浙江大学

本科实验报告

课程名称： BS 体系软件设计

姓　　名： 黄亦非

学　　院： 计算机科学与技术学院

　　系：

专　　业： 软件工程

学　　号： 3160104367

指导教师： 胡晓军

2019 年 06 月 27 日

一、 概述

1.1 文档说明

本项目的设计文档的阅读者主要为项目开发人员和项目提出者，方便了解项目的设计思路和整体架构。本次设计文档主要是从以下几个方面进行阐述：需求分析、用户类设计、总体设计、接口设计、系统数据设计、主要动作时序图设计，从这几个方面对于本次项目的开发的设计和计划过程做全面的阐述。

1.2 系统需求概述

本次项目“旧书交易网站”是一个辅助旧书在线交易的网站服务。旨在能够通过一个在线的网站，用户可以在线发布和检索相关的旧书，并且能够通过 ISBN 或者是书名链接到更加具体的书目信息等需求功能。更加详细的系统需求如下：

- 实现网站用户的基本登陆、注册功能。对于游客和网站用户两类角色实现权限的控制。注册时提供必要的字段，如用户名、密码、邮箱等，并且需要对邮箱的格式进行验证，并且保证用户名和邮箱在数据库当中是唯一的，也就是用户名和邮箱在设计当中都应当作为主键进行设计。
- 用户登陆之后，可以发布自己需要交易的书籍。在发布书籍的时候需要为该书籍编辑相关的信息，包括书名、原价、出售价、类别和内容、外观照片等信息，并且可以通过 ISBN 和书名链接到外部网站，比如京东或者当当类似的网站，提供更加详细的书目信息，方便买家进行考虑。
- 首页根据用户发布的信息进行聚合，在参考已有的旧书交易网站，这部分的需求应该是实现书目的分类，也就是可以根据类别进入到不同类别书目的查找。方便买家查询的时候能够根据书目类别进行查询和选购。并且支持多条件的排序，分页等功能。
- 用户可以自行设置交易的方式是寄送或者是线下进行交易，然后在实际交易的时候需要生成电子的订单，并且保存在数据库当中。然后可以根据交易的方式生成不同的订单，比如邮寄的订单就需要邮寄地址，邮政编码等必要的信息。
- 集成一个消息系统，消息系统对于一个交易网站还是非常有必要的。买家和卖家可以在消息系统当中进行沟通，最后达成一致完成交易。进阶功能可以考虑群组功能的加入，方便一类用户进行在线的交易，并且最好能实现实时通知的效果。

- 集成一个求购的模块，在求购的模块当中，用户可以发布自己想要的书籍，如果卖家刚好契合，则两者可以达成一致，完成交易。
- 支持书籍的搜索功能，能够通过书籍名称、ISBN、发布者 username 进行搜索。
- 能够实现购物车功能，并且卖家能够实时的修改订单状态，确认订单等。
- 用户能够修改个人的简历、修改密码，其他用户能够查看自己发布的旧书或者是求购的书籍，以及个人信息。
- 界面的样式需要同时适配 PC 端和手机端的浏览器页面显示。
- **增强需求：**完成 Andriod 或者 iphone 的手机客户端，基本功能需求如下，增加额外的需求。如支持当前的定位功能，在发布书籍信息的时候能够带上当前发布的位置，方便进行线下的交易，并且可以根据用户当前的位置去匹配附近的代卖书籍。
- 客户端软件的消息系统和订单的生成支持随时的进行推送的功能。

1.3 系统设计原则

- **整体性原则：**整个系统的开发设计都应该要在项目设计人员的能力范围之内，并且需要从系统的总体设计出发，同时也要考虑到实际的一些限制性因素等，囊括在设计开发的原则当中。
- **阶段性原则：** 项目的实际开发并不是一蹴而就的过程，而应该是一个阶段性的过程开发。设置阶段性的任务和完成质量、时间等实际性能考察工作量的指标。保证整个开发过程当中的稳定性和高可靠性。
- **客观可行原则：**整个开发计划当中必须适应现实情况当中的约束，在开发过程当中应该要结合实际，采取高效可靠的开发模式。
- **全面性原则：**从不同的角度和不同的层次来进行整个计划的设计，这样才能从多方面考虑到项目开发的设计。促进开发过程当中的，相关方面的协调和配合。

1.4 系统开发工作阶段时间

系统规划阶段	标志性事件	时间
需求分析阶段	软件需求分析和获取完成	2019.4
设计阶段:	系统设计报告书完成	2019.5
编码实现	项目基本实现	2019.6 中旬
测试阶段	测试计划	2019.6.15 -2019.7
移交阶段	项目提交	2019.7.1 之前
维护阶段	系统正常运行、定期维护	2019.7 提交后

二、 系统设计

2.1 用户类及其特征

用户类	特征与说明
游客	1. 次要的用户并且用户的权限比较低，需要的功能也是最少的。 2. 为了方便游客进入网站，提供比较简明的操作页面和注册引导。 3. 除去简单的浏览权限，其余的相关功能都需要登陆之后才能继续操作。
网站用户	1. 项目当中的主要角色，并且该项目的主要开发围绕着该角色展开。 2. 面向的用户数量可能很多，需要考虑在特定的时间之内的在线人数可能会很多，做好负载均衡。 3. 要求能够在线的发布需要交易的书籍信息，并且可以选择相对应的交易方

	式。 4. 可以进行书籍的分类查询。 5. 集成消息系统，在项目当中进行实时的交流功能。 6. 求购板块的实现，对应的用户能够及时的收到求购板块的回复。 7. 能够实现基本的用户操作界面和相关功能。

三、 总体设计

3.1 框架使用

3.1.1 公有框架

- **MySQL：**MySQL 是一款开源的关系型数据库管理软件，由于其性能高，成本低，可靠性强等特性，已经逐渐发展为比较受欢迎的开源数据库。为多种语言都提供了 API，并且支持多线程，优化了 SQL 查询算法，在中小型的网站开发当中是比较好的一种选择。所以在这次实现的项目当中选取的就是 MySQL 作为关系型数据库。
- **Docker：**Docker 是一个开源的软件项目，在 Linux 操作系统之上定义了一个软件抽象层，利用 Linux 核心当中的资源分离机制，例如 cgroups，来创建独立的 containers。Docker 非常方便我们打包自己的服务，并且快速的在服务器上进行部署，同时也很方便的获取其他公有服务的支持，在本项目当中使用的就是 docker 进行部署，并且使用了 redis 服务作为 django-channels 的消息缓存。
- **docker-compose:** docker-compose 项目是 Docker 官方的一个开源项目，负责实现对于容器集群的快速编排。实现一个 web 项目的时候，除了 web 服务器本身之外，还需要其他的外部服务来支持，docker-compose 提供了一种机制来串联这些 docker containers，它允许用户通过一个 docker-compose.yml 模板文件来定义一组相关联的应用容器。在项目当中我使用的就是 docker-compose 来进行容器的关联和管理，可以非常方便的将网站部署上线。

3.1.2 前端框架

- **Bootstrap：**本次前端采用的是 Bootstrap 模板来进行开发，Bootstrap 是一个开源的前端项目，并且提供了非常全面的文档，样式表基于 Less 和 Sass 开发，并且使用 Bootstrap 模板开发可以有效的适配手机、平板、PC 等设备，经过测试我的项目是可以适配这三种移动端的，它非常适用于快速开发响应式布局、移动设备优先的 web 项目。另外搭配 Django 的模板语言机制，使得能够实现一定程度的前后端分离开发。
- **jQuery：**jQuery 是一款跨浏览器的 javascript 库，极大的简化了 HTML 和 JavaScript 之间的操作。操作文档对象、选择文档对象模型元素、创建动画、ajax 程序等许多方面都能够非常方便的胜任。模块式的方式使得 jquery 能够创建功能完备，样式美观的动态网页和网络应用程序。

3.1.3 后端框架

- **Django：**Django 是一个开源的 web 应用框架，由 python 语言构建。采用的是 MVT 的设计模式，即 model、view、template 来进行设计和开发。Django 的 model 与数据库操作之间采用的是内部完备的 ORM 操作，使得在开发当中不用编写原生的 SQL 语句，有效的避免 SQL 注入攻击。Django 框架注重组建的重用性、敏捷开发的原则，并且继承了 python 语言的强大的灵活性和活跃的生态，在越来越多的第三方 package 的支持下，Django 框架持续的活跃在 web 程序开发当中。
- **Django-channels：**是 Django 的第三方中间件，主要是引入 websocket 的支持，HTTP 请求交给 Django 内置的 wsgi 服务器处理，websocket 请求交给 channels 当中的 asgi 服务器处理。并且该中间件同时支持同步和异步的操作，使用其他非常方便，并且使用 redis 作为消息缓存。
- **Redis：**Redis 是一个开源的，基于网络、内存、可持久化的键值对存储数据库。在本项目当中主要是做 channels 的本地消息缓存。所以在运行项目之前要先将 redis 开起来，后面在《使用手册》当中也会提到。
- **Django-password-reset：**这是 Django 的第三方插件，实现了完整的通过邮箱验证，重置密码等相关操作，并且非常方便的整合到项目当中，只需要在项目当中配置一个打开 smtp 协议支持的邮箱即可。

3.2 运行环境

3.2.1 硬件服务端：

项目	信息
处理器	Intel i5 及以上
内存	DDR3 or DDR4
网卡	支持 ZJUWLAN

3.2.2 硬件客户端：

项目	信息
操作系统	MacOS
服务器软件	Django 内置 server
数据库软件	MySQL 8

3.2.3 软件服务端：

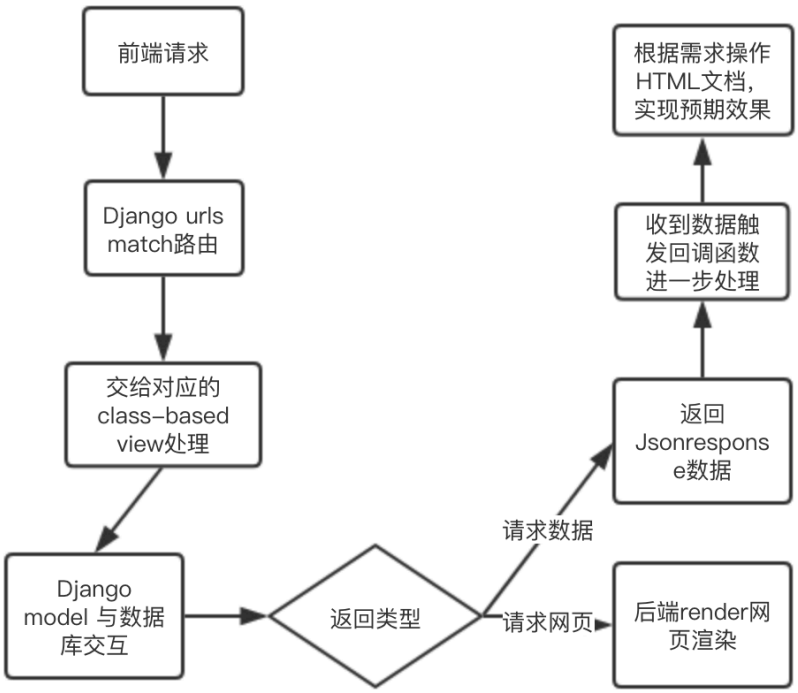
项目	信息
处理器	主流 Intel 或者 AMD 处理器
内存	128M 以上
存储	主流存储介质
网卡	支持 ZJUWLAN 即可

3.2.4 软件客户端：

项目	信息
操作系统	主流即可 Windows、Linux、MacOS
浏览器	Chrome、firefox 等主流浏览器

3.3项目结构

- BookTradeWeb：项目主目录
  - books：处理项目主要逻辑，包含内部的所有模板文件和 js 文件，和后端的逻辑处理。
  - BookTradeWeb：项目初始化部分，做的是路由分配和项目的配置 setting
  - chatting: 聊天模块，使用 Django-channels 做 websocket 请求处理。
  - media：资源文件夹，用户上传的头像、书籍照片等都会统一存放在这里，方便后端定义图片的路由。
  - password_reset: 密码找回模块，引入 Django-password-reset 插件
  - scripts: 项目部署的相关脚本文件
  - useraction: 处理用户相关模块，管理用户登录、注册等逻辑
  - docs: 项目相关文档
  - 其他: 相关配置文件、依赖配置等。
- **项目响应流程：**



四、 关键数据库设计

4.1 具体设计

表名	功能说明
user	网站用户
book	发布的书籍
bookoffer	订单
chattingmessage	消息
book_need	求购书籍
comment	评论
shopping_car	购物车
creditaccount	交易账户

4.2 表 user（网站用户）

user 表是网站的用户，其中包括以下的属性：

- id: 用户的 id
- password：密码，在插入数据库时 Django 自动进行加盐加密
- username：用户名
- email：用户邮箱
- telephone: 手机号
- introduction：个人介绍
- address：收货地址
- header_image: 用户的个人头像，存储的是 url

其中用户名和邮箱在系统当中都是唯一的，所以会对这两个值进行检查，如果重复要要求用户重新进行注册。

4.3 表 book

book 表是系统当中的书籍数据库表，其中包括以下的属性：

- id: 书籍的 id，唯一
- book_name: 书籍的名字
- origin_price: 原价
- sell_price: 现在出售的价格
- category: 出售书籍的类别
- book_introduction: 书籍的相关描述
- book_image: 书籍图片的 url
- ISBN: 书籍的 ISBN 号
- book_url: 链接到当当网上的具体链接
- publisher_name_id: 关联到用户表，保存发布者的 id
- store_remain_num: 库存数量
- trade_way: 指代交易方式，online 为在线交易，offline 为线下交易
- publish_time: 书籍发布的时间
- book_status: 书籍的 status，其中 0 代表的是出售的书籍，1 代表的是求购的书籍

4.4 表 bookoffer

bookoffer 是系统当中的订单的数据库表，具体的属性如下所示：

- id: 订单号
- sell_option: 指代是在线交易还是线下交易
- post_address: 交易的地址
- post_code: 邮寄时的邮寄地址
- status: 订单状态，分为待确认和已经确认两种状态
- complete_time: 订单完成的时间
- book_id: 订单对应的书籍的 id
- buy_side_id: 买方的 user id
- sell_side_id: 卖方的 user id
- complete_book_num: 成交的书籍数目

- complete_price: 完成交易时的交易总价
- contact_phone: 买家的联系电话

4.5 表 chattingmessage

chattingmessage 这张表主要记录的是系统当中不同用户进行通信的时候的消息记录，主要是方便在下次重新聊天的时候能够加载历史的聊天记录。

- id: 消息记录的 id
- message: 相关的消息内容
- create_time: 消息生成的时间
- recv_side_id: 接收消息的 user 对应的 user id
- send_side_id: 发送消息方的 user 对应的 user id

4.6 表 book_need

book_need 这张表主要记录的是用户的求购的书籍的相关信息，这张表的属性主要如下所示：

- id: 求购单对应的 id，是唯一的属性
- message: 对应 user 在发布求购的时候加上的自己想展示的 message
- book_id: 链接到 book 表当中的具体的书的 id

4.7 表 comment

comment 这张表主要是存放的对于每一本书的评论信息，主要属性如下所示：

- id: comment 的 id 号
- time: 对应的评论的时间
- content: 评论的相关内容
- book_id: 对应评论的书的 id
- commenter_id: 评论者 user 对应的 id 号
- score: 对于书籍的评分，1-5 分

4.8 表 shopping_car

shopping_car 这张表是系统当中的购物车的数据库表, 主要是存放的用户加入购物车当中的书籍的列表：

- id: 购物车当中 item 的 id
- added_number: 加入书籍的数量
- book_id: 加入书籍的对应的 book id
- book_owner_id: 加入书籍的用户的 user id
- address: 期望的成交地址
- contact_phone: 买家的电话号码

4.9 表 creditaccount

表 creditaccount 是用户在系统当中的账号系统，因为并没有接入真实的电子钱包，所以在系统当中模拟的电子账户，主要属性如下：

- id: accout 的 id 号，唯一
- account_money: 用户账户里的余额
- create_time: 账户创建的时间
- account_owner_id: 账户对应 user 的 id

五、 接口设计

系统当中的接口一部分是返回 json 格式数据，一部分是后端渲染模板，这里将主要的逻辑展示出来，具体的可以参考每个 app 当中的路由设计，其中请求数据部分都是以/api 路由开头，表示返回的 json 格式数据到前端。

- **用户登录**
  - url: /auth/login/
  - method: post
  - data
    - username: 用户名
    - password: 密码

- **用户注册**
  - url: /auth/register/
  - method: post
  - data
    - username: 用户名
    - password: 注册用密码
    - email: 用户邮箱，在前端使用 js 进行格式验证
  - response
    - message: 系统的反馈信息
    - code: 返回的状态码

- **用户注销**
  - url: /auth/logout/
  - method: post

- **用户个人简历页面**
  - url: /books/profile/
  - method: get

- **用户添加交易书籍**
  - url: /books/addlist/
  - method: post
  - data
    - book_name: 书籍名
    - ISBN: ISBN 号
    - book_description: 书籍的相关描述
    - book_category: 书籍的类别
    - origin_price: 书籍原价
    - current_price: 当前的出售价格
    - store_remain_num: 添加的库存数量
    - book_url: 当当网的具体的书籍链接
    - trade_way: 交易的方式，线下 or 线上
  - response
    - message: 返回的系统消息
    - code: 状态码

- **查看用户发布的书籍和相关信息**
  - url: /books/user_books/<user_id>/
  - method: get
  - data: user_id

- **查看在售的书籍详情**
  - url: /books/sell/<book_id>/
  - method: get
  - data: book_id

- **查看正在求购的书籍详情**
  - url: /books/buy/<book_id>/
  - method: get
  - data: book_id

- **展示具体类别的所有书籍:**
  - url: /books/show_books/<category>/
  - method: get
  - data: category，其中 all 代表所有的在售的旧书

- **用户所有购买和销售订单详情**
  - url: /books/shopping_car/
  - method: get

- **求购市场**
  - url: /books/book_need/
  - method: get

- **搜索引擎搜索书籍**
  - url: /books/search/
  - method: get
  - data: content，搜索的具体信息，包括用户名、ISBN、书名等一起搜索

- **用户修改密码**
  - url: /books/api/update_password
  - method: post
  - data
    - old-password: 原来密码
    - new-password: 新的密码

- **用户更新个人信息**
  - url: /books/api/update_profile/
  - method: post
  - data
    - phone: 手机号
    - address: 寄送地址
    - introduction: 个人简介

- **根据书名爬取当当网信息**
  - url: /books/api/search_link_ISBN/<book_name>/
  - method: post
  - data: book_name: 需要搜索的书名
  - response
    - link: 对应的当当网链接
    - description: 书籍描述
    - ISBN: 书籍的 ISBN 号
    - sell_price: 在售的价格

- **提交书籍评论**
  - url: /books/api/submit_comment/<book_id>/
  - method: post
  - data
    - comment_score: 评论的得分
    - comment_review: 评论的内容

- **加入购物车**
  - url: /books/api/add_to_shopping_car/
  - method: post
  - data
    - book_id: 添加的书籍的 id

- number: 添加的书籍的数量
- address: 交易的地址
- phone: 买家的联系电话

- **账户余额充值**
  - url: /books/api/add_credit_account/
  - method: post
  - data: number: 充值的金额

- **用户删除发布的旧书**
  - url: /books/api/delete_publish_book/<book_id>/
  - method: post
  - data: book_id: 需要删除的发布旧书的 id

- **用户修改发布旧书的信息**
  - url: /books/api/modify_book_info/
  - method: post
  - data
    - num: 修改后的库存数量
    - price: 修改后的当前售价
    - book_id: 需要修改的旧书的 book id

- **用户确认销售订单**
  - url: /books/api/verify_order/
  - method: post
  - data
    - order_id: 订单的订单号

- **搜索用户的个人信息**
  - url: /chatting/api/search_user_info/<user_id>/
  - method: post
  - data: user_id
  - response
    - username: 用户名
    - image_url: 用户头像的 url

- **展示聊天室界面**
  - url: /chatting/<room_name>/
  - method: get
  - data: room_name 房间名称

六、 出错设计

6.1 出错信息

系统当中可能出现的出错信息主要如下：

错误类型	含义	处理方法
输入非法	提交数据不符合逻辑要求，如必要的数据提交为空	显示错误，并且返回前一界面，等待用户重新提交
数据错误	提交的信息与数据库当中的信息矛盾，不符合数据库的要求	显示相对应的错误，然后等待用户重新提交数据
操作错误	进行操作非法，比如未登录的用户试图下单	显示操作失误，回到正确的页面
系统错误	网络故障、服务器故障	联系管理员进行维护

七、 主要功能时序图设计

7.1 登录

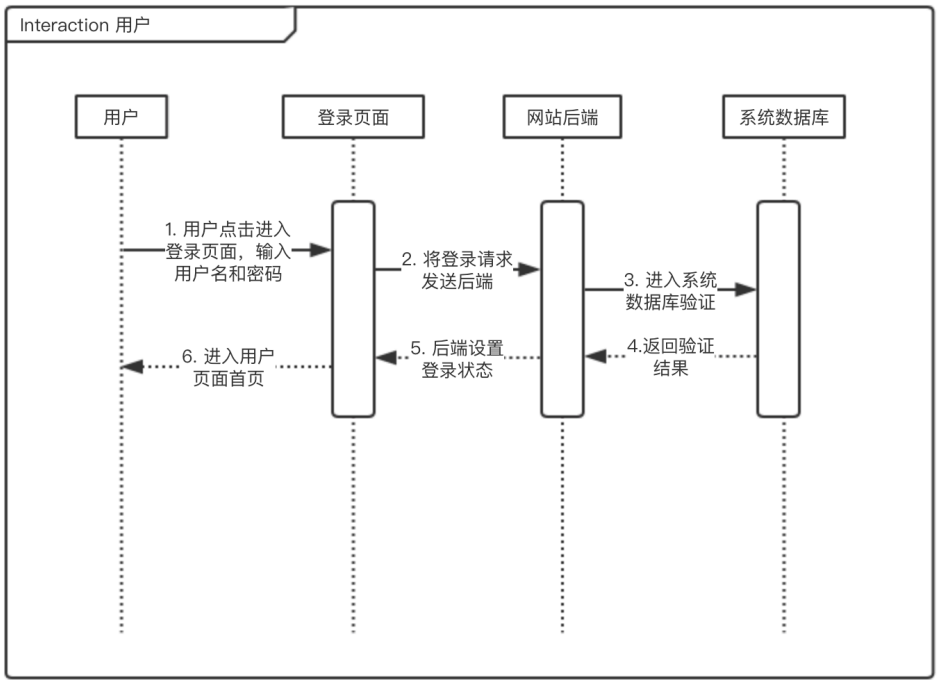


图 1:用户登录时序图

7.2 发布交易书籍

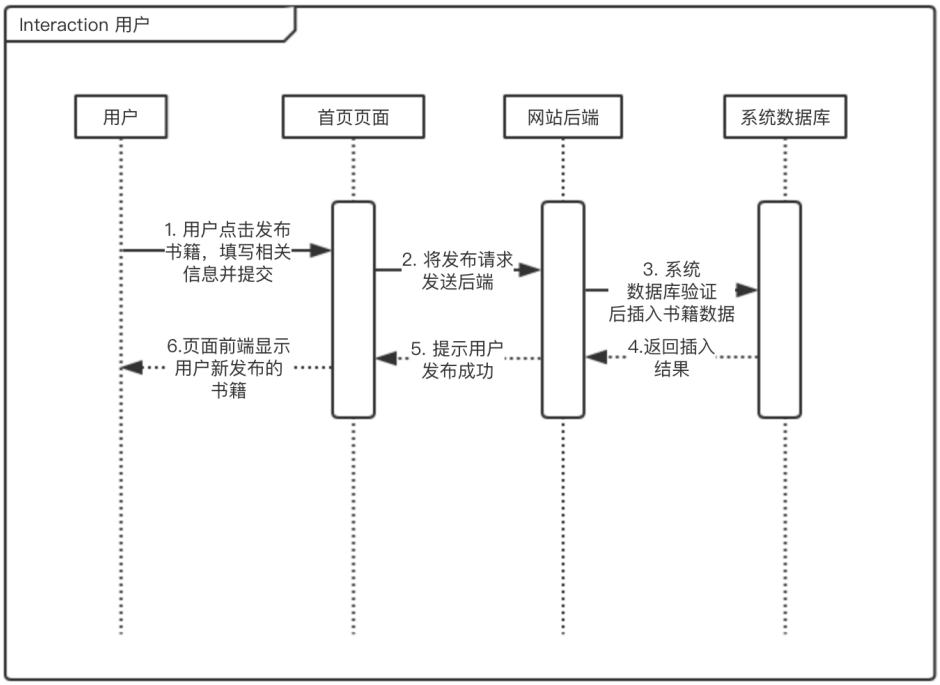


图 2:用户发布书籍时序图

7.3 书籍检索

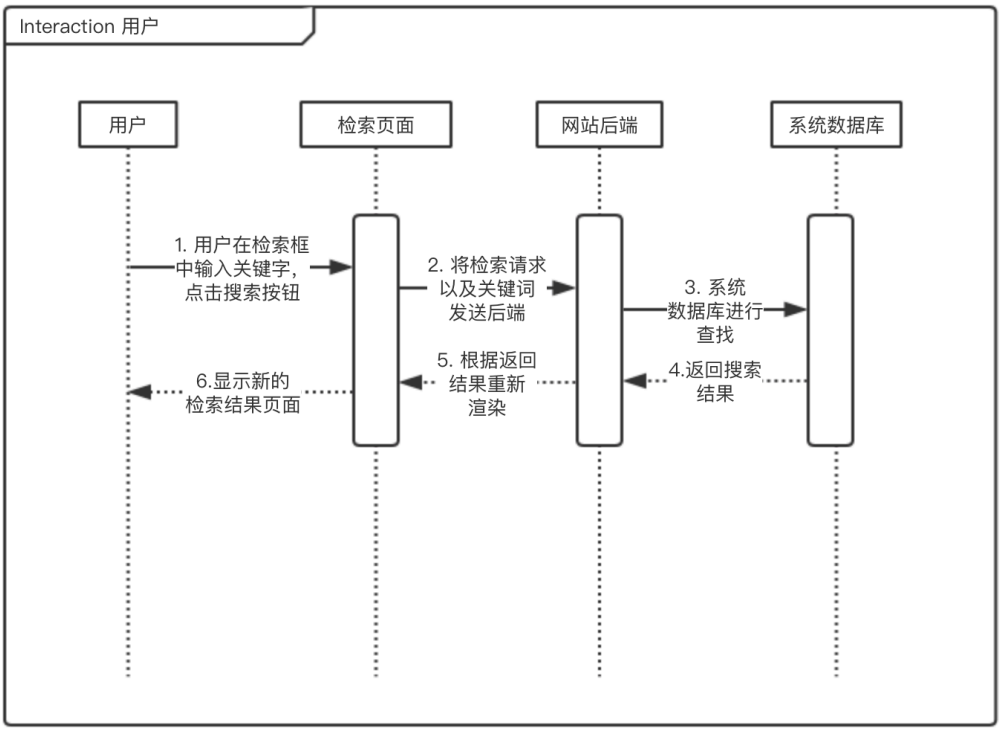


图 3:用户书籍检索时序图

7.4 用户交易书籍

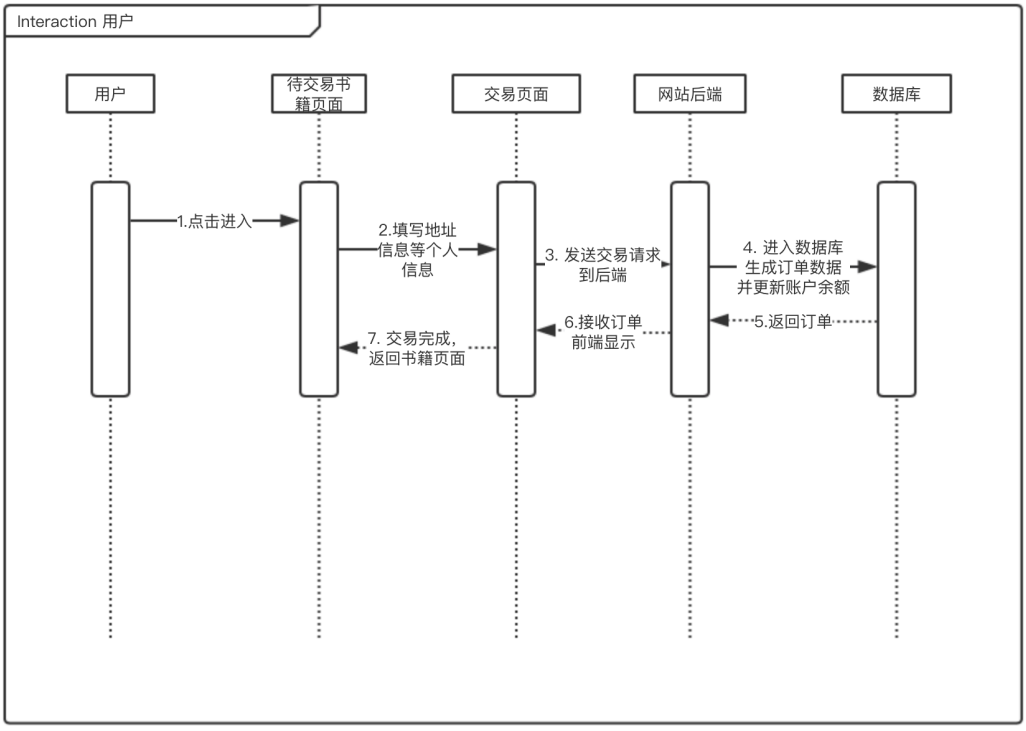


图 4:用户交易书籍时序图

7.5 用户发布求购信息

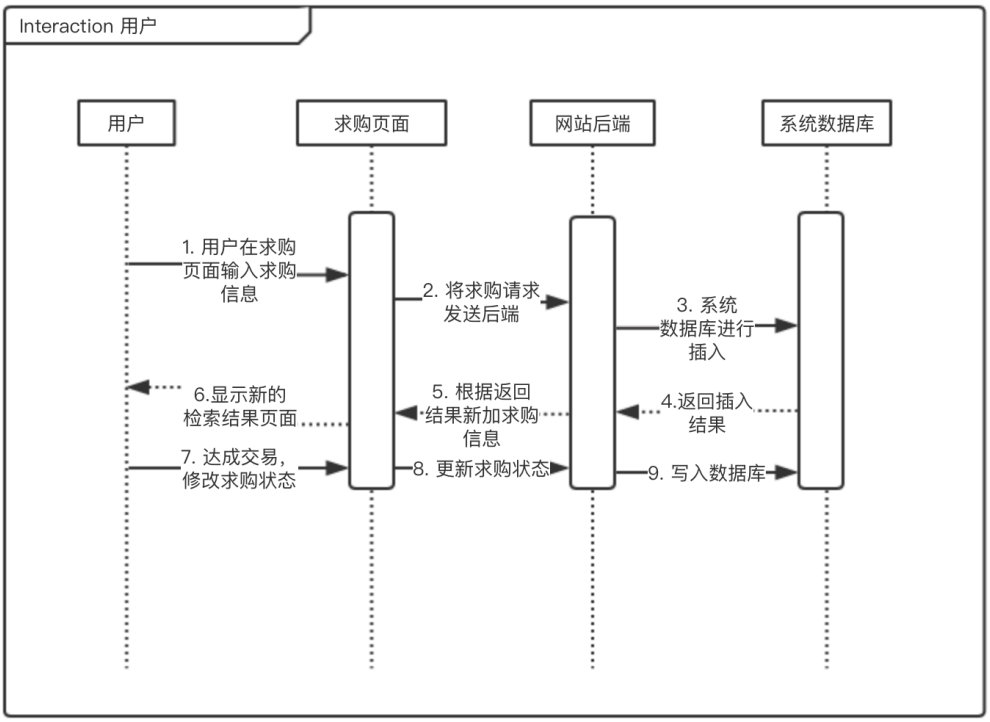


图 5:发布求购信息时序图

7.6 用户发送消息

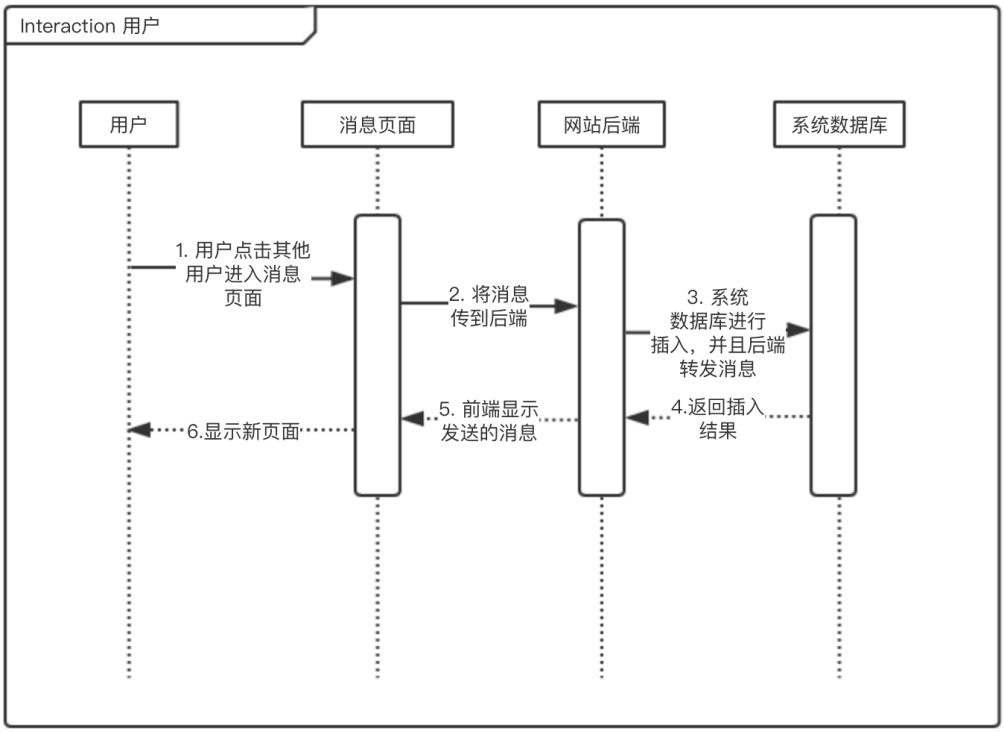


图 6:用户发送消息时序图