**二手书交易平台概要设计**

**（版本v1.0.0）**

中软国际信息技术有限公司

2018年1月11日



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 成文信息 | | | | | | | |
| 主题词： | | 概要设计 | | | | | |
| 作 者： | | 林加盛 | | 文档类别： | | | 技术方案、设计文档 |
| 审 核： | |  | |
| 批 准： | |  | | 文档性质： | | | 正式稿 |
| 主 送： | |  | | 存档日期： | | |  |
| 抄 送： | |  | | 发布日期： | | |  |
| 签收信息 | | | | | | | |
| 发送方 | | | 接收方： | 接收方： | | 接收方： | |
| 代表人： | | | 代表人： | 代表人： | | 代表人： | |
| 日期： | | | 日期： | 日期： | | 日期： | |
| 变更信息 | | | | | | | |
| 版本 | 原因 | | | | 作者 | 日期 | |

目录

第一章 引言 1

1.1 编写目的 1

1.2 背景 2

1.3 定义 3

1.4 参考资料 3

第二章 总体设计 4

2.1 需求规定 4

2.2 运行环境 6

2.2.1 运行环境网络结构说明 7

2.2.2 运行环境说明细则 8

2.2.3 支持软件 8

2.3 基本设计概念和处理流程 8

2.3.1 整体流程 8

2.3.2 设计概念 9

2.4 结构设计 19

2.4.1 功能模块划分 19

2.5 人工处理过程 47

第三章 接口设计 48

3.1 数据请求与返回接口 48

第四章 系统出错处理 50

4.1 出错信息 50

4.2 补救措施 50

4.3 系统维护设计 51

# 引言

## 编写目的

通过对二手书交易市场工作流程的深入研究，二手书交易平台系统项目组发现制约二手书交易效率的各种因素中最主要是找书难，对书籍不够了解的情况下 凭借书名购买书籍的现象。二手书交易平台运用计算机软件系统来实现二手书交易市场的信息化管理，采用面向对象的开发方法以及Python语言，依靠Pycharm作为开发工具,使用MySQL数据库存储数据。实现了征书墙，竞拍管理，购买管理等功能，解决找书难，书籍没有系统分类的现象。

此概要设计说明书对《二手书交易平台》的基本设计概念和处理流程、功能模块划分、程序总体结构、接口设计、出错处理做了全方位的概括性的说明，为该软件的详细设计奠定了基础，体现了二手书交易平台用户需求和应用系统之间的关系，最终实现目的：

1、对系统概要设计的阶段任务成果形成文档，以便阶段验收、评审，最终的文档验收。

2、对需求阶段的文档再次确认过程，对前一阶段需求没有做充分或错误的提出修改。

3、明确整个系统的功能框架和数据库结构，为下一阶段的详细设计、编码、和测试提供参考依据。

4、明确编码规范和命名规范，统一程序界面。

本设计书预期读者：系统详细设计人员；系统开发人员；软件维护人员；技术管理人员；执行软件质量保证计划的专门人员；参与本项目开发进程各阶段验证、确认以及负责为最后项目验收、鉴定提供相应报告的有关人员。

## 背景

从在二十世纪末期到现在，随着互联网的高速发展，各种各样的购物网站也都相继的出世，人们开始享受足不出户也能购买到自己想要的商品。这让我们意识到网购时代的来临，二十一世纪是网上交易的时代。而亚马逊网站——作为最专业的购书网站，成立于1995年，2010年该网站的销售额为130亿美元，是全球电子商务成功的代表。当然，它的成功也代表着人们对书籍的需求正在不断的扩大，对知识的渴求不断的加深，说明全球对书籍的需求量的巨大。所以，大量的需求也给二手书交易平台提供了保障。

虽然说，国内也有很多的淘书网站，但是，真真正正的做二手书的却非常的少，一些小的网站如孔夫子网站，跳蚤市场则相对来说规模小，缺乏完善的机制。而一些较大的网站如有路网以及秦有才旧书网却快被人遗忘。总的来说，大型的网站不足以满足二手书的薄利，而小型的旧书网站却存在许多管理方面的问题，也许它们更愿意被人收购。所以，二手书市场看起来很热闹，其实整个体系还不完整

待开发软件系统名称：二手书交易平台

项目任务提出者：公共事业产品部

项目开发者：林加盛

项目用户：普通消费者

项目运营方式：单一客户独立部署运营

## 定义

MIS: 管理信息系统

系统: 管理信息系统

## 参考资料

《二手书交易平台需求规格说明书》

# 总体设计

## 需求规定

(1) 对功能的规定

在充分理解《二手书交易平台需求规格说明书》的基础上，根据用户发布的书籍到书城，征书墙，竞拍的需求，设计此方案，来最大限度满足人民日益增长的购书的需要。

二手书交易平台包括三个部分：普通用户（购书者，发布者），系统管理员。系统的结构框图如下：

二手书交易平台

系统管理员

普通用户(购书者)

普通用户(发布者)

用户管理

书籍管理

订单管理

历史数据

支付管理

评论管理

发布求购

浏览书城

列表

详情

更新

参与竞拍

发布出售二手书信息

浏览在征书墙上求购信息

发布书籍竞拍消息

(2) 对性能的规定

精度：本软件数据传输、处理时数据要求保留2位小数点，超过两位以上小数采取四舍五入方式处理。

时间特性的要求：对于任何功能，数据的处理、响应、展现超时不应该超过1秒钟。

灵活性：系统在设计之初要考虑到系统以后的扩展，除了采用层次结构，保证功能实现与通讯接口最大限度的独立，在与通信平台的接口设计上尽量标准化，以实现和新系统的无缝连接。当客户端有更灵活的需求时，应该能够在不该服务端的情况下方便的扩展Android、IOS等原生APP客户端。

(3) 输入/输出要求

输入：普通文本数据使用操作人员键盘输入，涉及到的数字数据精度要求保留2位小数，用户头像及书籍图片部分为图片文件导入输入。

输出：系统所有交互结果均由屏幕输出。

(4) 数据管理能力要求

根据不同的需要，可以对数据库采取实时热备份或定时备份的策略。

(5) 故障处理要求

客户端硬件故障：一般性的计算机硬件故障,不能启动、硬件损坏，最好有备用机器。

通讯故障：客户端与服务器无法通讯，导致局部业务暂停，应检查通讯线路及设置，保证网络通讯畅通。

服务器硬件故障：WEB服务器站无法启动，导致办公业务暂停，应有备件或备机替代。

数据服务器硬件故障：数据无法访问，业务暂停，应有备件或备机替代。

客户端应用程序故障：部分业务处理不正常，应检查设置或浏览器等配套设置。

后台服务器故障：部分业务处理不正常，应检查设置或更换配套程序。

(6) 其它专门要求

系统的管理系统与维护系统设置了用户管理权限，采用帐号权限区分管理，系统用户密码使用MD5方式加密，对管理员采取有效的管理，最大限度降低内部管理、操作失误带来的不必要的损失。

## 运行环境

(1) 设备

数据库服务器:

Intel core i3

内存4G以上  
硬盘500GB

100M 网卡

应用服务器:

Intel core i3

内存4G以上  
硬盘500GB

100M 网卡

交换机：

100M交换机

无线路由器：

802.11BGN无线路由

客户端PC：

奔腾双核

内存2G以上

硬盘500G

100M网卡

移动客户端：

Android4.4+ PAD

IPAD2+

(2) 支持软件

服务器操作平台：Linux

应用服务器：Nginx

客户端：Chrome 56及以上

网络环境：Internet

支持协议：TCP/IP、SMTP、POP3、HTTP、SSL

数据库： MySQL5

支撑环境：Python3.5.6及以上

开发工具：Pycharm

设计工具: Visual Paradigm

(3) 接口

提供明确的业务规范和业务流程，给第三方系统提供开放的接口，以便提供灵活的集成机会，最终将该系统建设成为一个二手书综合业务平台，本软件所有的业务过程都有严格数据接口，具体数据传输采用HTTP协议，数据接口采用标准的Json格式集成异构系统。

(4) 控制

本系统采用B/S体系架构，服务器采用Nginx，运行只需在服务器端起动Nginx服务，客户端运行主流浏览器，访问服务器端地址和端口。即可运行。所有的控制利用PYTHON的多线程技术，实现对共享资源的一致性维护。

### 运行环境网络结构说明



### 运行环境说明细则

使用表格对运行环境的细节进行说明。

示例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 设备名称 | 设备型号 | 数量 | 安放位置 | 支持软件 | 说明 |
| 1 | 数据库服务器 | PC | 1台 | 外网 | MySQL5 | 数据存储 |
| 2 | 应用服务器 | PC | 1台 | 外网 | Nginx | 提供应用访问服务 |
| 3 | 客户端 | PC | 1台 | 内网 | Cmder，Chrome | 调试 |
| 4 | 移动客户端 | IPad/Android Pad | 按需 | 内网 | 浏览器 | 交易服务 |

### 支持软件

主流的浏览器进行兼容，对于低版本IE或者内核是IE的浏览器不友好。

## 基本设计概念和处理流程

### 整体流程

1. 用户在登录界面，可以输入手机和短信验证码登录，也可以输入登录名，密码后，提交。

2. 用户有效性检查：用户密码有效性检查，用户状态检查。

3. 根据用户端提供的用户名（手机号）、密码（短信验证码），分配一个加密后的Token，检查用户Token的正确性。

4. 若用户登录不合法，则返回用户登录不成功界面。

5. 若用户登录合法，进入用户登录后界面生成模块 。

6. 进入用户登录后界面视图，组织并生成用户登录后页面。

7. 根据客户的身份分别进入不同处理流程。

8. 用户点击导航栏的交易分类条目，则再进入详细的交易分类界面。

9. 客户点击通知信息条目，则在弹出对话框显示通知信息内容 。

10. 用户点击指向业务处理的交易分类条目，则立即进入相应的业务处理流程。

### 设计概念

#### 手机短信验证码模块

##### 设计模型

模型描述：

1. 输入数据：手机号
2. 内部处理逻辑：
   * 1. 用户请求获取手机短信验证码—>系统给用户发送短信验证码—>用户填写短信验证码—>匹配短信验证码是正确—>跳转界面
3. 输出数据： 获取到的推送消息字符串

##### 处理流程

本模块不对数据进行任何处理，对短信验证码和用户填写验证码进行比对判断。

##### 数据分析

模块需要处理的数据：短信验证码，本模块中传递的消息是一个标准的unicode字符串。

#### 图像实时预览模块

##### 设计模型

模型描述：

1. 输入数据：用户头像或书籍图片
2. 内部处理逻辑：
3. 用户选择图片—>上传图片到指定文件夹—>图片上传成功后向用户推送图片上传成功消息—>自动在图片区域预览上传成功的图片
4. 输出数据： 实时预览图片

##### 处理流程

需要将用户头像图片和书籍图片上传并正确拷贝到规定的文件夹中。

##### 数据分析

模块需要处理的数据：用户头像、书籍图片：图像文件（jpg、bmp、png）。

#### 用户验证模块

##### 设计模型

模型描述：

1. 输入数据： 用户名(手机号)

密码

1. 内部处理逻辑：
2. 用户输入用户名(手机号)密码—>匹配密码是否正确—>跳转界面
3. 输出数据： 登录验证错误提示信息

不同角色用户主界面

##### 处理流程

用户密码不能使用明文存储，因此在验证用户密码之前，需要先获取密码的MD5码，用于和数据库中的原始数据进行匹配。

##### 数据分析

模块需要处理的数据：用户名是一个长度不超过20个字符的字符串，手机号是一个长度不超过11位数字。密码是一个长度不超过18个字符的字符串，

#### 用户管理模块

##### 设计模型

模型描述：

1. 输入数据： 用户详细信息（用户名、用户密码、用户角色信息、用户头像，手机号，收货地址）。

用户ID。

列表页码数

1. 内部处理逻辑：
2. 注册新的用户，输入手机号，和短信验证码—>判定的用户短信验证码是否匹配—>将新的用户信息保存到数据库
3. 管理员提供用户ID—>从数据库中锁定该ID对应的用户—>给出输出用户成功提示。
4. 管理员提供用户ID—>从数据库中查询该ID对应的用户信息—>显示在管理界面中—>管理员修改用户状态—>将更新后的用户信息保存到数据库中
5. 管理员提供要显示的用户列表页码数—>从数据库中查询对应页码的数据—>生成JSON数据—>解析JSON并展现在界面上
6. 输出数据： 用户列表

用户详细信息

##### 处理流程

用户密码口令不能使用明文存储，因此在新增或更新用户时需要保存用户密码的MD5值。

##### 数据分析

模块需要处理的数据：用户列表及用户详细信息功能需要返回对应的JSON文档数据。

#### 书籍管理模块

##### 设计模型

模型描述：

1. 输入数据： 书籍详细信息（包含书籍编号、书籍名称、出版社、书籍版本、书籍描述、书籍价格、系统显示日期、一口价、书籍图像路径，书籍数量，成交价格）。

书籍ID。

列表页码数

1. 内部处理逻辑：
2. 用户要发布求购书籍信息—>将新的发布求购书籍信息保存到数据库—>刷新征书墙界面，显示发布求购书籍信息列表。
3. 用户要发布出售书籍信息—>将新的发布出售书籍信息保存到数据库—>刷新书城界面，显示发布出售书籍信息列表。
4. 用户要发布竞拍书籍信息—>将新的发布竞拍书籍信息保存到数据库—>刷新征书墙界面，显示发布求购书籍信息列表。
5. 用户要浏览书城，竞拍，征书墙界面——>从数据库中查询对应页码的数据—>生成JSON数据—>解析JSON并展现在界面上
6. 输出数据： 书籍列表

书籍详细信息

##### 处理流程

由于需要在页面中显示书籍详细信息中可能本身包含控制字符如双引号等，因此详细信息字符串需要预处理，将双引号等字符替换成自定义的占位符。发布出售书籍信息的时候，系统管理员可以介入发布出售书籍信息。

##### 数据分析

模块需要处理的数据：书籍列表及书籍详细信息功能需要返回对应的JSON文档数据。

#### 评论模块

##### 设计模型

模型描述：

1. 输入数据： 对书籍评论字符串。
2. 内部处理逻辑：
3. 用户输入对发布者的书籍进行评价—>发布者获取评价信息，显示评价信息—>发布者回复评价信息
4. 输出数据： 评价信息

##### 处理流程

对于违反平台条例的信息进行屏蔽后再发布

##### 数据分析

模块需要处理的数据：实时评价字符串：标准的unicode字符串。

#### 支付模块

##### 设计模型

模型描述：

1. 输入数据： 待支付订单号。
2. 内部处理逻辑：
3. 系统提供待支付订单号—>从数据库中查询该订单详细信息—>将该订单详细信息格式化为JSON数据—>将通知买单信息推送到用户界面
4. 输出数据：支付并返回支付后的订单详情

##### 处理流程

数据无特殊处理。

##### 数据分析

模块需要处理的数据：订单数据，包括订单的详细列表及购买时间、发布者信息、用户信息。

#### 订单模块

##### 设计模型

模型描述：

1. 输入据： 购物车中的书籍列表（书籍ID，对应的书籍数量）。
2. 内部处理逻辑：
3. 用户选择购物车中准备购买的书籍和书籍数量—>点击结算—>系统生成订单详情，将该订单保存到数据库中。
4. 用户点击查看订单详情—>选择支付方式—>点击确认支付—>支付成功，改变订单的状态—>已付款
5. 用户点击查看订单详情（支付未成功，订单超时）—>点击取消该订单，改变订单的状态—>已取消
6. 订单生成并由用户支付后—>发布者没有在规定的时间内接单（接单后没有在规定的时间内发货）—>改变订单的状态—>已退单
7. 发布者接收到系统发送的订单详情—>点击接单—>改变订单的状态—>已接单
8. 输出数据：订单的状态

##### 处理流程

数据无特殊处理。

##### 数据分析

模块需要处理的数据：订单编号，订单状态，订单详情。

#### 历史数据查询模块

##### 设计模型

模型描述：

1. 输入数据： 订单交易时间
2. 内部处理逻辑：
3. 系统管理员提供开始时间和结束时间作为查询条件—>查询订单交易时间在提供的开始时间和结束时间之间的订单数据—>将订单数据格式化为JSON文档—>显示查询结果
4. 输出数据：历史订单列表

##### 处理流程

数据无特殊处理。

##### 数据分析

模块需要处理的数据：订单详细信息，包括订单编号，订单交易时间，用户编号，发布者编号，书籍名，数量，总金额。

。

## 结构设计

### 用户管理模块

简介

此模块为用户管理模块，包括登录、注册、修改个人信息、修改密码、用户定位、查看订单、查看收藏、查看积分、积分兑换、我的消息。



图2-2-2-1用户管理活动图

2、功能列表

|  |  |
| --- | --- |
| 1 | 用户注册 |
| 2 | 用户登录 |
| 3 | 查看用户资料 |
| 4 | 修改用户资料 |
| 5 | 查看收货地址 |
| 6 | 添加收货地址 |
| 7 | 修改收货地址 |
| 8 | 删除收货地址 |
| 9 | 修改登录密码 |

#### 用户注册

**1 功能设计描述**

**（1）类**

**1）UserkeyController.py**

**2）UserkeyService.py**

**3）UserkeyRepository.py**

**4）SuperRepository.py**

**5）SuperRepositoryImpl.py**

**6）UserkeyPo.py**

**（2）类与类之间关系**



图2-2-2-1-1-1用户注册类图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| messageLogin.html | HTML | /resources/static/views/login | 用户验证码登录页面 |
| UserkeyController.py | PYTHON | com.lulu.eating.user.controller | 用户登录凭证controller类 |
| UserkeyService.py | PYTHON | com.lulu.eating.user.service | 用户登录凭证service类 |
| UserkeyRepository .py | PYTHON | com.lulu.eating.user.repository | 用户登录凭证repository类 |
| SuperRepository.py | PYTHON | com.wteam.superboot.core.repository | 超级Repository接口 |
| SuperRepositoryImpl .py | PYTHON | com.wteam.superboot.core.repository | 超级Repository实现类 |
| UserkeyPo.py | PYTHON | com.lulu.eating.user.entity.po | 用户登录凭证Po类 |

**2 功能实现说明**



图2-2-2-1-1-2用户注册时序图

#### 用户登录

**1 功能设计描述**

**（1）类**

**1）UserkeyController.py**

**2）UserkeyService.py**

**3）UserkeyRepository.py**

**4）SuperRepository.py**

**5）SuperRepositoryImpl.py**

**6）UserkeyPo.py**

**（2）类与类之间关系**



图2-2-2-1-2-1用户登录类图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| passwordLogin.html | HTML | /resources/static/views/login | 用户密码登录页面 |
| UserkeyController.py | PYTHON | com.lulu.eating.user.controller | 用户登录凭证controller类 |
| UserkeyService.py | PYTHON | com.lulu.eating.user.service | 用户登录凭证service类 |
| UserkeyRepository .py | PYTHON | com.lulu.eating.user.repository | 用户登录凭证repository类 |
| SuperRepository.py | PYTHON | com.wteam.superboot.core.repository | 超级Repository接口 |
| SuperRepositoryImpl .py | PYTHON | com.wteam.superboot.core.repository | 超级Repository实现类 |
| UserkeyPo.py | PYTHON | com.lulu.eating.user.entity.po | 用户登录凭证Po类 |

**2 功能实现说明**



图2-2-2-1-2-2用户登录时序图

#### 查看用户信息

**1 功能设计描述**

**（1）类**

**1）UserinfoController.py**

**2）UserinfoService.py**

**3）UserinfoRepository.py**

**4）SuperRepository.py**

**5）SuperRepositoryImpl.py**

**6）UserinfoPo.py**

**（2）类与类之间关系**



图2-2-2-1-3-1查看用户信息类图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| userinfo.html | HTML | /resources/static/views/userinfo | 用户信息页面 |
| UserinfoController.py | PYTHON | com.lulu.eating.user.controller | 用户信息controller类 |
| UserinfoService.py | PYTHON | com.lulu.eating.user.service | 用户信息service类 |
| UserinfoRepository .py | PYTHON | com.lulu.eating.user.repository | 用户信息repository类 |
| SuperRepository.py | PYTHON | com.wteam.superboot.core.repository | 超级Repository接口 |
| SuperRepositoryImpl .py | PYTHON | com.wteam.superboot.core.repository | 超级Repository实现类 |
| UserinfoPo.py | PYTHON | com.lulu.eating.user.entity.po | 用户信息Po类 |

**2 功能实现说明**



图2-2-2-1-3-2查看用户信息时序图

#### 修改用户信息

**1 功能设计描述**

**（1）类**

**1）UserinfoController.py**

**2）UserinfoService.py**

**3）UserinfoRepository.py**

**4）SuperRepository.py**

**5）SuperRepositoryImpl.py**

**6）UserinfoPo.py**

**（2）类与类之间关系**



图2-2-2-1-4-1修改用户信息类图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| userinfo.html | HTML | /resources/static/views/userinfo | 用户信息页面 |
| UserinfoController.py | PYTHON | com.lulu.eating.user.controller | 用户信息controller类 |
| UserinfoService.py | PYTHON | com.lulu.eating.user.service | 用户信息service类 |
| UserinfoRepository .py | PYTHON | com.lulu.eating.user.repository | 用户信息repository类 |
| SuperRepository.py | PYTHON | com.wteam.superboot.core.repository | 超级Repository接口 |
| SuperRepositoryImpl .py | PYTHON | com.wteam.superboot.core.repository | 超级Repository实现类 |
| UserinfoPo.py | PYTHON | com.lulu.eating.user.entity.po | 用户信息Po类 |

**2 功能实现说明**



图2-2-2-1-4-2修改用户信息时序图

#### 查看收货地址

**1 功能设计描述**

**（1）类**

**1）UserAddressController.py**

**2）UserAddressService.py**

**3）UserAddressRepository.py**

**4）SuperRepository.py**

**5）SuperRepositoryImpl.py**

**6）UserAddressPo.py**

**（2）类与类之间关系**



图2-2-2-1-5-1查看收货地址类图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| useraddress.html | HTML | /resources/static/views/useraddress | 用户收货地址页面 |
| UserAddressController.py | PYTHON | com.lulu.eating.user.controller | 用户地址controller类 |
| UserAddressService.py | PYTHON | com.lulu.eating.user.service | 用户地址service类 |
| UserAddressRepository .py | PYTHON | com.lulu.eating.user.repository | 用户地址repository类 |
| SuperRepository.py | PYTHON | com.wteam.superboot.core.repository | 超级Repository接口 |
| SuperRepositoryImpl .py | PYTHON | com.wteam.superboot.core.repository | 超级Repository实现类 |
| UserAddressPo.py | PYTHON | com.lulu.eating.user.entity.po | 用户地址Po类 |

**2 功能实现说明**



图2-2-2-1-5-2查看收货地址时序图

#### 添加收货地址

**1 功能设计描述**

**（1）类**

**1）UserAddressController.py**

**2）UserAddressService.py**

**3）UserAddressRepository.py**

**4）SuperRepository.py**

**5）SuperRepositoryImpl.py**

**6）UserAddressPo.py**

**（2）类与类之间关系**



图2-2-2-1-6-1添加收货地址类图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| useraddress.html | HTML | /resources/static/views/useraddress | 用户收货地址页面 |
| UserAddressController.py | PYTHON | com.lulu.eating.user.controller | 用户地址controller类 |
| UserAddressService.py | PYTHON | com.lulu.eating.user.service | 用户地址service类 |
| UserAddressRepository .py | PYTHON | com.lulu.eating.user.repository | 用户地址repository类 |
| SuperRepository.py | PYTHON | com.wteam.superboot.core.repository | 超级Repository接口 |
| SuperRepositoryImpl .py | PYTHON | com.wteam.superboot.core.repository | 超级Repository实现类 |
| UserAddressPo.py | PYTHON | com.lulu.eating.user.entity.po | 用户地址Po类 |

**2 功能实现说明**



图2-2-2-1-6-2添加收货地址时序图

#### 修改收货地址

**1 功能设计描述**

**（1）类**

**1）UserAddressController.py**

**2）UserAddressService.py**

**3）UserAddressRepository.py**

**4）SuperRepository.py**

**5）SuperRepositoryImpl.py**

**6）UserAddressPo.py**

**（2）类与类之间关系**



图2-2-2-1-7-1修改收货地址类图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| useraddress.html | HTML | /resources/static/views/useraddress | 用户收货地址页面 |
| UserAddressController.py | PYTHON | com.lulu.eating.user.controller | 用户地址controller类 |
| UserAddressService.py | PYTHON | com.lulu.eating.user.service | 用户地址service类 |
| UserAddressRepository .py | PYTHON | com.lulu.eating.user.repository | 用户地址repository类 |
| SuperRepository.py | PYTHON | com.wteam.superboot.core.repository | 超级Repository接口 |
| SuperRepositoryImpl .py | PYTHON | com.wteam.superboot.core.repository | 超级Repository实现类 |
| UserAddressPo.py | PYTHON | com.lulu.eating.user.entity.po | 用户地址Po类 |

**2 功能实现说明**



图2-2-2-1-7-2修改收货地址时序图

#### 删除收货地址

**1 功能设计描述**

**（1）类**

**1）UserAddressController.py**

**2）UserAddressService.py**

**3）UserAddressRepository.py**

**4）SuperRepository.py**

**5）SuperRepositoryImpl.py**

**6）UserAddressPo.py**

**（2）类与类之间关系**



图2-2-2-1-8-1删除收货地址类图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| useraddress.html | HTML | /resources/static/views/useraddress | 用户收货地址页面 |
| UserAddressController.py | PYTHON | com.lulu.eating.user.controller | 用户地址controller类 |
| UserAddressService.py | PYTHON | com.lulu.eating.user.service | 用户地址service类 |
| UserAddressRepository .py | PYTHON | com.lulu.eating.user.repository | 用户地址repository类 |
| SuperRepository.py | PYTHON | com.wteam.superboot.core.repository | 超级Repository接口 |
| SuperRepositoryImpl .py | PYTHON | com.wteam.superboot.core.repository | 超级Repository实现类 |
| UserAddressPo.py | PYTHON | com.lulu.eating.user.entity.po | 用户地址Po类 |

**2 功能实现说明**



图2-2-2-1-8-2删除收货地址时序图

#### 修改登录密码

**1 功能设计描述**

**（1）类**

**1）UserkeyController.py**

**2）UserkeyService.py**

**3）UserkeyRepository.py**

**4）SuperRepository.py**

**5）SuperRepositoryImpl.py**

**6）UserkeyPo.py**

**（2）类与类之间关系**



图2-2-2-1-9-1修改登录密码类图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| editUserPassword.html | HTML | /resources/static/views/useraddress | 修改登录密码页面 |
| UserkeyController.py | PYTHON | com.lulu.eating.user.controller | 用户登录凭证controller类 |
| UserkeyService.py | PYTHON | com.lulu.eating.user.service | 用户登录凭证service类 |
| UserkeyRepository .py | PYTHON | com.lulu.eating.user.repository | 用户登录凭证repository类 |
| SuperRepository.py | PYTHON | com.wteam.superboot.core.repository | 超级Repository接口 |
| SuperRepositoryImpl .py | PYTHON | com.wteam.superboot.core.repository | 超级Repository实现类 |
| UserkeyPo.py | PYTHON | com.lulu.eating.user.entity.po | 用户登录凭证Po类 |

**2 功能实现说明**



图2-2-2-1-9-2修改登录密码时序图

### 订单管理

1、简介

此为用户的订单管理的两个活动图，图1为用户查看所有订单及其详情，图2为用户下单，取消订单，催单，确定收货，退单



图2-2-2-4-1 订单模块活动图



图2-2-2-4-2 订单模块活动图2

2、功能列表

|  |  |
| --- | --- |
| 1 | 查看所有订单 |
| 2 | 查看订单详情 |
| 3 | 用户下单 |
| 4 | 用户取消订单 |
| 5 | 用户催单 |
| 6 | 确认收货 |
| 7 | 用户退单 |
| 8 | 发布者接单 |
| 9 | 发布者处理退单 |

#### 查看所有订单

**1 功能设计描述**

**（1）类**

**1）OrderPo.py**

**2）OrderItemPo.py**

**3）OrderVo.py**

**4）OrderItemVo.py**

**5）OrderController.py**

**6）****OrderService.py**

**7）OrderRepository.py**

**8）OrderItemRepository.py**

**（2）类与类之间关系**

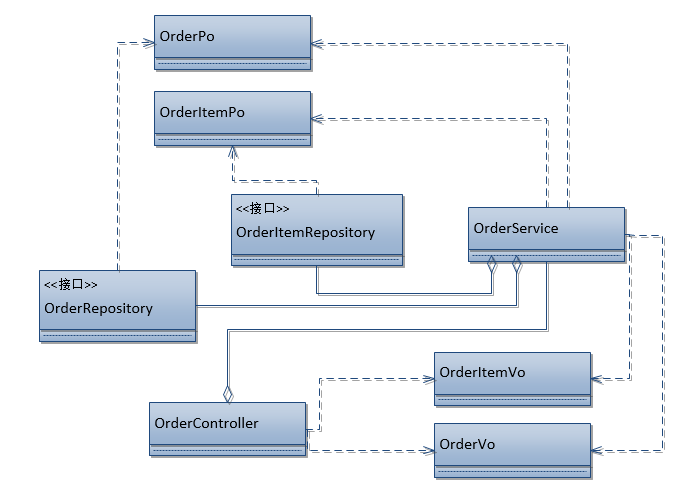


图2-2-2-4-1-1 类关系图

**（3）文件列表**

如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| OrderPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单类实体 |
| OrderItemPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单项目类实体 |
| OrderVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单类视图 |
| OrderItemVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单项目类视图 |
| OrderController.py | PYTHON | com.lulu.eatting.controller | 订单模块控制器 |
| OrderService.py | PYTHON | com.lulu.eatting.service | 订单模块业务逻辑层类 |
| OrderRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| OrderItemRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| userOrder.html | html | resources/static/views/order | 用户订单页面 |
| dealerOrder.html | html | resources/static/views/order | 发布者订单页面 |

**2 功能实现说明**

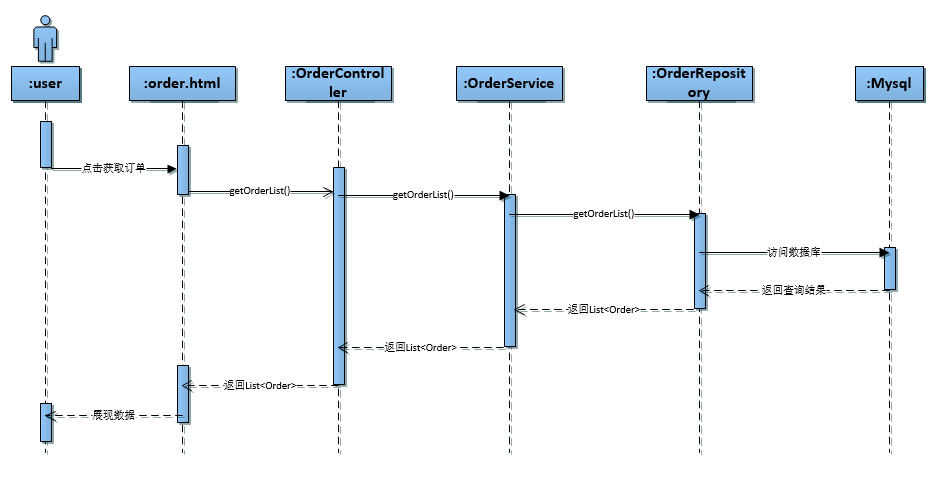


图2-2-2-4-1-2 查看所有订单时序图

#### 查看订单详情

**1 功能设计描述**

**（1）类**

**1）OrderPo.py**

**2）OrderItemPo.py**

**3）OrderVo.py**

**4）OrderItemVo.py**

**5）OrderController.py**

**6）OrderService.py**

**7）OrderRepository.py**

**8）OrderItemRepository.py**

**（2）类与类之间关系**

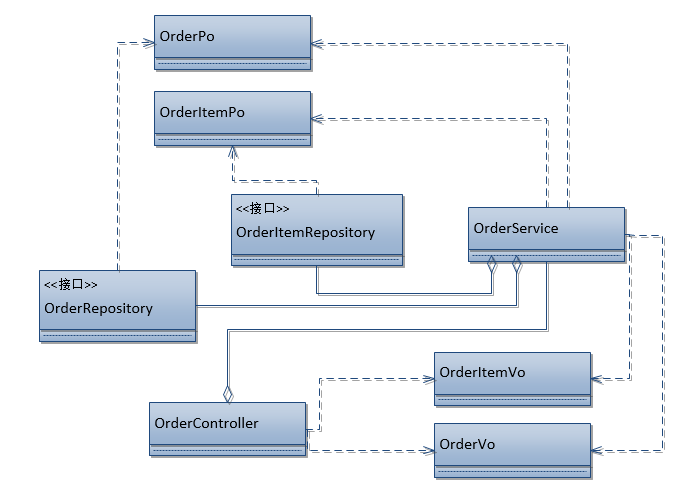


图2-2-2-4-1-2 类关系图

**（3）文件列表**

如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| OrderPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单类实体 |
| OrderItemPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单项目类实体 |
| OrderVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单类视图 |
| OrderItemVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单项目类视图 |
| OrderController.py | PYTHON | com.lulu.eatting.controller | 订单模块控制器 |
| OrderService.py | PYTHON | com.lulu.eatting.service | 订单模块业务逻辑层类 |
| OrderRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| OrderItemRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| userOrder.html | html | resources/static/views/order | 用户订单页面 |
| dealerOrder.html | html | resources/static/views/order | 发布者订单页面 |

**2 功能实现说明**

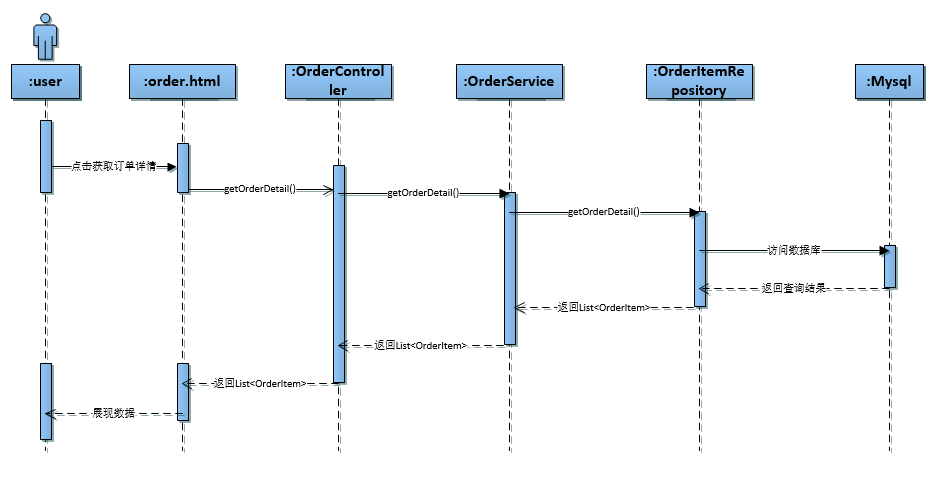


图2-2-2-4-2-2 查看订单详情时序图

#### 用户下单

**1 功能设计描述**

**（1）类**

**1）OrderPo.py**

**2）OrderItemPo.py**

**3）OrderVo.py**

**4）OrderItemVo.py**

**5）OrderController.py**

**6）OrderService.py**

**7）OrderRepository.py**

**8）OrderItemRepository.py**

**（2）类与类之间关系**

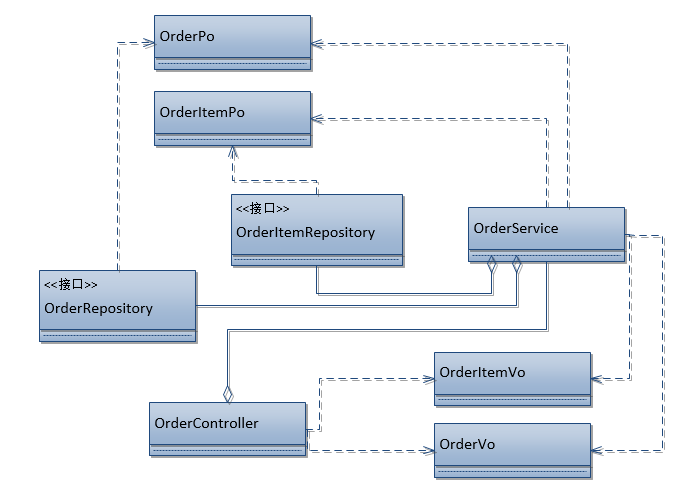


图2-2-2-4-3-1 类关系图

**（3）文件列表**

如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| OrderPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单类实体 |
| OrderItemPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单项目类实体 |
| OrderVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单类视图 |
| OrderItemVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单项目类视图 |
| OrderController.py | PYTHON | com.lulu.eatting.controller | 订单模块控制器 |
| OrderService.py | PYTHON | com.lulu.eatting.service | 订单模块业务逻辑层类 |
| OrderRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| OrderItemRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| createOrder.html | html | resources/static/views/order | 用户下单页面 |

**2 功能实现说明**

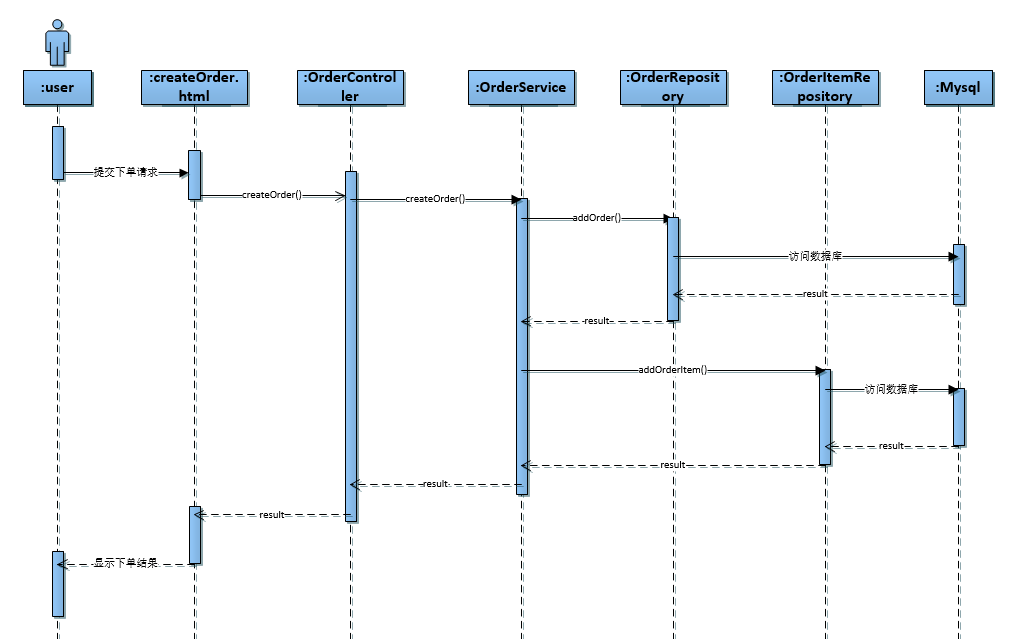


图2-2-2-4-1-3-2 用户下单时序图

#### 用户取消订单

**1 功能设计描述**

**（1）类**

**1）OrderPo.py**

**2）OrderItemPo.py**

**3）OrderVo.py**

**4）OrderItemVo.py**

**5）OrderController.py**

**6）OrderService.py**

**7）OrderRepository.py**

**8）OrderItemRepository.py**

**（2）类与类之间关系**

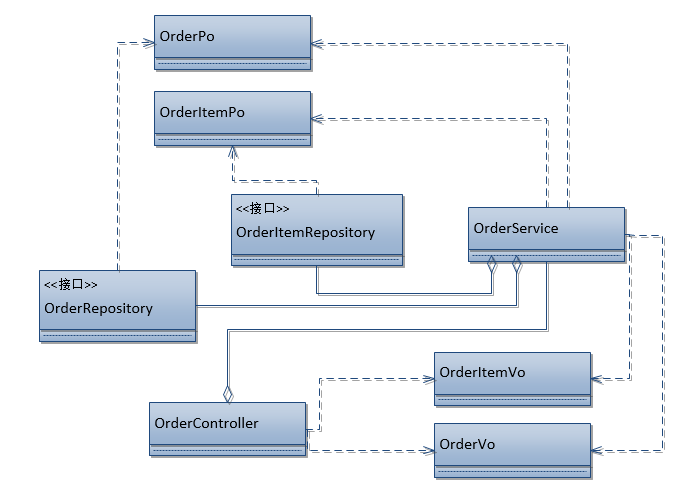


图2-2-2-4-4-1 类关系图

**（3）文件列表**

如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| OrderPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单类实体 |
| OrderItemPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单项目类实体 |
| OrderVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单类视图 |
| OrderItemVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单项目类视图 |
| OrderController.py | PYTHON | com.lulu.eatting.controller | 订单模块控制器 |
| OrderService.py | PYTHON | com.lulu.eatting.service | 订单模块业务逻辑层类 |
| OrderRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| OrderItemRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| userOrder.html | html | resources/static/views/order | 用户订单页面 |

**2 功能实现说明**

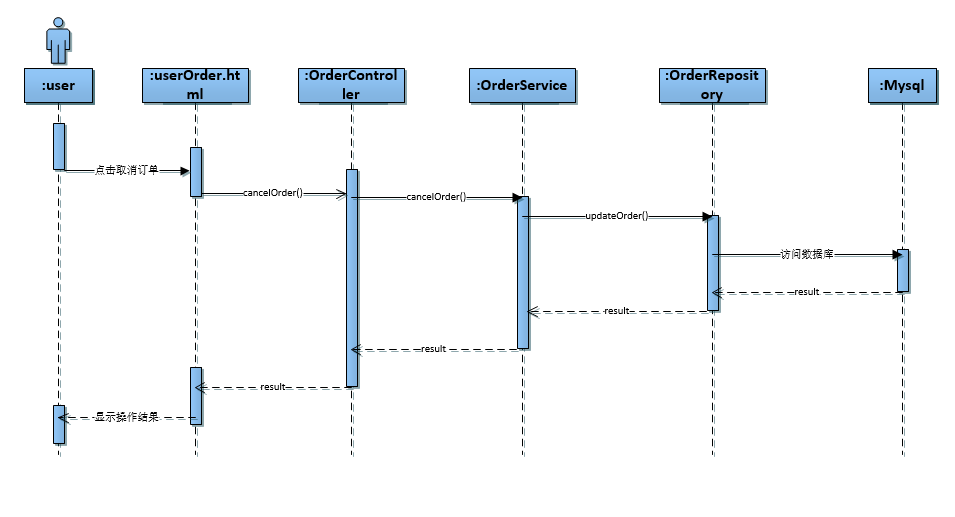


图2-2-2-4-4-2用户取消订单时序图

#### 用户催单

**1 功能设计描述**

**（1）类**

**1）OrderController.py**

**2）OrderService.py**

**（2）类与类之间关系**

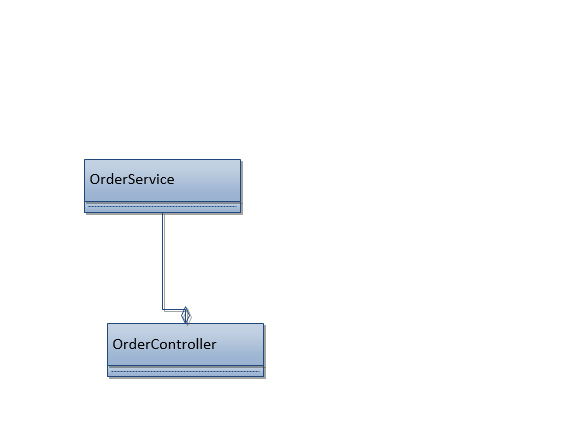


图2-2-2-4-5-1 类关系图

**（3）文件列表**

如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| OrderController.py | PYTHON | com.lulu.eatting.controller | 订单模块控制器 |
| OrderService.py | PYTHON | com.lulu.eatting.service | 订单模块业务逻辑层类 |
| userOrder.html | html | resources/static/views/order | 用户订单页面 |

**2 功能实现说明**

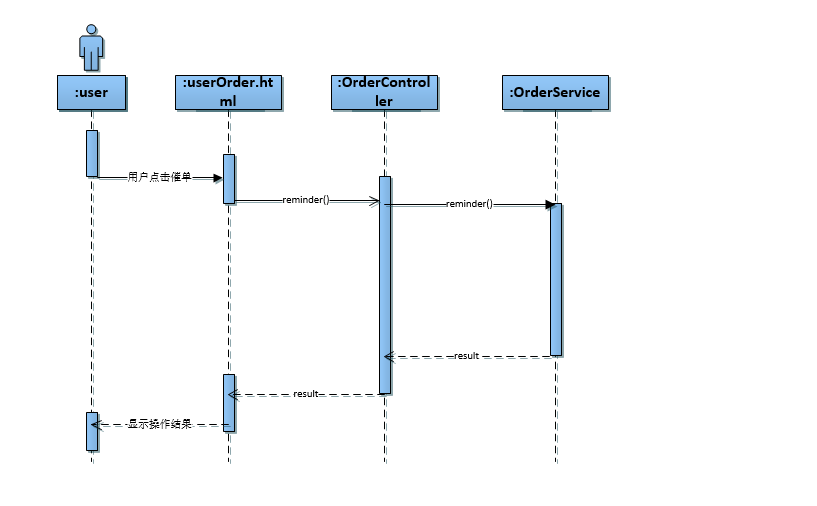


图2-2-2-4-5-1 用户催单时序图

#### 用户确认收货

**1 功能设计描述**

**（1）类**

**1）OrderPo.py**

**2）OrderVo.py**

**3）OrderController.py**

**4）OrderService.py**

**5）OrderRepository.py**

**（2）类与类之间关系**

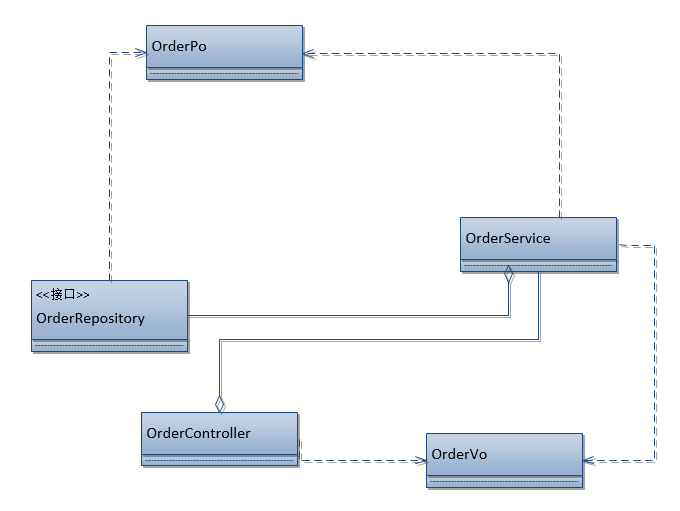


图2-2-2-4-6-1 类关系图

**（3）文件列表**

如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| OrderPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单类实体 |
| OrderVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单类视图 |
| OrderController.py | PYTHON | com.lulu.eatting.controller | 订单模块控制器 |
| OrderService.py | PYTHON | com.lulu.eatting.service | 订单模块业务逻辑层类 |
| OrderRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| userOrder.html | html | resources/static/views/order | 用户订单页面 |

**2 功能实现说明**

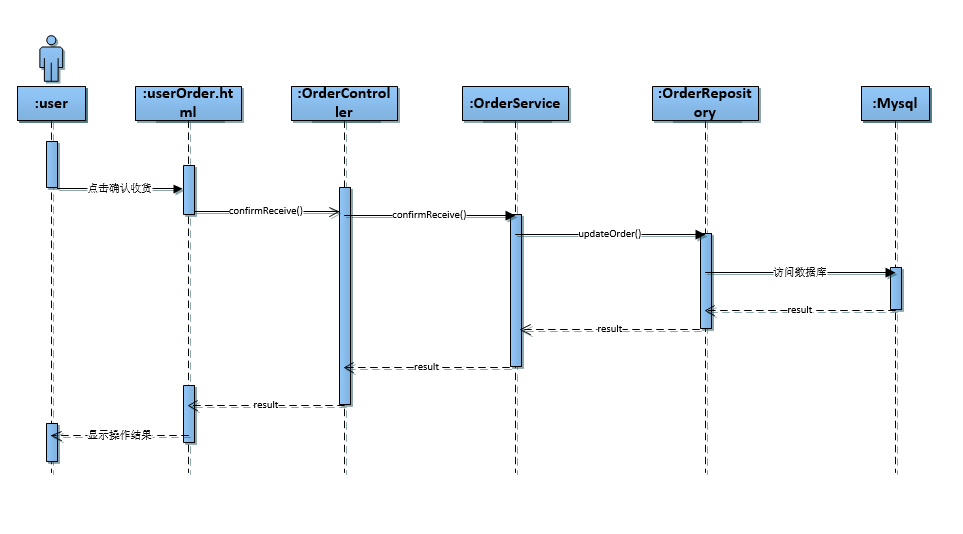


图2-2-2-4-6-1 用户确认收货时序图

#### 用户退单

**1 功能设计描述**

**（1）类**

**1）OrderPo.py**

**2）OrderVo.py**

**3）OrderController.py**

**4）OrderService.py**

**5）OrderRepository.py**

**（2）类与类之间关系**

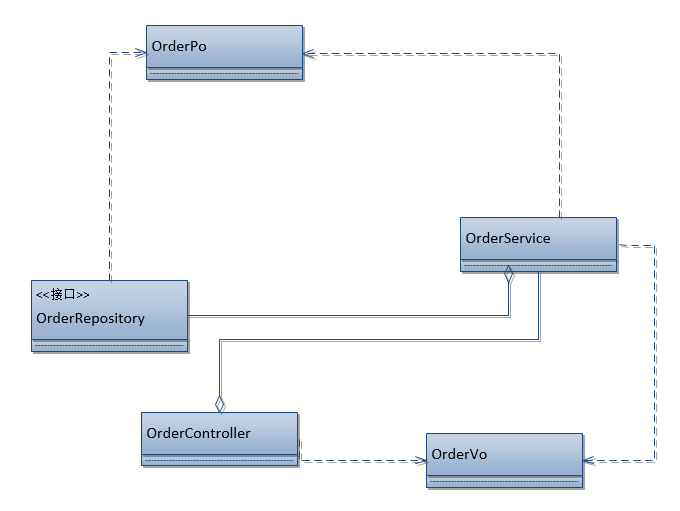


图2-2-2-4-7-1 类关系图

**（3）文件列表**

如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| OrderPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单类实体 |
| OrderVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单类视图 |
| OrderController.py | PYTHON | com.lulu.eatting.controller | 订单模块控制器 |
| OrderService.py | PYTHON | com.lulu.eatting.service | 订单模块业务逻辑层类 |
| OrderRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| userOrder.html | html | resources/static/views/order | 用户订单页面 |

**2 功能实现说明**

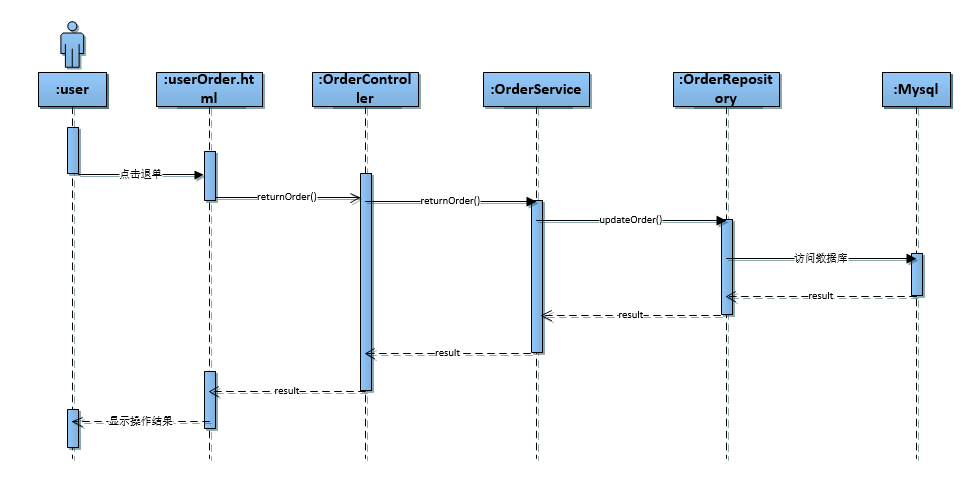


图2-2-2-4-7-1 用户退单时序图

#### 发布者接单

**1 功能设计描述**

**（1）类**

**1）OrderPo.py**

**2）OrderVo.py**

**3）OrderController.py**

**4）OrderService.py**

**5）OrderRepository.py**

**（2）类与类之间关系**

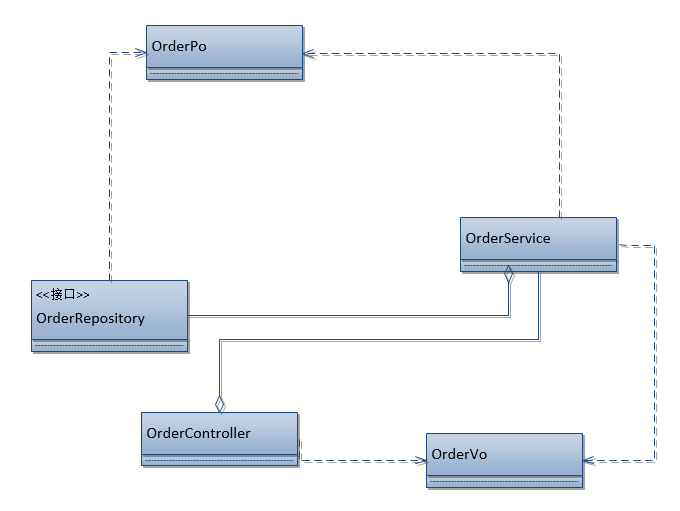


图2-2-2-4-8-1 类关系图

**（3）文件列表**

如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| OrderPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单类实体 |
| OrderVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单类视图 |
| OrderController.py | PYTHON | com.lulu.eatting.controller | 订单模块控制器 |
| OrderService.py | PYTHON | com.lulu.eatting.service | 订单模块业务逻辑层类 |
| OrderRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| dealerOrder.html | html | resources/static/views/order | 发布者订单页面 |

**2 功能实现说明**

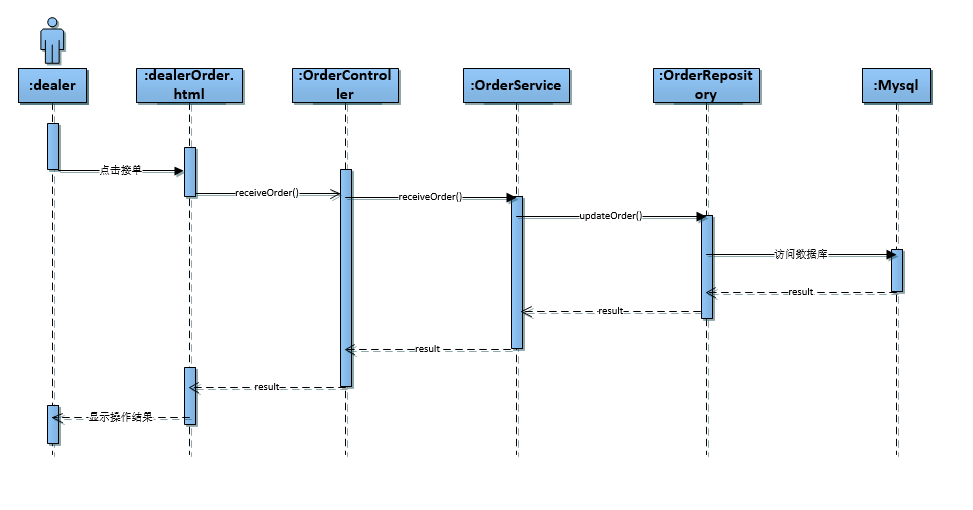


图2-2-2-4-8-2发布者接单时序图

#### 发布者处理退单

**1 功能设计描述**

**（1）类**

**1）OrderPo.py**

**2）OrderVo.py**

**3）OrderController.py**

**4）OrderService.py**

**5）OrderRepository.py**

**（2）类与类之间关系**

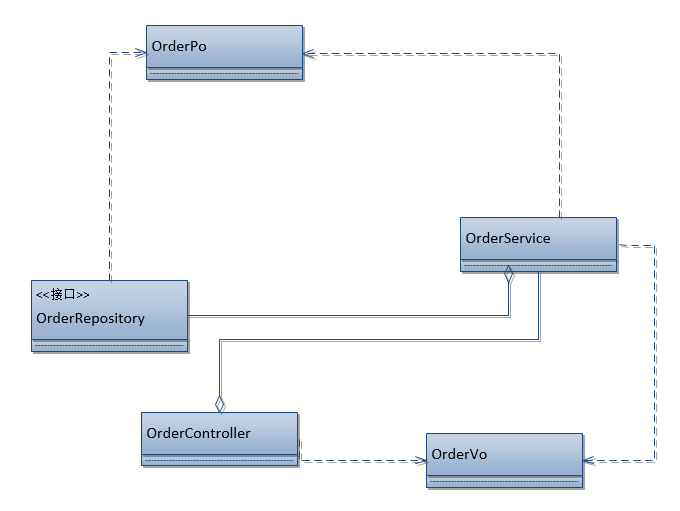


图2-2-2-4-9-1 类关系图

**（3）文件列表**

如下表所示

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| OrderPo.py | PYTHON | com.lulu.eatting.entiry.po | 订单类实体 |
| OrderVo.py | PYTHON | com.lulu.eatting.entiry.vo | 订单类视图 |
| OrderController.py | PYTHON | com.lulu.eatting.controller | 订单模块控制器 |
| OrderService.py | PYTHON | com.lulu.eatting.service | 订单模块业务逻辑层类 |
| OrderRepository.py | PYTHON | com.lulu.eatting.repository | 订单模块持久层接口 |
| dealerOrder.html | html | resources/static/views/order | 发布者订单页面 |

**2 功能实现说明**

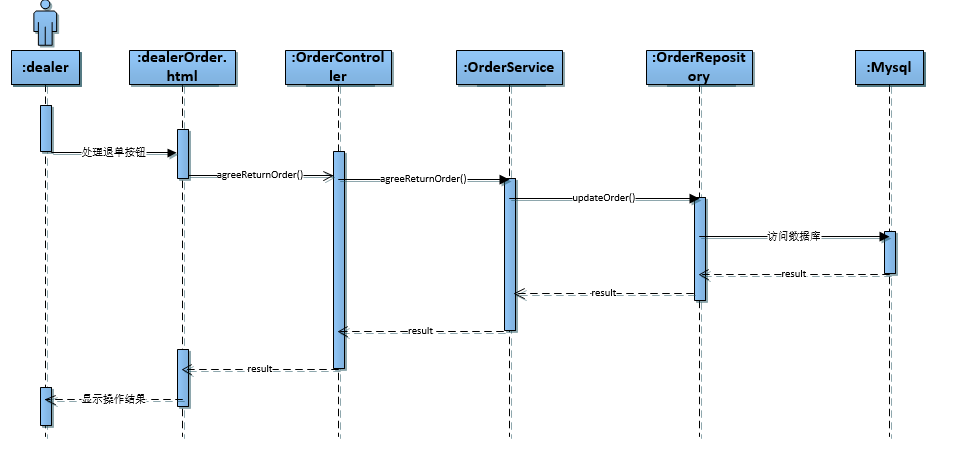


图2-2-2-4-9-2发布者处理退单时序图

### 评论管理

1、简介

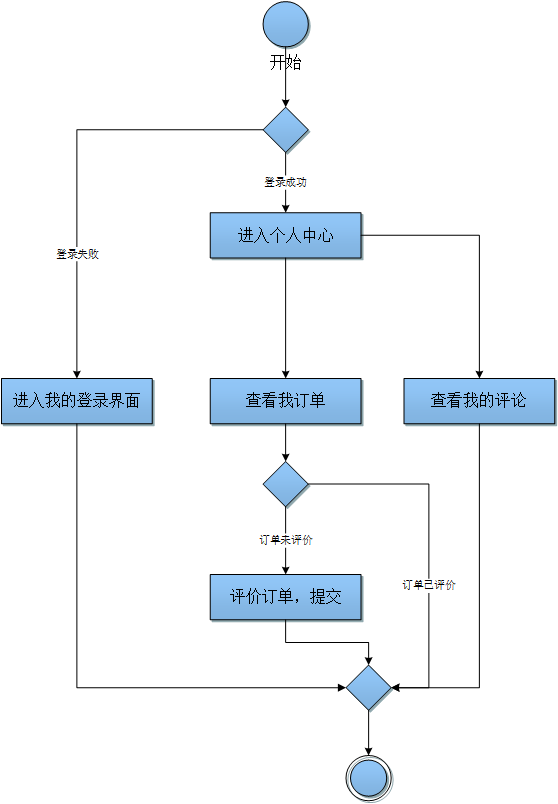
评论管理包含的子功能有：

用户评价管理：

用户能够查看自己所做的评论，对自己完成的订单中的商品进行评价，包括商品评分和备注信息。

商家评论管理：

商家能够查看所有评论，对完成的订单中未回复评论的商品进行回复。



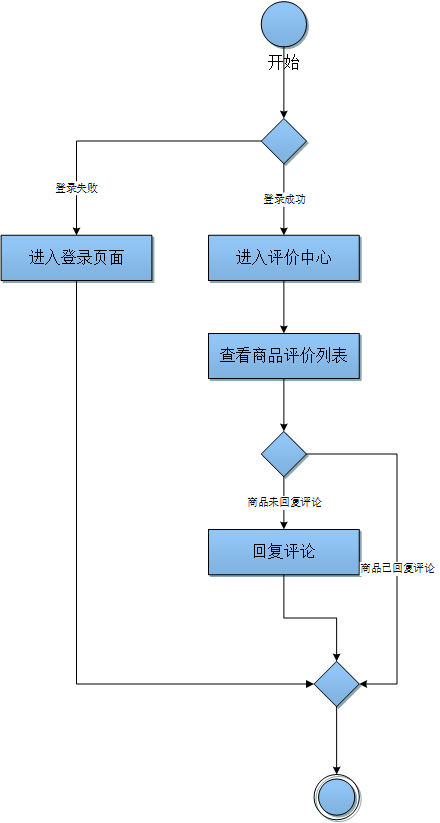


图2-2-2-5评论管理模块业务流程图

1. 功能列表

|  |  |
| --- | --- |
| 1 | 查看自己的评价 |
| 2 | 用户评价订单 |
| 3 | 发布者查看所有评价 |
| 4 | 发布者回复评论 |

#### 用户查看自己的评论

**1 功能设计描述**

**（1）类**

**1）BooksCommentPo.py**

**2）BooksCommentController.py**

**3）BooksCommentRepository.interface**

**4）BooksCommentService.py**

**5）BooksCommentVo.py**

**（2）类与类之间关系**

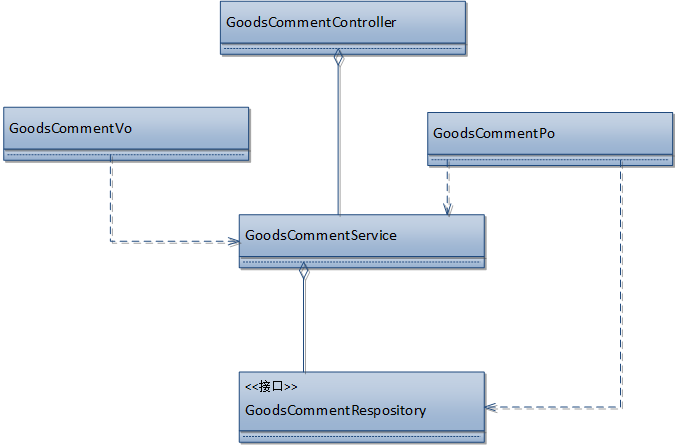


图2-2-2-5-1-1用户查看自己评价类关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| BooksCommentPo.py | PYTHON | com.lulu.eatting.entity.po | 数据层商品评论的实体类 |
| BooksCommentVo.py | PYTHON | com.lulu.eatting.entity.vo | 视图层商品评论的实体类 |
| BooksCommentController.py | PYTHON | com.lulu.eatting.controller | 商品评论的控制层 |
| BooksCommentService.py | PYTHON | com.lulu.eatting.service | 商品评论的业务逻辑操作 |
| BooksCommentRepository.interface | PYTHON | com.lulu.eatting.repository | 商品评论的数据访问接口 |
| goodCommentDetail.html | Html | resources/static/views/goodCommentDetail.html | 用户查看自己的的评论 |

**2 功能实现说明**

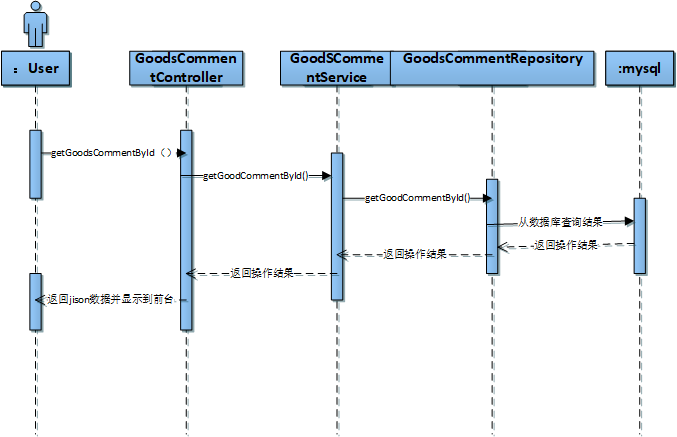


图2-2-2-5-1-2用户查看自己的评价时序图

#### 用户评价商品

**1 功能设计描述**

**（1）类**

**1）BooksCommentPo.py**

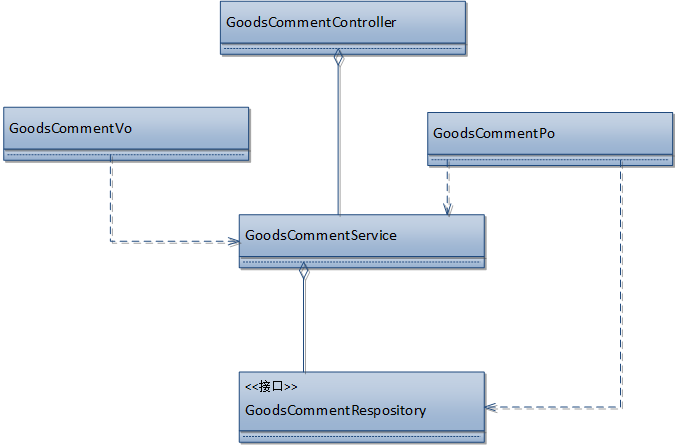
**2）BooksCommentController.py**

**3）BooksCommentRepository.interface**

**4）BooksCommentService.py**

**5）BooksCommentVo.py**

**（2）类与类之间关系**

图2-2-2-5-2-1用户评价商品类关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| BooksCommentPo.py | PYTHON | com.lulu.eatting.entity.po | 数据层商品评论的实体类 |
| BooksCommentVo.py | PYTHON | com.lulu.eatting.entity.vo | 视图层商品评论的实体类 |
| BooksCommentController.py | PYTHON | com.lulu.eatting.controller | 商品评论的控制层 |
| BooksCommentService.py | PYTHON | com.lulu.eatting.service | 商品评论的业务逻辑操作 |
| BooksCommentRepository.interface | PYTHON | com.lulu.eatting.repository | 商品评论的数据访问接口 |
| addBooksComment.html | Html | resources/static/views/addBooksComment.html | 用户评价商品 |

1. **功能实现说明**

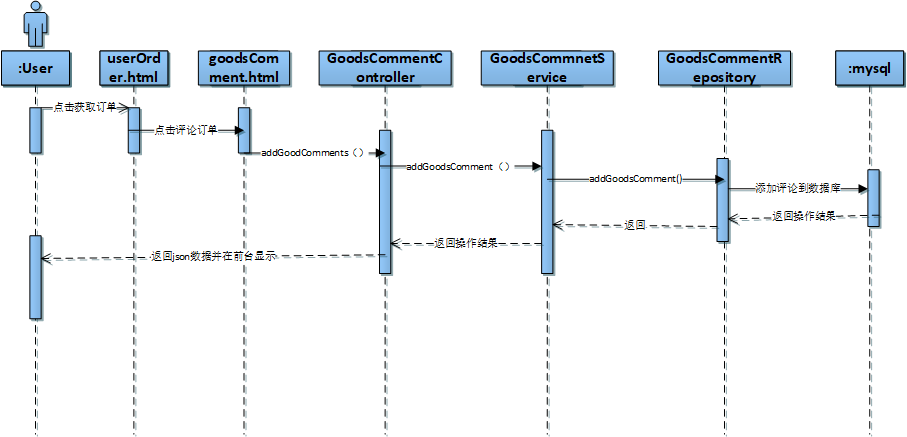
****

图2-2-2-5-2-2用户评价商品时序图

#### 发布者查看评价列表

**1 功能设计描述**

**（1）类**

**1）BooksCommentPo.py**

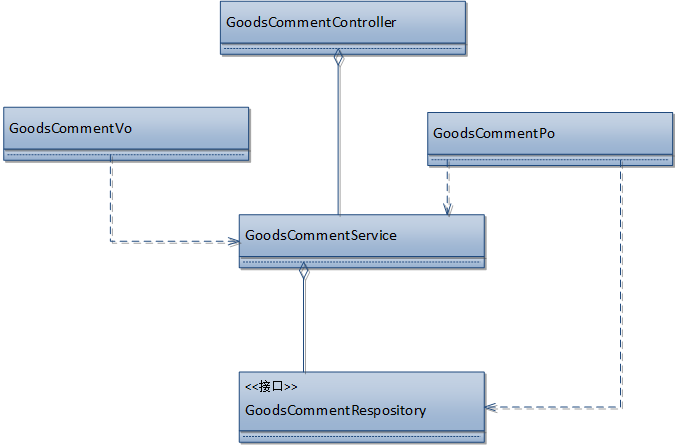
**2）BooksCommentController.py**

**3）BooksCommentRepository.interface**

**4）BooksCommentService.py**

**5）BooksCommentVo.py**

**（2）类与类之间关系**

图2-2-2-5-3-1发布者查看评价列表关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| BooksCommentPo.py | PYTHON | com.lulu.eatting.entity.po | 数据层商品评论的实体类 |
| BooksCommentVo.py | PYTHON | com.lulu.eatting.entity.vo | 视图层商品评论的实体类 |
| BooksCommentController.py | PYTHON | com.lulu.eatting.controller | 商品评论的控制层 |
| BooksCommentService.py | PYTHON | com.lulu.eatting.service | 商品评论的业务逻辑操作 |
| BooksCommentRepository.interface | PYTHON | com.lulu.eatting.repository | 商品评论的数据访问接口 |
| BooksCommentList.html | Html | resources/static/views/goodCommentList.html | 发布者查看所有商品评价列表 |

1. **功能实现说明**

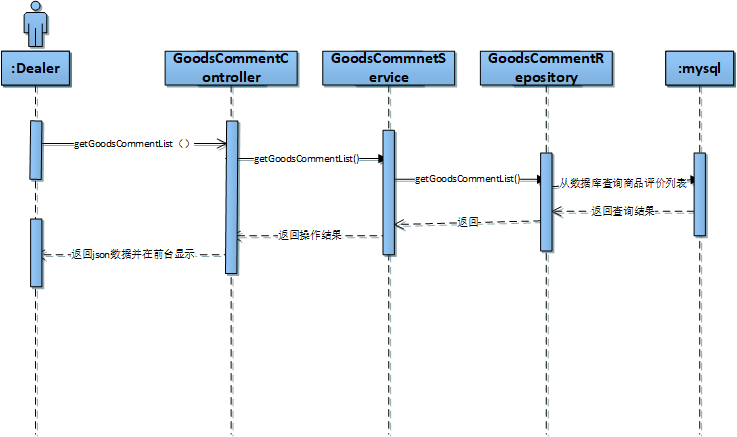
****

图2-2-2-5-3-2发布者查看商品评价时序图

#### 发布者回复商品评论

**1 功能设计描述**

**（1）类**

**1）BooksCommentPo.py**

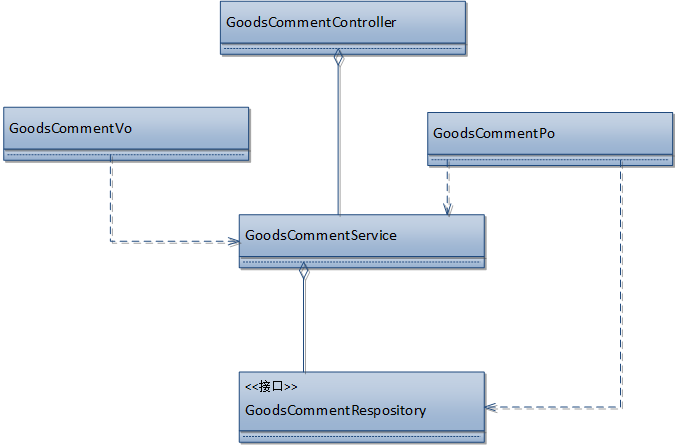
**2）BooksCommentController.py**

**3）BooksCommentRepository.interface**

**4）BooksCommentService.py**

**5）BooksCommentVo.py**

**（2）类与类之间关系**

图2-2-2-5-3-1发布者查看评价列表关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| BooksCommentPo.py | PYTHON | com.lulu.eatting.entity.po | 数据层商品评论的实体类 |
| BooksCommentVo.py | PYTHON | com.lulu.eatting.entity.vo | 视图层商品评论的实体类 |
| BooksCommentController.py | PYTHON | com.lulu.eatting.controller | 商品评论的控制层 |
| BooksCommentService.py | PYTHON | com.lulu.eatting.service | 商品评论的业务逻辑操作 |
| BooksCommentRepository.interface | PYTHON | com.lulu.eatting.repository | 商品评论的数据访问接口 |
| editBooksComment.html | Html | resources/static/views/editCommentList.html | 发布者回复商品评论 |

1. **功能实现说明**

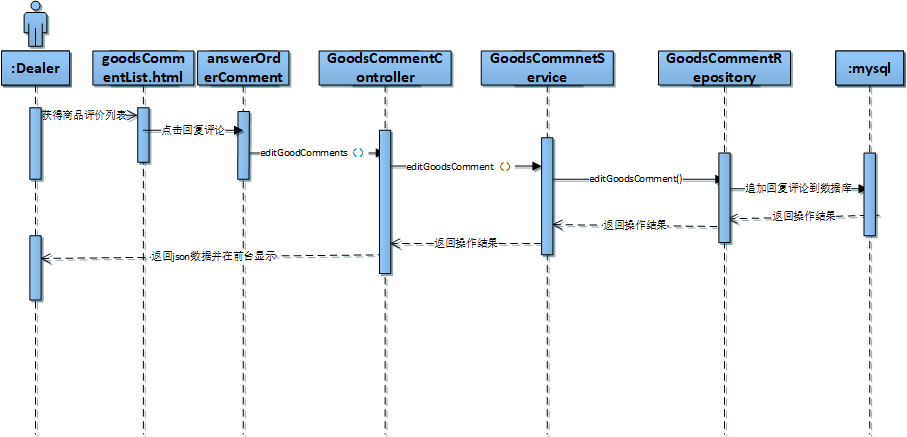
****

图2-2-2-5-3-2发布者回复商品评论时序图

### 支付管理

1、简介

此模块是在线支付模块，主要包括用户在线支付订单金额的功能，包括在线支付、退款。

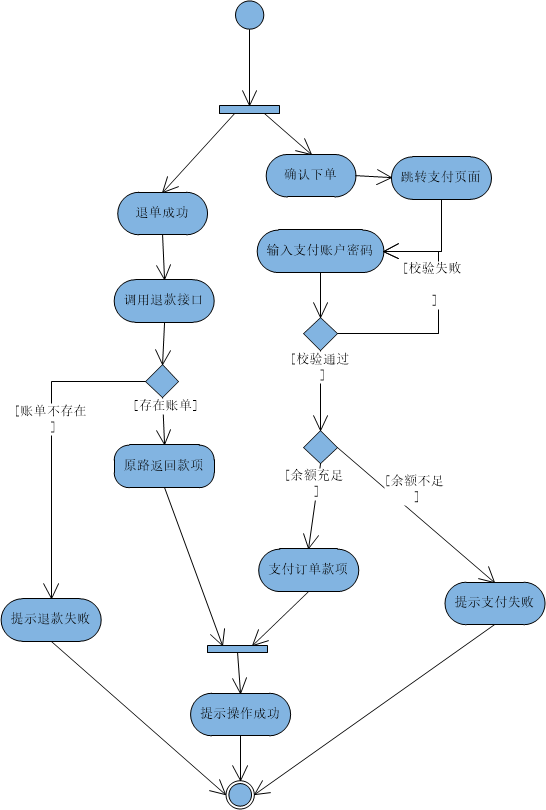


图2-2-7-1 支付模块活动图

2、功能列表

|  |  |
| --- | --- |
| 1 | 在线支付 |
| 2 | 退款 |

#### 在线支付

**1 功能设计描述**

**（1）类**

**1）OrderRepository.py**

**2）PaymentService.py**

**3）PaymentController.py**

**4）OrderPo.py**

**5）OrderVo.py**

**6）PaymentHelper.py**

**（2）类与类之间关系**

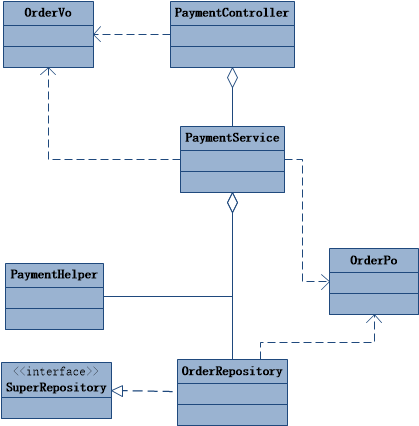
****

图2-2-2-7-1-1 在线支付功能类关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| userPayment.html | HTML | resources/static/views | 支付页面，用户从此跳转到支付页面 |
| OrderRepository.py | Python | com.lulu.eatting.repositotry | 订单数据访问接口 |
| PaymentService.py | Python | com.lulu.eatting.service | 支付模块业务实现类 |
| PaymentController.py | Python | com.lulu.eatting.controller | 支付模块页面控制器实现类 |
| OrderPo.py | Python | com.lulu.eatting.entity.po | 订单信息PO类 |
| OrderVo.py | Python | com.lulu.eatting.entity.vo | 订单信息VO类 |
| PaymentHelper.py | Python | com.lulu.eatting.utils | 用户在线支付辅助类 |

**2 功能实现说明**

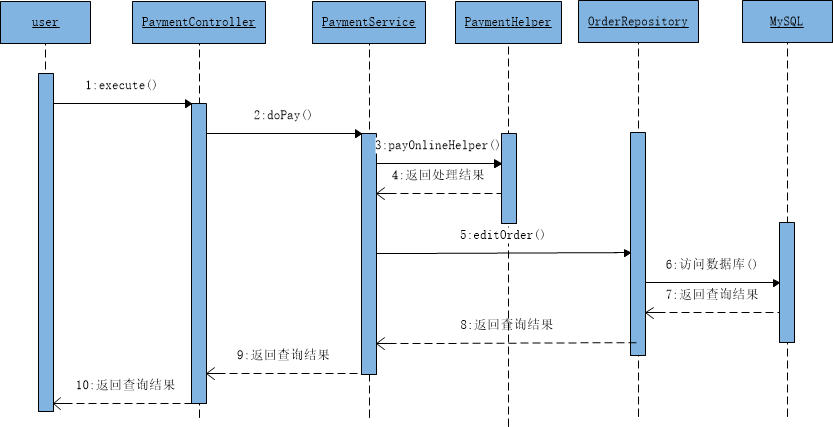
****

图2-2-2-7-1-2 在线支付功能时序图

#### 退款

**1 功能设计描述**

**（1）类**

**1）OrderRepository.py**

**2）PaymentService.py**

**3）PaymentController.py**

**4）OrderPo.py**

**5）OrderVo.py**

**6）PaymentHelper.py**

**（2）类与类之间关系**

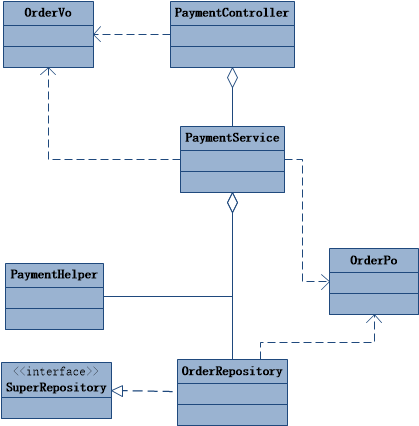
****

图2-2-2-7-2-1 退款功能类关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| userPayment.html | HTML | resources/static/views | 支付页面，用户从此跳转到支付页面 |
| OrderRepository.py | Python | com.lulu.eatting.repositotry | 订单数据访问接口 |
| PaymentService.py | Python | com.lulu.eatting.service | 支付模块业务实现类 |
| PaymentController.py | Python | com.lulu.eatting.controller | 支付模块页面控制器实现类 |
| OrderPo.py | Python | com.lulu.eatting.entity.po | 订单信息PO类 |
| OrderVo.py | Python | com.lulu.eatting.entity.vo | 订单信息VO类 |
| PaymentHelper.py | Python | com.lulu.eatting.utils | 用户在线支付辅助类 |

**2 功能实现说明**

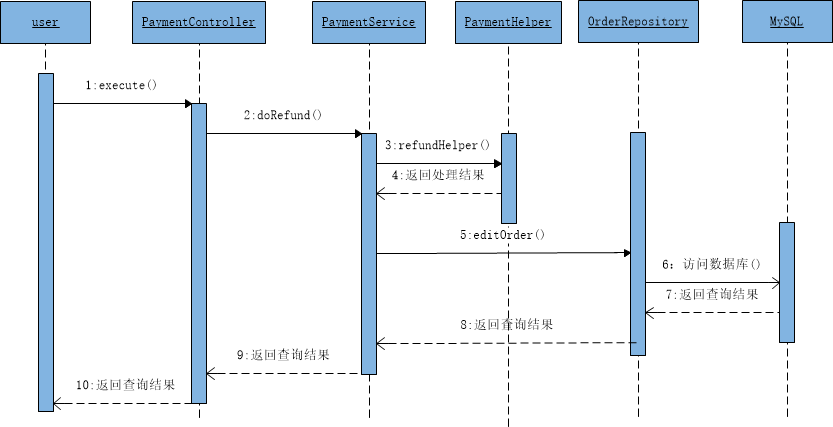
****

图2-2-2-7-2-2 退款功能时序图

### 书籍管理

1 简介：

书籍管理主要包括增删改查（征书墙，书城，竞拍）由发布者发布在平台上的书籍，

#### 添加书籍

**1 功能设计描述**

**（1）类**

**1）SuperReositoryImpl.py**

**2）SuperReository.py**

**3）BooksPo.py**

**4）BooksController.py**

**5）BooksService.py**

**6）BooksRepository.py**

**7）BookscategoryPo.py**

**（2）类与类之间关系**



图2-2-2-2-6-1 商家添加书籍类关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| dealerBooksList.html | HTML | Resources/static/views/dealer/dealerBooksList.html | 商家商品列表页面 |
| BooksPo.py | PYTHON | com.lulu.eatting.entity | 实体类Books |
| BookscategoryPo | PYTHON | com.lulu.eatting.entity | 实体类Bookscategory |
| BooksController.py | PYTHON | com.lulu.eatting.controller | 添加书籍控制器 |
| BooksService.py | PYTHON | com.lulu.eatting.service | 添加书籍逻辑层类 |
| BooksRepository.py | PYTHON | com.lulu.eatting.repository | 添加书籍持久层类 |
| SuperRepository.py | PYTHON | com.lulu.eatting.repository | 添加书籍持久层接口 |
| SuperRepositoryImpl.py | PYTHON | com.lulu.eatting.repository | 添加书籍持久层接口 |

**2 功能实现说明**



图2-2-2-2-6-2 商家添加书籍时序图

#### 删除书籍

**1 功能设计描述**

**（1）类**

**1）SuperReositoryImpl.py**

**2）SuperReository.py**

**3）BooksPo.py**

**4）BooksController.py**

**5）BooksService.py**

**6）BooksRepository.py**

**（2）类与类之间关系**



图2-2-2-2-7-1 商家删除书籍类关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| dealerBooksList.html | HTML | Resources/static/views/dealer/dealerBooksList.html | 商家商品列表页面 |
| BooksPo.py | PYTHON | com.lulu.eatting.entity | 实体类Books |
| BooksController.py | PYTHON | com.lulu.eatting.controller | 删除书籍控制器 |
| BooksService.py | PYTHON | com.lulu.eatting.service | 删除书籍逻辑层类 |
| BooksRepository.py | PYTHON | com.lulu.eatting.repository | 删除书籍持久层类 |
| SuperRepository.py | PYTHON | com.lulu.eatting.repository | 删除书籍持久层接口 |
| SuperRepositoryImpl.py | PYTHON | com.lulu.eatting.repository | 删除书籍持久层接口 |

**2 功能实现说明**



图2-2-2-2-7-2 商家删除书籍时序图

#### 修改书籍

**1 功能设计描述**

**（1）类**

**1）SuperReositoryImpl.py**

**2）SuperReository.py**

**3）BooksPo.py**

**4）BooksController.py**

**5）BooksService.py**

**6）BooksRepository.py**

**7）BookscategoryPo.py**

**（2）类与类之间关系**



图2-2-2-2-8-1 商家修改书籍类关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| dealerUpdateBooks.html | HTML | Resources/static/views/dealer/dealerUpdateBooks.html | 商家修改商品信息页面 |
| BooksPo.py | PYTHON | com.lulu.eatting.entity | 实体类Books |
| BookscategoryPo | PYTHON | com.lulu.eatting.entity | 实体类Bookscategory |
| BooksController.py | PYTHON | com.lulu.eatting.controller | 修改书籍控制器 |
| BooksService.py | PYTHON | com.lulu.eatting.service | 修改书籍逻辑层类 |
| BooksRepository.py | PYTHON | com.lulu.eatting.repository | 修改书籍持久层类 |
| SuperRepository.py | PYTHON | com.lulu.eatting.repository | 修改书籍持久层接口 |
| SuperRepositoryImpl.py | PYTHON | com.lulu.eatting.repository | 修改书籍持久层接口 |

**2 功能实现说明**



图2-2-2-2-8-2 商家修改书籍时序图

#### 查看书籍

**1 功能设计描述**

**（1）类**

**1）SuperReositoryImpl.py**

**2）SuperReository.py**

**3）BooksPo.py**

**4）BooksController.py**

**5）BooksService.py**

**6）BooksRepository.py**

**7）BookscategoryPo.py**

**（2）类与类之间关系**



图2-2-2-2-9-1 商家查看书籍类关系图

**（3）文件列表**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 存放位置 | 说明 |
| dealerUpdateBooks.html | HTML | Resources/static/views/dealer/dealerUpdateBooks.html | 商家修改商品信息页面 |
| BooksPo.py | PYTHON | com.lulu.eatting.entity | 实体类Books |
| BookscategoryPo | PYTHON | com.lulu.eatting.entity | 实体类Bookscategory |
| BooksController.py | PYTHON | com.lulu.eatting.controller | 查看书籍控制器 |
| BooksService.py | PYTHON | com.lulu.eatting.service | 查看书籍逻辑层类 |
| BooksRepository.py | PYTHON | com.lulu.eatting.repository | 查看书籍持久层类 |
| SuperRepository.py | PYTHON | com.lulu.eatting.repository | 查看书籍持久层接口 |
| SuperRepositoryImpl.py | PYTHON | com.lulu.eatting.repository | 查看书籍持久层接口 |

**2 功能实现说明**



图2-2-2-2-9-2 商家查看书籍时序图

# 接口设计

## 数据请求与返回接口

以分页形式获取书籍列表信息接口：

请求URL：http://域名:端口号/book/?page=页码数

数据返回格式：

// 20180110192439

// http://localhost:8000/book/?format=json&page=2

{

"count": 52,

"next": "http://localhost:8000/book/?format=json&page=3",

"previous": "http://localhost:8000/book/?format=json",

"results": [

{

"id": 11,

"category": {

"id": 2,

"sub\_cat": [

{

"id": 3,

"sub\_cat": [

],

"name": "三体",

"press": "科技世界",

"version": "第一版",

"desc": "科技届的幸事",

"price": 199,

"display\_time": "2018-01-11T11:40:27.197779",

"add\_time":"2018-01-02T11:40:27.197779",

"buyout\_price": 500,

"book\_image": "http://localhost:8000/media/books /images/3\_P\_1448945490837.jpg",

"book\_num":100,

"strike\_price":300,

},

{

。。。

# 系统出错处理

## 出错信息

所有出错信息均以字符串的方式，在系统运行界面中显示。所有出错信息分为三种：

一种是由于输入错误信息超出或不符合预定各式的错误，属于处理错误

二是由于系统的预设不能执行的错误，属于设定错误

三是由于网路传输超时、服务器响应超时等属于系统错误

对于处理错误需在操作成功判断及输入数据验证模块由数据进行数据分析，判断错误类型，再生成相应的错误提示语句，送到输出模块中，对于设定错误，应在开始提交信息类别中，依据权限等判定错误类别，再生成相应出错信息语句，输出到输出模块中。对于系统错误，根据Nginx服务器的响应内容，判断错误类别输出。出错信息必须给出相应的出错原因，如：

“用户名不存在”

“您的登录数据请求来源与钓鱼网站”

“用户身份信息由管理员给定，不能修改”

“两次输入的密码不一致”

“输入验证码错误”

## 补救措施

所有的客户机都必须安装不间断电源以防止停电或电压不稳造成的数据丢失的损失。若真断电时，客户机上将不会有太大的影响，主要是服务器上：在断电后恢复过程可采用数据库的日志文件，对其进行ROLLBACK 处理，对数据进行恢复。在网络传输方面，可考虑建立一条成本较低的后备网络，以保证当主网络断路时数据的通信。在硬件方面要选择较可靠、稳定的服务器机种，保证系统运行时的可靠性。

## 系统维护设计

维护方面主要为对服务器上的数据库数据进行维护。可使用 数据库管理系统的数据库的维护功能机制。例如，定期为数据库进行Backup，维护管理数据库死锁问题和维护数据库内数据的一致性等。