

山烛-教育视频网站

《系统设计说明书》

Growing light 团队

项目组成员： 苏杰阳 齐易捷
杨朕炫 张富源
林泽坤 潘增滢
杨思雨 黄伟源
陈起 刘晓君

日期：2021 年 4 月 24 日

目录

| | |
|---|----|
| 一 . 引言..... | 5 |
| 1.1 编写目的..... | 5 |
| 1.2 背景..... | 5 |
| 1.3 读者对象..... | 5 |
| 1.4 术语定义及说明..... | 5 |
| 二 . 体系结构设计..... | 7 |
| 2.1 项目概述..... | 7 |
| 2.2 项目环境..... | 7 |
| 2.2.1 系统开发环境..... | 7 |
| 2.2.2 系统部署环境..... | 8 |
| 2.3 设计约束..... | 8 |
| 2.4 设计策略..... | 9 |
| 2.5 系统架构..... | 10 |
| 2.5.1 体系结构..... | 10 |
| 2.5.2 包图..... | 12 |
| 2.5.3 模型视图..... | 13 |
| 三. 系统功能模块设计..... | 14 |
| 3.1 系统流程图..... | 14 |
| 3.2 系统功能模块层次设计..... | 15 |
| 3.3 系统功能模块详细设计..... | 15 |
| 3.3.1 平台功能模块..... | 15 |
| 3.3.2 后台功能模块..... | 26 |
| 四. 类图展示与算法说明..... | 32 |
| 4.1 类图展示..... | 32 |
| 4.1.1 总览..... | 32 |
| 4.1.2 CourseVideo 视频课程类..... | 32 |
| 4.1.3 Comment、LeafComment、CompositeComment 评论类..... | 34 |
| 4.1.4 CourseWare 课件类..... | 35 |
| 4.1.5 Post 帖子类..... | 35 |
| 4.1.6 VideoPostRecord 帖子视频关联类..... | 36 |

| | |
|---------------------------------------|----|
| 4.1.7 User 用户类..... | 36 |
| 4.1.8 Permission 权限类和 Role 角色类..... | 37 |
| 4.1.9 CollectionRecord 用户收藏记录类..... | 38 |
| 4.1.10 VoluntaryActivity 捐赠志愿活动类..... | 38 |
| 4.1.11 ActivityRecord 活动记录类..... | 39 |
| 4.2 推荐算法..... | 39 |
| 4.2.1 召回算法..... | 41 |
| 4.2.2 排序算法..... | 43 |
| 五. 接口设计..... | 44 |
| 5.1 接口统一说明..... | 44 |
| 5.1.1 统一返回格式..... | 44 |
| 5.1.2 状态码封装说明..... | 44 |
| 5.1.3 动态路由参数说明..... | 44 |
| 5.2 模块接口说明..... | 44 |
| 5.2.1 认证模块..... | 44 |
| 5.2.2 用户模块..... | 45 |
| 5.2.3 通用模块..... | 48 |
| 5.2.4 讨论模块..... | 49 |
| 5.2.5 讨论详情模块..... | 50 |
| 5.2.6 视频首页模块..... | 50 |
| 5.2.7 视频详情模块..... | 52 |
| 5.2.8 捐赠模块..... | 53 |
| 5.2.9 日志管理模块..... | 54 |
| 5.2.10 视频管理模块..... | 54 |
| 5.2.11 用户管理模块..... | 55 |
| 5.2.12 捐赠活动管理模块..... | 56 |
| 5.2.13 讨论管理模块..... | 57 |

修订记录:

| 版本号 | 作者 | 修改日期 | 备注 |
|------|---------------------|------|--------------------|
| V1.0 | 张富源 | 4.20 | 项目概述、设计约束、后台功能模块添加 |
| V1.1 | 苏杰阳 | 4.20 | 引言部分、项目环境添加 |
| V1.2 | 苏杰阳 | 4.21 | 接口部分添加 |
| V1.3 | 杨思雨 | 4.21 | 设计策略系统架构添加 |
| V1.4 | 齐易捷 | 4.21 | 类图添加 |
| V2.0 | 潘增滢 | 4.22 | 初次整合 |
| V2.1 | 杨思雨 | 4.22 | 平台功能模块添加 |
| V2.2 | 张富源 | 4.22 | 后台功能模块修订 |
| V2.3 | 苏杰阳 | 4.22 | 接口部分用户模块修订 |
| V2.4 | 杨思雨 | 4.22 | 系统功能模块层次图添加 |
| V3.0 | 苏杰阳、杨思雨、张富源、齐易捷、潘增滢 | 4.22 | 初审，修改发现的问题 |
| V3.1 | 杨思雨 | 4.23 | 目录修订 |
| V3.2 | 潘增滢 | 4.23 | 系统功能模块层次图修订 |
| V3.3 | 齐易捷 | 4.24 | 类图部分修改 |
| V3.4 | 潘增滢 | 4.24 | 文档排版字体等细节调整 |
| V4.0 | 苏杰阳、杨思雨、张富源、齐易捷、潘增滢 | 4.24 | 最终复审 |

一. 引言

1.1 编写目的

编写本文档的目的主要是详细说明山烛-教育视频网站系统的结构和内部设计，提供详细的系统结构设计，功能模型和后端接口设计的描述，并说明该项目所使用的技术以及软件，并指导项目的开发和部署，同时作为重要的技术参考资料，为未来的系统维护和功能扩展做参考。

1.2 背景

1. 软件系统名称：山烛-教育视频网站
2. 系统类型：web 端
3. 系统从属：无从属关系, 会与第三方进行接口对接等去实现捐赠区的功能实现。
4. 开发项目组名称：Growing light
5. 软件使用硬件背景：能够安装并使用支持 HTML5 的浏览器的硬件设施。

1.3 读者对象

1. 软件原型设计师：以该设计书为例，进行符合项目的原型设计，并不断加以完善。
2. 产品经理：以该份设计书为基准进行一系列的分工，管理，并不断完善该份设计书。
3. 开发人员：以该份设计书中所涉及的功能描述和接口文档进行实际上的开发，并进行矫正修改。
4. 测试人员：以该份设计书所设计的功能描述进行一系列的测试，以验收标准来作为测试的标准。
5. 用户：能够客观详细的了解该项目的所有功能。
6. 项目投资方：能够客观详细的了解整个项目的系统结构。

1.4 术语定义及说明

1. Webapp: web application, 基于 WEB 端的应用网页或网站。
2. PC: personal computer, 家用电子计算机。

3. JDK: JavaSEDevelopmentkit, java 开发工具包.
4. IDE: Integrated Development Environment, 集成开发环境。
5. MVVM: Model-View-ViewModel。框架开发逻辑。

二. 体系结构设计

2.1 项目概述

山烛视频项目是一个由“Growing Light”团队开发的主打线上扶贫教育的开放式平台。

尽管当前我国已基本实现脱贫,但仍有许多孩子因为各种因素而无法享受到良好的教育资源。而大部分大学生群体对于支教等志愿服务积极性高,却仍缺乏公开、安全的平台或环境。市场上的扶贫相关软件影响度较小,且功能较为单一,教育类软件针对性不强,商业化程度过高。“山烛”的提出就是为了解决这一社会痛点。它的目标用户为教育资源匮乏地区的义务阶段教育学生群体、青年志愿者(主要为大学生群体)以及社会上的爱心人士。它能为贫困地区孩子提供更多的教育机会,也能帮助想参与支教等志愿服务但受制于客观因素的热心人士贡献自己的力量。

功能描述:本网站主要分为四大功能区域 网站主页、分享讨论、捐赠中心、个人中心。

1. 网站主页:搜索功能、个人信息展示(抽屉)、公益广告、视频推荐、视频分类、播放视频
2. 分享讨论:搜索功能、查看帖子、讨论分类、发布帖子
3. 捐赠中心:活动列表、搜索功能、查看活动详情
4. 个人中心:我的视频、我的帖子、我的收藏、个人信息查看修改、发布视频、志愿者认证
5. 后台功能:日志功能、用户管理、视频管理、捐赠管理、讨论管理

2.2 项目环境

2.2.1 系统开发环境

1. IDE: java 后端开发编辑器
2. Maven: 后端项目依赖管理 00
3. Navicat: 数据库可视化工具
4. Mybatis: 后端数据库交互框架

5. Springboot: 后端 web 框架
6. Swagger: 后端接口可视化工具
7. Pycharm: python 后端开发编辑器
8. Flask: 基于 WSGI 的轻量级 web 应用框架。
9. Node.js: 前端基本环境, 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。
10. Npm : 前端项目依赖管理, 与 Node.js 捆绑使用。
11. Vue cli: 基于 Vue.js 的脚手架, 用于自动生成 vue.js+webpack 的项目模板。
12. VS Code: 全称为 Visual Studio Code, 前端开发编辑器。
13. Git: 项目版本管理。
14. Mysql: 数据库管理。
15. Chrome(filefox 等): 前端调试工具。

2.2.2 系统部署环境

2.2.2.1 java 后台部署环境

以阿里云官网购买的 centos 服务器作为后端运行的服务器, 使用 tomcat 来进行运行项目, 让前端能够访问到接口, 使用功能。

2.2.2.2 python 后台部署环境

以阿里云官网购买的 centos 服务器作为后端运行的服务器, 使用 gunicorn 来运行项目, 并使用 nginx 来进行端口映射实现正常的访问。

2.2.2.3 Vue 前端部署环境

以腾讯云官网购买的轻便应用服务器作为前端部署的服务器, 使用 nginx 进行相应的端口映射来使项目能在网络上成功访问。并通过域名的方式来访问, 如(我所使用的 www.colder.ltd)。

2.3 设计约束

1、需求约束：

(a) 本产品应遵循 web 开发的合法化的标准及网络绿色交流的合法化约束。

(b) 该产品约束的要求为功能正常，运行流畅、不卡顿、不闪退，适配于大部分主流浏览器。

(c) 软件质量的使用范围为适配于大部分主流浏览器，不会出现出现恶意漏洞导致用户数据泄露，保护用户的隐私安全。

2、隐含约束：

(a) 浏览器平台

(b) 数据库系统

(c) 编程工具

(d) 支持软件

2.4 设计策略

(1) 关键需求决定架构。关键需求决定架构有两个方面的涵义：一方面，功能需求与非功能需求数量众多，应该控制架构设计时需要详细分析的用例个数；另一方面，不同非功能需求之间往往具有相互制约性，应该权衡非功能需求之间的关系，找到影响架构的重点非功能需求。关键需求决定架构的策略有利于集中精力深入分析最为重要的需求。当架构工程师把全部精力扑在相对较少的关键需求上时，可以更为深入的分析这些需求，有利于得到透彻的认识，从而设计出合理的架构。

(2) 全面认识需求。从不同级别、不同类别梳理列表归纳总结，建立跟踪矩阵与影响分析表。一方面，将所有需求从不同的级别（组织级、用户级、开发级）分层梳理列表归纳总结，建立跟踪矩阵，既避免因遗漏需求而造成软件系统达不到要求，也避免开发人员一厢情愿也为用户制造没有实际意义的无用功能，同时，有助于软件架构设计人员对软件系统要求有全面的认识，清晰理清需求之间、需求与软件系统目的目标与核心业务、商业理由间的因果关系。另一方面，将所有需求划分为不同的类型（功能需求或用例、质量属性、约束与限制）进行梳理列表归纳总结，建立影响分析表，找出不同需求类型之间的相互支持、相互制

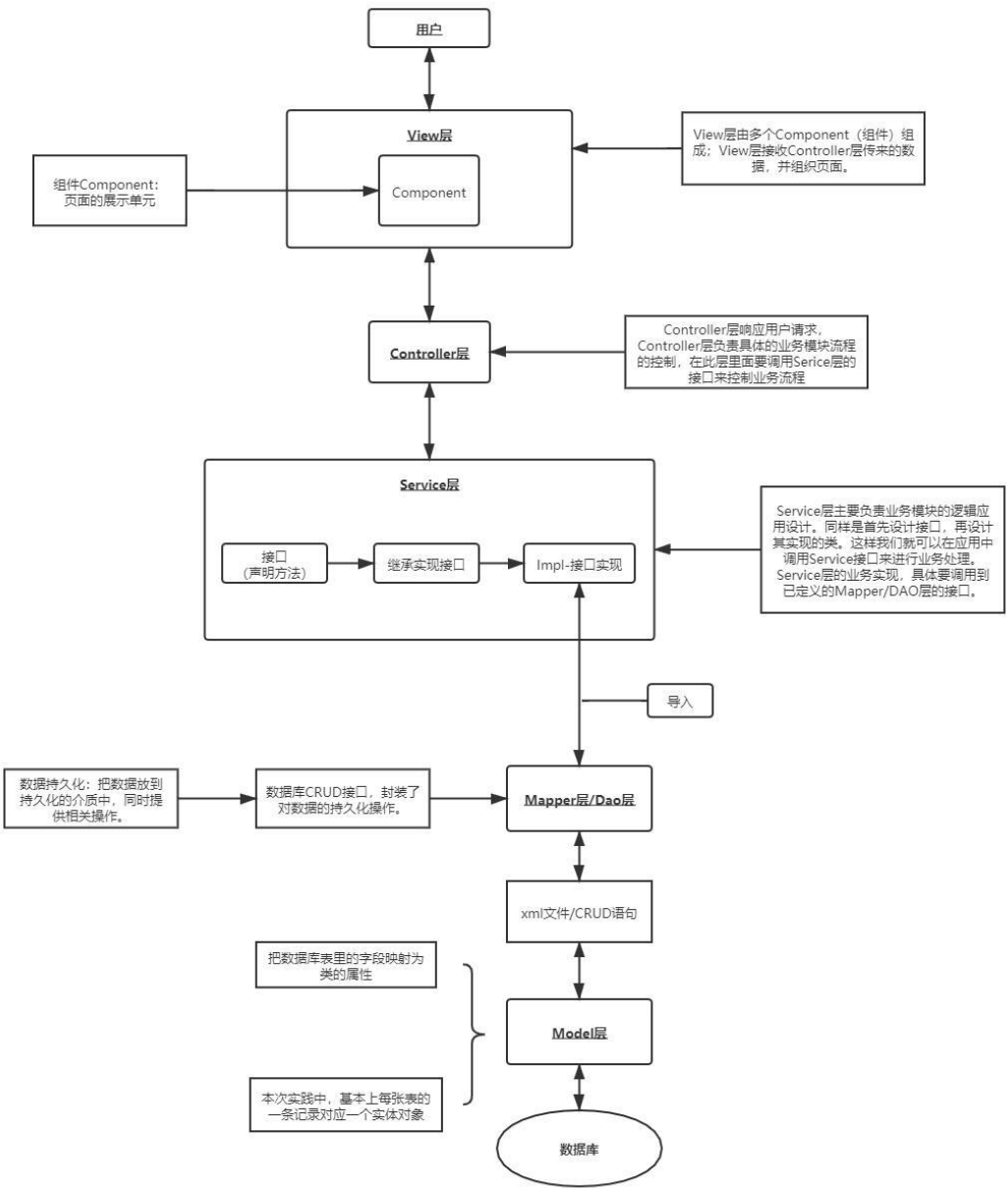
约关系的影响，并在深思熟虑之后做出合适的需求权衡和取舍。

(3) 多视图探寻架构。一次只从某一立场、某一视角出发围绕少数概念和技术展开，并分析对其它部分、其它立场视角分析结果的关系与影响。

(4) 尽早验证架构。架构设计是现代软件开发中最为关键的一环，架构设计是否合理将直接影响到软件系统最终是否成功。架构验证必须精挑细选能够触发主要设计决策参与执行的、或有较高技术风险的、或最影响用户满意度的一些功能，来尽早验证架构方案并做出调整，避免后期出现大规模返工。

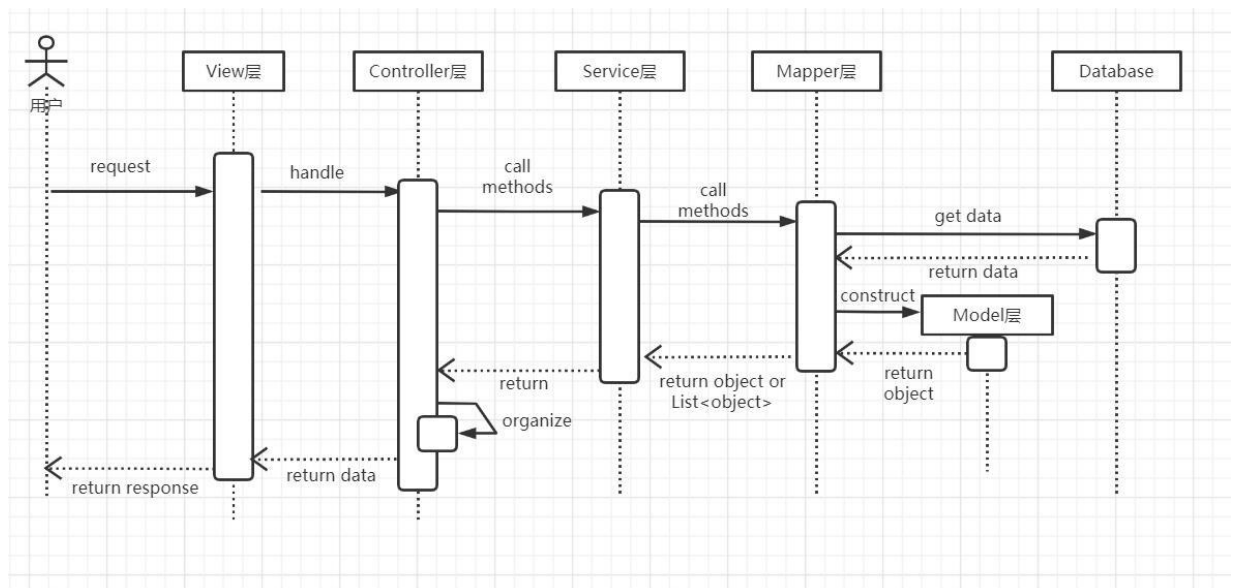
2.5 系统架构

2.5.1 体系结构



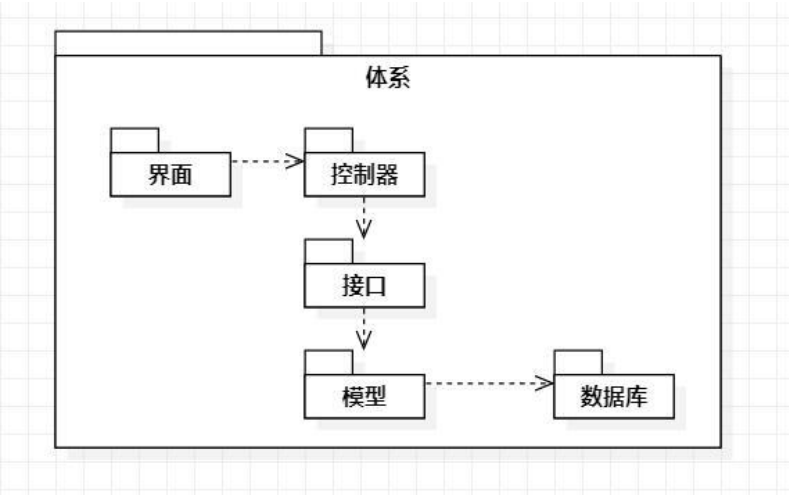
本次实践系统的体系结构如上图所示，用户首先与 View 层进行交互，View 层由 Vue 框架实现，其中一个页面包含多个 Component，即 Component 组件为页面组织的最小单元；用户点击页面之后，系统发送请求到 Controller 层，该层主要负责具体业务模块流程的控制，此层要调用到 Service 层的接口去完成业务，针对不同的业务流程有不同的控制器；Service 层主要负责某个具体业务的逻辑设计，我们调用 service 接口进行业务处理，service 调用到已经定义的 Mapper 层的接口，该层的封装有利于提高通用业务逻辑的独立性和重复利用性；Mapper 层负责数据的持续化存储，mapper 层直接与数据库打交道（执行 SQL 语句），将数据库中的记录组装成 Model 对象，并将对象返回给 Service 层；Model 层即为系统中的实体类，其属性和数据库中的字段基本保持一致。

各层的交互如下图所示：

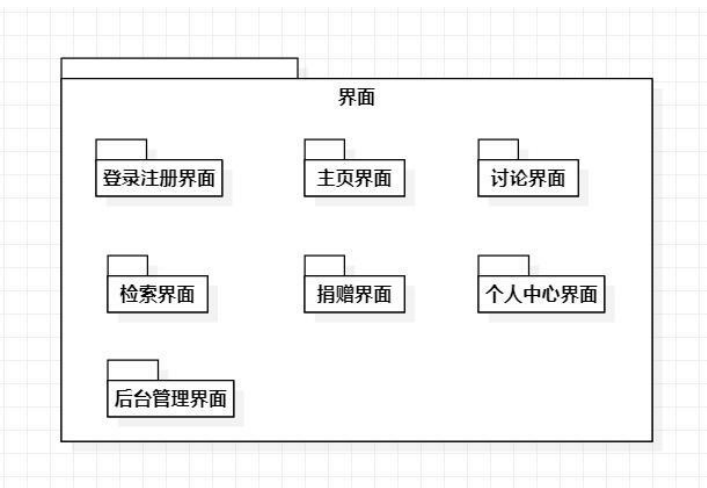


2.5.2 包图

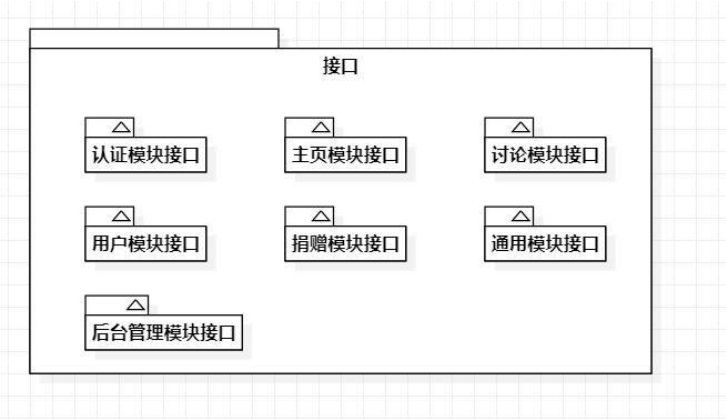
系统包图



界面包图

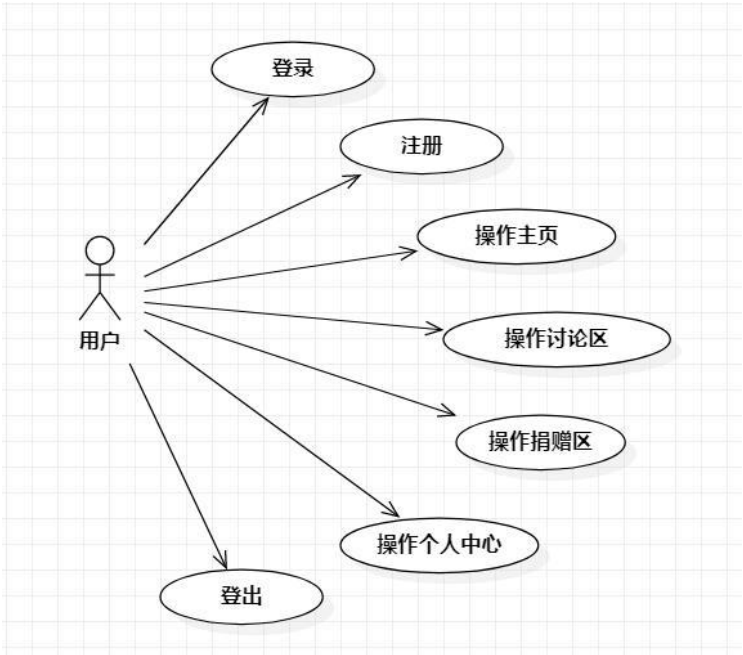


接口包图

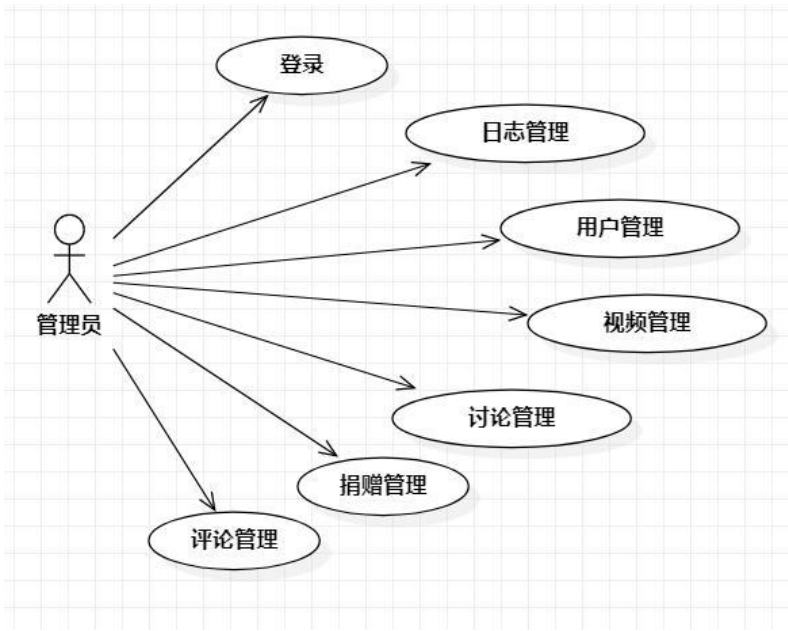


2.5.3 模型视图

用户用例图

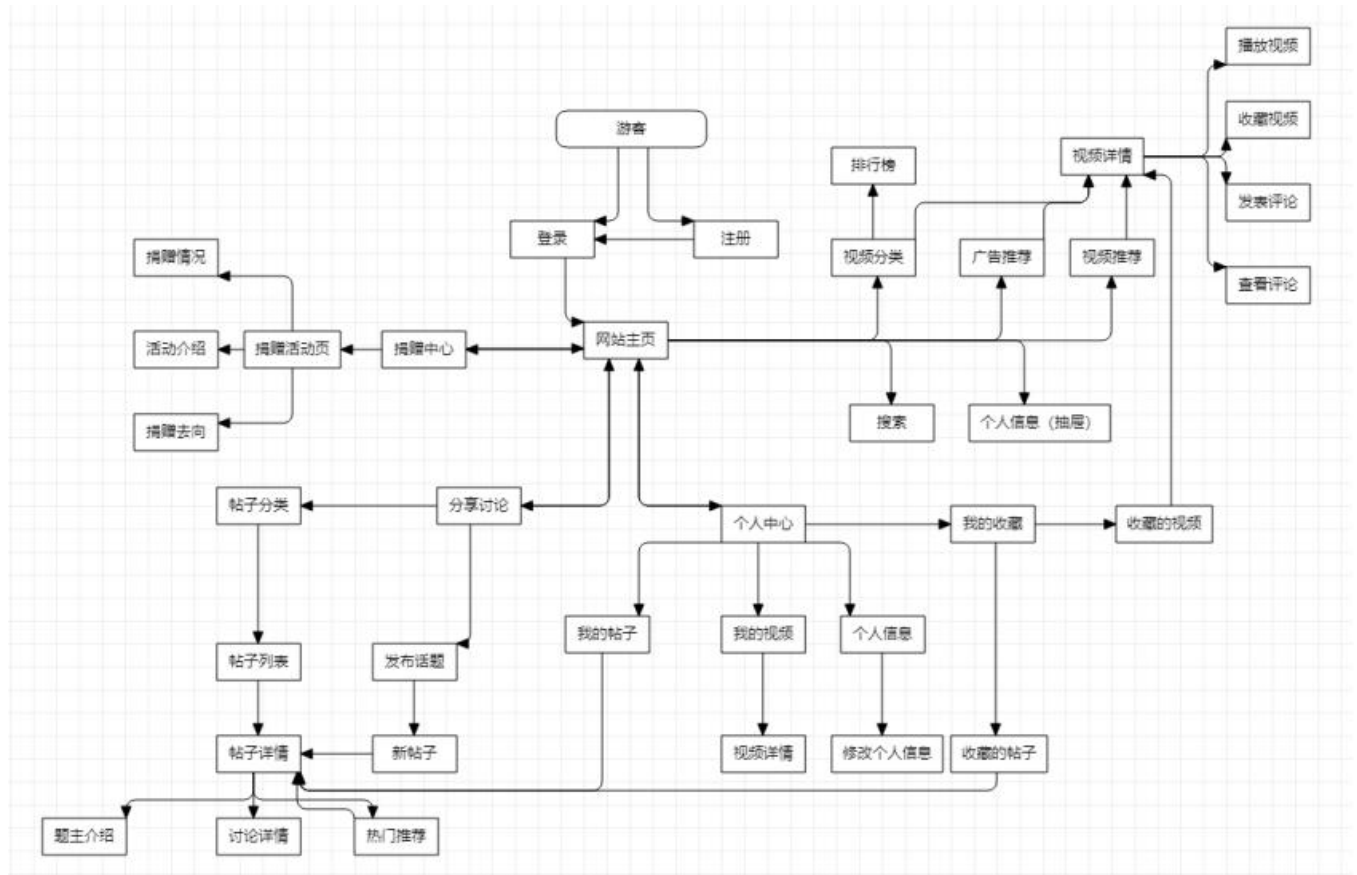


管理员用例图



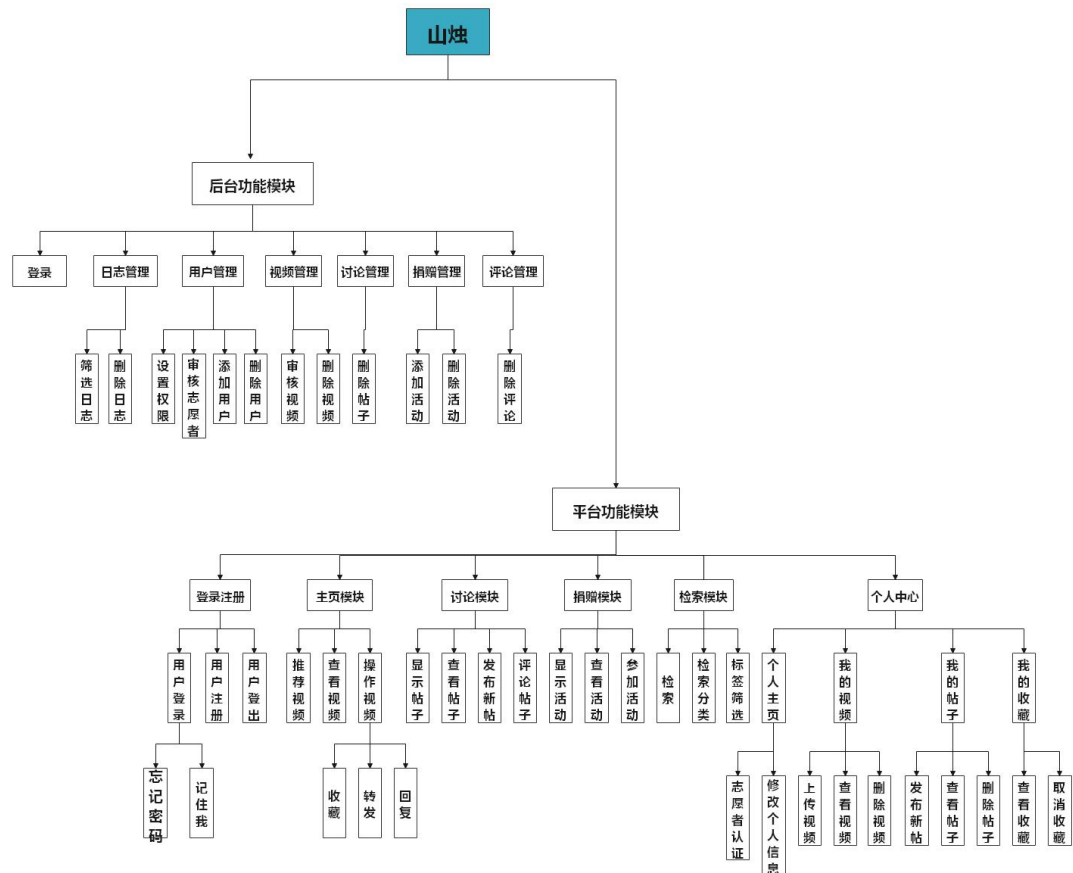
三. 系统功能模块设计

3.1 系统流程图



3.2 系统功能模块层次设计

根据业务和功能，将系统的逻辑结构划分为平台功能模块和后台功能模块，具体细节如下图所示：



3.3 系统功能模块详细设计

3.3.1 平台功能模块

3.3.1.1 主页模块

模块描述：平台的主页，有推荐视频，查看视频并对视频收藏，转发，评论等功能。

功能描述：

1. 推荐视频

功能描述：平台按标签分类向用户推荐各板块的热门视频。

输入数据：无。

返回信息：推荐的视频 id，封面，标题，简介，播放量等相关信息。

2. 查看视频

功能描述：用户点击视频简介，跳转至该视频的播放页面。

输入数据：无。

返回信息：该视频内容，发布者，关联的评论等信息。

3. 收藏视频

功能描述：收藏某个视频。

输入数据：无。

返回信息：收藏是否成功。

4. 转发视频

功能描述：转发某个视频。

输入数据：无。

返回信息：生成该视频的链接。

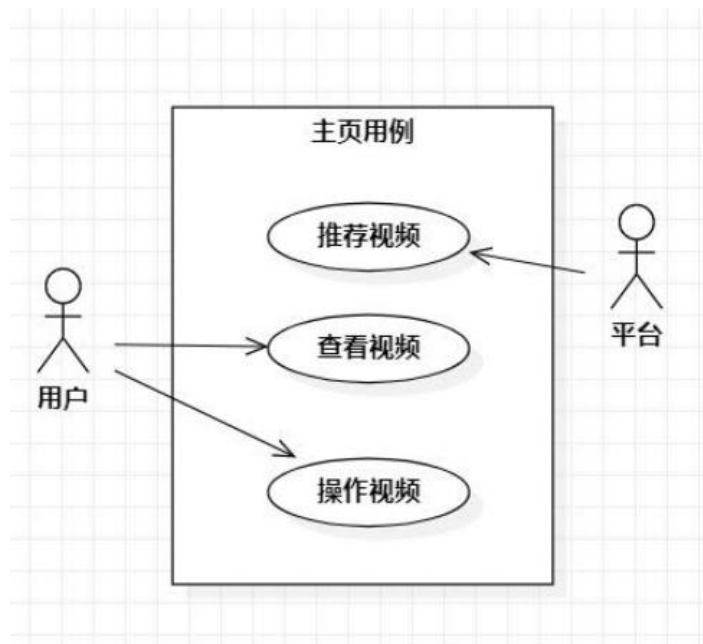
5. 评论视频

功能描述：评论某个视频

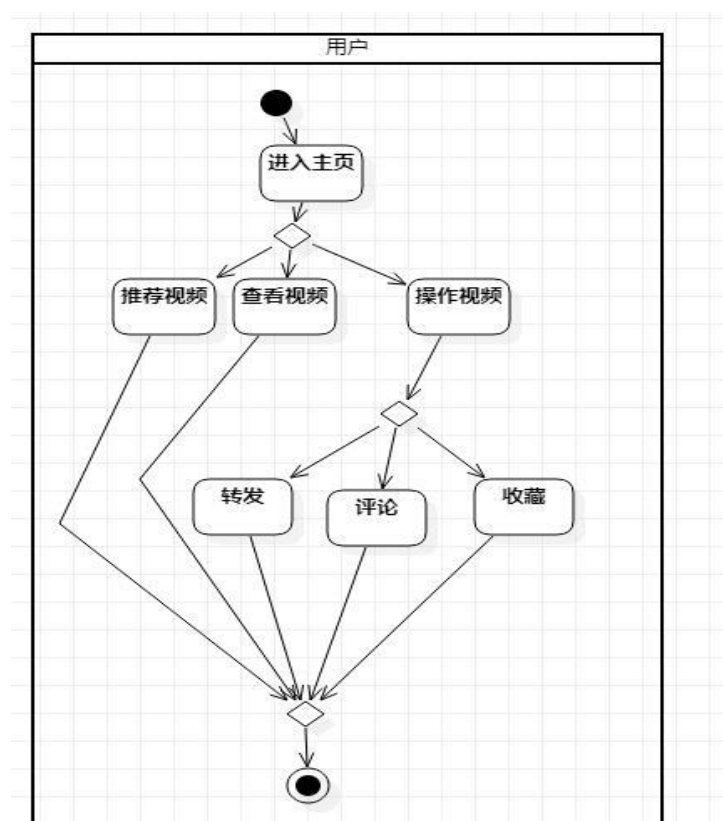
输入数据：评论内容。

返回信息：评论是否评论成功。

用例图



活动图



3.3.1.2 讨论模块

模块描述：平台的讨论区，有展示帖子，查看帖子，发布新帖并参与讨论，进行评论，举报的功能。

功能描述：

1. 展示帖子

功能描述：平台按分类向用户展示帖子，可按时间，热度排序。

输入数据：无。

返回信息：帖子信息列表。

2. 查看帖子

功能描述：用户点击帖子简介，跳转至帖子具体页面，显示帖子更多详细信息。

输入数据：无。

返回信息：帖子内容，相关评论，帖主等信息。

3. 发布新帖

功能描述：用户可发布新帖，通过管理员审核后，将会在讨论区展现。

输入数据：帖子信息表单。

返回信息：帖子是否上传成功信息，进入等待审核状态。

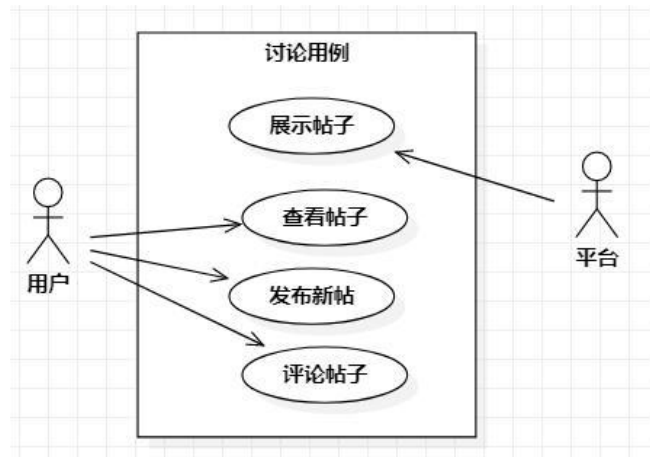
4. 评论帖子

功能描述：用户在某个帖子下进行评论。

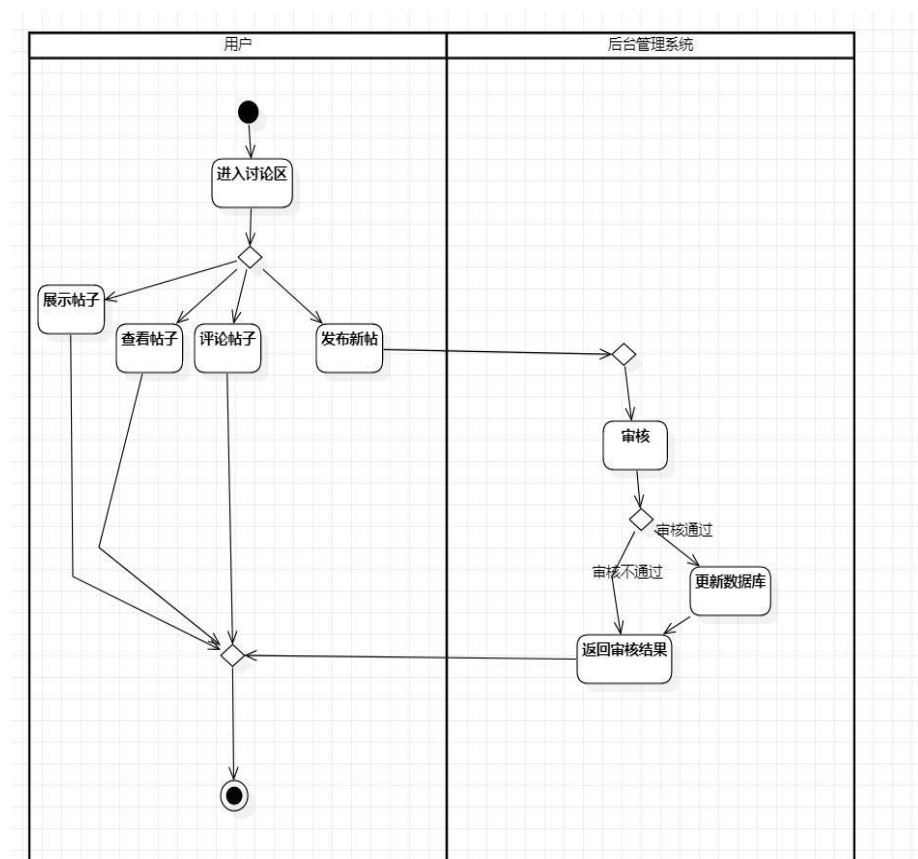
输入数据：评论的内容。

返回信息：评论是否发布成功信息。

用例图



活动图



3.3.1.3 捐赠模块

模块描述：向用户展示捐赠活动，用户可查看活动，参加活动

功能描述：

1. 展示活动

功能描述：向用户展示捐赠活动，按承办方（平台和第三方）分板块显示。

输入数据：无。

返回信息：捐赠活动的信息列表，活动属性。

2. 查看活动

功能描述：用户点击活动简介，跳转活动详细界面，显示活动更多详细信息。

输入数据：无。

返回信息：活动 id，标题，图片，详情，捐赠去向等信息。

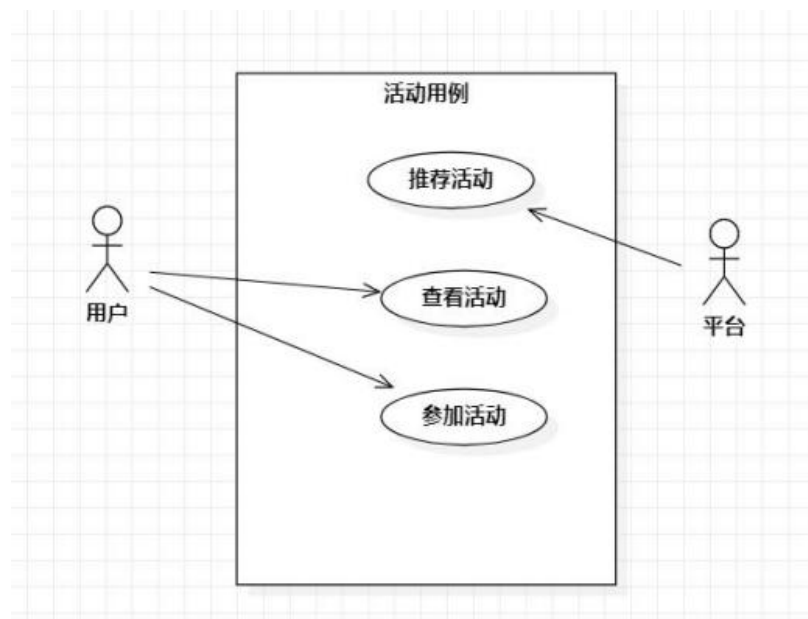
3. 参加活动

功能描述：用户参加活动，填写捐赠表单。

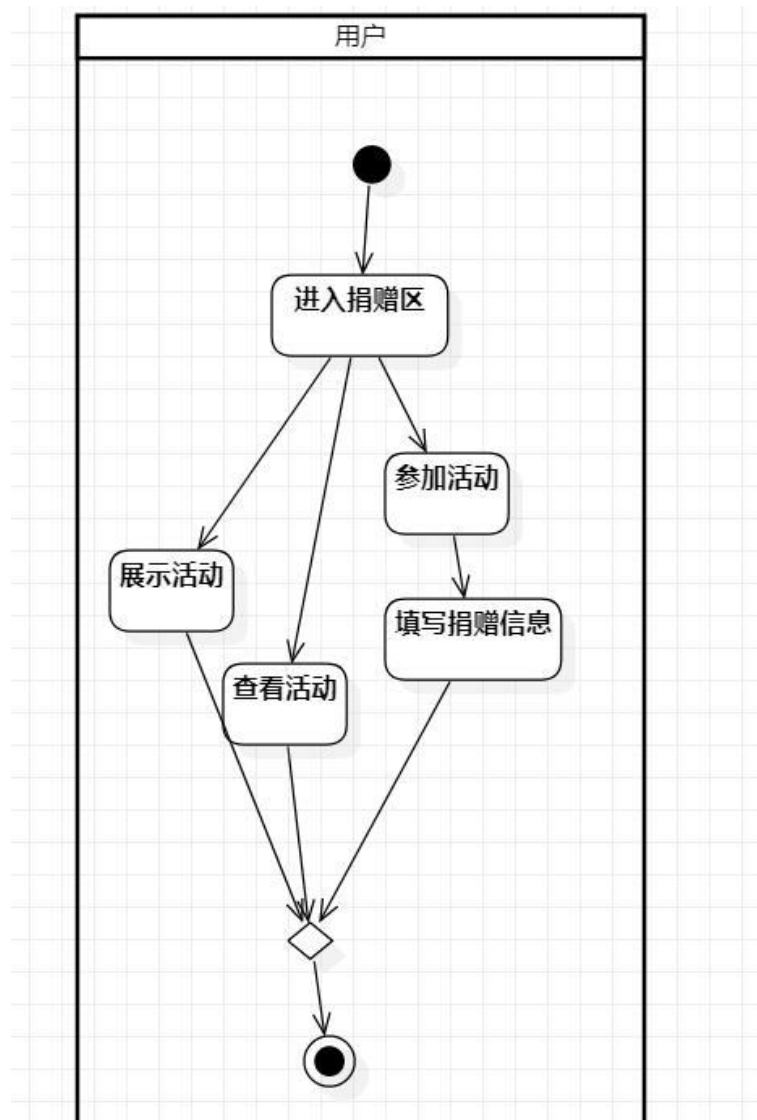
输入数据：活动 id，捐赠人实名，电话，捐赠物品清单，捐赠方式等信息。

返回信息：捐赠表单提交是否成功。

用例图



活动图



3.3.1.4 检索模块

模块描述：用户进行检索，检索结果可按标签，类别筛选

功能描述：

1. 检索

功能描述：用户输入检索信息，后台将检索内容返回给用户。

输入数据：检索内容。

返回信息：与检索内容相关的结果列表，分类标签。

2. 检索分类

功能描述：将检索得到的信息按视频，用户，帖子，捐赠分页显示。

输入数据：选择分类的标签。

返回信息：按标签显示的检索结果列表。

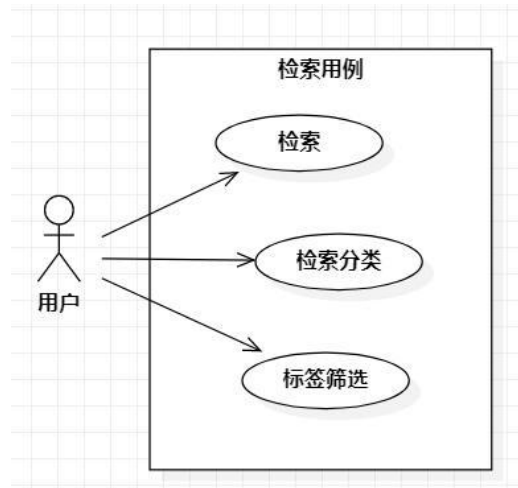
3. 标签筛选

功能描述：将检索得到的信息按选中的标签进行筛选。

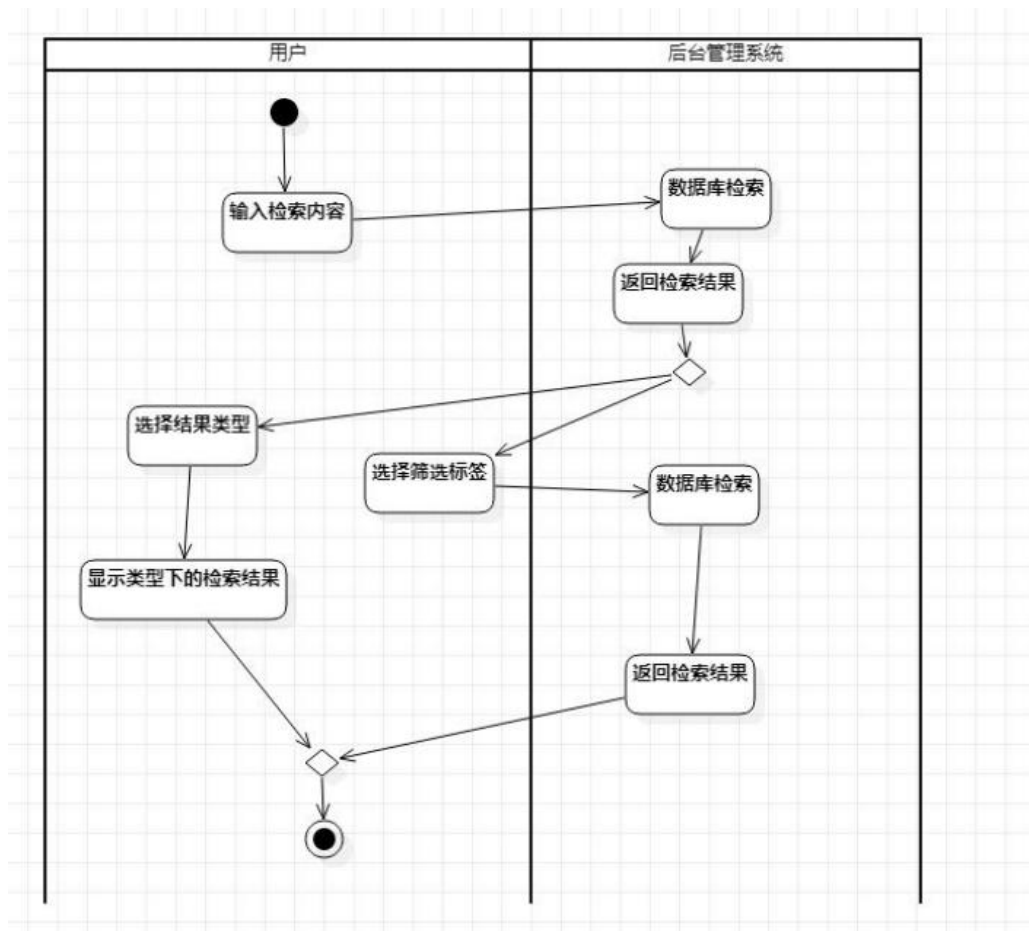
输入数据：选中的筛选标签。

返回信息：按标签筛选得到的检索结果列表。

用例图



活动图



3.3.1.5 登录注册模块

模块描述：用户登录注册平台，在登录过程具有忘记密码，记住我功能。

功能描述：

1. 注册

功能描述：游客填写个人信息，注册成为平台用户。

输入数据：注册个人信息表单，含昵称，密码，邮箱等信息。

返回信息：邮箱验证码，注册是否成功信息。

2. 登录

功能描述：用户填写用户名，密码，登录平台。

输入数据：用户名，密码。

返回信息：登录是否成功。

3. 忘记密码

功能描述：用户忘记密码，可通过邮箱找回密码。

输入数据：邮箱地址。

返回信息：找回是否成功，成功显示用户密码；失败则返回找回失败，及失败原因。

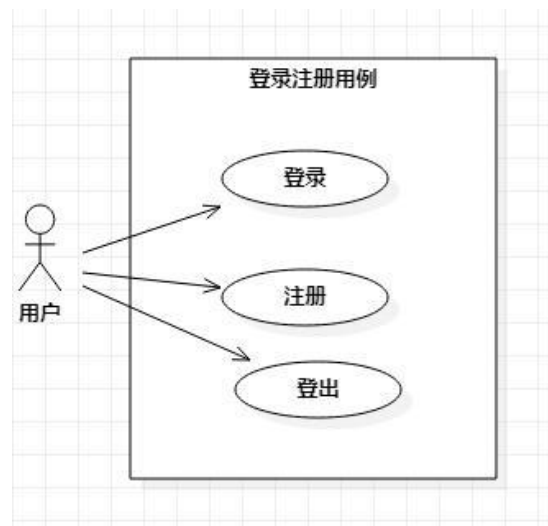
4. 记住我

功能描述：用户在登录时选择记住我，再次登录平台可以自动登录。

输入数据：用户名，用户密码。

返回信息：无。

用例图



The diagram illustrates the user management process, divided into two swimlanes: **用户** (User) and **后台管理系统** (Backend Management System).

用户 (User) Swimlane:

- Start:** A solid black circle representing the start of the process.
- Decision:** A diamond shape that branches into two paths: **注册** (Registration) and **登录** (Login).
- 注册 (Registration):** A use case that leads to **填写注册信息** (Fill registration information).
- 填写注册信息:** An activity that leads to **更新用户表** (Update user table) in the Backend Management System.
- 登录 (Login):** A use case that branches into two paths based on login status:
 - 上次登录, 未勾选记住我** (Last login, "Remember me" not checked): Leads to **返回登录成功** (Return login success).
 - 上次登录, 勾选记住我** (Last login, "Remember me" checked): Leads to **返回登录成功**.
- 忘记密码 (Forgot password):** A use case that leads to **填写邮箱** (Fill email).
- 填写邮箱:** An activity that leads to **发送验证码** (Send verification code) in the Backend Management System.
- 获得密码 (Get password):** A use case that leads back to **登录**.
- 输入登录信息 (Enter login information):** An activity that leads to **验证信息** (Verify information) in the Backend Management System.
- 验证信息:** A use case that leads to a decision diamond.
- Decision:** A diamond shape that branches into two paths:
 - 信息正确** (Information correct): Leads to **返回登录成功**.
 - 信息错误** (Information incorrect): Leads to **返回登录失败** (Return login failure).
- 返回登录成功:** A use case that leads to a final decision diamond.
- 返回登录失败:** A use case that leads to the final decision diamond.
- 登出 (Logout):** A use case that leads to the final decision diamond.
- End:** A bullseye symbol representing the end of the process.

后台管理系统 (Backend Management System) Swimlane:

- 更新用户表 (Update user table):** A use case triggered by **填写注册信息**.
- 发送验证码 (Send verification code):** A use case triggered by **填写邮箱**.
- 验证信息 (Verify information):** A use case triggered by **输入登录信息**.

模块描述：个人中心分为个人主页，我的视频，我的帖子，我的收藏四个小模块。

功能描述：

志愿者认证

功能描述：用户通过填写真实信息，上传身份证照片，完成实名

认证，成功方可参见捐赠活动等。

输入数据：用户真实信息表单，身份证照片。

返回信息：表单上传是否成功。

修改个人信息

功能描述：用户可编辑个人信息。

输入数据：用户修改后的个人信息。

返回信息：修改后信息是否规范，规范返回修改成功；反之显示修改失败及原因。

2. 我的主页

上传视频

功能描述：用户可在该页面上传发布新视频。

输入数据：视频名称，简介，标题等信息。

返回信息：是否成功上传视频，进入待审核状态。

查看视频

功能描述：用户点击视频简介，跳转至该视频的播放页面。

输入数据：无。

返回信息：该视频内容，发布者，关联的评论等信息。

删除视频

功能描述：用户对已经发布成功的视频进行删除。

输入数据：无。

返回信息：删除是否成功。

3. 我的帖子

发布新帖

功能描述：用户可在该页面发布新帖。

输入数据：帖子名称，简介，图片等信息。

返回信息：是否成功上传帖子。

查看帖子

功能描述：用户点击帖子简介，跳转至该帖子的详细页面。

输入数据：无。

返回信息：该帖子内容，发布者，关联的评论等信息。

删除帖子

功能描述：用户对已经发布成功的视频进行删除。

输入数据：无。

返回信息：删除是否成功。

4. 我的收藏

查看收藏

功能描述：用户可查看自己收藏的视频。

输入数据：无。

返回信息：收藏视频的信息列表。

取消收藏

功能描述：用户可取消对某个视频的收藏。

输入数据：无。

返回信息：取消操作是否成功。

5. 个人小窗

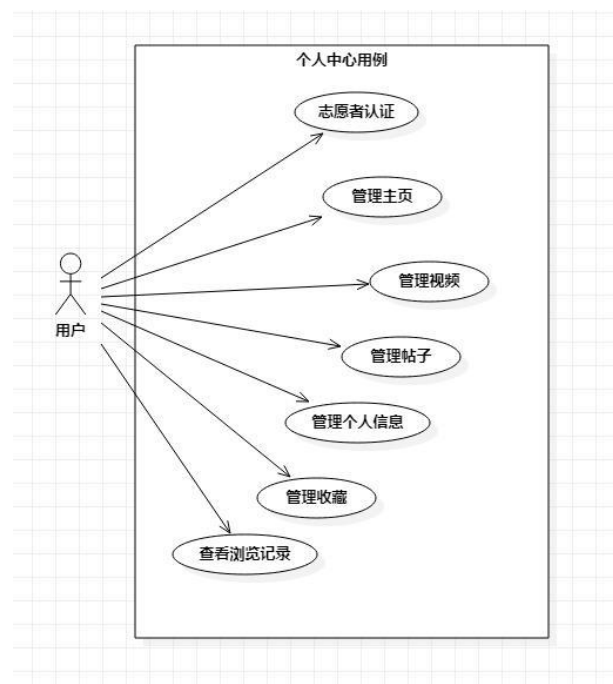
查看浏览记录

功能描述：用户可以查看最近的视频，帖子浏览记录。

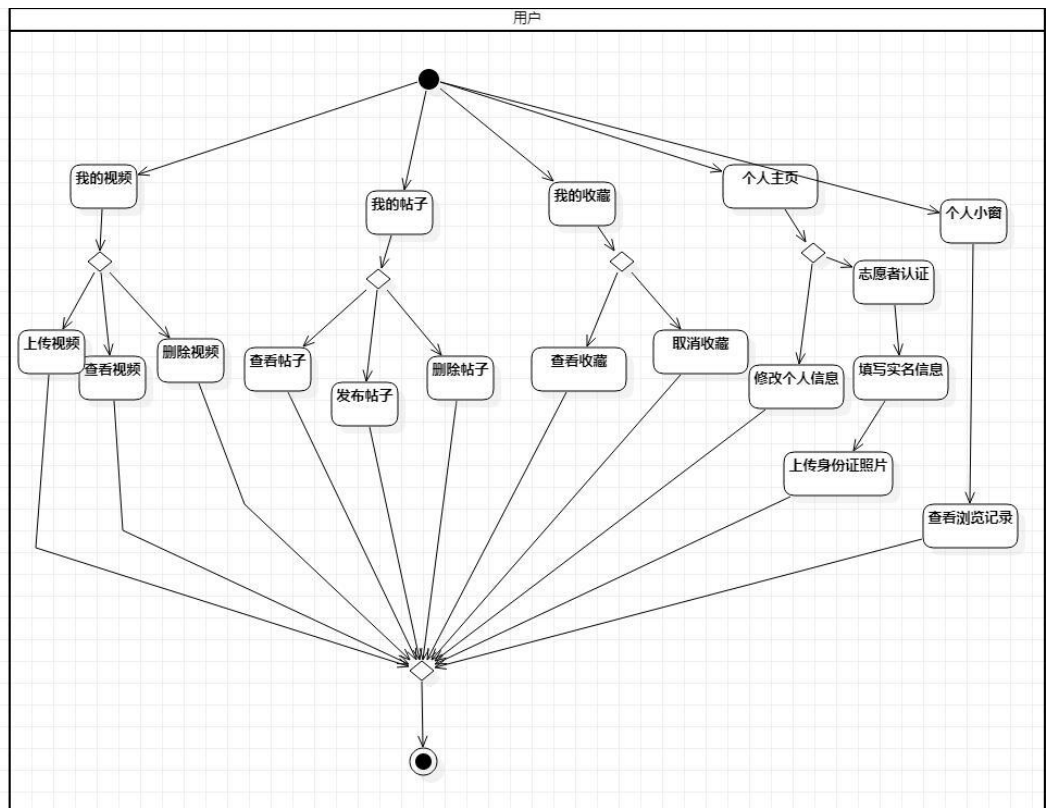
输入数据：无。

返回信息：返回相应的用户浏览视频，帖子记录信息。

用例图



活动图



3.3.2 后台功能模块

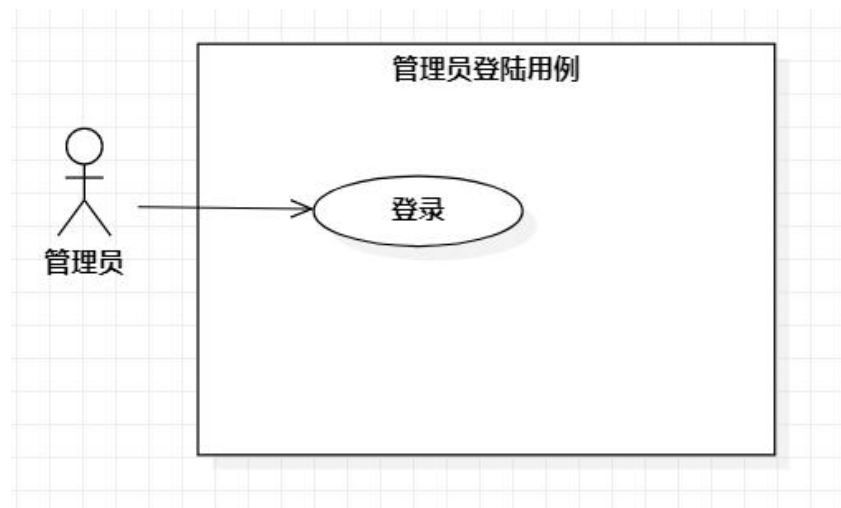
3.3.2.1 管理员模块

模块描述：管理员登陆平台

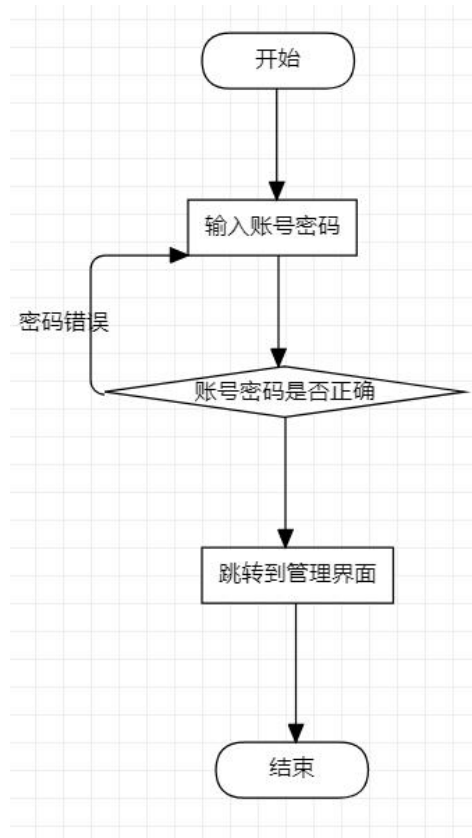
功能描述：管理员填写账号密码，登录平台

输入数据：用户名、密码

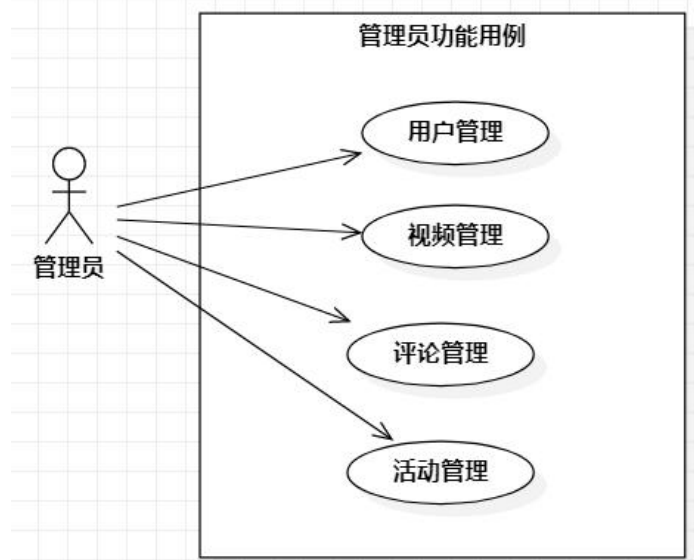
用例图：



活动图：



3.3.2.2 管理模块



3.3.2.2.1 用户管理模块

模块描述：管理员对用户查看用户、删除，添加用户，角色设置，志愿者审核的管理

功能描述：

1. 用户删除

功能描述：实现对用户的权限设置和删除用户。

输入数据：无。

返回信息：删除用户是否成功。

2. 添加用户：

功能描述：添加新用户 。

输入数据：用户名，密码、邮箱、手机号 。

返回信息：新的用户。

3. 用户角色设置

功能描述：修改用户的角色 。

输入数据：无。

返回信息：新的用户角色 。

4. 查看用户

功能描述：查看用户信息 。

输入数据：查询字 。

返回信息：用户信息。

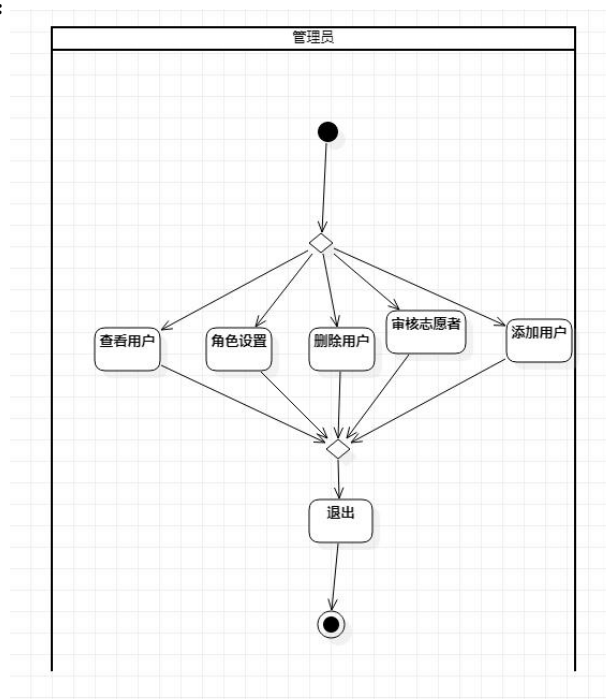
5. 志愿者审核

功能描述：审核支志愿者。

输入数据：无。

返回信息：审核是否通过 。

活动图：



3.3.2.2.2 视频管理模块

模块描述： 实现对视频的筛选，审核，删除功能

功能描述：

1. 视频筛选

功能描述： 实现视频筛选。

输入数据： 类型、视频状态、查询字。

返回信息： 符合条件视频列表。

2. 视频审核

功能描述： 实现视频审核。

输入数据： 无。

返回信息： 视频是否通过审核。

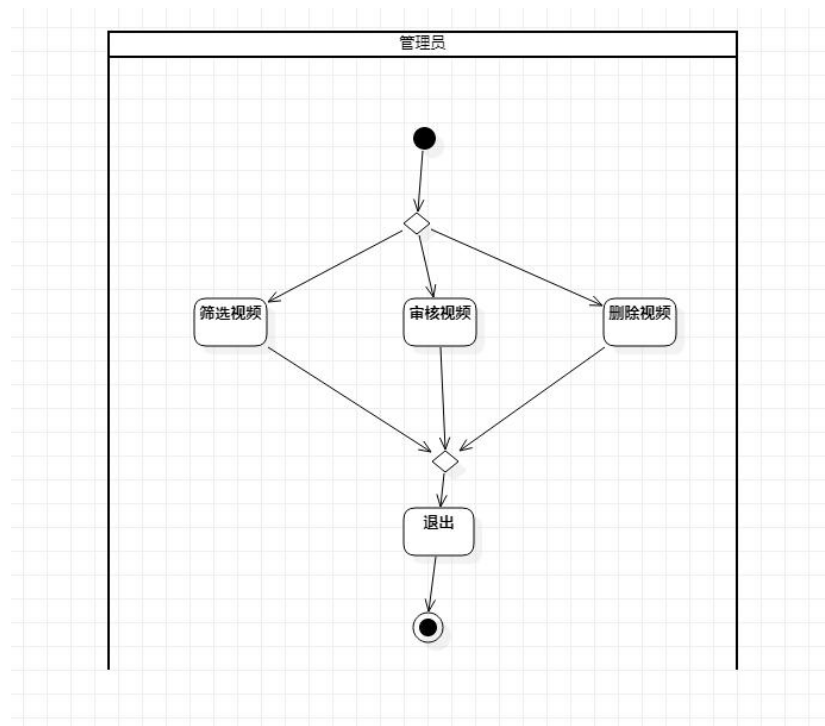
3. 删除视频

功能描述： 删除选中的视频。

输入数据： 无。

返回信息： 视频是否删除成功。

活动图：



3.3.2.2.3 讨论管理模块

模块描述： 实现讨论的查看、删除和删除评论

功能描述：

1. 查看讨论

功能描述： 查看讨论详情。

输入数据： 讨论类型、关键字。

返回信息： 讨论详情。

2. 删除讨论

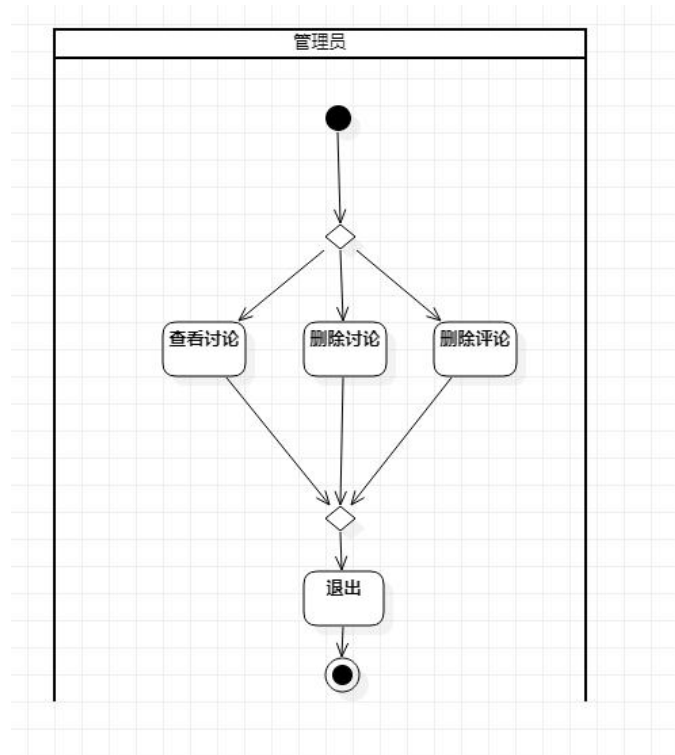
功能描述： 删除对应讨论。

输入数据： 无。
返回信息： 讨论是否删除成功。

3. 删除评论

功能描述： 删除对应评论。
输入数据： 无。
返回信息： 评论是否删除。

活动图：



3.3.2.2.4 捐赠管理模块

模块描述： 实现对捐赠活动的查看、删除和审核功能

功能描述：

1. 查看活动

功能描述： 查看活动详情。
输入数据： 活动类型、活动关键字、活动状态。
返回信息： 详细活动信息。

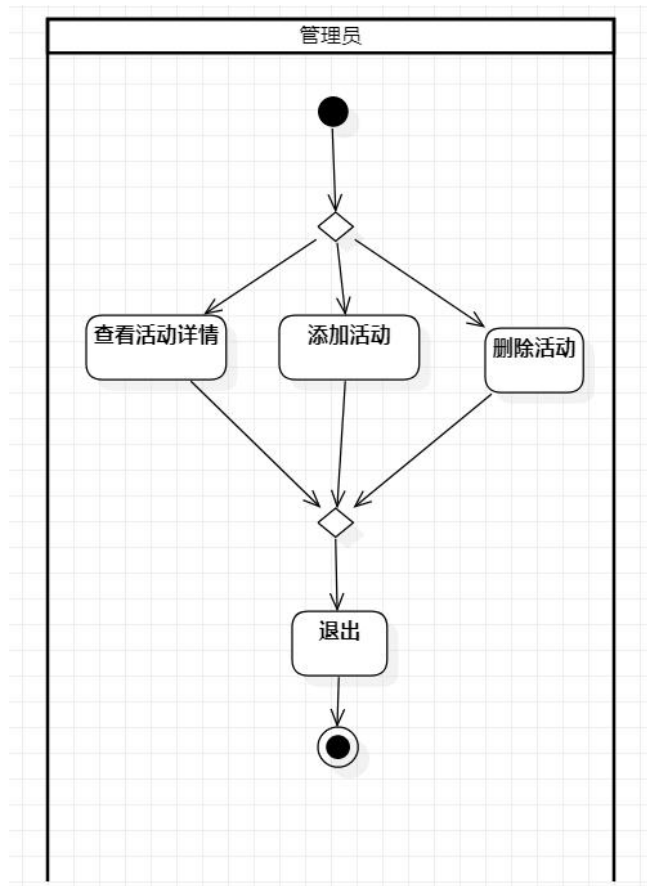
2. 删除活动

功能描述： 删除选中活动。
输入数据： 无。
返回信息： 活动是否删除。

3. 添加活动

功能描述： 实现新活动的审核。
输入数据： 活动 id，捐赠详情、捐赠去向。
返回信息： 活动是否成功添加。

活动图：



3.3.2.2.5 日志管理模块

模块描述： 实现对日志的查看和删除。

功能描述：

1. 查看日志

功能描述： 查看日志详情。

输入数据： 查询字、类型。

返回信息： 对应日志的详细信息。

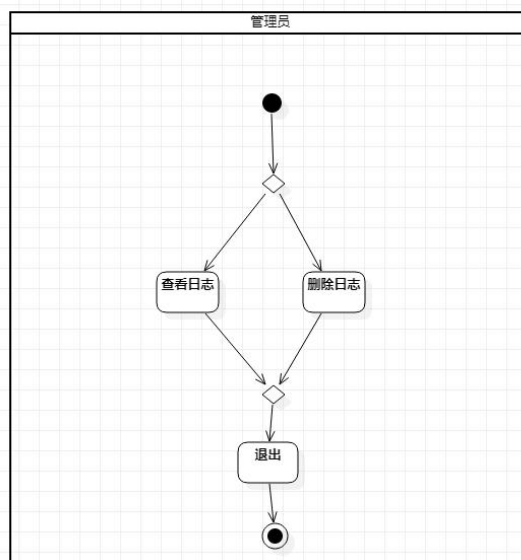
2. 删除日志

功能描述： 删除对应日志。

输入数据： 无。

返回信息： 日志是否成功删除。

活动图：

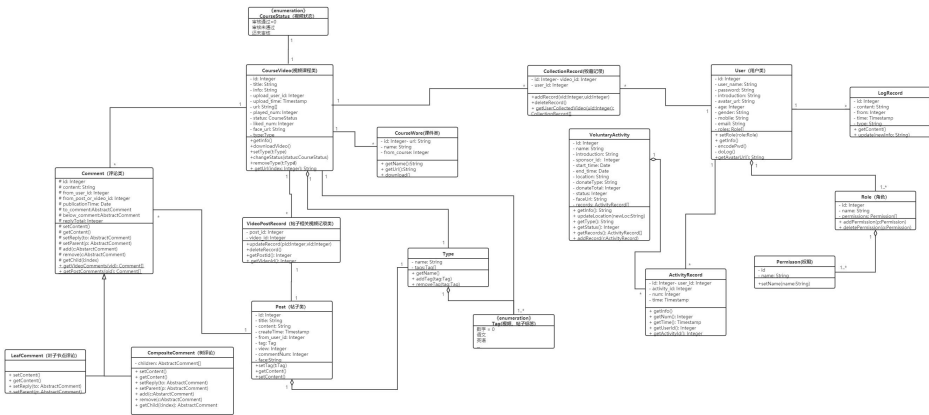


四. 类图展示与算法说明

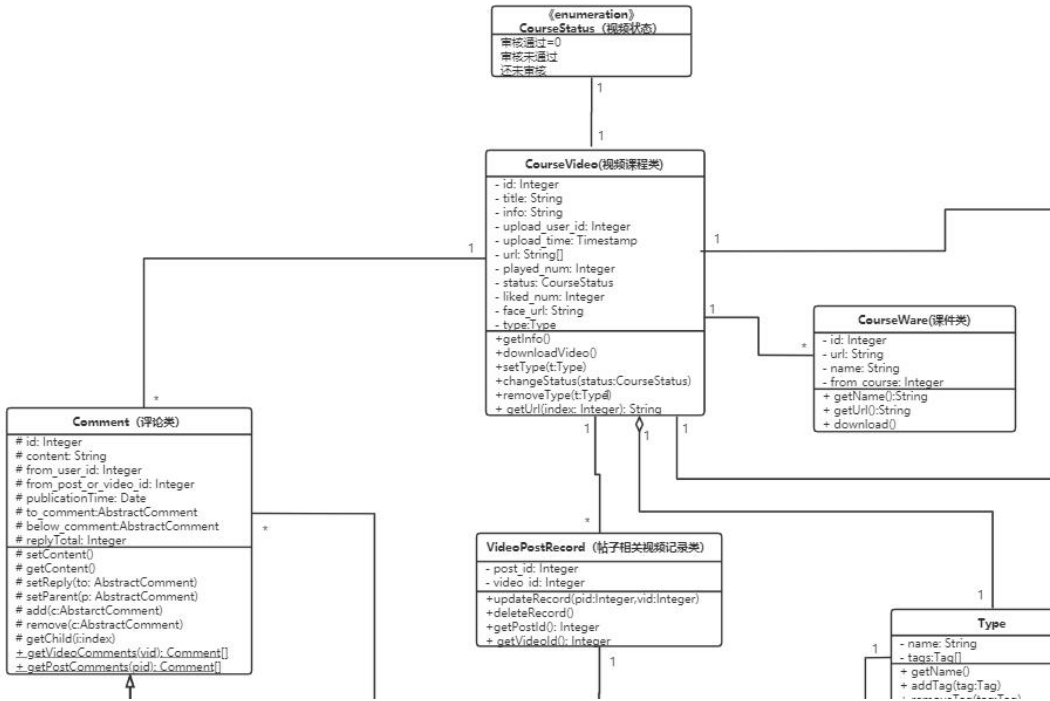
4.1 类图展示

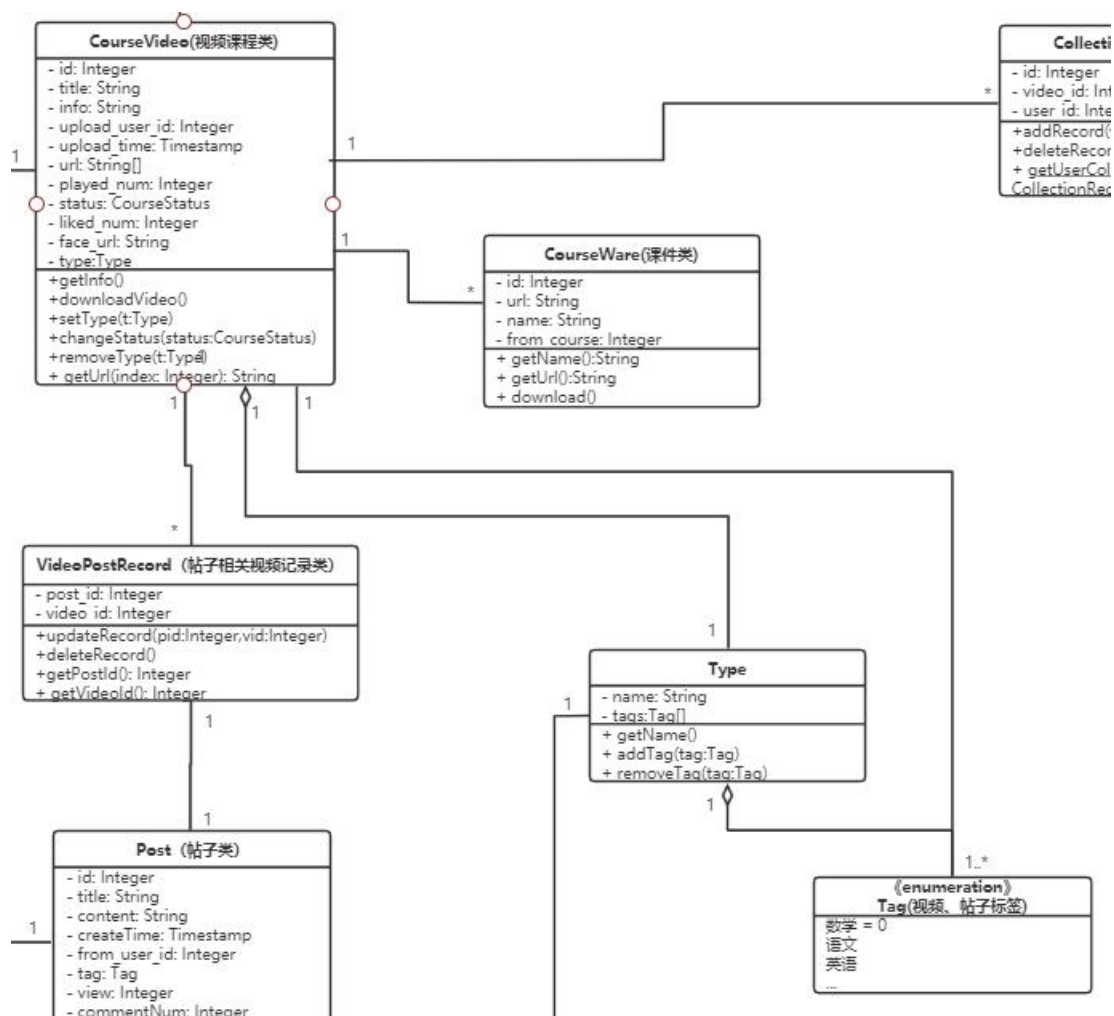
4.1.1 总览

(可右键图片后选择“预览”放大)



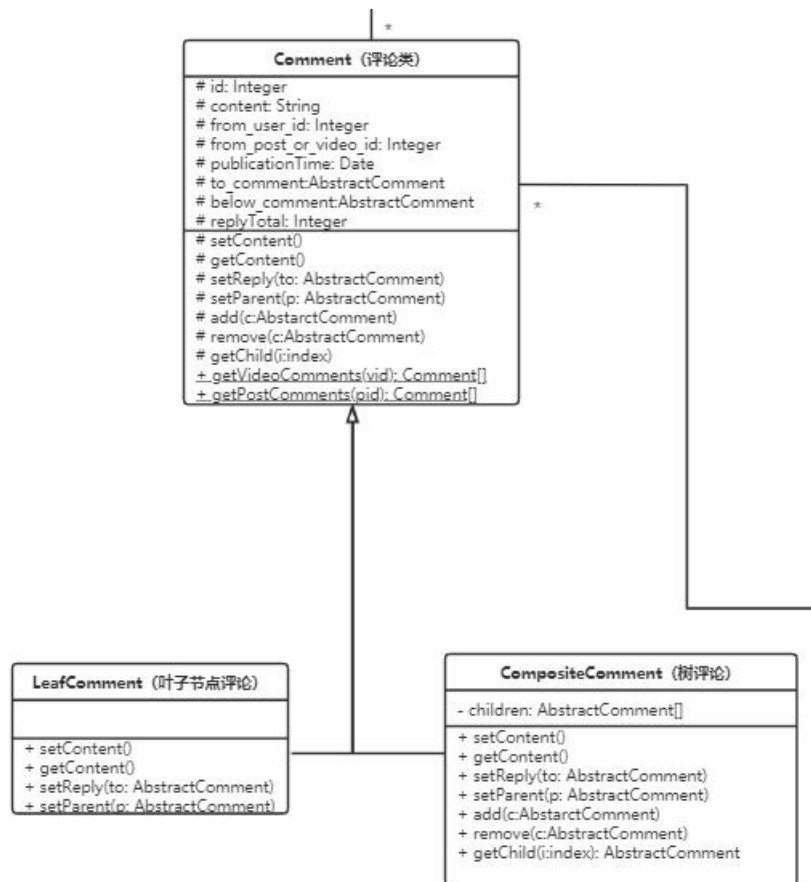
4.1.2 CourseVideo 视频课程类





该类描述一个用户上传的视频，主要属性包括了视频 id、视频标题 title、视频简介 info、上传视频的用户 id、视频上传时间、视频地址 url（url 为数组，代表用户分 p 上传视频，即一次上传多个子视频）、视频播放数量 played_num、视频状态 status（枚举值，包括未审核、审核未通过、审核通过）、视频收藏数量 liked_num、视频封面地址、视频类型 Type（启蒙、初中、小学等，一个视频类型包含多个标签 Tag，例如语文、数学、英语等）。

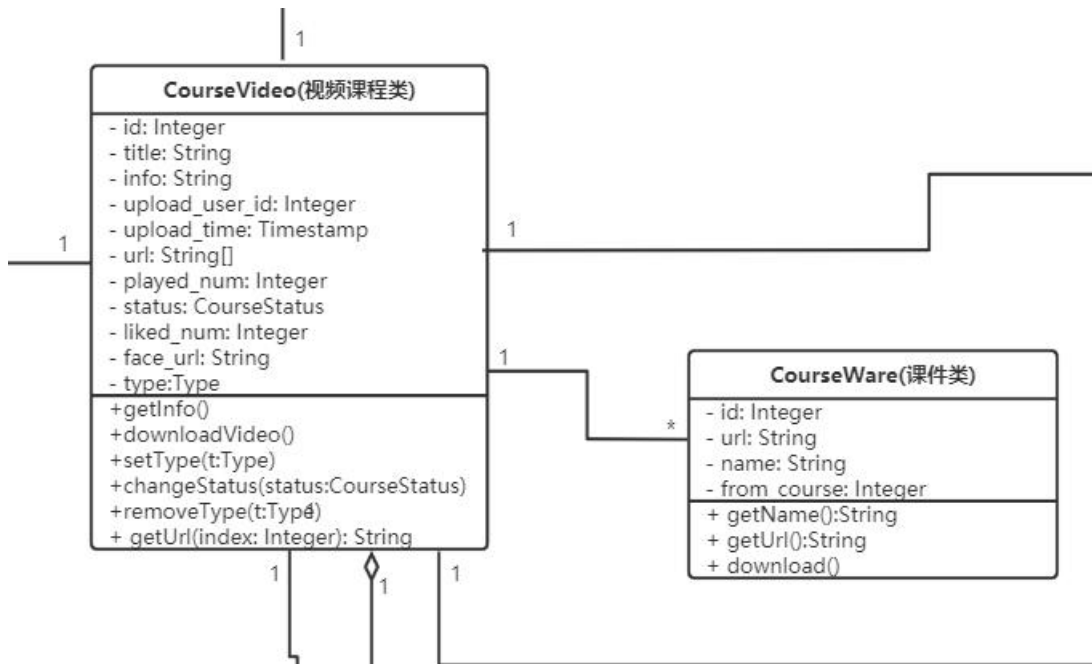
4.1.3 Comment、LeafComment、CompositeComment 评论类



该类描述了每条评论所需要的信息（包括视频评论和帖子评论），主要的属性有：评论 id、评论内容 content、评论所属用户的 id、评论所属视频或帖子的 id、评论发布的时间、回复的评论（若为空则说明本条评论不是一条回复其他评论的评论）、父评论（若为空则说明本条评论不是嵌套在某条评论之下）、回复评论的总数量。

考虑到评论呈现树形结构，所以这里采用了组合设计模式，包括了叶子结构的评论（即该评论下不再包含别的评论）和树结构的评论（评论下还包含其他的评论）。

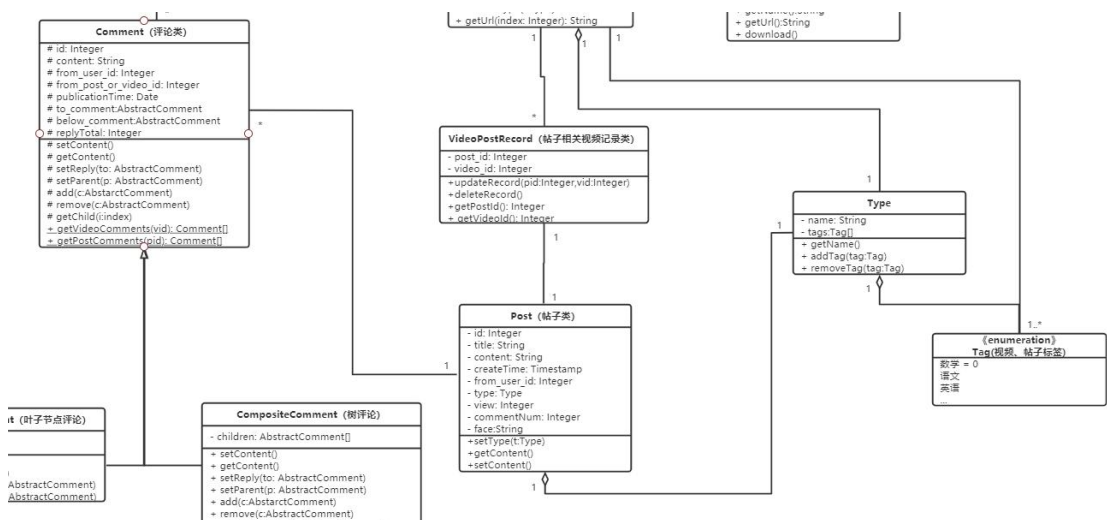
4.1.4 CourseWare 课件类



该类描述了每个课件所需要的信息，主要的属性有：课件 id、课件下载 url、课件的名字 name、课件所属的视频 id。

视频和课件是一对多的关系，一个视频对应多个课件。

4.1.5 Post 帖子类



该类描述了每个帖子所需要的信息，主要的属性有：帖子 id、帖子标题 title、帖

子简介 content、帖子创建的时间 createTime、发布帖子的用户 id fromUser、帖子的类别、帖子的历史浏览人数 view、帖子评论数量 commentNum、帖子封面地址 face。

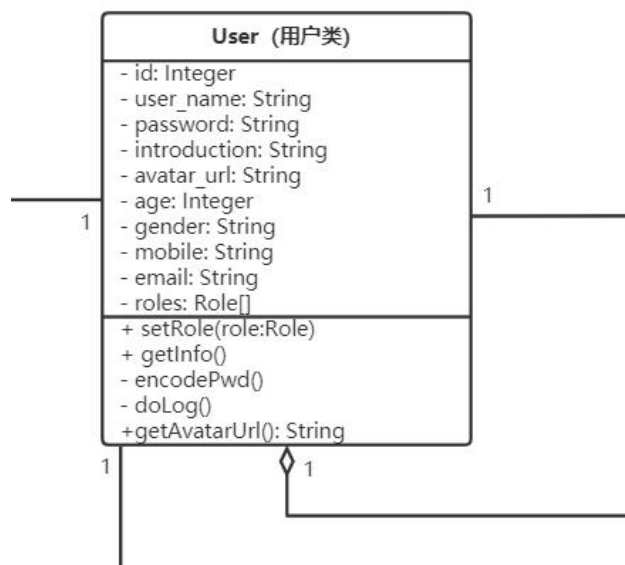
帖子与视频类共用一个 Type 类型。

4.1.6 VideoPostRecord 帖子视频关联类



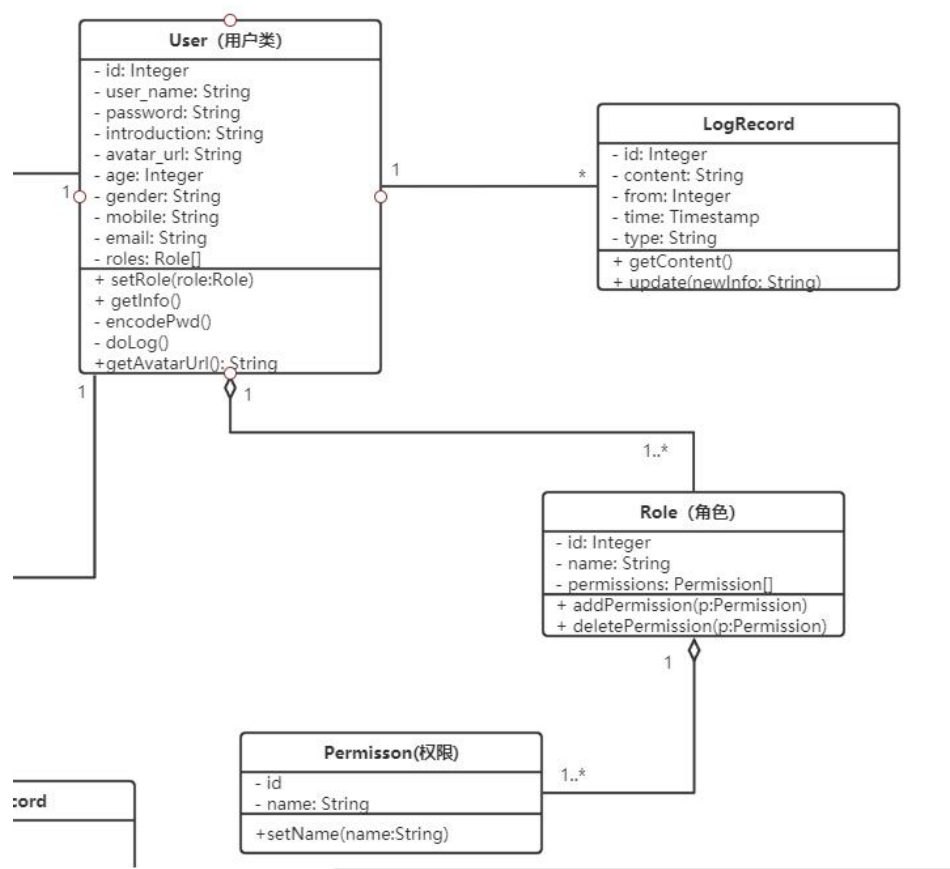
该类为 CourseVideo 视频和 Post 帖子的关联类，主要用于在每个视频的旁边推荐相关帖子。该类有两个属性，分别对应帖子的 id 和视频的 id。

4.1.7 User 用户类



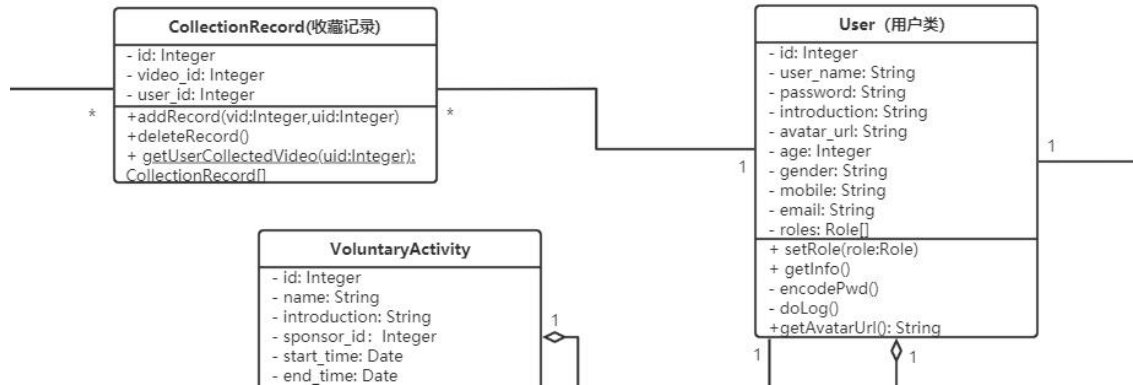
该类描述了系统中每个用户所需要的信息，主要的属性有：用户 id、用户名字 user_name、登录密码 password、用户简介 introduction、用户头像地址 avatar_url、用户年龄 age、用户性别 gender、用户手机号 mobile、用户邮箱 email、以及用户在系统中所扮演的角色 roles。

4.1.8 Permission 权限类和 Role 角色类



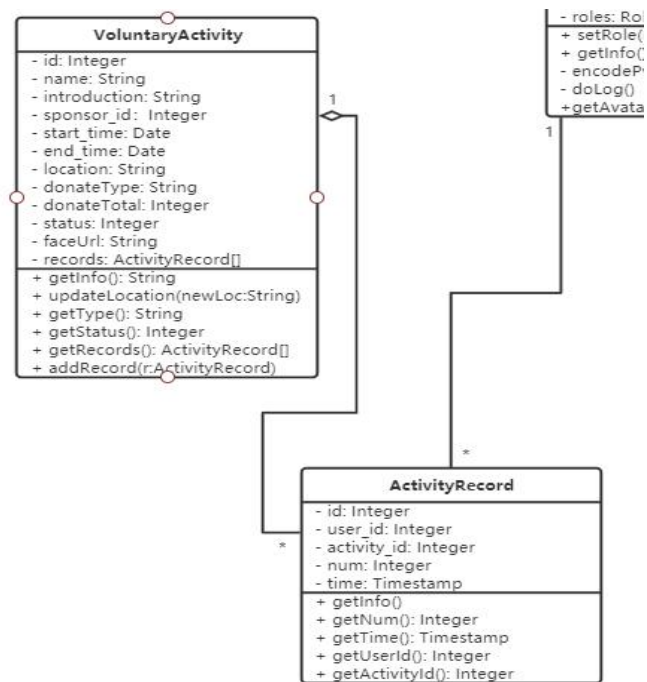
本系统使用 RBAC 权限控制，RBAC（Role-Based Access Control，基于角色的访问控制），就是用户通过角色与权限进行关联。简单地说，一个用户拥有若干角色，每一个角色拥有若干权限。这样，就构造出“用户-角色-权限”的授权模型。在这种模型中，用户与角色之间，角色与权限之间，一般者是多对多的关系。图中 User 类为用户类，User 类拥有多个 Role 类的实例，而一个 Role 类又拥有多个 Permission 类的实例。

4.1.9 CollectionRecord 用户收藏记录类



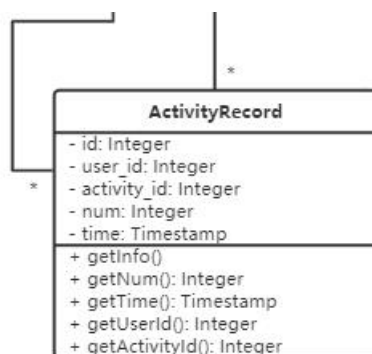
该类记录了用户收藏的视频，拥有属性有：用户 id 和视频 id；可以用过该类的静态方法来获取每个用户的收藏视频。

4.1.10 VoluntaryActivity 捐赠志愿活动类



该类描述了一个捐赠活动所需要的信息，主要属性有：活动的 id、活动名称 name、活动简介 introduction、发起人 id、开始时间 start_time、结束时间 end_time、活动地点 location、活动的类型、捐赠总数、活动的状态、活动封面的地址、以及该活动所拥有的参加记录（ActivityRecord 数组）。

4.1.11 ActivityRecord 活动记录类



该类作为 VoluntaryActivity 和 User 的中间类，记录每个用户所参加的活动，主要属性包括用户 id、活动的 id、捐赠数量、捐赠时间。

4.2 推荐算法

随着深度学习的火热，越来越多的问题都开始尝试采用深度学习算法来解决，包括推荐算法。16 年的时候，谷歌公开了 YouTube 的推荐算法，Deep Neural Networks for YouTube Recommendation，论文地址：<https://www.sci-hub.ren/10.1145/2959100.2959190>。该论文采用了深度学习算法，在效果上超越了最常用的矩阵分解算法，接下来将简单介绍一下该推荐算法的架构和原理。

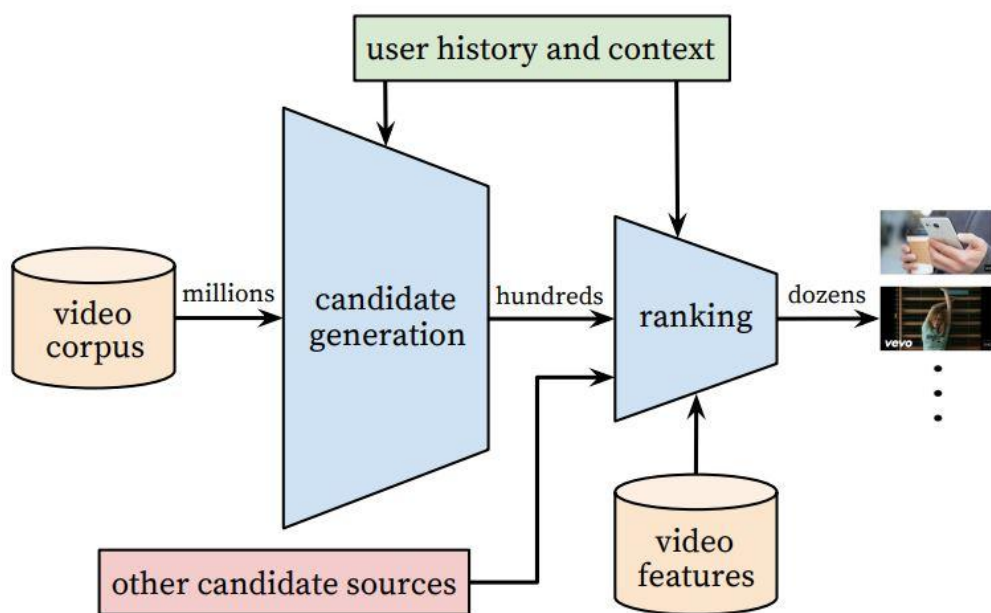


图 1 YouTubeNet 整体架构

整个推荐架构分为两部分，召回和排序，如上图所示。第一个蓝色的漏斗就是召回算法（candidate generation，候选产生），从百万级数量的视频物料库中筛选出几百个视频。除了这里采用的深度学习召回算法，还可以加入其他的召回视频源（如图中的红色方格所示），一起送给排序算法。因为计算量大，所以召回算法不可能也没必要采用所有特征，因此召回算法只采用了用户行为和场景特征（上图绿色方框所示，history and context）。

排序算法使用了更多的特征，给每个候选视频计算一个分数，并且按照分数从高到低排序，从几百个视频里边再筛选和排序出几十个视频推荐给用户。

4.2.1 召回算法

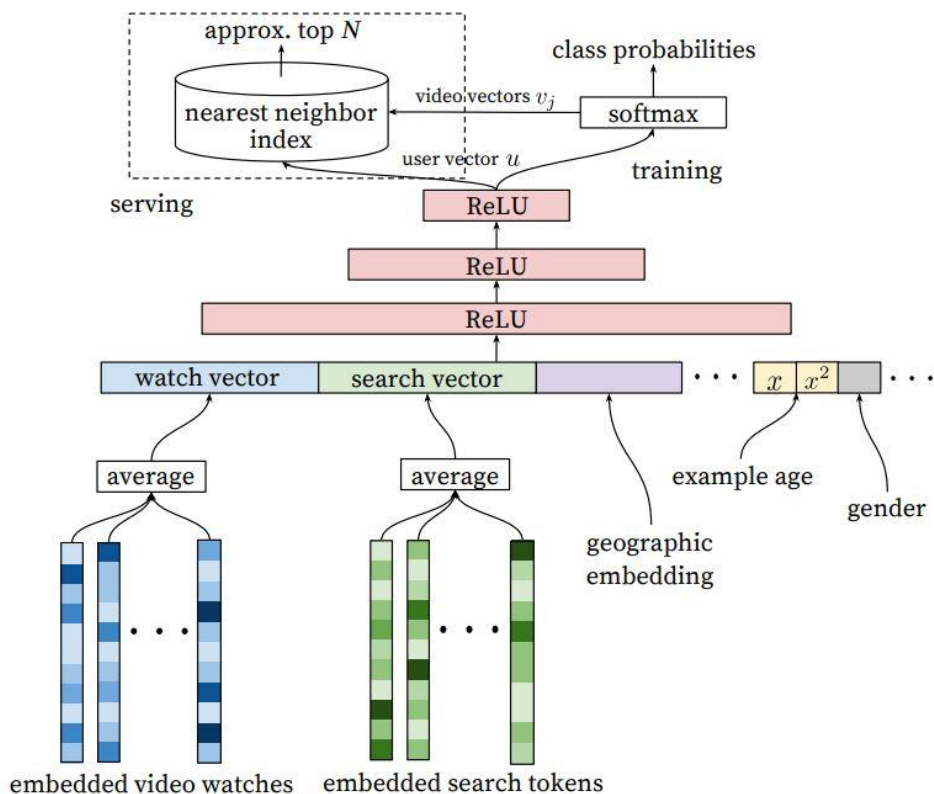


图 2 召回算法模型架构

召回阶段 (candidate generation) 就是要从推荐的百万级别的视频库中筛选出用户可能感兴趣的视频, 把推荐的视频库量级从百万级降到几百个。但是 YouTube 每秒同时访问的人数成千上万, 要同时实现每秒对每个用户都个性化的从百万视频候选集中挑出几百个用户感兴趣的视频, 就不可能每次做用户召回都跑一遍模型。其解决方法就和接下来要介绍的 embedding 应用相关。

在离线的模型训练阶段, 首先对于用户的历史观看视频做一个 embedded 操作, 将视频的特征信息 (CourseVideo 对象里包括的属性) 提取为一个向量, 再对所有视频向量求平均, 作为输入特征的一部分; 对于历史观看记录同理; geographic embedding, 代表了用户的地理信息, 本次实践将利用用户的特征信息 (User 对象里包括的属性) 替换地理信息; example age, 表示视频上传时间, 加入这个特征, 能够让模型知道视频的上传时间, 从而能够模拟视频发布之后的时间热度; gender, 代表用户的性别, 已经被包含在用户特征信息当中。

将以上信息进行一个向量拼接，再经过三层全连接的 ReLU 神经网络层，就可以得到用户特征向量（图中的 user vector u ）。将 user vector 输入 softmax 层，即可得到每个视频被推荐的概率，最后取概率最高的 N 个视频作为召回结果。

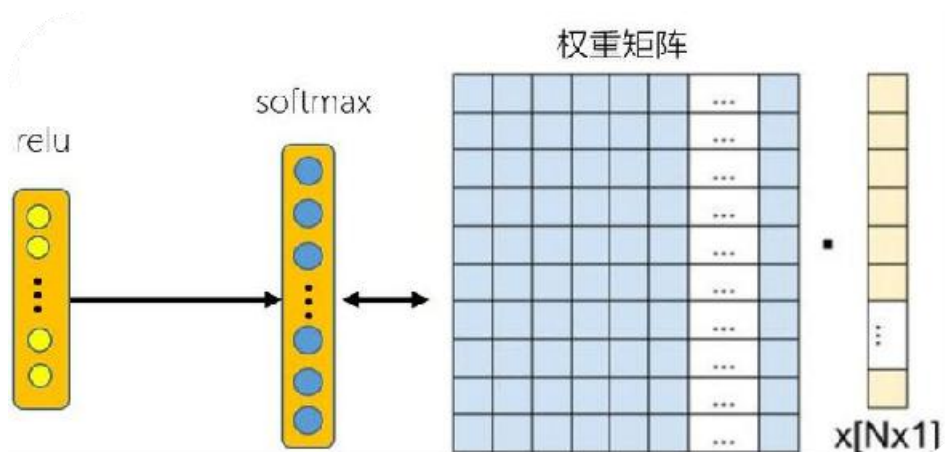


图 3 softmax 层内部原理

上图展示了 softmax 全连接层的内部原理，这里的 x 代表输入，就是用户特征向量 user vector，假设其是一个 $N \times 1$ 的向量；权重矩阵是一个 $T \times N$ 的矩阵，其中 T 代表了神经元的个数，也等于推荐视频的总数，权重矩阵里的每一行都是每一个 video 的 embedding，即每一个视频的特征向量；权重矩阵和 user vector 相乘的结果是一个 $T \times 1$ 的向量，在该向量上做一个 softmax 操作，即得到每一个视频被推荐的概率。

$$\text{softmax}(w_t = i|U, C) = \frac{e^{v_i \cdot u}}{\sum_{j \in V} e^{v_j \cdot u}}$$

图 4 softmax 计算公式

softmax 函数能将一个含任意实数的 T 维向量“压缩”到另一个 T 维实向量中，使得向量中每一个元素的范围都在 $(0, 1)$ 之间，并且所有元素的和为 1。

考虑到实现的因素，我们可以提前将 User Embedding 和 video Embedding 存储到线上内存数据库，通过内积运算再排序的方法得到 video 的排名，以此来提高召回速度和效率。

4.2.2 排序算法

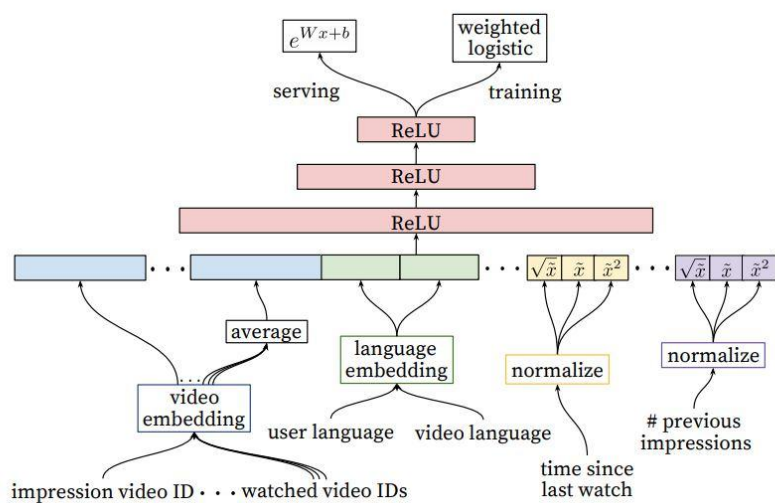


图 5 排序算法架构

排序模型和召回模型结构上非常类似，但在训练目标和输入上稍有不同。输入是用户特征信息以及单个视频的特征，预测目标为期望观看时间，训练方式采用加权逻辑回归，负样本的 label 为 0，正样本的 label 为根据用户观看时间设置权重 weight，当观看时长较长时则 label 值更大，这样就区分了样本的优劣。rank 模型在预测时直接输出的 $e^{(wx+b)}$ ，即期望观看时间。

五. 接口设计

5.1 接口统一说明

5.1.1 统一返回格式

```
{  
    'code' :    ,  
    'message' :    ,  
    'data' :  
}
```

说明：code 为状态码，message 为对应的状态信息，data 为所请求的数据。

5.1.2 状态码封装说明

1. code: 200 , message: “操作成功”；
2. code: 201 , message: “用户名不存在”；
3. code: 202 , message: “密码错误”；
4. code: 203 , message: “邮箱发送失败”；
5. code: 204 , message: “验证码错误”；
6. code: 205 , message: “操作失败”；

5.1.3 动态路由参数说明

1. id: 用户 id;
2. commentId: 评论 id;
3. did: 讨论 id;
4. vid: 视频 id;

5.2 模块接口说明

5.2.1 认证模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|----------------|------------|--------|--------------------------------------|
| /auth/login | 处理用户登录 | Post | Username, password, rememberMe, role |
| /auth/register | 处理用户注册 | Post | Username, password, email |
| /auth/logout | 处理用户登出(注销) | Delete | |

1. 登录接口：

参数说明：username：用户名；password：密码，rememberMe：记住我，一段时间能免密，role：用户身份登陆或者管理员身份登陆。

返回说明：登录是否成功信息

额外说明：登录需要在前端进行相应的表单验证。

2. 注册接口：

参数说明：username：用户名，password：密码，email:邮箱

返回说明：返回是否注册成功。

额外说明：注册需要在前端先进行表单验证，并通过邮箱验证码进行验证，成功才能注册。

3. 登出接口：

参数说明：无

返回说明：返回是否可以登出。

额外说明：登出则清除信息。

5.2.2 用户模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|-----------------|----------|------|-------------------------------|
| /user/{id}/info | 获得用户个人信息 | GET | |
| /user/{id}/info | 修改用户个人信息 | POST | Username, mobile, Email, info |

| | | | |
|--------------------------------|-------------|------|--|
| /user/{id}/resetpwd | 用户修改密码 | POST | Password |
| /user/{id}/viewHistory | 获取用户浏览记录 | GET | |
| /user/{id}/searchHistory | 获取用户搜索记录 | GET | |
| /user/{id}/myActivity | 查看用户参加的捐赠活动 | GET | |
| /user/{id}/myDonate | 查看用户的捐赠记录 | GET | |
| /user/{id}/addCollection/{cid} | 添加用户收藏 | GET | |
| /user/{id}/joinDonate/{id} | 用户参加捐赠活动 | GET | |
| /user/{id}/updateAvatar | 用户修改头像 | POST | avatar |
| /user/{id}/certification | 用户志愿者认证 | POST | name, ID, degree, img1, img2, img3, img4 |
| /user/{id}/viewCourses | 查看视频浏览记录 | GET | |
| /user/{id}/viewTalkings | 查看讨论浏览记录 | GET | |

1. 获取用户个人信息接口：

参数说明：无

返回说明：获取用户个人信息。

额外说明：需要在登陆后才可成功调用该接口。

2. 修改用户个人信息接口：

参数说明：username：用户名，mobile：手机号，email：邮箱，info：公告。

返回说明：返回是否修改成功信息。

额外说明：需要在登陆后才可调用该接口，并且在前端进行相应的表单验证。

3. 修改用户个人密码接口：

参数说明：password：密码

返回说明：返回是否修改密码成功信息。

额外说明：需要在登陆后才可成功调用该接口，并且在前端进行相应的表单验证。

4. 获取用户浏览记录接口：

参数说明：无。

返回说明：返回相应的用户浏览记录信息。

额外说明：需要在登陆后才可成功调用该接口。

5. 获取用户搜索记录接口：

参数说明：无。

返回说明：返回相应的用户搜索记录信息。

额外说明：需要在登陆后才可成功调用该接口。

6. 获取用户参加的捐赠活动接口：

参数说明：无。

返回说明：返回相应的用户所参与的捐赠活动信息。

额外说明：需要在登陆后才可成功调用该接口。

7. 添加用户收藏接口：

参数说明：无。

返回说明：返回是否成功收藏视频信息。

额外说明：需要在登陆后才可成功调用该接口。

8. 参加捐赠活动接口：

参数说明：无。

返回说明：返回是否成功参加捐赠活动信息。

额外说明：需要在登陆后才可成功调用该接口。

9. 修改用户头像接口：

参数说明：avatar：头像。

返回说明：返回是否成功修改头像信息。

额外说明：需要在登陆后才可成功调用该接口。

10. 志愿者认证接口：

参数说明：name:姓名，ID：身份证号，degree：学历，img1：身份证正面照，img2：身份证背面照，3：img3：身份证手持照，4：img4：学历证明照。

返回说明：返回是否成功提交志愿者认证信息。

额外说明：需要在登陆后才可成功调用该接口，并且还未成为志愿者身份。

11. 查看视频浏览记录接口：

参数说明：无。

返回说明：返回视频浏览记录信息。

额外说明：需要在登陆后才可成功调用该接口。

12 . 查看讨论浏览记录接口：

参数说明：无。

返回说明：返回讨论浏览记录信息。

额外说明：需要在登陆后才可成功调用该接口。

5.2.3 通用模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|----------------|-----------|------|-----------------------|
| /common/email | 发送邮箱验证码 | GET | |
| /common/valify | 判断验证码是否正确 | POST | inputCode,correctCode |
| /common/reset | 用户修改密码 | POST | Newpwd,resetKey,email |

1. 发送邮箱验证码接口：

参数说明：无。

返回说明：返回是否成功发送邮箱验证码信息。

额外说明：无

2. 判断验证码接口：

参数说明：inputCode：用户输入验证码，correctCode：正确验证码。

返回说明：返回是否验证成功信息。

额外说明：无。

3. 用户修改密码接口

参数说明：Newpwd：新密码，resetKey：密钥，email：邮箱。

返回说明：返回是否修改密码成功信息。

额外说明：无。

5.2.4 讨论模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|------------------|--------|------|---|
| /talkings/all | 获取讨论项 | POST | Query, sort, typeKey |
| /talkings/types | 获取讨论分类 | GET | |
| /talkings/upload | 讨论封面上传 | POST | File |
| /talkings/send | 添加讨论 | POST | Title, content, face , type, from_user |

1. 获取讨论项接口：

参数说明：query：查询关键字，sort：排序，typeKey：类型。

返回信息：返回讨论项信息。

额外说明：无。

2. 获取讨论分类接口：

参数说明：无。

返回信息：返回讨论分类信息。

额外说明：无。

3. 讨论封面上传接口：

参数说明：File：封面。

返回信息：返回是否上传封面成功信息。

额外说明：无。

4. 添加讨论接口：

参数说明：title：标题，content：内容，face：封面 url，type：类型，from_user：用户。

返回信息：返回是否成功发布信息。

额外说明：无。

5.2.5 讨论详情模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|-----------------------------|---------------|------|--|
| /detail/{did} | 某个讨论的详情信息 | GET | |
| /detail/replies/{commentId} | 获取某条讨论评论的回复信息 | GET | |
| /detail/comment/add | 添加评论(回复) | POST | Content, from_user, from_post, reply_id, comment_id, Publish_time |

1. 获取某个讨论的详情信息接口：

参数说明：无。

返回信息：返回某个讨论所有的信息。

额外说明：无。

2. 获取某条讨论回复信息：

参数说明：无。

返回信息：返回某条讨论的回复信息。

额外说明：无。

3. 添加评论(回复)接口：

参数说明：Content：内容，from_user：用户，from_post：讨论帖，reply_id：

回复人 id，comment_id：回复属于的最上级的评论 id，Publish_time：添加日期。

返回信息：返回是否添加讨论(回复)成功信息。

额外说明：无。

5.2.6 视频首页模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|------------------------|--------|------|--|
| /course/data | 首页视频信息 | GET | |
| /course/recommend/{id} | 首页推荐视频 | GET | |
| /course/types | 视频分类 | GET | |
| /course/upload | 上传视频 | POST | File |
| /course/add | 添加视频项 | POST | Filename, info, Title, from_user_id |
| /course/carousel | 首页轮播图 | GET | |
| /course/all | 获取视频项 | POST | Sort, query, typeKey |

1. 首页视频信息接口：

参数说明：无。

返回信息：返回所有首页视频信息。

额外说明：无。

2. 首页推荐视频接口：

参数说明：无。

返回信息：返回首页推荐视频信息。

额外说明：无。

3. 视频分类接口：

参数说明：无。

返回信息：返回视频分类信息。

额外说明：无。

4. 上传视频接口：

参数说明：File：视频。

返回信息：返回是否成功上传视频信息。

额外说明：无。

5. 添加视频接口：

参数说明：filename：视频名称，info：简介，title：视频标题，from_user_id：

用户 id

返回信息：返回是否添加视频到后台审核信息。

额外说明：是否已通过志愿者认证。

6. 首页轮播图接口：

参数说明：无。

返回信息：返回轮播图信息。

额外说明：无。

7. 获取视频项接口：

参数说明：query：查询关键字，sort：排序，typeKey：类型。

返回信息：返回筛选好的视频项信息。

额外说明：无。

5.2.7 视频详情模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|----------------------------|---------------|------|--|
| /video/detail/{vid} | 某个视频的信息 | GET | |
| /video/updateView/{vid} | 更新视频播放量 | GET | |
| /video/replies/{commentId} | 获取某条视频评论的回复信息 | GET | |
| /video/comment/add | 添加评论(回复) | POST | Content, from_user, from_course, reply_id, comment_id, Publish_time |

1. 获取某个视频信息接口：

参数说明：无。

返回信息：返回某个视频所对应的所有信息。

额外说明：无。

2. 更新视频播放量接口：

参数说明：无。

返回信息：返回是否更新成功信息。

额外说明：无。

3. 获取视频下的评论回复信息接口：

参数说明：无。

返回信息：返回某条讨论的回复信息。

额外说明：无。

4. 添加评论(回复)接口：

参数说明：Content：内容，from_user：用户，from_post：讨论帖，reply_id：

回复人 id，comment_id：回复属于的最上级的评论 id，Publish_time：添加日期。

返回信息：返回是否添加讨论(回复)成功信息。

额外说明：无。

5.2.8 捐赠模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|---------------------|-------------|------|---------------------------------------|
| /donate/all | 根据条件获取捐赠活动项 | POST | Sort, typeKey, query, status |
| /donate/update/{id} | 添加捐赠记录 | POST | From_user_id, donate_num, Donate_time |
| /donate/detail/{id} | 获取某个捐赠活动的详情 | GET | |

1. 获取捐赠活动项接口：

参数说明：sort：排序，typeKey：类型，query：关键字，status：状态

返回信息：返回所对应的捐赠活动项信息。

额外说明：无。

2. 添加捐赠记录接口：

参数说明：from_user_id：用户 id，donate_num：捐赠数，donate_time：捐赠日期。

返回信息：返回是否成功添加记录信息。

额外说明：无。

3. 获取捐赠活动详情接口：

参数说明：无。

返回信息：返回对应活动详细信息。

额外说明：无。

5.2.9 日志管理模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|--------------------|----------|--------|--------------------------------|
| /admin/logs/all | 根据条件筛选日志 | POST | Sort, query, type dateScope |
| /admin/logs/delete | 删除日志 | DELETE | |

1. 获取日志信息接口：

参数说明：Sort：排序，query：查询字，type：类型。

返回信息：返回所筛选出来的管理员操作日志信息。

额外说明：无。

2. 删除日志接口：

参数说明：无。

返回信息：返回是否删除成功信息。

额外说明：无。

5.2.10 视频管理模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|--------|------|------|-----|
|--------|------|------|-----|

| | | | |
|---------------------------|----------|--------|--------------------------|
| /admin/course/all | 根据条件筛选视频 | POST | Sort, type, state, query |
| /admin/course/{id}/pass | 审核通过某个视频 | GET | |
| /admin/course/{id}/delete | 删除某个视频 | DELETE | |

1. 展示所有视频信息接口：

参数说明：sort：排序，type：类型，state：状态，query：查询字

返回信息：返回所有筛选出的视频信息。

额外说明：无。

2. 审核视频接口：

参数说明：无。

返回信息：返回是否成功通过该视频。

额外说明：无。

3. 删除视频接口：

参数说明：无。

返回信息：返回是否成功删除视频信息。

额外说明：无。

5.2.11 用户管理模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|---------------------------|--------|--------|--------------------------------------|
| /admin/user/all | 获取所有用户 | POST | Sort, query |
| /admin/user/{id}/role | 修改用户角色 | POST | Role, from_user_id |
| /admin/user/{id}/delete | 删除用户 | DELETE | |
| /admin/user/add | 添加用户 | POST | Username, password, email Mobile, |
| /admin/user/certification | 审核志愿者 | GET | |

| | | | |
|--|----|--|--|
| | 身份 | | |
|--|----|--|--|

1. 获取所有用户接口：

参数说明：sort：排序，query：查询字。

返回信息：返回经过筛选的所有用户信息。

额外说明：无。

2. 修改用户角色接口：

参数说明：Role：角色，from_user_id：被修改用户 id。

返回信息：返回是否成功修改用户角色信息。

额外说明：无。

3. 删除用户接口：

参数说明：无。

返回信息：返回是否成功删除用户信息。

额外说明：无。

4. 添加用户接口：

参数说明：username：用户名，password：密码，email：邮箱，mobile：手机号。

返回信息：返回是否成功添加用户信息。

额外说明：无。

5. 审核志愿者身份接口：

参数说明：无。

返回信息：返回是否成功审核该身份信息。

额外说明：无。

5.2.12 捐赠活动管理模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|-------------------|----------------|------|-------------------|
| /admin/donate/all | 获取所有捐赠 活动详情 | POST | Sort, type, query |

| | | | |
|---------------------------|--------|--------|--|
| /admin/donate/{id}/delete | 删除某个活动 | DELETE | |
| /admin/donate/add | 发起捐赠活动 | POST | From_user_id, content, dname, type, starttime, Endtime, location |

1. 获取捐赠活动详情接口：

参数说明：sort：排序，type：类型，query：关键字，status：状态

返回信息：返回所对应的捐赠活动详情信息。

额外说明：无。

2. 删除某个活动接口：

参数说明：无。

返回信息：返回是否成功删除某个活动信息。

额外说明：无。

3. 发出捐赠活动接口：

参数说明：from_user_id：用户 id，content：简介，dname：捐赠活动名称，

type：类型，starttime：开始时间，endtime：结束时间，location：捐赠地。

返回信息：返回是否成功提交捐赠活动信息。

额外说明：无。

5.2.13 讨论管理模块

| 接口 url | 接口功能 | 请求方法 | 请求体 |
|---|----------|--------|-------------------|
| /admin/discussion/all | 获取所有讨论详情 | POST | Sort, type, query |
| /admin/discussion/{id}/deleteDiscussion | 删除某个讨论 | DELETE | |
| /admin/discussion/{id}/deleteComment | 删除某个评论 | DELETE | |

1. 获取所有讨论详情接口：

参数说明：sort： 排序， type： 类型， query： 关键字

返回信息： 返回所对应的讨论详情信息。

额外说明： 无。

2. 删除讨论接口：

参数说明： 无。

返回信息： 返回是否成功删除某个讨论信息。

额外说明： 无。

3. 删除某个活动接口：

参数说明： 无。

返回信息： 返回是否成功删除某个评论信息。

额外说明： 无。