# 《闲云》概要设计

# 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。我们做这个项目是基于这样的想法：大二学生在期末就将回迁本部，会有很多物品不便搬运只能留在这边，扔掉很可惜。这个平台为大二学生和大一学生搭建了一个沟通的桥梁，我们可以将闲置物品信息挂在平台上，大一学生如果刚好需要这些物品，就可以与卖家联系购买。这样可以让闲置的物品流动起来，减少资源浪费。

**1.2、系统目标**

在这一阶段程序要实现：

功能目标：用户可以提交商品信息、上传商品图片，商品信息分类分页显示、商品检索，与卖家聊天、显示聊天历史纪录，收藏商品、转发商品信息页

性能目标：提高数据上传速度、提高页面响应速度、提高检索速度

**1.3、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

**1.4、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、总体结构设计

**2.1、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。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2.1所示。接口标注在了各个箭头上。



图2.1 系统模块关系图

**2.2、设计思想**

小程序的基本设计思想就是把收藏、聊天、查看商品信息等功能分解到各个不同的模块，将相互关联的功能集中到一个页面上。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，提供像各个子模块跳转的接口。消息页提供转向聊天界面的接口，用户可以在这个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、模块设计

表3.1说明各模块的名称、功能、接口等信息。

表3.1 模块信息概览表

|  |  |  |
| --- | --- | --- |
| 模块名称 | 功能 | 接口 |
| add | 用户发布商品信息 | ChooseImage(e),removeImage(e),  handleImagePreview(e),  formSubmit(e) |
| Chat | 买家和卖家在线交流的聊天室页面 | Wx.navigateTo({url}),  onPullDownRefresh |
| Books | 图书类商品显示页面 | GoDetail(e),GetData() |
| Clothes | 服装类商品显示页面 | GoDetail(e),GetData() |
| Others | 其他类商品显示页面 | GoDetail(e),GetData() |
| Phones | 电子类商品显示页面 | GoDetail(e),GetData() |
| Snacks | 零食类商品显示页面 | GoDetail(e),GetData() |
| Detail | 商品详细信息展示页面，用户可以收藏该商品、进入与卖家聊天的聊天室、把该商品信息转发给好友 | LoadDetail(productId),  BindCollect(productId),  Getcollect(param),  Into\_chatRoom(event),  OnShareAppMessage(res) |
| Index | 小程序首页，商品信息显示和分类，发布商品，搜索商品 | BindViewTap(),goSearch(e),  Click(e) |
| Index1 | 历史聊天显示，搜索消息内容，未读消息显示，删除聊天记录，进入聊天室 | getChatList(),openSearch(),  onSearch(val),cancel(),clearInput(),  onInput(e),into\_chatRoom(event),  del\_chat(event) |
| Index2 | 获取用户信息，跳转到我的发布页和我的收藏页 | Click(e),wx.getUserInfo() |
| List | 显示列表页 | LoadList(content),goDetail(event) |
| Like | 用户的收藏夹 | GoDetail(e),GetData() |
| Release | 用户发布过的商品显示页，可以修改发布过的商品的信息或删除发布过的商品 | GoDetail(e),GetData() |
| search | 处理搜索操作，显示搜索历史 | bindKeyInput(e),goSearch(),  historyGoSearch(e),  bindClearHistory(),  onHistory(content),getHistory() |

**3.1、add模块**

3.1.1、功能描述

用户通过该模块发布商品信息。

3.1.2、接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| wx.switchTab({url:’pages/index/index’}) | information.images | 无 | 无 |

3.1.3、数据结构描述

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | 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** |

3.1.4、实现思路

对选择图片、删除图片、预览图片三个功能分别设计一个函数实现该功能。设计一个函数处理类别选择改变事件。对上传数据记录和文件部分，由于上传图片wx.uploadFile函数上传成功success方法的res获得的返回数据，也就是图片存储路径的数组外部不方便获取，所以在success方法里使用if条件判断，当图片全部上传完成后上传数据记录。

**3.2、chat模块**

3.2.1、功能描述

实现买家和卖家之间的聊天，支持发送文字、图片和表情包。

3.2.2、接口描述

环信提供的sdk接口

发送信息接口 ：WebIM.message(msgType, id)

事件监听接口 ：WebIM.conn.listen()

文字接收 ： onTextMessage()

图片接受 ： onPictureMessage()

表情接受 ： onEmojiMessage()

消息回执 ： onDelieverMessage()

退出聊天接口 ： WebIM.conn.close();

3.2.3、实现思路

主要是调用环信的接口，避免了搭建websocket的繁琐。

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。

**3.3、books模块**

3.3.1、功能描述

该模块实现书籍类商品记录的列表显示，功能流程图如图3.3.1所示。



图3.3.1 books功能流程图

3.3.2、接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| GoDetail | Tap | 跳转到  /pages/detail/detail?id=  ${e.currentTarget.dataset.id} | 无 |
| Getdata | 无 | Res.data | 出现提示框，提示“查询记录失败” |

3.3.3、数据结构描述

本功能区域使用的数据结构和3.1.3部分相同。

3.3.4、实现思路

通过查询云数据库获取书籍类商品全部记录，将所有记录保存到页面数据data.list数组内，在books.wxml内通过wx:for遍历list数组将所有记录显示出来。用户如果查看商品详细信息，点击商品对应的框通过goDetail函数跳转到相应商品详细信息显示页面。

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

3.4.1、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。它们的功能流程图和图3.3.1类似，只是将开始的“查询书籍类商品信息”改为查询服装类、其他类、电子类、零食类商品信息。

3.4.2~3.4.5与3.3.2~3.3.5相同

**3.5、detail模块**

3.5.1、功能描述

该模块主要显示商品图片、名称、价格、状态信息，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。

3.5.2、接口描述

获取商品详细信息接口：loadDetail(options.id) 获取商品的详细信息

商品收藏和取消收藏接口：getcollect bindCollect() 实现商品的收藏与取消

商品分享接口 onShareAppMessage() 实现商品的分享

聊天室跳转接口： into\_chatRoom 实现与商品发布者聊天的转跳

环信账号注册接口：register(openid) 利用用户的openid进行环信账号注册。

环信账号登陆接口：login(openid) 利用用户的openid进行环信账号登录

3.5.3、数据结构描述

用于收藏的collect 数据库表

Table3: 表名：收藏夹（collect）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** |

3.5.4、实现思路

获取商品详细信息：利用上一页面传递的商品Id参数，在数据库中检索该id，找到该商品，并显示商品的详细信息

商品收藏和取消收藏：先在数据库中查询是否收藏了该商品，如果是则当点击收藏按钮时，从收藏数据库中删除该商品；如果没有收藏，则在数据库中加上该商品

聊天室跳转接口： 利用用户openid作为路由，实现与商品发布者聊天的转跳

环信账号注册接口：利用用户的openid进行环信账号注册。

环信账号登陆接口 ：利用用户的openid进行环信账号登录

3.5.5、其他

用户只要进入detail页面点击聊天按钮，脚本就会自动为其注册环信账号，并且登录环信账号，让两者加好友，然后进入聊天页面，实现聊天功能。

**3.6、index模块**

3.6.1、功能描述

小程序的首页，提供部分商品信息列表显示，搜索框，发布按钮，以及各分类商品信息页面的按钮。

3.6.2、接口描述

表3.2描述了该模块与其他功能模块的每个接口的名称、输入信息、输出信息、异常处理。

表3.2

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| BindViewTap | 用户点击发布按钮 | 跳转到  pages/add/logs | 无 |
| GoSearch | 用户输入要搜索的商品信息 | 跳转到  pages/search/search | 无 |
| Click | 用户点击商品分类按钮 | 根据用户点击的商品分类跳转到该分类对应的页面 | 无 |

3.6.3、实现思路

利用分块设计的基本思路，将页面分成四大模块：搜索模块，分类模块，发布模块和商品展示模块，单独对其进行设计。页面提供了通往其他各模块的接口，用户执行某一操作小程序就跳转到处理相应操作的页面去。

3.6.4、其他

本模块有人机交互界面，类型是wxml和wxss渲染的web页面。

**3.7、index1模块**

3.7.1、功能描述

本模块和微信首页类似，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。页面上方提供搜索框，用户可以在搜索框中搜索历史聊天内容或曾与之聊天的卖家。点击搜索结果后同样可以跳转到与该用户的对话框。

3.7.2、接口描述

表3.3描述了该模块与其他功能模块的每个接口的名称、输入信息、输出信息、异常处理。

表3.3

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| Into\_chatRoom | Event | 跳转到页面"pages/chatroom/chatroom  ?username="+JSON.stringify(nameList) | 无 |

3.7.3、实现思路

首先获取聊天条目列表，页面渲染显示用户的聊天列表。设计一个处理搜索事件的函数处理用户对聊天内容和聊天对象的搜索。设计跳转到聊天室的接口，用户点击列表条目跳转到相应的聊天室。设计处理左滑条目事件的函数，在函数中将页面数据的chatlist该聊天条目对应的元素删除，同时删除相应的聊天记录数据。

**3.8、index2模块**

3.8.1、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。

3.8.2、接口描述

表3.4

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| Click | 用户点击按钮 | 跳转到我发布的页  Pages/my/release/release  或我喜欢的页  Pages/my/like/like | 无 |

3.8.3、实现思路

将头像与获取用户公开信息的接口wx.getUserInfo绑定，用户点击头像就可完成授权。这时原本灰色的头像变为用户的头像，头像下方显示用户昵称，表明小程序已获得用户授权。该模块提供通往其他页面的跳转接口。用户点击“我发布的”按钮可以跳转到小程序内的Pages/my/release/release页，点击“我喜欢的”可以跳转到Pages/my/like/like页。

**3.9、search模块**

3.9.1、功能描述

该模块实现搜索商品的功能，支持模糊查询。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纪录。

3.9.2、接口描述

表3.5

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| BindKeyInput | 用户在搜索框中输入的内容 | 无 | 无 |
| GoSearch | This.data.inputValue | 跳转到/pages/list/list?content=  ${content}  ，同时将搜索的内容添加历史纪录 | 如果搜索内容为空，不执行搜索操作 |
| HistoryGoSearch | 用户点击搜索历史记录中的一个 | 跳转到/pages/list/list?content=  ${content} | 无 |
| BindClearHistory | 用户点击清空历史按钮 | 出现提示框提示清空历史 | 无 |
| OnHistory | 用户的搜索内容 | 页面历史记录栏增加该记录 | 如果用户第一次搜索，向数据库中增加该用户的搜索记录；否则只需要修改数据库记录中的historyList属性 |
| GetHistory | 无 | Res.data.historyList | 无 |

3.9.3、数据结构描述

本模块使用的数据结构是

Table2 数据库的history表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | 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** |

3.9.4、实现思路

设计一个处理搜索事件的函数，在完成模糊查询的同时将搜索内容添加到搜索历史记录集合中。用户点击搜索后页面跳转到pages/list/list页面。模块有一个获取历史纪录的接口getHistory,用来从数据库中读取该用户的搜索记录并显示在历史记录栏中。用户点击以前的搜索内容也应该跳转到该搜索的搜索结果页中，需要有一个函数处理这一事件。模块需要设计一个清空历史记录的函数，与“清空历史”按钮绑定，以方便用户在不需要这些记录的时候将他们删除。

**3.10、list模块**

3.10.1、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。

3.10.2、接口描述

表3.6

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| OnLoad | Options | this.data.searchContent | 无 |
| Lower | 用户向上拉动页面 | 新的搜索记录列表 | 无 |
| LoadList | This.data.searchContent | 页面商品信息列表 | 没有新的内容，已到底就不再执行搜索 |
| GoDetail | 用户点击商品信息框事件 | 跳转到/pages/detail/detail?id=  ${event.currentTarget.dataset.id} | 无 |

3.10.3、数据结构描述

本功能区域使用的数据结构同3.1.3。

3.10.4、实现思路

在获取搜索内容后在数据库中进行查询，将查询到的结果构造成一个列表进行显示，用户如要查看商品详细信息，点击信息框跳转到pages/detail/detail页面。

**3.11、like模块**

3.11.1、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。

3.11.2、接口描述

表3.7

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| GoDetail | Tap | 跳转到  /pages/detail/detail?id=  ${e.currentTarget.dataset.id} | 无 |
| Getdata | 用户的openid | Res.data | 出现提示框，提示“查询记录失败” |

3.11.3、实现思路

在数据库history表中检索用户的收藏记录，将所有记录列表显示。用户如要查看商品详细信息，点击信息框转到page/detail/detail页面。

**3.12、release模块**

3.12.1、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。

3.12.2、接口描述

表3.8

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| GoDetail | Tap | 跳转到/pages/detail/detail?id=  ${e.currentTarget.dataset.id} | 无 |
| Getdata | 用户的openid | Res.data | 出现提示框，提示“查询记录失败” |

3.12.3、实现思路

在数据库goods表中检索用户的发布的商品记录，将所有记录列表显示。用户如要查看商品详细信息，点击信息框转到page/detail/detail页面。如果用户想修改商品信息，点击“修改”按钮将跳转到modify页面,对商品信息进行修改。

# 4、数据库与数据结构设计

**4.1、 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）

**4.2、 数据结构设计**

Table1: 表名：商品（goods）

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **名称** | 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** |

**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）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | 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** |

Table3: 表名：收藏夹（collect）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** |

Table4: 表名：用户（user）

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

**4.3、 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

# 5、接口设计

**5.1、外部接口**

表5.1描述小程序需要访问的外部接口的类型。

表5.1 外部接口类型一览表

|  |  |
| --- | --- |
| 外部接口 | 类型 |
| wx.chooseImage | 拍照或选择图片 |
| Wx.previewImage | 预览图片 |
| Wx.cloud.uploadFile | 上传文件 |
| Db.collection(‘table’).add/remove | 数据库记录增删 |
| Wx.getStorageSync | 访问本地缓存 |
| Wx.setStorage | 将数据存储在缓存 |
| Wx.getUserInfo | 获取用户信息 |
| Db.collection(‘table’).  where({条件}).get | 数据库记录查 |
| Db.collection(‘table’).doc(key).  Update/get | 数据库记录改查 |

**5.2、内部接口**

表5.2描述小程序主模块与其他模块及子系统的接口。

表5.2 内部接口一览表

|  |  |
| --- | --- |
| 内部接口 | 访问模块 |
| wx.switchTab | Pages/index/index |
| goDetail | pages/detail/detail?id=${e.currentTarget.dataset.id} |
| Into\_chatRoom | pages/chatroom/chatroom?username="+JSON.stringify(nameList) |
| bindViewTap | Pages/add/logs |
| goSearch | Pages/search/search |
| click | Pages/classify/\*/\* |