**深 圳 大 学**

**联合培养**

**深圳职业技术学院**

本 科 毕 业 论 文（设计）

题目: 移动web电子阅读器

姓名: **李俊瑾**

专业: **计算机科学与技术**

学院: **计算机工程学院**

学号: **13190023**

指导教师: **秦高德**

职称：

2017 年 4 月 3 日

**深圳大学深圳职业技术学院联合培养**

**本科毕业论文（设计）诚信声明**

本人郑重声明：所呈交的毕业论文（设计），题目《**基于HTML5技术的移动商城开发**》 是本人在指导教师的指导下，独立进行研究工作所取得的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明。除此之外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。本人完全意识到本声明的法律结果。

毕业论文（设计）作者签名：

日期： 年 月 日

目录

[【摘要】 1](#_Toc449879152)

[1. 引言 2](#_Toc449879153)

[1.1. 选题背景 2](#_Toc449879154)

[1.2. 国内现状 2](#_Toc449879155)

[1.3. 国外现状 2](#_Toc449879156)

[2. 概述 3](#_Toc449879157)

[2.1. 课题概述 3](#_Toc449879158)

[2.2. 涉及技术 3](#_Toc449879159)

[2.3. 论文内容简介 3](#_Toc449879160)

[2.3.1. 系统设计方案 3](#_Toc449879161)

[2.3.2. 开发环境 4](#_Toc449879162)

[2.3.3. 需求分析 4](#_Toc449879163)

[2.3.4. 系统设计 4](#_Toc449879164)

[2.3.5. 系统实现方案 4](#_Toc449879165)

[3. 系统设计方案 5](#_Toc449879166)

[3.1. 可行性分析 5](#_Toc449879167)

[3.1.1. 技术可行性 5](#_Toc449879168)

[3.1.2. 经济可行性 5](#_Toc449879169)

[3.1.3. 社会可行性 5](#_Toc449879170)

[3.2. 开发环境 6](#_Toc449879171)

[3.2.1. 硬件平台 6](#_Toc449879172)

[3.2.2. 软件平台 6](#_Toc449879173)

[3.3. 需求分析 7](#_Toc449879174)

[3.3.1. 服务端功能需求 7](#_Toc449879175)

[3.3.2. 管理端功能需求 8](#_Toc449879176)

[3.3.3. 系统性能需求 8](#_Toc449879177)

[3.4. 系统设计 9](#_Toc449879178)

[3.4.1. 总体设计目标 9](#_Toc449879179)

[3.4.2. 系统功能描述 9](#_Toc449879180)

[3.4.3. 系统功能模块图 9](#_Toc449879181)

[3.4.4. 业务流程图 12](#_Toc449879182)

[3.4.5. 数据字典 13](#_Toc449879183)

[3.4.6. 数据流分析 16](#_Toc449879184)

[4. 系统实现方案 21](#_Toc449879185)

[4.1. 实现步骤 21](#_Toc449879186)

[4.2. 响应式适配的实现 21](#_Toc449879187)

[4.3. 注册功能的实现 22](#_Toc449879188)

[4.4. 购物车编辑功能的实现 26](#_Toc449879189)

[4.5. 瀑布流显示的实现 30](#_Toc449879190)

[4.6. 商品轮播展示功能的实现 33](#_Toc449879191)

[5. 结束语 42](#_Toc449879192)

[【参考文献】 43](#_Toc449879193)

[致谢 44](#_Toc449879194)

[【Abstract】 45](#_Toc449879195)

**移动web电子阅读器**

计算机工程学院 计算机科学与技术

李俊瑾 学号：13190023

【摘要】

自从21世纪开始，因特网得到了飞速的发展。随着互联网时代的到来，也极大地改变人们生活的方方面面。如今，“互联网+”的生活模式正在渗透进人们的生活。其中尤为突出的是基于Internet的电子商务潮流。

本文论述的是名为“骚盒”的电子商城网站系统。它是一个基于HTML5技术开发的移动电商分销与管理平台。这个系统分为两大主题部分。在分销平台系统中，它具有会员注册与登录、浏览商品、个人店铺商品、用户购物车、查看店长日记、买家评论、用户订单、修改密码、用户银行卡绑定和其他用户信息等管理等功能；在管理平台系统中，它具有商城销售信息统计、个人店铺管理、个人产品管理、消息中心、店铺团队管理、用户收益管理、个人商城信息管理等商城后台功能。

本系统是根据B/S结构，基于HTML5**[1]**，CSS3**[2]**，JavaScript**[3]**等开发语言进行开发，同时采用采用集成的JavaScript轻量级框架Zepto.js**[4]**、和专为移动端服务的Frozen UI**[5]**前端框架进行辅助开发。

本系统主要是针对微信个人用户开发设计的微商综合管理平台，方便用户随时随地进行销售与管理活动。

**【关键字】**

骚盒;移动电商；html5；css3；JavaScript

# 引言

## 选题背景

在当今时代，随着社会的迅猛发展，互联网的趋势渐渐从PC时代转向移动互联时代，越来越多人的衣食住行都依赖于移动设备。“一机在手，行走天下”的生活方式渐渐在影响着大众，手机以其自身携带方便，功能齐全的优势，成为人们生活中必不可少的工具。因此基于移动端的产品开发成为一种潮流。

在这个小康时代，大众的衣食住行基本上都得到了满足，越来越多的人开始转向精神方面的享受。比如阅读，旅行，音乐，电影这些精神粮食。生活节奏越来越快，为了保证用户适应快节奏生活的同时又能享受到精神粮食，基于这方面的开发研究显然是一个必要又有前途的选择。本次选题是移动web电子阅读器，是一个轻量级的web app,，用户随时随地可以阅读小说。

## 国内现状

阅读对于国民来说是一件重要的事情。据新闻报导，2015年我国成年国民图书阅读率为58.4%，数字媒介阅读率为64.0%，在数字媒介中有60.0%的人是通过手机进行阅读的，也就是说在通过数字媒介阅读的国民中有超过一半的人是通过手机来阅读的，在我们日常生活中，出门坐地铁，使用手机阅读，打发时间也成为了一种流行。

当今国内也有很多知名的电子阅读器APP，如掌阅科技有限公司的掌阅APP，腾讯旗下的QQ阅读。

## 国外现状

国外也越来越多的人阅读方式在改变，渐渐从纸质书走向电子书，流行于阅读手机小说，据调查发现在欧美阅读电子书的用户中，在阅读终端的选择中，更偏向于选用手机来阅读，因为比起平板，电脑，阅读器来说，手机功能更多，携带更方便。

# 概述

## 课题概述

基于上述的背景以及国内外现状，开发一个适合当今社会流行趋势，符合大众需求的阅读APP是一件必要的事情。本项目是一个轻量级的电子阅读器web APP，web app是区别于原生native app及混合hybrid app的一种运行在浏览器端的网页，优势是轻量级，不需要像原生APP那样经过应用商店审核才更新，可以直接更新版本。分别有书城和书架两个外部界面，可以让用户方便快捷地在浏览器端打开，阅读小说。

本系统前端框架部分是基于HTML5为开发语言，CSS3技术编写页面样式，Zepto.js技术来实现动态HTML页面以及用户与界面的交互，后端接口使用基于Node.js平台的koa框架跟MongoDb来开发，其中还用了Vue.js来将接口数据渲染进去DOM。

HTML5：超文本标记语言第五代版本，增加了语义化标签，废除一些不好用的属性，同时还多了很多好用的新语法特性，比如本文会用到的本地缓存技术localStorage，还有支持多媒体的canvas，video，audio标记。HTML5很适合构建手机应用，最大的优势就在于可以在网页上调试跟修改；

CSS3：层叠样式表，构建精美网页的神器，可以有效精准地对页面的布局，颜色，字体进行控制。有了CSS3，前端开发者再也不用图片去制作小图标，因为CSS3有了圆角表格特性，比如本文用到的border-radius可以用来做圆角。以及新增的动画属性，变形(transform)、转换(transition)和动画(animation)，可以直接使用，让我们在不操作DOM结构的情况就实现了动画效果。

Zepto.js：轻量级的开源JavaScript库，本系统选用了该库，因为文件比较小巧，仅有8K多，在目前功能完备的库里面算是最小的，所以比起jQuery更适合移动端开发，兼容jQuery，让习惯了使用jQuery的开发者在不用耗费精力学习API的同时又能获得更快速度。

Vue.js:时下很火的用户界面框架，关注视图层，容易上手，支持组件化开发，通过尽可能简单的API实现数据绑定，让开发者不用去操作DOM。

Node.js:简单来说它就是运行在服务端的JavaScript, 它不是JavaScript应用，而是JavaScript平台，是一个基于Google的V8引擎的事件驱动I/O服务端JavaScript环境， 轻量且高效，为前端开发者带来便利，可以用熟悉的JavaScript语言去做web服务，使得前后端语言风格统一。

Koa1.0:基于node.js的web开发框架，是Express框架的人员开发出来的，可以用于快速搭建项目以及异步编程，比起Express更加轻量级，没有封装过多的中间件，让开发者只选择自己最需要的中间件去开发。其中koa的中间件generator函数是基于request请求，以以一个栈的结构形式去依次执行。

MongoDB：著名的NoSQL，文件存储类型为BODN（JSON的扩展），以键值对的形式存储。MongoDB中的集合和文档类似于关系型数据库中的表跟行。操作方便，性能高。

## 涉及技术

本系统前端框架部分是基于HTML5为开发语言，CSS3技术编写页面样式，Zepto.js技术来实现动态HTML页面以及用户与界面的交互，后端接口使用基于Node.js平台的koa框架跟MongoDb来开发，其中还用了Vue.js来将接口数据渲染进去DOM。

HTML5：超文本标记语言第五代版本，增加了语义化标签，废除一些不好用的属性，同时还多了很多好用的新语法特性，比如本文会用到的本地缓存技术localStorage，还有支持多媒体的canvas，video，audio标记。HTML5很适合构建手机应用，最大的优势就在于可以在网页上调试跟修改；

CSS3：层叠样式表，构建精美网页的神器，可以有效精准地对页面的布局，颜色，字体进行控制。有了CSS3，前端开发者再也不用图片去制作小图标，因为CSS3有了圆角表格特性，比如本文用到的border-radius可以用来做圆角。以及新增的动画属性，变形(transform)、转换(transition)和动画(animation)，可以直接使用，让我们在不操作DOM结构的情况就实现了动画效果。

Zepto.js：轻量级的开源JavaScript库，本系统选用了该库，因为文件比较小巧，仅有8K多，在目前功能完备的库里面算是最小的，所以比起jQuery更适合移动端开发，兼容jQuery，让习惯了使用jQuery的开发者在不用耗费精力学习API的同时又能获得更快速度。

Vue.js:时下很火的用户界面框架，关注视图层，容易上手，支持组件化开发，通过尽可能简单的API实现数据绑定，让开发者不用去操作DOM。

Node.js:简单来说它就是运行在服务端的JavaScript, 它不是JavaScript应用，而是JavaScript平台，是一个基于Google的V8引擎的事件驱动I/O服务端JavaScript环境， 轻量且高效，为前端开发者带来便利，可以用熟悉的JavaScript语言去做web服务，使得前后端语言风格统一。

Koa1.0:基于node.js的web开发框架，是Express框架的人员开发出来的，可以用于快速搭建项目以及异步编程，比起Express更加轻量级，没有封装过多的中间件，让开发者只选择自己最需要的中间件去开发。其中koa的中间件generator函数是基于request请求，以以一个栈的结构形式去依次执行。

MongoDB：著名的NoSQL，文件存储类型为BODN（JSON的扩展），以键值对的形式存储。MongoDB中的集合和文档类似于关系型数据库中的表跟行。操作方便，性能高。

本移动web电子阅读器按照结构模块开发，分别为前台书籍展示页面和后台接口开发，前端数据从后台接口获取，开发方式上以前端为核心。

## 论文内容简介

### 系统设计方案

系统设计方案简述了系统的整体设计原理和设计目标。

### 开发环境

开发环境部分介绍了系统对硬件和软件的参数要求和开发系统所用到的开发工具介绍。

### 需求分析

需求分析部分分别对前台用户使用和后台用户管理两个不同的操作平台进行了需求分析，罗列了一系列根据分析研究确定的用户需求。

### 系统设计

系统设计从全局的高度出发，整体概括了系统的业务逻辑模式。通过图表的形式更加直观明了地描绘出系统的运转过程和数据交互形式。

### 系统实现方案

系统实现方案详细地对系统主要的功能模块的开发和实现过程作了介绍，包括关键的核心代码片段，并附上了相应的注释说明。

### 系统运行测试

系统运行测试介绍了在系统开发工作完成之后对系统进行实际操作运行的情况，有实际的过程截图，并通过测试结果得出一些结论。

# 系统设计方案

## 可行性分析

在该系统中，可行性分析包括两个部分，从市场需求跟资源供应方面来看，由前面的分析可得知现在开发一款移动web电子阅读器，符合大众普遍要求。资源方面，现在互联网时代，资源共享，该系统的资源大部分可以从互联网上获取，放进去自己的项目里面，方便用户随时阅读。Web App跨平台，实现“一次开发，到处运行”，从各个方面来看，我们开发的这个系统是对用户而言有着积极的作用。

### 技术可行性

由前文列出的技术来看，HTML5跟CSS3新增的元素和各种属性，使得可以代替Flash动画功能，同时由于语义化表签的加入，代码的可读性也更强，更利于搜索引擎优化。在性能方面，从全球的几大主流站点都使用了HTML5来开发可以看出，它值得使用。后端接口开发使用node.js框架—koa，对于一个专注于前端开发的程序员来说是一件好事，可以使用自己熟悉JavaScript语言来的编写服务器端代码，交给Node.js去解释执行。因为，在技术上来看，使用主流的HTML5+CSS3，加上易于上手的Node.js是可行的。

### 经济可行性

开发一个Web App成本不高，无需购买服务器，后端接口可以使用开源框架koa以及免费的MongoDB，前端界面搭建，只有有一台笔记本，在任何操作系统下，随时随地可以进行开发。此处推荐使用Sublime Text3,轻量级，可以提高开发效率。调试程序可以使用谷歌浏览器自带开发者调试工具，功能强大，简单按下F12即可进行调试。从各个方面看，开发一个移动web电子阅读器几乎不用任何成本。

### 社会可行性

一般情况下，社会可行性是指所开发的系统对社会是否产生正面积极的影响，是否符合国家制定的法律法规，同时还要全面考面用户使用系统时是否存在恶意操作而出现违法行为。

#### 法律因素

开发的系统在未来上线之后，将会有专人对用户所销售的产品和用户的销售行为进行审核监管，对于可能会触犯法律法规的行为进行强制处理。另外，本系统开发过程由本人独立完成，没有进行过任何抄袭行为。

#### 用户使用可行性

本系统开发使用的环境是已经普及的微软window7操作系统。而HTML5开发的网站对操作系统几乎没有特殊要求，唯一可能会影响的使用是HTML5的部分属性在安卓操作系统和ios操作系统的兼容性问题。我们只需要在开发过程中对这部分属性进行兼容性处理即可解决上述问题。

## 开发环境

### 硬件平台

硬件平台的正确选择与设计，不仅仅是为了让我们设计的系统能够顺畅无误地运行，也是为了提升用户的使用体验。更重要的是，在当下硬件和软件的结合度越来越高的环境下，选择什么样的硬件平台就意味着相应的要使用什么样的系统开发语言和技术。只有两者的完美结合才能最终设计出高效率、高可靠性和安全性的电子商城系统，并且能够节约一定的投入成本。为了满足我们设计的初衷——轻量且简单易用，从设计之初，本系统定位的硬件配置就是以最为普遍的硬件标准。其标准配置如下：

系统用户的硬件配置：Intel 2.2G 处理器，2G内存，320G 磁盘存储空间；

系统管理者的硬件配置：Intel 2.5G 处理器，4G内存，500G 磁盘存储空间。

(注：尽管如此，我们依然建议用户和管理者在条件允许的情况上使用越高的配置越好，这必将对使用体验有非常大的提升。)

### 软件平台

软件平台是开发系统所需要的软件集成环境，通过需求分析，正确合理地选择开发的软件平台，不仅能够在前期的开发环境搭建中省时省力，同时也将对开发过程的进行起到一定的促进作用。从高效、实用、功能齐全等方面进行对比分析，本系统最终确定采用如下软件平台：

#### 操作系统

任何软件都必须在操作系统下才能运行。在微软发布的windows系统中，如今最新的是已经win10系统，本次开发是使用win7 64位操作系统，win7系统历时比较久，系统更加稳定，大部分用户在尝试了新发布的系统后仍然选用win7.

#### 开发软件

开发软件，顾名思义就是开发者进行系统开发时所使用到的软件一系列的软件和工具。对程序员而言，开发软件一般包括了脚本编辑器、服务器、数据库管理器、以及测试调试工具。本系统的开发软件列表如下：

* 脚本编辑器：

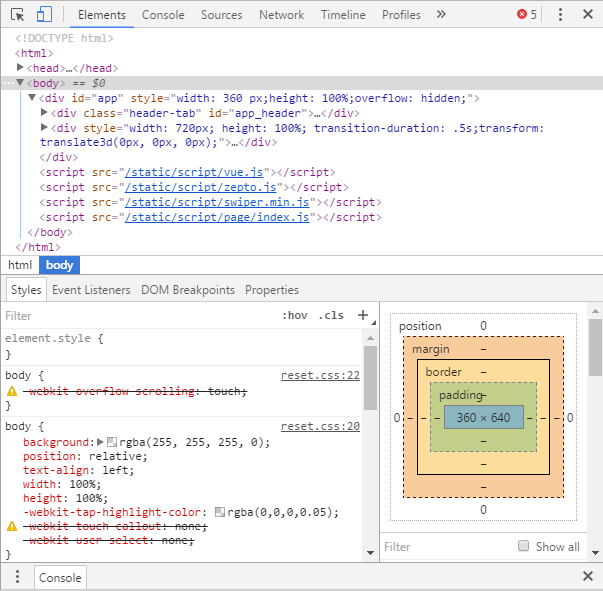
Sublime Text3。近年来大部分企业开发都使用这个工具。虽然是收费软件，但是可以无限次试用，黑色的用户界面，更加护眼，强大的功能，可以自定义快捷键；同时又错误提示，支持代码高亮。使用快捷键打开文件以及查找代码段方便。右侧还设有代码文件缩略图滚动条。方便快捷，值得使用。

* 调试工具

开发过程中难免遇到程序员最痛恨的bug，调试工具就像程序员手中的狙击枪，能够准确快速地捕捉到bug，并对其进行纠错处理。在用户界面HTML5的开发过程中，我们采用的是Chrome浏览器自带的调试功能，它能够完美地模拟手机端的页面展示效果，并且支持实时修改样式参数进行页面效果浏览测试。（当然，Firebug**[6]**同样可以解决）

代码调试时每个程序成功运行的必经之路，也是每个程序员都必须苦苦挣扎的一个点。在没有任何工具的情况下，调试代码会是一件艰难的事情。如今，好在Chrome浏览器自带了功能强大的开发者调试工具，Elements栏目查看以及测试修改页面样式，Console栏目可以打印输出以及测试，Sources栏目可以查看项目源码，Network可以查看页面资源请求情况，还有其他强大的功能，比如设置断点调试。作为前端开发者，熟悉掌握调试技能也是性能优化的必备条件。Chrome浏览器还有自带手机客户端模拟器，可以让开发者轻易看出项目在不同分辨率手机中的显示情况。

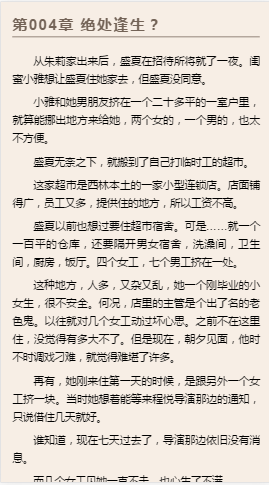


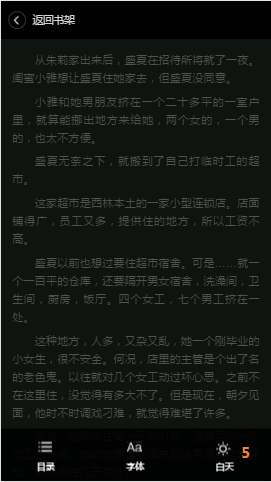
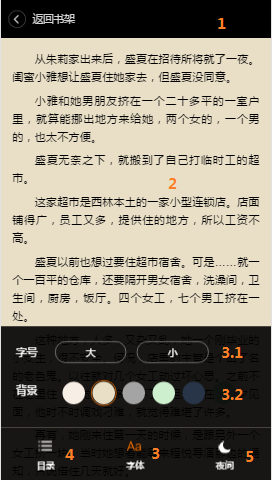


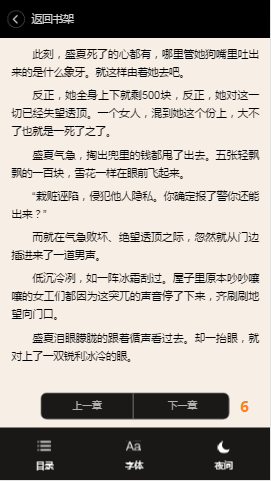
## 需求分析

对软件的需求进行分析，一来是为了开发出实用的系统，二来是为了减少后期用户需求与产品功能的不符而产生的不必要的反复开发过程。需求分析要求开发者对用户的需求进行不断的认识和逐步的细化，只有这样才能清晰准确地知道所开发的系统应该采用何种逻辑架构、具备何种功能等。本系统由分为两个部分开发：先对阅读小说的“阅读器”进行功能需求分析，看了市面上流行的大部分阅读器，总结发现阅读器需要具备三个基本功能，白天/夜间模式切换（夜间可设置护眼模式，提高用户体验），设置字号大小（用户阅读小说时会跟阅读器长时间接触，不同用户对字体大小要求会有不同），翻页按钮（方便用户使用），一下是我们对本系统需求的详细分析：

### 阅读器开发需求







#### 1）公共返回导航

在搭建一个页面时，总会从总到分，从头到尾去写结构。此处我们设置一个公共的导航条用于返回上一级界面。

#### 2）文章内容主体

阅读器的主体部分，用于展示小说内容，点击主体部分中间区域，即可唤出上下导航条以及设置面板，这种”show/hide”设置对用户比较友好，在用户需要设置页面样式的时候再单击唤出，不需要的时候则隐藏起来。

#### 3）字体按钮

点击该按钮可以唤出“字号大小设置”以及“背景颜色设置”的面板，让用户根据自己需求调整阅读模式。

#### 3.1）字号大小设置

考虑到不同用户对字体的要求会不一样，该功能用户可以根据自己的需要设置字体大小，本系统对页面字体大小设置有上下限的限制，让用户界面达到最大的和谐度。同时要保证浏览器可以记忆设置好的选项，即刷新之后依然呈现用户最近一次设置好的字号大小。

#### 3.2）背景颜色设置

分别针对白天有五个不同的背景颜色，以及针对夜间的一个黑色背景颜色。若用户想要改变背景颜色，单击颜色按钮即可将背景颜色调整至最舒适的状态。该按钮也有记忆用户上一次设置的功能。

#### 4）目录按钮

点击目录按钮会跳转至本书相应的目录详情页。

#### 5）白天/夜间模式切换按钮

白天模式默认的背景颜色为浅棕色（#e9dfc7），夜间模式的背景颜色为黑色（#000）；用户点击该按钮即可进行模式切换。

#### 6）上一章/下一章翻页按钮

点击“上一章”或“下一章”按钮可以对小说进行翻页显示。

#### 7）目录详情页

目录详情页展示了该书籍的具体章节信息。

### 书城展示列表需求





#### 1）网站header

书城网站采用水平侧滑的形式展示了两个一级页面，左右两边包括“签到”入口和“登录”入口。

#### 2）书城顶部功能区域

作为该Web App顶部功能区域，包括三个部分：搜索框，活动页轮播图，分类栏目。

#### 搜索框

输入“书名/作者/关键字”实现全局搜索，成为用户寻找书籍的快速接口。

#### 2.2）活动页轮播图

作为众多网站必备的元素，本系统的轮播图使用Swiper技术实现，它是目前应用比较广泛的移动端网页触摸内容滑动的JS插件。

#### 2.3）分类栏目

分类设有“男生频道”，“女生频道”，“分类”，“排行”，点击可以进入二级详情页面。

#### 3）书籍列表模块

书籍列表模块展示了网站热门小说，分别有以下版块：“本周最火”，“重磅推荐”，“女生最爱”，“男生最爱”，“限时免费“，“精选专题”共六大模块，其中“女生最爱”与“男生最爱”每次展示五本书，底部设置“换一换”按钮可切换书籍。

#### 4）书架

书架用于展示用户选中的书本，在右侧的切换按钮可以将书架上的书籍列表切换为内联显示形式。

#### 5）书籍详情

书籍详情页展示了书籍详情以及作者的其他相关作品。

#### 6）搜索详情页

搜索详情页面显示有默认搜索关键词，在输入框键入关键词可以得到搜索结果，若搜索不到然后内容，则显示“没有相应的搜索结果”。

### 3.3.3 服务端接口开发

服务端接口由Node.js驱动，安装koa框架用于快速搭建网站，其中服务器插件均使用Node.js自带的npm(Node.js包管理工具)安装。服务器接口主要包括首页接口，频道详情页数据接口，登录接口，注册接口，搜索接口。

### 3.3.4 系统性能需求

#### 1）设计标准要统一

项目开发时要求做到标准统一。对功能模块进行合理的划分，使得功能模块既能相对独立地进行工作，又能与其他功能模块达到高效地相互协作；代码编辑习惯要严谨、科学、合理，减少协同开发过程中的代码分歧；用户界面要简洁美观、用户使用易上手等。

#### 2）系统响应时间要求

所有数据交互的响应时间应保证在2秒内完成，因此要对数据库的数据进行合理的归档处理。

## 系统设计

系统设计是根据前面所论述的所有需求，然后选择高效的开发工具和环境，为系统实体的正式工作开发流程，所做的必要的前期准备工作。

### 总体设计目标

1. 开发出简洁优美的用户界面，运用合理色彩搭配原则提升用户友好度；
2. 开发的系统要能够实现对市面上主流手机屏幕大小和分辨率的良好适配，实现响应式布局；
3. 系统应当有一定的用户操作提示，指引用户在使用的过程中快速熟悉系统操作；
4. 具有比较好的交互性，在页面组件的控制操作中增加JavaScript功能，提升用户使用体验；
5. 设计系统时为日后的后台开发工作提供友好的可持续和可延续性。

### 系统功能描述

通过以上的需求分析，我们便可以开始进行总体的系统设计了。软件系统应该包含一系列的相对独立的子功能模块，在保证这些子功能模块完整运行其功能时，也要保证功能模块之间的高效协作运转，这样开发出来的才是一个完整高效的系统。

本系统为B/S结构应用，其中前端页面使用HTML5+CSS3+Zepto.js开发，服务端接口使用Node.js开发，接口主要为前端提供数据，本系统最重要的部分在于前端界面开发以及与用户交互部分的开发。

阅读器模块实现小说内容展示，阅读器背景颜色设置，字号大小设置，上下翻页功能以及点击返回首页等。

书城模块实现书籍展示，书籍搜索，签到，登录注册，点击进入书籍详情页等功能。

服务端接口实现了后端逻辑，实现

### 系统功能模块图

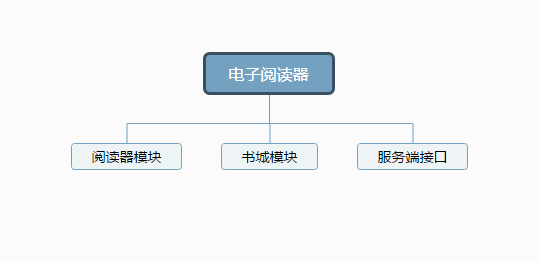


图1.系统功能模块图

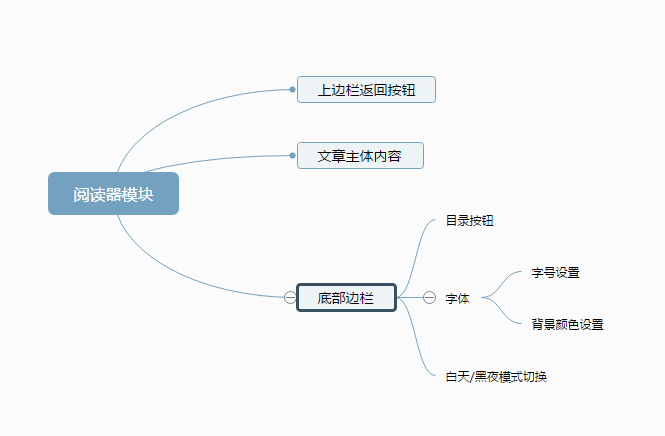


图2.阅读器功能模块图

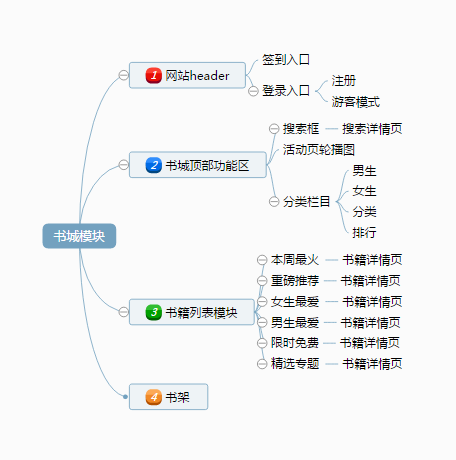


图3.书城功能模块图

绑定银行卡

否录

是录

确认支付

修改资料商品

浏览商品商品

立即购买商品

分享商品

加入购物车商品

进入店铺

登录

浏览商品

登录注册

否录

是录

找回密码商品

编辑购物车商品

填写订单信息

绑定银行卡

查看订单

确认收货，进行评价

结束购物

### 业务流程图

图4.系统全局业务流程图

### 数据字典

数据字典是把以上所分析的整体数据流中的数据信息加以定义和解释说明，让开发人员对系统的业务逻辑有更加深刻地理解，从宏观和微观两个方面对系统加以把握。更重要的是，在后期的调试运行中，也更有利于测试人员进行调试和排查系统漏洞。因此，在开发过程中，我们必须严格、规范地定义每一个具体的数据,形成系统专属的数据字典。

#### 用户分销信息表

作用：用于记录前台的用户个人信息

表1.用户分销信息表

|  |  |
| --- | --- |
| 数据存储名： | 用户信息 |
| 说明： | 记录用户的基本信息，如用户编号、用户账号、用户密码等关键信息 |
| 数据描述： | 用户ID+用户账号+用户密码+用户昵称+真实姓名+性别+地址+手机号码+头像+用户类别 |

#### 管理员信息表

作用：用于记录系统后台管理员的用户个人信息

表2.管理员信息表

|  |  |
| --- | --- |
| 数据存储名： | 管理员信息 |
| 说明： | 记录管理员的基本信息，如管理员编号、管理员姓名、管理员密码等信息进行记录 |
| 数据描述： | 管理员编号+管理员姓名+管理员账号+管理员密码 |

#### 用户银行卡信息表

作用：用于记录用户在系统中绑定的银行卡信息

表3.用户银行卡信息表

|  |  |
| --- | --- |
| 数据存储名： | 用户银行卡信息 |
| 说明： | 记录用户银行卡的基本信息，如用户银行卡编号、用户银行卡账号、用户银行卡密码等关键信息 |
| 数据描述： | 用户编号+用户银行卡编号+用户银行卡账号+用户银行卡密码+用户银行卡数量 |

#### 购物车信息表

作用：用于记录用户加入购物车的商品信息

表4.购物车信息表

|  |  |
| --- | --- |
| 数据存储名： | 购物车信息表 |
| 说明： | 记录购物车的基本信息，如商品编号、商品名称、商品价格、商品介绍等信息 |
| 数据描述： | 商品编号+商品所属类别+商品名称+商品价格+商品介绍+商品品牌+所属店铺+商品数量 |

#### 商品信息表

作用：主要记录对于商城系统所拥有的商品信息

表5.商品信息表

|  |  |
| --- | --- |
| 数据存储名： | 商品信息 |
| 说明： | 记录商品的基本信息，如商品编号、商品所属类别、商品名称、商品价格、商品介绍等信息 |
| 数据描述： | 商品编号+商品所属类别+商品名称+商品价格、商品介绍+商品折扣+商品品牌+商品原价+商品现价+商品所属店铺+商品上新状态 |

#### 订单信息表

作用：主要记录用户提交的订单信息

表6.订单信息表

|  |  |
| --- | --- |
| 数据存储名： | 订单信息 |
| 说明： | 记录订单的基本信息，如订单编号、订单所属用户、收货人姓名、收货人地址、收货人邮编、下单日期、订单状态等信息 |
| 数据描述： | 订单编号+订单所属的用户名称+收货人姓名+收件人地址+收件人邮编+收件人联系方式+订单提交日期+订单当前状态+订单物流信息 |

#### 订单中的商品信息表

作用：主要记录对于订单中详细购买商品的信息

表7.订单商品信息表

|  |  |
| --- | --- |
| 数据存储名： | 订单商品信息 |
| 说明： | 对订单商品的基本信息，如订单商品的编号、商品编号、所属类别、商品名称等信息进行记录 |
| 数据描述： | 订单编号、订单商品的编号、商品编号、所属类别、商品名称、商品价格、商品介绍+商品折扣、商品品牌、商品原价、商品现价、订单购买的商品数量 |

#### 店铺信息表

作用：主要用于记录店铺的详细信息

表8.店铺信息表

|  |  |
| --- | --- |
| 数据存储名： | 店铺信息 |
| 说明： | 对店铺的基本信息，如店铺编号、店铺名称、店铺所属类别、店铺logo等信息进行记录 |
| 数据描述： | 店铺编号+店铺名称+店铺所属类别+店铺logo+店铺简介+店主信息+店主联系方式 |

#### 商城收益信息表

作用：主要用于记录商城的收益信息

表9.商城收益信息表

|  |  |
| --- | --- |
| 数据存储名： | 商城收益信息表 |
| 说明： | 对商城收益的基本信息，如店铺编号、店铺名称、收益详情等信息进行记录 |
| 数据描述： | 店铺编号+店铺名称+收益详情+新增客户数量+订单数量 |

#### 商城消息中心信息表

作用：主要用于记录商城的推送信息内容

表10.消息中心信息表

|  |  |
| --- | --- |
| 数据存储名： | 商城消息中心信息表 |
| 说明： | 对商城消息中心信息的基本信息，如消息编号、消息内容、消息时间等信息进行记录 |
| 数据描述： | 消息编号+消息内容+消息时间+消息状态+消息收件人+消息发件人 |

#### 评价信息表

作用：主要用于记录店铺收到的评价信息

表11.评价信息表

|  |  |
| --- | --- |
| 数据存储名： | 评价信息表 |
| 说明： | 对店铺销售商品的评价信息的基本信息，如商品编号、评价内容、评价时间等信息进行记录 |
| 数据描述： | 商品编号+评价编号+评价内容+评价时间+评价状态+评价人名称 |

### 数据流分析

数据流是指系统的用户，在使用系统的过程中产生的数据，这些数据在用户和系统之间的传递过程中形成了数据流动，称为数据流。

#### 整体数据流

买家

骚盒移动分销管理系统

卖家

图5.整体数据流图

#### 系统模块数据流

登录数据流

表12.登录数据流

|  |  |
| --- | --- |
| 数据流名称 | 数据流组成 |
| 名称 | 登录数据流 |
| 简述 | 用户在分销端和管理端登录时输入的账号和密码 |
| 来源 | 用户在分销端和管理端登录时输入的账号和密码 |
| 去向 | 前台分销平台或后台管理平台 |
| 组成 | 用户的账号+密码 |

商品信息数据流

表13.商品信息数据流

|  |  |
| --- | --- |
| 数据流名称 | 数据流组成 |
| 名称 | 商品信息数据流 |
| 简述 | 所有商品的基本信息 |
| 来源 | 用户浏览商城货搜索商品时输入的商品信息 |
| 去向 | 商品展示模块 |
| 组成 | 商品类别+商品名称或商品品牌 |

购物车数据流

表14.购物车数据流

|  |  |
| --- | --- |
| 数据流名称 | 数据流组成 |
| 名称 | 购物车信息流 |
| 简述 | 用户将商品添加到购物车后的基本信息 |
| 来源 | 商品的信息+用户信息 |
| 去向 | 购物车编辑模块 |
| 组成 | 商品编号+商品名称+商品单价+所选商品规格+购买数量 |

订单信息数据流

表15.订单信息数据流

|  |  |
| --- | --- |
| 数据流名称 | 数据流组成 |
| 名称 | 订单信息流 |
| 简述 | 用户提交订单填写的基本信息 |
| 来源 | 用户提交的订单基本信息+用户信息 |
| 去向 | 订单编辑模块 |
| 组成 | 订单编号+用户账号+收货人名称+收货地址+收货人联系方式 |

用户信息数据流

表16.用户信息数据流

|  |  |
| --- | --- |
| 数据流名称 | 数据流组成 |
| 名称 | 用户信息流 |
| 简述 | 用户或管理员的个人信息 |
| 来源 | 用户或管理员的个人信息 |
| 去向 | 用户个人信息模块 |
| 组成 | 用户或管理员编号+用户或管理员账号+旧密码+新密码 |

店铺信息数据流

表17.店铺信息数据流

|  |  |
| --- | --- |
| 数据流名称 | 数据流组成 |
| 名称 | 店铺信息流 |
| 简述 | 个人店铺的基本信息 |
| 来源 | 用户信息+店铺信息 |
| 去向 | 店铺信息编辑模块 |
| 组成 | 用户账号+店铺名称+店铺Logo+店铺介绍 |

销售信息数据流

表18.销售信息数据流

|  |  |
| --- | --- |
| 数据流名称 | 数据流组成 |
| 名称 | 销售信息流 |
| 简述 | 个人店铺销售情况的基本信息 |
| 来源 | 用户信息+店铺信息+销售信息 |
| 去向 | 店铺销售情况统计模块 |
| 组成 | 用户账号+店铺名称+销售数据+统计时间 |

用户银行卡信息数据流

表19.用户银行卡信息数据流

|  |  |
| --- | --- |
| 数据流名称 | 数据流组成 |
| 名称 | 用户银行卡信息流 |
| 简述 | 用户绑定的银行卡的基本信息 |
| 来源 | 用户信息+银行卡信息 |
| 去向 | 用户银行卡绑定管理模块 |

# 系统实现方案

## 实现步骤

1.准备好本系统开发所需要的硬件和软件条件：

硬件：一台win7 64位笔记本电脑；

软件：Chrome浏览器:用于Web 电子阅读器的展示，模拟手机以及调试代码；

Sublime Text 3:作为本系统编码主要工具，提供各种高效率插件以及快捷键；

Git Bash:版本控制工具Git在Windows下的命令行工具，用于项目版本控制，确保开发过程代码的更新以及维护；

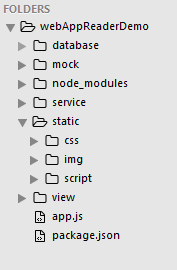
Node.js：JavaScript运行环境，用于便捷地搭建响应速度快、易于扩展的Web应用；

MongoDB：高性能的数据存储工具，用于该项目的数据存储；

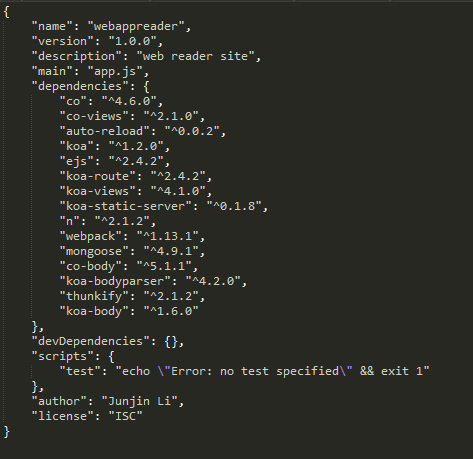
Robomongo: MongoDB的可视化工具，方便录入测试数据。



1. 新建项目文件夹webAppReaderDemo，在其下新建database文件夹用于存放跟数据库相关文件，mock文件夹用于存放调试过程的json模拟数据，node\_modules为安装项目依赖时自动上次的文件夹，service文件夹用于存放服务器端接口文件，static为静态资源文件夹，细分为css文件夹—用于存放项目样式文件，img文件--用于存放项目所需图片资源，script文件夹—用于存放项目的所有JavaScript文件，view文件夹用于存放所有视图文件，app.js作为整个项目的入口文件。



1. 项目依赖初始化



在整个项目目录下使用”npm install”命令行安装所有项目依赖模块，npm是一个包管理工具，开发人员可以根据自己需要去安装模块。以上所列出的模块就是本系统所需的全部模块，本系统为单页面应用加路由页面，所以我们引入了路由模块”koa-route”来实现路由，”ejs”为页面模板引擎，用于渲染页面，”mongoose”为MongoDB快速建模工具，”koa-static-server”是用于集成项目静态资源。

1. 项目入口文件app.js开发

var koa = require('koa'); //koa中间件

var controller = require('koa-route');

var app = new koa(); //渲染模板

var views = require('co-views');

var render = views('./view', { //view是存放路径

map: {html: 'ejs'} //模板引擎

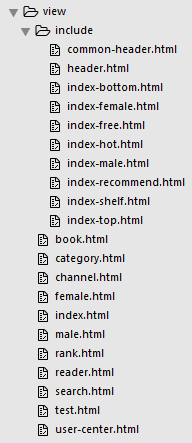
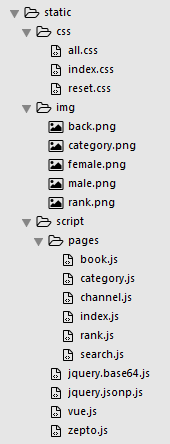
});

app.listen(3005);

console.log("Koa server is started!");

项目端口为3005，在项目目录下运行命令行”node app.js”,若koa服务开启成功，看到命令行控制台打印输出” Koa server is started!”。

1. 创建视图层文件

根据功能需求分析，因为本系统为单页面应用，书城主页面为index.html,其他均为路由页面，首页结构相对复杂，我们使用组件化方式开发，首页组件存放在include文件夹中，开发过程使用ejs模板的语句即可引入，如下所示：

<div class="container-scroll" style="top: 0px;">

<% include include/index-top.html%>

<% include include/index-hot.html%>

<% include include/index-recommend.html%>

<% include include/index-female.html%>

<% include include/index-male.html%>

<% include include/index-free.html%>

<% include include/index-bottom.html%>

</div>

阅读器开发为单独一整个模块，写在reader.html中，等开发完毕再集成到项目中。

页面样式开发方面，为了达到性能最佳，减少静态资源请求次数，我们将具备相同样式的元素使用同个类名，集成样式，这样子可以减少代码冗余，其中，把首页所有样式写进”index.css”,其他页面的样式写进“all.css”，”reset.css”用于重置HTML标签在浏览器里面的默认样式。

在script文件夹中有pages文件夹用于存放页面相关的js文件，其余的为开源js库。

1. 页面路由开发及测试前端流程

开发过程中随时对页面进行测试是一件必要且重要的事情，在我们还没将全部页面开发完成时，我们需要随时对页面进行测试。页面路由添加方式如下，在app.js文件里添加如下代码，其他页面路由也按照以下方式添加：

app.use(controller.get('/', function\*(){ //默认访问到首页

this.set('Cache-Control', 'no-cache');

this.body = yield render('index', {title: '书城首页'});

}));

app.use(controller.get('/search', function\*(){

this.set('Cache-Control', 'no-cache');

this.body = yield render('search', {title: '搜索', nav: '搜索'});

}));

在index在浏览器输入“http://127.0.0.1:3005/ ”，若能访问到首页文件，说明页面打通了。

1. 设计数据库模型

使用MongoDB的快速建模工具Mongoose，先连接到我们的本地数据库maplebookshop，在app.js文件夹里面创建连接数据库方式如下：

var mongoose = require('mongoose');

var db = mongoose.connect('mongodb://localhost/maplebookshop');//连接数据库

db.connection.on("error", function (error) { console.log("数据库连接失败：" + error); });

db.connection.on("open", function () { console.log("------数据库连接成功！------"); });

在Mongoose中有模式Schema,模型Model,文档Documents三个概念，建模过程如下：

/\*User表start\*/

//创建模式（相当于骨架），但是该模式还未和user集合关联

var Schema = mongoose.Schema;

var userSchema = new Schema({

name: String,

password: String

});

//创建User模型

var User = mongoose.model('user', userSchema);

//创建文档实体

var admin = new User({

name: "lijunjin",

password: "maple"

});

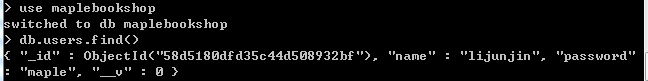
//保存文档实体

admin.save();

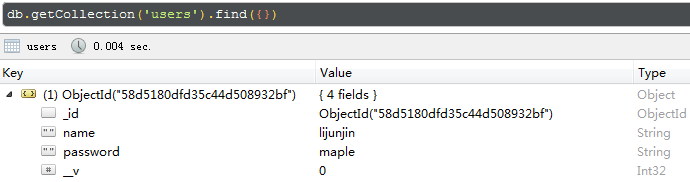
/\*User表end\*/

1. 在MongoDB客户端测试数据

为了确认数据是否被成功存储至数据库，在MongoDB客户端键入如下命令行即可查看数据，由下可见数据已被成功保存至数据库：



也可以使用MongoDB可视化工具RoboMongo来查看数据，更加方便直观：

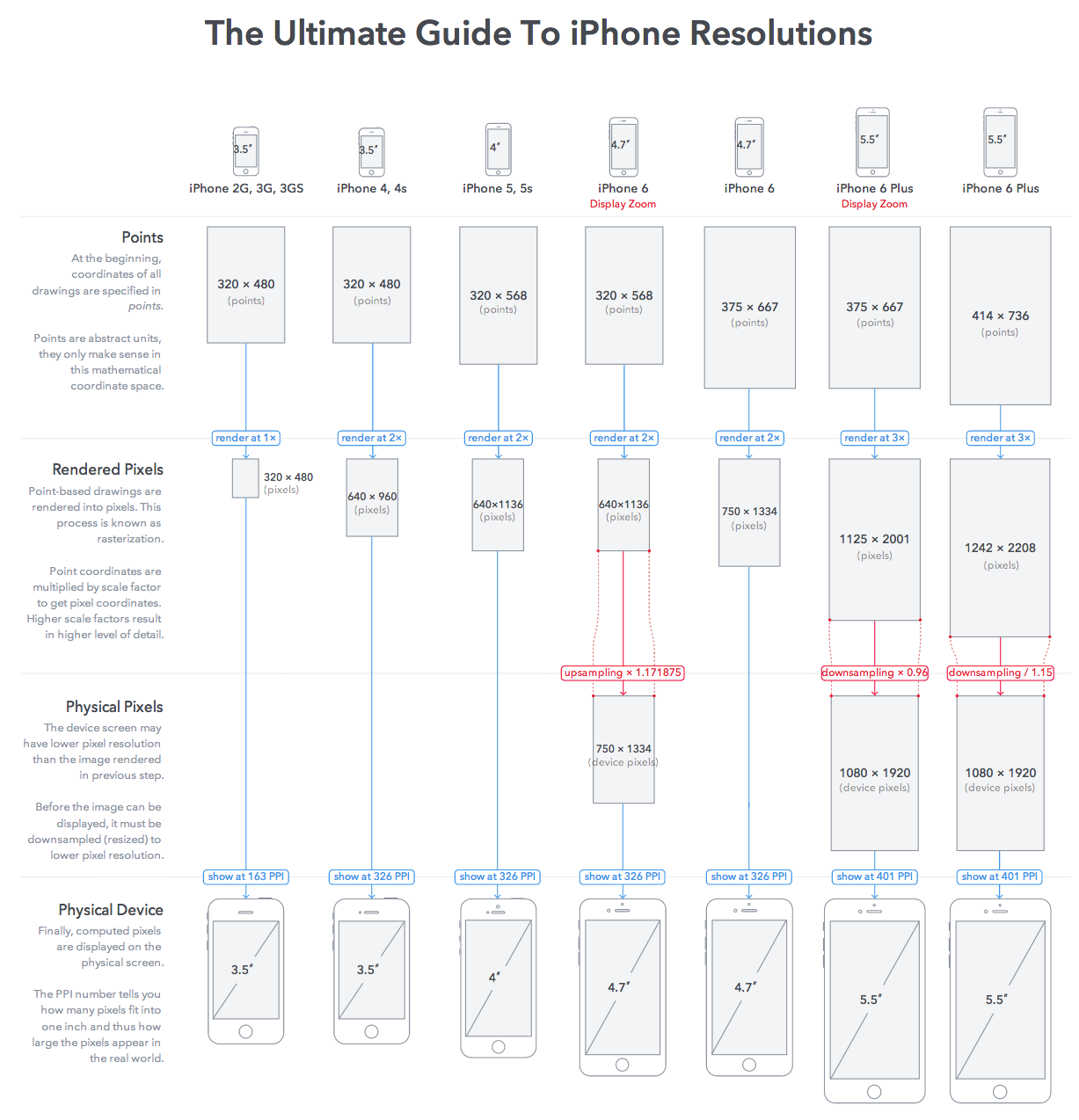


1. 开发后端逻辑

登录注册逻辑

书籍搜索逻辑

1. 页面自适应屏幕宽度代码改造



完成页面后需要对页面的功能以及样式进行调试，上图为网上出现的移动端需要适配的设备，由于移动端设备多，屏幕分辨率差距大，因此我们不能对一些要适配不同手机的结构写固定宽度，因此我们对首页header做自适应调整，如下所示：

1. 设计并提取出系统共用的开发文件，减少代码冗余，缩短工作流程。如：在css文件夹下新建common.css文件编写系统中复用率较高的导航栏和头部样式。
2. 将新建的saohe文件夹导入GitHub软件，在每个开发周期结束之后做一次版本更新工作；



## 响应式适配的实现

响应式布局的概念本身就是为了解决移动互联网的浏览问题而诞生的。它的设计原则旨在使一个网站能够兼容多个设备类型，而不是专门地为某种设备终端做一个唯一的版本。响应式布局能够为用户提供更加舒适的界面，同时也能提高用户的使用体验。

本项目运动的移动开发框架FrozenUI，其本身就拥有完美的响应式布局设计。但是，在我们系统进行开发设计的时候，并不能完全地依赖它的这种便捷，毕竟还有很多特定的设计是需要较大的自主性的。

**实现思路：**

本系统实现响应布局的方法是里利用rem**[７]**代替px作为界面元素尺寸的单位，rem的特点是只要设置了html根部的固定尺寸标准，其下级元素的尺寸即可根据这个标准进行相对大小设置。根据不同尺寸和分辨率大小的手机屏幕，我们需要根据浏览本系统的手机规格来设置这个标准，因此我们首先要获取的是浏览本系统的手机屏幕规格的参数信息。(ｃｓｓ３的媒体查询**[８]**功能也是实现响应布局的方法之一)

实现此功能的js代码如下：

(function (doc, win) {

var docEl = doc.documentElement,

resizeEvt = 'orientationchange' in window ? 'orientationchange' : 'resize',

recalc = function () {

var clientWidth = docEl.clientWidth;

if (!clientWidth) return;

docEl.style.fontSize = 20 \* (clientWidth / 375) + 'px';

};

if (!doc.addEventListener) return;

win.addEventListener(resizeEvt, recalc, false);

doc.addEventListener('DOMContentLoaded', recalc, false);

})(document, window);

**代码注释：**

定义一个函数来实现改变根据不同屏幕大小和分辨率设置html根部的font-size**[8]**属性值。首先取得html根部节点存进docEl变量；接着通过三元运算进行判断，若用户翻转了屏幕则根据翻转后的屏幕规格为标准，否则根据手机点钱屏幕尺寸进行设置；取得设备当前的屏幕宽度，若无法取得则直接返回不再执行；根据取得的宽度值与标准屏幕的宽度进行比值运算（在此将标准屏幕的宽度定为iphone 6的375px），再乘上20，表示在iphone6下的标准大小是20px，其他的屏幕尺寸则根据这个比例进行缩放设置；最后将函数加载在文件窗口的监听器上进行实时监听设置。即可实现响应式适配功能。

## 注册功能的实现

注册功能界面包含用户注册输入的基本信息输入框，包括账号（账号为用户的手机号码）、验证码、密码和重复密码，以及两个按钮：发送验证码按钮和确定注册按钮。

注册界面如下图所示：



图6.注册界面图

**实现思路：**

html5新增了一些输入框的属性，利用这些属性可以大大简化开发过程。注册输入框的内容要求都不能为空，通过设置input元素的required**［９］**属性进行输入是否为空的判断；正确的手机号格式为11位的纯数字，根据用户的输入进行判断其合法性；注册码要求与手机端接收到的号码一致，点击发送验证码后在一定时间后才可以再次点击发送验证码；对密码的输入无严格限制，要求密码和重复密码的两次输入保持相同即可。点击确定按钮后对上述输入的合法性进行判断。

注册功能的核心代码如下：

//设置验证码重新发送时间

var timer, flag=true;

$("#sendreg").tap(function(){

var that=$(this);

if(flag){

clearInterval(timer);

$(this).addClass("disabled");

var el;

el=$.tips({

content:'您的验证码为：123',

stayTime:2000,

type:"success"

})

el.on("tips:hide",function(){

})

var t=60;

timer=setInterval(function(){

if(t>1){

t--;

that.text(t+"s后可再发送");

flag=false;

}else{

that.text("发送验证码").removeClass("disabled");

clearInterval(timer);

flag=true;

}

},1000)

}

});

**代码注释：**

设置一个定时器，每隔60秒重新获得点击“发送验证码”的权限。首先，定义一个定时器和一个断点标示；当点击“发送验证码”按钮的时候触发函数；先清除定时器以防止定时器的重复叠加造成功能紊乱；点击按钮后按钮的颜色要变成灰色不可点击状态，故给按钮增加定义好的“disabled”类；定义弹出对话框的内容信息，设置其显示停留的时间为2秒；定义重新发送验证码的时间值t,设置一个每隔一秒调用一次的定时器，其作用是改变发送验证码按钮上的内容，每调用一次时间值t减1，当t的值减为0时，“发送验证码”按钮重新设置为可点击状态，并清除定时器。

//检查账户号码格式的合法性

function check() {

var phone=$$("phone").value;

var reg\_num = $$("reg\_num").value;

var pwd = $$("pwd").value;

var pwd1 = $$("pwd1").value;

if(!(/^\d+$/.test(phone))) {

var dia=$.dialog({

title:'【提示】',

content:'手机号码格式错误!',

button:["确定","取消"]

});

return false;

}

if(phone.length!=11){

var dia=$.dialog({

title:'【提示】',

content:'手机号码长度错误!',

button:["确定","取消"]

});

return false;}

**代码注释：**

注册时用户的账号要求是手机号码，因此账号格式必须和手机号码的格式一致。手机号码的格式为11位纯数字，因此，如果账号里面出现非数字字符，或者账号字符长度不足11位或超过11位都是不规范不合法的输入，要对这类输入进行提示。首先，先取得用户输入的账户值，对取出来的值进行正则表达式格式验证，正则表达式的格式设置为纯数字格式，如果验证的返回值为false，则提示用户输入格式错误；否则再对其长度进行验证，若长度值为11位数，则验证通过，否则弹出提示信息通知用户输入格式错误。

//判断两次输入的密码是否一致

if(pwd!=pwd1){

var dia=$.dialog({

title:'【提示】',

content:'两次输入的密码不一致!',

button:["确定","取消"]

});

return false;

}

var timer,flag=true;

}

});

**代码注释：**

为了防止用户单次输入密码时的不当操作导致设置的密码与用户实际需要设置的密码不同，增加了一个“再次输入密码”的输入框要求用户再次输入密码，通过这样来保证用户设置的密码的正确性。此项验证原理很简单，分别取出两次输入密码的值，将两个值进行比较，若两个值相等，证明输入的密码一致；若不相等，证明两次输入的密码不一致，至少其中一个密码输入错误，弹出错误信息提醒用户两次输入的密码不一致，要求用户重新输入密码。

## 购物车编辑功能的实现

购物车功能模块要实现对购物车内的商品实现单个勾选、全选、选择结算以及逻辑删除的操作。购物车里面的内容包括商品所属店铺，商品图片，商品名称，商品数量，商品单价以及商品在购物车里的状态，另外在其底部有结算购买按钮，实现对购物车的商品进行购买。

购物车的页面设计如下图：



图7.购物车界面图

**实现思路：**

购物车里面的商品前设置一个用来勾选的按钮，这个按钮有两个状态，一是实心高亮显示的勾选状态，另外一个是空心显示的未勾选状态。点击时按钮应在勾选和未勾选这两种状态中实现切换。当购物车里属于同一个店铺里面的商品都被勾选时，店铺名称前的勾选按钮即设为选中状态。结算价格要根据选中的商品的价格进行运算得出总价。

购物车的核心js代码如下：

//检查购物车的商品是否被全选

var all=false,

allSe=$(".cart-list .selectIcon"),

allBtn=$(".footer-btn .selectIcon"),

allDelBtn=$(".footer-btn2 .selectIcon");

function checkAll(){

var sum=0;

for(var i=0;i<allSe.length;i++){

if($(allSe[i]).hasClass("selected")){

sum++;

}

}

if(allSe.length==sum){

all=true;

}else{

all=false;

}

}

**代码注释：**

购物车里的商品可能不止一个，用户勾选商品进行结算购买，也可以点击“全选”按钮对购物车的所有商品进行结算购买。这就需要对用户的勾选操作进行判断。定义一个数组变量存放购物车里面的商品项，定义一个变量all实现全选状态判断，默认值为false，全选状态为true；当用户勾选商品前的按钮时，给按钮增加一个选中的样式“selected”；循环遍历数组当中拥有“selected”样式类的按钮数量，定义一个变量sum记录这个数量的多少；当这个数量等于购物车下所包含的商品数量时，表示商品被全部选中；修改全选判断标示all为true，否则为false。（关于selected样式类的设置的实现方法很简单，给按钮设置两种不同的背景即可。即未选中时是空白背景，选中后设置背景为相应的表示选中状态的图片。所以只需要在selected样式类中设置background属性就可以了。）

//检查店铺下的商品是否全选状态

function checkShop(that){

var s=that.parents(".cart-list").children().children(".proSelected").size();

var obj=that.parents(".cart-list").children().children(".proSelected");

var n=0;

if(that.hasClass("selected")){

for(var i=0;i<s;i++){

if($(obj[i]).hasClass("selected")){

n+=1;

}

}

if(n==s){

that.parents(".cart-list").children().children(".shopSelected").addClass("selected");

}else{

that.parents(".cart-list").children().children(".shopSelected").removeClass("selected");

}

}else{

that.parents(".cart-list").children().children(".shopSelected").removeClass("selected");

}

}

**代码注释：**

每个商品有其所属的店铺信息，购物车里的商品的都是属于某个店铺。因此也要判断购物车里的店铺下的商品是否全选状态，需要定义一个函数去检查每个店选框勾选还是去掉。代码原理与检查全选的代码是一致的。同样是定义一个样式类“selected”表示选择状态，当店铺被选中则给勾选按钮加上这个类，否则去除；取得店铺下的商铺存进一个数组变量中，循环判断数组中含有“selected”类的个数；当个数等于店铺下所有商品的个数时，店铺前的勾选按钮要被设置为选中状态，即加上“selected”类，否则去除此类。

//全选付款

allBtn.tap(function(){

checkAll();

if(!all){

$(this).addClass("selected");

allSe.addClass("selected");

all=false;

}else{

$(this).removeClass("selected");

allSe.removeClass("selected");

all=true;

}

});

**代码注释：**

点击“合计”按钮即将购物车里的所有商品勾选进行结算。在这里可以充分利用之前定义的判断全选的函数功能，执行完判断函数后全选标识符all的状态将会被改变，根据all的最终状态进行判断。首先执行一次判断全选的函数；接着判断全选标识符的状态，如果其为false，表示购物车里面的商品并未全部被选择，此时通过给未选择的设置“selected”样式类改变其勾选状态，并改变all的值为true；如果其为true，表示购物车的商品已经被全选，因此当点击全选按钮时，逻辑上判断应该是取消全选状态。

//全选删除

allDelBtn.tap(function(){

checkAll();

if(!all){

$(this).addClass("selected");

allSe.addClass("selected");

all=false;

}else{

$(this).removeClass("selected");

allSe.removeClass("selected");

all=true;

}

});

**代码注释：**

全选删除的逻辑和全选付款的逻辑是一致的。在执行判断全选状态的函数后根据全选标识符的状态进行处理。在此不再重复赘述。

## 瀑布流显示的实现

在个人日记栏目下，用户可以查看过往记录的日记内容，包括日记的作者、发布时间、发布内容、日记附属图片、以及日记的点赞数量等信息。

个人日记界面、瀑布流加载显示时的界面分别如下图所示：



图8.瀑布流显示界面图

**实现思路:**

随着用户日记数量的增加，页面显示的内容高度也将随之增加。如果一开始要求系统加载用户的所有日记的内容，必将造成响应时间的延长，影响用户的使用体验。解决这个问题的办法是，一开始只加载一部分的日记内容，在用户拖动滚动条往下拉的时候，加载下一部分日记内容。所以需要实现定义好“部分”的容量是多少。在加载的过程中，应该弹出提示框提醒用户正在加载内容。本系统通过stick.js框架实现瀑布流显示效果，编写一行代码<script type="text/javascript" src="js/lib/stick.js"></script>引入stick.js框架.

瀑布流的实现代码如下：

// 瀑布流动态加载数据

/\*\* 模拟ajax加载，提供假数据 \*\*/

var loadTime = 0;

function loadMore(){

if(++loadTime >= 8){

return;

}

setTimeout(function(){

var el=$.loading({

content:'加载中...',

})

setTimeout(function(){

el.loading("hide");

},300);

list = [{}];

list.forEach(function(item) {

var html='<div class="diary-list">'+

'<div class="desc hide">'+

'<p>'+

'盛小游戏 雷恩的饰品——狼齿 这是雷恩在野外遇到野狼把狼打死并一举拔下 这是勇士的象征！这是勇士的象征！这是勇士的象征！'+

'</p>'+

'<div class="published-time">'+

'<span>201601.02 12:21:11</span>'+

'</div>'+

'<button class="descBtn">展开</button>'+

'</div>'+

'<div class="diary-list-bd">'+

'<ul class="imglist imglist-1">'+

'<li><img src="images/upload/pro.jpg" alt=""></li>'+

'</ul>'+

'<ul class="imglist imglist-2">'+

'<li><img src="images/upload/pro.jpg" alt=""></li>'+

'<li><img src="images/upload/pro.jpg" alt=""></li>'+

'</ul>'+

'</div>'+

'</div>'+

'</div>'

stick.addItem(html, item.cover);

});

},300);

}

//定义瀑布流组件

var stick = new Stick({

container: document.getElementById('container'),

//column\_width: 200,

column\_gap: 0,

onNeedMore: loadMore

});

loadMore();

</script>

**代码注释：**

用户的日记数量是有限的，当对用户的所有日记加载完成后便不需要再进行瀑布流加载了。首先，定义一个变量loadTime来记录这个数量（在此假设用户的日记数量为8，数量大于8时则不再进行函数的调用）；定义一个定时器在300毫秒后执行，作用是弹出对话框“加载中...”，提醒用户系统正在加载日记，提升用户体验；再定义一个数组变量来存在要加载的日记内容，在这里由于缺少后台的支持，于是将日记内容写成事先准备好的固定内容（当具备后台支持时，只需将这部分内容替换成从数据库调取的内容即可）；对数组进行遍历操作，每次遍历时把内容存进html变量中即可；最后，声明并定义一个瀑布流组件，设置相关参数后执行之前定义的瀑布流加载函数，这样就实现了瀑布流加载显示日记内容的功能了。

## 商品轮播展示功能的实现

在商品详情界面里，用户可以浏览商品的全部信息和介绍。包括商品的图片、价格、库存量、商品评价等信息内容。为了实现商品更好的展示效果，需要设置一个轮播图**［１０］**来滚动显示商品的主要样片。

商品的界面如下图所示：

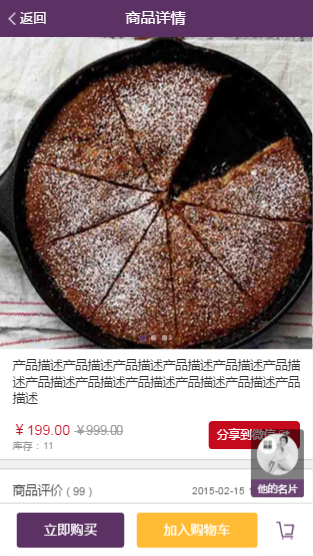


图9.商品轮播界面图

**实现思路：**

商品的展示图片是从右向左滚动的，每次滚动只显示一张图片，在一定的时间间隔后显示下一张。这个循环显示的过程并非一次性的，而是无限循环的。当然用户也可以根据需要通过操作控制图片的前后显示。在图片的下方设置一组小按钮，按钮个数和图片个数一样，高亮显示按钮来表示当前图片滚动到第几张了。

实现轮播显示的html代码和js代码如下：

<!-- html -->

<div class="ui-slider">

<ul class="ui-slider-content" style="width: 300%">

<li><span style="background-image:url(1.png)"></span></li>

<li><span style="background-image:url(2.png)"></span></li>

<li><span style="background-image:url(3.png)"></span></li>

</ul>

</div>

**代码注释：**

定义一个包含3个子项的无序列表存要轮播显示的三张图片（列表子项的数量根据需要显示的图片数量确定）因子项的个数是3，所以无序列表的宽度设置为300%。滚动时当前显示的界面宽度是100%，其余200%的图片内容将被遮挡。这样的原理就像通过一扇窗户看外界，外界的事物可以从窗户外任意改变位置，然而我们看到的永远是窗户大小的世界。

<!-- js -->

<script>

(function(){

var slider = new fz.Scroll('.ui-slider', {

role: 'slider',

indicator: true,

autoplay: true,

interval: 3000

});

})();

</script>

**代码注释：**

声明并定义一个新的轮播组件类型为fz.Scroll的slider变量，设置它的role属性值为slider，indicator值为true表示设置轮播的切换指示符为小圆点，autoplay的值为true表示设置轮播的方式为自动播放，interval的值为3000表示设置图片的轮播显示间隔为3秒。至此，商品轮播功能也就大功告成了。

# 系统运行测试

系统测试是在开发工作完成之后进行的。为了确保系统的稳定性与准确性，开发人员都要在系统开发初步完成后，将系统打包发给专门的测试人员进行测试。通过一系列测试程序，检验系统是否存在开发过程中被疏忽的漏洞以及功能的紊乱等问题。在经过测试——修改——再测试——在修改的循环过程，系统才能最终上线为用户所使用。

## 测试目的

1. 分析测试结果，得到系统质量评价
2. 研究系统可能存在的缺陷，为开发工程师提供修复和预防的建议
3. 对以后的软件开发避免系统缺陷有参考意义

## 测试对象

1. 注册登录
2. 用户购物浏览商品
3. 购物车功能
4. “我的”模块
5. 后台管理统计分析
6. 商品管理
7. 卖家销售信息

## 测试流程

1. 新用户进入注册界面先注册才能开始购物，注册页面如下图：



图10.注册页面

根据用户输入进行合法性判断，判断弹窗和发送验证码提示如下组图：

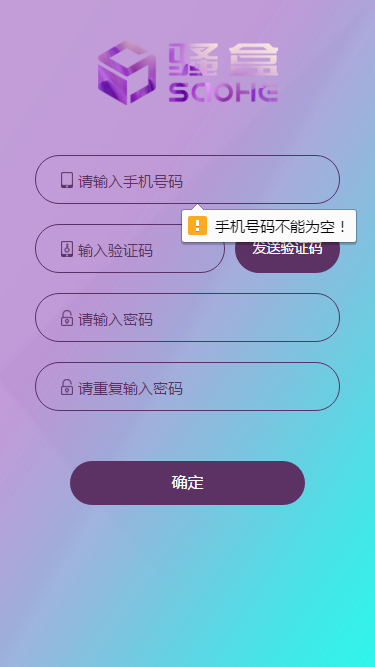


图11.判断弹窗图示 图12. 发送验证码提示

1. 注册成功后进入店铺首页，如下图：



图13.店铺首页

点击商品进入详情页，如下图：



图14.商品详情页面图

1. 点击加入购物车，商品进入购物车列表，如下图：

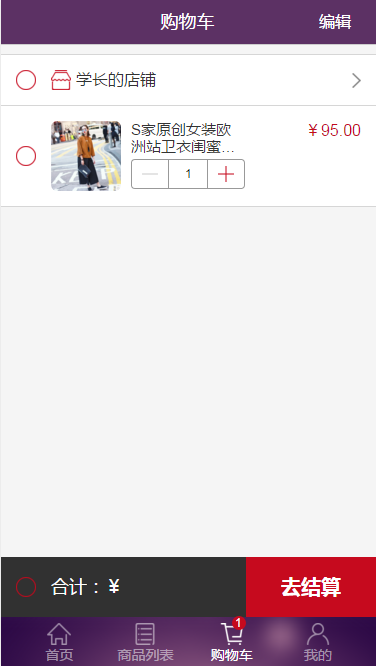


图15.购物车列表

点击“结算”按钮进入填写订单页面，如下图：



图16.填写订单信息

1. 点击“我的订单”进入填写订单页面，如下图：

图17.“我的”页面 图18.“我的订单”页面

1. 卖家进入后台管理首页可以看到店铺的销售统计信息，如下图：



图19.后台店铺收益统计图

1. 卖家进入商品管理菜单对店内商品进行上下架处理，如下图：



图20.商品管理界面

1. 卖家进入“我的”主菜单，查看管理店铺收益信息，如下图：



图21.“我的”管理菜单子项

## 测试结论

经过初步测试，系统的业务流程在逻辑处理上并没有什么问题。在功能的实现上，本系统基本满足用户的需求；动态效果的交互上，比较符合用户的使用习惯，并且具有一定的美观度。当然，没有什么事情是十全九美的。在页面的适配问题上，还需要做到精益求精。另外，在日后具备后台支持的情况下，会出现什么不可预知的错误，也有待探究。

# 结束语

利用HTML5开发网站是时下的热门潮流，特别是移动端的web开发。它不仅让开发工程师剩下了解决浏览器兼容问题的诸多麻烦，同时也让用户有了更好的使用体验。随着H5的发展，越来越多的企业将网站搬进了可移动设备中。相信今后掌握在用户手上的将会是更多更丰富的互联网世界。

本文论述了基于HTML5技术开发的移动电商网站，由于笔记从事前端开发工作，故对本系统的开发专注于页面设计与制作、页面布局的简洁美观、交互上的便捷性这些方面上。若要实现真正的用户数据的交互与存储，还需后台工程师的协助才能实现。在日后的工作生活中，我会不断学习提升自我技能水平，努力对本系统进行优化升级，完成其后台的开发工作。

通过一个多月的努力学习和工作，我独立地完成了利用HTML5技术的移动电商平台的设计开发工作和论文的撰写工作。从论文题目的确定，到亲手实践设计的实现，最后到论文的完成，每一步于我而言都是全新的尝试与挑战。虽然这个过程经历困难挫折和沮丧迷茫，但是论文写至此处之时，仍让笔记心生感慨，这必将是大学时光里难忘而又有意义的最后的旅程。“路漫漫其修远兮，吾将上下而求索。”

# 【参考文献】

【1】Adam Freeman.HTML5权威指南[M].北京：人民邮电出版社.2014：6.

【2】Elizabeth Castro,Bruce Hyslop. HTML5与CSS3基础教程[M].人民邮电，2014-05：209.

【3】David Flanagan.JavaScript权威指南[M].机械工业出版社，2012-10：32-60.

【4】[远人](http://www.imooc.com/u/101686/courses?sort=publish).移动端开发框架Zepto入门[在线].慕课网.http://www.imooc.com/learn/229：2016-05.

【5】腾讯手Q.Frozen UI[在线].github.http://frozenui.github.io/.2016-05.

【6】[姜维](http://www.imooc.com/u/102093/courses?sort=publish).Web调试工具[在线].慕课网.http://www.imooc.com/learn/137：2016-05.

【7】Nicholas C.Zakas. JavaScript高级程序设计[M].人民邮电.2012-03：56.

【８】大漠 Jack.十天精通CSS3[在线].慕课网.http://www.imooc.com/learn/33：2016-04.

【９】刘西杰 柳林.网页制作从入门到精通[M].人民邮电.2013-01：151-168.

【10】阿安.焦点轮播图特效[在线].慕课网.http://www.imooc.com/learn/18:2016-04.

# 致谢

在论文即将完结之际，我要在此对那些给予过我无私帮助的人们表示衷心的感谢。首先，，我要感谢我的指导老师范新灿副教授，感谢他由始至终对我实习工作的关怀和在我论文撰写过程中对我的指导。能够得到和如此明事理的老师学习的机会，给我的大学生涯画上了一个圆满的句号。在此祝他身体健康，工作顺利。

另外，我还要感谢我的公司和同事，感谢他们在我撰写毕业论文时对我的理解和支持，让我能够安心完成论文任务，也感谢我的同事在我开发过程中为我答疑解惑。没有他们的帮助和理解支持，此项目就无法顺利地如期完成。

最后还要感谢慕课网，正是因为有这样一个专注于互联网技能学习的教学网站，帮助我提升了自身技能，学习到了很多有用的知识。在此祝愿你们把网站建设得越来越好。

# 【Abstract】

At the beginning of the 21th century, Internet has developed more and more quickly. Since the age of the Internet is coming, it has greatly changed in all aspects of people’s lives. Currently, the pattern called “Internet +” is imperceptibly slipping into people’s lives, especially the trend of Electronic Commerce, which bases on the development of the Internet.

The paper is discussing about the electronic shopping website system of SAOHE, which is a mobile electronic shopping platform depending on the HTML5 to develop. In the one hand, when you are a buyer in the platform of the sale, you can register and login on the website and then begin your shopping trip. When some products catch your eyes, you can freely push your favorite products into the shopping cart without wasting your energy. If you make a decision to buy it, you can manage the orders and write down your information to make sure that the product can be send to you rightly. When you complete one order, you can write down your evaluation. Also you are able to manage your information and so on. In the other hand, while you are a manager in the platform of the management, it is clearly to see the sale data about your shop. Of course, you can manage the goods on yourself shop, on sale or off the sale. You can also manage the orders and the type of the products of the business. Meanwhile, the information of manager and users can be edited and so on.

The system bases on the pattern which is called “B/S” structure, with a series of the develop languages, such as HTML5, CSS3, JavaScript and so on. In the same time, Zepto.js and Frozen UI are used to assist the development of the system.

Last but not least, the system is a comprehensive and mobile electronic shopping system, which is primarily developed for the wechat users. It will be useful and helpful for the users to their business activities.

【Keyword】

SAOHE; e-commerce; HTML5, CSS3, JavaScript;

指导教师：范新灿 副教授