“开源软件杯”创新作品大赛

NANU系统

项目开发报告

项目仓库地址：https://github.com/ACIUvU/nanuAPP

技术小组 第七小组

小组成员  **陈佳璟（2017051604063）**

**姜弈昕（2017051604067）**

**王琛越（2017051604072）**

**郑梓航（2017051604065）**

**周瑜 （2017051604111）**

指导教师  **龚伟**

起止时间 2020年2月 ～ 2020年6月

重庆师范大学软件工程系

摘要

本系统开发的目的是开发一个可以随时分享查看喜欢的内容的平台。为用户提供他真正感兴趣的内容，充分利用生活的碎片化时间。本系统开发了安卓端和网页端，安卓端采用Android Studio进行编程，网页端采用MVC模式，选用了当今流行的ThinkPHP框架，具有开发效率高，设计灵活，界面友好等特点，采用了JAVA，PHP，HTML等语言。

**关键词：**MVC模式，数据库，ThinkPHP

**Abstract：**The purpose of this system development is to provide users a platform to see and share the interesting staff in life they find or they are in interested in.The content is the things that the user truly interested in,and it will help users to manage their fragment time. This system develop Android platform and Web platform.The Android platform adopts Android Studio to write code,and the Web side adopts MVC mode and adopts the popular ThinkPHP framework. It has the characteristics of high development efficiency, flexible design and friendly interface. The database used in this system is MySQL, the system uses JAVA,PHP, HTML and other languages.

**Key words：**MVC pattern;JAVA; ThinkPHP

目录

[一、开发背景 5](#_Toc290135713)

[二、任务概述 5](#_Toc290135713)

2.[1 运行环境 5](#_Toc290135714)

[2.2 用户特征 5](#_Toc290135715)

[三、需求分析 5](#_Toc290135713)

3.[1 需求规定 5](#_Toc290135714)

3.[2 功能需求 6](#_Toc290135715)

3.[2.1 APP端用户用例图 7](#_Toc290135714)

3.2.[2 管理员用例图](#_Toc290135715) 9

[3.2.3 Web端用户用例图 1](#_Toc290135716)0

[3.3 性能需求 1](#_Toc290135716)1

3.[3.1 精度 11](#_Toc290135714)

3.3.[2 时间特性要求 11](#_Toc290135715)

[3.3.3 灵活性 1](#_Toc290135716)1

3.3.[4 输入输出要求 11](#_Toc290135715)

[3.3.5 故障处理 12](#_Toc290135716)

[四、系统基本设计 1](#_Toc485717919)2

4.[1 预期功能 1](#_Toc485717920)2

4.[2 数据库表单初步设计 1](#_Toc485717921)2

[五、系统架构及功能模块 1](#_Toc485717923)3

5.[1 系统架构流程图 1](#_Toc485717924)3

5.[2 系统静态模型 1](#_Toc485717922)3

5.[3 系统动态模型 1](#_Toc485717922)4

5.[3.1 用户子系统顺序图和活动图 1](#_Toc290135714)4

5.3.[2 管理员子系统顺序图和活动图 1](#_Toc290135715)7

5.4 [功能模块 1](#_Toc485717925)7

[六、数据库设计 1](#_Toc485717926)8

6.[1 E-R图 1](#_Toc485717927)8

6.[2 表单设计 1](#_Toc485717927)8

[七、系统实现及系统测试 1](#_Toc485717928)8

[7.1 APP端的实现 1](#_Toc485717929)8

7.1.[1 登录注册模块 1](#_Toc485717930)8

7.1.[2 个人信息模块 2](#_Toc485717931)0

7.1.[3 文章管理模块 2](#_Toc485717932)1

[7.1.4 小组模块 2](#_Toc485717933)3

[7.1.5 正在测试的模块 2](#_Toc485717933)3

[7.2 网页端的实现 2](#_Toc485717933)5

7.2.[1 登录注册模块 2](#_Toc485717934)5

7.2.[2 文章管理模块 2](#_Toc485717935)7

7.2.[3 小组模块 3](#_Toc485717936)1

7.2.[4 待实现模块 3](#_Toc485717937)2

[7.3 测试部分总结 3](#_Toc485717933)2

[八、系统总结 3](#_Toc485717943)2

[九、附录 3](#_Toc485717947)3

## **开发背景**

NANU APP提供的这样一个平台，让我们看见这个相同的世界不同的一面，或是一个全新的世界。在这里大家可以匿名分享出自己喜欢的内容：影视作品、音乐、图书、美食、作家、科学家、各类观点......帮你发现这个世界不一样的精彩的一面，同时可以建立和加入不同的小组找到自己志同道合的人一起分享。

# 二、任务概述

## 2.1 运行环境

### （1）WEB端

编程语言：PHP，采用ThinkPHP框架

数据库：MySQL

软件开发环境：EPP3，PHPStudy

绘图工具：Enterprise Architect

### （2）Android端

编程语言：JAVA

数据库：SQLite

软件开发环境：Android Studio(安卓开发虚拟机)

绘图工具：Enterprise Architect

## 2.2 用户特征

系统的使用用户主要为想要分享文章的用户。

# 三、需求分析

## 3.1 需求规定

Web端：

用户：查看文章

管理员：登陆、注册、发布文章、修改文章、创建小组

Android端：

用户：发布文章、搜索文章、点赞文章、修改文章、关注用户、意见反馈、查看小组、加入小组、退出小组、点赞小组、登陆、注册、退出登陆、修改信息

管理员：发布文章、修改文章、创建小组、登陆、注册



图3-1用例图

## 3.2 功能需求

NANU APP和网页的参与者为用户和管理员，如图所示。

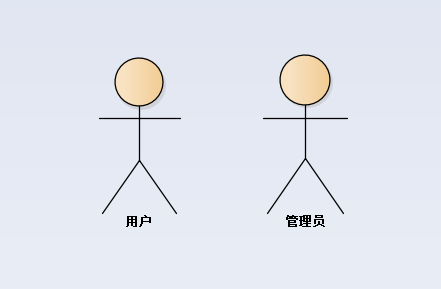


图4-1系统参与者

### 3.2.1 APP端用户用例图

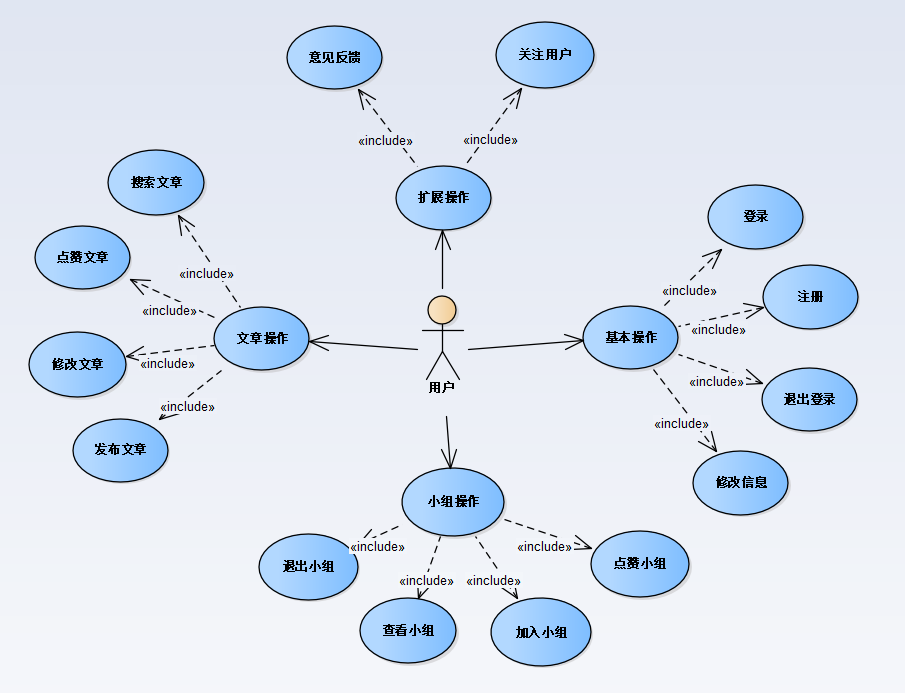


图4-2APP端用户用例图

### （1）用例：编辑个人信息

|  |  |
| --- | --- |
| **用例** | 编辑个人信息 |
| **用例叙述** | 用户可以编辑自己的基本信息 |
| **假设条件** | 1.系统提供了“个人信息”用户界面  2.系统中有一个个人信息数据库 |
| **前置条件** | 用户已成功登陆系统 |
| **后置条件** | 系统数据库接收用户编辑的记录 |
| **步骤序列** | 1.用户成功系统，进入系统首页  2.系统显示用户信息  3.用户对个人信息进行编辑 |
| **受益参与者** | 用户 |

### （2） 用例：发布文章

|  |  |
| --- | --- |
| **用例** | 发布文章 |
| **用例叙述** | 用户可以通过系统发布文章 |
| **假设条件** | 1.系统提供了用户发布文章界面  2.系统中有一个文章数据库 |
| **前置条件** | 用户已成功登陆系统 |
| **后置条件** | 系统数据库接收用户发布的文章的记录 |
| **步骤序列** | 1.用户成功系统，进入系统首页  2.用户选择发布文章 |
| **受益参与者** | 用户 |

### （3）用例：加入小组

|  |  |
| --- | --- |
| **用例** | 加入小组 |
| **用例叙述** | 用户可以通过系统加入小组 |
| **假设条件** | 1.系统提供加入小组界面  2.数据库中有用户加入的小组属性 |
| **前置条件** | 用户已成功登陆系统 |
| **后置条件** | NULL |
| **步骤序列** | 1.用户成功系统，进入系统首页  2.用户选择加入小组 |
| **受益参与者** | 用户 |

### （4）用例：查看文章

|  |  |
| --- | --- |
| **用例** | 查看文章 |
| **用例叙述** | 用户可以通过系统查看文章 |
| **假设条件** | 1.系统中有文章数据库 |
| **前置条件** | 用户已成功登陆系统 |
| **后置条件** | NULL |
| **步骤序列** | 1.用户成功系统，进入系统首页  2.用户选择查看文章 |
| **受益参与者** | 用户 |

## **3.2.2 APP端和WEB端管理员用例图**

## IMG_256

**图4-3 管理员用例图**

### （1）用例：创建小组

|  |  |
| --- | --- |
| **用例** | 创建小组 |
| **用例叙述** | 管理员可以通过系统创建小组 |
| **假设条件** | 1.系统提供了管理小组界面 |
| **前置条件** | 1.管理员已成功登陆系统  2.管理员已进入管理小组界面 |
| **后置条件** | 系统数据库接收管理员添加的记录 |
| **步骤序列** | 1.管理员成功登陆系统，进入系统界面  2.管理员选择添加小组模块  3.系统显示添加小组模块 |
| **受益参与者** | 管理员 |

### （2）用例：删除文章

|  |  |
| --- | --- |
| **用例** | 删除文章 |
| **用例叙述** | 管理员可以通过系统删除文章 |
| **假设条件** | 1.系统中有文章数据库 |
| **前置条件** | 1.管理员已成功登陆系统  2.管理员已进入管理文章界面 |
| **后置条件** | 系统数据库接收管理员删除的记录 |
| **步骤序列** | 1.管理员成功登陆系统，进入系统界面  2.管理员选择文章管理模块  3.系统显示文章管理模块  4.管理员选择删除文章 |
| **受益参与者** | 管理员 |

### （3）用例：编辑个人信息

|  |  |
| --- | --- |
| **用例** | 编辑个人信息 |
| **用例叙述** | 管理员可以通过系统编辑个人信息 |
| **假设条件** | 1. 系统提供了“个人信息”用户界面 2. 系统中有个人信息数据库 |
| **前置条件** | 1.管理员已成功登陆系统  2.管理员已进入信息管理界面 |
| **后置条件** | 系统数据库接收管理员修改的记录 |
| **步骤序列** | 1.管理员成功登陆系统，进入系统界面  2.系统显示管理员信息  3.管理员修改个人信息 |
| **受益参与者** | 管理员 |

## **3.2.3 WEB端用户用例图**

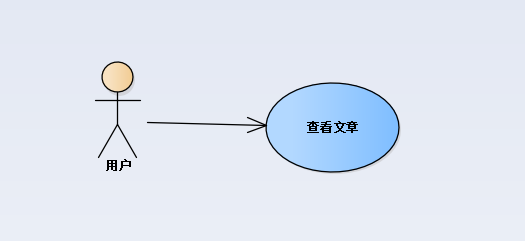


图4-4 用户用例图

### （1）用例：查看文章

|  |  |
| --- | --- |
| **用例名称** | **查看文章** |
| 参与者 | 用户 |
| 用例描述 | 该用例描述一个系统用户查看文章的行为 |
| 触发器 | 当用户点击查看文章时，用例触发 |
| 前置条件 | 需要完成登录操作 |
| 后置条件 | 如果数据库中有文章，则显示； |
| 基本事件流 | 1. 用户点击查看文章 2. 系统验证用户信息做出响应； 3. 用户点击不同小组的文章； 4. 系统从数据库中按小组分类进行显示； 5. 用户点进具体文章进行查看 |
| 扩展事件流 | A-4 该小组内无文章则不显示 |
| 结论 | 当用户查看到文章时，用例结束 |
| 数据需求 | D-1文章信息包括标题、点赞数、内容、作者、文章简图、小组号 |
| 业务规则 | 无 |

# 3.3 性能需求

## 3.3.1 精度

数量值：精确到小数后一位；

时间值：精确到秒，并以yyyy/mm/dd aa:bb:cc的形式描述；

搜索查找时不区分大小写；

搜索时显示与搜索相关字有关的文章

### 3.3.2 时间特性要求

该软件的时间特性要求如下：

a、响应时间：不超过10秒；

b、更新处理时间：不超过15秒；

c、数据的转换和传送时间：基本达到实时；

d、数据备份时间：2小时内

### 3.3.3 灵活性

a、操作方式上的变化：

该软件主要以手机APP形式进行操作；

b、运行环境的变化：

主要用于Android环境使用；

C、同其他软件的接口的变化：

在尽量保证接口不出现变动的情况下，允许接口的重载和再定义。接口命名规则必须是统一的。

d、精度和有效时限的变化：

精度在必须调正的情况下，可以上下浮动10%；有效时限则根据现实情况允许稍大范围的变化

e、计划的变化或改进。

当需求发生变化时，系统要有一定的适应能力，能为将来版本的变更提供支持。能在系统变更数据库和用户界面后，系统的现有设计和编码能最大程度的重用。整性约束。

### 3.3.4 输入输出要求

初始数据的输入：用户注册信息及发布的文章符合相关要求

输出：用户搜索时显示与搜索相关字相关的文章

### 3.3.5 故障处理

a、能够对用户输入的数据进行检查校验

b、能够及时发现系统在运行时的错误信息，并给出相应的提示，系统应具有一定的容错能力

c、数据库具有灾难备份机制，遇到意外中断的情况，也能在故障排除后恢复工作

d、定期对存储设备进行维护，以免造成损失

# 四、系统基本设计

## 4.1 预期功能

根据不同端的用户角色设定了不同的功能

网页端用户：查看文章

网页端管理员：发布文章、修改文章、创建小组、登陆、注册

APP用户：发布文章、搜索文章、点赞文章、修改文章、关注用户、意见反馈、查看小组、加入小组、退出小组、点赞小组、登陆、注册、退出登陆、修改信息

APP管理员：发布文章、修改文章、创建小组、登陆、注册

## 4.2 数据库表单初步设计

用户表（用户名、密码、头像、用户介绍、邮箱、用户昵称）

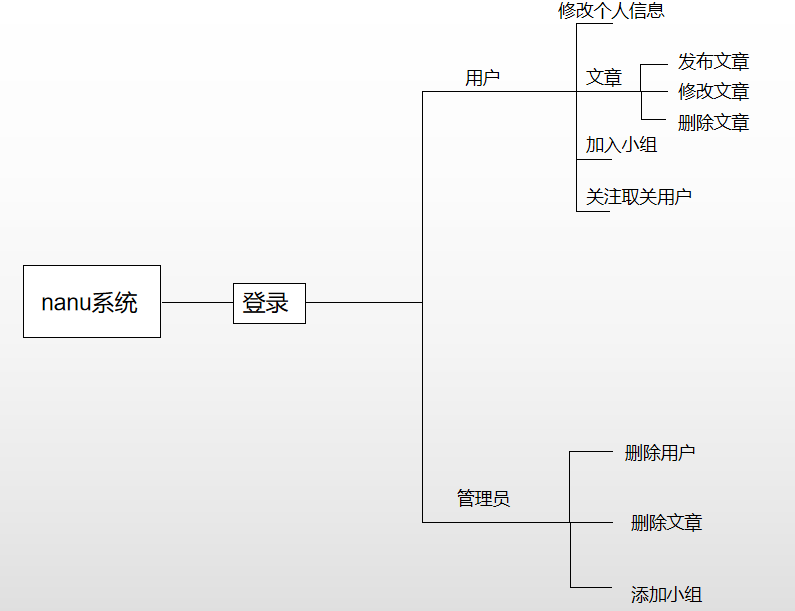
管理员（用户名、密码）

文章表（标题、点赞数、内容、作者、文章简图、小组号）

小组表（小组号、成员、小组名、点赞数）

# 五、系统架构及功能模块

## 5.1 系统架构流程图



点赞小组

图5-1 APP端系统架构流程图

### 5.2 系统静态模型

类模型是面向对象方法的核心，类模型从对象的角度描述系统的组成，描述对象及相互间的关系。系统已在前面进行过简单描述，通过检查问题陈述中的所有名词短语和代词，得到初始类：

如图5-2所示：

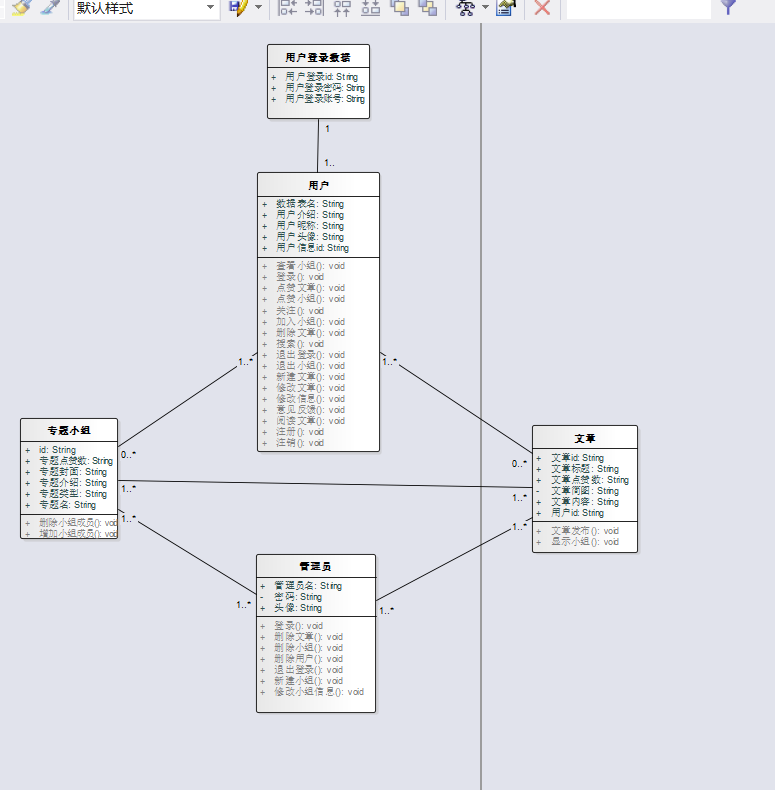


图5-2 类图

## 5.3 系统动态模型

系统的动态模型可以使用顺序图和活动图来进行描述。

**5.3.1 用户子系统顺序活动图**

## B`Y6%_I21G)JT0GMH593M)H

图5-3用户注册活动图

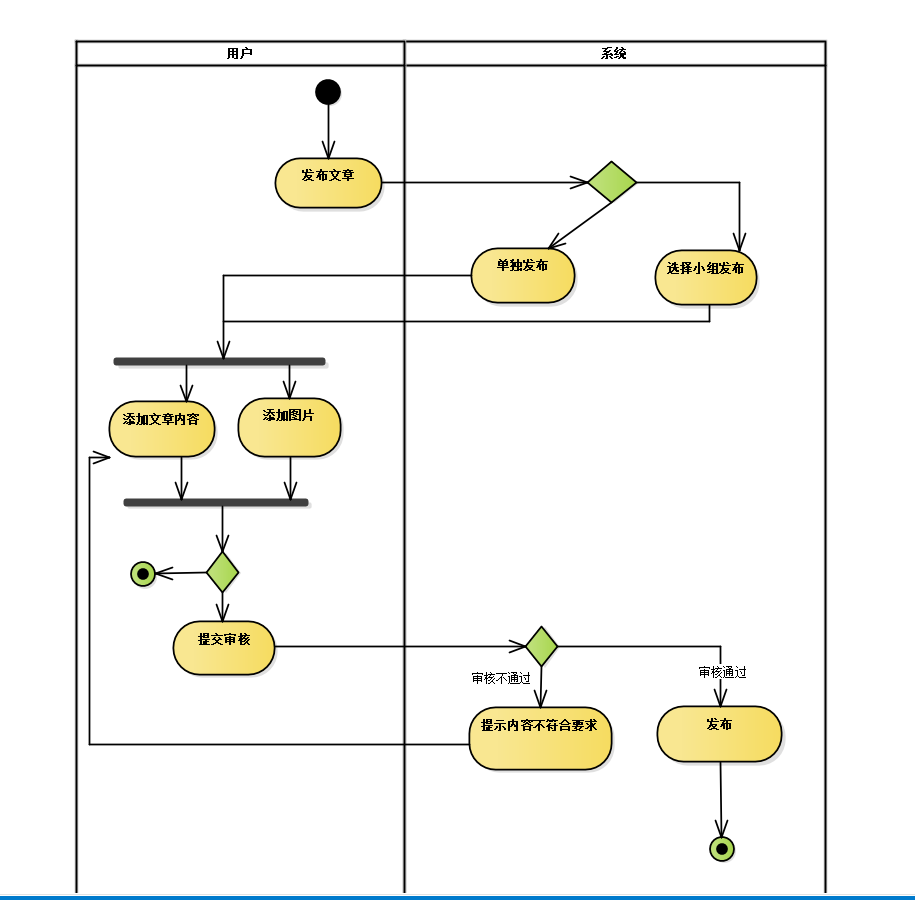


图5-4 用户发布文章活动图

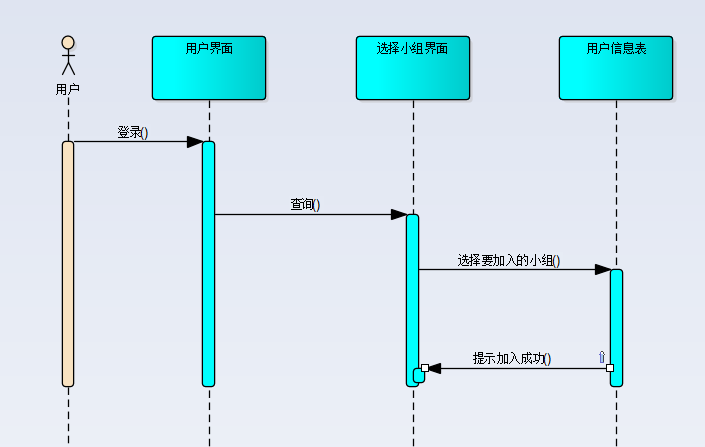


图5-5 用户加入小组顺序图

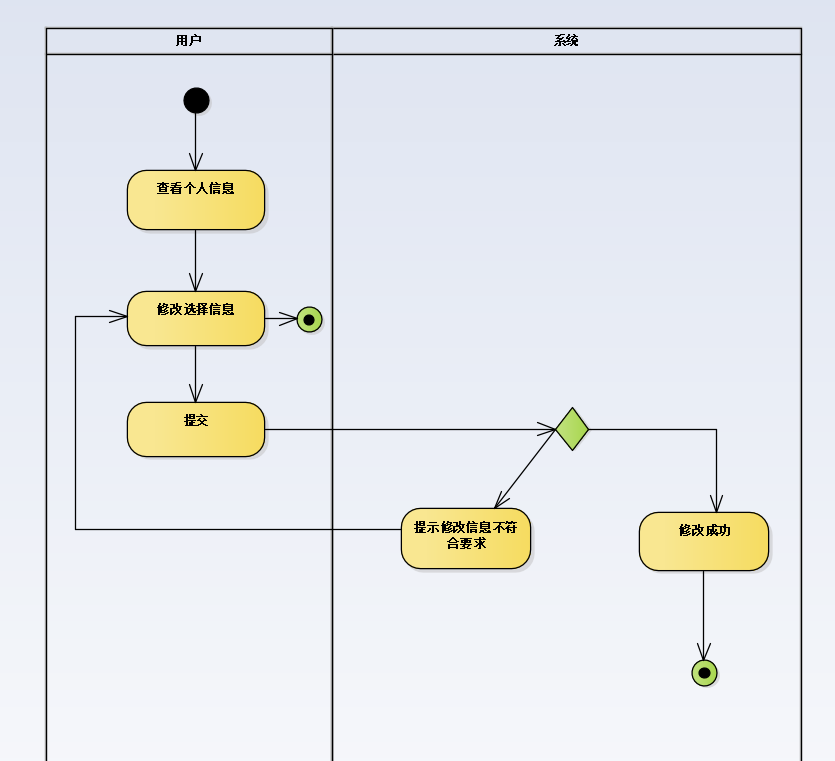


图5-6 用户修改个人信息活动图

### 5.3.2 管理员子系统顺序活动图

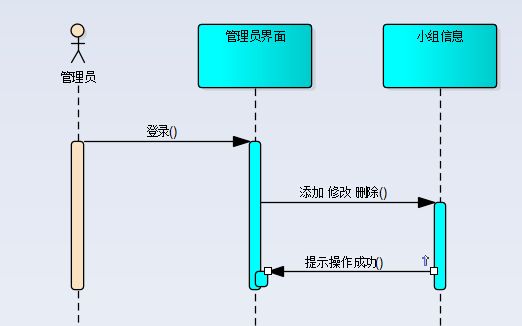


图5-7管理员管理小组信息顺序图

### 

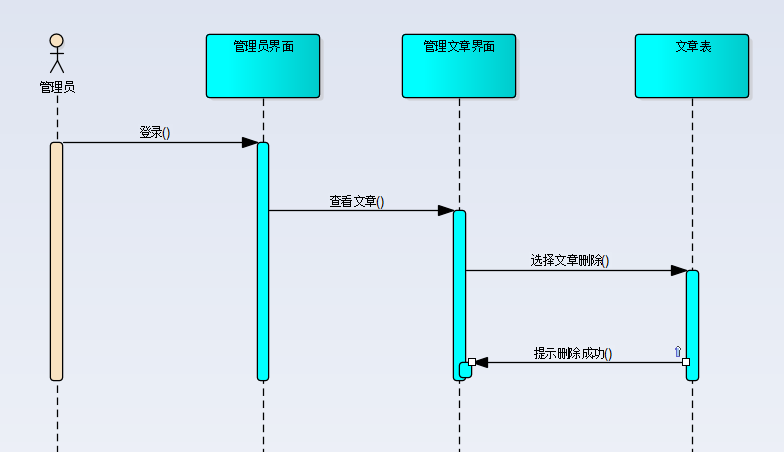


图5-8管理员管理文章顺序图

## 5.4 功能模块

用户：查看文章、发布文章、修改删除文章、加入小组、关注用户

管理员：管理小组、删除文章、删除用户

# 六、数据库设计

## **6.1 E-R图**

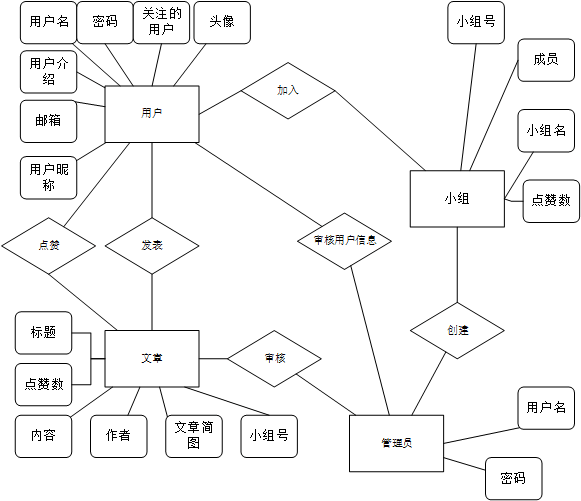


图6-1 E-R图

## 6.2 表单设计

登录信息（用户id、用户账号、密码）

用户信息（用户id、用户头像、用户昵称、用户介绍、关注的用户id）

文章信息（文章id、用户id、小组id、文章标题、文章内容、文章点赞数、文章图片名、文章图片地址、日期）

小组信息（小组id、小组名、小组封面、小组类型、小组介绍、小组点赞数）

# 七、系统实现及系统测试

## 7.1 APP端实现

### **7.1.1 登录与注册模块**



图7-1 注册成功



图7-2 用户名不存在登录失败

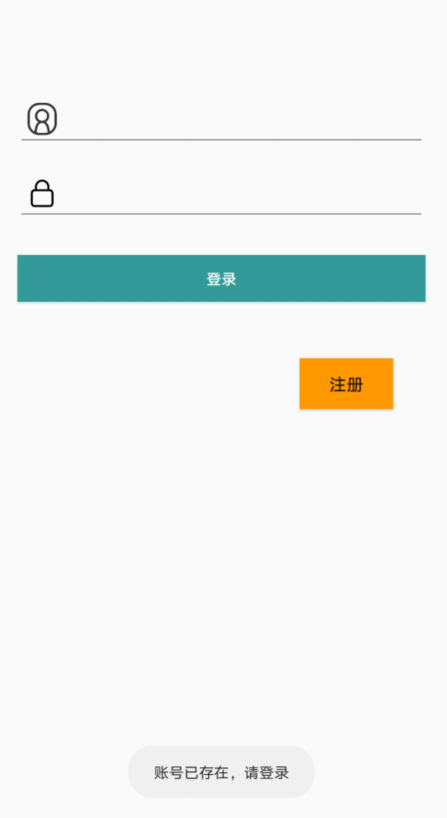


图7-3 重复注册账号

**7.1.2 个人信息管理模块**



图7-4 基本操作界面



图7-5 修改个人信息



图7-6 意见反馈

**7.1.3 文章管理模块**

  图7-7 文章界面 图7-8 搜索文章



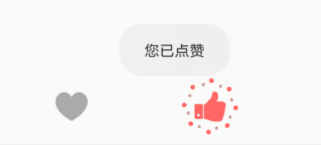
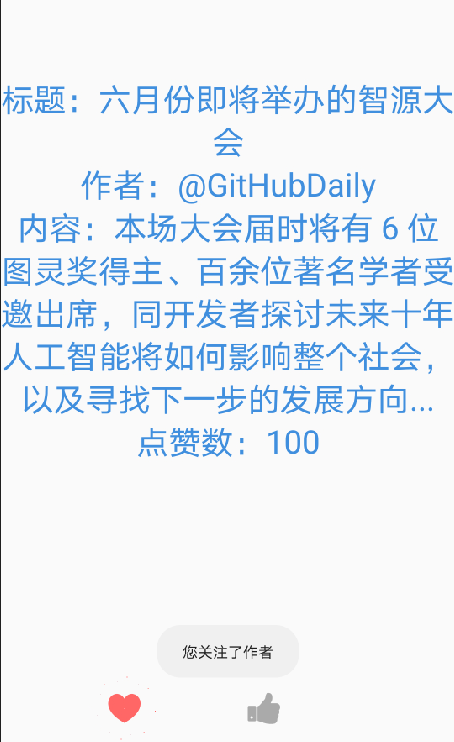


图7-9 发布点赞文章

**7.1.4 小组模块**

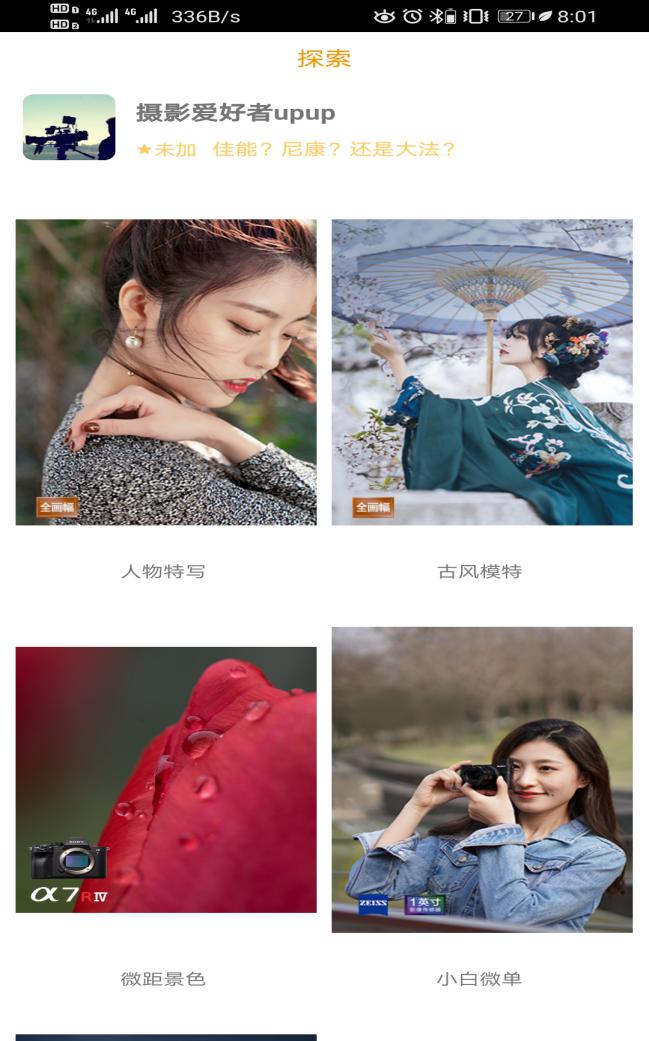
 

图7-9 小组界面

**7.1.5 正在测试的模块**

管理员操作的相关部分

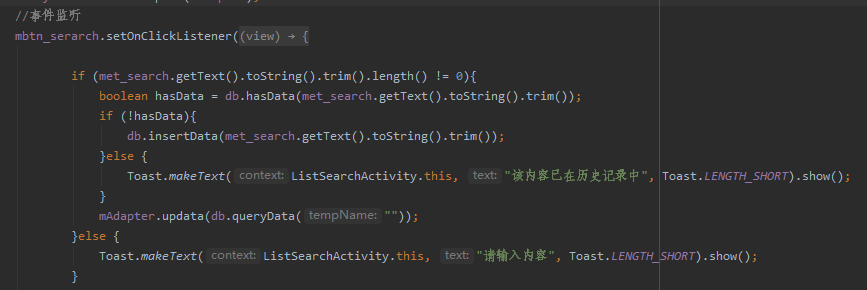
发布文章时提供部分图像处理功能

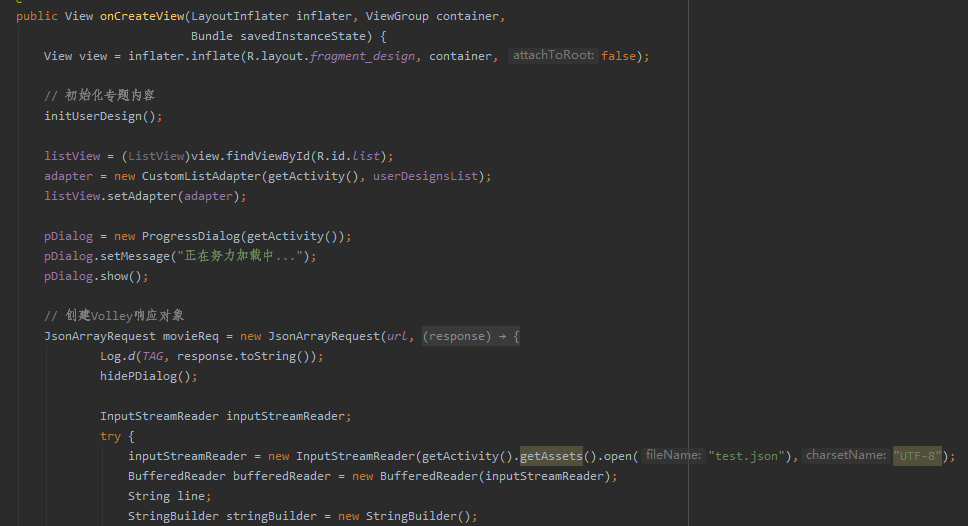
  

图7-10 图像处理

实现关键代码：







**7.2 网页端实现**

**7.2.1 登录注册模块**

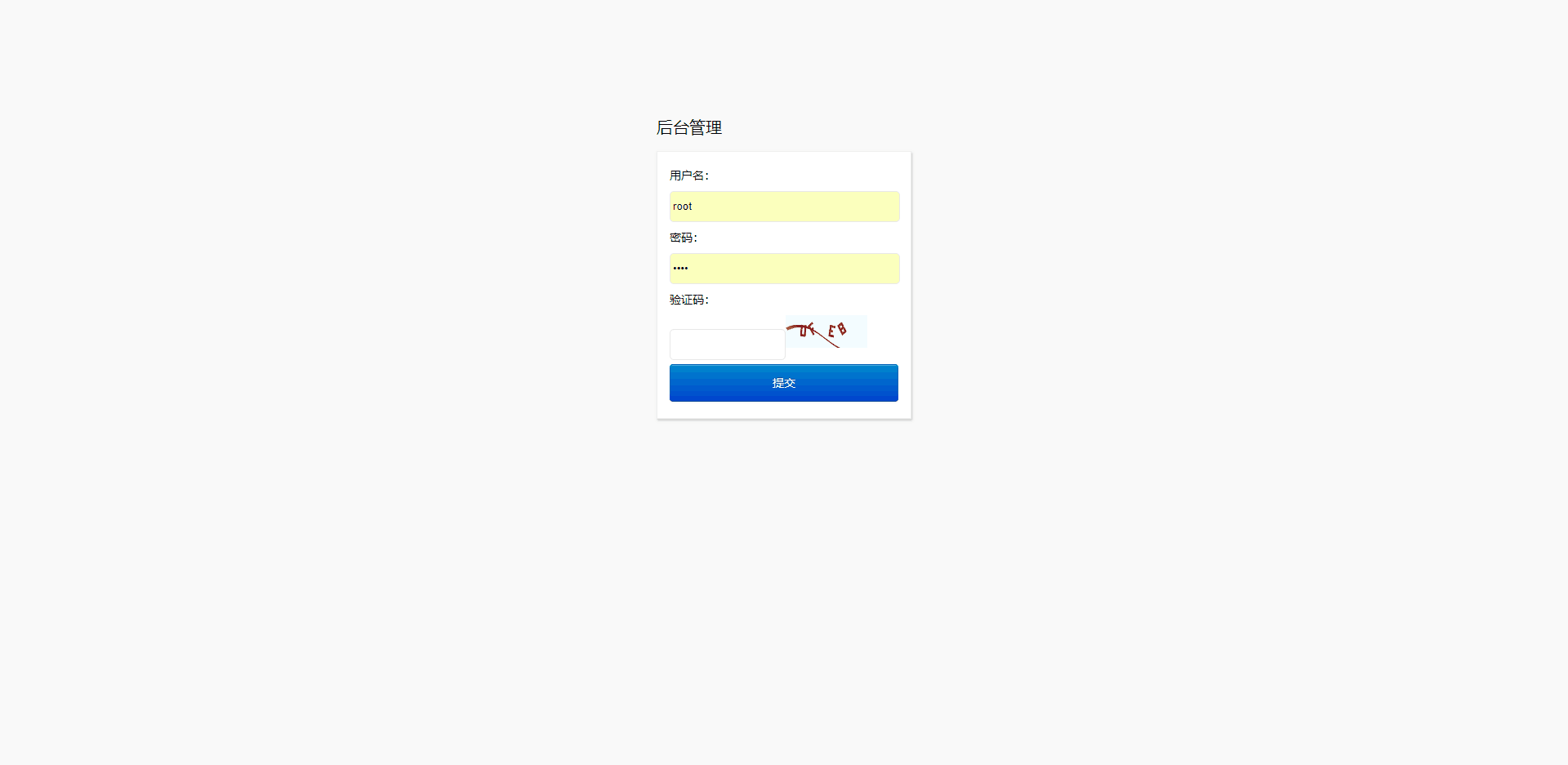


图7-11 登录

# HU_~`3~_@YQCGCWX~7EQN6S

图7-12 注册

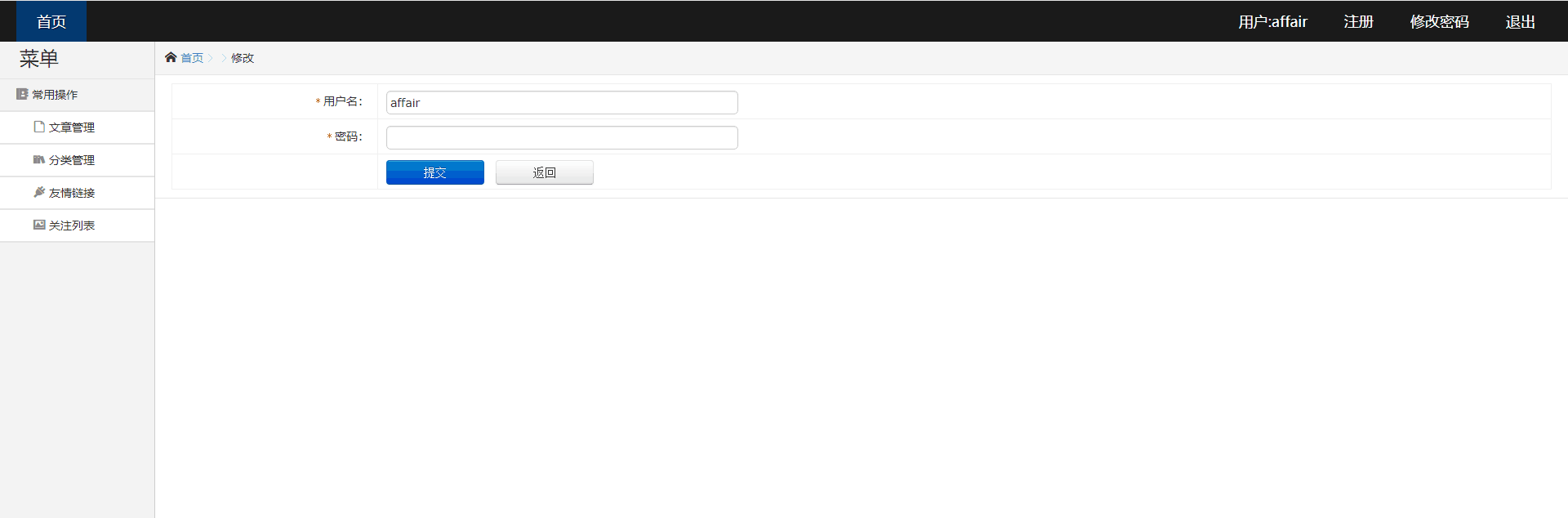
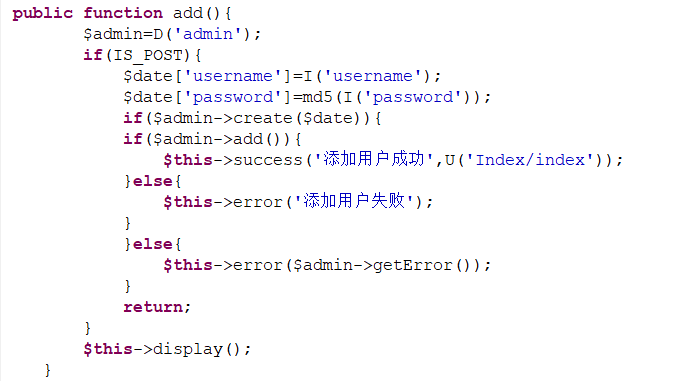


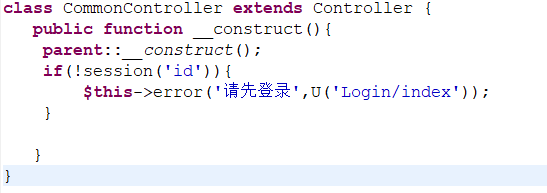
图7-13 修改密码

实现关键代码：









未登录无法进行相应操作

**7.2.2 文章管理模块**

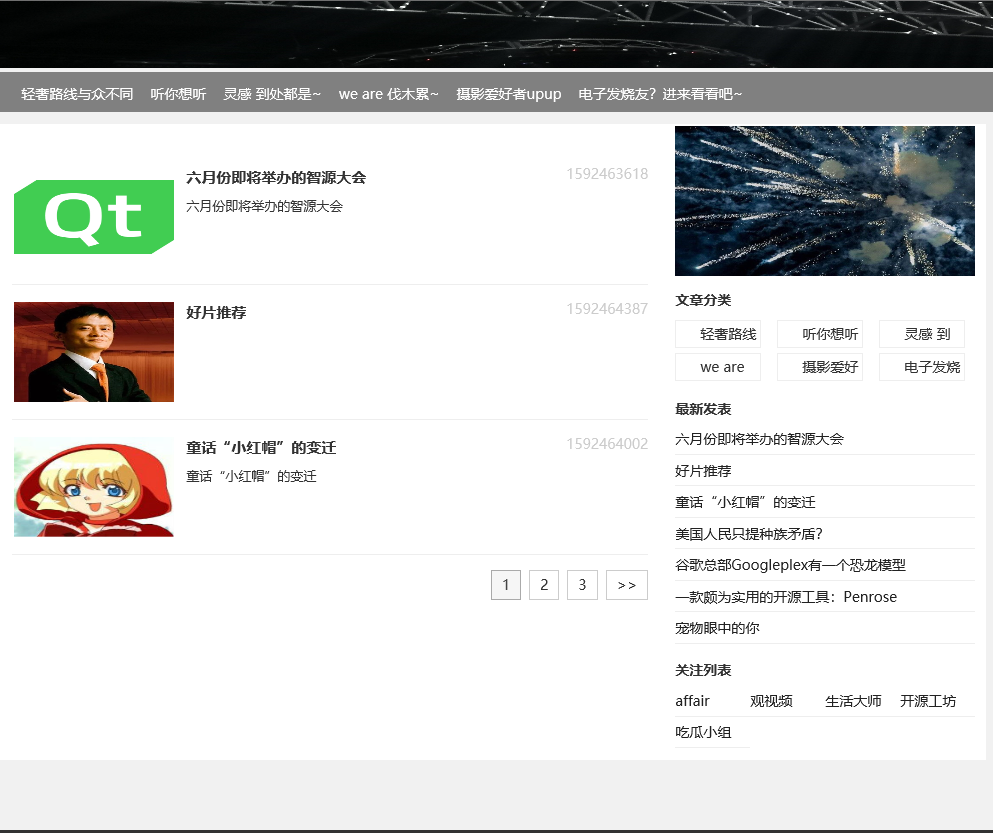


图7-14 查看文章界面



图7-15 一个小组里的文章

# {NFD7@5ZDP{4Q2Y3@2WXM[O

图7-16 修改文章

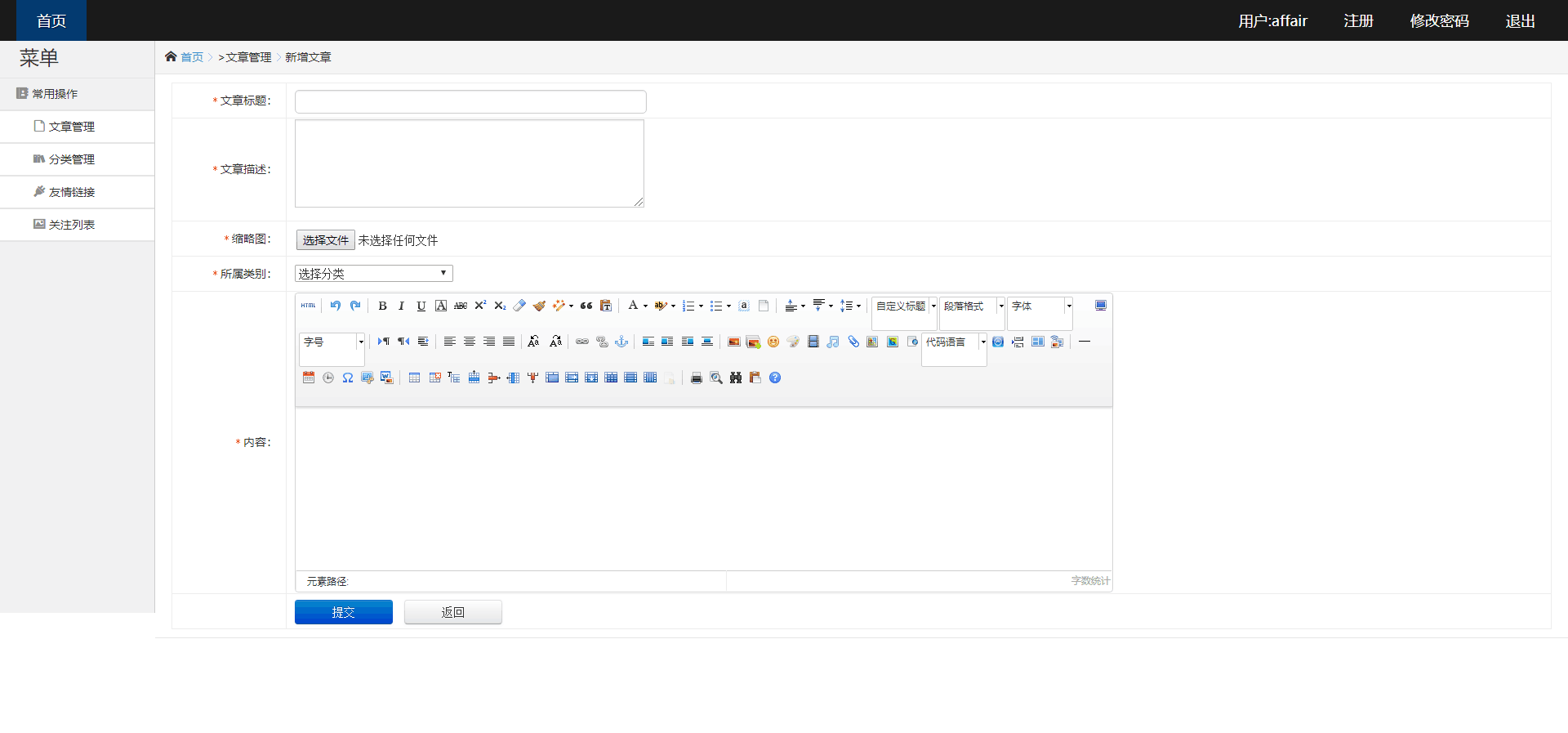
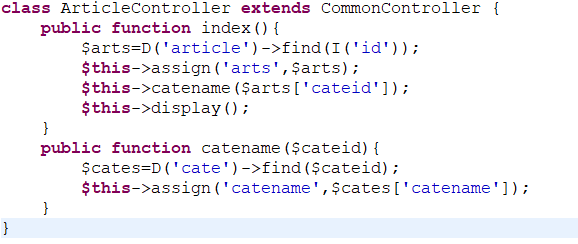
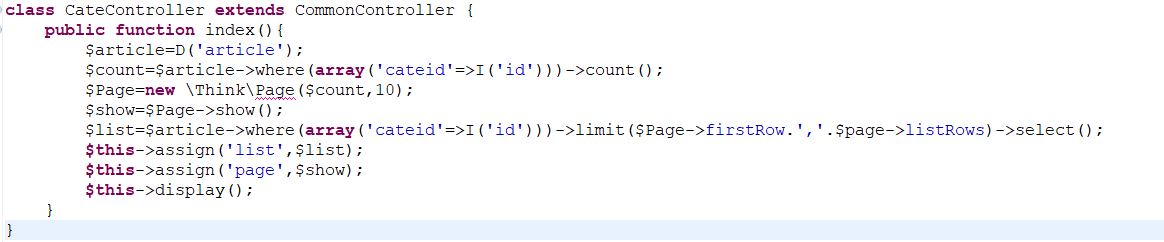


图7-17 发布文章

# 实现关键代码：



实现按小组分类展示文章



分别展示每个小组内的文章



发布新文章

# PYUOHY0FY0S1WM~LS[RTH`D

修改文章页面调用数据库之前文章的内容

# 7.2.3 小组模块

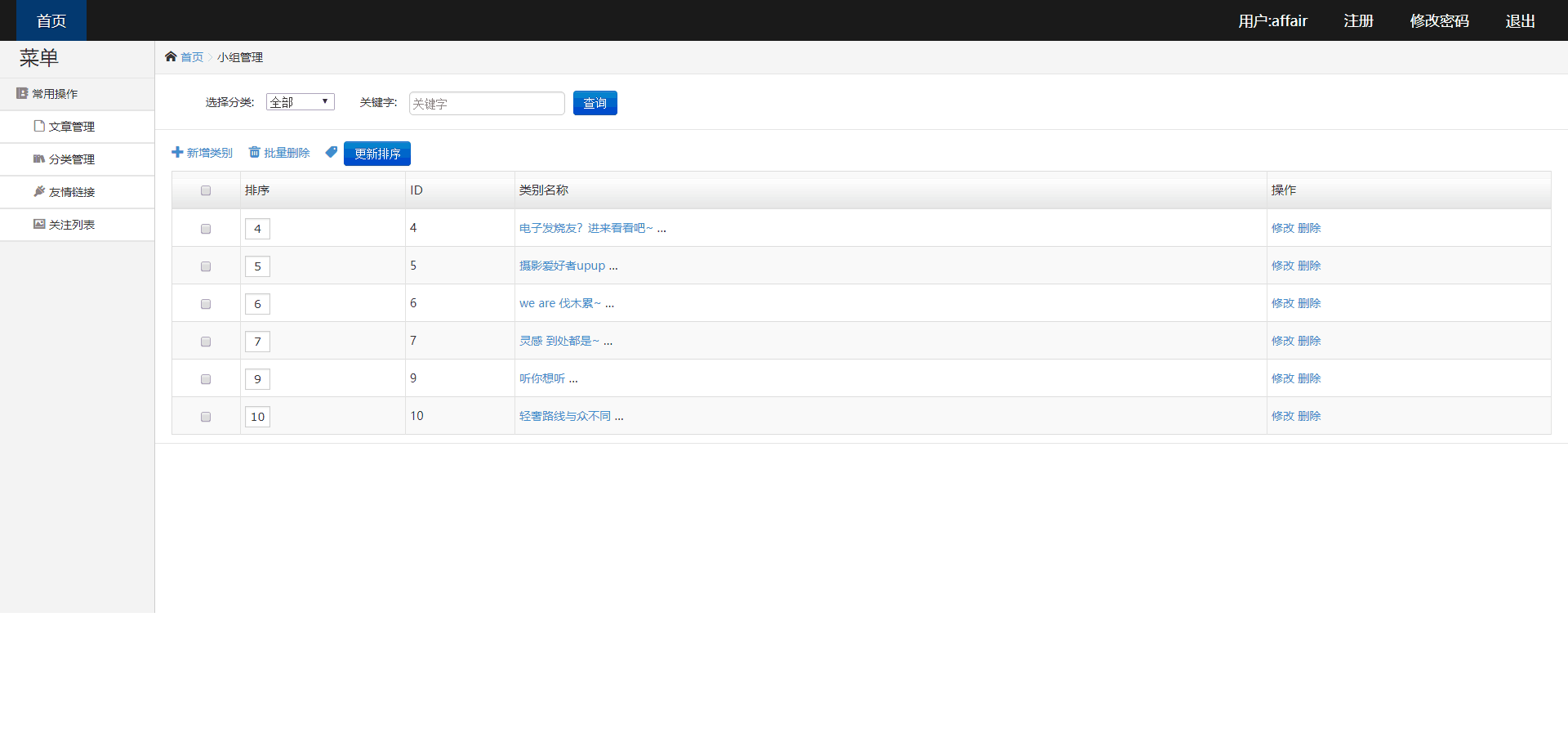


图7-18 小组管理

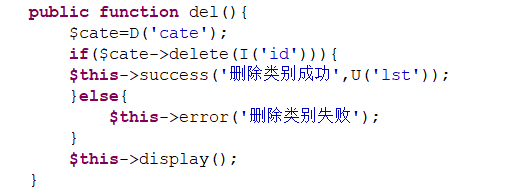
关键代码实现：



添加小组



修改小组



删除小组

**7.2.4 待实现模块**

关注用户

点赞文章

管理员相关操作

**7.3 测试部分总结**

|  |  |
| --- | --- |
| **其他重要测试** | **测试目的** |
| **接口正确性测试** | 检查系统能否与外部接口正常工作，检查事件之间的连接是否正确 |
| **数据库测试** | 调用各个数据库访问方法和进程，并在其中填充有效的和无效的数据或对数据的请求。确保数据按预期的方式填充，并且所有数据库事件都按正常方式出现；或者检查所返回的数据，确保检索到了正确的数据 |
| **性能测试** | 提取系统性能数据，检查系统是否满足在需求中所规定达到的性能 |

测试风险：

本次测试过程中，可能出现的风险如下：

1.开发提交测试版本比该计划延迟，发生此种情况时，执行测试的时间应该合理顺延；

2.如果测试计划执行过程中出现需求变更超出预计的情况，可能导致编写测试用例和执行测试相关工作量增加，测试进度延迟；

3.如果开发提测版本质量较低，bug较多，可能导致比该计划更多轮次的回归测试；

4.人员调整、人员经验以及对软件的熟悉度可能会影响测试计划；

5.测试需依赖于服务器环境，如果环境不稳定，如代码编译不通过、数据库异常等情况，可能会影响测试进度。

# 八、系统总结

通过本次对该系统的设计与实现，使我对整个系统软件开发流程有了更深一步的理解。每一步的分析都是必不可少的。要重视每一部分的开发分析，而不仅仅是最后的编码实现，前面对需求捕获架构设计是后面编码能顺利实现的基础。在实现的过程中，会遇到很多问题，使得预想的运行结果运行不出来，可以搜索系统报的错来看解决办法，或者检查一下代码，很多时候是自己的错误造成运行报错。整个开发团队的沟通也是项目成功与否的非常关键的因素，一定要保证项目中每个成员的工作进展，这次系统开发前期，由于上网课，很多时候缺乏沟通，导致组长很难监督其他组员工作进度，前期项目进展不顺。

本项目原定开发内容只有一个Android端的APP，后期经组内讨论加上的网页端。总体应是以APP为主，WEB为辅。所以在前期设计分析的时候重心都放在Android开发那边，而网页只安排了简单的功能。但在后期实现的时候，由于技术问题，Android开发并不顺利，没有把设计的功能都完成出来，导致整个项目分裂并且过于简单。

在以后的开发过程中，组长这个角色至关重要，应该慎重选择，要了解开发所用的的各种语言和工具，这样有利于督促组员及时上交阶段代码和对代码进行修改维护。而组员，也应该熟悉自己所要开发的产品和所要用到的语言和工具，否则无法解决开发过程中遇到的问题，并且应该把阶段的工作阶段完成，而不是堆到后期统一完成，后期就会因为工作量过多以及过多问题同时出现而不能解决，导致无法完成开发。

# 九、附录

### 参考文献：

1. Simon Bennett, Steve MoRobb, Ray Farmer.UML 2.2面向对象分析与设计[M].北京：清华大学出版社，2013.
2. 杨弘平，吕海华，李波，史江萍等.UML基础、建模与设计实践[M].北京：清华大学出版社，2012.
3. ThinkPHP3.2完全开发手册https://www.php.cn/course/1049.html