学易-云作业平台

软件架构文档

版本 <2.3>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2020/11/01 | <1.0> | 初稿 | 刘兆翰 |
| 2020/11/07 | <2.0> | 结构，文字大幅改动 | 杨亘 |
| 2020/11/09 | <2.1> | 改动文档 | 杨亘，刘兆翰 |
| 2020/11/10 | <2.2> | 改动文档 | 刘兆翰 |
| 2020/11/17 | <2.3> | 模型修改 | 杨亘 |

目录

1. 简介 2

1.1 目的 2

1.2 参考资料 2

2. 用例视图 2

3. 逻辑视图 2

3.1 概述 2

3.2 在构架方面具有重要意义的设计包 2

4. 进程视图 2

5. 部署视图 2

6. 实现视图 2

7. 数据视图 2

8. 核心设计思想 2

9. 核心算法设计 2

10. 关键质量设计 2

软件架构文档

# 简介

本文档使用逻辑视图、物理视图等多个视图定义了学易-云作业平台的架构全貌，记录了学易-云作业平台最重要的设计决策，并成为随后的详细设计与实现工作的战略指导原则。其中逻辑视图采用的是分层架构模式，而物理视图采用的是B-S架构模式。

## 目的

本文档使用许多不同的体系结构视图来描述学易-云作业平台系统的不同方面，从而提供了学易-云作业平台的全面体系结构概述。它旨在捕获和传达已在系统上做出的重要体系结构决策。

## 参考资料

云作业平台接口文档.doc

用例规约.doc

学易-云作业平台概念模型.oom

学易-云作业平台Use Cases.oom

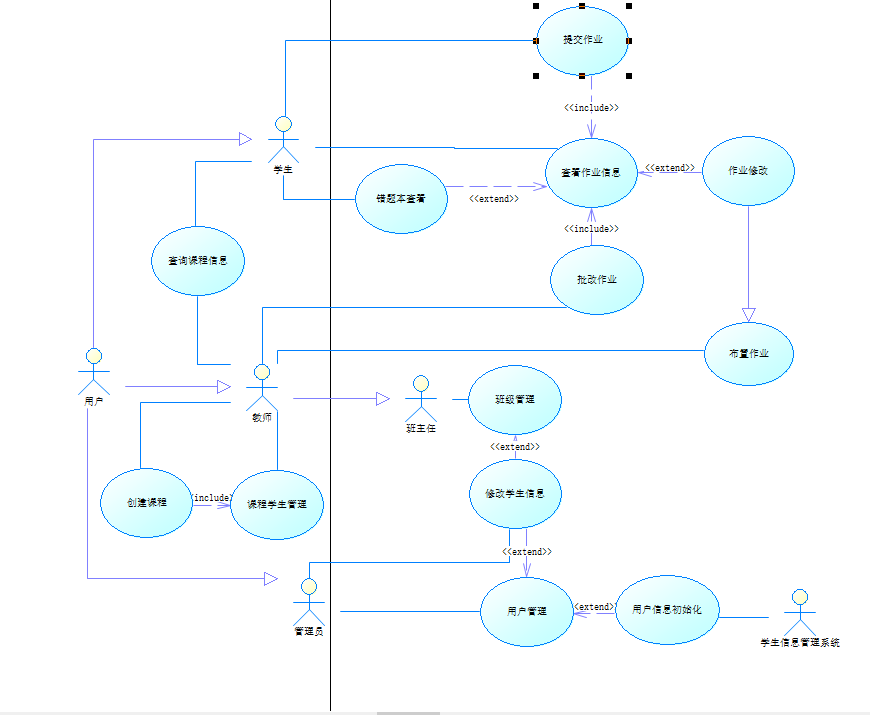
学易-云作业平台系统软件开发计划.doc

学易-云作业平台迭代计划.doc

学易-云作业平台总模型.oom

# 用例视图

**2.1主要用例关系视图**

****

2.1.1查询课程信息

简介：该用例允许学生和教师（管理员）两类用户查询名下课程的信息，信息包括课程大纲、课程进度、课程作业等，可以查询的课程有正在上的课程也有已经结课的历史课程。

2.1.2用户管理

简介：该用例支持平台管理员统一管理平台上的所有用户，包括学生以及教师，管理员可以由本地的excel文件批量导入用户信息名单或者申请从合作院校的教务系统导入。

2.1.3创建课程

简介：该用例允许教师创建一个课程。

2.1.4课程学生管理

简介：该用例允许教师在创建课程后对课程的上课学生进行管理，包括从备选名单中添加学生、从课程名单中删除正在上课的学生。

2.1.5布置作业

简介：该用例允许教师布置作业发布到平台上，并且可以修改作业的题目内容。

2.1.6批改作业

简介：该用例允许教师获取学生提交的作业，系统自动批改客观题并且由教师对主观题进行评分批注和评语。

2.1.7班级管理

简介：该用例与管理员的用户管理内容相似，不过权限范围不同。该用例允许班主任统一管理所在班级的学生信息，包括学生添加、删除、信息修改。

2.1.8提交作业

简介：该用例允许学生查看作业内容并提交作业到平台上。

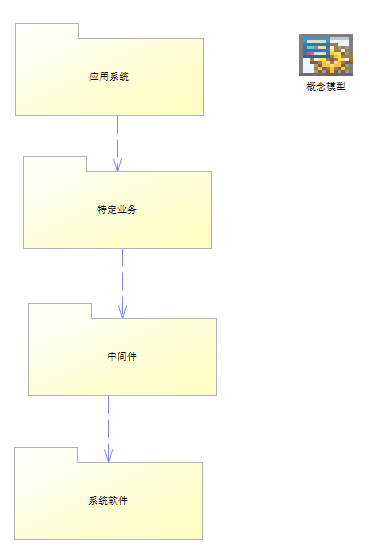
2.1.9错题本查看

简介：该用例允许学生查看自己以往错题汇编成的错题集，错题集由系统自动生成。

# 逻辑视图

## 概述

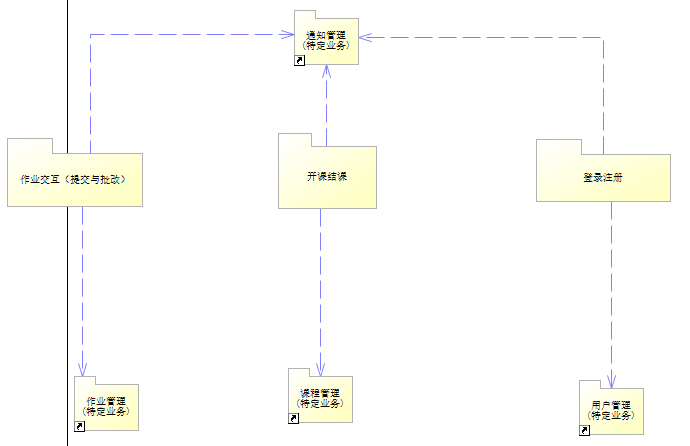
云作业平台的逻辑视图采用分层架构，由4个主要包组成：应用系统、特定业务、中间件、系统软件。



## 在构架方面具有重要意义的设计包

**3.2.1应用系统层**

应用系统层包括了学易-云作业平台中的部分项目干系人所涉及到的核心业务，包括作业交互，开课结课，登录注册三个部分。每一个核心业务中都包含十几个至数十个请求接口。



**5.1.2特定业务层**

特定业务层是核心业务层的具体细化，包括了课程管理、作业管理、通知管理、用户管理四个部分，其中用户管理还包括班级管理。这些是为了实现上层的核心业务所必须实现的基础业务。

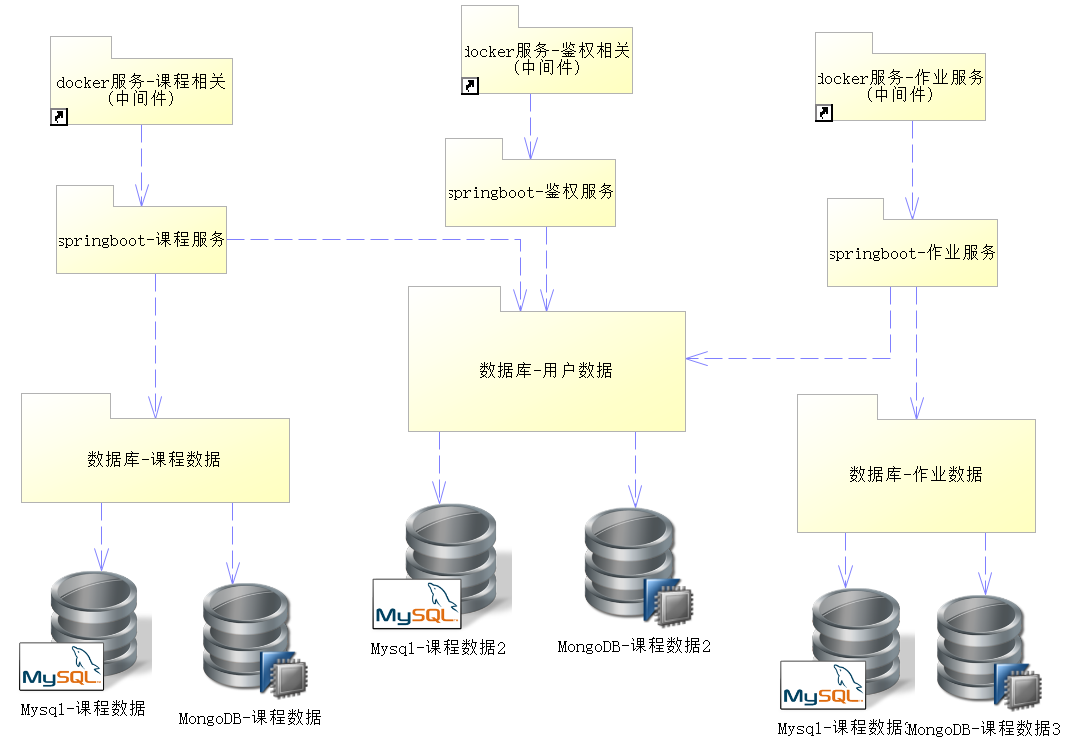
**5.1.3中间件层**

中间件层由网关和docker集群组成。用户的所有请求将会先经过网关后发往各个微服务，每个微服务由docker 包装并部署在服务器上，由docker swarm进行管理。



**5.1.4系统软件层**

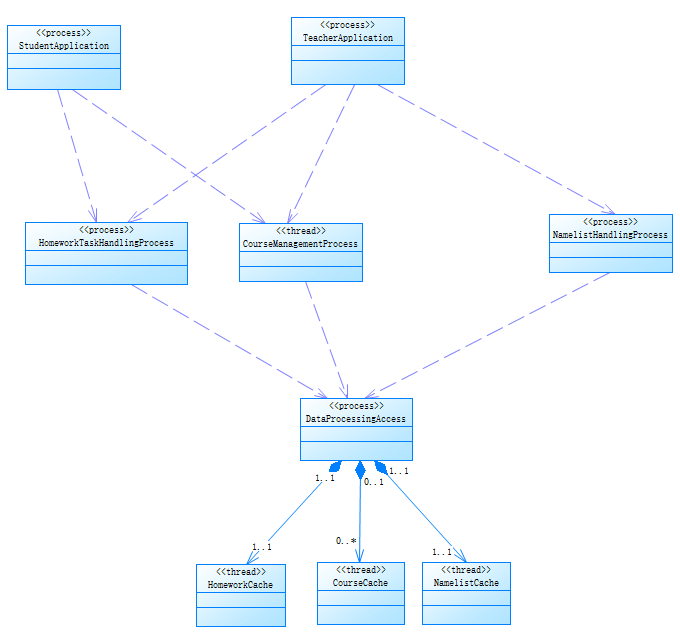
该系统软件层包含支持所有功能的Mysql关系型数据库和MongoDB非关系型数据库，以及三个springboot后端微服务。不同服务的数据储存在不同的数据库上，加快了数据访问的一致性和效率。



# 进程视图

对体系结构过程视图的描述。描述系统执行中涉及的任务（进程和线程），它们的交互作用和配置。还描述了对象和类对任务的分配。

4.1进程总视图Process



4.1.1

Teacherapplication, Studentapplication：

请求产生进程，包括与学生教师等用户交互的界面功能。

对于每一个用户都有一个UserApplication的进程与之对应。

4.1.2

CourseHandlingProcess, HomeworkHandlingProcess，NamelistHandlingProcess是请求处理进程

对于每一个正在使用课程/作业/鉴权功能的用户，都有一个请求处理进程与之对应。

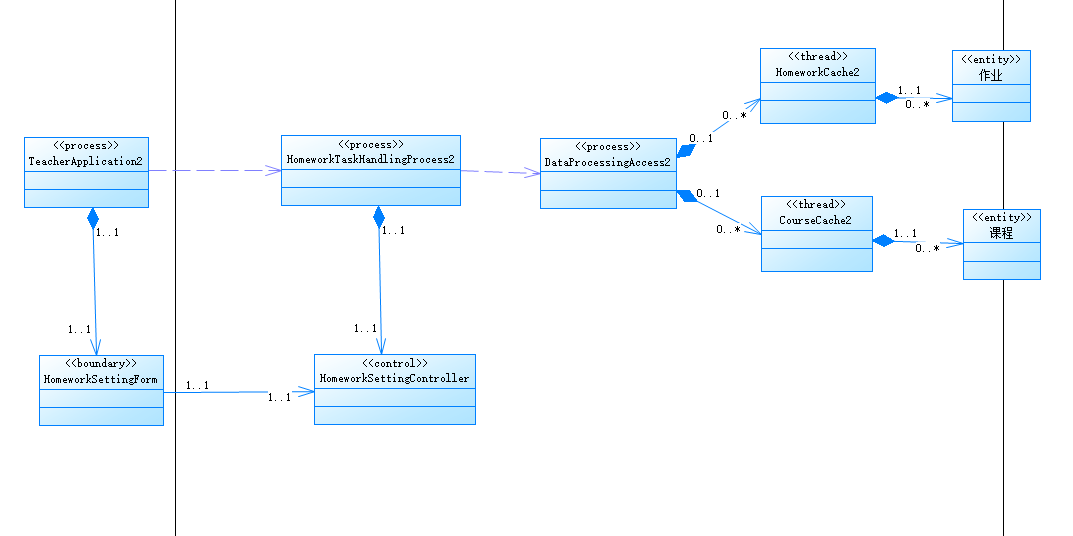
4.1.3

Dataprocessingaccess进程管理对于数据库系统访问。被所有正在访问的平台用户共用。

4.1.4

HomeworkCache、CoursesCache、NamelistCache分别用于异步处理到达后端服务的作业、课程、鉴权项目。

4.2轻量级进程-布置作业



# 部署视图

5.1 学易-云作业平台的物理视图，用户通过各种可以访问互联网的设备，在网页端访问云作业平台。用户在操作云作业平台时所发送的请求会通过网关分发到各个服务器上，再与各个数据库交互。其中数据库包含关系型数据库和非关系型数据库。以下会阐述每个部分的作用以及不同部分之间的交互。



5.2家用台式机/平板电脑/笔记本电脑

学生和教师通过本地PC机登录平台并使用云作业功能，这些台式机通过LAN直接连接到College Server。注册服务商使用这些本地PC来维护学生和教师的信息。

5.3作业服务器

作业服务器负责作业部分的数据库与用户端的对接，所有学生教师可以通过Internet连接到作业服务器。

5.4课程服务器

作业服务器负责课程部分的数据库与用户端的对接，所有学生教师可以通过Internet连接到课程服务器。

5.5鉴权服务器

作业服务器负责登录注册部分的数据库与用户端的对接，所有学生教师可以通过Internet连接到鉴权服务器。

5.6用户信息数据库

记录平台用户的所有信息支持后端服务，包括用户注册信息、课程信息、作业信息等

5.7 网关

用于接收和分发来自前端用户的请求，同时网关也会注册到注册中心上。

5.8 注册中心

用于服务发现与服务注册，不同微服务间相互调用时的中介平台。

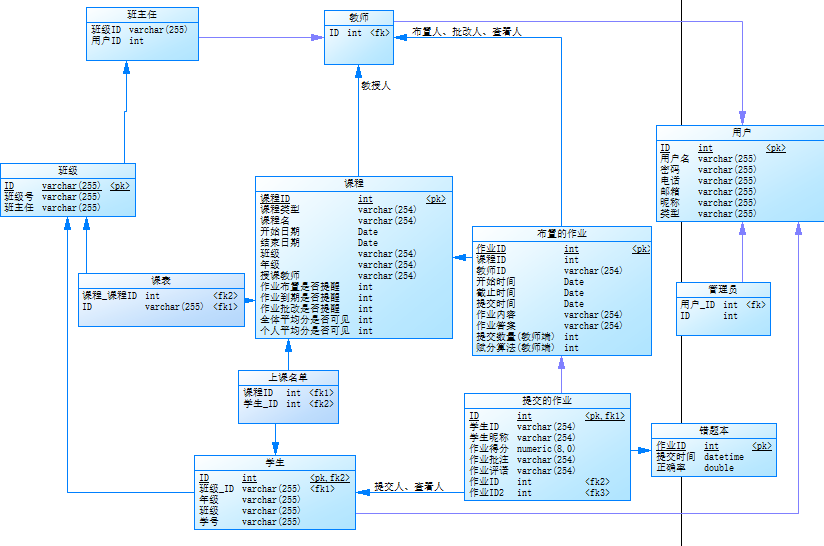
# 实现视图

学易-云作业平台的实现视图主要分为三个部分，一个为前端的客户端组件，包含了网页图形界面以及页面逻辑与应用。这个组件会通过指定的端口向三个主要服务组件：鉴权服务器组件，课程服务器组件，作业服务器组件发起请求。其中作业服务器组件和课程服务器组件还会通过智能服务接口调用一些第三方提供的外部服务（例如图像识别服务）。



# 数据视图

数据视图在学易-云作业平台中是较为关键的，平台有责任与义务持久化好每一位用户的使用数据，保证学习平台数据的一致性与数据统计的方便性。数据视图由概念模型生成，并在概念模型的基础上进行了细化，包括主键和外键的完善，数据冗余的添加。



# 关键设计思想

**性能方面：**

## 采用微服务的方式来设计和部署后端服务：大作业中要求支持1000个并发用户，（主要接口的）最长响应时间为3秒，因而一项服务涉及的工作不能太多，否则以项目经费所能承担起的服务器无法保证高并发低时延的要求。经过仔细观察，本云作业平台的服务可以基本被划分为鉴权服务，作业服务，课程服务三个方面，不同服务（模块）间低耦合，服务内部高内聚，因而我们全线采用了微服务的设计思想和设计理念，可以有效提高服务的质量和速度。

## 同样为了达到高并发低时延的要求，我们在设计sql时对结构化数据进行了大量冗余操作，保证简单的sql语句就可以拿到所需数据，无需大量的join操作，同时不会造成过多的储存冗余。举个例子：错题本相关服务的后端实现中，本来要先拿到学生完成的作业，根据作业id获取原本的作业内容，再与用户的表去做join，这样一来性能损失很严重。于是我们把老师布置的作业的部分字段放在了学生提交作业的表中，再把部分用户信息也放在了同一张表中，极大提升了性能。

# 核心算法设计

暂无

**10.关键质量设计**

|  |  |
| --- | --- |
| **质量因素** | **设计战术** |
| 易用性 | 1.每个页面附有指导性说明，方便不常上网的中小学生和老教师使用。  2.侧边栏提供导航式功能快捷入口，课程与作业全部详细功能项目被收录其中。  3.课程、作业、用户信息分模块处理，提高了并发兼容性，缩短了使用时的页面反应时间和文件传输时间。 |
| 可靠性 | 1.平台设置多个管理员监测系统运行，在运行故障时系统向管理员发送错误响应。  2.软件前后端所有功能在投入应用前分别在服务器上通过可靠性性能测试。  3.系统实现代码规定对输入的非法数据（如格式错误的学生信息）进行严格的筛选，避免查询时产生可能的异常。 |
| 性能 | 1.对结构化数据进行了大量冗余操作，保证简单的sql语句就可以拿到所需数据，无需大量的join操作，同时不会造成过多的储存冗余。极大提高了查询效率。  2.采用微服务模式，课程服务、作业服务、用户信息鉴权服务分模块处理，提高了后端运行的并发性。 |
| 安全性 | 1.登录注册采用邮箱验证+验证码检测的方式，确保用户登录的安全性。  2.在投入使用时，学生信息需要与从学校教务系统导入的信息匹配并完成验证才能参加课程，确保了授权的一致性。 |
| 可测试性 | 1.接口与实现可分离，前后端所有功能相对简单明了，可通过给定输入快速显示故障所在。  2.鉴权、课程、作业服务模块之间低耦合、高内聚，具有良好的可分解性来分别进行单元测试。  3.源代码注释详尽。接口文档功能、实现及参数描述清楚，易于测试者理解。  4.后端功能代码中使用大量断言语句来检查程序正常运行时不应发生但在调测时有可能发生的非法情况。 |
| 可维护性 | 1.软件功能模块、接口、功能内部过程清楚易理解，源代码注释详细、设计文档清晰详尽，使得云作业平台软件有很强的可理解性。  2.使用通用版本Mysql和MongoDB进行数据的日常维护，在应用中由管理员进行定期检查修改。  3.三个主要的微服务之间低耦合，在修改其中一个的功能与数据库时可以维持其他服务的完整独立性，实现局部化修改。 |