基于vue的校园二手交易app的设计 --- 以广东海洋大学为例

发现现象—分析现象，定义问题—对解决方案做需求分析--对解决方案做可行性分析—对app做系统设计—编写代码，实现功能—调试app，完善软件

设计总说明

绪论 > 设计背景及意义 + 二手交易app的现状和发展趋势 ---> 项目需求

软件开发及相关环境 > 开发环境 + 开发工具 + 开发相关技术

技术分析 > vue + vant + nodejs

前端静态实现 + 接口开发实现 + 数据库数据管理实现

项目布局与分析 > 总体开发流程与设计 + 页面UI设计 + 界面设计 + 项目目录结构

测试与发布 > 项目打包上线

设计总结

鸣谢

参考文献

3 可行性分析

3.1 经济可行性：

本软件面向的是海大所有学子，每个用户既是买家也是卖家，这不仅为由闲置物品的同学亦或是想入手物美价廉的物品的同学提供了一个渠道，而且没有中间商赚取差价；在方便同学们做交易的过程中，既能让买家清楚商品的品质，也能让卖家处置闲置物品。而且该软件占内存较小，面向的用户量不大，维护成本不高，并且推广后，可以十分方便地解决闲置物品处置的问题。这种便利受到同学们欢迎的成数是可观的。因此，该软件在经济上是完全可行的。

3.2 技术可行性：

本软件采用的是vue框架以及vant组件框架开发的前端页面，采用nodejs开发接口，采用mongodb数据库做数据管理，实现前后端数据交互，是项目开发常用技术，不需要其他任何复杂的技术或者应用软件的支持。综上，开发该软件是具有技术可行性的。

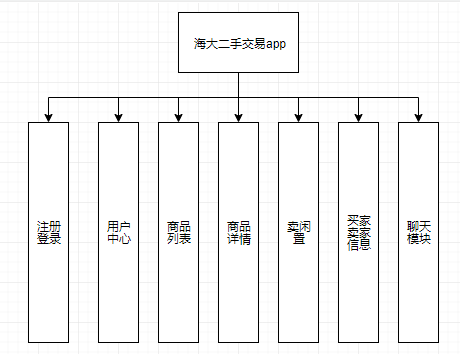
3.3 操作可行性：

本软件采用了滚动与分栏相结合的界面架构，所有操作都是生活中常用的操作，而且每个操作都是一目了然的，功能操作清晰可见，界面简洁清晰，操作方便。因此，该软件具备了操作可行性。

4 需求分析

4.1 功能需求

4.1.1 功能块划分



4.1.2 各功能模块概述

4.1.2.1 注册登录模块

1）用户通过学号、手机号、手机验证码注册并输入新密码成为平台用户

2）用户可以通过学号或手机号、密码登录到本app，从而进行商品浏览或者商品上传等操作

3）当用户忘记密码时，可通过注册手机的手机验证码的方式进行密码重置。

4.1.2.2 用户中心模块

1）用户可通过用户中心页面进入登录注册界面

2）登陆之后，可在用户中心设置个人姓名、宿舍、学号等信息

3）可查看个人已买、已卖、正在售卖、正在交易的商品

4）可对已买商品做评价操作，对正在售卖商品做编辑操作；

4.1.2.3 商品列表模块

1）针对不同类型的商品以标签切换的方式展示，通过点击不同类型进入各类商品列表

2）列表页展示一类商品的图片、几成新、卖家所在区域、以及价格

3）用户可通过下拉菜单选取查看东区、西区、中区各区的商品

4.1.2.4 商品详情模块

1）商品详情展示的是商品的照片、几成新、卖家转手原因、卖家名字、卖家所在区域、商品价格

2）通过点击卖家姓名可查看卖家信息

3）通过点击聊天图标可与卖家联系

4）通过点击商品详情折叠面板查看更多商品照片

4.1.2.5 卖闲置模块

1）可通过上传照片、编辑商品标题、价格、所在区域、转手原因、商品详情进行商品上架

4.1.2.6 买家卖家信息模块

1）展示用户的学号、所在区域、所在宿舍，以及用户以往交易过的商品所产生的评价记录

4.1.2.7 聊天模块

1）买家在有意愿与卖家做交易时，可通过聊天功能，与卖家协商好交易时间与交易地点等信息

5 app开发及相关环境

5.1 开发环境

本app开发是基于vue框架开发的webapp，其相关开发工具在window10运行，前期测试在chrome和firefox浏览器。

5.2 开发工具

5.2.1 Visual Studio Code（简称vsCode）

vsCode是一个轻量且强大的跨平台开源代码编辑器，支持windows、mac、linux系统。内置JavaScript、TypeScript和Nodejs支持，而且拥有丰富的插件生态系统，通过安装插件，几乎支持所有主流的编程语言的语法高亮、代码补全、自定义快捷键、括号配对。

vsCode内置了git版本控制，我们可以方便的提交、回滚、拉取和推送，而不需要自己创建各种‘最终版’文档。也可以通过快捷键（ctrl+shift+D）对代码进行调试，可以在图形界面上执行单步调试和多步调试。

5.2.2 mongodb数据库管理工具

MongoDB是一个基于分布式文件存储的数据库，即文件系统管理的物理资源是通过计算机网络和节点相连的，是一个介于关系数据库和非关系数据库之间的产品，是最像关系数据库的功能最为丰富的非关系数据库。MongoDB所支持的数据结构分散，支持类似于json的bson格式，所以可以存储类型复杂的数据。MongoDB最大的特点就是其强大的查询功能，其语法类似于面向对象语言，其特点为性能高、易于部署、易于使用、扩展性高。其主要功能特性为面向‘集合’存储（其基本单位是文档，一个文档即一条数据，是以‘键值对’的方式组成文档存储数据的，集合即一组文档），易于存储对象类型的数据，支持动态查询、完全索引，对于存储在数据库中的文档，我们不需要知道数据的结构。MongoDB支持windows、mac和linux系统。

5.2.3 Robo 3T数据可视化工具

Robo 3T是一款MongoDB数据库的数据可视化工具，可用于管理数据库内容以及内置了一个代码编辑区域，在数据库代码编辑方面提供了开发优化方案，可提高开发和数据保存效率。

5.2.4 NPM包管理工具

NPM（全称Node Package Manage），是nodejs官方的一种包管理工具，一般是随同nodejs一起安装的，常用于从NPM服务器下载第三方包到本地使用、从NPM服务器下载并安装别人已经编写好的命令行程序到本地使用，能够方便得让前端开发人员下载、安装、删除以及管理已经安装的包。

5.3 开发相关技术

5.3.1 HTML语言

HTML（Hyper Text Markup Language）超文本标记语言，它不是一种编程语言，而是一种标识性语言，是一门由一些标签组成的用来制作网页的语言。HTML语言是可以跨平台的，且支持不同数据格式的文件嵌入。

5.3.2 JavaScript语言

JacaScript是一种解释性脚本语言，它可以直接嵌入HTML页面，在绝大多数浏览器的支持下，可以在windows、mac、linux系统上运行，主要用来向HTML页面添加交互行为，也可以实现web页面的人机交互。

5.3.3 vue框架

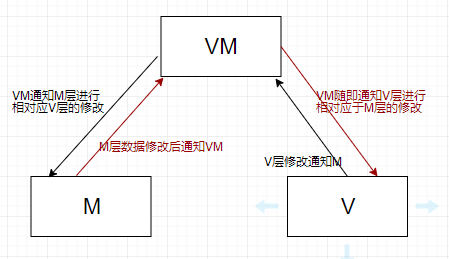
vue是一个用JavaScript写的库，是一个MVVM渐进式框架，MVVM是vue的设计模式，在vue框架中数据会自动驱动视图。

5.3.3.1 MVVM设计模式

view是视图，就是DOM；对应视图也就是html部分—代表UI，它负责将数据模型转换为UI在界面展示出来；model是模型，就是vue页面里的data—代表数据模型，也可以在模型里定义数据修改和操作的业务逻辑；viewModel主要用于监听模型数据，也就是data的改变和控制视图行为、处理用户交互，简单理解就是一个同步view和model的对象。

在MVVM模式下，view层和model层之间并没有直接的联系，而是通过viewModel作为桥梁进行交互，view层和model层之间的交互是双向的，所以，view层的数据变化会同步到model层，model层的数据变化也会同步到view层。

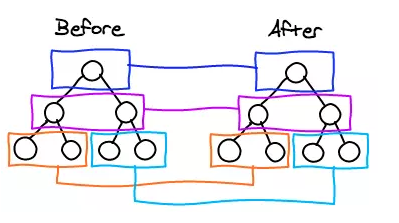
viewModel通过双向数据绑定把view层和model层连接了起来，而view层和model层之间的同步工作完全是自动的，所以，开发中开发者只需要关注业务逻辑而不需要手动的去操作dom节点，也不需要关注数据状态的同步问题，复杂的数据状态维护完全由MVVM模式来统一管理。



5.3.3.2 虚拟DOM（Virtual DOM）和diff算法

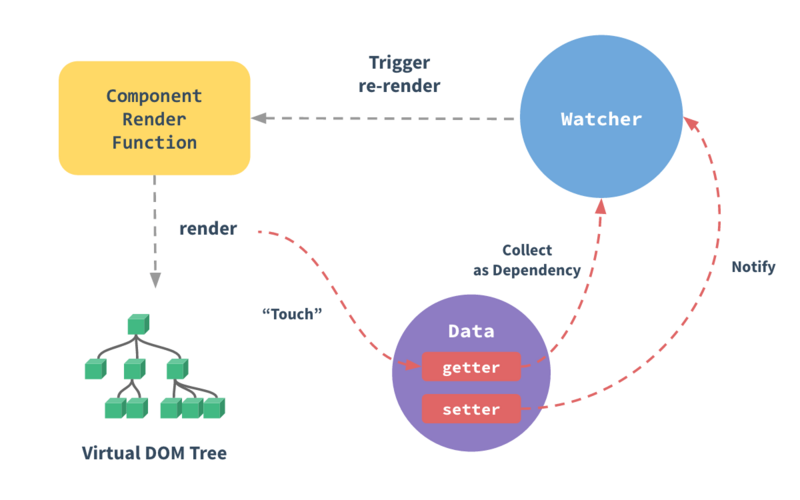
大量的DOM操作是很耗时的，vue框架是数据驱动视图的，用虚拟DOM解决节点操作的时效性。虚拟DOM是用js模拟的结构类似于对象的节点，是可以保存当前视图中所有DOM节点的属性和节点间的关系的树结构。

diff算法是进行新旧虚拟节点元素的对比并返回一个用来存储两个节点不同的地方的patchs对象，最后再用patchs记录消息去局部的更新视图层的DOM。新旧两个树节点完全比较的话需要的时间复杂度为，考虑到效率问题，在diff算法中只会对同级元素进行比较，其时间复杂度仅为，其实就是深层遍历并比较本层级的节点。diff算法的本质就是判断两个虚拟DOM的差异，并将差异更新到真实DOM中。



5.3.3.3 响应式原理

响应式，即数据发生改变时视图会重新渲染，匹配更新为最新的值。vue的响应式原理的核心是通过object.defindeProperty中的访问器属性中的getter和setter方法来拦截数据变化。每个vue实例都会拥有一个专属的观察者watcher，可用于实例更新。vue在视图层展示UI时，通过getter方法获取逻辑层的数据，所以当逻辑层的数据发生改变时，可通过getter方法通知视图层做视图更新；当逻辑层的数据改变时，vue通过属性的setter方法拦截数据变化，setter方法触发当前组件重新渲染，实现视图更新。



5.3.4 nodeJS

JavaScript是一门脚本语言，它需要一个运行环境。在nodeJS之前，JavaScript的运行环境是浏览器，也就是JavaScript在网页中才能跑起来。nodeJS的出现使JavaScript多了一个运行环境。nodeJS封装了一些i/o的线程池管理和网络的i/o操作，可以控制系统文件的读写和监听网络的输入输出，所以nodeJS又可被单纯的认为是一个可以运行JavaScript的服务器。

nodeJS大多被用于前端开发的开发环境，vue-cli也是运行在nodeJS环境下的脚手架，因为有了文件读写的能力，所以vue-cli可以帮我们打包vue核心模块，将代码转义、压缩，nodeJS可用来做服务器、移动端混合应用、移动端原生应用。

5.3.5 mongodb数据库操作

6 app布局设计与功能实现

6.1 总体开发流程与设计

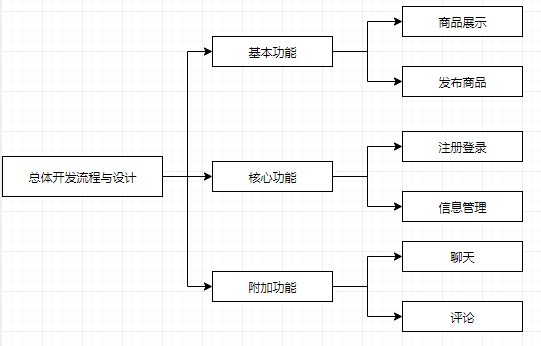


图6.1 总开发流程与设计图

6.2 app业务流程

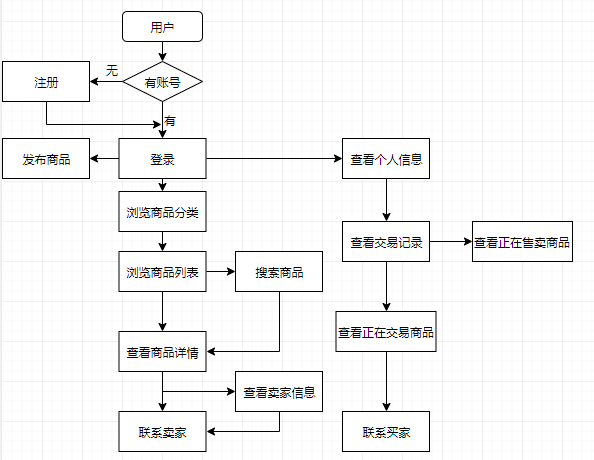


图6.2 app业务流程图

6.3 app各页面设计、功能介绍和实现

6.3.1 注册登录页

只有通过注册功能才能成为本平台用户，通过输入学号、手机号、短信验证码、密码这四个字段才能实现一个账号的注册功能。本平台的用户可以通过学号或者手机号与密码的组合进行登录操作。只有当用户登录了之后，才有访问收藏页面和商品详情页的权限。

注册功能的实现流程为：

1、后台随机生成一个4位的数字串构造验证码；

2、当用户点击‘发送验证码’按钮时，后台获取手机号，并且使用接口向短信平台发送手机号和验证码，然后短信平台再把验证码发送到该手机号上；

3、将验证码、操作时间存入Session中，作为后面验证使用；

4、当用户点击‘注册’按钮时，接收用户填写的学号、手机号、验证码、密码；

5、后台对比提交的验证码与Session中的验证码是否一致，同时判断提交动作是否在有效期内；

6、验证码正确且在有效期内，请求通过，将对应的字段加密后写入数据库的user表中。

登录功能的实现流程为：

1、前端获取用户输入的账号和密码，将账号和密码加密，发送请求；

2、后台接收到前端传入的数据，比较数据库中user表的xuehao字段或者mobile字段，判断该账户是否存在，再判断加密后的密码与对应数据是否一致；判断通过则做相应以及返回token；

3、前端获取后台响应，并将token存储在本地，用于做部分接口调用权限判断。

注册登录页面使用 vant 的 tab标签切换和 from表单组件实现页面的UI设计，以及利用 field组件提供的rules属性实现规则校验实现必填项的输入判断。界面效果图如下：

图6.3 登录注册页

6.3.2 首页

首页页面顶部是app名—海大交易，底部是可以点击切换页面的标签栏；本页面由三部分内容组成，第一部分是由海大四张风景图构成的轮播图，彰显海大风采；第二部分是交易商品的四个总分类：图书、生活百货、饰品、食物，这四类都是大学生活中常见到的闲置物品或者生活用品，点击各图标，都可跳转到对应分类页面；第三部分是各大类别的最新发布商品展示，点击商品可跳转到各商品的详情页面。

首页使用 vant 的 swipe轮播组件、grid宫格组件和 layout布局组件实现页面的UI设计，以及利用 iconfont框架的icon图标做页面展示。界面效果图如下：

图6.4 首页

6.3.3 分类页

分类页使用vant框架的TreeSelect分类选择组件实现页面的UI设计，可以通过点击左侧分类导航栏实现右侧不同商品子类别的显示，点击不同的商品子类别，可跳转至对应的商品列表。界面效果图如下：



6.3.4 商品列表页

在分类页面点击不同的商品子类可进入对应商品列表页面。当点击返回按钮时可以返回上一个访问页面；中间显示当前页面商品所属子类；当点击搜索按钮时可跳转至搜索页面；当点击默认按钮时弹出下拉框并产生遮罩效果，用户可选择指定区域，则页面仅展示当前选择的发布区域匹配的商品。

商品列表获取的实现流程：

1、点击跳转时通过url携带商品子类的signId；

2、进入商品列表页前端立即获取url携带的signId，发送请求到后台；

3、后台通过比较，当数据库goodsList表中商品的parentID等于该signId并且该商品的selled为0时，返回该商品数据；

4、前端获取后台响应的数据，做页面渲染。

指定区域商品展示的实现流程：

1、前端获取用户所选区域对应的addressCode和当前signId，发送请求给后台；

2、后台通过比较，当数据库goodsList表中商品的parentID等于signId和该商品的addressCode和传入的addressCode相等时，返回该商品数据；

3、前端获取后台响应的数据，做页面渲染。

商品列表页使用 vant框架的 NavBar导航栏组件、DropdownMenu下拉菜单组件和 layout布局组件实现页面UI设计，页面商品展示的信息有：商品图片、几成新、卖家所在区域、价格；点击商品可跳转到对应商品详情页。页面效果如下：

6.3.5 搜索页

搜索页面通过设定（或者通过算法统计用户搜索词热度）可产生热门搜索词，当用户点击任一热门搜索词时，页面会自发将该热词写入搜索框，并触发搜索功能。当用户在搜索框输入词汇并执行搜索功能，可获取与该词汇相关的商品，有相关商品则展示在页面上，没有则提示‘抱歉，无找到您想要的结果’；并且该词汇会被记录在历史搜索记录中，当点击‘清空历史搜索’时，可清空历史搜索记录。

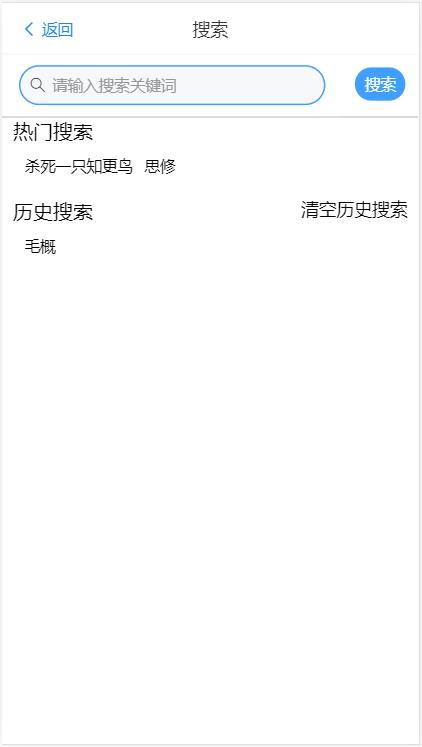
搜索功能的实现流程为：

1、点击热词或者搜索按钮时，前端获取搜索框的数据，发送请求给后端；

2、后端接收到前端发送的数据，采用模糊查询的方法，获取数据库goodsList表中title字段属性值包含该数据的商品，返回对应商品数据给前端；

3、前端获取后台响应的数据，做页面渲染。

搜索页使用 vant框架的 NavBar导航栏组件、Search搜索组件和 layout布局组件实现页面UI设计，搜索出来的商品所展示的信息有：商品图片、几成新、卖家所在区域、价格；点击商品可跳转到对应商品详情页。页面效果如下：

6.3.6 商品详情页

从首页的最新发布商品展示、商品列表页等多处都可直接跳转到商品的详情页，商品详情页的顶部有返回按钮和展示当前商品标题；商品展示信息有：图片、商品名称、几成新、转手原因、卖家、卖家所在区域以及商品详细介绍；当点击图片右下角的心形时，可实现收藏功能，此时实心的心形代表商品已收藏；当点击卖家姓名的时候可跳转至卖家信息页查看卖家个人信息；当点击商品详情时，可展开商品的详细介绍内容；当点击页面右下角的聊天图标时，可发起与卖家的对话。

商品详情的实现流程为：

1、任何进入商品详情页的url都要携带该商品的signId；

2、前端获取url携带的signId，获取本地存储的token，无token，则拒绝发起请求，并做登录提示，流程结束；有token，则发起请求；

3、后台通过比较，当数据库goodsDetail表中商品的signId等于传入的signId时，返回该商品数据；

4、前端获取后台响应的数据，做页面渲染。

收藏功能的实现流程为：

1、当用户执行收藏或者取消收藏操作时，前端获取该商品的signId和当前用户的id，发送请求给后台；

2、后台接收到前端传来的数据，通过对比，当数据库中user表的数据的id等于传入的id时，将传入的signId写入该数据；或者将当前id中的shoucang数组中的对应signId去除，并做响应。

3、前端获得响应，提示商品收藏（取消收藏）成功或者商品收藏（取消收藏）失败。

商品详情页使用 vant框架的 NavBar导航栏组件和 Collapse折叠面板组件实现页面UI设计。页面右下角的聊天图标使用固定定位固定在了视图的右下角。页面效果如下：

6.3.7 卖闲置页

当用户想发布闲置商品的时候，便可通过点击页面底部标签栏的卖闲置进行商品信息的填写，并做发布操作，当点击相机图标时，可选择手机相册图片进行上传；通过输入标题、价格、区域、几成新、转手原因、详情这六个字段和照片上传才能实现一个商品的发布功能。

发布商品功能的实现流程为：

1、用户点击提交按钮，前端获取图片信息以及输入框的信息，发送请求；

2、后端接收到前端发送的数据，将数据写入数据库的goodsDetail表，并做响应；

3、前端获得响应，提示商品发布成功或者商品发布失败。

卖闲置页面使用 vant 的 from表单组件和 Uploader文件上传组件实现页面的UI设计，以及利用 field组件提供的rules属性实现规则校验实现必填项的输入判断。界面效果图如下：

6.3.8 收藏页

当用户在商品详情页点击收藏时，用户的收藏页面就会有对应的商品展示，展示的商品信息有：商品图片、名称、转手原因、卖家、价格；点击卖家姓名可跳转至卖家信息页；将商品往右划，可做交易或者删除操作；执行交易操作，便会跳转至聊天页面，向卖家发起交易邀请；执行删除操作，则会将该商品移出个人收藏列表。

收藏商品展示的实现流程为：

1、进入收藏页时，前端获取用户id，发送请求；

2、后台接收到前端发送的数据，比较数据库中user表的数据，当数据的id与传入的id相等时，获取数据的shoucang数组，遍历该数组，提取signId；

3、将各signId去数据库的goodsDetail表中查找对应数据，将数据返回给前端；

4、前端获取后台响应的数据，做页面渲染。

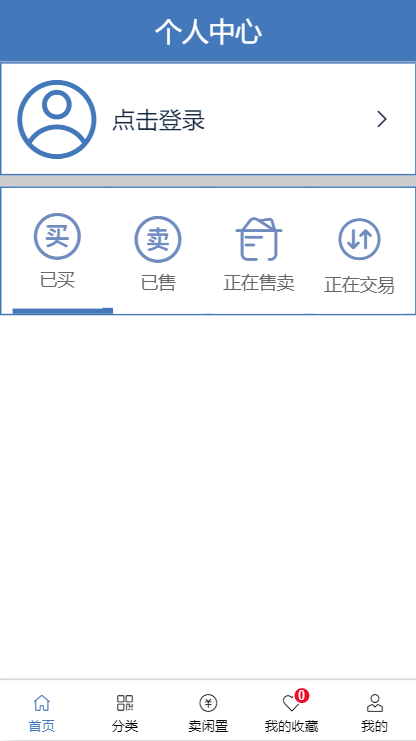
收藏页面使用 vant 的 SwipeCell滑动单元格组件实现页面的UI设计。界面效果图如下：



6.3.9 个人中心页

当用户未登录时，只有访问个人中心页的静态页面，只有点击登录功能。当用户登录过后，页面会展示用户姓名、地址、学号，以及点击已买、已卖、正在售卖、正在交易标签时，会显示对应商品列表，可执行不同的操作；当执行退出登录操作时，页面会变成未登录状态。

个人中心使用 vant 的 Grid宫格组件、SwipeCell滑动单元格组件、Tab标签页组件和 Rate评分组件实现页面的UI设计。界面效果图如下：



6.3.9.1 已买模块

当用户点击已买标签时，可展示该用户在本平台买过的商品，展示的商品信息有：图片、标题、卖家名称、价格、评论；点击卖家名称可跳转至卖家信息页；未评价的商品右滑之后会有评价功能，点击评价，会显示评价页面可做相对应的商品评价。

已买模块的实现流程为：

1、前端获取当前用户的id，以及本地的token，发送请求给后台；

2、后台接收到前端传入的数据，通过比较，当数据库中user表数据的id等于传入的id时，获取数据的yimai数组，遍历该数组，提取signId；

3、将各signId去数据库的goodsDetail表中查找对应数据，将数据返回给前端；

4、前端获取后台响应的数据，做页面渲染。

5、当用户执行评价操作时，前端获取当前商品的sellPeapleId，当前用户id以及输入框的数据，发送请求给后台；

6、后台接收到前端发送数据，通过对比，当数据库中user表中的id等于传入的sellPeapleId时，将其他传入的信息写入该数据的pingLun数组；返回操作结果给前端；

7、前端接收到后台的回应，页面刷新重新渲染。

模块效果图如下：

6.3.9.1 已售模块

当用户点击已售标签时，可展示该用户在本平台卖过的商品，展示的商品信息有：图片、标题、买家名称、价格、评论；点击买家名称可跳转至买家信息页；右滑之后会有删除功能，可在该用户信息下删除该商品展示。

模块效果图如下：



6.3.9.3 正在售卖模块

当用户点击正在售卖标签时，可展示该用户在本平台发布并且未交易的商品，展示的商品信息有：图片、标题、区域、转手原因、详情、价格；右滑之后会有编辑功能，点击编辑，会显示编辑页面可做相对应的商品编辑。

正在售卖模块的实现流程为：

1、前端获取当前用户的id，以及本地的token，发送请求给后台；

2、后台接收到前端传入的数据，通过比较，当数据库中user表数据的id等于传入的id时，获取数据的zhengzaisoumai数组，遍历该数组，提取signId；

3、将各signId去数据库的goodsDetail表中查找对应数据，将数据返回给前端；

4、前端获取后台响应的数据，做页面渲染。

5、当用户执行编辑操作时，前端获取当前商品的signId，当前用户id以及输入框的数据，发送请求给后台；

6、后台接收到前端发送数据，通过对比，当数据库中goodsDetail表中的signId等于传入的signId时，将其他传入的信息替换该数据；返回操作结果给前端；

7、前端接收到后台的回应，页面刷新重新渲染。

模块效果图如下：

6.3.9.4 正在交易模块

当用户点击正在交易标签时，会有正在转手和正在买入标签，点击可显示对于数据，可展示该用户正在交易的商品，展示的商品信息有：图片、买家或卖家姓名、交易区域、价格。点击买家或卖家姓名可跳转到对应信息页。

模块效果图如下：

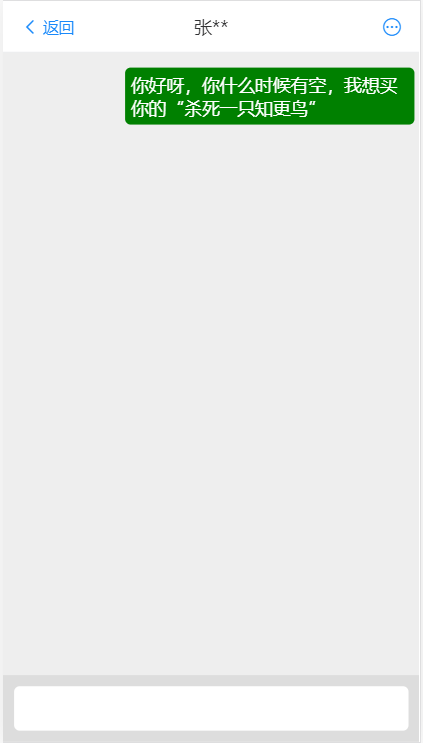
 

6.3.10 买家卖家信息页

当用户通过点击卖家或买家姓名可进入对应个人信息页，页面展示的内容有：姓名、学号、区域、宿舍以及以往交易过商品的同学的评价。点击右下角的聊天图标可进入聊天页面。



6.3.11 聊天页



6.3.12 忘记密码页