

DBMS 期末大作业报告

【组员】

梁玮瀚 17341093 人工智能与大数据方向
廖浩淳 17341096 人工智能与大数据方向
刘秉杰 17341104 超算方向

【项目名称】

头头难（微信小程序），原名“滴滴打球”，因商标问题被限制注册

【功能介绍】

主要功能为线上约球。系统会在用户登录时获取用户的昵称、头像 url、性别、位置等信息，并需要用户填写身高、体重、位置等个人信息。用户可以在“安排”页面点击“匹配野人”按钮寻找球友，等到另外一名用户点击此按钮，就将他们匹配起来，同时两位用户的界面上会弹出窗口显示对方的信息。

由于当前用户数目太少，无法个性化匹配球友。为演示该功能，我们爬取了虎扑的 NBA 球员数据，其中包括名字、头像 url、身高、体重、场均得分、投篮命中率（总体、三分、罚球）、球衣号码等。用户可以在“安排”页面点击“匹配球人”按钮，然后在随后弹出的页面里“定制”理想的对手，对此我们对用户输入做了检查，避免了非法的输入，当数据库中找到符合要求的球员时会用弹窗的形式把该球员显示给用户，对于没有无人满足条件的情况也会有提示。

对于数据库中存放的 NBA 球员数据，我们根据身高、体重、命中率等数据分别做了排行榜，用户可在“安排”页面点击“排行榜”按钮进行查看。

我们还对 UI 做了一些美化。在“首页”中我们放置了一些滚动的篮球美图、球队 logo 以及热门球员的预览信息。在“个人”页面中用户可以点击自己的头像或“我的名片”查看或修改个人信息；点击“主题切换”可以改变背景颜色。

【数据库设计】

我们使用的是微信开发者工具自带的云数据库，在微信公众平台可以查到相关接口，在微信开发者工具中点击“云开发”可以检查数据库内容

微信云数据库提供了索引管理，可以提高查找效率。

在程序中我们规定每个用户只能更新或增加与自己有关的记录，即限制在自己的 `_openid` 下。

- `login_users` 保存所有用户的个人信息，包括 `_id`（微信云数据库为每一条数据分配的 id）、`_openid`（微信小程序中每个用户的专有 id）、`avatarUrl`、`city`、`province`、`country`、`gender`、`height`、`weight`、`name`、`nickname`、`num`、

pos、session_key。

- players 保存爬取到的 NBA 球员数据, 包括_id、name、photo、num、height、weight、pts、shooting、threepts、freethrows。
- wait 保存匹配队列中的用户的_openid
- baller_games, 储存“匹配球人”的匹配记录, 包括_id、_openid、baller (球员姓名)、user (用户昵称)。

【前端设计】

共有 8 个 pages: login 为登录页面, index 为首页, match 为匹配页面, me 为点击“匹配球人”后弹出的页面, rank 为排行榜, wkiwi-user 为个人页, info 为点击“我的名片”弹出的个人信息, skin-change 为主题切换。

1. 登录

App.vue 中的 onLaunch 函数会检查用户是否登录, 没有登录就会跳转到 login 页面, 按下登录按钮后进行登录, 把用户的个人信息写到 login_user 中, 同时将其中一些信息放到 storage。

- 代码:

<App.vue>

```
<script>
export default {
  // #ifdef MP
  onLaunch: function() {
    wx.cloud.init({
      traceUser: true,
      env: 'liaohch3-j5hup'
    });
    var login_code = '';

    wx.checkSession({
      success () {
        wx.login({
          success (res) {
            if (res.code) {
              login_code = res.code;
            } else {
              console.log('登录失败!' + res.errMsg)
            }
          }
        })
      })
    uni.getUserInfo({
      success: function(userRes) {
        var userInfo = userRes.userInfo;
        console.log('userInfo', userInfo);
        console.log('login_code', login_code);

        uni.request({
          url: 'https://api.weixin.qq.com/sns/jscode2session?appid=wx708106e4bcd4934&secret=c95edcc7d59fbeff19fc6261af70dd2c&js_code=' + login_code + '&grant_type=authorization_code',
          success: requestRes => {
            console.log('requestRes', requestRes);
            var openid = requestRes.data.openid;
            var session_key = requestRes.data.session_key;

            // #ifdef MP-WEIXIN
            wx.cloud.callFunction({
              name: 'updateData',
              env: 'liaohch3-j5hup',
              data: {
                nickName: userInfo.nickName,
                avatarUrl: userInfo.avatarUrl,
                gender: userInfo.gender,
                province: userInfo.province,
                city: userInfo.city,
                country: userInfo.country
              }
            }).catch(err => {
              // handle error
              console.log('myerr', err);
            })
            console.log('数据库更新');
```

```

// #endif MP-WEIXIN -->
    console.log('opn', openid);

    uni.setStorageSync('nickName', userInfo.nickName);
    uni.setStorageSync('avatarUrl', userInfo.avatarUrl);
    uni.setStorageSync('gender', userInfo.gender);
    uni.setStorageSync('province', userInfo.province);
    uni.setStorageSync('city', userInfo.city);
    uni.setStorageSync('country', userInfo.country);
    uni.setStorageSync('current_skin', "#4191ea");
  }
})
})
},
fail () {
  // session key 已经失效，需要重新执行登录流程
  uni.showModal({
    title: '提醒',
    content: "请登录",
    success:function(){
      uni.navigateTo({
        url:"/pages/login/login"
      });
    },
  });
});
})
}
// #endif
</script>

```

<login.vue>

```

<template>
<!-- #ifdef MP-WEIXIN -->
  <view class="header">
    <image src="../../static/icon/wx.png"></image>
  </view>

  <view class="content">
    <view>申请获取以下权限</view>
    <text>获得你的公开信息(昵称, 头像等)</text>
  </view>

  <!-- <button class='bottom' type='primary' open-type='getUserInfo' lang='zh_CN' bindgetuserinfo='bindGetUserInfo'>
    授权登录
  </button -->
  <button type="primary" open-type="getUserInfo" @getUserInfo="getUserInfo" withCredentials="true">微信登陆</button>

  <!-- #endif -->
</view>
</template>

```

```

<script>
export default {
  data() {
    return {
      // #ifdef MP-WEIXIN
      // canIUse: wx.canIUse('button.open-type.getUserInfo'),
      canIUse: true,
      isHide: false,
      nickName: '',
      openid: '',
      session_key: '',
      avatarUrl: '',
      // #endif
    },
  },
  methods: {
    getUserInfo:function(res){
      var me = this;
      console.log('res', res);

      var userInfo = res.detail.userInfo;
      var nickName = userInfo.nickName;
      var avatarUrl = userInfo.avatarUrl;
      var gender = userInfo.gender; //性别 0: 未知, 1: 男, 2: 女
      var province = userInfo.province;
      var city = userInfo.city;
      var country = userInfo.country;
      if(!res.detail.iv){
        uni.showToast({
          title: "您取消了授权，登陆失败",
          icon: "none"
        });
        return false;
      }
    }
  }
}

```

```

uni.login({
  provider: 'weixin',
  success: function(loginRes) {
    // console.log(loginRes);
    uni.request({
      url: 'https://api.weixin.qq.com/sns/jscode2session?appid=wx708106e4bcd4934&secret=c95edcc7d59fbee1f19fc6261af70dd2c&js_code=' + loginRes.code + '&grant_type=authorization_code',

      success: res => {
        // 获取到用户的 openid
        me.openid = res.data.openid;
        me.session_key = res.data.session_key;
        const db = wx.cloud.database();
        const projectCollection = db.collection('login_users');
        projectCollection.add({
          data: {
            openid: me.openid,
            session_key: me.session_key,
            nickName: userInfo.nickName,
            avatarUrl: userInfo.avatarUrl,
            gender: userInfo.gender,
            province: userInfo.province,
            city: userInfo.city,
            country: userInfo.country
          }
        });
        success: res => {
          console.log(res);
        },
      },

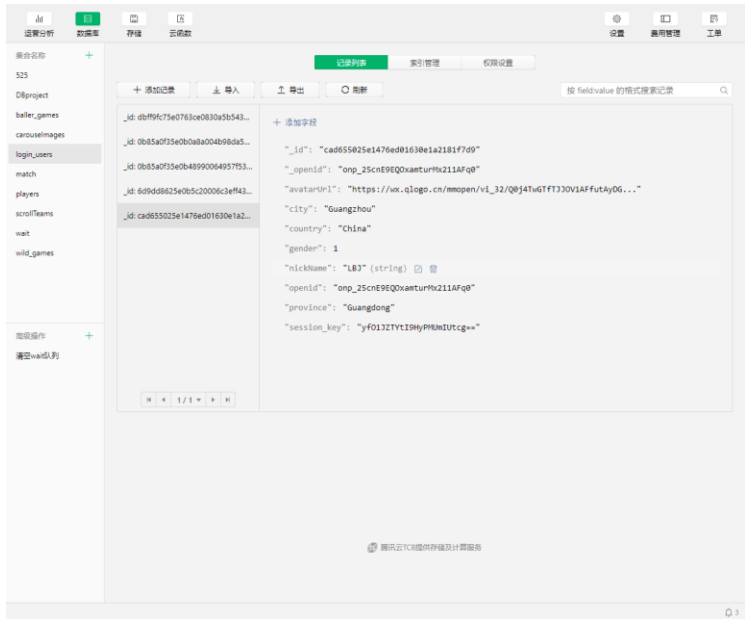
      uni.setStorageSync('suid', me.openid);
      uni.setStorageSync('srand', me.session_key);
      uni.setStorageSync('nickName', userInfo.nickName);
      uni.setStorageSync('avatarUrl', userInfo.avatarUrl);
      uni.setStorageSync('gender', userInfo.gender);
      uni.setStorageSync('province', userInfo.province);
      uni.setStorageSync('city', userInfo.city);
      uni.setStorageSync('country', userInfo.country);
    })
  },
  fail: function(r) {
    console.log(r);
  }
});
uni.navigateBack({
});
}
}
}
</script>

```

- 演示:



登录时



登录后的数据库和缓存

Key	Value	Type
__DC_STAT_UUID	"15783993821581120075"	String
suid	"onp_25cnE9EQxamturhx211AFq8"	String
srand	"yf0132TYtI9HyPHUmIutcg=="	String
nickname	"LB3"	String
avatarUrl	"https://wx.qlogo.cn/mmopen/vi_32/Q8j4TwGTFt33OV1AFfutAyDGVK18Bu60ok1lhPZnxbdIg7kz1Qq5rP9087YE01bkRT... <string is too large>"	String
gender	1	Number
province	"Guangdong"	String
city	"Guangzhou"	String
country	"China"	String

2. 首页

放了一些滚动的 CBA 赛场图片，CBA 球队 logo，以及 NBA 热门球员，NBA 球员数据是从虎扑爬取的。显示图片都是调用的组件。图片为新浪体育爬取到的 url。

- 代码：

<index.vue>

```

<template>
  <view class="page">

    <!-- 轮播图 -->
    <swiper :indicator-dots="true" :autoplay="true" :interval="3000" :duration="1000" class="carousel">
      <swiper-item v-for="(car, carIndex) in carouselList" :key="carIndex">
        <image :src="car.url" class="carousel"></image>
      </swiper-item>
    </swiper>

    <!-- 热门球队 -->
    <view class="page-block super-hot">
      <view class="hot-title-wapper">
        <image src="../../static/icon/remen.png" class="hot-icon"></image>
        <view class="hot-title">
          热门球队
        </view>
      </view>
    </view>

    <scroll-view scroll-x="true" class="page-block hot">
      <view class="single-poster" v-for="(scr, scrIndex) in scrollList" :key="scrIndex">
        <view class="poster-wapper">
          <image :src="scr.url" class="poster"></image>
          <view class="team-name">
            {{scr.name}}
          </view>
        </view>
      </view>
    </scroll-view>

    <!-- 热门球员 -->
    <view class="page-block super-hot">
      <view class="hot-title-wapper">
        <image src="../../static/icon/remen.png" class="hot-icon"></image>
        <view class="hot-title">
          热门球员
        </view>
      </view>
    </view>

    <view v-for="(player, playerIndex) in playersList" :key="playerIndex">
      <view class="" v-if="playerIndex < 10">
        <van-card
          :desc1="player.desc1"
          :desc2="player.desc2"
          :desc3="player.desc3"
          :desc4="player.desc4"
          :desc5="player.desc5"
          :title="player.name"
          :thumb="player.photo"
        >
      </view>
    </view>

  </view>
</template>

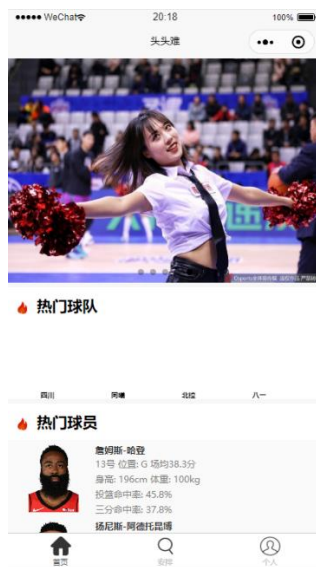
<script>
  import common from "../../common/common.js";
  var serverUrl = common.serverUrl;

  export default {
    data() {
      return {
        carouselList: [],
        scrollList: [],
        playersList: [],
        imageUrl: "https://ws1.sinaimg.cn/large/007RvR0Hgy1ga934esikg13046046dft.jpg",
        imageUrl: "https://img.cdn.aliyun.dcloud.net.cn/stream/plugin_screens/c0389120-c415-11e9-9f1e-d39825801d33_0.png?v=1576513486"
      }
    },
    onLoad() {
      var me = this;

      const db = wx.cloud.database();
      db.collection('carouseImages').get().then(res => {
        me.carouselList = res.data[0].data;
      })
      db.collection('scrollTeams').get().then(res => {
        me.scrollList = res.data[0].data;
      })
      db.collection('players').get().then(res => {
        me.playersList = res.data;
        for (var i = 0; i < me.playersList.length; i++) {
          var player = me.playersList[i];
          me.playersList[i].desc1 = player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分";
          me.playersList[i].desc2 = "身高: " + player.height + "cm 体重: " + player.weight + "kg";
          me.playersList[i].desc3 = "投篮命中率: " + player.shooting + "%";
          me.playersList[i].desc4 = "三分命中率: " + player.threeps + "%";
          me.playersList[i].desc5 = "罚球命中率: " + player.freethrows + "%";
          // + "投篮命中率: " + player.shooting + " 三分命中率: " + player.threeps + " 罚球命中率: " + player.freethrows
        }
      })
    },
    methods: {
    }
  }
</script>

```

- 演示:



3. 个人页面

这里的界面是套用的网上的模板，目前只做了“我的名片”和“主题切换”两个功能。点击后跳转到相应的二级页面。

- 代码:

<info.vue>

```
<template>
  <view>
    <view class="header">
      <view class="bg" :style="{ 'background-color': current_skin }">
        <!--
        <view :class="{ 'background-color': current_skin, width: 100%, height: 200upx, padding-top: 100upx }"> -->
      </view>
      <view class="box">
        <view class="box-hd">
          <view class="avator" @click="clickToInfo">
            
          </view>
          <view class="phone-number">{{nickname}}</view>
        </view>
        <view class="box-bd">
          <view class="item">
            <view class="icon"></view>
            <view class="text">我的通知</view>
          </view>
          <view class="item">
            <view class="icon"></view>
            <view class="text">我的收藏</view>
          </view>
          <view class="item">
            <view class="icon"></view>
            <view class="text">我的客服</view>
          </view>
        </view>
      </view>
    </view>
  </view>
</template>
```

```

<view class="list-content">
  <view class="list">
    <view class="li noborder" @click="ClickToInfo" >
      <view class="icon"><image src="../../static/user/card.png"></image></view>
      <view class="text">我的名片</view>
      <image class="to" src="../../static/user/to.png"></image>
    </view>
  </view>
  <view class="list">
    <view class="li " @click="changeSkin">
      <view class="icon"><image src="../../static/user/skin.png"></image></view>
      <view class="text">主题切换</view>
      <image class="to" src="../../static/user/to.png"></image>
    </view>
    <view class="li " >
      <view class="icon"><image src="../../static/user/help.png"></image></view>
      <view class="text">帮助中心</view>
      <image class="to" src="../../static/user/to.png"></image>
    </view>
    <view class="li " >
      <view class="icon"><image src="../../static/user/about.png"></image></view>
      <view class="text">关于我们</view>
      <image class="to" src="../../static/user/to.png"></image>
    </view>
    <view class="li " >
      <view class="icon"><image src="../../static/user/opinion.png"></image></view>
      <view class="text">意见反馈</view>
      <image class="to" src="../../static/user/to.png"></image>
    </view>
  </view>
  <view class="list">
    <view class="li noborder" >
      <view class="icon"><image src="../../static/user/set.png"></image></view>
      <view class="text">系统设置</view>
      <image class="to" src="../../static/user/to.png"></image>
    </view>
  </view>
</view>
</template>

<script>
  export default {
    data() {
      return {
        nickName: '',
        openid: '',
        session_key: '',
        avatarUrl: 'https://ws1.sinaimg.cn/large/007RvR0Hly1gack42dh16j307g0b4q6o.jpg',
        current_skin: '#4191ea',
      };
    },
    onShow() {
      this.current_skin = uni.getStorageSync('current_skin');
      this.avatarUrl = uni.getStorageSync('avatarUrl');
      this.nickName = uni.getStorageSync('nickName');
    },
    mounted() {
      this.current_skin = uni.getStorageSync('current_skin');
      this.avatarUrl = uni.getStorageSync('avatarUrl');
      this.nickName = uni.getStorageSync('nickName');
    },
    onLoad() {
      this.current_skin = uni.getStorageSync('current_skin');
      this.avatarUrl = uni.getStorageSync('avatarUrl');
      this.nickName = uni.getStorageSync('nickName');
    },
    methods: {
      changeSkin(){
        uni.navigateTo({
          url: '../skin-change/skin-change'
        });
      },
      ClickToInfo(){
        uni.navigateTo({
          url: '../info/info'
        });
      }
    }
  }
</script>

```

- 演示:



4. 主题切换页面

点击“主题切换”按钮，会跳转到 skin-change 页面。选择相应的颜色就能改变个人页面的主题色，同时将颜色保存到 storage。使用 radio-group 组件，绑定@change=”radioChange”。

- 代码：

<skin-change.vue>

```
<template>
  <view>
    <view class="list-content">
      <radio-group @change="radioChange">
        <label class="list" v-for="(item, index) in items" :key="item.value">
          <view class="text">{{item.name}}</view>
          <view>
            <radio :value="item.value" :checked="index === current" />
          </view>
        </label>
      </radio-group>
    </view>
  </view>
</template>
```

```

<script>
export default {
  data() {
    return {
      title: 'radio',
      items: [
        {
          value: '#F0E68C',
          name: '卡其布'
        },
        {
          value: '#0A70D6',
          name: '紫罗兰'
        },
        {
          value: '#4191ea',
          name: '活力蓝',
          checked: 'true'
        },
        {
          value: '#2E8B57',
          name: '中大绿'
        }
      ],
      current: 0,
      current_skin: '#4191ea',
    },
    onLoad() {
      this.current = uni.getStorageSync('current');
      this.current_skin = uni.setStorageSync('current_skin', evt.target.value);
    },
    methods: {
      radioChange(evt) {
        for (let i = 0; i < this.items.length; i++) {
          if (this.items[i].value === evt.target.value) {
            this.current = i;
            uni.setStorageSync('current', i);
            uni.setStorageSync('current_skin', evt.target.value);
            break;
          }
        }
      }
    }
  }
}
</script>

```

- 演示:



5. 名片页面

在“个人”点击我的名片按钮或最上方的头像，可以跳转到 info 页面。上面几行是系统在用户设备上读取的信息，不可修改。下面几行：姓名、身高、体重、位置等需要用户自行填写，按下“确认修改”即可在数据库中更新。按下返回则不保存修改，下次打开仍然是之前的数据。

此处涉及到云函数的概念，微信云服务器的某些操作需要在云端定义函数，然后在前端使用 `wx.cloud.callFunction` 调用云函数。

- 代码：

<wiki-users.vue>

```
<template>
  <view>
    <cmd-page-body type="top">
      <cmd-transition name="fade-up">
        <view>
          <cmd-cel-item title="头像" slot="right">
            <cmd-avatar :src="avatarurl"></cmd-avatar>
          </cmd-cel-item>
          <!-- <cmd-cel-item title="积分" addon="566" arrow></cmd-cel-item -->
          <cmd-cel-item title="昵称" :addon="nickname" ></cmd-cel-item>
          <cmd-cel-item title="性别" :addon="genders[gender]" ></cmd-cel-item>
          <cmd-cel-item title="地址" :addon="address" ></cmd-cel-item>
          <van-field required label="姓名" :value="temp_name" placeholder="请输入姓名" @change="changeName" border="false"></van-field>
          <van-field required label="身高" :value="temp_height" placeholder="请输入身高(cm)" @change="changeHeight" border="false"></van-field>
          <van-field required label="体重" :value="temp_weight" placeholder="请输入体重(kg)" @change="changeWeight" border="false"></van-field>
          <van-field required label="位置" :value="temp_pos" placeholder="请输入位置" @focus="showPopup"></van-field>
          <van-popup :show="isShow" custom-style="height: 40%; closeable close-icon="close" position="bottom" @close="onClose">
            <van-picker show-toolbar title="位置" :columns="positions" @cancel="onClose" @confirm="changePos"></van-picker>
          </van-popup>
          <van-field label="号码" :value="temp_num" placeholder="请输入号码" @change="changeNum" border="false"></van-field>
          <van-popup :show="isInvalid" @close="onClose">请输入有效信息</van-popup>
          <!-- <cmd-cel-item title="联系方式" addon="15676189581" arrow></cmd-cel-item>
          <cmd-cel-item title="证件号码" addon="45011200000002817" arrow></cmd-cel-item>
          <cmd-cel-item title="初始地址" addon="广西壮族自治区南宁市西乡塘区大学西路29号" arrow></cmd-cel-item>
          <cmd-cel-item title="修改密码" @click="fnClick('modify')">修改</cmd-cel-item -->
          <van-button size="large" @click="updata">确认修改</van-button>
          <button class="btn-logout">退出登录</button>
        </view>
      </cmd-transition>
    </cmd-page-body>

    <!-- <view class="">
      name: {{name}}
    </view -->

  </view>
</template>
```

```
export default {
  components: {
    cmdNavBar,
    cmdPageBody,
    cmdTransition,
    cmdCelItem,
    cmdAvatar
  },

  data() {
    console.log('data');
    return {
      checked: true,
      name: '',
      height: '',
      weight: '',
      pos: '',
      num: '',
      isShow: false,
      isInvalid: false,
      positions: ['PG', 'SG', 'SF', 'PF', 'C'],
      genders: ['未知', '男', '女'],
      temp_name: '',
      temp_height: '',
      temp_weight: '',
      temp_pos: '',
      temp_num: '',
      openid: uni.getStorageSync('suid'),
      nickname: uni.getStorageSync('nickName'),
      avatarurl: uni.getStorageSync('avatarurl'),
      country: uni.getStorageSync('country'),
      province: uni.getStorageSync('province'),
      city: uni.getStorageSync('city'),
      gender: uni.getStorageSync('gender'),
      session_key: uni.getStorageSync('srand'),
      address: ''
    }
  },
}
```

```

onLoad() {
  console.log('onload');
  const db = wx.cloud.database();

  var me = this; console.log('me.openid', me.openid);
  this.address = this.country+" "+this.province+" "+this.ci

  db.collection('login_users').where({
    _openid: me.openid,
  })
  .get().then(res=>{
    console.log(res);
    me.name = res.data[0].name;
    me.height = res.data[0].height;
    me.weight = res.data[0].weight;
    me.pos = res.data[0].pos;
    me.num = res.data[0].num;
    me.temp_name = me.name;
    me.temp_height = me.height;
    me.temp_weight = me.weight;
    me.temp_pos = me.pos;
    // this.temp_pos = '';
    me.temp_num = me.num;
  })
  .catch(err=>{
    console.log(err);
  });

  console.log('pos', this.pos);
  console.log('temp_pos', this.temp_pos);
},

```

```

methods: {
  /**
   * 点击触发
   * @param {Object} type 跳转页面或设置类型方式
   */
  updateData:function() {
    var me = this;
    if (me.temp_pos==' ' || isNaN(parseInt(me.temp_height)) || isNaN(parseInt(me.temp_weight)) || parseInt(me.temp_height) < 100 || parseInt(me.temp_height) > 300 || parseInt(me.temp_weight) < 35 || parseInt(me.temp_weight) > 200) {
      me.isInvalid = true;
      console.log("INVALID");
      return 0;
    }
    this.name = this.temp_name;
    this.height = this.temp_height;
    this.weight = this.temp_weight;
    this.pos = this.temp_pos;
    this.num = this.temp_num;
    console.log('pos', this.pos);
    console.log('temp_pos', this.temp_pos);
    if (me.num=='') {
      me.num=='0'
    }

    console.log('me.nickname', me.nickname);
    console.log('me.openid', me.openid);
    console.log('me.session_key', me.session_key);
    console.log('me.height', parseInt(me.height));
    // console.log(this);
  }
}

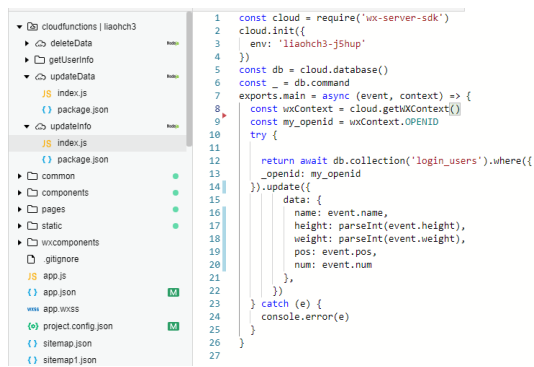
```

```

// #ifdef MP-WEIXIN
wx.cloud.callFunction({
  name: 'updateInfo',
  env: 'liaohch3-jshup',
  data: {
    name: me.name,
    height: parseInt(me.height),
    weight: parseInt(me.weight),
    pos: me.pos,
    num: me.num
  }
}).catch(err => {
  // handle error
  console.log('myerr', err);
})
console.log('数据库更新');
// #endif

success:res => {
  console.log(res);
  console.log('-----');
};
fail:res =>{
  console.log("失败", res);
};
console.log("++++++++++++++++++++");
},

```



云函数可在微信开发者工具查看



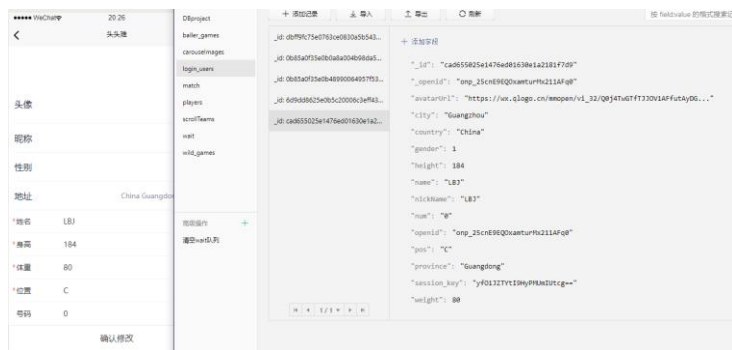
其他函数

- 演示 (修改):



原本位置为 SF，将其改为 C:





数据库中也就对应更新了

6. 安排页面

有三个按钮：

1. “匹配野人”可以在线匹配球友，当有一个用户点击按钮时就会将自己的信息放入 `wait` 表中，之后等待并持续监测 `wait` 中有没有新的记录，当第二个用户点击时就将这两个用户匹配起来，同时在两位用户的界面上弹窗提示；
2. “匹配球人”可以使用爬取到的 **NBA** 球员进行测试（因为当前用户太少，不方便演示“匹配野人”的功能），点击按钮后跳转到二级页面，在里面进行一些偏好设置即可；
3. “排行榜”可以显示 **NBA** 球员的数据排行（也是为了演示对数据库的访问和操作），点击后进入一个二级页面，页面上会显示各种排行榜。

- 代码：

<match. vue>

```
<template>
  <view>
    <view>
      <van-button type="primary" color="linear-gradient(to right, #4bb0ff, #6149f6)" block @click="matchBaller">匹配球人</van-button>
      <van-button type="primary" block @click="matchWild">匹配野人</van-button>
      <van-button type="primary" block @click="rank">排行榜</van-button>
    </view>
    <yangr-msg v-if="Show1" :title="title" type="success" :info="info1" btn="安排" @yangrMsgEvent="closeYangrMsg"></yangr-msg>
    <yangr-msg v-if="Show2" :title="title" type="success" :info="info2" btn="安排" @yangrMsgEvent="closeYangrMsg"></yangr-msg>
  </view>
</template>

<script>
  import yangrMsg from "@/components/yangr-msg/yangr-msg.vue"
  export default {
    components: { yangrMsg },
    data() {
      return {
        title: '匹配成功',
        info1: '',
        info2: '',
        Show1: false,
        Show2: false,
      }
    },
    methods: {
      closeYangrMsg() {
        this.Show1 = false;
        this.Show2 = false;
      },
      rank: function(event) {
        uni.navigateTo({
          url: '../rank/rank'
        })
      }
    }
  }
}
```

```

// #ifdef MP
matchWill: function (event) {
  var me = this;
  const db = wx.cloud.database();
  const waitCollection = db.collection('wait');
  var total = 0;

  waitCollection.count({
    success: function(res) {
      console.log(res.total);
      total = res.total;
      var watcher;
      if(total == 0){
        waitCollection.add({data: {}})
        watcher = waitCollection.watch({
          onChange: function(snapshot) {
            console.log('snapshot1', snapshot)
            if(snapshot.docs.length == 2){
              console.log('1号的对手', snapshot.docs[1]._openid)
              var his_nickname = '';
              db.collection('login_users').where({
                _openid: snapshot.docs[1]._openid
              }).get().then(get_res => {
                console.log('>', get_res.data[0].nickname)
                his_nickname = get_res.data[0].nickname;
                me.info1 = '对手: ' + his_nickname;
                me.Show1 = true;
                // #ifdef MP-WEIXIN
                wx.cloud.callFunction({
                  name: 'deleteData',
                  env: 'liaoohch3-jshup',
                  data: {}
                })
                // #endif
              })
            }
          },
          onError: function(err) {
            console.error('the watch closed because of error', err)
          }
        })
      }
    },
    onError: function(err) {
      console.error('the watch closed because of error', err)
    }
  })
}
}

```

```

} else if(total == 1){
  waitCollection.get()
  .then(res => {
    var his_nickname = '';
    db.collection('login_users').where({
      _openid: res.data[0]._openid
    }).get().then(get_res => {
      console.log('>', get_res.data[0].nickname)
      his_nickname = get_res.data[0].nickname;
      me.info2 = '对手: ' + his_nickname;
      me.Show2 = true;
    })
  })
}

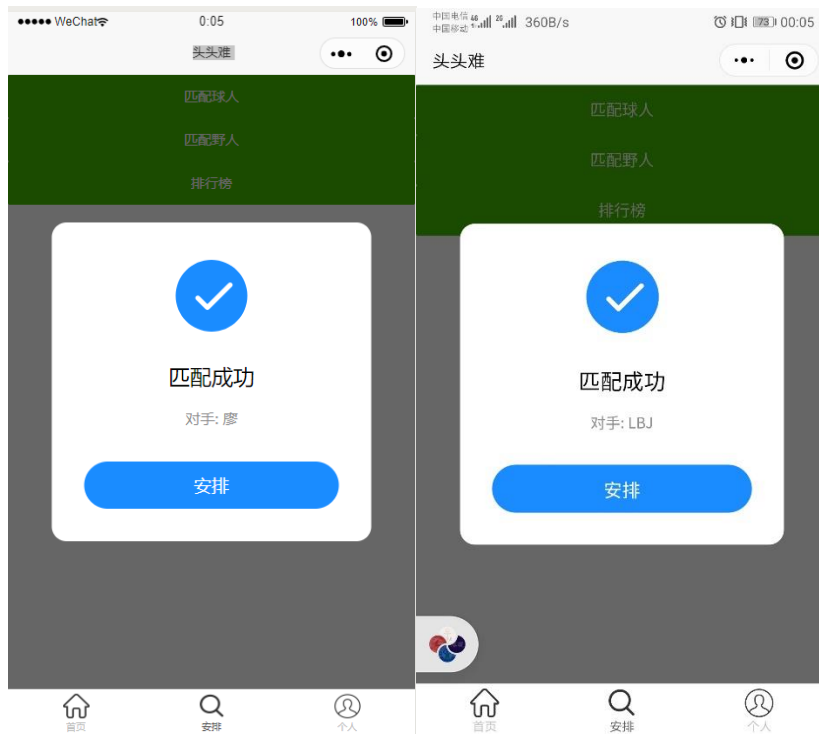
waitCollection.add({data: {}})
watcher.close()
// #ifdef MP-WEIXIN
wx.cloud.callFunction({
  name: 'deleteData',
  env: 'liaoohch3-jshup',
  data: {}
})
// #endif
},
fail: console.error
})

// #endif

// #ifdef MP
matchBaller: function () {
  uni.navigateTo({
    url: '../me/me'
  });
}
// #endif
}
}
</script>

```

- 演示:



7. “匹配球人”页面

点击匹配球人，跳转到 me 页面。用户通过选择器组件选择希望的身高、体重区间以及位置，然后点击“即可安排”按钮，如果找到就弹出成功字样，没找到也会弹窗提示。当用户没有输入完整信息就点击按钮时，也会弹窗提醒用户补全信息。

- 代码:

<me.vue>

```
<template>
  <view class="">

    <!-- #ifdef MP -->

    <van-cell-group>
      请描述您的理想型
      <!-- <van-field required label="姓名" :value="name" placeholder="请输入姓名" @change="changeName" border="false"></van-field> -->
      <van-field required label="身高" :value="height" placeholder="请选择身高" @focus="showPopup1"></van-field>
      <van-popup show="isShow1" custom-style="height: 40%" closeable close-icon="close" position="bottom" @close="onClose1">
        <van-picker show-toolbar title="身高" :columns="hts" @cancel="onClose1" @confirm="changeHeight"></van-picker>
      </van-popup>
      <van-field required label="体重" :value="weight" placeholder="请选择体重" @focus="showPopup2"></van-field>
      <van-popup show="isShow2" custom-style="height: 40%" closeable close-icon="close" position="bottom" @close="onClose2">
        <van-picker show-toolbar title="体重" :columns="wts" @cancel="onClose2" @confirm="changeWeight"></van-picker>
      </van-popup>
      <van-field required label="位置" :value="pos" placeholder="请选择位置" @focus="showPopup3"></van-field>
      <van-popup show="isShow3" custom-style="height: 40%" closeable close-icon="close" position="bottom" @close="onClose3">
        <van-picker show-toolbar title="位置" :columns="positions" @cancel="onClose3" @confirm="changePos"></van-picker>
      </van-popup>
      <van-popup show="isIncomplete" @close="invClose">请输入完整信息</van-popup>
      <van-popup show="notFound" @close="invClose">未找到匹配项</van-popup>
    </van-cell-group>
    <van-cell></van-cell>
    <van-button type="danger" size="large" @click="matchBaller">即刻安排</van-button>
    <!-- #endif -->
    <yangr-msg v-if="isMatched" title="找到球人啦" type="success" :info="info" btn="安排" @yangrMsgEvent="closeYangrMsg"></yangr-msg>
  </view>
</template>
```



```

<script>
import yangrMsg from "@components/yangr-msg/yangr-msg.vue"
export default {
  components: {yangrMsg},
  data() {
    return {
      height: '',
      weight: '',
      upper_height: '',
      lower_height: '',
      upper_weight: '',
      lower_weight: '',
      pos: '',
      notFound: false,
      isIncomplete: false,
      isShow1: false,
      isShow2: false,
      isShow3: false,
      isMatched: false,
      hts: ['小于190','190-210','大于210'],
      wts: ['小于90','90-100','大于100'],
      lower_hts: [150,190,211],
      upper_hts: [109,210,250],
      lower_wts: [50.0,90.0,100.1],
      upper_wts: [89.9,100.0,100.0],
      positions: ['G','F','C'],

      nickName: uni.getStorageSync('nickName'),
      openId: '',
      session_key: '',

      info: '',
    }
  },
  methods: {
    closeYangrMsg(){
      this.isMatched = false;
      this.arrange();
    },
    arrange() {
      var me = this;
      const db = wx.cloud.database();
      db.collection('baller_games').add({
        data:{
          user: me.nickName,
          baller: me.info
        },
        success: function(res) {
          // res 是一个对象, 其中有 _id 字段标识刚创建的记录的 id
          console.log(res)
        }
      })
    }
  },
}

```

匹配成功时将这条记录写到数据库中

```

...
changeHeight:function(event){
  this.height = event.detail.value;
  this.upper_height = this.upper_hts[event.detail.index];
  this.lower_height = this.lower_hts[event.detail.index];
  console.log(event);
  console.log(this.height);
  console.log(this.upper_height);
  console.log(this.lower_height);
  // console.log(this.height);
  this.isShow1 = false;
},
changeWeight:function(event){
  this.weight = event.detail.value;
  this.upper_weight = this.upper_wts[event.detail.index];
  this.lower_weight = this.lower_wts[event.detail.index];
  console.log(event);
  // console.log(this.weight);
  this.isShow2 = false;
},
changePos:function(event){
  this.pos = event.detail.value;
  console.log(event);
  console.log(this.pos);
  this.isShow3 = false;
  // console.log(this.weight);
},
}

```

```

showPopup1:function() {
  this.isShow1 = true;
  console.log(this.isShow);
},
onClose1:function() {
  this.isShow1 = false;
  console.log(this.isShow);
},
showPopup2:function() {
  this.isShow2 = true;
  console.log(this.isShow);
},
onClose2:function() {
  this.isShow2 = false;
  console.log(this.isShow);
},
showPopup3:function() {
  this.isShow3 = true;
  console.log(this.isShow);
},
onClose3:function() {
  this.isShow3 = false;
  console.log(this.isShow);
},
invClose:function() {
  this.notFound = false;
  this.isIncomplete = false;
},
}

```

操作组件触发的函数

```

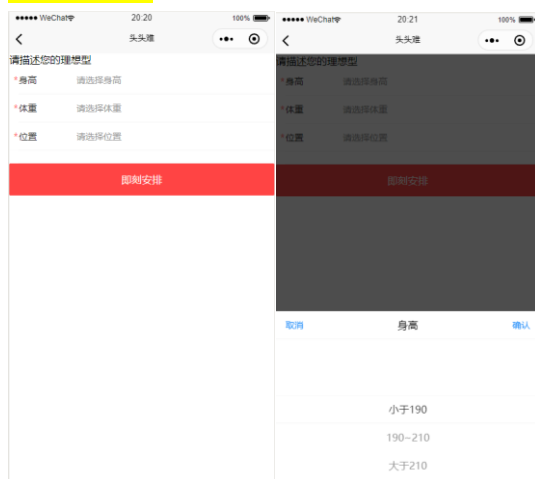
matchBaller:function() {
  var me = this;
  if(me.height!='' || me.weight!='' || me.pos!='') {
    me.isIncomplete = true;
    return 0;
  }
  const db = wx.cloud.database();
  const _ = db.command;

  db.collection('players').where({
    height: _.gte(me.lower_height).and(_.lte(me.upper_height)),
    weight: _.gte(me.lower_weight).and(_.lte(me.upper_weight)),
    pos: me.pos
  }).get({
    success: function(res) {
      // console.log(res);
      if(res.data.length==0) {
        me.notFound = true;
      }
      else {
        me.isMatched = true;
        console.log(res.data);
        me.info = res.data[0].name;
      }
    },
    fail: function() {
      me.notFound = true;
    }
  })
},
},
}
</script>

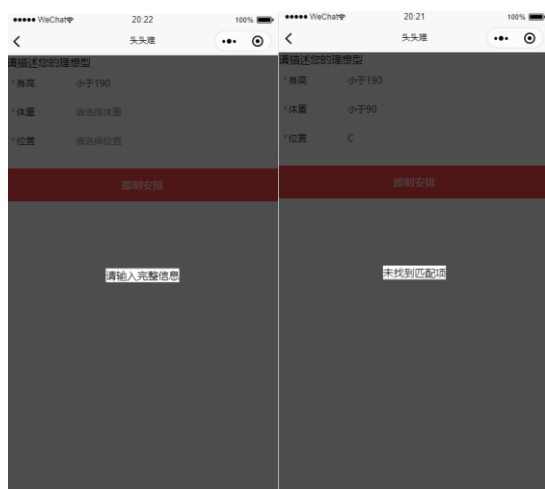
```

在 **player** 表中查询符合条件的球员，并返回所有结果中的第一个。

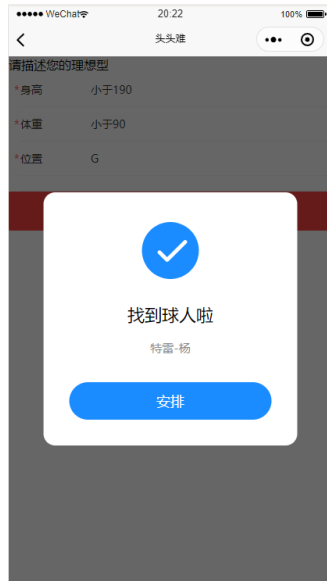
- 演示:



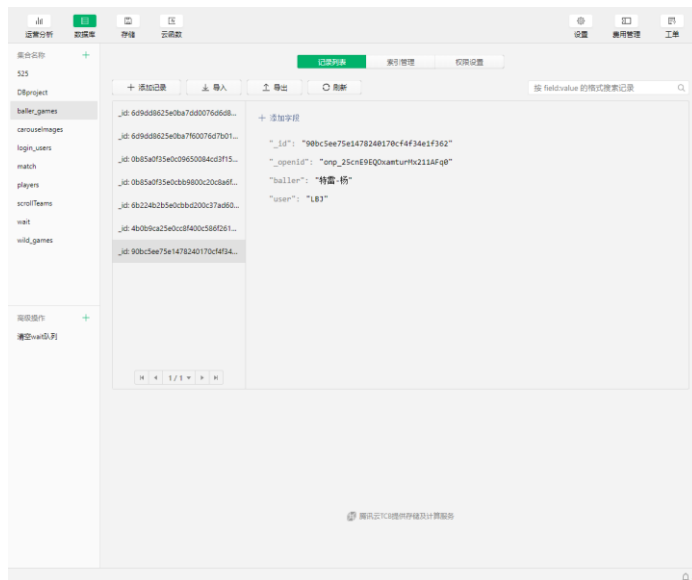
输入界面



错误提示



匹配成功



数据库中的记录。

8. “排行榜”页面

点击“排行榜”按钮，跳转到 `rank` 页面。在查询时使用 `orderBy` 对特定数据进行排序，并把查询结果放在只含有 8 条记录的“视图”中。

- 代码：

<rank.vue>

```

<template>
  <view class="list_box">
    <view class="left">
      <scroll-view scroll-y="true" :style="{ 'height':scrollHeight }">
        <view class="item"
          v-for="(item,index) in leftArray"
          :key="index"
          :class="{ 'active':index==leftIndex }"
          :data-index="index"
          @tap="leftTap"
        >{{item}}</view>
      </scroll-view>
    </view>
    <view class="main">
      <scroll-view scroll-y="true" :style="{ 'height':scrollHeight }" @scroll="mainScroll" :scroll-into-view="scrollInto" scroll-with-animation="true">
        <view class="item" v-for="(item,index) in mainArray" :key="index" :id="'item-' + index">
          <view class="title">
            <view>{{item.title}}</view>
          </view>
          <view class="goods" v-for="(item2,index2) in item.list" :key="index2">
            <image :src="item2.photo" mode=""></image>
            <view>
              <view>{{item2.name}}</view>
              <view class="describe">{{item2.desc1}}</view>
              <view class="money">{{item2.desc2}}</view>
            </view>
          </view>
        </scroll-view>
      </view>
    </template>
  </template>

```

```

<script>
export default {
  data() {
    return {
      scrollHeight: '500px',
      leftArray: [],
      mainArray: [],
      topArr: [],
      leftIndex: 0,
      scrollInto: ''
    }
  },
  onLoad() {
    uni.getSystemInfo({
      success: (res) => {
        /* 设置当前滚动容器的高，若非窗口的高度，请自行修改 */
        this.scrollHeight = `${res.windowHeight}px`;
      }
    });
  },
  mounted() {
    this.getListData();
  },
}

```

```

methods: {
  /* 获取列表数据 */
  getListData() {
    /* 若无真实数据，当前方法模拟数据 */
    const db = wx.cloud.database();
    let [left, main] = [[], []];
    let list = [];

    left.push('得分榜');
    let list1 = [];
    db.collection('players').orderBy('pts', 'desc').limit(8).get().then(res => {
      var temp = res.data;
      for (var i = 0; i < temp.length; i++) {
        var player = temp[i];
        temp[i].desc1 = player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分";
        temp[i].desc2 = "身高: " + player.height + "cm 体重: " + player.weight + "kg";
        temp[i].desc3 = "投篮命中率: " + player.shooting + "%";
        temp[i].desc4 = "三分命中率: " + player.threeps + "%";
        temp[i].desc5 = "罚球命中率: " + player.freethrows + "%";
        list1.push({
          name: player.name,
          photo: player.photo,
          desc1: player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分",
          desc2: "身高: " + player.height + "cm 体重: " + player.weight + "kg"
        });
      }
      // var list = list1;
      main.push({
        title: '得分榜',
        list: list1
      })
    })
    // list.splice(0, list.length);
    // console.log('list1', list);

    left.push('身高榜');
    let list2 = [];
    db.collection('players').orderBy('height', 'desc').limit(8).get().then(res => {
      var temp = res.data;
      // console.log('temp', temp);
      for (var i = 0; i < temp.length; i++) {
        var player = temp[i];
        temp[i].desc1 = player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分";
        temp[i].desc2 = "身高: " + player.height + "cm 体重: " + player.weight + "kg";
        temp[i].desc3 = "投篮命中率: " + player.shooting + "%";
        temp[i].desc4 = "三分命中率: " + player.threeps + "%";
        temp[i].desc5 = "罚球命中率: " + player.freethrows + "%";
        list2.push({
          name: player.name,
          photo: player.photo,
          desc1: player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分",
          desc2: "身高: " + player.height + "cm 体重: " + player.weight + "kg"
        });
      }
      // console.log('list2', list);
      main.push({
        title: '身高榜',
        list: list2
      })
    })
  }
}

```

```

left.push('体重榜');
let list3=[];
db.collection('players').orderBy('weight', 'desc').limit(8).get().then(res => {
    var temp = res.data;
    for (var i = 0; i < temp.length; i++) {
        var player = temp[i];
        temp[i].desc1 = player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分";
        temp[i].desc2 = "身高: " + player.height + "cm 体重: " + player.weight + "kg";
        temp[i].desc3 = "投篮命中率: " + player.shooting + "%";
        temp[i].desc4 = "三分命中率: " + player.threeps + "%";
        temp[i].desc5 = "罚球命中率: " + player.freethrows + "%";
        list3.push({
            name: player.name,
            photo: player.photo,
            desc1: player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分",
            desc2: "身高: " + player.height + "cm 体重: " + player.weight + "kg"
        });
    }
    main.push({
        title: '体重榜',
        list: list3
    })
})

```

```

left.push('罚球榜');
let list4=[];
db.collection('players').orderBy('freethrows', 'desc').limit(8).get().then(res => {
    var temp = res.data;
    for (var i = 0; i < temp.length; i++) {
        var player = temp[i];
        temp[i].desc1 = player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分";
        temp[i].desc2 = "身高: " + player.height + "cm 体重: " + player.weight + "kg";
        temp[i].desc3 = "投篮命中率: " + player.shooting + "%";
        temp[i].desc4 = "三分命中率: " + player.threeps + "%";
        temp[i].desc5 = "罚球命中率: " + player.freethrows + "%";
        list4.push({
            name: player.name,
            photo: player.photo,
            desc1: player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分",
            desc2: "身高: " + player.height + "cm 体重: " + player.weight + "kg"
        });
    }
    main.push({
        title: '罚球命中率榜',
        list: list4
    })
})

```

```

left.push('三分榜');
let lists=[];
db.collection('players').orderBy('threeps', 'desc').limit(8).get().then(res => {
    var temp = res.data;
    for (var i = 0; i < temp.length; i++) {
        var player = temp[i];
        temp[i].desc1 = player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分";
        temp[i].desc2 = "身高: " + player.height + "cm 体重: " + player.weight + "kg";
        temp[i].desc3 = "投篮命中率: " + player.shooting + "%";
        temp[i].desc4 = "三分命中率: " + player.threeps + "%";
        temp[i].desc5 = "罚球命中率: " + player.freethrows + "%";
        lists.push({
            name: player.name,
            photo: player.photo,
            desc1: player.num + "号 位置: " + player.pos + " 场均" + player.pts + "分",
            desc2: "身高: " + player.height + "cm 体重: " + player.weight + "kg"
        });
    }
    main.push({
        title: '三分命中率榜',
        list: lists
    })
})

```

```

        main.push({
            title: '命中率榜',
            list: list6
        })
    })
    this.leftArray=left;
    this.mainArray=main;

    this.$nextTick(()=>{
        this.getElementTop();
    });
},
/* 获取元素顶部信息 */
getElementTop(){
    /* Promise 对象数组 */
    let p_arr=[];

    /* 新建 Promise 方法 */
    let new_p=(selector)=>{
        return new Promise((resolve,reject)=>{
            let view = uni.createSelectorQuery().select(selector);
            view.boundingClientRect(data => {
                resolve(data.top);
            }).exec();
        })
    }

    /* 遍历数据, 创建相应的 Promise 数组数据 */
    this.mainArray.forEach((item,index)=>{
        p_arr.push(new_p("#item-${index}"));
    });

    /* 所有节点信息返回后调用该方法 */
    Promise.all(p_arr).then((data)=>{
        this.topArr=data;
    });
},

```

```

/* 主区域滚动监听 */
mainScroll(e){
  let top = e.detail.scrollTop;
  let index = 0;
  /* 查找当前滚动距离 */
  for(let i = (this.topArr.length-1); i >= 0; i--){
    /* 在部分安卓设备上，因手机逻辑分辨率与rpx单位计算不是整数，滚动距离与有误差，增加2px来完善该问题 */
    if((top+2) >= this.topArr[i]){
      index = i;
      break;
    }
  }
  this.leftIndex = (index < 0 ? 0 : index);
},
/* 左侧导航点击 */
leftTap(e){
  let index = e.currentTarget.dataset.index;
  this.scrollInto = `item-${index}`;
}
}
</script>

```

- 演示：



【心得体会】

在我们的合作下，总算是从前端到后台完整地做了一个小程序出来。

在做前端的过程中我们在 GitHub 找到了 vant-weapp 组件包，里面包含了一些好看又实用的组件，通过调用组件和根据我们自己的审美进行设计，小程序的界面还不至于太难看。前端设计的重点是页面的放置，我们在主界面上放置了三个栏位：首页、安排和个人，并从中引出一些二级页面，这些页面各自都有自己的职能，在上面也都提到了，使用体验上还是比较简洁有效的。

在后台设计中，我们发现微信开发者工具中提供了云数据库，在微信公众平台可以查到操作数据库的函数接口，虽然不是 SQL 语句，但增删改查的这些思想是互通的。后台设计的主要工作是考虑需要哪些表，以及表中需要什么属性，在一番讨论之后我们设计出了相互有联系的几张表。因为当前用户太少，不方便演示，所以我们在虎扑爬取了一些 NBA 球员的数据下来，测试时输入相应的信息，即可匹配符合条件的球员。

我们还从合作开发中学会了使用 git 工具，以解决代码的同步问题。

这次作业使我们收获了很多，谢谢老师和助教的指导！