



项目文档

🎯 纸上得来终觉浅，绝知此事要躬行。

一、项目介绍

项目背景

- 本项目采用 前后端 分离的方式开发七牛云1024]创作节项目网页短视频应用
- 致力于做一个自己认为有价值的作品，不仅是结果还有过程，学习到知识就好

项目分工

团队成员	方向	项目角色与负责事项
许贵华	前端	团队进度把控、UI 产品图主设计、前端项目开发以及静态部署
马秋林	后端	数据库设计、负责个人中心模块、互动接口开发
朱泽胜	后端	后端开发、框架搭建、负责视频模块

二、项目目标

（项目目标指在项目规划之初设定的项目总体目标。项目目标应该是在清晰定义的、可实现的且可测量的项目成功标准）

使用七牛云存储、七牛视频相关产品（如视频截帧等）开发一款Web端短视频应用。该应用主要实现功能有：视频播放，内容分类，用户注册、登录，点赞、分享视频，管理员管理视频等功能。

目标一（基础功能）

- 1. 视频播放：播放、暂停、进度条拖拽
- 2. 内容分类：视频内容分类页，如热门视频、体育频道
- 3. 视频切换：可通过上下键翻看视频

目标二（高级功能）

- 1. 账户系统：用户可登录，收藏视频
- 2. 推荐机制：自动为用户推荐更多兴趣频道
- 3. 互动功能：点赞、分享、关注、搜索等
- 4. 上传发布视频
- 5. 更多能够提升用户体验的功能

四、详细设计

4.1 技术选型

前端

Vue3+Vite+Pinia+Vue-router：使用 vue3 的生态实现 Web 单页面开发，与后端通过 接口 进行交互。

Antd-V：选择 antd-v 作为UI组件库，是因为完成本项目组件比较基础，elm、vant 这些 的也比较多就想换一个体验一下，为下一步学习 react 打基础。

Eslint+Prettier：使用熟悉的这两个库进行代码格式校验以及格式化，为后续维护扩展奠定基础。

Less+Axios+QS：使用 Less 预处理器提高 css 编写效率。Axios 和 QS 为接口交互提供服务。

后端

SpringBoot：方便项目的配置，使开发人员可以专注业务开发。

MyBatisPlus：持久层框架，简化了SQL的编写，提高开发效率。

MySQL：市面上用户群体较大的关系型数据库，保证稳定性。用于存储开发所需的数据。

七牛云：使用七牛云提供的API进行对象存储，主要存储图片与视频资源。

4.2 架构设计

4.2.1 整体架构

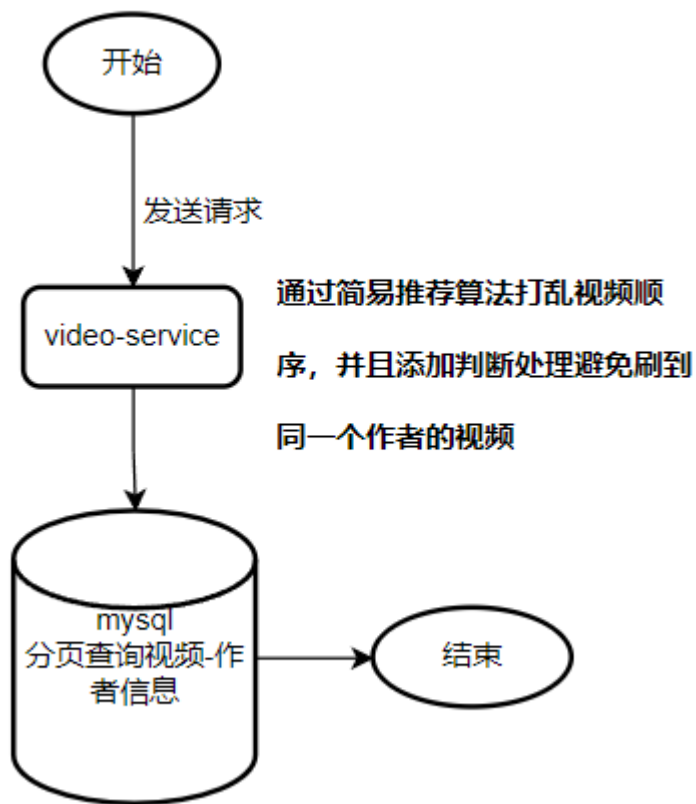
前端目录

- 1 src 下目录结构
- 2 |—assets # 静态文件
- 3 |—components # 个人组件库
- 4 | |—layout # 布局组件

```
5 | | | └─Content
6 | | |   └─Header
7 | | └─GhDialog # 弹窗组件
8 | | └─GhVideo # 视频组件
9 | |   └─GhVideoControls # 视频的控制组件
10 | └─hooks # 前端 hooks
11 | | └─controls
12 | | └─form
13 | |   └─upload
14 | └─router # 前端路由
15 | |   └─index
16 | └─stores # 前端仓库
17 | |   └─bussiness # 业务仓库
18 | └─tools # 常用工具
19 | | └─formatTime
20 | | └─axios
21 | |   └─request
22 | └─views # 前端页面
23 | |   └─HomeView # 首页
24 | |   └─PersonViiew # 个人中心页面
25 | |   └─UploadView # 上传页面
```

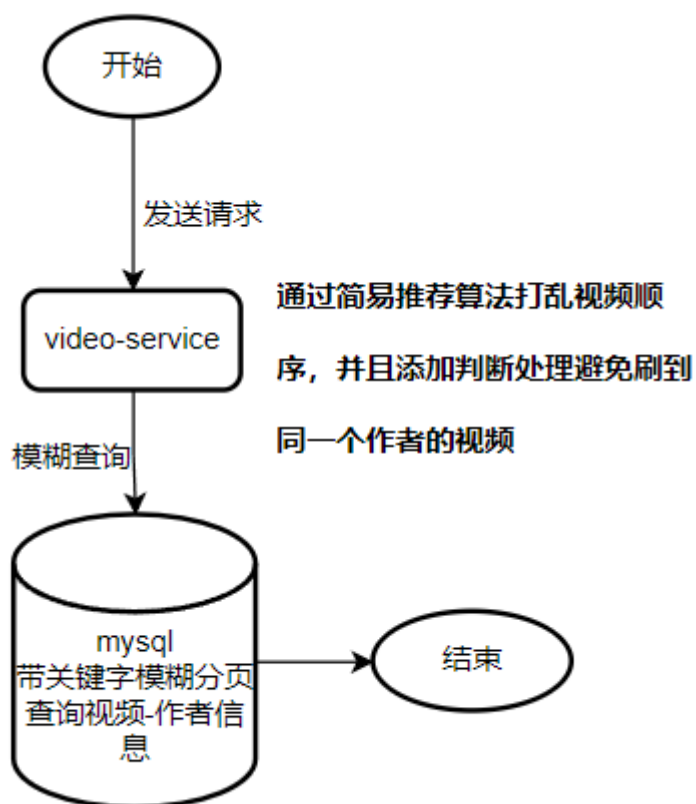
4.2.4 接口设计

4.2.4.1 推荐视频



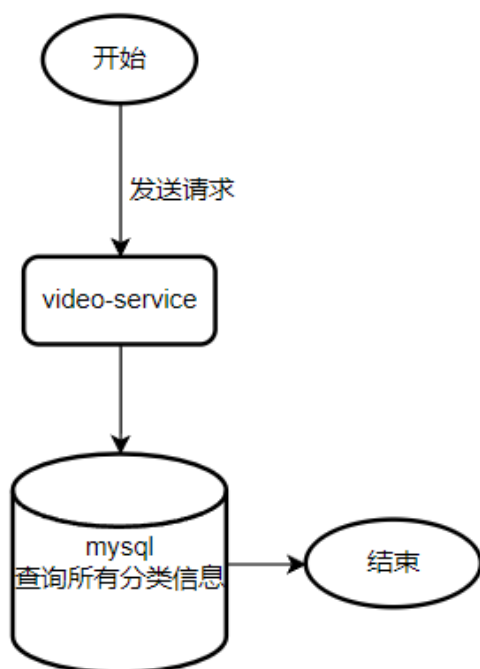
- 客户端发送请求获取视频列表
- 服务端进行数据库分页查询，获取分页结果并进行推荐算法处理后返回视频列表

4.2.4.2 搜索视频



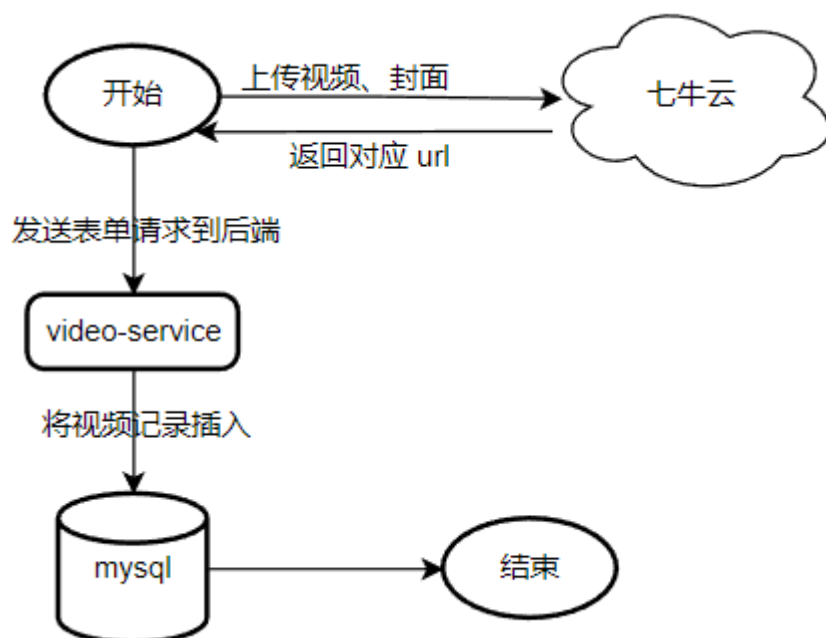
- 客户端在视频页搜索框中输入关键字，点击搜索后发送请求
- 服务端进行数据库带关键字的模糊分页查询，获取分页结果后进行推荐算法处理后返回视频列表

4.2.4.3 视频分类



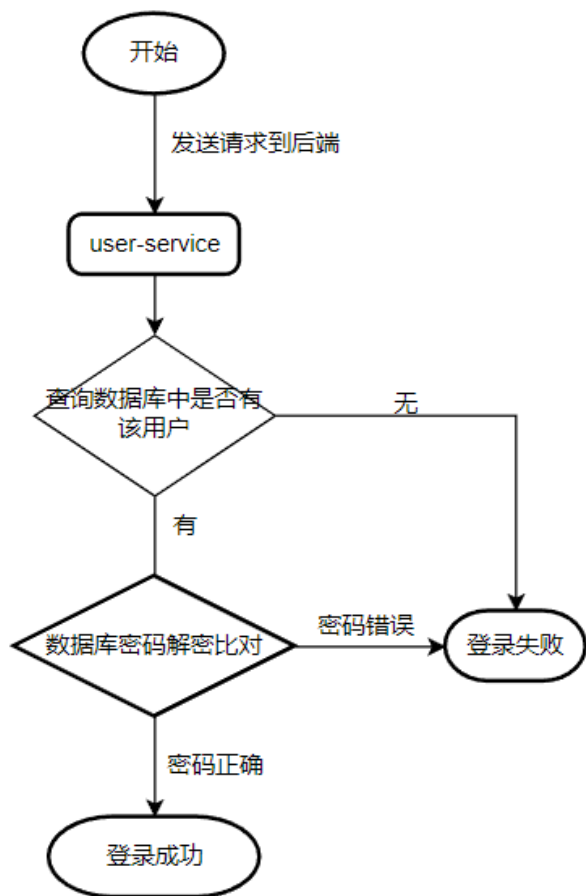
- 发送请求可获取所有视频分类

4.2.4.4 上传视频



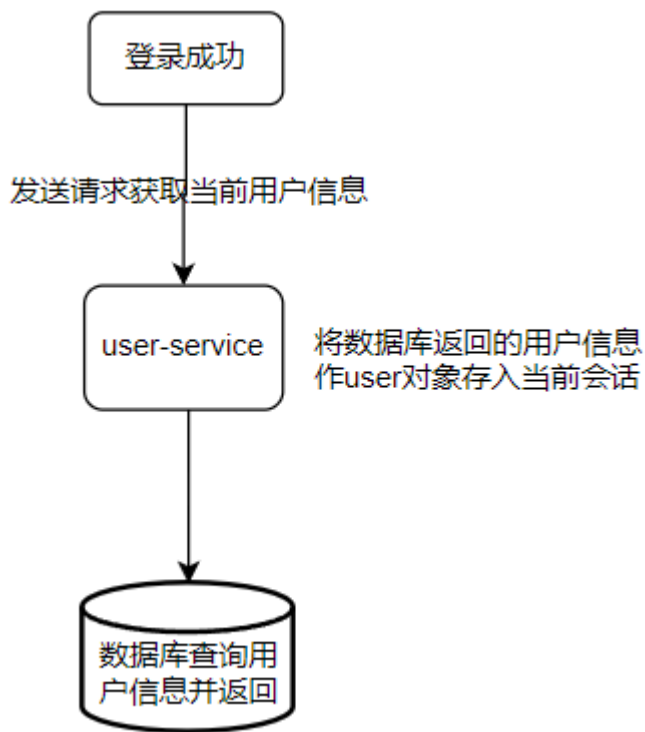
- 前端上传视频、封面至七牛云，将信息填写至请求表单
- 服务端从当前会话中获取用户基本信息并填充
- 将视频记录插入MySQL数据库

4.2.4.5 用户登录



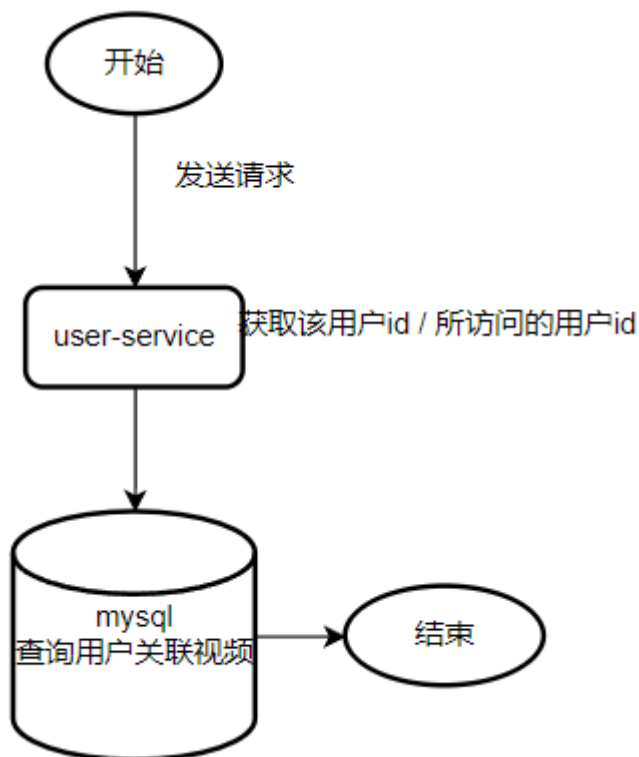
- 用户在登录页输入用户名和密码点击登录
- 服务器接收到请求后查询数据库有无该用户，没有则登录失败
- 有则将数据库用户密码解密后进行比对，比对成功则登录成功，比对失败则登录失败

4.2.4.6 用户信息



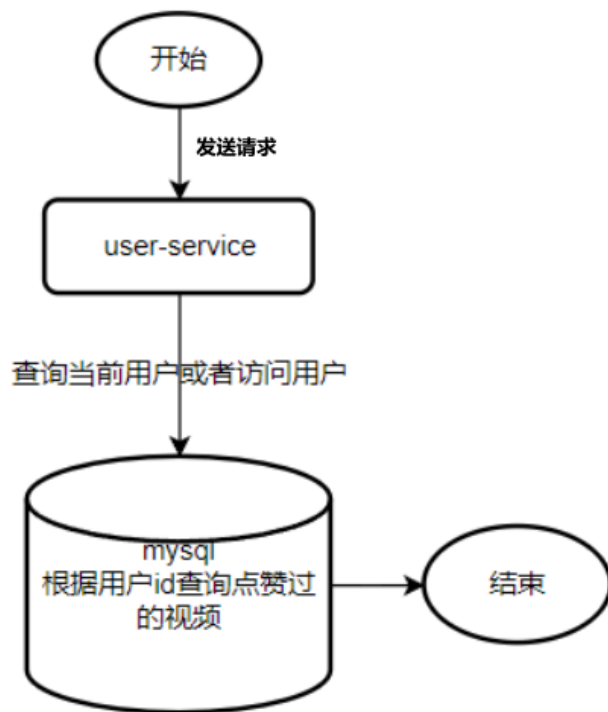
- 登录成功后发送请求获取当前用户信息
- 服务器查询数据库将信息存入当前会话中

4.2.4.7 用户作品



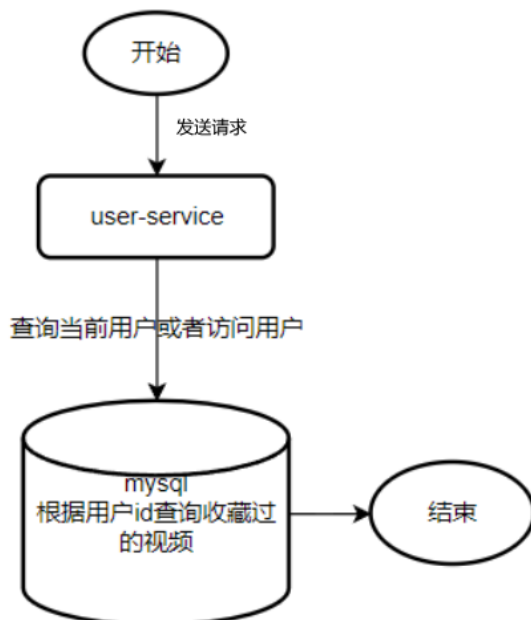
- 进入个人中心后，发送请求获取该用户的作品列表
- 服务器根据用户会话中用户id查询数据库中该用户关联的所有视频并返回

4.2.4.8 用户点赞的作品



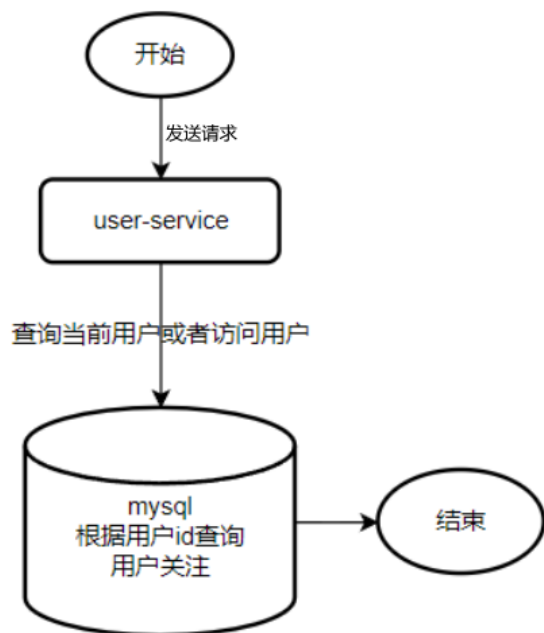
- 客户端发送查询请求
- 服务端根据参数设置查询的参数，查询数据库记录并返回

4.2.4.9 用户收藏的作品



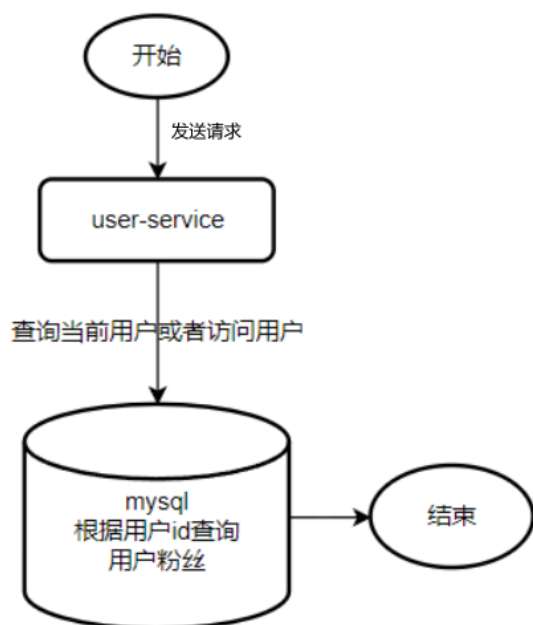
- 客户端发送查询请求
- 服务端根据参数设置查询的参数，查询数据库记录并返回

4.2.4.10 用户关注列表



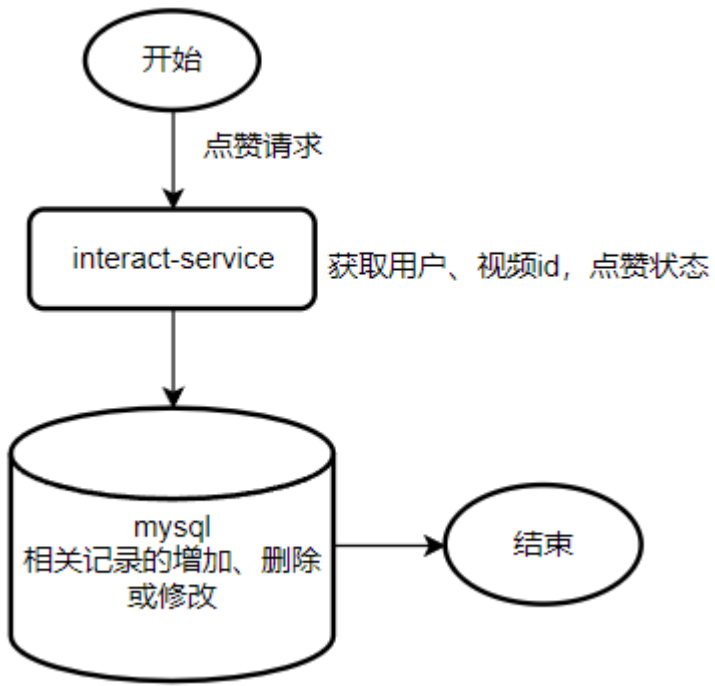
- 发送请求可获取用户关注列表

4.2.4.11 用户粉丝列表



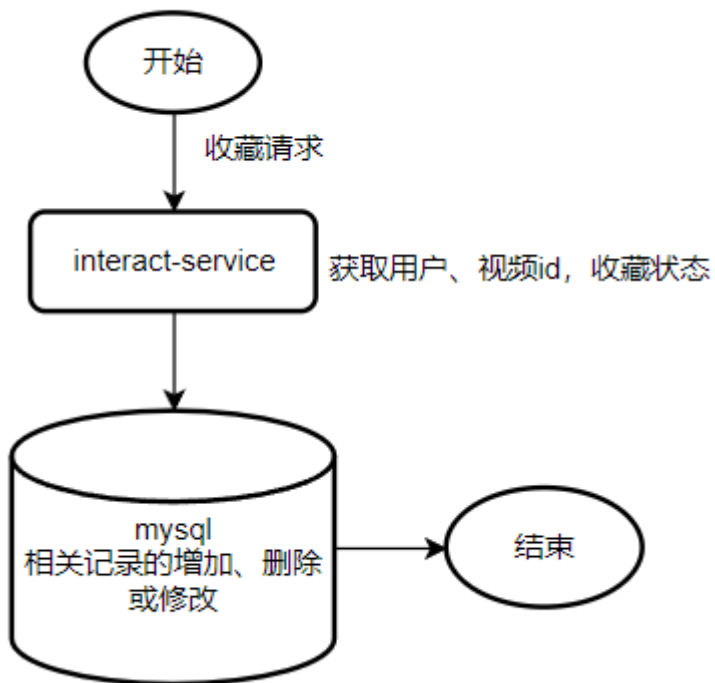
- 发送请求可获取用户粉丝列表

4.2.4.12 点赞视频



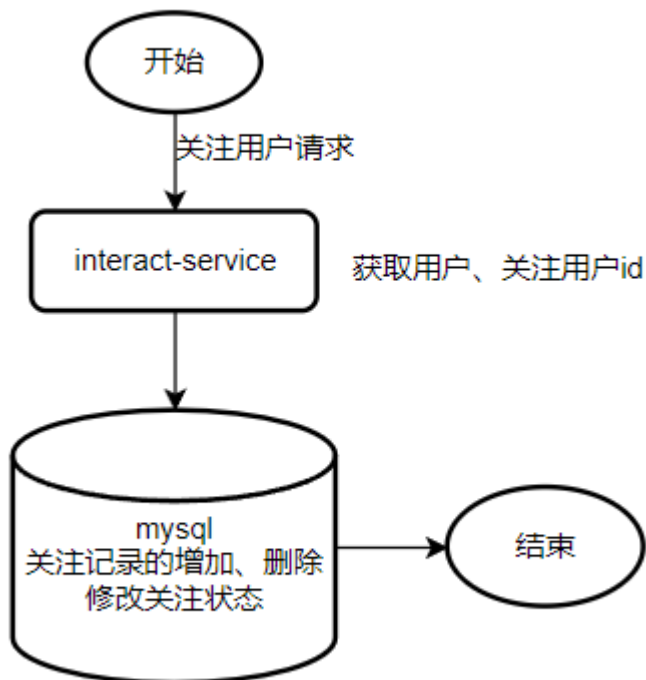
- 用户登录后，可以进行点赞操作，发送请求给服务端
- 服务端根据请求内容对数据库进行增加/删除用户点赞记录
- 同步更改操作视频的点赞数

4.2.4.13 收藏视频



- 用户登录后，可以进行收藏操作，发送请求给服务端
- 服务端根据请求内容对数据库进行增加/删除用户收藏记录
- 同步更改操作视频的收藏数

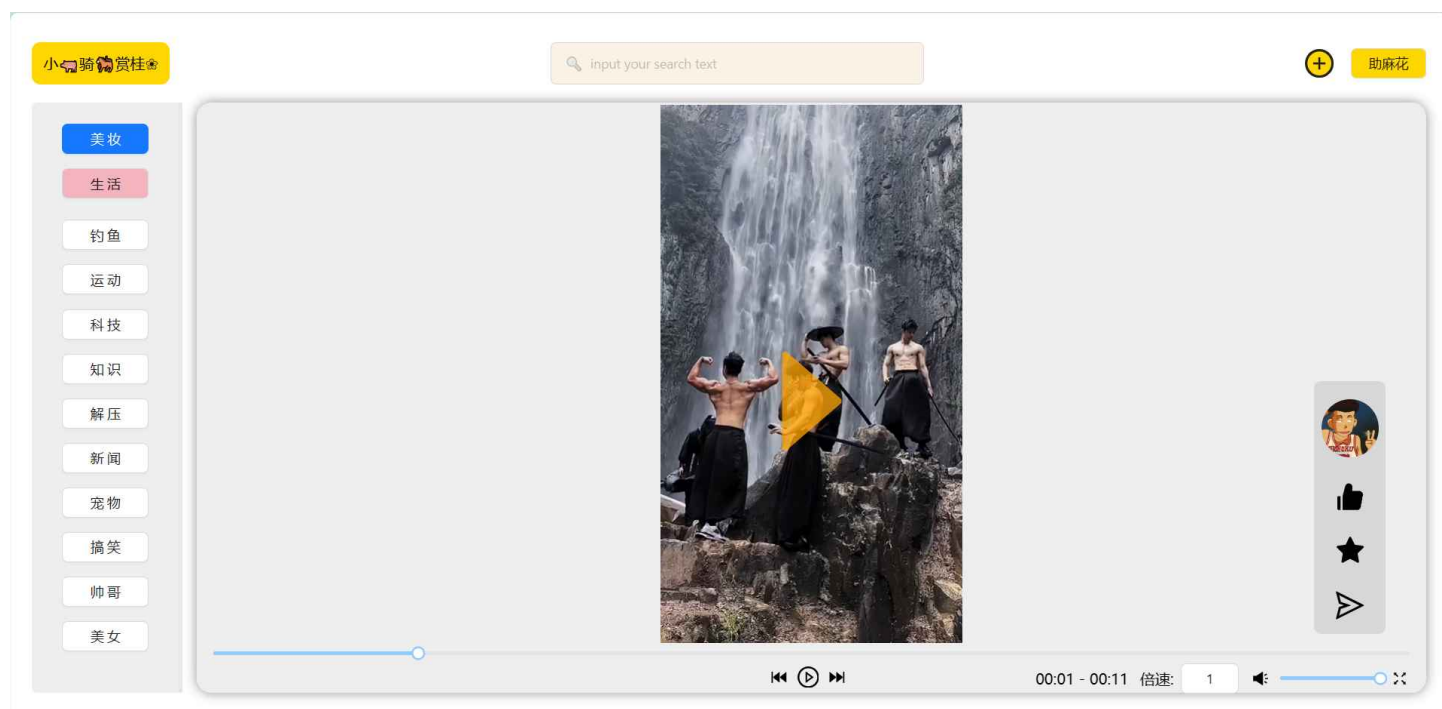
4.2.4.14 关注用户



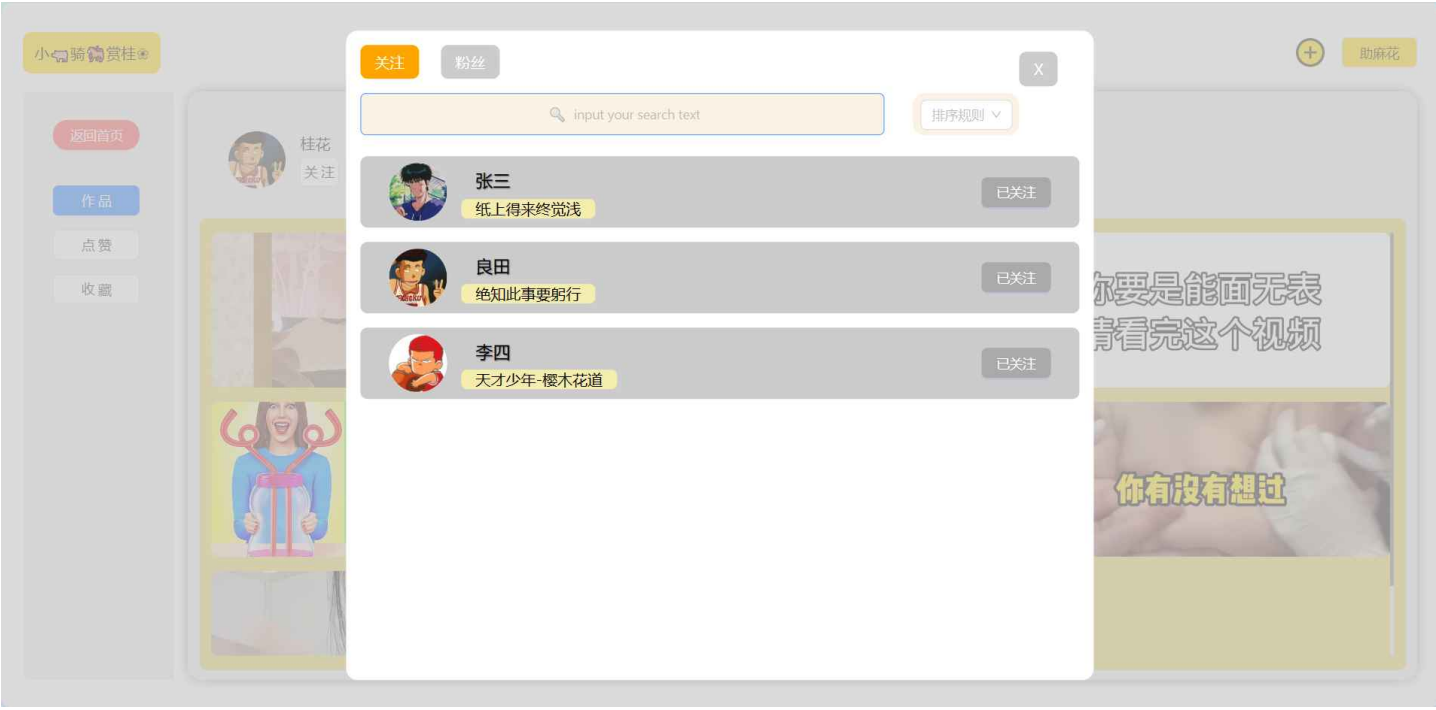
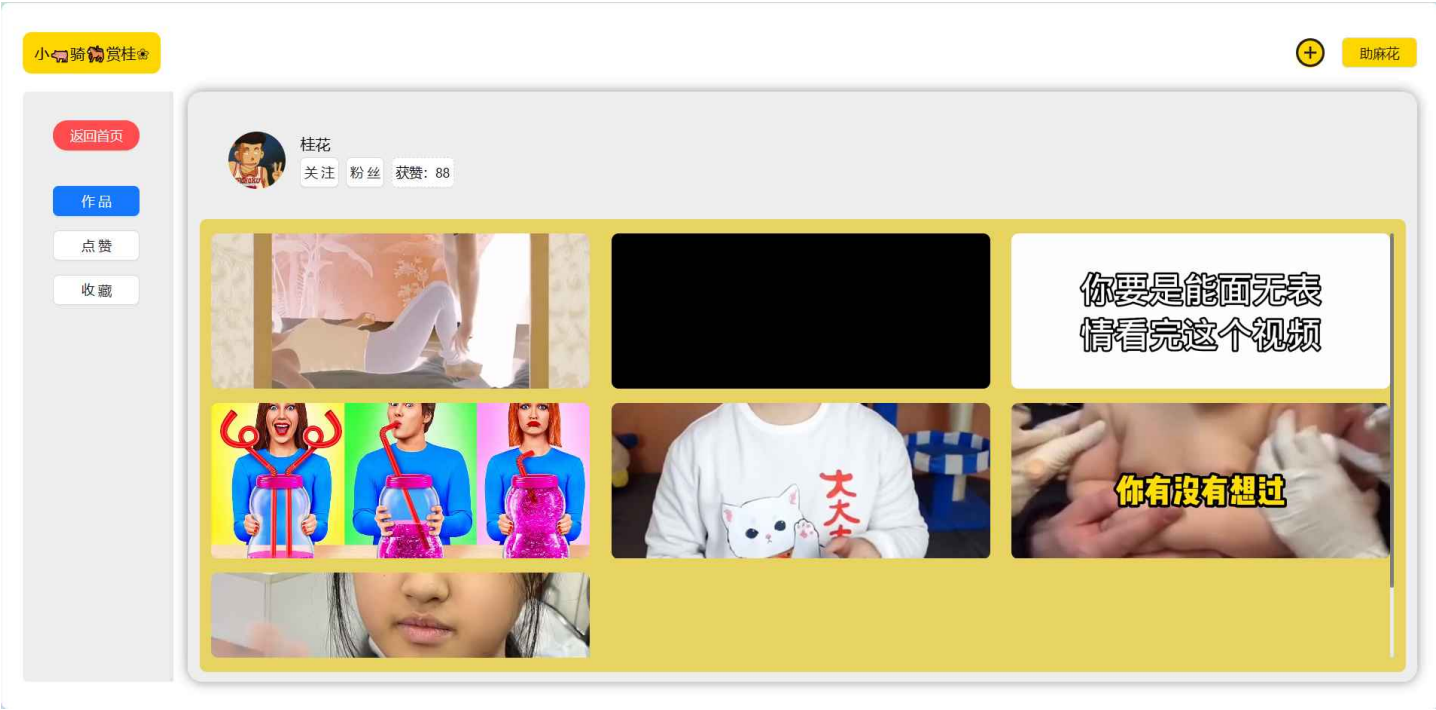
- 用户登录后，可以进行关注操作，发送请求给服务端
- 服务端根据请求内容对数据库进行增加/删除用户关注记录
- 更改用户间关注状态

4.3 前端项目

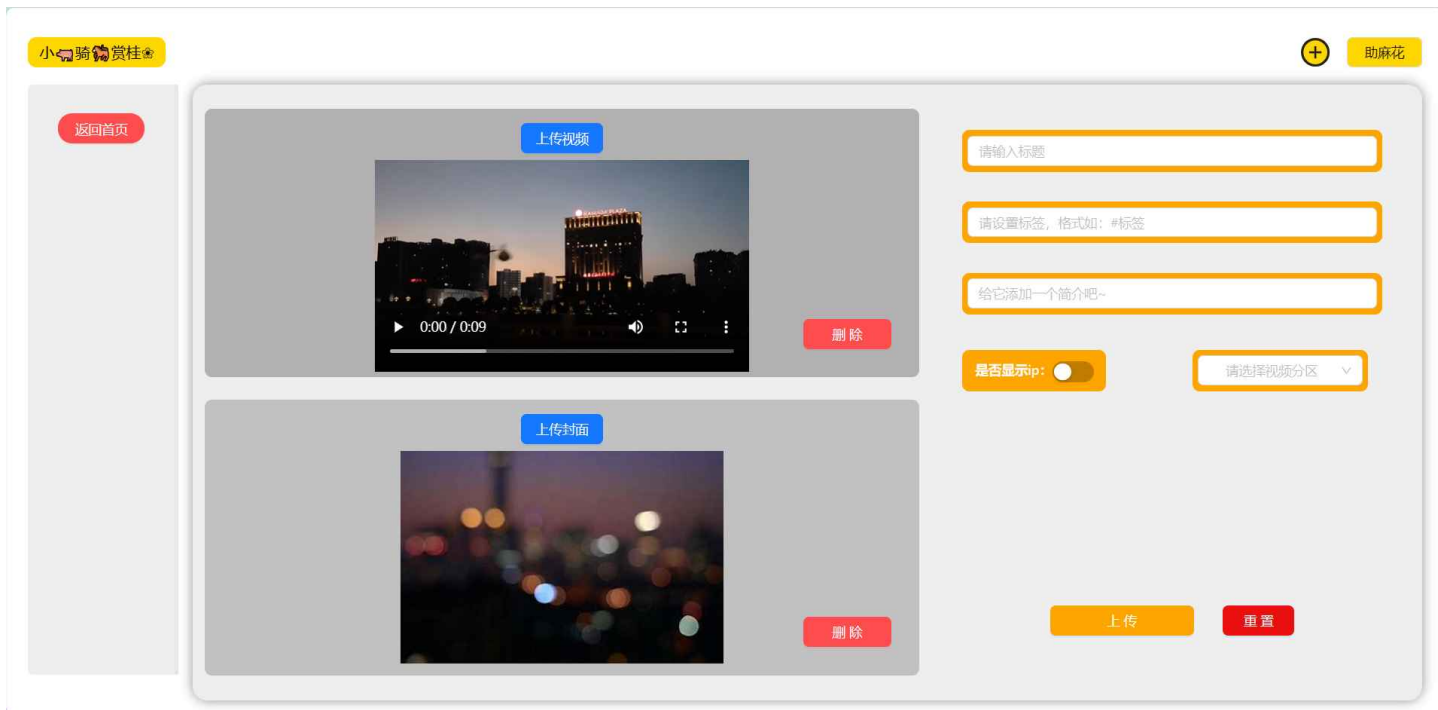
- 首页



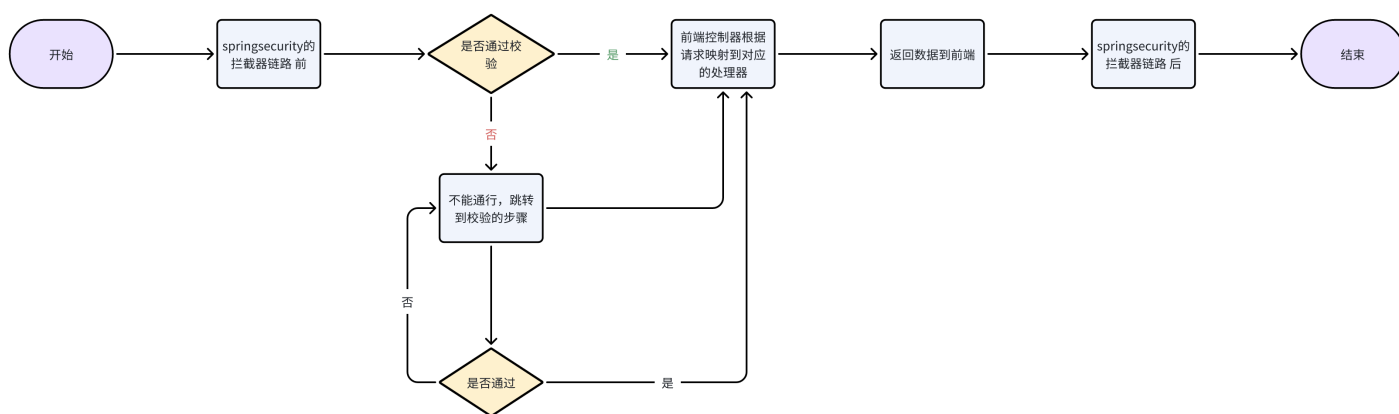
• 个人中心



• 文件上传以及预览



4.4 后端项目



4.5 技术难点

1. 视频推荐算法（未完整实现）：需要保证同一个用户的视频不会被连续推荐，重写打开应用获取推荐视频时不会刷到已经刷到的视频。
2. 已登录用户的情况下，在观看视频时，需要同步获取本条视频及作者与该观看用户的关注、点赞、收藏状态比较麻烦，可以在原来传输的UserVideoDTO实体中添加一个属性记录用户关注、点赞、收藏的状态。
3. 前端结合七牛云定制化实现文件上传、预览、视频播放控制器等功能。
4. 前端用户个人中心中视频的展示呈现为三栏布局，结合后端视频列表进行排列展示并且保证样式不被破坏。

五、项目会议

第一次会议：2023年10月26日

会议记录

- 分析题目要求，了解比赛评分要求等相关规则
- 具体做什么？包含哪些功能？难点大概是什么？
 - 先完成大赛要求的基本功能，第二阶段再进行优化
- 回顾去年七牛云的竞赛结果
 - 在B站上有视频，可以看看大致要做到什么程度
- 协作文档、会议频率、产品原型图
 - 使用飞书协作
 - 使用 即时设计 完成产品原型图
 - 会议每个阶段进行一次开会，同步进度同时也做下一步计划
- 项目分工
 - 先去了解去年大赛相关资料，同时创建仓库
 - 了解七牛云的使用
 - 把原型图在即时设计中画出来
 - 在飞书中把基础的文档写出来
 - 了解抖音、快手、小红书等产品
- 下一次会议
 - 在完成原型图后进行下一步规划，大概两天时间

第二次会议：2023年10月28日

会议记录

- 原型图进一步细节敲定
- 接口文档
 - 接口文档使用 yapi 管理，以及飞书管理相关思考和文档
- 分工
 - 前端开始进行页面开发以及简单交互
 - 后端数据库字段根据 原型图 进行设计
 - 后端搭建安全层、接口层

第三次会议：2023年11月3日

会议记录

- 进度汇报
 - 前端页面基本完成，可以开始对接接口
 - 泽胜把首页接口写好了，正在尝试做推荐算法，记录 权限 以及用户喜爱度
 - 秋林把个人中心的接口写出来一大半，并且完成了安全层的接口，只要部署后端项目就能对接
- 遇到的问题
 - 后端协作时，数据库来回传输并且修改不同步
 - 解决方法是将数据库上云，本地修改同步远程
 - 前端有的交互需要定制化，用时较长，例如视频的控制器、文件上传的模板、关注收藏按钮样式、弹窗的状态管理等
 - 解决方法是先将已完成的内容进行对接，非对接时间继续开发其他自制的功能与样式
 - 学校事情以及个人突发事情（成员受伤），开发时间被压缩
 - 固定的开发时间开发项目，经常被别的事打断没办法只能尽量保证进度
 - 如果是开发状态的问题，可以尝试换个工作环境，在图书馆或者远离房间等尽量高能量的地方
- 下一步计划
 - 抓紧开发，尽量快些完成接口对接，留一点时间出来写文档和视频录制

第四次会议：2023年11月6日（晚）

会议记录

- 进度汇报
 - 前端页面所有都已完成，功能逻辑也都以及实现，视频切换以及功能控制核心业务已完成，接口也已成功对接
 - 前端文件上传对接七牛云已成功，并且拿到 token 进行上传和预览，上传逻辑已实现，对接接口上传表单即可
 - 泽胜优化了主页接口，数据库表等，推荐算法因进度问题被砍了
 - 秋林把接口文档完善好了，接口开发完毕，并且将一些数据渲染到了数据库中
- 遇到的问题
 - 发现一些接口被遗漏
 - 及时补一下相关不影响的接口，太大的逻辑就不弄了，保证系统正常运行
 - 部署后端项目的阿里云服务器在 6 号晚上过期了，接口服务访问不了
 - 服务器白嫖的现在续费要130，我们项目也在修修补补也不太觉得能拿奖，觉得不值就只能新用户领了一个月服务器，重新把项目部署上去

- 服务器容量有限 mysql 安装受限制，只能争取成功部署前端完成正常的接口对接以及未对接的接口对接
- 竞赛材料准备
 - 文档明天也要着手写，把之前写的散落的文档整合起来写到一个文档中，先在飞书中一起编辑，完成后导出 PDF 上传到 github 中
 - 视频的录制，保证接口完成对接后，录制一个简单的 demo 视频打包上传到 github 中
 - 仓库记得开源，并且十二点后不要再推送代码
- 开干！

六、演示视频

- Github地址：<https://github.com/zhuzuxi/ShortVideo/tree/front/teamDoc>

七、项目总结

- 这次的团队比赛，我们收获颇深！我们团队的凝聚力通过这次比赛得以提升。在刚开始组队时，我们对于项目如何协调以及任务如何分配都不是十分的清楚，通过组织了多次的团队会议，共同确定了项目的总体需求，明确了具体的需求之后，我们的分工也变得明确。在之后的研发开发过程中，我们遇到困难时，并没有就此放弃，而是团队成员相互鼓励，互帮互助，最后共同地解决了问题，我们真正成为了一个默契的团队，这是一份难得的收获。
- 同时，我们的团队在这次比赛中也变得更加强大。无论是刚开始的调研，还是之后每一次的沟通交流，以及每一次的分工协作，我们都各尽所长，将自己的优势发挥到最大化；在每一次遇到困难时，我们都努力地去克服，通过团队的共同努力，攻克了难题。毋庸置疑的是，我们团队的每一个成员在这次比赛中都取得了巨大的进步。
- 任务分配时间的安排不够周全也是一次很深刻的教训。由于没有预料到突发事故，后期对接接口的时间比较仓促。下次进行项目开发除了每天的工作量考虑，还得给未知的事留下足够的时间，避免再次产生同样的问题。
- 后端两次数据库数据出差错，让我意识到数据库备份的重要性，无论是在云上还是本地都要做好备份，在不可避免情况下，是快速恢复生产开发的关键。
- 第一次进行线上团队协同开发，通过各种协同工具，多沟通交流，及时同步开发，保证设计思路的一致、统一，能够避免很多不必要的工作量。
- 项目的文档过程很重要，尤其是在设计阶段，在设计时写的文档能够记录当时的思路，并且在开发过程中能够加快实现效率。
- 在使用云服务器进行数据库和项目的部署时，并没有想象中那么难，只是一些命令的输入和应用的安装过程，不要因为没使用过就觉得用不来。
- 描述问题的能力也很重要，通过博客文章对技术点进行快速学习或回顾以及使用AI可以让开发效率更上一层楼。

和你们一起开发非常开心以及难得，未来希望各位前程似锦，都找到好工作！

也非常感谢你们的支持，让这个项目最终实现了出来，不说非常好，但这也是我的一段美好回忆！

未来江湖间，火车是向前开的！