



成绩

(采用四级记分制)

西北大学

本科毕业论文（设计）

题目：宠物医院平台的设计与实现

学生姓名 田震

学 号 2018117136

指导教师 董乐红

院 系 信息科学与技术学院

专 业 计算机科学与技术

年 级 2018 级

教务处制

诚信声明

本人郑重声明：本人所呈交的毕业论文（设计），是在导师的指导下独立进行研究所取得的成果。毕业论文（设计）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或在网上发表的论文。

特此声明。

论文作者签名：

日 期： 年 月 日

摘要

宠物医院平台的设计与实现

近年来，随着人们物质生活水平的提高，人们对于自身的精神生活也越来越重视。但是大部分人群，比如上班族、孤寡老人、留守儿童等缺乏亲人爱人的陪伴，转而将这种需求寄托到宠物身上。所以宠物数量增多，宠物医疗需求也随之增长。在信息时代，为了更有效便捷地获取宠物医疗服务，一款方便高效的宠物医院平台成为了人们的不二选择。

本文的主要内容有：（1）介绍这款宠物医院平台的开发背景、目的和研究意义。（2）介绍开发此平台用到的技术和知识。使用微信开发者工具开发一款微信小程序。（3）对宠物医院平台进行需求分析。（4）根据需求分析得到的功能模块给出设计思路、实现方法和测试结果。（5）总结整个项目的开发流程。

关键词：宠物医院；微信小程序；宠物商城

Abstract

Design and implementation of pet hospital platform

In recent years, with the improvement of people's material living standards, people pay more and more attention to their own spiritual life. However, most people, such as office workers, the elderly and left behind children, lack the company of relatives and lovers, and instead place this demand on pets. Therefore, as the number of pets increases, the demand for pet medical treatment also increases. In the information age, in order to obtain pet medical services more effectively and conveniently, a convenient and efficient pet hospital platform has become people's best choice.

The main contents of this paper are as follows: (1) introduce the development background, purpose and research significance of this pet hospital platform. (2) Introduce the technology and knowledge used to develop this platform. Use WeChat developer tools to develop a WeChat applet. (3) Analyze the needs of pet hospital platform. (4) According to the functional modules obtained from the demand analysis, the design idea, implementation method and test results are given. (5) Summarize the development process of the whole project.

keywords: Pet hospital; WeChat applet; Pet mall

目 录

| | |
|-----------------------------|----|
| 1 绪论 | 1 |
| 1.1 宠物医院平台发展背景 | 1 |
| 1.1.1 移动智能设备与软件发展现状分析 | 1 |
| 1.1.2 线下宠物医院发展现状分析 | 1 |
| 1.2 宠物医院平台发展现状分析 | 2 |
| 1.3 主要研究内容与目的 | 2 |
| 1.3.1 主要研究内容 | 2 |
| 1.3.2 主要研究目的 | 3 |
| 2 开发技术介绍 | 4 |
| 2.1 微信小程序介绍 | 4 |
| 2.2 开发工具及语言介绍 | 4 |
| 3 需求分析 | 5 |
| 3.1 基本业务流程介绍 | 5 |
| 3.2 功能性需求分析 | 5 |
| 3.2.1 购物功能需求分析 | 5 |
| 3.2.2 预约功能需求分析 | 6 |
| 3.2.3 急救功能需求分析 | 6 |
| 3.2.4 知识功能需求分析 | 6 |
| 3.3 非功能性需求分析 | 6 |
| 4 总体设计 | 7 |
| 4.1 总体工作流程设计 | 7 |
| 4.2 小程序功能设计 | 7 |
| 4.2.1 购物功能设计 | 7 |
| 4.2.2 预约功能设计 | 7 |
| 4.2.3 急救功能设计 | 7 |
| 4.2.4 知识功能设计 | 7 |
| 4.3 数据库设计 | 8 |
| 5 系统实现 | 10 |

| | |
|----------------------|-----------|
| 5.1 非功能性页面的实现..... | 10 |
| 5.1.1 用户注册页面的实现..... | 10 |
| 5.1.2 首页页面的实现..... | 11 |
| 5.1.3 个人中心页面的实现..... | 11 |
| 5.2 购物功能的实现..... | 12 |
| 5.2.1 宠物商城模块的实现..... | 12 |
| 5.2.2 购物车模块的实现..... | 14 |
| 5.2.3 订单模块的实现..... | 16 |
| 5.2.4 上下架模块的实现..... | 17 |
| 5.3 预约功能的实现..... | 19 |
| 5.3.1 线上预约模块的实现..... | 19 |
| 5.3.2 处理预约模块的实现..... | 21 |
| 5.4 急救功能的实现..... | 21 |
| 5.4.1 宠物急救模块的实现..... | 21 |
| 5.4.2 处理急救模块的实现..... | 23 |
| 5.5 知识功能的实现..... | 23 |
| 6 系统测试..... | 25 |
| 6.1 功能测试..... | 25 |
| 6.2 其他测试..... | 26 |
| 7 总结与展望..... | 27 |
| 7.1 总结..... | 27 |
| 7.2 展望..... | 27 |
| 参考文献..... | 28 |

1 绪论

1.1 宠物医院平台发展背景

本节主要从软硬件设备发展现状和宠物医院发展现状两方面阐述宠物医院平台的发展背景及可行性。

1.1.1 移动智能设备与软件发展现状分析

随着移动智能设备（手机、pad 等）的发展和兴起，人们在工作生活中处理事务的场所逐渐从线下转移到线上。因此，涉及各个领域的应用（app、小程序、公众号等）也应运而生。通过移动智能设备 and 应用便能极大的简化线下繁琐的办事流程，节省办事时间，提高办事效率。

根据 2022 年 2 月 25 日，中国互联网络信息中心（CNNIC）在京发布第 49 次《中国互联网络发展状况统计报告》^[1]（以下简称：《报告》）。《报告》显示，截至 2021 年 12 月，我国网民规模达 10.32 亿，较 2020 年 12 月增长 4296 万，互联网普及率达 73.0%。《报告》显示，2021 年我国互联网应用用户规模保持平稳增长。一是即时通信等应用基本实现普及。截至 2021 年 12 月，在网民中，即时通信、网络视频、短视频用户使用率分别为 97.5%、94.5%和 90.5%，用户规模分别达 10.07 亿、9.75 亿和 9.34 亿。二是在线办公、在线医疗等应用保持较快增长。截至 2021 年 12 月，在线办公、在线医疗用户规模分别达 4.69 亿和 2.98 亿，同比分别增长 35.7%和 38.7%，成为用户规模增长最快的两类应用。由于疫情影响，这两类应用迅速崛起，快速发展的同时也给人们的生活带来了很大的便利，因此，越来越多的行业都投入开发自己的办公应用。

1.1.2 线下宠物医院发展现状分析

根据《2019 中国宠物医院发展报告》^[2]，截止 2019 年 4 月，中国宠物医院数量超过 15000 家，其中非连锁医院占比近 90%；广东省宠物医院数量最多，达到 1200 家左右；江苏、四川、浙江、山东省宠物院都超过千家。根据宠物家“全国宠物医院（除港澳台）”相关数据，2020 年 2 月，全国宠物医院总数达到 2.28 万家。根据百度地图最新数据显示，2021 年 2 月，全国宠物医院总数增长至 3.08 万家。宠物医院总数年年上升，背后代表着宠物医疗市场的巨大利润，并且暗示着宠物医疗市场并未达到饱和状态。在这个时期，哪家医院的服务能带给用户更好的体验，便能在市场上表现出更强大的竞争力。

1.2 宠物医院平台发展现状分析

目前国内外宠物医院平台数量极多，粗略估算已超过 50 个。国内主流宠物平台主要有：波奇宠物、它嗅宠物、E 宠、爱它乐、瑞派宠物医院、仲信宠物医院等；国外主流宠物平台主要有 pet coach、tagg、pet phone、PetFirstAid 等。其中国内宠物行业内发展较好的要属波奇宠物，2020 年 9 月 30 日，波奇宠物以“BQ”为股票代码在纽约证券交易所挂牌，成为中国首个上市的宠物综合服务平台。波奇宠物在 2014 年便发布了波奇宠物 APP，足以看出他们的毒辣的商业眼光。

上述这些平台虽然琳琅满目，但是使用起来功能上却是大同小异的，他们大多都有以下特点：1. 界面设计简洁，给人很舒服的使用环境；2. 提供兽医预约、咨询等服务；3. 提供宠物药品、食品、用品线上购买服务；4. 提供专业养宠知识等。当然，也有许多宠物平台具有自己的个性化功能，比如：1. 波奇宠物 app 用户可以发送自己的养宠经验贴，也可以晒自己的萌宠；2. 它嗅宠物 app 提供 7*24 小时医生在线问诊服务，回复速度快、专业度高；3. 爱它乐 app 提供用品评测功能，让宠物主人不花冤枉钱，找到更优质的商品；4. 仲信宠物医院小程序提供预约挂号服务，节省用户排队的时间。

综上所述，一个具有竞争力的宠物医院平台需要具有以下功能：①提供宠物医生在线问诊功能；②提供宠物用品线上购买服务；③提供预约洗美、疫苗、挂号等功能；④提供专业养宠知识。

1.3 主要研究内容与目的

1.3.1 主要研究内容

经过对各大宠物医院平台的分析与研究，结合本课题自身的需求，本课题设计将宠物医院平台分为两大板块：宠物主人板块和工作人员板块。

宠物主人板块包含宠物商城模块、购物车模块、预约模块、宠物急救模块、宠物知识模块、我的订单模块、修改地址模块、查看预约模块。

工作人员板块包含上架商品模块、下架商品模块、处理预约模块、处理急救模块、订单模块。

通过对设置不同用户权限将用户分为宠物主人和工作人员两类，两类用户进入平台会看到不同的页面，通过数据库传输实现两类用户之间的交互。

1.3.2 主要研究目的

主要研究目的是实现一个基于微信小程序的宠物医院平台，此平台可以给宠物主人提供宠物急救常识，提高宠物救治率；提供预约服务，到店既服务，节约用户时间；提供宠物上门急救服务，保障宠物安全；提供宠物商城服务，方便用户购买宠物食品及药品。也可以帮助宠物医院工作人员完成工作，提高他们的工作效率。

2 开发技术介绍

2.1 微信小程序介绍

微信小程序^[4]是一种小程序。这是一种无需下载即可使用的应用程序。这也是一项创新技术。经过近两年的发展，新的微信小程序开发环境和开发者生态已经构建。小程序适用于使用时长短、使用频次低的场景。小程序相比传统 app 具有以下优点：不用安装，即用即走，节省空间流量；开发及维护成本低；推广更容易。

2.2 开发工具及语言介绍

开发工具：微信开发者工具 1.05.2203251

开发方式：云开发。微信云开发是微信团队联合腾讯云推出的专业的小程序开发服务。开发者可以使用云开发快速开发小程序、小游戏、公众号网页等，并且原生打通微信开放能力。开发者无需搭建服务器，可免鉴权直接使用平台提供的 API 进行业务开发。如图 2.1 所示，云开发提供以下功能。



图 2.1 云开发

开发语言：JavaScript^[3]、WXML^[6]、WXSS^[5]

3 需求分析

3.1 基本业务流程介绍

宠物主人：①打开小程序可以点击“宠物商城”，选择想要购买的商品将其加入购物车，可以点击下方按钮直接进入购物车，随后可以增减商品数量或者删除商品，点击“结算”来提交订单，完成付款后即可在我的订单中查看到订单信息。②点击“线上预约”进入预约页面，选择预约的项目及时间，填写预约信息，点击“提交”，随后即可在个人中心->我的预约页面查看预约信息。③点击“养宠知识”浏览养宠知识页面。④点击“宠物急救”，填写急救信息并拨打急救电话。

工作人员：①打开小程序可以点击“处理预约”，即可查看并处理用户的预约信息。②点击“处理急救”，即可处理急救信息。③点击“上架商品”，录入商品信息及图片，即可完成上架。④点击“下架商品”，选择要下架的商品点击“下架”即可完成下架。⑤点击“个人中心”->“全部”即可处理用户下达的商品订单。

3.2 功能性需求分析

3.2.1 购物功能需求分析

购物功能包含了宠物商城模块、购物车模块、修改地址模块、订单模块、上下架模块。

宠物商城模块需要展示数据库中所有商品，提供“加入购物车”按钮，点击后转变为调整数量键，数量最低不小于 1，最大不超过 99。选择好之后可以点击最下方的“购物车”按钮进入购物车模块。

购物车模块展示用户加入购物车中的商品，并计算总价。商品左侧提供选择框，用户可自己选择结算哪几种商品。购物车模块仍提供调整商品数量键，最大不超过 99。但当数量小于 1 时，提示用户是否删除该商品，根据用户反馈决定保留还是删除该商品。右下角“结算”按钮点击后生成订单，若支付成功订单状态记为“待发货”；若支付失败，订单状态记为“待付款”。随后清除购物车中已经形成订单的商品。随后可以在订单模块查看订单。

订单模块根据订单状态分类展示订单，也可以展示全部订单。工作人员可在订单模块处理“待发货”的订单，录入快递单号并提交，则该订单状态被改

为“待收货”。宠物主人在待付款订单页面对订单完成付款后，该订单状态被改为“待发货”，在待收货订单页面可以点击“确认收货”，则该订单状态改为“已完成”。

上下架模块中，工作人员在上架页面录入商品信息，后台将其写入数据库，以供宠物商城页面查询。在下架页面选择要下架的商品，点击“下架”，则从数据库中删除该商品信息。

3.2.2 预约功能需求分析

预约功能包含线上预约模块和处理预约模块。

宠物主人可以通过线上预约模块，提交预约项目、时间、预约人等信息，后台将其录入数据库，并标记状态为“未完成”。

工作人员可以通过处理预约模块，查看客户的预约信息，可以点击“已完成”按钮，将预约状态改为“已完成”。

3.2.3 急救功能需求分析

急救功能包含宠物急救模块和处理急救模块。

宠物主人可以通过宠物急救模块，提交急救地址、宠物症状、联系人等信息，后台将其录入数据库，并标记状态为“未出发”。提交后提醒宠物主人拨打急救电话确认急救信息。

工作人员可以通过处理急救模块，查看客户的急救信息，可以点击“已派遣医生出诊”按钮，将急救状态改为“已出发”。

3.2.4 知识功能需求分析

知识功能只涉及养宠知识模块，该模块需要先展示各种知识的标题，用户可以点击标题或标题右侧的下拉符号以展开下拉框，在下拉框中展示具体的养宠知识。

3.3 非功能性需求分析

(1) 性能要求：要求小程序具有流畅的使用体验，保证在用户使用时没用明显的卡顿和延迟。

(2) 兼容性：要求小程序能在目前市面上大部分手机微信上流畅运行。

(3) 界面设计：要求小程序页面设计简单优美，提供给用户良好的使用观感。

4 总体设计

4.1 总体工作流程设计

用户进入小程序后可选择注册成为宠物主人或工作人员中的一个，选择成为宠物主人可直接注册成功，选择成为工作人员则需求输入企业内部的验证码，验证码输入正确才可以注册成功。宠物主人注册成功后可开始购物、预约、急救、学习知识等。工作人员注册成功后即可开始处理用户订单、预约、急救等。

4.2 小程序功能设计

4.2.1 购物功能设计

宠物商城^[8]页面通过 WXML 中 view 的属性 wx:for 逐条展示从数据库查到的商品，每条展示框左侧展示商品图片，右侧展示商品名称、现价和原价。通过 wx:if 实现“加入购物车”按钮点击后变为调整数量按钮。

购物车^[9]页面通过 check-box 容器实现选择想要结算的商品。同样每条展示框左侧展示商品图片，右侧展示商品名称、现价、原价和数量调整按钮。

修改地址页面通过 picker 容器的 bindchange="bindRegionChange" 属性实现上滑菜单选择地址功能。

订单页面通过页面间传值实现同一页面显示不同状态下的订单。

上架商品页面通过 wx.chooseImage^[7] 实现图片上传，并通过 wx.cloud.uploadFile^[7] 将图片上传至云存储中。

4.2.2 预约功能设计

线上预约页面通过 picker 容器的 bindchange 属性实现上滑菜单选择预约项目、日期和时间。使用 form 容器提交其他预约信息。

处理预约页面展示所有未完成的预约项目，提供修改状态的 button。

4.2.3 急救功能设计

宠物急救页面通过 form 容器和 input 组件提交所有急救信息，并使用 wx.makePhoneCall 唤起手机拨号功能。

处理急救页面展示所有未出发的急救信息，提供修改状态的 button。

4.2.4 知识功能设计

养宠知识页面通过 wx:for 展示 list 里所有养宠知识，通过 wx:if 决定下拉框是否展开。

4.3 数据库设计

所有数据库操作均通过 `wx.cloud.database().collection()`^[7]及其下级属性组合实现。具体数据库设计如表 4.1-4.6 所示。

表 4.1 用户表(total-user)

| 属性 | 数据类型 | 含义 |
|---------------|---------|-------------|
| _id | string | 系统生成的 id |
| _openid | string | 小程序用户独有的 id |
| userInfo | object | 用户的公开信息 |
| userRank | boolean | 用户类型 |
| recipientInfo | object | 收件人信息 |

表 4.2 商品表(goods)

| 属性 | 数据类型 | 含义 |
|----------------|--------|-------------|
| _id | string | 系统生成的 id |
| _openid | string | 小程序用户独有的 id |
| imageUrl | string | 商品图片地址 |
| inShoppingCart | number | 是否在购物车里 |
| originalPrice | number | 原价 |
| price | number | 现价 |
| productName | string | 商品名称 |
| productQty | number | 商品数量 |

表 4.3 购物车表(cart)

| 属性 | | 数据类型 | | 含义 |
|---------|-------------|--------|--------|-------------------|
| _id | | string | | 系统生成的 id |
| _openid | | string | | 小程序用户独有的 id |
| goods | product_id | array | string | 位于购物车中的商品 id 及其数量 |
| | product qty | | number | |

表 4.4 订单表(order)

| 属性 | 数据类型 | 含义 |
|----------------|--------|-------------|
| _id | string | 系统生成的 id |
| _openid | string | 小程序用户独有的 id |
| selectId | array | 订单中的各商品 id |
| selectQty | array | 订单中的各商品数量 |
| state | string | 订单状态 |
| trackingNumber | string | 快递单号 |
| value | object | 用于程序端存储商品信息 |
| recipientInfo | object | 收货人信息 |

表 4.5 急救信息表 (aid)

| 属性 | 数据类型 | 含义 |
|---------|--------|-------------|
| _id | string | 系统生成的 id |
| _openid | string | 小程序用户独有的 id |
| addr | string | 急救地址 |
| name | string | 用户姓名 |
| spec | string | 宠物种类 |
| tel | string | 用户电话 |
| sym | string | 宠物症状 |
| state | string | 急救信息状态 |

表 4.6 预约信息表 (appoint)

| 属性 | 数据类型 | 含义 |
|---------|--------|-------------|
| _id | string | 系统生成的 id |
| _openid | string | 小程序用户独有的 id |
| date | string | 预约日期 |
| name | string | 用户姓名 |
| time | string | 预约具体时间 |
| tel | string | 用户电话 |
| project | string | 预约项目 |
| state | string | 预约状态 |

5 系统实现

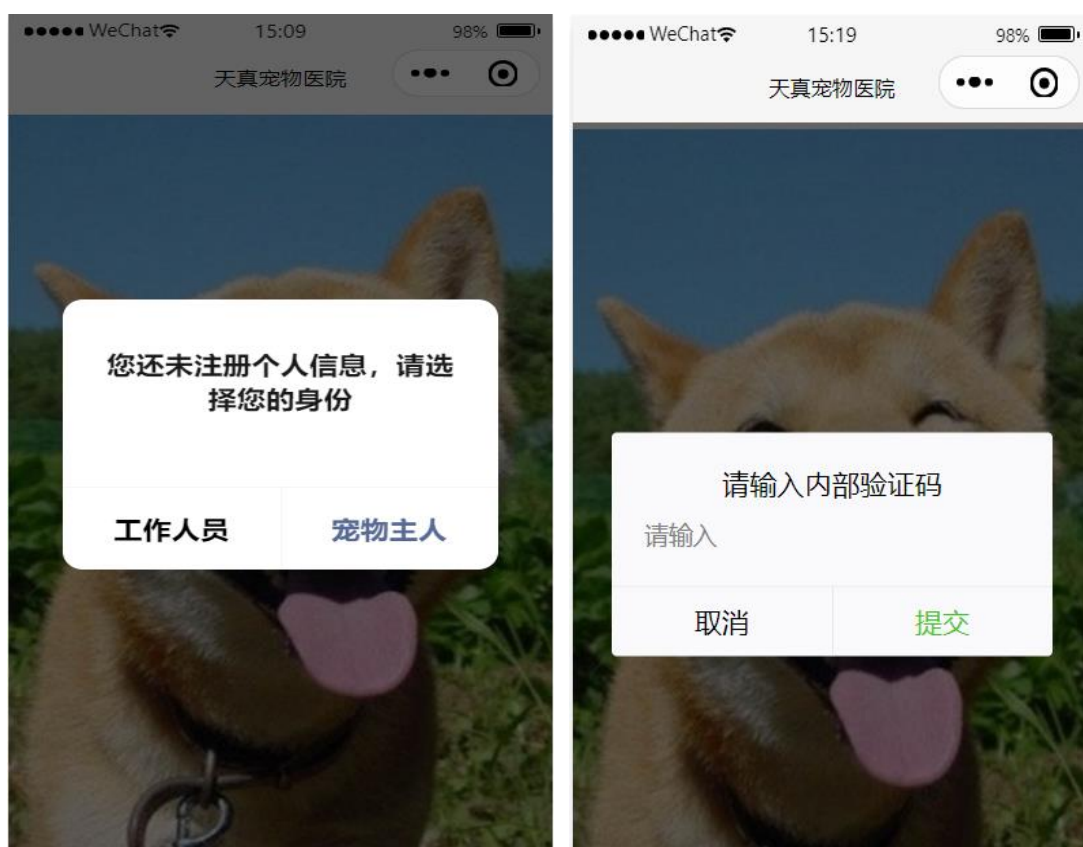
每个页面的实现都需要四个文件的支持：文件名.js(页面逻辑文件)、文件名.wxml(页面的结构文件)、文件名.wxss(页面样式表文件)、文件名.json(页面配置文件)。

5.1 非功能性页面的实现

这些非功能性页面，主要作用是引导用户进行下一步操作，提供内容明确、图标优美的按钮以供用户选择。

5.1.1 用户注册页面的实现

查询用户表(total-user)，若表中存在该用户信息，则根据已知信息直接进入对应的小程序首页。若表中没有该用户信息，如图 5.1(a)所示，使用 wx.showModal 弹出提示框让用户选择身份完成注册。若用户点击“宠物主人”则将用户类型录入用户表，并直接进入宠物主人首页。若用户点击“工作人员”，如图 5.1(b)所示，使用 modal 容器配合 input 组件，要求用户输入正确的内部验证码，才能注册成功，然后将用户类型录入用户表，并进入工作人员首页。



(a)

(b)

图 5.1 注册页面

5.1.2 首页页面的实现

如图 5.2(a)所示，宠物主人首页上方使用 swiper 和 image 组件实现了图片滑动展示，后续可在此处张贴广告。页面下方依次提供各大功能的入口按钮，并提供进入购物车和个人中心的导航栏。宠物医生在线咨询功能调用微信小程序 button^[7]的开放能力“打开客服会话”，在后台添加宠物医生的微信即可实现宠物主人与医生之间的对话。

如图 5.2(b)所示，工作人员首页则简单明了，只提供所需功能的入口按钮及进入个人中心的导航栏。

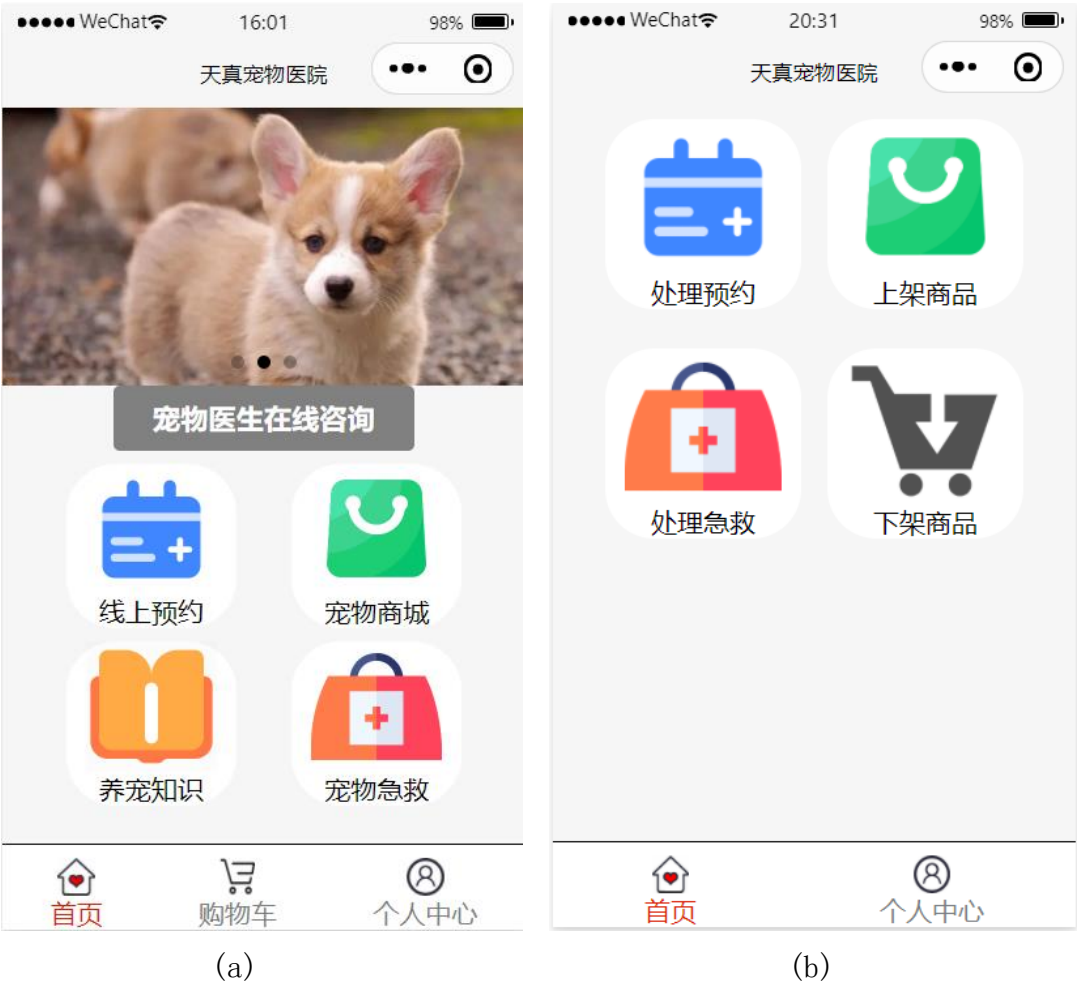


图 5.2 首页页面

5.1.3 个人中心页面的实现

如图 5.3 所示，宠物主人和工作人员的个人中心均提供各类订单的查询按钮。区别是宠物主人的个人中心页面还可以修改收货地址、查询自己的预约信息。所有用户首次进入个人中心页面可以选择授权获取微信头像与昵称，使用

wx.setStorage 将该信息存入本地缓存，则在本地缓存被清理之前，用户再次进入小程序个人中心页面，都不需要再次授权。



图 5.3 个人中心页面

5.2 购物功能的实现

5.2.1 宠物商城模块的实现

宠物商城页面实现^[10]效果如图 5.4 所示，主要功能有：通过 wx:for 展示所有已上架的商品；通过 wx:if 和 wx:else 实现点击“加入购物车”后按钮的切换；点击最下方按钮可以跳转至购物车页面。

本页面逻辑文件为 market.js，主要负责从数据库中查询商品信息，将结果赋值给数组 productList 以供结构文件使用。js 文件还负责用户点击“加入购物车”、“+”、“-”按钮后数据库中购物车表的更新。本页面结构文件为 market.wxml，核心代码如下：

```
<view class="list1" wx:for="{{productList}}" wx:key="produceId">

    <view><image class="image1" src="{{item.imageUrl}}"></image></view>
```

```

<view class="detail1">

  <text>{{item.productName}}</text>

  <text class="price1">现价: {{item.price}}</text>

  <text class="oriPrice1">原价: {{item.originalPrice}}</text>

  <view class="qtyinfo1" wx:if="{{item.inShoppingCart==0}}">

    <button class="button1" data-productid="{{item._id}}" data-index="{{index}}"
    bindtap="addToShoppingCart">加入购物车</button>

  </view>

  <view class="qtyinfo1" wx:else>

    <text class="minus1" data-productid="{{item._id}}" data-index="{{index}}" bindtap="tapMinus">-
  </text>

    <input class="qty1" type="text" data-_productid="{{item._id}}" data-index="{{index}}"
    value="{{item.productQty}}"/>

    <text class="plus1" data-productid="{{item._id}}" data-index="{{index}}" bindtap="tapPlus">+</text>

  </view>

</view>

</view>

<view class="liubai"></view>

<view class="cart" bindtap="toCart">

  <image class="cartimage" src="/images/cart.png"></image>

  <text class="carttext">购物车</text>

</view>

```



图 5.4 宠物商城页面

5.2.2 购物车模块的实现

购物车页面实现效果如图 5.5 所示，在购物车中可以查看已添加的商品，可以增减任意商品的购买数量，数量减为 0 时提醒用户是否删除该商品，每次进入购物车默认勾选所有购物车中的商品，支持用户任意勾选想要结算的商品。当用户调整所选商品和数量时，页面可以立刻更新当前选中商品的总价格。用户选好商品后，点击“结算”按钮，若未填写收货地址，则提示用户跳转填写；若已填写收货地址，则生成订单并进入支付环节，支付成功则订单状态设置为“待发货”，支付失败则将订单状态设置为“待付款”。

本页面逻辑文件为 cart.js，即时计算总价格的核心代码如下：

```
countAmount:function(ids){

    let arr1 = this.data.productList;

    if(arr1.length<1){return false;}

    var _this=this;

    this.setData({
```

```

        selectedProductQty: []
    })

    for(var i in ids){

        for(var j in _this.data.productList){

            if(ids[i] == _this.data.productList[j]._id){

                var p = _this.data.selectedProductQty;

                p.push(_this.data.productList[j].productQty)

                _this.setData({

                    selectedProductQty: p

                })

            }

        }

    }

    let amt=0.00;

    let selectIds="";

    for(let i=0;i<arr1.length;i++) {

        if (ids=="0" || ids.indexOf(arr1[i]._id.toString())>=0)

        {

            amt += arr1[i].price*arr1[i].productQty;

        }

        selectIds += ","+arr1[i]._id;

    }

    selectIds = selectIds.substring(1);

    if(ids!="0"){

        selectIds = ids;

    }

    this.setData({amount:amt.toFixed(2)});

    this.setData({selectedProductId:selectIds});

}

```

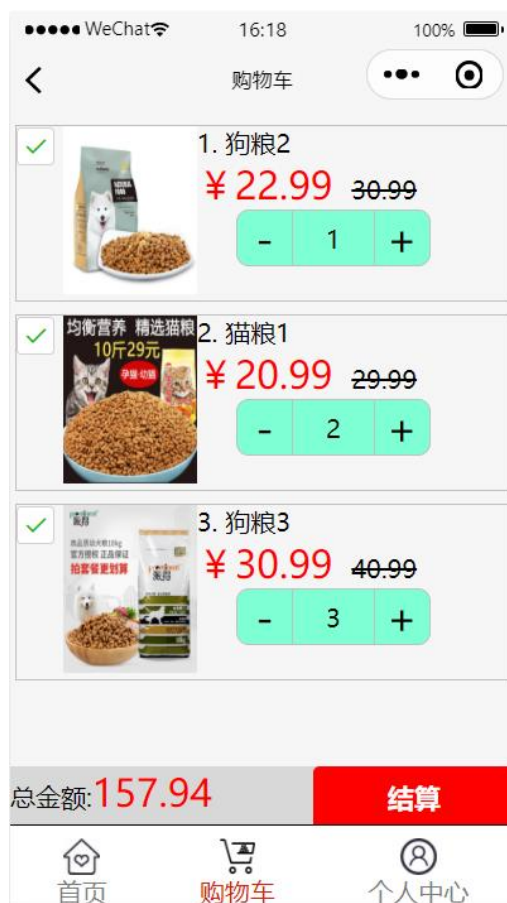


图 5.5 购物车页面

5.2.3 订单模块的实现

订单模块的实现效果如图 5.6 所示，订单模块使用页面间传值实现了同一页面的不同显示。在个人中心页面跳转至订单页面时，根据用户点击的按钮传入不同的值。具体实现代码如下：

meC.js:

```
bindOrder: function (options) {

    let orderState = options.currentTarget.dataset.state;

    wx.navigateTo({

        url: '/pages/meC/orderList/orderList?orderState='+orderState,

    })

},
```

orderList.js:

```
onLoad: function (options) {
```

```

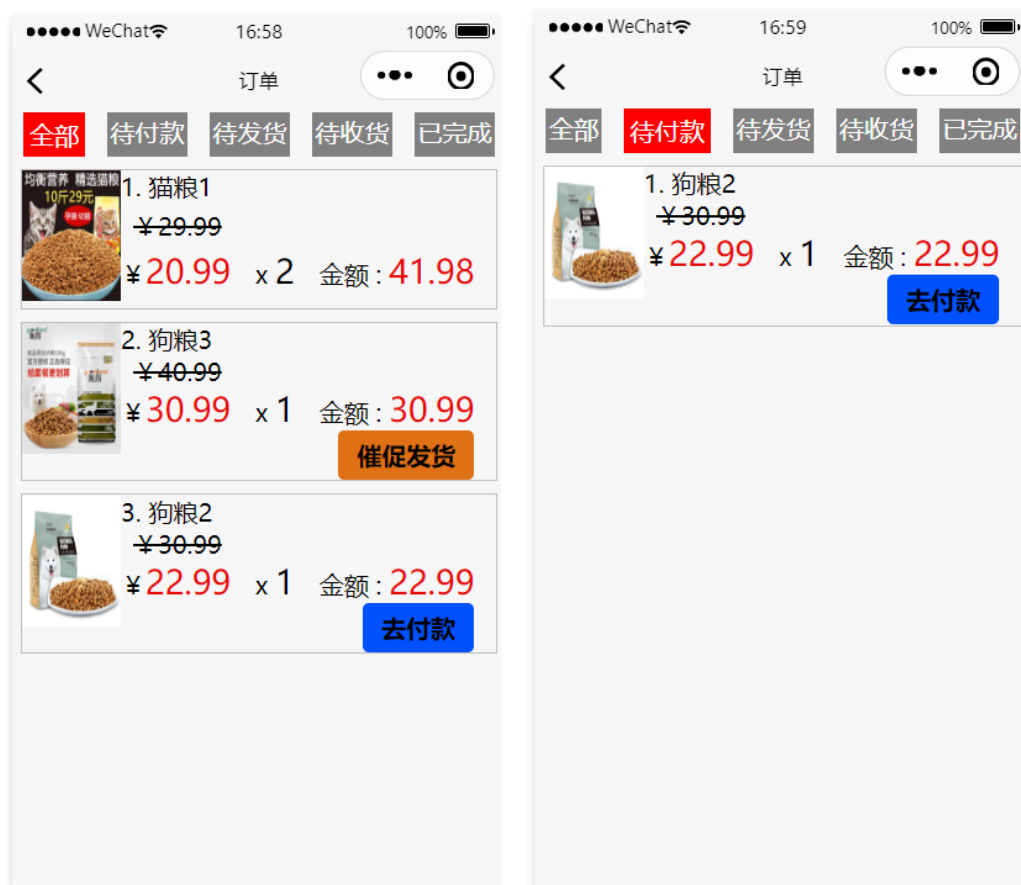
this.setData({stateNo:options.orderState});

this.initProductList();

},

```

进入订单页面后，在 onLoad 函数中获取传入的 orderState 值，根据该值渲染页面，就能得到不同的页面显示。



(a)

(b)

图 5.6 订单页面

5.2.4 上下架模块的实现

上架页面的实现效果如图 5.7 (a) 所示，录入商品信息并上传商品图片点击“提交”，js 文件将商品信息写入数据库商品表中，即可完成上架。本页面逻辑文件为 upgoods.js，上传图片的核心代码如下：

```

bindUpload: function (e) {

    var that = this

    wx.chooseImage({

```

```

count: 1, // 默认 1

sizeType: ["original", "compressed"], // 可以指定是原图还是压缩图，默认二者都有

sourceType: ["album", "camera"], // 可以指定来源是相册还是相机，默认二者都有

success: function (res) {

    // 返回选定照片的本地文件路径列表，tempFilePath 可以作为 img 标签的 src 属性显示图片

    var tempFilePaths = res.tempFilePaths

    that.data.imgs.push(tempFilePaths[0])

    that.setData({

        imgs: that.data.imgs

    })

    console.log(that.data.imgs)

}

})

}

```



(a)



(b)

图 5.7 上下架页面

下架页面的实现效果如图 5.7 (b) 所示，勾选需要下架的商品，点击“下架”，弹出是否确认下架的提示框，若用户点击确认，则删除数据库商品表中对应商品的信息，即为完成商品下架。本页面的结构文件为 downgoods.wxml，主要通过 check-box 组件实现勾选框，其核心代码如下：

```
<checkbox-group class="cbgroup1" bindchange="bindCheck">

<view class="list1" wx:for="{{productList}}" wx:key="produceId">

    <view>

        <checkbox value="{{item._id}}"/></checkbox>

    </view>

    <view>

        <image class="image1" src="{{item.imageUrl}}"/

        data-productid="{{item._id}}" bindtap="tapDetail"></image>

    </view>

    <view class="detail1">

        <text>{{index+1}}. {{item.productName}}</text>

        <view class="priceInfo1">

            <text class="price1">¥ {{item.price}}</text>

            <text class="oriPrice1">{{item.originalPrice}}</text>

        </view>

    </view>

</view>

</checkbox-group>
```

5.3 预约功能的实现

5.3.1 线上预约模块的实现

线上预约页面的实现效果如图 5.8 所示，用户使用上滑选择框选择预约项目、预约日期、预约时间、填写姓名和电话，点击“提交”，后台将预约信息写入数据库预约信息表中即为完成线上预约。此页面的核心在于使用 picker 组件实现上滑选择框，其核心代码如下：

```
<text class="text1">预约项目：</text>
```

```

<picker mode='selector' bindchange="bindPickerChange" value="{{index}}"
range="{{selectAry}}">

    <view class="input0">当前选择: {{selectAry[index]}}</view>

</picker>

<text class="text1">预约日期: </text>

<picker mode="date" value="{{date}}" start="2022-04-18" end="2025-12-31"
bindchange="bindDateChange">

    <view class="input0">

        当前选择: {{date}}

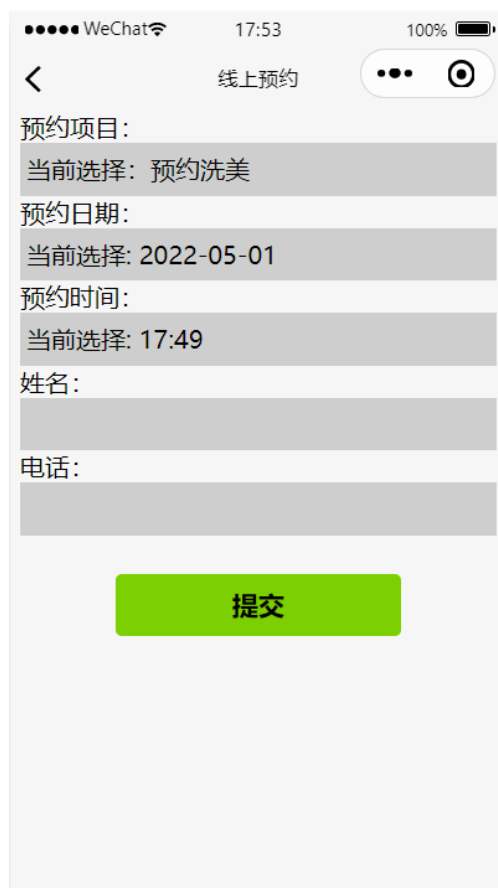
    </view>

</picker>

```



(a)



(b)

图 5.8 线上预约页面

5.3.2 处理预约模块的实现

处理预约页面的实现效果如图 5.9 所示，工作人员可以查看当前未完成的预约信息，可以点击“已完成”按钮将预约状态改为已完成。



图 5.9 处理预约页面

5.4 急救功能的实现

5.4.1 宠物急救模块的实现

宠物急救页面的实现效果如图 5.10 所示，用户填写急救信息并提交，系统提示用户拨打急救电话确认急救信息，用户点击确定则使用 wx.makePhoneCall 调用手机拨打电话功能转入拨号页面。其核心代码如下：

```
wx.showModal({
  title: '提交成功',
  content: '已收到您的呼救，请你拨打 8888888 确认信息，我们将尽快前往救援',
  showCancel: false,
  success (res) {
```

```

if (res.confirm) {
    wx.makePhoneCall({
        phoneNumber: '8888888' //仅为示例，并非真实的电话号码
    })

    setTimeout(() => {
        wx.redirectTo({
            url: '/pages/client/client',
        })
    }, 3000)
} else if (res.cancel) {
}
}
})

```



(a)



(b)

图 5.10 宠物急救页面

5.4.2 处理急救模块的实现

处理急救页面的实现效果如图 5.11 所示，工作人员可以查看当前未出发的急救信息，可以点击“已派遣医生出诊”按钮将急救状态改为已出发。



图 5.11 处理急救页面

5.5 知识功能的实现

知识功能由养宠知识页面实现，实现效果如图 5.12 所示。用户可以点击想查看知识的标题或标题后的下拉按钮，来展开下拉框显示具体的知识内容。此页面的逻辑文件为 knowledge.js，其核心代码如下：

```
onExplain: function(e) {  
    // 点击项目名称，收缩||展开 子系统  
    var thizz = this;  
    let index = e.currentTarget.dataset.index;  
    let projects = thizz.data.projects; // 定一个新变量存储数据，点击时获取当前字段并改变  
    projects[index].isOpen = !projects[index].isOpen  
  
    thizz.setData({  
        projects: projects  
    })  
}
```

此页面的结构文件为 knowledge.wxss，其核心代码如下：

```
<view class="wx-app-item" wx:for="{{projects}}" wx:for-item="items" wx:key="key">
  <view>
    <view class="wx-item">
      <view class="kkk" data-index="{{index}}" bindtap="onExplain">
        <text>{{items.name}}</text>
        <image wx:if="{{!items.isOpen}}" src="/images/jiantou.png" class="wx-more"></image>
        <image wx:if="{{items.isOpen}}" src="/images/jiantou1.png" class="wx-more"></image>
      </view>
      <view wx:if="{{items.isOpen}}">
        <view class="info">
          <view>{{items.info.text}}</view>
        </view>
      </view>
    </view>
  </view>
</view>
```

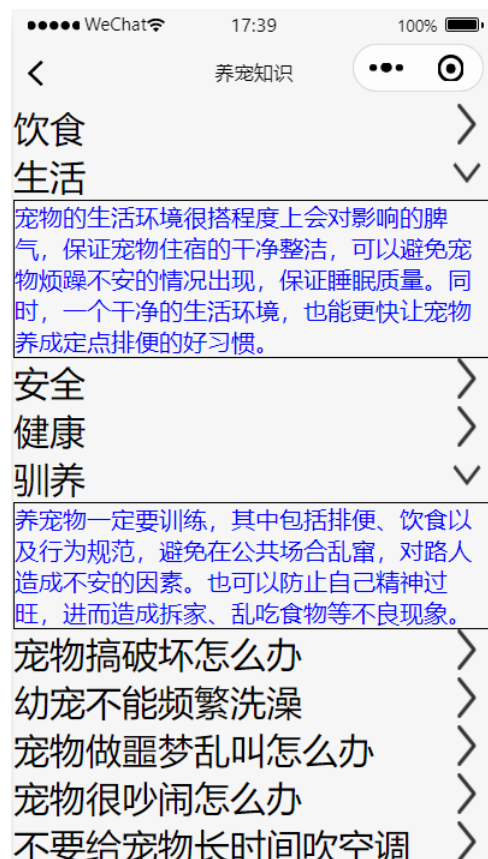


图 5.12 养宠知识页面

6 系统测试

6.1 功能测试

各项功能测试及测试结果如表 6.1 所示。

表 6.1 功能测试

| 测试内容 | 测试结果 |
|-------------------------|----------------|
| 注册页面（宠物主人注册） | 选择后直接完成注册，转入首页 |
| 注册页面（工作人员注册） 验证码输入正确 | 完成注册，转入首页 |
| 注册页面（工作人员注册） 验证码输入错误 | 注册失败，等待重新输入验证码 |
| 点击宠物主人首页各按钮 | 跳转正常 |
| 点击宠物商城“加入购物车”按钮 | 转变为加减按钮 |
| 点击宠物商城加减按钮 | 数量改变正常 |
| 点击宠物商城“购物车”按钮 | 跳转正常 |
| 点击购物车页面加减按钮 | 数量改变正常、总价计算正常 |
| 点击购物车页面勾选框 | 显示正常、总价计算正常 |
| 购物车页面商品数量减为 0 | 提示是否删除该商品，结果正常 |
| 点击购物车页面“结算”按钮 | 跳转正常 |
| 修改收货地址页面手机号填写有误 | 提交失败，显示失败原因 |
| 修改收货地址页面手机号填写正确 | 提交成功 |
| 各订单页面 | 显示正常 |
| 点击待付款页面“去付款”按钮 | 跳转正常 |
| 点击待收货页面“确认收货”按钮 | 跳转正常 |
| 线上预约页面 | 各选择框点击正常、提交正常 |
| 宠物急救页面 | 各输入框输入正常、提交正常 |
| 养宠知识页面 | 下拉键正常，内容显示正常 |
| 点击工作人员首页各按钮 | 跳转正常 |
| 上架商品页面 | 信息输入正常、图片上传正常 |

| | |
|----------------|--------------------------------|
| 下架商品页面 | 勾选框正常、下架正常 |
| 处理预约页面 | 页面显示预约信息正常、点击“已完成”按钮提醒是否更改预约状态 |
| 处理急救页面 | 页面显示急救信息正常、点击“已派遣”按钮提醒是否更改急救状态 |
| 点击待发货页面“去发货”按钮 | 示意用户输入快递单号完成发货 |
| 各订单页面 | 显示正常 |

6.2 其他测试

性能测试：经测试，该宠物医院平台性能良好、按键点击后反应速度敏捷，满足用户需求。

兼容性测试：经测试，该宠物医院平台可以在各大系统各大手机的微信上平稳运行。

7 总结与展望

7.1 总结

本文主要介绍了一款宠物医院微信小程序的开发过程，该小程序基本满足了宠物主人和工作人员两类人群的使用需求，该小程序界面简洁、跳转迅速、功能较为完善，但仍有许多不足，比如支付功能仅采用模拟支付，未做到实际支付；页面 UI 设计不够美观，仍待优化。

在信息时代大背景下，宠物医院为了提高企业竞争力要求开发一款方便快捷的宠物医院平台，本文主要介绍开发该平台的主要流程：

（1）介绍当前开发宠物医院平台的软硬件背景和线下宠物医院的发展现状，探讨和分析开发宠物医院平台的必要性和可行性；

（2）对宠物医院平台进行需求分析，分析出两大板块和十大模块，再对各模块进行详细的需求分析；

（3）根据需求分析给出各个模块的详细设计，并给出大致的实现思路；

（4）展示各模块实现效果及核心代码；

（5）对小程序进行系统测试，记录测试结果。

7.2 展望

在信息时代，各行各业与移动应用的结合是大势所趋，信息技术也在不断更新迭代。企业要想不被时代淘汰，就必须学会变革转型。我们要想跟上时代的步伐，就必须不断学习，时刻走在技术前沿。

参考文献

- [1] 中国互联网络信息中心 (CNNIC). 第 49 次中国互联网络发展状况统计报告[R]. 北京: 中国互联网络信息中心 (CNNIC), 2022.
- [2] 东西部小动物临床兽医医师大会. 2019 中国宠物医院发展报告[R]. 厦门: 东西部小动物临床兽医医师大会, 2019.
- [3] Nicholas C • Zakas. JavaScript 高级程序设计[M]. 第四版. 北京: 人民邮电出版社, 2020: 20-30.
- [4] 荣蓉、穆心驰、何金刚、于连林. 实战微信小程序[M]. 北京: 电子工业出版社, 2017: 40-60.
- [5] Eric A. Meyer. CSS 权威指南[M]. 第三版. 北京: 中国电力出版社, 2007: 10-20.
- [6] Adam Freeman. HTML5 权威指南[M]. 北京: 人民邮电出版社. 2014: 20-30.
- [7] 微信开放文档[EB/OL].
<https://developers.weixin.qq.com/miniprogram/dev/framework/>, 2022.
- [8] Zhang Yali. DESIGN AND IMPLEMENTATION ONLINE SHOPPING MALL SYSTEM BASED ON ASP.NET[C]. Proceedings of 2017 2nd Joint International Mechanical, Electronic and Information Technology Conference (JIMET 2017)
- [9] Cheol Park. The Effects of Nationality and Shopping Orientations on Online Shopping Cart Abandonment[C]. IEDRC. Proceedings of International Conference on Economics Marketing and Management (ICEMM2013). IEDRC, 2013:72-76.
- [10] 徐光耀, 马佳琳. 基于 JavaEE 技术的网上商城管理系统的设计与实现[J]. 办公自动化, 2017:22-24+41.