1. 关于目的

创建这个repo的初衷可能是帮大家一把，或者证明这个东西没那么难写，或者只是写着玩（类似的蛋疼事情我干过不少，比如用C++从零开始写人工神经网络）etc. 我只负责创建这个repo和这个文档，然后可能不定期出于无聊更新一下代码，大概率会更新但是不保证会更新，更不保证不会中途放弃。无论这个项目以后如何发展，希望不要扯到无意义的争论/撕逼中。理念上的话，作为一个作业，能简单就简单，实现功能是首要目的，所以不要吐槽为什么对需求定义这么低，为什么方法这么土，为什么连RSA加密都没有。先能用再说，以后要加什么花，那大家有兴趣就自便呗。

2. 开发思路

鉴于团队人数未知，进展计划未知（也许从头到尾只有我一个人自high，然后一群伸手党坐等，如果这样的话，相信我肯定会半途而废的），这个东西的开发模式可能会偏离标准的软件工程方法，而用一种比较接近直觉的、新手友好的方式，这样任何人想中途加入的话应该也会比较快入手。初步计划把相关模块拆分出来以后，每个模块分别各写各的，分别放在不同的目录里，可以各自作为一个独立的程序分别调试，然后把最恶心的东西放到最后（图形用户界面和业务流程类整合）。整合的时候肯定还要改一堆东西，但我认了。

然而这个事情并不等于完全没有设计。必要的类图/用例图该画还得画，规则该遵守还得遵守，不然搞出来就是一坨逻辑不清、到处依赖的翔，扯都扯不开。进度嘛随缘，如果有的话就在这个目录里。

3. 简要分析

这个东西真的没那么难做，要相信这只是个大作业，没让你把这个程序拿上App Store创收，或者承担全校师生的教学任务（利益相关：我连那老师是谁都不认识，也不会为课程团队说话，但也不要希望我为一些天天想搞大新闻的同学说话。搞这个完全是为了好玩，如果客观上顺便对某些同学起到了帮助作用那很荣幸，但要是有人抄代码被发现了锅也不是我的，不信的话请读一下GNU License的条款。代码就是代码，请不要牵涉道德/立场，我最讨厌这种无聊玩意）。所以很多网络应用要考虑的东西都不需要考虑（比如安全性、服务器荷载容量、各种异常情况等），说白了就是个玩具程序，即使最后写出一万行来，也不会有多复杂。或者说，这个东西按部就班做，是可以一点点做出来的（毕竟还有6周），没让你证明黎曼猜想，也没让你开发一个前所未有的船新算法出来不是吗？只不过这个分数没那么容易拿到而已嘛。我的理解就是，在T大混，不逼自己一把根本不知道自己有多牛逼不是吗？

好了进入正题，这玩意在我看来最奇葩的一点在于这个业务逻辑。照这个说法所有的用户（管理员/教师/学生）是在共用完全相同的程序。那谁是老大呢，应该是拿着管理员权限那个用户作为单中心验证机制对不对（不然怎么验证登录啊），然后教师和学生都只是普通用户。那么这事就有趣了，谁有那个数据库文件谁就是老大，然后Client和Server端完全不区分，服了。不过甲方说这样那就这样吧，细节问题，可能是为了照顾大家感受不用开发两个不同的程序吧（由此可以看出难度还是降低了不少的）。

然后就是这个模块设计了，要求文档上面哩哩啦啦列了12个功能，纯粹是吓唬人的，说白了就那么几个事：建立通讯，传输数据，记录状态日志。通讯的话要求用Socket那就Socket吧，语音和屏幕可以一起压成视频流推出去（如果采用单中心的话，对那个中心服务器还是有一定的带宽要求的，不过压制好了也许不会太离谱），然后答题、登录、签到、注意力这些东西都可以作为附加消息传输（我们也不用去管这个事如何防止学生开挂/作弊/假装听课其实在玩游戏etc），只要分清楚每个事要在学生端实现还是教师端实现还是服务器端实现就行了。然后底层操作的话，获取窗口/音源什么的就调用一下Windows/Mac的API，毕竟我们有CSDN和stackoverflow，这些都不是事。

总之这个程序设计费事的东西应该只有三大块，1建立网络传输逻辑/数据格式和相应的程序解释功能（比如一个单选题如何发送/修改/显示），2 通过OS的API真正实现它，3 搞出GUI把功能咪在一起。比较麻烦，但真的可行。

4. 如何参与项目

欢迎所有人，commit就好了。任何形式的贡献都可以，不一定非要代码。比代码更重要的其实是思路。任何参考资料/资源链接之类的都可以。有兴趣有热情且提交质量高的同学我会加到contributor里。毕竟开源的东西要靠社区，越多人贡献就越容易完成，行胜于言不是吗？

5. 开发环境

我的机子是Windows的，IDE用的VS 2012（别问为啥这么老，我懒得更新）。Mac的我管不着，但是希望有人能搞出相应的branch来。