



集美大学

JIMEI UNIVERSITY

题目 基于 Node.js+React 的
高校社团管理系统的设计与实现

学院	<u>计算机工程学院</u>	专业	<u>软件工程</u>
年级	<u>2019 级</u>	班级	<u>软件 1911</u>
姓名	<u>柴颖</u>	学号	<u>201921122025</u>
手机号	<u>19859252969</u>	邮箱	<u>doufukeai@163.com</u>

2023 年 6 月

目 录

第 1 章 背景与框架	1
1.1 项目背景	1
1.2 项目框架	1
1.2.1 前端	1
1.2.2 后端	2
第 2 章 系统需求分析	4
2.1 需求概述	4
2.2 参与者	5
2.3 用例图	6
2.3.1 管理员与超级管理员	6
2.3.2 社长与副社长.....	7
2.3.3 社团成员.....	8
2.3.4 普通用户.....	9
2.3.5 游客	10
2.4 用例规约	11
2.4.1 活动申请.....	11
2.4.2 活动审核.....	12
2.4.3 社团维护.....	13
2.4.4 活动发布.....	13
2.4.5 参加活动.....	14
2.4.6 活动成员管理.....	15
2.4.7 活动准备.....	16
2.4.8 发布活动通知.....	17

2.4.9 更新活动阶段.....	18
2.4.10 发布活动签到.....	19
2.4.11 人员评价.....	19
2.4.12 活动总结.....	20
2.4.13 社团解散.....	21
第3章 系统功能设计	23
3.1 用户管理与权限管理.....	23
3.2 社团管理	24
3.3 活动管理	25
第4章 类设计	27
4.1 类的设计	27
4.1.1 RBAC 权限管理类.....	27
4.1.2 社团管理类.....	28
4.1.3 活动管理类.....	28
4.1.4 系统类图.....	29
4.2 系统逻辑结构	30
第5章 数据库设计	32
5.1 类图转 ER 图	32
5.1.1 映射对象.....	32
5.1.2 继承关系映射.....	32
5.1.3 关联和聚合映射.....	33
5.2 ER 图	34
5.3 格式设计	34
5.4 表结构设计	35

第1章 背景与框架

1.1 项目背景

随着校园文化的不断丰富，校园里的社团越来越多，有关社团的各种信息也成倍增长。很多社团都是用手工记录的方式和聊天 App 来管理社团内部事务，例如社团在招新时用纸张记录添加成员，在开展活动时用聊天 App 记录活动情况等，然后再进行总结和存档；而随着社团数量的增加，这些手写记录和聊天 App 弊端也愈发明显，主要表现在不能及时地上传下达、学生意见的收集相对困难、活动批准时间较长、社团的相关文档传阅不方便等，这种管理方式就容易出现信息出错或丢失等缺点，而且时间一长，数据过多就会使以后查找信息及更新变得相当麻烦，因此很有必要开发一个高校社团管理系统来提高效率，实现社团成员基本信息的自动化管理。

高校社团管理系统方便社团的管理与宣传，可以有效地整合优质的线上线下资源。高校社团管理系统在管理方面，使社团的管理工作更加系统规范，信息的处理速度和准确性更高，方便社团信息资料归档，方便学校管理社团组织、方便社长安排活动，有效提升社团的管理效率。在学生方面，让大学生对社团及活动的开展有更加清晰的认识并能快速了解，增加对社团的知晓情况，提高学生参加社团积极性，增强社团与成员之间有效的沟通，从而丰富大学生的社团生活。

1.2 项目框架

高校社团管理系统分为前端及后端部分。

1.2.1 前端

前端使用的架构主要有 React + React-Router V6 + Ant Design + MobX。

React 是一个用于构建用户界面的 JavaScript 库，它可以帮助开发人员快速构建可复用的组件。React 的组件化思想使得开发人员能够将复杂的界面拆分

成多个独立的组件，从而提高代码的可维护性和可复用性。

React-Router V6 是一个用于管理 React 应用程序中路由的库，它可以帮助开发人员快速构建单页应用程序。在单页应用程序中，用户在浏览不同页面时不需要重新加载整个页面，而是通过动态更新页面内容来实现页面切换。这样可以提高用户体验，并且减少服务器压力。

Ant Design 是一个基于 React 的 UI 组件库，它提供了丰富的组件，可以帮助开发人员快速构建漂亮的界面。Ant Design 的组件都遵循统一的设计规范，并且提供了丰富的配置选项，可以满足不同场景下的需求。

Mobx 是一个状态管理库，它可以帮助开发人员管理应用程序中的状态。在复杂的应用程序中，状态管理是一个非常重要的问题。如果状态管理不当，会导致代码难以维护和调试。Mobx 通过提供一些简单易用的 API 来帮助开发人员管理状态，从而提高代码的可维护性。

这个框架将 React + React-Router V6 + Ant Design + MobX 结合起来，可以帮助开发人员快速构建高效、可维护的前端应用程序。它提供了一整套完整的解决方案，涵盖了前端开发中的各个方面，包括组件化开发、路由管理、状态管理和 UI 设计等。

1.2.2 后端

后端的主要技术架构为 Node.js + Express + MySQL。

Node.js 是一个开源的、跨平台的 JavaScript 运行时环境，它作为服务端，不用担心线程问题，而且具有高性能 I/O 用于实时应用，方便系统功能的聊天实现。

Express 是一个基于 Node.js 的 Web 开发框架，它提供了丰富的 API 和中间件，可以帮助开发人员快速构建 Web 应用程序。它的设计简洁、易用，可以大大提高开发效率。

MySQL 是一个流行的关系型数据库管理系统，它提供了丰富的 SQL 语言支持，可以帮助开发人员快速构建复杂的数据模型。它具有高性能、可扩展性强等特点，可以满足大多数应用程序的需求。

使用 Node.js + Express + MySQL 搭建后端可以提高开发效率、简化开发流

程，并且能够构建高性能、可扩展的应用程序。

第2章 系统需求分析

2.1 需求概述

(1) 用户管理

实现用户的注册、个人信息维护、注销等功能。对用户账号密码加密，防止信息泄露；对非该系统用户进行严格的隔离。

(2) 权限管理

权限管理分为菜单管理和角色管理两大类。

- 菜单管理：为用户展示具有层次的菜单，方便用户使用系统或管理系统。
- 角色管理：为用户分配角色，让角色代表用户在系统中操作。不同的角色拥有的功能不一样，角色等级越高，系统的功能越多。

(3) 场地管理

为社团或活动提供场地来源，保证它们都有空间施展，也防止它们之间出现场地冲突的现象。

(4) 社团管理

社团管理包括以下功能：

- 社团类型管理：定义社团类型，如文化类社团、娱乐类社团、运动类社团等等类型。
- 社团筹建：由想要成立新社团的人填写社团的相关信息，并提交到系统中，等待回复。管理员审批所有申请记录，并回复审批结果。
- 社团维护：更新社团信息，使社团始终保持在最新状态。
- 社团成员：展示所有社团成员的信息；社长可以管理社团成员的加入，退社，还能把加入社团的功能展示给所有用户。
- 学期总结：每半年总结一次社团的各种发展或成就。
- 费用管理：社团日常消费记录。

- 社团通知：给社团成员发布通知。

(5) 活动管理

活动管理是活动从发布到活动结束的整个过程，因此，活动管理分为以下功能。

- 活动类型管理：活动的类型是活动的关键词或代表。
- 活动申请审批：申请者填写活动信息，如活动的组织方式、活动目的、活动地点等等信息。管理员审核活动是否符合开展要求，并回复审批结果。
- 活动发布：发布活动的相关信息，显示浏览次数。
- 活动成员管理：展示参与活动的所有用户信息。
- 活动准备：预先准备好活动过程中所需要的道具、场地、工作人员。
- 活动过程管理：是管理活动各阶段的信息，可以分为以下功能：
- 活动通知：由活动负责人发布活动最新要求。
- 活动阶段发布：发布最新阶段的活动要求，进一步管理活动的进程。
- 活动签到：记录现场有多少人，并核对成员是否缺少。
- 人员评价：评价每一个活动参与者在此次活动的表现。
- 活动消费记录：记录活动中的每一笔开销。
- 活动总结：当活动结束后，活动负责人需要对此此次活动作总结报告，方便以后查询浏览。
- 查看历史活动信息：已经结束的活动。

2.2 参与者

从直接使用系统的人员中发现参与者。当系统出现问题时，管理员负责维护系统并修复 bug。在社团从创建到发展的过程中，社长扮演着重要的角色。但是，社团不仅仅需要社长，还需要社团成员的参与。此外，每一个在系统中注册的用户都是普通用户。

从管理上看，管理员需要管系统的方方面面，会导致管理员心有余而力不足，因此我们可以在管理员身上分出一部分功能。社长的职责也需要划分，因此根据职责不同可以将参与者划分为以下几种：

- 超级管理员：管理所有功能。

- 管理员：负责活动管理。
- 社长：社团负责人。
- 副社长：社团的活动负责人。
- 社团成员：已加入社团的用户。
- 普通用户：已在系统注册的用户。
- 游客：未在系统注册的用户。

2.3 用例图

2.3.1 管理员与超级管理员

负责高校社团管理的主要使用者是管理员与超级管理员。其中，超级管理员继承管理员，可以管理管理员的所有工作。

管理员主要负责活动管理。管理员管理活动类型的信息维护、类型的添加与删除；审核社长或副社长发来的活动请求，并回复审核结果；浏览社团发布的所有活动，包括已经结束的活动。

超级管理员负责用户管理、权限管理、社团管理和场地管理。超级管理员负责更新、添加和删除用户信息，并在必要时更改密码。超级管理员需要为每个角色分配菜单，以便为用户分配不同的权限。社团管理和活动管理类似，都需要审核社团申请、更新类型并浏览所有社团。场地管理主要是提供社团场地和活动场地的来源。

根据上面的分析，管理员与超级管理员的用例图如图 2-1 所示。

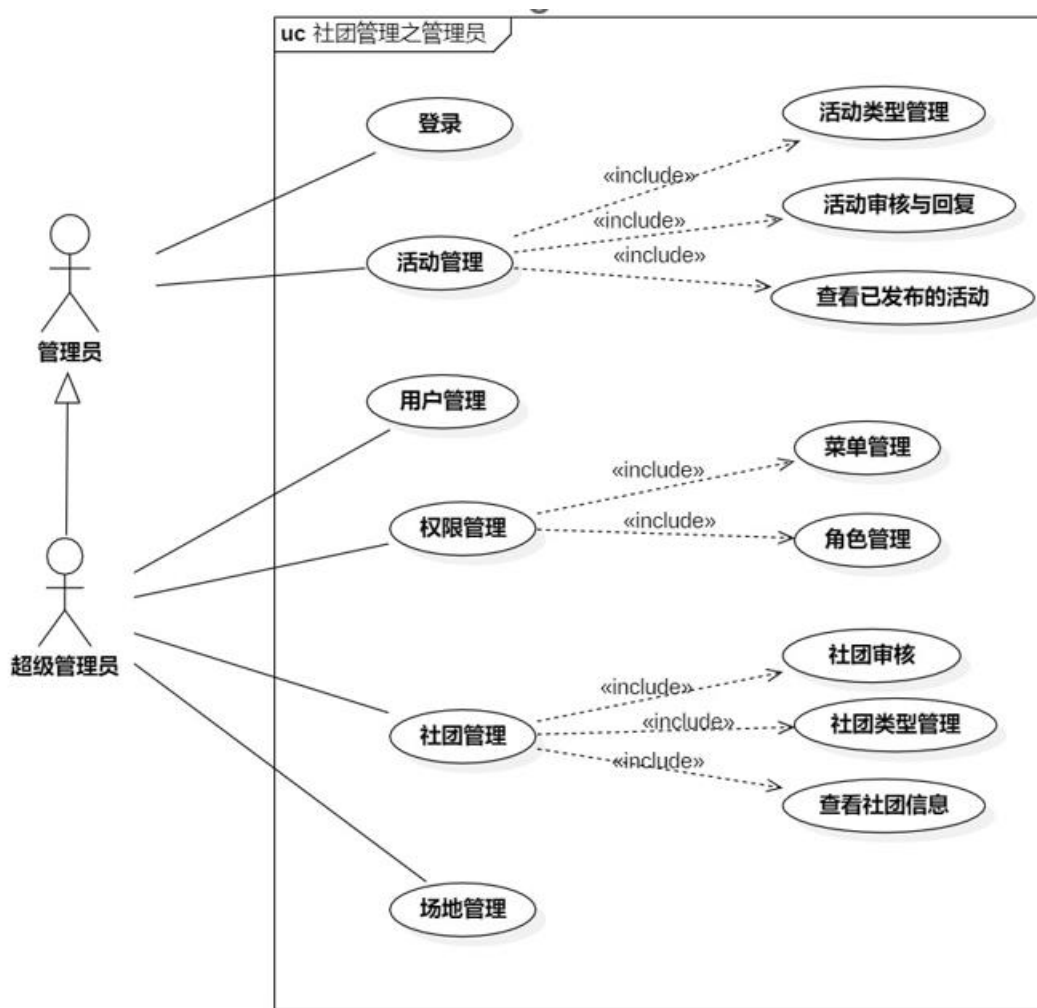


图2-1 管理员与超级管理员用例图

2.3.2 社长与副社长

高校社团管理系统的主要运行来源于社长与副社长，他们是系统的中坚力量，维持系统的正常发展。社长继承副社长，可以管理副社长的所有功能。

副社长主要负责社团活动方面。从活动的申请开始，副社长需要描述活动的大体过程，并需要指定指导老师，之后等待管理员回复。副社长等到审核通过后，便能正式发布活动，召集用户参与活动。活动过程中用到的相关道具，需要记录在系统中，方便以后查询。活动开展后，需要划分活动的各个阶段，每一阶段必须有活动的相关描述，以便活动能正常举行。活动结束后，需评价参与者表现并总结本次活动。

社长负责社团管理，包括申请、目标设定和成就总结。申请时，需描述社团的目标、贡献和具体计划。成立后，需定期总结社团成就，包括活动举办情况、成员

发展和对学校的贡献。此外，社长还需负责协调社团内部关系，维护社团和谐稳定。

根据上述的描述信息，社长与副社长的用例图如图 2-2 所示。

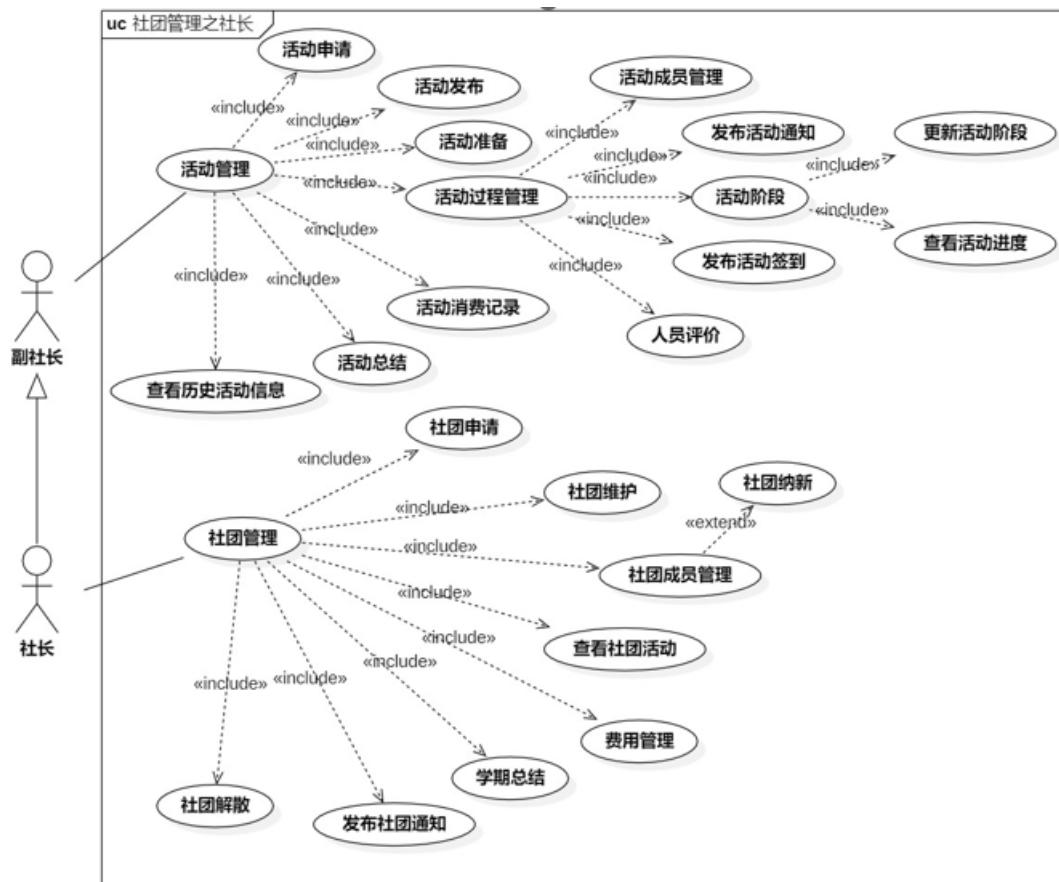


图2-2 社长与副社长用例图

2.3.3 社团成员

社团成员的主要职责包括参与社团组织的各类活动，接收来自社团的通知信息，以及查看社团过去取得的成就。这些职责可以通过如图 2-3 所示的用例图展示。

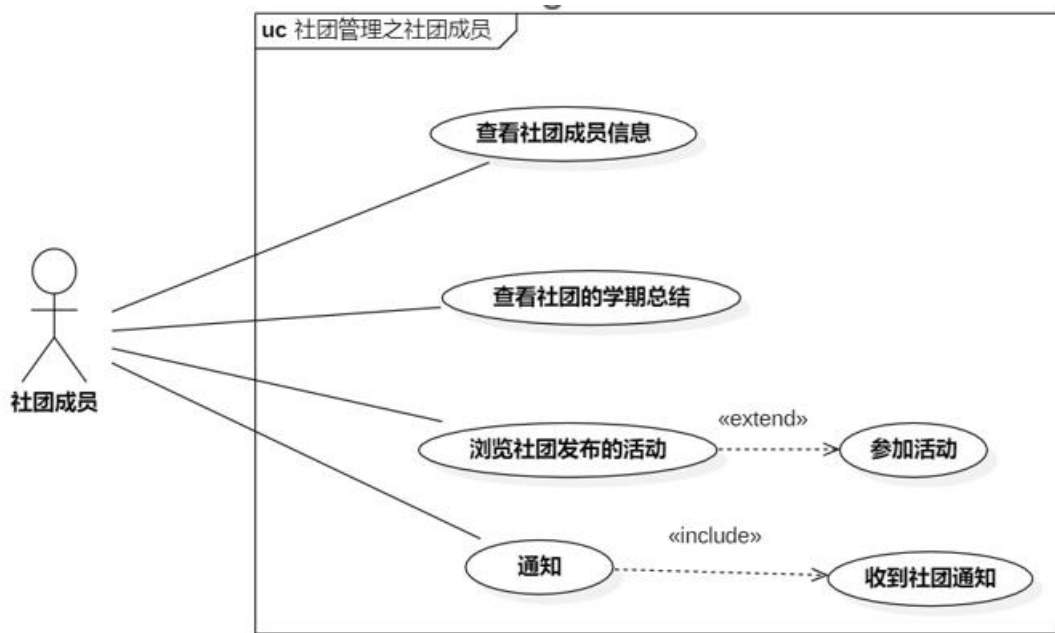


图2-3 社团成员用例图

2.3.4 普通用户

当普通用户登录系统后，他们可以查看并更新自己的个人信息。如果他们想要创建一个新的社团，可以提交申请表并等待管理员的回复。此外，用户还可以加入一个或多个社团来丰富自己的生活。如果用户想要参加公开活动，可以浏览最新发布的活动并选择自己喜欢的活动。在参加活动后，用户可以在活动中结识新朋友，并查看他们的个人信息以更好地了解他们。活动结束后，用户可以在历史活动中查看自己之前参加过的活动并了解自己的表现。

根据这些信息，可得出普通用户的用例图，如图 2-4 所示。

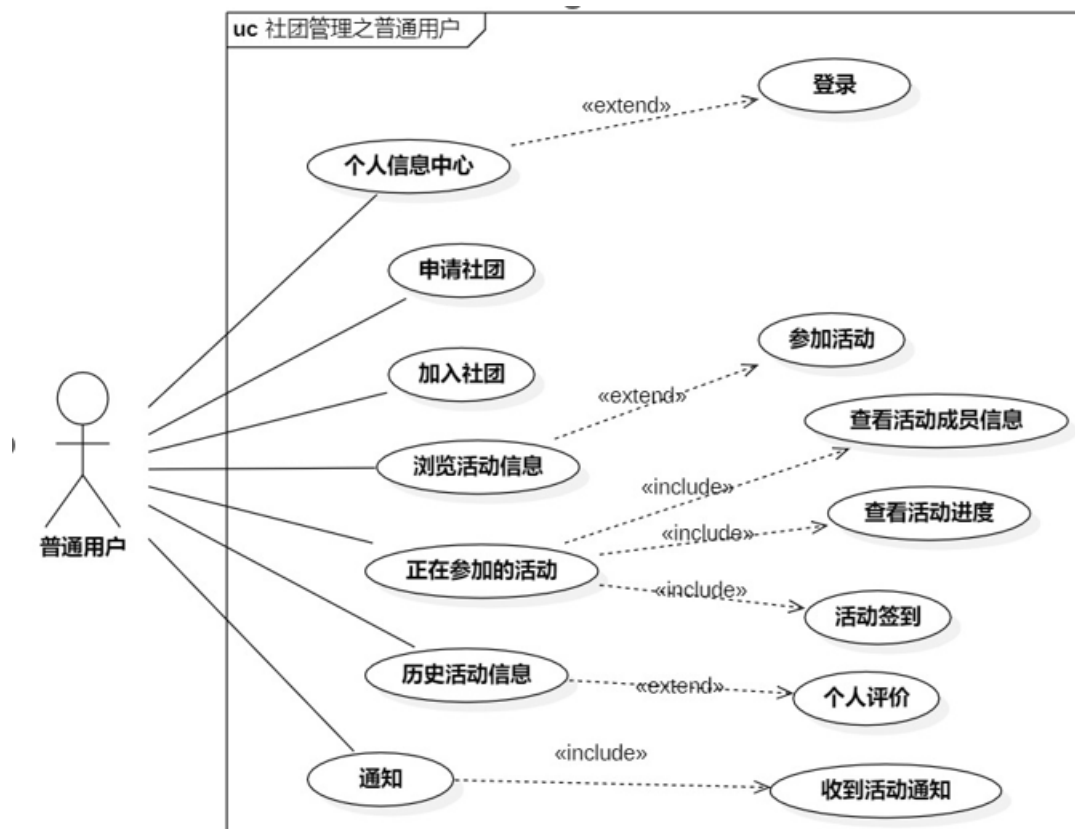


图2-4 普通用户用例图

2.3.5 游客

游客如果想要参与活动，必须在系统中注册用户，不然只能浏览活动信息，用例如图 2-5 所示。

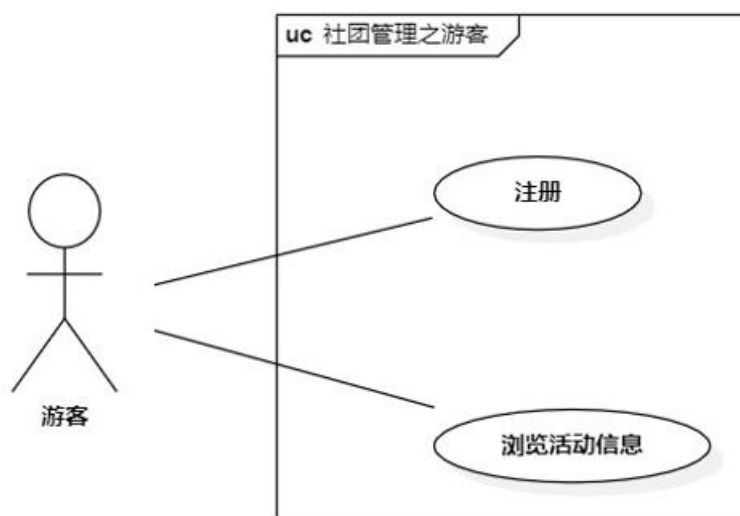


图2-5 游客用例图

2.4 用例规约

本系统用例过多，论文篇幅有限制，因此用例模型中的用例规约只展示核心部分的用户。

社团管理与活动管理都是系统最主要的部分，当社团管理和活动管理的用例在用例规约中差不多时，基本以活动管理的用例为主作为展示。

2.4.1 活动申请

社团申请和活动申请的用户规约都差不多，因此社团申请和活动申请只选其中一个用例，如活动申请。

活动申请是活动的开端，它需要填写与活动相关的信息，如活动的组织方式、活动的时间、地点等等信息，活动申请用例的用户规约如表 2-1 所示。

表2-1 活动申请

属性	内容
用例编号	UC01
用例名称	活动申请
用例简述	副社长填写活动信息后，提交申请
参与者	副社长
前置条件	副社长拥有活动申请权限，且成功登录系统
后置条件	活动申请记录保存在系统数据中心
基本流	1. 副社长填写活动的基本信息。 2. 副社长选择申请。 3. 系统保存申请记录，并提示：“等待管理员审核批准，并回复”。 4. 收到回复通知，申请通过。
备选流	2a. 信息填写错误 1. 系统提示：“格式错误”，并返回到基本流步骤 1。 2. 系统提示：“信息不完整”，并返回到基本流步骤 1。

表 2-1 续表

属性	内容
备选流	4a. 申请不通过 1. 收到回复：“活动不符合要求，请重新申请”，并返回到基本流步骤 1。
成功场景	活动申请通过
失败场景	活动申请失败

2.4.2 活动审核

社团审核和活动审核的用例规约基本一致，因此选择活动的活动审核作为用例规约展示。

活动审核是审核活动是否符合学校要求，是否适合组织开展。活动审核用例的用例规约如表 2-2 所示。

表2-2 活动审核

属性	内容
用例编号	UC02
用例名称	活动审核
用例简述	管理员审核活动申请，并通知申请者结果
参与者	管理员
前置条件	管理员拥有活动审核权限，且成功登录系统
后置条件	系统保存活动批改记录
基本流	1. 管理员收到活动申请提示，并选择查看。 2. 管理员根据学校要求审核活动，并回复申请者审核结果。 3. 系统保存审核结果。
备选流	无
成功场景	无
失败场景	无

2.4.3 社团维护

当社团的展示信息需要更改时,就需要社长更新社团信息。社团维护用例的用例规约如表 2-3 所示。

表2-3 社团维护

属性	内容
用例编号	UC03
用例名称	社团维护
用例简述	更新社团信息
参与者	社长
前置条件	社长登录系统
后置条件	系统保存社团信息
基本流	1. 社长打开维护界面。 2. 社长填写更新内容, 并选择更新。 3. 系统保存社团信息, 并提示: “社团更新成功”。
备选流	2a. 社长取消更新, 退出界面, 结束用例
成功场景	社团更新成功
失败场景	社团更新失败

2.4.4 活动发布

活动发布代表着活动即将开始, 活动的各类信息也会展示在系统上。活动发布用例的用例规约如表 2-4 所示。

表2-4 活动发布

属性	内容
用例编号	UC04
用例名称	活动发布
用例简述	副社长发布活动, 系统展示活动的各类信息
参与者	副社长
前置条件	副社长拥有活动发布权限, 且成功登录系统

表 2-4 续表

属性	内容
后置条件	系统保存活动信息
基本流	1. 副社长收到活动申请通过。 2. 副社长选择发布活动。 3. 系统保存活动信息，并提示：“活动发布成功”。
备选流	无
成功场景	活动发布成功
失败场景	无

2.4.5 参加活动

社团纳新和参加活动的用例规约基本一致，因此选择参加活动的用例规约作为展示。

参加活动是每个用户都拥有的权利，用户可以浏览各项活动，并选择参加。参加活动用例的用例规约如表 2-5 所示。

表2-5 参加活动

属性	内容
用例编号	UC05
用例名称	参加活动
用例简述	浏览各项活动信息，并选择参加
参与者	用户
前置条件	用户成功登录系统
后置条件	该活动添加用户信息
基本流	1. 用户浏览各项活动信息。 2. 用户选择喜欢的活动，并选择参加。 3. 活动保存该用户信息，并提示：“用户参加活动成功”。
备选流	2a. 活动参加失败 1. 用户取消参加，并返回到基本流步骤 1 中。

表 2-5 续表

属性	内容
备选流	2. 系统提示：“活动人数已满”，并返回到基本流步骤 1 中。
成功场景	成功参加活动
失败场景	活动参加失败

2.4.6 活动成员管理

社团成员管理和活动成员管理的用例规约差不多，因此也选择活动成员管理作为用例规约展示。

活动成员管理保存着所有参加这项活动的用户信息。副社长可以手动添加活动成员，也可以手动让用户退出该项活动。活动成员用例的用例规约如表 2-6 所示。

表2-6 活动成员管理

属性	内容
用例编号	UC06
用例名称	活动成员管理
用例简述	活动成员保存着所有参加这项活动的用户信息
参与者	参加活动的人，副社长
前置条件	用户成功登录系统，副社长拥有操作活动的权限
后置条件	系统保存添加或删除参加活动的人的信息
基本流	1. 参加活动的人、副社长浏览活动成员信息。 2. 副社长选择添加用户，并填写用户信息。 3. 系统更新活动成员信息，并提示：“添加用户成功”。 4. 系统给新参加活动的人发送通知：“成功参加这项活动”。 5. 副社长选择删除用户，并选择是那个用户。 6. 系统更新活动成员信息，并提示：“删除用户成功”。
备选流	3a. 添加用户失败 1. 副社长取消添加，并返回到基本流步骤 1 中。

表 2-6 续表

属性	内容
备选流	2. 系统提示：“活动人数已满”，并返回到基本流步骤 1 中。 3. 系统提示：“该用户已存在”，并返回到基本流步骤 1 中。
成功场景	1. 成功添加用户。 2. 成功删除用户。
失败场景	添加用户失败。

2.4.7 活动准备

在开展活动之前，活动负责人需要考虑好，此次活动都需要些什么，如工作人员、道具。活动准备用例的用例规约如表 2-7 所示。

表2-7 活动准备

属性	内容
用例编号	UC07
用例名称	活动准备
用例简述	开展活动前的准备
参与者	副社长
前置条件	副社长拥有操作活动的权限
后置条件	系统保存活动信息
基本流	1. 副社长选择导入道具的消费记录，并填写消费信息。 2. 系统保存活动消费信息。 3. 副社长选择删除消费记录，并选择其中一条消费记录。 4. 系统更新活动消费信息，并提示：“删除成功”。 5. 副社长选择添加活动职位，并填写活动职位的相关信息。 6. 系统更新活动信息，并提示：“添加职位成功”。 7. 副社长选择删除活动职位，并选择其中一个职位。 8. 系统更新活动信息，并提示：“删除职位成功”。 9. 副社长选择分配活动工作人员，显示有哪些可分配职位。

表 2-7 续表

属性	内容
基本流	10. 系统更新活动信息，并提示：“分配成功”。
备选流	6a. 添加职位失败：系统提示：“已经存在相同的职位”，并返回到基本流步骤 5 中。 10a. 分配职位失败：系统提示：“该职位人数已满”，并返回到基本流步骤 9 中。
成功场景	1. 成功添加消费记录。 2. 成功删除消费记录。 3. 成功添加职位 4. 成功删除职位 5. 成功分配职位

2.4.8 发布活动通知

发布社团通知和发布活动通知的用例规约基本一致，因此也选择发布活动通知的用例规约作为展示。发布活动通知用例的用例规约如表 2-8 所示。

表2-8 发布活动通知

属性	内容
用例编号	UC08
用例名称	发布活动通知
用例简述	通知活动的所有成员该项活动的最新要求
参与者	副社长
前置条件	副社长拥有操作活动的权限
后置条件	系统保存通知信息
基本流	1. 副社长选择发布活动通知。 2. 副社长填写通知内容，并选择发布。 3. 系统通知活动的所有成员。
备选流	2a. 副社长取消发布，系统结束操作，用例结束。

表 2-8 续表

属性	内容
成功场景	活动通知成功
失败场景	无

2.4.9 更新活动阶段

活动阶段是活动过程管理的核心，它是安排活动的所有流程。更新活动阶段有三个操作：更新、添加、删除。更新活动阶段用例的用例规约如表 2-9 所示。

表2-9 更新活动阶段

属性	内容
用例编号	UC09
用例名称	更新活动阶段
用例简述	发布最新阶段的活动要求，进一步管理活动的进程
参与者	副社长
前置条件	副社长拥有操作活动的权限
后置条件	系统更新活动信息
基本流	<ol style="list-style-type: none"> 1. 副社长查看活动进度。 2. 副社长选择更新某个活动阶段信息。 3. 副社长填写活动阶段信息，并选择更新。 4. 系统更新活动阶段信息。 5. 副社长选择增加活动阶段。 6. 副社长填写该阶段的活动要求，并选择发布。 7. 系统更新活动阶段信息。 8. 副社长选择删除活动阶段。 9. 副社长确认删除该阶段。 10. 系统更新活动阶段信息。
备选流	3a、6a、9a. 副社长取消该操作，并返回到基本流步骤 1 中。
成功场景	1. 增加活动阶段。

表 2-9 续表

属性	内容
成功场景	2. 更新活动阶段。 3. 删除活动阶段
失败场景	无

2.4.10 发布活动签到

记录现场有多少人，并核对成员是否缺少，它的用例规约如表 2-10 所示。

表2-10 发布活动签到

属性	内容
用例编号	UC10
用例名称	发布活动签到
用例简述	记录现场有多少人，并核对成员是否缺少
参与者	副社长
前置条件	副社长拥有操作活动的权限
后置条件	通知所有活动成员
基本流	1. 副社长浏览活动阶段信息，并选择发布签到。 2. 副社长填写签到内容，并选择发布。 3. 等待活动成员签到。 4. 系统更新签到信息，并统计成员报道情况。
备选流	2a. 副社长取消操作，系统结束操作，用例结束。
成功场景	发布签到成功
失败场景	无

2.4.11 人员评价

当活动结束后，活动负责人需要评价每个参加活动的人在此次活动的表现。人员评价用例的用例规约如表 2-11 所示。

表2-11 人员评价

属性	内容
用例编号	UC11
用例名称	人员评价
用例简述	评价每个人的表现
参与者	副社长
前置条件	副社长拥有操作活动的权限
后置条件	用户收到活动负责人对他们的评价
基本流	1. 活动结束后，副社长浏览活动成员信息。 2. 副社长选择人员评价，并填写评价内容。 3. 系统更新活动信息，并把评价内容发给用户。
备选流	2a. 副社长取消操作，系统结束操作，用例结束。
成功场景	评价成功
失败场景	无

2.4.12 活动总结

学期总结和活动总结的用例规约基本一致，唯一有点区别的是学期总结是社团每一个学期做一次总结，而活动总结是活动结束后做的总结，因此也选择活动总结作为用例规约展示。

每当活动结束后，副社长都要为活动做一次总结。总结活动的过程，记录活动的精彩画面。活动总结用例的用例规约如表 2-12 所示。

表2-12 活动总结

属性	内容
用例编号	UC12
用例名称	活动总结
用例简述	总结活动的过程，记录活动的精彩画面
参与者	副社长
前置条件	副社长拥有操作活动的权限

表 2-12 续表

属性	内容
后置条件	系统更新活动信息
基本流	1. 活动结束后，副社长浏览活动信息。 2. 副社长选择活动总结，填写相关内容，并选择提交。 3. 系统更新活动信息。
备选流	2a. 副社长取消操作，系统结束操作，用例结束。
成功场景	活动总结成功
失败场景	活动总结失败

2.4.13 社团解散

当社团发展不顺利，成员稀少，活动也很少时，社长可以申请解散社团。这样可以释放社团的相关资源，为其他社团的发展提供更多的机会。社团解散用例的用例规约如表 2-13 所示。

表2-13 社团解散

属性	内容
用例编号	UC13
用例名称	社团解散
用例简述	解散社团，释放资源
参与者	副社长
前置条件	社长进入社团解散界面
后置条件	系统更新社团信息
基本流	1. 社长进入社团解散界面。 2. 社长填写解散理由，并选择提交，等待回复。 3. 管理员审核社长的申请，并同意社长解散社团。 4. 系统更新社团信息。
备选流	2a. 社长取消操作，系统结束操作，用例结束。 3a. 管理员没通过社长申请，打回此操作，并返回到基本流步骤 1。

表 2-13 续表

属性	内容
成功场景	社团解散成功
失败场景	社团解散失败

第3章 系统功能设计

本系统包含四大模块，分别是用户管理、权限管理、社团管理和活动管理，如图 3-1 所示。



图3-1 高校社团管理功能结构图

3.1 用户管理与权限管理

用户管理可以实现注册、更新用户信息、删除用户以及查找用户等操作功能，如图 3-2 所示。

权限管理可以实现用户的权限划分，方便管理员对用户的管理。权限主要体现在菜单的操作和角色的分配。其中菜单功能可以实现添加菜单、更新菜单和删除菜单。而角色功能主要是为角色分配菜单与为用户分配角色，如图 3-2 所示。

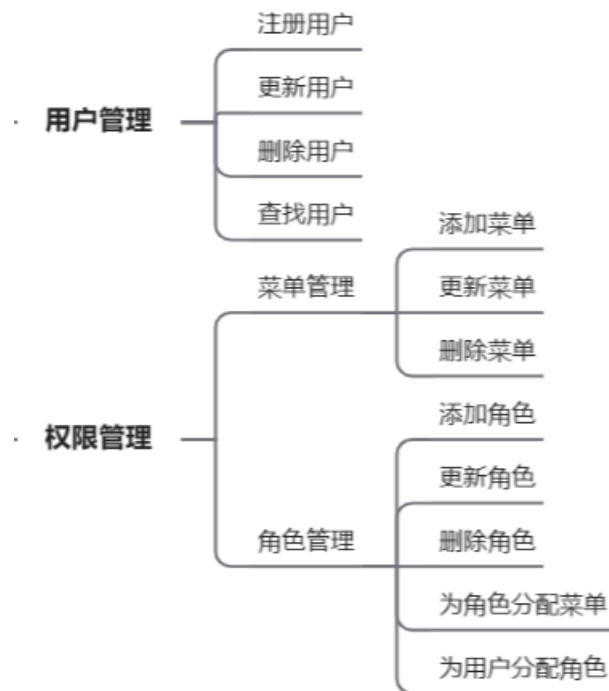


图3-2 用户与权限管理功能结构图

3.2 社团管理

社团管理的功能包括社团申请、社团信息维护、管理社团成员的加入与退出以及发布通知。学期总结功能则在学期结束时启用，如图 3-3 所示。



图3-3 活动管理功能结构图

3.3 活动管理

活动管理的功能包括活动申请、审核活动是否符合要求、发布活动、记录活动的消费记录、管理活动成员、发布活动通知、添加活动阶段、发布签到、评价参与者在活动中的表现以及总结本次活动的成就，如图 3-4 所示。

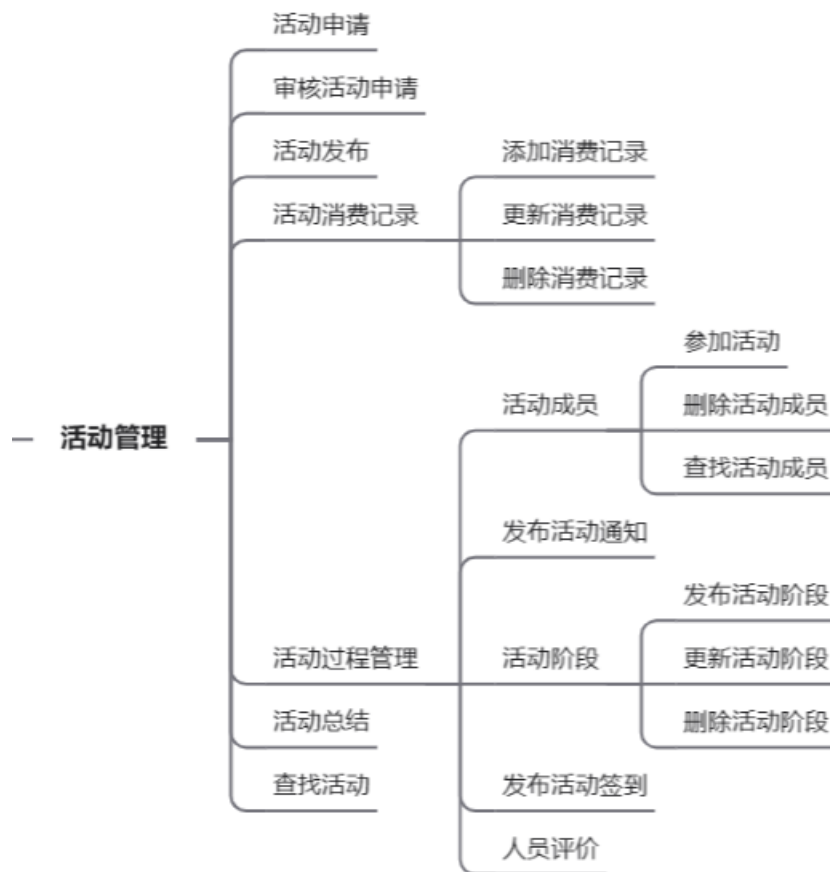


图3-4 活动管理功能结构图

第4章 类设计

4.1 类的设计

根据第二章的需求分析之后，可以在此基础上进行系统类的划分。本系统可以划分为三大类：RBAC 权限管理类、社团管理类和活动管理类。

4.1.1 RBAC 权限管理类

RBAC 权限管理类是对权限的划分与用户的管理，如图 4-1 所示。每个用户拥有的角色可以有多个，但不能没有，因为用户在系统中都会有权限。然后这些权限需要由管理员来分配，而管理员也来自用户，只是权限不同罢了。

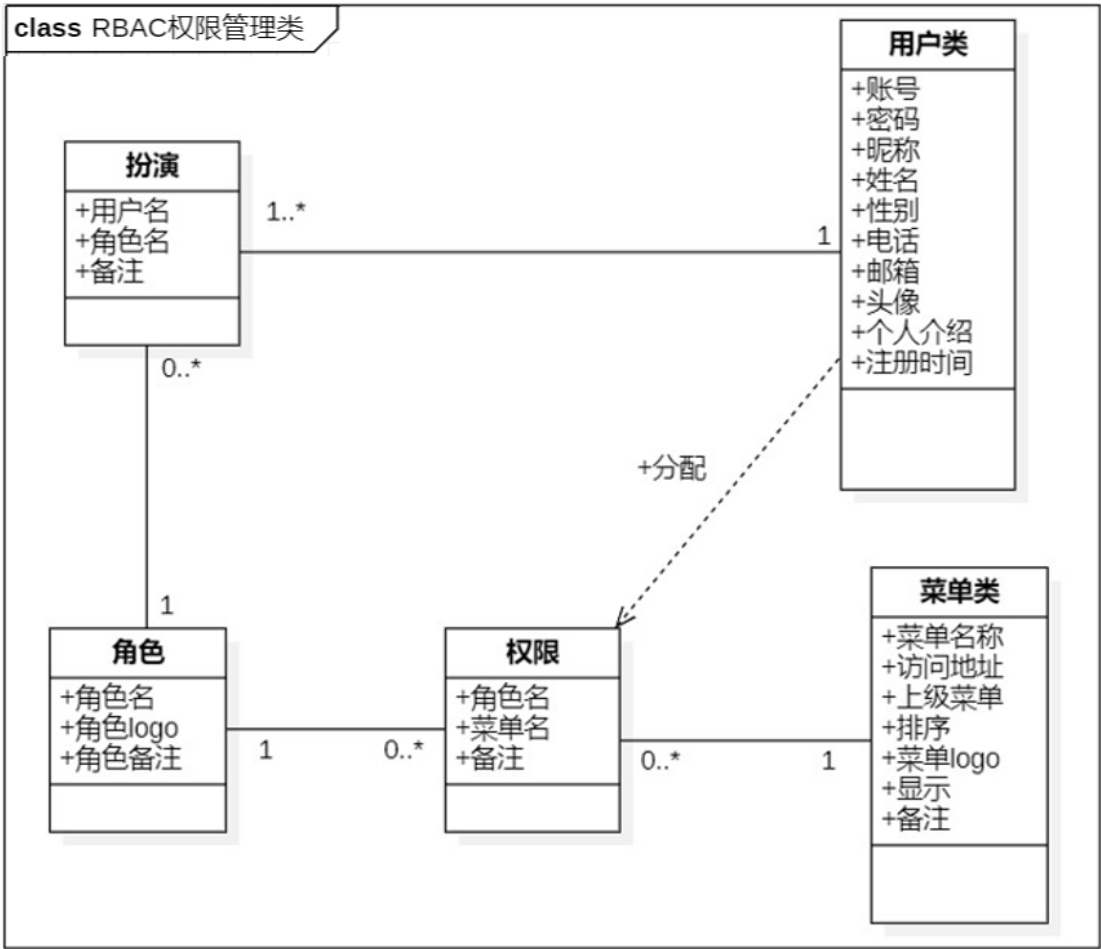


图4-1 RBAC 权限管理类

4.1.2 社团管理类

社团管理类是对社团的管理。如图4-2所示，社团的成立需要有用户去申请社团，因此社团申请与社团是组合关系；社团需要标识它是什么类型，不能没有类型，因为这会划分社团举办的活动会是哪方面的，因此社团类型与社团是组合关系；社团在成立之前，需要找一位或多位老师担任社团的指导老师，不然不能通过社团的申请，因此老师与社团也是组合关系。其他的有社团成员、社团通知、学期总结和账单这几个类与社团都是聚合关系，因为在社团的组成中，它们可以有，但也不是必须。

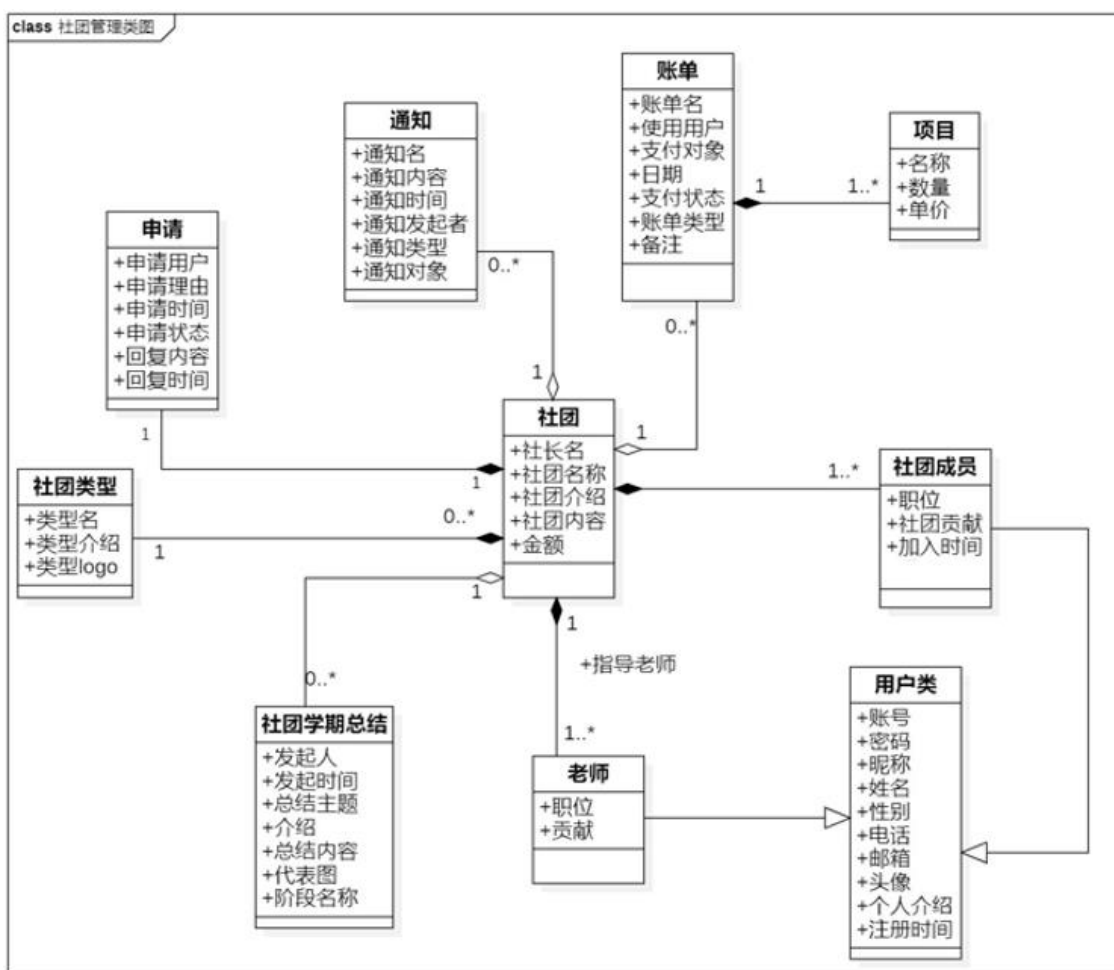


图4-2 社团管理类

4.1.3 活动管理类

活动管理类是围绕活动的方方面面实现的，如图4-3所示。其中，活动需要经历活动类型的选择、选择指导老师与活动申请等过程，因此它们和活动都是组合关

系。而其他的虽然都是活动中可能需要的，但并不是必须有的，因此是聚合管理。

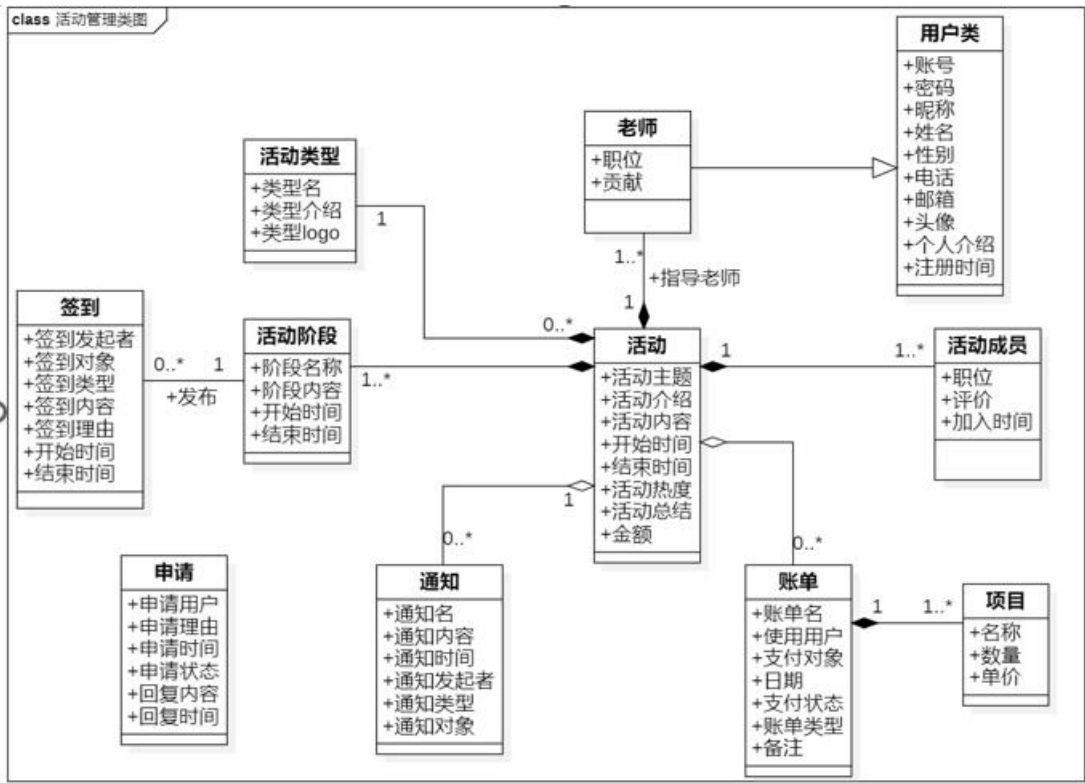


图4-3 活动管理类

4.1.4 系统类图

到此为止，类已经设计完毕，只需要把各个类组合连接就能得到高校社团管理的系统类图。因此，高校社团管理的类图设计如图 4-4 所示：

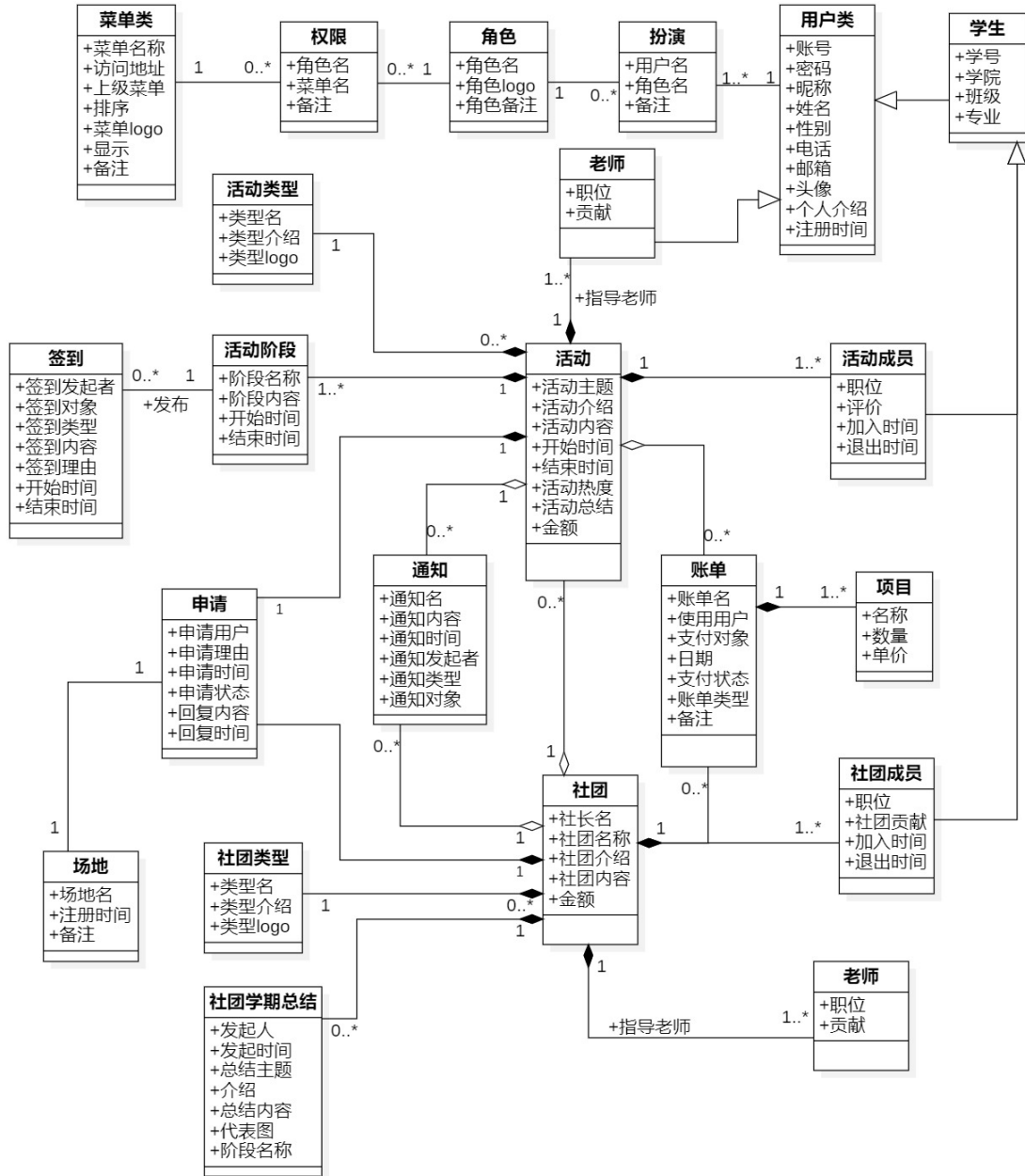


图4-4 高校社团管理系统类图

4.2 系统逻辑结构

本系统的多层逻辑架构如图 4-5 所示，其中 DAO 层进行数据持久化，高校社团管理系统具体业务的实现由 Router_Handle 层来完成，Router 层提供接口，App 层属于 Controller 层，为 View 层提供数据请求端口，View 层主要是视图界面的生成：

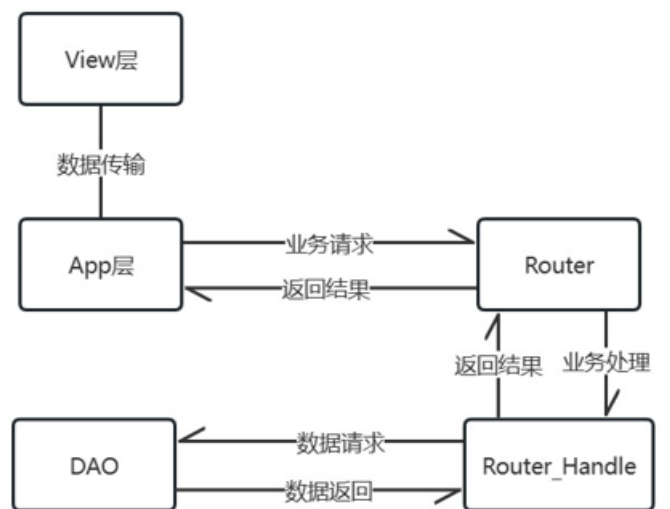


图4-5 系统结构图

第5章 数据库设计

5.1 类图转 ER 图

5.1.1 映射对象

为了避免关系表中出现重复的对象，需要确保对象和记录之间的一致性。在关系数据库中，每个对象都有一个唯一的对象标识符，用于唯一确定该对象。这个唯一标识符被称为主键。每个表都有一个主键，用于标识对象标识符。如果每个对象都有一个唯一的对象标识符，并且每个表都有一个主键，则每个对象都能被唯一地映射到表中的一行。

因此，如果某个对象没有对象标识符，则可以手动添加一个对象标识符，之后就能转换成关系数据库中的一张表，如图 5-1 所示。

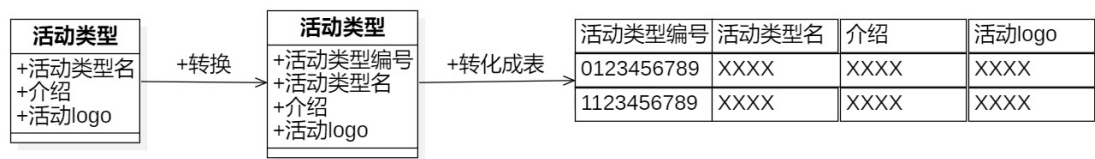


图5-1 映射对象图

5.1.2 继承关系映射

关系数据库不支持继承概念，当把带有继承关系的对象存进关系数据库时，需要考虑继承的映射解决方案。

本系统的解决方案是每个类使用一个数据表，如图 5-2 所示。

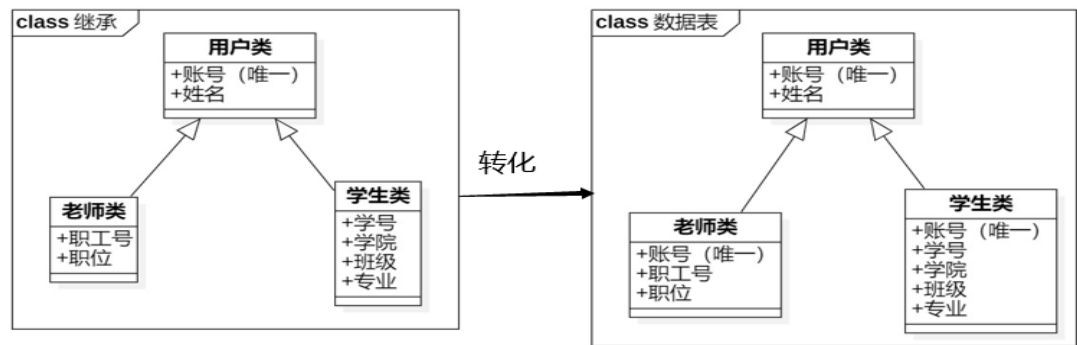


图5-2 继承映射图

5.1.3 关联和聚合映射

在关系数据库中通常使用外键实现类之间的关联。外键使一张表的一行记录与另一张表中的一行记录联系起来。映射“一对一”在关联时，需要知道关联的方向性。例如，申请与场地之间存在“一对一”的关联（在上一章的图 4-4 中有体现），申请类需要知道活动在哪儿举办，而反过来场地类不需要知道申请类的信息。因此在转换成表后，申请表中需要增加一个“场地编号”的外键，建立起与场地的关联。如果这个是一个双向关联，那场地表也就需要添加一个“申请编号”的外键，如图 5-3 所示。

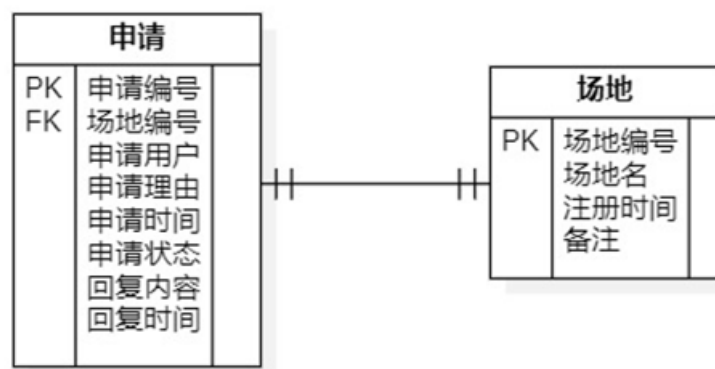


图5-3 关联映射图

如果是“一对多”的关联，如“账单”类与“项目”类之间是一对多的关系，一个账单可以同时拥有多个项目，一个项目需要分给一个账单完成。这时只要把账单表的主键作为外键放入项目表中，因为关系是在“多”的一方维护，如图 5-4 所示。

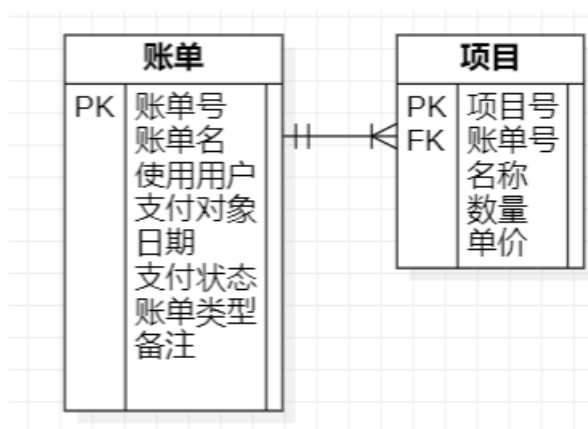


图5-4 聚合映射图

5.2 ER 图

根据上一节的规则可将上一章的图 4-4 转为如图 5-5 所示的 ER 图。

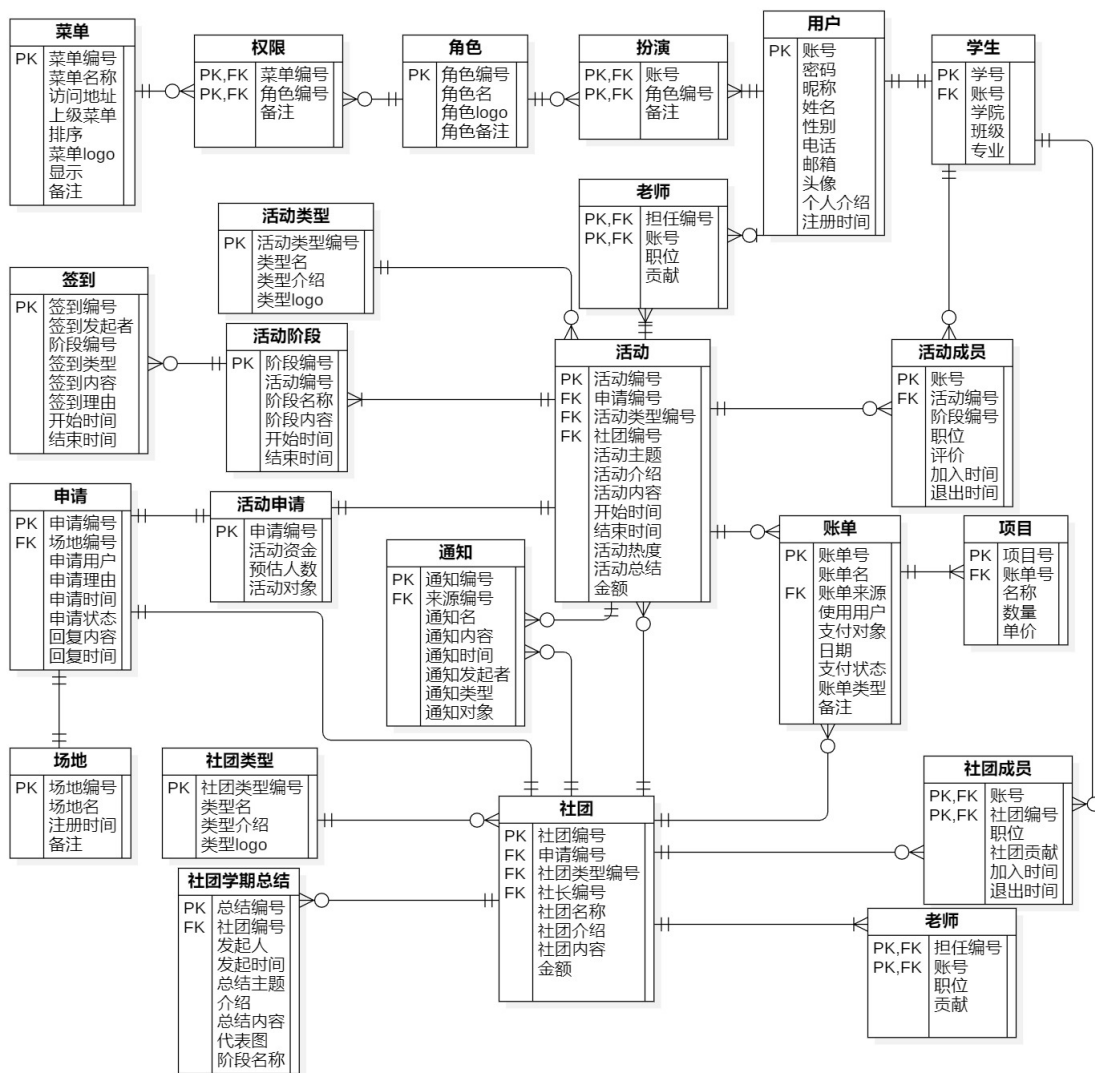


图5-5 ER 图

5.3 格式设计

格式设计是进行物理设计的前一步骤，它定义数据库字段的数据格式与大小。每一个数据库的表都必须按照格式设计的要求添加表格，格式大体内容如表 5-1 所示。

表5-1 格式设计表

名称	格式	含义	数据类型
编号	202305121200001234	18 位	Char
密码	Bcryptjs 加密	60 位	Char
时间	2023-05-12 12:00:00	不固定	datetime
电话	000 1234 5678	11 位	Char
邮箱	123@qq.com	最多 100 位	Varchar
主题	集美大学毕业论文	最多 20 个字	Varchar
介绍	爱好游泳	最多 130 个字	Varchar
内容	爱好游泳	不固定	Text
名称	计算机工程学院	最多 20 个字	Varchar
姓名	张三	最多 10 个字	Varchar
图片	/images/user/123.jpg	最多 200 位	Varchar
地址	/images/user	最多 100 位	Varchar
备注	计算机语言	最多 50 个字	Varchar
无符号整数	16777215	最大 16777215	mediumint unsigned
小数	9,999,999.99	9	Decimal
状态	1	1	Tinyint

5.4 表结构设计

(1) user 表

用户表 user 记录系统包括管理员在内的所有用户信息，其中包含用户账号的 user_id（作为主键）、账号密码、昵称、姓名、性别、电话、邮箱、头像、个人介绍，以及用户注册账号的时间。其中用户的账号密码不会直接存入数据库，需要通过加密之后，才能保存在数据库中。因此，用户表的结构如表 5-2 所示。

表5-2 用户表 user

名称	类型	主键	允许 null 值	备注
user_id	char(18)	是	否	账号
password	char(60)	否	否	密码
nickname	varchar(20)	否	是	昵称
user_name	varchar(10)	否	是	用户姓名
sex	enum	否	是	性别
phone	char(11)	否	是	电话
email	varchar(100)	否	是	邮箱
picture	varchar(200)	否	是	头像
intro	varchar(130)	否	是	个人介绍
regist_time	datetime	否	是	注册

(2) student 表

学生表 student 继承 user 表，因此它只是 user 表的扩展，多出来的字段有学号（作为主键）、账号（user 表的主键，当前表的外键）、学院、班级，以及专业。因此，学生表的结构如表 5-3 所示。

表5-3 学生表 student

名称	类型	主键	允许 null 值	备注
stu_id	char(18)	是	否	学号
user_id	char(18)	否	否	账号
college	varchar(20)	否	是	学院
classes	varchar(20)	否	是	班级
major	varchar(20)	否	是	专业

(3) role 表

角色表 role 记录系统所有的角色信息，其中包括角色编号、角色名、角色 logo 以及对角色的描述。因此，角色表的结构如表 5-4 所示。

表5-4 角色表 role

名称	类型	主键	允许 null 值	备注
role_id	char(18)	是	否	角色编号
role_name	varchar(10)	否	是	角色名
role_logo	varchar(200)	否	是	角色 logo
remark	varchar(50)	否	是	备注

(4) play_role 表

扮演表 play_role 记录用户在系统中所拥有的角色，它包含账号（user 表的主键，当前表的外键）、角色编号（role 表的主键，当前表的外键）以及备注。其中账号与角色编号共同组成当前表的主键。因此，扮演表的结构如表 5-5 所示。

表5-5 扮演表 play_role

名称	类型	主键	允许 null 值	备注
user_id	char(18)	是	否	账号
role_id	char(18)	是	否	角色编号
remark	varchar(50)	否	是	备注

(5) menu 表

菜单表 menu 记录系统所有的菜单，它包含的字段有菜单编号（当前表的主键）、菜单名称、访问地址、上级菜单、排序（用于排列菜单的先后顺序）、菜单 logo、是否显示本菜单以及对菜单的描述。因此，菜单表的结构如表 5-6 所示。

表5-6 菜单表 menu

名称	类型	主键	允许 null 值	备注
menu_id	char(18)	是	否	菜单编号
name	varchar(10)	否	是	菜单名称
href	varchar(100)	否	是	访问地址
parent_id	char(18)	否	是	上级菜单
sort	tinyint	否	否	排序

表 5-6 续表

名称	类型	主键	允许 null 值	备注
menu_logo	varchar(200)	否	是	菜单 logo
is_show	tinyint	否	否	是否显示
remarks	varchar(50)	否	是	备注

(6) menu_role 表

权限表 menu_role 记录系统所有角色对应的菜单,他维持角色能有那些权限。它拥有的字段有菜单编号(menu 表的主键,当前表的外键)、角色编号(role 表的主键,当前表的外键)以及备注。其中,菜单编号与角色编号共同组成当前表的主键。因此,权限表的结构如表 5-7 所示。

表5-7 权限表 menu_role

名称	类型	主键	允许 null 值	备注
menu_id	char(18)	是	否	菜单编号
role_id	char(18)	是	否	角色编号
remark	varchar(50)	否	是	备注

(7) area 表

场地表 area 是记录学校有哪些场地可以作为活动开展的地方,它包含的字段有场地编号(当前表的主键),场地名、注册时间以及备注。因此,场地表的结构表 5-8 所示。

表5-8 场地表 area

名称	类型	主键	允许 null 值	备注
area_id	char(18)	是	否	场地编号
area_name	varchar(20)	否	是	场地名
regist_time	datetime	否	是	注册
remark	varchar(50)	否	是	备注

(8) club_type 表

社团类型表 club_type 是记录所有社团的标签，如文化类社团这样的标签。它包含的字段有社团类型编号（当前表的主键）、类型名以及类型介绍。因此，社团类型表的结构如表 5-9 所示。

表5-9 社团类型表 club_type

名称	类型	主键	允许 null 值	备注
type_id	char(18)	是	否	社团类型编号
type_name	varchar(20)	否	是	类型名
type_content	varchar(130)	否	是	类型介绍
picture	varchar(200)	否	是	类型 logo

(9) apply 表

申请表 apply 是社团申请或活动申请中的必填信息，它记录申请是否通过以及为什么不通过等信息。它包含的字段有申请编号（当前表的主键）、场地编号（area 表的主键，当前表的外键）、申请用户（user 表的主键，当前表的外键）、申请理由、申请时间、申请状态（状态有：审核中、通过、未通过）、回复以及回复时间。因此，申请表的结构如表 5-10 所示。

表5-10 申请表 apply

名称	类型	主键	允许 null 值	备注
apply_id	char(18)	是	否	申请编号
area_id	char(18)	否	否	场地编号
apply_user	char(18)	否	否	申请用户
apply_content	varchar(130)	否	是	申请理由
apply_time	datetime	否	是	申请时间
apply_state	tinyint	否	否	申请状态
reply	varchar(130)	否	是	回复
reply_time	datetime	否	是	回复时间

(10) club 表

社团表 club 是记录系统中所有社团的基本信息，它包含的字段有社团编号（当

前表的主键)、申请编号(apply 表的主键,当前表的外键)、社团类型编号(club_type 表的主键,当前表的外键)、社长编号(user 表的主键,当前表的外键)、社团名称、社团介绍、社团内容以及金额。因此,社团表的结构如表 5-11 所示。

表5-11 社团表 club

名称	类型	主键	允许 null 值	备注
club_id	char(18)	是	否	社团编号
apply_id	char(18)	否	否	申请编号
type_id	char(18)	否	否	社团类型编号
user_id	char(18)	否	否	社长编号
club_name	varchar(20)	否	是	社团名称
club_intro	varchar(130)	否	是	社团介绍
club_content	text	否	是	社团内容
money	decimal(9,2)	否	否	金额

(11) club_report 表

社团学期总结表 club_report 是记录社团每个学期做了什么事,它包含总结编号(当前表的主键)、社团编号(club 表的主键,当前表的外键)、发起人(user 表的主键、当前表的外键)、总结主题、介绍、总结内容、代表图以及阶段名称。因此,社团学期表的结构如表 5-12 所示。

表5-12 社团学期总结表 club_report

名称	类型	主键	允许 null 值	备注
report_id	char(18)	是	否	总结编号
club_id	char(18)	否	否	社团编号
user_id	char(18)	否	否	发起人
report_title	varchar(20)	否	是	总结主题
report_intro	varchar(130)	否	是	介绍
report_content	text	否	是	总结内容

表 5-12 续表

名称	类型	主键	允许 null 值	备注
picture	varchar(200)	否	是	代表图
stage_name	varchar(20)	否	是	阶段名称

(12) club_member 表

社团成员表 club_member 是记录社团有哪些成员信息，它包含账号（user 表的主键，当前表的外键）、社团编号（club 表的主键，当前表的外键）、职位、社团贡献、加入时间以及退出时间。其中账号与社团编号共同组成当前表的主键。因此，社团成员表的结构如表 5-13 所示。

表5-13 社团成员表 club_member

名称	类型	主键	允许 null 值	备注
user_id	char(18)	是	否	账号
club_id	char(18)	是	否	社团编号
position	varchar(20)	否	是	职位
contribute	text	否	是	社团贡献
join_time	datetime	否	是	加入时间

(13) teacher 表

老师表 teacher 是记录社团或活动中有哪些指导老师的，它包含的字段有担任编号（club 表或 activity 表的主键，当前表的外键）、账号（user 表的主键，当前表的外键）、职位以及贡献。老师表的结构如表 5-14 所示。

表5-14 老师表 teacher

名称	类型	主键	允许 null 值	备注
bear_id	char(18)	否	否	担任编号
user_id	char(18)	否	否	账号
position	varchar(20)	否	是	职位
contribute	text	否	是	贡献

(14) activity_type 表

活动类型表 activity_type 是记录活动有哪些类型的,如文化类活动。它包含的字段活动类型编号(当前表的主键)、类型名、类型介绍以及类型 logo。因此,活动类型表的结构如表 5-15 所示。

表5-15 活动类型表 activity_type

名称	类型	主键	允许 null 值	备注
type_id	char(18)	是	否	活动类型编号
type_name	varchar(20)	否	是	类型名
type_content	varchar(130)	否	是	类型介绍
picture	varchar(200)	否	是	类型 logo

(15) activity 表

活动表 activity 是记录系统中所有活动的基本信息,它包含的字段有活动编号(当前表的主键)、申请编号(activity_apply 表的主键,当前表的外键)、活动类型编号(activity_type 表的主键,当前表的外键)、社团编号(club 表的主键,当前表的外键)、活动主题、活动介绍、活动内容、开始时间、结束时间、活动热度、活动总结以及活动资金。因此,活动表的结构如表 5-16 所示。

表5-16 活动表 activity

名称	类型	主键	允许 null 值	备注
activity_id	char(18)	是	否	活动编号
apply_id	char(18)	否	否	申请编号
type_id	char(18)	否	否	活动类型编号
club_id	char(18)	否	否	社团编号
activity_title	varchar(20)	否	是	活动主题
activity_intro	varchar(130)	否	是	活动介绍
activity_content	text	否	是	活动内容
start_time	datetime	否	是	开始时间
end_time	datetime	否	是	结束时间

表 5-16 续表

名称	类型	主键	允许 null 值	备注
activity_heat	mediumint	否	否	活动热度
activity_report	text	否	是	活动总结
money	decimal(9,2)	否	否	活动资金

(16) activity_stage 表

活动阶段表 activity_stage 是记录活动有哪些阶段的，它包含的字段有阶段编号（当前表的主键）、活动编号（activity 表的主键，当前表的外键）、阶段名称、阶段内容、开始时间以及结束时间。因此，活动阶段表的结构如表 5-17 所示。

表5-17 活动阶段表 activity_stage

名称	类型	主键	允许 null 值	备注
stage_id	char(18)	是	否	阶段编号
activity_id	char(18)	否	否	活动编号
stage_name	varchar(20)	否	是	阶段名称
stage_content	text	否	是	阶段内容
start_time	datetime	否	是	开始时间
end_time	datetime	否	是	结束时间

(17) activity_member 表

活动成员表 activity_member 是记录活动中有哪些用户参与此项活动的，它包含的字段有账号（user 表的主键，当前表的外键）、活动编号（activity 表的主键，当前表的外键）、阶段编号（activity_stage 表的主键，当前表的外键）、职位、评价、加入时间以及退出时间。其中，账号、活动编号和阶段编号共同组成当前表的主键。因此，活动成员表的结构如表 5-18 所示。

表5-18 活动成员表 activity_member

名称	类型	主键	允许 null 值	备注
user_id	char(18)	是	否	账号
activity_id	char(18)	否	否	活动编号
stage_id	char(18)	否	否	阶段编号
position	varchar(20)	否	是	职位
appraise	text	否	是	评价
join_time	datetime	否	是	加入时间
exit_time	datetime	否	是	退出时间

(18) bill 表

账单表 bill 是记录社团或活动的消费记录，它包含的字段有账单号（当前表的主键）、账单名、账单来源（club 表或 activity 表的主键，当前表的外键）、使用用户（user 表的主键，当前表的外键）、支付对象（club 表或 activity 表的主键，当前表的外键）、日期、支付状态（状态：未支付、支付成功、支付失败）、账单类型（类型：支出和收入）以及备注。因此，账单表的结构如表 5-19 所示。

表5-19 账单表 bill

名称	类型	主键	允许 null 值	备注
bill_id	char(18)	是	否	账单号
bill_name	varchar(20)	否	是	账单名
source_id	char(18)	否	是	账单来源
user_id	char(18)	否	是	使用用户
pay_object	char(18)	否	是	支付对象
date	datetime	否	是	日期
pay_state	tinyint	否	否	支付状态
bill_type	enum	否	是	账单类型
remark	varchar(50)	否	是	备注信息

(19) project 表

项目表 project 是账单的项，记录每一种消费项目，它包含的字段有项目号（当前表的主键）、账单号（bill 表的主键，当前表的外键），名称、数量以及单价。因此，项目表的结构如表 5-20 所示。

表5-20 项目表 project

名称	类型	主键	允许 null 值	备注
project_id	char(18)	是	否	项目号
bill_id	char(18)	否	否	账单号
project_name	varchar(20)	否	是	名称
project_num	mediumint	否	否	数量
price	decimal(9,2)	否	否	单价

(20) notice 表

通知表 notice 是记录用户想用系统通知哪些用户的，它包含的字段有通知编号（当前表的主键），来源编号（club 表或 activity 表的主键，当前表的外键）、通知名、通知内容、通知时间、通知发起者（user 表的主键，当前表的外键）、通知类型（类型：活动通知和社团通知）以及通知对象。因此，通知表的结构如表 5-21 所示。

表5-21 通知表 notice

名称	类型	主键	允许 null 值	备注
notice_id	char(18)	是	否	通知编号
source_id	char(18)	否	是	来源编号
notice_name	varchar(20)	否	是	通知名
notice_content	varchar(130)	否	是	通知内容
notice_time	datetime	否	是	通知时间
notice_type	enum	否	是	通知类型
notice_object	char(18)	否	是	通知对象

(21) sign 表

签到表 sign 是记录活动现场有哪些人到了，它包含的字段有签到编号（当前

表的主键)、签到发起者 (user 表的主键, 当前表的外键)、阶段编号 (activity_stage 的主键, 当前表的外键)、签到类型 (如数字签到)、签到内容、签到理由、开始时间以及结束时间。因此, 签到表的结构如表 5-22 所示。

表5-22 签到表 sign

名称	类型	主键	允许 null 值	备注
sign_id	char(18)	是	否	签到编号
user_id	char(18)	否	否	签到发起者
stage_id	char(18)	否	是	阶段编号
sign_type	enum	否	是	签到类型
sign_content	varchar(130)	否	是	签到内容
sign_reason	varchar(130)	否	否	签到理由
start_time	datetime	否	是	开始时间
end_time	datetime	否	是	结束时间