

福州大学

系统设计说明书

队 长： 梁晓键

队 员： 彭三福

队 员： 唐靖钧

队 员： 邹铭鸿

队 员： 王嘉泓

队 员： 陈凯琳

队 员： 林智信

队 员： 吕宇昕

队 员： 杨 泽

团队：知社

2020 年 04 月 10 日

目录

1.	引言.....	4
1.1.	编写目的.....	4
1.2.	读者对象.....	4
1.3.	参考文献.....	4
1.4.	术语与缩写解释.....	4
1.5.	文档约定.....	5
2.	总体设计.....	6
2.1.	系统运行环境设计.....	6
2.1.1.	硬件设备.....	6
2.1.2.	运行环境.....	6
2.1.3.	开发环境.....	6
2.2.	软件结构设计.....	6
2.2.1.	软件总体结构设计.....	6
2.2.2.	软件技术架构设计.....	9
2.2.3.	软件功能结构设计.....	9
2.3.	类图设计.....	11
3.	功能模块设计.....	12
3.1.	功能模块层次图.....	12
3.2.	功能模块设计.....	13
3.3.	功能模块描述.....	13
4.	接口设计.....	29
4.1.	接口设计规范.....	29
4.1.1.	请求格式.....	29
4.1.2.	响应格式.....	30
4.1.3.	认证方式.....	31
4.2.	用户服务.....	32
4.3.	社团管理.....	33
4.4.	申请与审核.....	34

4. 5.	公告.....	36
4. 6.	活动.....	37
4. 7.	活动论坛.....	38
4. 8.	活动评论.....	38
5.	系统安全和权限设计.....	38
5. 1.	安全性设计说明.....	38
5. 2.	权限设计说明.....	41

1. 引言

1.1. 编写目的

编写本系统设计说明书旨在描述本项目“知社”的系统结构和内部设计，内容包括：系统运行环境设计，主要描述了硬件设备、运行环境和开发环境；软件结构设计，主要描述了软件总体结构、软件技术架构和软件功能结构设计；类图设计；功能模块设计，主要描述了功能模块设计和层次图；接口设计，主要描述了接口设计说明；系统安全和权限设计，主要描述了安全性和权限设计说明。

本文档可作为：

- （1）向用户描述本项目功能的依据。
- （2）开发人员进行详细设计和编码的基础。
- （3）软件测试的依据。

1.2. 读者对象

本系统设计说明书的预期读者为用户、本项目小组的开发人员、测试人员、PM和未来的系统维护人员。

1.3. 参考文献

[1]邹欣.构建之法[M].3 版:人民邮电出版社,2014.

1.4. 术语与缩写解释

缩写、术语	解释
BS 架构	B/S 架构即浏览器和服务器架构模式，在这种架构下，用户工作界面是通过 WWW 浏览器来实现，极少部分事务逻辑在前端实现，但是主要事务逻辑在服务器端实现，形成所谓三层 3-tier 结构。
Spring Boot	Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定

缩写、术语	解释
	的方式来进行配置，从而使开发人员不再需要定义样板化的配置。
Vue.js	Vue 是一套用于构建用户界面的渐进式 JavaScript 框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，方便与第三方库或既有项目整合。
HTTP	HTTP 是一个简单的请求-响应协议，它通常运行在 TCP 之上。它指定了客户端可能发送给服务器什么样的消息以及得到什么样的响应。
Token	服务端生成的一串字符串，作为客户端进行请求的一个令牌，当第一次登录后，服务器生成一个 Token 便将此 Token 返回给客户端，以后客户端只需带上这个 Token 前来请求数据即可，无需再次带上用户名和密码。

表 1-1 术语与缩写解释列表

1.5. 文档约定

文档规定主要是指本系统设计说明书的排版约定，正文风格为：

标题 1：宋体，二号，粗体；

标题 2：宋体，三号，粗体；

标题 3：宋体，四号，粗体；

正文：仿宋 GB_2312，小四号；

列表：仿宋 GB_2312，五号；

行距：1.5 倍行距。

2. 总体设计

2.1. 系统运行环境设计

2.1.1. 硬件设备

硬件设备名称	主要配置	带宽	数量	用途
阿里云服务器	1 核 2G	1M	1 个	开发和搭载系统

表 2-1 硬件设备列表

2.1.2. 运行环境

平台	发行厂商	版本号	用途
火狐浏览器	Mozilla	72.0.2	用户通过浏览器进入系统

表 2-2 运行环境列表

2.1.3. 开发环境

序号	软件平台名称	发行厂商	版本号	补丁包版本号	用途
1	IDEA	JetBrains	2018	-	后端开发环境
2	Visual Studio Code	微软公司	1.42	-	前端开发环境

表 2-3 开发环境列表

2.2. 软件结构设计

2.2.1. 软件总体结构设计

系统采用 B/S 架构，前后端分离设计，前端通过后端接口的请求，实现相应数据交互，完成系统功能。一共分为三层：表现层、应用层和数据层。

- (1) 表现层。使用 Vue.js 框架进行前端开发。
- (2) 应用层。使用 Spring Boot 框架进行后端开发。
- (3) 数据层。使用 MYSQL 作为数据源。

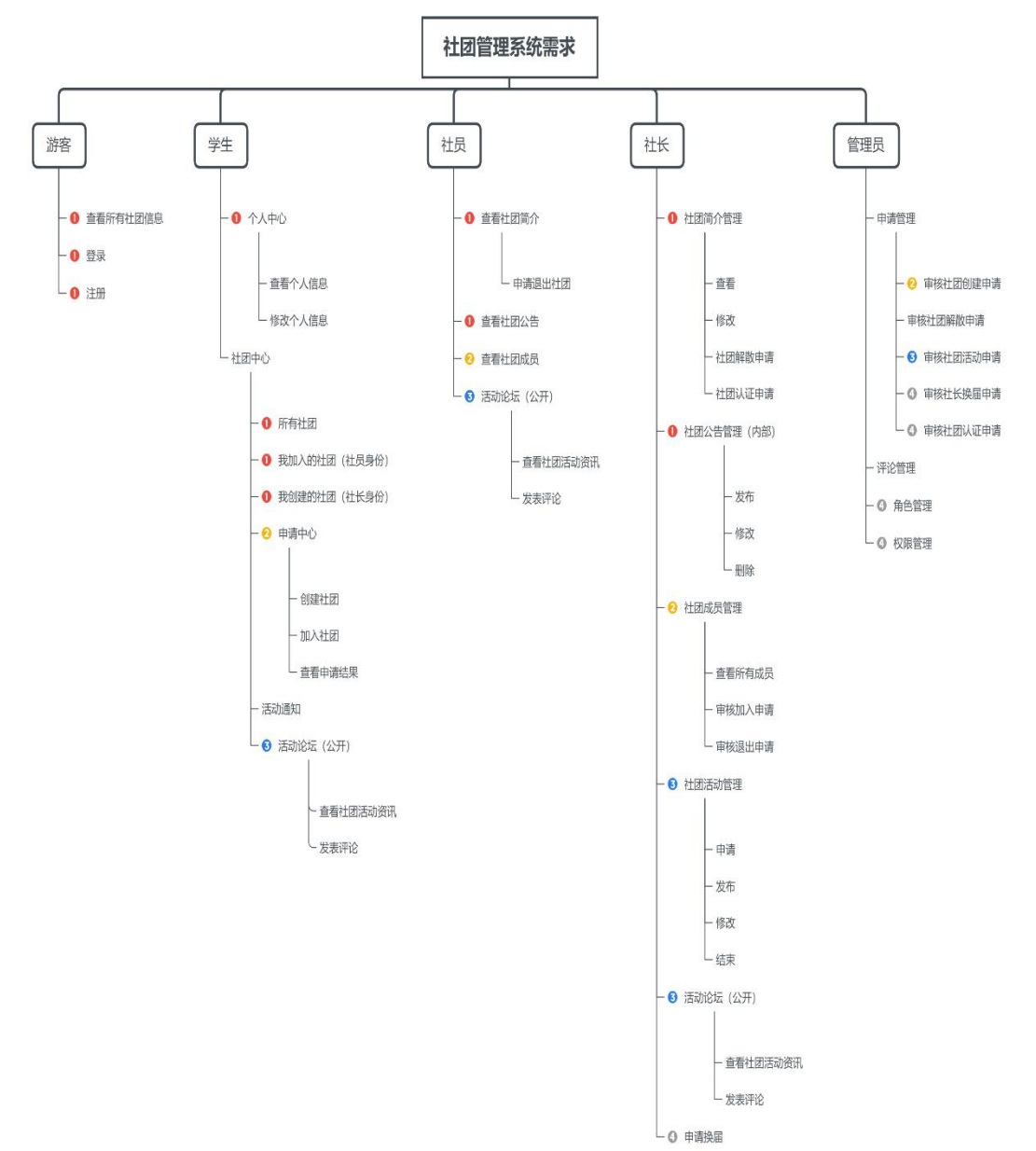


图 1-1 知社需求分析图

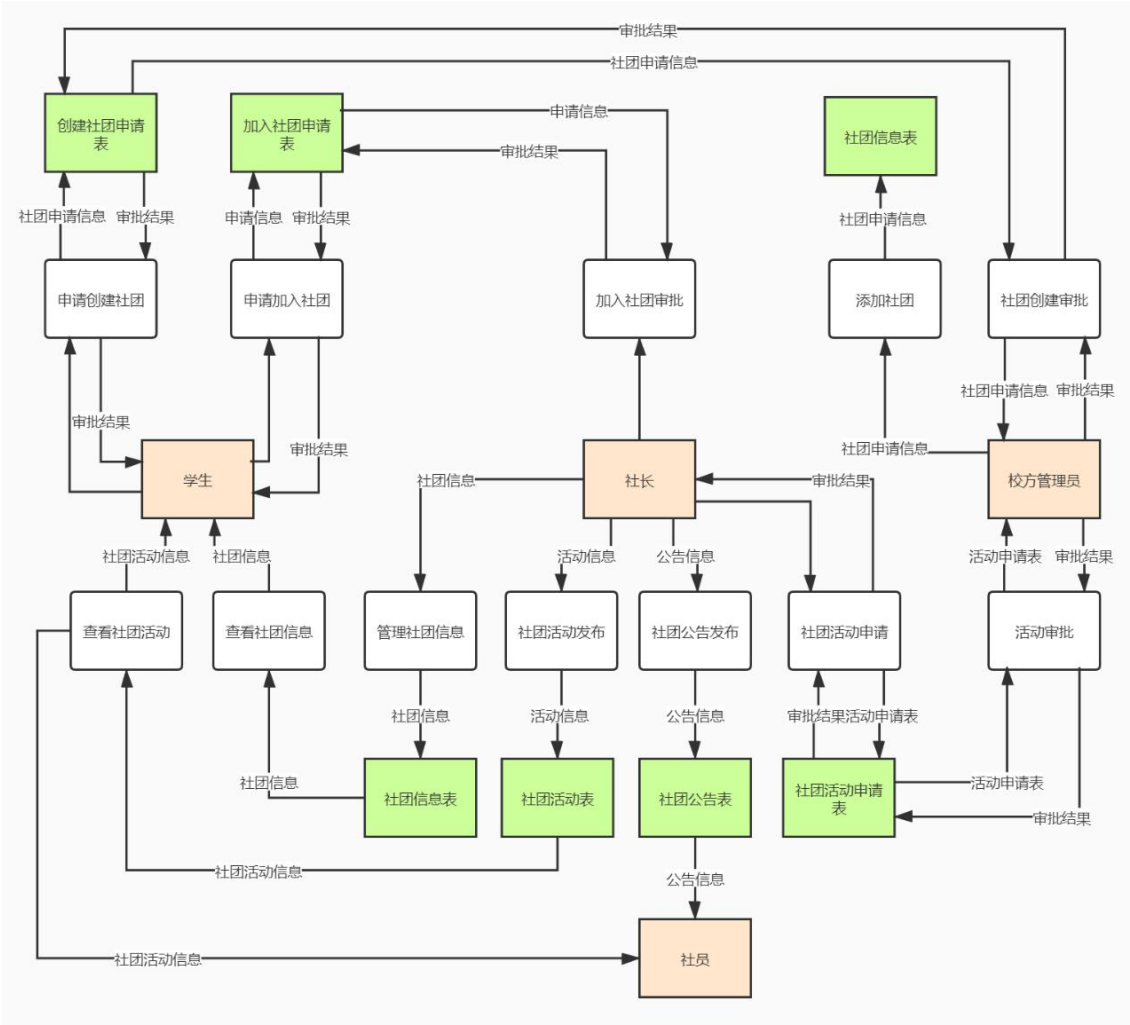


图 1-2 知社系统数据流图

2.2.2. 软件技术架构设计

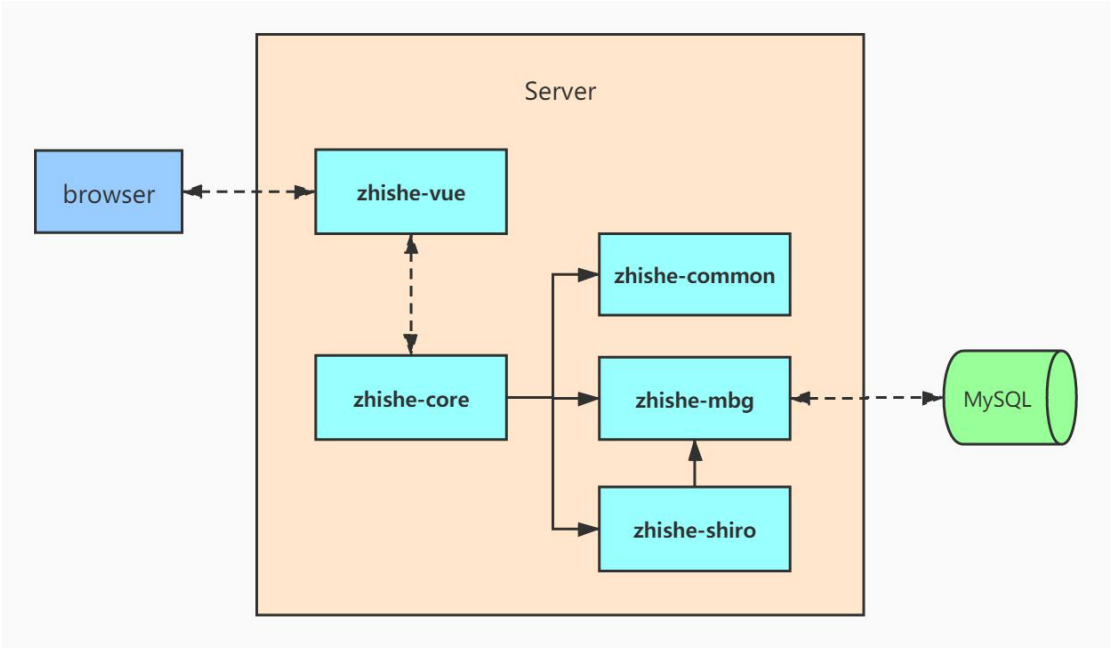


图 2-1 知社系统技术架构设计图

2.2.3. 软件功能结构设计

本项目的主要功能有账号管理、社团管理、论坛管理三个模块，其中又包含多个子模块，如下图 3-1。其中账号管理是针对用户账号登录、注册、查看和修改的功能模块，社团管理是针对已登录用户进行有关社团各种操作的功能模块，论坛管理是针对社团发布活动后进行发帖宣传、用户评论等的功能模块。

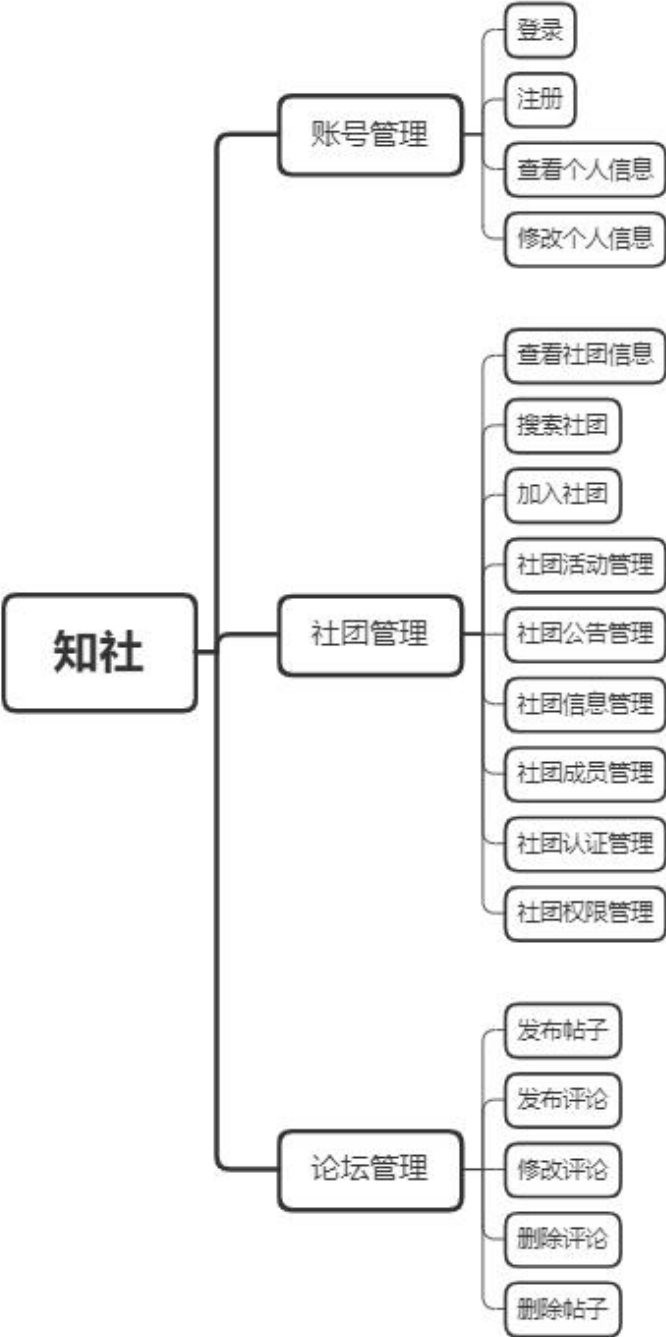


图 3-1 知社功能结构设计图

2.3. 类图设计

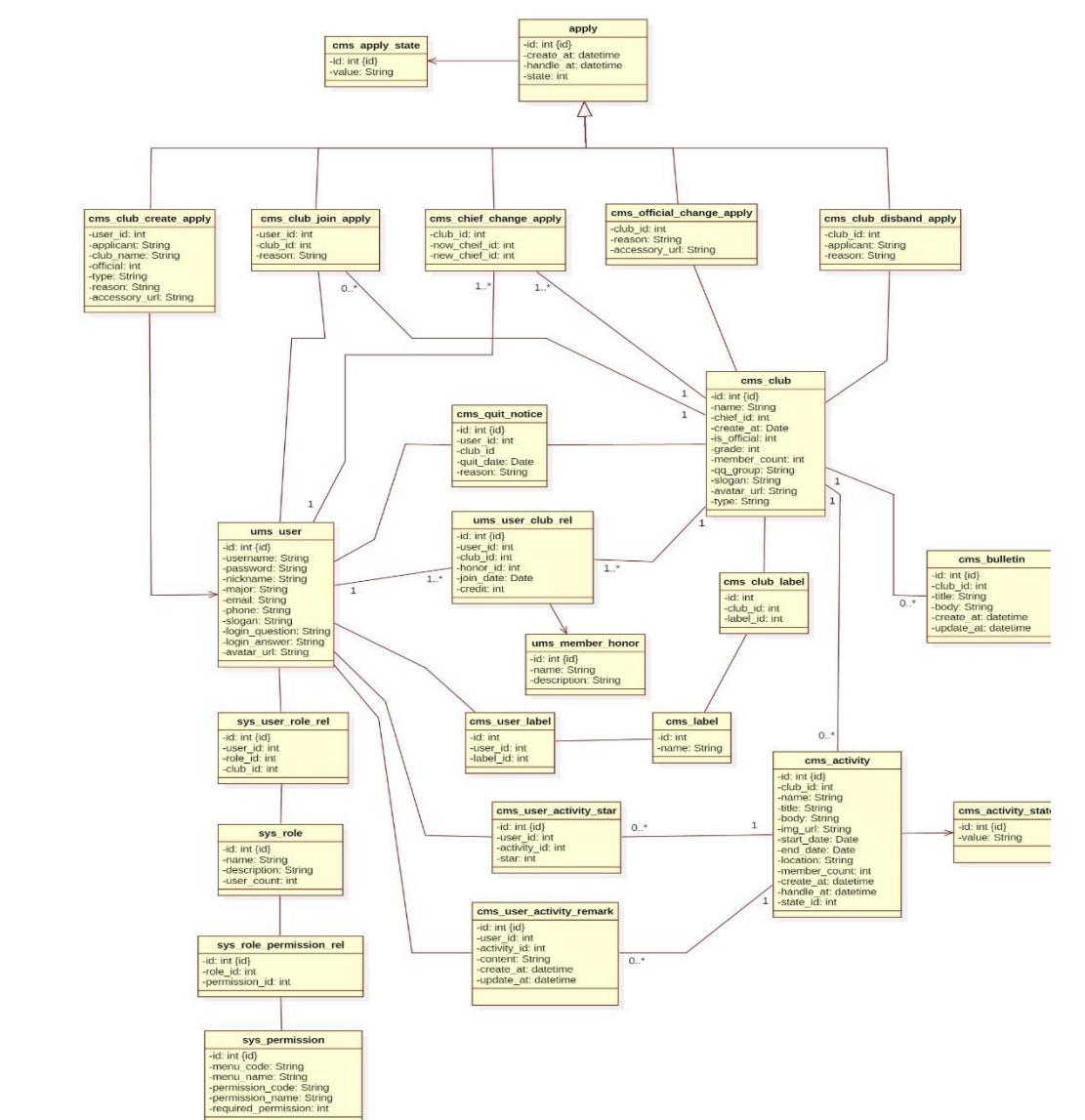


图 4-1 知社类图

cms_club 和 ums_user 分别是社团类和用户类。ums_user_club_rel 是用户与社团的关联类，与用户、社团均为一对多的关系，表明一个用户可以有多个社团，一个社团也对应多个用户。

cms_club_create_apply 、 cms_club_join_apply 、 cms_chief_change_apply 、 cms_official_change_apply、cms_club_disband_apply 继承自 apply 类，是社团审核类。

cms_user_label、cms_label、cms_club_label 是针对标签管理的类。

sys_user_role_rel、sys_role、sys_role_permission_rel、sys_permission 是针对权限管理的类。

cms_bulletin 是公告类。

3. 功能模块设计

3.1. 功能模块层次图

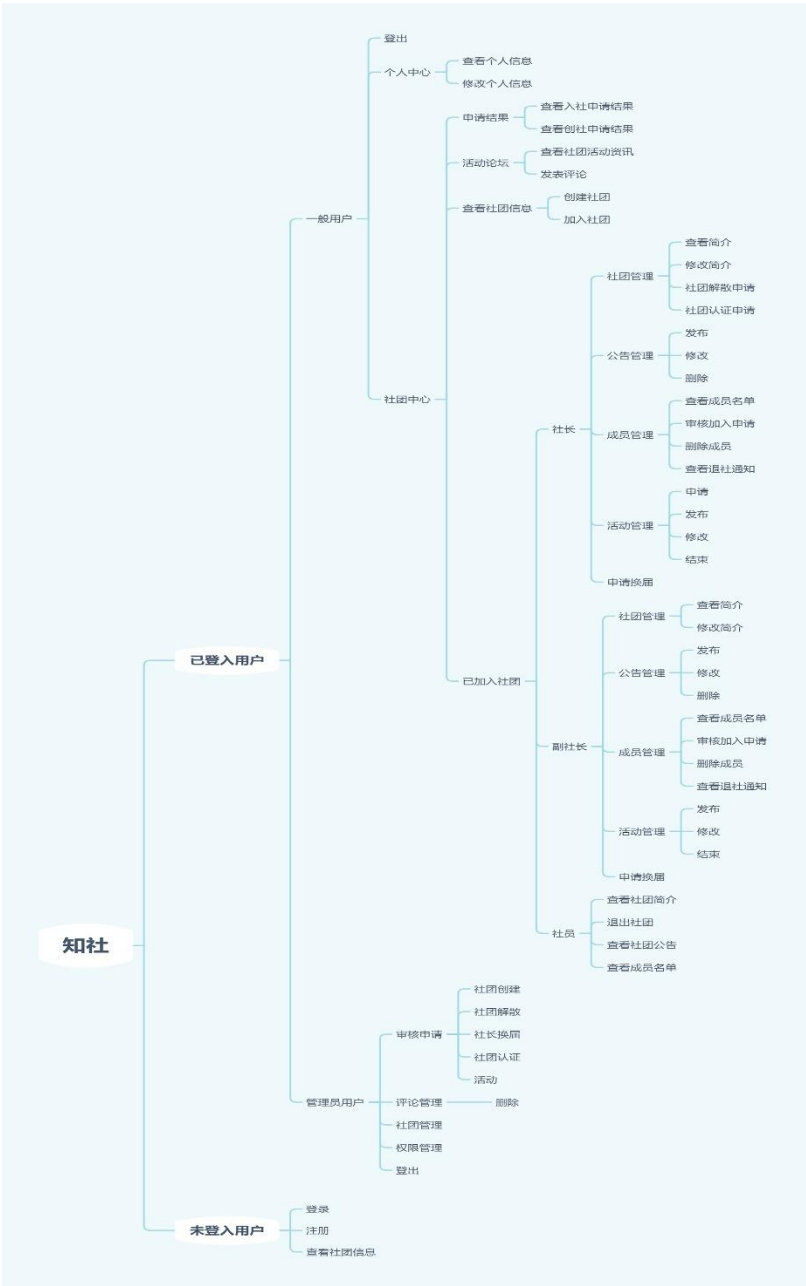


图 5-1 功能模块层次图

3.2. 功能模块设计

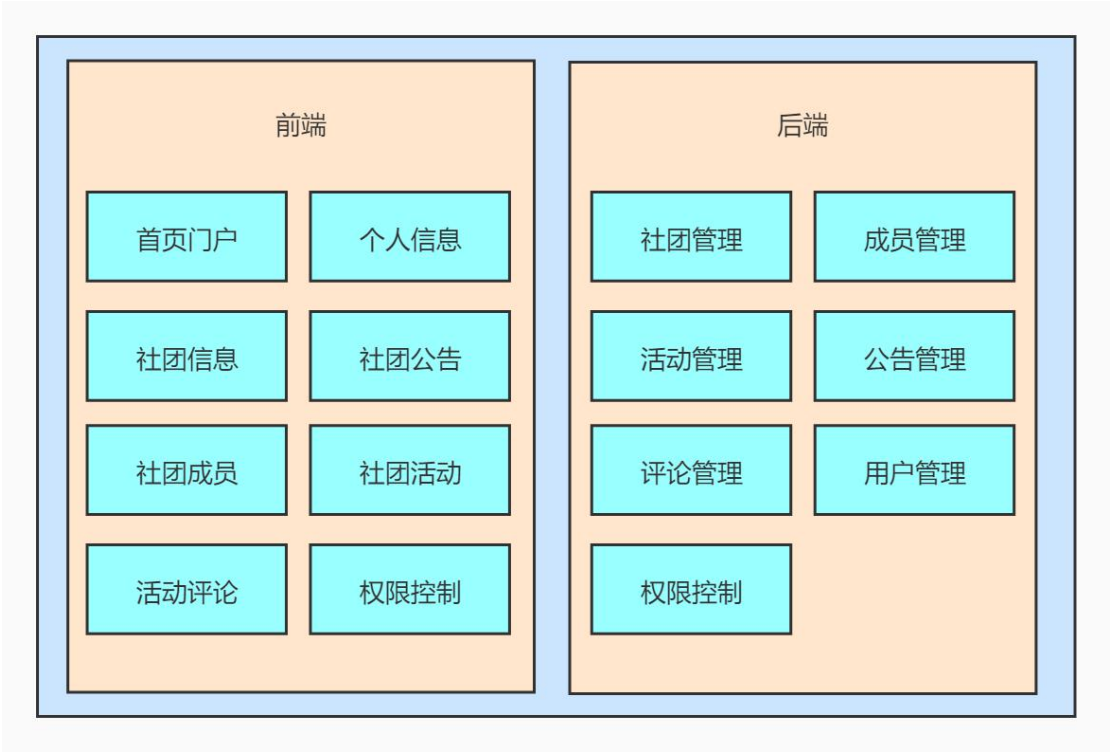


图 6-1 功能模块设计图

3.3. 功能模块描述

（一）身份描述

本项目的使用人群分为五类：游客、学生、社员、社长（包含副社长）和管理员。游客进行登录后转为学生身份，当学生在已登录页面点击进入加入的社团后，转成社员身份，当学生在已登录页面点击进入管理的社团后，转成社长身份。由于一名学生可能存在加入或管理多个社团的情况，所以身份会根据操作进行相应的变化，不同的身份对应有不同的权限以及页面展示。

（二）权限描述

	个人信息		社团管理							公告管理			成员管理							活动管理			
	查看	修改	搜索社团	申请入社	创建社团	查看简介	修改简介	解散申请	认证申请	发布	修改	删除	查看成员	删除成员	加社审核	申请退社	换届申请	退社通知	发表评论	申请活动	发布活动	修改活动	结束活动
游客	√	√	√	√	√	√													√				
学生	√	√	√	√	√	√							√			√	√		√				
社员	√	√	√	√	√	√	√			√	√	√	√	√	√			√	√		√	√	√
副社长	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√		√	√	√	√	√	√	√
社长	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√		√	√	√	√	√	√	√

图 7-1 权限描述表（1）

	审核申请					评论管理			
	社团创建	社团解散	社长换届	社团认证	活动申请	删除	修改	角色管理	权限管理
管理员	√	√	√	√	√	√	√	√	√

图 7-2 权限描述表（2）

（三）头衔、活跃度描述

已加入社团的用户（包括社员和社长）具有头衔和活跃度两个属性，活跃度初始值为 0。头衔是活跃度在不同阶段的等级展示，活跃度越高则等级越高。活跃度是依据登录次数、发表评论的数目进行统计。例如：登录一次，活跃度+1，发表评论一次，活跃度+5。活跃度 0~20 对应“冒泡”头衔，活跃度 21~50 对应“活跃”头衔等。

（四）用况图

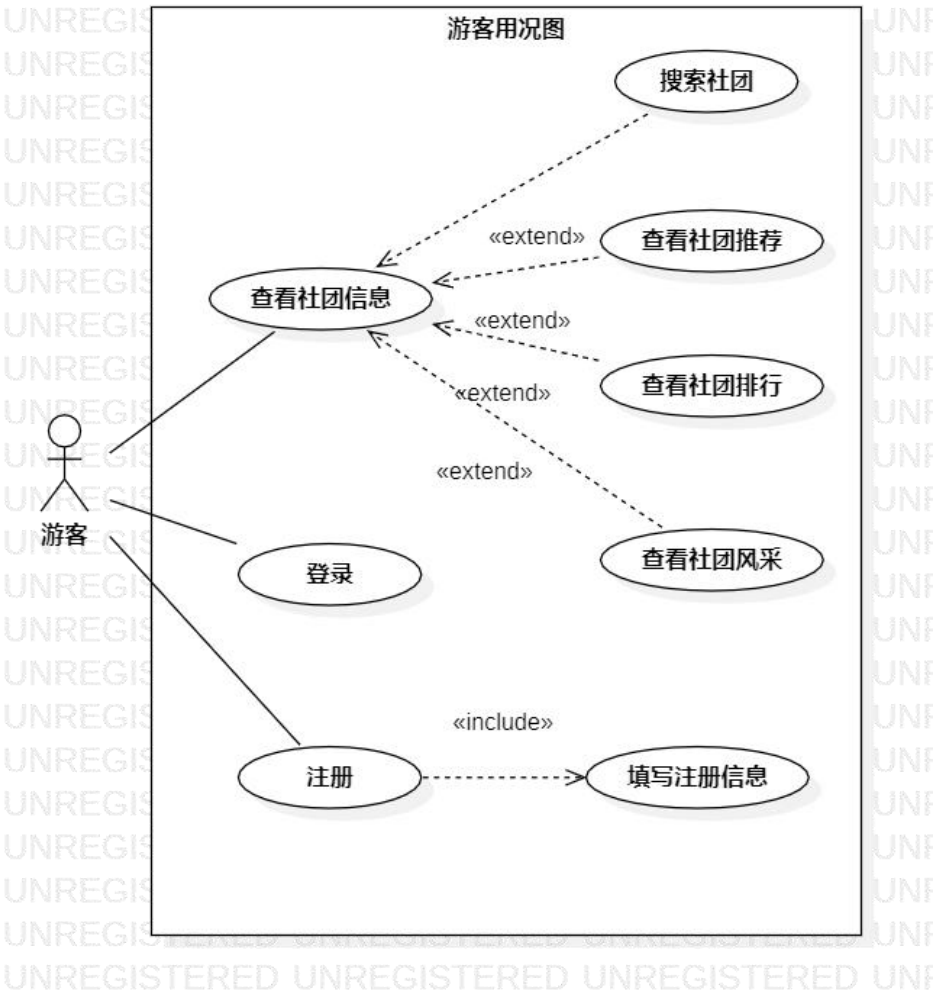


图 8-1 游客用况图

游客指尚未登录本系统的用户,仅可以查看各个社团信息、进行注册和登录操作。

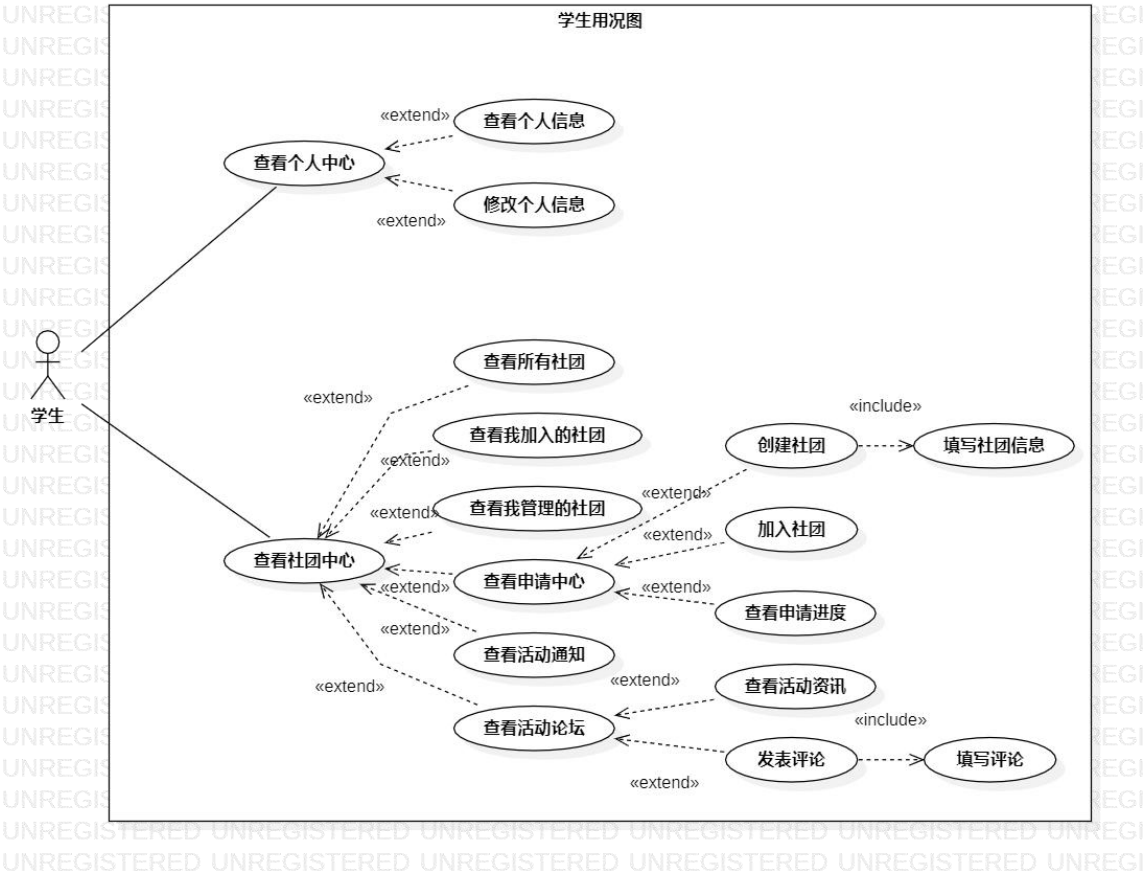


图 8-2 学生用况图

学生指已登录系统的用户，包括未加入社团的学生、已加入社团的社员和社长。用户登录后可以查看个人中心和社团中心。社团中心包括查看所有社团、用户已加入和管理的社团、申请中心、活动通知和活动论坛功能。用户进入我加入的社团会转为社员身份，进入我管理的社团会转为社长身份。

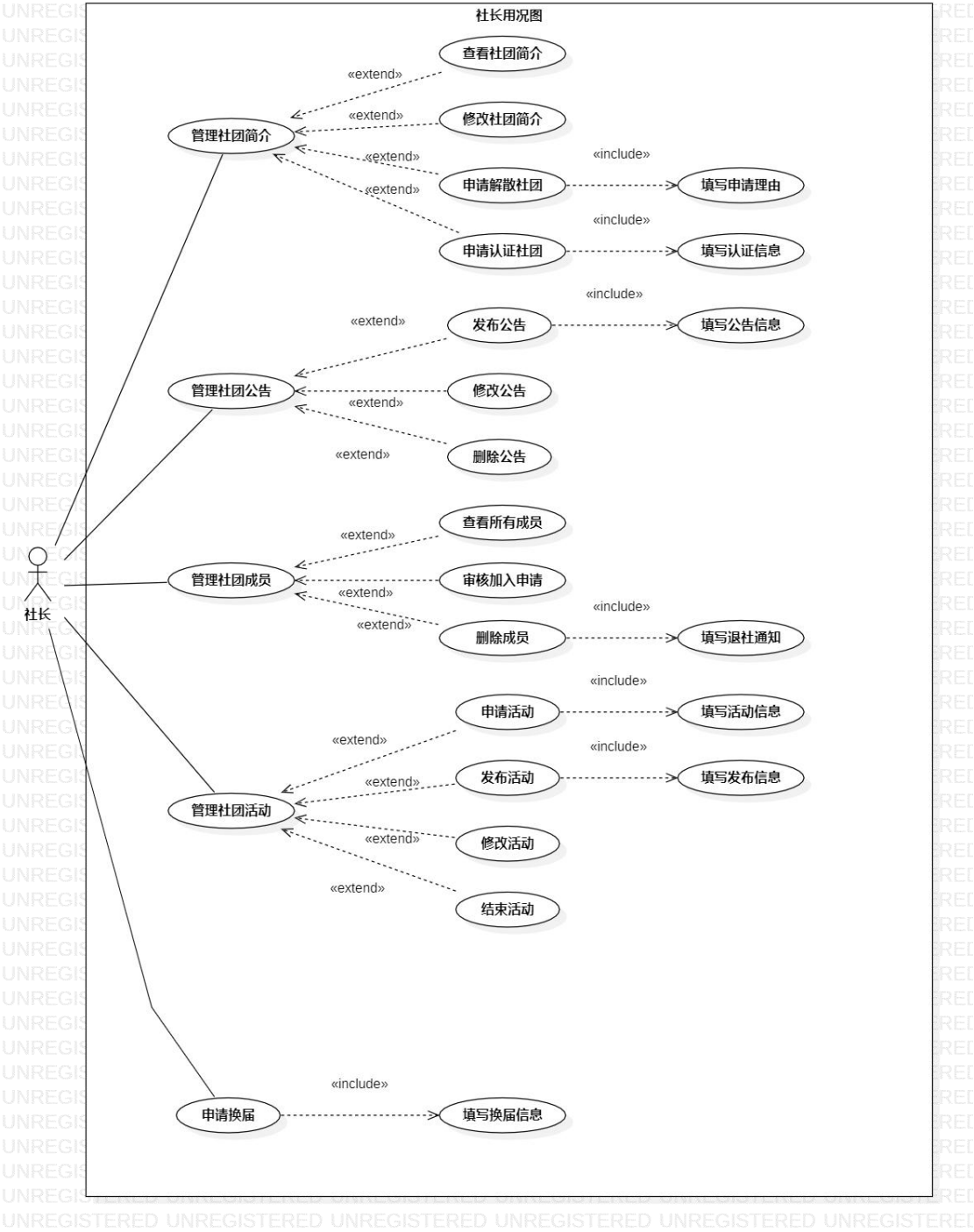


图 8-3 社长用况图

社长包括管理社团简介、社团公告、社团成员、社团活动、社团论坛和申请换届功能。

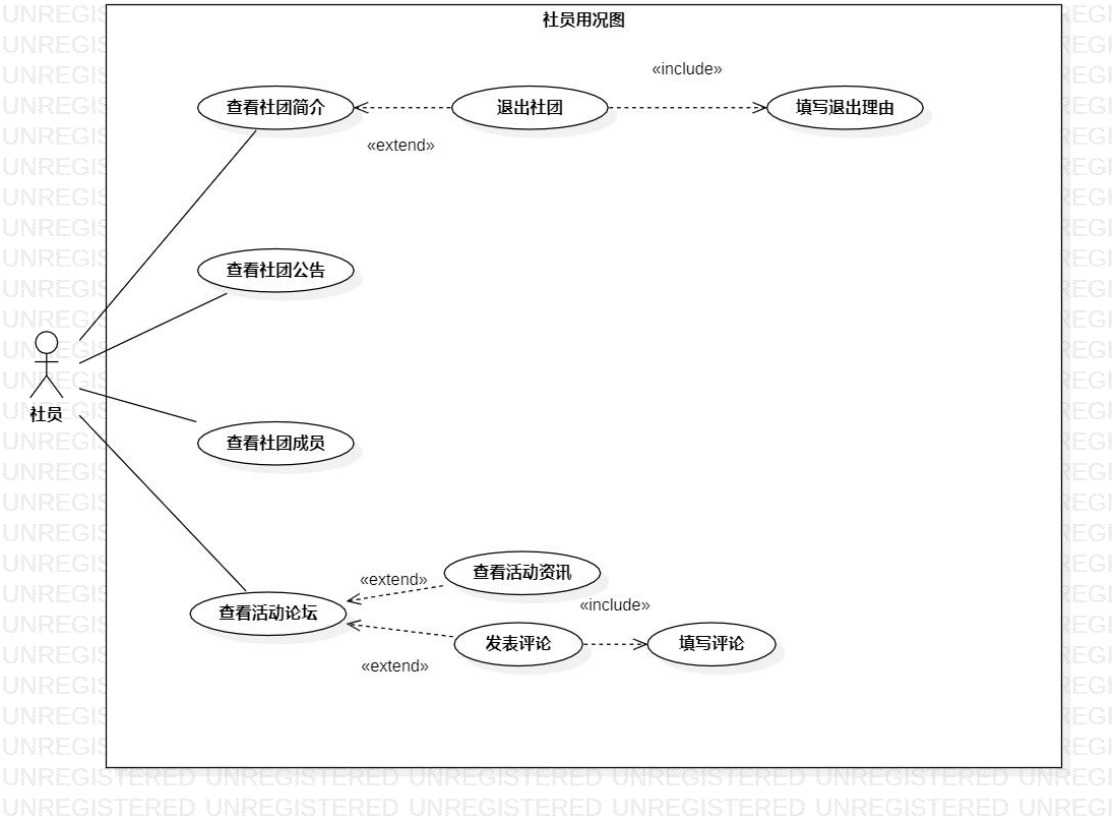


图 8-4 社员用况图

社员包括查看社团简介、社团公告、社团成员功能。

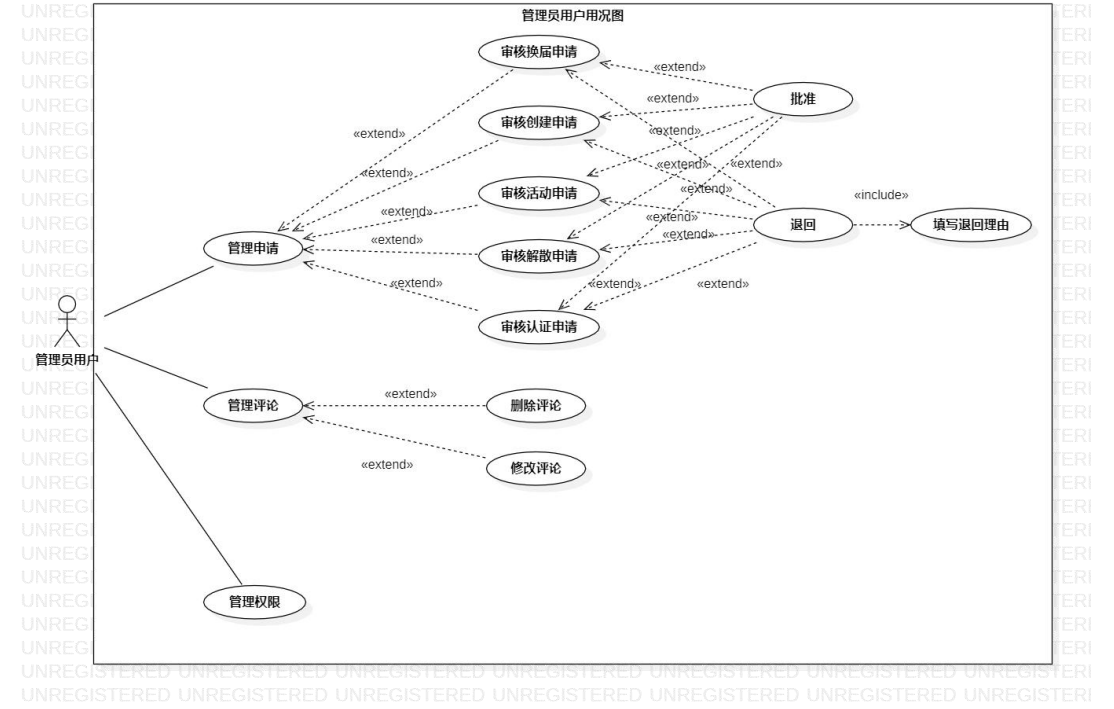


图 8-5 管理员用况图

管理员指社团联合委员会的高层管理员或老师。通过登录页面对输入账号的判别进入与学生用户不同的页面。管理员的主要功能包括管理申请、论坛评论、角色和权限。

（五）活动图

游客：

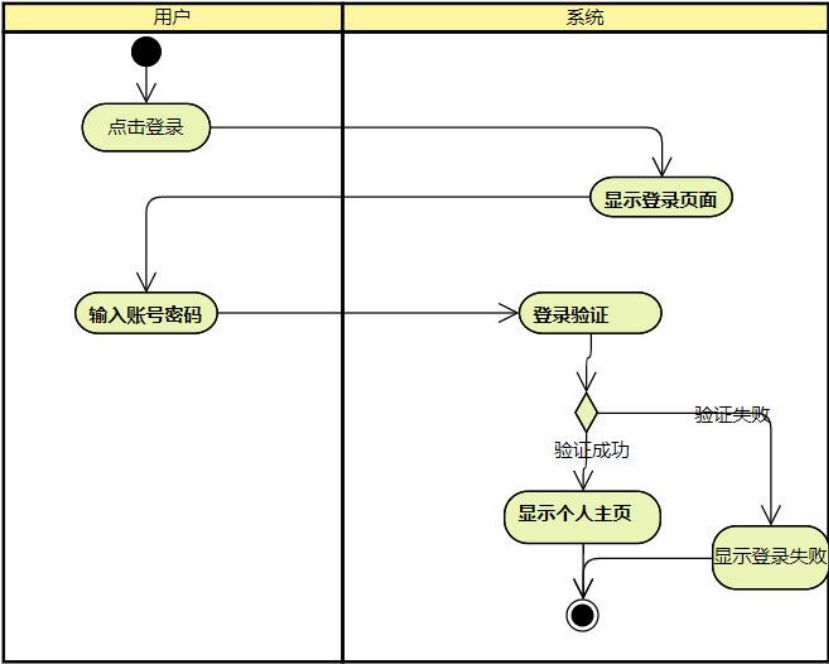


图 9-1 游客登录活动图

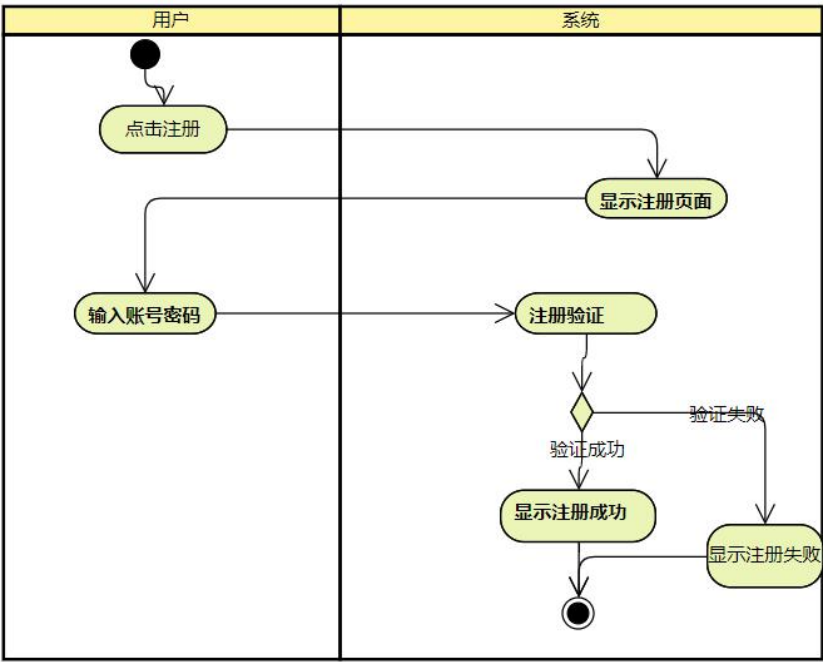


图 9-2 游客注册活动图

学生：

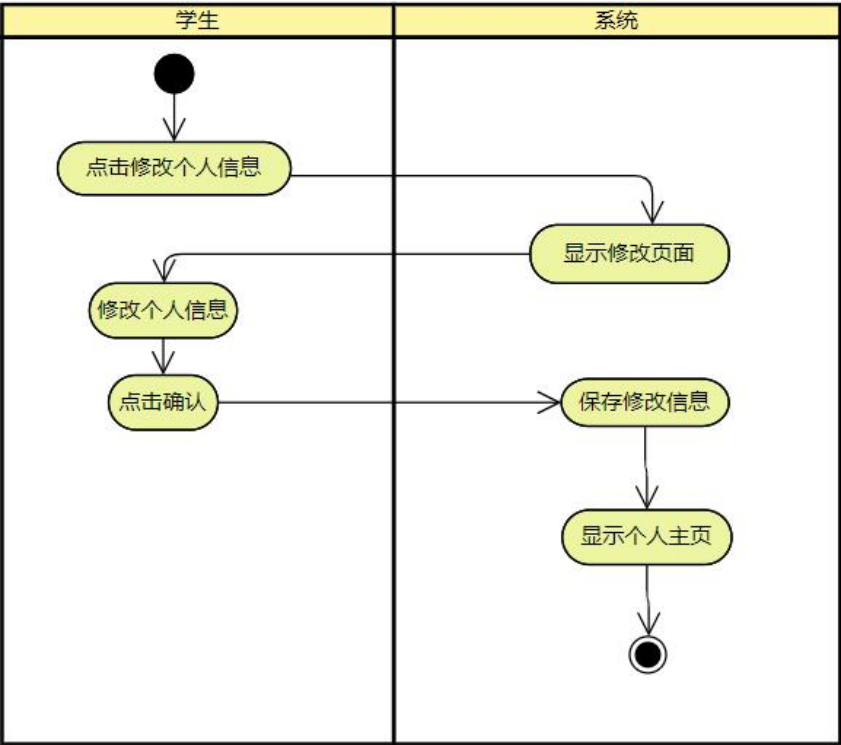


图 9-3 学生修改个人信息活动图

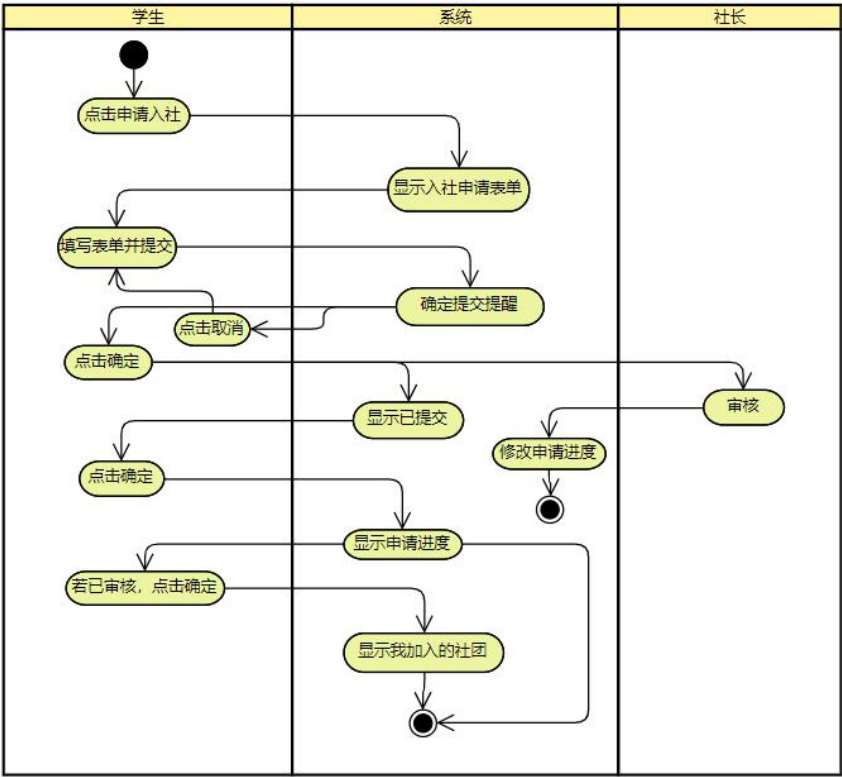


图 9-4 学生申请加入社团活动图

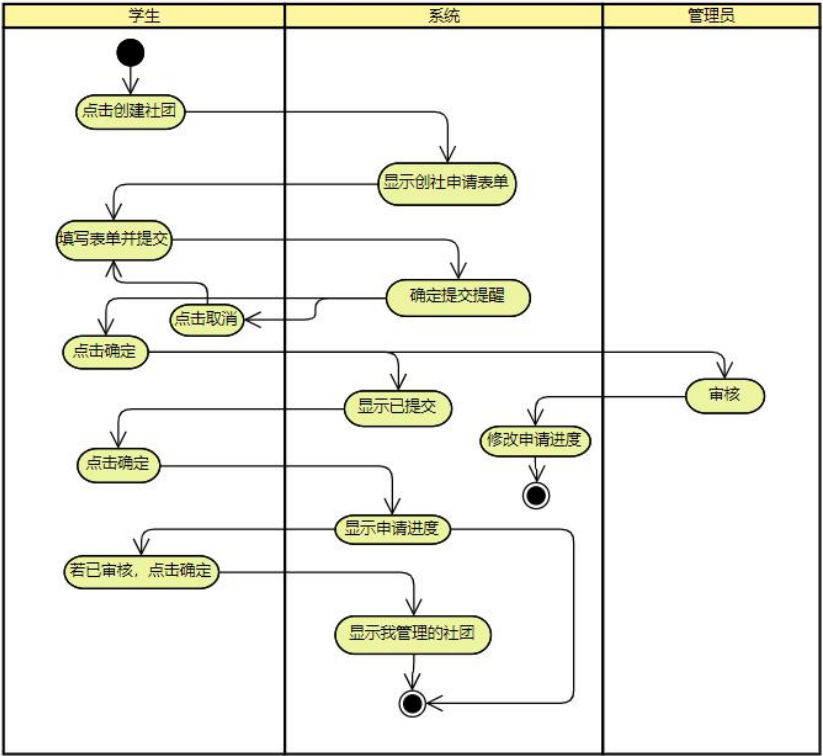


图 9-5 学生创建社团活动图

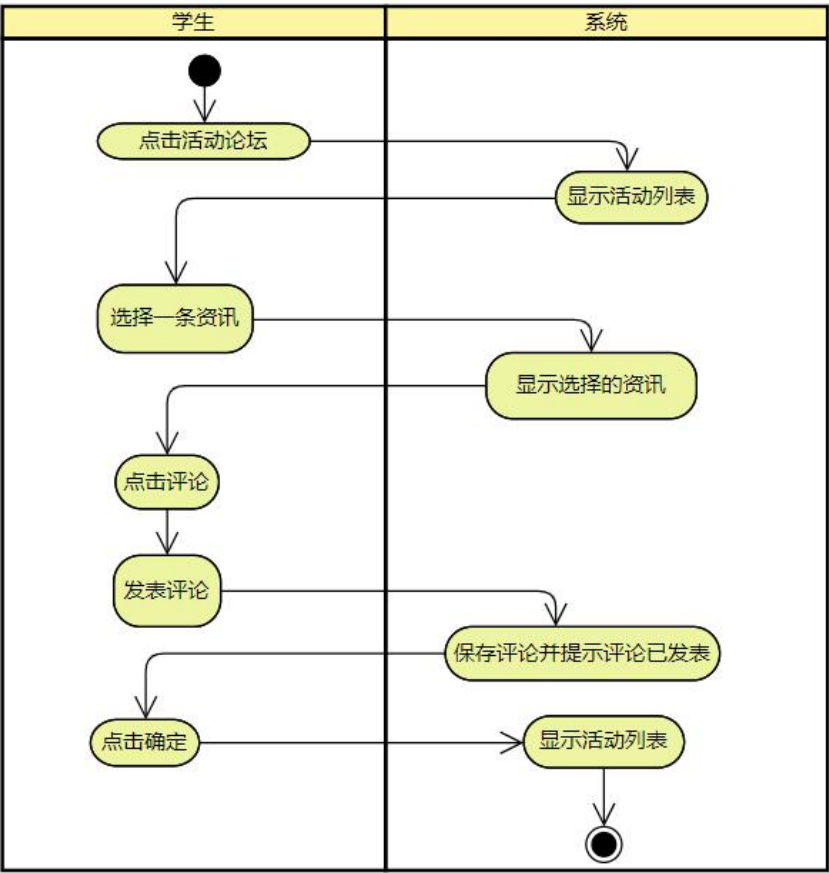


图 9-6 学生在活动论坛评论活动图

社员：

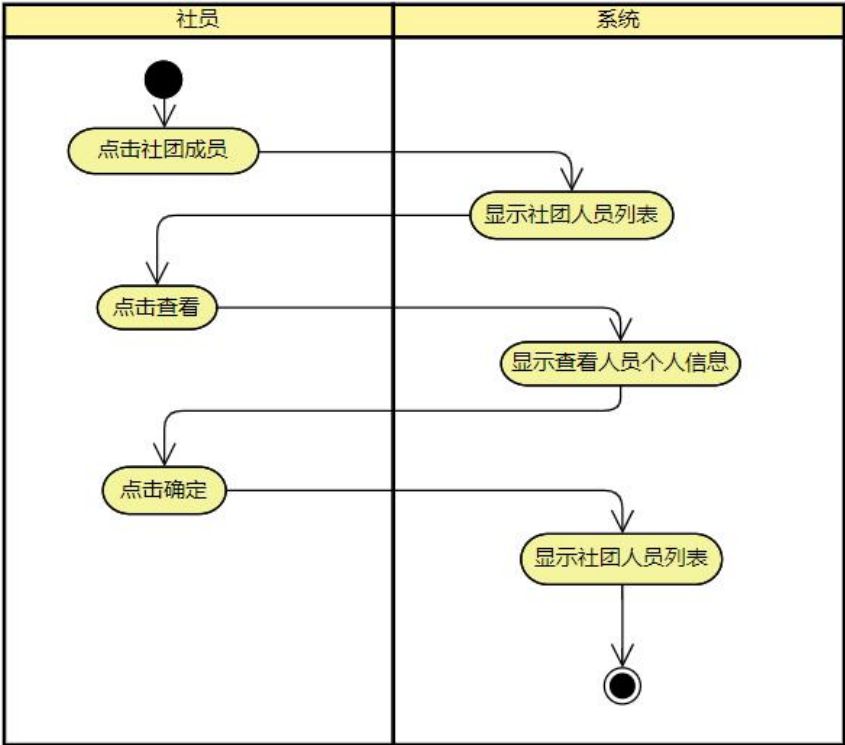


图 9-7 社员查看社团成员活动图

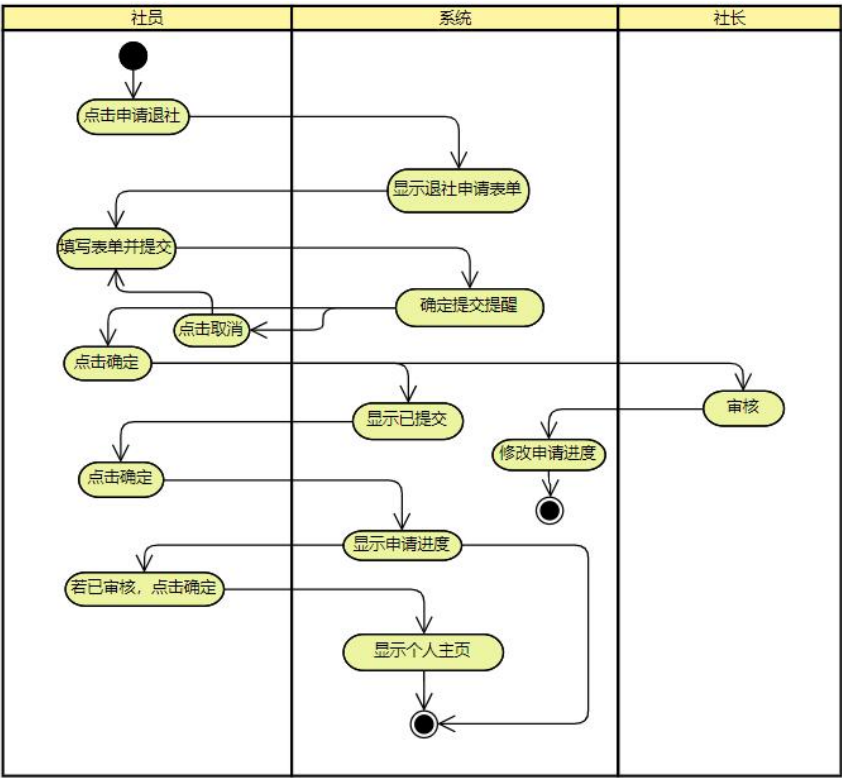


图 9-8 社员申请退出社团活动图

社长：

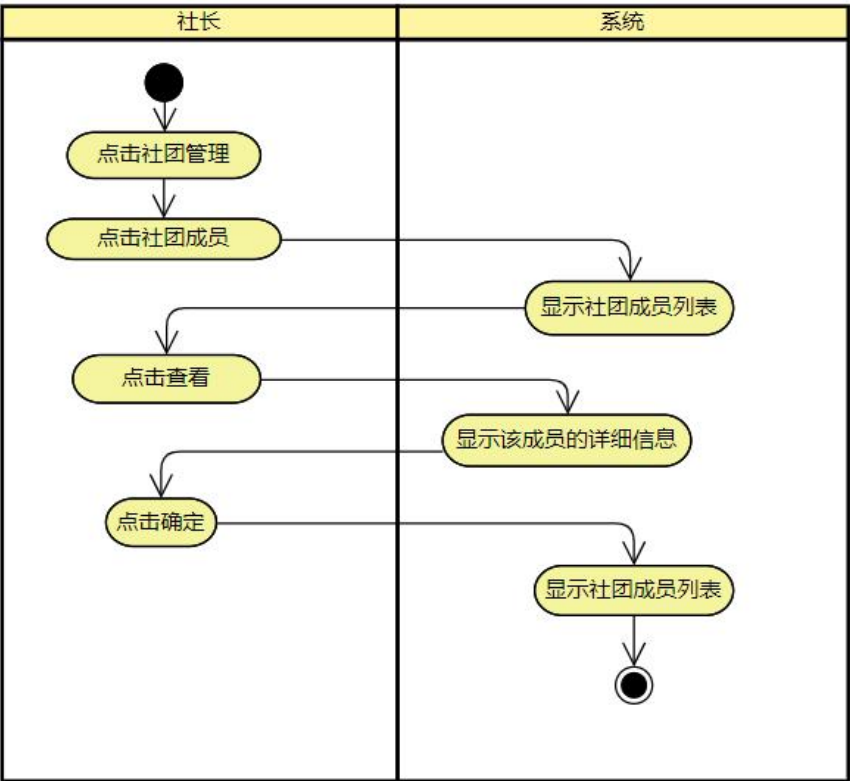


图 9-9 社长查看社团成员活动图

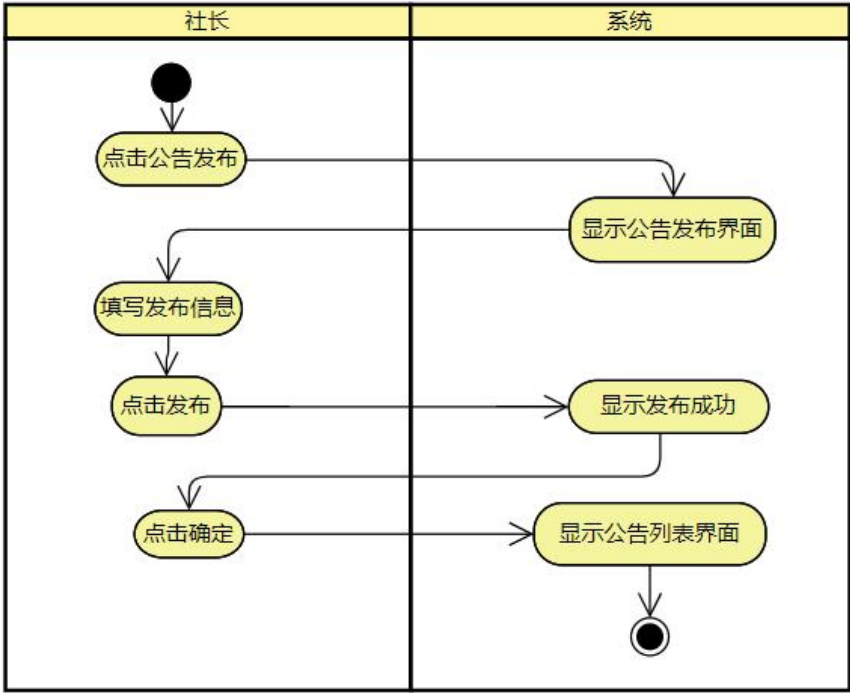


图 9-10 社长发布公告活动图

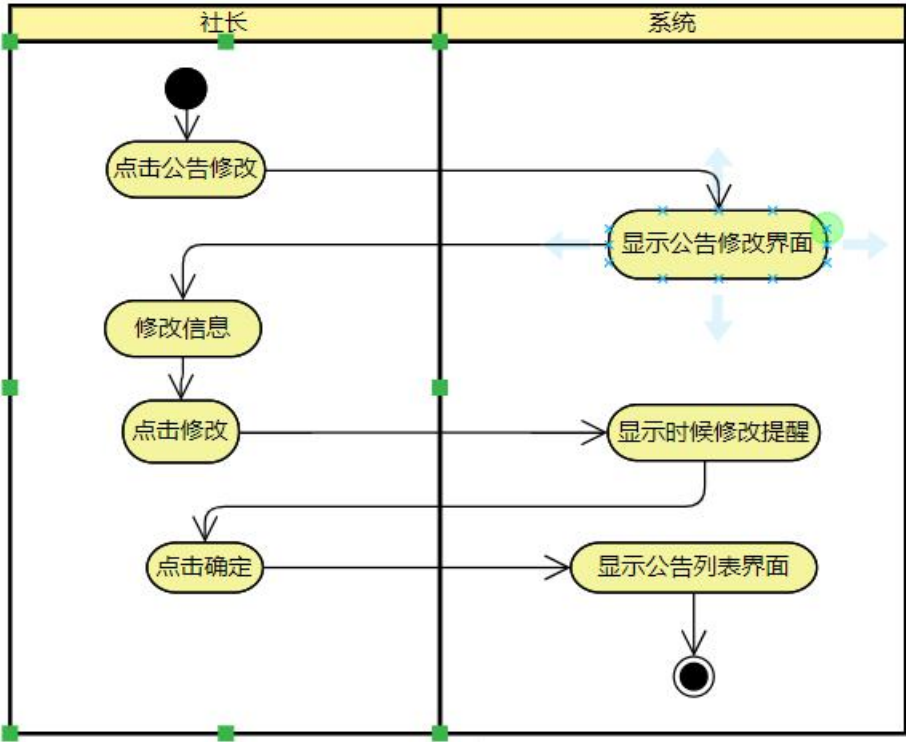


图 9-11 社长修改公告活动图

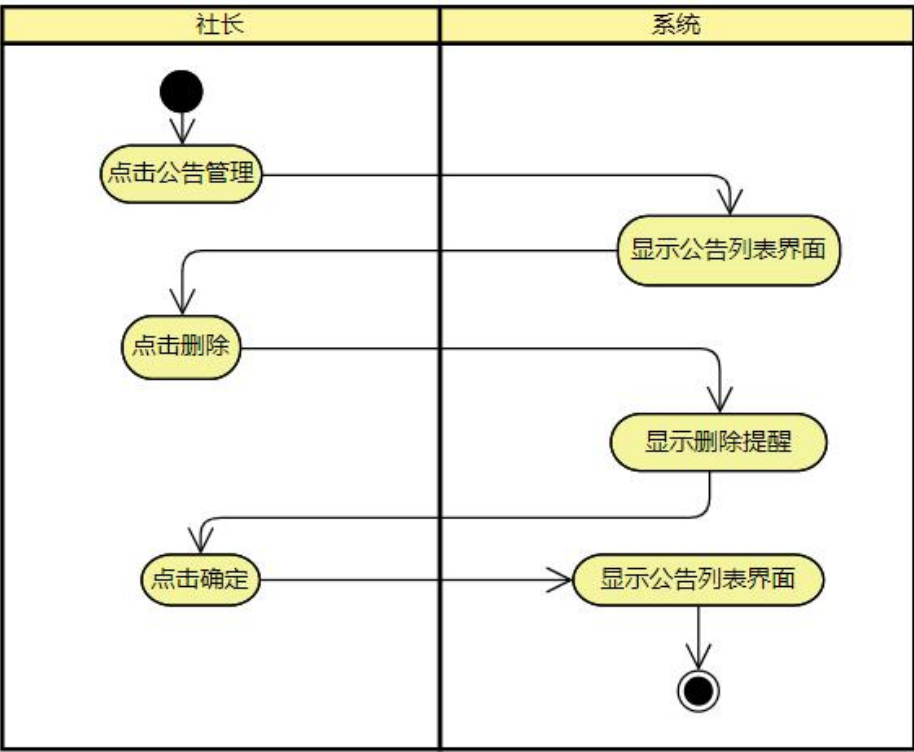


图 9-12 社长删除公告活动图

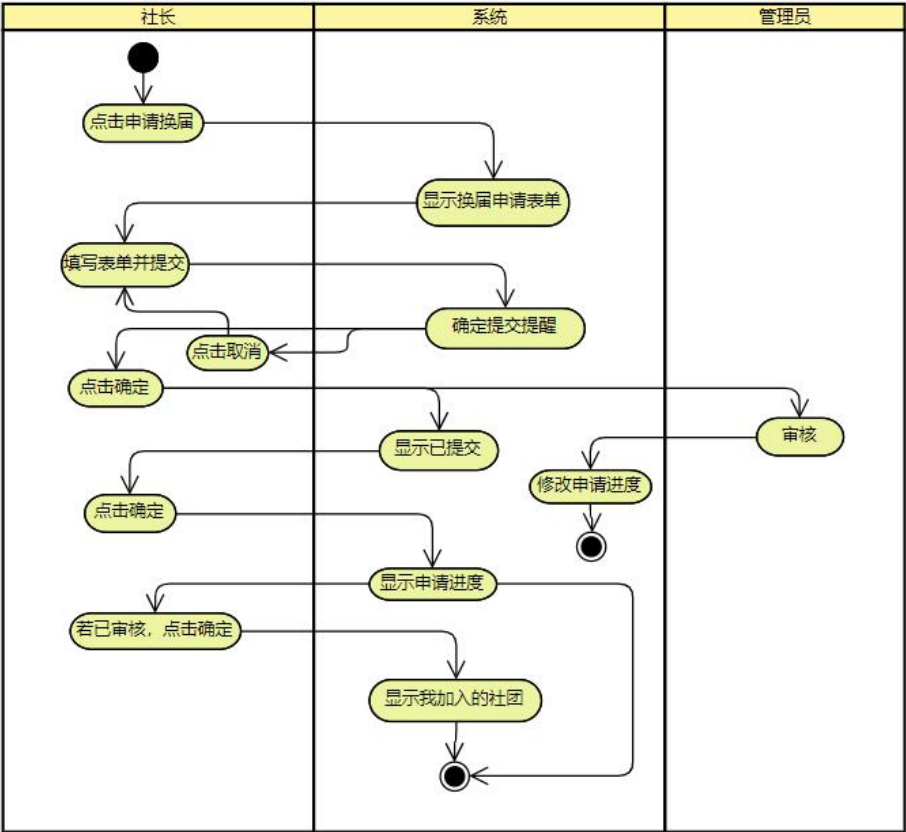


图 9-13 社长申请换届活动图

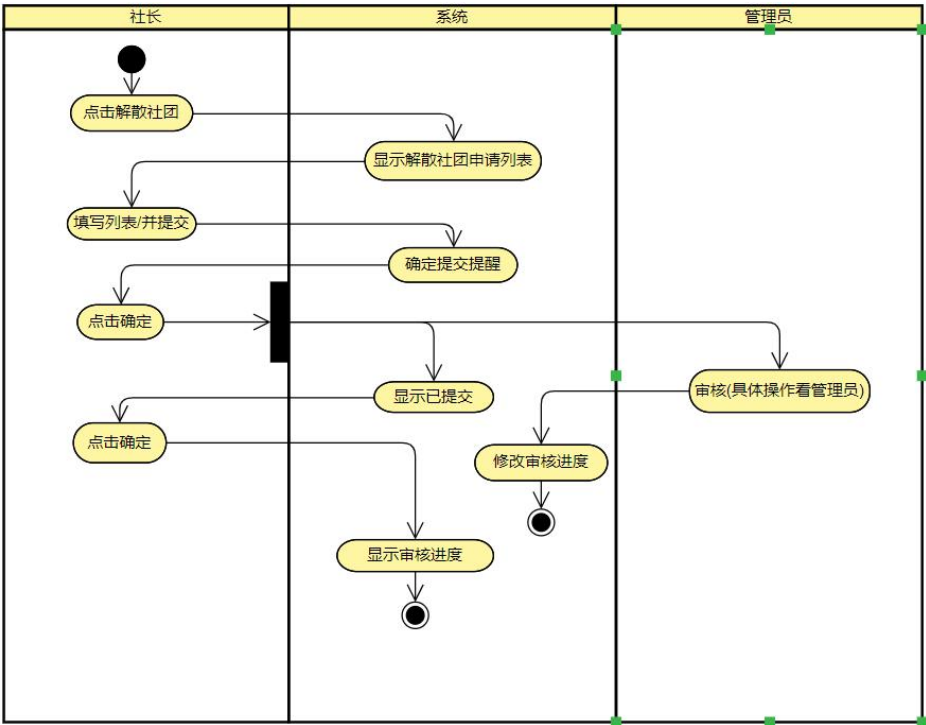


图 9-14 社长解散社团活动图

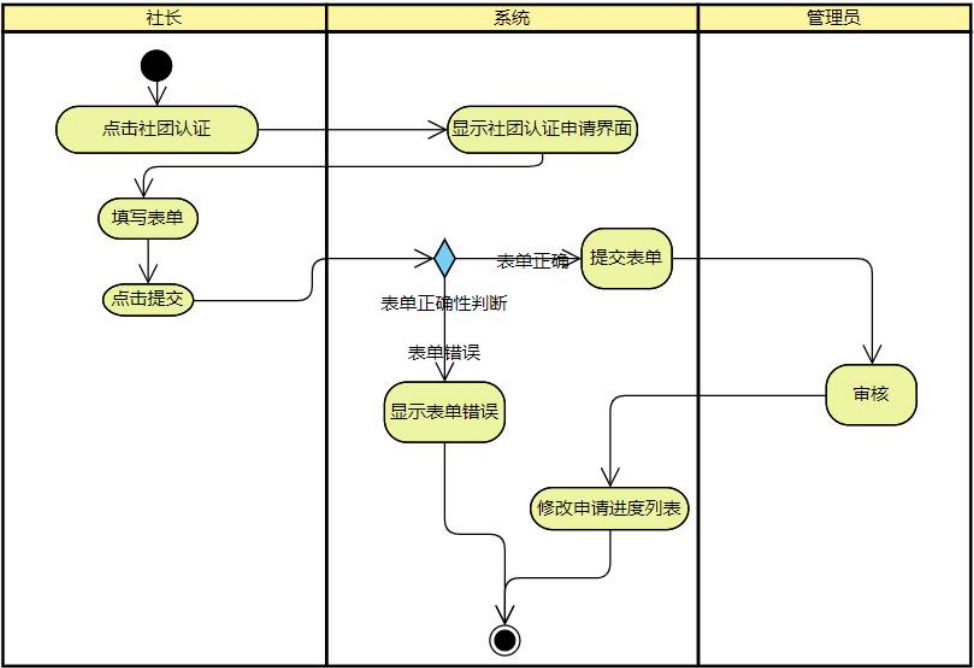


图 9-15 社长申请社团认证活动图

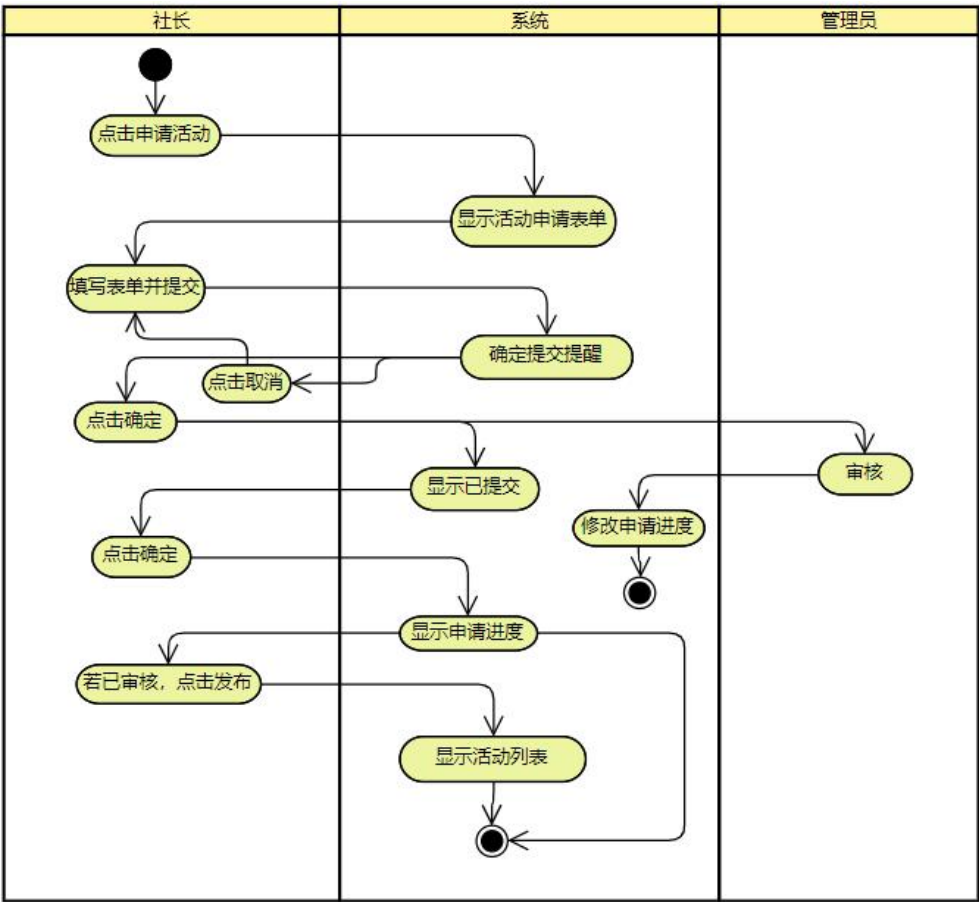


图 9-16 社长申请活动活动图

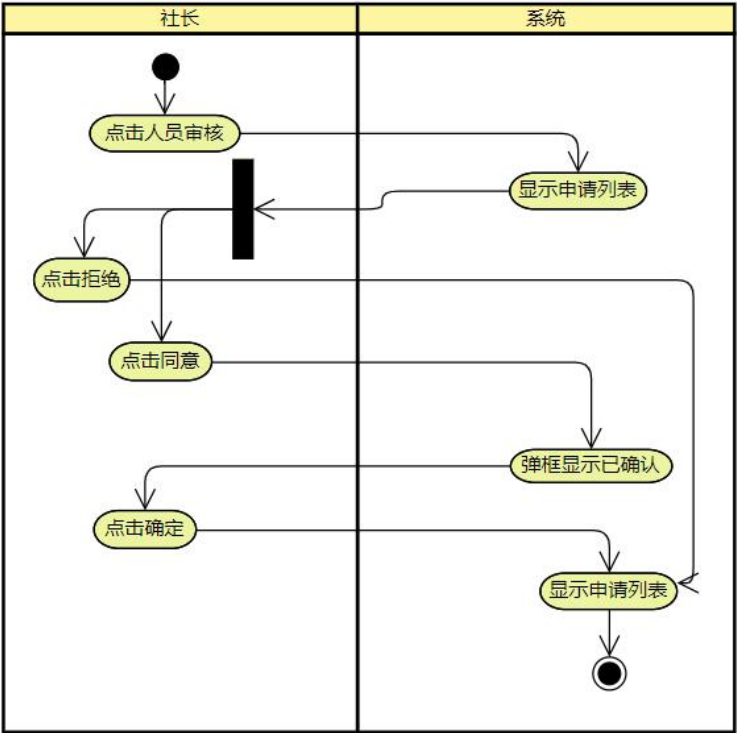


图 9-17 社长审核入社申请活动图

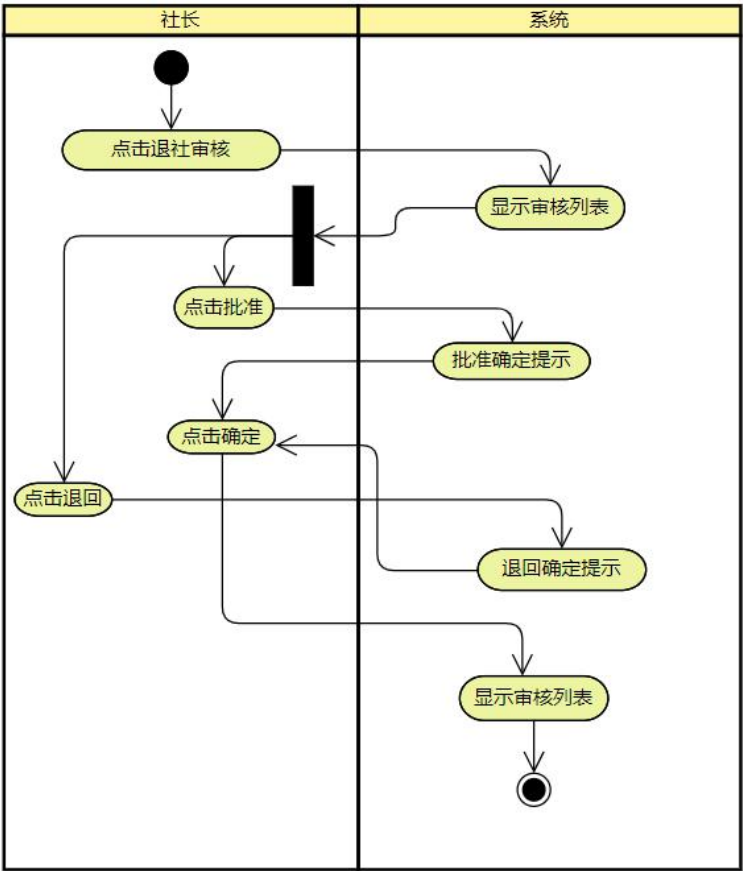


图 9-18 社长审核退社申请活动图

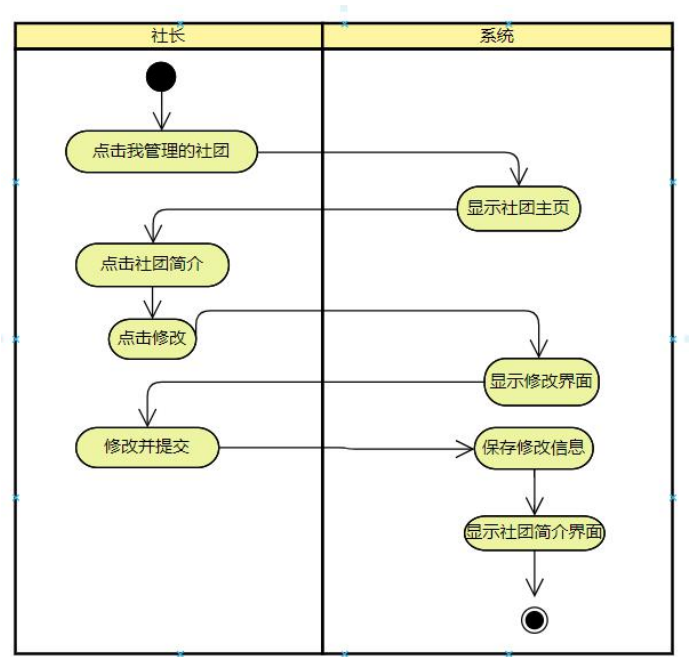


图 9-19 社长修改社团信息活动图

管理员：

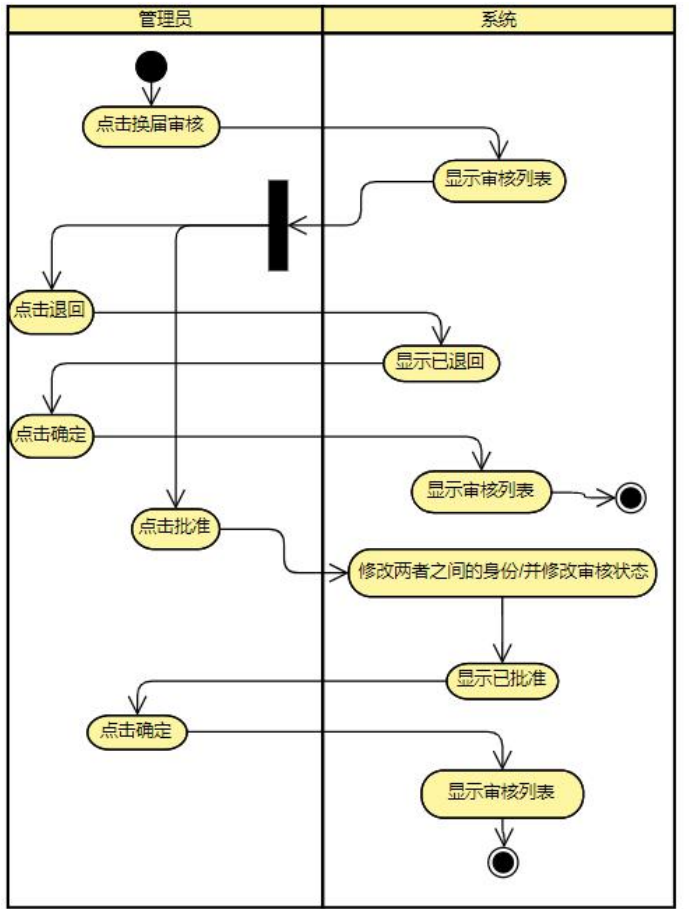


图 9-20 管理员换届审核活动图

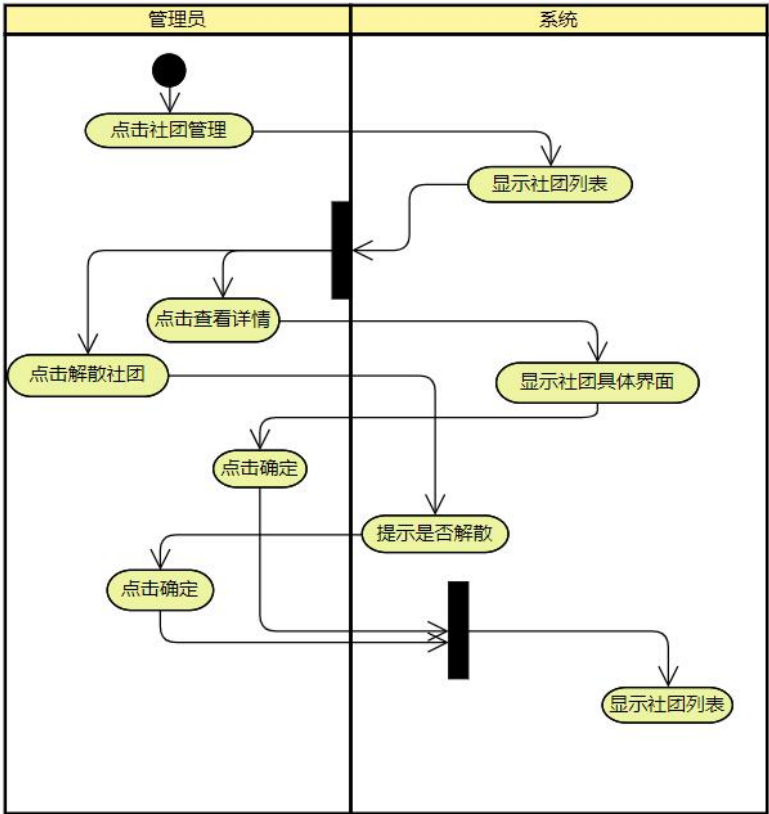


图 9-21 管理员管理社团活动图

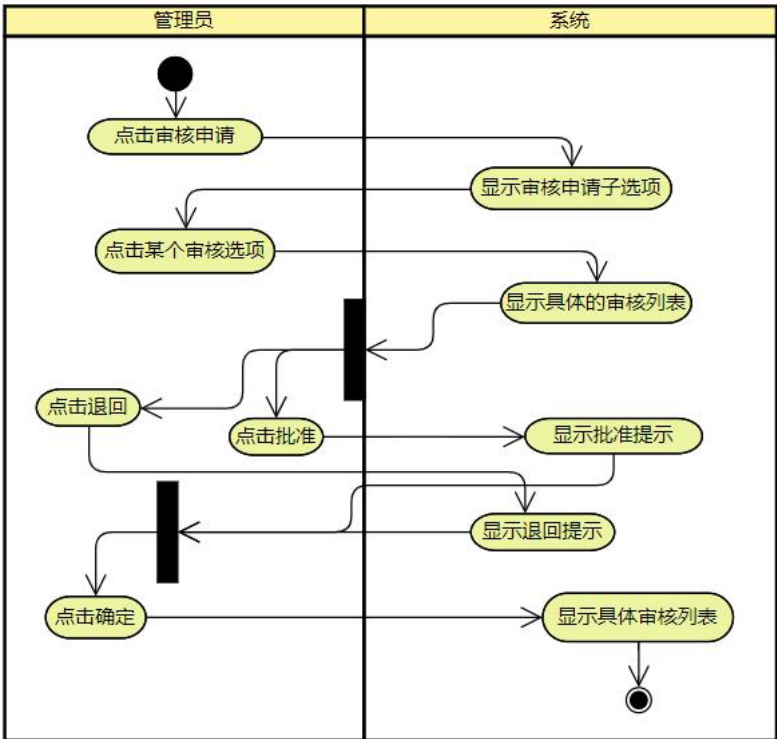


图 9-22 管理员审核申请活动图

4. 接口设计

4.1. 接口设计规范

4.1.1. 请求格式

采用 RESTful 风格的接口。

Verb	Description
HEAD	获取 HTTP 头部信息
GET	获取资源
POST	创建资源
PATCH	更新部分 JSON 内容，不常用
PUT	更新资源
DELETE	删除资源

表 3-1 HTTP 动词

API 自动对请求的元素分页，不同的 API 有不同的默认值，可以指定查询的最大长度，但对某些资源不起作用。请求后端数组数据时，统一传递四个分页参数。

参数	含义
page	请求页码
limit	每页的元素数量
sort	排序字段，例如 “add_time” 或者 “id”
order	升序降序，只能是 “desc” 或者 “asc”

表 3-2 分页参数

这里四个参数是可选的，后端应该设置默认参数，因此即使前端不设置，后端也

会自动返回合适的对象数组响应数据。

Example

```
GET /goods/list?page=1&limit=10
```

Response

```
HTTP/1.1 200 OK

{
  "total_count": 2525,
  "items": [
    ...
  ]
}
```

图 10-1 返回响应数据示例

返回参数说明：

响应体的 `total_count` 表示总数量， `items` 是数据，里面是要查询的元素。

4.1.2. 响应格式

```
Content-Type: application/json;charset=UTF-8
```

```
{
  body
}
```

图 10-2 响应格式

响应成功， `body` 直接返回 JSON 格式的内容：

```
[
  {
    "id": 1,
    "state": "open"
  },
  {
    "id": 2,
    "state": "open"
  }
]
```

图 10-3 返回内容

响应错误， `errors` 字段可选（一般是参数验证错误）：

```
{
  "message": "Validation Failed",
  "errors": [
    {
      "resource": "Issue",
      "field": "title",
      "code": "missing_field"
    }
  ]
}
```

图 10-4 错误响应返回内容

4.1.3. 认证方式

前后端采用 Token 来验证访问权限。前后端 Token 交换流程如下：

- 1.前端访问系统登录 API
- 2.成功以后，前端会接收后端响应的一个 Token，保存在本地
- 3.请求受保护 API，则采用自定义头部携带此 Token
- 4.后端检验 Token，成功则返回受保护的数据

访问受保护商场 API 采用自定义 X-Zhishe-Token 头部

- 1.前端访问小商城后端登录 API `/user/login`

```
POST /user/login

{
  "username": "admin",
  "password": "123456"
}
```

图 10-5 登录示例

- 2.成功以后，前端会接收后端响应的一个 Token

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0aGlzIGlzIGxpdGVtYWxsIHRva2VuIiwiaXNkIjoiaTUlOSUwFQUCIsImZcyI6IkkxJVEVNQUxMIiwiaXhwIjozNTUzMzI2ODUwLCJ1c2VySWQiOiJEsImldCI6MTU1NzIxOTY1MH0.XP0TuhupV_ttQsCr1KtAPZVlTbVzV0cnq_K0kXdbri0"
}
```

图 10-6 后端响应 Token 示例

3.请求受保护 API，则采用自定义头部携带此 Token

```
{
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0aGlzIGlzIGxpdGVtYWxsIHRva2VuIiwiaXNkIjoiaTUlOSUwFQUCIsImZcyI6IkkxJVEVNQUxMIiwiaXhwIjozNTUzMzI2ODUwLCJ1c2VySWQiOiJEsImldCI6MTU1NzIxOTY1MH0.XP0TuhupV_ttQsCr1KtAPZVlTbVzV0cnq_K0kXdbri0"
}
```

图 10-7 Token 示例

双重权限验证：

我们不仅要在接口上用注解控制访问权限，还可能要在业务逻辑中判断某个角色是不是对应该组织的角色。

举个例子：假如一个 A 社长在浏览器登录后，直接在浏览器地址栏输入删除 B 社团活动的链接，后端收到请求后，角色验证通过，但是 A 社长实际上不能操作其它社团的内容！

4. 2. 用户服务

接口名	接口说明	请求方法	请求参数/返回参数
/user/login	用户登录	POST	请求参数：用户名、密码 响应参数：登录是否成功， 成功返回 Token
/user/info	获取用户信息	GET	返回参数：用户 ID、角色、 权限、专业、Email、联系方式、 个性签名、头像地址
/users/register	用户注册	POST	请求参数：用户名、密码、

接口名	接口说明	请求方法	请求参数/返回参数
			个性签名、专业、联系方式、 宿舍地址 响应参数：成功返回 201
/users/:uid/question	根据用户 ID 获取密保问题，找回密码	GET	响应参数：密保问题
/users/:uid/answer	校验密保问题	POST	请求参数：用户 ID、待校验密保答案 响应参数：成功返回 204，失败返回 403
/user/info	用户修改个人信息	PUT	请求参数：密码、昵称、专业、联系方式、地址 响应参数：成功返回 204

表 4-1 用户服务接口设计表

4.3. 社团管理

接口名	接口说明	请求方法	请求参数/响应参数
/clubs/recommended	获取系统推荐的社团列表	GET	响应参数：成功返回 200，社团 ID、社团名字、社长 ID、社长姓名、社团图片地址
/clubs	查看社团列表	GET	响应参数：成功返回 200，社团 ID、社团名字、社长 ID、社长姓名、社团图片地址
/clubs/:name	根据名称关键字查找社团（模糊搜索）	GET	响应参数：成功返回 200，社团 ID、社团名字、社长 ID、社长姓名、社团图片地址
/clubs/:id	根据社团 ID 查找社团	GET	响应参数：成功返回 200，社团 ID、社团名字、社长 ID、

接口名	接口说明	请求方法	请求参数/响应参数
			社长姓名、社团图片地址
/clubs/:club/members	列出所有社团成员	GET	响应参数：成功返回 200，成员列表（用户名、昵称、头衔、角色、活跃度、头像地址）。如果没有社团成员返回 302
/clubs/:club/members/:userid	查看某个社团成员信息	GET	响应参数：成功返回 200，用户名、昵称、头衔、角色、活跃度、头像地址。未查找 到该成员返回 302
/clubs/:club/members/:userid	删除社团成员 (先经过身份验证)	DELETE	响应参数：成功返回 204

表 4-2 社团管理接口设计表

4.4. 申请与审核

接口名	接口说明	请求方法	请求参数/响应参数
/clubs/creations	普通学生提交社团创建申请表单	POST	请求参数：社团 ID、申请人姓名、现任社长 ID、新任社长 ID、附件 URL 响应参数：成功返回 201
/clubs/creations	管理员查看社团创建申请列表，进行进一步的审核	GET	响应参数：成功返回 200，申请列表（社团名称、创建时间、申请人姓名、申请原因、附件 URL、申请状态）
/clubs/creations/audit	管理员审核某个社团创建申请	PUT	请求参数：申请 ID、新申请状态 响应参数：成功返回 204

接口名	接口说明	请求方法	请求参数/响应参数
/clubs/dissolutions	社长提交解散社团申请表单	POST	请求参数：社团 ID、申请人姓名、申请原因、附件 URL， 成功返回 201
/clubs/dissolutions	管理员查看社团解散申请列表，进行进一步的审核	GET	响应参数：成功返回 200，申请列表（社团名称、创建时间、申请人姓名、申请原因、附件 URL、申请状态）
/clubs/dissolution/audit	管理员审核社团解散申请	PUT	请求参数：申请 ID、新申请状态 响应参数：成功返回 204
/clubs/join	学生提交入社申请	POST	请求参数：申请人 ID、社团 ID、申请原因 响应参数：成功返回 201
/clubs/:club/joins	社长查看入社申请列表，进行进一步的审核	GET	响应参数：成功返回 200，申请列表（申请人姓名、申请原因、申请时间、申请状态）
/clubs/joins/audit	社长审核入社申请	PUT	请求参数：申请 ID、新申请状态 响应参数：成功返回 204
/clubs/quit	社员提交退出社团表单	POST	请求参数：申请人 ID、社团 ID、申请原因 响应参数：成功返回 201
/clubs/:club/quit	社长查看成员退出社团通知列表	GET	响应参数：成功返回 200，退出社团通知列表（申请人姓名、申请原因、创建时间）
/clubs/leader/change	社长提交换届申请表单	POST	请求参数：社团 ID、旧社长 ID、新社长 ID、申请原因 响应参数：成功返回 201

接口名	接口说明	请求方法	请求参数/响应参数
/clubs/leader/changes	管理员查看社长换届申请列表	GET	响应参数：成功返回 200，申请列表（社团名称、旧社长姓名、新社长姓名、创建时间、申请状态）
/clubs/leader/changes	管理员审核社长换届申请	PUT	请求参数：申请 ID、新申请状态 响应参数：成功返回 204
/clubs/certifications	社长提交社团认证申请表单	POST	请求参数：社团 ID、申请人姓名、附件 URL 响应参数：成功返回 201
/clubs/certifications	管理员查看社长换届申请列表	GET	响应参数：成功返回 200，申请列表（社团名称、申请人姓名、附件 URL、创建时间、申请状态）
/clubs/certifications	管理员审核社团认证申请	PUT	请求参数：申请 ID、新申请状态 响应参数：成功返回 204

表 4-3 申请与审核接口设计表

4.5. 公告

接口名	接口说明	请求方法	请求参数/响应参数
/clubs/:club/bulletins	社长发布公告	POST	请求参数：公告标题、公告内容 响应参数：成功返回 201
/clubs/:club/bulletins	查看公告列表	GET	响应参数：成功返回 200，公告列表（公告标题、公告内容、创建时间）
/clubs/:club/bulletins	查看公告详情	GET	响应参数：成功返回 200，公

接口名	接口说明	请求方法	请求参数/响应参数
/:bulletin_id			告标题、公告内容、创建时间
/clubs/:club/bulletins/:bulletin_id	修改公告内容	PUT	请求参数：公告标题、公告内容 响应参数：成功返回 204
/clubs/bulletins/:bulletin_id	删除公告	DELETE	响应参数：成功返回 204

表 4-4 公告接口设计表

4. 6. 活动

接口名	接口说明	请求方法	请求参数/响应参数
/clubs/activities	社长申请活动	POST	请求参数：社团 ID、活动名称、活动标题、活动内容、开始时间、结束时间、活动地点 响应参数：成功返回 201
/clubs/activities/audit	管理员审核社团活动申请	PUT	请求参数：申请 ID、申请状态 响应参数：成功返回 204
/clubs/activities/state	社长修改社团活动状态	PUT	请求参数：申请 ID、申请状态 响应参数：成功返回 204
/clubs/activities/:id	社长修改社团活动	PUT	请求参数：活动名称、活动标题、活动内容、开始时间、结束时间、活动地点 响应参数：成功返回 204
/clubs/activities/:id	社长删除活动 (先验证权限)	DELETE	响应参数：成功返回 204

表 4-5 活动接口设计表

4. 7. 活动论坛

接口名	接口说明	请求方法	请求参数/响应参数
/forum/posts	查看活动论坛	GET	响应参数：成功返回 200， 活动列表（活动标题、活动内容、创建时间）
/forum/posts/:id	查看某个活动帖子	GET	响应参数：成功返回 200， 活动列表（活动标题、活动内容、创建时间，评论列表（评论用户名、评论内容、创建时间））
/forum/posts/:id	删除帖子	DELETE	响应参数：成功返回 204
	修改帖子		

表 4-6 活动论坛接口设计表

4. 8. 活动评论

接口名	接口说明	请求方法	请求参数/响应参数
/forum/posts/:id/rem arks	对某一帖子发表 评论	POST	请求参数：用户 ID、帖子 ID、 评论内容 响应参数：成功返回 201

表 4-7 活动评论接口设计表

5. 系统安全和权限设计

5. 1. 安全性设计说明

为了满足系统系统对权限和安全管理方面的要求，系统的设计主要满足以下几个安全设计

- 前后端通过 JWT 访问验证，确保系统的安全性。

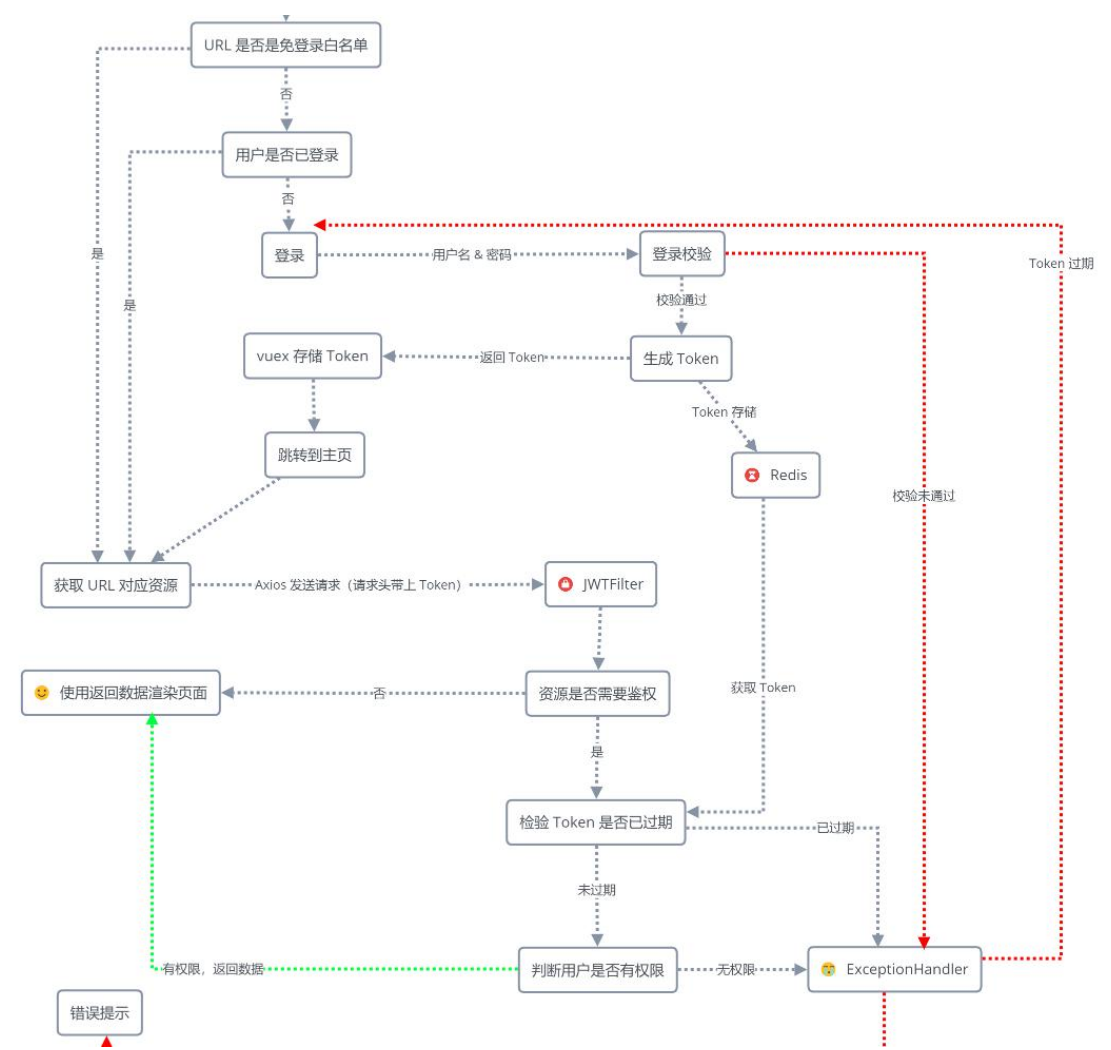


图 11-1 JWT 访问验证

- 每个普通用户登陆时按照其学号作为账号，管理员使用统一的账号密码进行登陆，即可进入到各自对应的子系统中
- 满足用户权限和管理可配置性，可以将系统内各功能模块（页面）灵活的划分给用户，使他们具有相应功能操作的权限
- 不同的身份进入到的社团页面是不同的，社长和社员则具有游客具有的所有权限，不同角色分配不同的页面、按钮、目录保证系统的安全性。示例如下：



图 12-1 社员页面



图 12-2 社长页面



图 12-3 游客页面

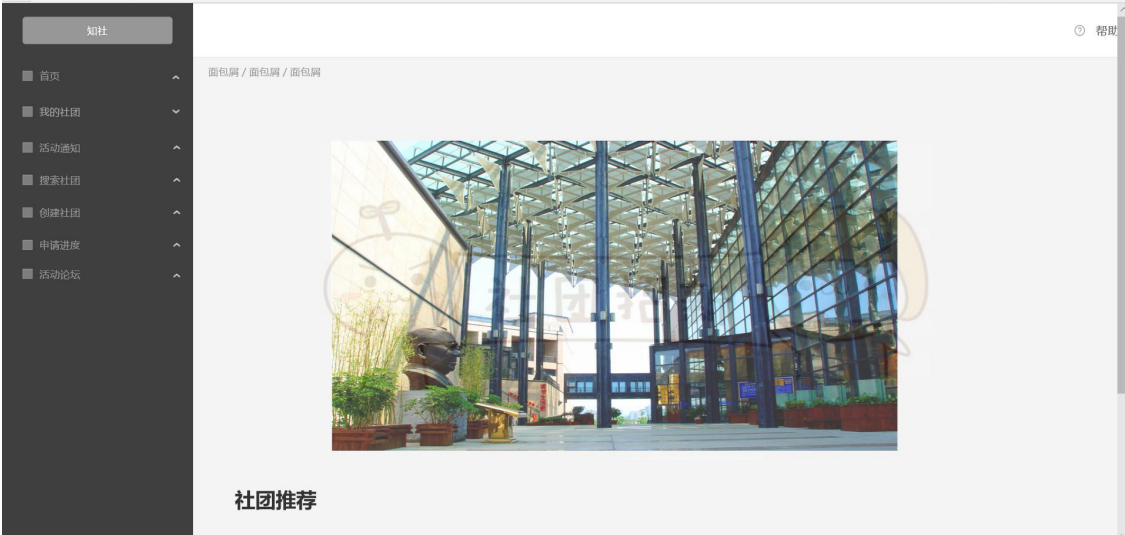


图 12-4 学生页面



图 12-5 管理员页面

5.2. 权限设计说明

权限管理是用户进行系统权限划分和设置的平台，按照管理的层次可划分为三个层面进行管理，用户、角色和权限。

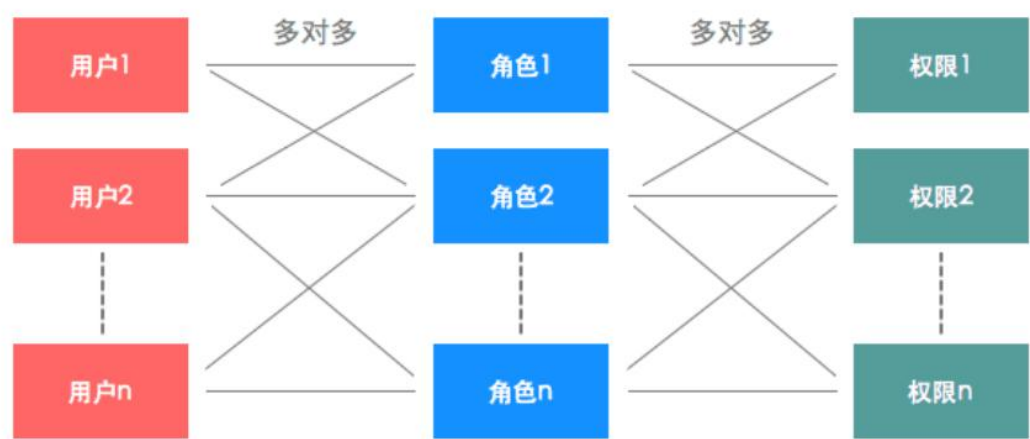


图 13-1 权限设计

1.用户

应用系统的具体操作者，用户可以自己拥有权限信息，可以归属于 0~n 个角色，可属于 0~n 个组。他的权限集是自身具有的权限、所属的各角色具有的权限、所属的各组具有的权限的合集。它与权限、角色、组之间的关系都是 n 对 n 的关系。

通过系统用户注册功能实现登录用户的信息保存。为保证各系统间采用统一的登录验证，可采用统一的用户信息表的办法，保证统一的用户信息数据来源。

用户表可包括:用户 ID、用户名、密码、角色信息、其他相关信息。

2.角色

为了对许多拥有相似权限的用户进行分类管理，定义了角色的概念，例如系统管理员、管理员、用户、访客等角色。角色具有上下级关系,可以形成树状视图，父级角色的权限是自身及它的所有子角色的权限的综合。父级角色的用户、父级角色的组同理可推。

与用户的关系为多对多，与权限的关系为多对多

在我们的系统中，可具体细分为五类角色，他们分别是：游客、普通学生、社员、社长、管理员。

3.权限

系统所有的权限信息，具有上下级关系，是一个树形结构

本系统拥有的所有权限表如下图所示

id	menu_code	menu_name	permission_code	permission_name
1	member	成员管理	member:list	列表
2	member	成员管理	member:update	修改
3	member	成员管理	member:delete	删除
4	member	成员管理	member:add	新增
7	member_exit	成员退出	member_exit:list	列表
8	position_applied	申请职位	position_applied:add	新增
9	announcement	公告管理	announcement:add	新增
10	announcement	公告管理	announcement:list	列表
11	announcement	公告管理	announcement:delete	删除
12	announcement	公告管理	announcement:update	修改
13	activity	活动管理	activity:list	列表
14	activity	活动管理	activity:add	新增
15	activity	活动管理	activity:delete	删除
16	activity	活动管理	activity:edit	修改
17	club	社团管理	club:list	列表
18	club	社团管理	club:delete	删除
19	club	社团管理	club:add	新增
20	club	社团管理	club:update	修改
21	remark	评论管理	remark:delete	删除

id	menu_code	menu_name	permission_code	permission_name
22	remark	评论管理	remark:add	新增
23	remark	评论管理	remark:list	列表
24	forum	论坛管理	forum:list	列表
25	forum	论坛管理	forum:add	新增
26	forum	论坛管理	forum:delete	删除
27	user	用户管理	user:add	新增
28	user	用户管理	user:delete	删除
29	user	用户管理	user:update	修改
30	user	用户管理	user:list	列表
31	apply	申请管理	apply:list	列表
32	apply	申请管理	apply:add	新增
33	apply	申请管理	apply:update	修改
34	apply	申请管理	apply:delete	删除
35	role	角色管理	role:list	列表
36	role	角色管理	role:add	新增
37	role	角色管理	role:update	修改
38	role	角色管理	role:delete	删除
39	authority	权限管理	authority:list	列表
40	authority	权限管理	authority:add	新增
41	authority	权限管理	authority:delete	删除
42	authority	权限管理	authority:update	修改

综上，系统权限应该有以下功能

- ①用户身份登陆
- ②子系统权限划分
- ③角色划分
- ④用户权限分配

4.安全管理

用户登陆后，可以通过将用户信息写入记录的 `session` 中，由后端取出 `session` 中的值进行判断来知道用户的身份从而分配不同的角色

5.数据表设计

权限管理部分有五个数据库表：用户表、角色表、权限表、用户角色相关表、角色权限相关表

用户表：记录用户信息

角色表：记录系统中拥有的角色信息

权限表：记录系统中拥有的所有权限信息

用户角色相关表：由于用户与角色是多对多关系，因此用此表来关联用户表和角色表

角色权限相关表：由于角色与权限是多对多关系，因此用此表来管理角色表和权限表