_bookmark55)
      2. [类图设计 53](#_bookmark56)
      3. [顺序图设计 55](#_bookmark57)
      4. [核心处理流程设计 56](#_bookmark58)
   6. [系统管理模块 57](#_bookmark59)
      1. [功能结构设计 57](#_bookmark60)
      2. [类图设计 57](#_bookmark61)
      3. [顺序图设计 62](#_bookmark62)
      4. [核心处理流程设计 63](#_bookmark63)
   7. [聊天管理模块 64](#_bookmark64)
      1. [功能结构设计 64](#_bookmark65)
      2. [类图设计 64](#_bookmark66)
      3. [顺序图设计 66](#_bookmark67)
      4. [核心处理流程设计 67](#_bookmark68)
   8. [收藏管理模块 68](#_bookmark69)
      1. [功能结构设计 68](#_bookmark70)
      2. [类图设计 68](#_bookmark71)
      3. [顺序图设计 70](#_bookmark72)
      4. [核心处理流程设计 71](#_bookmark73)

[7 **编码** 72](#_bookmark74)

* 1. [代码实现与核心算法 72](#_bookmark75)
     1. [代码的目录结构与接口 72](#_bookmark76)
     2. [核心算法代码实现 73](#_bookmark77)
  2. [代码优化分析 76](#_bookmark78)

[8 **测试** 78](#_bookmark79)

* 1. [测试方案设计 78](#_bookmark80)
     1. [测试策略 78](#_bookmark81)
     2. [测试进度安排 78](#_bookmark82)
     3. [测试资源 78](#_bookmark83)
     4. [关键测试点 78](#_bookmark84)
  2. [测试用例构建 78](#_bookmark85)
     1. [测试用例编写约定 78](#_bookmark86)
     2. [测试用例设计 79](#_bookmark87)
     3. [关键测试用例 79](#_bookmark88)
     4. [测试用例维护 80](#_bookmark89)

1. [**总结与展望** 81](#_bookmark90)
   1. [设计工作总结 81](#_bookmark91)
   2. [未来工作展望 81](#_bookmark92)

[**谢 辞** 82](#_bookmark93)

[**参考文献** 83](#_bookmark94)

[附录 A 外文翻译—原文部分 84](#_bookmark95)

[附录 B 外文翻译—译文部分 89](#_bookmark96)

[附录 C 软件使用说明书 93](#_bookmark97)

[附录 D 主要源代码 96](#_bookmark98)

**1** **绪 论**

### 研究的背景及意义

* + 1. 选题的背景

在我们国家近些年，不断地关心和重视对我们年轻一代的高素质教育，我们全国的高校在校人数也是同样的一年比一年多。同时也是随着时代科技、我们社会的市场经济形势的大力发展、大家传统观念的改变，在校大学生兼职其实已经在我们高校学生当中成为了一种喜闻乐见的现象。我们的高校生在自己的课外业余时间，在自己力所能及的范围找一份兼职，不仅能够让我们的学会如何与同事相处，还能够让我们提前适应日后的工作，也使得我们有一定的社会工作经验。尤其是近些年来，就业的形式愈发的严重， 例如再加上今年的的疫情影响，我们就业的前景更加的艰难。因此，高校的在校学生利用自己的课余时间去兼职，不仅可以积累工作的经验，对我们的自我能力也有不小的提高帮助。同时还能够或多或少的有一定的经济收入，帮助缓解家长们的经济压力负担。从整体来看，大学生兼职对于我们自己本身所带来的益处和帮助还是不少的，所以大学生兼职现象将会日益的增加，发展的趋势也终将会逐步的越来越壮大。

在目前的全国所有的高校里，学校和社会方面其实对于我们大学生利用课外的时间

去兼职的态度一直是模棱两可的，既不大力支持同时也不反对。就比如说在我们学校， 虽然学校会提供一些勤工俭学的机会，但是这样的岗位也很少，其次工资薪水也不是很高。因此就有很多人会选择到校外找各种兼职。这样一来又会产生很多的隐患，比如说出行的交通安全问题，身边甚至还会有很多的同学因为找兼职工作遇到诈骗等事件，与此同时长期以来，很多在校外兼职的同学也会因兼职耽误自己的学业。总的来说，兼职对于我们大学生的权益还是会存在一定的损害。而且我们的这一地带有众多的高 校，地方也处于南昌的近郊，环境也比较的偏，很多的同学要找兼职工作的话还需花费时间去市内或者附近寻找，不仅费时更加的费力。回到兼职工作本身来说，从事一份与自己专业相关的工作好处往往大于与自己专业无关，对我们在四年后的毕业走向工作岗位有很大的帮助。

* + 1. 国内外研究现状

在世界上较早开始实行高等教育的欧美发达国家，其教育管理机制也相对独立，大学之间的竞争也很激烈，在这种健康的竞争环境下，我们的同学也有了很大的进步空间和动力。同时在国外的高等教育中，一名普通的学生要完成他自己的学业所需要的学费

生活费等加起来也是个不菲的数目。对于一个普通的家庭来说，这也是个不小的负担。也就是说，我们有很多的国外的高校生必须在利用周末放假的时候去从事一些力所能及的兼职工作，以便于有一定的经济收入。所以甚至在一些私立大学中甚至还设有这方面专业的老师来指导学生如何去兼职，以便让学生提前有机会体验他们未来的专业工作风格和工作环境，解决一定的经济负担和压力。并为他们的未来制定良好职业规划。并且随着互联网技术的逐渐发展，每所学校都有自己相对完善的平台管理系统，这为学校的学生提供了很大的帮助在兼职方面。

但是在我们的国内，很大原因的受我们传统思维的影响,很多人觉得我们大学生在校园里就应该好好的专心于我们的学业。这可能是中国的家庭式教育的一个很普遍的观念，从小学到大学，我们作为家中的未来的希望，家长们对我们孩子的唯一的任务要求就是把学业负责好，其他的事都是家长在身后保驾护航，为我们处理好，更不用说让我们的孩子出去兼职工作了。但是近些年来，随着我们国家大力倡导我们的在校大学生创新创业，已经有很多的大学生慢慢的开始在校兼职工作了，以便积累一些工作实习的经验，同时社会的观念也慢慢的改变了。比如说在我们的身边也出现了很多的兼职相关的APP，比如说 58 同城、猎聘、斗米兼职等。很好的帮助我们的在校生省去了甚多在寻找兼职工作岗位的路上的麻烦。

* + 1. 研究的意义

网络技术愈发成熟，互联网技术与生活各个方面相结合，科技的发展对我们很多传统的工作等事务也带来了很大的变化和便利，设计基于 SSM 框架技术高校大学生兼职平台，在我们大学生的课外兼职方面主要有两个方面的优化。一方面，系统化处理我们的招聘信息的发布，可以减少人工处理常见的错误，巧妙的安排是人工的优势，能够省去我们在校的高校生用于找兼职信息的时间，并且很大的程度上保证了兼职信息的真实性和准确性。另一方面，自动化管理我们的招聘求职信息的发展必然趋势，自动化管理系统能够有效的节省人力资源，将会被自动化系统取代。可以使得我们的在校大学生在找兼职工作时更加的便捷、放心，真正的做到让我们的商家和兼职学生各取所需。高校大学生兼职平台的构建，不仅减轻了管理者的工作，还大大的推进了学生们的从事兼职。

同时对于我国的长远发展来讲，为了实现我们强国梦，我们的当代大学生可是这其中的中坚力量。因此就需要我们这中坚力量做到全面的发展，能够负担的起这一伟大而艰巨的任务。对我们的年轻一代学子们的培养，不仅仅只是在课本的书面知识上纸上谈兵式教育，更多的是要去现实当中实践。而我们的在校大学生的兼职行为，就很好的锻炼了大家的实践动手能力。无论是从事与自己专业相关联的兼职也好，还是与自己专业

无关的兼职也罢，都对我们的能力有很大的提升。当从事与专业相关的时，这样一来可以更加精尖我们的专业技能，在专业领域方面多贡献自己的一份力量；当无关时，也可以更加锻炼我们的工作能力、磨练我们的意志。不管从什么角度出发，兼职对于我们的在校大学生来讲，是对他们的大学教育的一个补善，促进我们的大学生的综合全面发展。

### 系统目标

我们的平台基于 B/S 架构，完成一个功能完整全面、界面交互友好人性化、操作简洁的高校大学生兼职平台。

以通过高校大学生兼职平台，我们的商家可通过在自己的账号登录到前台页面， 按照职务种类发布自己的兼职岗位需求，平台管理员会在后台对兼职信息进行审核，对兼职信息进行分类，为我们的在校大学生用户及时的提供兼职信息咨询服务和信息的发布。在浏览在我们附近的兼职职位信息时，对有意向的可以和发布者进行进一步的讨论， 以及在论坛可以随时查看其他同学的发帖并可以评论、收藏，极大的方便了大家的交流。同时也对我们的平台的游客可以查看职位列表、浏览职位信息、查看平台公告、查看论坛。最后我们的用户对于平台有任何的反馈意见都可以通过平台邮箱向我们平台系统反应。管理员在我们的后台登录后的后台管理界面，可以对我们平台的用户信息等管理、兼职信息管理、论坛帖子管理、聊天管理、广告管理、信箱管理、收藏管理以及系统的轮播列表、公告列表和系统设置，更加公开透明的保证了所有信息的真实性。

# 需求分析

### 业务需求

（一）一个没有注册的普通游客或者没有登陆的用户进入到平台的前台展示页面时，大家可以在平台的首页查看平台发布所有的兼职岗位需求信息，同时还能够分类兼职列表进而浏览职位信息，同时也可以点击查看平台公告以及查看浏览我们的兼职论坛中的帖子，但是作为游客或者还未登录的用户时只能够对论坛中的帖子进行浏览，并不能够进行发表或者评论。

（二）当我们的平台用户通过自己的账号和密码登录到平台时，可以在平台的首页查看平台发布所有的兼职岗位需求信息，同时还能够分类兼职列表进而浏览职位信息，对于自己感兴趣的岗位可点开与招聘发布者进行交流。同时也可以点击查看平台公告以及查看浏览我们的兼职论坛中的帖子，在看到有关的论坛发帖时可进行个人的评论，同时评论的内容平台的其他用户或者普通游客都能够看的见。在看到有关感兴趣的帖子或者是重要的帖子同时也可以进行点击收藏。在个人中心能够查看交流信息、查看个人收藏、修改个人信息、查看个人发帖等。

（三）具有配套的系统后台管理维护功能，在平台的管理员登入金后台管理系统后，可以为系统的管理员提供用户管理（添加平台用户、查询用户、用户权限设置、用户资格修改）、兼职信息的审核管理（兼职发布信息审查、查询、编辑、禁用）、对用户在论坛的发帖管理（论坛发帖的审查、查询、编辑、禁用以及相关帖子的评论查看）、招聘的聊天信息管理（用户之间聊天信息审查、查询、编辑、禁用）、平台广告管理（添加平台广告、查询广告、广告编辑修改）、用户的建议反馈信箱管理（平台建议信箱的查询、编辑、删除）、帖子的收藏信息管理（用户帖子收藏信息记录的查询、编辑、删除）、以及可以对我们系统的轮播列表、广告列表管理和系统设置，响应速度迅速，安全性能较高。

* + 1. 主要业务流程

（一）游客浏览活动图 如图 2-1 所示

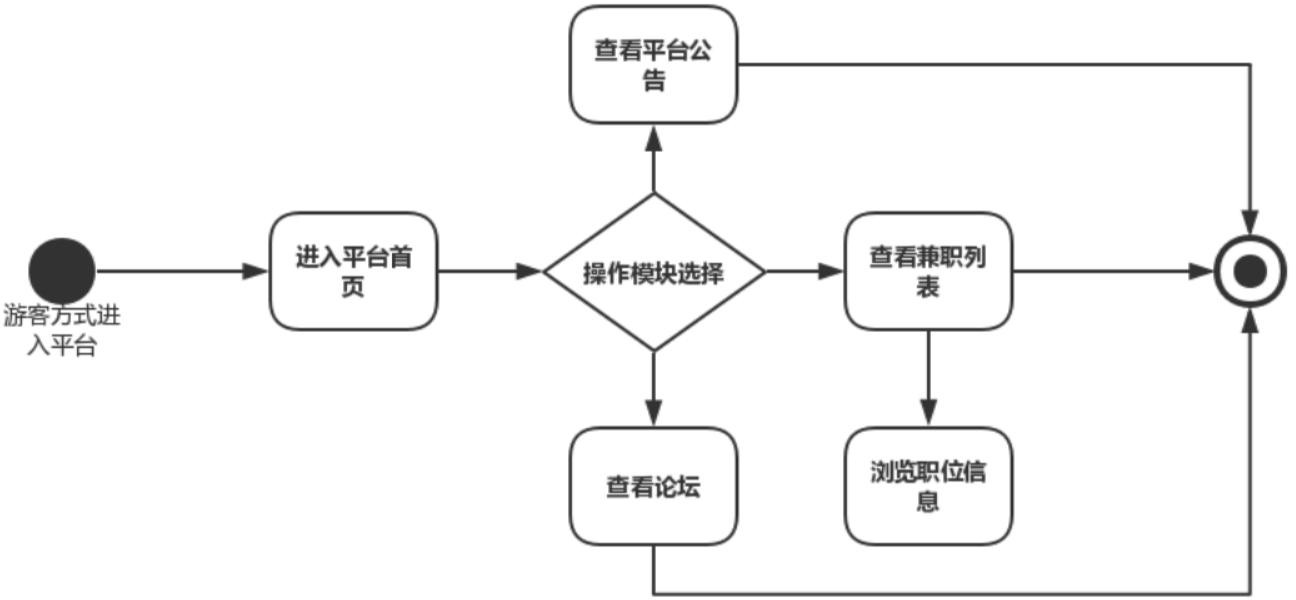


图 2-1 游客浏览活动图

（二）平台用户浏览平台活动图 如图 2-2 所示

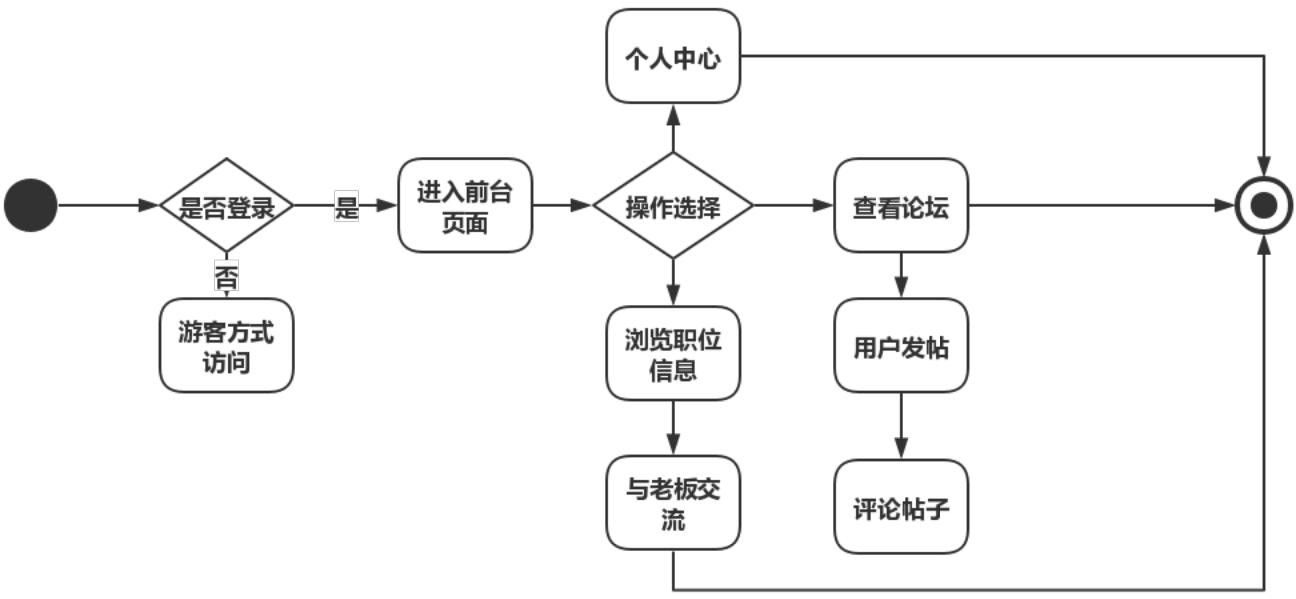


图 2-2 平台用户浏览平台活动图

（三）后台管理活动图 如图 2-3 所示

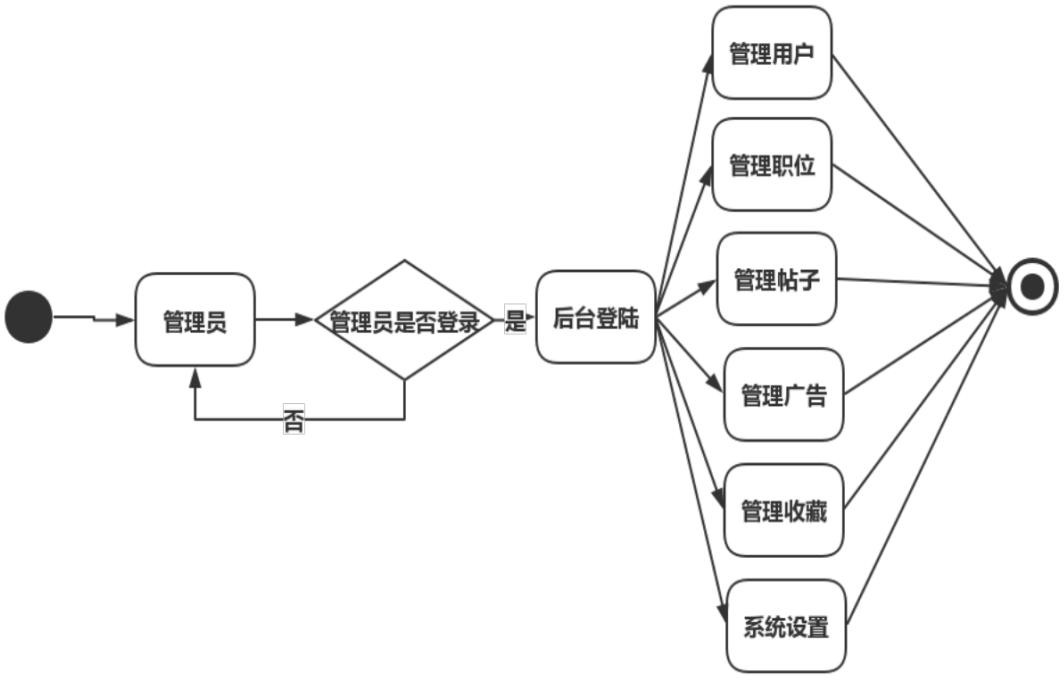


图 2-3 后台管理活动图

### 功能需求

* + 1. 角色分析

以下从大学生兼职平台系统的功能角度出发，系统总共存在三大类角色：普通游客用户、平台注册用户、系统管理员，具体的分析如下表所示：

表 2-1 系统角色表

|  |  |
| --- | --- |
| 角色 | 职责或功能 |
| 平台用户 | 查看兼职列表、与发布简直信息的老板进行沟通、浏览职位信息、查  看平台公告、产看论坛、发表帖子、对帖子进行评论、在个人中心查 看个人信息同时也可修改，查看他人发的信息，以及删除你的帖子等 |
| 游客用户 | 查看兼职列表、浏览职位信息、查看平台公告和论坛 |
| 系统管理员 | 在前台页面中管理员拥有的操作和功能与我们的用户差不多，能够查看兼职列表、与发布简直信息的老板进行沟通、浏览职位信息、查看平台公告、产看论坛、发表帖子、对帖子进行评论、在个人中心查看个人信息，查看他人发的信息，以及删除你的帖子等。同时还能够在后台管理页面中管理我们的用户、管理职位、管理帖子、管理广告、 管理用户的收藏、查看聊天信息、以及一些平台系统的设置，例如轮  播列表 |

* + 1. 业务功能

根据我们系统平台存在的三大类主要角色，得出了我们得系统总体用例图，总共有查看用户管理、兼职职位管理、论坛帖子管理、广告管理、系统管理、聊天管理、个人中心这几个主要的模块，如下图 2-4 所示：

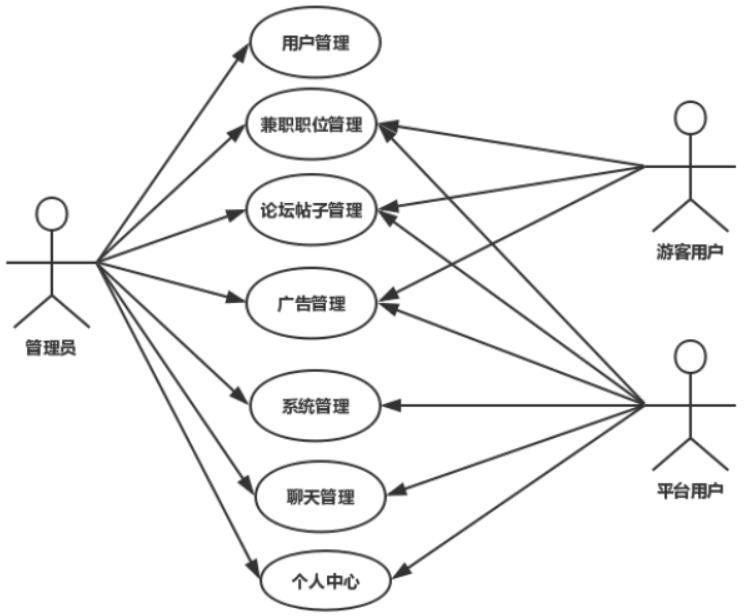


图 2-4 系统总体用例图

主要业务分有以下几个：

1. 用户管理

用户管理主要分为管理员在我们的后台管理系统中可手动添加新建平台用户、用户查询、用户信息的修改、用户的禁用或者删除，同时我们的游客也可在前台登录页面进行注册。用户管理用例图如图 2-5 所示：

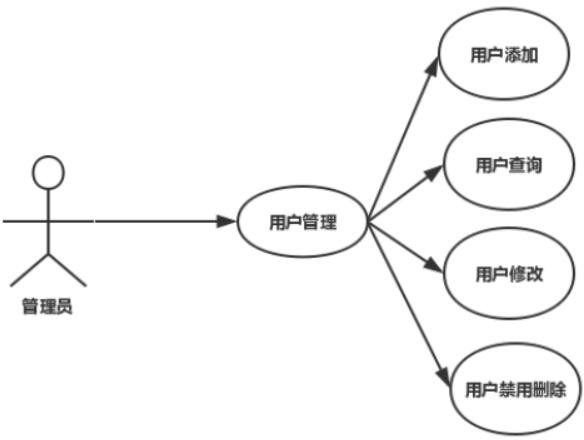


图 2-5 用户管理用例图

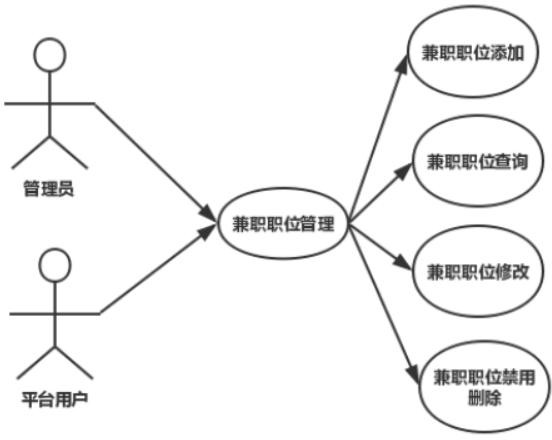
用例描述:

表 2-2 平台用户添加用例表

|  |  |
| --- | --- |
| 用例名称 | 平台用户添加 |
| 用例标识符 | YL01 |
| 执行者 | 管理员、平台用户 |
| 简要描述 | 管理员添加平台用户信息，对添加的用户信息进行规范检查后，然后保存  提交信息 |
| 前置条件 | 1. 管理员使用自己的账号密码成功登陆后台管理页面 2. 平台游客在前台的登录页面注册成为平台用户 |
| 基本事件流 | 1. 管理员在后台管理页面中选择点击用户管理，下拉框选择用户列 表，然后点击添加用户按钮 2. 按照要求输入用户的信息点击提交 3. 系统平台对输入的信息进行规范性验证 |
| 后置条件 | 添加成功 |

1. 兼职职位管理

兼职职位管理主要是对我们平台有兼职职位发布的商家所发布的职位信息进行管 理，包括兼职职位的添加、兼职职位的查询、兼职职位的编辑修改和兼职职位的禁用删除。兼职职位管理用例图如图 2-6 所示：



用例描述:

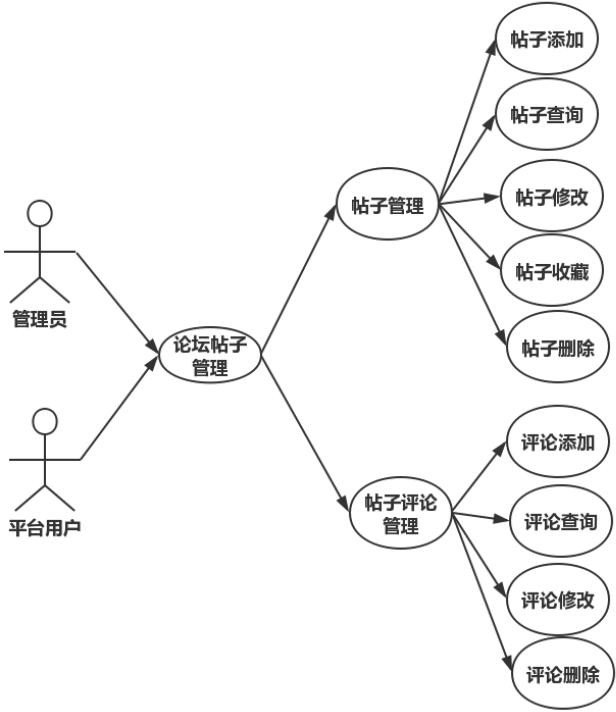
图 2-6 兼职职位管理用例图

表 2-3 兼职职位信息删除用例表

|  |  |
| --- | --- |
| 用例名称 | 兼职职位删除 |
| 用例标识符 | YL02 |
| 参与者 | 管理员 |
| 简要描述 | 管理员对无效的兼职职位信息删除系统 |
| 前置条件 | 管理员使用账号密码成功登陆后台，兼职职位信息已经存在平台数据中 |
| 基本事件流 | 1. 管理员在后台管理页面点击兼职职位管理，下拉框选择兼职列表 2. 选择要删除的兼职职位信息，点击禁用 |
| 后置条件 | 显示禁用成功 |

1. 论坛帖子管理

论坛帖子管理主要包括两大部分，用户的发帖管理和评论管理。包括论坛所发贴的增加、内容的查看、内容的修改、帖子的删除和帖子的收藏；评论管理包括评论的添加、评论的查询、论坛发帖的评论的编辑和帖子评论的删除。论坛帖子管理用例图如图 2-7 所示：



用例描述:

图 2-7 论坛帖子管理用例图

表 2-4 论坛帖子修改用例表

|  |  |
| --- | --- |
| 用例名称 | 论坛帖子修改 |
| 用例标识符 | YL3 |
| 执行者 | 管理员 |
| 简要描述 | 管理员对论坛帖子的信息进行修改保存 |
| 前置条件 | 管理员使用自己的账号密码成功登陆后台 |
| 基本事件流 | 1. 管理员在后台管理页面点击论坛帖子管理，下拉框选择论坛帖子列 表 2. 在需要需改的论坛帖子处点击编辑按钮，对需要修改的信息进行修 改 3. 系统平台对修改后的信息进行验证，然后点击保存提交 |
| 后置条件 | 修改成功 |

表 2-5 论坛帖子收藏用例表

|  |  |
| --- | --- |
| 用例名称 | 论坛帖子收藏 |
| 用例标识符 | YL4 |

续表 2-5

|  |  |
| --- | --- |
| 参与者 | 管理员 |
| 简要描述 | 管理员对自己感兴趣的论坛帖子进行收藏操作 |
| 前置条件 | 平台用户使用自己的账号密码成功登陆前台操作展示页面 |
| 基本事件流 | 1. 用户进入平台的前台展示页面 2. 在平台论坛进行帖子浏览 3. 对自己感兴趣的帖子点击收藏 |
| 后置条件 | 显示收藏成功 |

表 2-6 评论查询用例表

|  |  |
| --- | --- |
| 用例名称 | 评论查询 |
| 用例标识符 | YL5 |
| 执行者 | 管理员、平台用户 |
| 简要描述 | 管理员查询评论信息，或者平台用户在前台展示页面查看评论信息 |
| 前置条件 | 1. 管理员用自己的账号密码成功登陆后台，评论信息记录已成功保存 2. 平台用户使用自己的账号密码登录前台页面 |
| 基本事件流 | 1. 管理员在后台管理系统中点击评论管理模块，下拉框选择评论列表， 点击评论查询或者用户在前台展示页面查看评论信息 2. 管理员按要求输入查询关键字或者关键条件点击查询 3. 展示出按条件查询出来的评论信息 |
| 后置条件 | 查看成功 |

1. 广告管理

广告管理主要是对我们平台有广告发布合作的商家所发布的广告信息进行管理，包括广告的添加、广告的查询、广告的编辑修改和广告的禁用删除。广告管理用例图如图2-8 所示：

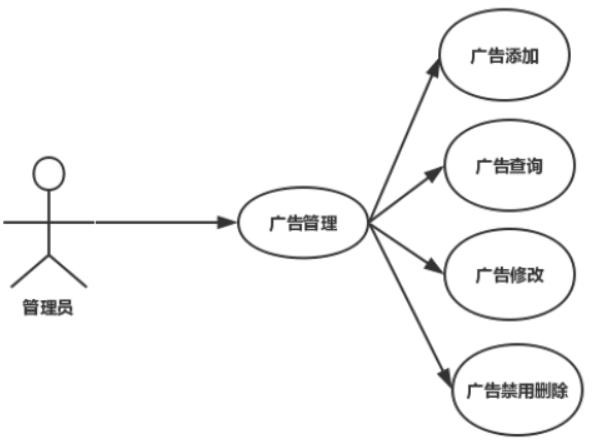


图 2-8 广告管理用例图

用例描述:

表 2-7 广告添加用例表

|  |  |
| --- | --- |
| 用例名称 | 广告添加 |
| 用例标识符 | YL6 |
| 执行者 | 管理员、平台用户 |
| 简要描述 | 管理员添加广告信息，对添加的信息进行规范检查后，然后提交信息 |
| 前置条件 | 1.管理员使用自己的账号密码成功登陆后台管理页面 |
| 基本事件流 | 1. 管理员在后台管理页面中选择点击广告管理，下拉框选择广告列表， 然后点击添加广告按钮 2. 按照要求输入广告信息内容点击提交 3. 系统平台对输入的广告信息内容进行规范性验证 |
| 后置条件 | 添加成功 |

1. 系统管理

平台管理员在后台系统可对平台进行系统管理，主要分为三大模块，分别是轮播列表管理、公告列表管理和系统设置。轮播列表管理包括轮播添加、轮播查询、轮播修改和轮播删除；公告列表管理包括公告添加、公告查询、公告修改和公告删除；系统设置包括平台名称、平台地址、欢迎语、联系方式设置。系统管理用例图如图 2-9 所示：

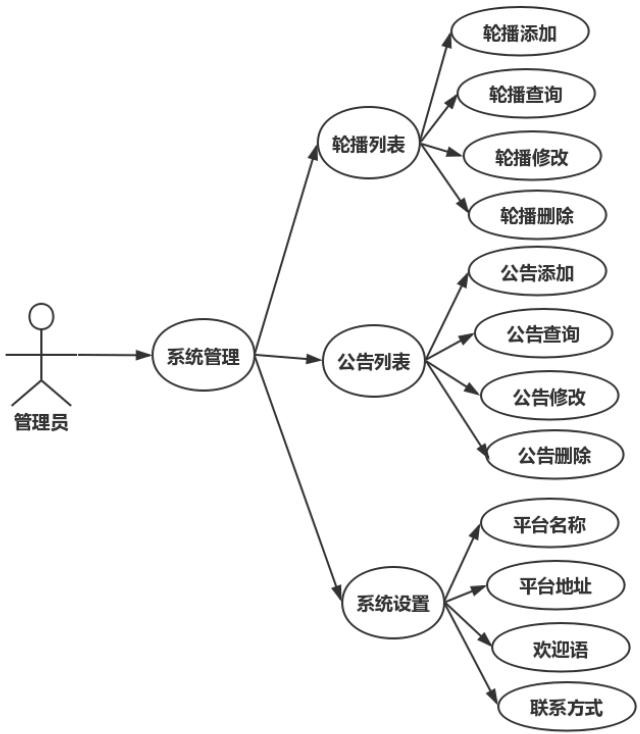


图 2-9 系统管理用例图

用例描述:

表 2-8 系统设置用例表

|  |  |
| --- | --- |
| 用例名称 | 系统设置 |
| 用例标识符 | YL7 |
| 参与者 | 管理员 |
| 简要描述 | 管理员对我们的平台前台展示相关信息进行设置 |
| 前置条件 | 管理员使用账号密码成功登陆后台，公告信息已经存在平台系统内 |
| 基本事件流 | 1. 管理员在后台管理页面点击系统管理，下拉框选择系统设置 2. 分别对我们的平台名称、平台地址、欢迎语和联系方式等进行编辑设置 |
| 后置条件 | 显示设置成功 |

1. 聊天管理

聊天管理主要是对我们平台的用户和商家就兼职职位相关问题的沟通聊天信息进行管理，包括聊天发起、聊天记录的查询、聊天记录的编辑修改和聊天记录的禁用删除。聊天管理用例图如图 2-10 所示：

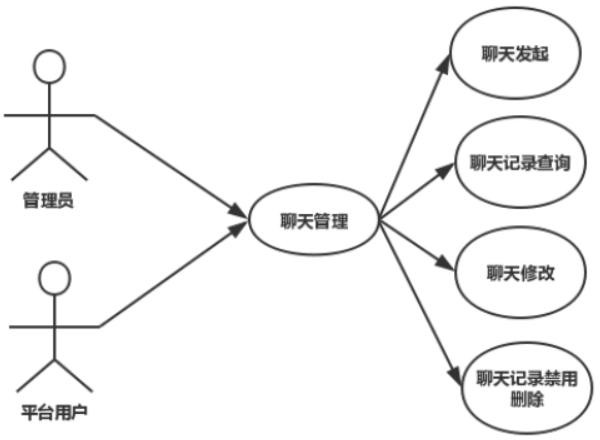


图 2-10 聊天管理用例图

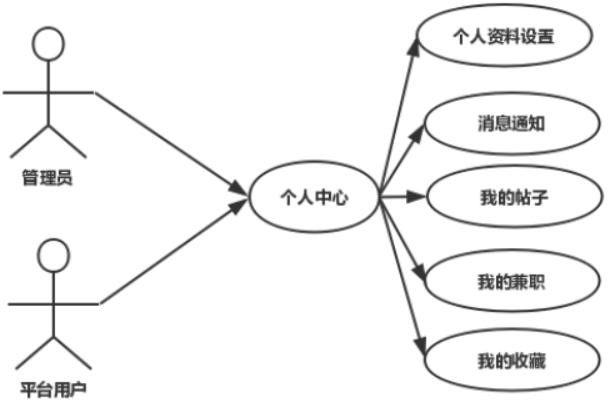
用例描述:

表 2-9 聊天发起用例表

|  |  |
| --- | --- |
| 用例名称 | 聊天发起 |
| 用例标识符 | YL8 |
| 执行者 | 管理员、平台用户 |
| 简要描述 | 平台用户在浏览兼职职位信息时，对有感兴趣的职位可点击相关头像与  商家进行直接的沟通 |
| 前置条件 | 平台用户使用自己账号密码登录的前台页面 |
| 基本事件流 | 1. 用户登陆到前台展示页面主页面。浏览兼职列表 2. 对自己感兴趣的职位可进行详细的了解 3. 有兼职的意愿的话可点击与商家交流 |
| 后置条件 | 发起成功 |

1. 个人中心

个人中心主要是对我们平台的用户在前台的个人中心页面可以进行操作的页面，个人中心包括个人资料设置、消息通知、我的帖子、我的兼职、我的收藏。个人中心用例图如图 2-11 所示：



用例描述:

图 2-11 个人中心用例图

表 2-10 个人资料设置用例表

|  |  |
| --- | --- |
| 用例名称 | 个人资料设置 |
| 用例标识符 | YL9 |
| 参与者 | 管理员、平台用户 |
| 简要描述 | 管理员或者平台用户在前台展示页面对自己的个人资料进行查看或者修  改保存 |
| 前置条件 | 管理员或者平台用户使用自己账号密码登录的前台页面，个人信息已经存  在平台数据中 |
| 基本事件流 | 1. 管理员或者平台用户在个人中心页面点击个人资料设置 2. 选择要修改的资料，编辑修改，点击保存 |
| 后置条件 | 显示修改成功 |

表 2-11 消息通知用例表

|  |  |
| --- | --- |
| 用例名称 | 消息通知 |
| 用例标识符 | YL10 |
| 参与者 | 管理员、平台用户 |
| 简要描述 | 管理员或者平台用户在前台展示页面对沟通消息的查看回复或者对自己所  发表的帖子其他用户的评论进行查看 |

续表 2-11

|  |  |
| --- | --- |
| 前置条件 | 管理员或者平台用户使用自己账号密码登录的前台页面，消息通知已经存  在平台数据中 |
| 基本事件流 | 1. 管理员或者平台用户在个人中心页面点击消息通知 2. 选择要回复的消息，编辑信息，点击发送 3. 点击评论查看他人的评论 |
| 后置条件 | 显示发送成功 |

表 2-12 我的帖子用例表

|  |  |
| --- | --- |
| 用例名称 | 我的帖子 |
| 用例标识符 | YL11 |
| 参与者 | 管理员、平台用户 |
| 简要描述 | 管理员或者平台用户在前台展示页面对沟通消息的查看自己所发表的论  坛帖子 |
| 前置条件 | 管理员或者平台用户使用自己账号密码登录的前台页面，我的帖子记录已  经存在平台数据中 |
| 基本事件流 | 1. 管理员或者平台用户在个人中心页面点击我的帖子 2. 点击查看我所发的帖子 |
| 后置条件 | 帖子成功展示 |

表 2-13 我的兼职用例表

|  |  |
| --- | --- |
| 用例名称 | 我的兼职 |
| 用例标识符 | YL12 |
| 参与者 | 管理员、平台用户 |
| 简要描述 | 管理员或者平台用户在前台展示页面对沟通消息的查看自己所发表的兼  职 |
| 前置条件 | 管理员或者平台用户使用自己账号密码登录的前台页面，我发表的兼职职  位记录已经存在平台数据中 |
| 基本事件流 | 1. 管理员或者平台用户在个人中心页面点击我的兼职 2. 点击查看我所发的兼职消息 |
| 后置条件 | 兼职信息成功展示 |

表 2-14 我的收藏用例表

|  |  |
| --- | --- |
| 用例名称 | 我的收藏 |

续表 2-14

|  |  |
| --- | --- |
| 用例标识符 | YL13 |
| 参与者 | 管理员、平台用户 |
| 简要描述 | 管理员或者平台用户在前台展示页面对沟通消息的查看自己所收藏的帖  子 |
| 前置条件 | 管理员或者平台用户使用自己账号密码登录的前台页面，我收藏的帖子记  录已经存在平台数据中 |
| 基本事件流 | 1. 管理员或者平台用户在个人中心页面点击我的收藏 2. 点击查看我收藏的帖子 |
| 后置条件 | 所收藏的帖子信息成功展示 |

### 非功能需求

* + 1. 环境需求

服务开发端:

表 2-15 服务器开发端环境

|  |  |
| --- | --- |
| 系统所使用数据库 | MySql 5.6 |
| 操作系统 | Windows XP |
| 服务器 | Apache Tomcat 8.5.11 |
| 数据库 | SQLServer 2005 |
| 集成开发环境 | IDEA 2019.1.1 |

客户端：

表 2-16 客户端环境

|  |  |
| --- | --- |
| 操作系统 | Windows 10 4G 以上 |
| 浏览器 | IE 谷歌 或者火狐浏览器 |
| 分辨率 | 最佳效果为 1920×1080 像素 |

* + 1. 性能需求

1. 可扩展性要求:主要考察的是我们的平台系统在应对在将来的各种变化时的调节能力，并在容量扩展、新业务扩展和系统接口连接等方面提供快速扩展。快速、方便、准确的应用。该系统还应提供与其他系统交换信息的接口，并提供技术支助服务。
2. 可管理性需求：平台系统要有一个合格的管理机制，以确保我们的系统在运行期间，平台的管理员可以便捷的来管理各种可能出现的状况。每个单元可以建立一个单独的文档流。
3. 容错性需求：在我们的故障发生时能够及时的给出判断，并且有相应的工具进行修复。以及对数据错误记录，一旦发生时能够预警，具有高容错能力以及发生错误时的能够做到一定的自主修复。
4. 易用性需求：平台系统应具有简洁无二义性的操作界面；系统必须能够针对不同的管理对象（类型）以分别执行授权。该平台系统的各操作简单直观，并且易于理解。
5. 可维护性需求:当我们的平台系统发生自身故障或者遭受到外界的攻击时,产生数据的丢失或者损坏时,可以及时的找回丢失的数据.对系统做到及时的维护
   * 1. 安全需求
6. 可靠性要求：我们的平台应能够一周每天 24 小时持续工作，当出现问题故障时，可以及时的做到故障问题的排除和我们数据的保存。当不可预测的错误发生系统在内部或外部时，可以及时的提供正确的紧急处理机制。
7. 安全要求：整个平台的建设要符合系统安全级别保护的三级请求。有一定信息安全防护措施，对我们的平台信息实现全面的保护。此外，应采用软件安全设置以防止合法用户意外破坏数据。
8. 系统运行安全：在面对非法入侵时，可以做出一定的及时补救措施。该系统平台能够做到定期的自动和手动备份数据，在我们的数据信息受到损坏时，能够对我们的数据进行一定的找回。

# 总体设计

### 系统设计的原则

我们要做的系统的好坏从系统设计的质量高低从根本上决定了。因此我们在软件开发之前，就要好好的考虑到在软件开发的过程中会出现的各种问题以及出现问题后相对应的解决办法。考虑到我们得系统的长远规划目标和系统的将来，我们将遵循以下设计原则：

1. 实用性原则：我们的大学生兼职平台应该以在校的学生适用为第一原则，满足我们的大学生的自己本身专业特性，合理的发布兼职信息，以我们的大学生为重点中心。不能一味的最求系统的先进性，因为科学技术的发展是日新月异的，但是实用性是要保持长期稳定不变的。
2. 先进性原则：要多考虑系统目前所所采用的技术或者框架的先进性。我们在开发的过程中，要尽量多多的学习和借鉴我们当前行业里的先进核心技术。进而来保证我们的系统平台在接下来的未来几年内，能够维持住他长期的核心竞争力，不至于因技术落后的原因带来各种潜在的威胁。
3. 规范性原则：在我们平台系统的开发的过程中，设计原则和规范一定要符合我们软件工程行业的特点和相关的标准，在合理规范的环境下开发，以保证我们的系统平台的秩序性。
4. 安全性原则：我们的系统应具备高效的防御机制，防止非法操作带来损害。系统为各个不同的角色用户提供严格权限资格审核机制，在数据库设计方面，我们的数据库资料要做到及时的备份恢复机制，关键的信息或者记录操作要做到提供日志记录。对各类角色用户的操作都有相应的记录。
5. 可扩展性原则：为了考虑在将来我们的平台系统会进行技术的更新，系统的绝大部分的模块都保留了相对应的扩展接口。因为在将来的兼职形式和模型将不会是静态的一成不变的，所以我必须为系统设计的未来开发做某些准备。在设计开始时，必须考虑未来可能存在的更改或者业务的扩展，稳定性是系统可持续发展的基础。

### 系统体系结构设计

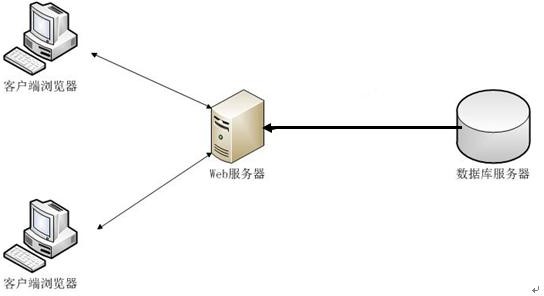


图 3-1 B/S 体系结构

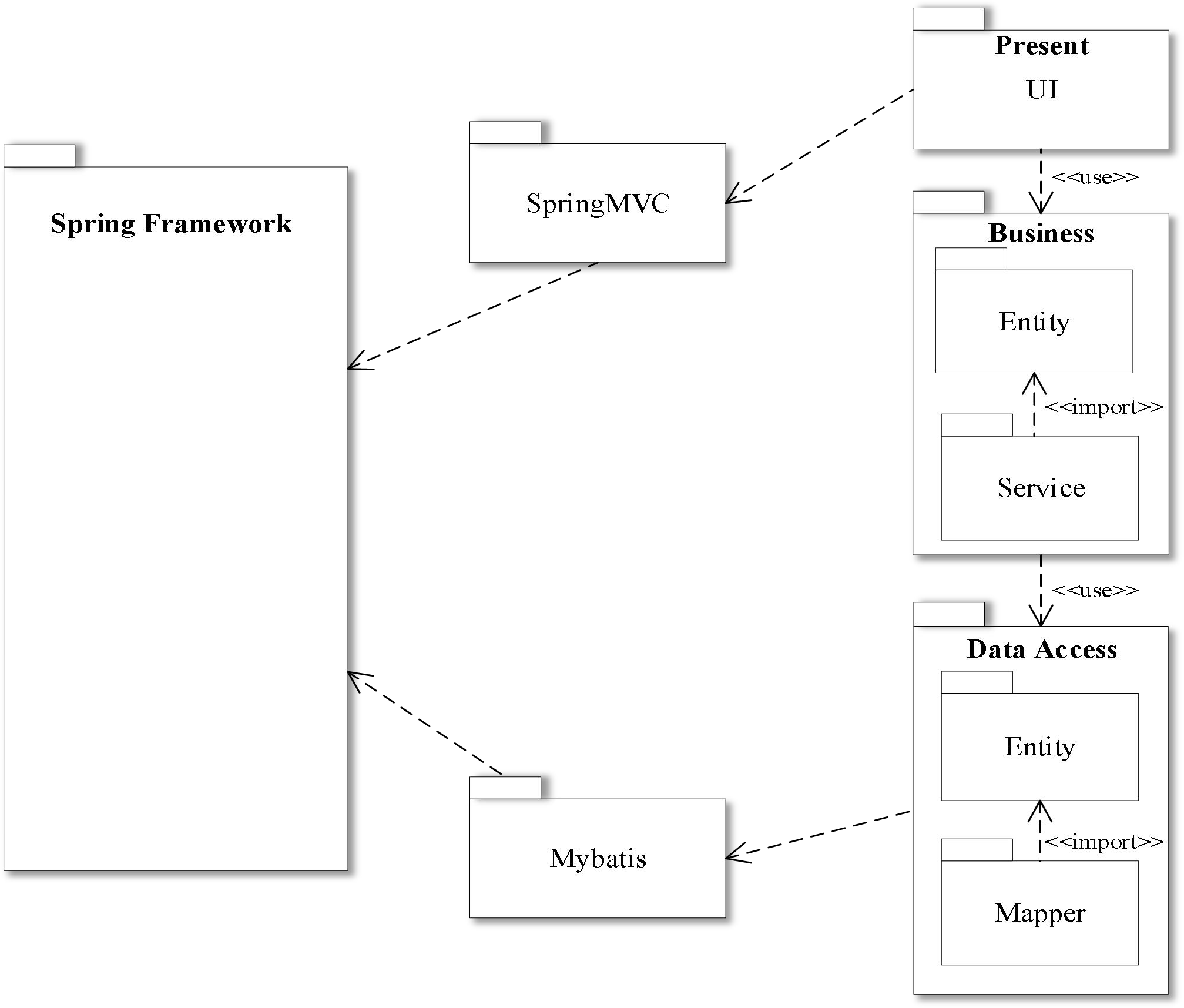
随着我们网络技术的发展，技术的不断成熟。在原来的 C/S 架构的基础之上，改进和发展到现在的 B/S 架构，即 Browser 和 Server。在 B/S 架构下，我们的用户的操作界面之类的，都是通过浏览器的图形化操作界面来实现的，小许的事务逻辑在我们的浏览器上实现。而其他的主要的还是在服务器段来实现。这种体系结构更加的方便简洁，成本也无需太高，仅仅要对我们的服务器进行网络管理。

图 3-2 系统结构图

如上图所示，根据 MVC 设计原理，该系统有以下几方面：表现层、业务层、数据层。表现层主要是系统平台界面，并依赖 SpringMVC 作为控制器层。业务层主要是实体类和

服务层类，它们取决于表现层。显示数据；数据层主要包括实体类和数据库操作类，这取决于 Mybatis 数据库操作框架，三层体系结构建立在 Spring 框架上。

本系统的服务器开发采用 SSM 框架设计。作为企业应用开发框架，该系统在应用服务和数据处理方面有较好的性能，但在服务器启动过程中需要建立 webSocket 服务,利用websocket 实现我们平台内的用户在网页界面上进行聊天交流。

### 系统功能结构设计

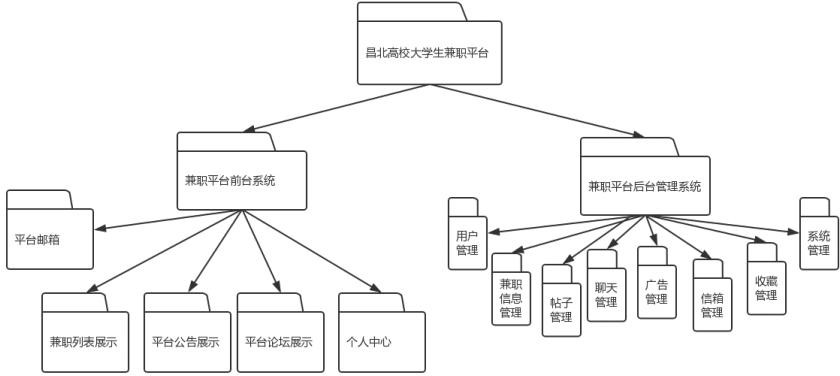


图 3-3 系统总包图

我们的兼职平台系统一共分为两个子系统，兼职平台前台页面展示子系统和兼职平台后台管理子系统。其中，兼职平台前台子系统又分为平台邮箱、兼职列表展示、平台公告展示、平台论坛展示、个人中心共 5 个页面模块。兼职平台后台管理子系统又分为用户管理、兼职信息管理、帖子管理、聊天管理，广告管理、信箱管理、收藏管理、系统管理 8 个模块。

1. 用户管理

用户管理主要分为管理员在我们的后台管理系统中可手动添加新建平台用户、用户查询、用户信息的修改、用户的禁用或者删除，同时我们的游客也可在前台登录页面进行注册。

1. 兼职职位管理

兼职职位管理主要是对我们平台有兼职职位发布的商家所发布的职位信息进行管 理，包括兼职职位的添加、兼职职位的查询、兼职职位的编辑修改和兼职职位的禁用删除。

1. 论坛帖子管理

论坛帖子管理主要包括两大部分，用户的发帖管理和评论管理。包括论坛所发贴的增加、内容的查看、内容的修改、帖子的删除和帖子的收藏；评论管理包括评论的添加、评论的查询、论坛发帖的评论的编辑和帖子评论的删除。

1. 广告投放管理

广告管理主要是对我们平台有广告发布合作的商家所发布的广告信息进行管理，包

括广告的添加、广告的查询、广告的编辑修改和广告的禁用删除。(5)系统管理

平台管理员在后台系统可对平台进行系统管理，主要分为三大模块，分别是轮播列表管理、公告列表管理和系统设置。轮播列表管理包括轮播添加、轮播查询、轮播修改和轮播删除；公告列表管理包括公告添加、公告查询、公告修改和公告删除；系统设置包括平台名称、平台地址、欢迎语、联系方式设置。

(6)聊天管理

聊天管理主要是对我们平台的用户和商家就兼职职位相关问题的沟通聊天信息进 行管理，包括聊天发起、聊天记录的查询、聊天记录的编辑修改和聊天记录的禁用删除。(7)个人中心

个人中心主要是对我们平台的用户在前台的个人中心页面可以进行操作的页面，个人中心包括个人资料设置、消息通知、我的帖子、我的兼职、我的收藏。

# 数据库设计

### 概念结构设计

#### 设计思路

根据对我们平台系统的各个功能的设计，大概有这几个实体:管理员、平台用户、广告，聊天、收藏、帖子、信箱、公告、兼职工作这几个表。

他们的对应关系:

1. 管理员与平台用户的关系为多对多
2. 管理员与广告关系为一对多
3. 管理员与公告关系为一对多
4. 管理员与聊天告关系为一对多
5. 管理员与收藏关系为一对多
6. 管理员与帖子关系为一对多
7. 管理员与邮箱关系为一对多
8. 管理员与兼职工作关系为一对多
9. 平台用户与聊天的关系为一对多
10. 平台用户与邮箱的关系为一对一
11. 平台用户与兼职工作的关系为一对多
12. 平台用户与收藏的关系为一对多
13. 平台用户与帖子的关系为一对多

具体的业务内容大概如下：管理员管理平台用户、平台广告、平台公告、平台用户聊天、平台用户收藏、平台帖子、平台用户的邮件、平台兼职信息、平台用户与商家用户聊天、平台用户发送建议邮箱给平台、平台用户浏览兼职工作信息、平台用户浏览论坛帖子、平台用户收藏感兴趣帖子、平台用户评论帖子。

#### E-R 图

图 4-1 实体关系 E-R 图

各实体类设计如下所示:

1. 广告实体图,如图 4-2 所示. 实体:广告

属性:id、发布人、标题、内容、图片、时间。

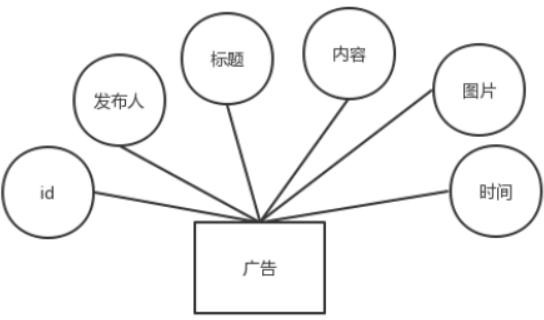


图 4-2 广告实体 E-R 图

1. 聊天实体图,如图 4-3 所示. 实体:聊天

属性:id、发送方、接收方、聊天内容、图片、信息状态。



图 4-3 聊天实体 E-R 图

1. 收藏实体图,如图 4-4 所示. 实体:收藏

属性:id、收藏人、帖子、时间。

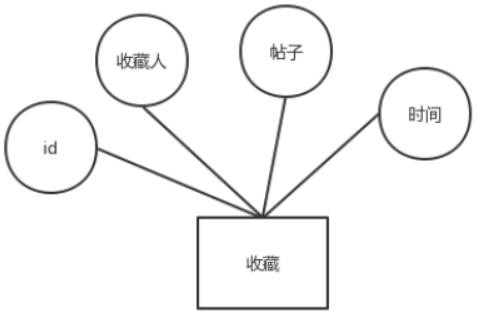


图 4-4 收藏实体 E-R 图

1. 帖子实体图,如图 4-5 所示. 实体:帖子

属性:帖子 id、发帖人、内容、图片、时间。



图 4-5 帖子实体 E-R 图

1. 信箱实体图,如图 4-6 所示. 实体:信箱

属性:id、发送人、标题、内容、图片、时间。

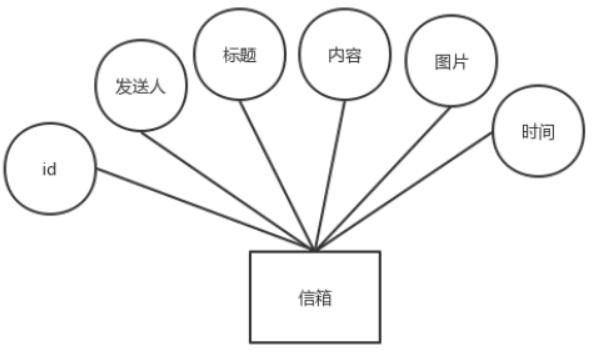


图 4-6 信箱实体 E-R 图

1. 广告实体图,如图 4-7 所示. 实体:公告

属性:id、发布人、标题、内容、图片、时间。

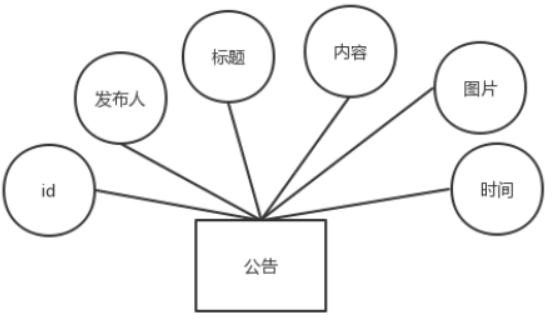


图 4-7 公告实体 E-R 图

1. 用户或者管理员实体图,如图 4-8 所示. 实体:用户或者管理员

属性:id、姓名、昵称、密码、角色种类、性别、电话、生日、邮箱、头像、密保问题、密保答案、时间。

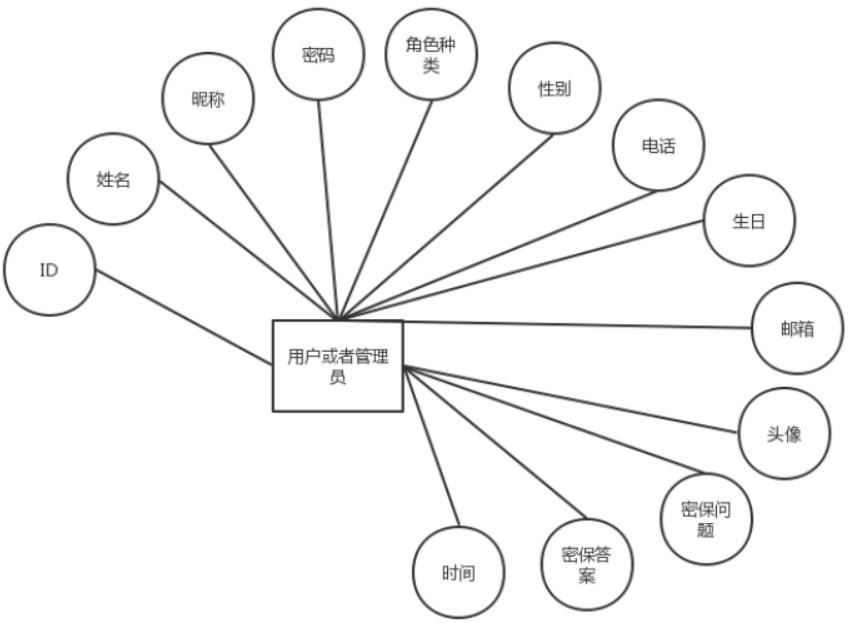


图 4-8 用户或者管理员实体 E-R 图

### 逻辑结构设计

#### 设计思路

在我们的数据库设计中，用每一个实体的 id 设为主键，当某一个的表与其他的表有关联时，及时的引入外键（索引），最后对我们的数据模型进行优化。

#### 逻辑模型

表 4-1 用户基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 主键 |
| user\_id | int | 11 | 0 | 1 |
| real\_name | varchar | 255 | 0 | -1 |
| nick\_name | varchar | 255 | 0 | -1 |
| password | varchar | 255 | 0 | -1 |
| user\_leven | int | 11 | 0 | -1 |
| user\_category | varchar | 255 | 0 | -1 |
| sex | int | 11 | 0 | -1 |
| phone | varchar | 255 | 0 | 0 |
| pirthday | datetime | 0 | 0 | -1 |
| email | varchar | 255 | 0 | -1 |
| header\_image | varchar | 255 | 0 | -1 |
| question | varchar | 255 | 0 | -1 |
| answer | varchar | 255 | 0 | -1 |
| create\_time | datetime | 0 | 0 | -1 |
| is\_effect | int | 11 | 0 | -1 |
| is\_work | int | 11 | 0 | -1 |
| remark | varchar | 255 | 0 | -1 |

表 4-2 兼职职位基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 主键 |
| id | int | 11 | 0 | 0 |
| user\_id | int | 11 | 0 | 0 |
| work\_category | varchar | 255 | 0 | -1 |
| phone | varchar | 255 | 0 | -1 |

续表 4-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| content | varchar | 500 | 0 | -1 |
| image | varchar | 255 | 0 | -1 |
| title | varchar | 255 | 0 | -1 |
| is\_auditing | int | 11 | 0 | 0 |
| is\_effect | int | 11 | 0 | 0 |
| create\_time | datetime | 0 | 0 | -1 |

表 4-3 公告基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 主键 |
| id | int | 11 | 0 | 0 |
| user\_id | int | 11 | 0 | 0 |
| image | varchar | 255 | 0 | -1 |
| title | varchar | 255 | 0 | -1 |
| content | varchar | 500 | 0 | -1 |
| create\_time | datetime | 0 | 0 | -1 |

表 4-4 信箱基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 主键 |
| id | int | 11 | 0 | 0 |
| user\_id | int | 11 | 0 | 0 |
| image | varchar | 255 | 0 | -1 |
| title | varchar | 255 | 0 | -1 |
| content | varchar | 500 | 0 | -1 |
| create\_time | datetime | 0 | 0 | -1 |

表 4-5 帖子基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 主键 |
| forum\_id | int | 11 | 0 | 0 |
| user\_id | int | 11 | 0 | 0 |
| image | varchar | 255 | 0 | -1 |
| title | varchar | 255 | 0 | -1 |

续表 4-5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| content | varchar | 500 | 0 | -1 |
| is\_effect | int | 11 | 0 | -1 |
| create\_time | datetime | 0 | 0 | -1 |

表 4-6 收藏基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 主键 |
| id | int | 11 | 0 | 0 |
| user\_id | int | 11 | 0 | 0 |
| forum\_id | int | 11 | 0 | 0 |
| create\_time | datetime | 0 | 0 | -1 |

表 4-7 聊天基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 主键 |
| id | int | 11 | 0 | 0 |
| user\_id\_fa | int | 11 | 0 | 0 |
| user\_id\_jie | int | 11 | 0 | 0 |
| image | varchar | 255 | 0 | -1 |
| title | varchar | 255 | 0 | -1 |
| content | varchar | 500 | 0 | -1 |
| create\_time | datetime | 0 | 0 | -1 |
| is\_look | int | 1 | 0 | 0 |
| is\_remove\_fa | int | 1 | 0 | 0 |
| is\_remove\_jie | int | 1 | 0 | 0 |
| chat\_signal | varchar | 255 | 0 | -1 |

表 4-8 广告基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名 | 类型 | 长度 | 小数点 | 主键 |
| id | int | 11 | 0 | 0 |
| user\_id | int | 11 | 0 | 0 |
| image | varchar | 255 | 0 | -1 |
| title | varchar | 255 | 0 | -1 |
| content | varchar | 500 | 0 | -1 |
| position | int | 11 | 0 | -1 |
| create\_time | datetime | 0 | 0 | -1 |

### 物理结构设计

#### 存取方式

一般通过设置索引，在当需要用到表中的某些字段时，将其设为索引，通过索引从数据库中取出数据，此平台系统采用的是 MySql 数据库进行存储。。

#### 存储结构

存储系统主要包括三个部分：内存中：buffer，MemoryContext;数据文件，临时文件; 日志文件，日志缓存。

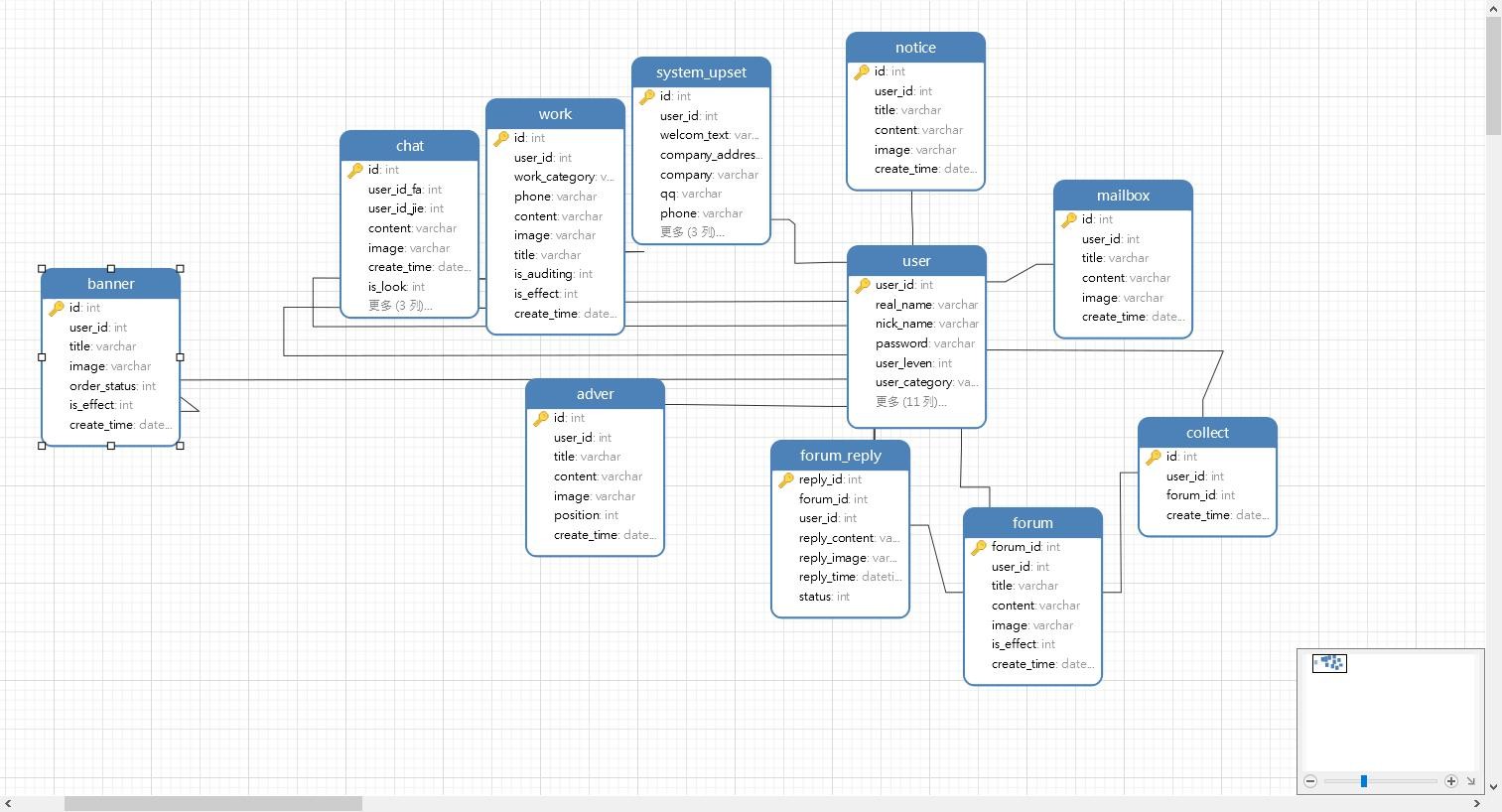
物理表设计如图 4-9 所示：

图 4-9 物理表设计图

# 界面设计

### 界面关系图

* + 1. 用户端界面关系图

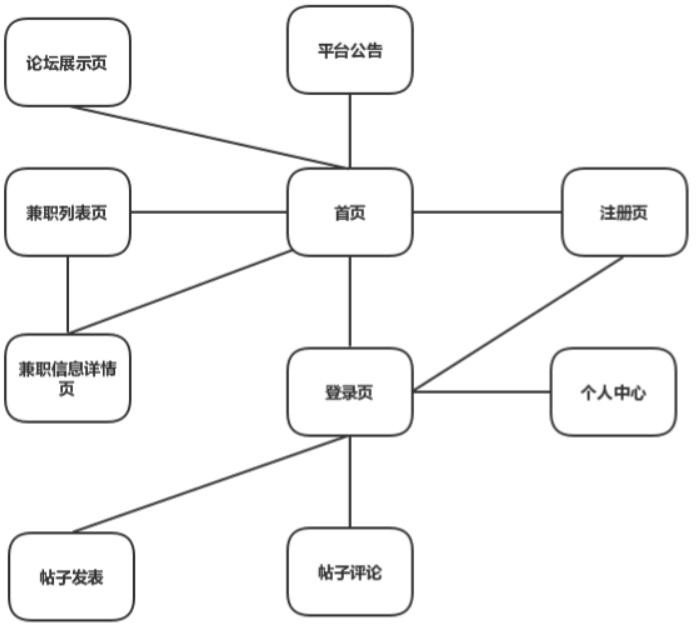
用户端的界面主要有首页、兼职信息展示、兼职信息分类、论坛交流、个人中心、平台公告、沟通聊天界面。用户端界面关系图如图 5-1 所示：

图 5-1 用户端界面关系图

* + 1. 后台管理界面关系图

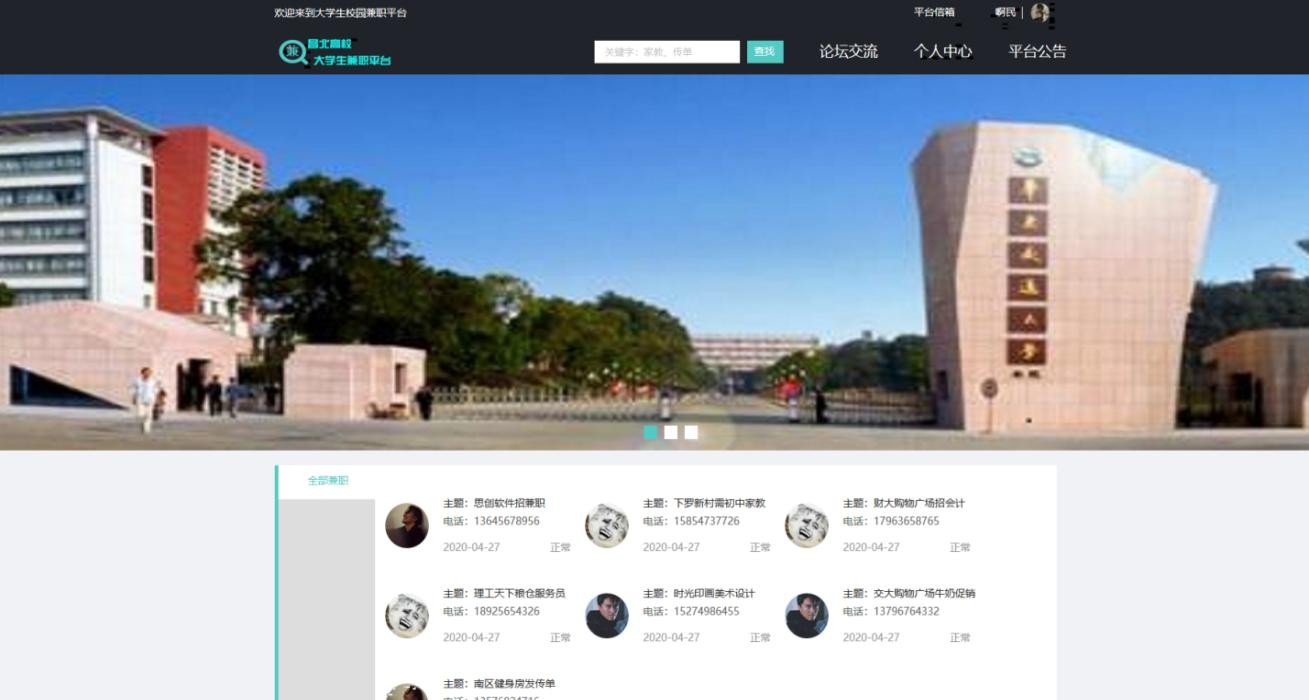
后台管理界面关系图如图 5-2 所示：



图 5-2 后台管理界面关系图

### 界面设计成果

* + 1. 主界面



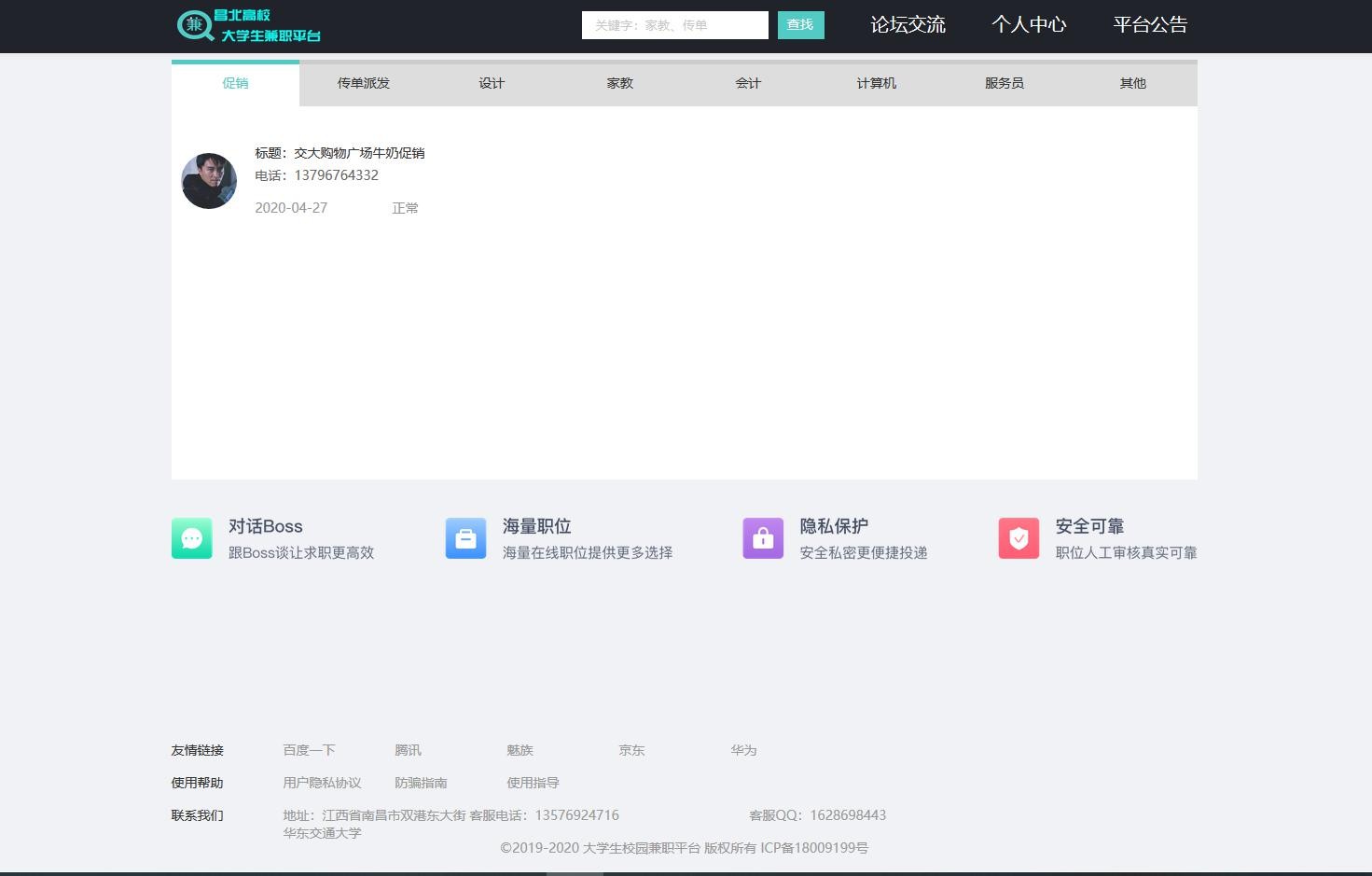


图 5-3 用户端主界面效果图

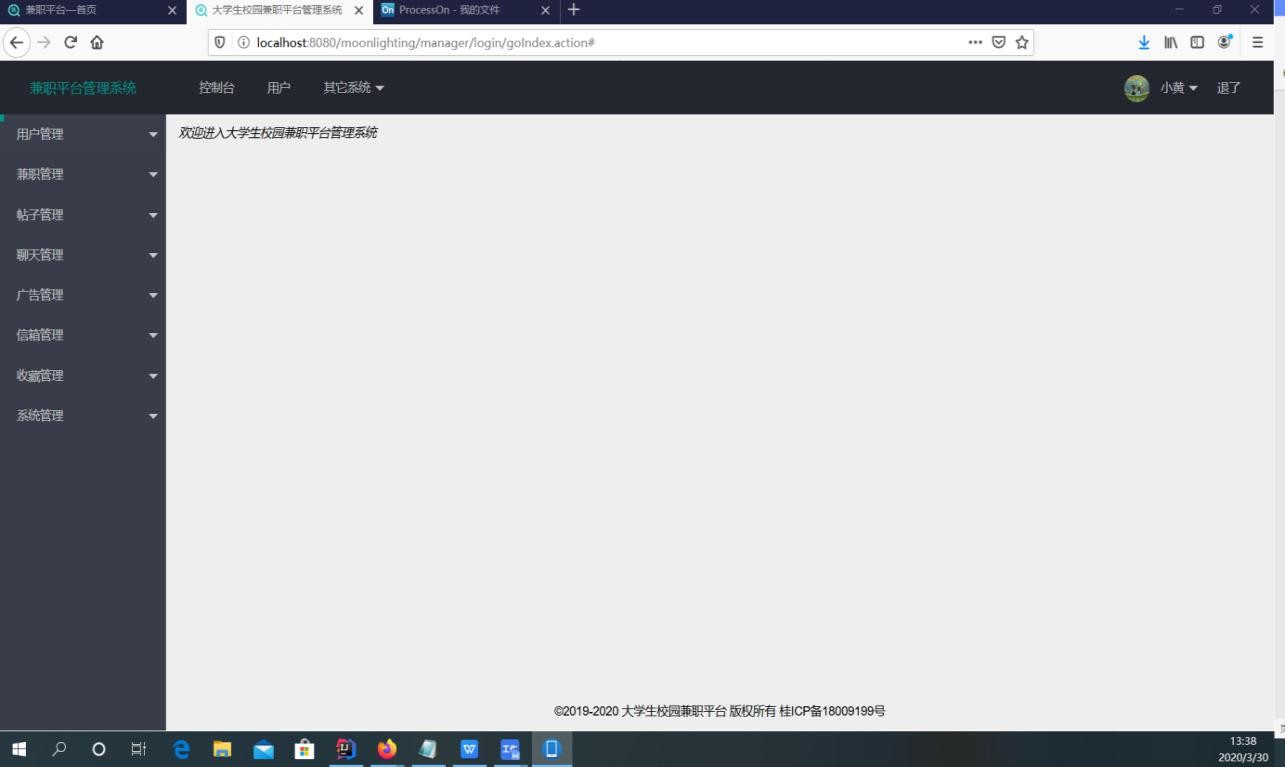


图 5-4 后台管理主界面效果图

* + 1. 子界面



图 5-5 兼职信息展示页面效果图



图 5-6 论坛展示页面效果图



图 5-7 个人中心页面效果图



图 5-8 平台公告页面效果图



图 5-9 平台信箱页面效果图

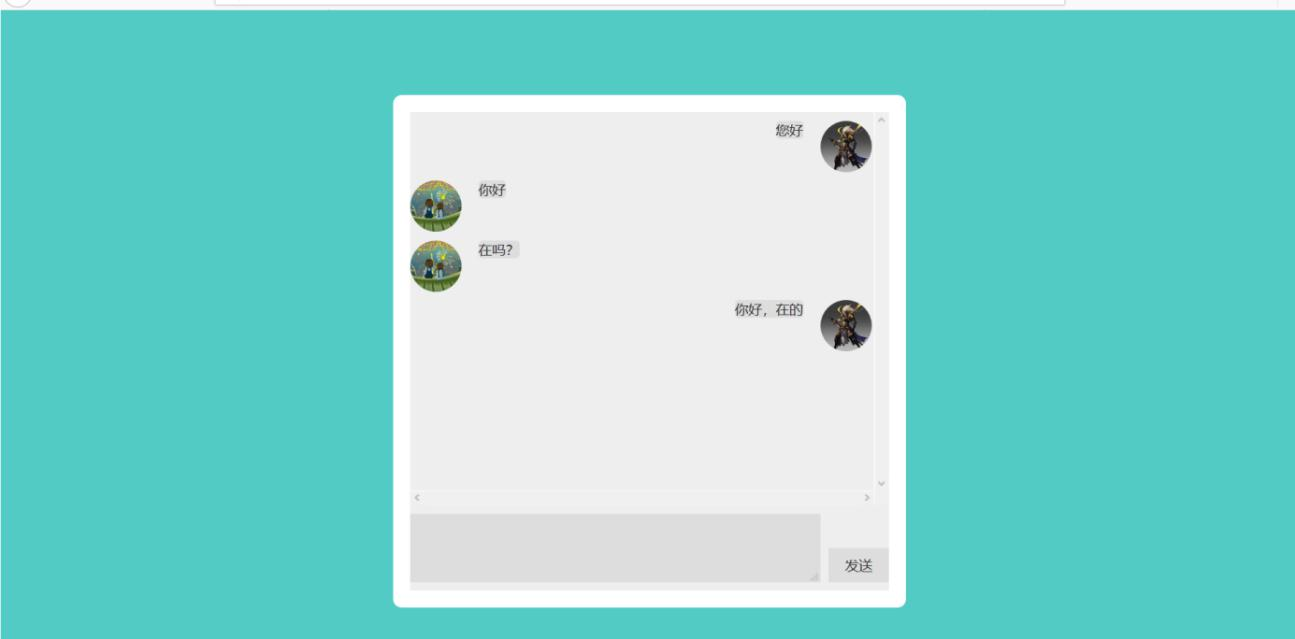


图 5-10 聊天沟通页面效果图

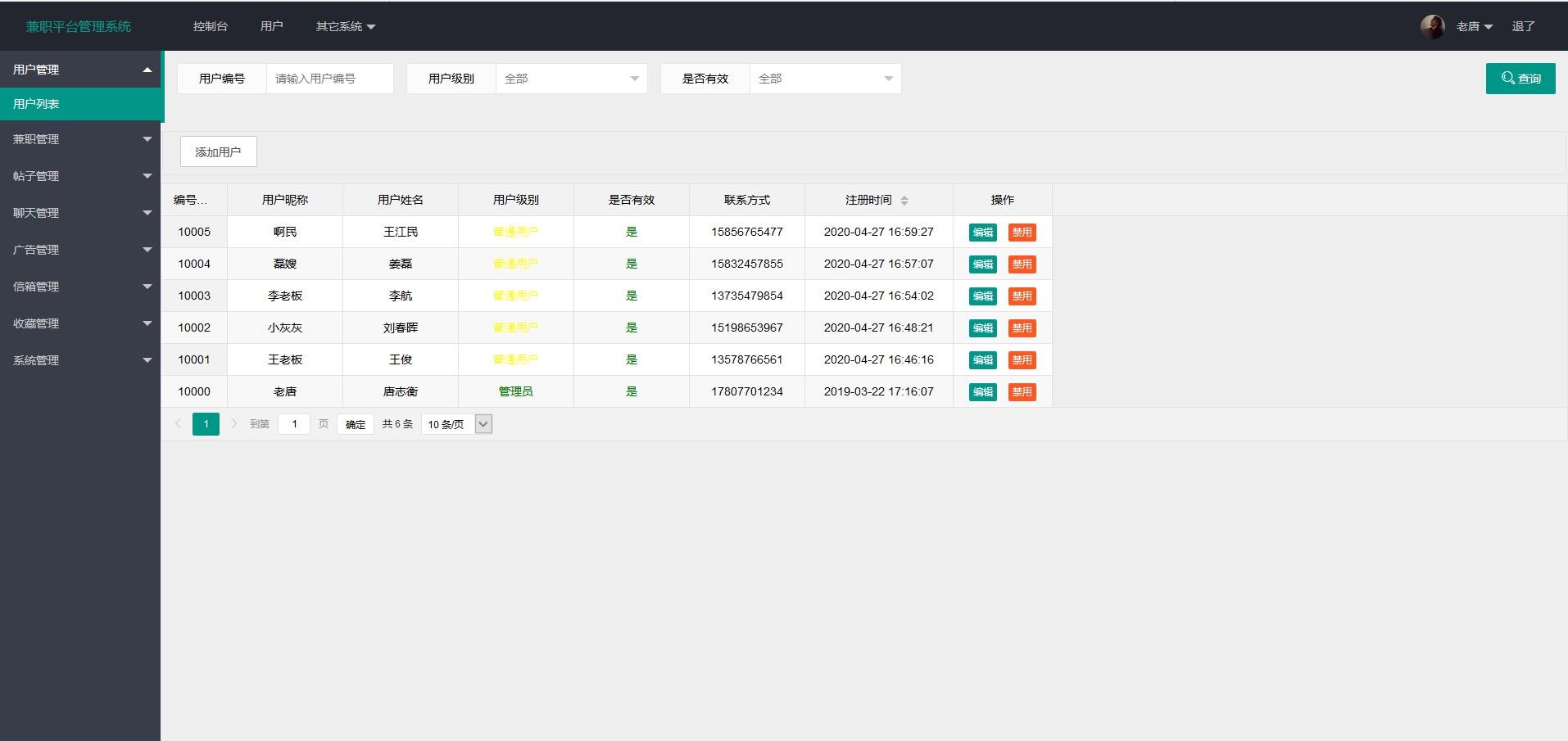


图 5-11 后台用户管理页面效果图

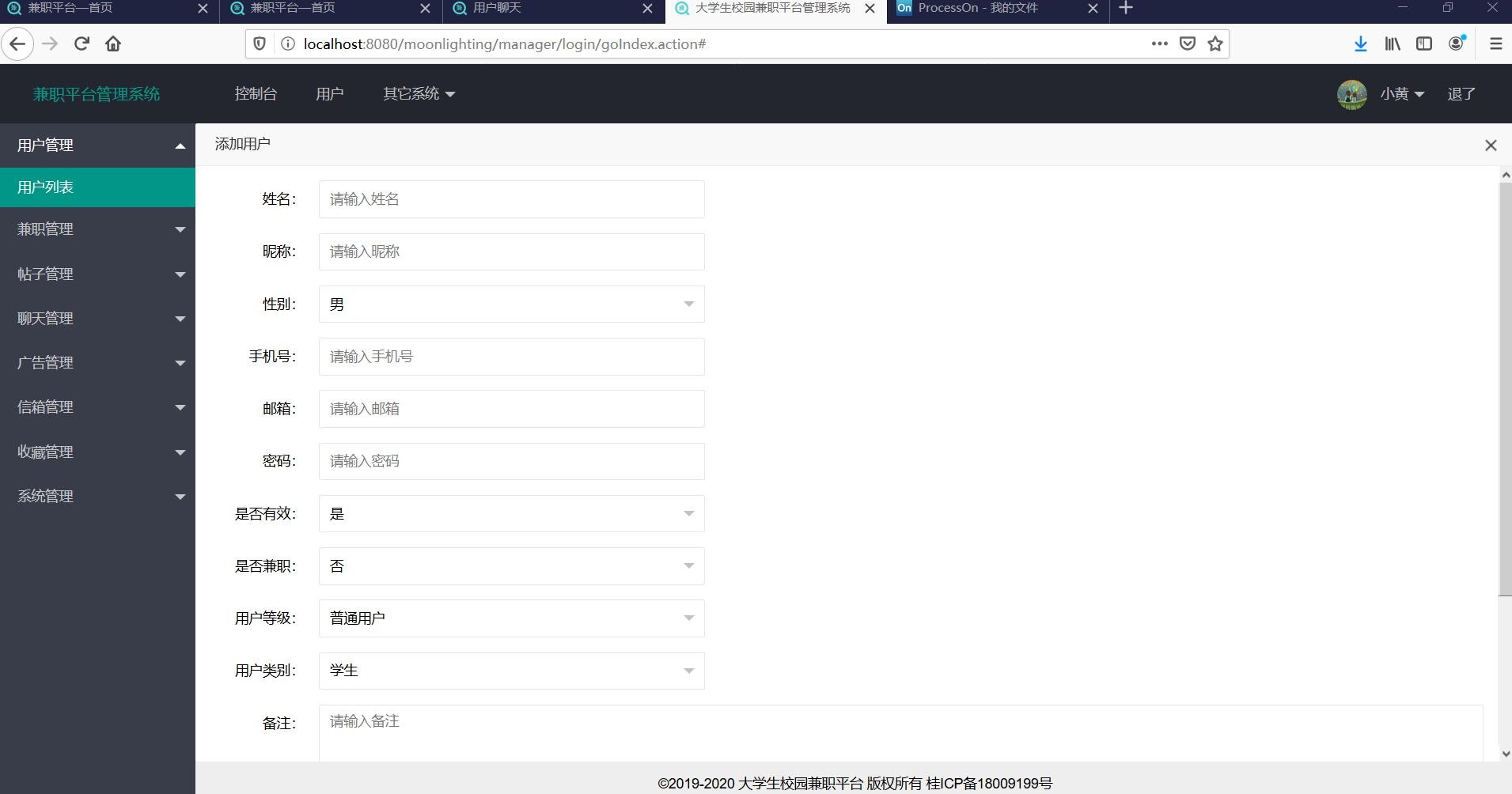


图 5-12 后台用户新增管理页面效果图

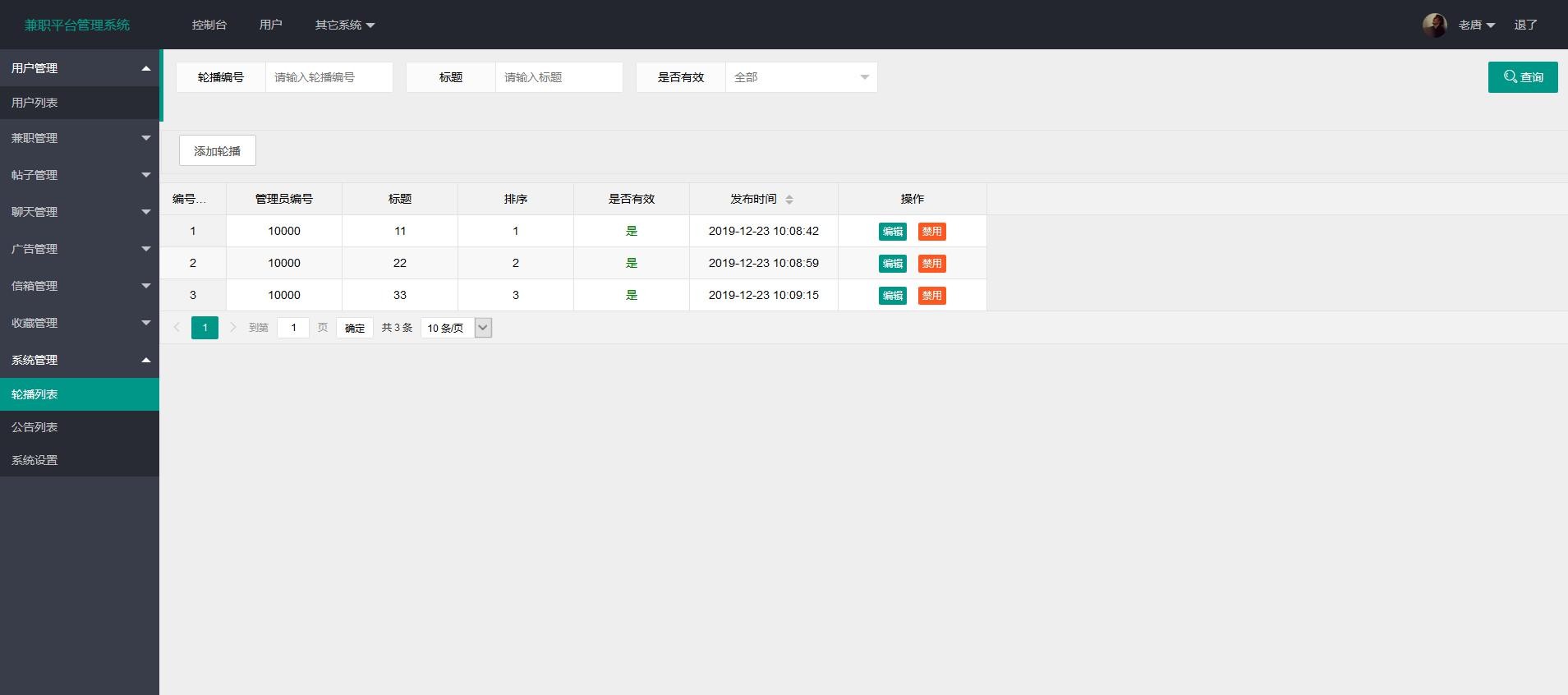


图 5-13 后台系统管理页面效果图

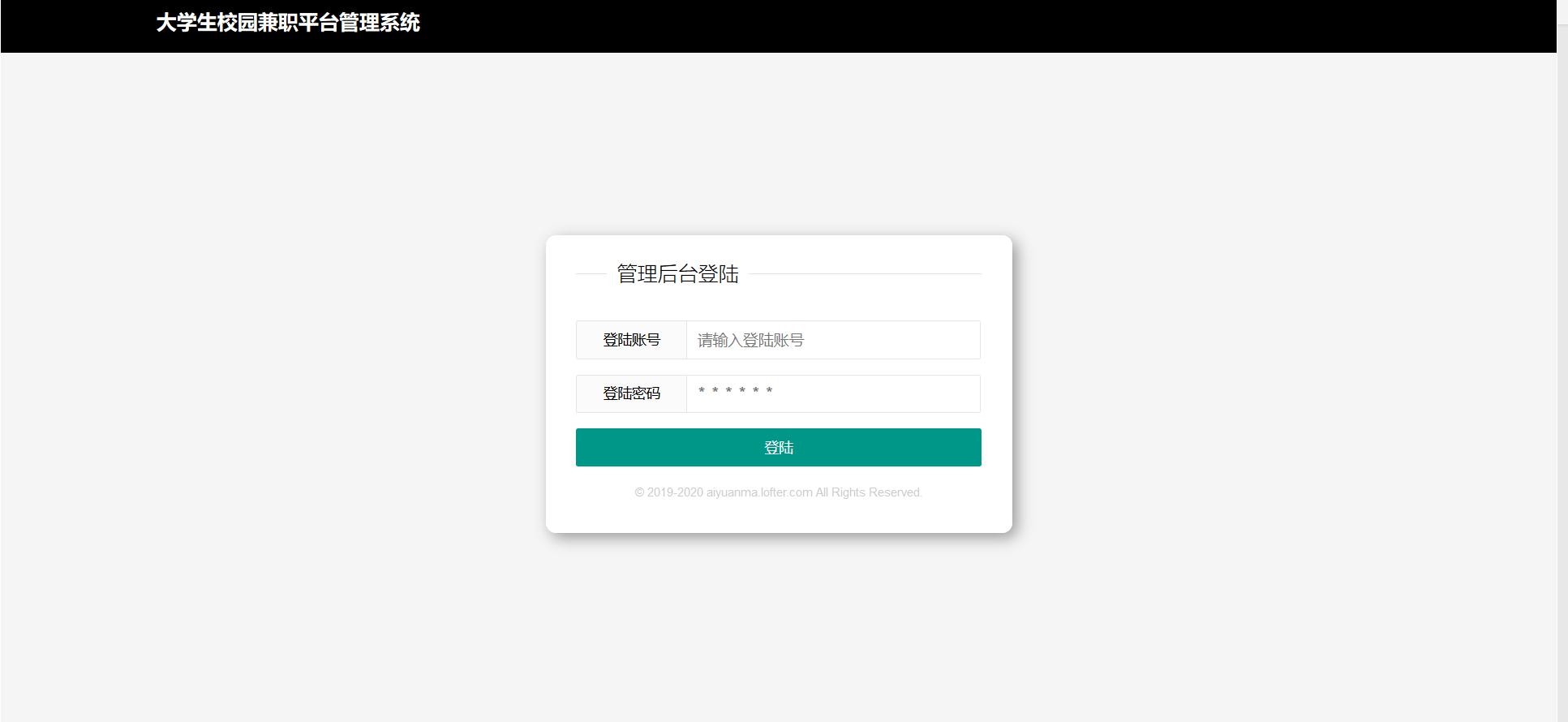


图 5-14 后台管理页登录页面效果图

# 详细设计

### 系统主要功能模块介绍

1. 用户管理模块

用户管理主要分为管理员在我们的后台管理系统中可手动添加新建平台用户、用户查询、用户信息的修改、用户的禁用或者删除，同时我们的游客也可在前台登录页面进行注册。

1. 兼职职位管理模块

兼职职位管理主要是对我们平台有兼职职位发布的商家所发布的职位信息进行管 理，包括兼职职位的添加、兼职职位的查询、兼职职位的编辑修改和兼职职位的禁用删除。

1. 论坛帖子管理模块

论坛帖子管理主要包括两大部分，用户的发帖管理和评论管理。包括论坛所发贴的增加、内容的查看、内容的修改、帖子的删除和帖子的收藏；评论管理包括评论的添加、评论的查询、论坛发帖的评论的编辑和帖子评论的删除。

1. 广告管理模块

广告管理主要是对我们平台有广告发布合作的商家所发布的广告信息进行管理，包括广告的添加、广告的查询、广告的编辑修改和广告的禁用删除。

1. 系统管理模块

平台管理员在后台系统可对平台进行系统管理，主要分为三大模块，分别是轮播列表管理、公告列表管理和系统设置。轮播列表管理包括轮播添加、轮播查询、轮播修改和轮播删除；公告列表管理包括公告添加、公告查询、公告修改和公告删除；系统设置包括在我们的前台对平台名称、平台地址、欢迎语、联系方式展示设置。

1. 聊天管理模块

聊天管理主要是对我们平台的用户和商家就兼职职位相关问题的沟通聊天信息进行管理，包括聊天发起，和在后台管理员对我们聊天记录的查询、聊天记录的编辑修改和聊天记录的禁用删除。

1. 收藏管理模块

我们的平台用户在前台页面浏览我们的平台论坛，在遇到自己感兴趣的帖子时，可进行收藏操作。同时我们的平台管理员，可在我们的后台管理页面中，对我们的收藏记录的查看，也可对记录的编辑修改和禁用删除。

### 用户管理功能模块设计

* + 1. 功能结构设计

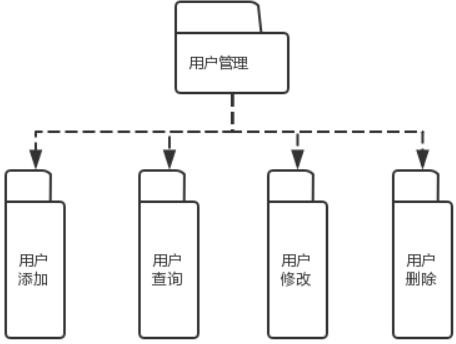


图 6-1 用户管理包图

* + 1. 类图设计

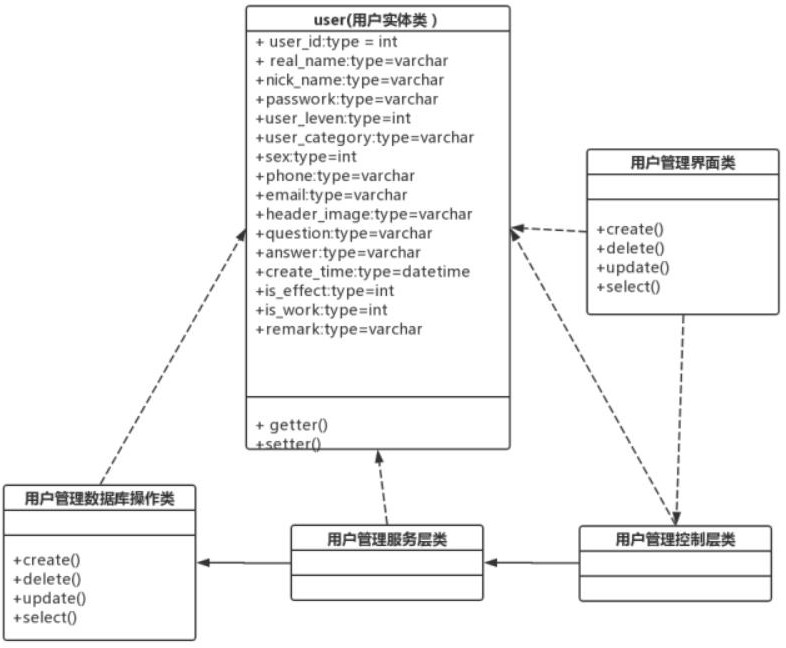


图 6-2 用户管理功能类图

表 6-1 用户管理界面类表

|  |  |  |
| --- | --- | --- |
| 用户管理界面类 | | |
| 类的描述：对我们用户的基本信息在前后端界面的管理操作 | | |
| 操作描  述 | create() | 在前台或者后台添加用户，填写用户基本信息 |
| delete() | 删除禁用用户信息 |
| update() | 不同身份在前后台修改用户各项基本信息。 |
| select() | 按一定的要求查询相关用户信息。 |

表 6-2 用户控制器类表

|  |  |  |
| --- | --- | --- |
| 用户管理控制器类 | | |
| 类的描述：封装用户管理的各项服务层操作方法 | | |
| 操作描  述 | create() | 在前台或者后台添加用户，填写用户基本信息 |
| delete() | 删除禁用用户信息 |
| update() | 不同身份在前后台修改用户各项基本信息。 |
| select() | 按一定的要求查询相关用户信息。 |

表 6-3 用户服务层类表

|  |  |  |
| --- | --- | --- |
| 用户管理服务层类 | | |
| 类的描述：封装对用户管理操作的服务层方法 | | |
| 操作描  述 | create() | 在前台或者后台添加用户，填写用户基本信息 |
| delete() | 删除禁用用户信息 |
| update() | 不同身份在前后台修改用户各项基本信息。 |
| select() | 按一定的要求查询相关用户信息。 |

表 6-4 用户数据访问层类表

|  |  |  |
| --- | --- | --- |
| 用户数据库操作类 | | |
| 类的描述：对数据库中的用户信息数据进行管理 | | |
| 操作描述 | create() | 在前台或者后台添加用户，填写用户基本信息 |
| delete() | 删除禁用用户信息 |
| update() | 不同身份在前后台修改用户各项基本信息。 |
| select() | 按一定的要求查询相关用户信息。 |

表 6-5 用户实体类表

|  |  |  |
| --- | --- | --- |
| 用户实体类 | | |
| 类的描述：用户信息实体对象类 | | |
| 属性描述 | user\_id | 用户 id |
| real\_name | 用户真实姓名 |
| nick\_name | 用户昵称 |
| password | 密码 |
| user\_leven | 用户等级 |
| user\_category | 用户类别 |
| sex | 用户性别 |
| phone | 用户电话 |
| pirthday | 用户生日 |
| email | 用户邮箱号码 |
| header\_image | 用户头像 |
| question | 密保问题 |
| answer | 密保答案 |
| create\_time | 创建时间 |
| is\_effect | 是否有效 |
| is\_work | 是否可以发布 |
| remark | 备注 |
| 操作描  述 | Setter（） Getter（） | 各实体属性的 set 与 get 方法 |

* + 1. 顺序图设计

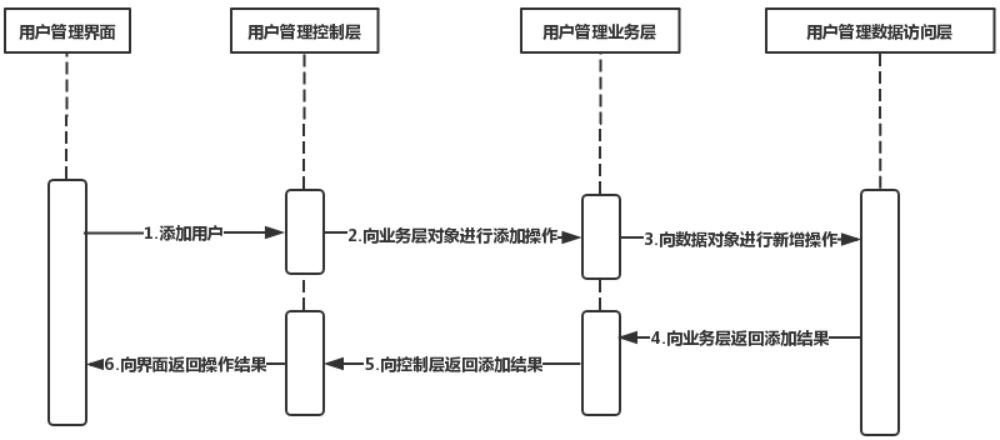


图 6-3 添加用户时序图

* + 1. 核心处理流程设计

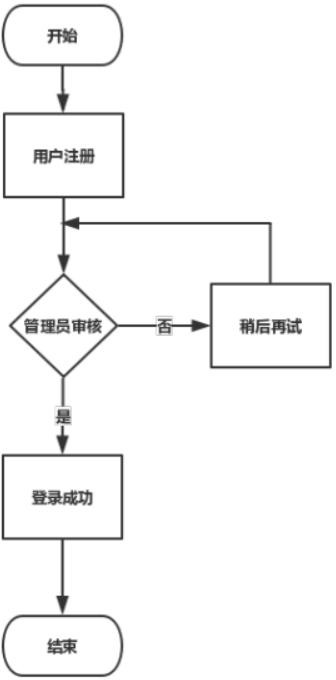


图 6-4 用户登录流程

### 兼职职位管理功能模块设计

* + 1. 功能结构设计

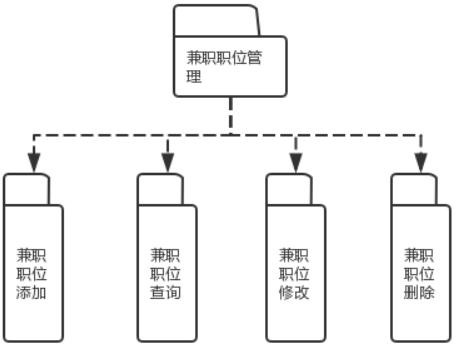


图 6-5 兼职职位管理包图

* + 1. 类图设计

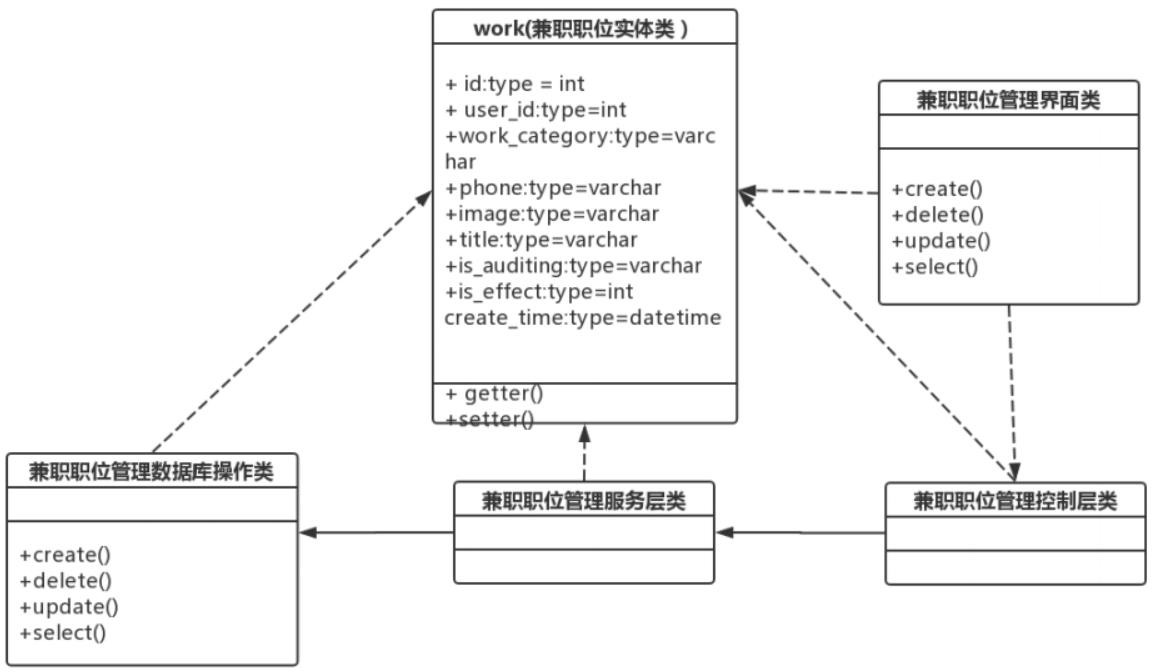


图 6-6 兼职职位管理功能类图

表 6-6 兼职职位管理面类表

|  |  |  |
| --- | --- | --- |
| 兼职职位管理界面类 | | |
| 类的描述：对我们的兼职职位管理在前后端进行图形化界面操作 | | |
| 操作描  述 | create() | 在前台或者后台添加兼职职位，填写职位基本情况 |
| delete() | 删除禁用无效的兼职职位信息记录 |
| update() | 不同身份在前后台修改兼职职位各项基本情况。 |
| select() | 按一定的要求条件查询相关兼职职位信息。 |

表 6-7 兼职职位控制器类表

|  |  |  |
| --- | --- | --- |
| 兼职职位管理控制器类 | | |
| 类的描述：封装兼职职位管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加兼职职位，填写职位基本情况 |
| delete() | 删除禁用无效的兼职职位信息记录 |
| update() | 不同身份在前后台修改兼职职位各项基本情况。 |
| select() | 按一定的要求条件查询相关兼职职位信息。 |

表 6-8 信息推送服务层类表

|  |  |  |
| --- | --- | --- |
| 兼职职位管理服务层类 | | |
| 类的描述：封装兼职职位管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加兼职职位，填写职位基本情况 |
| delete() | 删除禁用无效的兼职职位信息记录 |
| update() | 不同身份在前后台修改兼职职位各项基本情况。 |
| select() | 按一定的要求条件查询相关兼职职位信息。 |

表 6-9 兼职职位管理数据访问层类表

|  |  |  |
| --- | --- | --- |
| 兼职职位管理数据库操作类 | | |
| 类的描述：管理兼职职位管理数据库中的数据 | | |
| 操作描  述 | create() | 添加兼职职位，填写职位基本情况 |
| delete() | 删除禁用无效的兼职职位信息记录 |
| update() | 不同身份在前后台修改兼职职位各项基本情况。 |
| select() | 按一定的要求条件查询相关兼职职位信息。 |

表 6-10 兼职职位管理实体类表

|  |  |  |
| --- | --- | --- |
| 兼职职位实体类 | | |
| 类的描述：兼职职位实体对象类 | | |
| 属性描述 | id | 兼职职位 id |
| user\_id | 发布者编号 |
| work\_category | 兼职类别 |
| phone | 发布者电话 |
| content | 兼职内容 |
| image | 兼职相关图片 |
| title | 兼职信息标题 |
| is\_auditing | 是否通过管理员审核 |
| is\_effect | 信息是否有效 |
| create\_time | 发布时间 |
| 操作描  述 | Setter（） Getter（） | 对属性的 set 与 get 方法 |

* + 1. 顺序图设计

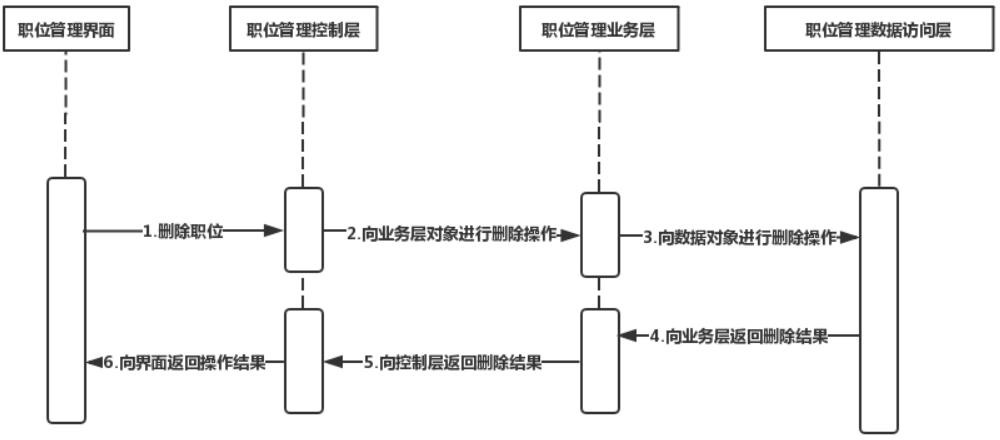


图 6-7 删除兼职职位时序图

* + 1. 核心处理流程设计

兼职职位管理功能核心处理流程图如下图所示：



图 6-8 兼职职位管理流程图

### 论坛帖子管理模块

* + 1. 功能结构设计

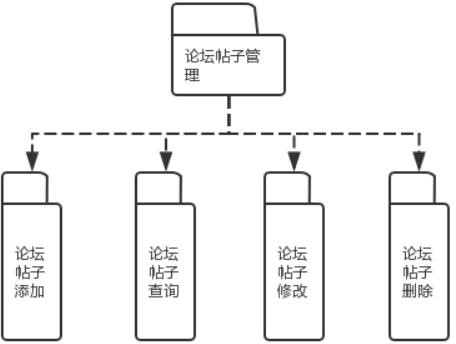


图 6-9 论坛帖子管理包图

* + 1. 类图设计

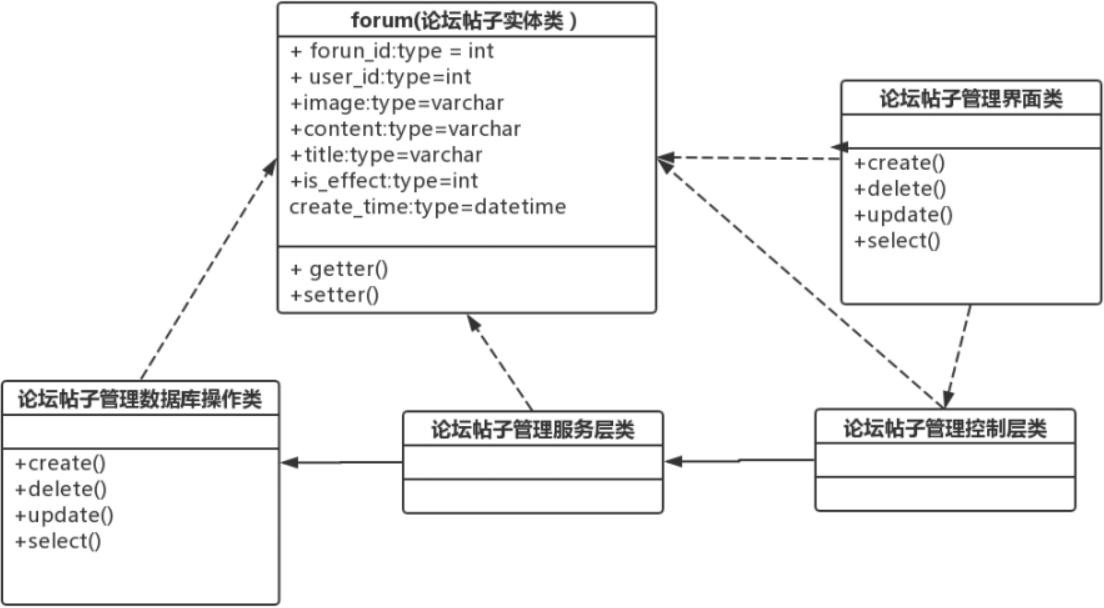


图 6-10 论坛帖子管理类图

表 6-11 论坛帖子管理面类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子管理界面类 | | |
| 类的描述：对我们的论坛帖子管理在前后端进行图形化界面操作 | | |
| 操作描  述 | create() | 在前台或者后台添加论坛帖子，填写帖子基本情况 |
| delete() | 删除禁用无效的论坛帖子信息记录 |
| update() | 不同身份在前后台修改论坛帖子各项基本情况。 |
| select() | 按一定的要求条件查询相关论坛帖子信息。 |

表 6-12 论坛帖子控制器类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子管理控制器类 | | |
| 类的描述：封装论坛帖子管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加论坛帖子，填写帖子基本情况 |
| delete() | 删除禁用无效的论坛帖子信息记录 |
| update() | 不同身份在前后台修改论坛帖子各项基本情况。 |
| select() | 按一定的要求条件查询相关论坛帖子信息。 |

表 6-13 论坛帖子服务层类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子管理服务层类 | | |
| 类的描述：封装论坛帖子管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加论坛帖子，填写帖子基本情况 |
| delete() | 删除禁用无效的论坛帖子信息记录 |
| update() | 不同身份在前后台修改论坛帖子各项基本情况。 |
| select() | 按一定的要求条件查询相关论坛帖子信息。 |

表 6-14 论坛帖子管理数据访问层类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子管理数据库操作类 | | |
| 类的描述：管理论坛帖子管理数据库中的数据 | | |
| 操作描  述 | create() | 添加论坛帖子，填写帖子基本情况 |
| delete() | 删除禁用无效的论坛帖子信息记录 |
| update() | 不同身份在前后台修改论坛帖子各项基本情况。 |
| select() | 按一定的要求条件查询相关论坛帖子信息。 |

表 6-15 论坛帖子管理实体类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子实体类 | | |
| 类的描述：论坛帖子实体对象类 | | |
| 属性描述 | forum\_id | 论坛帖子 id |
| user\_id | 发布者编号 |
| image | 帖子照片 |
| title | 帖子标题 |
| content | 帖子内容 |
| is\_effect | 帖子是否有效 |
| create\_time | 创建时间 |
| 操作描  述 | Setter（） Getter（） | 对属性的 set 与 get 方法 |

* + 1. 顺序图设计



图 6-11 修改论坛帖子时序图

* + 1. 核心处理流程设计

论坛帖子管理功能核心处理流程图如下图所示：

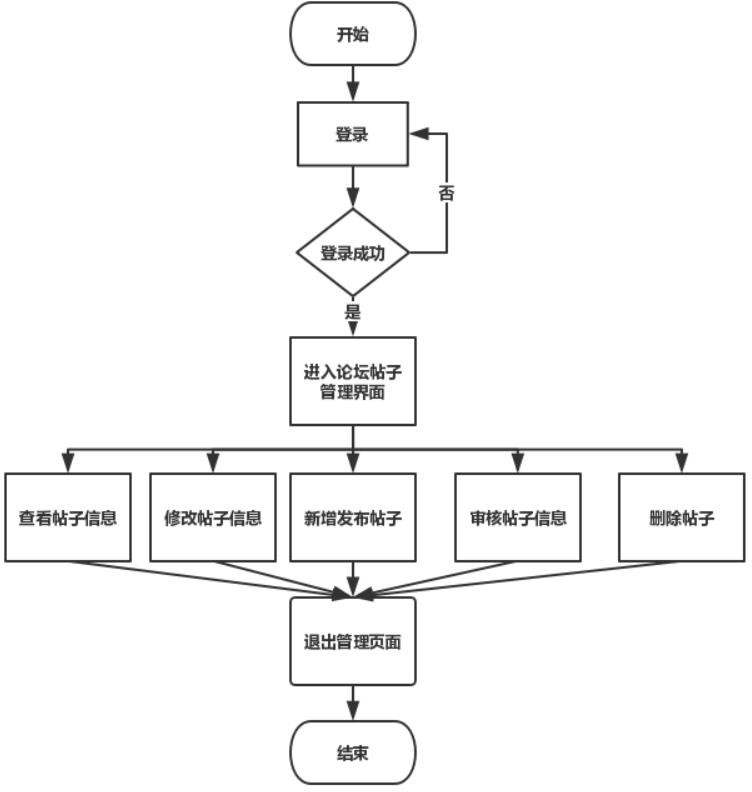


图 6-12 论坛帖子管理流程图

### 广告管理模块

* + 1. 功能结构设计

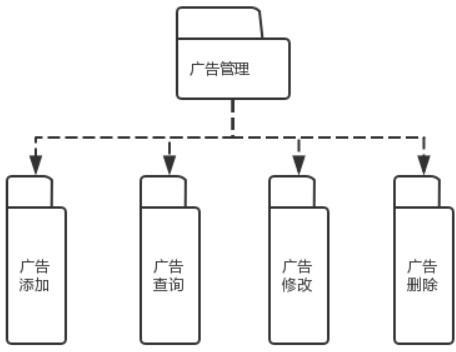


图 6-13 广告管理包图

* + 1. 类图设计

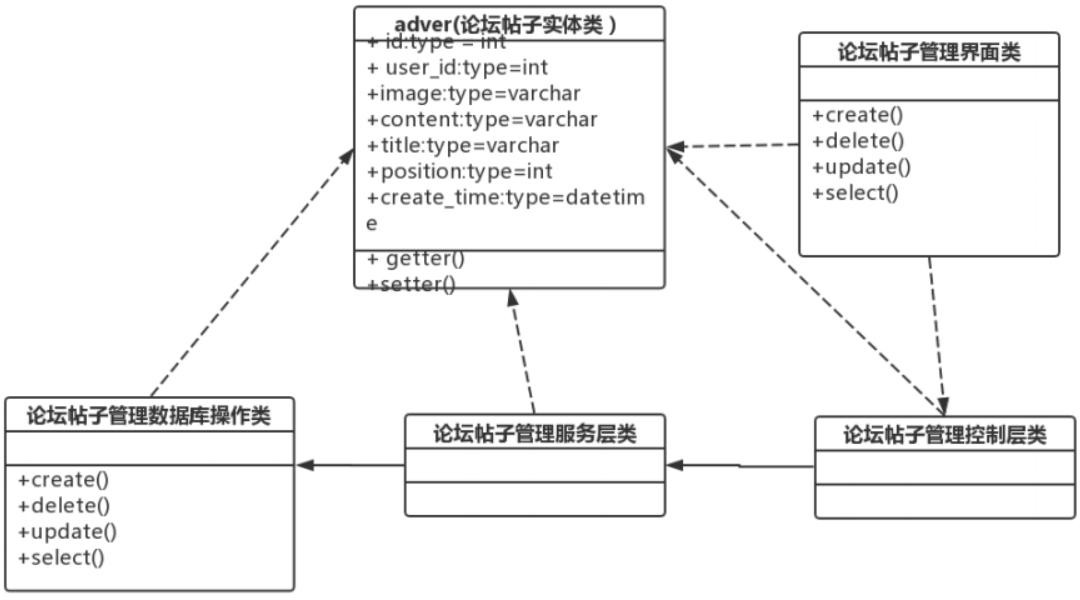


图 6-14 广告管理类图

表 6-16 广告管理面类表

|  |  |  |
| --- | --- | --- |
| 广告管理界面类 | | |
| 类的描述：对我们的广告管理在前后端进行图形化界面操作 | | |
| 操作描  述 | create() | 在前台或者后台添加广告，填写广告基本情况 |
| delete() | 删除禁用无效的广告信息记录 |
| update() | 不同身份在前后台修改广告各项基本情况。 |
| select() | 按一定的要求条件查询相关广告信息。 |

表 6-17 广告管理控制器类表

|  |  |  |
| --- | --- | --- |
| 广告管理控制器类 | | |
| 类的描述：封装广告管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加广告，填写广告基本情况 |
| delete() | 删除禁用无效的广告信息记录 |
| update() | 不同身份在前后台修改广告各项基本情况。 |
| select() | 按一定的要求条件查询相关广告信息。 |

表 6-18 广告管理服务层类表

|  |  |  |
| --- | --- | --- |
| 广告管理服务层类 | | |
| 类的描述：封装广告管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加广告，填写广告基本情况 |
| delete() | 删除禁用无效的广告信息记录 |
| update() | 不同身份在前后台修改广告各项基本情况。 |
| select() | 按一定的要求条件查询相关广告信息。 |

表 6-19 广告管理数据访问层类表

|  |  |  |
| --- | --- | --- |
| 广告管理数据库操作类 | | |
| 类的描述：管理广告管理数据库中的数据 | | |
| 操作描  述 | create() | 添加广告，填写广告基本情况 |
| delete() | 删除禁用无效的广告信息记录 |
| update() | 不同身份在前后台修改广告各项基本情况。 |
| select() | 按一定的要求条件查询相关广告信息。 |

表 6-20 广告管理实体类表

|  |  |  |
| --- | --- | --- |
| 广告实体类 | | |
| 类的描述：广告实体对象类 | | |
| 属性描述 | id | 广告 id |
| user\_id | 发布者编号 |
| image | 广告照片 |
| title | 广告标题 |
| content | 广告内容 |
| position | 广告位置 |
| create\_time | 广告创建时间 |
| 操作描  述 | Setter（） Getter（） | 对属性的 set 与 get 方法 |

* + 1. 顺序图设计

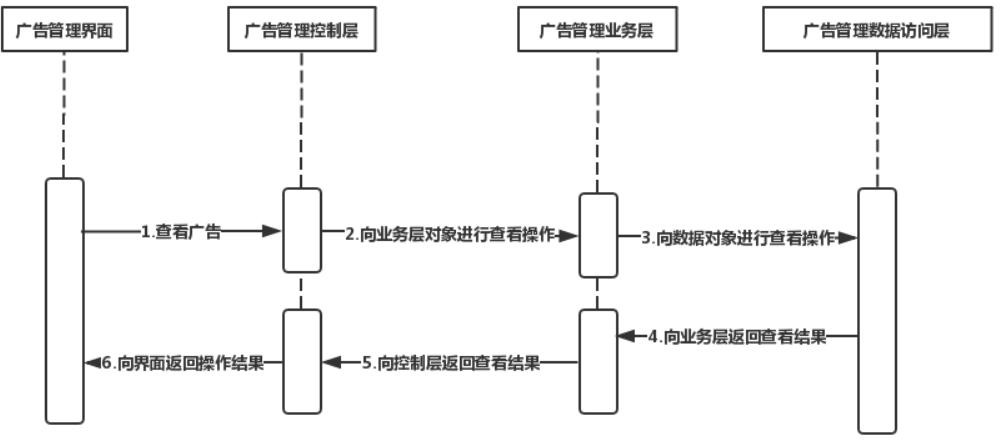


图 6-15 查看广告时序图

* + 1. 核心处理流程设计

广告管理功能核心处理流程图如下图所示：

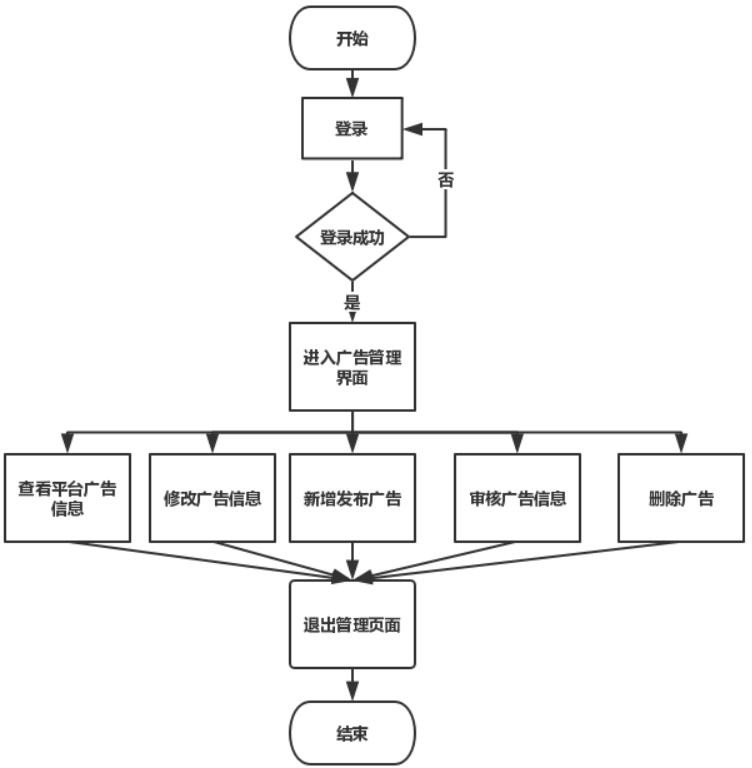


图 6-16 广告管理流程图

### 系统管理模块

* + 1. 功能结构设计

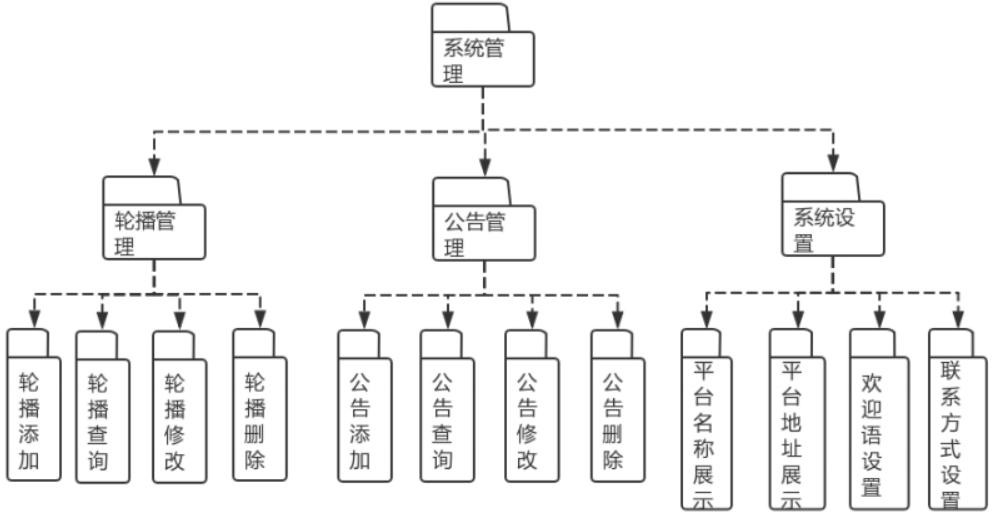


图 6-17 系统管理包图

* + 1. 类图设计

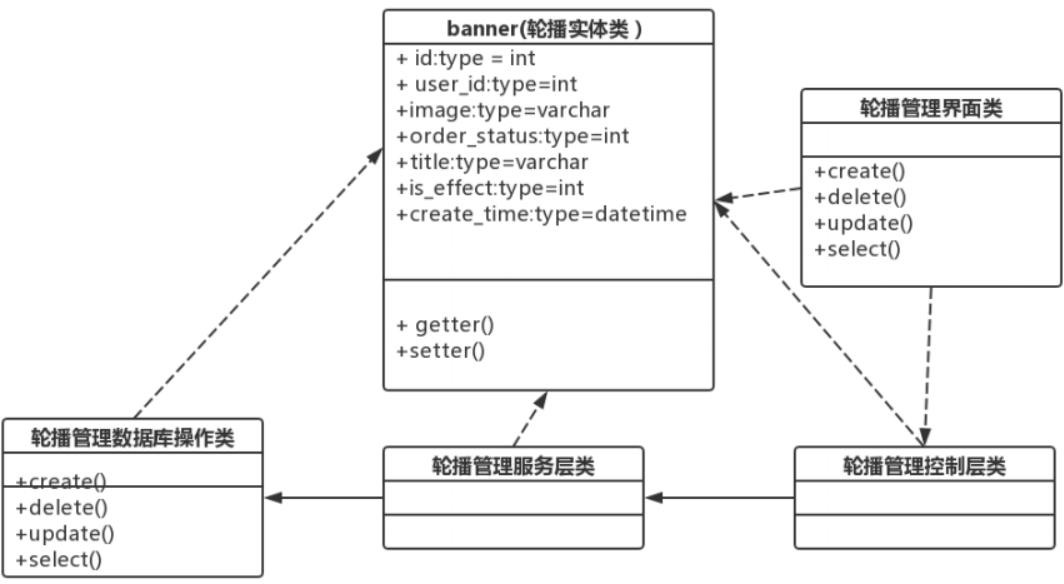


图 6-18 轮播管理类图

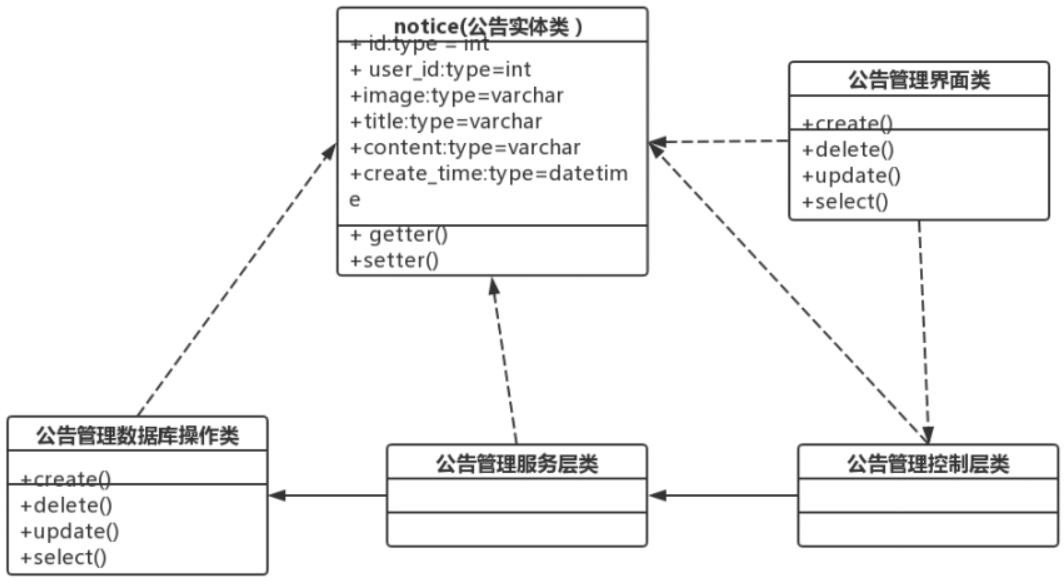


图 6-19 公告管理类图

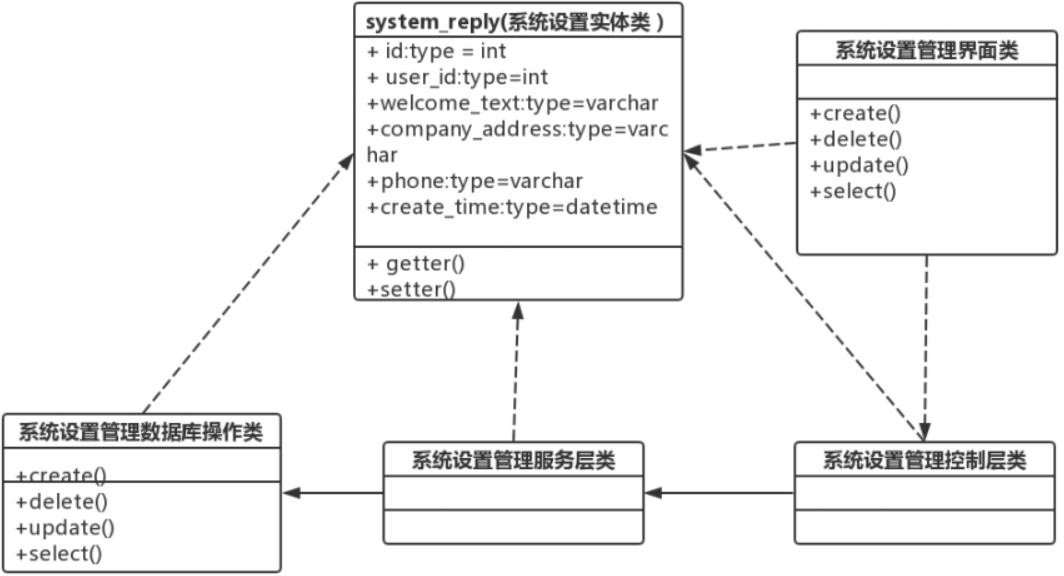


图 6-20 系统设置类图

表 6-21 轮播管理面类表

类的描述：对我们的轮播管理在前后端进行图形化界面操作

轮播管理界面类

续表 6-21

|  |  |  |
| --- | --- | --- |
| 操作描  述 | create() | 在前台或者后台添加轮播，填写轮播基本情况 |
| delete() | 删除禁用无效的轮播信息记录 |
| update() | 不同身份在前后台修改轮播各项基本情况。 |
| select() | 按一定的要求条件查询相关轮播信息。 |

表 6-22 轮播管理控制器类表

|  |  |  |
| --- | --- | --- |
| 轮播管理控制器类 | | |
| 类的描述：封装轮播管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加轮播，填写轮播基本情况 |
| delete() | 删除禁用无效的轮播信息记录 |
| update() | 不同身份在前后台修改轮播各项基本情况。 |
| select() | 按一定的要求条件查询相关轮播信息。 |

表 6-23 轮播管理服务层类表

|  |  |  |
| --- | --- | --- |
| 轮播管理服务层类 | | |
| 类的描述：封装轮播管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加轮播，填写轮播基本情况 |
| delete() | 删除禁用无效的轮播信息记录 |
| update() | 不同身份在前后台修改轮播各项基本情况。 |
| select() | 按一定的要求条件查询相关轮播信息。 |

表 6-24 轮播管理数据访问层类表

|  |  |  |
| --- | --- | --- |
| 轮播管理数据库操作类 | | |
| 类的描述：管理轮播管理数据库中的数据 | | |
| 操作描  述 | create() | 添加轮播，填写轮播基本情况 |
| delete() | 删除禁用无效的轮播信息记录 |
| update() | 不同身份在前后台修改轮播各项基本情况。 |
| select() | 按一定的要求条件查询相关轮播信息。 |

表 6-25 轮播管理实体类表

类的描述：轮播实体对象类

轮播实体类

续表 6-25

|  |  |  |
| --- | --- | --- |
| 属性描述 | id | 轮播 id |
| user\_id | 发布者编号 |
| image | 轮播照片 |
| title | 轮播标题 |
| content | 轮播内容 |
| position | 轮播位置 |
| 操作描  述 | Setter（） Getter（） | 对属性的 set 与 get 方法 |

表 6-26 公告管理面类表

|  |  |  |
| --- | --- | --- |
| 公告管理界面类 | | |
| 类的描述：对我们的公告管理在前后端进行图形化界面操作 | | |
| 操作描  述 | create() | 在前台或者后台添加公告，填写公告基本情况 |
| delete() | 删除禁用无效的公告信息记录 |
| update() | 不同身份在前后台修改公告各项基本情况。 |
| select() | 按一定的要求条件查询相关公告信息。 |

表 6-27 公告管理控制器类表

|  |  |  |
| --- | --- | --- |
| 公告管理控制器类 | | |
| 类的描述：封装公告管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加公告，填写公告基本情况 |
| delete() | 删除禁用无效的公告信息记录 |
| update() | 不同身份在前后台修改公告各项基本情况。 |
| select() | 按一定的要求条件查询相关公告信息。 |

表 6-28 公告管理服务层类表

|  |  |  |
| --- | --- | --- |
| 公告管理服务层类 | | |
| 类的描述：封装公告管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加公告，填写公告基本情况 |
| delete() | 删除禁用无效的公告信息记录 |
| update() | 不同身份在前后台修改公告各项基本情况。 |
| select() | 按一定的要求条件查询相关公告信息。 |

表 6-29 公告管理数据访问层类表

|  |  |  |
| --- | --- | --- |
| 公告管理数据库操作类 | | |
| 类的描述：管理公告管理数据库中的数据 | | |
| 操作描  述 | create() | 添加公告，填写公告基本情况 |
| delete() | 删除禁用无效的公告信息记录 |
| update() | 不同身份在前后台修改公告各项基本情况。 |
| select() | 按一定的要求条件查询相关公告信息。 |

表 6-30 公告管理实体类表

|  |  |  |
| --- | --- | --- |
| 公告实体类 | | |
| 类的描述：公告实体对象类 | | |
| 属性描述 | id | 公告 id |
| user\_id | 发布者编号 |
| image | 公告照片 |
| title | 公告标题 |
| content | 公告内容 |
| position | 公告位置 |
| create\_time | 公告创建时间 |
| 操作描  述 | Setter（） Getter（） | 对属性的 set 与 get 方法 |

表 6-31 系统设置实体类表

|  |  |  |
| --- | --- | --- |
| 系统设置实体类 | | |
| 类的描述：系统设置实体对象类 | | |
| 属性描述 | id | 公告 id |
| user\_id | 用户编号 |
| welcom\_text | 平台欢迎语 |
| company\_addres | 平台地址 |
| phone | 平台联系方式 |
| we\_chat | 微信 |
| company\_logo | 平台 LOGO |
| create\_time | 创建时间 |

续表 6-31

|  |  |  |
| --- | --- | --- |
| 操作描  述 | Setter（） Getter（） | 对属性的 set 与 get 方法 |

* + 1. 顺序图设计

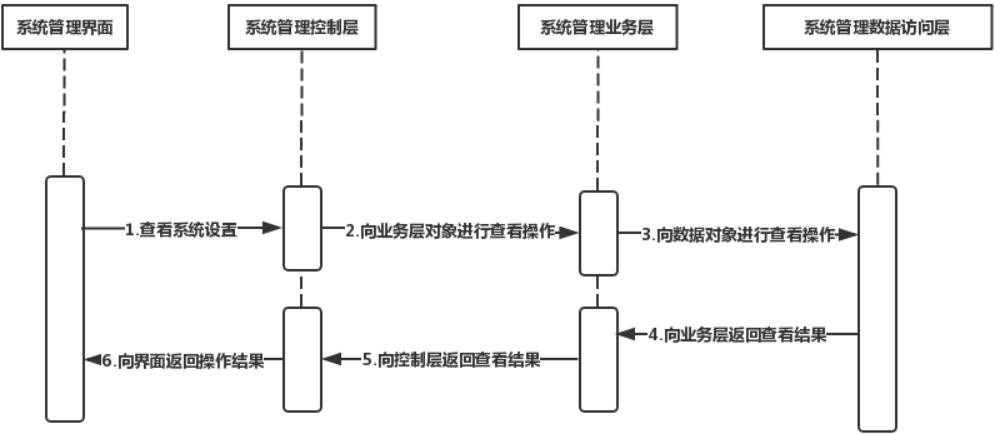


图 6-21 查看系统设置时序图

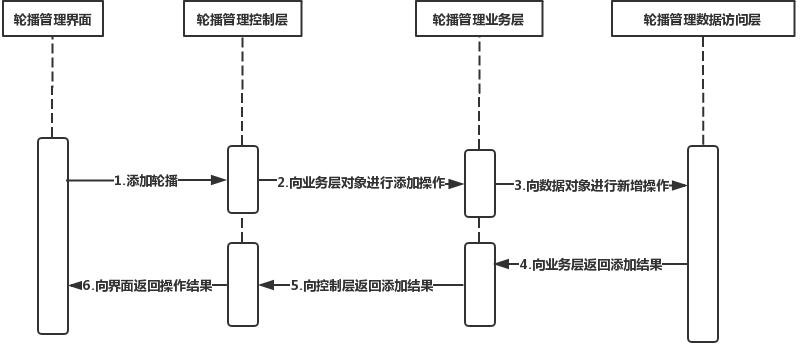


图 6-22 轮播添加管理时序图

* + 1. 核心处理流程设计

系统管理功能核心处理流程图如下图所示：

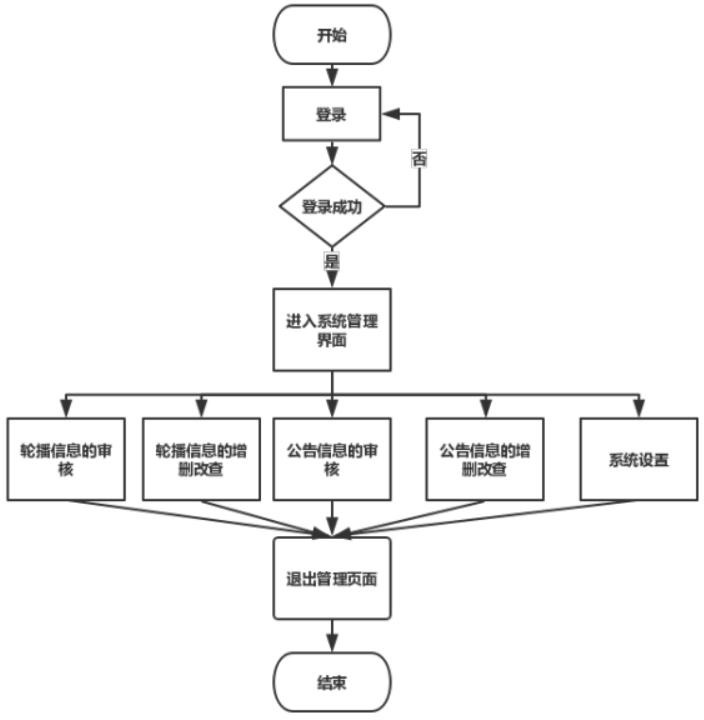


图 6-23 系统管理流程图

### 聊天管理模块

* + 1. 功能结构设计

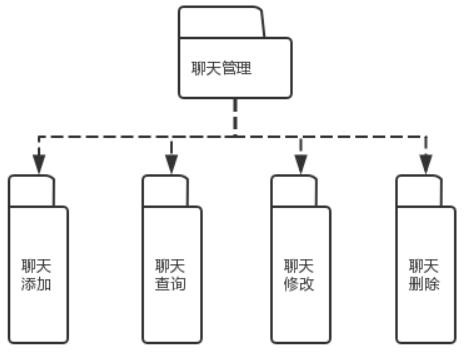


图 6-24 聊天管理包图

* + 1. 类图设计



图 6-25 聊天管理功能类图

表 6-32 聊天管理面类表

|  |  |  |
| --- | --- | --- |
| 聊天管理界面类 | | |
| 类的描述：对我们的聊天管理在前后端进行图形化界面操作 | | |
| 操作描  述 | create() | 在前台或者后台发起聊天，输入聊天信息 |
| delete() | 删除禁用无效的聊天信息记录 |
| update() | 不同身份在前后台修改聊天各项基本情况。 |
| select() | 按一定的要求条件查询相关聊天信息。 |

表 6-33 聊天控制器类表

|  |  |  |
| --- | --- | --- |
| 聊天管理控制器类 | | |
| 类的描述：封装聊天管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 在前台或者后台发起聊天，输入聊天信息 |
| delete() | 删除禁用无效的聊天信息记录 |
| update() | 不同身份在前后台修改聊天各项基本情况。 |
| select() | 按一定的要求条件查询相关聊天信息。 |

表 6-34 聊天服务层类表

|  |  |  |
| --- | --- | --- |
| 聊天管理服务层类 | | |
| 类的描述：封装聊天管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 在前台或者后台发起聊天，输入聊天信息 |
| delete() | 删除禁用无效的聊天信息记录 |
| update() | 不同身份在前后台修改聊天各项基本情况。 |
| select() | 按一定的要求条件查询相关聊天信息。 |

表 6-35 聊天管理数据访问层类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子管理数据库操作类 | | |
| 类的描述：管理聊天管理数据库中的数据 | | |
| 操作描  述 | create() | 在前台或者后台发起聊天，输入聊天信息 |
| delete() | 删除禁用无效的聊天信息记录 |
| update() | 不同身份在前后台修改聊天各项基本情况。 |
| select() | 按一定的要求条件查询相关聊天信息。 |

表 6-36 聊天管理实体类表

|  |  |  |
| --- | --- | --- |
| 聊天实体类 | | |
| 类的描述：聊天实体对象类 | | |
| 属性描述 | id | 聊天信息 id |
| user\_id\_fa | 聊天发送方 |
| user\_id\_jie | 聊天接收方 |
| image | 图片 |
| title | 聊天主题 |
| content | 聊天内容 |
| create\_time | 创建时间 |
| is\_look | 消息是否查看 |
| is\_remove\_fa | 发送方判断是否删除 |
| is\_remove\_jie | 接收方判断是否删除 |
| chat\_signal | 聊天信号 |
| 操作描  述 | Setter（） Getter（） | 对属性的 set 与 get 方法 |

* + 1. 顺序图设计

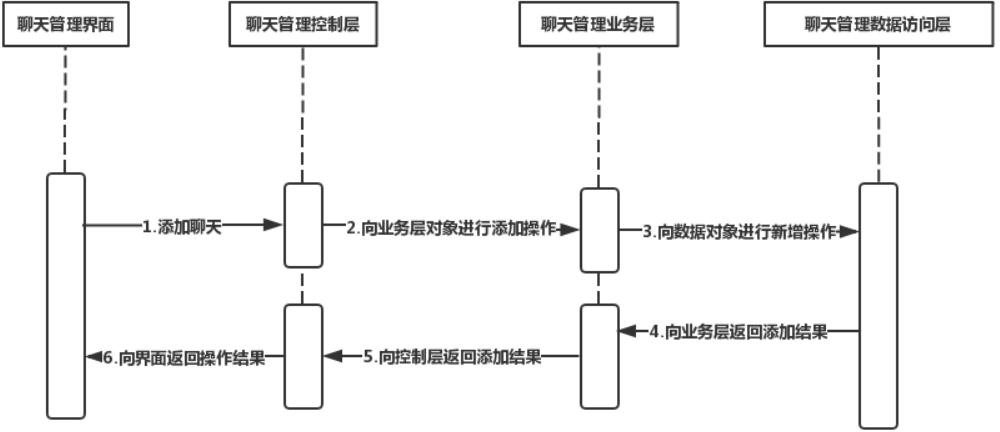


图 6-26 添加聊天时序图

* + 1. 核心处理流程设计

聊天管理功能核心处理流程图如下图所示：

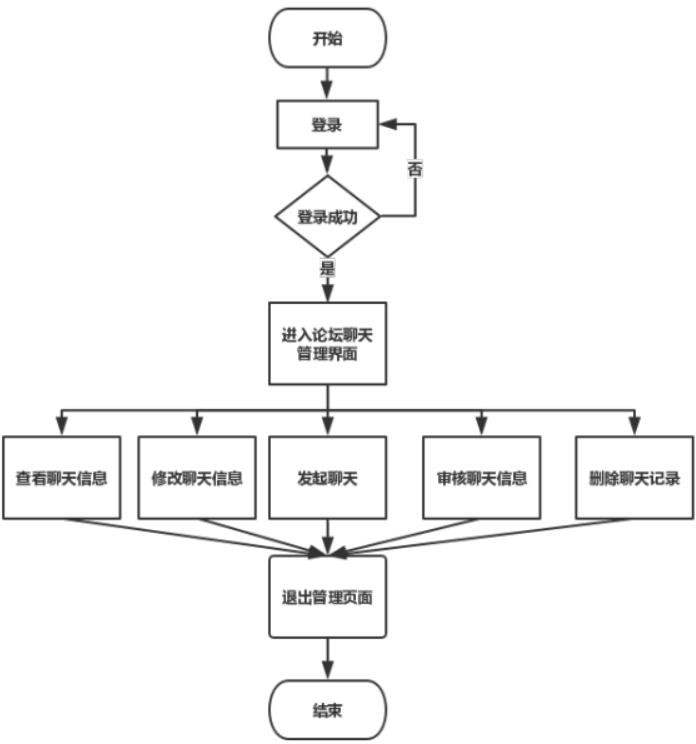


图 6-27 聊天流程图

### 收藏管理模块

* + 1. 功能结构设计

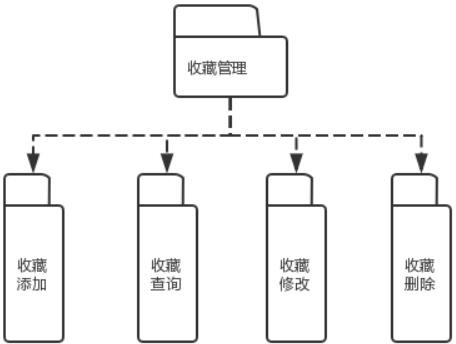


图 6-28 收藏管理包图

* + 1. 类图设计

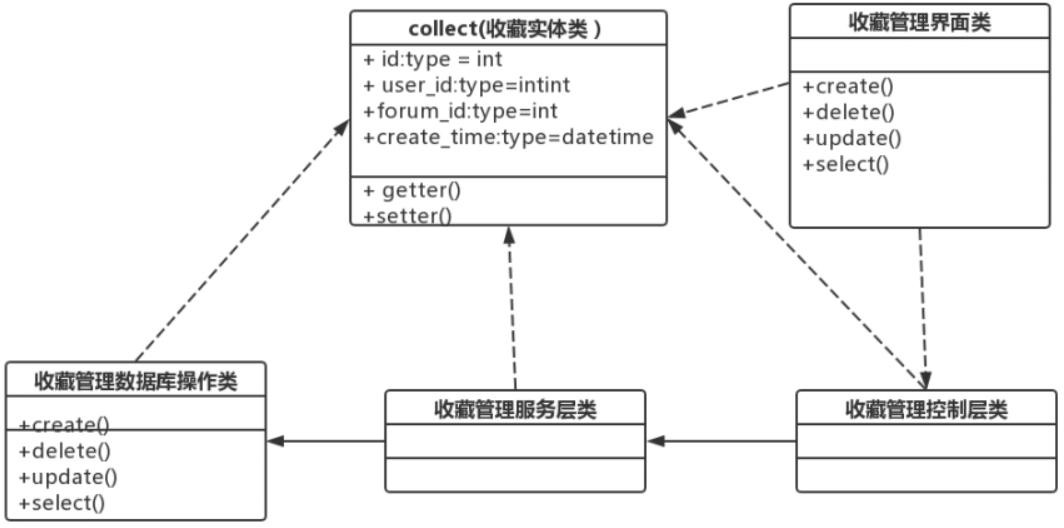


图 6-29 收藏管理类图

表 6-37 收藏管理面类表

|  |  |  |
| --- | --- | --- |
| 收藏管理界面类 | | |
| 类的描述：对我们的论坛帖子收藏管理在前后端进行图形化界面操作 | | |
| 操作描述 | create() | 在前台或者后台添加论坛帖子收藏，填写帖子收藏基本情况 |
| delete() | 删除禁用无效的论坛帖子收藏信息记录 |
| update() | 不同身份在前后台修改论坛帖子收藏各项基本情况。 |
| select() | 按一定的要求条件查询相关论坛帖子收藏信息。 |

表 6-38 论坛帖子收藏控制器类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子收藏管理控制器类 | | |
| 类的描述：封装论坛帖子收藏管理操作的各项服务层方法 | | |
| 操作描  述 | create() | 添加论坛帖子收藏，填写帖子收藏基本情况 |
| delete() | 删除禁用无效的论坛帖子收藏信息记录 |
| update() | 不同身份在前后台修改论坛帖子收藏各项基本情况。 |
| select() | 按一定的要求条件查询相关论坛帖子收藏信息。 |

表 6-39 论坛帖子收藏服务层类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子收藏管理服务层类 | | |
| 类的描述：封装论坛帖子收藏管理操作的各项服务层方法 | | |
| 操作描述 | create() | 添加论坛帖子收藏，填写帖子收藏基本情况 |
| delete() | 删除禁用无效的论坛帖子收藏信息记录 |
| update() | 不同身份在前后台修改论坛帖子收藏各项基本情况。 |
| select() | 按一定的要求条件查询相关论坛帖子收藏信息。 |

表 6-40 论坛帖子收藏管理数据访问层类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子收藏管理数据库操作类 | | |
| 类的描述：管理论坛帖子收藏数据库中的数据 | | |
| 操作描述 | create() | 添加论坛帖子收藏，填写帖子收藏基本情况 |
| delete() | 删除禁用无效的论坛帖子收藏信息记录 |
| update() | 不同身份在前后台修改论坛帖子收藏各项基本情况。 |
| select() | 按一定的要求条件查询相关论坛帖子收藏信息。 |

表 6-41 论坛帖子收藏管理实体类表

|  |  |  |
| --- | --- | --- |
| 论坛帖子收藏实体类 | | |
| 类的描述：论坛帖子收藏实体对象类 | | |
| 属性描  述 | id | 论坛帖子 id |
| user\_id | 收藏者编号 |
| forum\_id | 帖子编号 |
| Create\_time | 创建时间 |
| 操作描  述 | Setter（） Getter（） | 对属性的 set 与 get 方法 |

* + 1. 顺序图设计



图 6-30 删除帖子收藏时序图

* + 1. 核心处理流程设计

帖子收藏管理功能核心处理流程图如下图所示：

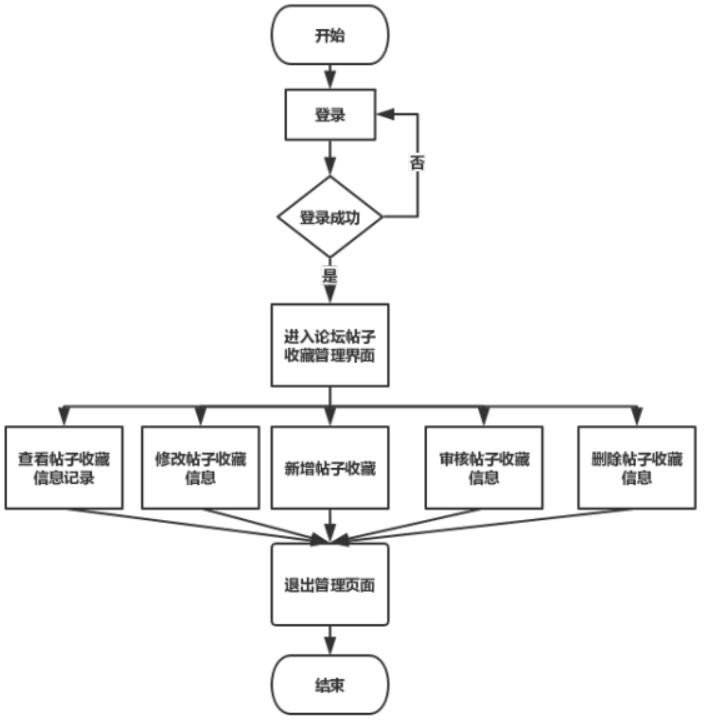


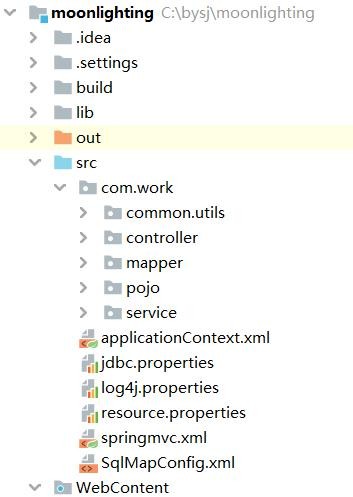
图 6-31 帖子收藏流程图

# 编码

### 代码实现与核心算法

* + 1. 代码的目录结构与接口

我们的项目是在 Idea 中编写完成的，采用 maven 项目管理工具来管理。具体的代码目录结构如下图：



在资源目录 src 的项目下，首先编写实体类 pojo，定义对象的属性，通过引用数据库中表的字段进行设置，然后编写 Mapper.xml（Mybatis），定义你的函数并与数据库相对应这些操作（例如 CRUD），然后开发 Mapper.java 层代码，第二个是服务层代码的开发编写，接受来自控制层的参数。再接着编写控制层 java 代码。最后，编写要调用的 JSP 页面代码，请求哪些参数以及需要获取哪些数据。

* + 1. 核心算法代码实现

全局异常处理相关代码：



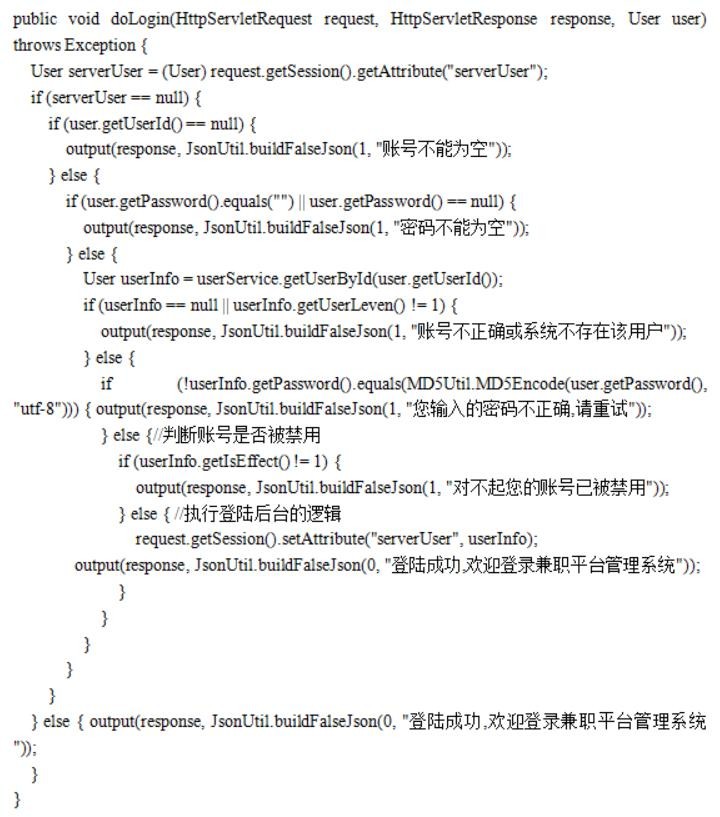
页面跳转相关代码：



返回总记录数相关代码：



执行登录相关代码：



上传与下载相关代码:



### 代码优化分析

项目在开发的过程中，虽然有一定的模块化规范开发，但是在开发过程中我们的代码的开发过程中，仍然的存在很多重复性的冗余的代码，这样一来大大的浪费了很多的空间。同时在规范化命名的这一方面我们也有很大的提高空间，规范化的命名可以让在未来的团

队开发省去很多的不必要的麻烦和障碍，规范化命名也是我们从一开始就要养成大的一个好的习惯。最后我们的代码在注释这一方面也需要很大的改进，一个好的项目代码，完整规范的代码注释是必不可少的。

# 测试

### 测试方案设计

* + 1. 测试策略

在系统基本完成后，对我们的平台进行系统测试，确定[测试过程](https://www.baidu.com/s?wd=%E6%B5%8B%E8%AF%95%E8%BF%87%E7%A8%8B&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)要使用的测试技术和工具，以及在测试过程中相对应的标准。

* + 1. 测试进度安排

对我们平台用户操作的涉及到的每一个功能点进行测试，在每完成一个点就进行相对应的小的功能测试，最后整体的项目完工后，在进行详细的系统化整体测试。

* + 1. 测试资源

本项目的测试人员同时为我们项目的开发人员，测试软件是我们的 web 浏览器。

* + 1. 关键测试点

1. 用户注册测试
2. 用户登录测试
3. 管理员登录测试
4. 用户浏览职位信息测试
5. 用户聊天测试
6. 用户发帖测试
7. 用户帖子评论测试
8. 用户收藏帖子测试
9. 管理员管理后台测试

### 测试用例构建

* + 1. 测试用例编写约定

1. 我们在测试的过程中，针对某一个需要测试的点，要编写好多条不同的测试用例和

数据。

1. 在项目开发的不同阶段，由我们的开发人员或者测试人员编写不同的测试用例
2. 对所要测试的功能应做好分块的不同测试
3. 在测试的同时我们的而测试人员要正确的操作步骤。用例的编写要符合我们的测试的功能。
   * 1. 测试用例设计

我们的测试用例编写包括测试模块、各项功能、预期目标、测试结果。此次的测试有以下几个：用户注册测试、用户登录测试、管理员登录测试、用户浏览职位信息测试、用户聊天测试、用户发帖测试、用户帖子评论测试、用户收藏帖子测试、管理员管理后台测试。

* + 1. 关键测试用例

表 8-1 关键测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试模块** | **各项功能** | **预期目标** | **测试结果** |
| 用户注册登录 | 新用户注册 | 将注册页面所需的信息都填写完整提交后，提示注册成功 | 完成 |
| 用户登录 | 输入正确的密码和账号都进入到前台页面 | 完成 |
| 密码找回 | 通过注册时设置的密保问题找回密码 | 完成 |
| 登录推出 | 我们的平台用户在点击退出按钮后即可退出登录 | 完成 |
| 管理员登录 | 管理员在后台登陆页面登陆 | 输入正确的管理员账号和密码后进入到后台管  理页面 | 完成 |
| 用户浏览职位信息测试 | 用户浏览不同的职位信息测试 | 用户在前台页面通过列表展示滑动浏览我们的兼职招聘职位信息，点  击可查看信息详情 | 完成 |

续表 8-1

|  |  |  |  |
| --- | --- | --- | --- |
| 用户聊天测试 | 用户和感兴趣职位的发布者聊天 | 对有感兴趣职位的发布者头像点击，选择与 ta 聊天便可发起聊天 | 完成 |
| 用户发帖测试 | 用户在论坛模块发帖 | 在论坛模块编辑我们的内容，点击提交。之后便可在论坛看到我们所  发的信息 | 完成 |
| 用户帖子评论测试 | 用户帖子评论 | 在论坛浏览我们的帖子  时，可点击评论编辑框编写评论并提交 | 完成 |
| 用户帖子收藏测试 | 用户帖子收藏 | 在论坛浏览我们的帖子  时，可点击收藏按钮收藏感兴趣的帖子 | 完成 |
| 管理员管理后台测试。 | 管理员管理对后台各项  功能信息的新增 | 点击新增按钮便可添加  新的记录 | 完成 |
| 管理员管理对后台各项  功能信息的编辑 | 点击编辑按钮便可对信  息进行修改 | 完成 |
| 管理员管理对后台各项  功能信息的删除 | 点击禁用按钮便可删除  信息 | 完成 |
| 管理员管理对后台各项  功能信息的查看 | 点击搜索便可查看相关  的信息 | 完成 |

* + 1. 测试用例维护

1. 对我们在测试中找出的问题进行维护修改，使用不同的测试用例再次进行测试
2. 在后期的新的开发中对新增加的功能也需进行测试

# 总结与展望

### 设计工作总结

本次的毕业设计，是在考研结束后开始的。由于去年大半年的时间都用来准备 12 月份的研究生初试。在大学的几年本科生涯学习中虽然书本上的理论知识学习的还算是可以， 但是大体上来说在动手方面还是有很大的欠缺，所以此次的项目设计完成也是遇到了很多的问题，但是最终经过多个月的努力，终于大概的完成了毕业设计，系统也能够基本上运行跑起来。

我们主要完成了一下几点工作：

1. 对我们社会上目前兼职平台环境的发展趋势和现状进行了解
2. 对 SSM 框架相关的知识进行一次较为详细的学习和应用
3. 对大学生兼职平台进行系统的需求分析和系统设计
4. 完成数据库的设计和建设
5. 对兼职平台进行开发和编码实现

由于是没有选者出去校外的实训，代码的动手能力不是很强很扎实。但是从确定选题开始便也确定了自己项目开发所用的框架等。同时技术也不是很强，于是开始自己跟随着视频教学等一步步的学习，学习如何使用框架开发一个完整的项目。

这次的毕业设计正好也处在疫情之中，大家都是每个人自己在家开发。在开发的过程中也是遇到了很多的问题，本来的选题是三人小组的某市高新区管委会自动化办公系统， 但是由于疫情原因无法技术很好的沟通交流，于是换了现在的选题。在确定选题后大开发过程中很多问题自己都不懂，可以问他人，最多的是自己百度解决。这一次的设计过程也使得自己的解决问题的能力有所提升，有的时候百度就是一个很好的老师。我的这个项目是 SSM 框架的项目，这个框架在之前也没有怎么接触和学习。通过此次的学习和实战的经历，我相信对我以后在学习其他的知识方面会有更大的帮助。

总的来说，这一次的毕业设计的过程，更多的是给自己一个锻炼提升的好机会，让自己变得更加的有耐心，有毅力和面对困难的决心。

### 未来工作展望

我们的高校大学生兼职平台的功能实现虽然大概的达到了预期的效果，但是在后期还是需要完善很多。功能需求也会随着我们在校大学生的意见不断地进行更新改进，同时也要利用新的技术来优化升级我们的平台系统，使得系统更加的稳定和功能强大。

此次的毕业设计过后，就将要正式的结束自己的大学本科生涯了。在以后的专业工作当中，我希望自己能够在代码的编写过程中更加的规范严谨。不管以后是否还会在从事业专业相关的工作，我觉得做事规范严谨的态度是未来所有工作都需要的，也是必不可少的。

# 谢 辞

# 参考文献

[1]吴进毅, 赵鑫, 杨世浩, et al. 大学生兼职网站的设计与开发[J]. 电脑知识与技术, 2019, 15(02):69-70. [2]程亚男. (2014). 基于 web 的大学生兼职管理系统的设计与开发. 信息通信(05), 104.

[3][论高校对大学生校外兼职的因应策略](https://kns.cnki.net/kcms/detail/detail.aspx?filename=SZJY200801057&dbcode=CJFQ&dbname=CJFD2008&v)[J]. 唐其宝. 徐州教育学院学报. 2008(01) [4][大学生兼职的动因分析](https://kns.cnki.net/kcms/detail/detail.aspx?filename=CQGY200801089&dbcode=CJFQ&dbname=CJFD2008&v)[J]. 李倩. 重庆科技学院学报(社会科学版). 2008(01) [5][大学生兼职平台建设的可行性研究](https://kns.cnki.net/kcms/detail/detail.aspx?filename=CYYT201708070&dbcode=CJFQ&dbname=CJFD2017&v)[J]. 曹源. 产业与科技论坛. 2017(08)

1. [大学生兼职行为特征研析](https://kns.cnki.net/kcms/detail/detail.aspx?filename=ZGQL201401023&dbcode=CJFQ&dbname=CJFD2014&v)[J]. 姚远. 中国青年研究. 2014(01)
2. 贾晓芳, 沈泽刚.JavaWeb 应用开发中的常见乱码形式及解决方法[J].软件导刊, 2017 [8]王循.javaweb 快速开发框架中部分关键技术初步研究[D].吉林大学，2015.

[9] 翟剑锟.Spring 框架技术分析及应用研究[D].中国科学院大学（工程管理与信息技术学院），2013 [10]王珊，萨师煊. 数据库系统概论(第 5 版)[M]. 北京：高等教育出版社，2014.

1. 仲萃豪. 软件开发与软件架构[M]. 北京：科学出版社，2013.
2. 姜桂洪. MySQL 数据库应用与开发[M]. 北京：清华大学出版社，2018. [13]张甦. MySQL 王者晋级之路[M]. 北京：电子工业出版社，2018.
3. 陈晓勇. MySQL DBA 修炼之道[M]. 北京：机械工业出版社，2017.
4. 贺春旸. MYSQL 管理之道:性能调优、高可用与监控[M]. 北京：机械工业出版社，2016. [16]杨开振. 深入浅出 MyBatis 技术原理与实战[M]. 北京：电子工业出版社，2016.

[17]王福强. Spring Boot 揭秘:快速构建微服务体系[M]. 北京：机械工业出版社，2016. [18]肖睿，龙颖. Vue 企业开发实战[M]. 北京：人民邮电出版社，2018.

[19]陈陆扬. Vue.js 前端开发快速入门与专业应用[M]. 北京：人民邮电出版社，2017. [20]张耀春. Vue.js 权威指南[M]. 北京： 电子工业出版社，2016.

[21]刘增辉. MyBatis 从入门到精通[M]. 北京：电子工业出版社，2017. [22]仲萃豪. 软件开发与软件架构[M]. 北京：科学出版社，2013.

[23]姜桂洪. MySQL 数据库应用与开发[M]. 北京：清华大学出版社，2018. [24]张甦. MySQL 王者晋级之路[M]. 北京：电子工业出版社，2018.

1. 陈晓勇. MySQL DBA 修炼之道[M]. 北京：机械工业出版社，2017.
2. 贺春旸. MYSQL 管理之道:性能调优、高可用与监控[M]. 北京：机械工业出版社，2016. [27]郑阿奇. Java EE 实用教程[M]. 北京：电子工业出版社，2018.
3. 尉哲明. 基于 Java 的综合课程设计[M]. 北京：清华大学出版社，2014.
4. 刘增杰. MySQL5.7 从入门到精通[M]. 北京：清华大学出版社，2016.
5. Abdelhak Mesbah,Jean-Louis Lanet,Mohamed Mezghiche. Reverse engineering a Java Card memory management algorithm[J]. Computers & Security,2017,66:
6. Yao, Dengke, Su, Shaowei, Feng, Xiaoyong. Method and system for file transfer, instant messaging

terminal, and computer storage medium[J]. 2018.

## 附录 A 外文翻译—原文部分

[Di Lu,](http://scholar.cnki.net/new/home/search?ad=1&searchWord-input-author=Di%20Lu)[Yuping Qiu,](http://scholar.cnki.net/new/home/search?ad=1&searchWord-input-author=Yuping%20Qiu)[Chun Qian,](http://scholar.cnki.net/new/home/search?ad=1&searchWord-input-author=Chun%20Qian)[Xi Wang,](http://scholar.cnki.net/new/home/search?ad=1&searchWord-input-author=Xi%20Wang)[Wei Tan](http://scholar.cnki.net/new/home/search?ad=1&searchWord-input-author=Wei%20Tan).Design of Campus Resource Sharing Platform based on SSM Framework

Currently, there is no systematic and comprehensive resource sharing service platform for undergraduate students to acquire various learning resources,which leads to the phenomenon of low utilization rate of learning resources and high repetition rate of resources.The ultimate goal of this paper is to design a campus cloud resource sharing service platform to resolve this problem.The platform will enable undergraduate students to acquire learning resources in a more convenient comprehensive way through scattered learning resources integration.Through the campus resource sharing system, we can search the learning materials we need online at any time and any place.We can also discuss problems through the mode of regional discussion.Resources can be divided into explicit resources and implicit resources. Explicit resources are ordinary uploaded resources, which can be directly searched by users. Invisible resources can be obtained through discussion and evaluation, which is more convenient for users to learn.

The resource sharing system adopts MVC mode, which can separate the page display, domain logic and database access, making it convenient for developers to develop and maintain the system and redevelop the system.When making frame selection, we compared common MVC framework, SSH framework and SSM framework.In the end, we chose the SSM integration framework (the integration of three open-sourcing frameworks,Spring MVC + Spring + Mybatis, or SSM for short).The SSM integration framework can be very convenient, and the university develops powerful Web applications. At the same time, it is also a typical MVC framework. It divides the whole system into four layers: presentation layer (JSP and related JS), control layer, service layer and database access layer. MyBatis is an excellent data persistence framework.It supports custom SQL, storing procedures, and advanced mapping.MyBatis framework has both the advantages of Hibernate and JDBC. Programmers can write SQL statements by themselves to facilitate optimization of SQL statements, while avoiding almost all JDBC codes and manually setting parameters and obtaining result sets.

The overall design of the system includes five modules:user management,role management, resource management, comment management and classification management.[5] User management module includes: user registration, user information modification,user basic information query.The background administrator has the functions of adding users, querying all users and managing user status.

Role management module includes:permission management module(only open to the administrator), the role's addition, deletion, modification and check, role’s authorization.

Classification management module includes: add, delete and modify classification. Resource management module includes: learning resource upload,download,description,sorting optimization of resources, resource classification. Comment management module includes: resource comment, topic discussion, comment reply, comment review.

The operating objects of the resource-sharing platform include resources, resource categories, comments, etc. The analysis abstracts the platform into five entities: user, role, resource, comment and classification.The specific relationships among the entities are as follows(Enumeration only): A user can publish multiple resources, a resource is only published by one person, with a one-to-many relationship. A learning resource can be operated by multiple users, and a user can operate multiple resources with many-to-many relationships. A classification can only belong to one large classification, a large classification can have multiple small classification, the relationship is one-to-many.

The resource sharing system adopts B/S three-layer mode,which consists of front-end and background.The front section adopts techniques such as Ace framework

+BootStrap+JQuery+AJAX+CSS and JS encapsulation.The background is mainly SpringMVC+Mybatis +Spring and Shiro.

According to the basic framework of resource-sharing service system, the persistent layer is the compilation of MyBatis.Mybatis framework is divided into two different components, DAO and SQLMap, in which DAO component abstracts the representation location and mode of data access layer and persistent layer of application.

The DAO component is a data access interface that exists between the business logic and the database resources to establish a connection with the database.

In a resource-sharing system, another component in the persistent layer, the SQL Map component, performs SQL statement execution and Java object return through the configured XML file.The configuration file includes the basic configuration and custom configuration of mybatis-config.xml file.

The control layer carries out the course of data acquisition and process, which mainly includes two major categories of Controller and Service.The Controller class makes a request to fetch various types of data and the request data of the front-end user. The Controller class method naming and the request path is unique.

There is also a business logic layer in the control layer that implements logical processing of data.It is divided into logical interface class Manager and implementation class ManagerImpl.

The presentation layer mainly implements two functions, the first is to realize user data input, and the second is to display the data requested by the user.In the preceding Jsp page, the data is displayed by CSS style, JQuery, ajax and other basic methods.Take file upload as an

example, file upload, query, delete, modify, and so on will appear.The implementation method is to write he entire front-end framework for CSS style, JQuery implements some data processing and verification, and ajax implements asynchronous update of data.

This paper aims to design a "campus cloud" resource sharing service platform to improve the efficiency of undergraduate students' access to learning resources.Through the research of SSM framework, this paper determines the development of campus resource sharing system by SSM framework.The development process of the whole system is described in detail from database design to data persistence layer, control layer and presentation layer of the whole framework.Finally, the platform will realize user management, campus resource management, campus resource classification management, comment management and other functions, so that college students can acquire learning resources more conveniently and comprehensively.

Weijie Hua, Hongxiang Ren\* and Mingyang Wang. Ship Cargo Compartment Environment Measurement and Control System Based on SSM Framework

The system is divided into three parts: application layer, cargo compartment environment sensing layer, ventilation and physical control layer. In the environment perception layer, the nodes composed of the precise sensor group and the microcontroller collect the environment information in the cargo compartment and transmit it to the client. In the physical control layer, the microcontroller transfers signals to the relay module, and controls the cargo cabin ventilation system by using the relay on-off. In the application layer, the Web application system and the server monitoring mechanism are constructed. By receiving the connection requests of the Socket client, establishing connection,data parsing, storage and other technologies, the ship company and the shipboard staff can access the server through the PC and APP terminals, the ship company and the ship staff can keep watch on the ship's cargo hold environment parameters and the control of the ventilation system.

The server development in this system is designed by SSM framework. As an enterprise application development framework, it has better performance in application service and data processing, but this system needs to establish Socket service during server startup process, and always monitor the client's request on the ship. Connect and receive the Socket packet sent by it. In the SSM framework, Listener is a Servlet listener that listens for method calls or property changes of an instance object. When the listener object has the above event, a listener method will be executed immediately. After the client configures the network parameters, the TCP connection is set up to the server, so a listener of the listening client is fused on the server side. And add the configuration file< listener-class > com.hu.temp.socket1.ServerSocketListener

</listener-class> in the web.xml file of the SSM framework, and then merge the Socket listener

into the SSM framework. Join the multi-threading mechanism during the period, the server puts the processing flow into a separate thread after each client connection request is received, and then waits for the next client connection request, thereby eliminating the blocking of the server-side receiving request.Achieve multiple ship shipping environment parameters to the server.

The system adopts the mobile WebAPP front-end framework design in the development of user APP. The diversified development method including HTML,CSS and JavaScript technology makes WebAPP have better interactivity, and the advantages of one-time development and multi-platform application can eliminate the difficulty of project transplantation and development, the problem of long development cycle .With MUI as the UI layer, DCloud as the middleware layer, and Android and IOS as the terminal system layer, the mobile WebAPP is slow to run and can't compare the experience of the native APP. MUI does not encapsulate the Domoperation, does not require JavaScript to parse the HTML tag during runtime, reduces the consumption of mobile resources, and speeds up the page loading speed, making the system switches between the ship cargo compartment detection and setting environmental parameter threshold page more smoothly, and the speed is more stable when accessing the server cargo compartment environment data.

System Implementation

The ship cargo compartment environment measurement and control system adopts the SSM framework technology in application layer development, has better access performance and rapid development capability, and is the preferred combination of Web application development framework . The system adopts the MVC software architecture model on the access storage. As shown in Figure 3, the model separates the control layer of the data from the presentation layer, facilitating the subsequent maintenance and expansion of the program, and making the reuse of some programs possible, improving the efficiency of development . Under the framework of SSM, the data stream is read and written to the connected client through the establishment of the input and output stream, and the data interaction of the front and back background Json is realized by using the ResponseBody and RequestBody annotations. The environment parameters of the ship cargo hold are updated with Ajax technology without reloading the whole page. Thus making system application access development.APP uses MUI open source framework design to transmit data by accessing server address and port number to reduce the complexity of the program.

The system uses multiple node designs in the collection of cargo tank environmental parameters, and the nodes are placed in the cabin according to the actual conditions of the goods transported by the ship. Each acquisition node is composed of Arduino MCU plus sensor group.

The Arduino core processor is ATmega328. It has 14 digital input/output ports (6 of which can be used as PWM output) and 6 analog inputs. It has good performance in information collection . The sensor group is composed of DHT22 temperature and humidity sensor, MQ-7 carbon monoxide detection sensor and MH-Z19 carbon dioxide detection sensor.

The physical control layer is an important part of the ship cargo compartment environment measurement and control system. Its own stable and accurate operation response plays a vital role in the normal operation of the system. Because the working voltage of the ventilation motor in the cargo compartment of the ship is large, it is very dangerous to directly control the switch of the ventilation motor by the single-chip microcomputer. Therefore, the system uses the optocoupler relay to control the suction of the AC contactor to achieve the effect of indirect control.

After the shipping company or the ship's staff sets the upper and lower limits of the cargo compartment environmental parameters, when the cargo compartment environmental parameters exceed the set value, the output signal of the microcontroller is output to the optocoupler relay. The conduction of the field effect tube is controlled through the reception of light and photoelectric elements radiated by its internal light-emitting diodes, and then the AC contactor is triggered to indirectly control the operation of the cargo hold fan equipment.

In this paper, the Socket interface is integrated into the Web development framework, and a kind of ship cargo hold monitoring and control system based on the SSM framework is constructed. The system enables the ship staff and the ship company to check the environment parameters of the cargo cabin in real time, and can control the environmental parameters in the cargo cabin by setting the upper and lower limits of the parameters, thus increasing the safety of the ship in shipping and reducing the loss in the process of cargo transportation. Thereby increasing the safety of the ship in shipping and reducing the loss of goods in the transportation process.In the future research, we can introduce the expert system or the modeling based on the historical data, use the fuzzy algorithm to carry on the intelligent control. The application layer development introduces the technology of load balancing and distributed processing high concurrency to solve the problem of large access to the system.

篇一：

## 附录 B 外文翻译—译文部分

目前，大学生获取各种学习资源缺乏系统、全面的资源共享服务平台，导致了学习资源利用率低、资源重复率高的现象。本文的最终目标是设计一个校园云资源共享服务平台来解决这个问题。该平台将通过零散的学习资源整合，使大学生更方便、更全面地获取学习资源。通过校园资源共享系统，我们可以随时随地在网上搜索我们需要的学习资料。我们也可以通过区域讨论的方式来讨论问题。资源可以分为显式资源和隐式资源。显式资源是普通的上传资源，用户可以直接搜索到。通过讨论和评估可以获得无形的资源，更方便用户学习。

资源共享系统采用 MVC 模式，可以将页面显示、域逻辑和数据库访问分离，方便开发人员对系统进行开发和维护，以及对系统进行二次开发。在进行框架选择时，我们比较了常用的 MVC 框架、SSH 框架和 SSM 框架。最后，我们选择了 SSM 集成框架(三种开源框架的集成，Spring MVC + Spring + Mybatis，简称 SSM)。SSM 集成框架非常方便，学校开发了强大的 Web 应用程序。同时，它也是一个典型的 MVC 框架。它将整个系统分为四层:表示层(JSP 和相关的 JS)、控制层、服务层和数据库访问层。MyBatis 是一个优秀的数据持久性框架。它支持自定义 SQL、存储过程和高级映射。MyBatis 框架具有 Hibernate 和JDBC 的优点。程序员可以自己编写 SQL 语句来促进 SQL 语句的优化，同时避免几乎所有的 JDBC 代码，并手动设置参数和获取结果集。

系统的总体设计包括五个模块:用户管理、角色管理、资源管理、评论管理和分类管理。用户管理模块包括:用户注册、用户信息修改、用户基本信息查询。后台管理员具有添加用户、查询所有用户和管理用户状态的功能。

角色管理模块包括:权限管理模块(仅对管理员开放)、角色的添加、删除、修改和检查、角色的授权。分类管理模块包括:添加、删除和修改分类。资源管理模块包括:学习资源上传、下载、描述、资源排序优化、资源分类。评论管理模块包括:资源评论、话题讨论、评论回复、评论评论。

资源共享平台的操作对象包括资源、资源类别、评论等。分析将平台抽象为五个实体: 用户、角色、资源、评论和分类。实体之间的具体关系如下(仅枚举):一个用户可以发布多个资源，一个资源只由一个人发布，并且是一对多关系。一个学习资源可以由多个用户操作，一个用户可以操作多个具有多对多关系的资源。一个分类只能属于一个大的分类，一个大的分类可以有多个小的分类，关系是一对多的。

资源共享系统采用 B/S 三层模式，由前端和后台两部分组成。前端部分采用了 Ace 框架

+BootStrap+JQuery+AJAX+CSS、JS 封装等技术。背景主要是 SpringMVC+Mybatis +Spring

和 Shiro。

根据资源共享服务系统的基本框架，持久化层是 MyBatis 的编译。Mybatis 框架分为

DAO 和 SQLMap 两个不同的组件，其中 DAO 组件抽象了数据访问层和应用程序持久层的

表示位置和模式。

DAO 组件是存在于业务逻辑和数据库资源之间的数据访问接口，用于建立与数据库的连接。

在资源共享系统中，持久层中的另一个组件，即 SQL Map 组件，通过配置的 XML 文件执行 SQL 语句和 Java 对象返回。配置文件包括基本配置和自定义配置的 mybatisi

-config.xml 文件。

控制层进行数据采集和处理的过程，主要包括控制器和服务两大类。控制器类发出一个请求来获取各种类型的数据和前端用户的请求数据。控制器类方法的命名和请求路径是唯一的。

控制层中还有一个业务逻辑层，用于实现数据的逻辑处理。它分为逻辑接口类管理器和实现类管理器。

表现层主要实现两个功能，一是实现用户数据输入，二是显示用户请求的数据。在前面的 Jsp 页面中，数据是通过 CSS 样式、JQuery、ajax 等基本方法来显示的。以文件上传为例，文件上传、查询、删除、修改等都会出现。实现方法是编写完整的CSS 前端框架， JQuery 实现部分数据处理和验证，ajax 实现数据的异步更新。

本文旨在设计一个“校园云”资源共享服务平台，提高大学生获取学习资源的效率。本文通过对 SSM 框架的研究，确定了利用 SSM 框架开发校园资源共享系统。详细描述了整个系统从数据库设计到整个框架的数据持久层、控制层和表示层的开发过程。最后，该平台将实现用户管理、校园资源管理、校园资源分类管理、评论管理等功能，使大学生更方便、更全面地获取学习资源。

篇二：

系统分为应用层、货舱环境感知层、通风物理控制层三部分。在环境感知层中，由精密传感器组和微控制器组成的节点采集货舱内的环境信息，并将其传输给客户端。在物理控制层，单片机将信号传输给继电器模块，利用继电器的开关控制货舱通风系统。在应用层，构建了 Web 应用系统和服务器监控机制。客户接受套接字的连接请求,建立连接,数据解析、存储和其他技术,船公司和船上人员可以通过电脑访问服务器和应用终端,船公司和船人员可以继续关注船舶货舱环境参数和通风系统的控制。

本系统的服务器开发采用 SSM 框架设计。作为企业应用开发框架，该系统在应用服务和数据处理方面有较好的性能，但在服务器启动过程中需要建立 Socket 服务，并始终在船上监控客户端的请求。连接并接收它发送的套接字包。在 SSM 框架中，Listener 是一个Servlet 侦听器，它侦听实例对象的方法调用或属性更改。当侦听器对象具有上述事件时， 将立即执行侦听器方法。客户机配置网络参数后，将 TCP 连接设置到服务器，因此侦听客户 机 的 侦 听 器 将 在 服 务 器 端 进 行 融 合 。 并 添 加 配 置 文 件 < listener-class > com.hu.tem .socket1。SSM 框架的 web.xml 文件中的 ServerSocketListener ，然后将套接字

侦听器合并到 SSM 框架中。在此期间，服务器在接收到每个客户端连接请求后将处理流放入单独的线程中，然后等待下一个客户端连接请求，从而消除了服务器端接收请求的阻塞。实现向服务器发送多个 ship 环境参数。

系统采用移动 WebAPP 前端框架设计发展的用户应用程序。多元化的开发方法,包括HTML、CSS 和 JavaScript 技术使应用具有更好的交互性,一次性开发的优势和多平台应用程序可以消除项目移植和开发的难度,开发周期长的问题;梅 UI 层,DCloud 中间件层,和Android 和 IOS 终端系统层,移动网络应用是运行缓慢和不能比较本地应用的经验。梅不封装Dom 操作,不需要JavaScript 运行时解析HTML 标记,减少移动资源的消耗,并加快页面加载速度,使系统在船舶货舱检测与环境参数阈值页面设置之间切换更加顺畅，在访问服务器货舱环境数据时速度更加稳定。

系统实现：

船舶货舱环境测控系统在应用层开发中采用了 SSM 框架技术，具有较好的访问性能和快速的开发能力，是 Web 应用开发框架的首选组合。本系统在访问存储上采用 MVC 软件架构模型。如图 3 所示，该模型将数据的控制层与表示层分离开来，便于程序的后续维护和扩展，并使一些程序的重用成为可能，从而提高了开发的效率。在 SSM 框架下，通过建立输入和输出流将数据流读写到连接的客户端，并使用 ResponseBody 和 RequestBody 注解实现前后后台 Json 的数据交互。利用 Ajax 技术更新船舶货舱环境参数，无需重新加载整个页面。从而进行系统应用程序访问开发。APP 采用 MUI 开源框架设计，通过访问服务器地址和端口号来传输数据，降低了程序的复杂度。

本系统采用多节点设计来采集货舱环境参数，并根据船舶所运货物的实际情况将节点布置在舱内。每个采集节点由 Arduino MCU +传感器组构成。Arduino 的核心处理器是ATmega328。它有 14 个数字输入/输出端口(其中 6 个可以用作PWM 输出)和 6 个模拟输入。具有良好的信息采集性能。传感器组由 DHT22 温湿度传感器、MQ-7 一氧化碳检测传感器和 MH-Z19 二氧化碳检测传感器组成。

物理控制层是船舶货舱环境测控系统的重要组成部分。其自身稳定准确的运行响应对系统的正常运行起着至关重要的作用。由于船舶货舱通风电机的工作电压较大，用单片机直接控制通风电机的开关是非常危险的。因此，系统采用光耦继电器控制交流接触器的吸力，达到间接控制的效果。

船舶公司或船舶工作人员设定货舱环境参数的上下限后，当货舱环境参数超过设定值时，单片机的输出信号输出到光耦继电器。现场效应管的传导是通过接收其内部发光二极管辐射的光和光电元件来控制的，然后触发交流接触器来间接控制货舱风机设备的运行。本文将 Socket 接口集成到 Web 开发框架中，构建了一种基于 SSM 框架的船舶货舱监

控系统。系统使船人员和货物的船公司检查环境参数实时小屋,和可以控制环境参数的货运舱通过设置参数的上限和下限,从而增加船的安全在运输和货物运输的过程中减少损失。从而增加船舶在运输中的安全性，减少货物在运输过程中的损失。在今后的研究中，可以引

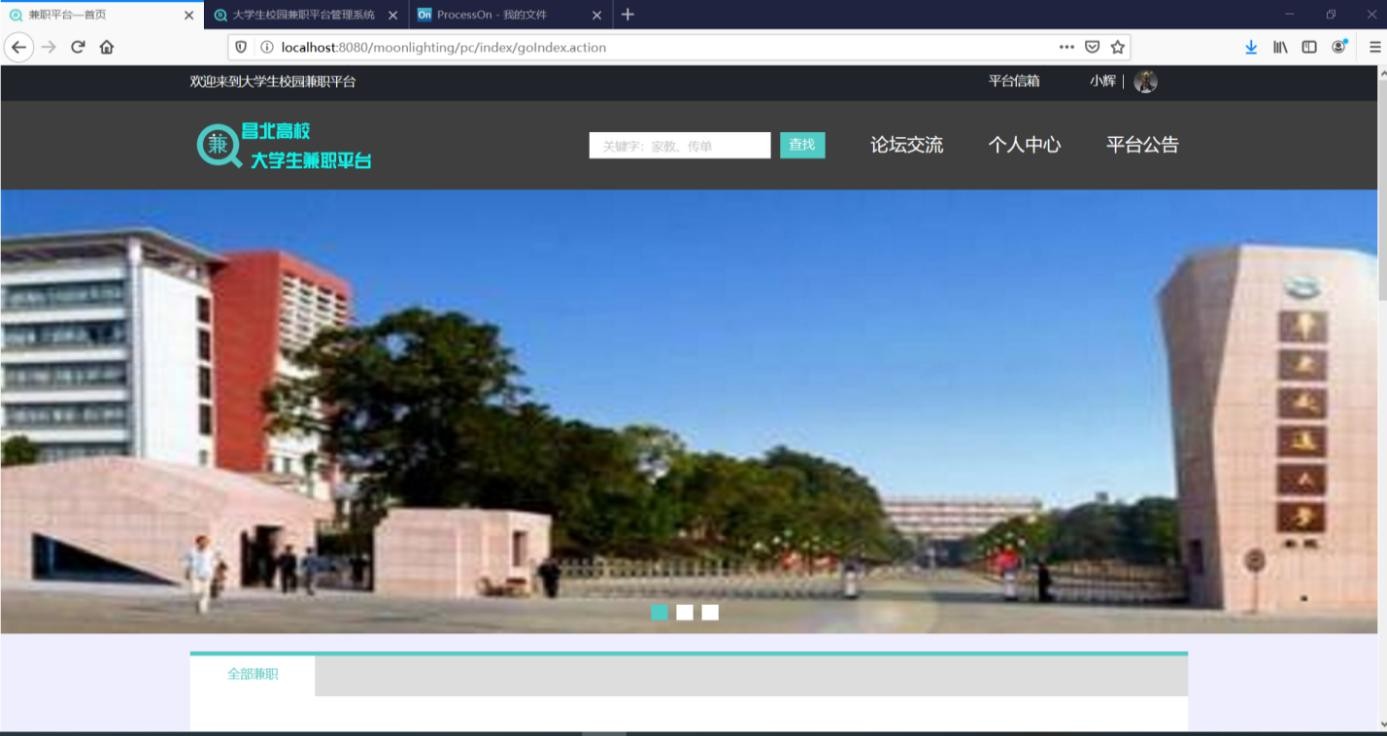
入专家系统或基于历史数据的建模，利用模糊算法进行智能控制。应用层开发引入负载均衡和分布式处理高并发技术，解决了系统的大访问问题。

前后台访问地址

## 附录 C 软件使用说明书

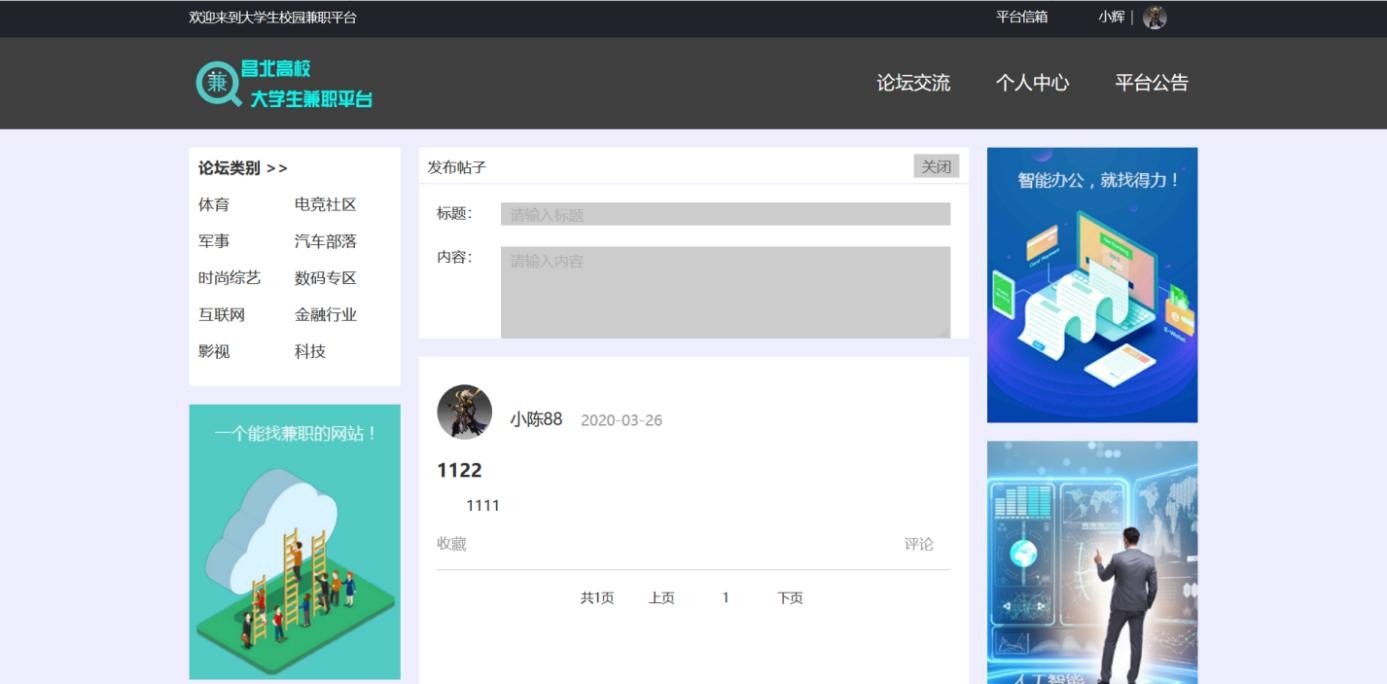
前台：http://localhost:8080/moonlighting/pc/index/goIndex.action

后台：http://localhost:8080/moonlighting/manager/login/goServerLogin.action

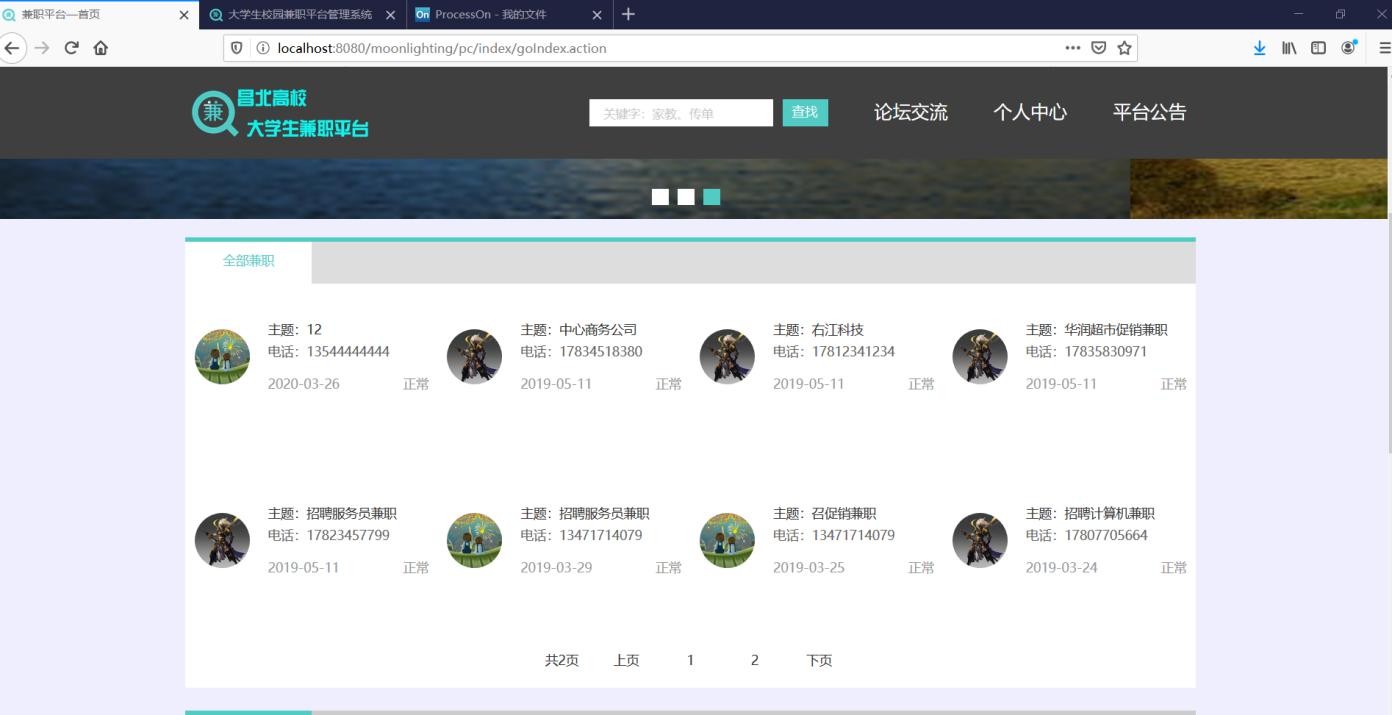
用户或者游客在通过我们的链接访问到我们的前台页面后可对我们的主页进行访问如下图所示：

图附录 C-1 前台页面

查看兼职信息和浏览论坛如下图所示：

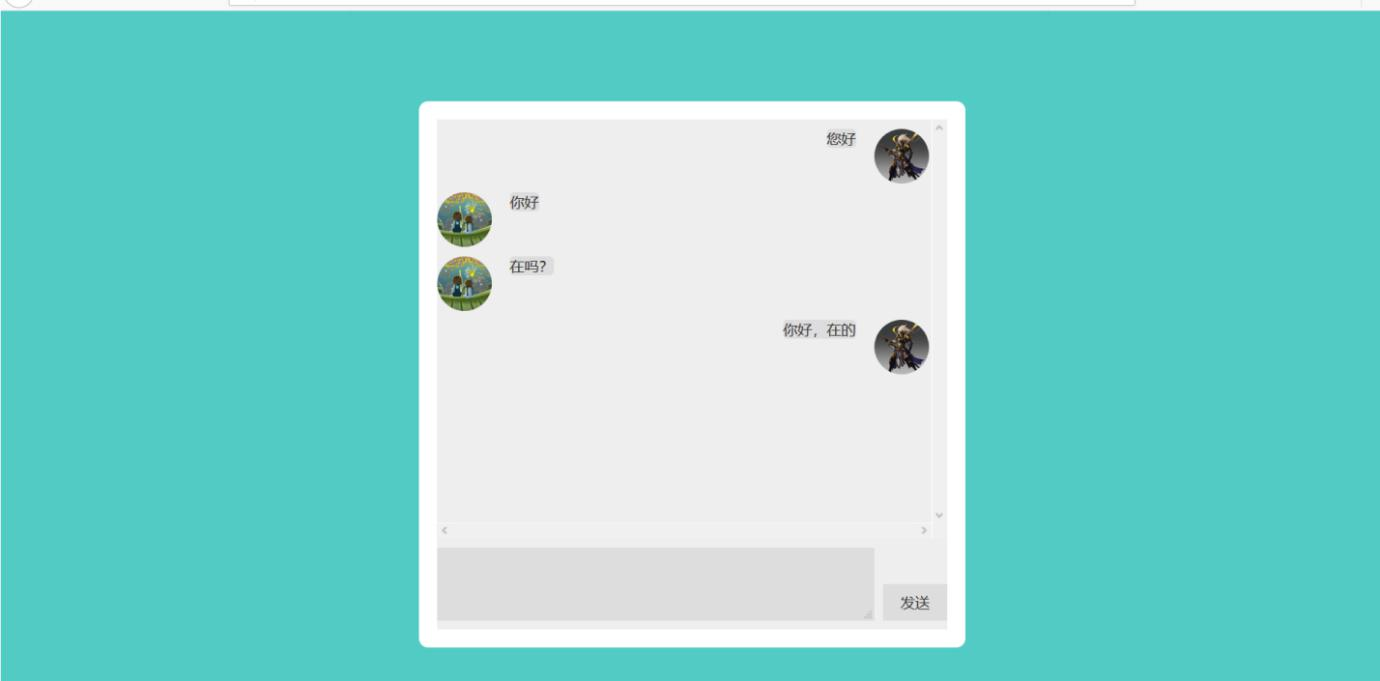


图附录 C-2 兼职信息页面



图附录 C-3 浏览论坛页面

用户有资格进行与老板交流和评论帖子：



图附录 C-4 与老板交流页面

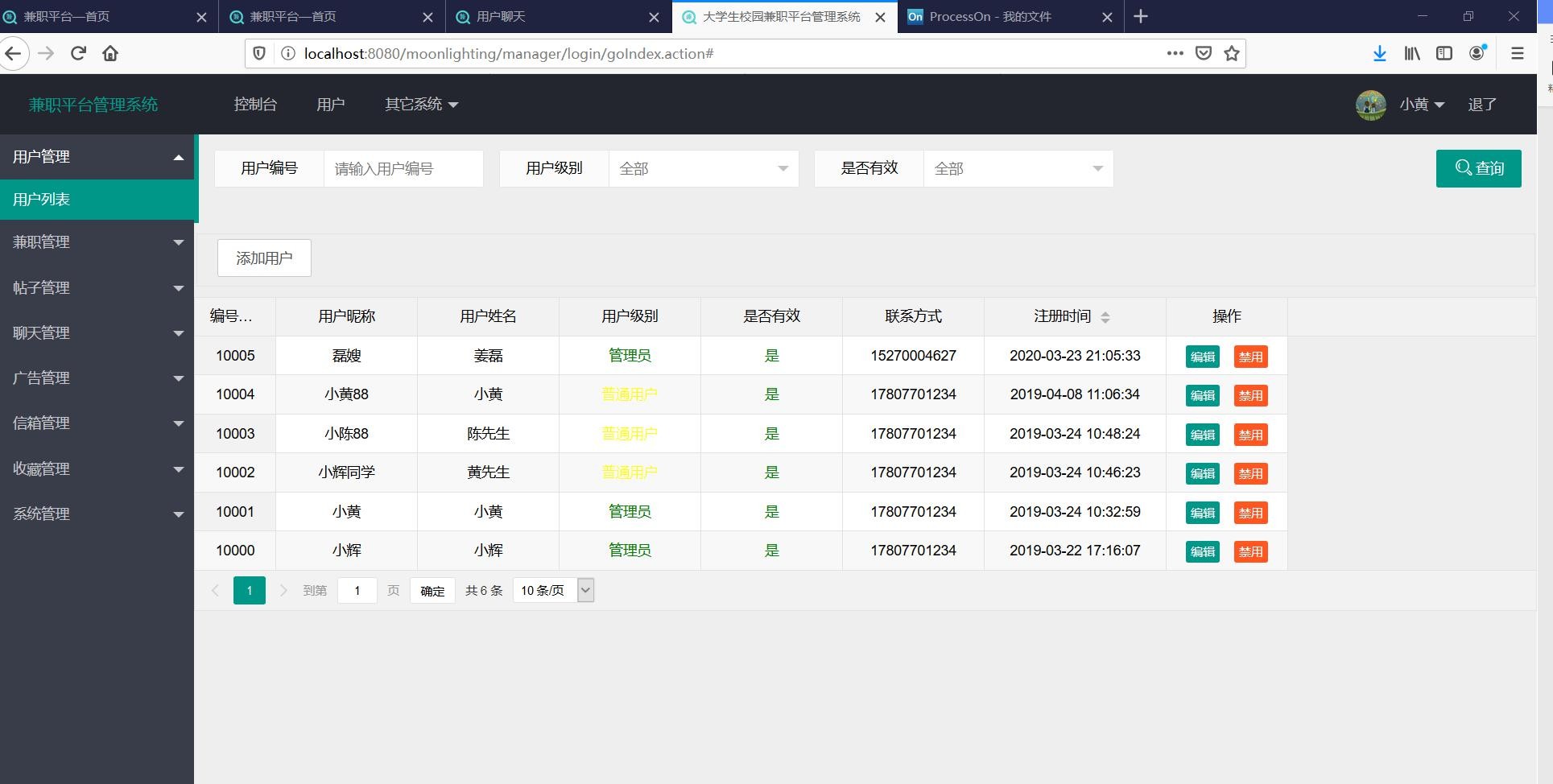


图附录 C-5 帖子评论页面

个人中心操作如下图所示如下图所示：



图附录 C-6 个人中心操作页面

管理员使用管理员账号登录后台管理系统如下图所示：

图附录 C-7 后台管理页面

## 附录 D 主要源代码

public class BaseUtil extends WebMvcConfigurerAdapter {

/\*\*

\* Logger for this class

\*/

private static final Logger logger = Logger.getLogger(BaseUtil.class);

public void writeJson(HttpServletResponse response, Object object) { try {

String json = JSON.toJSONStringWithDateFormat(object, "yyyy-MM-dd HH:mm:ss");

response.setContentType("text/html;charset=utf-8"); response.getWriter().write(json); response.getWriter().flush();

} catch (IOException e) { logger.error("返回流失败");

} finally {

try {

response.getWriter().close();

} catch (IOException e) {

logger.error("关闭输出流失败");

}

}

}

protected void output(HttpServletResponse response, String jsonResult) { try {

response.setContentType("application/json; charset=UTF-8"); response.setHeader("Access-Control-Allow-Origin", "\*"); response.setHeader("Access-Control-Allow-Methods", "POST,GET,OPTIONS"); response.setHeader("Access-Control-Allow-Credentials", "true"); response.setContentLength(jsonResult.getBytes("UTF-8").length);

logger.debug("response json: " + jsonResult);

// response.getOutputStream().write(jsonResult.getBytes());

// printStream.println(result);

PrintWriter printWriter = response.getWriter(); printWriter.write(jsonResult); printWriter.flush();

// response.getOutputStream().

} catch (Exception e) {

logger.error("Error output json data to the client!!!orginal json=" + jsonResult, e);

}

}

protected void writeString(String str) {

HttpServletResponse response = ((ServletWebRequest) RequestContextHolder.getRequestAttributes()).getResponse();

response.setContentType("application/json; charset=UTF-8"); PrintWriter printWriter;

try {

printWriter = response.getWriter(); printWriter.write(str); printWriter.flush();

} catch (IOException e) { e.printStackTrace();

}

}

private Map<String, Object> session;

public Map<String, Object> getSession() { return session;

}

public void setSession(Map<String, Object> arg0) { session = arg0;

}

}

//密码

public class MD5Util {

private static String byteArrayToHexString(byte b[]) { StringBuffer resultSb = new StringBuffer();

for (int i = 0; i < b.length; i++) resultSb.append(byteToHexString(b[i]));

return resultSb.toString();

}

private static String byteToHexString(byte b) { int n = b;

if (n < 0)

n += 256;

int d1 = n / 16; int d2 = n % 16;

return hexDigits[d1] + hexDigits[d2];

}

public static String MD5Encode(String origin, String charsetname) { String resultString = null;

try {

resultString = new String(origin);

MessageDigest md = MessageDigest.getInstance("MD5"); if (charsetname == null || "".equals(charsetname))

resultString = byteArrayToHexString(md.digest(resultString

.getBytes()));

else

resultString = byteArrayToHexString(md.digest(resultString

.getBytes(charsetname)));

} catch (Exception exception) {

}

return resultString;

}

/\*\*

* 获得 16 个长度的十六进制的 UUID

\*

* @return UUID

\*/

public static String get16UUID() { UUID id = UUID.randomUUID(); String[] idd = id.toString().split("-"); return idd[0] + idd[1] + idd[2];

}

private static final String hexDigits[] = { "0", "1", "2", "3", "4", "5",

"6", "7", "8", "9", "a", "b", "c", "d", "e", "f" };