

# 基于 Node.js 的艺术品销售平台设计与实现

专    业：网络工程

学    号：201310420106

学    生：周洋

指导教师：张君雁（校内）

**摘要：**本文论述了一个基于 Node.js 技术和 MySQL 数据库结合的艺术品销售平台，系统主要功能包括用户登录注册、商品搜索查看、商品结算等。选用基于 Node.js 的 Express 框架作为 Web 应用开发框架，使用纯前端技术完成系统开发，并给出了部分实现代码。该系统为使用者提供一个全新的在线购物平台，交互功能强，能满足大多数用户对网上购物的使用需求，有较好的应用前景。

通过分析各种类型在线购物平台的特点、现状以及发展趋势，本文提出了以艺术品 B2C 购物为背景的在线购物平台，并且已经实现了各个关键模块的具体功能。从功能上看，本系统主要为消费者提供登录注册、商品搜索查看、购物车管理、个人信息资料修改等。本系统基于浏览器/服务器模型，采用 JavaScript 技术和 Node.js 的 Express 框架，后台数据库系统选用 MySQL，采用纯前端技术实现在线艺术品销售系统，简洁美观，可用性强。

**关键词：**艺术品销售平台；Node.js；MySQL

# Design and Implementation of Artwork Sales Platform Based on Node.js

Specialty: Network Engineering

Student Number: 201310420106

Student: Zhou Yang

Supervisor: Zhang Junyan(School)

**Abstract:** This article discusses a combination of Node.js technology and MySQL database of art sales platform, the system's main functions including user login, user registration, product search and view, commodity settlement and so on. This system choose the Express framework based on Node.js as the framework of Web application development, using pure front-end technology to complete the system development, and give some implementation code. The system provides users with a new online shopping platform, interactive features, to meet the majority of users on the use of online shopping needs, have a better application prospects.

By analyzing the characteristics, current situation and development trend of various types of online shopping platform, this paper puts forward the online shopping platform with B2C shopping as the background, and has realized the specific function of each key module. From the functional point of view, the system mainly for consumers to provide registration, product search and browsing, shopping cart management, personal information and other changes. The system is based on the browser / server model, using JavaScript technology and Node.js Express framework, the background database system using MySQL, using pure front-end technology to achieve online art sales system, simple and beautiful.

**Key words:** Art sales platform; Node.js; MySQL

## 目录

|                                    |    |
|------------------------------------|----|
| 绪 论.....                           | 1  |
| 1 相关技术与工具研究.....                   | 3  |
| 1.1 Sublime Text 和 Node.js 概述..... | 3  |
| 1.2 Express 框架和 EJS 模板.....        | 4  |
| 1.3 MySQL 数据库与 Ajax 技术.....        | 5  |
| 2 艺术品销售平台系统分析.....                 | 7  |
| 2.1 系统需求分析.....                    | 7  |
| 2.2 系统性能分析.....                    | 8  |
| 2.3 系统流程分析.....                    | 9  |
| 2.4 系统功能分析.....                    | 9  |
| 3 数据库设计与实现.....                    | 12 |
| 3.1 数据库设计.....                     | 12 |
| 3.1.1 E-R 图.....                   | 12 |
| 3.1.2 表基本设计.....                   | 15 |
| 3.2 数据库结构设计.....                   | 16 |
| 4 系统详细设计.....                      | 20 |
| 4.1 数据访问的实现.....                   | 20 |
| 4.2 功能实现.....                      | 21 |
| 4.2.1 登录注册模块.....                  | 21 |
| 4.2.2 商品查询及浏览模块.....               | 24 |
| 4.2.3 购物车模块.....                   | 26 |
| 5 系统实现与测试.....                     | 28 |
| 5.1 网站的实现.....                     | 28 |
| 5.2 测试.....                        | 33 |
| 5.2.1 测试计划.....                    | 33 |
| 5.2.2 详细测试内容.....                  | 34 |
| 5.2.3 测试执行及效果情况.....               | 36 |
| 结 论.....                           | 37 |
| 参考文献.....                          | 38 |
| 致 谢.....                           | 39 |

## 绪 论

### （1）艺术品销售平台开发背景及意义

随着生活水平的不断提高，人们对精神方面的需求也随之越来越大，文化产业在中国会是大势所趋，从当前电影市场的火爆场面便可见一斑，而艺术品交易还是一个没有被过多颠覆的领域，艺术本源于生活，却少有融于你我所处一方世界。

传统艺术品的销售渠道无非包括这几种，一：拍卖行拍卖；二：线下实体店销售；三：直接从艺术家手中收购；第一种方式虽说质量以及真伪得到了保证，但是竞拍者的不断出价导致艺术品价格直线升高，造成普通民众只能望而却步，难以触及艺术品这一块。第二种的话虽说价格不像前者那么离谱，但是艺术品质量、真伪参差不齐，即使找藏品鉴定专家说不定也遇到托，最后被别人坑了还在替人数钱。最后一种虽说价格合理，质量、真伪有保障，但是需要的是与艺术家良好的人缘与关系。在这种情势下急需要有一种既能让普通民众能够触及，又能够保证艺术品真伪，还能对售后服务提供保障的方式，因此艺术品在线销售平台应运而生<sup>[1]</sup>。

针对用户有消费的欲望而艺术家的作品又难以传播的现状，加之如今互联网的发展已经十分成熟了。大可通过互联网为艺术品需求方和艺术品提供方构架起的交易桥梁，将创作者（艺术品所有者）和收藏者直接连接起来，把具有消费、投资属性的艺术品销售给目标用户。做只做艺术品并且只做装饰版画这一板块，专注专一专业，区别于淘宝类电商似的大杂烩，以垂直艺术品电商的思路切入行业，结合互联网社交属性，为绘画艺术品创作者和需求者提供一个交流、交易电商平台，以平台为保障，为用户提供良好的售后服务，让用户更加放心<sup>[2]</sup>。

### （2）国内外研究现状

随着改革开放的不断推进和我国经济的快速发展，国内艺术品市场呈现出了繁荣景象。以艺术品拍卖市场为案例，截止到 2000 年为止，国内艺术品拍卖市场整体交易额达到 12.5 亿元，截止到 2010 年，整体交易额的数字已然达到 573 亿元人民币。大量画廊、拍卖公司、艺术网站应运而生。艺术品市场变得如此繁荣的原因有如下几点：首先是国家经济高速发展带来了市场的繁荣；其次人们对精神文化类产品的需求随着生活质量的提高也愈加旺盛；最后艺术品的快速升值带来投资热，投资热又驱动艺术品的快速升值。与此同时，原有的传统传播模式受互联网发展的推动受到了颠覆。而艺术品整合

营销伴随着互联网技术的普及和电子商务的发展，也开始走到线上。20 世纪 90 年代后期，国内最早一批艺术品电商成立，“嘉德在线”等成为标杆；2005 至 2009 年间，雅昌艺术网、赵涌在线、盛世收藏、博宝网等先后上线；2011 年淘宝开始艺术品在线拍卖，这标志着我国综合类电商开始正式进军艺术品在线交易市场；2013 年起，中国艺术品电商进入快速发展时期，一大批艺术在线交易平台涌现，呈现出爆发式增长。

在连续几年迅速增长、创下了无数拍卖纪录之后，全球拍卖市场正逐渐紧缩。artnet 网站的数据统计显示艺术市场在 2014-2015 年间经历了大幅的下跌，数据显示，市场规模从 163 亿美元下跌到了 2015 年的 148 亿美元。但与近期报道相反的是，国外艺术电商市场在同一时期却看起来一往无前，呈现出了 10% 的年度增长，艺术品拍卖总额高达 65 亿美元，占据全球艺术品拍卖市场总量的 40%<sup>[3]</sup>。

### （3）主要研究内容

本文采用 JavaScript+Node.js+MySQL 技术实现整个电商平台，以操作简便、界面美观、灵活实用三个方面为开发目标。在功能上将整个项目划分为五个模块：登录注册模块、商品查询及浏览模块、购物车模块、商品结算模块、个人中心模块。

登录注册模块主要是用户的注册、登录以及购物流程的实现，通过调用第三方短信验证接口实现注册功能，防止竞争对手恶意注册造成服务器崩溃。其中，用户需要完成注册并且登录之后方可进行个人信息的修改、购买商品以及查看或取消订单。登录注册采用实时数据库查询的方式，判定是否已注册从而对应不同的功能，简化用户操作。

商品查询及浏览模块具体实现了商品的罗列、商品详细信息的浏览以及商品查询等功能。本系统考虑到用户实用性，增加多种数据筛选方式（尺寸、颜色等），供用户筛选出理想的商品。由于这是一个艺术品版画的电商网站，于是提供了多种家庭生活背景，用户在详情页可以拖动艺术品来查看是否适合家庭装修风格。

购物车模块主要是实现商品选购、加入购物车、移除购物车物品、清空购物车等。

商品结算模块是用户选好商品后去结算到交易完成之间的过程。本系统采用模拟交易的方式实现交易过程<sup>[4]</sup>。

个人中心模块中用户能够浏览已经成交的订单，能够修改个人信息以及头像，还能修改登录密码等。

在完成以上基础知识的基础上，本文主要分析了系统设计的两个关键步骤：概要设计和详细设计。概要设计中分为数据库管理操作设计和客户端操作设计。详细设计主要阐述了登录注册、商品模糊查询等功能模块的实现，并给出了相应的系统设计代码。

## 1 相关技术与工具研究

软件的开发离不开技术和工具的支撑。本章将对本艺术品销售平台开发过程中使用到的技术和工具进行详细介绍，通过对其深入研究，为之后系统设计做好知识准备。

### 1.1 Sublime Text 和 Node.js 概述

#### （1）代码编辑器 Sublime Text

Sublime Text 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。由程序员 Jon Skinner 在 2008 年 1 月份开发出来，开发最初被设计为一个具有丰富扩展功能的 Vim。

Sublime Text 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。该编辑器在界面上比较有特色的是能够实现支持多种布局和显示代码缩略图，通过右侧的文件略缩图滑动条，可以很方便快捷地观察当前窗口在文件的那个位置。

Sublime Text 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）Node.js 概述

Node.js 是一个在 Google V8 引擎上进行了封装的 JavaScript 运行环境。由于 V8 引擎执行 JavaScript 的速度非常快，性能非常好，Node.js 在其基础上对一些特殊用例进行了优化，提供了替代的 API，使得 V8 在非浏览器环境下运行得更加流畅。

Node.js 是一个基于 Chrome 的 JavaScript 运行时建立的平台，用于快捷、方便地搭建响应速度快、易于扩展的网络应用。由于 Node.js 使用事件驱动，非阻塞 I/O 模型，使得其轻量、高效，非常适合在分布式设备上运行数据密集型的实时应用，并且 Node.js 的包管理器 npm，是全球最大的开源库生态系统，丰富的插件使得用户开发起来更加容易<sup>[5]</sup>。

本系统则采用 Node.js 的 MySQL 模块用于与数据库的交互，采用 Node.js 的 Express 框架用做前后台交互的中间件。

## 1.2 Express 框架和 EJS 模板

### （1）Express 框架简述

Express 是一个基于 Node.js 平台，快速、开放、极简的 Web 开发框架。提供了一系列强大特性用以帮助创建各种 Web 应用，和丰富的 HTTP 工具，使用 Express 可以快速地搭建一个完整功能的网站<sup>[6]</sup>。

Express 框架核心特性：

- ① 可以设置中间件用来响应 HTTP 请求。
- ② 定义了路由表用于对应不同的 HTTP 请求动作。
- ③ 可以通过向模板传递参数来动态渲染 HTML 页面。

### （2）EJS 模板引擎技术

EJS 是一个简单高效的 JavaScript 模板语言。通过数据和模板，可以生成 HTML 标记文本。可以说 EJS 是一个 JavaScript 库，可以同时运行在客户端和服务端，在客户端安装只需要直接引入文件即可，在服务器端可以通过 npm 包安装。

EJS 模板特点：

- ① 快速编译和渲染
- ② 简单的模板标签
- ③ 自定义标记分隔符
- ④ 支持文本包含
- ⑤ 支持浏览器端和服务端
- ⑥ 模板静态缓存
- ⑦ 支持 express 视图系统

EJS 语法和功能：

- ② 缓存功能，能够缓存已经解析好的 Html 模版；
- ② `<% code %>`用于执行其中 JavaScript 代码；
- ③ `<%= code %>`会对 code 进行 Html 转义；
- ④ `<%- code %>`将不会进行转义，这一行代码不会执行，像是被注释了一样，然后显示原来的 html。也不会影响整个页面的执行；
- ⑤ 支持自定义标签，比如'`<%`'可以使用'`{ { , '%>'用 } }'`代替；
- ⑥ 利用`<%- include filename %>`加载其他页面模版；

### 1.3 MySQL 数据库与 Ajax 技术

#### （1）MySQL 数据库与 SQL 语言

MySQL 是目前最流行的关系型数据库管理系统，由瑞典 MySQL AB 公司开发。由于关系数据库将数据保存在不同的表中，而不是将所有数据存放在一个大仓库内，这就增加了其运行速度并提高了灵活性。

SQL 语言是一门高级的非过程化编程语言，允许用户在高层数据结构上工作。不要用户指定对数据的存放方法，也不需要用户了解具体的数据存放方式，所以具有完全不同底层结构的不同数据库系统，可以使用相同的结构化查询语言作为数据输入与管理的接口。结构化查询语言语句可以嵌套，这使得具有极大的灵活性和强大的功能。

#### SQL 语言特点

- ① SQL 语言集数据查询、数据操纵、数据定义和数据控制于一体；
- ② 面向集合的语言；
- ③ 非过程语言；
- ④ 类似自然语言，简洁易用；
- ⑤ 自含式语言，又是嵌入式语言，可独立使用，也可嵌入到宿主语言中；

#### （2）Ajax

Ajax 即 “Asynchronous Javascript And XML”（异步 JavaScript 和 XML），指的是一种用来创建交互式网页应用的网页开发技术。Ajax 通过在后台与服务器进行少量数据交换，使得这种技术可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

Ajax 是一种用于创建速度更快同时交互性更强的 Web 应用程序技术，并不是一门新的编程语言。使用 Javascript 向服务器提出请求并处理响应但此过程并不会阻塞用户，其核心对象为 XMLHttpRequest。通过这个对象，JavaScript 可在不重载页面的情况与 Web 服务器交换数据，即在不需要刷新页面的情况下，就可以产生局部刷新的效果。Ajax 在浏览器与 Web 服务器之间使用异步数据传输（HTTP 请求），这样就可使网页从服务器请求少量的信息，而不是整个页面。其原理是在会话的开始时候，浏览器加载一个 Ajax 引擎，这个引擎采用 JavaScript 编写并且通常在一个隐藏的 frame 中，其负责绘制用户界面以及与服务端通讯。

目前，Ajax 技术已经被广泛视为现代网站的核心技术之一，在本系统中登录注册处采用 Ajax 技术，使得用户可以瞬时了解自己是否注册，免除先提交后判断的烦恼，提



升用户体验，并且在本系统中凡是与后端有交互的页面均采用了 Ajax 技术。

本章介绍了本艺术品销售平台将要使用到的工具和技术。主要从以下几个方面进行了介绍：开发工具 Sublime Text；数据库系统 MySQL；脚本语言 JavaScript；模板引擎 EJS；前后端中间件 Node.js 及其 Express 框架；数据通信方式 Ajax。总之，本章为后续章节的分析、设计奠定了坚实的基础。

## 2 艺术品销售平台系统分析

需求分析是软件开发的重要基础，其结论直接影响了软件的功能是否完善，直接决定了软件的开发是否成功。

本章主要从网上购物的需求作为出发点，详细阐述了需要实现的功能和流程。

### 2.1 系统需求分析

按照软件工程理论，系统的需求分析和设计是项目开发成功与否的关键因素，而其中需求分析优势又是重中之重。所谓的"需求分析"，指的是对将要解决的问题进行详细的分析，弄清楚问题的要求是什么，最后应得到什么结果。本系统致力于提供良好的艺术品展示，以及艺术品销售为核心的服务，将自己艺术品展现在客户眼中，使客户通过在线网站便可轻松自由的选购自己理想的艺术品画作。

由于现如今传统艺术品选购存在着一系列的问题，所以本系统以如何在最大限度上降低传统购物带来的劣势为出发点，提出了如下几个基本的需求：

用户需求：

- （1）用户需求最大的是艺术品的查询，购买以及个人信息的修改；
- （2）用户可以根据个人喜好和需求添加、删除选购的艺术品；
- （3）用户使用系统要实现快捷、方便的在线选购服务；

艺术品清算需求：

- （1）填写或者选择个人收货地址；
- （2）确认购买物品及价格；
- （3）选择合适的付款方式；

购物车需求：

- （1）浏览当前购物车中的艺术品；
- （2）清空购物车中的艺术品；
- （3）结算当前选购的物品；

简化用户的操作需求：

本系统简化了选购流程，用户只要简单的三步即可买到心仪的商品，即提升了用户的体验，又加大了系统的运行效率；为了增强与用户之间的交流沟通，在主页上以侧边提示的方式留下留言框，以便于用户建议同时便于做出更好的改进；为了方便用户快速

地找到自己想要的艺术品,既提供了多种筛选条件,又提供了高级搜索功能,在提升用户体验的同时也减轻了服务器的压力,一举两得;

总而言之,本节对本艺术品销售平台进行了详细的需求分析,为之后的系统流程分析打下坚实的基础。

## 2.2 系统性能分析

### (1) 可行性分析

技术可行性:本系统采用 MVC 设计模式,使用 Node.js 的 Express 框架进行开发,前台 HTML 页面利用原生 JavaScript 处理页面脚本,使开发更加高效,提示信息更加完善,界面更加友好。后台选用 MySQL 数据库,其高效小巧的优势满足了系统使用时的性能需求。

操作可行性:本系统主要针对的是热爱网络购物的消费人群,其系统的操作复杂度相比其他需要具备复杂的电脑专业知识的电商网站而言是十分简单的,不需要使用者的专业知识过硬,而是会简单的电脑操作就能够很好的操作使用本系统。本网站的简单操作和可用性恰巧说明了这套系统的操作是可行的。

经济可行性:现实的销售经营过程中,供应商和顾客只能通过上门咨询、电话沟通等形式来实现各种产品信息的传递获取,导致产品的宣传受到限制,并且时间与空间的局限性严重影响了产品的销售,在无形中也提升了产品的销售成本。使用本系统可以花更少的时间和金钱来建立企业商务网络,以此来使企业和消费者之间的经济活动变得更加灵便、主动,扭转这种现状。从成本可行性分析来看,该系统充分展现了将产品利益最大化的企业原则<sup>[7]</sup>。

社会可行性:互联网的普及使得传统购物的诸多缺点暴露出来,人们更倾向于利用零碎的时间通过电脑及手机等设备就能完成网上购物,所以本系统具有社会可行性。

### (2) 系统稳定性

同一时间可能会有多个用户访问系统,若访问数量比较少,则系统稳定性方面出问题的概率较低,若访问量过大,则可能造成系统崩溃等问题。所以 Web 系统应对系统的稳定性作分析,通过对代码、访问模式、资源的请求与释放等方面的优化来达到改善系统稳定性的目的。

### (3) 数据安全性

对于用户密码等敏感信息的处理方式尤为重要。对于用户的权限,应该严格控制,禁止跨权限操作<sup>[8]</sup>。

## 2.3 系统流程分析

良好的系统流程分析，有助于更加清晰明了的了解本系统的工作流程以及功能需求，还能从用户的角度去理解如何设计的更好。

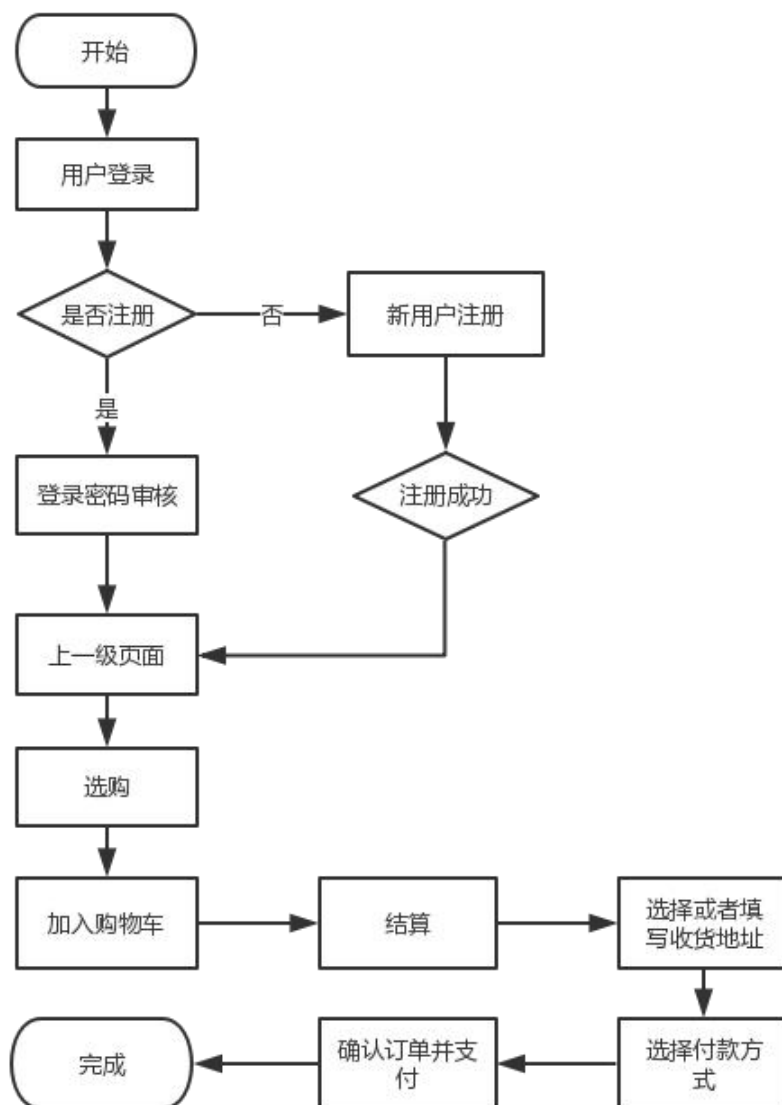


图 2-1 系统流程分析图

本系统的流程分析如图 2-1 所示。当用户执行加入购物车操作时需要登录权限，跳转到登录注册页面，根据用户名来判断是否已注册，在登录或者注册成功后返回上一级页面，加入商品进购物车，选择收货地址，完成结算。

## 2.4 系统功能分析

如果将系统的流程分析比作躯干，那么系统功能分析则是心脏部位，将引导整个系统的运转。软件的功能模块划分指的是在软件设计过程中，为了实现对系统开发流程的过程进行管理，确保系统的稳定可靠以及后期时候的可维护性，从而按照一定的准则对

软件开发进行模块的划分。由于在系统开发过程中，有些需求在功能属性上会存在相应的关联性，但是有些需求之间的联系又很少。如果在设计的时候，不对需求进行归类划分的话，在后期的开发过程中很容易造成混乱<sup>[9]</sup>。

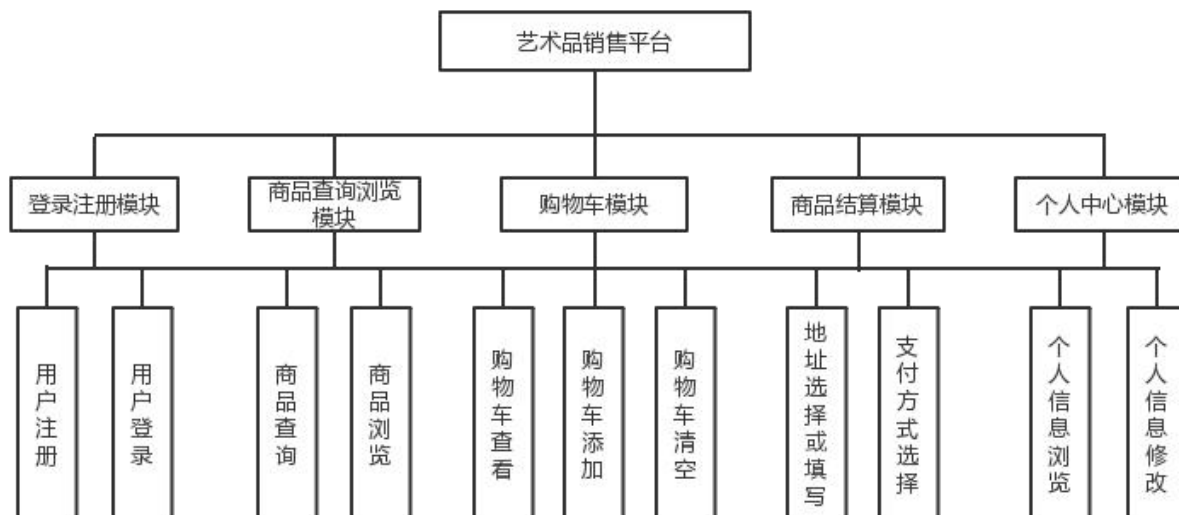


图 2-2 功能结构图

在本系统中将艺术品销售平台划分为以下几个模块：登录注册模块、商品查询及浏览模块、购物车模块、商品结算模块、个人中心模块。系统功能结构图如 2-2 所示。

#### （1）登录注册模块功能

登录注册作为电商网站不可或缺的一部分，自然而然是非常重要的。考虑到用户的习惯以及体验，本系统采用手机号作为注册账号，采用实时查询的方式判断用户是否已经注册，摒弃了让用户选择登录还是注册的传统方式。考虑到安全性以及数据库的稳定性，本系统调用第三方短信验证平台 LeanCloud 来对用户注册做验证。

#### （2）商品查询浏览模块

查询：用户通过顶部导航栏可以选择以艺术家类来查看艺术品，亦或者直接进入艺术品商城里面，根据提供的筛选条件进行艺术品筛选，如果依然不满足需求还可以通过搜索框借助模糊搜索进行查找。

浏览：当用户找到理想的商品后可以查看商品的具体信息，由于艺术品受空间等因素的影响，本系统还提供了多种生活背景和相框供用户参考，然后根据商品的清晰图片信息以及详细的文字描述和价格判断是否进行登录注册然后再去购买。不管用户是否注册登录，都让他们对本网站的商品有了一定的了解。

### （3）购物车模块

购物车作为贯穿电商平台整个流程的模块，重要程度不言而喻。好比一根线连接起整个购物的过程。

购物车查看：购物车里面具体应该有什么商品，应该让用户看了一目了然，由于本系统是艺术品商城，因此不涉及到数量的增减。

购物车添加和清空：当用户选中商品后添加进购物车，然后在购物车中还可以进行删除或者是清空的修改。

### （4）商品结算模块

用户在确认好购买商品后进入结算页面，选择预先设置好的地址或者是填写新的地址信息，然后选择支付方式进行支付。受其他因素影响，目前本系统的支付采用模拟的方式进行。

### （5）个人中心模块

个人中心模块主要实现对自己信息的修改，对订单、收货地址等信息的查询以及修改等。

本章完成了对艺术品销售平台的需求分析。主要从技术可行性、功能、流程三个方面对其进行了阐述，对系统的功能模块划分及要求有了清晰的认识。通过对整个系统的分析和设计，为接下来系统的功能实现奠定了坚实的基础。

### 3 数据库设计与实现

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。

#### 3.1 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。

##### 3.1.1 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<sup>[10]</sup>。

（1）用户实体属性主要包括：

用户 id、用户名、用户电话、用户密码、用户性别、用户头像 id。如图 3-1 所示。

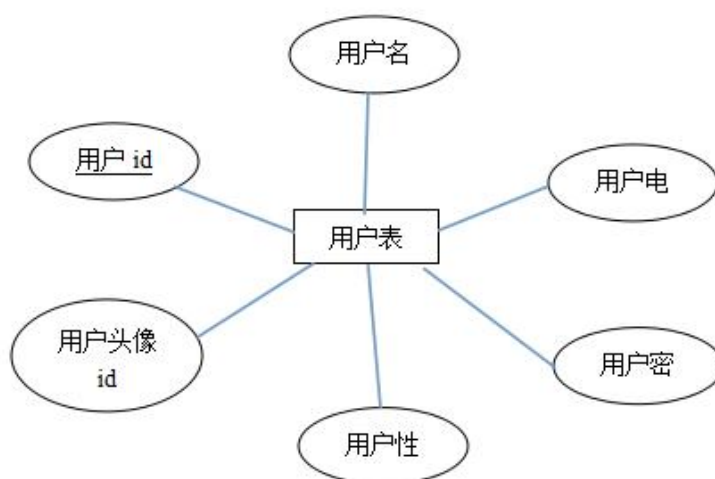


图 3-1 用户表 E-R 图

（2）头像实体属性包括：

用户头像 id、头像路径。如图 3-2 所示



图 3-2 用户头像表 E-R 图

（3）地址实体属性包括：

用户地址 id、用户地址、用户 id、收货人电话、收货人姓名。如图 3-3 所示。

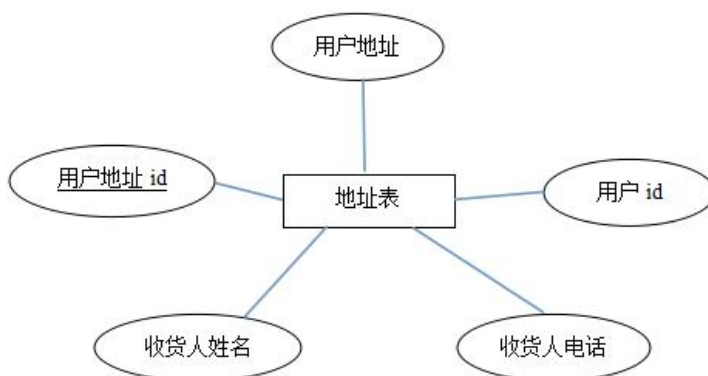


图 3-3 地址表 E-R 图

（4）商品实体属性包括：

商品 id、商品名、艺术家 id、商品宽、商品高、商品价格、商品色系 id、所属空间 id、装修风格 id、艺术分类 id、艺术风格 id、艺术主题 id、上架时间、商品状态 id、商品材质、商品路径。如图 3-4 所示



图 3-4 商品表 E-R 图



（5）艺术风格实体属性包括：

艺术风格 id、艺术风格名。如图 3-5 所示。

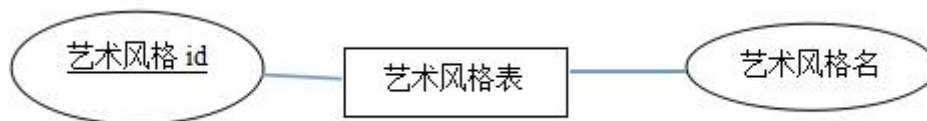


图 3-5 艺术风格表 E-R 图

（6）艺术家实体属性包括：

艺术家 id、艺术家名、职称、艺术家描述、艺术家头像路径。如图 3-6 所示。

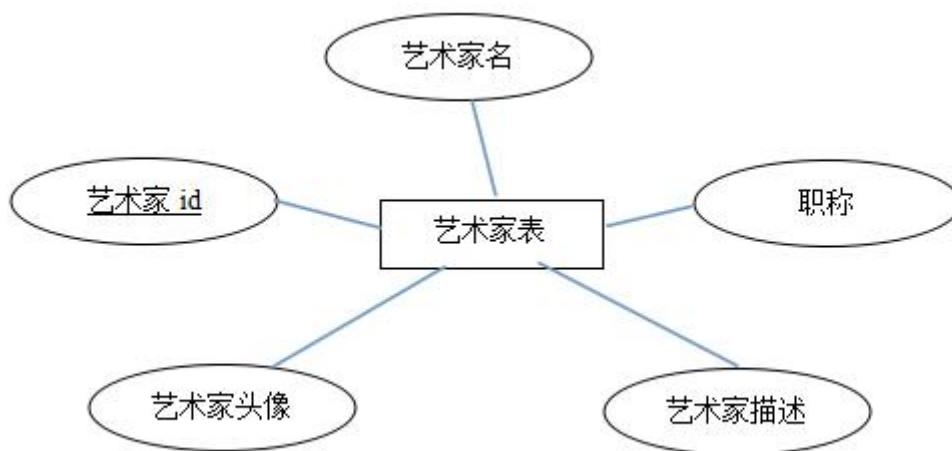


图 3-6 艺术家表 E-R 图

（7）商品细节实体属性包括：

商品细节描述 id、商品细节图、商品细节描述。如图 3-7 所示。

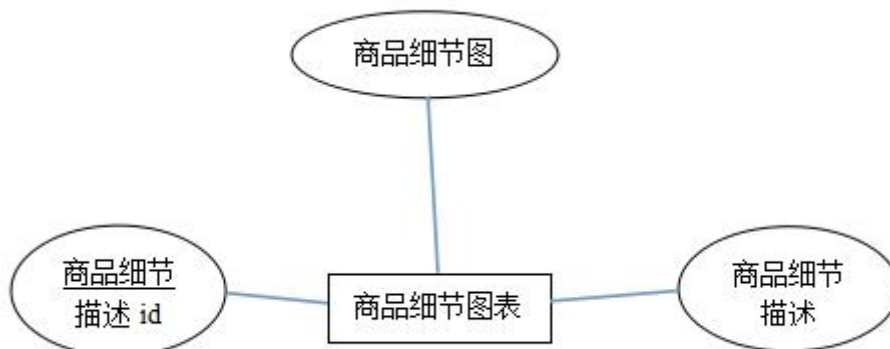


图 3-7 商品细节图表 E-R 图

（8）订单实体属性包括：

订单 id、用户 id、用户地址 id、订单状态、支付时间。如图 3-8 所示。

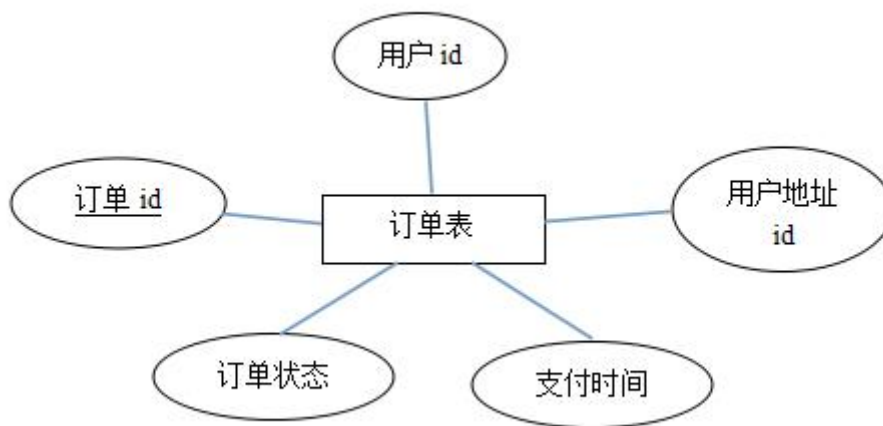


图 3-8 订单表 E-R 图

（9）购物车实体属性包括：

购物车 id、用户 id、商品 id、状态。如图 3-9 所示。

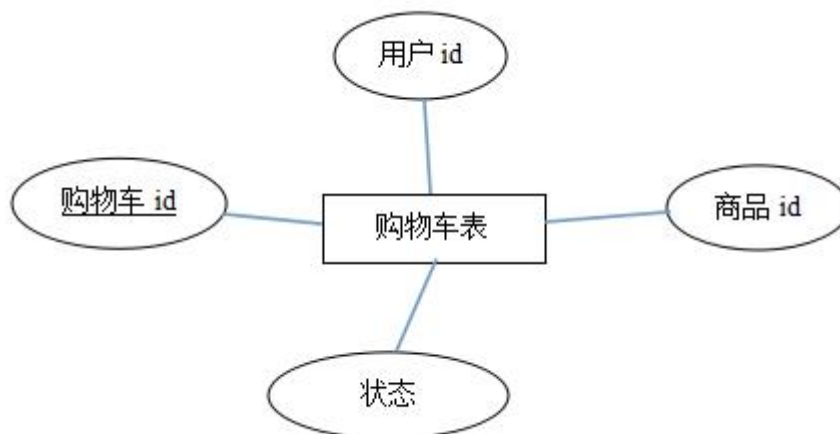


图 3-9 购物车表 E-R 图

### 3.1.2 表基本设计

在本系统数据库中，根据需求分析的结果建立以下基本表结构：

（1）用户表

t\_user (u\_id, usrename, tel, password, sex, sex)

（2）头像表

t\_head (h\_id, h\_imgsrc)

（3）地址表

t\_address (addressid, addressName, u\_id, addTel, addName)

## (4) 商品表

t\_goods (g\_id, g\_nam, artistid, g\_width, g\_height, price, colorid, spaceid, artClassid, artStyleid, artThemeid, startTime, g\_state, material, goodsSrc)

## (5) 艺术风格表

t\_artStyle (artStyleid, artStyleName)

## (6) 艺术家表

t\_artist (artistid, artistName, professional, describeOne, describeTwo, artistSrc)

## (7) 商品细节表

t\_detailsPicture (detailsid, detailsSrc, discription)

## (8) 订单表

t\_order (o\_id, u\_id, addressid, o\_state, payTime)

## (9) 购物车表

t\_shoppingCart (cart\_id, u\_id, g\_id, state)

## 3.2 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 (t\_user)，包括六个字段：u\_id（用户 id），username（用户名），tel（用户电话），password（用户密码）。其中 u\_id 为主键，具体设计见表 3-1。

表 3-1 t\_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   | u_id     | int         | Y  |    | N  | 自增长 |
| 用户名     | username | varchar(20) |    |    | N  |     |
| 用户电话    | tel      | char(11)    |    |    | Y  | 唯一  |
| 用户密码    | password | varchar(20) |    |    | Y  |     |
| 用户性别    | sex      | varchar(10) |    |    | N  |     |
| 用户头像 id | h_id     | int         |    | Y  |    |     |

## (2) 用户头像表

用户头像表 (t\_head)，包括两个个字段：h\_id（用户头像 id），h\_imgsrc（头像路径）具体设计见表 3-2。

表 3-2 t\_head（头像表）

| 中文名     | 字段名      | 数据类型        | 主键 | 外键 | 必填 | 备注  |
|---------|----------|-------------|----|----|----|-----|
| 用户头像 id | h_id     | int         | Y  |    | N  | 自增长 |
| 头像路径    | h_imgsrc | varchar(30) |    |    | N  |     |

### （3）用户地址表

用户地址表（t\_address），包括五个字段：addressid（用户地址 id），addressName（用户地址），u\_id（用户 id），addTel（收货人电话），addTel（收货人姓名）。其中 addressid 为主键，具体设计见表 3-3。

表 3-3 t\_address（地址表）

| 中文名     | 字段名         | 数据类型         | 主键 | 外键 | 必填 | 备注  |
|---------|-------------|--------------|----|----|----|-----|
| 用户地址 id | addressid   | int          | Y  |    | N  | 自增长 |
| 用户地址    | addressName | varchar(100) |    |    |    |     |
| 用户 id   | u_id        | int          |    | Y  |    |     |
| 收货人电话   | addTel      | char(11)     |    |    | Y  |     |
| 收货人姓名   | addName     | varchar(30)  |    |    | Y  |     |

### （4）商品表

商品表（t\_goods），包括十四个字段：g\_id（商品 id），g\_name（商品名），artistid（艺术家 id），g\_width（商品宽），g\_height（商品高），price（商品价格），colorid（商品色系 id），spaceid（所属空间 id），spaceStyleid（装修风格 id），artClassid（艺术分类 id），artStyleid（艺术风格 id），artThemeid（艺术主题 id），startTime（上架时间），g\_state（商品状态），material（商品材质），goodsSrc（商品路径）。其中 g\_id 为主键，具体设计见表 3-4。

表 3-4 t\_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   | g_id       | int          | Y  |    | N  | 自增长          |
| 商品名     | g_name     | varchar (20) |    |    | Y  |              |
| 艺术家 id  | artistid   | int          |    | Y  |    |              |
| 商品宽     | g_width    | varchar (20) |    |    | Y  |              |
| 商品高     | g_height   | varchar (20) |    |    | Y  |              |
| 商品价格    | price      | int          |    |    | Y  |              |
| 所属空间 id | spaceid    | int          |    | Y  |    |              |
| 艺术分类 id | artClassid | int          |    | Y  |    |              |
| 上架时间    | startTime  | varchar (20) |    |    | Y  |              |
| 商品状态 id | g_state    | int          |    |    |    | 0 已出售; 1 未出售 |
| 商品材质    | material   | varchar (10) |    |    | Y  |              |
| 商品路径    | goodsSrc   | varchar(30)  |    |    | Y  |              |

## (5) 艺术风格表

艺术风格表 (t\_artStyle)，包括两个字段：artStyleid（艺术风格 id），artStyleName（艺术风格名）。其中 artStyleid 为主键，具体设计见表 3-5。

表 3-5 t\_artStyle（艺术风格表）

| 中文名     | 字段名          | 数据类型         | 主键 | 外键 | 必填 | 备注  |
|---------|--------------|--------------|----|----|----|-----|
| 艺术风格 id | artStyleid   | int          | Y  |    | N  | 自增长 |
| 艺术风格名   | artStyleName | varchar (20) |    |    | Y  |     |

## (6) 艺术家表

艺术家表 (t\_artist)，包括六个字段：artistid（艺术家 id），artistName（艺术家名），professional（职称），describeOne（艺术家描述 1），describeTwo（艺术家描述 2），artistSrc（头像路径）。其中 artistid 为主键，具体设计见表 3-6。

表 3-6 t\_artist(艺术家表)

| 中文名    | 字段名          | 数据类型           | 主键 | 外键 | 必填 | 备注  |
|--------|--------------|----------------|----|----|----|-----|
| 艺术家 id | artistid     | int            | Y  |    | N  | 自增长 |
| 艺术家名   | artistName   | varchar (20)   |    |    | Y  |     |
| 职称     | professional | varchar (20)   |    |    | Y  |     |
| 艺术家描述  | describeOne  | varchar (1000) |    |    | Y  |     |
| 头像路径   | artistSrc    | Varchar(30)    |    |    | Y  |     |

## (7) 商品细节图表

商品细节图表(t\_detailsPicture)，包括三个字段：detailsid(细节描述 id)，detailsSrc(商品细节图)，discription(商品细节描述)。其中 detailsid 为主键，具体设计见表 3-7。

表 3-7 t\_detailsPicture(商品细节图表)

| 中文名     | 字段名         | 数据类型           | 主键 | 外键 | 必填 | 备注 |
|---------|-------------|----------------|----|----|----|----|
| 细节描述 id | detailsid   | int            |    | Y  |    |    |
| 商品细节图   | detailsSrc  | varchar (30)   |    |    | Y  |    |
| 商品细节描述  | discription | varchar (1000) |    |    | Y  |    |

## (8) 订单表

订单表(t\_order)，包括五个字段：o\_id(订单 id)，u\_id(用户 id)，addressid(用户地址 id)，o\_state(订单状态)，payTime(支付时间)。其中 o\_id 为主键，具体设计见表 3-8。

表 3-8 t\_order(订单表)

| 中文名     | 字段名       | 数据类型      | 主键 | 外键 | 必填 | 备注                         |
|---------|-----------|-----------|----|----|----|----------------------------|
| 订单 id   | o_id      | int       | Y  |    |    | 自增长                        |
| 用户 id   | u_id      | int       |    | Y  |    |                            |
| 用户地址 id | addressid | int       |    | Y  |    |                            |
| 订单状态    | o_state   | int       |    |    |    | 1 未支付;<br>2 已支付;<br>3 已完成; |
| 支付时间    | payTime   | timestamp |    |    | Y  | 自动写入支付<br>时间               |

## (9) 购物车表

购物车表(t\_shoppingCart)，包括四个字段：cart\_id(购物车 id)，u\_id(用户 id)，g\_id(商品 id)，state(状态)。其中 cart\_id 为主键，具体设计见表 3-9。

表 3-9 t\_shoppingCart(购物车表)

| 中文名    | 字段名     | 数据类型 | 主键 | 外键 | 必填 | 备注         |
|--------|---------|------|----|----|----|------------|
| 购物车 id | cart_id | int  | Y  |    | N  | 自增长        |
| 用户 id  | u_id    | int  |    | Y  |    |            |
| 商品 id  | g_id    | int  |    | Y  |    |            |
| 状态     | state   | int  |    |    |    | 0:取消; 1:勾选 |

## 4 系统详细设计

本文需要实现的是一套依赖纯前端技术实现的网上购物商城，因此系统的详细实现过程是另一个关键点。在本章中将主要介绍系统开发过程中使用的关键技术，包括前后端交互的中间件 Node.js、快速创建动态网页的 Ajax 技术、提高安全性能的验证码技术等。更加值得一提的是本系统通过 JavaScript 技术使得用户可以随意推动商品，并且动态切换画框，更加吸引用户眼球<sup>[11]</sup>。

### 4.1 数据访问的实现

数据访问为业务规则提供数据支撑，在本系统整个设计中数据访问依靠 MySQL 数据库实现，借助 Node.js 的 MySQL 模块，可以创建数据库连接并对数据库执行增删查改操作。本系统中用于数据库操作的各个成员函数功能主要包括：

#### （1）与数据库操作相关函数

```
const util = require("./util.js");           // 加载数据库操作函数
exports.answerAsk = function(req, resp) {};  // 首页留言操作
exports.goLogin = function(req, resp) {};    // 判断是否登录
exports.checkTel = function(req, resp) {};    // 检测用户是否存在
exports.login = function(req, resp) {};       // 登录
exports.checkC = function(req, resp) {};      // 检测用户名密码是否匹配
exports.showImg = function(req, resp) {};     // 查询商品信息
exports.pagetotal = function(req, resp) {};   // 查询总页数
exports.search = function(req, resp) {};      // 首页搜索
exports.getPaintPageTotal = function(req, resp) {}; // 获取搜索的艺术品分页总数
exports.getPaintResult = function(req, resp) {}; // 获取搜索的艺术品详情
exports.artistSearch = function(req, resp) {}; // 艺术家搜索
exports.getAritistResult = function(req, resp) {}; // 获取搜索的艺术家详情
exports.getAritistPageTotal = function(req, resp) {}; // 获取搜索的艺术家分页总数
```

#### （2）数据库连接类函数

```
const mysql = require("mysql"); // 请求加载 MySQL 模块
exports.connect = function(sql,param,callback){
    var db = mysql.createConnection({ // 创建数据库连接对象 createConnection
```

```

    host:"",      // 配置主机地址
    port:"",      // 配置端口号
    user:"",      // 数据库用户名
    password:"",  // 数据库密码
    database:""   // 数据库名称
});

db.connect();    // 打开数据库连接

db.query(sql,param,callback); // 通过 query 发送 SQL 语句,回调函数处理响应结果

db.end();        // 关闭数据库连接

```

## 4.2 功能实现

系统功能的实现主要是业务表示和外观层的设计与实现，本章中将根据功能的不同从三个功能模块说明系统功能的实现<sup>[12]</sup>。

### 4.2.1 登录注册模块

设计过程：用户点击用户名输入框输入用户名，系统识别到鼠标抬起后，将设置一个 200 毫秒的延迟然后开始进行数据库的查找，根据是否查找到相应数据从而决定显示注册页面还是登录页面，动态搜索减少用户操作频度，提升用户体验。在登录页面调用第三方平台 LeanCloud 短信验证，增加安全性。

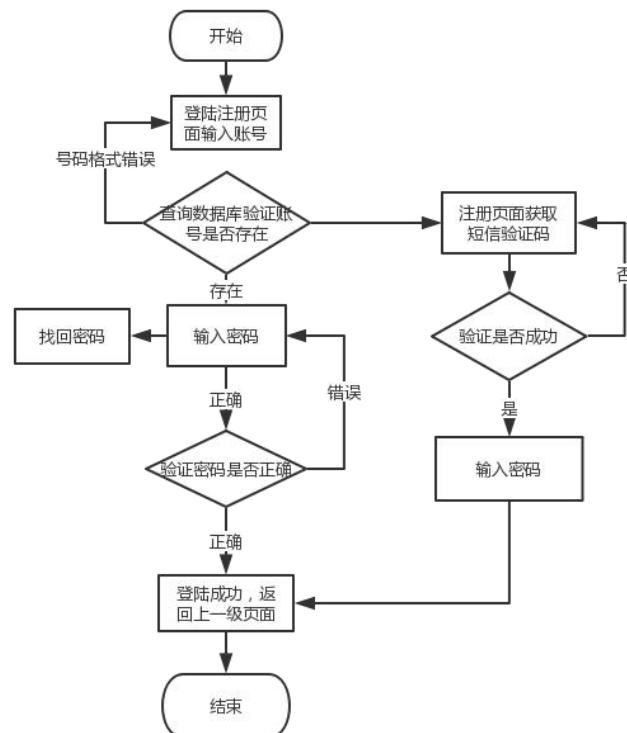


图 4-1 登录流程图



登录流程如图 4-1 所示。用户输入用户名，使用正则表达式进行验证，然后数据库查询用户是否存在，从而决定直接输入密码还是进行注册操作。在登录界面可以选择忘记密码进行密码重置，在注册界面需要输入手机验证码然后输入密码才能完成注册。

登录原型图如图 4-2 所示。用户输入账号，系统判定是否已注册来决定是否进入注册页面。

图 4-2 登录原型图

注册原型图如图 4-3 所示。系统判定数据库无此用户存在时提示用户获取验证码、输入密码完成注册。

图 4-3 注册原型图

相关核心代码：

```
function yanzheng(){ // 登录注册验证函数
    if(tel.match(a)){
        ajax(method,action,content,function(data){ // 判定用户是否存在
            if(data=="用户已存在"){countDown();} // 存在，显示密码输入框
```

```
        if(data=="新用户"){submit.value="注册";} // 不存在，调用注册界面
    }}}
    var APP_ID = "";           // LeanCloud ID 配置
    var APP_KEY = "";          // 应用 Key，用来校验权限
    AV.init({                   // 应用初始化
        appId: APP_ID,
        appKey: APP_KEY
    });
    var countdown=60; // 验证码倒计时，60 秒
    var timeout2;
    // LeanCloud 发送短信验证码
    function send(){
        AV.Cloud.requestSmsCode({ // 调用 api 请求发送短信验证码
            mobilePhoneNumber: telV, // 目标电话
            name: 'ArtLife',          // 短信头
            op: '注册验证',           // 短信功能说明
            ttl: 6                    // 验证码失效时间
        }).then(function() {}, function(err){});
    }
}
```

当用户名正则匹配通过之后会调用 Ajax 向后台验证用户的存在与否，根据返回的值来决定显示的内容。注册时调用 LeanCloud 的短信验证码，首先配置其 Key 和 ID，然后将手机号码作为参数调用其 RequestSmsCode 方法请求验证码。

注册页面调用了第三方平台 LeanCloud 的短信验证服务，用户输入手机号码将收到一条验证消息，输入验证码验证成功后才能输入用户密码信息。短信验证码的发送过程总共有三方参与：客户端、LeanCloud 和电信运营商。LeanCloud 短信验证码的发送、验证短信验证码的过程如图 4-4 所示。

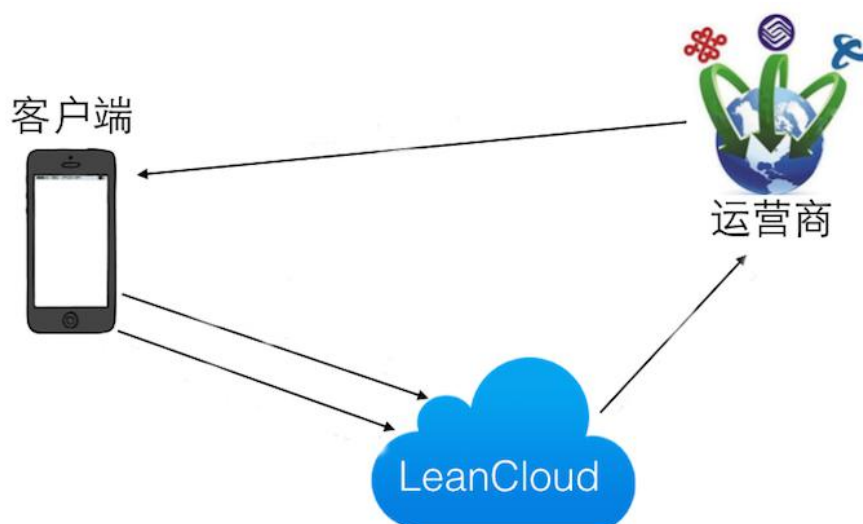


图 4-4 短信发送、验证流程

- （1）首先是客户端向 LeanCloud 发送请求，向特定手机号码发送验证码；
- （2）当 LeanCloud 云端收到请求后，生成验证码，将短信内容发送到运营商通道；
- （3）运营商根据收到的验证码和手机号下发短信（或语音）；
- （4）客户端输入收到的验证码，向 LeanCloud 验证手机号码及验证码的合法性。

#### 4.2.2 商品查询及浏览模块

设计过程：商品浏览模块采用 DIV+CSS 实现瀑布流式布局，使得显示效果更加引人入胜，在搜索方面浏览模块中既提供尺寸、颜色等搜索方式，又提供了搜索框实现模糊，借助 AJAX 技术使得页面搜索加载更快。在商品浏览模块中，采用 JavaScript 技术实现商品拖动以及动态添加画框，带给用户更加良好的体验和视觉感受。这个模块用户无需登录即可操作，若需要加入购物车或者购买则需要登录权限<sup>[13]</sup>。商品浏览原型图如图 4-5 所示<sup>[13]</sup>。



图 4-5 商品浏览原型图

相关核心代码：

```
var zhuangtai=false;
```

```
function myDrag(){ // 拖动函数
```

```
    var clickE=window.event||arguments[0]; // 获取 window.event 事件
```

```
    zhuangtai=true; // 初始化状态为 false
```

```
    var gensuiX=clickE.clientX-artBox.offsetLeft; // 鼠标按下点距盒子左边距离
```

```
    var gensuiY=clickE.clientY-artBox.offsetTop; // 鼠标按下点距盒子右边距离
```

```
    document.onmousemove=function(){
```

```
        var moveE=window.event||arguments[0];
```

```
        if(zhuangtai==true){ // 当状态为 true，允许拖动
```

```
            artBox.style.left=moveE.clientX-gensuiX+"px"; // 拖动盒子左偏移
```

```
            artBox.style.top=moveE.clientY-gensuiY+"px"; // 拖动盒子上偏移
```

```
            if(parseInt(artBox.style.left)<=125){ // 左边界值说明，最小为 125px
```

```
                artBox.style.left="125px";}
```

```
            if(parseInt(artBox.style.left)>=1005){ // 右边界值说明，最大为 1005px
```

```
                artBox.style.left="1005px";}
```

```
            if(parseInt(artBox.style.top)<=0){ // 上边界值说明，最小为 0px
```

```
artBox.style.top="0px";}

}}};

document.onmouseup=function(){
    zhuangtai=false;}} // 鼠标抬起，恢复状态为 false
```

鼠标按下时获取其 Event 事件，得到其距离窗口的左边距，同时得到盒子距窗口的左边距，二者相减得到鼠标按下点距盒子的左边距，这样每次拖动只需要将这个按下点距离减去并将结果作为盒子的左偏移，即可得到拖动效果。

#### 4.2.3 购物车模块

设计过程：本模块实现的是用户登录网站后对商品的选购。当用户登录网站后，通过导航栏进入商品列表页面，选中想要购买的商品，查看详情加入购物车，确定商品信息填写收货地址提交订单并支付，完成商品选购。购物车模块流程图如图 4-6 所示。

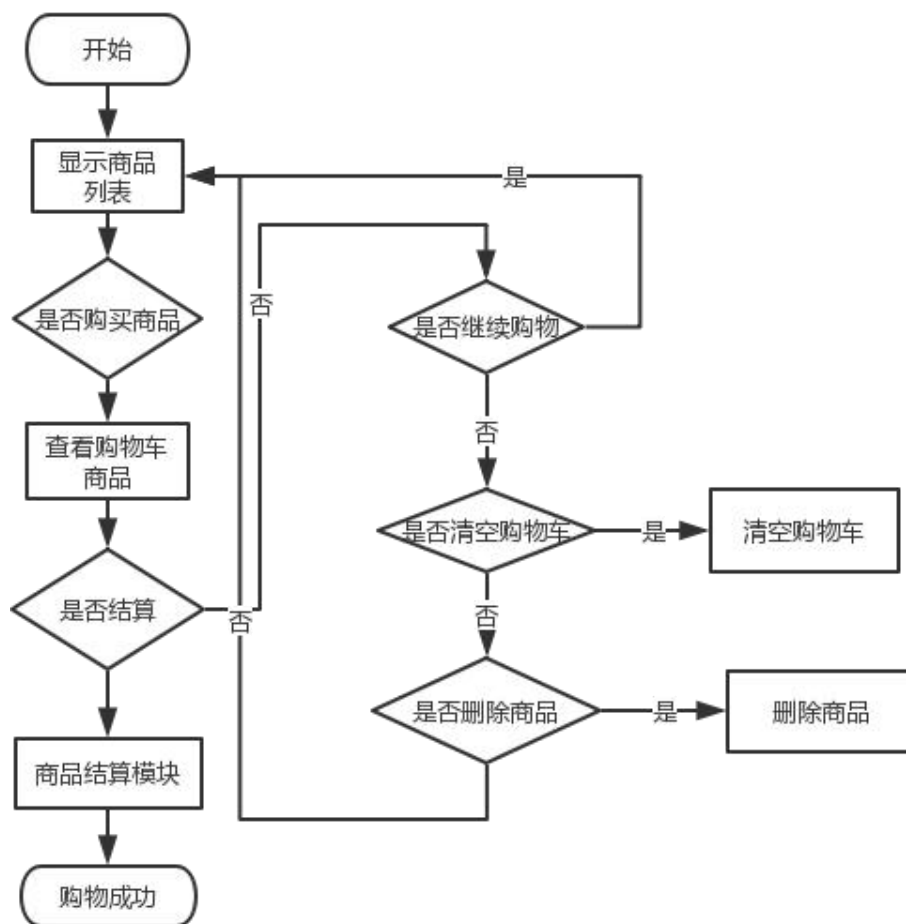


图 4-6 购物车流程图

相关核心代码：

```
exports.showCartdetail=function(req,resp){ // 展示购物车详情
    var username=req.session.username; // 获取 request 对象上的 username
    var sql="UPDATE t_shoppingcart SET state=0 WHERE u_id=(SELECT u_id
FROM t_user WHERE tel=?)"; // SQL 语句，用于将商品状态改为 0
    util.connect(sql,[username],function(){
        util.connect(sql,param,function(err,data){ // 连接数据库
            if (data!==undefined){ // 当数据库查询结果有值时，绑定数据
                resp.render("ShoppingCart",{shopCartdetail:data,username:req.session.username,hSrc:req.session.header});
            }
        })
    });
};
```

点击购物车时首先应将购物车存在的商品状态改变。根据 Session 上面的用户名拼接 Sql 语句，将当前用户购物车商品状态全部设置为零，表示商品已经加入购物车中。然后如果数据库返回成功的话再渲染购物车页面。

本章在数据访问的实现中介绍数据库操作的相关函数，同时还介绍了本系统封装的数据库操作函数。在第二部分中从登陆注册、商品查询及浏览、购物车三个模块中详细阐述了系统的设计方案及相关核心代码。至此整个系统的设计开发基本完成。

## 5 系统实现与测试

本艺术品销售平台是一个在线的 B2C（Business-to-Customer）模式电商网站，其设计主要分为前台页面设计和后台逻辑控制两个方面。本章主要以截图形式介绍和说明系统的最终实现样式。

### 5.1 网站的实现

本艺术品销售平台可以实现用户注册、商品浏览、商品查看详细信息、选购商品、查看订单等功能。页面首页如图 5-1 所示：



图 5-1 首页效果图

从网站的首页上我们可以看到本系统的所有功能，登录、注册、分类查看所有商品信息，并且随着浏览商品将会一步步更加清晰的引导用户。

在本系统的设计过程中，主要有以下几个亮点：

#### (1) 导航栏的设计

图 5-2 是本系统的导航栏所在，通过其实现了对整个网站的划分以及引导，采用半透明加黑底白字动态切换的方式，使得其看上去更醒目，带来更加良好的用户体验。



图 5-2 导航栏

导航栏是一个网站设计成功与否的重要评判标准，同时也决定着用户能否快速成功的获取到信息。在电商网站中决定着用户选购商品的体验度以及影响着电商主体的经济效益。导航应该具有清晰的内容划分和连接模式，可以从初始页开始领着用户进行一步



一步的深入了解，可以让用户一眼便知要获取的信息在哪里。电商网站导航栏的设计应该以简洁、醒目、划分明确为目标，因为用户一旦无法最短时间里找到目标，必然会选择离开。在本系统中将导航栏划分为首页、购艺术、艺术家、数字展览馆四个模块，用户能够快速定位到想要到达的地方，并且通过滚动条的监听来动态切换背景和文字颜色，使得无论在何种地方皆是最醒目的，如图 5-3 所示。



图 5-3 导航栏状态二

## （2）登录注册功能实现

如图 5-4 为网站登录注册页面。用户输入完手机号后系统后台会自行判断是否已注册过，从而来决定接下来显示的内容。



图 5-4 登录注册页面



如图 5-5 为网站的登录页面，用户输入注册过的手机号会出现密码输入框。

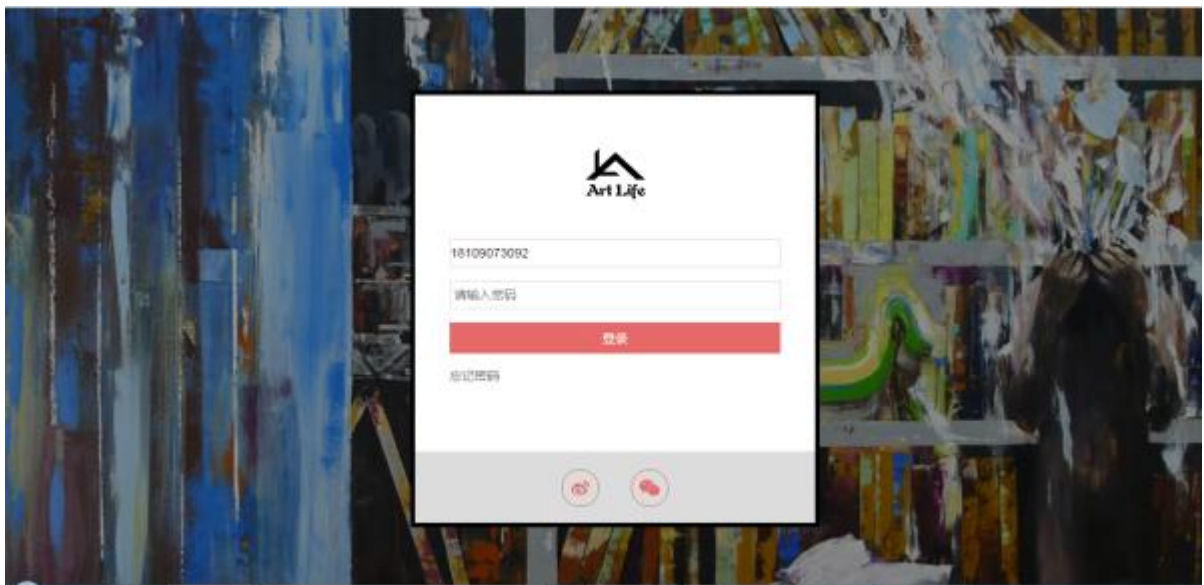


图 5-5 登录页面

如图 5-6 所示为本网站的注册页面，提示用户获取验证码并输入，然后设置用户登录密码信息。

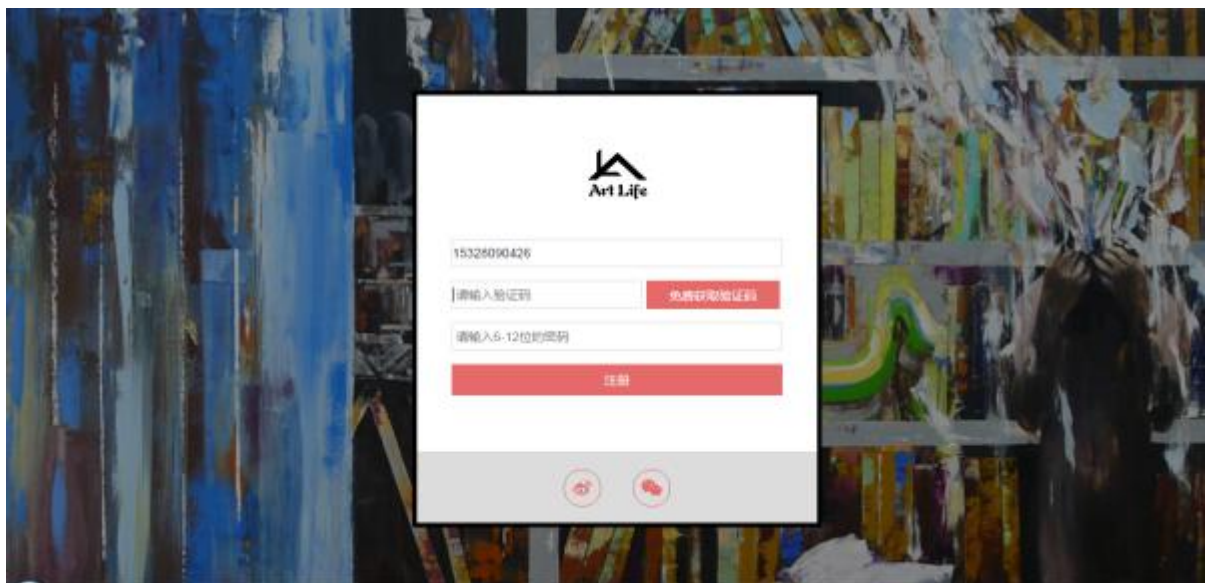


图 5-6 注册页面

#### （4）商品浏览功能实现

图 5-7 所示是本系统的商品浏览界面，当用户点击导航栏的购艺术即可查看本系统中所有类别的商品，采用瀑布流式布局，商品竖直显示，摒弃传统横向整齐排列带来的视觉疲劳，更加吸引用户。

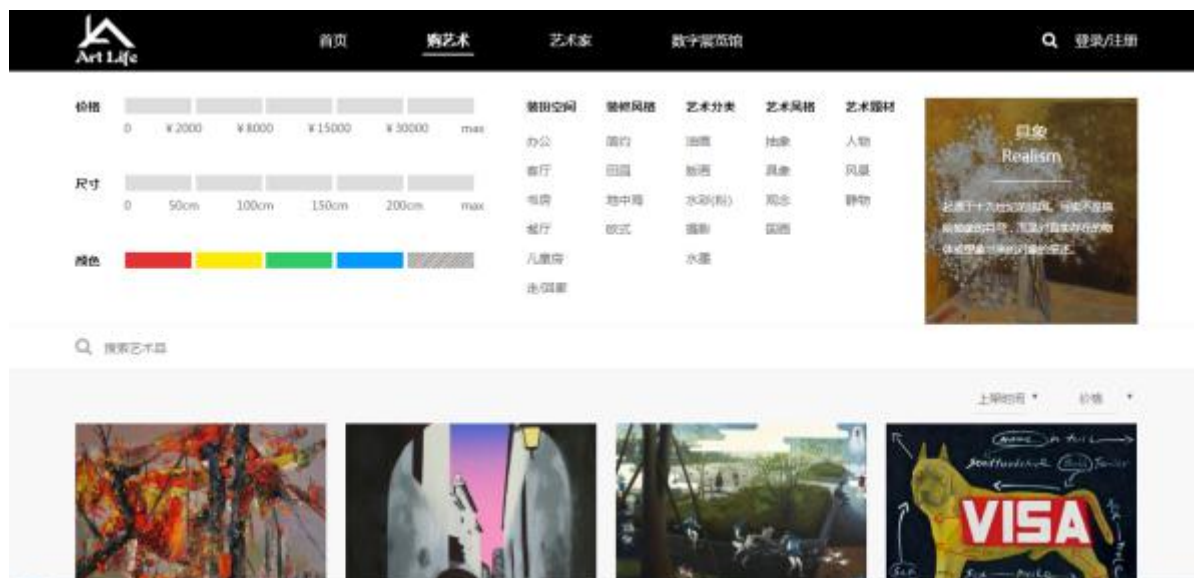


图 5-7 商品浏览界面

在商品浏览中用户可以点击按颜色、尺寸、价格等条件对想要购买商品进行筛选，其次还可以通过搜索框进行商品名模糊搜索，如图 5-8 所示。



图 5-8 商品筛选及模糊搜索

### （5）商品详情查看

如图 5-9 所示为本系统详情界面。当用户选定商品后点击商品图片或者名称进入到商品详细介绍页面，在此页面可以查看商品详细信息、推荐信息、保养信息等。



图 5-9 商品详情页

在商品详情页面还可以点击下方的更多背景、展示画框来动态改变商品背景，以满足用户不同场景的需要，如图 5-10 所示。

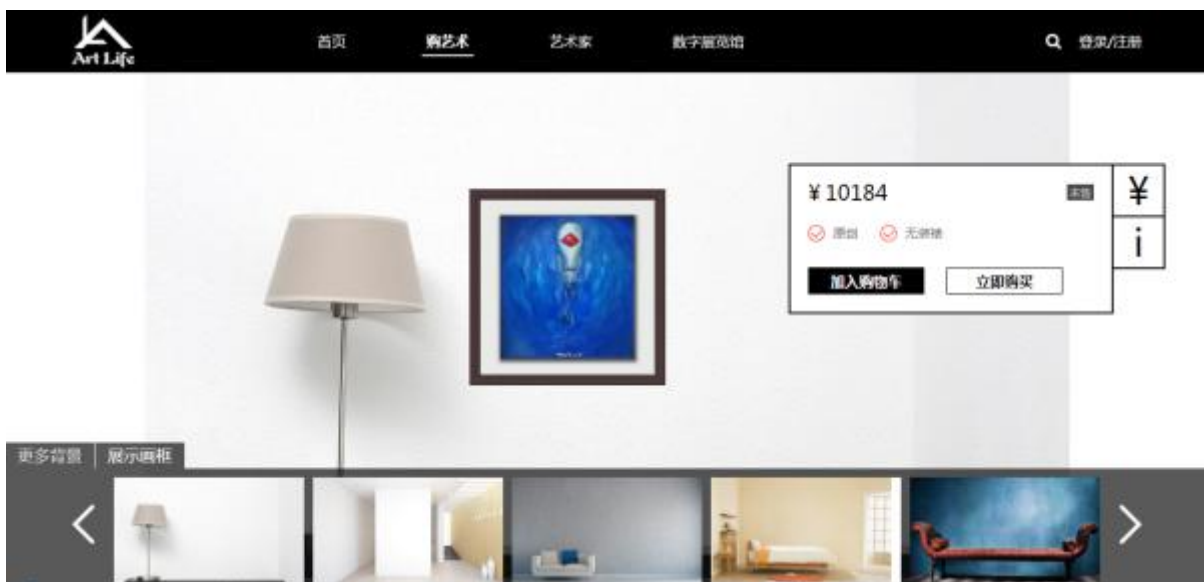


图 5-10 商品背景切换

## （6）商品结算功能实现

如图 5-11 是商品结算页面实现。用户点击结算系统自动下单，用户按照提示选择或者新增收货地址，选择支付方式完成支付流程。

图 5-11 商品结算

## 5.2 测试

软件开发过程中最后也是最关键的一步----软件测试，软件测试对软件的质量严格把关，在软件开发流程中发挥着至关重要的作用，无论是软件功能性的检验还是对软件安全性的保障，软件测试都有着无可替代的地位。只有在软件测试上下足功夫，才能及时发现软件在整个软件设计过程中存在的问题并加以纠正，使得软件质量有保障，才能要让一款新的软件更好的适应市场的需求，得到用户的认可。

### 5.2.1 测试计划

(1) 测试方法：黑盒测试法。

(2) 测试用例设计原则。

测试用例设计的最基本要求：覆盖住所要测试的功能。并且还应做到单个用例覆盖最小化原则、测试用例替代产品文档功能原则、单次投入成本和多次投入成本原则、使测试结果分析和调试最简单化原则。本系统测试用例应该考虑覆盖方面如下：

(1) 正确性测试：输入用户的实际数据用以验证系统是否满足需求要求；

(2) 容错性测试：查看软件在存在异常条件下的行为。输入非法或者非正常数据检查程序是否做出相应处理；

(3) 接口测试：检测系统间各个功能模块和接口是否正常。测试的重点是要检查



数据的传递，交换，控制管理过程；

（4）数据库测试：按照数据库设计规范对软件系统的数据库结构、数据表及表之间数据调用关系进行的测试；

（5）压力测试：不是在常规条件下运行手动或自动测试，而是在计算机数量较少或系统资源匮乏的条件下运行测试<sup>[14]</sup>；

### 5.2.2 详细测试内容

#### （1）首页测试

在浏览器地址栏输入：<http://localhost:8888/index.html>

输入效果如图 5-12 所示

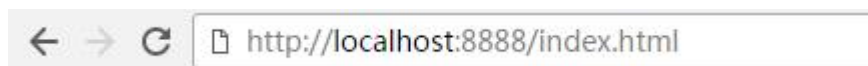


图 5-12 首页测试

测试结果如图 5-13 所示。

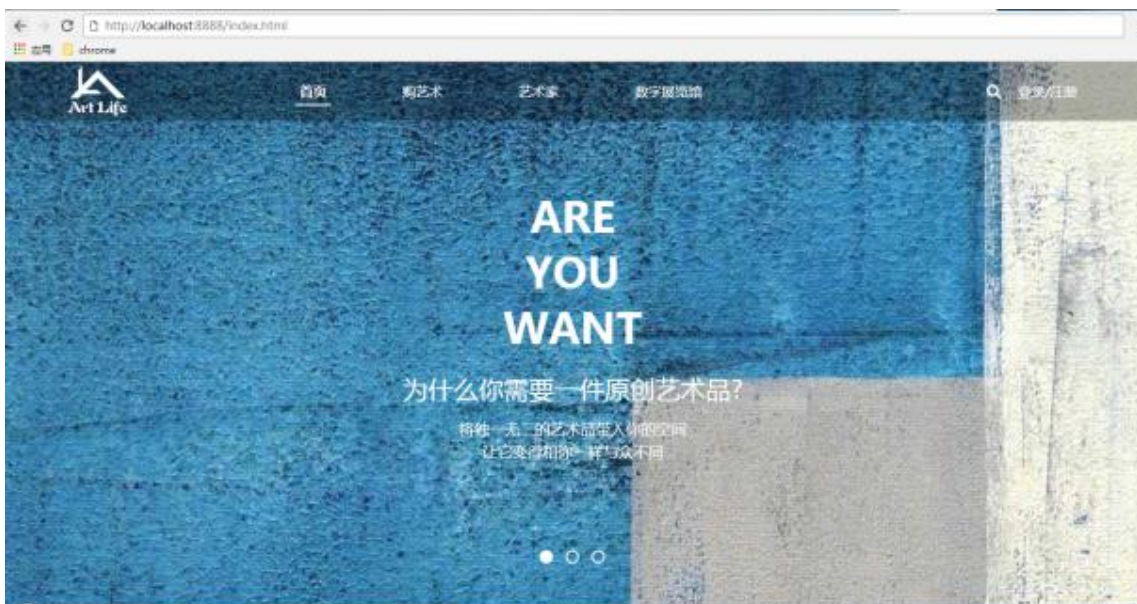


图 5-13 首页测试结果

### （2）用户登录测试

测试用户账号是否存在，若不存在，再去注册，若存在，测试密码是否正确。如图 5-14 所示。

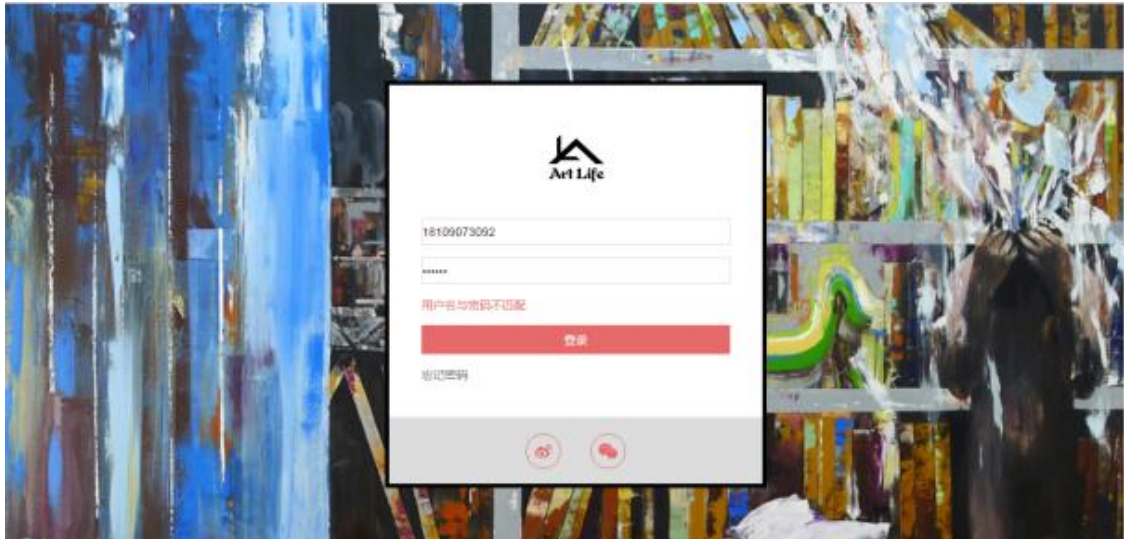


图 5-14 登录测试

### （3）用户注册测试

当系统无法找到匹配用户信息时提示用户注册，用户输入验证码、密码完成注册功能。如图 5-15 所示。

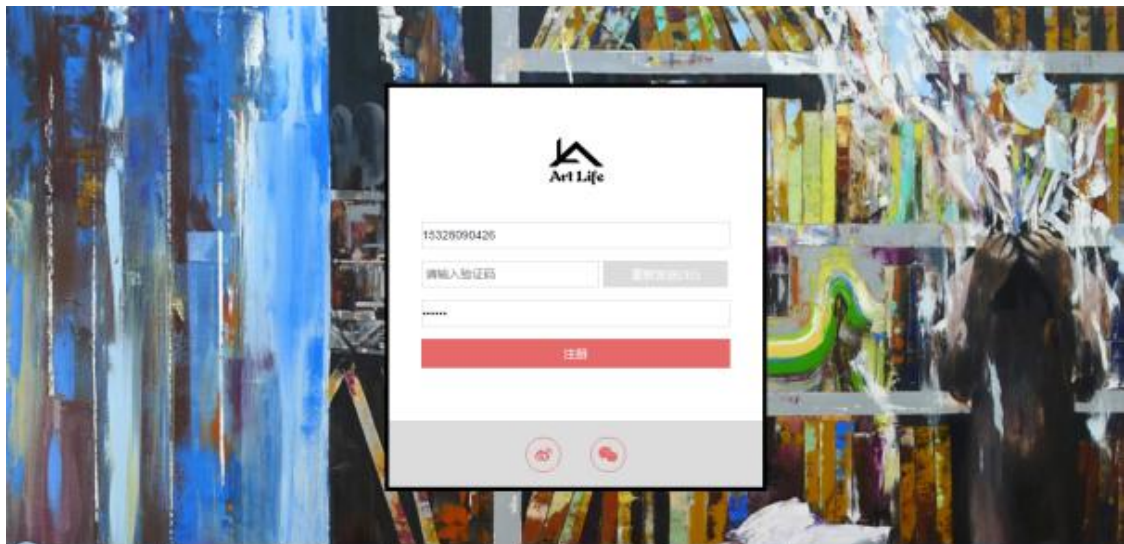


图 5-15 注册测试

（4）购物车功能测试：（如图 5-16 所示）



图 5-16 购物车测试

5.2.3 测试执行及效果情况

（1）测试执行情况如表 5-1 所示

表 5-1 测试执行情况

| 测试类型   | 测试内容       | 测试目的      |
|--------|------------|-----------|
| 功能测试   | 用户登录注册     | 核实功能实现情况  |
| 功能测试   | 商品浏览、查询、购买 | 核实功能实现情况  |
| 用户界面测试 | 导航栏、链接、颜色  | 风格是否统一、美观 |

（2）测试效果说明

针对系统所有的功能、提示、按钮等，采用黑盒法对本系统进行了多次测试，当系统在测试用例和多次操作时，系统运行无误，通过测试。结论如表 5-2 所示。

表 5-2 测试效果情况

| 测试内容            | 测试结论             |
|-----------------|------------------|
| 界面测试            | 界面友好，色调统一，测试结果良好 |
| 登录、注册测试         | 密码过于简单，有待改进      |
| 功能测试（商品、购物车、订单） | 基本通过测试           |

本章采用黑盒法对本系统进行了详细测试，测试用例力求做到覆盖所有适用范围，边界情况详细考虑，无漏洞。最终系统通过整个测试，在试运行中满足需求开发中提到的所有要求，正常运行<sup>[15]</sup>。

## 结 论

本次毕业设计总的来说还是比较成功的，从选题到答辩，整个过程都比较规范。在分析理解需求的基础上，按照软件开发的要求已经完成了开发。虽说在开发过程中遇到了不少技术上的难点，但通过查阅资料以及请教老师都能解决，最终成功通过测试且运行状态良好。这算是对自己几个月以来的努力最好的回报，本章将对此次开发做出总结。

本系统优点：

（1）瀑布流式布局：商品展示页面采用瀑布流式布局，摒弃传统横向布局带来的视觉疲劳，让用户更清晰的浏览商品，提升用户体验度和好感度；

（2）3D 图片展厅：该模块实现方式来源于某网站，通过 3D 显示方式，不仅可以让用户更加直观查看商品图片，更能吸引用户的眼光，增加访问量；

（3）界面简洁统一：本系统设计时本着风格简洁、色调统一为目标，采用黑白红三色作为主色调，醒目而不是优雅；

（4）系统人性化：在本系统中，无论是登录注册还是购买支付，都本着简化操作，让用户以简便操作带来更好体验为原则。因此更加符合如今快节奏的生活方式；

缺点：

（1）功能缺失：目前系统的功能还是较为简单，只是完成了电商网站主要的购物流程，从系统的发展长远考虑，还有待进一步扩充更多功能；

（2）安全因素：安全因素作为计算机行业一个重要问题，不容忽视，在本系统中除了短信验证码以外几乎没有考虑安全性问题，这也是一个重大的隐患；

虽说存在部分缺点，但是总的来看本系统还是符合预期要求的。将在以后的维护与版本更新中，对存在的问题和不足的功能进行改善。



## 参考文献

- [1] 初怡. 中国艺术品市场营销分析[D]. 沈阳: 鲁迅美术学院, 2013.
- [2] 马银芳. 艺术品网站的设计开发[D]. 济南: 山东大学, 2009.
- [3] Adam Polak, Timothy Kelman, Paul Murray, *et al.* Hyperspectral imaging combined with data classification techniques as an aid for artwork authentication [J]. *Journal of Cultural Heritage*, 2017, 01(13): 42-47.
- [4] 郝菊香. 优乐汇网上购物商城的设计与实现[D]. 西安: 西安电子科技大学, 2013.
- [5] 许会元, 何利力. NodeJS 的异步非阻塞 I/O 研究[J]. *工业控制计算机*, 2015, (03): 27-129.
- [6] 程桂花, 沈炜, 何松林, 等. Node.js 中 Express 框架路由机制的研究[J]. *工业控制计算机*, 2016, (08): 101-102.
- [7] 刘鑫. 基于 JSP 的网上购物系统研究与设计[D]. 北京: 北京邮电大学, 2013.
- [8] Hadi Nahari, Ronald L. Krutz. *Web Commerce Security: Design and Development*[M]. New York: Wiley. 2011.
- [9] 李阳. Miche 女包购物商城系统设计与实现[D]. 济南: 山东大学, 2014.
- [10] Schwartz B, Zaitsev P. *High Performance MySQL*[M]. Sebastopol: O'Reilly Media, Inc. 2012.
- [11] 朱海水, 赵亚娟. 网上商城购物系统设计与实现[J]. *无线互联科技*, 2015, (12): 66-68.
- [12] Pedro Teixeira. *Professional Node.js Building Javascript Based Scalable Software*[M]. Birmingham: Wrox. 2012.
- [13] Nicholas C, Zakas. *Professional JavaScript for Web Developers 3rd Edition*[M]. New Jersey: John Wiley & Sons, Inc. 2012.
- [14] 周煜, 周国庆, 奚文骏. 软件测试技术概述[J]. *中国测试技术*, 2005, (03): 56-58+75.
- [15] 陈文海. 软件测试管理工具的研究与实现[D]. 北京: 中国科学院研究生院(软件研究所), 2003.

## 致 谢

从刚开始的一片空白到如今的设计完成离不开张君雁老师的细心指导，在这段设计时间中，随着老师的思路指引我们学会了很多！不仅仅是理论上的，更多的是自己实践能力的提升。从开题报告，确定课题方向和题目，随后又在老师的指导下论证方案，大胆设计，编写程序，完成开题报告，在老师不断的讲解开导下，也一遍又一遍的修改代码，修改程序流程，完善了初步设计，工作不是到此就结束了，接下来的日子完成了设计资料的收集，对程序代码又进行了详细的分析，最后在张君雁老师指导下，进行了最后的修改和调试，至此，设计大致完成了。

不积跬步何以至千里，毕业设计能够顺利的完成，也归功于大学期间各位任课老师平时认真负责的教学和指导教师孜孜不倦的指导和关心，使我能够掌握和运用专业知识完成毕业设计。正是有了他们的悉心帮助和支持，才使毕业设计顺利完成，在此再次向学校和老师致以衷心的感谢，感谢你们的大力关心和教导。