哈爾濱工程大學

需求设计报告

项目名称：家庭影音系统设计

学院： 软件学院

专业班级： 20222011

学号姓名（工作 70%）： 2022201111 王一杭

学号姓名（工作 30%）： 2022201120 王泽臣

学号姓名（工作 00%）： N/A

提交日期： 2024/4/23

需求规约

更改履历

序号	版本	更改时间	更改人	更改章节	状态	更改描述
1	V1.0	2024-3-31	王宇华		新建	
2	V1.1	2024-4-2	王一杭	引言&需求	增加	明确基本需求
3	V1.2	2024-4-8	王一杭	需求规约	修改	第二次迭代，明确功能模块
4	V1.3	2024-4-10	王一杭	设计报告	增加	明确基本功能模块
5	V2.0	2024-4-14	王一杭	需求规约	修改	第三次迭代，修改设计理念
6	V2.1	2024-4-17	王一杭	设计报告	修改	重写部分模块结构，符合新版设计，采用总线设计
7	V3.0	2024-4-23	王一杭	ALL	修改	重绘部分类图、添加 ER 图，明确程序内部通信

状态：新建、增加、修改、删除。

目　录

1 引言

1.1 目的

本文档旨在指导开发家庭影音系统，其目标人群为 BT/PT 且具有 NAS（一种私人云盘）的用户，为其提供一个可控制 NAS 下载、播放、流式播放的跨平台软件。

1.2 背景

- 在如今时代，流媒体平台，包括爱奇艺、腾讯视频、bilibili 等国内网站，Youtube、巴哈姆特動畫瘋、niconico、Netflix、Disney+等国外网站（划分由 GFW 控制）普遍具有视频分辨率较低，码率不足，压制过损等现象，因而使对于影音品质有要求的用户流向 BT 下载和 PT 下载。
- NAS 设备及服务提供商，例如群晖、威联通等，提供的下载控件（如 qbittorrent）和流媒体控件（如 jellyfin）多以插件形式安装于 NAS 中，需要利用浏览器访问 NAS 管理页面、应用程序管理页面进行控制，不具有用户权限分级策略，且无高保真输出方式。
- MPV 作为新时代的全格式播放软件，具有较为优越的解码性能，原生的蓝光机播放水准，以及方便的 UI 控制接口，开源的程序代码。

1.3 参考资料

1. Jellyfin Wikipedia(https://zh.wikipedia.org/zh-cn/Jellyfin)
2. Jellyfin 首页(https://jellyfin.org/)
3. MPV 首页(https://mpv.io/)

1.4 术语

1. BT：P2P 下载技术，使用磁力链接和.torrent 文件作为下载链接
2. PT：P2P 下载技术，由于 BT 下载双方过于对等，导致资源丢失和下载方贡献率过低而产生的会员制 P2P 协议
3. NAS：网络附属存储器，提供 SMB 服务或者 FTP 服务的局域网设备，有些服务商提供 HTTP 协议的上传/下载控件。
4. 流式播放：基于局域网上的远程视频资源本地播放
5. 流媒体平台：互联网视频播放平台
6. GFW：中国互联网防火长城，一种似乎存在的东西。
7. qbittorrent：基于 c++开发的开源 P2P 下载软件，提供 WebUI 和 BT 下载程序。
8. jellyfin：基于 C#开发的多媒体应用程序套装，提供 Web 控制和应用程序控制，依靠挂载网络硬盘实现其媒体库。
9. MPV：基于 python 开发的视频解码程序，提供较为宽泛的视频格式支持，提供较为准确的视频解码

2 任务概述

概述：

设计和开发一个家庭影音管理系统，旨在提供一个集中管理家庭媒体内容的解决方案。该系统将允许用户从一个统一的界面中控制各种音频和视频设备，包括电视、音响系统、投影仪等，并能够轻松地访问和管理家庭中的各种媒体内容，如音乐、电影、照片等。

目标：

1. 集成性和兼容性： 确保系统可以控制 HDMI/DP 设备的输出，并提供软件级的识别功能，便于添加新的外部设备。同时提供简单易用的接口，便于外部插件的接入。
2. UI 设计： 设计直观且易于使用的界面，使用户能够轻松地浏览影音库，操作系统功能。并且提供宽泛的设备支持，使得 UI 无论是通过智能手机、平板电脑还是电视屏幕均可以较为直观的被操控。
3. 内容管理和访问： 提供有效的媒体内容管理功能，包括搜索、分类、标记和组织功能，使用户能够快速找到他们想要的媒体内容。
4. 远程访问： 实现远程访问功能，使用户可以通过互联网远程管理和控制家庭娱乐系统。
5. 下载服务： 实现基于 RSS 订阅服务的自动 BT/PT 下载，并提供可视化的正则匹配，使得自动下载内容可以被自动标记、分类、归档。

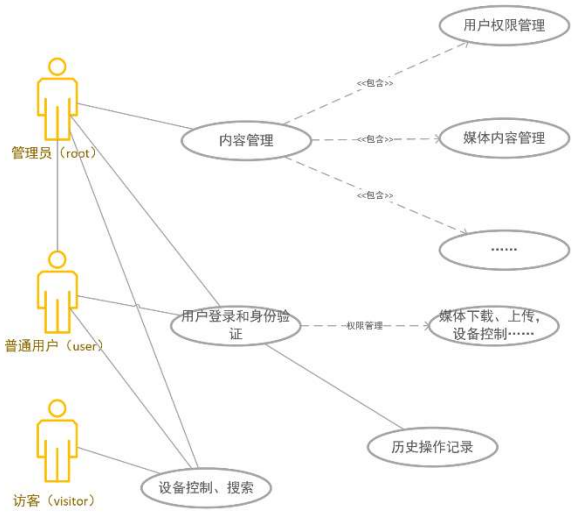
3 需求规定

3.1 一般性需求

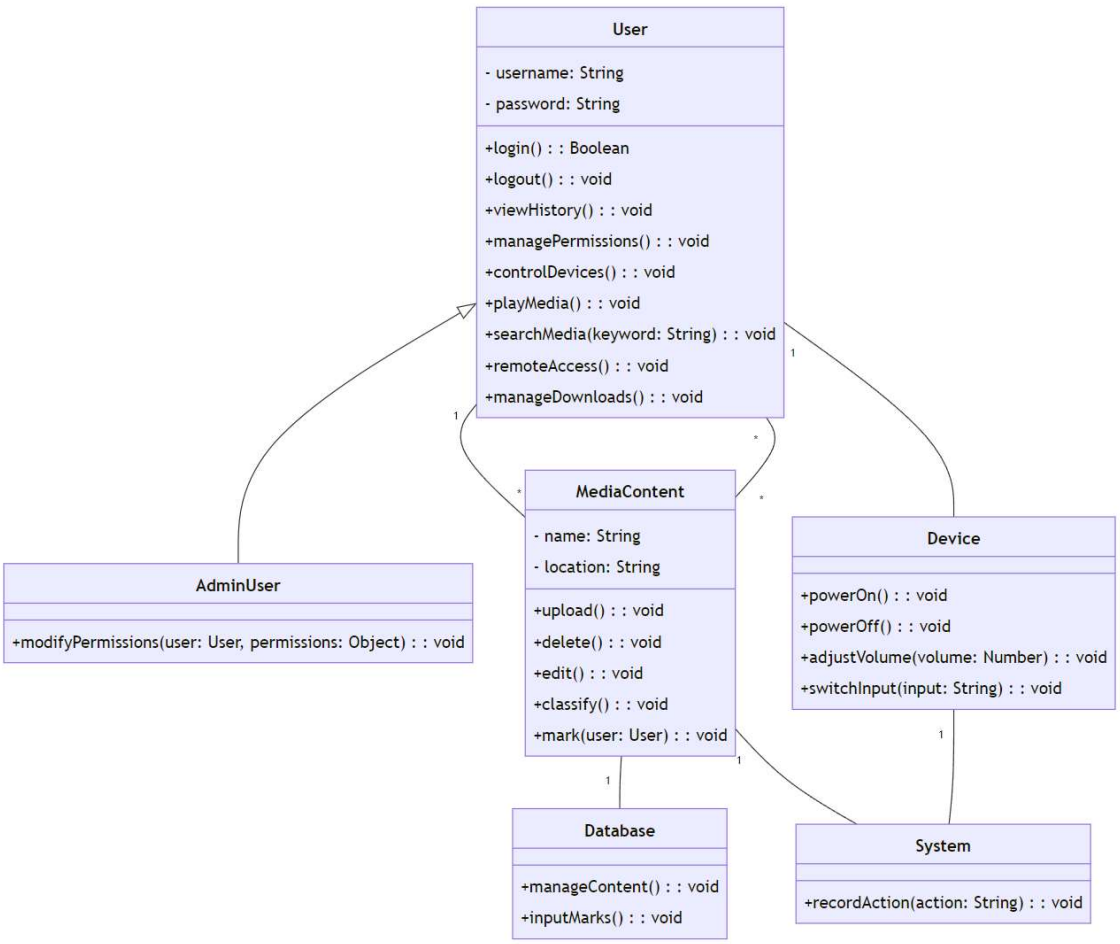
1. 性能需求：本系统响应时间应在 1000ms 以下，吞吐量和并发数不必过高，仅需要处理一般家庭的情况，并发数在 20 以内。但考虑到或许本系统可用于学校等场所使用，并发数或许可进一步提高。
2. 可靠性需求：本系统在出现异常时应记录日志，展式异常页面 10s 并回退至上一页，同时异常页面应当提供回退按钮。
3. 安全性需求：出于本系统的局域网内的应用情况，或许没有必要进行过多的安全性要求。
4. 可维护性需求：本系统应当具有较为清晰的开发文档，代码逻辑应当较为完善的书写在注释中，便于其他开发者基于本系统开发分支。本系统应当以模块的方式进行开发，下载控件、播放控件、管理控件等应当为独立的子模块。
5. 可移植性需求：本系统的服务端应当被部署于 Linux 下，使用 MySQL 作为数据库；本系统的客户端为基于 election 开发的跨平台软件，其数据库也为 MySQL，应当基本适配 Android、Linux、Windows、PlayStation、Xbox 平台。
6. 易用性需求：本系统应当具有且仅具有一个设置页面，用于设置各子模块的配置，本系统应当有 UI 引导按钮，其效果为详细展开/隐藏各按钮的描述信息。
7. 兼容性需求：本系统应当兼容群晖的管理界面，UnRAID 的管理页面，qbittorrent 的 WebUI，Jellyfin 的 WebUI，具体表现为调用其接口进行相关管理。同时本系统应当兼容 FTP 和 SMB 协议，使用其挂载网络硬盘。

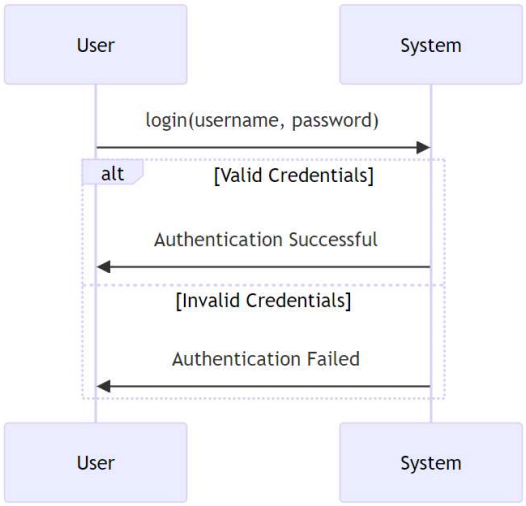
3.2 功能性需求

1. 总用例图

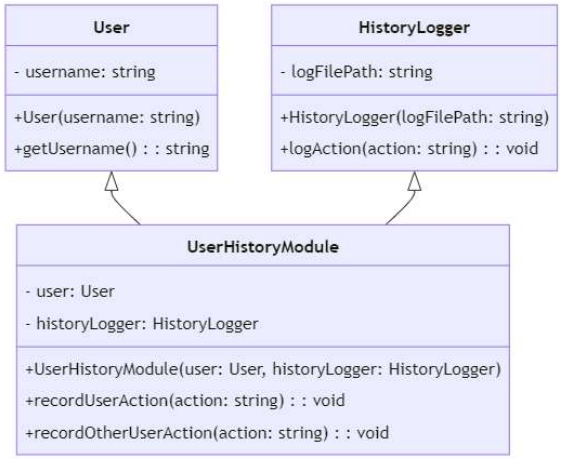


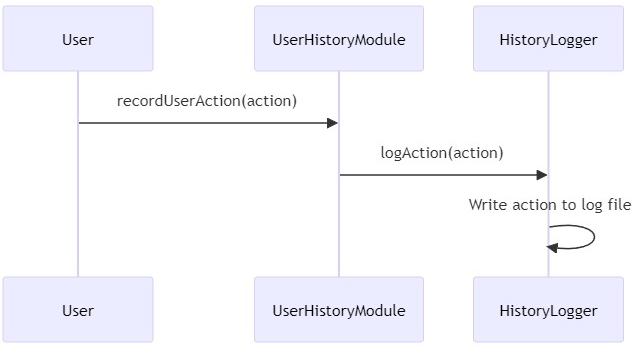
2. 详细需求：
   a) 系统组件
      i. 用户登录模块：本系统应当提供用户登录页面，并区分管理员和一般用户的权限。



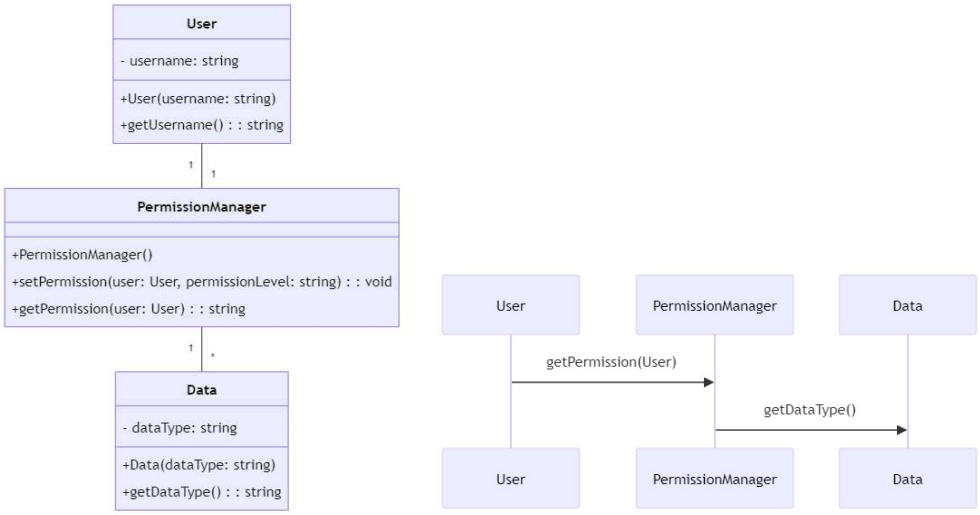


ii. 用户历史记录模块：本系统应当记录用户的操作历史，并将其日志储存至系统端./users/UserName/history/watching_history.log 下。同时也应当记录其他用户可进行的操作的行为的历史。

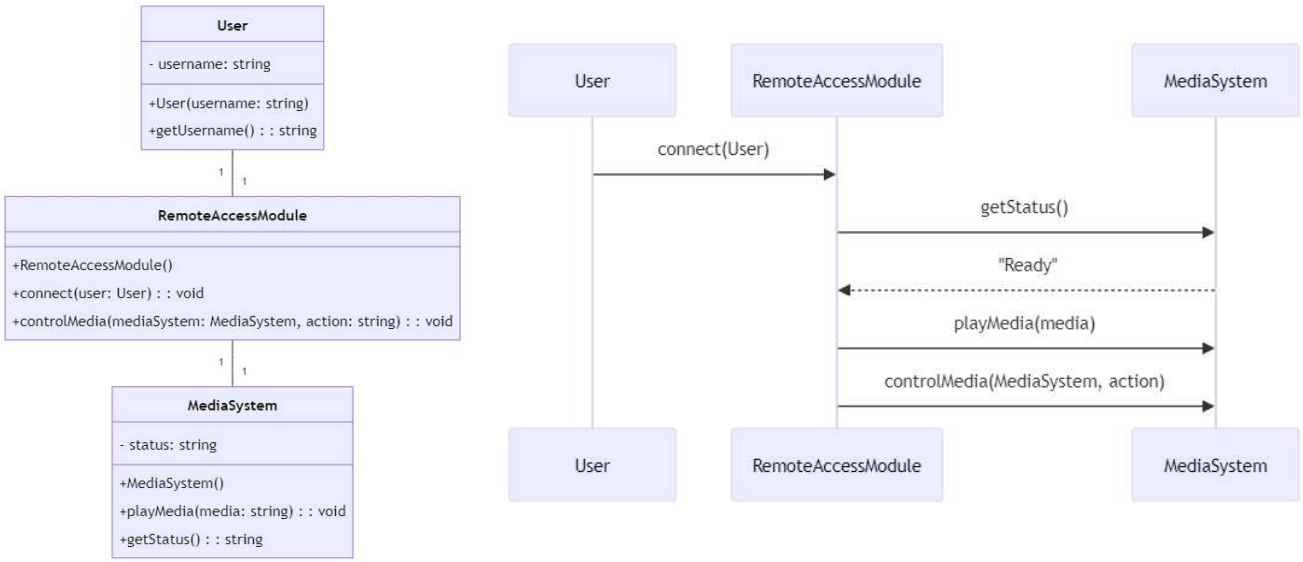




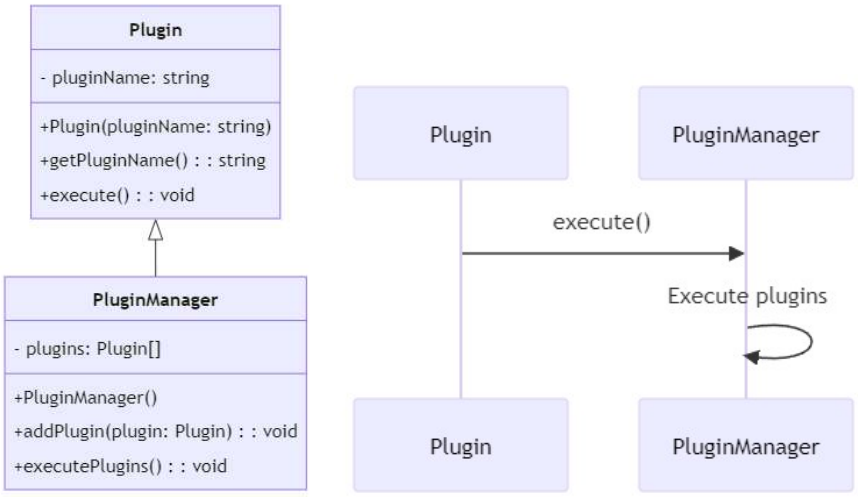
iii. 权限管理模块：本系统应当对用户可接触到的数据进行分级，例如照片的、视频的年龄分级。同时也应当为管理员用户提供修改用户权限的功能。



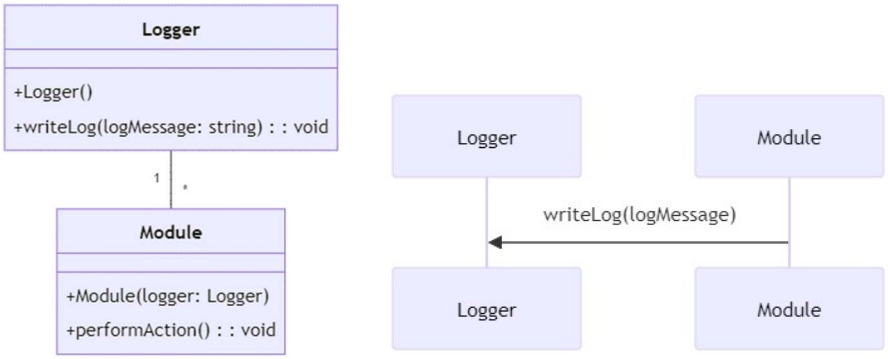
iv. 远程访问模块：用户应该能够通过互联网远程访问和控制家庭影音系统，包括远程播放媒体内容等功能。



v. 插件接入模块: 用于接入用户定义的插件，同时使用此模块书写本系统其他功能模块

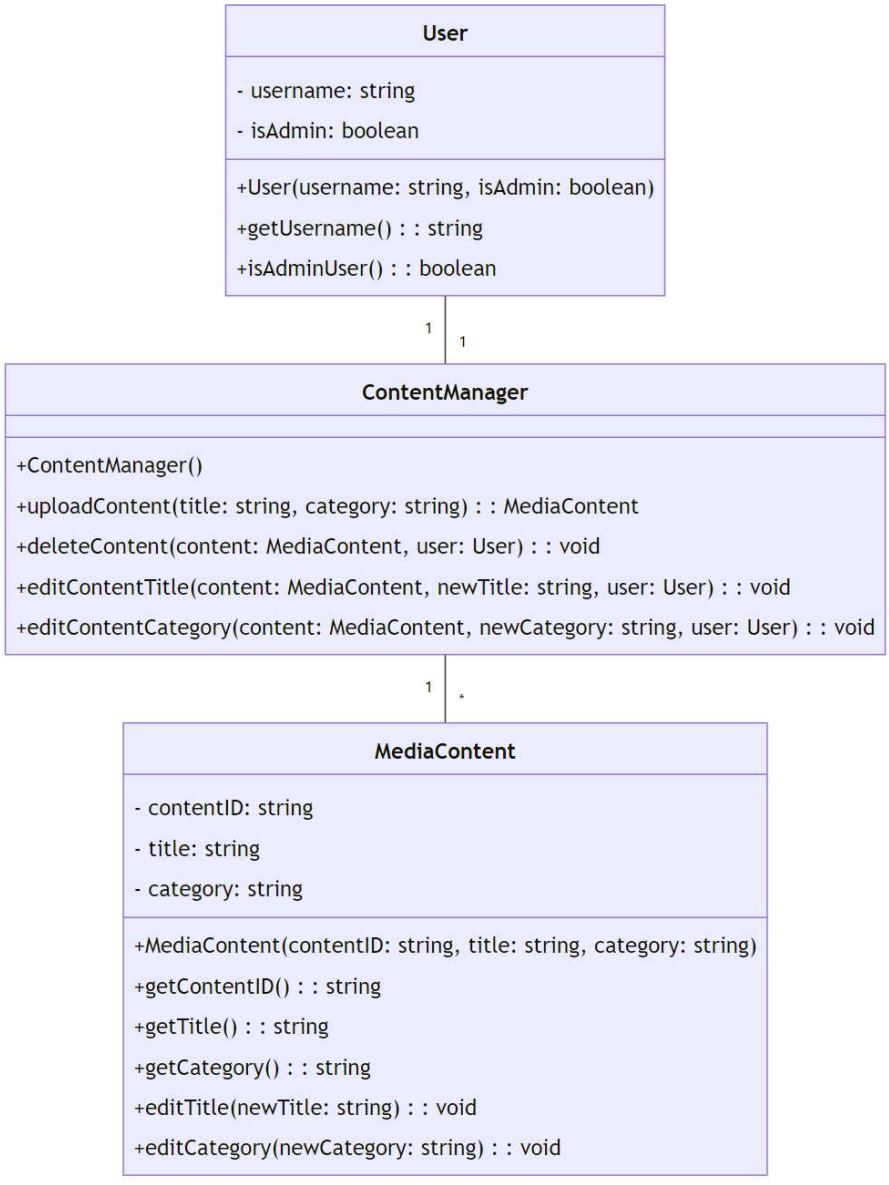


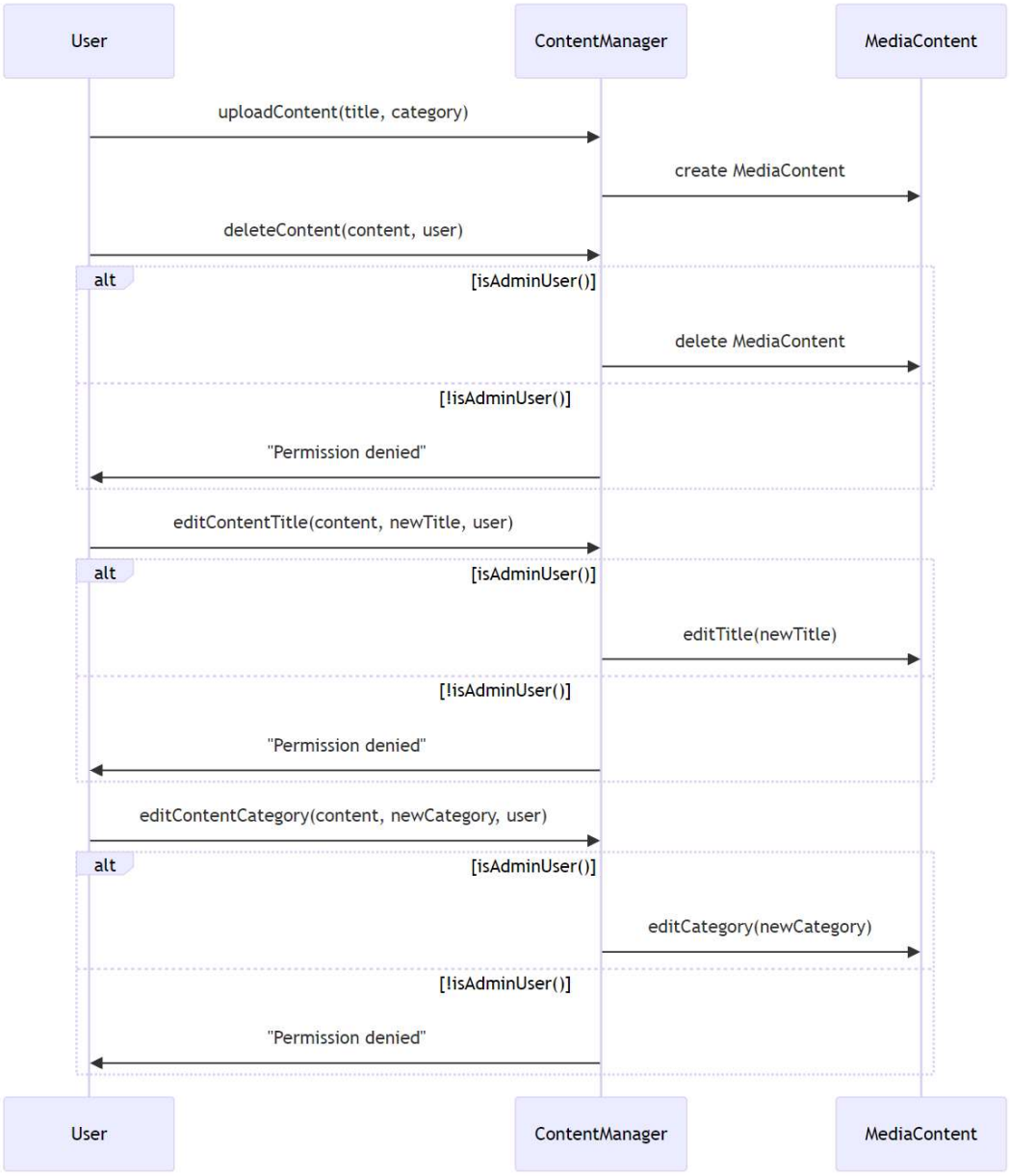
vi. 日志书写模块： 为其它功能模块提供书写日志的接口



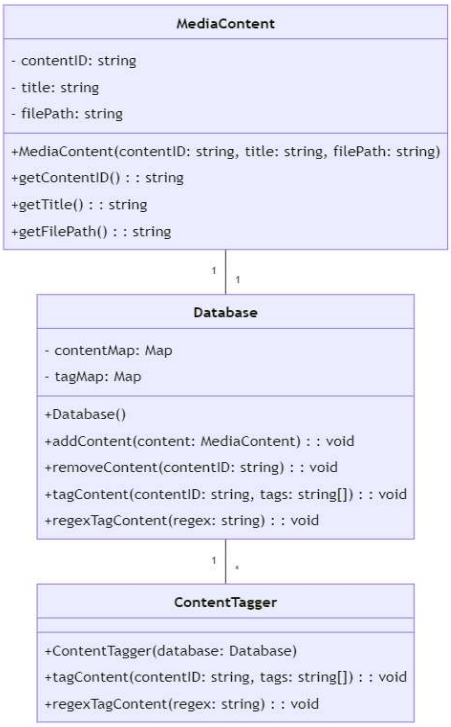
b) 内容组件

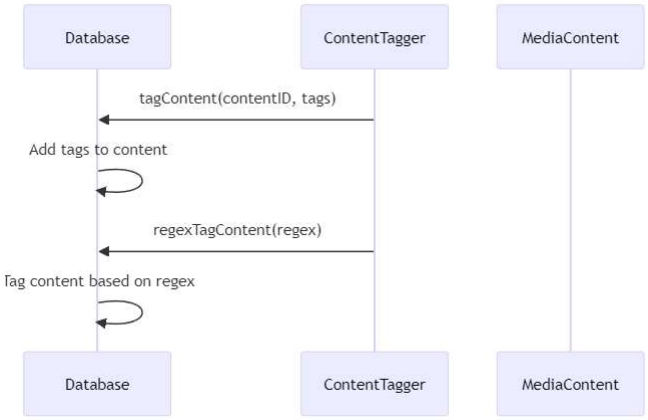
i. 内容管理模块：用户应该能够管理家庭影音系统中的媒体内容，包括上传、删除、编辑、分类和标记等操作。其中，删除操作不对非管理员用户组提供。



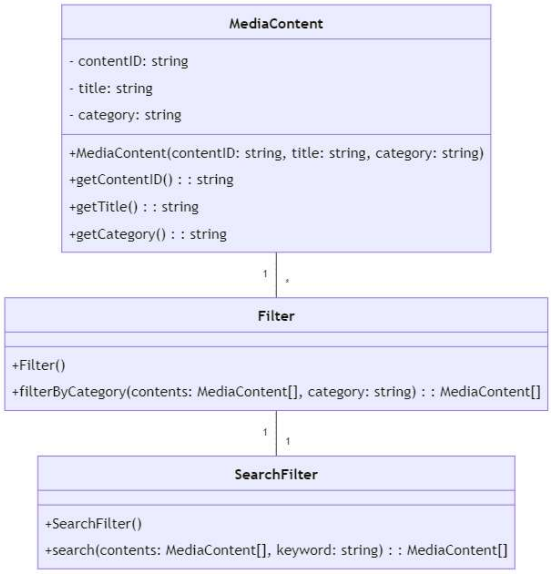


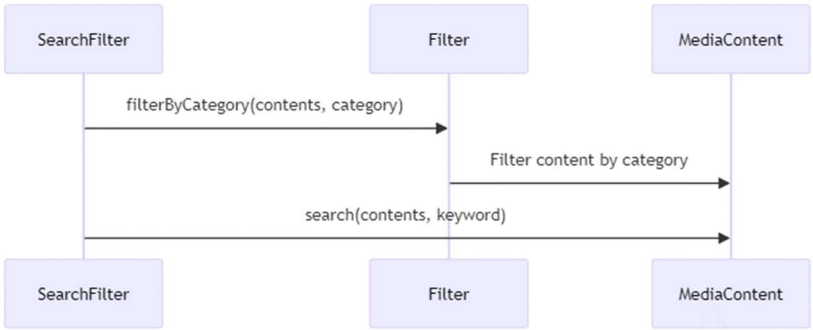
ii. 内容标记模块：媒体内容应当由数据库进行管理，其中键值为名称，对应参数包括媒体内容的绝对位置等，其标记也应当一同输入数据库，可由用户自行标记，也可由正则匹配文件名标记。





iii. 搜索过滤模块：用户应该能够通过关键字搜索媒体内容，并能够根据不同的条件进行过滤和排序，以快速找到所需内容。

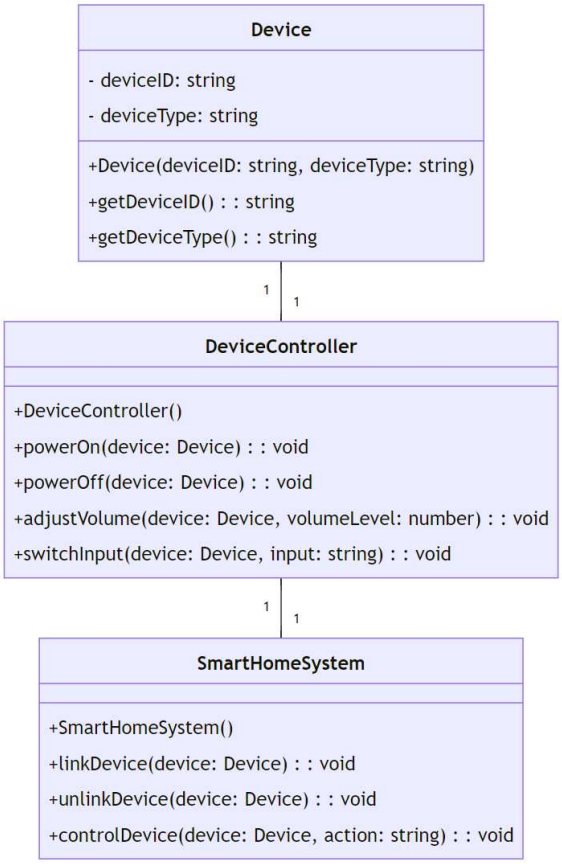


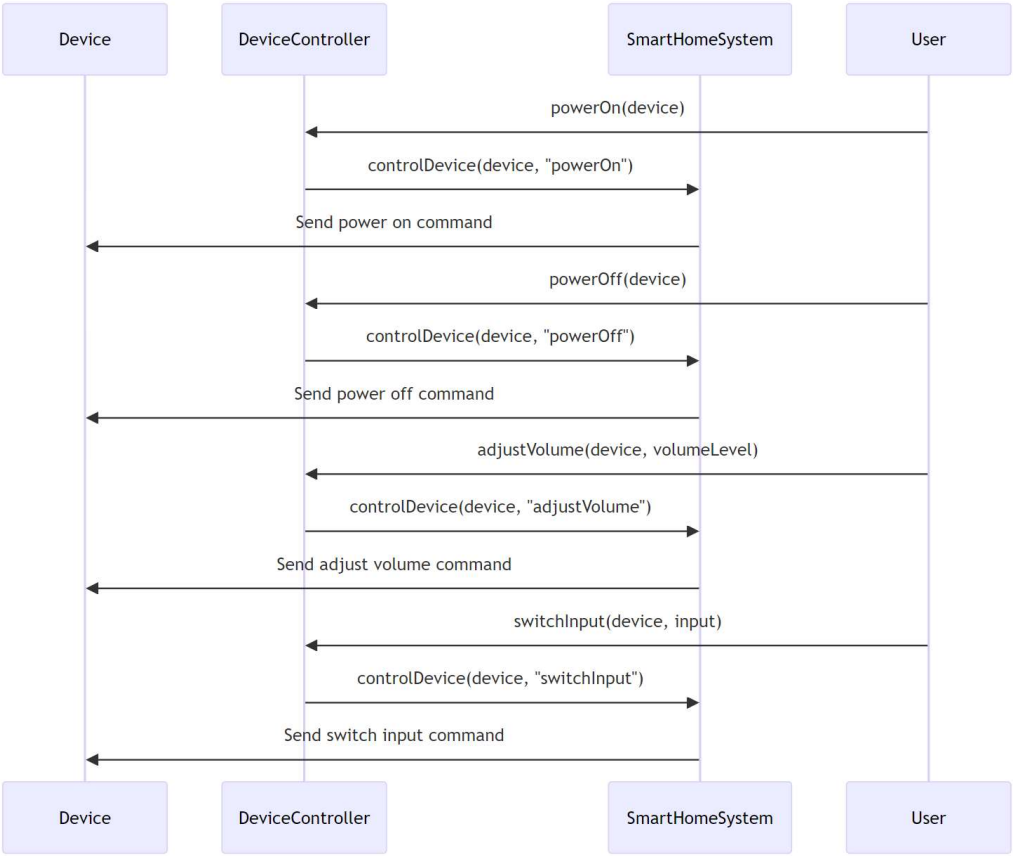


c) 播放组件

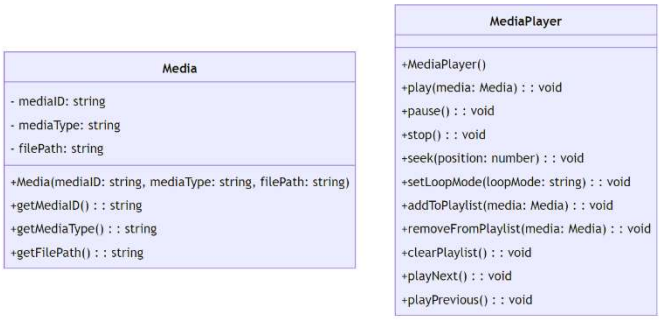
i. 设备控制模块：用户应该能够控制和管理家庭影音系统中的各种设备，如电视、音响系统、投影仪等，包括开关机、音量调节、输入切换等功能。如若可能，应当与智能家居系统联

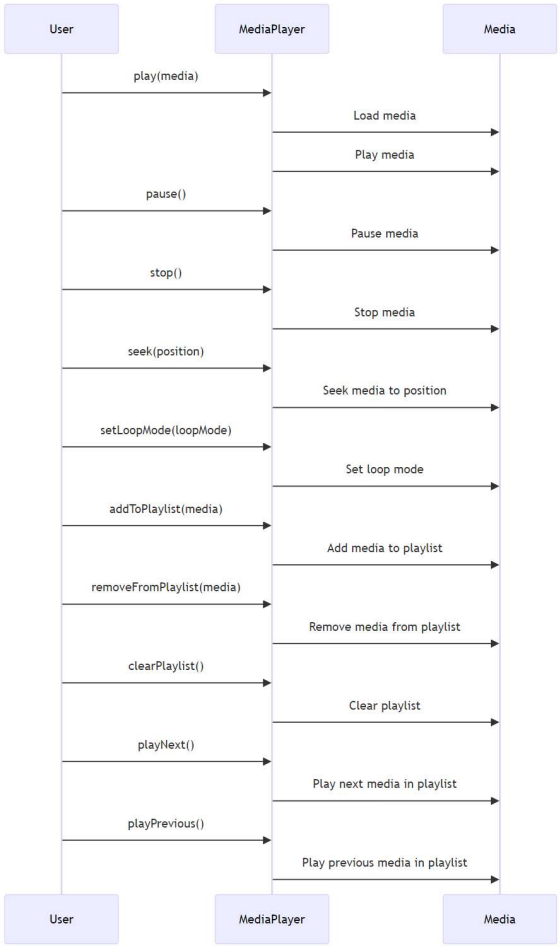
动。





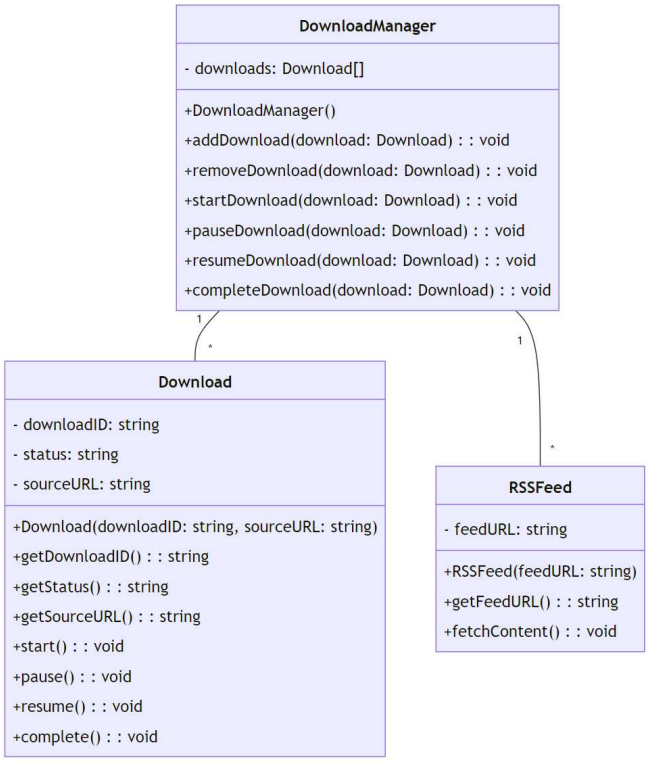
ii. 媒体播放模块：用户应该能够播放各种媒体内容，如音乐、电影、照片等，并能够控制播放进度、循环模式、播放列表等。本系统应当基于 MPV 实现

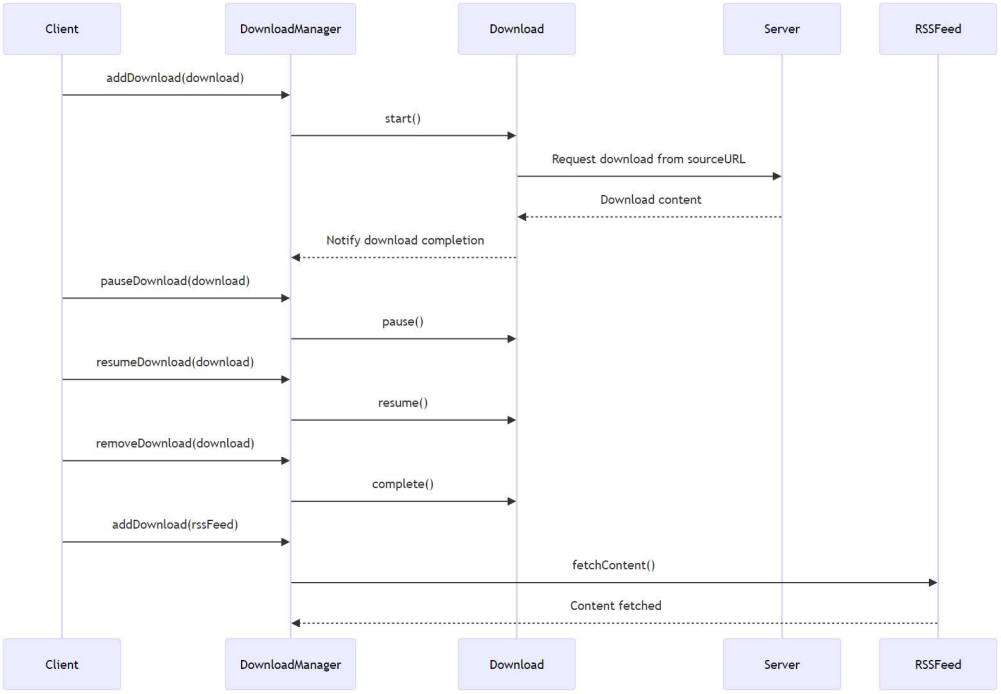




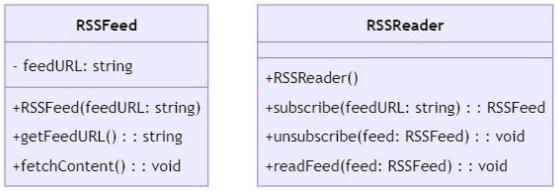
d) 下载组件

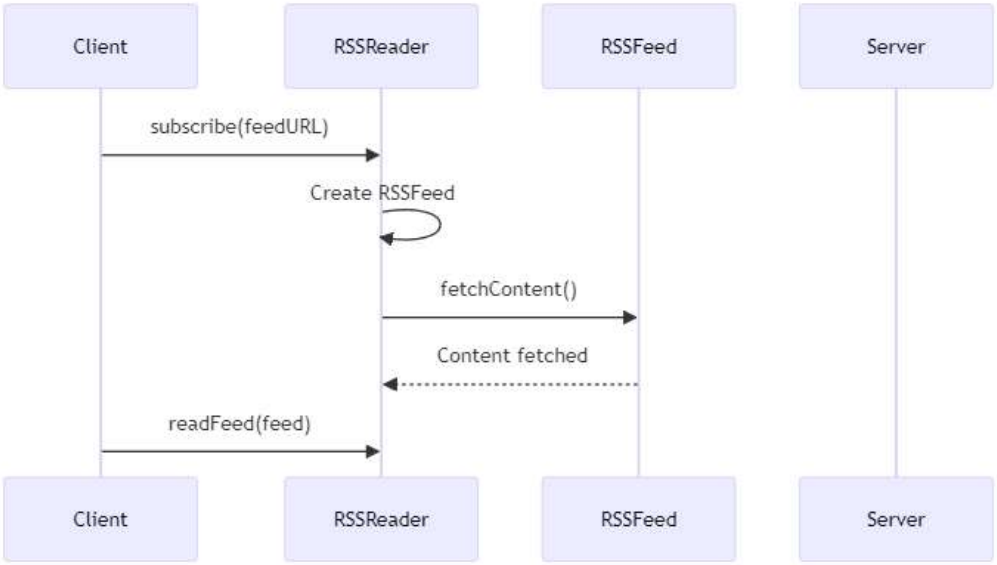
i. 下载管理模块：系统应当提供下载接口，此接口可以由客户端进行调用，将下载内容添加到服务端进行下载；也可以由服务端进行调用，执行 RSS 订阅服务。



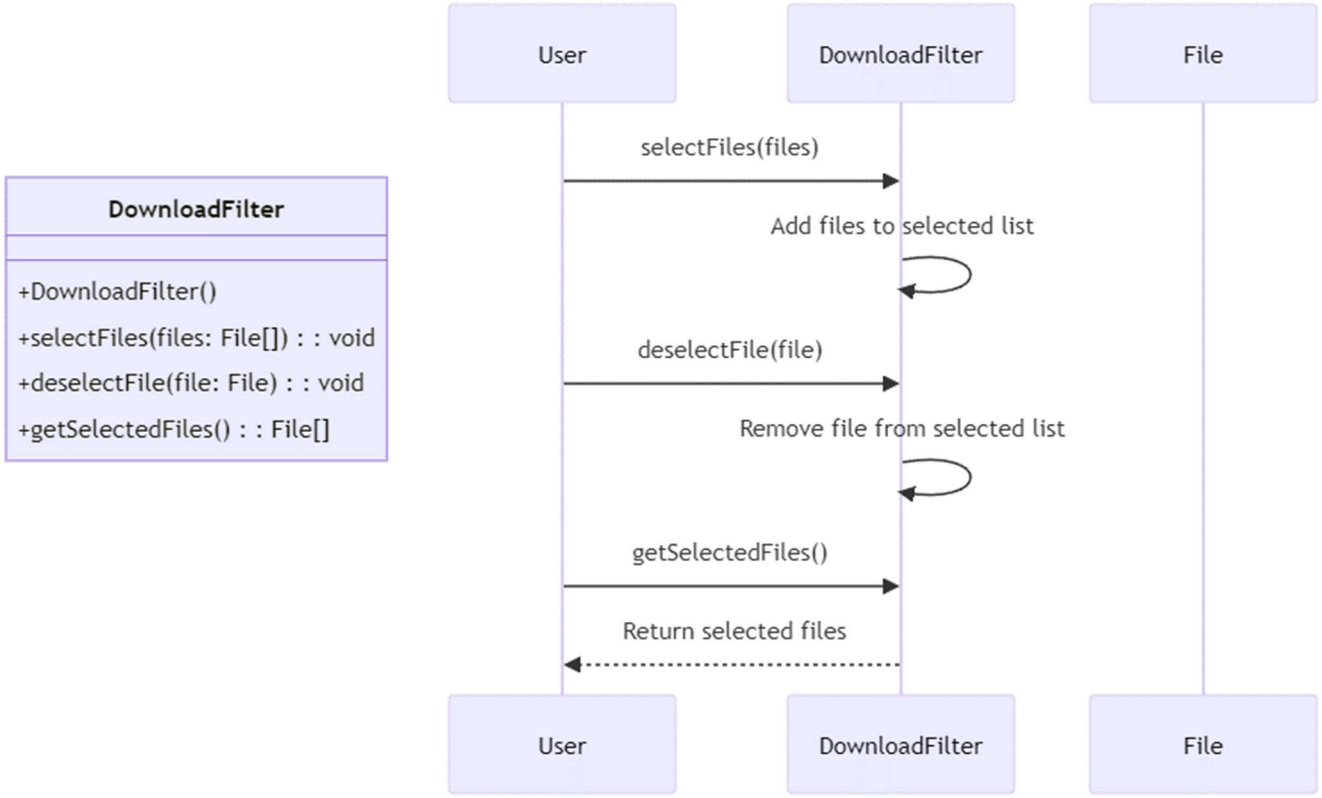


ii. RSS 订阅模块：





iii. 下载筛选模块：本模块由用户调用，提供界面使得用户可以手动批量选择或者单个选择要下载的文件



4 运行环境规定

4.1 服务端

4.1.1 软件环境：

1. 系统：Archlinux
2. 依赖软件：Qt6、Election、docker、MPV、qbittorrent

4.1.2 硬件环境

1. CPU 最低主频：2Ghz
2. 最低运行内存：512MiB
3. 最低储存要求：1GiB

4.2 客户端

4.2.1 软件环境

1. 依赖软件：Election、MPV
2. 运行环境：
   a) 移动端：Android

b) PC 端：Debian/arch/nixos/Windows11/Windows10
c) 主机端：Xbox/PlayStation

4.2.2 硬件环境

1. 最低 CPU 主频：3Ghz
2. 最低显卡要求：
   a) Inter：i3-9 代以上的核显
   b) NAVDIA：GTX1050
   c) AMD：RX5700
3. 最低内存：4GiB
4. 最低储存：16GiB

软件设计报告

目　录

1 引言

1.1 编制目的

本报告详细完成对家庭影音系统的整体设计，达到指导开发的目的。同时实现和测试人员及用户的沟通。

本报告面向详细设计人员、开发人员、测试人员及最终用户编写，是了解系统的导航。

1.2 词汇表

词汇名称	词汇含义	备注
家庭影音系统基本功能	家庭影音系统的基本功能需求，包括视频播放、音频播放、网络硬盘挂载、用户界面设计和接入设备识别	
通信协议	系统中客户端和服务端之间采用的通信协议，如 HTTP 协议	
服务端功能	服务端提供的各项功能，包括媒体文件的下载、存储和管理，网络硬盘的挂载，文件上传，用户登录，展示库存文件目录等	
客户端功能	客户端的功能模块，包括用户交互、媒体文件播放、视频/图片上传等	
技术选型	选择的技术方案，如 election 框架、Java、MySQL 数据库、Docker 等	
用户认证体系	系统采用的用户认证方式，基于相对于 IP 的认证体系	
数据交换方式	客户端和服务端之间数据交换的方式，包括 HTTP 的 get 与 post 请求	
视频处理方式	系统处理视频的方式，包括裁切为 5 秒时长的片段	
数据库结构设计	数据库中媒体文件元数据存储和用户信息存储的设计	
客户端样式设	客户端界面的样式设计，包括登录页面样式、	

计	主界面样式等	
服务端业务逻辑	服务端的各项业务逻辑，如用户身份验证、媒体文件上传和管理、代理网络配置等	
测试和优化	对系统进行全面测试，根据测试结果进行优化以提高系统的性能和用户体验	
部署和维护	将系统部署到目标环境中，并进行定期维护和更新，包括修复 bug、增加新功能等	

1.3 参考资料

1. Jellyfin Wikipedia(https://zh.wikipedia.org/zh-cn/Jellyfin)
2. Jellyfin 首页(https://jellyfin.org/)
3. MPV 首页(https://mpv.io/)
4. mpv.net-DW(https://github.com/diana7127/mpv.net-DW)
5. potplayer(https://potplayer.daum.net/)
6. qbittorrent(https://www.qbittorrent.org/)
7. PT-下载从入门到养老 (https://iecho.cc/2019/01/09/ PT-下载从入门到养老)
8. Plex Media Server(https://www.plex.tv/)

2 系统开发环境

操作系统： Windows11、ArchLinux with hyprlalnd、Chrome
集成开发工具： Vscode、IDEA
编译环境： Python3.90、gcc13.2.0、JDK 11.0.12、Qt6、nodejs
Web 服务器： ArchLinux

3 系统设计思路

1. 需求分析：
   a) 家庭影音系统的基本功能需求应包括视频、音频播放功能，网络硬盘挂载，用户界面设计，接入设备识别，
   b) 服务端和客户端的通信协议可为 HTTP 协议，通过记录局域网 ip-端口访问数据。
   c) 服务端与客户端的数据交换应当通过 HTTP 的 get 与 post 执行，分发内容为通用格式数据包，考虑到家用微型机可能无大量转码能力，因而不通过预转码将视频转换为通用播放格式，而采用视频裁切的方式将视频裁切为 5s 时长的片段，进而载入通用数据包进行传输。
2. 架构设计：
   a) 将系统划分为服务端和客户端两个主要模块，服务端负责提供媒体文件的下载、存储和管理，客户端负责与用户交互和播放媒体文件，同时允许用户上传视频/图片。
   b) 服务端需要设计存储日志文件的数据库结构，以及提供网络硬盘挂载、下载服务接入、文件上传、用户登录、展示库存文件目录的接口。
   c) 客户端需要设计用户界面，包括文件浏览、播放控制等功能。
3. 技术选型：
   a) 客户端选用 election 框架，便于进行跨平台编译，同时 election 的基于 css 的样式可能使用户界面更加美观。
   b) 服务端选用 Java 作为开发语言，使用 MySQL 数据库，考虑到其稳定性需求和低功耗需求或许可使用 docker 作为容器服务装载服务端程序，同时便于服务重启，回档。
4. 服务端设计：
   a) 考虑到局域网服务的特性，服务端采用基于相对于 IP 的用户认证体系，避免使用 token 和 cookie 进行认证。用户信息应当被存储与数据库中，考虑到稳定性和可靠性需求，本数据库应当被冷备份于网络硬盘之中，与服务端隔离。
   b) 服务端提供接口，用于传输各功能分页的 HTML 格式信息
   c) 设计服务端的业务逻辑，包括用户身份验证、媒体文件的上传和管理、网络硬盘的挂载、基于 clash-verge 的代理网络配置等功能。
   d) 设计数据库结构，存储媒体文件的元数据信息，以及用户信息等。
5. 客户端设计：
   a) 登录页面设计：
      i. 样式： 以用户头像为中心的，背景为白色的页面，用户头像捆绑登录按钮。
      ii. 行为：
         1. 登录按钮： 向服务端发送 get 报文，该报文应当提供本机局域网 IP 和子网掩码，用户名称及密码，密码应当采用标准哈希加密，密文传输。接收服务端验证信息。
         2. 登录行为： 向服务端发送 get 报文，请求影视库内容、功能模块，调用主界面绘图函数。
   b) 主界面设计： 视频库页面
      i. 样式： 以图文详情的列表模式展示影视库内容，渲染影视选择按钮，渲染菜单按钮。菜单按钮位置如下

1. Android 平台：渲染于右下角便于单手点击的位置，点击展开为左上 1/4 圆的扇形结构菜单，均分为 4 个选择按钮，依次为
2. Linux 平台/Windows 平台：渲染于左上角，以 M 作为快捷键，选单为列表形式，均提供快捷键。
3. PS／Xbox 平台：打印于右下角按 L1 开启菜单，标准轮盘形式。

ii. 行为：

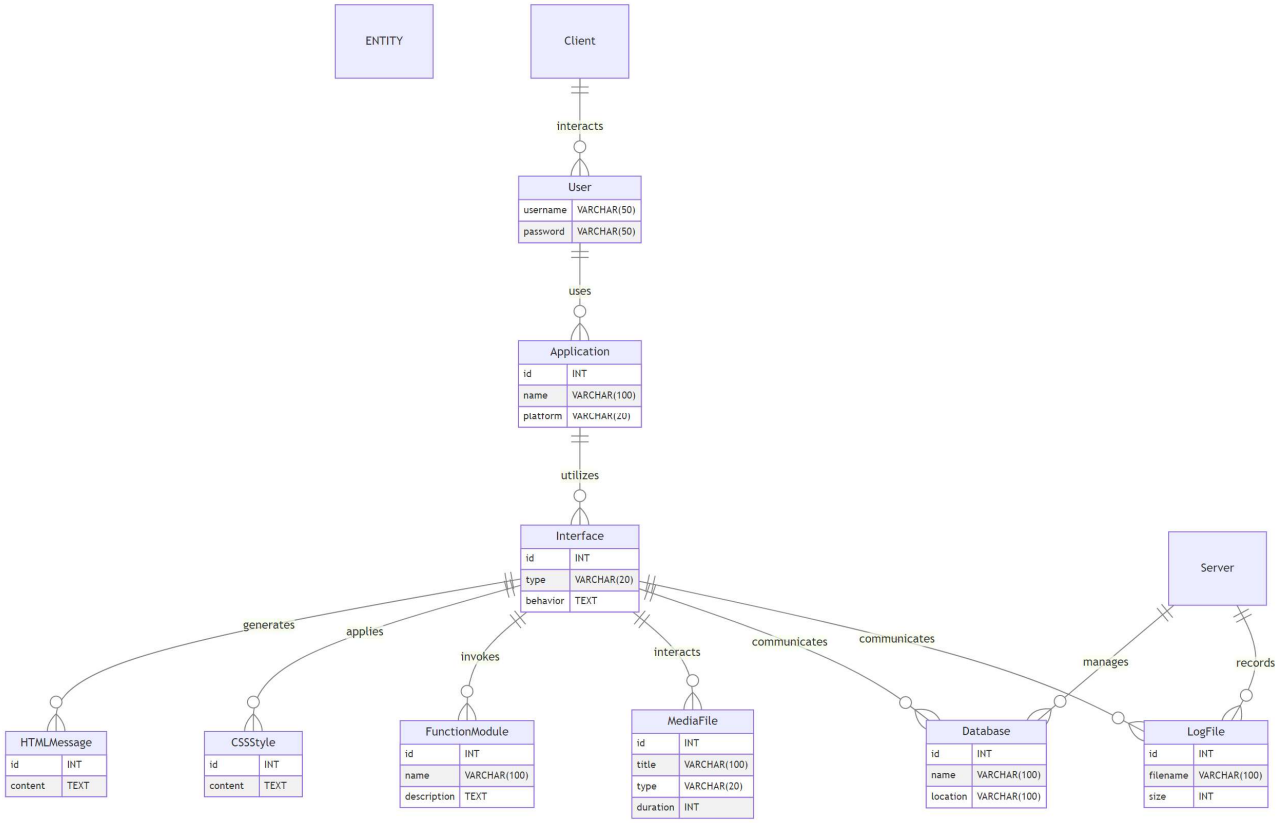
1. 从服务端接受报文，如遇错误返回值应当重试
2. 菜单 bottom 被触发应当按照其行为调用 css 绘图，并渲染 3 个 bottom，其行为为调用各分页组件。
3. 对于视频列表，应当在 bottom 被触动后发送 get 报文，转到视频简介页面。

iii. 附属界面设计：

1. 影音页面：
   a) 资源简介页面：
      i. 目录树
      ii. 音频树
      iii. 视频介绍
      iv. 作品评论（HTML 内嵌页面）
   b) 视频播放界面：
   c) 音乐播放界面：
2. 下载管理页面：
   a) 下载进度页面
   b) RSS 订阅页面
   c) PT 下载管理页面
3. 设置页面：
   a) 功能开关页面
   b) 历史记录/日志查询界面：
   c) 代理配置页面：
   d) 相册展示页面：

6. 测试和优化：
   a) 对系统进行全面的测试，确保各个功能模块的正常运行和稳定性。
   b) 根据测试结果进行优化，提高系统的性能和用户体验。
7. 部署和维护：
   a) 部署系统到目标环境中，确保系统能够正常运行。
   b) 定期对系统进行维护和更新，修复 bug，增加新功能，提升系统的可用性和稳定性。

E-R 图：



4 功能模块设计（核心）

4.1 *下载控件*

4.1.1 功能说明

1. **P2P 下载：**
    i. BT 下载：考虑到部分 NAS 系统存在下载插件，因而提供如下两种构建方式
        a. 服务端捆绑 qbittorrent 软件，侦听 localhost8080 端口，调用 qbittorrent 的 WebUI 接口，下载文件到本地，通过 qbittorrent 的下载文件转移功能，将下载完成后的文件转移到 NAS 上。
        b. 服务端记录搭载 qbittorrent 插件的 NAS 的局域网 IP 位置，侦听该 IP 的特定端口，调用 qbittorrent 的 WebUI 接口，直接在网络硬盘上下载内容。
    ii. PT 下载：PT 下载仍使用 qbittorrent 软件，因而此处考虑其 PT 下载管理方面内容
        a. 由于 pt 下载需要密钥，因而应有密钥管理系统
        b. 考虑可能会接入多个 pt 站的下载内容，因而其下载应当被标记来源
2. **http 下载：** 仍旧考虑调用 NAS 服务与调用本地服务。
    i. 调用 curl 指令，从网络下载内容，上传至 NAS 中。
    ii. 调用 sh 文件，ssh 访问 NAS 后台，调用 curl 指令，下载内容。
3. **RSS 订阅服务：** 为方便管理，应当提供多个 RSS 文件配置方式
    i. 手动审核下载：
        a. 该功能打开时，读取数据库，生成带图列表。

b. 提供 bottom，触发后使得该内容输送至下载模块。

ii. 自动审核机制：

a. 接收订阅推送，执行匹配代码，匹配成功则推送至下载模块，失败则推送至数据库

b. 接受订阅管理模块的匹配语句，解释为代码。

iii. RSS 订阅管理：

a. 读取 RSS 订阅推送的内容，输送至自动审核模块

b. 管理 RSS 订阅代码文件

c. 提供各 RSS 订阅文件的自动匹配规则编写，输送至自动审核模块。

4.1.2 类、方法设计

4.1.2.1DownloadControl 类

该类的功能：控制下载功能。

返回值	方法名	功能	参数说明
startP2PDownload	启动 P2P 下载	源地址、目标路径	
startHTTPDownload	启动 HTTP 下载	源地址、目标路径	
startRSSDownload	启动 RSS 订阅下载	订阅源 URL、目标路径	

4.1.2.2 PTDownloadManager 类

该类的功能：控制 PT 下载功能。

返回值	方法名	功能	参数说明
managePTDownload	管理 PT 下载	PT 站点、下载内容、密钥	

4.1.2.3 RSSManager 类

该类的功能：RSS 订阅功能。

返回值	方法名	功能	参数说明
	manualAudit	手动审核 RSS 内容	RSS 源 URL
	automaticAudit	自动审核 RSS 内容	RSS 源 URL
	manageSubscriptions	管理 RSS 订阅	订阅源 URL、匹配规则
	matchCompilation	预编译正则匹配	匹配规则

4.1.2.4 DataStorage 类

该类的功能：下载记录存储功能。

返回值	方法名	功能	参数说明
saveDownloadRecord	保存下载记录	下载信息	
retrieveDownloadRecord	检索下载记录	下载条件	

4.1.2.5 PTKeyManager 类

该类的功能：PT 密钥管理功能。

返回值	方法名	功能	参数说明
	manageKeys	管理 PT 下载密钥	密钥信息

4.1.3 相关数据表

下载记录表
PT 下载密钥表
RSS 订阅管理表

4.1.4 接口设计

1. **startP2PDownload**
    - **功能：** 启动 P2P 下载任务。
    - **HTTP 方法：** POST
    - **URL 路径：** /p2p/download
    - **请求参数：**
        - source_url (string): 下载源地址
        - destination_path (string): 下载文件存储路径
    - **响应：**
        - 成功：HTTP 状态码 200，返回下载任务 ID 和状态信息。
        - 失败：HTTP 状态码 400，返回错误信息。
2. **startHTTPDownload**
    - **功能：** 启动 HTTP 下载任务。
    - **HTTP 方法：** POST
    - **URL 路径：** /http/download
    - **请求参数：**
        - source_url (string): 下载源地址
        - destination_path (string): 下载文件存储路径
    - **响应：**
        - 成功：HTTP 状态码 200，返回下载任务 ID 和状态信息。
        - 失败：HTTP 状态码 400，返回错误信息。
3. **startRSSDownload**
    - **功能：** 启动 RSS 订阅下载任务。
    - **HTTP 方法：** POST
    - **URL 路径：** /rss/download
    - **请求参数：**
        - feed_url (string): RSS 订阅源地址
        - destination_path (string): 下载文件存储路径
    - **响应：**
        - 成功：HTTP 状态码 200，返回下载任务 ID 和状态信息。
        - 失败：HTTP 状态码 400，返回错误信息。
4. **managePTDownload**
    - **功能：** 管理 PT 下载任务。
    - **HTTP 方法：** POST
    - **URL 路径：** /pt/download
    - **请求参数：**
        - site_name (string): PT 站点名称
        - download_content (string): 下载内容
        - key (string): 密钥
    - **响应：**

- 成功：HTTP 状态码 200，返回下载任务 ID 和状态信息。
- 失败：HTTP 状态码 400，返回错误信息。

5. **manualAudit**
   - **功能：** 手动审核 RSS 内容。
   - **HTTP 方法：** POST
   - **URL 路径：** /rss/manual_audit
   - **请求参数：**
     - feed_url (string): RSS 订阅源地址
   - **响应：**
     - 成功：HTTP 状态码 200，返回审核结果信息。
     - 失败：HTTP 状态码 400，返回错误信息。
6. **automaticAudit**
   - **功能：** 自动审核 RSS 内容。
   - **HTTP 方法：** POST
   - **URL 路径：** /rss/automatic_audit
   - **请求参数：**
     - feed_url (string): RSS 订阅源地址
   - **响应：**
     - 成功：HTTP 状态码 200，返回审核结果信息。
     - 失败：HTTP 状态码 400，返回错误信息。
7. **manageSubscriptions**
   - **功能：** 管理 RSS 订阅。
   - **HTTP 方法：** POST
   - **URL 路径：** /rss/manage_subscriptions
   - **请求参数：**
     - feed_url (string): RSS 订阅源地址
     - matching_rules (string): 匹配规则
   - **响应：**
     - 成功：HTTP 状态码 200，返回操作结果信息。
     - 失败：HTTP 状态码 400，返回错误信息。

4.2 播放控件

4.2.1 功能说明

1. 播放组件
   i. 调用 mpv 程序，解码并渲染视频及音频。
2. 视频播放控制组件
   i. 基于 election 的控件，与播放组件交互，控制其行为，控制本地播放。
   ii. Android 端提供遥控器页面，操作 PS/Xbox 端播放组件（如果在线）。
   iii. 渲染应当与 mpv-net-DW 分支界面类似。
3. 视频列表展示组件
   i. 应当提供可隐藏的半透明播放列表。
   ii. 调用硬盘挂载所提供的接口，或调用数据库接口，以得到目录树结构。
4. 播放日志记录功能
   i. 记录视频播放历史，包括播放时间、播放内容等信息。

4.2.2 类、方法设计

4.2.2.1 VideoPlayer 类

该类的功能：用于控制视频播放功能。

返回值	方法名	功能	参数说明
	play()	播放视频	
	stop()	停止播放	

4.2.2.2 VideoControlComponent 类

该类的功能：该类用于控制视频播放行为。

返回值	方法名	功能	参数说明
	controlPlay()	控制视频播放行为	
	controlPause()	控制视频暂停	
	controlStop()	控制视频停止	

4.2.2.3 VideoListDisplayComponent 类

该类的功能：该类用于展示视频列表。

返回值	方法名	功能	参数说明
	showList()	展示视频列表	
	hideList()	隐藏视频列表	

4.2.2.4PlayLogRecorder 类

该类的功能：该类用于记录视频播放日志。

返回值	方法名	功能	参数说明
recordLog()	记录播放日志	播放内容、播放时间等信息	

4.2.3 相关数据表

播放日志表

4.2.4 接口设计

1. **playVideo**
   - **功能：** 播放视频。
   - **HTTP 方法：** POST
   - **URL 路径：** /video/play
   - **请求参数：**
     - video_id (string): 视频 ID
   - **响应：**
     - 成功：HTTP 状态码 200，返回成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。
2. **stopVideo**
   - **功能：** 停止视频播放。
   - **HTTP 方法：** POST
   - **URL 路径：** /video/stop

- **请求参数：**
  - video_id (string): 视频 ID
- **响应：**
  - 成功：HTTP 状态码 200，返回成功信息。
  - 失败：HTTP 状态码 400，返回错误信息。

3. **pauseVideo**
   - **功能：** 暂停视频播放。
   - **HTTP 方法：** POST
   - **URL 路径：** /video/pause
   - **请求参数：**
     - video_id (string): 视频 ID
   - **响应：**
     - 成功：HTTP 状态码 200，返回成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。
4. **showVideoList**
   - **功能：** 展示视频列表。
   - **HTTP 方法：** GET
   - **URL 路径：** /video/list
   - **请求参数：** 无
   - **响应：**
     - 成功：HTTP 状态码 200，返回视频列表信息。
     - 失败：HTTP 状态码 400，返回错误信息。
5. **hideVideoList**
   - **功能：** 隐藏视频列表。
   - **HTTP 方法：** POST
   - **URL 路径：** /video/list/hide
   - **请求参数：** 无
   - **响应：**
     - 成功：HTTP 状态码 200，返回成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。
6. **recordPlayLog**
   - **功能：** 记录视频播放日志。
   - **HTTP 方法：** POST
   - **URL 路径：** /video/log/record
   - **请求参数：**
     - video_id (string): 视频 ID
     - play_time (string): 播放时间
   - **响应：**
     - 成功：HTTP 状态码 200，返回成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。

4.3 代理配置控件

4.3.1 功能说明

提供标准的 mihono（clash-verge）内核的代理组件，进而提供网络代理功能，便于访问通过 DDNS 技术或者 VPN 技术访问局域网。

1. 控制组件
    i. 提供 yaml 配置文件读取操作，从网络硬盘读取或调用下载组件从网址下载。
    ii. 提供较为丰富的控制组件，包括选择节点、tun 模式等等。
2. 自启动组件

4.3.2 类、方法设计

4.3.2.1 ProxyConfig 类

该类的功能：该类用于管理代理配置相关的操作。。

返回值	方法名	功能	参数说明
read_config_from_disk	从本地磁盘读取代理配置	文件路径：代理配置文件的路径	
download_config_from_url	从指定网址下载代理配置	URL：代理配置文件的下载链接	
select_node	选择特定节点	节点名称：要连接的节点的名称	
set_tun_mode	设置 TUN 模式	模式：TUN 模式的设置参数	

4.3.2.2 AutoStartComponent 类

该类的功能：该类用于控制代理自启动功能。。

返回值	方法名	功能	参数说明
	enable_auto_start	启用代理自启动	无
	disable_auto_start	禁用代理自启动	无

4.3.3 相关数据表

ProxyConfig 表：该数据表存储代理配置相关的信息。

4.3.4 接口设计

1. **readProxyConfigFromDisk**
    - **功能：** 从本地磁盘读取代理配置文件。
    - **HTTP 方法：** POST
    - **URL 路径：** /proxy/config/read
    - **请求参数：**
        - file_path (string): 代理配置文件的路径
    - **响应：**
        - 成功：HTTP 状态码 200，返回成功信息。
        - 失败：HTTP 状态码 400，返回错误信息。
2. **downloadProxyConfigFromURL**
    - **功能：** 从指定网址下载代理配置文件。
    - **HTTP 方法：** POST
    - **URL 路径：** /proxy/config/download
    - **请求参数：**
        - url (string): 代理配置文件的下载链接
    - **响应：**
        - 成功：HTTP 状态码 200，返回成功信息。

    - 失败：HTTP 状态码 400，返回错误信息。
3. **selectProxyNode**
    - **功能：** 选择特定节点。
    - **HTTP 方法：** POST
    - **URL 路径：** /proxy/node/select
    - **请求参数：**
        - node_name (string): 要连接的节点的名称
    - **响应：**
        - 成功：HTTP 状态码 200，返回成功信息。
        - 失败：HTTP 状态码 400，返回错误信息。
4. **setTUNMode**
    - **功能：** 设置 TUN 模式。
    - **HTTP 方法：** POST
    - **URL 路径：** /proxy/mode/set
    - **请求参数：**
        - mode (string): TUN 模式的设置参数
    - **响应：**
        - 成功：HTTP 状态码 200，返回成功信息。
        - 失败：HTTP 状态码 400，返回错误信息。
5. **enableAutoStart**
    - **功能：** 启用代理自启动。
    - **HTTP 方法：** POST
    - **URL 路径：** /proxy/autostart/enable
    - **请求参数：** 无
    - **响应：**
        - 成功：HTTP 状态码 200，返回成功信息。
        - 失败：HTTP 状态码 400，返回错误信息。
6. **disableAutoStart**
    - **功能：** 禁用代理自启动。
    - **HTTP 方法：** POST
    - **URL 路径：** /proxy/autostart/disable
    - **请求参数：** 无
    - **响应：**
        - 成功：HTTP 状态码 200，返回成功信息。
        - 失败：HTTP 状态码 400，返回错误信息。

4.4 硬盘挂载控件

4.4.1 功能说明

1. 提供网络硬盘挂载功能，支持 ftp、SMB 等常用协议。
2. 支持上传、下载、只读等功能。
3. 将操作、硬盘目录等记录于数据库中。

4.4.2 类、方法设计

4.4.2.1 DiskMountControl 类

该类的功能：用于控制硬盘挂载功能。

返回值	方法名	功能	参数说明
	mountDisk	挂载硬盘	硬盘地址、协议类型、用户名、密码、目标路径
	unmountDisk	卸载硬盘	目标路径
	setPermissions	设置硬盘权限	目标路径、权限设置

4.4.2.2 DiskOperationLogger 类

该类的功能：用于记录硬盘操作日志。

返回值	方法名	功能	参数说明
	logOperation	记录硬盘操作	操作类型、目标路径、操作时间
	retrieveLogs	检索操作日志	操作条件、时间范围

4.4.3 相关数据表

硬盘操作日志表

4.4.4 接口设计

1. **mountDisk**
   - **功能：** 挂载硬盘。
   - **HTTP 方法：** POST
   - **URL 路径：** /disk/mount
   - **请求参数：**
     - disk_address (string): 硬盘地址
     - protocol (string): 挂载协议类型（ftp、SMB 等）
     - username (string): 用户名
     - password (string): 密码
     - target_path (string): 挂载目标路径
   - **响应：**
     - 成功：HTTP 状态码 200，返回挂载成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。
2. **unmountDisk**
   - **功能：** 卸载硬盘。
   - **HTTP 方法：** POST
   - **URL 路径：** /disk/unmount
   - **请求参数：**
     - target_path (string): 卸载目标路径
   - **响应：**
     - 成功：HTTP 状态码 200，返回卸载成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。
3. **setPermissions**
   - **功能：** 设置硬盘权限。
   - **HTTP 方法：** POST
   - **URL 路径：** /disk/permissions/set
   - **请求参数：**
     - target_path (string): 目标路径

- permissions (string): 权限设置（只读、读写等）
- **响应：**
  - 成功：HTTP 状态码 200，返回权限设置成功信息。
  - 失败：HTTP 状态码 400，返回错误信息。

4. **logOperation**
   - **功能：** 记录硬盘操作日志。
   - **HTTP 方法：** POST
   - **URL 路径：** /disk/log
   - **请求参数：**
     - operation_type (string): 操作类型（挂载、卸载等）
     - target_path (string): 目标路径
     - operation_time (string): 操作时间
   - **响应：**
     - 成功：HTTP 状态码 200，返回日志记录成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。
5. **retrieveLogs**
   - **功能：** 检索硬盘操作日志。
   - **HTTP 方法：** GET
   - **URL 路径：** /disk/logs
   - **请求参数：**
     - operation_type (string): 操作类型（可选）
     - time_range (string): 时间范围（可选）
   - **响应：**
     - 成功：HTTP 状态码 200，返回操作日志信息。
     - 失败：HTTP 状态码 400，返回错误信息。

4.5 *日志书写控件*

4.5.1 功能说明

1. 提供与数据库交互的接口，记录数据。
2. 捕捉其他控件出现的异常，记录于日志。
3. 接收其他控件输出的日志。

4.5.2 类、方法设计

4.5.2.1 LogWriter 类

该类的功能：记录日志。

返回值	方法名	功能	参数说明
	writeToDatabase	将日志写入数据库	日志内容
	logException	记录异常信息	异常对象
	receiveLog	接收其他控件输出的日志	日志内容

4.5.2.2 LogDatabaseInterface 类

该类的功能：用于与数据库交互，实现日志的持久化存储。

返回值	方法名	功能	参数说明
	saveLog	将日志保存到数据库	日志内容
	retrieveLogs	检索数据库中的日志	检索条件

4.5.3 相关数据表

日志表（包括日志 ID、时间戳、日志内容等字段）

4.5.4 接口设计

1. **writeToDatabase**
   - **功能：** 将日志写入数据库。
   - **HTTP 方法：** POST
   - **URL 路径：** /log/write
   - **请求参数：**
     - log_content (string): 日志内容
   - **响应：**
     - 成功：HTTP 状态码 200，返回成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。
2. **logException**
   - **功能：** 记录异常信息。
   - **HTTP 方法：** POST
   - **URL 路径：** /log/exception
   - **请求参数：**
     - exception_details (string): 异常信息的详细描述
   - **响应：**
     - 成功：HTTP 状态码 200，返回成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。
3. **receiveLog**
   - **功能：** 接收其他控件输出的日志。
   - **HTTP 方法：** POST
   - **URL 路径：** /log/receive
   - **请求参数：**
     - log_content (string): 日志内容
   - **响应：**
     - 成功：HTTP 状态码 200，返回成功信息。
     - 失败：HTTP 状态码 400，返回错误信息。

5 数据库设计

5.1 功能说明

这个数据库设计旨在支持一个程序的需求，包括下载记录、PT 下载密钥、RSS 订阅管理、播放日志、代理配置、硬盘操作日志和一般日志的管理。每个表的设计旨在满足特定功能需求，并在整体设计中具有一致性和有效性。

5.2 数据表设计

5.2.1 表名：DownloadRecords，下载记录表

字段名	类型	默认	注释
record_id	INT	AUTO_INCREMENT	下载记录的唯一标识符
file_name	VARCHAR(255)		下载文件名
download_date	DATETIME	CURRENT_TIMESTAMP	下载日期时间
download_status	ENUM('pending', 'completed', 'failed')	'pending'	下载状态

5.2.2 表名：PTDownloadKeys，PT 下载密钥表

字段名	类型	默认	注释
key_id	INT	AUTO_INCREMENT	密钥记录的唯一标识符
key_value	VARCHAR(100)		PT 下载密钥值
description	TEXT		密钥描述信息

5.2.3 表名：RSSFeeds，RSS 订阅管理表

字段名	类型	默认	注释
feed_id	INT	AUTO_INCREMENT	订阅的唯一标识符
feed_url	VARCHAR(255)		RSS 订阅的 URL 地址
last_checked	DATETIME		上次检查订阅的时间

5.2.4 表名：PlayLogs，播放日志表

字段名	类型	默认	注释
log_id	INT	AUTO_INCREMENT	播放日志的唯一标识符
media_name	VARCHAR(255)		媒体文件名
play_date	DATETIME	CURRENT_TIMESTAMP	播放日期时间
play_duration	INT	0	播放时长（秒）

5.2.5 表名：ProxyConfig，代理配置表

字段名	类型	默认	注释
config_id	INT	AUTO_INCREMENT	配置记录的唯一标识符
proxy_type	ENUM('HTTP', 'HTTPS', 'SOCKS')		代理类型
proxy_host	VARCHAR(100)		代理主机地址
proxy_port	INT		代理端口
username	VARCHAR(50)		认证用户名
password	VARCHAR(50)		认证密码

5.2.6 表名：DiskOperationLogs，硬盘操作日志表

字段名	类型	默认	注释
log_id	INT	AUTO_INCREMENT	日志的唯一标识符
operation_type	ENUM('copy', 'move', 'delete', 'rename')		操作类型
file_path	VARCHAR(255)		文件路径
operation_date	DATETIME	CURRENT_TIMESTAMP	操作日期时间

5.2.7 表名：GeneralLogs，日志表

字段名	类型	默认	注释
log_id	INT	AUTO_INCREMENT	日志的唯一标识符
timestamp	DATETIME	CURRENT_TIMESTAMP	日志时间戳
log_content	TEXT		日志内容