

毕业设计说明书

**基于SSM的游戏论坛的设计与开发**

二零二零年 五月 十六日

学位论文原创性声明

本人所提交的学位论文基于SSM的游戏论坛的设计与开发，是在导师的指导下，独立进行研究工作所取得的原创性成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中标明。

本声明的法律后果由本人承担。

学位论文版权使用授权书

本学位论文作者完全了解河北农业大学有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权河北农业大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

保密的学位论文在\_\_\_\_\_\_\_年解密后适用本授权书。

**摘 要**

随着互联网飞速的发展，Internet已经成为了生活必须品。互联网大大缩短了人们交流的距离，在网络中视频、聊天、发布个人博客等方式已经成为了人们互相之间沟通、交流信息的最主要方式。除此之外，为了方便某一个圈子，或者某一个领域的人们探讨问题、交流意见，Internet上出现了论坛服务BBS。在线论坛可以非常方便的让人们对某一领域发表自己遇到的问题，即发表主题、发帖，随后，论坛上的其他人员会根据自己的经验、见解回答问题、发表意见、或者提出解决思路。

开发本论坛系统的目的是提供一个供ark-方舟生存进化游戏用户，交流经验，提出问题，发表看法的网上社区。故而，此论坛系统的基本功能首先是发布帖子，其次是其他人可以随意浏览不同分板块中的帖子。论坛系统还需要提供注册登录服务，在用户注册系统成为系统会员之后，还可以对帖子发布评论信息，对评论信息进行回复。没有登录的用户（游客）只能浏览信息。

本系统主要使用了Java语言，后台采用zookeeper+dubbo实现分布式部署，前端使用LayUI轻量级框架开发。运用MySQL数据库，构建了一个B/S架构体系的小型游戏论坛服务系统。由于使用了前后端分离设计思想，此系统具有操作灵活、反应灵敏、安全稳定、运行可靠的特点。

**关键词：**游戏论坛；bbs；B/S模式；论坛

**Abstract**

With the rapid development of the Internet, the Internet has become a necessity of life. The Internet has greatly shortened the distance for people to communicate. Video, chatting on the Internet, and posting personal blogs have become the most important way for people to communicate and exchange information with each other. In addition, in order to facilitate people in a certain circle or a certain field to discuss problems or exchange opinions, a forum service BBS has appeared on the Internet. Online forums are very convenient for people to post their own problems in a certain field, that is, to post topics and post. Later, other people on the forum will answer questions, express opinions, or propose solutions based on their own experience and insights.

The purpose of developing this forum system is to provide an online community for ark-ark survival evolution game users to exchange experiences, ask questions, and express opinions. Therefore, the basic function of this forum system is to publish posts first, followed by others to browse posts in different sub-sections at will. The forum system also needs to provide registration and login services. After the user registration system becomes a system member, it can also post comment information on the post and reply to the comment information. Users (guests) who are not logged in can only browse the information.

This system mainly uses Java language, the background uses zookeeper + dubbo to achieve distributed deployment, and the front end uses LayUI lightweight framework for development. Using the MySQL database, a small game forum service system with a B / S architecture system was constructed. Due to the use of front-end and back-end separation design ideas, this system has the characteristics of flexible operation, sensitive reaction, safety and stability, and reliable operation.

**Keywords:** game forum; bbs; B / S mode; forum

目 录

[1 绪论 1](#_Toc40900972)

[1.1 系统开发背景 1](#_Toc40900973)

[1.2 系统开发意义 1](#_Toc40900974)

[1.3 系统基本设计概念 1](#_Toc40900975)

[2 相关技术介绍 2](#_Toc40900976)

[2.1 分布式系统架构 2](#_Toc40900977)

[2.2 Servlet技术 2](#_Toc40900978)

[2.3 SSM框架 3](#_Toc40900979)

[2.4 LayUILayUI前端框架 3](#_Toc40900980)

[2.5 MySQL技术 3](#_Toc40900981)

[2.6 Redis内存数据库 3](#_Toc40900982)

[**2.7** zookeeper+dubbo远程服务调用 4](#_Toc40900983)

[**2.8** vsftpd文件管理 4](#_Toc40900984)

[**2.9** nginx反向代理技术 4](#_Toc40900985)

[2.10 Maven管理工具 4](#_Toc40900986)

[3 需求分析 5](#_Toc40900987)

[3.1 业务需求 5](#_Toc40900988)

[3.2 用户需求 5](#_Toc40900989)

[3.3 功能需求 6](#_Toc40900990)

[3.4 性能需求 7](#_Toc40900991)

[3.5 可扩展性需求 8](#_Toc40900992)

[3.6 系统开发环境 8](#_Toc40900993)

[4 系统分析与设计 8](#_Toc40900994)

[4.1 系统组织结构 8](#_Toc40900995)

[4.2 系统主要操作业务流程图 10](#_Toc40900996)

[4.2.1 后台管理操作流程图 10](#_Toc40900997)

[4.2.2 SSO登录模块业务流程图 11](#_Toc40900998)

[4.2.3 板块内容管理业务流程图 12](#_Toc40900999)

[4.2.4帖子发布业务流程图 13](#_Toc40901000)

[4.2.5用户收藏文章流程图 14](#_Toc40901001)

[4.3 数据库设计 14](#_Toc40901002)

[4.3.1 概念结构设计 14](#_Toc40901003)

[4.3.2 物理结构设计 19](#_Toc40901004)

[5 系统实现 23](#_Toc40901005)

[5.1 后台管理系统 23](#_Toc40901006)

[5.1.1后台管理系统的登录 23](#_Toc40901007)

[5.1.2 用户信息管理模块 24](#_Toc40901008)

[5.1.3 用户收藏管理模块 25](#_Toc40901009)

[5.1.4 帖子信息管理模块 26](#_Toc40901010)

[5.1.5 板块内容管理模块 27](#_Toc40901011)

[5.1.6 回帖管理管理模块 28](#_Toc40901012)

[5.1.7多级回复信息管理 28](#_Toc40901013)

[5.1.8 信息管理模块 29](#_Toc40901014)

[5.2 前台论坛系统 30](#_Toc40901015)

[5.2.1 论坛主页 30](#_Toc40901016)

[5.2.2 板块详情页 30](#_Toc40901017)

[5.2.3 文章内容展示页 31](#_Toc40901018)

[5.2.4 多级评论展示 32](#_Toc40901019)

[5.2.5用户中心页面 33](#_Toc40901020)

[5.2.6用户主页 34](#_Toc40901021)

[5.2.7用户信息修改页 35](#_Toc40901022)

[5.2.8站内消息显示页面 35](#_Toc40901023)

[结束语 36](#_Toc40901024)

[参考文献 37](#_Toc40901025)

[致谢 38](#_Toc40901026)

# 1 绪论

## 1.1 系统开发背景

随着Internet的飞速发展，互联网时代来临，人们交流的方式越来越方便，各种网络信息发布平台应运而出。20世纪90年代，中文论坛以惊人的速度发展起来，最为引人注目的是以民间力量主导的天涯论坛和以政府力量主导的人民网强国论坛。此二者均在20多年之后成为互联网重量级的意见表达渠道和对话平台。近年来，天涯网江河日下，网络论坛逐渐变为业余、小众人员的具有相同交流主题的信息交互服务。由于网络论坛的交流时空异步性，对于具有同一兴趣爱好或者同一圈子的人员之间的信息交互极为便利。以百度贴吧为例，每一个贴吧就是一个主题，每一个主题下聚拢了大量对此关注的人员。截至到2019年，贴吧用户仍居高不下，活跃用户高达3亿。由此可见，针对某一兴趣，以某一圈子为基点开发的论坛服务仍旧大有可为。

## 1.2 系统开发意义

近年来，网络游戏越来越成为人们放松和排解压力的娱乐方式。于此同时，游戏周边服务也变得越来越重要。一个完善的游戏交流平台，可以提高游戏受众的娱乐氛围，保证游戏的长久运行。不仅如此，当游戏受众遇到游戏中的问题时，或者有好的想法或者遭遇，一个分享的渠道必不可少。游戏论坛对于以上问题就具有重要意义，相关游戏论坛的出现可以更加方便的让游戏受众互动交流，解决在游戏中遇到的问题。因此，本系统的设计与实现有着很重要的现实意义。

## 1.3 系统基本设计概念

本论坛系统可分成六大模块，各模块又有着自己的子模块。其中，各子模块并不是完全相互独立的，彼此间有着少量的联系，使得整个系统有良好的高内聚低耦合性，本系统使用mavan工具对模块进行整合。采用“瀑布模型”的开发模型，具体过程如图1-1所示。

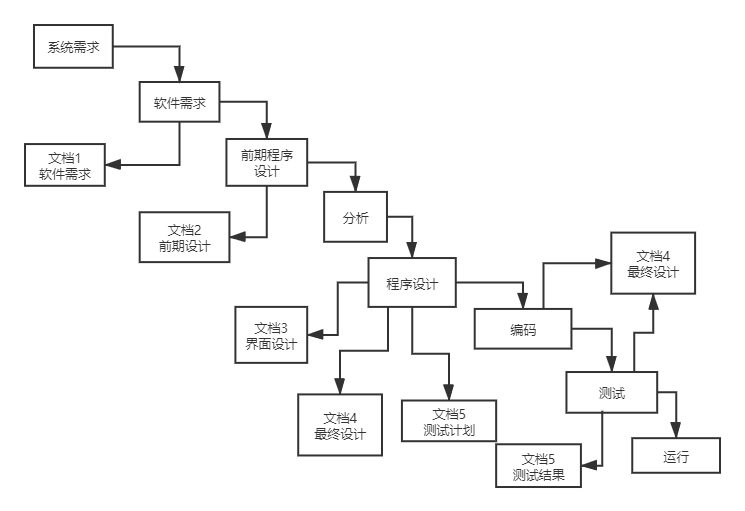


图1-1瀑布模型系统开发图

# 2 相关技术介绍

本系统基于B/S结构，采用分布式系统架构，以Java语言为主要编程语言，使用Servlet技术，结合SSM后台框架和LayUI轻量级前端框架，在idea开发平台上操作MySQL数据库相关数据来进行系统研发。使用Redis内存数据库进行数据缓存和session共享，使用zookeeper+dubbo技术实现远程服务调用，vsftpd+nginx实现远程文件上传，开发过程中使用Manav管理工具进行依赖管理。

## 2.1 分布式系统架构

分布式就是一个业务拆分成多个子业务，部署在不同的服务器上。

例如：电商系统可以拆分成商品，订单，用户等子系统。这就是分布式。对于单应用架构来说，随着用户越来越多，数据量越来越大，服务器负载会越来越高。为了提高并发量，将同一个服务发布在不同的系统上，做成一个集群，这就构成了分布式的系统架构。此架构更好的利用了系统的性能，提高了容错率和并发访问。

## 2.2 Servlet技术

Servlet技术：Servlet其实就是Java中的一个类。确切的说，Servlet是一个容器对象，当浏览器输入一个URL的时候，会向服务器发送一个http协议，Servlet技术会封装http协议内容成为一个HttpServletRequst请求对象，将此对象传入Servlet容器。开发者在Servlet容器中处理这个请求，处理完毕会返回一个HttpServletResponse响应对象给服务器，服务器解析响应对象成为http请求响应给浏览器。

## 2.3 SSM框架

SSM框架是三种技术的集合，SpringMVC框架、Spring框架、Mybatis框架。其中SpringMVC是Spring框架中的一部分，SpringMVC主要负责处理用户请求，他的核心也是一个Servlet即DispatcherServlet。相当于传统MVC架构中的Controller控制层。Spring可以看作是一个容器技术，服务器中所有对象分离开来，又像粘合剂一样整合在一起，他的核心理念是控制反转，通过配置文件将项目中的bean对象注入到需要的地方，不需要程序员去new对象。Mybatis技术是对JDBC的封装，他让底层数据库操作变得透明，它通过sqlSessionFactory拿到一个sqlSession对象，使用sqlSession对象和数据库交互。

## 2.4 LayUI前端框架

LayUI是一个开箱即用的轻量级前端框架，由专业前端人员开发，面向的是全层次的前后端开发者，低门槛，简而全是他的特点。LayUI采用自身模块规范编写，遵循原生HTML/CSS/JS的书写规范，所以门槛低，拿来即用。此框架本身体积小巧、轻盈，但是内容丰富、饱满，组件丰富，并且社区活跃度极高。LayUI包含了常见的网页解决方案，即使有一些有瑕疵的地方，其社区内也可以找到解决方案。

## 2.5 MySQL技术

MySql最初是由mysql AB公司开发，后来被SUN公司收购，不久SUN公司又被Oricle公司收购。Mysql和Oracle数据库、Sql Server数据库一样，都是关系型数据库。他的数据主要以文件形式保存在磁盘中，不同于内存数据库，关系型数据库最大的特点是具有事务的一致性。MySql支持sql规范，可用于非常复杂的查询。管理Mysql的时候可用mysql命令管理，不过更多的是用桌面管理工具，界面简单方便。此项目使用Navicat for mysql桌面管理工具，配合idea中的数据库功能，大幅度提高开发效率。两个工具结合使用，可解决使用mysql过程中遇到的大部分需求。查询的时候会以表格形式显示在管理界面中。Idea的数据库功能在写sql语句时，会有非常智能的提示。

## 2.6 Redis内存数据库

Redis是一个开源的内存数据库。它不仅仅支持简单的key-value形式的数据保存，还提供了list，set，zset，hash等复杂的数据结构存储功能。虽然Redis是一个内存型数据库，但是通过配置，也是可以通过日志持久化存储的。内存型数据库因为其数据保存在内存当中，而内存又具有高速读写的特点，所以内存数据库中的数据均可进行高速读写，性能极高。在进行系统设计的时候，通常用来缓存从关系型数据库中读取到的数据，在极少量的修改场景下，直接从内存型数据库中读取数据，大幅度的提高了系统性能。本系统使用Reids数据库实现分布式开发中的Session共享功能。Redis还支持集群化部署，最小可部署三主三从的一个小型集群，大大提高了容错率。

## **2.7** zookeeper+dubbo远程服务调用

在分布式开发的系统中，经常出现各种各样的难题，很大一部分都是因为缺少一个协调机制，zookeeper就是提供这个协调机制的服务。zookeeper通常用来解决分布式服务中的数据管理问题。zookeeper提供的功能主要包括：配置维护、统一命名服务、状态同步、集群管理等。Dubbo是阿里巴巴提供的一个优秀的开源服务框架。在分布式开发系统中，不可避免的涉及到了RPC（Remote Procedure Call）--远程过程调用，它可以通过网络从远程服务器上获取使用到的服务，而这个发布服务和提供远程调用的服务框架就是dubbo。通过Dubbo可将服务发布在远程服务器的zookeeper注册中心中，zookeeper协调管理远程请求，通过Dubbo响应远程请求的服务，这个过程就实现了RPC远程过程调用。

## **2.8** vsftpd文件管理

无论什么系统，上传文件功能必不可少。本系统采用ftp服务上传文件。vsftpd是very secure FTP daemon的缩写，主要用于在Internet中控制文件的传输。他的特点是安全，可以运行在linux系统上，而且是一个免费的开源软件，并且他还支持许多其他ftp服务器不支持的特征。

## **2.9** nginx反向代理技术

nginx是一个网络资源反向代理服务器，使用nginx，可以将主机上的网络资源，反向代理到置顶的域名之上。本系统是由vsftpd上传文件，使用nginx反向代理上传文件的路径，是文件可通过一个url路径就能访问到。

## 2.10 Maven管理工具

Maven是基于项目对象模型来管理项目构建的软件项目管理工具。是一款强大的自动化构建工具。通过maven，可以更加方便的进行项目的编译、测试、运行和打包。不仅如此，maven统一帮助我们管理第三方依赖，自动导入项目依赖。

# 3 需求分析

## 3.1 业务需求

本系统的业务需求如下：

1、提供一个供方舟生存进化游戏受众交流的平台。

2、能够安全有效的处理多个用户交流问题。

3、解决游戏受众交流困难，有问题没有地方申诉的痛点。

4、及时的消息提示系统，用户可以在最短的时间内得到问题的回答或者解决思路。

5、在系统出问题的时候，可以最快进行报修处理。

6、保障用户隐私。

## 3.2 用户需求

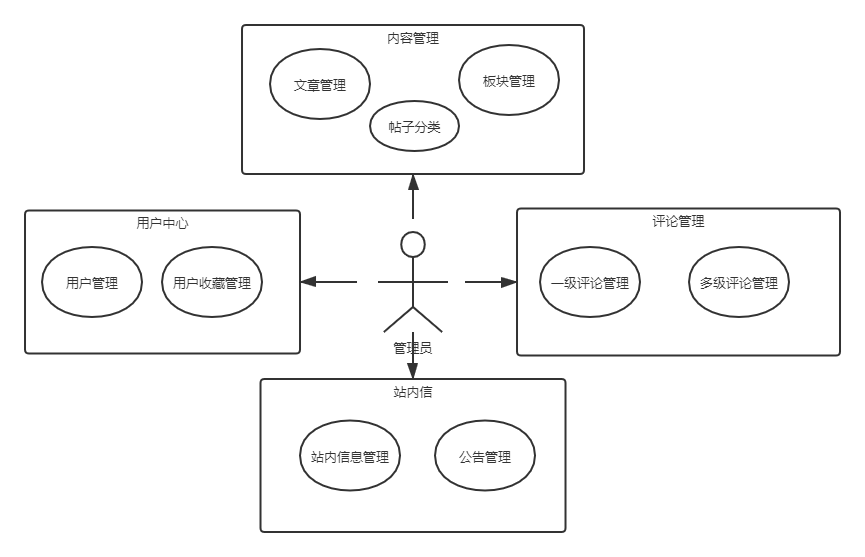
1、管理员应具有最高权限，可登录后台系统和前台系统。可在后台系统中对板块、帖子、分类、信息、用户、帖子收藏、多级评论、站内信息进行管理。

图3-1系统管理员角色权限图

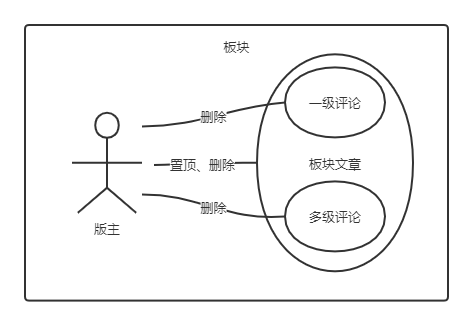
2、版主：可对板块中的文章进行置顶、取消置顶、删除、删除其文章的一级评论和多级评论。同时版主角色属于系统会员，会员具有的权限，版主均具有。

图3-2版主权限图

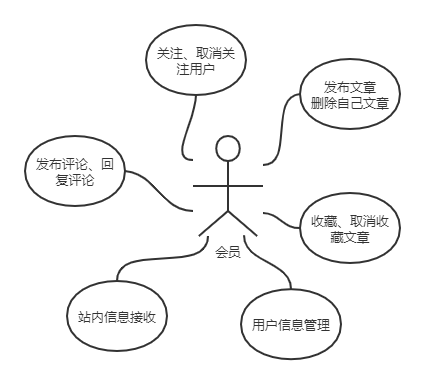
3、系统会员：浏览帖子、发布帖子、发布帖子时添加帖子分类、可将帖子添加到指定板块（非公告）、关注/取关其他用户、评论文章信息、回复评论信息、接收站内信。

图3-3会员权限图

4、游客:游客不需要登录，系统中无账户。只可进行浏览帖子，查看公告。

## 3.3 功能需求

针对论坛系统各部分需求，本系统主要具有以下功能。

（1）用户管理：主要实现对用户的信息管理，用户可否使用、是否是管理员或版主、会员、更改用户信息、删除用户信息。此功能主要提供管理员使用。同时会员需要提供更改自己用户信息的功能。

（2）用户收藏管理：管理用户收藏的文章，管理员可管理全员收藏，会员只可管理自己的收藏信息。

（3）文章管理：添加或删除修改一个文章信息。管理员可管理所有的文章信息，查询指定板块内容。版主只可管理自己板块内的文章信息（公告板块专属于管理员）。会员可发布自己的文章，所有用户（包括游客）均可浏览文章。

（4）分类管理：分类是对于文章来说，每一篇文章均有自己对应的一个或者多个分类标签。除游客以外，所有用户在添加文章的是由，均可自定义新的分类标签。

（5）板块管理：添加或者删除一个板块，一个板块是多篇文章的主题分类，管理员可管理板块的添加删除更改，可设定此板块的版主信息，卡查询指定板块内容。

（6）回帖管理：系统会员可在一个文章下进行回帖评论。版主可删除对应板块内的回帖信息。管理员可对回帖信息删除修改。

（7）回复管理：对于一个回帖，系统会员可以对回帖进行评论，即对回帖的回复信息。版主可删除回复信息。管理员可对回复修改和删除，查询指定回复内容。

（8）站内信息管理：当用户评论或回复一篇文章、关注某一个用户，对应的用户会接收到站内信息，提示某个用户对自己评论，或者关注了自己。

（9）单点登录功能：当一个用户在前台登录的时候，如果是管理员，可自动登录后台系统，不需要重复登录。

（10）热度文章导航：在主页中，需要有热度导航功能，站内浏览量比较高得文章需要放在瞩目得地方，方便用户访问。

（11）热评文章导航：站内信息评论比较多得文章，需要做导航功能，使用户便于访问。

（11）文章发布图片上传及表情展示：用户在发布一篇文章得时候，能够上传文件信息，可以在文章内容中有emoji表情。

（12）板块内文章分页：一个板块内如果文章内容过多，需要添加分页功能，便于用户查看。

（13）回帖、回复流式加载：一篇文章如果评论内容过多，可手动选择是否加载新的评论内容。

## 3.4 性能需求

本系统主要提供给游戏受众使用，用户量大，所以要支持高并发访问，系统运行必须稳定，抗压，可靠运行。

## 3.5 可扩展性需求

随着系统得使用，可能会有新的需求，所以需要能够扩展新的模块。本系统采用分布式开发架构，不同的模块部署在不同得系统中。在添加新的模块时，只需要重新编写一个系统，调用远程发布的rpc服务即可，扩展功能十分方便。

## 3.6 系统开发环境

本系统的开发语言为Java语言，可以跨平台使用，本人系统为Windows10。一些其他的技术如Redis内存数据库等需要部署在Linux系统上。其详细开发环境如表3-1所示。

表3-1开发环境表

|  |  |
| --- | --- |
| 需求名称 | 详细要求 |
| 开发工具 | IntelliJ IDEA 2020.1 |
| 服务器软件 | Apache Tomcat/8.5.50 |
| 数据库  数据库桌面管理工具 | Mysql5.8  Navicat for mysql 12.1.25 |
| 服务器操作系统  Redis内存数据库  内存数据库桌面管理工具  Zookeeper+Dubbo  nginx+vsftpd  Linux远程连接工具 | Windows10 64位  CentOS Linux release 7.6.1810  RedisDesktopManager  CentOS Linux release 7.6.1810  CentOS Linux release 7.6.1810  XShell 6 |
| 开发语言 | JAVA8 |
| 浏览器 | Google Chrome 81.0.4044.138 |
| 开发方式 | 个人开发 |
| 平台 | J2EE |
| 类库  画图工具 | JDK1.8  ProsessON在线作图 |

# 4 系统分析与设计

这部分是系统分析和设计阶段，不涉及实现，所以建议还是分析根据调研和分析，需要设计哪些功能模块？每个模块实现什么功能，分析描写清楚即可。

## 4.1 系统组织结构

本游戏论坛系统包括四个子系统：rpc远程服务系统，前台系统，后台管理系统，SSO单点登录系统。

* rpc远程服务系统只有一个功能：发布远程服务。
* 后台管理系统包括四个模块：内容管理模块、用户中心模块、评论管理模块、站内信息管理模块。
* 前台系统包括六个模块：主页信息展示模块、板块信息展示模块、文章信息展示模块、个人信息展示模块、站内信息收取模块、个人信息更改模块。
* SSO单点登录系统只包含一个功能：即登录注册模块。
* 四个子系统全部依赖一个工具模块。

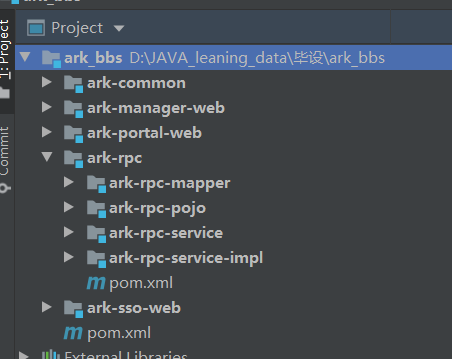
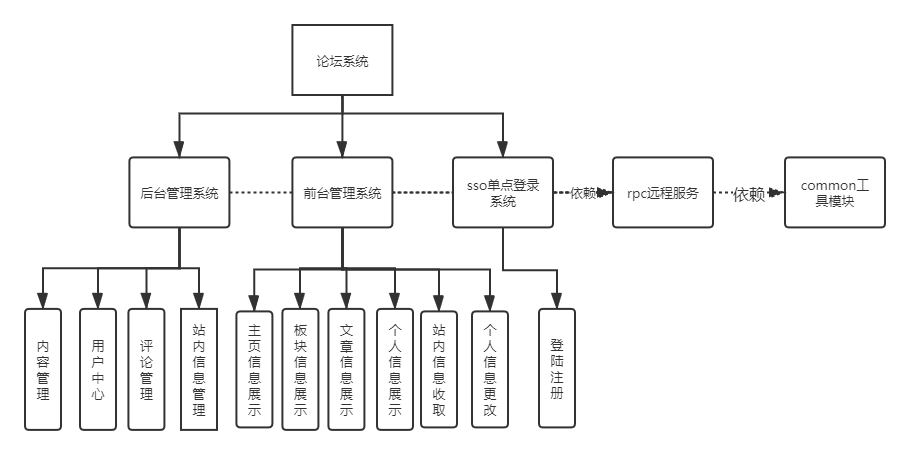
具体项目架构如图4-1所示：这一块是不应该放到实现中去，为实现系统功能模块，搭建的系统服务，那个对应实现相应的功能。放在这里感觉不太合适。这样的话4-2的图也需要调整一下

图4-1项目架构图示

* 模块介绍： ark\_bbs：parent项目，打包模式为pom，主要定义了各个系统依赖版本。
* ark-common：工具模块，打包模式为jar，包含了一些常用到的工具类，以及额外定义的实体类。
* ark-manager-web：后台管理系统，打包模式为war，此系统是一个web系统，发布在tomcat服务器上提供给管理员管理后台。
* ark-portal-web：前台系统，打包模式为war，同样是一个web系统，发布在tomcat服务器上提供前台访问服务。
* ark-rpc：rpc远程服务提供系统，包含四个小模块：
* ark-rpc-pojo：数据库对应的实体类。
* ark-rpc-mapper：MyBatis框架操作数据库的持久层。
* ark-rpc-service：供远程服务调用的服务接口API。
* ark-rpc-service-impl：远程调用的服务具体实现类。
* ark-sso-web：单点登录系统，打包模式为web。此系统也是一个web系统，主要提供注册登录服务。

具体项目功能结构如图4-2所示。

图4-2项目功能结构图

## 4.2 系统主要操作业务流程图

### 4.2.1 后台管理操作流程图

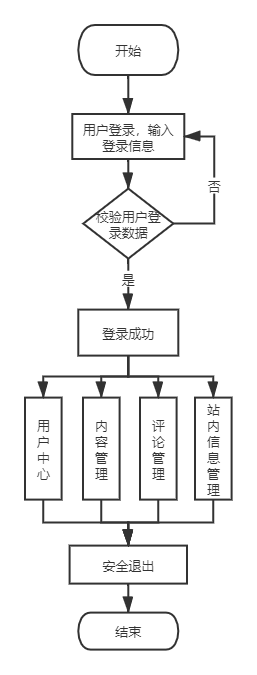
后台管理只能管理员进行操作，管理员在登录成功之后，可管理内容管理模块、用户中心模块、评论管理模块、站内信息管理模块。具体业务流程如图4-3所示。

图4-3后台管理系统业务流程图

### 4.2.2 SSO登录模块业务流程图

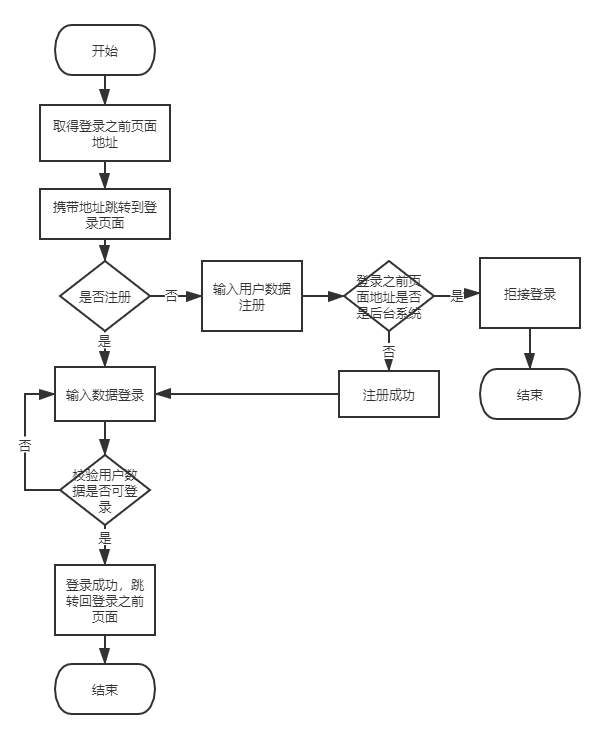
SSO登录的特点是，不管从哪个系统进入登录页面，在登陆成功之后，都可以返回到原来的系统。具体操作流程如图4-4所示。

图4-4 SSO单点登录业务流程图

### 4.2.3 板块内容管理业务流程图

板块内容管理业务主要由版主操作，所以在对板块内容操作之前需要检查登录用户是否是版主，如果不是版主，不显示可操作选项。如果是版主，可正常操作。具体操作业务流程如图4-5。

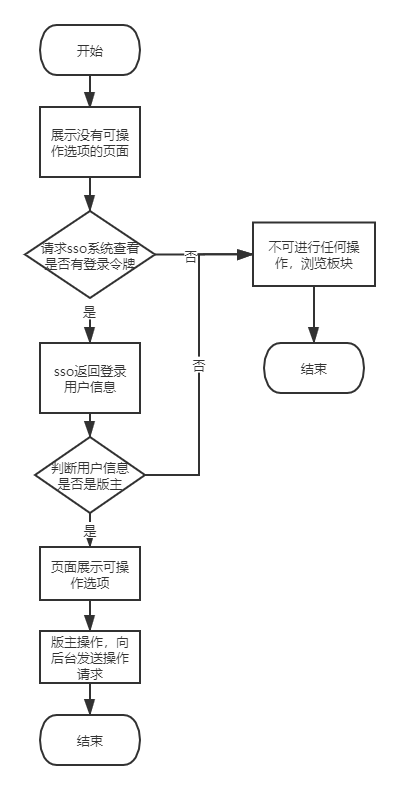


图4-5板块内容操作流程图

### 4.2.4帖子发布业务流程图

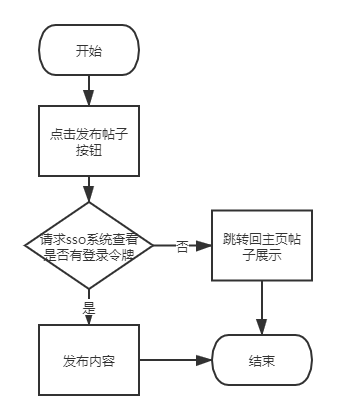
帖子发布需要由注册用户完成，故而发布时需要检查用户是否登录，如果没有登录则跳转登录页面，如果登录成功可正常发布帖子。具体操作流程如图4-6。

图4-6帖子发布业务流程图

### 4.2.5用户收藏文章流程图

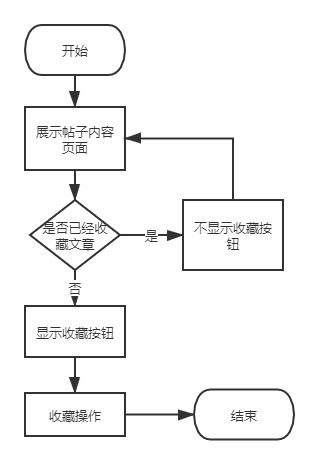
在用户收藏文章之前需要判断此文章是否已经收藏过了，如果已经收藏过了，不需要显示收藏按钮，否则可收藏。在点击收藏按钮的时候，需要判断用户是否登录，如果没有登录，则跳转到登录页面。具体操作流程如图4-7。

图4-7帖子发布业务流程图

## 4.3 数据库设计

### 4.3.1 概念结构设计

经过分析本系统一共有9个实体。分别为用户实体、板块实体、帖子实体、帖子类型实体、收藏实体、回帖实体、多级评论实体、站内消息实体、站内消息日志。

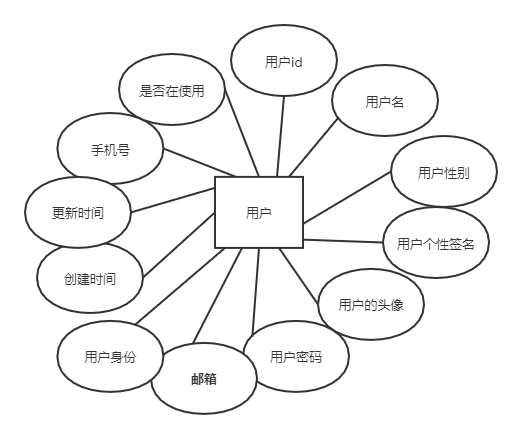
1. 用户实体，有用户id、用户名、用户性别、个性签名、头像链接、密码、邮箱、手机号、身份类别、是否正在使用、创建时间、更改时间属性。具体如图4-8所示。

图4-8用户实体图

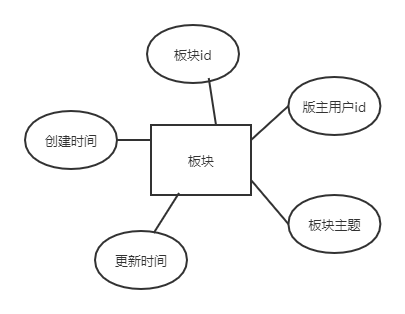
1. 板块实体，具有板块id、版主用户id、板块主题、创建时间、更新时间属性，具体如图4-9所示。

图4-9板块实体图

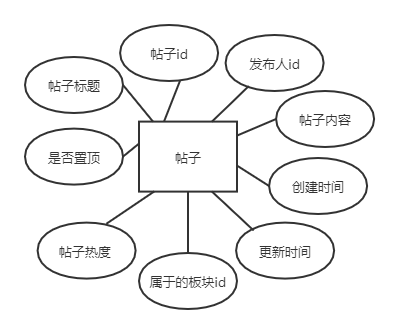
1. 帖子实体，具有帖子编号、发布人id、帖子标题、帖子内容、板块id、热度、是否置顶、创建时间、更新时间属性，具体如图4-10所示。

图4-10帖子实体图

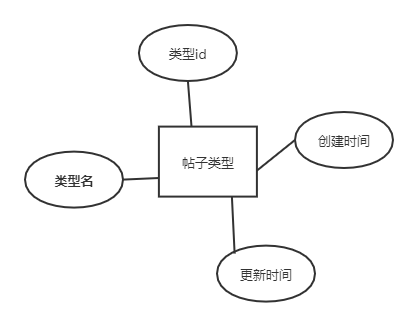
1. 帖子类型实体，具有类型id、类型名、创建时间、更新时间属性，具体如图4-11所示。

图4-11帖子类型实体图

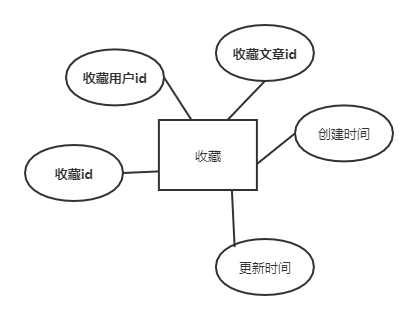
1. 收藏实体，具有收藏id、收藏用户id、收藏文章id、创建时间、更新时间属性，具体如图4-12所示。

图4-12收藏实体图

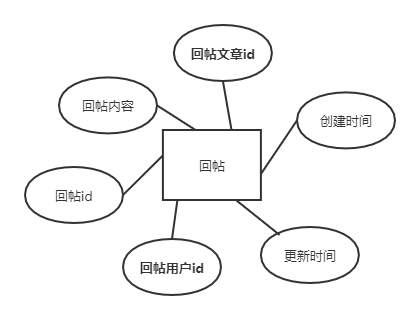
1. 回帖实体，具有回帖id、回帖内容、回帖文章id、回帖用户id、创建时间、更新时间属性，具体如图4-13所示。

图4-13回帖实体图

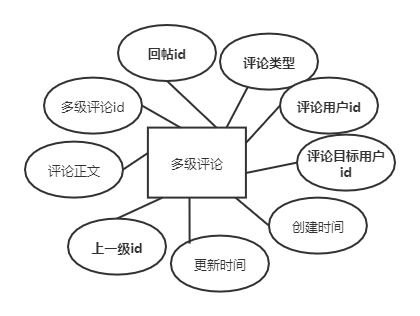
1. 多级评论实体，具有多级评论id、回帖id、评论正文、上一级id、评论类型、评论用户id、评论目标用户id、创建时间、更新时间属性，具体如图4-14所示。

图4-14多级评论实体图

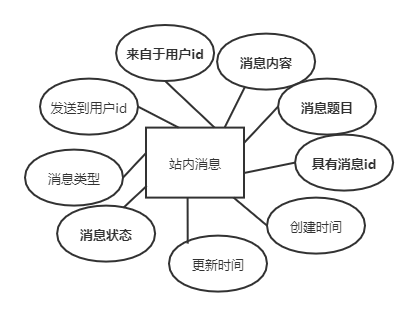
1. 站内消息实体，具有消息id、消息题目、消息内容、来自于用户id、发送到用户id、消息类型、消息状态、创建时间、更新时间属性，具体如图4-16所示。

图4-16 站内消息实体图

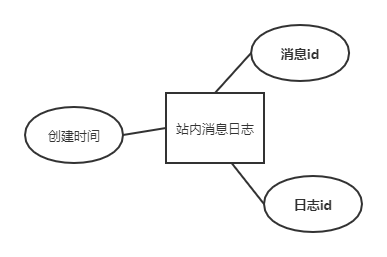
1. 站内消息日志实体：具有日志id、消息id、创建时间属性。具体如图4-17所示。

图4-17 站内消息日志实体图

以上是所有实体图，其中用户和关注粉丝实体是一对一关系、用户和用户粉丝是一对一关系，用户和帖子实体、收藏实体、回帖实体、多级评论实体、消息实体、消息日志实体、板块实体均为一对多关系。帖子和帖子类型为多对多关系。

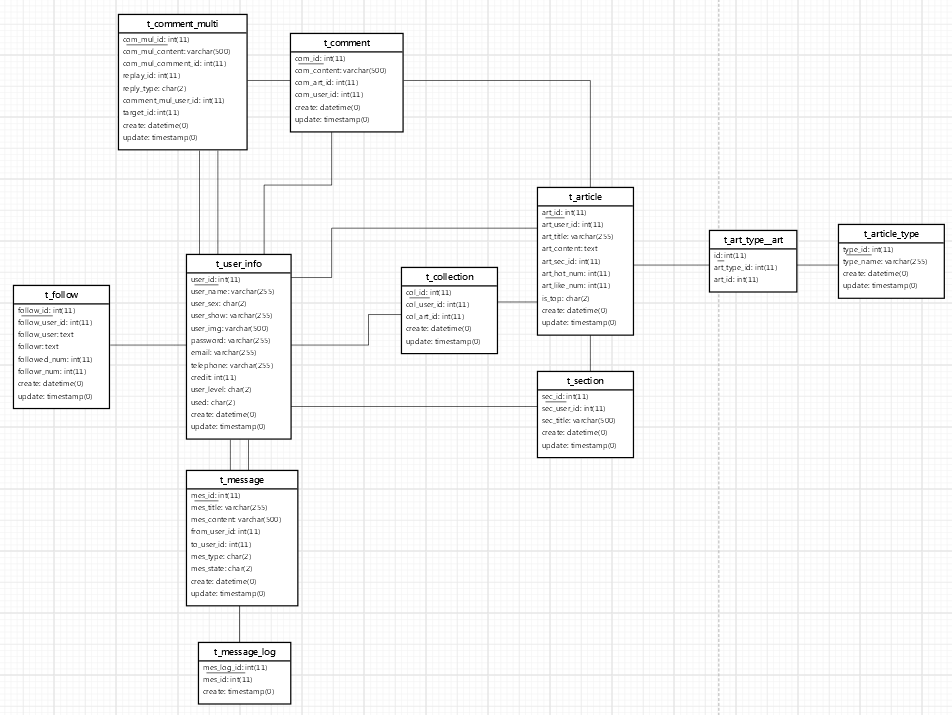
具体关系如图4-18所示。

图4-18概念结构设计图

### 4.3.2 物理结构设计

根据系统各实体属性分析，一共设计了用户表、板块表、帖子表、帖子类型表、帖子类型和帖子映射表、收藏表、回帖表、多级评论表、关注和粉丝表、站内消息表、站内消息日志表共11个数据库表。

1. 用户表t\_user\_info：用来存放用户信息，主要有user\_id，user\_name等字段，具体如表4-1所示。

表4-1 t\_user\_info表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性描述 | 属性类型 | 约束 |
| user\_id | 用户id | int(11) | primary key NOT NULL AUTO\_INCREMENT |
| user\_name  user\_sex  user\_show  user\_img  password  email | 用户名  用户性别  用户个性签名  用户头像链接  用户密码  用户邮箱 | varchar(255)  char(2)  Varchar（255）  Varchar（255）  Varchar（255）  varchar（255） |  |
| telephone  user\_level  used  create  update | 手机号  级别：1会员 2管理员  状态：0有效 1禁用  创建时间  更新时间 | varchar(255)  char(2)  datetime（0）  timestamp（0） | ON UPDATE CURRENT\_TIMESTAMP(0) |

1. 板块表t\_section：用来存储板块主题信息，主要有板块id、版主信息、板块内容等字段。具体如表4-2所示。

表4-2 t\_section表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性描述 | 属性类型 | 约束 |
| sec\_id | 板块id | int(11) | primary key NOT NULL AUTO\_INCREMENT |
| sec\_user\_id | 版主用户id | varchar(30) | references t\_user\_info(user\_id) |
| sec\_title | 板块主题 | varchar(500) |  |
| create | 创建时间 | datetime(0) |  |
| update | 更新时间 | timestamp(0) | ON UPDATE CURRENT\_TIMESTAMP(0) |

1. 帖子表t\_article：用来存储用户的发帖信息，具有文章id、文章标题、发帖用户id等字段，具体如表4-3所示。

表4-3 t\_article表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性描述 | 属性类型 | 约束 |
| art\_id | 文章id | int(11) | primary key NOT NULL AUTO\_INCREMENT |
| art\_user\_id  art\_title  art\_content  art\_sec\_id  art\_hot\_num  is\_top  create  update | 发帖用户id  文章标题  文章内容  所属板块id  文章热度  是否置顶1：置顶  创建时间  更新时间 | int(11)  varchar（20）  text  int(11）  int(11)  char（2）  datatime(0) timestamp（0） | references t\_user\_info(user\_id)  references t\_section(sec\_id)  ON UPDATE CURRENT\_TIMESTAMP(0) |

1. 帖子类型表t\_article\_type，：存储发布的帖子分类信息，主要含有类型id、类型名、创建时间等字段，具体如表4-4所示。

表4-4 t\_article\_type表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性描述 | 属性类型 | 约束 |
| type\_id | 类型id | int(11) | primary key NOT NULL AUTO\_INCREMENT |
| type\_name | 类型名 | Varchar（255） |  |
| create  update | 创建时间  更新时间 | datatime（0）  timestamp(0) | ON UPDATE CURRENT\_TIMESTAMP(0) |

1. 帖子和帖子类型映射表t\_art\_type\_\_art：此表主要用来映射帖子和帖子类型的多对多关系，是一个中间表，具有id，art\_type\_id，art\_id等字段，具体如表4-5所示。

表4-5 t\_art\_type\_\_art表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 属性描述 | 属性类型 | 约束 |
| id  art\_type\_id  art\_id | 映射编号  类型id  文章id | int（11）  int(11)  int(11) | primary key NOT NULL AUTO\_INCREMENT  references t\_art\_type(type\_id)  references t\_article(art\_id) |

1. 收藏表t\_collection：用来存储用户的收藏文章信息，主要包括col\_id、col\_user\_id、col\_art\_id等字段。具体如表4-6所示。

表4-6 t\_collection表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 属性描述 | 属性类型 | 约束 |
| col\_id  col\_user\_id  col\_art\_id  create  update | 收藏id  收藏对应的用户id  收藏的文章id  创建时间  更新时间 | int(11)  int（11）  int(11)  datatime(0)  timestamp(0) | primary key NOT NULL AUTO\_INCREMENT  references t\_user\_info(user\_id)  references t\_article(art\_id)  ON UPDATE CURRENT\_TIMESTAMP(0) |

1. 回帖表t\_comment：存储用户回复帖子的回帖信息，主要含有com\_id、com\_content、com\_art\_id等字段，具体如表4-7所示。

表4-7 t\_comment表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性描述 | 属性类型 | 约束 |
| com\_id  com\_content  com\_art\_id  com\_user\_id  create  update | 回帖id  回帖内容  回复的帖子id  回帖用户id  创建时间  更新时间 | int(11)  varchar(500)  int(11)  int(11)  datatime（0）  timestamp（0） | primary key NOT NULL AUTO\_INCREMENT  references t\_article(art\_id)  references t\_user\_info(user\_id)  ON UPDATE CURRENT\_TIMESTAMP(0) |

1. 多级回复表t\_comment\_multi：用来存储用户回复回帖信息的表。主要含有com\_mul\_id、com\_mul\_content、com\_mul\_comment\_id、reply\_id等字段，具体如表4-8所示。

表4-8 t\_comment\_multi表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性描述 | 属性类型 | 约束 |
| com\_mul\_id  com\_mul\_content  com\_mul\_comment\_id  reply\_id  reply\_type  comment\_mul\_user\_id  target\_id  create  update | 多级评论id  多级评论内容  多级评论对应的回帖id  回复的上一级id  回复类型  多级评论的用户id  评论的目标用户id  评论时间  更新时间 | int(11)  varchar(500)  int(11)  int(11)  char(2)  int(11)  int(11)  datatime(0)  timestamp(0) | primary key not null auto\_increment  references t\_comment(com\_id)  references t\_user\_info(user\_id)  references t\_user\_info(user\_id)  ON UPDATE CURRENT\_TIMESTAMP(0) |

1. 关注粉丝表t\_follow：存储用户的关注用户、粉丝用户，和用户是一对一关系。主要具有follow\_id, follow\_user\_id, follow\_user等字段，具体如表4-9所示。

表4-9 t\_follow表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性描述 | 属性类型 | 约束 |
| follow\_id  follow\_user\_id  follow\_user  followr  followed\_num  followr\_num  create  update | 表id  对应的用户id  关注的人，存储json数据  此人粉丝，存储json数据  关注的人数量  粉丝数量  添加数据时间  更新数据时间 | int(11)  int(11)  text  text  int(11)  int(11)  datatime(0)  timestamp（0） | primary key not null auto\_increment  references t\_user\_info(user\_id)  ON UPDATE CURRENT\_TIMESTAMP(0) |

1. 站内信表t\_message：用来存储站内信，当用户评论信息或者关注人员的时候，会产生一封站内信息。主要有mes\_id、mes\_title、mes\_content、from\_user\_id等字段。具体如表4-10所示。

表4-10 t\_message表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性描述 | 属性类型 | 约束 |
| mes\_id  mes\_title  mes\_content  from\_user\_id  to\_user\_id  mes\_type  mes\_state  create  update | 信息id  信息标题  信息内容  来源用户id  发送到的用户id  信息类型  信息状态  信息产生时间  信息产生变化时间 | int(11)  varchar（255）  varchar（500）  int(11)  int(11)  char(2)  char(2)  datatime(0)  timestamp(0) | primary key not null auto\_increment  references t\_user\_info(user\_id)  references t\_user\_info(user\_id)  ON UPDATE CURRENT\_TIMESTAMP(0) |

1. 站内信息日志表t\_message\_log：存储是否有用户读取信息情况，当用户读取信息或者删除信息时，会产生一个信息日志。主要具有mes\_log\_id、mes\_id等字段，具体如表4-11所示。

表4-11 t\_message\_log表

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 属性描述 | 属性类型 | 约束 |
| mes\_log\_id  mes\_id  create | 日志id  消息id  日志创建时间 | int(11)  int(11)  datatime | primary key not null auto\_increment  references t\_message(mes\_id) |

# 5 系统实现

## 5.1 后台管理系统

### 5.1.1后台管理系统的登录

本系统采用分布式架构,前后端分离开发，所以SSO登录系统和后台管理系统并不是一个web项目。SSO登录系统部署在http://localhost:8083，后台管理系统部署在http://localhost:8080。在浏览器输入http://localhost:8080时，管理系统的拦截器会拦截请求，查看是否携带用户令牌（cookie实现），如果没有携带用户令牌即用户未登录，返回一个重定向请求，携带登录成功之后的重定向路径跨域访问http://localhost:8083/html/login/login.html?redirectUrl=http://localhost:8080。即登录页面，登录页面如图5-1所示

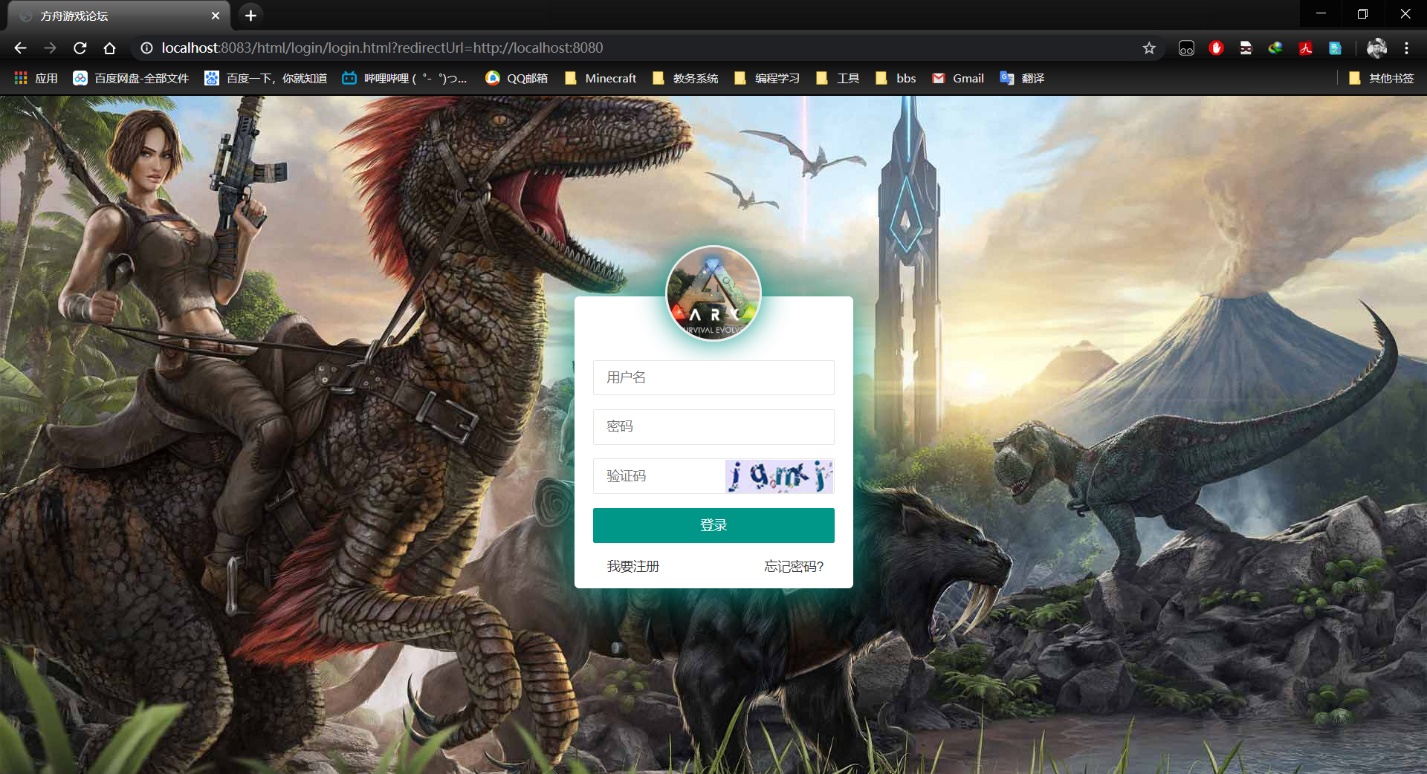


图5-1方舟游戏论坛系统登录界面图

用户输入登录信息，前台首先会验证用户输入信息是否合法，当点击登录的时候，向后台发送一个ajax请求，如果后台验证成功，会携带带有用户令牌的cookie响应给浏览器，并发送一个重定向代码，重定向路径即为redirectUrl参数，再次访问后台管理系统的主页，此次拦截器拦截请求的时候，用户携带了登录令牌，即可跳转入后台首页。后台首页如图5-2所示。

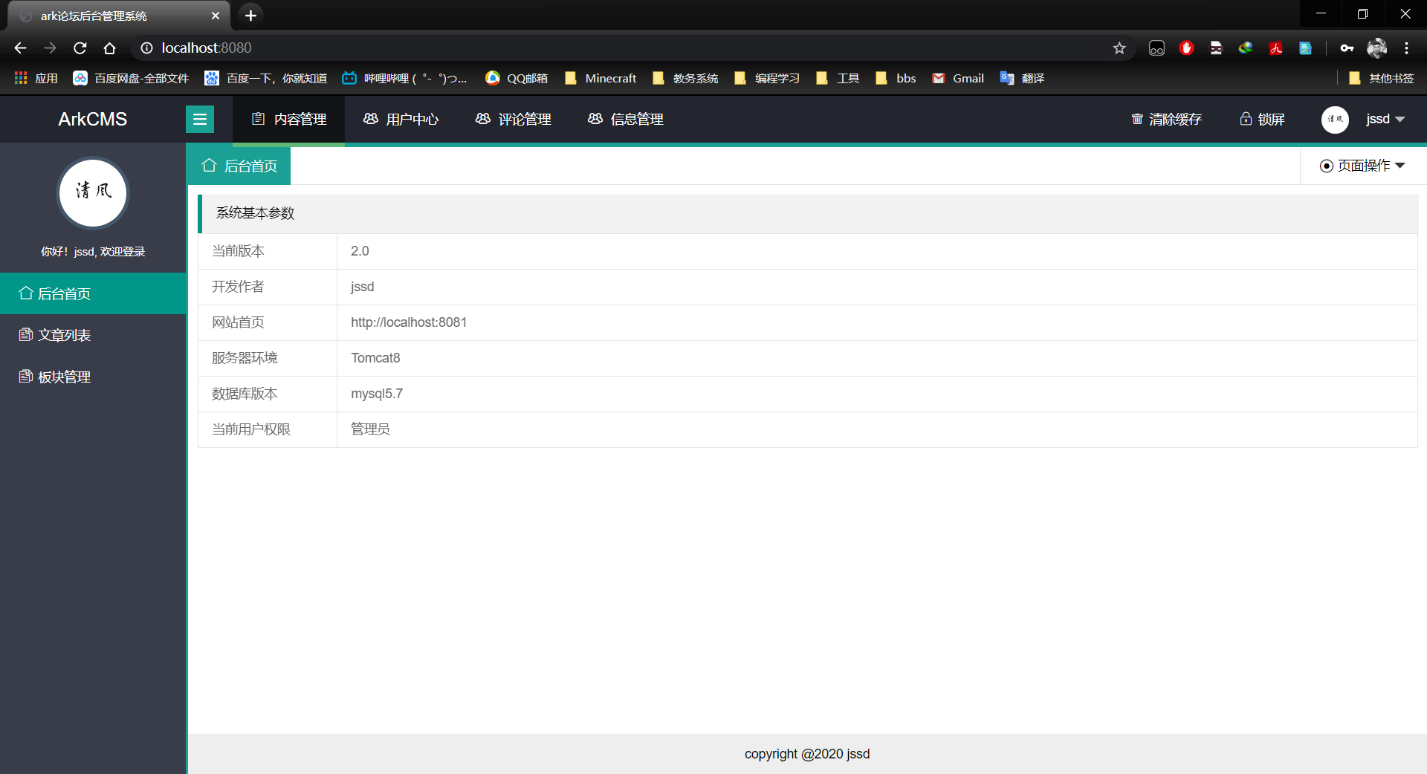
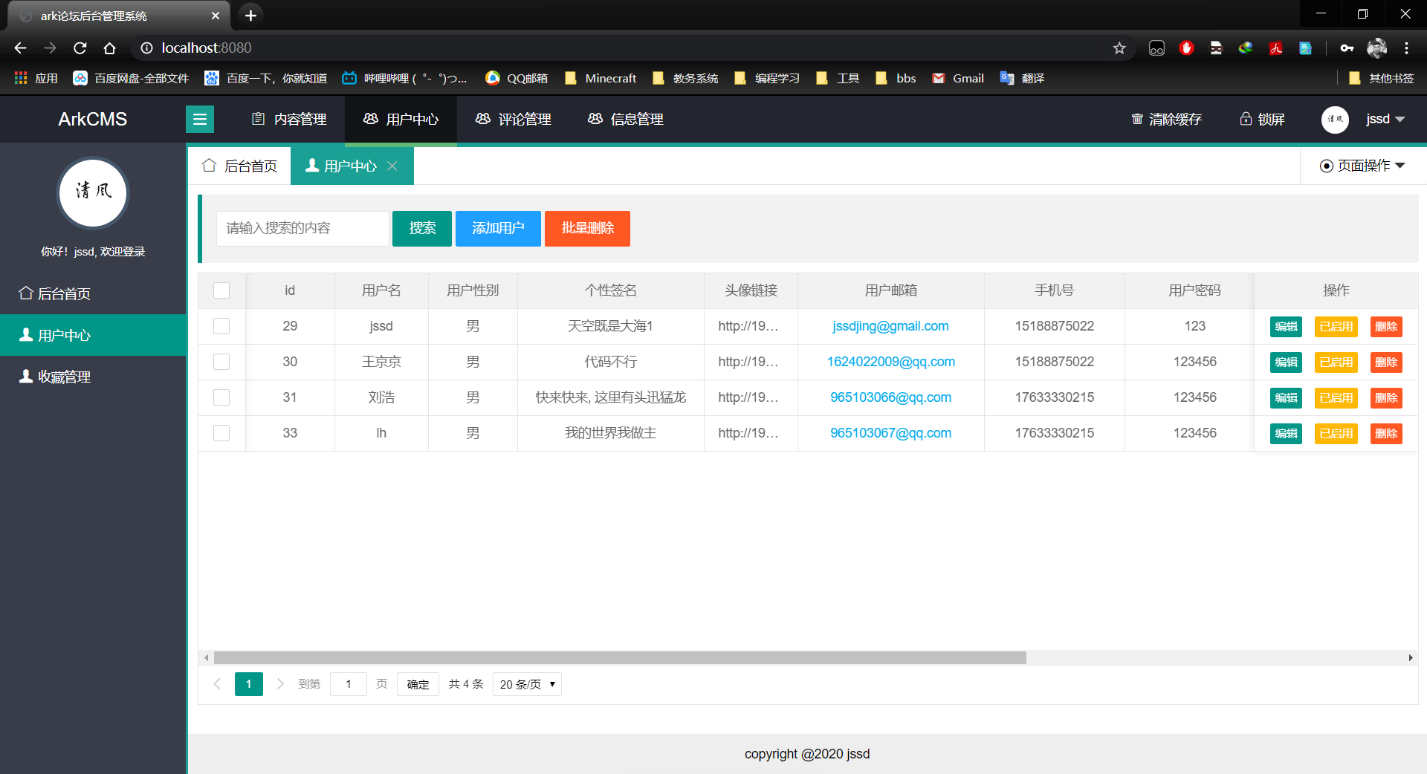


图5-2后台管理系统主界面图

### 5.1.2 用户信息管理模块

选择上侧导航栏的用户中心选项卡，左侧导航栏会出现用户信息和收藏管理选项卡。点击用户信息，即可进入用户信息管理中心。用户信息管理中心显示的是所有用户的详细信息，可以按照用户名来进行检索用户。当用户信息比较多的时候，即可选择分页查看。右侧操作栏中可以编辑用户信息，启用废弃用户，和删除用户。选择指定用户，可批量删除。点击添加用户，可录入新的用户信息。用户管理界面如图5-3所示。添加编辑用户界面如图5-4所示。



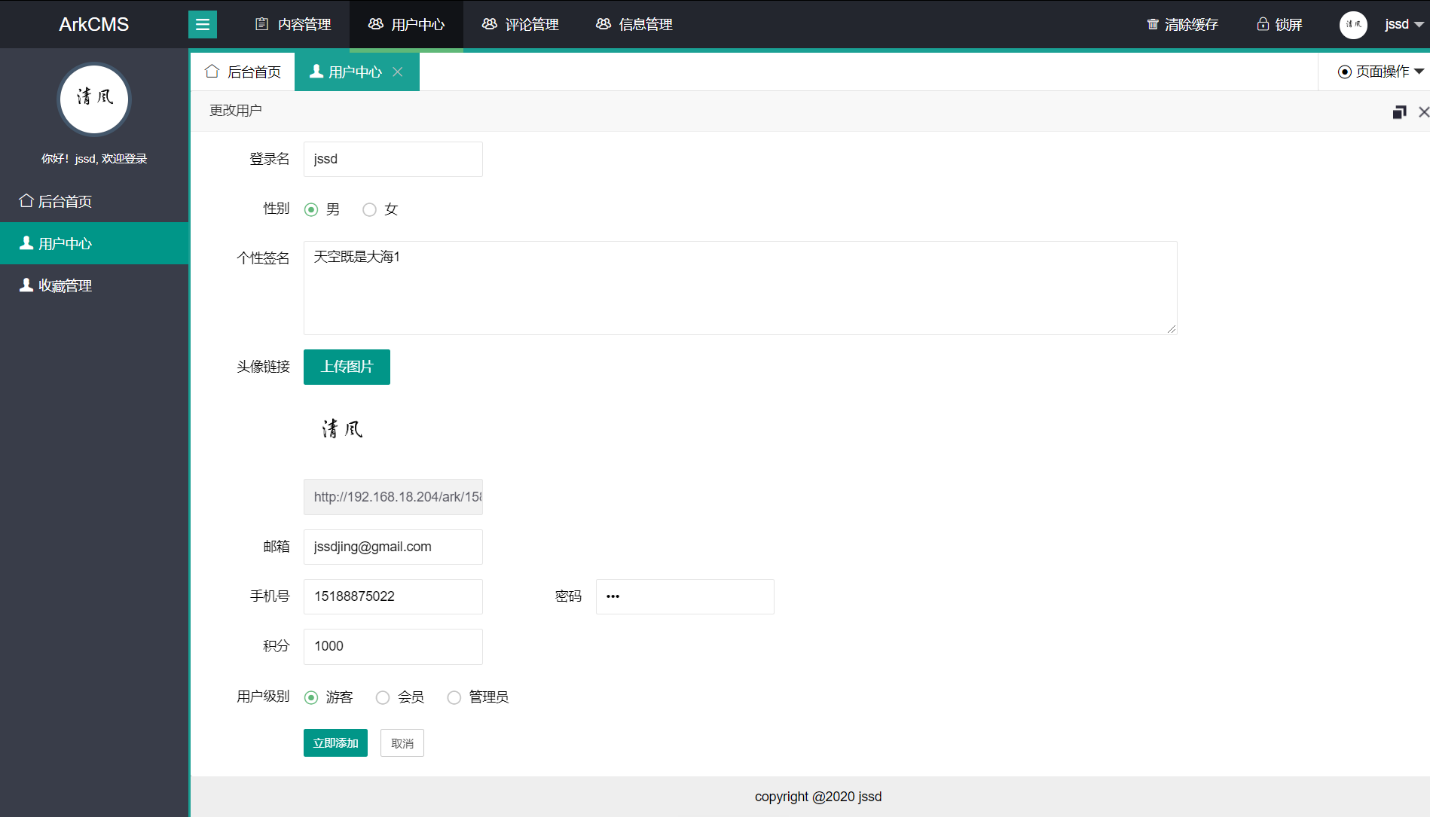
图5-3用户信息管理界面图

图5-3用户信息添加或修改管理界面图

### 5.1.3 用户收藏管理模块

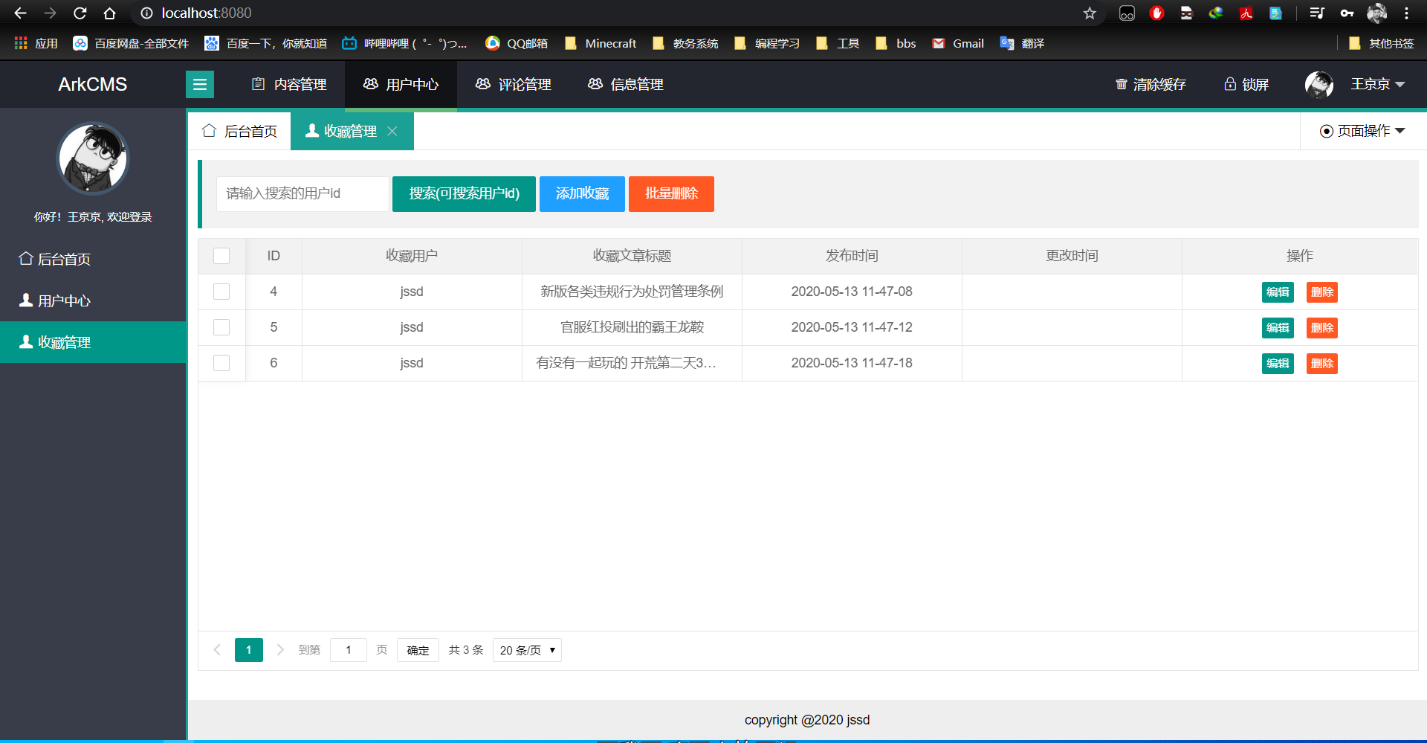
在用户中心的子导航栏中，选择收藏管理，即可打开收藏管理页面。在收藏管理页面中，可以根据用户id搜索指定用户的收藏内容。同样可以编辑用户收藏内容。选择多个收藏，可批量删除。具体管理页面如图5-4所示。

图5-4用户收藏管理界面图

### 5.1.4 帖子信息管理模块

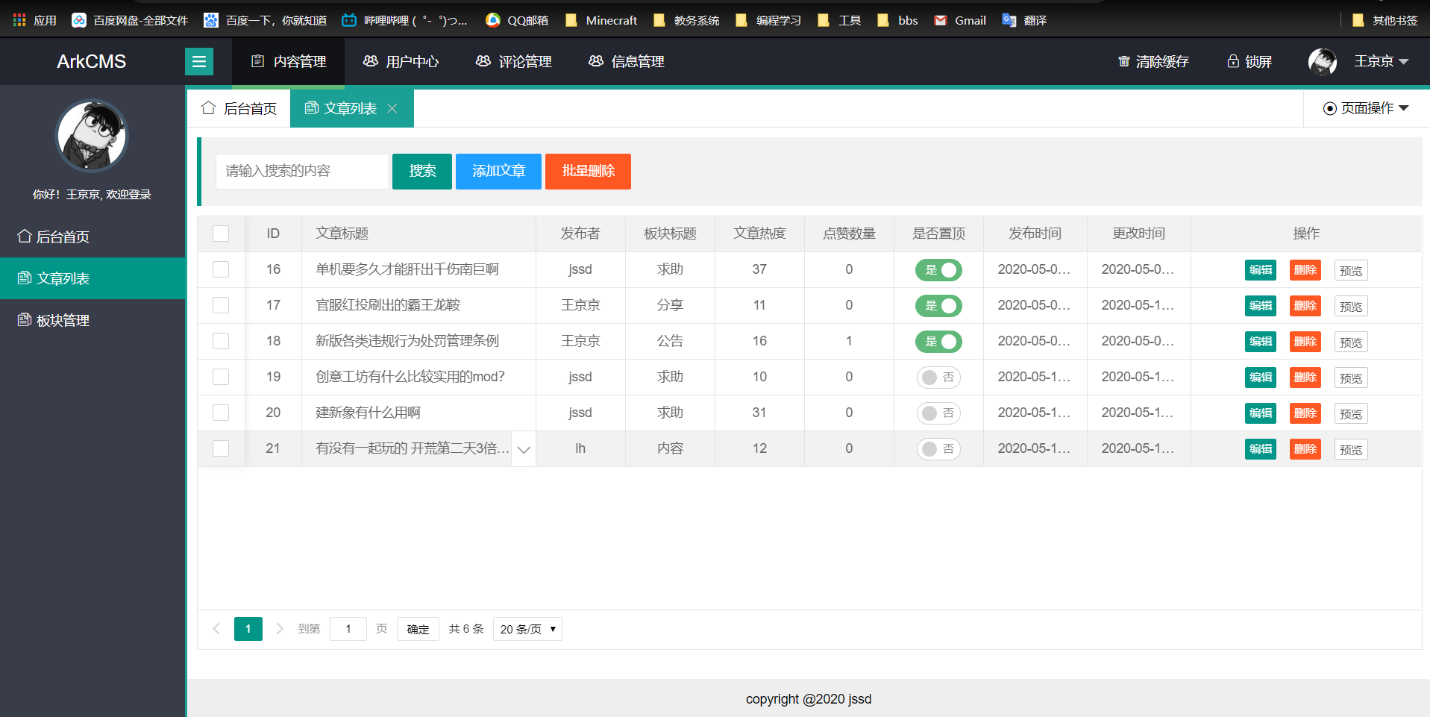
选择上侧导航菜单栏“内容管理”进入该模块后，可以看到左侧有文章列表的字导航，选择文章列表子导航，即可进入帖子内容管理模块。在搜索栏中，可键入文章标题模糊搜索一篇帖子，也可键入用户名，搜索指定用户发布的帖子。列表可将指定文章置顶，编辑、删除。选择多篇文章，可批量删除文章。在文章过多时，也可分页显示。文章展示如图5-5所示。

图5-5帖子详情列表展示图

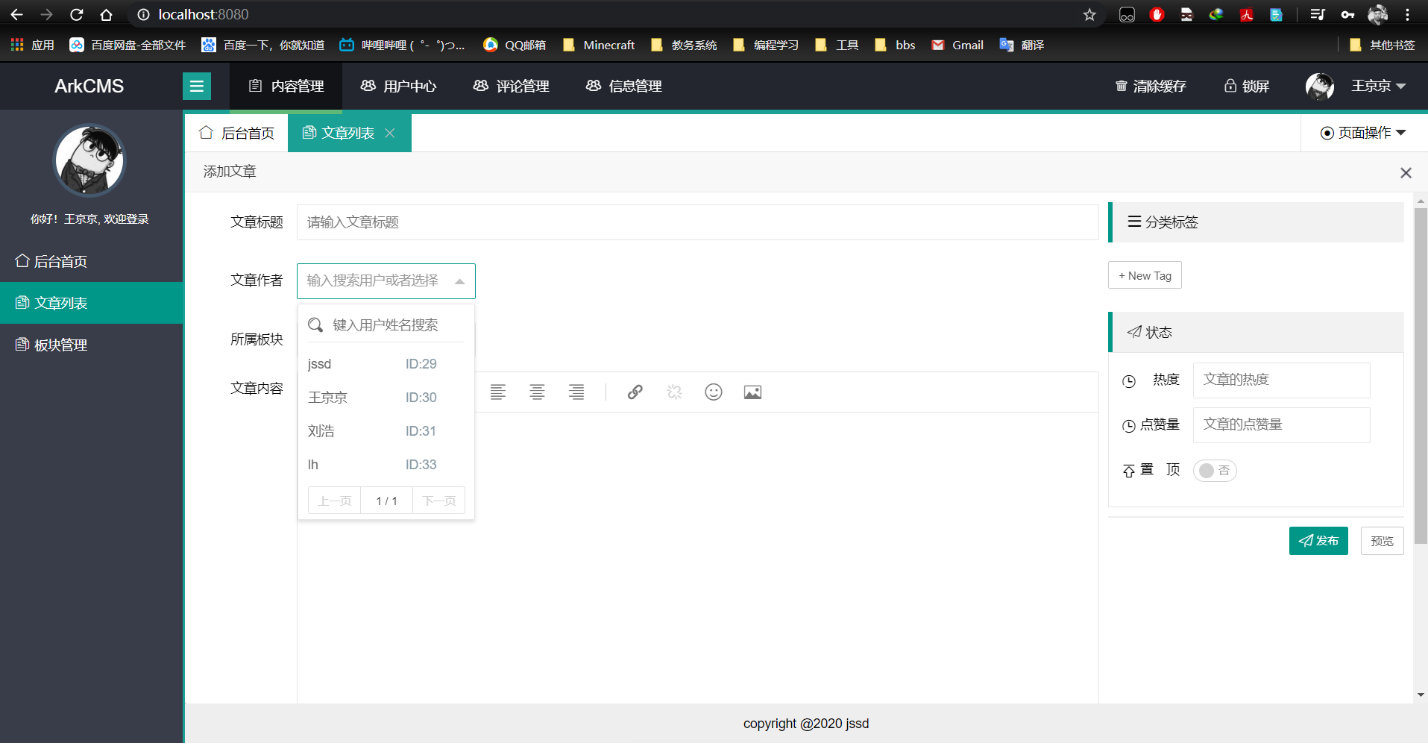
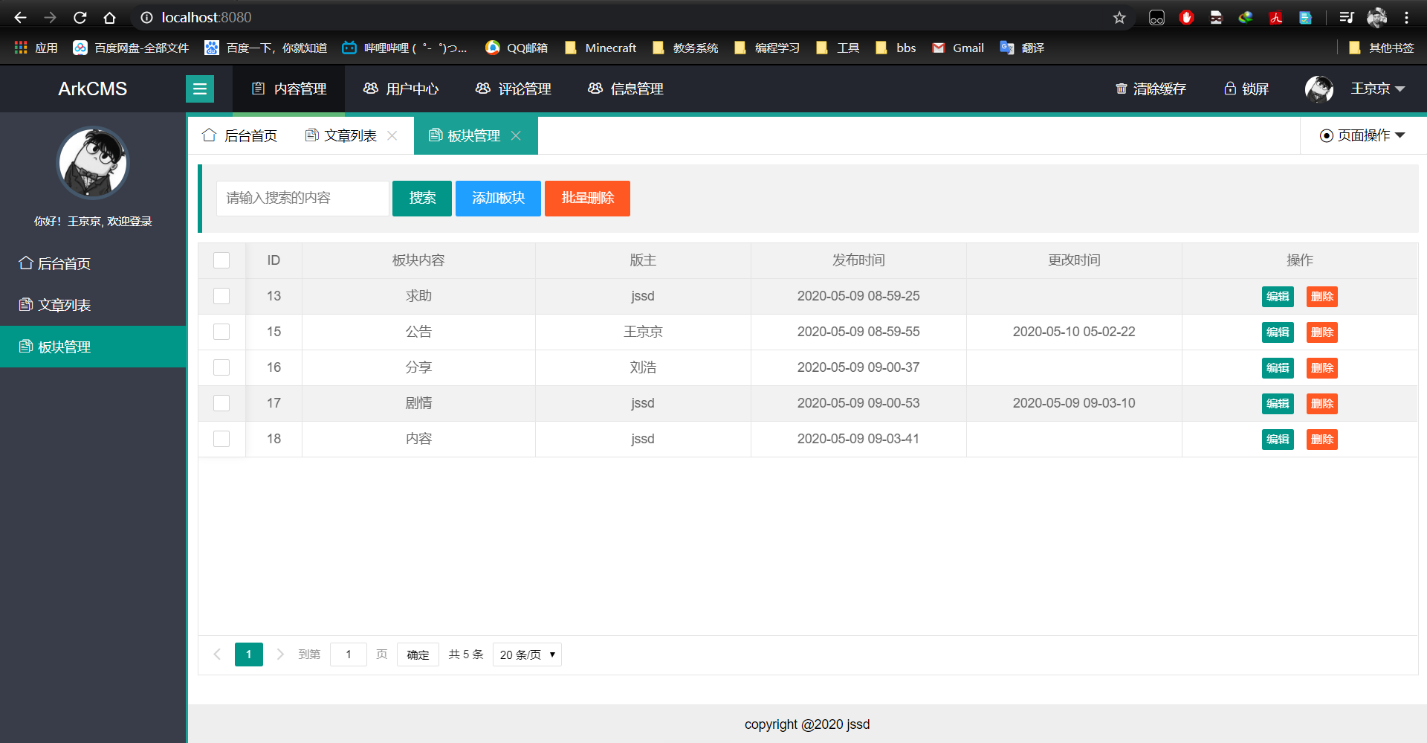
选择添加文章，或者编辑文章，可添加或者编辑一篇文章。添加文章的时候，可下拉用户列表，选择文章属于哪个用户发布，如果下拉列表中不存在，可键入搜索指定用户。板块操作类似。右侧分类标签中，可键入关键字为文章分类。操作详情如图5-6所示。

图5-6帖子添加编辑展示图

### 5.1.5 板块内容管理模块

在内容管理导航下，选择左侧板块管理子导航可进入板块管理模块。在板块管理模块中，分页显示了多个板块的信息，在搜索栏中，可以输入板块内容进行搜索板块内容。选择多个板块，可以批量删除。点击添加板块，可以添加一个板块信息，点击编辑，可以修改板块的版主信息。具体操作页面如图5-7所示，添加或编辑板块信息如图5-8所示。

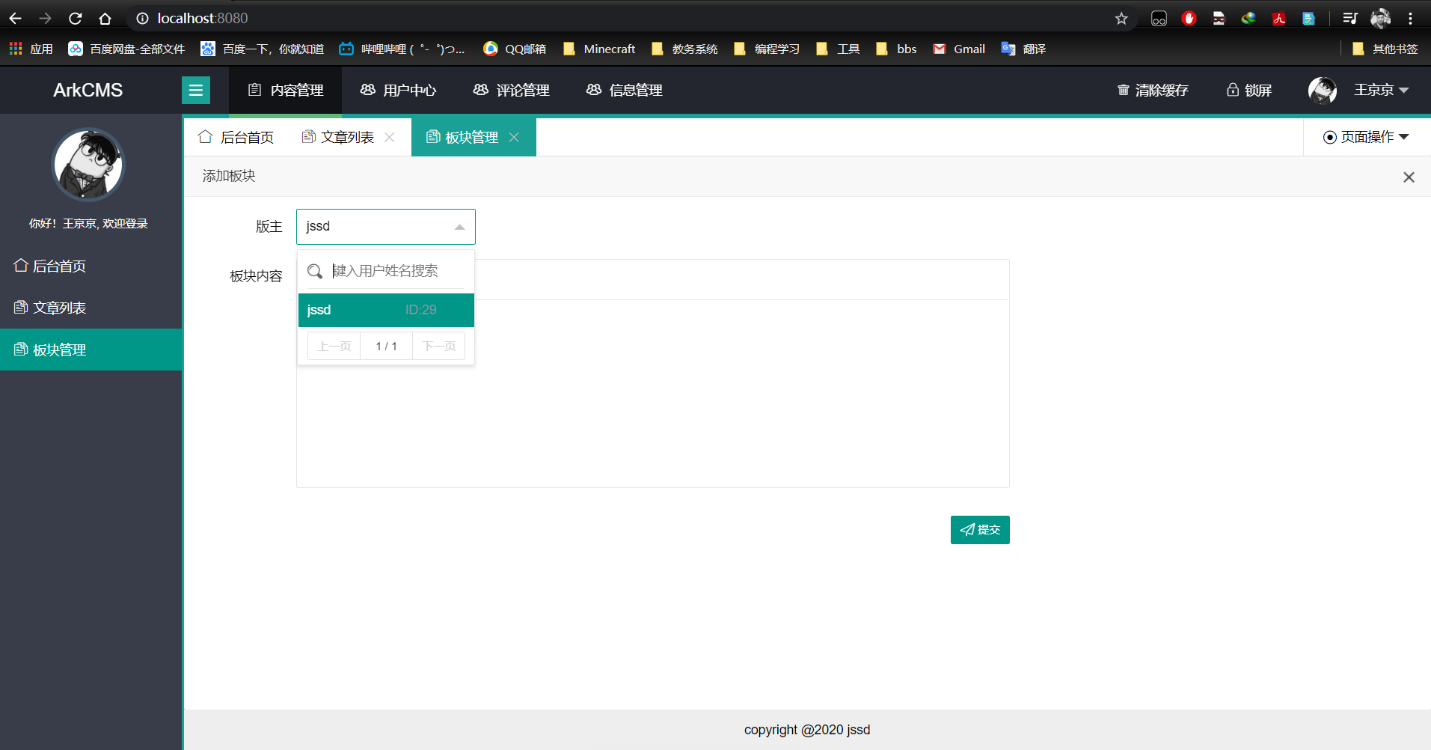
图5-7 板块信息管理界面图

图5-8 板块信息修改/添加界面图

### 5.1.6 回帖管理管理模块

选择上侧导航菜单栏“评论管理”，可以看到左侧有一级评论列表和多级评论列表两个子导航，点击一级评论列表，即可进入回帖管理模块。进入该模块，可分页显示所有回帖信息，在搜索框中输入文章id，可搜索指定文章的回帖信息。点击编辑，可以更改评论内容。选择多个评论内容，点击批量删除可实现批量删除功能。详细操作如图5-9所示。

图5-9 回帖信息详情界面图

### 5.1.7多级回复信息管理

在评论管理导航下，点击多级评论列表子导航，即可看到多级评论信息的分页展示。在搜索栏中，可以根据不同选项搜索多级评论。点击删除可以删除一个多级评论，选择多个评论信息，可以实现批量删除功能。点击添加多级评论按钮，可以添加一个多级评论信息。具体操作详情如图5-10所示。

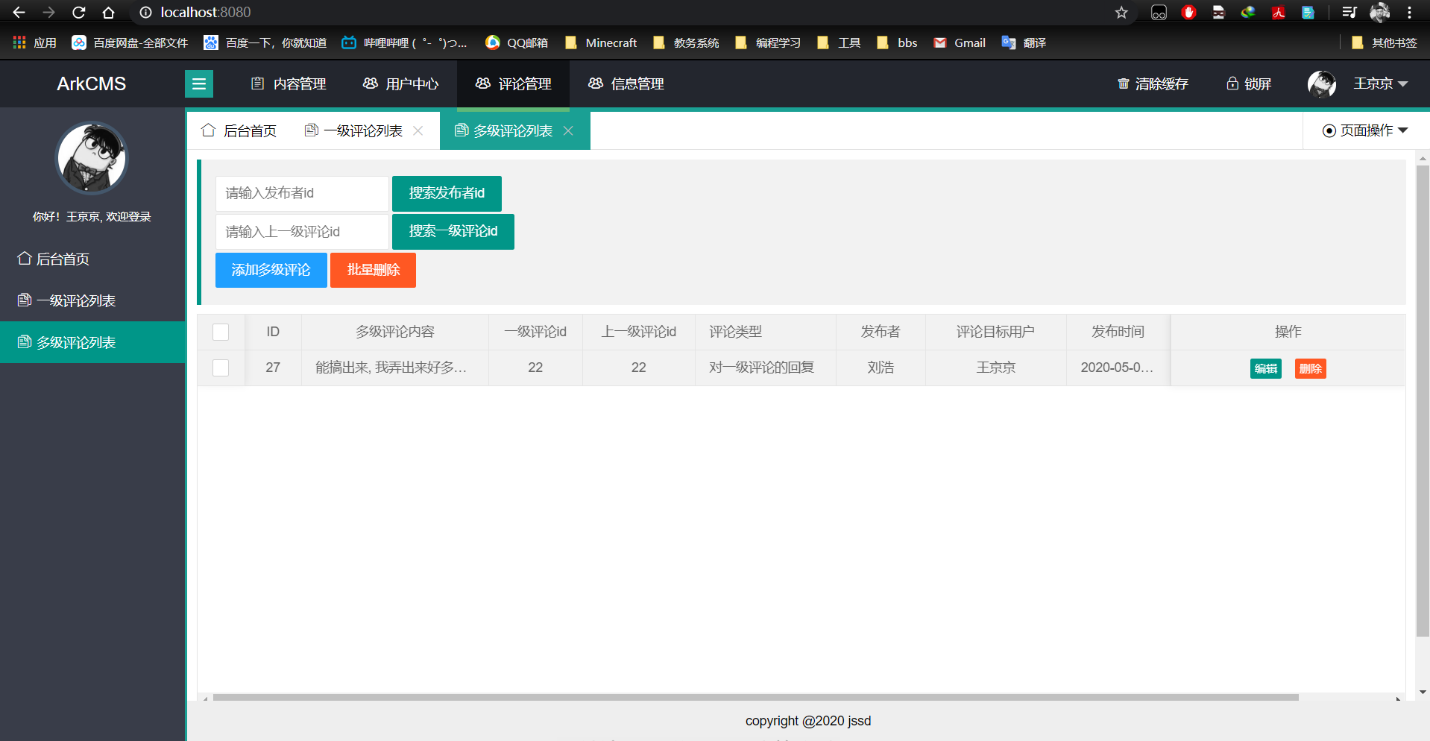


图5-10 多级评论管理界面图

### 5.1.8 信息管理模块

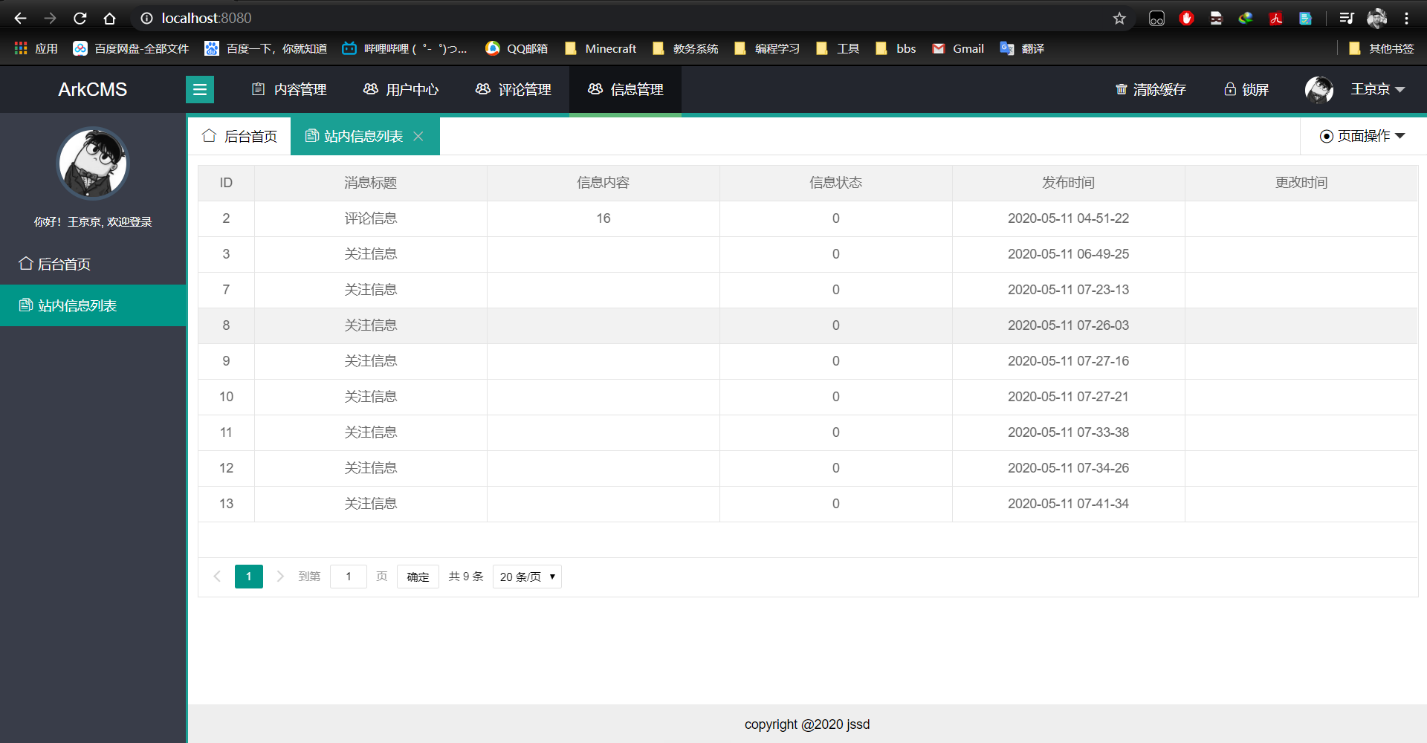
选择上侧导航菜单栏中的“信息管理”，可以看到左侧出现了站内信息列表子导航，点击此导航，可以分页展示站内信息。因为站内信息是系统自动产生，故不可更改，只能展示。如图5-11所示。

图5-11站内信息展示界面图

## 5.2 前台论坛系统

### 5.2.1 论坛主页

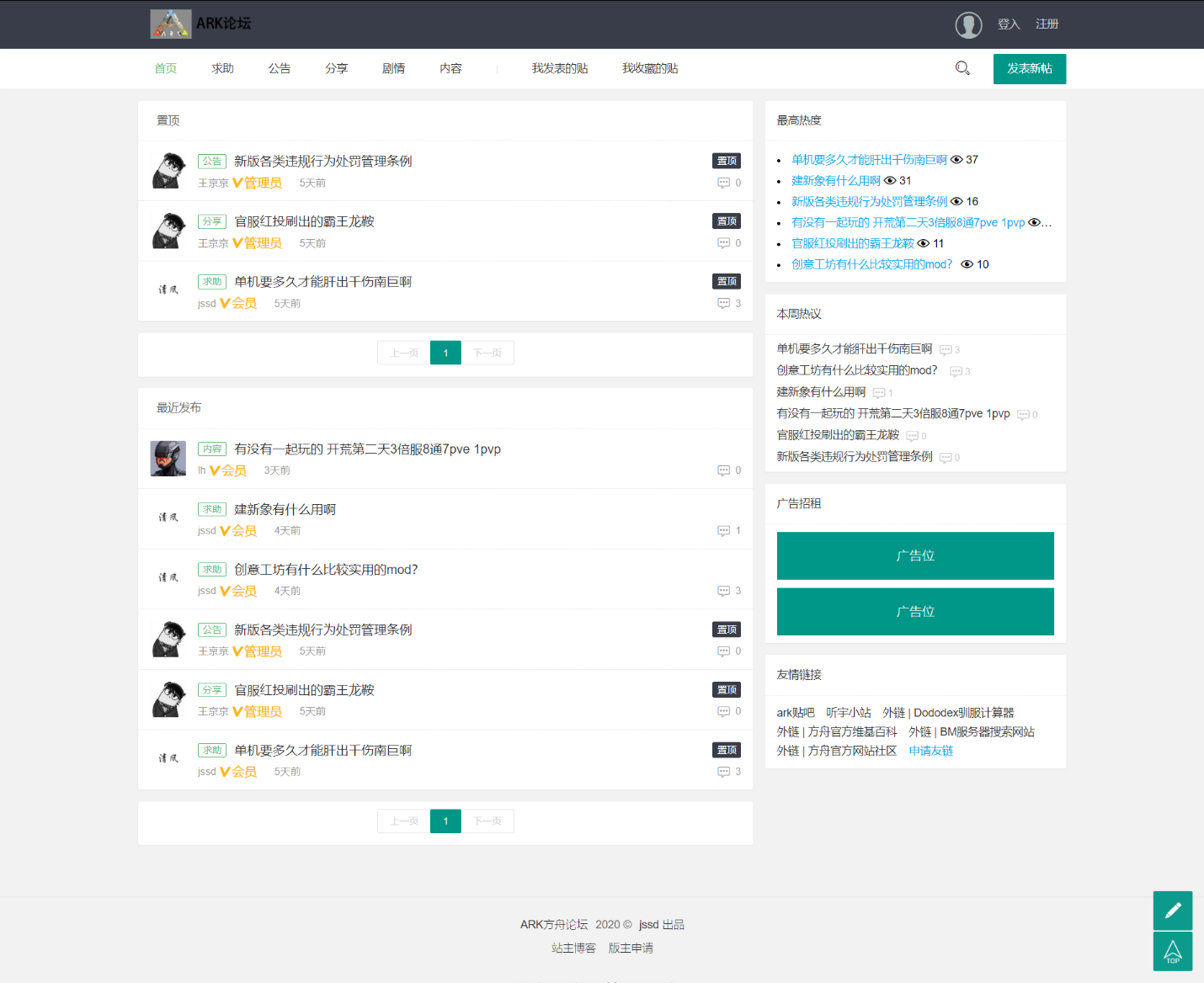
论坛系统部署在http://localhost:8081下。在浏览器中输入链接，可直接访问到论坛首页页面。在首页中，会分页显示置顶文章，最近发布内容。导航栏会显示多个板块信息。右侧分多个展示卡，分别显示浏览次数最多得几篇文章导航、评论最多得几篇文章导航、广告栏和友情链接。前端页面如图5-14所示。

图5-14论坛主页界面图

### 5.2.2 板块详情页

当点击上侧导航栏中的任一选项卡， 即可进入对应的板块详情页。板块详情页会分页显示当前板块中的帖子内容，以时间排序。右侧展示卡中会分别展示最高热度的文章链接、讨论最多的文章链接。详情如图5-15所示。

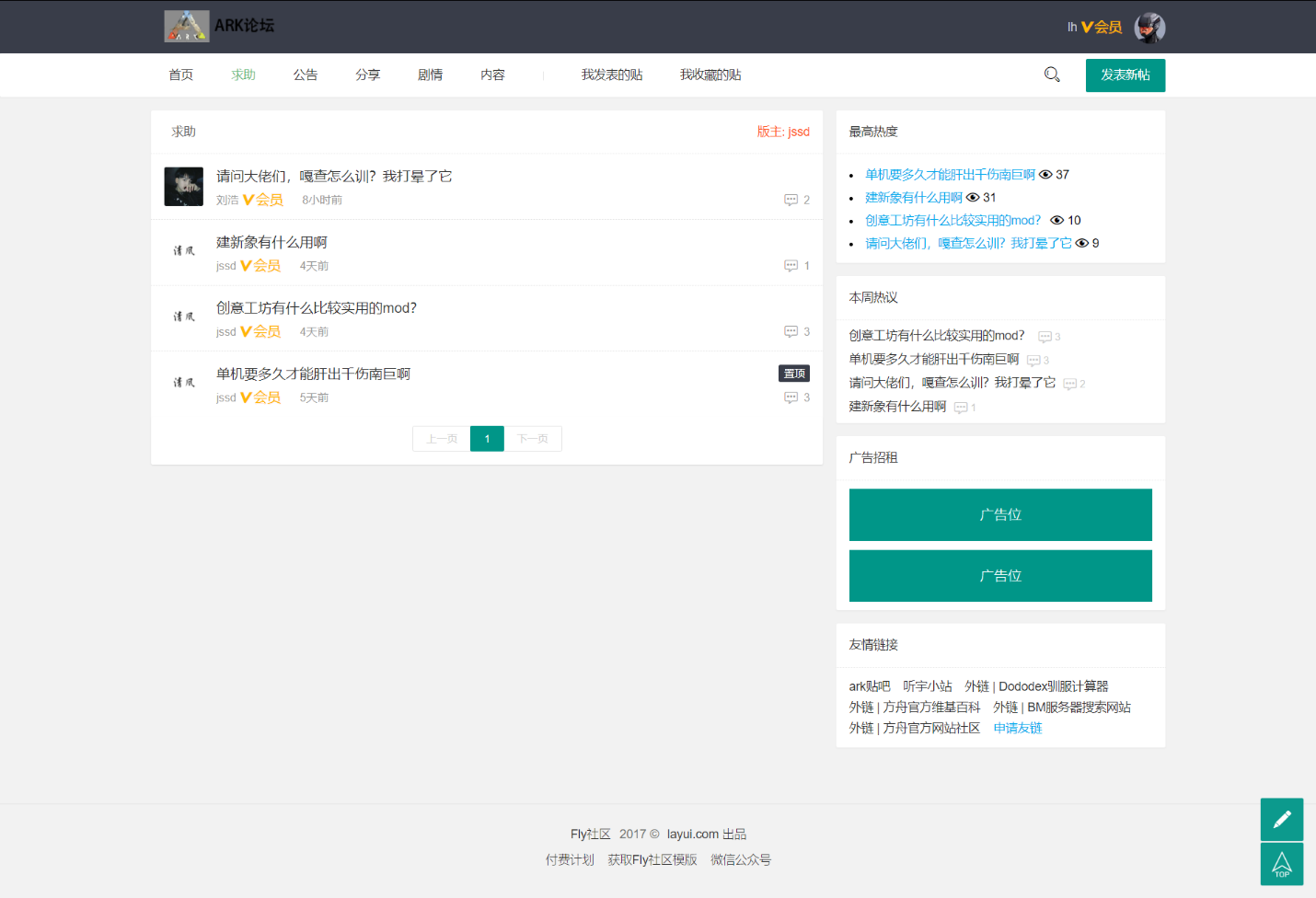


图5-15板块详情界面图

### 5.2.3 文章内容展示页

当你想要浏览一篇文章的时候，点击任意页面的文章标题，均可调转到文章详情页面。文章详情页面会显示文章标题、发布用户、发布时间、是否置顶、文章内容、文章类型等信息。右侧展示卡中分别展示了发布用户的信息、文章分类信息、本周讨论最多的文章等信息。页面底部，会显示用户的评论信息，如果评论信息过多，可流式加载。也可以发布回帖信息，前提是需要登录，如果你没有登录就选择回帖，那么系统会自动跳转到登录页面。详情如图5-16所示。

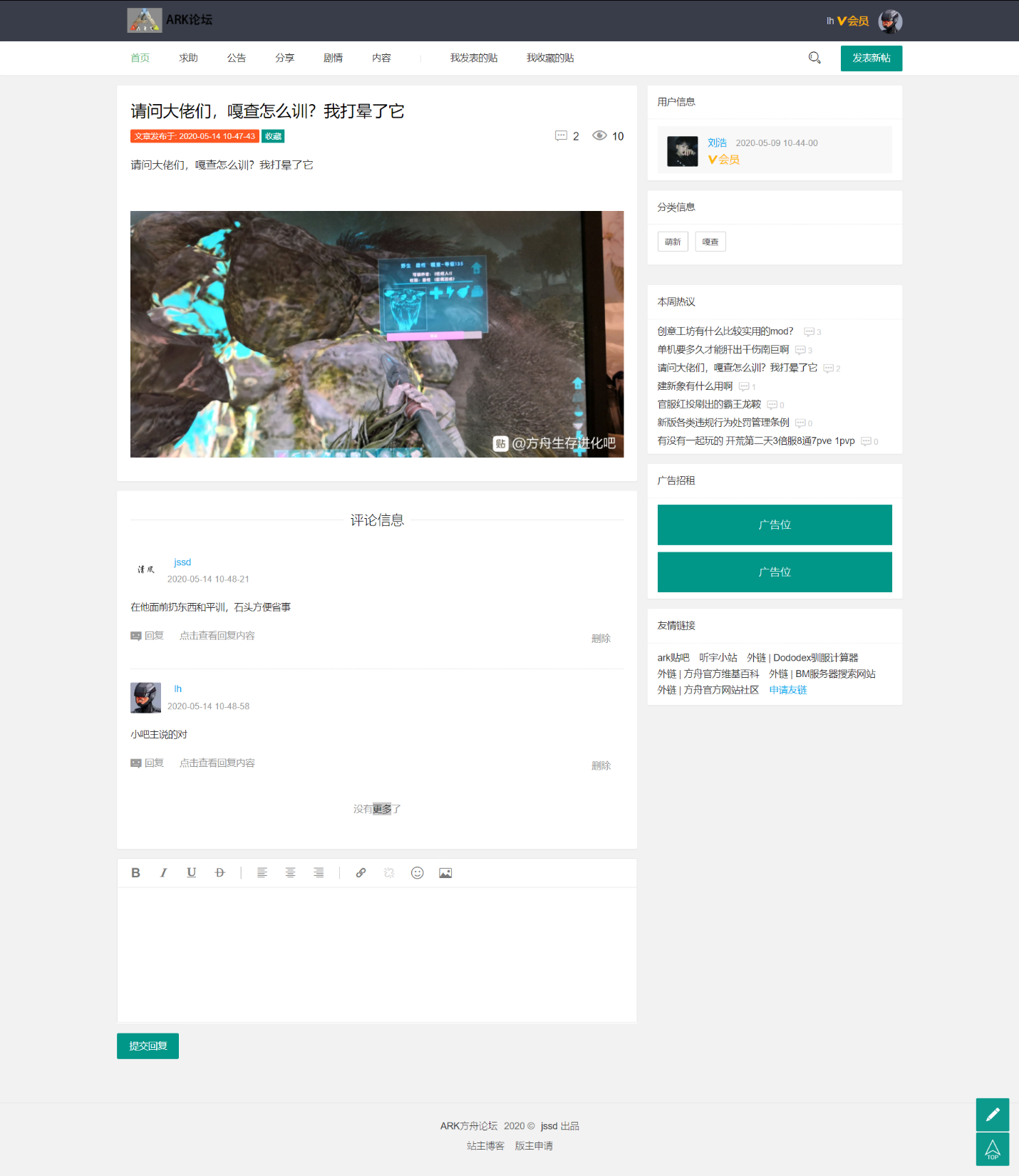
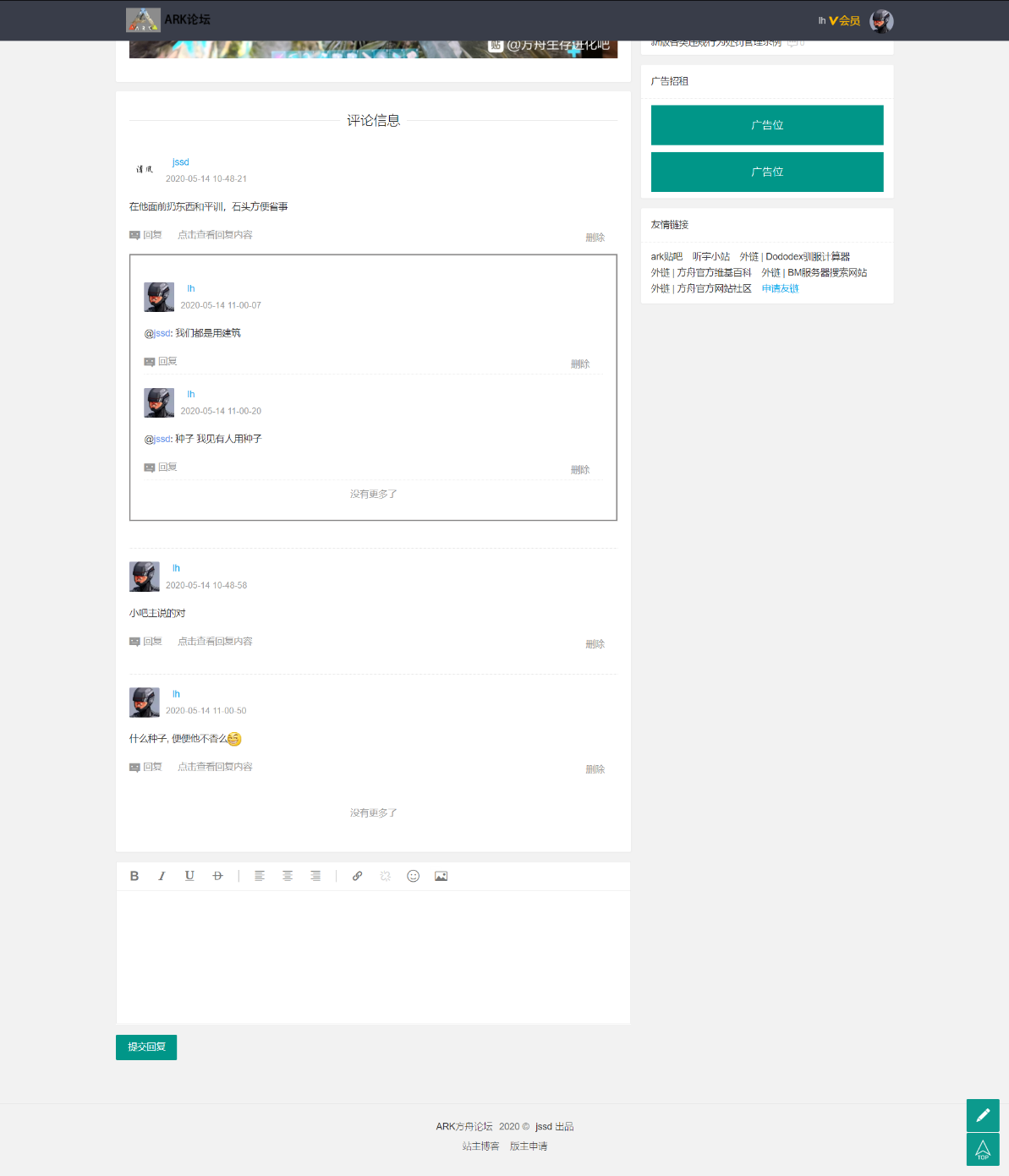


图5-16帖子详情展示图

### 5.2.4 多级评论展示

在文章的详细页面中，可现实对文章的回帖信息一级对回帖信息的评论信息，当回复数据过多时，会流式加载更多的评论信息。详情如图5-17所示。

图5-17评论及多级评论界面图

### 5.2.5用户中心页面

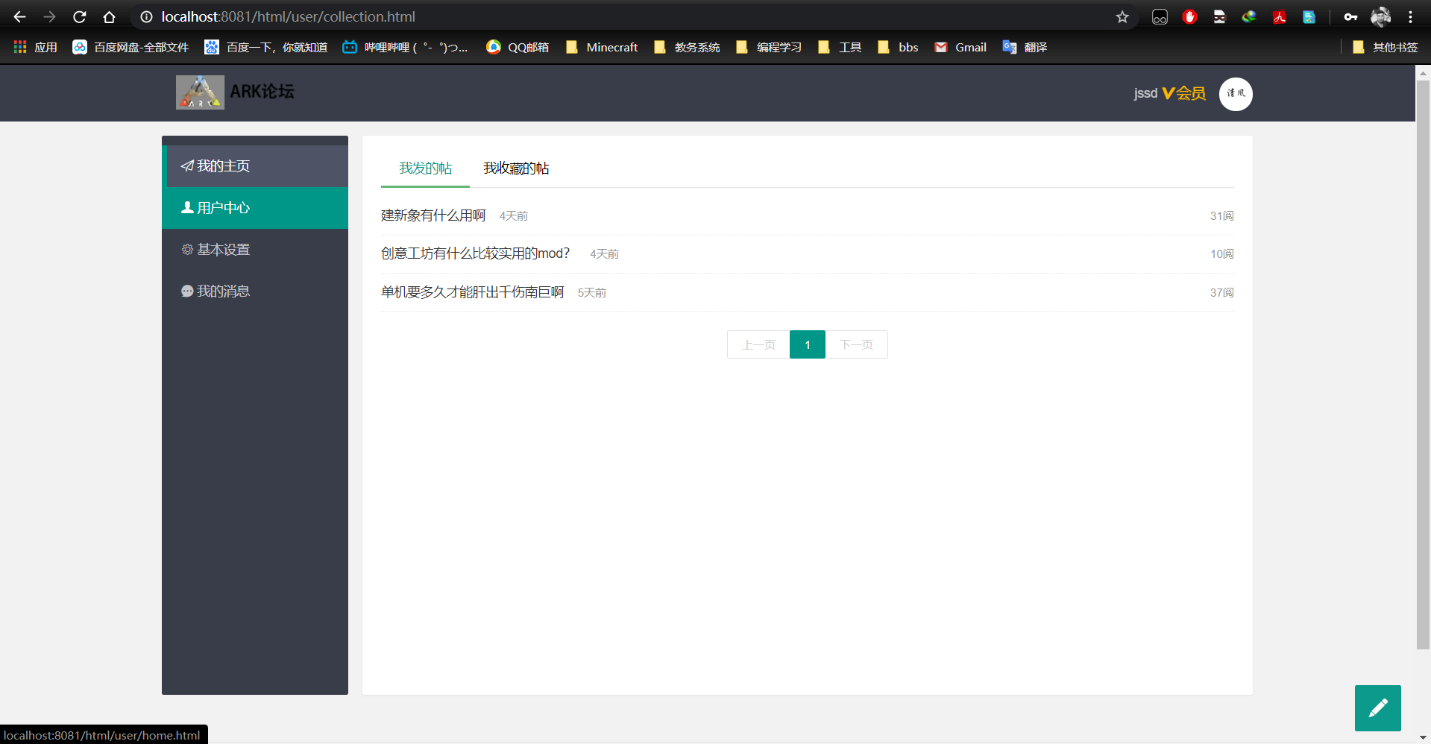
当点击上侧导航中我发表的帖子或我收藏的帖子时，会跳转到用户中心，可查看我发布的帖子信息和我收藏的帖子信息。分页显示，以时间排序。具体如图5-18所示。

图5-18用户中心界面图

### 5.2.6用户主页

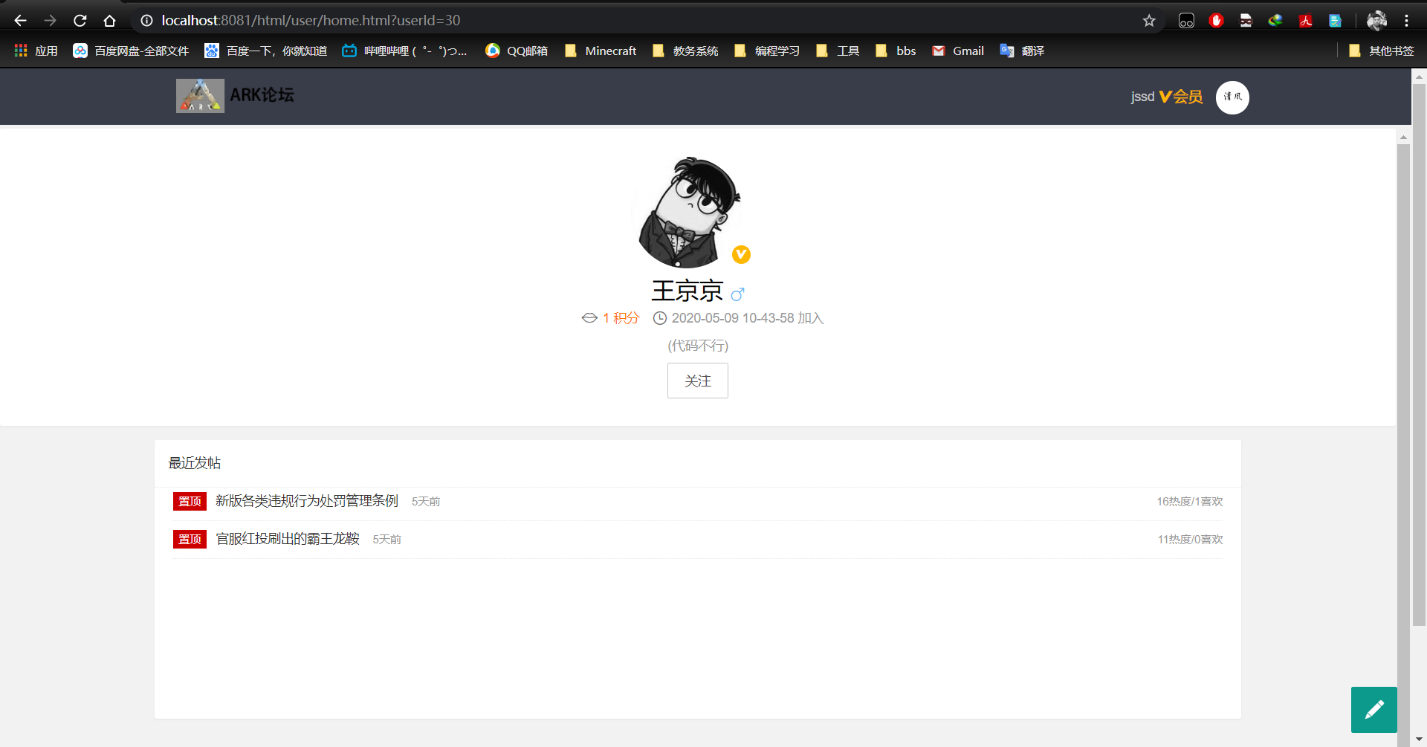
当点击任一用户头像，或从自己头像处，可进入用户主页，用户主页中可看到用户的基本信息，当你浏览他人的信息时，可以选择关注用户。详情如图5-19所示。

图5-19用户主页图

### 5.2.7用户信息修改页

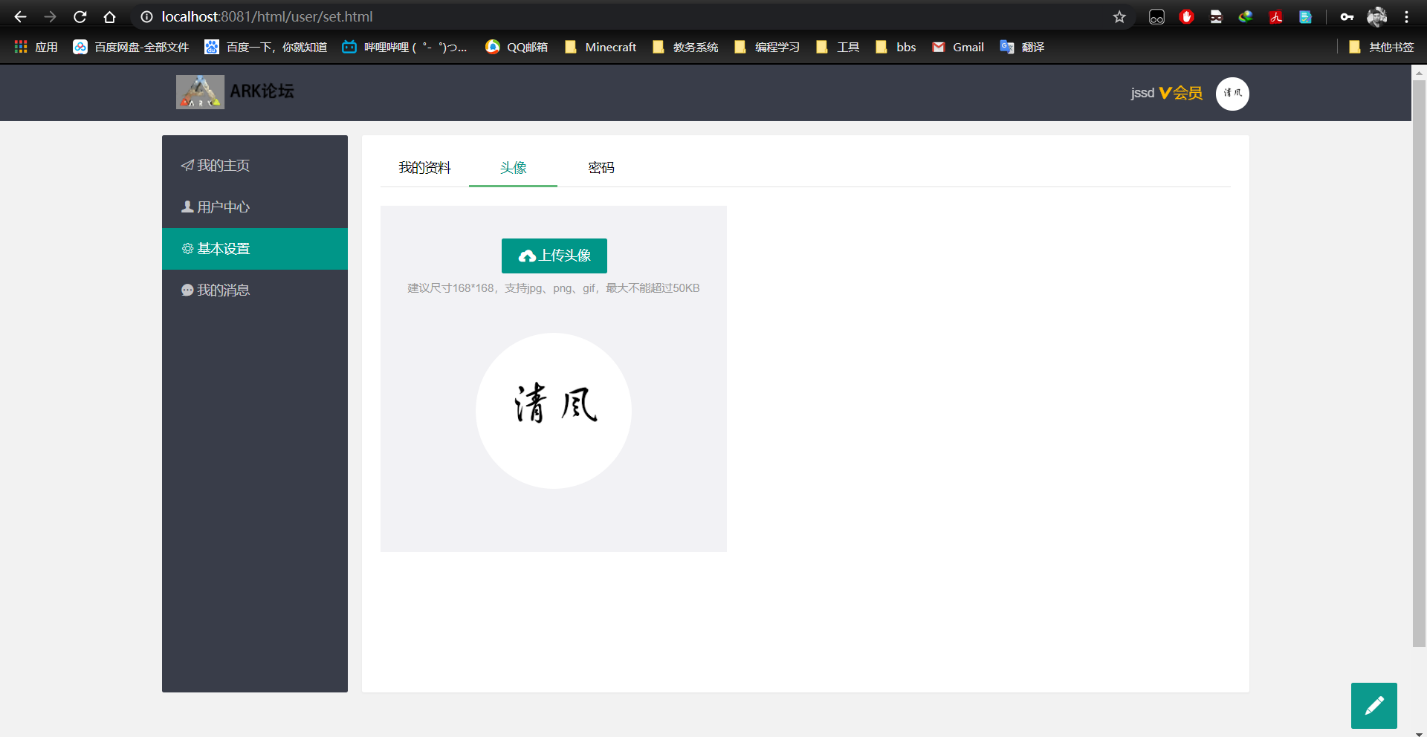
点击基本设计导航，可进入用户信息修改页面。在用户信息修改页面，可以更改用户本身信息，比如修改密码，重新上传头像，修改用户名、性别等。具体如图5-20所示。

图5-20用户信息修改页面

### 5.2.8站内消息显示页面

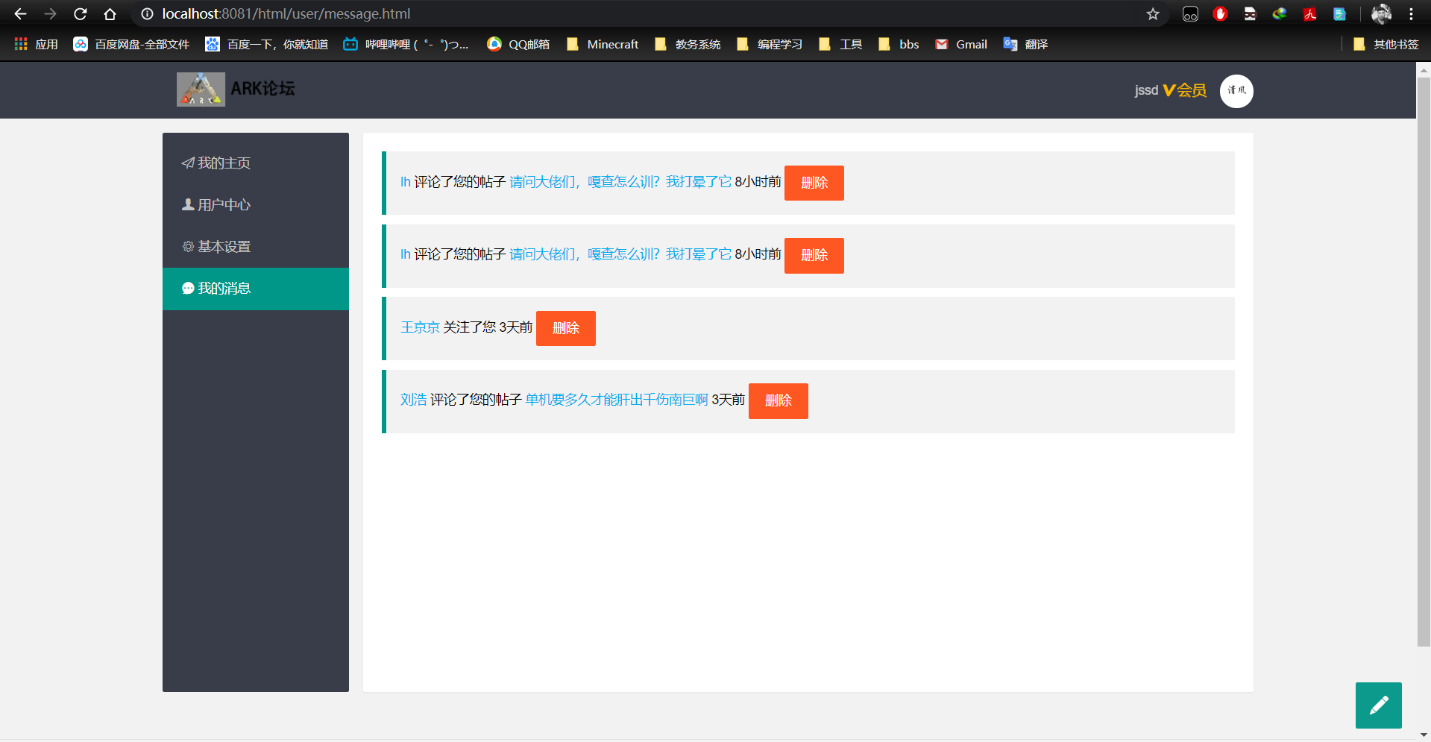
当一个用户评论了你发布的帖子，或者回复了你的回帖信息，亦或者其他用户关注你，都会触发系统产生一封站内信，当你点击我的消息选项时，会展示站内信息。具体如图5-21所示。

图5-21站内信息展示页

# 结束语

本系统的设计与实现采用了前后端分离、分布式开发的设计思想，主要采用了LayUI前端框架、SSM后端框架、rpc远程部署等技术，开发了一个B/S体系结构的小型论坛系统。通过查阅大量的网络资料，在进行市场调研以及需求分析之后，我明确的了解到一个小型轮系论坛系统需要具备的基本功能。之后，我进行了详细的业务需求分析、用户需求分析、功能需求分析等，一步一步的确认了整个系统的需求，为程序开发奠定了充分的基础。在对系统的分析和设计的过程中，从系统部分设计到系统整体结构设计，从系统的每个业务模块流程图示到数据库设计，每一个模块的设计与实现都充分运用了学到的知识，使得系统功能更为完备；在系统实现中过程中，每一个模块的功能都进行了相关的实现，不同的用户角色对应自己具备的权限，保证了系统的正常运行，完成了一个简单便捷的小型游戏论坛系统。

当然，这个小型游戏论坛还有很多不足之处, 例如用户登录模块不够安全，存在漏洞；在查询帖子的时候查询效率低下，占用大量的系统资源；不能详细的按照文章信息搜索文章；页面设计不美观、影响用户体验等问题。后续我会对系统进行优化重构，希望该系统可以运用到生活之中。

# 参考文献

[1]刘广升. 基于Java的BBS论坛系统的设计与实现[D].山东大学,2013.

[2]萍. 中国游戏网络媒体现状研究[D].南昌大学,2013.

[3]林,陈盈西,鲍正德,刘怡亨,余万. 浅谈国内BBS论坛的现状、发展与管理[J]. 电脑迷,2018(06):202.

[4]张李进. 网络化生存之游戏社区[D].华中师范大学,2011.

[5]张国平 , 李亚丽 , 马丽 . 基于 Struts 框架的网上论坛 BBS 系统设计与实现 [J].电脑知识与技术 ,2011,7(34):8846-8848.

[6]程婷婷. 公共领域视域中的网络论坛分析[J]. 创新,2011,5(05):110-113

[7]赵永亮,唐锡晋. 基于天涯论坛的BBS在线行为分析[J]. 系统科学与数学,2015,35(02):129-141.

[8]赵康,谢静,李璐. 我国网络论坛发展状况分析[J]. 科技传播,2014,6(02):273-275+102.

[9]韩劢. 大学生网络论坛的心理分析[J]. 广州大学学报(社会科学版),2010,9(03):73-77.

# 致谢

历经近三个月的时间，我终于实现了整个小型游戏论坛系统，这是一个曲折又令人开心的过程。在这个过程中，我有过苦恼，有过bug修不好的难受，当然，也有修好bug之后的喜悦。我更加明白了一名计算机程序从业人员的不容易。很多细节方面的东西，不自己手动编写，不能深刻的体会到。自己手动编写了，才体会到其中的困难和不容易。

在大学里面学习的知识大部分都是理论知识，一些知识再用不到的时候，不能深刻的理解和体会，不能融会贯通。当自己动手实现一些东西的时候，才能更加理解那些知识。我非常感谢大学中我们信息学院的所有老师，没有你们兢兢业业的传授我们知识，就没有现在充满自信的我们。还要感谢所有帮助过我的人，感谢我的舍友，在我遇到问题时，一起交流，分享解决方案。

最后再次真诚的感谢我的指导老师常淑惠老师，在她的认真负责的指导下，我的小型游戏论坛系统才能正常的完工。