iCourse网络课程平台

项目报告

(PM)张诗琦141250190 董轶波131250171

王子安141250146 徐江河141250161

袁楚宏141250178 袁阳阳141250179

张鑫龙141250195 张云飞141250197

|  |  |  |
| --- | --- | --- |
| 文档版本 | 更新内容 | 更新时间 |
| v0.1 | 主要功能点和场景分析、两种架构选择及对应的架构图设计、非功能需求及ASR描述、两种结构的比较、具体实现技术的比较、部分MVC架构的ADD过程 | 2017-3-6 |
| v0.3 | 全部MVC架构的ADD过程 | 2017-3-6 |
| v0.7 | 全部Shared-Data架构的ADD过程 | 2017-3-7 |
| v0.8 | ATAM分析过程 | 2017-3-11 |
| v0.9 | 系统类图设计和组件/连接件到实现类的映射 | 2017-3-12 |
| v1.0 | 完善文档剩余内容 | 2017-3-14 |

目录

[1. 主要功能点和操作场景分析 3](#_Toc477039650)

[1.1. 主要功能点 3](#_Toc477039651)

[1.2. 操作场景分析 3](#_Toc477039652)

[1.2.1. 教师功能 4](#_Toc477039653)

[1.2.2. 学生功能 6](#_Toc477039654)

[1.2.3. 管理员功能 8](#_Toc477039655)

[1.2.4. 系统其他功能 8](#_Toc477039656)

[2. 两种架构选择及对应的架构图设计 9](#_Toc477039657)

[2.1. MVC架构 9](#_Toc477039658)

[2.1.1. 模块视图 9](#_Toc477039659)

[2.1.2. 组件-连接件视图 9](#_Toc477039660)

[2.2. Shared-Data架构 10](#_Toc477039661)

[2.2.1. 模块视图 10](#_Toc477039662)

[2.2.2. 组件-连接件视图 10](#_Toc477039663)

[3. 非功能需求及ASR描述 11](#_Toc477039664)

[3.1. 可用性 11](#_Toc477039665)

[3.2. 性能 11](#_Toc477039666)

[3.2.1. 性能场景1：负载 11](#_Toc477039667)

[3.2.2. 性能场景2：容量 11](#_Toc477039668)

[3.2.3. 性能场景3：实时性 12](#_Toc477039669)

[3.3. 易用性 12](#_Toc477039670)

[3.4. 安全性 12](#_Toc477039671)

[3.5. 互操作性 13](#_Toc477039672)

[3.6. 可修改性 13](#_Toc477039673)

[3.7. 可维护性 13](#_Toc477039674)

[4. 系统类图设计 13](#_Toc477039675)

[5. 组件/连接件到实现类的映射 14](#_Toc477039676)

[6. 两种结构的比较及最终选择 31](#_Toc477039677)

[6.1. 整体比较 31](#_Toc477039678)

[6.2. 基于系统实际的最终选择 33](#_Toc477039679)

[7. 具体实现技术的比较及最终选择 15](#_Toc477039680)

[8. 基于MVC架构的ADD过程 16](#_Toc477039681)

[8.1. 第一次迭代结果 16](#_Toc477039682)

[8.2. 第二次迭代过程 16](#_Toc477039683)

[8.2.1. 识别所选模块的ASR 16](#_Toc477039684)

[8.2.2. 每个ASR可选的设计决策 17](#_Toc477039685)

[8.2.3. 设计决策的选择及分析 18](#_Toc477039686)

[8.2.4. 第二次迭代结果 19](#_Toc477039687)

[8.3. 第三次迭代过程 19](#_Toc477039688)

[8.3.1. 识别所选模块的ASR 19](#_Toc477039689)

[8.3.2. 每个ASR可选的设计决策 20](#_Toc477039690)

[8.3.3. 设计决策的选择及分析 20](#_Toc477039691)

[8.3.4. 第三次迭代结果 21](#_Toc477039692)

[8.4. 第四次迭代过程 21](#_Toc477039693)

[8.4.1 识别所选模块的ASR 21](#_Toc477039694)

[8.4.2 每个ASR可选的设计决策 21](#_Toc477039695)

[8.4.3 设计决策的选择及分析 22](#_Toc477039696)

[8.4.4 第四次迭代结果 23](#_Toc477039697)

[9. 基于Shared-Data架构的ADD过程 23](#_Toc477039698)

[9.1. 第一次迭代过程 23](#_Toc477039699)

[9.2. 第二次迭代过程 24](#_Toc477039700)

[9.2.1. 识别所选模块的ASR 24](#_Toc477039701)

[9.2.2. 每个ASR可选的设计决策 25](#_Toc477039702)

[9.2.3. 设计决策的选择及分析 25](#_Toc477039703)

[9.2.4. 第二次迭代结果 26](#_Toc477039704)

[9.3. 第三次迭代过程 27](#_Toc477039705)

[9.3.1. 识别所选模块的ASR 27](#_Toc477039706)

[9.3.2. 每个ASR可选的设计决策 27](#_Toc477039707)

[9.3.3. 设计决策的选择及分析 28](#_Toc477039708)

[9.3.4. 第二次迭代结果 29](#_Toc477039709)

[9.4. 第四次迭代过程 29](#_Toc477039710)

[9.4.1. 识别所选模块的ASR 29](#_Toc477039711)

[9.4.2. 每个ASR可选的设计决策 29](#_Toc477039712)

[9.4.3. 设计决策的选择及分析 30](#_Toc477039713)

[9.4.4. 第四次迭代结果 31](#_Toc477039714)

[10. ATAM分析过程 31](#_Toc477039715)

[10.1. 质量属性效用树 33](#_Toc477039716)

[10.2. ATAM分析 34](#_Toc477039717)

[10.3. 敏感点和权衡点 41](#_Toc477039718)

[10.3.1. 敏感点 41](#_Toc477039719)

[10.3.2. 权衡点 42](#_Toc477039720)

[10.4. 风险与非风险 42](#_Toc477039721)

[10.4.1. 风险 43](#_Toc477039722)

[10.4.2. 非风险 43](#_Toc477039723)

[11. 挑战和经验 43](#_Toc477039724)

[12. 组员和分工 44](#_Toc477039725)

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 |

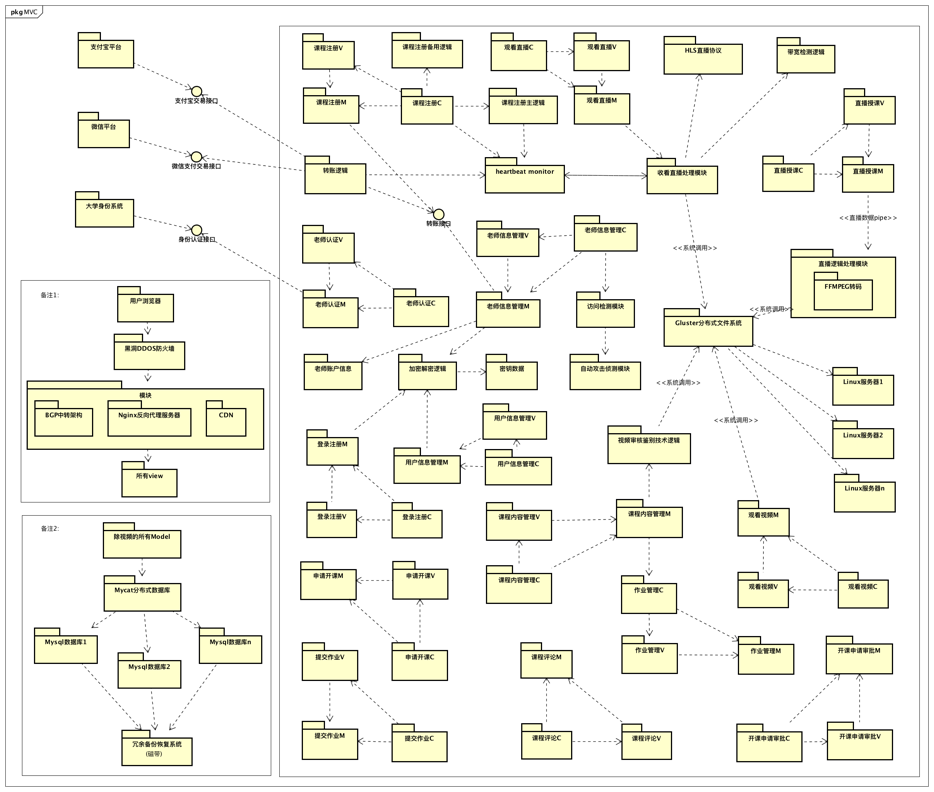
* + - 1. 教师身份审核

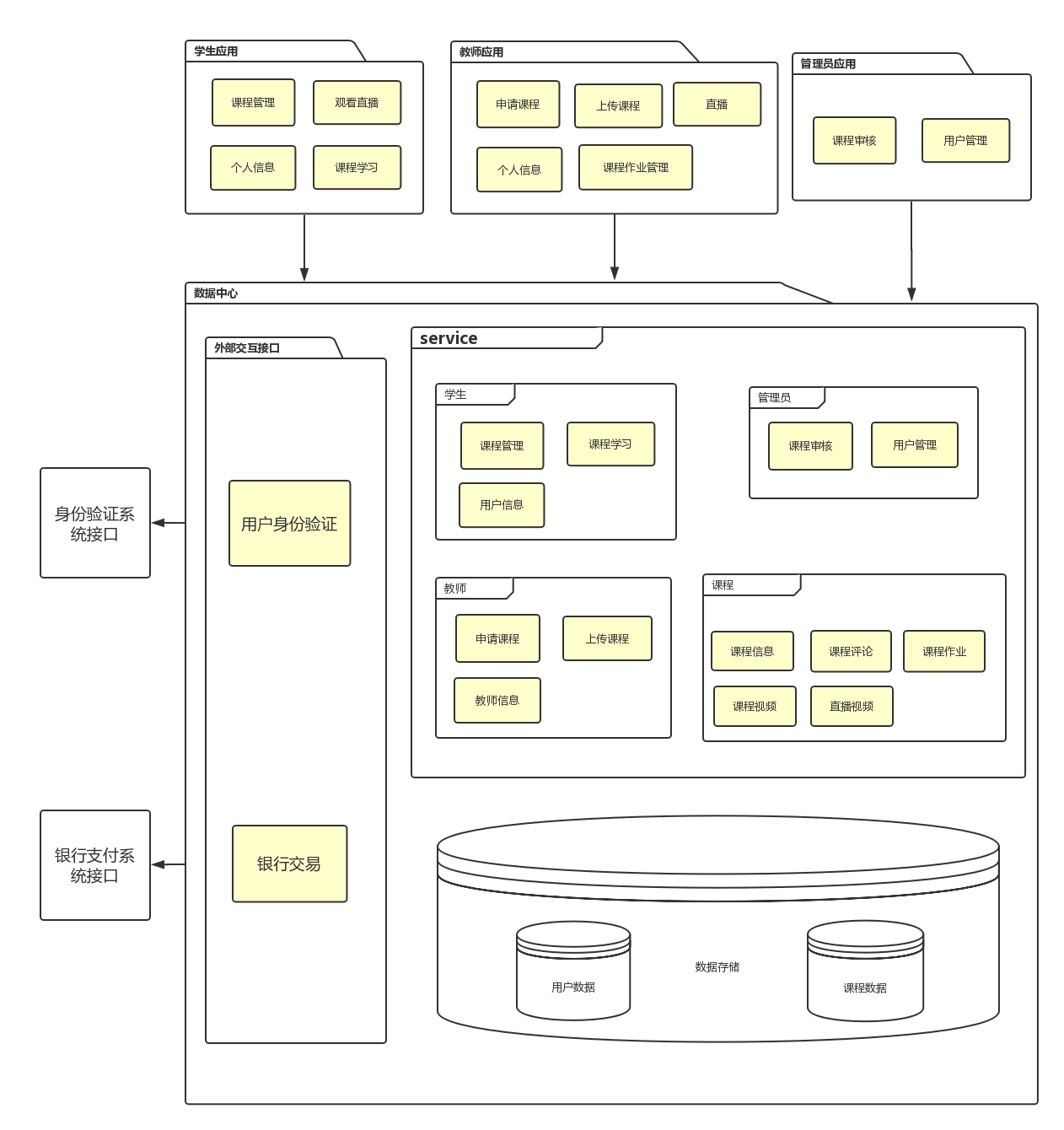
|  |  |
| --- | --- |
|  | 行为 |
| 刺激 | 教师提交认证申请 |
| 响应 | 系统提示收到新的教师认证申请 |
| 刺激 | 系统审批教师认证通过 |
| 响应 | 系统通知相关教师申请通过 |
| 刺激 | 系统审批教师认证申请不通过 |
| 响应 | 系统通知相关教师申请驳回 |

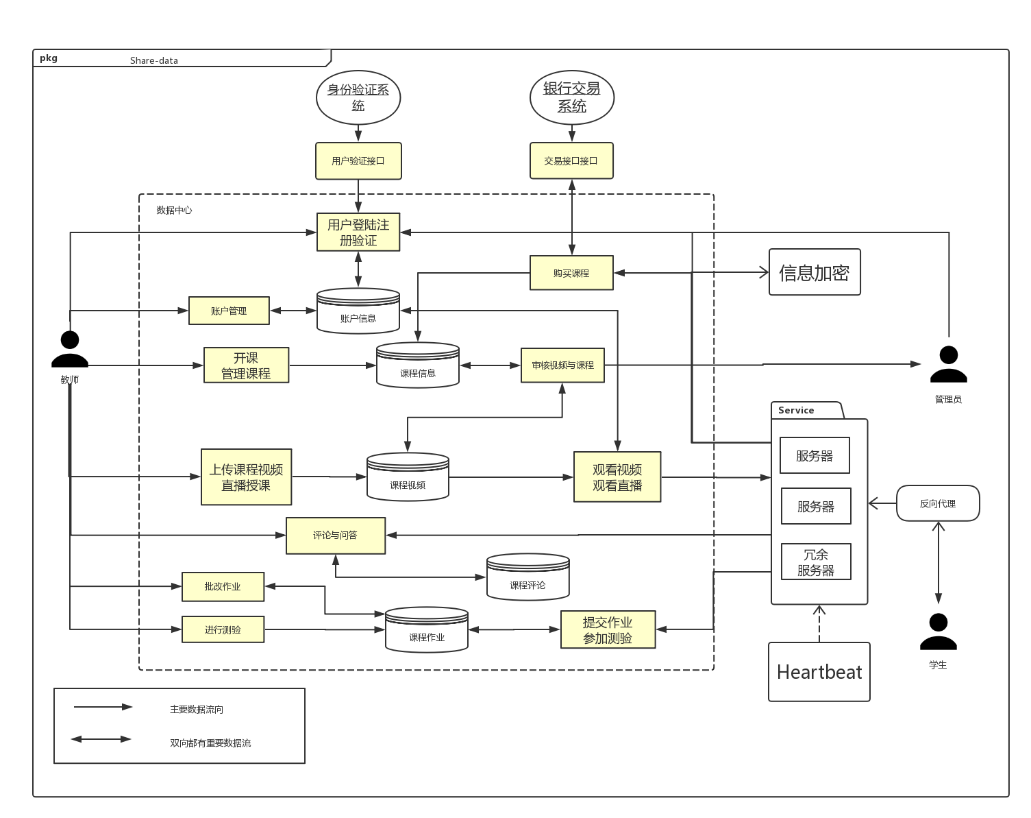
1. **两种架构选择及对应的架构图设计**

注：清晰的架构图设计详见附属的images文件夹。

* 1. **MVC架构**
     1. **模块视图**
     2. **组件-连接件视图**



* 1. **Shared-Data架构**
     1. **模块视图**
     2. **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10000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500T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15秒 |

* 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 | 用户观看直播 |
| 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 | 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. 修改时间低于业界平均水平 3. 修改一个模块时，对其他模块的修改的代码行数不超过10% 4. 修改时不影响无关的系统功能 |

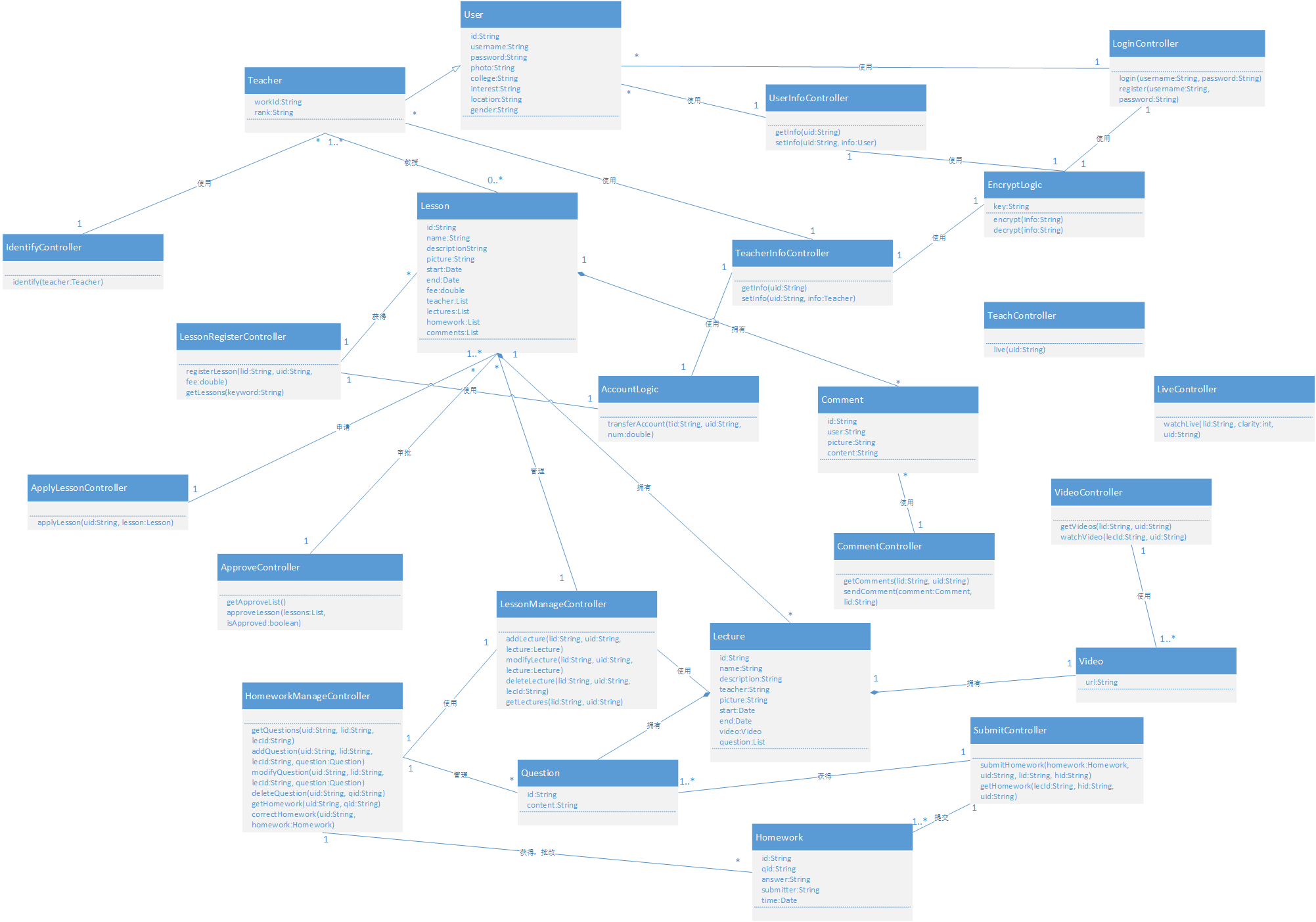
* 1. **可维护性**

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**

注：清晰的系统类图设计详见附属的images文件夹。

****

1. **组件/连接件到实现类的映射**

|  |  |
| --- | --- |
| 实现类 | 组件/连接件 |
| User | 老师信息管理M  用户信息管理M  老师认证M  登录注册M |
| Teacher | 老师信息管理M  老师认证M  登录注册M |
| LoginController | 登录注册C |
| UserInfoController | 用户信息管理C |
| EncryptLogic | 加密解密逻辑 |
| IdentifyController | 老师认证C |
| Lesson | 课程注册M  课程内容管理M  开课申请审批M  申请开课M |
| TeacherInfoController | 老师信息管理C |
| TeachController | 直播授课C |
| LessonRegisterController | 课程注册C |
| AccountLogic | 转账逻辑 |
| Comment | 课程评论M |
| LiveController | 观看直播C |
| ApplyLessonController | 申请开课C |
| ApproveController | 开课申请审批C |
| LessonManageController | 课程内容管理C |
| CommentController | 课程评论C |
| VideoController | 观看视频C |
| Video | 观看视频M |
| Lecture | 课程内容管理M |
| HomeworkManageController | 作业管理C |
| Question | 作业管理M  提交作业M |
| SubmitController | 提交作业C |
| Homework | 作业管理M  提交作业M |

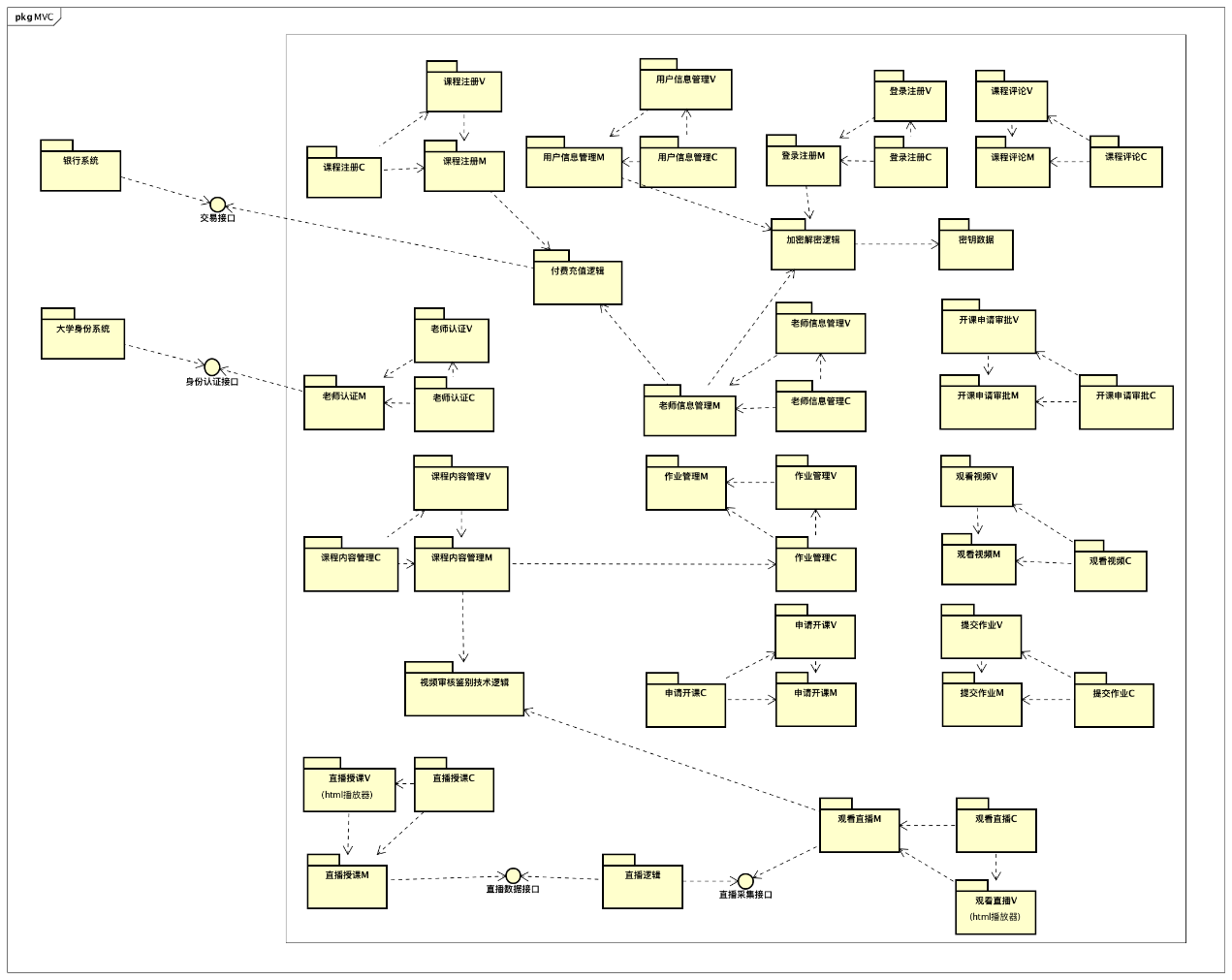
1. **具体实现技术的比较**

注：最终技术的选择可见ADD的详细描述

|  |  |  |
| --- | --- | --- |
| 类别 | 实现方式 | 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 | 对高并发请求支持较好，有较为成熟的开发框架和技术支持,例如Nginx |
| CDN | 有效提高分布在较广地域范围内的用户使用系统时的数据传输速度和稳定性，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 | 成本低，技术成熟，操作简单，容量大。 |
| 双机热备 | 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 |

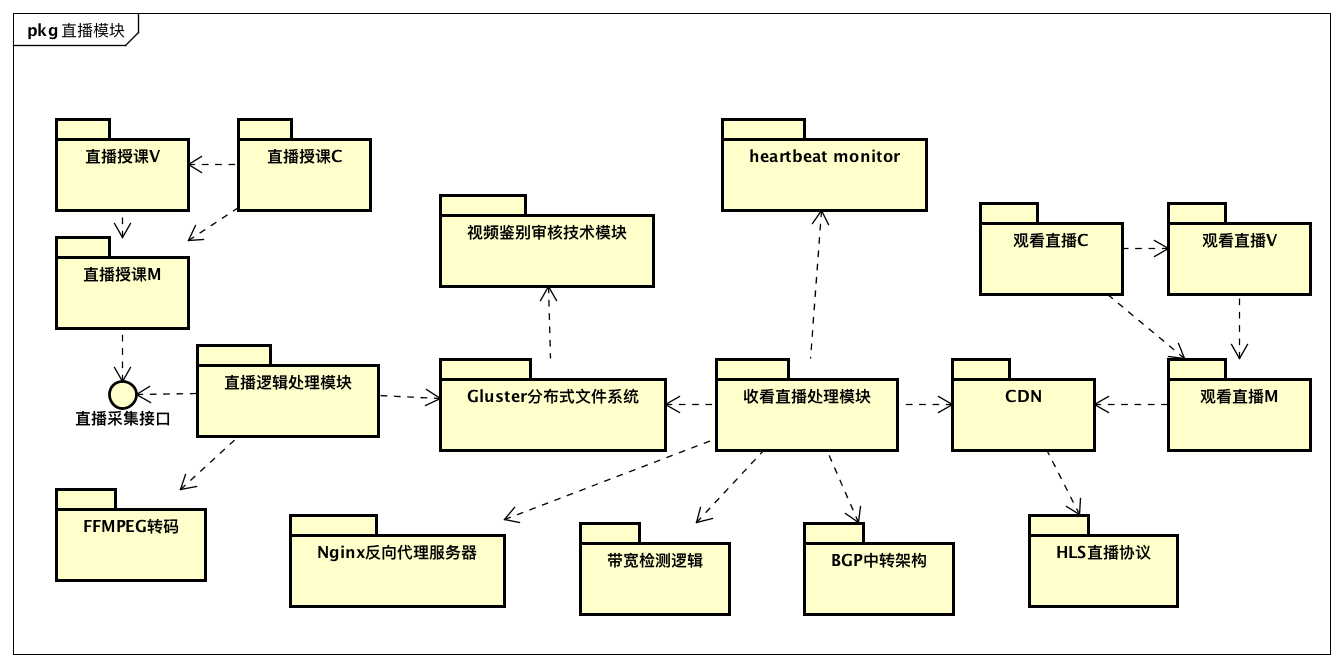
* + - 1. 互操作性

|  |  |
| --- | --- |
| 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**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Architectual Drivers | Importance | Difficulty |
| 1 | Senario1: 可用性，大量学生访问系统时，系统要保持可用 | High | Medium |
| 2 | 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 |
| 3 | 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. 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. 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。 |

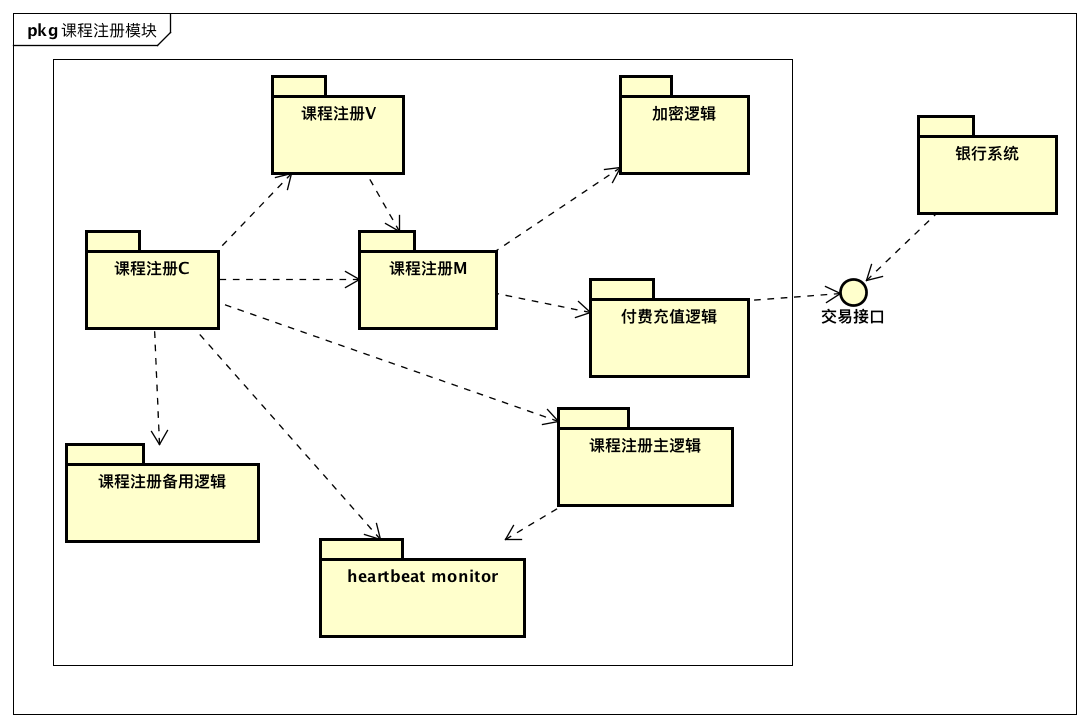
* + - 1. 可操作性

|  |  |
| --- | --- |
| 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**



* 1. **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**

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

* + 1. **每个ASR可选的设计决策**
       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. 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. **设计决策的选择及分析**
       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. 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 |

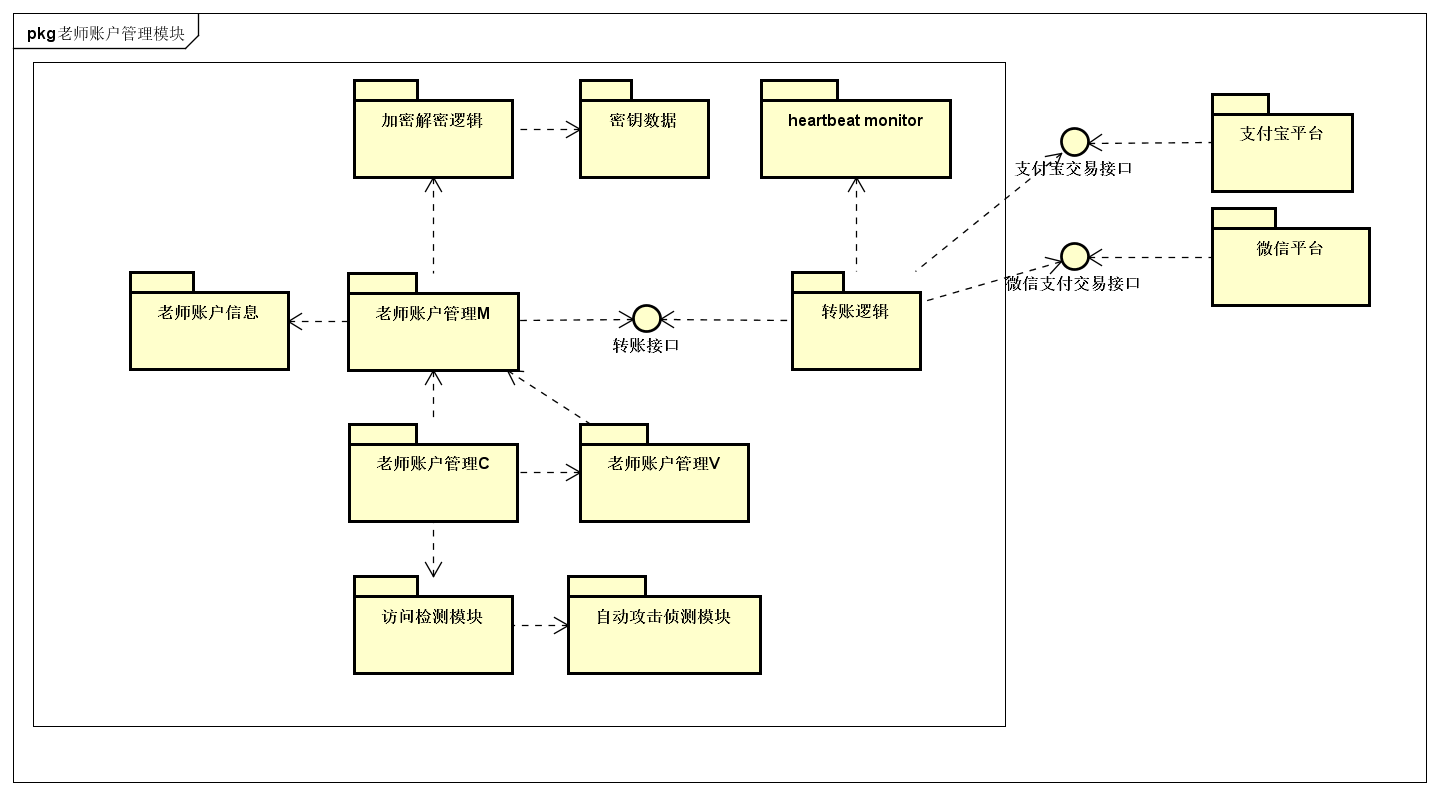
* + - 1. 互操作性

|  |  |
| --- | --- |
| 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. **第四次迭代结果**

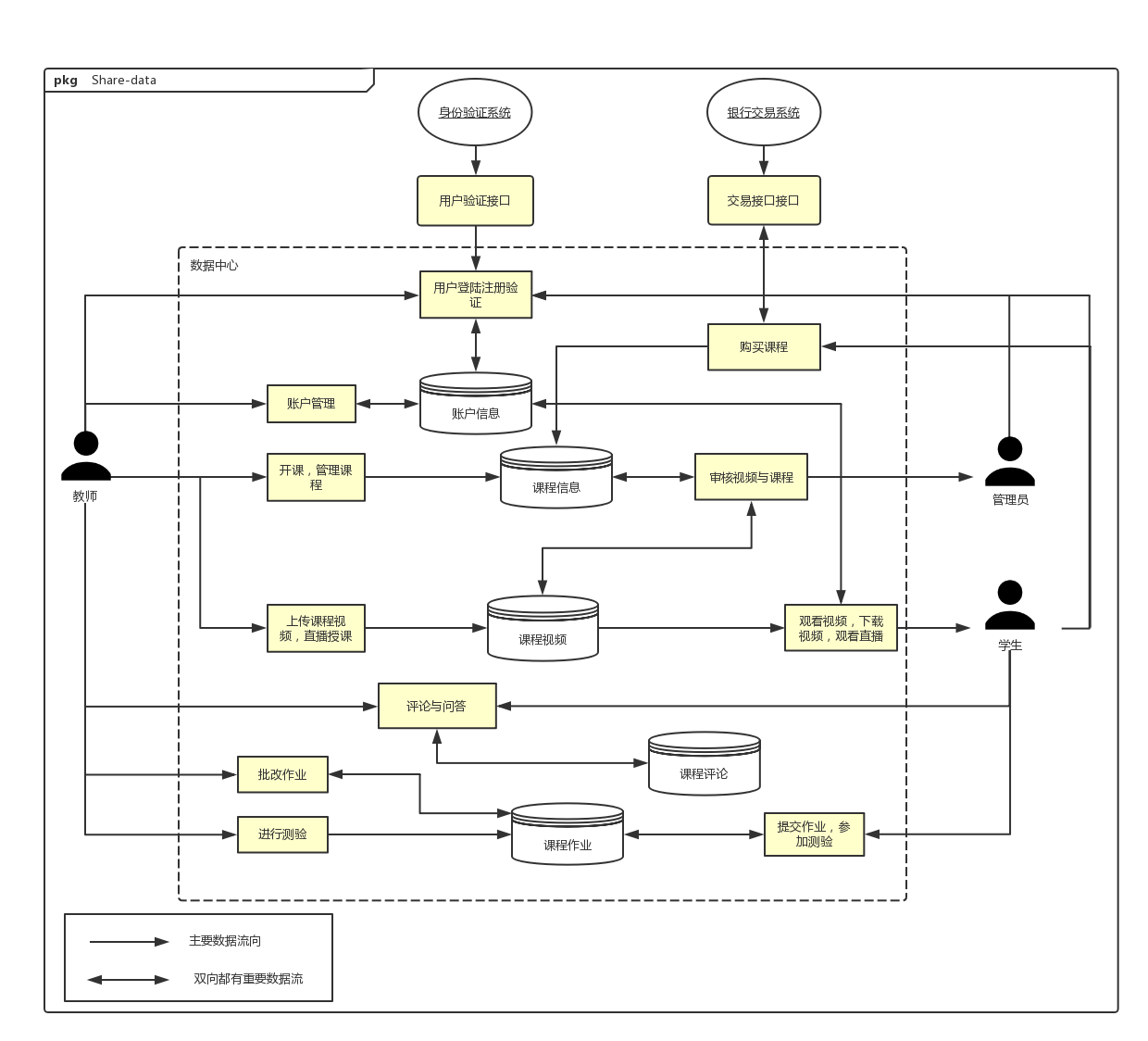


* 1. **第五次迭代之后**

受篇幅所限，第五次迭代直至完成架构的过程此处不再给出，最终结果请参照2.1.。

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 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 |
| 3 | 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 |
| 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. 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。 |
| 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. 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 |
| 基于HTML5的视频播放方式 | HTML5的新特性，播放视频更加方便，正在被越来越广泛地使用。 |
| 基于FLASH的视频播放方式 | 传统的视频播放方式，目前在视频播放方面仍然占有重要的地位。 |

* + 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. 实时性

|  |  |
| --- | --- |
| 可选的设计决策或模式 | 理由和分析 |
| 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。 |

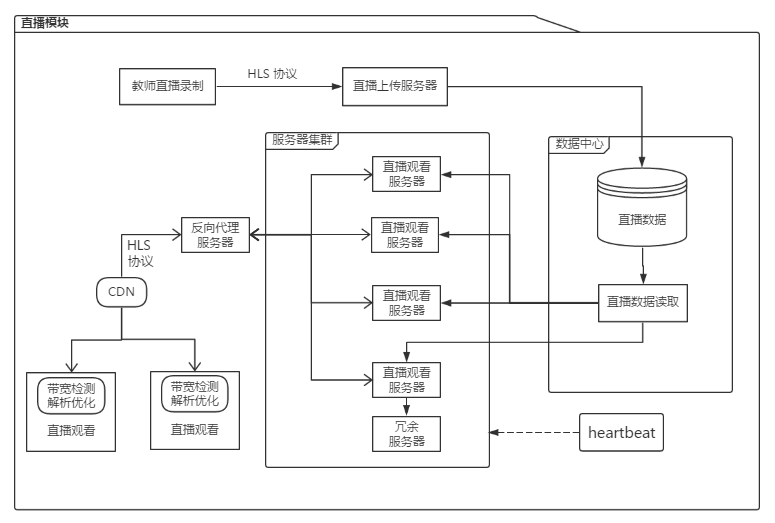
* + - 1. 负载性能

|  |  |
| --- | --- |
| 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CDN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

|  |  |
| --- | --- |
| 可选的设计决策和模式 | 理由和分析 |
| 带宽检测 | 采用，可以灵活地根据带宽调整用户的视频清晰度，提高用户的观看体验。 |
| FFMPEG转码 | 采用，通过视频转码，保证视频格式的统一性，方便用户观看视频。 |
| 基于HTML5的视频播放方式 | 采用，HTML5提供了网页视频播放功能，比传统的flash播放体验更好。 |
| 基于FLASH的视频播放方式 | 采用，HTML5支持的浏览器有限，考虑到用户使用的浏览器平台较多，FLASH播放的方式仍然保留。 |

* + 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**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Architectual Drivers | Importance | Difficulty |
| 1 | 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 |
| 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 |

* + 1. **每个ASR可选的设计决策**
       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. 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. **设计决策的选择及分析**
       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 |

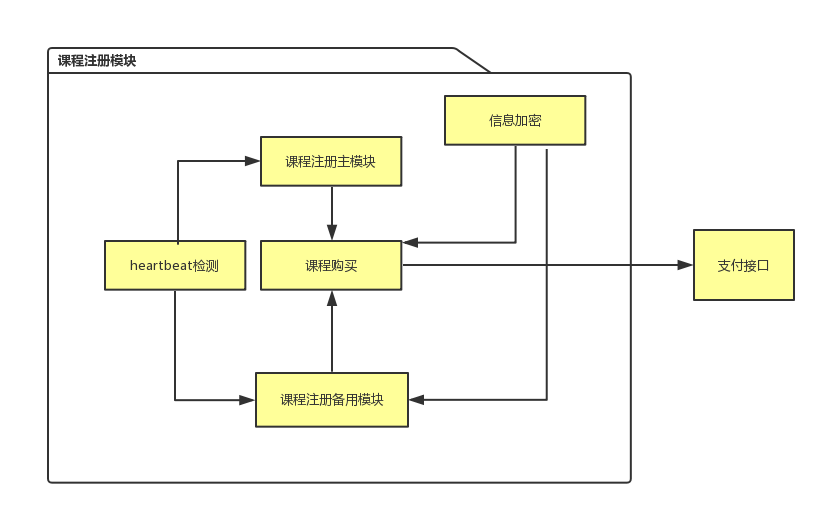
* + - 1. 可用性

|  |  |
| --- | --- |
| 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. 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. **第三次迭代结果**

****

* 1. **第四次迭代过程**

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

* + 1. **识别所选模块的ASR**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Architectual 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 |

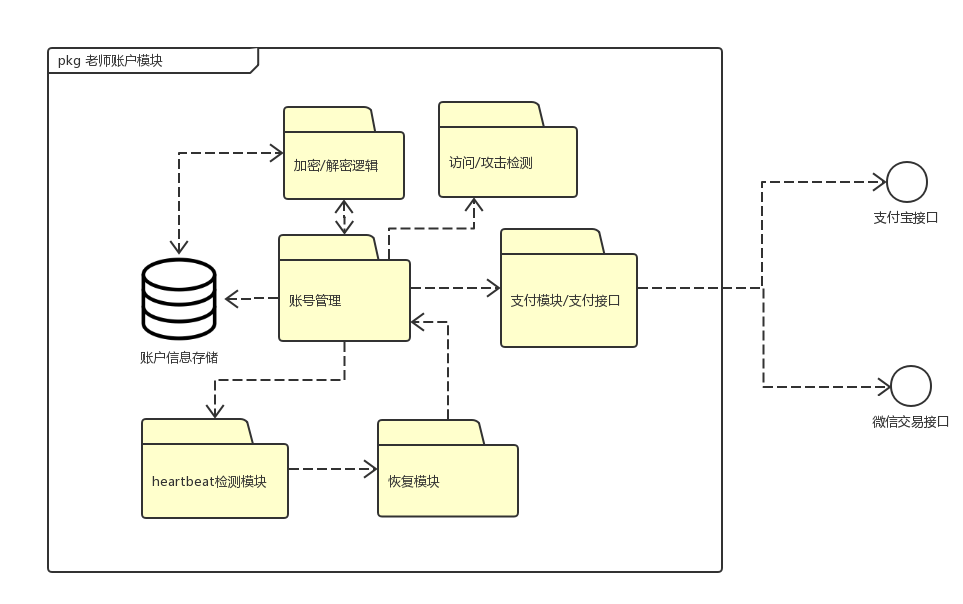
* + - 1. 互操作性

|  |  |
| --- | --- |
| 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. **第五次迭代之后**

受篇幅所限，第五次迭代直至完成架构的过程此处不再给出，最终结果请参照2.2.。

1. **两种结构的比较及最终选择**
   1. **整体比较**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Shared-Data | | 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 | 没有明显区别 | | | |
| 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 | 耦合度高，对代码、模块的维护比较困难 | 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**

我们在经过ADD之后，再次比较了两种架构的优缺点，最终决定选择MVC架构作为本系统设计的高层架构，理由如下：

1. 基于文档6.2的架构比较，我们发现MVC架构从安全性、可修改性、开发难度、负载能力等方面优于Shared-Data架构。安全性在涉及金钱交易的本系统是非常关键的质量属性；负载能力在直播这个系统核心功能中很重要；而可修改性、开发人员对架构熟练程度对于系统开发的成本有举足轻重的影响。

2. Shared-Data架构优势主要在于视频资源集中便于维护，数据生产者、消费者方便单独测试。但是在进一步深化对ASR的分析之后，我们认为为了保证性能（主要在于视频资源的传输）与可用性，采用分布式存储来保存、分发视频资源的策略是必需的。然而Shared-Data这一架构对于分布式存储额支持较差，不能满足性能、可用性需求。另外，Shared-Data在可测试性方面的优势由于本架构中数据生产者、消费者之间存在实际的数据耦合，其实不显著优于MVC架构。

综合以上分析，我们选择MVC架构作为我们的系统设计架构。

1. **ATAM分析过程**
   1. **质量属性效用树**

|  |  |  |
| --- | --- | --- |
| 质量属性 | 属性求精 | 场景 |
| 安全性 | 数据加密 | A1:用户数据被窃取，窃取者无法破译加密后的用户密码等敏感信息(H,H) |
| 攻击监测和防御 | A2:系统受到外界恶意攻击，系统监测到攻击后，系统防火墙会识别攻击并阻断攻击(H,H) |
| 拒绝可疑访问 | A3:系统接收到某个未授权用户的访问请求，系统拒绝访问(H,M) |
| 互操作性 | 多接口管理 | A4:当学生用户进行课程付费/教师用户使用转账功能时，发现并使用交易接口的成功率不低于99%(H,M) |
| 接口数据一致 | A5:当系统发起或收到与大学身份验证系统/直播采集系统的请求和响应时，系统成功接受并操作数据的概率不低于99%(H,M) |
| 性能 | 实时性 | A6: 有5000人同时观看直播时，直播延时不超过3s。(H,H) |
| 容量 | A7: 系统能存储500T的视频，10T的用户信息，10T的课程信息(H,L) |
| 负载 | A8: 系统在10000个用户并发访问下能正常运转(H,H) |
| A9: 用户访问请求的平均等待时间低于1.5秒(H,H) |
| 可用性 | 服务可用率 | A10: 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)(H,M) |
| 故障检测、恢复 | A11：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(H,M) |
| A12：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(M,L) |
| 易用性 | 自适应直播 | A13：用户观看到高清，低卡顿的课程直播，并且能根据当前网速来自适应获得最佳体验(H，H) |
| 操作简单 | A14：使用系统的用户，能够熟练简单掌握系统的功能的时间在一天以内(H，L) |
| 可维护性 |  | A15：与系统交互的银行系统、身份验证系统等外部系统提供的接口发生更改时，相关模块的修改在0.5个人月内完成(M,M) |
| A16：当增加新的作业文件格式、音频格式时，维护相应代码的时间应不超过2小时，且受影响的代码量小于1% (H，M) |
| A17：当系统部署环境的新特性影响系统原有功能的实现时，系统应能在0.5个人月内完成更新，且成本不超过总开发成本的5% (M,M) |

* 1. **ATAM分析**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A1** | 系统受到外界恶意攻击，系统监测到攻击后，系统防火墙会识别攻击并阻断攻击，若攻击无法阻止，用户数据被窃取后窃取者无法破译 | | | |
| **质量属性** | 安全性 | | | |
| **环境** | 运行时 | | | |
| **刺激** | 外部攻击 | | | |
| **响应** | 系统监测到攻击后，防火墙识别攻击并阻断攻击 | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| 自动攻击侦测 | S1 | T1 | R1 |  |
| 使用DDOS分防火墙 | S2 |  |  |  |
| 对信息进行加密 | S3 |  |  |  |
| 拒绝可疑访问 | S4 | T2 |  | N1 |
| **理由说明** | 本项目与用户的银行及各支付平台账户相关联，所以对安全性有很高要求。  攻击者被系统识别可疑访问并拒绝，可能使用各种方式对系统发动攻击。为安全起见，自动侦测攻击并使用防火墙阻止攻击。虽然可能降低部分性能，但是这种风险是可以接受的，而泄密的风险是不可接受的。并且对数据进行强加密，保证在发生泄密的最坏情况下用户数据不被窃取。 | | | |
| **相关架构图** | F:\software\Tencent\QQ\QQData\390412006\Image\C2C\9{N~$6@Z%PVMNV2)I2[8JVT.png | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A4** | 当学生用户进行课程付费/教师用户使用转账功能时 | | | |
| **质量属性** | 互操作性 | | | |
| **环境** | 系统正常运行时 | | | |
| **刺激** | 学生用户发起课程付费/教师用户发起转账的请求和响应 | | | |
| **响应** | 系统成功提供接口并且用户使用交易接口的成功率不低于99% | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| 剪裁外部接口 | S5 | T3 |  | N2 |
| 服务列表定位 | S6 |  |  |  |
| **理由说明** | 通过转账逻辑模块裁剪支付平台支付控制逻辑，减少系统在适配多支付平台的多类型数据上的负担，同时减少了用户支付上的操作负担，提高了易用性。 | | | |
| **相关架构图** |  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A6** | 用户在系统上观看讲座直播 | | | |
| **质量属性** | 性能 | | | |
| **环境** | 联网状态 | | | |
| **刺激** | 学生用户希望在系统上观看讲座直播 | | | |
| **响应** | 系统直播视频打开快，延时短 | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| HLS协议 | S7 | T4 | R2 |  |
| BGP中转架构 | S8 |  |  | N3 |
| **理由说明** | 使用HLS(HTTP LIVE STREAMING)协议，延时虽然较高，但仍在可接受的范围内。且兼容性比较好，提高了可移植性。在性能和可移植性兼备的情况下，选择了HLS新协议。BGP中转架构可以较好地提高响应效率，降低延时。 | | | |
| **相关架构图** |  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A7** | 系统上线一段时间后，用户量上升，开设的课程增多，上传的课程视频数量膨胀，授课直播增加。 | | | |
| **质量属性** | 性能(容量) | | | |
| **环境** | 运行时 | | | |
| **刺激** | 用户配置个人信息/账户/上传课程视频/直播授课 | | | |
| **响应** | 系统存储用户上传的数据，包括个人信息，账户配置，课程视频，直播缓存等。 | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| 使用GlusterFS分布式文件系统 | S9 | T5 | R3 |  |
| 使用数据库集群 | S10 |  |  | N4 |
| **理由说明** | 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；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。这同时也保证了系统容量的可扩展性与服务器资源的动态分配，避免浪费 | | | |
| **相关架构图** |  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A8** | 在某个时间段内有数千个学生用户同时访问网站，用户的请求包括观看课程视频和直播，注册课程等。 | | | |
| **质量属性** | 性能(负载) | | | |
| **环境** | 运行时 | | | |
| **刺激** | 学生观看课程视频/观看直播/注册报名课程 | | | |
| **响应** | 系统正常响应每个用户的请求 | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| 使用反向代理 | S11 | T6 |  | N5 |
| **理由说明** | 使用反向代理将用户请求进行分发，可以减少单个服务器运行的压力，并且使用了反向代理后还可以通过增加服务器数量来提高系统的负载性能 | | | |
| **相关架构图** | F:\software\Tencent\QQ\QQData\390412006\Image\C2C\W$GEHZM%~[GHQVP76LGTJ}5.png | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A10** | 系统运行过程中随着服务，服务器总会产生少量异常影响服务器持续服务性能，于是需要周期性停机维护。 | | | |
| **质量属性** | 可用性(服务可用率) | | | |
| **环境** | 运行时、维护时 | | | |
| **刺激** | 服务器因为持续服务而逐渐不稳定 | | | |
| **响应** | 服务器周期性段时间停机维护。 | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| 使用热备份与实时负载均衡服务器提供备用服务 | S12 | T7 |  | N6 |
| **理由说明** | 通过热备份与实时负载均衡服务器提供备用服务，可以保证运行时主服务器突发不可运行时快速恢复的异常时能启用热备份维持服务，保障系统服务时间覆盖率。 | | | |
| **相关架构图** |  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A11，A12** | 系统运行过程中可能伴随着数据库断线、直播老师掉线等外部模块失去连接的问题。 | | | |
| **质量属性** | 可用性(故障检测、恢复) | | | |
| **环境** | 运行时 | | | |
| **刺激** | 服务器通讯对象断开了网络连接 | | | |
| **响应** | 服务器暂停数据沟通，尝试重新与目标建立网络连接 | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| Heartbeat检测 | S13 | T8 | R4 |  |
| Ping-echo检测 | S13 | T8 | R5 |  |
| **理由说明** | 通过通信检测，系统能够尽早发现连接异常时间，并采取隔离资源，尝试重连的恢复措施。 | | | |
| **相关架构图** |  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A13** | 用户观看到高清，低卡顿的课程直播，并且能根据当前网速来自适应 | | | |
| **质量属性** | 易用性 | | | |
| **环境** | 运行时 | | | |
| **刺激** | 学生用户希望在观看讲师授课直播 | | | |
| **响应** | 系统直播视频清晰流畅，低卡顿 | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| 使用带宽检测 | S14 |  |  | N7 |
| 使用FFMPEG转码 | S15 | T9 |  | N8 |
| **理由说明** | 使用带宽检测技术，根据带宽能自适应直播清晰度，减少卡顿。使用FFMPEG转码，可以保证视频质量，同时提高了可移植性 | | | |
| **相关架构图** |  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **场景：A16** | 用户上传作业文件或音频文件 | | | |
| **质量属性** | 可维护性 | | | |
| **环境** | 系统正常运行 | | | |
| **刺激** | 用户希望系统支持更多的文件和音频格式 | | | |
| **响应** | 运维人员升级系统并发布更新通知 | | | |
| **架构决策** | **敏感点** | **权衡点** | **风险** | **非风险** |
| 划分模块 | S1 |  |  |  |
| 抽象接口 | S2 | T1 |  |  |
| **理由说明** | 进行模块划分，实现低耦合高内聚，可以有效地将修改和影响控制在一定的范围内，减少系统维护时的代码量，增强系统的可修改性和可维护性。  目前互联网上的文件、音频格式诸多，且仍处在变化之中，想要统一文件格式当前无法实现，除了支持常见的格式外，用户很可能希望系统能支持一些其他的格式。因此需要对具体的实现进行抽象接口的设计，以增强灵活性和可修改性。 | | | |
| **相关架构图** |  | | | |

* 1. **敏感点和权衡点**
     1. **敏感点**

|  |  |  |
| --- | --- | --- |
| # | 架构决策 | 理由 |
| S1 | 自动攻击侦测 | 外界恶意攻击出现时可以自动监测，有效提高了安全性，是安全性的敏感点。不间断的带来大量的服务器额外开销，对性能有负面影响，是性能的敏感点 |
| S2 | 使用DDOS分防火墙 | 对各种常见的攻击行为进行识别，并通过集成的机制实时对这些攻击流量进行处理和阻止，是安全性的敏感点，防止大规模恶意攻击导致系统瘫痪，是可用性的敏感点 |
| S3 | 对信息进行加密 | 有效保护用户信息安全，对安全性有积极作用，是安全性的敏感点 |
| S4 | 拒绝可疑访问 | 最大限度保证系统服务只被授权的合法用户访问，是安全性的敏感点。可能会误拒绝正常用户的请求，可用性的敏感点 |
| S5 | 剪裁外部接口 | 对各大支付平台的支付接口进行裁剪并统一向用户提供，是互操作性的敏感点，系统内交易接口可能受到攻击，对安全性有负面影响，是安全性的敏感点 |
| S6 | 服务列表定位 | 借助服务列表定位访问特定服务，是可用性的敏感点 |
| S7 | HLS协议 | 兼容性较好，是可移植性的敏感点。但有一定的时间延迟，也是性能的敏感点。 |
| S8 | BGP中转架构 | 可以较好地提高响应效率，降低响应时间，是性能的敏感点。 |
| S9 | 使用GlusterFS分布式文件系统 | 分布式文件系统可以有效提高系统存储的文件量，是容量的敏感点，提供冗余存储，当某些服务器故障时可以通过其他服务器上传，下载对应的文件，是可用性的敏感点 |
| S10 | 使用数据库集群 | 数据库集群可以有效提高系统存储的数据量，是容量的敏感点，  读写数据时可以提高效率，是性能的敏感点，备份恢复都很容易，是可用性的敏感点 |
| S11 | 使用反向代理 | 反向代理可以有效提高系统的负载性能，加快请求响应速度，是性能的敏感点，当某些服务器故障时可以通过反向代理将请求分发到其他服务器上，保证用户的正常使用，是可用性的敏感点，反向代理服务器容易成为被攻击的对象，所以也是安全性的敏感点 |
| S12 | 使用热备份与实时负载均衡服务器提供备用服务 | 热备份与实时负载均衡服务器可以避免服务器意外宕机引起服务不可用，是系统服务可用性的敏感点，用户访问服务额外经过负载均衡服务器会影响总体响应时间，可能提升服务速度，但也会多一次访问请求，也是性能的敏感点。 |
| S13 | Heartbeat/Ping-echo检测 | 检测机制可以保证尽快了解到资源不可用的情况，方便恢复，是故障检测恢复的敏感点；检测机制引入了额外的网络通信量，会占用带宽，是性能的敏感点。 |
| S14 | 使用带宽检测 | 及时检测用户带宽，根据带宽能自适应直播清晰度，减少卡顿，提高易用性 |
| S15 | 使用FFMPEG转码 | FFMPEG转码提供录制、转换以及流化音视频的完整解决方案，有助于播放清晰的视频，对用户体验具有极大的提高；并且其包含先进的音频/视频编解码库，平台型好，也提高了系统的可移植性。 |
| S16 | 模块划分 | 将系统模块化，可以在需要修改时有效地减少修改和影响的范围，是可修改性和可维护性的敏感点 |
| S17 | 抽象接口 | 对系统进行接口的抽象，可以在需要修改时只对必要的模块进行修改而不影响其他模块，是可修改性和可维护性的敏感点 |

* + 1. **权衡点**

|  |  |  |
| --- | --- | --- |
| # | 架构决策 | 理由 |
| T1 | 自动攻击侦测 | 提高了安全性，增加了服务器的额外开销，降低性能，是安全性和性能的权衡点 |
| T2 | 拒绝可疑访问 | 提高了安全性，可能会降低可用性，是安全性和可用性的敏感点 |
| T3 | 剪裁外部接口 | 提高了互操作性和易用性，降低了安全性，是互操作性和安全性的敏感点 |
| T4 | HLS协议 | 兼容性较好，但是会对响应造成少量延迟。是性能和可移植性的权衡点。 |
| T5 | 使用GlusterFS分布式文件系统 | 分布式文件系统有效提高容量，但是GlusterFS集群采用全对等式架构，每个节点在集群中的地位是完全对等的，集群配置信息和卷配置信息在所有节点之间实时同步，一旦配置信息发生变化，信息需要实时同步到其他所有节点，保证配置信息一致性，否则GlusterFS就无法正常工作，是容量和可维护性的权衡点 |
| T6 | 使用反向代理 | 反向代理服务器提高性能的同时也会容易成为被攻击的对象，是安全性和性能的权衡点 |
| T7 | 使用热备份与实时负载均衡服务器提供备用服务 | 增加实时负载均衡服务器改善服务提供但增加访问次数，是可用性与性能的权衡点。 |
| T8 | Heartbeat/Ping-echo检测 | 检测机制使用额外网络通信检测外部网络资源的健康程度，是可用性与性能的权衡点。 |
| T9 | 使用FFMPEG转码 | FFMPEG转码进程过大会占用过多CPU，影响性能，但考虑到系统需要的转码需求不是很多，故可不用太多考虑 |
| T10 | 抽象接口 | 提高了可维护性但是也增加了设计成本 |

* 1. **风险与非风险**
     1. **风险**

|  |  |  |
| --- | --- | --- |
| # | 架构决策 | 理由 |
| R1 | 自动攻击侦测 | 系统遭受攻击是小概率事件，一直进行不间断检测可能导致资源的浪费和性能的下降，影响性能 |
| R2 | HLS协议 | 会对响应造成一定的延迟，影响性能的实时性方面。 |
| R3 | 使用GlusterFS分布式文件系统 | 面对大量小文件的访问表现不佳，集群管理在规模较大时可维护性差，负载均衡功能不完善 |
| R4 | Heartbeat检测 | 接收来自资源的heartbeat，当资源应当空闲时仍然会接收到通信。如果资源种类多，会为服务器带来检测负担。 |
| R5 | Ping-echo检测 | 检测可以随着实际请求的发出顺便进行，每次检测至少使用两次通信，会为带宽带来额外负担。 |

* + 1. **非风险**

|  |  |  |
| --- | --- | --- |
| # | 架构决策 | 理由 |
| N1 | 拒绝可疑访问 | 多用户系统和涉密系统的标准配置 |
| N2 | 剪裁外部接口 | 通过安全控制和数据日志防止异常数据注入 |
| N3 | BGP中转架构 | 有效地提高了响应效率，改善了性能的实时性方面。 |
| N4 | 使用数据库集群 | 通过数据库集群提高系统的容量，在数据库服务器上添加安全模块可以防止非法访问、操作 |
| N5 | 使用反向代理 | 通过反向代理来实现负载均衡，提高系统的负载性能，在反向代理服务器上添加安全模块可以保证系统的安全性 |
| N6 | 使用热备份与实时负载均衡服务器提供备用服务 | 增加这些模块会改善可用性，而少量增加系统运行金钱成本。采用决策可以自建服务器或购买相关服务。 |
| N7 | 使用带宽检测 | 低成本，减少卡顿，提高易用性 |
| N8 | 使用FFMPEG转码 | 使用FFMPEG转码对视频高清以及可移植性具有较大优势 |

1. **挑战和经验**

本项目是iCourse网络课程平台，是一个基于web的学习与授课系统。但不同于市面上应用广泛的例如imooc，Coursera之类的网课平台，本系统融入了直播授课的新功能，以及对讲师实名注册等要求，因此，性能、安全性、可用性、互操作性、可修改性、可维护性、易用性是系统的首要架构驱动因素，我们一一对其进行分析考虑，并在最初提出了MVC和Shared-Data两种认为具有可行性的架构模式。

我们首先遇到的挑战是Shared-Data架构的一些不完备性，虽然不影响大局，但在设计后期，某些模块之间体现了不可避免的数据耦合，组员们通过查找资料，分析加上合理的设想最后完成了Shared-Data的相关架构设计；另一个挑战是系统功能的直播模块，由于大家之前都没做过类似的应用，所以对所有人来说，都是一块崭新的领域，组员们积极参与讨论，查阅大量直播相关的资料以及市面上现有的架构，解析各个模块的作用，比较各种协议、各种组件的合适与否，最终完成了直播模块的设计。

通过架构设计，我们对软件体系结构活动的整个流程有了真切的体验。在ADD中，我们通过亲自实践掌握了其要领。ADD是一个分解并迭代的过程，质量属性是其主要的驱动因素，我们将早期研究的质量属性作为输入，并在每一次的迭代中选择合适的战术和模式，对元素进行分配并实例化。在ATAM中，我们描绘效用树，列举出项目的敏感点，权衡点，风险和非风险并进行分析。这些都加深了我们在课堂中学到的知识，真正做到了学以致用。另外，我们通过研究自己的项目，明白了随着计算机网络基础设施(例如数据库技术、分布式计算存储技术)的演化升级，旧的技术会被淘汰，新的战略会被更新，从而说明软件架构是一个逐步求精、不断衍变的过程。

最后，通过尝试第一次八人大规模的组队实践，大家明白了团队合作与分工的重要性，体会到了其中的困难，也享受到了其中的乐趣，更加激发了所有人对软件架构设计的兴趣，可谓收获颇丰。

1. **组员和分工**

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 分工 |
| 董轶波 | 131250171 | 1. 负责iCourse MVC架构的设计，设计该架构的C&C视图并绘制 2. 负责ASR 互操作性和易用性的场景描写 3. 查找易用性的相关资料 4. 在ADD中负责MVC架构关于直播模块中的迭代，编写相关的产物 5. 在ATAM中负责易用性的编写 6. iCourse项目的界面原型总设计，主页编码，并用框架将所有view组织 |
| 王子安 | 141250146 | 1. iCourse MVC架构的设计和模块视图的绘制 2. 负责ASR可修改性的场景描写，并查找容量性能相关的资料 3. 在ATAM中负责互操作性的编写 4. 参与iCourse项目中课程注册、视频观看等界面原型代码 5. 本文档的汇总工作、格式调整和版本更新 |
| 徐江河 | 141250161 | 1. 参与Shared-Data架构模块视图、组件-连接件视图的修改 2. 编写ASR描述中的“可用性”部分 3. 编写架构比较与最终选择中“可用性”部分 4. 编写Shared-Data架构ADD第四次迭代过程文档部分 5. 编写ATAM分析中“可用性分析”部分 |
| 袁楚宏 | 141250178 | 1. Shared-Data的架构设计 2. ASR的安全性部分 3. 类图设计，组件/连接件到实现类的映射 4. 安全性的实现技术 5. data-shared第二次迭代的易用性和实时性部分 6. ATAM的实时性部分。 |
| 袁阳阳 | 141250179 | 1. 负责学生和管理员的功能点和操作场景描述的编写 2. 负责ASR实时性的场景描述，并在技术和策略选择中查找提高实时性的相关资料 3. 在ADD过程中负责MVC架构关于课程注册模块的迭代以及相关产物的编写 4. 在ATAM过程中负责可维护性相关部分的编写 5. 负责关于直播部分view层的代码编写 6. 与架构人员共同讨论和决定架构策略和相关技术决定 |
| 张诗琦 | 141250190 | 1. 主要功能点和操作场景分析的教室功能部分 2. 非功能需求及ASR描述的容量性能部分 3. Shared-Data架构的ADD第三次迭代 4. ATAM分析过程中的安全性部分 5. 原型构建中的教师功能部分。 |
| 张鑫龙 | 141250195 | 1. 非功能需求及ASR描述的可维护性部分 2. 可维护性，可修改性的实现技术 3. MVC架构的ADD第四次迭代 4. ATAM分析中的容量性能部分 |
| 张云飞 | 141250197 | 1. 负责share-data架构的设计，share-data架构的模块视图，组件-连接件视图的绘制和修改 2. 负责ASR性能负载的场景描写，查找相关资料完成负载相关的技术选择部分 3. 在share-data架构ADD第二次迭代中负责可用性和负载性能部分 4. ATAM中负责负载性能部分。 |