

凌众视频-架构设计文档

一、项目技术栈选择

1. 前端
2. 后端

二、重要接口总结

1. 登录授权模块
 - 1.1. 验证码登录
 - 1.2. Token刷新
2. 关注和取消关注
3. 视频发布接口
4. 视频推荐接口设计
 - 4.1. 未登录时视频推荐
 - 4.2. 登录后视频推荐
5. 视频搜索接口
 - 5.1. 视频数据同步
 - 5.2. 高亮搜索
6. 点赞收藏模块
 - 6.1. 点赞视频
 - 6.2. 取消点赞
 - 6.3. 收藏视频
 - 6.4. 取消收藏
 - 6.5. 获取当前用户点赞过的视频
 - 6.6. 获取当前用收藏过的视频
 - 6.7. 查看用户是否喜欢或收藏过该视频
7. 视频评论模块
 - 7.1. 发布评论或回复
 - 7.2. 获得视频下所有评论及回复
 - 7.3. 获得用户发表的评论
 - 7.4. 点赞评论

- 7.5. 取消点赞评论
- 7.6. 判断是否是该用户的评论
- 7.7. 删除评论
- 7.8. 获取该视频下此用户点赞过的评论
- 7.9. 判断用户是否点赞过该评论
- 7.10. 通过评论id查看评论详细内容

8. 消息通知模块

- 8.1. 获取视频点赞消息
- 8.2. 获取视频收藏消息
- 8.3. 获取评论点赞消息
- 8.4. 获取被回复的消息
- 8.5. 获取视频被评论的消息

三、人员安排

一、项目技术栈选择

1. 前端

技术栈：

- 1. 框架：Vue3
- 2. 脚手架：Vite
- 3. UI组件库：ElementUI Plus
- 4. ajax请求：axios
- 5. 项目状态管理：pinia
- 6. css预处理器：scss
- 7. 库管理器：npm

2. 后端

- 1. SpringBoot：项目开发框架
- 2. SpringSecurity OAuth2：使用密码模式，重写了登录逻辑，使用邮箱验证码方式登录，并通过Redis进行token的刷新操作。
- 3. Mybatis-Plus：项目持久层框架，单表查询时可直接使用Api进行操作简单方便，连表查询时支持

xml格式书写sql扩展性较强。

4. Elasticsearch：担任项目视频搜索引擎，使用ik分词器对视频描述，视频用户名等进行分词，并使用定时任务定期更新索引库。
5. Redis：用于项目验证码存储，token的刷新，在项目的视频推荐功能中起到了不可替代的作用。
6. 七牛云：用于项目的视频存储和用户头像存储。
7. Swagger：用于项目Api文档生成，为前后端交互带来了极大的便利。

二、重要接口总结

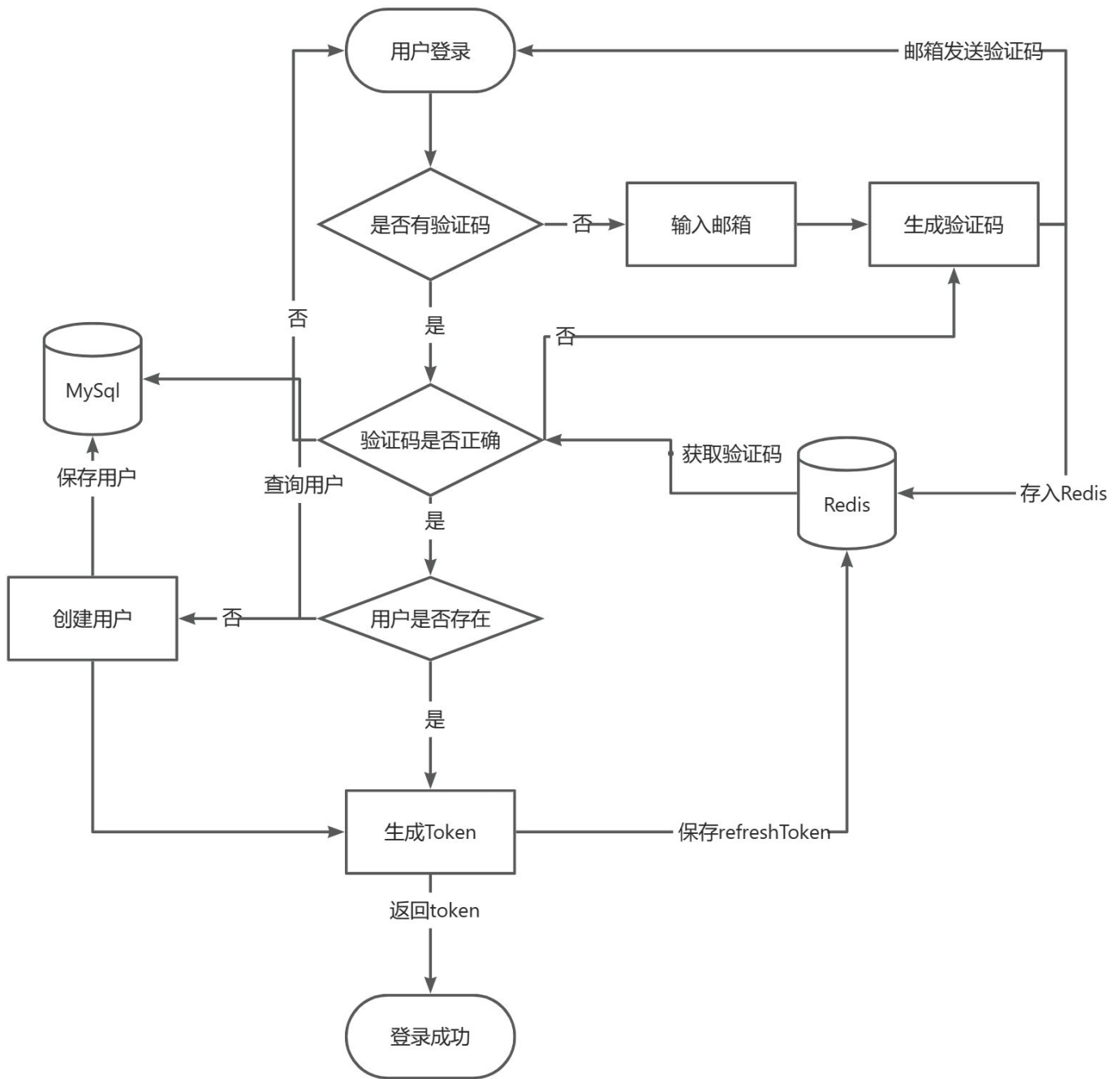
1. 登录授权模块

1.1. 验证码登录

具体流程：

1. 用户输入邮箱，并点击发送验证码。
2. 生成验证码，并以用户邮箱为key存入Redis，将验证码发送到用户邮箱。
3. 用户点击登录，用户信息以字符串方式到达UserDetailsService，在loadUserByUsername中进行验证码校验，若输入的验证码正确则继续下一步，否则保存提醒用户。
4. 实现了DaoAuthenticationProvider类并重写了additionalAuthenticationChecks方法，不进行密码校验
5. 生成token颁发令牌，将refreshToken保存到Redis刷新Token时使用

具体流程图如下：

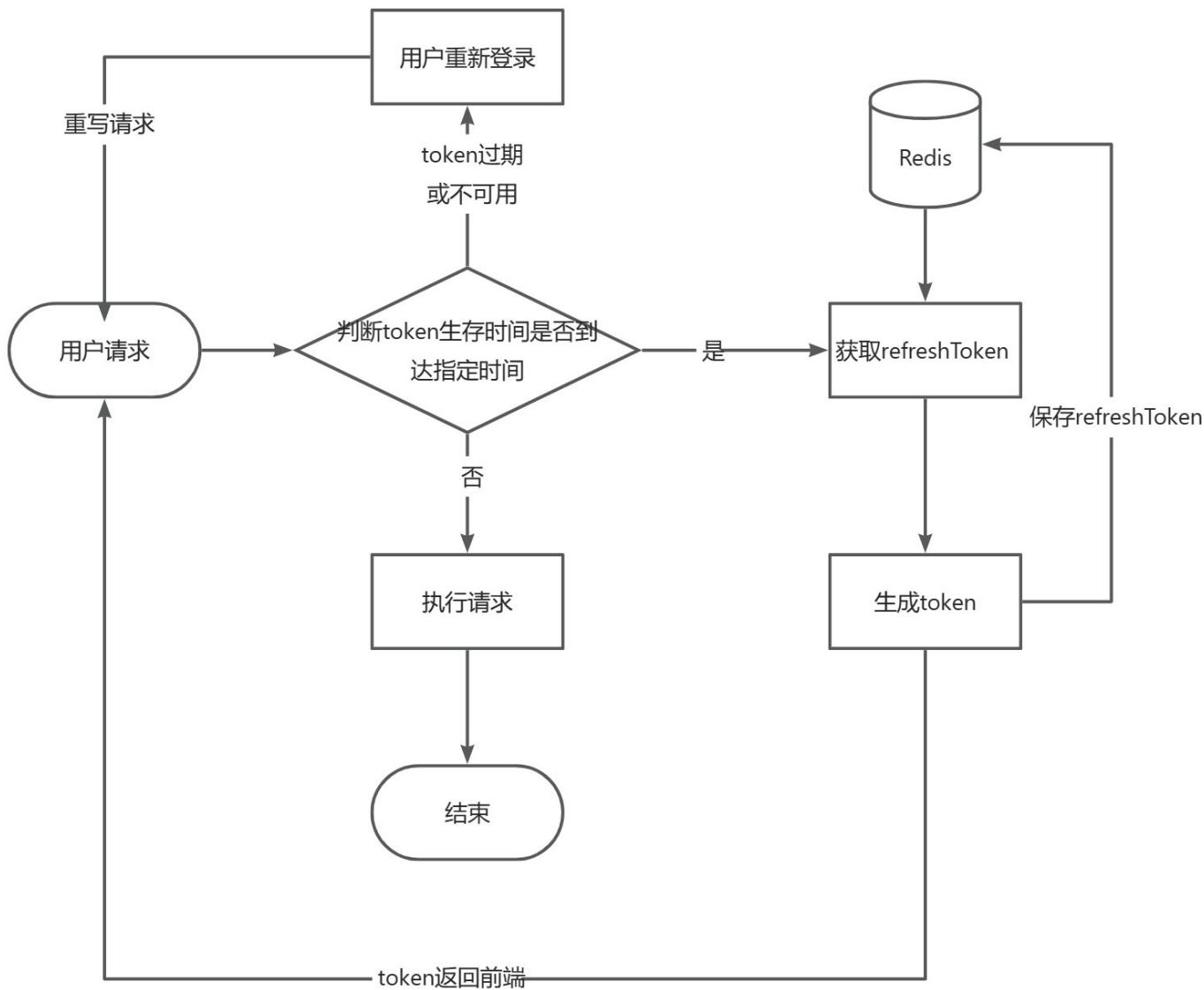


1.2. Token刷新

具体步骤如下：

1. 用户登录后，前端每次请求都携带token。
2. 请求首先到达拦截器，验证token的过期时间是否剩下5小时。
3. 如果未到达指定时间，放行。
4. 如果到达了指定时间，获取token中的用户信息，获取Redis中的refreshToken。
5. 携带refreshToken去申请token返回前端，并将新生成的refreshToken重新放入Redis。
6. 前端携带新token重写发起请求。

流程图如下：

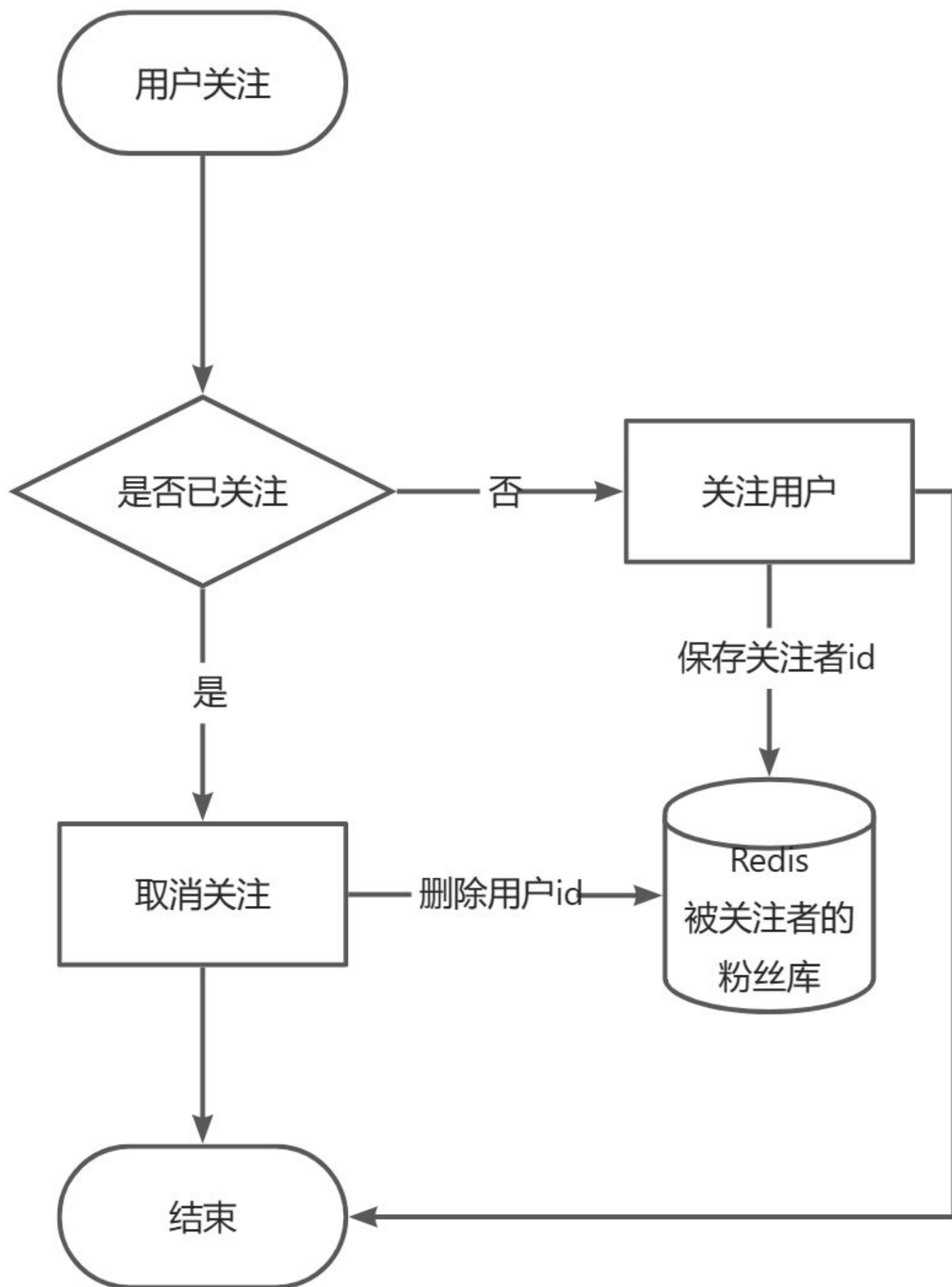


2. 关注和取消关注

具体步骤如下：

- 1. 用户登录后可关注其他用户。
- 2. 判断是否关注该用户
- 3. 未关注，添加到Mysql关注表中，并将该用户添加到对方的粉丝库中（Redis实现）。
- 4. 当对方发视频时会从Redis中读取到粉丝ID。
- 5. 根据粉丝ID将视频ID推送到粉丝的视频库中（Redis实现）。
- 6. 当粉丝登录时从视频库中获取视频推送给粉丝。

流程图如下：

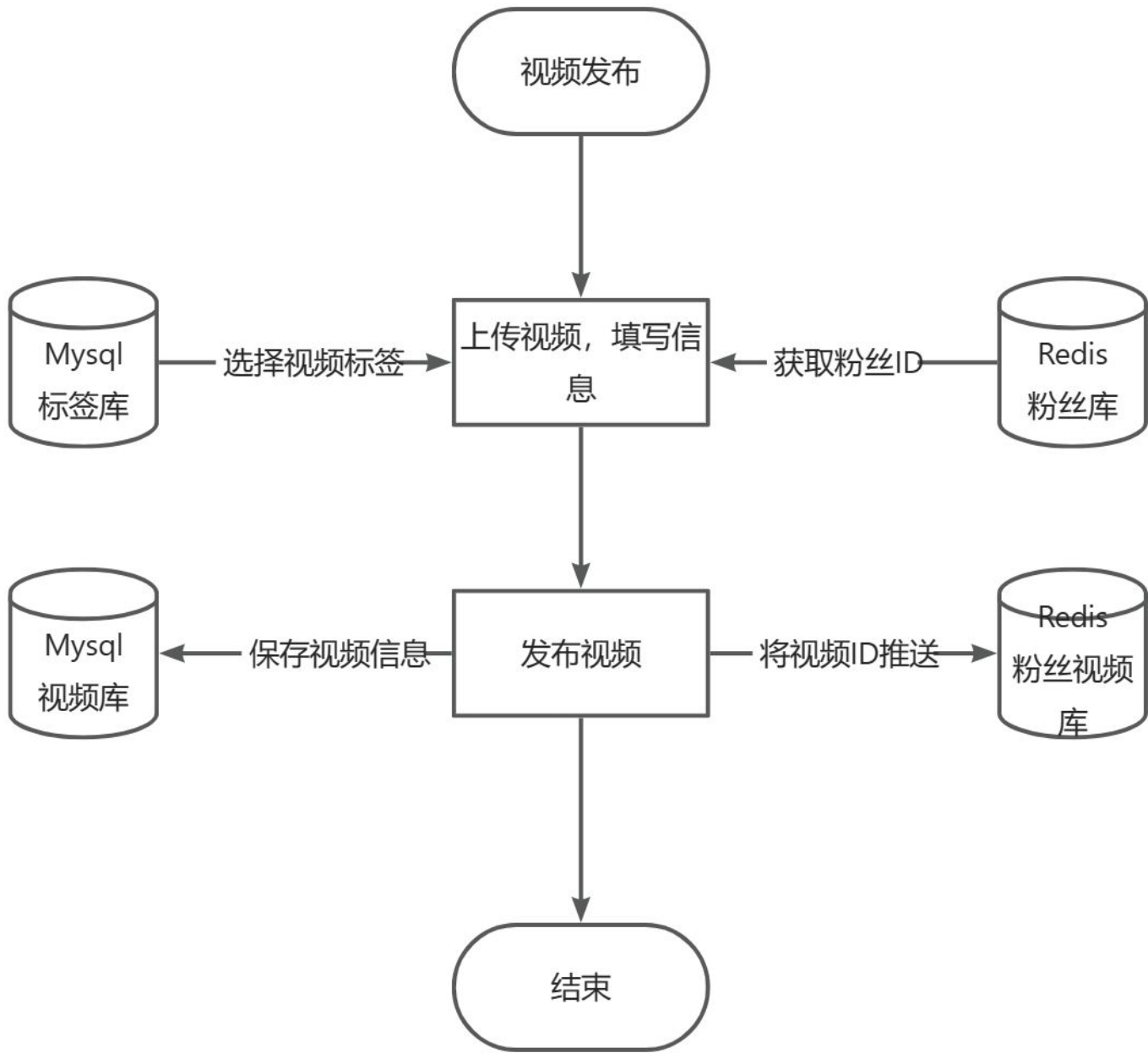


3. 视频发布接口

具体步骤如下：

- 1. 选择视频标签，填写视频基本信息。
- 2. 上传视频到七牛云。
- 3. 从Redis中获取粉丝列表。
- 4. 将视频ID保存到粉丝的视频库中。
- 5. 发布视频，保存视频信息到Mysql中。

流程图如下：



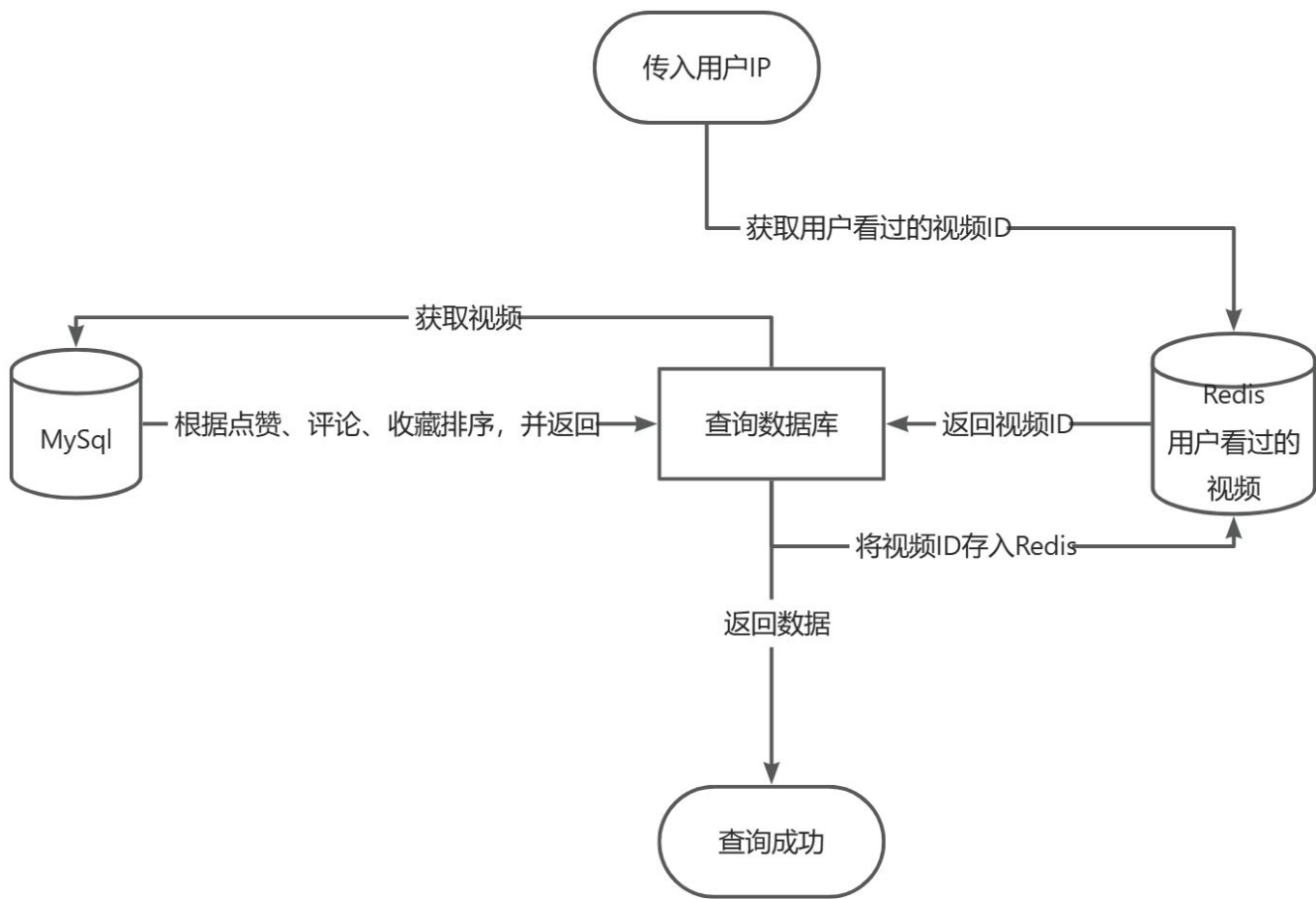
4. 视频推荐接口设计

4.1. 未登录时视频推荐

具体步骤如下：

1. 传入用户IP值（防止用户刷新页面刷到重复的视频）。
2. 以用户IP为key，从Redis的set集合中获取用户看过的视频ID集合。
3. 根据视频ID集合排除看过的视频，并通过视频的点赞、评论、收藏量的百分比计算分数倒序排列，将视频数据分页。
4. 使用IP作为key，看过的视频ID作为value，将用户看过的视频添加到Redis的set集合中，并设置过期时间。
5. 将视频相关的数据返回给前端。

流程图如下：



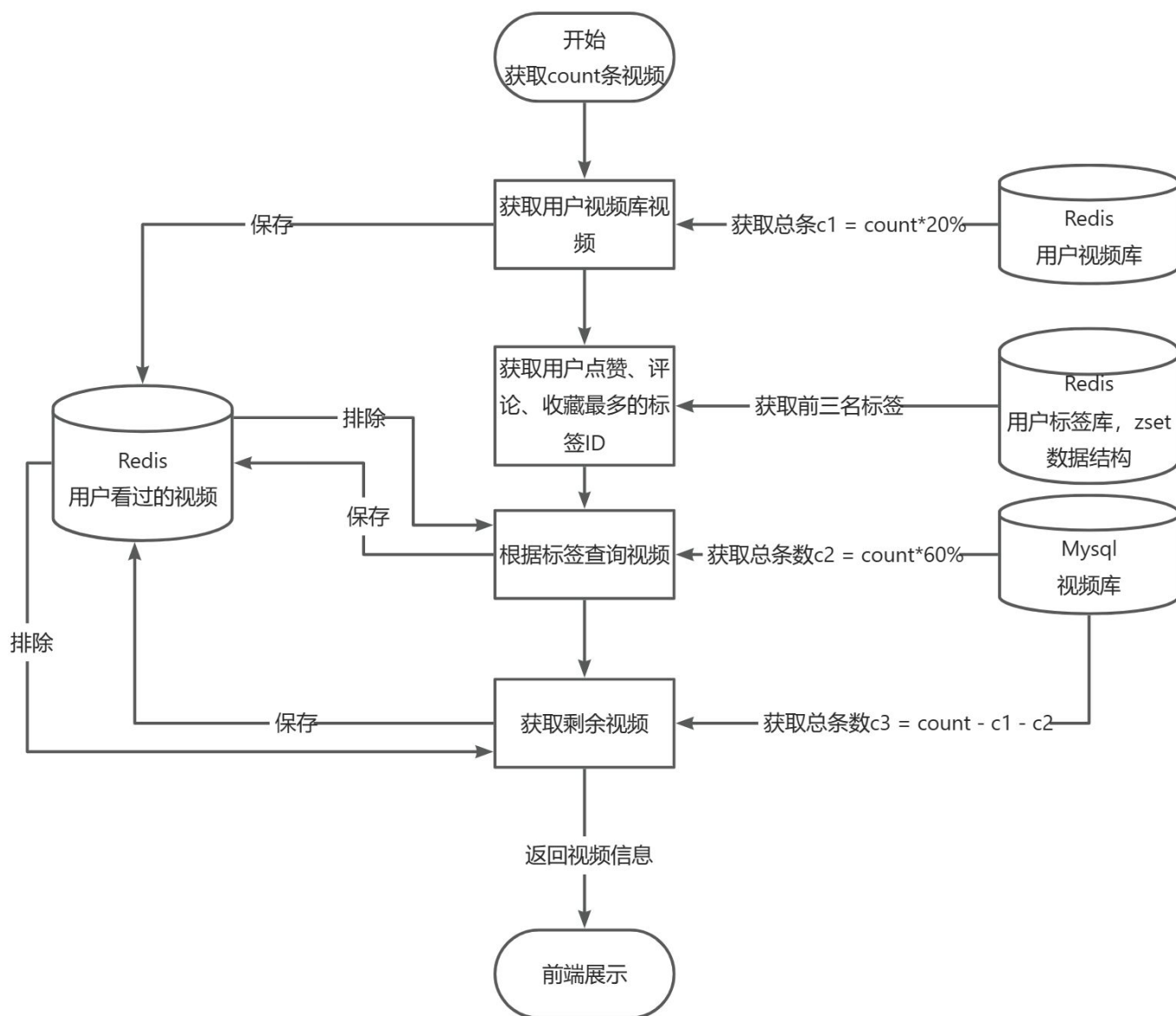
4.2. 登录后视频推荐

具体步骤如下：

1. 传入获取的视频条数count，从token中获得用户主键ID。

2. 从用户关注的人发布的视频库（Redis中，详见用户关注和发布视频接口）中随机选择 $\text{count} \times 20\%$ 条视频。
3. 从用户点赞、收藏视频的标签库中获取用户点赞，收藏最多的视频标签ID的前三（使用Redis的zset数据结构）。
4. 根据标签ID和未登录时的视频推荐策略获取视频数量为 $\text{count} \times 60\%$ ，并保存到用户看过的视频库中（使用Redis的set结构实现）。
5. 根据视频点赞、收藏、评论量计算视频分数，并排除用户看过的视频，获取 $\text{count} - \text{上两次获取视频的总和}$ （上面获取的视频可能达不到标准数量），并保存到用户看过的视频库中。

流程图如下：



5. 视频搜索接口

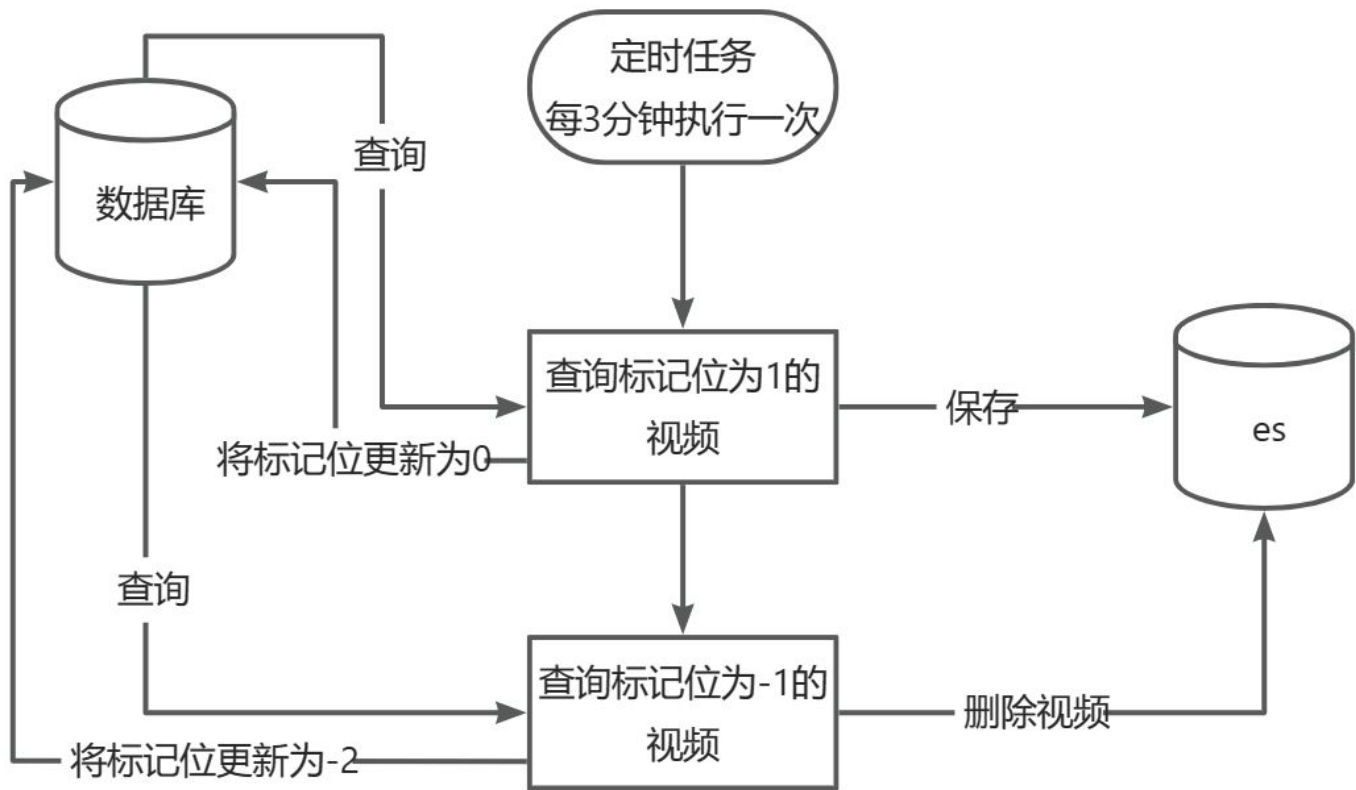
5.1. 视频数据同步

视频状态位说明：0正常，1未同步到es，-1数据库删除，-2数据库和es都删除

具体步骤如下：

- 1. 发布视频时将视频状态位标记为1。
- 2. 开启定时任务，每3分钟扫描数据库，找到标记位为1的视频和标记位为-1的视频。
- 3. 将标记位为1的视频批量插入到es中，并将数据库中的标记位变为0。
- 4. 将标记位为-1的视频从es中删除，并更新数据库标记位为-2。

流程图如下：

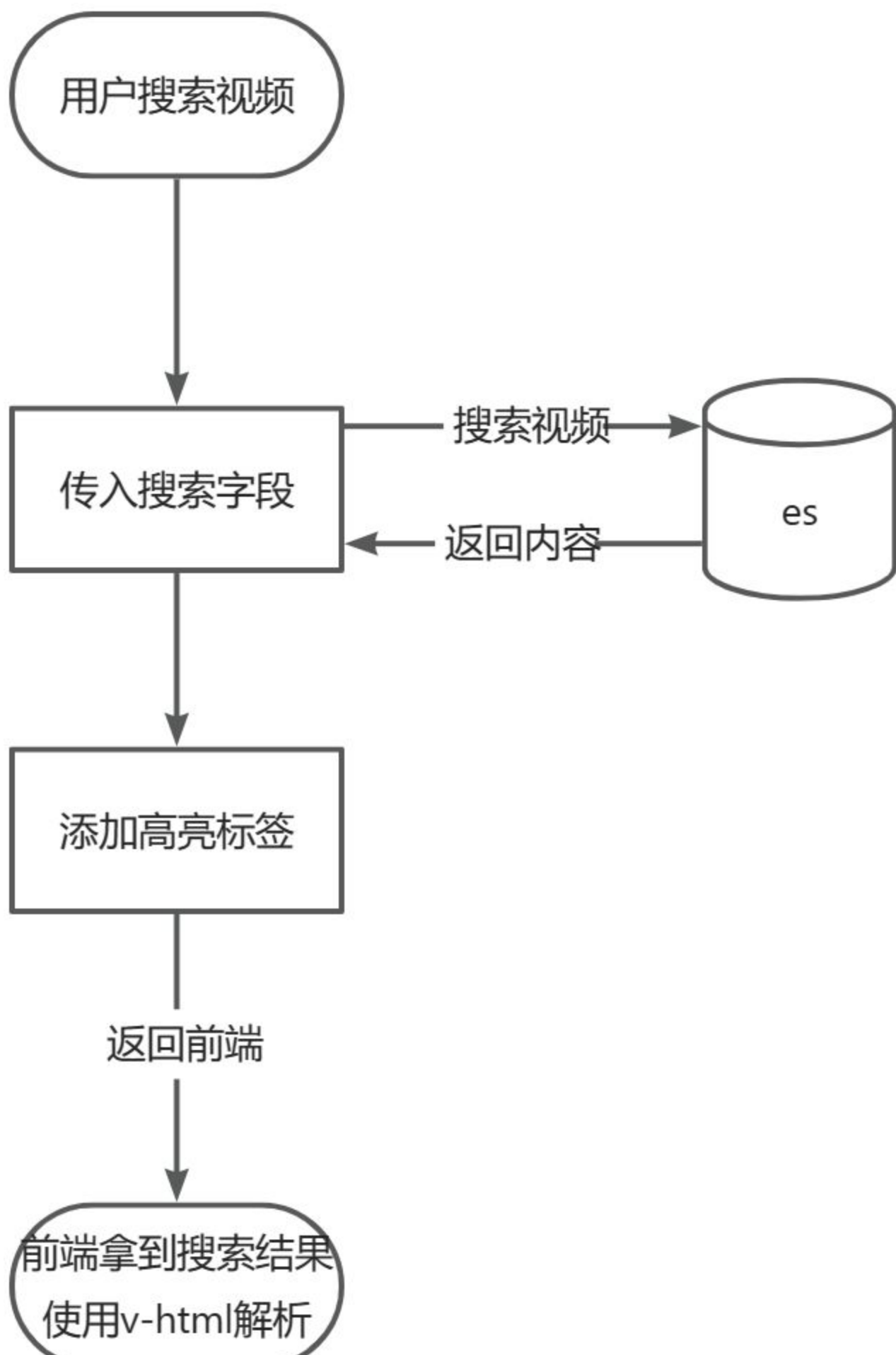


5.2. 高亮搜索

具体步骤如下：

- 1. 用户输入搜索的内容。
- 2. 根据ik分词器进行分词搜索。
- 3. 将搜索到的关键词添加搜索内容标签。
- 4. 将搜索到的内容返回前端，前端通过v-html解析标签，完成高亮搜索。

流程图如下：

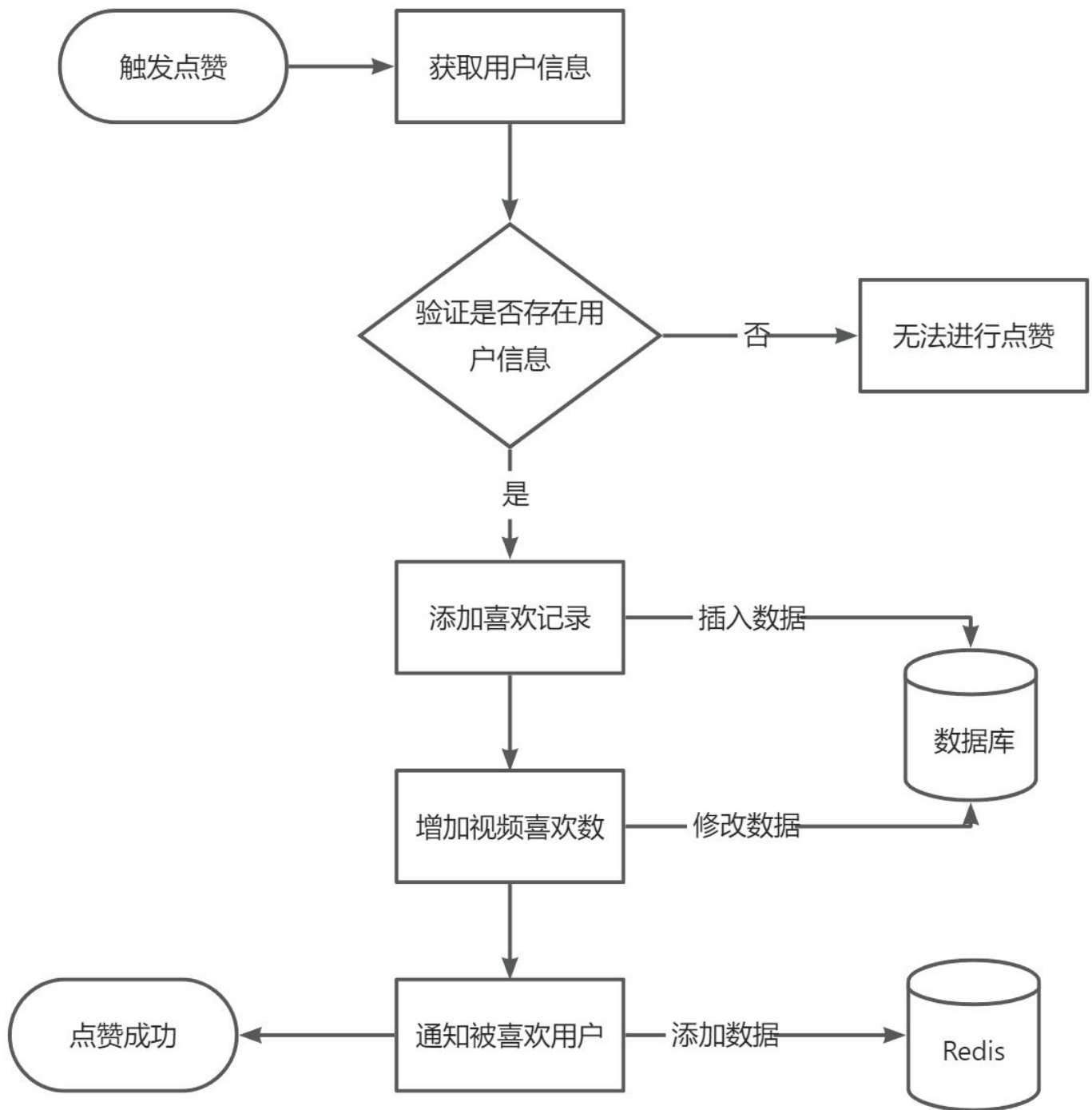


6. 点赞收藏模块

- 该模块包含业务：视频喜欢操作业务、视频收藏记录业务、消息通知业务、视频数据操作业务
- 业务相关技术：用户验证-SpringSecurity、持久化信息保存（记录信息和数据信息修改）-MySQL+MybatisPlus、消息通知-Redis

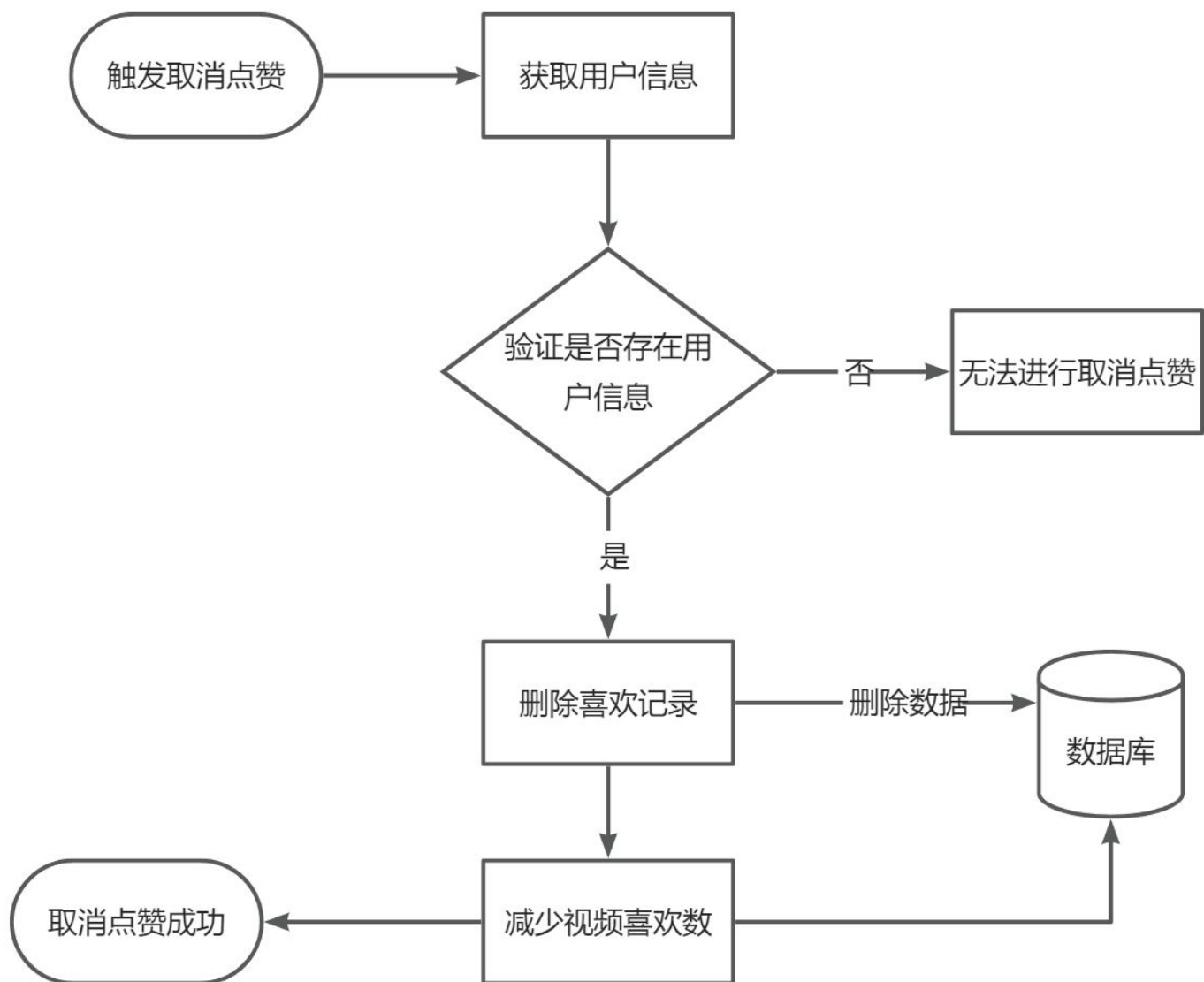
6.1. 点赞视频

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和视频相关信息保存用户喜欢对应视频的记录，保存进数据库中，如果不存在用户信息无法进行点赞
3. 由于喜欢某个视频，该视频的喜欢数增加，也需要进行对视频数据的修改，点赞数增加
4. 发布该视频的用户需要收到被喜欢的信息，于是通过redis进行保存喜欢的该视频的用户信息和对应视频的信息



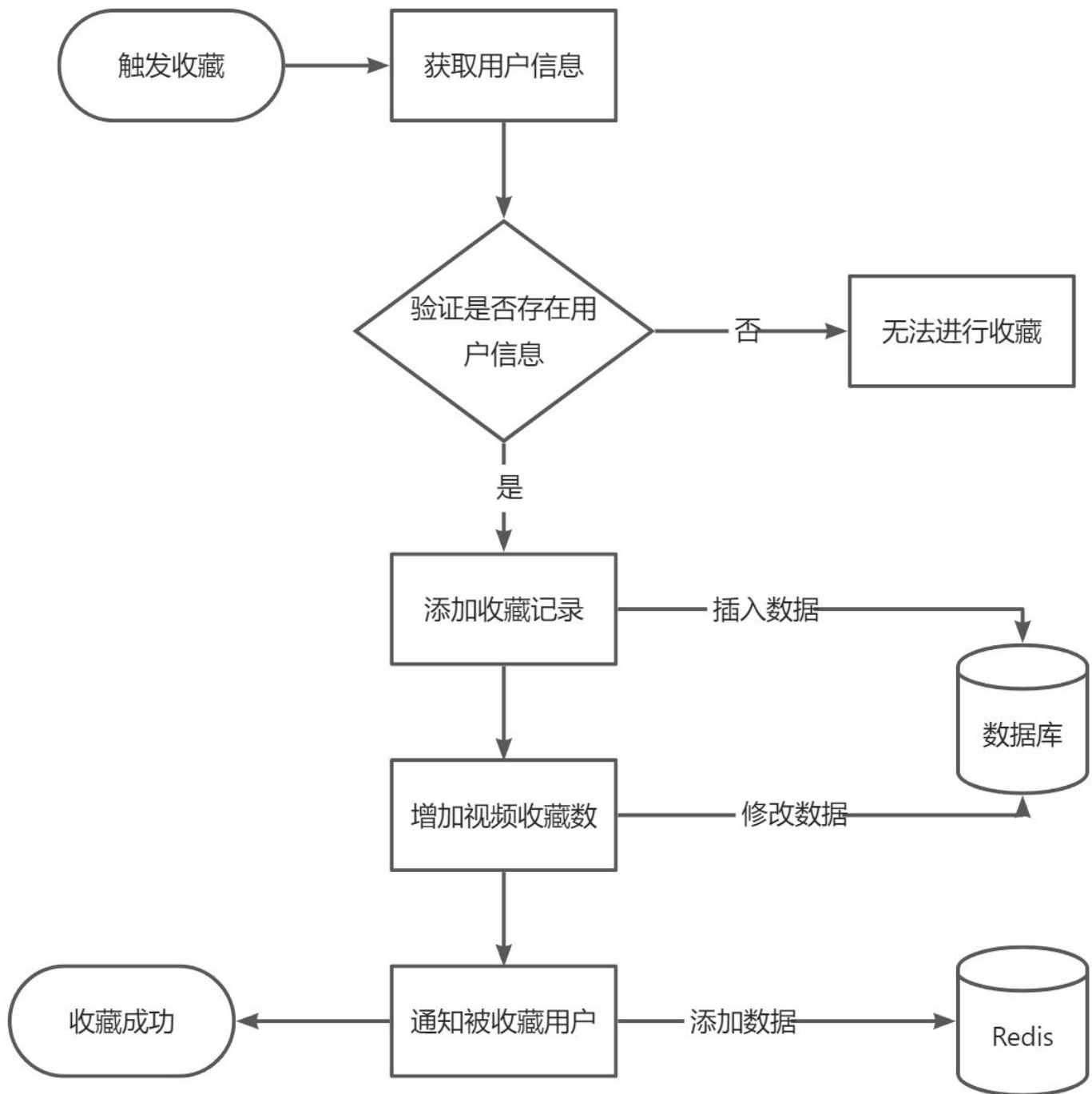
6.2. 取消点赞

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和视频相关信息对喜欢记录进行删除，如果不存在用户信息无法进行取消点赞
3. 由于取消喜欢某个视频，该视频的喜欢数减少，也需要进行对视频数据的修改，点赞数减一
4. 取消点赞则不需要进行消息通知



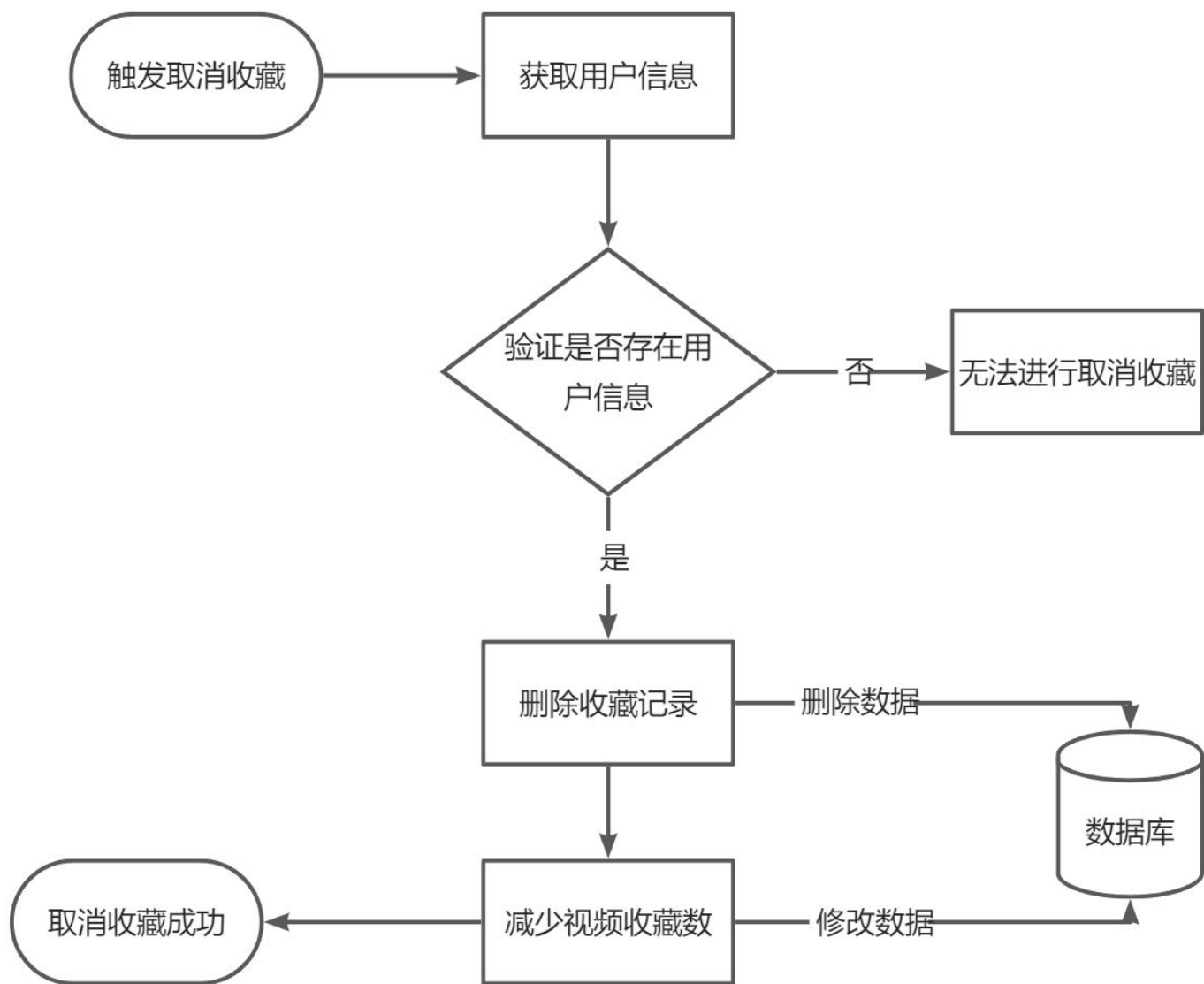
6.3. 收藏视频

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和视频相关信息保存用户收藏对应视频的记录，保存进数据库中，如果不存在用户信息无法进行收藏
3. 由于收藏某个视频，该视频的收藏数增加，也需要进行对视频数据的修改，收藏数增加
4. 发布该视频的用户需要收到被收藏的信息，于是通过redis进行保存收藏的该视频的用户信息和对应视频的信息



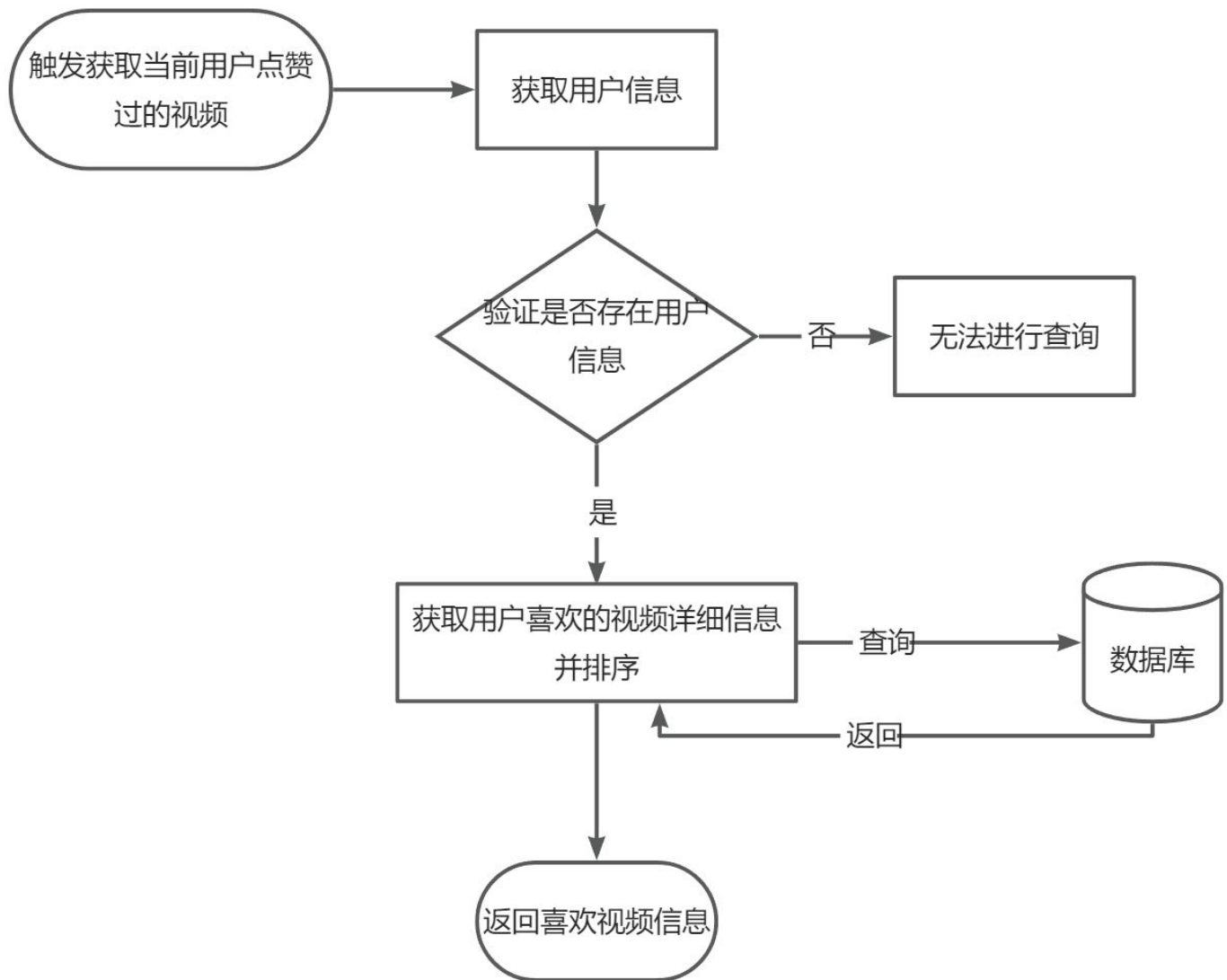
6.4. 取消收藏

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和视频相关信息对收藏记录进行删除，如果不存在用户信息无法进行取消收藏
3. 由于取消收藏某个视频，该视频的收藏数减少，也需要进行对视频数据的修改，收藏数减一
4. 取消收藏则不需要进行消息通知



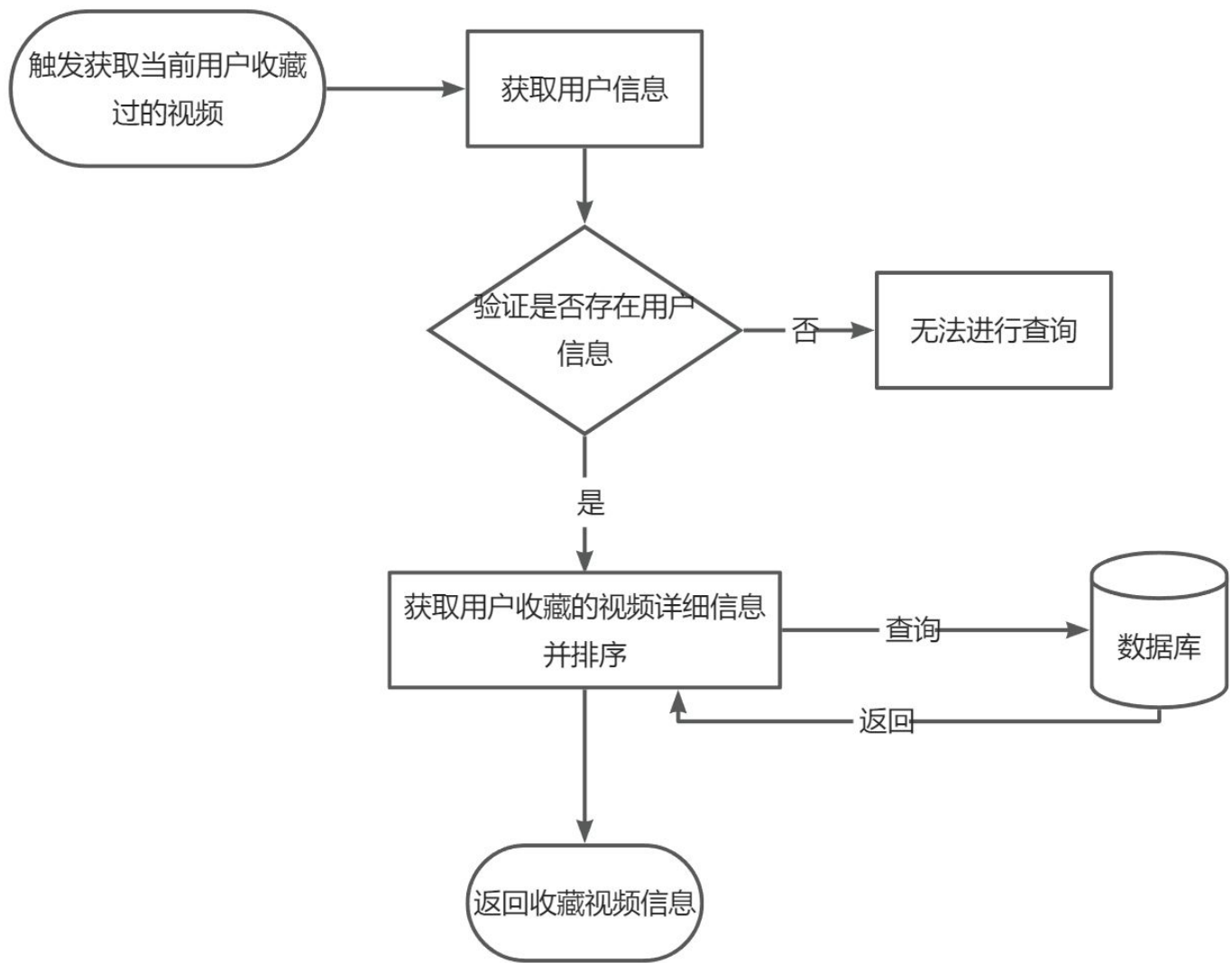
6.5. 获取当前用户点赞过的视频

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id对点赞记录进行查询，连表视频消息信息表，查询到视频详细信息，如果不存在用户信息则无法进行查询
3. 喜欢过的视频对记录时间进行排序返回给前端数据集



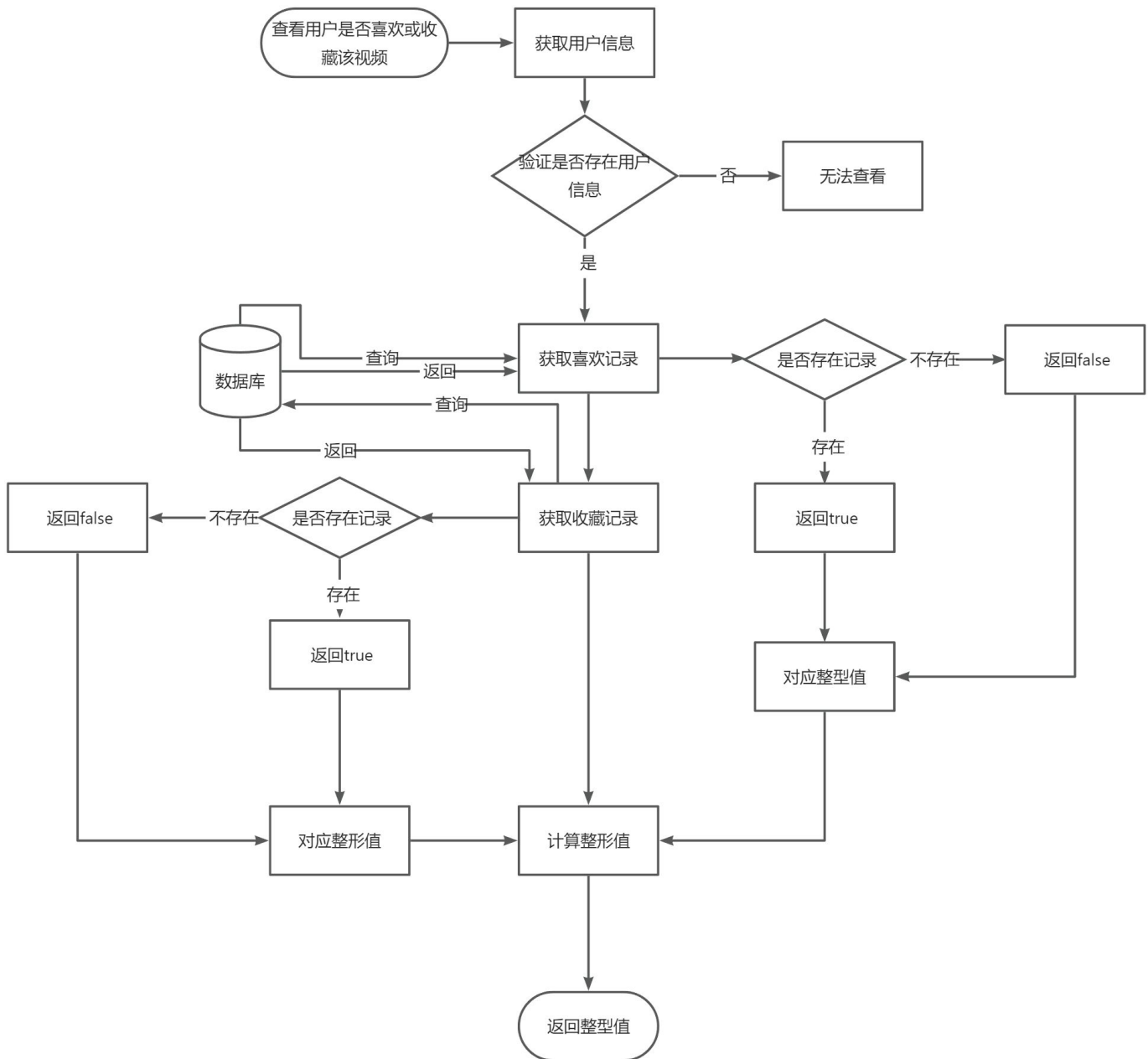
6.6. 获取当前用收藏过的视频

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id对收藏记录进行查询，连表视频消息信息表，查询到视频详细信息，如果不存在用户信息则无法进行查询
3. 收藏过的视频对记录时间进行排序返回给前端数据集



6.7. 查看用户是否喜欢或收藏过该视频

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和视频信息对喜欢和收藏记录进行查询，如果不存在用户信息则无法进行查询
3. 如果喜欢记录存在用户喜欢该视频的记录，则返回标记值true，否则返回false
4. 如果收藏记录存在用户收藏该视频的记录，则返回标记值true，否则返回false
5. 通过标记值进行用整型值分类，返回0表示未喜欢且未收藏，返回1表示喜欢但未收藏，返回2表示收藏但未喜欢，返回3表示喜欢且收藏



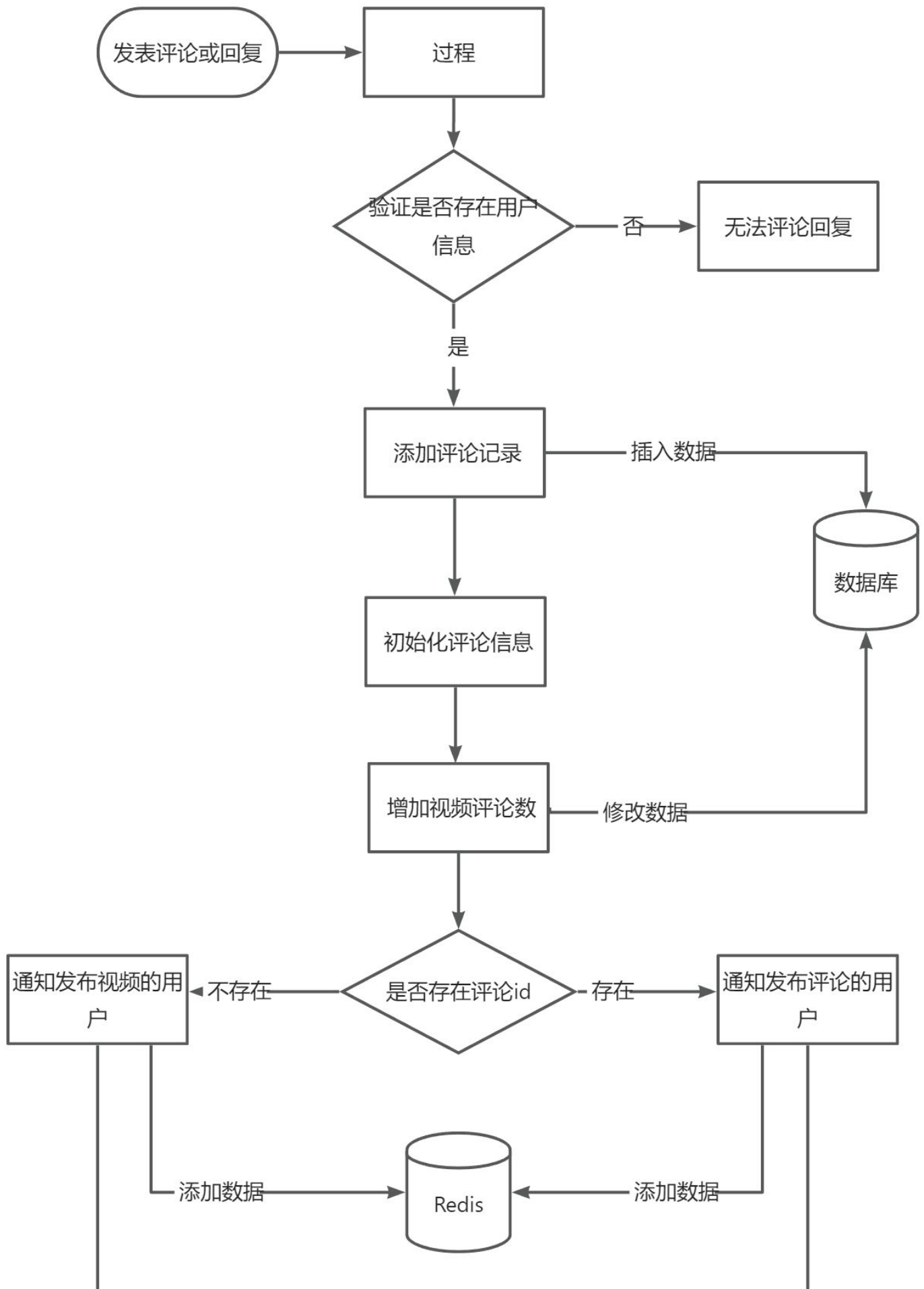
7. 视频评论模块

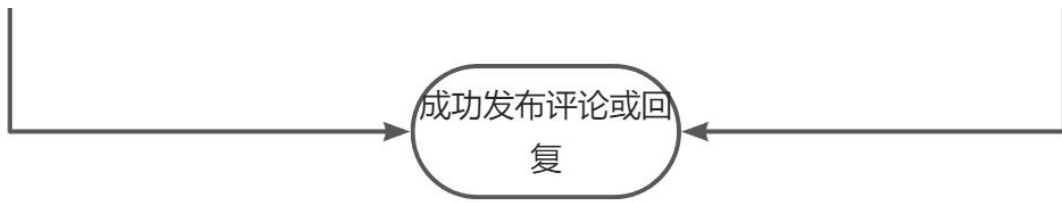
- 该模块包含业务： **评论信息操作业务** 、 **评论点赞记录业务** 、 **消息通知业务** 、 **视频数据操作业务**
- 业务相关技术：用户验证-SpringSecurity、持久化信息保存（记录信息和数据信息修改）-MySQL+MybatisPlus、消息通知-Redis

7.1. 发布评论或回复

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和视频信息可进行该视频下的评论，如果不存在用户信息则无法进行评论
3. 判断是否附带评论id，如果附带评论id则是回复，不附带评论id则是视频下的直接评论

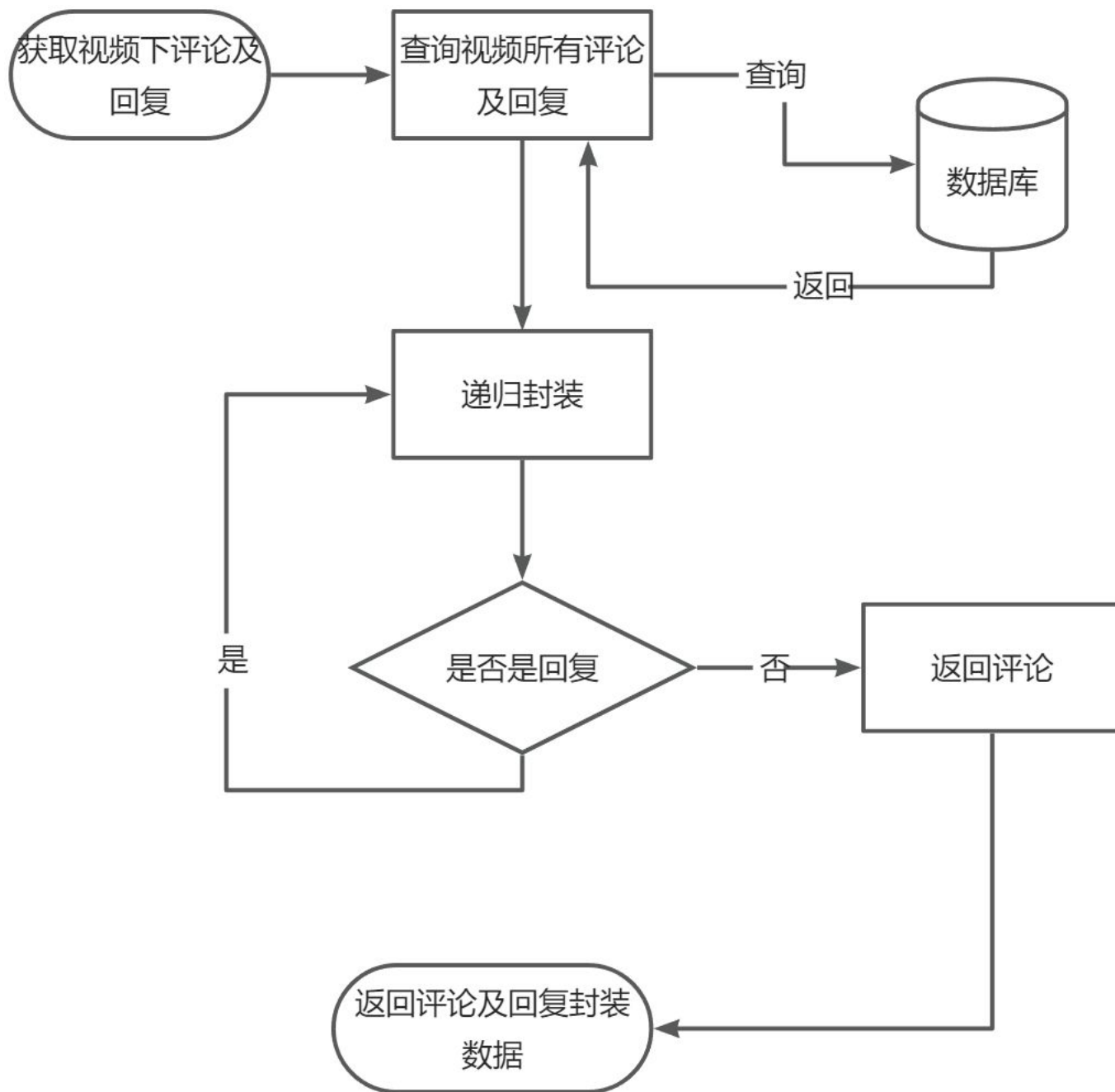
4. 添加评论记录于数据库中
5. 视频的评论数也会进行增加，对视频评论数进行加一
6. 如果是直接评论则会通知发布视频的用户评论的内容即用户信息
7. 如果是回复，则会通知发布该评论的用户被回复的内容和用户信息





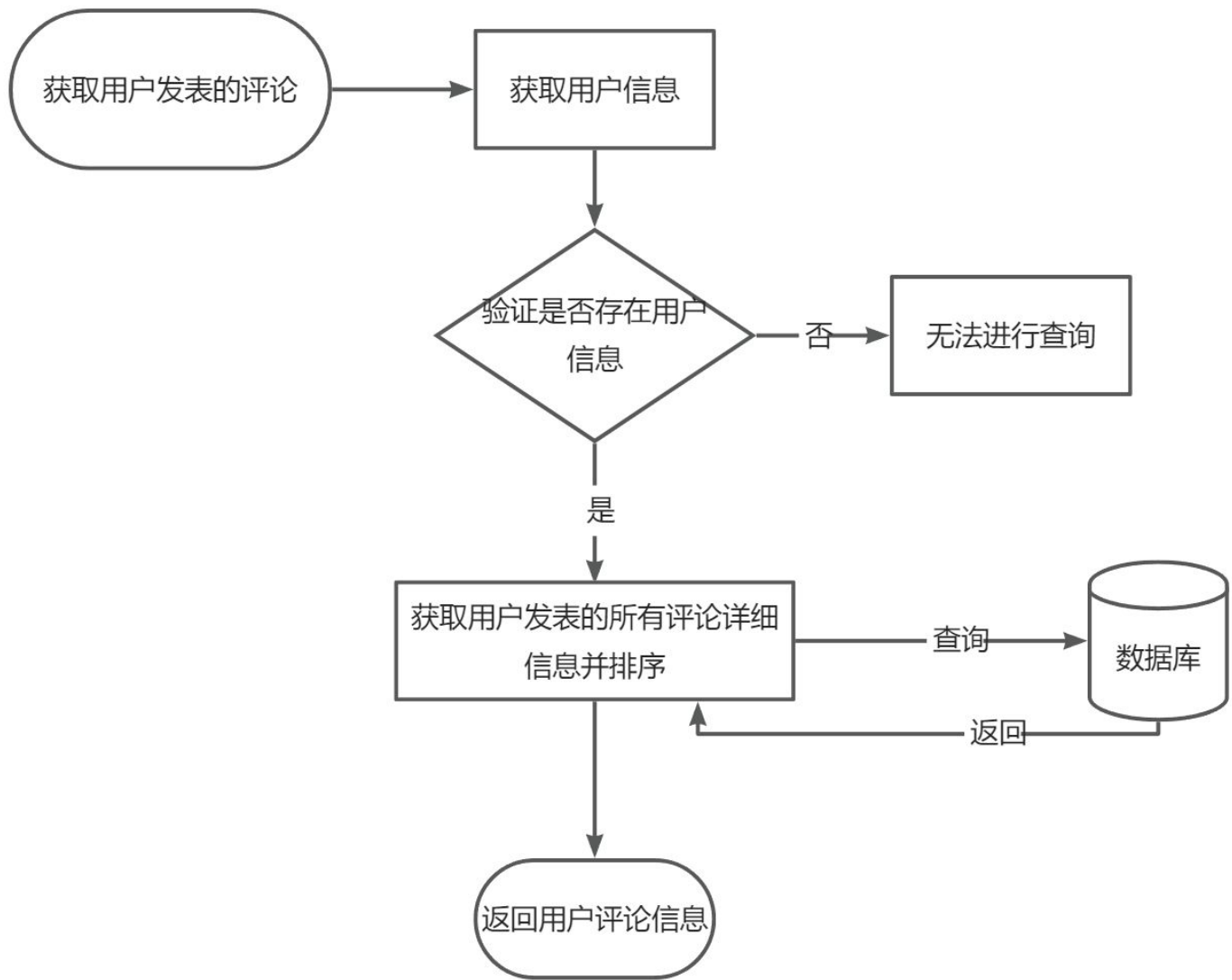
7.2. 获得视频下所有评论及回复

1. 通过视频id查询该视频所有的评论及回复
2. 递归封装，通过判断是否是回复，是回复则封装到父评论中
3. 返回封装后的评论，这些评论均是视频直接评论，包含此评论下的所有回复



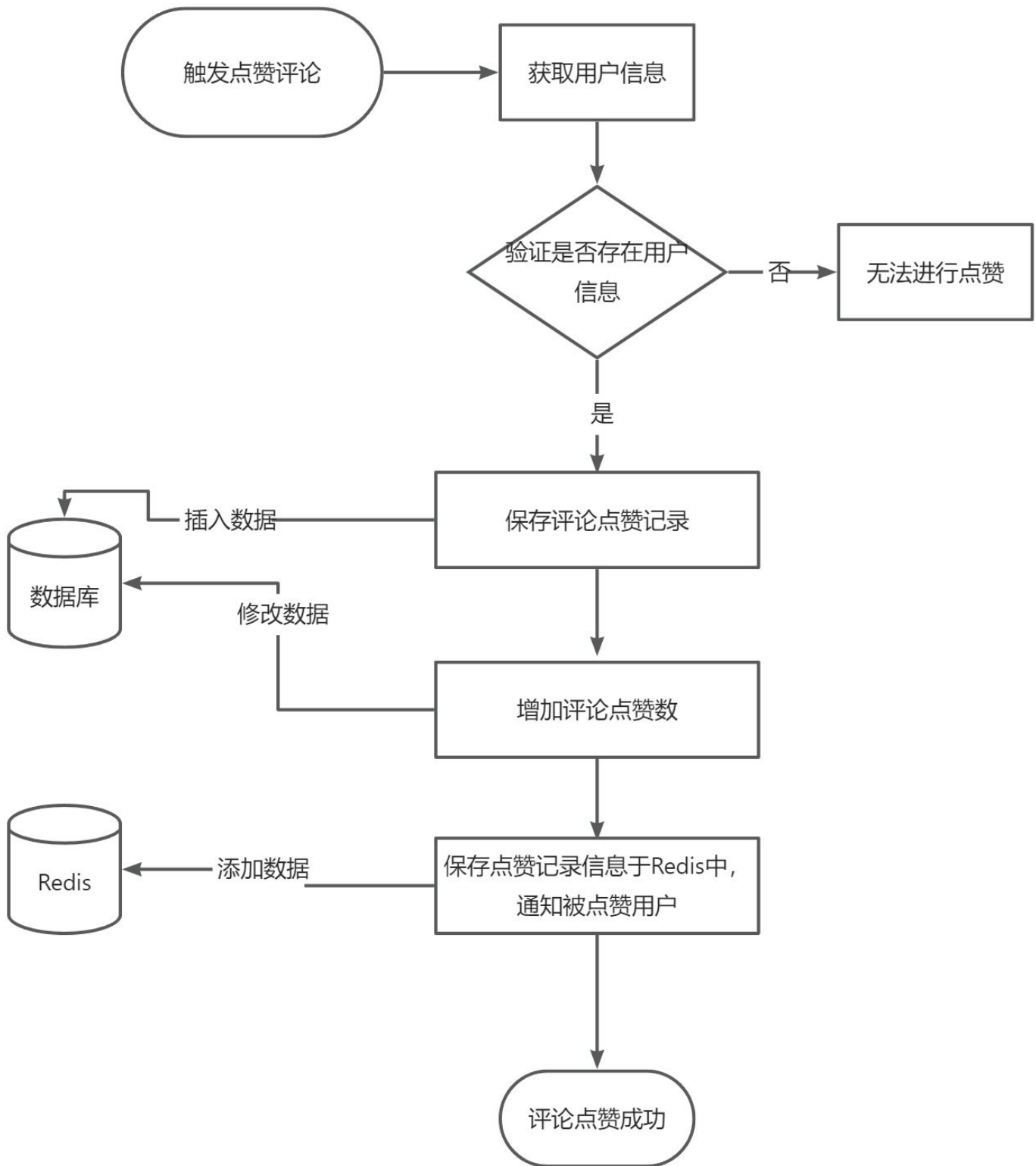
7.3. 获得用户发表的评论

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id获取所有该用户评论的详细信息，如果不存在用户信息则无法进行查询
3. 查询结果通过发表时间进行排序



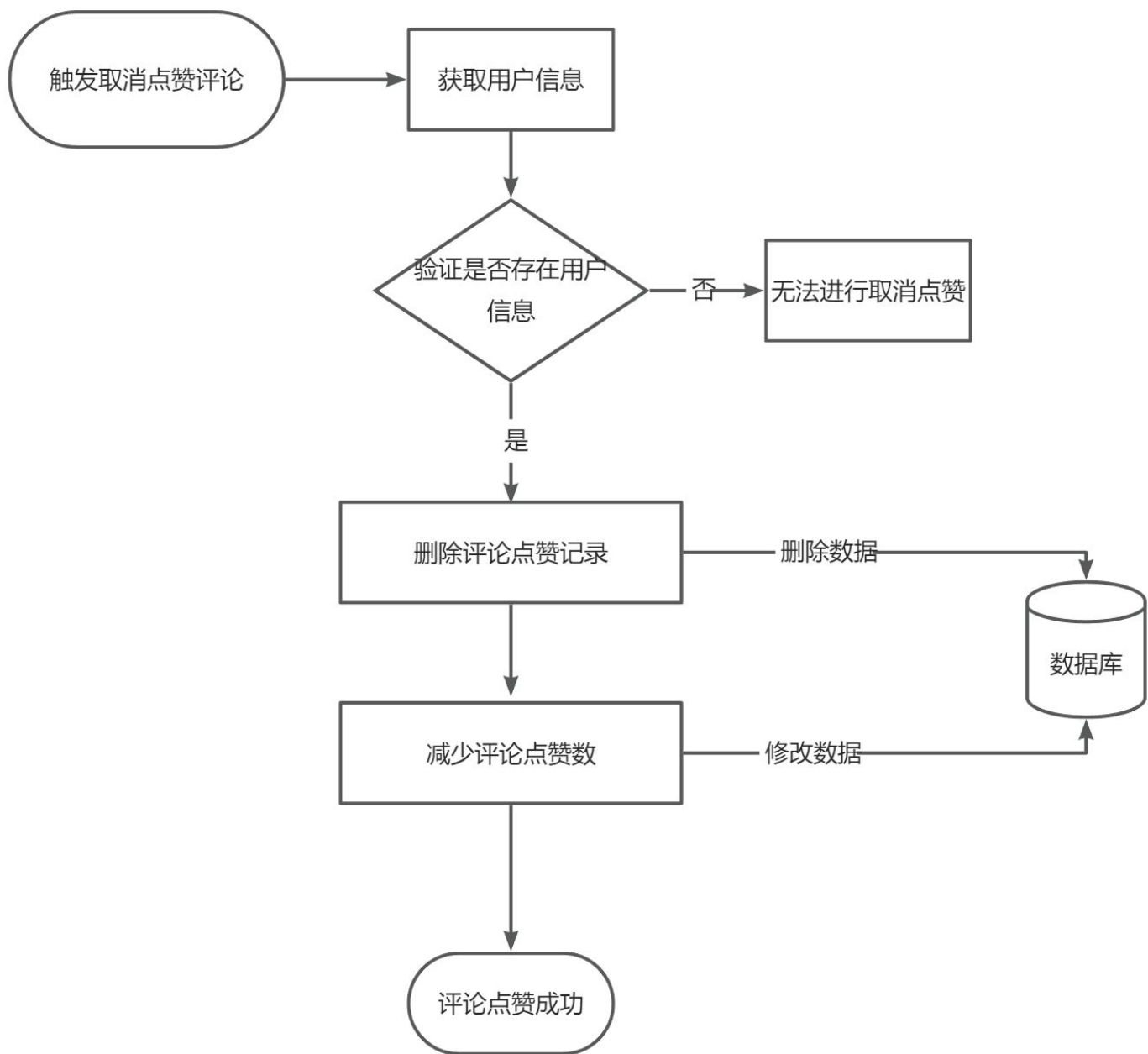
7.4. 点赞评论

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和评论id保存评论点赞记录，如果不存在用户信息则无法进行点赞
3. 点赞记录成功添加，则进行更新评论点赞数，评论点赞数加一
4. 为了被点赞者得知评论被点赞，点赞用户的信息需要进行保存到redis用于消息的通知



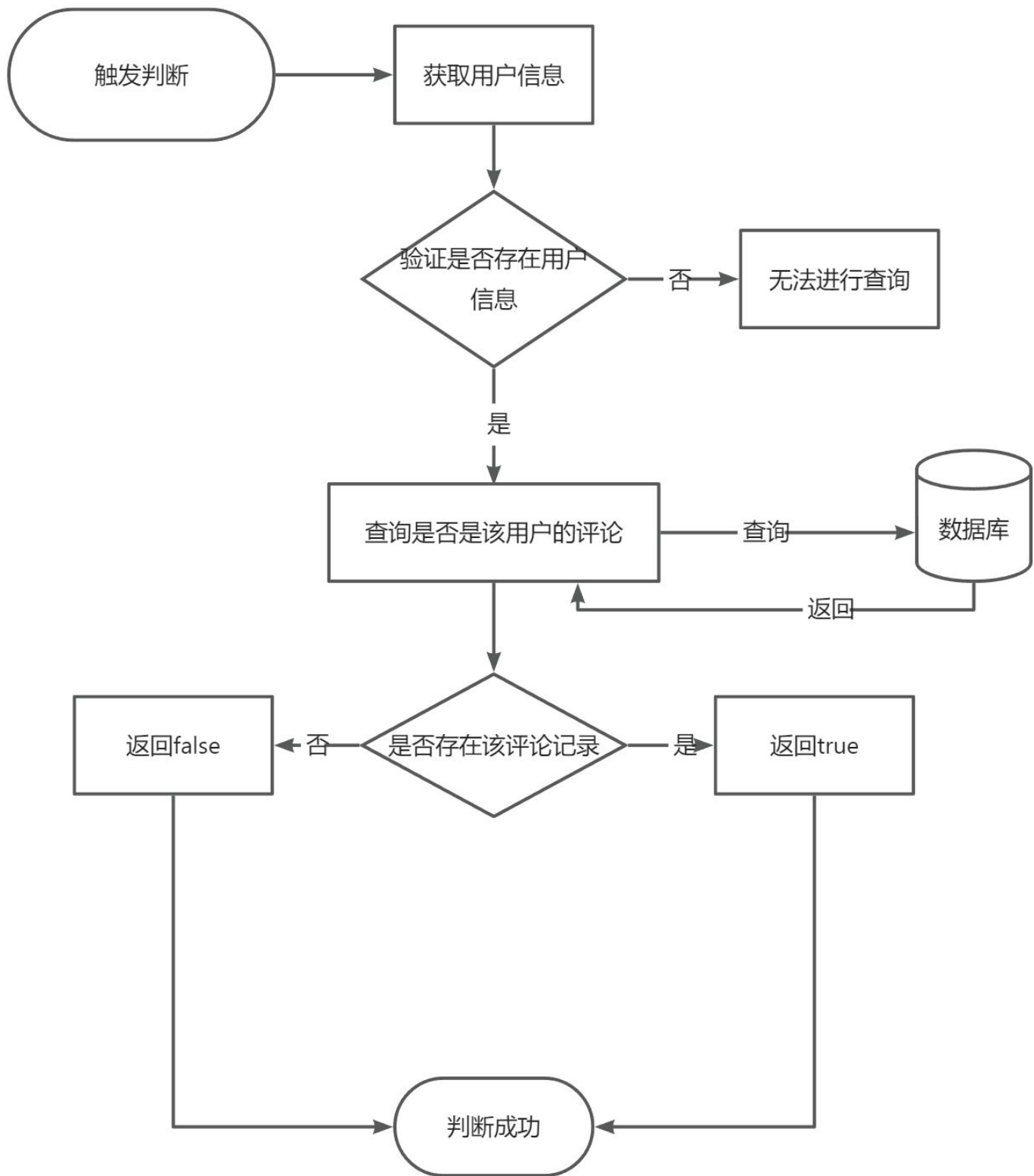
7.5. 取消点赞评论

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和评论id删除评论点赞记录，如果不存在用户信息则无法进行取消点赞
3. 点赞记录成功删除，则进行更新评论点赞数，评论点赞数减一
4. 取消点赞则不需要进行消息通知



7.6. 判断是否是该用户的评论

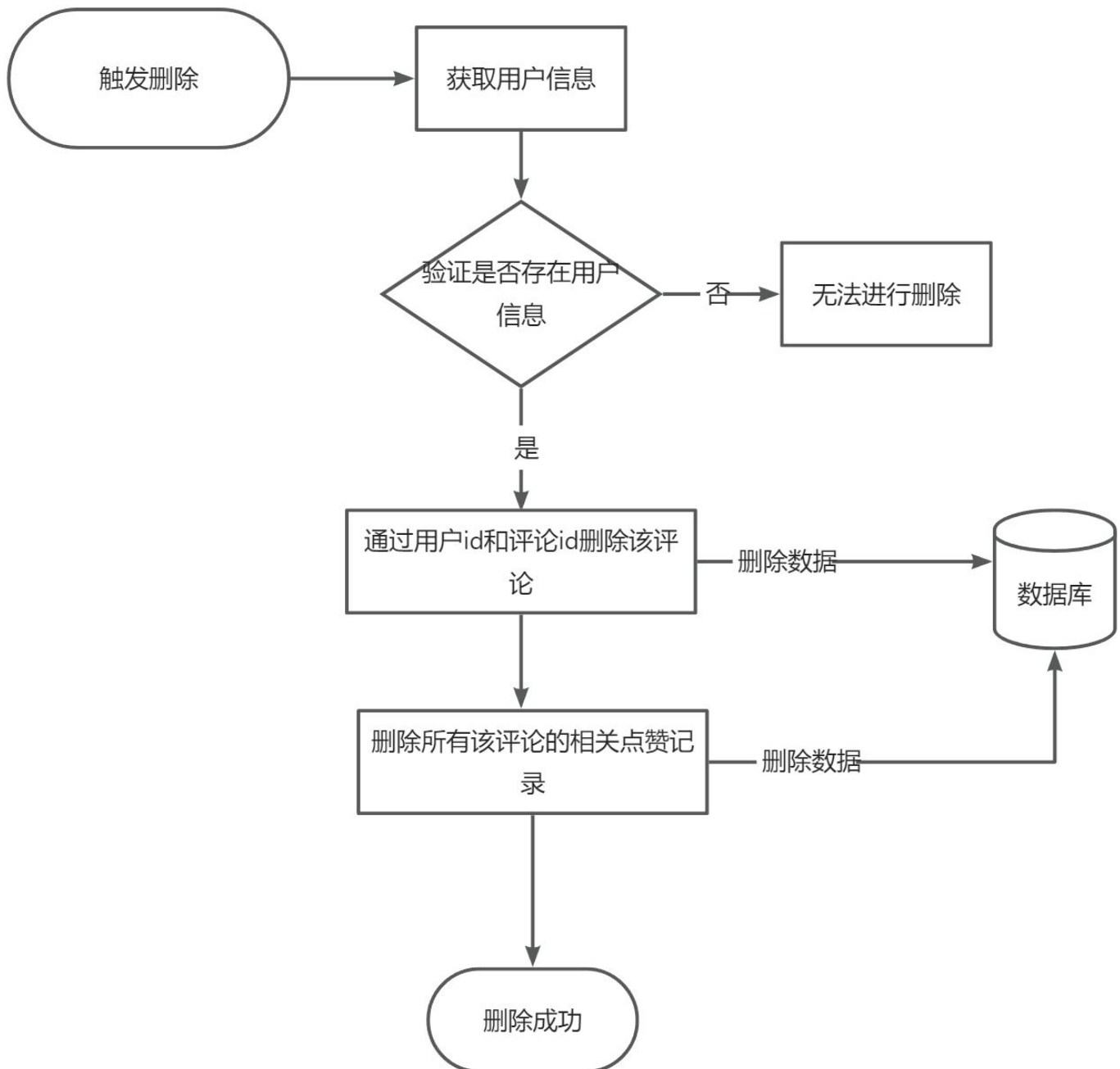
1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和评论id查询是否是该用户评论，如果不存在用户信息则无法进行查询
3. 如果是该用户的评论则返回true，否则返回false



7.7. 删除评论

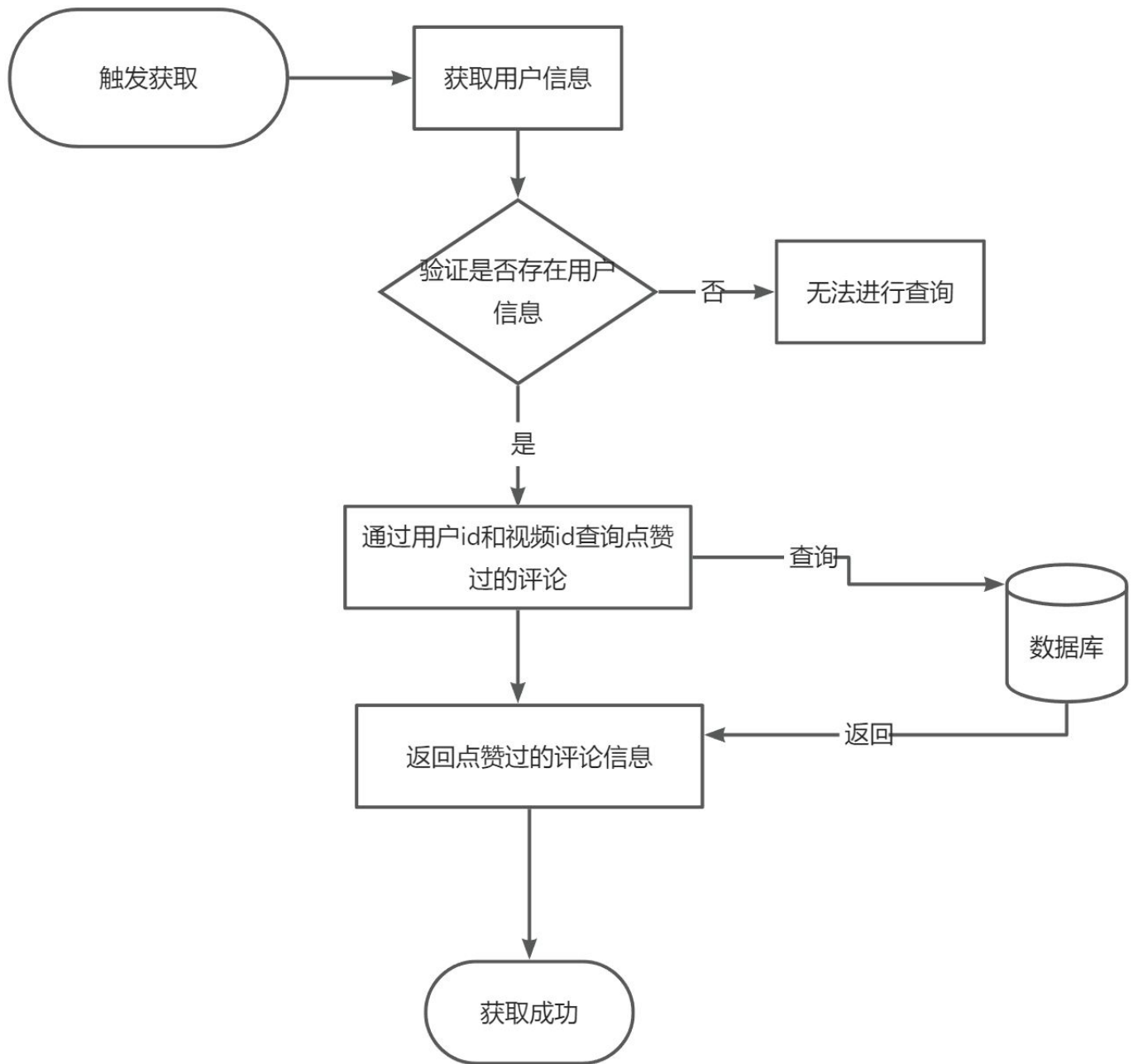
1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和评论id进行删除该评论，如果不存在用户信息则无法进行删除

3. 评论删除成功，对应的删除所有该评论的所有点赞记录



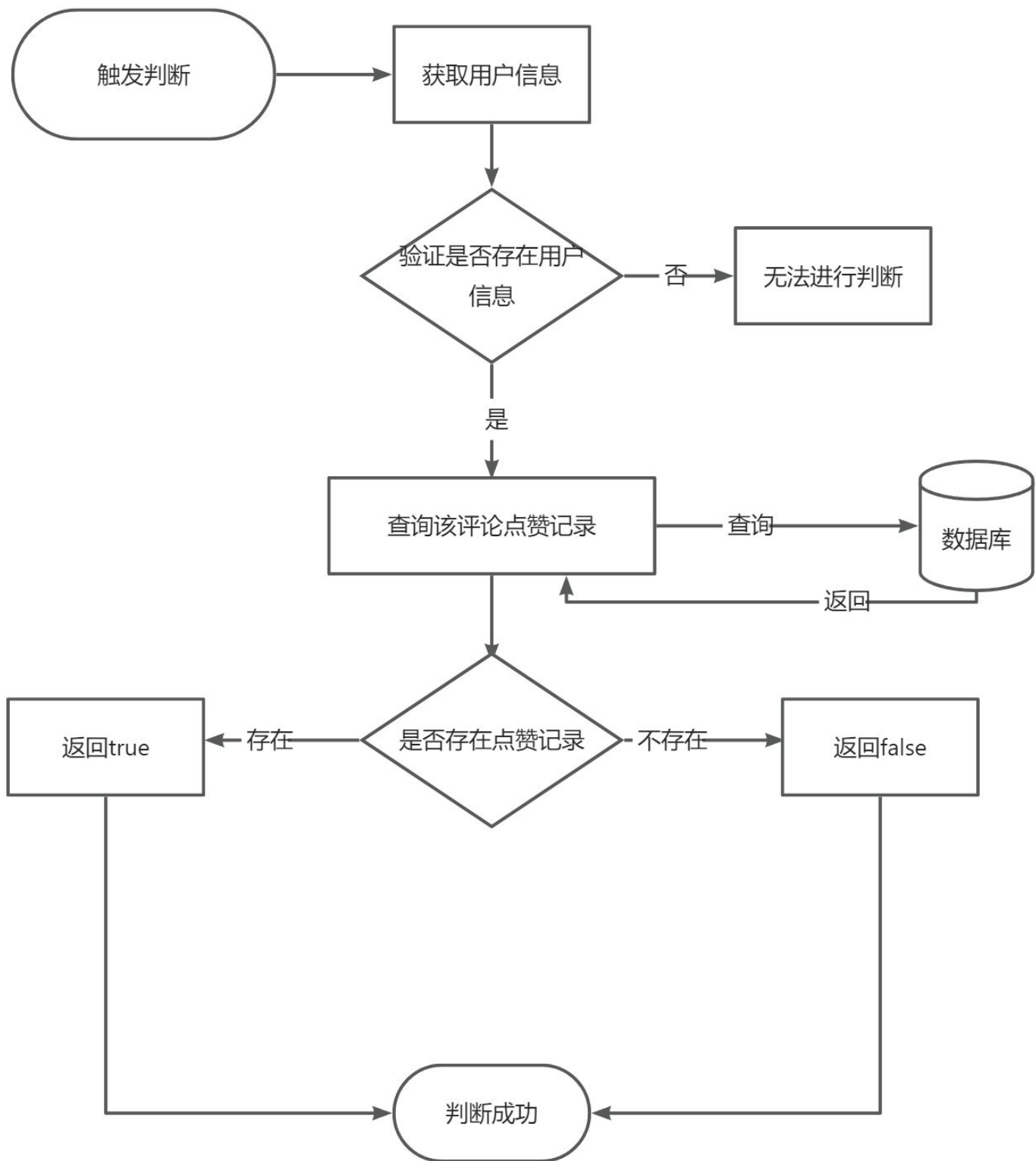
7.8. 获取该视频下此用户点赞过的评论

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和视频id进行查询该用户点赞的评论，如果不存在用户信息则无法进行查询
3. 返回用户点赞过的评论信息



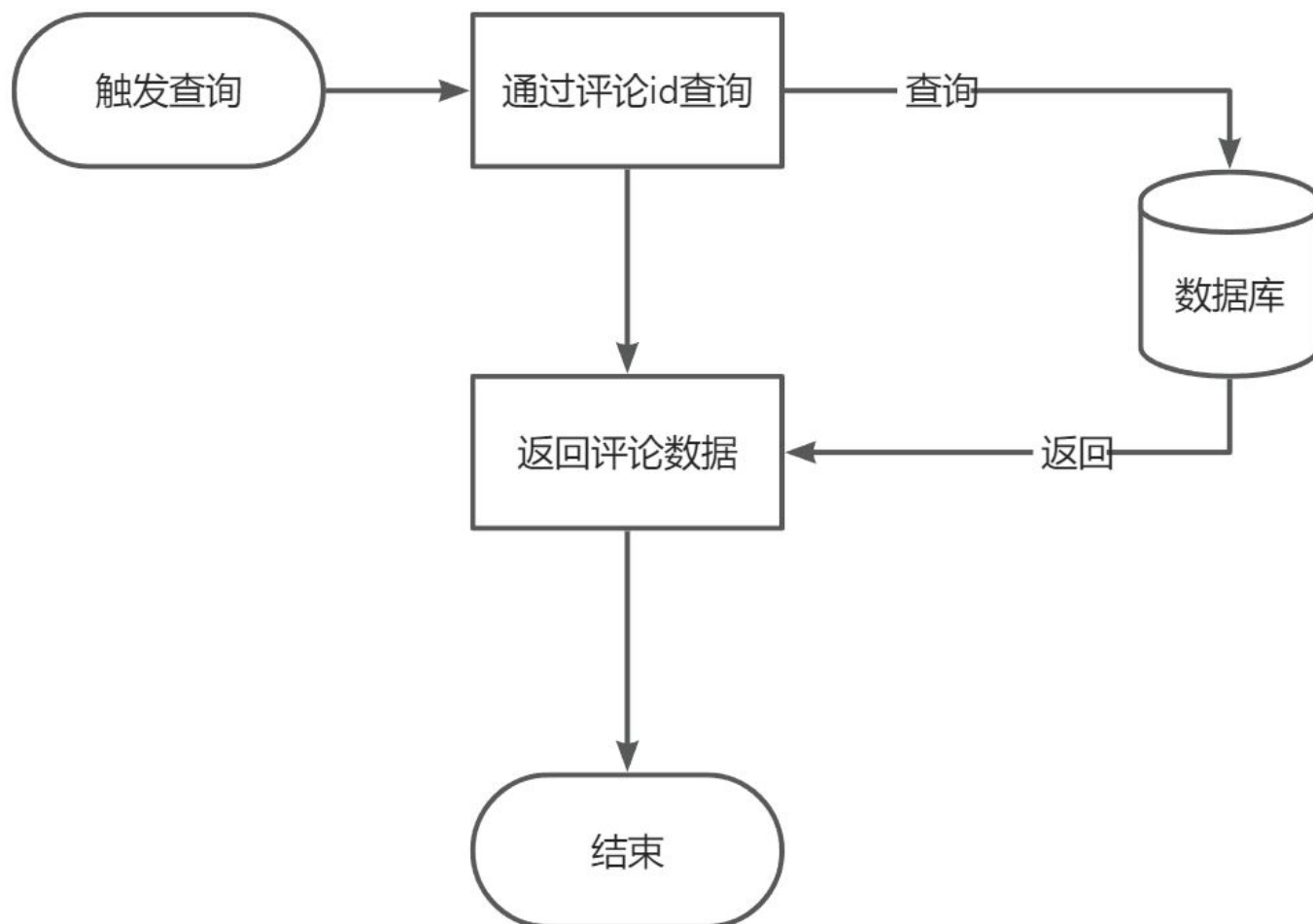
7.9. 判断用户是否点赞过该评论

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id和评论id进行判断该用户是否点赞该评论，如果不存在用户信息则无法进行判断
3. 返回状态信息



7.10. 通过评论id查看评论详细内容

1. 通过评论id查询该评论详细信息
2. 返回评论详细信息

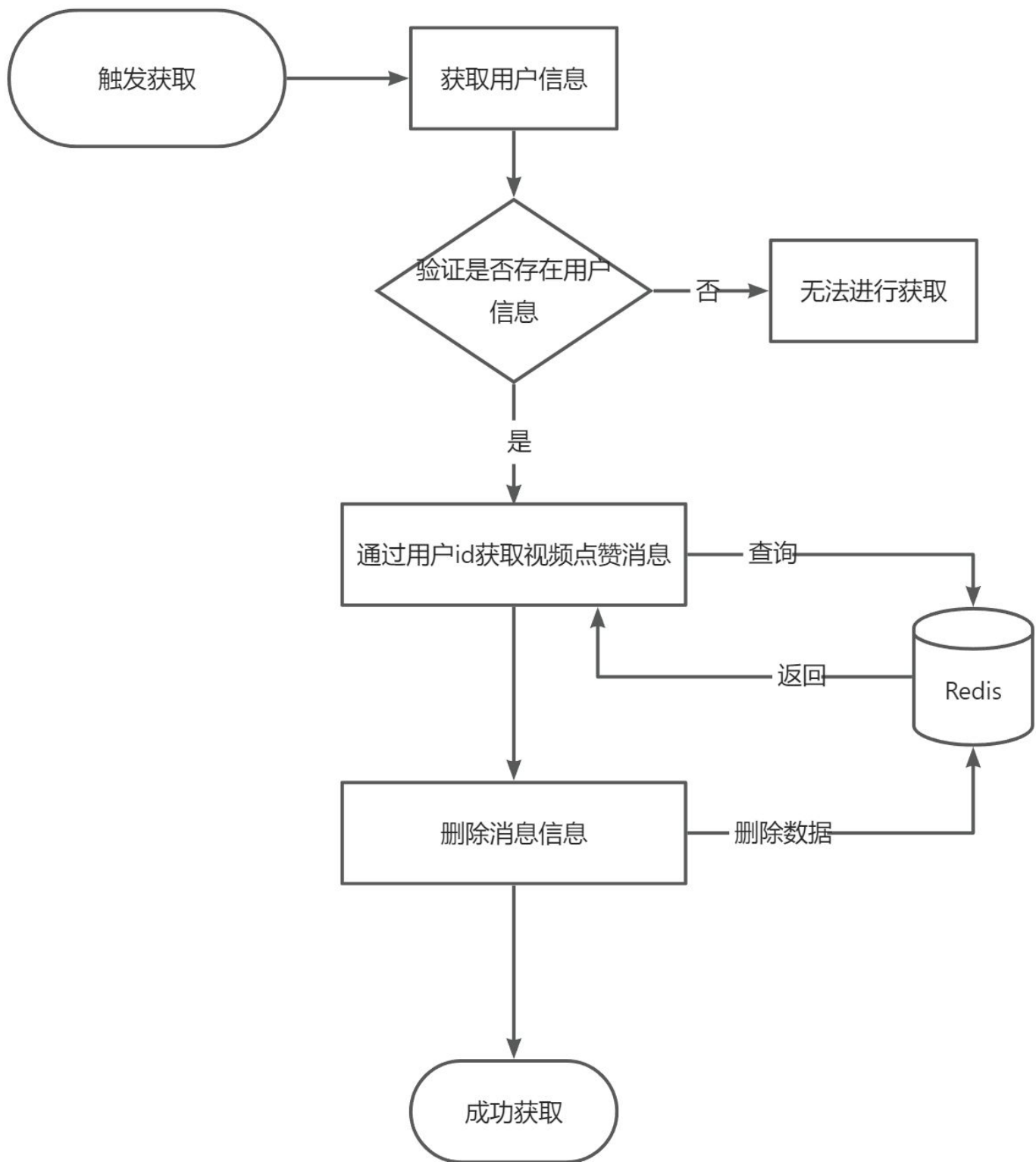


8. 消息通知模块

- 该模块包含业务：消息通知业务
- 业务相关技术：用户验证-SpringSecurity、消息通知-Redis

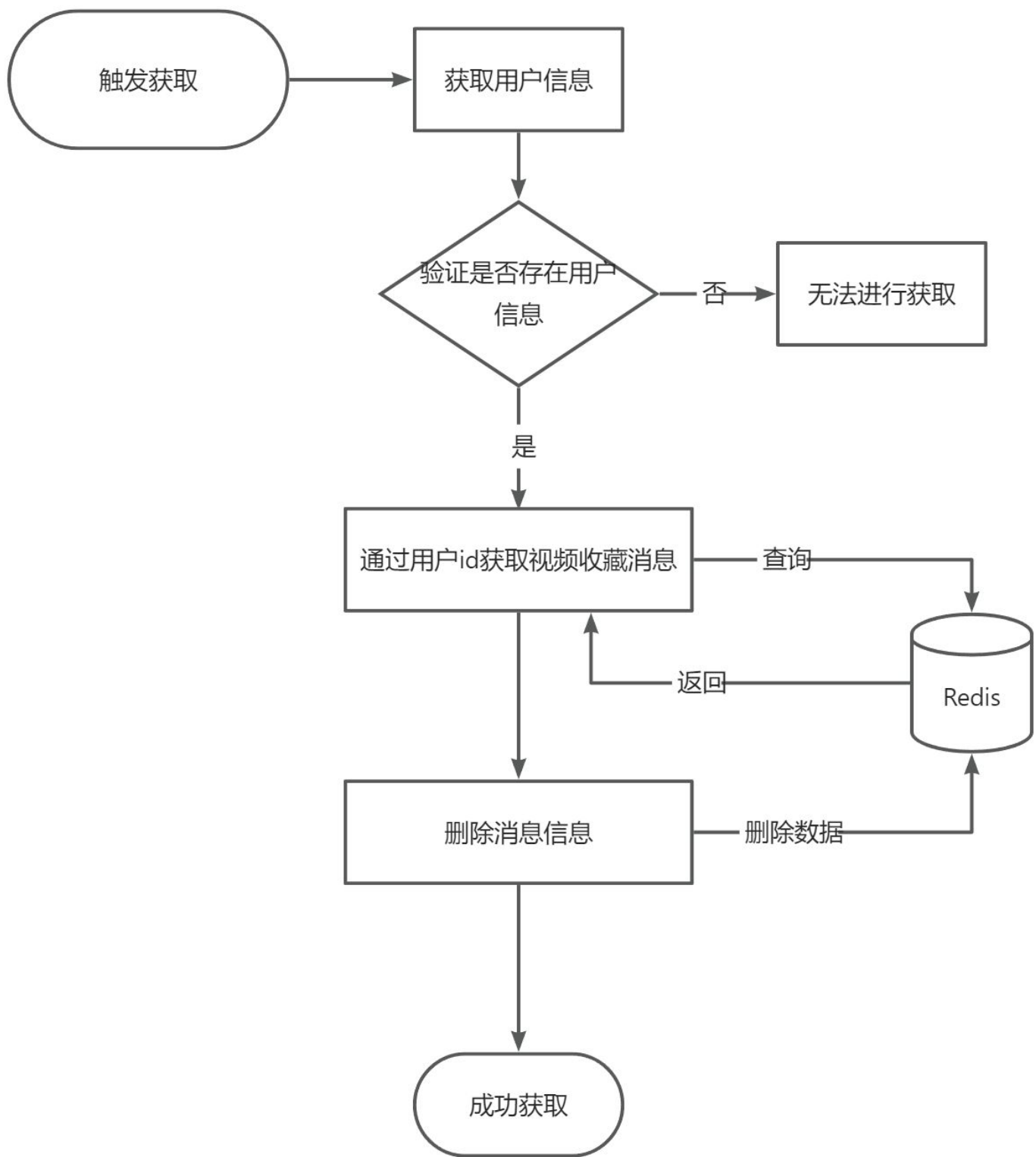
8.1. 获取视频点赞消息

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id进行获取所有视频点赞消息，如果不存在用户信息则无法进行获取
3. 获取后在redis中删除消息信息
4. 返回点赞消息信息



8.2. 获取视频收藏消息

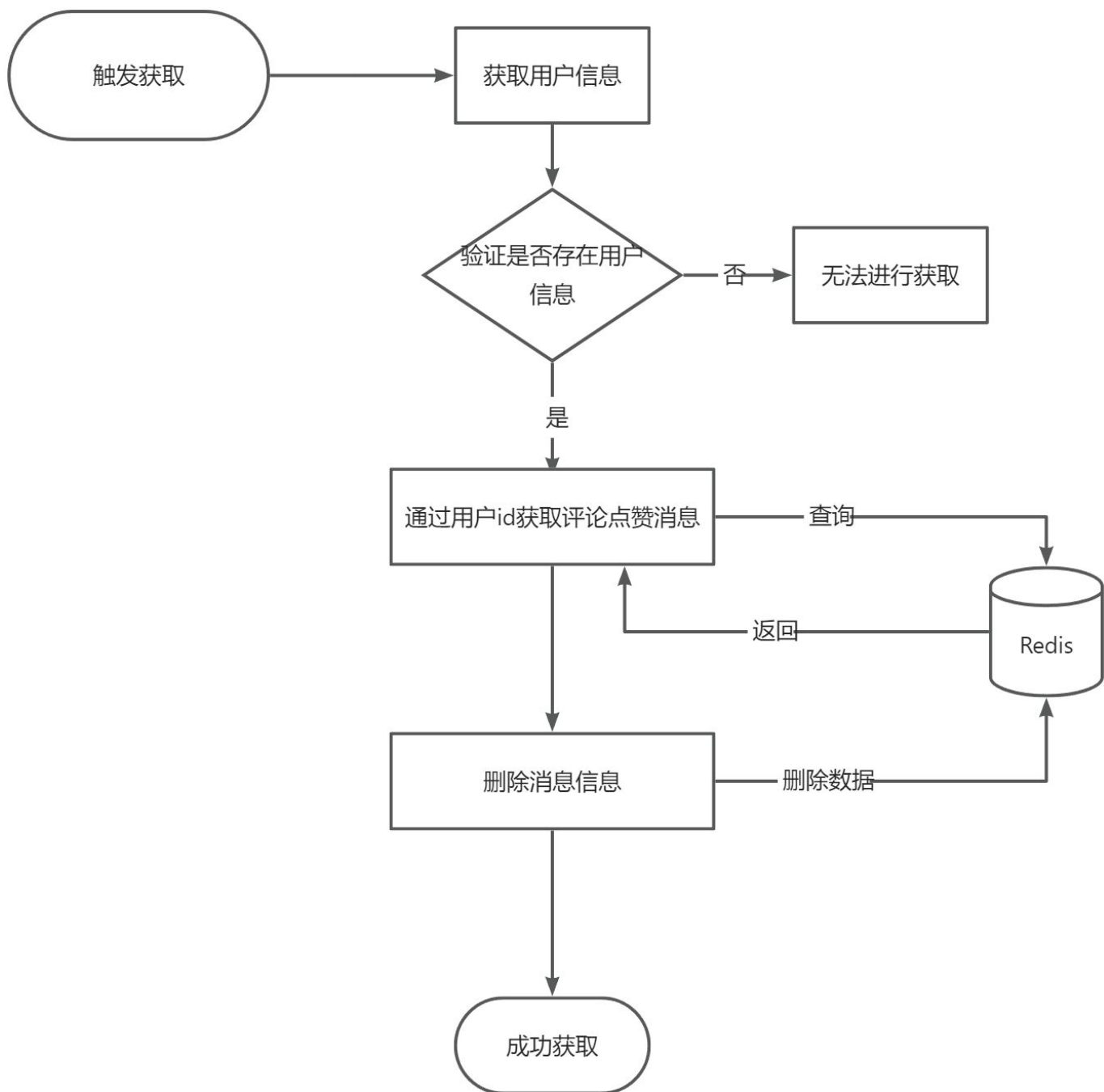
1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id进行获取所有视频收藏消息，如果不存在用户信息则无法进行获取
3. 获取后在redis中删除消息信息
4. 返回收藏消息信息



8.3. 获取评论点赞消息

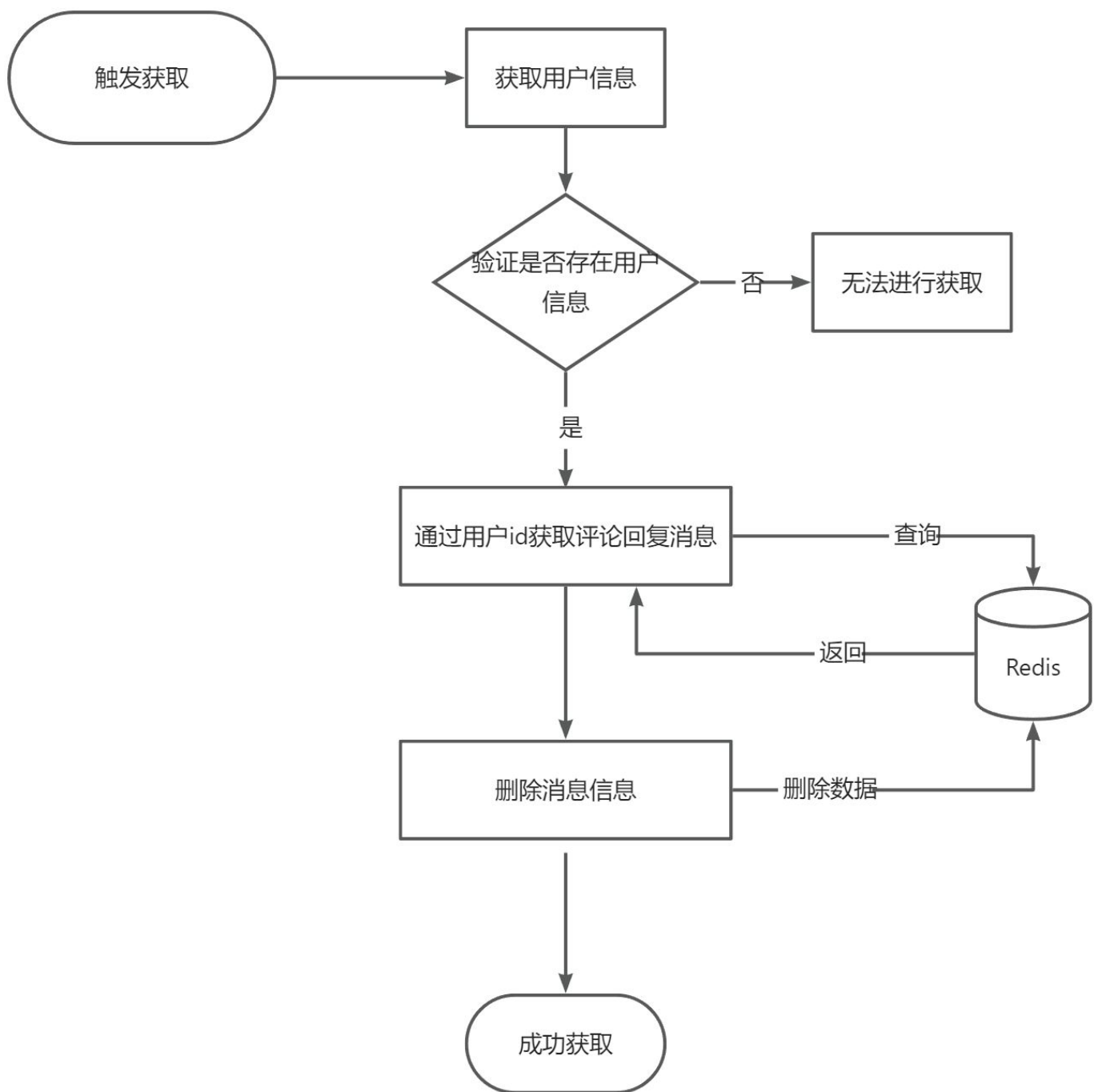
1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id进行获取所有评论点赞消息，如果不存在用户信息则无法进行获取
3. 获取后在redis中删除消息信息

4. 返回评论点赞消息信息



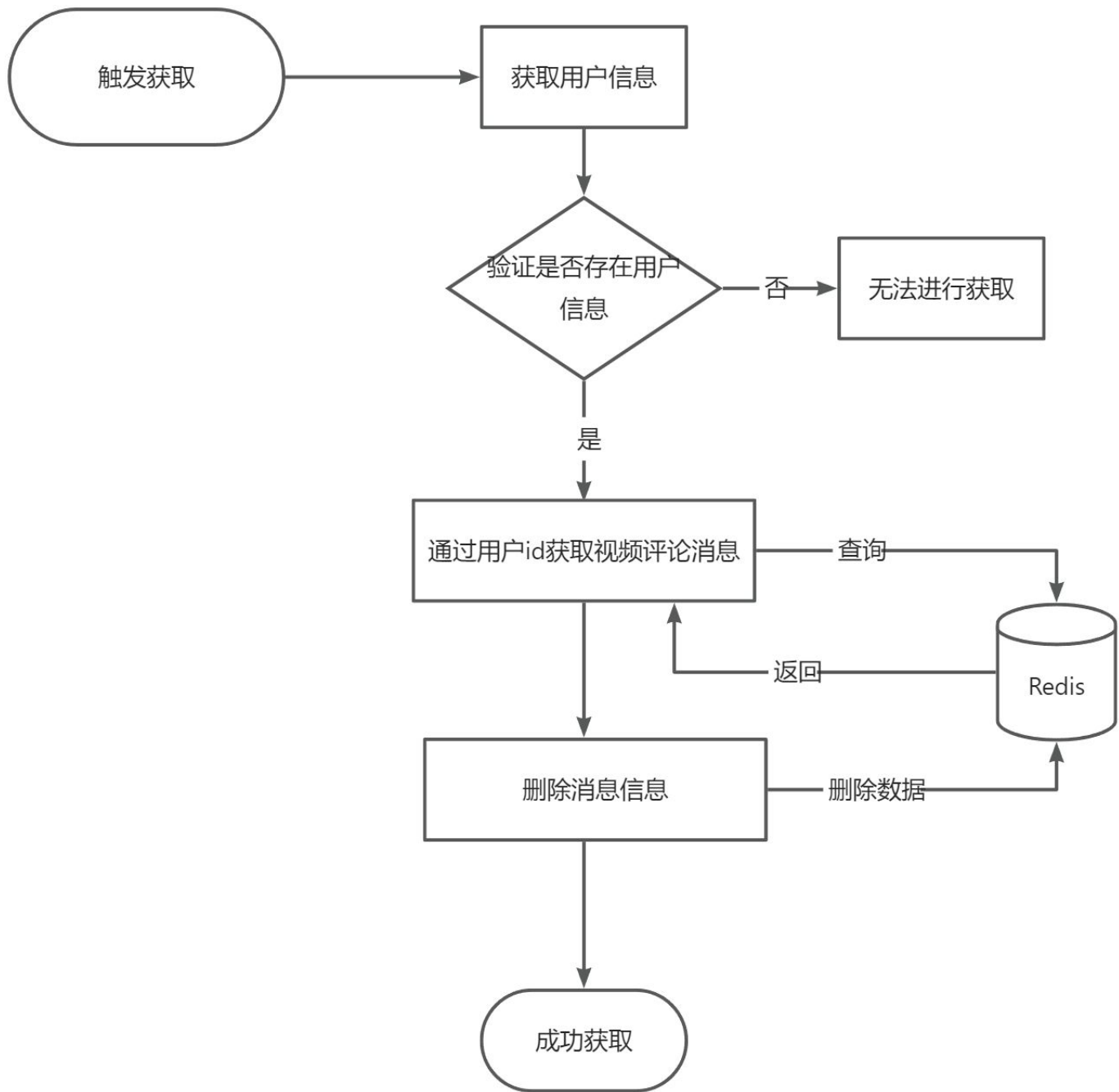
8.4. 获取被回复的消息

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id进行获取所有评论回复消息，如果不存在用户信息则无法进行获取
3. 获取后在redis中删除消息信息
4. 返回评论回复消息信息



8.5. 获取视频被评论的消息

1. 验证是否登录，通过SpringSecurity验证token获取用户信息
2. 如果存在用户信息，通过用户id进行获取所有视频评论消息，如果不存在用户信息则无法进行获取
3. 获取后在redis中删除消息信息
4. 返回视频评论消息信息



三、人员安排

人员	负责模块
李君祥	数据库创建，用户登录鉴权，用户关注模块编写，视频上传模块编写，视频搜索模块编写 视频标签模块编写，视频推荐模块编写

孙铭杰	数据库表设计，视频点赞收藏模块编写，视频评论模块编写，消息通知模块编写
陈士博	前端架构设计，前端页面设计，接口开发