软件体系结构文档模板

文档修改历史

修改人员	日期	修改原因	版本号
郑伟鑫	2020.07.08	完成逻辑视图、开发包图、模块职责	V1.0
王一辉	2020.07.09	完善逻辑视图、开发包图、用户层界面分解	V1.1
王一辉	2020.07.09	完成用户界面层分解、业务逻辑层分解、数据层分解、信息视角	V1.2
吴子玥	2020.07.09	完善业务逻辑层分解	V1.3
王一辉	2020.07.10	修正了模块划分与模块列表的内容	V1.4

1.引言

1.1编制目的

本报告详细完成对CourseLearning在线课程学习平台的概要设计，达到指导详细设计和开发的目的，同时实现和测试人员以及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

1.2词汇表

词汇名称	词汇含义	备注

1.3参考资料

（1）IEEE标准

（2）《软件工程与计算（卷二）软件开发的技术基础》

（3）courseLearning 系统用例文档（仅迭代三）

（4）courseLearning 系统需求规格说明文档 （仅迭代三）

2.产品概述

参考CourseLearning系统用例文档和软件需求规格说明中对产品的概括描述。

注明：本文档仅包括迭代三需求的相关说明

3.逻辑视图

- 处理静态设计模型

  CourseLearning在线学习平台系统中，选择了**分层体系结构风格**，将系统分为4部分（presentation, controller, service, data）能够很好地示意整个高层抽象。Presentation部分包含了GUI页面的实现，Controller部分负责接受前端发送的请求并分发给相应的Service，service部分负责业务逻辑的实现，data部分负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如下图所示。

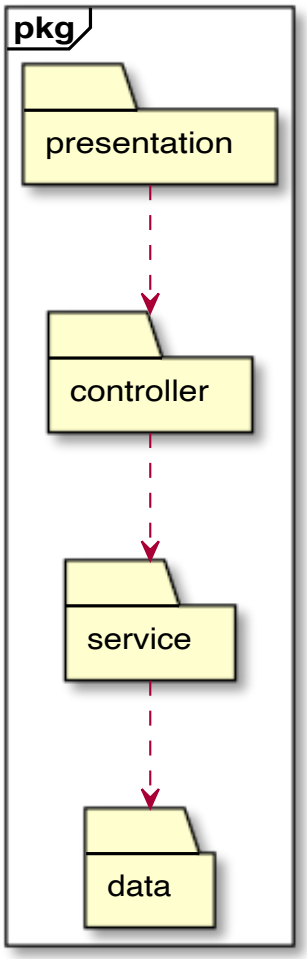


图1　参照体系结构风格的包图表达逻辑视角

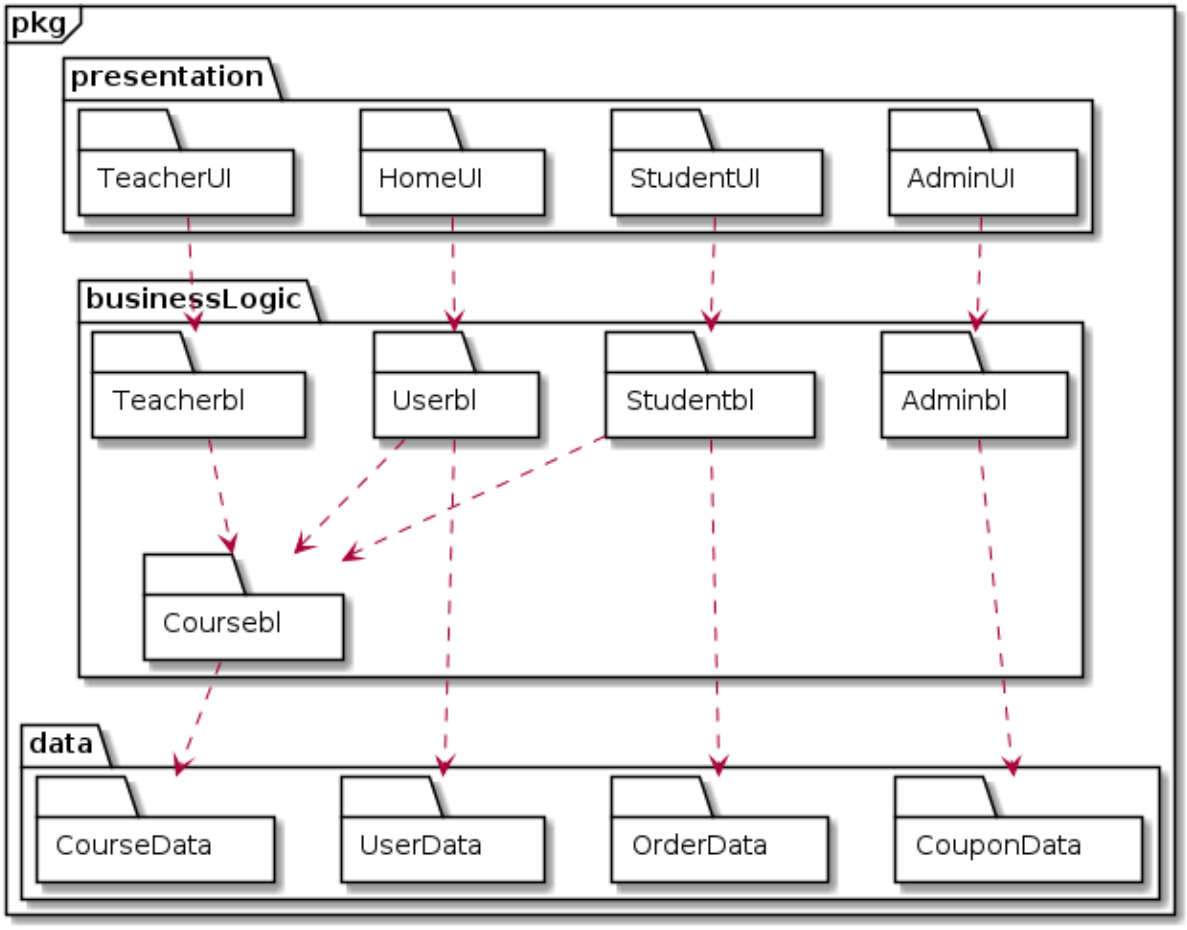


图2　软件体系结构逻辑设计方案

4.组合视图

4.1开发包图

- 表示软件组件在开发时环境中的静态组织
  - 描述开发包以及相互间的依赖
  - 绘制开发包图

开发（物理）包	依赖的其他开发包
ReviewUI	ReviewController，Vue组件库包，vo, RESTAPI
CommentsUI	CommentsController, Vue组件库包，vo, RESTAPI
AssignmentsUI	AssignmentController，Vue组件库包，vo, RESTAPI
TestUI	TestController，Vue组件库包，vo, RESTAPI
RechargeUI	RechargeController，Vue组件库包，vo, RESTAPI
UserUI	TestController，Vue组件库包，vo, RESTAPI
RentOrderUI	RentOrderController，Vue组件库包，vo, RESTAPI
ReviewController	ReviewService

CommentsController	CommentsService
AssignmentController	AssignmentsService
TestController	TestService
RechargeController	RechargeService
UserController	UserService
RentOrderController	RentOrderService
TestService	TestMapper
AssignmentService	AssignmentMapper
CommentsService	CommentsMapper
ReviewService	CommentsMapper, ReviewService, InformService
InformService	InformMapper
RechargeService	RechargeOrderMapper
RentOrderService	RentOrderMapper, UserService, CourseOrderService
UserService	RentOrderService, UserMapper, InfromService
CourseOrderService	CourseOrderMapper
CourseOrderMapper	Datausability, po
TestMapper	Datausability, po
AssignmentMapper	Datausability, po
CommentsMapper	Datausability, po
ReviewMapper	Datausability, po
InformMapper	Datausability, po
RechargeOrderMapper	Datausability, po
RentOrderMapper	Datausability, po
UserMapper	Datausability, po
VO	
PO	
Vue组件库	
Java RMI	

Databaseutibility	JDBC, Mysql
RESTAPI	

courseLearning平台客户端开发包图如图3所示，服务器端开发包图如图4所示。

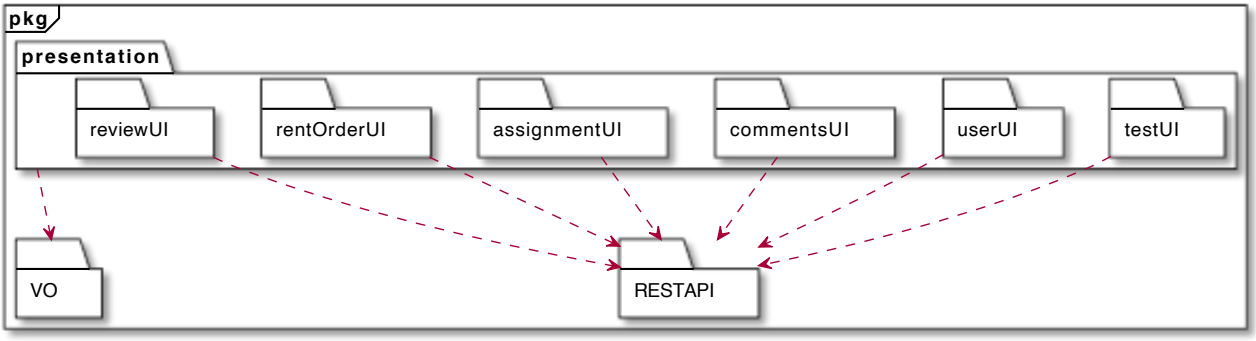


图3 courseLearning平台客户端开发包图

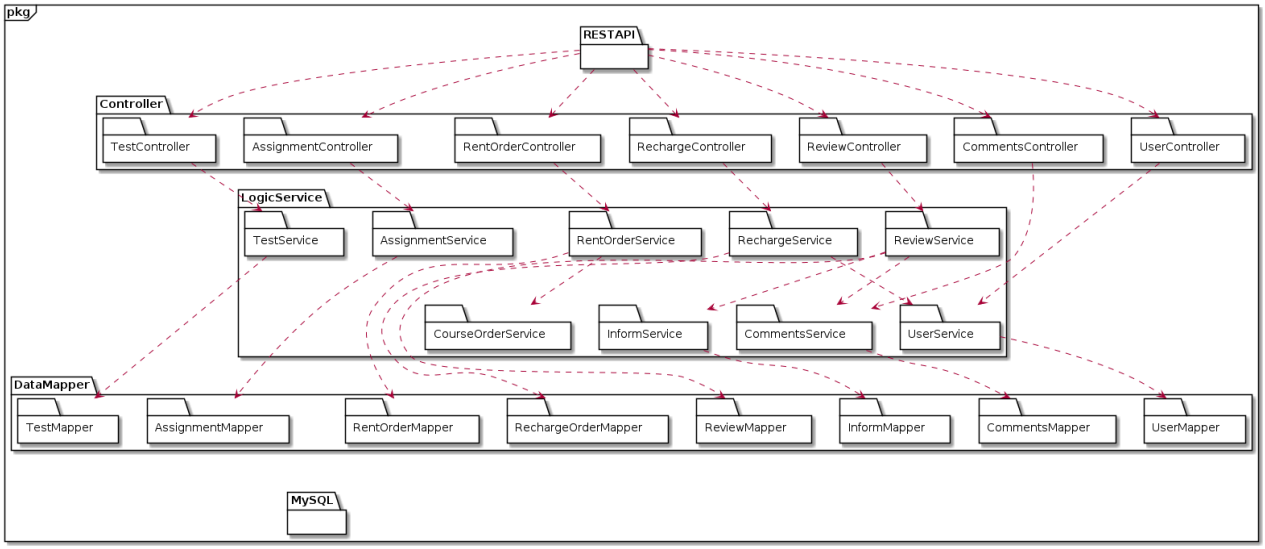
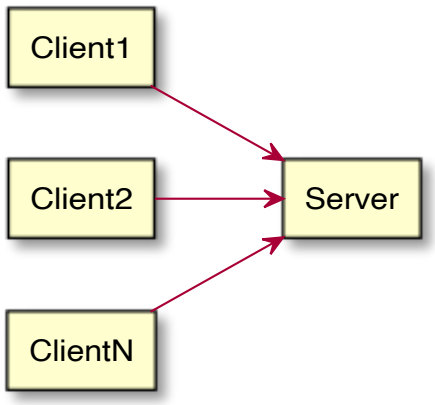


图3 courseLearning平台服务端开发包图

4.2运行时进程

- 示意图：



4.3物理部署

- 处理如何将软件组件映射到硬件基础设施
- 部署图

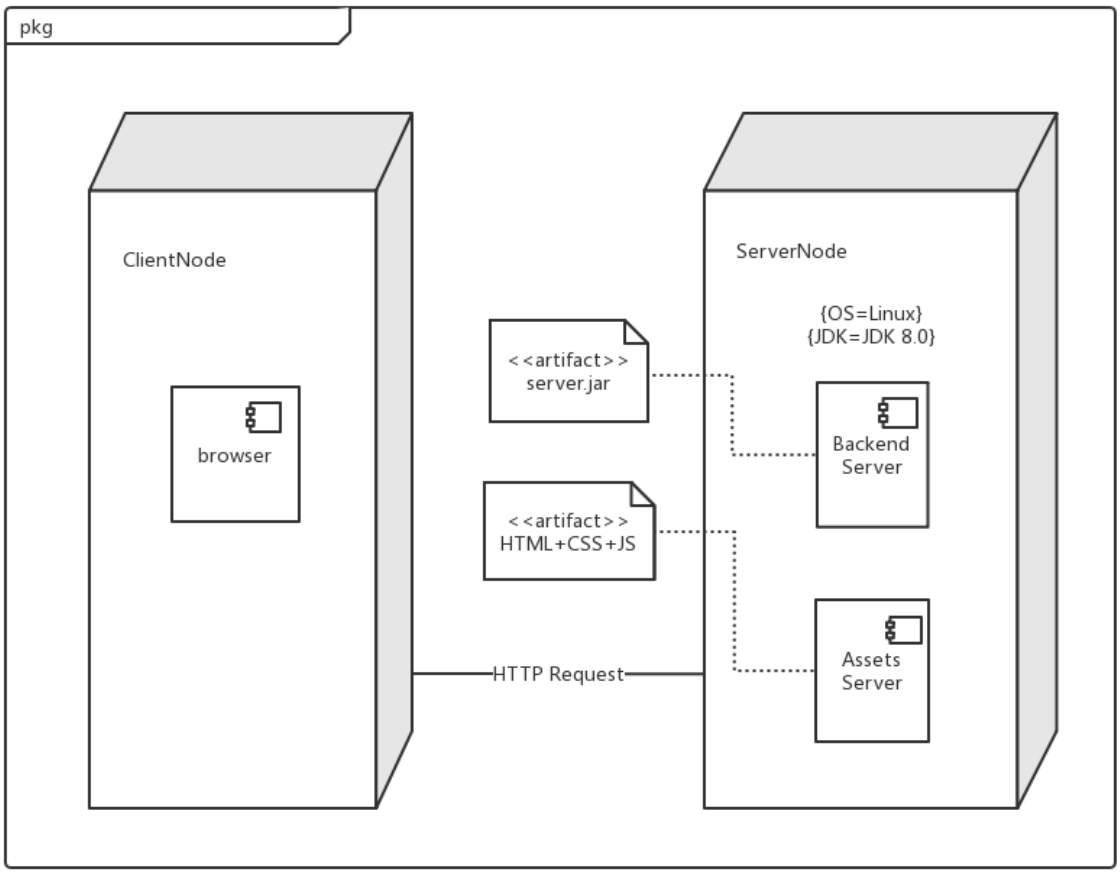


图5 部署图

5.架构设计

- 描述功能分解和如何在不同的层中安排软件模块
  - 描述架构中的对象，包含架构图
  - 描述组件接口信息
    - 包括：语法、前置条件、后置条件

5.1模块职责

客户端模块和服务端模块视图分别如图1和图2所示。客户端各层和服务端各层职责分别如表1和表2所示。

- 模块视图

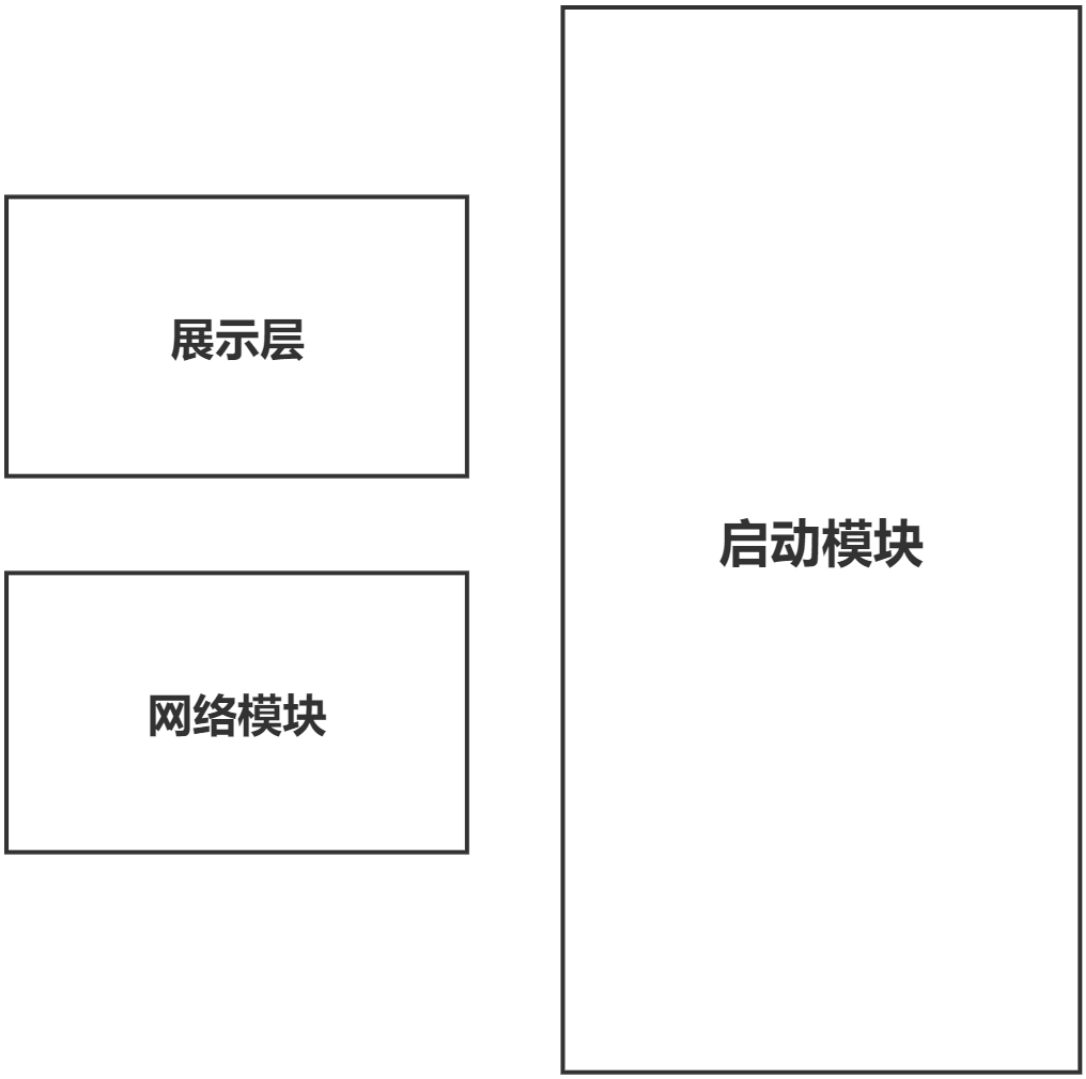


图1 客户端模块视图

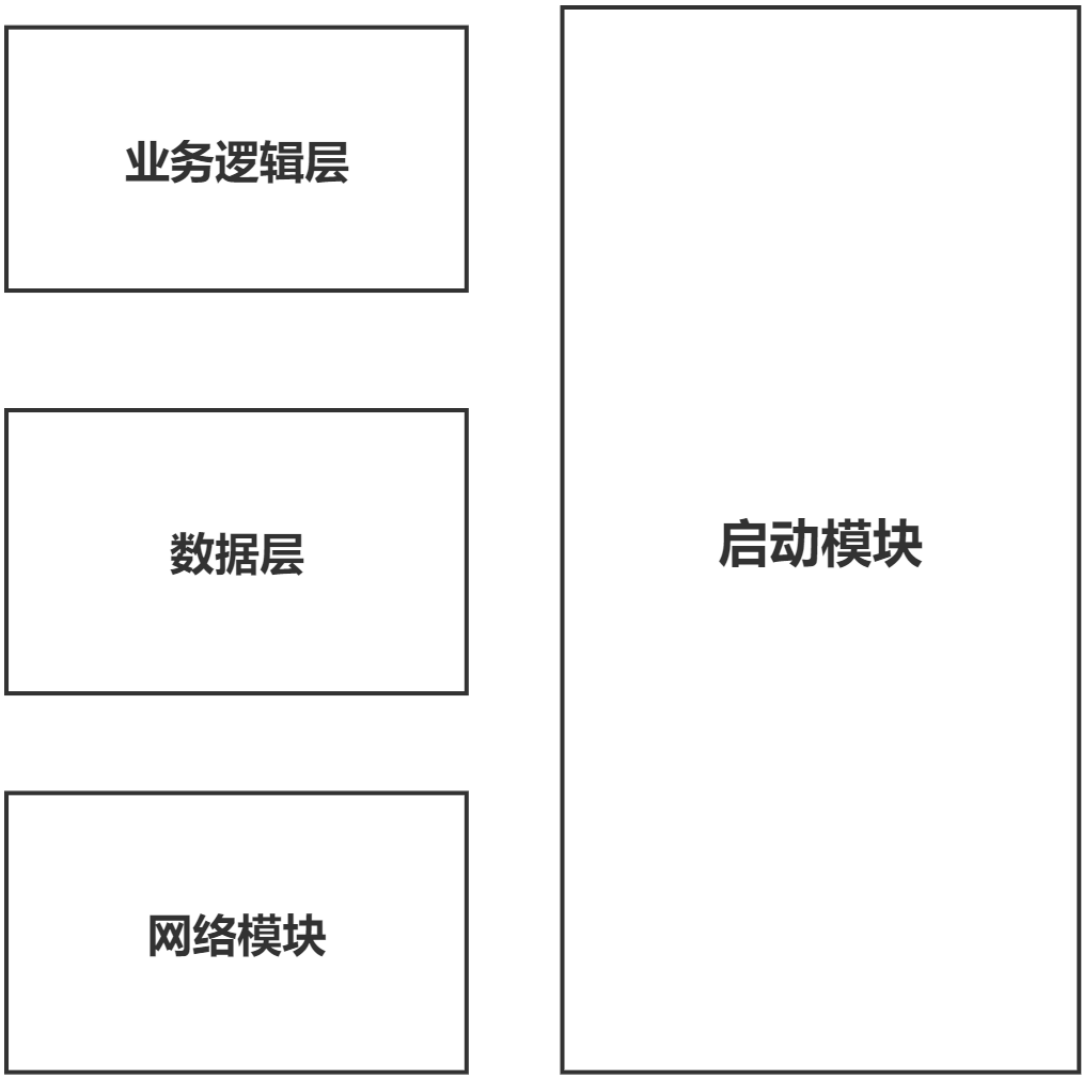


图2 服务器端模块视图

- 表1　客户端各层职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
用户界面层	courseLearning客户端用户界面，使用Vue.js框架实现
客户端网络模块	使用RESTful风格接口通过HTTP请求实现前后端通信

- 表2　服务器端各层职责

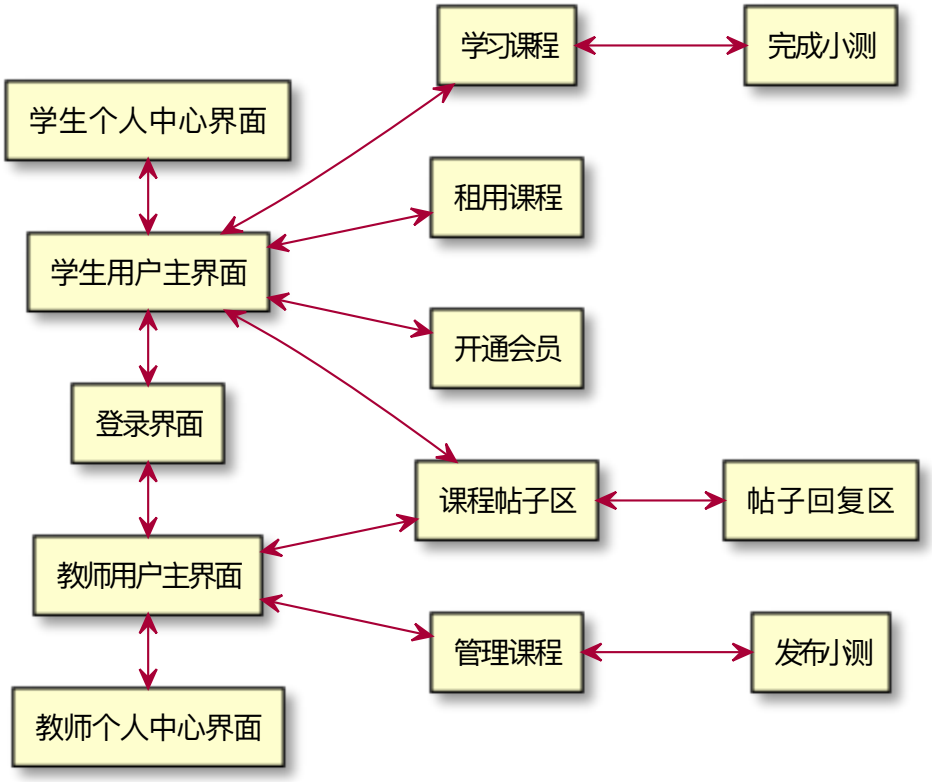
层	职责
启动模块	负责初始化网络通信机制，启动后端服务器
业务逻辑层	对于用户界面的输入进行业务处理逻辑
数据层	负责数据的持久化即数据访问接口
服务器端网络模块	通过RESTful风格的接口处理前端发出的请求

- 层之间调用接口

RESTAPI	客户端展示层	服务端控制层
CommentsService UserService RentOrderService AssignmentService CourseOrderservice TestService ReviewService InformService RechargeService	服务端控制层	服务端业务逻辑层
CommentsMapper UserMapper RentOrderMapper AssignmentMapper CourseOrderMapper TestMapper AssignmentMapper ReviewMapper InformMapper	服务端业务逻辑层	服务器端数据层

5.2 用户界面层分解

根据需求，系统存在13个用户界面：学生搜索课程界面、学生个人中心界面、学生已购课程界面、学生购买课程界面、学生用户主界面、学生学习课程界面、注册界面、登录界面、未登录主界面、管理课程界面、教师用户主界面、添加课程界面、教师个人中心界面。界面跳转如下图所示。

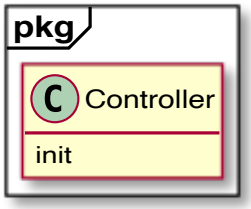


用户界面跳转

5.2.1 职责

- 类图

服务器端和客户端的用户界面设计接口是一致的，只是具体的页面不一样。用户界面类如图所示。



模块	职责
CourseController	负责课程界面相关
CourseOrderController	负责订单界面相关
CourseWareController	负责课程课件相关
FileController	负责文件相关服务
UserController	负责用户界面相关
ReviewController	负责评论界面相关
CommentsController	负责讨论区界面相关
AssignmentController	负责发布测试界面相关
TestController	负责完成测试界面相关
RechargeController	负责账户界面相关
RentOrderController	负责租用界面相关

5.2.2 接口规范

Comments模块的接口规范

接口名字	语法	前置条件	后置条件
CommentsController.getComments	getComments(Integer id)	输入合法	从CommentsService获得预期的帖子
CommentsController.getAllComments	getAllComments(Integer courseId, Integer selectRule)	selectRule为0或1	从CommentsService获得全部的帖子并且按需求排序完成
CommentsController.createComments	createComments(Comments VO)	输入合法	将帖子信息传递给后端CommentsService
CommentsController.deleteComments	delete(Integer commentsId)	输入合法	将被删除的帖子信息传递给后端CommentsService
ReviewController.createReview	createReview(Integer commentsId, ReviewVO reviewVO)	输入合法	将回复信息传递给后端ReviewService
ReviewController.getReview	getReview(Integer commentsId)	输入合法	从ReviewService获得预期的帖子
InformController.getAllInfo	getAllInfo(Integer uid)	无	从InfromMapper中获得所有的消息通知
InformController.deleteByPrimaryKey	deleteByPrimaryKey(Integer id)	无	根据主键删除InformMapper中的信息

Comments模块的服务接口

服务名	服务
CommentsService.getAllComments(Integer courseId, Integer selectRule)	从CommentsMapper获得全部的帖子
CommentsService.getComments(Integer CommentsID)	从CommentsMapper获得预期的帖子
CommentsService.deleteComments(Integer commentsId)	将帖子信息传递给CommentsMapper
CommentsService.createComments(CommentsVO commentsVO)	将帖子信息传递给CommentsMapper
ReviewService.getReview	从ReviewMapper获得预期的回复
ReviewService.createReview	将回复信息传递给ReviewMapper
InformService.informMapper.selectAllInforms	从InformMapper中获得用户全部的消息通知
InformService.informMapper.deleteByPrimaryKey	在InformMapper中删除消息通知

signment模块的接口规范

接口名字	语法	前置条件	后置条件
AssignmentController.getAssignments	getAssignments(Integer courseId)	输入合法	从AssignmentMapper获得预期的问题
AssignmentController.addAssignments	add(AssignmentVO assignment)	输入合法	将问题信息传递给后端AssignmentService
AssignmentController.editAssignments	edit(AssignmentVO assignment)	输入合法	将问题信息传递给后端AssignmentService
AssignmentController.deleteAssignments	delete(Integer courseId)	输入合法	将问题信息传递给后端AssignmentService
TestController.getTestById	getTestById(Integer id)	无	从TestMapper中获得需求测试
TestController.addTest	addTest(TestVO vo)	无	向TestMapper中添加需求测试
TestController.delTest	delTest(Integer id)	无	在TestMapper中删除需求测试
TestController.getStutest	getStutest(Integer courseId)	无	从TestMapper中获得学生需求测试
TestController.getStuAnswer	getStuAnswer(Integer studentId,Integer testId)	无	从TestMapper中获得学生需求答案

Assignment模块的服务接口

服务名	服务
AssignmetService.getAssignments	从AssignmentMapper获得预期的问题
AssignmetService.add	将问题信息传递给后端AssignmentService
AssignmetService.edit	将问题信息传递给后端AssignmentService
AssignmetService.delete	将问题信息传递给后端AssignmentService
TestService.addTest	创建测试，并添加到TestMapper
TestService.delTest	根据testID删除测试，更新TestMapper
TestService.getStutest	根据courseId获取学生查看的测试，包括未开始，在进行，已经结束三种状态
TestService.getStutest	根据courseId获取学生查看的测试，包括未开始，在进行，已经结束三种状态
TestService.submitAnswer	提交需求测试的所有答案

RentOrder模块的接口规范

接口名字	语法	前置条件	后置条件
RentOrderController.getOrder	getOrder(Integer userid, Integer courseid)	输入合法	从RentOrderMapper获得预期的问题
RentOrderController.createOrder	createOrder(Integer userid,Integer courseid, Integer days)	输入合法	将租用订单信息传递给后端RentOrderService
RentOrderController.payOrder	payOrder(Integer orderid)	输入合法	将租用订单信息传递给后端RentOrderService

RentOrder模块的服务接口

服务名	服务
RentOrderService.getRentOrder	从RentOrderMapper获得预期的问题
RentOrderService.createRentOrder	将租用订单信息传递给后端RentOrderMapper
RentOrderService.payRentOrder	将租用订单信息传递给后端RentOrderMapper

User模块的接口规范

接口名字	语法	前置条件	后置条件
UserController.isVIP	isVIP(Integer uid)	输入合法	从UserMapper中查询用户是否是VIP
UserController.openVIP	openVIP(VIPFormVO vipform)	输入合法	将开通VIP订单信息传递给后端UserService
UserController.getInforms	getInforms(Integer uid)	输入合法	从InformMapper获得预期的通知
UserController.deleteByPrimaryKey	deleteByPrimaryKey(Integer id)	输入合法	将被删除通知的id传递给后端UserService

User模块的服务接口

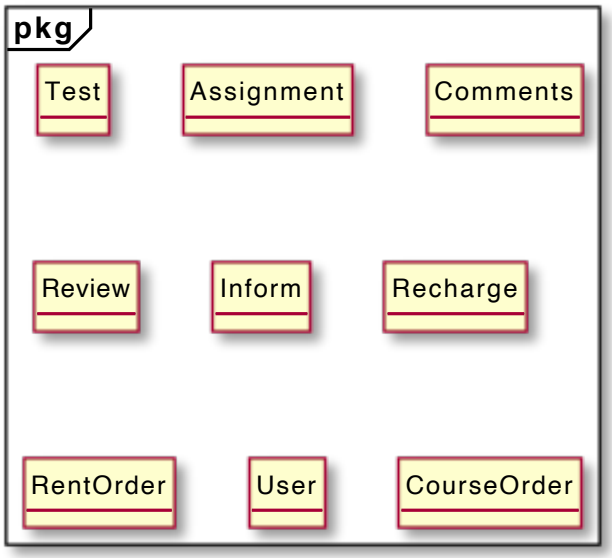
服务名	服务
userService.isVIP	从UserMapper中查询用户是否是VIP
userService.openVIP	将开通VIP订单信息传递给后端UserMapper
InformService.getAllInfo	从InformMapper获得预期的通知
InformService.deleteByPrimaryKey	将被删除通知的id传递给后端InformMapper

5.2.3 用户界面模块设计原理

用户界面利用用Vue.js框架实现。

5.3 业务逻辑层分解

业务逻辑层分解如图所示:

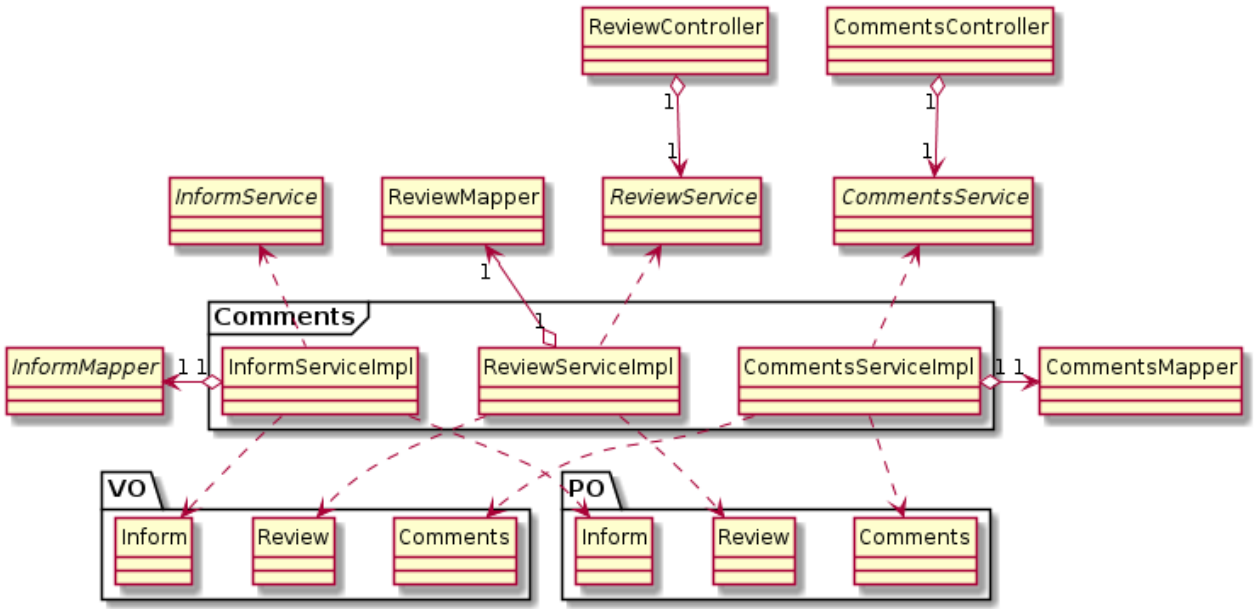


5.3.1 职责

模块	职责
Userbl	负责用户注册界面的业务 负责用户登陆界面的业务 负责用户查看个人信息的业务 负责用户查看个人信息的业务 负责管理员用户查看网站统计信息的业务 负责管理员用户发布课程优惠券的业务 负责处理用户开通VIP请求的业务 负责用户查看自己VIP信息的业务 负责回复通知提醒的业务 负责提醒用户回复通知的服务
Assignmentbl	负责教师创建测试的服务 负责用户查看测试列表的服务 负责用户进入具体测试内容的服务 负责提供用户得分情况和正确解析的服务 负责用户查看测试信息的业务 负责用户提交测试的业务 负责教师创建题目的服务 负责用户查看题目的服务
Commentsbl	负责用户查看讨论区的服务 负责用户在讨论区发帖的业务 负责根据用户需求按不同顺序展示帖子列表的业务 负责用户进入详细帖子内容的服务 负责用户查看帖子的评论区的服务 负责用户发布评论的服务
RentOrderbl	负责用户租用课程的业务
CourseOrder	负责课程订单的相关业务（非第三阶段需求，简略描述）

5.3.2 接口规范

Commentsbl模块的关键类图



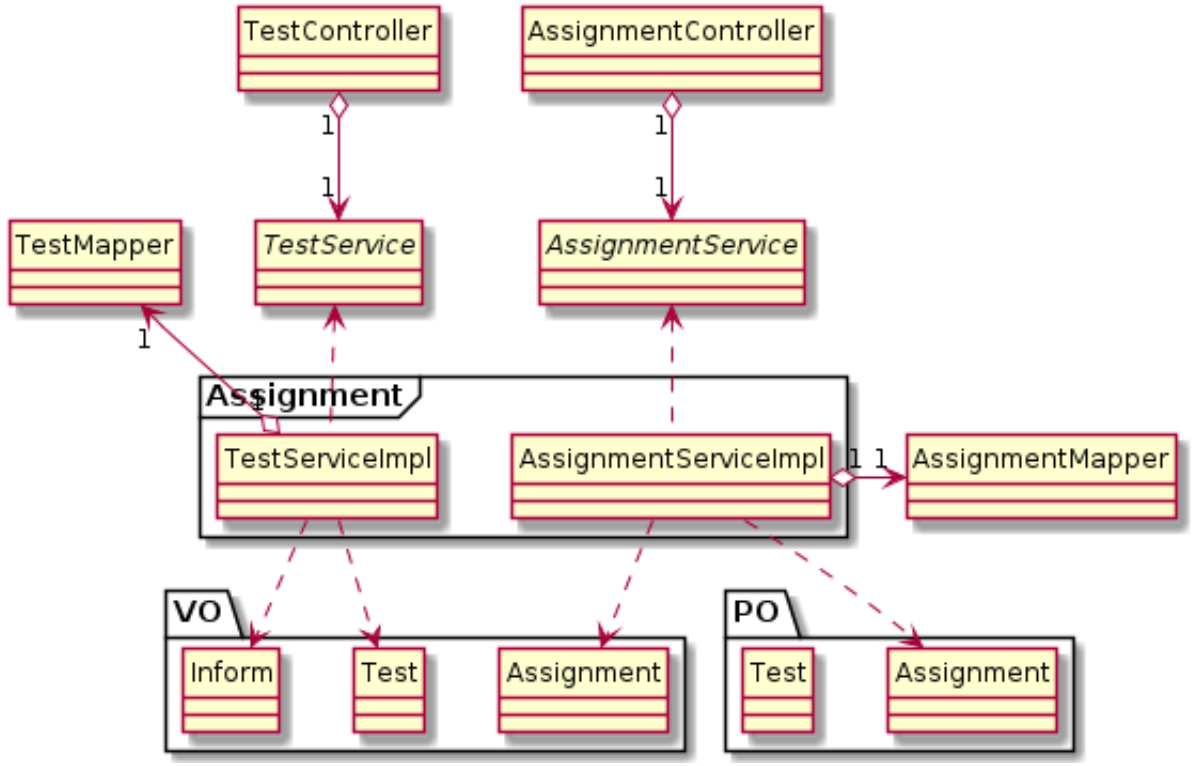
Commentsbl模块的接口规范

接口名字	语法	前置条件	后置条件
CommentsService.getAllComments	getAllComments(Integer courseId, Integer selectRule)	无	从CommentsMapper获得预期的帖子
CommentsService.getComments	getComments(Integer CommentsID)	无	从CommentsMapper获得全部的帖子
CommentsService.createComments	createComments(CommentsVO commentsVO)	无	将帖子信息传递给CommentsMapper
CommentsService.deleteComments	deleteComments(Integer commentsId)	无	将帖子信息传递给CommentsMapper
ReviewService.getAllReview	getAllReview(Integer cid)	无	从ReviewMappper中获得和某帖子相关的全部评论
ReviewService.createReview	createReview(Integer commentsId, ReviewVO reviewVO)	无	将评论信息传递给ReviewMapper

Commentsbl模块需要的服务

服务名	服务
CommentsMapper.selectCommentsByPrimaryKey	根据主键返回需求的帖子
CommentsMapper.selectCommentsByCourseId	根据课程好返回和课程相关的所有帖子
CommentsMapper.insert	插入帖子
CommentsMapper.deleteByPrimaryKey	根据主键删除帖子
ReviewMapper.selectReviewsByCommentsId	根据帖子号返回帖子的所有评论
ReviewMapper.selectCommentsByPrimaryKey	根据主键返回全部帖子
ReviewMapper.commentsService.updateComments	根据主键修改评论
ReviewMapper.informService.insert	向InformMapper中插入Inform
ReviewMapper.reviewMapper.insert	向ReviewMapper中插入review

Assignmentbl模块的关键类图



Assignmentbl模块的接口规范

接口名字	语法	前置条件	后置条件
AssignmentService.getAssignments	getAssignments(Integer courseId)	无	从AssignmentsMapper中获得对应课程的全部的题目
AssignmentService.getAssignmentByTestId	getAssignmentByTestId（Integer id)	无	从AssignmentsMapper中获得对应具体某次的题目
AssignmentService.getTestById	getTestById(Integer id)	无	从TestMapper中获得需求测试
TestService.addTest	addTest(TestVO vo)	无	向TestMapper中添加需求测试
TestService.delTest	delTest(Integer id)	无	在TestMapper中删除需求测试
TestService.getStutest	getStutest(Integer courseId)	无	从TestMapper中获得学生需求测试
TestService.getStuAnswer	getStuAnswer(Integer studentId,Integer testId)	无	从TestMapper中获得学生需求答案

Assignmentbl模块需要的服务

服务名	服务
AssignmentMapper.insert	创建题目,并添加到AssignmentsMapper
AssignmentMapper.update	编辑题目内容并更新题目
AssignmentMapper.delete	删除题目并更新AssignmentsMapper
TestMapper.addTest	创建测试，并添加到TestMapper
TestMapper.delTest	根据testID删除测试，更新TestMapper
TestMapper.getStutest	根据courseId获取学生查看的测试，包括未开始，在进行，已经结束三种状态
TestMapper.getStuAnswer	根据studentId、 testId获取学生需求测试的答案
TestMapper.submitAnswer	提交需求测试的所有答案
AssignmentMapper.selectByCourseId	根据课程id获取相关测试
AssignmentMapper.selectByTestId	根据测试id获取相关测试
AssignmentMapper.deleteAssignmentTestByAId	根据题目id删除题目数据

RentOrderbl模块的关键类图

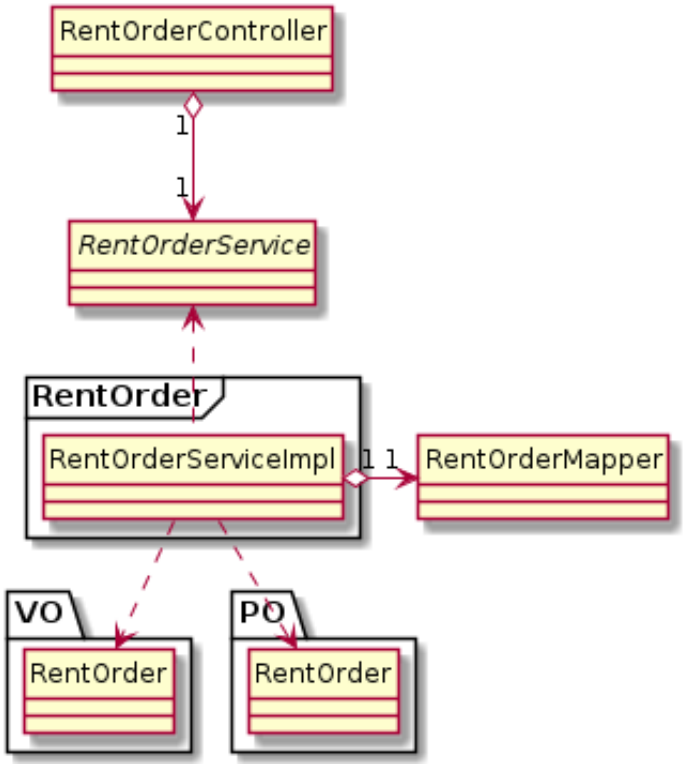
RentOrderbl模块的接口规范

接口名字	语法	前置条件	后置条件
RentOrderService.getRentOrder	getRentOrder(Integer userid, Integer courseid)	无	从RentOrderMapper中获取订单
RentOrderService.createRentOrder	createRentOrder(Integer userId, Integer courseId, Integer days)	无	在RentOrderMapper中创建订单
RentOrderService.payRentOrder	payRentOrder(Integer orderId)	无	建立租用订单
RentOrderService.updateRentOrder	updateRentOrder(Integer orderId, Integer orderStatus)	无	更新RentOrderMapper中的数据

RentOrderbl模块需要的服务

服务名	服务
RentOrderMapper.selectByUidCid	根据uid和cid查询订单
courseOrderService.getCourseOrders	根据uid查询课程订单
orderMapper.insert	向ordermapper中插入订单
orderMapper.selectByPrimaryKey	根据主键在OrderMapper中查找订单
userService.getUser	根据uid在UserMapper中获得用户信息
userService.decreaseBalance	减少用户余额
orderMapper.updateByPrimaryKey	根据主键更新OrderMapper中的信息
RentOrderFactory.getOrder	从订单工厂获得订单对象

Userbl模块的关键类图



Userbl模块的接口规范

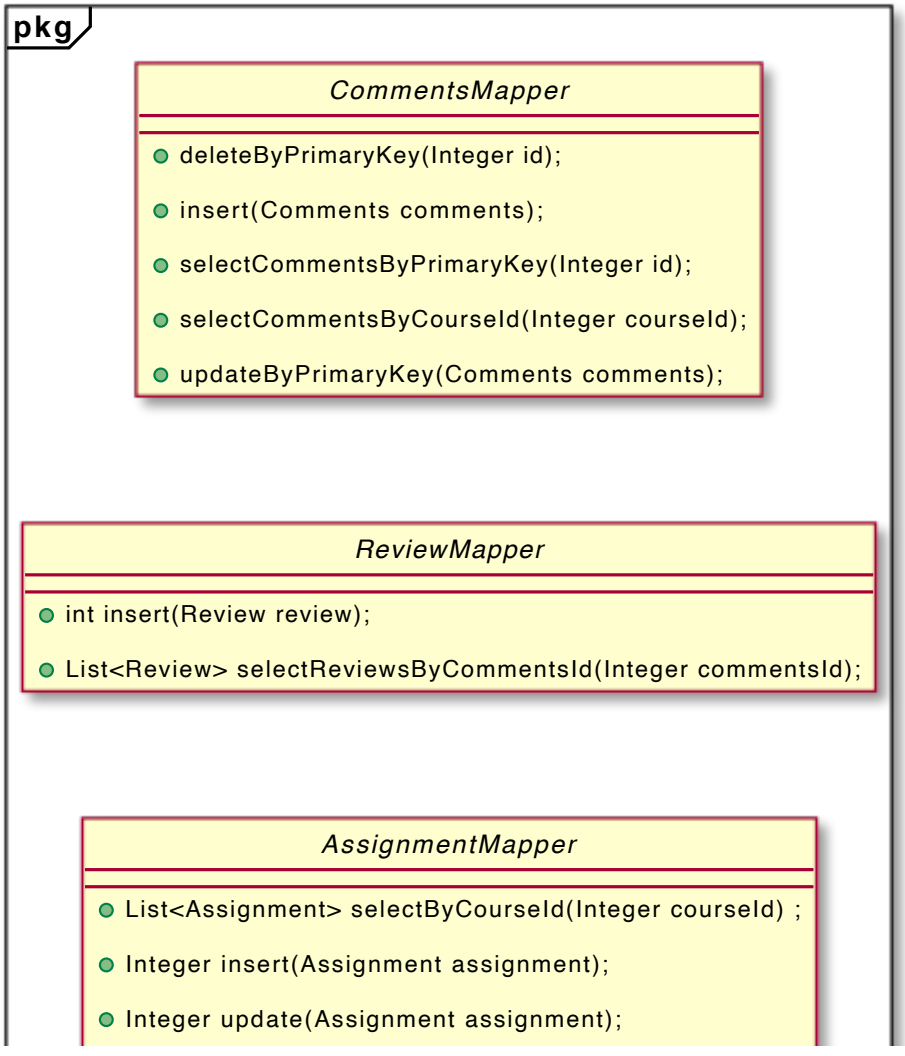
接口名字	语法	前置条件	后置条件
UserService.isVIP	isVIP(Integer uid)	无	返回用户是否为VIP
UserService.openVIP	openVIP(Integer userId, Integer vipPrice, Integer duration)	无	开通了会员，并计入UserMapper
InformService.getAllInfo	getAllInfo(Integer uid)	无	从InfromMapper中获得所有的消息通知
InformService.deleteByPrimaryKey	deleteByPrimaryKey(Integer id)	无	根据主键删除InformMapper中的信息

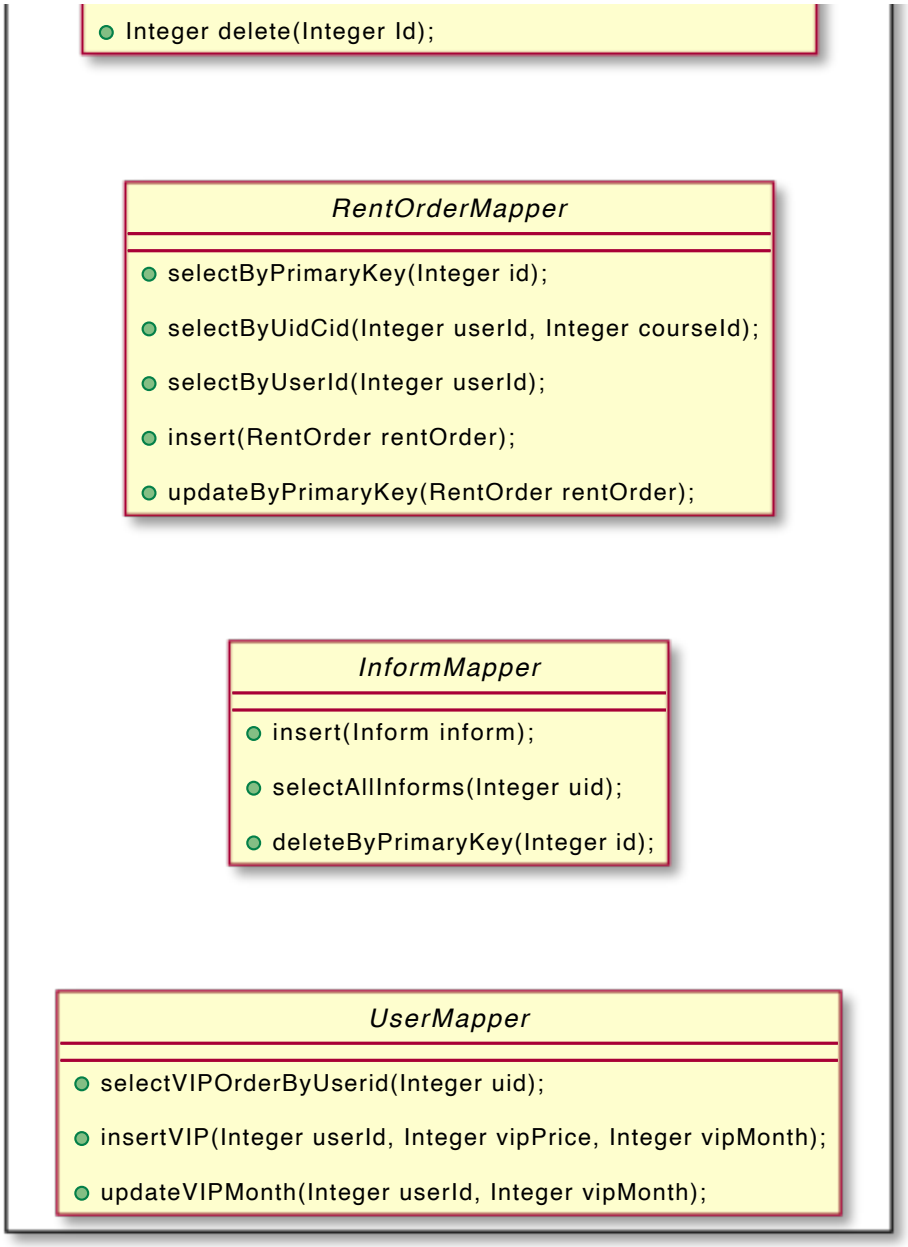
Userbl模块需要的服务

服务	服务名
userMapper.selectVIPOrderByUserid	根据uid查询vip订单
userMapper.updateVIPMonth	更新用户的VIP时长
userMapper.insertVIP	插入VIP信息
userMapper.decreaseBalance	扣除用户余额
informMapper.selectAllInforms	从InformMapper中获得用户全部的消息通知
informMapper.deleteByPrimaryKey	在InformMapper中删除消息通知

5.4 数据层分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。Comments, Review, Assignment, RentOrder, Inform等业务逻辑需要的服务分别由CommentsMapper, ReviewMapper, AssignmentMapper, RentOrderMapper, InformMapper接口提供。数据层模块的描述具体如下图所示。





5.4.1 职责

数据层模块(仅购买相关)的职责如下表所示。

模块	职责
CommentsMapper	持久化数据库的接口。提供帖子信息的集体载入、集体保存、增、删、改、查服务。
ReviewMapper	持久化数据库的接口。提供回复信息的集体载入、集体保存、增、删、改、查服务。
AssignmentMapper	持久化数据库的接口。提供练习题目的集体载入、集体保存、增、删、改、查服务。
RentOrderMapper	持久化数据库的接口。提供租用订单的集体载入、集体保存、增、删、改、查服务。
InformMapper	持久化数据库的接口。提供回帖通知的集体载入、集体保存、增、删、改、查服务。

5.4.2 接口规范

数据层模块的借口规范如下表所示。

接口名称	语法	前置条件	后置条件
CommentsMapper.deleteByPrimaryKey	int deleteByPrimaryKey(Integer id);	数据库中存在相同ID的PO	删除一个PO
CommentsMapper.insert	int insert(Comments comments);	数据库中不存在相同ID的PO	在数据库中增加一个PO
CommentsMapper.selectCommentsByPrimaryKey	Comments selectCommentsByPrimaryKey(Integer id);	无	按ID查找返回相应的PO结果
CommentsMapper.selectCommentsByCourseId	List< Comments> selectCommentsByCourseId(Integer courseId);	无	按ID查找返回相应的PO结果
CommentsMapper.updateByPrimaryKey	int updateByPrimaryKey(Comments comments);	数据库中存在相同ID的PO	更新一个PO
ReviewMapper.insert	int insert(Review review);	数据库中不存在相同ID的PO	在数据库中增加一个PO
ReviewMapper.selectReviewsByCommentsId	List< Review> selectReviewsByCommentsId(Integer commentsId);	无	按ID查找返回相应的PO结果
AssignmentMapper.selectByCourseId	List< Assignment> selectByCourseId(Integer courseId) ;	无	按ID查找返回相应的PO结果
AssignmentMapper.insert	Integer insert(Assignment assignment);	数据库中不存在相同ID的PO	在数据库中增加一个PO
AssignmentMapper.update	Integer update(Assignment assignment);	数据库中存在相同ID的PO	更新一个PO
AssignmentMapper.delete	Integer delete(Integer Id);	数据库中存在相同ID的PO	删除一个PO
RentOrderMapper.selectByPrimaryKey	RentOrder selectByPrimaryKey(Integer id);	无	按ID查找返回相应的PO结果
RentOrderMapper.selectByUidCid	RentOrder selectByUidCid(Integer userId, Integer courseId);	无	按ID查找返回相应的PO结果
RentOrderMapper.selectByUserId	List< RentOrder> selectByUserId(Integer userId);	无	按ID查找返回相应的PO结果
RentOrderMapper.insert	int insert(RentOrder rentOrder);	数据库中不存在相同ID的PO	在数据库中增加一个PO
RentOrderMapper.updateByPrimaryKey	int updateByPrimaryKey(RentOrder rentOrder);	数据库中存在相同ID的PO	更新一个PO
InformMapper.insert	int insert(Inform inform);	数据库中不存在相同ID的PO	在数据库中增加一个PO
InformMapper.selectAllInforms	List< Inform> selectAllInforms(Integer uid);	无	按ID查找返回相应的PO结果
InformMapper.deleteByPrimaryKey	int deleteByPrimaryKey(Integer id);	数据库中存在相同ID的PO	删除一个PO
UserMapper.updateVIPMonth	void updateVIPMonth(Integer userId, Integer vipMonth);	数据库中存在相同ID的PO	更新一个PO
UserMapper.insertVIP	void insertVIP(Integer userId, Integer vipPrice, Integer vipMonth);	数据库中不存在相同ID的PO	在数据库中增加一个PO
UserMapper.selectVIPOrderByUserid	List< VIPorder> selectVIPOrderByUserid(Integer uid);	无	按ID查找返回相应的PO结果

6. 信息视角

- 描述数据持久化对象(PO)：系统的PO类就是对应的相关的实体类，在此只做简单的介绍。

6.1 Assignment_PO

属性	定义	类型
courseId	课程id	Integer
Id	题目id	Integer
title	题目标题	String
type	题目种类	Integer
answer	题目答案	String
AOption	A选项	String
BOption	B选项	String
COption	C选项	String
DOption	D选项	String
analysis	解析	String
Score	分数	Integer

6.2 Comments_PO

属性	定义	类型
id	帖子ID	Integer
Title	帖子标题	String
subtitle	课程id	String
abstraction	帖子内容	String
courseId	帖子所属课程ID	Integer
createTime	创建时间	Date
uid	创建用户id	Integer
uname	创建用户名称	String
updateT	更新时间	Date

6.3 Review_PO

属性	定义	类型
id	回复id	Integer
Content	回复内容	String
commentsId	被回复的帖子ID	Integer
reviewTime	回复时间	Date
uid	回复用户的ID	Integer
uname	回复用户的昵称	String

6.4 Test_PO

属性	定义	类型
Id	测试Id	Integer
courseId	测试所属课程Id	Integer
testname	测试名称	String
beginTime	测试开始时间	Date
endTime	测试结束时间	Date
assignmentIdList	测试包含的题目的ID	List< Integer>
score	测试得分	Integer

6.5 RentOrder_PO

属性	定义	类型
Id	租用订单ID	Integer
userId	用户ID	Integer
courseId	课程ID	Integer
Cost	租用价格	Integer
courseName	课程名称	String
createTime	创建时间	Date
endTime	结束时间	Date
Status	订单状态	Integer
Days	租用时长	Integer

持久化帖子对象Comments_PO的定义如图所示：

```java
package cn.seecoder.courselearning.po.course;

import cn.seecoder.courselearning.vo.course.CommentsVO;
import lombok.NonNull;

import java.util.Date;

public class Comments {

    /**
     * 帖子id
     */
    private Integer id;

    /**
     * 标题
     */
    private String title;

    /**
     * 副标题
     */
    private String subTitle;

    /**
     * 摘要
     */
    private String abstraction;
```

```java

    /**
     * 课程id
     */
    private Integer courseId;

    /**
     * 创建帖子的时间
     */
    private Date createTime;

    /**
     * 发表用户id
     */
    private Integer uid;

    /**
     * 用户名字
     */
    private String uname;

    /**
     * 最新回复时间
     */
    private Date updateT;

    public Date getUpdateT() {
        return updateT;
    }

    public void setUpdateTime(Date updateTime) {
        this.updateT = updateTime;
    }


    public String getAbstraction() {
        return abstraction;
    }

    public void setAbstraction(String abstraction) {
        this.abstraction = abstraction;
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
```

```
78      }
79
80      public String getTitle() {
81          return title;
82      }
83
84      public void setTitle(String title) {
85          this.title = title;
86      }
87
88      public Date getCreateTime() {
89          return createTime;
90      }
91
92      public void setCreateTime(Date createTime) {
93          this.createTime = createTime;
94      }
95
96      public String getUname() {
97          return uname;
98      }
99
100     public void setUname(String uname) {
101         this.uname = uname;
102     }
103
104     public Integer getUid() {
105         return uid;
106     }
107
108     public void setUid(Integer uid) {
109         this.uid = uid;
110     }
111
112     public String getsubTitle() {
113         return subTitle;
114     }
115
116     public void setsubTitle(String subTitle) {
117         this.subTitle = subTitle;
118     }
119
120     public Integer getCourseId() {
121         return courseId;
122     }
123
124     public void setCourseId(Integer courseId) {
125         this.courseId = courseId;
126     }
```

```
127
128     public Comments() {}
129
130     public Comments(Integer id, String title, String subTitle, String
    abstraction, Integer courseId, Date createTime, Integer uid, String
    uname, Date updateT) {
131         this.id = id;
132         this.title = title;
133         this.subTitle = subTitle;
134         this.abstraction = abstraction;
135         this.courseId = courseId;
136         this.createTime = createTime;
137         this.uid = uid;
138         this.uname = uname;
139         this.updateT = updateT;
140     }
141
142     public Comments(@NonNull CommentsVO commentsVO) {
143         this.abstraction = commentsVO.getAbstraction();
144         this.courseId = commentsVO.getCourseId();
145         this.createTime = commentsVO.getCreateTime();
146         this.id = commentsVO.getId();
147         this.subTitle = commentsVO.getSubTitle();
148         this.title = commentsVO.getTitle();
149         this.uid = commentsVO.getUid();
150         this.uname = commentsVO.getUname();
151         this.updateT = commentsVO.getUpdateT();
152     }
153 }
154
```

6.6 数据库表

数据库中包含的表
assignments表
Comments_Info表
coupon表
course表
course_likes表
Course_order表
Course_order_coupon表
course_ware表
Information_Info表
Recharge_order表
Review_Info表
User_coupon表
User_info表
Vip_order表