

《校园二手物品交易系统设计与实现》检测报告

PaperOK

总相似率：24.54%

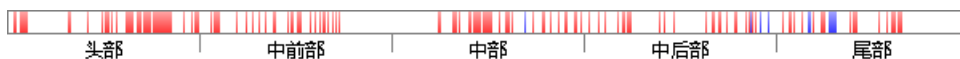
基本信息

文档名称	校园二手物品交易系统设计与实现
报告编号	1bfe7243-e215-49b6-b33b-24d678f27490
文档字数	23170
提交人姓名	张熊鑫
提交方式	粘贴文本检测
检测范围	学位论文库（含硕博）、学术期刊库、会议论文库、法律法规库、互联网资源库、自建比对库
提交时间	2018-05-02 13:49:13

检测报告指标详情

原创率	抄袭率	引用率	字数统计	参考文献字数
75.46%	23.35%	1.19%	23170	-

相似片段位置图



注：红色部分为重度相似，橙色部分为中度相似，蓝色部分为引用部分

相似片段详情（仅显示前10条）

序号	篇名	来源	命中率
1	网上订餐系统的设计与实现	学位论文库	1.22%
2	Python最新Web编程框架Flask研究	学术期刊库	0.67%
3	基于XPCOM的分布式信息交流系统的设计与实现	学位论文库	0.64%
4	新型大学校园二手物品交易平台的构建	互联网资源库	0.64%
5	新型大学校园二手物品交易平台的构建	互联网资源库	0.61%
6	基于php开发的网上商城--<个人毕业设计> - 小霸王易水寒 - 博客园	互联网资源库	0.6%
7	查看用户,游戏豆,充值用户,实卡,赠送游戏币,充值金额,用户,游戏,...	互联网资源库	0.54%
8	网上书店系统	互联网资源库	0.54%
9	三维化的VLCC船货油装卸系统	学位论文库	0.53%
10	一个很流行的版本控制系统Git学习笔记-腾讯云资讯	互联网资源库	0.47%

文档原文标注

攀枝花学院本科毕业设计（论文）

校园二手物品交易系统设计与实现

学生姓名：张熊鑫

学生学号：201410904046

院（系）：数学与计算机学院

年级专业：2014级软件工程

指导教师：张靖 教授

助理指导教师：

二〇一八年六月

摘 要

随着绿色环保生态生活理念的推行越来越多的大学生参与到勤俭节约的行动当中,二手物品的重复循环利用成了大家越来越关注的话题,全国各大学也积极构建自己的二手物品交易平台,在校园中营造节俭之风。随之出现的大量“跳蚤”市场,校园淘宝街,淘宝周,这些平台能否为广大在校师生提供长期的方便的二手物品置换渠道有待考证。在提倡节约环保理念的同时我们更加希望构建一种区别于传统模式的新型的时尚的大学校园二手物品交易平台。实现实体交易与电子商务的有效结合。能够实现二手物品有效快速的重复使用,能够为卖家提供更为广阔的物品展示空间,能够为买家提供更为多样化的购物选择。

此次毕业设计中,对大学生群体对于二手物品交易的需求进行了深入的调研,并采用Python编程语言、Javascript动态网页技术和MYSQL数据库进行系统开发。系统的核心功能涵盖用户注册与登录、二手物品发布与浏览、订单管理、在线购买、账户停用与启用、后台查看统计信息等。

关键字 电商, 二手交易, Python, MYSQL

ABSTRACT

Keywords

目 录

摘 要 I

ABSTRACT II

1 绪论 - 1 -

1.1 选题背景及研究意义 - 1 -

1.1.1 选题背景 - 1 -

1.1.2 研究意义 - 1 -

1.2 课题研究的主要内容 - 1 -

2 相关技术与工具简介 - 2 -

2.1 Sublime Text和Git概述 - 2 -

2.1.1 Sublime Text代码编辑器 - 2 -

2.1.2 Git代码版本管理工具 - 2 -

2.2 Python开发语言简介 - 2 -

2.3 Flask框架简介 - 3 -

2.4 前端技术简介 - 3 -

2.4.1 MVVM简介 - 3 -

2.4.2 Vue简介 - 4 -

2.4.3 SPA技术简介 - 4 -

2.5 MYSQL 简介 - 4 -

3 系统分析 - 5 -

3.1 可行性分析 - 5 -

3.1.1 技术可行性分析 - 5 -

3.1.2 经济可行性分析 - 5 -

3.1.3 操作可行性分析 - 5 -

3.2需求分析 - 6 -

3.2.1 功能需求 - 6 -

3.2.2 性能指标 - 6 -

3.3 用例分析 - 7 -

3.3.1 管理员用户 - 7 -

3.3.2 普通用户 - 8 -

4 数据库设计 - 9 -

4.1 数据库设计 - 9 -

4.1.1 E-R图 - 9 -

4.1.1 表基本设计 - 11 -

4.2数据库结构设计 - 11 -

5 系统设计 - 14 -

5.1 数据访问实现 - 14 -

5.2 功能实现 - 15 -

5.2.1 注册登录模块 - 15 -

5.2.2 发布闲置模块 - 18 -

5.2.3 个人信息模块 - 21 -

5.2.4 后台统计模块 - 23 -

5.2.5 用户管理模块 - 25 -

5.2.6 浏览商品模块 - 27 -

6 系统测试 - 29 -

6.1测试记录 - 29 -

6.1测试结论 - 30 -

总 结 - 31 -

参 考 文 献 32

致 谢 33

1 绪论

1.1 选题背景及研究意义

1.1.1 选题背景

现代社会,随着校园网络覆盖面积越来越广,各类电子商务类网站越来越健全细化,各类商场超市也瞄准校园市场纷纷在学校周边设立分店,很大程度上的解决大学生对于生活学习上的各类商品的需求。伴随着大学生的购买能力的提高,每年的升学和毕业期间新物的购入与旧物的处理过程中,每个学生在大学四年都会积攒大量二手物品。

1.1.2 研究意义

高校人数众多,均在10000人以上,有的甚至更多,每年老师、学生将产生大量的二手物品并可以重新利用,如书籍、运动物品、电子产品、床上物品、衣物等等,充分利用二手物品将有效节约能耗,降低学生开支,并促进生态校园创建。互联网的广泛普及和发展,各种形式的金融、商业活动都依托互联网进行交易,给人们的生活带来了许多便利,人们越来越重视电子商务。通过在线系统宣传管理以及交易可以极大的实现二手物品再利用,校园二手物品交易系统是校园电子商务的一个典型代表。高校校园是年轻人聚集的扎堆聚集的地方,师生们有着极大的二手物品买卖需求,比如教科书、手机、电脑、网球拍等。尤其当毕业季来临,毕业生如何妥善处理不方便带走的生活物品成了件棘手的事情。于是开始整合高校校园的二手资源,将他们分门别类的规划入二手物品交易平台上,有其实意义。

1.2 课题研究的主要内容

本系统结合当前的校园环境以及目前大学生的生活习惯与节奏，依赖Python这一高级编程语言设计的网上图书销售系统。该系统符合传统的Windows操作习惯，使用起来基本无任何困难，服务对象为全体民众。本次研究的内容包括了：用户注册与登录、二手物品发布与浏览、订单管理、在线购买、账户停用与启用等若干功能。

本文介绍了基于Python的校园二手物品交易系统的开发过程，开发工具和框架的选择，功能的设计，数据的呈现以及开发过程中遇到的问题和解决的办法，最后还有程序的测试等等。

2 相关技术与工具简介

2.1 Visual Studio Code和Git概述

2.1.1 Visual Studio Code代码编辑器

Visual Studio Code是一个轻巧、简便的代码编辑器，同时也是HTML和散文先进的文本编辑器。由微软开发的，同时支持Windows、Linux和macOS操作系统的开源文本编辑器。开发最初被设计为一个具有丰富扩展功能的Vim。

Visual Studio Code能够实现代码完成功能，同时还支持多种编程语言的语法高亮，并且还拥有代码片段的功能，可以将常用的代码片段作为模板保存起来，在需要时随时调用。Visual Studio Code基于Electron开发。Electron是一个基于Chromium的项目，可用于开发基于Node.js的本地应用程序。Visual Studio Code使用Blink排版引擎渲染用户界面。虽然Code基于Electron框架，但并不是Atom的复刻。Code基于代号“Monaco”的编辑器核心制作，与Visual Studio Team Services相同。

Visual Studio Code还能够实现编辑状态恢复的能力，即当用户修改了一个文件，但忘记保存，这时意外退出软件，软件将不会询问用户是否要保存，因为无论是用户自发退出还是意外崩溃退出，下次启动软件后，将会完整恢复到之前的编辑状态，就跟退出之前时候一样。

2.1.2 Git代码版本管理工具

Git是一种非常流行的分布式版本控制系统，它和其他版本控制系统的主要差别在于Git只关心文件数据的整体是否发生变化，而大多数版本其他系统只关心文件内容的具体差异，这类系统（CVS，Subversion，Perforce，Bazaar等等）每次记录有哪些文件作了更新，以及都更新了哪些行的什么内容。Git另一个比较好的地方在于绝大多数操作都可以在本地执行，而每个本地都可以从服务器获取一份完整的仓库代码，而且在没网的时候仍然可以修改和使用大部分命令，在方便的时候再跟服务器进行同步，这样可以更好的实现多人联合编程。

2.2 Python开发语言简介

Python是一种面向对象的解释型计算机程序设计语言，由荷兰人Guido van Rossum于1989年发明，第一个公开发行人版发行于1991年。Python是纯粹的自由软件，源代码和解释器CPython遵循GPL协议。Python语法简洁清晰，特色之一是强制用空白符(white space)作为语句缩进。

Python具有丰富和强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地联结在一起。常见的一种应用情形是，使用Python快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如3D游戏中的图形渲染模块，性能要求特别高，就可以用C/C++重写，而后封装为Python可以调用的扩展类库。需要注意的是在您使用扩展类库时可能需要考虑平台问题，某些可能不提供跨平台的实现。

2.3 Flask框架简介

Flask是一个使用 Python 编写的轻量级 Web 应用框架。其 WSGI 工具箱采用 Werkzeug，模板引擎则使用 Jinja2。Flask使用 BSD 授权。Flask也被称为“microframework”，因为它使用简单的核心，用 extension 增加其他功能。Flask没有默认使用的数据库、窗体验证工具。

Flask也被称为“microframework”，因为它使用简单的核心，用 extension 增加其他功能。Flask没有默认使用的数据库、窗体验证工具。

Flask是当下流行的Web框架，它是用Python实现的。Flask显著的特点是：它是一个“微”框架。“微”意味着Flask旨在保持核心的简单，但同时又易于扩展。默认情况下，Flask 不包含数据库抽象层、表单验证，或是其它任何已有多种库可以胜任的功能。然而，Flask 支持用扩展来给应用添加这些功能。众多的扩展提供了数据库集成、表单验证、上传处理、各种各样的开放认证技术等功能。Flask的这些特性，使得它在Web开发方面变得非常流行。

2.4 前端技术简介

2.4.1 MVVM简介

MVVM有助于将图形用户界面的开发与业务逻辑或后端逻辑（数据模型）的开发分离开来，这是通过置标语言或GUI代码实现的。MVVM的视图模型是一个值转换器，这意味着视图模型负责从模型中暴露（转换）数据对象，以便轻松管理和呈现对象。在这方面，视图模型比视图做得更多，并且处理大部分视图的显示逻辑。视图模型可以实现中介者模式，组织对视图所支持的用例集的后端逻辑的访问。

MVVM由微软架构师Ken Cooper和Ted Peters开发，通过利用WPF（微软.NET图形系统）和Silverlight（WPF的互联网应用派生品）的特性来简化用户界面事件驱动程序的设计。微软的WPF和Silverlight架构师之一John Gossman于2005年在他的博客上发表了MVVM。

MVVM是一个源自服务端的开发模式，谷歌公司于2014首次将其引入到前端开发领域（AngularJs），AngularJs的出现在前端开发界掀起轩然大波，使用AngularJs之后的前端开发不用去操作dom就能实时响应更改，这不仅提高了页面性能，更提高了生产力。本系统所采用的Vue也是一种基于mvvm模式的前端框架。

2.4.2 Vue简介

Vue是一套基于javascript脚本编程语言的MVVM框架，主要用于渐进式的构建用户界面。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

2.4.3 SPA技术简介

单页应用（英语：single-page application，缩写SPA）是一种网络应用程序或网站的模型，它通过动态重写当前页面来与用户交互，而非传统的从服务器重新加载整个新页面。这种方法避免了页面之间切换打断用户体验，使应用程序更像一个桌面应用程序。在单页应用中，所有必要的代码（HTML、JavaScript和CSS）都通过单个页面的加载而检索，或者根据需要（通常是为响应用户操作）动态装载适当的资源并添加到页面。尽管可以用位置散列或HTML5历史API来提供应用程序中单独逻辑页面的感知和导航能力，但页面在过程中的任何时间点都不会重新加载，也不会将控制转移到其他页面。与单页应用的交互通常涉及到与网页服务器后端的动态通信。

2.5 MYSQL 简介

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

3 系统分析

3.1 可行性分析

3.1.1 技术可行性分析

本系统采用现代、易扩展的前后端分离系统架构，即服务端需要做接口的开发，而不用关心界面如何实现以及以什么方式实现。这样不仅能做到前后端充分解耦，而且提高了服务端代码的多端可用性，即后台只需要一套代码即可为PC客户端、web网页、手机APP提供服务。在具体实现方面，使用了mysql数据库以及基于python3的flask框架进行服务端开发，软件架构符合RESTful规范。前台HTML页面使用vue框架实现脚本交互，使得开发更加高效、快捷。更为值得一提的是整个前台页面使用了element-ui界面框架，使得页面元素更加丰富、操作更符合当代年轻人的使用习惯。美国著名的社交网站“脸书”在后台系统中大量使用Python。综上所述，本系统具备技术可行性。

3.1.2 经济可行性分析

在互联网浪潮的推动下，平民百姓也可以轻易的享受到科技进步带来的便利。尤其在最近几年，在国家政策引导下，各大运营商都对宽带进行了降低资费、提升网速，大众可以以极小的经济开销接入互联网。而移动互联网的普及又是另一场新的革命，手机性能越来越好，上网速度也越来越快。同时，手机的价格也呈现出多样化，虽然在高端市场会出现动辄七八千元的产品，但是仍然存在很大批量性价比极高的国产手机，有的运营商甚至会做一些“安宽带送手机”的活动。总而言之，现在要获得一部性能优异的手机经济门槛极低。从以上角度说来，本系统具备经济可行性。

3.1.3 操作可行性分析

从上世纪九十年代开始，我国就开始关注互联网技术的发展，尤其注重在高校中，让学生们能够通过信息化的手段，更便捷的进行多样化的学习。同时也让教职员工大量的使用信息化的产品简化办公流程、提升教学质量。经过数十载的实践，师生们已经具备极强的操作在线系统的能力。因此，本系统具备操作可行性。

3.2需求分析

3.2.1 功能需求

开发任何一个程序系统都是旨在解决生产生活中的问题。为本系统的顺利完成，我走访了大量的在校师生，收集了一些较为集中的功能性需求，完成了基于Python的校园二手物品在线交易平台的设计与开发，本系统主要功能如下所示：

① 注册登录：使用用户手机号作为系统账号，通过实时发送短信验证码的方式进行权限验证。可以设置并修改个人昵称、收货地址、支付宝账号等个人信息。

② 发布二手物品：用户可以在商城中发布自己的闲置物品，并支持商品定价、上传商品图片、填写物品名称、添加描述信息等。系统会定期检查商城中的物品是否违规，对发布违禁物品的用户进行账号停用并下架相关物品。

③ 浏览二手物品商城：可以在商城中浏览自己或其他用户发布的二手物品。单个物品信息应该至

少包括物品名称、价格、发布者昵称和手机号，如果发布者上传了物品的图片，那么应该展示该物品缩略图，否则展示默认图。

④ 在线购买：支持微信扫码支付。

⑤ 个人信息查看与修改：可以查看自己的账户创建日期、昵称、支付宝账号、手机号码、收货地址，可以修改昵称、支付宝账号、收货地址。

⑥ 订单管理：可以看到自己买入或卖出的物品交易信息。订单分为交易中、交易完成两个状态。

⑦ 账号管理：系统管理员有权限对普通用户的账号进行停用或启用。通常来说，停用某一个账号的原因是该用户进行了违规操作，比如重复多次发布同样的商品到系统中，或是发布了“烧水棒”之类的违禁物品。

⑧ 后台查看统计数据：系统管理员可以以月为维度查看系统的交易统计、注册用户统计、商品分类统计。

3.2.2 性能指标

对于一个成熟的大型电商类网站来说，性能指标几近严苛。而本系统受限于投入资源不足，在性能方面自然做不到尽善尽美。不过，在有限的条件下，系统依然做出很大程度上的性能优化，具体的性能指标如下：

① 响应时间：要求对用户操作进行快速响应，对于异常情况要给出友好、易读的提示语，应极力避免页面反应迟钝甚至卡死的现象。接口的响应时间应尽可能的控制在1s之内。

② 系统安全性：鉴于大多数人互联网安全意识不强，比如会在不同的网站使用相同的密码，那么一旦在某个网站的信息泄露，就会造成连带的安全问题。而本系统使用了实时短信验证的方式进行注册、登录，不去存储用户密码。所以可以更大程度保证用户信息安全。

③ 稳定性：当系统上线之后，应保证在正常用户使用时间段之外进行升级维护，而每一次升级不能影响原有数据和功能。作为一个可上线系统，应至少保证用户在99%的情况下可用。

④ 高并发承载：对于一些特定的时间节点，可能会产生大量用户同时使用系统，系统应保证高并发情况下可用。

⑤ 资源利用率：服务器上的资源通常是指CPU占用率、内存使用率、磁盘使用率。一般，当CPU占用率达到70%,服务就接近饱和了。对内存的使用更是要密切注意，应避免内存泄露。磁盘的使用我们主要关注读写性能和剩余空间，应及时清除垃圾文件，并及时扩容。

3.3 用例分析

本系统是面向所有师生的在线二手商品交易网站，使用人员只涉及到管理员和普通用户两种用户，针对这两种用户类型，系统做了权限控制。

3.3.1 管理员用户

系统管理员账号由网站运营方进行管理，为区分管理员与普通用户，系统为管理员账户的登录新增了接口。使用管理员账户登录系统后可以查看后台统计数据，比如按月查看注册人数、交易额等，也可以通过搜索用户手机号对账户进行停用。管理员用户用例图如图3.1所示。

图3.1 管理员用户用例图

3.3.2 普通用户

本系统本着公平、开放的互联网精神，并没有对注册者进行身份验证，换言之，只要有买卖二手物品的需求的人员，都可以使用本系统。普通用户可以通过在我们的网站上注册获得发布闲置二手物品的权限，同时系统支持查看并修改个人信息、订单管理、浏览商城等功能，普通用户用例图如图3.2所示。

图3.2 普通用户用例图

?

4 数据库设计

数据库的设计是计算机软件设计过程中的重要内容，是支撑计算机软件系统运行的一个重要因素，数据库设计的好坏将会直接影响到系统运行的稳定与否。由此看来，数据库设计显得尤为重要。一个优秀的数据库设计在一个性能要求较高的计算机软件应用中的重要性不言而喻。本章根据艺术品销售平台系统分析的结果将详细阐述整个数据库的设计及实现过程。

4.1 数据库设计

系统处理和存储的内容就是数据，本艺术品销售平台是一个以数据作为支撑的 Web 应用程序，使用关系数据库作为数据的存储方式。因此，在完成整个系统的分析后，首要任务便是完成数据库的设计，为接下来的详细设计做好准备工作。

4.1.1 E-R图

将需求分析得到的用户需求抽象为信息结构即概念模型的过程称之为概念结构，是各种数据模型的共同基础，比起数据模型更独立于机器、也更加更抽象，从而更加稳定。E-R 图也称实体-联系图，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型，是表示概念结构的有效方式。

(1) 用户实体属性主要包括：

用户ID、用户昵称、用户手机号、用户状态、收货地址、创建时间、支付宝账号。如图4-1所示。

图4.1 用户表E-R图

(2) 商品实体属性主要包括：

商品ID、发布者ID、发布者昵称、商品图片url、商品缩略图url、商品状态、商品名称、商品描述、商品价格、商品原价、分类ID、创建时间。如图4.2所示。

图4.2 商品表E-R图

(3) 订单实体属性主要包括：

订单ID、商品ID、发布者ID、发布者昵称、购买者昵称、购买者ID、商品缩略图、商品价格、订单状态。

图4.3 订单表E-R图

4.1.1 表基本设计

在本系统的数据库中，建立了以下表结构：

(1) 用户表

User(userId,loginName,userPhone,userPhoto,userStatus,createTime,alipay,stress)

(2) 商品表

Goods(goodsId,ownerId,ownerName,goodsImg,goodsName,isSale,goodsDesc,attrCatId,createTime,currentPrice,originalPrice,goodsThums)

(3) 订单表

Goods_order(orderId,goodsId,ownerId,ownerName,customerId,customerName,goodsThums,price,orderStatus,ownerDisplay,customerDisplay)

4.2数据库结构设计

通过 E-R 图对数据库的字段（包括名称、数据类型、长度）、主键、外键、约束、触发器等进行设计与详细实现。

(1) 用户信息表：

用户信息表(user)，包括八个字段：userId(用户id)，loginName(用户昵称)，userPhone(用户电话)、userStatus(用户状态)、createTime(创建时间)、alipay(支付宝账号)、stress(收货地址)。

表4.1 user(用户表)

中文名	字段名	数据类型	主键	外键	必填	备注
-----	-----	------	----	----	----	----

用户id	userId	int(11)	Y	Y	自增	
------	--------	---------	---	---	----	--

用户昵称	loginName	Varchar(50)	Y			
------	-----------	-------------	---	--	--	--

用户电话	userPhone	int(11)	Y			
------	-----------	---------	---	--	--	--

用户状态	userStatus	Int(1)	Y			
------	------------	--------	---	--	--	--

创建时间	createTime	datetime	Y			
------	------------	----------	---	--	--	--

支付宝账号	alipay	varchar(255)	Y			
-------	--------	--------------	---	--	--	--

收货地址	stress	varchar(255)	Y			
------	--------	--------------	---	--	--	--

(2) 商品表：

商品表(goods)，包含11个字段：goodsId(商品id)，ownerId(发布者id)，ownerName(发布者昵称)，goodsImg(商品图片url)，goodsName(商品名称)，isSale(商品状态)，goodsDesc(商品描述)，attrCatId(分类Id)，createTime(创建时间)，currentPrice(商品价格)，originalPrice(商品原价)，goodsThums(商品缩略图url)

表4.2 goods(商品表)

中文名 字段名 数据类型 主键 外键 必填 备注

商品id goodsId Varchar(255) Y Y 自增

发布者id ownerId Int(11) Y Y

发布者昵称 ownerName Varchar(255) Y

商品图片url goodsImg Varchar(255) N

商品名称 goodsName Varchar(255) Y

商品状态 isSale Int(1) Y

商品描述 goodsDesc Varchar(255) N

分类Id attrCatId Int(1) Y

创建时间 createTime datetime Y

商品价格 currentPrice Float(8,2) Y

商品原价 originalPrice Float(8,2) N

商品缩略图url goodsThums Varchar(255) N

(3) 订单表：

订单表(goods_order)，包含11个字段：orderId(订单id)，goodsId(商品id)，ownerId(发布者id)，ownerName(发布者昵称)，customerId(购买者id)，customerName(购买者昵称)，goodsThums(商品缩略图url)，price(订单价格)，orderStatus(订单状态)，ownerDisplay(发布者可见)，customerDisplay(购买者可见)

中文名 字段名 数据类型 主键 外键 必填 备注

订单id orderId Varchar(255) Y Y 自增

商品id goodsId Varchar(255) Y Y

发布者id ownerId Varchar(255) Y

发布者昵称 ownerName Varchar(255) Y

购买者id customerId Varchar(255) Y

购买者昵称 customerName Varchar(255) Y

商品缩略图url goodsThums Varchar(255) N

订单价格 price Float(8,2) Y

订单状态 orderStatus Int(1) Y

发布者可见 ownerDisplay Int(1) Y

购买者可见 customerDisplay Int(1) Y

?

5 系统设计

5.1 数据访问实现

任何系统的正常工作都离不开数据库的读写操作，本系统的数据读写使用了flask-mysql模块，借助该模块，可以创建、连接数据库，并进行增删查改操作。其中用于数据库操作的各个模块的功能主要包括：

(1) 连接数据库相关的代码

```
from flask.ext.mysql import MySQL #引入mysql模块

mysql = MySQL() #实例化mysql对象

app = Flask(__name__) #实例化mysql对象

app.config['MYSQL_DATABASE_USER'] = 'root' #设置数据库用户

app.config['MYSQL_DATABASE_PASSWORD'] = 'root' #设置数据库密码

app.config['MYSQL_DATABASE_DB'] = 'secondhand' #选择数据库

app.config['MYSQL_DATABASE_HOST'] = 'localhost' #设置地址

mysql.init_app(app) #应用以上数据库配置

conn = mysql.connect() #创建连接并赋值给一个全局变量
```

(2) 操作数据表相关的代码

```
sql = "select userStatus, loginName from user where userPhone=%s;" % (userPhone) # 查询用户是否存在sql语句
```

```
sql = "insert into user(userId, loginName, userPhone, alipay, stress, userStatus, createTime) values ('%s', '%s', '%s', '%s', '%s', 1, '%s')" % (userId, loginName, userPhone, alipay, stress, createTime) #注册新用户sql语句
```

```
sql = "update user set alipay='%s', loginName='%s', stress='%s' where userId='%s';" % (alipay, loginName, stress, userId) #更改用户信息sql语句
```

```
sql = "select * from admin where adminId='%s';" % (adminId) #管理员登录验证sql语句
```

```
sql = "update user set userStatus='%s' where userId='%s';" % (userStatus, userId) #控制用户停用与启用sql
```

```
sql = "select * from user where userStatus=0;" #查询被禁用账户sql
```

```
sql = "select * from user where userId='%s';" % (userId) #通过id查询用户sql语句
```

```
sql = "select * from goods limit %s, %s;" % (start, pageSize) #商品浏览分页功能sql语句
```

```
sql = "insert into goods values ('%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s');"
% (goodsId, ownerId, ownerName, goodsImg, goodsName, isSale, goodsDesc, attrCatId,
createTime, currentPrice, originalPrice, goodsThums) #添加商品sql语句
```

```
sql = "select createNy, count(*) from user group by createNy; "
```

```
# 按月查询注册统计
```

```
sql = "select createNy, count(*) from goods_order group by createNy; " # 按月查询交易统计
```

```
cursor = g.db.cursor() #创建连接对象
```

```
cursor.execute(sql) #执行sql
```

```
data = cursor.fetchall() #接收sql语句执行结果
```

```
g.db.commit() #提交事务，否则相关更改不会影响数据库
```

```
g.db.close() #释放连接
```

5.2 功能实现

5.2.1 注册登录模块

想要正常使用本系统所提供的全部功能，用户需要使用手机号进行注册，然后进行登录，如果检测到该手机号已经注册过，系统将为其自动登录。另外，本系统完全摒弃了传统的密码登录方式，采用了实时短信验证的方式，进而很大程度上提高了系统的安全性。短信验证码使用了“秒滴科技有限公司”提供的接口，实现逻辑是先由客户端javascript代码生成一个随机6位数，然后加密存储到客户端session中，用户在手机上收到验证码后及时输入到页面与session中缓存的加密验证码进行比较验证。相关代码如下所示。

```
import md5 from 'md5' //引入md5加密模块
```

```
//手机号码正则验证函数
```

```
mobile (str) {  
    return /^1[0-9]{10}$/.test(str)  
}
```

```
// 生成六位数随机码
```

```
random () {
```

```

var num = "

for (let i = 0; i < 6; i++) {

num += Math.floor(Math.random() * 10)

}

console.log(num)

return num

},

var flag = checkType.check(this.phoneNum, 'mobile') ////手机号码验证

if (!flag) {

this.sendCodeCallback('error', '手机号码不合法')

} else {

var code = this.random()

var m = 5

var params = {

accountSid: '26d1714cd0614834a0d62db2c002a730',

to: this.phoneNum,

timestamp: Date.parse(new Date()),

smsContent: `【惠物品】您的验证码为${code}，请于${m}分钟内正确输入，如非本人操作，请忽略此短信。`,

sig:md5('26d1714cd0614834a0d62db2c002a73055ba8bef73654108b7688c84de9c9ee2' +
Date.parse(new Date()))

}

var pwd = md5(`${code}${this.phoneNum}xcool`)

store.set('pwd', pwd)

ajax('api/getCode', { method: 'POST', params })

.then(res => {

this.timeFlag = true

this.getCode = 60

```



```

this.sendCodeCallback('success', '验证码发送成功，请注意查收！')

setTimeout(() => {

store.remove('pwd')

}, 30000)

this.timer = setInterval(() => {

this.getCode === 0 ? this.isTime() : this.getCode--

}, 1000)

})

}

```

其中params为第三方短信验证接口所需参数，由于安全限制，只能由服务端发起请求，所以在客户端将相关参数封装后发到了后台，由后台去执行，相关后台代码如下。

发送post请求函数

```
def sendPost(url, data, headers):
```

```
    r = requests.post(url, data, headers)
```

```
    return json.loads(r.text)
```

短信验证平台接口

```
class GetCode(restful.Resource):
```

```
    def post(self):
```

```
        # result = make_response()
```

```
        postdata = request.args
```

```
        url = 'https://api.miaodiyun.com/20150822/industrySMS/sendSMS'
```

```
        headers = {'content-type': 'application/x-www-form-urlencoded'}
```

```
        result = outApi.sendPost(url, postdata, headers)
```

```
        return result
```

得益于这种创新的登录验证方式，页面用不着堆砌诸如记住密码、找回密码之类的链接，所以页面可以设计得极其的简洁、美观。注册登录界面如图5.1所示。

图5.1 注册/登录界面图

在正确输入手机号和验证码后，系统将查询该手机号是否已经注册过，如果注册过将跳转到首页

, 否则会让用户填写一些必要的个人信息, 填写表单如图5.2所示。

图5.2 填写个人信息界面图

5.2.2 发布闲置模块

用户登录系统后, 可以点击页面左上角的“发布闲置”按钮发布自己的闲置物品到系统商城中, 添加页面如图5.3所示。

图5.3 发布闲置物品界面图

在实现过程中, 考虑到系统性能优化, 并没有将图片直接存在系统文件服务器上, 而是选用“七牛云”提供的对象存储服务。这样做最大的好处是可以为图片资源提供CDN加速, 进而提高网页加载速度。当然, 由于数据库可以存储图片的网络地址, 系统就没有设置文件服务器的必要, 这样一来, 使得系统具有极强的迁移能力。相关的代码如下所示。

```
# 服务端获取图片上传token
```

```
def getUploadToken(self):
```

```
    q = Auth(AccessKey, SecretKey)
```

```
    token = q.upload_token(bucket_name, self, delay)
```

```
    return token
```

```
//客户端七牛云上传图片接口
```

```
upload (f, token, key) {
```

```
    var xhr = new XMLHttpRequest()
```

```
    xhr.open('POST', region, true)
```

```
    var formData
```

```
    formData = new FormData()
```

```
    formData.append('key', key)
```

```
    formData.append('token', token)
```

```
    formData.append('file', f)
```

```
    xhr.onreadystatechange = (res) => {
```

```
        if (xhr.readyState === 4 && xhr.status === 200 && xhr.responseText !== '') {
```

```
            var result = JSON.parse(xhr.responseText)
```

```
            console.log(result, '上传图片返回')
```

```
this.form.goodsImg = `${ImgBaseUrl}/${result.key}`  
  
this.form.goodsThums = `${ImgBaseUrl}/${result.key}/thumb`  
  
this.commitItem()  
  
} else {  
  
  // console.log(res, '上传失败')  
  
}  
  
}  
  
xhr.send(formData)  
  
}
```

在上传图片到七牛云的过程中，为解决图片命名重复的问题，在客户端对本地图片进行了重命名，命名规则是先读取文件后缀，然后在前面追加uuid，相关代码如下所示。

```
// 本地图片上传  
  
update (e) {  
  
  let file = e.target.files[0]  
  
  this.file = file  
  
  switch (file.type) {  
  
    case 'image/x-icon':  
  
      this.fileType = `.icon`  
  
      break  
  
    case 'image/png':  
  
      this.fileType = `.png`  
  
      break  
  
    default:  
  
      this.fileType = `.jpg`  
  
  }  
  
  let src = URL.createObjectURL(file)  
  
  this.imgSrc = src
```

```

var fileName = uuid()

this.key = `${fileName}${this.fileType}`

},

// 客户端简单实现的生成uuid函数

export const uuid = () => {

  return 'xxxxxxxxxxxx4xxxyxxxxxxxx'.replace(/[xy]/g, c => {

    var r = Math.random() * 16 | 0

    var v = c === 'x' ? r : (r & 0x3 | 0x8)

    return v.toString(16)

  })

}

```

5.2.3 个人信息模块

考虑到用户的支付宝账号、收货地址等个人信息可能会发生变动，为此系统提供了修改个人信息的功能，用户点击主页面右上角的“个人中心”即可进入到个人信息展示页，点击“修改”按钮即可对可更改信息进行更改。[个人信息展示页面如图5.4。](#)

图5.4 查看个人信息界面图

个人信息修改页面如图5.5所示。

图5.5 修改个人信息界面图

考虑到查看个人信息与修改个人信息页面相似度极大，所以可以共用一套代码，其中使用“修改”按钮去绑定支付宝账号、昵称、收货地址input是否可写的属性，在点击之后立刻将按钮的值改为“提交”并绑定唯一的id。总体来说，实现代码复用的方式就是为按钮绑定不同的属性，点击之后根据属性id决定应该响应什么操作，相关的代码如下所示。

```

// 前端代码

modify () {

  if (this.btn.id === 0) {

    this.writeable = false

    this.btn = {

      id: 1,

      value: '提交'

```

```

    }

    } else {

    let params = {

    alipay: this.alipay,

    loginName: this.loginName,

    stress: this.stress,

    userId: this.userPhone

    }

    CommonService.modifyUserInfo(params)

    .then(() => {

    this.queryUserInfo()

    this.btn = {

    id: 0,

    value: '修改'

    }

    this.writeable = true

    })

    .catch()

    }

    }

```

后端代码

```

class ModifyUserInfo(restful.Resource):

    def post(self):

        userId = request.args.get('userId')

        alipay = request.args.get('alipay')

        stress = request.args.get('stress')

        loginName = request.args.get('loginName')

```

```

result = user.modify(alipay, loginName, stress, userId)

return result

def modify(alipay, loginName, stress, userId):

    code = 10000

    sql = "update user set alipay='%s', loginName='%s', stress='%s' where userId='%s';" % (alipay,
loginName, stress, userId)

    cursor = g.db.cursor()

    cursor.execute(sql)

    data = cursor.fetchone()

    g.db.commit()

    if (data == None):

        result = 'modify success!'

        return {'code': code, 'message': result}

    return {'code': code, 'message': 'fail'}

```

5.2.4 后台统计模块

作为电商类系统，按各种维度统计用户信息及系统信息很有必要。本模块选用百度开源的 echarts 框架进行数据可视化。并针对不同的信息类型应用饼图、直方图、折线图来展示数据。

(1) 商品分类统计

将用户每月发布的二手物品进行分类统计分析，可以有针对性的帮助平台运营。分类统计图如图 5.6 所示。

图5.6 分类统计

相关代码如下所示：

```

# 统计商品表分类信息sql语句

sql = "select attrCatId count(*) from goods group by attrCatId"

// 统计商品表分类信息应用echarts框架代码

let attrChart = this.$refs.attr

this.attrChart = echarts.init(attrChart)

var data = attrChartData().arrtOption

```



```

data.series[0].data = this.attrList

this.attrChart.setOption(data)

let userChart = this.$refs.user

this.userChart = echarts.init(userChart)

var userData = userChartData().userData

userData.xAxis.data = this.buyData

userData.series[0].data = this.userDataX

this.userChart.setOption(userData)

let buyChart = this.$refs.buy

this.buyChart = echarts.init(buyChart)

var buyData = buyChartData().buyDataX

buyData.xAxis.data = this.buyData

buyData.series[0].data = this.buyDataY

this.buyChart.setOption(buyData)

```

(2) 交易统计

按月统计用户交易量，如图5.7所示

图5.7 交易统计

实现过程中仅sql语句和统计图类型与分类统计图不一致，前端代码不再做重复说明，相关代码如下：

```

# 交易统计信息sql语句

select createNy, count(*) from goods_order group by createNy;

```

(3) 注册统计

按月统计用户交易量，如图5.8所示

图5.8 注册统计

实现过程中仅sql语句和统计图类型与分类统计图不一致，前端代码不再做重复说明，相关代码如下：

```

# 注册统计信息sql语句

```

```
select createNy, count(*) from user group by createNy;
```

5.2.5 用户管理模块

(1) 禁用账户

用管理员账户登录后台，可以通过手机号码检索用户，检索成功后可以对该账号进行禁用。相关代码如下所示：

```
def controlUser(userId, userStatus):  
  
    sql = "update user set userStatus='%s' where userId='%s';" % (userStatus, userId)  
  
    cursor = g.db.cursor()  
  
    cursor.execute(sql)  
  
    data = cursor.fetchone()  
  
    g.db.commit()  
  
    if data:  
  
        return True  
  
    return False
```

页面效果如图5.9所示。

图5.9 禁用用户

(2) 启用账户

用管理员账户登录后台，可以查看被禁用用户列表，并选择启用某一个或多个用户。相关代码如下所示：

```
def illegalAccounts():  
  
    sql = "select * from user where userStatus=0;"  
  
    cursor = g.db.cursor()  
  
    cursor.execute(sql)  
  
    rows = cursor.fetchall()  
  
    result = []  
  
    for row in rows:  
  
        result.append({ 'loginName': row[1], 'userPhone': row[0] })  
  
    return {'result': result}
```

页面效果如图5.10所示。

图5.10 启用用户

5.2.6 浏览商品模块

浏览商品模块是本系统的默认首页，无论用户是否登录系统，都可以访问。相关代码如下所示：

```
def getGoodsList(pageNum, pageSize):

    pageNum = int(pageNum)

    pageSize = int(pageSize)

    start = pageSize * (pageNum - 1)

    sql = "select * from goods where isSale=0 order by createTime desc limit %s, %s;" % (start,
    pageSize)

    cursor = g.db.cursor()

    cursor.execute(sql)

    rows = cursor.fetchall()

    result = []

    for row in rows:

        result.append({'goodsId': row[0], 'ownerId': row[1], 'goodsName': row[4], 'ownerName': row[2],
        'currentPrice': row[9], 'originalPrice': row[10], 'goodsImg': row[3], 'goodsThums': row[11] })

    return result

// 前端代码

// 点击“加载更多”所绑定的函数

add () {

    this.params.pageNum++

    this.getList()

},

// 获取商品列表

getList () {

    CommonService.goodsList(this.params)

    .then(res => {
```

```
console.log(res)

this.goodsList = this.goodsList.concat(res)

})

.catch(err => {

console.log(err)

})

}
```

在分页功能的实现过程中，选择了更为直观、现代的分页方式。具体效果如图5.11所示。

图5.11 浏览商品？

6 系统测试

系统测试是指在开发工作完成后按照一定测试方法对系统的进行全方位、多角度评测。完善的系统测试是软件开发周期中必不可少的环节，良好的测试为系统质量提供了重要保证。常见的测试项包括正确性测试、容错性测试、接口测试、数据一致性测试、压力测试等。

6.1测试记录

本着真实、可信的原则，在开发工作完成后，系统对各项功能进行了全面的测试，在具体操作的时候，采用黑盒测试法，力求测试用例覆盖到用户最微小操作。详细测试结果如表6.1所示。

表6.1 测试详情

编号 测试项 输入 预期 输出

1 注册 输入错误手机号或不填注册信息 注册失败 提示手机号格式或相关信息填写不正确
输入错误验证码 注册失败 提示验证码错误
正确输入手机号、验证码，填写注册信息 注册成功 跳转到首页

2 登录 输入错误手机号 登录失败
提示手机号格式不正确
输入错误验证码 登录失败
提示验证码不正确
输入正确验证码后更改手机号 登录失败
提示验证码不正确
正确输入手机号和验证码 登录成功 跳转到首页

3 个人中心 未登录时点击 不能进入页面 跳转到登录页面

登陆后点击 进入相关页面 跳转到个人中心

4 商品浏览 点击“加载更多”按钮 如果数据库中还有数据，则再加载10项商品 与预期结果一致

5 购买商品 点击“立即购买”按钮 进入扫码支付页面，支付成功后生成订单 与预期结果一致

6 禁用账户 登录管理员账户，通过手机号码查询用户并将其禁用 该用户将不能再登录系统。使用被禁用账户登录网站，系统提示“该账号被禁用”

7 启用账户 在被禁用列表中可以选启用某一账户 解除禁用 该账号可正常登录

8 查看统计 登录管理员账户，进入后台管理页 可以看到交易统计、注册统计、商品分类统计 与预期结果一致

6.1测试结论

经过两轮完整的系统测试，完成了交付使用前系统开发的全部流程。在编写测试用例的时候也将各种边界情况考虑在内，系统均能做出正确反应。因篇幅有限，不能将所有细枝末节整理成表格。总而言之，测试的结果十分令人满意，系统不仅在功能上均能满足需求，而且在性能上有超出预期的表现

?

总 结

本次毕业设计主要是围绕高校中二手物品的再利用做信息化建设，在充足的走访调查之后，将师生们较为集中的需求都做到了细致入微的系统实现。具体步骤是先进行需求分析，然后做系统设计，再运用Python、Flask、mysql、javascript、html等web技术开发了一个校园二手物品交易系统，最后以论文的形式详细阐述本课题所解决的生产生活中的问题，以及论述为完成本次毕业设计所采用的技术、系统开发的主要工作等。

经过背景及研究意义的调查再到需求分析与系统设计，在充足的前期工作的基础上本系统实现了管理员用户对普通用户停用、启用控制，查看注册统计、分类统计、交易统计等管理员功能，以及普通用户的注册登陆、查看并修改个人信息、订单管理、浏览商城、在线购买等用户功能。

本系统优点：

- ① UI界面简洁友好；
- ② 使用短信验证码进行身份校验，安全性极高；
- ③ 图片上传使用第三方分布式资源部署平台，即使得系统不至于臃肿，又提高了系统运行速度；
- ④ 使用前后台分离架构，使系统具有多端可用性。

本系统缺点：

- ① 支持平台不全，没有适配移动端网页；
- ② 兼容性不佳，IE9以下版本会出现页布局混乱的情况。

总的说来，虽然系统存在一些缺点，但就易用性和功能完备性来说是完全符合设计需求的。互联网产品的每次进步都离不开用户参与，我相信只要虚心听取用户意见，系统一定可以在以后的迭代升级中变得更好。

参考文献

- [1] 李迎辉.Python开发Rails框架——Django框架介绍.程序员, 2006, (11) .
- [2] 胡能发.基于UML图书销售系统分析与设计[J]. 电脑知识与技术.2015(26).
- [3] Chun W J.Python核心编程[M].宋吉广,译.2版.北京:人民邮电出版社,2008.
- [4] 韩松. 软件工程：面向对象和传统的方法（原书第8版）[M].北京：机械工业出版社.2016.
- [5] 哲思社区.可爱的Python[M].北京:电子工业出版社,2009.
- [6] 鲁特兹.Python学习手册[M].3版.侯靖,译.北京:机械工业出版社,2009.
- [7] 刘瑞新. 网页设计与制作教程（HTML+CSS+JavaScript）[M].北京：机械工业出版社.2016.
- [8] 李辉. 数据库系统原理及MySQL应用教程[M].北京：机械工业出版社.2016.
- [9] 易剑波.基于MVVM模式的WEB前端框架的研究[J].信息与电脑:理论版, 2016 (19) :76-77+84.
- [10] 吕林涛, 万经华, 周红芳.基于AJAX的Web无刷新页面快速更新数据方法[J].计算机应用研究, 2006 (11) :199-200, 223.
- [11] 杜波依斯. MySQL技术内幕[M]. 第4版.人民邮电出版社. 2014.
- [12] 何玉洁. 数据库原理与应用教程 第4版[M].北京：机械工业出版社.2016.
- [13] 郑人杰. 软件工程概论 第2版[M].北京：机械工业出版社.2016.
- [14] Miguel Grinberg.Flask Web Development[M].2nd ed.USA:O'Reilly Media Inc,2014:7-17
- [15] Bruce Eckel,Bert Bates.Chinese Journal of Computers .20016 (10) ：80-83
- [16] Bryan Basham, Bert Bates.Chinese Journal of Computers . 20014 (5) ：27-30
- [17] 高静, 段会川.JSON数据传输效率研究[J].计算机工程与设计, 2011 (7) :2267-2270.
- [18] 叶锋.Python最新Web编程框架Flask研究[J].电脑编程技巧与维护.2015(15)
- [19] 李程慧.基于开源框架Flask快速构建实验室预约系统[J].电脑知识与技术.2016(10)
- [20] 张海藩.软件工程导论 第四版. [M]. 北京：清华大学出版社,2015.

致谢

整个毕业设计的完成，离不开张靖老师的精心指导，不管是初期的需求分析，还是中期系统的设计，又或是最后论文的编写，张老师都耐心的给予我指导和意见。在这个过程中，我的专业技能和文档的编写都得到了一定程度的提高。这也使得我感受到张老师职业精神和责任感。我很庆幸我遇

到这样一位导师。在此，我对张靖教授表示诚挚的感谢以及真心的祝福。同时还要感谢数学与计算机学院的领导们，体谅我们这些在外实习的学生，能后在校外完成毕业设计。感谢攀枝花学院给了我一个快乐而又难忘的大学生活。