* **Project Title**

**iCourse网络课程平台**

* **Team Information**

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 职责 |
| 袁阳阳 | 141250179 | 需求分析人员，文档编写 |
| 袁楚宏 | 141250178 | 架构设计，开发人员 |
| 王子安 | 141250146 | 架构设计，开发人员 |
| 徐江河 | 141250161 | 架构设计，测试人员 |
| 董轶波 | 131250171 | 架构设计，开发人员 |
| 张鑫龙 | 141250195 | 架构设计，测试人员 |
| 张云飞 | 141250197 | 架构设计，测试人员 |
| 张诗琦 | 141250190 | 需求分析人员，文档编写 |

* **Project Overview**

**主要功能：**

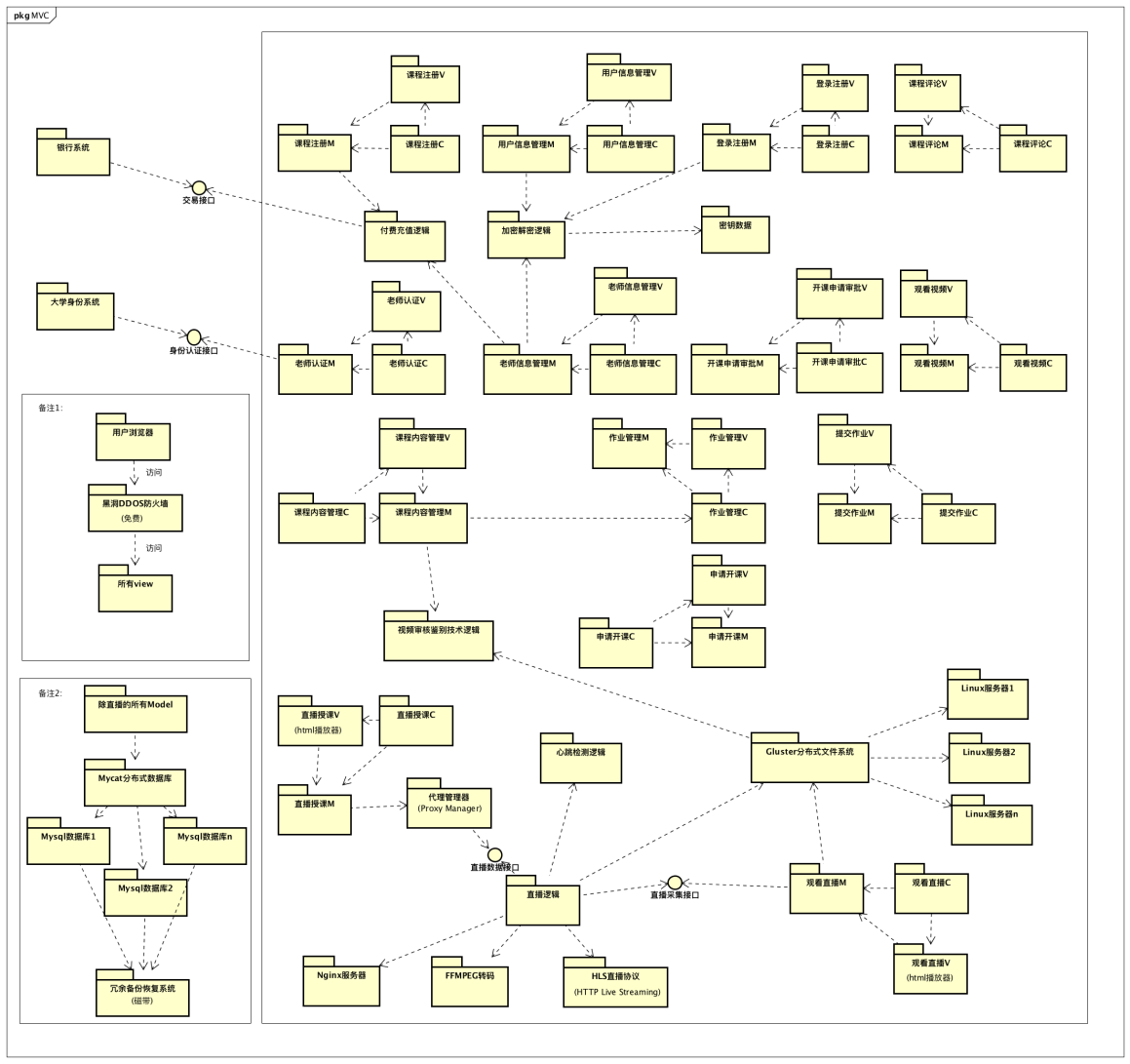
1. 教师功能：认证资格，申请开课，账户管理，课程管理，直播授课，作业管理
2. 学生功能：注册课程，观看课程视频，下载课程视频，发表评论，提交作业，观看直播，个人账户管理
3. 系统管理员功能：审批申请（教师开课审批，视频审批）
4. 系统其他功能：视频自动审核，教师身份验证

**预期输出：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 系统名称 | 子系统 | 预期输出 | 备注 |
| 1 | 学生系统 | 课程报名 | 报名成功/未成功 | 对于某些需要收费的课程，学生需要缴纳费用后方能报名成功 |
| 2 | 学生系统 | 在线支付 | 支付成功/失败 |  |
| 3 | 学生系统 | 观看视频 | 课程视频 |  |
| 4 | 学生系统 | 下载视频 | 下载成功/失败 |  |
| 5 | 学生系统 | 课程评论 | 发表/回复评论 |  |
| 6 | 学生系统 | 作业提交 | 提交成功/失败 |  |
| 7 | 学生系统 | 个人信息管理 | 个人信息 | 包含学生课程信息和账户信息 |
| 8 | 学生系统 | 观看直播 | 直播视频/直播中断 |  |
| 9 | 管理员系统 | 处理开课申请 | 同意/不同意教师开课 | 不同意开课需给出驳回理由 |
| 10 | 审核认证系统 | 视频自动审核 | 视频健康允许播出/  视频有害禁止播出 |  |
| 11 | 审核认证系统 | 教师身份认证 | 认证成功/失败 | 系统对教师上传的证件信息进行认证 |
| 12 | 教师系统 | 资格认证 | 认证成功/认证失败 | 教师向系统上传证件信息，提出认证请求 |
| 13 | 教师系统 | 申请开课 | 申请成功/申请失败 |  |
| 14 | 教师系统 | 课程管理 | 课程设置更改成功/不成功 | 教师可以向系统添加与课程有关的资料等 |
| 15 | 教师系统 | 账户管理 | 账户设置更改成功/不成功 |  |
| 16 | 教师系统 | 直播授课 | 直播授课成功/不成功 |  |
| 17 | 教师系统 | 作业管理 | 作业设置更改成功/不成功 | 教师可以布置作业、测试以及批改学生作业并给出反馈 |

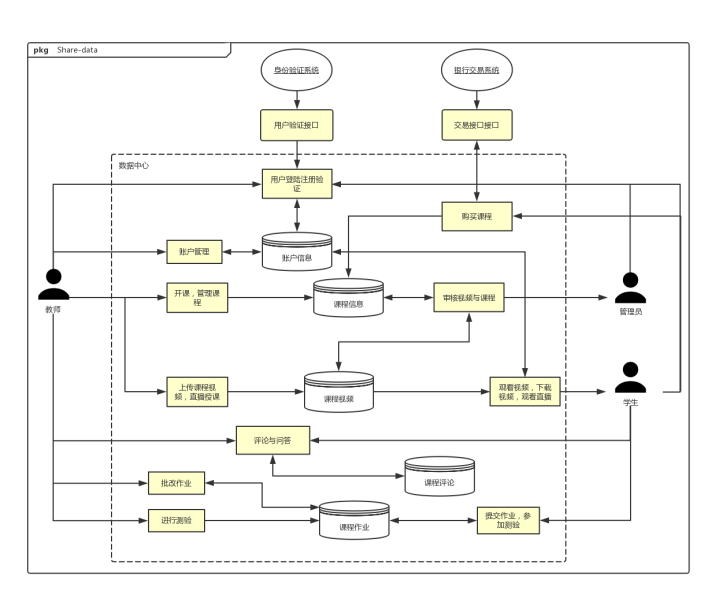
* **Project Design**

1. **MVC**



优点：  
1. 模块之间具有灵活的通信关系，系统需要为教师、学生推送通知，有利于信息传递的实现。  
2. 目前已经具有较多成熟的MVC的框架可供参考，在许多项目中有不错的实践  
3. 松散耦合，有利于网课系统各个模块独立开发，具有较高的可修改性，能够快速开发

1. **Shared-Data**

****

优点：

1. 网课系统用户量大、数据流量大，对于数据消费模块质量要求高。数据共享架构的数据消费单元较为独立，易于测试。
2. 数据读写分离，有利于老师相关功能模块（数据生产者）与学生功能模块（数据消费者）独立进行演化，避免耦合影响。
3. 具有高可用性和数据横向可扩展性。

* **Project Implementation Plan**

作为一个规模中等的B/S项目，本系统此采用PHP实现，开发平台为PhpStorm，实现过程中重点注意一下几点：

需要使用的外部接口、服务和现成系统有：身份验证系统、银行及平台支付系统、使用HLS直播服务器实现直播功能，HLS技术成熟，容易使用，开源，适合本项目需要。

实现过程中对于可用性、性能和安全性要注意：

1. 当有热门课程开始报名时出现较多同学“抢”课程的情况，对系统的可靠性有较高要求。因此，对于注册退选功能设置备份，设置心跳机制检测主模块，心跳中断则切换至备用模块。另一方面，所有数据有冗余备份，以防意外，也便于崩溃发生后的恢复。
2. 课程直播时请求量大，流量大，用proxyManager统一管理接入点资源，对客户端屏蔽接入点细节。
3. 课程视频及内容存储需要大量空间，处于性价比考虑，需要实现分布式数据库服务器。
4. 用户及账号信息经过加密解密模块，提供防火墙黑洞DDOS攻击