

湖南安全技术职业学院

Hunan Vocational Institute of Safety Technology

课程设计成果说明书

（全日制专科生）

毕业设计题目：	基于 Vue2 京西商城设计与实现
专 业 班 级：	移动互联应用技术 2201 班
二 级 学 院：	信息工程学院
学 生 姓 名：	陈 永 博
指 导 老 师：	刘 湘 黔
专 业 带 头 人：	周 文 君

教务处制

撰写说明

一、评阅人至少为 2 人，必须为校内在岗在编教师，并承担评阅责任。二、设计思路：概括性地介绍毕业设计的基本情况，包括选题背景、

设计意义和价值，设计的基本目标和设计思路等方面。

三、设计内容：根据项目实施流程，把项目在需求分析、概要设计、详细设计、项目实现、项目测试（评价）等环节中具体工作内容和任务阐述清楚。

四、设计过程：阐述在该毕业设计过程中的主要时间节点、主要工作内容、主要设计和实验实践活动、工作方法、工具和手段，毕业设计过程中遇到的问题及解决方法等方面。

五、作品及特点：阐述本次毕业设计的最终作品及其表现形式，并总结设计作品的主要特点。方案、作品等作为附件。

六、参考资料：毕业设计中参考和引用的文献、书籍、技术标准和网络资源等。

七、参考格式

1. 版面 A4，页边距上 3.7cm，下 3.5，左 2.8cm，右 2.6cm。页码置于页底边，居中。

2. 正文小四号宋体；章节标题 4 号，黑体，顶格；小节标题为小 4 号，黑体，顶格；表、图标题 5 号，黑体；表、图内容 5 号，宋体；“参考资料”小 4 号，黑体，居中；参考文献内容 5 号，宋体，顶格。

3. 行距为 21 磅，章节标题上下均空一行，小节标题上空一行。

4. 具体格式可由二级学院自行制定。八、A4 纸双面打印。

毕业设计独创性声明

本人郑重声明：所提交的毕业设计成果是本人在指导教师的指导下，独立完成所取得的成果。内容真实可靠，不存在剽窃和抄袭他人成果的不端行为。除在成果报告书中注明引用的内容外，本毕业设计不含其它个人或集体已经发表或者撰写过的成果。如发现毕业设计中存在抄袭、剽窃等不端行为，本人愿意承担本声明的法律责任和一切后果。

学生姓名：陈永博 日期：2024-1-5

目录

1. 设计思路.....	2
1.1 选题背景.....	2
1.2 设计意义和价值.....	2
1.3 设计目标.....	3
1.4 设计思路.....	3
2. 设计内容.....	4
2.1 需求分析.....	4
2.3 详细设计.....	6
2.3.1 引导界面.....	6
2.3.2 登陆注册界面.....	7
2.3.3 学习模式界面.....	9
2.3.4 中英学习、单词分析界面.....	11
2.3.5 英中学习、选词填空界面.....	13
2.3.6 拼写学习、拼写填空界面.....	15
2.3.7 单词复习、学习记录界面.....	18
2.3.8 已学习、已学会界面.....	20
2.3.9 未学习、我的界面.....	21
2.3.10 个人主页界面.....	22
2.3.11 评价打分、意见反馈界面.....	24
2.3.12 选择词库、词汇量界面.....	25
2.3.13 错题记录、我的收藏界面.....	27
2.4 项目测试.....	28
3. 设计过程.....	29
3.1 工作进度安排.....	29
3.2 实施方法及手段.....	29
3.3 遇到的问题及解决方案.....	30
4. 作品及特点.....	31
4.1 作品展示.....	31
4.2 作品特点.....	34
参考资料.....	35
致谢.....	36

1. 设计思路

1.1 选题背景

Vue是一个独立的社区驱动的项目，它是由尤雨溪在2014年作为其个人项目创建，是一个成熟的、经历了无数实战考验的框架。Vue（发音为 /vju:/，类似 view）是一款用于构建用户界面的JavaScript框架。它基于标准HTML、CSS和JavaScript构建，并提供了一套声明式的、组件化的编程模型，帮助开发者高效地开发用户界面。

Vue是目前生产环境中使用最广泛的JavaScript框架之一，可以轻松处理大多数web应用的场景，并且几乎不需要手动优化，并且Vue完全有能力处理大规模的应用。本次设计的作品就是基于Vue2框架的京西商城。

1.2 设计意义和价值

随着互联网技术的迅速发展和普及，电子商务已经成为全球经济发展的重要引擎。在线购物商城作为电子商务的主要形式，为消费者提供了一个便捷、高效的购物平台。设计与实现一个基于vue2框架的在线购物商城网站，不仅可以满足消费者的购物需求，还可以为企业提供更广阔的市场和更多的商业机会。因此，本研究具有重要的现实意义和商业价值。

本设计的是一款基于vue2的京西商城，通过这一项目的设计开发，我严重发现了自己在专业知识学习中存在着的各种问题，也深刻发现了在项目开发中自我能力的不足，还需要更多的努力。并且让我学习到更多未曾接触到的专业知识，让我的思想觉悟也更上一层楼。

1.3 设计目标

本项目是一款基于Vue2框架的京西商城，界面要求简洁明了，减少不必要的资源浪费。利用 vs code IDE 结合 JavaScript 编程语言进行设计。要求具备电商购物的基本功能，能适用于各种类型手机，系统能够正常运行，各个页面交互性较好。本项目要求至少实现以下目标。

(1) 本项目要求实现用户可以通过登陆获取 token 并保存用户登录数据，能够随时随地进行退出登录。

(2) 实现网站的各种记录、订单、个人信息、购物车、商品分类、广告轮播、商品评论等功能。

(3) 实现在需要登录 token 权证的页面验证是否登录 并跳转目标页面或提示用户需要登录。

(4) 实现能随时查看，搜索，购买，评价商品。

1.4 设计思路

(1) 通过文献研究法在互联网上了解项目的相关研究现状以及问题的解决方法。该系统在 Windows 10 操作系统环境中使用的 JavaScript 编程技术，采用前端开发工具 vs code，来开发基于 vue2 框架京西商城。

(2) 将本项目分为一个个细小模块分别进行完成，再对其进行组装整合形成一个完整且有关联的项目。本项目将分为首页界面（主页面）、登陆界面、个人中心界面、搜索页面、商品界面、商品分类界面、商品详情页面、购物车界面、订单页面界面、支付页面界面十个页面，每个模块都该有相互的联系。

(3) 本项目数据来源于一个外界的API接口，系统通过获取此API接口数据，将数据显示在单词学习模式中。

(4) 本项目作为一个商城项目，在 项目设计界面时必须做到简单大方、不过分花里胡哨和繁琐，且应当做到色调统一，让用户在进入界面时心情是愉悦的、眼睛是放松的，并且各个功能的展示也必须应当做到能让用户一见便知其意且使用方法简单便捷。

2. 设计内容

2.1 需求分析

本项目用户可以手机在身上的前提下无时无刻不分地点进行商品选购，大大方便了因为足不出户或随时随地选购商品，提高了购物的效率。本项目运用 vs code 进行项目开发。本项目是一款基于 vue2 框架开发的商城APP，界面简单大方、通俗易懂，内存空间小，尽可能地减少了一切本不应该浪费的资源，符合绝大部分用户的要求，应该从其下几个方面来进行设计。

- (1) 本设计需要具有一定的安全性、独立性、准确性和可靠性，且所占用内存空间小。
- (2) 登陆界面的设计，用户能够通过登陆且保存数据和 token 权证。
- (3) 主界面（首页）的设计，尽可能要做到界面简单大方、通俗易懂，便于达到用户对主界面的各个功能一见便知其意的要求，便于用户的操作使用。
- (4) 分类界面的设计，用户能够通过点击不同的分类来查看不同类别的商品。
- (5) 购物车界面的设计，用户能够随时随地将自己喜欢的是商品加入购物车。
- (6) 商品详情界面的设计，用户能够对查看各种商品的详情信息、评论、商品图片。
- (7) 订单列表界面的设计，用户能够对自己的购买过的商品进行查看。
- (8) 搜索界面的设计，用户能够输入关键词搜索自己想查看的商品。
- (9) 支付界面的设计，显示用户想购买商品的价格、数量，邮寄地址，优惠等；用户可在此页面支付订单
- (10) 个人信息界面的设计，用户能查看自己的个人信息 如手机号、余额、并提供退出登录功能。

2.2 概要设计

本商城 APP 的设计是直接使用于用户的，所以界面的设计需要简单大方、通俗易懂，让用户对于看见此 APP 的功能能够达到一见便知其意的效果，由此方便用户对此 APP 的操作与使用。本 APP 是一款基于 vue2 框架开发的商城APP，它的主要功能模块有其下几个方面。

(1) 首页界面

程序启动首先进入首页，首页显示推荐商品轮播图、热门分类导航、热门商品列表。

(2) 商品分类界面

点击不同的分类展示不同的商品列表，用户可以通过底部导航栏的选择来跳转到此界面。

主要分为包括热门推荐、手机、电脑、冰箱、洗衣机、电视等19个大分类，每个大分类下又有若干个小分类。点击分类都会跳转到相应分类的单词商品列表界面。

(3) 购物车界面

该页面展示用户添加到购物车的商品

主要有删除、结算和选择购物车商品三个功能。

(4) 个人中心界面

展示用户个人信息。

(5) 商品列表界面

展示不同分类商品列表，或根据用户搜索关键词来进行匹配的商品列表。

(6) 搜索界面

提供给用户搜索商品的功能。

(7) 商品详情界面

展示商品详情信息，如图片、价格、评论等。

(8) 登陆界面

用户可以通过个人中心界面的登录按钮直接跳转到登陆界面进行登陆，或者在需要 token 权证的界面弹出的提示框跳转到登陆界面进行登陆。

(9) 支付界面

用户在此界面可进行提交订单的支付

(10) 订单列表

用户在此界面查看自己的订单列表

2.3 详细设计

京西商城 作为一个商城APP，设计APP界面时，应当以整洁、简单、大方为前提，保持各界面、各功能设计色调、风格统一。然后，操作系统需求应当尽可能的偏向于简单操作，应尽量做到让用户会使用手机便能懂得如何操作此APP。由此达到一个把商品选购为首要目标，而让用户在使用本APP购物的同时心情愉悦的目的。

2.3.1 首页界面

首页界面是第一个出现在打开APP后的界面，其主要目的是达到展示APP主页功能的效果，因此需要对

此界面保持一定的整洁大方。登录界面如图 2-2 示。



图 2-2 主界面图

本 APP 的首页界面有且只有 1 个，APP 打开后将会直接进入首页界面，在首页界面展示 APP 主页功能和信息，下部提供导航栏进行主要页面的跳转引导界面核心代码如下。

```

1 <template>
2 <div class="home" v-if="proList && navList && bannerList">
3   <!-- 导航条 -->
4   <van-nav-bar title="京西" placeholder="" />
5
6   <!-- 搜索框 -->
7   <van-search readonly shape="round" background="#ffffff" placeholder="请在此输入搜索关键词" @click="$router.push('/search')" />
8
9   <!-- 轮播图 -->
10  <van-swipe class="my-swipe" :autoplay="3000" indicator-color="white">
11    <van-swipe-item v-for="(item, index) in bannerList" :key="index">
12      
13    </van-swipe-item>
14  </van-swipe>
15
16  <!-- 导航 -->
17  <van-grid :border="false" column-num="5" icon-size="40" style="padding: 5px">
18    <van-grid-item v-for="(item, index) in navList" :key="index"
19      :icon="item.imageUrl" :text="item.text"
20      @click="$router.push('/category')" />
21  </van-grid>
22
23  <!-- 主会场 -->
24  <div class="main">
25    
26  </div>
27
28  <!-- 猜你喜欢 -->
29  <div class="guess">
30    <p class="guess-title">— 猜你喜欢 —</p>
31
32    <div class="goods-list">
33      <GoodsItem v-for="item in proList" :key="item.goods_id" :proItem="item"></GoodsItem>
34    </div>
35  </div>
36 </div>
37 </template>
38
39 <script>
40 import GoodsItem from '@components/GoodsItem.vue'
41 import { getHomeData } from '@api/home'
42 export default {
43   name: 'HomePage',
44   components: {
45     GoodsItem
46   },
47   data () {
48     return {
49       bannerList: [],
50       navList: [],
51       proList: []
52     }
53   },
54   async created () {
55     const { data: { pageData } } = await getHomeData()
56     this.bannerList = pageData.items[1].data
57     this.navList = pageData.items[4].data
58     this.proList = pageData.items[6].data
59   }
60 }
61 </script>

```

2.3.2 登陆界面

为保持本 APP 的总体风格一致，登陆界面以整洁大方为目标进行设计。本界面使用 css 样式对控件进行摆放，用 input 标签 进行输入框编辑，Button 进行按钮出发登录事件。 登陆界面如图 2-3 所示,注册界面如图 2-4 示。

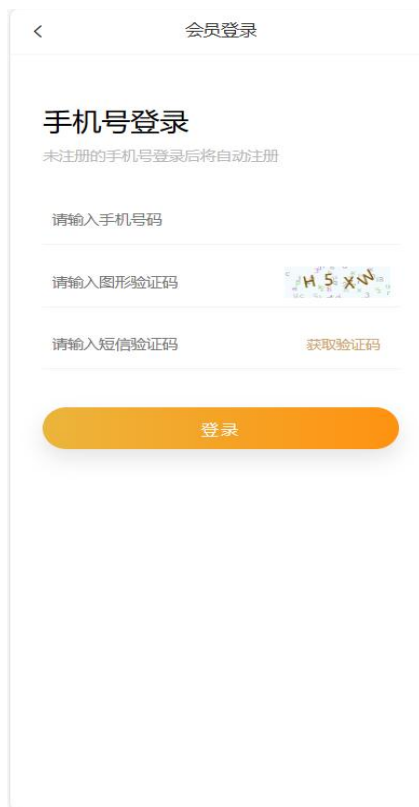
The image shows a mobile application login screen titled '会员登录' (Member Login). The main heading is '手机号登录' (Phone Number Login), with a subtext '未注册的手机号登录后将自动注册' (After logging in with an unregistered phone number, it will automatically register). There are three input fields: '请输入手机号码' (Please enter phone number), '请输入图形验证码' (Please enter graphical verification code) which includes a small image of a captcha with characters 'H, 5, X, W', and '请输入短信验证码' (Please enter SMS verification code) with a '获取验证码' (Get verification code) button next to it. At the bottom is a large orange '登录' (Login) button.

图 2-3 登陆界面

在登陆界面中最重要的中最重要的技术在于登陆账号。本 APP 使用了Vuex保存用户数据，通过发起API接口请求向用户发送短信验证码、验证图形验证码

登陆界面核心代码如下：

```

1 <script>
2 import { getPicCode, getMsgCode, login } from '@api/login'
3 // import { Toast } from 'vant'
4
5 export default {
6   name: 'LoginPage',
7   data () {
8     return {
9       picKey: '', // 图形验证码key
10      picUrl: '', // 图形验证码图片
11      totalSeconds: 60, // 倒计时的总秒数
12      second: 60, // 倒计时秒数
13      timer: null, // 定时器
14      mobile: '', // 手机号
15      picCode: '', // 用户输入的图形验证码
16      msgCode: '' // 短信验证码
17    }
18  },
19
20  async created () {
21    this.getPicCode()
22  },
23
24  destroyed () {
25    // 离开页面 清除定时器
26    clearInterval(this.timer)
27  },
28
29  methods: {
30    // 获取图形验证码
31    async getPicCode () {
32      const res = await getPicCode()
33      // const { data: { key, base64 } } = await request.get('/captcha/image')
34      this.picKey = res.data.key
35      this.picUrl = res.data.base64
36    },
37    // 校验手机号 和 图形验证码 是否合法
38    validFn () {
39      if (!/^1[3-9]\d{9}$/.test(this.mobile)) {
40        this.$toast('请输入正确的手机号')
41        return false
42      }
43      if (!/^w{4}$/.test(this.picCode)) {
44        this.$toast('请输入正确的图形验证码')
45        return false
46      }
47      return true
48    },
49    // 获取短信验证码
50    async getCode () {
51      // 校验
52      if (!this.validFn()) {
53        return
54      }
55      if (this.second === this.totalSeconds && !this.timer) {
56        await getMsgCode(this.picCode, this.picKey, this.mobile)
57        this.$toast('短信验证码发送成功(246810)')
58        // 开始倒计时
59        this.timer = setInterval(() => {
60          this.second--
61          if (this.second <= 0) {
62            // 重置倒计时
63            this.second = this.totalSeconds
64            clearInterval(this.timer)
65            this.timer = null
66          }
67        }, 1000)
68      }
69    },
70    // 登录
71    async login () {
72      if (!this.validFn()) {
73        return
74      }
75      if (!/^d{6}$/.test(this.msgCode)) {
76        this.$toast('请输入正确的短信验证码')
77        return
78      }
79      const res = await login(this.mobile, this.msgCode)
80      this.$store.commit('user/setUserInfo', res.data)
81      this.$toast('登录成功')
82
83      // 判断地址栏是否有回跳地址
84      const url = this.$route.query.backUrl || '/'
85      this.$router.replace(url)
86    }
87  }
88 }
89 </script>

```

2.3.3 商品分类界面



图 2-5 商品分类界面

在商品分类页面中最重要的技术是界面的相互跳转。商品分类界面的主要代码如下：



```
1 <script>
2 import { getCategoryData } from '@api/category'
3 export default {
4   name: 'CategoryPage',
5   created () {
6     this.getCategoryList()
7   },
8   data () {
9     return {
10       list: [],
11       activeIndex: 0
12     }
13   },
14   methods: {
15     async getCategoryList () {
16       const { data: { list } } = await getCategoryData()
17       this.list = list
18     }
19   }
20 }
21 </script>
```

2.3.4 购物车界面

当用户点击主界面的下方导航栏购物车图标时，即可从主界面跳转到购物车界面查看购物车、对购物车进行编辑、结算。在购物车界面中使用 css 样式布局 当用户未登录或购物车无商品是显示购物车为空，当购物车有商品是显示购物车列表。

购物车商品列表可以选中购物车和修改商品数量，并在下方显示总价，下方同时还提供全选和反选选项

购物车如图 2-6 所示所示。

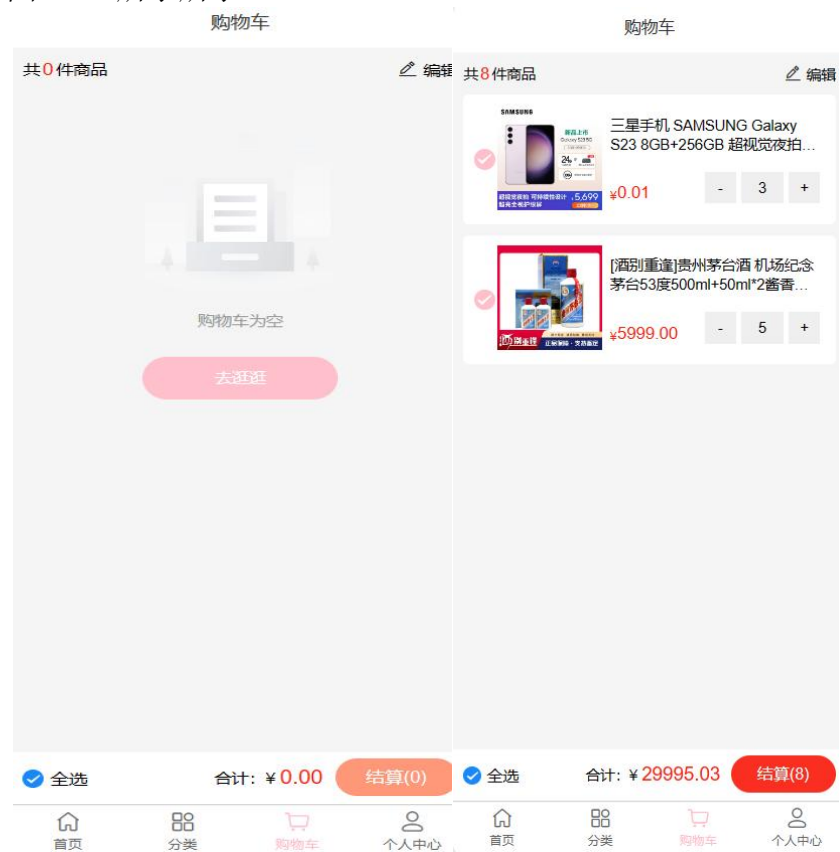


图 2-6 购物车界面图

在购物车界面中，必须实现商品列表的显示，商品的是否选择的功能，选中商品总价，全选与反选功能。点击结算按钮跳转到支付页面

购物车界面主要代码如下


```

1 <script>
2 import CountBox from '@/components/CountBox.vue'
3 import { mapState, mapGetters } from 'vuex'
4
5 export default {
6   name: 'CartPage',
7   components: { CountBox },
8   data () {
9     return {
10       show: true
11     }
12   },
13   computed: {
14     ...mapState('cart', ['cartList']),
15     ...mapGetters('cart', ['cartTotal', 'cartActive', 'cartActiveTotal', 'cartActivePrice', 'isAllChecked'])
16   },
17   created () {
18     // 必须是登录的用户才能获取购物车列表
19     if (this.$store.getters.token) {
20       this.$store.dispatch('cart/getCartAction')
21     }
22   },
23   methods: {
24     toggleCheck (goodsId) {
25       this.$store.commit('cart/toggleCheck', goodsId)
26     },
27     toggleAllCheck () {
28       this.$store.commit('cart/toggleAllCheck', !this.isAllChecked)
29     },
30     changeCart (goodsNum, goodsId, goodsSkuId) {
31       this.$store.dispatch('cart/changeCartAction', {
32         goodsNum,
33         goodsId,
34         goodsSkuId
35       })
36     },
37     handleDel () {
38       if (this.cartActiveTotal === 0) {
39         return
40       }
41       this.$store.dispatch('cart/delCartAction')
42       this.show = true
43     },
44     goPay () {
45       // 判断是否选中商品
46       if (this.cartActiveTotal > 0) {
47         this.$router.push({
48           path: '/pay',
49           query: {
50             mode: 'cart',
51             cartIds: this.cartActive.map(item => item.id).join(',')
52           }
53         })
54       }
55     }
56   },
57   watch: {
58     show (newValue) {
59       if (newValue === false) {
60         this.$store.commit('cart/toggleAllCheck', false)
61       } else {
62         this.$store.commit('cart/toggleAllCheck', true)
63       }
64     }
65   }
66 }
67 </script>

```

2.3.5 个人中心界面

当用户点击主界面的个人中心图标时，即可从主界面跳转到个人中心界面查看个人信息。当点击登录时即可跳转到登录页面 点击退出登录即可退出登录，页面个图标实现对应功能，

个人中心如图 2-8 所示所示。



图 2-8 英中学习界面

个人中心界面的主要代码如下：

```

1 <script>
2 import { getUserInfoDetail } from '@api/userInfo'
3 export default {
4   name: 'UserPage',
5   data () {
6     return {
7       detail: {}
8     }
9   },
10  created () {
11    if (this.isLogin) {
12      this.getUserInfoDetail()
13    }
14  },
15  computed: {
16    isLogin () {
17      return this.$store.getters.token
18    }
19  },
20  methods: {
21    async getUserInfoDetail () {
22      const { data: { userInfo } } = await getUserInfoDetail()
23      this.detail = userInfo
24    },
25    logout () {
26      if (this.$store.getters.token) {
27        this.$dialog.confirm({
28          title: '温馨提示',
29          message: '你确认要退出么? '
30        })
31        .then(() => {
32          this.$store.dispatch('user/logout')
33          this.$toast.success('退出成功')
34        })
35        .catch(() => {
36
37        })
38      }
39    }
40  }
41 }
42 </script>

```

2.3.6 商品详情界面

当点击主界面或商品列表的商品卡片时，即可从主界面跳转到商品详情界面查看商品详情。

商品详情包括有商品商品轮播图、商品标题、商品价格、商品评论、加入购物车和购买功能

点击加入购物车即可弹出选择框，选择添加数量，再次点击加入购物车即可将商品加入到购物车，可在购物车页面查看

点击立即购买同购物车再次点击即可跳转到支付页面

商品详情界面如图 2-9所示，



图 2-9 拼写学习界面

商品详情界面主要代码如下：

```

1 <script>
2 import { getProductDetail, getProductComment } from '@api/product'
3 import { addCart, getCartTotal } from '@api/cart'
4 import defaultImg from '@assets/default-avatar.png'
5 import CountBox from '@components/CountBox.vue'
6 import loginConfirm from '@mixins/loginConfirm'
7
8 export default {
9   name: 'ProDetail',
10  components: { CountBox },
11  mixins: [loginConfirm],
12  data () {
13    return {
14      addCount: 1,
15      mode: 'cart',
16      show: false,
17      activeNames: ['1'],
18      images: [],
19      current: 0,
20      detail: {},
21      comment: [],
22      total: 0,
23      defaultImg,
24      cartTotal: 0
25    }
26  },
27  created () {
28    this.getDetail()
29    this.getComments()
30    this.getCartTotal()
31  },
32  computed: {
33  },
34  methods: {
35    onChange (index) {
36      this.current = index
37    },
38    async getDetail () {
39      const { data: { detail } } = await getProductDetail(this.$route.params.id)
40      this.detail = detail
41      this.images = detail.goods_images
42    },
43    async getComments () {
44      const { data: { list, total } } = await getProductComment({
45        goodsId: this.$route.params.id,
46        limit: 5
47      })
48      this.comment = list
49      this.total = total
50    },
51    async getCartTotal () {
52      const { data: { cartTotal } } = await getCartTotal()
53      this.cartTotal = cartTotal
54    },
55    addFn () {
56      this.mode = 'cart'
57      this.show = true
58    },
59    buyFn () {
60      this.mode = 'buyNow'
61      this.show = true
62    },
63    async addCart () {
64      if (this.loginConfirm()) {
65        return
66      }
67      const { data } = await addCart({
68        goodsId: this.$route.params.id,
69        goodsNum: this.addCount,
70        goodsSkuId: this.detail.skuList[0].goods_sku_id
71      })
72      this.cartTotal = data.cartTotal
73      this.$toast.success('添加成功')
74      this.show = false
75    },
76    goBuyNow () {
77      if (this.loginConfirm()) {
78        return
79      }
80      this.$router.push({
81        path: '/pay',
82        query: {
83          mode: 'buyNow',
84          goodsId: this.$route.params.id,
85          goodsSkuId: this.detail.skuList[0].goods_sku_id,
86          goodsNum: this.addCount
87        }
88      })
89    }
90  }
91 }
92 }
93 </script>

```


2.3.7 搜索界面

当用户点击主界面的搜索框即可跳转到搜索界面，搜索界面布局由css样式实现，顶部包含一个搜索框，点击搜索框即可输入内容，点击搜索即可搜索商品，搜索记录会保存在搜索框下方，点击删除即可删除历史搜索记录

搜索界面如下图 2- 10所示

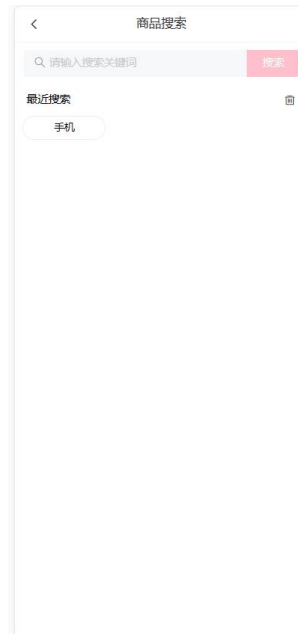


图 2-10 搜索界面

搜索界面主要代码如下

```
1 <script>
2 import { getHistory, setHistory } from '@/utils/storage'
3 export default {
4   name: 'SearchIndex',
5   data () {
6     return {
7       keyword: '',
8       history: getHistory()
9     }
10  },
11  methods: {
12    search (keyword) {
13      if (keyword.trim() !== '') {
14        this.history = this.history.filter(item => item !== keyword)
15        this.history.unshift(keyword)
16
17        setHistory(this.history)
18
19        this.$router.push(`/searchList?keyword=${keyword}`)
20      } else {
21        this.$toast.fail('请输入搜索关键词')
22      }
23    },
24    clear () {
25      this.history = []
26      setHistory(this.history)
27    }
28  }
29 }
30 </script>
31
```


2.3.8 商品列表界面

点击主界面推荐商品卡片，搜索页面搜索按钮，分类页面分类选项即可跳转到商品列表页面，列表页面展示商品的卡片，卡片包括商品图片、商品价格、商品销量、优惠等，点击商品卡片即可跳转到商品详情页，商品列表界面如图 2-11所示。



已学习界面和已学会界面主要代码如下：

```
1 <script>
2 import GoodsItem from '@components/GoodsItem.vue'
3 import { getProductList } from '@api/product'
4 export default {
5   name: 'SearchIndex',
6   components: {
7     GoodsItem
8   },
9   data () {
10    return {
11      product: [],
12      sortPrice: '0'
13    }
14  },
15  async created () {
16    const { data: { list } } = await getProductList({
17      categoryId: this.$route.query.categoryId,
18      goodsName: this.$route.query.keyword,
19      page: '1'
20    })
21    this.product = list.data
22  },
23  methods: {
24    async sort (sortType) {
25      this.sortPrice = this.sortPrice === '0' ? '1' : '0'
26      const { data: { list } } = await getProductList({
27        goodsName: this.$route.query.keyword,
28        page: '1',
29        sortPrice: this.sortPrice,
30        sortType: sortType
31      })
32      this.product = list.data
33    }
34  }
35 }
36 </script>
```

2.3.9 支付界面

当用户在商品详情页点击立即购买按钮或在购物车点击结算时，跳转掉支付页面，支付页面展示邮寄地址、需要购买的商品列表卡片、商品总计、商品优惠等，商品卡片包括商品图片、商品数量、商品单价等；点击提交订单即可支付；支付界面如图 2-12 所示。



图 2-12 支付界面

支付界面主要代码如下：

```

1 <script>
2 import { getAddressList } from '@api/address'
3 import { checkOrder, submitOrder } from '@api/order'
4 export default {
5   name: 'PayIndex',
6   data () {
7     return {
8       remark: '',
9       addressList: [],
10      orderInfo: {},
11      personal: {}
12    }
13  },
14  created () {
15    this.getAddressList()
16    this.checkOrder()
17  },
18  computed: {
19    setAddress () {
20      return this.addressList[0] || {}
21    },
22    mode () {
23      return this.$route.query.mode
24    },
25    cartIds () {
26      return this.$route.query.cartIds
27    },
28    goodsId () {
29      return this.$route.query.goodsId
30    },
31    goodsSkuId () {
32      return this.$route.query.goodsSkuId
33    },
34    goodsNum () {
35      return this.$route.query.goodsNum
36    }
37  },
38  methods: {
39    async getAddressList () {
40      const { data: { list } } = await getAddressList()
41      this.addressList = list
42    },
43    async checkOrder () {
44      if (this.mode === 'cart') {
45        const { data: { order, personal } } = await checkOrder(this.mode, {
46          cartIds: this.cartIds
47        })
48        this.orderInfo = order
49        this.personal = personal
50      }
51      if (this.mode === 'buyNow') {
52        const { data: { order, personal } } = await checkOrder(this.mode, {
53          goodsId: this.goodsId,
54          goodsSkuId: this.goodsSkuId,
55          goodsNum: this.goodsNum
56        })
57        this.orderInfo = order
58        this.personal = personal
59      }
60    },
61    // 提交订单
62    async submitOrder () {
63      if (this.mode === 'cart') {
64        await submitOrder(this.mode, {
65          remark: this.remark,
66          cartIds: this.cartIds
67        })
68      }
69      if (this.mode === 'buyNow') {
70        await submitOrder(this.mode, {
71          remark: this.remark,
72          goodsId: this.goodsId,
73          goodsSkuId: this.goodsSkuId,
74          goodsNum: this.goodsNum
75        })
76      }
77      this.$toast.success('支付成功')
78      this.$router.replace('/myOrder')
79    }
80  }
81 }
82 </script>

```

2.3.10 订单列表页面

当用户点击个人中心页面的我的订单时，即可从个人中心界面跳转到我的订单界面的操作。订单列表包括顶部标签栏，点击即可切换，下方展示所有订单卡片，订单列表界面



图 2-13 订单列表界面

订单列表界面主要代码如下：

```

1 <script>
2 import OrderListItem from '@components/OrderListItem.vue'
3 import { getMyOrderList } from '@api/order'
4
5 export default {
6   name: 'OrderPage',
7   components: {
8     OrderListItem
9   },
10  data () {
11    return {
12      active: 0,
13      page: 1,
14      list: []
15    }
16  },
17  methods: {
18    async getOrderList () {
19      const { data: { list } } = await getMyOrderList(this.active, this.page)
20      list.data.forEach((item) => {
21        item.total_num = 0
22        item.goods.forEach(goods => {
23          item.total_num += goods.total_num
24        })
25      })
26      this.list = list.data
27    }
28  },
29  watch: {
30    active: {
31      immediate: true,
32      handler () {
33        this.getOrderList()
34      }
35    }
36  }
37 }
38 </script>

```

3. 设计过程

3.1 工作进度安排

工作进度安排被分为五个部分。

(1) 项目选题开题阶段

时间：2023 年 1 月 2 日~2024 年 1 月 5 日

以所有课程设计题目为前提，整合自己所具有的所有专业知识然后确立选题为“基于Vue2框架京西商城 APP 的设计与实现”。

本 APP 使用 Win10 系统笔记本电脑的 vs code 开发工具进行开发设计，然后用安卓手机进行软件调试，以安卓 10.0 以上版本为主。

(2) 项目设计阶段

时间：2023 年 1 月 2 日~2024 年 1 月 5 日

本 APP 界面整洁大方，界面功能一目了然，操作简单。本 APP 用户可以不在意时间地点进行学习。帮助用户提高学习效率和时间利用率。

本 APP 实现了商城APP的基本功能，实现了购物车页面能够随时随地与加入购物车进行同步数据的功能。

(3) 项目实施阶段

时间：2023 年 1 月 2 日~2024 年 1 月 5 日

在这个时间段内，需要查找不同的资料再进行整理，配置好开发所需要的开发环境，然后对项目进行编译、测试。

(4) 作品及文档定稿阶段

时间：2023 年 1 月 2 日~2024 年 1 月 5 日

在这个时间段内，通过自己所编译的代码和指导老师的指导确定文档编写的格式，然后再对文档进行编写。

3.2 实施方法及手段

使用 vscode 软件，运用 JavaScript 来设计的一款基于 vue2框架 的京西商城 APP。在设计实现的过程中，主要运用到了以下的几种研究方法。

(1) 文献研究法，通过上网查找不同的资料，并对资料进行整理。

(2) 实践研究法，通过不断地对安卓手机进行调试，以最直观调试 APP 的方式来

解决问题。通过对比其他类似的商城 APP 进行实践研究，与自己整体思路进行对比，熟悉其他的 APP 来研发自己的商城 APP。

(3) 专家咨询法，通过向 Android 相关的专业老师、同学请教，解决自己在项目开发中遇到的瓶颈及困难，不断地提高自己。

3.3 遇到的问题及解决方案

无。

4. 作品及特点

4.1 作品展示

本 APP 是在 Android Studio 平台下开发的一款基于 Android 手机操作平台的英语单词学习 APP。作品中各页面如图所示。



图 4-1 首页界面图



图 4-2 商品分类界面图



图 4-3 购物车界面图



图 4-4 登录界面图



图 4-5 支付界面图



图 4-6 商品详情界面图



图 4-7 订单列表界面图

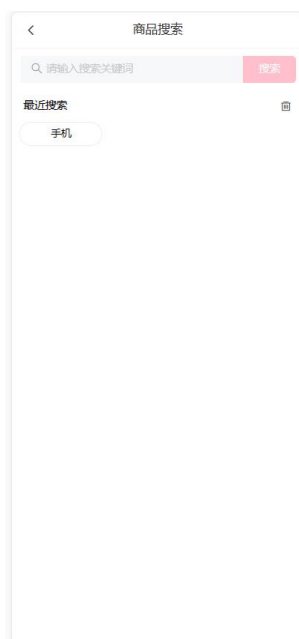


图 4-8 搜索界面图



图 4-9 商品列表界面图



图 4-10 个人中心界面图

4.2 作品特点

1. 本 APP 用户可通过个人中心界面的退出登陆按钮进行退出登陆的操作。
2. 本 APP 在主界面具有广告轮播的效果。

参考资料

- [1] 黄雪莉. 脑王记单词 App 的设计与实现[D]. 华中科技大学, 2016.
- [2] 骆诗奇. 基于 android 的英文学习系统的设计与实现[D]. 电子科技大学, 2014.
- [3] 韩秋玲. 基于 Android 的英语学习系统设计与实现[D]. 电子科技大学, 2012.
- [4] Android 的相关知识 <https://baike.baidu.com/item/Android/60243>

致谢

时间过的很快，俩年半的大专生涯也面临着结束，在这段大专生活学习中，我由衷的感谢家人们的支持、老师们的淳淳教诲和同学们的乐于助人，才让我在这个俩年半里得到成长。我想这一定是我这辈子都不会忘记的一段记忆。

首先，非常感谢刘湘黔老师、周文君老师在我准备毕设的这一整个阶段过程中给我的耐心教导。在他们的不断指导下，我的毕设才得以完成。他们不求回报的精神深深感动到了我，所以在这个毕设从开始到结束我不仅学习到了专业知识，在思想上也得到了很大的提升。

其次，我也非常感谢我的同学，他们在我搞毕设这个阶段也给了不少的意见，如果说老师是我的毕设路上的指引绳，那同学便是我毕设路上的孔明灯，照亮我的前方之路。

最后，我还需要感谢湖南安全技术职业学院为我们提供的美好学习氛围和环境资源。在此我由衷的向各老师同学表达至高的谢意。