摘要

在网络在线教学的普及过程中，云课堂系统将在传统教学当中发挥着重要作用。近年来，云课堂教学系统将不在局限于在线教学场景，而是深入传统教学，为传统线下教学赋能。WebRTC 技术的出现给互联网音视频行业带来了巨大的转机，使得 Web 端能具备在线音视频交流的能力。

本文将基于 WebRTC 实现一个 Web 版云课堂教学系统，通过完善的课程管理机制实现教学的基础工作，利用 WebRTC 技术打造 Web 低延迟直播教学，利用多样化的教学辅助工具提升教学效率和便利性，结合 ChatGPT 大模型技术来实现作业批改场景提效工作。设计与实现一个现代化云课堂教学系统。为广大教师和学生提供高质量、便捷的在线教育服务。旨在满足在线教学和课程管理的需求。

关键词：云课堂，WebRTC，在线教学，ChatGPT

ABSTRACT

During the popularization of online education through network-based teaching, the cloud-based classroom system plays a significant role in traditional education. In recent years, the cloud-based classroom system has expanded beyond online teaching scenarios and has empowered traditional offline education. The emergence of WebRTC technology has brought a significant turning point to the Internet audio and video industry.

This paper aims to develop a web-based cloud classroom teaching system using WebRTC. The system will incorporate a comprehensive course management mechanism as the foundation of the teaching process. By leveraging WebRTC technology, the system will facilitate low-latency live streaming of web-based teaching sessions. Additionally, a variety of teaching aids will be employed to enhance teaching efficiency and convenience. Furthermore, the integration of the ChatGPT large model technology will be utilized to improve the efficiency of assignment grading scenarios. The design and implementation of this modern cloud classroom teaching system aim to provide high-quality and convenient online education services for teachers and students. The system intends to fulfill the requirements of online teaching and course management.

Keywords: Cloud-based Classroom, WebRTC, Online Education, ChatGPT

目录

第 1 章 引言

云课堂教学系统概念的提出早在 2012 年便有人提出，彼时对云课堂系统的设想是利用互联网的资源去完善传统教学的课堂内容以及打造一种全新的教学环境让学生能更好参与教学当中[1]。彼时的云课堂系统大都是以 MOOC（Massive Open Online Course）类似的教学资源分享平台[2]。MOOC 是一种在线学习模式，是指由各种不同机构或者教育平台提供的免费或低价在线课程，任何人都可以通过互联网不受时间和空间的限制去进行自主学习。MOOC 的出现，对于教育信息化有着重要的推动作用。

直到 2019 年底爆发的新冠疫情给网络教学行业带来了新的发展机遇，为了应对新冠疫情带来的学校延期开学的情况，各地学校通过网络在线直播的方式来实现远程网络教学[3]。此时大都使用的是传统在线会议系统来实现远程网络教学，这些在线会议系统诞生之初并未考虑到网络教学，因此在一些教学的场景中存在些许不足，例如教学资源、教学作业、课程人员、课程录制等方面。近些年随着互联网技术的不断发展，云课堂教学系统也在不断的发展和改进。 今天，云课堂教学系统不仅可以用于远程教学，还可以将在线课程和传统线下教学进行深度融合，以更好地辅助教师进行教学和学生的学习。 这意味着，云课堂教学系统将成为未来教育的重要组成部分，并将在教育领域发挥更大的作用。在今年 ChatGPT 的爆火中，人工智能技术快速进入了大众的视野。大型语言模型在教育领域也将带来更多的发展机遇[4]。将人工智能技术去应用到云课堂教学系统当中无疑会带来更好的体验。

综上，本文将研究相关技术，开发一个深度契合教学的云课堂教学系统。该系统不仅能实现线上教学功能，还将开发辅助教学相关功能，例如课程管理、录制工具、资料管理、作业管理等功能。针对教师，本系统还将提供教学辅助工具，例如教学资料库、课程录制工具、作业管理系统、课程视频系统等，以此来提高教学效率。同时，系统还将接入 OpenAI，利用大模型来进行自动化批阅作业，帮助教师减轻工作压力。针对学生，本系统将提供便捷的课程管理，方便学生整理自己的课程信息，提升学习效率。同时，学生还可以通过系统与教师和同学进行互动和交流，以此来拓展学习渠道和方式。本文将实现一个现代化云课堂教学系统，旨在帮助教师和学生更加方便高效地学习，并提高教学质量和效果。

第 2 章 需求分析

2.1 系统概述

本论文将设计和实现一个基于 Web 的云课堂教学系统，旨在满足在线教学和课程管理的需求。该系统将在 Web 浏览器中运行，用户无需安装任何额外的软件即可使用。除了提供主要的在线教学功能外，该系统还将为教学场景开发相关的辅助功能，例如提供屏幕云录制功能，方便教师录制教学内容并实现快速便捷的录制和分享功能；提供教学资源上传分享功能，便于管理和快速分享教学课件资料；提供自动化作业批阅和收集功能，降低教师批阅和整理作业的复杂性。通过这些辅助功能，本系统将大大提高在线教学和课程管理的效率和便捷性，满足教学过程中的各种需求。

2.2 功能需求

(1) 课程管理

要成为一个完善的课堂辅助利器，仅实现在线教学是不够的，如何更好的成为一个教学辅助工具，与市场上的在线教育软件做出差异化的关键就在于是否拥有一个完善的课程管理系统。为了契合教学需求，本系统将打造课程管理、课件管理、课程作业管理、课程视频管理四大功能。课程管理提供多种课程加入方式：老师邀请、用户申请、申请码；课件管理可以让老师更好的分享和管理自己的课程资料，学生也能更方便的去获取和下载；课程作业管理则方便老师去发布作业，提供强大的富文本编辑器和附件上传功能，能更好的兼容各类作业，同时学生也能更加方便查看自己的作业情况，本系统还提供一键导出所有学生的课程作业归档文件，方便老师去存留学生的作业信息，再也不需要老师去慢慢整理了。视频管理功能可以管理老师的课堂录制或者云录制视频，更加方便了学生去查看。

(2) Web 在线直播

本系统需要实现教师在 Web 端可以对自己的屏幕或摄像头进行推流直播，而学生也能在 Web 端去观看教师的直播内容。这是本系统的核心功能，通过 Web 端的在线直播可以让云课堂系统更加便捷，对于教师而言可以随时随地去进行直播教学直播，不必去下载任何软件和进行各种参数配置就能快速便捷的在线教学。对于学生而言也同样随时随地仅通过一个浏览器就能观看在线直播教学，方便了学习。

(3) 直播视频自动录制

在线教学的直播内容是很有必要去进行保存的，课程直播的录制回放可以方便学生对课程的重复学习以及对教学资源的分享。因此在云课堂教学系统中有一个直播视频的自动录制系统是十分必要的。通过直播视频流的自动化录制将会方便教师和学生的使用和提升效率，当教师在进行直播授课时，无需关注如何去进行课程录制，也不需要配置任何参数即可实现课程直播回放的自动生成。

(4) 屏幕云录制

在教学的场景中，屏幕录制是一个很常见的需求。它可以帮助教师或者学生去记录一些信息。本系统将要实现屏幕云录制功能，在本系统上可以实现开箱即用的在线云录制功能，并能对录制文件进行管理，可以下载录制文件或者分享分享录制文件。虽然 Web 端实现屏幕录制不是难事，但是市场上的一些主流 Web 端录制工具都是只支持本地录制。在这个使用很频繁的功能上无论是本地客户端录制或者是浏览器本地录制都需要耗费大量时间去上传到云端才能去分享给其他用户。而且课程教学的录制文件是极其庞大的，传统的方式都是老师将文件上传到网盘或者社交软件用户在去下载使用，无论对于老师或者学生都是很耗费时间的事情。本系统将解决这项痛点，实现一个开箱即用的录制功能，让用户更加方便的仅需屏幕录制，同时实现云录制的功能，录制结束文件在云端生成，分享视频直接分享一个链接即可。

(5) 自动化批改作业

在今年 ChatGPT 的爆火中，人工智能走向了大众的眼中，现在，AI 已经可以去替代我们去干很多事情了。我在想是否可以将 AI 融入到本系统当中，使之更加易用，去替代一些我们日常的重复性活动。于是我想到了批改作业，在教学当中老师的一大活动便是批改大量的作业，是否可以将这项工作交给 AI 去干呢，这样老师可以在 AI 批改之后稍微检查一下就行了。不必花费大量精力去重复性的批改作业。于是本系统将对接 OpenAI，实现在用户提交作业之后由 AI 自动去批改用户的作业给出评分和评语。

2.3 性能需求

(1) 实现 Web 直播低延时

本系统是一个教学系统，对于在线课程教学而言，最重要的便是低延迟，因为老师需要经常与学生去进行互动交流，在在线教育领域，延迟过大是不可接受的。因此本系统的在线直播的平均延时将控制在 300ms 左右，最大延时将不超过 1s。要实现超低延时的直播方案一半会选取 SRT 协议或者 WebRTC（Web Real-Time Communications）协议，由于 WebRTC 协议在各个平台都有成熟的 SDK，兼容性和可维护性都很强[5]，因此本系统将采用 WebRTC 协议。

(2) 实现高并发直播间聊天室

在实际教学当中，为了应对频繁的教师与学生聊天互动，直播聊天室的设计需要满足高并发的需求，特别是公开课的场景，当学生的数量非常大的时候，如果此时聊天室里大量用户同时使用的话，有可能会造成聊天系统的崩溃，进而影响到整个系统。因此本系统的设计将会避免将聊天系统耦合在业务系统当中，同时会考虑到大量用户使用的场景，将基于高性能的 Netty 框架打造一个极高性能 Web 聊天服务器。

(3) 实现录制回放的实时生成

直播系统的回放生成速度会很大程度上影响用户的体验，特别是教学类的场景，如果能做到实时生成直播回放将会极大提升用户体验和提高用户效率。为了实现这个需求，本系统将借助 HLS（Http Live Streaming）的流媒体技术[6]，这本是应用在直播上的一种流媒体传输协议，它使用 M3U8 文件来组织音视频内容，并使用 TS（Transport Stream）格式来传输音视频数据。其原理是将视频流切割成一个个小的片段，通过一个文本文件去记录这些片段的信息，从而实现直播流的分发。而本系统将仿照此技术设计思想，通过额外拉取直播的流去切割成一个个视频片段去直接上传到文件存储服务器，同时会在缓存服务器中去记录这些片段信息，当教师结束课程的时候，则会请求服务器通过将缓存中的片段信息生成 M3U8 格式的文件。用户通过访问 M3U8 文件即可实现视频的播放，以此来实现录制回放的实时生成。

第 3 章 技术选型

3.1 WebRTC

3.1.1 技术概述

WebRTC 是一项实时通讯技术, 它是谷歌于 2011 年 5 月发布的一项开源技术, 该技术允许网络应用或者站点，在不借助中间媒介的情况下，建立浏览器之间点对点的连接，在不借助任何额外的第三方软件或插件的情况下实现音频和视频数据的传输[7]。

在 WebRTC 中，参与视频通讯的双方必须先交换 SDP(Session Description Protocol) [8]信息，SDP 是一种会话描述协议，SDP 携带有关媒体类型、格式、编解码器的信息。以此来实现通信中双方视频流编解码的格式统一，避免用户 A 无法解码用户 B 编码的视频流情况。通常客户端需要借助信令服务器来交换 SDP 信息和网络信息，两台电脑通过中间的信令服务器交换彼此的 SDP 后，即可建立点对点的连接，如图 3-1 所示。

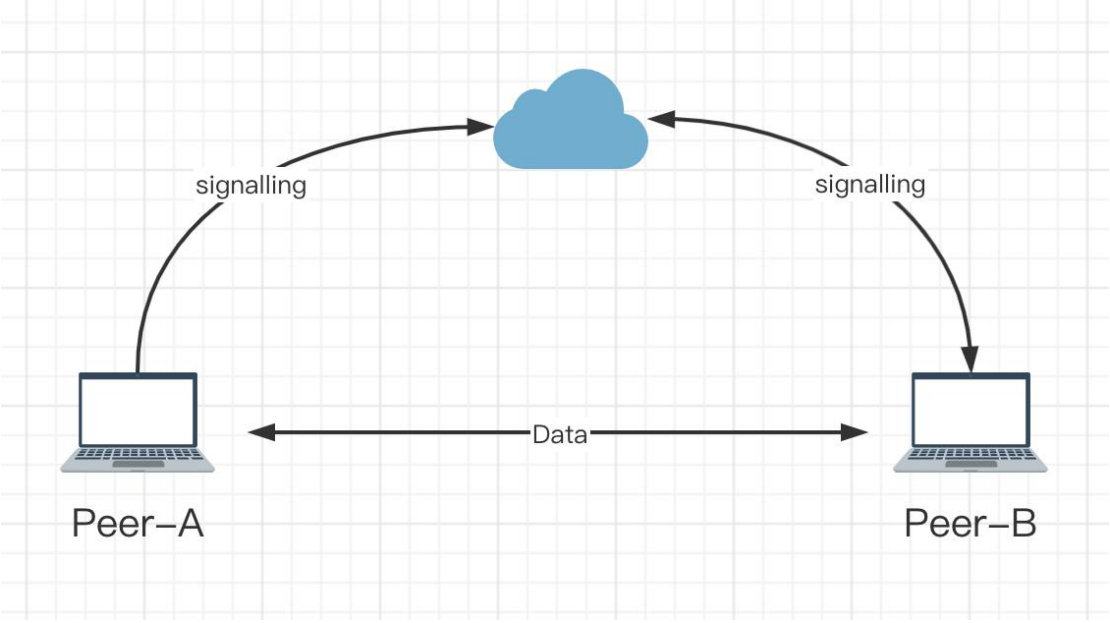

图 3-1 建立 WebRTC 连接过程

在 2020 年受新冠的影响下，Chrome 用户通过 WebRTC 技术来观看视频流的数量增加了 100 倍[9]，这还不包括在隐身模式下和不选择共享统计信息的用户。而各大浏览器厂商对 WebRTC 技术也给予了极大的支持，纷纷积极适配，各浏览器版本对 WebRTC 的支持情况，如图 3-2 所示。

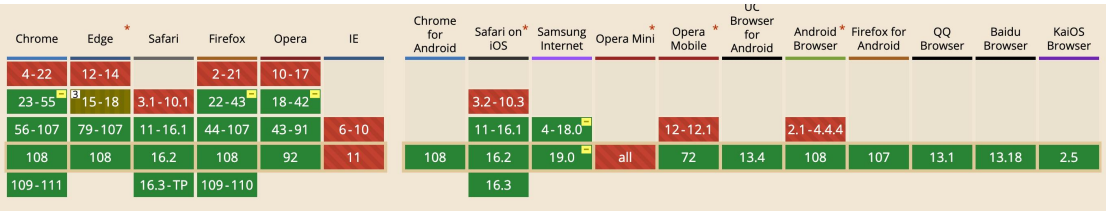

图 3-2 各浏览器版本对 webrtc 相关技术的支持统计，截图来自 https://caniuse.com/#search=webrtc

综上，如果需要在 Web 端实现在线直播，WebRTC 是一个绕不开的技术，结合近几年 WebRTC 技术的发展情况以及未来发展趋势，可以发现在实时音视频领域，WebRTC 将是一个核心的技术点。本系统也因此基于此技术来设计与实现。

3.1.2 技术应用

采用 WebRTC 技术来实现 Web 端的在线直播场景，如果直播观看人数不多的情况下，效果是很好的。但是一旦人数增加，其弊端也逐渐增大，原因是 WebRTC 是建立点对点的连接，如果人数很多的情况下，需要建立起很复杂的网络拓扑结构，带来的后果就是客户端的网络带宽压力将变得极其庞大，因为每个客户端都需要跟其他任何一个客户端去建立连接。在直播教学场景当中就体现在教师端需要和每个学生去建立连接，如图 3-3 所示。

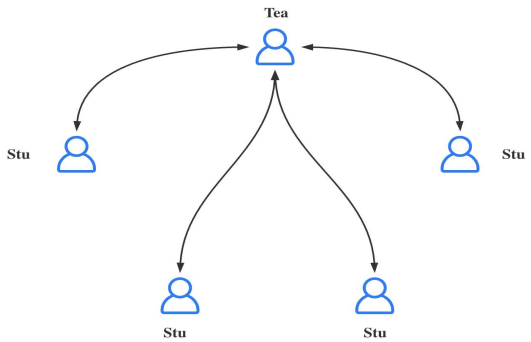

图 3-3 基于原生 WebRTC 在线直播时的网络拓扑

因此教师端的网络带宽压力会随着学生的增多而增大，一旦达到教师客户端网络上行带宽阈值，则可能会导致在线直播间崩溃。无疑直接采用 WebRTC 来作为云课堂直播系统设计是不合理的。为了解决这个问题，需要在原始架构的基础上在中间增加一层流媒体服务器来解决，这样对教师而言只需要将自己的直播流发送到中间服务器，对于学生也只需要从中间服务器去拉取直播流。如图 3-4 所示。

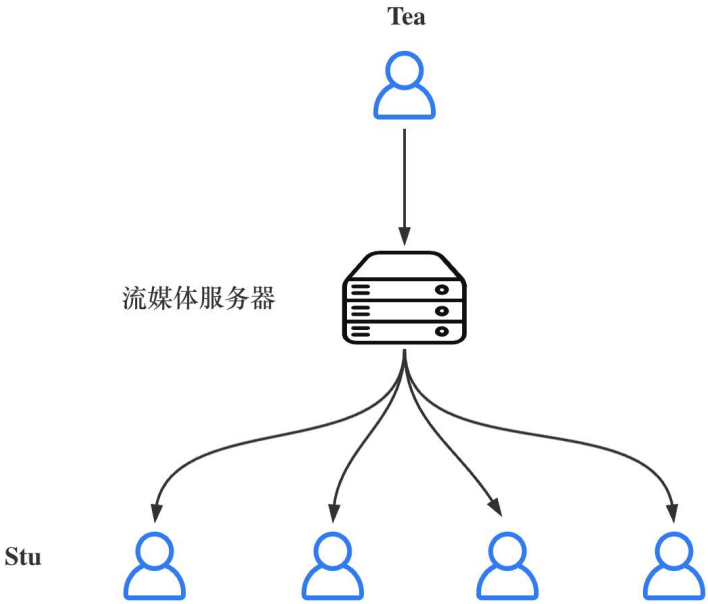

图 3-4 增加中间层的 WebRTC 在线直播时的网络拓扑

从而将带宽压力转移给服务器，避免用户端带宽增加导致系统崩溃。而服务器则可以通过集群的方式来分摊压力。本系统将采用开源流媒体服务器 SRS（Simple Realtime Server）来结合 WebRTC 来打造在线直播教学系统[10]。SRS 是一个简单高效的实时视频服务器，支持 RTMP、WebRTC、HLS、HTTP-FLV、SRT 等多种实时流媒体协议。也是高性能的流媒体服务器，是同类服务器的 2~3 倍性能，提供非常完整的概念和一致性设计。

3.2 WebSocket

3.2.1 技术概述

WebSocket[11]是一种能在 Web 浏览器和服务器之间进行实时、双向数据通信的技术。它是 HTML5 中新增的一种协议，与 HTTP 不同的是，WebSocket 建立一次连接后，客户端和服务器之间可以保持长连接，实现数据的实时传输，而不需要客户端不断地向服务器发送请求，这样可以避免浪费大量的带宽和服务器资源。在实际应用中，WebSocket 可以应用于聊天室、多人协同编辑、在线游戏、实时股票行情等实时数据传输场景，非常适合于需要实现实时数据交互的应用。相比于传统的 HTTP 协议，WebSocket 的优点在于它能够更高效地实现实时数据传输，具有更快的响应速度，更加节省带宽和服务器资源，提升了用户体验。
在 Web 端实现全双工通信，WebSocket 无疑是最好的选择。WebSocket 的出现正是为了弥补 HTTP 半双工通信的缺陷，具体实现原理是通过 HTTP 协议进行一个握手，然后建立起 TCP 的长连接，在此基础上用真正的 WebSocket 协议进行通信。

3.2.2 技术应用

基于 WebSocket 全双工通信的特点，本系统将采用此技术来实现直播间的聊天功能，因为在聊天场景中客户端需要和服务器频繁进行信息交流，而服务端也要能实现向客户端主动推送消息的能力，，使用传统的 HTTP 轮训显然不太适合，因此需要建立一种长连接的全双工通信，进而实现服务端和客户端的双向数据传输。

为了保障服务器支撑更多的 WebSocket 客户端连接数，本系统将采用 Netty 框架来构建 WebSocket 服务器，Netty 是一个基于 Java 语言开发的高性能、异步的网络框架，可以用于开发追求高性能的网络应用程序。Netty 的核心是一套基于 NIO 的异步 I/O 模型，使用了一系列高度优化的算法和数据结构，可以在高并发和大负载的情况下保持高效稳定的性能。同时，Netty 也提供了一套完整的网络编程框架，包括各种常见的协议和组件，例如 HTTP、WebSocket、TCP、UDP 等。

3.3 FFmpeg

3.3.1 技术概述

FFmpeg 是一个用 C 语言开发的的开源音视频处理工具，可以用于转码、剪辑、合成、压缩、解码等多种音视频处理任务。FFmpeg 是基于命令行的工具，可以使用简单的命令行参数来完成各种音视频处理任务。它支持多种音视频格式，还支持多种编码和解码器，例如 H.264、AAC、VP8 等。用户可以使用 FFmpeg 对音视频进行各种操作，例如提取音频或视频、调整视频大小和帧率、添加水印、合并视频等。FFmpeg 是一个非常强大和灵活的音视频处理工具，因此广泛应用于多个领域。它可以用于音视频编辑、视频转码、视频剪辑、流媒体服务器、多媒体播放器等。

3.3.2 技术应用

本系统需要录制和处理视频文件，因此选择了 FFmpeg 作为视频处理工具。不过，FFmpeg 是基于命令行的工具，无法直接在后端应用中通过接口调用。为了解决这个问题，本系统采用了 Go 语言的 Gin 框架来对 FFmpeg 进行封装，Gin 框架是一款基于 Go 语言的 Web 框架。以此提供 HTTP 接口供其他服务进行调用。通过这种方式，其他服务可以通过网络调用该接口，以实现对 FFmpeg 的操作和视频文件处理。这样就能方便地使用 FFmpeg 进行视频处理，并且能够更好地与其他应用进行集成。

3.6 MinIO

3.6.1 技术概述

MinIO 是一个流行的开源对象存储服务，可以在本地或云端部署，用于存储和管理大量的数据对象。它的设计初衷是为了提供一套简单易用、高性能、可扩展的对象存储方案，可以支持多种数据类型，例如图片、视频、文档等。MinIO 最大的优点是它具有非常高的性能和可扩展性，可以轻松地扩展存储容量和并发访问量，可以满足不同规模和需求的用户。此外，MinIO 还支持 Amazon S3[12]协议，Amazon S3 是由亚马逊公司提供的一种云端存储服务，它能够提供高可用性、可伸缩性和安全性的存储服务，可以存储和检索任意数量的数据对象，支持在任意时间、任意地点、任意规模的应用程序中使用。这就意味着使用 MinIO 可以让用户轻松地将数据导入和导出到其他平台和服务中。从而使系统的可维护性和扩展性大大增加，同时也让数据备份变得更加容易。

3.6.2 技术应用

本系统将使用 MinIO 搭建云课堂教学系统的中心文件存储服务器，用来存储图片、视频以及各种文件，以便于实现文件的下载和分享等功能。通过后端使用 MinIO 的 SDK 去实现文件的上传，将文件管理服务单独独立出整个系统，有利于文件的统一管理以及降低整个系统的耦合性，使用 MinIO 作为文件存储服务器，可以将文件存储在多个磁盘上，从而实现文件存储的可用性和可靠性。MinIO 支持分布式部署，可以在多个节点之间进行负载均衡，以提高系统的性能和可扩展性。同时，MinIO 还提供了灵活的访问控制和权限管理功能，可以根据用户和角色进行细粒度的权限控制。总之，采用 MinIO 作为中心文件存储服务器，可以为云课堂教学系统提供高可用性、高扩展性和高性能的文件存储服务，以及便捷的视频分享功能。

第 4 章 系统设计

4.1 系统架构设计

4.1.1 架构概述

本系统的架构图如图 4-1 所示。

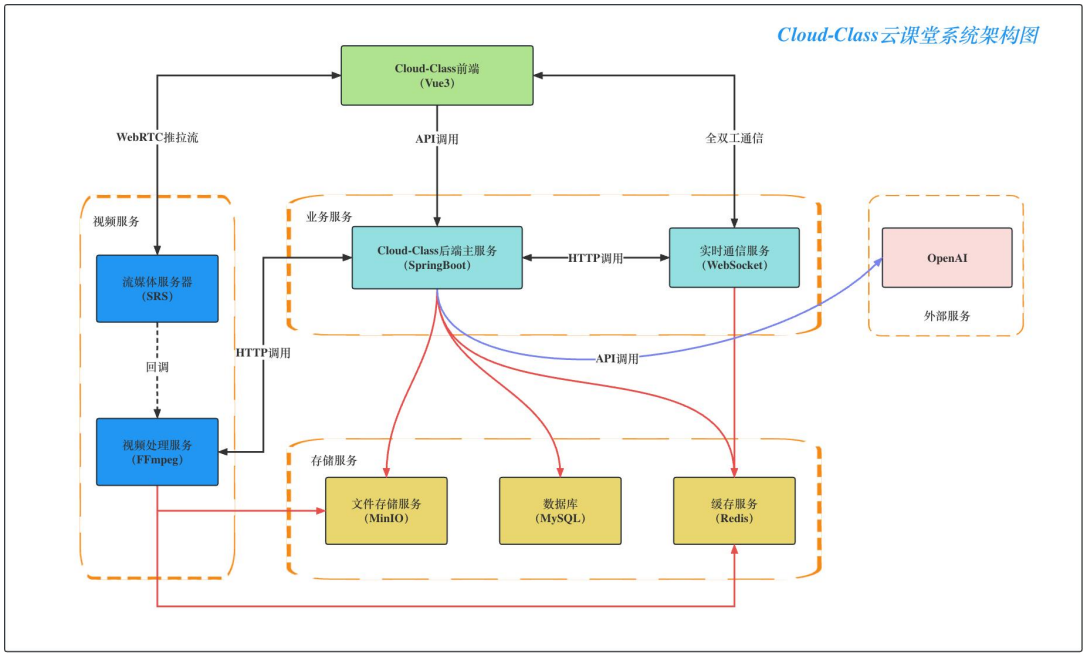

图 4-1 基于 WebRTC 的云课堂教学系统架构图

本系统总体上采用了前后端分离的设计，前端是用户访问本系统的入口。后端服务有四大模块，分别是业务服务、视频服务、存储服务和外部服务。业务服务是本系统的核心后端服务，主要是系统功能的实现；视频服务是本系统中直播和录制相关功能的实现；存储服务主要是为本系统提供数据持久化支持；外部服务是本系统需要调取开发 API 的服务。

4.1.2 基础服务

(1) 流媒体服务器

本系统的直播流媒体服务器使用的是 SRS 搭建的，通过配置 WebRTC 推流与拉流实现本系统的核心直播功能。同时通过配置切片回调来实现直播的自动化录制功能，提高了教学资源的利用率和效率。

(2) 数据库

首先，本系统采用 MySQL 作为主要的数据库存储服务。MySQL 是一种流行的关系型数据库管理系统，具有高可用性、高可靠性和高性能等特点。MySQL 提供了丰富的功能和灵活的配置选项，可以满足云课堂教学系统对于数据存储和管理的各种需求。通过 MySQL，本系统可以存储和管理用户信息、课程信息、作业信息、答案信息等核心数据，以支持系统的各种功能和服务。

(3) 缓存服务

本系统采用 Redis 作为中心化缓存服务。Redis 是一种基于内存的数据结构存储系统，具有高速读写、数据持久化、分布式集群等特点。在云课堂教学系统中，Redis 主要用于缓存课程信息、作业信息、用户信息等业务数据，以提高系统的性能和响应速度。通过 Redis，本系统可以快速地读取和写入缓存数据，避免频繁地查询数据库，减少系统负载和响应时间。

(4) 文件存储服务

最后，本系统采用 MinIO 来实现文件存储服务。MinIO 是一种高性能、高可用性的对象存储系统，支持分布式部署、S3 API 等特点。在云课堂教学系统中，MinIO 主要用于存储各种文件和媒体资源，包括课程视频、作业文件、答案图片等。通过 MinIO，本系统可以快速地上传、下载和分享各种文件，以满足用户的需求。

4.1.3 业务系统

本系统采用了前后端分离的设计，前端使用 Vue3、TypeScript、Vite、Pinia 等框架和技术实现，提供用户 Web 访问页面。与此同时，后端服务共有三个：系统业务主服务、实时通讯服务、视频处理服务。

(1) 系统业务主服务

系统业务主服务是本系统的后端主服务，主要负责业务系统的相关接口提供和与前端服务通信。该模块采用的是 Java 语言和 SpringBoot 框架搭建。负责本系统的数据库操作以及调用其他模块工作，此外该模块也通过暴露相关接口来实现其他模块的信息查询工作。

(2) 实时通讯服务

实时通讯服务则是负责直播间聊天室的通信服务后台，承担聊天信息的中转以及直播间消息指令的收发。该模块使用的是 Java 语言，基于 Netty 框架实现的一个 WebSocket 后台服务器。由于 Netty 框架具有高性能、高并发、可扩展性等优点，因此可以有效地支持大量的在线用户同时进行聊天，保证了直播间聊天室的稳定运行。

(3) 视频处理服务

视频处理服务则是负责录制视频的剪辑转码工作和处理直播录制回调的工作。该模块使用的是 Go 语言，基于 Gin 框架同时封装了 FFmpeg 来实现的。通过 Gin 框架提供的 HTTP 接口，其他服务可以方便地调用该服务，从而实现通过网络调用接口来操作 FFmpeg 去处理视频文件，提高了视频处理的效率和可扩展性。同时该模块通过接收流媒体服务器的回调来来实现自动化上传录制切片文件到文件服务器和记录切片的信息。

4.2 系统功能设计

4.2.1 功能概述

本系统将分为四大模块：直播系统、录制系统、课程系统、消息系统。如图 4-2 所示。

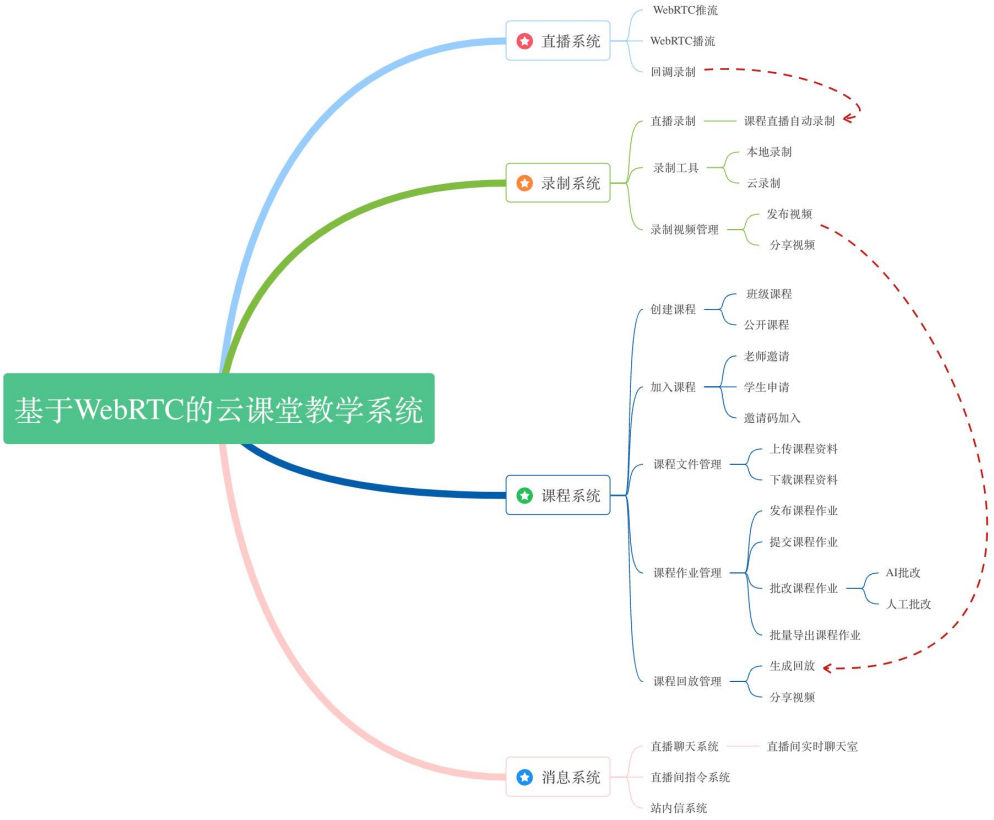

图 4-2 基于 WebRTC 的云课堂教学系统功能图

4.2.2 课程系统

课程系统是本系统的核心业务系统，本系统的所有功能都是围绕课程展开的，对于本系统而言用户在系统层面是没有角色区分的，也就是说用户可以选择自己去创建一个课程或者加入一个课程。本系统的用户角色划分是在一个课程层面的，用户在一个课程内部有教师和学生两种身份，教师就是创建该课程的用户，学生是加入该课程的用户。用户加入一个课程可以有三种方式：教师邀请用户、用户直接申请、用户通过邀请码加入，如图 4-3 所示。

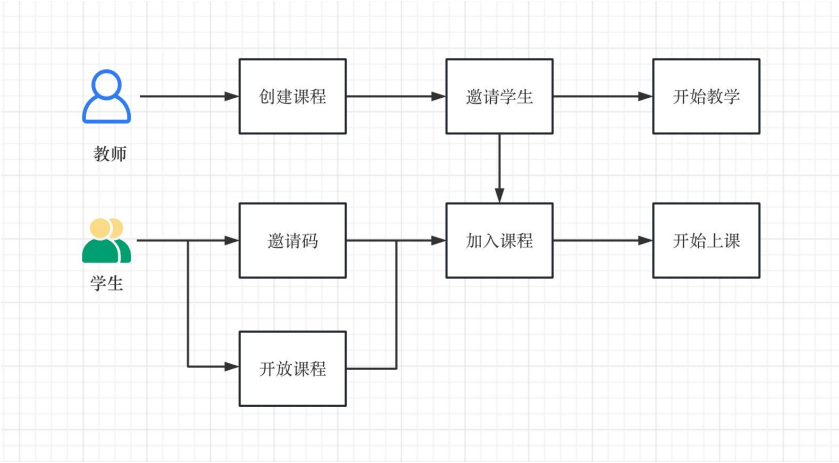

图 4-3 课程系统工作流程

课程的分类有两种：公开课程和班级课程。公开课是指用户可以在主页的公开课程查询到的课程，用户可以直接申请加入公开课程，公开课程没有课程人数上限。班级课程是指用户无法直接查询到的课程，加入班级课程只有老师邀请或者邀请码加入的方式，班级课程有 200 人数限制。课程系统下面有课程资料管理、课程回放管理、课程作业管理，通过课程资料管理，教师可以发布课程相关学习资料用来供学生下载查看。课程回放管理是教师课程直播的录制，方便用户可以在去重复查看课程教学视频，教师也能将自己录制的视频发布到课程当中。

课程作业管理是课程文件管理中核心的功能，在传统布置课程作业的场景中主要有两个痛点，一是教师要花费大量精力去批阅大量的作业，这是一个重复性的活动，二是作业的收集归档功能，教师需要对所有学生的作业进行一个整理和统一格式的操作，方便后续的作业存档。该功能也是教学场景当中最为频繁使用的。教师可以发布课程作业，同时可以选择是否 AI 自动批阅。当选择 AI 批改后，学生在提交作业之后，本系统便会收集学生作答信息自动请求 OpenAI 接口获取批改结果，如图 4-4 所示。

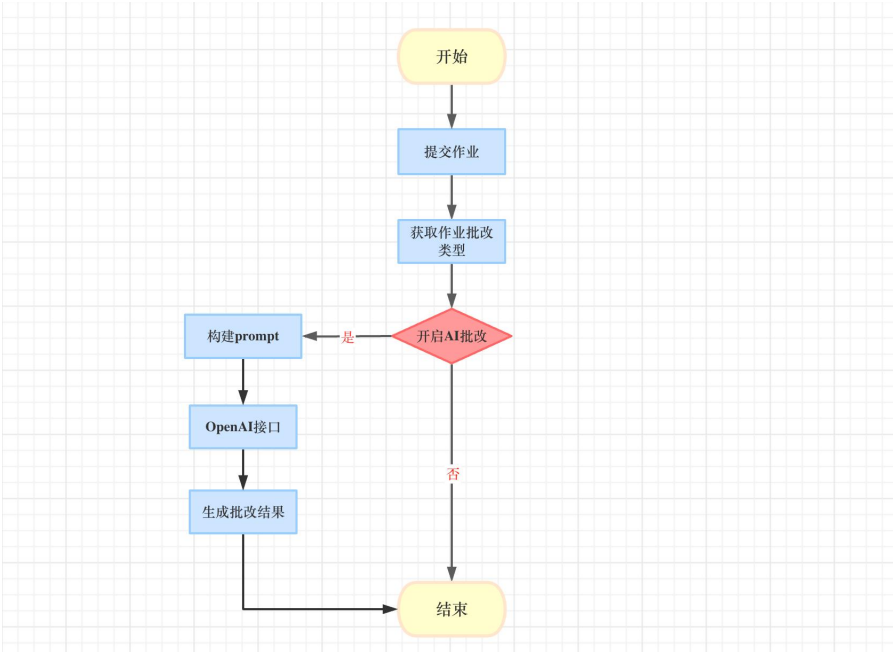

图 4-4 自动化批改作业流程图

教师可以查看用户作业的提交信息，也可以对 AI 批改的结果进行修改。当教师想要收集学生提交作业信息进行归档时，本系统也提供此功能，通过收集每个学生的提交信息，将提交信息按照标准模板转化为 PDF 文件，在将所有 PDF 文件进行压缩操作返回给教师所有人的提交信息。

4.2.3 直播系统

直播系统是本系统的核心模块，此部分主要是实现 WebRTC 推流和 WebRTC 拉流的功能，此模块是基于开源流媒体服务器 SRS 来搭建的直播服务器，用户通过服务器获取推拉流地址，通过推拉流地址与流媒体服务器建立连接，通过 WebRTC 技术来实现 Web 端推流和拉流，如图 4-3 所示。

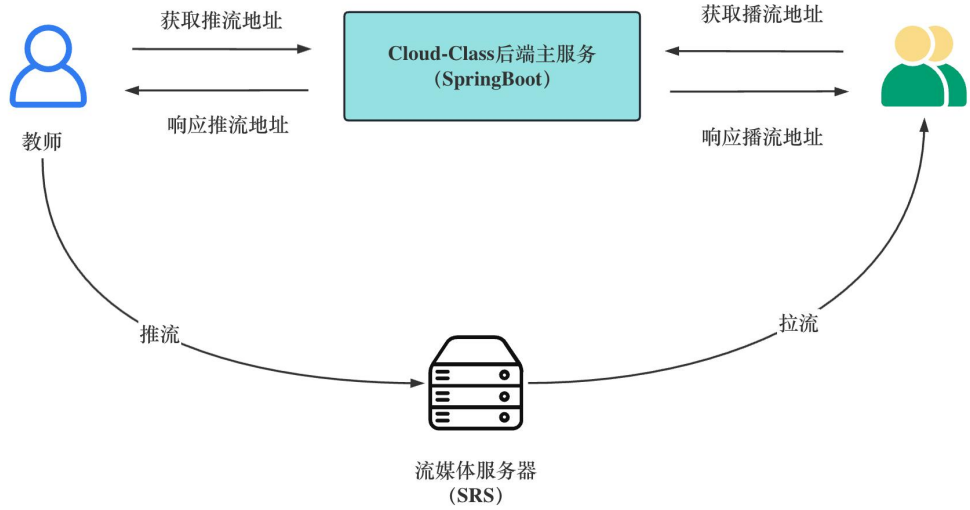

图 4-5 直播系统工作原理

此系统的具体工作流程是，首先由教师在 Web 端开启直播开启直播，此时会去获取直播房间相关信息，例如推流地址、直播间人员、课程信息等，当教师选择共享屏幕或者共享摄像头之后，会生成相应的视频流对象，之后会构建一个 RTCPeerConnection，它代表了本地端机器与远端机器的一条连接。然后便是获取本地的 SDP 信息，拿到本地的 SDP 信息之后便会将本地的 SDP 信息发送给流媒体服务器，流媒体服务器在收到客户端的 SDP 信息之后便会与服务器的 SDP 信息进行比较返回一个公用的 SDP 信息，此时客户端将会等待服务器响应的 SDP 信息，一旦接收到服务器响应的 SDP 信息，此时客户端便可以和流媒体服务器去建立 WebRTC 连接了，建立连接之后便可以进行发送或者接收视频流了。

4.2.4 录制系统

录制系统是本系统的一个工具系统，旨在为用户提供方便的录制工具应用，其关键的两部分是直播自动录制和云录制工具。

(1) 直播自动录制

此模块是为了实现教师直播教学内容的自动录制生成直播回放视频。为了实现直播的自动化录制以及直播录制视频的实时生成，本模块的实现原理是通过将直播视频流文件切割成一个个小的视频片段，具体流程如图 4-4 所示。

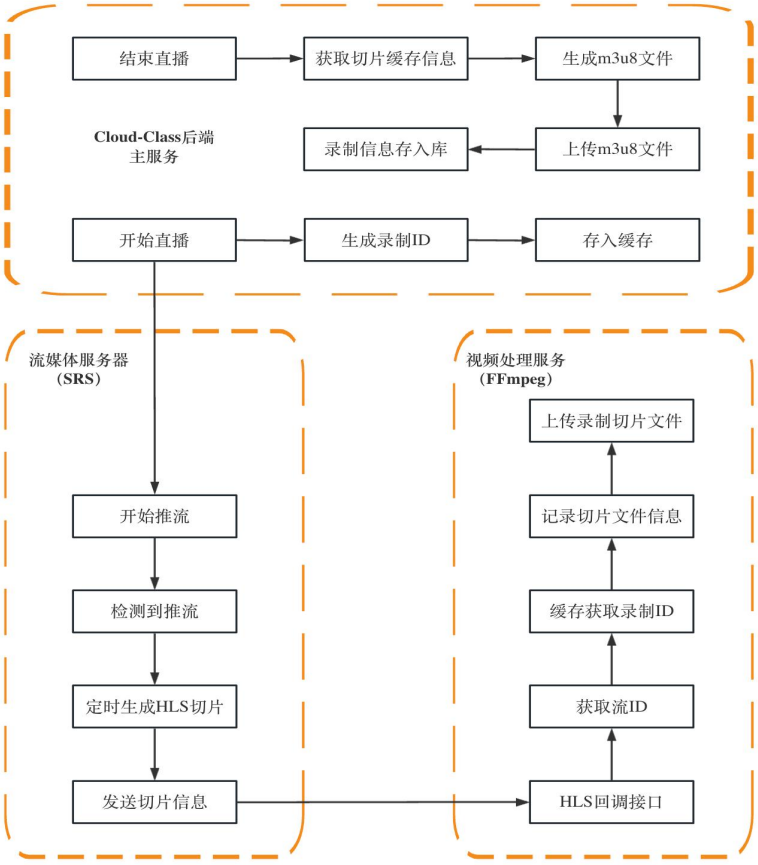

图 4-6 直播自动录制工作流程

当流媒体服务器 SRS 在检测到推流时，通过配置回调接口，来实现获取流文件切片，通过视频处理服务来记录切片文件详细信息并存储到缓存服务当中，如流名称、切片名、切片时长、切片编号、切片文件路径等。然后再将切片文件上传到文件存储服务器。当结束直播时便可以在缓存当中获取这些切片文件信息，在通过切片文件信息去生成 M3U8 文件。最后就能生成录制视频相关信息了。用户获取 M3U8 文件即可播放视频。

(2) 云录制工具

云录制工具是提供给用户去进行视频录制或者屏幕录制的一个工具型应用，支持云录制。其工作原理是如图 4-4 所示。

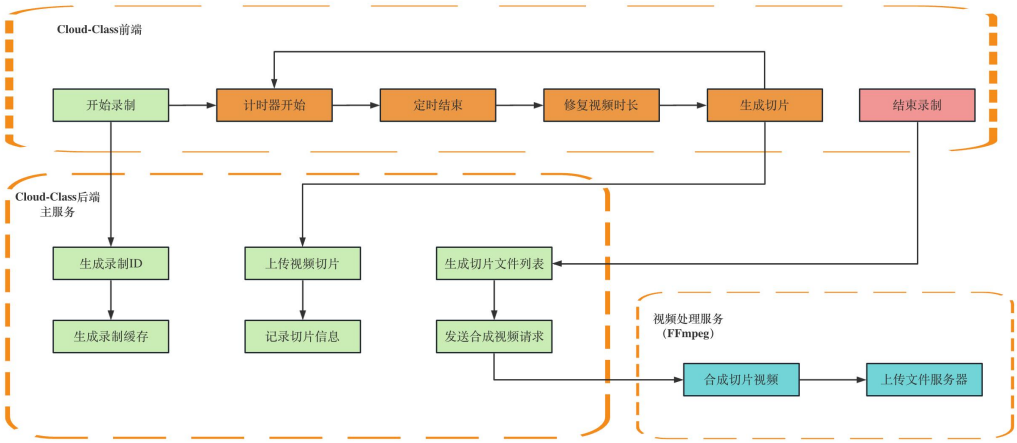

图 4-7 Web 云录制工作流程

当用户开启云录制后，会向后端发送时一个开始录制的请求，此时后端服务会生成一个唯一的录制 ID，返回给前端，同时也会在缓存当中记录改录制 ID 的相关录制消息，例如录制用户、录制状态、录制开始时间、结束时间、切片信息等。当前端拿到录制 ID 后便可以正式开始录制了，为了实现实时云录制的效果，传统的录制完成在上传的方案就无法使用了。因此本系统采用了切片上传的方式来实现，其原理就是每隔一段时间将录制好的视频上传到服务器，由于是短时间的视频上传速度是很快的，服务器在收到切片视频后会先保存在本地缓存，同时记录切片文件信息，当用户结束录制的时间，后端再将所有的录制切片文件进行合成处理，由于是简单的视频拼接并不设计视频编解码，因此这个过程的速度是非常快的，当合成完毕后便会上传给文件存储服务器，由于视频处理服务器会跟文件存储服务部署在同一个内网当中，所有这个过程的速度也是很快的。通过以上流程来进行云录制，就能实现对于用户来说无感云录制，用户在结束录制后无需等待漫长的视频上传时间，同时云录制完毕后也是能秒级的生成录制视频。本系统采用的录制框架是 RecordRTC，这是一个开源的 Web 录制框架，但是这个框架有个缺陷，就是录制的视频没有时间信息。为了解决这个缺陷，本系统的做法是通过定时产生切片的同时去记录切片的时长信息，在去修复没有时间的切片文件。

4.2.5 消息系统

消息系统是本系统的沟通与通知功能实现的模块，本系统的消息系统分为三个部分：直播聊天消息、直播间指令消息、站内信。

直播聊天消息是通过 WebSocket 服务器来收发，本系统的 WebSocket 服务器是基于 Netty 框架构建的。其工作流程是客户端在进入直播间之后先发起 WebSocket 链接，当 WebSocket 服务器收到客户建立连接请求之后，首先会将用户的 token 拿去请求业务系统，在校验通过之后服务端才会同意建立连接，当直播房间内的用户发送消息的时候，会将消息内容直接发送到 WebSocket 服务器，服务器会请求业务系统获取直播房间全部用户信息，在将消息给发送给所有用户。为了实现转发消息的高性能，本系统使用 Java19 最新的虚拟线程技术，虚拟线程是一种协程[13]技术，协程技术是一种轻量级的并发编程模型，它允许程序在单个线程中实现并发执行多个任务。协程是比线程更轻量级的并发单位。在传统的多线程模型中，每个线程都需要占用一定的系统资源，如堆栈、线程控制块等。而协程只需占用较小的内存空间，可以在一个线程中创建大量的协程，因此更节省系统资源。从而实现更高的并发量与吞吐量，特别契合本系统中发送群消息的场景。

直播间指令消息是指当教师开启、关闭摄像头或者屏幕的时候要通知直播间的学生端去进行相应的加载与关闭的操作，还有教师在结束上课时也要通知直播间用户课程已结束。此类指令消息由于是服务端向客户端主动推送，因此此功能也是通过 WebSocket 来实现的。

站内信的功能是用来提醒用户一些业务操作的执行结果以及系统提醒。包括加入课程结果反馈、课程作业发布提醒、作业批改结果通知、课程直播开始提醒等情况通知。本模块的实现是放在业务系统当中，因为站内信功能是一个业务耦合功能，会设计到消息的反馈，并不是一个单向的通知。其中最重要的便是学生申请加入课程，教师可以在收到消息提醒时去进行同意或拒绝操作。因此这种业务耦合性模块并未将其与主业务系统分开。

4.3 数据库设计

4.3.1 数据库概述

本系统的数据库将基于 MySQL 打造，具体数据表情况如表 4-1 所示。

表 4-1 数据表汇总

t_user	用户信息表
t_course	课程信息表
t_course_user	课程用户关系表
t_course_file	课程文件表
t_course_playback	课程回放表
t_course_job	课程作业表
t_course_job_commit	课程作业提交表
t_live	课程直播信息表
t_video	录制视频表
t_message	站内信息表

4.3.2 数据表设计

(1) t_user

用户信息表具体设计如表 4-2 所示。

表 4-2 用户信息表详情

字段名	字段类型	字段信息
id	int	用户 id
user_name	varchar	用户账号
password	varchar	用户密码
avatar	varchar	用户头像链接地址
real_name	varchar	真实姓名
age	int	年龄
sex	tinyint	性别， NO(1,”未知”),MAN(2,”男”),WOMAN(3,”女”)
tel	varchar	手机号
email	varchar	邮箱
status	tinyint	状态，0:停用,1:正常

(2) t_course

课程信息表具体设计如表 4-3 所示。

表 4-3 课程信息表详情

字段名	字段类型	字段信息
id	int	课程 id
course_name	varchar	课程名
course_code	varchar	课程唯一编码
intro	varchar	课程简介
course_type	tinyint	课程类型,1：班级课,2：公开课;
cover	varchar	课程封面图片地址
teacher_id	int	外键，用户 id'
start_time	datetime	课程开始日期
end_time	datetime	课程结束日期
join_end_time	datetime	加入课程的截止时间
week_arrange	varchar	课程每周安排
user_num	int	课程上限人数
need_review	tinyint	加入课程是否需要审核，默认 0 需要
need_open	tinyint	是否公开，默认公开 0

(3) t_course_user

课程用户关系表具体设计如表 4-4 所示。

表 4-4 课程用户关系表详情

字段名	字段类型	字段信息
id	int	课程-用户关系 id
course_id	int	外键，课程 id
student_id	int	外键，用户 id
join_type	tinyint	加入类型，1:老师邀请，2:用户申请
selected_time	datetime	选课时间
status	tinyint	状态枚举, 1:未审核, 2:通过, 3:驳回

(4) t_course_file

课程文件信息表具体设计如表 4-5 所示。

表 4-5 课程文件信息表详情

字段名	字段类型	字段信息
id	int	课程文件 id
course_id	int	外键，课程 id
upload_user_name	varchar	上传用户账号
file_name	varchar	文件名
cover	varchar	文件图标地址
size	varchar	文件大小
file_intro	varchar	文件简介
upload_time	datetime	上传时间
file_link	varchar	文件下载地址

(5) t_course_playback

课程回放信息表具体设计如表 4-6 所示。

表 4-6 课程回放信息表详情

字段名	字段类型	字段信息
id	int	课程回放 id
course_id	int	外键，课程 id
upload_user_id	int	上传用户 id
playback_name	varchar	回放名
cover	varchar	视频封面图片地址
upload_time	datetime	发布日期
playback_file_link	varchar	回放文件地址

(6) t_course_job

课程作业信息表具体设计如表 4-7 所示。

表 4-7 课程作业信息表详情

字段名	字段类型	字段信息
id	int	课程作业 id
course_id	int	外键，课程 id
job_name	varchar	作业名
job_intro	varchar	作业简介
content	longtext	作业内容
ai_review	tinyint	是否开启 AI 自动批改，0:不开启，1:开启
playback_file_link	varchar	回放文件地址
course_id	int	外键，课程 id
start_time	datetime	作业发布日期
end_time	datetime	作业截止提交日期
job_file_link	varchar	作业附件下载地址

(7) t_course_job_commit

课程作业提交信息表具体设计如表 4-8 所示。

表 4-8 课程作业提交信息表详情

字段名	字段类型	字段信息
id	int	作业提交内容 id
course_job_id	int	外键，作业 id
student_id	int	外键，学生 id
content	int	提交内容
commit_time	datetime	提交日期
commit_file_link	varchar	提交附件下载地址
summary_link	varchar	作业批改文件下载地址
ai_check	tinyint	是否 ai 批改
status	tinyint	状态枚举, 1:未批改, 2:已批改, 3:驳回
score	int	批改分数 (0~10)
comment	longtext	作业批改评语

(8) t_live

课程直播信息表具体设计如表 4-9 所示。

表 4-9 课程直播信息表详情

字段名	字段类型	字段信息
id	int	直播房间 id
room_id	varchar	直播房间唯一编码
user_video_publish	varchar	摄像头 publish 地址
user_video_play	varchar	摄像头 live 地址
screen_video_publish	varchar	屏幕 publish 地址
screen_video_play	int	屏幕 live 地址
course_id	int	外键，课程 id

(9) t_video

录制视频信息表具体设计如表 4-10 所示。

表 4-10 录制视频信息表详情

字段名	字段类型	字段信息
id	int	录制视频 id
record_id	varchar	录制视频唯一编码
user_id	int	外键，用户 id
video_name	varchar	录制视频名称
video_intro	varchar	录制视频简介
video_file_link	varchar	录制视频文件地址
video_cover_link	varchar	录制视频封面图片地址
record_start	datetime	录制视频开始时间
record_end	datetime	录制视频结束时间
open	tinyint	视频是否公开

(10) t_message

站内信息通知表具体设计如表 4-11 所示。

表 4-11 站内信息通知表详情

字段名	字段类型	字段信息
id	int	信息 id
msg_name	varchar	信息名称
msg_type	tinyint	信息类型
msg_read	tinyint	信息状态，是否已读
send_time	datetime	发送时间
send_id	int	外键，发送者用户 id
receive_id	int	外键，接收者用户 id
msg_content	longtext	信息内容
need_reply	tinyint	是否需要回复，默认不需要
confirm_url	varchar	确认信息 token
refuse_url	varchar	拒绝信息 token

第 5 章 系统实现

5.1 课程系统实现

5.1.1 课程管理

课程管理分为加入的课程和创建的课程，如图 5-1 和 5-2 所示。

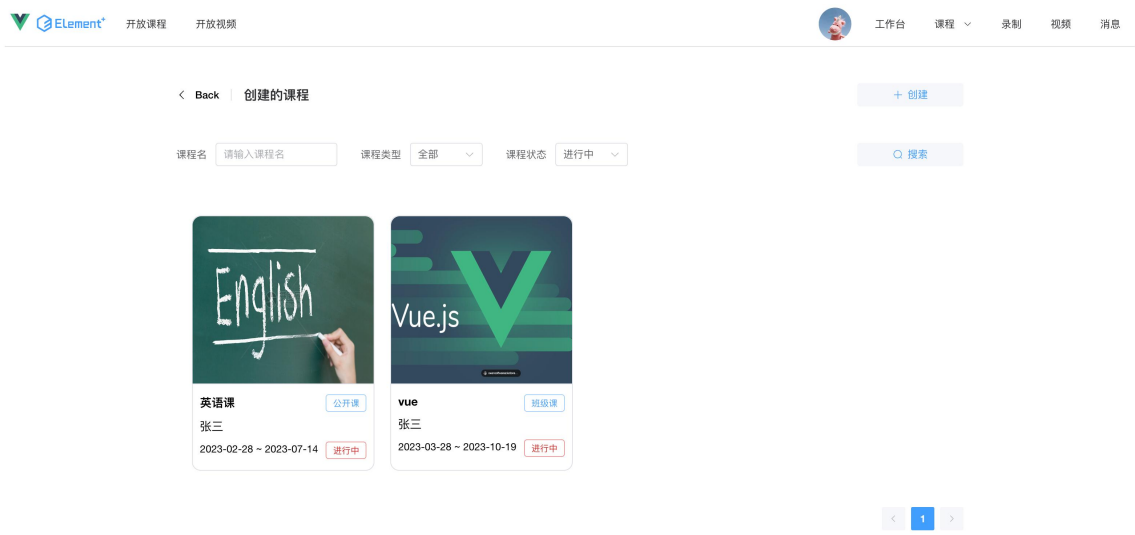

图 5-1 用户创建的课程

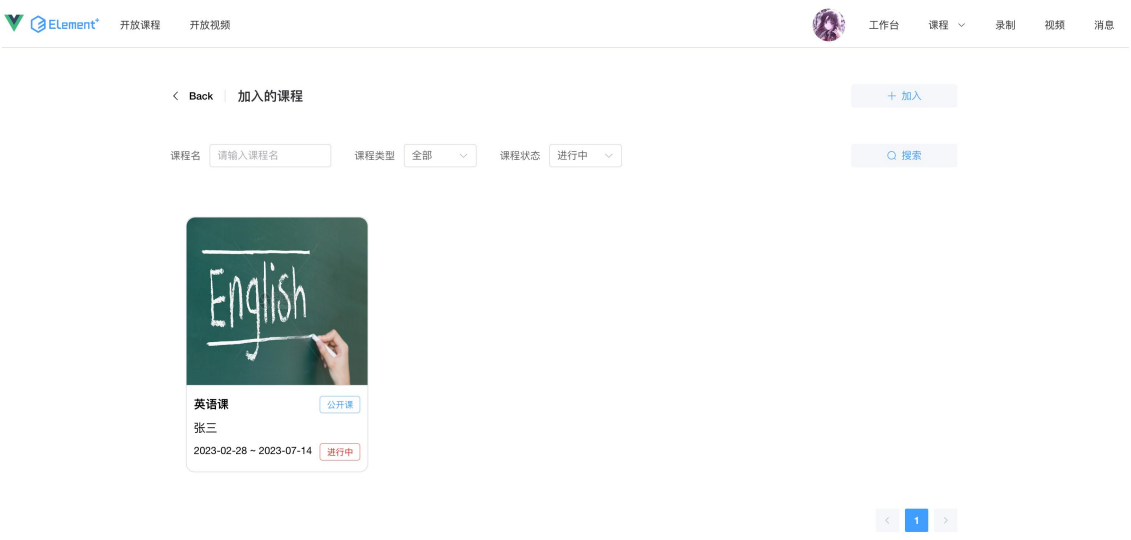

图 5-2 用户加入的课程

5.1.2 作业 AI 批改

通过在学生提交作业后构建 prompt 去请求 OpenAI 的接口来实现学生作业的自动批改，prompt 的设置通过结合作业题目和学生作答然后加上打分范围和打分要求，在实际使用中选择“text-davinci-003”模型，同时将 temperature 设置为 0，因为批改作业是一个相对严格的操作，需要将 temperature 降低来实现 GPT 回答的客观性。AI 批改作业效果如图 5-3 所示。

图 5-3 作业 AI 批改结果

5.2 直播系统实现

5.2.1 在线直播

教师可以选择屏幕共享或者打开摄像头来进行直播，如图 5-4 所示，学生可以在线观看教师直播，如图 5-5 所示。

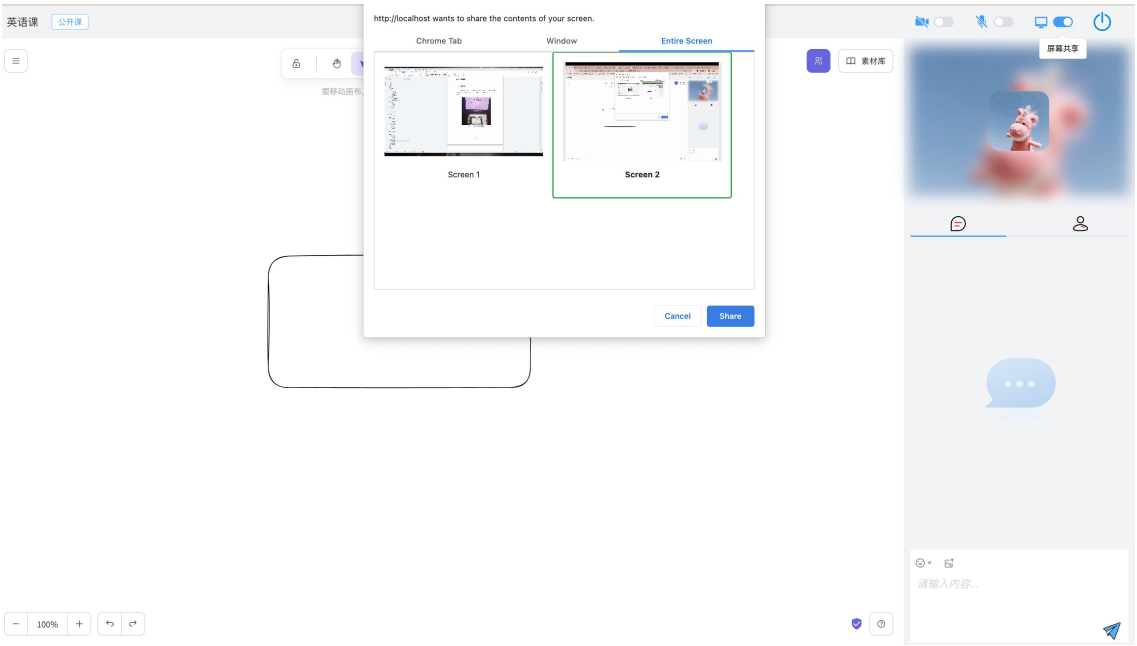

图 5-4 教师在线直播

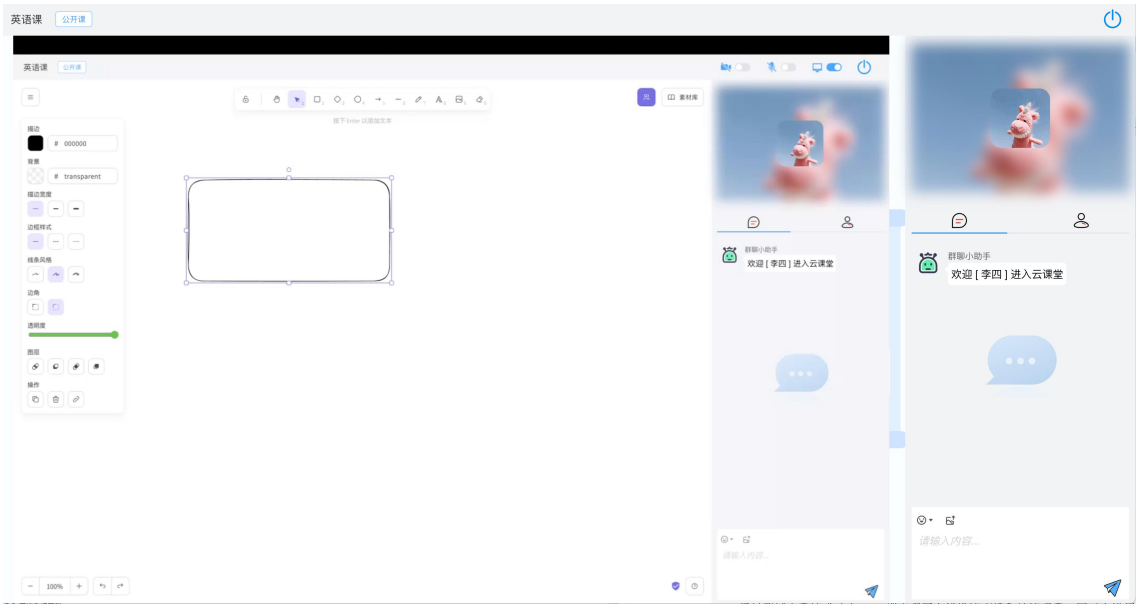

图 5-5 学生观看直播

5.2.2 直播延迟

经过测试本系统成功在 Web 端实现了在线推流直播和拉流观看。同时在进行共享屏幕直播时的延时不超过 300ms，成功实现了低延时直播的需求。如图 5-6 所示。

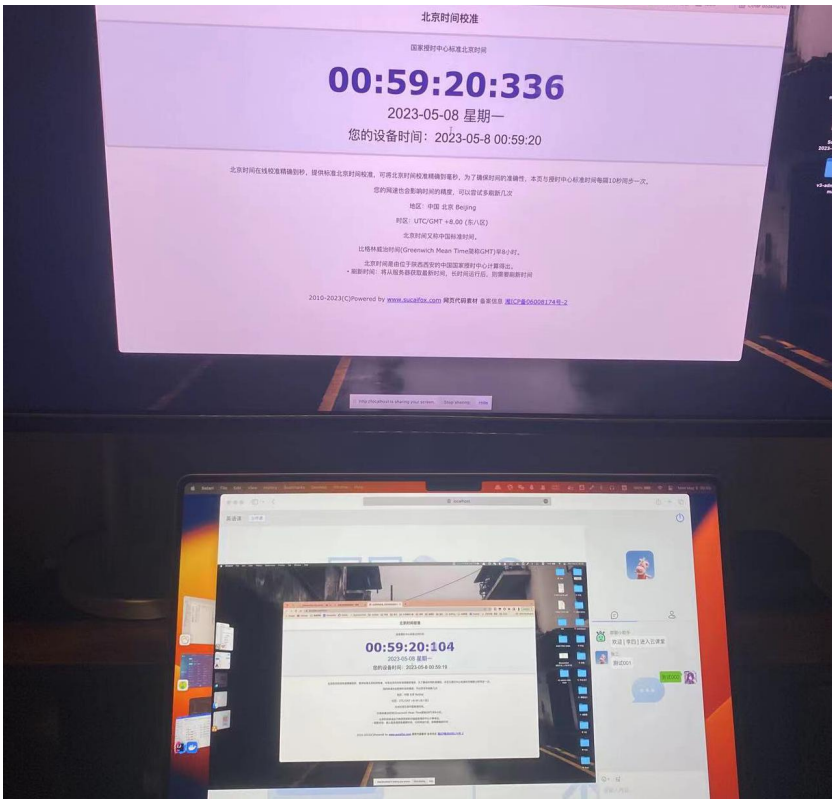

图 5-6 Web 在线课堂直播延时测试

5.3 录制系统实现

5.3.1 直播录制

当直播结束之后系统会自动将课程直播进行录制，如图 5-7 所示。

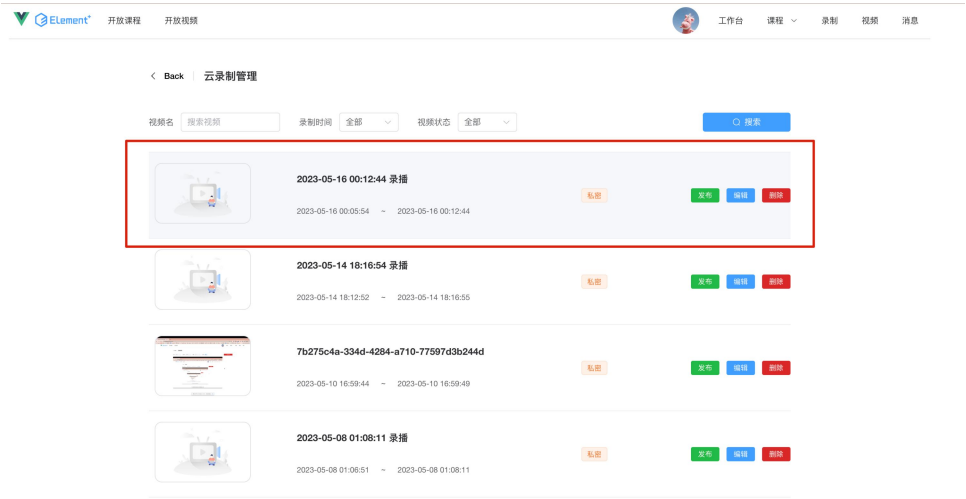

图 5-7 直播结束生成录制回放

5.3.2 录制工具

本系统提供的录制工具可以实现用户选择屏幕或者摄像头进行录制，开放本地录制和云录制的选择，使用云录制可以实现录制视频的秒级生成，如图 5-8 所示。

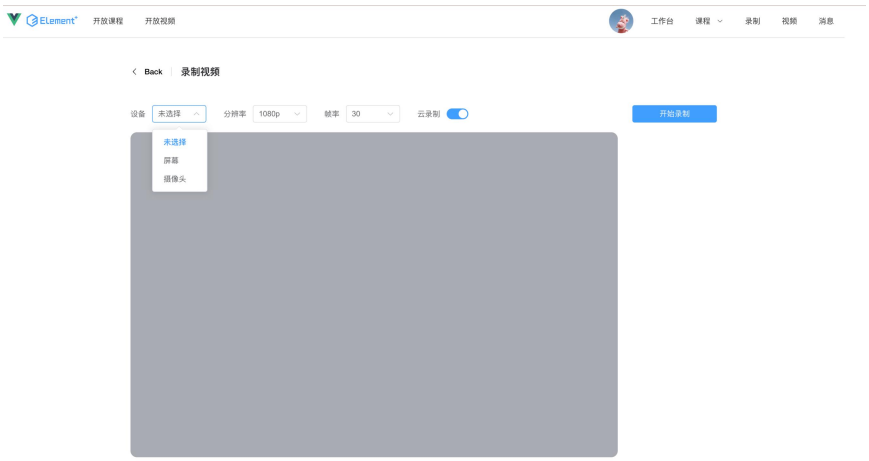

图 5-8 Web 录制工具

5.4 消息系统实现

5.4.1 直播聊天

在同一课堂的教师和学生可以进行聊天交流，如图 5-9 所示。

图 5-9 教师和学生在课堂聊天

5.4.2 站内信

用户可以管理和查看自己的消息通知，包括加入课程结果反馈、课程作业发布提醒、作业批改结果通知、课程直播开始提醒等情况通知，如图 5-10 所示。同时能对需要反馈的信息进行处理，如图 5-11 所示。

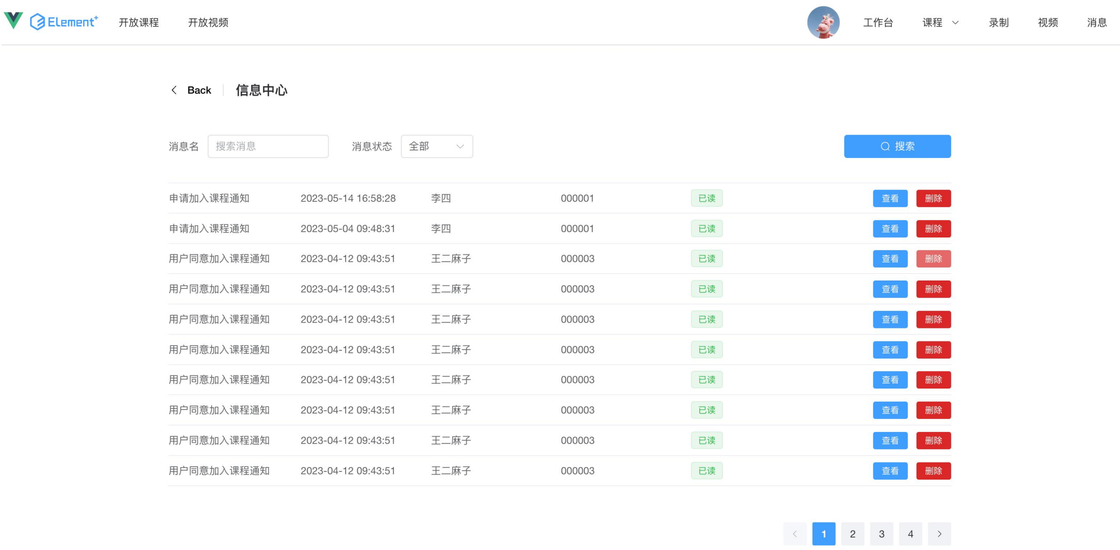

图 5-10 信息管理中心

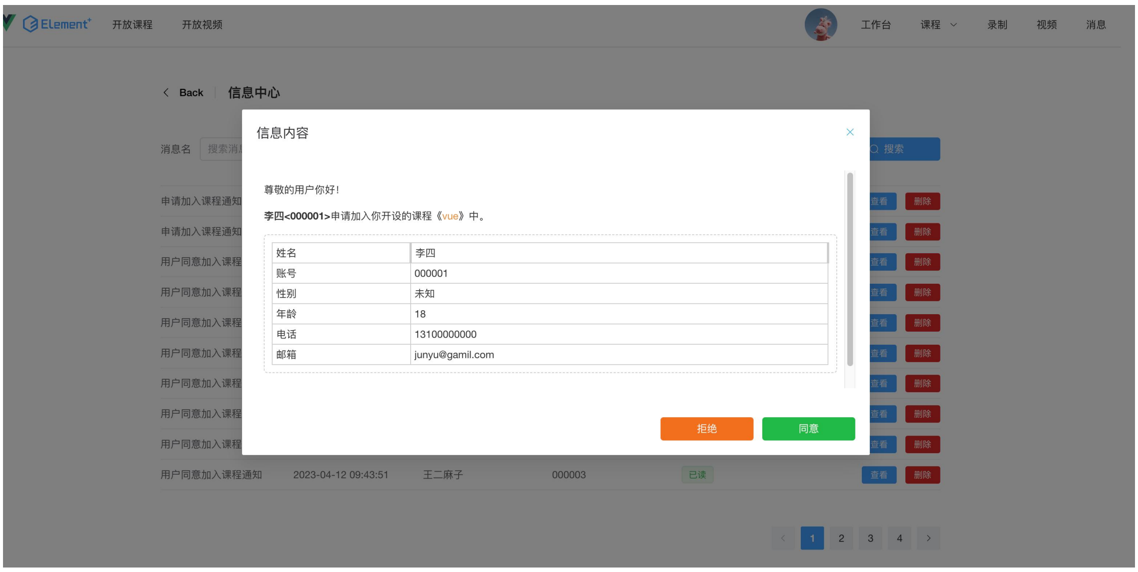

图 5-11 处理回复信息

第 6 章 总结和展望

本文主要介绍了基于 WebRTC 的云课堂教学系统的设计与实现。首先，阐述了系统的需求，通过对功能和性能两方面来进行需求分析，明确了本系统的设计目标与期望的实现结果。然后通过进行相关技术的研究，通过分析各种技术在云课堂教学系统中的应用场景和优势，最终进行选择最合适的技术，通过对这些技术的深入研究来设计其在本系统当中的应用，完成的相关技术的改进和完善来实现更契合本系统的功能。然后通过系统架构设计，奠定了本系统的主体架构与模块。通过进一步的数据库设计来确定本系统中业务数据的定义。然后深入系统的具体功能，通过选择合适技术来实现相应需求。最后通过实际测试来检验本系统的实现结果，通过与最初的需求分析来进行比较，验证了本系统的完成结果和高性能。

综上所述，基于 WebRTC 的云课堂教学系统在实现了实时音视频教学和教学辅助功能的基础上，仍有许多发展的空间。未来，云课堂教学系统还可以与人工智能、大模型、大数据分析等技术深度结合，提供个性化教学推荐和学习评估，为教师和学生提供更精准的教学和学习支持，与时俱进地适应教育行业的需求，为广大教师和学生提供高质量、便捷的在线教育服务。

致谢

首先，我要衷心感谢我的指导老师程志刚老师，程老师在我整个研究过程中给予了我宝贵的指导、支持和专业知识。我还要深深感谢浙江科技学院的老师们，他们的宝贵见解、建设性批评和鼓励给予了我极大的帮助。他们的专业知识和对学术卓越的承诺对我的研究道路产生了深远的影响。

参考文献

[1] 唐晓勇.移动互联技术支撑下的学习变革——“云课堂”的创新设计与实践思考[J].中小学信息技术教育,2012,No.125(05):11-12.

[2] Baggaley J. MOOC rampant[J]. Distance education, 2013, 34(3): 368-378.

[3] 焦建利,周晓清,陈泽璇.疫情防控背景下“停课不停学”在线教学案例研究[J].中国电化教育,2020(03):106-113.

[4] Kasneci E, Seßler K, Küchemann S, et al. ChatGPT for good? On opportunities and challenges of large language models for education[J]. Learning and Individual Differences, 2023, 103: 102274.

[5] 杨敬一,张辰.冬奥场馆现场信号超低延时直播的设计与实现[J].广播电视网络,2022,29(07):67-69.DOI:10.16045/j.cnki.catvtec.2022.07.035.

[6] 丁耀.基于 Http Live Streaming 的直播技术研究[J].软件,2014,35(03):129-131+135.

[7] Loreto S, Romano S P. Real-time communications in the web: Issues, achievements, and ongoing standardization efforts[J]. IEEE Internet Computing, 2012, 16(5): 68-73.

[8] Handley M, Jacobson V, Perkins C. SDP: session description protocol[R]. 2006.

[9] Blum N, Lachapelle S, Alvestrand H. WebRTC: Real-time communication for the open web platform[J]. Communications of the ACM, 2021, 64(8): 50-54.

[10] 胡国强,周兆永,信朝霞.基于 SRS 的开源直播系统的设计与实现[J].现代电子技术,2016,39(16):36-39+43.DOI:10.16652/j.issn.1004-373x.2016.16.009.

[11] Fette I, Melnikov A. The websocket protocol[R]. 2011.

[12] Palankar M R, Iamnitchi A, Ripeanu M, et al. Amazon S3 for science grids: a viable solution?[C]//Proceedings of the 2008 international workshop on Data-aware distributed computing. 2008: 55-64.

[13] 刘书健. 基于协程的高并发的分析与研究[D].昆明理工大学,2016.