

学校代码: 10004

密级: 公 开

北京交通大学

BEIJING JIAOTONG UNIVERSITY

硕士专业学位论文

直播聚合平台的设计与实现

作者姓名 刘新奇

工程领域 软件工程

指导教师 刘 铎 副教授

培养院系 软件学院

二零一九年五月

北京交通大学

硕士专业学位论文

直播聚合平台的设计与实现

Design and Implementation of Live Streaming Aggregation Platform

作者：刘新奇

导师：刘 铎

北京交通大学

2019年5月

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：刘新奇

导师签名：刘铮

签字日期：2019年6月21日

签字日期：2019年6月21日

学校代码：10004

密级：公开

北京交通大学

硕士专业学位论文

直播聚合平台的设计与实现

Design and Implementation of Live Streaming Aggregation Platform

作者姓名：刘新奇

学 号：17126178

导师姓名：刘 铎

职 称：副教授

工程硕士专业领域：软件工程

学位级别：硕士

北京交通大学

2019年5月

致谢

时光荏苒，两年的硕士求学生涯已接近尾声，历时一年的时间，我终于将这篇论文如期顺利撰写完成，几多欣喜，几多坎坷，这两年的求学时光都是我生命中最宝贵的财富和最精彩的人生体验。

论文的撰写过程中我遇到了很多困难与障碍，都在老师和同学的帮助下顺利解决。非常幸运可以到公司实际接触真实产品的开发项目，这一个全新领域的学习与实践也让我自己在软件工程这个专业领域的探索更为专业和深入。

尤其要感谢我的导师刘铎和我的企业指导老师徐德立，他们在论文的撰写上给予我无私的指导和帮助，给我提供论文改进与优化的方向，在我论文完成的过程中本着严谨的态度，为我提供了学习的平台，并提出了很多宝贵的意见。还要感谢我的同学和朋友，尤其是张建成、贾冠一、殷国栋在我论文撰写过程中给予了我很多帮助，提供了很多学习的素材，还在论文的撰写和排版等过程中提供热情的帮助。另外，在校图书馆查找资料的时候，图书馆的老师和学生也给我提供了很多方面的支持与帮助，感谢他们辛勤的工作与无私帮助。

最后，特别要感谢刘美琪同学，在我论文的撰写过程中给予了我很大的支持。感谢一路陪伴我的父母和亲人，是你们的不断鼓励与包容支持让我能在求学道路上不断前进！

摘要

近年来，互联网行业直播类节目大为火热，各类直播平台层出不穷，作者所实习的部门基于公司所掌握的技术可以实现有效地解析直播地址从而实现直播聚合功能，经过精心的产品设计，研发了一款满足当前观看直播的用户群体需求的直播聚合应用，极大地方直播用户观看直播节目。并且积极寻找新的突破点，在功能上进行不断完善与创新。

本论文主要介绍为用户提供全网最全直播功能的直播聚合平台的设计与实现，不仅能够实现直播应用的常规功能还能实现聚合众多直播平台于一家的功能。本系统分为直播门户端与后台管理端，核心业务用例分别为：直播门户端的主界面功能模块、直播搜索模块、直播播放模块、个人功能模块以及直播后台管理端的直播聚合模块、直播数据管理模块、账号信息管理模块。

其中作者设计与实现的部分有：（1）主界面功能模块：能够给用户提供了直播列表等主界面信息以及直播详情页的展示。（2）直播播放模块：能够实现汇聚的各家直播平台的直播内容正常播放。（3）直播搜索模块：用户能够按照主播名字、所属直播平台、直播间 ID 等条件进行直播检索。（4）直播聚合模块：能够将不同平台的直播数据聚合在一起。（5）用户登录模块：用户能注册账号登录或者使用第三方登录本应用，然后进行关注、收藏等操作。（6）直播数据管理模块：能够借助服务器功能实现违规敏感内容的管理以及评论管理。以上所有功能点都依据软件工程的开发流程进行实现。本系统基于 Java 语言开发，客户端开发环境采用 Android Studio，前期使用网络通信框架 OkHttp 提供高性能，以及 Volley 队列管理方案，后期使用 RxJava 配合 Retrofit 的方式实现异步网络请求。整体架构采用的是 MVC，业务逻辑和 UI 逻辑的分离，契合高内聚低耦合的设计原则。选用 MySQL 作为数据库。提高了该系统的灵活性和可扩展性，通过地址反解析以及加入不同编解码器来兼容多种视频格式，实现直播聚合功能。通过对 Ijkplayer 进行二次开发来实现播放扩展功能。

本论文完成的直播聚合平台，经过对各模块进行功能测试，均已达到了理想的效果，可以为使用该平台的用户提供便捷的直播服务。

关键词：直播；聚合平台；Android；Ijkplayer

ABSTRACT

In recent years, The Internet industry live shows are very popular, all kinds of live platform emerge in endlessly, the Author's internship department based on the technique that company has can effectively use all kinds of live address resolution and aggregation, Through the product design carefully, developed a live application which could meet the current polymer applications, has made great convenience for users who love watching live broadcast. And looking for new breakthroughs in the function actively to achieve more improvement and innovation.

This system can not only realize the routine functions of live broadcast applications, but also realize the function of aggregating numerous live broadcast platforms into only one. This system is divided into live portal side with the background management, a total of seven core business use cases, respectively, were Main interface function module, live broadcast search, live broadcast, personal module, live broadcast aggregation, live broadcast data management, account information management.

The part that author participated in the design and implementation includes: (1) Main interface display module. (2) Live broadcast module. (3) Live broadcast retrieval module. (4) Live broadcast aggregation module. (5) User login module. (7) Live broadcast data management module. This system is based on Java language development, client development environment using Android Studio, network communication framework OkHttp is used to provide high performance, as well as Volley queue management scheme. The overall architecture adopts MVC, The separation of business logic and UI logic fits the design principle of high cohesion and low coupling. Choose MYSQL as the database. It improves the flexibility and extensibility of the system, and through address reverse resolution and adding different codecs to be compatible with a variety of video formats to realizes the function of live broadcast aggregation. Through the secondary development of Ijkplayer to achieve playback extension function.

The live broadcast aggregation platform completed in this paper has achieved the desired effect after functional tests of each module, and can provide convenient live broadcast service for users of the platform.

KEYWORDS: Live broadcast; Aggregation platform; Android; Ijkplayer

目录

摘要	v
ABSTRACT.....	vii
1 引言	1
1.1 项目背景	1
1.2 国内外发展现状	1
1.3 目的及意义	3
1.4 论文的主要内容和组织结构	4
1.5 本章小结	4
2 相关理论及技术概述	5
2.1 Android MVP 模式.....	5
2.2 OkHttp 和 Volley	5
2.3 RxJava 和 Retrofit	6
2.4 Spring Security	6
2.5 MVC 架构	7
2.6 地址反解析技术和 Ijkplayer.....	8
2.7 本章小结	10
3 直播聚合平台需求分析	11
3.1 直播聚合平台概述	11
3.1.1 产品特点分析	11
3.1.2 用户特征分析	12
3.2 功能性需求分析	13
3.2.1 直播平台门户端	13
3.2.2 直播后台管理端	18
3.3 非功能性需求分析	21
3.4 本章小结	22
4 直播聚合平台概要设计	23
4.1 系统设计目标	23
4.2 系统架构设计	24
4.2.1 系统逻辑架构	24
4.2.2 系统客户端技术实现架构	24

4.2.3 系统管理端技术实现架构	26
4.3 系统总体功能结构	26
4.4 数据存储设计	28
4.4.1 数据库设计	28
4.4.2 本地数据缓存	33
4.4.3 服务器接口	35
4.5 本章小结	38
5 直播聚合平台详细设计与实现	39
5.1 主界面功能模块	39
5.1.1 加载主界面	39
5.1.2 直播详情界面	45
5.2 直播搜索模块	49
5.3 直播播放模块	53
5.4 用户登录模块	56
5.5 直播聚合模块	60
5.6 直播数据管理模块	62
5.6.1 评论管理	62
5.6.2 违规内容管理	63
5.7 本章小结	64
6 系统测试	65
6.1 测试环境	65
6.2 功能性测试	66
6.3 非功能性测试	71
6.4 本章小结	72
7 总结与展望	73
7.1 工作总结	73
7.2 工作展望	74
参考文献	75
作者简历及攻读硕士学位期间取得的研究成果	77
独创性声明	79
学位论文数据集	81

1 引言

本章主要阐述直播聚合平台的项目背景，在国内外与本课题相关的产品的研究与发展现状，简要叙述了项目的目的和意义，最后说明了本论文的组织结构与主要内容。

1.1 项目背景

当今互联网行业直播类节目大为火热，各类直播平台层出不穷，仅 2018 上半年，一直都高速增长的中国网络直播带宽指数在第二季度便达到 332，同比增长 41.9%。网络直播带宽总量在 6 月创出新峰值，同比增长达 48%。经过分析，中国移动直播市场增长的四大主要因素为消费升级和不断增加的购买力；智能手机中使用的更先进的技术；网络游戏和电竞行业在年轻人中日益普及；移动支付日益普及。

目前，中国是全球最大的直播市场。2018 年平均月活跃用户数量为 2.97 亿，预计 2022 年将增长到 5.18 亿，复合年增长率为 24.6%。截止 2018 年中国移动直播行业市场规模已经达到 122.8 亿元，实现了突破九十亿元大关的目标。经过相关专家预测这一市场规模还将进一步增长，预计在 2019 年实现增幅 159%，规模 317.8 亿元，导致移动直播行业市场规模如此快速增长的因素主要还是行业技术的成熟以及行业盈利手段逐渐多元化。

基于这样的环境背景，加上对于广大直播用户对于直播应用的需求进行了广泛地调研，本人实习所在的部门决定基于所掌握的技术有效的将各类直播进行地址解析及聚合，经过精心的产品设计，研发了一款满足当前观看直播的用户群体需求的直播聚合应用，极大地方便了喜爱直播行业的用户观看直播。并且积极寻找新的突破点，在功能上进行不断完善与创新。

1.2 国内外发展现状

目前，整个直播行业正处在一个高速发展的时期，无论是传统的秀场直播和游戏直播，还是现在较为时尚的泛娱乐直播，直播应用已经覆盖了各个领域。观看直播的用户也在逐年成倍数增加。

目前较为有代表性的直播平台主要是一些主流的娱乐直播平台，主要分为几

类：以 9158 直播平台、YY 直播平台为代表的娱乐秀场直播平台；以熊猫直播平台、斗鱼直播平台、虎牙直播平台为代表的游戏电竞直播平台。以搜狐体育、腾讯体育直播平台为代表的体育竞技类直播平台，以及以秒拍、抖音、一直播为代表的明星网红直播平台^[1]。这些平台各有特点，经历了互联网飞速发展的整个过程，也使得他们形成的产业链更为完备。每个类型的直播平台都有自己的优势。

娱乐秀场直播平台主要特点是发展最早，多数依靠音乐经济以及美女经济，目前市场格局相对稳定，因为其技术学习难度不高，所以主播复制难度低，复制率极高，从而也就导致竞争较为激烈。

游戏电竞直播平台主要特点是发展迅速，这不得不得益于 2011 年拳头公司研发出的一款叫做《英雄联盟》的游戏，这款游戏迅速席卷全球，在腾讯公司的加持下也立即称霸中国，成为当时乃至今日都炙手可热的竞技类游戏，因为它的火爆，很多职业电竞选手在退役后选择做一名专业的游戏主播，从而不断催化游戏直播的行业，形成了如今的火热局面，因为游戏专业性较强所以主播复制难度大，竞争力大于娱乐秀场直播。

体育竞技类直播平台主要特点是时效性较强，很多用户都为资深体育迷，或者为了篮球、或者为了足球、或者为了羽毛球，种类较多但是专业性较强，这种直播平台往往是配以解说进行第三方转播，拥有一定版权，基本不可复制。

明星网红直播平台的主要特点就是类型繁多、较为新鲜有吸引力，例如直播吃饭、直播户外聊天、直播聊天看电影，虽然是很日常的活动，但是一旦打上主播的特点标签后就会变得定位清晰，发展迅速。

通过对传统直播以及现行的新潮直播进行分析，很多公司便将目光聚集在了将这些直播同类型或者不同类型聚合在一起的直播聚合平台领域，那么目前存在的几款直播聚合平台主要有以下几个：

(1) 游久直播应用。这是由游久网研发的直播聚合应用，发布于 2017 年 9 月，软件大小：20M。游久直播应用主要提供赛事高清直播、图文、沙盘直播，热门主播时间预告，其定位是老牌电竞门户网站游久网为广大电竞爱好者推出的各类电竞比赛资讯、比分数据、现场实时播报的赛事类信息应用。有上万的用户量。它的缺点是聚合直播类型较少，内容丰富度不够，目前只有游戏类直播。功能也较为单一，不能实现录屏等功能^[2]。

(2) 龟龟直播应用。发布于 2018 年 2 月 10 日，软件大小：5.95M。平台聚合了最新高清流畅的游戏直播和电子竞技赛事直播，游戏直播主要包括《英雄联盟》直播、《穿越火线》直播、《DOTA2》直播、《地下城与勇士》直播等热门游戏直播和赛事直播，同时也具有美女娱乐直播和顶级游戏解说直播，此直播应用能够为玩家提供内容丰富、推送及时的游戏直播服务，具备如下优点功能：1) 能够

自动刷新直播列表信息，实现数据间隔分钟刷新，让用户看到最新的直播；2) 能够实现全球直播，直播范围不局限于国内；3) 具备智能状态，能够帮助用户获取主播实时状态，使得用户更方便快捷地获取主播的上下线时间。

(3) Z 直播应用。是一款安卓平台上的看直播神器。它是一款综合型的聚合直播应用，目前支持的平台包括：斗鱼、龙珠、bilibili、企鹅、虎牙直播等全网直播平台，其清晰度可调节，支持弹幕模式以及去广告清爽版。这一款直播平台的优点在于能够聚合比较多的直播应用，例如虎牙、斗鱼、企鹅、熊猫等等；并且还可以进行悬浮小窗口播放，可以一边浏览别的内容一边看；同时支持搜索功能，搜索网络资源，输入关键字 点击搜索即可。缺点是这款直播应用目前还没有推荐功能以及用户历史记录功能^[3]。

以上几款直播应用是目前功能比较齐全的直播聚合平台应用，在娱乐行业的推动下，随着技术的进步以及现代社会资本的涌入，互联网直播将以更快的速度实现技术飞跃，不断提升用户体验，刷新直播定义，这也带给了直播聚合平台行业无限的发展潜力。

1.3 目的及意义

直播聚合平台这一项目来自作者所在的实习公司，历时半年完成这一项目，在本项目的设计与开发过程中，作者全程参与了各个阶段，并独立完成了诸多功能模块。经过需求分析阶段的调查研究结果，本项目的主要目的是为了解决以下问题：

(1) 直播平台过多，用户选择不方便的问题。

(2) 对应主播的直播只能在对平台观看，想看不同主播需要下载多个应用，从而造成的资源浪费以及占用存储空间的问题。

(3) 主播跳槽后，得到通知不及时，而且往往需要更换直播应用的问题。

针对这些问题，作者对直播聚合平台这一项目进行了完备的分析与设计，最终系统实现后的意义有：

(1) 能够更好的方便用户观看自己喜爱的主播直播，解决主播应用太多导致的多个应用占用内存问题，解决主播跳槽后需要更换直播应用问题。

(2) 能够为公司其他产品进行导流，例如部门游戏应用，同时可以沉淀忠实用户，为以后进行新项目的研发打下基础^[4]。

(3) 提高观看直播用户群体的观看体验，提高用户观看喜爱的直播内容的效率。

1.4 论文的主要内容和组织结构

该项目核心的业务有主界面显示：能够实现加载主界面和直播详情界面展示的功能；直播聚合：可以聚合多家直播；直播播放：能够实现正常的直播播放，同时实现后台播放、小窗口播放、录屏播放等功能；直播搜索：能够按照用户的习惯去搜索直播；个人功能：主要实现用户登录、收藏转发、评论管理等核心功能，论文着重介绍以上几个功能点的设计与实现。

本论文将从如下几个方面对直播聚合平台的设计与实现进行阐述：

第一章：引言，主要介绍了论文的项目背景，对直播聚合平台同类产品进行调研，描述了论文的目的及意义，向读者清楚阐述本论文的主要内容与组织结构。

第二章：相关理论及技术概述，对直播聚合平台实现与设计的过程中所需要用到的相关理论和技术进行说明。

第三章：直播聚合平台需求分析，介绍本系统的需求、不仅包括功能性需求，还包括非功能性需求，主要对所设计的功能进行了用例描述。

第四章：直播聚合平台概要设计，主要包括系统的设计目标、整体架构设计、系统功能模块划分和数据库设计。

第五章：直播聚合平台详细设计与实现，结合流程图、类图和时序图及文字说明，详细介绍了直播聚合平台本人参与实现功能的核心功能点实现细节。

第六章：系统测试，对直播聚合平台的测试环境进行说明，针对功能性测试和非功能性测试分别给出测试用例与测试结果。

第七章：总结与展望，对个人主要工作进行总结，明确未来需要完成的目标。

1.5 本章小结

本章主要介绍了本论文的项目背景，经过对直播聚合平台同类产品进行调研得到了该项目的潜力以及未来广阔的可能性，同时本章也描述了论文的目的及意义，向读者清楚地阐述了本论文的主要内容与组织结构。使读者更加了解本系统。

2 相关理论及技术概述

本章对开发过程中所使用的相关理论与技术架构进行阐述，只有熟练掌握了这些理论技术知识才能够更好地去设计与实现系统。主要包含客户端的 Android MVP 模式、OKHTTP 和 Volley，RxJava 和 Retrofit，以及后台服务端的 Spring Security、MVC 设计模式，地址反解析技术和 Ijkplayer 六个部分。

2.1 Android MVP 模式

通过了解 MVC (Model View Controller) 模式可以去更好地理解 MVP (Model View Presenter) 模式。它实际上是 MVC 模式在 Android 上的一种变体，因为按照 MVC 模式来说，Activity 应该属于 View 这一层。但是实质上它既承担了 View 又包含了 Controller，这对于开发与维护来说并不友好，所以为了降低耦合度，需要把 Activity 的 View 和 Controller 抽离出来就变成了 View 和 Presenter，这就变体成为了 MVP 模式。

MVP 设计模式在 Android 项目中承担着非常重要的角色，它可以做到极大地优化 Activity 和 Fragment。具有如下优点：

- (1) 修改视图完全不影响模型，可以做到模型与视图的分离。
- (2) 因为所有的交互都发生在 Presenter 内部，所以会使得模型的使用效率更高。
- (3) 因为视图的变化总是比模型的变化频繁，所以如果使用 MVP 模式可以使得一个 Presenter 在不需要改变 Presenter 逻辑的情况下就被用于多个视图，那么这个特性便会非常实用。
- (4) 可以脱离用户接口来测试这些逻辑，只需要把它们放在 Presenter 中便可实现。

2.2 OkHttp 和 Volley

OkHttp 是一个开源项目，用来处理网络请求，是安卓端最火热的轻量级框架。它允许连接到同一个主机地址的所有请求，这样可以提高请求效率，共享 Socket，减少对服务器的请求次数，通过连接池减少了请求延迟，能够减少重复的网络请求，一般是通过缓存响应数据来实现，能够减少对数据流量的消耗，同时还可以自动去处理 GZip 压缩。

Volley 是 2013 年 Google I/O 上发布的一款网络框架，基于 Android 平台，能使网络通信更快，更简单，更健全。它的优点是默认 Android 2.3 及以上基于 `HttpURLConnection`，2.3 以下使用基于 `HttpClient`；能够提供符合 Http 缓存语义的缓存机制；能够提供请求队列的优先级排序；能够提供多样的取消机制；能够提供简便的图片加载工具，综上所述确实是一个优秀的框架。

由于 Volley 和 OkHttp 各有优缺点，Volley 不支持 HTTPS，而 OkHttp 支持 HTTPS。Volley 的扩展性又非常强，因此可以将 Volley 和 OkHttp 结合使用，这也是目前较为主流的 Android 网络架构。

同时，在系统研发后期还用到了新的网络请求技术 RxJava+Retrofit，下一小节将进行详细地介绍。

2.3 RxJava 和 Retrofit

RxJava 是一个异步框架，基于观察者设计模式，被观察者 `Observable`，观察者 `Observer`，它可接收 `Observable` 通知，被观察者通过 `subscribe` 方法申请订阅注册观察者。它的很大的一个优点就是简洁，而异步操作很关键的一点也是程序的简洁性，因为在调度过程比较复杂的情况下，异步代码经常会既难写也难被读懂。Android 创造的 `AsyncTask` 和 `Handler`，其实都是为了让异步代码更加简洁。RxJava 的简洁的与众不同之处在于随着程序逻辑变得越来越复杂，它依然能够保持简洁。

而 Retrofit 是 Square 公司开发的一款针对 Android 网络请求的框架，Retrofit2 底层是基于 OkHttp 实现的。

异步请求相对同步请求更简便和快捷，开发者只需要在 `OnResponse` 和 `OnFailure` 中处理对应回调即可。但是这种回调方式本身也有不方便的地方。因为回调直接是在 UI 线程，如果在 `OnResponse` 中回调的数据还要进行耗时操作，比如和数据库中的数据对比，或者返回结果是图片的 Url 需要再次通过网络请求得到网络图片，上述回调的方式就需要再开线程来处理，而使用 RxJava 的话，其优点在于异步操作和线程切换，这样就可以比较优雅和轻松的解决上述问题。所以在系统设计与实现的后期采用了此套网络请求技术。

2.4 Spring Security

Spring Security 基于 Spring 框架^[5]，此安全框架基于 Spring AOP 和 Servlet 过滤器，可以为开发者提供声明式的安全访问控制解决方式，它提供全面的安全性解决方案，Spring Security 的主要核心功能为认证和授权。一般系统使用来此框架

来保证系统安全性。

用户认证和用户授权这两个部分共同组成 Spring Security。一般都是用来开发安全级别更高的应用。用户认证即验证此用户是否有合法权限来登录访问此系统，在用户按照要求提交用户名和密码后系统会通过校验用户名和密码来完成认证过程。

用户授权就是在用户认证的基础上将不同用户按照所赋予的权限不同区分开来。即使在同一个系统中很多用户都是会被赋予不同权限的，例如有的用户只能查看浏览某些信息^[6]，而有的用户可以修改某些信息，一般系统都会分为管理员和普通用户。他们分别对应各自的不同权限。所以 Spring Security 的主要核心功能为认证和授权，系统的架构也是在这两个核心功能基础上去实现的。流程图如图 2-1 所示。

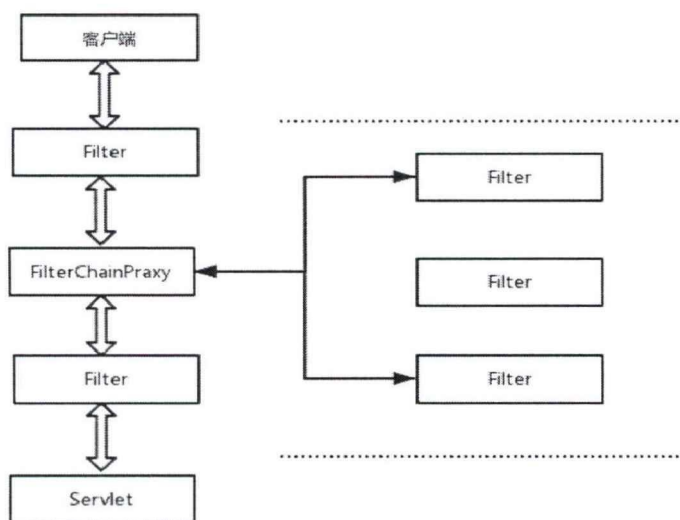


图 2-1 Spring Security 工作流程图
Figure 2-1 Flowchart of Spring Security

2.5 MVC 架构

MVC 是一种软件设计模式，最初应用于 JSP (Java Server Pages) 开发中，随后成为 J2EE 平台的主要设计模式，现在，在大型的复杂项目开发和 Web 应用开发中应用较为广泛。MVC 顾名思义，它主要分为三个部分：视图、模型、控制器。

(1) 视图。用户能够看到并产生交互的页面也就是应用程序的表示层为视图。在 Web 应用中，交互界面可能是 HTML 界面、XML 界面和应用 let 界面等，一个应用可以对应多个视图，通常情况下 MVC 对视图操作包括数据采集和处理。通

过 HTML 元素组成一个界面，就是视图。HTML 在应用中仍然是视图里的关键部分，当然，也出现大量的新兴技术，其中有 Macromedia Flash、类似 WML、XML/XSL、XHTML 等的部分标识语言以及 Web Services。

(2) 模型。模型代表了业务规则以及企业的数据库。模型在 MVC 的组成中处理任务最多。它能够使用类似 EJBs、ColdFusion、Components 的构件对象对数据库进行处理。因为模型和数据格式没有必然联系，返回的数据是一个中立体，而且只要编写一次模型的代码就能够在多个视图起到重要作用，所以不仅不同视图的数据能够由一个模型供给，还使代码的重叠程度大大降低。

(3) 控制器。这是收取使用者录入的数据并且使用视图、模型来实现其需求的装置。控制器在向外传送 HTML 表单或者单击 Web 页面中的超链接时，并没有进行输出或者处理，仅仅是收取请求以及找到合适模型构件^[7]来解决请求，再通过已经决定的用视图展示模型处理后产生的数据。

MVC 分层不仅能够使得不同的开发人员可同时开发视图、控制器逻辑和业务逻辑，还能简化分组开发。其结构图如图 2-2 所示。

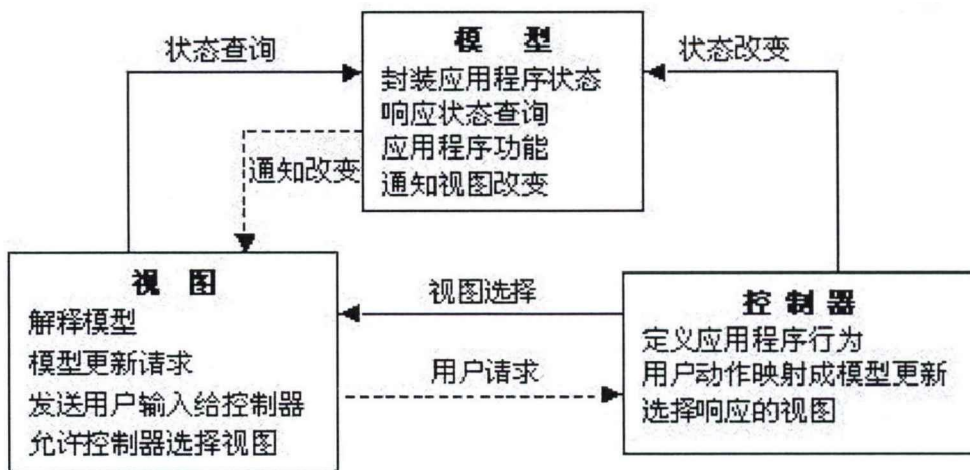


图 2-2 MVC 设计模式结构图
Figure 2-2 MVC Design Pattern Structure Diagram

2.6 地址反解析技术和 Ijkplayer

地址反解析技术是团队通过对直播聚合平台核心代码的编译以及经验总结形成的较为系统的解析直播地址的方法，在最初的技术探索过程中，主要是以斗鱼直播平台的应用为例进行研究，然后成功找到了一套地址解析的技术模板，通过直播地址反解析编码将不同直播平台的直播地址进行解析，然后使用 Ijkplayer 加

入不同解码器^[15]来兼容不同直播格式，因为此项技术涉及到公司内部保密信息，故只作介绍不作详细阐述。

Ijkplayer 是一个基于 ffmpeg 的轻量级 Android/iOS 直播播放器，实现了跨平台的功能，API 易于集成；编译配置可裁剪，方便控制安装包大小。Ijkplayer 支持硬解码和软解码。在本平台的直播聚合方面：通过打包编译 Ijkplayer，对 Ijkplayer 进行二次开发，加入不同编解码器来兼容不同平台的多种直播格式，保证应用在不同平台直播格式不受影响。深入直播流直播数据解析流程，在解析出直播数据和音频数据时，从当前直播关键帧开始，对直播流进行录制。满足直播录制的用户需求。播放器的基本框架图如图 2-3 所示。

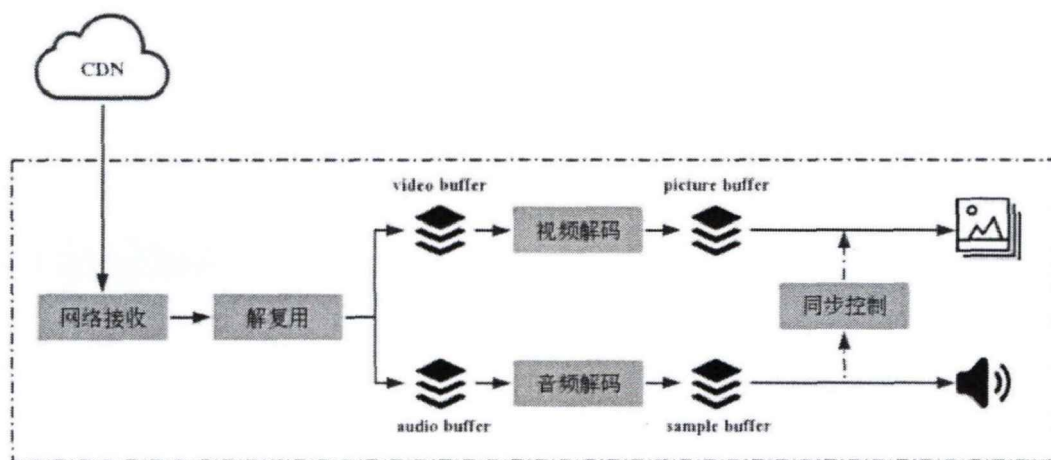


图 2-3 Ijkplayer 播放器基本框架图
Figure 2-3 Ijkplayer Basic Frame Diagram

Ijkplayer 底层是基于 ffplay 的，首先需要了解 ffplay^[16]的代码处理流程。ffplay 是 ffmpeg 项目提供的播放器示例。Ijkplayer 在底层重写了 ffplay.c 文件,并且增加了对移动端硬件解码部分，视频渲染部分，以及音频播放部分的实现，这些部分在 Android 和 IOS 下有不同的实现对比图如图 2-4 所示。

Platform	硬件解码	视频渲染	音频播放
IOS	VideoToolBox	OpenGL ES	AudioQueue
Android	MediaCodec	OpenGL ES、ANativeWindow	OpenSL ES、AudioTrack

图 2-4 Android 和 IOS 的不同实现对比图
Figure 2-4 Different Implementations for Android and IOS

2.7 本章小结

本章主要对直播聚合平台系统中的关键技术及其在系统中应用进行了阐述，系统客户端在 Android 环境下开发，通过模块化设计思路提高了该系统的灵活性，使用当前较火的 MVP 模式提高效率，前期使用 OKHTTP 和 Volley 的架构技术完成网络请求，后期则使用了 RxJava 配合 Retrofit 的方式实现异步网络请求。后台服务端则通过 MVC 设计模式解耦业务逻辑与 UI 逻辑，契合高内聚低耦合的设计原则。使用 Spring Security 来确保业务层的安全性，通过地址反解析技术以及打包编译 Ijkplayer 实现直播聚合功能，通过对 Ijkplayer 进行二次开发来实现播放功能。数据库使用的是 MySQL，通过 GreenDAO 可以避免手动编写大量的 SQL 语句，而直接通过其封装好的方法简单快速地实现数据库的操作。

3 直播聚合平台需求分析

需求分析是软件项目中最根本也是最重要的部分，如果需求有误，则最终设计出来的产品将无法解决用户问题，自然在市场争夺中也不占优势。需求的本意就是应用程序或者项目必须提供的一些功能特性，在开发过程中，需要通过需求指导开发以及用户反馈不断确定需求方向^[8]，并确保项目朝着正确的方向推进。需求必须具备的基本属性有：清晰，没有歧义，一致，优先级排序及可验性。

3.1 直播聚合平台概述

在对系统进行需求分析之前，首先需要对产品的特点和产品的使用对象进行深度的分析，只有当研究者深入的了解了产品的特点和用户的特点，才能将需求分析做到极致，本小节将对直播聚合平台的特点和用户群体的特点进行深入的分析。

3.1.1 产品特点分析

现在直播界有很多主流直播平台，例如以游戏直播为主要类型的虎牙直播、企鹅直播、斗鱼直播等，每个平台都有自己的人气主播，而每个主播只能在自己所属的固定直播平台进行直播，这就就会出现很多需要解决的问题比如：直播平台过多，用户选择不方便的问题；又或者对应主播的直播只能在对应平台观看，想看不同主播需要下载多个应用从而造成的资源浪费问题^[17]，所以一个合格的直播聚合平台需有如下几个功能：

(1) 主界面功能：主要是直播的主界面加载，要有新用户引导、并且能够显示直播列表，显示导航栏，并且能够展示直播详情页

(2) 直播播放：可以观看当前所有主流直播公司的在线直播，实现实时同步播放，同时可以实现挂起应用后还能听到语音的后台播放、退出当前直播房间后能够缩小在左下角的小窗口播放功能、以及用户能对当前正在播放的直播录制的功能。

(3) 直播搜索：要有个性化搜索、精确搜索、模糊搜索等功能。用户可以通过个性化设置搜索喜欢的直播栏目从而获得推荐内容。

(4) 个人功能：能够实现用户登录，能够关注该主播，查看观看历史、收藏该直播。用户能够对于自己感兴趣的直播内容进行评论等。后台管理系统也可以

对用户的个人信息、收藏、关注进行管理。

(5) 直播聚合：资源探测后进行加密地址反解析过程，加入不同编解码器来兼容不同平台的多种直播格式。设立合理的策略做到主播与原平台在线同步。避免出现主播在该直播平台上线本人却显示主播不在线的情况。

(6) 直播数据管理：管理员通过后台对于直播内容是否涉及敏感内容的管理、对于用户评论内容管理、对于用户录屏文件的管理。

(7) 账号信息管理：后台能够实现权限认证、能够实现对于用户个人信息的管理、对于用户收藏信息的管理、管理员还将实现对于用户投诉内容的管理。

3.1.2 用户特征分析

直播聚合平台的使用者主要包括直播观众以及系统管理员。系统管理员拥有最高权限，可以对直播平台、主播、用户、直播内容进行管理。观看直播的使用者主要为热爱游戏、影视直播的群体，多为年轻人，以及希望节省移动设备存储空间的人群。

经过调查，本文所设计与实现的直播聚合平台最主要的服务对象群体、观众有如下几个特征：

(1) 用户群体年轻：年轻人的思想新潮，接受新事物的机会较多也较快，且因为他们的适应力较强往往能够在新鲜事物上做出很多成绩。经过调查，目前最受观众喜爱的主播平均年龄都在 30 岁以下，而且观众的平均年龄也在 25 岁以下，从主播到用户都是较为的年轻一代，随着 5G 时代的到来，这一年轻化的趋势会愈发明显。

(2) 经济发达地区观众占比较高：直播行业最早就是在经济发达地带发展起来的，因为在这些城市的经济条件较好，互联网服务水平也更高，在这些地区出现主播的频率也较高，大批量关注主播的用户也多在经济发达地区，这与经济发达地区的人群消费能力较强、意识较为前卫也有关系。

(3) 喜欢方便快捷，追求高效：当今社会，更多的人们在追求方便快捷的生活方式，追求极简主义与更高的效率，在如今各类直播平台的爆发式增长之下，很多喜爱直播的用户亟待需要一款能够集合百家之所长的直播聚合平台应用来提高观看直播时的效率。

(4) 女性占比逐年增高：传统的娱乐直播一般都以游戏直播为主，而直播软件用户也多为男性，但是随着腾讯游戏王者荣耀的兴起，越来越多的女性开始步入游戏行业，尤其是当代女大学生对于游戏直播的热爱趋势逐年增加，这也反映了直播行业用户受众之广的现象。

3.2 功能性需求分析

通过对用户的调研与访问，最终确定本直播聚合平台分为直播后台管理端和直播平台门户端。直播后台管理端主要包括直播聚合功能，即能够将不同直播平台的直播内容聚合在一个平台播放；直播数据管理功能，即由管理员对后台进行操作，有管理员对直播平台内的所有直播间进行管理，包括维护直播间信息、封禁直播间、评论管理等功能；同时还有账号信息管理，可以由管理员对用户账号资料进行管理，对用户收藏和投诉内容进行管理^[10]；直播平台门户端主要面向广大观看直播的用户，能够实现主界面功能，即可以显示系统主界面并且查看直播详情页；直播播放功能即能实现正常的播放以及其他拓展播放等功能；直播搜索功能即能实现依据不同搜索条件去搜索直播内容；个人功能即可以实现用户的个人操作，例如对喜欢的直播评论、收藏、分享。

3.2.1 直播平台门户端

直播平台门户端主要包括加载主界面功能、直播播放功能、直播搜索功能和个人功能，直播平台门户端的用例图如图 3-1 所示。

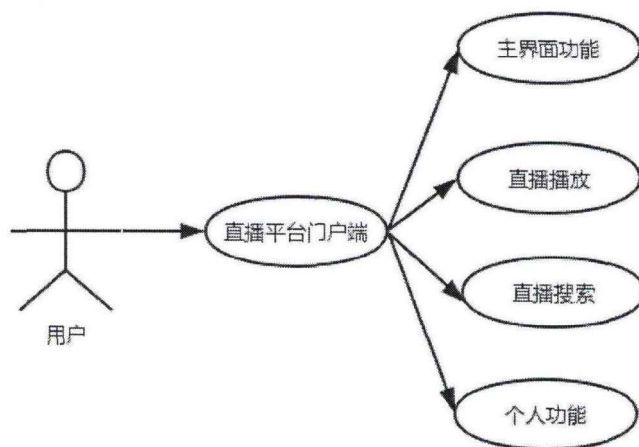


图 3-1 直播平台门户用例图

Figure 3-1 Use Case Diagram of Live Platform Portal

(1) 主界面功能

主界面功能包含加载主界面功能、显示直播详情页、显示直播列表功能，主界面功能用例图如图 3-2 所示。

1) 加载主界面

主界面是系统与用户直接进行交互的界面，它的加载速度、简洁程度、流畅度等因素，都影响了用户对一个应用的体验，从而也会影响用户对于系统的看法与好感度。

2) 显示直播详情页

用户发现喜欢的直播节目后点击进入直播间，此时首先展现的即直播详情页，直播的详情界面除了直播视频内容外还需要给用户展示直播间名称、主播名称、是否有关关注该直播间、简介等基本信息。同时在直播内容的下方还要有多种类型的推荐直播内容，可以是相关主题，也可以是同一平台的直播内容，这一部分的内容将会以列表形式于下方展示。

3) 显示直播列表

当用户点开应用时会启动 launcher，判断应用是否有新的版本^[11]，如果有新的版本则提示用户进行更新，若无新的版本则进入首页时，默认进入推荐的首页，需要能看到不同机构下所有的直播间，并能够在栏目里看到不同的类别，在首页可以显示当前正在直播的与未直播的直播间列表，在“栏目”页可以依照直播内容进行分类查看，在每一个类别里显示直播间列表。

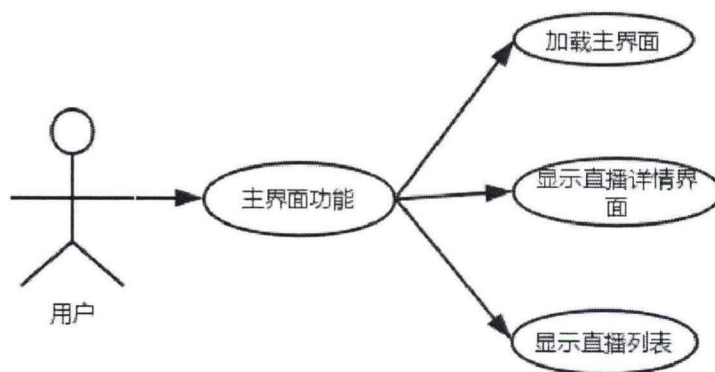


图 3-2 主界面功能用例图

Figure 3-2 Use Case Diagram of Main Interface Function

(2) 直播播放

直播播放需要能够实现调节清晰度、录屏、小窗口播放、后台播放等功能，直播播放用例图如图 3-3 所示。

1) 调节清晰度

作为面向广大直播观众的移动端应用，本产品基础功能就是实现直播播放功能，当用户进入直播详情页后直播画面开始加载，此时直播画面为最小化状态，用户可以点击右下角放大按钮进行放大至全屏播放。并且在画面的右下角可以实

现清晰度的调节。实现整个直播的正常播放功能。

2) 录屏功能

时下短视频及直播类应用极其流行，抖音快手等软件的出现使得很多普通用户都可以上传高质量直播成为坐拥上万粉丝的网络红人，所以提供高质量的直播录制功能已经成为现如今直播平台必不可少的一个功能，拥有这个功能可以使用户在观看到精彩的直播时一键录制，从而积累上传视频的素材。

具体需求为播放器右侧，增加一个主播时刻的录制按钮，锁屏按钮移到左边，弹幕按钮在右上角，录制按钮在右下角，用户在第一次进入播放器时给用户录屏引导。在用户按住录制按钮录制开始^[12]，松手结束录制。录制时间要求有上限时间 20 秒与下限时间 5 秒。整体录制时间用户可自定义，长度必须大于或等于下限时间且小于或等于上限时间。

录制成功后，自动保存录制的 mp4 到主播时刻，画面从全屏“动画化”移动到左下角的“缩略图区域”，下方弹出分享模块，直播画面与声音继续进行，左下角缩略图为静图（取自录制第一张图片，用直播间缩略图兜底），添加播放按钮。点击左下角的预览图区域放大全屏播放录制的直播，此时有“收起”按钮，点击“收起”返回之前的状态并且继续直播。

中下部为分享区：主播头条，朋友圈，微信好友，qq 好友，上方文案为“录制完成，将精彩分享到”。右下角为“稍后再说”，配以文字：（录制视频将暂存至主播时刻）。点击“稍后再说”关闭分享区域，如是在锁屏状态下录制成功的，则弹出的分享模块可点击。

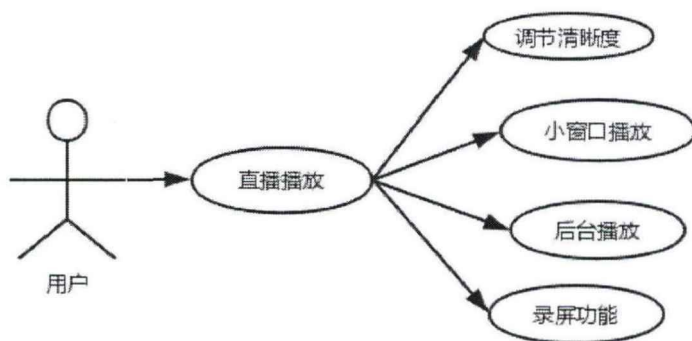


图 3-3 直播播放用例图

Figure 3-3 Use Case Diagram of Live Broadcast

3) 小窗口播放

现在主流的直播应用像斗鱼、虎牙、熊猫 TV 等都已经现小窗口播放功能，用户可以在设置里自行选择是否打开此功能，如果开启此功能，则在用户从当前的

直播间返回首页时，当前直播内容将缩小到右下角，这样用户即可一边浏览收益也直播列表一边观看当前直播，在寻找新的直播间的过程中也不耽误精彩内容。

4) 后台播放

根据调查研究表明，很多用户有听直播的习惯，较多数用户会认为主播的声音更具有吸引力，也因此喜欢观看该主播的节目；另外一些用户则是遇到喜欢的直播以后，在观看一段时间发现眼睛疲倦或者因为需要做其他事情而不得不将直播程序挂起在后台，此时需要后台将直播的声音继续播放，用户既可以凭借主播的解说来享受直播带来的快乐。综上所述，决定增加直播后台播放功能。

(3) 直播搜索

目前主流的直播软件数量较多，每一个直播软件平台又包含成千上万的主播，作者所设计的直播聚合平台将聚合数以万计的直播信息，那么要在这数以万计的信息中找到自己想要看的直播，搜索功能则是必不可少的，经过对用户观看直播行为习惯的调研，本聚合平台需要实现按照主播名字、直播平台名称、直播间 ID、推荐热门词进行搜索的功能，直播搜索用例图如图 3-4 所示。

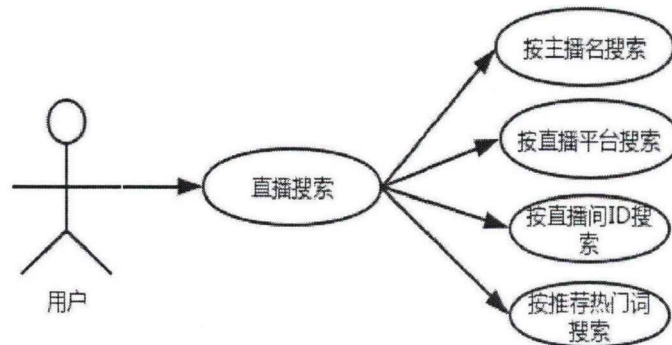


图 3-4 直播搜索用例图

Figure 3-4 Use Case Diagram of Live Search

1) 按照主播名字搜索。很多直播平台都有自己的人气主播，多数用户能够记住这些较为流行的主播名字。且被主播个人吸引来观看直播的用户极多，所以依据主播名及简称来搜索直播的这一行为频率较高。

2) 依据推荐热门词搜索。本应用在搜索框的正下方会展示近期高频搜索热词，用户可以点击这些高频热词进行一键搜索，热词均为近期一周之内用户搜索的次数最多的词，能从一定程度上满足大部分用户的需求。

3) 依据直播间 ID 搜索。因为各大直播平台都有自己的直播间，每个直播间都会有直播间 ID，难免会有 ID 相同但是分属于不同直播间平台的直播间，此时搜索直播 ID 即会出现多个结果，属于模糊搜索。

4) 依据直播平台名称进行搜索。在用户的行为习惯调查中调研团队发现有很

多用户会依据主播所在的平台去查找这一主播的直播间，先找到所属熟悉的平台然后再去找这一平台里喜欢的主播，所以设计了依据直播平台名称进行模糊搜索的这一功能。

(4) 个人功能

个人功能应该实现用户登录、用户评论、收藏转发，能够使得用户和系统有一个较为良好的交互体验，个人功能用例图如图 3-5 所示。

1) 用户登录

当用户来到首页时默认是以游客身份在进行浏览，但是若想使用后续的收藏、关注、分享等功能则必须选择登录状态，本系统登录状态分为两种，一种是第三方登录，支持微信微博与 QQ；另一种则是注册手机号后手机号登录，登录之后可以完善个人信息，更加方便后台对于用户的管理。

2) 评论功能

经过调查，当用户对某一个直播内容极其认可时或者观看到感兴趣的直播内容则会有很大概率想要发表自己的想法，由此看来增加评论功能非常有必要。于是本应用增加了评论功能，而且用户之间可以相互回复评论的设定也使得用户对于此应用的依赖程度有所提高，有利于提高用户黏度。

3) 收藏转发

本应用增加了收藏转发功能，在个人页面设置了“我的收藏”按钮，用户可以收藏喜欢的直播间，同时在直播详情页设置了转发按钮，可以一键分享给好友。

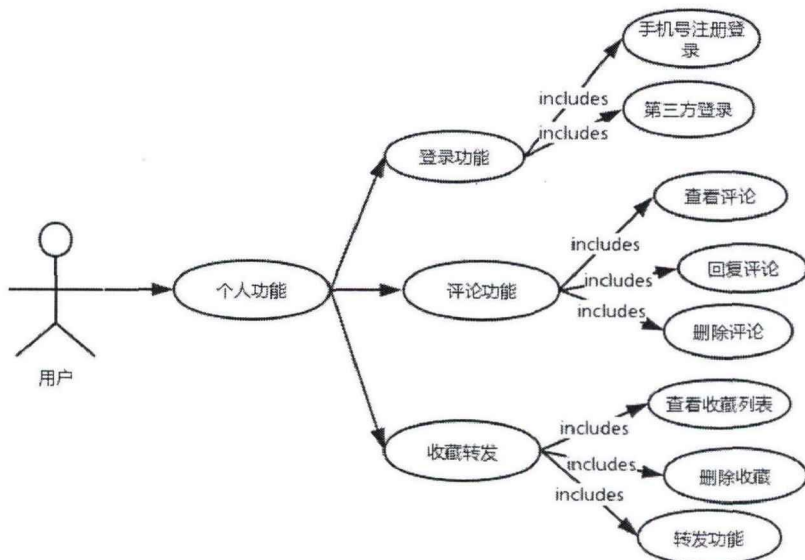


图 3-5 个人功能用例图

Figure 3-5 Use Case Diagram of Personal Function

3.2.2 直播后台管理端

直播后台管理类主要包含直播聚合：实现将不同直播平台聚合在一起；直播数据管理：管理员可以对直播内容进行管理。例如：违规内容管理、评论管理等；账号信息管理：管理员可以对用户账号的常规信息进行管理，对投诉内容进行管理，以及实现直播聚合功能，用例图如图 3-6 所示。

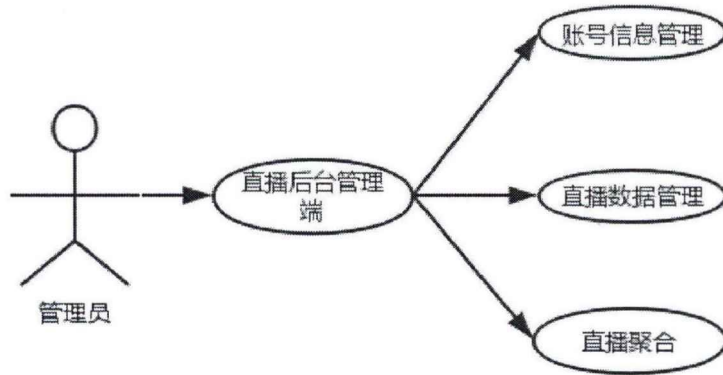


图 3-6 直播后台管理端用例图

Figure 3-6 Use Case Diagram of Live Background Management

(1) 直播聚合功能

直播聚合功能主要是包含聚合直播、聚合平台管理、性能优化功能，用例图如图 3-7 所示。

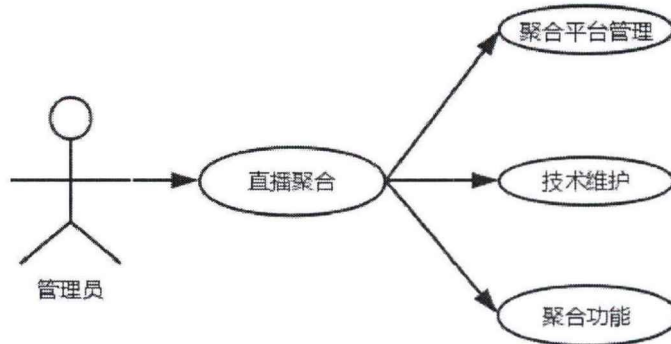


图 3-7 直播聚合用例图

Figure 3-7 Use Case Diagram of Live Aggregation

1) 聚合直播

随着直播平台的不断增多，用户需要下载多个应用来观看喜爱的直播，为了节省用户的存储空间，也节省用户挑选直播应用的时间，提高观看直播的效率，

本直播聚合平台将实现把多家直播平台的直播内容聚合在一起的功能。

2) 聚合平台管理

聚合平台数量增多后需要对它们有一个统一的管理，使得之后再增加直播平台或者封禁直播平台以及批量操作所有的直播平台更加方便。

3) 技术维护

直播聚合功能主要是依据现有技术去实现，在播放时间上还可能存在一定延迟，需要不断去优化，不断去提高技术水平。

(2) 账号信息管理

账号信息管理将要实现权限认证功能、用户资料管理功能以及投诉管理功能，用例图如图 3-8 所示。

1) 用户资料管理

当管理员发现用户信息有误或者需要修改时可以通过后台来对用户的一些信息进行修改。

2) 投诉信息管理

作者在直播详情页设置了投诉按钮，用户在观看直播过程中如果发现有违规直播内容则可以通过点击投诉按钮进行投诉，管理员可以及时查看投诉信息并且核验相应直播内容，如果确实涉及敏感内容则将对直播间进行封禁处理。

3) 权限认证功能

权限认证在保证特定用户能使用特定功能和确保网站安全方面起着至关重要的作用。本系统后台管理端实现权限认证功能，即实现对于用户和管理员不同的权限赋予功能，由于本直播平台管理端只有特定身份的用户才能够操作，对于系统管理员而言当系统管理员登录系统时需要对其身份进行认证，当身份认证通过后便能对管理端中的直播信息等内容进行管理。

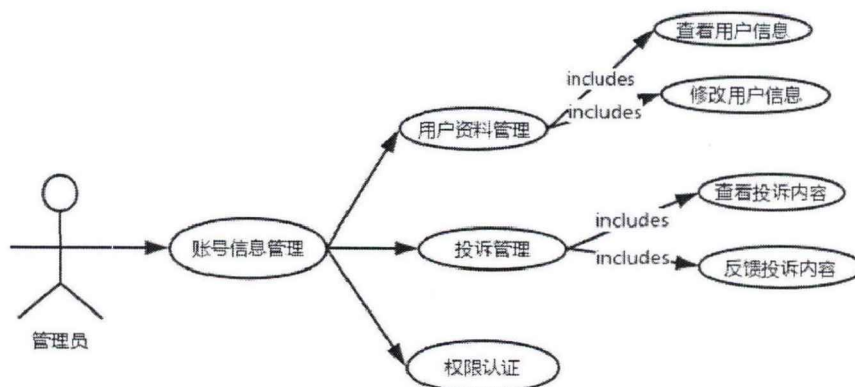


图 3-8 账号信息管理用例图

Figure 3-8 Use Case Diagram of Account Information Management

(3) 直播数据管理

直播数据管理将实现管理员对于直播间违规内容的管理、对于录屏文件的管理以及对于用户评论的管理，用例图如图 3-9 所示。

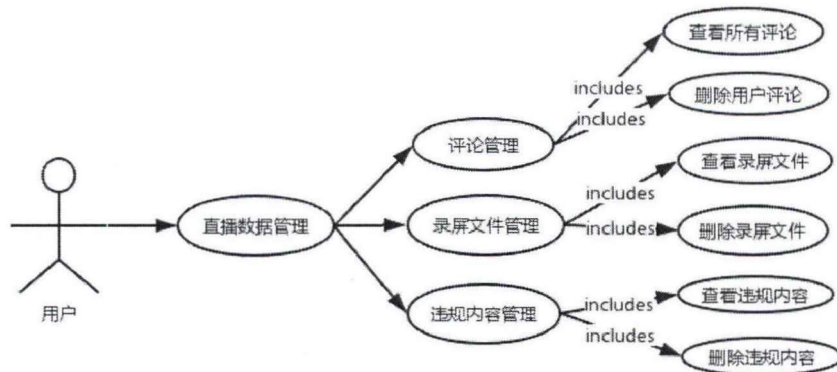


图 3-9 后台数据管理用例图

Figure 3-9 Use Case Diagram of Background Data Management

1) 违规内容管理。

近年来，直播领域因为其使用者众多，盈利模式成熟，吸引了大量观众，有很多商家和用户不断地关注这一行业，直播平台的发展势头良好，呈现高速增长态势，从最初的娱乐直播平台斗鱼到现在的明星常用的一直播应用等等^[13]，直播平台数量呈现急速增长态势，同行业的从业者增多那么商业竞争也就会增多，这时很多竞争者便开始采取一些不正当竞争手段来提升自身平台流量，从而提升竞争力，这样的行为已经是违反道德和法律的不良行为，再加上直播行业面向的受众很多都是未成年人，往往可能心智还不成熟，这些用户因为喜欢某个主播或者某款游戏来到直播间，遇到这样的不良直播信息没有抵抗诱惑的能力，很可能会被影响从而走上歧途，对未成年人造成十分不好的影响。更会形成非常不良的社会风气。目前这一现象已经被国家和有关监管部门也注意到了，并且已经出台相关政策加以规范。除了国家的监督以外，作为直播平台的设计者，更要通过加强自身监管技术来营造这一行业的优良风气，要不断提高相关技术来对这种不良现象进行遏制，直播违规内容管理十分有必要。

本直播聚合平台将实现违规内容管理功能：主要用到的技术手段是通过服务器对直播间直播画面的定时截图^[14]，然后调用图像处理算法鉴定这些直播画面，将疑似敏感画面保存至管理端后台，然后由管理员对这些违规内容进行管理，即在系统的管理端会依托于服务器的鉴定违规内容功能，进行直播画面的定时鉴定，在看到违规画面后能够及时定位直播间并封禁直播间，为了肃清整个直播间环境，

培养良好的风气，尽全力做到发现一个不良内容处理一个直播间。

2) 录屏文件管理

本直播聚合平台在直播平台门户端功能中已经实现录屏功能，那么对应的后台管理系统中将增加录屏文件管理功能，用户和管理员都可以实现如下操作：查看录制文件大小；点击播放这些录制文件；当不再需要这些存储文件时，可以删除这些录制文件。

3) 评论管理

在评论管理中，管理员和用户都可以看到直播间的评论信息，可以实现如下操作：查看评论；删除评论。并且可以通过对评论的关键词检索然后发现自己感兴趣的评论。

3.3 非功能性需求分析

系统的非功能性需求在功能性需求的基础上进一步保证了直播聚合平台系统在上线之后能给用户带来更好更优质的直播服务，具体来说，首先需要保证直播时的稳定性，能够尽量与原直播平台同步，其次需要保证管理端操作的安全性，不泄露用户信息，因此本系统主要从时效性、可靠性、安全性、兼容性、扩展性五个方面进行分析。

(1) 时效性

因为此直播聚合平台所聚合的直播内容是通过地址反解析抓取而得，所以需要提高即时传输效率，保证本平台的直播视频播放和原平台播放速度相差不下 3s。

(2) 兼容性

兼容性指的是直播聚合平台客户端可以在不同的安卓手机型号上实现同样流畅的播放，实现正常功能，同时也可以兼容不同直播平台的不同视频格式，后台服务端也可以在不同的系统上运行。

(3) 安全性

本直播聚合平台的使用者主要是观看直播的用户群体，当用户登录使用本直播平台时会保存个人信息存储在数据库中，保证直播平台使用者信息的安全对直播平台来说是十分重要的。此外为保证系统的可靠性，直播平台服务器的安全是另一个需要重视的安全问题。

(4) 易用性

为使得让本直播平台的使用者能够很快的上手，平台设计初期在项目的界面设计上花费了很大的精力，确保本项目的界面让使用者感到友好，此外在用户初次登录本系统时会有新手指引，初次使用者可以快速上手，保证良好的体验。

(5) 扩展性

直播聚合平台应该具有良好的可拓展性，可以在原有的直播平台的基础上拓展新的直播平台

3.4 本章小结

本章对直播聚合平台的需求进行了分析，首先对直播平台的特点和用户的特点进行了分析，确保在需求分析时能够面面俱到。接下来在具体的功能性需求分析中，借助用例图对直播平台中所要满足的功能需求进行了详细的阐述，将整个系统的功能划分为直播平台门户端与直播后台管理端，在直播门户端中将要实现加载主界面、直播详情界面、直播播放、直播搜索、用户登录、个人功能等功能；在直播后台管理端中将实现直播聚合、直播数据管理、评论管理、违规内容管理等功能。在非功能性需求分析中，主要通过时效性、兼容性、安全性、易用性，拓展性五个方面对直播聚合平台的非功能性需求进行了分析。

4 直播聚合平台概要设计

依据前一章的需求分析，本章将对直播聚合平台进行概要设计。在概要设计阶段首先需要确定系统的设计目标，然后确定本直播聚合平台的架构，包括逻辑架构以及技术实现架构，接着划分直播聚合平台的功能模块，同时对系统的各个模块的功能和数据库做出合理的设计。本章认真参照软件工程设计原则，在整个设计过程中保持一个准则：即做到模块间的高内聚低耦合^[15]。为之后系统的详细设计与实现提供保障。

4.1 系统设计目标

直播聚合平台的设计目标是能够开发出一款直播平台最全、播放功能完善、安全易用、用户体验极佳的直播平台客户端，同时也配以一套便于管理、功能强大的后台管理端系统，通过第三章需求分析可知整个直播聚合平台将具备以下特点：

(1) 聚合更多直播平台

目前人气火爆、较为主流的几款直播应用有虎牙直播、斗鱼直播、熊猫直播、龙珠直播等等，各个直播平台都有自己的当家人气主播，那么本人直播聚合平台设计的第一目标就是将各大直播平台的直播内容全部汇聚到一起，尽量为用户节省存储空间以及提高用户观看直播的便捷性。

(2) 全面完善直播功能

即使本系统汇聚了较多的直播平台，但是也不能影响本系统的最主要的播放功能，相反本系统要为用户提供更为良好的观看直播体验，于是在系统的需求阶段就做了大量的问卷调查以及走访工作，最终确定系统的播放功能除了要实现正常的暂停、全屏、调节清晰度等功能外本系统还要实现小窗口播放、后台播放以及录屏功能。

(3) 使用户操作简单便捷

直播用户对于直播类应用的操作要求越简单越好，为此本人设计了简洁的页面，同时本系统也增加了直播搜索功能，用户可以按照关键字模糊查询也可以按照平台简称和推荐热门词精确查询；同时系统也增加了直播推荐功能，能够依据用户的喜好与行为习惯为用户推荐相关直播节目，极大地提高了用户观看直播地效率。

4.2 系统架构设计

成功的软件设计绝不仅仅能够满足用户提出的功能性需求，还需满足常规的非功能性需求，而设计出符合系统需求的架构则是保证系统顺利运行的重中之重，从某种意义上讲它比功能性需求更加重要，良好的架构设计能给系统的性能带来很大的提高。

4.2.1 系统逻辑架构

直播聚合平台系统逻辑架构按照分层理念将系统分为上层表示层、中层业务逻辑层和下层数据管理层^[21]三个部分，是典型的 MVC 架构。表示层位于最上层，包含首页导航栏、一些功能的操作界面、个人功能的页面等与用户进行交互的界面。业务逻辑层在中间发挥着承上启下的重要作用，依据制定的需求它设计出相关的业务流程，包含系统的核心业务直播平台、直播平台内容信息、用户信息等的管理业务。数据处理层在底层，在这一层则完成对于数据库的访问工作，包括临时数据、缓存数据的持久化和缓存的刷新，同时还包括对象和数据表之间的映射。整个系统的逻辑架构图如图 4-1 所示。

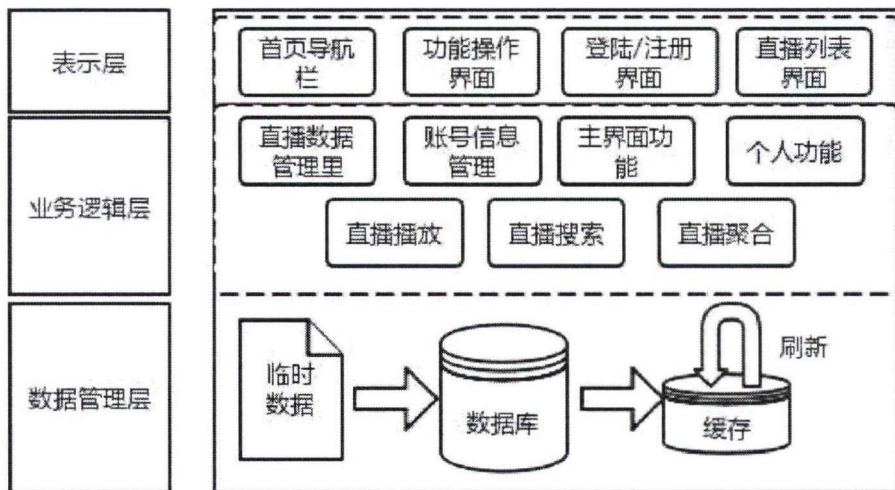


图 4-1 系统逻辑架构图

Figure 4-1 Architecture Diagram of System Logical

4.2.2 系统客户端技术实现架构

直播聚合平台的可维护性较强，因为客户端主要采用 MVP 设计模式。修改视图时可以对模型产生影响，可以将模型和视图分开，且分层思想能够极大地提

高效率，视图层的功能是接收用户操作、最直接的进行交互^[17]，此方式的目的是为输出数据，同时也允许用户进行操作；模型层的作用是处理数据请求、通过单一模型提供多个视图数据来减少代码的重复性；表示层接收用户的输入，并为实现用户的需求而调用模型和视图，通过模型层的回调接口拿到数据，调用视图层对应的接口将数据传递给 View 层。

本系统应用 MVP 思想的设计模式，解耦业务逻辑与 UI 逻辑。本系统的客户端技术架构图如图 4-2 所示。

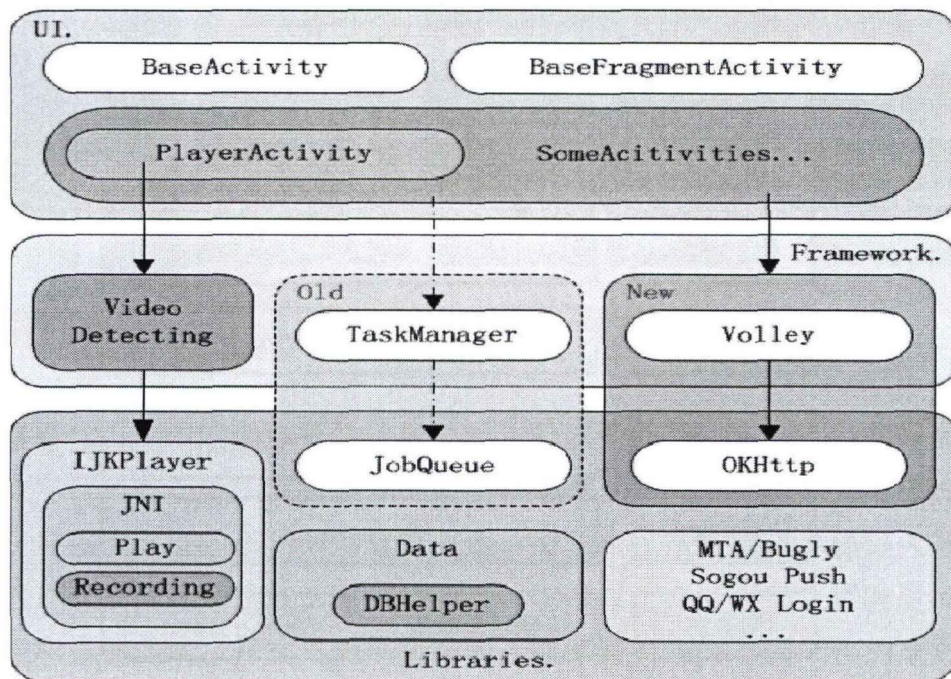


图 4-2 系统客户端技术架构图

Figure 4-2 Architecture Diagram of System Client Technical

上层就是用户看到的各个界面，有两个父类，下层是具体的网络请求，数据库操作，播放器的实现，中间层是承接上下层的一个封装层或者管理层^[18]。

UI层根据需求可选择继承 BaseActivity 或 BaseFragmentActivity, PlayerActivity 是其中较为重要的播放界面，采用 Ijkplayer 框架播放，对 Ijkplayer 进行的二次开发使其支持录屏需求，播放前需进行资源探测拿到真实的直播源地址，再把结构交给底层的 Ijkplayer 去播放。

网络请求框架基于 Volley 封装的框架, Volley 底层采用 OkHttp 作为底层通信, 后期改用新的网络请求技术 RxJava 加 Retrofit 来实现。

数据的存储目前采用 MySQL 以及自定义注解结合反射的方式生 SQL 进行的, 后续可优化, 数据存储部分包含了各个业务功能的抽象对象, 主要作用是对数据进行存储和处理。

4.2.3 系统管理端技术实现架构

直播聚合平台管理端即后台服务端的主要功能是对直播平台日常数据进行管理，使用较为典型和成熟的 MVC 结构，数据库主要使用 MySQL，同时用 Redis 辅助来减轻系统访问数据库的压力。为了使系统能够满足同时在移动端和 PC 端进行浏览的需求，系统采用了 Vue 框架和 SUI Mobile 所提供的 UI 库，它们都具有轻量、高效的特点，保护系统的业务层则是通过 Spring Security 来实现，控制层和服务层采用了 Spring Boot 来简化项目中的各种配置^[19]，数据库持久层框架使用了 Mybatis，系统管理端技术实现架构图如图 4-3 所示。

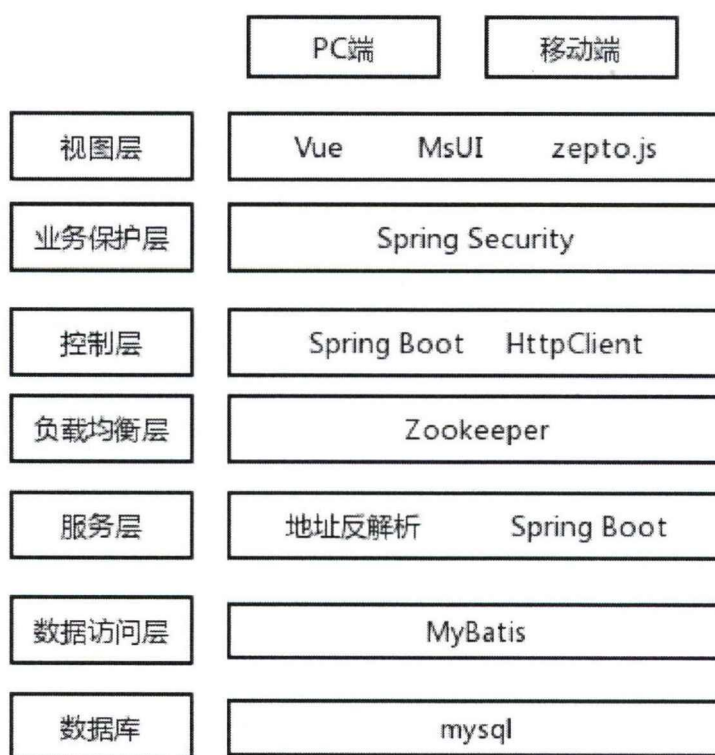


图 4-3 系统管理端技术架构图

Figure 4-3 Architecture Diagram of System Management Technical

4.3 系统总体功能结构

通过对直播聚合平台系统进行需求分析，共获得 7 个核心业务用例，分别为直播平台门户端的直播搜索模块、直播播放模块、个人功能模块、主界面功能模块以及直播后台管理端的直播聚合模块、直播数据管理模块、账号信息管理模块，通过系统总体功能结构模块图可以更加直观地展示系统的功能模块，如图 4-4 所示。

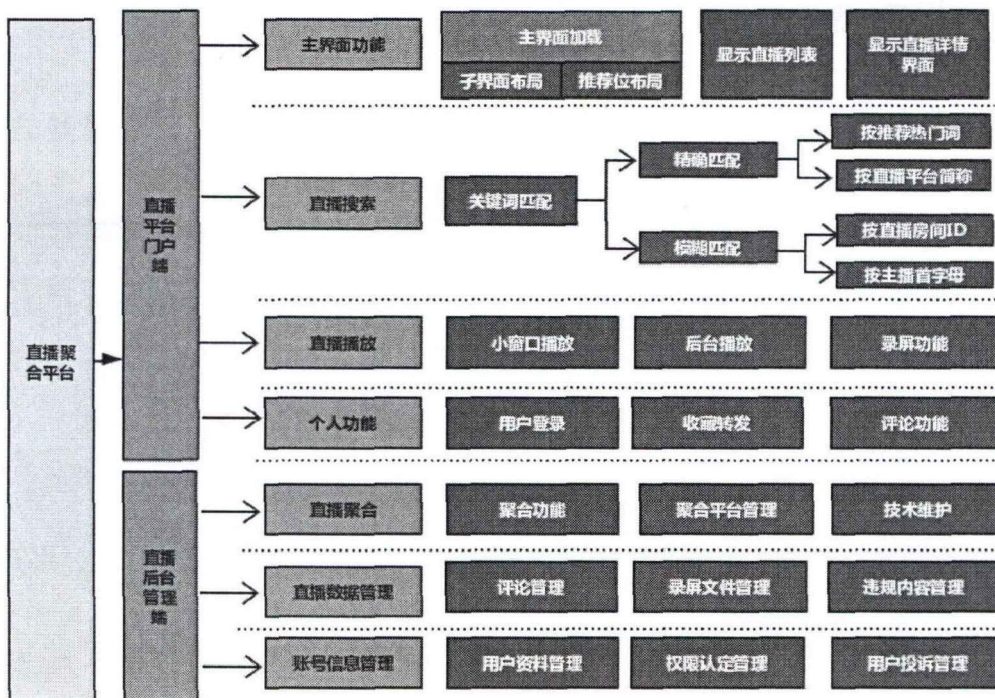


图 4-4 系统总体功能模块结构图

Figure 4-4 Structure Diagram of System Function Module

直播平台门户端的功能模块主要针对观看直播的用户。其中，主界面加载是和用户交互最多的功能，在用户打开应用程序时，加载主界面的同时还将加载子界面。在带有推荐位的子界面加载完成后，用户可以从直播主界面进入直播详情界面，用户可以浏览系统默认的直播列表，也可以选择某一类直播查看其相应的直播列表和直播详情。且在直播展示列表上能够看到各种不同平台的直播节目也从前端体现了本系统的聚合功能。关键词匹配技术为直播搜索这一模块提供技术支持，可以进行模糊搜索与精确搜索，用户在可以按照系统推荐的近期热词或者直播平台来进行精确搜索；也可以选择依据直播间 id 或者主播名字相关词去进行模糊搜索，因为会出现很多相近的 ID。直播播放作为本聚合平台的模块，除了正常播放能够对分辨率的选择外，还增加了小窗口播放、后台播放、录屏功能，同时在播放详情页的正下方有“猜你喜欢”的相关推荐列表。在直播平台门户类中将非主要功能的用户登录、收藏转发和评论合并为个人功能模块^[22]。

直播后台管理端的功能模块主要面向系统管理员，可以对前端实现的功能所返回的信息进行管理，系统存储部分包含了各个业务功能的抽象对象。账号信息管理主要是对用户的个人信息进行管理、对认定权限进行管理、对投诉内容进行管理；直播数据管理则包含评论管理、违规内容管理以及录屏文件管理。同时本系统的核心功能即直播聚合模块也在后端进行实现，管理员可以对本系统聚合的平台进行管理，可以在聚合技术更新时对直播平台进行维护。

4.4 数据存储设计

通过需求分析和模块设计作者已经对整个系统的结构有了比较清楚的认识，这对设计数据存储是十分有帮助的。合理的数据存储也是避免数据冗余的十分高效的方式，直播聚合平台的数据库架设在远程服务器上，客户端系统数据进行访问的方式是通过服务器接口来进行的。下面通过数据库设计、本地数据缓存、服务器接口三个方面对本系统的数据存储进行设计。

4.4.1 数据库设计

直播聚合平台的实体有直播间、直播平台、直播收藏列表、用户、用户角色、管理员、主播、评论记录、录屏文件。一个用户对应一个用户角色，一个用户角色可以有多个用户，因此是多对一的关系。一个管理员可以管理多个直播平台，而一个直播平台可以被多个管理员管理，因此是多对多的关系。同理，可以理解其他实体之间的关系，本系统整体 E-R 图如图 4-5 所示。

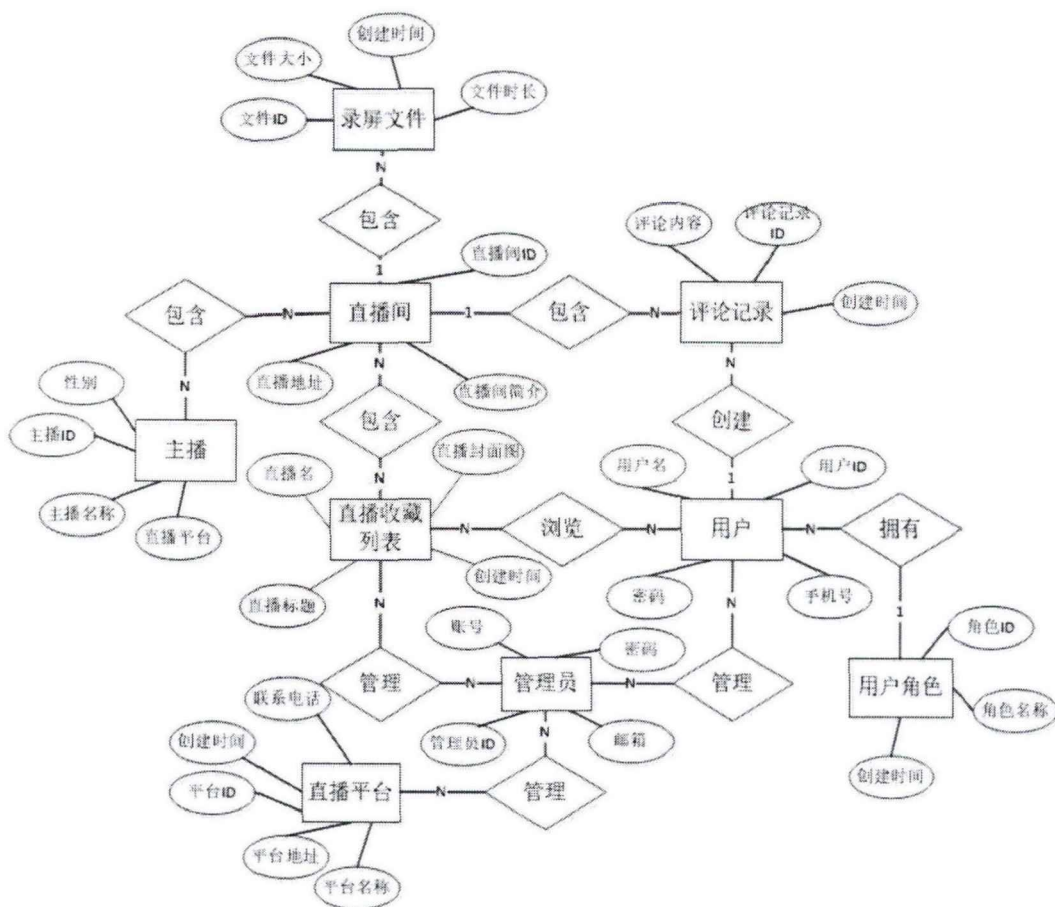


图 4-5 系统 E-R 图

Figure 4-5 E-R Diagram of System

根据系统 E-R 图，设计出直播间数据库表 live_info，对应实体关系图中的直播间实体，其详细表结构如表 4-1 所示。

表 4-1 直播间数据库表
Table 4-1 The Table of LiveRoom

表名: live_info				
序号	字段名	类型	属性	描述
1	Liveroom_id	varchar(32)	主键、自增	直播间 ID
2	host_id	varchar(255)	非空	主播 ID
3	host_name	varchar(512)	非空	主播名
4	live_title	varchar(255)	非空	直播间名
5	Liveroom_url	varchar(32)	主键、非空	直播间地址
6	Liveroom_desc	varchar(32)	非空	直播间简介
7	total	bigint(20)	非空	观看人数
8	country	varchar(255)	允许为空	国家
9	area	int(11)	允许为空	地区
10	Liveroom_link	varchar(1600)	非空	原地址链接
11	state	tinyint(1)	非空	直播状态
12	create_time	timestamp	非空	创建时间
13	update_time	timestamp	非空	更新时间

直播收藏列表数据库表 favorite，对应实体关系图中的直播收藏列表实体，其详细表结构如表 4-2 所示。

表 4-2 直播收藏列表数据库表
Table 4-2 The Table of Favorite Live List

表名: favorite				
序号	字段名	类型	属性	描述
1	Liveroom_id	bigint(20)	主键、自增	直播间 ID
2	user_id	varchar(255)	非空	用户 ID
3	live_title	varchar(255)	允许为空	直播名
4	create_time	timestamp	非空	创建时间
5	update_time	timestamp	非空	更新时间
6	live_image	blob	非空	直播封面图

根据系统 E-R 图，设计出直播平台表 `org_info`，对应实体关系图中的歌手信息实体，其数据库表如表 4-3 所示。

表 4-3 直播平台数据库表
Table 4-3 The Table of Live Platform

表名: <code>org_info</code>				
序号	字段名	类型	属性	描述
1	<code>org_id</code>	<code>bigint(20)</code>	主键、空	直播平台 ID
2	<code>org_name</code>	<code>varchar(255)</code>	非空	直播平台名
3	<code>country</code>	<code>varchar(255)</code>	允许为空	国家
4	<code>language</code>	<code>varchar(255)</code>	允许为空	语言
4	<code>host_total</code>	<code>varchar(255)</code>	允许为空	主播总数
5	<code>org_intro</code>	<code>text</code>	允许为空	平台介绍
6	<code>c_name</code>	<code>varchar(255)</code>	非空	负责人名字
7	<code>c_phone</code>	<code>varchar(255)</code>	非空	联系电话
8	<code>c_email</code>	<code>varchar(255)</code>	允许为空	负责人邮箱
9	<code>org_addres</code>	<code>varchar(255)</code>	允许为空	平台地址
11	<code>create_time</code>	<code>timestamp</code>	非空	创建时间
12	<code>update_time</code>	<code>timestamp</code>	非空	更新时间

用户角色表 `user_role`，对应实体关系图中的用户角色实体，其详细表结构如表 4-4 所示。

表 4-4 用户角色数据库表
Table 4-4 The Table of User Role

表名: <code>user_role</code>				
序号	字段名	类型	属性	描述
1	<code>role_id</code>	<code>bigint(20)</code>	主键、自增	用户 ID
2	<code>role</code>	<code>varchar(32)</code>	非空	角色名称
3	<code>create_time</code>	<code>timestamp</code>	非空	创建时间
4	<code>update_time</code>	<code>timestamp</code>	非空	更新时间

根据系统 E-R 图，用户表 `user_info`，对应实体关系图中的用户实体，其数据库表如表 4-5 所示。

表 4-5 用户数据库表
Table 4-5 The Table of User

表名: user_info				
序号	字段名	类型	属性	描述
1	user_id	bigint(20)	主键、自增	用户编号
2	user_name	varchar(255)	唯一	用户名
3	password	varchar(255)	非空	密码
4	user_account	varchar(255)	非空	用户账号
4	user_phone	varchar(255)	非空	用户手机号
5	user_right	varchar(255)	非空	用户权限
6	user_pic	varchar(255)	非空	用户头像
7	state	tinyint(1)	非空	用户状态
8	create_time	timestamp	非空	创建时间
9	update_time	timestamp	非空	更新时间

根据系统 E-R 图，设计主播表 host，对应实体关系图中的主播实体，其详细表结构如表 4-6 所示。

表 4-6 主播数据库表
Table 4-6 The Table of LiveHost

表名: host				
序号	字段名	类型	属性	描述
1	host_id	varchar(255)	主键、非空	主播 ID
2	host_name	varchar(512)	非空	主播名
3	live_title	varchar(255)	非空	直播间名
4	host_email	varchar(255)	非空	主播邮箱
5	host_QQ	varchar(255)	非空	主播 QQ
5	host_sex	int(11)	非空	主播性别
6	host_pic	varchar(255)	允许为空	主播头像
7	org_id	bigint(20)	非空	直播平台 ID
8	create_time	timestamp	非空	添加时间
9	update_time	timestamp	非空	更新时间

管理员表 c_info，对应 E-R 图中的管理员实体，其详细表结构如表 4-7 所示

表 4-7 管理员数据库表
Table 4-7 The Table of Administrator

表名: c_info				
序号	字段名	类型	属性	描述
1	c_id	bigint(20)	主键、自增	管理员编号
2	c_name	varchar(255)	唯一	昵称
3	password	varchar(255)	非空	密码
4	c_right	varchar(255)	非空	权限
5	c_email	varchar(255)	非空	管理员邮箱
6	c_phone	varchar(255)	非空	管理员手机
7	c_account	varchar(255)	非空	管理员账号
8	c_state	tinyint(1)	非空	管理员状态
9	create_time	timestamp	非空	创建时间
10	update_time	timestamp	非空	更新时间

录制文件表 record_file, 对应实体关系图中的录制文件实体, 其详细表结构如表 4-8 所示。

表 4-8 录制文件数据库表
Table 4-8 The Table of Record File

表名: record_file				
序号	字段名	类型	属性	描述
1	live_record_id	bigint(20)	主键	录制文件 ID
2	Liveroom_id	varchar(32)	非空	直播间 ID
3	host_id	varchar(255)	非空	主播 ID
4	host_name	varchar(512)	非空	主播名
5	file_url	varchar(255)	非空	文件地址
6	file_size	varchar(1024)	非空	文件大小
7	duration	varchar(255)	允许为空	文件时长
8	create_time	timestamp	非空	创建时间
9	update_time	timestamp	非空	更新时间

评论记录表 comment, 对应实体关系图中的录制文件实体, 其详细表结构如表 4-9 所示。

表 4-9 评论记录数据库表
Table 4-9 The Table of Comment Record

表名: comment				
序号	字段名	类型	属性	描述
1	live_comment_id	bigint(20)	主键	评论记录 ID
2	Liveroom_id	varchar(32)	非空	直播间 ID
3	user_id	varchar(255)	非空	用户 ID
4	user_name	varchar(512)	非空	用户名
5	user_pic	varchar(255)	非空	用户头像
6	comment	varchar(1024)	非空	评论内容
8	create_time	timestamp	非空	创建时间

4.4.2 本地数据缓存

本地缓存信息有以下三种：

(1) 直播封面图信息

客户端主界面加载以及直播列表展示等功能中直播封面图这一数据用到的较为频繁，例如：直播间数量非常多，可能会有成千上万个，它们在首页分页显示，但是每一页显示的数量却是有限的，用户在一页页翻页查找自己的想看的直播节目时，直播信息将会被频繁访问，比起文字来这些所展示的信息中封面图片占很大比重，也十分影响访问速率。因此将直播封面图片数据缓存在本地可以极大的提升用户流畅度体验，直播封面图片缓存数据^[20]详细信息见表 4-10。

表 4-10 直播封面图缓存数据详细信息
Table 4-10 Live Image Cache Data Table

序号	字段名	类型	属性	描述
1	url	TEXT	主键	直播封面图地址
2	live_image	BLOB	-	图片
3	time	int	-	添加该缓存的时间

在直播封面图加载的时候，先判断本地缓存有无数据，若无本地缓存数据，则访问本地文件系统，若文件系统也没有，则访问服务器，抓取直播封面信息，然后同时存入本地文件系统和缓存。直播封面图加载到本地的流程如图 4-6 所示。

(2) 用户信息列表

用户在本系统中可以实现登录，从而进行收藏、转发、评论直播的操作，那么用户在一次登录之后就可以把信息存储在本地缓存中，这样在下一次登陆的时候用户的登陆信息可以直接显示，基于以上考虑，现制定用户信息也将有本地缓存以此来提高用户效率，用户信息列表详细信息如表 4-11 所示。

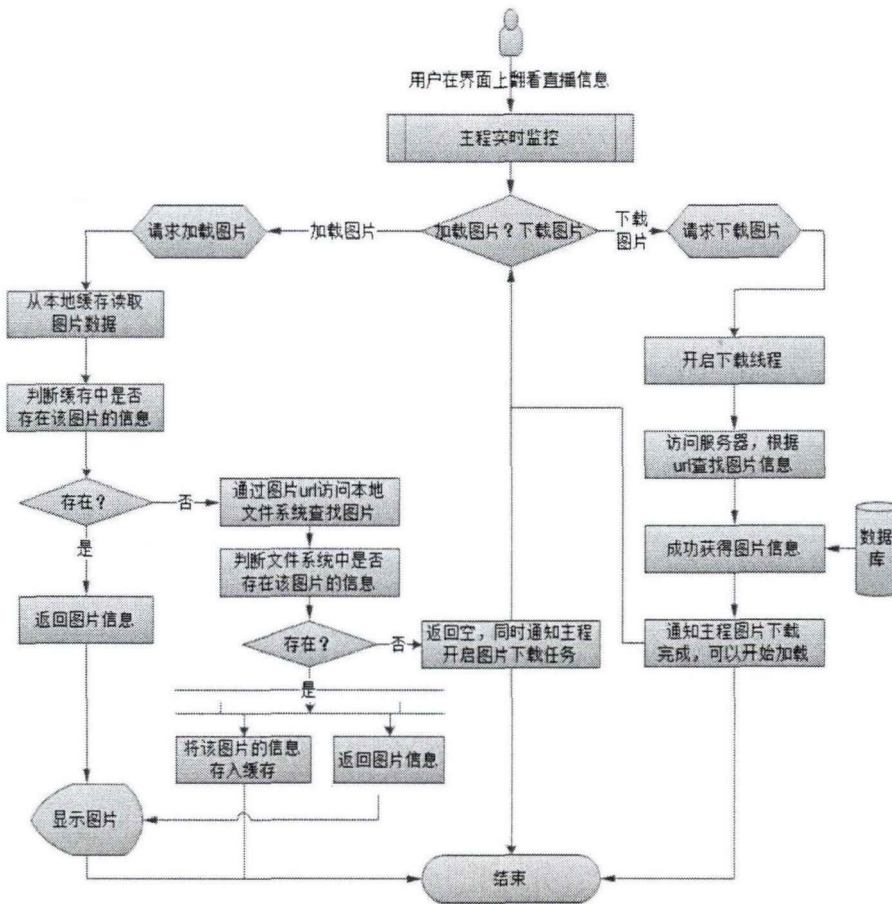


图 4-6 直播封面图加载缓存流程图

Figure 4-6 Workflow Diagram for Live Image Loading and Saving to Buffer

表 4-11 用户信息列表详细信息
Table 4-11 User information Data Table

序号	字段名	类型	属性	描述
1	user_phone	text	主键	注册手机号
2	password	varchar	-	密码
3	user_name	varchar	-	用户名
4	Userage	Int	-	用户年龄
5	image	blob	-	用户头像
3	time	Int	-	添加该缓存的时间

(3) 用户收藏列表

用户在本系统中可以登录，从而进行收藏、转发、评论直播的操作，那么涉及到的用户相关的表则有用户收藏信息表，因为用户的收藏列表也被经常访问，所以也需要设计本地缓存，用户收藏列表详细信息如表 4-12 所示。

表 4-12 用户收藏列表详细信息
Table 4-12 User Collection Data Table

序号	字段名	类型	属性	描述
1	Liveroom_id	bigint	主键、自增	直播间 ID
2	user_id	varchar	非空	用户 ID
3	live_title	varchar	允许为空	直播名
4	create_time	timestamp	非空	创建时间
5	update_time	timestamp	非空	更新时间

4.4.3 服务器接口

客户端的 GET、POST 请求会由 Servlet 接收，然后进行相应逻辑处理后将要返回的数据封装成 json 格式写出去，使用服务器接口存储数据的优势在于支持多处理器（相同类型）的水平扩展，支持多个服务器的水平扩展，支持垂直扩展，服务器可以移植到功能更强的计算机上，不涉及处理数据的重新分布问题。

如表 4-13 至表 4-21 所示是部分服务器接口的对照表结果。

表 4-13 直播信息接口对照表
Table 4-13 Live Information Interfaces Table

接口名称	API_URL_LIVEINFO
接口说明	直播信息接口
URL	http://<TuServer>/itv/liveinfo
请求方式	POST
Request 信息	id: 直播 idcommentator: 主播名 viewers: 观众数 sourcename: 直播来源 url: 直播页面 urlrawcoverimage: 直播缩略图 gamecoverimage: 游戏缩略图 title: 直播标题 name: 直播游戏名 infotype: live, 表示直播。
封装结果	LiveInfo

表 4-14 直播列表接口对照表
Table 4-14 Live List Interfaces Table

接口名称	API_URL_LIVE_LIST
接口说明	直播列表接口
URL	http://<TuServer>/itv/liveList
请求方式	POST
Request 信息	id:直播 id-abstract: 直播摘要; updatetime: 更新时间; viewtimes: 观看数; sourcename: 直播来源名称; url: 直播播放页面 url; duration: 时长; rawcoverimage: 直播缩略图; gamecoverimage: 游戏缩略图;
封装结果	List<LiveInfo>

表 4-15 直播平台接口对照表
Table 4-15 Live Platform Interfaces Table

接口名称	API_URL_PLATFORM
接口说明	直播平台接口
URL	http://<TuServer>/itv/Liveplatform
请求方式	POST
Request 信息	name : 平台中文名称; iconurl: 平台 icon 地址; androiddownloadurl: android 平台下载地址; iosdownloadurl: ios 平台下载地址; openurl: 唤起应用的打开地址头; displayname: 显示的平台名称, 例如斗鱼显示斗鱼 TV; sourcename_enum: 直播平台对应的枚举
封装结果	List<Episode>

表 4-16 通用搜索接口对照表
Table 4-16 General Search Interfaces Table

接口名称	API_URL_SEARCH_COMMON
接口说明	通用搜索接口
URL	http://<TuServer>/itv/liveSearch
请求方式	POST
Request 信息	apiKey, cpId, keyword, type, pageNo, pageSize
封装结果	List<LiveInfo>

表 4-17 热门搜索词接口对照表
Table 4-17 Hot Words Search Interfaces Table

接口名称	API_URL_SEARCH_HOT_WORDS
接口说明	热门搜索词列表
URL	http://<TuServer>/itv/livesearchHotWords
请求方式	POST
Request 信息	apiKey, cpId, keyword
封装结果	RealtimeSearchResult 类

表 4-18 历史记录查询接口对照表
Table 4-18 History Record Interfaces Table

接口名称	API_URL_LIVE_HISTORY_LIST
接口说明	历史记录查询
URL	http://<TuServer>/itv/historyList
请求方式	POST
Request 信息	apiKey, cpId, pageNo, pageSize
封装结果	List<LiveInfo>

表 4-19 用户历史记录操作接口对照表
Table 4-19 History Action Interfaces Table

接口名称	API_URL_LIVE_HISTORY_ACTION
接口说明	用户历史记录操作
URL	http://<TuServer>/itv/livehistory
请求方式	POST
Request 信息	apiKey, cpId, LiveId, tvId, vid, terminalId, videoPlayTime
封装结果	Backhistorylist 类

表 4-20 播放接口对照表
Table 4-20 Live Play Interfaces Table

接口名称	API_URL_MPLAY
接口说明	播放接口
URL	http://<TuServer>/itv/mplay
请求方式	POST

表 4-20 (续表)
Table 4-20(Cont.)

接口名称	API_URL_MPLAY
Request 信息	apiKey, cpId, albumId, tvid, vid, terminalId, videoPlayTime,
封装结果	Json 串: code 字段

表 4-21 内容鉴权服务器接口对照表
Table 4-21 Play Check Interfaces Table

接口名称	API_URL_PLAY_CHECK
接口说明	内容鉴权接口
URL	http://<TuServer>/itv/liveplayCheck
请求方式	POST
Request 信息	apiKey, cpId, albumId, tvid, vid, terminalId, videoPlayTime,
封装结果	Json 串: code 字段

4.5 本章小结

本章从四个方面：系统设计目标、系统架构、系统总体功能结构、数据存储及使用来对直播聚合平台系统进行了概要设计。系统设计目标简明扼要，目的明确。系统架构包括系统逻辑架构、系统客户端技术实现架构、系统后台服务端技术实现架构三个层面。系统功能结构则从直播平台门户端以及直播数据管理端两个维度清楚地划分了功能模块，力求高内聚低耦合。数据存储设计分为数据库设计、本地缓存以及服务器接口，通过数据库整体设计 E-R 图以每一个实体对应的数据库表阐述了本系统的数据库设计详情，经过系统清晰完整地概要设计后可以作为接下来的详细设计与实现奠定良好的基础。

5 直播聚合平台详细设计与实现

在经过了系统的需求分析以及概要设计以后，本章介绍项目在软件生命周期中极为重要的一个阶段即直播聚合平台的详细设计阶段，并且经过编码也将实现系统功能。详细设计地好坏直接决定了系统功能的优劣，因此这一章也是本文的重点内容。依据已经完成的第三章需求分析以及第四章概要设计的内容，本直播聚合平台的需求已经充分明确，也对各个模块的功能进行了合理的划分，本章将以直播聚合平台门户端与后台管理端中本人所参与的功能模块为例来清楚地阐述详细设计与实现地过程。

5.1 主界面功能模块

在本文的概要设计中，系统总体功能结构的第一个模块即为直播聚合平台门户端中的主界面模块，这是最为直接展示给用户的功能模块也是本系统的重点模块，其中的加载主界面功能以及直播详情界面功能的设计与实现如下。

5.1.1 加载主界面

直播客户端在加载主界面时首先启动 Launcher 并播放加载动画，如果当前客户端有新版本时会在启动时有相应地系统提示询问用户是否要更新最新的客户端版本如果没有新版本时则启动程序结束后直接进入主程序。

直播客户端主界面由热门主播、为你推荐、栏目、个人页面和直播分类五个子界面构成，一边启动主程序一边会将五个子界面的数据从服务器调过来，除了一些推荐信息以外，浏览记录以及收藏信息都会囊括其中，然后呈现给用户。整个加载主界面功能模块流程如图 5-1 所示。

(1) 加载主界面模块设计

在加载主界面模块用到各类作用如下：

1) MainActivity。负责处理公有事件，它提供 ViewPager，用于显示子界面，并外部存储数据进行初始化。是主界面 Activity 的基础类，极其重要

2) DataLoadingService。用于获得数据，它被 Activity 开启，从而可以去初始化主界面的数据或者其他设备数据，一般都是初始化的数据。而且还具有很重要的使得数据重新加载的功能，即便是在网络较差导致断网的时候也可以完成数据的重新加载。

3) **MainPageController**。可以从模型层读取数据，它最大的作用是用来对主界面所有需要进行异步处理与网络请求的工作进行处理。

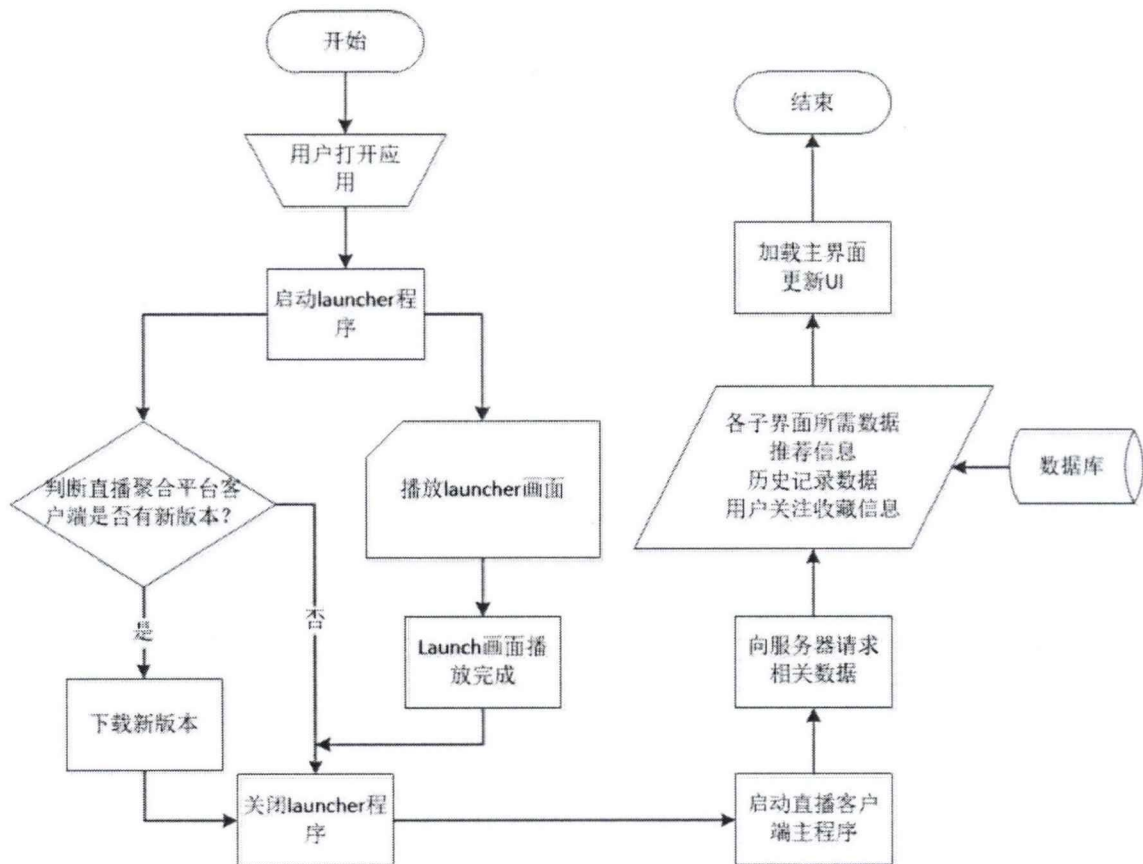


图 5-1 加载主界面流程图

Figure 5-1 Workflow Diagram for Main View Loading

4) **MainPageApi**。定义了生成 url 所需的基本信息，并提供了解析公共数据的方法，依据相关参数来完成，主要作用是生成相应的地址^[21]，之后从模型层获取数据再封装数据并将数据返回。

(2) 子界面相关类设计

主界面加载的同时系统服务器会请求子界面信息与此同时也加载子界面的所有图片，最终展现给用户为子界面。以子界面为例，介绍其相关类设计。

如图 5-3 所示为加载主界面同时加载子界面类图，除了图 5-2 中主要的 4 个类以外，还包括以下类：

1) **ImageUtil**。其主要作用是缓存图片，同时它还可以实现异步，在此条件下下载图片。

2) **TabRecommend**。顾名思义是对推荐页面布局进行一个定义与设置，并且会负责与用户的部分交互。

3) TabRecommendItemView。当推荐页面布局完成后，需要用这个类来定义相关推荐界面的格式，以及实现监听作用。

4) RecommendInfo。是推荐信息的 Bean 类。

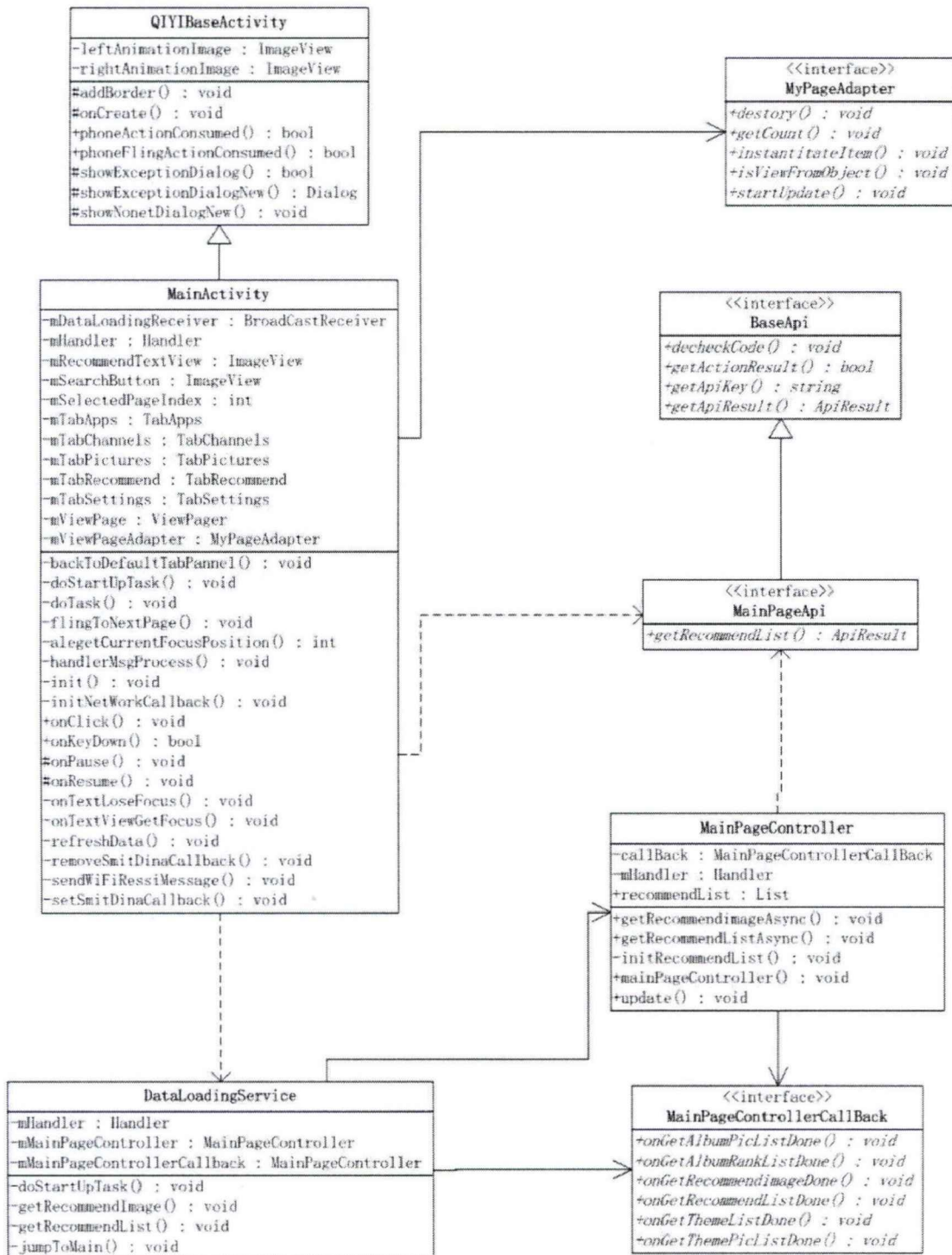


图 5-2 加载主界面类图
Figure 5-2 Main Class Diagram for Main View

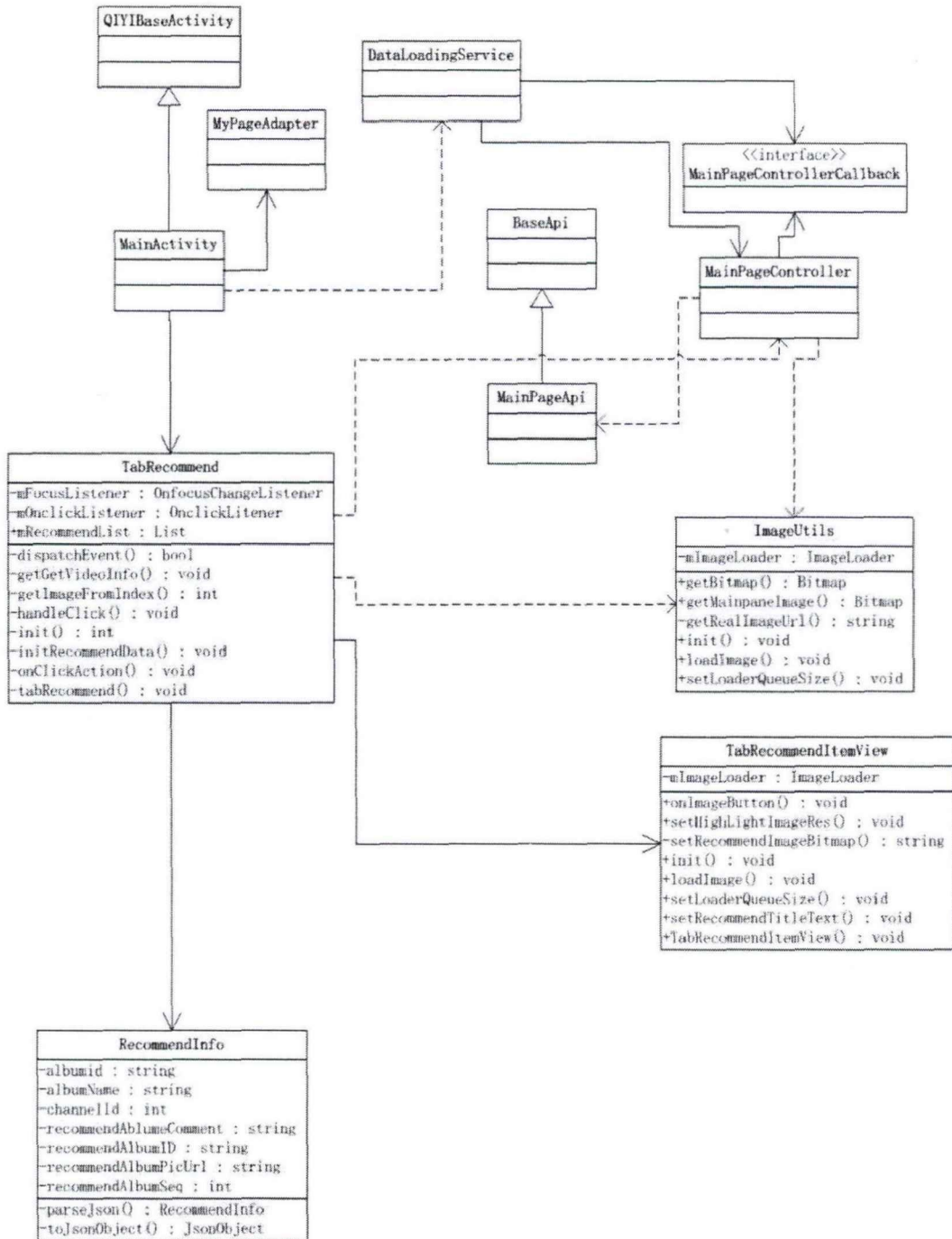


图 5-3 加载主界面同时加载子界面类图
Figure 5-3 Class Diagram for Main View Loading

(3) 主界面加载的实现逻辑

如图 5-4 所示，用户启动主界面类时便开始初始化主界面，此时相应的服务也同时开启，例如数据加载服务、推荐信息列表服务等，MainPageController 将开启异步任务来获得推荐信息列表，同时回调通知数据加载服务，在系统的推荐列表下载完成之后，MainPageController 进行下一步工作即直播封面图的获取，此时的

直播相关推荐列表以及下载完成，然后后台服务通过调用相关工具将下载好的图片保存到本地，当以上操作完成后后台的服务发送完成广播，主界面类接受此广播，然后更新推荐页面布局，至此，包含推荐子界面的主界面加载完成。该功能的时序图如图 5-4 所示。

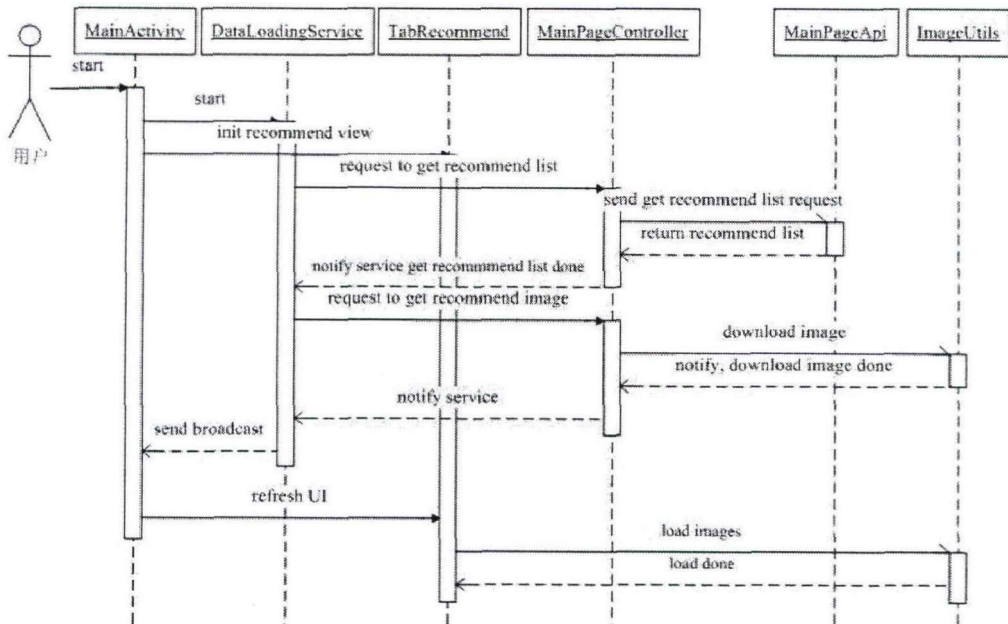


图 5-4 主界面加载功能时序图

Figure 5-4 Sequence Diagram for Main View Loading

依据类的设计与逻辑实现，通过编码现已实现该功能，实现图如图 5-5 所示。



图 5-5 加载主界面 launcher 启动界面截图

Figure 5-5 Interface for launcher Start

当 launcher 界面加载完成后, 会出现新版本更新提示, 提示用户更新版本, 用户可以选择“暂不升级”或者“立刻升级”, 此功能实现图如图 5-6 所示。

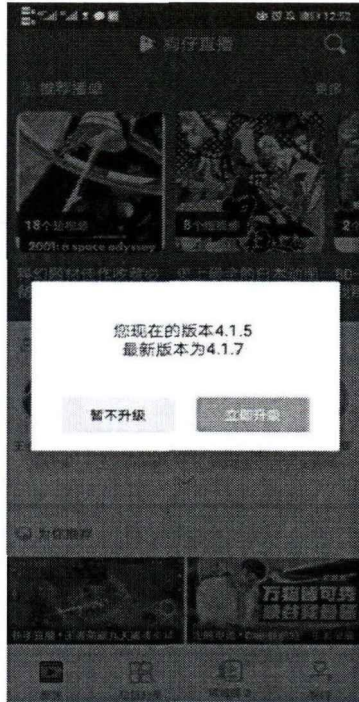


图 5-6 加载主界面版本更新界面截图
Figure 5-6 Interface for Main View Loading

当用户进入应用程序主界面时, 主界面以及子界面会同时加载完成, 此时子界面的布局信息已经下载完毕, 此功能实现图如图 5-7 所示。



图 5-7 应用程序主界面截图
Figure 5-7 Interface for Main Application

5.1.2 直播详情界面

从界面加载以及用户相关操作来看直播详情界面功能的实现，当用户点击某直播视频时，后台立即将获得该直播信息，然后与子界面加载服务一起将直播信息下载完毕，同时将获得此用户的历史播放记录信息以及收藏信息，并且依据有无相关同类型直播来决定更新哪一类型的相关推荐列表，此时直播的详情界面加载完毕，该模块的功能流程图如图 5-8 所示。

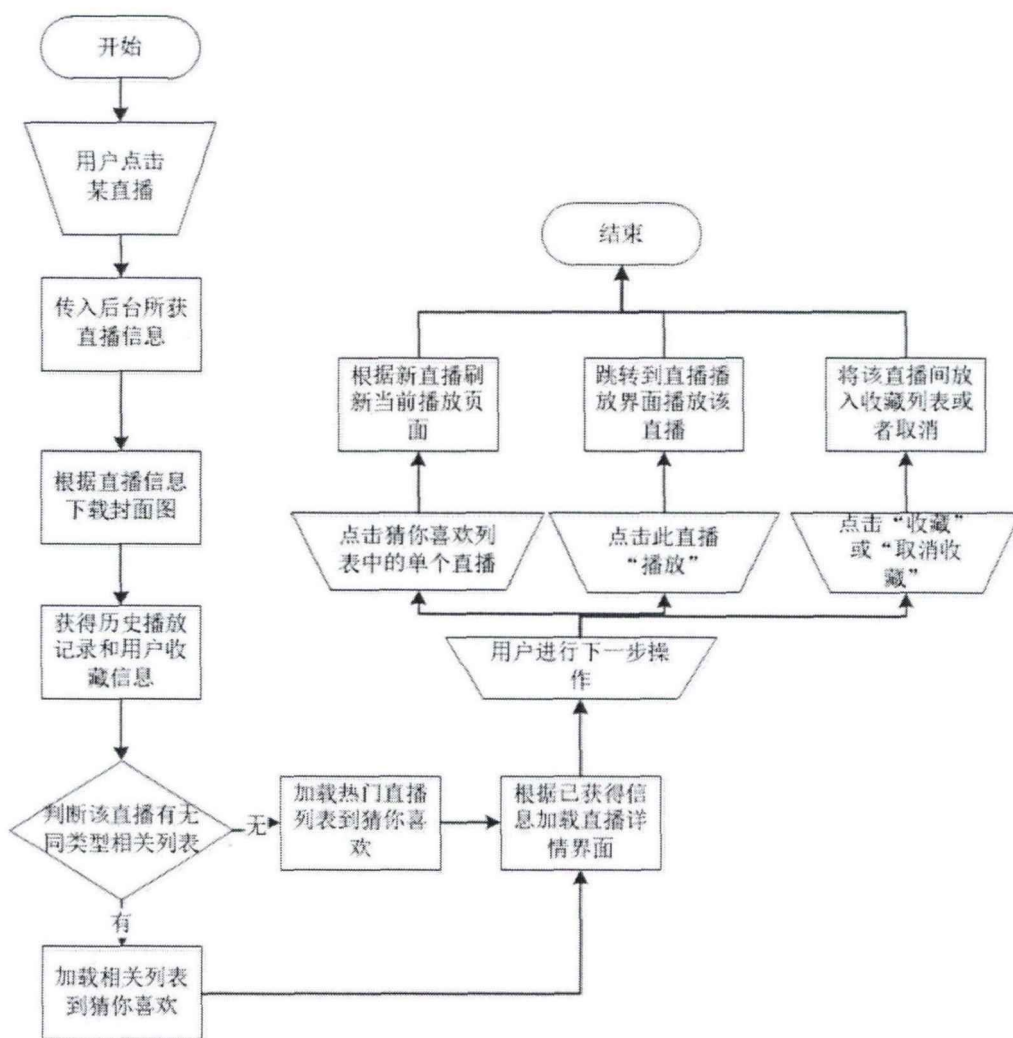


图 5-8 直播详情界面加载及操作流程

Figure 5-8 Workflow Diagram for Live Detail View Loading and Operation

(1) 直播详情相关的类设计

该功能用到的类主要有 AlbumLDetailActivity、ViewLiveDetail、AlbumDetailBottomPanel、GuessyouLikeWidget、EpisodeGridWidget，业务逻辑实现类有 VideoInfoController、AlbumPhotoController、HistoryLController、FavorLController，用于进行数据处理的类有 VideoInfoApi、LivePhotoApi、UserLApi、

ImageLUtills, Bean 类主要是 AlbumLInfo。以上这些类实现了界面布局, 类的关系设计如图 5-9 所示。

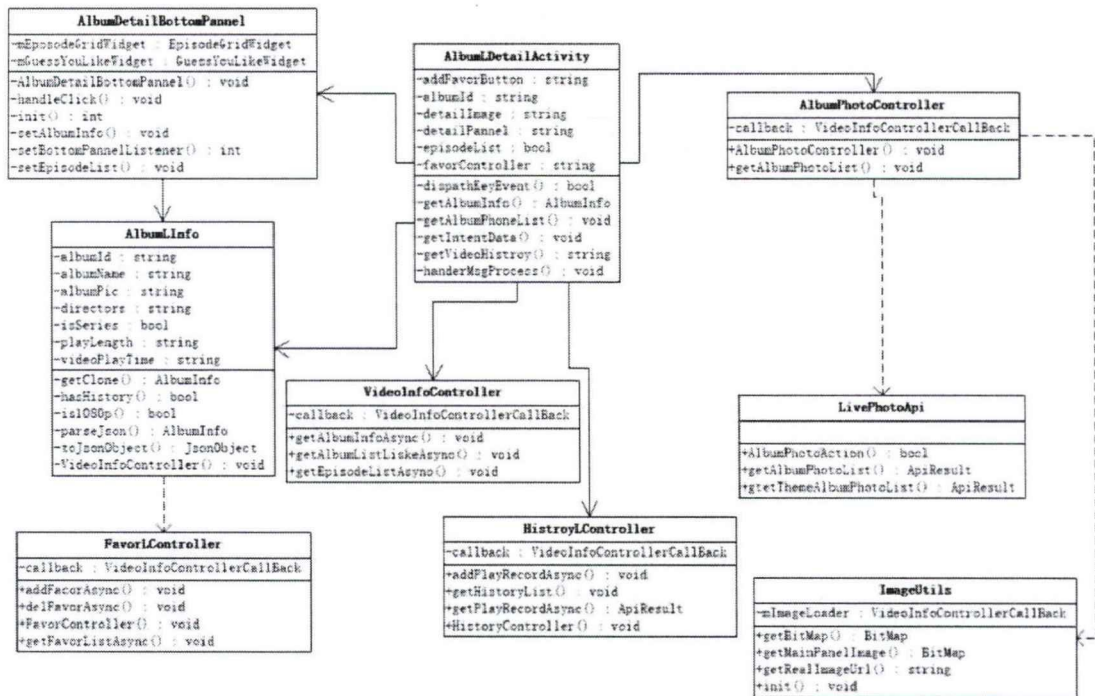


图 5-9 直播详情界面功能类图
Figure 5-9 Class Diagram for Live Detail

1) AlbumLDetailActivity 是直播详情页的 Activity。在布局上, 将 ViewLAlbumDetail 位于页面上端, 用于显示直播详细信息 (直播封面图、直播名称、主播、简介等) 以及播放、收藏等功能按钮; AlbumDetailBottomPanel 在页面下部, 负责相关直播列表的布局, 它保留了 GuessyouLikeWidget 和 EpisodeGridLWidget 的引用。在加载直播详情页时向服务器请求相关直播列表的数据以及图片信息, 此时 AlbumLDetailActivity 里定义了一个 GuessYouLikeWidget 的实例, 由它来存储数据, 然后设置到 AlbumDetailBottomPanel 的 GuessYouLikeWidget 里, 在页面加载时更新到 UI 上。

2) VideoInfoController、AlbumphotoController、HistoryLController、FavorLController 是本系统的业务逻辑控制器, 均有一个共同的特点就是都可以开启异步线程, 此时它们会分别调用 VideoInfoApi、LivePhotoApi、UserLApi, 然后通过对于服务器的请求而获得直播详细信息以及所需的收藏列表和历史信息。

3) Bean 类是由 VideoInfoApi、AlbumLPhotoApi、UserLApi 从服务器获得的数据封装而成: 然后将 List 返回给各自的业务逻辑控制器。

4) AlbumLInfo 是获得详细信息的直播 Bean 类, ImageLUtills 负责异步下载直

播封面图片并进行缓存。

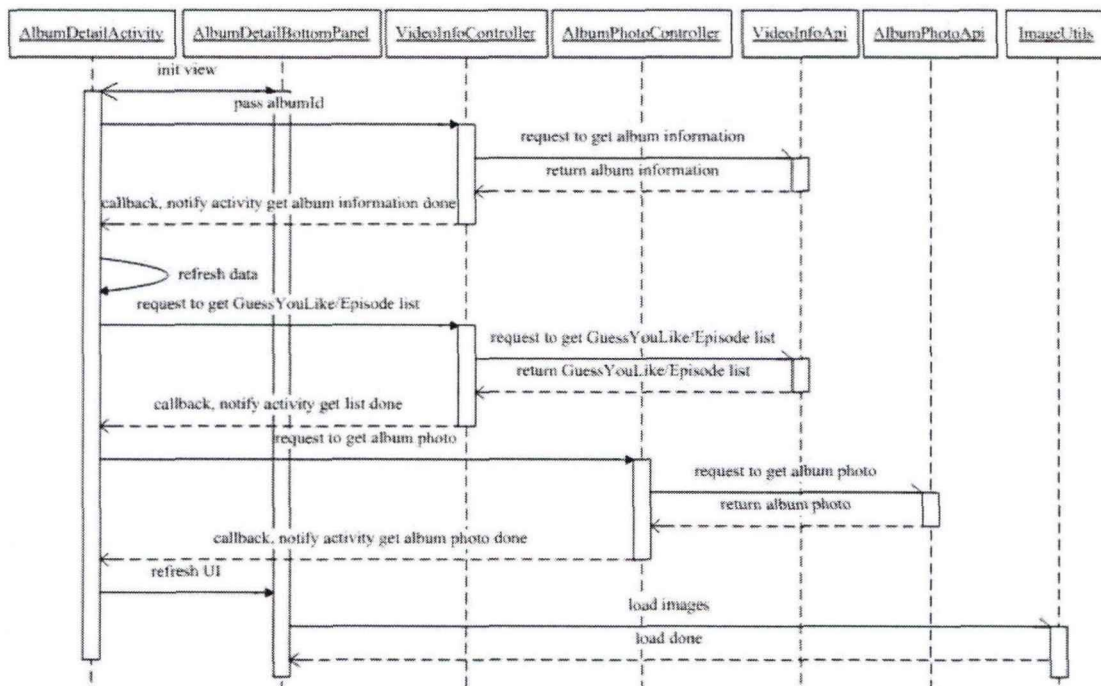


图 5-10 直播详情界面加载时序图

Figure 5-10 Sequence Diagram for Live Detail View Loading

(2) 直播详情界面加载的实现逻辑

如图 5-10 所示，以进入直播页时为例，当用户点击某个直播进入详情页时，AlbumDetailActivity 开始初始化页面布局，同时将命令传给控制器，通知其下载直播详细信息。此时 VideoInfoController 开启任务，通过 API 从服务器下载该直播的详细信息。之后以相同的方法下载猜你喜欢列表。即 VideoInfoController 调用 VideoInfoApi 从服务器下载猜你喜欢相关列表^[22]，下载完成后，将数据封装回调。下一步，再向控制器发送下载直播封面图的请求，同样开启异步任务调用相关 API 从服务器下载图片，完成后将数据封装回调。最后更新界面布局和视图并显示加载好的直播封面图。

在加载直播详细信息显示到界面上的同时还需要去获得该直播的历史播放记录，并且判定该直播否已经被用户收藏，之后根据判断的结果去更新界面信息，这里用到了 HistoryLController 和 FavorLController，由于直播历史播放记录在直播相信信息中有记录，之前已经获取得到，所以可以直接从缓存中读取数据，更新并显示到界面。当用户通过点击按钮实现收藏或取消收藏时，AlbumDetailActivity 通知但是用户收藏列表将要以相同的方法通过 FavorLController 开启异步任务调用相关 API 从服务器获取数据，待到直播详细信息以及收藏列表和播放历史记录都获得后怎可以更新界面，整个直播详情界面功能完成的时序图如图 5-11 所示。

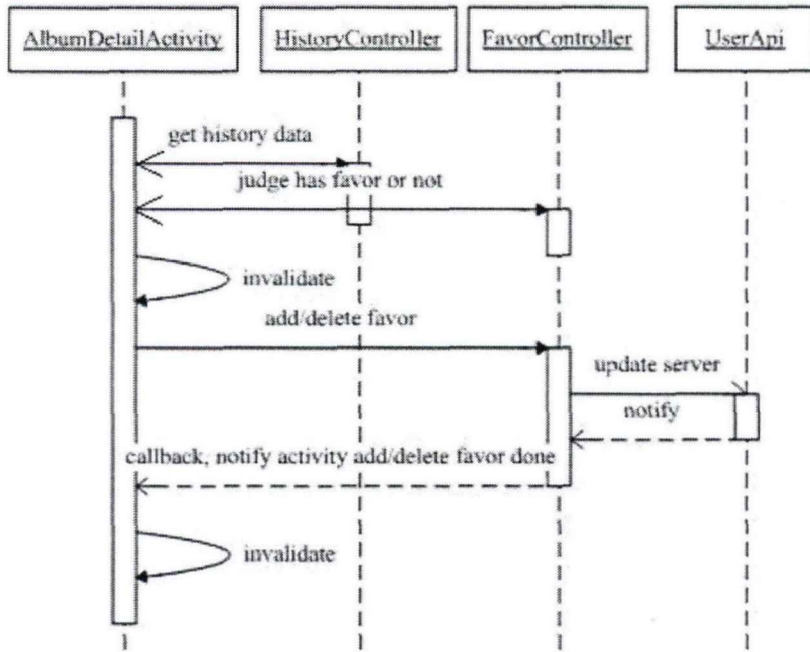


图 5-11 直播详情界面功能时序图
Figure 5-11 Sequence Diagram for Live Detail View

根据类的设计与逻辑实现，通过编码，已实现该功能，界面截图如图 5-12 所示。



图 5-12 直播详情界面实现图
Figure 5-12 Interface for Live Detail View

5.2 直播搜索模块

直播搜索功能以及搜索界面的初始化展示共同构成了直播搜索模块，当用户进入搜索界面时，在加载页面布局的同时将会被提供当前一周内的主要搜索热词显示在搜索框下方，然后后台将依据用户点击热词与否以及输入搜索框的内容来动态更新搜索内容，从而更新搜索的结果页。直播搜索模块功能流程如图 5-13 所示。

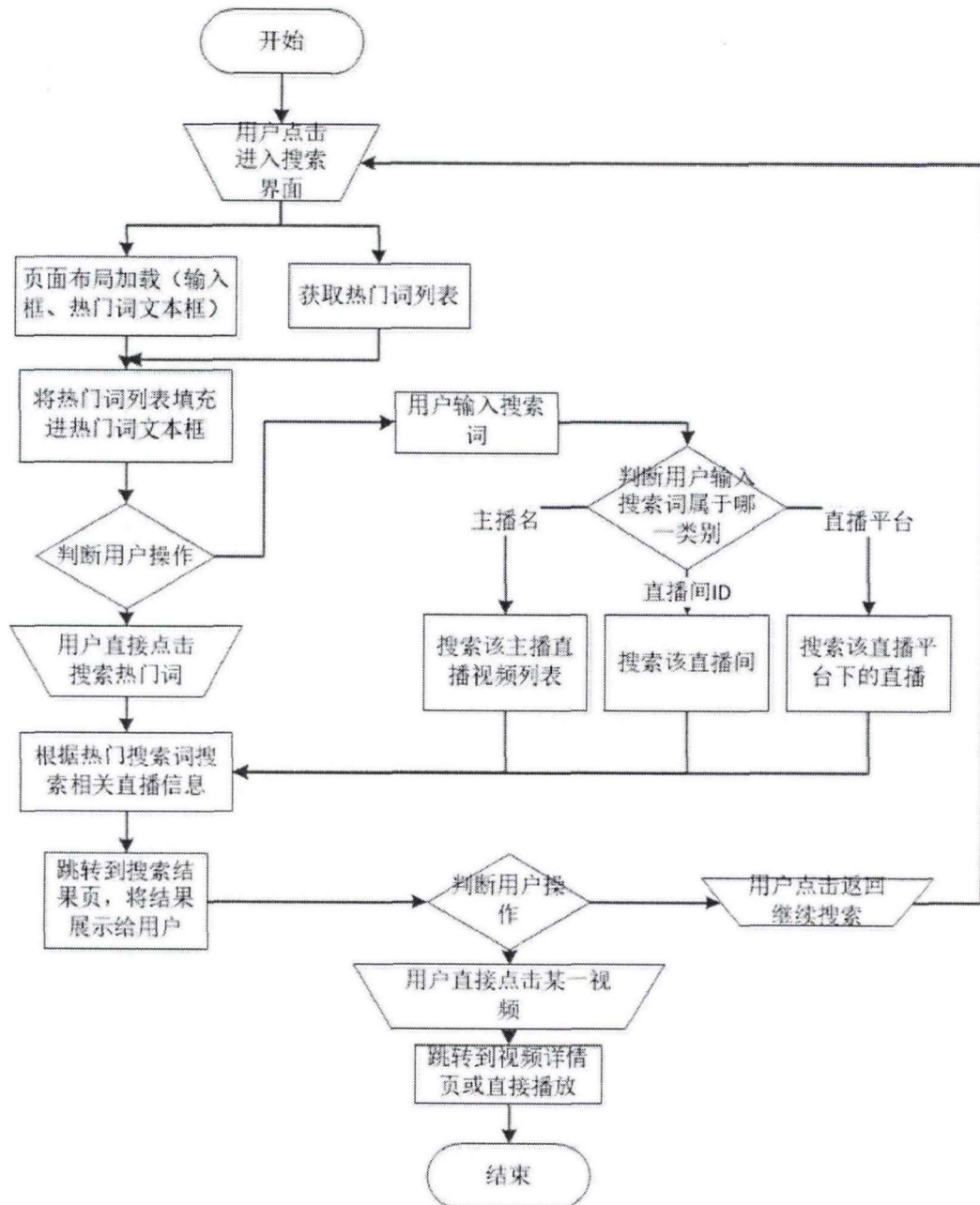


图 5-13 直播搜索功能流程图

Figure 5-13 Workflow Diagram for Live Searching

(1) 直播搜索功能的类设计

SearchActivity 和 SearchResultActivity 是用来控制界面布局的类，SearchLController 是用于实现业务逻辑的有，SearchLapi 是用于进行数据处理的类，Bean 类：RealtimeSearchResult。如图 5-14 所示。

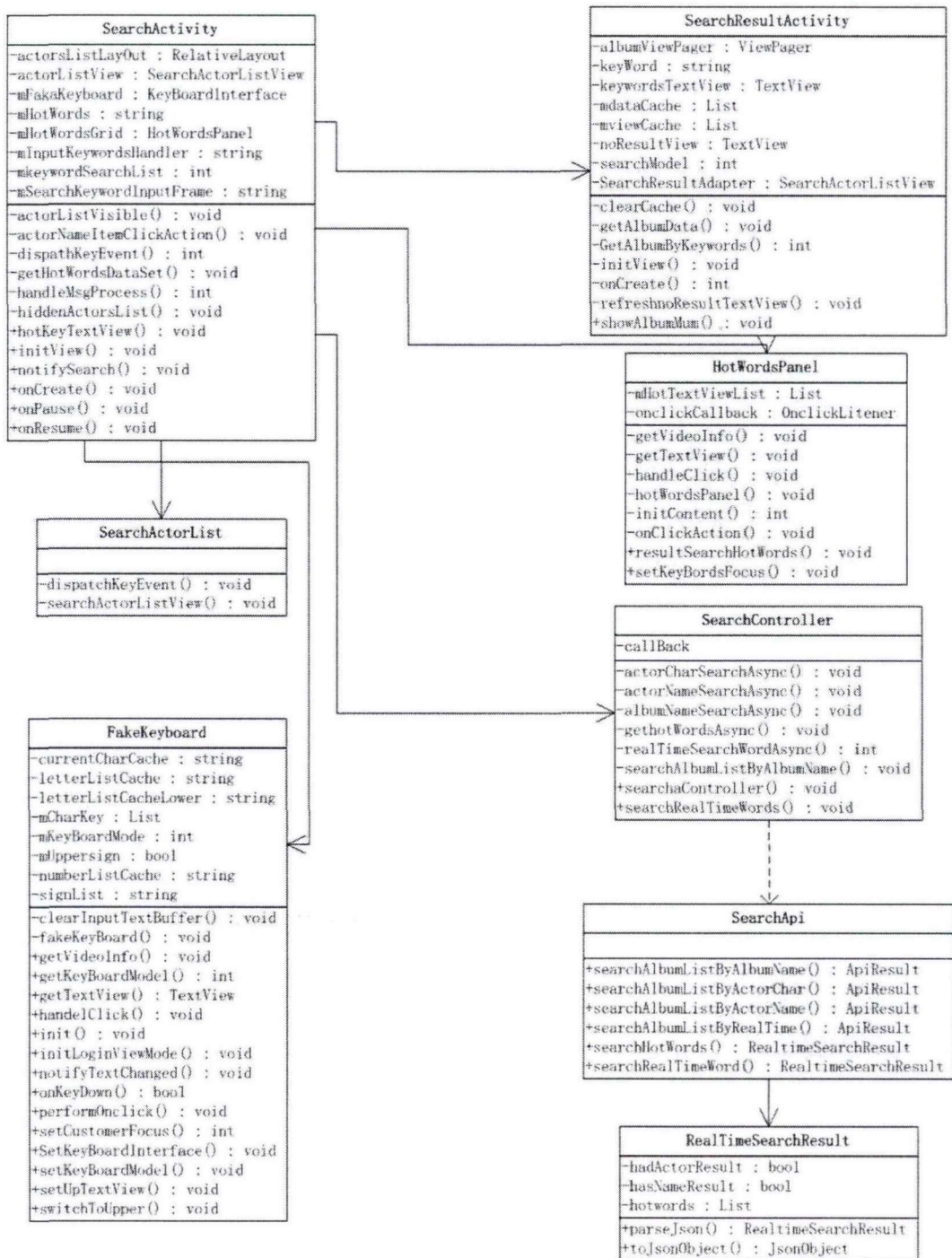


图 5-14 直播搜索功能类图
Figure 5-14 Class Diagram for Live Searching

1) SearchLActivity 负责搜索界面。SearchResultLActivity 负责搜索结果界面。EdiLtText、HotwordsLPanel 和 FakeLKeyboard, 以及搜索主播时弹出的用于显示主播列表的 SearchActorListView 共同构成了搜索界面。在界面初始化时, 它们还设置了保存数据缓存、页面缓存、时下热词等信息。

2) SearchLController。负责处理具体的搜索逻辑^[23], 它开启异步线程, 进行热门词实时搜索、按直播间名、按主搜索、按主播名字或者首字母、按照直播平台名字、按直播间 ID 等, 通过调研 API 向服务器请求搜索到的直播结果, 并将数据封装后返回给 SearchLController。

3) RealtimeSearchResult。属于 Bean 类, 主要作用是存放当今时下热门搜索词。

(2) 直播搜索功能的实现逻辑

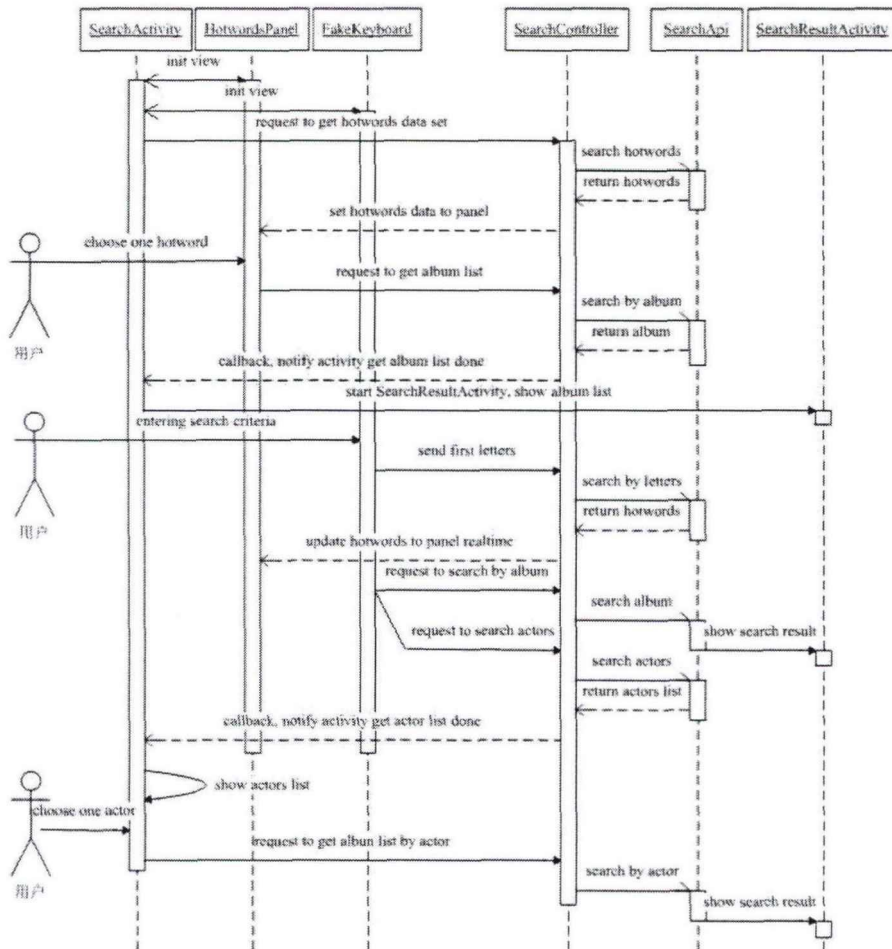


图 5-15 直播搜索功能时序图

Figure 5-15 Sequence Diagram Diagram for Live Searching

直播搜索功能的实现逻辑时序图如图 5-15 所示。当用户进入直播搜索界面时, SearchLActivity 将会被启动, 然后进行初始化加载页面布局, 同时加载好关键字输

入框以及热门词面板。此时后台数据库传输来的热门词列表数据会填充到热门词面板，那么因为在这里将涉及到用户的操作，不同的操作会影响流程从而出现不同的结果：

如果用户选择直接点击热门词，那么此时 `SearchLController` 会获取一个来自于 `SearchLActivity` 的直播列表请求，并且传递相关用户的参数。然后 `SearchLController` 开启异步任务调用 API 搜索相关直播，然后将数据封装并返回给 `SearchLController`，最后由 `SearchLActivity` 将搜索结果显示出来。

如果是用户选择在输入框自行输入关键字搜索时，`SearchLController` 搜索热门词列表将会被实时调用，并且新的热词列表会被更新^[24]，如果此时用户选择新的热门词，则进程和之前的相同，如果用户点击搜索按钮，那么控制器将根据输入的关键字开启异步调用 API 进行搜索并将搜索结果返回，关于模糊还是精确搜索，将与用户输入的关键字有关，后台将调用关键字匹配技术去识别采用模糊匹配还是精确匹配，最终将搜索到的结果显示出来。

通过对于直播搜索模块的逻辑设计，经过编码，现已实现该功能，在搜索界面可以提供搜索框。在搜索框下方会有实时搜索热词，用户可以实现按照主播名字、直播平台名称、直播间 ID、推荐热门词进行搜索，功能实现界面如图 5-16 所示。



图 5-16 直播搜索功能界面截图
Figure 5-16 Interface for Live Searching

5.3 直播播放模块

播放界面加载和用户直播播放的操作控制共同构成了直播播放模块。用户点击自己认为有趣的直播视频信息或者直接搜索想要看的直播视频时^[25]，播放界面就开始加载，同时关于播放功能的子界面布局以及相关操作按钮也加载完成。本课题使用 Ijkplayer 以及 VideoView 视频框架来实现直播播放功能。

(1) 直播播放界面的类设计

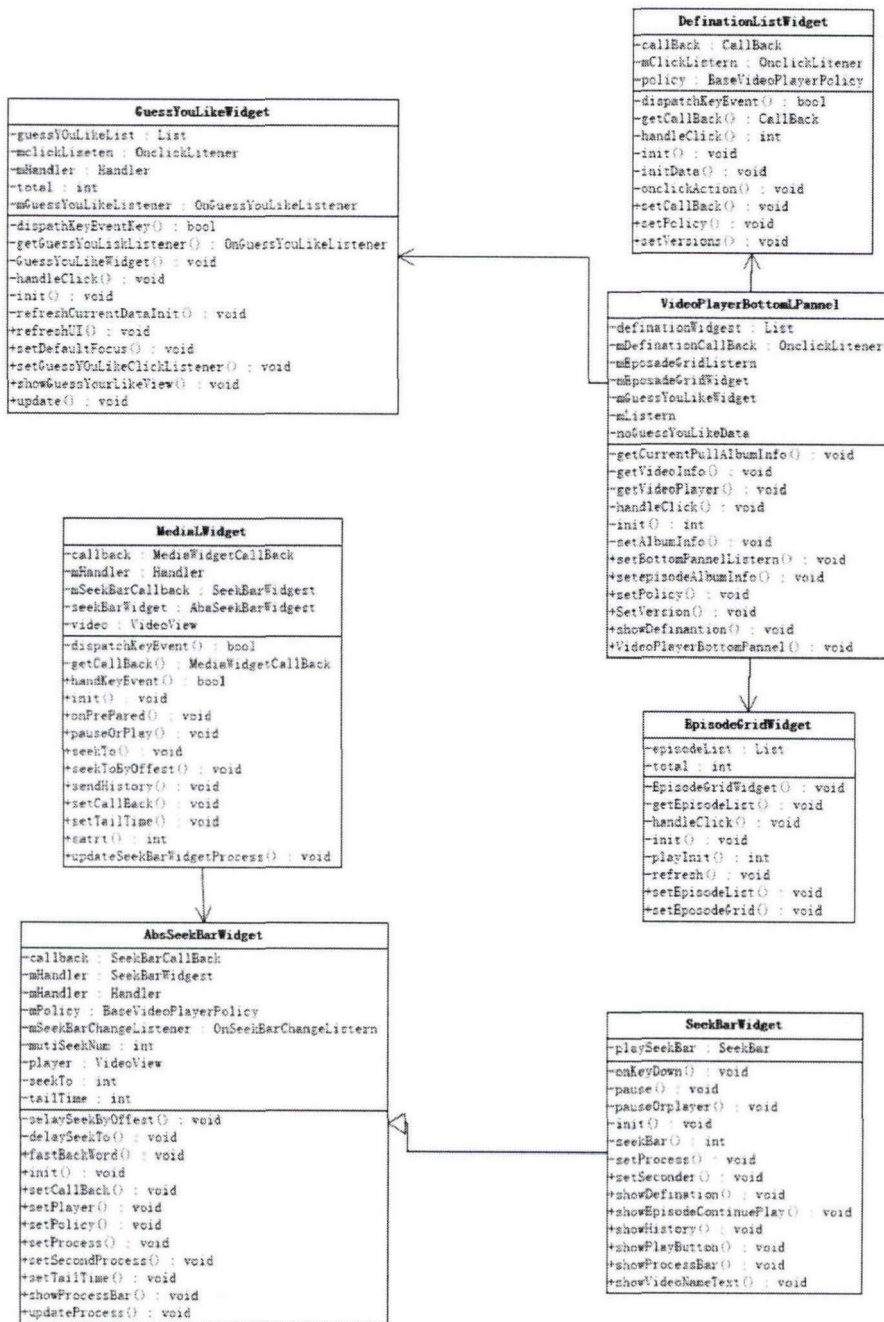


图 5-17 直播播放界面类图

Figure 5-17 Class Diagram for Live Playback Layout

如图 5-17 所示，播放界面类有：MediaLWidget 和 VideoPlayerBottomLPanel。MediaLWidget 是直播播放区域，VideoPlayerBottomLPanel 是底部的控制面板包含很多控件，例如分管直播暂停与开始的控件，以及与猜你喜欢列表相关的控件。

(2) 直播播放功能的类设计

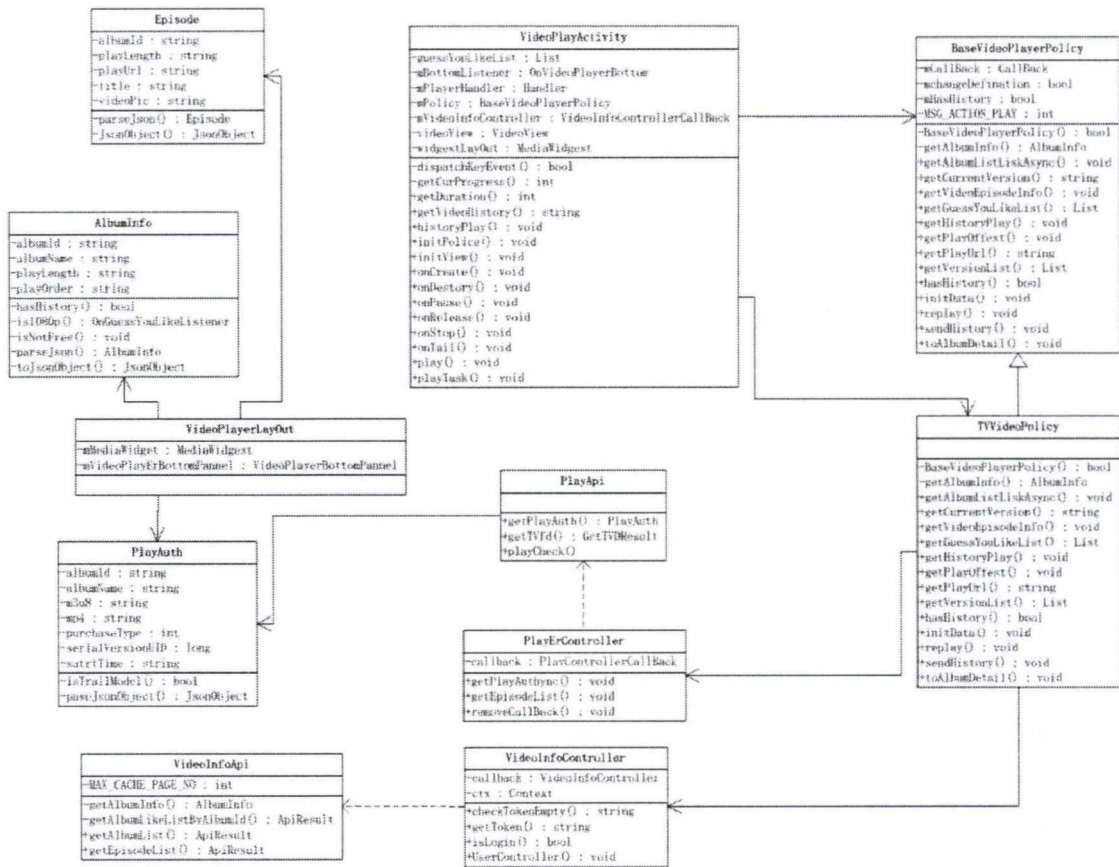


图 5-18 直播播放功能类图
Figure 5-18 Class Diagram for Live Playback

如图 5-18 所示，在直播播放功能用到的类主要有：VideoPlayLActivity 是界面布局控制；TVVideoLPolicy、PlayLController、VideoInfoController 是业务逻辑类；PlayLApi、VideoInfoApi 是数据处理的类；AlbumInfo 是 Bean 类。

1) VideoPlayLActivity。主要用于初始化整个播放界面的布局，并且申请启动直播的播放服务

2) TVVideoLPolicy。使用各种控制器处理业务逻辑^[26]，例如播放直播用到了 PlayController，获得猜你喜欢列表、直播列表、直播详细信息用到了 VideoInfoController，播放直播用到了 UserController，获得和更新直播播放历史记录用到了 HistoryController。

(3) 直播播放的实现逻辑

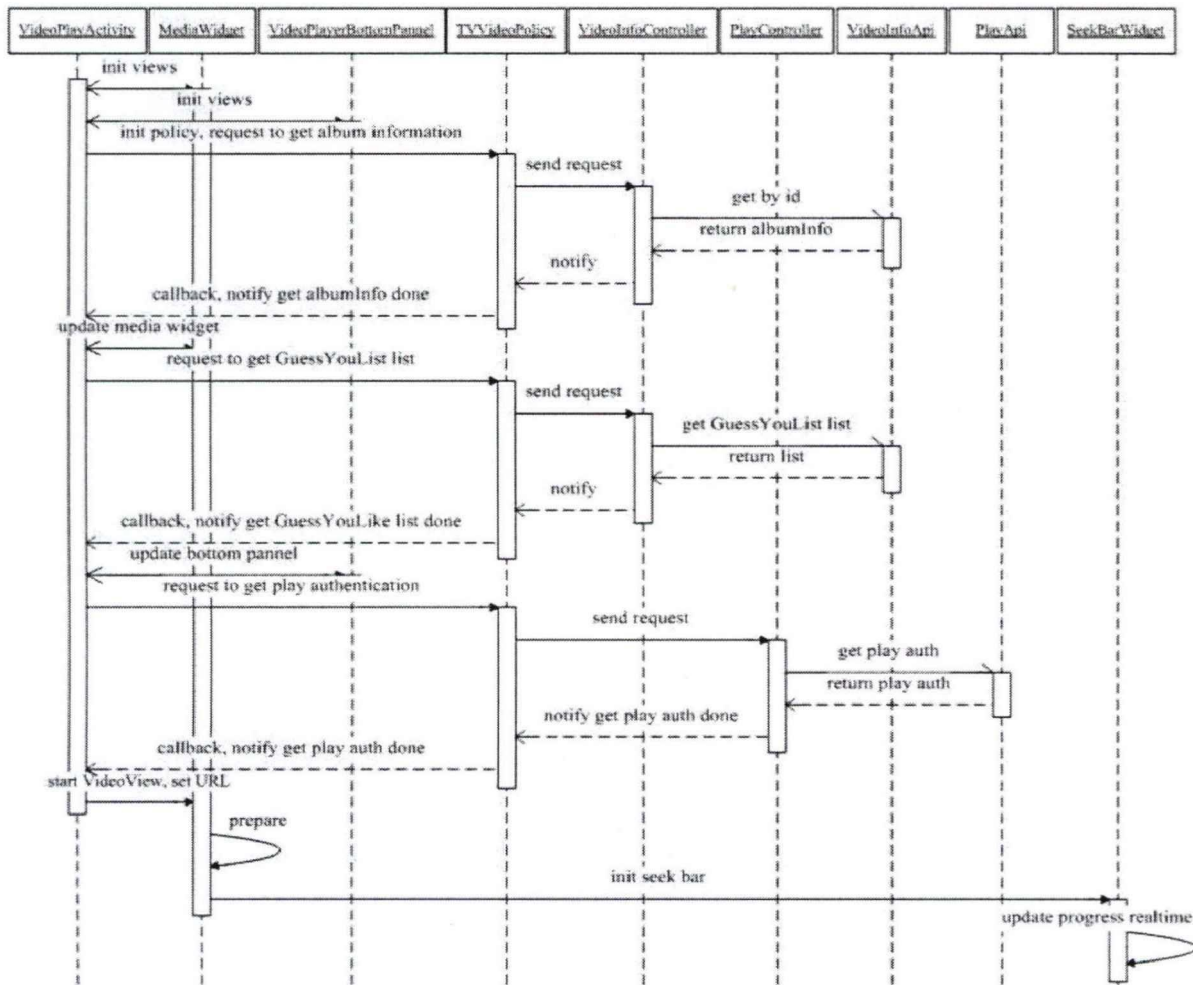


图 5-19 直播播放页面加载时序图

Figure 5-19 Sequence Diagram for Live Playback View Loading

如图 5-19 所示，用户遇到感兴趣的直播视频时，点击其中一个进行播放，那么首先由 `VideoPlayLActivity` 初始化页面布局并进行一系列数据的初始化请求工作，然后 `VideoInfoController` 开启异步线程调用相关 API 获取直播信息；主要是通过地址反解析技术获得的直播数据然后通过 `Ijkplayer` 二次编译后融合到直播平台中的数据，当这些直播数据被获取以后，`HistoryLController` 则会判断该直播是否有播放历史记录；同时开启另一个异步线程调用相关 API 获取猜你喜欢列表。然后这些 `Controller` 将搜索结果返回，将相应的信息填充到控件上，更新播放界面，另外还会进行一些数据完整性检查以及播放鉴权的相关服务调用，这些都完成以后直播页面的播放界面加载完成，然后 `MediaWidget` 收到来自 `VideoPlayLActivity` 的直播地址 `url`，启动播放服务，进行直播播放之前最后的准备工作。

根据对于以上类的设计与逻辑实现，通过编码，已实现该功能，界面截图如

图 5-20 所示。

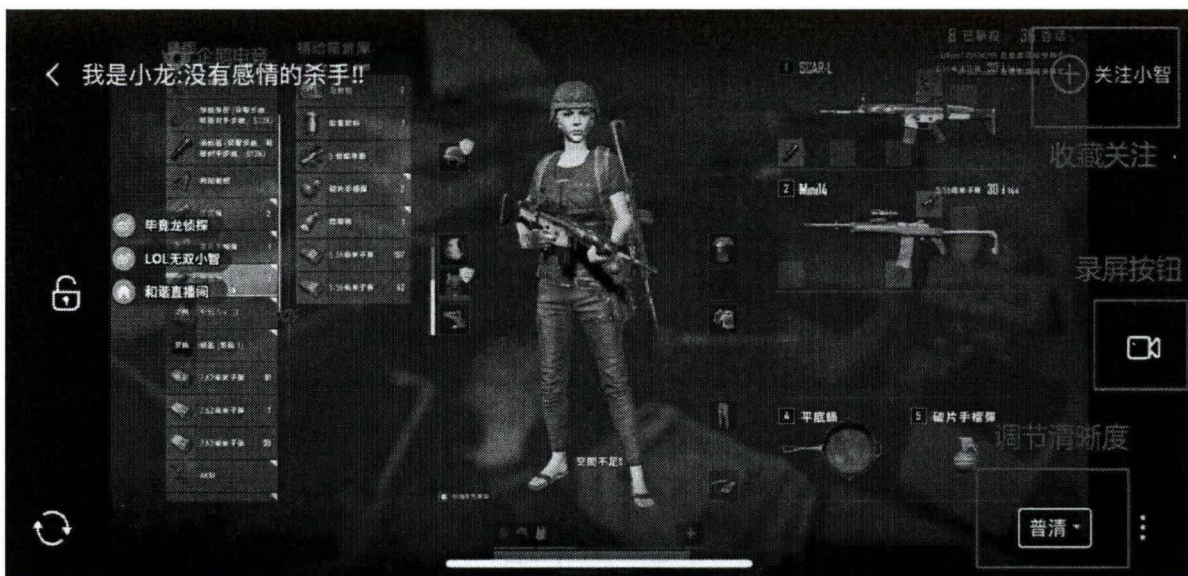


图 5-20 直播播放页面实现截图

Figure 5-20 Interface Diagram for Live Playback View

5.4 用户登录模块

本直播平台用户可以使用手机号注册账号登录，同时用户也可以使用第三方账号登录，例如当用户选择 QQ 第三方登录时，会通过 AppId、回调地址和随机码生成 URL,当用户登录成功后会通过 回调地址返回 code,通过调用 OauthController 中的 QQoauth 方法通过 code 获取用户的 AccessToken，然后使用 AccessToken 调用第三方接口获取用户 OpenId 得到用户头像、性别等身份信息封装进 map 中返回给前端。微信登录流程与此相同。注登录应用可以实现查看自己关注的主播、收藏直播间，转发分享等游客不能实现的功能。

当用户使用直播聚合平台客户端时，首先默认是以游客的身份使用本应用，然后在用户想要进行更多的个人功能操作时就会出现提示用户登录的界面，此时用户来到个人登录界面，可以选择手机号登录或者第三方登录，对于手机号登录，则需要用户先点击注册按钮，注册后即可登录；对于第三方登录则提供腾讯微信、QQ 以及微博的登录接口，用户可以将第三方信息与本人手机号信息进行绑定。同时用户的个人登录信息也会被缓存到本地数据库，这样就更加方便用户下次登录，当用户登录完成后即会有很多的个人功能类似收藏、转发分享、评论等功能可以被解锁使用。

用户登录模块的功能流程图，如图 5-21 所示。

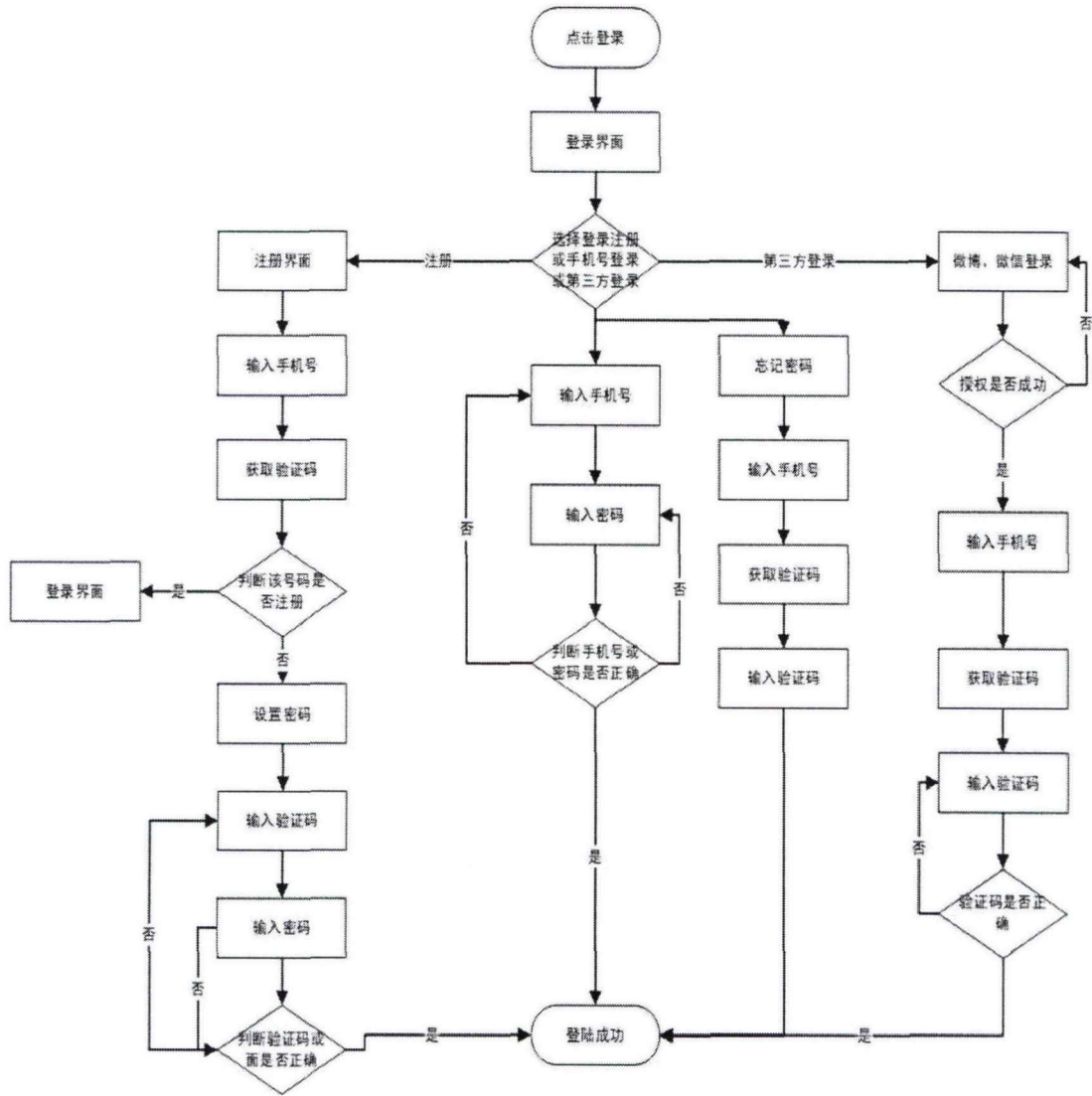


图 5-21 用户登录功能流程图

Figure 5-21 Workflow Diagram of User Login function

如图 5-22 是用户基本事件里的登陆功能模块时序图。

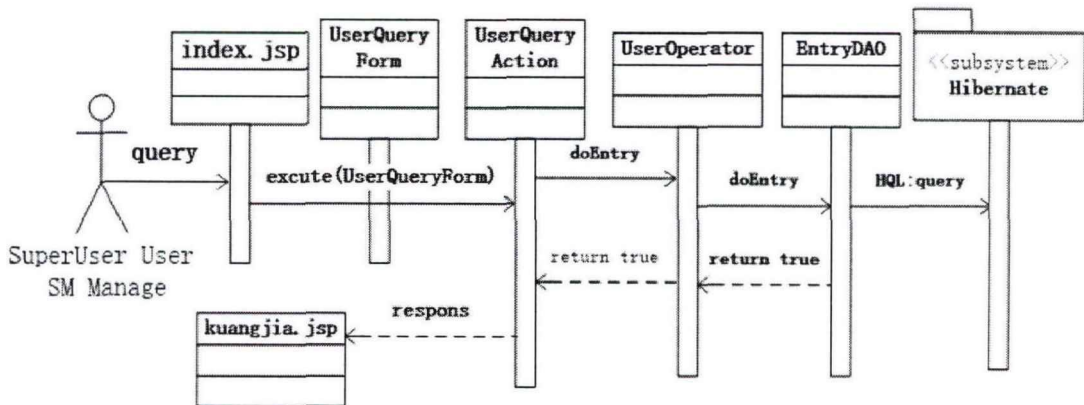


图 5-22 用户登录功能时序图

Figure 5-22 User Login function class diagram

步骤说明：

- 1.当用户进入 index.jsp 页面，并根据页面内容填写，填写完成后，点击“登录”。
- 2.Index.jsp 将请求登录信息提交到 UserQueryAction，信息被封装传递到 UserQueryAction。UserOperator 内的 doEntry 方法被 UserQueryAction 调用，然后通过 UserDAO 调用 Hibernate 框架完成数据的匹配。
- 3.如果返回值为 true， UserQueryAction 会接收到 UserOperator 返回的值，然后 UserQueryAction 调用 response 方法进入 kuangjia.jsp 页面，完成登录。

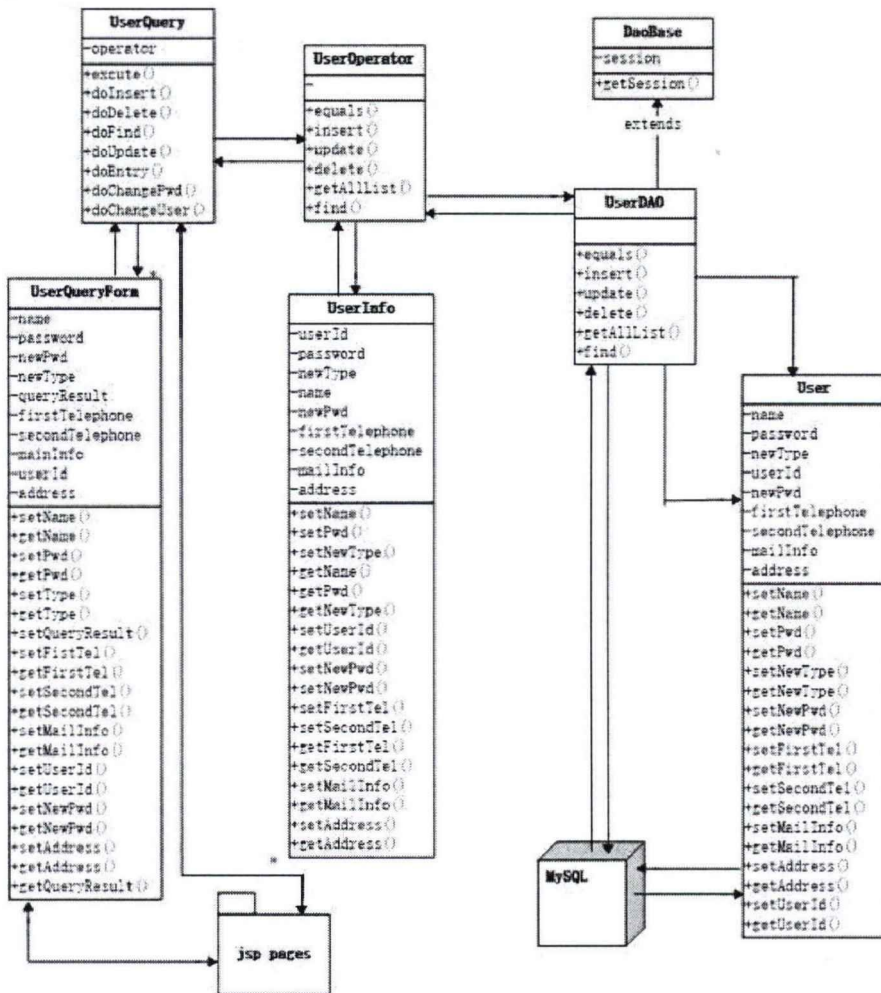


图 5-23 用户登录功能类图
Figure 5-23 User Login function class diagram

类之间的关系如图 5-23 所示。用户登录模块相关的类设计该功能涉及的类主要有 UserQuery、UserQueryform、UserOperator、UserInfo、User, UserDAO。根据以上类的设计与逻辑实现，通过编码，已实现该功能，在用户登录主界面可以实现用户注册以及手机号登录，同时也提供第三方 QQ、微信、微博登录方式，用户登录主界面截图如图 5-24 所示。



图 5-24 用户登录功能主界面图
Figure 5-24 Interface of User Login function

当用户点击快速注册时，即可以跳转到注册界面，此功能实现截图如图 5-25 所示。



图 5-25 用户登录功能界面图
Figure 5-25 Interface of User Registered Function

5.5 直播聚合模块

直播聚合功能的实现主要是以下流程：首先在聚合各家平台之前，先调用地址反解析协议解析应用平台地址，然后传回判断应用地址是否可用，如果可用，则验证直播信息是否可用，如果可用的话那就同步该直播，同时将相关直播信息写入数据库；如果在验证解析地址后发现不可用那么就显示直播异常^[26]，或者在验证直播信息的同时不可用，也要在详情界面显示直播异常。然后更新直播界面将这些直播信息都显示到界面上

该模块的功能流程图如图 5-26 所示。

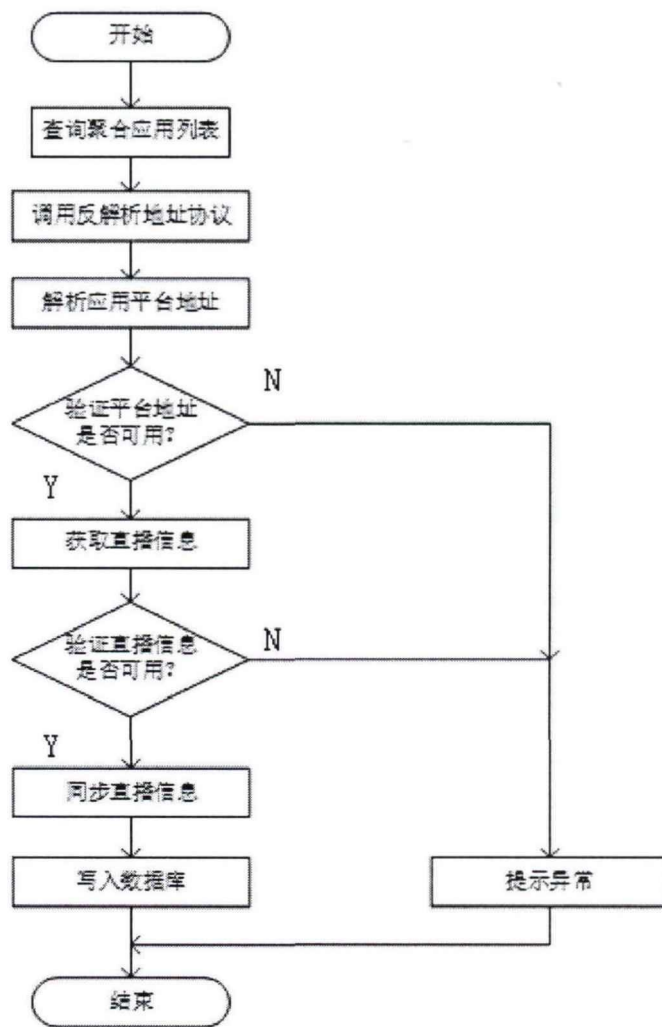


图 5-26 直播聚合功能流程图

Figure 5-26 Workflow Diagram of Live Aggregation

直播聚合功能模块的类的设计如图 5-28 所示。

聚合服务类 `AggregationService` 通过调用应用 `AppDao` 获得直播应用列表^[28],

然后去调用 `analysisUtil` 来进行应用地址的反解析，然后由 `checkUtil` 类去验证地址，如果是正确的，则由类 `AggregationDao` 将其信息存储到数据库中。

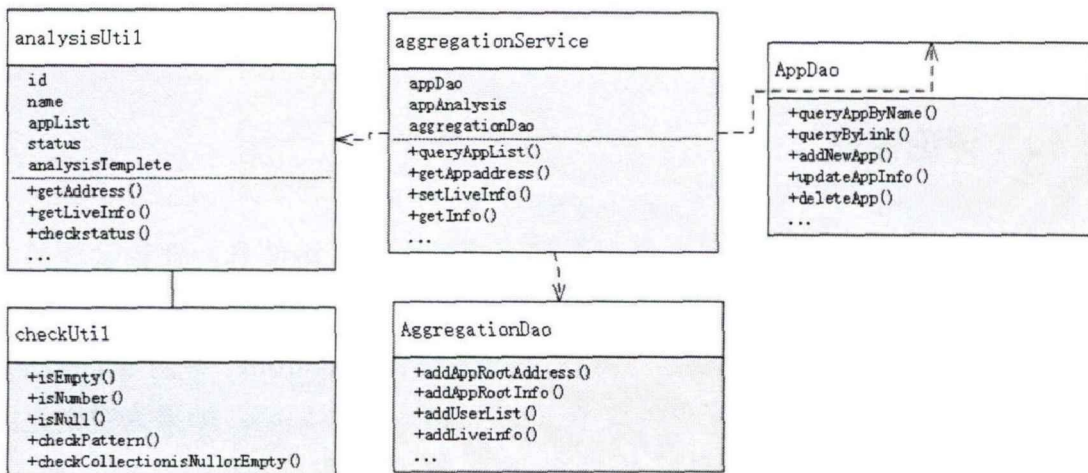


图 5-27 直播聚合功能类图

Figure 5-27 Class Diagram of Live Aggregation

根据类的设计与逻辑实现，通过编码，已实现该功能，可以看到在同一个界面能够显示不同直播平台的直播节目，界面截图如图 5-29 所示。



图 5-28 直播聚合功能界面展示图

Figure 5-28 Interface Diagram of Live Aggregation

5.6 直播数据管理模块

直播数据管理模块主要面向管理员，本人在其中参与设计与实现的功能是评论管理功能以及违规内容管理功能。

5.6.1 评论管理

当管理员点击评论管理，便会看到违规的所有评论信息，评论管理的类图如图 5-30 所示。

当管理员点击评论管理时，LiveRoomCommentController 中的 list 方法开始被调用，此方法首先会判断请求参数中 liveRoomId 是否为空，如果为空则返回错误提示，如果不为空，则会将前端的请求数据传入 Query 对象，然后调用 LiveRoomCommentService 中的 queryList 方法并将 Query 传入，该方法会继续调用 LiveRoomCommentDao 中的 queryList 方法，将查询到的数据封装为 LiveRoomCommentEntity 对象放入 List 中返回给前端^[26]。

管理员还可以对评论进行批量删除，当管理员点击删除按钮时，LiveRoomCommentController 中的 deleteBatch 方法开始被调用，LiveRoomCommentService 中的 deleteBatch 方法会继续调用 LiveRoomCommentDao 中的 deleteBatch 方法并将评论从数据库中删除，删除成功返回删除成功消息。

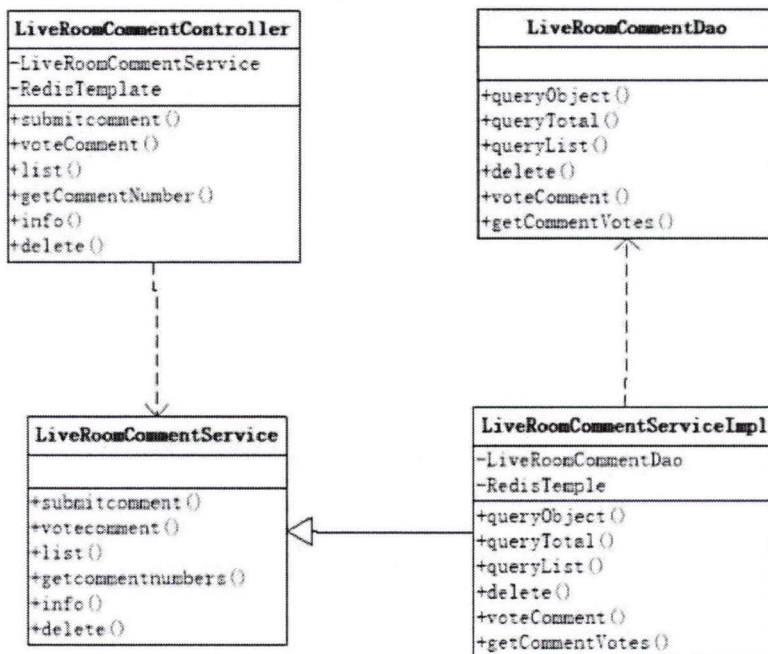


图 5-29 评论管理功能类图

Figure 5-29 Class Diagram of Comments Management

5.6.2 违规内容管理

违规内容管理的类图如图 5-31 所示。

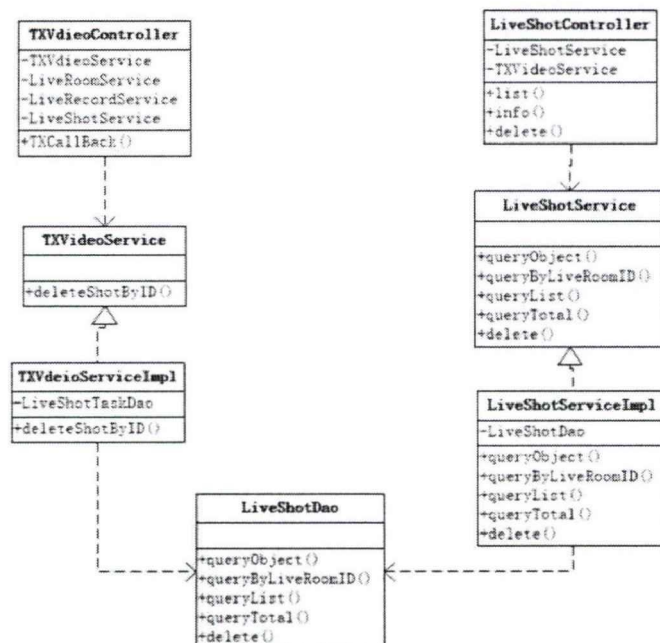


图 5-30 违规内容管理功能类图

Figure 5-30 Class Diagram of Illegal Content Management

当有疑似违规内容被服务器鉴定出来后，便会以 JSON 的格式发送消息通知到管理端，通过消息通知中的 `Event_type` 值是否超过 200 来判断有无检测到违规画面，如果超过 200 则提取消息中直播地址等属性，然后封装进 `LiveShotEntity` 实体中，调用 `save` 方法，并将 `LiveShotEntity` 传入，然后继续调用 `liveShotDao` 中的 `save` 方法，将属性信息存储到数据库中，并返回保存成功提示。

当管理员点击违规内容管理，`liveShotController` 中的 `list` 方法开始被调用，此时涉及违规内容的截图就会在页面中显示。同时前端传来的 JSON 格式的请求数据会被转换为 `map` 结构，然后将 `map` 封装为 `Query` 对象，然后 `liveShotService` 中的 `queryList` 方法和 `queryTotal` 方法被调用，将 `Query` 对象传入 `queryList` 方法，`liveShotDao` 中的 `queryList` 方法继续被调用，从而可以在数据库中查询出截图文件的记录，并返回 `List` 格式的数据，`liveShotDao` 中的 `queryToal` 方法也会被调用，目的是用来获取录制文件的总数并返回，然后 `liveShotController` 将查询到的录制文相关数据传入 `PageUtils`，再放进 `map` 中返回到前端。

当管理员点击删除时，开始调用 `liveShotController` 中的 `delete` 方法，该违规内容截图文件即从服务器存储中被删除。先通过 `delete` 方法获取所选文件编号，然后调用 `queryObject` 方法传入文件 ID，继续调用 `liveShotdDao` 中的 `queiyObject` 方法

来封装查询到的数据，封装为 LiveShotEntity 实体对象，然后返回给控制器，然后调用 deleteShotByld 方法，该方法会将用于标识服务器身份的 SecretIdLSecretKey 以及请求方式等信息封装到一个 map 中，然后实例化对象^[28]，并传进两个参数，第一个参数是需要调用的服务，封装好的配置信息是第二个参数，后与服务器对接将文件 filed 封装进 map 中，QcloudApiModuleCenter 对象的 call 方法被调用，同时传入封装好的请求参数，然后向接口发送请求并返回请求结果。

5.7 本章小结

本章结合概要设计中的结论，用流程图、类图和时序图结合的方式，对直播聚合平台中本人参与设计与实现地模块功能进行详细设计，分别为是：主界面功能模块、直播详情界面模块、直播搜索模块、直播播放模块、用户登录模块、直播聚合模块、违规内容管理模块、评论管理模块，对每个功能点的流程通过流程图做直观展现，对实现功能编码用到的类、主要属性及方法通过类图进行详细介绍，让读者能够清晰了解编码中各个类的调用关系和依赖关系，对功能实现的方法调用步骤描述采用时序图的方式展现，明确每一个方法操作在类中引起转换的触发事件。并且对于其中地多个模块给出了编码实现后的界面截图，使得读者能够更加形象直观地了解本系统的详细设计与实现过程。

6 系统测试

软件测试极其重要，是系统设计与实现过程中不可或缺的步骤，通过这个环节可以发现程序设计中的不足从而及时改正，是保证项目质量的极其重要的手段。在直播聚合平台完成之后，对系统进行相应的测试是必不可少的。本章主要对系统的测试环境、功能性测试、非功能性测试进行介绍^[29]。

6.1 测试环境

直播聚合平台的测试环境有两种，分为服务端测试环境和客户端测试环境。服务端测试环境见表 6-1 所示。

表 6-1 服务端测试环境
Table 6-1 Server Testing Environment

操作系统	Windows Server 10
内存	8GB
CPU	8 核
带宽	5Mbps
数据库环境	Mysql Server 5.6.32
JDK 版本	jdk-8u144-windows-x64
硬盘容量	1TB
端口	8080

客户端测试环境见表 6-2 所示。

表 6-2 客户端测试环境
Table 6-2 Client Testing Environment

操作系统	Android 7.0
型号	Moto Z
版本号	NCC25.106-15
RAM	4GB
ROM	64GB
显示	2560x1440
系统服务	开启系统网络服务

6.2 功能性测试

软件功能性测试就是对产品的各项功能进行验证，根据功能性测试用例逐项测试，检查产品是否达到用户要求的功能。本节针对系统功能设计了典型的测试用例来完成功能性测试，经过运行这些测试用例^[30]，然后将测试结果与已知能够通过的标准进行比较。

表 6-3 为加载主界面功能测试用例，主要验证在用户打开客户端时能否正常显示主界面，通过表 6-3 可以看到加载主界面功能实现状况良好。

表 6-3 加载主界面功能测试用例
Table 6-3 Test Case of Main View Loading

测试用例编号	TC001
测试用例名称	直播应用客户端加载主界面功能测试
目的	测试客户端的请求是否能正常调用后台服务从而加载主界面
优先级	1
前置条件	数据库服务正常，网络连接正常
测试流程	1.启动后台服务平台 2.点击打开客户端 APP
预期结果	客户端在 1s 内播放 launcher 画面然后出现版本更新提示
测试结果	客户端在 1s 内播放 launcher 画面然后出现版本更新提示，数据新增操作日志，服务平台中对应方法正常被调用
结论	测试结果符合预期，测试通过

表 6-4 为直播详情界面功能测试用例，主要验证能够在应用程序内实现直播详情界面正常显示，通过表 6-4 可以看到直播详情界面功能实现状况良好。

表 6-4 直播详情界面功能测试用例
Table 6-4 Test Case of Live Detail View Loading and Operation

测试用例编号	TC002
测试用例名称	直播应用客户端直播详情界面功能测试
目的	测试客户端直播详情界面是否能正常显示以及后台服务是否 正常被调用
优先级	1
前置条件	数据库服务正常，网络连接正常
测试流程	1.启动后台服务平台

表 6-4 (续表)
Table 6-4(Cont.)

测试用例编号	TC002
测试流程	2.打开客户端 APP，点击任意直播节目
预期结果	客户端 1s 内成功加载直播详情页，展示直播内容、主播名字、评论界面以及相关直播列表
测试结果	客户端 1s 内成功加载直播详情页，展示直播内容、主播名字、评论界面以及相关直播列表。查询数据库中浏览记录表新增记录一条，服务平台中相关服务被成功调用
结论	测试结果符合预期，测试通过

表 6-5 直播搜索功能测试用例，主要测试用户能否实现通过主播名字搜索到相关主播的直播间；通过直播平台搜索到该平台所有直播内容、通过直播间 ID 搜索到该直播间以及通过搜索界面提供的搜索热门词进行搜索。通过表 6-5 可以看到直播搜索功能实现状况良好。

表 6-5 直播搜索功能测试用例
Table 6-5 Test Case of Live Searching

测试用例编号	TC003
测试用例名称	直播搜索功能测试用例
目的	判断能否根据不同搜索条件达到用户搜索直播的目的
优先级	1
前置条件	数据库服务正常，网络连接正常
测试流程	1.打开 APP 并登录 2.点击主界面右上角搜索框标识 3.在搜索框中依次输入主播名、直播平台名、主播 ID，以及点击搜索热词进行搜索
预期结果	依次输入主播名、直播平台名、主播 ID，以及点击搜索热词进行搜索均可以正常显示搜索结果列表，且主播名和搜索热词为精确搜索、直播间 ID 与直播平台名称为模糊搜索
测试结果	依次输入主播名、直播平台名、主播 ID，以及点击搜索热词进行搜索均可以正常显示搜索结果列表；且主播名和搜索热词为精确结果、5 分钟后推荐热词会实时更新，后台相关服务正常提供
结论	测试结果符合预期，测试通过

表 6-6 为直播聚合平台直播播放功能的测试用例，主要验证用户通过表 6-6 可以看到该功能点的优化实现状况良好。

表 6-6 直播播放功能测试用例
Table 6-6 Test Case of Live Playback

测试用例编号	TC004
测试用例名称	直播播放功能测试用例
目的	判断能否实现直播正常播放功能以及外延功能
优先级	1
前置条件	数据库服务正常，网络连接正常
测试流程	1.打开 APP 并登录 2.点击主界面任意一个直播间。 3.点击清晰度调节按钮 4.点击录屏按钮 5.点击缩放按钮
预期结果	直播内容正常播放，直播外延功能可以实现调节清晰度、放大缩小直播画面、录屏功能。
测试结果	直播内容正常播放，直播外延功能可以实现调节清晰度、放大缩小直播画面、录屏功能。数据库增加播放记录日志。后台相关服务正常调用。
结论	测试结果符合预期，测试通过

表 6-7 为用户登录功能测试用例，主要验证系统后台能够正常提供登录服，用户能够实现手机号登录与第三方登录功能，通过表 6-7 可以看到用户登录实现状况良好。

表 6-7 用户登录功能测试用例
Table 6-7 Test Case of User Login

测试用例编号	TC005
测试用例名称	用户登录功能测试
目的	验证用户能否实现正常的手机号注册、登录以及使用第三方 QQ、微信登录
优先级	1
前置条件	数据库服务正常，网络连接正常

表 6-7 (续表)
Table 6-7(Cont.)

测试用例编号	TC005
预期结果	用户可以正常实现手机号注册及登录功能，也可以使用第三方例如 QQ、微信进行登录。
测试结果	用户可以正常实现手机号注册及登录功能，也可以使用第三方例如 QQ、微信进行登录。本地数据缓存能够正常保存用户账户信息，后台数据库能够正常更新日志。
结论	测试结果符合预期，测试通过

表 6-8 为直播聚合功能测试用例，主要验证系统后台能否正常提供聚合各家直播平台的功能，通过表 6-8 可以看到直播聚合功能实现状况良好。

表 6-8 直播聚合功能测试用例
Table 6-8 Test Case of User Login

测试用例编号	TC006
测试用例名称	直播聚合功能测试用例
目的	判断能否提供直播聚合功能
优先级	1
前置条件	数据库服务正常，网络连接正常
测试流程	1.打开 APP 并登录 2.点击主界面“栏目” 3.在搜索框中依次输入“虎牙”、“斗鱼”、“龙珠”等直播平台名、进行搜索 4.点击各个不同平台的直播间
预期结果	在同一页面可以展示各个不同平台的直播间，搜索结果可以搜索到各个不同的直播平台，同时各个不同的直播平台的直播均可以正常播放
测试结果	在同一页面可以展示各个不同平台的直播间，搜索结果可以搜索到各个不同的直播平台，同时各个不同的直播平台的直播均可以正常播放；后台相关服务正常提供
结论	测试结果符合预期，测试通过

表 6-9 为评论管理功能测试用例，主要验证管理员能否在系统后台实现对用户评论的管理功能，通过表 6-9 可以看到评论管理功能实现状况良好。

表 6-9 评论管理功能测试用例
Table 6-9 Test Case of Comment Management

测试用例编号	TC007
测试用例名称	评论管理功能测试用例
目的	判断能否实现后台的评论管理功能
优先级	2
前置条件	数据库服务正常，网络连接正常
测试流程	1.管理员登陆后台 2.查看用户在某一个直播间评论 3.删除敏感评论 4.回复用户的评论
预期结果	管理员能够对于直播间内的评论进行检查、删除、回复
测试结果	管理员能够对于直播间内的评论进行检查、删除、回复，后台书数据库实时更新、相关服务正常提供
结论	测试结果符合预期，测试通过

表 6-10 为违规内容管理功能测试用例，主要验证管理员能否在系统后台实现对违规直播间的管理功能，通过表 6-10 可以看到违规内容管理功能实现状况良好。

表 6-10 违规内容管理功能测试用例
Table 6-10 Test Case of Comment Management

测试用例编号	TC008
测试用例名称	违规内容管理功能测试用例
目的	判断能否实现后台的违规内容管理功能
优先级	1
前置条件	数据库服务正常，网络连接正常
测试流程	1.管理员登陆后台 2.查看违规直播间列表信息 3.封禁违规直播间/解除封禁 4.封禁违规账号/解除封禁
预期结果	管理员能够查看违规直播间列表、能够封禁或者解除封禁违规直播间、能够封禁或者解除违规账号
测试结果	管理员能够查看违规直播间列表、能够封禁或者解除封禁违规直播间、能够封禁或者解除违规账号，后台数据库实时更新
结论	测试结果符合预期，测试通过

6.3 非功能性测试

完整的软件需求包括功能需求和非功能需求，非功能需求是不可忽视的一个重要环节，一个软件系统的好坏往往是通过其非功能性需求评判的。本节将对应非功能需求中时效性、可靠性、稳定性和易用性等要求，结合非功能性测试用例来验证整个服务平台设计的合理性。非功能性测试用例如表 6-11 所示。

表 6-11 系统非功能性测试用例
Table 6-11 Test Case of System Non-function

测试用例编号	TC009
测试用例名称	系统非功能性测试
目的	检验系统的和时效性、可靠性、安全性和易用性
优先级	1
前置条件	1.客户端与服务端功能性测试通过 2.客户端与服务端均已启动，网络状况良好
执行时间	2018-09-20 至 2018-11-30
测试流程	1.整个系统连续 10 个工作日运行，不关闭 2.多客户端 (>500 人) 不同时间段访问服务平台，多客户端同一时间段同时访问服务平台 3.对比在客户端和原直播平台打开同一直播的时间 4.记录第一次启动 APP 到出现能够操作地页面间隔时间 5.客户端每隔随机时间切换直播平台进行观看 6.在观看直播过程中随机切换流量与无线网络 7.用户试图登录管理员后台界面
预期结果	在测试用例执行时间内，系统正常运行
测试结果	在测试用例执行时间内，整个系统在 10 个工作日内能不宕机稳定运行；多客户端在不同时间段与同一时间段均可正常访问服务平台；与原直播平台播放时间差小于 1s；APP 首次启动到出现第一个可操作的页面的平均时间间隔为 1.5 秒；客户端随机快速切换不同直播平台进行观看时没有发生闪退现象；在观看直播过程中随机切换不同网络形式不会出现闪退现象，且会及时出现流量预警提示框。普通用户无法登录管理员后台界面，且密码丢失时可以通过绑定手机号找回。
结论	测试结果符合预期，测试通过

6.4 本章小结

本章主要从系统的测试环境、系统功能性和非功能性方面进行分类别测试，详细展示了加载主界面、直播详情界面、直播搜索、直播播放、用户登录、直播聚合、评论管理、违规内容管理的功能性测试用例以及针对系统时效性和易用性的非功能性测试用例，保障系统上线后能够提供高质量的服务。

7 总结与展望

本文通过对当前直播行业的分析确定了所研发项目的方向，从软件生命周期的角度展示了直播聚合平台从设计到开发的全过程，本章将对系统分析、设计和实现的各个阶段做出总结，并结合当今直播行业的发展状况对未来的工作做出展望。

7.1 工作总结

本论文从直播行业的背景出发，分析了直播行业之所以如此火热的原因，阐述了目前直播平台还存在的一些问题且直播聚合平台数量较少以及直播行业在国内外发展的现状，然后对本论文所设计实现的直播聚合平台的意义给予了充分的肯定。在介绍了本系统所使用的技术之后接着对系统的功能性需求和非功能性需求进行了详细地描述，在此基础上确定了系统的设计目标，然后确定了系统的架构、包含系统逻辑架构以及客户端与管理端的技术架构，继而划分了系统主要的功能模块，然后设计了系统的数据库以及缓存和服务器接口，说明了系统的存储方式。在详细设计阶段则借助类图、流程图以及时序图对各个模块的详细设计过程做出了阐述，并配以实现界面截图达到更加形象地展示效果。最后通过功能性测试用例和非功能性测试用例确保了系统上线之后的稳定运行。现对项目的成果总结如下：

完成了直播聚合平台主界面显示模块地设计与实现，包含加载主界面以及直播详情界面展示，能够确保用户有一个良好地界面交互。

完成了直播聚合平台播放模块的设计与实现，能够给用户正常观看直播的功能，并实现了后台播放、录屏等拓展功能，确保满足用户观看直播地需求。

完成了直播聚合平台搜索模块的设计与实现，支持模糊匹配与精确匹配，确保了用户能够在海量数据中完成对直播内容的快速查找，提高用户观看直播的效率。

完成了直播聚合平台直播聚合模块的设计与实现，实现了对不同直播平台的完美兼容，这也是本系统的核心功能，确保了满足用户在一个软件观看多家不同直播的需求。

完成了直播平台用户登录模块的设计与实现，能够给用户正常提供第三方登录或者手机号注册登录功能，在登录之后能够实现收藏、分享、评论等更多交换功能。

完成了直播数据管理模块地设计与实现，包含违规内容管理、评论管理功能、

管理员可以在后台对于违规直播间、违规账号进行处理，确保直播平台风气清正。

目前直播聚合平台一期项目已经通过测试并拥有注册用户上万人，项目中的各个模块均能够按照设计时的要求正常运行。该系统能够提高用户观看直播类节目的体验，能够更好的方便直播受众观看自己喜爱的主播直播，解决主播应用太多导致的多个应用占用内存问题，解决主播跳槽后需要更换直播应用问题。同时该系统能够为公司其他产品进行导流，例如部门游戏应用，同时可以沉淀忠实用户，为以后进行新项目的研发打下基础。

7.2 工作展望

目前除了娱乐类直播与影视类直播以外，还有很多世界顶级的体育赛事以及高端活动在提供直播服务，未来可以加入更多种类的直播类型，让本直播聚合平台达到更加全面的聚合效果。

同时本直播聚合平台目前不支持弹幕消息，因为和直播的几家公司有过版权约定，不会在聚合后实现弹幕功能，同时也将会为几家直播平台导流，希望日后在弹幕功能方面有所突破。

本直播平台目前的违规内容管理功能主要是依靠服务器识别然后交由人工去管理里，在未来的工作中将加入过滤功能，直接由系统实现对不合法、不合规、不健康等内容的过滤。

目前很多国外直播节目纷纷在尝试虚拟现实直播，虚拟现实直播因其独有的特点，往往能令观众有一种身临其境的感觉，能够给观众带来前所未有的极致体验，在今天的春节联欢晚会上，中央电视台也引入了虚拟现实直播，得到了观众的一致好评。随着虚拟现实技术的不断发展，未来虚拟现实设备肯定会像智能手机一样成为人们生活的必需品，而虚拟现实技术因其沉浸性和交互性的特点必将给直播行业带来颠覆性的革命，成为直播平台抢夺的下一个风口。在下阶段的工作中，公司会提前展开在虚拟现实直播领域中的布局，预研相关技术，争取在虚拟现实直播行业抢占更大的市场。

参考文献

- [1] Dickon Ross. The Teardown -The Logitech Revue set-top box for Google TV[J]. Engineering technology, 2011.
- [2] Veljko Ilkic, Marko Kovacevic, Milos Milanovic, Branimir Kovacevic, Petar Jovanovic. One implementation of multimedia content presentation for the Android-based set-top box[J], 2012.
- [3] Darko Dejanovic, Davor Rapic, Nemanja Fimic, Nikola Kuzmanovic, A proposal for VMS extension of Android based DTV STB[J]. Member, IEEE. 2013.
- [4] Jorn De Boever, Dirk De Grooff. Peer-to-Peer Content Distribution and Over-The-Top TV: An Analysis of Value Networks[M]. Handbook of Peer-to-Peer Networking, 2010.
- [5] 陈洁. 网络直播平台: 内容与资本的较量[J]. 视听界, 2016(03):63-67.
- [6] Google. Google TV Home page for development. <http://www.google.com/tv/>, 2010.
- [7] 曲涛, 臧海平. 当前网络直播存在的问题及监管建议[J]. 青年记者, 2016(26):15-17.
- [8] 张受. 热闹的“网红”: 网络直播平台发展中的问题及对策[J]. 中国记者, 2016(05):64-65.
- [9] 喻国明. 从技术逻辑到社交平台: 视频直播新形态的价值探讨[J]. 新闻与写作, 2017(02):51-54.
- [10] Xing Min, Xiang Siyuan, Cai Lin, A Real-Time Adaptive Algorithm for Video Streaming over Multiple Wireless Access Networks[J]. IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, 2014.
- [11] Soskic Nenad, Tapavicki Dejan Popov, Subotic Milos. A Proposal for Second Screen DTV and Remote Control Application [C]. // Friedrich P, Cycon HL, Clauss J. Berlin, GERMANY, 2013.
- [12] Kovacev, Mladen, Miljkovic, Goran, Korac, Bogdan et al. The realization of a digital recorder function for STB based on Android[C]. // 2012 20th Telecommunications Forum. [v.1]. 2012.
- [13] 徐哲桐. 互联网直播行业收入确认和计量研究[D]. 中国财政科学研究院, 2017.
- [14] 靳岩, 姚尚朗. Google Android 开发入门与实战[M]. 北京: 人民邮电出版社, 2009.
- [15] Irene D. Shiny, Dhanalakshmi R. Video Surveillance System And Content Sharing Between PC And Mobile Using Android[C]. // IEEE. Chennai, INDIA. FEB 21-22, 2013.
- [16] Medic Stevan, Spiric Nikola, Tanackovic Sreten. Proposition for Mosaic Video Playback on Android based DTV Devices[C]. // Friedrich P, Cycon HL, Clauss J. Berlin, GERMANY, 2013.
- [17] Jabbar Waheb A., Ismail M., Nordin R. Peer-to-Peer Communication on Android-Based Mobile Devices: Middleware and Protocols[C]. // IEEE. Hammamet, TUNISIA. APR 28-30, 2013.
- [18] 王楠楠, 华中, 寇志强等. 基于 Android 的改进双 Loader 的设计与实现[J]. 电视技术, 2014.
- [19] 金云帆. 爱奇艺游戏直播产品营销推广策划[D]. 浙江大学, 2017. [20] BILL WONG. Will Android Open The Set-Top Box[J]. Electronic design, 2009.
- [20] 杨敏. 互联网电视机顶盒客户端系统的设计与实现[D]. 北京交通大学, 2014.
- [21] Wu Longfe, Du Xiaojiang, Fu Xinwen, Security Threats to Mobile Multimedia Applications: Camera-Based Attacks on Mobile Phones[J]. IEEE COMMUNICATIONS MAGAZINE, 2014.
- [22] Lee Chanhee, Kim Jonghwa, Cho Seong-je, Unified security enhancement framework for the Android operating system[J]. JOURNAL OF SUPERCOMPUTING, 2014.
- [23] (美) 惠腾 (Whitten, J.L.), (美) 本特利 (Bentley, L.D.). 肖刚, 孙慧译. 系统分析

- 与设计方法[M].北京:机械工业出版社,2007-08.
- [24] 魏栋,谭功全,叶建平等.Android系统的内存管理研究[J].单片机与嵌入式系统应用,2012.
- [25] Jeong Jinkyu, Kim Hwanju, Hwang Jeaho, Rigorous Rental Memory Management for Embedded Systems[J], JOURNAL OF SUPERCOMPUTING, 2013.
- [26] 刘洋.基于云计算的企业端直播平台的设计与实现[D].北京交通大学,2018.
- [27] 姜朔昶.浅谈 Android 在数字机顶盒中应用的可行性及其未来发展[J].有线电视技术, 2011(9).
- [28] Wan Pengbo, Jia Haichun, Peng Wenli, Bitmap Vectorization and Realization of Shoes Style[C]. // ISIP, Huangshan, PEOPLES R CHINA, 2009.
- [29] 网易论坛.未来电视机发展趋势. <http://digibbs.tech.163.com/bbs/jiadian/300725939.html>.
- [30] 王蓁蓁.软件测试理论初步框架[J].计算机科学,2014,41(03):12-16+35.

作者简历及攻读硕士学位期间取得的研究成果

刘新奇，男，1995年生。

2017年9月至2019年7月，就读于北京交通大学软件学院，软件工程专业，攻读硕士学位。

2013年9月至2017年7月，就读于太原科技大学计算机学院，软件工程专业，获工学学士学位。

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：刘新奇 签字日期：2019年6月21日

学位论文数据集

表 1.1: 数据集页

关键词*	密级*	中图分类号	UDC	论文资助
直播; 聚合平台; Android;ijkplayer	公开	TP311.5		
学位授予单位名称*	学位授予单位代码*	学位类别*	学位级别*	
北京交通大学	10004	工程	硕士	
论文题名*	并列题名			论文语种*
直播聚合平台的设计与实现				中文
作者姓名*	刘新奇	学号*	17126178	
培养单位名称*	培养单位代码*	培养单位地址	邮编	
北京交通大学	10004	北京市海淀区西直 门外上园村3号	100044	
工程领域*	研究方向*	学制*	学位授予年*	
软件工程	软件工程	两年	2019年	
论文提交日期*	2019年6月			
导师姓名*	刘铎	职称*	副教授	
评阅人	答辩委员会主席*	答辩委员会成员		
	孙卫真			
电子版论文提交格式 文本(<input checked="" type="checkbox"/>) 图像(<input type="checkbox"/>) 视频(<input type="checkbox"/>) 音频(<input type="checkbox"/>) 多媒体(<input type="checkbox"/>) 其他(<input type="checkbox"/>)				
推荐格式: application/msword; application/pdf				
电子版论文出版(发布)者	电子版论文出版(发布)地		权限声明	
论文总页数*	76			
共 33 项, 其中带*为必填数据, 为 21 项。				