iCourse网络课程平台项目报告

目录

[**1.** **主要功能点和操作场景分析** 3](#_Toc476664331)

[**1.1.** **主要功能点** 3](#_Toc476664332)

[**1.2.** **操作场景分析** 3](#_Toc476664333)

[**1.2.1.** **教师功能** 3](#_Toc476664334)

[**1.2.2.** **学生功能** 5](#_Toc476664335)

[**1.2.3.** **管理员功能** 7](#_Toc476664336)

[**1.2.4.** **系统其他功能** 7](#_Toc476664337)

[**2.** **两种架构选择及对应的架构图设计** 8](#_Toc476664338)

[**2.1.** **MVC架构** 8](#_Toc476664339)

[**2.1.1.** **模块视图** 8](#_Toc476664340)

[**2.1.2.** **组件-连接件视图** 10](#_Toc476664341)

[**2.2.** **data-shared架构** 10](#_Toc476664342)

[**2.2.1.** **模块视图** 10](#_Toc476664343)

[**2.2.2.** **组件-连接件视图** 11](#_Toc476664344)

[**3.** **非功能需求及ASR描述** 12](#_Toc476664345)

[**3.1.** **可用性** 12](#_Toc476664346)

[**3.2.** **性能** 13](#_Toc476664347)

[**3.2.1.** **性能场景1：负载** 13](#_Toc476664348)

[**3.2.2.** **性能场景2：容量** 13](#_Toc476664349)

[**3.2.3.** **性能场景3：实时性** 13](#_Toc476664350)

[**3.3.** **易用性** 14](#_Toc476664351)

[**3.4.** **安全性** 14](#_Toc476664352)

[**3.5.** **互操作性** 14](#_Toc476664353)

[**3.6.** **可修改性** 15](#_Toc476664354)

[**3.7.** **可维护性** 15](#_Toc476664355)

[**4.** **系统类图设计** 15](#_Toc476664356)

[**5.** **组件/连接件到实现类的映射** 15](#_Toc476664357)

[**6.** **两种结构的比较及最终选择** 15](#_Toc476664358)

[**6.1.** **整体比较** 15](#_Toc476664359)

[**6.2.** **基于系统实际的最终选择** 17](#_Toc476664360)

[**7.** **具体实现技术的比较及最终选择** 17](#_Toc476664361)

[**8.** **基于MVC架构的ADD过程** 17](#_Toc476664362)

[**8.1.** **第一次迭代结果** 17](#_Toc476664363)

[**8.2.** **第二次迭代过程** 18](#_Toc476664364)

[**8.2.1.** **识别所选模块的ASR** 18](#_Toc476664365)

[**8.2.2.** **每个ASR可选的设计决策** 19](#_Toc476664366)

[**8.2.3.** **设计决策的选择及分析** 20](#_Toc476664367)

[**8.2.4.** **第二次迭代结果** 21](#_Toc476664368)

[**8.3.** **第三次迭代过程** 21](#_Toc476664369)

[**8.3.1.** **识别所选模块的ASR** 21](#_Toc476664370)

[**8.3.2.** **每个ASR可选的设计决策** 22](#_Toc476664371)

[**8.3.3.** **设计决策的选择及分析** 22](#_Toc476664372)

[**8.3.4.** **第三次迭代结果** 23](#_Toc476664373)

[**8.4 第四次迭代过程** 24](#_Toc476664374)

[**8.4.1** **识别所选模块的ASR** 24](#_Toc476664375)

[**8.4.2** **每个ASR可选的设计决策** 24](#_Toc476664376)

[**8.4.3** **设计决策的选择及分析** 25](#_Toc476664377)

[**8.4.4** **第四次迭代结果** 26](#_Toc476664378)

[**9.** **基于Shared-Data架构的ADD过程** 26](#_Toc476664379)

[**9.1.** **第一次迭代过程** 26](#_Toc476664380)

[**9.2.** **第二次迭代过程** 26](#_Toc476664381)

[**9.2.1.** **识别所选模块的ASR** 26](#_Toc476664382)

[**9.2.2.** **每个ASR可选的设计决策** 27](#_Toc476664383)

[**9.2.3.** **设计决策的选择及分析** 27](#_Toc476664384)

[**9.2.4.** **第二次迭代结果** 28](#_Toc476664385)

[**9.3.** **第三次迭代过程** 29](#_Toc476664386)

[**9.3.1.** **识别所选模块的ASR** 29](#_Toc476664387)

[**9.3.2.** **每个ASR可选的设计决策** 29](#_Toc476664388)

[**9.3.3.** **设计决策的选择及分析** 30](#_Toc476664389)

[**10.** **ATAM分析过程** 31](#_Toc476664390)

[**11.** **挑战和经验** 31](#_Toc476664391)

[**12.** **组员和分工** 32](#_Toc476664392)

1. **主要功能点和操作场景分析**
   1. **主要功能点**

* 教师功能
* 资格认证
* 开课申请
* 账户管理
* 课程管理
* 直播授课
* 作业管理
* 学生功能
* 课程注册
* 观看课程视频
* 发表评论
* 提交作业
* 观看直播
* 个人信息管理
* 管理员功能
* 教师开课审批
* 系统其他功能
  + 视频自动审核
  + 教师身份审核
  1. **操作场景分析**
     1. **教师功能**
        1. 资格认证

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师用户请求进行资格认证 |
| 响应 | 系统显示认证资料填写界面 |
| 刺激 | 用户填写进行资格认证所需资料并提交 |
| 响应 | 系统将资料自动给后台认证模块，显示申请成功，待审批 |
| 刺激 | 用户填写资格认证所需资料不完整 |
| 响应 | 系统提示用户完成填写所需资料 |

* + - 1. 开课申请

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师用户请求新开一门课程 |
| 响应 | 系统显示开课申请资料填写界面 |
| 刺激 | 教师用户填写资料 |
| 响应 | 系统保存资料并向管理员发送开课申请，提示用户申请已发出，等待审批 |
| 刺激 | 教师用户填写的资料不完整 |
| 响应 | 系统提示将资料填写完整 |

* + - 1. 账户管理

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师用户进入个人账户管理 |
| 响应 | 系统显示账户的详细信息，包括余额等 |
| 刺激 | 教师用户选择转出账户余额 |
| 响应 | 系统显示与该账户绑定的支付宝、微信账户列表 |
| 刺激 | 教师用户选择要转出的账户，选择转出 |
| 响应 | 系统提示教师用户输入支付密码 |
| 刺激 | 教师用户输入支付密码 |
| 响应 | 若密码正确，则完成支付；若密码错误则提醒再次输入 |
| 刺激 | 教师用户请求添加绑定账户 |
| 响应 | 系统显示账户资料填写界面 |
| 刺激 | 教师用户进行资料填写 |
| 响应 | 系统完成绑定并保存信息 |
| 刺激 | 教师用户要求修改支付密码 |
| 响应 | 系统提示用户输入原支付密码 |
| 刺激 | 教师用户输入原支付密码 |
| 响应 | 若原支付密码输入正确，则提示用户输入新支付密码；否则提示用户密码错误，返回上一级界面 |
| 刺激 | 教师用户输入新支付密码 |
| 响应 | 系统保存新的支付密码，提示用户修改成功 |

* + - 1. 课程管理

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师用户请求查看授课列表 |
| 响应 | 系统显示授课列表 |
| 刺激 | 教师用户请求对某一课程进行管理 |
| 响应 | 系统进入课程管理界面 |
| 刺激 | 教师用户选择添加课程内容，如文档、视频等 |
| 响应 | 系统提示教师用户上传 |
| 刺激 | 教师上传文档、视频等课程内容 |
| 响应 | 系统保存上传的内容并提示选择该课程的用户有新添加内容 |
| 刺激 | 教师用户选择删除课程内容 |
| 响应 | 系统显示课程内容列表 |
| 刺激 | 教师用户选择要删除的内容并删除 |
| 响应 | 系统提示是否确定删除 |
| 刺激 | 用户确定删除 |
| 响应 | 系统进行删除并返回显示课程内容列表 |
| 刺激 | 用户取消删除 |
| 响应 | 系统显示课程内容列表 |

* + - 1. 直播授课

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师用户请求查看授课列表 |
| 响应 | 系统显示授课列表 |
| 刺激 | 教师用户选择某一课程 |
| 响应 | 系统显示该课程详细信息页面 |
| 刺激 | 教师用户选择发布直播授课通知 |
| 响应 | 系统提示教师用户填写直播授课时间等信息 |
| 刺激 | 教师用户填写直播授课的信息 |
| 响应 | 系统保存直播通知并提醒选择该课程的用户直播时间 |
| 刺激 | 教师用户开始直播授课 |
| 响应 | 系统输出直播授课内容并对内容进行缓存 |

* + - 1. 作业管理

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师用户请求查看授课列表 |
| 响应 | 系统显示授课列表 |
| 刺激 | 教师用户请求对某一课程进行作业管理 |
| 响应 | 系统进入作业管理页面 |
| 刺激 | 教师用户选择添加新的作业 |
| 响应 | 系统进入作业内容填写页面 |
| 刺激 | 教师用户填写作业内容，如作业标题、截止时间、提交格式、作业题目等 |
| 响应 | 系统保存作业内容并提示选择该课程的用户有新的作业，返回作业管理页面 |
| 刺激 | 教室用户选择添加新的测试 |
| 响应 | 系统进入测试内容填写页面 |
| 刺激 | 教室用户填写测试内容，如测试标题、测试时间、测试题目等 |
| 响应 | 系统保存测验内容并提示选择该课程的用户有新的测验，返回作业管理界面 |
| 刺激 | 教师用户选择进行作业批改 |
| 响应 | 系统显示该课程已经截止提交的作业列表 |
| 刺激 | 教师用户选择要进行批改的某次作业 |
| 响应 | 系统显示该次作业下还未批改的作业列表 |
| 刺激 | 教师用户选择某一个作业 |
| 响应 | 系统显示该提交作业的详细内容 |
| 刺激 | 教师用户填写分数 |
| 响应 | 系统保存分数并提醒提交该作业的用户最新作业批改情况，若该次作业仍有未批作业则返回未批改的作业列表，否则提醒教师本次作业已经批改完成，返回已经截止提交的作业列表 |

* + 1. **学生功能**
       1. 课程注册

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 用户请求查看课程列表 |
| 响应 | 系统显示课程列表 |
| 刺激 | 用户请求查看某一课程详细信息 |
| 响应 | 系统显示该课程详细信息页面 |
| 刺激 | 用户请求注册某一课程 |
| 响应 | 系统显示课程注册成功 |
| 刺激 | 用户请求注册的课程需要收费 |
| 响应 | 系统转向付费页面 |
| 刺激 | 用户完成付费 |
| 响应 | 系统显示课程注册成功 |
| 刺激 | 用户取消付费 |
| 响应 | 系统转向课程信息页面 |

* + - 1. 观看课程视频

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 用户请求播放课程视频 |
| 响应 | 系统开始播放课程视频 |
| 刺激 | 用户请求暂停课程视频 |
| 响应 | 系统暂停课程视频 |
| 刺激 | 用户观看完本节课视频 |
| 响应 | 系统提示即将自动播放下一节课程 |
| 刺激 | 用户未看完本节课程视频就退出播放 |
| 响应 | 系统保存用户观看进度 |

* + - 1. 发表评论

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 用户发表课程评论 |
| 响应 | 系统显示评论成功并添加该评论 |
| 刺激 | 用户删除课程评论 |
| 响应 | 系统删除该课程评论 |
| 刺激 | 用户回复他人的课程评论 |
| 响应 | 系统显示回复的课程评论 |
| 刺激 | 用户评论被他人回复 |
| 响应 | 系统提示用户收到回复 |
| 刺激 | 用户查看回复 |
| 响应 | 系统显示评论回复页面 |

* + - 1. 提交作业

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 用户请求完成在线作业 |
| 响应 | 系统转向作业页面 |
| 刺激 | 用户完成作业并请求提交 |
| 响应 | 系统显示作业已提交 |
| 刺激 | 作业批改完成 |
| 响应 | 系统通知用户作业已批改完成 |

* + - 1. 观看直播

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 用户查看已开始未结束或即将开始的直播课程列表 |
| 响应 | 系统显示已开始未结束或即将开始的直播课程列表 |
| 刺激 | 用户请求查看课程详细信息 |
| 响应 | 系统显示课程详细信息 |
| 刺激 | 用户要求设置直播开始提醒 |
| 响应 | 系统设置直播开始提醒 |
| 刺激 | 用户设置提醒时间到达 |
| 响应 | 系统向用户发送通知 |
| 刺激 | 用户请求观看直播课程 |
| 响应 | 系统播放直播课程 |

* + - 1. 个人信息管理

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 学生请求查看个人信息中心 |
| 响应 | 系统转向学生个人信息中心页面 |
| 刺激 | 学生请求查看付费记录 |
| 响应 | 系统显示付费记录 |
| 刺激 | 学生请求查看正在学习的课程列表 |
| 响应 | 系统显示正在学习的课程列表和学习进度 |
| 刺激 | 学生请求查看已学习课程列表 |
| 响应 | 系统显示已学习课程列表 |

* + 1. **管理员功能**
       1. 教师开课审批

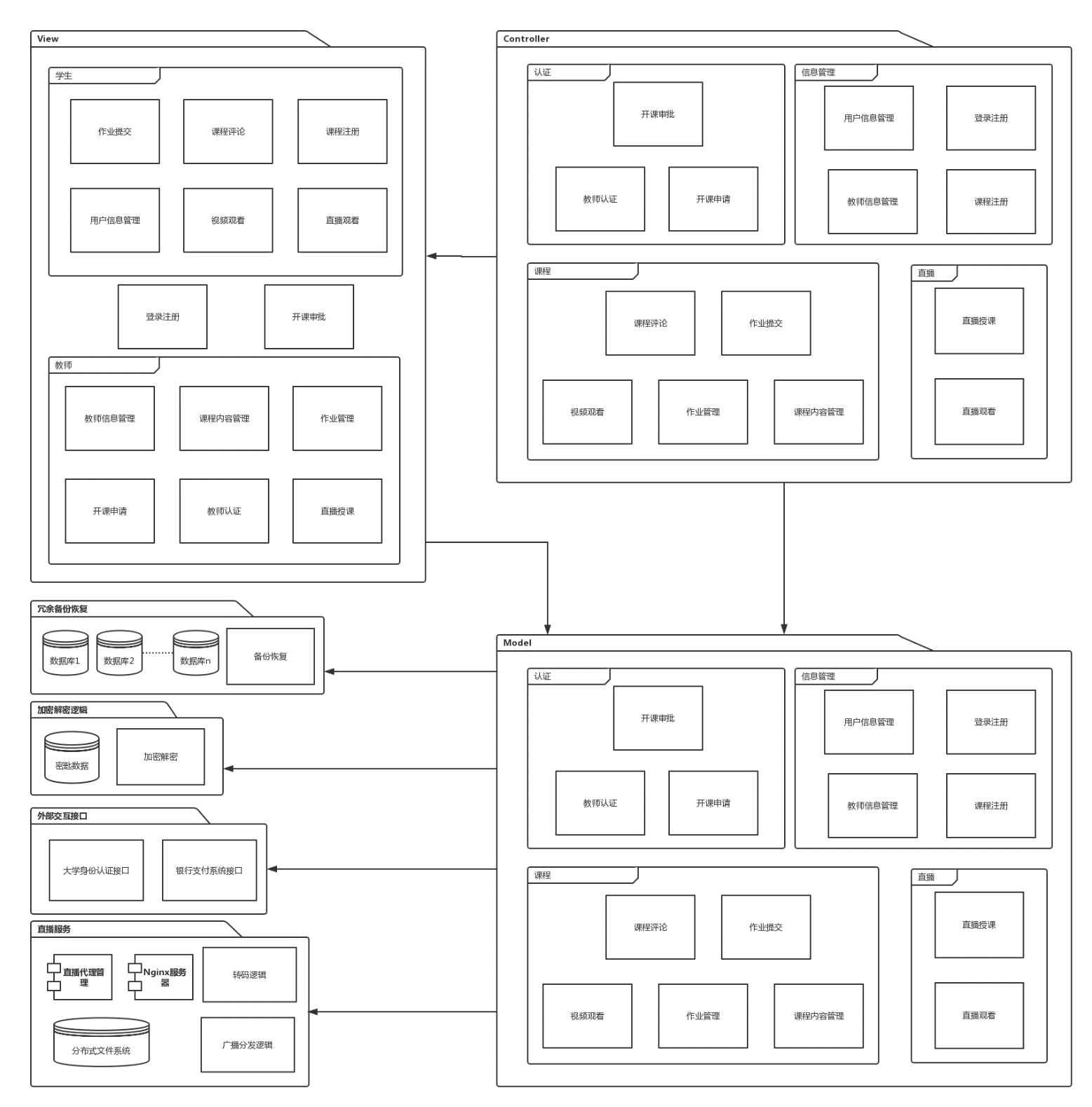
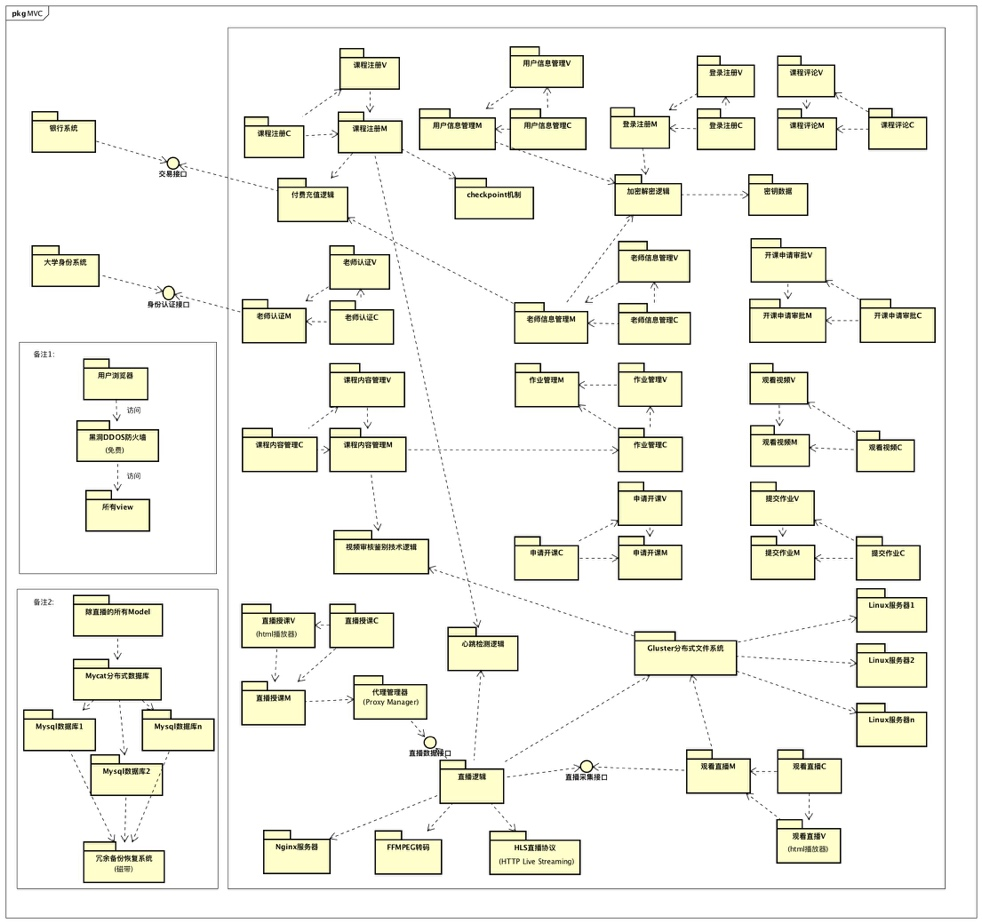
|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师提交开课申请 |
| 响应 | 系统提示管理员收到开课申请 |
| 刺激 | 管理员审批开课申请通过 |
| 响应 | 系统通知相关教师申请通过 |
| 刺激 | 管理员审批开课申请不通过 |
| 响应 | 系统提示输入拒绝理由 |
| 刺激 | 管理员填写完拒绝理由并确定 |
| 响应 | 系统通知相关教师申请驳回 |

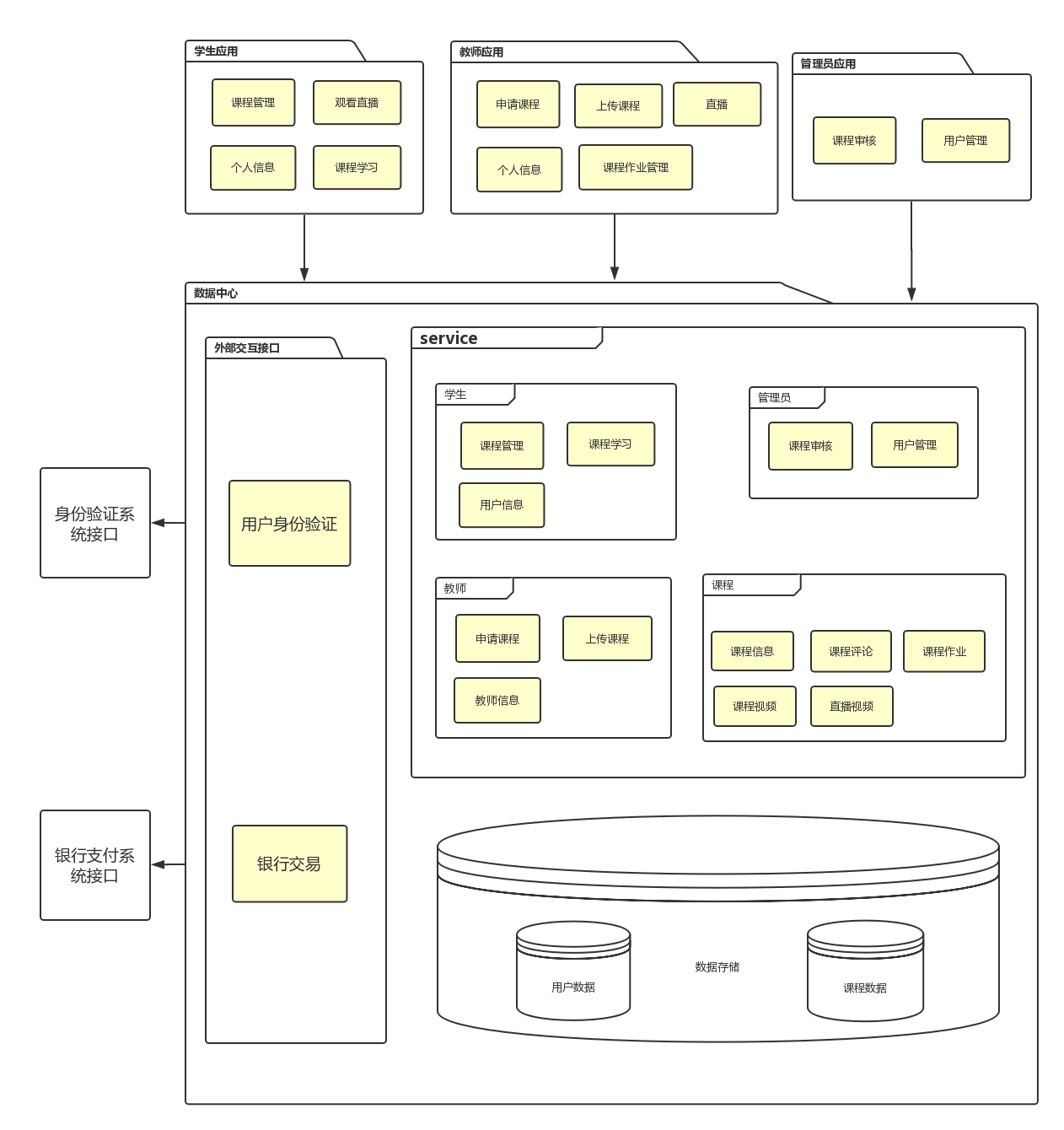
* + 1. **系统其他功能**
       1. 视频自动审核

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师给课程添加新视频内容 |
| 响应 | 系统提示收到新的课程视频 |
| 刺激 | 系统自动审批视频内容通过 |
| 响应 | 系统将视频内容显示在对应的课程内容里 |
| 刺激 | 系统自动审批视频内容不通过 |
| 响应 | 系统通知相关教师新添加的课程视频未通过审核 |

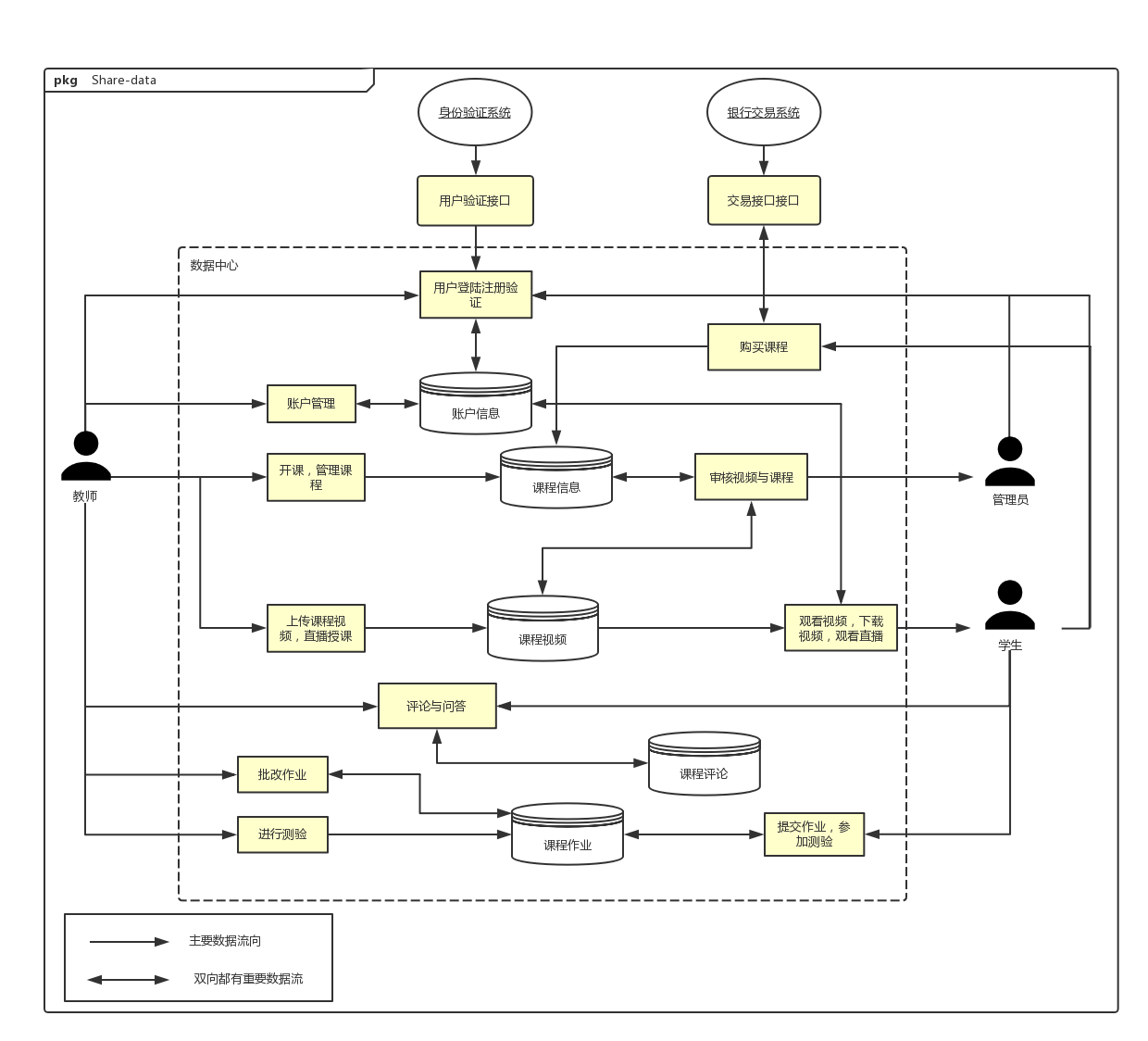
* + - 1. 教师身份审核

|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师提交认证申请 |
| 响应 | 系统提示收到新的教师认证申请 |
| 刺激 | 系统审批教师认证通过 |
| 响应 | 系统通知相关教师申请通过 |
| 刺激 | 系统审批教师认证申请不通过 |
| 响应 | 系统通知相关教师申请驳回 |

1. **两种架构选择及对应的架构图设计**
   1. **MVC架构**
      1. **模块视图**
      2. **组件-连接件视图**
   2. **data-shared架构**
      1. **模块视图**

****

* + 1. **组件-连接件视图**

****

1. **非功能需求及ASR描述**
   1. **可用性**

scenario1

|  |  |
| --- | --- |
| 场景组成部分 | 可能的值 |
| 源 | 使用系统的学生、教师、管理员等用户角色。 |
| 刺激 | 用户通过网页浏览器访问网站，使用系统的各功能。 |
| 制品 | 整个软件系统 |
| 环境 | 网站日常运行时，服务器、网络环境稳定，负载不超过设计上限。 |
| 响应 | 网站能够持续提供服务。 |
| 响应度量 | 1. 每周网站服务下线总持续时间不超过5分钟，时间段不出现在服务繁忙期（8:00AM~12:00PM） 2. 数据库出现服务故障之后能在5s内开始启用备份数据库，维持视频、个人信息服务等，不影响用户观看非直拨课程 3. 系统能够在直播课程的教师断线之后，5s内探测到异常，并在10s内通知直播收看用户直播者掉线的信息 |

* 1. **性能**
     1. **性能场景1：负载**

scenario2

|  |  |
| --- | --- |
| 场景组成部分 | 可能的值 |
| 源 | 使用系统的学生/教师 |
| 刺激 | 学生观看课程视频/观看直播/提交课程作业 |
| 制品 | 课程视频子系统/直播子系统/课程作业管理子系统 |
| 环境 | 运行状态 |
| 响应 | 系统正常相应每个用户的请求 |
| 响应度量 | 1. 系统在6000个用户并发访问的情况下，能够正常完成每个用户的请求 2. 用户访问请求的平均等待时间要低于1.5秒 3. 用户访问请求的最长等待时间不能超过3.0秒 |

* + 1. **性能场景2：容量**

scenario3

|  |  |
| --- | --- |
| 场景组成部分 | 可能的值 |
| 源 | 系统 |
| 刺激 | 系统希望进行课程数据（视频、文档、作业、评论等）的存储 |
| 制品 | 存储课程数据的系统数据库 |
| 环境 | 运行状态 |
| 响应 | 系统正确地、完整地、一致地存储大量课程数据 |
| 响应度量 | 1. 系统能存储至少100T的课程数据，并且保证数据的一致性、正确性和完整性 2. 系统能缓存直播内容至少20T、缓存时间至少为2天，并且保证数据的一致性、正确性和完整性 |

* + 1. **性能场景3：实时性**

scenario4

|  |  |
| --- | --- |
| 场景组成部分 | 可能的值 |
| 源 | 访问系统的学生用户 |
| 刺激 | 用户希望在iCourse系统上观看课程直播 |
| 制品 | 系统的课程直播子系统 |
| 环境 | 联网状态 |
| 响应 | 系统向用户播放直播视频 |
| 响应度量 | 1. 用户可以在5秒内打开直播视频 2. 直播延迟不超过15s |

* 1. **易用性**

scenario5

|  |  |
| --- | --- |
| 场景组成部分 | 可能的值 |
| 源 | 使用系统的用户 |
| 刺激 | 用户在观看直播时 |
| 制品 | 系统界面/用户接口 |
| 环境 | 系统运行/用户观看直播时 |
| 响应 | 系统提供高品质、高清晰度、低卡顿的直播画面 |
| 响应度量 | 1. 用户的满意度为99%及以上 |

* 1. **安全性**

scenario6

|  |  |
| --- | --- |
| 场景的部分 | 可能的值 |
| 源 | 经过/未经过本系统授权的个人或系统 |
| 刺激 | 试图添加/修改/删除数据，访问系统服务，窃取用户信息 |
| 制品 | 系统服务、个人信息数据、账户数据 |
| 环境 | 已经联网 |
| 响应 | 对用户进行验证，加密用户的账户信息，阻止未授权用户访问，对恶意修改的数据进行回滚 |
| 响应度量 | 1. 未授权的个人或系统不能访问相关的账户信息 2. 数据被恶意修改时可以在5min内实现恢复 |

* 1. **互操作性**

scenario7

|  |  |
| --- | --- |
| 场景组成部分 | 可能的值 |
| 源 | 系统/大学身份系统接口/银行系统接口/直播采集接口 |
| 刺激 | 系统与大学身份系统接口之间的数据交换/系统与银行系统接口  之间的数据交换/直播采集接口的信息交换/系统间的请求和响应 |
| 制品 | 希望进行数据交换、共享的上述实体 |
| 环境 | 系统运行时的状态模式 |
| 响应 | 数据交换请求被接受，并正常交换数据 |
| 响应度量 | 1. 数据正确交换的概率为99.9 及以上 2. 各大平台的API都能适应于系统，相互匹配 |

* 1. **可修改性**

scenario8

|  |  |
| --- | --- |
| 场景的部分 | 可能的值 |
| 源 | 系统开发人员 |
| 刺激 | 系统开发人员希望修改iCourse系统用户界面/数据标准/控制逻辑等 |
| 制品 | 代码/数据/配置文件等 |
| 环境 | 设计/开发/维护系统时 |
| 响应 | 需要修改的模块被正确的修改，最小化对其他模块的修改，并不影响其他功能的实现 |
| 响应度量 | 1. 每个模块的修改可以在2人月内完成 2. 修改预算不超过总预算的10% 3. 对其他模块的修改不超过10% 4. 不影响无关的系统功能 |

* 1. **可维护性**

scenario9

|  |  |
| --- | --- |
| 场景的部分 | 可能的值 |
| 源 | 系统开发者 |
| 刺激 | 系统开发人员希望对iCourse的数据进行维护，或者对系统进行变更/升级 |
| 制品 | 系统数据/各个功能模块 |
| 环境 | 系统维护时 |
| 响应 | 快速完成数据备份，准确、高效地完成系统变更 |
| 响应度量 | 1. 每次数据备份时间不超过2人日 2. 每个模块的修改时间不超过5人日 3. 变更后系统完成测试的时间不超过5人日 |

1. **系统类图设计**
2. **组件/连接件到实现类的映射**
3. **两种结构的比较及最终选择**
   1. **整体比较**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | data-shared | | MVC | |
| 比较方面 | **pros** | **cons** | **pros** | **cons** |
| 安全性 |  | 个人信息数据和视频数据集中存放。数据中心易成为被攻击的对象，发生单点失效的可能性更大 | 控制器对数据的控制比较分散，受到攻击后影响较小 |  |
| 可用性 | 数据比较集中心跳易于实现对整个数据中心的监测。  向外封装了主从数据库，易于完成错误恢复实现高可用性 |  |  | 数据分散，心跳对异常的监测较为困难 |
| 互操作性 | 没有明显区别 | | | |
| 可修改性 |  | 数据的消费者和生产者在数据方面紧密耦合不容易修改 | 模块内聚性强、耦合度低，容易修改而不影响其他模块 |  |
| 性能：负载 | 封装了数据的获取、存储操作，易于实现支持并发数据存取，保证系统在高负载情况下正常运行 | 数据中心架构不当易成为负载的瓶颈 |  |  |
| 性能：容量 | 没有明显区别 | | | |
| 性能：实时性 | 没有明显差别 | | | |
| 可维护性 | 视频资源、数据较为集中，对数据、资源的维护更加简单 |  | 模块内聚性强、耦合度低，易进行维护 |  |
| 可测试性 | 数据生产者和数据消费者之间的逻辑借助数据中心完成隔离，数据生产者和数据消费者的相应模块可以分别测试 |  | 模块内聚性强、耦合度低，桩测试和驱动测试易于实施 |  |
| 易用性 | 没有明显差别 | | | |
| 开发难度 |  | 框架资料少，无可借鉴成熟案例，框架搭建出现失误的可能性大 | 有较多成熟高效的MVC框架，开发速度快、开发成本低 |  |

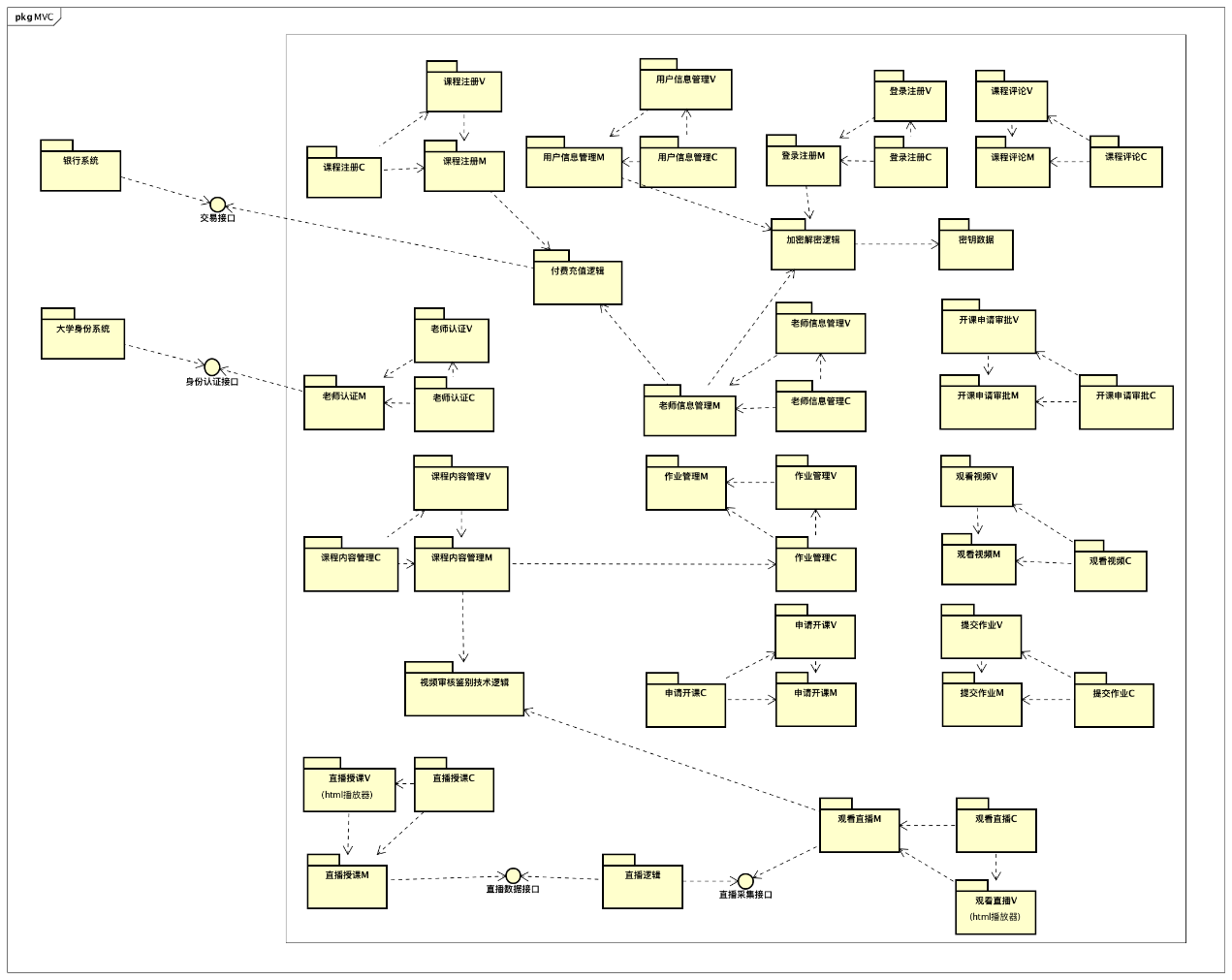
* 1. **基于系统实际的最终选择**

1. **具体实现技术的比较及最终选择**

|  |  |  |
| --- | --- | --- |
| 类别 | 实现方式 | 选择与解释 |
| 易用性保障 | 带宽检测 | 比较容易实现的技术，成本低 |
| 视频转码 | 将已经压缩编码的视频[码流](http://baike.baidu.com/view/1218710.htm)转换成另一个视频码流，以适应不同的网络带宽、不同的终端处理能力和不同的用户需求 |
| 安全保障 | 用户数据加密 | 成熟的安全手段，避免泄露用户隐私 |
| 自动攻击侦测 | 系统关系到老师用户的账户安全 |
| 防火墙 | 限制未知用户非法的访问 |
| 可用性措施 | Heartbeat检测 | 直播环节的实时性要求比较高，更适合于heartbeat,相比于ping-echo流量少，周期性，负载低的时候，也可以保证所有数据库的可用性 |
| Ping-echo | 持续观看视频的时候采取，实时确保可用性 |
| 服务器冗余 | 方便系统遇到故障后，能够及时地恢复 |
| 数据备份 | 对于静态视频课程资源（非直播课程），备份回滚可以保障资源不意外丢失；对于想要保存下来的直播视频，也可以实时保存，纳入静态资源中。 |
| 提高负载措施 | 反向代理服务器 | 有较为成熟的开发框架和技术支持,例如Nginx |
| CDN | 技术成熟 |
| 提高容量措施 | 数据库集群 | 通过数据库集群，提高系统存储数据的能力，同时对重要数据进行冗余备份。 |
| 分布式文件系统 | 通过分布式文件系统的负载均衡、分散存储解决大量大文件（视频）存取问题。 |
| 实时性措施 | 减少数据传输延迟 | 可以利用BGP中转架构或者客户端解析优化策略等技术减少延迟 |
| 直播协议 | 简单易于实现，RTP，HLS，RTMP等协议可供选择，可以在延迟和跨平台之间权衡 |
| 数据备份 | 磁带机 | 采用。成本低，技术成熟，操作简单，容量大。 |
| 双机热备 | 未采用。成本高，超出了系统对数据安全性的要求。 |

1. **基于MVC架构的ADD过程**
   1. **第一次迭代结果**

在第一次迭代中，在MVC架构的基础上，参考已经划分好的、较为清晰的功能性需求，系统完成了模块划分和整体的设计。



* 1. **第二次迭代过程**

在第二次过程中，选取系统中较为复杂、重要的直播模块进行细化和分解，以适应相应非功能属性（负载性能、实时性能、容量性能、易用性、互操作性、可用性）的质量要求。

* + 1. **识别所选模块的ASR**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Architectural Drivers | Importance | Difficulty |
| 1 | Scenario 2：负载性能 | high | high |
| 2 | Scenario 4：实时性能 | high | high |
| 3 | Scenario 3：容量性能 | high | high |
| 4 | Scenario 5：易用性 | high | medium |
| 5 | Scenario 7：互操作性 | high | medium |
| 6 | Scenario 1：可用性 | medium | medium |

* + 1. **每个ASR可选的设计决策**
       1. 负载性能

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| CDN | 技术成熟，尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快、更稳定 |
| 增加可用资源 | 使用速度更快的处理器、额外的处理资源、额外的内存、速度更快的网络 |
| 反向代理服务器 | 有较为成熟的开发框架和技术支持，目前用的比较多的是nginx反向代理服务器 |
| 维持多个副本 | 减少服务器上进行计算时的出现的资源争用，例如设置可设置告诉缓存 |

* + - 1. 实时性能

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| 使用RTMP(Real Time Messaging Protocol)协议 | 支持flashplayer可以支持直播播放，延迟一般在2~5秒，开源软件和开源库支持完整 |
| 使用HLS（HTTP Live Streaming）协议 | HTML5可以直接打开播放；这个意味着可以把一个直播链接通过微信等转发分享，不需要安装任何独立的APP，有浏览器即可，所以流行度很高。延时在5~7秒（可能大于10秒） |
| RTP（Real-time Transport Protocol）协议 | RTP在视频监控、视频会议、IP电话上有广泛的应用，实时性强。但国内各大CDN厂商没有支持基于RTP直播的 |
| HTTP-FLV协议 | 即使用HTTP协议流式的传输媒体内容。相对于RTMP，HTTP更简单和广为人知，内容延迟同样可以做到2~5秒，打开速度更快。因为没有繁杂的握手，就延迟看，优于RTMP |
| BGP中转架构 | 为直播发布和直播观看隶属不同网络运行商的情况构造最短传输路径，从而增强稳定性和减少延时 |
| 客户端解析优化策略 | 本机缓存域名的解析结果，对域名进行预解析，每次需要直播推流和播放的时候不再需要再进行DNS过程。此处节省几十到几百毫秒的打开延迟 |

* + - 1. 容量性能

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| Gluster分布式文件系统 | 开源的分布式文件系统，在扩展性、可靠性、性能、维护性等方面具有独特的优势 |
| Lustre分布式文件系统 | lustre是一个大规模的、安全可靠的，具备高可用性的集群文件系统 |
| HDFS分布式文件系统 | 错误检测和快速、自动的恢复是HDFS的核心架构目标，高吞吐量 |

* + - 1. 易用性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| 带宽检测 | 比较容易实现的技术，成本低 |
| FFMPEG视频转码 | 将已经压缩编码的视频[码流](http://baike.baidu.com/view/1218710.htm)转换成另一个视频码流，以适应不同的网络带宽、不同的终端处理能力和不同的用户需求 |

* + - 1. 互操作性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| 在提供的服务列表中定位所需要的服务 | 主要针对所需的服务较多且有层次性的情况 |
| 使用控制机制来管理、协调服务的调用 | 针对所要服务的访问需求量较多的情况 |
| 抽取并剪裁接口 | 针对请求响应交换的数据不一致情况 |

* + - 1. 可用性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| Heartbeat | 周期性的检测故障技术 |
| Ping-echo | 错误探测技术 |

* + 1. **设计决策的选择及分析**
       1. 负载性能

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 决策理由 |
| CDN | 采用。成熟的负载均衡技术，避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快更稳定，既可以提高网络流量，又可以提高网络的整体性能。 |
| 增加可用资源 | 不采用。增加成本，代价过高 |
| 反向代理服务器 | 采用。可使用Nginx反向代理服务器，成熟的技术支持 |
| 维持多个副本 | 不采用。同步与一致需要的成本过高 |

* + - 1. 实时性能

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| RTMP协议和HTTP-FLV协议 | 不采用。虽然这两种协议的延时很低，但需要依赖Flash播放，跨平台性低，降低用户体验，降低了可用性和易用性 |
| HLS协议 | 采用。虽然HLS协议的延时略高，但已能满足系统需求，且HLS协议跨平台性高，HTML5即可直接打开播放 |
| RTP协议 | 不采用。虽然采用RTP协议的延迟低，但是国内各大CDN厂商没有基于RTP直播的，实施难度过大。 |
| BGP中转架构 | 采用。可以在不同网络运营商之间构造最短路径，从而增强稳定性和减少延时 |

* + - 1. 容量性能

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| Gluster分布式文件系统 | 采用。POSIX访问接口，兼容性好。灵活实现实现高性能和高可用。支持FUSE。镜像冗余，自动故障恢复。开源，没有直接成本 |
| Lustre分布式文件系统 | 不采用。POSIX/MPI访问接口，但是没有冗余保护和故障恢复，且部署过于复杂 |
| HDFS分布式文件系统 | 不采用。私有访问接口，随机读写文件支持不足。多用户写和任意文件修改支持差。流式读取文件系统数据，利于视频缓冲。基于HA解决单点失效问题。有元数据，存在性能和容量扩展上限。 |

* + - 1. 易用性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| 带宽检测 | 采用。比较容易实现的技术，成本低 |
| FFMPEG视频转码 | 采用。适用于Windows和linux多平台，运用广泛 |

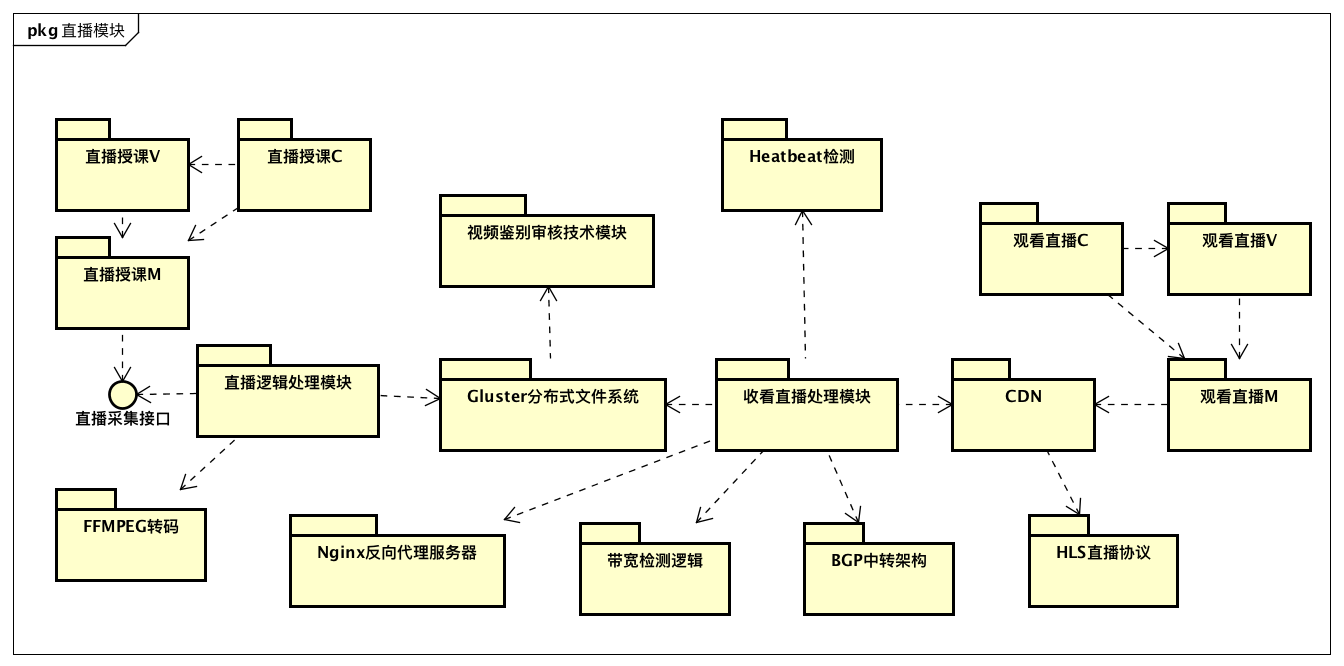
* + - 1. 互操作性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 决策理由 |
| 在提供的服务列表中定位所需要的服务 | 不采用。仅是关于直播的服务。 |
| 使用控制机制来管理、协调服务的调用 | 不采用。仅是关于直播的服务。 |
| 抽取并剪裁接口 | 采用。用于统一不同的数据格式。 |

* + - 1. 可用性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| Heartbeat | 采用。直播环节的实时性要求比较高，更适合于heartbeat,相比于ping-echo流量少，周期性，负载低的时候，也可以保证所有数据库的可用性 |
| Ping-echo | 不采用。对比于heartbeat需要更多的信息流量 |

* + 1. **第二次迭代结果**



* 1. **第三次迭代过程**

选取课程注册模块作为系统元素进行分解。在质量属性方面，该模块与负载性能、互操作性、安全性、实时性等关系密切。

* + 1. **识别所选模块的ASR**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Achitectural Drivers | Importance | Difficulty |
| 1 | Senario2: 性能负载，有大量学生同时对某一或某些课程进行注册 | High | High |
| 2 | Senario1: 可用性，大量学生访问系统时，系统要保持可用 | High | Medium |
| 3 | Senario7: 互操作性，对于需要付费的课程，学生注册时系统需要与银行系统交互 | High | Low |
| 4 | Senario6: 安全性，学生支付课程费用时，必须保障网络环境的安全 | High | Medium |

* + 1. **每个ASR可选的设计决策**
       1. 性能负载

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 负载均衡器 | 可以保证以相对较优的方式分配请求，但需要着重考虑成本问题 |
| 反向代理 | Nginx，具有高性能，轻量级的优势，处理高并发请求时有相当高的性能 |
| 内容分发网络CND | 可以根据用户的区域信息分配最近的CDN服务器，分解服务器的压力，但需要考虑CDN服务器的数据同步问题 |
| DNS负载均衡 | 由DNS服务器充当负载均衡调度器，将用户请求分散到多台服务器上，但是调试成本高，分配策略也有很强的局限性。 |
| 增加可用资源 | 有助于提高性能，但是需要考虑成本投入 |

* + - 1. 可用性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| Heartbeat检测 | 周期性地检测系统服务是否可用 |
| Ping-echo | 客户端发起请求，检测系统服务是否可用 |
| 主动冗余 | 所有冗余组件以并行方式对事件作出响应，如果发生错误，系统可以快速恢复 |
| 被动冗余 | 主要组件对事件作出响应，并通知其他组件进行必要的更新，如果发生错误，系统可以在一定时间内恢复 |

* + - 1. 可操作性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 在服务列表中定位所需服务 | 当所需服务较多时可以快速定位所需服务 |
| 选取和剪裁外部系统的接口 | 外部系统提供的接口可能有很多，有些接口返回的数据可能过多，并不是系统所需要的 |
| 提供多种数据格式 | 多种数据格式来保证系统可以使用不同的外部服务，以防出现数据不匹配的问题 |

* + - 1. 安全性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 用户数据传输加密 | 加密用户数据以防被盗用，保障财产安全 |
| 自动攻击检测 | 针对网络攻击进行检测，但是会增加系统运行负担 |
| 拒绝可疑访问 | 可以较好地保证访问的合法性，但是可能会引起误伤 |

* + 1. **设计决策的选择及分析**
       1. 性能负载

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 负载均衡器 | 不采用，第三方软件虽然可以保证以相对较优的方式分配请求，但是成本过高。 |
| 反向代理 | 采用，高性能，轻量级，拥有众多请求分配策略，有助于处理高并发请求且技术成熟。 |
| 内容分发网络CND | 采用，可以很好的分担服务器的压力，且国内各大CND服务商发展成熟。 |
| DNS负载均衡 | 不采用，分配策略具有局限性，无法根据实际服务器的实时负载差异来调整调度策略，记录缓存过于复杂，并不能很好地完成工作量分配。 |
| 增加可用资源 | 不采用，主要是成本过高。 |

* + - 1. 可用性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| Heartbeat检测 | 采用，周期性检查可以及时检测服务宕机以采取措施 |
| Ping-echo | 不采用，系统服务不可用时可能不能及时检测 |
| 主动冗余 | 不采用，多个备件并发对事件作出响应，需要系统提供更多资源，会增加成本 |
| 被动冗余 | 采用，课程注册虽然要求很高的可用性，但由于课程名额无限制，用户更关系自己是否报名成功，在一定时间内恢复系统是可以接受的。 |

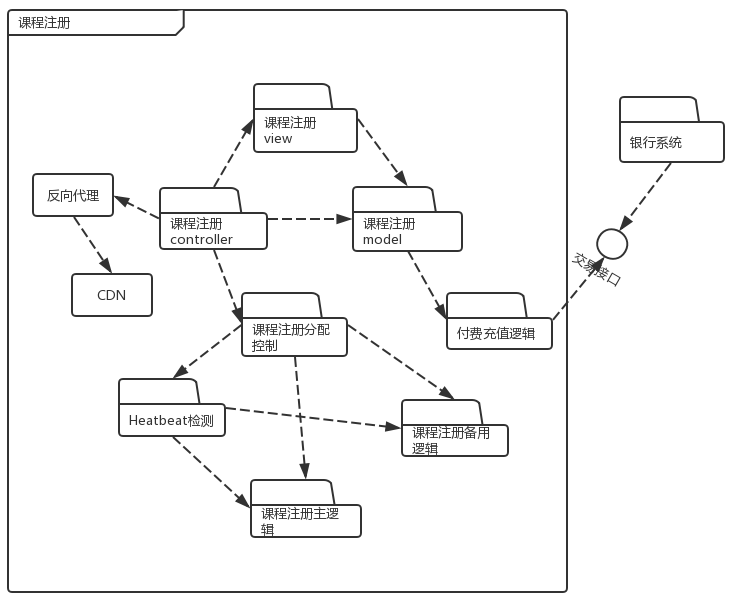
* + - 1. 可操作性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 在服务列表中定位所需服务 | 不采用，这里只用到外部系统的支付服务，并不是应对多种服务需求。 |
| 选取和剪裁外部系统的接口 | 采用，选取需要使用的支付服务接口，对接口重新剪裁有助于获取更加明确的支付信息。 |
| 提供多种数据格式 | 采用，不同的支付服务可能采用不同的数据格式，提供多种数据格式有利于与多个外部服务的适配 |

* + - 1. 安全性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 用户数据传输加密 | 采用，加密用户数据以防被盗用，保障财产安全 |
| 自动攻击检测 | 不采用，加重了系统的运行负担，且第三方支付服务本身对安全性也有较好地保障 |
| 拒绝可疑访问 | 不采用，可能会误伤正常用户的访问，降低用户体验和系统的可用性 |

* + 1. **第三次迭代结果**



**8.4 第四次迭代过程**

选取老师账户管理模块作为系统元素进行分解。在质量属性方面，该模块与互操作性、安全性、可用性，可修改性等关系密切。

* + 1. **识别所选模块的ASR**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Achitectural Drivers | Importance | Difficulty |
| 1 | Scenario 1:可用性 | High | Medium |
| 2 | Scenario 6:安全性 | High | Medium |
| 3 | Scenario 7:互操作性 | High | Low |
| 4 | Scenario 8:可修改性 | Medium | Medium |

* + 1. **每个ASR可选的设计决策**
       1. 可用性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| Heartbeat检测 | 周期性地检测系统服务是否可用 |
| Ping-echo | 客户端发起请求，检测系统服务是否可用 |
| 主动冗余 | 所有冗余组件以并行方式对事件作出响应 |
| 被动冗余 | 主要组件对事件作出响应，并通知其他组件进行必要的更新 |

* + - 1. 安全性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 用户数据加密 | 成熟的安全手段，避免泄露用户隐私，保障账户安全 |
| 自动攻击侦测 | 及时探测到外部攻击并做出反应 |
| 拒绝可疑访问 | 最大程度地保证用户的合法性 |

* + - 1. 互操作性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 在服务列表中定位所需服务 | 当所需服务较多时可以快速定位所需服务 |
| 选取和剪裁外部系统的接口 | 外部系统提供的接口可能有很多，有些接口返回的数据可能过多，并不是系统所需要的 |
| 提供多种数据格式 | 多种数据格式来保证系统可以使用不同的外部服务 |
| 使用控制机制来管理、协调服务的调用 | 针对所要服务的访问需求量较多的情况 |

* + - 1. 可修改性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 划分模块 | 缩小修改影响的范围 |
| 添加接口 | 服务与实现分离，隔离需要修改的模块 |
| 抽象通用服务 | 将使用频繁的服务提取出来共用 |

* + 1. **设计决策的选择及分析**
       1. 可用性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| Heartbeat检测 | 采用，可以及时探测到错误的出现，做出应对。 |
| Ping-echo | 不采用，需要相对于heartbeat更多的信息流量，且对错误响应不够快 |
| 主动冗余 | 不采用，成本过高，无法承担 |
| 被动冗余 | 采用，大多数情况下，老师账户管理这一部分的负载不重，即使相关操作失败，只要能在一定时间内恢复，影响并不大。 |

* + - 1. 安全性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 用户数据加密 | 采用，成熟的安全手段，实现容易，效果明显 |
| 自动攻击侦测 | 采用，关系到账户内的资产，老师数量有限，账户管理操作也不集中，系统负荷并不重 |
| 拒绝可疑访问 | 采用，以合法用户偶尔的访问失败为代价提高安全性，可以接受 |

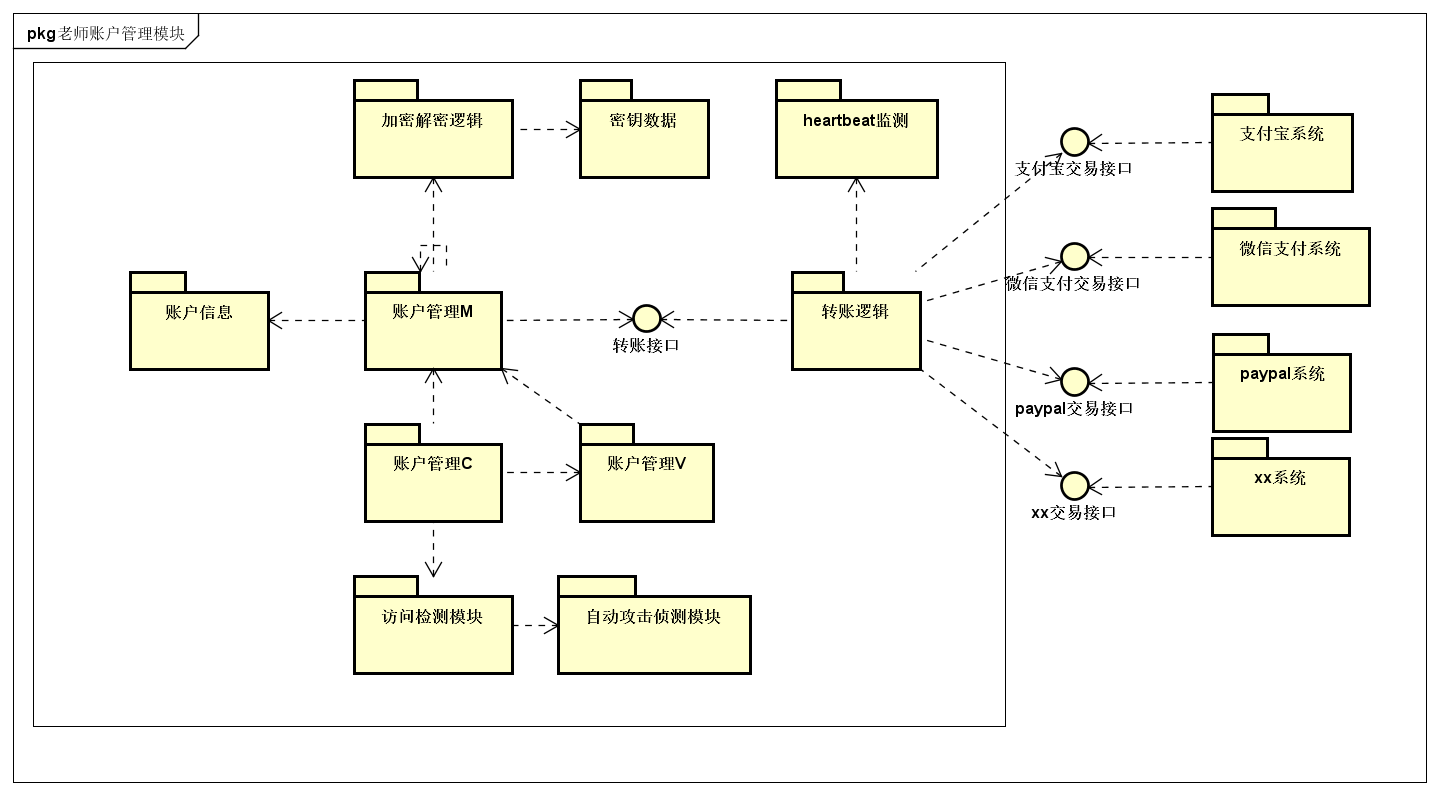
* + - 1. 互操作性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 在服务列表中定位所需服务 | 不采用，这里只用到外部系统的转账服务，并不是应对多种服务需求。 |
| 选取和剪裁外部系统的接口 | 采用，选取需要使用的转账服务接口，对接口重新剪裁有助于获取更加明确的支付信息。 |
| 提供多种数据格式 | 采用，不同的转账提现服务可能采用不同的数据格式，提供多种数据格式有利于与多个外部服务的适配 |
| 使用控制机制来管理、协调服务的调用 | 不采用，所需服务访问不多 |

* + - 1. 可修改性

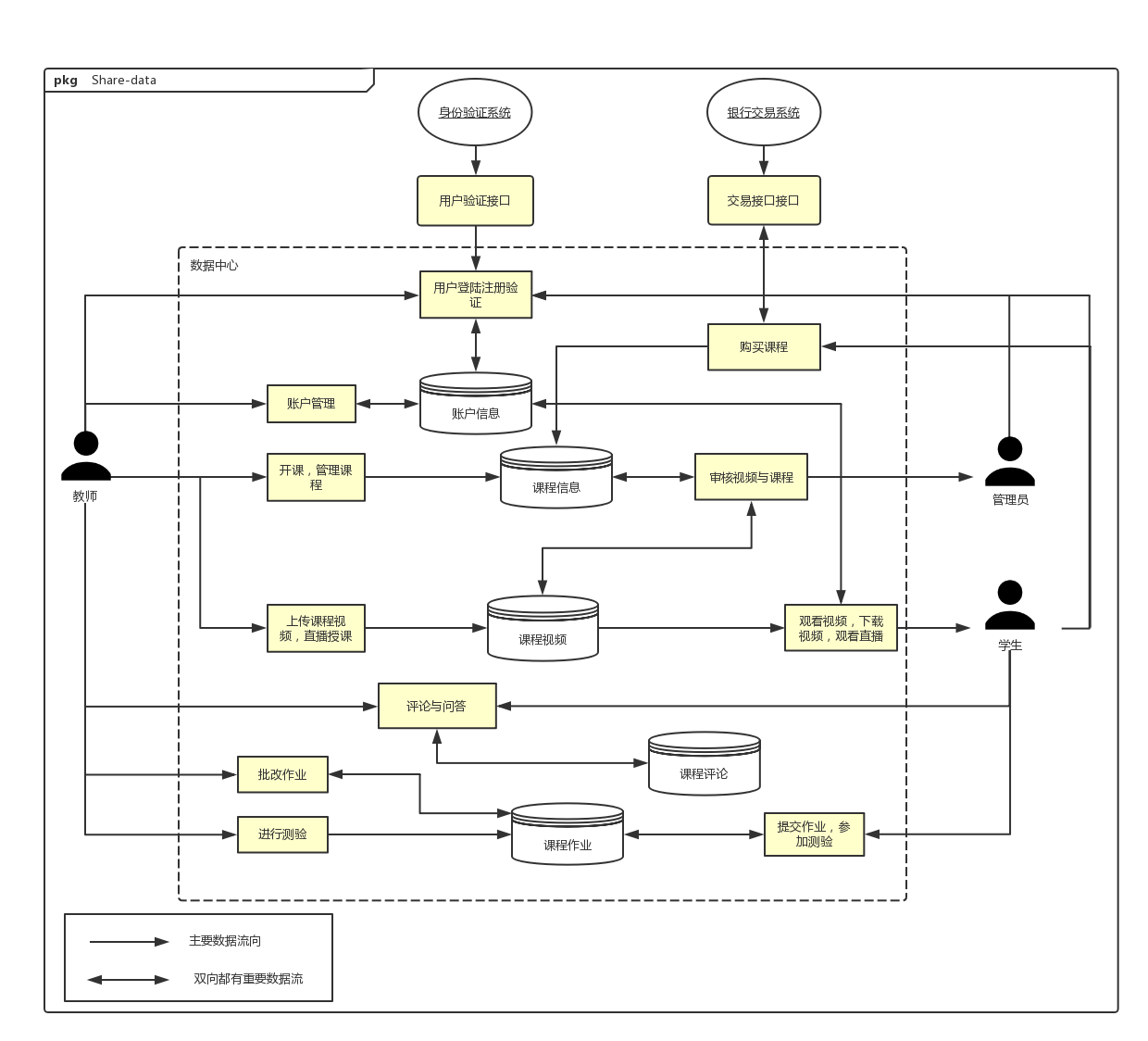
|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 划分模块 | 采用，常用且有效 |
| 添加接口 | 采用，是拓展支持的提现平台的基础 |
| 抽象通用服务 | 不采用，提现平台间数据不同，没有可通用的部分 |

* + 1. **第四次迭代结果**



1. **基于Shared-Data架构的ADD过程**
   1. **第一次迭代过程**

在第一次迭代中，在Shared-Data架构的基础上，参考已经划分好的、较为清晰的功能性需求，系统完成了模块划分和整体的设计。

****

* 1. **第二次迭代过程**

选取课程直播模块作为系统元素进行分解。在质量属性方面，该模块与可用性、负载性能、易用性和实时性等关系密切。

* + 1. **识别所选模块的ASR**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Architectural Drivers | Importance | Difficulty |
| 1 | Scenario 1：可用性,课程直播时，要保证用户的观看不能出现中断 | high | medium |
| 2 | Scenario 2：负载性能，课程直播时可能会有大量的用户访问，系统需要支持10000人以内的并发访问 | medium | high |
| 3 | Scenario 4：实时性，用户访问网站和课程的直播都不能有过长的延时 | high | high |
| 4 | Scenario 5：易用性，保障用户体验 | medium | low |

* + 1. **每个ASR可选的设计决策**
       1. 可用性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 分析 |
| Heartbeat检测 | 直播实时性要求高，heartbeat可以周期性地对系统运行情况进行检测 |
| Ping-echo | 组件会在需要时的发出请求命令，并根据系统的回复判断系统是否正常运行 |
| 服务器冗余 | 对服务器运行时的组件进行备份，在提供服务的服务器出现故障时，冗余的服务器作为备份来使用。 |

* + - 1. 负载性能

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 分析 |
| 负载均衡 | 将大量的用户请求分发到不同的服务器上，以减轻单个服务器的访问压力 |
| CDN | 可以根据用户的区域信息分配最近的CDN服务器，分解服务器的压力，但需要考虑CDN服务器的数据同步问题 |
| 增加硬件计算设备 | 提高计算能力，但是需要增加成本投入 |

* + - 1. 易用性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| 带宽检测 | 需要考虑如何使副本保持一致和同步 |
| 视频转码 | 使用速度更快的处理器、额外的处理资源、额外的内存、速度更快的网络。但是会增加成本 |

* + - 1. 实时性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 分析 |
| BGP中转架构 | 为直播发布和直播观看隶属不同网络运行商的情况构造最短传输路径，从而增强稳定性和减少延时。 |
| 客户端解析优化策略 | 本机缓存域名的解析结果，对域名进行预解析，每次需要直播推流和播放的时候不再需要再进行DNS过程。此处节省几十到几百毫秒的打开延迟。 |
| RTP协议 | RTP在视频监控、视频会议、IP电话上有广泛的应用，实时性强。但国内各大CDN厂商没有支持基于RTP直播的。 |
| HLS 协议 | HTML5可以直接打开播放，这意味着可以把一个直播链接通过微信等应用转发分享，不需要安装任何独立的APP，有浏览器即可，流行度很高。延时在5~7秒（可能大于10秒）。 |
| RTMP协议 | 支持flash player可以支持直播播放，延迟一般在2~5秒，开源软件和开源库支持完整。 |
| HTTP-FLV | 相对于RTMP，HTTP更简单和广为人知，内容延迟同样可以做到2~5秒，打开速度更快。因为没有繁杂的握手，就延迟看，优于RTMP;（同样依赖于flash, 跨平台性差）。 |

* + 1. **设计决策的选择及分析**
       1. 可用性

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| Heartbeat检测 | 采用，课程直播对实时性要求高，heartbeat可以在错误出现时及时监测到 |
| Ping-echo | 不采用，不能保证错误出现时能及时监测到 |
| 服务器冗余（主动冗余） | 采用，课程直播对实时性要求高，主动冗余虽然需要更多资源，但发生错误时可以迅速恢复系统正常运行。 |
| 服务器冗余（被动冗余） | 不采用，发现错误时恢复系统运行耗时较长，不能满足直播实时性要求。 |

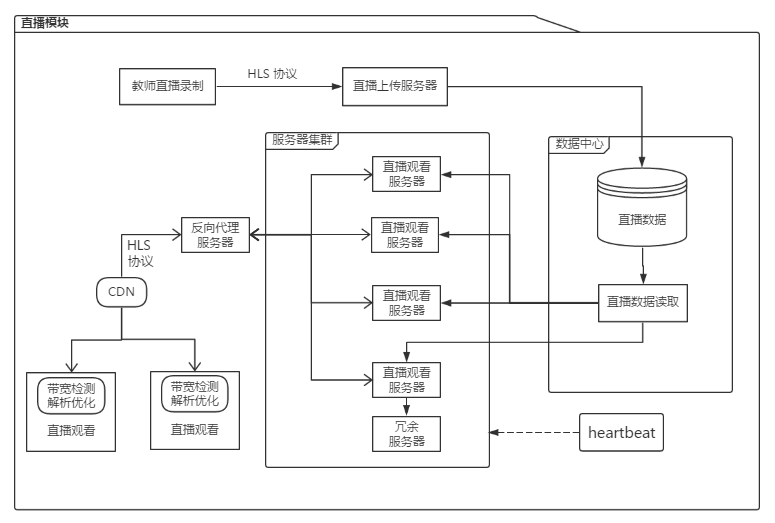
* + - 1. 负载性能

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 负载均衡（反向代理服务器） | 采用，效果好且技术较为成熟 |
| 负载均衡（硬件负载均衡） | 不采用，成本高 |
| 负载均衡（DNS负载均衡） | 不采用，对实时性支持不够 |
| CDN | 采用，国内CND服务商提供的技术服务成熟。 |
| 增加硬件计算设备 | 不采用，成本高 |

* + - 1. 易用性
      2. 实时性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 决策理由 |
| BGP中转架构 | 采用。可以在不同网络运营商之间构造最短路径，从而增强稳定性和减少延时 |
| 客户端解析优化策略 | 采用。直播推流和播放不再进行DNS解析过程，减少打开延迟。 |
| RTP协议 | 未采用。虽然采用RTP协议的延迟低，但是国内各大CDN厂商没有基于RTP直播的，实施难度过大。。 |
| HLS 协议 | 采用。虽然HLS协议的延时略高，但已能满足系统需求，且HLS协议跨平台性高，HTML5即可直接打开播放。 |
| RTMP协议 | 未采用，虽然延迟较低，但是依赖于flash，跨平台性差，严重损害了易用性。 |
| HTTP-FLV | 未采用，虽然延迟较低，但是依赖于flash，跨平台性差。 |

* + 1. **第二次迭代结果**



* 1. **第三次迭代过程**

选取课程注册模块作为系统元素进行分解。在质量属性方面，该模块与性能负载、互操作性、安全性等关系密切。

* + 1. **识别所选模块的ASR**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Achitectural Drivers | Importance | Difficulty |
| 1 | Senario2: 性能负载，系统能够满足至少500个用户同时进行课程的注册操作 | High | High |
| 2 | Senario1: 可用性，大量用户同时进行注册时系统保持可用 | High | Medium |
| 3 | Senario7: 互操作性，付费课程在注册时需要与外部支付系统交互 | High | Low |
| 4 | Senario6: 安全性，用户支付课程费用时，必须保障网络环境的安全以确保用户的资金和信息安全 | High | High |

* + 1. **每个ASR可选的设计决策**
       1. **性能负载**

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 负载均衡器 | 可以保证以相对较优的方式分配请求，但需要着重考虑成本问题 |
| 反向代理 | Nginx，具有高性能，轻量级的优势，处理高并发请求时有相当高的性能 |
| CND | 可以根据用户的区域信息分配最近的CDN服务器，分解服务器的压力，但需要考虑CDN服务器的数据同步问题 |
| DNS负载均衡 | 由DNS服务器充当负载均衡调度器，将用户请求分散到多台服务器上，但是调试成本高，分配策略也有很强的局限性。 |
| 增加可用资源 | 使用更高速的处理器和处理资源、更大的内存和更快的网络，有助于提高性能，但是成本预算会增加。 |

* + - 1. **可用性**

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 心跳检测 | 一个组件定期发送心跳信息，另一个组件监测收听该信息。可以周期性地检测系统服务是否可用 |
| Ping-Echo | 一个组件发出命令，另一个自审查组件在预定义的时间内给予响应，通常用于检测系统服务是否可用 |
| 主动冗余 | 所有冗余组件以并行方式对事件作出响应，如果发生错误，立即使用备份组件，可以实现快速恢复 |
| 被动冗余 | 分为Primary组件和Secondary组件。Primary组件对事件作出响应，并通知Secondary组件进行必须的状态更新，如果发生错误，Secondary组件可以接管工作，停机时间较主动冗余长，但成本较低 |
| 备用件 | 用于在计算平台更换各种不同的故障组件。出现故障后将其重新启动为适当的软件配置并对其状态进行初始化 |

* + - 1. **互操作性**

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 搜索服务列表，定位需要的服务 | 多种服务请求下快速定位所需服务，减少响应时间，提高响应速度 |
| 选取和剪裁外部系统的接口 | 外部系统提供的接口可能有很多，有些接口返回的数据可能过多，并不是系统所需要的 |
| 提供多种数据格式 | 多种数据格式来保证系统可以使用不同的外部服务，以防出现数据不匹配的问题 |

* + - 1. **安全性**

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 数据加密 | 存储和传输用户数据时都进行加密 |
| 自动监测攻击 | 监控对系统的入侵、验证信息的一致性并总结攻击历史特征进行防护，会给系统带来额外开销 |
| 拒绝可疑访问 | 保证访问的都是有授权的用户，有助于保证合法性，但是可能会引起误伤 |
| 维护审批记录 | 有效防止数据被恶意修改 |

* + 1. **设计决策的选择及分析**
       1. **性能负载**

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 负载均衡器 | 不采用，第三方软件虽然可以保证以相对较优的方式分配请求，但是成本过高。 |
| 反向代理 | 采用，高性能，轻量级，拥有众多请求分配策略，有助于处理高并发请求且技术成熟。 |
| CND | 采用，可以很好的分担服务器的压力，且国内各大CND服务商发展成熟。 |
| DNS负载均衡 | 不采用，分配策略具有局限性，无法根据实际服务器的实时负载差异来调整调度策略，记录缓存过于复杂，并不能很好地完成工作量分配。 |
| 增加可用资源 | 不采用，成本过高。 |

* + - 1. **可用性**

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| Heartbeat检测 | 采用，周期性检查可以及时检测服务宕机以采取措施 |
| Ping-echo | 不采用，检测不及时，可能出现服务已经不可用但仍然能检测得到 |
| 主动冗余 | 采用，多个备件并发对事件作出响应，需要系统提供更多资源，会增加成本 |
| 被动冗余 | 采用，课程注册虽然要求很高的可用性，但由于课程名额无限制，用户更关系自己是否报名成功，在一定时间内恢复系统是可以接受的。 |
| 备份件 | 不采用，成本较高 |

* + - 1. **可操作性**

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 在服务列表中定位所需服务 | 不采用，这里只用到外部系统的支付服务，并不是应对多种服务需求。 |
| 选取和剪裁外部系统的接口 | 采用，选取需要使用的支付服务接口，对接口重新剪裁有助于获取更加明确的支付信息。 |
| 提供多种数据格式 | 采用，不同的支付服务可能采用不同的数据格式，提供多种数据格式有利于与多个外部服务的适配 |

* + - 1. **安全性**

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 数据加密 | 采用，加密用户数据以防被泄漏，保障财产和个人信息安全 |
| 自动监测攻击 | 不采用，加重了系统的运行负担，且第三方支付服务本身对安全性也有较好地保障 |
| 拒绝可疑访问 | 不采用，可能会误伤正常用户的访问，降低用户体验和系统的可用性 |
| 维护审批记录 | 采用，可以有效快速恢复 |

1. **ATAM分析过程**
2. **挑战和经验**
3. **组员和分工**