

关于“校园万事屋”小程序的说明介绍文档

组长：刘育麟

组员：陈泔鋈

组员：冯鑫泽

组员：陆张驰

组员：蒋沂霄

想法初衷：

- (1) 校园表白墙 plus，将闲置交换，表白，树洞，信息发布，失物招领，匿名聊天室等功能进行整合，以便快捷，方便地满足校内需求。同时取消传统表白墙中发布者的工作，节约人力。也便于使用者的使用，一站式满足基本需求。
- (2) 较之传统功能，在失物招领方面，可以通过小程序记录物品的流通信息，更便于追溯与确认。同时，对各种物品分类发布与展示，便于招领快速有效地进行。
- (3) 在树洞方面，在原有树洞的基础上，增设“匿名聊天室”功能。让“倾诉”不再是单向的传输，而是一种沟通与反馈。尤其是在疫情期间，或考试周这种心理压力较大的时期，便与同学更好的有效沟通，调节情绪，缓解烦恼与压力。
- (4) 在表白方面，区别于传统的表白墙，让表白与树洞不只于文字，而是图片与音频与视频的功能，使其更具有温度。
- (5) 在闲置交易方面，发布与查看更加简洁高效。可以有针对性地对不同分类物品进行查看，节省时间。

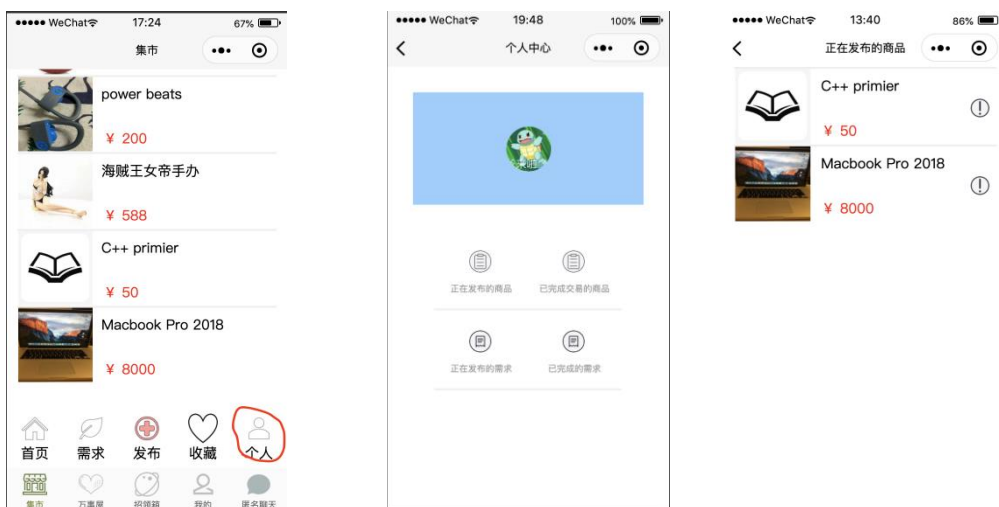
前置条件：用户使用微信认证登录，同时经过统一身份认证。

具体介绍：

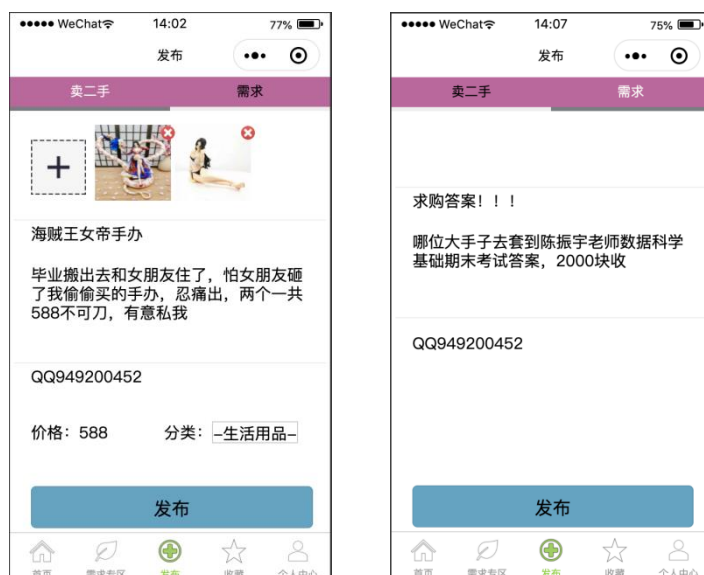
- (1) 万事屋集市（模拟“南小闲”）：



(1) 用户点击个人中心可查看自己正在发布的物品和需求、已完成的交易和需求



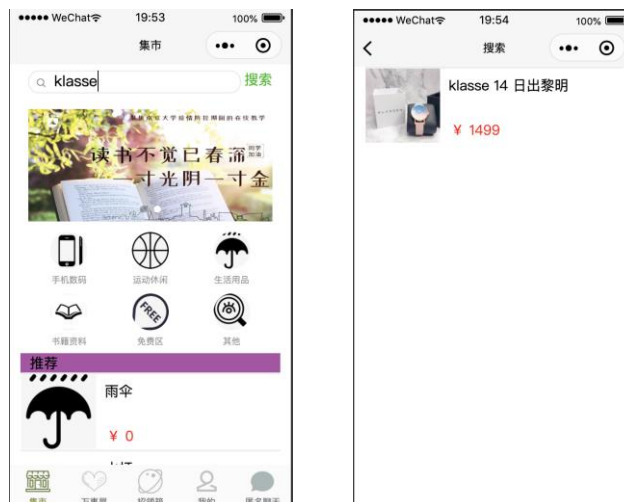
(2) 点击页面下方“发布”按钮可以发布自己想要出售的闲置物品或向他人求购闲置物品



(3) 点击页面下方“需求专区”可以浏览所有人（包括自己）的求购信息

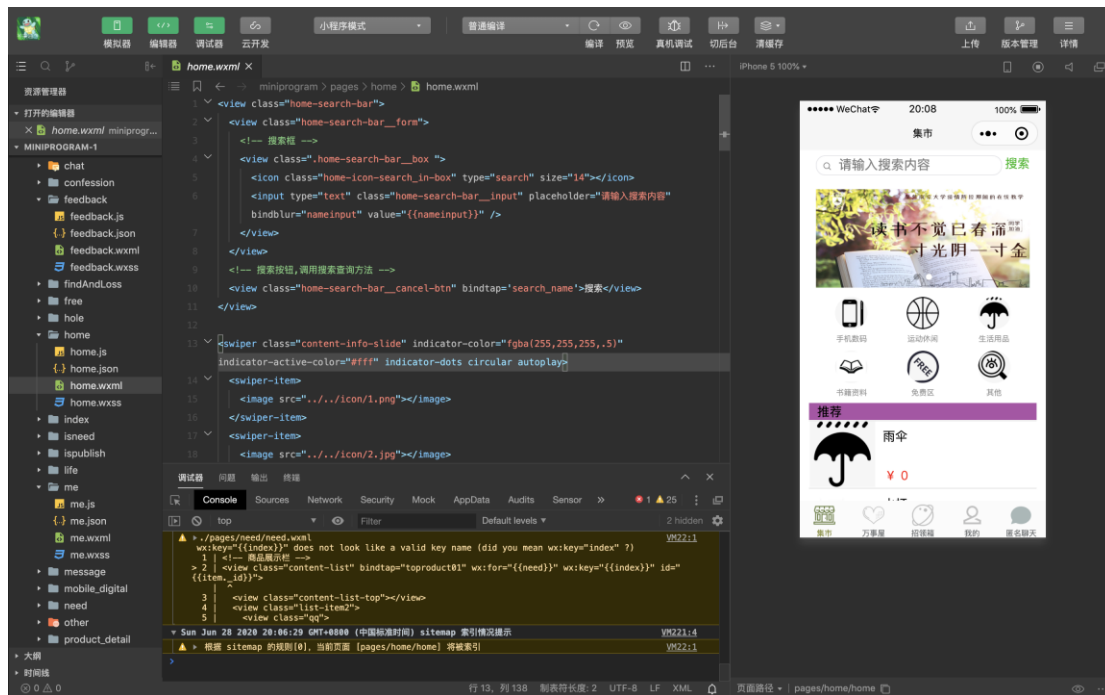


(4) 在首页点击搜索可以搜索相关物品信息

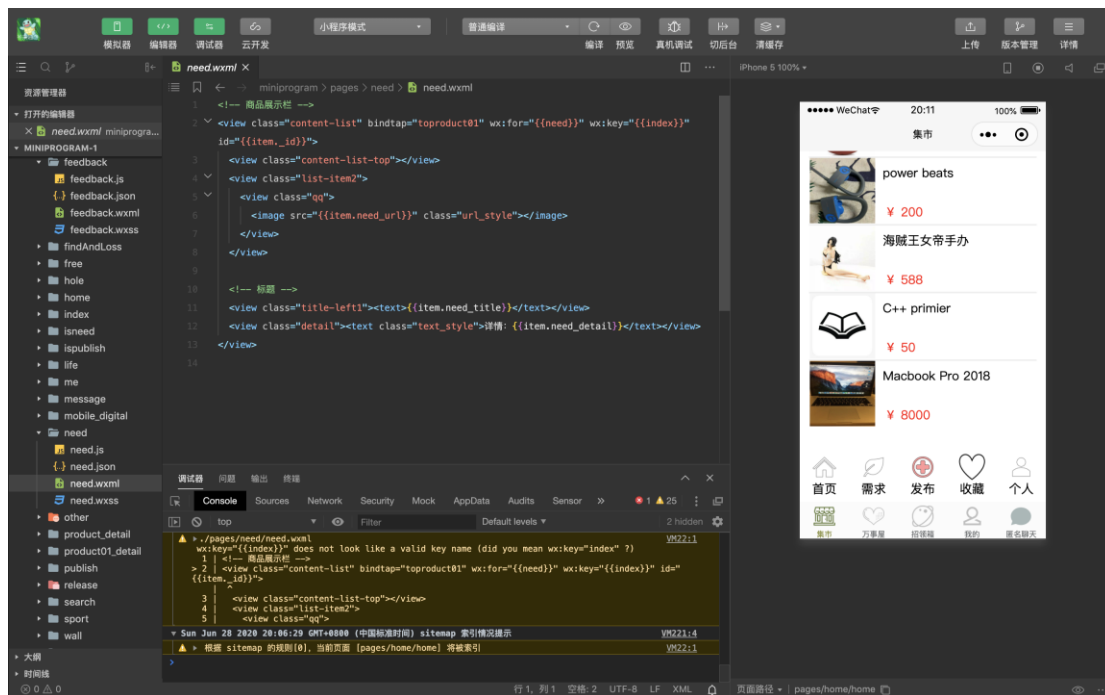


技术支持:

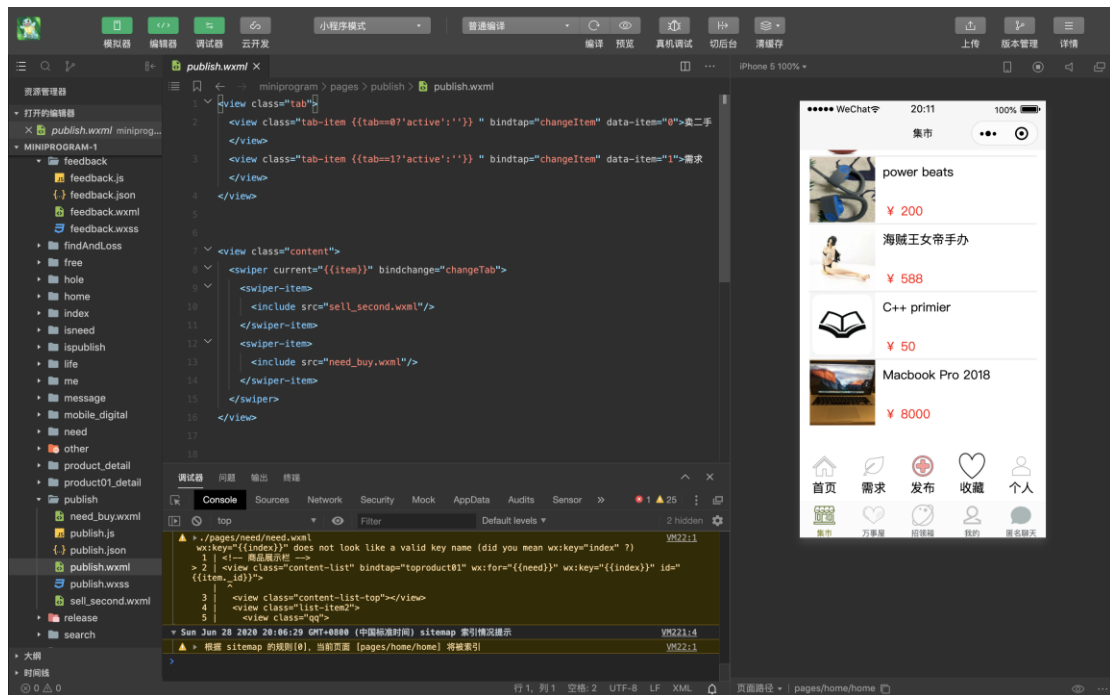
- (1) 微信小程序云开发技术
- (2) 在 pages/home/home 系列文件中完成对首页的构建



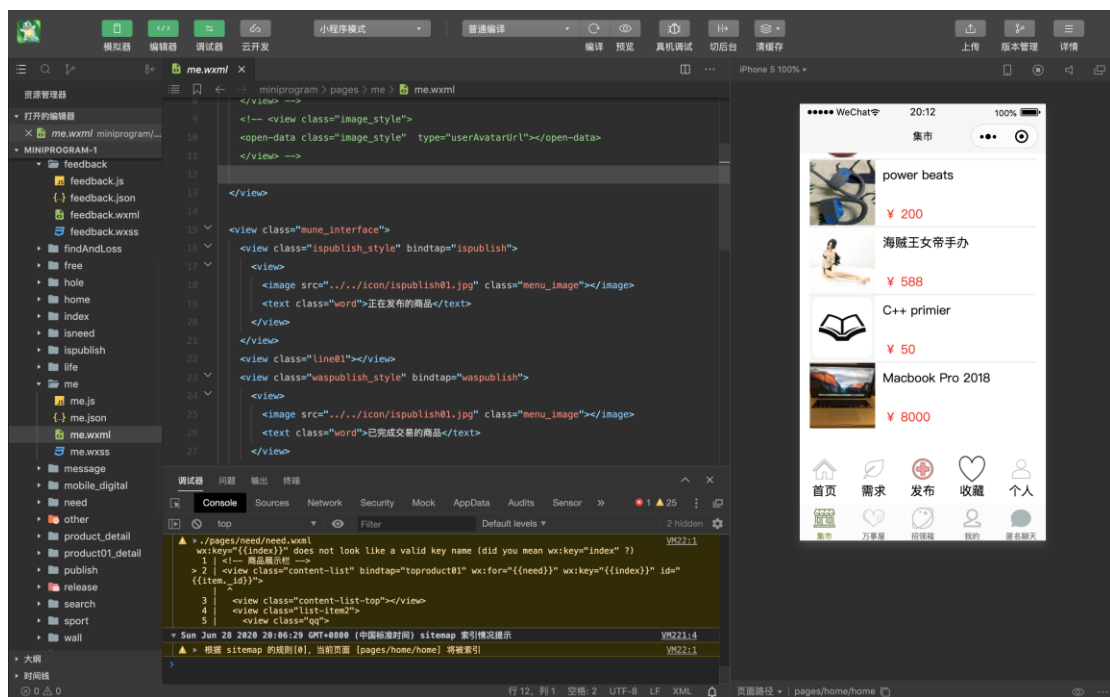
(3) 在 pages/need/need 系列文件中完成对需求专区页面的构建



(4) 在 pages/publish/publish 系列文件中完成对发布页面的构建



(5) 在 pages/me/me 系列文件中完成对个人中心页面的构建



(2) 万事屋页面 (模拟“表白墙”)



- (1) 使用者可以发布表白（显示发送者头像）或树洞（不显示发送者头像）到该程序上。（纯文字或文字+图片）



- (2) 使用者可以查看树洞或表白墙列表。



(图为树洞列表)



(图为表白列表)

(3) 使用者可以查看每日日志。

技术支持:

- (1) 小程序云开发技术
- (2) 组件间通讯使用了 properties。

代码展示:

- (1) 在 pages/wall/wall 系列文件中完成对于初始页面的构建。

```
2 <scroll-view scroll-y="true" class="container" bindscrolltoupper="upper" upper-threshold="10"
  lower-threshold="5" bindscrolltolower="lower" scroll-into-view="{{toView}}" scroll-top="{{
  {scrollTop}}">
3 <view class="search flex-wrap">
4   <view class="search-left flex-item">
5     <image src="../../images/search.png" style="mode:aspectFit;width:20px;height:20px;
  display: inline-block;"></image>
6     <input placeholder="搜索关键字, 寻找小纸条" placeholder-class="search-placeholder"/>
7     <view class="search-right flex-item" style="mode:aspectFit;width:20px;height:20px;
  display: inline-block;"></view>
8   </view>
9   <view class="title" style="text-align: center; margin-top: 10px;">
10    <image src="../../images/titlePic.jpg" style="mode:widthFix;width:100%;
  border-bottom-left-radius:20rpx;border-bottom-right-radius:20rpx;"></image>
11  </view>
12  <view class="intruoduction" bindtap="torelease">
13    <image src="../../images/release.png" style="width:65rpx;height:60rpx;margin-left:15%;
  margin-top:10%;background-color:#27321f;border-radius:50rpx;"></image>
14    <text class="introduction_text">点击发布我的内容...</text>
15  </view>
16  <view class="intruoduction" bindtap="todateTag">
17    <image src="../../images/canlendar.png" style="width:65rpx;height:65rpx;margin-left:15%;
  margin-top:10%;background-color:#27321f;border-radius:50rpx;"></image>
18    <text class="introduction_text">点击查看今日日志...</text>
19  </view>
20  <view class="intruoduction" bindtap="todetails">
21    <image src="../../images/clover.png" style="width:65rpx;height:65rpx;margin-left:15%;
  margin-top:10%;background-color:#27321f;border-radius:50rpx;"></image>
22    <text class="introduction_text">点击查看表白列表...</text>
23  </view>
24  <view class="intruoduction" bindtap="tohole">
25    <image src="../../images/secretkey.png" style="width:65rpx;height:65rpx;margin-left:15%;
  margin-top:10%;background-color:#27321f;border-radius:50rpx;"></image>
26  </view>
27 </scroll-view>
```

/miniprogram/pages/wall/wall.wxml 2.3 KB 80 master* 行 42, 列 6 WXML

ity AppData Audits Sensor Storage Trace Wxml 4


```
on    hole.wxml    lost.wxml    inputDetail.js    release.wxml    category.js    wall.wxss ×
1    /**index.wxss**/
2    .container{
3      height: 1500rpx;
4    }
5    .introduction{
6      margin-top:40%;
7      display:flex;
8      flex-direction:row;
9      align-items:center;
10     /* justify-content:center; */
11     background-color:■ #504f37;
12   }
13   .introduction_text{
14     padding-top:0%;
15     margin-top:0%;
16     margin-bottom: 10%;
17     margin-left: 10%;
18     /* justify-content: center; */
19     font-family: STHeiti Light[STXihei];
20     color:■ #637d3c;
21     /* background-color:#504f37; */
22   }
23   .btn-group{
24     background:■ #504f37;
25     size:mini; 1
26   }
27   .btn-release{
28     background:#504f37;
29     size:mini;
30   }

/miniprogram/pages/wall/wall.wxss  3.1 KB  80 master*  行 25, 列 15  WXSS
AppData  Audits  Sensor  Storage  Trace  Wxml  4  1 hidden
```

```
wall.js ×    confession.json    index.js - getWallRelease    index.js - getRelease
18    this.setData({
19      show: temp
20    })
21    console.log(this.data.inputVal)
22  },
23  torelease: function (e) {
24    wx.navigateTo({
25      url: '/pages/release/release',
26    })
27  },
28  todateTag: function (e) {
29    wx.navigateTo({
30      url: '/pages/canlendar/canlendar',
31    })
32  },
33  todetails: function (e) {
34    wx.navigateTo({
35      url: '/pages/confession/confession',
36    })
37  },
38  tohole: function (e) {
39    wx.navigateTo({
40      url: '/pages/love/love',
41    })
42  },
43  // goRelease: function () {
44  //    wx.navigateTo({
45  //      url: '/pages/release'
46  //    })
47  //  }
```

(tohole, torelease, toconfession, tocanlendar 完成从主页到各个页面的跳转)

(图为部分代码展示)

(2) pages/release/release 负责内容的发布。调用<love-category>组件。同时在

<love-category>组件中使用云开发中的云函数将发布的数据添加到数据库中。

```
32   methods: {
33     checkUploadForm(releaseType) {
34       // console.log(releaseType)
35       let that = this
36       let promises = []
37       if (that.data.releaseText) {
38         that.setData({
39           releaseType,
40         })
41         if (that.data.imgLocalPath.length > 0) {
42           wx.showLoading({
43             title: '正在发布...'
44           });
45
46           that.data.imgLocalPath.forEach(element => {
47             promises.push(that.promiseUploadImg(element))
48           })
49
50           Promise.all(promises)
51             .then((imgCloudPath) => {
52               console.log(imgCloudPath);
53               that.uploadRelease(imgCloudPath)
54             })
55             .catch((err) => { console.log(err); })
56
57         } else {
58           that.uploadRelease()
59         }
60       }
61     }
62   }
```

infoCard.js	release.js	love-input.js	love-input.json	love-input.wxml	release.wx
<pre>1 <!-- components/inputDetail/inputDetail.wxml --> 2 <view class="lost-found__info"> 3 <view class="releaseInfo"> 4 <!-- 输入详细 --> 5 <view class="infoInput"> 6 <view class="inputDetail"> 7 <view class="inputDetail_bd"> 8 <textarea bindinput="onInputChange" class="detail-textarea" placeholder="{ 9 {confessionOrHole}}" 10 value="{{releaseText}}" style="height: 220rpx" /> 11 <!-- data-clickStatus="{{clickStatus}}" --> 12 </view> 13 </view> 14 </view> 15 <!-- 上传照片 --> 16 <view class="uploadImg"> 17 <view class="uploader"> 18 <view class="uploader_files" id="uploaderFiles"> 19 <block wx:for="{{imgLocalPath}}" wx:key="*this"> 20 <view class="uploader_file" bindtap="previewImage" id="{{item}}"> 21 <image class="uploader_img" src="{{item}}" mode="aspectFill" /> 22 </view> 23 </block> 24 </view> 25 <view wx:if="{{imgLocalPath.length < 6}}" class="uploader_input-box"> 26 <view class="uploader_input" bindtap="chooseImage"></view> 27 </view> 28 </view> 29 </view></pre>					

```

1  {
2    "usingComponents": {
3      "love-input": "../../components/love-input/love-input"
4    },
5    "navigationBarTitleText": "发布"
6  }

```

(3) 在 pages/confession/confession 与 pages/hole/hole 中分别完成对于表白墙与树洞的内容的展示。同时调用<love-info>组件，并通过云函数完成对于数据库中数据的获取。

```

59  */
60  onShow: function () {
61    let that = this
62    wx.showNavigationBarLoading();
63    wx.cloud.callFunction({
64      name: 'getWallRelease',
65      data: {
66        tabBarText: that.data.tabbarText,
67        currentPage: that.data.currentPage
68      },
69      success: res => {
70        that.setData({
71          middleArr: res.result,
72        })
73        console.log(that.data.middleArr)
74        that.showPage()
75      },
76      fail: (err) => {
77        console.log(err)
78      },
79      complete: () => {
80        wx.hideNavigationBarLoading()
81      }
82    })
83  },
84
85  /**
86   * 生命周期函数--监听页面隐藏
87   */
88  onHide: function () {

```

```

31  */
32  onReady: function () {
33
34  },
35
36  showPage() {
37      // console.log('test');
38      let that = this
39      if (true) {
40          that.setData({
41              |   releaseInfo: that.data.middleArr
42          })
43      } else {
44          let newArr = []
45          //const val = that.data.thingsType
46          that.data.middleArr.forEach(item => {
47              if (true) {
48                  |   newArr.push(item)
49              }
50          })
51          console.log(newArr);
52          that.setData({
53              |   releaseInfo: newArr
54          })
55      }
56  },
57  /**
58   * 生命周期函数--监听页面显示
59   */

```

(4) 部分云函数展示:

index.js - getWallRelease	index.js - getRelease	confession.wxml	release.wxss	wall.js
<pre> 1 云函数入口文件 2 const cloud = require('wx-server-sdk') 3 const env = 'whisky-c8d7v' 4 cloud.init() 5 const db = cloud.database({ env }); //获取数据库的句柄 6 7 // 云函数入口函数 8 exports.main = async (event, context) => { 9 // const openId = cloud.getWXContext().OPENID; 10 let releaseList = await db.collection('user-release') 11 .where({ 12 deleted: false 13 }) 14 .get() 15 const currentPage = event.currentPage 16 const releaseSize = 1000 17 const sliceRelease = releaseSize * currentPage 18 const newReleaseList = releaseList.data.slice(-sliceRelease) 19 let returnResult = [] 20 for (let item of newReleaseList) { 21 const oneRelease = await db.collection(event.tabbarText) 22 .where({ 23 _id: item.releaseId, 24 deleted: false 25 }) 26 .get() 27 28 // returnResult.push(oneRelease.data) 29 if (oneRelease.data.length > 0) { 30 const userInfo = await db.collection('goodUser') </pre>				

```

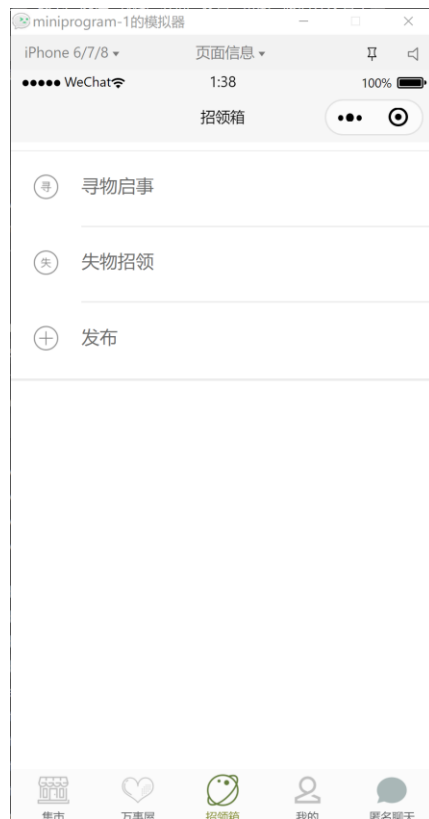
28   }
29   // 云函数入口函数
30   exports.main = async (event, context) => {
31     const userInfo = event.userInfo
32     const createTime = await getTime()
33     return await db.collection(event.releaseType).add({
34       data: {
35         releaseType: event.releaseType,
36         releaseImg: event.releaseImg,
37         releaseText: event.releaseText,
38         releaseCall: event.releaseCall,
39         releaseRemind: event.releaseRemind,
40
41         createBy: userInfo.openId,
42         createTime: createTime,
43         deleted: false
44       }
45     })
46     .then(res => {
47       return db.collection('user-release').add({
48         data: {
49           releaseId: res._id,
50           userId: userInfo.openId,
51           deleted: false,
52         }
53       })
54     })
55   }

```

(5) 以上仅为部分代码碎片展示，详见完整源码。

(3) 万事屋招领箱（模拟“权服侠”）

功能展示



- (1) 使用者可以发布 寻物启事 或 失物招领 到该程序上。(纯文字或文字+图片)



- (2) 使用者可以查看寻物启事和失物招领的列表，点击不同的分类可以显示不同的类型。通过联系方式可以联系到对方。





技术支持:

(1) 云开发技术:

利用微信开发者工具提供的云开发环境, 实现了云函数的使用, 同时利用云数据库, 将发布后的失物招领/寻物启事的数据上传到云端, 同时失物招领/寻物启事中带有图片将会上传到云端的文件存储中, 大大提高了程序的可用性。云开发可以很好的帮助我们同时完成前后端的配置, 在开发过程中加深我们对后端数据的理解和运用。

(2) 组件通讯

组件间的通讯使用了 **properties**

代码展示:

(1) 在 `pages/index/index.js` 中进行用户授权的时候, 读取用户信息, 并调用云函数 `createUser`, 在 `createUser` 中将用户信息记录到云端数据库 `goodUser` 中, 同时用户头像记录到云端存储中。

```

wx.cloud.callFunction({
  name: 'createUser',
  data: {
    avatarUrl: this.data.userInfo.avatarUrl,
    nickName: this.data.userInfo.nickName,
  },
  success: res => {
    // console.log(res);
    console.log('用户信息', this.data.userInfo)
    app.globalData.avatarUrl = this.data.userInfo.avatarUrl
    app.globalData.nickName = this.data.userInfo.nickName
    app.globalData.isLogin = true

    console.log(app.globalData.isLogin );
  }
})

```

```

// 先查询有无用户openId
const checkUser = await db.collection('goodUser')
  .where({
    openId: userInfo.openId
  })
  .get()

// 如果有用户，则更新基本用户信息
if (checkUser.data.length > 0) {
  await db.collection('goodUser').doc(checkUser.data[0]._id)
    .update({
      data: {
        avatarUrl: event.avatarUrl,
        nickName: event.nickName,
      }
    })
} else {
  const insertResult = await db.collection('goodUser').add({
    data: {
      avatarUrl: event.avatarUrl,
      nickName: event.nickName,
      openId: event.userInfo.openId,
      createTime: new Date()
    }
  })
}
}

```

(2) 在 `pages/findAndLoss/release/release.json` 中使用自定义组件 `category` 和 `inputDetail`，`category` 组件用来展示并选择不同的发布类别（失物招领/寻物启事），`inputDetail` 组件则是用来处理界面上输入的各种数据，同时调用云函数 `uploadInput` 来传输数据到云端数据库和存储，根据选择不同的发布类型（失物招领/寻物启事）会存到不同的数据库（found/lost），同时记录下发布人的 `openId`。发布结束后会跳转到对应的发布界面。

```
release.js  release.json  index.js ...\index  inputDet
miniprogram > pages > findAndLoss > release > {} release
1 {
2   "usingComponents": {
3     "category": ".../.../components/category/category",
4     "inputdetail": ".../.../components/inputDetail/inputDetail"
5   },
6   "navigationBarTitleText": "发布"
7 }
```

```
/**
 * 组件的方法列表
 */
methods: {
  tabClick: function (e) {
    this.setData({
      activeIndex: e.currentTarget.id
    })
    let outActiveIndex = e.currentTarget.id
    this.triggerEvent('onTitleChange', outActiveIndex)
  }
}
```

```
miniprogram > components > inputDetail > inputDetail.js > methods > uploadRele
promiseUploadImg(element) {
  return new Promise((resolve, reject) => {
    //一次只能上传一张图片到数据库
    const filePath = element;
    let a = filePath.lastIndexOf('.');
    let b = filePath.lastIndexOf('.', a - 1);
    let c = filePath.substring(b + 1, a);
    const cloudPath = c + filePath.match(/\.([^.]+)?$/);
    console.log(cloudPath);

    wx.cloud.uploadFile({
      filePath,
      cloudPath,
      success: (res) => {
        console.log('上传成功', res)
        resolve(res.fileID)
      },
      fail: (err) => {
        console.log('上传失败', err)
      }
    });
  })
},

uploadRelease(imgCloudPath) {
  console.log('uploadRelease');
```

行 107, 列 50 空格: 2 UTF-8 LF JavaScript

```

miniprogram > components > inputDetail > InputDetail.js > methods > uploadRelease

uploadRelease(imgCloudPath) {
  console.log('uploadRelease');
  // let imgCloudPath = imgCloudPath || []
  let that = this
  console.log(that);
  wx.cloud.callFunction({
    name: "uploadInput",
    data: {
      releaseType: that.data.releaseType,
      releaseImg: imgCloudPath,
      releaseText: that.data.releaseText,
      thingsType: that.data.thingsType,
      releaseCall: that.data.releaseCall,
      releaseRemind: that.data.releaseRemind,
    },
    success: () => {
      wx.hideLoading();
      console.log(imgCloudPath);
      const releaseType = that.data.releaseType
      wx.redirectTo({
        url: '/pages/findAndLoss/${releaseType}/${releaseType}'
      })
      console.log('/pages/findAndLoss/${releaseType}/${releaseType}');
      wx.hideNavigationBarLoading()
    },
    fail: (err) => {
      console.log(err);
    }
  });
}

```

```

cloudfunctions | 当前环境: whisky > uploadInput > index.js > getTime

exports.main = async (event, context) => {
  const userInfo = event.userInfo
  const createTime = await getTime()
  return await db.collection(event.releaseType).add({
    data: {
      releaseType: event.releaseType,
      releaseImg: event.releaseImg,
      releaseText: event.releaseText,
      thingsType: event.thingsType,
      releaseCall: event.releaseCall,
      releaseRemind: event.releaseRemind,

      createBy: userInfo.openId,
      createTime: createTime,
      deleted: false
    }
  })
  .then(res => {
    return db.collection('user-release').add({
      data: {
        releaseId: res._id,
        userId: userInfo.openId,
        deleted: false,
      }
    })
  })
}

```

(3) pages/findAndLoss/found/found 和 pages/findAndLoss/lost/lost 大体构成相同, 都使用组件 category 和 infoCard。category 组件用来选择并展示不同的类别的物品;

found.js/lost.js 调用云函数 getRelease 来读取 found/lost 数据库中的已发布信息及其与发布者的对应关系数据库 user-release 以及用户信息数据库 goodUser，然后用 infoCard 组件来组织每一个要显示的已发布信息，最后展示在页面上。

```
1 <!-- miniprogram/pages/lost/lost.wxml -->
2 <view class="navbar">
3   <scroll-view scroll-x="{{true}}">
4     <category tabs="{{['全部','证件','书籍文具','电子设备','生活用品','其他']}}" catch:onTitleChange=
5   </scroll-view>
6 </view>
7 <view class="lostContent">
8   <view class="inforCards">
9     <infocard id="infoCard" class="infoCard" releaseInfo="{{releaseInfo}}"></infocard>
10  </view>
11 </view>
```

```
miniprogram > pages > findAndLoss > found > JS found.js > onShow > name
onShow: function () {
  let that = this
  wx.showNavigationBarLoading();
  wx.cloud.callFunction({
    name: 'getRelease',
    data: {
      tabbarText: that.data.tabbarText,
      currentPage: that.data.currentPage
    },
    success: res => {
      that.setData({
        middleArr: res.result,
      })
      console.log(that.data.middleArr)
      that.showPage()
    },
    fail: (err) => {
      console.log(err)
    },
    complete: () => {
      wx.hideNavigationBarLoading()
    }
  })
},
/**
 * 生命周期函数--监听页面隐藏
 */
}
```

行 65, 列 24 (选中 10) 空格: 2 UTF-8 LF JavaScript

```
cloudfunctions | 当前环境: whisky > getRelease > index.js > main > exports.n
8 exports.main = async (event, context) => {
9   // const openId = cloud.getWXContext().OPENID;
10  let releaseList = await db.collection('user-release')
11    .where({
12      deleted: false
13    })
14    .get()
15  const currentPage = event.currentPage
16  const releaseSize = 1000
17  const sliceRelease = releaseSize * currentPage
18  const newReleaseList = releaseList.data.slice(-sliceRelease)
19  let returnResult = []
20  for (let item of newReleaseList) {
21    const oneRelease = await db.collection(event.tabbarText)
22      .where({
23        _id: item.releaseId,
24        deleted: false
25      })
26      .get()
27
28    // returnResult.push(oneRelease.data)
29    if (oneRelease.data.length > 0) {
30      const userInfo = await db.collection('goodUser')
31        .where({
32          openId: oneRelease.data[0].createBy
33        })
34        .get()
35    }
36  }
37  return returnResult
38}
```

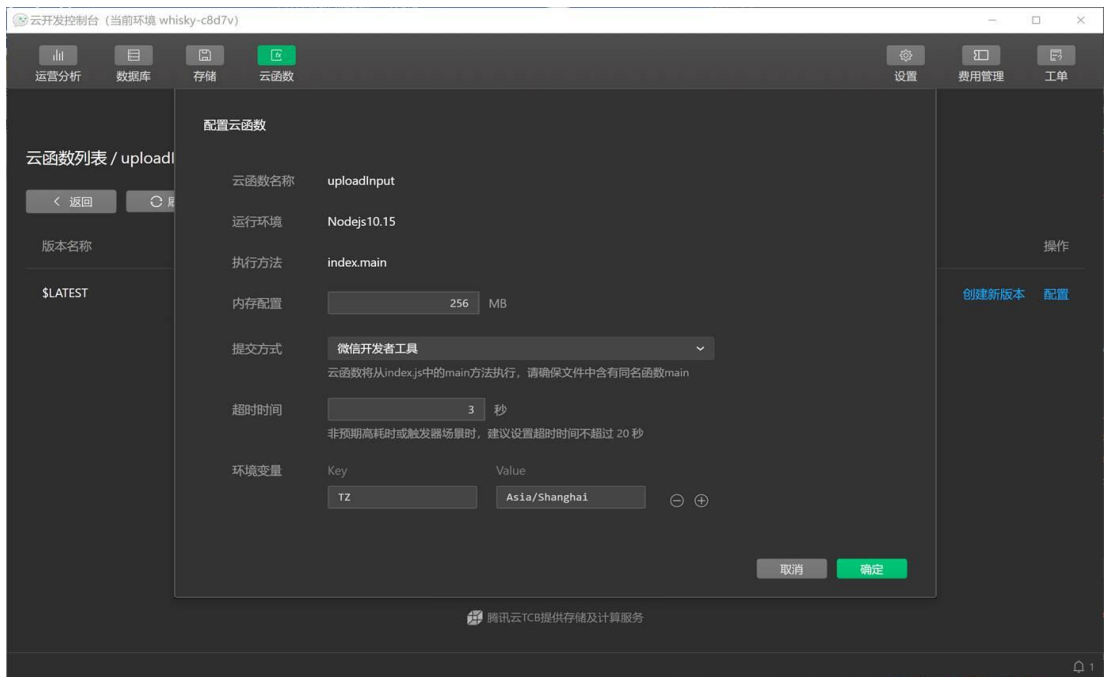
行 13, 列 7 空格: 2 UTF-8 LF JavaScript

```
miniprogram > components > infoCard > infoCard.wxml
<!-- components/infoCard/infoCard.wxml -->
<block wx:for="{{releaseInfo}}" wx:key="index" wx:for-item="releaseItem">
  <view class="infoCard">
    <view class="cardContent" id="text-expander" bindtap="textExpander">
      <!-- 点击展开 bindtap="textExpander" -->
      <view class="text-expander__content" {{textExpa == false ? 'close' : ''}}">
        {{releaseItem.releaseText}}
      </view>
      <view class="call">
        <text>\n</text>
        <view>{{releaseItem.releaseCall}}</view>
      </view>
    </view>
    <view class="{{releaseItem.releaseImg.length > 0 ? 'cardImgs' : ''}}">
      <block wx:for="{{releaseItem.releaseImg}}" wx:key="index">
        <view class="imgItem" bindtap="previewImage" id="{{item}}">
          <image src="{{item}}" alt="" mode="scaleToFill" />
        </view>
      </block>
    </view>
    <view class="cardFoot">
      <view class="foot footAvatar">
        <image src="{{releaseItem.createBy.avatarUrl}}" alt="" />
      </view>
      <view class="foot footTime">{{releaseItem.createTime}}</view>
      <view class="foot footCall">
        <image src="{{callIcon}}" alt="" />
      </view>
    </view>
  </view>
</block>
```

行 21, 列 28 空格: 2 UTF-8 LF XML

(4) 以上仅为部分代码碎片展示，详见完整源码。

(5) 注意：鉴于微信小程序的云服务器的时间是 UTC 时间，需要在云开发控制台中对云函数 uploadInput 进行配置, 修改环境变量为 TZ Asia/Shanghai, 将时间修改为 UTC+8。



(4) 匿名聊天

功能：

- 1) 进入聊天页面开始随机匹配在房间的用户
- 2) 双方匹配之后进行匿名聊天
- 3) 聊天过程中收发的数据包仅包含了聊天内容 经过服务器作为第三方周转 无法获取对方的信息




使用技术：

HTTPS 加密传输（测试时用了本地 HTTP 服务器）

数据库储存

Django 框架

代码展示：



```
son love-into.wxml index.js - wait-uploadinput message.wxml index.js - chat X messa
14 var httpConnector = {
15   url: 'http://101.132.120.92:8085/', // for test
16   res_data: '',
17   GET: function(url_last){
18     var that = this;
19     wx.request({
20       url: this.url + url_last,
21       header: {
22         'content-type': 'application/json'
23       },
24       success (res) {
25         that.res_data = res.data.toString();
26         //console.log("GET: " + that.res_data);
27       }
28     })
29   },
30 },
31 POST: function(url_last, info){
32   var that = this;
33   wx.request({
34     url: this.url + url_last,
35     data: info,
36     method: "POST",
37     header: {
38       'content-type': 'application/x-www-form-urlencoded'
39     },
40     success (res) {
41       that.res_data = res.data.toString();
42       //console.log("POST: " + that.res_data);
```

```

73  onLoad: function () {
74      var that = this;
75      wx.getUserInfo({
76          success: res => {
77              user_id = res.userInfo
78              console.log(user_id);
79          }
80      })
81  })
82  console.log(user_id);
83  this.setData({
84      message: '',
85      text: "",
86      has_find: false
87  })
88  httpConnector.GET("test")
89  setTimeout(function () {
90      if (httpConnector.res_data != "Success"){
91          that.setData({
92              hidden: true,
93              text: "网络异常"
94          })
95          console.error("网络异常")
96      }
97      else{
98          var addUserInterval = setInterval(function () {
99              httpConnector.POST("add", {"user": user_id})
100              if (httpConnector.res_data == "0"){
101                  that.setData({
102                      hidden: true

```

ove-info.wxml	index.js - wall-uploainput	message.wxml	index.wxml
<pre> 1 <!-- written by 陆张驰 --> 2 3 <view class="historycon" hidden="{{!hidden}}" > 4 <scroll-view scroll-y="true" class="history"> 5 <text>{{text}}</text> 6 </scroll-view> 7 </view> 8 <view> 9 <loading hidden="{{hidden}}"> 10 正在随机挑选一位陌生人..... 11 </loading> 12 </view> 13 <view> 14 <view class="sendmessage" hidden="{{!hidden}}"> 15 <input id="message" value="{{message}}" 16 bindinput="getInput"></input> 17 <button bindtap="add">发送</button> 18 </view> 19 </view> </pre>			

注意：本匿名聊天系统需要提前开启服务器（未开启服务器，可能出现“网络异常”现象），服务器代码见源码。