乐拍视界项目文档

项目小组 第 2 小组

小组成员 柏翔 刘金林 兰寅银 朱灿银 周俊

联系方式 17748752006

重庆师范大学软件工程系

摘要

“乐拍视界”项目旨在解决年轻群体创作音乐短视频时操作繁琐、社交孤立的痛点，通过打造一个集智能拍摄、海量配乐、一键美化与沉浸式社交于一体的移动平台，开创短音乐视频新赛道。项目计划以 6-12 个月完成开发，核心是整合音乐库、智能算法与推荐系统，旨在通过极致流畅的一体化体验，快速吸引用户并构建活跃社区，最终成为引领年轻文化的领先平台。

日期	修改	描述	作者
2025 年 11 月 4 日	0.1	完成了立项和愿景	柏翔 刘金林 兰寅银 朱灿银 周俊
2025 年 11 月 20 日	0.2	完成了用况建模	兰寅银
2025 年 12 月 1 日	0.3	完成了需求分析	兰寅银

目录

摘要.....2

第 1 章 立项.....9

1.1. 项目起源与提案.....9

1. 发现问题.....9

2. 根源分析.....9

3. 系统边界定义.....9

4. 约束确定.....9

5. 提案构想.....9

1.2. Business Case.....10

1. 摘要.....10

2. 市场机遇.....10

3. 目标市场与客户细分.....11

4. 竞争优势.....11

5. 市场营销与用户获取策略.....11

6. 风险与应对.....11

7. 成本估算.....12

8. 项目目标.....12

第 2 章 愿景.....13

2.1. 问题陈述.....13

1. 问题一.....13

2. 问题二.....13

3. 问题三.....14

2.2. 涉众与用户.....14

1. 涉众.....14

2. 用户.....16

2.3. 关键涉众和用户的需要.....16

2.4. 产品概述.....19

1. 产品定位陈述.....19

2. 完整的产品概述.....19

2.1. 能力概述.....19

2.2. 客户效益（Benefits）.....20

2.3. 假设和依赖.....20

2.4. 取舍和竞争.....21

2.5. 产品特性.....21

1. 内容消费者.....21

2. 内容创作者……22
3. 管理类……23
4. 业务处理类……24
4.1. 审核人员……24
4.2. 客服……25
2.6. 其他产品需求……25
2.7. 特性优先级……26
2.8.补充说明……28
1. 法律 / 规定性需求……28
2. 应用开发标准……28
3. 质量属性需求……29
4. 设计与实现约束……29
5. 其他补充需求……29
第 3 章 用况建模……30
3.1. 术语表……30
3.2. 参与者清单……31
1. 识别主参与者……31
2. 参与者简要描述……31
2.1. 普通用户……31
2.2. 内容审核员……31
2.3. 客服……31
2.4. 运营人员……31
3.3. 乐拍视界的普通用户主要用况……31
1. 浏览个性化视频流用况……32
1.1. 简要描述……32
1.2. 前置条件……32
1.3. 后置条件……32
1.4. 基本事件流……32
1.5. 备选流……33
3.4. 乐拍视界的内容审核员主要用况……35
1. 审核用户发布内容用况……36
1.1. 简要描述……36
1.2. 前置条件……36
1.3. 后置条件……36
1.4. 基本事件流……36
1.5. 备选流……37
2. 管理内容审核规则用况……38

2.1. 简要描述.....38
2.2. 前置条件.....38
2.3. 后置条件.....38
2.4. 基本事件流.....38
2.5. 备选流.....39
3.5. 乐拍视界的客服主要用况.....40
1. 处理用户咨询与投诉用况.....40
1.1. 简要描述.....40
1.2. 前置条件.....40
1.3. 后置条件.....40
1.4. 基本事件流.....40
1.5. 备选流.....41
2. 管理客服知识库用况.....42
2.1. 简要描述.....42
2.2. 前置条件.....42
2.3. 后置条件.....42
2.4. 基本事件流.....42
2.5. 备选流.....43
3.6. 乐拍视界的运营人员主要用况.....44
1. 策划并执行运营活动用况.....44
1.1. 简要描述.....44
1.2. 前置条件.....44
1.3. 后置条件.....45
1.4. 基本事件流.....45
1.5. 备选流.....45
2. 管理内容推荐策略用况.....46
2.1. 简要描述.....46
2.2. 前置条件.....46
2.3. 后置条件.....46
2.4. 基本事件流.....46
2.5. 备选流.....47
3. 监控平台运营数据用况.....48
3.1. 简要描述.....48
3.2. 前置条件.....48
3.3. 后置条件.....48
3.4. 基本事件流.....48
3.5. 备选流.....49

第 4 章 需求分析.....50
4.1. 分析模型概述.....50
1. 分析模型的目标与关注点.....50
2. 分析模型的组成.....50
3. 用况的实现（Realization）.....50
4.2. 健壮性分析.....51
1. 边界类.....51
2. 实体类.....52
3. 控制类.....52
4.3. 交互建模.....53
1. 浏览个性化视频.....53
1.1. 参与对象（协作角色）.....53
1.2. 通讯图.....54
2. 审核视频.....55
2.1. 参与对象（协作角色）.....55
2.2. 通讯图.....55
3. 创建发布活动.....56
3.1. 参与对象（协作角色）.....56
3.2. 通讯图.....56
4.4. 职责分配与 CRC 卡.....57
1. 浏览个性化视频相关类.....57
1.1. Class Name: FeedController (控制类).....57
1.2. Class Name: RecommendationModel (实体类).....57
1.3. Class Name: Video (实体类).....58
1.4. Class Name: User (实体类).....58
2. 审核用户发布内容相关类.....58
2.1. Class Name: AuditController (控制类).....58
2.2. Class Name: AuditTask (实体类).....58
3. 策划并执行运营活动相关类.....59
3.1. Class Name: ActivityController (控制类).....59
3.2. Class Name: OperationActivity (实体类).....59
4.5. 整合分析类图.....59
1. 实体类 (Entity Classes).....59
1.1. Video (短视频).....59
1.2. User (用户).....60
1.3. Music (音乐).....60
1.4. AuditTask (审核任务).....60
1.5. OperationActivity (运营活动).....61

2. 控制类 (Control Classes)....61
3. 边界类 (Boundary Classes)....62
4. 类关联关系....62
4.1. 用户与视频 (发布关系)....62
4.2. 视频与音乐 (使用关系)....62
4.3. 审核任务与视频 (审核对象关系)....63
4.4. 审核任务与用户 (执行关系)....63
4.5. 运营活动与视频 (聚合关系)....64

第 5 章 架构设计....65
5.1. 设计目标与策略....65
1. 设计目标与优先级....65
1.1. 高性能与响应速度....65
1.2. 可用性与易用性....65
1.3. 可扩展性与伸缩性....65
1.4. 可靠性与稳定性....66
1.5. 可维护性....66
1.6. 安全性与合规性....66
2. 设计权衡....67
2.1. 性能 vs. 成本....67
2.2. 可用性 vs. 一致性....67
2.3. 结构化分层 vs. 运行效率....68
2.4. 客户端富交互 vs. 瘦客户端....68
2.5. 安全性 vs. 用户体验....69
3. 技术平台选择....69
3.1. 客户端平台....69
3.2. 服务端平台....69
3.3. 数据管理平台....70
3.4. 通信与中间件....70
5.2. 系统的架构风格....71
1. 总体架构模式....71
1.1. 表现层....71
1.2. 应用逻辑层....71
1.3. 领域层....72
1.4. 数据源层....72
2. 分层策略....72
2.1. 封闭分层的实施规则....73
2.2. 策略选择的理由....73

2.3. 对性能损耗的应对.....73
3. 通信风格.....74
3.1. 客户端-服务器风格.....74
3.2. 基于中介的异步通信风格.....74
3.3. 内部服务通信.....75
5.3. 子系统分解与逻辑视图.....75
1. 子系统划分.....75
1.1. 服务端子系统.....75
1.2. 客户端子系统.....76
1.3. 子系统间的依赖关系.....77
2. 客户端架构：MVC 模式应用.....77
2.1. 模型.....77
2.2. 视图.....78
2.3. 控制器.....78
2.4. 传播机制.....79
3. 服务端架构.....79
3.1. 应用逻辑组件.....79
3.2. 领域模型组件.....80
5.4. 并发设计与进程视图.....81
1. 并发需求识别.....81
2. 进程与线程管理.....81
3. 同步与通信.....81
5.5. 物理架构与部署视图.....81
1. 硬件节点映射.....81
2. 软件组件分配.....82
3. 网络拓扑与通信基础设施.....82
5.6. 数据管理策略.....82
1. 持久化存储方案.....82
2. 数据库管理系统选择.....82
5.7. 系统设计标准与规范.....82
1. 用户接口标准.....82
2. 内部接口与编码规范.....82
第 6 章 详细设计.....83
后记.....84
参考文献.....85

第1章 立项

1.1. 项目起源与提案

1. 发现问题

在当前的移动互联网环境下，我们观察到青少年群体中兴起了一种自发的、富有创造力的内容创作模式：他们使用手机或数码相机录制生活中的精彩片段（如舞蹈、滑板、搞笑短剧等），然后借助另一台设备（如电脑、MP3 播放器或另一部手机）播放背景音乐，通过后期剪辑或直接跟拍的方式，将画面与音乐结合，最终将作品上传到论坛、博客或早期视频网站进行分享。这个过程虽然充满热情，但存在明显的技术断层和体验割裂：

操作繁琐： 用户需要在不同设备间切换，涉及文件传输、音画对齐等复杂步骤，创作门槛高。

即时性差： 无法实现“即想即拍、即拍即享”，灵感与创作冲动在繁琐的流程中被消耗。

社交孤立： 分享渠道分散，缺乏一个专注于此类短音乐视频的平台，创作者难以找到同好，无法形成有效的互动和反馈闭环。

2. 根源分析

工具层面：缺乏一体化移动端创作工具，音画对齐、特效添加需跨设备操作

平台层面：内容分发依赖主动搜索，无法精准匹配用户兴趣

生态层面：观看与创作场景分离，互动形式浅层，缺乏闭环激励

3. 系统边界定义

内部系统：创作工具模块、推荐算法模块、社交互动模块、内容审核模块、运营管理模块

外部交互：音乐版权方接口、云服务供应商、应用商店、电信运营商、监管机构数据上报通道

4. 约束确定

法律约束：需遵守音乐版权法规、网络内容合规要求、用户数据隐私保护条例

技术约束：适配 Android 7.0 及以上版本，支持主流机型硬件特性

资源约束：初期开发团队 5 人，首年运营成本控制在 2100 万 - 4800 万

5. 提案构想

因此，我们提议开发“乐拍视界”——一款真正实现视频拍摄、智能配乐、一键美化、无缝社交一体化的移动应用程序。

对于视频拍摄者： 我们将提供海量正版热门音乐库，用户可以在拍摄前或拍摄后轻松选择配乐；应用内置智能节拍识别功能，能自动将视频画面与音乐高潮点对齐；提供多种电影级的滤镜和转场特效，让零基础的普通用户也能在几分钟内创作出酷炫的、富有感染力的音乐短视频。

对于内容消费者与社交分享者，我们致力于打造一个以音乐为纽带、以视频为载体的沉浸式社交平台。平台将用户从传统的“搜索—观看”单向模式中解放出来，转向“发现—互动—再创作”的闭环体验，构建一个真正“懂你”的音乐视频互动社区：

在内容层面，我们通过强大的个性化推荐算法，实现从“人找内容”到“内容找人”的转变。系统会根据用户的观看偏好、互动行为与音乐口味，智能推送感兴趣的视频，降低选择成本，提升沉浸感。

在社交层面，我们重视人与人之间的连接与互动。用户可通过“同款音乐”功能进行创意比拼，通过点赞、评论、分享和关注等行为，与其他创作者建立联系。形成一个紧密的、充满活力的创意社群。这些互动能够促进内容的流动与传播。

1.2. Business Case

1. 摘要

该项目瞄准 15-30 岁的年轻群体，旨在解决其制作高质量、富有表现力的音乐短视频时面临的“操作复杂、社交低效”的核心痛点。通过将专业的视频拍摄、庞大的音乐库、智能的剪辑工具与强大的社交功能无缝整合，“乐拍视界”将开创“移动短音乐视频社交”这一全新赛道。我们预期通过快速获取用户、构建内容生态，并最终通过广告、虚拟商品、直播和品牌合作等多种方式实现盈利，占据市场领导地位。

2. 市场机遇

移动设备普及： 智能手机性能不断提升，网络开始普及，为移动端视频的拍摄、上传和播放提供了硬件基础。

用户行为变迁： 年轻一代是“视觉系”原住民，他们更倾向于用图像和视频而非文字进行表达和社交。

音乐需求旺盛： 音乐是年轻人表达情绪、彰显个性的重要载体，与视频结合拥有巨大的想象空间。

市场空白： 现有社交平台（如微博、QQ 空间）或视频平台（如优酷、Youtube）均未提供专为“短音乐视频”设计的、便捷的观看，创作与社交一体化体验。

3. 目标市场与客户细分

核心用户： 15-25 岁的学生和年轻白领。他们追求潮流、乐于表达、渴望认同、拥有强烈的社交需求。

次要用户： 25-30 岁的都市青年，以及有记录和分享孩子成长瞬间需求的年轻父母。

潜在用户： 寻求与年轻消费者建立连接的品牌方、广告主，以及希望通过内容创作获得影响力的创作者/KOL。

4. 竞争优势

产品体验优势： 打开即播放，单列信息流，上下滑动切换的模式让观看视频简单轻松。提供低门槛的创作体验，将海量正版音乐库、一键式美颜滤镜、丰富的特效模板整合进 App。用户不需要专业技巧，就能轻松创作出看起来“很酷”的视频。一体化产品体验，构建了从发现、音乐选择、特效拍摄、一键发布到互动分享的极致流畅闭环，其无缝体验显著优于功能割裂的传统音视频工具。

技术与算法优势： 该产品采用基于深度学习的推荐算法，能通过用户极短时间的观看行为（完播率、点赞、评论、转发、停留时长等），精准地建模用户兴趣，并实时调整推荐内容。并且可以给用户在拍摄时提供滤镜、美颜、贴纸、背景、时间特效（如慢动作、快动作）等功能。

5. 市场营销与用户获取策略

冷启动：从艺术院校、舞蹈社团、街舞爱好者等垂直社群切入，邀请种子用户，生产高质量内容。

校园推广：与全国高校合作，举办“校园音乐视频大赛”，快速在目标人群中建立知名度。

线上营销：在微博、贴吧、QQ 空间等年轻人聚集的社交平台进行内容投放和话题炒作。

6. 风险与应对

竞争风险：其他软件的模仿与跟进。对策：以快速迭代产品，不断更新技术，提升用户体验为基础，花重金提高产品知名度，拓展商业合作，最终实现市场上的稳固地位。

版权风险：音乐版权纠纷。对策：初期与音乐版权代理商建立正规合作，后期建立自己的版权库。

内容风险：用户上传违规内容。对策：建立“AI 算法+人工审核”的内容审核机制，确保内容健康。

盈利风险：无法高效盈利。对策：谨慎探索多元化的商业模式，优先保障用户体验。

7. 成本估算

一次性开发成本：

开发团队 5 人，打造一个基础版本约需 6-12 个月，按平均月薪 1 万计算，1 个月约 5 万。初期用户量少，采用云服务（如阿里云、腾讯云）。视频存储和带宽是主要开销，初期每月约 1-3 万。最低开发成本最低为 36 万，最高为 96 万。

持续运营成本：

团队扩充至 30-40 人，新增运营、市场、算法、审核等岗位，年度人力成本约 800 万-1200 万。

带宽与服务器成本（随用户量指数增长）：若日活达到百万级，月度带宽成本可能达到 50 万-200 万，年度 600 万-2400 万。

市场营销费用：用户获取与品牌建设，预估 500 万-1000 万。

音乐版权与内容审核：预估 1 年 200 万。

首年持续运营总成本预估：2100 万- 4800 万人民币。

8. 项目目标

短期目标（6-12 个月）： 成功上线 Android 版本，积累首批核心种子用户，建立活跃的创作者社区，实现每日用户创作视频量过万。

中期目标（1-2 年）： 成为国内青少年群体中最受欢迎的短音乐视频社交平台，形成独特的社区文化，探索初步的商业化模式。

长期目标（3-5 年）： 构建以“乐拍视界”为核心的短视频内容生态，成为引领年轻文化潮流的重要阵地，并拓展至海外市场。

第2章愿景

2.1. 问题陈述

1. 问题一

要素	描述
问题	普通用户的创作过程割裂且技术门槛高，缺乏一个集成化工具，能够让他们在移动端轻松、快速地将视频与热门音乐结合，并添加专业特效的创作。现有流程依赖多个割裂的设备和复杂软件，操作繁琐。
影响	创作者、平台内容生态
结果	用户旺盛的创作热情和灵感被技术难题所压制。繁琐的流程导致用户从“灵感”到“作品”的转化率极低，大量创意被浪费。平台内容生态依赖少数技术娴熟的创作者，内容风格单一，数量增长缓慢。绝大多数潜在创作者保持沉默，无法形成百花齐放的社区氛围。
优点	该产品提供了一体化移动端创作工具，内嵌海量正版音乐库、智能剪辑功能和丰富特效。 实现“所想即所得”，用户只需选择音乐和片段，算法自动完成音画对齐与节奏匹配，将创作门槛降至最低。 激发创作欲望，让每个人都能轻松制作出酷炫的音乐短视频，从而极大地丰富了平台内容的多样性和数量。

2. 问题二

要素	描述
问题	内容发现效率低下，用户留存困难。传统平台依赖用户主动搜索和订阅，这是一种“人找内容”的低效模式，无法根据用户的潜在兴趣进行精准内容推荐，导致用户陷入选择疲劳。
影响	平台运营方，用户
结果	用户难以持续发现感兴趣的内容，浏览体验枯燥。平台无法为用户创造“上瘾”的沉浸感，导致用户使用时长短、流失率高。 平台活跃度增长陷入瓶颈，无法让用户稳定留在平台。即使有优质内容，也无法被

	对的人看到，内容价值无法最大化。
优点	引入基于 AI 的个性化推荐引擎，通过分析用户行为，精准推送其可能感兴趣的内容，从“人找内容”变为“内容找人”。 打造全屏沉浸式信息流，提供无缝、零思考的浏览体验，最大化用户的专注度和停留时长。

3. 问题三

要素	描述
问题	社交互动薄弱，从观看到创作的转化路径断裂。现有平台的社交互动停留在浅层的点赞和评论，且观看与创作是两个完全割裂的场景。用户即使被视频激发起创作欲望，也缺乏无缝、低成本的创作方式。
影响	用户、平台生态
结果	用户仅是被动的内容消费者，难以深度融入社区。创作灵感因复杂的转化路径而转瞬即逝，平台无法将高涨的观看情绪有效转化为创作行为。 平台里没有热闹的交流感，更像用户各自刷内容的 “冷清空间”，用户之间没形成关联。内容生态缺乏自生长的动力，需要持续的外部刺激来维持内容产出，运营成本高。
优点	构建以“同款音乐”和“挑战赛”为核心的互动机制，用户可一键使用同款模板参与创作，实现“看了就想拍”。 深度整合社交功能，如好友动态、合拍等，强化用户之间的创意互动与联系。 形成“观看-灵感-创作-互动”的完美闭环，驱动内容的自我繁荣。

2.2. 涉众与用户

1. 涉众

涉众	涉众类型	简要描述
项目经理	开发团队	制定项目计划，管理进度、风险和资源，保证项目按期交付
测试人员	开发团队	测试系统的功能、性能、数据安全

需求分析师	开发团队	与用户沟通，获取需求和想法，将这些需求转化为详细的需求规格说明书。
编码员	开发团队	负责编码实现系统
审核人员	平台维护人员	审核用户发布的作品，删除违规的作品并提醒作者,处理违规用户
客服	平台维护人员	解答用户的问题
官方账号运营团队	平台维护人员	运营平台的官方账号，通过发布内容、策划线上活动（如挑战赛）来激发社区活力，提升用户粘性
国家互联网信息办公室	监管机构	负责互联网信息内容的管理，落实互联网信息传播的方针政策，指导、协调和督促有关部门加强网络内容管理，并依法查处违法违规网站和应用。
工业和信息化部	监管机构	负责电信与互联网行业的管理，监管范围包括网络基础设施、信息服务业务许可、网络与信息安全技术标准等。
国家版权局	监管机构	负责监管平台上的版权问题，处理用户上传内容可能涉及的音乐、视频、文字等版权侵权纠纷。
公安部	监管机构	负责网络安全保卫工作，打击利用平台进行的网络诈骗、传播违法信息、侵犯公民个人信息等违法犯罪活动。
用户	用户	观看或发布音视频
音乐版权方	版权方	提供音乐的版权，防止音乐作品在未经授权的情况下被平台用户使用
应用商店	合作伙伴	审核应用资质，确保应用上架符合相关法律法规和平台规定
电信运营商	合作伙伴	提供网络连接
云服务商	合作伙伴	提供云服务

品牌方	合作伙伴	提供钱让平台发布广告，他们是平台未来实现流量变现、进行商业合作的关键对象。
投资者	发起人	资金提供者，关注投资回报

2. 用户

用户	用户类型	简要描述
观看者	内容消费者	寻求轻松、有趣的娱乐方式，打发碎片时间；通过浏览内容获得放松。 或为获取知识、技能、资讯，主动搜索或关注科普、技能、资讯类内容，追求内容的专业性、实用性和可学习性。
官方机构	内容创作者	宣传活动，发布热点视频吸引流量代表平台运营官方账号，通过策划宣传活动、发布热点视频、输出平台动态等方式吸引流量、传递品牌价值、维护用户关系。
博主	内容创作者	渴望进行自我表达，获得关注和认同；需要低门槛、高效的创作工具；希望通过分享日常、生活技巧等内容，与同好交流，积累粉丝。
运营团队	管理类	负责内容生态搭建、创作者扶持与管理，统筹内容审核、流量运营及用户维护，保障平台有序运转与持续增长。
审核人员	业务处理类	对平台内容、用户行为进行合规性审核，依据平台规则筛查违规信息，保障平台内容生态的健康、安全与合规。
客服	业务处理类	及时解答用户在平台使用过程中的疑问、投诉与建议，处理用户诉求，提升用户使用体验与满意度。

2.3. 关键涉众和用户的需要

关键涉众	需要
投资者	1. 商业回报：清晰的盈利路径（如广告、虚拟商品、电商）和投资回报率。 2. 增长潜力：高速的用户增长、市场占有率及平台网络效应的形成。 3. 竞争壁垒：产品相较于潜在竞争者的独特优势和可持续性（如技术、社区文化）。
音乐版权方	1. 版权保护与收益：其音乐作品被合法使用，并能获得公平的版权分成。 2. 作品推广：平台能成为其新歌、新艺人推广的有效渠道，扩大音乐影响力。 3. 数据洞察：获得其音乐在平台上的使用数据（如播放量、热门视频等），以指导宣发。
政府监管机构	1. 内容合规：平台内容符合国家安全、社会公序良俗及青少年保护法规。 2. 数据安全：用户数据（尤其是未成年人数据）的收集、使用符合相关法律要求。 3. 协同治理：平台能积极配合监管要求，建立有效的自查与清理机制。

关键用户	需要
观看者	1. 个性化娱乐：平台能“懂我”，通过精准算法持续提供我感兴趣的内容，带来轻松愉悦的“杀时间”体验。 2. 社交归属感：能关注喜欢的创作者，与同好互动（点赞、评论），感觉自己身处一个有趣的社区。 3. 内容新鲜度：总能发现新的潮流、热门话题和创意形式，保持对平台的新鲜感和期待。

	4. 精准获取内容：通过关键词搜索、标签分类快速找到目标内容，过滤无效信息，高效获得想要的内容。 5. 内容权威性：内容来源可信、逻辑清晰，有实操步骤或权威依据，能真正了解世界，扩充自己。 6. 视频体系化：支持内容收藏、合集查看，方便按主题连贯观看。
博主	1. 低门槛表达：提供简单易用、功能强大的创作工具（音乐库、特效、模板），让创意能轻松实现。 2. 获得认可与影响力：获得粉丝、点赞、评论等社交正反馈，积累个人影响力，满足表达欲和成就感。 3. 创作灵感启发：能方便地追踪热门挑战和流行趋势，获得持续创作的灵感和动力。 4.提高粉丝留存度：增加账户热度。 5.保证粉丝活跃度：扩大账户知名度。
运营团队	1. 生态健康运转：通过数据工具监控内容质量、用户行为，及时调整运营策略，维持正向生态。 2. 创作者扶持：搭建分层扶持体系，提供流量倾斜、工具特权等资源，激励优质创作者留存。 3. 增长与转化：策划热点活动、优化流量分发机制，提升用户活跃度、留存率及商业转化效率。
官方机构	1. 高效宣传：借助平台流量优势，让宣传内容精准触达目标受众，提升活动或品牌曝光度。 2. 权威形象传递：通过官方认证标识、合规内容审核支持，保障信息发布的权威性和可信度。 3. 互动与反馈：搭建与用户的沟通渠道，收集公众意见，提升品牌好感度和用户粘性。
审核人员	1. 高效审核工具：提供智能筛查、关键词识别等辅助工具，提升违规内容识别效率，降低工作负荷。

客服	2. 明确审核标准：有清晰、统一的规则手册和更新机制，避免审核判断偏差。 3. 风险预警支持：对高风险内容、敏感话题提前预警，保障审核工作的准确性和及时性。 1. 高效响应工具：整合用户咨询渠道，提供快捷回复模板、问题分类标签，快速处理用户诉求。 2. 问题解决支持：获取平台规则、功能说明等权威资料，能准确解答用户疑问或协调处理复杂问题。 3. 用户反馈同步：建立反馈机制，将用户高频问题、建议同步给相关团队，优化服务体验。

2.4. 产品概述

1. 产品定位陈述

For	渴望表达自我的年轻一代与寻求轻松、愉悦内容消费的移动互联网用户
Who	他们需要一种简单、有趣的方式来创作和分享生活，并渴望获得即时的社交互动与认同。但现有工具创作门槛高，平台内容分发效率低下，观看与创作行为割裂。
The	乐拍视界
That	是一款集音乐短视频创作、智能推荐与沉浸式社交于一体的移动平台。我们通过一体化创作工具和精准的推荐算法，为用户提供零门槛的创意表达和高度个性化的内容消费体验，让每个人都能轻松记录和分享生活中的乐趣。
Ulike	不同于功能复杂、以长视频和搜索为核心的 YouTube，也不同于功能单一、缺乏社交生态的 Dubsmash。
Our product	我们提供了从创意激发、极简创作到精准分发与互动的完整闭环，构建了一个充满活力的创意社区。

2. 完整的产品概述

2.1. 能力概述

乐拍视界作为移动端短音乐视频社交应用，核心向用户提供三大核心能力。

创作方面，通过内置海量正版音乐库、智能剪辑工具、美颜滤镜、动态贴纸等功能，以及“一键出片” 模板，实现低门槛视频创作与美化；

消费方面，以全屏上下滑动的沉浸式信息流为载体，通过个性化推荐算法，为用户推送定制化内容流；

社交方面，借助点赞、评论、关注、分享、私信等功能。构建以音乐为核心的创作互动与灵感交流闭环，覆盖“创作-消费- 社交”全场景需求。

2.2. 客户效益（Benefits）

涉众类型	核心效益	对应产品特性
创作者	降低创意表达门槛，快速实现创作想法	智能音乐匹配、简易拍摄美化、创意特效库、“一键出片” 模板
	获得社交认同与灵感启发，提升创作动力	“拍同款” 功能、同款音乐聚合页
消费者	高效获取感兴趣的内容，获得沉浸娱乐体验	个性化推荐算法、全屏沉浸式信息流
	发现潮流内容与同好，增强社区归属感	互动功能
品牌 / 运营方	实现品牌宣传与商业收益转化	直播带货、广告投放系统

2.3. 假设和依赖

核心假设

1. 用户对短音乐视频的创作需求持续存在，且倾向于 “低操作成本、高创意呈现”的创作模式。

2. 个性化推荐算法能精准捕捉用户兴趣，持续提升用户留存与使用时长。

3. 以音乐为核心的社交互动模式，能有效激发用户参与度，形成稳定的社区生态。

关键依赖

1. 版权依赖：需持续与主流音乐版权方合作，保障曲库的合法性、丰富度与时效性。

2. 技术依赖：依赖智能节拍识别、推荐算法、实时特效渲染等技术的稳定迭代与优

化。

3. 环境依赖：适配主流移动端操作系统（Android），需兼容不同品牌、型号的手机硬件与系统版本。

4. 生态依赖：需吸引足够数量的创作者产出优质内容，同时积累初始用户群体，形成 “创作 - 消费” 的正向循环。

2.4. 取舍和竞争

核心取舍

1. 功能取舍：优先聚焦 “音乐 + 短视频” 核心场景，简化复杂编辑功能，暂时放弃长视频、多场景综合剪辑等非核心能力，确保创作门槛足够低。

2. 内容取舍：侧重年轻化、潮流化、娱乐化内容生态，暂时弱化专业知识。

竞争对比

竞争产品	优势	劣势	本产品差异化优势
快手	下沉市场渗透深、社区氛围浓厚、真实感强	内容精致度不足、潮流属性弱、推荐精准度待提升	信息流更沉浸、创作工具更侧重音乐适配，潮流化内容导向更明确
YouTube	内容生态丰富、长短视频全覆盖、搜索功能强	功能复杂、操作门槛高、社交互动性弱	聚焦移动端短视频，简化操作流程，强化 “创作 - 社交” 闭环，更贴合碎片化使用场景。

2.5. 产品特性

1. 内容消费者

核心需要	对应系统特性
获得个性化娱乐体验，高效 “杀时间”	精准推荐算法：基于用户兴趣与行为，持续推送感兴趣的短视频。
	全屏沉浸体验：上下滑动无缝切换视频，最大化娱乐沉浸感。
融入有趣社区，获得社交归属感	多元互动工具：提供关注、点赞、评论、私信等功能，构建互动社区。

	粉丝团体系：支持用户加入专属粉丝团，增强与创作者的联结。
持续接触新鲜潮流，保持平台新鲜感	热点聚合页面：实时更新热门话题与挑战，一站式追踪全网潮流。
	创意模板库：提供海量同款音乐、特效和拍摄模板，降低参与门槛。
高效获取实用知识或技能，解决实际问题	垂直知识库：按用途、技能等分类聚合深度内容。
	精准搜索筛选：支持关键词搜索，并可按最新、最热等维度筛选。
	“干货”标识系统：对知识密度高的视频进行标记，便于用户识别。
视频体系化，避免碎片化	合集/列表功能：创作者可将系列内容整理成合集，支持连续观看。
	视频进度跟踪：自动记录在合集中的观看进度，支持断点续看。
	笔记与收藏功能：支持在看视频时记录要点，并分类收藏内容。
辨别信息真伪，获取可靠内容	创作者认证体系：对教育、医学等领域的专业人士进行身份认证。
	事实核查机制：与权威机构合作，对热门技能、知识类视频进行事实标注。
	优质创作者推荐：在相关领域优先推荐经过认证的用户。
激发兴趣，探索未知领域	知识科普话题：设立如“科普一下”等官方话题，降低认知门槛。
	跨领域推荐：在用户原有兴趣基础上，智能推荐关联领域的入门内容。

2. 内容创作者

核心需要	对应系统特性
低门槛实现创意表达，快速完成作品制作	海量正版音乐库：提供分类清晰、一键使用的正版音乐与音效。
	剪辑工具集：添加字幕、添加背景音乐、添加特效、裁剪视频等辅助工具。
	实时美颜与滤镜：提供多档美颜调节与风格化滤镜，提升画面质感。
	“一键出片”模板：提供海量创意模板，替换素材即可快速生成优质视频。
获得社交正反馈，积累粉丝与影响力	实时数据看板：清晰展示作品点赞、评论、转发、涨粉等核心数据。
	粉丝分层管理：支持对粉丝进行分组、标记，实现精细化社群运营。
	私信互动管理：提供高效的私信收发与批量管理功能。
获取创作灵感，紧跟行业趋势	热门挑战榜单：实时展示平台最热门的挑战与话题。
	潮流内容推荐：根据创作者领域个性化推送热门内容与同行佳作。
	同款音乐聚合页：热门 BGM 及使用该音乐的高赞作品集中展示。
提高粉丝留存度与保证粉丝活跃度	直播功能：支持与粉丝实时互动。
	特殊标识回复评论：博主回复粉丝评论带有特殊标识。

3. 管理类

核心需要	对应系统特性
搭建健康内容生态，保障平台合规运转	审核任务管理后台：支持任务批量分配、优先级设置与审核员绩效统计。

核心需要	对应系统特性
扶持优质创作者，提升创作者留存	创作数据分析工具：为创作者提供作品、粉丝、收益等多维度数据分析。
提升用户活跃度与平台增长	热点活动策划工具：支持快速创建、发布和推广线上挑战赛等运营活动。
	用户分层运营模块：基于用户行为标签，实现精准的 Push 与消息触达。
	流量分发调控中心：可对推荐算法策略进行人工干预与权重调整。
优化商业化转化效率	广告投放管理后台：管理广告位库存，监控填充率等核心指标。
	电商转化数据监测：实时跟踪从内容曝光到商品成交的全链路数据。
	品牌合作管理系统：管理合作项目流程，并评估项目 ROI。

4. 业务处理类

4.1. 审核人员

核心需要	对应系统特性
高效完成合规审核，降低工作负荷	智能预审与分发：系统自动预筛并标记高风险内容，提升审核效率。
	批量处理功能：支持对同类低风险内容进行一键通过操作和一键拒绝操作。
	审核工作流引擎：根据内容类型和风险等级自动分配任务队列。
确保审核标准统一，减少判断偏差	实时规则查询手册：提供在线、可搜索的详细审核规则与案例库。
	审核结果抽样复核：系统自动对审核结果进行抽样，由质检

	员进行复核。
及时预警高风险内容，保障审核准确性	敏感话题实时预警：对突发热点事件及相关内容进行自动标记与提权。

4.2. 客服

核心需要	对应系统特性
快速响应用户诉求，提升问题处理效率	智能快捷回复模板：针对常见问题提供标准化回复模板，一键发送。
同步用户反馈，助力产品优化	用户反馈标签化收集：支持为每一条用户反馈打上问题类型与优先级标签。
	反馈数据看板：统计高频问题、用户满意度趋势，并生成周期性报告。

2.6. 其他产品需求

类别	需求描述
性能需求	1. 响应速度： 95%的视频请求应在 1 秒内开始播放。 2. 流畅性： App 主界面滑动及视频播放帧率应稳定在 60fps。 3. 并发能力： 系统需支持百万级日活用户的并发访问与内容上传。
可用性需求	1. 直观易用： 新用户无需教程即可完成首次视频发布。 2. 一致性： 全平台保持统一的 UI/UX 设计语言和交互逻辑。 3. 无障碍： 支持系统级字体大小调整，关键元素有足够的对比度。
可靠性需求	1. 系统稳定性： App 崩溃率低于 0.1%。 2. 数据持久性： 用户数据与创作内容需有 99.9%的可靠性保障。
安全性需求	1. 数据安全： 用户密码加密存储，通信链路全程加密。 2. 内容安全： 具备反垃圾、反作弊、反爬虫机制。 3. 隐私保护： 严格遵循隐私法规，提供隐私设置开关，明确告知数据用途。
兼容性需求	1. 系统版本： 支持 Android 7.0 及以上版本。 2. 机型适配： 适配市场主流品牌及全面屏、刘海屏等特殊屏幕。

2.7. 特性优先级

编号	特性	优先级
1	精准推荐算法： 基于用户兴趣与行为，持续推送感兴趣的短视频。	Must
2	全屏沉浸体验： 上下滑动无缝切换视频。	Must
3	多元互动工具： 提供关注、点赞、评论、私信等功能	Must
4	粉丝团体系： 支持用户加入专属粉丝团。	Should
5	热点聚合页面： 实时展示平台最热门的挑战与话题。	Should
6	创意模板库： 提供海量同款音乐、特效和拍摄模板，降低参与门槛。	Must
7	垂直知识库： 按用途、技能等分类聚合深度内容。	Should
8	精准搜索筛选： 支持关键词搜索，并可按最新、最热等维度筛选。	Must
9	“干货”标识系统： 对知识密度高的视频进行标记，便于用户识别。	Must
10	合集/列表功能： 创作者可将系列内容整理成合集，支持连续观看。	Must
11	观看进度跟踪： 自动记录在合集中的观看进度，支持断点续看。	Must
12	收藏功能： 收藏重要或喜欢的内容，并可分类收藏内容。	Must
13	笔记功能：支持在看视频时记录重要时间戳，为该视频添加对应文本内容。	Could
14	创作者认证体系： 对教育、医学等领域的专业人士进行身份认证。	Must
15	事实核查机制： 与权威机构合作，对热门知识类视频进行事实标注。	Won't
16	优质创作者推荐： 在相关领域优先推荐经过认证的用户。	Could
17	提供正版音乐库：提供分类清晰、一键使用的正版音乐与音效。	Must
18	剪辑工具集： 添加字幕、添加背景音乐、添加特效、裁剪视频等辅	Must

	助工具。	
19	实时美颜与滤镜： 提供多档美颜调节与风格化滤镜，提升画面质感。	Must
20	“一键出片”模板： 提供海量创意模板，替换素材即可快速生成优质视频。	Could
21	实时数据看板： 清晰展示作品点赞、评论、转发、涨粉等核心数据。	Must
22	粉丝分层管理： 支持对粉丝进行分组、标记，实现精细化社群运营。	Must
23	私信互动管理： 提供高效的私信收发与批量管理功能。	Must
24	同款音乐聚合页： 热门 BGM 及使用该音乐的高赞作品集中展示。	Must
25	商品挂载功能： 支持在视频和直播中挂载商品，直接引导销售。	Must
26	直播带货工具集： 提供直播间商品橱窗、优惠券、抽奖等营销工具。	Must
27	品牌合作接单平台： 为创作者与品牌方提供官方、安全的合作对接渠道。	Must
28	广告分成的接口： 创作者参与流量分成，从平台广告收入中获益。	Could
29	智能审核系统： 应用 AI 模型对文本、图像、视频进行违规内容预筛查。	Could
30	审核任务管理后台： 支持任务批量分配、优先级设置与审核员绩效统计。	Must
31	创作者成长体系： 设计等级与权益，对应不同的流量扶持与工具特权。	Should
32	创作数据分析工具： 为创作者提供作品、粉丝、收益等多维度数据分析。	Must
33	流量分发调控中心： 可对推荐算法策略进行人工干预与权重调整。	Must

34	用户分层运营模块： 基于用户行为标签，实现精准的 Push 与消息触达。	Could
35	广告投放管理后台： 管理广告位库存，监控填充率等核心指标。	Should
36	电商转化数据监测： 实时跟踪从内容曝光到商品成交的全链路数据。	Should
37	品牌合作管理系统： 管理合作项目流程，并评估项目投资回报率。	Won’t
38	批量处理功能： 支持对同类低风险内容进行一键通过操作和一键拒绝操作。	Should
39	审核工作流引擎： 根据内容类型和内容标签自动分配任务队列。	Should
40	实时规则查询手册： 提供在线、可搜索的详细审核规则与案例库。	Must
41	审核结果抽样复核： 系统自动对审核结果进行抽样，由质检员进行复核。	Should
42	敏感话题实时预警： 对突发热点事件及相关内容进行标记，并上报。	Could
43	智能快捷回复模板： 针对常见问题提供标准化回复模板，一键发送。	Must
44	用户反馈标签化收集： 支持为每一条用户反馈打上问题类型与优先级标签。	Should
45	反馈数据看板： 统计高频问题、用户满意度趋势，并生成周期性报告。	Should

2.8.补充说明

1. 法律 / 规定性需求

音乐版权：所有配乐需获得版权方合法授权，建立版权分成机制

内容合规：符合《网络安全法》《未成年人保护法》《互联网信息服务管理办法》

数据隐私：遵循个人信息保护相关法规，明确告知用户数据收集与使用用途

2. 应用开发标准

开发遵循 Rational Unified Process（RUP）规范

代码开发符合 Android 应用开发最佳实践

文档编写遵循统一模板，确保可读性与一致性

3. 质量属性需求

可用性：系统全年可用性≥99.9%，故障恢复时间≤1 小时

可靠性：单用户日均崩溃次数≤0.01 次

性能：视频上传速度≥1MB/s（4G 网络环境），页面加载时间≤2 秒

可支持性：提供完整的技术文档，支持远程故障排查

4. 设计与实现约束

技术栈：前端采用 QML 开发，后端采用 C++开发

部署环境：依赖阿里云服务器与存储服务

兼容性：适配市场占有率前 20 的 Android 机型，支持全面屏、刘海屏等特殊屏幕

5. 其他补充需求

闲置处理：用户会话闲置超过 20 秒时，系统自动降低视频清晰度以节省流量；闲置超过 5 分钟，弹出互动提示

未成年人保护：提供青少年模式，过滤不适宜内容，限制使用时长

第3章 用况建模

3.1. 术语表

名称	定义
普通用户	在乐拍视界平台中注册并使用系统功能的个体，可以是内容消费者或内容创作者
内容审核员	负责审核用户发布的内容，确保符合平台内容规范和法律法规的专业人员
客服	负责解答用户问题、处理用户投诉和反馈的专业服务人员
运营人员	负责平台内容生态建设、用户运营和活动策划的管理人员
视频流	系统根据用户偏好实时生成的个性化视频内容序列
同款音乐挑战	基于特定音乐发起的创作活动，鼓励用户使用同一音乐创作不同内容
创作模板	预设的视频拍摄和编辑方案，用户可基于模板快速创作内容
音乐短视频	时长在 15 秒至 3 分钟之间，以音乐为核心元素的短视频内容
基本流	用况执行的正常、预期路径
备选流	基本流之外的变体路径，包括异常处理、可选操作等

3.2. 参与者清单

1. 识别主参与者

用户类型	参与者
内容消费者、内容创作者	普通用户
业务处理者	内容审核员、客服
管理类	运营人员

2. 参与者简要描述

2.1. 普通用户

普通用户是系统的核心使用者，可以在"乐拍视界"中浏览推荐视频流，并在浏览过程中进行创作、社交互动、私信交流等操作。

2.2. 内容审核员

内容审核员负责对用户上传的音乐短视频、评论及互动内容进行合规性审查，依据平台规范对违规内容进行处理，以确保平台内容生态的健康合规与安全。

2.3. 客服

客服负责通过系统接收和处理用户反馈、投诉、咨询等各类问题，与用户沟通并提供解决方案，提升用户体验和满意度。

2.4. 运营人员

运营人员负责平台内容生态建设、用户运营和活动策划，通过运营手段提升用户活跃度、留存率和平台影响力。

图 3-1“乐拍视界”中的参与者

3.3. 乐拍视界的普通用户主要用况

普通用户只有一个主用况：浏览个性化视频流。该用况覆盖用户从启动应用到浏览内容的全过程，包括可能的可选操作（如创作、社交等）。创作音乐短视频、参与社交活动、私信、管理个人资料和内容作为这个用况的备选流。

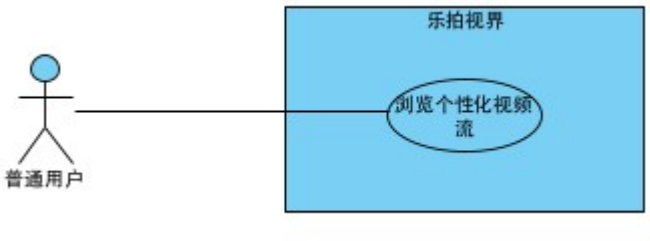

图 3-2“乐拍视界”中的普通用户主要用况

1. 浏览个性化视频流用况

1.1. 简要描述

普通用户浏览系统根据其兴趣偏好推荐的个性化视频流，通过上下滑动切换视频，获得娱乐和信息消费体验。在浏览过程中，用户可选择进行创作、社交互动、私信或管理个人资料等可选操作。

1.2. 前置条件

1. 用户已成功登录系统
2. 系统有足够的视频内容可供推荐

1.3. 后置条件

1. 用户的互动行为被正确记录
2. 推荐算法根据用户行为更新兴趣模型
3. 系统记录用户的浏览历史

1.4. 基本事件流

{启动应用}

1. 用况开始于用户打开应用主界面

{生成推荐列表}

2. 系统根据用户兴趣偏好生成个性化视频推荐列表

{显示视频}

3. 系统显示第一个推荐视频，开始自动播放

{切换视频}

4. 用户上下滑动屏幕切换视频

{更新用户画像}

5. 系统记录用户的浏览行为（如观看时长、滑动行为）并更新兴趣模型

{退出应用}

6. 用户退出应用时，用况结束

1.5. 备选流

A1 处理网络连接问题

在 {启动应用} 或 {生成推荐列表} 时，如果设备无网络连接：

1. 系统显示离线提示信息
2. 系统提供缓存的推荐内容供用户浏览
3. 用况继续执行基本流

A2 处理新用户或无历史数据

在 {生成推荐列表} 时，如果用户是新用户或缺乏历史行为数据：

1. 系统基于热门内容、通用偏好生成默认推荐
2. 用况继续执行基本流

A3 处理视频加载失败

在 {显示视频} 或 {切换视频} 时，如果视频内容加载失败：

1. 系统显示加载失败提示
2. 系统自动跳过该视频并显示下一个推荐视频
3. 用况继续执行基本流

A4 处理内容举报

在 {显示视频} 时，如果用户选择举报不当内容：

1. 系统记录举报信息
2. 系统将举报内容转交审核队列
3. 系统显示举报成功确认
4. 用况继续执行基本流

A5 用户选择创作音乐短视频

在{显示视频}时，如果用户选择创作视频：

1. 系统显示创作模式选择界面（拍摄或上传）
2. 用户选择创作模式并授予必要权限
3. 系统提供相应的创作工具
4. 用户录制或选择视频素材
5. 系统提供音乐选择界面
6. 用户浏览音乐库并选择背景音乐
7. 系统自动将视频与音乐节拍对齐
8. 用户使用编辑工具添加特效、滤镜、文字等
9. 用户设置视频标题、描述和可见性
10. 系统验证视频内容符合基本规范
11. 用户确认发布视频
12. 系统将视频提交到内容审核队列
13. 用况继续执行基本流

A6 用户选择参与社交互动

在{显示视频}时，如果用户选择进行社交互动：

1. 用户选择互动类型（点赞、评论、关注、分享等）和互动对象
2. 系统验证操作权限和内容状态
3. 系统记录互动行为
4. 系统向相关用户发送通知
5. 系统更新社交关系数据
6. 用况继续执行基本流

A7 用户选择发送私信

在{显示视频}时，如果用户选择发送私信：

1. 用户选择目标联系人
2. 用户编写消息内容

3. 用户发送消息

4. 系统验证消息内容和接收方状态

5. 系统存储消息并标记发送时间

6. 系统向接收方推送新消息通知

7. 系统更新对话列表

8. 用况继续执行基本流

A8 用户选择管理个人资料和内容

在{显示视频}时，如果用户选择管理个人资料：

1. 用户进入个人中心界面

2. 用户选择要管理的项目类型（个人信息、收藏内容、发布历史等）

3. 系统显示对应的管理界面和内容

4. 用户执行具体的管理操作

5. 系统验证操作权限和数据完整性

6. 系统更新相关数据和状态

7. 用况继续执行基本流

A9 用户搜索视频内容

在 {显示视频} 或 {切换视频} 时，如果用户选择搜索视频：

1. 用户进入搜索界面，输入搜索关键词

2. 系统根据关键词匹配视频标题、描述、标签和创作者信息

3. 系统展示搜索结果列表，按相关性排序

4. 用户选择并观看搜索结果中的视频

5. 系统记录搜索行为和结果点击

6. 用况继续执行基本流

3.4. 乐拍视界的内容审核员主要用况

1. 审核用户发布内容：内容审核员的核心工作，处理系统自动筛选出的疑似违规内容

2. 管理内容审核规则：维护和优化审核系统的工作，确保了审核工作的长期效力和准确性

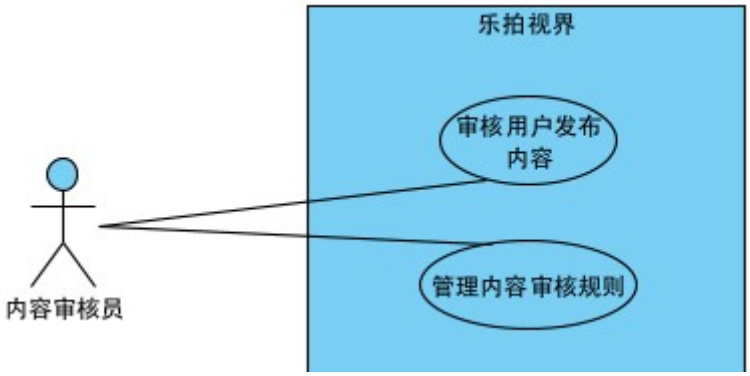

图 3-3“乐拍视界”中的内容审核员主要用况

1. 审核用户发布内容用况

1.1. 简要描述

内容审核员对系统自动筛查出的疑似违规内容（包括视频、评论、私信等）进行人工复核，并根据平台规范作出相应处理决定。

1.2. 前置条件

1. 内容审核员已成功登录系统

2. 系统内存在待处理（或重新分配）的疑似违规内容队列

1.3. 后置条件

1. 内容项被标记为最终状态（如“通过”、“驳回”）

2. 生成审核操作记录，包括审核员、时间、处理原因等

3. 若内容被驳回，系统会执行相应的处置动作（如删除内容、对发布者发出警告等）

1.4. 基本事件流

{进入审核界面}

1. 用况开始于内容审核员进入审核工作主界面

{分配审核内容}

2. 系统从"待审核队列"中分配（或审核员主动领取）一项待审核内容，并展示内容详情（如视频、文字、发布者信息等）

{审查内容详情}

3. 内容审核员根据平台审核规范，仔细审查内容的各个方面

{做出审核决定}

4. 审核员根据判断做出处理决定

{记录审核结果}

5. 系统记录审核结果，并更新该内容的状态

{结束审核任务}

6. 审核员继续处理下一项内容，或用况结束

1.5. 备选流

A1 处理轻微违规内容

在 {做出审核决定} 时，如果审核员认定内容属于轻微违规（如用词不雅）：

1. 系统提示审核员选择预设的违规类型
2. 审核员确认违规类型和处理方式
3. 系统除记录结果外，自动向内容发布者发送违规提醒
4. 用况继续执行基本流

A2 标记并移交复杂内容

在 {做出审核决定} 时，如果审核员认为内容复杂，无法立即做出判断（如可能涉及新型违规手法或法律边界模糊）：

1. 审核员将该内容标记为"待讨论"
2. 系统将内容转交至专家仲裁层或更资深的审核员进行处理
3. 系统记录移交状态和备注信息
4. 用况继续执行基本流

A3 批量处理同类内容

在 {分配审核内容} 时，如果待审核内容来自同一用户或为高度相似的内容：

1. 审核员选择启用"批量处理"模式
2. 系统展示内容列表和批量操作选项

3. 审核员进行一次判断后，将结果应用于同批所有内容

4. 系统批量更新所有相关内容状态

5. 用况继续执行基本流

A4 处理审核标准查询

在 {审查内容详情} 时，如果审核员对审核标准有疑问：

1. 审核员查询相关审核规则和案例

2. 系统显示详细的规则说明和类似案例

3. 审核员基于规则说明做出判断

4. 用况继续执行基本流

2. 管理内容审核规则用况

2.1. 简要描述

具有更高权限的内容审核员（或规则管理员）根据最新政策、业务变化或审核反馈，对系统自动审核所依赖的规则库（如敏感词库、语义模型）进行更新、测试和优化。

2.2. 前置条件

1. 内容审核员已成功登录系统

2. 该审核员拥有"规则管理"的特殊权限

2.3. 后置条件

1. 审核规则库被成功更新

2. 记录规则变更日志（版本、修改人、时间、原因）

2.4. 基本事件流

{进入规则管理}

1. 用况开始于规则管理员进入"审核规则管理"界面

{展示规则列表}

2. 系统展示当前的规则列表、分类及状态

{执行规则操作}

3. 规则管理员执行管理操作，如新增、修改、禁用或启用某条规则

{验证规则变更}

4. 系统验证规则修改的权限和有效性

{保存规则变更}

5. 系统保存更新后的规则，并记录版本日志

{结束规则管理}

6. 用况结束

2.5. 备选流

A1 测试与验证规则效果

在 {执行规则操作} 时，如果规则管理员需要测试新规则效果：

1. 规则管理员选择在测试环境中运行新规则
2. 系统在测试环境中执行规则测试
3. 系统展示新旧规则的对比效果（准确率、效率等）
4. 规则管理员基于测试结果决定是否正式发布规则
5. 用况继续执行基本流

A2 基于反馈优化规则

在 {执行规则操作} 时，如果规则管理员需要基于实际案例优化规则：

1. 规则管理员查看某条规则的"误判案例"或"漏判案例"统计
2. 系统根据案例分析推荐规则调整方案
3. 规则管理员采纳建议并对规则进行优化
4. 用况继续执行基本流

A3 导入外部规则库

在 {执行规则操作} 时，如果规则管理员需要批量导入规则：

1. 规则管理员选择导入外部规则文件
2. 系统验证文件格式和内容有效性
3. 系统解析并展示导入的规则内容
4. 规则管理员确认导入内容和冲突解决方案

5. 系统批量更新规则库

6. 用况继续执行基本流

3.5. 乐拍视界的客服主要用况

1. 处理用户咨询与投诉：这是客服的核心工作，直接面向用户解决问题

2. 管理客服知识库：这是支撑客服工作的重要后台管理功能

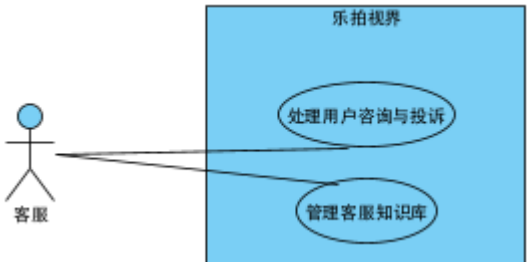

图 3-4“乐拍视界”中的客服主要用况

1. 处理用户咨询与投诉用况

1.1. 简要描述

客服通过系统接收用户提交的咨询、投诉、反馈等工单，进行调查分析并与用户沟通，最终解决问题。

1.2. 前置条件

1. 客服已成功登录客服工作台系统

2. 系统内存在待处理的用户工单队列

1.3. 后置条件

1. 用户工单被标记为最终处理状态

2. 生成完整的工单处理记录，包括沟通记录、解决方案等

3. 用户满意度得到记录

1.4. 基本事件流

{进入工作台}

1. 用况开始于客服登录系统并进入客服工作台主界面

{分配工单}

2. 系统从"待处理工单队列"中分配一个新的用户工单给客服

{了解问题}

3. 客服仔细阅读工单内容，了解用户的问题或投诉详情

{联系用户}

4. 客服通过系统提供的沟通渠道与用户进行联系和沟通

{分析解决方案}

5. 客服分析问题原因，调查相关背景信息，并形成解决方案

{执行处理}

6. 客服向用户解释问题原因并提供解决方案，执行必要的处理操作

{关闭工单}

7. 系统记录处理结果，客服关闭工单并记录处理摘要

{结束处理}

8. 用况结束

1.5. 备选流

A1 处理常见问题快速回复

在 {分析解决方案} 时，如果客服识别该问题属于常见问题：

1. 客服从知识库中调用预设的标准回复模板
2. 系统提供一键插入模板内容的功能
3. 客服根据具体情况微调模板内容后发送给用户
4. 用况继续执行基本流

A2 升级复杂问题至专家支持

在 {分析解决方案} 时，如果客服无法独立解决复杂技术问题：

1. 客服将工单标记为"需要专家支持"
2. 系统将工单转派至专业技术支持团队
3. 系统记录转派原因和当前处理进展
4. 用况继续执行基本流

A3 处理用户追加反馈

在 {关闭工单} 时，如果用户在工单关闭前追加新的反馈：

1. 系统提示客服有新的用户消息
2. 客服重新与用户沟通，了解追加的问题
3. 根据新情况补充处理方案
4. 用况继续执行基本流

A4 处理紧急投诉事件

在 {了解问题} 时，如果客服识别该工单属于紧急投诉：

1. 系统自动提升工单优先级
2. 客服按照紧急投诉处理流程优先处理
3. 系统记录紧急处理过程和结果
4. 用况继续执行基本流

2. 管理客服知识库用况

2.1. 简要描述

客服人员维护和更新客服知识库，包括添加新的常见问题解答、更新解决方案、优化回复模板等，以提升客服团队的工作效率和服务质量。

2.2. 前置条件

1. 客服已成功登录系统
2. 该客服具有知识库管理权限

2.3. 后置条件

1. 知识库内容得到更新和完善
2. 记录知识库变更历史
3. 相关客服人员接收到知识库更新通知

2.4. 基本事件流

{进入知识库管理}

1. 用况开始于客服进入知识库管理界面

{展示知识库}

2. 系统展示知识库分类结构和内容列表

{执行知识库操作}

3. 客服执行知识库管理操作，如新增、修改或删除知识条目

{验证操作内容}

4. 系统验证操作权限和内容规范性

{保存知识库变更}

5. 系统保存知识库变更，更新版本信息

{结束知识库管理}

6. 用况结束

2.5. 备选流

A1 基于工单统计优化知识库

在 {执行知识库操作} 时，如果客服需要基于实际工单数据优化知识库：

1. 系统提供工单统计和分析数据
2. 客服识别高频问题和解决方案缺口
3. 基于分析结果新增或修改知识库条目
4. 用况继续执行基本流

A2 处理知识库内容审核

在 {验证操作内容} 时，如果系统检测到知识库内容需要审核：

1. 系统将新增或修改的内容标记为"待审核"
2. 知识库管理员或资深客服进行内容审核
3. 审核通过后内容正式发布到知识库
4. 用况继续执行基本流

A3 批量导入知识库内容

在 {执行知识库操作} 时，如果客服需要批量更新知识库：

1. 客服选择批量导入功能
2. 系统提供导入模板和格式要求

3. 客服上传整理好的知识库文件

4. 系统解析并验证导入内容

5. 系统批量更新知识库内容

6. 用况继续执行基本流

A4 设置知识库权限和可见性

在 {执行知识库操作} 时，如果客服需要设置不同内容的访问权限：

1. 客服设置知识条目的可见性范围（如全员可见、仅客服可见等）

2. 系统根据设置更新权限控制

3. 不同角色的用户只能看到相应权限的内容

4. 用况继续执行基本流

3.6. 乐拍视界的运营人员主要用况

1. 策划并执行运营活动：通过策划挑战赛、话题活动等提升用户参与度

2. 管理内容推荐策略：优化内容分发和推荐机制

3. 监控平台运营数据：通过数据分析指导运营决策

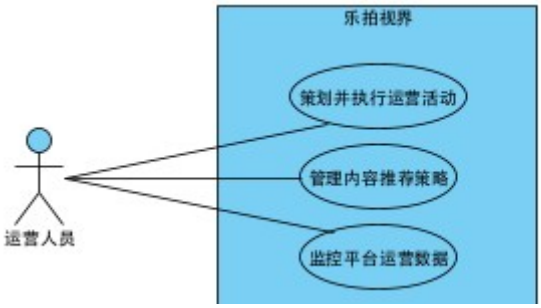

图 3-5“乐拍视界”中的运营人员主要用况

1. 策划并执行运营活动用况

1.1. 简要描述

运营人员策划、创建、发布和监控线上运营活动（如音乐挑战赛、话题活动等），旨在提升用户参与度和内容产出。

1.2. 前置条件

1. 运营人员已成功登录运营管理后台

2. 运营人员具有活动策划和发布权限

1.3. 后置条件

1. 运营活动成功创建并发布到平台
2. 活动数据监控机制已建立
3. 活动效果报告生成

1.4. 基本事件流

{进入活动管理}

1. 用况开始于运营人员进入活动管理界面

{创建活动}

2. 运营人员创建新的运营活动，填写活动基本信息（名称、主题、时间等）

{验证活动信息}

3. 系统验证活动信息的完整性和合理性

{配置活动规则}

4. 运营人员配置活动规则、奖励机制和展示位置

{设置监控指标}

5. 运营人员设置活动数据监控指标和报告频率

{发布活动}

6. 运营人员发布活动，系统将活动推送到指定用户群体

{监控活动效果}

7. 系统开始收集活动参与数据，生成实时监控报告

{结束活动管理}

8. 用况结束

1.5. 备选流

A1 调整活动策略

在 {监控活动效果} 时，如果活动参与度未达预期：

1. 运营人员分析活动数据，识别问题原因
2. 运营人员调整活动规则或奖励机制

3. 系统应用调整并重新推送活动信息

4. 用况继续执行基本流

A2 处理活动异常情况

在 {监控活动效果} 时，如果发现异常参与行为（如刷量、违规内容）：

1. 系统自动标记异常参与行为

2. 运营人员审查异常情况并采取处理措施

3. 系统清理异常数据并更新活动统计

4. 用况继续执行基本流

A3 延长或提前结束活动

在 {监控活动效果} 时，如果需要调整活动时间：

1. 运营人员根据活动效果决定延长或提前结束活动

2. 系统更新活动时间设置并通知参与用户

3. 用况继续执行基本流

2. 管理内容推荐策略用况

2.1. 简要描述

运营人员通过调整推荐算法参数、设置内容权重、管理热门话题等方式，优化平台内容分发效果和用户体验。

2.2. 前置条件

1. 运营人员已成功登录运营管理后台

2. 运营人员具有推荐策略管理权限

2.3. 后置条件

1. 推荐策略配置得到更新

2. 系统按照新的策略进行内容分发

3. 记录策略变更历史

2.4. 基本事件流

{进入策略管理}

1. 用况开始于运营人员进入推荐策略管理界面

{展示当前策略}

2. 系统展示当前推荐算法的配置参数和效果指标

{分析推荐效果}

3. 运营人员分析推荐效果数据，识别优化机会

{调整策略参数}

4. 运营人员调整推荐算法参数或内容权重设置

{验证参数变更}

5. 系统验证参数变更的合理性和安全性

{应用新策略}

6. 系统应用新的推荐策略，在部分用户群体中进行 A/B 测试

{监控策略效果}

7. 系统监控新策略的效果数据，与旧策略进行对比分析

{结束策略管理}

8. 用况结束

2.5. 备选流

A1 处理内容多样性问题

在 {分析推荐效果} 时，如果发现推荐内容同质化严重：

1. 运营人员调整多样性参数，增加内容探索权重
2. 系统重新计算推荐结果，增加新类型内容的曝光
3. 用况继续执行基本流

A2 紧急干预热点内容

在 {调整策略参数} 时，如果需要紧急推广或抑制特定内容：

1. 运营人员设置特定内容的权重系数
2. 系统立即调整该内容在推荐流中的位置
3. 用况继续执行基本流

A3 回滚策略变更

在 {监控策略效果} 时，如果新策略导致关键指标下降：

1. 运营人员决定回滚到之前的策略版本
2. 系统恢复之前的参数配置
3. 用况继续执行基本流

3. 监控平台运营数据用况

3.1. 简要描述

运营人员通过数据看板监控平台关键运营指标，分析用户行为趋势，发现运营问题并制定相应优化策略。

3.2. 前置条件

1. 运营人员已成功登录运营管理后台
2. 系统数据收集和处理功能正常运行

3.3. 后置条件

1. 运营人员获得平台运营状况的全面了解
2. 识别出需要优化的运营问题
3. 制定相应的优化行动计划

3.4. 基本事件流

{进入数据看板}

1. 用况开始于运营人员进入运营数据看板界面

{展示核心指标}

2. 系统展示关键运营指标（DAU、留存率、内容产出等）的实时数据

{分析数据趋势}

3. 运营人员分析指标趋势，识别异常波动或潜在问题

{深入分析问题}

4. 运营人员钻取详细数据，深入分析问题原因

{制定优化方案}

5. 运营人员基于分析结果制定优化建议或行动计划

{记录分析结果}

6. 系统记录分析结果和优化方案

{结束数据分析}

7. 用况结束

3.5. 备选流

A1 处理数据异常告警

在 {展示核心指标} 时，如果系统检测到关键指标异常：

1. 系统触发异常告警，突出显示问题指标
2. 运营人员立即查看告警详情，分析异常原因
3. 运营人员制定紧急应对措施
4. 用况继续执行基本流

A2 生成周期性运营报告

在 {记录分析结果} 时，如果需要生成正式运营报告：

1. 运营人员选择报告模板和时间范围
2. 系统自动生成包含图表和分析的运营报告
3. 运营人员审核并完善报告内容
4. 系统分发报告给相关利益相关者
5. 用况继续执行基本流

A3 设置数据监控告警

在 {分析数据趋势} 时，如果发现需要持续监控的关键模式：

1. 运营人员设置自定义数据监控规则和告警阈值
2. 系统保存监控规则，开始持续监控
3. 当触发告警条件时系统自动通知运营人员
4. 用况继续执行基本流

第4章需求分析

4.1. 分析模型概述

旨在生成“乐拍视界”短视频社交平台的分析模型。该模型核心目的是在忽略物理实现细节（如具体编程语言、数据库类型或特定 UI 控件）的前提下，对系统的逻辑结构和业务行为进行详细说明。

1. 分析模型的目标与关注点

分析模型关注的是“乐拍视界”系统的逻辑行为。它通过分析问题情景的逻辑结构以及各个逻辑元素相互交互的方式，详细说明各项需求。

1. 逻辑视角：本模型将从整体上建模“乐拍视界”所期望的应用系统业务行为，建立独立于任何特定实现方法的逻辑结构。

2. 完善需求：通过分析，我们将检查不同需求之间可能存在的冲突，确保对“乐拍视界”涉及的人（如普通用户、审核员）、事（如视频发布、审核流程）和概念（如推荐算法策略）有清晰、无歧义的理解。

3. 设计基础：本模型将旨在减少软件设计和构建中的错误与不一致性。

2. 分析模型的组成

本分析模型主要由一组分析类组成，这些类通过协作来实现定义的用况。我们将采用健壮性分析（Robustness Analysis）技术，将类划分为以下三种标准衍型（Stereotypes）：

1. 边界类（Boundary Classes）：用于建模“乐拍视界”系统与其参与者（如观看者、创作者、运营人员）之间的交互。它们代表了系统与外部世界的逻辑接口（例如：视频浏览界面、审核后台界面），主要负责管理跨越系统边界的信息传送。

2. 实体类（Entity Classes）：用于建模“乐拍视界”应用领域中的核心现象或概念的信息及相关行为。这些类通常源自领域模型，代表了需要系统永久存储的信息（例如：短视频、音乐信息、用户信息、审核记录）。

3. 控制类（Control Classes）：用于表示用况逻辑中的协调、排序、事务及控制。它们充当边界类与实体类之间的“胶水”，负责处理特定于用况的计算和调度方面（例如：处理推荐流的分发逻辑、协调视频上传与转码流程）。

3. 用况的实现（Realization）

通过用况实现来构建模型。对于“乐拍视界”的关键用况（如“浏览个性化视频流”、“审核用户发布内容”等），我们将：

1. 识别参与该用况的一组分析类（协作）；

2. 使用通信图（Communication Diagram）来展示这些对象如何在特定情景下通过消息传递进行动态交互；

3. 最终整合形成完整的分析类图，展示类之间的静态结构、关联及多重性。

4.2. 健壮性分析

根据 Jacobson 的分类方法，我们将分析模型中的类分为三种特定的衍型：边界类（Boundary）、实体类（Entity）和控制类（Control）。

1. 边界类

边界类用于对系统与外部世界（参与者）之间的交互进行建模。在“乐拍视界”中，边界类并不直接表示具体的 UI 控件（如按钮或窗口），而是代表系统主要的逻辑接口，负责处理输入和输出。

根据参与者和用况，识别出以下关键边界类：

1. MainFeedUI（主页视频流界面）

关联用况：浏览个性化视频流

描述：负责向普通用户展示全屏沉浸式视频流，捕获用户的滑动（切换视频）、点赞、评论等交互操作，并将请求传递给系统。

2. VideoCreationUI（创作工具界面）

关联用况：浏览个性化视频流（备选流：用户选择创作音乐短视频）

描述：提供拍摄、音乐选择、特效编辑的交互入口，负责接收用户录制的视频数据和编辑指令。

3. AuditWorkbenchUI（审核工作台界面）

关联用况：审核用户发布内容

描述：面向内容审核员的界面，负责展示待审核的视频内容及违规详情，并接收审核员提交的“通过”或“驳回”决定。

4. OperationsDashboardUI（运营管理后台界面）

关联用况：策划并执行运营活动、监控平台运营数据

描述：面向运营人员的界面，用于输入活动规则、发布挑战赛以及展示关键运营指标的

数据看板。

2. 实体类

实体类用于对应用领域中的现象、概念、人或事物的相关信息及行为进行建模。这些类表示系统必须永久存储的信息。在“乐拍视界”中，这些类源自对业务领域的理解。

根据项目文档中的业务描述，识别出以下关键实体类：

1. User（用户）

描述：表示平台的注册用户（包括内容消费者和创作者）。系统需存储其基本信息、兴趣画像及社交关系（关注/粉丝）。

2. Video（短视频）

描述：核心业务对象。存储视频文件的元数据（如 URL、时长）、统计数据（点赞数、播放量）以及状态（审核中、已发布、已下架）。

3. Music（音乐）

描述：表示曲库中的音乐条目。存储版权信息、音频文件链接及对应的节拍点数据，是“同款音乐”功能的基础。

4. AuditTask（审核任务）

描述：表示一条待处理的审核工作项。记录送审内容、审核状态、处理人及处理时间，用于追踪审核流程。

5. OperationActivity（运营活动）

描述：表示平台发起的挑战赛或话题活动。存储活动规则、时间范围及关联的视频集合。

3. 控制类

控制类用于表示协调、排序、事务处理以及对其他对象的控制。它们充当边界对象和实体对象之间的“胶水”，负责将用况中的逻辑行为（特定于用况的计算和调度）与具体的实体数据分离开来。

按照“每个用况通常对应一个控制类”的原则，识别出以下关键控制类：

1. FeedController（推荐流控制器）

关联用况：浏览个性化视频流

职责：协调视频流的加载逻辑。它接收来自 MainFeedUI 的请求，调用推荐算法（逻辑），查询 Video 和 User 实体，并返回排序后的视频列表。

2. CreationController（创作发布控制器）

关联用况：浏览个性化视频流（备选流：创作）

职责：管理视频制作流程。协调 Music 实体的加载，处理视频合成逻辑，并创建新的 Video 实体，将其状态设置为“待审核”。

3. AuditController（审核控制器）

关联用况：审核用户发布内容

职责：管理审核工作流。负责从队列中提取 AuditTask，将数据分发给 AuditWorkbenchUI，并根据审核员的决策更新 Video 实体的状态（如设为“公开”或“违规删除”）。

4. ActivityController（活动运营控制器）

关联用况：策划并执行运营活动

职责：处理运营活动的生命周期。负责验证活动规则，创建 OperationActivity 实体，并协调活动内容的推送。

4.3. 交互建模

通过建立协作（Collaboration），并使用通信图来展示对象之间的动态交互和消息传递方案。交互设计遵循健壮性分析规则：参与者与边界对象交互，边界对象与控制对象交互，控制对象协调实体对象。

1. 浏览个性化视频

1.1. 参与对象（协作角色）

1. 普通用户 (Actor)：发起浏览请求。

2. :MainFeedUI (Boundary)：负责展示视频流界面，接收用户操作。

3. :FeedController (Control)：负责协调推荐逻辑，获取数据。

4. :RecommendationModel (Entity)：封装推荐算法逻辑，提供视频 ID 列表。

5. :Video (Entity)：存储视频的具体内容（URL、标题）。

6. :User (Entity)：存储视频创作者的信息（头像、昵称）。

1.2. 通讯图

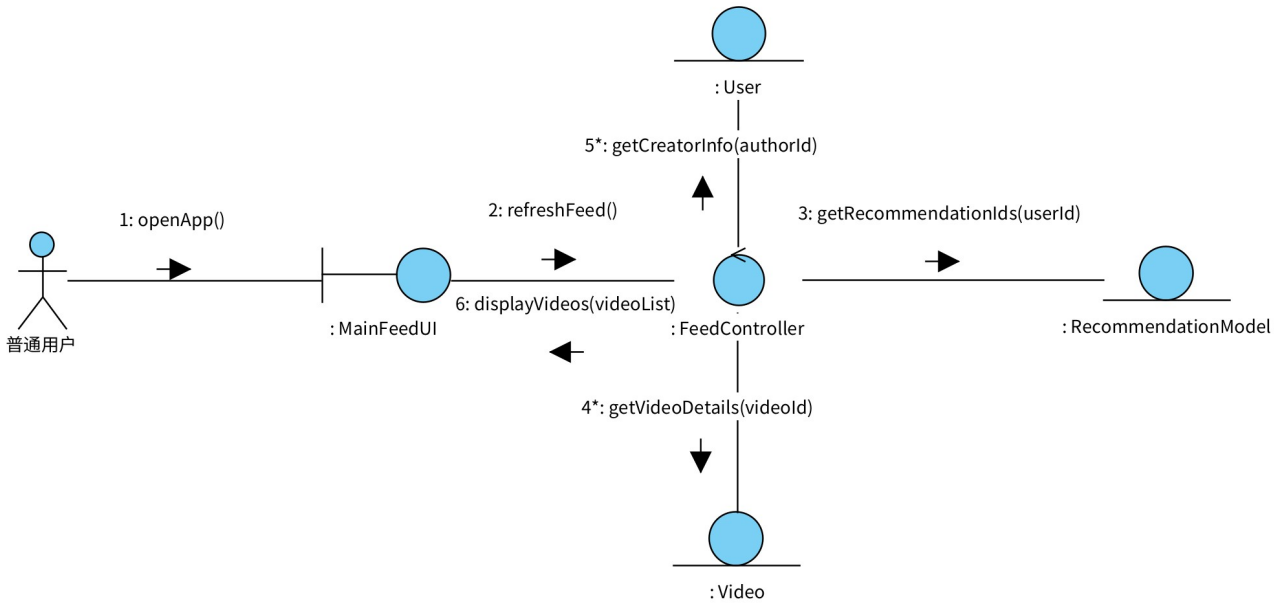

图 4-1“乐拍视界”中的浏览个性化视频通讯图

1. 启动与请求：

1: openApp()：普通用户启动应用，:MainFeedUI 初始化。

2: refreshFeed()：:MainFeedUI 向 :FeedController 发送刷新请求。

2. 获取推荐列表：

3: getRecommendationIds(userId)：:FeedController 调用 :RecommendationModel，获取一组推荐的视频 ID 列表。

3. 构建视频数据（迭代）：

4*: getVideoDetails(videoId)：:FeedController 根据 ID 列表，向 :Video 对象发送消息，获取视频元数据（如播放地址）。

5*: getCreatorInfo(authorId)：:FeedController 根据视频关联的作者 ID，向 :User (创作者) 对象发送消息，获取作者昵称和头像。

4. 展示结果：

6: displayVideos(videoList)：:FeedController 将组装好的数据返回给 :Main-

FeedUI，界面渲染视频流供用户观看。

2. 审核视频

2.1. 参与对象（协作角色）

1. 内容审核员 (Actor)：执行审核操作。

2. :AuditWorkbenchUI (Boundary)：审核工作台界面，展示视频和操作按钮。

3. :AuditController (Control)：管理审核流程状态流转。

4. :AuditTask (Entity)：记录审核任务的状态和分配情况。

5. :Video (Entity)：被审核的视频对象，包含状态属性。

2.2. 通讯图

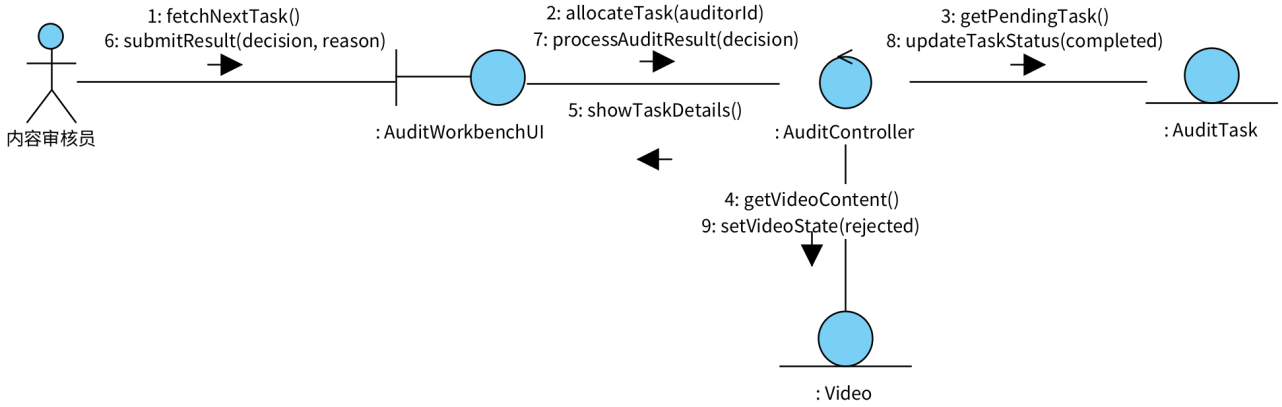

图 4-2“乐拍视界”中的审核视频通讯图

1. 获取任务：

1: fetchNextTask()：内容审核员在 :AuditWorkbenchUI 点击“获取任务”。

2: allocateTask(auditorId)：界面向 :AuditController 请求分配任务。

3: getPendingTask()：:AuditController 从 :AuditTask 队列中获取一个待处理的任务。

4: getVideoContent()：:AuditController 根据任务关联，从 :Video 获取视频播放链接和描述文本。

5: showTaskDetails()：:AuditController 将内容返回给界面进行展示。

2. 提交审核结果：

6: submitResult(decision, reason)：审核员观看后，在 :AuditWorkbenchUI 提

交决策（如“驳回”及原因）。

7: processAuditResult(decision)：界面将决策数据发送给 :AuditController。

3. 更新状态：

8: updateTaskStatus(completed)：:AuditController 更新 :AuditTask 的状态为“已结束”。

9: setVideoState(rejected)：:AuditController 向 :Video 发送消息，将视频状态更新为“公开”或“违规下架”等。

3. 创建发布活动

3.1. 参与对象（协作角色）

1. 运营人员 (Actor)：活动发起者。
2. :OperationsDashboardUI (Boundary)：运营管理后台界面。
3. :ActivityController (Control)：负责活动的创建逻辑和验证。
4. :OperationActivity (Entity)：存储活动规则、时间等信息。

3.2. 通讯图

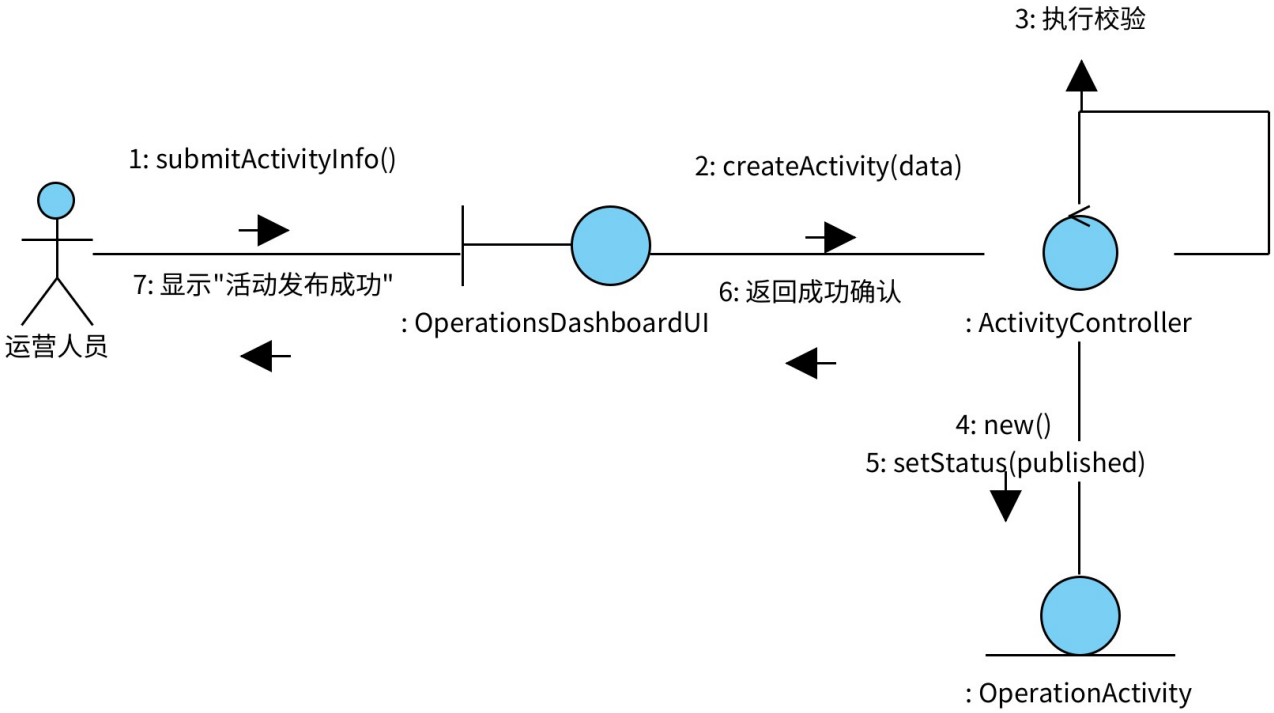

图 4-3“乐拍视界”中的创建发布活动通讯图

1. 提交活动信息：

运营人员在 :OperationsDashboardUI 输入活动详情（名称、规则、时间），发送 submitActivityInfo() 消息。

2. 验证与创建：

:OperationsDashboardUI 将数据传递给 :ActivityController，发送 createActivity(data) 消息。

:ActivityController 执行内部校验逻辑（如检查时间冲突）。

校验通过后，:ActivityController 向 :OperationActivity 发送 new() 创建消息，生成一个新的活动实例。

3. 发布与反馈：

:ActivityController 向 :OperationActivity 发送 setStatus(published) 消息，将活动设为生效。

:ActivityController 向 :OperationsDashboardUI 返回成功确认。

界面向运营人员显示“活动发布成功”。

4.4. 职责分配与 CRC 卡

为了验证分析类的完整性并合理分配系统职责，我们为核心用况中的关键类创建了 CRC 卡。以下卡片描述了类的高层职责以及为了完成这些职责所需的协作者。

1. 浏览个性化视频相关类

1.1. Class Name: FeedController (控制类)

Responsibilities (职责)	**Collaborations (协作)**
协调个性化推荐列表的生成与刷新	RecommendationModel
获取视频的详细元数据与播放地址	Video
获取视频创作者的个人信息（头像、昵称）	User
将组装好的视频流数据传递给界面展示	MainFeedUI

1.2. Class Name: RecommendationModel (实体类)

Responsibilities (职责)	**Collaborations (协作)**
维护推荐算法策略与规则	
根据用户 ID 计算并返回推荐视频 ID 列表	User

Responsibilities (职责)	Collaborations (协作)
接收并处理用户行为反馈以更新兴趣模型	

1.3. Class Name: Video (实体类)

Responsibilities (职责)	Collaborations (协作)
维护视频核心元数据（URL、标题、描述、标签）	
维护视频的发布状态（审核中、已发布、已下架）	
提供关联的背景音乐信息	Music
记录并提供统计数据（点赞数、评论数、转发数）	

1.4. Class Name: User (实体类)

Responsibilities (职责)	Collaborations (协作)
维护用户基本信息（ID、昵称、头像、认证状态）	
维护用户的社交关系（关注列表、粉丝列表）	
维护用户的兴趣画像数据	

2. 审核用户发布内容相关类

2.1. Class Name: AuditController (控制类)

Responsibilities (职责)	Collaborations (协作)
获取并分配待处理的审核任务	AuditTask
获取待审核视频的完整内容供预览	Video
处理审核员提交的决策结果（通过/驳回）	AuditWorkbenchUI
根据审核结果更新视频的可见性状态	Video
记录审核操作日志	AuditTask

2.2. Class Name: AuditTask (实体类)

Responsibilities (职责)	Collaborations (协作)
维护审核任务队列与状态（待审核、处理中、已结束）	
关联具体的待审核视频对象	Video
记录负责该任务的审核员信息	User (Auditor)
记录审核结束时间与处理意见	

3. 策划并执行运营活动相关类

3.1. Class Name: ActivityController (控制类)

Responsibilities (职责)	Collaborations (协作)
验证活动规则与时间的冲突性	
创建并初始化新的运营活动实例	OperationActivity
发布活动并将其推送到前台	OperationActivity, MainFeedUI
收集活动参与数据并生成报告	

3.2. Class Name: OperationActivity (实体类)

Responsibilities (职责)	Collaborations (协作)
维护活动详情（名称、规则、起止时间）	
维护活动状态（草稿、进行中、已结束）	
聚合参与该活动的视频列表	Video

4.5. 整合分析类图

通过对“浏览个性化视频流”、“审核用户发布内容”及“策划并执行运营活动”等核心用况的分析，我们将局部产生的类图进行整合，形成了“乐拍视界”的整体分析类图。该模型展示了系统逻辑架构中的静态结构。

1. 实体类 (Entity Classes)

1.1. Video (短视频)

属性: videoID, title, url, createTime, status (审核中/已发布/已下架), likeCount, commentCount

操作: getDetails(), updateStatus(newStatus), incrementStats(type)

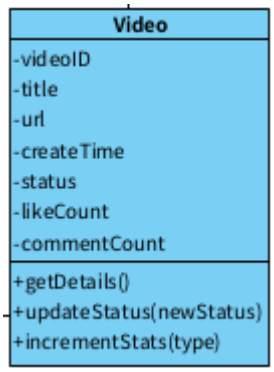

图 4-4“乐拍视界”中的 Video 类图

1.2. User (用户)

属性: userID, nickname, avatarUrl, role (普通用户/审核员/运营), interestTags

操作: getInfo(), updateProfile(data)

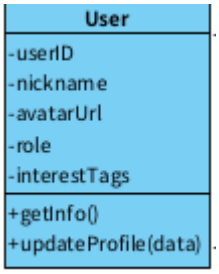

图 4-5“乐拍视界”中的 User 类图

1.3. Music (音乐)

属性: musicID, title, artist, audioUrl, copyrightStatus

操作: getMusicInfo()

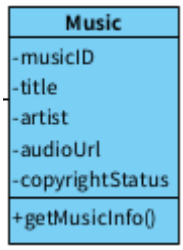

图 4-6“乐拍视界”中的 Music 类图

1.4. AuditTask (审核任务)

属性: taskID, createTime, auditStatus (待处理/通过/驳回), rejectReason

操作: assignTo(auditorID), updateResult(decision, reason)

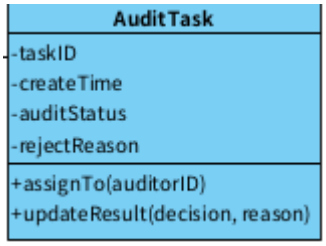

图 4-7“乐拍视界”中的 AuditTask 类图

1.5. OperationActivity (运营活动)

属性: activityID, name, rules, startTime, endTime, state

操作: publish(), addVideo(videoID)

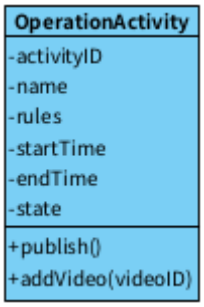

图 4-8“乐拍视界”中的 OperationActivity 类图

2. 控制类 (Control Classes)

1. FeedController: refreshFeed(userId), loadMore()

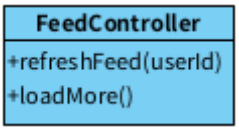

图 4-9“乐拍视界”中的 FeedController 类图

2. AuditController: getPendingTask(), submitAuditResult(taskID, decision)

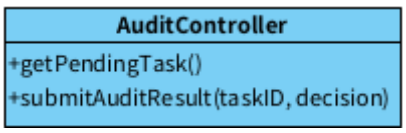

图 4-10“乐拍视界”中的 AuditController 类图

3. ActivityController: createActivity(info), publishActivity(activityID)

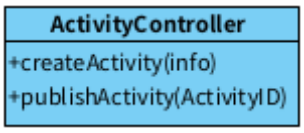

图 4-11“乐拍视界”中的 ActivityController 类图

3. 边界类 (Boundary Classes)

MainFeedUI: displayVideos(list), showError(msg)

图 4-12“乐拍视界”中的 MainFeedUI 类图

AuditWorkbenchUI: showTask(videoData), showSuccess(msg)

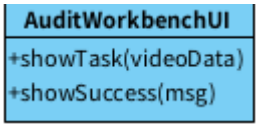

图 4-13“乐拍视界”中的 AuditWorkbenchUI 类图

4. 类关联关系

4.1. 用户与视频 (发布关系)

关联: User —— Video

多重性: 1 (User) <---> 0..* (Video)

说明: 一个用户可以发布零个或多个视频；一个视频必须属于且仅属于一个创作者。

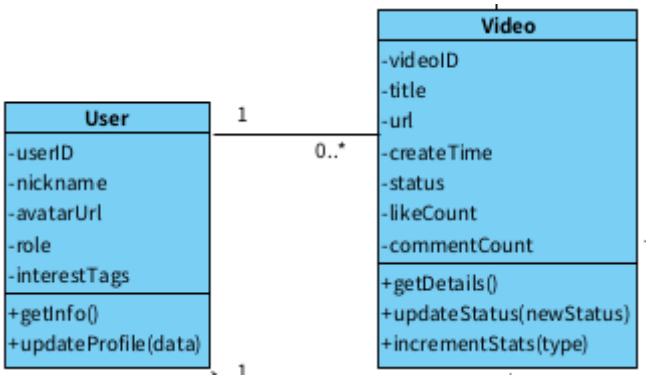

图 4-14“乐拍视界”中的用户与视频 (发布关系)图

4.2. 视频与音乐 (使用关系)

关联: Video —— Music

多重性: 0..* (Video) <---> 0..1 (Music)

说明: 一个视频可以使用零首（原声）或一首背景音乐；一首音乐可以被多个视频使用。

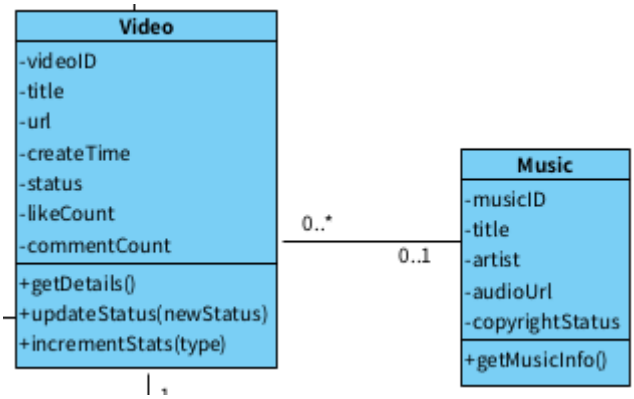

图 4-15“乐拍视界”中的视频与音乐 (使用关系)图

4.3. 审核任务与视频 (审核对象关系)

关联: AuditTask —— Video

多重性: 0..1 (AuditTask) <---> 1 (Video)

说明: 一个审核任务针对且仅针对一个视频；一个视频在特定时间点关联零个或一个活跃的审核任务。

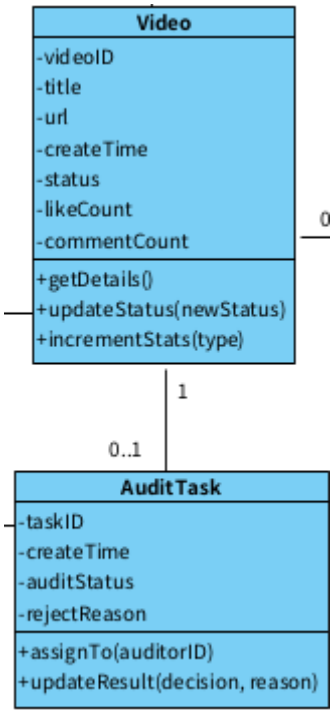

图 4-16“乐拍视界”中的审核任务与视频 (审核对象关系)图

4.4. 审核任务与用户 (执行关系)

关联: User (Auditor) —— AuditTask

多重性: 1 (User) <---> 0..* (AuditTask)

说明: 一个审核员可以处理多个审核任务；一个具体的审核任务由一名审核员负责。

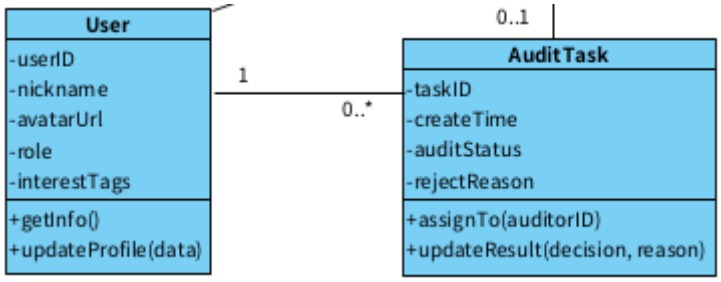

图 4-17“乐拍视界”中的审核任务与用户 (执行关系)图

4.5. 运营活动与视频 (聚合关系)

关联: OperationActivity —— Video

多重性: 0..1 (OperationActivity) <---> 0..* (Video)

说明: 一个活动包含多个参与视频；一个视频可以不参与活动或参与一个活动。

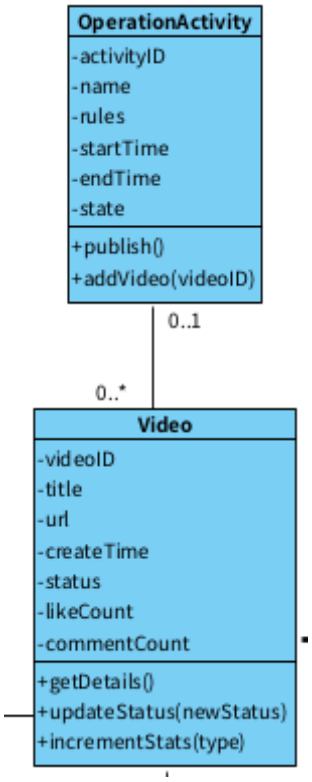

图 4-18“乐拍视界”中的运营活动与视频 (聚合关系)图

第5章架构设计

5.1. 设计目标与策略

1. 设计目标与优先级

本系统的架构设计旨在构建一个高性能、高可用且易于扩展的短视频社交平台。依据项目愿景与需求规格说明书，我们将设计目标按优先级划分为关键目标、重要目标和一般目标，并为每个目标设定了可度量的验收标准。

1.1. 高性能与响应速度

优先级： 关键

描述： 作为一款以“沉浸式体验”和“即时创作”为核心的短视频应用，系统的响应速度直接决定用户的留存率。架构必须支持海量多媒体数据的极速传输与实时渲染。

可度量指标：

首屏加载： 95% 的视频请求必须在 1 秒 内开始播放。

交互流畅度： App 主界面滑动及视频播放帧率需稳定在 60fps，无明显卡顿。

上传速度： 在 4G 网络环境下，视频上传速度需达到 1MB/s 以上。

并发处理： 系统需支持百万级日活用户的并发访问，峰值时段无服务降级。

1.2. 可用性与易用性

优先级： 关键

描述： 针对年轻用户群体，系统需提供极致简化的操作体验，确保“零门槛”创作与浏览。架构设计需支持灵活的 UI 变更和个性化推荐。

可度量指标：

学习成本： 新用户无需查看帮助文档即可完成首次视频拍摄与发布。

容错性： 在弱网环境下，应用仍能保持基本浏览功能（如加载缓存内容），不发生崩溃。

个性化： 推荐算法需在用户冷启动后 10 次滑动操作内，显著提升内容匹配度。

1.3. 可扩展性与伸缩性

优先级： 重要

描述： 鉴于项目预期的用户指数级增长（从校园冷启动到千万级用户），以及未来商业化功能（直播、电商）的拓展，架构必须具备水平扩展能力，避免重构。

可度量指标：

水平扩展： 服务端需支持通过增加服务器节点线性提升处理能力，无需修改代码。

功能扩展： 新增功能模块时，不影响现有核心模块的稳定性，且集成周期不超过 2 周。

1.4. 可靠性与稳定性

优先级： 重要

描述： 平台需保证服务的持续在线和数据的完整性，特别是在用户创作内容（UGC）的存储上，绝不允许数据丢失。

可度量指标：

系统可用性： 全年系统可用性需达到 99.9%（即全年停机时间不超过 8.76 小时），故障恢复时间（MTTR）需小于 1 小时。

崩溃率： 客户端单用户日均崩溃次数需控制在 0.01 次 以下。

数据持久性： 用户创作内容与个人数据的持久性保障需达到 99.99%。

1.5. 可维护性

优先级： 一般

描述： 系统应采用迭代开发模式，架构必须具备清晰的模块边界，降低代码耦合度，以便于其他成员快速上手和故障排查。

可度量指标：

模块耦合： 子系统间的依赖关系必须单向或通过接口隔离，修改一个子系统的内部实现不应波及其他子系统。

故障定位： 提供完整的日志追踪链路，开发人员应能在 10 分钟内定位生产环境报错的根本原因。

1.6. 安全性与合规性

优先级： 强制约束

描述： 必须严格遵守《网络安全法》及版权法规，保护用户隐私，防止未授权访问和恶

意内容传播。

可度量指标：

数据加密： 用户密码及敏感信息在传输和存储过程中必须 100% 加密。

内容审核： 100% 的上传视频需经过“AI+人工”双重审核机制，确保违规内容不被公开分发。

权限控制： 严格的 ACL（访问控制列表）机制，确保用户只能操作属于自己的资源。

2. 设计权衡

在构建“乐拍视界”系统的过程中，我们识别出以下几对核心的互斥目标。基于项目的商业愿景和技术约束，我们做出了明确的取舍决策。

2.1. 性能 vs. 成本

冲突点：

为了实现“秒开”（1 秒内起播）和“零卡顿”的极致体验，需要大量的边缘节点服务器（CDN）、昂贵的高带宽资源以及高性能的实时转码集群。

这将显著增加项目的初期基础设施投入和运营成本（OPEX），与创业初期控制成本的经济性目标相冲突。

设计决策： 优先保障性能。

理由：

对于短视频社交应用，用户体验是生存的基石。视频加载的延迟直接导致用户流失，其损失远高于基础设施成本。

策略： 采用“分级存储与分发”策略。对热点视频不计成本地进行全网 CDN 加速；对长尾冷门视频采用廉价的对象存储，接受稍高的加载延迟，以此在总体上平衡成本。

2.2. 可用性 vs. 一致性

冲突点：

在分布式系统环境下，为了保证百万级用户同时在线互动（点赞、评论、转发），系统必须具备极高的可用性。

然而，要保证所有用户在同一时刻看到完全一致的数据（如视频点赞数），需要复杂的同步锁定机制，这会降低系统的吞吐量和可用性。

设计决策： 在社交互动数据上，优先保证可用性（最终一致性）；在账户资金数据上，

优先保证强一致性。

理由：

用户对“点赞数”的实时精确性不敏感，但对点击后的“无响应”或“报错”非常敏感。因此，社交数据采用最终一致性模型（Base 理论），允许短时间的数据延迟同步。

涉及虚拟币购买、创作者收益提现等资金操作，必须采用强一致性事务（ACID），哪怕牺牲一部分性能，也要确保数据绝对准确。

2.3. 结构化分层 vs. 运行效率

冲突点：

分层架构（表现层、应用逻辑层、领域层、数据层）能极大地提升系统的可维护性和可复用性。

但是，分层架构引入了额外的中间层调用开销，数据在层级间传递和转换（如 DTO 到 Domain Object 的转换）会消耗 CPU 资源并增加延迟，这与高性能目标存在轻微冲突。

设计决策： 坚持严格的分层架构，通过缓存机制弥补效率损耗。

理由：

在项目团队中，代码的可维护性和模块解耦是团队协作的基础。若为了微小的性能提升而打破分层（如在表现层直接访问数据库），将导致系统架构腐化，难以维护。

策略： 对于关键路径（如视频流获取），通过引入 Redis 缓存层来减少穿透到数据库的层级调用，从而在保持架构清晰的同时保障性能。

2.4. 客户端富交互 vs. 瘦客户端

冲突点：

瘦客户端（主要逻辑在服务端）易于维护和更新，对设备要求低，但难以实现复杂的实时视频特效。

富客户端（主要逻辑在本地）能充分利用手机 GPU 进行渲染，提供流畅的拍摄和编辑体验，但增加了客户端的体积和适配难度（Android 机型碎片化）。

设计决策： 采用富客户端架构，应用 MVC 模式。

理由：

“乐拍视界”的核心竞争力在于“创作工具”，即拍摄时的美颜、滤镜和特效。这些操作涉及大量的实时图像处理，必须在本地完成，无法容忍网络传输带来的延迟。

服务端仅负责逻辑编排、数据存储和推荐计算，客户端承担主要的交互和渲染职责。

2.5. 安全性 vs. 用户体验

冲突点：

严格的内容审核和安全校验（如视频上传前的全量扫描）可能导致发布流程变慢，增加用户等待时间，破坏“即拍即享”的体验。

设计决策： 采用异步安全机制，前台体验优先。

理由：

用户发布视频后，系统应立即反馈“发布成功”（乐观 UI 更新），让用户可以继续浏览。

实际的安全审核（AI 识别 + 人工复核）在后台异步队列中进行。若审核不通过，再通过消息通知用户并下架视频。这种权衡既满足了合规性要求，又保护了用户的流畅体验。

3. 技术平台选择

本节将明确系统在物理设计阶段所采用的具体硬件、软件及网络技术栈。这些选择旨在满足前述的高性能、高可用性目标，并严格遵循项目文档中既定的技术约束。

3.1. 客户端平台

操作系统目标： Android 7.0 及以上版本

决策依据： 依据项目约束，需覆盖市场主流机型。

开发技术栈： Qt / QML (Quick)

决策依据：

性能与体验： QML 基于 OpenGL/Vulkan 渲染，能够提供流畅的 60fps 动画和高性能的视频滤镜渲染，满足“沉浸式体验”的 UI 需求。

底层能力： Qt 框架底层为 C++，能高效调用本地摄像头、音频处理库（如 FFmpeg）及 GPU 资源，确保视频拍摄和编辑的实时性。

跨平台潜力： 虽然目前仅针对 Android，但 Qt 的跨平台特性为未来拓展至 iOS 或桌面端预留了低成本迁移路径。

3.2. 服务端平台

操作系统： Linux

决策依据： 提供稳定、高效的运行环境，拥有最完善的服务器端工具链支持。

开发语言与框架： C++ (配合高性能网络库，Drogon 和 POCO)。

决策依据：

极致性能： 依据项目文档约束，后端采用 C++ 开发。在处理高并发视频流分发、实时转码及复杂的推荐算法计算时，C++ 能提供优于 Java/Python 的运行时效率和内存控制能力。

资源利用率： 在初期资金有限（服务器资源有限）的情况下，C++ 服务端能以更低的 CPU 和内存占用支撑更高的并发量。

3.3. 数据管理平台

采用混合存储架构，以适应不同类型数据的访问特征。

关系型数据库： PostgreSQL

决策依据：

对象-关系特性 (ORDBMS)： PostgreSQL 对复杂查询和 JSONB 数据类型的原生支持，非常适合存储用户画像、社交关系图谱等半结构化数据，比传统 MySQL 提供了更强的灵活性。

可靠性与并发： 作为一个成熟的开源数据库，PG 在高并发写入下的 MVCC（多版本并发控制）机制能有效支撑社交互动数据（评论、私信）的爆发式增长。

扩展性： 支持丰富的插件生态（如 PostGIS 用于未来可能的地理位置功能），符合系统的可扩展性目标。

缓存系统： Redis

用途： 缓存热点视频列表、用户 Session、高频计数（点赞数），减轻数据库压力，降低响应延迟。

对象存储 (Object Storage)： 云服务商 OSS

用途： 存储视频实体文件、音频文件和图片资源。利用云厂商的分布式存储能力保障海量非结构化数据的持久性（99.99%）。

3.4. 通信与中间件

通信协议： HTTP/2 (RESTful API)

数据序列化： JSON

决策依据： JSON 具有良好的可读性和通用性，便于前后端调试；HTTP/2 的多路复用特性可显著降低移动网络下的连接延迟。

消息队列： RabbitMQ

决策依据： 用于系统解耦和异步处理（如视频上传后的转码、审核）。RabbitMQ 在任务分发的可靠性上强。

反向代理与负载均衡： Nginx

决策依据： 作为统一的流量入口，处理 SSL 卸载、静态资源服务及后端 C++ 应用服务器的负载均衡。

5.2. 系统的架构风格

1. 总体架构模式

为了有效管理“乐拍视界”系统的复杂性，确保各组件职责单一且解耦，我们将系统在垂直方向上划分为四个不同的抽象层次。每一层都封装了特定的功能，并通过定义良好的接口向其直接上层提供服务。

1.1. 表现层

定位：系统的最顶层，直接面向最终用户。

职责：

用户交互：负责所有图形用户界面（GUI）的渲染，包括全屏沉浸式视频流展示、个人主页、消息列表等。

输入捕获：捕获用户的触摸手势（滑动、双击点赞）、摄像头输入及音频输入。

本地渲染：利用设备的 GPU 资源，实时处理视频滤镜、特效预览及美颜效果。

与下层交互：该层不包含核心业务逻辑，它通过网络协议调用应用逻辑层的接口来获取数据或提交操作。

1.2. 应用逻辑层

定位：系统的服务编排层，位于服务端，是业务功能的入口。

职责：

流程编排：协调和控制用况的执行流程。例如，在“发布视频”用况中，它负责协调权限检查、文件上传、元数据创建等步骤。

安全与控制：执行用户认证、授权以及请求频率限制。

DTO 转换：将下层返回的领域对象转换为适合表现层展示的数据传输对象。

与下层交互：该层不直接处理数据的持久化细节，而是调用领域层的核心实体逻辑。

1.3. 领域层

定位：系统的核心层，包含所有业务实体与规则，独立于具体的应用流程和外部接口。

职责：

业务实体：封装核心概念及其状态，如 Video（视频）、User（用户）、Music（音乐）、AuditTask（审核任务）。

业务规则：实现与状态相关的核心逻辑。例如，计算视频的热度分值、判断用户等级权益、校验评论内容的合法性规则。

状态维护：维护实体的生命周期状态（如视频从“审核中”到“已发布”的状态流转）。

与下层交互：该层定义了数据访问的接口，但具体的数据库操作委托给数据源层实现。

1.4. 数据源层

定位：系统的最底层，负责基础设施与数据的物理存储。

职责：

数据持久化：通过 PostgreSQL 数据库驱动，实现领域对象的 CRUD 操作及复杂查询。

缓存管理：操作 Redis 集群，处理热点数据的缓存读写，保障高并发下的响应速度。

文件存储接口：封装对云端对象存储（OSS）的访问细节，负责视频、音频等二进制大文件（BLOB）的物理存取。

特性：该层对上层隐藏了具体的数据库方言和存储细节，确保即使更换存储介质（如从 OSS 切换到自建 Ceph），上层业务逻辑也无需修改。

2. 分层策略

为了最大化系统的可维护性并降低层级间的耦合度，本系统总体采用封闭分层策略。这意味着架构中的每一层（Layer N）只能调用其直接下层（Layer N-1）所公开的接口和服务，严禁跨层调用。

2.1. 封闭分层的实施规则

表现层 → 应用逻辑层：

表现层（客户端）不能绕过应用逻辑层直接访问领域对象或底层数据库。所有的业务请求（如“发布视频”）必须通过网络接口（RESTful API）发送给应用逻辑层进行身份验证和流程编排。

目的：确保安全性，隐藏后端实现细节，使客户端与服务端解耦。

应用逻辑层 → 领域层：

应用逻辑层负责协调任务，但具体的业务规则判断（如“是否满足升级条件”）必须委托给领域层处理，不得直接操作数据源层的 SQL 语句。

目的：防止业务逻辑泄露到服务脚本中，保持领域模型的纯净和复用性。

领域层 → 数据源层：

领域层定义数据访问接口，数据源层负责实现这些接口。领域层不依赖具体的数据存储技术（如 PostgreSQL 或 Redis 的具体 API），仅依赖抽象接口。

目的：实现依赖倒置，确保更换数据库实现时不影响核心业务规则。

2.2. 策略选择的理由

选择封闭分层基于以下权衡：

降低依赖与连锁效应：

封闭分层建立了严格的防火墙。如果数据源层（Layer 4）的数据库表结构发生变更，其影响仅限于领域层（Layer 3）的映射逻辑，而应用逻辑层（Layer 2）和表现层（Layer 1）无需任何修改。这对于一个处于快速迭代期、数据库结构可能频繁变动的项目至关重要。

增强可维护性：

严格的调用约束使得系统结构清晰，新加入团队的开发人员可以快速理解每一层的职责边界，避免出现“意大利面条式代码”（Spaghetti Code）。

标准化接口：

促使层与层之间定义清晰、稳定的契约（Contracts），有利于自动化测试的开展（例如，可以轻松 Mock 掉数据源层来单独测试领域层）。

2.3. 对性能损耗的应对

封闭分层可能导致性能开销（数据需逐层传递）。针对“乐拍视界”的高性能需求，我们采取以下措施缓解：

数据传输对象 (DTO) 优化：在应用逻辑层将领域对象转换为扁平化的 DTO，减少跨网络传输（表现层与应用逻辑层之间）的数据量。

缓存穿透：虽然逻辑上是封闭分层，但在数据源层内部引入 Redis 缓存。当上层请求数据时，如果缓存命中，则以极低延迟返回，从而抵消分层带来的调用栈开销。

3. 通信风格

为了确保系统的松散耦合和高可扩展性，本系统在不同层级和子系统间主要采用 客户端-服务器风格，并辅以 基于中介的异步通信。

3.1. 客户端-服务器风格

这是系统表现层与应用逻辑层交互的主导风格。

交互模式：

单向依赖：表现层（作为 Client）需知道应用逻辑层（作为 Server）的 API 地址和接口契约，主动发起请求（Request）。

被动响应：应用逻辑层不依赖也不了解具体客户端的实现细节，它处于监听状态，被动接收请求并返回响应。

通信协议：

采用 RESTful API 架构风格，基于 HTTP/2 协议。

理由：Client-Server 风格易于维护。HTTP/2 的多路复用特性解决了移动端高延迟问题，且 REST 风格的无状态性（Statelessness）使得服务端可以轻松进行水平扩展（Scale-out），无需在应用服务器内存中维护客户端的会话状态。

数据交换格式：

JSON：用于控制信令与元数据（如用户信息、视频列表）的传输，因其具备自描述性和易调试性。

二进制流：用于视频流媒体数据的传输，直接利用 HTTP Range 请求支持断点续传和流式播放。

3.2. 基于中介的异步通信风格

针对视频处理（转码、审核）等耗时操作，为了避免阻塞客户端的主线程，采用了中介/代

理（Broker） 模式的变体。

交互模式：

解耦：生产者（应用逻辑层）将耗时任务封装为消息，发送给中介（Message Broker，即消息队列），然后立即向客户端返回“处理中”状态。

异步消费：消费者（转码服务、审核服务）从中介订阅并获取消息进行处理。生产者与消费者互不感知，实现了时间和空间上的解耦。

组件：

消息中介 (Broker)：采用 RabbitMQ

应用场景：

用户上传视频后，系统异步触发转码任务。

用户点赞后，系统异步触发统计数据的更新和推送通知。

3.3. 内部服务通信

在服务端内部（应用逻辑层与各微服务/子系统之间），为了追求极致性能，采用更高效的通信方式。

远程过程调用 (RPC)：

采用 gRPC 框架，基于 Protobuf（Protocol Buffers）序列化协议。

理由：相比文本格式的 JSON，Protobuf 的二进制序列化体积更小、解析速度更快，能显著降低内部服务间调用的网络开销和延迟，符合系统“高性能”的设计目标。

5.3. 子系统分解与逻辑视图

1. 子系统划分

为了应对“乐拍视界”业务的复杂性并支持并行开发，我们将服务端与客户端的逻辑结构进一步细化为以下关键子系统。

1.1. 服务端子系统

这些子系统主要位于应用逻辑层与领域层，共同协作以提供完整的后端服务能力。

用户管理子系统

职责：全权负责与“人”相关的业务逻辑。

功能域：用户注册与登录认证、个人档案管理、实名认证状态维护、以及用户社交

关系链（关注、粉丝列表）的管理。

关键实体：User, Account, Relationship。

内容管理子系统

职责：全权负责核心数字资产（视频与音乐）的生命周期管理。

功能域：视频元数据（标题、描述、标签）的存储与检索、视频流地址的生成与签名、正版音乐库的维护与版权关联。

关键实体：Video, Music, Tag。

社交互动子系统

职责：处理高频的用户交互行为，确保社区活跃度。

功能域：视频的点赞、评论管理、私信发送与接收、分享计数统计。该子系统设计需特别针对高并发写入进行优化。

关键实体：Comment, Like, Message。

推荐引擎子系统

职责：负责“内容找人”的核心算法逻辑，实现个性化分发。

功能域：用户行为日志的收集与分析（构建用户画像）、协同过滤算法的执行、个性化视频流（Feed）列表的生成。

关键实体：UserPreference, RecommendationFeed。

运营与审核子系统

职责：负责平台的合规性控制与营销活动管理。

功能域：内容审核工作流（机审+人审任务分配）、违规内容处理、运营活动（如挑战赛）的配置与发布、Banner 广告位管理。

关键实体：AuditTask, OperationActivity, Report。

1.2. 客户端子系统

客户端依据用户的使用场景进行分区，以支持模块化的 UI 开发与维护。

创作子系统

职责：提供视频生产工具。

功能：摄像头采集、实时美颜滤镜渲染、音频合成、视频剪辑与特效处理、上传任务管理。

消费子系统

职责：提供沉浸式的浏览体验。

功能：视频流播放器内核管理、手势交互处理（上下滑动切换）、弹幕与评论区的渲染。

个人中心子系统

职责：展示用户资产与社交状态。

功能：个人主页展示、作品管理（本地草稿箱与已发布作品）、设置与隐私管理。

1.3. 子系统间的依赖关系

为降低耦合度，子系统间通过定义良好的接口（Well-defined Interfaces）进行交互：

推荐子系统 依赖 用户管理子系统（获取画像）和 内容管理子系统（获取视频池），但仅通过只读接口访问，不修改对方数据。

社交互动子系统 依赖 内容管理子系统（关联视频 ID），采用松散耦合的引用关系。

所有子系统均通过基础设施层提供的通用服务接口访问数据库和消息队列。

2. 客户端架构：MVC 模式应用

客户端被划分为三个核心组件集：模型、视图和 控制器，并依靠传播机制 (Propagation Mechanism) 进行状态同步。

2.1. 模型

定位：客户端的核心，封装了应用的状态（State）和业务逻辑功能。它独立于具体的界面表现形式，不知道具体的视图如何展示数据。

关键组件：

VideoFeedModel：维护当前视频流列表数据、当前播放索引、视频预加载缓冲状态。它负责处理数据的获取逻辑（如“加载下一页”），但不关心视频是全屏展示还是列表展示。

CreationModel：维护拍摄过程中的状态，包括录制时长、当前选中的滤镜 ID、背景音乐（BGM）的播放进度以及临时文件路径。

UserContextModel：维护当前登录用户的身份信息（Token）、全局配置以及未

读消息计数。

职责：

提供数据访问接口（Getters/Setters）。

执行核心逻辑（如计算视频是否已看完）。

在状态发生变化时（如数据加载完成、属性变更），触发通知机制。

2.2. 视图

定位：负责向用户呈现信息。它从模型中获取数据进行渲染，并订阅模型的变化通知以保持界面同步。

关键组件：

MainFeedView：沉浸式视频播放容器。负责渲染视频纹理（Texture）、覆盖在视频上的点赞图标、评论数文本以及进度条。它依据 VideoFeedModel 的数据进行绘制。

CameraPreviewView：拍摄取景框。负责实时显示经过 GPU 处理后的摄像头画面。它根据 CreationModel 中的滤镜状态应用不同的 Shader 效果。

InteractionOverlayView：显示评论列表弹窗、分享面板等交互元素。

职责：

初始化：在创建时从模型读取初始状态进行渲染。

更新：接收到模型的更新通知后，重绘受影响的 UI 区域（例如，当点赞数增加时，仅刷新数字文本，不重载视频）。

关联：创建并持有对应的控制器实例。

2.3. 控制器

定位：处理用户输入。它将用户的操作映射为模型的更新指令或视图的显示策略变更。

关键组件：

FeedGestureController：监听屏幕的触摸事件。

行为：识别“上滑”手势，调用 VideoFeedModel.nextVideo()；识别“双击”手势，调用 VideoFeedModel.likeVideo()。

RecorderController：监听拍摄按钮的点击事件。

行为：按下按钮时调用 CreationModel.startRecording()，松开时调用 sto-

pRecording()。

NavigationController：处理底部 Tab 栏的点击，负责在不同视图（如首页 vs. 个人中心）之间切换，这属于策略模式的应用（改变视图的显示而不改变模型数据）。

职责：

解析底层事件（如 Touch, Click）。

向模型发送指令以更改状态。

直接向视图发送指令以改变非数据性的显示状态（如显示/隐藏加载动画）。

2.4. 传播机制

定义：这是 MVC 架构中连接模型与视图的纽带，实现了观察者模式 (Observer Pattern)。

实现方式：

因为技术平台选择了 Qt/QML，我们利用其的 Signal & Slot (信号与槽) 机制作为传播机制。

交互流程：

1. 注册：MainFeedView 在初始化时连接到 VideoFeedModel 的信号（如 dataChanged）。

2. 变更：FeedGestureController 修改模型（如切换视频）。

3. 通知：VideoFeedModel 修改内部索引后，发射 currentVideoChanged 信号。

4. 更新：MainFeedView 的槽函数被自动触发，读取新的视频 URL 并命令播放器加载新资源。

优势：确立了模型与视图的松散耦合，模型无需持有视图的引用即可通知视图更新。

3. 服务端架构

服务端架构核心由应用逻辑层与领域层组成，通过定义明确的接口和数据传输对象（DTO）进行协作。

3.1. 应用逻辑组件

定位：作为服务端的“控制器”角色，是客户端请求的直接入口点。

设计模式：采用 Facade（外观）模式或 Service Layer（服务层）模式。每个子系统对

外暴露一组粗粒度的服务接口。

关键组件：

UserAppService：负责协调用户注册、登录认证流程。

职责：调用认证工具校验 Token，调用领域层创建用户实体，生成并返回 UserDTO。

VideoPublishService：负责视频发布的全流程编排。

职责：接收客户端上传请求 → 验证用户上传权限 → 调用领域服务创建视频记录 → 发送“转码”消息到消息队列 → 返回“发布中”状态给客户端。

InteractionAppService：处理点赞和评论请求。

职责：验证操作合法性（如是否重复点赞），调用领域层更新计数，并处理缓存更新策略。

FeedAppService：负责首页视频流的组装。

职责：调用推荐子系统获取视频 ID 列表，批量查询视频元数据和作者信息，组装成客户端所需的聚合对象。

核心职责：

事务管理：定义业务操作的事务边界，确保数据的一致性。

DTO 转换：负责领域对象与数据传输对象（DTO）之间的相互转换，屏蔽领域模型内部细节。

流程编排：不包含核心业务规则（如“什么样的视频算违规”），仅负责按顺序调用领域对象或外部服务。

3.2. 领域模型组件

定位：系统的核心资产，包含所有业务逻辑、状态校验和规则，独立于任何应用流程或技术框架。

设计模式：采用 Rich Domain Model（充血模型），避免贫血模型。

关键组件：

实体：

Video：封装视频的核心状态（Draft, Transcoding, Published, Banned）。

包含业务方法如 publish()（发布）、addLike()（增加点赞）。

User：封装用户状态。包含方法如 follow(targetUser)（关注某人）。

AuditTask：封装审核任务。包含方法如 approve()（通过）、reject(reason)（驳回）。

值对象：

VideoUrl：封装视频链接，包含验证 URL 格式和有效性的逻辑。

Tag：视频标签，不可变对象。

领域服务：

RecommendationService：封装核心推荐算法逻辑，计算用户与视频的匹配度。这是不属于单一实体的复杂领域逻辑。

仓储接口：

VideoRepository、UserRepository：在领域层定义持久化操作的接口契约，但不包含实现（实现位于数据源层）。

核心职责：

业务规则校验：例如，在 User.follow() 方法内部校验是否已关注、是否达到关注上限。

状态一致性维护：确保实体在任何时刻都处于合法的业务状态。

5.4. 并发设计与进程视图

1. 并发需求识别

2. 进程与线程管理

3. 同步与通信

5.5. 物理架构与部署视图

1. 硬件节点映射

2. 软件组件分配

3. 网络拓扑与通信基础设施

5.6. 数据管理策略

1. 持久化存储方案

2. 数据库管理系统选择

5.7. 系统设计标准与规范

1. 用户接口标准

2. 内部接口与编码规范

第6章详细设计

后记

正文内容，方正仿宋，小四，首行缩进。正文内容，方正仿宋，小四，首行缩进。正文内容，方正仿宋，小四，首行缩进。

参考文献

[1] YOUNG.RSS 是什么？[EB/OL]. http://jingpin.org/what-is-rss/.

[2] 杨博，彭博.RSS 提要分析与阅读器设计[R].成都：四川大学计算机学院，2007：42-43.

[3] 逸出络然.RSS 技术的原理[EB/OL].http://yclran.blog.163.com/blog/static /97945496200911103411155 8/.

[4] 佚名.Qt 是什么[EB/OL]. http://qt.nokia.com/title-cn.

[5] 佚名.Model/View Programming[EB/OL]. http://doc.trolltech.com/4.6/model-view-programming.html.

[6] [加拿大]Jasmin Blanchette[英]Mark Summerfield 著 闫锋欣,曾泉人,张志强译.

[7] C++ GUI Qt4 编程（第二版）[M].电子工业出版社：2008:182-206,291-305.

[8] 佚名.XML Processing[EB/OL]. http://doc.trolltech.com/4.6/xml-processing.html.

[9] Michael Blala James Rumbangh 著.UML 面向对象建模与设计（第 2 版）[M].北京：人民邮电出版社,2006:136-235.

[10] 胡海静,王育平,等. XML 技术精粹[M]. 北京：机械工业出版社，2001:17-19.