这个文档我会想到哪写到哪，主要是作为一个非正式的思路参考/技术汇总。可能随时会更新（打脸），欢迎补充修正。

1. 通信架构

根据要求，同一个程序要按登录方式划分不同用户角色（虽然很奇怪），包括三类：教师、学生、管理员。由于管理员端负责维护账号信息，因此逻辑上讲教师和学生登录时都要先和管理员端建立通信，可把管理员作为Server，教师和学生作为Client（只是功能上的划分），然后Server作为总控分别和各个Client之间建立连接，传输数据（声音+画面的视频流）和控制信息（登录/登出、点名、试题等）。~~Server很累。~~演示的时候应该会在同一台机子上同时运行Server和教师端Client（有毒），然后围观/评委以Client形式登录。

通信协议方面，个人倾向于用UDP而非TCP进行直播流传输，因为可以Multicast而且对数据完整性要求不高（丢就丢了呗，谁家看视频还不卡一下呢）。控制信息的传输理论上可能TCP更安全稳定一些，但本着先完成最基本功能再考虑花里胡哨的原则~~（俗称迭代开发）~~，先暂时试下都用UDP吧，不行再说。

Windows下我用Winsock进行Socket编程~~（Mac暂且自求多福）~~，简单粗暴还有官方文档。我一般不喜欢搞没用的第三方库，效率足够的话能简单尽量简单。~~不稳定的~~传送门：

<https://docs.microsoft.com/en-us/windows/win32/winsock/windows-sockets-start-page-2>

2. 视频/音频流

要求实时传输画面和声音，肯定是要压缩转码的。这个应用似乎对实时性要求还挺高（B站等直播平台的rtmp推流一般都有几分钟的延迟，但点名回答问题/限时作答这种功能还是要延迟能少则少，雨课堂在这方面表现极其差劲），所以可能要在画质和延迟之间进行一个权衡。像Zoom/微信视频这种实时性就比较好（有个好~~土豆~~服务器真的很重要）。

这个东西我也没仔细搞过，姑且试试FFmpeg吧。开源的，应该至少能用。

<https://www.ffmpeg.org/download.html#get-sources>

3. GUI

个人认为MFC够了，~~因为暂时不考虑苹果用户的感受~~（Qt花里胡哨的，但是可以跨平台，希望以后有Contributor做这个）。

4. 业务逻辑

整个程序的核心逻辑其实就是个消息驱动的事件响应，只是消息种类多了点，但每一个拆出来并不复杂，Client端要根据Server发来的消息调用相应的功能（播放画面、显示题目），然后也会向Server发送消息（登录、提交题目），Server主要就是个传话筒，负责维护每个Client的状态并实时更新的，顺便记录下日志（小本本）。剩下的像什么捕获窗口啦做题计时器啦都属于犄角旮旯的杂碎功能，找对了API的话一个函数搞定。