**迭代评估报告**

　　　　　　　　　　　　　　　　　　评估日期：11.17

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 17 | 项目名称 | 云作业平台 |
| 迭代名称 | 迭代二：技术原型迭代 | 实际起止日期 | 10.17-11.17 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）  **前端返工：**   1. 统一了各个页面的样式，修改了UI元素的摆放位置。 2. 错题页面添加了历史作业错误率的统计图表。 3. 课程统计页面修改了几张图表。   **功能方面：**  1.鉴权：完成了用户注册登录的功能并完善细节。可以检验手机号与邮箱的格式有效性与唯一性，并且完成向邮箱或手机发送验证码的流程，根据验证码完成注册。  2.信息管理：用户可以通过个人信息页面管理、修改信息。个人信息存放在后端鉴权数据库中。其中教师用户可以上传教师半身照，该照片会展示在课程信息页面中。  3.课程：  3.1 课程信息展示：正确展示完整课程信息，包括课程简介、课程大纲、授课教师、老师寄语、开始截止时间、授课班级，课程公告，课程作业，统计信息等。  3.2 课程管理：教师用户可以创建课程、删除自己创建的课程。除了填写上面的课程信息外，还能够指定课程中的通知发布逻辑与权限，比如禁止学生查看课程平均分，发布作业时自动发布通知等。  3.3 课程权限控制：一门课的授课内容可以对所有用户公开，但是不属于该门课的用户(无论学生或老师)只能查看基本信息、公告和作业。授课教师用户可以修改自己所创建课程的课程信息，也可以查看所有上课同学的成绩信息；上课同学只能查看自己的成绩信息和统计数据。  对于已截止的课程，授课教师仍可以查看信息，但不能再修改。  3.4 课程公告管理：授课教师可以发布、修改与删除课程公告，可以选择是否发送通知。若发送通知，则所有上课学生都会收到公告发布的邮件通知。  3.5 课程成员管理：授课教师可以添加与删除课程用户，其中可以批量从班级中添加学生用户。  3.6 课程统计信息：课程中有对一名学生用户作业的基本统计信息。包含了作业平均得分、近来作业平均得分、排名、作业完成时间变化等，方便教师进行统计。  4 作业：  4.1  4.2  4.3  4.4  4.5  4.6 作业批改：授课教师可以批改该课程中的作业。教师可以快速切换学生进行作业批改，可以在批改时一键查看作业内容与作业答案，一键评论评分，并实时显示该作业的平均分方便教师控分。教师可以在学生的作业上进行批注，批注支持以图层的形式与图片一起保存。  5 班级管理：班级是学易云作业平台的重要概念。教师可以创建班级，自动成为该班级的班主任。班主任可以修改班级信息，添加或删除成员等。  **文档与建模：**  1.编写了以2-3日为单位的迭代计划  2.完成了迭代评估报告。  3.完成了学易-云作业平台的总模型文件，包括物理视图，逻辑视图，实现视图，数据视图，进程视图以及六个用例实现。  4.完成并优化了架构文档。  5.运用GOF设计模式对大作业进行详细设计，列出抽象工厂、适配器两种设计模式在系统中的使用，画出了它们的UML类图。  6.优化了Use-case图与概念模型。  7.设计了核心算法：作业文件的传输与压缩。 | | | |
| **评审/测试的结果：**  文档评审：所有文档均经过两次评审，一次是提交前的修改和打磨，另一次是提交后周一下午的反思和优化。经评审，学易-云作业平台的各项文档均合格完成。  模型评审：概念模型，用例规约模型根据小组成员的实际代码进行实现，经过多次增删、完善，由多位小组成员给出修改意见，最终构建出成熟的模型。  **功能测试：**  对新用户注册，老用户登录，学生提交作业，老师批改作业，老师发布作业，老师统计作业，老师创建课程，错题集，通知信息，班级管理，图像识别等功能做了全面的人工测试。 | | | |
| 问题、变更和返工：  **问题：**  技术问题  本次迭代中遇到了大量的技术问题，以下列举几个关键问题与解决方案。   1. 邮件服务出现故障：我们的邮件服务和短信服务采用的是阿里云提供的接口，在调试时发现网易云邮箱只能给自己发邮件，而无法给其他邮箱发邮件。为了不影响项目推进，我们先将用到邮件服务的地方换成短信服务，保证提醒功能，之后再逐步修复邮箱相关的问题。 2. Eureka的服务发现问题：在配置了注册中心(Eureka Server)和在各个服务上配置了服务提供中心(Eureka Client)后，发现已有的服务并无法正确注册到注册中心上。在确认了配置yml文件后，发现是未指定每个服务的名称导致注册中心不能正确发现服务，在yml中添加服务名称后成功解决。 3. 前后端文件传输问题：这个问题困扰了我们小组许久。将前端文件用MultipartFile传输，存放都后端后，一开始从后端无法读取到相应的文件，后来发现是docker的文件夹映射机制导致了根目录的变化。接着发现从后端以MultipartFile形式向前端发送文件总是失败，于是采用文件流，却会因为文件过大而报错。调整配置参数后，正确拿到了文件流，但是无法将其转换成图片显示。最后我们将其提前转为了base64格式，实现了前后端图片文件的传输。   架构与文档问题：   1. Powerdesigner使用问题：进入第二阶段的迭代后，需要大量使用PowerDesigner进行架构和建模，因此组里原本主要负责前端框架的成员也参与了建模的工作。由于对软件不够熟练，短期迭代任务内交付的成果不尽人意。针对此，我们让组里熟练使用该软件的同学一对一指导和帮助不熟练的成员，顺利解决了问题。 2. 对微服务架构的理解偏差：一开始时我们认为只要将服务分离，数据库分库就是微服务架构，后面经课上老师的指正，我们发现之前的逻辑架构和物理架构等都没有很好的体现微服务的思想。于是我们根据课上PPT与网上资料，改进了项目的各种架构与视图，添加了注册中心，网关等机制，搭建了基础的微服务架构。   **变更和返工：**  由于组员之间沟通较好，本组没有大规模的变更和返工情况发生。每周一会根据课程内容，对前一周完成的任务进行细微修改。 | | | |
| 经验和教训：  1. 技术是为功能服务的。我们应该想着如何去实现一个大的功能，之后再逐步细化要用哪些手段去达成这个功能，而不是一定要在项目里使用某个技术，轻则画蛇添足，重则影响开发的效率。例如让教师从图片提取文字来布置作业的功能，重点是思考如何简化教师用户布置作业的流程，提升用户体验，而非是一定要用某种AI框架来解决问题。  2. 提前攻克技术难点。本次项目的技术难点主要是作业批改部分，包含了主观题的批注和选择题的自动批改部分。这部分我们从比较早就开始研究，其中失败了许多次，遇到了各种技术问题，花了两周才完成功能。而相对机械化的任务例如用例实现则可以放在后期完成。  3. 先交流，后查询资料。一位组员遇到的技术难题可能另一位组员早已遇到过，此时多进行组内交流可以大大提升开发的效率。如果总是遇到任何问题都闷头查询资料，不仅会花费大量的时间同时也会影响技术实现的一致性。 | | | |