**文档编号：*易行交易平台* – SDS – 3*.1***

***易行交易平台***

**软件设计规格说明书**

**日期：2020/10/24**

**文档变更历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **变更日期** | **变更人员** | **变更内容详情描述** | **版本** |
| 1 | 2020/10/13 | 曾钟鑫 林高毅 | 易行交易平台软件设计规格说明书 | v1.0 |
| 2 | 2020/10/20 | 曾钟鑫 林高毅 | 完成易行交易平台软件设计规格说明书的最终版本 | v2.0 |
| 3 | 2020/10/24 | 曾钟鑫 李强 徐敬博 林高毅 刘凯威 石梓玥 | 完成数据库具体设计，添加类的描述与说明，界面设计的完善，软件体系结构图的修正。 | v3.0 |
| 4 | 2020/10/28 | 曾钟鑫 | 进一步完善数据库 | V3.1 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

**1、引言**

**1.1 编写目的**

**1.2 读者对象**

**1.3 软件项目概述**

**1.4 文档概述**

**1.5 定义**

**1.6 参考资料**

**2、软件设计约束**

**2.1 软件设计目标和原则**

**2.2 软件设计的约束和限制**

**3、软件设计**

**3.1 软件体系结构设计**

**3.2 用户界面设计**

**3.3 用例设计**

**3.4 类设计**

**3.5 数据设计**

**3.6 部署设计**

# 1、引言

### 1.1 编写目的

软件设计过程包括的体系结构设计、用户界面设计、用例设计、子系统/构件设计、类设计和数据设计分别从不同的层次（从宏观到微观、从全局到局部）、不同的视角（从结构到行为、从模块到数据）对软件系统进行了设计，产生了不用的软件制品（如体系结构模型、用例实现模型、用户界面模型、子系统/构件模型、数据设计模型、部署模型等等）。在完成上述所有设计工作之后，需要将这些软件设计成果进行整合，形成一个系统、完整的软件设计方案。本文档以软件设计规格说明书的形式描述了该设计方案，方便评审人员对设计方案的正确性、合理性等方面进行评审。

### 1.2 读者对象

用户、软件设计人员、程序员、软件需求分析人员、质量保证人员、软件测试工程师、配置管理工程师。

### 1.3 软件项目概述

项目名称: 易行交易平台系统

用户单位: 北京信息科技大学

开发单位: 北京信息科技大学计算机学院17级软件工程专业易行团队

软件项目的背景和大致功能：

易行交易平台系统提供许多方便学生学习与生活物品的信息，例如考虑以后不再使用的课本教材，书籍资料、想要淘汰更新的电脑，pad，手机，台灯，桌子等。

很多学生想要出手的书籍大多是平时被堆在宿舍，后来学生毕业时没有办法，只能被一次性低价摊售，甚至最后只是成为一堆废纸被丢进垃圾桶。这样不仅造成纸张的浪费，风吹日晒时还造成环境污染。而现在，学生可以将这些物品信息挂在易行交易平台系统，学生之间双向选择，创造一定的经济价值。

易行交易平台系统面向的用户大多是本校学生或者老师，在网络上发布的信息都是真实的可查询到的信息，减少了网络交易的风险，大大增加了交易的安全性。

### 1.4 文档概述

1）软件的设计约束部分。它包括软件设计目标和原则、软件设计受到的约束和限制。

2）软件的设计部分。它主要分为软件体系结构设计、用户界面设计、用例设计、类设计、数据设计以及部署设计。

### 1.5 定义

无

### 1.6 参考资料

[1].软件工程.齐治昌，谭庆平，宁洪.北京:高等教育出版社，2012

[2].需求分析与设计.马素霞译.北京:机械工业出版社，2009

[3].面向使用的软件设计.刘正捷译.北京:机械工业出版社，2011

# 2、软件设计约束

### 2.1 软件设计目标和原则

软件设计的目标是，根据软件系统的需求（包括功能性需求和非功能性需求），综合考虑软件开发过程中的各种制约因素（如技术、资源、进度等），遵循软件工程的设计原则（如模块化、信息隐藏、问题分解等），给出软件系统的实现解决方案和蓝图，产生可指导编码实现的设计模型及文档。

软件设计活动还须遵循相关的策略和原则，以指导软件设计人员的行为，并对设计成果提出约束和要求。具体的，这些设计策略和原则描述如下。

(1) 抽象和逐步求精的原则

(2) 模块化与高内聚度、低耦合度的原则

(3) 信息隐藏的原则

(4) 多视点以及关注点分离的原则

(5) 软件重用的原则

(6) 迭代设计的原则

(7) 可追踪性的原则

### 2.2 软件设计的约束和限制

运行环境要求：windows10

开发语言：Java

标准规范：编码规范书

开发工具：IDEA

# 3、软件设计

### 3.1 软件体系结构设计

(1)图3.1.1描述的是易行交易平台的系统架构设计，图中界面层和业务逻辑层的类来自于需求用例模型和用例交互模型；基础服务层的“MySQL通讯协议”是用于后端数据库进行操作访问，“SpringBoot”免去大量的配置文件，提高开发效率，减少代码冗余。

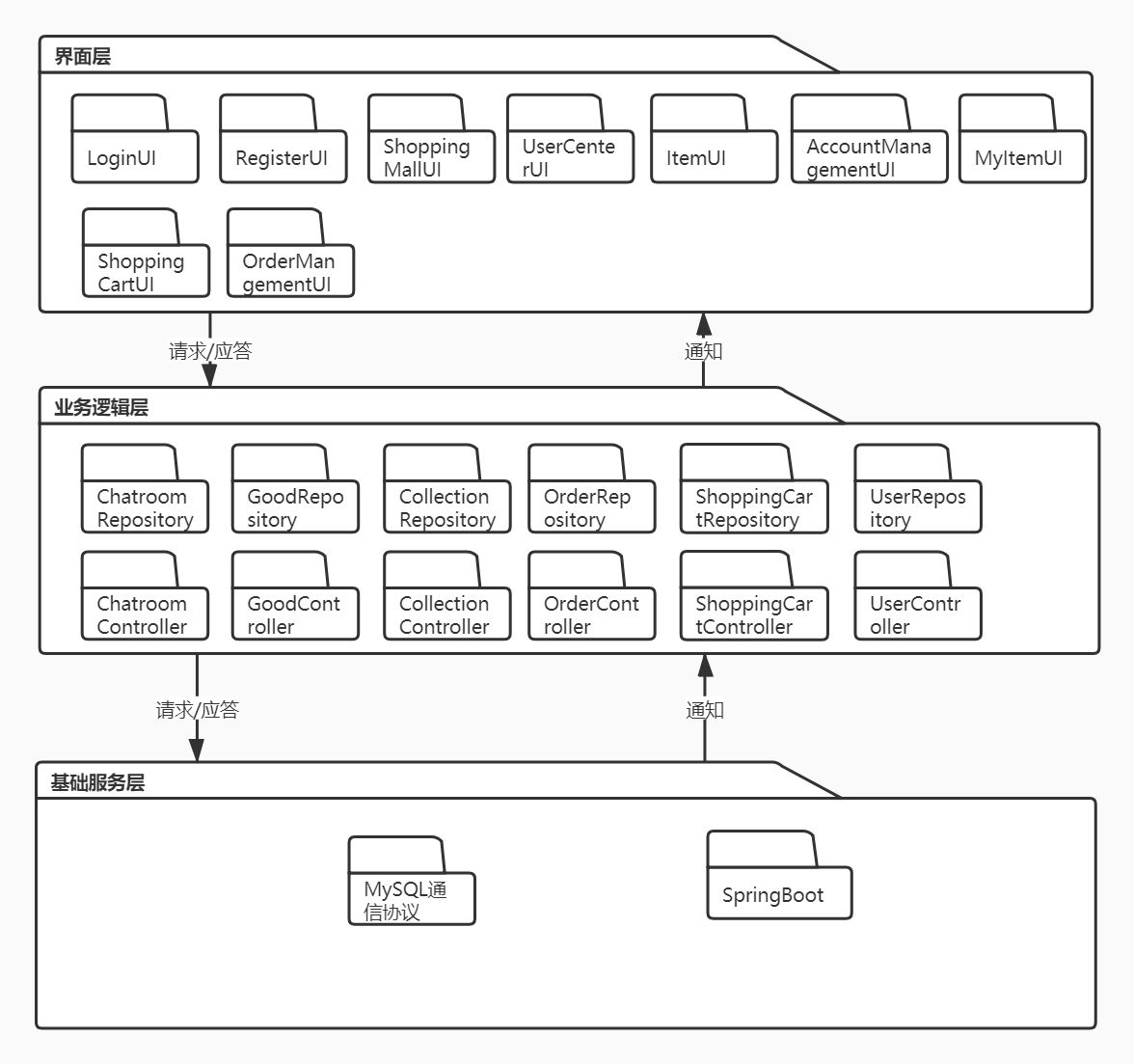


图3.1.1 易行交易平台的系统架构设计

(2)

|  |  |
| --- | --- |
| 模块名称 | 模块介绍 |
| 账号管理模块 | 账号管理模块包括四个基本功能：账号登录、账号登出、账号注册、账号注销 |
| 用户信息管理模块 | 用户信息管理模块包括两个基本功能：修改用户基本信息、绑定学号信息 |
| 买家模块 | 买家模块包括三个基本功能：购买信息、评价、交流 |
| 卖家模块 | 卖家模块包括两个基本功能：发布商品、交流 |
| 订单模块 | 订单模块包括三个基本功能：愿望单、退货、确认收货 |
| 管理员模块 | 管理员模块包括两个基本功能：商品审核、用户审核 |

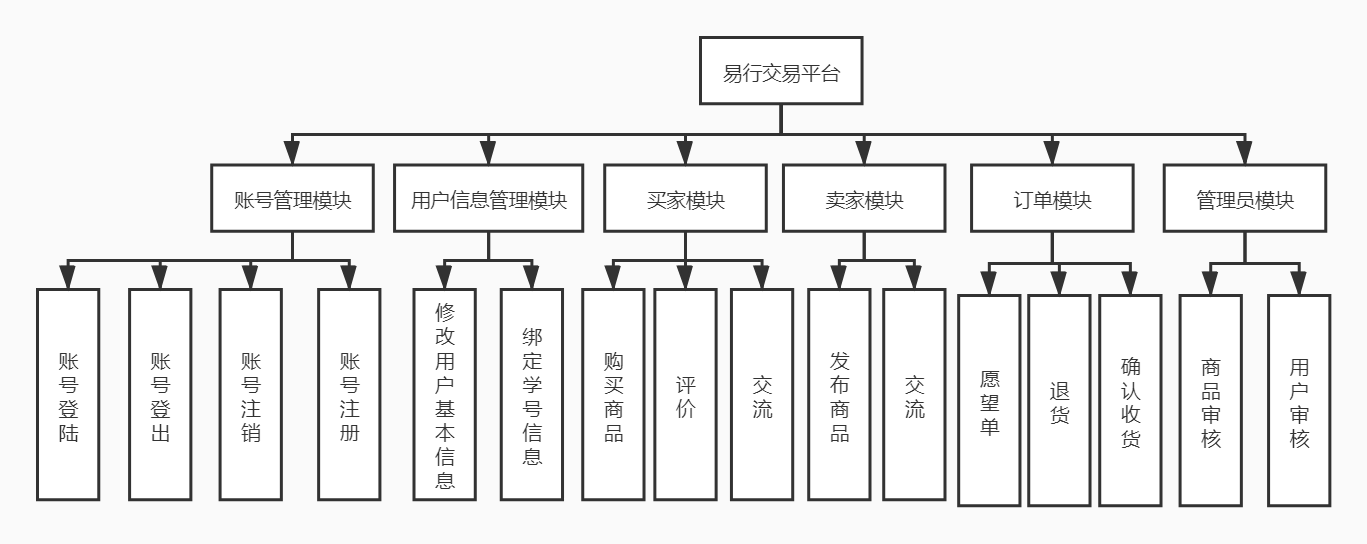
******

图3.1.2易行交易平台的功能模块图

### 3.2 用户界面设计

登陆界面“LoginUI”，用户进入系统的首个页面，用户可以输入密码和账号登陆。

注册界面“RegisterUI”，若用户无账号，需到注册界面输入学号和密码等信息完成注册，注册完成后返回登录界面登录。

商城界面“ShoppingMallUI”，用户登录完成后的首个页面，负责展示在销产品。

商品界面“itemUI”，显示商品的详细信息，在商城界面用户选中商品后跳转。可进行购买、查看等操作，操作完成后返回商城界面。

用户中心页面“UserCenterUI”，负责管理所有与用户有关的信息，可由此页面跳转至账号管理页面、个人发布商品页面、购物车页面、订单管理页面。

交流页面“CommunicationUI”，负责用户和商家沟通交流。

账号管理页面“AccountManagementUI”，用户可由此页面更改账户的详细信息。

个人商品界面“MyItemUI”,显示用户发布的商品列表和信息。

购物车界面“ShoppingCartUI”,显示用户所有的下单列表，可在此界面进行下单购买或撤销下单。

订单管理页面“OrderManagementUI”，用户查看所有订单信息。

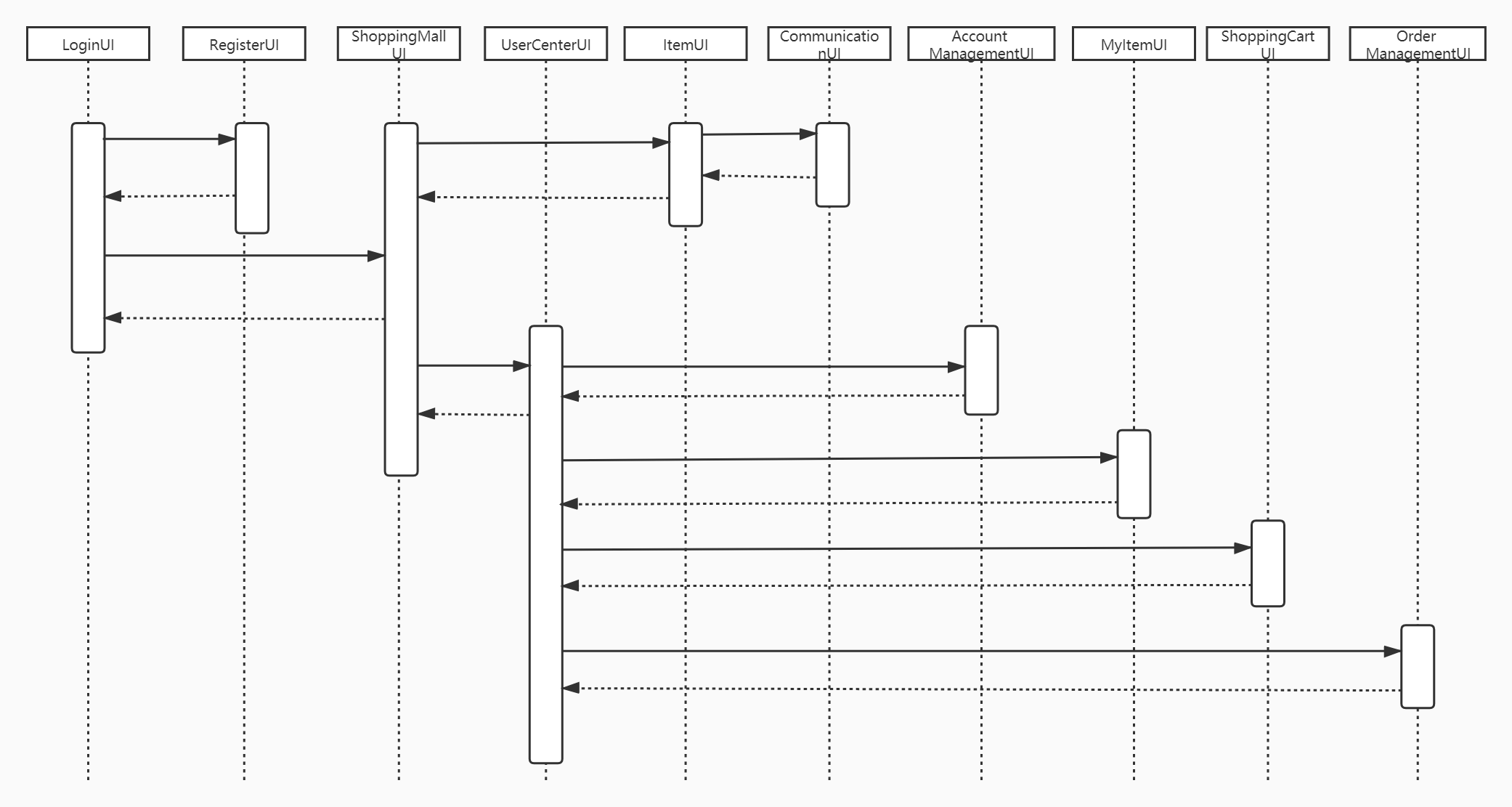
******

图3.2.1 易行交易平台界面跳转关系的时序图

(1)本界面为易行交易平台登录界面

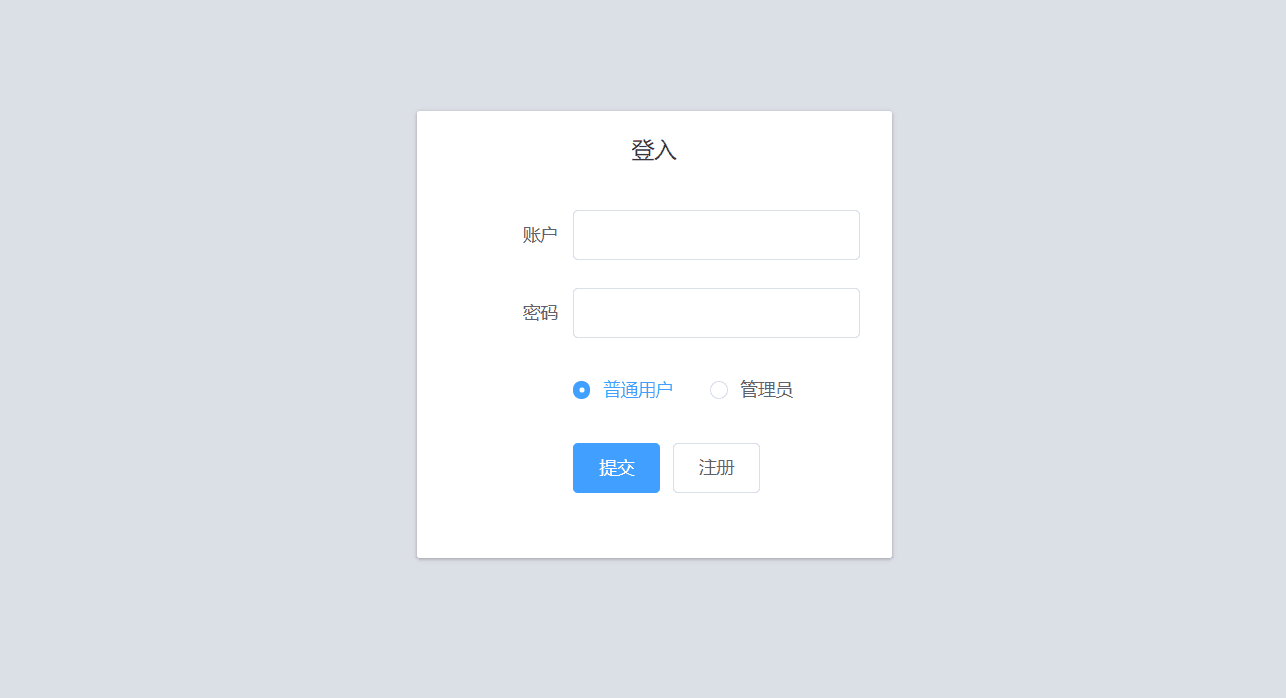


图3.2.2易行交易平台登陆页面

(2)本界面为易行交易平台用户中心界面

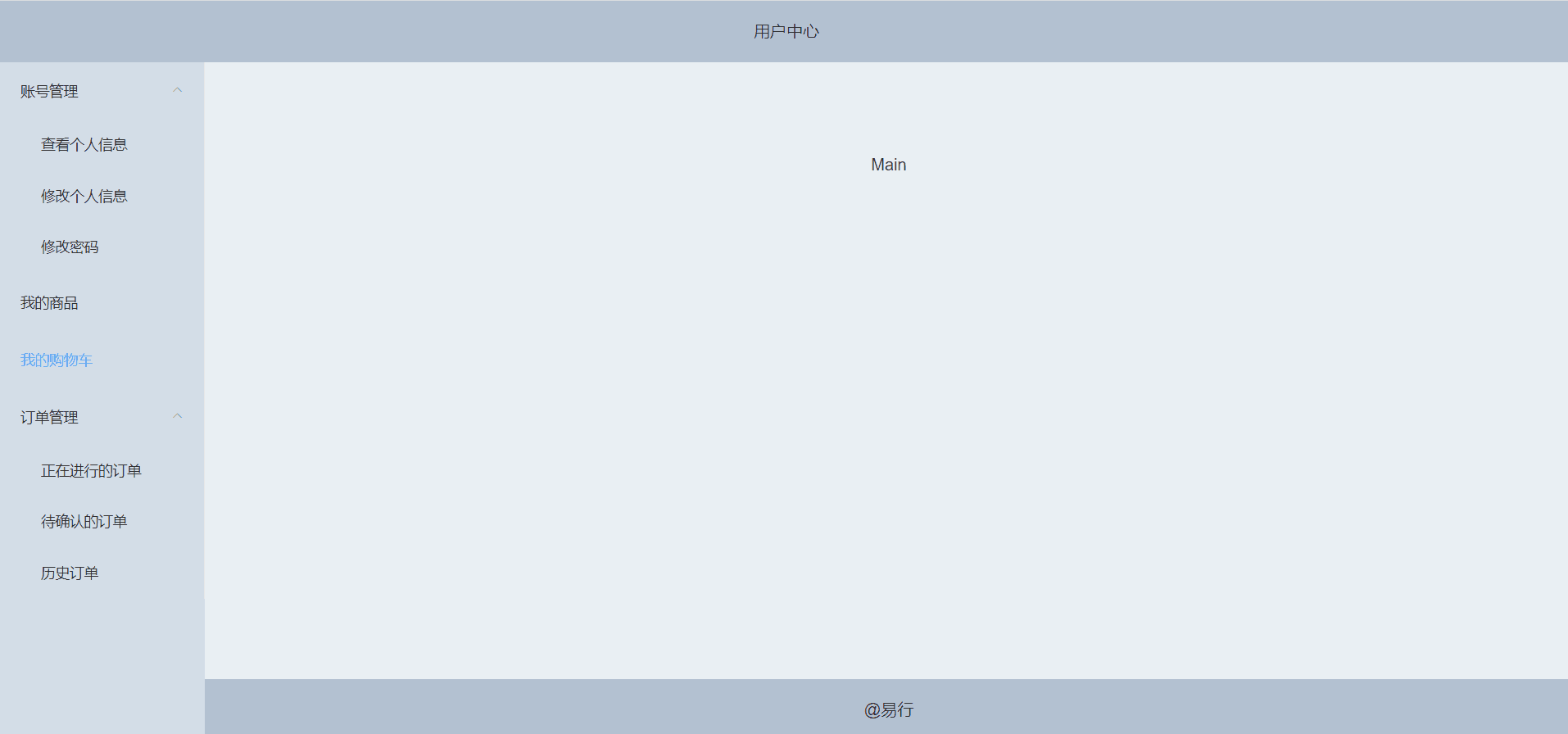


图3.2.3易行交易平台用户中心页面

### 3.3 用例设计

(1)“用户登录”用例实现的设计方案

用户通过边界类“LoginUI”对象输入登录的账户和密码后，向控制类“LoginManager”对象发送“login(acc, psd)”，请求登录。“LoginManager”对象接收到消息后，向实体类“UserLibrary”对象发消息“query”以查证用户提交的账号和密码是否在数据库中，若成功查询则进入商城页面。

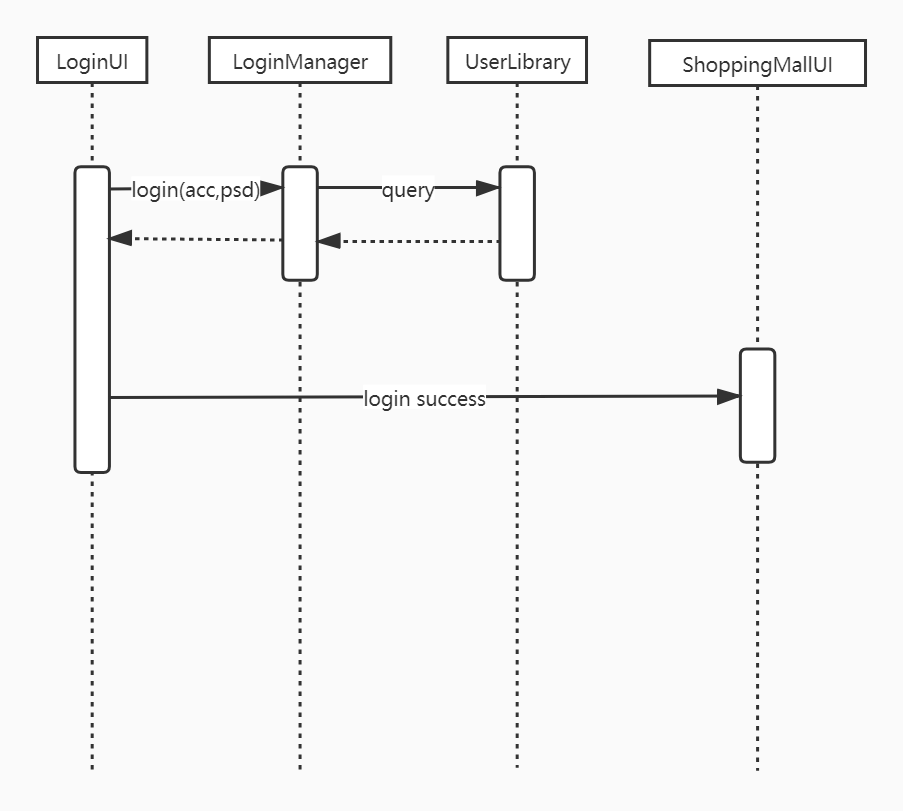
******

图3.3.1“用户登录”用例实现的设计方案

(2)“用户注册”用例实现的设计方案

用户通过边界类“LoginUI”对象跳转至“RegisterUI”对象，“RegisterUI”对象向控制类“RegisterManager”对象发送“login(acc, psd)”，请求注册。“RegisterManager”对象接收到消息后，向实体类“UserLibrary”对象发送消息“insert”将用户提交的信息插入数据库，若成功则进入登录页面。

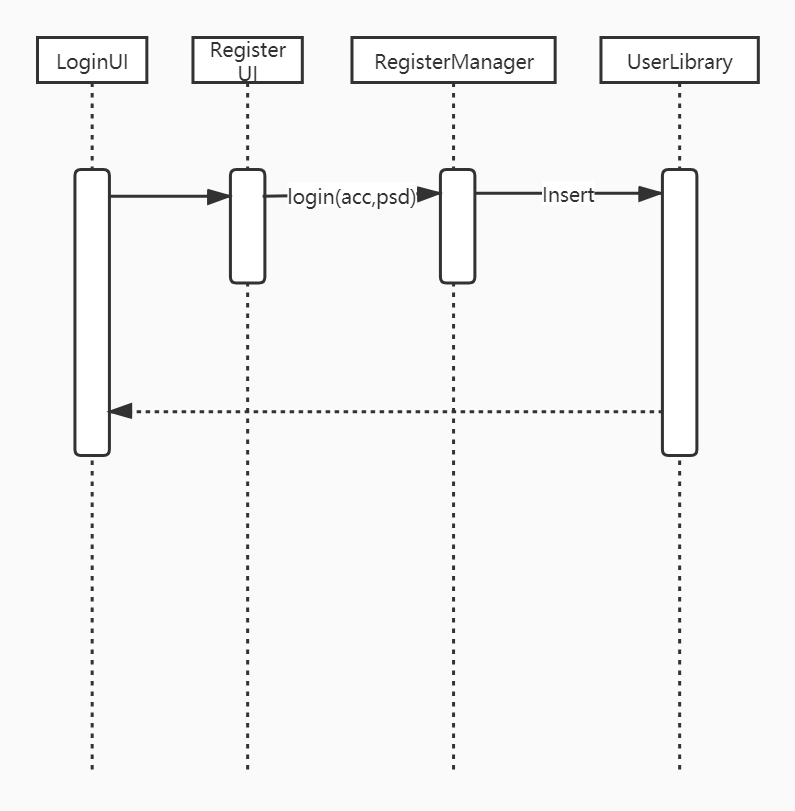
******

图3.3.2“用户注册”用例实现的设计方案

(3)“用户信息修改”用例实现的设计方案

用户通过边界类“AccountManagementUI”修改后的用户信息，向控制类“AccountManager”对象发送“user(information)”，请求登录。“AccountManager”对象接收到消息后，向实体类“UserinformationLibrary”对象发消息“alter”以修改用户信息，若成功修改则进入商城页面。

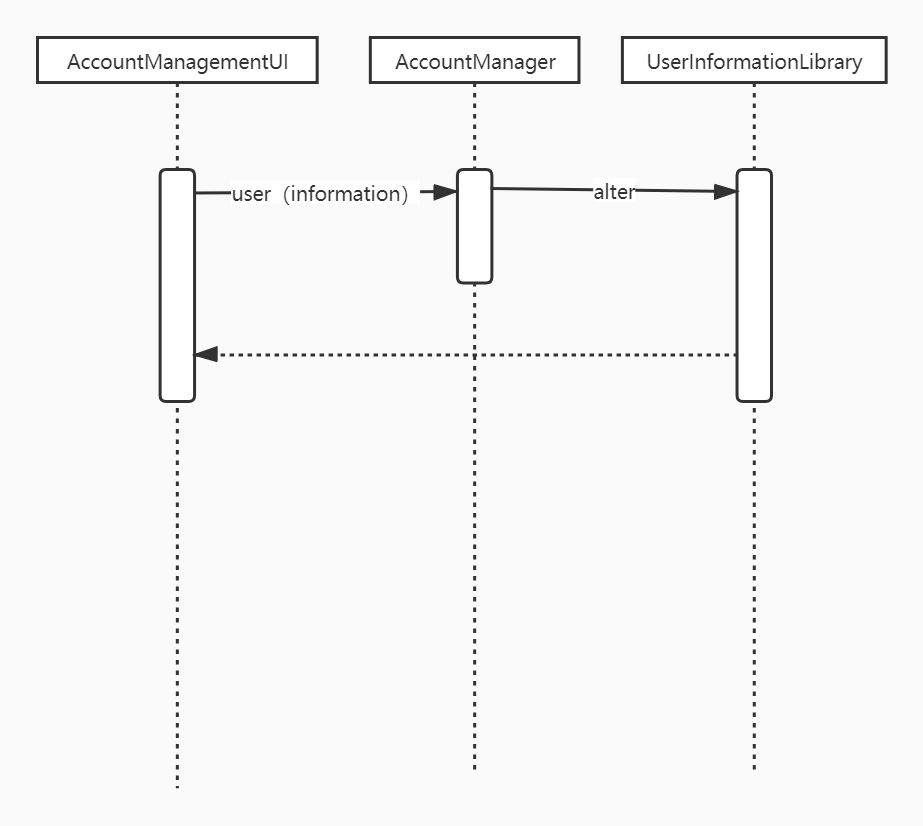
******

图3.3.3“用户信息修改”用例实现的设计方案

### 3.4 类设计

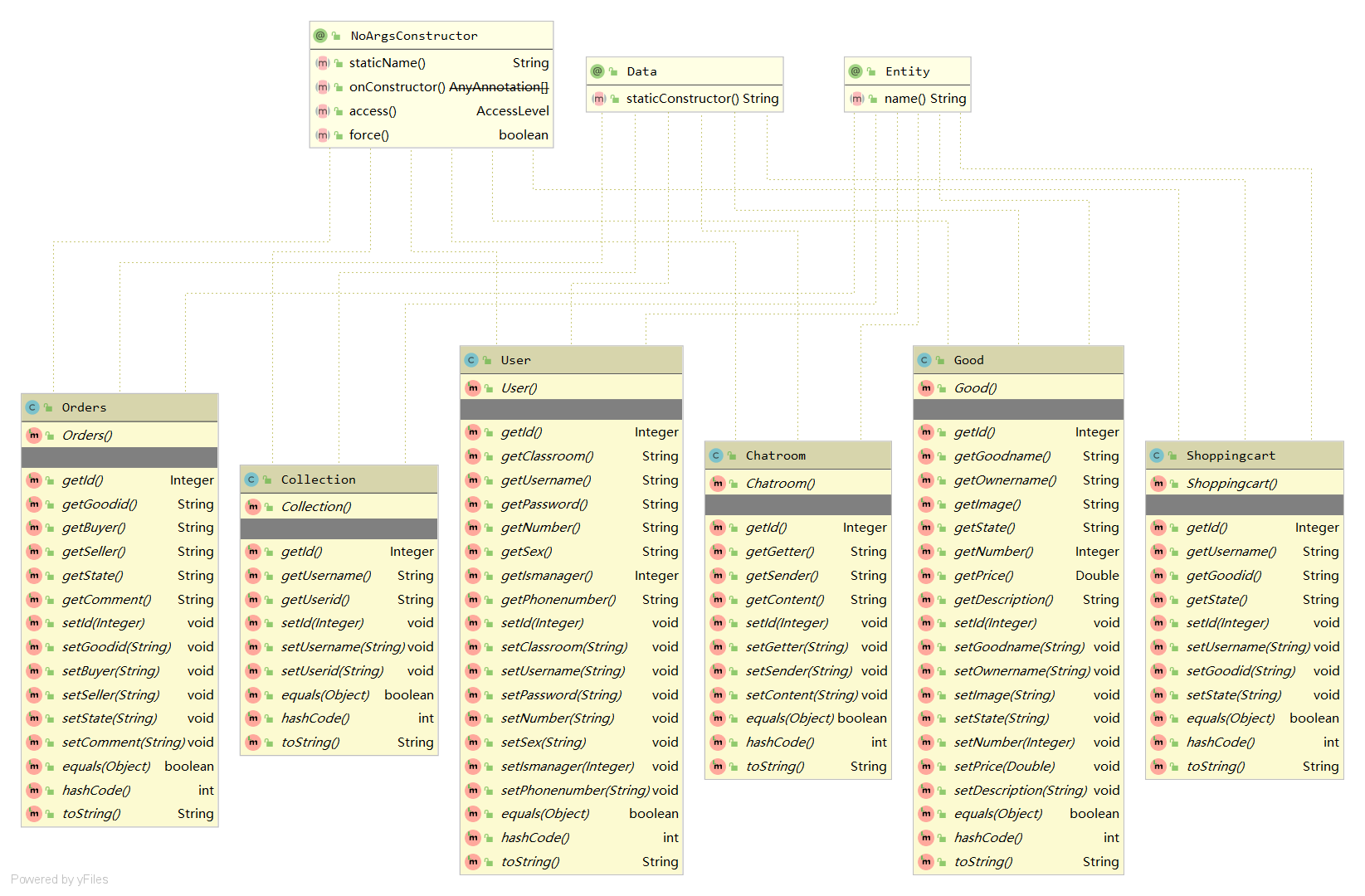
****

图3.4.1易行交易平台实体类类图设计

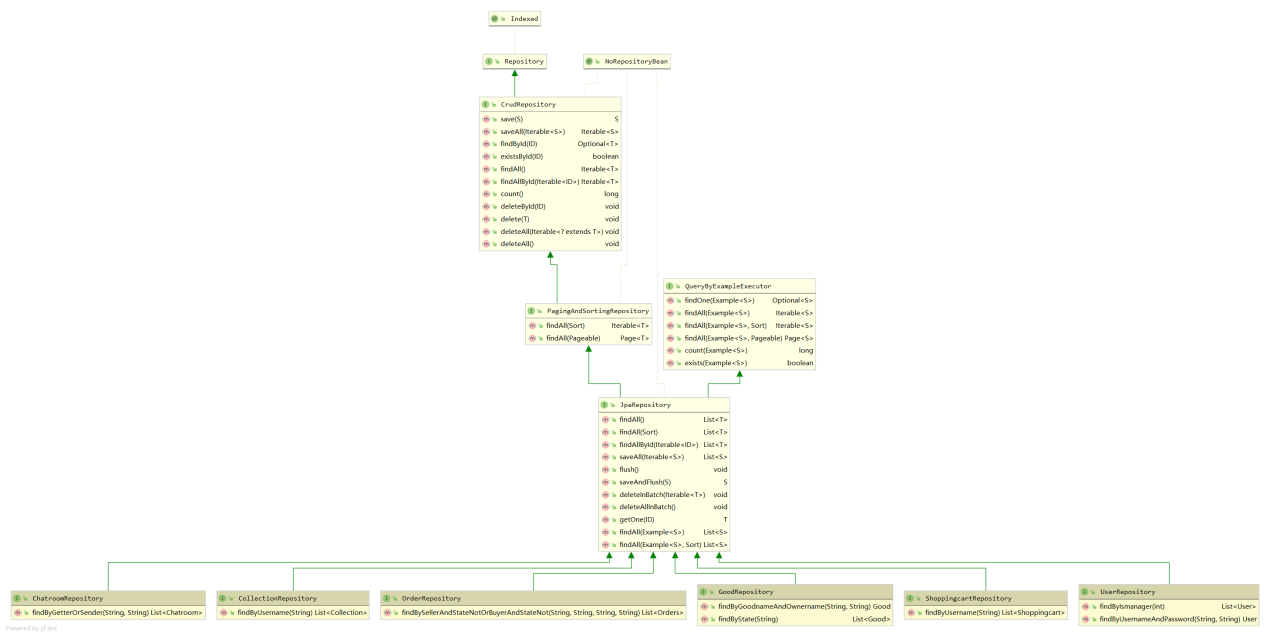


图3.4.2易行交易平台数据库操作类图设计

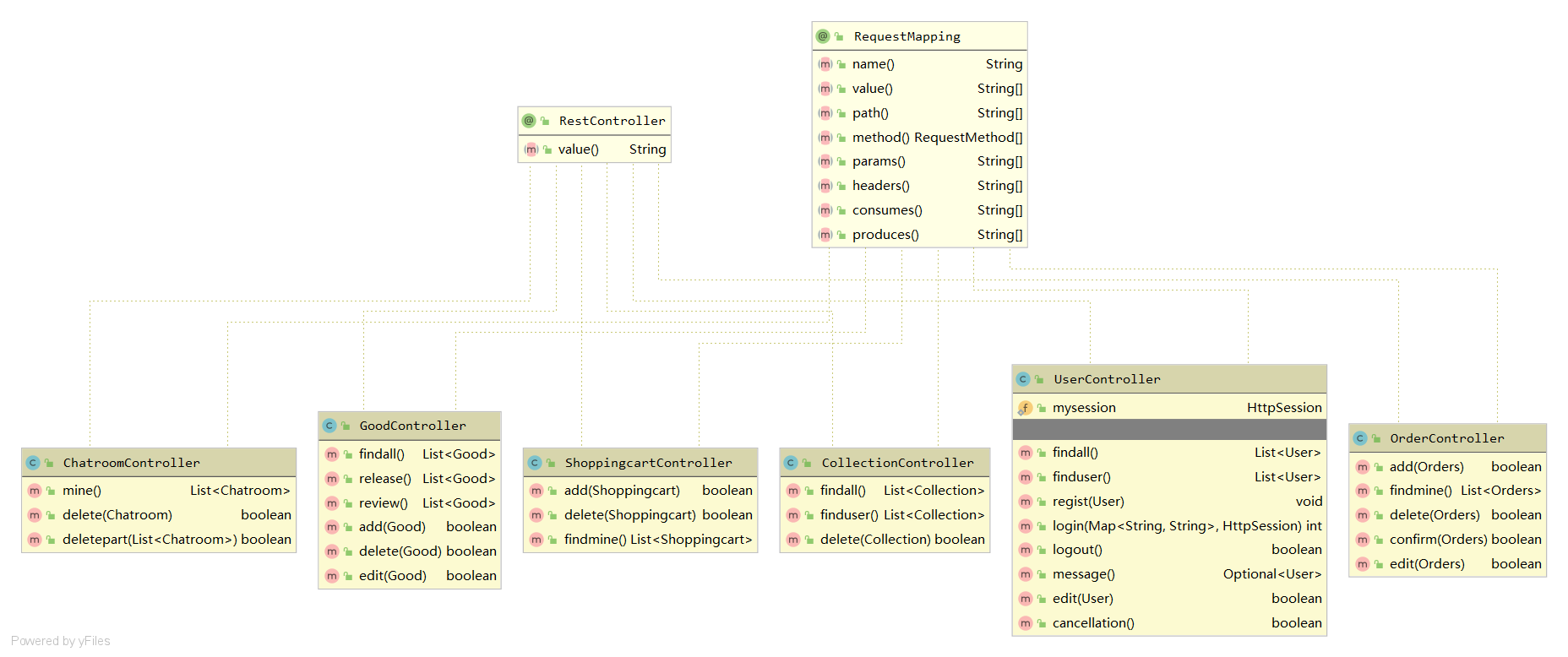


图3.4.3易行交易平台控制层类图设计

entity包：实体类存储各个实体的相关属性与数据库属性以及相关数据类型相对应。

repository包：数据库的操作类，继承JpaRepository接口，在里面可以自定义查询等相关操作。

service包：封装相关的前端请求对应的操作，调用数据库等代码的实现。

controller包：根据对应的url地址拦截前端的地请求，对相应的请求调用相应的service包的相应的类中的方法。

下面以商品相关类为例进行介绍：

商品实体类：

|  |  |
| --- | --- |
| 方法名 | 说明 |
| get/set | 对商品对象属性进行设置和查询 |
| good() | 无参构造方法 |

GoodRepository(商品数据库操作)类：

|  |  |
| --- | --- |
| 方法名 | 说明 |
| findByGoodnameAndOwnername() | 根据商品名和物主名进行查找商品 |
| findByState() | 根据状态返回商品列表 |

GoodController(商品控制)类:

|  |  |
| --- | --- |
| 方法名 | 说明 |
| findall() | 返回所有商品列表 |
| release() | 查找发布的商品 |
| review() | 查找审核商品 |
| add() | 添加商品 |
| delete() | 删除商品 |
| edit() | 编辑商品 |

### 3.5 数据设计

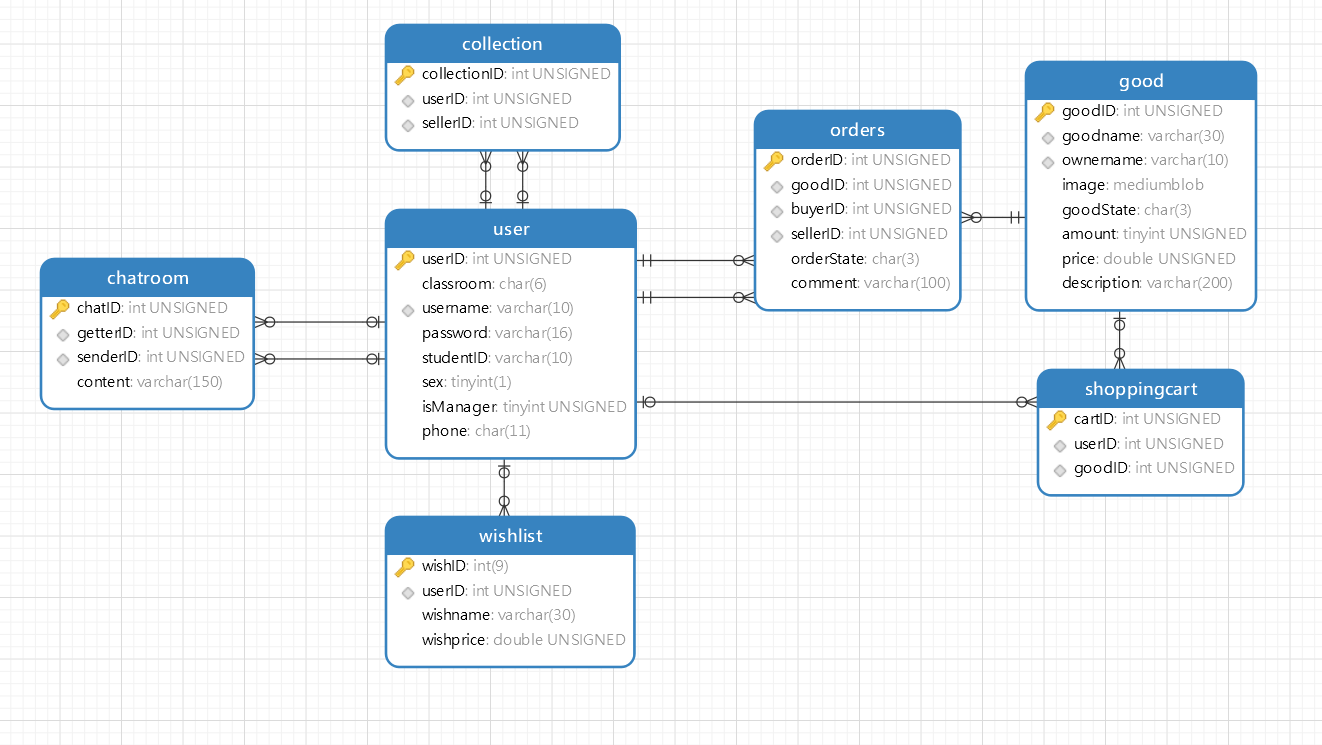


图3.5.1易行交易平台数据库设计

用户表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | java数据类型 | 数据库数据类型 | 长度 | 主键/外键 | 注释 |
| userID | int | int | 9 | 主键 | 用户id |
| classroom | string | char | 6 |  | 班级(例如：软工1702) |
| username | string | varchar | 10 |  | 用户名 |
| password | string | varchar | 16 |  | 密码 |
| studentID | string | char | 10 |  | 学号 |
| sex | int | tinyint | 1 |  | 性别 |
| is\_manager | int | tinyint | 1 |  | 是否为管理员(1表示是) |
| phone | string | char | 11 |  | 电话号码 |

商品表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | java数据类型 | 数据库数据类型 | 长度 | 主键/外键 | 注释 |
| goodID | int | int | 9 | 主键 | 商品id |
| goodname | string | varchar | 30 |  | 商品名称 |
| ownername | string | varchar | 10 | 外键 | 物主名 |
| image | byte[] | mediunblob | 1 |  | 图片(16M) |
| good\_state | string | char | 3 |  | 状态：已发布 待审核 已下架 已出售 |
| amount | int | tinyint | 3 |  | 数量 |
| price | double | double | 8 |  | 价格(小数点后两位) |
| description | string | varchar | 200 |  | 商品描述 |

订单表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | java数据类型 | 数据库数据类型 | 长度 | 主键/外键 | 注释 |
| orderID | int | int | 9 | 主键 | 订单id |
| goodID | int | int | 9 | 外键 | 商品id |
| buyerID | int | int | 9 | 外键 | 买家id |
| sellerID | int | int | 9 | 外键 | 卖家id |
| order\_state | string | char | 5 |  | 状态：进行中 待确认 (待确认一方)已取消 已完成 已退货 |
| comment | string | varchar | 100 |  | 评论 |
| is\_show | int | tinyint | 1 |  | 订单取消通知标识 |

购物车表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | java数据类型 | 数据库数据类型 | 长度 | 主键/外键 | 注释 |
| cartID | int | int | 9 | 主键 | 购物车id |
| userID | int | int | 9 | 外键 | 购物车车主id |
| goodID | int | int | 9 | 外键 | 商品id |

聊天记录表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | java数据类型 | 数据库数据类型 | 长度 | 主键/外键 | 注释 |
| chatID | int | int | 9 | 主键 | 聊天id |
| getterID | int | int | 9 | 外键 | 接受者id |
| senderID | int | int | 9 | 外键 | 发送者id |
| content | string | varchar | 150 |  | 消息内容 |
| is\_show | int | tinyint | 1 |  | 是否读取标识 |

愿望单表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | java数据类型 | 数据库数据类型 | 长度 | 主键/外键 | 注释 |
| wishID | int | int | 9 | 主键 | 愿望id |
| userID | int | int | 9 | 外键 | 愿望单用户id |
| wishname | string | varchar | 9 |  | 想要的商品名 |
| wishprice | double | double | 150 |  | 想要的价格 |

收藏表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | java数据类型 | 数据库数据类型 | 长度 | 主键/外键 | 注释 |
| collectionID | int | int | 9 | 主键 | 收藏id |
| buyerID | int | int | 9 | 外键 | 收藏id |
| sellerID | int | int | 9 | 外键 | 被收藏id |

举报表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | java数据类型 | 数据库数据类型 | 长度 | 主键/外键 | 注释 |
| reportID | int | int | 9 | 主键 | 举报id |
| buyerID | int | int | 9 | 外键 | 举报者id |
| sellerID | int | int | 9 | 外键 | 被举报者id |
| description | String | varchar | 300 |  | 举报描述 |

### 3.6 部署设计

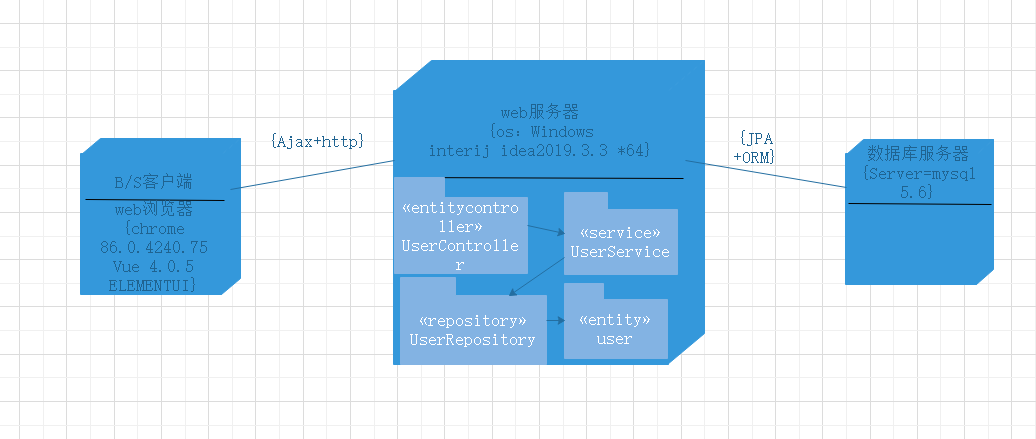
****

图3.6.1易行交易平台部署图

本交易平台使用B/S架构。

web客户端：前端使用vue框架结合element-ui组件搭建前端界面，使用浏览器输入网址即可访问，使用ajax向后端服务器发送请求，从而获取数据。

服务器端：使用springboot框架拦截前端发出的请求，调用相关的controller根据不同的url进行相对应的处理，在后端使用jpa对MySQL数据库进行相关数据的处理，包括增删改查等。