# 《闲云》详细设计

# 1、系统概述

**1.1、系统简介**

闲云是一款面向校园用户的二手信息发布平台，基于微信小程序实现。一方面校园用户在日常学习生活中难免会产生一些闲置物品，另一方面有的用户对相关物品有需求可是没有必要或不愿购买全新的商品。有关的供求信息往往不平衡，这时一个二手信息发布平台呼之欲出。闲云可以为校园用户提供二手商品信息交换的平台，这样用户物品的闲置率将大大降低，资源会得到有效的再利用，减少资源浪费。

**1.2、系统运行环境**

硬件平台：安装有微信app的智能手机

操作系统：所有支持微信app的手机操作系统（包括但不限于Android，iOS）

数据库系统：微信小程序云开发提供的基于JavaScript的云数据库

编程平台：Windows操作系统

网络协议：https协议

**1.3、开发环境**

工程工具：微信web开发者工具 版本Stable v1.02.1904090

Notepad++ 版本v7.5.9

Microsoft Visio 2016

开发语言：JavaScript，HTML，CSS

# 2、数据结构说明

本章说明本程序系统中使用的全局数据常量、变量和数据结构。

2.1、常量

本系统中使用的全局数据常量如下。

|  |  |  |  |
| --- | --- | --- | --- |
| 数据文件名称 | 所在目录 | 常量名称 | 具体功能 |
| msgstorage.js | Comps/chat/msgstorage.js | msgStorage | 存储消息 |
| Msgtype.js | comps/chat/msgtype | msgType | 消息类型 |

2.2、变量

本系统中使用的全局数据变量、所在目录及功能说明如下。

|  |  |  |
| --- | --- | --- |
| 全局变量名称 | 所在目录 | 功能说明 |
| unReadMessageNum | app.js | 聊天室未读消息数 |
| UserInfo | App.js | 微信用户公开信息 |
| OpenId | app.js | 微信用户Openid |
| yourname | App.js | 聊天室用户名 |
| saveFriendList | App.js | 用户好友列表 |
| saveGroupInvitedList | App.js | 用户收到的加群邀请 |
| isIPX | App.js | 判断用户设备是否为iPhone X |

2.3、数据结构

包括数据结构名称，功能说明，具体数据结构说明（定义、注释、取值）等。

Table1: 表名：商品（goods）

数据结构各属性及其数据类型如下所示。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **属性** | 商品ID  **id** | 用户ID  openid | 商品名称  name | 类别  category | 图片  image | 商品描述  info | 价格  price | 创建  时间  time |
| **数据类型** | **String** | **String** | **String** | **String** | **array** | **String** | **String** | **Date** |

本数据结构存储用户上传的商品信息。

**注意图片image只保存图片的路径，图片会上传到存储空间中**

Table2: 表名：搜索历史（history）

数据结构各属性及其数据类型如下所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **属性** | 搜索历史ID  **id** | 用户ID  openid | 搜索历史描述  description | 搜索历史内容historyList |
| **数据类型** | **String** | **String** | **String** | **array** |

本数据结构存储用户的搜索历史记录。每一个用户的所有搜索历史纪录都保存在数组historyList中。

Table3: 表名：收藏夹（collect）

数据结构各属性及其数据类型如下所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | 收藏ID  **id** | 用户ID  openid | 收藏描述  description | 收藏商品内容like |
| **数据类型** | **String** | **String** | **String** | **array** |

本数据结构存储用户的收藏记录，每一个用户收藏的所有商品位于数组like中。

Table4: 表名：用户（user）   
数据结构各属性及其数据类型如下所示

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 用户ID  openid | 用户昵称  nickname | 用户位置  locate |
| **数据类型** | **String** | **String** | **String** |

本数据结构存储用户的个人信息，包括openid，用户昵称和用户位置（用户所在的学校和校区）。

# 3、模块设计

**3.1、软件结构**

按照功能的不同小程序划分出若干子系统，分别是首页index，发布商品页add，分类classify，搜索页search，结果列表页list，商品详细信息页detail，消息页index1，聊天室页chatroom，我的页index2，商品信息修改页面modify。其中classify向下分出5个模块：书籍页books，服装页clothes，电子页phones,零食页snacks,其他页others。我的页提供向两个子系统之间的接口，两个子系统分别是我发布的页my/release，我喜欢的页my/like。

各子系统、模块之间的关系图如图3.1所示。接口标注在了各个箭头上。



图3.1 模块关系图

**3.2、功能设计说明**

小程序页面下方的三个按钮“首页”、“消息”、“我的”控制向这三个页面的跳转。首页集中了发布、搜索和分类查看商品的功能，提供向各个子模块跳转的接口。消息页提供转向聊天界面的接口，用户可以在聊天室页面看到和谁聊过天以及聊天的内容。我的页集中了用户的个性化信息，用户可以在“我发布的”查看、修改和删除其曾经发布的商品的信息，可以在“我收藏的”页查看自己收藏的商品。

**3.3、index模块**

详细描述各功能模块的功能、数据结构、具体算法和流程等。

3.3.1、设计图



图3.2 index页面设计图

页面设计图如右。

3.3.2、功能描述

小程序主页面，提供向多个子页面跳转的接口，包括：搜索页面，商品分类页面，发布页面。在五种分类的按钮下边是数据库中所有商品信息列表显示。

3.3.3、输入数据

用户在搜索框中输入要搜索的商品名称。在后续处理搜索事件的页面中系统在数据库中检索用户输入的商品名称，如果没有这件商品小程序提示“暂无此类商品”。

该模块从数据库goods表中获取商品记录，由于是将所有商品信息显示在页面下方，所有没有设置搜索条件。

3.3.4、输出数据

模块在检索数据库后得到this.data.list，在wxml中使用wx:for将list中的所有商品信息逐条显示。每条商品信息框中显示商品的名称、类别、价格。

3.3.5、数据设计

本模块中的局部数据结构是this.data.list，他是页面在加载和显示时查询数据库goods表中的所有商品信息得到的所有商品信息的列表。Wxml通过wx:for将list中的所有记录逐条列表显示构成页面下半部分。定义与2.3节Table1表相同。

3.3.6、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

1. getdata()

所在文件：pages/index/index.js

功能：获取数据库goods表所有商品记录，将返回结果写入全局变量list

格式与一般JavaScript函数格式相同

参数：无

全局变量：this.data.list

局部变量：res,err

返回值：无

算法说明和使用约束：无

1. onLoad()

所在文件：pages/index/index.js

功能：监听页面加载，调用getdata()，清理小程序本地数据缓存

格式与一般JavaScript函数格式相同

参数：无

全局变量、局部变量、返回值：无

算法说明和使用约束：无

1. onShow()

所在文件：pages/index/index.js

功能：监听页面显示，调用getdata()

格式与一般JavaScript函数格式相同

参数：无

全局变量、局部变量、返回值：无

算法说明和使用约束：无

1. bindViewTap()

所在文件：pages/index/index.js

功能：跳转到商品信息发布页面../add/logs

格式与一般JavaScript函数格式相同

参数：无

全局变量、局部变量、返回值：无

算法说明和使用约束：无

1. goSearch(e)

所在文件：pages/index/index.js

功能：跳转到处理搜索事件的页面../search/search

格式与一般JavaScript函数格式相同

参数：无

全局变量、局部变量、返回值：无

算法说明和使用约束：无

1. click(e)

所在文件：pages/index/index.js

功能：处理跳转到商品分类页面

格式与一般JavaScript函数格式相同

参数：无

全局变量、局部变量、返回值：无

算法说明和使用约束：无

1. goDetail(e)

所在文件：pages/index/index.js

功能：跳转到商品详细信息页面

格式与一般JavaScript函数格式相同

参数：无

全局变量、局部变量、返回值：无

算法说明和使用约束：无

3.3.8 全局数据结构与该模块的关系

该模块访问了全局数据结构goods。

**3.4、add模块**

3.4.1、设计图

图3.3 add模块设计图

3.4.2、功能描述

Add模块实现用户输入商品名称、类别、价格和商品描述，选择商品图片，和提交商品信息到数据库、把图片文件上传到云端的功能。其中上传图片部分支持上传不超过3张图片。

3.4.3、输入数据

用户输入商品名称、价格和商品描述，从列表中选择商品类别。用户点击添加图片按钮后可以从手机相册选择图片也可以拍照得到图片。点击提交按钮上传以上所有数据。由于设置商品名称、价格和商品图片是必填项，所以检查出这些项目中的任何一项没填或者用户没有添加图片都会出现相应的错误信息提示。

3.4.4、输出数据

在用户进行提交操作后，用户输入的所有数据被封装成一个对象，这个对象的属性包括：商品名称、价格、类别、是否被收藏、描述、创建时间、图片在云端的路径。之后在图片上传完成后这个对象会作为一条数据记录上传到数据库的goods表中。图片会以文件的形式上传到云存储空间中。

3.4.5、数据设计

本模块中的局部数据结构名为toSubmit，意为“要提交的数据记录”，这个对象就是要上传到数据库中的记录。他的定义和goods表的数据定义完全一致，不论是属性名还是变量类型。数据项和数据属性定义详见2.3节Table1.云数据库由腾讯云设计和维护，数据文件和数据存储格式对外部不可见，用户不必关心具体的存储细节。

3.4.6、算法和流程

模块根据输入数据将商品信息和文件上传到云数据库和云存储空间的程序流程图如图3.3所示。



图3.4 add模块程序流程图

3.4.7、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

1. chooseImage(e)

所在文件：pages/add/logs.js

功能：从本地选择图片，将本地图片路径保存到this.data.images

格式：与一般JavaScript函数格式相同

参数：e

全局变量：this.data.images

局部变量：res

返回值：无

算法说明、使用约束：无

1. removeImage(e)

所在文件：pages/add/logs.js

功能：从this.data.images中移除一张图片

格式：与一般JavaScript函数格式相同

参数：e

全局变量：this.data.images

局部变量：res

返回值：无

算法说明、使用约束：无

1. bindPickerChange(e)

所在文件：pages/add/logs.js

功能：选择商品类别

格式：与一般JavaScript函数格式相同

参数：e

全局变量：this.data.index

局部变量：无

返回值：无

算法说明、使用约束：无

1. formSubmit(e)

所在文件：pages/add/logs.js

功能：判断数据一致性，将数据项提交到数据库中，将图片上传到云存储空间中，提交成功后返回首页，出现错误提示错误信息

格式：与一般JavaScript函数格式相同

参数：e

全局变量：this.data.images, this.data.fileID, this.data.array, this.data.index

局部变量：res,err,e

返回值：无

算法说明：首先检查数据完整性，如果数据不完整拒绝提交，提示用户补全数据。如果必填项数据完整程序进入第二步。第二步上传图片，并将图片在云端的路径保存到全局变量this.data.fileID中。如果上传失败出现模态框提示“图片上传失败”。第三步在图片全部上传完成后将数据提交到数据库，提交成功后显示模态框“提交成功”，返回小程序首页，上传失败出现模态框提示数据提交失败。

使用约束：变量过多依赖于本模块的全局变量，所以只适合在本模块中使用，不适合作为独立的全局函数供其他模块复用

3.4.8 全局数据结构与该模块的关系

该模块访问了全局数据结构goods，定义详见2.3节Table 1。

3.5、chat模块

3.5.1、设计图

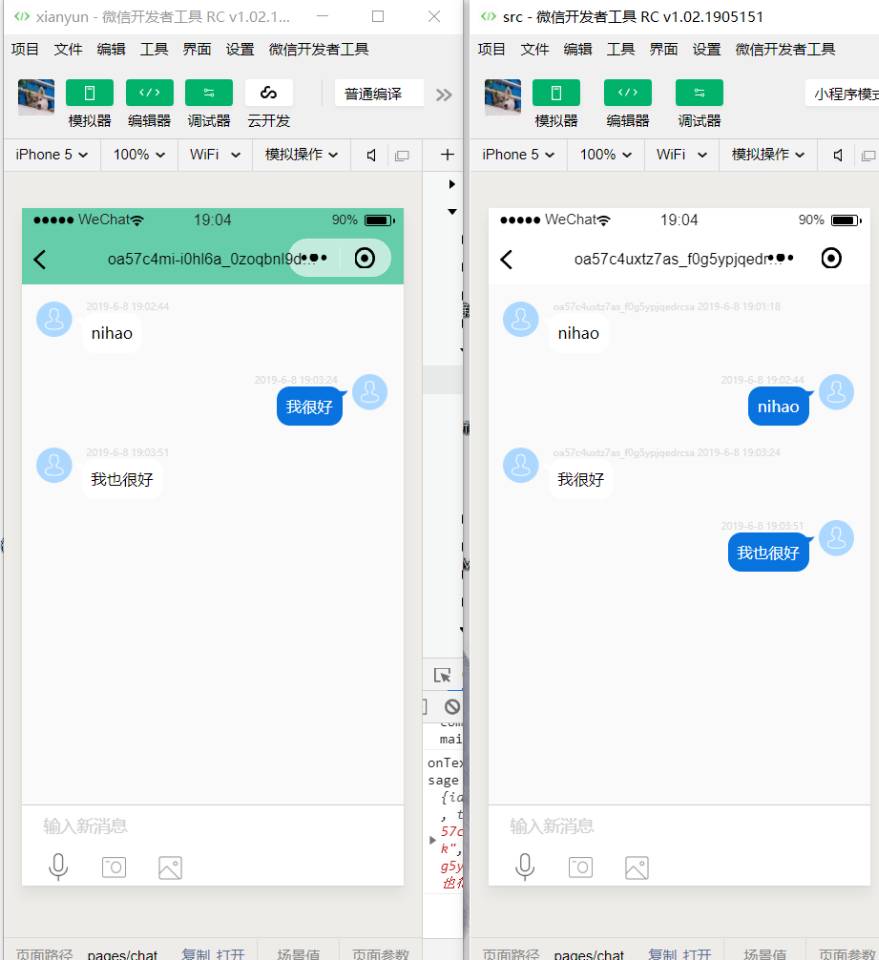


图3.5 chat设计图

3.5.2、功能描述

模块实现买家和卖家之间的聊天，支持发送文字和图片。

3.5.3、输入数据

用户通过手机键盘输入要发送的文字消息。输入框支持从手机相册中选择图片发送或者打开相机拍摄照片并发送。

模块聊天功能的实现主要是调用环信云的接口，从服务器中获取聊天记录数据。使用相关的查询语句就可以获取这些数据。

3.5.4、输出数据

用户发送的文字消息和图片在页面中以对话框的形式呈现。对话框上方标注消息的发送时间。

3.5.5、算法和流程

用户进入本小程序，后台就会记录其唯一标识符openid，并利用其openid注册环信账号，当用户点击聊天时，后台将商品所有者和该用户加好友，并利用环信的服务器和接口实现两人的聊天。

3.5.6、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

1. onload(options)

所在文件：pages/chatroom/chatroom.js

功能：监听页面加载

参数：options=系统传入的url参数

全局变量：this.data.username

局部变量：username

返回值：无

1. onPullDownRefresh()

所在文件：pages/chatroom/chatroom.js

功能：监听用户下拉动作

参数：无

全局变量：无

局部变量：无

返回值：无

**3.6、books模块**

3.6.1、设计图



图3.6 books模块设计图 图3.7 books功能流程图

3.6.2、功能描述

该模块实现书籍类商品记录的列表显示。

3.6.3、输入数据

该模块从数据库goods表中获取category（类别）为“书籍”的所有数据项。查询条件：category:”书籍”。

3.6.4、输出数据

模块查询到goods表的所有category为商品的数据项将这些数据项储存在this.data.list中，在books.wxml中以list的index为wx:key，wx:for遍历list将所有记录的名称、类别、价格、图片封装成一个信息框，列表显示所有书籍类商品。

3.6.5、算法和流程

该模块根据输入数据产生输出数据的程序流程图如图3.7所示。

3.6.6、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

1. getdata()

所在文件：pages/books/books.js

功能：查询数据库goods表中category=”书籍”的商品，将所有符合条件的数据项保存在this.data.list中

参数：无

全局变量：this.data.list

局部变量：res,err

返回值：无

(2)goDetail(e)

所在文件：pages/books/books.js

功能：跳转到商品详细信息页/pages/detail/detail?id=${e.currentTarget.dataset.id}

参数：e

全局变量、局部变量、返回值：无

3.6.7 全局数据结构与该模块的关系

该模块访问了全局数据结构goods，定义见2.3节table1。

**3.7、clothes, others, phones, snacks模块**

3.7.1、设计图

与3.6.1节设计图类似，限于篇幅不再配图。

3.7.2、功能描述

Clothes模块实现服装类商品信息列表显示，others模块实现其他类商品信息列表显示，phones模块实现电子类商品信息列表显示，snacks模块实现零食类商品信息列表显示。

3.7.3、输入数据

详细描述从物理模型中的哪些表获取数据以及获取这些数据的条件。

clothes模块从数据库goods表中获取category（类别）为“服装”的所有数据项。查询条件：category:”服装”。

others模块从数据库goods表中获取category（类别）为“其他”的所有数据项。查询条件：category:”其他”。

phones模块从数据库goods表中获取category（类别）为“电子”的所有数据项。查询条件：category:”电子”。

snacks模块从数据库goods表中获取category（类别）为“零食”的所有数据项。查询条件：category:”零食”。

3.7.4、输出数据

与3.6.4节类似，clothes模块列表显示所有服装类商品，others模块列表显示所有其他类商品，phones模块列表显示所有电子类商品，snacks模块列表显示所有零食类商品。

3.7.5、算法和流程

这四个模块根据输入数据产生输出数据的算法和流程和books模块相似，算法流程图也完全一致。只需要把图3.7开头的“查询书籍类商品所有记录”替换成查询服装、其他、电子、零食类商品所有记录。

3.7.6、函数说明

该部分与3.6.6节完全类似，限于篇幅，不再赘述

3.7.7 全局数据结构与该模块的关系

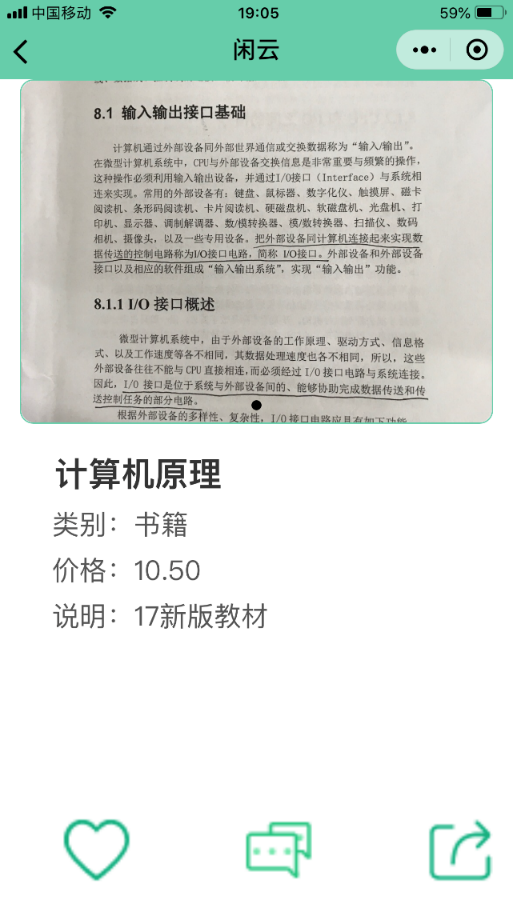
该模块访问了全局数据结构goods。

**3.8、detail模块**

3.8.1、设计图

如图3.8所示。

3.8.2、功能描述

该模块用于向用户展示商品详细信息，显示商品图片、名称、类别、价格和说明，图片以轮播图的形式展现数据库中该商品的所有图片。页面底部提供收藏、聊天、转发按钮。点击收藏则商品信息进入用户收藏夹，点击聊天则进入与卖家聊天的聊天室，点击转发用户可以和好友分享商品信息。

3.8.3、输入数据

LoadDetail函数从数据库goods表中获取该商品的信息，查询条件：\_id: \_.eq(productId)。GetCollect函数从数据库goods表中查询该商品信息，条件：\_id: productid。

3.8.4、输出数据

模块通过查询数据库得到商品图片、名称、类别、价格、描述，将这些数据呈现在页面中。

3.8.5、算法和流程

程序流程图如图3.9所示。

3.8.6、函数说明

1. function into\_chatRoom(event){}

文件：pages/detail/detail.js

功能：跳转到和商品所有者的聊天页面

参数：event

全局变量：app.globalData.yourname,

图3.8 detail模块设计图

app.globalData.OpenId, ownerid, productid,

IsCollect, token

局部变量：my,namelist

返回值：无

1. getToken()

文件：pages/detail/detail.js

功能：获取token

参数：无

全局变量：token

局部变量：res

返回值：无

1. add(u1, u2, pram)

文件：pages/detail/detail.js

功能：在环信后台将买家和卖家添加好友

参数：u1,u2,pram

局部变量：res

返回值：无

1. loadDetail(pram)

文件：pages/detail/detail.js

功能：在数据库中查询\_id=pram的数据项，将结果写入全局变量detail，并获取卖家的openid，为聊天室添加好友做准备

参数:pram

全局变量：ownerid, this.data.detail

局部变量：res 返回值：无

 图3.9 detail程序流程图

1. getCollect()

文件：pages/detail/detail.js

功能：查询该商品是否被用户收藏

参数：无

全局变量：IsCollect, this.data.collected

局部变量：res

返回值：无

1. bindCollect()

文件：pages/detail/detail.js

功能：处理用户的收藏相关操作

参数：无

全局变量：IsCollect, productid, this.data.collected

局部变量：res

返回值：无

3.8.7 全局数据结构与该模块的关系

该模块访问了全局数据结构goods。

**3.9、index1模块**

3.9.1、设计图

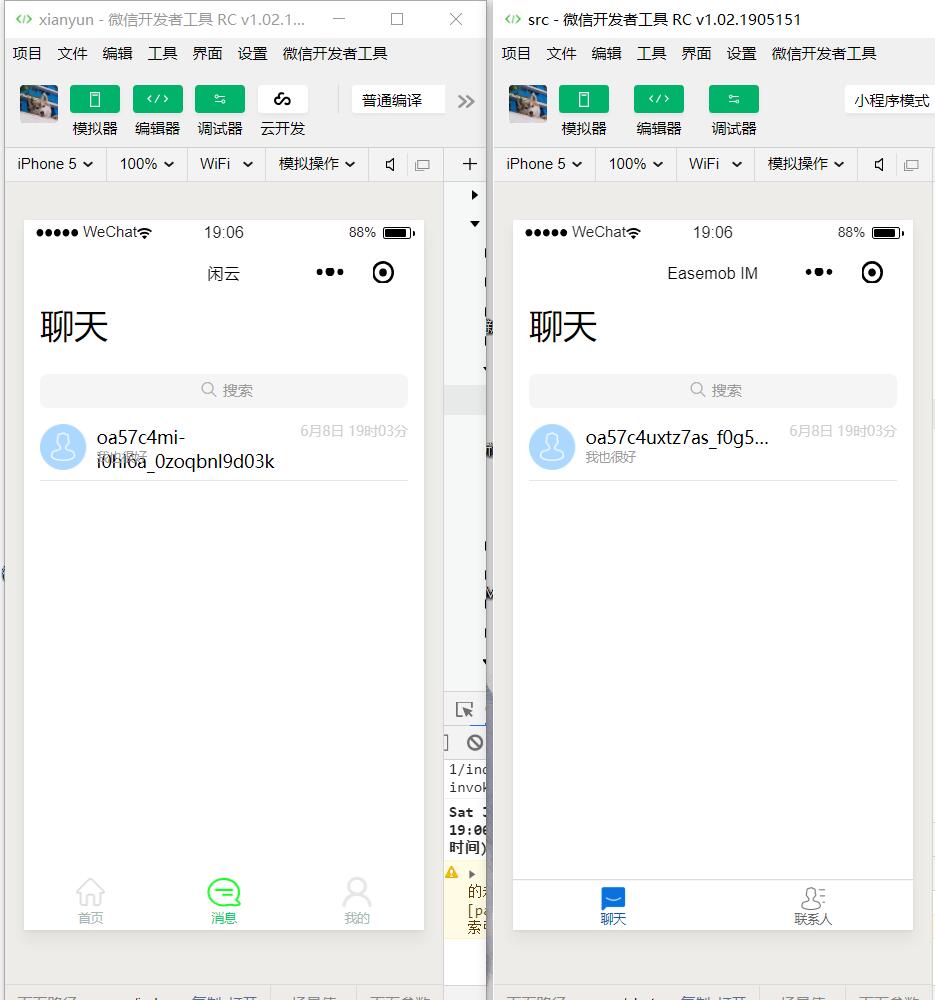


图3.10 index1设计图 图3.11 index2设计图

3.9.2、功能描述

本模块显示用户曾经发过消息的对话框，对话框显示用户和卖家交流过的最后一条消息。点击对话框用户可以进入与这个卖家聊天的界面。用户向左滑动对话框可以删除这个对话框，同时可以删除和这个卖家曾经聊天的记录。

3.9.3、输出数据

本模块基于环信聊天sdk开发，模块通过环信云服务器获取用户聊天历史数据，sdk在获取到数据后在聊天界面中将历史对话框展示。

**3.10、index2模块**

3.10.1、设计图

设计图如图3.11

3.10.2、功能描述

该页面具有获取用户信息的功能，用户点击头像则对小程序进行授权，小程序可以获取用户的公开信息。用户头像的下方有两个按钮：我发布的和我喜欢的，点击这两个按钮可分别跳转到用户发布过的商品列表页和用户收藏夹。

3.10.3、输入数据

用户点击获取头像昵称键小程序自动获取微信用户公开信息。

3.10.4、输出数据

小程序得到用户授权后在原来“获取头像昵称”键的位置显示用户的头像和昵称。

3.10.5、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

1. function click(e){}

所在文件：pages/index2/index.js

功能：跳转到“我发布的”、“我喜欢的”

参数：e

全局变量、局部变量、返回值：无

1. function getUserInfo(e){}

所在文件：pages/index2/index.js

功能：获取用户公开信息

参数：e

全局变量：app.globalData.userInfo，this.data.userInfo, this.data.hasUserInfo

局部变量、返回值：无

1. onLoad()

所在文件：pages/index2/index.js

功能：监听页面加载

全局变量：app.globalData.userInfo，this.data.userInfo, this.data.hasUserInfo，this.data.canIUse

局部变量、返回值：无

**3.11、search模块**

3.11.1、设计图

图3.12 search设计图

3.11.2、功能描述

该模块实现搜索商品的功能。该页面显示搜索历史纪录，点击相应的记录可以完成对该内容的再一次搜索。页面上有清空历史按钮，点击即可清空搜索历史记录。

3.11.3、输入数据

用户在搜索框输入要搜索的商品的名称。

和历史纪录有关的操作是从history表获取数据，查看历史纪录的条件是：

\_openid: this.data.openid,

\_id: 'history' + this.data.openid，

更新历史纪录根据记录主码'history' + that.data.openid

读取历史纪录根据记录主码'history' + that.data.openid，

清空历史记录的查询条件同样根据记录主码。

3.11.4、输出数据

产生的数据有inputValue和historyList，inputValue是用户的输入内容，传给list模块进行搜索。HistoryList是用户的搜索历史纪录，在wxml中遍历列表将所有内容显示在“搜索历史“的下方。

3.11.5、数据设计

相关数据库表是Table 2 History表。

3.11.6、算法和流程

详细描述根据输入数据产生输出数据的算法和流程。

用户输入搜索内容后点击“搜索”界面执行goSearch函数跳转到结果列表页list，同时将搜索内容append到history表主键\_id=’history’+this.data.openid的history中。在返回后页面再一次加载时再一次调用getHistory函数，刚才的搜索记录就显示在页面上。用户点击历史搜索内容，执行historyGoSearch函数，搜索相关内容。用户点击“清空历史”按钮会执行bindClearHistory函数，执行成功后会将history表主键为'history' + this.data.openid的数据项的historyList属性更改为空列表。同时将this.data.historyList更改为空列表。

**3.12、list模块**

3.12.1、设计图

图3.13 list设计图

3.12.2、功能描述

本页面实现搜索商品并将搜索结果列表显示的功能，搜索内容为search页传递进来的字符串。设置每页显示10条记录。页面具有上拉加载功能。

3.12.3、输入数据

从goods表中获取数据，查询操作的条件是：

name: {

$regex: '.\*' + content,

$options: 'i'

}

3.12.4、输出数据

LoadList函数查询到数据后将数据保存在页面全局变量this.data.list，wxml中遍历list逐条显示数据项。

3.12.5、算法和流程

页面加载时调用loadList函数，loadList函数在goods表中搜索符合条件的数据项，并将返回结果写入this.data.list中，然后页面显示所有的这些数据项，是分页显示，每页显示10条记录。用户上拉页面页面内容刷新。如果到了列表底部或者没有搜索到符合条件的商品，页面显示“---我是有底线的---”。用户点击数据项跳转到相应商品详细信息页。

3.12.6、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

函数名称：loadList

所在文件：pages/list/list.js

参数：content

全局变量：this.data.isOver, this.data.loading, this.data.noList, this.data.list, this.data.index, this.data.number

局部变量：res, that, db

返回值:无

3.12.7、 全局数据结构与该模块的关系

该模块访问了goods表。

**3.13、like模块**

3.13.1、设计图

和list界面基本一致，只是用户收藏夹为空时显示“暂无此类商品”。您可以参考图3.13.

3.13.2、功能描述

该模块是用户的收藏夹，列表显示用户收藏过的商品的记录。

3.13.3、输入数据

从goods表中获取collect属性值为true的所有数据项。

3.13.4、输出数据

getdata函数查询到数据后将数据保存在页面全局变量this.data.list，wxml中遍历list逐条显示数据项。

3.13.5、算法和流程

页面加载时调用getdata函数，getdata从goods表中查询所有属性collect=true的数据项，查询成功后将返回数据赋给this.data.list，之后遍历并显示。如果res.data.length==0，表示用户收藏夹为空，提示“暂无此类商品”。

3.13.6、函数说明

函数名称：getdata

功能：查询用户收藏的所有商品

全局变量：this.data.list

局部变量：res, err

返回值：无

3.13.7 全局数据结构与该模块的关系

该模块访问了goods表。

**3.14、release模块**

3.14.1、设计图

如图3.14所示。

图3.14 release设计图

3.14.2、功能描述

显示用户发布过的商品列表，用户可以查看、修改发布过的商品信息。如果商品发布5天后被暂时下架，用户需要在这里确认商品状态是继续在售还是永久下架。

3.14.3、输入数据

从数据库goods表中查询\_openid== wx.getStorageSync("myUsername")的所有数据项，将其写入this.data.list。

3.14.4、算法和流程

详细描述根据输入数据产生输出数据的算法和流程。

页面加载时调用getdata函数，getdata从goods表中查询所有属性\_openid== wx.getStorageSync("myUsername")的数据项，查询成功后将返回数据赋给this.data.list，之后遍历并显示。如果res.data.length==0，表示用户收藏夹为空，提示“暂无此类商品”。

3.14．5、函数说明

具体说明模块中的各个函数，包括函数名称及其所在文件，功能，格式，参数，全局变量，局部变量，返回值，算法说明，使用约束等。

1. getdata（）

和上一节getdata()基本相同，只是查询用户发布的商品

1. XiuGai(e)

功能：跳转到商品信息修改页面

参数：e

返回值：跳转到`/pages/modify/modify?id=${e.currentTarget.dataset.id}`

1. ShanChu(e)

功能：从数据库中删除该数据项

参数：e

全局变量：无

局部变量：res, err

返回值：无

3.14.6 全局数据结构与该模块的关系

该模块访问了goods表。

**3.15、modify模块**

3.15.1、设计图

如图3.15所示。

图3.15 modify设计图

3.15.2、功能描述

修改商品信息，在诸如用户填写错误想要修改信息的情况下进入这个页面修改。

3.15.3、输入数据

用户填写想要修改的项目，不填写默认为不修改。

从goods表中获取\_id==option.id的数据项。

3.15.4、输出数据

详细描述模块1所产生的数据以及这些数据的表现形式。

用户修改后的数据会更新到goods表中主键为parm的数据项，即可完成对数据库中该商品的相关属性值的更新。

3.15.5 全局数据结构与该模块的关系

该模块访问了全局数据结构goods。

# 4、 接口设计

**4.1、 用户接口**

1. 填写并发布商品接口：用来填写和上传商品的名称类别价格说明和图片

chooseImage(e) { wx.chooseImage({ ……… }) }

图片选择接口：定义上传图片的张数，大小类型 pages/add/log.js 45行

removeImage(e) { this.data.images.splice(idx, 1) }

删除图片接口：删除已经选择的图片 pages/add/log.js 59行

handleImagePreview(e) { wx.previewImage({ })}

图片预览接口：预览已经选择的图片 pages/add/log.js 65行

bindPickerChange(e) { this.setData({ })}

商品类别选择接口：选择商品的类别 pages/add/log.js 74行

formSubmit(e) { var information = e.detail.value;}

信息上传接口：上传商品的基本信息和图片 pages/add/log.js 82行

1. 搜索并查看商品信息的接口：搜索商品信息并展示

goSearch() { wx.navigateTo({ })}

搜索页接口：点击搜索栏进入搜索页接口 pages/search/search.js 32行

getHistory() { db.collection('history').get }

读取历史记录接口： 读取历史搜索记录 pages/search/search.js 116行

onHistory(content) { db.collection('history') }

添加历史记录接口：添加新的搜索历史记录 pages/search/search.js 69行

bindClearHistory() { }

清空历史记录的接口：清除历史搜索记录 pages/search/search.js 51行

goDetail:function(event) {wx.navigateTo({ })}

商品详情页面接口：转跳到商品的详细信息 pages/list/list.js 33行

loadList(content) { }

商品列表接口：模糊搜索，以列表展示信息 pages/list/list.js 52行

1. 收藏接口：收藏取消收藏商品

getcollect(){ }

查询收藏接口：查询商品是否被收藏 pages/detail/detail.js 125行

bindCollect() { }

收藏商品接口：点击按钮收藏，再点取消收藏 pages/detail/detail.js 152行

1. 查看发布商品接口：查看，删除修改商品的信息

getdata: function () { }

查看发布商品接口：查看自己发布的商品 pages/release/release.js 54行

XiuGai(e){wx.navigateTo({ })}

修改商品信息接口：修改自己发布商品信息 pages/release/release.js 29行

ShanChu(e){ db.collection('goods'). remove({ })}

删除商品接口 ：删除自己发布商品的信息 pages/release/release.js 35行

1. 聊天室接口

into\_chatRoom: function (event) { wx.navigateTo({ })}

进入聊天室接口：进入买卖双方聊天室 pages/detail/detail.js 43行

listGroups() { }

消息列表接口：查看消息列表 pages/index1/index.js 103行

**4.2、 外部接口**

**微信小程序服务器的接口**

wx.login({ success: res => { wx.request({ })}})

通过向微信小程序服务器请求获取用户的唯一标识符openid app.js 156行

**环信服务器的接口（聊天功能）**

function register(id) {WebIM.utils.registerUser(options);}

环信账号注册接口：利用用户openid自动注册环信账号 app.js 54行

function login(id) { WebIM.conn.open({ })}

环信账号登陆接口：利用注册的账号自动连接环信服务器 app.js 91行

getToken() { wx.request({ })}

获取Token接口：向服务器请求用户登录Token pages/release/release.js 70行

add(u1, u2, pram){ wx.request({ })}

添加好友接口：利用rest捆绑发布者添加好友 pages/release/release.js 91行

into\_chatRoom: function (event) { wx.navigateTo({ })}

进入聊天室接口：进入买卖双方聊天室 pages/detail/detail.js 43行

sendMessage(){WebIM.conn.send(msg.body);}

发送消息接口：买卖双方交流发送消息图片 comps/chat/inputbar/main.js 67行

WebIM.conn.listen({ })

监听接口：监听收到的消息，连接服务器状态 app.js 240行

require("sdk/libs/strophe");

引入环信sdk接口：导入环信聊天功能sdk app.js 1行

**4.3、 内部接口**

1. goDetail(e)接口将商品列表中的商品id作为路由，转跳到商品详情页面，检索并展示商品的详细信息

调用：

goDetail(e) { pages/index/index.js 73行

wx.navigateTo({

url: `/pages/detail/detail?id=${e.currentTarget.dataset.id}' })}

特别说明： 此接口在 pages/release/release.js pages/list/list.js pages/classify/ 等页面均有出现

2. into\_chatRoom接口将商品发布者的openid作为路由,转跳到与该商品发布者聊天的聊天室，实现聊天功能

调用：

into\_chatRoom: function (event) { pages/detail/detail.js 43行

wx.navigateTo({

url: "../chatroom/chatroom?username=" + JSON.stringify(nameList) });

# 5、数据库设计

**5.1、 数据库及数据表**

数据库系统：小程序云开发提供的数据库

数据库名称：闲云

数据表设计：

Table1: 表名：商品（goods）

属性：商品ID（id）、用户ID（openid）、商品名称（name）、类别（category）、图片（image）、商品描述（info）、价格（price）

Table2: 表名：搜索历史（history）

属性：搜索历史ID（id）、用户ID（openid）、搜索历史描述（description）、搜索历史内容（historyList）

Table3: 表名：收藏夹（collect）

属性：收藏ID（id）、用户ID（openid）、收藏描述（description）、收藏商品内容（like）

Table4: 表名：用户（user）

属性： 用户ID（openid）、用户昵称（nicknane）、用户位置（locate）

**5.2、 数据存储设计**

存储要求：符合相应的array或者string类型即可，符合小程序云开发的语法规则。

访问方法：js通过 db.collection（”table”）.where({ 条件}).add/ remove /update/get实现对数据库的增删改查

设计考虑：图片的存储可以通过图片转化为二进制码或者图片路径；转化为二进制码比较冗长，路径的存储相比之下比较简单，此时开发者本机也成了图片的存储空间。

保密条件：小程序云开发本身是在小程序的基础库层面的封装，你无需再引入其他库就可以使用，外部无法调用