

Yuntube 直播课堂 设计文档

吕芳锐 曾恬静 廖雅雯
苏 展 郭 帅 侯明烨

目录

引言3

1.1 编写目的3

1.2 项目背景:3

1.3 参考资料3

设计思路.....3

2.1 Web 系统的实现4

2.2 视频、音频采集转码与推流的实现.....4

2.3 即时反馈功能（聊天室）的实现5

功能分配.....5

3.1 教师.....5

3.2 学生.....5

3.3 管理员6

详细设计.....6

4.1 直播编码的实现.....6

4.2 服务器推流及客户端接收的实现7

4.3 在线聊天室的实现8

4.4 课堂管理的实现.....8

4.5 网页设计9

引言

1.1 编写目的

确定直播课堂系统的总体设计、总体结构、模块设计、接口设计为详细设计奠定基础。

1.2 项目背景：

说明：

本项目的名称：Yuntube 直播课堂

本项目的应用范围：各大高校

用户：高校学生、老师

1.3 参考资料

《实用软件工程》(第二版) 郑人杰 殷人昆 陶永雷 清华大学出版社

《软件工程——Java 语言实现》 Stephen R. Schach 机械工业出版社

《实践者的研究方法》 Roger S. Pressman 机械工业出版社

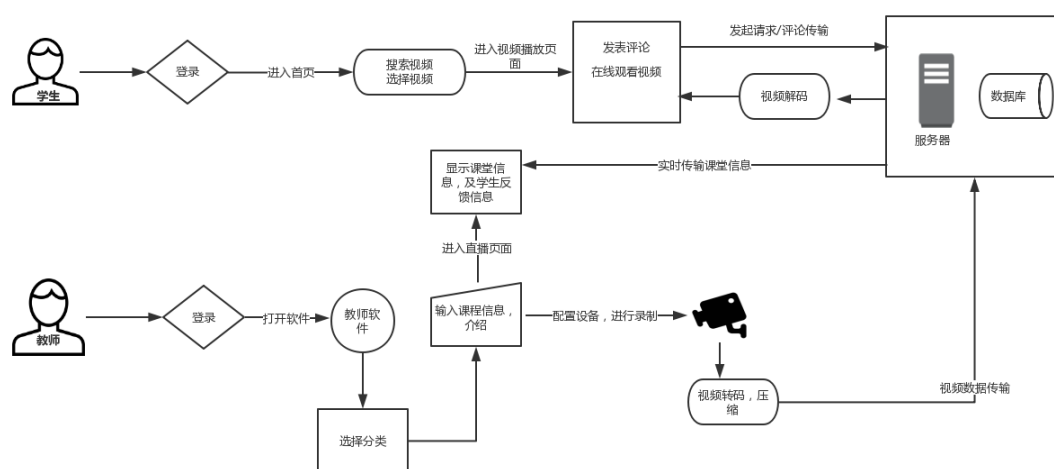
设计思路

根据设计目标，我们要做一个支持直播的课堂 web 系统，主要工作就分为以下四部分：

2.1 Web 系统的实现

首先我们设计一个 web 网页系统，web 网页系统实现主要分为以下几个步骤：

1) 后台服务器环境配置，我们网页服务器环境是 Ubuntu 操作系统，在其上配置 java 环境，安装 mysql 数据库和 8.0tomcat，数据库安装完之后根据功能需求构造 E_R 图，然后在 mysql 中搭建数据库。Er 图如下：



2) 接下来是 web 系统功能实现，主要分为两部分：教师端的功能实现，包括就教师登录，教师个人中心，教师开课申请，教师开始直播，课程即时反馈，课程作业及公告发布等，另一部分部分是学生端的主要功能实现，包括登录及个人中心，搜索课程，申请听课，进入听课，作业提交以及发表评论等功能。我们 web 系统实现主要基于 java 和 jsp 两门语言实现，对于学生或者老师的请求，通过 jsp 提交到后台服务器，服务器根据数据库的和请求计算出结果反馈给客户端，客户端获取信息后再显示在界面上，完成动态信息获取的目的。

2.2 视频、音频采集转码与推流的实现



2.3 即时反馈功能（聊天室）的实现

受限于 http 协议的单向通信，如果实现即时聊天室的话就需要时刻请求服务器，加重服务器负担，而且效果差，于是我们选择可以双向通信的 websocket 协议搭建聊天室，websocket 支持客户端和远程服务器的双向通信，服务器可以主动向客户端发起请求并发送信息，于是每当一个用户在聊天室中发送信息时，服务器收到信息并且广播到所有在此直播间的用户，达到即时聊天室的目的，教师端添加手机 app 聊天室功能，以便教师在上课时即时与学生沟通，而不用时刻看着自己的电脑。

功能分配

3.1 教师

功 能 模 块	浏览	添加	修改	删除	查询	授权
个人主页	★		★			
课程列表	★				★	
聊天室	★	★		★	★	
课堂管理	★	★	★	★	★	
查询课程	★				★	
后台管理						

3.2 学生

功 能 模 块	浏览	添加	修改	删除	查询	授权
个人主页	★		★			
课程列表	★				★	
聊天室	★	★		★	★	

课堂管理						
查询课程	★				★	
后台管理						

3.3 管理员

功 能 模 块	浏览	添加	修改	删除	查询	授权
个人主页	★		★	★	★	
课程列表	★	★	★	★	★	
聊天室	★	★		★	★	
课堂管理	★	★	★	★	★	★
查询课程	★	★	★	★	★	
后台管理	★	★	★	★	★	★

详细设计

4.1 直播编码的实现

ActionScript 采用 Sorenson Spark 或 On2 VP6 编解码器编码的 FLV 文件。与之前的技术相比，On2 VP6 视频编解码器使用的带宽较少，并且提供了额外的马赛克消除滤镜和色度去环滤镜。

VP6 编码器的基本原理，将很大的原始视频编码成很小的视频文件，其核心就是去除掉视频文件中的冗余信息——空间冗余、时间冗余、编码冗余、视觉冗余、知识冗余。

VP6 主要采用了 H.264 标准，其编码过程包含以下几点：

1) 帧内预测编码

帧内编码主要适用于减少图片的空间冗余，在给定帧中充分利用了相邻宏块的空间相关性，对于一个给定的宏块进行编码时，可以根据周围的宏块预测，然后对预测值和实际值的差值进行编码。

2) 帧间预测编码

帧间预测编码主要利用了连续帧中的时间冗余来进行运动估计和补偿。

- 3) 整数变换
- 4) 量化
- 5) 熵编码

ActionScript 中提供了 NetStream 类来进行播放 FLV 文件的流，并提供了类 VP6VideoStreamSettings，可以为每一个 NetStream 设置指定视频压缩设置，来实现直播编码的实现

4.2 服务器推流及客户端接收的实现

在推流方面使用了 ActionScript 来处理 RTMP 流, ActionScript 中有已经封装好的流文件和现成的接口函数，可以用于直接调用。

ActionScript 中进行客户端推送 RTMP:

- 1) 先建立网络连接，客户端连接上流媒体服务器

```
nc = new NetConnection()
nc.connect("rtmp://[流媒体服务器的 ipv6 地址]:1935/live")
```

- 2) 建立网络流

```
ns = new NetStream(nc)
```

- 3) 添加摄像头与音频

```
ns.attachCamera(cam) // 添加摄像头， 可以关于 cam 进行参数设置
```

```
ns.attachAudio(mic) // 添加音频， 可以关于 mic 进行参数设置
```

- 4) 网络流的上传

```
ns.publish("直播间名", "live")
```

Actionscript 中进行客户端视频播放:

- 1) 先建立网络连接，客户端连接上流媒体服务器

```
nc = new NetConnection()
nc.connect("rtmp://[流媒体服务器的 ipv6 地址]:1935/live")
```

- 2) 建立网络流

```
ns = new NetStream(nc)
```

3) 播放网络流

`ns.play("想要观看的直播间名")` // 即可进行观看

4.3 在线聊天室的实现

在实现聊天室的过程中遇上了两个难点。

1) http 协议带来的局限性

HTTP 协议有一个缺陷：通信只能由客户端发起。这种单向请求的特点，注定了如果服务器有连续的状态变化，客户端要获知就非常麻烦。

因此我们在实现聊天室时采用了 websocket 协议。可以通过类似于 java 中 socket 的方式进行网络间的通信。

2) 唯一的 websocket 通信地址，无限分隔的直播间。

websocket 的服务器地址通常都只有一个，而我们需要达到的效果是每个独立的直播间应有自己独立的聊天室，彼此之间互不干扰。

为此，我们设计了一套独立的信息筛选拆分规则。用户在发送信息时会将自己的 id 及房间号的 roomid 通过用户不可见的方式隐藏在发送的信息中。实际上每个直播间内收到的信息都是一样的，即所有用户发送的消息。但在 jsp 页面中，我们通过自己设定的规则，将所有信息通过之前提到的 id 和 roomid，筛选所需要在本页面显示。其次，我们在 websocket 协议通过 servlet 过滤器机制加了一层包装，能够在服务器层先接收到用户所发送的信息，对信息进行一些基本的筛选，比如：过滤不文明发言等。也可用于后期通过统计用户在课堂上的发言内容进行分析以拓展其它功能。

4.4 课堂管理的实现

1) 课程文件的存储

对于文件的储存方式，我们不仅仅要考虑到如何存放能使查找文件速度最快，也应设计好文件名的命名规则从而避免文件名重复而导致文件覆盖的问题。

在文件储存上，我们将课程相关文件均存在 course 文件夹下。对于每一堂课，我们根据其课程号 courseId 唯一创建了一个文件夹。文件名命名时，我们去当前时间精确到毫秒作为其文件名并在数据库中将其原文件名和存放文件名进行一一对应的存储。

当所需要通过程序获取到文件时，我们通过文件的 id 可以获取到对应的存储位置和实际名字。从而实现了平台文件的管理。

4.5 网页设计

网站整体使用 JSP 进行开发，前端页面由 HTML + javascript 编写，基于 jquery 开发并使用 bootstrap 框架构建页面。

前端搭建遵循开发规范：统一使用 UTF-8 编码格式；采用树状目录结构，针对不同功能模块，建立不同子目录；文件名及页面 id 遵循命名要求，方便后期调试修改。

针对页面设计，亦云网站整体以代表积极、主动、开放的红色为主色调，辅以暖橙色，力求给用户带来良好的使用体验。

首页设计分为三大模块，页头导航栏、进行经典课程展示的轮播模块、热门文章、话题、讨论的展示模块、课程推荐模块。页头导航栏保证用户无论在哪个界面都可快速返回首页、进入个人中心、进行查询搜索等功能。其他各模块则会引导用户体验网站的不同功能。