

**第 29 届“冯如杯”**

**学生学术科技作品竞赛论文**

**偕行**

——北航校园跳蚤市场

2019 年 4 月

摘 要

北航学生现在已经有自己的跳蚤市场，但现在仅为QQ群的形式，为了增加交易的安全性和便捷性。在我们团队进行调研后，选择了大家投票最多的微信小程序，进行"偕行"这一项目的开发，并希望以后能作为北航跳蚤市场的交易软件。同时也准备在得到信息办同意后接入学校的统一认证服务，以此来保障交易的安全，保护学生们的权益，方便定位购买、出售者。同时也可以帮助增加校内物品的流动性，买卖更加便利，发挥跳蚤市场的真正作用。

关于功能，其可以称之为：北航跳蚤市场微信小程序。但和“同袍”相像的，我们更愿意命名其为“偕行”，（脩我甲兵，与子偕行）让他人的物品，和购买者一起偕行。

关键词： 北航，跳蚤市场，微信小程序，偕行

Abstract

Beihang students now have their own flea market, but now only in the form of QQ group, in order to increase the safety and convenience of trading. After our team's investigation, we chose the Wechat program, which we voted most, to carry out the development of this project, and hope that it can be used as a trading software in the flea market of Beihang in the future. At the same time, it is also ready to access the unified certification service of the school after obtaining the consent of the Information Office, so as to ensure the security of transactions, protect the rights and interests of students, and facilitate the positioning of purchasers and sellers. At the same time, it can also help to increase the liquidity of school goods, facilitate the sale and play the real role of flea market.

Based on the function, it can be called: Beihang flea market Wechat applet. But like the "same gown", we prefer to name it "homing" (manicuring armour soldiers, homing with children) to let other people's goods go together with the buyer.

Key words: Beihang, flea market, Wechat program, go together

目 录

[摘 要](#_Toc32494) ⅰ

[Abstract](#_Toc32494) ⅱ

[正 文](#_Toc19456_WPSOffice_Level1) [1](#_Toc19456_WPSOffice_Level1)

[一、作品核心要点分析](#_Toc5497_WPSOffice_Level1) [1](#_Toc5497_WPSOffice_Level1)

[1.1项目产生背景](#_Toc93_WPSOffice_Level1) [1](#_Toc93_WPSOffice_Level1)

[1.2 项目创新点简介](#_Toc120_WPSOffice_Level1) [1](#_Toc120_WPSOffice_Level1)

[1.3 项目难点与解决方案](#_Toc5254_WPSOffice_Level1) [2](#_Toc5254_WPSOffice_Level1)

[二、 作品构成简介](#_Toc22320_WPSOffice_Level1) [3](#_Toc22320_WPSOffice_Level1)

[2.1 偕行信息展示](#_Toc5497_WPSOffice_Level2) [3](#_Toc5497_WPSOffice_Level2)

[三、 项目实现与代码分析](#_Toc2652_WPSOffice_Level1) [11](#_Toc2652_WPSOffice_Level1)

[3.1 分类页面与商品上传页面](#_Toc93_WPSOffice_Level2) [11](#_Toc93_WPSOffice_Level2)

[3.2 商品详情与收藏夹页面](#_Toc120_WPSOffice_Level2) [13](#_Toc120_WPSOffice_Level2)

[3.3 个人信息页面与地址设置页面](#_Toc5254_WPSOffice_Level2) [15](#_Toc5254_WPSOffice_Level2)

[3.4 个人店铺页面与常见问题页面](#_Toc22320_WPSOffice_Level2) [16](#_Toc22320_WPSOffice_Level2)

[结 论](#_Toc4428_WPSOffice_Level1) [19](#_Toc4428_WPSOffice_Level1)

[参考文献](#_Toc8885_WPSOffice_Level1) [20](#_Toc8885_WPSOffice_Level1)

正 文

一、作品核心要点分析

1.1项目产生背景

北航、央财已经一起创建了了简易的跳蚤市场，但仅为qq群的形式。所以，为了增加交易的安全和便捷性，经过QQ群内的投票调研，选择了得票最多的微信小程序，来进行"偕行"这一项目的开发。

希望能作为北航内闲置物品的交易软件。

1.2 项目创新点简介

1.迄今都没有一个真正好用的工具来实现这个功能，所以我们决定进行尝试。

2.采用微信小程序的方式进行实现，避免了安卓ios的兼容问题。

3.鉴于北航的两校区特色，建立专门的检索系统，和“北航镖局”进行联合，做到两校区间，每个校区内的交易无碍。

4.接入统一认证服务，免去繁琐的认证程序，同时可以保证安全（暂时仅校内开放，程序功能完善后再全面推出）。

1.3 项目难点与解决方案

1.商品上传和下载展示时和数据库间的交互实现

解决方案：前期利用微信自带的云开发进行商品数据库的交互，后期考虑到商品的数量会接入大内存的数据库，并采取云主机的交互方式。

2.为保证系统的安全性，保护商品和同学们的信息，设计出专门的密钥和解密系统。

解决方案：首先是统一认证的第一层安全保障；其次在密钥的设置、解密方面，会在后期学期了密钥设置的有关知识后尝试进行添加。

3.检索系统中算法的实现，以及对商品储存标签进行设计，保证搜索时同时拥有速度和精度

解决方案：标签的设置还在考虑当中，对于搜索算法，暂时使用的是微信小程序云云数据库中自带的搜索算法，后期转向外部数据库时会修改搜索算法为基于SQL语言的全新算法，使效率和精确性更加优秀。

4.统一认证服务的接入实现

解决方案：统一认证服务的接入，本团队已经有和信息化办公室进行联系，对方也在得知我们的项目后表示乐意让我们使用，在项目投入运行前会解决这一问题。

1. 作品构成简介

2.1 偕行信息展示



**图2-1 偕行首页**



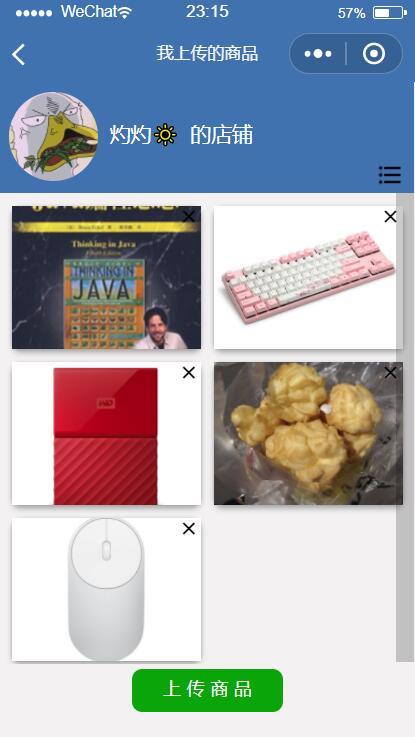
**图2-2 商品详情页面**



**图2-3 个人信息页面**



**图2-4 商品上传页面**

****

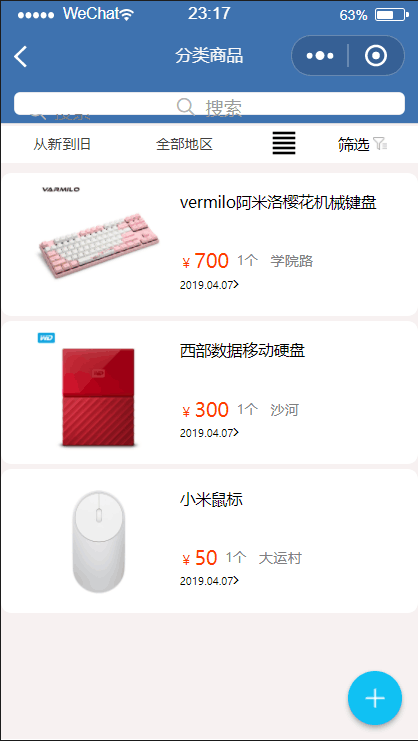
**图2-5 个人店铺页面**

****

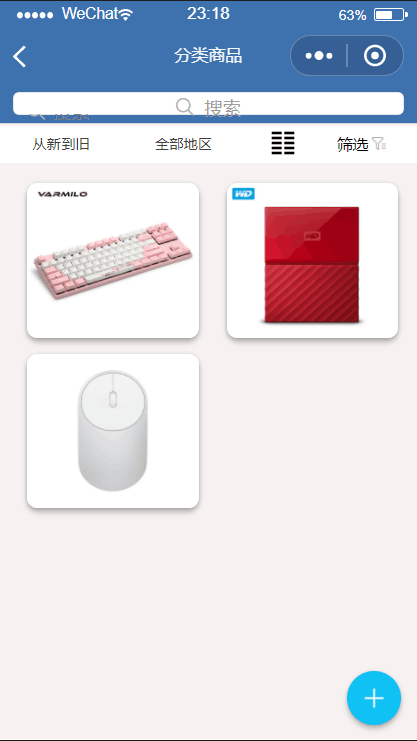
**图2-6 收藏夹页面**

****

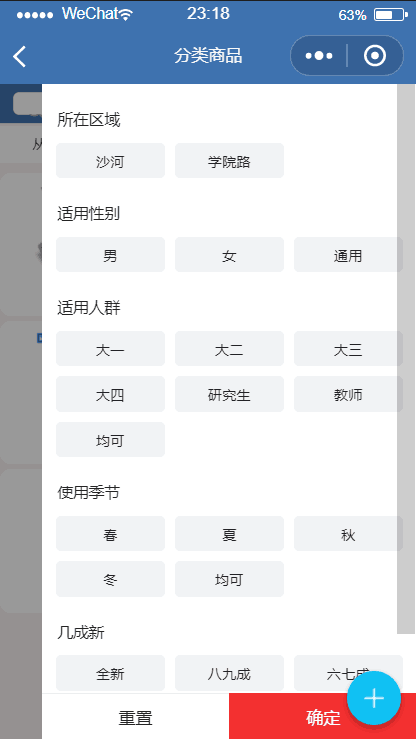
**图2-7 商品分类页面**

****

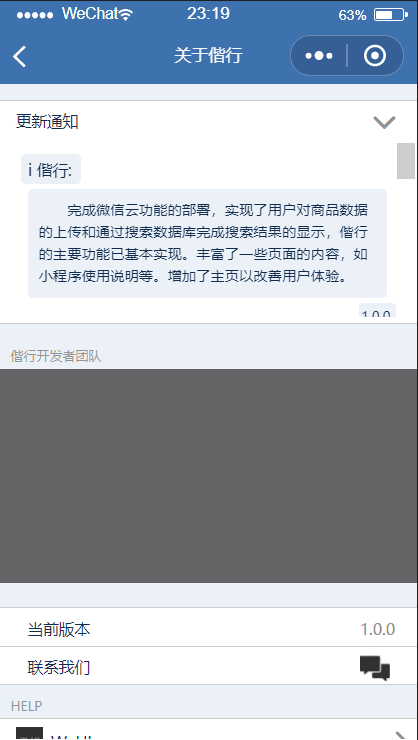
**图2-8 分类子类页面1**

****

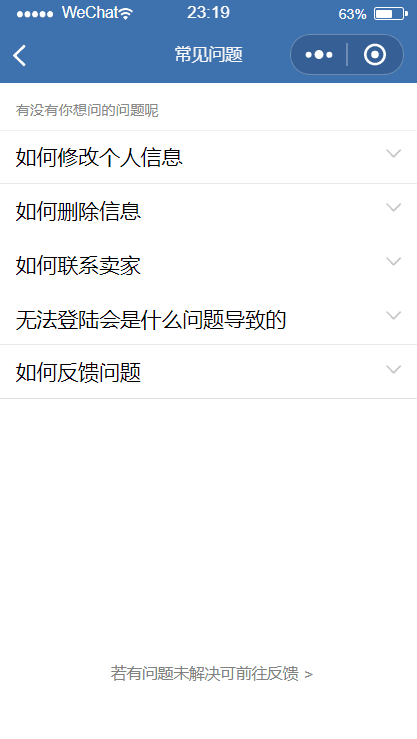
**图2-9 分类子类页面2**

****

**图2-10 筛选功能展示**

****

**图2-11 关于偕行页面**

****

**图2-12 常见问题页面**

****

**图2-13 信息设置展示**

****

**图2-14 地址更改页面**

1. 项目实现与代码分析

3.1 分类页面与商品上传页面

本项目的选择商品分类页面，我们预先制定了以下几个方案：1，首先以“学习”、“娱乐”等为主要分类建立选项卡，再在各个选项卡之下细分；2，将上述选项卡替换为大标题且所有分类放在同一个页面中；3，取消主要分类，直接将细分的分类放在同一个页面中。经过讨论，我们发现以现有的细分选项数量，如果再使用主要分类会显得繁琐，且某些选项如“电子产品”难以确定划分至“学习”或是“娱乐”。所以，我们最终采用了第三种方案，设置了“电子产品”、“书籍”、“生活用品”、“虚拟产品”、“零食”、“票券”、“其他”这七个分类选项。在后期的调整中，我们又将这七个选项修改、划分为“学习旧书”、“学习用品”、“娱乐用书”、“电子产品”、“虚拟产品”、“生活用品”、“化妆品”、“零食”、“票券”、“其他”这十个分类选项。如此分类，主要是分析了北航未来花园的跳蚤市场和咨询了同学的实际需求，发现主要的交易倾向集中在旧书尤其是课本、电子产品、化妆品这三类。所以，在页面布局上，我们将这三个选项按钮着重展示，具体表现为占据了整个页面的左半部分。该页面的代码实现较为简单，即用<view>标签设置多个选项按钮，然后进行外观的修改和排版布局，再增加相应的文字图片说明。在后期汇总时，还向各个按钮添加了点击事件，使得这些按钮能导航至相应的分类搜索结果页面。如果未来还需加入更多的分类选项，可直接增加<view>标签，设置相应的外观和事件即可。

此外，考虑到页面跳转的灵活性和避免单个页面元素过多，我们提供了悬浮按钮的功能，点击展开后有分别可导航至“关于偕行”、“收藏夹”、“常见问题”、“我的店铺”页面的子按钮。由于开发者工具并没有提供悬浮按钮的组件，我们使用了GitHub上的Wux Weapp组件库中的悬浮按钮组件，并进行了相应的调整，增加了可移动的容器区域使得该按钮可在整个页面上随意移动。在测试的过程中，我们发现该区域会覆盖分类按钮，导致一部分按钮无法触发点击事件。查阅相关开发文档后，我们对相应组件的层级进行了调整，将移动区域的层级放在最下层，成功解决了这一问题。

本项目的商品上传页面，考虑到北航两个校区相距较远的实际情况，在名称、价格等基本商品信息之外，我们增加了商品所在地（沙河或学院路校区）的选项和手机号、微信号的填写，以便买家卖家后续联系。代码实现上，名称、价格、描述这些信息采用了普通的文本框输入，并分别进行了相关限制，如名称最大长度、价格只能调出数字键盘等。商品所属分类、所在地这些具有固定选项的，使用了常见的滚动选项条。而商品图片的上传，则是通过开发者工具自带的文件上传组件。由于上述组件都是表单组件，故最后的提交按钮绑定了提交表单的点击事件，可将用户填写的以上信息提交。开发初期，在暂未开启云开发功能的情况下，该提交是将所有信息输出到控制台以便调试、确认。当中后期进入云开发环节、拥有云数据库和云存储的情况下，表单的提交便修改为了向云数据库中上传商品信息，具体为将单个商品的各个属性的文字信息附加上属性名上传至数据库，成为数据库中的一项。其中，由于该数据库无法直接存储图片，所以必须利用云存储功能存储图片，这就需要解决如何将数据库中的某一项和存储中的某一图片一一对应起来的问题，使得商品展示界面能正确显示信息和对应的图片。对此，我们的解决方案是点击上传按钮之后，先完成图片至云存储的上传，并用相应方法保存随机生成以防冲突的文件名字符串，再将该字符串作为商品属性的成员一起上传至云数据库。调用该商品信息时只需用该字符串在云存储中搜索即可完成一一对应。而在编写过程中，我们发现上传至云数据库时，文件名字符串经常为空。经过研究发现，由于相应的上传API是多线程执行，会发生图片还未上传至云存储、字符串未接收返回的文件名时，云数据库上传就已完成的冲突。我们的解决方案是将这两部分改为嵌套，强制在完成图片上传获取文件名字符串之后再进行其他部分的上传，成功解决了这一问题。

3.2 商品详情与收藏夹页面

3.2.1 商品详情页面

**实现的功能：**

1.显示相应物品的基本信息，包括图片、名称、价格；

2.显示物品的详细信息，包括详细介绍、数量、位置（也即在哪个校区，学院路或者沙河）、新旧程度、上传时间、商家微信号及商家电话号码。其中商家微信号及商家电话号码可点击复制到剪贴板，方便用户。

3.收藏按钮，实现收藏页面内的商品。

4.“+”按钮及其附属功能（我的店铺、常见问题、收藏夹、关于偕行），链接跳转到相关页面。

**如何实现：**

1.显示基本信息调用相关数据库，配合wxml和wxss的相关设计，即可完成。

2.显示物品的详细信息，也是通过调用相关数据库，配合wxml和wxss的相关设计来完成的，其中复制商家微信号和商家电话号码的功能通过setClipboardData函数实现设置系统剪贴板，通过currentTarget.dataset.index函数实现访问当前点击处的数据，通过getClipboardData函数实现获取系统剪贴板数据。通过点击事件即可实现将当前点击处的数据复制到剪贴板的功能。

3.“+”按钮及其附属功能：这个也不是我写的2333（但是代码看得懂）使用movable-view组件实现按钮固定在右下角，使用wux-fab-button组件实现这个按钮的雏形，接下来设定按钮的基本属性，在按钮的点击事件中链接到相关页面即可。

**页面设计想法：**

将用户最关注的信息放置在显眼的上半部分，并且字号较大，收藏按钮的位置也是一眼就可以看见，这样方便用户一眼看到关键信息。

下半部分的详细信息采用类似表格的表现形式，方便用户筛选自己想要了解的详细信息。一键复制商家相关信息的功能是为了方便顾客联系商家。

**可维护性分析：**

每一个view组件之间的分割清晰，且均是基于大边框进行位置设定，修改时较为容易，可维护性较好。

3.2.2 收藏夹页面

**实现的功能：**

1.根据用户收藏过的商品，显示商品的相关信息（图片、名称、价格、剩余数量、所在地址），其中图片实现根据长边保持原有长宽比例显示。

2.当用户要取消收藏时的删除按钮。

**如何实现：**  
 1.获取用户的open\_id，搜索到用户对应的数据库，获取该用户的collection数据库的商品的good\_id，展示相应物品及其信息。

2.对右上角的“X”设定点击事件，点击后即执行删库操作，在数据库中该用户对应的collection中删除该商品的相干信息。

3.设置image组件的mode属性为'aspectFit'，该属性可以实现图片保持原有长宽比显示全图。

**页面设计想法：**

在这个页面中参考了淘宝和京东的收藏夹页面，图片放置在左边、文字放置在右边也符合大多数使用者的习惯，这是较为常见的页面设置。

阴影的设置是为了让每个商品之间的分块更为明显，也使页面更符合大众审美。

**可维护性分析：**

每一个view组件之间的分割清晰，且均是基于大边框进行位置设定，修改时较为容易，可维护性较好。

商品的view组件通过class属性命名而不是id，使得每一个商品的外观设置相同，并且维护时只需要修改一次，即可实现整个页面的维护。

3.3 个人信息页面与地址设置页面

**功能简介：**个人信息界面的主要功能就是上传用户的基本信息，并在云数据库中予以储存。原定的计划是将设置界面的入口放在个人信息界面里，但考虑到设置选项不是很多，最后选择了新增悬浮窗的方法来解决设置界面的问题。

**实现方式：**调用云开发的API以及云函数login()，读取用户的openid，以每个用户独有的openid作为辨识，调用wx.cloud.database将用户信息储存为一条条单独的记录。用表格组件<form>和输入栏组件<input>将用户填写的记录储存在云数据库中。当用户需要更改信息时，调用云开发API db.collection().where()将原有信息保存在本地并利用<input>组件的placeholder属性将原有的信息作为提示反馈给用户。最后依靠条件判断，若已存在当前用户的记录，则使用db.collection().update()来更新记录，若不存在，则使用上述的db.collection().add()来新增记录。而且由微信云开发提供数据安全保障，保证所有用户无法读取到非自己创建的记录。

**页面布局：** 由于微信小程序<view>组件的灵活性，绝大多数组件使用的是<view>组件。并且充分利用了display: flex弹性盒子的特点，将需要平铺对称分布的单元，如底部的菜单栏等，格式化地调整好位置，方便统一风格，也利于后期维护。关于设计想法，将页面分为单独的块能让用户一眼了解到每个块的结构和功能，块内元素居中对称分布不失为一种经典的方案。考虑到操作的便捷性以及程序页面加载耗时，设置界面并没有单独地分为一个新的界面，而是做成悬浮窗的形式，一来便于管理，不用写很多的跳转函数，二来简化操作，优化用户体验。地址更改的页面，考虑到校区仅有三个，所以并没有使用,<input>组件要求输入校区名，而是使用<picker>组件，枚举出待选项，同样也是为了简化操作，还能保证数据的合法性。

**代码可维护性：**统一了代码风格，并且全局使用了块状分布和<view>组件，大大提高了后期代码的可维护性。

3.4 个人店铺页面与常见问题页面

3.4.1 个人店铺页面

个人店铺通过把店铺分为上下两大部分，上半部分引用了微信的开放组件获取姓名头像，通过和前一个页面（即个人信息）的对比实现无差别转场，保持体验的流畅性。在上半部分提供按钮可以改变排列方式，用绝对布局和隐藏溢出实现了两种排列方式的转换，并改进商品部分下部与底部的距离以减少空间压迫感。下半部分主要功能是增加商品，通过小程序提供的云开发功能实现网络交互实时更新，并且根据用户的登录后产生的\_openid以识别用户身份，并按照\_openid设置权限，保证不应该看见的受到保护，为了提供更好的体验，我们将网络交互功能放在页面之外，避免网络延迟造成的信息不同步。增删部分使用了云平台提供的api，动态改变用户的数据信息，并且减少了自己写服务器的开发与维护难度。

个人店铺作为个人信息的子页面，需要保持界面的一致性，故通过测量长度等详细数据保证了转场一致性。为了后期的维护方便，主要部分分成上下两半，避免任意一部分的头重脚轻。为进一步降低耦合度，我们通过尽量使用外部样式和外部逻辑代码，避免后期修改大面积改动局部代码。制作过程中遇到了设置border-radius无效的问题，经检查发现是内部元素撑开了外部容器，最后通过设置overflow为hidden实现了内外部分均显示为圆形的目的。此外受限于微信小程序的一些特性，产生了意外的条件渲染失灵的现象，我们通过采用另外一种办法即设置display属性绕开了这一问题，实现了点击之后成功切换排列方式的目的。商品的展示来源是云开发数据库，我们从云端获取数据之后暂存到本地的数组之中并编号，之后若是删除，就先在本地数组之中移除而不是先去云端，这样能减少浪费在网络延迟的时间，提升用户体验。老版的店铺设计中每次请求云端数据都会花费大量时间在网络延迟上，并且会遇到数据同步问题，最后通过本地化存储克服了难关。在18年4月份，因为微信的升级，原有的request方法失效，彼时我们并不知情，在尝试多次无法解决问题之后，我们才最终发现这个问题，通过调用新提供的组件完成了原有的设计。另外因之前的设计未考虑到后期增加绝对定位的问题，直接修改破坏了整体架构，通过恢复原来的架构与部分重构，我们最后保持了原有的不被破坏和新功能的正常添加。

3.4.2 常见问题页面

常见问题页面借助了开放组件wux的功能，其下的手风琴组件为大量信息的传递提供了可能，通过点击小标题即可查看或者隐藏该小标题下的详细信息，部分参考了淘宝的页面布局，配色以暖色调为主，让人有一种冲动购买的欲望。

常见问题最初的打算是使用类似表单结构来呈现，但高密度的信息会让用户感到嘈杂拥挤，因此我们从wux组件得到了灵感，借用他们已经做好的组件为自己所用，使信息更加精炼，并且小标题的可展开保证了完整性不受破坏。常见问题部分我们整理归纳并截取了部分购物网站上常见的问题，根据实际情况主要归纳出了以下五个模块，其中首先将大家最有可能遇到的问题即个人信息的修改摆在最醒目的位置，当用户发布或者购买时难免会涉及到地址的修改，因此能迅速查阅到个人信息修改方法是很有必要的。在使用过程中，涉及到货物售罄和出清的情况，发布者很可能需要删除相关信息，而消费者方面也会面临物品过期或者不再感兴趣的情况，此时提供一套删除的解决方案才是当务之急，因此第二个位置我们留给了信息删除。沟通是建立信任的桥梁，通过买卖双方充分有效且合理合法的沟通，能避免许多因为想当然的差错，为解决这一难题，我们提供了沟通渠道，方便双方坦诚相待互相了解，减少误会和摩擦的产生。初次载入很可能的情况就是无法登陆，我们根据实际情况总结了无法登陆的几个可能原因，并将其收录于问题部分，为新上手的用户提供便捷的咨询通道。最后为了提升用户体验同时也是为了自身的完善与进步，我们开辟了反馈问题渠道，通过收集问题为下一阶段的优化做好准备，当局者迷旁观者清，我们深知自己的认知难免存在偏颇与狭隘，因此广纳忠言是我们开发必不可缺的一部分，通过问题栏提供的方案或者直接点击最下方的反馈快捷渠道，用户可以参与体验改进，为我们的开发提供宝贵的箴言。通过我们与用户的沟通，相信许多考虑得不完全不充分的问题能够得以暴露并被纠正，能最大限度地保证产品的完善性。

制作过程中，我们最初使用并不是很熟悉的wxml，套用了曾经接触过的html知识，因为忽略语言特性的区别而且自身的不熟练，造成了我们最初的耦合程度和可复用性不尽人意，随着制作的进行，我们完整学习了html的一套知识体系，并且逐渐了解了一部分微信的特性，渐渐我们能分析出之前难以解决的症状的原因，也能通过逐渐的了解提前避免一些不希望的效果。为了兼容前期的代码，我们把难以处理的逻辑部分重新整理，用最少的改动翻新了整个页面，提高了代码的健壮程度，其中最早做的部分因为质量较低被完全舍弃重新制作，考虑到小部分作为基础部分的代码难以从头再改，在部分地方考虑了兼容处理以避免意想不到的结果，整体的可维护性随时间而越来越强，因为一直尽量追求方便后期改动，实际操作时能明显感受到后期的代码风格易懂而且易维护。

结 论

经过我们团队的开发，对微信小程序的使用已经轻车熟路了，现在基本功能也已经实现，后续还会继续完善。

相信在得到学校的支持后本小程序一定能够上线，并且成为北航学生中耳熟能详的一个校园工具。

参考文献

[1]微信开发团队.[微信小程序设计指南](https://developers.weixin.qq.com/miniprogram/design/)

[EB/OL].https://developers.weixin.qq.com/miniprogram/design/index.html?t=19040212.

[2]微信开发团队.[微信小程序开发](https://developers.weixin.qq.com/miniprogram/design/)指南

[EB/OL].https://developers.weixin.qq.com/miniprogram/dev/index.html?t=19040421.

[3]微信开发团队.小程序·云开发

[EB/OL].https://developers.weixin.qq.com/miniprogram/dev/wxcloud/basis/getting-started.html.